YOUR TITLE HERE

STAT 231: Calendar Query

Sebastian Montesinos

Last updated March 4, 2022

Introduction

Data collection

I coded my data into three broad categories: 'work', for anything related to school (ie. class and homework), 'personal' for anything related to recreation (ie. gaming, seeing friends) and 'extracurricular' for any work outside of class (ie. job applications, clubs, independent projects). I decided to code the work category as four subcategories corresponding to my four classes: data science, oceanography, religion, and my thesis. I decided not to further divide the other two categories since my questions did not require doing so. So, in the end, I had six ways I marked off time on my calendar: 4 for my classes, 1 for personal time, and 1 for extracurricular. The units for all of these categories were in time, specifically minutes spent on each activity, which I used google calendar to code in.

```
# Data import and preliminary wrangling
calendar_data <- "SMontesinosCalendarQuery.ics" %>%
  ## Use ical package to import into R
  ical_parse_df() %>%
  ## Convert to "tibble" data frame format
  as_tibble() %>%
  ## calendar event descriptions are in a variable called "summary"
  ## "activity" is a more relevant/informative variable name
  rename(activity = summary) %>%
  mutate(
    ## Specify time zone (defaults to UTC otherwise)
    start_datetime = with_tz(start, tzone = "America/New_York"),
    end_datetime = with_tz(end, tzone = "America/New_York"),
    ## Compute duration of each activity in hours
    ## Feel free to use minutes instead
    duration = interval(start_datetime, end_datetime) / hours(1),
    ## Convert text to lower case and trim spaces to help clean up
    ## potential inconsistencies in formatting
    activity = str_to_lower(activity),
    ## separate date from time
    date = floor_date(start_datetime, unit = "day"),
    ## Examples of ways to parse dates, times (keep only what you need!)
    year = year(date),
    month = month(date, label = FALSE),
    day = day(date),
```

```
    day_of_week = wday(date, label = TRUE),
    day_of_year = yday(date)) %>%
  ## remove spurious year (added to every Google calendar)
  filter(year != 1969) %>%
  ## Turning the date variable into a date type
  mutate(date = ymd(date)) %>%
  ## Including only dates after I started collecting for the project
  filter(year >= 2022 & month >= 2 & day >= 17) %>%
  ##Removing trailing whitespace
  mutate(activity = str_trim(activity)) %>%
  ##Replacing spaces with underscores in activity names
  mutate(activity = str_replace(activity, " ", "_")) %>%
  ##Creating another column that records whether the observation was made on the weekday or weekend
  mutate(week_status = case_when(day_of_week == "Sat" | day_of_week == "Sun" ~ "Weekend",
                                 TRUE ~ "Weekday"))
```

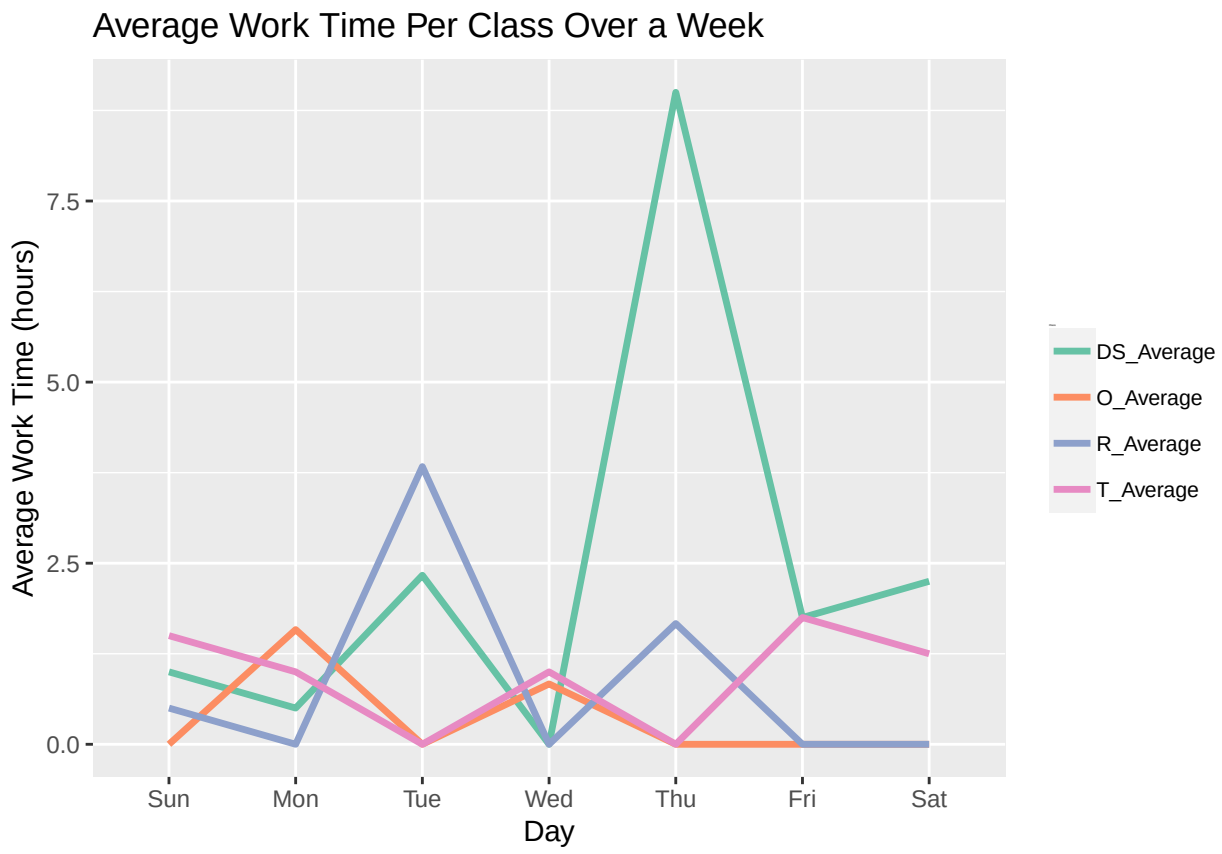
```
# Example of preparing dataset for first visualization

# Computing total duration for each activity per day of the week
activities_total <- calendar_data %>%
  group_by(date, day_of_week, activity) %>%
  summarize(duration = sum(duration),
#Including number of total observations
            ) %>%
  pivot_wider(names_from = activity, values_from = duration)

#Filling in
activities_total[is.na(activities_total)] = 0

activities_average <- activities_total %>%
  group_by(day_of_week) %>%
  summarise(DS_Average = mean(data_science),
            O_Average = mean(oceanography),
            R_Average = mean(religion),
            T_Average = mean(thesis)) %>%
  pivot_longer(-day_of_week,
               names_to = "Activity",
               values_to = "Duration")
```

```
# Preparing dataset for second visualization


activities_comparison <- calendar_data %>%
#Selecting relevant variables
  select(day, activity, duration, week_status) %>%
# Adding a unique row identifier so I can pivot
  mutate(row = row_number()) %>%
# Pivoting wider to get each activity by day
  pivot_wider(names_from = activity, values_from = duration) %>%
# Dropping row identifier
  select(-row)
#replacing NA values with Os
activities_comparison[is.na(activities_comparison)] = 0
```

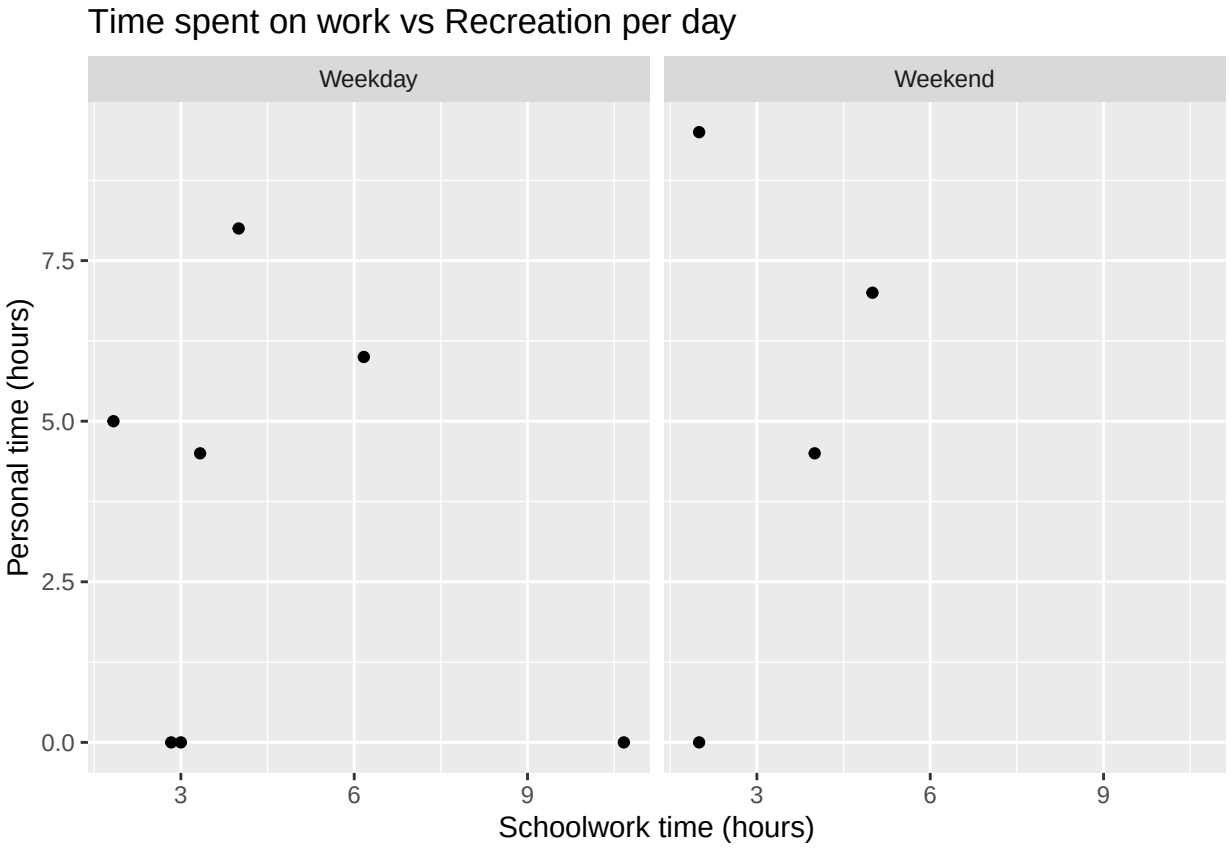
```r
# Grouping by day and summing the activity duration into categories I want to compare
activities_comparison2 <- activities_comparison %>%
  group_by(day, week_status) %>%
# Calculating total school work time, personal time, and extracurricular time
  summarise(work_time = sum(thesis, oceanography, religion, data_science), personal_time = sum(personal
#
```

Results

```r
# Code for first data visualization
# Be sure to provide meaningful title and axes labels and
# resize figure appropriately
# Only code for your first visualization should be here (no or very minimal wrangling code)
# Remove all these comments!
p <- ggplot(data = activities_average,
            aes(x=day_of_week,
                y = Duration,
                color = Activity,
                group = Activity)) +
  geom_line(size = 1.2, alpha = 1) +
  scale_color_discrete(labels = c("Data Science",
                                  "Oceanography",
                                  "Apocalyptic Religion",
                                  "Thesis")) +
  scale_color_brewer(type = "qual",
                     palette = 7) +
  labs(DS_Average = "Data Science",
       title = "Average Work Time Per Class Over a Week",
       x = "Day",
       y = "Average Work Time (hours)",
       color = "Class") +
  theme(legend.text = element_text(size = 8)) +
  theme(legend.title = element_text(size = 1))

p
```




```
#Plotting total work time against personal time per day
n <- ggplot(data = activities_comparison2, aes(x= work_time, y = personal_time)) +
#Creating 1 plot for weekends and one for weekdays
  facet_wrap(~week_status, nrow = 1) +
#Using points to represent each day
  geom_point() +
#Labeling the graph
  labs(title = "Time spent on work vs Recreation per day",
       y = "Personal time (hours)",
       x = "Schoolwork time (hours)")
n
```




```
# Code for table
# Only code for your table should be here (no or very minimal wrangling code)
```

Conclusions

Reflection