STAT641_HW02_Abbas_Jalili

Abbas Jalili

3/5/2022

```
pacman::p_load(resampledata, tidyverse, boot)
```

Q.1:

The data set Bangladesh has measurements on water quality from 271 wells in Bangladesh. There are two missing values in the chlorine variable. Use the following R code to remove these two observations.

```
data("Bangladesh")
head(Bangladesh)
```

```
##   Arsenic Chlorine Cobalt
## 1    2400      6.2   0.42
## 2       6    116.0   0.45
## 3     904     14.8   0.63
## 4     321     35.9   0.68
## 5    1280     18.9   0.58
## 6     151      7.8   0.35
```

```
df <- with(Bangladesh, Chlorine[!is.na(Chlorine)])
```

1.a):

Find a 95% CI for the mean μ of chlorine levels in Bangladesh wells.

```
#finding the Chlorine's mean:

m_chio <- mean(df)

s <- sd(df)
n <- length(df)

se <- s/sqrt(n)
```

```
upper_b <- m_chio + 1.96 *se
lower_b <- m_chio - 1.96 *se

print(paste(round(lower_b , 2), round(upper_b, 2)))
```

```
## [1] "52.99 103.18"
```

We are 95% confident that the chlorine's mean is between 52.99 and 103.18.

1.b):

Find the 95% bootstrap percentile, bootstrap t, and Bca confidence intervals for the mean chlorine level, and compare results. Which confidence interval will you report?

```
set.seed(123)

theta_hat <- function(df, i){
  r <- mean(df[i])
  r.var <- var(df[i])
  return(c(r, r.var))
}

boot_mean <- boot(data = df, statistic = theta_hat, R = 5000)

boot.ci(boot_mean)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_mean)
##
## Intervals :
## Level      Normal              Basic             Studentized
## 95%   ( 52.64, 103.43 )   ( 51.51, 101.93 )   ( 57.00, 113.40 )
##
## Level     Percentile            BCa
## 95%   ( 54.23, 104.66 )   ( 56.79, 109.13 )
## Calculations and Intervals on Original Scale
```

I prefer to report the BCa as the 95% CI. I tried to check all of the types for reporting the CI in one summary table.If we check the confidence interval in different types, we can see that Normal, Basic, and Percentile are very close to the original, but Studentized and BCa are quite different. This means the data set is not following the normality. In theory BCa is the best result to report the confidence interval.

Q.2:

Dataset catsM contains a set of data on the heart weights and body weights of 97 male cats. We investigate the dependence of heart weight (in g) on body weight (in kg). The data set is available in the boot package.

2.a):

Investigate the data set by first fitting a straight line regression and creating diagnostic plots.

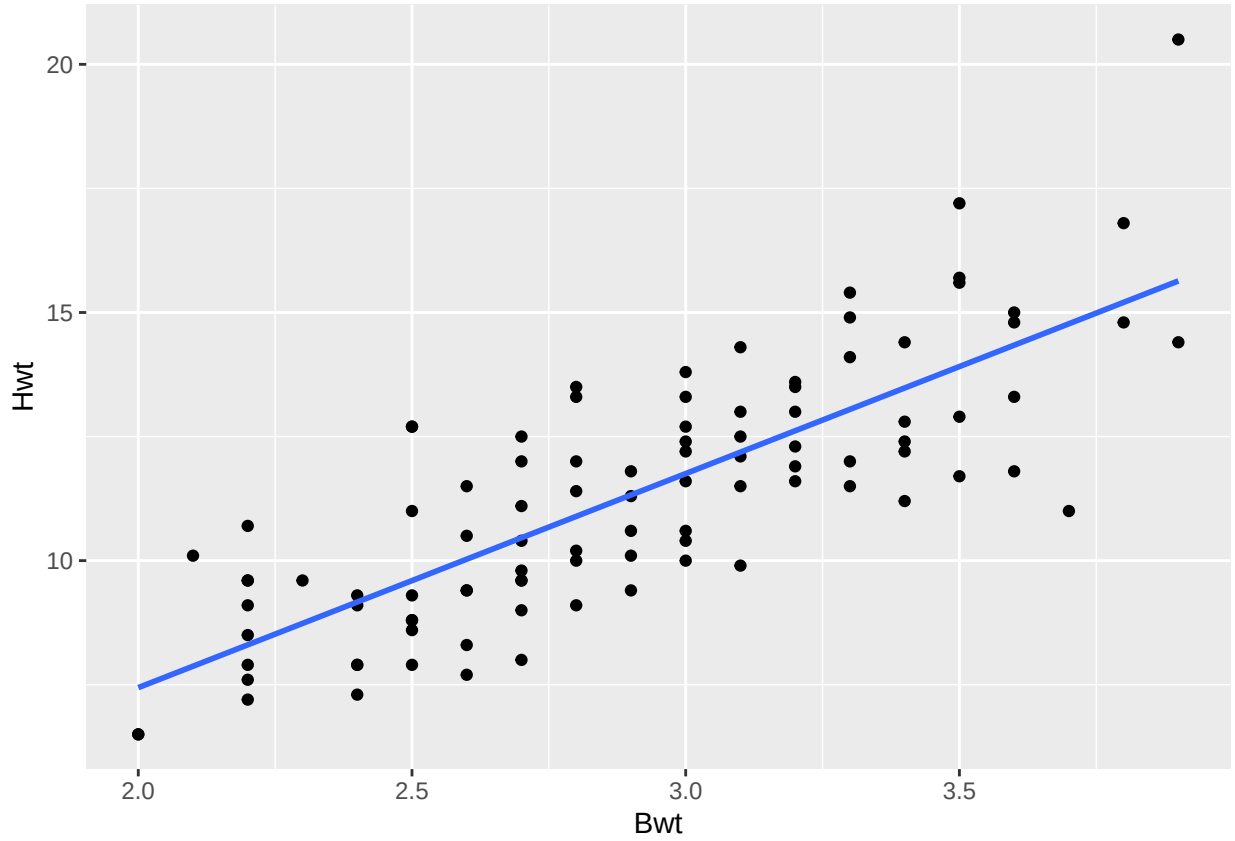
```
#checking the dataset and fit the regression line:

head(catsM, 5)
```

```
##   Sex Bwt  Hwt
## 1   M 2.0  6.5
## 2   M 2.0  6.5
## 3   M 2.1 10.1
## 4   M 2.2  7.2
## 5   M 2.2  7.6
```

```
model <- lm(Hwt ~ Bwt, data = catsM)
summary(model)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = catsM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7728 -1.0478 -0.2976  0.9835  4.8646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.1841     0.9983  -1.186    0.239
## Bwt           4.3127     0.3399  12.688   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.557 on 95 degrees of freedom
## Multiple R-squared:  0.6289, Adjusted R-squared:  0.625
## F-statistic:   161 on 1 and 95 DF,  p-value: < 2.2e-16
```

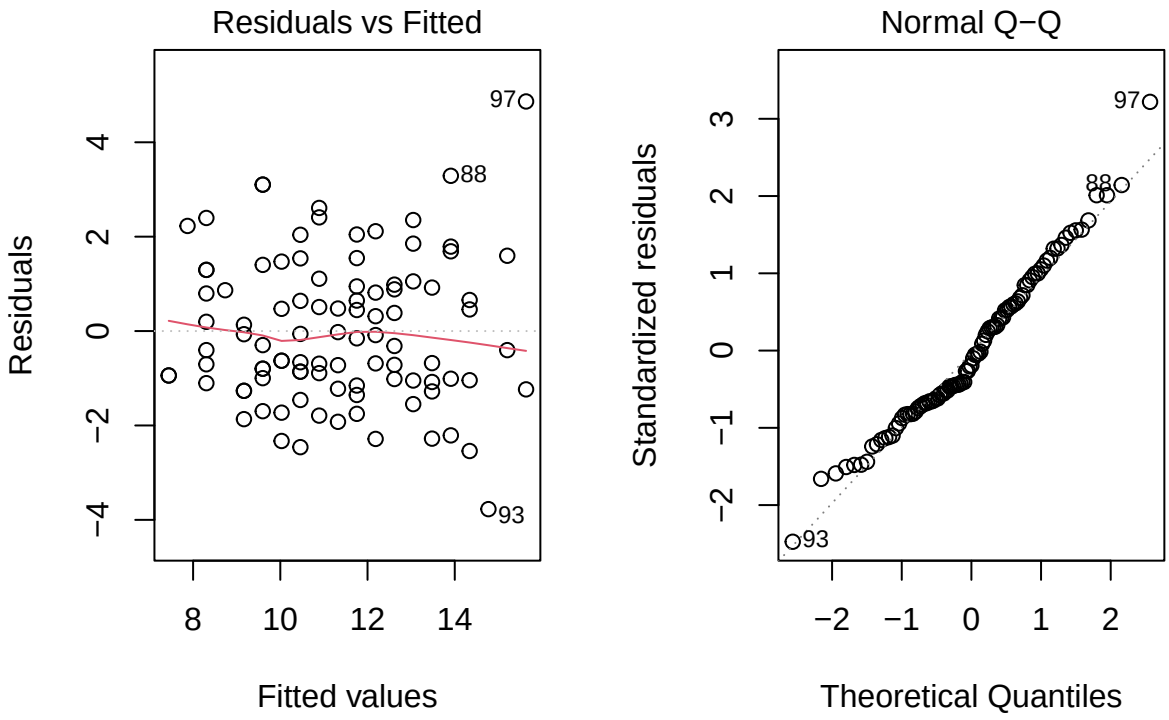
```
ggplot(data = catsM, aes(Bwt, Hwt))+
  geom_point()+
  geom_smooth(formula = y~x, method = 'lm', se = FALSE)
```

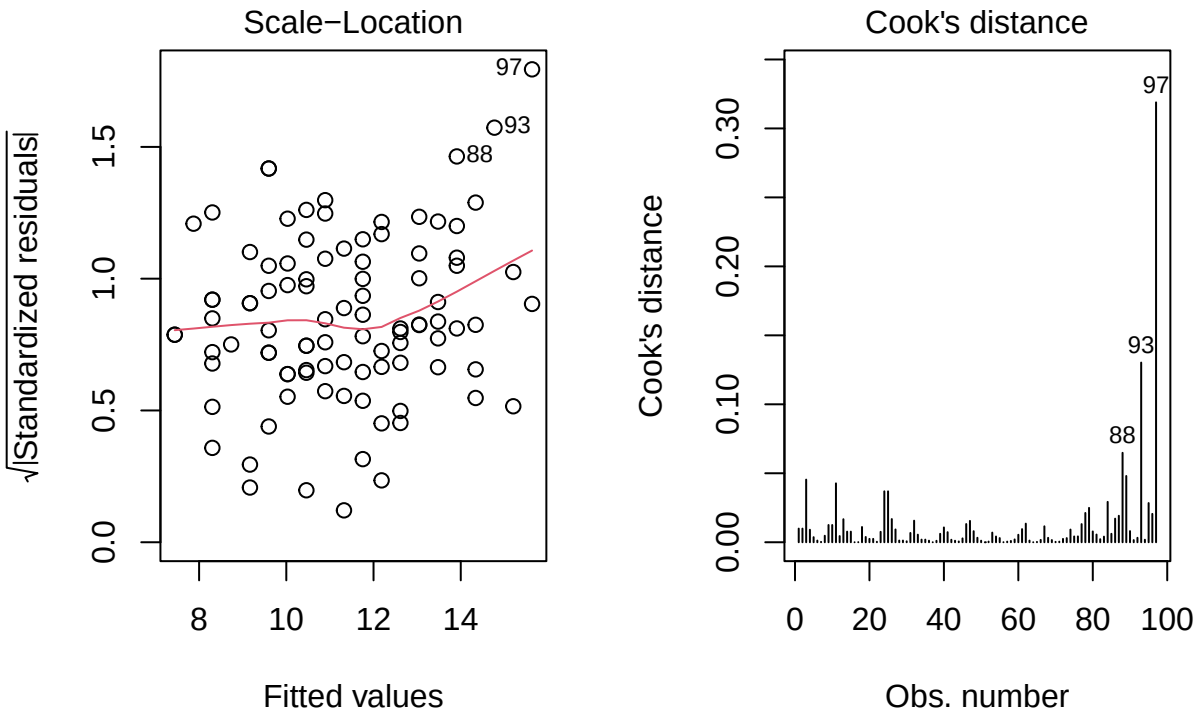


Based the regression line showed in the scatter plot above, it shows most of data point are scattered around

the regression line except two outliers. We can investigate those point in diagnostic plots in the next part:

```
#plotting the regression model:

par(mfrow = c(1,2))
plot(model, 1:4)
```





By checking the assumptions, we don't see a pattern in residual-fitted plot, but the QQ-plot shows some outliers. It is possibility of skewed distribution too. With checking the cook's distance plot, we can see three data points are far two the rest of the data points which can effect on the distribution.

2.b):

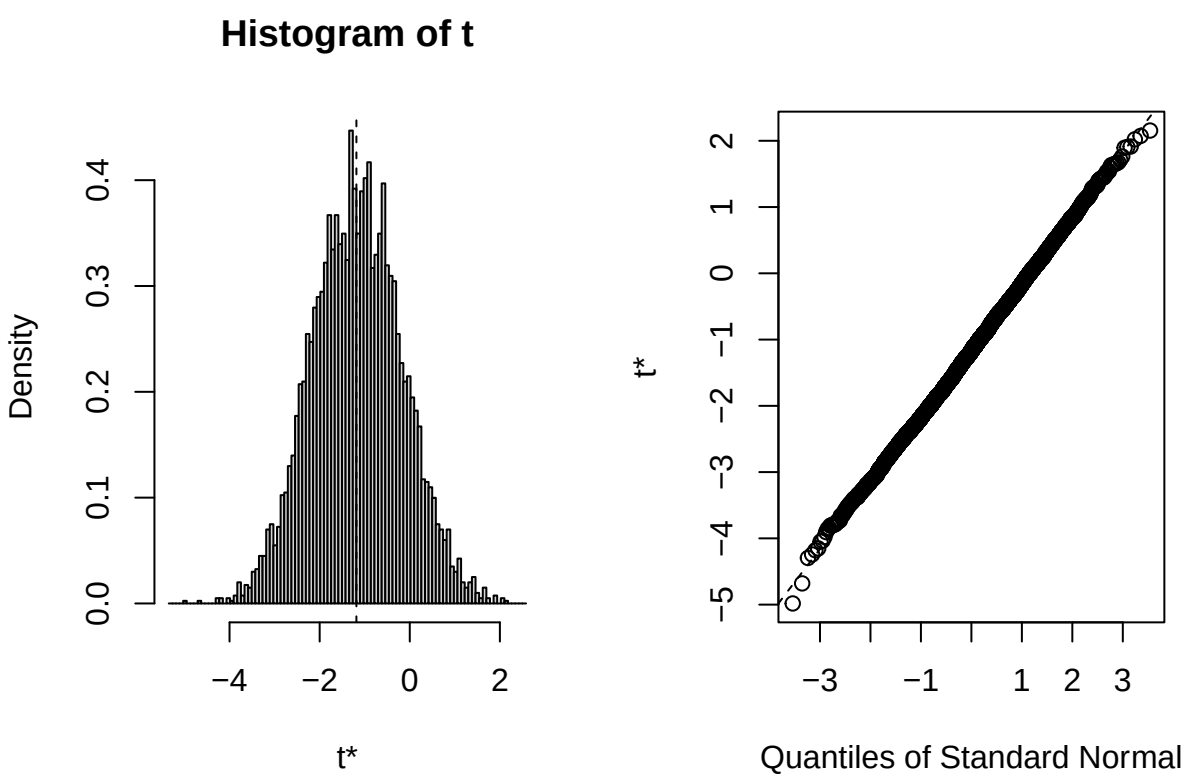
Next, perform model-based bootstrap regression (residual resampling). Are the bootstrap estimates for intercept and slopes appear normal? Is the model-based standard error for the original fit accurate?

```
set.seed(111)

fit <- fitted(model)
e <- residuals(model)
X <- model.matrix(model)

boot.fixed <- function(data, i){

  yb <- fit + e[i]
  mod <- lm(yb ~ X - 1)
  coefficients(mod)
}

catsm_fixed_boot <- boot(catsM, boot.fixed, 5000)
catsm_fixed_boot
```

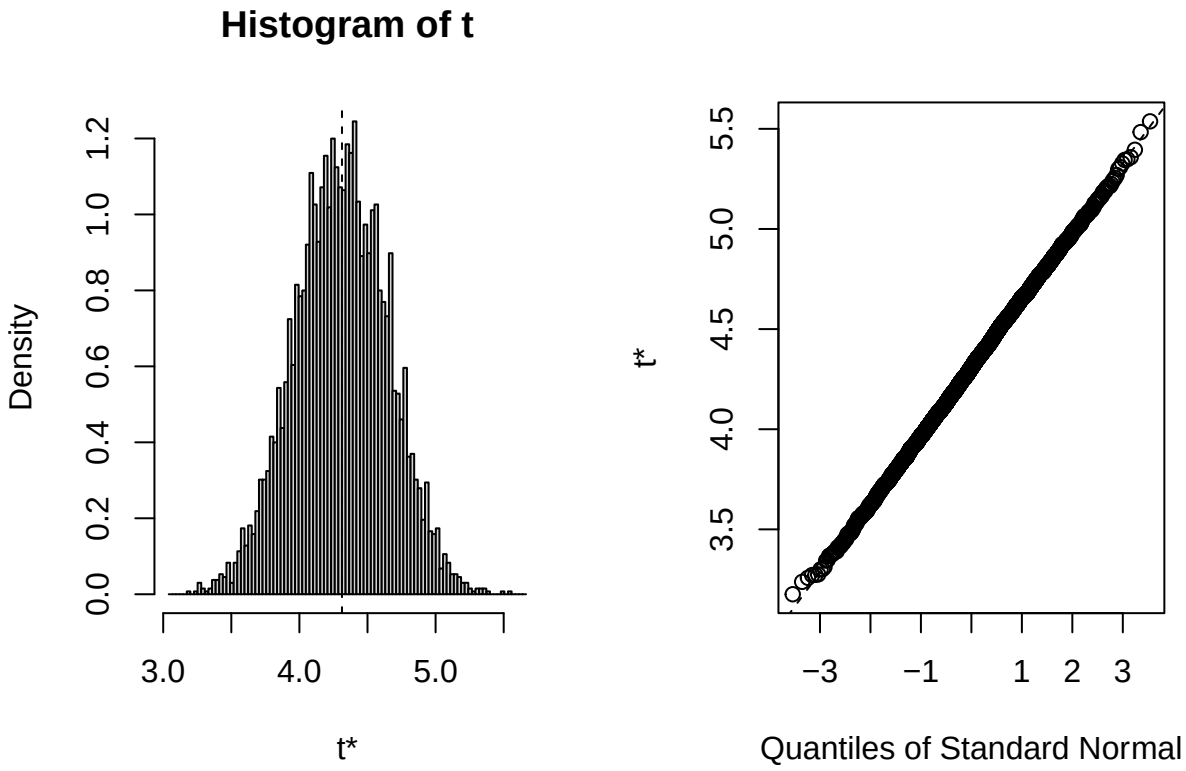
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
##
##
## Call:
## boot(data = catsM, statistic = boot.fixed, R = 5000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* -1.184088  0.017399948   0.9978046
## t2*  4.312679 -0.005595399   0.3405818
```

```
par(mfrow = c(1,2))
plot(catsm_fixed_boot, index = 1)
```




```
par(mfrow = c(1,2))
plot(catsm_fixed_boot, index = 2)
```



After performing model-based bootstrap regression, we clearly see the improvement in normality. The bootstrap estimates for intercept and slopes are appear normal.Since the original standard error and residual resampling have the almost same standard error, we can conclude that residual resampling is accurate, and we can get a normal distribution for the residual resampling.

2.c):

Do you think the results are effected by any single observation?

No. because the standard error for the original fit and bootstrap method are pretty close, those single observations are not effected the result. We could improve the normality with using the residual resampling and it is accurate also.

2.d):

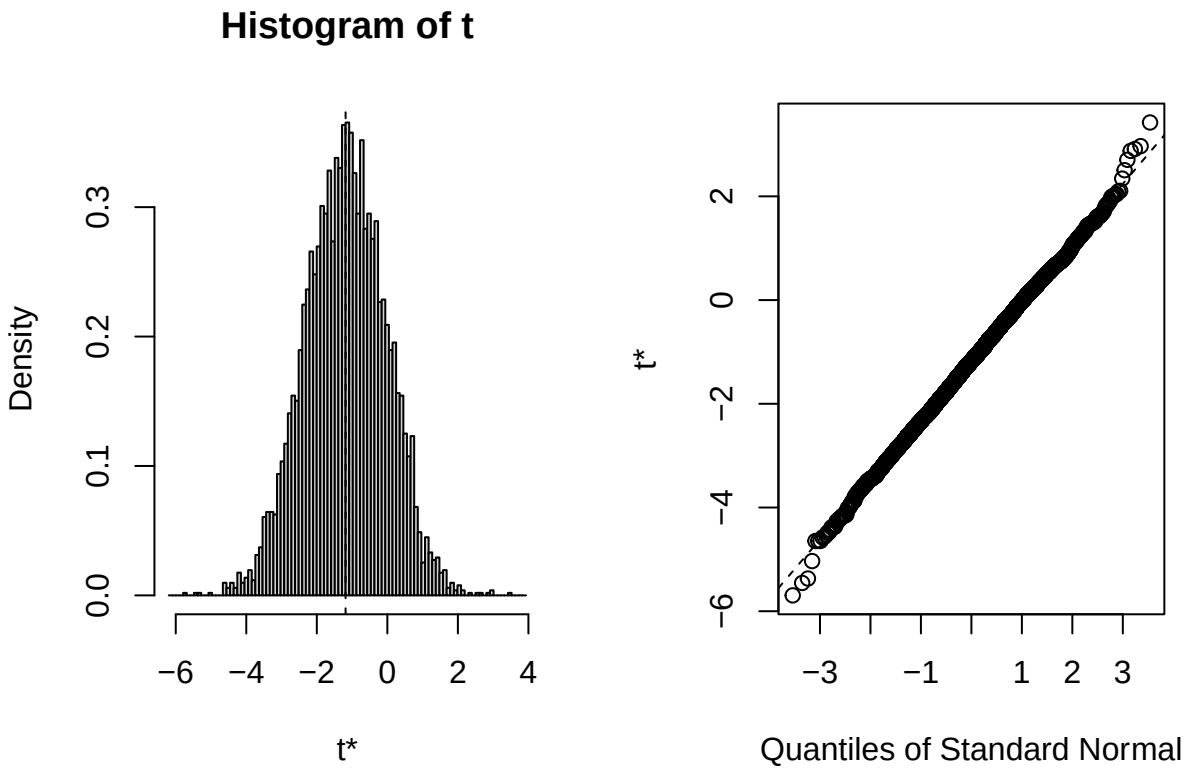
Perform the observation resampling method. And compare the results with (b) and (c).

```
set.seed(111)

boot.catsm <- function(data, i){

  data <- data[i,]
  model <- lm(Hwt ~ Bwt,data = data)
  coefficients(model)
}

model_boot <- boot(catsM, boot.catsm, 5000)
```

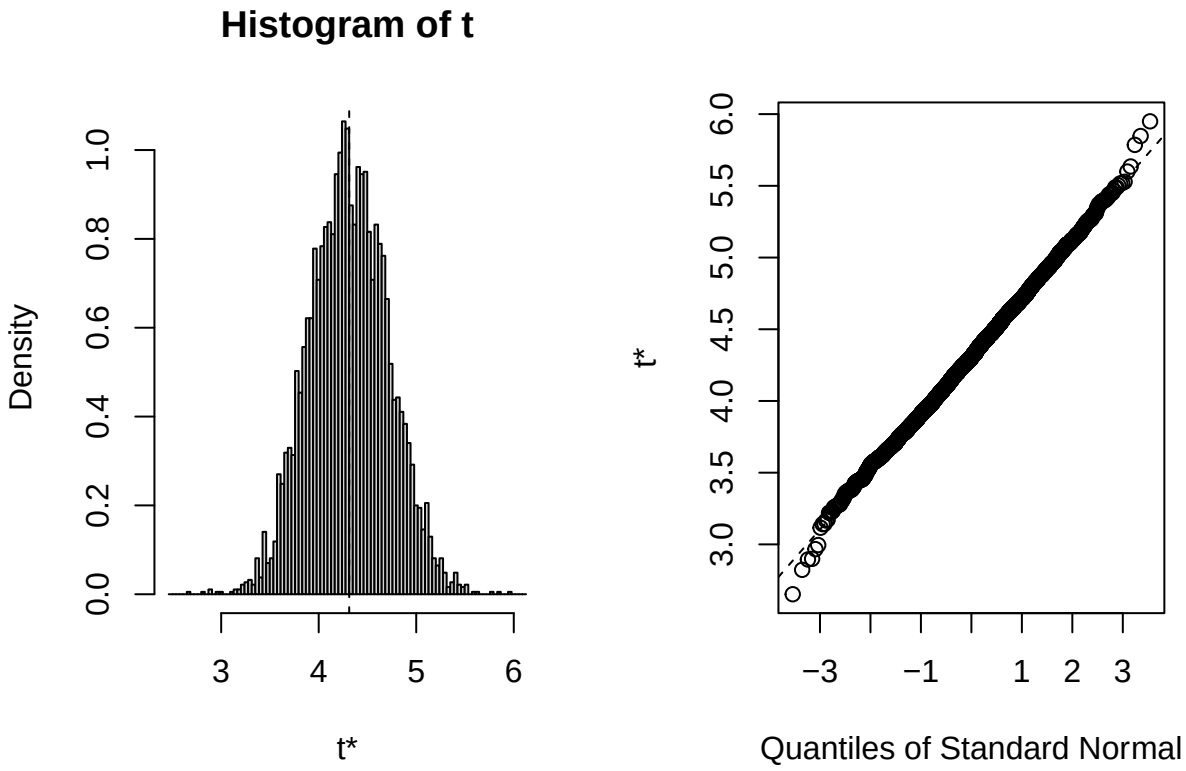
```
model_boot
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = catsM, statistic = boot.catsm, R = 5000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* -1.184088 -0.0027691248   1.1422032
## t2*  4.312679  0.0008239218   0.4030422
```

```
par(mfrow = c(1,2))
plot(model_boot, index = 1)
```




```
par(mfrow = c(1,2))
plot(model_boot, index = 2)
```



After checking the standard error for all the summary tables, we can see the residual resampling have the lowest standard error values. However, the original regression model has almost same values in standard error, but the normality assumption is satisfied in residual resampling.

Q.3:

The file Phillies2009 contains data from the 2009 season for the baseball team the Philadelphia Phillies.

```
str(Phillies2009)
```

```
## 'data.frame':    162 obs. of  7 variables:
##  $ Date      : Factor w/ 162 levels "1-Aug","1-Jul",..: 130 141 147 7 12 17 23 33 38 43 ...
##  $ Location  : Factor w/ 2 levels "Away","Home": 2 2 2 1 1 1 1 1 2 2 ...
##  $ Outcome   : Factor w/ 2 levels "Lose","Win": 1 1 2 1 2 2 2 1 1 1 ...
##  $ Hits      : int  4 6 11 7 15 13 10 5 14 8 ...
##  $ Doubles   : int  2 1 3 2 3 3 3 1 3 2 ...
##  $ HomeRuns  : int  0 0 1 1 1 2 3 0 1 3 ...
##  $ StrikeOuts: int  6 3 6 3 6 4 7 3 5 7 ...
```

3.a):

Find the mean number of strike outs per game (StrikeOuts) for the home and the away games (Location).

```
Phillies <- Phillies2009 %>%
  arrange(Location)
head(Phillies, 5)
```

```
##     Date Location Outcome Hits Doubles HomeRuns StrikeOuts
## 1 10-Apr     Away    Lose    7       2        1          3
## 2 11-Apr     Away     Win   15       3        1          6
## 3 12-Apr     Away     Win   13       3        2          4
## 4 13-Apr     Away     Win   10       3        3          7
## 5 16-Apr     Away    Lose    5       1        0          3
```

```
new_phillies <- Phillies2009 %>%
  group_by(Location)%>%
  summarize(StrikeOuts_mean = round(mean(StrikeOuts), 2),
            n =n())

new_phillies
```

```
## # A tibble: 2 x 3
##   Location StrikeOuts_mean     n
##   <fct>              <dbl> <int>
## 1 Away                7.31    81
## 2 Home                6.95    81
```

After grouping the data set based on the location, the mean of the StrikeOuts for away is 7.31 and for home is 6.95.

```
home <- subset(Phillies2009, Location== "Home", StrikeOuts)
away <- subset(Phillies2009, Location== "Away", StrikeOuts)

obs_diff <- mean(away$StrikeOuts)    - mean(home$StrikeOuts)
obs_diff
```

```
## [1] 0.3580247
```

```
t.test(away$StrikeOuts, home$StrikeOuts, alternative = 'greater', var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  away$StrikeOuts and home$StrikeOuts
## t = 0.8316, df = 160, p-value = 0.2034
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.3542521        Inf
## sample estimates:
## mean of x mean of y
##  7.308642  6.950617
```

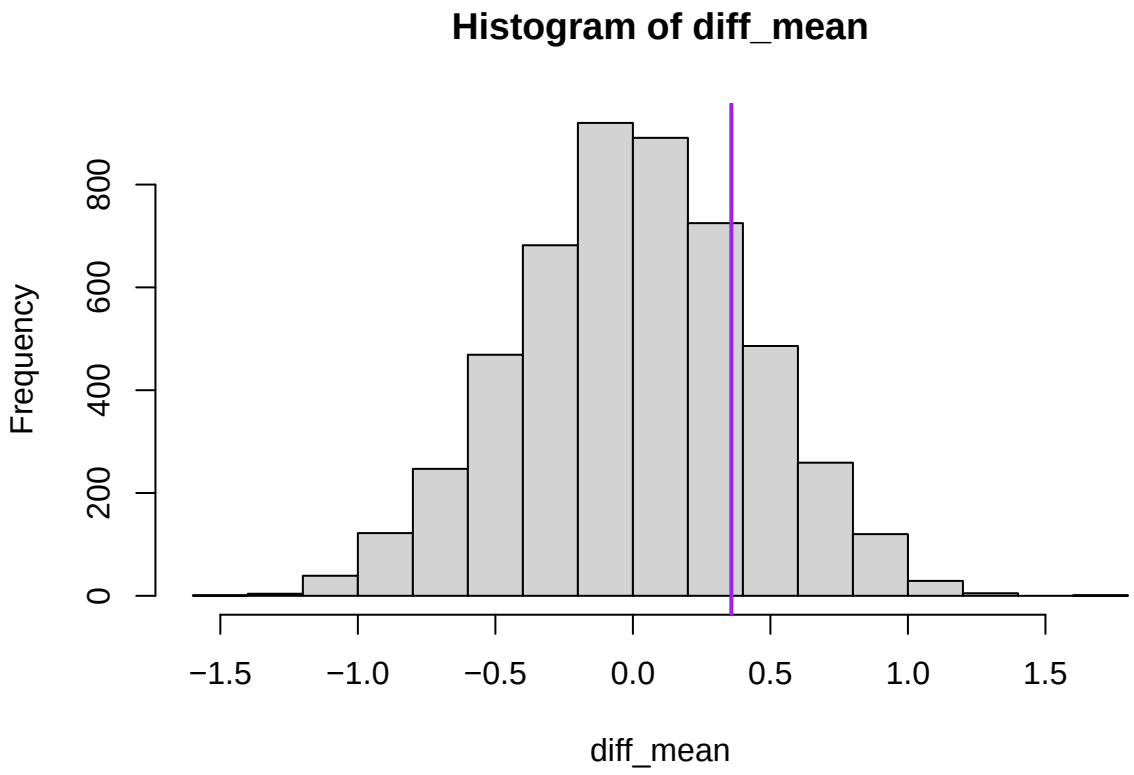
The mean difference of the observation is 0.3580247. The p-value in Two sample t-test is 0.2034 which is greater than of the alpha 0.05, so we fail to reject the null hypothesis. This means there is no difference between strikeouts means for away and home.

3.b):

Perform a permutation test to see if the difference in means is statistically significant.

```
set.seed(123)

diff_mean <- numeric()
```

```r
for (i in 1:5000) {
  pm <- sample(Phillies$StrikeOuts, 162, replace = FALSE)
  diff_mean[i] <- mean(pm[1:81]) - mean(pm[82:162])

}


hist(diff_mean)
abline(v=obs_diff, lwd=2, col="purple")
```




```r
length(diff_mean[diff_mean >= obs_diff])/5000
```

```
## [1] 0.2122
```

After performing a permutation test, the difference in means is 0.2122 which means it is statistically significant since the proportion of the permutation result is pretty close the result of the two sample t-test (0.2034).