STAT 526 HW 1

STAT 526 HW 1

Problem 0

Name

Bowen Zheng

Problem 1

A multiple regression, involving 88 cases and 6 predictors, resulted in an R^2 = 0.48. Based on this information, what is the F statistic, its degrees of freedom, and P-value?

We have 6 predictors with n = 88. The degree of freedom is $df_1 = 6$. The degree of freedom error is $df_2 = 88 - (6 + 1) = 81$.

We can calculate our F-statistics with R squared.

$$F = \frac{R^2/df_1}{(1 - R^2)/df_2} = \frac{0.48/6)}{(1 - 0.48)/81} = 12.46$$

Our F-statistic is 12.46 with 6 and 81 degrees of freedom. We can use a calculate and find that the corresponding P-value is approximately 0.

Problem 2

In Slide 18 of Topic 1, an F test is described to compare a more flexible model with a reduced one (e.g., some parameters in the reduced model are set to 0). When a multiple regression has replications at certain sets of X, a lack of fit test can be performed using this same framework. The full model does not put any assumptions on the means for each set of X. The reduced model assumes the means are a linear function of X. Assuming that there are C sets of X in the data set and Set i has n_i observations (n_i > 1 for some sets), write out the lack-of-fit test in this framework by specifying the full and reduced models, their degrees of freedom, and the associated error degrees of freedom. This total number of observations is n = $\sum_{i=1}^{C} n_i$.

The full model:

Each set of C has their own mean. We have C sets of X.

Our equation is $ Y_{ij} = *i* + {ij}$ where $i \in 1, 2, \ldots, C$.

This means we have C number of parameters and our degree of freedom is C and our degree freedom of error is $n - C$.

The reduced model:

We now have a linear relation / model for the means: $Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \cdots + \beta_k x_{ijk} + \epsilon_{ij}$.

With k features from the data, we have $k+1$ parameters used in our model. The degree of freedom is $k+1$ with a degree of freedom error of $n-k-1$.

When performing the test

Using the two models we can compute things like SSE and SSM and ultimately, the difference in their degrees of freedom is $(n-k-1)-(n-C)=C-k-1$, which will be used when performing the lack of fit test.

Problem 3

The analysis of the Georgia data set in Topic 1 did not account for the fact that the response variable (proportion undercount) may have nonconstant variances. Perform a weighted regression fit of the final model on Slide 42 assuming the binomial setting for the proportion undercount and summarize the results. Also compare these results with those shown in Topic 1.

We first grab the code from topic 1 so we can have access to the final model on slide 42.

```
library(faraway)
data(gavote)

### Lists the first 6 datalines of the data set
head(gavote)

### Summarizes the structure of the data set
str(gavote)

### Get summary statistics for each of the variables
summary(gavote)

### Because number of votes highly skewed, will look at percent undercount
percunder <- (gavote$ballots - gavote$votes)/gavote$ballots

### Generate histogram of percent undercount
hist(percunder,xlab="Percent",las=1,main="Undercount")
```

```
### Generate density with data shown at bottom
plot(density(percunder),main="Percent Undercount",las=1)
rug(percunder)
```

```
### Define new percent variables for Gore and Bush votes
pergore = gavote$gore/gavote$votes
perbush = gavote$bush/gavote$votes

### Generate a scatterplot matrix of numeric variables
pairs(~percunder+gavote$perAA+pergore+perbush,pch=20)
```

```
### Generate side-by-side boxplots
plot(percunder~rural,gavote,las=1,ylab="Percent")
```

```
plot(percunder~equip,gavote,las=1)
```

```
###Fitting a linear model using the lm function
model1 = lm(percunder ~ pergore + perAA, gavote)

###Obtain summary information from model fit
summary(model1)

###Reduced summary information function proposed by Faraway
sumary(model1)

###Requesting ANOVA Table but be wary this is using Type I SS
anova(model1)

### This library contains function allowing Type III SS.  Be wary
### of its use too.  Often need to change options using
options(contrasts = c("contr.sum", "contr.poly"))

library(car)
Anova(model1, type=3)

### Generate a 2x2 panel of diagnostic plots
par(mar=c(2,2,2,2),mfrow=c(2,2))
plot(model1,cex=0.65,cex.axis=0.7,cex.lab=0.5)
```

```
### Creating centered variables.  Can help with multicolinearity when
### considering polynomials and interactions
cpergore = pergore - mean(pergore)
cperAA = gavote$perAA - mean(gavote$perAA)

### Fitting alternative model
model2 = lm(percunder ~ cperAA+cpergore*rural+equip, gavote)
sumary(model2)

### General linear test comparing the two models
anova(model1,model2)

### Consider dropping single predictors.  Again performing general linear test
drop1(model2,test="F")

### New model dropping insignificant terms from previous function
model3 = lm(percunder ~ cpergore+rural+equip, gavote)
sumary(model3)
anova(model2,model3)

### Defining maximum model from which to select from
modelmax = lm(percunder ~ (equip+econ+rural+atlanta)^2 + (equip+econ+rural+atlanta)*(pergore+per
AA), gavote)

### Using AIC to reduce model
modelbest = step(modelmax,trace=FALSE)
summary(modelbest)

drop1(modelbest,test="F")

### Determing best model

modelbetter2 = lm(percunder ~ equip+econ+rural+perAA+equip:econ+equip:perAA, gavote)
drop1(modelbetter2,test="F")
sumary(modelbetter2)

### Getting tables of predictions
pdf <- data.frame(econ=rep(levels(gavote$econ),5),equip=rep(levels(gavote$equip), rep(3,5)), per
AA=0.233, rural="rural")
ppr = predict(modelbetter2,new=pdf)
```

```
## Warning in predict.lm(modelbetter2, new = pdf): prediction from rank-deficient
## fit; attr(*, "non-estim") has doubtful cases
```

```
xtabs(round(ppr,3)~econ+equip,pdf)

pdf <- data.frame(econ=rep(levels(gavote$econ),5),equip=rep(levels(gavote$equip), rep(3,5)), per
AA=0.233, rural="urban")
ppu = predict(modelbetter2,new=pdf)
```

```
## Warning in predict.lm(modelbetter2, new = pdf): prediction from rank-deficient
## fit; attr(*, "non-estim") has doubtful cases
```

```
xtabs(round(ppu,3)~econ+equip,pdf)
```

OK now let us update the model with the weights being the count of the ballots. If we are assuming binomial with variance $Var(p) = \frac{p(1-p)}{n}$ and use the $w \propto \frac{1}{\sigma^2}$ then the weight of the ballots should be inversely proportional to the variance. Given that we do not know the true proportion of undercounting, the weight then is simply just n, the number of ballots.

```
model_wls <- update(modelbetter2, weights = ballots, data = gavote)
summary(model_wls)
```

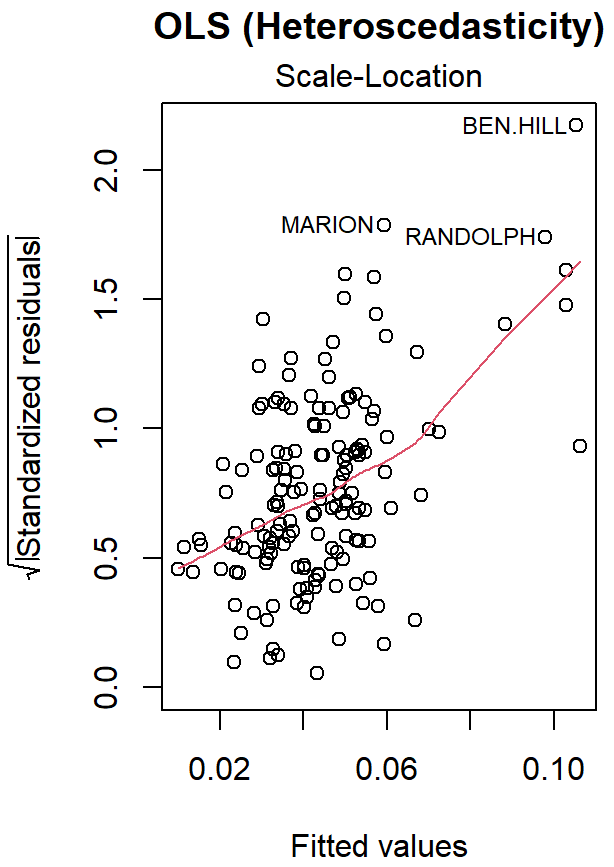
```
##
## Call:
## lm(formula = percunder ~ equip + econ + rural + perAA + equip:econ +
##     equip:perAA, data = gavote, weights = ballots)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8195 -0.8363 -0.0707  0.8911  7.3157
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.035228   0.014556   2.420  0.01680 *
## equip1        0.006031   0.015292   0.394  0.69389
## equip2       -0.005891   0.015045  -0.392  0.69597
## equip3        0.020433   0.015975   1.279  0.20301
## equip4       -0.056013   0.057479  -0.974  0.33149
## econ1         0.002216   0.002349   0.943  0.34705
## econ2         0.020012   0.003753   5.332 3.80e-07 ***
## rural1        0.004483   0.001780   2.519  0.01290 *
## perAA        -0.004733   0.040301  -0.117  0.90668
## equip1:econ1 -0.002237   0.004207  -0.532  0.59568
## equip2:econ1 -0.001931   0.003324  -0.581  0.56215
## equip3:econ1 -0.009352   0.003480  -2.687  0.00808 **
## equip4:econ1        NA         NA      NA       NA
## equip1:econ2 -0.004732   0.005244  -0.902  0.36838
## equip2:econ2 -0.011625   0.004983  -2.333  0.02108 *
## equip3:econ2  0.024526   0.005924   4.140 5.97e-05 ***
## equip4:econ2        NA         NA      NA       NA
## equip1:perAA -0.038067   0.043786  -0.869  0.38612
## equip2:perAA  0.055830   0.043329   1.289  0.19969
## equip3:perAA -0.041768   0.045311  -0.922  0.35822
## equip4:perAA  0.091829   0.156455   0.587  0.55819
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.765 on 140 degrees of freedom
## Multiple R-squared:  0.6388, Adjusted R-squared:  0.5924
## F-statistic: 13.76 on 18 and 140 DF,  p-value: < 2.2e-16
```

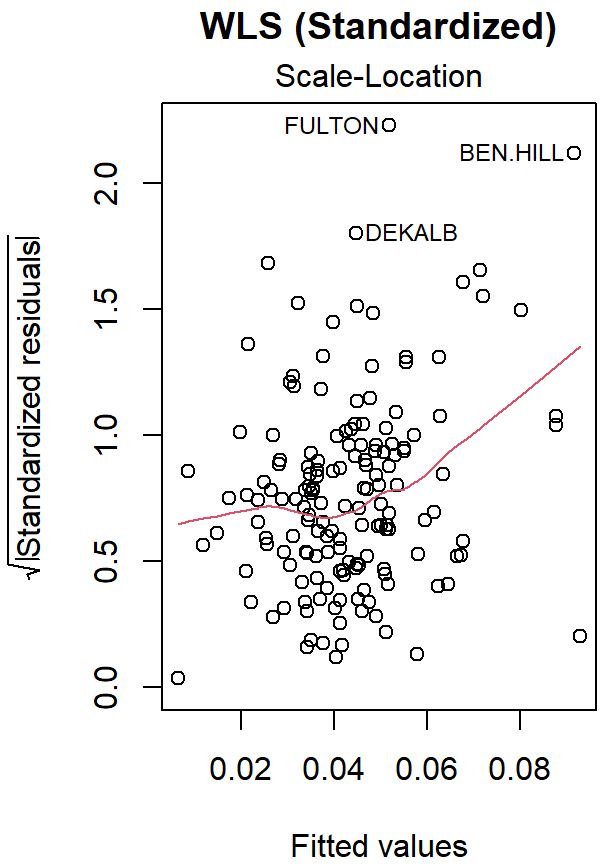
```
par(mfrow=c(1,2))
plot(modelbetter2, which = 3, main = "OLS (Heteroscedasticity)")
```

```
## Warning: not plotting observations with leverage one:
##   103, 131
```

```
plot(model_wls, which = 3, main = "WLS (Standardized)")
```

```
## Warning: not plotting observations with leverage one:
##   103, 131
```






```
### Getting tables of predictions
pdf <- data.frame(econ=rep(levels(gavote$econ),5),equip=rep(levels(gavote$equip), rep(3,5)), per
AA=0.233, rural="rural")
ppr = predict(model_wls,new=pdf)
```

```
## Warning in predict.lm(model_wls, new = pdf): prediction from rank-deficient
## fit; attr(*, "non-estim") has doubtful cases
```

```
xtabs(round(ppr,3)~econ+equip,pdf)
```

```
##         equip
## econ      LEVER  OS-CC  OS-PC  PAPER  PUNCH
##   middle  0.036  0.046  0.042  0.006  0.074
##   poor    0.051  0.054  0.094  0.024  0.070
##   rich    0.021  0.037  0.012 -0.018  0.031
```

```
pdf <- data.frame(econ=rep(levels(gavote$econ),5),equip=rep(levels(gavote$equip), rep(3,5)), per
AA=0.233, rural="urban")
ppu = predict(model_wls,new=pdf)
```

```
## Warning in predict.lm(model_wls, new = pdf): prediction from rank-deficient
## fit; attr(*, "non-estim") has doubtful cases
```

```
xtabs(round(ppu,3)~econ+equip,pdf)
```

```
##         equip
## econ      LEVER  OS-CC  OS-PC  PAPER  PUNCH
##   middle  0.027  0.037  0.033 -0.003  0.065
##   poor    0.042  0.045  0.085  0.015  0.061
##   rich    0.012  0.028  0.003 -0.027  0.022
```

We can finally view our results. They are indeed quite similar to what we saw in topic one. The predicted percent under count is nearly identical (at most off by less than one percent) to our OLS model. However, it seems that the variables in our model have changed in significance slightly. In fact, we seem to have lost some variables like equip4:econ2.

We can also do this simply in GLM with binomial family:

```
model_binomial <- glm(cbind(ballots - votes, votes) ~ equip + econ + rural + perAA + equip:econ
+ equip:perAA,
                      family = binomial,
                      data = gavote)

summary(model_binomial)
```

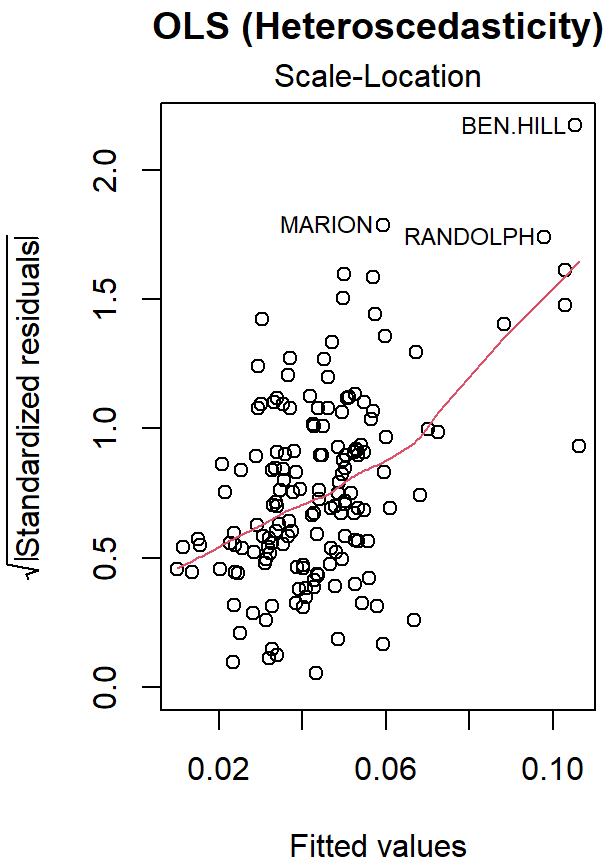
```
##
## Call:
## glm(formula = cbind(ballots - votes, votes) ~ equip + econ +
##     rural + perAA + equip:econ + equip:perAA, family = binomial,
##     data = gavote)
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.469192   0.047910 -72.410  < 2e-16 ***
## equip1        0.231098   0.050194   4.604 4.14e-06 ***
## equip2       -0.084478   0.049232  -1.716  0.08618 .
## equip3        0.364160   0.051146   7.120 1.08e-12 ***
## equip4       -1.427867   0.189568  -7.532 4.99e-14 ***
## econ1         0.157869   0.006965  22.666  < 2e-16 ***
## econ2         0.506878   0.010523  48.170  < 2e-16 ***
## rural1        0.145423   0.005644  25.764  < 2e-16 ***
## perAA         0.006082   0.113390   0.054  0.95722
## equip1:econ1 -0.104953   0.014514  -7.231 4.78e-13 ***
## equip2:econ1 -0.087904   0.009811  -8.960  < 2e-16 ***
## equip3:econ1 -0.030274   0.010211  -2.965  0.00303 **
## equip4:econ1        NA         NA      NA       NA
## equip1:econ2 -0.081072   0.016426  -4.936 7.99e-07 ***
## equip2:econ2 -0.306585   0.013417 -22.851  < 2e-16 ***
## equip3:econ2  0.703329   0.015397  45.681  < 2e-16 ***
## equip4:econ2        NA         NA      NA       NA
## equip1:perAA -1.076911   0.123558  -8.716  < 2e-16 ***
## equip2:perAA  1.416790   0.122538  11.562  < 2e-16 ***
## equip3:perAA -1.256054   0.128296  -9.790  < 2e-16 ***
## equip4:perAA  2.364298   0.440529   5.367 8.01e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36829  on 158  degrees of freedom
## Residual deviance: 11305  on 140  degrees of freedom
## AIC: 12513
##
## Number of Fisher Scoring iterations: 4
```

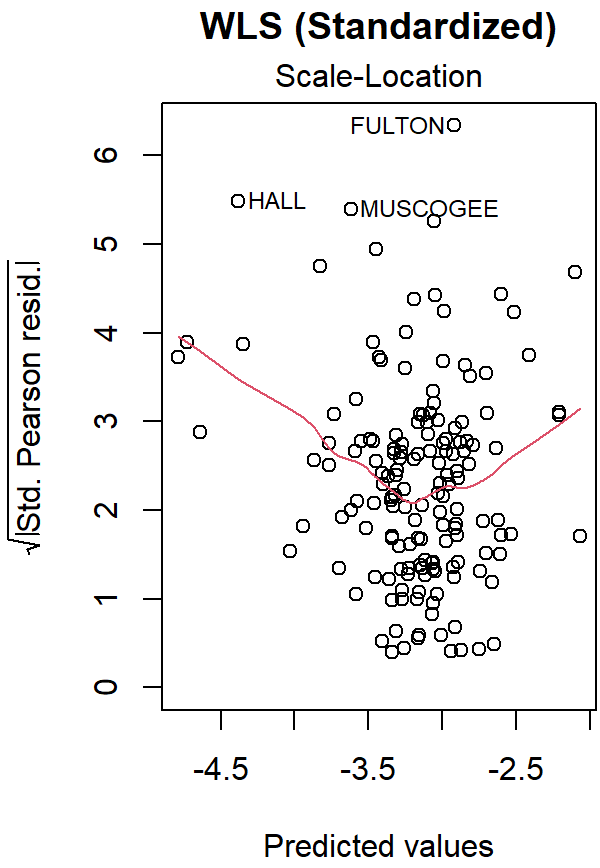
```
par(mfrow=c(1,2))
plot(modelbetter2, which = 3, main = "OLS (Heteroscedasticity)")
```

```
## Warning: not plotting observations with leverage one:
##   103, 131
```

```
plot(model_binomial, which = 3, main = "WLS (Standardized)")
```

```
## Warning: not plotting observations with leverage one:
##   103, 131
```






```
### Getting tables of predictions
pdf <- data.frame(econ=rep(levels(gavote$econ),5),equip=rep(levels(gavote$equip), rep(3,5)), per
AA=0.233, rural="rural")
ppr = predict(model_wls,new=pdf)
```

```
## Warning in predict.lm(model_wls, new = pdf): prediction from rank-deficient
## fit; attr(*, "non-estim") has doubtful cases
```

```
xtabs(round(ppr,3)~econ+equip,pdf)
```

```
##         equip
## econ      LEVER  OS-CC  OS-PC  PAPER  PUNCH
##   middle  0.036  0.046  0.042  0.006  0.074
##   poor    0.051  0.054  0.094  0.024  0.070
##   rich    0.021  0.037  0.012 -0.018  0.031
```

```
pdf <- data.frame(econ=rep(levels(gavote$econ),5),equip=rep(levels(gavote$equip), rep(3,5)), per
AA=0.233, rural="urban")
ppu = predict(model_binomial, newdata = pdf, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
xtabs(round(ppu, 3) ~ econ + equip, pdf)
```

```
##         equip
## econ     LEVER OS-CC OS-PC PAPER PUNCH
##   middle 0.027 0.036 0.032 0.013 0.066
##   poor   0.039 0.040 0.089 0.018 0.055
##   rich   0.016 0.026 0.008 0.006 0.026
```

```
exp(coef(model_binomial))
```

```
##  (Intercept)       equip1       equip2       equip3       equip4        econ1
##   0.03114217   1.25998243   0.91899215   1.43930426   0.23981988   1.17101240
##        econ2       rural1        perAA equip1:econ1 equip2:econ1 equip3:econ1
##   1.66009994   1.15652845   1.00610068   0.90036674   0.91584834   0.97017955
## equip4:econ1 equip1:econ2 equip2:econ2 equip3:econ2 equip4:econ2 equip1:perAA
##           NA   0.92212693   0.73595603   2.02046672           NA   0.34064630
## equip2:perAA equip3:perAA equip4:perAA
##   4.12385974   0.28477548  10.63657324
```

We can look at the e to the coefficients power of our GLM model to try to interpret our odds ratio. We see that economy increases the undercount the most while the equipment type increases and decreases by only a little. There is one equipment and economy combination that seems to be most likely to cause undercounts (punch cards and low economy). This is similar results to what our OLS and WLS models above tell us. It is hard to compare GLM model to OLS model in terms of comparing the coefficient values but the results between the two are similar.

The outcomes of the two models (OLS vs GLM) prediction results actually vary more than OLS and WLS. This is likely because it knows this is a binomial model that can only have probabilities between 0 and 1.

Problem 4

On Slide 24 of Topic 1, you are provided a formula for the β1 coefficient given X2 is already in the model that is a function of the β1 estimate given it is the only predictor in the model. Derive this equation using the fact that β1 for the model Y |X1, X2 can be obtained by regressing the residuals of Y |X2 vs X1|X2 and the relationship between Pearson correlation and the slope in simple linear regression.

We start by standardizing our variables. Let $Z_Y = \frac{Y - \mu_Y}{S_Y}$, $Z_1 = \frac{X_1 - \mu_{X_1}}{S_{X_1}}$, and $Z_2 = \frac{X_2 - \mu_{X_2}}{S_{X_2}}$. Then our residual of Y adjusted for X_2 is $e_{y|2} = Z_Y - r_{y2}Z_2$ and the residual of X_1 adjusted for X_2 is $e_{1|2} = Z_1 - r_{12}Z_2$.

The multiple regression coefficient β_1 is found by regressing $e_{y|2}$ on $e_{1|2}$. Since the means of the residuals are 0, the formula is $\beta_1 = \dfrac{Cov(e_{y|2}, e_{1|2})}{Var(e_{1|2})}$.

We first solve for the denominator. We find

$$\begin{aligned}
\text{Var}(e_{1|2}) &= E[(Z_1 - r_{12}Z_2)^2] \\
&= E[Z_1^2 - 2r_{12}Z_1Z_2 + r_{12}^2Z_2^2] \\
&= E[Z_1^2] - 2r_{12}E[Z_1Z_2] + r_{12}^2E[Z_2^2] \\
&= 1 - 2r_{12}^2 + r_{12}^2 \\
&= 1 - r_{12}^2
\end{aligned}$$

Note that since the variables are standardized, we know that $E[Z_i^2] = 1$ and that $E[Z_iZ_j] = r_{ij}$. Also that $r_{ii} = 1$.

Next we solve for the numerator.

$$\begin{aligned}
\text{Cov}(e_{y|2}, e_{1|2}) &= E[(Z_Y - r_{y2}Z_2)(Z_1 - r_{12}Z_2)] \\
&= E[Z_YZ_1 - r_{12}Z_YZ_2 - r_{y2}Z_2Z_1 + r_{y2}r_{12}Z_2^2] \\
&= E[Z_YZ_1] - r_{12}E[Z_YZ_2] - r_{y2}E[Z_2Z_1] + r_{y2}r_{12}EZ_2^2 \\
&= r_{y1} - r_{12}r_{y2} - r_{y2}r_{12} + r_{y2}r_{12} \\
&= r_{y1} - r_{y2}r_{12}
\end{aligned}$$

Putting it all together and we convert back to unstandarized X_1, X_2 , we find

$$\beta_1 = \frac{r_{y1} - r_{y2}r_{12}}{1 - r_{12}^2}$$

$$\begin{aligned}
\hat{\beta}_1 &= \beta_1 \cdot \frac{S_Y}{S_{X_1}} \\
&= \beta_1 \cdot \sqrt{\frac{s_Y^2}{s_{X_1}^2}} \\
&= \frac{r_{y1} - r_{y2}r_{12}}{1 - r_{12}^2} \cdot \sqrt{\frac{s_Y^2}{s_{X_1}^2}} \\
&= \frac{r_{y1}\dfrac{S_Y}{S_{X_1}} - \sqrt{\dfrac{s_Y^2}{s_{X_1}^2}}r_{y2}r_{12}}{1 - r_{12}^2} \\
&= \frac{\hat{\beta}_1' - \sqrt{\dfrac{s_Y^2}{s_{X_1}^2}}r_{y2}r_{12}}{1 - r_{12}^2}
\end{aligned}$$

where the model coefficient for the slop given Y regressed on only X_1 is $\hat{\beta}_1' = r_{y1}\dfrac{S_Y}{S_{X_1}}$.

Therefore, we have computed the relationship of the parameters between the model based only on X_1 and the ordinary regression model based on both X_1 and X_2.

Problem 5

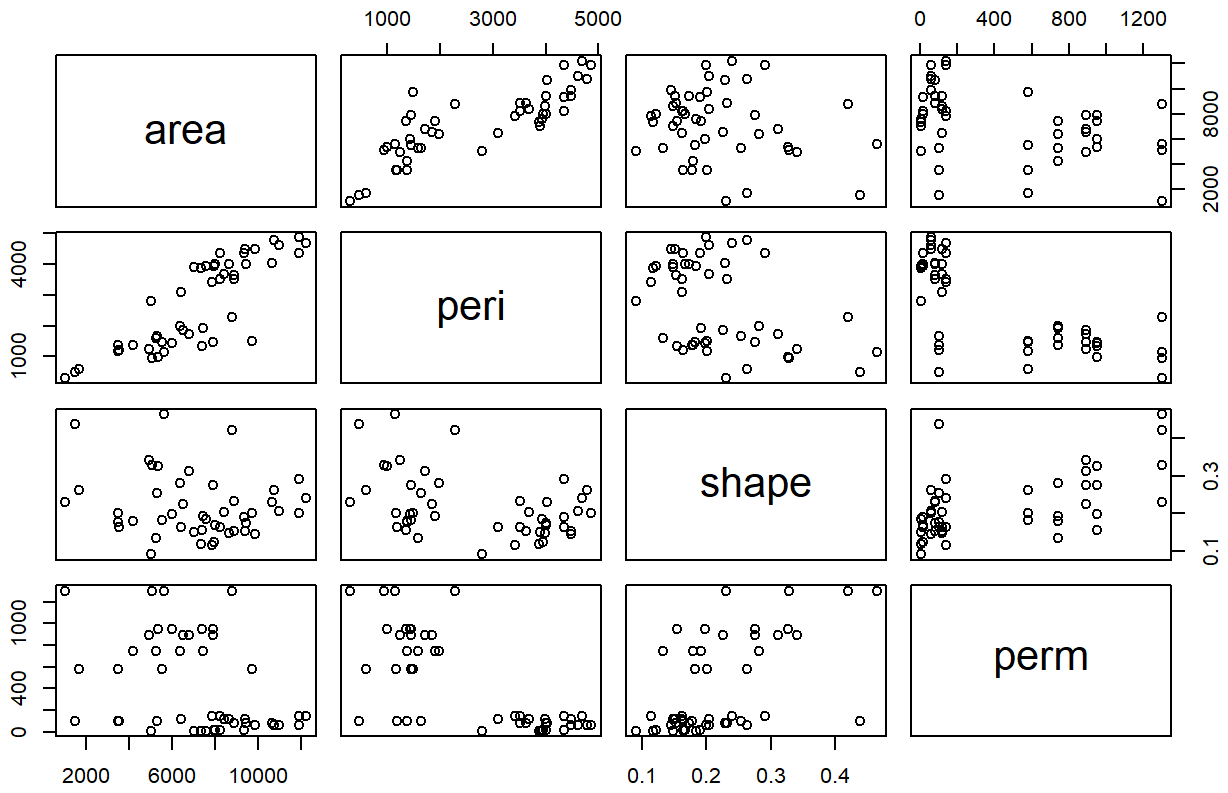
The dataset rock in the faraway library contains 48 rock samples obtained from twelve core samples from petroleum reservoirs sampled a four cross sections. Each rock was measured for permeability. Characteristics of the rock were its total perimeter of pores, total area of pores, and shape. Your goal is to determine the “best” linear model using these characteristics. Faraway (page 24) summarizes various steps to consider in this model selection process. Please describe your steps to derive the best model, what is your best model and any output and figures to support this model.

Begin with data exploration

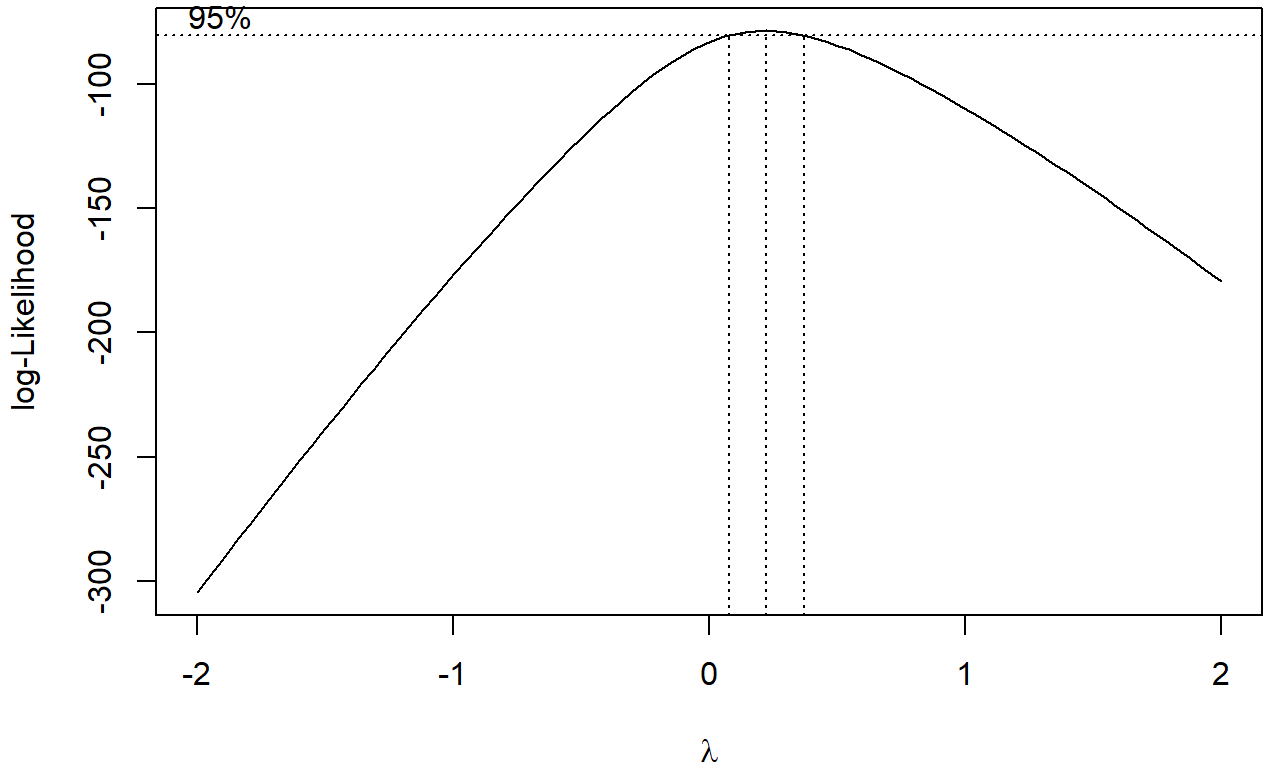
```
#Clear the workspace
rm(list=ls())
library(faraway)
library(MASS)

data(rock)
summary(rock)
```

```
##       area            peri            shape              perm       
##  Min.   : 1016   Min.   : 308.6   Min.   :0.09033   Min.   :   6.30  
##  1st Qu.: 5305   1st Qu.:1414.9   1st Qu.:0.16226   1st Qu.:  76.45  
##  Median : 7487   Median :2536.2   Median :0.19886   Median : 130.50  
##  Mean   : 7188   Mean   :2682.2   Mean   :0.21811   Mean   : 415.45  
##  3rd Qu.: 8870   3rd Qu.:3989.5   3rd Qu.:0.26267   3rd Qu.: 777.50  
##  Max.   :12212   Max.   :4864.2   Max.   :0.46413   Max.   :1300.00
```

```
pairs(rock, main="Scatterplot Matrix of Rock Data")
```




```
# Use Box-Cox to determine best transformation for perm
lmod <- lm(perm ~ area + peri + shape, data = rock)
bc <- boxcox(lmod, plotit = TRUE)
```

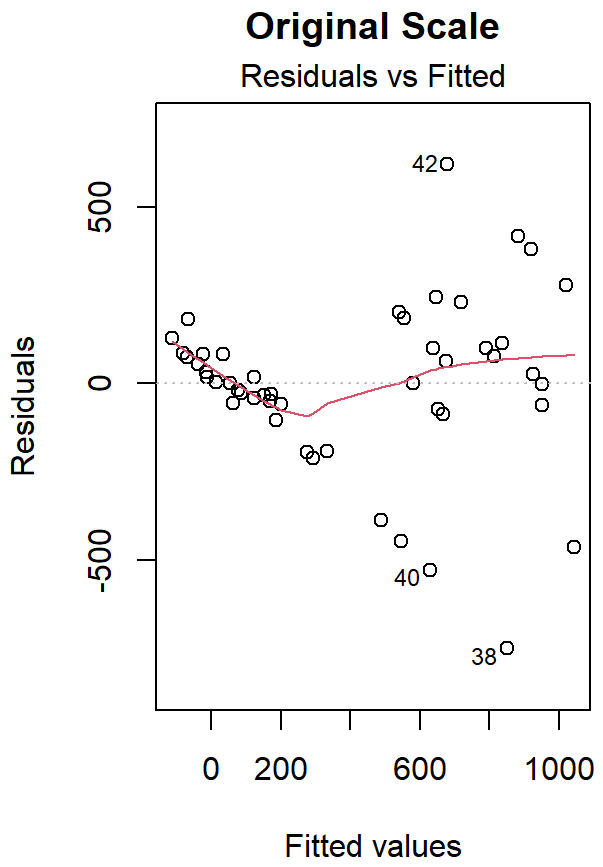


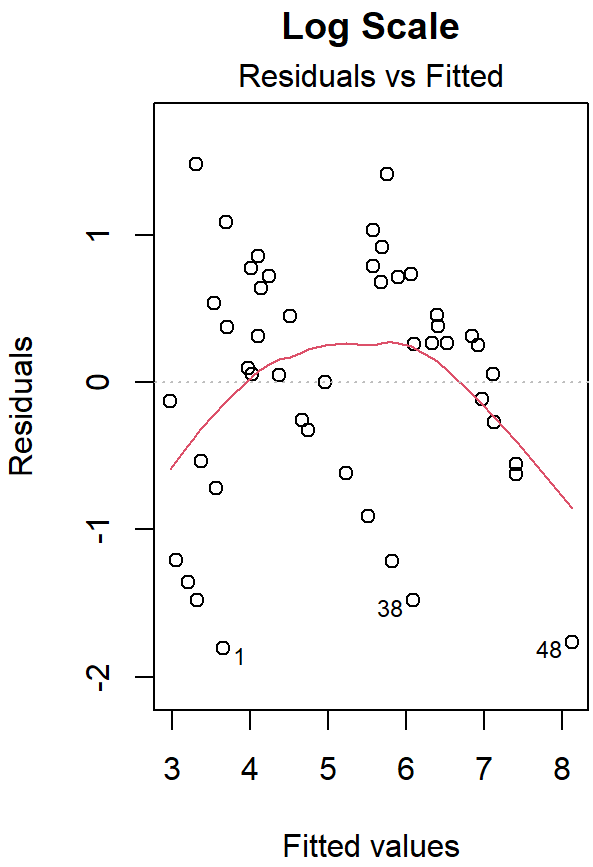

```
# 3. Identify the exact lambda that maximizes the log-likelihood
lambda <- bc$x[which.max(bc$y)]
print(lambda)
```

```
## [1] 0.2222222
```

It seems like we should probably choose lambda in the confidence interval. For interpretability, it is probably best to pick 0, which is close to being in the 95% confidence interval. We can do a quick initial analysis to check this.

```
# Compare Base Model vs. Log-Transformed Model
full_model <- lm(perm ~ area + peri + shape, data = rock)
log_model  <- lm(log(perm) ~ area + peri + shape, data = rock)

par(mfrow=c(1,2))
plot(full_model, which=1, main="Original Scale")
plot(log_model, which=1, main="Log Scale")
```





Now perhaps a step-wise AIC model selection:

```
best_mod <- step(log_model, direction = "both")
```

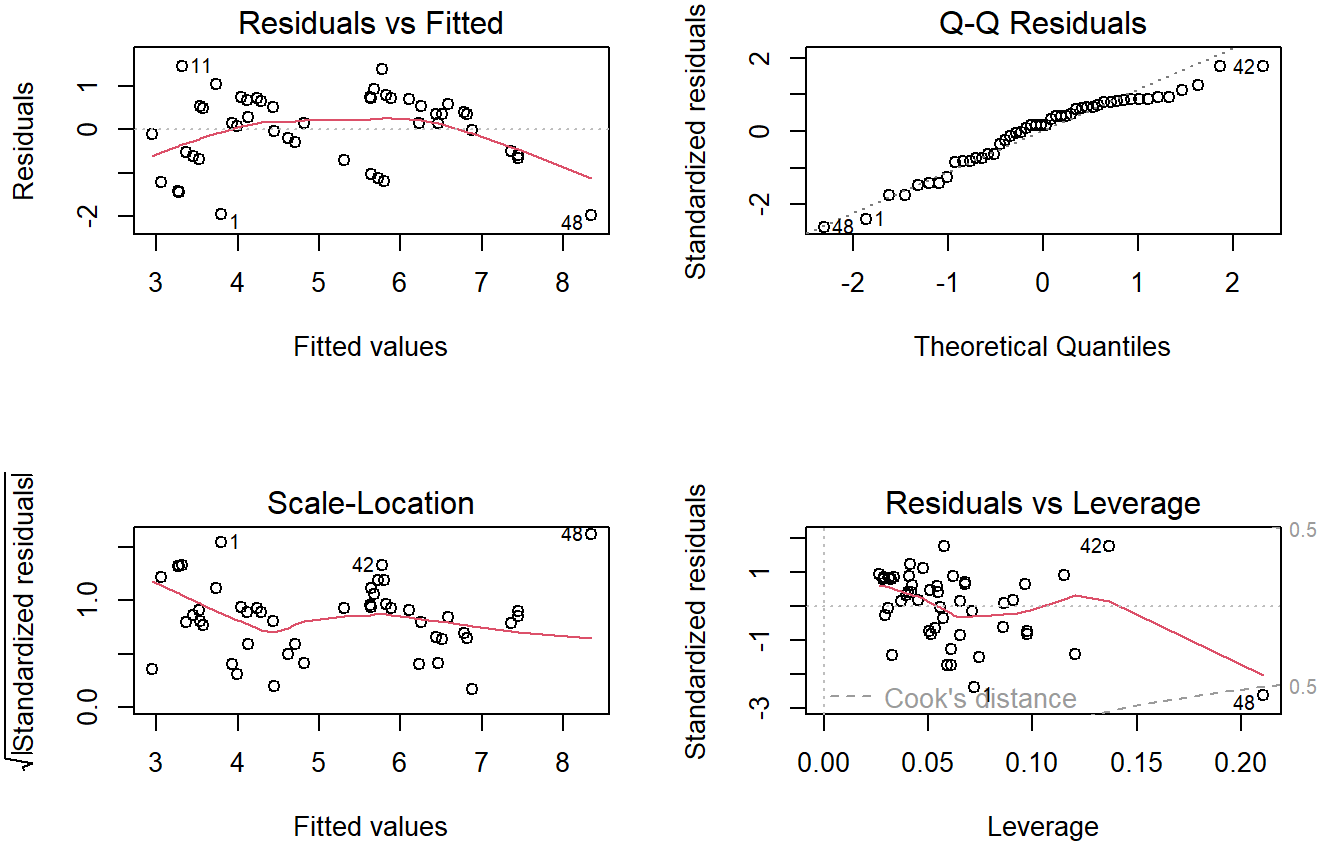
```
## Start:  AIC=-11.54
## log(perm) ~ area + peri + shape
##
##         Df Sum of Sq    RSS     AIC
## - shape  1     0.727 32.675 -12.460
## <none>               31.949 -11.539
## - area   1    22.788 54.736  12.304
## - peri   1    53.988 85.937  33.956
##
## Step:  AIC=-12.46
## log(perm) ~ area + peri
##
##         Df Sum of Sq     RSS     AIC
## <none>                32.675 -12.460
## + shape  1     0.727  31.949 -11.539
## - area   1    28.977  61.652  16.015
## - peri   1    81.412 114.088  45.557
```

```
summary(best_mod)
```

```
##
## Call:
## lm(formula = log(perm) ~ area + peri, data = rock)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9801 -0.5936  0.1406  0.6637  1.4581
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.746e+00  3.621e-01  15.867  < 2e-16 ***
## area         5.144e-04  8.143e-05   6.317 1.05e-07 ***
## peri        -1.616e-03  1.526e-04 -10.589 8.41e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8521 on 45 degrees of freedom
## Multiple R-squared:  0.7426, Adjusted R-squared:  0.7311
## F-statistic:  64.9 on 2 and 45 DF,  p-value: 5.49e-14
```

Our best model keeps area and perimeter. We can now check the diagnostics.

```
par(mfrow=c(2,2))
plot(best_mod)
```



Looking at the plots, we try to ascertain whether our assumptions for the model hold true. In this case we see no distinct pattern in the residuals vs fitted plot and the Normal QQ plot shows a diagonal line, showing normal distribution. Things look good.

We can interpret our final model. Positive coefficient for area means that larger pores generally increase permeability and negative coefficient for perimeter means that for a fixed area, a larger perimeter implies lower permeability.