STAT 201: Midterm 1 Extra Coding Practice 2

Possible solutions

In many cities, there is a bike-share system where people can rent bicycles by the hour. People can use the bikes by either registering as members and purchasing a pass at a discounted rate, or by being a casual rider and paying per trip. We have data from Washington D.C.'s bike-share system from June-August 2012. The data are loaded in and stored as `bikes`. Each case is one rental. The variables are as follows:

- `month`: month (6 = June, 7 = July, 8 = August)
- `day`: day of the month (1-30)
- `hour`: hour (0, 6-23, where 0 = 12:00am, 6 = 6:00am, ..., 23 = 11:00pm)
- `day_week`: day of the week
- `type`: whether the renter was a member or a non-member ("registered" or "casual")

0. Add libraries necessary for data wrangling, plotting, and making pretty tables.

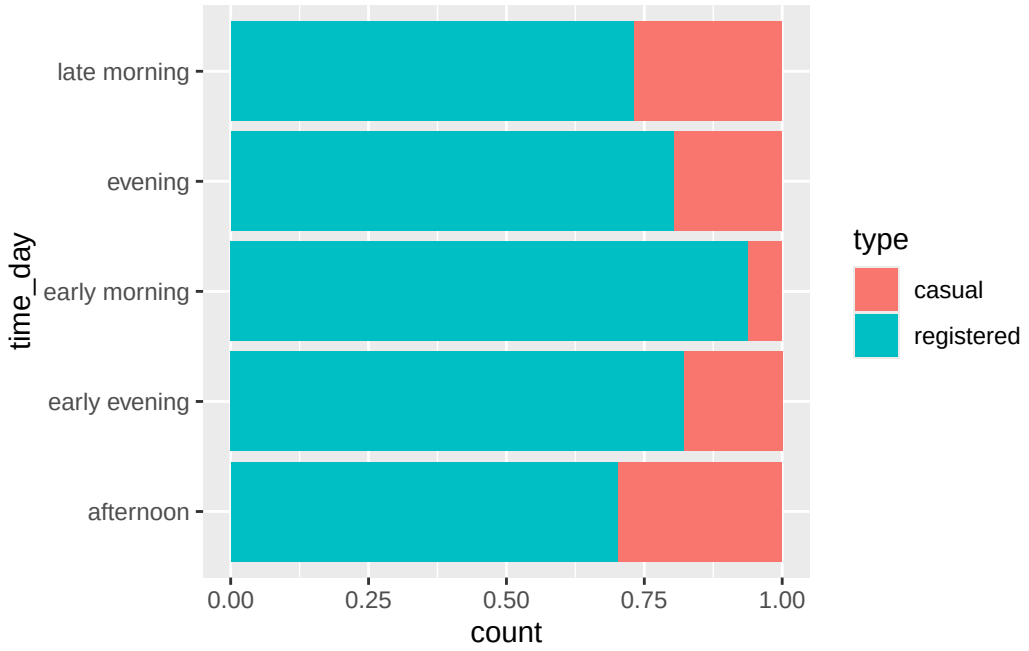
1. Modify the `bikes` data frame to include a variable that represents the time of day where:

    - Rentals from 6:00-8:00am are "early morning"
    - Rentals from 9:00am-12:00pm are "late morning"
    - Rentals from 1:00pm-4:00pm are "afternoon"
    - Rentals from 5:00pm-8:00pm are "early evening"
    - Rentals at any other time are "evening"

```
bikes <- bikes |>
  mutate(time_day = case_when(
    hour %in% 6:8 ~ "early morning",
    hour %in% 9:12 ~ "late morning",
    hour %in% 13:16 ~ "afternoon",
    hour %in% 17:20 ~ "early evening",
    T ~ "evening"
  ))
```

2. For rentals in 2011, do the time of day and the type of renter appear associated? Create an appropriate, well-labeled visualization and interpret it to answer the question.

```
bikes |>
  ggplot(aes(y = time_day, fill = type)) +
  geom_bar(position = "fill")
```



3. Create a beautiful table that displays the proportion of rentals by registered users for each day of the week, displayed in order of highest to lowest. Your table should only retain the day of the week and the proportion of registered users. What do you notice?

```
bikes |>
  group_by(day_week) |>
  count(type) |>
  mutate(prop = n/sum(n)) |>
  ungroup() |>
  filter(type == "registered") |>
  select(-type, -n) |>
  arrange(-prop) |>
  kable()
```

day_week	prop
Tue	0.8542034
Thu	0.8447327
Mon	0.8438705

day_week	prop
Wed	0.8331428
Fri	0.8165413
Sun	0.6742984
Sat	0.6579029

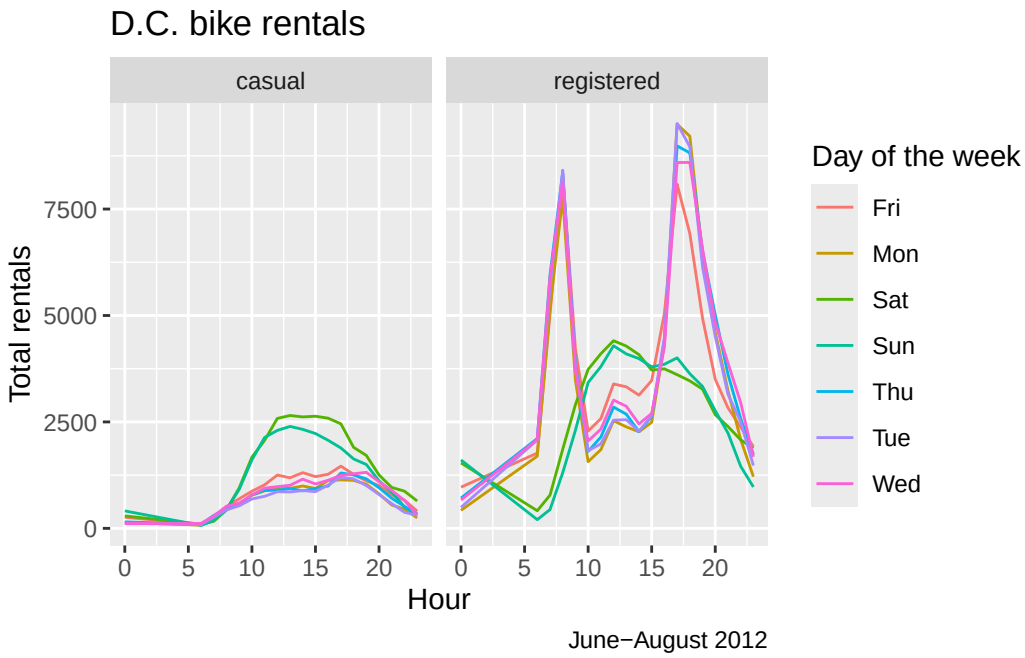
4. Create a beautiful summary table that displays the mean and standard deviation of daily number of bike rentals in June 2012.

```
bikes |>
  filter(month == 6) |>
  group_by(day) |>
  count() |>
  ungroup() |>
  summarise(avg = mean(n), sd = sd(n)) |>
  kable()
```

avg	sd
6624.167	934.0287

5. Re-create the graph seen here.

```
bikes |>
  group_by(type, hour, day_week) |>
  count(type) |>
  ggplot(aes(x = hour, y = n, col = day_week)) +
  geom_line() +
  facet_wrap(~ type) +
  labs(x = "Hour", y = "Total rentals", colour = "Day of the week",
       title = "D.C. bike rentals",
       caption = "June-August 2012")
```


D.C. bike rentals
casual
registered
Total rentals
7500
5000
2500
0
0
5
10
15
20
Hour
June−August 2012
Day of the week
Fri
Mon
Sat
Sun
Thu
Tue
Wed