Lab 1
Krish Agarwal| 21112016|
2BSc DS| R for Analytics

Q1) Assume that we have registered the height and weight for four people: Heights in cm are 180, 165, 160, 193; weights in kg are 87, 58, 65, 100. Make two vectors, height and weight, with the data. The bodymass index (BMI) is defined as

a. weight in kg / (height in m)^2 b.Make a vector with the BMI values for the four people. c.Finally make a vector with the weights for those people who have a BMI larger than 25.

```
#Creating 3 vectors for each - height(in cm), weight(in kgs), height(in cms)
height <- c(180, 165, 160, 193)
weight <- c(87, 58, 65, 100)
heightm <- height/100

#Creating a BMI vector
bmi <- weight/(heightm)^2
bmi
```

```
## [1] 26.85185 21.30395 25.39062 26.84636
```

```
cat("\n")
```

```
#Displaying BMI of people greater than 25 via indexing
bmi[bmi>25]
```

```
## [1] 26.85185 25.39062 26.84636
```

Q2) Create a list of 50 employees with IDs.There are three categories of the employees. The IDs can be used to identify the levels. [Please note there should be atleast 10 employees in each category.] The pattern of the ids are: 200xx - Senior Manager 210xx - Jr Manager 220xx - Team Lead

```
#Creating 3 list of categorized employees
senior_manager <- seq(length = 15, from = 20000)
jr_manager <- seq(length = 15, from = 21000)
team_lead <- seq(length = 20, from = 22000)

#Concatinating the 3 lists into 1
empf <- c(senior_manager, jr_manager, team_lead)
print("Employees:")
```

```
## [1] "Employees:"
```

```
cat("     -----------\n")
```

```
##      -----------
```

```
empf
```

```
##  [1] 20000 20001 20002 20003 20004 20005 20006 20007 20008 20009 20010 20011
## [13] 20012 20013 20014 21000 21001 21002 21003 21004 21005 21006 21007 21008
## [25] 21009 21010 21011 21012 21013 21014 22000 22001 22002 22003 22004 22005
## [37] 22006 22007 22008 22009 22010 22011 22012 22013 22014 22015 22016 22017
## [49] 22018 22019
```

Q3) Create a seperate vector that holds the gender type of these 50 employees.

```
#Setting a seed valule so the value doesnot change very time the chunk is executed
set.seed(155)

#creating a vector holding the values as "M" and "F"
gender <- c("M", "F")

#Creating 50 random M/F values
genderf <- sample(gender, 50, replace = T)
genderf
```

```
##  [1] "F" "F" "F" "M" "F" "M" "F" "F" "F" "F" "F" "M" "F" "F" "F" "M" "F" "M" "M"
## [20] "M" "M" "F" "M" "M" "M" "F" "F" "F" "M" "M" "F" "F" "M" "F" "M" "M" "M" "F"
## [39] "M" "F" "F" "F" "F" "F" "F" "M" "F" "F" "F" "F"
```

```
cat("\n")
```

```
#Assigning the genders to the employees
genderF <- genderf
names(genderF) <- empf
print("Gender of all the employees:")
```

```
## [1] "Gender of all the employees:"
```

```
cat("     --------------------------------\n")
```

```
##      --------------------------------
```

```
genderF
```

```
## 20000 20001 20002 20003 20004 20005 20006 20007 20008 20009 20010 20011 20012
##   "F"   "F"   "F"   "M"   "F"   "M"   "F"   "F"   "F"   "F"   "F"   "M"   "F"
## 20013 20014 21000 21001 21002 21003 21004 21005 21006 21007 21008 21009 21010
##   "F"   "F"   "M"   "F"   "M"   "M"   "M"   "M"   "F"   "M"   "M"   "M"   "F"
## 21011 21012 21013 21014 22000 22001 22002 22003 22004 22005 22006 22007 22008
##   "F"   "F"   "M"   "M"   "F"   "F"   "M"   "F"   "M"   "M"   "M"   "F"   "M"
## 22009 22010 22011 22012 22013 22014 22015 22016 22017 22018 22019
##   "F"   "F"   "F"   "F"   "F"   "F"   "M"   "F"   "F"   "F"   "F"
```

Q4) Group the employees as male or female and count the number of male and female employees. Also,Find the male:female ratio.

```r
#Gropuing of male employees via indexing
print("Male Employees:")
```

```
## [1] "Male Employees:"
```

```r
cat("    --------------------\n")
```

```
##     --------------------
```

```r
gender_male <- genderF[genderF=="M"]
gender_male
```

```
## 20003 20005 20011 21000 21002 21003 21004 21005 21007 21008 21009 21013 21014
##   "M"   "M"   "M"   "M"   "M"   "M"   "M"   "M"   "M"   "M"   "M"   "M"   "M"
## 22002 22004 22005 22006 22008 22015
##   "M"   "M"   "M"   "M"   "M"   "M"
```

```r
cat("\n")
```

```r
#Grouping of female employees via indexing
print("Female Employees:")
```

```
## [1] "Female Employees:"
```

```r
cat("    --------------------\n")
```

```
##     --------------------
```

```r
gender_female <- genderF[genderF=="F"]
gender_female
```

```
## 20000 20001 20002 20004 20006 20007 20008 20009 20010 20012 20013 20014 21001
##   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"
## 21006 21010 21011 21012 22000 22001 22003 22007 22009 22010 22011 22012 22013
##   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"   "F"
## 22014 22016 22017 22018 22019
##   "F"   "F"   "F"   "F"   "F"
```

Q5) Create a vector that holds the value of salary of each employees.

```r
#Setting a seed value so the value does not change very time the chunk is executed
set.seed(155)

#Creating 50 random salary values
salary <- c(40000:100000)
salaryf <- sample(salary, 50, replace = T)
salaryf
```

```
##  [1] 68429 89720 61552 76595 74713 52198 70463 94158 75238 52884 69937 54500
## [13] 69591 53729 91764 68045 97957 55598 91333 73885 68553 75117 82230 98949
## [25] 40233 47955 99814 99234 46736 50783 81859 78188 59922 62068 77466 55797
## [37] 68186 97687 68346 41464 80195 66074 67083 90463 55593 76482 58741 98084
## [49] 83615 97928
```

```
cat("\n")
```

```
#Assigning the salaries to the employees
salaryF <- salaryf
names(salaryF) <- empf
print("Salaries of Employees:")
```

```
## [1] "Salaries of Employees:"
```

```
cat("     ----------------------\n")
```

```
##      ----------------------
```

```
salaryF
```

```
## 20000 20001 20002 20003 20004 20005 20006 20007 20008 20009 20010 20011 20012
## 68429 89720 61552 76595 74713 52198 70463 94158 75238 52884 69937 54500 69591
## 20013 20014 21000 21001 21002 21003 21004 21005 21006 21007 21008 21009 21010
## 53729 91764 68045 97957 55598 91333 73885 68553 75117 82230 98949 40233 47955
## 21011 21012 21013 21014 22000 22001 22002 22003 22004 22005 22006 22007 22008
## 99814 99234 46736 50783 81859 78188 59922 62068 77466 55797 68186 97687 68346
## 22009 22010 22011 22012 22013 22014 22015 22016 22017 22018 22019
## 41464 80195 66074 67083 90463 55593 76482 58741 98084 83615 97928
```

Q6) Find the average pay of employees gender wise.

```
avg_pay <- tapply(salaryf, genderf, mean)
print("Average Pay of employees gender-wise:")
```

```
## [1] "Average Pay of employees gender-wise:"
```

```
cat("    ---------------------------------------\n")
```

```
##     ---------------------------------------
```

```
avg_pay
```

```
##        F        M
## 75848.29 66623.00
```

Q7) The employees were rated out of 10 for their service by the company. The review was taken from the peers. Create a vector that store the ratings of this review.

```r
#Setting a seed valule so the value doesnot change very time the chunk is executed
set.seed(200)

#Creating 50 random rating values
ratings <- c(1:10)
ratingsf <- sample(ratings, 50, replace = T)
ratingsf
```

```
##  [1]  6  2  8  7  5 10  2  6  8  8  4  6  8  4  6  6  7  6  3  3  5  1  6  8  7
## [26]  8  2  9  6  4  6  4  3  4  3  5  8  5  6  2  6  8  6  9  1  5  6  3  9  4
```

```r
cat("\n")
```

```r
#Assigning the ratings to the employees
ratingsF <- ratingsf
names(ratingsF) <- empf
print("Employee Ratings:")
```

```
## [1] "Employee Ratings:"
```

```r
cat("     ------------------\n")
```

```
##      ------------------
```

```r
ratingsF
```

```
## 20000 20001 20002 20003 20004 20005 20006 20007 20008 20009 20010 20011 20012
##     6     2     8     7     5    10     2     6     8     8     4     6     8
## 20013 20014 21000 21001 21002 21003 21004 21005 21006 21007 21008 21009 21010
##     4     6     6     7     6     3     3     5     1     6     8     7     8
## 21011 21012 21013 21014 22000 22001 22002 22003 22004 22005 22006 22007 22008
##     2     9     6     4     6     4     3     4     3     5     8     5     6
## 22009 22010 22011 22012 22013 22014 22015 22016 22017 22018 22019
##     2     6     8     6     9     1     5     6     3     9     4
```

Q8) As the employees 21010,20012,22008,21004,20007,22010 were new, the ratings for these employees were kept as NA.

```r
#Converting the ratings of the new employees to NA
ratingsF[c(26, 13, 39, 20, 8, 41)] <- NA

print("New Employee Ratings:")
```

```
## [1] "New Employee Ratings:"
```

```r
cat("     ------------------------\n")
```

```
##      ------------------------
```

```
ratingsF
```

```
## 20000 20001 20002 20003 20004 20005 20006 20007 20008 20009 20010 20011 20012
##     6     2     8     7     5    10     2    NA     8     8     4     6    NA
## 20013 20014 21000 21001 21002 21003 21004 21005 21006 21007 21008 21009 21010
##     4     6     6     7     6     3    NA     5     1     6     8     7    NA
## 21011 21012 21013 21014 22000 22001 22002 22003 22004 22005 22006 22007 22008
##     2     9     6     4     6     4     3     4     3     5     8     5    NA
## 22009 22010 22011 22012 22013 22014 22015 22016 22017 22018 22019
##     2    NA     8     6     9     1     5     6     3     9     4
```

Q9) Count the number of employees who were eligible for the ratings.

```
#Complementing the is.na function as to returning all the FALSE value which implies the ratings
print("Employees eligible for ratings:")
```

```
## [1] "Employees eligible for ratings:"
```

```
cat("     ------------------------------\n")
```

```
##      ------------------------------
```

```
#Non-NA values assigned to a variable
non_na <- ratingsF[!is.na(ratingsF)]
length(non_na)
```

```
## [1] 44
```

Q10) Count the number of empoyees who got ratings above 7.0 and less than 5.0

```
#Ratings above 7
print("Number of employees with rating above 7:")
```

```
## [1] "Number of employees with rating above 7:"
```

```
cat("     ----------------------------------------\n")
```

```
##      ----------------------------------------
```

```
above_seven <- non_na[non_na>7]
length(above_seven)
```

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

```
cat("\n\n")
```

```r
#Ratings below 5
print("Number of employees with rating below 5:")
```

```
## [1] "Number of employees with rating below 5:"
```

```r
cat("     ----------------------------------------\n")
```

```
##      ----------------------------------------
```

```r
below_five <- non_na[non_na<5]
length(below_five)
```

```
## [1] 16
```