p8105_hw1_mc5698

2024-09-20

```
knitr::opts_chunk$set(echo = TRUE)
```

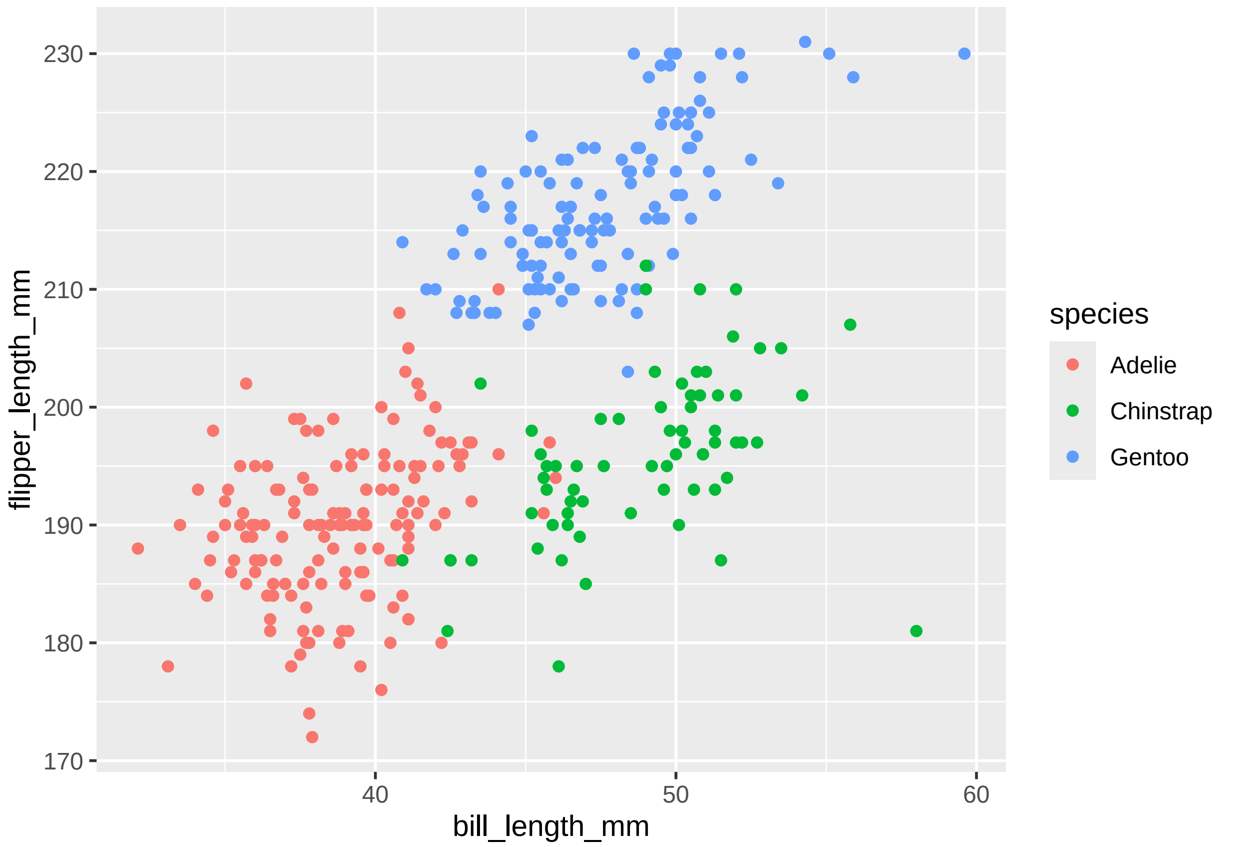
#question 1

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

The penguin dataset has 8 columns and 344 rows. There are 8 variables in this dataset include species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex, year. It records 3 kinds of penguins and they are Adelie, Gentoo, Chinstrap. They are from different islands include Torgersen, Biscoe, Dream. Their average flipper length is 200.9152047 mm.

```
library(ggplot2)
plotp = ggplot(penguins, aes(x = bill_length_mm, y = flipper_length_mm, color = species)) + geom_point(
plotp
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```r
ggsave("penguins scatter plot.pdf", plot=plotp)
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

#question 2

```r
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
```

```r
df = tibble(
  random_sample = rnorm(10),
  vec_char = sample(letters, 10),
  vec_logical = rnorm(10)>0,
  vec_factor = factor(sample(c("level 1", "level 2", "level 3"),10, replace = TRUE))
)
df
```

```
## # A tibble: 10 x 4
##    random_sample vec_char vec_logical vec_factor
##            <dbl> <chr>    <lgl>       <fct>
##  1       -0.658  w        FALSE       level 3
##  2       -0.238  c        FALSE       level 3
##  3       -1.61   k        FALSE       level 3
##  4       -0.887  u        FALSE       level 2
##  5       -0.246  g        TRUE        level 3
##  6       -0.886  m        FALSE       level 3
##  7       -0.351  i        FALSE       level 2
##  8       -0.0673 l        TRUE        level 1
##  9       -0.270  v        TRUE        level 2
## 10        0.170  e        FALSE       level 2
```

```r
mean(df%>%pull(random_sample))
```

```
## [1] -0.5044849
```

```r
mean(df%>%pull(vec_char))
```

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

```r
mean(df%>%pull(vec_logical))
```

```
## [1] 0.3
```

```r
mean(df%>%pull(vec_factor))
```

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

When I try to apply mean function to these variables, only vectors in `random_sample` and `vec_logical` work in this function and others show `NA` in the output.

```r
mean(as.numeric(df$vec_char))
```

```
## Warning in mean(as.numeric(df$vec_char)): NAs introduced by coercion
```

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

```r
mean(as.numeric(df$vec_logical))
```

```
## [1] 0.3
```

```r
mean(as.numeric(df$vec_factor))
```

```
## [1] 2.4
```

When I applies the as.numeric function to the logical, character, and factor variables, vectors in `vec_logical` and `vec_factor` could converted to numeric but vectors in `vec_char` still show `NA` in the output. Vectors in `vec_logical` can be `True` or `False`, which can be converted to 1 or 0, while `vec_char` cannot convert to numeric directly. This function would help me to convert vectors to numeric first and then apply the mean function on them. Therefore, the mean of `vec_logical`is 0.3 and the mean of `vec_factor` is 2.4.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

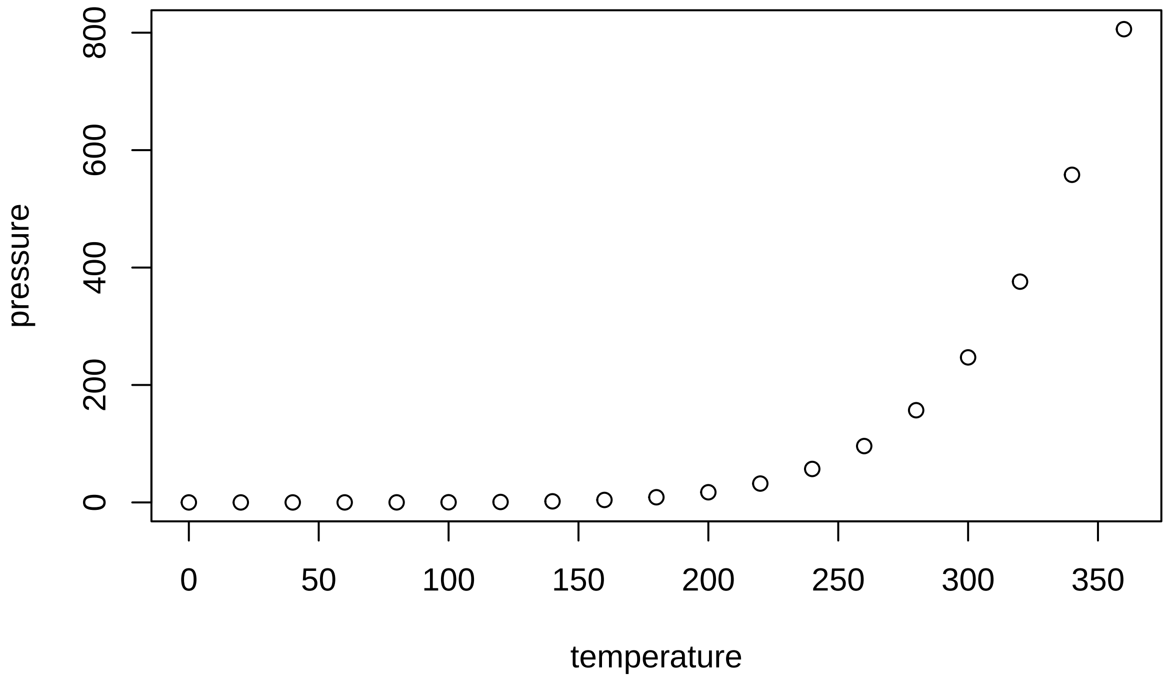
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.