Assignment 2 - QBS 177

Code

Spencer Bertsch
Jan. 2022

Some helpful commands:
* Insert a new code chunk: *Cmd+Option+I*
* Run a code cell: *Cmd+Shift+Enter*
* Preview the notebook HTML file: *Cmd+Shift+K*

Imports

Hide

```
library(glue)
```

First we need to change our current working directory to the correct directory for assignment 2

Hide

```
setwd('/Users/spencerbertsch/Desktop/dev/phd_courses/MATH177/assignment2')
getwd()
```

```
[1] "/Users/spencerbertsch/Desktop/dev/phd_courses/MATH177/assignment2"
```

Now we can load our data into memory

Hide

```
load('dbp.Rdata')
ls()
```

```
 [1] "ci"             "dbp"            "dev.adj"        "dev.allelic"    "dev.genotypic"
"lrt.pvalue"     "result.adj"
 [8] "result.allelic" "result.inter"   "result.snp12"   "snp.beta"       "snp.data"
```

Hide

```
dbp[1:5,]
```

	fam <int>	pid <int>	fid <int>	mid <int>	sex <fctr>	trait <dbl>	affection <fctr>	age <int>	rs1101 <fctr>
4928	4928	1	0	0	1	85.51	1	66	4
1838	1838	1	0	0	1	84.51	1	67	2
2450	2450	1	0	0	1	84.30	1	89	2

	fam <int>	pid <int>	fid <int>	mid <int>	sex <fctr>	trait <dbl>	affection <fctr>	age <int>	rs1101 <fctr>
647	647	1	0	0	2	89.14	1	36	4
2772	2772	1	0	0	1	90.39	1	54	3

5 rows | 1-10 of 28 columns

Question 1

Logistic regression on a single SNP genotype

Hide

```
result.snp12 = glm (affection ~ rs1112, family=binomial("logit"), data=dbp)

print(result.snp12)
```

```
Call:  glm(formula = affection ~ rs1112, family = binomial("logit"),
    data = dbp)

Coefficients:
(Intercept)      rs11123      rs11124
    -0.4449       0.7582       1.5435

Degrees of Freedom: 599 Total (i.e. Null);  597 Residual
Null Deviance:      831.8
Residual Deviance: 797.7    AIC: 803.7
```

From assignment: “The coefficients table lists the estimated values for the regression coefficients β as well as their standard errors. It further contains the P-values as obtained from a Wald test.”

Hide

```
print ( summary(result.snp12) )
```

```
Call:
glm(formula = affection ~ rs1112, family = binomial("logit"),
    data = dbp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6651  -0.9952  -0.1183   1.0476   1.3712

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4449     0.1189  -3.741 0.000183 ***
rs11123       0.7582     0.1746   4.343 1.40e-05 ***
rs11124       1.5435     0.3416   4.518 6.24e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 797.75  on 597  degrees of freedom
AIC: 803.75

Number of Fisher Scoring iterations: 4
```

Hide

```
# calculate the chi squared statistic for the trained logistic regression model
dev.genotypic = anova (result.snp12, test="Chi")
print('Here we can see the results of the ANOVA:')
```

```
[1] "Here we can see the results of the ANOVA:"
```

Hide

```
print(dev.genotypic)
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: affection

Terms added sequentially (first to last)

       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     599     831.78
rs1112  2    34.03       597     797.75 4.078e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
lrt.pvalue = pchisq(dev.genotypic[dim(dev.genotypic)[1],"Deviance"], df=2, ncp=0, FALSE)
print (glue('And finally the P-value for the Likelihood Ratio Test for our Factor predic
tor {lrt.pvalue}'))
```

```
And finally the P-value for the Likelihood Ratio Test for our Factor predictor 4.0778562
4557897e-08
```

Hide

```
print ('Here we can see the summary of the logistic regression model')
```

```
[1] "Here we can see the summary of the logistic regression model"
```

Hide

```
print (summary(result.snp12)$coefficients)
```

```
             Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -0.4449068  0.1189351 -3.740754 1.834691e-04
rs11123      0.7582015  0.1745740  4.343154 1.404519e-05
rs11124      1.5435191  0.3416277  4.518132 6.238747e-06
```

Here we can calculate the odds ratios for the two dummy variables that were generated when we modeled the rs1112 factor variable. We remember from class that we can obtain the odds ratio by simply raising *e* to the power of the beta value to get the odds ratio for that beta.

Hide

```
snp.beta = summary(result.snp12)$coefficients[2:3,1]

print(glue('Beta value for dummy variable rs1113: {round(snp.beta[1], 6)}'))
```

```
Beta value for dummy variable rs1113: 0.758201
```

Hide

```
print(glue('Odds Ratio for dummy variable rs1113: {round(exp(snp.beta[1]), 6)}'))
```

```
Odds Ratio for dummy variable rs1113: 2.134434
```

Hide

```
print(glue('Beta value for dummy variable rs1114: {round(snp.beta[2], 6)}'))
```

```
Beta value for dummy variable rs1114: 1.543519
```

Hide

```
print(glue('Odds Ratio for dummy variable rs1114: {round(exp(snp.beta[2]), 6)}'))
```

```
Odds Ratio for dummy variable rs1114: 4.681034
```

We can also find the the 95% confidence interval for the odds ratios calculated above!

Hide

```
ci = confint (result.snp12)
```

```
Waiting for profiling to be done...
```

Hide

```
print (ci)
```

```
                2.5 %     97.5 %
(Intercept) -0.6802726 -0.2135169
rs11123      0.4176220  1.1023701
rs11124      0.8984800  2.2475097
```

Hide

```
print ( exp(ci) )
```

```
               2.5 %    97.5 %
(Intercept) 0.5064789 0.8077385
rs11123     1.5183466 3.0112947
rs11124     2.4558674 9.4641382
```

Question 1 Part 2

Using a Numeric Predictor Variable

In order to run an Allelic (Multiplicative) model, we need to change the data type of our predictor variable from ‘factor’ to numeric. We do that below and run our analysis of the resulting logistic regression model again.

Hide

```
snp.data = dbp[,c("affection", "rs1112")]
print('Summary while the predictor variable (rs1112) is a factor variable;')
```

```
[1] "Summary while the predictor variable (rs1112) is a factor variable;"
```

Hide

```
summary(snp.data)
```

```
 affection rs1112
 0:300     2:297
 1:300     3:251
           4: 52
```

Hide

```
snp.data[,"rs1112"] <- as.numeric(snp.data[,"rs1112"]) - 1
print('Summary after the predictor variable (rs1112) was changed to a numeric variable:'
)
```

```
[1] "Summary after the predictor variable (rs1112) was changed to a numeric variable:"
```

Hide

```
summary(snp.data)
```

```
 affection     rs1112
 0:300     Min.   :0.0000
 1:300     1st Qu.:0.0000
           Median :1.0000
           Mean   :0.5917
           3rd Qu.:1.0000
           Max.   :2.0000
```

Now that our predictor variable rs1112 is numeric, we need to re-train the logistic regression model on the new (numeric) predictor variable. From assignment: “The coefficients table lists the estimated values for the regression coefficients β as well as their standard errors. It further contains the P-values as obtained from a Wald test.”

Hide

```
#retrain the logistic regression model
result.allelic = glm (affection ~ rs1112, family=binomial("logit"), data=snp.data)
summary(result.allelic)
```

```
Call:
glm(formula = affection ~ rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6582  -0.9944  -0.1154   1.0456   1.3722

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4470     0.1142  -3.913 9.10e-05 ***
rs1112        0.7652     0.1356   5.642 1.68e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 797.75  on 598  degrees of freedom
AIC: 801.75

Number of Fisher Scoring iterations: 4
```

Use Anova on the trained logistic regression model to find the Chi squared statistic

Hide

```
dev.allelic = anova (result.allelic, test="Chi")
print(dev.allelic)
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: affection

Terms added sequentially (first to last)

       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     599     831.78
rs1112  1   34.026       598     797.75 5.438e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lastly we can find the p-value for the likelihod ratio test on the logistic regression model trained on the numeric predictor variable.

Hide

```
lrt.pvalue = pchisq(dev.allelic[dim(dev.allelic)[1],"Deviance"], df=2, ncp=0, FALSE)
print (glue('And lastly the p-value for the Likelihood Ratio Test for our Numeric predic
tor {lrt.pvalue}'))
```

```
And lastly the p-value for the Likelihood Ratio Test for our Numeric predictor 4.0861114
1250953e-08
```

Question 2

Adjustment for the effects of covariates and of other SNPs

For this question we will be using more than a single predictor variable. In order to make our lives easier we can create a subset of the initial dataframe that contains only the columns that we need for our analysis.

Hide

```
snp.data = dbp[,c("affection", "trait","sex", "age", "rs1112", "rs1117")]
summary(snp.data)
```

```
 affection     trait          sex          age         rs1112  rs1117
 0:300     Min.   : 60.50   1:329   Min.   :18.00   2:297   2:396
 1:300     1st Qu.: 77.44   2:271   1st Qu.:38.00   3:251   3:190
           Median : 82.00           Median :55.00   4: 52   4: 14
           Mean   : 81.85           Mean   :55.49
           3rd Qu.: 86.09           3rd Qu.:74.00
           Max.   :101.49           Max.   :90.00
```

Hide

```
# here we remove 1 from each newly numeric column in the dataframe so that our values ca
n range between [0, 1, 2] instead of [1, 2, 3]
snp.data[,"rs1112"] <- as.numeric(snp.data[,"rs1112"]) - 1
snp.data[,"rs1117"] <- as.numeric(snp.data[,"rs1117"]) - 1
```

Hide

```
result.adj = glm (affection ~ age + rs1112, family=binomial("logit"), data=snp.data)
summary(result.adj)
```

```
Call:
glm(formula = affection ~ age + rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6776  -1.0066  -0.1132   1.0550   1.3937

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.520422   0.250956  -2.074   0.0381 *
age          0.001322   0.004020   0.329   0.7423
rs1112       0.765189   0.135624   5.642 1.68e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 797.64  on 597  degrees of freedom
AIC: 803.64

Number of Fisher Scoring iterations: 4
```

Hide

```
result.adj = glm (affection ~ sex + rs1112, family=binomial("logit"), data=snp.data)
summary(result.adj)
```

```
Call:
glm(formula = affection ~ sex + rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.82645  -1.12415  -0.09007   1.21323   1.57462

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.08386    0.13730  -0.611    0.541
sex2        -0.81412    0.17253  -4.719 2.37e-06 ***
rs1112       0.77139    0.13840   5.574 2.49e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 774.98  on 597  degrees of freedom
AIC: 780.98

Number of Fisher Scoring iterations: 4
```

Hide

```
result.adj = glm (affection ~ sex + age + rs1112, family=binomial("logit"),
                  data=snp.data)
summary(result.adj)
```

```
Call:
glm(formula = affection ~ sex + age + rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.84985  -1.12493  -0.08714   1.19367   1.60989

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.198133   0.263732  -0.751    0.452
sex2        -0.817603   0.172736  -4.733 2.21e-06 ***
age          0.002084   0.004105   0.508    0.612
rs1112       0.771546   0.138411   5.574 2.48e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 774.72  on 596  degrees of freedom
AIC: 782.72

Number of Fisher Scoring iterations: 4
```

We can now adjust for the effects of other SNPs in the model! Here we are adjusting for the SNP rs1117 to determine whether or not it influences the model predictions.

Hide

```
result.adj = glm (affection ~ rs1117 + rs1112, family=binomial("logit"), data=snp.data)
summary(result.adj)
```

```
Call:
glm(formula = affection ~ rs1117 + rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7636  -0.9923  -0.1518   1.1154   1.3745

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4523     0.1144  -3.955 7.66e-05 ***
rs1117        0.2853     0.2297   1.242  0.21431
rs1112        0.5999     0.1883   3.186  0.00144 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 796.21  on 597  degrees of freedom
AIC: 802.21

Number of Fisher Scoring iterations: 4
```

Hide

```
dev.adj = anova (result.adj, test="Chi")
print(dev.adj)
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: affection

Terms added sequentially (first to last)

       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     599     831.78
rs1117  1   25.064       598     806.71 5.547e-07 ***
rs1112  1   10.501       597     796.21  0.001193 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

And we can find the p-value for the likelihod ratio test on the logistic regression model adjusted for the effect of the other SNP: rs1117.

Hide

```
lrt.pvalue = pchisq(dev.adj[dim(dev.adj)[1],"Deviance"], df=2, ncp=0, FALSE)
print (glue('And lastly the p-value for the Likelihood Ratio Test for the logistic regre
ssion model adjusted for 1117 is {lrt.pvalue}'))
```

```
And lastly the p-value for the Likelihood Ratio Test for the logistic regression model a
djusted for 1117 is 0.00524538266873438
```

Question 3

Analysis of quantitative instead of dichotomized trait

Here we leave logistic regression behind and swap out our binned (binary) response variable for a continuous one. The binary response variable that we have been predicting is just a binned version of a continuous variable called 'trait'. We will now fit linear regression models to predict the 'trait' variable using the **lm** function in R. The trait here is the patient's blood pressure and the previous bucketed, binary response variable *affection* was 1 if the patient had abnormal blood pressure, and 0 if the blood pressure was considered normal.

Here we can run a regression model with a single predictor variable: SNP = rs1112.

Hide

```
result.adj = lm (trait ~ rs1112, data=snp.data)
summary(result.adj)
```

```

Call:
lm(formula = trait ~ rs1112, data = snp.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-22.5556  -3.9106   0.2194   4.0144  15.4809 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  80.1021     0.3301 242.680  < 2e-16 ***
rs1112        2.9535     0.3774   7.826 2.29e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.954 on 598 degrees of freedom
Multiple R-squared:  0.09291,   Adjusted R-squared:  0.09139 
F-statistic: 61.25 on 1 and 598 DF,  p-value: 2.292e-14
```

Here we can run a regression model with the rs1112 predictor and also adjust for sex in the model.

Hide

```
result.adj = lm (trait ~ sex + rs1112, data=snp.data)
summary(result.adj)
```

```
Call:
lm(formula = trait ~ sex + rs1112, data = snp.data)

Residuals:
     Min       1Q   Median       3Q      Max
-20.9404  -3.6272   0.2234   3.7815  16.3480

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  81.4542     0.3904 208.654  < 2e-16 ***
sex2         -2.8823     0.4748  -6.071 2.27e-09 ***
rs1112        2.8685     0.3668   7.820 2.41e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.784 on 597 degrees of freedom
Multiple R-squared:  0.1456,    Adjusted R-squared:  0.1428
F-statistic: 50.89 on 2 and 597 DF,  p-value: < 2.2e-16
```

Question 4

Gene-environment (GxE) and gene-gene (GxG) interaction

This is an axciting section in which we get to discover how different variables interact with one another as they are used in a model.

Hide

```
result.inter = glm (affection ~ sex * rs1112, family=binomial("logit"), data=snp.data)
summary(result.inter)
```

```
Call:
glm(formula = affection ~ sex * rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8388  -1.1205  -0.0965   1.2176   1.5685

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.09415    0.15371  -0.613 0.540174
sex2        -0.79026    0.23515  -3.361 0.000777 ***
rs1112       0.79049    0.18896   4.183 2.87e-05 ***
sex2:rs1112 -0.04141    0.27771  -0.149 0.881472
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 774.96  on 596  degrees of freedom
AIC: 782.96

Number of Fisher Scoring iterations: 4
```

Hide

```
result.inter = glm (affection ~ age * rs1112, family=binomial("logit"), data=snp.data)
summary(result.inter)
```

```
Call:
glm(formula = affection ~ age * rs1112, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8044  -1.0479  -0.1256   1.0606   1.4655

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.764365   0.328207  -2.329  0.01986 *
age          0.005719   0.005508   1.038  0.29909
rs1112       1.193715   0.393377   3.035  0.00241 **
age:rs1112  -0.007716   0.006585  -1.172  0.24130
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 796.26  on 596  degrees of freedom
AIC: 804.26

Number of Fisher Scoring iterations: 4
```

Hide

```
result.inter = glm (affection ~ rs1112 * rs1117, family=binomial("logit"), data=snp.dat
a)
summary(result.inter)
```

```
Call:
glm(formula = affection ~ rs1112 * rs1117, family = binomial("logit"),
    data = snp.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7167  -0.9899  -0.1342   1.1126   1.3773

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.45855    0.11749  -3.903  9.5e-05 ***
rs1112         0.61285    0.19612   3.125  0.00178 **
rs1117         0.37232    0.43522   0.855  0.39228
rs1112:rs1117 -0.07464    0.31590  -0.236  0.81323
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.78  on 599  degrees of freedom
Residual deviance: 796.16  on 596  degrees of freedom
AIC: 804.16

Number of Fisher Scoring iterations: 4
```