p8105_hw1_jm5509

Echo

2022-09-19

Problem 1

This a short description of the penguins dataset. The function of str() and summary() illustrate the names and values of important variables.

```
data('penguins',package='palmerpenguins')
str(penguins)
```

```
## tibble [344 x 8] (S3: tbl_df/tbl/data.frame)
##  $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
##  $ bill_depth_mm    : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
##  $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
##  $ body_mass_g      : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
##  $ sex              : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...
##  $ year             : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

```
# Shows the length of the tibble, numeric variables and int varibles; and the levels of the factor vari
summary(penguins)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

```
# Shows the the level number of the factor variables, and basic statistical values of numeric variables

nrow(penguins)
```

```
## [1] 344
```

```
ncol(penguins)
```

```
## [1] 8
```

```
mean(penguins$flipper_length_mm)
```

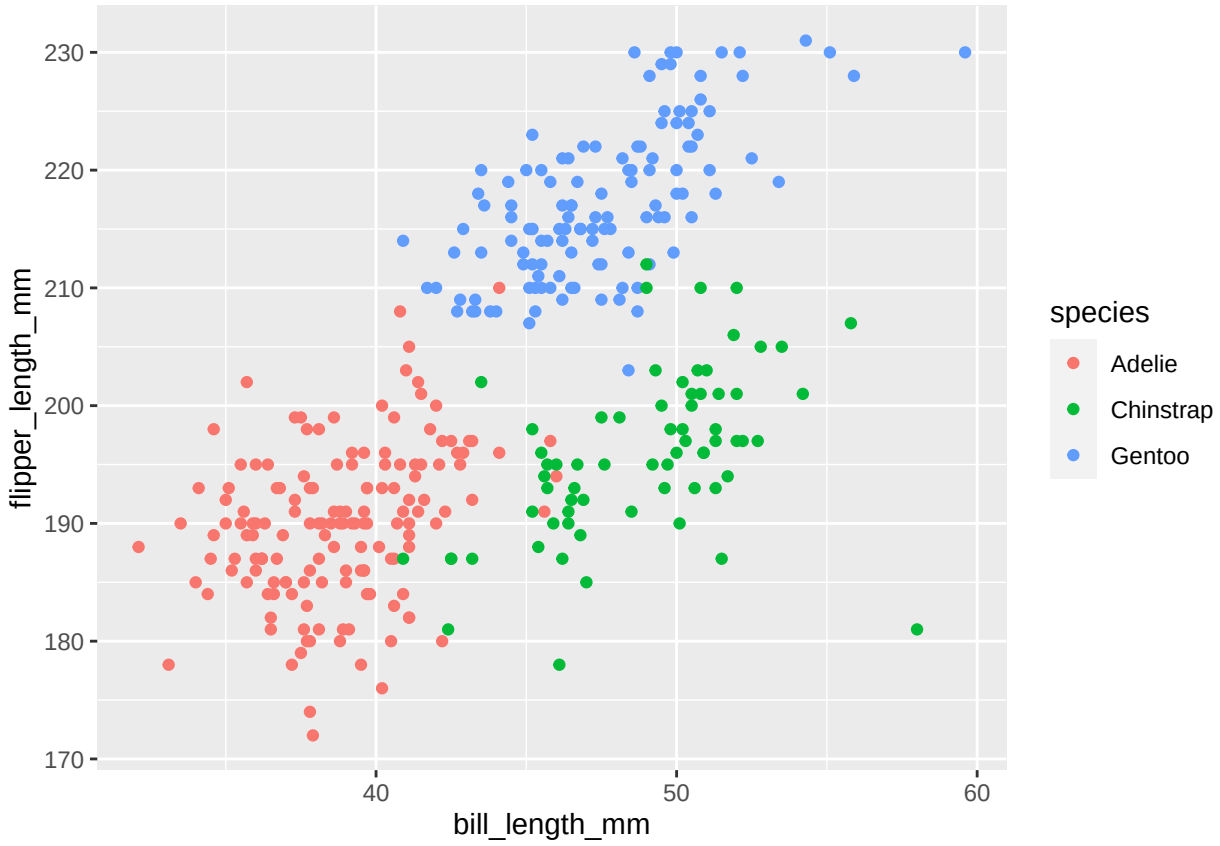
```
## [1] NA
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
ggplot(penguins,aes(x=bill_length_mm,y=flipper_length_mm, color=species))+geom_point()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
ggsave('penguins.pdf')
```

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

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

Problem 2

This solution is displayed as follows. In this case, we create a dataframe comprised of 4 variables with different data types. Among them, only the numeric and the logical value could be taken the mean value.

```
library(tidyverse)
df =
tibble(
norm=rnorm(n=10),
logical=norm>0,
character=c('a','b','c','d','e','f','g','h','i','j'),
factor=as.factor(c(rep('paper',3),rep('scissors',4),rep('rock',3)))
)

mean(df %>% pull(1))
```

```
## [1] -0.304996
```

```
mean(df %>% pull(2))
```

```
## [1] 0.5
```

```
mean(df %>% pull(3))
```

```
## Warning in mean.default(df %>% pull(3)): argument is not numeric or logical:
## returning NA
```

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

```
mean(df %>% pull(4))
```

```
## Warning in mean.default(df %>% pull(4)): argument is not numeric or logical:
## returning NA
```

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

We further convert three other variables to numeric ones. It turns out that only logical and factor vectors could be converted.

```
as.numeric(df %>% pull(2)) # logical
as.numeric(df %>% pull(3)) # character
as.numeric(df %>% pull(4)) # factor
```