Homework 03

Victor Benito Garcia Rocha

7/14/2022

Context

We are going to work with a "Mental Health in Tech Survey" dataset, a series of 2016 survey responses aiming to capture mental health statuses and workplace attitudes among tech employees.

```
library(tidyverse) #for data manipulation
library(stringr) #string characters
```

Step 1

Import the information in the file "mental_health_in_tech_2016.csv" into a data frame. Review your dataset

```
mental_health_in_tech <- read_csv("mental_health_in_tech_2016-1.csv")
```

```
## Rows: 1146 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Employees, Health_benefits, Gender, Country_birth, Country_work, Wo...
## dbl (5): Observation, Self_employed, Tech_company, IT_role, Age
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
##  [1] "Observation"     "Self_employed"   "Employees"       "Tech_company"
##  [5] "IT_role"         "Health_benefits" "Age"             "Gender"
##  [9] "Country_birth"   "Country_work"    "Work_remotely"
```

```
## # A tibble: 6 x 11
##   Observation Self_employed Employees Tech_company IT_role Health_benefits   Age
##         <dbl>         <dbl> <chr>            <dbl>   <dbl> <chr>           <dbl>
## 1           1             0 26-100               1       1 Not eligible f~    39
## 2           2             0 jun-25               1       1 No                 29
## 3           3             0 jun-25               1       1 No                 38
## 4           4             0 jun-25              NA       1 Yes                43
## 5           5             0 More tha~            1       1 Yes                42
## 6           6             0 26-100               1       1 I don't know       30
## # i 4 more variables: Gender <chr>, Country_birth <chr>, Country_work <chr>,
## #   Work_remotely <chr>
```

Step 2

Check if there are string inconsistencies (typos, capitalization errors, etc), and correct them. Hint: Focus on the Employees and Gender variables

```
#Employees variable (how many employees work in the company)
unique(mental_health_in_tech$Employees)
```

```
## [1] "26-100"         "jun-25"         "More than 1000" "100-500"
## [5] "500-1000"       "01-may"
```

```
mental_health_in_tech$Employees <- mental_health_in_tech$Employees %>%
  str_replace("01-may", "1-5") %>%
  str_replace("jun-25", "6-25")
unique(mental_health_in_tech$Employees)
```

```
## [1] "26-100"         "6-25"           "More than 1000" "100-500"
## [5] "500-1000"       "1-5"
```

```
#Gender variable (Gender of the employee)
unique(mental_health_in_tech$Gender)
```

```
##  [1] "Male"   "male"   "Female" "M"      "female" "m"      "f"      "MALE"
##  [9] "F"      NA
```

```
male_labels <- c("Male", "male", "M", "m", "MALE")
female_labels <- c("Female", "female", "F", "f")

mental_health_in_tech <- mental_health_in_tech %>%
  mutate(Gender = case_when(
    Gender %in% male_labels ~ "Male",
    Gender %in% female_labels ~ "Female",
    TRUE ~ NA_character_
))
```

```
unique(mental_health_in_tech$Gender)
```

```
## [1] "Male"   "Female" NA
```

Step 3

Are ther some missing values? Replace them (you can use the mean, median, mode or other value) In any case, you have to justify your decision.

```
# Print the columns that have NAs
print(colnames(mental_health_in_tech)[colSums(is.na(mental_health_in_tech)) > 0])
```

```
## [1] "Tech_company" "IT_role"      "Gender"
```

- Tech_company variable has values like 1 and 0, it's not really possible to infer it precisely neither make an average/mode since it's a binary variable. Since this is a survey, we can assume that the missing values are from people who don't work in a tech company (I assume that 0 means "No").

```
unique(mental_health_in_tech$Tech_company)
```

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

```
mental_health_in_tech$Tech_company <- mental_health_in_tech$Tech_company %>%
  replace_na(0)
```

Now the problem is that it's numeric, but it should be a factor.

```
class(mental_health_in_tech$Tech_company)
```

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

```
mental_health_in_tech$Tech_company <- as.factor(mental_health_in_tech$Tech_company)
class(mental_health_in_tech$Tech_company)
```

```
## [1] "factor"
```

- Let's print the columns Tech_company and IT_Role when the IT_Role is NA. We can see that when the IT_Role is NA, the Tech_company is 0, so it's safe to assume that the intended value is "No" (0). We also change the data type to factor.

```
unique(mental_health_in_tech$IT_role)
```

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

```
na <- mental_health_in_tech %>%
  filter(is.na(IT_role)) %>%
  select(Tech_company, IT_role)
```

```
mental_health_in_tech$IT_role <- mental_health_in_tech$IT_role %>%
  replace_na(0)
```

```
class(mental_health_in_tech$IT_role)
```

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

```
mental_health_in_tech$IT_role <- as.factor(mental_health_in_tech$IT_role)
class(mental_health_in_tech$IT_role)
```

```
## [1] "factor"
```

- There's no reliable way to infer the Gender of the person. Since data integrity is fundamental and it afffects only one row, we can drop it. We should also change the data type to factor.

```r
mental_health_in_tech %>%
  filter(is.na(Gender))
```

```
## # A tibble: 1 x 11
##   Observation Self_employed Employees Tech_company IT_role Health_benefits   Age
##         <dbl>         <dbl> <chr>     <fct>        <fct>   <chr>           <dbl>
## 1        1047             0 100-500   0            0       Yes                26
## # i 4 more variables: Gender <chr>, Country_birth <chr>, Country_work <chr>,
## #   Work_remotely <chr>
```

```r
mental_health_in_tech <- mental_health_in_tech %>%
  drop_na()
```

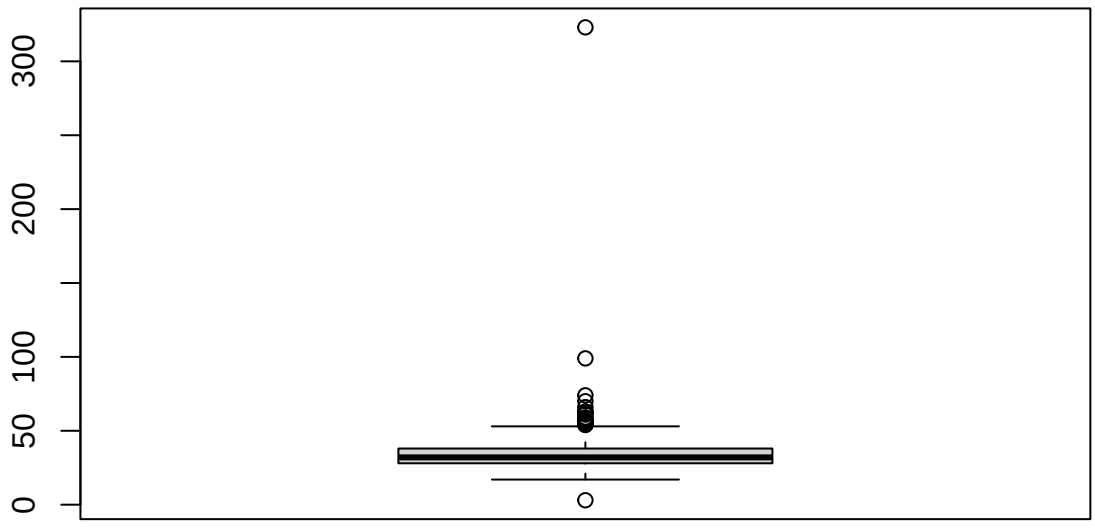
```r
mental_health_in_tech$Gender <- as.factor(mental_health_in_tech$Gender)
class(mental_health_in_tech$Gender)
```

```
## [1] "factor"
```

Step 4

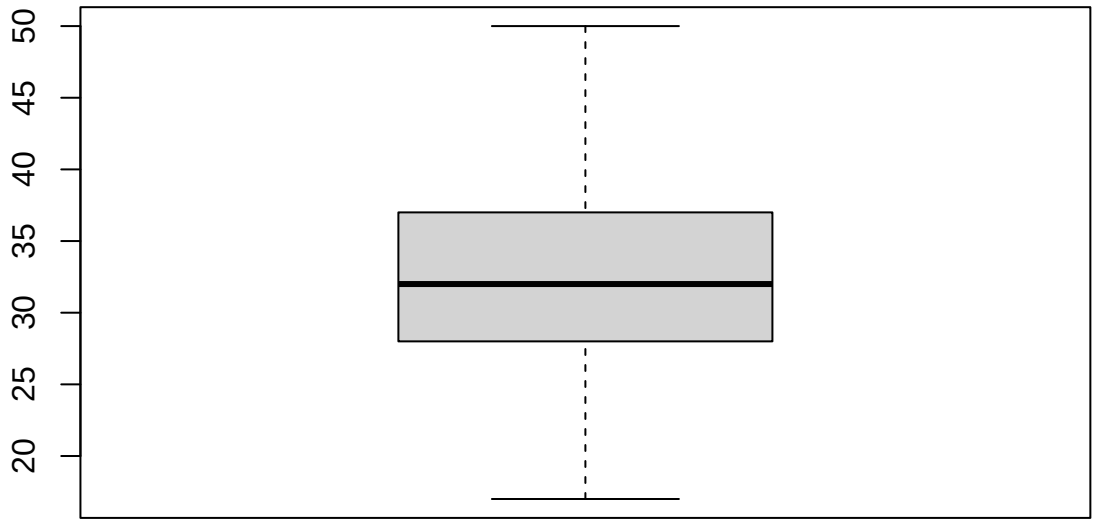
Look for outliers in the Age variable and remove them.

```r
boxplot(mental_health_in_tech$Age)
```



```r
Q1 <- quantile(mental_health_in_tech$Age, 0.25)
Q3 <- quantile(mental_health_in_tech$Age, 0.75)
IQR <- Q3 - Q1

lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

mental_health_in_tech <- mental_health_in_tech %>%
  filter(Age >= lower_bound & Age <= upper_bound)

# There are still 3 outliers left when recalculated.
# Since the instruction is to remove them, another iteration is needed.

Q1 <- quantile(mental_health_in_tech$Age, 0.25)
Q3 <- quantile(mental_health_in_tech$Age, 0.75)
IQR <- Q3 - Q1

lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

mental_health_in_tech <- mental_health_in_tech %>%
  filter(Age >= lower_bound & Age <= upper_bound)
```

```r
boxplot(mental_health_in_tech$Age)
```



Step 5

Review if the data type assigned to each variable is adecuate. If not, change it.(Numbers must be integers or doubles, some characters must be factors, etc)

- Observation: Numeric, it's an ID, so it should be character.

```
class(mental_health_in_tech$Observation)
```

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

```
mental_health_in_tech$Observation <- as.character(mental_health_in_tech$Observation)
class(mental_health_in_tech$Observation)
```

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

- Self_employed: Numeric, it's a binary variable, so it should be a factor.

```
class(mental_health_in_tech$Self_employed)
```

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

```
mental_health_in_tech$Self_employed <- as.factor(mental_health_in_tech$Self_employed)
class(mental_health_in_tech$Self_employed)
```

```
## [1] "factor"
```

- Employees: Character, it's a range of employees, so it should be a factor.

```
class(mental_health_in_tech$Employees)
```

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

```
mental_health_in_tech$Employees <- as.factor(mental_health_in_tech$Employees)
class(mental_health_in_tech$Employees)
```

```
## [1] "factor"
```

- Tech_company: We already changed it to a factor.

```
class(mental_health_in_tech$Tech_company)
```

```
## [1] "factor"
```

- IT_role: We already changed it to a factor.

```r
class(mental_health_in_tech$IT_role)
```

```
## [1] "factor"
```

- Health_benefits: Numeric, its a list so it should be a factor.

```r
class(mental_health_in_tech$Health_benefits)
```

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

```r
mental_health_in_tech$Health_benefits <- as.factor(
  mental_health_in_tech$Health_benefits
)
class(mental_health_in_tech$Health_benefits)
```

```
## [1] "factor"
```

- Age: Numeric, which it's okay but we can change it to integer.

```r
class(mental_health_in_tech$Age)
```

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

```r
mental_health_in_tech$Age <- as.integer(mental_health_in_tech$Age)
class(mental_health_in_tech$Age)
```

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

- Gender: We already changed it to a factor.

```r
class(mental_health_in_tech$Gender)
```

```
## [1] "factor"
```

- Country_birth: Character, it's a country, so it should be a factor.

```r
class(mental_health_in_tech$Country_birth)
```

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

```r
mental_health_in_tech$Country_birth <- as.factor(mental_health_in_tech$Country_birth)
class(mental_health_in_tech$Country_birth)
```

```
## [1] "factor"
```

- Work_remotely: Character, it's a list so it should be a factor.

```r
class(mental_health_in_tech$Work_remotely)
```

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

```r
mental_health_in_tech$Work_remotely <- as.factor(mental_health_in_tech$Work_remotely)
class(mental_health_in_tech$Work_remotely)
```

```
## [1] "factor"
```