Session 5 - Function Writing

Alex Mounsey

21/11/2020

Exercise 1

The aim of this exercise is to write a function to compute summary statistics that deal with missing values. Real data often contains missing values, and dealing with them requires special care. In R, missing values are denoted as `NA`.

Here is a very simple dataset which contains three missing values:

```
x <- c(2, 6, NA, 5, 2, 1, NA, 6, 6, 7, NA, 4, 0)
```

You can confirm that there are three `NA` values:

```
sum(is.na(x))
```

```
## [1] 3
```

The number of non-missing values can be found, too:

```
sum(!is.na(x))
```

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

To compute the sample mean of the non-missing values, we need to tell the `mean()` function to remove the `NA` values:

```
mean(x, na.rm = T)
```

```
## [1] 3.9
```

Task 1

Write a function called `mean_na()` which automatically removes missing values. Use two comment lines at the beginning of your function to record the input and output of the function. Illustrate the use of your function:

```r
mean_na <- function(x) {
  #' Compute the mean of non-missing values from a given vector
  #'
  #' @param x (vector) The data to calculate the mean of
  #' @return (int) The mean of non-missing values

  return(mean(x, na.rm = T))
}

mean_na(x)
```

```
## [1] 3.9
```

Task 2

Write a function called `statistics_na()` which returns the minimum, mean, standard deviation, and maximum of the input data. Give these quantities appropriate names. Use comment lines to record the input and output of your function and illustrate the use of your function:

```r
statistics_na <- function(x) {
  #' Compute summary statistics of non-missing values from a given vector
  #'
  #' @param x (vector) The data to calculate summary statistics of
  #' @return (list) Summary statistics of non-missing values

  return(list(
    min = min(x, na.rm = T),
    mean = mean(x, na.rm = T),
    std = sd(x, na.rm = T),
    max = max(x, na.rm = T)
  ))
}

statistics_na(x)
```

```
## $min
## [1] 0
##
## $mean
## [1] 3.9
##
## $std
## [1] 2.469818
##
## $max
## [1] 7
```

An individual value can be extracted as follows:

```r
statistics_na(x)$mean
```

```
## [1] 3.9
```

Task 3

Modify your function to also return the number of `NA` values:

```
statistics_na <- function(x) {
  #' Compute summary statistics of non-missing values from a given vector
  #'
  #' @param x (vector) The data to calculate summary statistics of
  #' @return (list) Summary statistics of non-missing values

  return(list(
    min = min(x, na.rm = T),
    mean = mean(x, na.rm = T),
    std = sd(x, na.rm = T),
    max = max(x, na.rm = T),
    na = sum(is.na(x))
  ))
}

statistics_na(x)
```

```
## $min
## [1] 0
##
## $mean
## [1] 3.9
##
## $std
## [1] 2.469818
##
## $max
## [1] 7
##
## $na
## [1] 3
```

Task 4

Include `na.rm` as an argument of your function, set to `TRUE` by default. Modify the calculations that you perform in your function so that they make use of the `na.rm` argument. Update the comment lines accordingly:

```
statistics_na <- function(x, na_rm = T) {
  #' Compute summary statistics of values from a given vector
  #'
  #' @param x (vector) The data to calculate summary statistics of
  #' @param na_rm (bool) Whether to include missing values in the calculations (default = TRUE)
  #' @return (list) Summary statistics of non-missing values

  return(list(
    min = min(x, na.rm = na_rm),
    mean = mean(x, na.rm = na_rm),
    std = sd(x, na.rm = na_rm),
    max = max(x, na.rm = na_rm),
    na = sum(is.na(x))
  ))
}

statistics_na(x, na_rm = F)
```

```
## $min
## [1] NA
##
## $mean
## [1] NA
##
## $std
## [1] NA
##
## $max
## [1] NA
##
## $na
## [1] 3
```

Exercise 2

The aim of this exercise is to write a function which produces a histogram using `ggplot2` and another function that computes a set of summary statistics using `dplyr`.

Task 1

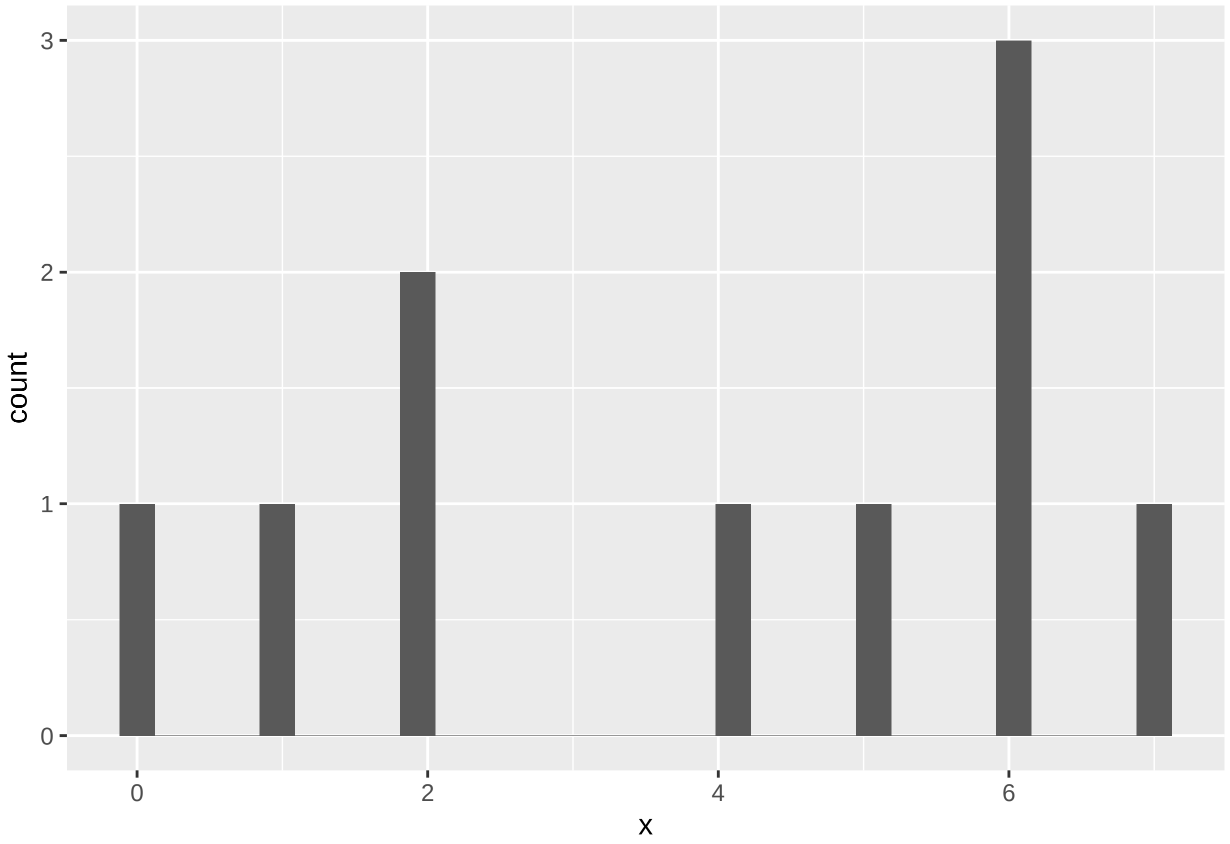
Write a function called `display_data()` which takes a numeric data vector as input and produces a `ggplot2` histogram of that data. Record the input and ouput of the function with comment lines:

```
display_data <- function(x) {
  #' Produces a histogram from a given vector
  #'
  #' @param x (vector) The data to produce a histogram from
  #' @return (plot) Histogram of the provided data

  df <- data.frame(x = x)  # Convert input data to a dataframe

  return(ggplot(data = df, aes(x = x)) +
      geom_histogram())
}

display_data(x)
```



Task 2

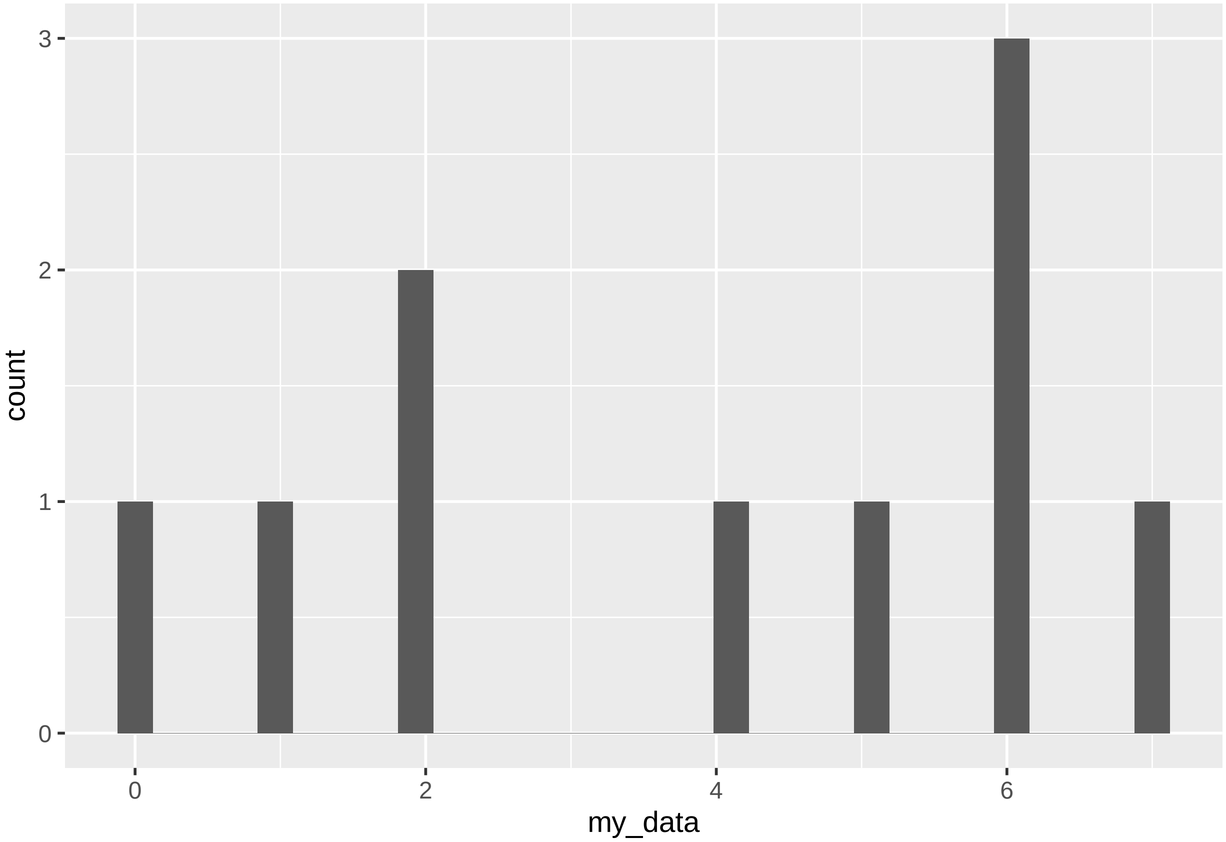
Modify your function so that you also pass an argument, `x_lab` for the x-axis. Set to "my_data" by default.

```r
display_data <- function(x, x_lab = "my_data") {
  #' Produces a histogram from a given vector
  #'
  #' @param x (vector) The data to produce a histogram from
  #' @param x_lab (str) The x-axis label
  #' @return (plot) Histogram of the provided data

  df <- data.frame(x = x)

  return(ggplot(data = df, aes(x = x)) +
      geom_histogram() +
      labs(x = x_lab))
}

display_data(x)
```



Task 3

Write a function, `summarise_data()`, which uses `dplyr` to compute and return the sample minimum, mean, standard deviation, and maximum of the provided data, together with a count of `NA` values. Your function should handle missing values in an appropriate way. Ensure that your function is suitably commented:

```r
summarise_data <- function(x, na_rm = T) {
  #' Produces summary statistics of values from a provided vector
  #'
  #' @param x (vector) The data to produce summary statistics of
  #' @param na_rm (bool) Include NA values in calculations (default = TRUE)
  #' @return (dataframe) Summary statistics of the provudied data

  df <- data.frame(x = x)

  return(summarise(df,
                   min = min(x, na.rm = na_rm),
                   mean = mean(x, na.rm = na_rm),
                   std = sd(x, na.rm = na_rm),
                   max = max(x, na.rm = na_rm),
                   na = sum(is.na(x))))
}

summarise_data(x, F)
```

```
##   min mean std max na
## 1  NA   NA  NA  NA  3
```

Task 4

Modify your function to return the sample size and the standard error. The standard error is the standard deviation divided by the square root of the sample size. It is a measure of the error or unreliability of the sample mean:

```
summarise_data <- function(x, na.rm = TRUE){
  #' Produces summary statistics of values from a provided vector
  #'
  #' @param x (vector) The data to produce summary statistics of
  #' @param na_rm (bool) Include NA values in calculations (default = TRUE)
  #' @return (dataframe) Summary statistics of the provided data

  df <- data.frame(x = x)

  return(
    summarise(df,
              min = min(x, na.rm = na.rm),
              avg = mean(x, na.rm = na.rm),
              std = sd(x, na.rm = na.rm),
              max = max(x, na.rm = na.rm),
              NAs = sum(is.na(x)),
              ste = sd(x, na.rm = na.rm) / sqrt(sum(!is.na(x))),
              size = sum(!is.na(x)))
  )
}

summarise_data(x, FALSE)
```

```
##   min avg std max NAs ste size
## 1  NA  NA  NA  NA   3  NA   10
```