Assignment 2

Madison Wozniak

9/9/21

Assignment 2

You are asked to submit both the R Markdown file and its pdf output.

Q1. Using R, compute the following

$$\frac{0.35 - 0.3}{\sqrt{\frac{0.2*(1-0.4)}{50}}}$$

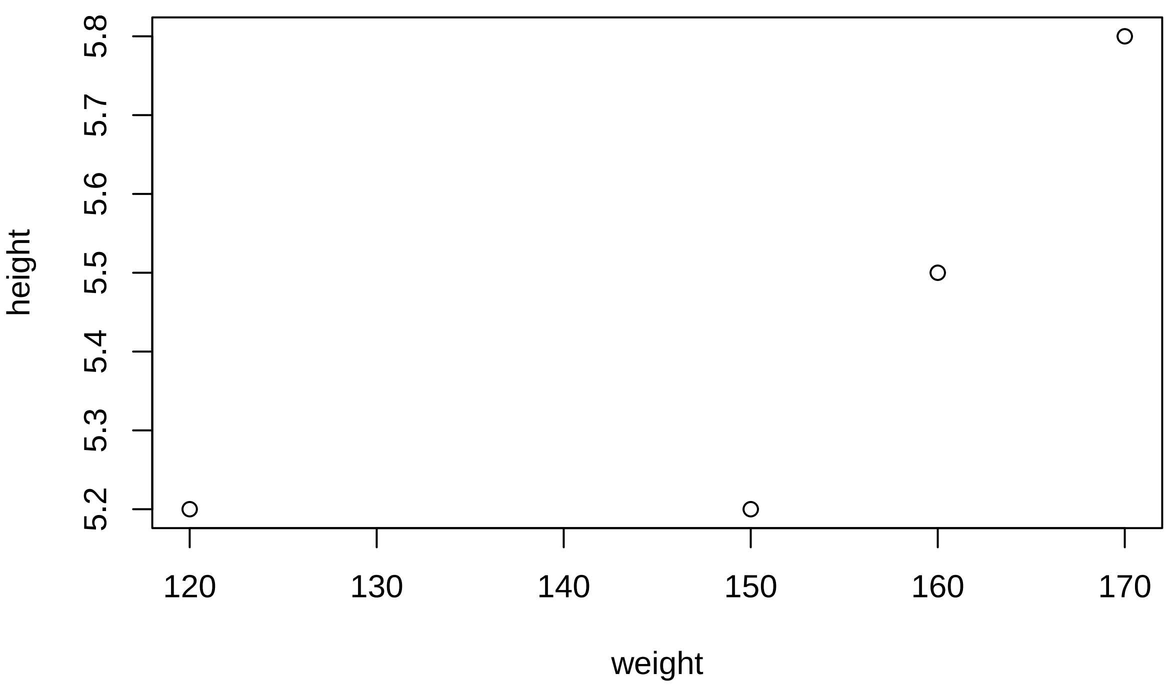
Answer

```
(0.35-0.3)/(sqrt((0.2*(1-0.4))/(50)))
```

```
## [1] 1.020621
```

Q2. Define two variables`weight = [150, 160, 170, 120]` and `height = [5.2, 5.5, 5.8, 5.2]` and plot weight vs height.

```
weight<-c(150,160,170,120)
height<-c(5.2,5.5,5.8,5.2)
plot(weight,height)
```



Q3. Without running any code, predict the outcome of each line:

x <- c(2, 3, 5, 7, 11, 13, 17, 19, 23, 29) x[1:5]

x<-c(1,2,3,4,5)

x[c(1, 4)]

Select rows 1 and 4

x[-c(2, 5)]

Remove rows 2 and 5

Q4. Generate the matrix below

$$\begin{bmatrix} 1 & 4 & 7 & 3 \\ 2 & 4 & 3 & 8 \\ 3 & 2 & 1 & 5 \end{bmatrix}$$

```
G<-matrix(c(1,2,3,4,4,2,7,3,1,3,8,5),3,4)
G
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7    3
## [2,]    2    4    3    8
## [3,]    3    2    1    5
```

(a) Report the 2nd and the 3rd row.

```
G[c(2,3),]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    4    3    8
## [2,]    3    2    1    5
```

(b) Report all columns except the 2nd one.

```
G[,-2]
```

```
##      [,1] [,2] [,3]
## [1,]    1    7    3
## [2,]    2    3    8
## [3,]    3    1    5
```

(c) Rename row and column names to your names of choice.

```
rownames(G)<-c("lions","tigers","bears")
colnames(G)<-c("pink","purple","green","gold")
```

(d) Call the second and the third row using the names you defined.

```
G[c("tigers","bears"),]
```

```
##        pink purple green gold
## tigers    2      4     3    8
## bears     3      2     1    5
```

Q5. Create a dataframe with four features (columns), first is called `no_bedrooms`, `location`, `age`, `price`. Here is the info for five houses:

House 1: 4, 'Boston', 35, $500K

House 2: 1, 'San Francisco', 55, $900K

House 3: 4, 'Hartford', 87, $300K

House 4: 3, ‘Houston’, 45, $280K

House 5: 3, ‘Seattle’, 35, $850K

```r
no_bedrooms<-c(4,1,4,3,3)
location<-c("Boston","San Francisco","Hartford","Houston","Seattle")
age<-c(35,55,87,45,35)
price<-c(500000,900000,300000,280000,850000)
df<-data.frame(no_bedrooms,location,age,price)
df
```

```
##   no_bedrooms      location age  price
## 1           4        Boston  35 500000
## 2           1 San Francisco  55 900000
## 3           4      Hartford  87 300000
## 4           3       Houston  45 280000
## 5           3       Seattle  35 850000
```

Q6. Load the dataset that is already built-in data in R: `data(mtcars)`.

(a) How many observations are there in this dataset?

```r
data(mtcars)
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
```

```
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

32 observations (b) How many variables does this dataset have? What are the names of these variables?

11 variables: `mpg,cyl,disp,hp,drat,wt,qsec,vs,am,gear,carb`

(c) What are the classes of the variables?

```
class(mtcars$mpg)
```

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

```
class(mtcars$cyl)
```

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

```
class(mtcars$disp)
```

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

```
class(mtcars$hp)
```

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

```
class(mtcars$drat)
```

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

```
class(mtcars$wt)
```

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

```
class(mtcars$qsec)
```

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

(d) What are the statistics of `mpg` such as mean, median, standard deviation.

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

```
median(mtcars$mpg)
```

```
## [1] 19.2
```

```
sd(mtcars$mpg)
```

```
## [1] 6.026948
```

(e) How many observations are with `drat`>3?

```
log_drat<-(mtcars$drat>3)
```

Q7. Start by loading the library and data:

```
library(dslabs)
data(murders)
murders
```

```
##                    state abb        region population total
## 1                Alabama  AL         South    4779736   135
## 2                 Alaska  AK          West     710231    19
## 3                Arizona  AZ          West    6392017   232
## 4               Arkansas  AR         South    2915918    93
## 5             California  CA          West   37253956  1257
## 6               Colorado  CO          West    5029196    65
## 7            Connecticut  CT     Northeast    3574097    97
## 8               Delaware  DE         South     897934    38
## 9   District of Columbia  DC         South     601723    99
## 10               Florida  FL         South   19687653   669
## 11               Georgia  GA         South    9920000   376
## 12                Hawaii  HI          West    1360301     7
## 13                 Idaho  ID          West    1567582    12
## 14              Illinois  IL North Central   12830632   364
## 15               Indiana  IN North Central    6483802   142
## 16                  Iowa  IA North Central    3046355    21
## 17                Kansas  KS North Central    2853118    63
## 18              Kentucky  KY         South    4339367   116
## 19             Louisiana  LA         South    4533372   351
## 20                 Maine  ME     Northeast    1328361    11
## 21              Maryland  MD         South    5773552   293
## 22         Massachusetts  MA     Northeast    6547629   118
## 23              Michigan  MI North Central    9883640   413
## 24             Minnesota  MN North Central    5303925    53
## 25           Mississippi  MS         South    2967297   120
## 26              Missouri  MO North Central    5988927   321
## 27               Montana  MT          West     989415    12
## 28              Nebraska  NE North Central    1826341    32
## 29                Nevada  NV          West    2700551    84
## 30         New Hampshire  NH     Northeast    1316470     5
## 31            New Jersey  NJ     Northeast    8791894   246
## 32            New Mexico  NM          West    2059179    67
## 33              New York  NY     Northeast   19378102   517
## 34        North Carolina  NC         South    9535483   286
## 35          North Dakota  ND North Central     672591     4
## 36                  Ohio  OH North Central   11536504   310
## 37              Oklahoma  OK         South    3751351   111
## 38                Oregon  OR          West    3831074    36
## 39          Pennsylvania  PA     Northeast   12702379   457
## 40          Rhode Island  RI     Northeast    1052567    16
## 41        South Carolina  SC         South    4625364   207
## 42          South Dakota  SD North Central     814180     8
## 43             Tennessee  TN         South    6346105   219
## 44                 Texas  TX         South   25145561   805
## 45                  Utah  UT          West    2763885    22
```

```
## 46              Vermont  VT     Northeast    625741     2
## 47             Virginia  VA         South   8001024   250
## 48           Washington  WA          West   6724540    93
## 49        West Virginia  WV         South   1852994    27
## 50            Wisconsin  WI North Central   5686986    97
## 51              Wyoming  WY          West    563626     5
```

(a) Compute the per 100,000 murder rate for each state and store it in an object called `murder_rate`. Then use logical operators to create a logical vector named low that tells us which entries of murder_rate are lower than 1.

```
murder_rate<-(murders$total)/(murders$population)*100000
murder_rate
```

```
##  [1]  2.8244238  2.6751860  3.6295273  3.1893901  3.3741383  1.2924531
##  [7]  2.7139722  4.2319369 16.4527532  3.3980688  3.7903226  0.5145920
## [13]  0.7655102  2.8369608  2.1900730  0.6893484  2.2081106  2.6732010
## [19]  7.7425810  0.8280881  5.0748655  1.8021791  4.1786225  0.9992600
## [25]  4.0440846  5.3598917  1.2128379  1.7521372  3.1104763  0.3798036
## [31]  2.7980319  3.2537239  2.6679599  2.9993237  0.5947151  2.6871225
## [37]  2.9589340  0.9396843  3.5977513  1.5200933  4.4753235  0.9825837
## [43]  3.4509357  3.2013603  0.7959810  0.3196211  3.1246001  1.3829942
## [49]  1.4571013  1.7056487  0.8871131
```

```
low<-murder_rate<1
low
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [13]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE
## [25] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
## [37] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
## [49] FALSE FALSE  TRUE
```

(b) Now use the results from the previous exercise and the function which to determine the indices of murder_rate associated with values lower than 1.

```
which(low)
```

```
##  [1] 12 13 16 20 24 30 35 38 42 45 46 51
```

(c) Use the results from the previous exercise to report the names of the states with murder rates lower than 1.

```
murders$state[low]
```

```
##  [1] "Hawaii"        "Idaho"         "Iowa"          "Maine"
##  [5] "Minnesota"     "New Hampshire" "North Dakota"  "Oregon"
##  [9] "South Dakota"  "Utah"          "Vermont"       "Wyoming"
```

(d) Now extend the code from exercises 2 and 3 to report the states in the Northeast with murder rates lower than 1. Hint: use the previously defined logical vector low and the logical operator &.

```
murders$state[low&(murders$region=="Northeast")]
```

```
## [1] "Maine"         "New Hampshire" "Vermont"
```

(e) In a previous exercise we computed the murder rate for each state and the average of these numbers. How many states are below the average?

```
mean(murder_rate)
```

```
## [1] 2.779125
```

```
murders$state[mean(murder_rate)]
```

```
## [1] "Alaska"
```

(f) Use the match function to identify the states with abbreviations AK, MI, and IA. Hint: start by defining an index of the entries of murders$abb that match the three abbreviations, then use the [operator to extract the states.

```
murders$abb
```

```
##  [1] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "DC" "FL" "GA" "HI" "ID" "IL" "IN"
## [16] "IA" "KS" "KY" "LA" "ME" "MD" "MA" "MI" "MN" "MS" "MO" "MT" "NE" "NV" "NH"
## [31] "NJ" "NM" "NY" "NC" "ND" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT"
## [46] "VT" "VA" "WA" "WV" "WI" "WY"
```

```
match(c("AK","MI","IA"),murders$abb)
```

```
## [1]  2 23 16
```

(g) Use the %in% operator to create a logical vector that answers the question: which of the following are actual abbreviations: MA, ME, MI, MO, MU?

```
murders$abb%in%c("MA","ME","MI","MO","MU")
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## [25] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE FALSE
```

(h) Extend the code you used in exercise 7 to report the one entry that is not an actual abbreviation. Hint: use the ! operator, which turns FALSE into TRUE and vice versa, then which to obtain an index.

```
!murders$abb%in%c("MA","ME","MI","MO","MU")
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE
## [25]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [49]  TRUE  TRUE  TRUE
```

(i) Create a histogram of the state populations.

```
hist(murders$population)
```

(j) Generate boxplots of the state populations by region.

```
boxplot(population~region,data=murders)
```