FLS301_2021

Inferential Stats with R

Empowerment for Local People Foundation

Lagos: Dec 8-9, 2021

Congratulation for making the leap. If you enjoy solving problems, you will enjoy `R`. But first let us enjoy a `garri solution` using 1 bowl and then 2 bowls of `water` by adding `garri` and `sugar`.

R simply works as a programming language that let us create objects and use reuse them as we want in subsequent iterations.

```
sugar <- 1+2

garri <- 8-2

water  <- 1 #1 bowl

garri_solution_one <-sugar + garri + water
garri_solution_two <-sugar + garri + (2*water)
garri_solution_one
```

```
## [1] 10
```

```
garri_solution_two
```

```
## [1] 11
```

In R, we have to work in a specific folder our system called working directory. That is where everything happens!

Lets take a quick look at our current working directory.

```
getwd()
```

```
## [1] "/cloud/project"
```

You can use the function `setwd` to change/set a new working directory

```
setwd("/cloud/project")
```

We can set path directly. The easiest way to do this is to a set default working directory: Session > Set Working Directory.

R Data Structures

Vector: A vector is simply a list of items that are of the same type. They are six types of atomic vectors-logical, integer, character, raw, double, and complex.

Matrices: A matrix is a two dimensional data set with columns and rows.

List: A list in R can contain many different data types inside it. A list is a collection of data which is ordered and changeable.

Data Frames:Data Frames are data displayed in a format as a table.

Factors: Factors are used to categorize data.

OTHER R Synthax and Keywords

Objects: vector, list, matrix, array, factor, and data frame.

Functions.

Rows, Columns.

Method.

Loops.

Packages.

Working Directory.

R Operators

Arithmetic operators (+, -, /, ^, x %% y) Assignment operators (<-) Comparison operators (==. !=, >=) Logical operators (&, |, !) Miscellaneous operators (%in%)

Create dataset for our analysis

Here we want to create a dataset of 6 variables consisting data about 20 staff in organization. The variables are `Gender`, `Weight`, `income`, `rating`, `marital status` and whether staff stays in the `city central`.

```
Gender <- c("Male", "Male", "Male", "Male", "Male",
            "Male", "Male", "Male", "Male", "Male",
            "FeMale","FeMale", "FeMale","FeMale", "FeMale",
           "FeMale","FeMale","FeMale","FeMale","FeMale")
weight <- c(89, 75, 88, 75, 49, 89, 110, 120, 89, 75,
            75, 76, 87, 110, 67, 76, 43, 55, 59, 60)
income <- c(50000, 95000, 120000, 800000, 650000, 92000, 94000, 222000, 543000,75000,
            63000, 40000, 99000, 450000, 180000, 190000, 96000, 780000, 150000, 342000)
rating <- c(5, 1, 2, 4, 9, 9, 8, 1, 9, 7,
            5, 6, 6, 1, 1, 1, 3, 6, 9, 4)
Marstatus <- c("Married", "Married","Single", "Single","Single",
             "Single", "Divorced","Single", "Married","Single",
             "Married", "Single","Single", "Divorced","Single",
             "Single", "Divorced", "Single","Married", "Divorced")
CityCentral <- c("Yes","No","Yes","Yes","Yes","Yes","Yes","Yes","No","Yes",
                 "No","No","Yes","Yes","No","Yes","No","Yes","No","No")
```

Binding two columns - cbind

Concatenate which is 'c' allows us to group different things into one object.

Next, we are taking 2 objects into a column and we telling R to use `cbind` to take these different columns and merge them as one data frame is an object in R.

```r
officew <- as.data.frame(cbind(Gender, weight, income, rating, Marstatus,
                               CityCentral))
officew
```

```
##    Gender weight income rating Marstatus CityCentral
## 1    Male     89  50000      5   Married         Yes
## 2    Male     75  95000      1   Married          No
## 3    Male     88 120000      2    Single         Yes
## 4    Male     75  8e+05      4    Single         Yes
## 5    Male     49 650000      9    Single         Yes
## 6    Male     89  92000      9    Single         Yes
## 7    Male    110  94000      8  Divorced         Yes
## 8    Male    120 222000      1    Single         Yes
## 9    Male     89 543000      9   Married          No
## 10   Male     75  75000      7    Single         Yes
## 11 FeMale     75  63000      5   Married          No
## 12 FeMale     76  40000      6    Single          No
## 13 FeMale     87  99000      6    Single         Yes
## 14 FeMale    110 450000      1  Divorced         Yes
## 15 FeMale     67 180000      1    Single          No
## 16 FeMale     76 190000      1    Single         Yes
## 17 FeMale     43  96000      3  Divorced          No
## 18 FeMale     55 780000      6    Single         Yes
## 19 FeMale     59 150000      9   Married          No
## 20 FeMale     60 342000      4  Divorced          No
```

Factor is another way of calling categorical variable in R. The `as.data.frame` changes the factor (categorical) into a data frame without necessarily changing the class. The `c` only works if the number of rows in each variable is the same.

```r
meanincome <- mean(officew$income)
```

```
## Warning in mean.default(officew$income): argument is not numeric or logical:
## returning NA
```

```r
modeincome <- mode(officew$income)
modeincome
```

```
## [1] "character"
```

```r
officew$income <- as.numeric(as.character(officew$income))
officew$Marstatus<- as.factor(as.character(officew$Marstatus))
sdincome <- sd(officew$income)
varincome <- var(officew$income)
varincome
```

```
## [1] 62240786842
```

```r
class(officew$income)
```

```
## [1] "numeric"
```

```r
head(officew, n = 5)
```

```
##   Gender weight income rating Marstatus CityCentral
## 1   Male     89  50000      5   Married         Yes
## 2   Male     75  95000      1   Married          No
## 3   Male     88 120000      2    Single         Yes
## 4   Male     75 800000      4    Single         Yes
```

```
## 5   Male     49 650000      9    Single         Yes
```

summary stats

Let us use rbind (rowbind) to bind the rows of two different dataset together. The row names of the two datasets must be same for it to work

Create dataset for men with 3 variables

```
Gender1 <- c("Male", "Male", "Male", "Male", "Male",
            "Male", "Male", "Male", "Male", "Male", "Male")
weight1 <- c(89, 75, 88, 75, 49, 89, 110, 120, 89, NA, 75)

rating1 <- c(5, 1, 2, 4, 9, 9, 8, 1, 9,NA, 7)

officew_men22<- as.data.frame(cbind(Gender1, weight1, rating1))
officew_men22
```

```
##    Gender1 weight1 rating1
## 1     Male      89       5
## 2     Male      75       1
## 3     Male      88       2
## 4     Male      75       4
## 5     Male      49       9
## 6     Male      89       9
## 7     Male     110       8
## 8     Male     120       1
## 9     Male      89       9
## 10    Male    <NA>    <NA>
## 11    Male      75       7
```

Create dataset for women with 3 variables

```
Gender1 <- c("FeMale","FeMale", "FeMale","FeMale", "FeMale",
            "FeMale","FeMale","FeMale","FeMale", "FeMale", "FeMale")
weight1 <- c(75, 76, 87, 110, 67, 76, 43, NA, 55, 59, 60)

rating1 <- c( 4, 6, 4, 1, 1, 4, 3, 6,NA, 9, 4)


officew_women22<- as.data.frame(cbind(Gender1, weight1, rating1))
officew_women22
```

```
##    Gender1 weight1 rating1
## 1   FeMale      75       4
## 2   FeMale      76       6
## 3   FeMale      87       4
## 4   FeMale     110       1
## 5   FeMale      67       1
## 6   FeMale      76       4
## 7   FeMale      43       3
## 8   FeMale    <NA>       6
## 9   FeMale      55    <NA>
## 10  FeMale      59       9
## 11  FeMale      60       4
```

Row Bind

We are binding both female and male dataset with `rbind`. Since they have the same number of rows, we can bind:

```
officew_full <- rbind(officew_men22, officew_women22)
officew_full
```

```
##    Gender1 weight1 rating1
## 1     Male      89       5
## 2     Male      75       1
## 3     Male      88       2
## 4     Male      75       4
## 5     Male      49       9
## 6     Male      89       9
## 7     Male     110       8
## 8     Male     120       1
## 9     Male      89       9
## 10    Male    <NA>    <NA>
## 11    Male      75       7
## 12  FeMale      75       4
## 13  FeMale      76       6
## 14  FeMale      87       4
## 15  FeMale     110       1
## 16  FeMale      67       1
## 17  FeMale      76       4
## 18  FeMale      43       3
## 19  FeMale    <NA>       6
## 20  FeMale      55    <NA>
## 21  FeMale      59       9
## 22  FeMale      60       4
```

REMOVING NAs

```
officew_women22_nona <- officew_women22[!is.na(officew_women22$rating1)
                     &!is.na(officew_women22$weight1), ]

officew_women22_nona
```

```
##    Gender1 weight1 rating1
## 1   FeMale      75       4
## 2   FeMale      76       6
## 3   FeMale      87       4
## 4   FeMale     110       1
## 5   FeMale      67       1
## 6   FeMale      76       4
## 7   FeMale      43       3
## 10  FeMale      59       9
## 11  FeMale      60       4
```

The new dataset `officew_women22_nona` has no missing values(NAs)

#Exporting files

```
library(openxlsx)# export to excel
library(haven)

write.csv(officew, "officew.csv") #export to csv
write_sav(officew, "officew.sav")#export to spss
```

#Importing files

```
library(readr)
officew_wd <- read_csv("officew.csv")
```

#Importing files from Github

```
#install.packages("readr")
#library(readxl)

library(openxlsx)# export to excel
library(RCurl)
x <- getURL("https://raw.githubusercontent.com/abiola1864/FLS301/main/officew.csv")
officew<- read.csv(text = x)
head(officew)
```

```
##   X Gender weight income rating Marstatus CityCentral
## 1 1   Male     89  50000      5   Married         Yes
## 2 2   Male     75  95000      1   Married          No
## 3 3   Male     88 120000      2    Single         Yes
## 4 4   Male     75 800000      4    Single         Yes
## 5 5   Male     49 650000      9    Single         Yes
## 6 6   Male     89  92000      9    Single         Yes
```

Subsetting and Filtering

You will find two ways you can subset a data using base R. Additionally, with the subset and select functions , subset both Gender variable and select as the required row.

```
officew_women1 <- officew[officew$Gender == "FeMale",]
officew_women2 <- subset(officew, Gender == "FeMale")
officew_women3 <- subset(officew, Gender == "FeMale", select =
                           c("weight"))



officew_women2
```

```
##     X Gender weight income rating Marstatus CityCentral
## 11 11 FeMale     75  63000      5   Married          No
## 12 12 FeMale     76  40000      6    Single          No
## 13 13 FeMale     87  99000      6    Single         Yes
## 14 14 FeMale    110 450000      1  Divorced         Yes
## 15 15 FeMale     67 180000      1    Single          No
## 16 16 FeMale     76 190000      1    Single         Yes
## 17 17 FeMale     43  96000      3  Divorced          No
## 18 18 FeMale     55 780000      6    Single         Yes
## 19 19 FeMale     59 150000      9   Married          No
## 20 20 FeMale     60 342000      4  Divorced          No
```

```r
officew_women3
```

```
##    weight
## 11     75
## 12     76
## 13     87
## 14    110
## 15     67
## 16     76
## 17     43
## 18     55
## 19     59
## 20     60
```

```r
# male
officew_men1 <- officew[officew$Gender == "Male",]
officew_men2 <- subset(officew, Gender == "Male")
officew_men3 <- subset(officew, Gender == "Male", select =
                          c("weight"))
```

We will be using several tools from the tidyr and dplyr packages to achieve data wrangling. Remember we already know some functions from this packages: drop_na, etc…

```r
summary(officew_men3$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    49.0    75.0    88.5    85.9    89.0   120.0
```

```r
summary(officew_women3$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   43.00   59.25   71.00   70.80   76.00  110.00
```

standard deviation:

```r
sd(officew$weight, na.rm = T)
```

```
## [1] 20.25957
```

For the sample size, we need to omit all missing values. the length(), which() and is.na() functions can help us:

```r
length(which(!is.na(officew$weight)))
```

```
## [1] 20
```

Let us select only the 3rd and 5th variable (Income and marital status)

```r
officew_3_5 <- officew[c(3,5)]
officew_3_5
```

```
##    weight rating
## 1      89      5
## 2      75      1
## 3      88      2
## 4      75      4
## 5      49      9
## 6      89      9
## 7     110      8
## 8     120      1
```

```
## 9      89      9
## 10     75      7
## 11     75      5
## 12     76      6
## 13     87      6
## 14    110      1
## 15     67      1
## 16     76      1
## 17     43      3
## 18     55      6
## 19     59      9
## 20     60      4
```

Let us exclude 3rd and 5th variable (Income and marital status)

```
officew_1_2_4 <- officew[c(-3,-5)]
officew_1_2_4
```

```
##     X Gender income Marstatus CityCentral
## 1   1   Male  50000   Married         Yes
## 2   2   Male  95000   Married          No
## 3   3   Male 120000    Single         Yes
## 4   4   Male 800000    Single         Yes
## 5   5   Male 650000    Single         Yes
## 6   6   Male  92000    Single         Yes
## 7   7   Male  94000  Divorced         Yes
## 8   8   Male 222000    Single         Yes
## 9   9   Male 543000   Married          No
## 10 10   Male  75000    Single         Yes
## 11 11 FeMale  63000   Married          No
## 12 12 FeMale  40000    Single          No
## 13 13 FeMale  99000    Single         Yes
## 14 14 FeMale 450000  Divorced         Yes
## 15 15 FeMale 180000    Single          No
## 16 16 FeMale 190000    Single         Yes
## 17 17 FeMale  96000  Divorced          No
## 18 18 FeMale 780000    Single         Yes
## 19 19 FeMale 150000   Married          No
## 20 20 FeMale 342000  Divorced          No
```

Here is another way of writing what we wrote above (including and not excluding)

```
office_1_2_4B <- officew[c(1,2,4)]
office_1_2_4B
```

```
##     X Gender income
## 1   1   Male  50000
## 2   2   Male  95000
## 3   3   Male 120000
## 4   4   Male 800000
## 5   5   Male 650000
## 6   6   Male  92000
## 7   7   Male  94000
## 8   8   Male 222000
## 9   9   Male 543000
## 10 10   Male  75000
```

```
## 11 11 FeMale  63000
## 12 12 FeMale  40000
## 13 13 FeMale  99000
## 14 14 FeMale 450000
## 15 15 FeMale 180000
## 16 16 FeMale 190000
## 17 17 FeMale  96000
## 18 18 FeMale 780000
## 19 19 FeMale 150000
## 20 20 FeMale 342000
```

Include 1st and 2nd variable(column), and 4 and the 4th and 5th observation(row)

```
officew_weight_inc_ <- officew[c(1:2),c(4:5)]
officew_weight_inc_
```

```
##   income rating
## 1  50000      5
## 2  95000      1
```

#Conditional Subsetting

In R, `|` means it returns TRUE if one of the statement is TRUE. In R `&` means it returns TRUE if both elements are TRUE

We want to subset a dataframe of female staff whose income or if any staff earn N100,000 and above (AND). Returns the result for any of the conditions met.

```
office_f_hincome <- subset(officew, income >=100000| Gender %in% "FeMale",
                   select=c(1:5))
office_f_hincome
```

```
##     X Gender weight income rating
## 3   3   Male     88 120000      2
## 4   4   Male     75 800000      4
## 5   5   Male     49 650000      9
## 8   8   Male    120 222000      1
## 9   9   Male     89 543000      9
## 11 11 FeMale     75  63000      5
## 12 12 FeMale     76  40000      6
## 13 13 FeMale     87  99000      6
## 14 14 FeMale    110 450000      1
## 15 15 FeMale     67 180000      1
## 16 16 FeMale     76 190000      1
## 17 17 FeMale     43  96000      3
## 18 18 FeMale     55 780000      6
## 19 19 FeMale     59 150000      9
## 20 20 FeMale     60 342000      4
```

We want to subset a dataframe of female staff whose income is N100,000 and above (AND). The two conditions have to be met here.

```
office_f_hincome1<- subset(officew, income >=100000 & Gender %in% "FeMale",
                   select=c(1:5))

office_f_hincome1
```

```
##     X Gender weight income rating
## 14 14 FeMale    110 450000      1
```

```
## 15 15 FeMale     67 180000      1
## 16 16 FeMale     76 190000      1
## 18 18 FeMale     55 780000      6
## 19 19 FeMale     59 150000      9
## 20 20 FeMale     60 342000      4
```

#DPLYR

If you have not instaled the packages, you have to install them first by removing the ash symbols!

```
# install.packages("tidyr")
# install.packages("dplyr")
library(tidyr)
library(dplyr)
```

We are not getting accurate interval level summary because R see it as a factor.

```
class(officew_women3$weight)
```

```
## [1] "integer"
```

```
class(officew_men3$weight)
```

```
## [1] "integer"
```

Let us change the class from character to numeric or interval

```
officew_women3$weight <-
  as.numeric(as.character(officew_women3$weight))
```

```
officew_men3$weight <-
  as.numeric(as.character(officew_men3$weight))
```

Let us check us again, great!

```
class(officew_women3$weight)
```

```
## [1] "numeric"
```

```
class(officew_men3$weight)
```

```
## [1] "numeric"
```

We can now compare the means of gender:

```
summary(officew_women3$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   43.00   59.25   71.00   70.80   76.00  110.00
```

Sampling Inference with t and Z test

#What is the probabilty that a random male staff will weigh above 125.3

```
rand_mstaff <- (125.3118 - mean(officew_men3$weight))/sd(officew_men3$weight)
1-pnorm(rand_mstaff)
```

```
## [1] 0.02275009
```

```
1- pnorm(2)
```

```
## [1] 0.02275013
```

What is the probabilty that a random male staff will be obsess (weigh above 100) It is 2.3 percent

```
rand_menObestaff <- (100 - mean(officew_men3$weight))/sd(officew_men3$weight)
1-pnorm(rand_menObestaff)
```

```
## [1] 0.2371433
```

What is the probabilty that a random female staff will be . overweight (weigh above 100 is overweight) It is 0.05 percent (less than 1%)

```
rand_womenObestaff <- (100 - mean(officew_women3$weight))/
  sd(officew_women3$weight)

1-pnorm(rand_womenObestaff)
```

```
## [1] 0.05968218
```

HYPOTHESIS TESTING

We know that the probability that the company will hire men with with obseity is 2.3%, and for women is about 1%. It is clear that the company hires more men with obesity than woman. However, what we do not know if this difference is due to error in our random sampling or truly reflect the differences in the entire staff. We are going to use HYPOTHESIS TESTING. By assuming first, that the difference between them is zero referred to as NULL

#Method 1 and 2: P-value & T-test Do the t-test

```
t.test(officew_women3$weight, officew_men3$weight)
```

```
##
##  Welch Two Sample t-test
##
## data:  officew_women3$weight and officew_men3$weight
## t = -1.7555, df = 17.956, p-value = 0.09621
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -33.173987   2.973987
## sample estimates:
## mean of x mean of y
##      70.8      85.9
```

Alternative check of the t-statistic:

This is similar to what we did earlier with sampling inference. HERE, the mean is Zero, and the figure (Weight) we want to get the probability for is the the mean difference. We want to know the probability of the mean difference. H1 is the mean difference. H0 is the NULL

#6 steps to understanding the P-value

1 We want to test the probability that makes us believe that an effect(15)or difference between two groups is NOT happening by random chance

2.We assume the mean effect between these groups is zero meaning we assume that there is no effect

3. We run a ttest to get the prob of that effect happening and check the equivalent Probabiltiy. Note that we still assume our mean difference is zero

4.We can say that if that probability we get is 5% or less, then that is probability of that effect occuring at an assumption of a mean of zero.

5. It means the probability of that effect happening if we set our mean difference to zero is very low. Since the probability that it will occur is very low, we should not accept that our mean difference is zero.

6. So we reject that our mean difference is zero, and take the alternative hypothesis.

Remember that if the T-value is 2 (or 1.96) or more, it means the probability is 2.2% and 5% (two-tail) or less. This means we have 5% or less probability, that we will by chance get the difference in mean (15.1 -the effect) with the assumption that the null hypothesis is true (mean is zero). We are assuming that there is no difference, but that assumption will only occur 5% or less if the difference is -15.1. So due to this, we reject the null hypothesis and accept the alternative

Method 1

First: Calculate Standard Errors for both groups

```
se.women <- sd(officew_women3$weight) / sqrt(length(officew_women3$weight))
se.women
```

```
## [1] 5.928837
```

```
se.men <- sd(officew_men3$weight) / sqrt(length(officew_men3$weight))
se.men
```

```
## [1] 6.231551
```

sum (standardized version of) both standard errors:

```
se.diff <-   sqrt((se.women^2 + se.men^2))
se.diff
```

```
## [1] 8.601356
```

then calculate confidence intervals:

```
# t = (H1 - H0) / sem.diff
```

```
mean.diff <- mean(officew_women3$weight) - mean(officew_men3$weight)
mean.diff
```

```
## [1] -15.1
```

The t-value is

```
t <- (mean.diff - mean(0)) / se.diff
t # bigger than 1.96?
```

```
## [1] -1.755537
```

calculating t-value at 95% confidence interval and 18 degree of freedom

```
qt(0.975, 18)
```

```
## [1] 2.100922
```

I use a critical t value for 0.05 significance and 18 degrees of freedom The degree of freedom is calculated by:

```
dof <- nrow(officew_women3$weight) + nrow(officew_men3$weight) - 2
dof
```

```
## numeric(0)
```

the confidence interval

```
(qt(0.975, 18) * se.diff)
```

```
## [1] 18.07078
```

```
upper.ci <- mean.diff + (qt(0.975, 18) * se.diff)

lower.ci <- mean.diff - (qt(0.975, 18) * se.diff)
lower.ci
```

```
## [1] -33.17078
```

```
upper.ci
```

```
## [1] 2.970779
```

#Quiz Example

```
weight_men <- c(89, 75, 88, 75, 49, 89, 110, 120, 89, 75)

weight_women <- c(75, 76, 87, 110, 67, 76, 43, 55, 59, 60)
weight_men
```

```
##  [1]  89  75  88  75  49  89 110 120  89  75
```

```
weight_women
```

```
##  [1]  75  76  87 110  67  76  43  55  59  60
```

#Method 1

```
t.test(weight_men,weight_women)
```

```
##
##  Welch Two Sample t-test
##
## data:  weight_men and weight_women
## t = 1.7555, df = 17.956, p-value = 0.09621
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.973987 33.173987
## sample estimates:
## mean of x mean of y
##      85.9      70.8
```

#Method 2 (by hand)

```
se.quiz.women <- sd(weight_women) / sqrt(length(weight_women ))
se.quiz.women
```

```
## [1] 5.928837
```

```
se.quiz.men <- sd(weight_men) / sqrt(length(weight_men ))
se.quiz.men
```

```
## [1] 6.231551
```

Let us check the mean and standard deviation of men and women

```
mean(weight_men )
```

```
## [1] 85.9
```

```
sd(weight_men)
```

```
## [1] 19.70589
```

```
mean(weight_women )
```

```
## [1] 70.8
```

```
sd(weight_women)
```

```
## [1] 18.74863
```

We then need to the standardized difference between the two standard errors

```
se.diff_quiz <-   sqrt((se.quiz.men^2  + se.quiz.women^2 ))
se.diff_quiz
```

```
## [1] 8.601356
```

The difference in the mean of the two gender

```
mean.diff55 <- mean(weight_men) - mean(weight_women )
mean.diff55
```

```
## [1] 15.1
```

The upper bound of the confidence interval

```
upper.ci_quiz <- mean.diff55 + (qt(0.975, 18) * se.diff_quiz)
upper.ci_quiz
```

```
## [1] 33.17078
```

The lower bound of the confidence interval

```
lower.ci_quiz <- mean.diff55 - (qt(0.975, 18) * se.diff_quiz)
lower.ci_quiz
```

```
## [1] -2.970779
```

tvalue

```
tvaluequiz <- (mean.diff55 - mean(0)) / se.diff_quiz #t value
tvaluequiz
```

```
## [1] 1.755537
```

Men weighing above 100 kg

```
obess_quizm<- (100 - mean(weight_men))/sd(weight_men)
1-pnorm(obess_quizm)
```

```
## [1] 0.2371433
```

Wowen weighing above 100 kg

```
obess_quizf<- (100 - mean(weight_women))/sd(weight_women)
1-pnorm(obess_quizf)
```

```
## [1] 0.05968218
```

For question of proportion in the quiz

```
p <- 0.22
n <- 1200
se.prop<-sqrt(p*(1-p))/sqrt (n)

upperCI.prop <- p +(qt(0.975, (n-1))*se.prop)
upperCI.prop
```

```
## [1] 0.2434614
```

```
lowerCI.prop <- p -(qt(0.975, (n-1))*se.prop)
lowerCI.prop
```

```
## [1] 0.1965386
```

Interpretation? Can we reject H0?

No, we cannot reject the null hypothesis that the difference in the mean of both gender is zero at 95% confidence. We are 95% confident that the difference in mean of both gender is zero. We can also say that we cannot reject the null hypothesis that the difference in the mean of both gender will happen by random chance 5 times or more out of every 100 occurrence.

Using the First Method (t-value test), the t-statistic is 1.96, and our t-value is -1.75. If we plot this in a graph, 1.75 falls within regions lower than 1.96 (0.05 p-value), but we were only ready to accept region at -1.96 and above it

Using the second method (p-value test), our p-value as well is higher than 0.05. Our P-value is the probability of getting a result as extreme as our test statistic, assuming our NULL hypothesis is true that there is no difference in the mean.

Using the third method (CI test), our confidence interval is -33 to 2.with the mean as 15.1. Zero (0)is within the confidence interval that we are are 95 % confident the difference in mean between both gender can also be 0. We cannot reject the null hypothesis in this regard.

ANOTHER EXAMPLE OF T-TEST WITH NORMAL DISTRIBUTION

Make simulations replicable:

```
set.seed(101112)
```

disable scientific notation:

```
options(scipen=999)
```

We start by creating two different normally distributed variables:

```
var1 <- rnorm(50, mean = 0, sd = 1)
var1
```

```
##  [1] -0.75886627  1.35359814 -0.20107037 -0.44020778  1.29733664 -1.77972690
##  [7] -0.83939342  0.95828995  0.42984356  0.57647495  0.02182424  0.05155345
## [13] -1.44215582  3.09711393  0.60303556  1.56984330 -0.30714872 -0.87877014
## [19]  0.99365026  1.10491075  0.05902839 -0.72572592  0.61683306  0.76522140
## [25]  1.12996161  0.04939211  0.64416160 -0.20130345  0.79669099  0.84365116
## [31] -1.07974123  0.91799752 -1.82842883 -0.16075665  0.14761583 -1.46805313
## [37]  1.52113035  0.97308093  0.02674715 -0.08527182 -2.18919556 -0.09255031
## [43]  1.12027192  0.36227175 -1.08154375  2.31478977 -1.77177149 -1.66071484
```

```
## [49] -0.76352840  0.56684880
```

```r
var2 <- rnorm(100, mean = 0.5, sd = 3)
var2
```

```
##   [1]  0.50209787  0.90784791 -0.22941116  3.51909605  1.28583811  3.24834278
##   [7]  0.48008027  8.41920197 -0.93041455  4.62418359  2.88894985 -1.32263923
##  [13]  3.31639820 -3.02120655  1.41778850 -0.63817913  1.07666747  1.78747131
##  [19]  6.60213712 -4.44306297  3.67788246  3.30703994 -4.62364943  2.03825362
##  [25]  4.84540154 -1.72471828  0.41937474 -2.84170751  3.65945560 -5.21100921
##  [31]  4.37137774 -4.94866039 -5.60707819 -0.88447897 -0.52563202 -4.07731505
##  [37] -0.29093857  2.05032961  6.91266138  0.02084897 -2.78871444  1.17091235
##  [43]  1.35525897  3.02618274 -1.70754625 -1.62039019  6.22818944 -1.25320076
##  [49]  0.34662568  2.01195870  0.38273763 -1.89338378  1.46254231  0.48318039
##  [55]  1.15986634 -5.60941130  2.54299540  3.30143024 -1.77764401 -0.74128823
##  [61] -2.51898754  0.52648564 -3.82201313 -3.18110417 -0.39275039  3.04378387
##  [67] -3.34115600 -3.67115026  1.65222304  2.20707289  0.39242646 -4.34476888
##  [73]  0.47902270 -2.37498287 -2.36238327 -0.29233467  0.44808159  4.60699570
##  [79]  0.05713016 -0.77166218  2.89230534  1.50039635  1.43260044 -0.69679826
##  [85]  4.09618435 -2.92627607 -5.84466681  0.49394818 -0.90040102 -1.41043344
##  [91]  0.40839684  0.91381982  2.22293636  2.51697495  2.24030033  2.37997298
##  [97]  0.87324808  4.85636485 -0.59974494  0.12610685
```

What are their means?

```r
mean(var1)
```

```
## [1] 0.1031449
```

```r
mean(var2)
```

```
## [1] 0.3705409
```

Is there a significant difference?

```r
t.test(var1, var2)
```

```
##
##  Welch Two Sample t-test
##
## data:  var1 and var2
## t = -0.80081, df = 140.91, p-value = 0.4246
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9275146  0.3927225
## sample estimates:
## mean of x mean of y
## 0.1031449 0.3705409
```

Chi-square

Chi-Square test in R is a statistical method which used to determine if two categorical variables have a significant correlation between them. The difference with x2 is between the observed frequency (fo) and the expected frequency (fe).

H0: every i.v. category should have the same distribution across the d.v. as the total,i.e. i.v. doesn't matter.

Let us assume you that in the process of the review, an argument from one of your HR staff is that men are more single than women in the organization.

We are interested in knowing whether gender affect being single We want to know if whether either you are a male of female has an effect on the marital status

Here gender is the IV and Marital Status is the DV

STEP 1- DERIVE A CONTIGENCY TABLE Let us first divide our martital status into two concrete divisions -

Single Vs Not Single

```
table(officew$Marstatus)
```

```
##
## Divorced  Married   Single
##        4        5       11
```

```
table(officew$Marstatus=="Single")
```

```
##
## FALSE  TRUE
##     9    11
```

```
officew$NewMarStatus<- ifelse(officew$Marstatus=="Single",
                        "Single", "Not Single")
officew$NewMarStatus
```

```
##  [1] "Not Single" "Not Single" "Single"     "Single"     "Single"
##  [6] "Single"     "Not Single" "Single"     "Not Single" "Single"
## [11] "Not Single" "Single"     "Single"     "Not Single" "Single"
## [16] "Single"     "Not Single" "Single"     "Not Single" "Not Single"
```

We then use the table function to show the cross tab

Converting NewMarStatus to a Factor

```
officew$NewMarStatus <- as.factor(as.character(officew$NewMarStatus))
officew$NewMarStatus
```

```
##  [1] Not Single Not Single Single     Single     Single     Single
##  [7] Not Single Single     Not Single Single     Not Single Single
## [13] Single     Not Single Single     Single     Not Single Single
## [19] Not Single Not Single
## Levels: Not Single Single
```

```
table(officew$Gender, officew$NewMarStatus)
```

```
##
##          Not Single Single
##   FeMale          5      5
##   Male            4      6
```

To get the percentages, we use prop.table function

```
prop.table(table( officew$NewMarStatus,officew$Gender), 2)
```

```
##
##               FeMale Male
##   Not Single     0.5  0.4
##   Single         0.5  0.6
```

50 percent of female are single, and 60 percent of males are single. We can say the effect of being a male is 10 percentage point higher for men than women.

```
prop.table(table(officew$NewMarStatus, officew$Gender), 1)
```

```
##
##                 FeMale      Male
##   Not Single 0.5555556 0.4444444
##   Single     0.4545455 0.5454545
```

#ALTERNATIVE 2: Install Gmodel package

```
install.packages("gmodels")
library(gmodels)
CrossTable(officew$Gender, officew$NewMarStatus)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  20
##
##
##                | officew$NewMarStatus
## officew$Gender | Not Single |     Single |  Row Total |
## ---------------|------------|------------|------------|
##         FeMale |          5 |          5 |         10 |
##                |      0.056 |      0.045 |            |
##                |      0.500 |      0.500 |      0.500 |
##                |      0.556 |      0.455 |            |
##                |      0.250 |      0.250 |            |
## ---------------|------------|------------|------------|
##           Male |          4 |          6 |         10 |
##                |      0.056 |      0.045 |            |
##                |      0.400 |      0.600 |      0.500 |
##                |      0.444 |      0.545 |            |
##                |      0.200 |      0.300 |            |
## ---------------|------------|------------|------------|
##   Column Total |          9 |         11 |         20 |
##                |      0.450 |      0.550 |            |
## ---------------|------------|------------|------------|
##
##
```

Let us assume we want to CONTROL for location. We think we can also use the location to staff (either they stay in the central city across both genders to know wnether they are single or not. You just need to insert the new variable to the TABLE function

```
table(officew$Gender, officew$NewMarStatus, officew$CityCentral)
```

```
## , ,  = No
```

```
##
##
##          Not Single Single
##   FeMale          4      2
##   Male            2      0
##
## , ,  = Yes
##
##
##          Not Single Single
##   FeMale          1      3
##   Male            2      6
```

```
prop.table (table(officew$Gender, officew$NewMarStatus, officew$CityCentral), 3)
```

```
## , ,  = No
##
##
##          Not Single     Single
##   FeMale 0.50000000 0.25000000
##   Male   0.25000000 0.00000000
##
## , ,  = Yes
##
##
##          Not Single     Single
##   FeMale 0.08333333 0.25000000
##   Male   0.16666667 0.50000000
```

It might be more convenient to create two subsets of the data one for those who live in Central Area, and one for those who don't.

For people who live in the central area

```
officew_central <- officew[officew$CityCentral=="Yes",]
officew_central
```

```
##     X Gender weight income rating Marstatus CityCentral NewMarStatus
## 1   1   Male     89  50000      5   Married         Yes   Not Single
## 3   3   Male     88 120000      2    Single         Yes       Single
## 4   4   Male     75 800000      4    Single         Yes       Single
## 5   5   Male     49 650000      9    Single         Yes       Single
## 6   6   Male     89  92000      9    Single         Yes       Single
## 7   7   Male    110  94000      8  Divorced         Yes   Not Single
## 8   8   Male    120 222000      1    Single         Yes       Single
## 10 10   Male     75  75000      7    Single         Yes       Single
## 13 13 FeMale     87  99000      6    Single         Yes       Single
## 14 14 FeMale    110 450000      1  Divorced         Yes   Not Single
## 16 16 FeMale     76 190000      1    Single         Yes       Single
## 18 18 FeMale     55 780000      6    Single         Yes       Single
```

For people who DO NOT live in the central area

```
officew_Nocentral <- officew[officew$CityCentral=="No",]
officew_Nocentral
```

```
##     X Gender weight income rating Marstatus CityCentral NewMarStatus
## 2   2   Male     75  95000      1   Married          No   Not Single
```

```
## 9   9   Male     89 543000      9   Married          No   Not Single
## 11 11 FeMale     75  63000      5   Married          No   Not Single
## 12 12 FeMale     76  40000      6    Single          No       Single
## 15 15 FeMale     67 180000      1    Single          No       Single
## 17 17 FeMale     43  96000      3  Divorced          No   Not Single
## 19 19 FeMale     59 150000      9   Married          No   Not Single
## 20 20 FeMale     60 342000      4  Divorced          No   Not Single
```

With these subsets, you can obtain the cross-tabulations separately and in percentage form

For people who live in the central area.

```
prop.table (table(officew_central$Gender, officew_central$NewMarStatus),2)
```

```
##
##          Not Single    Single
##   FeMale  0.3333333 0.3333333
##   Male    0.6666667 0.6666667
```

For people who DO NOT live in the central area

```
prop.table (table(officew_Nocentral$Gender, officew_Nocentral$NewMarStatus),2)
```

```
##
##          Not Single    Single
##   FeMale  0.6666667 1.0000000
##   Male    0.3333333 0.0000000
```

STEP 2: CONDUCT a t-test and check the chi square(x2) and p value

```
chisq.test(officew$Gender,officew$NewMarStatus,correct=FALSE)
```

```
## Warning in chisq.test(officew$Gender, officew$NewMarStatus, correct = FALSE):
## Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  officew$Gender and officew$NewMarStatus
## X-squared = 0.20202, df = 1, p-value = 0.6531
```

It gave the warning because many of the expected values will be very small and therefore the approximations of p may not be right.

In R you can use chisq.test(a, simulate.p.value = TRUE) to use simulate p values.

```
chisq.test(officew$Gender,officew$NewMarStatus, simulate.p.value = TRUE)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  officew$Gender and officew$NewMarStatus
## X-squared = 0.20202, df = NA, p-value = 1
```

However, with such small cell sizes, all estimates will be poor. It might be good to just test pass vs. fail (deleting "no show").

Either with chi-square or logistic regression. Indeed, since it is pretty clear that the pass/fail grade is a dependent variable,logistic regression might be better

Correlation

packages:

```
install.packages("corrplot") # Install the corrplot library, for nice-looking
# correlation plots. Do this once.
install.packages("ggplot2") # Install the ggplot2 package, for high-quality
# graphs
install.packages("cowplot") # Install the cowplot package, to arrange plots
# into a grid
install.packages("ggpubr")
```

```
library(corrplot) # Plotting nice correlation matrix
library(cowplot) # arranging plots into a grid
library(ggplot2) # high-quality graphs
```

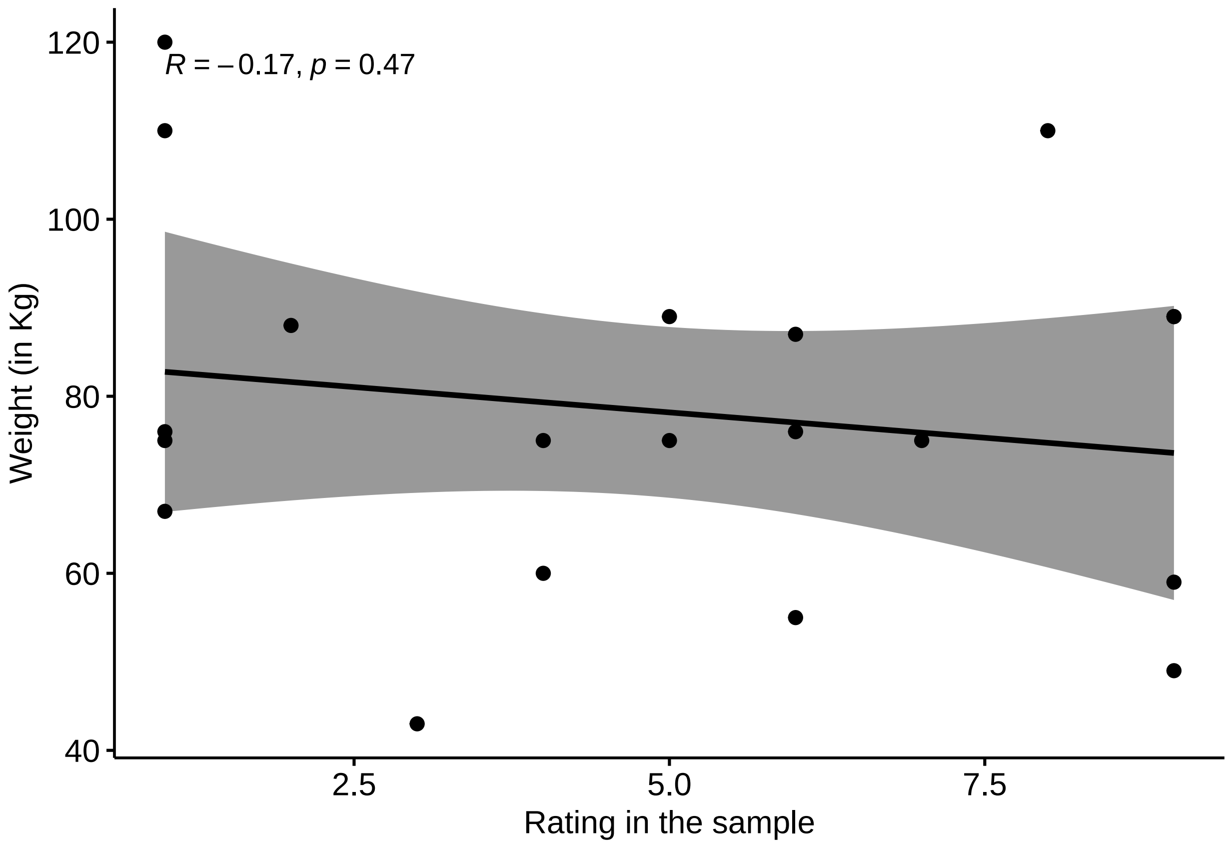
Optional packages: check the data properties

```
str(officew)
```

```
## 'data.frame':	20 obs. of  8 variables:
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender      : chr  "Male" "Male" "Male" "Male" ...
##  $ weight      : int  89 75 88 75 49 89 110 120 89 75 ...
##  $ income      : num  50000 95000 120000 800000 650000 92000 94000 222000 543000 75000 ...
##  $ rating      : int  5 1 2 4 9 9 8 1 9 7 ...
##  $ Marstatus   : chr  "Married" "Married" "Single" "Single" ...
##  $ CityCentral : chr  "Yes" "No" "Yes" "Yes" ...
##  $ NewMarStatus: Factor w/ 2 levels "Not Single","Single": 1 1 2 2 2 2 1 2 1 2 ...
```

plot the graph , use y as income

```
library("ggpubr")
ggscatter(officew, x = "rating", y = "weight",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Rating in the sample", ylab = "Weight (in Kg)")
```




```
cor(officew$weight,officew$rating)
```

```
## [1] -0.1713778
```

```
str(officew)
```

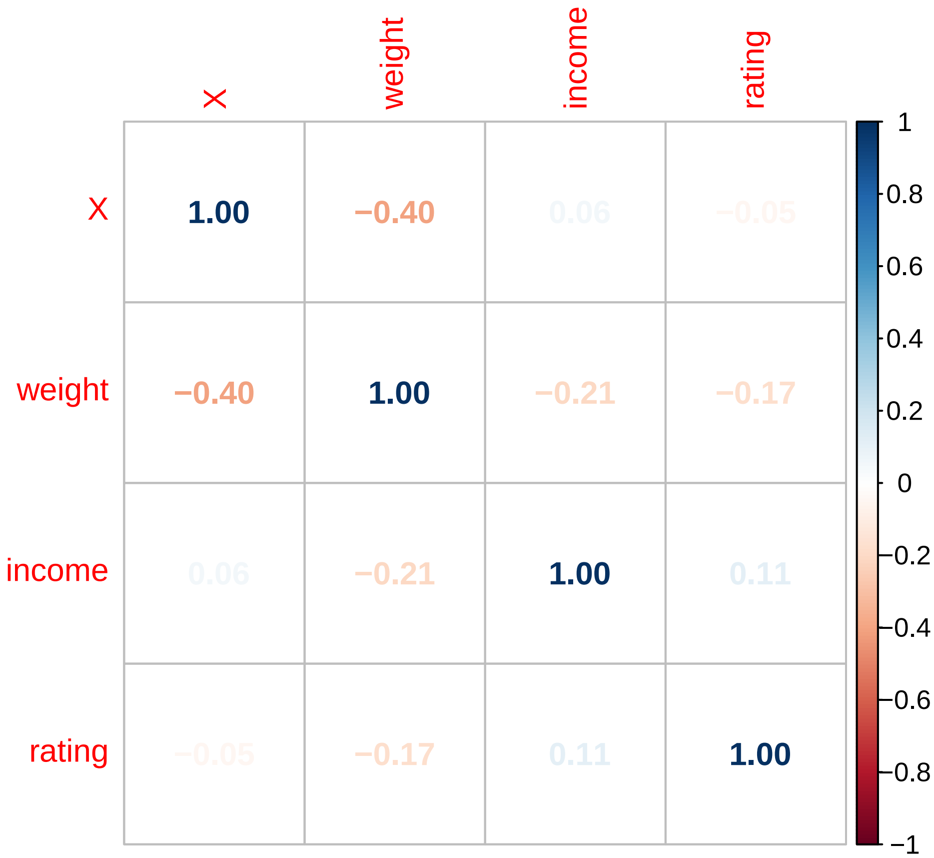
```
## 'data.frame':    20 obs. of  8 variables:
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender      : chr  "Male" "Male" "Male" "Male" ...
##  $ weight      : int  89 75 88 75 49 89 110 120 89 75 ...
##  $ income      : num  50000 95000 120000 800000 650000 92000 94000 222000 543000 75000 ...
##  $ rating      : int  5 1 2 4 9 9 8 1 9 7 ...
##  $ Marstatus   : chr  "Married" "Married" "Single" "Single" ...
##  $ CityCentral : chr  "Yes" "No" "Yes" "Yes" ...
##  $ NewMarStatus: Factor w/ 2 levels "Not Single","Single": 1 1 2 2 2 2 1 2 1 2 ...
```

```
corr1<- cor(officew[c(-2,-6,-7,-8)]) #we do it without the
#last non-numeric variable "type" which are indexied in 3, 4 and 5
corr1
```

```
##                  X     weight      income      rating
## X       1.00000000 -0.4037699  0.05917688 -0.04549472
## weight -0.40376994  1.0000000 -0.20964475 -0.17137780
## income  0.05917688 -0.2096447  1.00000000  0.11266232
## rating -0.04549472 -0.1713778  0.11266232  1.00000000
```

Nice correlation matrix ?corrplot

```
library(corrplot)
plot1<-corrplot(corr1, method = "number") # Try with different methods!
```




```
plot1
```

```
## $corr
##                  X     weight      income      rating
## X       1.00000000 -0.4037699  0.05917688 -0.04549472
## weight -0.40376994  1.0000000 -0.20964475 -0.17137780
## income  0.05917688 -0.2096447  1.00000000  0.11266232
## rating -0.04549472 -0.1713778  0.11266232  1.00000000
##
## $corrPos
##     xName  yName x y        corr
## 1       X      X 1 4  1.00000000
## 2       X weight 1 3 -0.40376994
## 3       X income 1 2  0.05917688
## 4       X rating 1 1 -0.04549472
## 5  weight      X 2 4 -0.40376994
## 6  weight weight 2 3  1.00000000
## 7  weight income 2 2 -0.20964475
## 8  weight rating 2 1 -0.17137780
## 9  income      X 3 4  0.05917688
## 10 income weight 3 3 -0.20964475
## 11 income income 3 2  1.00000000
## 12 income rating 3 1  0.11266232
## 13 rating      X 4 4 -0.04549472
## 14 rating weight 4 3 -0.17137780
## 15 rating income 4 2  0.11266232
## 16 rating rating 4 1  1.00000000
```

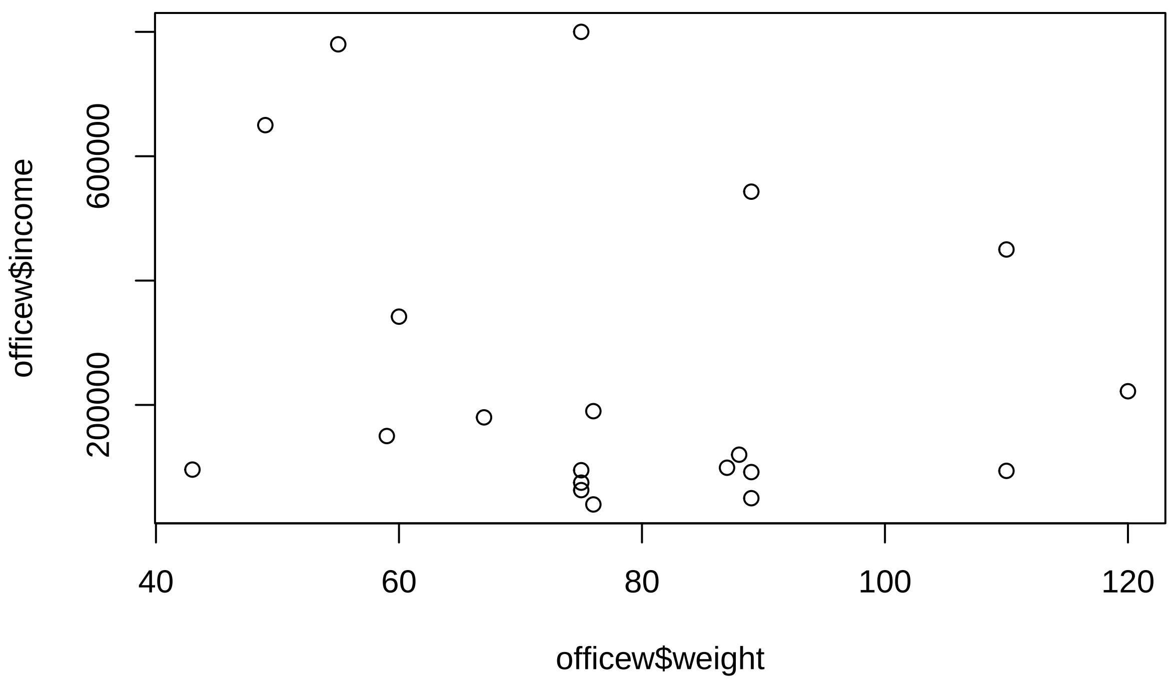
```
##
## $arg
## $arg$type
## [1] "full"
```

REGRESSION

STEP 1, CHECK THE PLOT

How does this look? Let's plot the data!

```
plot2<-plot(officew$weight, officew$income) # First x axis, then y axis:
```




```
plot2
```

```
## NULL
```

Our independent variable is income

STEP 2 Run your first (bivariate) regression

```
myfirstreg <- lm(weight ~ income, data= officew) #First you run it
summary(myfirstreg) #Then you see the output
```

```
##
## Call:
## lm(formula = weight ~ income, data = officew)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.083 -13.100  -4.760   7.456  41.062
##
```

```
## Coefficients:
##                Estimate  Std. Error t value      Pr(>|t|)
## (Intercept) 82.71765777  6.61544011   12.50 0.00000000026 ***
## income      -0.00001702  0.00001872   -0.91         0.375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.35 on 18 degrees of freedom
## Multiple R-squared:  0.04395,    Adjusted R-squared:  -0.009163
## F-statistic: 0.8275 on 1 and 18 DF,  p-value: 0.375
```

```
confint(myfirstreg, level=0.99)
```

```
##                      0.5 %          99.5 %
## (Intercept) 63.67550720175 101.75980833575
## income      -0.00007089542   0.00003684625
```

#STEP 3 PRINT your Result

```
install.packages("stargazer") # Install stargazer, for nice-looking regression
# tables. Do this once
library(stargazer)
# Lets get a nice table out of it
stargazer(myfirstreg, title="Regression Results", out="reg.txt")
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Tue, Dec 07, 2021 - 23:33:34
## \begin{table}[!htbp] \centering
##   \caption{Regression Results}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{1}{c}{\textit{Dependent variable:}} \\
## \cline{2-2}
## \\[-1.8ex] & weight \\
## \hline \\[-1.8ex]
##  income & $-$0.00002 \\
##   & (0.00002) \\
##   & \\
##  Constant & 82.718$^{***}$ \\
##   & (6.615) \\
##   & \\
## \hline \\[-1.8ex]
## Observations & 20 \\
## R$^{2}$ & 0.044 \\
## Adjusted R$^{2}$ & $-$0.009 \\
## Residual Std. Error & 20.352 (df = 18) \\
## F Statistic & 0.827 (df = 1; 18) \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{1}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

#STEP 4 Interprete the result

```
options(scipen=999) #run this once to turn off scientific notation in
#your reg output
myfirstreg2 <- lm(weight ~ income+rating, data= officew) #First you run it
summary(myfirstreg2) #Then you see the output
```

```
##
## Call:
## lm(formula = weight ~ income + rating, data = officew)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.714 -14.604  -4.316   8.731  37.258
##
## Coefficients:
##                Estimate  Std. Error t value     Pr(>|t|)
## (Intercept) 87.21803671  9.77667182   8.921 0.0000000803 ***
## income      -0.00001566  0.00001916  -0.817        0.425
## rating      -1.00034050  1.57672115  -0.634        0.534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.7 on 17 degrees of freedom
## Multiple R-squared:  0.06606,    Adjusted R-squared:  -0.04381
## F-statistic: 0.6013 on 2 and 17 DF,  p-value: 0.5594
```

MULTICOLLONEARITY (VIF & TOLERANCE) & Post Treatment Effect

```
install.packages("carData")
install.packages("car")
library(car)
library(carData)
```

VIF and Tolerance Let us test for the multicollinearity of both rating and income on our dependent variable

'VIF test The square root of the VIF tells us by which factor at which the standard error for the coefficient of the IV will be larger than if that if it had 0 correlation with other independent variables.

?vif

```
vif(myfirstreg2)
```

```
##   income   rating
## 1.012856 1.012856
```

The Standard Error of the CE of income will be inflated by 1.012 if we include it in the model.

Tolerance is proportion of the model's independent variables not explained by other independent variables. The tolerance is the inverse of the vif and is the Percent of variance in the predictor that cannot be accounted for by other predictors.

```
1/vif(myfirstreg2)
```

```
##    income    rating
## 0.9873072 0.9873072
```

For example, if you run the VIF, 98 percent of the variance of income cannot be explained by other. This is where there is no correlation. It is what is unique to this variable income, that can't be explained by any other in the set

Post Treatment Bias

```
library(AER)
data("Fatalities")
```

```
fatal <- lm(fatal~beertax + youngdrivers + miles + pop, data = Fatalities)
summary(fatal) # model with a number of covariates to isolate effect of drunk driving
```

```
##
## Call:
## lm(formula = fatal ~ beertax + youngdrivers + miles + pop, data = Fatalities)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1196.13  -110.66    -7.06   116.84  1355.82
##
## Coefficients:
##                  Estimate     Std. Error t value            Pr(>|t|)
## (Intercept)  -811.074536041  146.194713239  -5.548    0.000000059272093 ***
## beertax       225.858365309   30.185471310   7.482    0.000000000000664 ***
## youngdrivers 1133.885402562  591.746236039   1.916               0.0562 .
## miles           0.065143934    0.009817656   6.635    0.000000000132276 ***
## pop             0.000182334    0.000002887  63.148 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252.4 on 331 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.927
## F-statistic:  1064 on 4 and 331 DF,  p-value: < 0.00000000000000022
```

```
fatal.ptb <- lm(fatal ~ beertax +youngdrivers + miles + pop + spirits,
                data = Fatalities)
summary(fatal.ptb) # adding control for the mechanism (spirits consumption)
```

```
##
## Call:
## lm(formula = fatal ~ beertax + youngdrivers + miles + pop + spirits,
##     data = Fatalities)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1191.92  -110.87    -5.48   118.65  1355.78
##
## Coefficients:
##                  Estimate     Std. Error t value            Pr(>|t|)
## (Intercept)  -773.900635635  154.946661376  -4.995     0.00000095728464 ***
## beertax       224.347841268   30.278022192   7.410     0.00000000000107 ***
## youngdrivers 1105.951671017  593.407198244   1.864               0.0632 .
## miles           0.064623804    0.009850549   6.560     0.00000000020754 ***
## pop             0.000182124    0.000002904  62.719 < 0.0000000000000002 ***
```

```
## spirits       -14.863322043   20.407940241  -0.728              0.4669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252.6 on 330 degrees of freedom
## Multiple R-squared:  0.928,  Adjusted R-squared:  0.9269
## F-statistic: 850.3 on 5 and 330 DF,  p-value: < 0.00000000000000022
```

```
stargazer(fatal, fatal.ptb, type = "text", data = Fatalities,
          out = "fatal.reg.txt")
```

```
##
## ======================================================================
##                                    Dependent variable:
##                     --------------------------------------------------
##                                           fatal
##                               (1)                       (2)
## ----------------------------------------------------------------------
## beertax                   225.858***                224.348***
##                            (30.185)                  (30.278)
##
## youngdrivers              1,133.885*                1,105.952*
##                           (591.746)                 (593.407)
##
## miles                      0.065***                  0.065***
##                            (0.010)                   (0.010)
##
## pop                       0.0002***                 0.0002***
##                           (0.00000)                 (0.00000)
##
## spirits                                              -14.863
##                                                      (20.408)
##
## Constant                 -811.075***               -773.901***
##                           (146.195)                 (154.947)
##
## ----------------------------------------------------------------------
## Observations                 336                       336
## R2                          0.928                     0.928
## Adjusted R2                 0.927                     0.927
## Residual Std. Error    252.400 (df = 331)        252.579 (df = 330)
## F Statistic         1,064.206*** (df = 4; 331) 850.264*** (df = 5; 330)
## ======================================================================
## Note:                                      *p<0.1; **p<0.05; ***p<0.01
##
## ===========================================================================================================
## Statistic     N      Mean       St. Dev.      Min       Pctl(25)     Pctl(75)       Max
## -----------------------------------------------------------------------------------------------------------
## spirits      336     1.754        0.684       0.790       1.300        2.012        4.900
## unemp        336     7.347        2.533       2.400       5.475        8.900       18.000
## income       336  13,880.180    2,253.046   9,513.762  12,085.850   15,175.120   22,193.460
## emppop       336    60.806        4.722      42.993      57.691       64.413       71.269
## beertax      336     0.513        0.478       0.043       0.209        0.652        2.721
## baptist      336     7.157        9.763       0.000       0.627       13.127       30.356
## mormon       336     2.802        9.665       0.100       0.272        0.629       65.916
```

```
## drinkage     336    20.456         0.899        18           20           21           21
## dry          336     4.267         9.501       0.000        0.000        2.425        45.792
## youngdrivers 336     0.186         0.025       0.073        0.170        0.202        0.282
## miles        336   7,890.754     1,475.659    4,576.346    7,182.539    8,504.015    26,148.270
## fatal        336    928.664       934.051       79          293.8       1,063.5       5,504
## nfatal       336    182.583       188.431       13           53.8         212         1,049
## sfatal       336    109.949       108.540        8            35          131          603
## fatal1517    336    62.610        55.729         3           25.8          77          318
## nfatal1517   336    12.262        12.253         0            4           15.2          76
## fatal1820    336    106.661       104.224        7            38         130.2         601
## nfatal1820   336    33.527        33.238         0            11           44          196
## fatal2124    336    126.872       131.789       12            42         150.5         770
## nfatal2124   336    41.378        42.930         1            13           49          249
## afatal       336    293.333       303.581      24.600       90.498      363.958     2,094.900
## pop          336 4,930,272.000 5,073,704.000 478,999.700 1,545,251.000 5,751,735.000 28,314,028.000
## pop1517      336  230,815.500   229,896.300  21,000.020   71,749.930   270,500.200  1,172,000.000
## pop1820      336  249,090.400   249,345.600  20,999.960   76,962.120   308,311.400  1,321,004.000
## pop2124      336  336,389.900   345,304.400  30,000.160  103,500.000  413,000.100  1,892,998.000
## milestot     336   37,101.490    37,454.370     3,993      11,691.5     44,139.8      241,575
## unempus      336     7.529         1.479        5.500        6.200        9.600        9.700
## emppopus     336    59.971         1.585       57.800       57.900       61.500       62.300
## gsp          336     0.025         0.043       -0.124        0.001        0.057        0.142
## -----------------------------------------------------------------------------------------------
```

Here controling for the mechanism causes part of the effect of beertax to be mathematically "soaked up". Admittedly, the effect is a little weak.

Dummy Variable and Binomial Regression

Convert gender to dummy variable, where male is 1 and female is 0. Male is our baseline variable

```
officew$highincome<- ifelse(officew$income>120000, 1, 0)
officew$highincome
```

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

```
officew$overweight<- ifelse(officew$weight >=100, 1, 0)
officew$overweight
```

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

Run regression

```
reg11 <- lm(overweight ~ Gender,data = officew)
summary (reg11)
```

```
##
## Call:
## lm(formula = overweight ~ Gender, data = officew)
##
## Residuals:
##    Min      1Q Median      3Q     Max
##   -0.2    -0.2   -0.1    -0.1     0.9
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.1000     0.1179   0.849    0.407
## GenderMale     0.1000     0.1667   0.600    0.556
##
## Residual standard error: 0.3727 on 18 degrees of freedom
## Multiple R-squared:  0.01961,    Adjusted R-squared:  -0.03486
## F-statistic:  0.36 on 1 and 18 DF,  p-value: 0.556
```

#INTEPRETE THE RESULT Men who are overweight will weigh 10 kg higher than the reference group - females who are overweighted

Run regression

```
options(scripen = 999)
reg10<- lm(overweight ~ Gender + officew$highincome,data = officew)
summary (reg10)
```

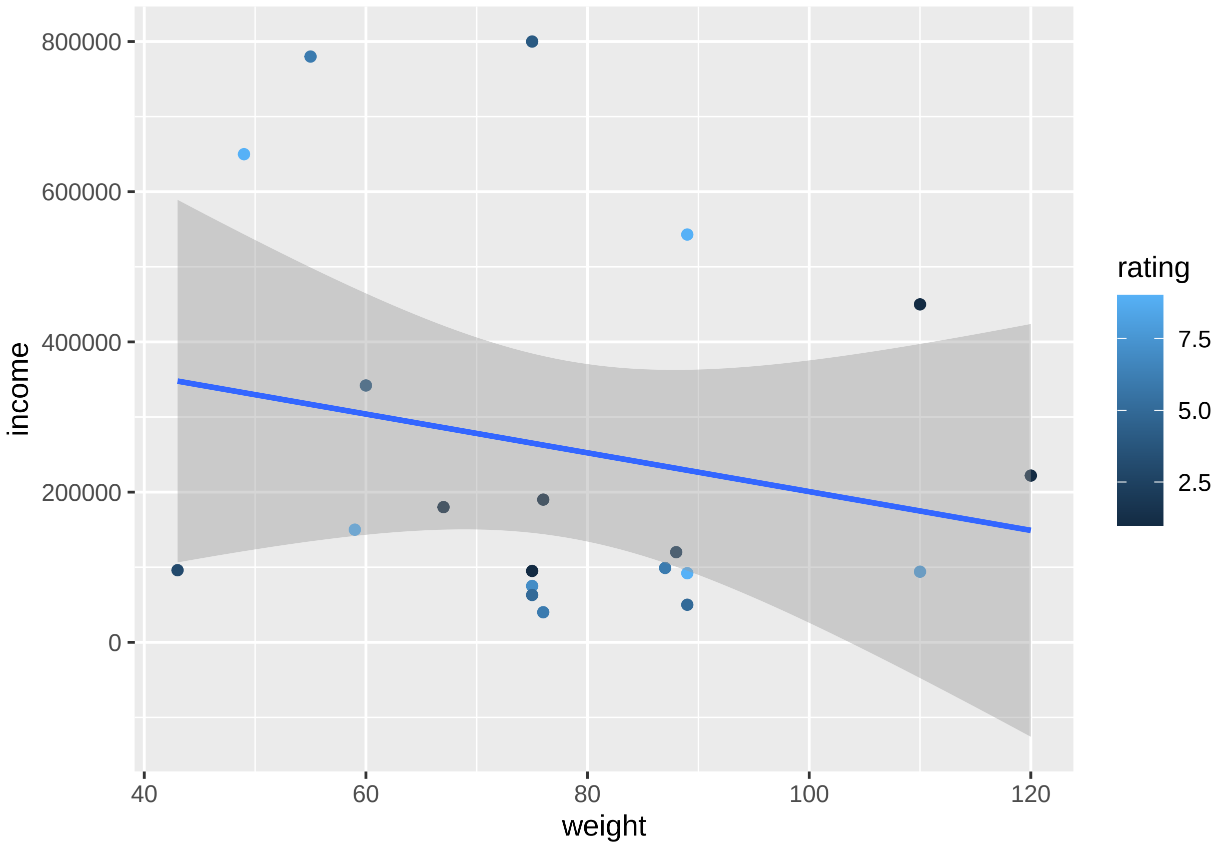
```
##
## Call:
## lm(formula = overweight ~ Gender + officew$highincome, data = officew)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.275 -0.150 -0.150 -0.025  0.850
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.0250     0.1580   0.158    0.876
## GenderMale           0.1250     0.1724   0.725    0.478
## officew$highincome   0.1250     0.1724   0.725    0.478
##
## Residual standard error: 0.3777 on 17 degrees of freedom
## Multiple R-squared:  0.04902,    Adjusted R-squared:  -0.06286
## F-statistic: 0.4381 on 2 and 17 DF,  p-value: 0.6523
```

This suggests that, after effects of highincome are taken into account, men will weight 12kg higher than the reference group (women).

GGPPLOT and INTERACTION EFFECT

Graphically using ggplot

```
plot3<-ggplot(officew, aes(x = weight, y = income, colour = rating)) +
  geom_point() +
  geom_smooth(method = "lm")
plot3
```



What if we are interested on how the effect of `highincome` works across `Gender` which is the interaction effect of both variables. You will use *

```
options(scripen =100, "digits"=3)
reg16 <- lm(overweight ~ Gender + highincome+ Gender*highincome,data = officew)
summary (reg16)
```

```
##
## Call:
## lm(formula = overweight ~ Gender + highincome + Gender * highincome,
##     data = officew)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.250 -0.167 -0.167  0.000  0.833
##
## Coefficients:
##                                    Estimate              Std. Error t value
## (Intercept)           0.0000000000000000621  0.1943203969393503261    0.00
## GenderMale            0.1666666666666666019  0.2508665537248394584    0.66
## highincome            0.1666666666666665464  0.2508665537248394584    0.66
## GenderMale:highincome -0.0833333333333331761  0.3547788826234666293   -0.23
##                       Pr(>|t|)
## (Intercept)               1.00
## GenderMale                0.52
## highincome                0.52
## GenderMale:highincome     0.82
##
```

```
## Residual standard error: 0.389 on 16 degrees of freedom
## Multiple R-squared:  0.0523, Adjusted R-squared:  -0.125
## F-statistic: 0.294 on 3 and 16 DF,  p-value: 0.829
```

```
reg15 <- lm(weight ~ Gender + highincome+ Gender*highincome,data = officew)
summary (reg15)
```

```
##
## Call:
## lm(formula = weight ~ Gender + highincome + Gender * highincome,
##     data = officew)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34.25 -12.29   0.83   5.75  38.83
##
## Coefficients:
##                      Estimate Std. Error t value  Pr(>|t|)
## (Intercept)            70.250     10.162    6.91 0.0000035 ***
## GenderMale             17.417     13.120    1.33      0.20
## highincome              0.917     13.120    0.07      0.95
## GenderMale:highincome  -5.333     18.554   -0.29      0.78
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.3 on 16 degrees of freedom
## Multiple R-squared:  0.152,  Adjusted R-squared:  -0.00647
## F-statistic: 0.959 on 3 and 16 DF,  p-value: 0.436
```

Interpretation.

You will also notice that the Rsquared has increased to 15 percent which means the model now account more variation of the dependent variable `weight`. That means explaining the effect of gender on `weight` works through the `income` staff receives.

The `weight` of men with `higherincome` is reduced by 5kg compared to women with `higherincome`. However, on average, men weigh (17.4kg -5.3kg) about 11.9kg more than effect of `income` held constant.

The average `weight` of men, the effect of `income` held constant, can still be derived as (70+17.4-(5.333)) = 82.1kg. The average `weight` of men = 70.2 kg (which is the intercept).

The `weight` of men over women (82-1 - 70.2)kg is 11.9kg which is what we got earlier.