Homework 2 - Solutions

STAT 212 (Fall 2022)

10/21/2022

Problem 1

(a)

In the following derivations I write $\sum_{i=1}^{n}$ as simply $\sum$, where the limits are assumed to be from 1 to n and the index is i.

The least squares objective is

$$\sum(\log Y_i - \beta_0 - \beta_1 x_i)^2,$$

and setting derviatives with respect to β_0 and β_1 equal to 0 gives

$$-2\sum(\log Y_i - \beta_0 + \beta_1 x_i) = 0 \quad \text{and}$$
$$-2\sum(\log Y_i - \beta_0 + \beta_1 x_i)x_i = 0.$$

Notice that these equation are the same as those for the usual simple linear regression, as in page 14 of the notes. The only difference is that instead of Y_i we have $\log Y_i$. Then the least squares estimates should be the same as in simple linear regression, only we replace Y_i with $\log Y_i$ and $\bar{Y}$ with $\bar{Y}_{\log}$:

$$\hat{\beta}_1 = \frac{\sum(\log Y_i - \bar{Y}_{\log})(x_i - \bar{x})}{\sum(x_i - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{Y}_{\log} - \hat{\beta}_1 \bar{x}$$

Problem 2

(a)

```r
source("AIC-Leaps.R")   # make sure AIC-leaps.R is in your working directory
df <- read.csv("Baseball-Salary-Data.csv")
df <- df[, -18]

leaps_ic <- leaps.AIC(df[, 2:17], df[, 1])
```

```
## [1] "AIC values"
##  [1] 5562.674 5464.568 5414.059 5403.523 5388.926 5381.472 5377.825 5377.144
##  [9] 5376.926 5377.207 5377.837 5378.910 5380.296 5381.541 5382.850 5384.824
## [1] "BIC values"
##  [1] 5574.134 5479.849 5433.159 5426.444 5415.666 5412.032 5412.206 5415.345
##  [9] 5418.947 5423.048 5427.499 5432.391 5437.598 5442.663 5447.792 5453.585
```

```r
leaps_output <- leaps(df[, 2:17], y = df[, 1], nbest = 1)
```

Model Size	AIC	BIC
1	5562.674	5574.134
2	5464.568	5479.849
3	5414.059	5433.159
4	5403.523	5426.444
5	5388.926	5415.666
6	5381.472	5412.032
7	5377.825	5412.206
8	5377.144	5415.345
9	5376.926	5418.947
10	5377.207	5423.048
11	5377.837	5427.499
12	5378.910	5432.391
13	5380.296	5437.598
14	5381.541	5442.663
15	5382.850	5447.792
16	5384.824	5453.585

Based on the table the lowest values for BIC is the 6 parameter model. BIC is penalizes the number of parameters more harshly than AIC and often gives better predictive results, so we will choose the 6 parameter model. The variables of this model are

```r
var_names <- colnames(df[, 2:17])
var_mask <- leaps_output$which[6, ]
model_vars <- var_names[var_mask]
model_vars
```

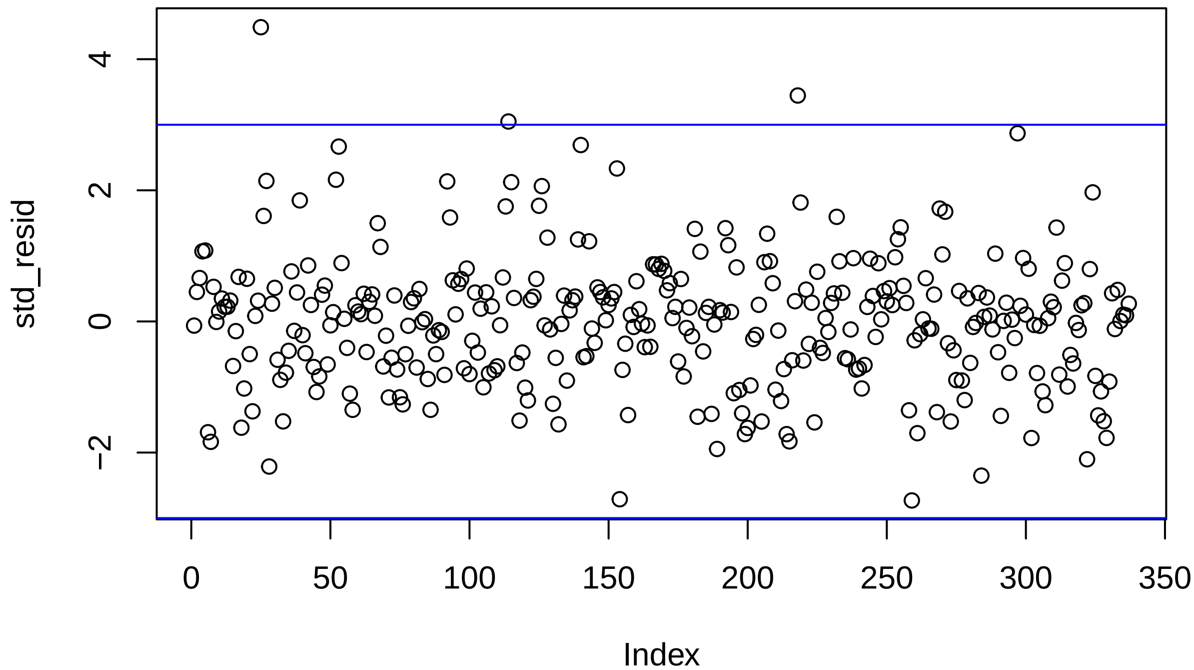
```
## [1] "home.runs"            "rbi"                  "strike.outs"
## [4] "stolen.bases"         "free.agent.eligible"  "arbitration.eligible"
```

(b)

We fit the chosen model, get standardized residuals, and plot standardized residuals with lines indicating the 3 standard deviations threshold.

```r
fit <- lm(salary ~ ., data = df[, c("salary", model_vars)])
resids <- fit$residuals
std_resid <- (resids - mean(resids)) / sd(resids)

plot(std_resid)
abline(h = c(-3, 3), col = "blue")
```



We find the indexes of the players with residuals exceeding the threshhold:

```
extremes <- which(abs(std_resid) > 3)
extremes
```

```
##  25 114 218
##  25 114 218
```

Let's look at them:

```
df[extremes, ]
```

```
##     salary batting.average on.base.percent runs hits doubles triples home.runs
## 25    6100           0.302           0.391  102  174      44       6        18
## 114   3600           0.235           0.353   39   67      10       0        11
## 218   5300           0.316           0.397   78  153      35       3        31
##     rbi walks strike.outs stolen.bases errors free.agent.eligible free.agent
## 25  100    90          67            2     15                   1          1
## 114  33    48          92           14      3                   1          0
## 218 100    65         121            6      7                   1          1
##     arbitration.eligible arbitration
## 25                     0           0
## 114                    0           0
## 218                    0           0
```

Compare to the mean of each variable:

```
colMeans(df)
```

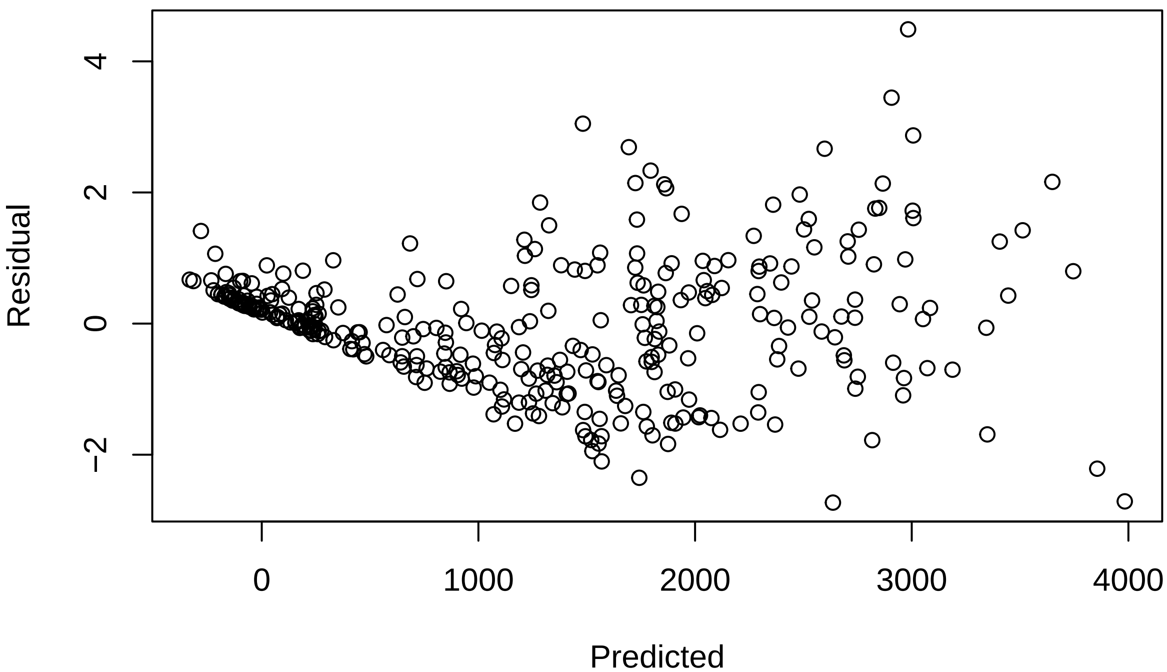
```
##               salary      batting.average      on.base.percent
##         1.248528e+03         2.578249e-01         3.239733e-01
##                 runs                 hits              doubles
##         4.669733e+01         9.283383e+01         1.667359e+01
##              triples            home.runs                  rbi
##         2.338279e+00         9.097923e+00         4.402077e+01
##                walks          strike.outs         stolen.bases
```

```
##          3.501780e+01          5.670623e+01          8.246291e+00
##                errors  free.agent.eligible            free.agent
##          6.771513e+00          3.976261e-01          1.157270e-01
## arbitration.eligible           arbitration
##          1.928783e-01          2.967359e-02
```

These players have much higher salaries than what the model predicted. They have much higher rbis than average. They also have more walks than average.

(c)

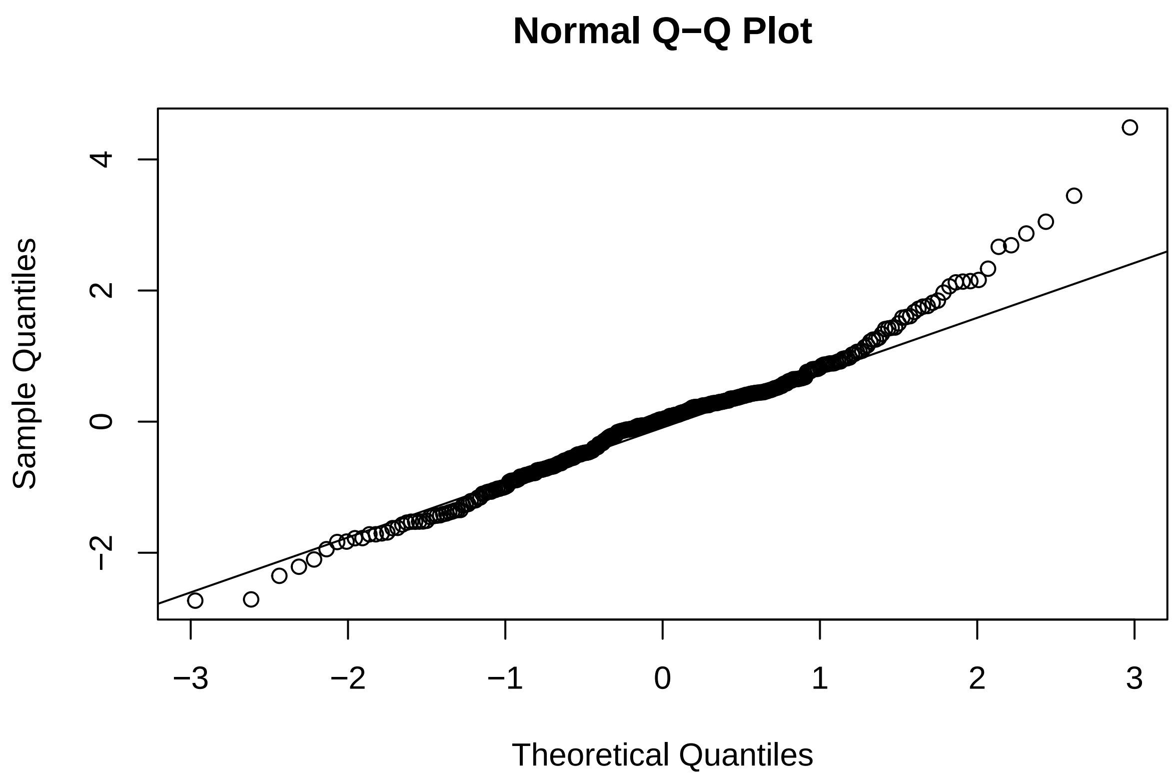
```
plot(fit$fitted.values, std_resid, xlab = "Predicted", ylab = "Residual")
```



We see that the variance of residuals increases for increasing predicted value, violating the homoskedasticity assumption of linear regression.

(d)

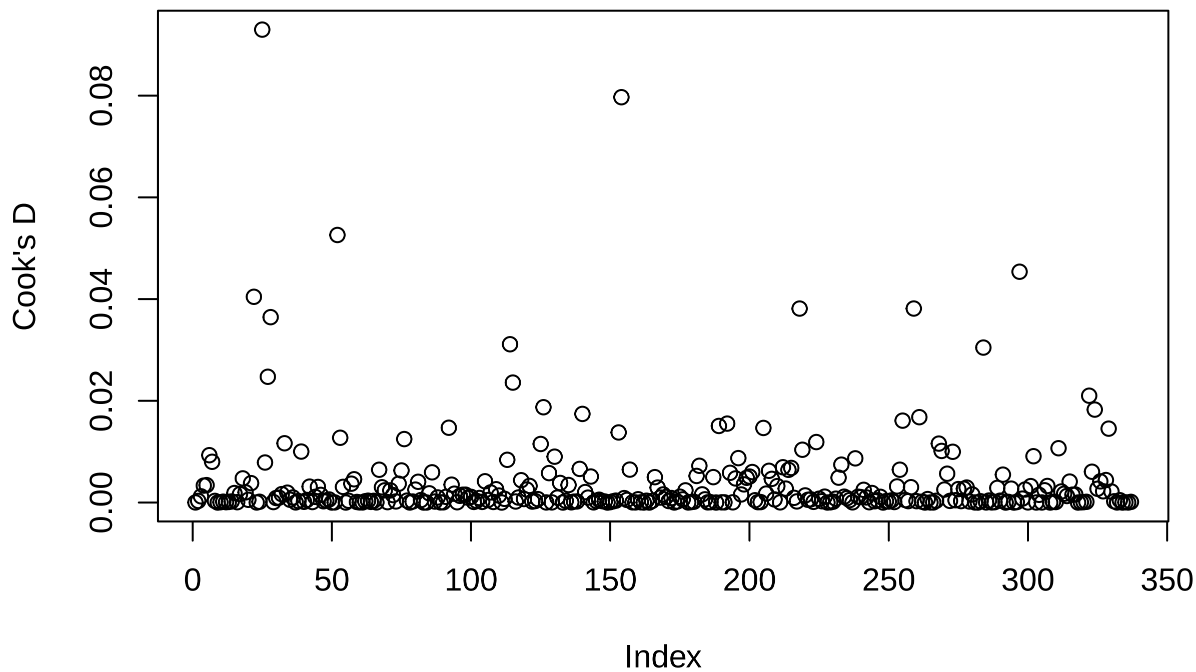
```
qqnorm(std_resid)
qqline(std_resid)
```



We see that the higher quantiles of the standardized residual distribution do not match the higher quantiles of a standard normal distribution, suggesting a violation of the normality assumption for residuals.

(e)

```
cd <- cooks.distance(fit)
plot(cd, ylab = "Cook's D")
```



No observations have Cook's D greater than 1.5, so no points are influential.