Fungible Weights in Logistic Regression

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jeff Alan Jones

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Niels Waller
Dissertation Advisor

Setempter 2013

©Jeff Alan Jones 2013

ALL RIGHTS RESERVED

Abstract

In this paper we discuss and develop methods for assessing parameter sensitivity in logistic regression models. To set the stage for this work, we review Waller’s (2008) equations for computing standardized fungible weights in multiple linear regression (MLR) and we then extend this work by showing how to compute unstandardized fungible weights. Next, we describe what factors affect fungible weight variability for a given model. To complete this section, we describe a likelihood based method for computing fungible weights that may be appropriate when standard MLR distributional assumptions are satisfied. Having reviewed fungible weights in linear regression, we next describe two methods for computing fungible weights in logistic regression. To demonstrate the utility of these methods, we computed fungible logistic weights for two well known datasets with binary outcome variables. To better understand what factors affect logistic fungible weights, we used Monte Carlo simulations to evaluate coefficient sensitivity in samples drawn from 64 populations with a wide range of data characteristics. Interestingly, our results showed that for many data-parameter configurations—e.g., latent coefficient of determination (R^2_{β}), structure of the predictor correlation matrix ($\mathbf{R}_{xx}$), orientation of the vector of regression coefficients with respect to the eigenvectors of $\mathbf{R}_{xx}$, and the number of predictors, p —that both methods are affected by these design factors in a similar manner. Specifically, R^2_{β} determined the size of the fungible weights and both $\mathbf{R}_{xx}$ and the orientation of β determined the shape of the fungible weight contours in $\mathbb{R}^p$. The number of predictors did not have an appreciable affect other than increasing the computation time of the two algorithms. Finally, we discuss future directions for research on fungible weights in other models.

Contents

Abstract	i
List of Tables	iv
List of Figures	xi
1 Introduction	1
2 Fungible Weights in Multiple Linear Regression	4
2.1 Derivation and Geometry	4
2.2 Extending Fungible Weights to Unstandardized Coefficients in Linear Regression: Ellipsoid Method	9
2.3 Extending Fungible Weights: Factors that Affect the Ellipsoid Method	12
2.3.1 Coefficient of Determination, R^2_{β}	12
2.3.2 Predictor Correlational Structure, $\mathbf{R}_{xx}$	17
2.3.3 Orientation of the Vectors of Standardized Regression Coefficients	19
2.3.4 Number of Predictors, p	23
2.4 Extending Fungible Weights to Unstandardized Coefficients in Linear Regression: Log-Likelihood Method	23
3 Fungible Weights in Multiple Logistic Regression	28
3.1 The Logistic Regression Model	28
3.2 Extending Fungible Weights to Logistic Regression: Ellipsoid Method	38
3.3 Extending Fungible Weights to Logistic Regression: Log-Likelihood Method	39
4 Re-analysis of Published Models	44
4.1 Aspirin Study	44
4.2 Smoking Study	47

4.3	Reanalysis of Published Models: Summary and Extensions	54
5	Simulation Design	56
5.1	Factors to be Varied	56
5.2	Data Generation for Fungible Weight Computation	58
6	Results	60
6.1	Simulation Study Results Comparing the Ellipsoid and Log-Likelihood Methods	60
6.1.1	Coefficient of Determination	60
6.1.2	Structure of the Predictor Correlation Matrix	71
6.1.3	Orientation of β With Respect to the Eigenvectors of $\mathbf{R}_{xx}$. .	80
6.1.4	Comparing the Ellipsoid and Log-Likelihood Methods	83
6.2	Examining the Effect of Incorrect Model Specification	94
7	Discussion/Future Directions	102
	References	109
	Appendix A: Figures from All Simulation Conditions	114
	Appendix B: R Code for Ellipsoid Method	243
	Appendix C: R Code for Log-Likelihood Method	246

List of Tables

1	Data from the Aspirin Component of the Physician's Health Study . . .	45
2	Logistic Regression Model for the Aspirin Study	45
3	Logistic Regression Model Reported by Adwere-Boamah (2009) . . .	48
4	Reproduced Logistic Regression Model	49
5	Two Different Sets of Fungible Weights for The Smoking Study	51
6	Coefficients for Models Generated from $R^2_{\beta} = \{.20, .80\}$ and $\mathbf{R}_{xx} =$ One-Factor	61
7	Coefficients for Models Generated from $R^2_{\beta} = .20$ and $\kappa = \{3, 9\}$. . .	72
8	Coefficients for Models Generated from $R^2_{\beta} = .80$ and $\kappa = 3$	80
9	Range of Fungible Weights for Models 17 and 20	81
10	Logistic Regression Coefficients for Models With $R^2_{\beta} = .20$ and $\mathbf{R}_{xx} =$ One-Factor	84
11	Correlations and Log-Likelihood Values for the Fungible Weights from Model 21	87
12	Logistic Regression Models for the Correct and Incorrect Specification	96

List of Figures

1	Ellipsoid and Plane Intersection	8
2	Fungible Weights as a Function of the Coefficient of Determination .	14
3	Scaled Fungible Weights as a Function of the Coefficient of Determina- tion	16
4	Fungible Weights as a Function of the Predictor Correlation Matrix .	18
5	Ellipsoid Defined by $\beta_i' \mathbf{R}_{xx} \beta_i = R_\beta^2$	20
6	Fungible Weights as a Function of the Orientation of β	22
7	Log-Likelihood Function.	25
8	Intersection of the Log-Likelihood Function and a Plane	26
9	Simple Logistic Regression Model	31
10	Newton-Raphson Algorithm	36
11	Fungible Weights for the Aspirin Study	46
12	Fungible Weight Contours for Smoking Study - Ellipsoid Method . . .	50
13	Fungible Weight Contours for Smoking Study - Log-Likelihood Method	53
14	Fungible Weight Contours for Models 1 and 5	62
15	Fungible Weight Contours for Models 2 and 6	65
16	Fungible Weight Contours for Models 3 and 7	66
17	Fungible Weight Contours for Models 4 and 8	67
18	Scaled Fungible Weight Contours for Models 1–8	69
19	Fungible Weight Contours for Models 9 and 13	73
20	Fungible Weight Contours for Models 10 and 14	77
21	Fungible Weight Contours for Models 11 and 15	78
22	Fungible Weight Contours for Models 12 and 16	79
23	Fungible Weight Contours for Models 17 and 20.	82
24	Fungible Weight Contours for Model 21	85
25	Fungible Weight Contours for Model 22	90

26	Fungible Weight Contours for Model 23	91
27	Fungible Weight Contours for Model 24	92
28	Ellipsoid Method for the Incorrect and Correct Model	97
29	Log-Likelihood Method for the Incorrect and Correct Model	100
30	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \mathbf{v}_1$	115
31	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$.	116
32	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$.	117
33	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \mathbf{v}_3$	118
34	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \mathbf{v}_1$. . .	119
35	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) =$ 30°	120
36	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) =$ 60°	121
37	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \mathbf{v}_3$. . .	122
38	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \mathbf{v}_1$	123
39	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$.	124
40	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$.	125
41	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \mathbf{v}_3$	126
42	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \mathbf{v}_1$. . .	127
43	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) =$ 30°	128
44	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) =$ 60°	129
45	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \mathbf{v}_3$. . .	130
46	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\beta \mathbf{v}_1$	131
47	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$	132
48	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$	133

49	Ellipsoid Method: $R_{\beta}^2 = .20; \kappa = 3; p = 3; \beta \mathbf{v}_3$	134
50	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 3; p = 3; \beta \mathbf{v}_1$	135
51	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 3; p = 3; \angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	136
52	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 3; p = 3; \angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	137
53	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 3; p = 3; \beta \mathbf{v}_3$	138
54	Ellipsoid Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \beta \mathbf{v}_1$	139
55	Ellipsoid Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \angle(\beta, \mathbf{v}_1) = 30^\circ$	140
56	Ellipsoid Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \angle(\beta, \mathbf{v}_1) = 60^\circ$	141
57	Ellipsoid Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \beta \mathbf{v}_3$	142
58	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \beta \mathbf{v}_1$	143
59	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	144
60	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	145
61	Log-Likelihood Method: $R_{\beta}^2 = .20; \kappa = 9; p = 3; \beta \mathbf{v}_3$	146
62	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \beta \mathbf{v}_1$	147
63	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \angle(\beta, \mathbf{v}_1) = 30^\circ$.	148
64	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \angle(\beta, \mathbf{v}_1) = 60^\circ$.	149
65	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \beta \mathbf{v}_3$	150
66	Log-Likelihood Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \beta \mathbf{v}_1$. . .	151
67	Log-Likelihood Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \angle(\beta, \mathbf{v}_1) =$ 30°	152
68	Log-Likelihood Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \angle(\beta, \mathbf{v}_1) =$ 60°	153
69	Log-Likelihood Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 1$ Factor; $p = 3; \beta \mathbf{v}_3$. . .	154
70	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 2$ Factor; $p = 3; \beta \mathbf{v}_1$	155
71	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 2$ Factor; $p = 3; \angle(\beta, \mathbf{v}_1) = 30^\circ$.	156
72	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 2$ Factor; $p = 3; \angle(\beta, \mathbf{v}_1) = 60^\circ$.	157
73	Ellipsoid Method: $R_{\beta}^2 = .80; \mathbf{R}_{xx} = 2$ Factor; $p = 3; \beta \mathbf{v}_3$	158

74	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \mathbf{v}_1$. . .	159
75	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$	160
76	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$	161
77	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \mathbf{v}_3$. . .	162
78	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta \mathbf{v}_1$	163
79	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$	164
80	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$	165
81	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta \mathbf{v}_3$	166
82	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta \mathbf{v}_1$	167
83	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	168
84	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	169
85	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta \mathbf{v}_3$	170
86	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta \mathbf{v}_1$	171
87	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$	172
88	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$	173
89	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta \mathbf{v}_3$	174
90	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta \mathbf{v}_1$	175
91	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	176
92	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	177
93	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta \mathbf{v}_3$	178
94	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_1$	179
95	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$.	180
96	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$.	181
97	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_3$	182
98	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_1$. . .	183

99	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$	184
100	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$	185
101	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_3$. . .	186
102	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_1$	187
103	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$.	188
104	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$.	189
105	Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_3$	190
106	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_1$. . .	191
107	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$	192
108	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$	193
109	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_3$. . .	194
110	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_1$	195
111	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$	196
112	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$	197
113	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_3$	198
114	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_1$	199
115	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$. . .	200
116	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$. . .	201
117	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_3$	202
118	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_1$	203
119	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$	204
120	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$	205
121	Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\boldsymbol{\beta} \mathbf{v}_3$	206

122	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\beta \mathbf{v}_1$	207
123	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	208
124	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	209
125	Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\beta \mathbf{v}_3$	210
126	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_1$	211
127	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$.	212
128	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$.	213
129	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_3$	214
130	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_1$. . .	215
131	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) =$ 30°	216
132	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) =$ 60°	217
133	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta \mathbf{v}_3$. . .	218
134	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta \mathbf{v}_1$	219
135	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$.	220
136	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$.	221
137	Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta \mathbf{v}_3$	222
138	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta \mathbf{v}_1$. . .	223
139	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) =$ 30°	224
140	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) =$ 60°	225
141	Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta \mathbf{v}_3$. . .	226
142	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\beta \mathbf{v}_1$	227
143	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$	228
144	Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$	229

145	Ellipsoid Method: $R_{\beta}^2 = .80; \kappa = 3; p = 5; \beta \mathbf{v}_3$	230
146	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 3; p = 5; \beta \mathbf{v}_1$	231
147	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 3; p = 5; \angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	232
148	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 3; p = 5; \angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	233
149	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 3; p = 5; \beta \mathbf{v}_3$	234
150	Ellipsoid Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \beta \mathbf{v}_1$	235
151	Ellipsoid Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \angle(\beta, \mathbf{v}_1) = 30^\circ$	236
152	Ellipsoid Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \angle(\beta, \mathbf{v}_1) = 60^\circ$	237
153	Ellipsoid Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \beta \mathbf{v}_3$	238
154	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \beta \mathbf{v}_1$	239
155	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \angle(\beta, \mathbf{v}_1) = 30^\circ$. . .	240
156	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \angle(\beta, \mathbf{v}_1) = 60^\circ$. . .	241
157	Log-Likelihood Method: $R_{\beta}^2 = .80; \kappa = 9; p = 5; \beta \mathbf{v}_3$	242

1 Introduction

Logistic regression is one of the “most commonly used method[s] for developing predictive models for dichotomous outcomes in medicine” (Tu, 1996, p. 1225), the social sciences (e.g., psychology: Pamper, 2000), and in machine learning research on classification (Perlich, Provost, & Simonoff, 2003). Virtually every statistical package (e.g., R, SAS, SPSS) has routines for estimating and evaluating logistic regression models. Moreover, model evaluative techniques such as residual diagnostics (Pregibon, 1981), statistical tests, and confidence intervals are widely available to users of logistic regression (e.g, Agresti, 2007; Menard, 2001). However, what is unavailable is a method of studying parameter sensitivity in these models. The goal of this paper is to develop and evaluate such a methodology.

Parameter sensitivity has been defined as the “relation of parameter changes to the fit of the model” (Green, 1977, p. 264). A common goal in parameter sensitivity analysis is to determine how much a model’s parameter estimates can change without drastically affecting the fit of the model. As noted by Green (1977), “if a parameter has little effect on the fit of a model, then it provides a weak basis for scientific conclusions (p. 264; for similar conclusions see Pek, 2012). Despite this cautionary note, logistic regression users frequently (e.g., Goldman, 2001; Menard, 2004) interpret coefficient values without conducting any such parameter sensitivity analysis. Yet one cannot fault them because no systematic way of analyzing parameter sensitivity exists for logistic regression. In multiple regression, parameter sensitivity has been studied in the form of alternate weighting methods (e.g., see Dana & Dawes, 2004; Rozeboom, 1979; Wainer, 1976, 1978; and Wilks, 1938) and statistical and geometrical analyses (Goldberger, 1968; Green, 1977; Koopman, 1988; Waller, 2008; Waller & Jones, 2009). Importantly, the recent geometrical work by Waller (2008; see also Waller & Jones, 2009) on “fungible weights” in standardized (i.e., all variables have zero means and unit variances) multiple regression provides

a template for developing a method of studying parameter sensitivity in logistic regression.

In his work on fungible weights, Waller (2008) showed that an infinite set of alternate weight vectors could be constructed to yield the *same* suboptimal model fit defined by the coefficient of determination, R^2 , and the sum of squared errors, SSE. In an example provided by Waller, these fungible weight vectors could often be quite different from the optimal weights (i.e., the Ordinary Least Squares, OLS, estimates), yet produce a coefficient of determination that is only .005 less than the optimal coefficient of determination (Waller, 2008, Table 1). This finding should be troubling to any researcher who interprets standardized regression coefficients as measures of relative importance or draws conclusions “on the basis of small differences in weights that, although possibly statistically significant, are of no practical consequence” (Green, 1977, p. 276).

Although Waller’s (2008) method of constructing fungible weights was developed for multiple *linear* regression, the method can be extended to other models. For example, fungible weights analysis has been extended to Latent Curve Models (MacCallum, Lee, & Browne, 2012) and Structural Equation Models (SEM, MacCallum, Browne, & Lee, 2009; Pek, 2012). In this paper, we extend the use of fungible weights to logistic regression models. The remainder of the paper is structured as follows. In Section 2, we review the mathematics and geometry of fungible weights in multiple regression and describe two extensions: (1) computing fungible weights for unstandardized regression coefficients, and (2) computing fungible weights from the log-likelihood function. Moreover, we examine what factors affect fungible weights in multiple regression. In Section 3, we present the logistic regression model and extend the fungible weights analyses to logistic regression coefficients. In Section 4, we reanalyze a few well-known datasets using our fungible weight methods to demonstrate the utility of parameter sensitivity analyses in

logistic regression. Next, in Section 5, we describe a simulation study that is designed to investigate what data-parameter configurations affect the fungible weights in logistic regression. Then, in Section 6, we summarize the results of our simulation study. Finally, in Section 7 we discuss our findings and present future research ideas on examining parameter sensitivity via fungible weights.

2 Fungible Weights in Multiple Linear Regression

2.1 Derivation and Geometry

In this section, we review the derivation and geometry of fungible weights in standardized multiple regression (Waller, 2008; Waller & Jones, 2009). To set notation, let $\mathbf{y}$ be an $n \times 1$ vector of standardized criterion scores (i.e., the criterion has been scaled to have zero mean and unit variance) and let $\mathbf{X}$ be an $n \times p$ full-rank matrix of standardized predictor scores. Then, the regression of $\mathbf{y}$ on $\mathbf{X}$ can be written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \tag{1}$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector of standardized regression coefficients, and $\mathbf{e}$ is an $n \times 1$ vector of errors. As is well known, the OLS estimator of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{r}_{\mathbf{xy}}, \tag{2}$$

where $\mathbf{R}_{\mathbf{xx}}$ denotes a $p \times p$ matrix of predictor correlations, and $\mathbf{r}_{\mathbf{xy}}$ denotes a $p \times 1$ vector of predictor-criterion correlations. Equation (2) minimizes the sums of squared errors, $\text{SSE}_{\boldsymbol{\beta}} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$, and maximizes the correlation ($R_{\boldsymbol{\beta}}$; and coefficient of determination $R_{\boldsymbol{\beta}}^2$) between the criterion, $\mathbf{y}$, and the predicted criterion, $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$. In other words, any alternate weight vector, $\boldsymbol{\alpha} \neq \boldsymbol{\beta}$, will yield an $\text{SSE}_{\boldsymbol{\alpha}} > \text{SSE}_{\boldsymbol{\beta}}$ and an $R_{\boldsymbol{\alpha}}^2 < R_{\boldsymbol{\beta}}^2$. In regression models with more than two predictors, Waller (2008) showed that “each pair of $\text{SSE}_{\boldsymbol{\alpha}}$ and $R_{\boldsymbol{\alpha}}^2$ values is associated with an infinite number of alternate [weight vectors]” (p. 691; notation was changed to be consistent with the current paper). Waller (2008) deemed these alternate weights *fungible* in the sense that they yield a constant $R_{\boldsymbol{\alpha}}^2$ and $\text{SSE}_{\boldsymbol{\alpha}}$. In the following paragraphs, we review Waller’s (2008) procedure for constructing fun-

gible weight vectors.

To begin, let $\hat{\mathbf{y}}_b = \mathbf{X}\boldsymbol{\beta}$ and $\hat{\mathbf{y}}_\alpha = \mathbf{X}\boldsymbol{\alpha}$ denote the predicted criterion scores using OLS weights and a set of alternate weights, respectively. Then, the correlation between these two composites is

$$r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\beta} = \frac{\boldsymbol{\alpha}' \mathbf{R}_{xx} \boldsymbol{\beta}}{(\boldsymbol{\alpha}' \mathbf{R}_{xx} \boldsymbol{\alpha})^{1/2} (\boldsymbol{\beta}' \mathbf{R}_{xx} \boldsymbol{\beta})^{1/2}} \quad (3)$$

(Green, 1977, p. 273). To generate an infinite set of fungible weight vectors, $\boldsymbol{\alpha}$, first decompose $\mathbf{R}_{xx}$ as

$$\mathbf{R}_{xx} = \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}', \quad (4)$$

where $\mathbf{V}$ is a $p \times p$ orthogonal matrix of eigenvectors, and $\boldsymbol{\Lambda}$ is a $p \times p$ diagonal matrix of ordered $(\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p)$ eigenvalues. If we substitute the right-hand-side of (4) into (3), we can transform the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ into unit-length vectors as

$$\mathbf{u} = \frac{\boldsymbol{\Lambda}^{1/2} \mathbf{V}' \boldsymbol{\beta}}{(\boldsymbol{\beta}' \mathbf{R}_{xx} \boldsymbol{\beta})^{1/2}}, \quad (5)$$

and

$$\mathbf{k} = \frac{\boldsymbol{\Lambda}^{1/2} \mathbf{V}' \boldsymbol{\alpha}}{(\boldsymbol{\alpha}' \mathbf{R}_{xx} \boldsymbol{\alpha})^{1/2}}, \quad (6)$$

such that the correlation between $\hat{\mathbf{y}}_\alpha$ and $\hat{\mathbf{y}}_\beta$ is simply

$$r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\beta} = \mathbf{k}' \mathbf{u}. \quad (7)$$

Given a $\mathbf{u}$, we can construct an infinite number of $\mathbf{k}$ that satisfy (3). To show this, let

$$\mathbf{k} = r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\beta} \mathbf{u} + \sqrt{1 - r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\beta}^2} \mathbf{M} \mathbf{z}, \quad (8)$$

where $\mathbf{M}$ is a $p \times (p-1)$ orthogonal matrix such that $\mathbf{M}' \mathbf{u} = \mathbf{0}$, and $\mathbf{z}$ is a $(p-1) \times 1$ random unit-length vector. Substituting the right-hand-side of (8) for $\mathbf{k}$ in (7) we

have

$$\begin{aligned}
\mathbf{k}'\mathbf{u} &= \left(r_{\hat{y}_\alpha \hat{y}_\beta} \mathbf{u} + \sqrt{1 - r_{\hat{y}_\alpha \hat{y}_\beta}^2} \mathbf{M} \mathbf{z} \right)' \mathbf{u} \\
&= r_{\hat{y}_\alpha \hat{y}_\beta} \mathbf{u}'\mathbf{u} + \sqrt{1 - r_{\hat{y}_\alpha \hat{y}_\beta}^2} \mathbf{z}' \mathbf{M}' \mathbf{u} \\
&= r_{\hat{y}_\alpha \hat{y}_\beta} \cdot 1 + \sqrt{1 - r_{\hat{y}_\alpha \hat{y}_\beta}^2} \cdot 0 \\
&= r_{\hat{y}_\alpha \hat{y}_\beta}.
\end{aligned}$$

Because an infinite number of $\mathbf{z}$ can be constructed, it follows that an infinite number of $\mathbf{k}$ can be constructed. Let $\mathbf{k}_i$ ($i = 1, \dots, \infty$) denote the i th vector among this infinite set. Given a $\mathbf{k}_i$, we can now construct the corresponding $\boldsymbol{\alpha}_i$. To do so, premultiply $\mathbf{k}_i$ by $\mathbf{V}\boldsymbol{\Lambda}^{-1/2}$,

$$\tilde{\boldsymbol{\alpha}}_i = \mathbf{V}\boldsymbol{\Lambda}^{-1/2} \mathbf{k}_i. \quad (9)$$

The transformation in (9) rotates (and scales) $\mathbf{k}_i$ to align with $\boldsymbol{\alpha}_i$. However, the rotation does not scale $\tilde{\boldsymbol{\alpha}}_i$ to minimize SSE_α . To minimize SSE_α , we must find the constant, s , such that

$$\text{SSE}_\alpha = (\mathbf{y} - \mathbf{X} s \tilde{\boldsymbol{\alpha}}_i)' (\mathbf{y} - \mathbf{X} s \tilde{\boldsymbol{\alpha}}_i) \quad (10)$$

is at a minimum. To do so, take the derivative of the right-hand-side of (10) with respect to s and set it to zero, such that

$$0 = -2\mathbf{y}'\mathbf{X}\tilde{\boldsymbol{\alpha}}_i + 2s\tilde{\boldsymbol{\alpha}}_i'\mathbf{X}'\mathbf{X}\tilde{\boldsymbol{\alpha}}_i. \quad (11)$$

Solving (11) for s yields

$$s = \frac{\mathbf{y}'\mathbf{X}\tilde{\boldsymbol{\alpha}}_i}{\tilde{\boldsymbol{\alpha}}_i'\mathbf{X}'\mathbf{X}\tilde{\boldsymbol{\alpha}}_i}, \quad (12)$$

which can be rewritten as

$$s = \frac{\mathbf{r}'_{xy} \tilde{\boldsymbol{\alpha}}_i}{\tilde{\boldsymbol{\alpha}}'_i \mathbf{R}_{xx} \tilde{\boldsymbol{\alpha}}_i}. \quad (13)$$

Using (13), we can compute $\boldsymbol{\alpha}_i$ as

$$\boldsymbol{\alpha}_i = s \tilde{\boldsymbol{\alpha}}_i,$$

which expands to

$$\boldsymbol{\alpha}_i = \left(\mathbf{r}'_{xy} \mathbf{V} \boldsymbol{\Lambda}^{-1/2} \mathbf{k}_i \right) \mathbf{V} \boldsymbol{\Lambda}^{-1/2} \mathbf{k}_i. \quad (14)$$

With (14) we can construct an infinite number of alternate weight vectors that yield the same $r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\beta}$ in (3). Moreover, as discussed by Waller (2008), this implies that

$$R_\alpha^2 = r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\alpha}^2 = \boldsymbol{\alpha}'_i \mathbf{R}_{xx} \boldsymbol{\alpha}_i \quad (15)$$

is constant for a given $r_{\hat{\mathbf{y}}_\alpha \hat{\mathbf{y}}_\beta}$. This fact has important geometrical implications, which we now review.

As discussed by Waller (2008; see also Waller & Jones, 2009), the quadratic form in (15) defines a hyper-ellipsoid in p -dimensional space. Moreover, it can be shown that the set of fungible weight vectors for a given R_α^2 must lie at the intersection of (15) with the bilinear form

$$\boldsymbol{\alpha}'_i \mathbf{R}_{xx} \boldsymbol{\beta} = R_\alpha^2, \quad (16)$$

that defines a $p - 1$ dimensional hyper-plane. The intersection of these two geometrical objects is illustrated in Figure 1.

Figure 1: Ellipsoid and Plane Intersection

The intersection of the ellipsoid, $\alpha_i' \mathbf{R}_{xx} \alpha_i = R_\alpha^2$, and the plane, $\alpha_i' \mathbf{R}_{xx} \beta = R_\alpha^2$.

Figure 1 illustrates the intersection of an ellipsoid with a plane in three dimensions. Coefficients $a1$, $a2$, and $a3$ correspond to fungible weights for OLS coefficients β_1 , β_2 , and β_3 , respectively. Importantly, the intersection contour in Figure 1 is a two-dimensional ellipse. In higher dimensions ($p > 3$), the intersection surface is defined by a $(p - 1)$ -dimensional hyper-ellipsoid (Ferguson, 1979). Moreover, as mentioned above, the weight vectors that define the intersection contour satisfy both (15) and (16), and therefore determine a set of fungible weight vectors that yield a constant R_α^2 (and SSE_α). Because of the geometrical implications, we refer to Waller's (2008) method of constructing fungible weights as the 'Ellipsoid

Method'. In the following section, we extend this approach to unstandardized regression coefficients.

2.2 Extending Fungible Weights to Unstandardized Coefficients in Linear Regression: Ellipsoid Method

In this section, we re-derive Waller's (2008) equations for fungible weights so that they are appropriate for unstandardized regression coefficients. To set notation, let $\mathbf{y}$ be an $n \times 1$ vector of mean-centered criterion scores (i.e., the criterion has been scaled to have zero mean) and let $\mathbf{X}$ be an $n \times p$ matrix of mean-centered predictor scores. Then, the regression of $\mathbf{y}$ on $\mathbf{X}$ can be written as

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}, \quad (17)$$

where $\mathbf{b}$ is a $p \times 1$ vector of unstandardized regression coefficients, and $\mathbf{e}$ is an $n \times 1$ vector of errors. As is well known, the OLS estimator of $\mathbf{b}$ is

$$\mathbf{b} = \Sigma_{xx}^{-1} \boldsymbol{\sigma}_{xy}, \quad (18)$$

where Σ_{xx} is a $p \times p$ matrix of predictor covariances, and $\boldsymbol{\sigma}_{xy}$ is a $p \times 1$ vector of predictor-criterion covariances. As discussed in Section 2.1, the OLS vector minimizes the sums of squared errors, $\text{SSE}_{\mathbf{b}}$, and maximizes the correlation, $R_{\mathbf{b}}$, between the criterion and the predicted criterion, $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$. Moreover, any alternate weight vector, $\mathbf{a} \neq \mathbf{b}$, is associated with an $\text{SSE}_{\mathbf{a}} > \text{SSE}_{\mathbf{b}}$ and an $R_{\mathbf{a}}^2 < R_{\mathbf{b}}^2$. For regression models with $p \geq 3$, an infinite number of alternate weight vectors, $\mathbf{a}_i$, yield the same $\text{SSE}_{\mathbf{a}}$ and $R_{\mathbf{a}}^2$. Following the notation and derivation outlined in the previous section, we now show how to construct a set of fungible weights for unstandardized regression coefficients. It will be seen that the derivation outlined in Section 2.1 holds for unstandardized regression coefficients by replacing the standardized parts

$\mathbf{R}_{xx}$, r_{xy} , β , and α with their unstandardized counterparts Σ_{xx} , σ_{xy} , $\mathbf{b}$, and $\mathbf{a}$.

To begin, let $\hat{\mathbf{y}}_b = \mathbf{X}\mathbf{b}$ and $\hat{\mathbf{y}}_a = \mathbf{X}\mathbf{a}$ denote the predicted criterion scores using OLS weights and alternate weights, respectively. Then, the correlation between these two composites can be computed as

$$r_{\hat{\mathbf{y}}_a \hat{\mathbf{y}}_b} = \frac{\mathbf{a}' \Sigma_{xx} \mathbf{b}}{(\mathbf{a}' \Sigma_{xx} \mathbf{a})^{1/2} (\mathbf{b}' \Sigma_{xx} \mathbf{b})^{1/2}} \quad (19)$$

(Green, 1977, p. 273). To generate the infinite set of fungible weight vectors, $\mathbf{a}$, first decompose Σ_{xx} as

$$\Sigma_{xx} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}', \quad (20)$$

where $\mathbf{U}$ is a $p \times p$ orthogonal matrix of eigenvectors, and $\mathbf{\Lambda}$ is a $p \times p$ diagonal matrix of ordered ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$) eigenvalues. By substituting the right-hand-side of (20) into (19), we can transform the vectors $\mathbf{a}$ and $\mathbf{b}$ into unit-length vectors,

$$\mathbf{g} = \frac{\mathbf{\Lambda}^{1/2} \mathbf{U}' \mathbf{b}}{(\mathbf{b}' \Sigma_{xx} \mathbf{b})^{1/2}} \quad (21)$$

and

$$\mathbf{h} = \frac{\mathbf{\Lambda}^{1/2} \mathbf{U}' \mathbf{a}}{(\mathbf{a}' \Sigma_{xx} \mathbf{a})^{1/2}}, \quad (22)$$

such that the correlation between $\hat{\mathbf{y}}_a$ and $\hat{\mathbf{y}}_b$ is

$$r_{\hat{\mathbf{y}}_a \hat{\mathbf{y}}_b} = \mathbf{h}' \mathbf{g}. \quad (23)$$

Given a $\mathbf{g}$, we can compute an infinite number of unit length vectors, $\mathbf{h}_i$ ($i = 1, \dots, \infty$), such that (23) holds. To show this, let

$$\mathbf{h}_i = r_{\hat{\mathbf{y}}_a \hat{\mathbf{y}}_b} \mathbf{g} + \sqrt{1 - r_{\hat{\mathbf{y}}_a \hat{\mathbf{y}}_b}^2} \mathbf{O} \mathbf{z}_i, \quad (24)$$

where $\mathbf{O}$ is a $p \times (p-1)$ orthogonal matrix such that $\mathbf{O}' \mathbf{g} = \mathbf{0}$, and $\mathbf{z}_i$ is a $(p-1) \times 1$

random unit-length vector. Because an infinite number of $\mathbf{z}_i$ can be constructed, it follows that an infinite number of $\mathbf{h}_i$ can be constructed. To construct an $\mathbf{a}_i$ from an $\mathbf{h}_i$, first premultiply $\mathbf{h}_i$ by $\mathbf{U}\mathbf{\Lambda}^{-1/2}$, so that

$$\tilde{\mathbf{a}}_i = \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{h}_i. \quad (25)$$

The transformation in (25) rotates (and scales) $\mathbf{h}_i$ to align with an $\mathbf{a}_i$. However, (25) is not scaled to minimize $\text{SSE}_{\mathbf{a}}$. To minimize $\text{SSE}_{\mathbf{a}}$, we multiply (25) by a scaling constant

$$s = \boldsymbol{\sigma}'_{xy} \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{h}_i, \quad (26)$$

such that

$$\begin{aligned} \mathbf{a}_i &= s\tilde{\mathbf{a}}_i \\ &= \left(\boldsymbol{\sigma}'_{xy} \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{h}_i \right) \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{h}_i. \end{aligned} \quad (27)$$

As in Section 2.1, a constructed alternate weight vector, $\mathbf{a}$, produces a linear composite, $\hat{\mathbf{y}}_{\mathbf{a}}$, that correlates $r_{\hat{\mathbf{y}}_{\mathbf{a}}\hat{\mathbf{y}}_{\mathbf{b}}}$ with the predicted criterion $\hat{\mathbf{y}}_{\mathbf{b}}$ and $R_{\mathbf{a}}$ with the criterion $\mathbf{y}$. Not surprisingly, equations (19)–(27) mirror those from Section 2.1. However, the fungible weight vectors produced in this section are associated with unstandardized regression coefficients rather than standardized. Interestingly, we can use the fungible weights for unstandardized regression coefficients to compute fungible weights for the intercept term, b_0 , if the predictor and criterion means are available. To demonstrate this, let $\mathbf{1}$ denote an $n \times 1$ vector of ones. Then, for a given alternate weight vector, $\mathbf{a}$, we seek an intercept, a_0 , such that

$$\text{SSE}_{\mathbf{a}} = (\mathbf{y} - a_0\mathbf{1} - \mathbf{X}\mathbf{a})'(\mathbf{y} - a_0\mathbf{1} - \mathbf{X}\mathbf{a}) \quad (28)$$

is at a minimum. To find a_0 , take the derivative of (28) with respect to a_0 and set

it to zero as

$$0 = -2\mathbf{y}'\mathbf{1} + 2a_0\mathbf{1}'\mathbf{1} + 2\mathbf{1}'\mathbf{X}\mathbf{a}. \quad (29)$$

Solving for a_0 yields

$$a_0 = \mu_y - \boldsymbol{\mu}_x' \mathbf{a}, \quad (30)$$

where μ_y denotes the mean of $\mathbf{y}$, and $\boldsymbol{\mu}_x$ denotes the $p \times 1$ vector of predictor means. Notice that (30) resembles the equation for computing the OLS intercept except that $\mathbf{b}$ has been replaced with $\mathbf{a}$. Taken together, the vector $\begin{bmatrix} a_0 & \mathbf{a} \end{bmatrix}'$ is a fungible weight vector for regression coefficients computed on raw data. The R function in Appendix B computes fungible weight vectors for a desired R_a^2 using the method outlined in this section. If predictor and criterion means are supplied, the function will also compute fungible intercepts.

2.3 Extending Fungible Weights: Factors that Affect the Ellipsoid Method

In this section, we examine how different aspects of the data affect fungible weights computed from the Ellipsoid Method. Specifically, we examine: (1) the coefficient of determination between the criterion and the predictors, R_β^2 , (2) correlational structure of the predictors, $\mathbf{R}_{xx}$, (3) orientation of $\boldsymbol{\beta}$ with respect to the eigenvectors of $\mathbf{R}_{xx}$, and (4) number of predictors, p .

2.3.1 Coefficient of Determination, R_β^2

The first factor we discuss is the coefficient of determination, R_β^2 . As shown by Waller (2008, p. 697, Eq. 24), R_β^2 is linked to the correlation between two linear composites, $r_{\hat{\mathbf{y}}_\alpha, \hat{\mathbf{y}}_\beta}$, as

$$r_{\hat{\mathbf{y}}_\alpha, \hat{\mathbf{y}}_\beta} = \left(1 - \frac{R_\beta^2 - R_\alpha^2}{R_\beta^2} \right)^{1/2}, \quad (31)$$

which can be rewritten as

$$r_{\hat{y}_\alpha \hat{y}_\beta} = \left(1 - \frac{\theta}{R_\beta^2}\right)^{1/2}, \quad (32)$$

where θ denotes a user-defined decrement in model fit. Further, if $\theta = R_\beta^2 - R_\alpha^2$ is held constant, then all else being equal, a larger R_β^2 leads to larger fungible weight contours. For example, consider two regression models that correspond to a predictor correlation matrix of

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.00 & -0.24 & 0.33 \\ -0.24 & 1.00 & -0.59 \\ 0.33 & -0.59 & 1.00 \end{bmatrix},$$

and two coefficients of determination, $R_\beta^2 = .30$ and $R_\beta^2 = .90$. Further, suppose the model with $R_\beta^2 = .30$ has $\beta = \begin{bmatrix} .39 & .39 & .39 \end{bmatrix}'$, and the model with $R_\beta^2 = .90$ has $\beta = \begin{bmatrix} .67 & .67 & .67 \end{bmatrix}'$. To generate fungible weight contours for these models, we set $\theta = R_\beta^2 - R_\alpha^2 = .10$ such that $R_\alpha^2 = .20$ for the first model and $R_\alpha^2 = .80$ for the second model. Figure 2 illustrates how larger R_β^2 leads to larger fungible weight contours.

Figure 2 displays three panels: Panel A displays two fungible weight contours for alternate weights, a_1 and a_2 ; Panel B displays two fungible weight contours for alternate weights a_1 and a_3 ; Panel C displays two fungible weight contours for alternate weights a_2 and a_3 . The blue fungible weight contours correspond to the linear regression model with an $R_\beta^2 = .90$; the red fungible weight contours correspond to the linear regression model with an $R_\beta^2 = .30$. Notice that in all panels, the blue fungible weight contours are larger than the corresponding red fungible weight contours. Though the blue fungible weight contours are larger than the red, suggesting more model sensitivity, the composites created by the blue fungible weight vectors are more highly related to the OLS composite than the composite created by the

Figure 2: Fungible Weights as a Function of the Coefficient of Determination

Panels A–C show three different pairs of fungible weights for two different values of R_a^2 .

red fungible weights vectors. Specifically, $r_{\hat{y}_\alpha \hat{y}_\beta} = .94$ for the blue fungible contour, and $r_{\hat{y}_\alpha \hat{y}_\beta} = .82$ for the red fungible contour,

Though interesting, the above finding is not too surprising given the fact the regression coefficients from the $R^2_\beta = .90$ condition are nearly double the size of the coefficients from the $R^2_\beta = .30$ condition. Because larger R^2_β necessarily increases the size of the regression coefficients, the fungible weight contours also increase in size. Moreover, if we examine how the fungible weights vary *relative* to their OLS counterparts, we see a similar effect. That is, models with larger coefficients of determination produce less variable fungible weights. Figure 3 illustrates this point.

Figure 3: Scaled Fungible Weights as a Function of the Coefficient of Determination

Panels A–C show three pairs of scaled fungible weights for two different values of R_β^2 .

Figure 3 displays the same three panels as Figure 2 with the red fungible weight

contours corresponding to the linear regression model with an $R^2_{\beta} = .30$ and the blue fungible weight contours corresponding to the linear regression model with an $R^2_{\beta} = .90$. However, instead of plotting the absolute size of the fungible weights (as we did in Figure 2), we have now plotted the relative size of the fungible weights. Specifically, we have divided each fungible weight by its corresponding OLS coefficient. Dividing out the OLS coefficient allows us to examine how the fungible weights vary relative to their generating OLS coefficient. Interestingly, Figure 3 shows that once we have divided out the OLS coefficients, the fungible weight contours from the $R^2_{\beta} = .90$ condition are all smaller than the contours from the $R^2_{\beta} = .30$ condition. Taken together, the findings in Figures 2 and 3 suggest that larger R^2_{β} lead to larger fungible weight contours in an *absolute* sense, but lead to smaller fungible weight contours in a *relative* sense.

2.3.2 Predictor Correlational Structure, $\mathbf{R}_{xx}$

The second factor we discuss is the correlational structure of the predictors. Under the Ellipsoid Method, the predictor correlation matrix, $\mathbf{R}_{xx}$, is the coefficient matrix of the hyper-dimensional ellipsoid defined by $\boldsymbol{\alpha}'_i \mathbf{R}_{xx} \boldsymbol{\alpha}_i = R^2_{\alpha}$, where $i = 1, \dots, \infty$ (see Section 2.1). The coefficient matrix determines both the shape and orientation of the hyper-ellipsoid. In effect, the correlation matrix also determines the shape of the fungible weight contours. For example, consider two regression models with $R^2_{\beta} = .50$ and $\boldsymbol{\beta} = \begin{bmatrix} .486 & .004 & .490 \end{bmatrix}'$. However, suppose the first model corresponds to a correlation matrix of

$$\mathbf{R}_{xx}^{(1)} = \begin{bmatrix} 1.00 & -0.05 & 0.05 \\ -0.05 & 1.00 & 0.05 \\ 0.05 & 0.05 & 1.00 \end{bmatrix},$$

whereas the second model corresponds to a correlation matrix of

$$\mathbf{R}_{xx}^{(2)} = \begin{bmatrix} 1.00 & -0.91 & 0.06 \\ -0.91 & 1.00 & -0.42 \\ 0.06 & -0.42 & 1.00 \end{bmatrix}.$$

(see Waller, 2013, for an algorithm to generate correlation matrices that satisfy $\boldsymbol{\beta}' \mathbf{R}_{xx} \boldsymbol{\beta} = R_{\beta}^2$ for a given R_{β}^2 and $\boldsymbol{\beta}$). To generate fungible weights for these models, we set $\theta = .005$, such that $R_{\alpha}^2 = .495$. Figure 4 illustrates how different correlation matrices affect the fungible weights.

Figure 4: Fungible Weights as a Function of the Predictor Correlation Matrix

Two fungible weight contours created from the same vector of regression coefficients, $\boldsymbol{\beta} = [.486 \ .004 \ .490]'$, but different predictor correlation matrices, $\mathbf{R}_{xx}$. The red fungible weight contour was generated using an $\mathbf{R}_{xx}$ with eigenvalues = $\{1.05, 1.05, .90\}$, whereas the blue fungible weight contour was generated using an $\mathbf{R}_{xx}$ with eigenvalues = $\{2.03, .95, .02\}$.

Figure 4 displays two fungible weight contours. The red fungible weight contour

corresponds to the $\mathbf{R}_{xx}^{(1)}$ condition, whereas the blue fungible weight contour corresponds to the $\mathbf{R}_{xx}^{(2)}$ condition. Note that both contours were generated from the same coefficient of determination and vector of standardized regression coefficients. However, the shapes and orientations of the contours are different. This difference is determined by the correlational structure of the predictors. Specifically, the eigenvalues of the correlation matrix determine the shape of the contours, whereas the eigenvectors determine their orientation. The eigenvalues of $\mathbf{R}_{xx}^{(1)}$ are $\{1.05, 1.05, 0.90\}$ whereas the eigenvalues of $\mathbf{R}_{xx}^{(2)}$ are $\{2.03, 0.95, 0.02\}$. Importantly, the size of the eigenvalues determines the shape of the hyper-ellipsoid defined by $\boldsymbol{\alpha}'_i \mathbf{R}_{xx} \boldsymbol{\alpha}_i = R_{\alpha}^2$ via the equation for the semi-axis lengths

$$l_i = \sqrt{\frac{R_{\alpha}^2}{\lambda_i}},$$

where l_i denotes the i th semi-axis length, and λ_i denotes the i th ordered eigenvalue ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$) of $\mathbf{R}_{xx}$ (see Hadi, 1996, Ch. 10; Waller & Jones, 2011).

Note that larger eigenvalues are associated with smaller semi-axis lengths, whereas smaller eigenvalues are associated with larger semi-axis lengths. Using this information, we know that the ellipsoid defined by the first correlation matrix is roughly spherical (because the eigenvalues are all close in size), whereas the ellipsoid defined by the second correlation matrix is cigar-shaped. The two fungible weight contours in Figure 4 lie at the intersection of these two ellipsoids with the hyper-planes defined by $\boldsymbol{\alpha}'_i \mathbf{R}_{xx} \boldsymbol{\beta} = R_{\alpha}^2$ (see (16)). Because of the different correlational structure of the two examples, the intersection results in different fungible weight contours.

2.3.3 Orientation of the Vectors of Standardized Regression Coefficients

The third factor we discuss is the orientation of the vectors of standardized regression coefficients, $\boldsymbol{\beta}$ (which in turn varies the predictor-criterion correlations, $\mathbf{r}_{xy} = \mathbf{R}_{xx} \boldsymbol{\beta}$) with respect to the eigenvectors of $\mathbf{R}_{xx}$. To better understand this factor, we first discuss the geometry of multiple regression and hyper-dimensional

ellipsoids. As reviewed by Waller & Jones (2011), for a fixed $\mathbf{R}_{xx}$ and R_β^2 , there are an infinite number of β_i ($i = 1, \dots, \infty$) that satisfy

$$\beta_i' \mathbf{R}_{xx} \beta_i = R_\beta^2. \quad (33)$$

As discussed earlier, for fixed $\mathbf{R}_{xx}$ and R_β^2 , the terminal points of β_i in (33) define a p -dimensional hyper-ellipsoid (Waller & Jones, 2011). Figure 5 illustrates this point.

Figure 5: Ellipsoid Defined by $\beta_i' \mathbf{R}_{xx} \beta_i = R_\beta^2$

The ellipsoid of all possible vectors of standardized regression coefficients that satisfy $\beta_i' \mathbf{R}_X \beta_i = R_\beta^2$. The vectors β_1 and β_2 are highlighted to illustrate this point.

The ellipsoid in Figure 5 is composed of the endpoints of all standardized regression coefficient vectors, β_i , that satisfy (33) for a given $\mathbf{R}_{xx}$ and R_β^2 . Two co-efficient vectors of this infinite set, β_1 and β_2 , have been highlighted. Note that because each β_i is paired with a distinct predicted criterion (because $\hat{\mathbf{y}}_i = \mathbf{X}\beta_i$), (33) can also be thought of as the geometrical representation of all possible regression models for a set of standardized fixed predictors and a constant R_β^2 (Waller

& Jones, 2011). As can be seen from Figure 5, different values of β are associated with different levels of curvature on the hyper-ellipsoid. This fact implies that the location of β and the structure of $\mathbf{R}_{xx}$ jointly affects fungible weight contours computed from the Ellipsoid Method. To demonstrate this point, we examine two linear regression models that differ only in terms of the orientation of β .

Suppose that we have two models, each of which satisfy the same correlation matrix

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.00 & -0.49 & 0.49 \\ -0.49 & 1.00 & 0.49 \\ 0.49 & 0.49 & 1.00 \end{bmatrix},$$

with eigenvalues = $\{1.49, 1.49, 0.02\}$ and coefficient of determination of $R_\beta^2 = .10$.

Further, suppose that in the first model β is collinear with the first eigenvector of $\mathbf{R}_{xx}$, such that $\beta || \mathbf{v}_1 = \begin{bmatrix} 0.16 & 0.31 & -0.41 \end{bmatrix}'$, and in the second model β is collinear with the last eigenvector of $\mathbf{R}_{xx}$, such that $\beta || \mathbf{v}_3 = \begin{bmatrix} 3.34 & 3.34 & -3.34 \end{bmatrix}'$. To generate fungible weights for these models, we set $\theta = .01$, such that $R_\alpha^2 = .09$.

Figure 6 illustrates how the different orientations of β affect the fungible weight contours for this example.

Figure 6: Fungible Weights as a Function of the Orientation of β

Both panels display the same two fungible weight contours created from the same $\mathbf{R}_{xx}$, $R_{\beta}^2 = .10$, and $\theta = .01$. The red fungible weight contour corresponds to β collinear with the first eigenvector of $\mathbf{R}_{xx}$, $\beta||\mathbf{v}_1 = [0.16 \ 0.31 \ -0.41]'$, whereas the blue fungible weight contour corresponds to β collinear with the last eigenvector of $\mathbf{R}_{xx}$, $\beta||\mathbf{v}_3 = [3.34 \ 3.34 \ -3.34]'$.

The two panels in Figure 6 display different viewpoints of two fungible weight contours. The red fungible weight contour corresponds to the model with $\beta||\mathbf{v}_1 = [0.16 \ 0.31 \ -0.41]'$, whereas the blue fungible weight contour corresponds to the model with $\beta||\mathbf{v}_3 = [3.34 \ 3.34 \ -3.34]'$. The left panel of Figure 6 displays these two contours from the viewpoint of the first eigenvector; the right panel of Figure 6 displays these two contours from the viewpoint of the last eigenvector. Notice that the contour associated with the first eigenvector is long and narrow, and resembles a cigar, whereas the contour associated with the last eigenvector is circular. These shapes are a result of both the structure of the correlation matrix and the orientation of β .

2.3.4 Number of Predictors, p

The fourth factor we discuss is the number of predictors, p . Though the number of predictors should not affect the fungible weight contours directly, the number of predictors should have an indirect effect via the pattern of predictor correlations. For a given condition number, $\kappa = \sqrt{\lambda_1/\lambda_p}$ (a popular index of multicollinearity; see Cohen, Cohen, West, & Aiken, 2003, p. 424), the pattern of predictor correlations is limited by the number of predictors (and in the two predictor case, the condition number completely determines the correlation). Because the number of predictors affects the correlation matrix, the number of predictors also (in part) determines the shape and orientation of the fungible weight contours.

2.4 Extending Fungible Weights to Unstandardized Coefficients in Linear Regression: Log-Likelihood Method

In the previous section, we discussed how to compute fungible weights for unstandardized regression coefficients using the Ellipsoid Method. Unfortunately, to use the Ellipsoid Method the number of regressors must be greater than two.¹ In this section, we describe a method that permits the estimation of fungible unstandardized regression weights with only one predictor—but it comes with a cost. Specifically, if we impose the assumption that the errors are normally distributed, then we can use the maximum likelihood (ML) fit function to estimate fungible weights.

Let L denote the likelihood function,

$$L = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp \left[-\frac{1}{2} \frac{(y_i - b_0 - \mathbf{b}'\mathbf{x}_i)^2}{\sigma_e^2} \right], \quad (34)$$

¹Technically, the ellipsoid approach can be used with two predictors; however, there are only two solutions for any suboptimal R_a^2 and associated SSE_a . For a single predictor any alternate coefficient will yield the optimal R_b^2 , but a suboptimal SSE.

and let L_{ML} denote the maximum of the function. As is well known (Rencher, 2008, pp. 158–159), the maximum likelihood estimates of the regression coefficients are equivalent to the OLS estimates discussed earlier. Typically, when computing the maximum likelihood estimates, one works with the log-likelihood, rather than the likelihood function. Let $\ln L$ denote the log-likelihood function,

$$\ln L = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma_e^2) - \frac{1}{2}\sum_{i=1}^n \left[\frac{(y_i - b_0 + \mathbf{b}'\mathbf{x}_i)^2}{\sigma_e^2} \right], \quad (35)$$

and let $\ln L_{ML}$ represent the maximum of this function. An example of a log-likelihood function for a single predictor regression model is presented in Figure 7.

Figure 7: Log-Likelihood Function.

The log-likelihood function for various values of the regression coefficients. The yellow dot indicates the maximum likelihood estimates of b_0 and b_1 .

Figure 7 displays the log-likelihood function in (35) for the simple linear regression model

$$\hat{y} = 1 + 2x, \quad (36)$$

for various values of b_0 and b_1 .² Notice that the log-likelihood reaches a maximum

²Technically the log-likelihood function depends on b_0 , b_1 , and σ_e^2 . The log-likelihood function values were computed by varying all three parameters, but we have only plotted the log-likelihood function with respect to the b_0 and b_1 values.

at the yellow dot. This point corresponds to the ML (and OLS) estimates, $b_0 = 1$ and $b_1 = 2$, of the unstandardized regression coefficients. Note, for models that are considered in this paper, the ML estimator is unique. However, moving slightly off of the maximum, there exists an infinite number of b_0 , b_1 , and σ_e^2 that yield the same log-likelihood value, $\ln L_f < \ln L_{ML}$, where f stands for fungible. This point is demonstrated in Figure 8.

Figure 8: Intersection of the Log-Likelihood Function and a Plane

The log-likelihood function for various values of the regression coefficients. The intersection of the plane and the log-likelihood function reveals an infinite set of fungible weight vectors.

Figure 8 displays the intersection of the log-likelihood function and a plane. Notice that the plane is set at $\ln L_f < \ln L_{ML}$. Further notice that the intersection of the log-likelihood function with the plane yields an intersection contour. This intersection is conceptually equivalent to the intersection ellipse formed by the ellipsoid and plane in Figure 1. Moreover, just as the intersection ellipse yielded an infinite number of fungible weights with constant $R_{\mathbf{a}}$, the intersection contour formed by the log-likelihood function and a plane yields an infinite number of fungible weights with constant $\ln L_f$. In Section 3.3, we describe an optimization algorithm to generate fungible sets of coefficients for a given log-likelihood value $\ln L_f < \ln L_{ML}$.

3 Fungible Weights in Multiple Logistic Regression

In the previous section, we described two algorithms for computing fungible weights in multiple regression. Specifically, we described Waller’s (2008) Ellipsoid Method for computing fungible weights in multiple regression, and we described a Log-Likelihood Method for computing fungible weights in multiple regression. Below we extend these two methods to logistic regression.

3.1 The Logistic Regression Model

In this section, we describe the logistic regression model for predicting a binary outcome, y_i , from a set of regressors, $\mathbf{x}_i$. Specifically, we discuss how to set up the model, interpret the model coefficients, and estimate the model via maximum likelihood estimation.

One way to conceptualize the logistic regression model (see Greene, 2003, pp. 666–669) is to assume that there exists a latent variable, y_i^* , that underlies an observed binary variable, y_i . Further, it is assumed that the relationship between y_i^* and a $p \times 1$ vector of predictors, $\mathbf{x}_i$, is linear in form,

$$y_i^* = b_0 + \mathbf{b}'\mathbf{x}_i + \epsilon_i, \tag{37}$$

where b_0 is an intercept, $\mathbf{b}$ is an $p \times 1$ vector of coefficients, and ϵ_i is a latent random variable with a standard logistic distribution³,

$$f(\epsilon_i) = \frac{\exp(\epsilon_i)}{[1 + \exp(\epsilon_i)]^2},$$

with location parameter, $\mu = 0$, and scale parameter, $s = 1$. However, in real-data situations, we have to work with y_i rather than the unknown y_i^* . To relate the

³Note that if ϵ_i was assumed to have a standard normal distribution we would then be working with the Probit regression model.

binary response, y_i , to the the vector of predictors, $\mathbf{x}_i$, we first specify

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (38)$$

Using (38) and (37) we can model the probability of $y_i = 1|\mathbf{x}_i$ as

$$P(y_i = 1|\mathbf{x}_i) = P(y_i^* > 0|\mathbf{x}_i), \quad (39)$$

which can be rewritten as

$$P(y_i = 1|\mathbf{x}_i) = P(b_0 + \mathbf{b}'\mathbf{x}_i + \epsilon_i > 0). \quad (40)$$

Next, rewrite the right-hand-side of (40) as

$$P(b_0 + \mathbf{b}'\mathbf{x}_i + \epsilon_i > 0) = P(\epsilon_i > -b_0 - \mathbf{b}'\mathbf{x}_i), \quad (41)$$

which simplifies to

$$P(\epsilon_i < b_0 + \mathbf{b}'\mathbf{x}_i), \quad (42)$$

due to the symmetry of the logistic distribution. Because ϵ_i is distributed standard logistic, we have

$$P(\epsilon_i < b_0 + \mathbf{b}'\mathbf{x}_i) = \int_{-\infty}^{b_0 + \mathbf{b}'\mathbf{x}_i} \left(\frac{\exp(\epsilon_i)}{[1 + \exp(\epsilon_i)]^2} \right) d\epsilon_i = \frac{\exp(b_0 + \mathbf{b}'\mathbf{x}_i)}{1 + \exp(b_0 + \mathbf{b}'\mathbf{x}_i)}. \quad (43)$$

Therefore, the probability that y_i equals 1 conditional on the response vector, $\mathbf{x}_i$, is simply

$$P(y_i = 1|\mathbf{x}_i) = \frac{\exp(b_0 + \mathbf{b}'\mathbf{x}_i)}{1 + \exp(b_0 + \mathbf{b}'\mathbf{x}_i)}. \quad (44)$$

Although we formalized the logistic regression model as having two underlying la-

tent variables (y_i^* and ϵ_i), other formalizations are possible. For example, the end result (i.e., Equation 44) can also be obtained via the general linear model setup by assuming that y_i is a Bernoulli (or Binomial) random variable and using the logit link function (see Agresti, 2002, p. 121–123).

With the logistic regression model described, we now move onto coefficient interpretation. To begin, consider the simple logistic regression model,

$$P(y_i = 1|x_i) = \frac{\exp(b_0 + b_1 x_i)}{1 + \exp(b_0 + b_1 x_i)}. \quad (45)$$

Because of the nonlinear relationship between y_i and x_i , the interpretation of b_1 in (45) is different from the simple linear regression model. For simple linear regression, b_1 is interpreted as the expected change in the response for a unit increase in the predictor. For simple logistic regression, b_1 cannot be interpreted this way because the expected change in the response depends on where the unit increase takes place along the predictor. Figure 9 illustrates this point.

Figure 9: Simple Logistic Regression Model

This simple logistic regression model has $b_0 = 0$ and $b_1 = 1$.

Figure 9 displays a logistic regression model with $b_0 = 0$ and $b_1 = 1$. If we evaluate (45) with these parameter estimates at $x = 0$ and $x = 1$, we find that the estimated probability increases from 0.5 (when $x = 0$) to 0.73 (when $x = 1$), a difference of 0.23. However, if we evaluate (45) at $x = 1$ and $x = 2$ (still a unit difference), we find that the estimated probability increases from 0.73 (when $x = 1$) to 0.88 (when $x = 2$), a difference of 0.15. Because the change in the estimated probability depends on where the unit increase in the predictor takes place, we cannot

interpret b_1 as we would in a simple linear regression.

A common method of interpreting the coefficients is by using the odds ratio. To compute an odds ratio, we first rewrite (44) as

$$\frac{P(y_i = 1|x_i)}{1 - P(y_i = 1|x_i)} = \text{odds}[P(y_i = 1|x_i)] = \exp(b_0 + b_1 x_i). \quad (46)$$

Using (46) one can compute the odds ratio of the estimated probability for a unit increase in x_i by evaluating

$$\frac{\text{odds}[P(y_i = 1|x_i = x_i + 1)]}{\text{odds}[P(y_i = 1|x_i = x_i)]} = \frac{\exp[b_0 + b_1(x_i + 1)]}{\exp[b_0 + b_1(x_i)]}, \quad (47)$$

which simplifies to

$$\frac{\exp[b_0 + b_1(x_i + 1)]}{\exp[b_0 + b_1 x_i]} = \frac{\exp(b_0) \cdot \exp(b_1 x_i) \cdot \exp(b_1)}{\exp(b_0) \cdot \exp(b_1 x_i)} = \exp(b_1). \quad (48)$$

Equation (48) shows that the odds of $y_i = 1$ (relative to the odds of $y_i = 0$) increase by a multiplicative factor of $\exp(b_1)$ for every unit increase in x_i . As shown by (48), this increase is constant no matter where the unit increase in x_i takes place. Though the above description took place under a simple logistic regression model, the extension to the multiple logistic regression is straightforward. As in multiple linear regression, one usually interprets a single coefficient by holding all other predictors constant. So, for a unit increase in $x_{1,i}$ while holding all other predictors constant, the odds of $y_i = 1$ increase by multiplicative factor of $\exp(b_1)$. With that said, we now move on to model estimation.

Unlike OLS estimation (or ML estimation) for linear regression coefficients, a closed form solution for logistic regression coefficients does not exist. One method of estimating the coefficients is to use an optimization algorithm to obtain the maximum of the likelihood equation,

$$L(\mathbf{b}) = \prod_{i=1}^n P(y_i = 1|\mathbf{x}_i)^{y_i} [1 - P(y_i = 1|\mathbf{x}_i)]^{1-y_i}, \quad (49)$$

where $P(y_i = 1|\mathbf{x}_i)$ is shown in (44) (alternatively, one could use iteratively re-weighted least squares estimation, Agresti, 2009, p. 195). As in linear regression, one generally maximizes the log-likelihood equation,

$$\ln L(\mathbf{b}) = \sum_{i=1}^n y_i \ln [P(y_i = 1|\mathbf{x}_i)] + (1 - y_i) \ln [1 - P(y_i = 1|\mathbf{x}_i)], \quad (50)$$

which, after some algebra, reduces to

$$\ln L(\mathbf{b}) = \sum_{i=1}^n y_i [b_0 + \mathbf{b}'\mathbf{x}_i] - \ln [1 + \exp(b_0 + \mathbf{b}'\mathbf{x}_i)]. \quad (51)$$

One popular method of maximization is the Newton-Raphson algorithm (Khuri, 2003, p. 331), although other methods exist (e.g., Fisher Scoring: Jennrich & Sampson, 1976; Gradient Ascent/Descent: Khuri, 2003, p. 330). The Newton-Raphson algorithm is a generic procedure used to find the minimum/maximum of a (possibly vector-valued) function. We briefly review this algorithm before applying it to the problem at hand.

Suppose we are interested in the maximum of a vector-valued function, $f(\boldsymbol{\gamma})$, where $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_p]'$. To find the maximum, first approximate $f(\boldsymbol{\gamma})$ in the neighborhood of $\boldsymbol{\gamma}^{(0)}$ using a second-order Taylor Series Expansion (Stein & Barcellos, 1992, p. 624), $g(\boldsymbol{\gamma})$, as

$$g(\boldsymbol{\gamma}) = f(\boldsymbol{\gamma}^{(0)}) + (\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(0)})' \nabla f(\boldsymbol{\gamma}^{(0)}) + \frac{1}{2} (\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(0)})' \mathbf{H}(\boldsymbol{\gamma}^{(0)}) (\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(0)}), \quad (52)$$

where $\nabla f(\boldsymbol{\gamma}^{(0)})$ denotes the $p \times 1$ gradient vector of $f(\boldsymbol{\gamma})$ with respect to $\boldsymbol{\gamma}$ evaluated at $\boldsymbol{\gamma}^{(0)}$, and $\mathbf{H}(\boldsymbol{\gamma}^{(0)})$ denotes the $p \times p$ Hessian matrix of $f(\boldsymbol{\gamma})$ with respect to

γ evaluated at $\gamma^{(0)}$. Then, rewrite (52) as

$$\begin{aligned} g(\gamma) = & f(\gamma^{(0)}) + \gamma' \nabla f(\gamma^{(0)}) - \gamma^{(0)'} \nabla f(\gamma^{(0)}) + \frac{1}{2} \gamma' \mathbf{H}(\gamma^{(0)}) \gamma - \\ & \gamma^{(0)'} \mathbf{H}(\gamma^{(0)}) \gamma + \frac{1}{2} \gamma^{(0)'} \mathbf{H}(\gamma^{(0)}) \gamma^{(0)}. \end{aligned} \quad (53)$$

Because (53) is a quadratic equation approximating the log-likelihood function, it too has a global maximum. Moreover, this maximum can serve as an approximate maximum of $f(\gamma)$. To maximize (53), take the partial derivatives of (52) with respect to γ ,

$$\nabla g(\gamma) = \nabla f(\gamma^{(0)}) + \mathbf{H}(\gamma^{(0)}) \gamma - \mathbf{H}(\gamma^{(0)}) \gamma^{(0)}. \quad (54)$$

Next, let $\gamma^{(1)}$ denote the maximum of (53) and substitute it for γ in (54). It follows that (53) reduces to

$$\mathbf{0} = \nabla f(\gamma^{(0)}) + \mathbf{H}(\gamma^{(0)}) \gamma^{(1)} - \mathbf{H}(\gamma^{(0)}) \gamma^{(0)}. \quad (55)$$

Finally, rearrange (55) to put all terms containing $\gamma^{(1)}$ on the left-hand-side of the equals sign,

$$- \mathbf{H}(\gamma^{(0)}) \gamma^{(1)} = \nabla f(\gamma^{(0)}) - \mathbf{H}(\gamma^{(0)}) \gamma^{(0)}, \quad (56)$$

which can be rewritten as

$$\gamma^{(1)} = \gamma^{(0)} - \mathbf{H}^{-1}(\gamma^{(0)}) \nabla f(\gamma^{(0)}), \quad (57)$$

under the assumption that $\mathbf{H}$ is nonsingular at $\gamma^{(0)}$. Because $\gamma^{(1)}$ maximizes $g(\gamma)$, it approximately maximizes $f(\gamma)$. This approximation can be further refined by again applying the second-order Taylor Series expansion of $f(\gamma)$ in the neighborhood of $\gamma^{(1)}$, replacing $\gamma^{(0)}$ with $\gamma^{(1)}$ in equations (52)–(57). This procedure is repeated until $\|\gamma^{(i+1)} - \gamma^{(i)}\| < \epsilon$, where ϵ denotes some small tolerance value. Once

the tolerance limit has been reached, a local maximum has essentially been found. Figure 10 illustrates the above procedure for a scalar function.

Figure 10 displays the function $f(x) = -20x^2 - x^4$ and three iterations of the Newton-Raphson algorithm. At iteration 1, the second-order Taylor Series approximates $f(x)$ in the neighborhood of $x = -2$, with an approximate maximum of -0.73 . At iteration 2, the second-order Taylor Series then approximates $f(x)$ in the neighborhood of -0.73 , with an approximate maximum of -0.07 . Finally, at iteration 3, the second-order Taylor Series approximates $f(x)$ in the neighborhood of -0.07 , with an approximate maximum of 0 . Further iterations of the Newton-Raphson algorithm continue to yield 0 , thus implying that it has converged. With the algorithm described, we are now ready to apply it to the log-likelihood function for logistic regression.

As just shown, to use the Newton-Raphson algorithm we require the first (gradient) and second (Hessian) partial derivatives of the log-likelihood function with respect to the regression coefficients. To make the derivatives more manageable, we first rewrite the model as

$$P(y_i = 1|\mathbf{x}_i) = \frac{\exp(\mathbf{b}'\mathbf{x}_i)}{1 + \exp(\mathbf{b}'\mathbf{x}_i)}, \quad (58)$$

where $\mathbf{b}$ now contains the intercept, b_0 , and $\mathbf{x}_i$ has a 1 for the intercept term. Substituting the right-hand-side of (58) into (50) yields the following expression for the log-likelihood function,

$$\ln L(\mathbf{b}) = \sum_{i=1}^n y_i [\mathbf{b}'\mathbf{x}_i] - \ln [1 + \exp(\mathbf{b}'\mathbf{x}_i)]. \quad (59)$$

Taking the first and second partial derivatives of (59) with respect to $\mathbf{b}$ yields the $(p + 1) \times 1$ gradient vector,

Figure 10: Newton-Raphson Algorithm

Three iterations of the Newton-Raphson algorithm for computing the maximum of $f(x) = -20x^2 - x^4$.

$$\mathbf{g} = \frac{\partial \ln L(\mathbf{b})}{\partial \mathbf{b}} = \sum_{i=1}^n \left[y_i - \frac{\exp(\mathbf{b}' \mathbf{x}_i)}{1 + \exp(\mathbf{b}' \mathbf{x}_i)} \right] \mathbf{x}_i, \quad (60)$$

and the $(p+1) \times (p+1)$ Hessian matrix,

$$\mathbf{H} = \frac{\partial^2 \ln L(\mathbf{b})}{\partial \mathbf{b}^2} = - \sum_{i=1}^n \left(\frac{\exp(\mathbf{b}' \mathbf{x}_i)}{[1 + \exp(\mathbf{b}' \mathbf{x}_i)]^2} \right) \mathbf{x}_i \mathbf{x}_i'. \quad (61)$$

Then, as shown in (57), an approximation to the maximum at iteration $i+1$ is

$$\mathbf{b}^{(i+1)} = \mathbf{b}^{(i)} - \mathbf{H}^{-1}(\mathbf{b}^{(i)}) \mathbf{g}(\mathbf{b}^{(i)}), \quad (62)$$

where $\mathbf{H}(\mathbf{b}^{(i)})$ and $\mathbf{g}(\mathbf{b}^{(i)})$ denote the Hessian and the gradient of (51) evaluated at $\mathbf{b}^{(i)}$. The Newton-Raphson algorithm is carried out until $\|\mathbf{b}^{(i+1)} - \mathbf{b}^{(i)}\| < \epsilon$ for some small value of ϵ . Once a set of parameter estimates has been obtained, we can use odds ratios (as described earlier) to interpret the coefficients, and the Wald or Likelihood Ratio Tests (Agresti, 2007, p. 107) to conduct significance tests. Conducting a Wald test for the regression coefficients requires the covariance matrix of $\hat{\mathbf{b}}$. We can estimate the covariance matrix as

$$\widehat{\text{cov}}(\hat{\mathbf{b}}) = -n^{-1} \mathbf{H}^{-1}(\hat{\mathbf{b}}_{ML}), \quad (63)$$

where $\text{cov}(\cdot)$ denotes covariance and $\mathbf{H}(\hat{\mathbf{b}}_{ML})$ denotes the Hessian of (51) evaluated at the maximum likelihood values of the regression coefficients. Then, the Wald Test for the j th coefficient against the Nil hypothesis (Cohen, 1994) is

$$z = \frac{\hat{b}_j}{\text{se}(\hat{b}_j)},$$

where

$$\text{se}(\hat{b}_j) = \text{cov}(\hat{\mathbf{b}})_{[j,j]}^{1/2},$$

and z is asymptotically distributed standard normal. With the logistic regression model described, we now discuss methods of computing fungible weights.

3.2 Extending Fungible Weights to Logistic Regression: Ellipsoid Method

In this section we extend the results of Section 2.2 to logistic regression. Specifically, we show how to use the Ellipsoid Method to generate fungible weights for logistic regression models. To do so, recall (see (19)) that the correlation between two linear composites can be computed as

$$r_{\hat{y}_a \hat{y}_b} = \frac{\mathbf{a}' \Sigma_{xx} \mathbf{b}}{(\mathbf{a}' \Sigma_{xx} \mathbf{a})^{1/2} (\mathbf{b}' \Sigma_{xx} \mathbf{b})^{1/2}}.$$

Unfortunately, in logistic regression we are not working with linear composites but nonlinear composites of the form

$$P(y_i = 1 | \mathbf{x}_i) = \frac{\exp(\mathbf{b}' \mathbf{x}_i)}{1 + \exp(\mathbf{b}' \mathbf{x}_i)}$$

(see 58). However, we can transform the above nonlinear composite into a linear composite by using the logit transformation (Agresti, p. 99, 2007)

$$\text{logit}[P(y_i = 1 | \mathbf{x}_i)] = \ln \left[\frac{P(y_i = 1 | \mathbf{x}_i)}{1 - P(y_i = 1 | \mathbf{x}_i)} \right] = \mathbf{b}' \mathbf{x}_i. \quad (64)$$

Equation (64) shows that the predicted log odds is a linear function of the predictors. Using this transformation, we can implement the ellipsoid method outlined in Section 2.2 to generate fungible weights for a specific value of the correlation between two linear composites, $r_{\hat{y}_a \hat{y}_b}$. Once fungible weights have been computed, we can then transform them back into a nonlinear composite using the inverse logit transformation. Once in this form, we can explore the effect of particular sets of

fungible weights on the odds ratios, estimated probabilities, and likelihood function values.

Before leaving this section, we note a potential shortcoming of the Ellipsoid Method when applied to logistic regression. Namely, the formulae for generating fungible weights relies on coefficients of determination for *linear* regression models. Recall that the correlation between two linear composites constructed by OLS regression weights and fungible weights is

$$r_{\hat{\mathbf{y}}_a \hat{\mathbf{y}}_b} = \left(1 - \frac{R_b^2 - R_a^2}{R_b^2} \right)^{1/2}.$$

Unfortunately, no direct analog of the coefficient of determination exists for logistic regression models (Menard, 2000). Specifically, whereas the linear regression coefficient of determination is interpreted as the “proportional reduction in the error sum of squares” (Menard, 2000, p. 20), the logistic regression version(s) (Menard, 2000) is not interpreted as such because the logistic regression weights are not computed to minimize squared errors. To overcome this potential limitation (in addition to the limitations addressed in Section 2.2), we now describe the Log-Likelihood Method of generating fungible weights.⁴

3.3 Extending Fungible Weights to Logistic Regression: Log-Likelihood Method

In the previous section, we discussed how to compute fungible weights for unstandardized regression coefficients in logistic regression using the Ellipsoid Method. Unfortunately, as was discussed in Sections 2.2 and 3.2, the Ellipsoid Method can only be used when the number of regressors is greater than two. To overcome this

⁴Note, we could consider computing an R_b^2 measure at the latent level as $\frac{\text{var}(\hat{\mathbf{y}}_b)}{\text{var}(\hat{\mathbf{y}}_b) + \text{var}(\mathbf{e})}$, where $\text{var}(\hat{\mathbf{y}}_b)$ denotes the variance of the predicted log-odds, and $\text{var}(\mathbf{e})$ denote the variance of the standard logistic distribution, $\pi^2/3$.

limitation, we now extend the Log-Likelihood Method from Section 2.3 to logistic regression.

To begin, recall that log-likelihood function for logistic regression (see (59)) can be written as

$$\ln L(\mathbf{b}) = \sum_{i=1}^n y_i [\mathbf{b}'\mathbf{x}_i] - \ln [1 + \exp(\mathbf{b}'\mathbf{x}_i)],$$

where $\mathbf{b} = \begin{bmatrix} b_0 & b_1 & \dots & b_p \end{bmatrix}'$ and $\mathbf{x}_i = \begin{bmatrix} 1 & x_1 & \dots & x_p \end{bmatrix}'$. Further, let $\mathbf{b}_{ML}$ denote the unique, maximum likelihood estimate of $\mathbf{b}$, and let $\ln L_{ML}$ denote the log-likelihood function value evaluated at $\mathbf{b}_{ML}$. As described in Section 2.3, any likelihood value less than the maximum, $\ln L_f < \ln L_{ML}$, corresponds an infinite number of fungible coefficient vectors, $\mathbf{b}_f$.

Instead of working with $\ln L_f$ directly, we first re-express $\ln L_F$ as a function of the likelihood-ratio pseudo- R^2 measure (Menard, 2000),

$$R_L^2 = 1 - \frac{\ln L_{ML}}{\ln L_0}, \quad (65)$$

where $\ln L_0$ denotes the log-likelihood value with only the intercept in the model. Note that R_L^2 cannot be interpreted as an OLS coefficient of determination. That is, R_L^2 cannot be interpreted as the “proportional reduction in the error sum of squares” (Menard, 2000, p. 20)⁵, but rather as “the proportional reduction in the -2 log-likelihood statistic” (Menard, 2000, p. 20). Because R_L^2 is associated with the maximum likelihood estimate of the regression coefficients, reducing R_L^2 will result in an infinite number of fungible weights. Let $R_{L_f}^2$ denote a smaller value of the pseudo- R^2 , such that $R_{L_f}^2 = R_L^2 - \epsilon$, where ϵ is some small positive number. Then,

⁵Note that in linear regression, the coefficient of determination can also be interpreted as the “proportion of criterion-variance that is predictable from the regressors”.

we can rewrite (65) in terms of $R_{L_f}^2$,

$$R_{L_f}^2 = 1 - \frac{\ln L_f}{\ln L_0}, \quad (66)$$

where $\ln L_f$ replaces $\ln L_{ML}$ to account for the reduction in model fit. Solving for $\ln L_f$ yields

$$\ln L_f = \ln L_0(1 - R_{L_f}^2). \quad (67)$$

It follows that $\ln L_f$ is the log-likelihood value associated with the suboptimal model fit, $R_{L_f}^2$. With our method of reducing model fit, we now explain how to generate fungible weights for any value of the log-likelihood function smaller than the maximum.

To begin, we first use (67) to rewrite the log-likelihood function in terms of the reduced fit as

$$\ln L_0(1 - R_{L_f}^2) = \ln L(\mathbf{b}_f) = \sum_{i=1}^n y_i [\mathbf{b}'_f \mathbf{x}_i] - \ln [1 + \exp(\mathbf{b}'_f \mathbf{x}_i)], \quad (68)$$

where $\mathbf{b}_f$ denotes a $(p+1) \times 1$ fungible weight vector. Just as with the original log-likelihood function (59), a closed form solution for $\mathbf{b}_f$ in (68) does not exist. More importantly, a unique solution to (68) does not exist. As discussed earlier, there are an infinite number of $\mathbf{b}_f$ that yield a specific value of $\ln L_f$. One way to find a solution for (68) is to rewrite it as

$$0 = \sum_{i=1}^n y_i [\mathbf{b}'_f \mathbf{x}_i] - \ln [1 + \exp(\mathbf{b}'_f \mathbf{x}_i)] - \ln L_0(1 - R_{L_f}^2) \quad (69)$$

and then use an algorithm to find a root. One such algorithm is Newton's method (Ben-Israel & Greville, 2003, p. 40–43; see also Ben-Israel, 1965).

To use Newton's method for computing roots, we first rewrite (69) as the follow-

ing function

$$f(\mathbf{b}_f) = \sum_{i=1}^n y_i [\mathbf{b}'_f \mathbf{x}_i] - \ln [1 + \exp(\mathbf{b}'_f \mathbf{x}_i)] - \ln L_0(1 - R_{L_f}^2). \quad (70)$$

Then, we approximate $f(\mathbf{b}_f)$ in the neighborhood of $\mathbf{b}^{(0)}$ by a first-order Taylor Series expansion, $g(\mathbf{b}_f)$, as

$$g(\mathbf{b}_f) = f(\mathbf{b}^{(0)}) + \nabla f(\mathbf{b}^{(0)})'(\mathbf{b}_f - \mathbf{b}^{(0)}). \quad (71)$$

Let $g(\mathbf{b}_f) = 0$ at the point $\mathbf{b}^{(1)}$. Then, substituting $\mathbf{b}^{(1)}$ for $\mathbf{b}_f$, we can rewrite (71) as

$$0 = f(\mathbf{b}^{(0)}) + \nabla f(\mathbf{b}^{(0)})'(\mathbf{b}^{(1)} - \mathbf{b}^{(0)}). \quad (72)$$

Our goal then is to solve (72) for $\mathbf{b}^{(1)}$. If $\nabla f(\mathbf{b}^{(0)})$ in (72) was a square, nonsingular matrix, we could easily solve for $\mathbf{b}^{(1)}$ as

$$\mathbf{b}^{(1)} = \mathbf{b}^{(0)} - \nabla f(\mathbf{b}^{(0)})^{-1} f(\mathbf{b}^{(0)}), \quad (73)$$

where $\nabla f(\mathbf{b}^{(0)})^{-1}$ denotes the inverse of ∇f evaluated at $\mathbf{b}^{(0)}$. However, because ∇f is a $p \times 1$ gradient vector of the function in (70), we cannot take its inverse. Fortunately, as shown by Ben-Israel & Greville (2003; see also, Ben-Israel, 1965), we can use the Moore-Penrose generalized inverse in place of $\nabla f(\mathbf{b}^{(0)})^{-1}$ in (73). Using this fact, rewrite (73) as

$$\mathbf{b}^{(1)} = \mathbf{b}^{(0)} - \nabla f(\mathbf{b}^{(0)})^- f(\mathbf{b}^{(0)}), \quad (74)$$

where $\nabla f(\mathbf{b}^{(0)})^-$ denotes the Moore-Penrose generalized inverse evaluated at $\mathbf{b}^{(0)}$. It follows that (74) is the root of $g(\mathbf{b}_f)$ and is thus an approximate root of $f(\mathbf{b}_f)$. This approximation can be further refined by again applying the first-order Taylor Series

expansion of $f(\mathbf{b}_f)$ in the neighborhood of $\mathbf{b}^{(1)}$, replacing $\mathbf{b}^{(0)}$ with $\mathbf{b}^{(1)}$ in equations (71)–(74). This procedure is repeated until $\|\mathbf{b}^{(i+1)} - \mathbf{b}^{(i)}\| < \epsilon$, where ϵ denotes some small tolerance value. Once the tolerance limit is reached, a root has been essentially found. Importantly, this root is the desired vector of fungible regression coefficients that solves (68). However, the above procedure yields only one fungible regression coefficient vector out of the infinite number possible. To generate more fungible regression coefficient vectors, the procedure is repeated using a different starting value for $\mathbf{b}^{(0)}$. An algorithm to generate N fungible weight vectors is as follows:

1. Choose a random $(p + 1) \times 1$ starting point, $\mathbf{b}^{(0)}$, near $\mathbf{b}_{ML}$.
2. Apply Newton's method until convergence is reached to obtain a fungible weight vector, $\mathbf{b}_f$, that yields a user-defined value of R_{Lf}^2 .
3. Store $\mathbf{b}_f$.
4. Repeat Steps 1–3 N times to generate N fungible weight vectors.

An R program that implements the above algorithm is included in Appendix C.

4 Re-analysis of Published Models

To motivate the use of fungible weights in logistic regression, we first reanalyze two well known datasets (Adwere-Boamah, 2009; Steering Committee of the Physicians Health Study Research Group, 1988) and examine whether the results discussed by the authors hold up under a fungible weights analysis. Specifically, we will use the procedures outlined in Sections 3.2 and 3.3 to reanalyze the Aspirin Component of the Physicians’ Health Study’s (Steering Committee of the Physicians Health Study Research Group, 1988) results on the survival rate of heart attack patients, and the 2009 CDC Youth Risk Behavior Surveillance Survey results on cigarette use among high school students (Adwere-Boamah, 2009). Of interest is whether the authors’ interpretations of the logistic regression coefficients is consistent with an in depth parameter sensitivity analysis by way of fungible weights.

4.1 Aspirin Study

The Aspirin Component of the Physicians’ Health Study is a well-known (Rosenthal, 1990) randomized double-blind experiment that used $n = 22,071$ participants and tested whether taking aspirin every other day reduces the death rate from cardiovascular disease (Steering Committee of the Physicians Health Study Research Group, 1988). Interestingly, this study was cut short after the external Data Monitoring Board of the Physicians’ Health Study deemed the study’s continuation unethical as a “statistically extreme beneficial effect on nonfatal and fatal myocardial infarction had been found” (Steering Committee of the Physicians Health Study Research Group, 1988, p. 262). In other words, the study’s preliminary results suggested that taking aspirin every other day greatly improved the survival rate from cardiovascular disease. Specifically, the odds of survival for those who took aspirin were 1.83 times as high as the group who did not take aspirin. Using this informa-

tion and the original dataset, we can rerun the analysis using logistic regression and determine the parameter sensitivity of the odds ratio.

Table 1: Data from the Aspirin Component of the Physician’s Health Study

	Heart Attack	No Heart Attack	Total
Aspirin	104	10,933	11,037
No Aspirin	189	10,845	11,034
Total	293	21,778	22,071

Table 2: Logistic Regression Model for the Aspirin Study

Predictor	Coefficient	SE	z	$\Pr(> z)$	$\text{CI}_l^\dagger$	$\text{CI}_u^\dagger$
Intercept	-4.655	0.099	-47.250	< 0.001	-4.848	-4.462
Aspirin	0.605	0.123	4.929	< 0.001	0.365	0.846

$$R_L^2 = 0.0081$$

$\dagger$ CI_l and CI_u denote the lower and upper bounds of the 95% Wald-based confidence intervals.

Table 1 displays the data from the Aspirin Study. In this example, both the regressor and criterion are dichotomous variables, allowing for simple reproduction of the data. Table 2 displays the logistic regression model summary of these data. Notice that $\exp(0.605) = 1.83$, the odds ratio from the original study. Assuming the distributional assumptions are met (at the least approximately), then the the 95% CI for the Aspirin coefficient (as well as the p value) informs us that the coefficient is significant at $\alpha = .05$. Using this information⁶, the authors of the Aspirin Study declared that aspirin had a “beneficial effect on nonfatal and fatal myocardial infarction” (Steering Committee of the Physicians Health Study Re-

⁶Technically, the authors did not use odds ratios or logistic regression, but rather relative risks. For more details, see Steering Committee of the Physicians Health Study Research Group (1988).

search Group, 1988, p. 262). Using the data and the information from the logistic regression model, we can use the Log-Likelihood Method to generate fungible weights for a small decrement in model fit. Table 2 shows that $R_L^2 = 0.0081$. To compute fungible weights, we decreased R_L^2 by .0001, so that $R_{Lf}^2 = 0.0080$. Note that because there is a single predictor, the Ellipsoid Method for generating fungible weights cannot be used. To assess the sensitivity of the logistic regression model coefficients, and therefore the odds ratio, we generated 20,000 fungible weight vectors. Figure 11 summarizes these results.

Figure 11: Fungible Weights for the Aspirin Study

The fungible weight contour for the Aspirin Study logistic regression model.

Figure 11 displays the fungible weight contour for the intercept and slope co-

efficients from the logistic regression model in Table 2. Notice that the coefficient for Aspirin use, a_1 , ranges from 0.54 to 0.67. Recall that the odds ratio from the original logistic regression model was 1.83 and the 95% confidence interval was $[0.37, 0.85]$. Interestingly, the range of the fungible weight for the logistic regression coefficient is actually inside the limits of this 95% confidence interval. This tight range suggests that the interpretation offered by the Steering Committee holds up under the sensitivity analysis and is very robust to a small change in model fit.

4.2 Smoking Study

The Smoking Study (Adwere-Boamah, 2009) analyzed a subset ($n = 11,683$) of the dataset from the 2009 CDC Youth Risk Behavior Surveillance Survey to assess the relationship between youth smoking behavior and five variables: (1) race, (2) cocaine use, (3) first time of cigarette use, (4) feeling sad/hopeless, and (5) physical inactivity (Adwere-Boamah, 2009). To determine the relationship between cigarette use and the five covariates, Adwere-Boamah (2009) fit a logistic regression model. After fitting the logistic regression model, Adwere-Boamah concluded that “[t]he strongest predictors of youth smoking behavior [were] race, frequency of cocaine use and physically inactive behavior” (p. 5). In other words, these three predictors produced the largest odds ratios among the set of five. Moreover, Adwere-Boamah (2009) claimed that race was the “strongest predictor” (p. 4) of cigarette use, and reported that the odds of “smoking are increased by a factor of 5.0 if the student is White compared to an African American” (p. 5). To assess whether these strong conclusions hold up under a sensitivity analysis, we computed fungible weights using both the Ellipsoid and Log-Likelihood Methods. To begin, we review the results of the logistic regression model examined by Adwere-Boamah (2009).

Table 3: Logistic Regression Model Reported by Adwere-Boamah (2009)

Predictor	Coefficient	SE	Odds Ratio
Intercept	-7.23	0.22	0.00
Race White	1.62	0.18	5.04
Hispanic	-0.46	0.27	0.63
Cocaine Use	1.26	0.07	3.52
Physical Inactivity	0.54	0.08	1.71
Sad/Hopeless	0.53	0.08	1.70
First Time Smoking	0.49	0.02	1.64

Note: These values were taken from Table 1 of Adwere-Boamah (2009).

Table 3 displays the logistic regression model reported by Adwere-Boamah (2009). Notice that race (White) has the highest odds ratio followed by cocaine use and physical inactivity. However, because the selection rules were not clearly articulated in Adwere-Boamah’s (2009) article, we were unable to exactly reproduce his results. Specifically, we had a sample size of $n = 11,424$ rather than $n = 11,683$, and our coefficient values were slightly different than the ones reported in Table 3. Table 4 displays our findings.

Table 4: Reproduced Logistic Regression Model

Predictor	Coefficient	SE	Odds Ratio
Intercept	-6.97	0.22	0.00
Race White	1.68	0.18	5.35
Hispanic	-0.38	0.27	0.68
Cocaine Use	1.17	0.07	3.21
Physical Inactivity	0.44	0.08	1.56
Sad/Hopeless	0.50	0.08	1.65
First Time Smoking	0.46	0.02	1.59

Note: This model is slightly different from the one reported in Adwere-Boamah (2009). $R_L^2 = 0.2618$

Table 4 displays our attempt at fitting the logistic regression model reported by Adwere-Boamah (2009). Though our logistic regression model does not exactly match Adwere-Boamah’s (2009), the coefficient values are close. Moreover, the conclusions drawn by Adwere-Boamah (2009) match our reproduced model as well. Because the coefficients are similar to those reported by Adwere-Boamah (2009), we felt confident in conducting a sensitivity analysis to examine his conclusions. To that end, we use the model reported in Table 4 to construct fungible weights using both the Ellipsoid and Log-Likelihood Methods. We first describe the Ellipsoid Method results.

To compute fungible weights using the Ellipsoid Method (see Section 3.2), we set the correlation between the log-odds linear composite formed by the logistic regression coefficients and the log-odds linear composite formed by the fungible weights to $r_{\hat{y}_b \hat{y}_a} = 0.99$. To assess the sensitivity of the coefficients reported in Table 4, we generated 150,000 fungible weight vectors. Figure 12 displays these results.

Figure 12: Fungible Weight Contours for Smoking Study - Ellipsoid Method

Panels A–O show the fungible weight contours computed from the Ellipsoid Method for all pairs of coefficients, not including the intercept.

Figure 12 displays fifteen panels, each of which shows the fungible weight contour for each pair of coefficients. Coefficients a_1 , a_2 , a_3 , a_4 , a_5 , and a_6 represent the fungible weights for Race Hispanic, Race White, First Time Smoking, Sad/Hopeless, Cocaine Use, and Physical Inactivity, respectively. A quick glance through the panels reveals that some of the fungible weights for variables Sad/Hopeless, and Physical Inactivity can be negative, whereas the some of the fungible weights for variable Race Hispanic can be positive. This finding implies that the odds ratios for these variables are sometimes in the opposite direction of those reported in Table 4. In other words, for some fungible weights, the odds of smoking *decrease* by some factor for every unit increase in Sad/Hopeless or Physical Inactivity, whereas the odds *increase* by some factor if the person is Hispanic compared to African-American. Importantly, these findings run counter to what was reported by Adwere-Boamah (2009). Moreover, the “strongest predictor”—Race White—as reported by Adwere-Boamah, is not always the strongest predictor among the fungible weights. Table 5 shows two such examples.

Table 5: Two Different Sets of Fungible Weights for The Smoking Study

Predictor	Original Coefficients	Fungible Set 1	Fungible Set 2
Race White	1.68	1.14	0.95
Hispanic	-0.38	-0.72	-0.82
Cocaine Use	1.17	1.10	1.86
Physical Inactivity	0.44	0.21	0.48
Sad/Hopeless	0.50	1.25	0.33
First Time Smoking	0.46	0.43	0.44

Table 5 displays two fungible weight vectors along with the original coefficients reported in Table 4. Notice in Fungible Set 1 that Race White, Cocaine Use, and

Sad/Hopeless are approximately equal, however, in Fungible Set 2, we see that Cocaine Use is nearly twice the size of Race White, making it the “strongest predictor”. These findings, in conjunction with the earlier findings, call Adwere-Boamah’s interpretation of his model into question. With a slight decrement in model fit, the logistic regression fungible weights computed by the Ellipsoid Method vary substantially. A similar pattern occurs when we reanalyze the data using the Log-Likelihood Method.

Using the raw data and the information from the logistic regression model in Table 4, we can reduce the model fit and use the Log-Likelihood Method to generate fungible weights. Table 4 shows the model fit as $R_L^2 = 0.26$. To compute fungible weights, we set $R_{Lf}^2 = 0.25$. Using this value, we then computed 150,000 fungible weight vectors to assess the sensitivity of the the logistic regression model reported by Adwere-Boamah (2009). Figure 13 displays these results.

Figure 13: Fungible Weight Contours for Smoking Study - Log-Likelihood Method

Panels A–O show the fungible weight contours computed from the Log-Likelihood Method for all pairs of coefficients, not including the intercept.

Figure 13 displays fifteen panels, each of which shows the fungible weight contours for each pair of coefficients computed using the Log-Likelihood Method. Notice that the pattern of results is similar to those from Figure 12. Namely, the fungible weights for variables Sad/Hopeless, and Physical Inactivity can be negative, whereas the fungible weight for variable Race Hispanic can be positive. Again, this finding disagrees with what was reported by Adwere-Boamah (2009). Moreover, just as it was in the Ellipsoid Method, Race White is not always the “strongest predictor”. It should be noted that though the two methods of generating fungible weights yields similar results, they do not yield equivalent results. For example, the range of fungible weights are different between the two methods in an inconsistent manner. Specifically, coefficient a_1 (variable Race Hispanic) has a wider range under the Log-Likelihood Method, whereas coefficient a_5 has a wider range under the Ellipsoid Method. However, the differences are to be expected because the methods rely on different fit functions with no direct translation between them. That is, the Ellipsoid Method relies on the correlation between two log-odds linear composites, whereas the Log-Likelihood Method relies on the Log-Likelihood function value (parameterized as the likelihood ratio pseudo- R^2). Because the two methods rely on different fit functions, it is not surprising that the range of the fungible weights are different between the two methods.

4.3 Reanalysis of Published Models: Summary and Extensions

The previous two sections subjected published logistic regression models to sensitivity analyses using the methods described in this paper. Specifically, we reanalyzed the data from the Aspirin Study (Steering Committee of the Physicians Health Study Research Group, 1988) using the Log-Likelihood Method, and we reanalyzed the data from the Smoking Study (Adwere-Boamah, 2009) using both the

Ellipsoid and Log-Likelihood Methods. In the Aspirin Study reanalysis, we found that the interpretations offered by the author held up under our sensitivity analysis. In the Smoking Study reanalysis, we found that the interpretations offered by the author did not hold up under a sensitivity analysis. Both reanalyses demonstrated the usefulness of using fungible weights in examining parameter sensitivity. As noted by Pek (2012), the fungible weight “contours communicate how stable parameter estimates remain in describing the data under a practically indifferent model fit” (p. 31). In the Aspirin Study, the parameters were very robust to minor model fit perturbations, whereas the coefficients from the Smoking Study were not robust to minor model fit perturbations. Although both datasets contained a large number of cases ($n = 22,071$ for the Aspirin Study, and $n = 11,424$ for the Smoking Study), and all but one of the coefficients were significantly different from zero (Race Hispanic $p = .08$), the two fungible weights analyses led to different conclusions. This finding draws an important distinction between sampling variability and parameter sensitivity. Specifically, as noted by Green (1977, see also Waller, 2008),

Sampling variability depends mainly on the number of cases. Parameter sensitivity is, by contrast, independent of n . With the entire population at hand, sampling variability disappears, but parameter sensitivity is still a problem. The model will still not fit perfectly; further, two quite different sets of parameters may still yield nearly the same fit of the model to the data. In a sense, the issue of sensitivity, or tolerable parameter variation, is more a question about the nature of the model than about the data being described (p. 264).

Pilot simulations examining the effect of sample size on our methods support Green’s (1977) claim. In the next section, we describe a simulation study used to evaluate what aspects of the data affect the computation of fungible weights.

5 Simulation Design

5.1 Factors to be Varied

In Sections 3.2 and 3.3, we outlined two approaches of computing fungible weights in logistic regression. Because the approaches operate on different fit functions, properties of the data might differentially affect both methods. To investigate this possibility, we construct populations that differ along the data-parameter configurations that were discussed in Section 2.3. Specifically, we vary: (1) the coefficient of determination between the criterion and the predictors at the latent level, R_{β}^2 , (2) correlational structure of the predictors, $\mathbf{R}_{xx}$, (3) orientation of β with respect to the eigenvectors of $\mathbf{R}_{xx}$, and (4) number of predictors, p . Note that although the population is described in terms of standardized data (i.e., data that have been scaled to have zero means and unit variances), samples from the population will not be standardized. Generating the data to be standardized allows us to make use of published algorithms (e.g., **GenCorr**: Jones, 2010; see also Marsaglia & Olkin, 1984; **rarc**: Waller & Jones, 2011) and relates this study’s findings to those from previous research (Jones & Waller, in press; Waller, 2011; Waller & Jones, 2011).

For our simulation study we chose two factor levels for R_{β}^2 , four factor levels for $\mathbf{R}_{xx}$, four factor levels for the orientation of β , and two factor levels for the number of predictors. Specifically, we generated data to be consistent with $R_{\beta}^2 = \{.20, .80\}$ and $p = \{3, 5\}$ for the different factor levels of $\mathbf{R}_{xx}$ and orientation of β . For $\mathbf{R}_{xx}$ we chose three classes of correlational structure: (1) a one-factor model correlation matrix, (2) an orthogonal two-factor model correlation matrix, and (3) randomly

structured correlation matrices based on eigenvalues. For $p = 3$ we use

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.0 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{bmatrix}$$

for the one-factor model correlation matrix and

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.0 & 0.5 & 0.0 \\ 0.5 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

for the orthogonal two-factor model correlation matrix. For $p = 5$ we use

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1.0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1.0 \end{bmatrix}$$

for the one-factor model correlation matrix and

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.0 & 0.5 & 0.5 & 0.0 & 0.0 \\ 0.5 & 1.0 & 0.5 & 0.0 & 0.0 \\ 0.5 & 0.5 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.5 & 1.0 \end{bmatrix}$$

for the orthogonal two-factor model correlation matrix. For the randomly structured correlation matrices we generated correlation matrices to have condition numbers of $\kappa = 3$, or 9, using an algorithm by Marsaglia and Olkin (1984; Jones, 2010).

Note that κ represents the matrix condition number and is a popular measure of multicollinearity (Cohen, Cohen, West, & Aiken, 2003, p. 424).⁷ Finally, for the orientation of β , we examine four vectors of regression coefficients that lie in the plane defined by the first and last eigenvectors of the correlation matrices described earlier. Specifically, we choose β_1 collinear with the first eigenvector, β_2 30 degrees from first eigenvector (offset in the direction of the last eigenvector), β_3 60 degrees from the first eigenvector, and β_4 collinear with the last eigenvector. We chose these vectors to relate our findings to those of previous research (Jones & Waller, in press; Waller & Jones, 2010, 2011; Waller, 2011).

5.2 Data Generation for Fungible Weight Computation

Combinations of the aforementioned factor levels yields 64 datasets. Each dataset is generated to have $n = 500$ cases using the R (R Core Team, 2012) programming language as follows:

1. Generate a p dimensional correlation matrix that satisfies one of the aforementioned correlational structures.
2. Generate multivariate normal data that yields the correlation matrix in Step 1 using the R function `mvrnorm` (Venables & Ripley, 2002) with the argument `empirical = TRUE`. That is, the data are not sampled from a distribution with a specified correlation matrix, but rather, the data are generated to exactly yield the specified correlation matrix.
3. Generate β to have a desired coefficient of determination, R^2_β , and orientation with the eigenvectors of R_{xx} .

⁷We note that there is not a clear consensus on what value of κ indicates problematic multicollinearity. Cohen, Cohen, West, and Aiken (2003, p. 424) suggest that $\kappa > 30$ indicates severe multicollinearity. However, in a more recent publication, Lazardis (2007, p. 134), suggested that multicollinearity becomes more severe as the condition number for a correlation matrix exceeds 5.4.

4. Generate the latent dependent variable $\mathbf{y}^*$ as

$$\mathbf{y}^* = \mathbf{X}\boldsymbol{\beta}^* + \mathbf{e},$$

where $\mathbf{e}$ is constructed to be orthogonal to the predictors and distributed standard logistic. Because the variance of $\mathbf{e}$ is $\pi^2/3$ rather than one, we have to rescale $\boldsymbol{\beta}$ as $\boldsymbol{\beta}^* = \sqrt{\frac{\pi^2}{3(1-R_{\boldsymbol{\beta}}^2)}}\boldsymbol{\beta}$ to maintain the desired $R_{\boldsymbol{\beta}}^2$.

5. Construct the binary variable, $\mathbf{y}$, as

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{otherwise.} \end{cases}$$

For each dataset, we then generate 100,000 fungible weight vectors using both the Ellipsoid and Log-Likelihood Methods. For the Ellipsoid Method we decrease model fit by fixing $R_{\boldsymbol{\beta}}^2 - R_{\boldsymbol{\alpha}}^2 = .01$, where $R_{\boldsymbol{\beta}}^2 = \text{cor}(\mathbf{X}\boldsymbol{\beta}, \mathbf{y})^2$ with $\mathbf{y}$ being a dichotomous variable. For the Log-Likelihood Method we decrease model fit by fixing $R_{L_f}^2 = R_L^2 - 0.01$. The results of these simulations are summarized in the following section.

6 Results

6.1 Simulation Study Results Comparing the Ellipsoid and Log-Likelihood Methods

In this section we review the results of our simulation study. First, we examine how the different data-parameter combinations affect the Log-Likelihood Method for computing fungible weights. Next, we examine how the Ellipsoid Method for computing fungible weights compares to the Log-Likelihood Method. Of interest is whether the design factors affect the two methods in a similar way.

6.1.1 Coefficient of Determination

To begin, we examine how R_{β}^2 affects the fungible weight contours computed from the Log-Likelihood Method. Specifically, we examine the fungible weight contours for logistic regression models that were fit to data generated with the one-factor predictor correlation matrix

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.0 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{bmatrix},$$

latent $R_{\beta}^2 = \{.20, .80\}$, and β_i ($i = 1, 2, 3, 4$) oriented at 0, 30, 60, and 90 degrees from the first eigenvector of $\mathbf{R}_{xx}$. Recall that we constrained β_i to lie in the plane spanned by the first ($\mathbf{v}_1$) and last ($\mathbf{v}_3$) eigenvectors of the above $\mathbf{R}_{xx}$ such that β_1 was collinear with $\mathbf{v}_1$, β_2 was 30 degrees from $\mathbf{v}_1$, β_3 was 60 degrees from $\mathbf{v}_1$, and β_4 was collinear with $\mathbf{v}_3$. After generating the data to meet these specifications resulting in eight datasets, we fit logistic regression models and computed fungible weights. Table 6 displays the logistic regression coefficients for these eight fitted models.

Table 6: Coefficients for Models Generated from $R^2_{\beta} = \{.20, .80\}$ and $\mathbf{R}_{xx} = \text{One-Factor}$

		Location of β_i	$\beta_{i[1]}^a$	$\beta_{i[2]}$	$\beta_{i[3]}$
$R^2_{\beta} = .20$	Model 1	$\beta_i \mathbf{v}_1^b$	-0.47	-0.22	-0.41
	Model 2	$\angle(\beta_i, \mathbf{v}_1) = 30^\circ$	-0.40	-0.69	-0.10
	Model 3	$\angle(\beta_i, \mathbf{v}_1) = 60^\circ$	-0.22	-0.84	0.26
	Model 4	$\beta_i \mathbf{v}_3$	0.06	-1.11	0.97
$R^2_{\beta} = .80$	Model 5	$\beta_i \mathbf{v}_1$	-1.36	-1.57	-1.61
	Model 6	$\angle(\beta_i, \mathbf{v}_1) = 30^\circ$	-1.29	-2.79	-0.58
	Model 7	$\angle(\beta_i, \mathbf{v}_1) = 60^\circ$	-0.96	-3.75	1.25
	Model 8	$\beta_i \mathbf{v}_3$	0.07	-3.31	3.21

^a $\beta_{i[j]}$ denotes the j th element of β_i .

^b The notation $\mathbf{a} || \mathbf{b}$ denotes that vectors $\mathbf{a}$ and $\mathbf{b}$ are parallel.

Notice that Models 1 and 5 were both generated from $\beta || \mathbf{v}_1$ and the one-factor correlation matrix. These two models only differ in their latent coefficient of determinations, R^2_{β} . Models 2 and 6, 3 and 7, and 4 and 8 follow this pattern as well. Further, notice that for every model comparison, the coefficients for $R^2_{\beta} = .80$ condition are all larger than the corresponding coefficients in the $R^2_{\beta} = .20$ condition. This difference in coefficient size (and R^2_{β}) can be seen in Figure 14.

Figure 14: Fungible Weight Contours for Models 1 and 5

The red fungible weight contours correspond to the $R_\beta^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\beta_i ||\mathbf{v}_1$ condition. The blue fungible weight contours correspond to the $R_\beta^2 = .80$, one-factor $\mathbf{R}_{xx}$, and $\beta_i ||\mathbf{v}_1$ condition.

Figure 14 displays the fungible weight contours for Models 1 and 5. The red fungible weight contours correspond to the $R^2_{\beta} = .20$ condition, whereas the blue fungible weight contours correspond to the $R^2_{\beta} = .80$ condition. First, notice that the red fungible weight contour is smaller than the blue fungible weight contour. This finding is in accord with what we saw in Figure 2 (see Section 2.3.1) with the Ellipsoid Method. Namely, that larger coefficients of determinations lead to larger fungible weight contours, all else being equal. In other words, larger coefficients have more room to move around (vary) than their smaller counterparts (however recall from Section 2.3.1 that the opposite is true after we take into account the original coefficient values). Further, notice that the red and blue fungible contours in each panel are not oriented in a similar direction. Figure 14 shows that the red fungible contour has not only been shrunk (because of the smaller R^2_{β}) but also rotated 90 degrees. Importantly, these differences in orientation were not seen (nor could they have been due to the geometry) in the Ellipsoid Method fungible weight contours of Figure 2. This finding could be an artifact from how we generated the data and estimated the models. Recall from Section 5.2, that the data were first generated at the latent level and then the criterion was dichotomized. After the dichotomization, the logistic regression models were estimated. Because of the dichotomization and model estimation, the exact values of the generating parameters were not fixed at the model level, only at the latent level. This procedure allowed for the fitted models to differ in more ways than they did at the latent level, thus potentially introducing unforeseen effects. To test this possibility, we generated several more datasets that fit the conditions represented in Figure 14. Interestingly, in each of these replications, the fungible weight contours aligned in a similar manner as they did in Figure 14. That is, for Models 1 and 5, the fungible weight contours were never oriented in the same direction. Despite spending considerable time running additional simulations and analyzing their results, we currently do not under-

stand what is causing the alignments to differ between these two conditions. That being said, we now review the results from other model pairs where their fungible weight contours that are in a consistent alignment. These findings are displayed in Figures 15–17.

Figures 15–17 display the fungible weight contours for Models 2 and 6, Models 3 and 7, and Models 4 and 8, respectively. The red fungible weight contours correspond to the $R_{\beta}^2 = .20$ conditions, whereas the blue fungible weight contours correspond to the $R_{\beta}^2 = .80$ conditions. In these figures, we see a similar pattern to those results summarized in Figure 14. Namely, the $R_{\beta}^2 = .20$ fungible weight contours are all smaller in absolute size than their $R_{\beta}^2 = .80$ counterparts. Again, this finding is not surprising given the results of Figure 14 and what we discussed in Section 2.3.1. A second aspect of these figures to notice is that the $R_{\beta}^2 = .20$ fungible weight contours are all in a similar alignment to their respective $R_{\beta}^2 = .80$ counterparts. This finding is different from what we saw in Figure 14, and is in agreement with what we saw from the Ellipsoid Method in Figure 2 (see Section 2.3.1). It should be noted that the alignments are not perfect (see Figure 15 Panel A or Figure 16 Panel A for two examples) however, but are closer than they were in Figure 14.

Figure 15: Fungible Weight Contours for Models 2 and 6

The red fungible weight contours correspond to the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, v_1) = 30^\circ$ condition. The blue fungible weight contours correspond to the $R_{\beta}^2 = .80$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, v_1) = 30^\circ$ condition.

Figure 16: Fungible Weight Contours for Models 3 and 7

The red fungible weight contours correspond to the $R^2_{\beta} = .20$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, v_1) = 60^\circ$ condition. The blue fungible weight contours correspond to the $R^2_{\beta} = .80$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, v_1) = 60^\circ$ condition.

Figure 17: Fungible Weight Contours for Models 4 and 8

The red fungible weight contours correspond to the $R^2_{\beta} = .20$, one-factor $\mathbf{R}_{xx}$, and $\beta_i || \mathbf{v}_3$ condition. The blue fungible weight contours correspond to the $R^2_{\beta} = .80$, one-factor $\mathbf{R}_{xx}$, and $\beta_i || \mathbf{v}_3$ condition.

Before we leave this section, we must point out that we have only looked at how R^2_{β} affects the fungible weights in an absolute sense. Recall from Section 2.3.1, that larger R^2_{β} lead to larger fungible weight contours in an *absolute* sense (which was replicated using the Log-Likelihood Method), but lead to smaller fungible weight contours in a *relative* sense. To see whether a similar effect happens in the Log-Likelihood Method, we divided each fungible weight by its corresponding logistic regression coefficient (see Table 6). Figure 18 summarizes these results for each model comparison.

Figure 18: Scaled Fungible Weight Contours for Models 1–8

The red fungible weight contours correspond to the $R_\beta^2 = .20$, one-factor $\mathbf{R}_{xx}$, and all β conditions. The blue fungible weight contours correspond to the $R_\beta^2 = .80$, one-factor $\mathbf{R}_{xx}$, and all β conditions.

Figure 18 displays the scaled fungible weight contours for Models 1–8. Specifically, Panels A–C display Models 1 and 5, Panels D–F display Models 2 and 6, Panels G–I display Models 3 and 7, and Panels J–L display Models 4 and 8. Just like before, the red fungible weight contours correspond to the $R^2_{\beta} = .20$ conditions, whereas the blue fungible weight contours correspond to the $R^2_{\beta} = .80$ conditions. Notice that for most panels, the fungible weight contours for the $R^2_{\beta} = .20$ condition are larger than the corresponding fungible weight contours for the $R^2_{\beta} = .80$ condition. This finding is in accord with what we saw in Section 2.3.1 with Figure 3. Namely, that larger R^2_{β} conditions lead to smaller fungible weight contours relative to the original model coefficients. Interestingly, the conditions represented by Panels J and K do not exhibit this effect. It appears that for these particular coefficients, that larger R^2_{β} does not lead to smaller fungible weight contours relative to the original model coefficients. Instead, we see that the two contours are roughly the same size. This finding is most likely due to a combination of the Log-Likelihood Method operating on a different fit function than the Ellipsoid Method, and the way we generated the data.

In aggregate, the findings from these conditions suggest that R^2_{β} affects the Log-Likelihood Method of generating fungible weight contours in a similar fashion as the Ellipsoid Method. Namely, that larger coefficients of determination yield larger fungible weight contours in an *absolute* sense, but generally yield smaller contours *relative* to their logistic regression coefficients, all else being equal. Moreover, the findings from Figure 14 suggest that unlike in the Ellipsoid Method, in the Log-Likelihood Method, R^2_{β} has the potential to not only affect the size of the fungible weight contours, but also their orientation. These findings are replicated in the other conditions that were outlined in Section 5 and all graphical summaries of unreported (and reported) results are in Appendix A.

6.1.2 Structure of the Predictor Correlation Matrix

Having reviewed how the coefficient of determination affects the Log-Likelihood Method of computing fungible weights for logistic regression coefficients, we now turn our attention to the structure of the predictor correlation matrix.

Recall that we used four types of correlation matrices for each of the $p = \{3, 5\}$ conditions: (1) correlation matrices generated from a one-factor model such that all variables correlate .50, (2) correlation matrices generated from an orthogonal two-factor model such that variables from one factor correlate .50, variables from the second factor correlate .50, and variables from one factor do not correlate with the variables from the second factor (i.e., a block diagonal structure), (3) correlation matrices generated from eigenvalues (Marsaglia & Olkin, 1984; see also Jones, 2010) to yield a condition number $\kappa = 3$, and (4), correlation matrices generated from eigenvalues to yield a condition number $\kappa = 9$. To examine how different correlation matrices affect the computation of fungible weights via the Log-Likelihood Method, we compare fungible weight contours generated from the $\kappa = 3$ correlation matrix

$$\mathbf{R}_{\mathbf{x}\mathbf{x}, \kappa=3} = \begin{bmatrix} 1.00 & -0.18 & 0.34 \\ -0.18 & 1.00 & -0.77 \\ 0.34 & -0.77 & 1.00 \end{bmatrix},$$

and the $\kappa = 9$ correlation matrix

$$\mathbf{R}_{\mathbf{x}\mathbf{x}, \kappa=9} = \begin{bmatrix} 1.00 & -0.17 & 0.72 \\ -0.17 & 1.00 & 0.53 \\ 0.72 & 0.53 & 1.00 \end{bmatrix},$$

with the same latent $R_{\beta}^2 = .20$, and β_i oriented at 0, 30, 60, and 90 degrees from the first eigenvector of $\mathbf{R}_{\mathbf{x}\mathbf{x}}$. After generating the data to meet these specifications

(eight different datasets), we fit logistic regression models to each dataset and computed fungible weights. Table 7 displays the logistic regression coefficients for these eight fitted models.

Table 7: Coefficients for Models Generated from $R_{\mathcal{B}}^2 = .20$ and $\kappa = \{3, 9\}$

		Location of β_i	$\beta_{i[1]}^a$	$\beta_{i[2]}$	$\beta_{i[3]}$
$\kappa = 3$	Model 9	$\beta_i \mathbf{v}_1^b$	0.26	-0.53	0.42
	Model 10	$\angle(\beta_i, \mathbf{v}_1) = 30^\circ$	0.13	-0.08	0.72
	Model 11	$\angle(\beta_i, \mathbf{v}_1) = 60^\circ$	0.07	0.25	0.98
	Model 12	$\beta_i \mathbf{v}_3$	-0.23	1.27	1.17
$\kappa = 9$	Model 13	$\beta_i \mathbf{v}_1$	0.68	0.32	0.37
	Model 14	$\angle(\beta_i, \mathbf{v}_1) = 30^\circ$	-0.10	-0.15	1.08
	Model 15	$\angle(\beta_i, \mathbf{v}_1) = 60^\circ$	-0.52	-0.47	1.44
	Model 16	$\beta_i \mathbf{v}_3$	-3.79	-3.16	4.49

^a $\beta_{i[j]}$ denotes the j th element of β_i .

^b The notation $\mathbf{a} || \mathbf{b}$ denotes that vectors $\mathbf{a}$ and $\mathbf{b}$ are parallel.

Table 7 displays the coefficients for the aforementioned eight logistic regression models. As with Table 6, the models in Table 7 were generated in pairs. For example, Models 9 and 13 were both generated from $\beta_i || \mathbf{v}_1$ and $R_{\mathcal{B}}^2 = .20$. These two models only differ in their predictor correlation matrices. The top of half of Table 7 corresponds to the correlation matrix generated to yield $\kappa = 3$, whereas the bottom half of Table 7 corresponds to the correlation matrix generated to yield $\kappa = 9$. Models 10 and 14, 11 and 15, and 12 and 16 follow this pattern as well. The effect of different correlation matrices (for $\kappa = 3$ and $\kappa = 9$) on the fungible weight contours can be seen in Figure 19.

Figure 19: Fungible Weight Contours for Models 9 and 13

The red fungible weight contours correspond to the $\kappa = 9$ $\mathbf{R}_{xx}$, $R_{\beta}^2 = .20$, and $\beta_i ||\mathbf{v}_1$ condition. The blue fungible weight contours correspond to the $\kappa = 3$ $\mathbf{R}_{xx}$, $R_{\beta}^2 = .20$, and $\beta_i ||\mathbf{v}_1$ condition.

Figure 19 displays the fungible weight contours for Models 9 and 13. The red fungible weight contours correspond to the $\kappa = 3$ condition, whereas the blue fungible weight contours correspond to the $\kappa = 9$ condition. Notice that the fungible weight contours for the $\kappa = 3$ condition are all smaller and in a different orientation than the corresponding fungible weight contours for the $\kappa = 9$ condition. Although this finding is similar to what we saw in Figures 14–17 where we compared models for different coefficients of determination, the difference in contours is not due to coefficient of determination as it was held constant across the two models at the latent level.⁸ Rather, the difference in size and orientation is determined by the different correlational structures associated with $\kappa = 3$ and $\kappa = 9$. Specifically, the eigenvalues of these two correlation matrices determine the shape and size of the fungible weight contours. The eigenvalues of the $\kappa = 3$ correlation matrix are $\{1.92, 0.87, 0.21\}$, whereas eigenvalues of the $\kappa = 9$ correlation matrix are $\{1.82, 1.16, 0.02\}$. Recall from Section 2.3.2, that the size of the eigenvalues determines the shape of the hyper-ellipsoid defined by $\beta_i' \mathbf{R}_{xx} \beta_i = R_{\beta}^2$ via the equation for the semi-axis lengths

$$l_i = \sqrt{\frac{R_{\beta}^2}{\lambda_i}},$$

where l_i denotes the i th semi-axis length, and λ_i denotes the i th ordered eigenvalue ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$) of $\mathbf{R}_{xx}$. Again, we note that larger eigenvalues are associated with smaller semi-axis lengths, whereas smaller eigenvalues are associated with larger semi-axis lengths. Recall from Section 2.3.2 that the fungible weight contours for the Ellipsoid Method are greatly affected by the ellipsoid semi-axes' lengths (see Figure 4). Interestingly, it would appear from Figure 19 that the Log-Likelihood Method is affected in a similar fashion. However, we note that the Log-Likelihood Method does not rely on the $\beta_i' \mathbf{R}_{xx} \beta_i = R_{\beta}^2$ fit function, and therefore the exact

⁸It should be noted that though R_{β}^2 was held constant at the latent level, this does not imply that the model fit of the two models were equal. Specifically, $R_L^2 = .11$ for Model 9, whereas $R_L^2 = .13$ for Model 13. Though this difference in model fit does affect the fungible weight contours (see Section 2.3.1 and Figures 14–17), it does not dramatically alter the shape of the contours as we see in Figure 19.

effect of the correlation structure on the shape of the fungible weight contour is less predictable than it is in the Ellipsoid Method. That being said, Figure 19 does suggest that the correlation structure has an analogous effect on the Log-Likelihood Method for computing fungible weights as it does in the Ellipsoid Method. Namely, the eigenstructure determines the shape of the fungible weight contours. We reach a similar conclusion inspecting Figures 20–22.

Figures 20–22 display the fungible weight contours for Models 10 and 14, Models 11 and 15, and Models 12 and 16, respectively. The red fungible weight contours correspond to the $\kappa = 3$ conditions, whereas the blue fungible weight contours correspond to the $\kappa = 9$ conditions. In these figures, we see a similar pattern to the results summarized in Figure 19. Namely, the $\kappa = 3$ fungible weight contours are all smaller and in a different orientation than their $\kappa = 9$ counterparts. Again, this finding is not surprising given the results of Figure 19 and Section 2.3.2. Despite this similarity among the four figures, Figure 22 stands out. Notice in each Panel that the two fungible weight contours are much more separated than they are in Figures 19–21. This finding is, again, due to the structure of the correlation matrix. At the latent level, both vectors of regression coefficients are collinear with the last eigenvector of the correlation matrix. Examining the eigenvalues discussed earlier, we can see that the last ellipsoidal axis of the $\kappa = 9$ condition is significantly longer than the $\kappa = 3$ condition. This longer axis allows for longer vectors of regression coefficients, which translates to bigger coefficients. Table 7 shows that the coefficients for Model 16 are much larger than the coefficients for Model 12. This finding explains the larger degree of separation we see in Figure 22. In aggregate, the findings from these conditions suggest that the structure of the predictor correlation matrix affects the Log-Likelihood Method of generating fungible weights in a similar fashion as the Ellipsoid Method. Namely, the eigenstructure of the correlation matrix determines the shape (and in some ways the size) of the fungible weight contours,

all else being equal. Moreover, these findings are replicated by the other conditions that were outlined in Section 5 (see Appendix A for graphical summaries of all conditions).

Figure 20: Fungible Weight Contours for Models 10 and 14

The red fungible weight contours correspond to the $\kappa = 3 \mathbf{R}_{xx}$, $R_\beta^2 = .20$, and $\angle(\beta_i, \mathbf{v}_1) = 30^\circ$ condition. The blue fungible weight contours correspond to the $\kappa = 9 \mathbf{R}_{xx}$, $R_\beta^2 = .20$, and $\angle(\beta_i, \mathbf{v}_1) = 30^\circ$ condition.

Figure 21: Fungible Weight Contours for Models 11 and 15

The red fungible weight contours correspond to the $\kappa = 3$ $\mathbf{R}_{xx}$, $R_{\beta}^2 = .20$, and $\angle(\beta_i, \mathbf{v}_1) = 60^\circ$ condition. The blue fungible weight contours correspond to the $\kappa = 9$ $\mathbf{R}_{xx}$, $R_{\beta}^2 = .20$, and $\angle(\beta_i, \mathbf{v}_1) = 60^\circ$ condition.

Figure 22: Fungible Weight Contours for Models 12 and 16

The red fungible weight contours correspond to the $\kappa = 3$ $\mathbf{R}_{xx}$, $R_{\beta}^2 = .20$, and $\beta_i ||\mathbf{v}_3$ condition. The blue fungible weight contours correspond to the $\kappa = 9$ $\mathbf{R}_{xx}$, $R_{\beta}^2 = .20$, and $\beta_i ||\mathbf{v}_3$ condition.

6.1.3 Orientation of β With Respect to the Eigenvectors of R_{xx}

Having reviewed how different correlation matrices affect the Log-Likelihood Method of computing fungible weight for logistic regression coefficients, we now turn our attention to the orientation of the vector of regression coefficients with respect to the eigenvectors of R_{xx} . To that end, we compare the fungible weight contours for the four logistic regression models fit to the data generated to have the $\kappa = 3$ predictor correlation matrix reported earlier and a coefficient of determination of $R^2_{\beta} = .80$ (all fungible weight contour results are in Appendix A). Table 8 displays the coefficients for these four fitted models.

Table 8: Coefficients for Models Generated from $R^2_{\beta} = .80$ and $\kappa = 3$

	Location of β_i	$\beta_{i[1]}^a$	$\beta_{i[2]}$	$\beta_{i[3]}$
Model 17	$\beta_i \mathbf{v}_1^b$	1.23	-2.09	1.75
Model 18	$\angle(\beta_i, \mathbf{v}_1) = 30^\circ$	0.61	-0.47	3.01
Model 19	$\angle(\beta_i, \mathbf{v}_1) = 60^\circ$	-0.14	1.03	4.51
Model 20	$\beta_i \mathbf{v}_3$	-0.98	4.14	4.62

^a $\beta_{i[j]}$ denotes the j th element of β_i .

^b The notation $\mathbf{a} || \mathbf{b}$ denotes that vectors $\mathbf{a}$ and $\mathbf{b}$ are parallel.

As seen in Table 8, Model 17 corresponds to the $\beta_i || \mathbf{v}_1$ condition, Model 18 corresponds to the $\angle(\beta_i, \mathbf{v}_1) = 30^\circ$ condition, Model 19 corresponds to the $\angle(\beta_i, \mathbf{v}_1) = 60^\circ$ condition, and Model 20 corresponds to the $\beta_i || \mathbf{v}_3$ condition. Because all four models differ only in terms of the position of the vector of regression coefficients, we can see how this factor affects the fungible weight contours. To see this effect, we graphically summarize the fungible weight contours for only Models 17 and 20 as the results for the other models are similar (see Appendix A for graphical summaries of all conditions). Before we examine the fungible weight contours for

these models, we first note that unlike in previous graphical summaries, we center the contours at the origin. We centered the fungible weight contours because the range of coefficient values differ between the two models. The ranges of the fungible weights for each model are displayed in Table 9.

Table 9: Range of Fungible Weights for Models 17 and 20

Location of β_i		$\beta_{i[1]}^a$		$\beta_{i[2]}$		$\beta_{i[3]}$	
		Low	High	Low	High	Low	High
Model 17	$\beta_i \mathbf{v}_1^b$	0.75	1.78	-2.93	-1.36	1.01	2.59
Model 20	$\beta_i \mathbf{v}_3$	-1.41	-0.58	3.22	5.22	3.62	5.82

^a $\beta_{i[j]}$ denotes the j th element of β_i .

^b The notation $\mathbf{a} || \mathbf{b}$ denotes that vectors $\mathbf{a}$ and $\mathbf{b}$ are parallel.

As can be seen in Table 9, the ranges of fungible weights for a given coefficient are not the same. For example, the range for coefficient 2 for Model 17 is -2.93 through -1.36 , whereas the range for coefficient 2 for Model 20 is 3.22 through 5.22 . Because the ranges are disparate, plotting the contours uncentered would make comparison difficult.

Figure 23 displays the fungible weight contours for Models 17 and 20. The red fungible weight contours correspond to Model 20, whereas the blue fungible weight contours correspond to Model 17. Panel A displays the contours for coefficients 1 and 2, Panel B displays the contours for coefficients 1 and 3, and Panel C displays the contours for coefficients 2 and 3. First, notice in Panel A that the two fungible contours are similar, whereas in Panels B and C the fungible contours are different. In Panel B, we see that the contour for Model 20 is in a different orientation than the contour for Model 17, but the shapes of the two contours are similar. However, in Panel C we see the reverse. Specifically, the alignment (or orientation) of the two

Figure 23: Fungible Weight Contours for Models 17 and 20.

The red fungible weight contours correspond to the $\beta_i ||v_i||$, $R_{\beta}^2 = .80$, and $\kappa = 3 \mathbf{R}_{xx}$ condition. The blue fungible weight contours correspond to the $\beta_i ||v_i||$, $R_{\beta}^2 = .80$, and $\kappa = 3 \mathbf{R}_{xx}$ condition.

contours are similar, whereas the shapes are different. The shape of the contour for Model 17 is cigar-shaped, whereas the shape of the contour for Model 20 is egg-shaped. These findings are in accord with the Ellipsoid Method as shown in Section 2.3.3. Namely, as the vector of regression coefficients moved in space, the shape and orientation of the fungible weight contours changed. Though these changes are predictable in the Ellipsoid Method (due to the geometry), these changes are less predictable in the Log-Likelihood Method. Moreover, the coefficients depend both on the (latent) R_{β}^2 and the predictor correlation structure. In other words, moving the vector of regression coefficients from the first to last eigenvector (as we do in Models 17 and 20) will not have the same effect on the fungible weight contours computed from different values of $\mathbf{R}_{xx}$ and R_{β}^2 .

Before we leave this section, we should point out that we have only reviewed fungible weight contours for simulation conditions involving three predictors. We did not review the results from the five predictor conditions because the results for the $p = 5$ conditions for different R_{β}^2 , $\mathbf{R}_{xx}$, and β were similar to those in the corresponding $p = 3$ conditions as shown in Figures 14–23. In Section 2.3.4, we claimed that the number of predictors should have an effect on the fungible weight contours, because the number of predictors limits the correlational structure. However, it appears that moving from 3 to 5 predictors does not alter the correlational structure enough to have an appreciable effect on the fungible weight contours. Because of this finding, we chose to focus on the $p = 3$ conditions, as they are easier to compare across conditions due to the limited number of fungible weight pairs (3 for three predictors versus 10 for five predictors).

6.1.4 Comparing the Ellipsoid and Log-Likelihood Methods

Having reviewed how the different simulation factors affect the Log-Likelihood Method of generating fungible weight contours, we now compare the Log-Likelihood and El-

lipsoid Methods for four simulation conditions. Because these two methods rely on different fit functions, different aspects of the data may affect the fungible weight contours in different ways. To explore this possibility, we examine the fungible weight contours from the three predictor logistic regression models that were constructed from the single-factor correlation matrix reported earlier, latent $R_{\beta}^2 = .20$, and β_i oriented at 0, 30, 60, and 90 degrees from the first eigenvector of $\mathbf{R}_{xx}$. We chose these four conditions as they seem to reflect the general pattern of results of all the simulation conditions. In other words, these four conditions provide a good summary of how the fungible weight contours differ between the Ellipsoid and Log-Likelihood Methods. To begin, we report the estimated logistic regression coefficients from these models in Table 10.

Table 10: Logistic Regression Coefficients for Models With $R_{\beta}^2 = .20$ and $\mathbf{R}_{xx} =$ One-Factor

	Location of β_i	$\beta_{i[1]}^a$	$\beta_{i[2]}$	$\beta_{i[3]}$
Model 21	$\beta_i \mathbf{v}_1^b$	-0.47	-0.22	-0.41
Model 22	$\angle(\beta_i, \mathbf{v}_1) = 30^\circ$	-0.40	-0.69	-0.10
Model 23	$\angle(\beta_i, \mathbf{v}_1) = 60^\circ$	-0.22	-0.84	0.26
Model 24	$\beta_i \mathbf{v}_6$	0.06	-1.11	0.97

^a $\beta_{i[j]}$ denotes the j th element of β_i .

^b The notation $\mathbf{a} || \mathbf{b}$ denotes that vectors $\mathbf{a}$ and $\mathbf{b}$ are parallel.

As seen in Table 10, Model 21 corresponds to the $\beta_i || \mathbf{v}_1$ condition, Model 22 corresponds to the $\angle(\beta_i, \mathbf{v}_1) = 30^\circ$ condition, Model 23 corresponds to the $\angle(\beta_i, \mathbf{v}_1) = 60^\circ$ condition, and Model 24 corresponds to the $\beta_i || \mathbf{v}_3$ condition. For each of these models we generated 100,000 fungible weights using both the Ellipsoid and Log-Likelihood Methods. The results of these simulations are summarized in Figures 24–27.

Figure 24: Fungible Weight Contours for Model 21

The red fungible weight contours correspond to the Ellipsoid Method for the $R_\beta^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\beta_i ||\mathbf{v}_1$ condition. The blue fungible weight contours correspond to the Log-Likelihood Method for the $R_\beta^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\beta_i ||\mathbf{v}_1$ condition.

Figure 24 displays the Ellipsoid and Log-Likelihood Method’s fungible weight contours for the coefficients in Model 21. The red fungible weight contours correspond to the Ellipsoid Method, whereas the blue fungible weight contours correspond to the Log-Likelihood Method. First, notice that the red fungible weight contour is not filled in (i.e., not a solid), whereas the blue fungible weight contour is. This difference is a consequence of using the Ellipsoid Method to construct fungible weights for three predictors. With three predictors, $\beta' \mathbf{R}_{xx} \beta = R_\beta^2$ defines a three-dimensional ellipsoid. Any planar intersection with a three-dimensional ellipsoid results in a two-dimensional ellipse embedded in three-space (Ferguson, 1979). Thus, each fungible weight contour created by the Ellipsoid Method for this condition (and all three predictor conditions) is a projection of an ellipse into one of the coordinate axis planes. However, once we move beyond three predictors, $\beta' \mathbf{R}_{xx} \beta = R_\beta^2$ defines a hyper-ellipsoid. Any hyper-planar intersection with a p -dimensional hyper-ellipsoid results in a $p - 1$ dimensional (hyper) ellipsoid. Projections of any (hyper) ellipsoid into one of the coordinate axis planes results in an ellipse with a filled surface. Appendix A contains examples of this effect for the $p = 5$ conditions.

A second aspect of Figure 24 that warrants comment is that the fungible weight contours for the Ellipsoid Method are all smaller than their Log-Likelihood counterparts and embedded within them. We see this result because the Ellipsoid and Log-Likelihood Methods rely on different fit functions, and therefore different model perturbations. Because the model fit and perturbations differ between the two methods, one should not expect the contours to match. Moreover, the exact amount of perturbation for each method is not equivalent as there is not a direct (or at least a unique) translation between a correlation (the Ellipsoid Method fit criteria) and a log-likelihood function value (the Log-Likelihood Method fit criteria). Because of this difference, the Ellipsoid Method fungible weight contours are sometimes em-

bedded in the Log-Likelihood contours (as seen in Figure 24, and Figures 25–27), whereas in other conditions, the Log-Likelihood Method contours are sometimes embedded in the Ellipsoid Method contours (e.g., compare Figures 63 and 67 in Appendix A). However, just because the Ellipsoid Method contours are embedded in the Log-Likelihood Method contours in Figure 24, this finding does not imply that the Ellipsoid Method fungible weight vectors yield the same log-likelihood value (when plugged into (68); see Section 3.3) that the Log-Likelihood Method fungible weight vectors yield. Table 11 illustrates this point.

Table 11: Correlations and Log-Likelihood Values for the Fungible Weights from Model 21

Method	Fit ^a	Min	1st Q	Median	Mean	3rd Q	Max
EM ^b	$\ln L$	-308.219	-308.203	-308.192	-308.192	-308.177	-308.165
	$r_{\hat{\mathbf{y}}_{\alpha}\hat{\mathbf{y}}_{\beta}}$	0.967	0.967	0.967	0.967	0.967	0.967
LLM ^c	$\ln L$	-308.785	-308.785	-308.785	-308.785	-308.785	-308.785
	$r_{\hat{\mathbf{y}}_{\alpha}\hat{\mathbf{y}}_{\beta}}$	0.959	0.964	0.968	0.970	0.972	1.000

^a Fit denotes the fit value for each method.

^b EM denotes the Ellipsoid Method.

^c LLM denotes the Log-Likelihood Method.

Table 11 displays the ranges of log-likelihood values ($\ln L$) and correlations ($r_{\hat{\mathbf{y}}_{\alpha}\hat{\mathbf{y}}_{\beta}}$; recall that $r_{\hat{\mathbf{y}}_{\alpha}\hat{\mathbf{y}}_{\beta}}$ denotes the correlation between the predicted log-odds formed by the original model coefficients and the predicted log-odds formed by the fungible weight coefficients) for both the Ellipsoid Method (EM) and the Log-Likelihood Method (LLM). Notice, that as expected, the Log-Likelihood Method fungible weight vectors yield a constant log-likelihood value of -308.785. However, the Ellipsoid Method fungible weight vectors yield log-likelihood values that range from -308.219

to -308.165 . Further, we see that all of the log-likelihood values from the Ellipsoid Method are larger than the log-likelihood value from the Log-Likelihood Method. This finding is due to the somewhat arbitrary way we set the decrement in model fit for both methods. Had we chosen different values for either (or both) method(s), we could have seen the opposite result. However, for this condition, the difference in log-likelihood values explains why the Ellipsoid Method fungible weight contour is embedded within the Log-Likelihood Method contour. Specifically, the Ellipsoid Method contour occurs higher up on the log-likelihood function than the Log-Likelihood Method contour (see Figures 7 and 8 in Section 2.4 for reference). Because the Ellipsoid Method contour occurs higher up in the log-likelihood function, projecting the contour into the (hyper-) plane orthogonal to the axis represented by the log-likelihood function value ($\ln L$ in Figures 7 and 8) results in a smaller contour that is embedded with the Log-Likelihood Method contour.

A second aspect of Table 11 that warrants comment is that the $r_{\hat{y}_\alpha \hat{y}_\beta}$ formed by Log-Likelihood Method fungible weight vectors is not constant. Although this finding is not surprising given what we saw in Figure 24, the range of correlations, 0.9591992 – 0.999998 , (in addition to the range of log-likelihood values in the Ellipsoid Method) reinforces the fact the Log-Likelihood Method operates on a different fit function than the Ellipsoid Method, and one should not expect that the two methods yield fungible weight vectors with equivalent model fit. Overall, the findings in Table 11 suggest, that at least for this example, the Ellipsoid Method could not produce a fungible weight vector that yields the log-likelihood value set by the Log-Likelihood Method, whereas the Log-Likelihood Method could potentially produce a fungible weight vector that yields the $r_{\hat{y}_\alpha \hat{y}_\beta}$ set by the Ellipsoid Method.

Before leaving Figure 24, we want to draw the reader’s attention to an additional detail. Notice that the shape and orientation of the fungible weight contours computed using the Ellipsoid and Log-Likelihood Methods are in general agree-

ment. However, this finding is not always the case as seen in Figures 25–27.

Figure 25: Fungible Weight Contours for Model 22

The red fungible weight contours correspond to the Ellipsoid Method for the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, \mathbf{v}_1) = 30^\circ$ condition. The blue fungible weight contours correspond to the Log-Likelihood Method for the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, \mathbf{v}_1) = 30^\circ$ condition.

Figure 26: Fungible Weight Contours for Model 23

The red fungible weight contours correspond to the Ellipsoid Method for the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, \mathbf{v}_1) = 60^\circ$ condition. The blue fungible weight contours correspond to the Log-Likelihood Method for the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\angle(\beta_i, \mathbf{v}_1) = 60^\circ$ condition.

Figure 27: Fungible Weight Contours for Model 24

The red fungible weight contours correspond to the Ellipsoid Method for the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\beta_i || \mathbf{v}_3$ condition. The blue fungible weight contours correspond to the Log-Likelihood Method for the $R_{\beta}^2 = .20$, one-factor $\mathbf{R}_{xx}$, and $\beta_i || \mathbf{v}_3$ condition.

Figures 25–27 display the fungible weight contours for Models 22–24. Again, the red fungible weight contours correspond to the Ellipsoid Method, whereas the blue fungible weight contours correspond to the Log-Likelihood Method. Notice in Panels A and C of Figure 25 that the fungible contours for the Ellipsoid Method begin to tilt in a different direction than the corresponding fungible weight contours for the Log-Likelihood Method. This tilting continues as we move β toward the last eigenvector of $\mathbf{R}_{xx}$ as depicted in Figures 26 and 27. In Figure 26, when $\angle(\beta, \mathbf{v}_1) = 60^\circ$, we see in Panels A and C that the fungible weight contours for the Ellipsoid Method continue to rotate (counter-clockwise in Panel A and clockwise in Panel C) and become more narrow. Finally, in Figure 27, we see in all three Panels that the Ellipsoid Method contours rotate and change shape. The most striking example of this finding is in Panel C of Figure 27. Notice that the Ellipsoid Method contour resembles a line. Note, however, that it is not a line, but rather a very thin ellipse. The results demonstrated in Panel C of Figure 27, as well as the results from all Panels in Figures 25–27, are due to the orientation of β with respect to the eigenvectors of $\mathbf{R}_{xx}$. Recall from Section 2.3.3, that the orientation of β can have a profound effect on the fungible weight contours generated from the Ellipsoid Method. However, what was not apparent in Section 2.3.3 was how this factor differentially affected the Ellipsoid and Log-Likelihood Methods. These results and those reported in earlier Figures (as well as those in Appendix A) suggest that the orientation of β with respect to $\mathbf{R}_{xx}$ has a stronger affect on the fungible weights computed from the Ellipsoid Method than it does on the fungible weights computed from the Log-Likelihood Method.

The results reported in Figures 14–27 provide a good summary of how the Ellipsoid Method compares to the Log-Likelihood Method. Recall that we began by examining how the Log-Likelihood Method for computing fungible weights was affected by three design factors: (1) latent coefficient of determination, (2) structure

of the predictor correlation matrix, and (3) orientation of the vector of regression coefficients with respect to the eigenvectors of $\mathbf{R}_{xx}$. First, we saw that coefficient of determination affected the Log-Likelihood Method in a similar fashion as the Ellipsoid Method. Namely, that larger coefficients of determination yield larger fungible weight contours in an *absolute* sense, but generally yield smaller contours *relative* to their logistic regression coefficients, all else being equal. Second, we saw that the structure of the predictor correlation matrix also affects the Log-Likelihood Method in a similar way as the Ellipsoid Method. That is, the eigenstructure of the correlation matrix determines the shape (and size) of the fungible weight contours, all else being equal. Third, we again saw that the orientation of the vectors of regression coefficients affected the Log-Likelihood Method in a similar way as the Ellipsoid Method. Specifically, as the orientation of β changes, the shape and orientation of the fungible weight contours changes (see Figure 27). While these changes were somewhat predictable in the Ellipsoid Method due to its geometry, they were less predictable in the Log-Likelihood Method. As mentioned earlier, when a plane intersects with an ellipsoid, the intersection solid is a $p - 1$ dimensional ellipsoid. Moreover, because the structure of the ellipsoid is determined by the correlation matrix, and the position of the plane is determined by β , we can predict the orientation of the fungible weight contour (at least for three predictors). However, it is difficult to predict the orientation of the Log-Likelihood Method contours because the way $\mathbf{R}_{xx}$ affects the log-likelihood surface is less clear than how $\mathbf{R}_{xx}$ affects $\beta_i' \mathbf{R}_{xx} \beta_i = R_{\beta}^2$.

6.2 Examining the Effect of Incorrect Model Specification

One shortcoming of the previous simulation study is that all the models used were fully specified. In other words, we were working with the full model that was specified in the population—there were no extra or missing predictors. Our simu-

lation then, is not entirely representative of what researchers face in applied problems. To get a better idea of how incorrectly specified models affect the fungible weight contours, we ran an additional simulation with a single model. Specifically, we generated $n = 500$ cases for a five predictor logistic regression model with a correlation matrix generated to yield a $k = 3$,

$$\mathbf{R}_{xx} = \begin{bmatrix} 1.00 & 0.01 & 0.39 & -0.03 & 0.33 \\ 0.01 & 1.00 & 0.10 & 0.04 & -0.29 \\ 0.39 & 0.10 & 1.00 & -0.26 & -0.38 \\ -0.03 & 0.04 & -0.26 & 1.00 & -0.22 \\ 0.33 & -0.29 & -0.38 & -0.22 & 1.00 \end{bmatrix},$$

a coefficient of determination with the latent criterion variable of $R_{\beta}^2 = .50$, and a vector of regression coefficients at the latent level of

$$\beta^* = \begin{bmatrix} -0.98 & 0.96 & 2.11 & 1.53 & 2.28 \end{bmatrix}'$$

(note that $\beta = [-0.38 \ 0.37 \ 0.82 \ 0.60 \ 0.60]'$ such that $\beta' \mathbf{R}_{xx} \beta = .50$). After generating the data to yield these parameter values, we dichotomized the latent criterion as outlined in Section 5.2 and fit two logistic regression models. The first model was the correctly specified model including all five predictors, whereas the second model was the incorrectly specified model including only the first three predictors. Table 12 reports the logistic regression coefficient values for these two models.

Table 12: Logistic Regression Models for the Correct and Incorrect Specification

Variables	Coefficients	
	Correct	Incorrect
Intercept	0.04	0.04
X_1	2.20	0.84
X_2	0.34	0.78
X_3	2.67	1.06
X_4	0.37	—
X_5	2.09	—

Note: $R_L^2 = 0.34$ for the correct model; $R_L^2 = .14$ for the incorrect model.

Notice in Table 12 that the coefficient values are quite different between the two models. This finding is not surprising because more variables were included in the correct model which affects model fit— $R_L^2 = .34$ for the correct model vs. $R_L^2 = .14$ for the incorrect model (note at the latent level $R_\beta^2 = .50$ for the correct model and $R_\beta^2 = .15$ for the incorrect model). Because the coefficients are different, one would expect that the fungible weight contours for the two models would be different as well. To examine this possibility, we computed fungible weights using both the Ellipsoid Method and the Log-Likelihood Method. We first describe the Ellipsoid Method results.

To compute fungible weights using the Ellipsoid Method (see Section 3.2), we set the correlation between the log-odds linear composite formed by the logistic regression coefficients and the log-odds linear composite formed by the fungible weights to $r_{\hat{y}_a \hat{y}_b} = 0.99$. We then computed 100,000 fungible weight vectors for both the incorrectly and correctly specified logistic regression model reported in Table 12. Figure 28 displays these results.

Figure 28: Ellipsoid Method for the Incorrect and Correct Model

The red fungible weight contours correspond to the incorrectly specified model, whereas the blue fungible weight contours correspond to the correctly specified model.

Figure 28 displays three panels, each of which displays two fungible weight contours computed using the Ellipsoid Method. The red fungible weight contours correspond to the incorrectly specified model, whereas the blue fungible weight contours correspond to the correctly specified model. Coefficient a_1 represents the fungible weight for X_1 , coefficient a_2 represents the fungible weight for X_2 , and coefficient a_3 the fungible weight for X_3 . There are three aspects of Figure 28 that warrant comment. First, notice that in each panel, the fungible contour for the incorrectly specified model is smaller than the fungible contour for the correctly specified model. This finding is not surprising, however, because the two models differ in terms of model fit. The correctly specified model had a latent coefficient of determination of $R^2_{\beta} = .50$, whereas the incorrectly specified model had a latent coefficient of determination of $R^2_{\beta} = .15$. This difference in R^2_{β} leads to different sized fungible weight contours, as described in Section 2.3.2 and earlier results. Second, notice in each panel that the fungible contours do not occupy the same position in space. This finding is also not surprising given how different the fitted logistic regression coefficient values were for the two models (see Table 12). Third, notice that in Panels A and C (and to a lesser extent in Panel B) that the shapes of the two contours are in slightly different orientations. This difference in orientation is due to the geometry of the intersection of the hyper-plane and hyper-ellipsoid used to compute the fungible weights. Specifically, adding additional predictors changes the orientation of the regression coefficients with the eigenvectors of the predictor correlation matrix. For example, consider a vector of regression coefficients that is collinear with the first eigenvector of a three-predictor correlation matrix. Now, suppose we have added two additional predictors. The resulting vector of regression coefficients is very unlikely to be collinear with the first eigenvector of the new correlation matrix. This change affects the shape of the fungible weight contours as described in Section 2.3.2 and 2.3.3. In aggregate these findings suggest that the fungible weight

contours differ between correctly and incorrectly specified models. We find a similar pattern of results for the fungible weights computed using the Log-Likelihood Method.

To compute fungible weights using the Log-Likelihood Method (see Section 3.3), we reduced the fit of each model. Recall from Table 12 that $R_L^2 = .34$ for the correct model and $R_L^2 = .14$ for the incorrect model. Therefore, we reduced both R_L^2 values by 0.01 and computed 100,000 fungible weight vectors for each model. Figure 29 displays these results.

Figure 29: Log-Likelihood Method for the Incorrect and Correct Model

The red fungible weight contours correspond to the incorrectly specified model, whereas the blue fungible weight contours correspond to the correctly specified model.

Figure 29 displays three panels, each of which displays two fungible weight contours computed using the Log-Likelihood Method. The red fungible weight contours correspond to the incorrectly specified model, whereas the blue fungible weight contours correspond to the correctly specified model. Notice in Figure 29 that the graphical summary of the fungible weights resembles the pattern of results seen in Figure 28. Namely, the size and position of the fungible weight contours differ. However, unlike the fungible weight contours in Figure 28, the two contours in each panel of Figure 29 are in a similar alignment. In other words, the orientations of the fungible contours computed from the correctly and incorrectly specified model are very similar despite different information being used to construct the two. However, this finding may be a quirk of these particular data. Future research will need to explore this finding more carefully.

Before leaving this section, a final point warrants discussion. Although the fungible weight contours differed markedly between the incorrectly and correctly specified models, these findings cannot, and should not be used to judge whether a model is “correct”. In this particular example, the correct model had larger fungible weight contours because the correct model had more predictors, and therefore a larger R^2_{β} , than the incorrect model. In applied research the reverse could also be true. That is, an applied researcher might include predictors that do not belong in the model, and the researcher would never know. Fungible weights should be used to assess the parameter sensitivity of a model’s coefficients to aid in interpretation, rather than be used to judge whether the model under question is the “correct model”.

7 Discussion/Future Directions

Logistic regression is one of the most popular models for predicting dichotomous outcomes in medicine (Tu, 1996), social sciences (e.g., Psychology: Pamper, 2000), and machine learning (Perlich, Provost, & Simonoff, 2003). Due to its popularity, methods for fitting logistic regression models are available in virtually all statistical packages (e.g., R, SAS, SPSS). Moreover, all of these packages provide a means to evaluate the model in terms statistical tests and confidence intervals. However, one evaluative technique that these packages do not provide is a method of assessing parameter sensitivity. With this point in mind, we set out to develop and test parameter sensitivity methods in logistic regression. Specifically, we extended Waller's (2008; Waller & Jones, 2009) work on fungible weights to logistic regression (which we call the Ellipsoid Method), and we derived a log-likelihood based method of computing fungible weights for logistic regression (which we call the Log-Likelihood Method). To motivate the use of these methods in assessing parameter sensitivity, we reanalyzed two studies: (1) the Aspirin Component of the Physicians' Health Study (Aspirin Study; Steering Committee of the Physicians Health Study Research Group, 1988), and (2) the 2009 CDC Youth Risk Behavior Surveillance Survey results on cigarette use among high school students (Smoking Study; Adwere-Boamah, 2009). Of interest was whether the authors' interpretations of the logistic regression coefficients was consistent with an in depth sensitivity analysis. We found that the coefficient interpretations offered in the Aspirin Study held up under the sensitivity analysis, whereas the coefficient interpretations offered in the Smoking Study did not hold up under a sensitivity analysis. In other words, we showed how computing fungible weights can either lend support to conclusions or call conclusions into question. To better understand what aspects of the data affect the computation of fungible weights, we conducted a large-scale simulation study.

In our simulation study we varied four factors: (1) the coefficient of determination at the latent level, R_β^2 , (2) the structure of the predictor correlation matrix, $\mathbf{R}_{xx}$, (3) the orientation of the vectors of regression coefficients with the eigenvectors of $\mathbf{R}_{xx}$, and (4) the number of predictors, p . Our findings suggest that the fungible weights computed from both the Ellipsoid and Log-Likelihood Methods are affected by these factors in a similar manner. First, we found that larger coefficients of determination yield larger *absolute* fungible weight contours than smaller coefficients of determination, all else being equal (i.e., fixed $\mathbf{R}_{xx}$ and orientation of β with respect to the eigenvectors of $\mathbf{R}_{xx}$). However, larger coefficients of determination generally yield smaller *relative* fungible weight contours than smaller coefficients of determination (again, all else being equal). In other words, when the fungible weights were divided by the original logistic regression coefficients (i.e., the model coefficients that were used to generate the fungible weights), we found that larger coefficients of determination yielded smaller fungible weight contours than smaller coefficients of determination. Second, we found that the eigenstructure of the correlation matrix determines the shape (and in some ways the size) of the fungible weight contours, all else being equal. This finding was particularly salient when comparing the $\kappa = 3$ to the $\kappa = 9$ condition with three predictors, $R_\beta^2 = .20$, and $\beta \parallel \mathbf{v}_1$. The Log-Likelihood Method fungible weight contours for the $\kappa = 3$ condition were somewhat egg-shaped, whereas the contours for $\kappa = 9$ were cigar shaped. In other words, the range of values that fungible weights can take for a given decrement in model fit are somewhat controlled by the structure of the correlation matrix. Third, we found that the orientation of the vectors of regression coefficients with the eigenvectors of $\mathbf{R}_{xx}$ for a given R_β^2 and $\mathbf{R}_{xx}$ also affects the shape and orientation of the fungible weight contours. For example, in the condition where $R_\beta^2 = .80$ and $\kappa = 3$, the fungible weight contour associated with variables 2 and 3 was egg-shaped when $\beta \parallel \mathbf{v}_1$ and cigar-shaped when $\beta \parallel \mathbf{v}_3$. This find-

ing in conjunction with those regarding the correlational structure suggests that the shape and orientation of the fungible weight contours is jointly determined by both the orientation of the vector of regression and coefficients and the structure of the predictor correlation matrix. It should be noted that though the orientation of β affects both the Ellipsoid and Log-Likelihood Methods in a similar manner, the effect seems to be much more pronounced in the Ellipsoid Method (e.g., see Figure 27). Finally, we found that the number of predictors did not really have much of an affect on the shape, size, or orientation of the fungible weight contours. This finding was due to only examining models with three and five predictors. Had we looked at models with 10 or 15 predictors, we might have seen an effect due to the correlational structure being limited by the number of predictors. That being said, the number of predictors did affect the computation time required for the algorithms. Specifically, the conditions with $p = 5$ took longer to generate the same amount of fungible weights as the $p = 3$ conditions. This finding, however, is not surprising as the algorithm has to do more computations as the number of predictors is increased.

Though the Ellipsoid and Log-Likelihood methods respond similarly to different changes in the data, there are a couple of important differences between the two methods. First, the Ellipsoid and Log-Likelihood Methods rely on different fit functions. This fact makes direct comparisons difficult between the two methods as a difference could be due to the fit function, the particular design factor that is being varied, or a combination of the two. It is likely that the observed difference between the two methods regarding the orientation of β was a combination of these two sources. A second difference between the two methods lies in the interpretation of the weights. Recall that the Ellipsoid Method relies on the predicted log-odds from the model when computing fungible weight vectors. The fungible weight vectors computed from the Ellipsoid Method can then be used to create alternate

predicted log-odds that relate a predetermined amount (less than the optimal predicted log-odds) to some underlying latent variable, while being very similar to the optimal predicted log-odds. In other words, the Ellipsoid Method fungible weights are constructed based on the assumed underlying latent linear model, rather than the logistic model. The Log-Likelihood Method, on the other hand, operates directly on the coefficients in the observed logistic regression model, and therefore is not bound to an underlying latent linear model interpretation. That being said, the aforementioned simulation did not find much of a practical difference between the two methods except for computation time, with the Ellipsoid Method being quicker than the Log-Likelihood Method. Researchers who wish to compute fungible weights for logistic regression models can choose either approach, but bear in mind that the interpretation of the weights are tied to the fit function and, therefore, the underlying model of interest, be it a latent linear model or an observed logistic regression model.

Interestingly, the fungible weight algorithms derived in this paper for assessing parameter sensitivity can be easily extended to accommodate other types of models. One popular class of models that these algorithms can accommodate is the family of item response theory models (IRT; Embretson & Reise, 2000). Like logistic regression, IRT models can be used to model binary outcomes as a probability. However, what is unique about IRT models is that they are often used to model an examinee's incorrect/correct response to individual test items (e.g., a multiple choice type item) as a function of a latent examinee ability and various item properties. One popular IRT model is the unidimensional, binary, three-parameter logistic model

$$\Pr(y_{ij} = 1) = \Pr(y_{ij} = 1 | \theta_i, \alpha_j, \beta_j, \gamma_j) = \gamma_j + (1 - \gamma_j) \frac{\exp[\alpha_j(\theta_i - \beta_j)]}{1 + \exp[\alpha_j(\theta_i - \beta_j)]}, \quad (75)$$

where y_{ij} denotes the i th examinee's response to the j th item, θ_i denotes the i th examinee's ability, α_j denotes the item's discrimination parameter, β_j denotes the item's difficult parameter, γ_j denotes the item's pseudo-guessing parameter. Notice that the form of the item response function (IRF) in (75) is very similar to the logistic regression model. However, unlike the logistic regression model where the number of parameters is determined by the number of predictors, the number of parameters in the IRT model are determined by a combination of the number of examinees and the number of items. For example, for 100 examinees taking an exam composed of 20 items, each of which has three parameters, the total number of model parameters is 160, all of which need to be estimated. There are various ways of estimating the parameters (e.g., joint maximum likelihood, marginal maximum likelihood; see Embretson & Reise, 2000, p. 210–219), most of which rely on finding the maximum of the model likelihood equation

$$L(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{i=1}^N \prod_{j=1}^n \Pr(y_{ij} = 1)^{y_{ij}} [1 - \Pr(y_{ij} = 1)]^{1-y_{ij}}, \quad (76)$$

where $\boldsymbol{\theta}$ denotes an $N \times 1$ vector of examinee ability parameters, $\boldsymbol{\alpha}$ denotes an $n \times 1$ vector of item discrimination parameters, $\boldsymbol{\beta}$ denotes an $n \times 1$ vector of item difficulty parameters, and $\boldsymbol{\gamma}$ denotes an $n \times 1$ vector of pseudo-guessing parameters. As in logistic regression, one generally maximizes the log-likelihood equation,

$$\ln L(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{i=1}^N \sum_{j=1}^n y_{ij} \ln[\Pr(y_{ij} = 1)] + (1 - y_{ij}) \ln[1 - \Pr(y_{ij} = 1)]. \quad (77)$$

Once parameters have been estimated that maximize (77), we can use a variant of the Log-Likelihood Method (see Section 3.3) to generate fungible weights for both person and item parameters. As discussed earlier, to use the Log-Likelihood Method one requires the first derivatives of $\ln L(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ with respect to $\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\boldsymbol{\gamma}$, and a method of reducing the model fit. Calculating the derivatives of (77)

is relatively straightforward (see Embretson & Reise, 2000, p. 221; see also Baker, 1992, Ch. 2), whereas reducing the model fit is less straightforward. We could simply set the log-likelihood function value to one slightly lower than the maximum and generate fungible weights. However, as with logistic regression, the log-likelihood metric is not a very friendly metric to work in as it is difficult to know how much a decrement in model fit one is incurring. The way around this problem is to tie the log-likelihood function value to a metric that is easier to interpret as we did for logistic regression with the likelihood-ratio, pseudo coefficient of determination, R_L^2 . Unfortunately, in IRT, there are two types of fit indices: (1) person fit, and (2) item fit. Person fit indices assess how well a particular examinee fits the estimated IRT model, whereas item fit indices assess how well a particular item fits the estimated IRT model. These fit indices might provide a meaningful way to lower the model fit to allow for the computation of fungible weights (see Reise & Widaman, 1999). However, these fit indices act at the individual level. In other words, each item and each person will have its own model fit statistic, and therefore fungible weights would necessarily be created using individual log-likelihood functions rather than the joint log-likelihood function. Whether using individual log-likelihood functions makes a practical difference (aside from computation time) in computing fungible weights needs to be examined.

Once algorithms for assessing parameter sensitivity are available for IRT models they could be used to determine how sensitive item parameters are in large scale testing situations like computerized adaptive testing (CAT; see Weiss & Kingsbury, 1984). In CAT, the items in an item bank are calibrated (i.e., the item parameters are estimated), put on a common metric, and then the items are subsequently used on future examinees. In high stakes CAT (e.g., Graduate Record Examination; North American Pharmacist Licensure Examination) it is paramount that the items have been properly calibrated and are stable from one administration to another.

Future research could determine whether conducting a parameter sensitivity analysis via fungible weights could help the administrator choose items that are most likely to exhibit these properties.

As we have discussed, the goal of a parameter sensitivity analysis is to determine how much a model's parameter estimates can change without drastically affecting the fit of the model. As noted by Pek (2012), “[i]f the size of parameter values change radically under a minute perturbation to model fit, such parameter values provide a weak basis for scientific conclusions” (p. 2; see also Green, 1977). Despite cautionary notes of this type, researchers frequently offer coefficient interpretations that rely on the model coefficient's relative magnitudes without conducting such a parameter sensitivity analysis (e.g., Adwere-Boamah, 2009; Goldman, 2001; Menard, 2004). In this paper, we have derived and provided two algorithms for assessing parameter sensitivity for logistic regression. Moreover, we have shown how these algorithms can be used to lend support to conclusions (e.g., Aspirin Study) as well as call conclusions into question (e.g., Smoking Study). Using one of our algorithms to assess parameter sensitivity adds additional information about the model above and beyond standard model evaluative techniques (e.g., confidence intervals), and thus can bolster the model interpretations so desired by applied researchers.

References

- Adwere-Boamah, J. (2009). Multiple logistic regression analysis of cigarette use among high school students. *Journal of Case Studies in Education*.
- Agresti, A. (2007). *An introduction to categorical data analysis*. Wiley-Interscience.
- Baker, F. B. (1992). *Item response theory: Parameter estimation techniques*. New York, NY: Marcel Dekker, Inc.
- Cohen, J. (1994). The earth is round ($p < .05$). *American Psychologist*, 49, 997–1003.
- Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/ correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Dana, J., & Dawes, R. M. (2004). The superiority of simple alternatives to regression for social science predictions. *Journal of Educational and Behavior Statistics*, 29, 317–331.
- Dawson, R. (1995). The unusual episode data revisited. *Journal of Statistics Education*, 3.
- Ferguson, C. C. (1979). Intersections of ellipsoids and planes of arbitrary orientation and position. *Mathematical Geology*, 11, 329–336.
- Goldberger, A. S. (1968). *Topics in regression analysis*. New York: Macmillan.
- Goldman, B. M. (2006). Toward an understanding of employment discrimination claiming: An integration of organizational justice and social information processing theories. *Personnel Psychology*, 54, 361–386.
- Green, B. F. (1977). Parameter sensitivity in multivariate methods. *Multivariate Behavioral Research*, 12, 263–287.
- Greene, W. H. (2003). *Econometric analysis* (Fifth Ed.). Upper Saddle River,

NJ: Prentice-Hall.

Hadi, A. S. (1996). *Matrix algebra as a tool*. Belmont, CA: Duxbury Press.

Jennrich, R. I., & Sampson, P. F. (1976). Newton-Raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, 18, 11-17.

Jones, J. A. (2010). GenCorr: An R routine to generate correlation matrices from a user-defined eigenvalue structure. *Applied Psychological Measurement*, 34, 68-69.

Jones, J. A. & Waller, N. G. (in press). Computing confidence intervals for standardized regression coefficients. *Psychological Methods*.

Kelley, K. & Maxwell, S. E. (2003). Sample size for multiple regression: obtaining regression coefficients that are accurate, not simply significant. *Psychological Methods*, 8, 305-321.

Khuri, A. I. (2003). *Advanced calculus with applications in statistics*. Wiley-Interscience.

Koopman, R. F. (1988). On the sensitivity of a composite to its weights. *Psychometrika*, 53, 547-552.

Lazardis, A. (2007). A note regarding the condition number: The case of spurious and latent multicollinearity. *Quality and Quantity*, 41, 123-135.

MacCallum R. C., Browne, M. W., & Lee, T. (2009). *Fungible parameter estimates in structural equation modeling*. Paper presented at the Annual Meeting of the Psychometric Society. Salishan Resort, Oregon.

MacCallum, R. C., Lee, T., & Browne, M. W. (2012). Fungible parameter estimates in latent curve models. In M. C. Edwards, & R. C. MacCallum (Eds.), *Current issues in the theory and application of latent variable models* (pp. 183-197). Routledge Academic.

Marsaglia, G. & Olkin, I. (1984). Generating correlation matrices. *SIAM Jour-*

nal on Scientific and Statistical Computing, 5, 470–475.

Menard, S. (2000). Coefficients of determination for multiple logistic regression analysis. *The American Statistician*, 54(1), 17–24.

Menard, S. (2001). *Applied logistic regression analysis*. Sage Publications, Incorporated.

Menard, S. (2004). Six approaches to calculating standardized logistic regression coefficients. *The American Statistician*, 58(3), 218–223.

Pek, J (2012). *Fungible parameter contours and confidence regions in structural equation models*. (Unpublished doctoral dissertation). University of North Carolina at Chapel Hill.

Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *The Journal of Machine Learning Research*, 4, 211–255.

Pregibon, D. (1981). Logistic regression diagnostics. *The Annals of Statistics*, 9, 705–724.

R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.

Reise, S. P. & Widaman, K. F. (1999). Assessing the fit of measurement models at the individual level: A comparison of item response theory and covariance structure approaches. *Psychological Methods*, 4, 3–21.

Rencher, A. C. (2008). *Linear models in statistics* (2nd ed.). Hoboken, NJ: John Wiley and Sons, Inc.

Rosenthal, R. (1990). How are we doing in soft psychology? *American Psychologist*, 45(6), 775–777.

Rozeboom, W. W. (1979). Sensitivity of a linear composite of predictor items to differential item weighting. *Psychometrika*, 44, 289–296.

- Steering Committee of the Physicians Health Study Research Group (1988). Preliminary report: Findings from the aspirin component of the ongoing physicians' health study. *The New England Journal of Medicine*, 318, 262–264.
- Stein, S. & Barcellos, A. (1992). *Calculus and analytic geometry* (5th ed.). McGraw-Hill.
- Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49, 1225–1231.
- Venables, W. N. & Ripley, B. D. (2002) *Modern applied statistics with S* (4th ed.). Springer, New York. ISBN 0-387-95457-0.
- Wainer, H. (1976). Estimating coefficients in linear models: It don't make no nevermind. *Psychological Bulletin*, 83, 213–217.
- Wainer, H. (1978). On the sensitivity of regression and regressors. *Psychological Bulletin*, 85, 267–273.
- Waller, N. G. (2008). Fungible weights in multiple regression. *Psychometrika*, 73, 691–703.
- Waller, N. G. (2011). The geometry of enhancement in multiple regression. *Psychometrika*, 76, 634–649.
- Waller, N. G. (2013). Fungible correlation matrices: A new tool for evaluating penalized regression models. Unpublished Manuscript.
- Waller, N. G. & Jones, J. A. (2009). Locating the extrema of fungible regression weights. *Psychometrika*, 74, 589–602.
- Waller, N. & Jones, J. (2010). Correlation weights in multiple regression. *Psychometrika*, 75(1), 58–69.
- Waller, N. & Jones, J. (2011). Investigating the performance of alternate regression weights by studying all possible criteria in regression models with a fixed set of predictors. *Psychometrika*, 76(3), 410–439.

Weiss, D. J. & Kingsbury, G. (1984). Application of computerized adaptive testing to educational problems. *Journal of Educational Measurement*, 21(4), 361–375.

Wilks, S. S. (1938). Weighting schemes for linear functions of correlated variables when there is no dependent variable. *Psychometrika*, 3, 23–40.

Yuan, K. H. & Chan, W. (2011). Biases and standard errors of standardized regression coefficients. *Psychometrika*, 76, 670–690.

Appendix A: Figures from All Simulation Conditions

Figure 30: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta \|\mathbf{v}_1$ for the Ellipsoid Method

Figure 31: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 32: Ellipsoid Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 33: Ellipsoid Method: $R_{\beta}^2 = .20$; $R_{xx} = 1$ Factor; $p = 3$; $\beta \|v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $R_{xx} = 1$ Factor, $p = 3$, and $\beta \|v_3$ for the Ellipsoid Method.

Figure 34: Log-Likelihood Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 35: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 36: Log-Likelihood Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 360$ for the Log-Likelihood Method.

Figure 37: Log-Likelihood Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \|\mathbf{v}_3$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta \|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 38: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 39: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 40: Ellipsoid Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 41: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta \|\mathbf{v}_3\|$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta \|\mathbf{v}_3\|$ for the Ellipsoid Method.

Figure 42: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 43: Log-Likelihood Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 44: Log-Likelihood Method: $R_\beta^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_\beta^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 360$ for the Log-Likelihood Method.

Figure 45: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 46: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 47: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 48: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 49: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Ellipsoid Method.

Figure 50: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 51: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 52: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 53: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 3$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 3$, and $\beta||v_3$ for the Log-Likelihood Method.

Figure 54: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 55: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 56: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 57: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Ellipsoid Method.

Figure 58: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 59: Log-Likelihood Method: $R^2_{\beta} = .20$; $\kappa = 9$; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\kappa = 9$, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 60: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 61: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 3$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 3$, and $\beta||v_3$ for the Log-Likelihood Method.

Figure 62: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta \|\mathbf{v}_1$ for the Ellipsoid Method

Figure 63: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 64: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 65: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta||\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta||\mathbf{v}_3$ for the Ellipsoid Method.

Figure 66: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 67: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 68: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 360$ for the Log-Likelihood Method.

Figure 69: Log-Likelihood Method: $R_\beta^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 3$; $\beta \|\mathbf{v}_3$

Graphical summary of the simulation condition $R_\beta^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 3$, and $\beta \|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 70: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 71: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 72: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 73: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Ellipsoid Method.

Figure 74: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 75: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 76: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 360$ for the Log-Likelihood Method.

Figure 77: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 78: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 79: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 80: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 81: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Ellipsoid Method.

Figure 82: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 83: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 84: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 85: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 3$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 3$, and $\beta||v_3$ for the Log-Likelihood Method.

Figure 86: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 87: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 88: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\angle(\beta, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 89: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Ellipsoid Method.

Figure 90: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 91: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\angle(\beta, v_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 92: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 93: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 3$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 3$, and $\beta\|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 94: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 95: Ellipsoid Method: $R_{\beta}^2 = .20$; $R_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $R_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 96: Ellipsoid Method: $R_{\beta}^2 = .20$; $R_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $R_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 97: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta||\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta||\mathbf{v}_3$ for the Ellipsoid Method.

Figure 98: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta||\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta||\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 99: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 100: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 360^\circ$ for the Log-Likelihood Method.

Figure 101: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta\|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 102: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta||\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta||\mathbf{v}_1$ for the Ellipsoid Method

Figure 103: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 104: Ellipsoid Method: $R_{\beta}^2 = .20$; $R_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $R_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 105: Ellipsoid Method: $R_{\beta}^2 = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta||\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta||\mathbf{v}_3$ for the Ellipsoid Method.

Figure 106: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 107: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 108: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 360^\circ$ for the Log-Likelihood Method.

Figure 109: Log-Likelihood Method: $R^2_{\beta} = .20$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta\|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 110: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\beta||v_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 5$, and $\beta||v_1$ for the Ellipsoid Method

Figure 111: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 112: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 113: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 5$, and $\beta||v_3$ for the Ellipsoid Method.

Figure 114: Log-Likelihood Method: $R^2_{\beta} = .20$; $\kappa = 3$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\kappa = 3$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 115: Log-Likelihood Method: $R^2_{\beta} = .20$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 116: Log-Likelihood Method: $R^2_{\beta} = .20$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 117: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 3$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 3$, $p = 5$, and $\beta||v_3$ for the Log-Likelihood Method.

Figure 118: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 119: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 120: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 121: Ellipsoid Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 5$, and $\beta||v_3$ for the Ellipsoid Method.

Figure 122: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 123: Log-Likelihood Method: $R_{\beta}^2 = .20$; $\kappa = 9$; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .20$, $\kappa = 9$, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 124: Log-Likelihood Method: $R^2_{\beta} = .20$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 125: Log-Likelihood Method: $R^2_{\beta} = .20$; $\kappa = 9$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R^2_{\beta} = .20$, $\kappa = 9$, $p = 5$, and $\beta||v_3$ for the Log-Likelihood Method.

Figure 126: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta||\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta||\mathbf{v}_1$ for the Ellipsoid Method

Figure 127: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 128: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 129: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta||\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta||\mathbf{v}_3$ for the Ellipsoid Method.

Figure 130: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 131: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 132: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 360$ for the Log-Likelihood Method.

Figure 133: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 1$ Factor; $p = 5$; $\beta||\mathbf{v}_3$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 1$ Factor, $p = 5$, and $\beta||\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 134: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta||\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta||\mathbf{v}_1$ for the Ellipsoid Method

Figure 135: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\boldsymbol{\beta}, \mathbf{v}_1) = 30^\circ$ for the Ellipsoid Method.

Figure 136: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 60^\circ$ for the Ellipsoid Method.

Figure 137: Ellipsoid Method: $R_{\beta}^2 = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta||\mathbf{v}_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta||\mathbf{v}_3$ for the Ellipsoid Method.

Figure 138: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 139: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 140: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\angle(\beta, \mathbf{v}_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\angle(\beta, \mathbf{v}_1) = 360$ for the Log-Likelihood Method.

Figure 141: Log-Likelihood Method: $R^2_{\beta} = .80$; $\mathbf{R}_{xx} = 2$ Factor; $p = 5$; $\beta\|\mathbf{v}_3$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\mathbf{R}_{xx} = 2$ Factor, $p = 5$, and $\beta\|\mathbf{v}_3$ for the Log-Likelihood Method.

Figure 142: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 143: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 144: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 145: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 3$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 3$, $p = 5$, and $\beta||v_3$ for the Ellipsoid Method.

Figure 146: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 3$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 3$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 147: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 148: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 3$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 3$, $p = 5$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 149: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 3$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 3$, $p = 5$, and $\beta||v_3$ for the Log-Likelihood Method.

Figure 150: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Ellipsoid Method

Figure 151: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Ellipsoid Method.

Figure 152: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 60^\circ$ for the Ellipsoid Method.

Figure 153: Ellipsoid Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 5$, and $\beta||v_3$ for the Ellipsoid Method.

Figure 154: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 9$; $p = 5$; $\beta\|\mathbf{v}_1$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 9$, $p = 5$, and $\beta\|\mathbf{v}_1$ for the Log-Likelihood Method.

Figure 155: Log-Likelihood Method: $R_{\beta}^2 = .80$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 30^\circ$

Graphical summary of the simulation condition $R_{\beta}^2 = .80$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 30^\circ$ for the Log-Likelihood Method.

Figure 156: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 9$; $p = 5$; $\angle(\beta, v_1) = 60^\circ$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 9$, $p = 5$, and $\angle(\beta, v_1) = 360$ for the Log-Likelihood Method.

Figure 157: Log-Likelihood Method: $R^2_{\beta} = .80$; $\kappa = 9$; $p = 5$; $\beta||v_3$

Graphical summary of the simulation condition $R^2_{\beta} = .80$, $\kappa = 9$, $p = 5$, and $\beta||v_3$ for the Log-Likelihood Method.

Appendix B: R Code for Ellipsoid Method

In this appendix we report an R script that computes fungible weights for logistic regression using the Ellipsoid Method described in Section 3.2.

```
#####
# This function computes fungible weights for unstandardized      #
# regression coefficients from a logistic regression.              #
#                                                                    #
# Arguments:                                                        #
# b          - Regression coefficients from logistic regression    #
# Sxx        - A p x p matrix of predictor covariances           #
# sxy        - A p x 1 vector of predictor-criterion covariances  #
# Xmn        - A p x 1 vector of predictor means                 #
# ymn        - Criterion mean                                     #
# cor        - Desired correlation between the composite created  #
#               from the logistic regression coefficients and the  #
#               composite created from fungible weights           #
# numfun     - Number of fungible weight vectors to generate     #
#                                                                    #
# Output:                                                          #
# coefs      - Matrix of fungible weights                         #
#####

funCov <- function(b, Sxx, sxy, Xmn=0, ymn=0, cor, numfun) {

# norms the columns of a matrix
norm <- function(x) {
  x <- as.matrix(x)
  xx <- t(x)%*%x
  x%*%diag(diag(xx)^-.5,nrow=nrow(xx),ncol=ncol(xx))
}

p <- ncol(Sxx)

UDU <- eigen(Sxx)

U <- UDU$vector
D <- diag(UDU$val)
Dhalf <- sqrt(D)
Dinvhalf <- diag(1/diag(Dhalf))
```

```

if(length(b) == ncol(Sxx)+1) b <- b[-1]

g <- sqrt(D)%*%t(U)%*%b/as.numeric(sqrt(t(b)%*%Sxx)%*%b))

intercept <- matrix(0,numfun,1); colnames(intercept) <- "a0"
a <- matrix(0,numfun,p)
colnames(a) <- colnames(a,do.NULL=FALSE)
for(j in 1:p) colnames(a)[j] <- paste("a",j,sep='')

# Generate fungible weights

for(i in 1:numfun) {

  tmp <- matrix(c(g,rnorm(p*(p-1))),p,p)
  Q <- qr.Q(qr(tmp))[, -1]

  v <- norm(rnorm((p-1)))

  hn <- cor*g + sqrt(1-cor^2)*0%*%v

  atil <- U%*%Dinvhalf%*%hn

  s <- as.numeric(t(sxy)%*%atil)
  s.sign <- sign(s)

  a[i,] <- s.sign*s*atil

  intercept[i] <- ymn - Xmn%*%a[i,]
  names(intercept) <- "a0"

} # End loop to generate fungible weights

if(all(Xmn == 0) && ymn == 0) return(coefs <- a)
else return(coefs <- cbind(intercept,a))

} # End funCov function
#####
##### Example #####
#####

set.seed(668)
X <- cbind(rnorm(100),rnorm(100),rnorm(100))
ystar <- rnorm(100)
y <- rep(0,100); y[ystar>0] <- 1

```

```

m <- glm(y ~ X, family='binomial')

Rsq <- 1 - m$dev/m$null

b <- coef(m)

log.odds <- log(m$fitted/(1-m$fitted))

Sxx <- cov(X); sxy <- cov(X, log.odds)
Xmn <- apply(X,2,mean); ymn <- mean(m$fitted)

numfun <- 1000
r <- .99

out <- funCov(b=b, Sxx=Sxx, sxy=sxy, Xmn=Xmn,
              ymn=ymn, cor=r, numfun=numfun)

# Output
> cor(X %*% b[-1], X %*% out[sample(1,1:numfun),2:4])
      [,1]
[1,] 0.99

```

Appendix C: R Code for Log-Likelihood Method

In this appendix we report an R script that calls the optimization algorithm “newton” written in C for computing fungible regression weights for logistic regression. The R code is presented first, followed by the C code.

```
#####  
# Example R script using the compiled C code "newton" #  
# to compute fungible weights for logistic regression #  
#####  
  
require(MASS); require(fmsb)  
  
# Computes the log-likelihood value for a suboptimal  
# value of Rsq based on the likelihood-ratio pseudo-Rsq  
reducedLL <- function(llNULL, Rsq, N) {  
  llNULL - N*log(1-Rsq*(1-exp(2*llNULL/N)))/2  
}  
  
R <- matrix(.5,3,3); diag(R) <- 1  
b <- eigen(R)$ve[,1]  
olsRsq <- .5  
b <- b*sqrt(olsRsq/as.numeric(t(b)%*%R%*%b))  
  
set.seed(667)  
N <- 500  
X <- mvrnorm(N,mu=rep(0,3),Sigma=R,emp=T)  
e <- lm(rlogis(N)^X)$res  
  
vyh <- olsRsq*var(e)/(1-olsRsq)  
b <- b * sqrt(vyh/olsRsq)  
  
ystar <- X%*%b + e  
y <- rep(0,N); y[ystar > 0] <- 1  
  
out <- summary(m <- glm(y ~ X, family="binomial"))  
  
llNull <- -out$null/2  
Rsq <- NagelkerkeR2(m)$R2  
  
# compute suboptimal likelihood value by lowering the  
# likelihood-ratio pseudo-Rsq by .01  
subLL <- reducedLL(llNull, Rsq-.01, N)
```

```

rm(.Random.seed,pos=.GlobalEnv)
set.seed(666)
rstarts <- mvrnorm(1000,coef(m),Sigma=diag(2,length(coef(m)))

dyn.load("FLR_NEWTON.so")

# compute fungible weights for logistic regression coefficients
bs <- .Call("newton", cbind(1,X), y, 1e-8, rstarts, subLL)

dyn.unload("FLR_NEWTON.so")

#####
##### Begin C Code #####
#####

#include <R.h>
#include <Rinternals.h>
#include <math.h>
#include <Rmath.h>

/* Function that gets the dimensions of matrices from R*/
#define getDims(A) INTEGER(coerceVector(getAttrib(A,R_DimSymbol),INTSXP))

/* Function that implements the Newton-Method to compute fungible */
/* fungible regression weights for logistic regression. */
SEXP newton(SEXP X, SEXP y, SEXP tol, SEXP rstarts, SEXP val) {

    int obs, npars, nflr;

    obs = getDims(X)[0];
    npars = getDims(X)[1];
    nflr = getDims(rstarts)[0];

    int const np=npars;
    double *py, *pX, *pval, *ptol, *prstarts;

    PROTECT(y = coerceVector(y, REALSXP));
    PROTECT(X = coerceVector(X, REALSXP));
    PROTECT(tol = coerceVector(tol, REALSXP));
    PROTECT(rstarts = coerceVector(rstarts, REALSXP));
    PROTECT(val = coerceVector(val, REALSXP));

    SEXP bout;
    double *pbout;

```



```

py = REAL(y); pX = REAL(X); ptol = REAL(tol); prstarts = REAL(rstarts);
pval = REAL(val);

PROTECT(bout = allocMatrix(REALSXP, nflr, npars));
pbout = REAL(bout);

double TOL=99.0, bstart[np], g[np], dll[np], part, p, dp[np];
double ginv[np], gtmp, bn[np], ll;

int i, j, k, l, m, n, o, q, r, s, w, z, breakflag;

/* Loop that computes N fungible weight vectors */
for(z = 0; z < nflr; z++) {

    breakflag = 0;
    TOL = 99.0;

    /* Grab the start values */
    for(i = 0; i < npars; i++) bstart[i] = prstarts[z + nflr * i];

    /* Newton Method for finding the root */
    while(TOL > *ptol) {

        for(s = 0; s < npars; s++) dll[s] = 0.0;

        ll = 0;

        /* Compute the log-likelihood and the gradient of log-likelihood*/
        for(j = 0; j < obs; j++){

            part = 1.0;

            for(k = 0; k < npars; k++) part *= exp(-pX[j+obs*k]*bstart[k]);
            for(l = 0; l < npars; l++) {
                dp[l] = (part/((1+part)*(1+part)))*pX[j + obs*l];
            }

            p = 1/(1 + part);

            ll += py[j] * log(p) + (1 - py[j]) * log(1 - p);

            for(q = 0; q < npars; q++) {
                dll[q] += py[j]*dp[q]*(1/p) - (1 - py[j])*dp[q]*(1/(1 - p));
            }
        }
    }
}

```

```

    }

    if(isnan(dll[0])) {
        breakflag = 1;
        break;
    }

    ll = ll - *pval;

    /* Compute generalized inverse of the gradient */
    gtmp = 0.0;
    for(m = 0; m < npars; m++) gtmp += dll[m] * dll[m];
    for(n = 0; n < npars; n++) ginv[n] = (1/gtmp) * dll[n];

    /* Compute the approximate root */
    for(r = 0; r < npars; r++) bn[r] = bstart[r] - ginv[r] * ll;
    for(r = 0; r < npars; r++) bstart[r] = bn[r];
    TOL = fabs(ll);
}

if(breakflag == 1) for(w = 0; w < npars; w++) pbout[z+nflr*w] = NAN;
else for(w = 0; w < npars; w++) pbout[z + nflr*w] = bn[w];

}

UNPROTECT(6);
return(bout);
}

```