# STAT5703 HW1 Ex4

*Chao Huang (ch3474), Wancheng Chen (wc2687), Chengchao Jin (cj2628)*

## Exercise 4.

### Load dataset

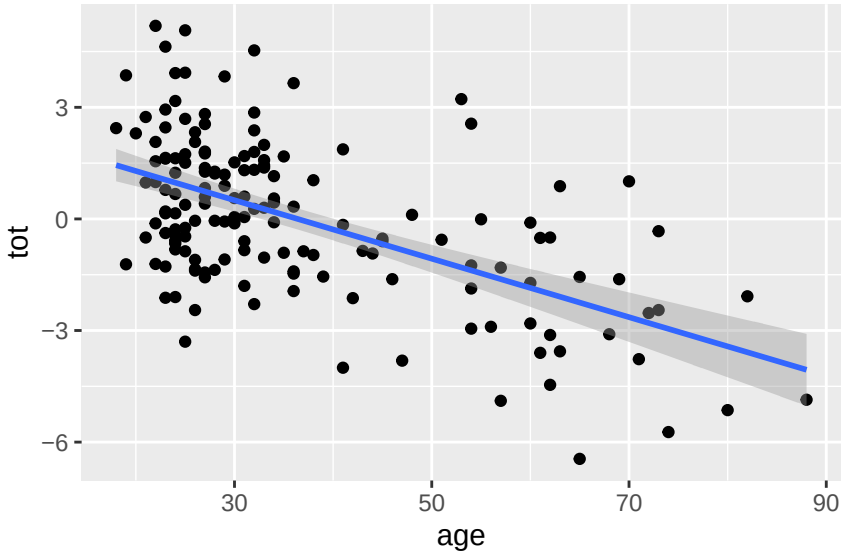
```
lines <- readLines("kidney.txt")
```

```
numbers_vec <- lapply(lines,
  function (line)  stringr::str_extract_all(line, "[-]?\\d+[\\.]?\\d*")) %>%
  unlist(recursive = FALSE) %>%
  Filter(f = function(x) length(x) == 3) %>%
  Map(f = function(x) lapply(x, as.numeric))
```

```
df <- do.call(rbind.data.frame, numbers_vec)
colnames(df) <- c("id", "age", "tot")
rownames(df) <- df$id
```

### Question 1.

```
library(ggplot2)
scatterPlot <- ggplot(df, mapping = aes(x=age, y=tot)) +
  geom_point() +
  geom_smooth(method='lm')
scatterPlot
```




The scatter plot shows that "age" and "tot" have a negative relationship and it could be fitted with a linear model.

### Question 2.

I would like to use tot as the response variable because it is more intuitively reasonable to say that the tot function is affected by age.

**Question 3.**

```
Corr = cor(df$age, df$tot)
Corr
```

```
## [1] -0.5718387
```

Without any calculation, I expect the intercept to be positive and the slope to be negative. First, as age variable increases, the overall function of kidney, tot, tends to be lower and therefore the slope should be negative. Then, around age=20, the values of tot scatter around tot=0. So clearly, since the slope is negative, the value of tot should be larger than 0 at age=0. Intuitively, the overall function of kidney for a baby should be positive. So the intercept should be positive.

**Question 4.**

For the first model, $\alpha$ denotes the expected value when the independent variable is 0, while $\beta$ denotes how much the response will change if the independent variable is increased or decreased by 1.

For the second model, now $\beta$ denotes how much the response will change if the **difference** between independent variable and its mean value is increased or decreased by 1. And *alpha* denotes the expected value for the mean of independent variable.

**Question 5.**

For the first model,

```
linearModel <- lm(tot ~ age, data = df)
summary(linearModel)
```

```
##
## Call:
## lm(formula = tot ~ age, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2018 -1.3451  0.0765  1.0719  4.5252
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.860027   0.359565   7.954 3.53e-13 ***
## age         -0.078588   0.009056  -8.678 5.18e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.801 on 155 degrees of freedom
## Multiple R-squared:  0.327,  Adjusted R-squared:  0.3227
## F-statistic: 75.31 on 1 and 155 DF,  p-value: 5.182e-15
```

From the model, we can see that $\alpha$=2.860027 and $\beta$=-0.078588. So, when age=0, the tot is estimated to be 2.860027. Also, for each year increase of age, the tot is estimated to decrease by 0.078588. Both parameters have p-value much smaller than 0.05 so they are both statistically significant.

For the second model,

```
cent_df <- df %>%
  dplyr::mutate(cent_age = age - mean(age))
```

```r
centralLinearModel <- lm(tot ~ cent_age, data = cent_df)
summary(centralLinearModel)
```

```
##
## Call:
## lm(formula = tot ~ cent_age, data = cent_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2018 -1.3451  0.0765  1.0719  4.5252
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001911  0.1437373  -0.001    0.999
## cent_age    -0.0785884  0.0090558  -8.678 5.18e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.801 on 155 degrees of freedom
## Multiple R-squared:  0.327,  Adjusted R-squared:  0.3227
## F-statistic: 75.31 on 1 and 155 DF,  p-value: 5.182e-15
```

From the model, we can see that $\alpha$=-0.0001911 and $\beta$=-0.078588. Since there is a shift for every independent variable for centralization, the $\alpha$ was changed while $\beta$ remains the same value. $\beta$ has p-value much smaller than 0.05 so it's still statistically significant. However, $\alpha$ has a very large p-value, which means that we don't have enough evidence to reject the null hypothesis that the intercept is a non-zero value.

### Question 6.

I would use the geometry to intepret these two parameters. Using linear algebra, one can prove that least square estimates minimize the squared distance between $X\beta$ and $y$ (both in vector form). In this way, the residual $y - X\beta$ should be orthogonal to the horizontal plane spanned by the columns of $X$. And therefore, least square estimates provide the optimal group of coefficients of the projected vector on the two-dimension plane with minimal loss.

### Question 7.

```r
beta <- as.numeric(linearModel$coefficients["age"])
alpha <- as.numeric(linearModel$coefficients["(Intercept)"])
predict <- function (age) alpha + beta * age
```

```r
predict(100)
```

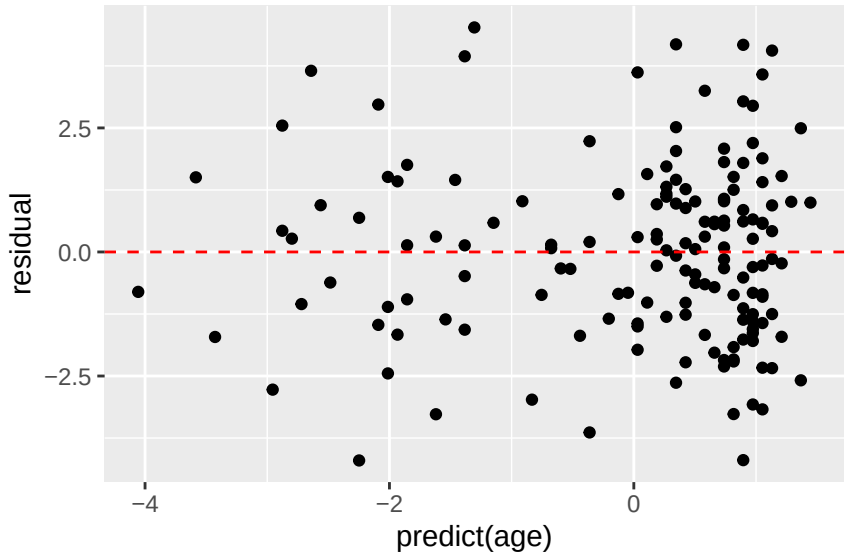
```
## [1] -4.998815
```

The prediction seems reasonable, since from the scatter plot above, the expected value for age=100 is assumed to be between -4.5 and -5.

### Question 8.

```r
res_df <- df %>%
  dplyr::mutate(prediction = predict(df$age)) %>%
  dplyr::mutate(residual = tot - prediction)
```

```
ggplot(res_df) + geom_point(aes(x=predict(age), y=residual)) + geom_hline(yintercept=0,linetype="dashed
```




The plot shows that the residuals are randomly distributed around 0, so it seems reasonable to assume that errors $\epsilon_i$ are i.i.d..

**Question 9.**

```
minus <- function(x, y) max(y,x) - min(y,x)
betaIntNormal <- Reduce(minus, confint(linearModel)[2,])
betaIntAsym <- Reduce(minus, confint.default(linearModel)[2, ])
```

First, let's assume the noise terms are all normal i.i.d. random variables. Then we have,

$$\hat{\beta} \sim N\left(\beta, \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)$$

Then by substituting $S^2$ for $\sigma^2$, $\hat{\beta}$ has a student distribution, so the confidence interval should be,

$$\hat{\beta} \pm t_{\alpha/2,n-2} \times \sqrt{\frac{MSE}{\sum (x_i - \bar{x})^2}}$$

where $MSE = \frac{1}{n-2}\sum(y_i - \hat{\alpha} - \hat{\beta}x_i)^2$. So the confidence interval can be numerically calculated as,

```
n <- nrow(df)
MSE <- 1/(n-2) * sum((df$tot - df$age * beta - alpha)^2)
sigma_square <- sum((df$age - mean(df$age))^2)
half <- stats::qt(0.025, n-2) * sqrt(MSE / sigma_square)

lb <- beta - half
rb <- beta + half
c(lb, rb)
```

```
## [1] -0.06069970 -0.09647713
```

So the confidence interval for $\beta$ is $[-0.0965, -0.0607]$.

Next, for the asymptotic and i.i.d assumption, using the results from Exercise 3 Problem 4.4, the asymptotic distribution of $\beta$ can be derived as below, with the help of Lyapunov Central Limit Theorem,

$$\sqrt{n}(\hat{\beta}_{LS} - \beta) \xrightarrow{\mathcal{D}} N(0, \sigma^2/\sigma_X^2)$$

where $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \bar{X_n}\right)^2 = \sigma_X^2$. By move $sqrt(n)$ to the right side, we have,

$$(\hat{\beta}_{LS} - \beta) \xrightarrow{\mathcal{D}} N(0, \sigma^2/(n\sigma_X^2)) = N(0, \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2})$$

which is exactly the same with the normal assumption. Therefore, the asymptotic confidence interval should be the same as well.

The second model may seem to be a weaker assumption at first sight, since it doesn't assume a normal prior on the noise terms. However, since the asymptotic distribution is exactly the same with that of normal assumption, the second model also requires $n$ goes to infinity. Therefore, the normal assumption is a more reasonable case, and it can fit all dataset regardless of the number of data samples.

**Question 10.**

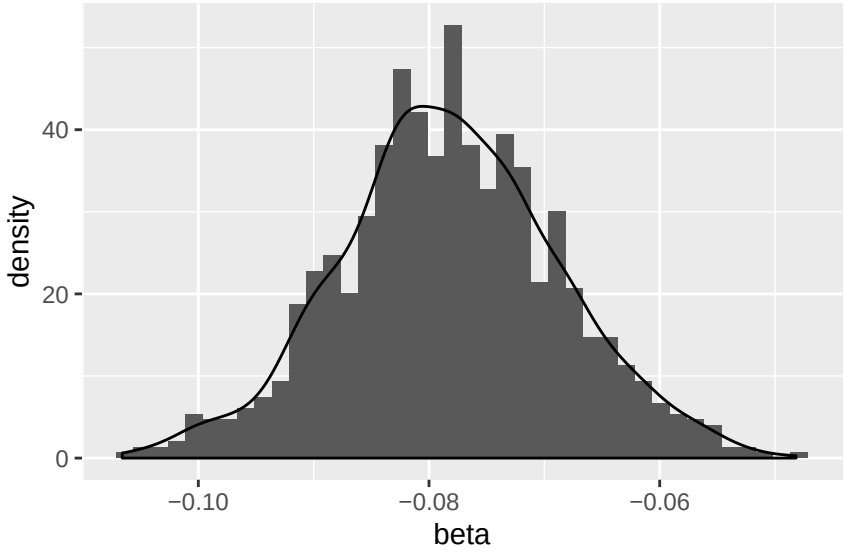
```
boot.stat <- function(data, indices){
data <- data[indices, ] # select cases in bootstrap sample
mod <- lm(tot ~ age, data=data) # refit model
coef(mod)["age"] # return coefficient vector
}
```

```
set.seed(12345) # for reproducibility
df.boot <- boot::boot(data=df, statistic=boot.stat, R=1000)
```

```
bootResult <- as.data.frame(df.boot$t) %>%
  dplyr::rename(beta=V1)
```

First we use a histogram and the corresponding density function to see the bootstrap distribution,

```
ggplot(bootResult) +
  geom_histogram(aes(x=beta,y=..density..), bins=40) +
  geom_density(aes(x=beta, y=..density..))
```




Then we can use the bootstrap results to compute the confidence interval,

```
confInts <- boot::boot.ci(df.boot)
```

```
## Warning in boot::boot.ci(df.boot): bootstrap variances needed for
## studentized intervals
```

```
confInts$basic[4:5]
```

```
## [1] -0.09837192 -0.05948389
```

The bootstrap interval is $[-0.0984, -0.0595]$, which is slightly wider than the analytical result, which is $[-0.0965, -0.0607]$.

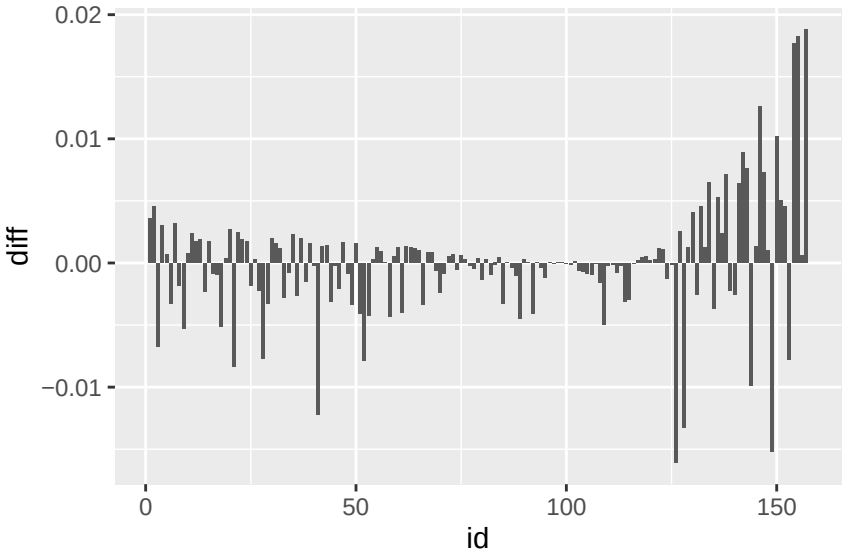
**Question 11.**

```
LeaveOneOutCorr <- function (idx)  {
  df_tmp <- df %>% dplyr::filter(id != idx)
  cor(df_tmp$age, df_tmp$tot) - Corr
}
```

```
corr_diff <- unlist(Map(LeaveOneOutCorr, seq.int(1, nrow(df))))
df_lou <- df %>% dplyr::mutate(diff=corr_diff)
```

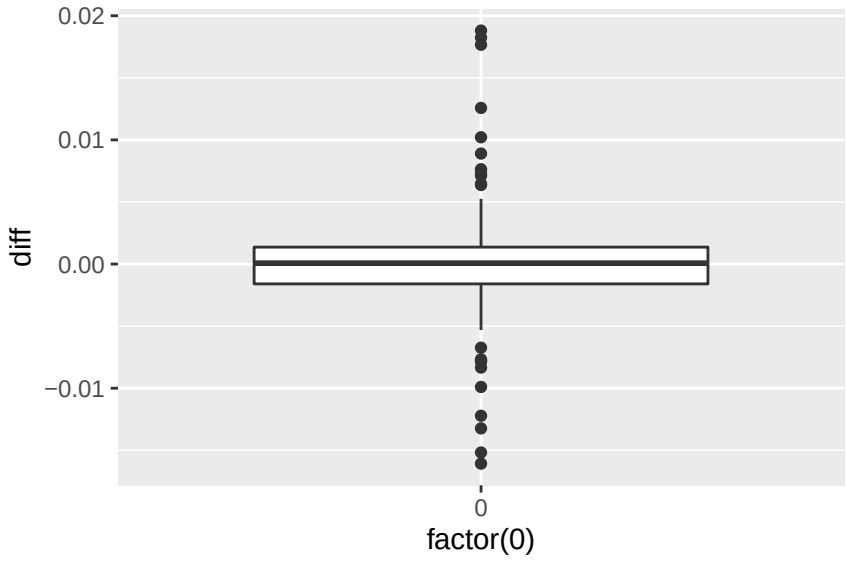
From the below plot, we may notice that there are quite large differences on the right hand side,

```
ggplot(df_lou) + geom_col(aes(x=id, y=diff))
```




We then use a boxplot to find those outliers,

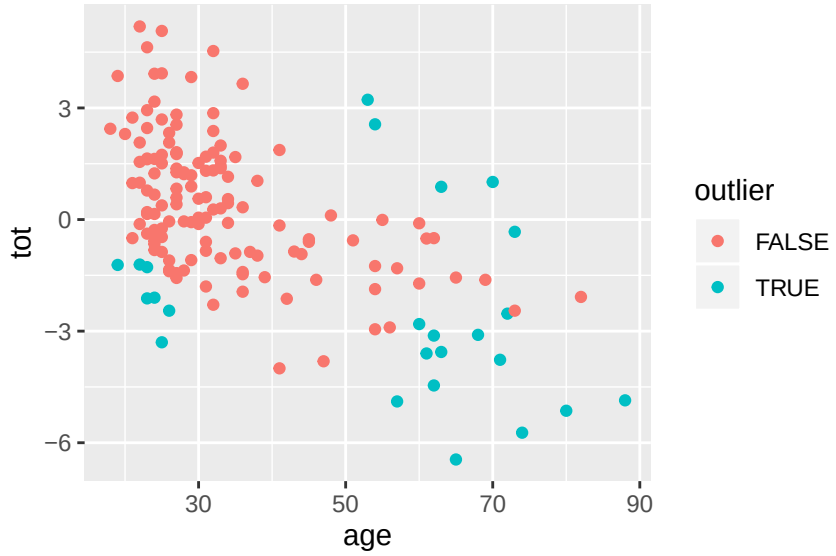
```
ggplot(df_lou, aes(x=factor(0), y=diff)) + geom_boxplot()
```




We then choose those points with absolute differences greater than 0.005 as the outliers,

```
outlierDetect <- function(corrVal) {
  ifelse (abs(corrVal) > 0.005, TRUE, FALSE)
}
df_outlier <- df_lou %>% dplyr::mutate(outlier=outlierDetect(diff))
```

```
ggplot(df_outlier) + geom_point(aes(x=age, y=tot, color=outlier))
```




In conclusion, there are some data points which are more influential than others. In the above plot, they are marked as “outlier”s with special leave-one-out differences.