My first Quarto

Marty Faville

## Start here

I am writing this for new code (Insert or shortcut Ctrl + Alt + i)

################################################  
### Code 1: Read in data  
################################################  
  
################################################  
### The code in this file covers the following:  
################################################  
  
 # Read in external data (Excel files, CSVs) with readr and readxl  
 # Initial data exploration  
 # Set up GitHub  
   
# Libraries needed  
  
library(here) # easier file referencing

Warning: package 'here' was built under R version 4.4.2

here() starts at C:/Marty\_R\_working\_folder/r-examples

library(tidyverse) # reading in CSV files and more

Warning: package 'tidyverse' was built under R version 4.4.2

Warning: package 'ggplot2' was built under R version 4.4.2

Warning: package 'tibble' was built under R version 4.4.2

Warning: package 'tidyr' was built under R version 4.4.2

Warning: package 'readr' was built under R version 4.4.2

Warning: package 'purrr' was built under R version 4.4.2

Warning: package 'dplyr' was built under R version 4.4.2

Warning: package 'stringr' was built under R version 4.4.2

Warning: package 'forcats' was built under R version 4.4.2

Warning: package 'lubridate' was built under R version 4.4.2

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
✔ purrr 1.0.4

── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(readxl) # reading in Excel files

Warning: package 'readxl' was built under R version 4.4.2

## Read in external data (Excel files, CSVs) with readr and readxl  
  
# Our first data set is a comma-separated-value (CSV) file containing visitation   
# data for all National Parks in California (ca\_np.csv)  
  
ca\_np <- read\_csv(here("data", "ca\_np.csv"))

Rows: 789 Columns: 7  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (5): region, state, code, park\_name, type  
dbl (2): visitors, year  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Our second data set is a single Excel worksheet containing only visitation for   
# Channel Islands National Park (ci\_np.xlsx)  
  
ci\_np <- readxl::read\_excel(here("data", "ci\_np.xlsx"))  
  
## Initial data exploration  
  
# Can look at the data in the environment tab,   
# or with functions such as follows:  
  
names(ca\_np) # check what all the variable names are

[1] "region" "state" "code" "park\_name" "type" "visitors"   
[7] "year"

head(ca\_np) # first 6 rows of the data

# A tibble: 6 × 7  
 region state code park\_name type visitors year  
 <chr> <chr> <chr> <chr> <chr> <dbl> <dbl>  
1 PW CA CHIS Channel Islands National Park National Park 1200 1963  
2 PW CA CHIS Channel Islands National Park National Park 1500 1964  
3 PW CA CHIS Channel Islands National Park National Park 1600 1965  
4 PW CA CHIS Channel Islands National Park National Park 300 1966  
5 PW CA CHIS Channel Islands National Park National Park 15700 1967  
6 PW CA CHIS Channel Islands National Park National Park 31000 1968

summary(ca\_np) # quick summary of each variable

region state code park\_name   
 Length:789 Length:789 Length:789 Length:789   
 Class :character Class :character Class :character Class :character   
 Mode :character Mode :character Mode :character Mode :character   
   
   
   
 type visitors year   
 Length:789 Min. : 300 Min. :1904   
 Class :character 1st Qu.: 111385 1st Qu.:1946   
 Mode :character Median : 393219 Median :1973   
 Mean : 607632 Mean :1970   
 3rd Qu.: 797800 3rd Qu.:1995   
 Max. :5028868 Max. :2016

## Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

1 + 1

[1] 2

You can add options to executable code like this

[1] 4

The echo: false option disables the printing of code (only output is displayed).