Using OCT Biomarkers to Predict Visual Acuity after One Year of VEGF Therapy

Brian Dang and Ethan Mai

12/6/2021

```
AMD <- as.data.frame(read.csv("~/Downloads/dataframev2.csv", header = TRUE, stringsAsFactors = FALSE))
library(ggplot2)
library(GGally)
library(knitr)
library(car)
library(MASS)
library(lmtest)
```

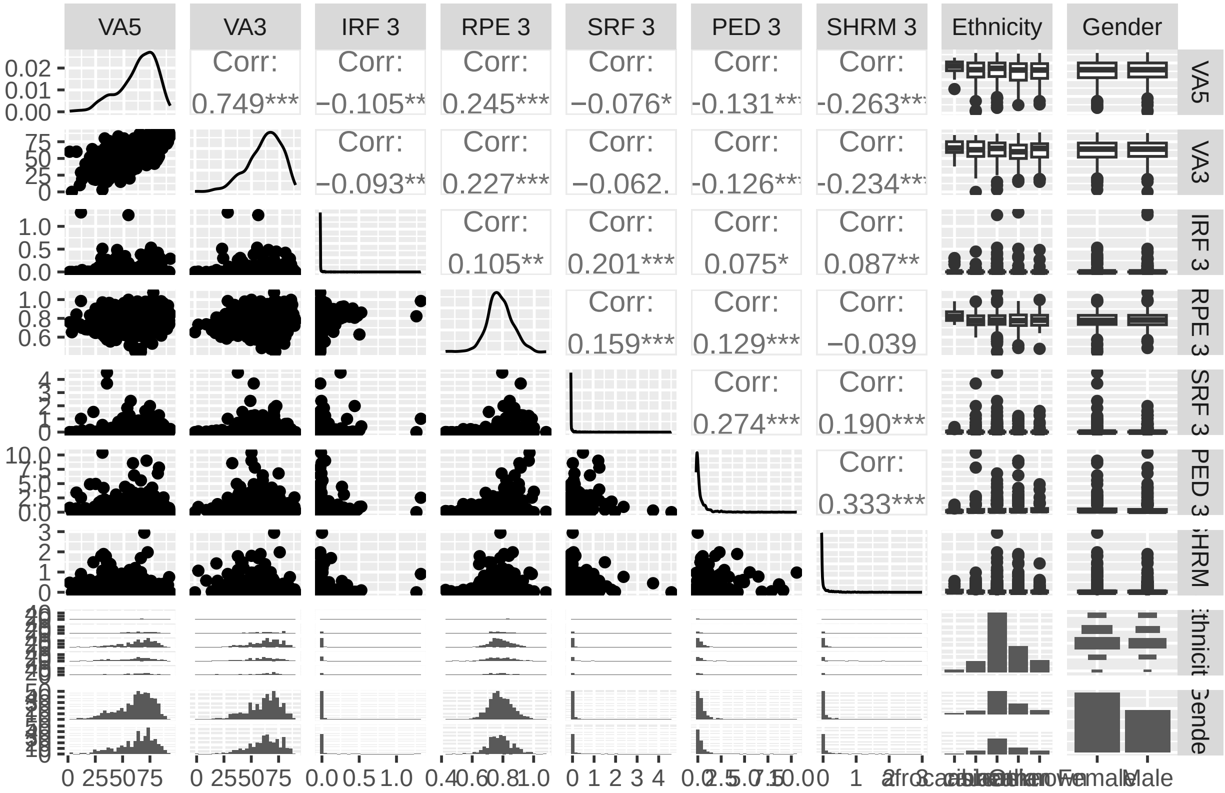
Table of Contents:

Note that appropriate data transformations were applied after looking at initial model fits (i.e. transformations were applied in the 'Model 2' and 'Model 3' sections).

1. Model 1

```
ggpairs(AMD,columns = c(10,8,13,18,23,28,33,3,5), columnLabels = c("VA5","VA3","IRF 3", "RPE 3","SRF 3"
```

Fig 1. Pairs plot of our untransformed response and predictors



```
model1 <- lm(va3~vol_irf3 + vol_rpe3 + vol_srf3+ vol_ped3+vol_shrm3 + ethnicity+gender, data = AMD)
```

```
tab_1 <- data.frame(round(summary(model1)$coefficients,4))
kable(tab_1, caption = "OLS Coefficient Estimates for Model 1, $adj R^2=0.1149$", col.names = c("Estima
```

Table 1: OLS Coefficient Estimates for Model 1, $adj R^2 = 0.1149$

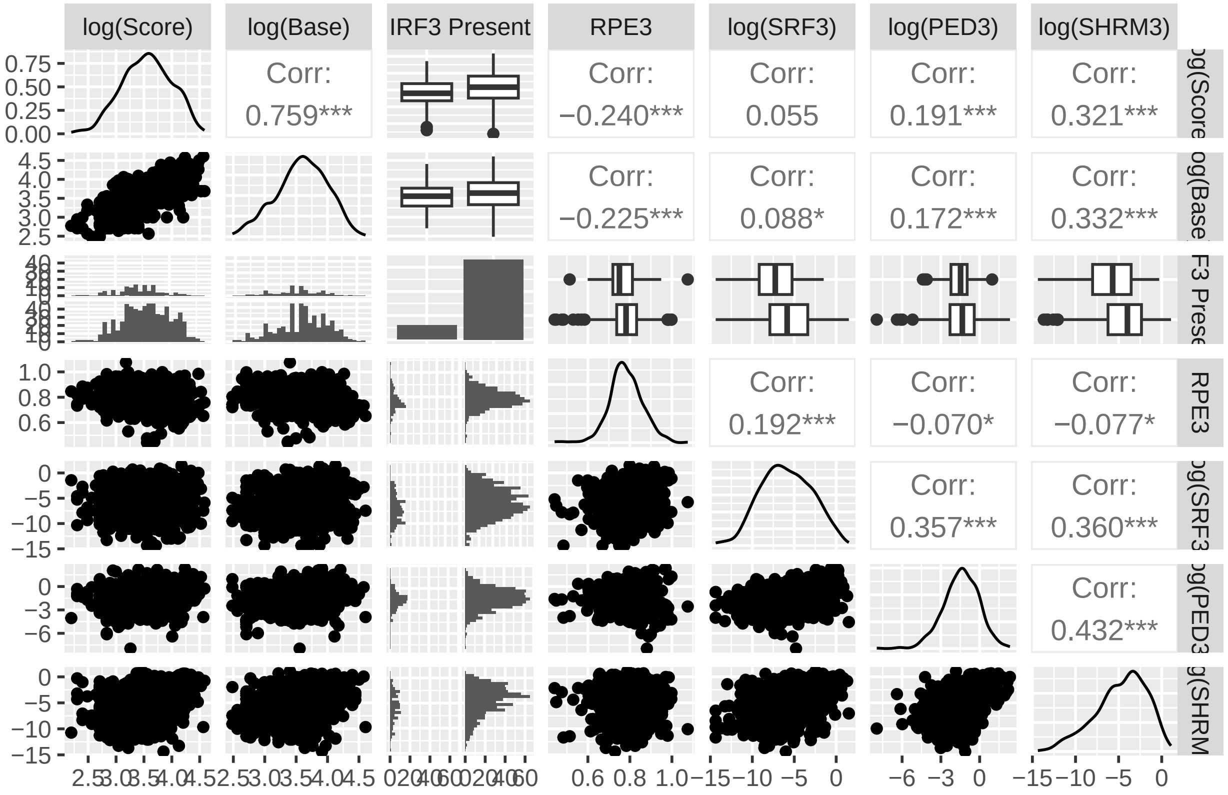
	Estimate	Std. Error	t value	P(>|t|)
(Intercept)	30.0106	5.7460	5.2229	0.0000
vol_irf3	-18.1565	6.1956	-2.9306	0.0035
vol_rpe3	45.6047	6.0287	7.5646	0.0000
vol_srf3	-1.4079	1.7843	-0.7890	0.4303
vol_ped3	-1.2780	0.5620	-2.2739	0.0232
vol_shrm3	-11.0385	2.0284	-5.4420	0.0000
ethnicityasian	-2.2534	3.2704	-0.6890	0.4910
ethnicitycaucasian	-1.4482	2.9783	-0.4863	0.6269
ethnicityOther	-3.9451	3.0666	-1.2865	0.1986
ethnicityunknown	-2.9883	3.2381	-0.9229	0.3563
genderMale	0.4665	0.9505	0.4908	0.6237

2. Model 2

```
AMD$score <- 100-AMD$va5
AMD$base <-100-AMD$va3
AMD <- transform(AMD, lscore = log(score), lbase=log(base), lsrf3=log(vol_srf3),lped3=log(vol_ped3), lsh
ind <- which(AMD$vol_srf3 < 0.0000001)
AMD.sub <- AMD[-ind,]
ind2<- which(AMD.sub$vol_shrm3 < 0.0000001)
AMD.sub <- AMD.sub[-ind2,]
AMD.sub$irf3 <- factor(ifelse(AMD.sub$vol_irf3 < 0.0000001, "No","Yes"))
model2 <- lm(lscore~ lbase+as.factor(irf3) + vol_rpe3 + lsrf3+ lped3+lshrm3,data = AMD.sub)
```

```
ggpairs(AMD.sub,columns = c(44,45,49,18,46,47,48), columnLabels = c("log(Score)","log(Base)","IRF3 Pres
```

Fig 2. Pairs plot of our transformed response and predictors



```
tab_2 <- data.frame(round(summary(model2)$coefficients,4))
kable(tab_2, caption = "OLS Coefficient Estimates for Model 2, $adj R^2=0.5907$", col.names = c("Estima
```

Table 2: OLS Coefficient Estimates for Model 2, $adj R^2 = 0.5907$

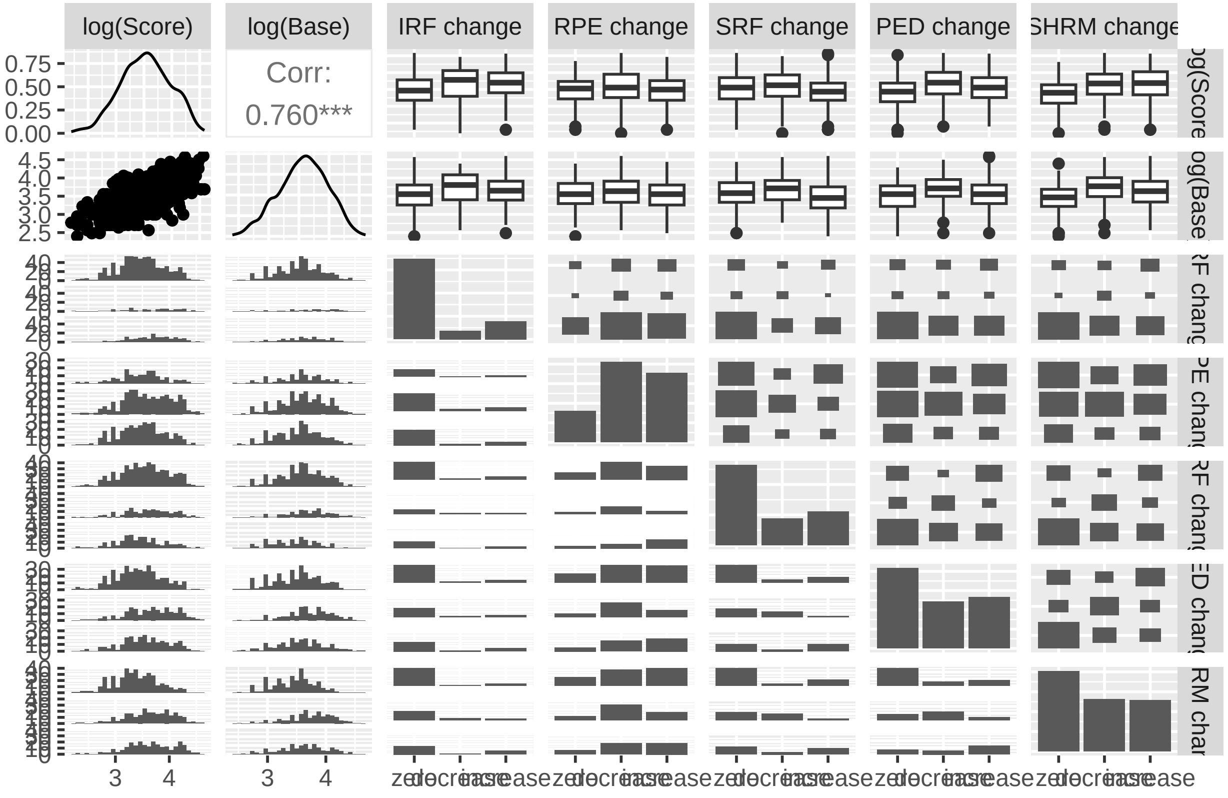
	Estimate	Std. Error	t value	P(>|t|)
(Intercept)	1.0103	0.1660	6.0847	0.0000
lbase	0.7793	0.0271	28.7220	0.0000
as.factor(irf3)Yes	0.0866	0.0289	3.0009	0.0028
vol_rpe3	-0.3861	0.1358	-2.8437	0.0046
lsrf3	-0.0067	0.0038	-1.7364	0.0829
lped3	0.0161	0.0085	1.8918	0.0589
lshrm3	0.0093	0.0041	2.2431	0.0252

3. Model 3 (Final Model):

```
AMD$irf_change <- round(AMD$vol_irf5-AMD$vol_irf3,4)
AMD$rpe_change <- round(AMD$vol_rpe5-AMD$vol_rpe3,4)
AMD$srf_change <- round(AMD$vol_srf5-AMD$vol_srf3,4)
AMD$ped_change <- round(AMD$vol_ped5-AMD$vol_ped3,4)
AMD$shrm_change <- round(AMD$vol_shrm5-AMD$vol_shrm3,4)

AMD$irfcat <-cut(AMD$irf_change, breaks = c(-Inf,-.005,.005,Inf), labels = c("decrease","zero","increase
AMD$rpecat <-cut(AMD$rpe_change, breaks = c(-Inf,-.01,.01,Inf), labels = c("decrease","zero","increase")
AMD$srfcat <-cut(AMD$srf_change, breaks = c(-Inf,-.01,.01,Inf), labels = c("decrease","zero","increase")
AMD$pedcat <-cut(AMD$ped_change, breaks = c(-Inf,-.05,.05,Inf), labels = c("decrease","zero","increase")
AMD$shrmcat <-cut(AMD$shrm_change, breaks = c(-Inf,-.005,.005,Inf), labels = c("decrease","zero","increa
AMD$irfcat<- relevel(AMD$irfcat,"zero")
AMD$rpecat<- relevel(AMD$rpecat,"zero")
AMD$srfcat<- relevel(AMD$srfcat,"zero")
AMD$pedcat<- relevel(AMD$pedcat,"zero")
AMD$shrmcat<- relevel(AMD$shrmcat,"zero")
```

```
ggpairs(AMD,columns = c(44,45,54,55,56,57,58), columnLabels = c("log(Score)","log(Base)","IRF change",
```

Fig 3 Pairs plot of our response and predictos after factorizing the OCT biom



```
model3 <- lm(lscore~lbase+irfcat+rpecat+srfcat+pedcat+shrmcat,data = AMD)
tab_3 <- data.frame(round(summary(model3)$coefficients,4))
kable(tab_3, caption = "OLS Coefficient Estimates for Model 3, $adj R^2=0.6076$", col.names = c("Estima
```

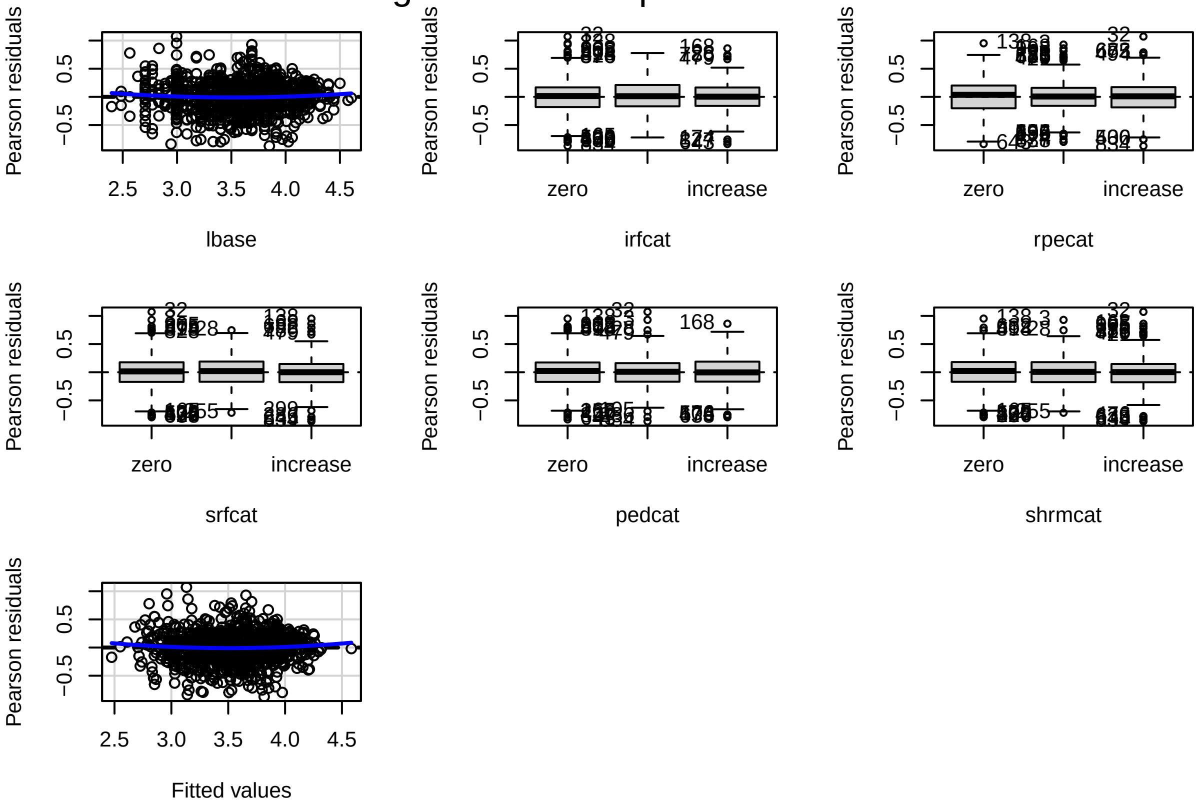
Table 3: OLS Coefficient Estimates for Model 3, $adjR^2 = 0.6076$

	Estimate	Std. Error	t value	P(>|t|)
(Intercept)	0.5404	0.0879	6.1495	0.0000
lbase	0.8107	0.0245	33.1292	0.0000
irfcatdecrease	0.0673	0.0350	1.9219	0.0549
irfcatincrease	0.1096	0.0253	4.3259	0.0000
rpecatdecrease	0.0621	0.0261	2.3794	0.0175
rpecatincrease	0.0048	0.0266	0.1809	0.8565
srfcatdecrease	-0.0654	0.0261	-2.5118	0.0122
srfcatincrease	-0.0094	0.0239	-0.3932	0.6942
pedcatdecrease	0.0463	0.0244	1.8943	0.0585
pedcatincrease	0.0339	0.0236	1.4348	0.1517
shrmcatdecrease	0.0166	0.0252	0.6605	0.5091
shrmcatincrease	0.1126	0.0244	4.6225	0.0000

4. Model Diagnostics on Model 3

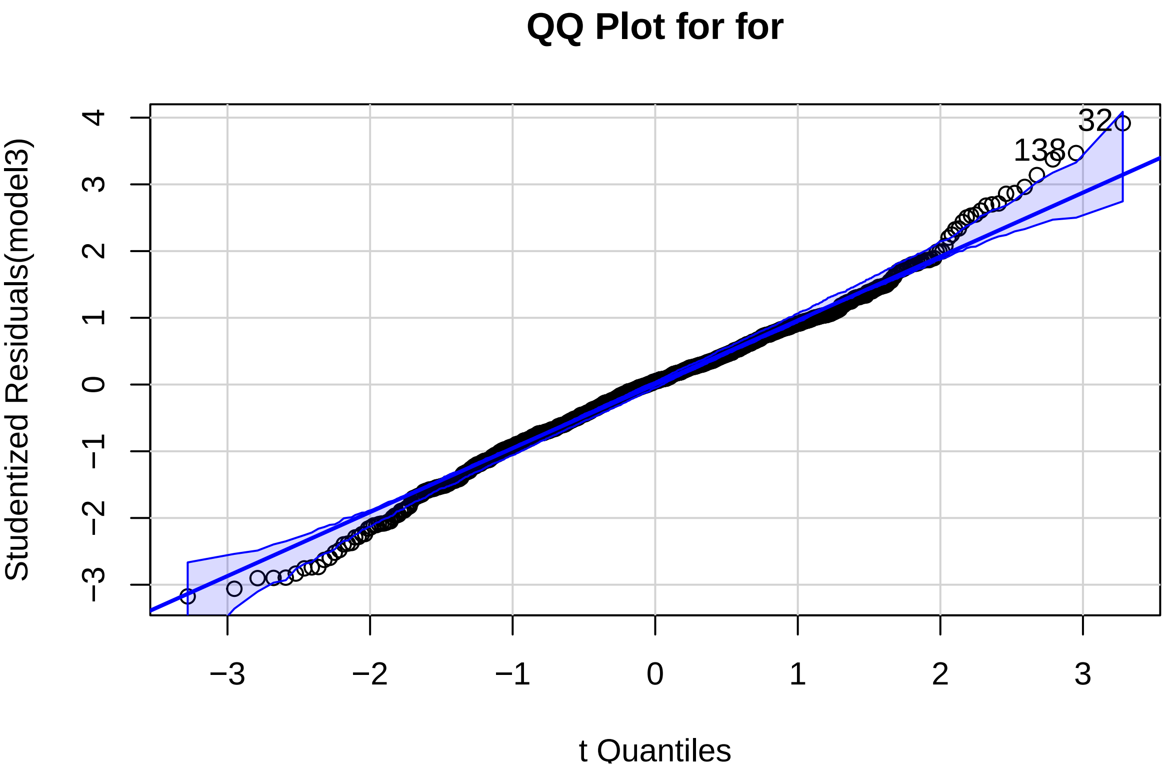
```
residualPlots(model3,main="Fig 4. Residuals plot of Model 3")
```

Fig 4. Residuals plot of Model 3



```
##              Test stat Pr(>|Test stat|)
## lbase           1.3768           0.1689
## irfcat
## rpecat
## srfcat
## pedcat
## shrmcat
## Tukey test    1.4106           0.1584
```

```
qqPlot(model3,main = "QQ Plot for for")
```



```
## [1]  32 138
```

```
ncvTest(model3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 9.647212, Df = 1, p = 0.0018964
```

5. ANOVA

```
model31 <- lm(lscore~lbase,data = AMD)
anova(model31,model3)
```

```
## Analysis of Variance Table
##
## Model 1: lscore ~ lbase
## Model 2: lscore ~ lbase + irfcat + rpecat + srfcat + pedcat + shrmcat
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    924 76.920
## 2    914 70.645 10    6.2748 8.1183 1.174e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(model3)
```

```
## Anova Table (Type II tests)
```

```
##
## Response: lscore
##            Sum Sq  Df   F value    Pr(>F)    
## lbase     84.832   1 1097.5448 < 2.2e-16 ***
## irfcat     1.577   2   10.2009 4.155e-05 ***
## rpecat     0.746   2    4.8277   0.00821 ** 
## srfcat     0.488   2    3.1595   0.04291 *  
## pedcat     0.326   2    2.1066   0.12224    
## shrmcat    1.824   2   11.7965 8.747e-06 ***
## Residuals 70.645 914                        
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6. Variable Selection

```
stepAIC(model3, direction = "backward")
```

```
## Start:  AIC=-2358.79
## lscore ~ lbase + irfcat + rpecat + srfcat + pedcat + shrmcat
## 
##           Df Sum of Sq     RSS     AIC
## <none>                  70.645 -2358.8
## - pedcat   2     0.326  70.971 -2358.5
## - srfcat   2     0.488  71.134 -2356.4
## - rpecat   2     0.746  71.391 -2353.1
## - irfcat   2     1.577  72.222 -2342.3
## - shrmcat  2     1.824  72.469 -2339.2
## - lbase    1    84.832 155.477 -1630.3
```

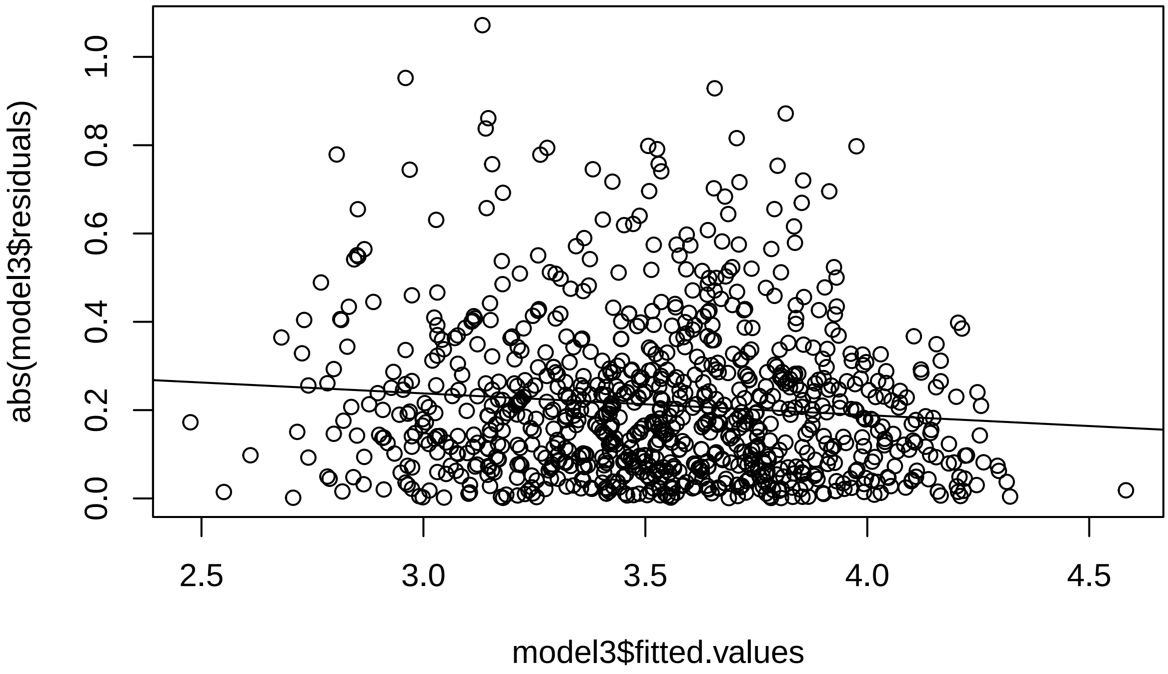
```
## 
## Call:
## lm(formula = lscore ~ lbase + irfcat + rpecat + srfcat + pedcat + 
##     shrmcat, data = AMD)
## 
## Coefficients:
##     (Intercept)             lbase    irfcatdecrease    irfcatincrease  
##        0.540425          0.810726          0.067257          0.109575  
##  rpecatdecrease    rpecatincrease    srfcatdecrease    srfcatincrease  
##        0.062142          0.004817         -0.065444         -0.009389  
##  pedcatdecrease    pedcatincrease   shrmcatdecrease   shrmcatincrease  
##        0.046310          0.033859          0.016629          0.112622
```

7. Addressing Heteroscedasticity

7.a. Weighted Least Squares

7.b. Choosing Weights

```
emodel<- lm(abs(model3$residuals)~model3$fitted.values)
plot(abs(model3$residuals) ~ model3$fitted.values)
abline(emodel)
```



7.c. Fitting WLS

```
wt <- 1/emodel$fitted.values^2
wmodel3 <- lm(log(score)~log(base)+irfcat+rpecat+srfcat+pedcat+shrmcat,data = AMD,weights = wt)
tab_3w <- data.frame(round(summary(wmodel3)$coefficients,4))
kable(tab_3w, caption = "Weighted Least Squares Estimates for Model 3", col.names = c("Estimate","Std. 
```

Table 4: Weighted Least Squares Estimates for Model 3

	Estimate	Std. Error	t value	P(>|t|)
(Intercept)	0.5222	0.0892	5.8546	0.0000
log(base)	0.8146	0.0246	33.1785	0.0000
irfcatdecrease	0.0699	0.0336	2.0820	0.0376
irfcatincrease	0.1060	0.0244	4.3534	0.0000
rpecatdecrease	0.0703	0.0259	2.7167	0.0067
rpecatincrease	0.0083	0.0266	0.3101	0.7565
srfcatdecrease	-0.0643	0.0253	-2.5375	0.0113
srfcatincrease	-0.0129	0.0238	-0.5430	0.5873
pedcatdecrease	0.0488	0.0239	2.0431	0.0413
pedcatincrease	0.0330	0.0234	1.4096	0.1590
shrmcatdecrease	0.0148	0.0249	0.5947	0.5522
shrmcatincrease	0.1143	0.0241	4.7460	0.0000

```
ncvTest(wmodel3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.01799714, Df = 1, p = 0.89328
```

7.d. Sandwich Estimator

```
model3.sandwich = coeftest(model3,vcov. = hccm(model3, type="hc3"))
sandwich.se = round(sqrt(diag(hccm(model3, type="hc3"))), 4)
coef_est = data.frame(round(cbind(coef(model3),coef(wmodel3),model3.sandwich[,1]),4))
se_est = data.frame(round(cbind(sqrt(diag(vcov(model3))),sqrt(diag(vcov(wmodel3))),sandwich.se),4))
colnames(coef_est) = colnames(se_est) = c("OLS","WLS","Sandwich")
se_est2 = data.frame(round(cbind(sqrt(diag(vcov(model3))),sqrt(diag(vcov(wmodel3)))),4))

kable(se_est2, caption = "Comparison of SE for OLS and WLS estimation for Model 3", col.names = c("OLS"
```

Table 5: Comparison of SE for OLS and WLS estimation for Model 3

	OLS	WLS
(Intercept)	0.0879	0.0892
lbase	0.0245	0.0246
irfcatdecrease	0.0350	0.0336
irfcatincrease	0.0253	0.0244
rpecatdecrease	0.0261	0.0259
rpecatincrease	0.0266	0.0266
srfcatdecrease	0.0261	0.0253
srfcatincrease	0.0239	0.0238
pedcatdecrease	0.0244	0.0239
pedcatincrease	0.0236	0.0234
shrmcatdecrease	0.0252	0.0249
shrmcatincrease	0.0244	0.0241

7.e. Bootstrap

```
set.seed(123)
nboot = 1000
n=nrow(AMD)
dat.boot <- matrix(0,nboot,12)
for (i in 1:nboot){
  indices <- sample(seq(1,n), replace=T)
  m.boot <- lm(log(score)~log(base)+irfcat+rpecat+srfcat+pedcat+shrmcat,data = AMD[indices,])
  dat.boot[i,] <- m.boot$coefficients
}
mean.boot <- round(apply(dat.boot, 2, mean),4)
se.boot<-round(apply(dat.boot, 2, sd),4)
coef_est$Bootstrap = mean.boot
se_est$Bootstrap = se.boot
```

```
print(se_est)
```

```
##                    OLS    WLS Sandwich Bootstrap
## (Intercept)     0.0879 0.0892   0.0927    0.0922
## lbase           0.0245 0.0246   0.0257    0.0258
## irfcatdecrease  0.0350 0.0336   0.0344    0.0331
## irfcatincrease  0.0253 0.0244   0.0260    0.0261
## rpecatdecrease  0.0261 0.0259   0.0271    0.0264
## rpecatincrease  0.0266 0.0266   0.0277    0.0271
## srfcatdecrease  0.0261 0.0253   0.0263    0.0265
## srfcatincrease  0.0239 0.0238   0.0244    0.0248
## pedcatdecrease  0.0244 0.0239   0.0249    0.0246
## pedcatincrease  0.0236 0.0234   0.0234    0.0228
## shrmcatdecrease 0.0252 0.0249   0.0244    0.0238
## shrmcatincrease 0.0244 0.0241   0.0256    0.0264
```

8. Cross Validation

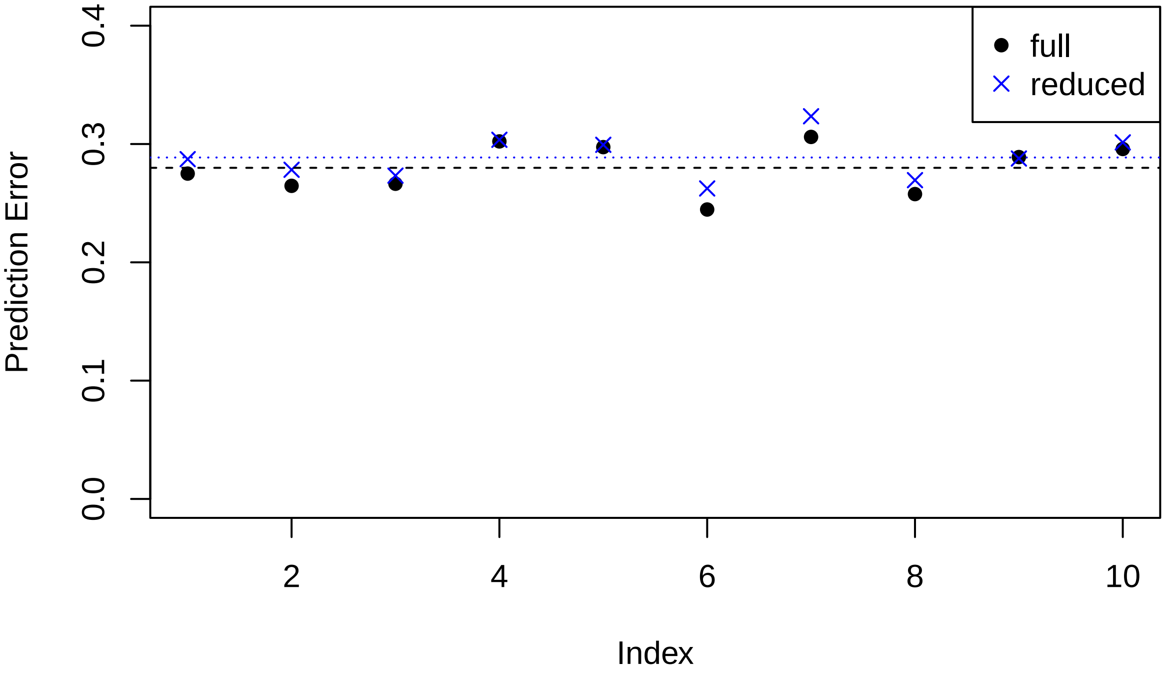
```r
k <- 10
n.obs <- nrow(AMD)
random.order <- sample(seq(1,n.obs),n.obs, replace=F)
cv.size <- rep(floor(n.obs/k),k)
cum.size <- c(0, cumsum(cv.size[-length(cv.size)]))
error.reduced <- rep(0,k)
error.full <- rep(0,k)
for (i in 1:k) {
  inds <- (cum.size[i]+1):(cum.size[i]+cv.size[i])
  AMD.subset <- AMD[-random.order[inds],]
  lm.reduced <- lm(lscore ~ log(base), data=AMD.subset)
  lm.full <- lm(lscore~lbase+irfcat+rpecat+srfcat+pedcat+shrmcat,data = AMD.subset)
  test.data <- AMD[random.order[inds],]
  predict.full <- predict.lm(lm.full,test.data)
  error.full[i] <- sqrt(mean((test.data$lscore - predict.full)^2))
  predict.reduced <- predict.lm(lm.reduced,test.data)
  error.reduced[i] <- sqrt(mean((test.data$lscore - predict.reduced)^2))
}

RMSPE.full = mean(error.full)
RMSPE.reduced = mean(error.reduced)
RMSPE.full
```

```
## [1] 0.279869
```

```r
RMSPE.reduced
```

```
## [1] 0.2886124
```

```r
plot(error.full, ylim=c(0,0.4), pch=16,ylab="Prediction Error")
points(error.reduced, col='blue', pch=4) #generally black points are below blue points
abline(h=mean(error.full), lty=2, col='black')
abline(h=mean(error.reduced), lty=3, col='blue')
legend('topright', legend=c('full','reduced'), pch=c(16,4), col=c('black','blue'))
```


0.0
0.1
0.2
0.3
0.4
Prediction Error
2
4
6
8
10
Index
● full
× reduced