IS5 in R: Stats Starts Here (Chapter 1)

Nicholas Horton (nhorton@amherst.edu)

December 19, 2020

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fifth Edition of *Intro Stats* (2018) by De Veaux, Velleman, and Bock. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at http://nhorton.people.amherst.edu/is5.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (https://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

Chapter 1: Stats Starts Here

Section 1.1: What is Statistics?

Section 1.2: Data

Section 1.3: Variables

See table on page 7.

```
library(mosaic)
options(digits = 3)
Tour <-
  readr::read_csv("http://nhorton.people.amherst.edu/is5/data/Tour_de_France_2016.csv") %>%
  janitor::clean_names()
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Year = col_double(),
##   Winner = col_character(),
##   Country = col_character(),
##   Age = col_double(),
##   Team = col_character(),
##   `Total Time(h.min.sec)` = col_character(),
##   `Total Time(h)` = col_double(),
##   Average.Speed = col_double(),
##   Stages = col_double(),
##   `Total Distance Ridden` = col_double(),
##   `Starting Riders` = col_double(),
##   `Finishing Riders` = col_double()
## )
```

By default, `read_csv()` prints the variable names. These messages can be suppressed using the `message=FALSE` code chunk option to save space and improve readability.

```
names(Tour)
```

```
##  [1] "year"                 "winner"               "country"
##  [4] "age"                  "team"                 "total_time_h_min_sec"
##  [7] "total_time_h"         "average_speed"        "stages"
## [10] "total_distance_ridden" "starting_riders"      "finishing_riders"
```

```
glimpse(Tour)
```

```
## Rows: 103
## Columns: 12
## $ year                  <dbl> 1903, 1904, 1905, 1906, 1907, 1908, 1909, 191...
## $ winner                <chr> "Maurice Garin", "Henri Cornet", "Louis Trous...
## $ country               <chr> "France", "France", "France", "France", "Fran...
## $ age                   <dbl> 32, 20, 24, 27, 24, 25, 22, 21, 27, 24, 23, 2...
## $ team                  <chr> "La Fran\u008daise", "Cycles JC", "Peugeot", ...
## $ total_time_h_min_sec  <chr> "94.33.00", "96.05.56", "110.26.58", "189.34....
## $ total_time_h          <dbl> 94.5, 96.1, 110.4, 189.6, 158.8, 156.9, 157.0...
## $ average_speed         <dbl> 25.7, 25.3, 27.1, 24.5, 28.5, 28.7, 28.7, 29....
## $ stages                <dbl> 6, 6, 11, 13, 14, 14, 14, 15, 15, 15, 15, 15,...
## $ total_distance_ridden <dbl> 2428, 2428, 2994, 4637, 4488, 4488, 4497, 473...
## $ starting_riders       <dbl> 60, 88, 60, 82, 93, 112, 150, 110, 84, 131, 1...
## $ finishing_riders      <dbl> 21, 27, 24, 14, 33, 36, 55, 41, 28, 41, 25, 5...
```

```
head(Tour, 3)
```

```
## # A tibble: 3 x 12
##    year winner country   age team  total_time_h_mi~ total_time_h average_speed
##   <dbl> <chr>  <chr>   <dbl> <chr> <chr>                   <dbl>         <dbl>
## 1  1903 Mauri~ France     32 "La ~ 94.33.00                 94.6          25.7
## 2  1904 Henri~ France     20 "Cyc~ 96.05.56                 96.1          25.3
## 3  1905 Louis~ France     24 "Peu~ 110.26.58               110.           27.1
## # ... with 4 more variables: stages <dbl>, total_distance_ridden <dbl>,
## #   starting_riders <dbl>, finishing_riders <dbl>
```

```
tail(Tour, 8) %>%
  select(winner, year, country)
```

```
## # A tibble: 8 x 3
##   winner              year country
##   <chr>              <dbl> <chr>
## 1 Contador Alberto    2009 Spain
## 2 Andy Schleck        2010 Luxembourg
## 3 Cadel Evans         2011 Australia
## 4 Bradley Wiggins     2012 Great Britain
## 5 Christopher Froome  2013 Great Britain
## 6 Vincezo Nibali      2014 Italy
## 7 Cristopher Froome   2015 Great Britain
## 8 Cristopher Froome   2016 Great Britain
```

Piping (`%>%`) takes the output of the line of code and uses it in the next.

Let's find who was the winner in 1998 We use the `filter()` command.

```
filter(Tour, year == 1998) %>%
  select(winner, year, country)
```

```
## # A tibble: 1 x 3
##   winner         year country
##   <chr>         <dbl> <chr>
## 1 Marco Pantani  1998 Italy
```

How many stages were there in the tour in the year that Alberto Contador won? We can also use the `filter()` command.

```
filter(Tour, winner == "Contador Alberto") %>%
  select(winner, year, stages)
```

```
## # A tibble: 2 x 3
##   winner            year stages
##   <chr>            <dbl>  <dbl>
## 1 Contador Alberto  2007     21
## 2 Contador Alberto  2009     21
```

Note that the following command generates the same output.

```
Tour %>%
  filter(winner == "Contador Alberto") %>%
  select(winner, year, stages)
```

```
## # A tibble: 2 x 3
##   winner            year stages
##   <chr>            <dbl>  <dbl>
## 1 Contador Alberto  2007     21
## 2 Contador Alberto  2009     21
```

The pipe operator (%>%) can be used to connect one dataframe or command to another.

What was the slowest average speed of any tour? Fastest? Again, we use `filter()` but this time in conjunction with the `min()` function.

```
filter(Tour, average_speed == min(average_speed)) %>%
  select(year, average_speed)
```

```
## # A tibble: 1 x 2
##    year average_speed
##   <dbl>         <dbl>
## 1  1919          24.1
```

```
filter(Tour, average_speed == max(average_speed)) %>%
  select(year, average_speed)
```

```
## # A tibble: 1 x 2
##    year average_speed
##   <dbl>         <dbl>
## 1  2005          41.7
```

```
df_stats(~average_speed, data = Tour)
```

How can we summarize the distribution of Average Speeds?

```
##        response  min   Q1 median   Q3  max mean  sd   n missing
## 1 average_speed 24.1 29.5   35.4 38.7 41.7 34.1 5.2 103       0
```

Note that `~x` denotes the simplest form of the general modelling language (used to indicate a single variable in mosaic).