STAT 425 Assignment 6

Due Monday, April 19, 11:59 pm. Submit through Moodle.

Name: (insert your name here)

Netid: (insert)

Submit your computational work both as an R markdown (*.Rmd) document and as a pdf, along with any files needed to run the code. Embed your answers to each problem in the document below after the question statement. If you have hand-written work, please scan or take pictures of it and include in a pdf file, ideally combined with your pdf output file from R Markdown.

Most relevant class notes: 8.Shrinkage, R_Shrink.Rmd, 9.1.OneWayAnova1, 9.2.OneWayAnova2. We also use some of our methods from earlier in the class.

Problem 1

Consider the `fat` data from the `faraway` library in **R**. The following code is an example of how to select a random test set of 25 observations, and to use the remaining observations as the training set. In the code, we set the random seed to make the result reproducible, but this seed can be changed.

```r
library(faraway)
n=dim(fat)[1]
set.seed(12357)
testid = sample(n, 25, replace=FALSE)
trainid = -testid
test = fat[testid,]
train = fat[trainid,]
```

We will compare several regression methods using train/test evaluation.

a) For the `fat` data, create a randomly selected test set of 25 observations and a training set consisting of all the other observations, removing the variables `brozek` and `density` from the data. Display the first 6 rows of the training and test sets. Also display the dimensions of the training data frame and test data frame.

Answer:

```r
fattrain = train[ ,-c(1, 3)]
fattest = test[ ,-c(1,3)]
```

First six rows of the “train”:

```r
fattrain[1:6, ]
```

```
##   siri age weight height adipos  free neck chest abdom   hip thigh knee ankle
## 1 12.3  23 154.25  67.75   23.7 134.9 36.2  93.1  85.2  94.5  59.0 37.3  21.9
## 2  6.1  22 173.25  72.25   23.4 161.3 38.5  93.6  83.0  98.7  58.7 37.3  23.4
## 3 25.3  22 154.00  66.25   24.7 116.0 34.0  95.8  87.9  99.2  59.6 38.9  24.0
## 4 10.4  26 184.75  72.25   24.9 164.7 37.4 101.8  86.4 101.2  60.1 37.3  22.8
## 5 28.7  24 184.25  71.25   25.6 133.1 34.4  97.3 100.0 101.9  63.2 42.2  24.0
## 6 20.9  24 210.25  74.75   26.5 167.0 39.0 104.5  94.4 107.8  66.0 42.0  25.6
##   biceps forearm wrist
## 1   32.0    27.4  17.1
## 2   30.5    28.9  18.2
## 3   28.8    25.2  16.6
## 4   32.4    29.4  18.2
## 5   32.2    27.7  17.7
## 6   35.7    30.6  18.8
```

Dimensions of “train”:

```r
dim(fattrain)
```

```
## [1] 227  16
```

First six rows of the “test”:

```r
fattest[1:6, ]
```

```
##     siri age weight height adipos  free neck chest abdom   hip thigh knee ankle
## 91  20.5  46 177.00  70.00   25.4 141.3 37.2  99.7  95.6 102.2  58.3 38.2  22.5
## 95   9.0  47 184.25  74.50   23.4 166.6 37.3  99.6  88.8 101.4  57.4 39.6  24.6
## 36  40.1  49 191.75  65.00   32.0 118.4 38.4 118.5 113.1 113.8  61.9 38.3  21.9
## 43  31.6  48 217.00  70.00   31.2 151.1 37.3 113.3 111.2 114.1  67.7 40.9  25.0
## 175 25.3  36 226.75  71.75   31.0 170.9 41.5 115.3 108.8 114.4  69.2 42.4  24.0
## 121 27.9  52 206.50  74.50   26.2 150.7 40.8 104.3  99.2 104.1  58.5 39.3  24.6
##     biceps forearm wrist
## 91    29.1    27.7  17.7
## 95    30.3    27.9  17.8
## 36    32.0    29.8  17.0
## 43    36.7    29.8  18.4
## 175   35.4    21.0  20.1
## 121   33.9    31.2  19.5
```

Dimensions of “test”:

```
dim(fattest)
```

```
## [1] 25 16
```

b) Use the training data to estimate the linear regression of `siri` on all of the other variables except for `brozek` and `density`. Then use the test data to compute the estimated mean square error for prediction.

Answer:

```
mse <- function(y1, y2)
{
  mean((y1 - y2)^2)
}
```

```
lmmodel = lm(siri ~ ., data = fattrain)
mse( predict(lmmodel, newdata = fattest), fattest$siri)
```

```
## [1] 2.090238
```

c) Repeat exercise b) for linear regression with variables selected using the BIC criterion (leaps and bounds or stepwise)

Answer:

```
BIClm = step(lmmodel, direction = c("both"), k = log(n))
```

```
## Start:  AIC=262.13
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - hip      1       0.0  487.8 256.61
## - neck     1       0.3  488.1 256.74
## - wrist    1       0.6  488.4 256.88
## - age      1       1.0  488.9 257.08
## - knee     1       3.1  490.9 258.04
## - height   1       3.1  490.9 258.04
## - biceps   1       6.2  494.0 259.45
## - ankle    1       7.9  495.8 260.26
## <none>                  487.8 262.13
## - chest    1      22.0  509.8 266.60
## - forearm  1      24.6  512.4 267.75
## - thigh    1      26.1  514.0 268.45
## - abdom    1      29.7  517.5 270.00
## - adipos   1      41.8  529.6 275.24
## - weight   1     531.0 1018.8 423.77
## - free     1    3347.2 3835.1 724.67
```

```
##
## Step:  AIC=256.61
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - neck     1       0.3  488.2 251.23
## - wrist    1       0.6  488.4 251.35
## - age      1       1.0  488.9 251.55
## - knee     1       3.1  491.0 252.52
## - height   1       3.2  491.0 252.54
## - biceps   1       6.2  494.0 253.94
## - ankle    1       7.9  495.8 254.74
## <none>                  487.8 256.61
## - chest    1      23.8  511.7 261.91
## + hip      1       0.0  487.8 262.13
## - forearm  1      24.7  512.6 262.30
## - thigh    1      28.9  516.7 264.13
## - abdom    1      31.1  519.0 265.12
## - adipos   1      42.6  530.5 270.09
## - weight   1     668.3 1156.2 446.95
## - free     1    3389.8 3877.7 721.65
##
## Step:  AIC=251.23
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - wrist    1       0.5  488.6 245.91
## - age      1       0.8  489.0 246.09
## - height   1       3.0  491.2 247.10
## - knee     1       3.5  491.7 247.34
## - biceps   1       6.0  494.2 248.48
## - ankle    1       8.2  496.4 249.49
## <none>                  488.2 251.23
## - chest    1      23.9  512.1 256.57
## + neck     1       0.3  487.8 256.61
## + hip      1       0.0  488.1 256.74
## - forearm  1      24.5  512.6 256.81
## - thigh    1      28.8  516.9 258.69
## - abdom    1      30.9  519.1 259.65
## - adipos   1      44.8  532.9 265.62
## - weight   1     670.8 1159.0 441.97
## - free     1    3524.5 4012.6 723.88
##
```

```
## Step:  AIC=245.91
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - age      1       1.8  490.4 241.22
## - height   1       3.3  492.0 241.92
## - knee     1       3.6  492.2 242.03
## - biceps   1       6.0  494.6 243.16
## - ankle    1       9.7  498.3 244.82
## <none>                  488.6 245.91
## - chest    1      23.6  512.2 251.07
## + wrist    1       0.5  488.2 251.23
## + neck     1       0.2  488.4 251.35
## + hip      1       0.0  488.6 251.42
## - thigh    1      28.5  517.1 253.23
## - forearm  1      30.2  518.9 254.01
## - abdom    1      31.3  519.9 254.45
## - adipos   1      44.3  533.0 260.10
## - weight   1     671.9 1160.6 436.75
## - free     1    3794.7 4283.3 733.17
##
## Step:  AIC=241.22
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - height   1       3.1  493.5 237.11
## - knee     1       5.5  495.9 238.20
## - biceps   1       7.0  497.4 238.90
## - ankle    1       9.4  499.8 239.99
## <none>                  490.4 241.22
## + age      1       1.8  488.6 245.91
## + wrist    1       1.4  489.0 246.09
## + hip      1       0.0  490.4 246.74
## + neck     1       0.0  490.4 246.75
## - chest    1      24.9  515.3 246.91
## - thigh    1      27.7  518.1 248.16
## - forearm  1      29.1  519.5 248.75
## - abdom    1      41.3  531.7 254.02
## - adipos   1      44.1  534.5 255.23
## - weight   1     682.4 1172.8 433.61
## - free     1    3794.8 4285.3 727.75
##
## Step:  AIC=237.11
## siri ~ weight + adipos + free + chest + abdom + thigh + knee +
##     ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - knee     1       4.9  498.4 233.81
## - biceps   1       7.3  500.9 234.94
## - ankle    1      10.0  503.5 236.13
## <none>                  493.5 237.11
## + height   1       3.1  490.4 241.22
## + wrist    1       1.8  491.8 241.83
## + age      1       1.6  492.0 241.92
## + hip      1       0.1  493.4 242.60
## + neck     1       0.0  493.5 242.63
## - chest    1      25.0  518.5 242.78
## - thigh    1      25.5  519.0 243.02
## - forearm  1      30.5  524.0 245.18
## - abdom    1      43.1  536.6 250.58
## - adipos   1      77.3  570.8 264.60
## - weight   1     806.1 1299.6 451.38
## - free     1    3810.4 4303.9 723.20
##
## Step:  AIC=233.81
## siri ~ weight + adipos + free + chest + abdom + thigh + ankle +
##     biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - biceps   1       6.9  505.3 231.41
## <none>                  498.4 233.81
## - ankle    1      13.9  512.3 234.53
## + knee     1       4.9  493.5 237.11
## + age      1       3.3  495.1 237.85
## + wrist    1       2.5  495.8 238.18
## + height   1       2.5  495.9 238.20
## - chest    1      24.5  522.9 239.18
## + hip      1       0.0  498.4 239.33
## + neck     1       0.0  498.4 239.34
## - forearm  1      32.6  531.0 242.68
## - thigh    1      33.4  531.8 243.02
## - abdom    1      48.7  547.1 249.45
## - adipos   1      91.6  589.9 266.57
## - weight   1     876.9 1375.3 458.70
## - free     1    3808.9 4307.3 717.85
##
## Step:  AIC=231.41
## siri ~ weight + adipos + free + chest + abdom + thigh + ankle +
##     forearm
##
##            Df Sum of Sq    RSS    AIC
## <none>                   505.3 231.41
## - ankle     1      13.0  518.3 231.64
## + biceps    1       6.9  498.4 233.81
## + knee      1       4.4  500.9 234.94
## + age       1       4.3  501.0 234.98
## + wrist     1       3.0  502.3 235.60
## + height    1       2.8  502.5 235.66
## + hip       1       0.2  505.1 236.86
## + neck      1       0.1  505.2 236.91
## - chest     1      26.7  532.0 237.55
## - thigh     1      39.8  545.1 243.08
## - abdom     1      45.2  550.4 245.31
## - forearm   1      49.1  554.4 246.92
## - adipos    1      86.7  592.0 261.82
## - weight    1     909.3 1414.6 459.57
## - free      1    3805.8 4311.1 712.52
```

```
mse(predict(BIClm, newdata = fattest), fattest$siri)
```

```
## [1] 2.349131
```

d) Repeat exercise b) for scaled principal components regression, where you keep enough components to account for 90% of the variation in predictor variables.

Answer:

```
summary(prcomp(fattrain[,-1], scale = TRUE))
```

```
## Importance of components:
##                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.091 1.2565 1.02148 0.80898 0.76463 0.57765 0.56200
## Proportion of Variance 0.637 0.1052 0.06956 0.04363 0.03898 0.02225 0.02106
## Cumulative Proportion  0.637 0.7423 0.81184 0.85547 0.89445 0.91670 0.93775
##                            PC8    PC9    PC10   PC11    PC12    PC13   PC14
## Standard deviation     0.51538 0.4330 0.42112 0.3507 0.27785 0.21964 0.1937
## Proportion of Variance 0.01771 0.0125 0.01182 0.0082 0.00515 0.00322 0.0025
## Cumulative Proportion  0.95546 0.9680 0.97978 0.9880 0.99313 0.99635 0.9989
##                           PC15
## Standard deviation     0.13157
## Proportion of Variance 0.00115
## Cumulative Proportion  1.00000
```

As we can see, the cumulative proportion will exceed 90% when we use the first 6 principal components. Therefore, in the following prediction, we will use the first 6 components.

```r
library("pls")
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```r
pc = pcr(siri ~ ., data = fattrain)
pred = predict(pc, fattest, ncomp = 6)
mse(pred, fattest$siri)
```

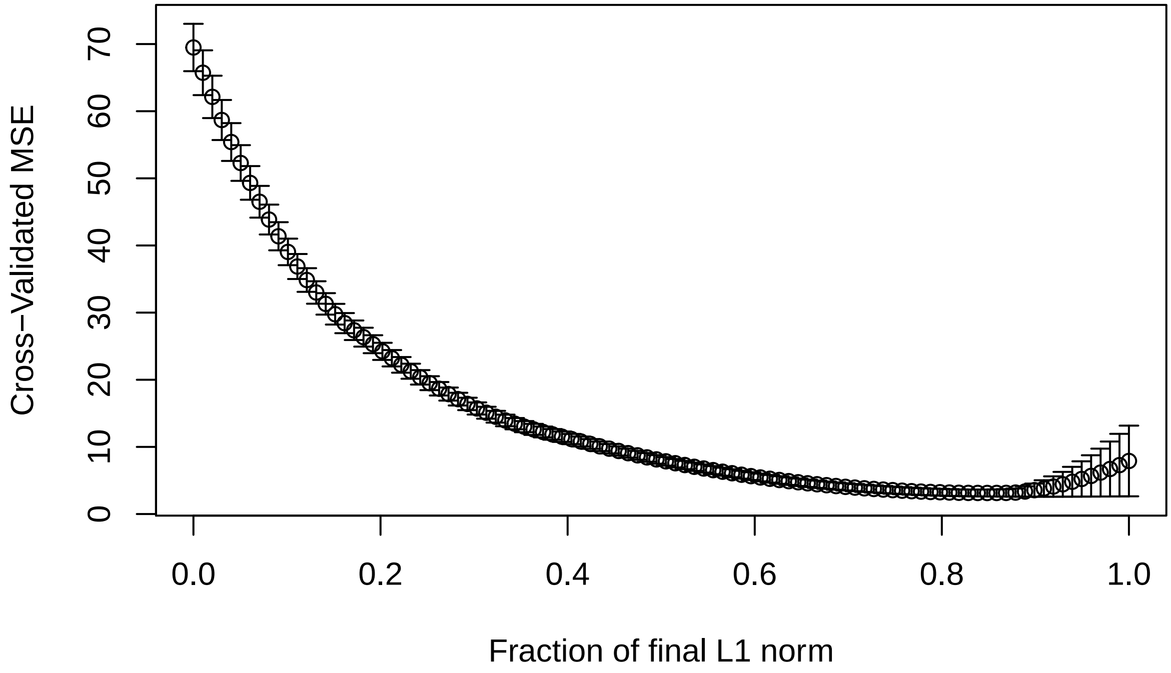
```
## [1] 2.041127
```

e) Repeat exercise b) for Lasso regression, where the amount of shrinkage is selected by 10-fold cross-validation.

```r
library("lars")
```

```
## Loaded lars 1.2
```

```r
ls = lars(as.matrix(fattrain[,-1]) , fattrain$siri, type = "lasso")
cvml = cv.lars(as.matrix(fattrain[,-1]) , fattrain$siri)
```



```r
svm = cvml$index[which.min(cvml$cv)]
predls = predict(ls, fattest[,-1], s = svm, type = "fit", mode = "fraction")$fit
mse(predls, fattest$siri)
```

```
## [1] 2.358614
```

Answer:

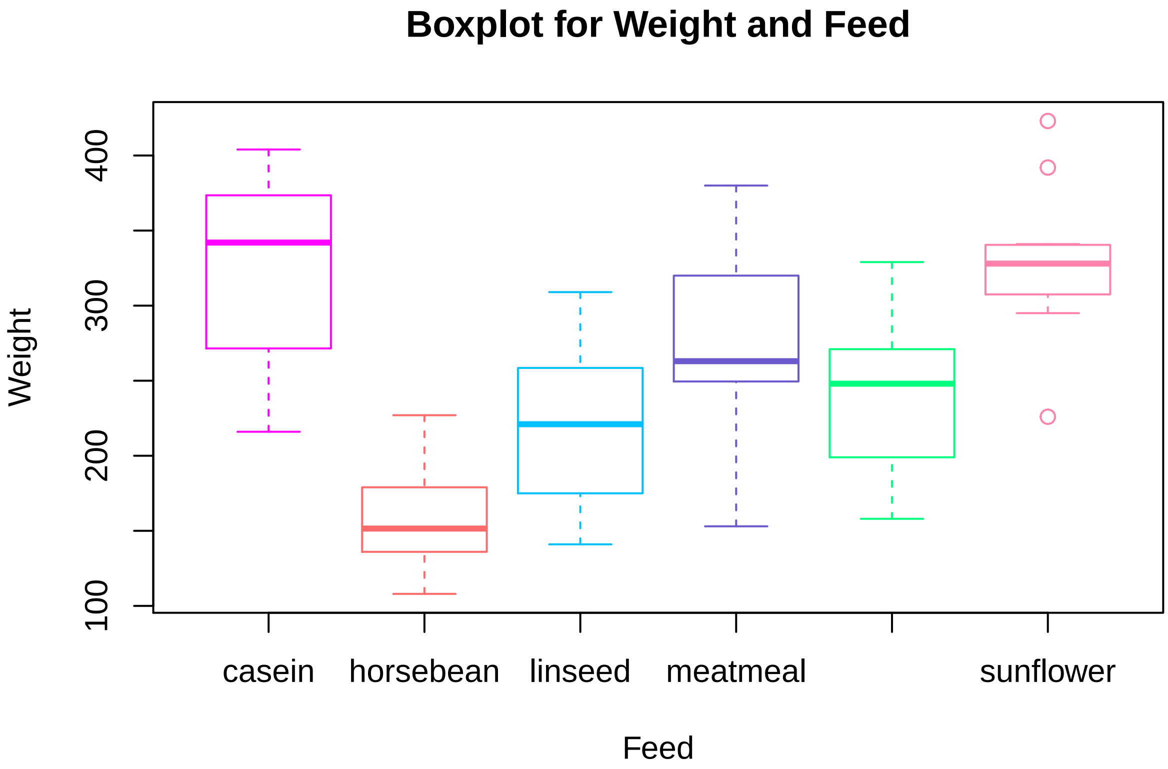
Problem 2

Consider the `chickwts` data in library `datasets`, which compares weights of the chicks randomized into several different groups given different feed supplements.

a) Make boxplots of weight versus feed. Comment on whether the plots show evidence of differences between groups, and whether the data appear consistent with the assumption of normally distributed responses with equal variances.

Answer: We can make a boxplot as below.

```
cols <- c('magenta', 'indianred1', 'deepskyblue',
          'slateblue', 'springgreen', 'palevioletred1')
boxplot(weight ~ feed, data = chickwts,
        col = 'white',
        border = cols,
        xlab = 'Feed', ylab = 'Weight',
        main = 'Boxplot for Weight and Feed')
```



Note that indeed there are differences between groups since many of the boxes do not overlap and the centers appear to vary a lot. For example, for the feed `casein` and `horsebean`, nearly 90% of their data do not match. The normality with equal variances assumption do not hold since many of the boxes do not have equal inter-quartile ranges; see `sunflower` and `casein` for examples.

b) Perform an F test for equality of treatment means. State the null and alternative hypotheses, and indicate whether there is a significant feed effect at level $\alpha = 0.05$.

Answer: Let the means for equal group be μ_i for $i = 1, \ldots, 6$. The hypotheses are as below

$$H_0\colon\ \mu_1 = \cdots = \mu_6 \quad \text{versus} \quad H_1\colon\ \text{Otherwise}$$

We can perform an F-test as below.

```r
mod <- aov(weight ~ feed, data = chickwts)
summary(mod)
```

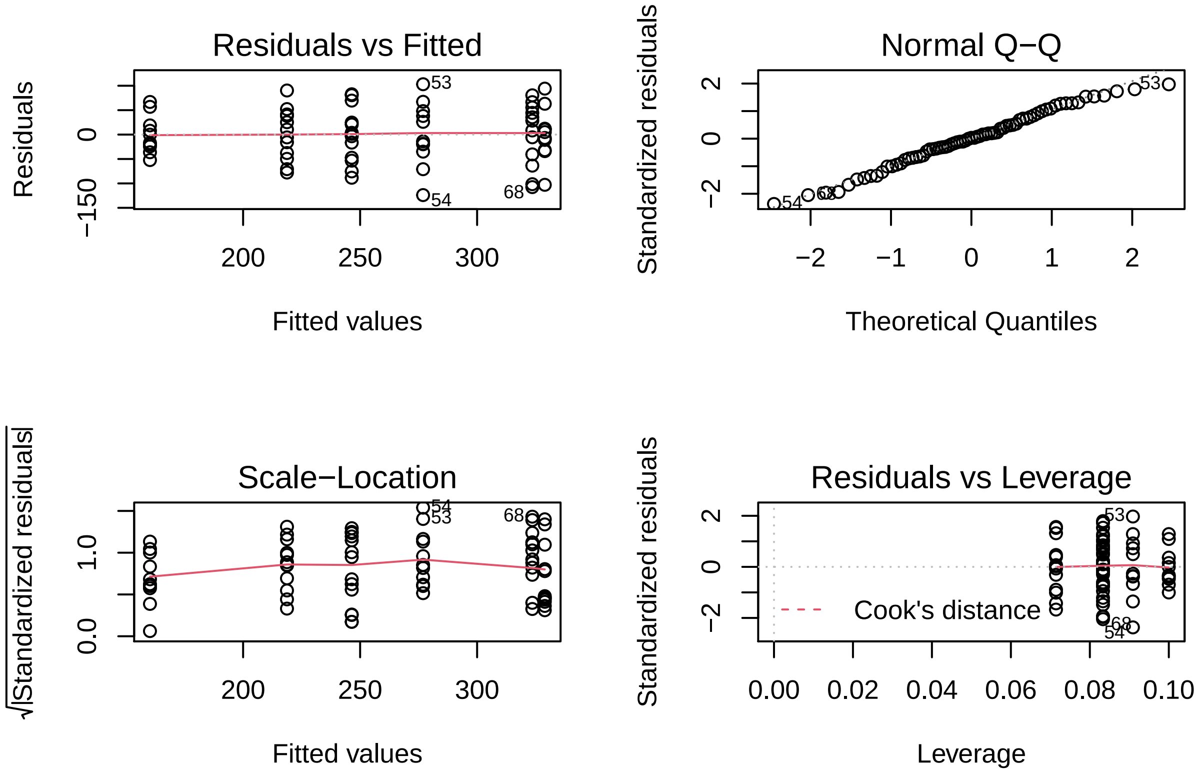
```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## feed         5 231129   46226   15.37 5.94e-10 ***
## Residuals   65 195556    3009                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Indeed, with a p-value $< .001$, we reject H_0 and conclude that there is a significant feed effects.

c) Check the model assumptions using plots of residuals versus fitted values, QQ plot of standardized residuals versus noraml quantiles, and plot of absolute residuals versus fitted values. Comment on what the plots say about the appropriateness of the assumptions of the F test.

Answer: We can make a diagnostics plot as below.

```r
par(mfrow = c(2, 2))
plot(mod)
```



As we can see from the upperleft plot, the linearity assumption is fine with an almost horizontal line. From the QQ plot, we can see that the normality assumption is fine with points closely scattered around the reference line. For the scale-location plot, we can see some trends with a smoothing line that is horizontal. It implies that the constant variance assumptions do not hold as analyzed in (a).

d) Use the Bonferroni method to test all the pairwise differences between treatment means, controlling the family-wise type I error rate at level $\alpha = 0.05$.

Answer: We can use Bonferroni method to do the test as below.

```
with(chickwts, pairwise.t.test(x = weight, g = feed,
p.adjust.method = 'bonferroni'))
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  weight and feed
##
##           casein  horsebean linseed meatmeal soybean
## horsebean 3.1e-08 -         -       -        -
## linseed   0.00022 0.22833   -       -        -
## meatmeal  0.68350 0.00011   0.20218 -        -
## soybean   0.00998 0.00487   1.00000 1.00000  -
## sunflower 1.00000 1.2e-08   9.3e-05 0.39653  0.00447
##
```

```
## P value adjustment method: bonferroni
```

e) Use the Tukey method to obtain all the pairwise confidence intervals for differences between treatment means, with family-wise confidence level of at least 95%.

Answer: We can use Tukey method to obtain the pairwise confidence intervals as below.

```
TukeyHSD(mod)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weight ~ feed, data = chickwts)
##
## $feed
##                            diff         lwr       upr     p adj
## horsebean-casein    -163.383333 -232.346876 -94.41979 0.0000000
## linseed-casein      -104.833333 -170.587491 -39.07918 0.0002100
## meatmeal-casein      -46.674242 -113.906207  20.55772 0.3324584
## soybean-casein       -77.154762 -140.517054 -13.79247 0.0083653
## sunflower-casein       5.333333  -60.420825  71.08749 0.9998902
## linseed-horsebean     58.550000  -10.413543 127.51354 0.1413329
## meatmeal-horsebean   116.709091   46.335105 187.08308 0.0001062
## soybean-horsebean     86.228571   19.541684 152.91546 0.0042167
## sunflower-horsebean  168.716667   99.753124 237.68021 0.0000000
## meatmeal-linseed      58.159091   -9.072873 125.39106 0.1276965
## soybean-linseed       27.678571  -35.683721  91.04086 0.7932853
## sunflower-linseed    110.166667   44.412509 175.92082 0.0000884
## soybean-meatmeal     -30.480519  -95.375109  34.41407 0.7391356
## sunflower-meatmeal    52.007576  -15.224388 119.23954 0.2206962
## sunflower-soybean     82.488095   19.125803 145.85039 0.0038845
```

Problem 3:

Consider the `infmort` data in library `faraway`. The data include per capita income, infant mortality per 1000 births, and oil exporter status for 5 regions of the world.

a) Perform a one-way ANOVA with `mortality` as the response and `region` as the predictor. Is the test significant at level 0.05?

Answer:

```
library(faraway)
data(infmort)

fit = lm(mortality ~ region, data = infmort)
anova(fit)
```

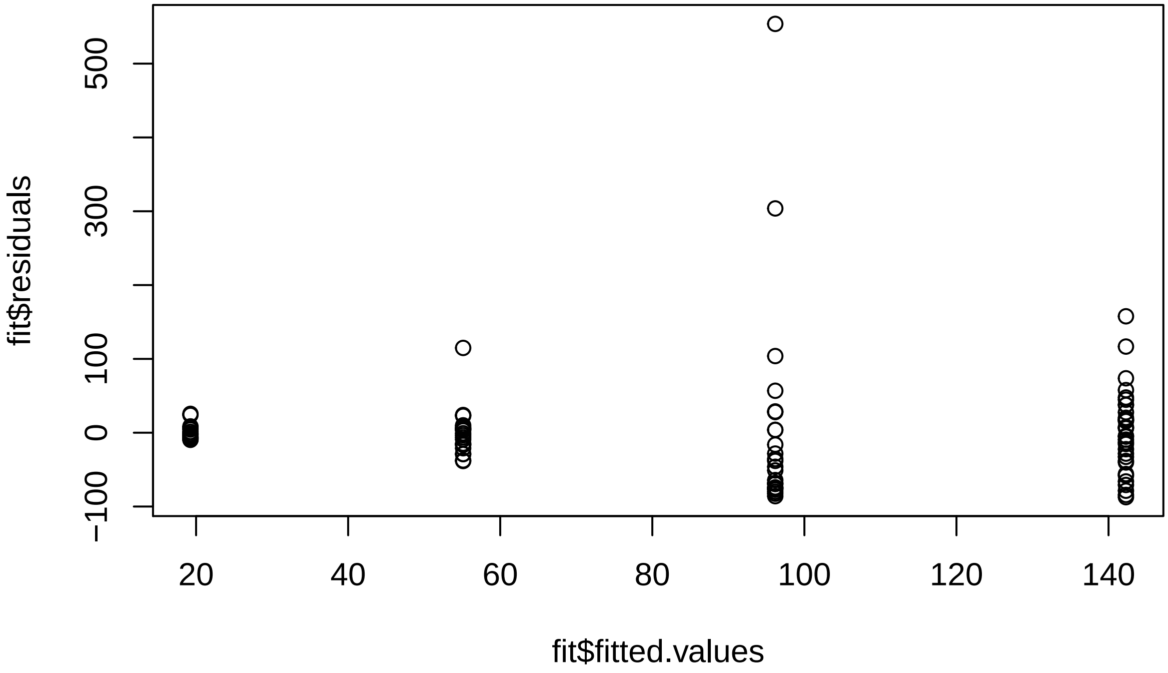
```
## Analysis of Variance Table
##
## Response: mortality
##           Df Sum Sq Mean Sq F value    Pr(>F)
## region     3 210752   70251  11.103 2.494e-06 ***
## Residuals 97 613743    6327
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a p-value of 2.494e-06, we have enough evidence to reject the null. We can conclude the mortality rates are not the same across regions.

b) Check the residuals of the model to see if you detect any problems with the model assumptions such as Normal errors with constant variance.

Answer:

```
plot(fit$fitted.values, fit$residuals)
```

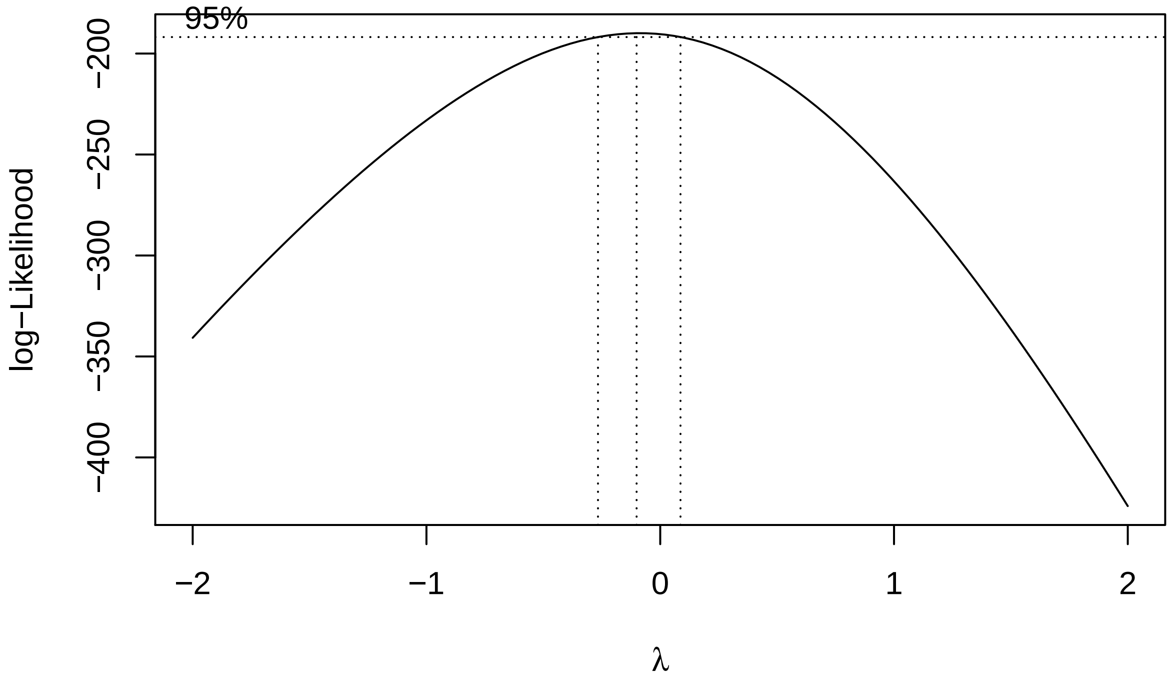


The variance in residuals seems to increase as fitted values increase, so there seems to be evidence of heteroscedasticity. This violates our model assumption.

c) Use the boxcox method to select a transformation of the response. Is the log transformation ($\lambda = 0$) included in the 95% confidence interval for the transformation parameter λ?

Answer:

```
library(MASS)
trans = boxcox(fit)
```



```
lambda = trans$x[trans$y == max(trans$y)]
lambda
```

```
## [1] -0.1010101
```

The log transformation of lambda = 0 is included. The optimal response is lambda = -0.1010101 as shown above.

d) Redo parts a) and b) using log mortality as the response.

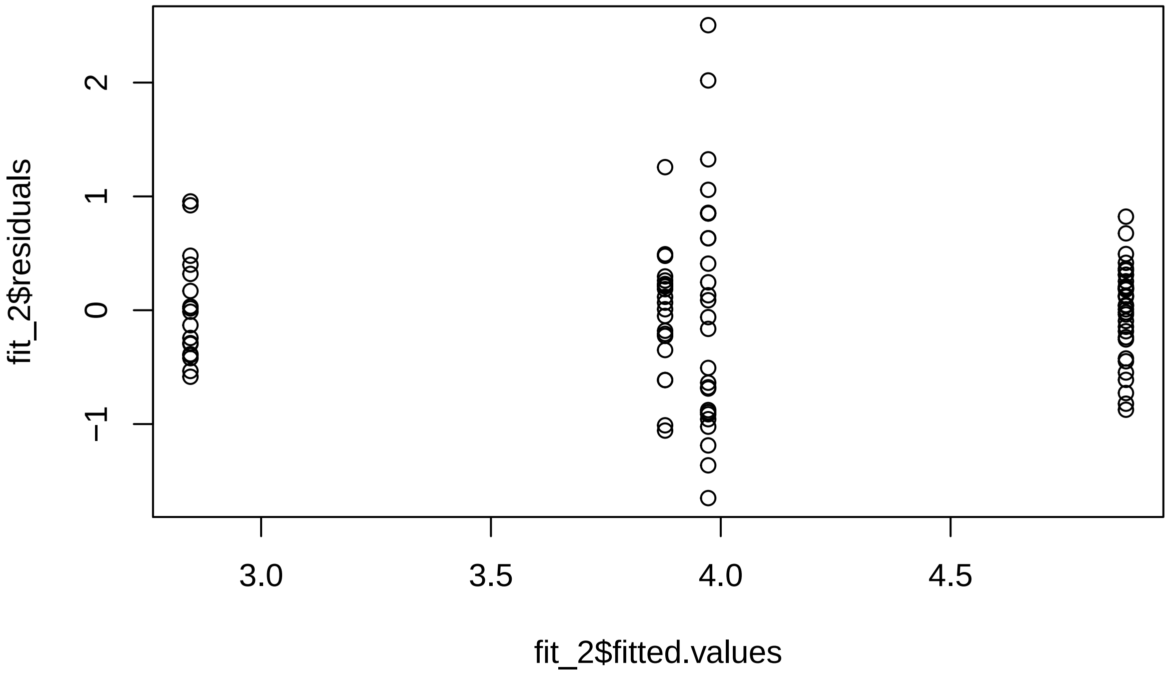
Answer:

```
fit_2 = lm(I(log(mortality)) ~ region, data = infmort)
anova(fit_2)
```

```
## Analysis of Variance Table
##
## Response: I(log(mortality))
##           Df Sum Sq Mean Sq F value    Pr(>F)
## region     3 50.395 16.7985  37.568 3.373e-16 ***
## Residuals 97 43.373  0.4471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a p-value of 0, we have enough evidence to reject the null. We can conclude the log mortality rates are not the same across regions.

```
plot(fit_2$fitted.values, fit_2$residuals)
```



Although the variance in residuals is bigger in the middle of the fitted values, the spread as a whole is less cone-shaped and is an improvement from the residual plot of the previous model.

e) With log mortality as the response, which pairs of regions are significantly different, controlling the family-wise type I error rate at 0.05?

Answer:

```r
pairwise.t.test(I(log(infmort$mortality)), infmort$region, p.adjust.method = "bonferroni
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  I(log(infmort$mortality)) and infmort$region
##
##          Africa  Europe  Asia
## Europe   < 2e-16 -       -
## Asia     4.9e-06 1.6e-06 -
## Americas 2.0e-06 2.7e-05 1
##
## P value adjustment method: bonferroni
```

We can conclude that all pairs of regions are statistically different except Americas vs Asia.