HW01_appendix

Zach White

January 28, 2017

```r
# First one
ars.data = read.table("http://www2.stat.duke.edu/~st118/fye12takehome.txt", sep = " ", header = TRUE)
ars.data = read.table("C://Users//Zachary//Desktop//Winter 2017//STA 723 Case Studies//STA_723_case//HW
head(ars.data)
```

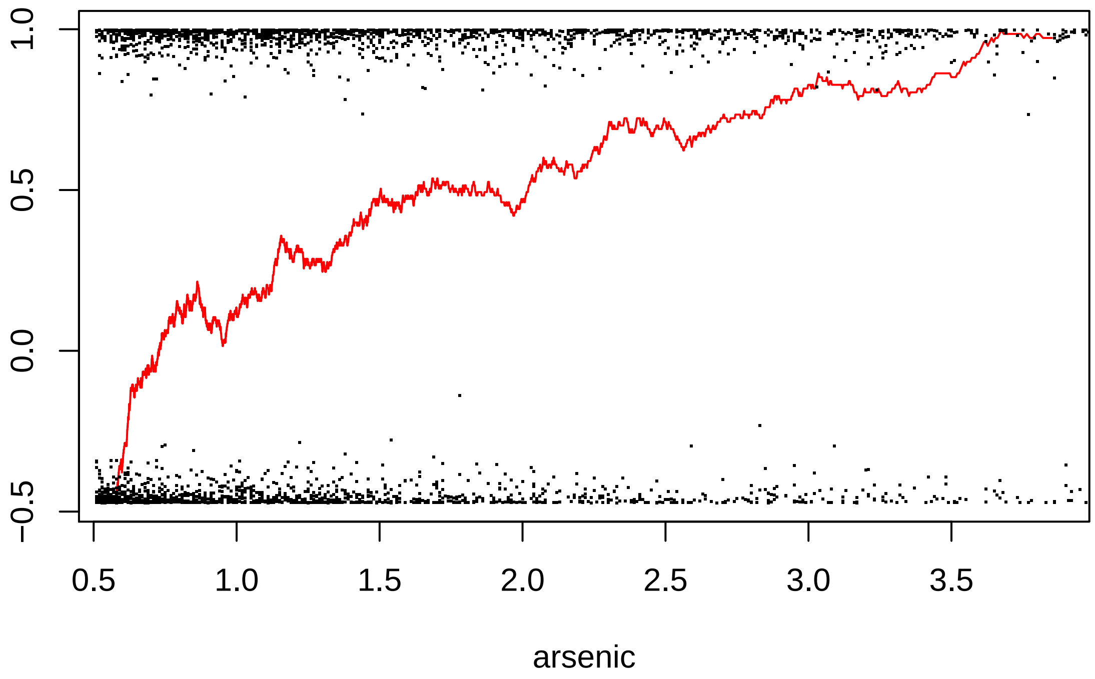
```
##   switch arsenic   dist assoc educ
## 1      1    2.36 16.826     0    0
## 2      1    0.71 47.322     0    0
## 3      0    2.07 20.967     0   10
## 4      1    1.15 21.486     0   12
## 5      1    1.10 40.874     1   14
## 6      1    3.90 69.518     1    9
```

```r
summary(ars.data)
```

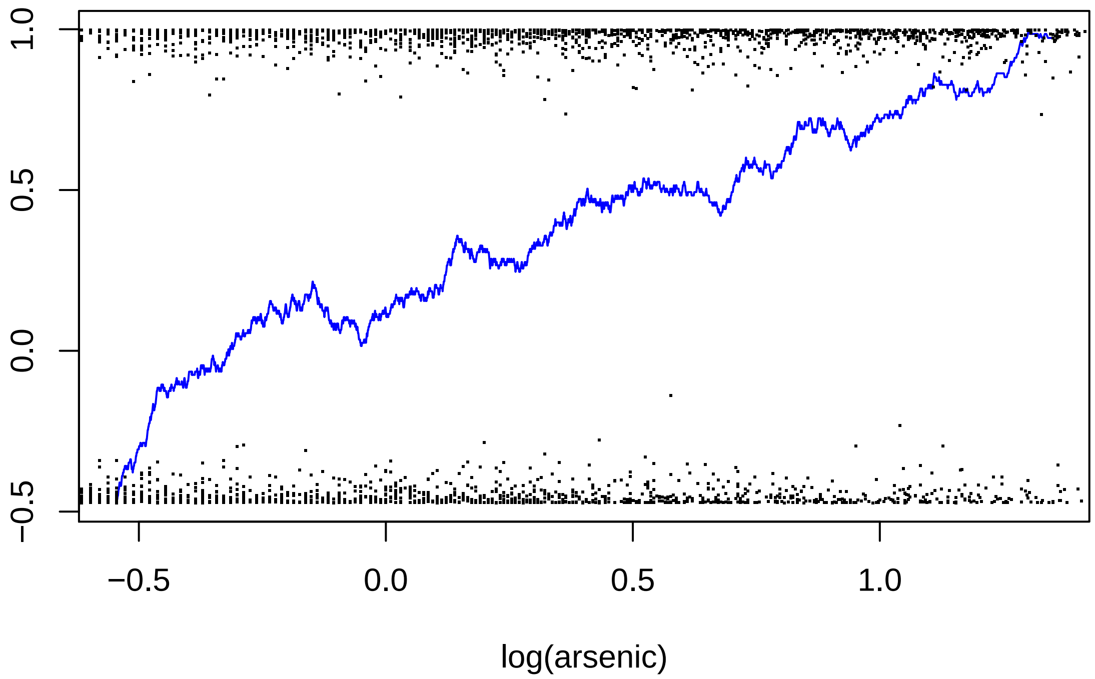
```
##      switch          arsenic           dist             assoc
##  Min.   :0.0000   Min.   :0.510   Min.   :  0.387   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.820   1st Qu.: 21.117   1st Qu.:0.0000
##  Median :1.0000   Median :1.300   Median : 36.761   Median :0.0000
##  Mean   :0.5752   Mean   :1.657   Mean   : 48.332   Mean   :0.4228
##  3rd Qu.:1.0000   3rd Qu.:2.200   3rd Qu.: 64.041   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :9.650   Max.   :339.531   Max.   :1.0000
##       educ
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 5.000
##  Mean   : 4.828
##  3rd Qu.: 8.000
##  Max.   :17.000
```

```r
# visualize switch(arsenic)/(log(arsenic))
# Moving window:
data_arsenic_order <- ars.data[order(ars.data$arsenic),]
n = dim(ars.data)[1]
window_size = 400
avg_switch_window <- rep(0,n-window_size)
avg_arsenic_window <- rep(0,n-window_size)
for (i in 1:(n-window_size)) {
  avg_switch_window[i] <- mean(data_arsenic_order$switch[i:(i+window_size)])
  avg_arsenic_window[i] <- mean(data_arsenic_order$arsenic[i:(i+window_size)])
}

par(mfrow = c(1,1))
plot(avg_arsenic_window, log(avg_switch_window/(1-avg_switch_window)), col="red",
     type="l", xlab="arsenic", ylab="")
temp <- (ars.data$switch==0)*(min(log(avg_switch_window/(1-avg_switch_window)))+(rnorm(n,0,.15))^2)+
        (ars.data$switch==1)*(max(log(avg_switch_window/(1-avg_switch_window)))-rnorm(n,0,.15)^2)
points(ars.data$arsenic, temp, pch=".")
```

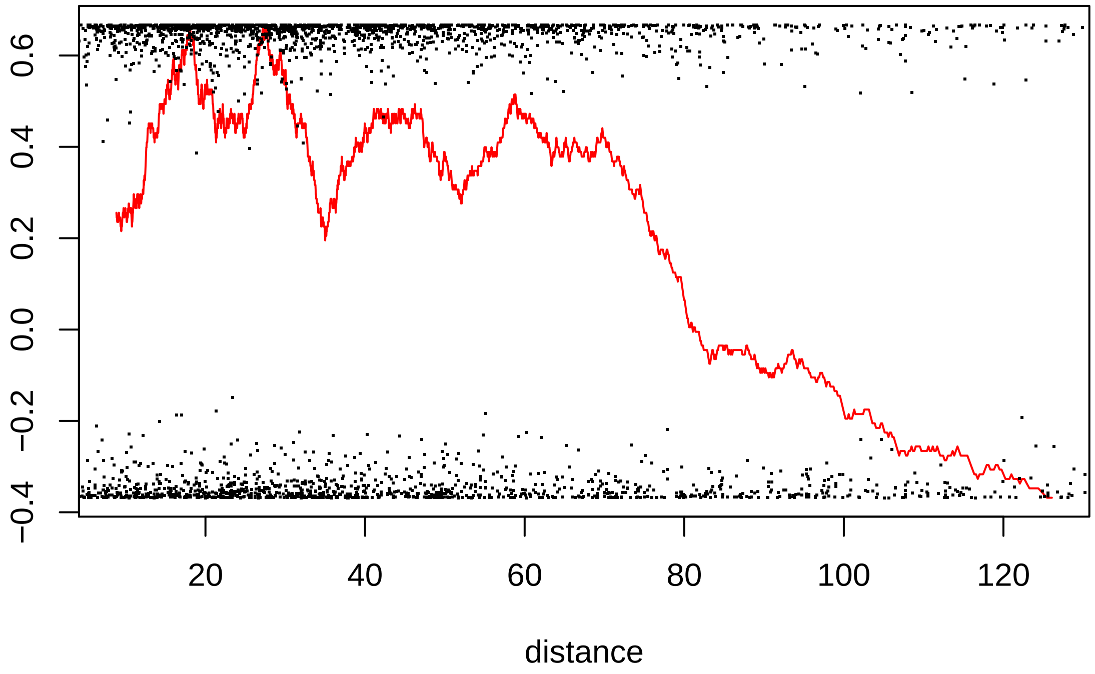


```
#plot(avg_arsenic_window, avg_switch_window, col="red", type="l", ylab="p")
plot(log(avg_arsenic_window), log(avg_switch_window/(1-avg_switch_window)), col="blue",
     type="l", xlab="log(arsenic)", ylab="")
points(log(ars.data$arsenic), temp, pch=".")
```

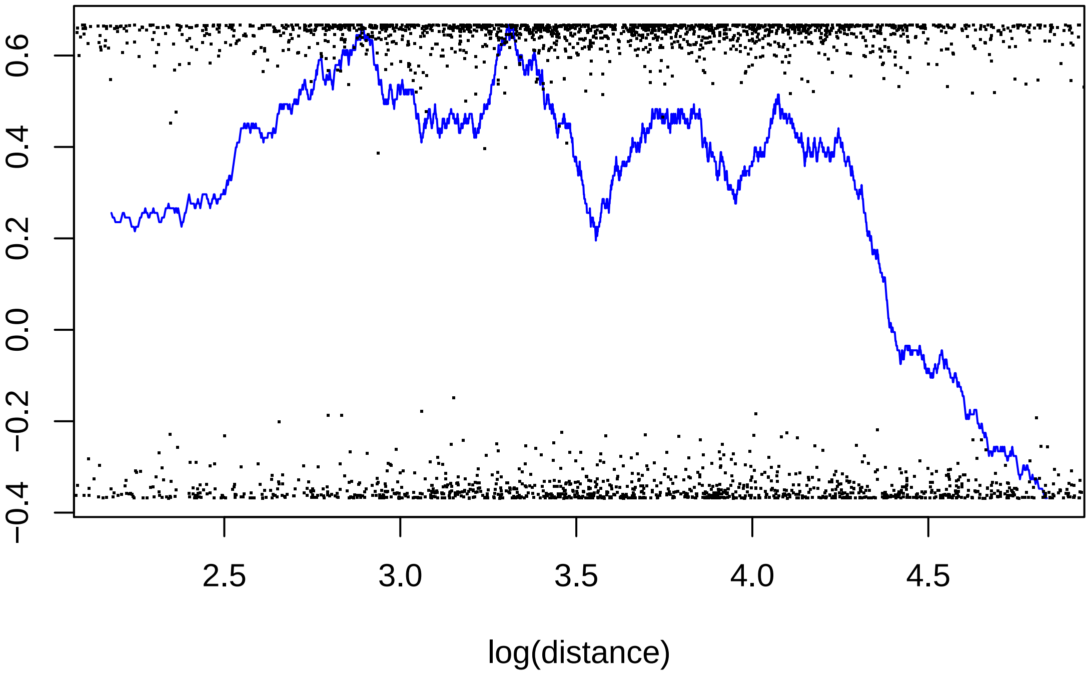


```
#plot(log(avg_arsenic_window), avg_switch_window, col="blue", type="l", ylab="p")


# visualize switch(distance)/(log(distance))
data_distance_order <- ars.data[order(ars.data$dist),]
window_size = 400
avg_switch_window <- rep(0,n-window_size)
avg_distance_window <- rep(0,n-window_size)
for (i in 1:(n-window_size)) {
  avg_switch_window[i] <- mean(data_distance_order$switch[i:(i+window_size)])
  avg_distance_window[i] <- mean(data_distance_order$dist[i:(i+window_size)])
}
#par(mfrow = c(1,1))
plot(avg_distance_window, log(avg_switch_window/(1-avg_switch_window)), col="red", type="l", xlab="dista
temp <- (ars.data$switch==0)*(min(log(avg_switch_window/(1-avg_switch_window)))+(rnorm(n,0,.15))^2)+
  (ars.data$switch==1)*(max(log(avg_switch_window/(1-avg_switch_window)))-rnorm(n,0,.15)^2)
points(ars.data$dist, temp, pch=".")
```



```
plot(log(avg_distance_window), log(avg_switch_window/(1-avg_switch_window)), col="blue", type="l", xlab
points(log(ars.data$dist), temp, pch=".")
```



```
# Consider spline, perhaps in shorter distances, terrain or other factors are more pertinent.
# Perhaps control with mode of transporatation!

ars.data$arsenic <- log(ars.data$arsenic)

hist(ars.data$educ)
# Factor education:
ars.data$educ <- 1*as.numeric(ars.data$educ>=1 & ars.data$educ<=6) +
                  2*as.numeric(ars.data$educ>=7 & ars.data$educ<=12) +
                  3*as.numeric(ars.data$educ>=13)
ars.data$educ <- as.factor(ars.data$educ)

sum(ars.data$switch==0 & ars.data$assoc==0)
```

```
## [1] 714
```

```
sum(ars.data$switch==0 & ars.data$assoc==1)
```

```
## [1] 569
```

```
sum(ars.data$switch==1 & ars.data$assoc==0)
```

```
## [1] 1029
```

```
sum(ars.data$switch==1 & ars.data$assoc==1)
```

```
## [1] 708
```

```
# <- if anything, a negative relation
```

```r
#*****************************************

library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
par(mfrow = c(1,1))

# Full logistic Model
full.log = glm(switch~. , data = ars.data, family = binomial())
summary(full.log)
```

```
##
## Call:
## glm(formula = switch ~ ., family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1311  -1.1647   0.7248   1.0281   1.8925
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.497509   0.091643   5.429 5.67e-08 ***
## arsenic      0.891292   0.069071  12.904  < 2e-16 ***
## dist        -0.009768   0.001063  -9.185  < 2e-16 ***
## assoc       -0.116250   0.077872  -1.493    0.135
## educ1       -0.130415   0.092372  -1.412    0.158
## educ2        0.436359   0.102701   4.249 2.15e-05 ***
## educ3        0.276419   0.258175   1.071    0.284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3858.0  on 3013  degrees of freedom
## AIC: 3872
##
## Number of Fisher Scoring iterations: 4
```

```r
full.log.roc=roc(ars.data$switch ~ predict(full.log))
full.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(full.log))
##
## Data: predict(full.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch 1).
## Area under the curve: 0.6636
```

```r
#anova(full.log, test = "Chisq")
#plot(full.log.roc, col="red")

#take out assoc:
full.noassoc.log = glm(switch~.-assoc , data = ars.data, family = binomial())
summary(full.noassoc.log)
```

```
##
## Call:
## glm(formula = switch ~ . - assoc, family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1126  -1.1668   0.7252   1.0308   1.9164
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.450121   0.085889   5.241 1.60e-07 ***
## arsenic      0.892967   0.069039  12.934  < 2e-16 ***
## dist        -0.009764   0.001064  -9.179  < 2e-16 ***
## educ1       -0.137319   0.092239  -1.489    0.137
## educ2        0.434442   0.102667   4.232 2.32e-05 ***
## educ3        0.292248   0.257718   1.134    0.257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3860.2  on 3014  degrees of freedom
## AIC: 3872.2
##
## Number of Fisher Scoring iterations: 4
```

```r
full.noassoc.log.roc=roc(ars.data$switch ~ predict(full.noassoc.log))
full.noassoc.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(full.noassoc.log))
##
## Data: predict(full.noassoc.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch 1
## Area under the curve: 0.6632
```

```r
#anova(full.log, test = "Chisq")
#plot(full.log.roc, col="red")

# Nah, just take the interaction between arsenic and distance:
interactions2.log = glm(switch ~ arsenic + dist + educ + arsenic*educ, data = ars.data, family = binomi
summary(interactions2.log)
```

```
##
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic * educ,
##     family = binomial(), data = ars.data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2929  -1.1568   0.7193   1.0438   1.9010
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.506752   0.090133   5.622 1.88e-08 ***
## arsenic        0.715716   0.117606   6.086 1.16e-09 ***
## dist          -0.009804   0.001069  -9.173  < 2e-16 ***
## educ1         -0.176360   0.102510  -1.720  0.08536 .
## educ2          0.314622   0.111536   2.821  0.00479 **
## educ3          0.291093   0.281352   1.035  0.30085
## arsenic:educ1  0.121341   0.153325   0.791  0.42871
## arsenic:educ2  0.545686   0.181827   3.001  0.00269 **
## arsenic:educ3 -0.042254   0.459142  -0.092  0.92668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3850.2  on 3011  degrees of freedom
## AIC: 3868.2
##
## Number of Fisher Scoring iterations: 4
```

```
#anova(interactions2.log, test = "Chisq")
interactions2.log.roc=roc(ars.data$switch ~ predict(interactions2.log))
interactions2.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(interactions2.log))
##
## Data: predict(interactions2.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch
## Area under the curve: 0.6673
```

```
# Nah, just take the interaction between arsenic and distance add assoc:
interactions3.log = glm(switch ~ arsenic + dist + educ + +assoc + arsenic*educ, data = ars.data, family
summary(interactions2.log)
```

```
##
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic * educ,
##     family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2929  -1.1568   0.7193   1.0438   1.9010
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.506752   0.090133   5.622 1.88e-08 ***
## arsenic        0.715716   0.117606   6.086 1.16e-09 ***
## dist          -0.009804   0.001069  -9.173  < 2e-16 ***
## educ1         -0.176360   0.102510  -1.720  0.08536 .
## educ2          0.314622   0.111536   2.821  0.00479 **
## educ3          0.291093   0.281352   1.035  0.30085
## arsenic:educ1  0.121341   0.153325   0.791  0.42871
## arsenic:educ2  0.545686   0.181827   3.001  0.00269 **
## arsenic:educ3 -0.042254   0.459142  -0.092  0.92668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3850.2  on 3011  degrees of freedom
## AIC: 3868.2
##
## Number of Fisher Scoring iterations: 4
```

```r
#anova(interactions2.log, test = "Chisq")
interactions3.log.roc=roc(ars.data$switch ~ predict(interactions3.log))
interactions3.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(interactions3.log))
##
## Data: predict(interactions3.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch
## Area under the curve: 0.6678
```

```r
# all interactions:
all.interactions.log = glm(switch ~ .*., data = ars.data, family = binomial())
summary(all.interactions.log)
```

```
##
## Call:
## glm(formula = switch ~ . * ., family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3229  -1.1520   0.6936   1.0315   2.1303
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.528174   0.141823   3.724 0.000196 ***
## arsenic        0.953303   0.162943   5.851 4.90e-09 ***
## dist          -0.010908   0.002443  -4.466 7.99e-06 ***
## assoc         -0.040947   0.175237  -0.234 0.815244
## educ1          0.057646   0.176825   0.326 0.744419
## educ2          0.118917   0.187807   0.633 0.526610
## educ3          0.292203   0.426289   0.685 0.493055
## arsenic:dist  -0.001913   0.001896  -1.009 0.313098
## arsenic:assoc -0.336922   0.141263  -2.385 0.017076 *
## arsenic:educ1  0.194385   0.165102   1.177 0.239051
## arsenic:educ2  0.414763   0.190257   2.180 0.029257 *
## arsenic:educ3 -0.229161   0.480811  -0.477 0.633637
## dist:assoc     0.003332   0.002223   1.499 0.133866
## dist:educ1    -0.003734   0.002725  -1.370 0.170534
## dist:educ2     0.006255   0.002848   2.196 0.028060 *
## dist:educ3     0.007135   0.006365   1.121 0.262273
## assoc:educ1   -0.182249   0.187138  -0.974 0.330118
## assoc:educ2   -0.212180   0.209260  -1.014 0.310604
## assoc:educ3   -1.124787   0.572373  -1.965 0.049399 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3821.9  on 3001  degrees of freedom
## AIC: 3859.9
##
## Number of Fisher Scoring iterations: 4
```

```
all.interactions.log.roc=roc(ars.data$switch ~ predict(all.interactions.log))
all.interactions.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(all.interactions.log))
##
## Data: predict(all.interactions.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$swit
## Area under the curve: 0.674
```

```
#plot(all.interactions.log.roc, add=TRUE, col='blue')

# null:
all.interactions.log = glm(switch ~ .*., data = ars.data, family = binomial())
summary(all.interactions.log)
```

```
##
## Call:
## glm(formula = switch ~ . * ., family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3229  -1.1520   0.6936   1.0315   2.1303
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.528174   0.141823   3.724 0.000196 ***
## arsenic         0.953303   0.162943   5.851 4.90e-09 ***
## dist           -0.010908   0.002443  -4.466 7.99e-06 ***
## assoc          -0.040947   0.175237  -0.234 0.815244
## educ1           0.057646   0.176825   0.326 0.744419
## educ2           0.118917   0.187807   0.633 0.526610
## educ3           0.292203   0.426289   0.685 0.493055
## arsenic:dist   -0.001913   0.001896  -1.009 0.313098
## arsenic:assoc -0.336922   0.141263  -2.385 0.017076 *
## arsenic:educ1  0.194385   0.165102   1.177 0.239051
## arsenic:educ2  0.414763   0.190257   2.180 0.029257 *
## arsenic:educ3 -0.229161   0.480811  -0.477 0.633637
## dist:assoc     0.003332   0.002223   1.499 0.133866
## dist:educ1    -0.003734   0.002725  -1.370 0.170534
## dist:educ2     0.006255   0.002848   2.196 0.028060 *
## dist:educ3     0.007135   0.006365   1.121 0.262273
## assoc:educ1   -0.182249   0.187138  -0.974 0.330118
## assoc:educ2   -0.212180   0.209260  -1.014 0.310604
## assoc:educ3   -1.124787   0.572373  -1.965 0.049399 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3821.9  on 3001  degrees of freedom
## AIC: 3859.9
##
## Number of Fisher Scoring iterations: 4
```

```
all.interactions.log.roc=roc(ars.data$switch ~ predict(all.interactions.log))
all.interactions.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(all.interactions.log))
##
## Data: predict(all.interactions.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$swit
## Area under the curve: 0.674
```

```
# not impressive, only association interactions?
interactions1.log <- glm(switch~.+assoc*., data = ars.data, family = binomial())
summary(interactions1.log)
```

```
##
## Call:
## glm(formula = switch ~ . + assoc * ., family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2196  -1.1627   0.7181   1.0341   1.9762
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.455585   0.113319   4.020 5.81e-05 ***
## arsenic        1.023274   0.093056  10.996  < 2e-16 ***
## dist          -0.010743   0.001411  -7.612 2.70e-14 ***
## assoc         -0.010770   0.174063  -0.062 0.950662
## educ1         -0.064215   0.123106  -0.522 0.601930
## educ2          0.510379   0.135257   3.773 0.000161 ***
## educ3          0.594086   0.317860   1.869 0.061620 .
## arsenic:assoc -0.310333   0.139381  -2.227 0.025980 *
## dist:assoc     0.002446   0.002153   1.136 0.255879
## assoc:educ1   -0.171762   0.186719  -0.920 0.357628
## assoc:educ2   -0.188923   0.208331  -0.907 0.364492
## assoc:educ3   -1.084969   0.582186  -1.864 0.062376 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3849.2  on 3008  degrees of freedom
## AIC: 3873.2
##
## Number of Fisher Scoring iterations: 4
```

```
anova(interactions1.log, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           3019     4118.1
## arsenic        1  128.811      3018     3989.3 < 2.2e-16 ***
## dist           1   90.921      3017     3898.4 < 2.2e-16 ***
## assoc          1    2.946      3016     3895.4   0.08607 .
## educ           3   37.420      3013     3858.0 3.749e-08 ***
## arsenic:assoc  1    3.655      3012     3854.3   0.05591 .
## dist:assoc     1    1.189      3011     3853.2   0.27560
## assoc:educ     3    3.987      3008     3849.2   0.26282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
interactions1.log.roc=roc(ars.data$switch ~ predict(interactions1.log))
plot(interactions1.log.roc, add=TRUE, col='green')
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(interactions1.log))
##
## Data: predict(interactions1.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch
## Area under the curve: 0.6664
```

```
# Nah, just take the interaction between arsenic and distance:
interactions2.log = glm(switch ~ arsenic + dist + educ + arsenic*educ, data = ars.data, family = binomi
summary(interactions2.log)
```

```
##
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic * educ,
##     family = binomial(), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2929  -1.1568   0.7193   1.0438   1.9010
```

```
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.506752   0.090133   5.622 1.88e-08 ***
## arsenic         0.715716   0.117606   6.086 1.16e-09 ***
## dist           -0.009804   0.001069  -9.173  < 2e-16 ***
## educ1          -0.176360   0.102510  -1.720  0.08536 .
## educ2           0.314622   0.111536   2.821  0.00479 **
## educ3           0.291093   0.281352   1.035  0.30085
## arsenic:educ1   0.121341   0.153325   0.791  0.42871
## arsenic:educ2   0.545686   0.181827   3.001  0.00269 **
## arsenic:educ3 -0.042254   0.459142  -0.092  0.92668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3850.2  on 3011  degrees of freedom
## AIC: 3868.2
##
## Number of Fisher Scoring iterations: 4
```

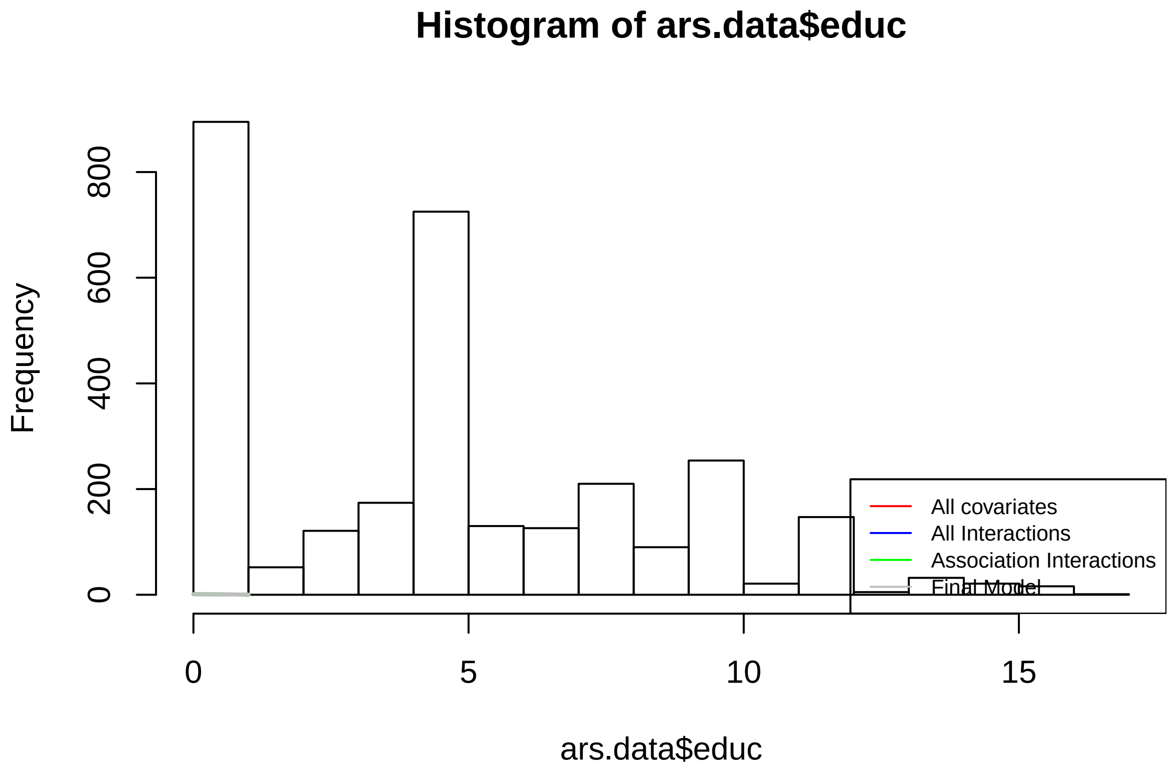
```
anova(interactions2.log, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           3019     4118.1
## arsenic        1  128.811      3018     3989.3 < 2.2e-16 ***
## dist           1   90.921      3017     3898.4 < 2.2e-16 ***
## educ           3   38.139      3014     3860.2 2.641e-08 ***
## arsenic:educ   3   10.033      3011     3850.2   0.01829 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
interactions2.log.roc=roc(ars.data$switch ~ predict(interactions2.log))
plot(interactions2.log.roc, add=TRUE, col='grey')
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(interactions2.log))
##
## Data: predict(interactions2.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch
## Area under the curve: 0.6673
```

```
legend("bottomright", lty=1, col=c("red", "blue", "green", "grey"),
       legend=c("All covariates", "All Interactions", "Association Interactions", "Final Model"),cex=0.
```



```
interactions2.log.coef <- interactions2.log$coefficients
exp(confint.default(interactions2.log))
```

```
##                      2.5 %    97.5 %
## (Intercept)    1.3910999 1.9806187
## arsenic        1.6245205 2.5759499
## dist           0.9881714 0.9923201
## educ1          0.6857265 1.0248599
## educ2          1.1007767 1.7044246
## educ3          0.7707833 2.3222418
## arsenic:educ1 0.8359633 1.5247832
## arsenic:educ2 1.2084184 2.4646733
## arsenic:educ3 0.3897862 2.3576096
```

```
interactions2.log.roc
```

```
##
## Call:
## roc.formula(formula = ars.data$switch ~ predict(interactions2.log))
##
## Data: predict(interactions2.log) in 1283 controls (ars.data$switch 0) < 1737 cases (ars.data$switch
## Area under the curve: 0.6673
```

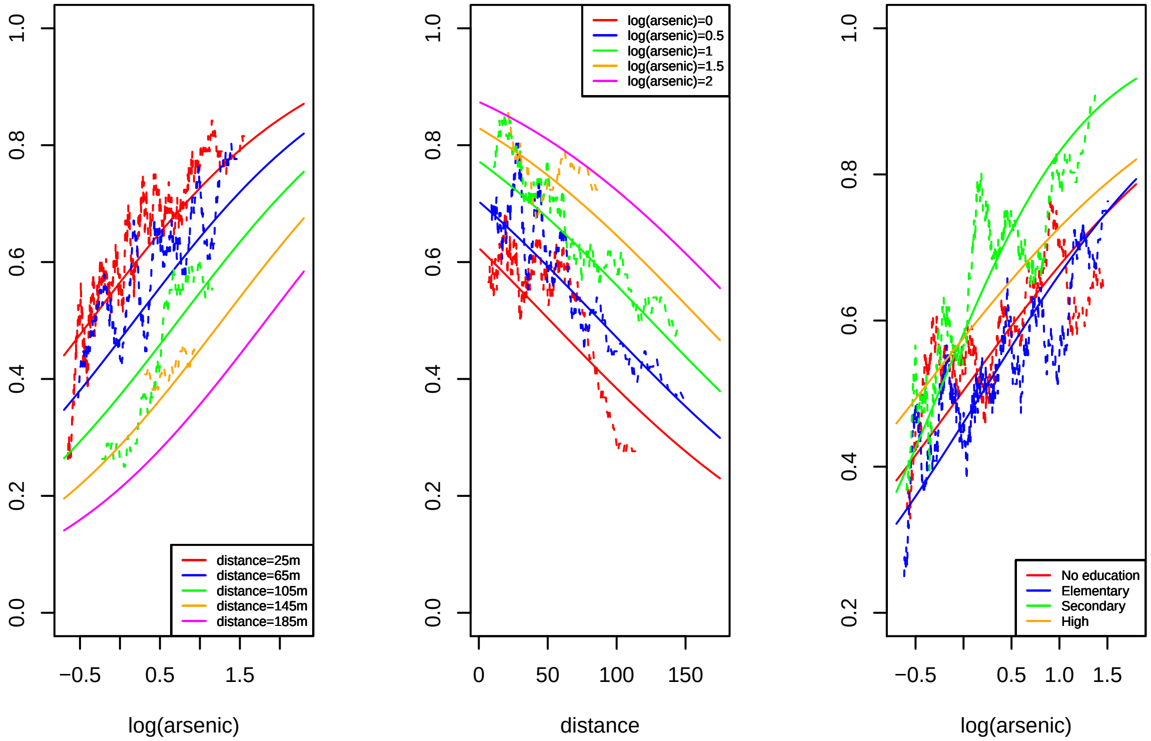
```
 library(aod)
 wald.test(b = coef(interactions2.log), Sigma = vcov(interactions2.log), Terms = 1:9)
```

```
## Wald test:
## ----------
```

```
##
## Chi-squared test:
## X2 = 274.7, df = 9, P(> X2) = 0.0
```

```
#**********************************************
# Switch(arsenic,distance)|education,association
par(mfrow = c(1,3))
cols=c("red", "blue", "green", "orange", "magenta", "black")
data_arsenic_order <- ars.data[order(ars.data$arsenic),]
dseq=seq(25,185,40)
delta=20
el=0
sc=0
hg=0
for (d in dseq) {
  arsenic_range = seq(-.7, 2.3, 0.01)
  fitted_switch <- rep(0, length(arsenic_range))
  for (i in 1:length(arsenic_range)) {
    fitted_switch[i] = 1/(1+exp(-sum(interactions2.log.coef*
                                      c(1,arsenic_range[i],d,el,sc,hg,
                                        arsenic_range[i]*el,arsenic_range[i]*sc,arsenic_range[i]*hg)))
  }
  if (d==dseq[1]) { plot(arsenic_range, fitted_switch, type="l",
                         ylim=c(0,1), col=cols[which(dseq==d)], ylab="", xlab="log(arsenic)") }
  if (d>dseq[1]) { lines(arsenic_range, fitted_switch, type="l", col=cols[which(dseq==d)]) }
  #***********
  temp_data=ars.data[which(ars.data$dist>=d-delta & ars.data$dist<=d+delta),c(1,2)]
  temp_data_arsenic_order <- temp_data[order(temp_data$arsenic),]
  n = dim(temp_data)[1]
  window_size = 75
  if (n > window_size) {
    avg_switch_window <- rep(0,n-window_size)
    avg_arsenic_window <- rep(0,n-window_size)
    for (i in 1:(n-window_size)) {
      avg_switch_window[i] <- mean(temp_data_arsenic_order$switch[i:(i+window_size)])
      avg_arsenic_window[i] <- mean(temp_data_arsenic_order$arsenic[i:(i+window_size)])
    }
    lines(avg_arsenic_window, avg_switch_window, col=cols[which(dseq==d)], lty=2)
  }
  legend("bottomright", legend=paste("distance=",dseq,"m",sep=""), cex=0.6, col=cols[1:length(dseq)], l
}
#**********************************************
# Switch(distance,arsenic)|education,association
cols=c("red", "blue", "green", "orange", "magenta", "black")
data_dist_order <- ars.data[order(ars.data$dist),]
aseq=seq(0,2,0.5)
delta=0.25
el=0
sc=0
hg=0
for (a in aseq) {
  dist_range = seq(1,175, 1)
  fitted_switch <- rep(0, length(dist_range))
  for (i in 1:length(dist_range)) {
    fitted_switch[i] = 1/(1+exp(-sum(interactions2.log.coef*
                                       c(1,a,dist_range[i],el,sc,hg,
                                         a*el,a*sc,a*hg))))
  }
  if (a==aseq[1]) { plot(dist_range, fitted_switch, type="l",
                         ylim=c(0,1), col=cols[which(aseq==a)], ylab="", xlab="distance") }
  if (a>aseq[1]) { lines(dist_range, fitted_switch, type="l", col=cols[which(aseq==a)]) }
  #***********
  temp_data=ars.data[which(ars.data$arsenic>=a-delta & ars.data$arsenic<=a+delta),c(1,3)]
  temp_data_dist_order <- temp_data[order(temp_data$dist),]
  n = dim(temp_data)[1]
  window_size = 75
  if (n > window_size) {
    avg_switch_window <- rep(0,n-window_size)
    avg_dist_window <- rep(0,n-window_size)
    for (i in 1:(n-window_size)) {
      avg_switch_window[i] <- mean(temp_data_dist_order$switch[i:(i+window_size)])
      avg_dist_window[i] <- mean(temp_data_dist_order$dist[i:(i+window_size)])
    }
    lines(avg_dist_window, avg_switch_window, col=cols[which(aseq==a)], lty=2)
  }
  legend("topright", legend=paste("log(arsenic)=",aseq,sep=""), cex=0.6, col=cols[1:length(aseq)], lty=
}

#***********************************************
# Switch(arsenic,distance)|education,association
cols=c("red", "blue", "green", "orange", "magenta", "black")
data_educ_order <- ars.data[order(ars.data$arsenic),]
eseq=c(0,1,2,3)
el=0
sc=0
hg=0
d=50
for (e in eseq) {
  el=(e==1); sc=(e==2); hg=(e==3)
  arsenic_range = seq(-.7, 1.8, 0.01)
  fitted_switch <- rep(0, length(arsenic_range))
  for (i in 1:length(arsenic_range)) {
    fitted_switch[i] = 1/(1+exp(-sum(interactions2.log.coef*
                                       c(1,arsenic_range[i],d,el,sc,hg,
                                         arsenic_range[i]*el,arsenic_range[i]*sc,arsenic_range[i]*hg)))
  }
  if (e==eseq[1]) { plot(arsenic_range, fitted_switch, type="l",
                         ylim=c(0.2,1), col=cols[which(eseq==e)], ylab="", xlab="log(arsenic)") }
  if (e>eseq[1]) { lines(arsenic_range, fitted_switch, type="l", col=cols[which(eseq==e)]) }
  #***********
  temp_data=ars.data[which(ars.data$educ==e),c(1,2)]
  temp_data_arsenic_order <- temp_data[order(temp_data$arsenic),]
  n = dim(temp_data)[1]
  window_size = 75
  if (n > window_size) {
    avg_switch_window <- rep(0,n-window_size)
    avg_arsenic_window <- rep(0,n-window_size)
    for (i in 1:(n-window_size)) {
      avg_switch_window[i] <- mean(temp_data_arsenic_order$switch[i:(i+window_size)])
      avg_arsenic_window[i] <- mean(temp_data_arsenic_order$arsenic[i:(i+window_size)])
    }
    lines(avg_arsenic_window, avg_switch_window, col=cols[which(eseq==e)], lty=2)
  }
  legend("bottomright", legend=c("No education","Elementary", "Secondary", "High"), cex=0.6, col=cols[1
}
```



```
#**********************************************

#**********************************************
#
#
# interactions.log.coef <- interactions.log$coefficients
# a=1/(1+exp(-sum(interactions.log.coef*c(1,log(0.6),log(1+1),0,1,0,log(0.6)*log(1+1)))))
# b=1/(1+exp(-sum(interactions.log.coef*c(1,log(0.6),log(1*1.1+1),0,1,0,log(0.6)*log(1*1.1+1)))))
# b-a
#
# interactions.prob = glm(switch ~ arsenic + dist + educ + arsenic*dist, data = ars.data, family = bino
# summary(interactions.prob)
# interactions.prob.coef <- interactions.prob$coefficients
# a=pnorm(sum(interactions.prob.coef*c(1,log(0.6),log(1+1),0,1,0,log(0.6)*log(1+1))))
# b=pnorm(sum(interactions.prob.coef*c(1,log(0.6),log(1*1.1+1),0,1,0,log(0.6)*log(1*1.1+1))))
# b-a
#
```

```
# interactions.lpm = glm(switch ~ arsenic + dist + educ + arsenic*dist, data = ars.data)
# summary(interactions.lpm)
# interactions.lpm.coef <- interactions.lpm$coefficients
# a=(sum(interactions.prob.coef*c(1,log(0.6),log(1+1),0,1,0,log(0.6)*log(1+1))))
# b=(sum(interactions.prob.coef*c(1,log(0.6),log(1*1.1+1),0,1,0,log(0.6)*log(1*1.1+1))))
# b-a

#***********************************************
# interactions.log.coef <- interactions.log$coefficients
#
#
# before=1/(1+exp(-sum(interactions.log.coef*c(1,log(a),log(d+1),0,se.bef,0,log(a)*log(d+1)))))
# after=1/(1+exp(-sum(interactions.log.coef*c(1,log(a*(1+.1*inc.a)),log(d*(1+.1*inc.d)+1),0,se.aft,0,lo
# "logit"
# after-before
#
# interactions.prob = glm(switch ~ arsenic + dist + educ + arsenic*dist, data = ars.data, family = bino
# #summary(interactions.prob)
# interactions.prob.coef <- interactions.prob$coefficients
# before=pnorm(sum(interactions.prob.coef*c(1,log(a),log(d+1),0,se.bef,0,log(a)*log(d+1))))
# after=pnorm(sum(interactions.prob.coef*c(1,log(a*(1+.1*inc.a)),log(d*(1+.1*inc.d)+1),0,se.aft,0,log(a
# "probit"
# after-before
#
# interactions.lpm = glm(switch ~ arsenic + dist + educ + arsenic*dist, data = ars.data)
# #summary(interactions.lpm)
# interactions.lpm.coef <- interactions.lpm$coefficients
# before=(sum(interactions.prob.coef*c(1,log(a),log(d+1),0,se.bef,0,log(a)*log(d+1))))
# after=(sum(interactions.prob.coef*c(1,log(a*(1+.1*inc.a)),log(d*(1+.1*inc.d)+1),0,se.aft,0,log(a*(1+.
# "LPM"
# after-before

# Full probit Model
full.prob = glm(switch ~ arsenic + dist + educ + arsenic*educ, data = ars.data, family = binomial(link=
summary(full.prob)
```

```
##
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic * educ,
##     family = binomial(link = "probit"), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3491  -1.1576   0.7209   1.0465   1.9185
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.3112408  0.0557709   5.581 2.40e-08 ***
## arsenic         0.4366394  0.0718328   6.079 1.21e-09 ***
## dist           -0.0059884  0.0006472  -9.253  < 2e-16 ***
## educ1          -0.1119231  0.0637093  -1.757  0.07896 .
## educ2           0.1895629  0.0689990   2.747  0.00601 **
## educ3           0.1771603  0.1741325   1.017  0.30897
## arsenic:educ1   0.0815048  0.0936921   0.870  0.38434
## arsenic:educ2  0.3147140  0.1083791   2.904  0.00369 **
## arsenic:educ3 -0.0204427  0.2789975  -0.073  0.94159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3850.9  on 3011  degrees of freedom
## AIC: 3868.9
##
## Number of Fisher Scoring iterations: 4
```

```
# ANOVA
anova(full.prob, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: probit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                          3019     4118.1
## arsenic       1  128.683      3018     3989.4 < 2.2e-16 ***
## dist          1   90.514      3017     3898.9 < 2.2e-16 ***
## educ          3   38.732      3014     3860.2 1.978e-08 ***
## arsenic:educ  3    9.267      3011     3850.9   0.02594 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# **************************** ENOUGH FOR TODAY!


# Add interactions?
# all interactions:
interactions0.prob = glm(switch ~ .*., data = ars.data, family = binomial(link="probit"))
summary(interactions0.prob)
```

```
##
## Call:
## glm(formula = switch ~ . * ., family = binomial(link = "probit"),
##     data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3883  -1.1551   0.6945   1.0338   2.1837
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.325474   0.087015   3.740 0.000184 ***
## arsenic          0.574427   0.098057   5.858 4.68e-09 ***
## dist            -0.006692   0.001475  -4.536 5.72e-06 ***
## assoc         -0.023511   0.107860  -0.218 0.827447
## educ1          0.032402   0.108248   0.299 0.764691
## educ2          0.065476   0.114200   0.573 0.566412
## educ3          0.172733   0.259657   0.665 0.505902
## arsenic:dist  -0.001112   0.001147  -0.969 0.332370
## arsenic:assoc -0.197231   0.084669  -2.329 0.019836 *
## arsenic:educ1  0.122644   0.099533   1.232 0.217877
## arsenic:educ2  0.241976   0.112883   2.144 0.032065 *
## arsenic:educ3 -0.129353   0.289235  -0.447 0.654713
## dist:assoc     0.001984   0.001342   1.478 0.139382
## dist:educ1    -0.002290   0.001641  -1.395 0.162882
## dist:educ2     0.003892   0.001722   2.260 0.023825 *
## dist:educ3     0.004497   0.003915   1.149 0.250641
## assoc:educ1   -0.112270   0.115351  -0.973 0.330411
## assoc:educ2   -0.122032   0.127795  -0.955 0.339628
## assoc:educ3   -0.698455   0.353773  -1.974 0.048347 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3822.3  on 3001  degrees of freedom
## AIC: 3860.3
##
## Number of Fisher Scoring iterations: 4
```

```
anova(interactions0.prob, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: probit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           3019     4118.1
## arsenic         1  128.683      3018     3989.4 < 2.2e-16 ***
## dist            1   90.514      3017     3898.9 < 2.2e-16 ***
## assoc           1    3.182      3016     3895.7  0.074445 .
## educ            3   37.893      3013     3857.8 2.977e-08 ***
## arsenic:dist    1    1.395      3012     3856.4  0.237603
## arsenic:assoc   1    3.623      3011     3852.8  0.056997 .
## arsenic:educ    3    9.095      3008     3843.7  0.028055 *
## dist:assoc      1    1.163      3007     3842.6  0.280898
## dist:educ       3   15.815      3004     3826.7  0.001237 **
## assoc:educ      3    4.420      3001     3822.3  0.219506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# not impressive, only association interactions?
interactions1.prob <- glm(switch~.+assoc*., data = ars.data, family = binomial(link="probit"))
```

```r
summary(interactions1.prob)
```

```
##
## Call:
## glm(formula = switch ~ . + assoc * ., family = binomial(link = "probit"),
##     data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2797  -1.1622   0.7181   1.0374   1.9952
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.2763855  0.0695274   3.975 7.03e-05 ***
## arsenic        0.6226740  0.0551199  11.297  < 2e-16 ***
## dist          -0.0065389  0.0008503  -7.690 1.47e-14 ***
## assoc         -0.0029836  0.1072502  -0.028 0.977807
## educ1         -0.0383086  0.0755016  -0.507 0.611883
## educ2          0.3135608  0.0821373   3.818 0.000135 ***
## educ3          0.3529625  0.1899343   1.858 0.063121 .
## arsenic:assoc -0.1832937  0.0837360  -2.189 0.028601 *
## dist:assoc     0.0014241  0.0013074   1.089 0.276052
## assoc:educ1   -0.1074750  0.1151315  -0.933 0.350563
## assoc:educ2   -0.1123326  0.1274442  -0.881 0.378087
## assoc:educ3   -0.6495038  0.3561022  -1.824 0.068163 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3849.3  on 3008  degrees of freedom
## AIC: 3873.3
##
## Number of Fisher Scoring iterations: 4
```

```r
anova(interactions1.prob, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: probit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           3019     4118.1
## arsenic        1  128.683      3018     3989.4 < 2.2e-16 ***
## dist           1   90.514      3017     3898.9 < 2.2e-16 ***
## assoc          1    3.182      3016     3895.7   0.07444 .
## educ           3   37.893      3013     3857.8 2.977e-08 ***
## arsenic:assoc  1    3.628      3012     3854.2   0.05683 .
## dist:assoc     1    1.093      3011     3853.1   0.29574
## assoc:educ     3    3.849      3008     3849.3   0.27826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Nah, just take the interaction between arsenic and distance:
interactions.prob = glm(switch ~ arsenic + dist + educ + arsenic*dist, data = ars.data, family = binomia
summary(interactions.prob)
```

```
##
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic * dist,
##     family = binomial(link = "probit"), data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2155  -1.1613   0.7171   1.0384   1.8193
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2559726  0.0551487   4.641 3.46e-06 ***
## arsenic       0.6075209  0.0662642   9.168  < 2e-16 ***
## dist         -0.0053617  0.0008255  -6.495 8.28e-11 ***
## educ1        -0.0876376  0.0569163  -1.540    0.124
## educ2         0.2669081  0.0627516   4.253 2.11e-05 ***
## educ3         0.1746511  0.1571672   1.111    0.266
## arsenic:dist -0.0013263  0.0011197  -1.185    0.236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 3858.8  on 3013  degrees of freedom
## AIC: 3872.8
##
## Number of Fisher Scoring iterations: 4
```

```
anova(interactions.prob, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: probit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                          3019     4118.1
## arsenic       1  128.683      3018     3989.4 < 2.2e-16 ***
## dist          1   90.514      3017     3898.9 < 2.2e-16 ***
## educ          3   38.732      3014     3860.2 1.978e-08 ***
## arsenic:dist  1    1.410      3013     3858.8     0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#***************************************************

# Full LPM Model
full.lpm = lm(switch~arsenic + dist + educ + arsenic*dist, data = ars.data)
summary(full.lpm)
```

```
##
## Call:
## lm(formula = switch ~ arsenic + dist + educ + arsenic * dist,
##     data = ars.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9932 -0.4934  0.2287  0.4219  0.8258
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5967511  0.0203349  29.346  < 2e-16 ***
## arsenic       0.2161613  0.0232881   9.282  < 2e-16 ***
## dist         -0.0019841  0.0002981  -6.657 3.32e-11 ***
## educ1        -0.0338856  0.0210016  -1.613    0.107
## educ2         0.0942968  0.0227998   4.136 3.63e-05 ***
## educ3         0.0638483  0.0570453   1.119    0.263
## arsenic:dist -0.0003947  0.0004038  -0.978    0.328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4742 on 3013 degrees of freedom
## Multiple R-squared:  0.08171,    Adjusted R-squared:  0.07988
## F-statistic: 44.68 on 6 and 3013 DF,  p-value: < 2.2e-16
```

```
# ANOVA
anova(full.lpm, test = "Chisq")
```

```
## Analysis of Variance Table
##
## Response: switch
##                Df Sum Sq Mean Sq  F value    Pr(>F)
## arsenic         1  30.79 30.7886 136.8957 < 2.2e-16 ***
## dist            1  20.72 20.7235  92.1432 < 2.2e-16 ***
## educ            3   8.57  2.8568  12.7022 3.005e-08 ***
## arsenic:dist    1   0.21  0.2150   0.9558    0.3283
## Residuals    3013 677.64  0.2249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Add interactions?
# all interactions:
interactions0.lpm = lm(switch ~ .*., data = ars.data)
summary(interactions0.lpm)
```

```
##
## Call:
```

```
## lm(formula = switch ~ . * ., data = ars.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0281 -0.4901  0.2209  0.4182  0.9575
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.6213854  0.0317657  19.562  < 2e-16 ***
## arsenic        0.2074315  0.0345887   5.997 2.25e-09 ***
## dist          -0.0024634  0.0005232  -4.708 2.62e-06 ***
## assoc         -0.0029885  0.0394679  -0.076  0.93965
## educ1          0.0018478  0.0392106   0.047  0.96242
## educ2          0.0123932  0.0411985   0.301  0.76358
## educ3          0.0675644  0.0942017   0.717  0.47329
## arsenic:dist  -0.0003627  0.0004051  -0.895  0.37066
## arsenic:assoc -0.0580418  0.0296153  -1.960  0.05010 .
## arsenic:educ1  0.0354968  0.0352431   1.007  0.31392
## arsenic:educ2  0.0617973  0.0388413   1.591  0.11171
## arsenic:educ3 -0.0615126  0.1023015  -0.601  0.54769
## dist:assoc     0.0005567  0.0004688   1.187  0.23517
## dist:educ1    -0.0006149  0.0005706  -1.078  0.28130
## dist:educ2     0.0016457  0.0006139   2.681  0.00739 **
## dist:educ3     0.0016819  0.0014341   1.173  0.24099
## assoc:educ1   -0.0414802  0.0424295  -0.978  0.32834
## assoc:educ2   -0.0387873  0.0463156  -0.837  0.40240
## assoc:educ3   -0.2652858  0.1309461  -2.026  0.04286 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4726 on 3001 degrees of freedom
## Multiple R-squared:  0.09174,    Adjusted R-squared:  0.08629
## F-statistic: 16.84 on 18 and 3001 DF,  p-value: < 2.2e-16
```

```
anova(interactions0.lpm, test = "Chisq")
```

```
## Analysis of Variance Table
##
## Response: switch
##                 Df Sum Sq Mean Sq  F value    Pr(>F)
## arsenic          1  30.79 30.7886 137.8562 < 2.2e-16 ***
## dist             1  20.72 20.7235  92.7897 < 2.2e-16 ***
## assoc            1   0.67  0.6711   3.0048 0.0831213 .
## educ             3   8.39  2.7976  12.5262 3.877e-08 ***
## arsenic:dist     1   0.21  0.2059   0.9217 0.3370947
## arsenic:assoc    1   0.62  0.6176   2.7651 0.0964423 .
## arsenic:educ     3   1.25  0.4157   1.8613 0.1339473
## dist:assoc       1   0.17  0.1676   0.7504 0.3864274
## dist:educ        3   3.88  1.2924   5.7869 0.0006092 ***
## assoc:educ       3   1.01  0.3359   1.5040 0.2114626
## Residuals     3001 670.24  0.2233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# not impressive, only association interactions?
interactions1.lpm <- glm(switch~.+assoc*., data = ars.data)
summary(interactions1.lpm)
```

```
##
## Call:
## glm(formula = switch ~ . + assoc * ., data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0131  -0.4930   0.2312   0.4211   0.8860
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.6009456  0.0254366  23.625  < 2e-16 ***
## arsenic         0.2198471  0.0189653  11.592  < 2e-16 ***
## dist           -0.0023133  0.0002997  -7.718  1.6e-14 ***
## assoc           0.0035093  0.0395237   0.089 0.929256
## educ1          -0.0153595  0.0276389  -0.556 0.578444
## educ2           0.1090045  0.0295324   3.691 0.000227 ***
## educ3           0.1252669  0.0665917   1.881 0.060052 .
## arsenic:assoc -0.0555097  0.0295991  -1.875 0.060837 .
## dist:assoc      0.0004038  0.0004681   0.863 0.388362
## assoc:educ1    -0.0413967  0.0425200  -0.974 0.330342
## assoc:educ2    -0.0363324  0.0464161  -0.783 0.433833
## assoc:educ3    -0.2428133  0.1300431  -1.867 0.061974 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2246356)
##
##     Null deviance: 737.94  on 3019  degrees of freedom
## Residual deviance: 675.70  on 3008  degrees of freedom
## AIC: 4074.7
##
## Number of Fisher Scoring iterations: 2
```

```r
anova(interactions1.lpm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           3019     737.94
## arsenic        1  30.7886      3018     707.15 < 2.2e-16 ***
## dist           1  20.7235      3017     686.43 < 2.2e-16 ***
## assoc          1   0.6711      3016     685.75   0.08391 .
## educ           3   8.3927      3013     677.36 3.858e-08 ***
## arsenic:assoc  1   0.6257      3012     676.74  0.09512 .
## dist:assoc     1   0.1550      3011     676.58  0.40609
## assoc:educ     3   0.8768      3008     675.70  0.27211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Nah, just take the interaction between arsenic and distance:
interactions.lpm = glm(switch ~ arsenic + dist + educ + arsenic*dist, data = ars.data)
summary(interactions.lpm)
```

```
##
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic * dist,
##     data = ars.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9932  -0.4934   0.2287   0.4219   0.8258
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5967511  0.0203349  29.346  < 2e-16 ***
## arsenic        0.2161613  0.0232881   9.282  < 2e-16 ***
## dist          -0.0019841  0.0002981  -6.657 3.32e-11 ***
## educ1         -0.0338856  0.0210016  -1.613    0.107
## educ2          0.0942968  0.0227998   4.136 3.63e-05 ***
## educ3          0.0638483  0.0570453   1.119    0.263
## arsenic:dist  -0.0003947  0.0004038  -0.978    0.328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2249054)
##
##     Null deviance: 737.94  on 3019  degrees of freedom
## Residual deviance: 677.64  on 3013  degrees of freedom
## AIC: 4073.3
##
## Number of Fisher Scoring iterations: 2
```

```
anova(interactions.lpm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                          3019     737.94
## arsenic       1  30.7886      3018     707.15 < 2.2e-16 ***
## dist          1  20.7235      3017     686.43 < 2.2e-16 ***
## educ          3   8.5704      3014     677.85 2.683e-08 ***
## arsenic:dist  1   0.2150      3013     677.64    0.3283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#****************************************************


#*****************************************************************

# Prediction:

# Cross validation
n = nrow(ars.data)
n.15 = floor(n*.15)
n.iter = 100
n.methods = 6
accuracy = matrix(0, nrow = n.iter, ncol = n.methods)
deviance = matrix(0, nrow = n.iter, ncol = n.methods)
colnames(deviance) = c("Baseline","Baseline_no_assoc", "Our_model", "Our_model_with_assoc", "All_intera
colnames(accuracy) = c("Baseline","Baseline_no_assoc", "Our_model", "Our_model_with_assoc", "All_intera
for(i in 1:n.iter){
  test.index = sample(n,n.15)
  train.data = ars.data[-test.index,]
  test.data = ars.data[test.index,]
  # Baseline model
  ## Fit the model
  full.train.lm = glm(switch~., data= train.data, family = binomial())
  ## Predict using the full model
  full.fitted = predict(full.train.lm, newdata = test.data, type="response")
  ## Classifying the predictions as switched or not switched
  full.results = ifelse(full.fitted > .5, 1, 0)
  ## Calculating the Miscalculation rate
  misclass.full = mean(full.results != test.data$switch)
  ## Converting misclassification rate to accuracy. I did it this way because I thought misclassificati
  accuracy[i,1] = 1-misclass.full
  deviance[i,1] = summary(full.train.lm)$deviance

  # Taking out association out of the full model
  almost.train.lm = glm(switch~arsenic+dist+educ, data = train.data, family = binomial())
  almost.fitted = predict(almost.train.lm, newdata = test.data, type="response")
  almost.results = ifelse(almost.fitted > .5,1,0)
  misclass.almost = mean(almost.results != test.data$switch)
  accuracy[i,2] = 1-misclass.almost
  deviance[i,2] = summary(almost.train.lm)$deviance

  # With interactions = arsenic:education, educ:dist, arsenic:dist, I didn't included the association i
  interaction.lm = glm(switch~arsenic+dist+educ+arsenic:educ, data = train.data, family = binomial())
  interaction.fitted = predict(interaction.lm, newdata = test.data, type="response")
  interaction.results = ifelse(interaction.fitted > .5,1,0)
  misclass.interaction = mean(interaction.results != test.data$switch)
  accuracy[i,3] = 1 - misclass.interaction
  deviance[i,3] = summary(interaction.lm)$deviance

  # With interactions = arsenic:education, educ:dist, arsenic:dist, I didn't included the association i
  interaction1.lm = glm(switch~arsenic+dist+assoc+educ+arsenic:educ, data = train.data, family = binomi
  interaction1.fitted = predict(interaction1.lm, newdata = test.data, type="response")
  interaction1.results = ifelse(interaction1.fitted > .5,1,0)
  misclass.interaction1 = mean(interaction1.results != test.data$switch)
  accuracy[i,4] = 1 - misclass.interaction1
  deviance[i,4] = summary(interaction1.lm)$deviance


  # Without the arsenic and distance interaction because it didn't seem effective
  without.ars.dist.int = glm(switch~.*., data = train.data[,c(1,2,3,4,5)], family = binomial())
  without.fitted = predict(without.ars.dist.int, newdata = test.data[,c(1,2,3,4,5)], type="response")
  without.results = ifelse(without.fitted > .5, 1,0)
  missclass.without = mean(without.results != test.data$switch)
  accuracy[i,5] = 1 - missclass.without
  deviance[i,5] = summary(without.ars.dist.int)$deviance

  # Null stuff
  deviance[i,6] = summary(without.ars.dist.int)$null
  null.rate = table(train.data$switch)[2] / nrow(train.data)
  # Not sure about this next step
  accuracy[i,6] = null.rate


}

# Some analyses
apply(accuracy,2,mean)
```

```
##             Baseline    Baseline_no_assoc            Our_model
##            0.6225828            0.6227594            0.6265784
## Our_model_with_assoc     All_interactions                 null
##            0.6290949            0.6247682            0.5751811
```

```
apply(accuracy,2,sd)
```

```
##             Baseline    Baseline_no_assoc            Our_model
##          0.019931751          0.020757845          0.020057954
## Our_model_with_assoc     All_interactions                 null
##          0.020836152          0.021533208          0.004454931
```

```
apply(deviance,2,mean)
```

```
##             Baseline    Baseline_no_assoc            Our_model
##             3276.187             3278.347             3269.079
## Our_model_with_assoc     All_interactions                 null
##             3266.936             3243.211             3500.154
```

```
apply(deviance,2,sd)
```

```
##             Baseline    Baseline_no_assoc            Our_model
##            12.909415            12.964218            13.426843
## Our_model_with_assoc     All_interactions                 null
##            13.329284            13.419119             6.948123
```

```
# Deviance
summary(full.train.lm)$deviance
```

```
## [1] 3295.882
```

```r
summary(almost.train.lm)$deviance
```

```
## [1] 3297.334
```

```r
summary(interaction.lm)$deviance
```

```
## [1] 3288.209
```

```r
summary(without.ars.dist.int)$deviance
```

```
## [1] 3258.678
```

```r
# Summaries
summary(full.train.lm)
```

```
## 
## Call:
## glm(formula = switch ~ ., family = binomial(), data = train.data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.1106  -1.1665   0.7251   1.0376   1.8849  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.446989   0.099054   4.513  6.4e-06 ***
## arsenic      0.890121   0.075094  11.853  < 2e-16 ***
## dist        -0.009509   0.001144  -8.310  < 2e-16 ***
## assoc       -0.101571   0.084271  -1.205 0.228094    
## educ1       -0.105215   0.100073  -1.051 0.293083    
## educ2        0.427072   0.111277   3.838 0.000124 ***
## educ3        0.322285   0.270060   1.193 0.232720    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 3508.8  on 2566  degrees of freedom
## Residual deviance: 3295.9  on 2560  degrees of freedom
## AIC: 3309.9
## 
## Number of Fisher Scoring iterations: 4
```

```r
summary(almost.train.lm)
```

```
## 
## Call:
## glm(formula = switch ~ arsenic + dist + educ, family = binomial(), 
##     data = train.data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.0958  -1.1650   0.7284   1.0400   1.9066  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.406541   0.093141   4.365 1.27e-05 ***
## arsenic      0.892783   0.075052  11.896  < 2e-16 ***
## dist        -0.009512   0.001145  -8.310  < 2e-16 ***
## educ1       -0.112563   0.099878  -1.127 0.259737    
## educ2        0.424573   0.111238   3.817 0.000135 ***
## educ3        0.332991   0.269780   1.234 0.217088    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 3508.8  on 2566  degrees of freedom
## Residual deviance: 3297.3  on 2561  degrees of freedom
## AIC: 3309.3
## 
## Number of Fisher Scoring iterations: 4
```

```
summary(interaction.lm)
```

```
## 
## Call:
## glm(formula = switch ~ arsenic + dist + educ + arsenic:educ, 
##     family = binomial(), data = train.data)
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
## -2.178  -1.155   0.717   1.051   1.901  
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)    0.470284   0.097847   4.806 1.54e-06 ***
## arsenic        0.696581   0.127334   5.471 4.49e-08 ***
## dist          -0.009542   0.001150  -8.295  < 2e-16 ***
## educ1         -0.168165   0.111541  -1.508  0.13164    
## educ2          0.298004   0.121200   2.459  0.01394 *  
## educ3          0.363816   0.289074   1.259  0.20819    
## arsenic:educ1  0.172551   0.166499   1.036  0.30004    
## arsenic:educ2  0.552553   0.196186   2.816  0.00486 ** 
## arsenic:educ3 -0.247097   0.481966  -0.513  0.60817    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 3508.8  on 2566  degrees of freedom
## Residual deviance: 3288.2  on 2558  degrees of freedom
## AIC: 3306.2
## 
## Number of Fisher Scoring iterations: 4
```

```
summary(without.ars.dist.int)
```

```
## 
## Call:
## glm(formula = switch ~ . * ., family = binomial(), data = train.data[, 
##     c(1, 2, 3, 4, 5)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2010  -1.1532   0.7005   1.0368   2.2062
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.548951   0.153674   3.572 0.000354 ***
## arsenic        0.960475   0.176073   5.455 4.90e-08 ***
## dist          -0.011512   0.002681  -4.293 1.76e-05 ***
## assoc         -0.153719   0.190955  -0.805 0.420820
## educ1          0.015184   0.190584   0.080 0.936498
## educ2          0.026792   0.203135   0.132 0.895068
## educ3          0.328033   0.443901   0.739 0.459920
## arsenic:dist  -0.001967   0.002064  -0.953 0.340573
## arsenic:assoc -0.393676   0.154064  -2.555 0.010610 *
## arsenic:educ1  0.254740   0.180472   1.412 0.158091
## arsenic:educ2  0.393524   0.206786   1.903 0.057034 .
## arsenic:educ3 -0.429639   0.507310  -0.847 0.397053
## dist:assoc     0.005012   0.002408   2.081 0.037426 *
## dist:educ1    -0.003959   0.002941  -1.346 0.178268
## dist:educ2     0.007301   0.003105   2.352 0.018688 *
## dist:educ3     0.008062   0.006490   1.242 0.214197
## assoc:educ1   -0.063494   0.203304  -0.312 0.754806
## assoc:educ2   -0.143994   0.227071  -0.634 0.525991
## assoc:educ3   -1.116673   0.583381  -1.914 0.055602 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3508.8  on 2566  degrees of freedom
## Residual deviance: 3258.7  on 2548  degrees of freedom
## AIC: 3296.7
##
## Number of Fisher Scoring iterations: 4
```

```
# ANOVA analyses
anova(full.train.lm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     2566     3508.8
## arsenic  1  107.722      2565     3401.1 < 2.2e-16 ***
## dist     1   74.312      2564     3326.8 < 2.2e-16 ***
## assoc    1    1.978      2563     3324.8    0.1596
## educ     3   28.914      2560     3295.9 2.334e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(almost.train.lm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     2566     3508.8
## arsenic  1  107.722      2565     3401.1 < 2.2e-16 ***
## dist     1   74.312      2564     3326.8 < 2.2e-16 ***
## educ     3   29.440      2561     3297.3  1.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(interaction.lm, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                          2566     3508.8
## arsenic       1  107.722      2565     3401.1 < 2.2e-16 ***
## dist          1   74.312      2564     3326.8 < 2.2e-16 ***
## educ          3   29.440      2561     3297.3  1.81e-06 ***
## arsenic:educ  3    9.125      2558     3288.2   0.02767 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(without.ars.dist.int, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: switch
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           2566     3508.8
## arsenic        1  107.722      2565     3401.1 < 2.2e-16 ***
## dist           1   74.312      2564     3326.8 < 2.2e-16 ***
## assoc          1    1.978      2563     3324.8 0.1596433    
## educ           3   28.914      2560     3295.9 2.334e-06 ***
## arsenic:dist   1    1.573      2559     3294.3 0.2097642    
## arsenic:assoc  1    3.514      2558     3290.8 0.0608355 .  
## arsenic:educ   3    9.074      2555     3281.7 0.0283253 *  
## dist:assoc     1    2.783      2554     3278.9 0.0952625 .  
## dist:educ      3   16.370      2551     3262.6 0.0009519 ***
## assoc:educ     3    3.889      2548     3258.7 0.2737368    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# These are some more tests we could do to validate the results,
# but they might not be necessary if we end up doing the logit, probit, and lpm

# ROC curve

# Confusion matrices
#fitted.results = predict(full.train.log)
```