PH245 Homework 2 Solution

Xiangyu Hu, Toki Sherbakov

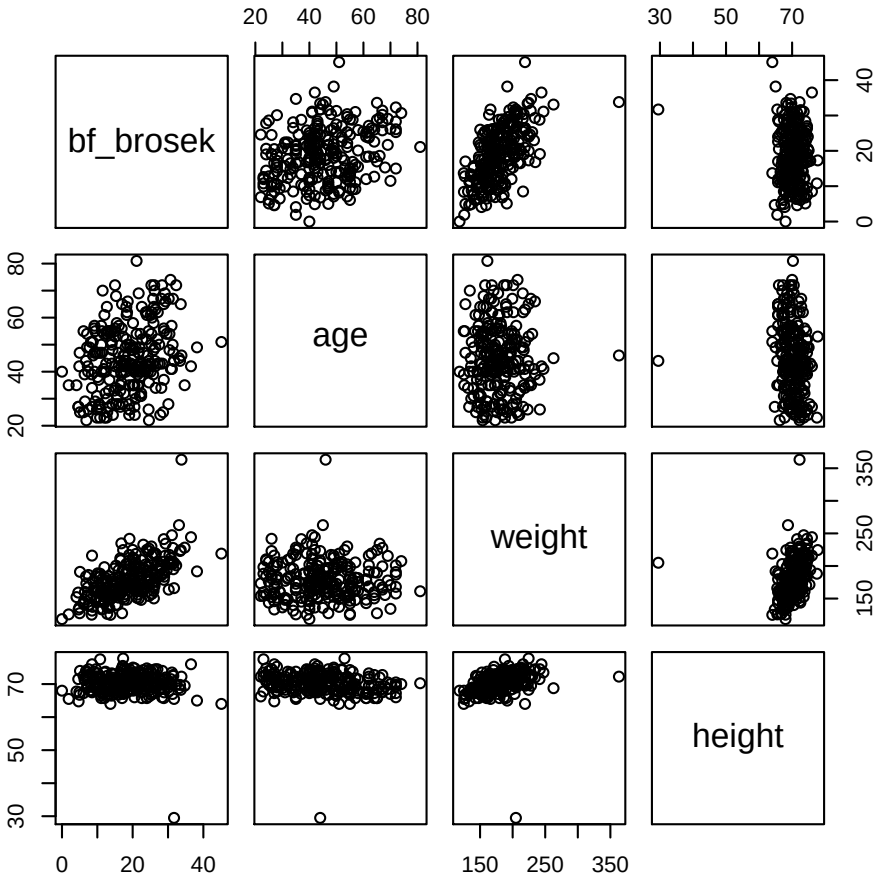
October 27, 2014

Total: 20 pts, Extra: 2 pts

Problem 1.

(a) (*1 pt*)

```
### PART A ###
# set work directory using setwd("The path you stored your data")
setwd("~/Documents/ZhengShi/Berkeley/GSI/PH245/HW/hmwk2/hmwk2_data")
# Read in data
body = read.table(file="Data-HW2-Bodyfat.txt", header = FALSE, quote="")

# Change the column names based on readme file
colnames(body) = c("case", "bf_brosek", "bf_siri", "density", "age", "weight", "height", "adipose",
                   "ff_weight", "neck", "chest", "abdomen", "hip", "thigh", "knee", "ankle",
                   "biceps", "forearm", "wrist")

# Scatterplot matrix of % body fat using brosek's equation, age, height, and weight
pairs(~bf_brosek + age + weight + height, data = body)
```



Based on the scatterplot matrix we see multiple things:
- Age (in years) seems to have a slight positive relationship with body fat %.
- Weight (in lbs) seems to have a strong positive relationship with body fat %
- Height (in inches) does not vary too much (mostly clustered between 65 and 75, with one strange value at around 30). Body fat % varys greatly for heights making it hard to capture this marginal relationship between height and body fat %.

And all the relationships can be described as linear, there are no obvious curvatures on our scatterplots.

(b) (*1 pt*)

```
### PART B ###

# Fit linear regression with % body fat using brosek's equation vs. age, weight, and height.
model1 = lm(bf_brosek ~ age + weight + height, data = body)

# Look at summary of model1
summary(model1)
```

```
##
## Call:
## lm(formula = bf_brosek ~ age + weight + height, data = body)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.002  -4.110  -0.037   3.487  14.458
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

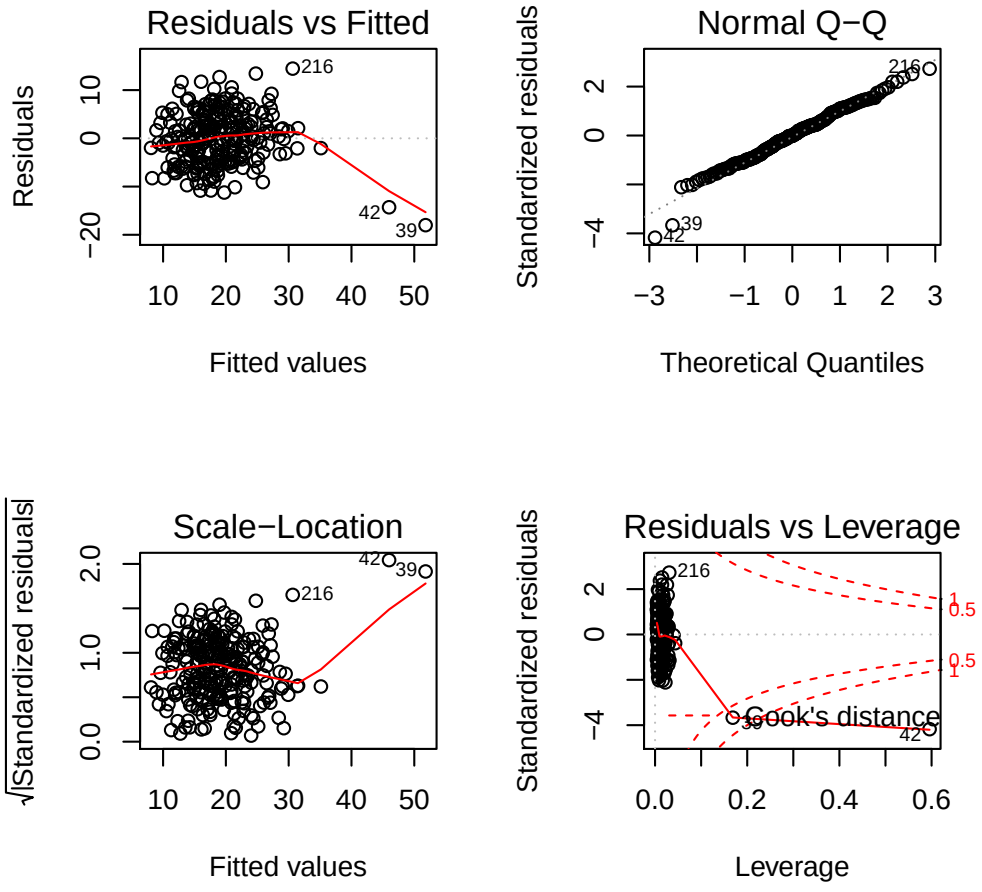
```
## (Intercept)  17.7214     6.9295    2.56    0.011 *
## age           0.1558     0.0274    5.69  3.6e-08 ***
## weight        0.1837     0.0122   15.11  < 2e-16 ***
## height       -0.5510     0.0990   -5.56  6.9e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.38 on 248 degrees of freedom
## Multiple R-squared:  0.524,Adjusted R-squared:  0.518
## F-statistic: 90.9 on 3 and 248 DF,  p-value: <2e-16
```

Each predictor has a significant effect on the outcome (body fat % using brosek's equation).We will examine the coefficient for weight more closely.

(c) *(5 pts)*
Based on the model, for a one pound increase in weight, we expect body fat % to increase by 0.18373%. The p-value of this coefficient (when H0 = 0) is <2e-16, which is essentially 0. That means, according to our model, weight has a significant positive effect on body fat % using brosek's equation.

(d) *(5 pts)*

```
### PART D ###
# Look at residual/diagnostic plots of model1
# Allign 4 plots together
par(mfrow=c(2,2))
plot(model1)
```



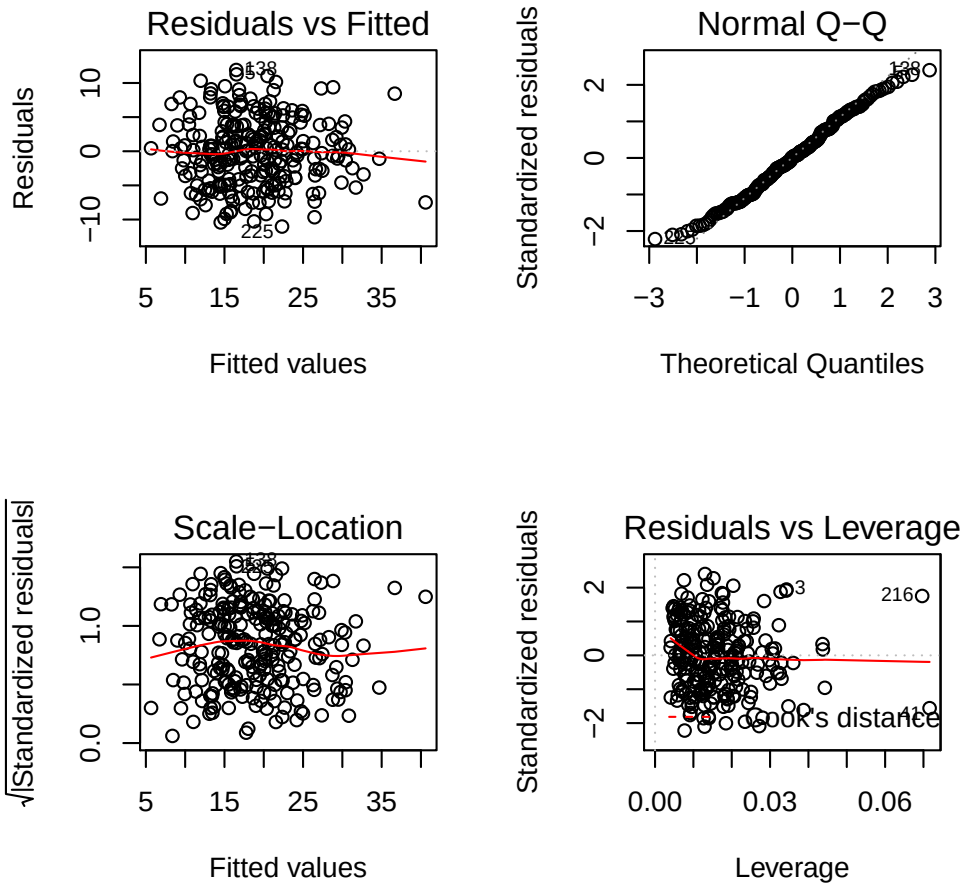
```
dev.off()

## null device
##           1
```

We are only focusing on the first plot (Residuals vs Fitted values). Based on that plot, a couple assumptions come into question: First, there is possibly a violation in linearity because the line is NOT perfectly horizontal. It becomes negative after the fitted value of about 35, but this could be due to outliers. Another possible violation is the constant variance of errors. The residuals have a few values that deviate a lot from the line, but once again, these points could be outliers.

Looking at the Normal Q-Q plot, we see that the points follow the diagonal line rather well, with the exception of a few points (which are probably outliers). This tells us that the assumption that our error terms are normally distributed seems OK.

(e) *(1 pt)*

```
### PART E ###

# It seems like the only points to remove are points 39 and 42.
body.sub = body[-c(39, 42),]

# Refit linear model with subsetted dataset
model2 = lm(bf_brosek ~ age + weight + height, data = body.sub)

# Look at summary of model2
summary(model2)

##
## Call:
## lm(formula = bf_brosek ~ age + weight + height, data = body.sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.026  -3.654   0.057   3.759  11.901
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.3198     9.6335    5.64  4.7e-08 ***
## age           0.1257     0.0260    4.84  2.3e-06 ***
## weight        0.2352     0.0137   17.12  < 2e-16 ***
## height       -1.1809     0.1464   -8.07  3.2e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.99 on 246 degrees of freedom
## Multiple R-squared:  0.584,Adjusted R-squared:  0.579
## F-statistic:  115 on 3 and 246 DF,  p-value: <2e-16

# Look at residual/diagnostic plots of model2
par(mfrow=c(2,2))
plot(model2)
```



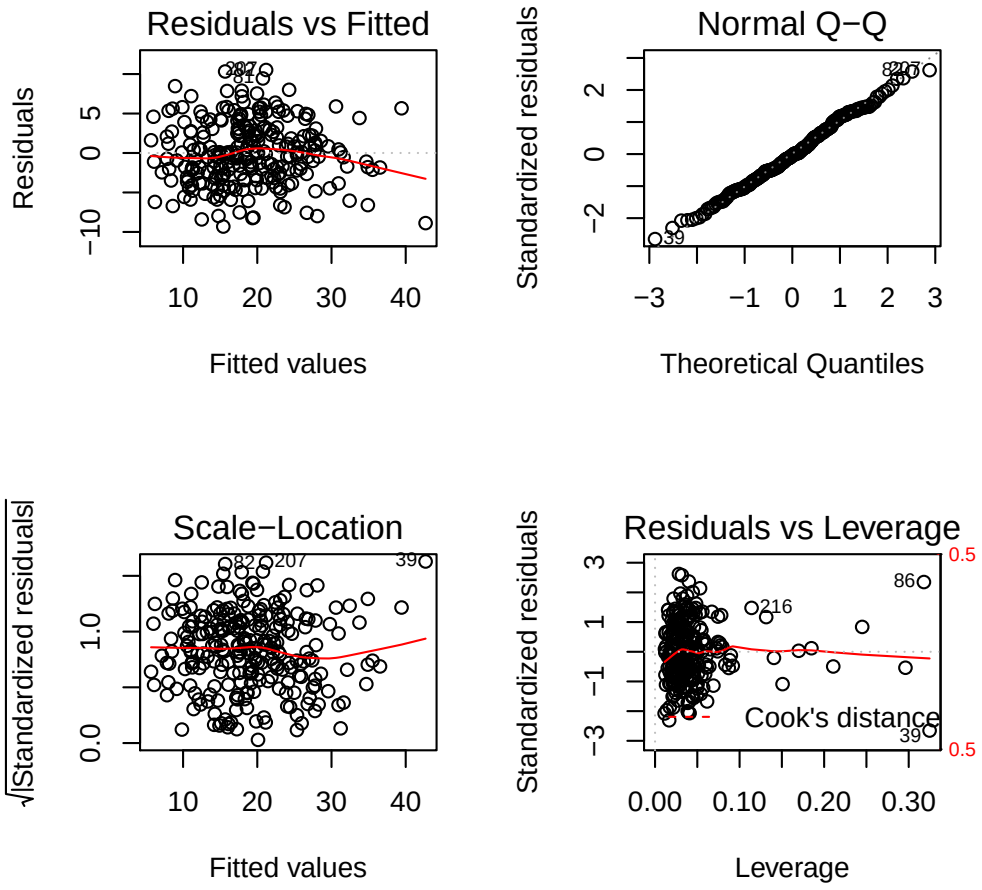

```
dev.off()

## null device
##           1
```

Based on the residual vs fitted values plot, the assumptions for a linear model seem to be OK here. The residuals do not indicate any nonlinear relationship going on, so the linearity assumption seems valid. The residuals do not flare out in the plot, indicating that there seems to be a constant variance. Based on the Normal Q-Q plot, the assumption of normality in the errors seems to be OK because the values follow the diagonal. After removing those two residuals, the fit seemed to improve and the assumptions seem valid.

(f) *(1 pt)*

```
### PART F ###

# Make design matrix of the 10 body circumference measurements
X = as.matrix(body[,10:19])

# Fit linear model with body fat % using brosek's equation vs. the 10 body circumference measurements
model3 = lm(body$bf_brosek ~ X)

# Look at summary of model3
summary(model3)

##
## Call:
## lm(formula = body$bf_brosek ~ X)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.316 -2.744 -0.158  2.839 10.515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.22875    6.21431    1.16  0.24588
## Xneck       -0.58195    0.20858   -2.79  0.00569 **
## Xchest      -0.09085    0.08543   -1.06  0.28866
## Xabdomen     0.96023    0.07158   13.41  < 2e-16 ***
## Xhip        -0.39135    0.11269   -3.47  0.00061 ***
## Xthigh       0.13371    0.12492    1.07  0.28554
## Xknee       -0.09406    0.21239   -0.44  0.65828
## Xankle       0.00422    0.20318    0.02  0.98344
## Xbiceps      0.11120    0.15912    0.70  0.48533
## Xforearm     0.34454    0.18551    1.86  0.06450 .
## Xwrist      -1.35347    0.47141   -2.87  0.00445 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.07 on 241 degrees of freedom
## Multiple R-squared:  0.735,Adjusted R-squared:  0.724
## F-statistic: 66.9 on 10 and 241 DF,  p-value: <2e-16

# Look at residual/diagnostic plots of model3
par(mfrow=c(2,2))
plot(model3)
```

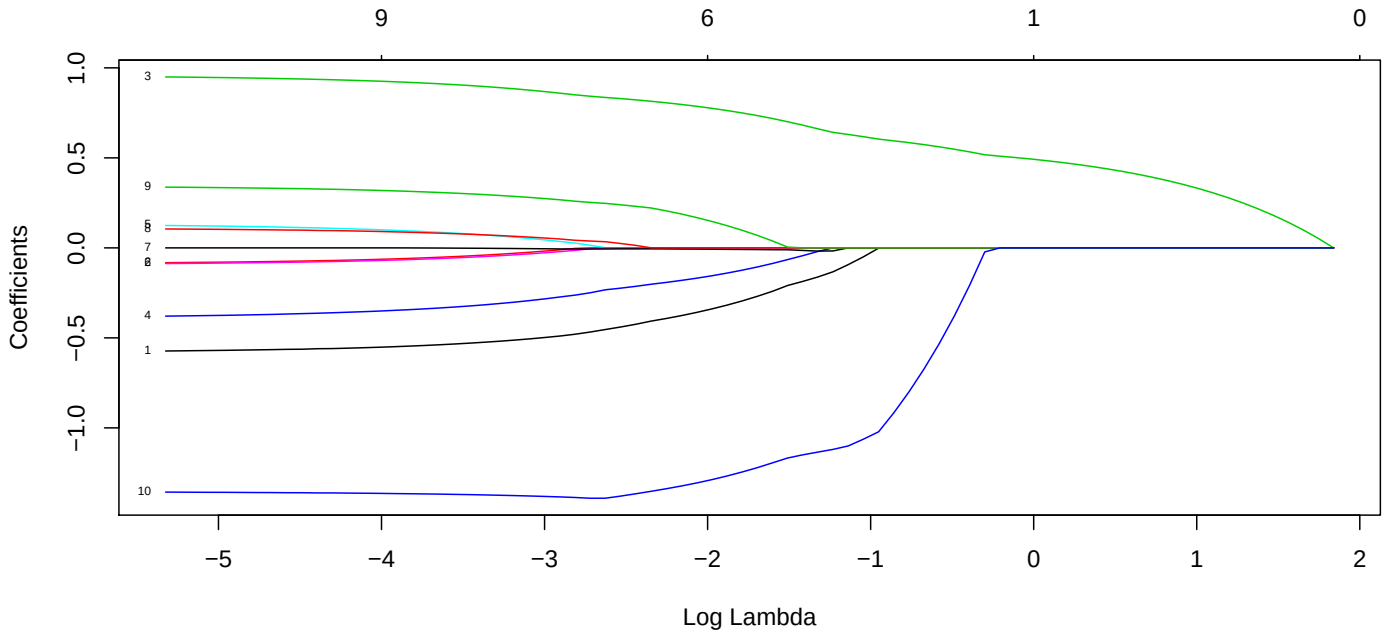



```
dev.off()

## null device
##           1
```

Based on the residuals vs fitted values plot, it seems very similar to part E. The assumptions seem valid.

(g) (EXTRA POINT QUESTION) (*1 pt*)

```
### PART G (EXTRA POINT QUESTION) ###

# Install the glmnet package if you haven't already and load it into R
# install.packages("glmnet")
library(glmnet)

## Loading required package:  Matrix
## Loading required package:  methods
## Loaded glmnet 1.9-8

# Fit lasso model (we are under the gaussian family)
model4 = glmnet(X, body$bf_brosek, family = "gaussian")

# Plot the lasso model
plot(model4, label=TRUE, cex = 1.5, xvar="lambda")
```



Based on the plot, it seems that the variables that have more importance on the model (in order) are: variable 3 (abdomen), variable 10 (wrist), variable 1 (neck), variable 4 (hip), and variable 9(forearm). The other variables' effects were shrunk, thus are deemed less important to the outcome. The plot can be interpreted as this: If you regularize the coefficients a lot (requires a big $\log(\lambda)$), then you will have an empty model because you aren't really allowing for many coefficients to enter the model. As you decrease this regularization, variables begin to enter the model based on their importance/influence on the outcome. So, as you move more to the left of the plot, it approaches the general OLS fit.

This seems to agree with summary(model3) in general because the same variables have the most significant effects on the outcome. Lasso reflects this well.

Problem 2.
(a) *(1 pt)*

```
### PART A ###

# set work directory using setwd("The path you stored your data")
setwd("~/Documents/ZhengShi/Berkeley/GSI/PH245/HW/hmwk2/hmwk2_data")
# read data
renal = read.csv(file="Data-HW2-Renal.csv", header = TRUE, sep=";")

# Let's see how many observations there are for each category of disease
table(renal$group)

##
##   1   2   3
##  23  52 125

# Subset the data to only look at category 2 and category 3 disease (removes 23 observations)
renal.sub = renal[-which(renal$group == 1),]

# Recode category 2 disease to 1 and category 3 disease to 0.
renal.sub$grp2_dummy = rep(0, nrow(renal.sub))
renal.sub$grp2_dummy[which(renal.sub$group==2)] = 1

# Fit logistic regression with group dummy variable we created in the previous step
# as outcome and the biomarkers Uvol, Ucreat, Podocin,
# Nephrin, Aqup2, and TSFb1 as predictors
```

```r
model1 = glm(grp2_dummy ~ uvol + Ucreat + podocin + nephrin + aqup2 + TGFb1,
             family = "binomial", data = renal.sub)

# Look at summary of model1
summary(model1)
```

```
##
## Call:
## glm(formula = grp2_dummy ~ uvol + Ucreat + podocin + nephrin +
##     aqup2 + TGFb1, family = "binomial", data = renal.sub)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.510  -0.872  -0.671   1.148   2.209
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.39e+00   7.57e-01   -3.15   0.0016 **
## uvol         3.99e-02   1.99e-02    2.00   0.0454 *
## Ucreat       4.46e-01   3.44e-01    1.30   0.1951
## podocin      1.66e+04   2.76e+04    0.60   0.5478
## nephrin      2.08e+04   1.16e+04    1.79   0.0733 .
## aqup2       -2.78e+03   3.88e+03   -0.72   0.4735
## TGFb1       -4.82e+03   2.25e+03   -2.15   0.0318 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 214.35  on 176  degrees of freedom
## Residual deviance: 197.33  on 170  degrees of freedom
## AIC: 211.3
##
## Number of Fisher Scoring iterations: 5
```

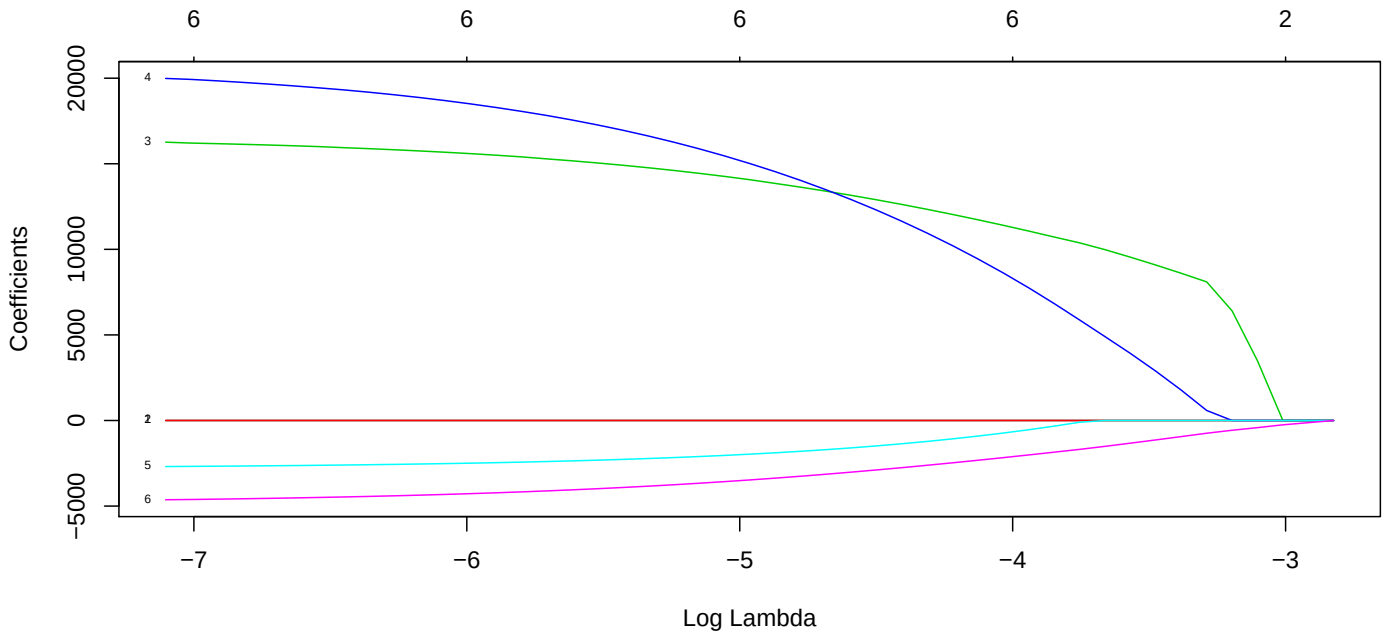
uvol and TGFb1 have a significant effect on the outcome, but there are some standard erros that are extremely high. We will look into TGFb1 more closely in part b.

(b) (*5 pts*)
The coefficient for TGFb1 is -4.823e03. This can be interpreted as: a one-unit increase in the biomarker TGFb1 decreases the log odds of having category 2 disease by -4823. In other words, for a one-unit increase in TGFb1, the odds of having category 2 disease decreases by a factor of $e(-4823)$, which is about 0 ($OR = 0$). This effect is significant at the 0.05 level with a p-value of 0.03175. This is hard to interpret because the values of TGFb1 are so small and the SE is very high. It would make a lot more sense to perform a logistic regression on the log of the values so that the coefficient could be interpretable.

(c) (*1 pt*)

```r
### PART C (EXTRA POINT QUESTION) ###

# Install the glmnet package if you haven't already and load it into R
# install.packages("glmnet")
library(glmnet)

# Make design matrix to be used in glmnet()
X = as.matrix(renal.sub[,c("uvol", "Ucreat", "podocin", "nephrin", "aqup2", "TGFb1")])

# Fit lasso to data
model2 = glmnet(X, renal.sub$grp2_dummy, family = "binomial")
```

```
# Plot the lasso fit
plot(model2, label = TRUE, xvar="lambda")
```

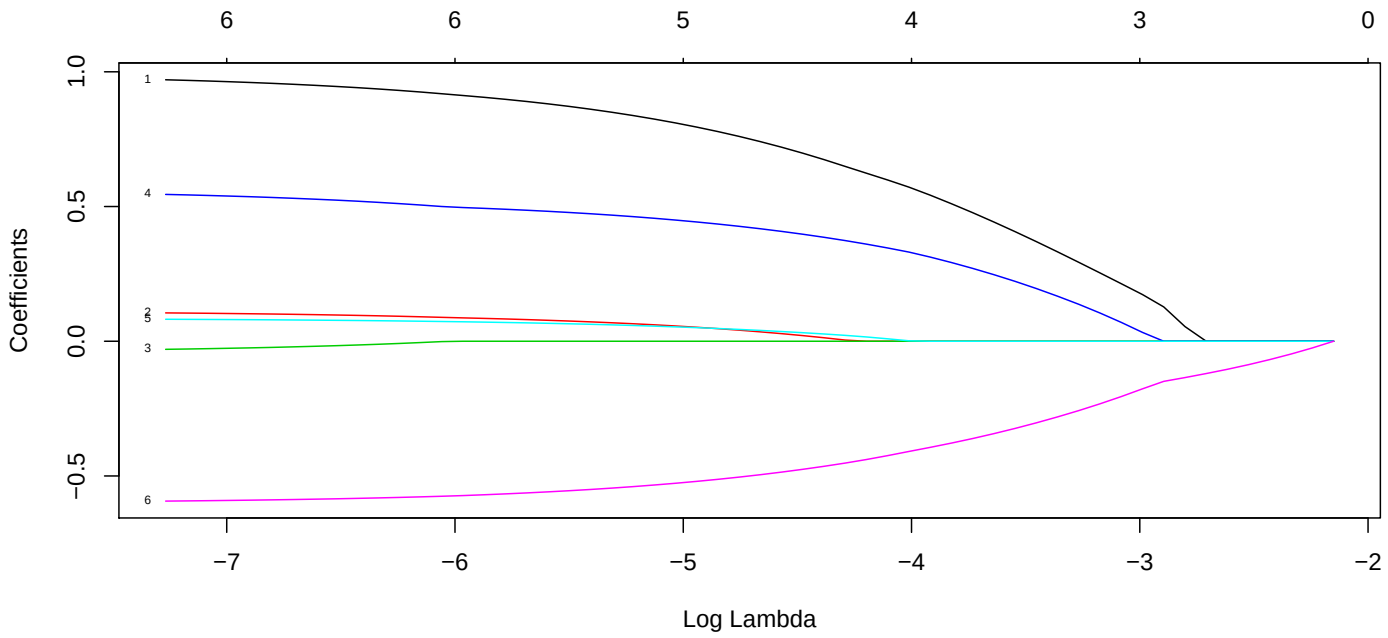


The plot does not entirely agree with model1. TGFb1 is first included in the model, which agrees with model1, but then podocin is included, which does not agree with model1. There could be some weird stuff going on due to the high standard errors and small values of some of the biomarkers.

Additional steps (Optional) It would be best to redo this entire analysis using the log values of the biomarkers to avoid high standard errors and extremely low values for some of the biomarkers.

```
model3 = glm(grp2_dummy ~ log(uvol) + log(Ucreat) + log(podocin) + log(nephrin) + log(aqup2)
             + log(TGFb1), family = "binomial", data = renal.sub)
summary(model3)

##
## Call:
## glm(formula = grp2_dummy ~ log(uvol) + log(Ucreat) + log(podocin) +
##     log(nephrin) + log(aqup2) + log(TGFb1), family = "binomial",
##     data = renal.sub)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.723  -0.788  -0.582   0.978   2.269
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.5106     2.8799   -1.22    0.223
## log(uvol)      0.9933     0.5967    1.66    0.096 .
## log(Ucreat)    0.1122     0.2922    0.38    0.701
## log(podocin)  -0.0428     0.1836   -0.23    0.815
## log(nephrin)   0.5650     0.2404    2.35    0.019 *
## log(aqup2)     0.0853     0.1442    0.59    0.554
## log(TGFb1)    -0.6013     0.1397   -4.31  1.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 214.35  on 176  degrees of freedom
## Residual deviance: 186.00  on 170  degrees of freedom
## AIC: 200
##
## Number of Fisher Scoring iterations: 5

# Now, we see that standard errors are no longer huge anymore. log(nephrin) and log(TGFb1) are
# significant. log(uvol) is almost significant. It will be significant if we use different
# significance level e.g. 0.1.

# now let's fit lasso
model4 = glmnet(log(X), renal.sub$grp2_dummy, family = "binomial")
plot(model4, label = TRUE, xvar="lambda")
```



Based on the plot, the variables in the order of importance are variable 6 (log(TGFb1)), variable 1 (log(uvol)), and variable 4 log(nephrin), which roughly matches with the logistic model above