Exercise 1 SISG Association Mapping

1. Load the LHON.txt data file into your R session.

Can read the data directly from the website if your computer is connected online

```
LHON=read.table("http://faculty.washington.edu/tathornt/sisg/LHON.txt",header=TRUE)
```

If file is saved on your computer, could also read the data in from the directory that contains the file.

```
LHON2=read.table("LHON.txt",header=TRUE)
```

View the first few lines of the LHON data

```
head(LHON)
#   IID GENO   PHENO
# 1 ID1   TT CONTROL
# 2 ID2   CT CONTROL
# 3 ID3   TT    CASE
# 4 ID4   CT CONTROL
# 5 ID5   TT CONTROL
# 6 ID6   TT CONTROL
```

Get information about the types of variables in the LHON data frame

```
str(LHON)
# 'data.frame': 328 obs. of  3 variables:
#  $ IID  : Factor w/ 328 levels "ID1","ID10","ID100",..: 1 112 223 263 274 285 296 307 318 2 ...
#  $ GENO : Factor w/ 3 levels "CC","CT","TT": 3 2 3 2 3 3 1 3 3 3 ...
#  $ PHENO: Factor w/ 2 levels "CASE","CONTROL": 2 2 1 2 2 2 2 2 2 2 ...
```

2. Logistic regression

First create a 0 and 1 phenotype variable indicating Case/Control Status to perform the logistic regression analysis

```
LHON$newpheno=with(LHON,ifelse(PHENO=="CASE",1,0))
```

What would be the reference genotype for a logistic regression analysis? Use the levels command in R. The first factor will be the reference genotype.

```
levels(LHON$GENO)
# [1] "CC" "CT" "TT"
```

2a. Perform the logistic regression analysis from session 4 for this data with CC as the reference genotype.

```
logistmod1=glm(newpheno~GENO,family=binomial(link="logit"),data=LHON)
```

View the summary results of the logistic regression model, including parameter estimates and standard errors

```
summary(logistmod1)
#
# Call:
# glm(formula = newpheno ~ GENO, family = binomial(link = "logit"),
#     data = LHON)
#
# Deviance Residuals:
#     Min       1Q   Median       3Q      Max
# -0.9695  -0.8701  -0.8701   1.5197   2.1093
#
# Coefficients:
#             Estimate Std. Error z value Pr(>|z|)
# (Intercept)  -0.5108     0.5164  -0.989   0.3226
# GENOCT       -1.5994     0.6378  -2.508   0.0122 *
# GENOTT       -0.2654     0.5349  -0.496   0.6197
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 383.49  on 327  degrees of freedom
# Residual deviance: 368.48  on 325  degrees of freedom
# AIC: 374.48
#
# Number of Fisher Scoring iterations: 4
```

2b. Obtain odds ratios and confidence intervals for the CT and TT genotypes.

Can obtain the odds ratio estimates by exponentiating the coefficient estimates from the logistic regression model. What is the odds ratio for the CT genotype?

```
exp(-1.5994)
# [1] 0.2020177
```

Can obtain a confidence interval for the odds ratio parameter for the CT genotype use the standard error of the coefficient estimates from the logistic regression model

```
myse=1.96*(.6378)
CI=c(-1.5994-myse,-1.5994+myse)
exp(CI)
# [1] 0.05787394 0.70517308
```

Similarly can obtain odds ratio estimates and 95% confidence intervals for genotype TT.

```
exp(-.2654)
# [1] 0.7668991
myse=1.96*(.5349)
CI=c(-.2654-myse,-.2654+myse)
exp(CI)
# [1] 0.2687956 2.1880353
```

Alternatively, can obtain the odds ratio estimates and confidence intervale for all paramaters in the logistic regression model by using the *coef()* and *confint.default()* function

```
exp(coef(logistmod1))
# (Intercept)      GENOCT      GENOTT
#   0.6000000   0.2020202   0.7668712
exp(confint.default(logistmod1))
#                  2.5 %   97.5 %
# (Intercept) 0.21806837 1.650858
# GENOCT      0.05787424 0.705187
# GENOTT      0.26878265 2.187981
```

Way too many significant digits to report. Use the *round()* function

```
round(exp(coef(logistmod1)),2)
# (Intercept)      GENOCT      GENOTT
#        0.60        0.20        0.77
round(exp(confint.default(logistmod1)),2)
#             2.5 % 97.5 %
# (Intercept)  0.22   1.65
# GENOCT       0.06   0.71
# GENOTT       0.27   2.19
```

3. Logistic regression with TT as the reference genotype

Use the relevel function to create a new genotype vector with reference genotype TT

```
LHON$NEWGENO=with(LHON,relevel(GENO, ref = "TT"))
levels(LHON$NEWGENO)
# [1] "TT" "CC" "CT"
```

Perform the logistic regression analysis TT as the reference genotype.

```
logistmod2=glm(newpheno~NEWGENO,family=binomial(link="logit"),data=LHON)
```

View the summary results of the logistic regression model, including parameter estimates and standard errors

```
summary(logistmod2)
#
# Call:
# glm(formula = newpheno ~ NEWGENO, family = binomial(link = "logit"),
#     data = LHON)
#
# Deviance Residuals:
#     Min       1Q   Median       3Q      Max
# -0.9695  -0.8701  -0.8701   1.5197   2.1093
#
# Coefficients:
#             Estimate Std. Error z value Pr(>|z|)
# (Intercept)  -0.7763     0.1395  -5.563 2.64e-08 ***
# NEWGENOCC     0.2654     0.5349   0.496 0.619739
# NEWGENOCT    -1.3340     0.3995  -3.339 0.000841 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 383.49  on 327  degrees of freedom
# Residual deviance: 368.48  on 325  degrees of freedom
# AIC: 374.48
#
# Number of Fisher Scoring iterations: 4
```

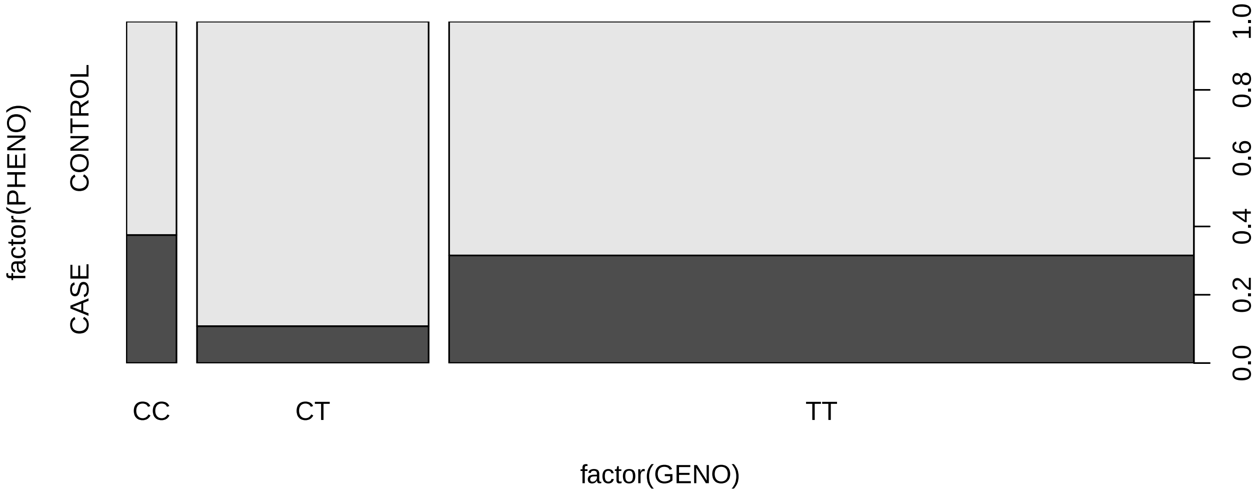
Obtain the odds ratio estimates and confidence intervale for all paramaters in the logistic regression model

```
exp(coef(logistmod2))
# (Intercept)   NEWGENOCC   NEWGENOCT
#   0.4601227   1.3040000   0.2634343
exp(confint.default(logistmod2))
#                 2.5 %    97.5 %
# (Intercept) 0.3500310 0.6048404
# NEWGENOCC   0.4570423 3.7204782
# NEWGENOCT   0.1203945 0.5764187
```

Why are the odds ratios different for CT now?

The reference genotype group has changed from CC to TT, and the TT genotype group is much larger, which increases the precision of the estimate of the effect. Can see this more clearly with a stacked barplot

```
plot(factor(PHENO)~factor(GENO), data=LHON)
```



Can also conduct a logistic regression based on an additive logistic regression model. First create a genotype variable with an additive coding based on the counts of the number of T alleles

```
LHON$genoadd <- with(LHON, 0 + 1*(GENO=="CT") + 2*(GENO=="TT"))
```

Now perform the logistic regression analysis with the additive genotype coding

```
logistmod3 <- glm(newpheno~genoadd,family=binomial(link="logit"),data=LHON)
summary(logistmod3)
#
# Call:
# glm(formula = newpheno ~ genoadd, family = binomial(link = "logit"),
#     data = LHON)
#
# Deviance Residuals:
#     Min       1Q   Median       3Q      Max
# -0.8436  -0.8436  -0.6854   1.5531   1.9797
#
```

```
# Coefficients:
#             Estimate Std. Error z value Pr(>|z|)    
# (Intercept)  -1.8077     0.4554  -3.970  7.2e-05 ***
# genoadd       0.4787     0.2505   1.911   0.0559 .  
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# 
# (Dispersion parameter for binomial family taken to be 1)
# 
#     Null deviance: 383.49  on 327  degrees of freedom
# Residual deviance: 379.47  on 326  degrees of freedom
# AIC: 383.47
# 
# Number of Fisher Scoring iterations: 4
```

Obtain the odds ratio estimates and confidence intervale for all paramaters in the logistic regression model

```
round(exp(coef(logistmod3)),3)
# (Intercept)     genoadd 
#       0.164       1.614 
round(exp(confint.default(logistmod3)),3)
#             2.5 % 97.5 %
# (Intercept) 0.067  0.400
# genoadd     0.988  2.637
```