Midterm2-yslin3-STAT425

Rosa Lin

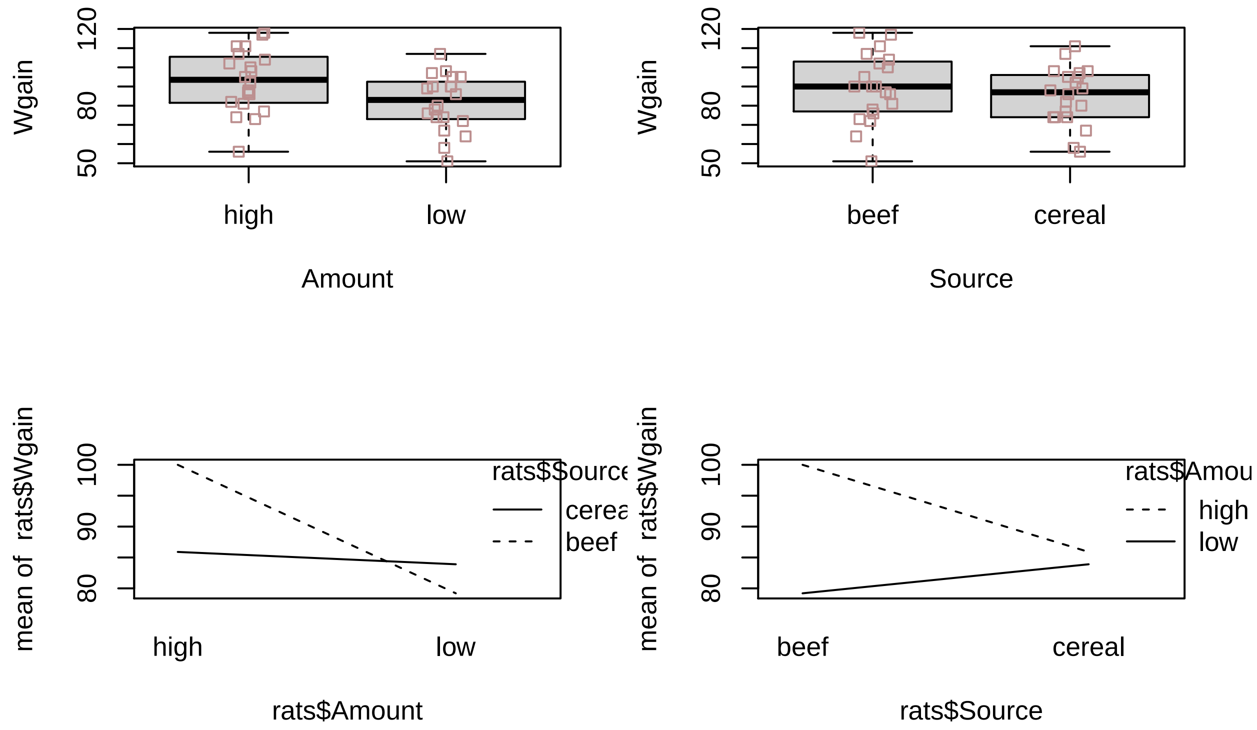
11/14/2020

Question 1

In an experiment to study the gain in weight of rats fed on four different diets, distinguished by the amount of protein (low and high) and by source of protein (beef and cereal), 10 rats were randomized to each of the 4 treatments and the weight gain in grams recorded. The data is in the file rats.txt. The variables are: ID = Observation Number; Source: Source of Protein (beef or cereal); Amount: Amount of protein (low, high); Wgain = Gain in weight in grams

(a) Make appropriate plots of the data.

```r
par(mfrow = c(2,2))
boxplot(Wgain ~ Amount , data = rats, outline = FALSE)
stripchart(Wgain ~ Amount, data = rats, method = "jitter",
           col = "rosybrown", vertical = TRUE, add = TRUE)
boxplot(Wgain ~ Source, data = rats, outline = FALSE)
stripchart(Wgain ~ Source, data = rats, method = "jitter",
           col = "rosybrown", vertical = TRUE, add = TRUE)
interaction.plot(rats$Amount, rats$Source, rats$Wgain)
interaction.plot(rats$Source, rats$Amount, rats$Wgain)
```



(b) Determine whether there is an interaction between amount of protein and source of protein.

```
int_mod = lm(Wgain ~ Amount * Source, data = rats)
anova(int_mod)
```

```
## Analysis of Variance Table
##
## Response: Wgain
##               Df Sum Sq Mean Sq F value  Pr(>F)
## Amount         1 1299.6 1299.60  5.8123 0.02114 *
## Source         1  220.9  220.90  0.9879 0.32688
## Amount:Source  1  883.6  883.60  3.9518 0.05447 .
## Residuals     36 8049.4  223.59
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is no interaction between amount of protein and source of protein. We need to try an additive model.

```
add_mod = lm(Wgain ~ Amount + Source, data = rats)
anova(add_mod)
```

```
## Analysis of Variance Table
##
## Response: Wgain
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Amount     1 1299.6 1299.60  5.3829 0.02596 *
## Source     1  220.9  220.90  0.9150 0.34501
## Residuals 37 8933.0  241.43
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

(c) Determine whether there are statistically significant differences between amounts and also source of protein.

```r
anova(lm(Wgain ~ Amount, data = rats))
```

```
## Analysis of Variance Table
##
## Response: Wgain
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Amount     1 1299.6 1299.60  5.3949 0.02565 *
## Residuals 38 9153.9  240.89
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

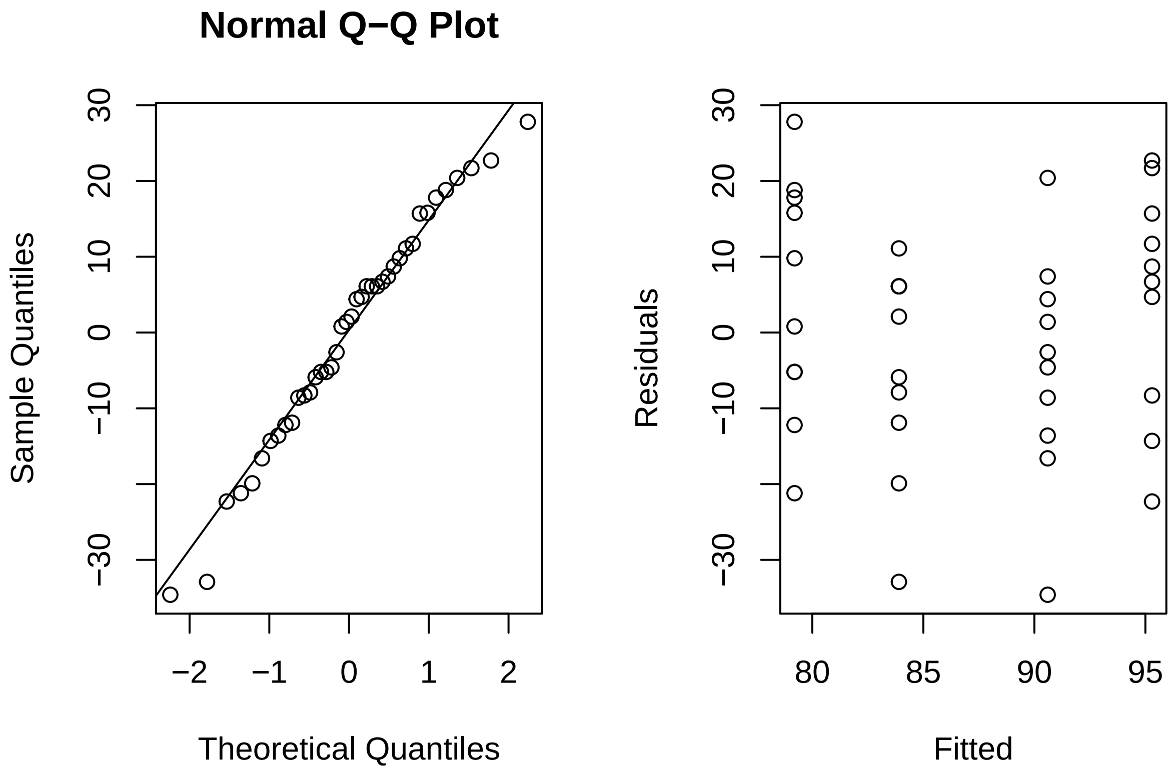
```r
anova(lm(Wgain ~ Source, data = rats))
```

```
## Analysis of Variance Table
##
## Response: Wgain
##           Df  Sum Sq Mean Sq F value Pr(>F)
## Source     1   220.9  220.90  0.8203 0.3708
## Residuals 38 10232.6  269.28
```

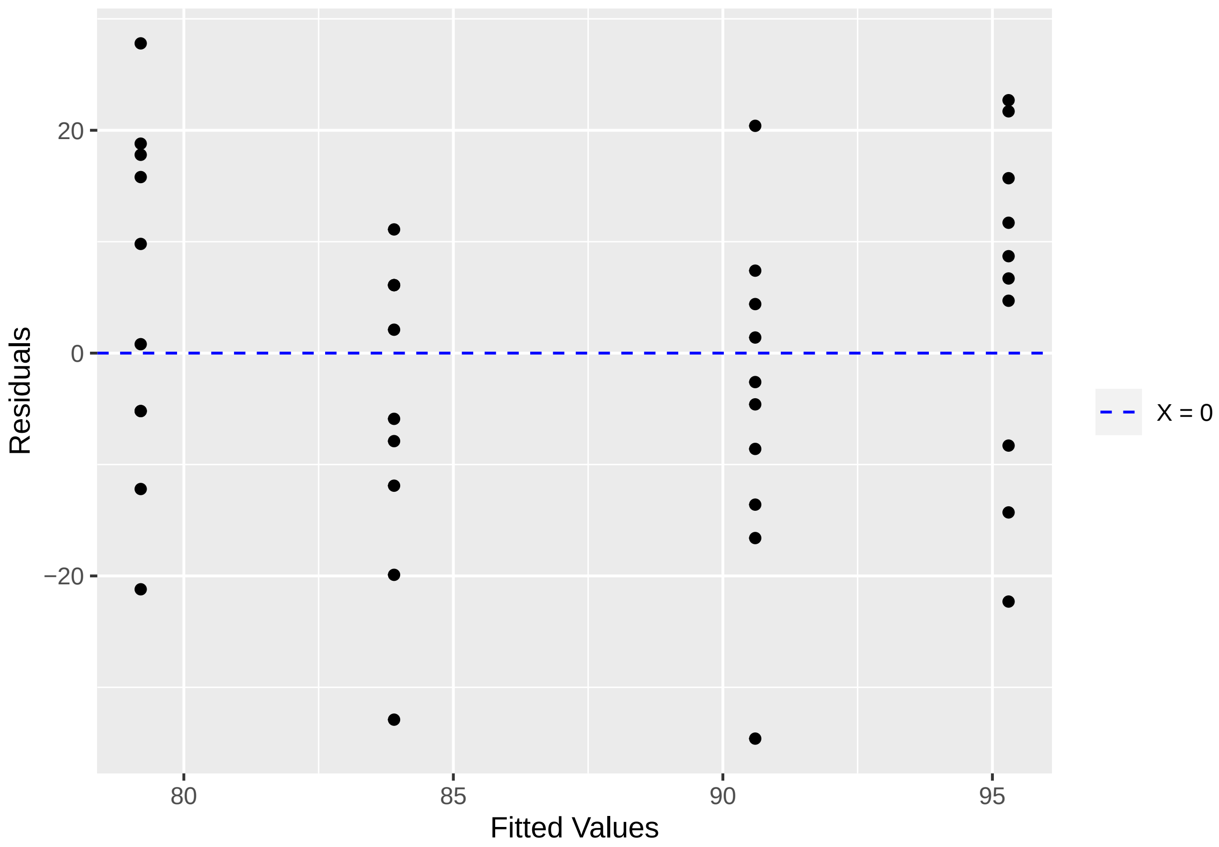
According to the ANOVA table, there is a statistically significant difference between amount of protein , but not between source of protein.

(d) Present regression diagnostics for your chosen model and comment whether the model assumptions have been met.

```r
par(mfrow = c(1,2))
qqnorm(add_mod$residuals)
qqline(add_mod$residuals)
plot(add_mod$fitted.values, add_mod$residuals, xlab = "Fitted", ylab = "Residuals")
```



```r
ggplot() +
  geom_point(aes(x = add_mod$fitted.values,
                 y = add_mod$residuals)) +
  geom_hline(aes(yintercept = 0, linetype = "X = 0"), color = 'blue') +
  labs(linetype = "",
       x = "Fitted Values", y = "Residuals") +
  scale_linetype_manual(values = c(2))
```



```
bptest(add_mod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  add_mod
## BP = 0.072726, df = 2, p-value = 0.9643
```

```
shapiro.test(add_mod$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  add_mod$residuals
## W = 0.97995, p-value = 0.6878
```

```
dwtest(add_mod)
```

```
##
##  Durbin-Watson test
##
## data:  add_mod
## DW = 2.2677, p-value = 0.8393
## alternative hypothesis: true autocorrelation is greater than 0
```

The Breusch-Pagan test tells us that this model is homoskedastic since we fail to reject the null when p-value is 0.9643. Using the Shapiro-Wilk test, the p-value is 0.6878. Thus, we can conclude that the residuals are normally distributed. Furthermore, using the Durbin-Watson test, we can conclude that errors are not correlated at a significant level since the p-value is 0.8393.

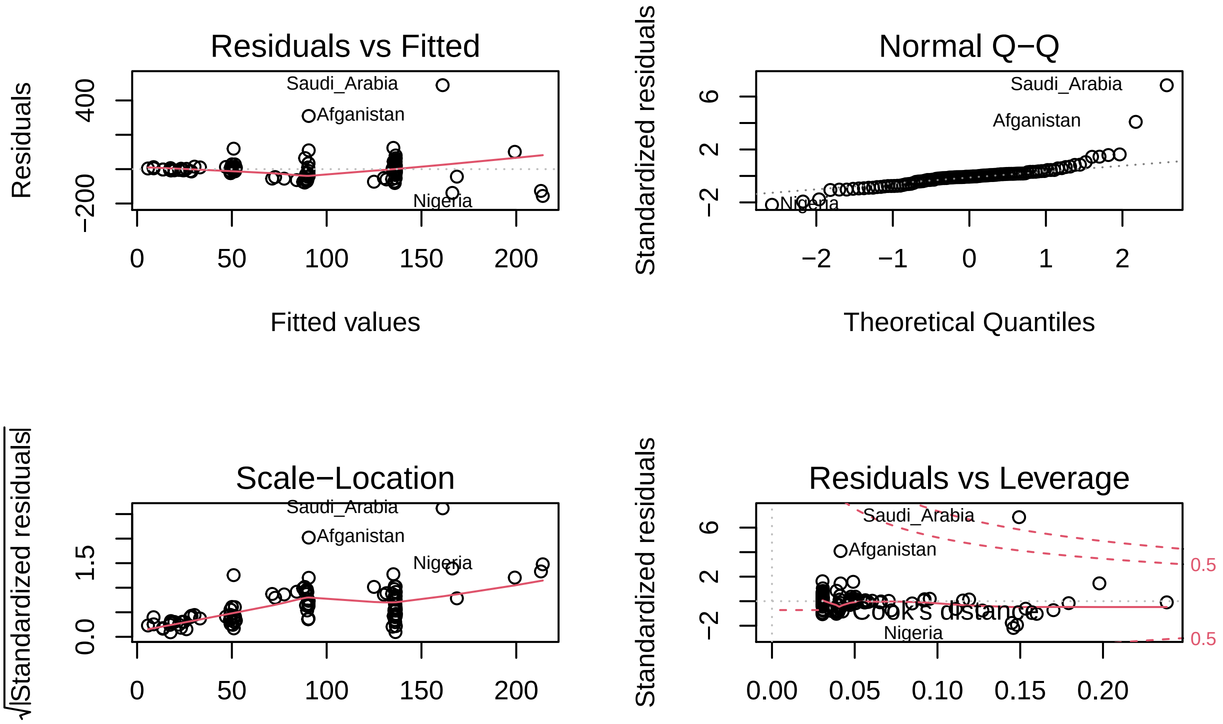
Question 2

Using the infmort data from the faraway library, find a simple model for the infant mortality in terms of the other variables. Be alert for transformations and unusual points. Interpret your model by explaining what the regression parameter estimates mean

```
sim_mod = lm(mortality ~ ., data = infmort)
summary(sim_mod)
```

```
##
## Call:
## lm(formula = mortality ~ ., data = infmort)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -156.00  -32.20   -4.44   13.65  488.82
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.152e+02  2.974e+01   7.234 1.19e-10 ***
## regionEurope       -1.015e+02  3.073e+01  -3.303 0.001351 **
## regionAsia         -4.589e+01  2.014e+01  -2.278 0.024977 *
## regionAmericas     -8.365e+01  2.180e+01  -3.837 0.000224 ***
## income             -5.290e-03  7.404e-03  -0.714 0.476685
## oilno oil exports -7.834e+01  2.891e+01  -2.710 0.007992 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 77.36 on 95 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.3105, Adjusted R-squared:  0.2742
## F-statistic: 8.556 on 5 and 95 DF,  p-value: 1.015e-06
```

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

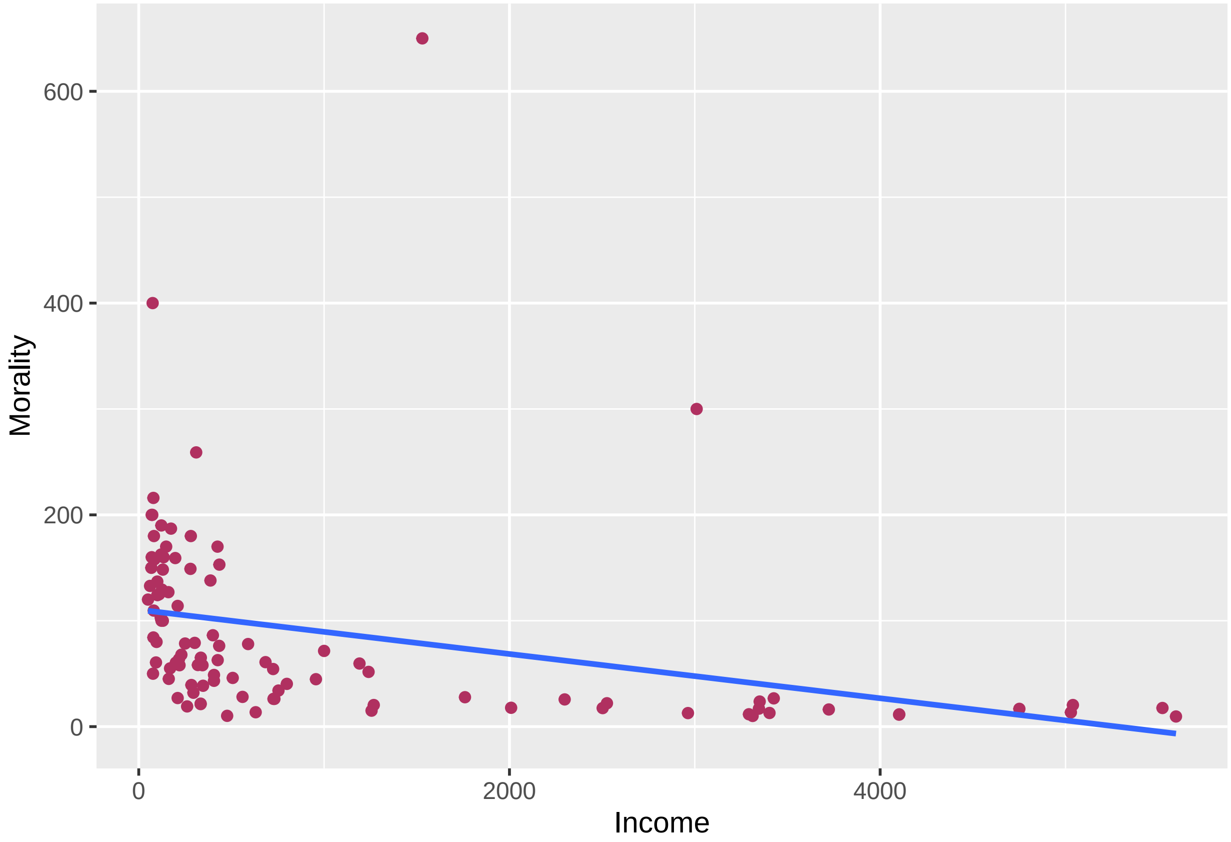


First we fit the model with all variables. We can see from the results above that only income is significant.

```
ggplot(data = infmort, aes(x = income, y = mortality)) +
  geom_point(color = 'maroon') +
  geom_smooth(method = 'lm', formula = y ~ x, se = FALSE) +
  labs(x = 'Income', y = 'Morality')
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

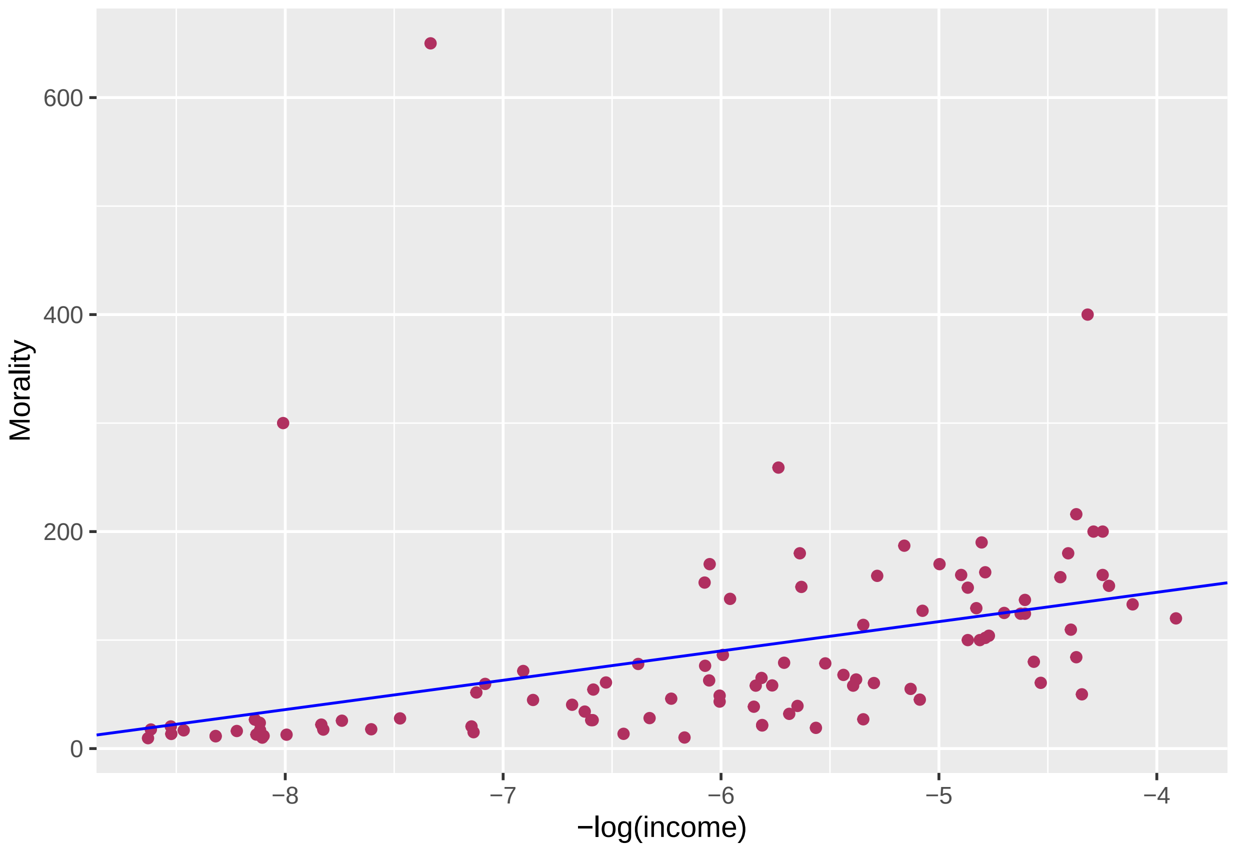
```
## Warning: Removed 4 rows containing missing values (geom_point).
```



Plot does not fit well.

```
log_mod = lm(mortality ~ I(-log(income)), data = infmort)
ggplot(data = infmort, aes(x = -log(income), y = mortality)) +
  geom_point(color = 'maroon') +
  geom_abline(intercept = coef(log_mod)[1], slope = coef(log_mod)[2], color = 'blue') +
  labs(x = '-log(income)', y = 'Morality')
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```



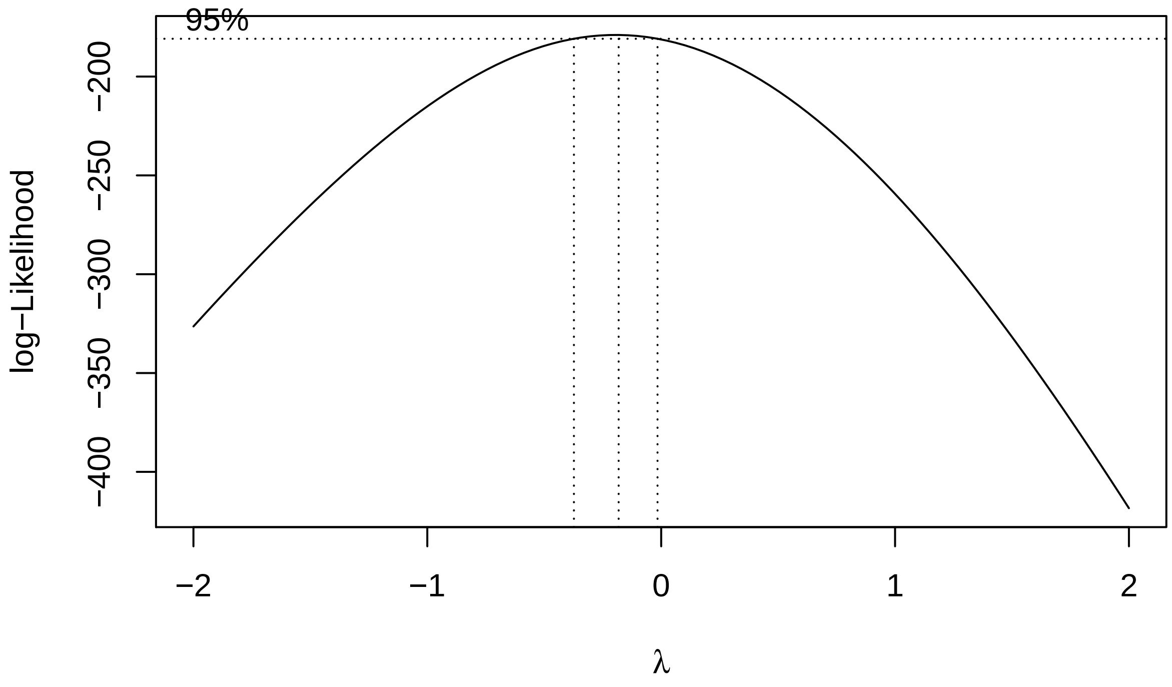
Plot seems to fit better after using -log(income) instead of all variables as predictors.

```
sim_mod2 = lm(mortality ~ region + oil + I(-log(income)), data = infmort)
summary(sim_mod2)
```

```
##
## Call:
## lm(formula = mortality ~ region + oil + I(-log(income)), data = infmort)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -157.89  -34.33   -3.33   14.68  497.00
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          276.479     57.696   4.792 6.09e-06 ***
## regionEurope         -84.334     33.411  -2.524  0.01326 *
## regionAsia           -41.095     20.508  -2.004  0.04793 *
## regionAmericas       -72.369     24.011  -3.014  0.00331 **
## oilno oil exports    -84.558     29.290  -2.887  0.00482 **
## I(-log(income))       11.234      8.666   1.296  0.19799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.89 on 95 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.3188, Adjusted R-squared:  0.283
## F-statistic: 8.894 on 5 and 95 DF,  p-value: 5.906e-07
```

Refit the model, but this time using -log(income). Most numbers are insignificant, but -log(income) is still significant.

```
sim_mod3 = boxcox(sim_mod)
```



Transform Y using log:

```
trans_mod = lm(log(mortality) ~ ., data = infmort)
summary(trans_mod)
```

```
##
## Call:
## lm(formula = log(mortality) ~ ., data = infmort)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63943 -0.32722  0.03984  0.30077  2.28450
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.383e+00  2.373e-01  22.686  < 2e-16 ***
## regionEurope        -1.341e+00  2.451e-01  -5.469 3.65e-07 ***
## regionAsia        -8.294e-01  1.607e-01  -5.161 1.34e-06 ***
## regionAmericas    -8.377e-01  1.739e-01  -4.818 5.49e-06 ***
## income            -2.358e-04  5.906e-05  -3.993 0.000129 ***
## oilno oil exports -4.790e-01  2.306e-01  -2.077 0.040511 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.6171 on 95 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.6142, Adjusted R-squared:  0.5939
## F-statistic: 30.25 on 5 and 95 DF,  p-value: < 2.2e-16
```

All variables are significant. The next step is to combine the Y transformation and -log(income).

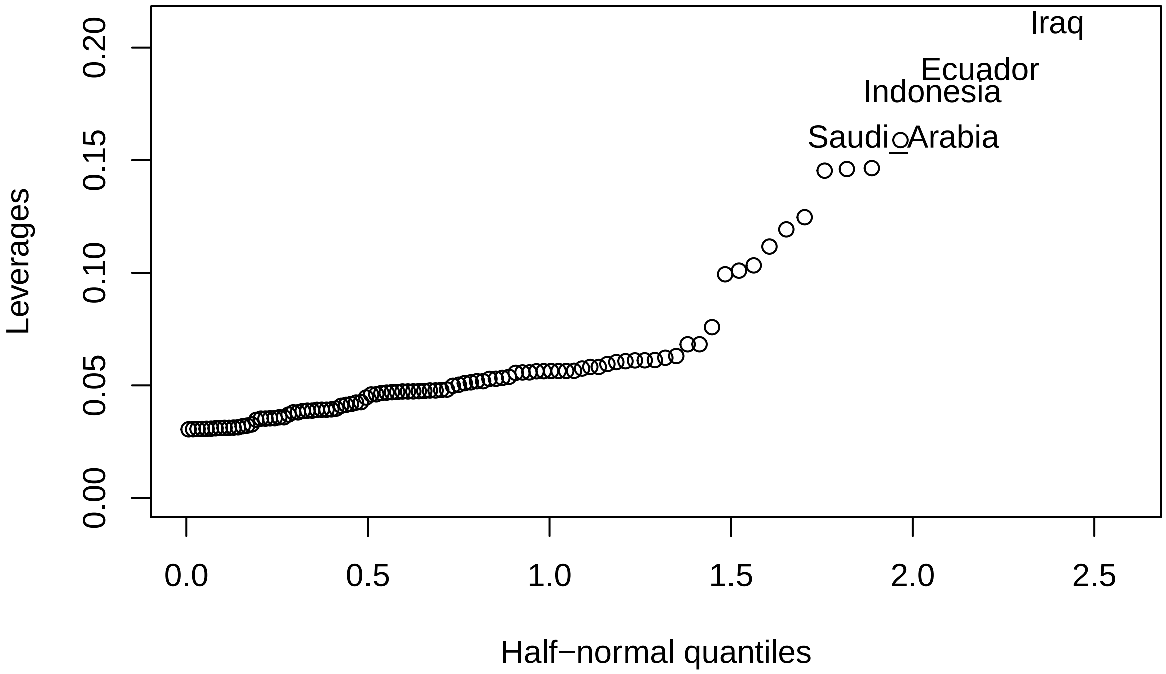
```r
fin_mod = lm(log(mortality) ~ region + oil + I(-log(income)), data = infmort)
```

High Leverages Observations:

```r
n = nrow(model.matrix(fin_mod))
p = ncol(model.matrix(fin_mod))
lev = influence(fin_mod)$hat
lev[lev>2*p/n]
```

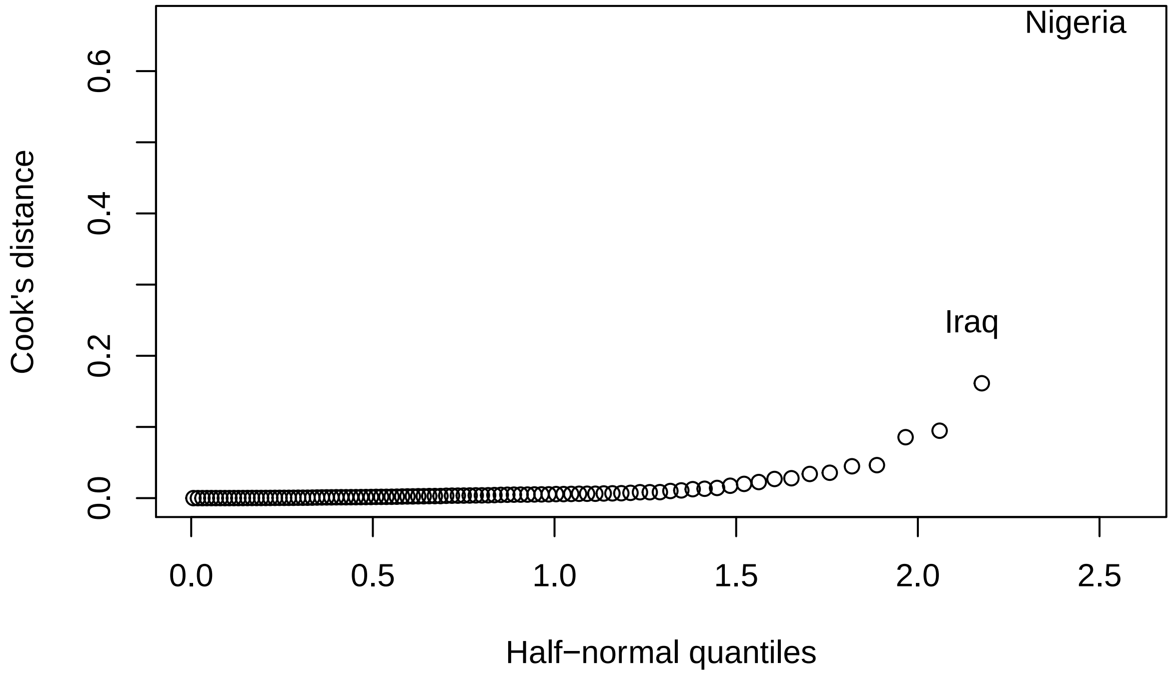
```
## Australia           New_Zealand         Algeria             Ecuador
##           0.1192854           0.1246915           0.1453532           0.1906007
## Indonesia           Iraq                Libya               Nigeria
##           0.1807461           0.1464732           0.2099893           0.1460743
## Saudi_Arabia        Venezuela
##           0.1589187           0.1599469
```

```r
halfnorm(lev, 4, labs = row.names(infmort), ylab = "Leverages")
```



Influential Observations:

```
io = cooks.distance(fin_mod)
halfnorm(io, labs = row.names(infmort), ylab = "Cook's distance")
```



Outliers/Bonferroni Correction:

```
out = rstudent(fin_mod)
qt(0.05 / (2*n), n - p - 1)
```

```
## [1] -3.609027
```

```
sort(abs(out), decreasing = TRUE)[1:5]
```

```
## Saudi_Arabia        Afganistan          Papua_New_Guinea    Nigeria
##           5.183180           2.939646           2.526513           2.440519
## Libya
##           2.407762
```

```
summary(fin_mod)
```

```
##
## Call:
## lm(formula = log(mortality) ~ region + oil + I(-log(income)),
##     data = infmort)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4208 -0.3062 -0.0331  0.3091  2.4897
```

```
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         7.19231    0.44331  16.224  < 2e-16 ***
## regionEurope       -1.03383    0.25672  -4.027 0.000114 ***
## regionAsia         -0.71292    0.15757  -4.524 1.75e-05 ***
## regionAmericas     -0.54984    0.18449  -2.980 0.003657 ** 
## oilno oil exports -0.64021    0.22505  -2.845 0.005444 ** 
## I(-log(income))     0.33985    0.06658   5.104 1.70e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5908 on 95 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.6464, Adjusted R-squared:  0.6278
## F-statistic: 34.73 on 5 and 95 DF,  p-value: < 2.2e-16
```

According to the results above, if the region is in Africa, there will be a 7.19231 increase in mortality. There is a significant positive relationship between income and mortality for African countries. On the other hand,there is a negative relationship between income and mortality for the other regions. If the region is in Europe, there will be a -1.03383 decrease in mortality. If the region is in Asia, there will be a -0.71292 decrease in mortality. If the region is in Americas, there will be a -0.54984 decrease in mortality. If there are no oil exports, there will be a -0.64021 decrease in mortality. This indicates that no oil exporter countries have a positive effect on mortality. In addition, a one percent increase in income is associated with a 0.33985 decrease in mortality.

Question 3

Some near infrared spectra (NIR) on 60 samples of gasoline and corresponding octane numbers can be found by data(gasoline,package=”pls”). The NIR spectra were measured using diffuse reflectance as log(1/R) from 900 nm to 1700 nm in 2 nm intervals, giving 401 wavelengths. Compute the mean value for each wavelength and predict the corresponding response octane number using the following methods:

```r
data(gasoline,package="pls")
str(gasoline)
```

```
## ’data.frame’:    60 obs. of  2 variables:
##  $ octane: num  85.3 85.2 88.5 83.4 87.9 ...
##  $ NIR   : ’AsIs’ num [1:60, 1:401] -0.0502 -0.0442 -0.0469 -0.0467 -0.0509 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:60] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:401] "900 nm" "902 nm" "904 nm" "906 nm" ...
```

```r
newgasoline<-as.matrix(gasoline)
newgasoline <- data.frame(newgasoline)
dim(newgasoline)
```

```
## [1]  60 402
```

```r
mean_val = apply(newgasoline, 2, mean)
mean_data = data.frame(mean_val)
```

(a) Principal Components Regression

```
sam_mod = newgasoline[, 2:402]
dim(sam_mod)
```

```
## [1]  60 401
```

```
pcr_mod = prcomp(sam_mod)
pcr1_mod = prcomp(sam_mod, scale = TRUE)
pcr2_mod = lm(newgasoline$octane ~ pcr1_mod$x[,1:3])
summary(pcr2_mod)
```

```
##
## Call:
## lm(formula = newgasoline$octane ~ pcr1_mod$x[, 1:3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74496 -0.15881 -0.00751  0.14873  0.73011
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         87.177500   0.038500 2264.34   <2e-16 ***
## pcr1_mod$x[, 1:3]PC1  0.026849   0.002289   11.73   <2e-16 ***
## pcr1_mod$x[, 1:3]PC2  0.069240   0.004724   14.66   <2e-16 ***
## pcr1_mod$x[, 1:3]PC3 -0.288511   0.008527  -33.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2982 on 56 degrees of freedom
## Multiple R-squared:  0.9639, Adjusted R-squared:  0.962
## F-statistic:   499 on 3 and 56 DF,  p-value: < 2.2e-16
```

```
fin_mod = predict(pcr2_mod, mean_data)
```

```
## Warning: 'newdata' had 402 rows but variables found have 60 rows
```

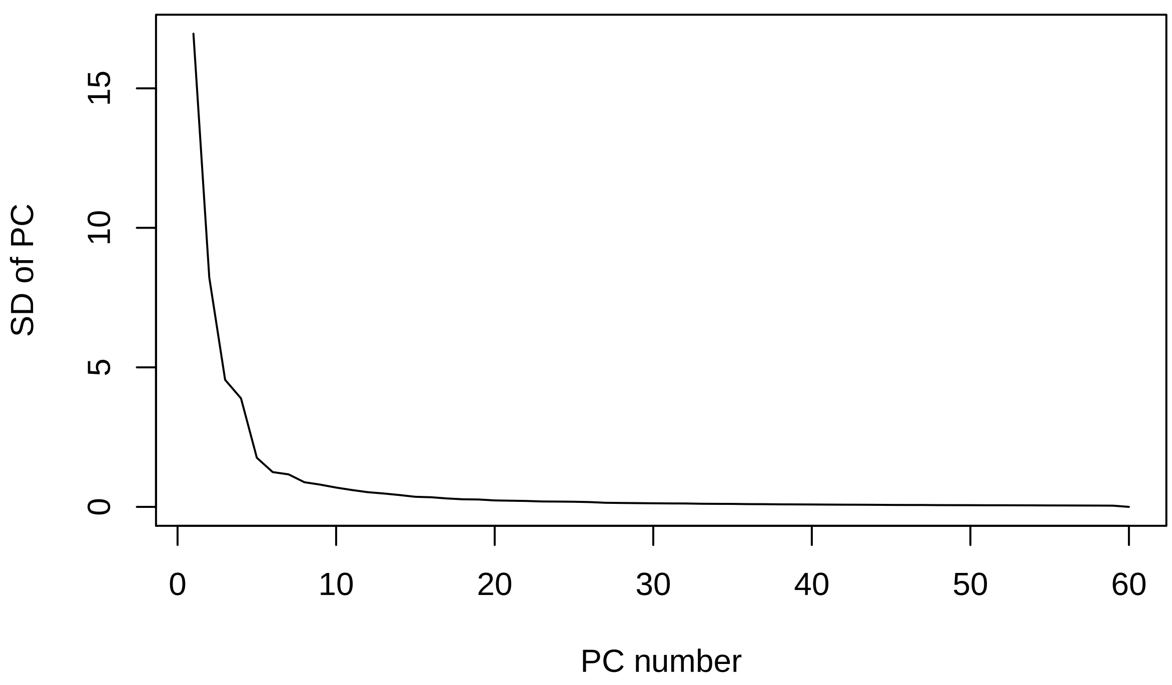
```
tail(fin_mod)
```

```
##       55       56       57       58       59       60
## 85.46975 84.95466 87.75969 87.21015 89.48921 86.98199
```

```
mean2_val = apply(newgasoline[,-1], 2, mean)
fin2_mod = pcr(gasoline$octane ~ gasoline[,-1])
predict(fin2_mod, t(as.matrix(mean2_val)), ncomp = 3)
```

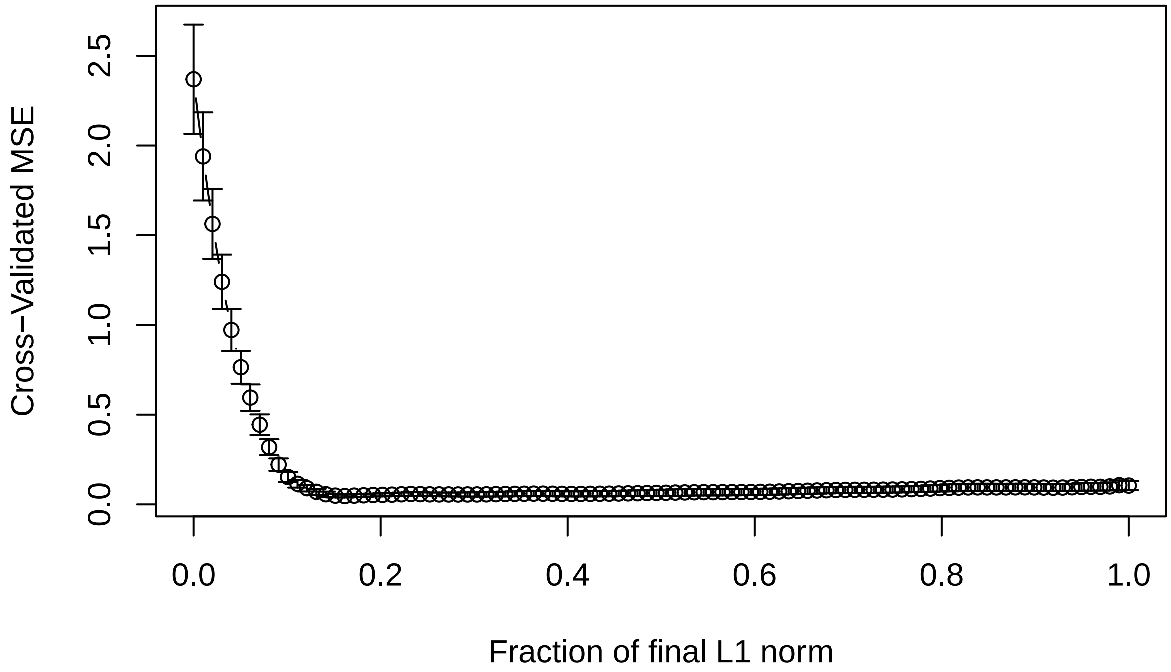
```
## , , 3 comps
##
##      gasoline$octane
## [1,]         87.1775
```

```r
plot(pcr1_mod$sdev, type = "l", ylab = "SD of PC", xlab = "PC number")
```



(b) Lasso Regression

```r
newgasoline <- as.matrix(gasoline)
lasso_mod = lars(newgasoline[, -1], gasoline$octane, type = "lasso")
lasso_cv = cv.lars(newgasoline[, -1], gasoline$octane)
```



```
fit = lasso_cv$index[which.min(lasso_cv$cv)]
predict(lasso_mod, t(as.matrix(mean2_val)), s = fit, mode = "fraction")$fit
```

```
## [1] 87.1775
```

(c) Ridge Regression (Extra Credit)

```
ridge_mod = lm.ridge(gasoline$octane ~ newgasoline[,-1], lambda = seq(0, 50, len = 101))
which.min(ridge_mod$GCV)
```

```
##  9.0
##   19
```

```
cbind(1, as.matrix(t(mean2_val))) %*% coef(ridge_mod)[19, ]
```

```
##          [,1]
## [1,] 87.1775
```