Data Viz

Qahhar

2023-12-30

```
## load packages
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(forcats)
library(ggthemes)
library(knitr)
```

Part 1: Exploratory Data Analysis

The data we'll be using for this part of the project were downloaded from kaggle, and include information about "pet licenses issued by the Seattle Animal Shelter between 2005 and early 2017." We'll be exploring these data and generating a few exploratory plots in the first part of the project.

The Data

First, we'll read the data in from our `data/raw_data` directory.

```
# pets <- read.csv("/cloud/project/data/raw_data/seattle_pet_licenses.csv",
# stringsAsFactors = FALSE)

pets <- read.csv("/cloud/project/data/raw_data/seattle_pet_licenses.csv")

head(pets, n = 3)
```

```
##   animal_s_name      license_issue_date license_number
## 1          Ozzy 2005-03-29T00:00:00.000         130651
## 2          Jack 2009-12-23T00:00:00.000         898148
## 3        Ginger 2006-01-20T00:00:00.000          29654
##                       primary_breed     secondary_breed species zip_code
## 1 Dachshund, Standard Smooth Haired                          Dog    98104
## 2              Schnauzer, Miniature        Terrier, Rat      Dog    98107
## 3                 Retriever, Golden Retriever, Labrador     Dog    98117
```

It is noticed that a column `licence_issue_date` is not in the right format

```
pets <- pets %>%
  mutate(license_issue_date = ymd_hms(license_issue_date)) %>%
  mutate(license_issue_date = format(license_issue_date, "%Y-%m-%d")) %>%
  arrange(desc(license_issue_date))

# pets <- pets %>%
#   mutate(date =  lubridate::ymd_hms(license_issue_date),
#          ymd = as.yearmon(pets$date, "%y%m"))

head(pets, n = 3)
```

```
##   animal_s_name license_issue_date license_number      primary_breed
## 1          Foxy         2016-12-31         823671 Domestic Shorthair
## 2         Penny         2016-12-31             NA           Brittany
## 3 Snickerdoodle         2016-12-31         722662 Domestic Shorthair
##   secondary_breed species zip_code
## 1                     Cat    98107
## 2                     Dog    98133
## 3                     Cat    98133
```

Explore the Data

```
#Understanding the structure

glimpse(pets)
```

```
## Rows: 66,042
## Columns: 7
```

```
## $ animal_s_name      <chr> "Foxy", "Penny", "Snickerdoodle", "Jasper", "Gabbie~
## $ license_issue_date <chr> "2016-12-31", "2016-12-31", "2016-12-31", "2016-12-~
## $ license_number     <int> 823671, NA, 722662, 21434, 723818, NA, NA, 833723, ~
## $ primary_breed      <chr> "Domestic Shorthair", "Brittany", "Domestic Shortha~
## $ secondary_breed    <chr> "", "", "", "Retriever, Labrador", "", "", "", "", ~
## $ species            <chr> "Cat", "Dog", "Cat", "Dog", "Dog", "Dog", "Dog", "C~
## $ zip_code           <chr> "98107", "98133", "98133", "98136", "98136", "98112~
```

```
colnames(pets)
```

```
## [1] "animal_s_name"      "license_issue_date" "license_number"
## [4] "primary_breed"      "secondary_breed"    "species"
## [7] "zip_code"
```

```
# #Summarizing factor/character columns only
# char_cols <- pets %>% select_if(is.character)
# summary(char_cols)
#
# summary(pets$species)
```

- How many pet licenses are included in the dataset?

```
sum(!is.na(pets$license_number))
```

```
## [1] 43885
```

- How many unique pet names are included in the dataset (animal_s_name)

```
length(unique(pets$animal_s_name))
```

```
## [1] 15796
```

15796 unique names out of 66042. Find the more common names?

```
# Use the pipe operator to find the most common names
ten_most_common_names <- pets %>%
  na.omit() %>%
  filter(animal_s_name != "") %>%
  group_by(animal_s_name) %>%
  summarize(count = n()) %>%
  arrange(desc(count)) %>%
  top_n(10)
```

```
## Selecting by count
```

```
# Display the result
head(ten_most_common_names, n = 3)
```

```
## # A tibble: 3 x 2
##   animal_s_name count
##   <chr>         <int>
## 1 Lucy            411
## 2 Bella           318
## 3 Charlie         295
```

- How many different species are included in this dataset (species)?

```
length(unique(pets$species))
```

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

- Which species are included in the dataset?

```
unique(pets$species)
```

```
## [1] "Cat"       "Dog"       "Livestock"
```

```
#Count of each unique specie
species_counts <- pets %>%
  group_by(species) %>%
  summarise(count = n())

print(species_counts)
```

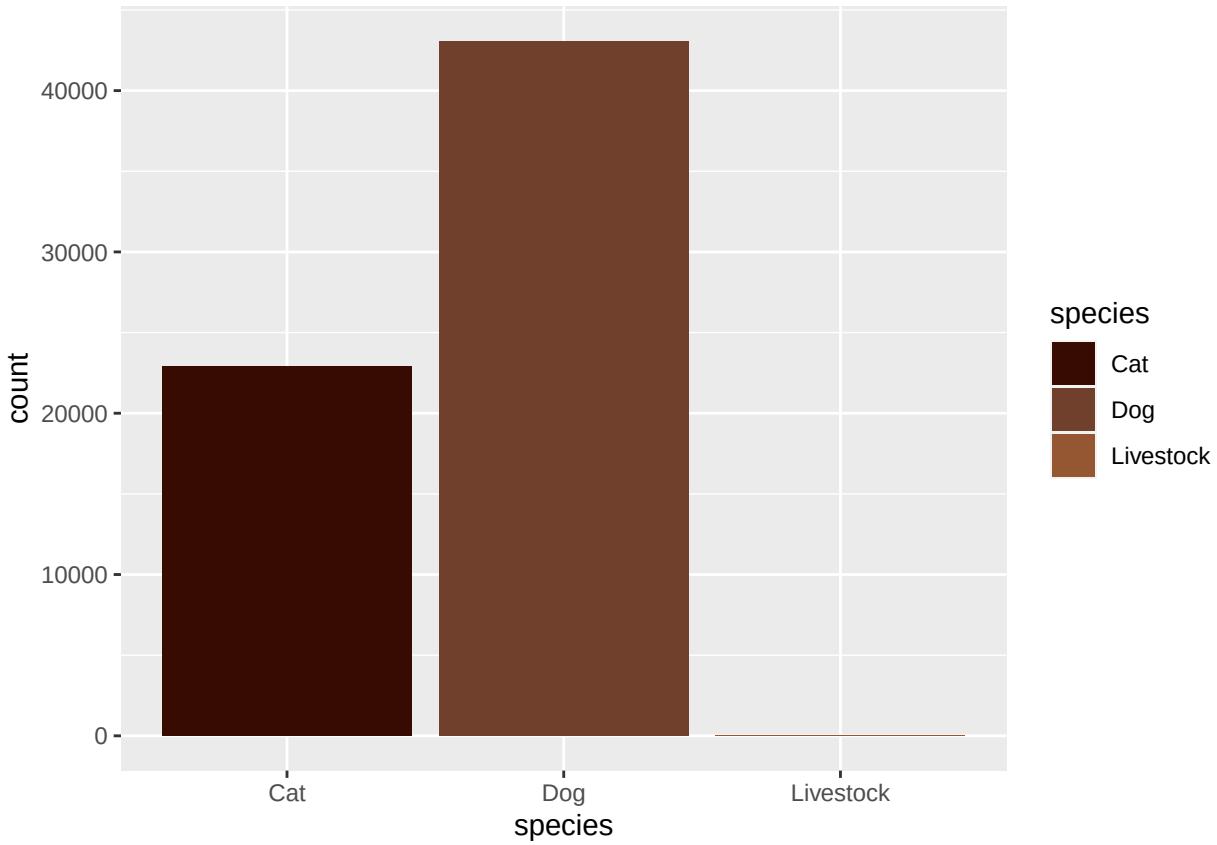
```
## # A tibble: 3 x 2
##   species   count
##   <chr>     <int>
## 1 Cat       22915
## 2 Dog       43076
## 3 Livestock    51
```

Visualize the Data

```
## visual breakdown of how many of each species

colors <- c('#370B01', "#70402C", "#955732")
ggplot(data = pets) +
  geom_bar(aes(x=species, fill = species, fill = species))+
  scale_fill_manual(values = colors)
```

```
## Warning: Duplicated aesthetics after name standardisation: fill
```




```
## Table: Most frequent Cat Name
pets %>%
  filter(species == "Cat", animal_s_name != "") %>%
  group_by(animal_s_name) %>%
  summarise(n = n()) %>%
  arrange(-n) %>%
  top_n(n = 10) %>%
  knitr::kable(., caption = "Top 10 Cat Names in Seattle")
```

```
## Selecting by n
```

Table 1: Top 10 Cat Names in Seattle

animal_s_name	n
Lucy	150
Max	120
Luna	119
Bella	113
Oliver	108
Charlie	99
Lily	93
Jack	87
Sophie	81
Shadow	69

```r
#using either arrange(desc(n)) or arrange(-n) will achieve the same result:

# arrange(desc(n)): Sorts the values in descending order based on the column n.
# arrange(-n): Also sorts the values in descending order based on the column n.
```

```r
## Table: Most frequent Dog Name
pets %>%
  filter(species == "Dog", animal_s_name != "") %>%
  group_by(animal_s_name) %>%
  summarise(n = n()) %>%
  arrange(-n) %>%
  top_n(n = 10) %>%
  knitr::kable(., caption = "Top 10 Dog Names in Seattle")
```

```
## Selecting by n
```

Table 2: Top 10 Dog Names in Seattle

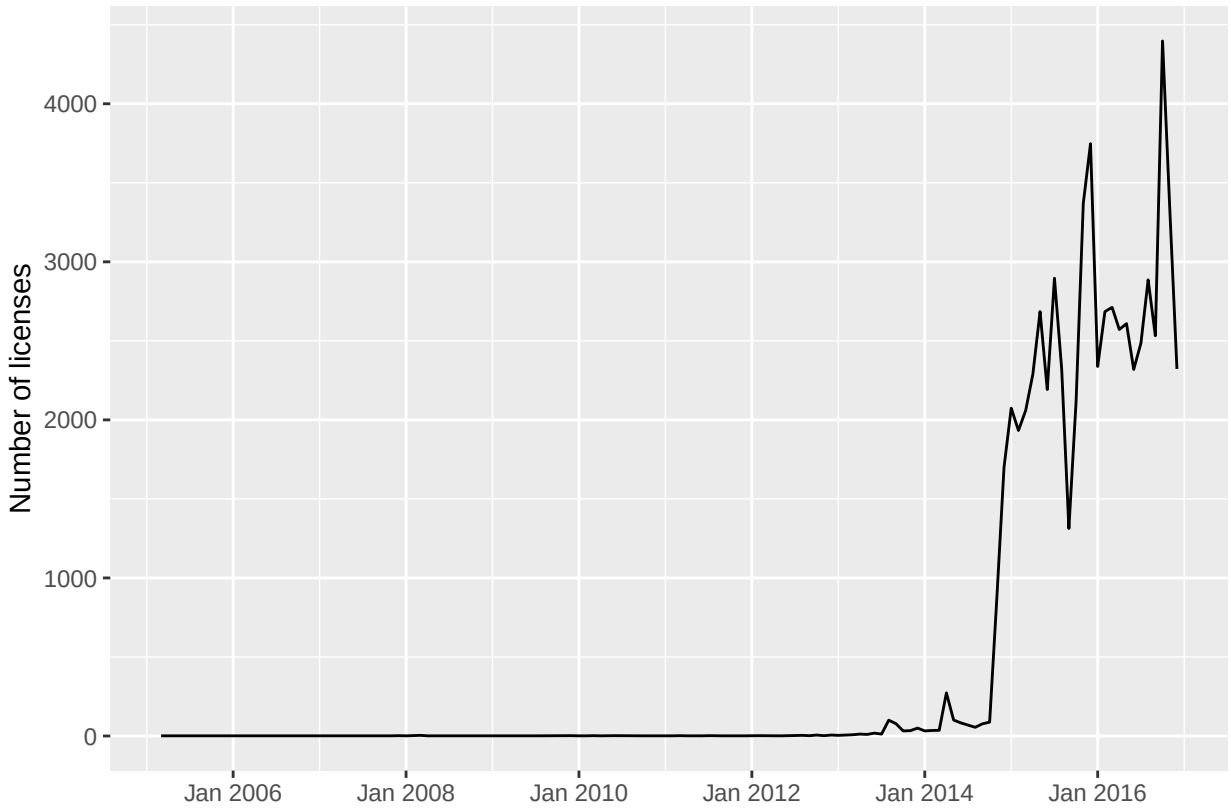
animal_s_name	n
Lucy	416
Charlie	348
Bella	338
Buddy	256
Daisy	256
Max	254
Molly	243
Luna	242
Lola	232
Maggie	226

- How the number of licenses recorded has changed over time?

```r
# add date and ym columns
# pets$ym <- as.yearmon(pets$license_issue_date, "%y%m")

pets$ym <- as.yearmon(pets$license_issue_date, "%Y-%m-%d")

# how the number of licenses recorded has changed over time
pets %>%
  group_by(ym) %>%
  ## count number within each group
  summarize(n= n()) %>%
  ggplot(., aes(ym, n)) +
  ## geom name for line chart
  geom_line() +
  scale_x_yearmon() +
  xlab("") +
  ylab("Number of licenses")
```



Part 2: Explanatory Data Analysis

The data used in this part of the project were dowloaded from FiveThirtyEight - `steak-survey`. They were originally used in the article: How Americans Like Their Steak. The goal of this part of the project will be to recreate the data visualization used in this article.

The Data

The data focus on a question of choice,

Consider the following hypothetical situations - In Lottery A, you have a 50 chance of success with a payout of 100

- In Lottery B you have a 90 chance of success with a payout of 20

Assuming you have 10 to bet, would you play Lottery A or Lottery B.

```
## read in the data
steak <- read.csv("/cloud/project/data/raw_data/steak-risk-survey.csv") %>%
  ## remove first row which just includes the word "Response" in each column
  slice(2:n())

head(steak, n = 0)
```

```
##  [1] RespondentID
##  [2] Consider.the.following.hypothetical.situations...br.In.Lottery.A..you.have.a.50..chance.of.succ
##  [3] Do.you.ever.smoke.cigarettes.
##  [4] Do.you.ever.drink.alcohol.
##  [5] Do.you.ever.gamble.
```

```
##  [6] Have.you.ever.been.skydiving.
##  [7] Do.you.ever.drive.above.the.speed.limit.
##  [8] Have.you.ever.cheated.on.your.significant.other.
##  [9] Do.you.eat.steak.
## [10] How.do.you.like.your.steak.prepared.
## [11] Gender
## [12] Age
## [13] Household.Income
## [14] Education
## [15] Location..Census.Region.
## <0 rows> (or 0-length row.names)
```

Column names are lengthy, try changing?

```
steak <- steak %>%
  rename(lottery_choice = Consider.the.following.hypothetical.situations...br.In.Lottery.A..you.have.a.5

glimpse(steak)
```

```
## Rows: 550
## Columns: 15
## $ RespondentID                                 <dbl> 3237565956, 323498234~
## $ lottery_choice                               <chr> "Lottery B", "Lottery~
## $ Do.you.ever.smoke.cigarettes.                <chr> "", "No", "No", "Yes"~
## $ Do.you.ever.drink.alcohol.                   <chr> "", "Yes", "Yes", "Ye~
## $ Do.you.ever.gamble.                          <chr> "", "No", "Yes", "Yes~
## $ Have.you.ever.been.skydiving.                <chr> "", "No", "No", "No",~
## $ Do.you.ever.drive.above.the.speed.limit.     <chr> "", "No", "Yes", "Yes~
## $ Have.you.ever.cheated.on.your.significant.other. <chr> "", "No", "Yes", "Yes~
## $ Do.you.eat.steak.                            <chr> "", "Yes", "Yes", "Ye~
## $ How.do.you.like.your.steak.prepared.         <chr> "", "Medium rare", "R~
## $ Gender                                       <chr> "", "Male", "Male", "~
## $ Age                                          <chr> "", "> 60", "> 60", "~
## $ Household.Income                             <chr> "", "$50,000 - $99,99~
## $ Education                                    <chr> "", "Some college or ~
## $ Location..Census.Region.                     <chr> "", "East North Centr~
```

Explore the Data

- How many people participated in the survey?

```
nrow(steak)
```

```
## [1] 550
```

- How many people responded "Yes" to the question "Do you eat steak?"

```
yes_steak <- steak %>%
  filter(Do.you.eat.steak. == 'Yes') %>%
  summarise(count = n())

yes_steak
```

```
##   count
## 1   430
```

- How many different (unique) responses were there to the question "How do you like your steak prepared?"

```r
new_steak <- steak %>%
  filter(How.do.you.like.your.steak.prepared. != "")

length(unique(new_steak$How.do.you.like.your.steak.prepared.))
```

```
## [1] 5
```

```r
unique(new_steak$How.do.you.like.your.steak.prepared.)
```

```
## [1] "Medium rare" "Rare"        "Medium"      "Medium Well" "Well"
```

Wrangle the Data

```r
# Create a new data frame from steak data frame and assign it to the variable `pref`
pref <- steak %>%
  #code below creates a column 'steak_pref' with its values being the unique characters of the How.do.y
  mutate(steak_pref = factor(How.do.you.like.your.steak.prepared.,
                             levels = c("Well",
                                        "Medium Well",
                                        "Medium",
                                        "Medium rare",
                                        "Rare"))) %>%
  #Filters out the values in the steak_pref column that are notated as ''
  filter(steak_pref != "") %>%
  group_by(steak_pref) %>%
  #Using the summarize function we create a new col 'n' which basically takes the count of individual g
  summarise(n = n()) %>%
  # Create a new column that creates a ratio of each unique character, by dividing their sum over the t
  mutate(ratio = n / sum(n))

ncol(pref)
```

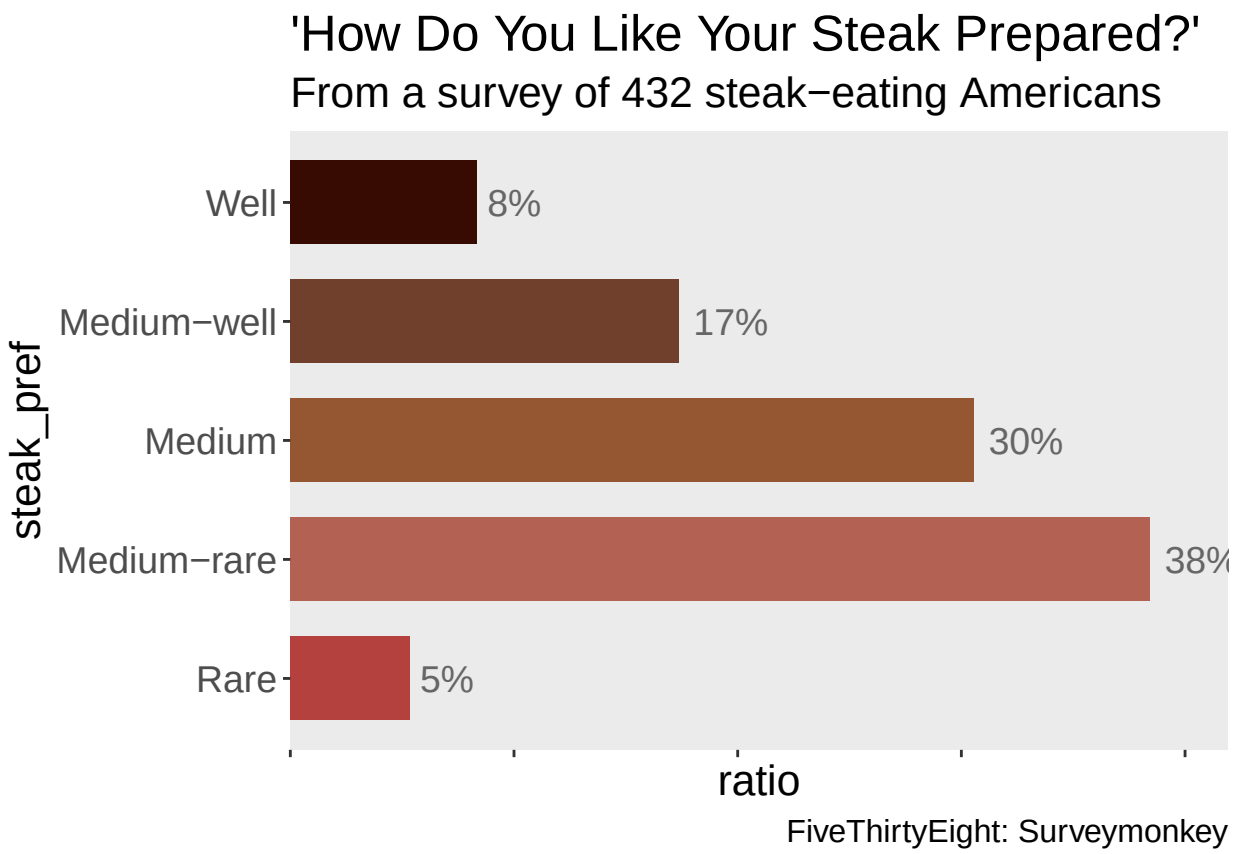
```
## [1] 3
```

Visualize the Data

```r
## generate the plot
p <- ggplot(pref) +
  ## specify you want to generate a bar chart
  geom_bar(aes(x = steak_pref, y = ratio, fill = steak_pref),
               stat = 'identity',
               width = 0.7) +
  ## this adds text labels (you don't have to change anything here)
  geom_text(aes(label = paste0(as.integer(ratio*100),"%"),
               x = steak_pref,
               y = ratio),
            stat= "identity",
            hjust = -0.2,
            size = 5,
            color = "grey40") +
  ## flip coordinates to make horizontal box plot
  coord_flip() +
  ## change the colors of the bars
  scale_fill_manual(values = c("#370B01",
```

```
                            "#70402C",
                            "#955732",
                            "#B36153",
                            "#B5413F")) +
  ## change the scale/labels of the steak-wellness variable (x-axis)
  scale_x_discrete(limits = levels(fct_rev(pref$steak_pref)),
                labels = c("Well",
                            "Medium Well" = "Medium-well",
                            "Medium",
                            "Medium rare"="Medium-rare",
                            "Rare")) +
  # change the scale/labels of the percent axis (y-axis)
  scale_y_continuous(labels = scales::percent,
                expand = c(mult = c(0,0),
                            add = c(0,0.035))) +
  ## change the title, subtitle, and caption
  labs(title="'How Do You Like Your Steak Prepared?'",
                subtitle="From a survey of 432 steak-eating Americans",
                caption="FiveThirtyEight: Surveymonkey") +
  ## change the theme (use ggthemes)
  theme_grey() +
  ## fine tune the theme
  theme(axis.text = element_text(size = 14),
                title = element_text(size = 16),
                legend.position="none",
                plot.caption=element_text(size = 12),
                panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                axis.text.x = element_blank()
  )

p
```



Save the Plot

```
## save plot to figures/explanatory_figures directory
ggsave(plot = p, filename = '/cloud/project/steak_graph.png', width = 8, height = 4)
```

Session Info

This tells people what software versions you were using when you ran this notebook.

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
## LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3;  LAPACK version 3.9.0
##
## locale:
##  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
##  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C.UTF-8
##  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
## [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
##
## time zone: UTC
## tzcode source: system (glibc)
```

```
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.45      ggthemes_5.0.0  forcats_1.0.0   zoo_1.8-12
## [5] lubridate_1.9.3 dplyr_1.1.4     ggplot2_3.4.4   janitor_2.2.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.4     highr_0.10       compiler_4.3.2   tidyselect_1.2.0
##  [5] stringr_1.5.1    snakecase_0.11.1 scales_1.3.0     yaml_2.3.8
##  [9] fastmap_1.1.1    lattice_0.21-9   R6_2.5.1         labeling_0.4.3
## [13] generics_0.1.3   tibble_3.2.1     munsell_0.5.0    pillar_1.9.0
## [17] rlang_1.1.2      utf8_1.2.4       stringi_1.8.3    xfun_0.41
## [21] timechange_0.2.0 cli_3.6.2        withr_2.5.2      magrittr_2.0.3
## [25] digest_0.6.33    grid_4.3.2       lifecycle_1.0.4  vctrs_0.6.5
## [29] evaluate_0.23    glue_1.6.2       farver_2.1.1     fansi_1.0.6
## [33] colorspace_2.1-0 purrr_1.0.2      rmarkdown_2.25   tools_4.3.2
## [37] pkgconfig_2.0.3  htmltools_0.5.7
```