p8105_hw1_kx2224

Kangyu Xu (kx2224)

2024-09-18

Question 1

Load the data

```
data("penguins", package = "palmerpenguins")
```

Short Description

The `penguins` dataset contains information about different species of penguins, including variables such as `species`, `island`, `bill_length_mm`, `bill_depth_mm`, `flipper_length_mm`, `body_mass_g`, `sex` and `year`. The dataset consists of 344 rows and 8 columns. The mean flipper length of the penguins is 200.9152047 mm.

Make a scatterplot

```
library(ggplot2)
firstScatterPlot = ggplot(data = penguins, aes(x = bill_length_mm, y = flipper_length_mm, colour = speci
  geom_point(na.rm = TRUE)+
  labs(title = "Scatter Plot of Flipper_length_mm (y) vs Bill_length_mm (x)")
```

Save the plot

```
ggsave("Scatter Plot of Flipper_length_mm (y) vs Bill_length_mm (x).jpg", plot = firstScatterPlot)
```

```
## Saving 6.5 x 4.5 in image
```

Question 2

Create the dataframe

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
set.seed(42)
randomSample = rnorm(10)
logicVector = randomSample>0
charVector = sample(letters, 10, replace = TRUE)
factorVector = sample(c("level1","level2","level3"), 10, replace = TRUE)
df = data.frame(randomSample, logicVector, charVector, factorVector)
df
```

```
##    randomSample logicVector charVector factorVector
## 1    1.37095845        TRUE          o       level3
## 2   -0.56469817       FALSE          c       level2
## 3    0.36312841        TRUE          i       level1
## 4    0.63286260        TRUE          y       level2
## 5    0.40426832        TRUE          d       level2
## 6   -0.10612452       FALSE          e       level3
## 7    1.51152200        TRUE          m       level3
## 8   -0.09465904       FALSE          e       level2
## 9    2.01842371        TRUE          t       level2
## 10  -0.06271410       FALSE          b       level2
```

Calculate the mean of variables

```
mean_randomSample = mean(df %>% pull(randomSample))
mean_logicVector = mean(df %>% pull(logicVector))
mean_charVector = try(mean(df %>% pull(charVector)), silent = TRUE)
```

```
## Warning in mean.default(df %>% pull(charVector)): argument is not numeric or
## logical: returning NA
```

```
mean_factorVector = try(mean(df %>% pull(factorVector)), silent = TRUE)
```

```
## Warning in mean.default(df %>% pull(factorVector)): argument is not numeric or
## logical: returning NA
```

```
# output the result
print(mean_randomSample)
```

```
## [1] 0.5472968
```

```r
print(mean_logicVector)
```

```
## [1] 0.6
```

```r
print(mean_charVector)
```

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

```r
print(mean_factorVector)
```

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

In conclusion, we can take the mean of "numeric" and "logical vector". But for "character" and "factor", the R cannot calculate the mean of them.

Convert variables

```r
numeric_logical = as.numeric(logicVector)
numeric_character = try(as.numeric(charVector), silent = TRUE)
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): NAs introduced
## by coercion
```

```r
numeric_factor = try(as.numeric(factorVector), silent = TRUE)
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): NAs introduced
## by coercion
```

```r
# output the result
print(numeric_logical)
```

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

```r
print(numeric_character)
```

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

```r
print(numeric_factor)
```

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

The result of conversion is that we succeeded to convert logical vector, but we failed to convert both factors and charactors. The reason is that logical values are coerced to 1 (TRUE) and 0 (FALSE), so this conversion works. But for characters and factors, they cannot be directly converted to numerics, so the result will be NA as showed above.

This result can help explain why we could only take the mean of logical vectors.