Cherry Blossom Run, 2017

(Chriss Jordan Oboa)

2/16/2022

This is a modified version of an activity first created by Dr. Parson for MATH 213 in Spring 2021.

As always, we start by loading some R packages. The `tidyverse` package includes tools for working with "tidy" data, which is the kind of data that we learned about in Chapter 1 of our textbook. Also load the `ggdist` and `ggridges` packages below. These will help us with graphing our data.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggdist)
library(ggridges)
```

```
##
## Attaching package: 'ggridges'
```

```
## The following objects are masked from 'package:ggdist':
##
##     scale_point_color_continuous, scale_point_color_discrete,
##     scale_point_colour_continuous, scale_point_colour_discrete,
##     scale_point_fill_continuous, scale_point_fill_discrete,
##     scale_point_size_continuous
```

The data

We will get the data for this activity directly from our textbook's web site. Fill in the code below to load the data following these steps:

- give the dataframe the name `run17` (without the backticks)
- remember to use the assignment operator `<-` (again, without the backticks)
- we can use the same `read_csv` command we have been using
- but we use the URL for the data which is https://www.openintro.org/data/csv/run17.csv instead of pointing R to a file in our project.

```
run17<-read_csv("https://www.openintro.org/data/csv/run17.csv")
```

```
## Rows: 19961 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): name, sex, city, event
## dbl (5): bib, age, net_sec, clock_sec, pace_sec
##
## 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.
```

Question 1

Which variables are categorical? Categorical are name, sex, city, event.

Question 2

Have R list the levels of each of the categorical variables. (Use the `distinct` command.)

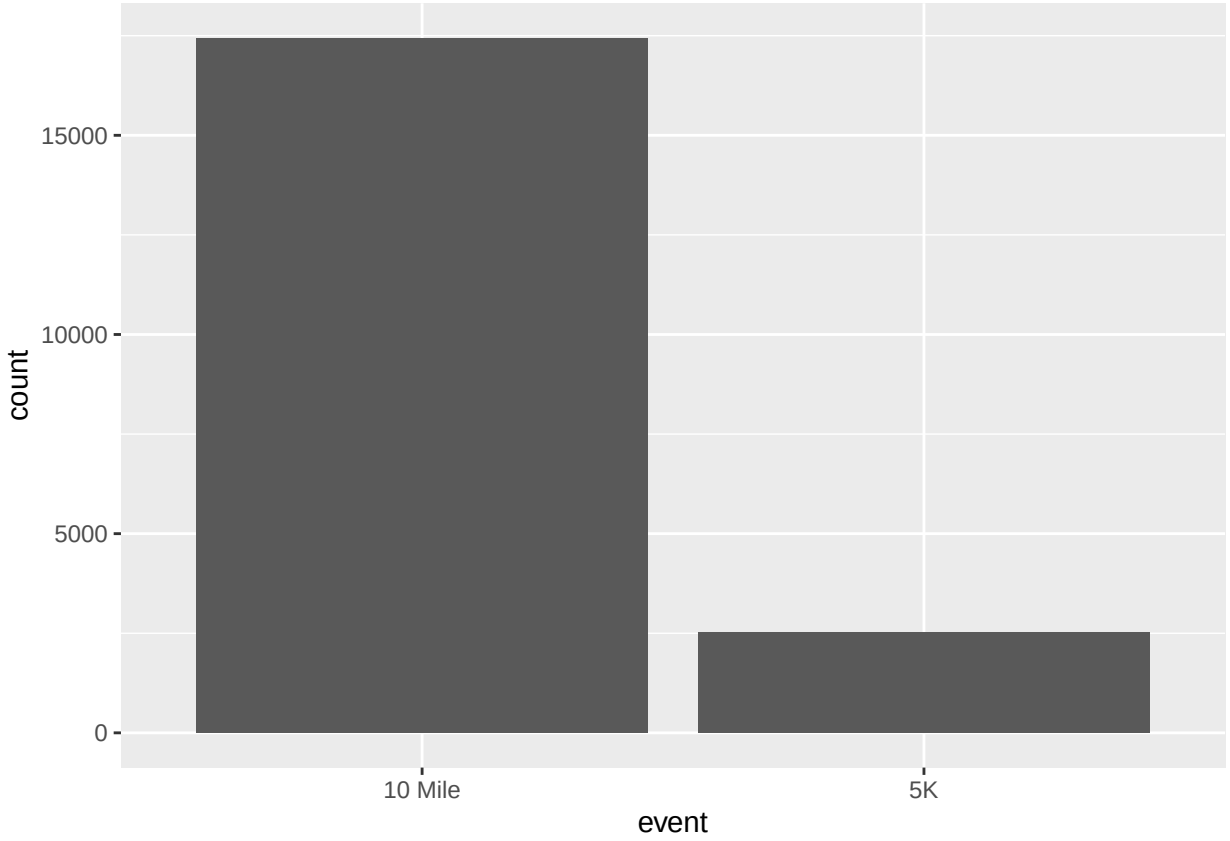
```
run17 %>%
  distinct(city, event, name, sex)
```

```
## # A tibble: 18,598 x 4
##    name         sex   city                event
##    <chr>        <chr> <chr>               <chr>
##  1 Hiwot G.     F     Ethiopia            10 Mile
##  2 Buze D.      F     Ethiopia            10 Mile
##  3 Gladys K.    F     Kenya               10 Mile
##  4 Mamitu D.    F     Ethiopia            10 Mile
##  5 Karolina N.  F     Poland              10 Mile
##  6 Firehiwot D. F     Ethiopia            10 Mile
##  7 Tara W.      F     Portland, OR        10 Mile
##  8 Nancy N.     F     Kenya               10 Mile
##  9 Hannah D.    F     Saratoga Springs, NY 10 Mile
## 10 Susanna S.   F     Reston, VA          10 Mile
## # ... with 18,588 more rows
```

Question 3

Create a bar graph of the distributions for `event`. Also create a frequency table for the distribution of this variable. Then create a relative frequency table. Write a sentence or two underneath your graph and tables to describe the distribution. Remember to speak in context and to back up what you say by pulling numbers from the graph and your tables.

```
ggplot(data = run17 , mapping = aes(x = event)) +
 geom_bar()
```




```
run17 %>%
  group_by(event) %>%
  summarize(n = n())
```

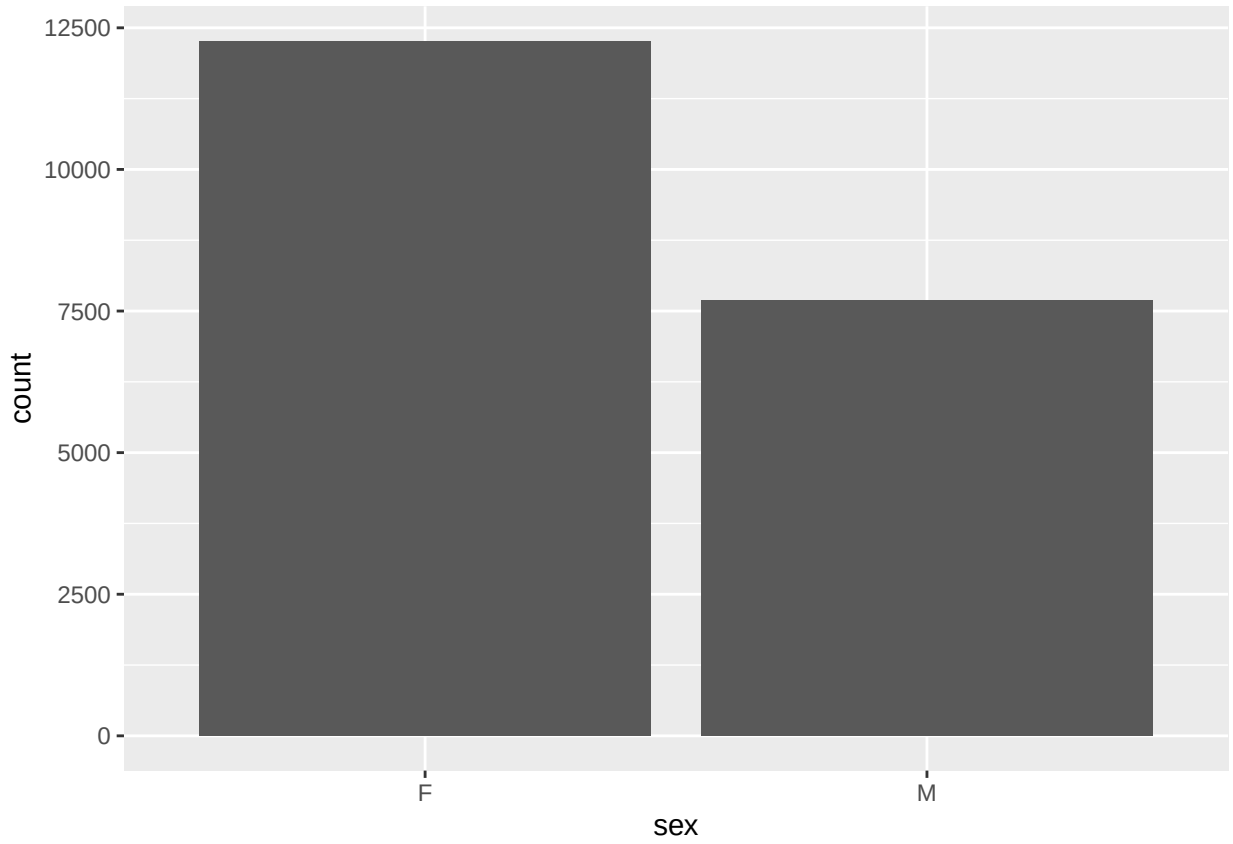
```
## # A tibble: 2 x 2
##   event       n
##   <chr>   <int>
## 1 10 Mile 17442
## 2 5K       2519
```

As shown on the graph, there are around 1750 cases running the 10k and 2500 cases running the 5k.

Question 4

Repeat the previous exercise for the variable **sex**. Create a bar graph, frequency table, relative frequency table, and write a description.

```
ggplot(data = run17 , mapping = aes(x = sex)) +
 geom_bar()
```



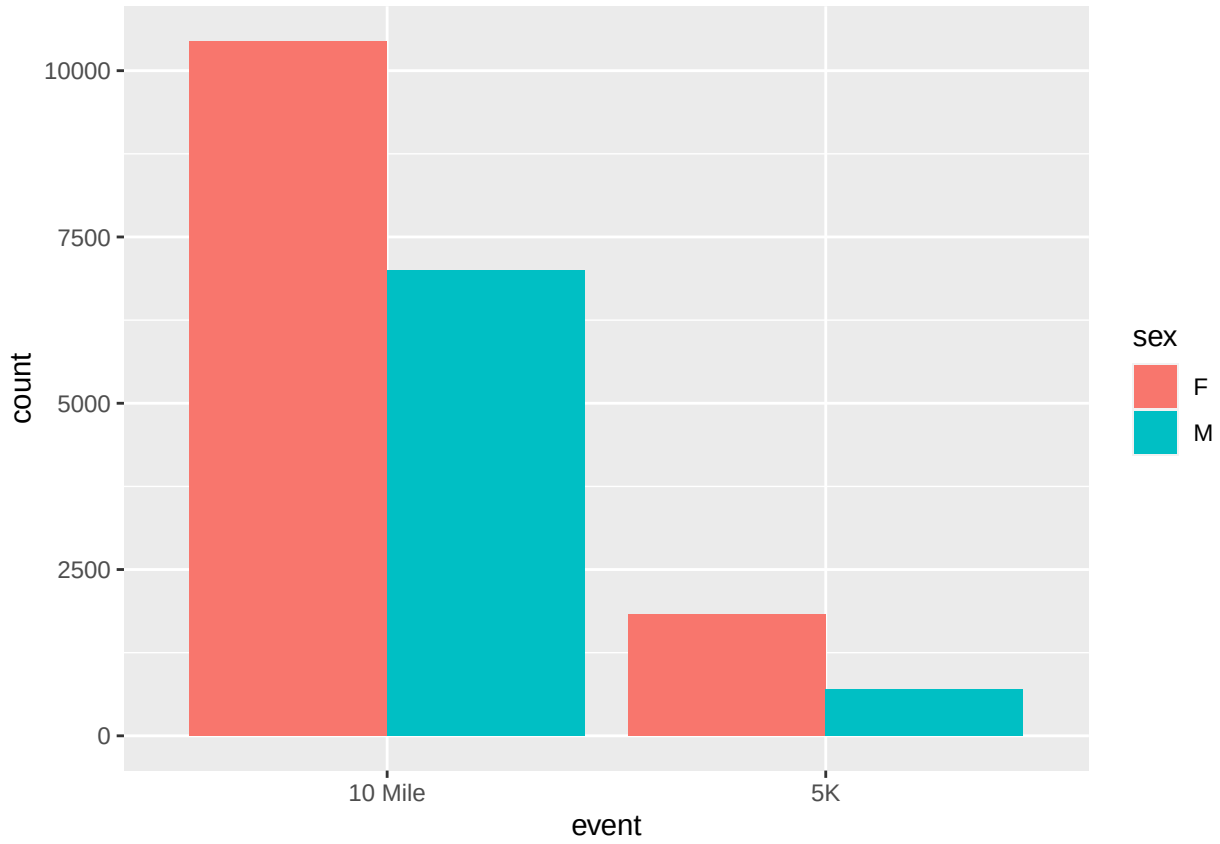

```
run17 %>%
  group_by(sex) %>%
  summarize(n = n())
```

```
## # A tibble: 2 x 2
##   sex       n
##   <chr> <int>
## 1 F     12267
## 2 M      7694
```

Question 5

Next, create a stacked bar graph and a dodged bar graph for the variable **event**, with fill **sex**. (The variable event should appear on the x-axis.) Also create a relative frequency table. Write some sentences to describe what you learn. Remember to speak in context and to back up what you say by pulling numbers from the graph and your tables.

```
ggplot(data = run17, mapping = aes(x = event, fill = sex)) +
  geom_bar(position = "dodge")
```




```
run17 %>%
  group_by(event, sex)%>%
  summarize(n = n()) %>%
  mutate(proportion = n/sum(n))
```

```
## `summarise()` has grouped output by 'event'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   event [2]
##   event   sex       n proportion
##   <chr>   <chr> <int>      <dbl>
## 1 10 Mile F     10446      0.599
## 2 10 Mile M      6996      0.401
## 3 5K      F      1821      0.723
## 4 5K      M       698      0.277
```

According to the charts and datas, the Cherry Blossom races had 19263. Most runnners were females and they were around 63% of all the cases. Of all the people in the 10k, only 40% are males. In addition, Of all the people in the 5k, only 72% are females.