Solution to Homework 2
Course : Stat-201

1. Divide the $i^{th}$ case by $\sqrt{w_i}$ to get

$$\tilde{y} = \tilde{X}B + \tilde{e},$$

where the $i^{th}$ row of $\tilde{X}$ is $(1x_i)/\sqrt{w_i}$, and the $i^{th}$ entry of $\tilde{e}$ is $e_i/\sqrt{w_i}$. Since $e \sim N(0, \sigma^2I)$, hence we are back to the homoscedastic situation. So we can apply the formulas given in class with $X$ replaced by $\tilde{X}$ everywhere.

2. One possible rule of thumb that is generally accepted is as follows. Any point that does not conform to the general trend of the other points, and is far from the main data cloud along the $x$-direction is highly likely to be a leverage point.

For the second part of the problem we can use the deleted residuals:

$$\tilde{r}_i = y_i - \hat{y}(i).$$

A better alternative is to use the RSTUDENT values, which are obtained by scaling these.

3. The following function computes the RSTUDENT values.

```r
rstudent <- function(X,y) {
  fit <- lm(y~X)
  r <- resid(fit)
  h <- diag(X%*%solve(t(X)%*%X,t(X)))
  s.sq <- sum(r*r)/fit$df.residual
  deleted.s.sq <- (n-p)*s.sq-r*r/(1-h)
  rstar <- r/sqrt(deleted.s.sq*(1-h))
  rstar
}
```

After storing the data in the `dat` matrix we invoke it as

```r
> rstar_rstudent(dat[,,-1],dat[,1])
```

The plot shows one clear leverage point.
4. Let us write the model as
\[ y = x\beta_1 + e. \]
Then we know that under iid normal distribution of the \(e_i\)’s the mle of \(\beta_1\) is given by the LSE:
\[ \hat{\beta}_1 = (x'x)^{-1}x'y. \]
Direct maximization shows that the mle of \(\sigma^2\) is
\[ \frac{1}{n} \sum (y_i - \hat{\beta}_1 x_i)^2. \]
Under the new model
\[ E(y) = X\beta, \]
where \(X = (1 \ x)\). So
\[ E(\hat{\beta}_1) = (x'x)^{-1}x'E(y) = (x'x)^{-1}x'X\beta. \]
Now
\[ (x'x)^{-1}x'X = (x'x)^{-1}(x'1, x'x) = (\frac{\sum x_i}{\sum x_i^2}, 1), \]
which is not same as \((0,1)\) in general. So \(\hat{\beta}_1\) is biased under the new model.

5. \(\hat{Y} = x'\hat{\beta}\), where \(x = (1 \ x)\). So
\[ E(\hat{Y}) = x'E(\beta) = x\beta \]
and \(V(\hat{Y}) = x'V(\beta)x = \sigma^2 x'(X'X)^{-1}x.\)

6. We store the data in the matrix \(dat\), and make the plots using the commands:
\[ > \text{par(mfrow=c(3,5))} \]
\[ > \text{for(i in 2:14) plot(dat[,i],dat[,1])} \]
Our model is

\[ y_i = \beta_0 + \sum_{j=1}^{13} \beta_j x_{ji} + \epsilon_i, \]

where \( y \) is the response, and \( x_j \)'s are the predictors. We assume that \( \epsilon_i \)'s are iid \( N(0, \sigma^2) \).

> fit <- lm(dat[,1] ~ 1+dat[, -1])
> summary(fit)

The relevant part of the summary is given below.

Coefficients:

|                |   Value | Std. Error |    t value | Pr(>|t|) |
|----------------|--------|------------|------------|----------|
| (Intercept)    | -691.8376 | 155.8879    | -4.4380     | 0.0001   |
| dat[, -1]V2    |  1.0398  |  0.4227     |  2.4599     | 0.0193   |
| dat[, -1]V3    | -8.3083  | 14.9116     | -0.5572     | 0.5812   |
| dat[, -1]V4    |  1.8016  |  0.6497     |  2.7732     | 0.0091   |
| dat[, -1]V5    |  1.6078  |  1.0587     |  1.5187     | 0.1384   |
| dat[, -1]V6    | -0.6673  |  1.1488     | -0.5808     | 0.5653   |
| dat[, -1]V7    | -0.0410  |  0.1535     | -0.2673     | 0.7909   |
| dat[, -1]V8    |  0.1648  |  0.2099     |  0.7850     | 0.4381   |
| dat[, -1]V9    | -0.0413  |  0.1295     | -0.3187     | 0.7520   |
| dat[, -1]V10   |  0.0072  |  0.0639     |  0.1123     | 0.9112   |
| dat[, -1]V11   | -0.6017  |  0.4372     | -1.3763     | 0.1780   |
| dat[, -1]V12   |  1.7923  |  0.8561     |  2.0935     | 0.0441   |
| dat[, -1]V13   |  0.1374  |  0.1058     |  1.2979     | 0.2033   |
We get $\hat{\sigma} = 21.94$ from the output line.

Residual standard error: 21.94 on 33 degrees of freedom

- We first compute the studentized residual values. Note the use of the `hat()` function.

```r
> stud.res <- resid(fit)/(21.94*sqrt(1-hat(dat[,1])))
```

The plot of the studentized residuals against the response variable shows an upward linear trend. This does not really mean a bad fit. It can be explained as follows. The plot of studentized residual against response is a proxy for the plot of $e_i$'s against $y_i$'s. Now $y_i$ has two components $\beta_0 + \sum \beta_j x_{ji}$ and $e_i$. Ideally there should not be any relationship between the first part and $e_i$. But the presence of the second part contributes a $45^\circ$ slope to plot. That’s why residuals should be plotted against $\hat{y}_i$.

- Two abruptly high standard errors signal potential presence of collinearity. To check this we plot the ridge trace using the command

```r
> ridge(scale(dat[,1]),scale(dat[,1]),0:10)
```

where the `ridge()` function is from problem 8 below. The steep changes in the curves for $\beta_4$ and $\beta_5$ show that these are involved in some collinearity.
7. The following command does it all.

```r
> fit_lm(y~1+x+x^2+x^3)
> summary(fit)
```

From the output of `summary` we see

Coefficients:

|                | Value  | Std. Error | t value | Pr(>|t|) |
|----------------|--------|------------|---------|----------|
| (Intercept)    | 1.0003 | 0.0664     | 15.0721 | 0.0000   |
| x              | 2.2060 | 0.0913     | 24.1732 | 0.0000   |
| I(x^2)         | -2.0045| 0.0423     | -47.3947| 0.0000   |
| I(x^3)         | 0.1082 | 0.0280     | 3.8681  | 0.0002   |

The p-value for testing if $\beta_3$ is non-zero, is less than 5%. So we conclude that the cubic term is really important.

8. We use the following function for making the ridge trace.

```r
ridge_function(X,y,lambda)
{
  n <- ncol(X)
  beta_matrix(0,length(lambda),n)
  XX <- t(X)%*%X
  iden <- diag(rep(1,n))
  for(i in 1:length(lambda))
    beta[i,] <- solve(XX+lambda[i]*iden,t(X)%*%y)
  plot(lambda,beta[,1],ylim=range(beta),type="l",ylab="beta_hat")
}
```
The ridge trace has already been shown. From the ridge trace we consider 1.5 to be reasonable value for $\lambda$. We use this value to get the estimates:

```r
> ridge(dat[,1],dat[,1],1.5)
[1,] 0.6157267 -0.9019253 1.348445 2.546494 -1.338546 0.08978109 -0.4377699
[1,] -0.2454139 -0.01914037 0.04234499 0.7195551 0.01556076 0.5592142
```