# Math 564: HW#6

Aleksei Sorokin | A20394300 | asorokin@hawk.iit.edu

## 1) Ex 12.4

```
dfraw = read.table('table12.15.txt',header=T)
dfraw$Failures = dfraw$Attempts-dfraw$Success
head(dfraw)
```

```
##   League Distance Success Attempts Z Failures
## 1    NFL     14.5      68       77 0        9
## 2    NFL     24.5      74       95 0       21
## 3    NFL     34.5      61      113 0       52
## 4    NFL     44.5      38      138 0      100
## 5    NFL     52.0       2       38 0       36
## 6    AFL     14.5      62       67 1        5
```

Create a new dataframe by copying each row Success number of times with MadeIt=1 and copying each row Failure number of times with MadeIt=0.

```
df_success = as.data.frame(lapply(dfraw,rep,dfraw$Success))
df_success$MadeIt = 1
df_failures = as.data.frame(lapply(dfraw,rep,dfraw$Failures))
df_failures$MadeIt = 0
df = rbind(df_success,df_failures)
df = df[,c('MadeIt','Distance','Z')]
df$Distance2 = df$Distance^2
df.nfl = df[df$Z==0,]
df.afl = df[df$Z==1,]
```

### Part a

```
mod.nfl = glm(MadeIt~Distance+Distance2,data=df.nfl,family=binomial(link='logit'))
summary(mod.nfl)
```

```
##
## Call:
## glm(formula = MadeIt ~ Distance + Distance2, family = binomial(link = "logit"),
##     data = df.nfl)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0545  -0.7610   0.5083   0.7068   2.1825
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.490203   1.018619   2.445   0.0145 *
## Distance    -0.013167   0.065990  -0.200   0.8419
## Distance2   -0.001513   0.001008  -1.500   0.1335
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 637.73  on 460  degrees of freedom
## Residual deviance: 491.37  on 458  degrees of freedom
## AIC: 497.37
##
## Number of Fisher Scoring iterations: 4
```

The NFL model is

$$\pi = \frac{\exp(2.49 - 0.0131X - 0.00151X^2)}{1 + \exp(2.49 - 0.0131X - 0.00151X^2)}.$$

```
mod.afl = glm(MadeIt~Distance+Distance2,data=df.afl,family=binomial(link='logit'))
summary(mod.afl)
```

```
##
## Call:
## glm(formula = MadeIt ~ Distance + Distance2, family = binomial(link = "logit"),
##     data = df.afl)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2189  -0.8981   0.4223   0.7822   1.6281
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.892466   1.189215   4.114 3.89e-05 ***
## Distance    -0.197046   0.074346  -2.650  0.00804 **
## Distance2    0.001604   0.001098   1.461  0.14394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 438.59  on 321  degrees of freedom
## Residual deviance: 361.33  on 319  degrees of freedom
## AIC: 367.33
##
## Number of Fisher Scoring iterations: 4
```

The AFL model is

$$\pi = \frac{\exp(4.89 - 0.197X + 0.00160X^2)}{1 + \exp(4.89 - 0.197X + 0.00160X^2)}.$$

**Part b**

```
mod = glm(MadeIt~Distance+Distance2+Z,data=df,family=binomial(link='logit'))
summary(mod)
```

```
##
## Call:
## glm(formula = MadeIt ~ Distance + Distance2 + Z, family = binomial(link = "logit"),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.1534  -0.8425   0.4552   0.7547   1.9566
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.5241844  0.7747793   4.549  5.4e-06 ***
## Distance    -0.0958710  0.0490209  -1.956   0.0505 .
## Distance2   -0.0001086  0.0007365  -0.147   0.8828
## Z            0.1037533  0.1698309   0.611   0.5413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1078.27  on 782  degrees of freedom
## Residual deviance:  858.73  on 779  degrees of freedom
## AIC: 866.73
##
## Number of Fisher Scoring iterations: 4
```

Thus, the combined model is

$$\pi = \frac{\exp(3.52 - 0.0959X - 0.000109X^2 + 0.104Z)}{1 + \exp(3.52 - 0.0959X - 0.000109X^2 + 0.104Z)}.$$

### Part c

No, the $p$-value for the coefficient of $X^2$ is 0.885.

### Part d

Yes, the $Z$ indicator variable is insignificant as indicated by the large $p$-value for the coefficient, 0.541.

## 2) Ex 12.6

```r
df = read.table('table12.6-table12.7.txt',header=T)
df = df[,c('RW','IR','SSPG','CC')]
df$CC = factor(df$CC,levels=c(3,2,1))
```

### Part a

```r
library(nnet)
# without RW
mn.wo_rw <- multinom(CC~IR+SSPG,data=df,trace=F)
summary(mn.wo_rw)
```

```
## Call:
## multinom(formula = CC ~ IR + SSPG, data = df, trace = F)
##
## Coefficients:
##   (Intercept)           IR       SSPG
## 2   -4.548408  0.003257602 0.01951007
## 1   -7.110590 -0.013427199 0.04259435
##
## Std. Errors:
##   (Intercept)          IR        SSPG
## 2   0.7714595 0.002292307 0.004451874
## 1   1.6882103 0.004651300 0.007973417
```

```
##
## Residual Deviance: 144.0578
## AIC: 156.0578
```

```
df$mn.wo_rw.pred = predict(mn.wo_rw,data=df,type='class')
classif_rate.wo_rw = mean(df$mn.wo_rw.pred==df$CC)
cat(sprintf('Multinomial accuracy without RW: %.3f\n',classif_rate.wo_rw))
```

```
## Multinomial accuracy without RW: 0.814
```

```
# with RW
mw_w_rw <- multinom(CC~IR+SSPG+RW,data=df,trace=F)
summary(mw_w_rw)
```

```
## Call:
## multinom(formula = CC ~ IR + SSPG + RW, data = df, trace = F)
##
## Coefficients:
##   (Intercept)           IR       SSPG        RW
## 2   -7.615261  0.003586749 0.01641449  3.472572
## 1   -1.845230 -0.013353688 0.04550552 -5.867196
##
## Std. Errors:
##   (Intercept)          IR        SSPG       RW
## 2    2.335615 0.002349168 0.004981886 2.446151
## 1    3.463507 0.005019289 0.009241721 3.866580
##
## Residual Deviance: 136.8293
## AIC: 152.8293
```

```
df$mn.w_rw.pred = predict(mw_w_rw,data=df,type='class')
classif_rate.w_rw = mean(df$mn.w_rw.pred==df$CC)
cat(sprintf('Multinomial accuracy with RW: %.3f',classif_rate.w_rw))
```

```
## Multinomial accuracy with RW: 0.828
```

There isn't a significant improvement in classification rate (accuracy) when we add RW to the multinomial logistic model with IR and SSPG.

### Part b

```
library(MASS)
# without RW
mo.wo_rw <- polr(CC~IR+SSPG,data=df,Hess=TRUE)
summary(mo.wo_rw)
```

```
## Call:
## polr(formula = CC ~ IR + SSPG, data = df, Hess = TRUE)
##
## Coefficients:
##          Value Std. Error t value
## IR   -0.004058   0.001746  -2.324
## SSPG  0.028141   0.003581   7.858
##
## Intercepts:
##     Value   Std. Error t value
## 3|2  4.1893  0.6622     6.3263
## 2|1  6.7944  0.8563     7.9348
##
```

```
## Residual Deviance: 163.498
## AIC: 171.498
```

```r
df$mo.wo_rw.pred = predict(mo.wo_rw,data=df,type='class')
classif_rate.wo_rw = mean(df$mo.wo_rw.pred==df$CC)
cat(sprintf('Ordinal accuracy without RW: %.3f\n',classif_rate.wo_rw))
```

```
## Ordinal accuracy without RW: 0.786
```

```r
# with RW
mo.w_rw <- polr(CC~IR+SSPG+RW,data=df,Hess=TRUE)
summary(mo.w_rw)
```

```
## Call:
## polr(formula = CC ~ IR + SSPG + RW, data = df, Hess = TRUE)
##
## Coefficients:
##          Value Std. Error t value
## IR   -0.003766   0.001783  -2.113
## SSPG  0.029279   0.003815   7.675
## RW   -1.923151   1.860977  -1.033
##
## Intercepts:
##     Value   Std. Error t value
## 3|2  2.5246  1.7152     1.4719
## 2|1  5.1527  1.7730     2.9061
##
## Residual Deviance: 162.4219
## AIC: 172.4219
```

```r
df$mo.w_rw.pred = predict(mo.w_rw,data=df,type='class')
classif_rate.w_rw = mean(df$mo.w_rw.pred==df$CC)
cat(sprintf('Ordinal accuracy with RW: %.3f',classif_rate.w_rw))
```

```
## Ordinal accuracy with RW: 0.786
```

Thus, there isn't a significant improvement in classification rate (accuracy) when we add RW to the ordinal logistic model with IR and SSPG.

## 3) Ex 13.1

```r
df = read.table('table6.6.txt',header=T)
```

### Least Squares Model

```r
m.ls = lm(Y~.,data=df)
summary(m.ls)
```

```
##
## Call:
## lm(formula = Y ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3351 -2.1281  0.1605  2.2670  5.6382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1402     3.1412  -0.045   0.9657
## N            64.9755    25.1959   2.579   0.0365 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.201 on 7 degrees of freedom
## Multiple R-squared:  0.4872, Adjusted R-squared:  0.4139
## F-statistic:  6.65 on 1 and 7 DF,  p-value: 0.03654
```

```
AIC(m.ls)
```

```
## [1] 55.11424
```

**Transformed Model**

```
m.tf = lm(sqrt(Y)~.,data=df)
summary(m.tf)
```

```
##
## Call:
## lm(formula = sqrt(Y) ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9690 -0.7655  0.1906  0.5874  1.0211
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1692     0.5783   2.022   0.0829 .
## N            11.8564     4.6382   2.556   0.0378 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7733 on 7 degrees of freedom
## Multiple R-squared:  0.4828, Adjusted R-squared:  0.4089
## F-statistic: 6.535 on 1 and 7 DF,  p-value: 0.03776
```

```
AIC(m.tf)
```

```
## [1] 24.65181
```

**Poisson Model**

```
m.poisson = glm(Y~.,data=df,family="poisson")
summary(m.poisson)
```

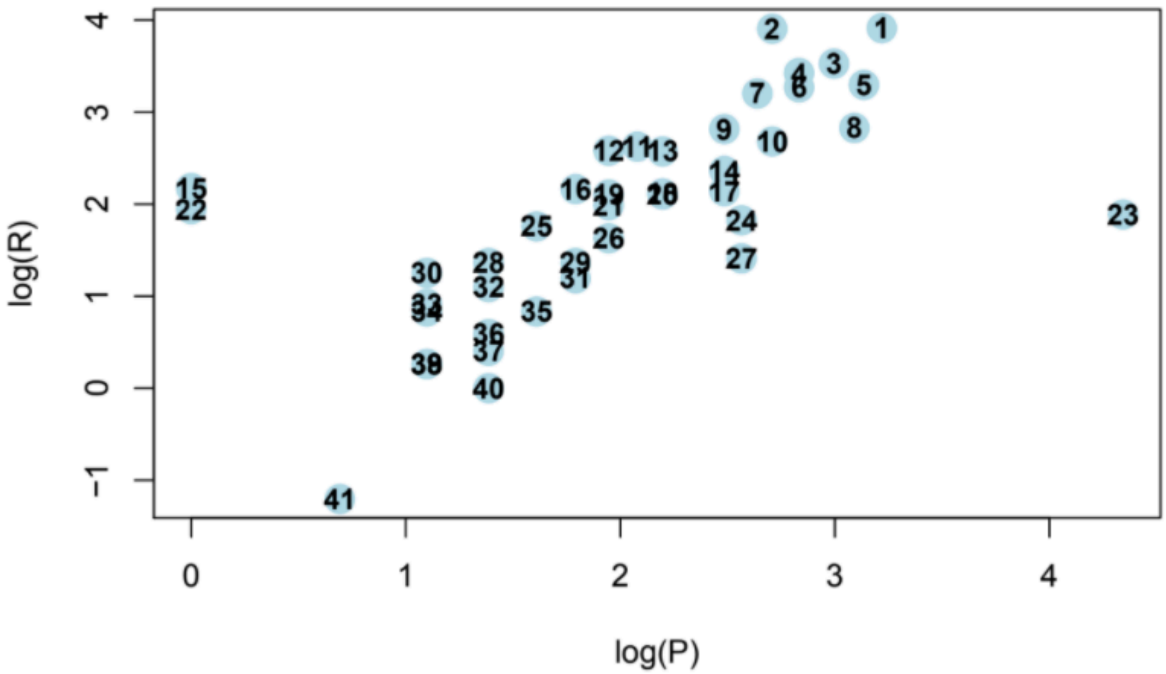
```
##
## Call:
## glm(formula = Y ~ ., family = "poisson", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.81894  -1.69082   0.06495   1.02407   2.06811
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8945     0.3265   2.739  0.00615 **
## N             8.5018     2.1575   3.941 8.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 31.859  on 8  degrees of freedom
## Residual deviance: 16.291  on 7  degrees of freedom
## AIC: 52.251
##
## Number of Fisher Scoring iterations: 5
```

The transformed model is the best as it has the lowest AIC.

## 4) Ex 13.4

```
df = read.table('table6.17.txt',header=T)
plot(log(R)~log(P),col="lightblue",pch=19,cex=2,data=df)
text(log(R)~log(P),labels=rownames(df),data=df,cex=0.9,font=2)
```




```
df_good = df[-c(15,22,23,41),]
```

The special features of each points are

- 15 is high leverage and an outlier
- 22 is high leverage and an outlier
- 23 is high leverage and an outlier
- 41 is high leverage

**Least Squares with all data points**

```
mod.ls.all = lm(log(R)~log(P),data=df)
summary(mod.ls.all)
```

```
##
## Call:
## lm(formula = log(R) ~ log(P), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01534 -0.53524  0.04836  0.50718  1.94245
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2323     0.3398   0.684    0.498
## log(P)        0.8354     0.1571   5.318 4.57e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8719 on 39 degrees of freedom
## Multiple R-squared:  0.4203, Adjusted R-squared:  0.4055
## F-statistic: 28.28 on 1 and 39 DF,  p-value: 4.575e-06
```

**Least Squares with problematic points removed**

```
mod.ls.good = lm(log(R)~log(P),data=df_good)
summary(mod.ls.good)
```

```
##
## Call:
## lm(formula = log(R) ~ log(P), data = df_good)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2759 -0.3202  0.1032  0.3523  1.0137
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.9941     0.2845  -3.494  0.00131 **
## log(P)        1.4351     0.1318  10.891 8.67e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5185 on 35 degrees of freedom
## Multiple R-squared:  0.7722, Adjusted R-squared:  0.7657
## F-statistic: 118.6 on 1 and 35 DF,  p-value: 8.667e-13
```

**Robust Regression**

```
library(MASS)
mod.rr = rlm(log(R)~log(P),data=df)
summary(mod.rr)
```

```
##
## Call: rlm(formula = log(R) ~ log(P), data = df)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -2.612335 -0.475388 -0.002696  0.453661  2.492415
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept) -0.3177  0.2755    -1.1532
## log(P)       1.1090  0.1273     8.7083
##
## Residual standard error: 0.7048 on 39 degrees of freedom
```

The coefficients for least squares with problematic points removed and the coefficients for robust regression are quite similar. Both sets of coefficients are far from those found by least squares with all data points included. In this case, removing the points and using least squares is better since robust regression still suffers from the masking effect of points 15 and 22. Notice that if we fit a robust regression model using the dataset with problematic points removed, the coefficients are almost exactly the same as the least squares fit to this dataset.

```r
library(MASS)
mod.rr = rlm(log(R)~log(P),data=df_good)
summary(mod.rr)
```

```
##
## Call: rlm(formula = log(R) ~ log(P), data = df_good)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29627 -0.34114  0.08237  0.33212  0.99344
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept) -0.9727  0.2796    -3.4789
## log(P)       1.4347  0.1295    11.0797
##
## Residual standard error: 0.5058 on 35 degrees of freedom
```

## 5) Ex 13.5

```r
dfraw = read.table('table12.15.txt',header=T)
dfraw$Failures = dfraw$Attempts-dfraw$Success
```

### Part a

```r
m.poisson = glm(Success~Distance+offset(log(Attempts)),data=dfraw,family='poisson')
summary(m.poisson)
```

```
##
## Call:
## glm(formula = Success ~ Distance + offset(log(Attempts)), family = "poisson",
##     data = dfraw)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.82704  -0.79951   0.01202   0.90292   1.16179
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.553215   0.123134   4.493 7.03e-06 ***
## Distance    -0.038326   0.004133  -9.274  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 103.831  on 9  degrees of freedom
## Residual deviance:  14.219  on 8  degrees of freedom
## AIC: 70.643
##
## Number of Fisher Scoring iterations: 4
```

### Part b

```r
dfraw$pi = dfraw$Success/dfraw$Attempts
dfraw$logodds = log(dfraw$pi/(1-dfraw$pi))
```

```r
mod.logistic = lm(logodds~Distance,data=dfraw)
summary(mod.logistic)
```

```
##
## Call:
## lm(formula = logodds ~ Distance, data = dfraw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09075 -0.16071  0.08111  0.14840  0.91232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8080     0.4606   8.267 3.45e-05 ***
## Distance     -0.1078     0.0126  -8.560 2.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.536 on 8 degrees of freedom
## Multiple R-squared:  0.9016, Adjusted R-squared:  0.8893
## F-statistic: 73.28 on 1 and 8 DF,  p-value: 2.674e-05
```

```r
AIC(mod.logistic)
```

```
## [1] 19.67341
```

### Part c

The logistic model is preferable since it has a significantly lower AIC than the Poisson model.