Statistics with R - Exercise 3

Philipp Satlawa - h0640348

30/01/2021

This document contains the answered questions of exercise 3 for the course “Statistics with R”.

Task 1 – Statistical Tests

our sample of size n = 18

```
x <- c(870, 930, 932, 935, 938, 1045, 1050, 1052, 1055,
       970, 980, 1001, 1009, 1027, 1030, 1032, 1040, 1046)
```

Task 2 - Functions

1. Create a function with the name quad_equ()

```
quad_equ <- function(coef){
  x1 <- -(coef[1]/2) - sqrt((coef[1]/2)^2-coef[2])
  x2 <- -(coef[1]/2) + sqrt((coef[1]/2)^2-coef[2])

  return(c(x1,x2))
}
coef <- c(3,-4)
quad_equ(coef)
```

```
## [1] -4  1
```

2. Extend function - check if the passed argument coef gives real solutions

```
quad_equ <- function(coef){
  if((coef[1]/2)^2-coef[2] >= 0){
    x1 <- -(coef[1]/2) - sqrt((coef[1]/2)^2-coef[2])
    x2 <- -(coef[1]/2) + sqrt((coef[1]/2)^2-coef[2])

    return(c(x1,x2))
```

```
  } else{
    print("The supplied vector x has no real solution.")
    #stop("The supplied vector x has no real solution.")
  }
}
coef <- c(3,-4)
quad_equ(coef)
```

```
## [1] -4  1
```

```
coef <- c(-2, 2)
quad_equ(coef)
```

```
## [1] "The supplied vector x has no real solution."
```

3. Extend function - check if the passed argument is valid

```
quad_equ <- function(coef){
  if(length(coef)==2){
    if((coef[1]/2)^2-coef[2] >= 0){
      x1 <- -(coef[1]/2) - sqrt((coef[1]/2)^2-coef[2])
      x2 <- -(coef[1]/2) + sqrt((coef[1]/2)^2-coef[2])

      return(c(x1,x2))
    } else{
      print("The supplied vector x has no real solution.")
    }
  } else{
    print("The supplied vector x has the wrong length.
         The input vector has to have length 2.
         Please enter a valid vector.")
  }
}
# test if the function is working correctly with a correct input
coef <- c(3,-4)
quad_equ(coef)
```

```
## [1] -4  1
```

```
# test if the function invokes an error with an incorrect input
coef <- c(3,-4, 9)
quad_equ(coef)
```

```
## [1] "The supplied vector x has the wrong length. \n         The input vector has to have length 2. \n
```

Task 3 - Graphics

Creation of a statistical graph

Load libraries

```
library(ggplot2)
library(data.table)
```

1. Create a data for the construction of a plot that mimics the template "Plot_9.png"

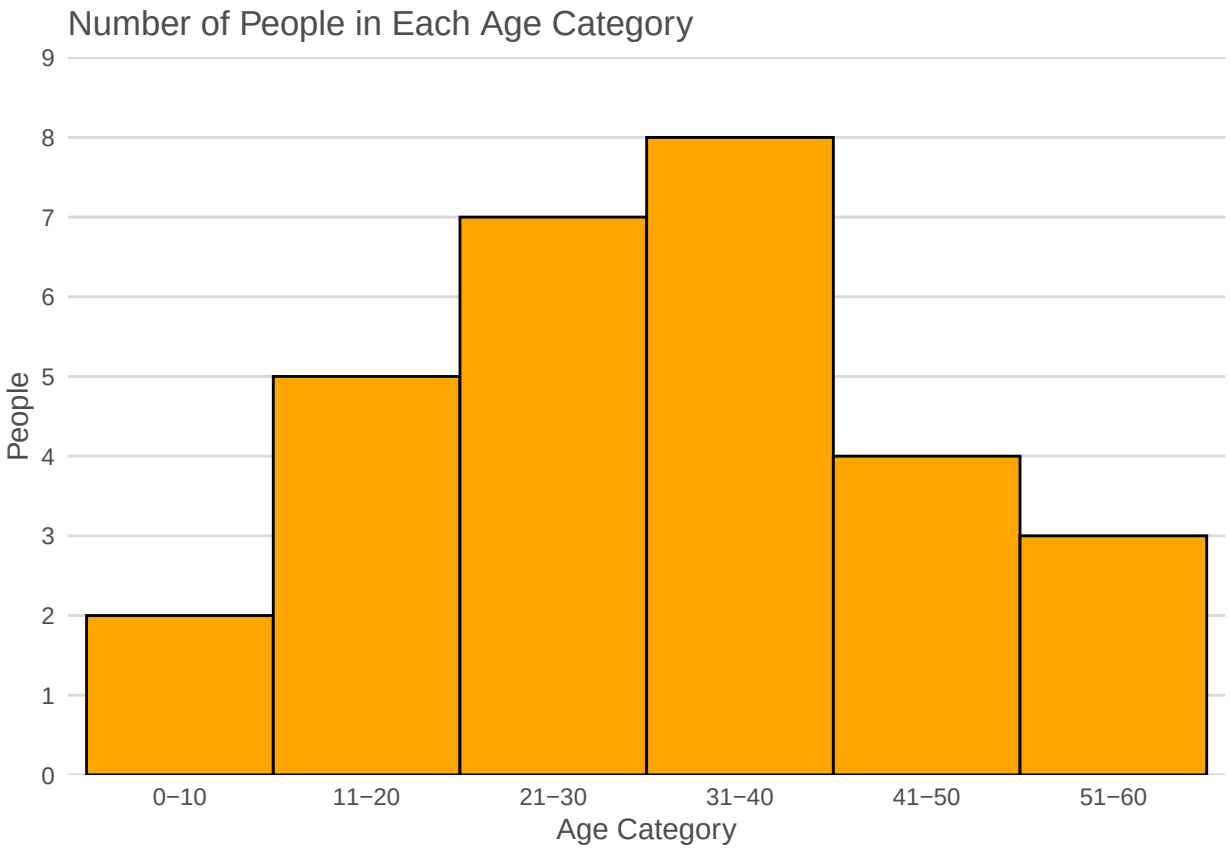
```
# create data
dt <- data.table(id = c(1:29), age = c(25, 21, 5, 15, 47, 33, 39, 56, 3, 45, 31, 28, 44, 15, 13, 22, 40
```

2. combine the feature "age" into age intervals

```
# create vector containing the cutting points
cut_points <- c(0,11,21,31,41,51,61)
# create vector containing the labels
labels <- c("0-10","11-20", "21-30", "31-40", "41-50", "51-60")
# categorize the feature "age" into the created labels
age_cat <- cut(dt$age, breaks=cut_points, include.lowest=TRUE, right=FALSE, labels=labels)
# add age_cat to data.table
dt[, ("age_category") := age_cat]
```

3. Create plot

```
# Create plot
p <- ggplot(data = dt, mapping = aes(x=age_category)) +
  geom_bar(stat="count", fill="orange", color="black", width = 1) +
  labs(title = "Number of People in Each Age Category", x = "Age Category", y = "People") +
  theme(panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(colour = "grey85"),
        panel.grid.major.x = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_text(colour = "grey30"),
        plot.title = element_text(colour = "grey30")) +
  scale_y_continuous(breaks = seq(0, 9, 1), expand=c(0,0), limits=c(0,9))
p
```


Number of People in Each Age Category
People
0
1
2
3
4
5
6
7
8
9
0−10
11−20
21−30
31−40
41−50
51−60
Age Category