Lab2_2229027

JiminLee_2229027

2023-09-15

```
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.2 ──
## ✔ ggplot2 3.3.6     ✔ purrr   1.0.2
## ✔ tibble  3.2.1     ✔ dplyr   1.1.3
## ✔ tidyr   1.2.1     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

The dataset spaghetti.txt contains the weight (in oz) of 20 spaghetti boxes of a famous pasta brand.

- Load the dataset spaghetti.txt in R, using the function read.table.

```
setwd('/Users/jimin/Desktop/데스크탑 - 이지민의 MacBook Air/ᄌ ᅵᄆ ᅵᆫ/ewha/202
3-2/Regression/week2')
data = read.table('spaghetti.txt')
head(data)

##   spaghetti
## 1  15.31335
## 2  15.28379
## 3  15.90502
## 4  16.75127
## 5  15.89350
## 6  14.04193
```

A consumers' association would like to sue the company, affirming that the mean box weight is lower than the nominal one (16 oz). To be sure about their statement, they ask you to perform a suitable test with level 1%.

First compute the sample mean and the standard deviation of the box weight.

```
mean.sp = mean(data$spaghetti)
sd.sp = sd(data$spaghetti)

cat('mean :',mean.sp,'\nsd :', sd.sp)
```

```
## mean : 15.48985
## sd : 0.8821747
```

Perform the test for the consumers’ association.

What is your null hypothesis H0?

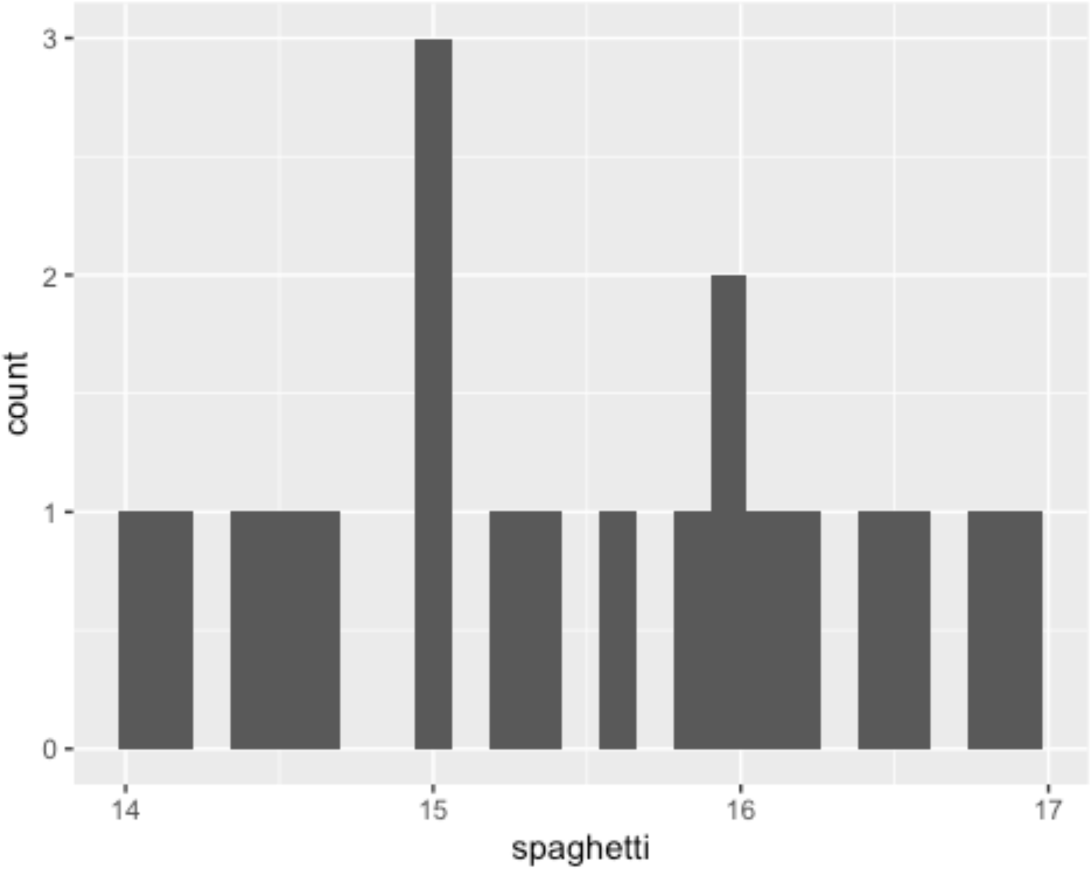
H0 : M = 16

What is your alternative hypothesis H1?

H1 : M < 16

What distribution can you assume for the test statistic (type of distribution and parameters), and why?

```
ggplot(data, aes(x=spaghetti)) + geom_histogram(binwidth = 0.12)
```



14 부터 17 까지 넓게 펼쳐져 있다. 특히 15 쪽에 집중된 모습을 보인다.

Compute the test statistic.

```
nrow(data) # num of obesrvations

## [1] 20
```

```
calTestStatistic <- function(s, n, x.bar, M) {
  se <- s / sqrt(n)
  t_value <- (x.bar - M) / se
  return(t_value)
}

t.stat = calTestStatistic(x.bar=mean.sp, M=16, s=sd.sp, n=nrow(data))
t.stat

## [1] -2.586199
```

Compute the test p-value.

```
tt = t.test(x=data$spaghetti, alternative='less', mu=16, conf.level = 0.99)
tt$p.value

## [1] 0.009055175

tt

##
##  One Sample t-test
##
## data:  data$spaghetti
## t = -2.5862, df = 19, p-value = 0.009055
## alternative hypothesis: true mean is less than 16
## 99 percent confidence interval:
##      -Inf 15.99078
## sample estimates:
## mean of x
##  15.48985
```

Do you reject the null hypothesis, at significance level 1%?

Yes, since t value is less than p value.

Compute the 99% two-sided confidence interval for the mean of the box weight.

```
t.test(x=data$spaghetti, alternative='two.sided', mu=16, conf.level = 0.99)$c
onf.int

## [1] 14.92550 16.05419
## attr(,"conf.level")
## [1] 0.99
```

Consider again the dataset record.txt that you used for Lab 1. This dataset contains running records obtained from athletes from different countries in various types of athletics events (sprints and middle-distance).

We have data about 55 countries (observations) and 6 records (variables): 100 meters, 200 meters, 400 meters, 800 meters, 1500 meters and 3000 meters.

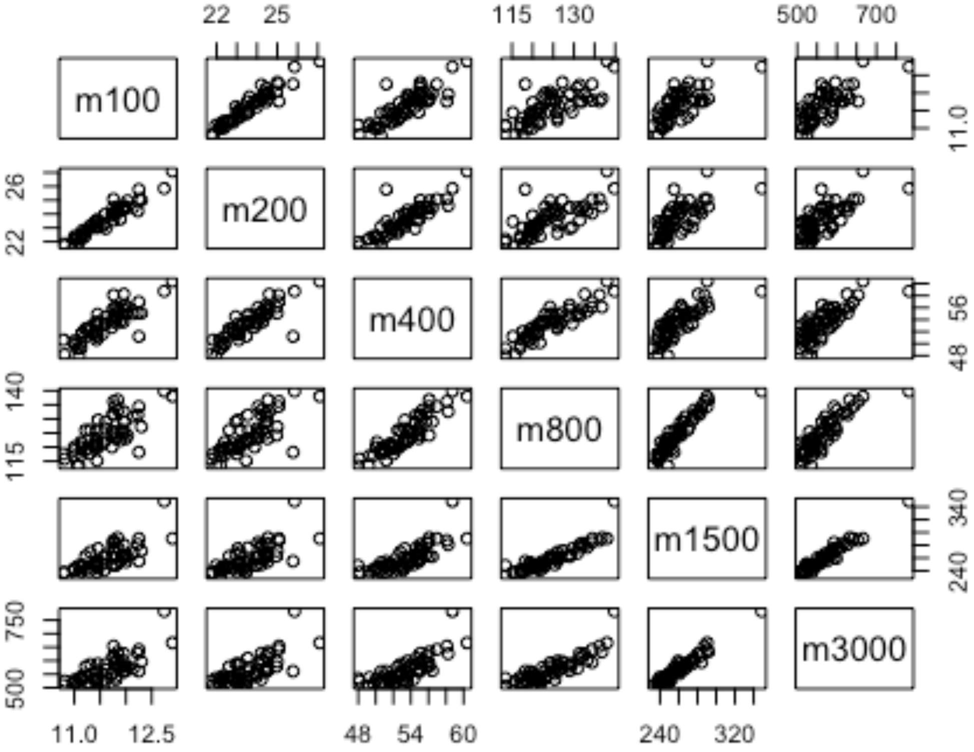
Load the dataset record.txt in R, using the function read.table (remember to set sep=’ ’)

```
data.2 = read.table('record.txt')
head(data.2)
```

```
##           m100  m200  m400  m800 m1500 m3000
## argentin 11.61 22.94 54.50 129.0 265.8 587.4
## australi 11.20 22.35 51.08 118.8 247.8 544.8
## austria  11.43 23.09 50.62 119.4 253.2 560.4
## belgium  11.41 23.04 52.00 120.0 248.4 532.8
## bermuda  11.46 23.05 53.30 129.6 274.8 588.6
## brazil   11.31 23.17 52.80 126.0 269.4 586.2
```

Draw a scatterplot and compute the correlation for all pairs of variables in the dataset. Interpret the results you obtained: what can you observe about the relationship among the variables?

```
pairs(data.2) # Positively increasing
```



Consider the variables m100 and m400.

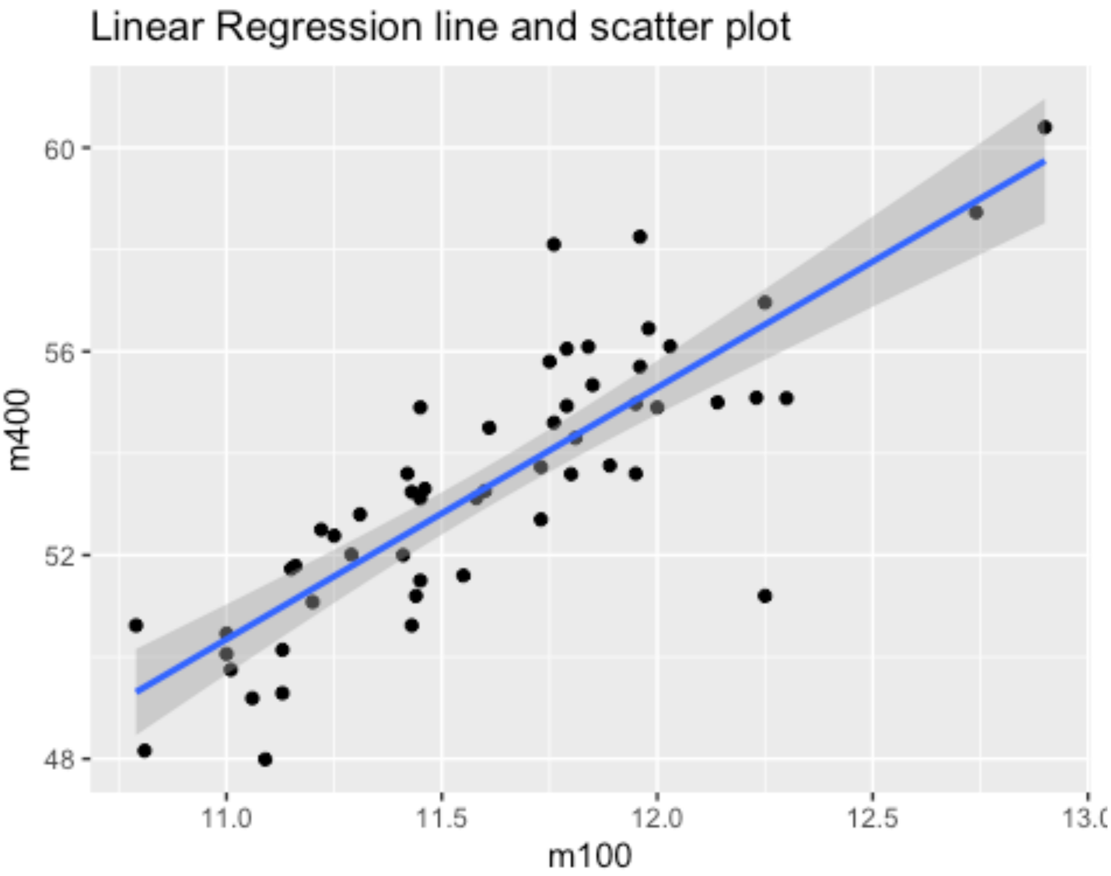
Using the equations(not lm function or matrix equation in r), compute the least square estimators for the coefficients of a single linear regression model, with response m100 and predictor m400. How do you interpret the slope or the regression line?

```
x = data.2$m100 # response
y = data.2$m400 # predictor
```

```
b1 = sum((y-mean(y))*(x-mean(x)))/sum((x-mean(x))^2)
b0 = mean(y) - mean(x)*b1

cat('b0 : ',b0,'\nb1 : ',b1)

## b0 :  -4.032628
## b1 :  4.943686
```

Produce a scatter plot for m1500 vs m800 with the fitted regression line superimposed.

```
plt = ggplot(data=data.2, aes(x=m100, y=m400)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x, geom = "smooth") +
  ggtitle('Linear Regression line and scatter plot')

plt
```



Re-compute the least square estimators of beta0 and beta1 using the matrix equation.

```
X = cbind(1, x)
b.vec = solve(t(X)%*%X)%*%t(X)%*%y
b.vec
```

```
##         [,1]
##    -4.032628
## x  4.943686
```

Re-compute the least square estimators of beta0 and beta1 using the R function lm, and visualize the summary of the regression.

```
lm.fit = lm(y~x) # linear regression
summary(lm.fit)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3275 -0.9858 -0.0523  0.7781  3.9949
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.033      5.209  -0.774    0.442
## x              4.944      0.448  11.034 2.41e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.489 on 53 degrees of freedom
## Multiple R-squared:  0.6967, Adjusted R-squared:  0.691
## F-statistic: 121.8 on 1 and 53 DF,  p-value: 2.41e-15
```

Which are n and p for the considered regression model?

n : total num of observations p : total num of parameters

Compute the fitted values and the residuals, using the estimated regression line.

```
print('fitted.values')

## [1] "fitted.values"

fitted(lm.fit)

##        1        2        3        4        5        6        7        8
## 53.36357 51.33666 52.47371 52.37483 52.62202 51.88047 55.98373 50.34792
##        9       10       11       12       13       14       15       16
## 55.29161 55.04443 53.31414 59.74093 55.09386 50.79286 52.42427 54.25344
##       17       18       19       20       21       22       23       24
## 50.99060 51.08948 49.40862 50.39736 50.34792 54.25344 54.50062 52.57258
##       25       26       27       28       29       30       31       32
## 55.04443 54.55006 52.47371 52.57258 51.78159 53.95681 53.95681 55.09386
##       33       34       35       36       37       38       39       40
## 56.52753 55.43992 56.42866 54.10513 54.74780 51.58385 53.06695 53.21526
##       41       42       43       44       45       46       47       48
## 56.52753 54.10513 50.99060 54.35231 52.52315 56.77472 54.30287 51.13891
##       49       50       51       52       53       54       55
## 52.57258 51.43553 54.05569 55.19274 49.30975 50.64454 58.94994
```

```r
print('residuals')
```

```
## [1] "residuals"
```

```r
residuals(lm.fit)
```

```
##           1           2           3           4           5           6
##  1.13642787 -0.25666069 -1.85370857 -0.37483484  0.67798083  0.91953380
##           7           8           9          10          11          12
## -0.98372595 -0.28792340 -0.39160985 -0.07442553 -0.05413527  0.65907235
##          13          14          15          16          17          18
##  3.15613761 -2.80285518  1.17572829  1.79656431 -0.85060264  0.64052363
##          19          20          21          22          23          24
## -1.24862297 -0.64736026  0.11207660  0.67656431  1.58937998 -1.07258230
##          25          26          27          28          29          30
## -1.44442553  0.78994312  0.76629143  2.32741770  0.22840753 -0.22681451
##          31          32          33          34          35          36
## -1.25681451  0.60613761 -5.32753146  0.66007956 -1.33865773  3.99487490
##          37          38          39          40          41          42
## -0.98780434  0.79615499 -1.46695095 -0.09526154  0.43246854  0.49487490
##          43          44          45          46          47          48
## -1.70060264 -0.05230942 -1.32314544 -1.69471578 -0.71287256  0.65108677
##          49          50          51          52          53          54
##  0.53741770  1.06446558  1.74431176  1.25726388  1.31025075 -1.45454459
##          55
## -0.21993782
```

Consider the 25th and 75th percentiles of the variable m100 on the dataset.Use the estimated regression line to estimate the fitted value of the variable m1500 at each of these two percentiles.

```r
m100.25th = quantile(data.2$m100, 0.25)
m100.75th = quantile(data.2$m100, 0.75)

predict(lm.fit, newdata = data.frame(x = m100.25th))
```

```
##      25%
## 51.68272
```

```r
predict(lm.fit, newdata = data.frame(x = m100.75th))
```

```
##      75%
## 54.89611
```

```r
m100.25th
```

```
##   25%
## 11.27
```