STAT641_FINAL_REPORT

Abbas Jalili

2/28/2022

Section 1:

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

```
body_fat <- read.csv("C:/Users/AJALI/Downloads/Compressed/archive_2/bodyfat.csv")
head(body_fat, 5)
```

```
##   Density BodyFat Age Weight Height Neck Chest Abdomen   Hip Thigh Knee Ankle
## 1  1.0708    12.3  23 154.25  67.75 36.2  93.1    85.2  94.5  59.0 37.3  21.9
## 2  1.0853     6.1  22 173.25  72.25 38.5  93.6    83.0  98.7  58.7 37.3  23.4
## 3  1.0414    25.3  22 154.00  66.25 34.0  95.8    87.9  99.2  59.6 38.9  24.0
## 4  1.0751    10.4  26 184.75  72.25 37.4 101.8    86.4 101.2  60.1 37.3  22.8
## 5  1.0340    28.7  24 184.25  71.25 34.4  97.3   100.0 101.9  63.2 42.2  24.0
##   Biceps Forearm Wrist
## 1   32.0    27.4  17.1
## 2   30.5    28.9  18.2
## 3   28.8    25.2  16.6
## 4   32.4    29.4  18.2
## 5   32.2    27.7  17.7
```

```
max(body_fat$Age)
```

```
## [1] 81
```

```
min(body_fat$Age)
```

```
## [1] 22
```

```
dim(body_fat)
```

```
## [1] 252  15
```

```
sum(is.na(body_fat))
```

```
## [1] 0
```

```
set.seed(123)
body_model <- lm(Age ~ Weight+Height+Density+Neck+Chest+Abdomen+Hip+Thigh+Knee+Ankle+
                   Biceps+Forearm+Wrist+ BodyFat,data = body_fat)
summary(body_model)
```

```
##
## Call:
## lm(formula = Age ~ Weight + Height + Density + Neck + Chest +
##     Abdomen + Hip + Thigh + Knee + Ankle + Biceps + Forearm +
##     Wrist + BodyFat, data = body_fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.2848  -5.4626   0.1289   5.7231  23.4432
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -177.45849  209.00887  -0.849 0.396712
## Weight        -0.32166    0.10539  -3.052 0.002531 **
## Height        -0.37010    0.18976  -1.950 0.052315 .
## Density      146.06427  187.75564   0.778 0.437375
## Neck           0.75398    0.46406   1.625 0.105546
## Chest          0.09976    0.19767   0.505 0.614253
## Abdomen        0.97212    0.20410   4.763 3.32e-06 ***
## Hip           -0.31695    0.29138  -1.088 0.277812
## Thigh         -1.64799    0.26888  -6.129 3.66e-09 ***
## Knee           1.84321    0.46645   3.952 0.000103 ***
## Ankle         -0.70695    0.44133  -1.602 0.110523
## Biceps         0.35701    0.34211   1.044 0.297758
## Forearm       -1.09586    0.39419  -2.780 0.005872 **
## Wrist          5.85616    1.01834   5.751 2.73e-08 ***
## BodyFat        0.56903    0.43518   1.308 0.192288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.568 on 237 degrees of freedom
## Multiple R-squared:  0.5635, Adjusted R-squared:  0.5377
## F-statistic: 21.86 on 14 and 237 DF,  p-value: < 2.2e-16
```

```
#making the null and full model fo anova:
null_model <- lm(Age ~ 1, data = body_fat)
full_model <- lm(Age ~ ., data = body_fat)
# Comparing a full model with all the predictors to the null model or an initial model.
anova(null_model, full_model)
```

```
## Analysis of Variance Table
##
## Model 1: Age ~ 1
## Model 2: Age ~ Density + BodyFat + Weight + Height + Neck + Chest + Abdomen +
##     Hip + Thigh + Knee + Ankle + Biceps + Forearm + Wrist
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    251 39862
## 2    237 17398 14     22463 21.857 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Use a stepwise selection method.
step(full_model, scope = list(lower = null_model, upper = full_model), trace = 0)
```

```
##
## Call:
## lm(formula = Age ~ BodyFat + Weight + Height + Neck + Abdomen +
##     Thigh + Knee + Ankle + Forearm + Wrist, data = body_fat)
##
## Coefficients:
## (Intercept)      BodyFat       Weight       Height         Neck      Abdomen
##    -27.9872       0.2724      -0.3322      -0.3597       0.9427       0.9344
##       Thigh         Knee        Ankle      Forearm        Wrist
##     -1.7646       1.7613      -0.7560      -0.8997       6.0410
```

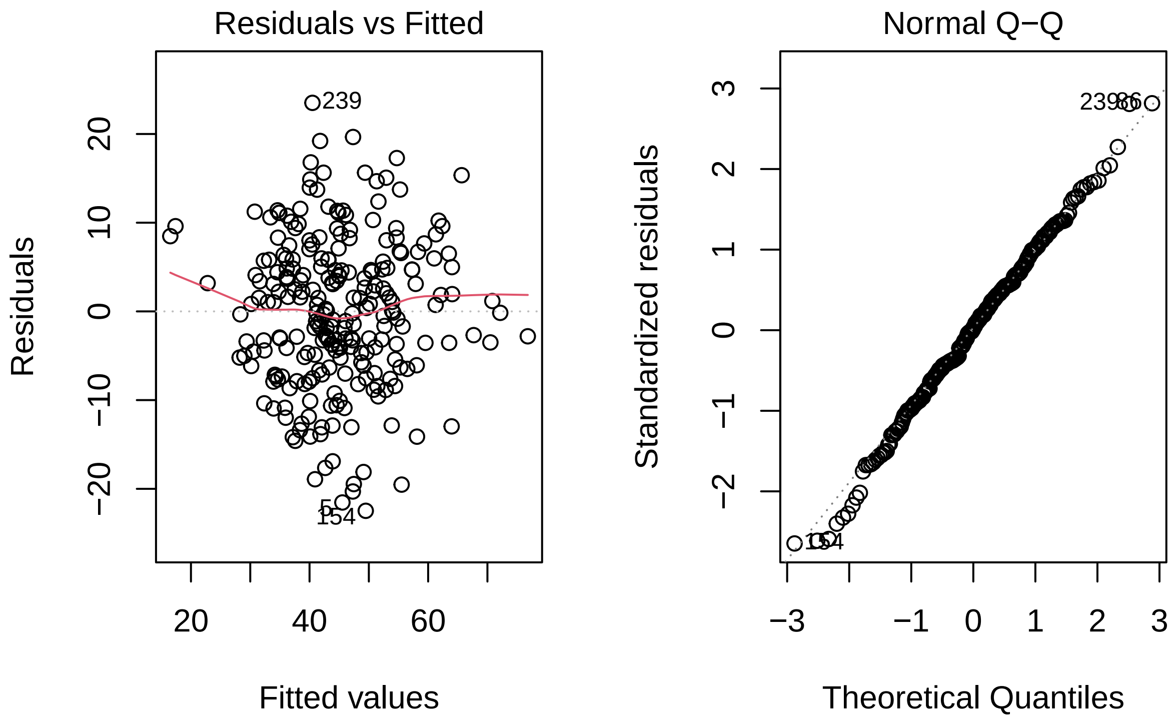
```
#fitting the model after stepwise model selection:
set.seed(123)
new_model <- lm(Age ~ BodyFat + Weight + Height + Neck + Abdomen + Thigh + Knee + Ankle +
                  Forearm + Wrist, data = body_fat)
summary(new_model)
```

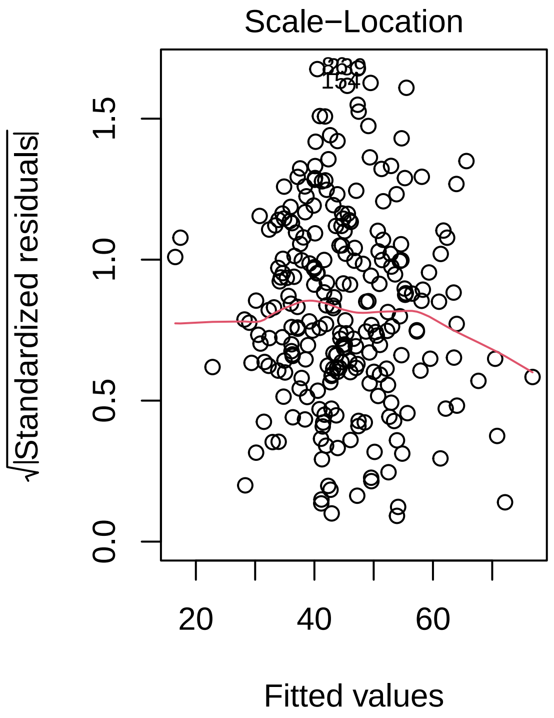
```
##
## Call:
## lm(formula = Age ~ BodyFat + Weight + Height + Neck + Abdomen +
##     Thigh + Knee + Ankle + Forearm + Wrist, data = body_fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.4793  -5.2087   0.1831   5.7306  23.5096
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -27.98721   25.32593  -1.105 0.270226
## BodyFat       0.27240    0.12676   2.149 0.032642 *
## Weight       -0.33216    0.07969  -4.168 4.29e-05 ***
## Height       -0.35967    0.18018  -1.996 0.047039 *
## Neck          0.94272    0.45030   2.094 0.037345 *
## Abdomen       0.93441    0.17905   5.219 3.89e-07 ***
## Thigh        -1.76456    0.23121  -7.632 5.38e-13 ***
## Knee          1.76132    0.46276   3.806 0.000179 ***
## Ankle        -0.75604    0.43736  -1.729 0.085156 .
## Forearm      -0.89971    0.37169  -2.421 0.016237 *
## Wrist         6.04102    1.01009   5.981 7.97e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.561 on 241 degrees of freedom
## Multiple R-squared:  0.5569, Adjusted R-squared:  0.5385
## F-statistic: 30.29 on 10 and 241 DF,  p-value: < 2.2e-16
```

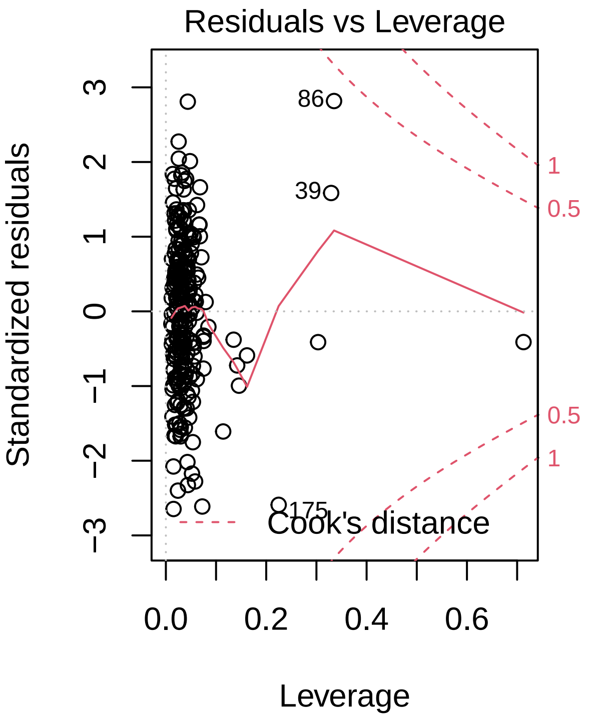
```
confint(new_model)
```

```
##                    2.5 %      97.5 %
## (Intercept) -77.87564336 21.901228198
## BodyFat       0.02269120  0.522106116
## Weight       -0.48914136 -0.175175713
## Height       -0.71458668 -0.004744036
## Neck          0.05569935  1.829740755
## Abdomen       0.58170485  1.287121152
## Thigh        -2.22001478 -1.309112889
## Knee          0.84975383  2.672888025
## Ankle        -1.61757906  0.105504026
## Forearm      -1.63189458 -0.167523516
## Wrist         4.05128710  8.030745229
```

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



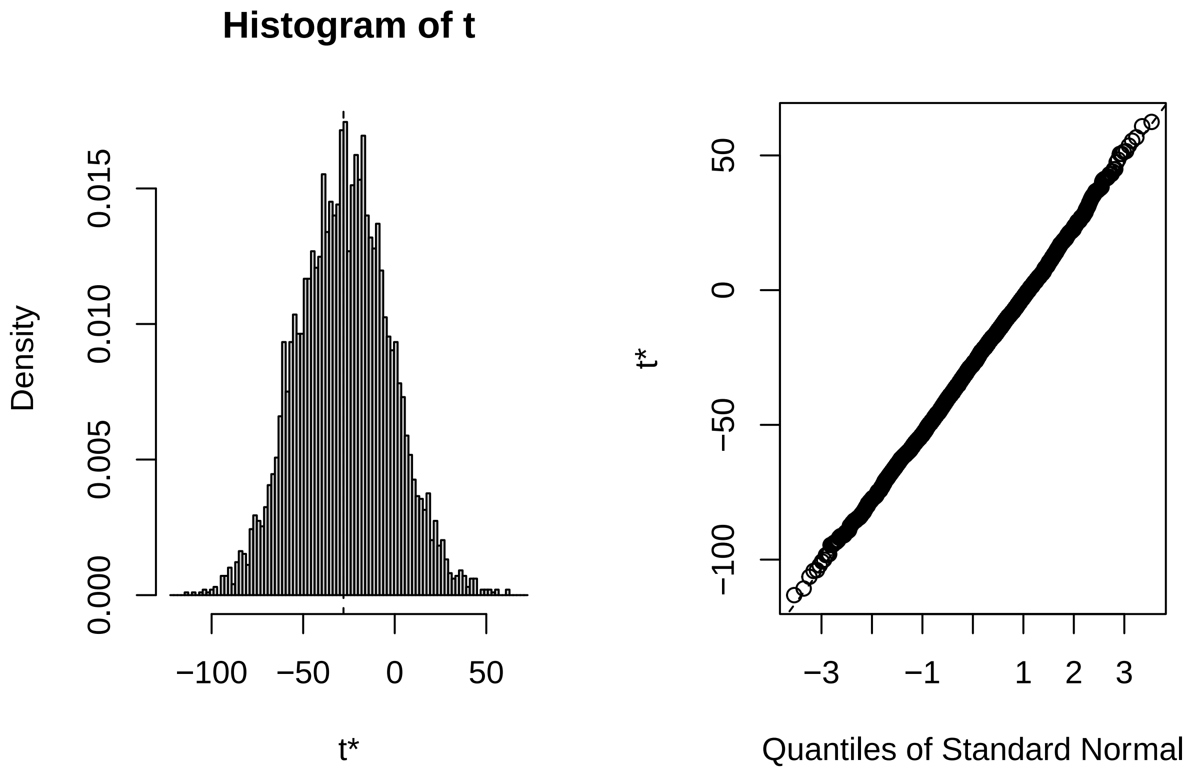

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
86
39
154
20
40
60
Fitted values


Residuals vs Leverage
Standardized residuals
−3
−2
−1
0
1
2
3
86
39
175
Cook's distance
1
0.5
0.5
1
0.0
0.2
0.4
0.6
Leverage

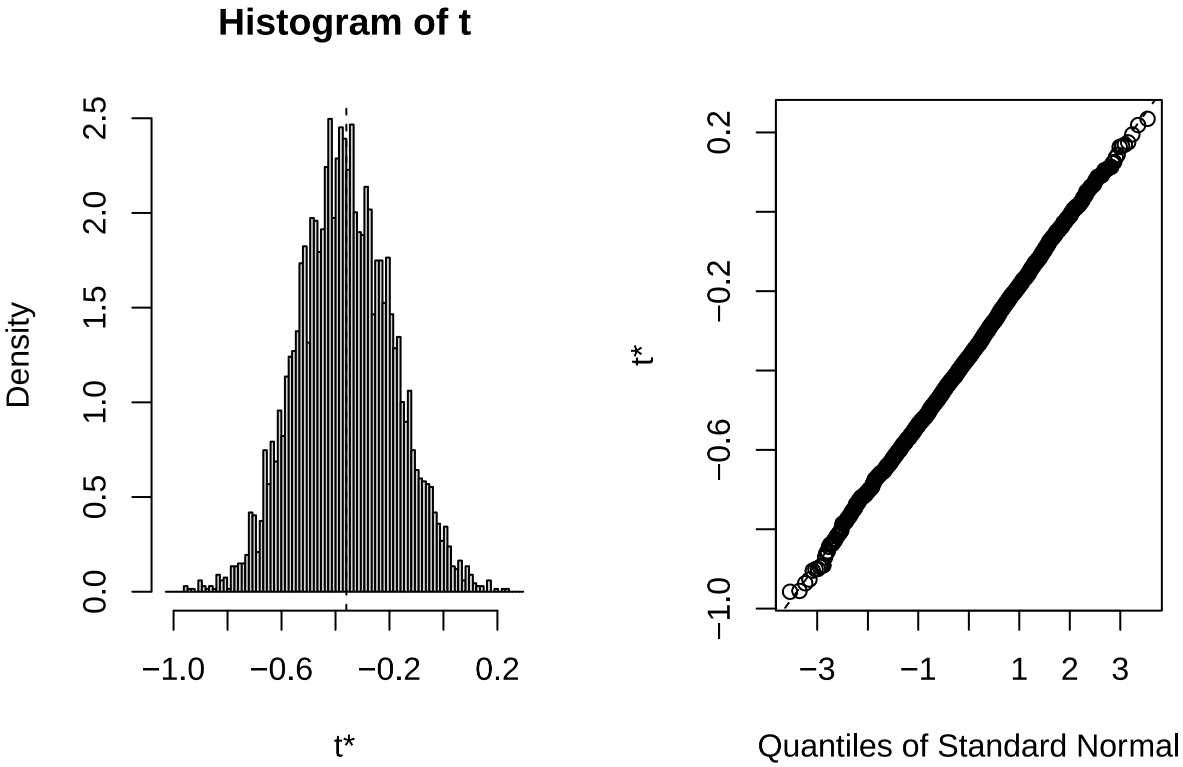
```
set.seed(123)
#residual bootstrap resampling(fixed X):
fit_body <- fitted(new_model)
e_body <- residuals(new_model)
X_body <- model.matrix(new_model)

boot.fixed_body <- function(data, i){
  y_body <- fit_body + e_body[i]
  mod_body <- lm(y_body ~ X_body - 1)
  coefficients(mod_body)
}
body_fixed_boot <- boot(body_fat, boot.fixed_body, 5000)
body_fixed_boot
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = body_fat, statistic = boot.fixed_body, R = 5000)
##
##
## Bootstrap Statistics :
##         original        bias    std. error
## t1*  -27.9872076  0.4119094186 25.20934737
## t2*    0.2723987 -0.0027860505  0.12379289
## t3*   -0.3321585  0.0009656434  0.07853564
## t4*   -0.3596654 -0.0028955306  0.17473208
## t5*    0.9427201  0.0037868682  0.43116909
## t6*    0.9344130  0.0029870094  0.17773945
## t7*   -1.7645638 -0.0026508901  0.22359025
## t8*    1.7613209 -0.0064416741  0.45841674
## t9*   -0.7560375  0.0036828213  0.42332464
## t10*  -0.8997090 -0.0001609307  0.35918134
## t11*   6.0410162 -0.0226273225  0.99493612
```

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



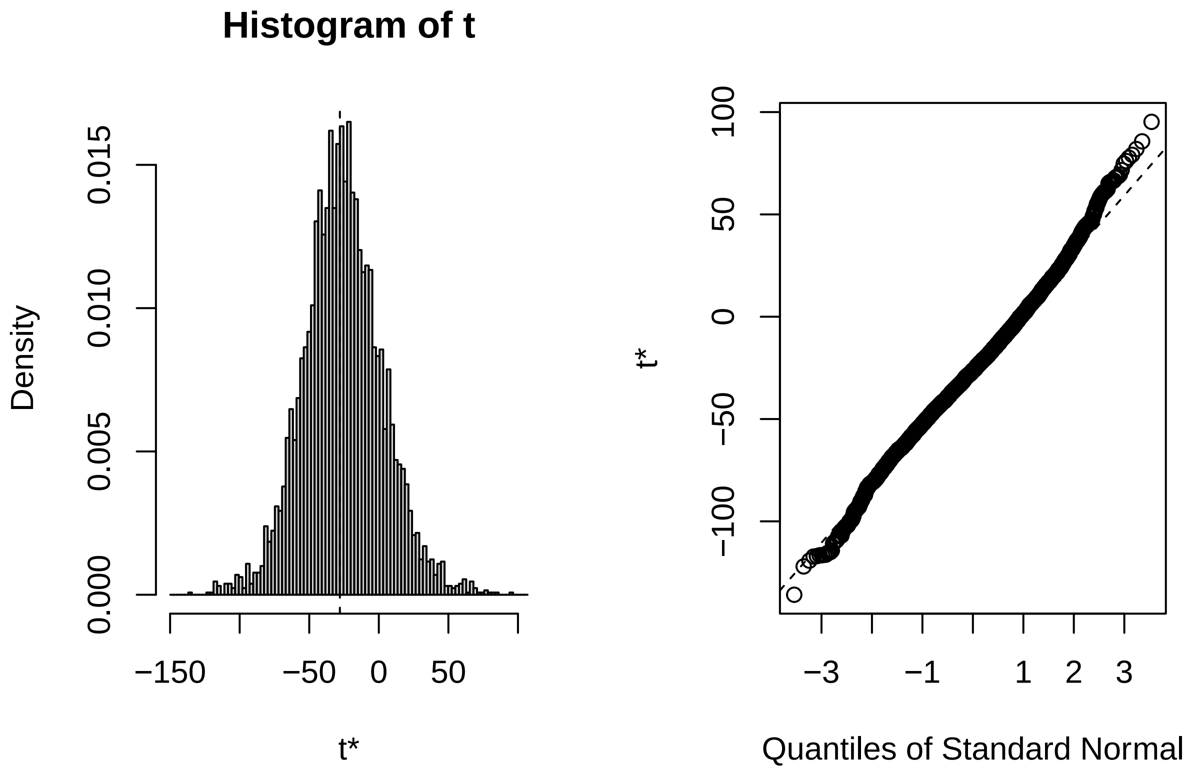
```r
plot(body_fixed_boot, index = 4)
```


Histogram of t
Density
0.0
0.5
1.0
1.5
2.0
2.5
−1.0
−0.6
−0.2
0.2
t*
t*
0.2
−0.2
−0.6
−1.0
−3
−1
1
2
3
Quantiles of Standard Normal

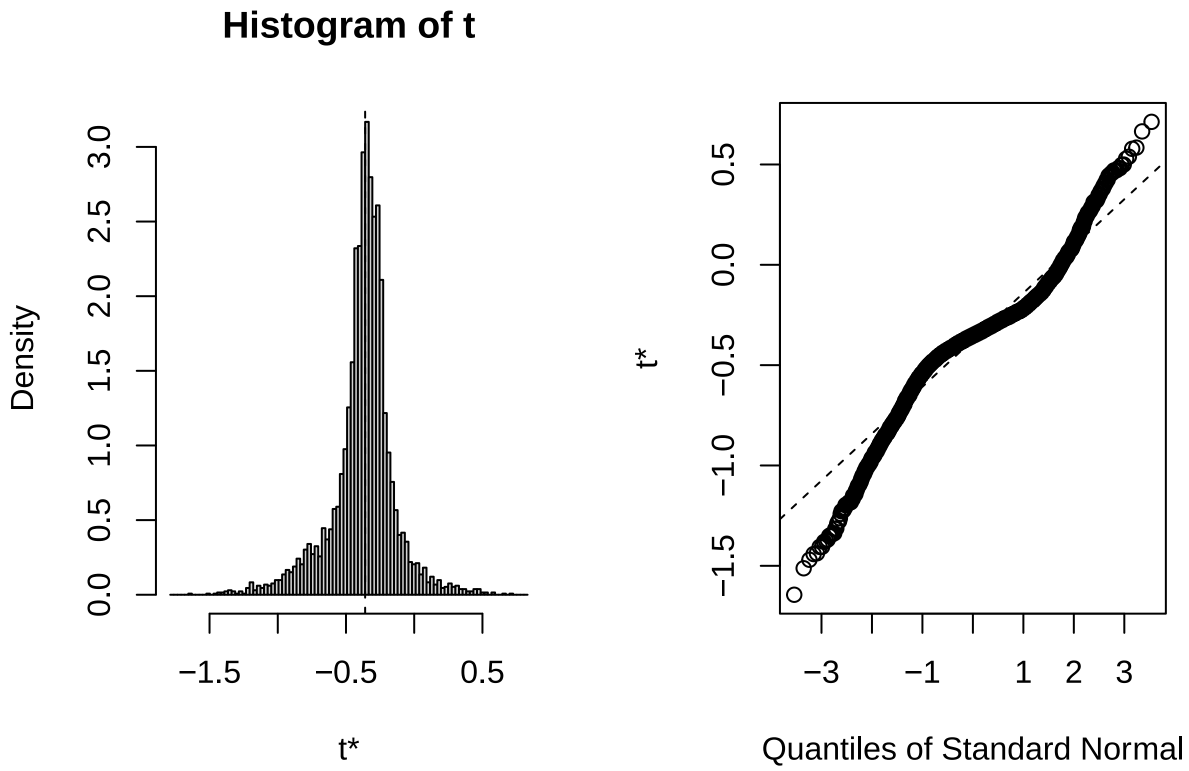
```
set.seed(123)
boot.body <- function(data, i){
  data <- data[i,]
  model_body <- lm(Age ~ BodyFat + Weight + Height + Neck + Abdomen + Thigh +
                     Knee + Ankle + Forearm + Wrist,data = data)
  coefficients(model_body)
}

model_boot_body <- boot(body_fat, boot.body, 5000)
model_boot_body
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = body_fat, statistic = boot.body, R = 5000)
##
##
## Bootstrap Statistics :
##         original        bias    std. error
## t1*  -27.9872076  2.373803686 28.28336557
## t2*    0.2723987  0.001326545  0.12238873
## t3*   -0.3321585  0.004402926  0.08832711
## t4*   -0.3596654 -0.014745298  0.23366021
## t5*    0.9427201 -0.051141811  0.46748994
## t6*    0.9344130 -0.010259643  0.16933894
## t7*   -1.7645638  0.016315230  0.23688076
## t8*    1.7613209  0.029031260  0.49994553
## t9*   -0.7560375 -0.187726802  0.71518077
## t10*  -0.8997090 -0.015257329  0.49710910
## t11*   6.0410162  0.188108013  1.00917611
```

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



```
plot(model_boot_body, index = 4)
```



```
boot.ci(model_boot_body, index = 4, type = c('norm', 'perc', 'bca'))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = model_boot_body, type = c("norm", "perc",
##     "bca"), index = 4)
##
## Intervals :
## Level      Normal              Percentile            BCa
## 95%   (-0.8029,  0.1130 )   (-0.9492,  0.0820 )   (-0.9942,  0.0421 )
## Calculations and Intervals on Original Scale
```

```
boot.ci(body_fixed_boot, index = 4, type = c('norm', 'perc', 'bca'))
```

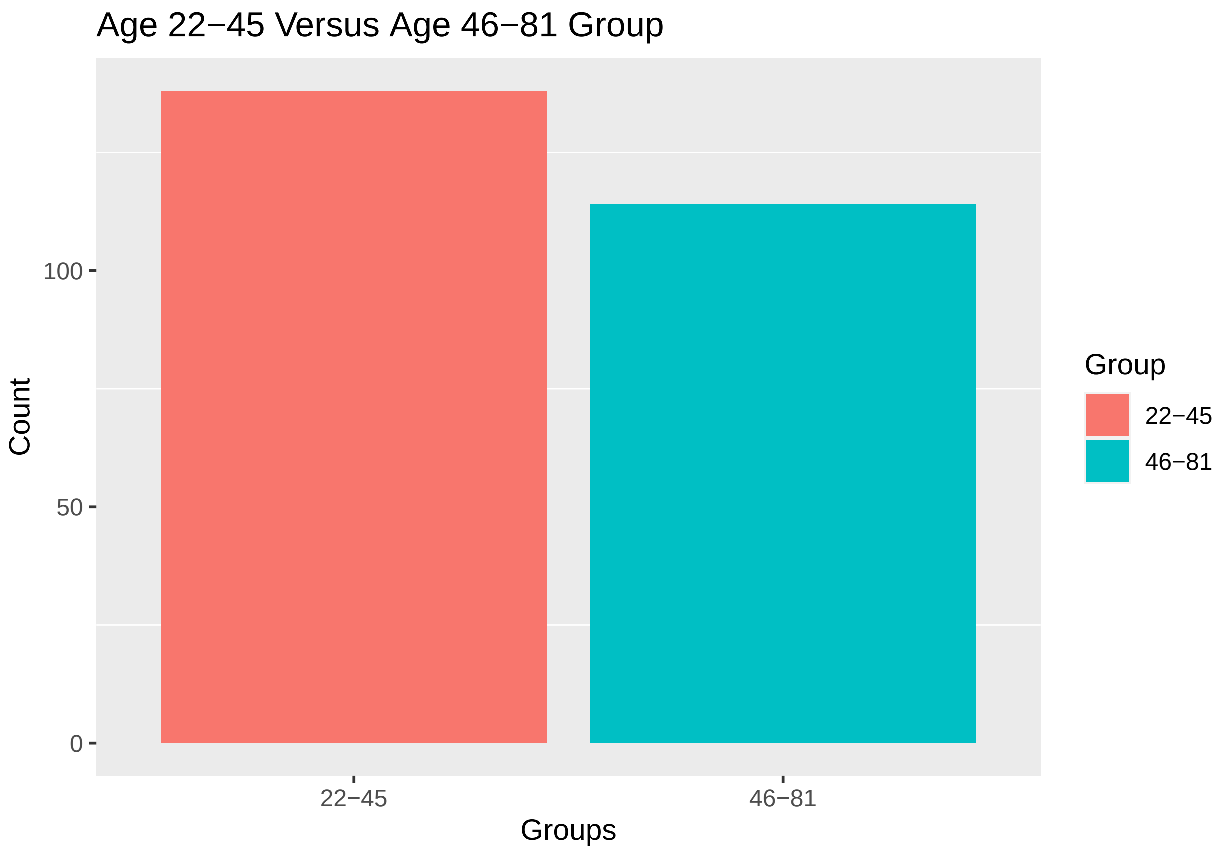
```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = body_fixed_boot, type = c("norm", "perc",
##     "bca"), index = 4)
##
## Intervals :
## Level      Normal              Percentile            BCa
## 95%   (-0.6992, -0.0143 )   (-0.6986, -0.0155 )   (-0.6799,  0.0002 )
## Calculations and Intervals on Original Scale
```

Section 2:

```
new_bodyfat <- body_fat %>% mutate(Group = ifelse(Age<=45, "22-45","46-81"))
head(new_bodyfat, 5)
```

```
##   Density BodyFat Age Weight Height Neck Chest Abdomen   Hip Thigh Knee Ankle
## 1  1.0708    12.3  23 154.25  67.75 36.2  93.1    85.2  94.5  59.0 37.3  21.9
## 2  1.0853     6.1  22 173.25  72.25 38.5  93.6    83.0  98.7  58.7 37.3  23.4
## 3  1.0414    25.3  22 154.00  66.25 34.0  95.8    87.9  99.2  59.6 38.9  24.0
## 4  1.0751    10.4  26 184.75  72.25 37.4 101.8    86.4 101.2  60.1 37.3  22.8
## 5  1.0340    28.7  24 184.25  71.25 34.4  97.3   100.0 101.9  63.2 42.2  24.0
##   Biceps Forearm Wrist Group
## 1   32.0    27.4  17.1 22-45
## 2   30.5    28.9  18.2 22-45
## 3   28.8    25.2  16.6 22-45
## 4   32.4    29.4  18.2 22-45
## 5   32.2    27.7  17.7 22-45
```

```
# Visualize barplot using a ggplot2 package
new_bodyfat %>%
  ggplot(aes(x = Group, fill = Group)) + geom_bar() + labs(title = "Age 22-45 Versus Age 46-81 Group",
```



```
new_df <- new_bodyfat %>% group_by(Group)%>%
  summarise(n =n(),
            Bodyfat_mean = mean(BodyFat))
new_df
```

```
## # A tibble: 2 x 3
##   Group      n Bodyfat_mean
##   <chr> <int>        <dbl>
## 1 22-45   138         18.0
## 2 46-81   114         20.6
```

```
set.seed(123)
# Calculate an observed test statistic.
obs_test_stat <- new_bodyfat %>%
  specify(BodyFat ~ Group)%>%
  calculate(stat = "diff in means", order = c("46-81","22-45"))
round(obs_test_stat,2)
```

```
## Response: BodyFat (numeric)
## Explanatory: Group (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  2.56
```

```
# To double check manually computed observed test statistics.
obs_diff_mean = 20.55088 -  17.99420
obs_diff_mean
```

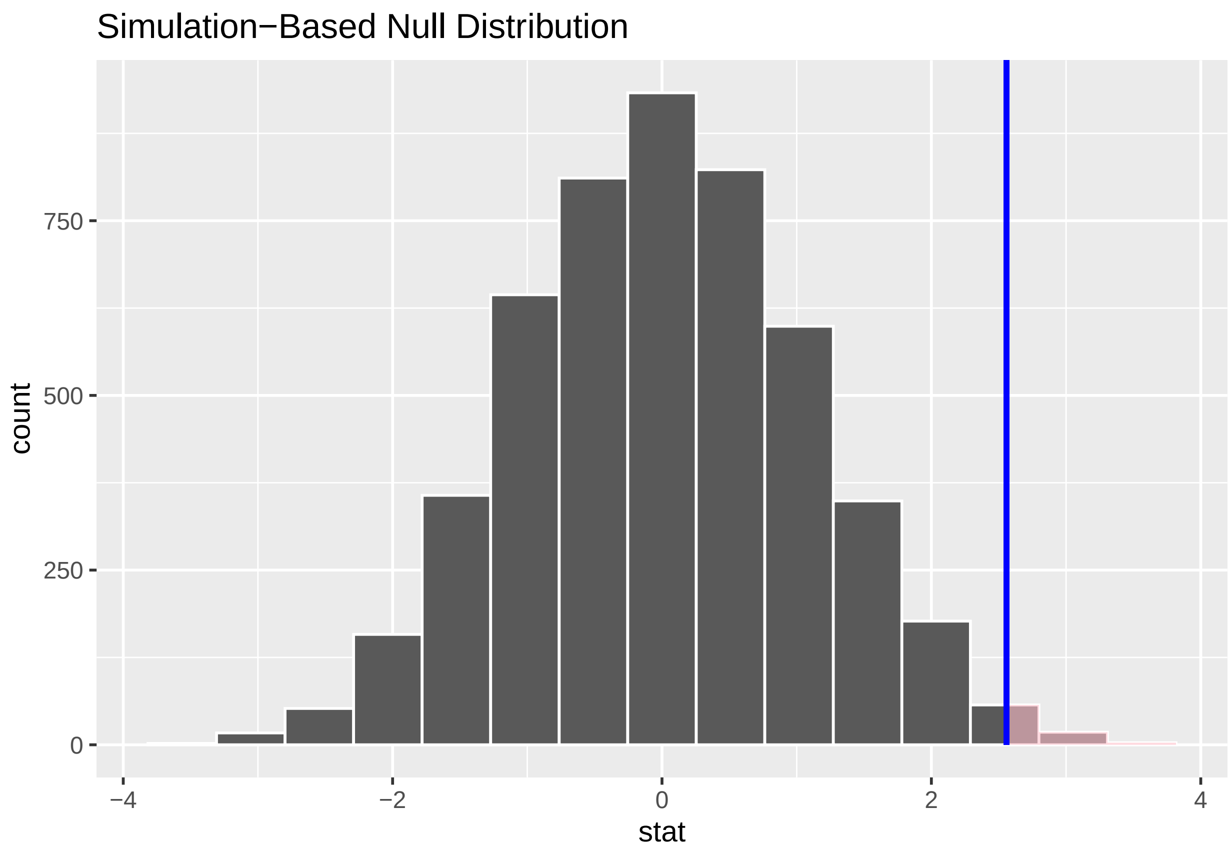
```
## [1] 2.55668
```

```
set.seed(123)
# Create the null distribution.
null_dist <- new_bodyfat %>%
  specify(BodyFat ~ Group)%>%
  hypothesize(null = "independence") %>%
  generate(reps = 5000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("46-81","22-45"))
head(null_dist,5)
```

```
## Response: BodyFat (numeric)
## Explanatory: Group (factor)
## Null Hypothesis: independence
## # A tibble: 5 x 2
##   replicate    stat
##       <int>   <dbl>
## 1         1  0.0834
## 2         2 -0.488
## 3         3  0.162
## 4         4  0.748
## 5         5  1.47
```

```
# Visualize the null distribution.
null_dist %>%
  visualize() +
  shade_p_value(obs_stat = obs_test_stat, direction = "right", col = "blue", lty = 1, lwd = 1)
```

```
## Warning: Duplicated aesthetics after name standardisation: size
```



```
set.seed(123)
# Compute the p-value.
null_dist %>%
  get_p_value(obs_stat = obs_test_stat, direction = "right")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.007
```