STA 230 Hw 2 Solutions

Book Exercises

Chapter 4

Problem 1 Here, we are just checking that their code goes from what was initially given below to somthing more legible

```
## Original mess
flights|>filter(dest=="IAH")|>group_by(year,month,day)|>summarize(n=n(),
delay=mean(arr_delay,na.rm=TRUE))|>filter(n>10)

flights|>filter(carrier=="UA",dest%in%c("IAH","HOU"),sched_dep_time>
0900,sched_arr_time<2000)|>group_by(flight)|>summarize(delay=mean(
arr_delay,na.rm=TRUE),cancelled=sum(is.na(arr_delay)),n=n())|>filter(n>10)
```

This is better

```
flights |>
  filter(dest=="IAH") |>
  group_by(year,month,day) |>
  summarize(n=n(), delay=mean(arr_delay,na.rm=TRUE)) |>
  filter(n>10)

flights |> filter(carrier=="UA",dest %in% c("IAH","HOU"),
                  sched_dep_time > 0900,
                  sched_arr_time < 2000) |>
  group_by(flight) |>
  summarize(delay=mean(arr_delay,na.rm=TRUE),
            cancelled=sum(is.na(arr_delay)),
            n=n()) |>
  filter(n > 10)
```

Chapter 19.3.4

```
library(nycflights13)
library(dplyr)

## Load into env
data("planes")
data("flights")
data("weather")
data("airports")
```

Problem 3

Does every departing flight have corresponding weather data for that hour?

We want to see if there are any flights (left) that have no corresponding entry in weather (right). Doing an anti join, we see there are quite a few at a limited number of time hours. Not sure why this should be

```
## Confirm this table not empty
tt <- anti_join(flights, weather, by = c("time_hour", "origin"))

## See flights with missing times
# anti_join(flights, weather, by = c("time_hour", "origin"))$time_hour %>% table()
```

Problem 7

Compute the average delay by destination, then join on the airports data frame so you can show the spatial distribution of delays. Here's an easy way to draw a map of the United States:

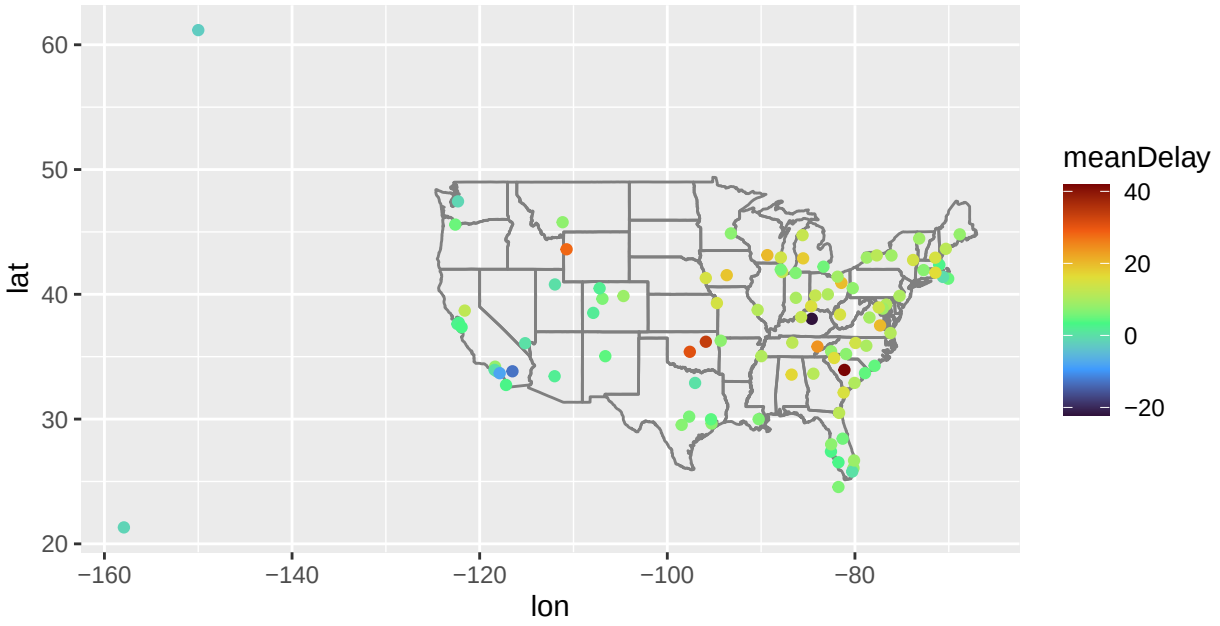
```
library(ggplot2)
airports |>
  semi_join(flights, join_by(faa == dest)) |>
  ggplot(aes(x = lon, y = lat)) +
    borders("state") +
    geom_point() +
    coord_quickmap()
```

Ok, let's start by aggregating times (and removing locations with missing values). Then we punch that joined dataset straight into ggplot2 using the template from above

```
airsum <- group_by(flights, dest) %>%
  filter(!is.na(arr_delay)) %>%
  summarize(meanDelay = mean(arr_delay, na.rm = TRUE))

## Combine with airports
tt <- inner_join(airports, airsum, by = join_by(faa == dest))

## Recreate plot they gave
  ggplot(tt, aes(x = lon, y = lat)) +
    borders("state") +
    geom_point(aes(color = meanDelay)) +
    coord_quickmap() +
  scale_color_continuous(type = "viridis", option = "H")
```



Extra Exercises

Question 1

The data frame `economics` is included in the `ggplot2` package and contains US economic data provided by the US Federal Reserve

```
library(ggplot2)

data(economics)
```

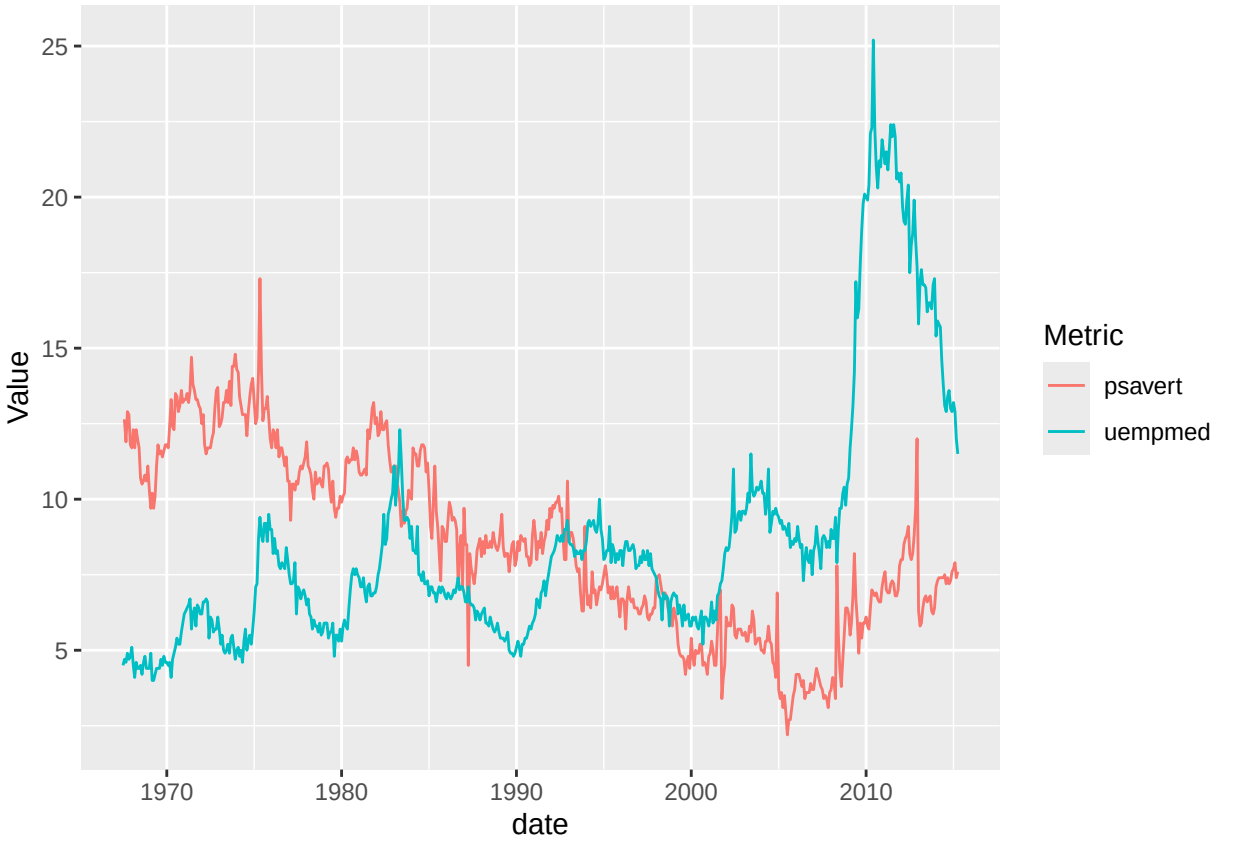
Using the `economics` dataset, do the following:

- Only retain those variables associated with `date`, `psavert` (personal savings rate), and `uempmed` (median duration of unemployment)
- Pivot the data so that each row contains one economic outcome
- Create a line graph in ggplot using color to differentiate the metrics. Briefly, what relationship do you see between median duration of unemployment and personal savings rate?

Solution

```
eco_long <- economics %>%
  select(date, psavert, uempmed) %>%
  pivot_longer(cols = !date,
               names_to = "Metric",
               values_to = "Value")
```

```
ggplot(eco_long, aes(date, Value, color = Metric)) +
  geom_line()
```



Question 2

For this question, we need to install that `Lahman` package which is a database of Major League Baseball statistics collected by Sean Lahman from the 1871-2016 seasons. The database contains several data.frames which can be loaded into our environment using the `data()` function.

```
# install.packages("Lahman")
library(Lahman)
data("Teams")
data("People")
data("Batting")
```

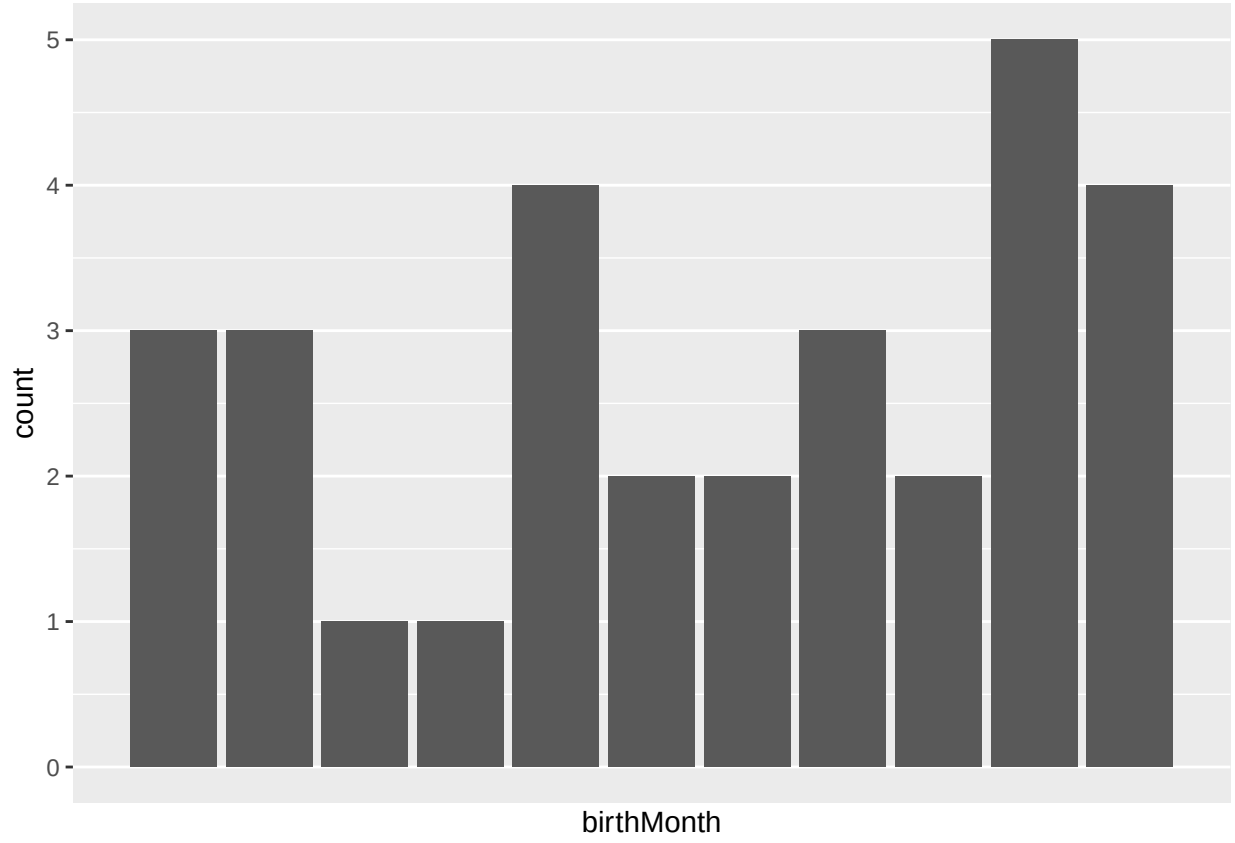
Part A: Use the `group_by()` and `summarize()` functions to find the total number of home runs for each player in the `Batting` data fame. Then, store the top 30 players (with the most career home runs) in a separate data frame. *Hint:* While there are a number of ways to select the top 30 players, the `dplyr` function `slice_head()` might be useful (`?slice_head`)

Part B: It has been hypothesized in several sports that an athlete's birth month is related to future success in sports. Using your data from Part A, join the birth month information from the `People` data frame. Then create a data visualization exploring whether birth months appear to be uniformly distributed among the players. Be sure that every month is represented on your axis, even if no players have a birthday in that month.

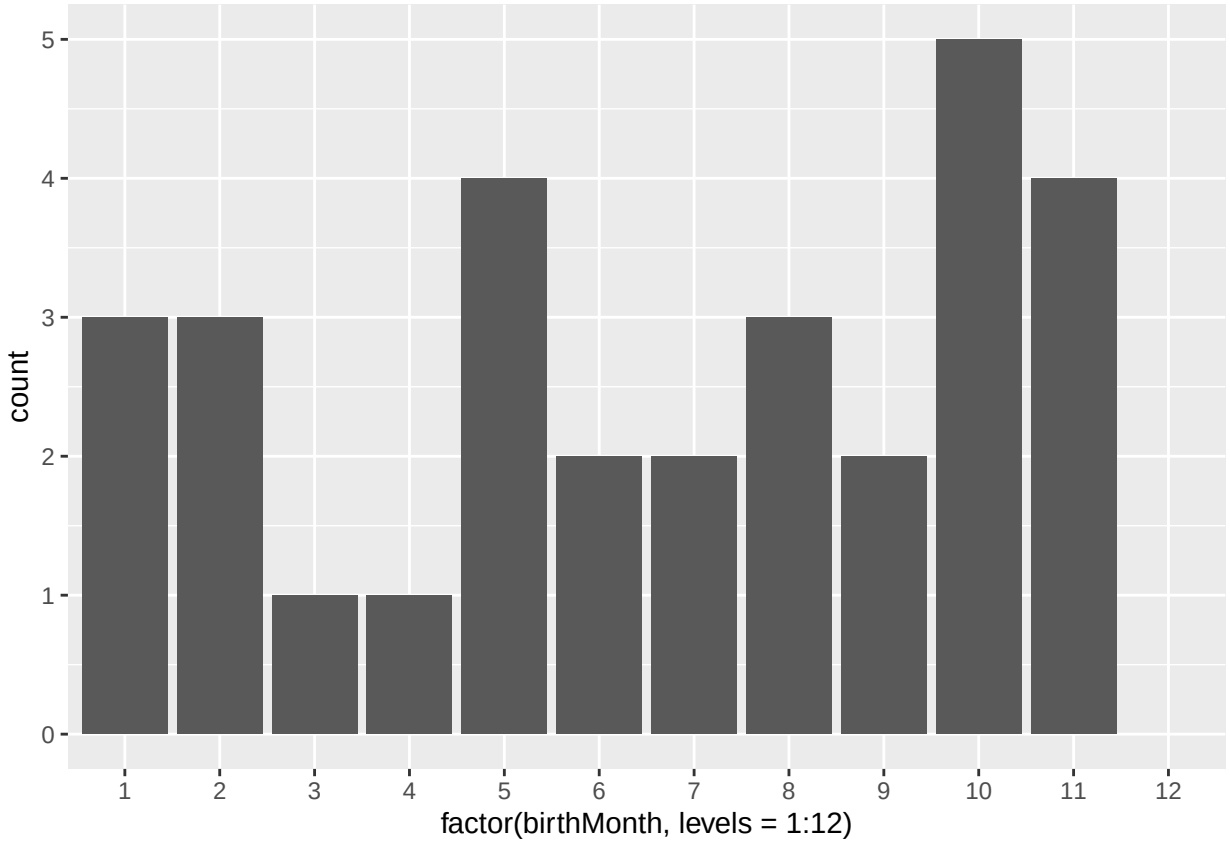
Solution

```r
## Part A
bat_30 <- Batting %>%
  group_by(playerID) %>%
  summarize(totalHR = sum(HR)) %>%
  arrange(desc(totalHR)) %>% slice_head(n = 30)

## Part B
bat_30 <- left_join(bat_30, People, by = "playerID")
ggplot(bat_30, aes(x = birthMonth)) + geom_bar() +
  scale_x_discrete(drop = FALSE)
```




```r
ggplot(bat_30, aes(x = factor(birthMonth, levels = 1:12))) + geom_bar() +
  scale_x_discrete(drop = FALSE)
```



Question 3

Using the `Teams` data frame in the `Lahman` package, display the top ten teams in terms of "slugging percentage" (SLG) since 1969.

SLG is computed as the team's total bases divided by the total "at bats" (`AB` in the data set). To find the total number of bases, you should assign a value of 1 for singles, 2 for doubles, 3 for triples, and 4 for home runs (that is, the sum of all of these will give you the total number of bases).

Hint: The variables `X2B`, `X3B`, and `HR` represent doubles, triples, and home runs, respectively. There is no variable for singles, but one can be computed using the variable `H` which represents the total number of hits. If we subtract the total number of doubles, triples, and home runs from the hits, we will be left with the total number of singles.

Sample output of *only the first three teams* is printed below to help validate your own solutions:

```
##   yearID teamID     SLG
## 1   2023    ATL 0.50080
## 2   2019    HOU 0.49546
## 3   2019    MIN 0.49407
```

```
## Solution
tt <- Teams %>% select(teamID, yearID, H, X2B, X3B, HR, AB)
tt %>%
  filter(yearID > 1969) %>%
  mutate(X1B = H - X2B - X3B - HR,
         totalBase = X1B + 2*X2B + 3*X3B + 4*HR,
         SLG = totalBase / AB) %>%
```

```r
  select(yearID, teamID, SLG) %>% arrange(desc(SLG)) %>% head(n = 10)
```

```
##    yearID teamID     SLG
## 1    2023    ATL 0.50080
## 2    2019    HOU 0.49546
## 3    2019    MIN 0.49407
## 4    2003    BOS 0.49090
## 5    2019    NYA 0.48988
## 6    1997    SEA 0.48450
## 7    1994    CLE 0.48384
## 8    1996    SEA 0.48359
## 9    2001    COL 0.48295
## 10   2020    LAN 0.48286
```