The Working Life: An Analysis of my Time Spent Working

SAT 231: Calendar Query

Andrew Leung

September 30, 2022

Introduction

Simply put, despite attending Amherst for over four years now, I've always felt like I spend too much of my time on work. So, I wanted to put that curiosity to the test. Over the past two weeks, I wanted to answer three main questions:

1. Do I spend more time than I think I do on coursework?
2. What kind of work do I spend the most time on? For which classes?
3. How much time do I spend on coursework per day?

By the end of this study, I hope I can have a data-backed way to improve my work ethic at Amherst for the little time I have left here. As the saying goes, the best time to plant a tree was 20 years ago. The second best time is today.

Methods

Data collection

Before starting each assignment, I would create a Google Calendar entry that was equal to the amount of time I think the assignment would take me to complete. It does not matter if this time entry overlaps with other events in my calendar as I am only concerned with my total estimate. Once I start working, I start a timer using the Toggl app.

Each assignment in Google Calendar and Toggl get the same naming convention: `[class]_[assignment_type]_[assignment_n`

- `class` = The major the class belongs to.
  - ARCH = Architectural Studies
  - ENST = Environmental Studies
  - PHIL = Philosophy
  - STAT = Statistics
- `assignment_type` = One of three kinds of assignments.

  - Reading
  - Writing
  - PSet, or problem set

- `assignment_number` = Keeps track of what number assignment this is of a certain type for a certain class.
  - For example, `STAT_PSet_3` would mean represent the 3rd problem set I've done for Statistics.
- `actual/estimate` = Helps to differentiate time entries in a table. "Actuals" were in Toggl and "Estimates" were in Google Calendar.

This naming convention is important for when I stop working, but do not finish the assignment. For example, if I work for 1 hour, get lunch, and then work for another 30 minutes later in the day on the same assignment, Toggl would keep record of a 1-hr time entry and a 30-min time entry with the same name.

Data wrangling

To start, I used **ical** to import my ICS file. I relied mainly on **tidyverse**, **lubridate**, **glue**, and **janitor** to wrangle data and clean up variable names.

To prepare my first visualization comparing my work time estimates and my actual time spent, I will first need to separate my naming convention (`[class]_[assignment_type]_[assignment_number]_[actual/estimate]`) into individual columns. Next, since I currently have more actual time entries than estimates, I will need to sum the time it took me to finish each assignment. Ideally, the number of sums should equal the number of estimates I have (since I only estimate the time it takes to finish an assignment once). Lastly, given the two tables have the same number of rows, I will join the two tables together by `class`, `assignment_type`, and `assignment_number`.

To create my second visualization looking at the distribution of how long each type of work takes and for each class, I will use the table of summed actual times created earlier and pare it down to only `class`, `assignment_type`, and the duration of each assignment.

For my last visualization looking at the amount of time I spend on work each day, I will simply sum the time spent on work by date.

Toggle Import

```
# toggl Import
toggl_import <- read_csv("data/Toggl_time_entries_2022-09-11_to_2022-09-26.csv")

# Wrangling
work_time_act <- toggl_import %>%
  # The toggle names by default are very difficult to work with, so I will
  # clean them up first using a function from janitor.
  clean_names() %>%
  mutate(
```

```
        # Because of my naming convention, I have to parse out each category
        # into its own column. To get the class, I will separate the first
        # 4 characters of the string.
        class = substr(description, 1, 4),
        # Next, I will extract the string in between the underscores
        # to get the assignment type.
        type = str_extract(description, "_(.*?)_"),
        # The extracted text still has underscores surrounding it. So, I will
        # replace them with empty strings to remove them.
        assignment_type = str_replace_all(type, "_", ""),
        # Lastly, I will extract the assignment number from the string.
        assignment_number = str_extract(description, "(\\d)"),
        # Create a date-time to calculate duration
        start = ymd_hms(glue("{start_date} {start_time}"),
                        tz = "America/New_York"),
        end = ymd_hms(glue("{end_date} {end_time}"),
                      tz = "America/New_York"),
        # Calculate duration in hours
        duration_hours_act = interval(start, end) / hours(1)) %>%
  # Remove unnecessary columns
  select(-c(user,
            email,
            client,
            project,
            task,
            billable,
            tags,
            amount_usd,
            type))
```

Google Calendar Import

```
# GCal data import
gcal_import <- ical_parse_df("data/Calendar Query_c_bc73343e356513f96921415dd15560074fa93f48810f255eac9

# Wrangling!
work_time_est <- gcal_import %>%
  mutate(
    # I will be going through the same process as I did with the toggl data
    # to separate the string into each variable column.
    class = substr(summary, 1, 4),
    # This Regex removes any value between two underscores
    type = str_extract(summary, "_(.*?)_"),
```

```
    # This gets rid of the underscores left over after the extraction
    assignment_type = str_replace_all(type, "_", ""),
    # This regex extracts any number in the string
    assignment_number = str_extract(summary, "(\\d)"),
    # Calculate duration of work time estimates between start and end times
    duration_hours_est = interval(start, end) / hours(1)) %>%
  # Removed unnecesary columns
  select(-c(uid,
            description,
            last.modified,
            status,
            type)) %>%
  clean_names()
```

Prep for actual vs estimate scatter plot

```
# To prepare the data for my scatter plot of estimated assignment time vs
# actual assignment time, I will first add how long each assignment took me.
actual_time_spent <- work_time_act %>%
  group_by(description,
           class,
           assignment_type,
           assignment_number) %>%
  summarise(
    N = n(),
    hour_per_assignment = sum(duration_hours_act))

# Next, I will join together this newly made table with a pared down
# table of my estimated work time
actual_vs_estimate <- work_time_est %>%
  select(c(class,
           assignment_type,
           assignment_number,
           duration_hours_est)) %>%
  right_join(actual_time_spent,
             class = class,
             assignment_type = assignment_type,
             assignment_number = assignment_number)

# Calculating a line of best fit for plot
model <- lm(hour_per_assignment ~ duration_hours_est,
            data = actual_vs_estimate)
```

Prep for box plot visualization of time spent by work type and class

```
time_spent_by_class <- work_time_act %>%
  select(c(class,
           assignment_type,
           assignment_number,
           duration_hours_act)) %>%
  group_by(class,
           assignment_type,
           assignment_number) %>%
  summarise(
    N = n(),
    total_hours = sum(duration_hours_act)
  )
```

Prep for time-series plot of time spent per day

```
work_time_per_day <- work_time_act %>%
  # Sum the amount of time spent on work each day
  group_by(start_date) %>%
  summarise(
    N = n(),
    hours_per_day = sum(duration_hours_act))
```

Prep for summary table

```
# Small summary table 1 of actual times by assignment
assignment_summaries_act <- work_time_act %>%
  group_by(assignment_type) %>%
  summarise(
    N = n(),
    total_hours = round(sum(duration_hours_act), 2),
    min = round(min(duration_hours_act), 2),
    median = round(median(duration_hours_act), 2),
    max = round(max(duration_hours_act), 2),
    sd = round(sd(duration_hours_act), 2)
  ) %>%
  rename(observations = N,
         class = assignment_type) %>%
  # act_or_est will only be used to join the 4 tables together.
  mutate(act_or_est = "Actual")
```

```r
# Small summary table 2 of actual times by class
class_summaries_act <- work_time_act %>%
  group_by(class) %>%
  summarise(
    N = n(),
    total_hours = round(sum(duration_hours_act), 2),
    min = round(min(duration_hours_act), 2),
    median = round(median(duration_hours_act), 2),
    max = round(max(duration_hours_act), 2),
    sd = round(sd(duration_hours_act), 2)
  ) %>%
  rename(observations = N) %>%
  mutate(act_or_est = "Actual")

# Small summary table 3 of estimated times by class
assignment_summaries_est <- work_time_est %>%
  group_by(assignment_type) %>%
  summarise(
    N = n(),
    total_hours = round(sum(duration_hours_est), 2),
    min = round(min(duration_hours_est), 2),
    median = round(median(duration_hours_est), 2),
    max = round(max(duration_hours_est), 2),
    sd = round(sd(duration_hours_est), 2)
  ) %>%
  rename(observations = N,
         class = assignment_type) %>%
  mutate(act_or_est = "Estimate")

# Small summary table 4 of estimated times by assignment
class_summaries_est <- work_time_est %>%
  group_by(class) %>%
  summarise(
    N = n(),
    total_hours = round(sum(duration_hours_est), 2),
    min = round(min(duration_hours_est), 2),
    median = round(median(duration_hours_est), 2),
    max = round(max(duration_hours_est), 2),
    sd = round(sd(duration_hours_est), 2)
  ) %>%
  rename(observations = N) %>%
  mutate(act_or_est = "Estimate")

# Combine the summary tables of actual times by row
```

```r
coursework_summary_act <- class_summaries_act %>%
  bind_rows(assignment_summaries_act) %>%
  ungroup() %>%
  rename(variable = class)

# Combine the summary tables of estimated times by row
coursework_summary_est <- class_summaries_est %>%
  bind_rows(assignment_summaries_est) %>%
  ungroup() %>%
  rename(variable = class)

# Combine the two larger summary tables into one large one
coursework_summary <- coursework_summary_act %>%
  bind_rows(coursework_summary_est)

# Reorder the rows
coursework_summary <- coursework_summary %>%
  select(act_or_est,
         variable,
         observations,
         total_hours,
         min,
         median,
         max,
         sd) %>%
  ungroup()

# Add indent headers for kable table
coursework_summary <- coursework_summary %>%
  add_row(tibble(variable = "Actual"),
          .before = 1) %>%
  add_row(tibble(variable = "Class"),
          .before = 2) %>%
  add_row(tibble(variable = "Assignment Type"),
          .before = 7) %>%
  add_row(tibble(variable = "Estimate"),
          .before = 11) %>%
  add_row(tibble(variable = "Class"),
          .before = 12) %>%
  add_row(tibble(variable = "Assignment Type"),
          .before = 17) %>%
  # This column was just to bind the two larger summary tables together,
  # so it can be omitted now.
  select(-c(act_or_est))
```

Statistical methods

I used the features in **ggplot2** to create all of my visualizations. To calculate the statistics for my summary table and for some of my visualizations, I used the **stats** package.

To visualize how my time estimates compare to how long I actually work, I created a scatter plot that plots each assignment by how long I thought it would take and how long it actually took. To visualize how correlated my data is, I calculated a line of best fit and drew it on the plot next to the line $y = x $, which represents the ideal scenario. (The ideal would be if I estimated my work to take 1 hour, I would actually take 1 hour.) I also calculated the correlation coefficient and drew it on my plot to numerically visualize the relationship between the variables. I thought these three pieces together (scatter plot, line of best fit vs y = x, and correlation coefficient) best answered if I spend more time than I think on my work as it shows the relationship between my estimates and actual time spent both visually and mathematically.

To visualize how long each type of work takes me for each class, I created a box plot with a jitter plot overlay. The box plot represents the distribution of how long assignments for each class took me to complete. The jitter plot represents the distribution of how long each type of assignment from each class took me to complete. The combination of the box plot and the jitter plot not only shows how long assignments take by type and by class, but it also allows me to visualize how consistently I work on various assignments for various classes.

To visualize how much time I spend on work per day, I created a line graph with a scatter plot overlay. Each point represents the total amount of hours I worked in a day. To add more context to the graph, I drew an orange overlay between two dates to represent the time I was in quarantine due to a positive COVID-19 test. This plot helped me not only derive a clear answer to the question of how much work I spend per day, but also trends on how much I work throughout the week.

Results

Figure 1 shows the relationship between how much time I thought an assignment would take and how long it actually took me. The graph shows that there is a moderately positive linear relationship ($r = 0.708$) between my estimates and my actual time spent on work. Based on the distribution of points relative to the regression line, it looks like I overestimate more than I underestimate; however when I do underestimate, I underestimate by a large margin. Furthermore, at a glance, it seems as though I tend to predict how long work for my environmental studies and architectural studies class will take fairly well, but am less accurate when it comes to estimating work for statistics or philosophy.

```
# How correlated is my data?
r <- round(
  cor(actual_vs_estimate$hour_per_assignment,
      actual_vs_estimate$duration_hours_est),
  3)

# Create visualization of my estimation vs my actual time spent doing work
# with the line of best fit and the line y = x
g <- ggplot(actual_vs_estimate,
            aes(x = duration_hours_est,
```

```
                y = hour_per_assignment)) +
  geom_point(aes(color = class),
             size = 3) +
  # Colorblind friendly palette
  scale_color_manual(values = c("ARCH" = "#E69F00",
                                "ENST" = "#56B4E9",
                                "PHIL" = "#CC79A7",
                                "STAT" = "#999999")) +
  # Drawing the line of best fit
  geom_abline(intercept = -0.1950,
              slope = 1.2066,
              color = "blue",
              size = 1) +
  # Drawing the line y = x
  geom_abline(intercept = 0,
              slope = 1,
              linetype = "dashed") +
  labs(color = "Class") +
  # r value label
  annotate("text",
           x = 6,
           y = .5,
           label = paste(c("Correlation = ", r),
                         collapse = "")) +
  # y = x label
  annotate("text",
           x = 7,
           y = 3.5,
           label = "y = x") +
  # Arrow for line dashed line y = x
  geom_segment(aes(x = 7, y = 4,
                   xend = 7, yend = 7),
               arrow = arrow(length = unit(0.25, "cm"))) +
  # Arrow for line of best fit
  geom_segment(aes(x = 6, y = 1,
                   xend = 6, yend = 7),
               arrow = arrow(length = unit(0.25, "cm"))) +
  # Changing the bounds to better view the intercepts
  xlim(0, 8) +
  ylim(-2.5, 10) +
  labs(title = "Actual vs Estimated Time Spent on Coursework",
       x = "Estimated Time Spent (Hrs)",
       y = "Actual Time Spent (Hrs)")

g
```

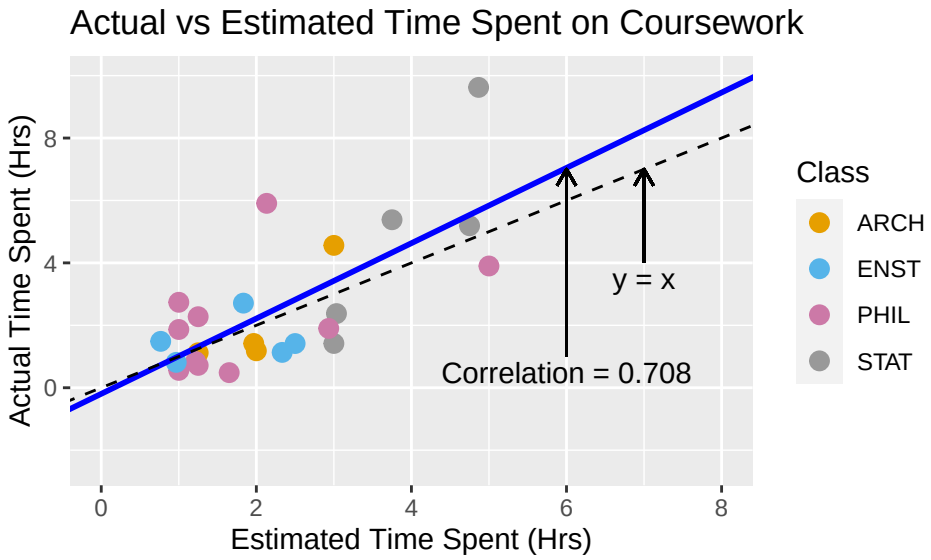

Figure 1: The relationship bewteen how much time I though an assignment would take to complete and how long it actually took to complete. The blue line represents the line of best fit, while the dashed line represents y = x, or the line that would predict if all my estimates were exactly correct. Each point represents an assignment and is colorcoded by class.

Figure 2 corroborates my findings from figure 1. The amount of time assignments took in environmental studies and architectural studies were fairly consistent, as shown by the relatively short boxes and short whiskers. For philosophy and statistics, on the other hand, assignment times varied much more widely. Not only was the average time for me to complete a philosophy assignment longer than ENST or ARCH, with about half of the assignments taking 2 or more hours, but philosophy also assigned the most amount of work in the two-week period, double the second most at 10 assignments (see Table 1). Statistics assignments took by far the longest, with a median time spent of over 5 hours. However, this is to be expected when problem sets are expected to be done over a school week rather than one or two days.

In general, writing assignments took longer on average than readings, which makes sense given writing assignments need more original thought than reading. A notable exception here was in my philosophy class. The pieces we read for this class often have very dense and complex arguments, so understanding them often takes longer than other writing assignments. Problem sets took me the longest on average out of any assignment, which also makes sense, since we are allotted five days to complete it.

```
bp <- ggplot(time_spent_by_class,
             aes(x = class,
                 y = total_hours,
                 color = class)) +
  geom_boxplot() +
  # Colorblind friendly palette
  scale_color_manual(values = c("ARCH" = "#E69F00",
                                "ENST" = "#56B4E9",
                                "PHIL" = "#CC79A7",
```

```
                                  "STAT" = "#999999")) +
  # Overlay a jitter plot of the assignments for each class
  geom_jitter(aes(shape = assignment_type,
                  color = "black")) +
  scale_shape_manual(values = c(0:2)) +
  labs(title = "Actual Time Spent on Coursework",
       subtitle = "By Class and Assignment Type",
       x = "Class Type",
       y = "Actual Time Spent (Hrs)",
       color = "Class",
       shape = "Assignment Type")

bp
```

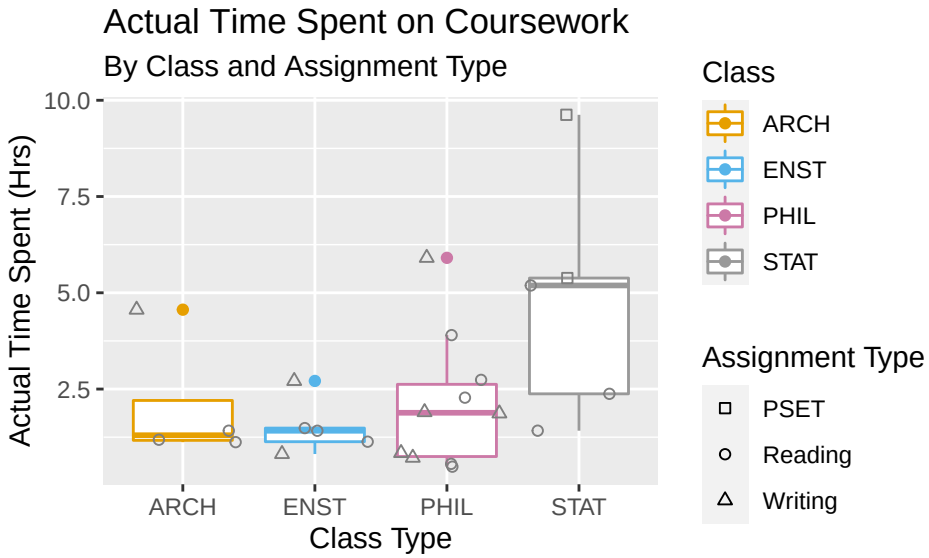

Figure 2: A box plot displaying the distribution of the amount of time it took me to complete assignments separated by class. There is a jitter plot overlay that shows each assignment by its type.

Figure 3 shows how long I worked in total on assignments per day. Also marked on the graph in orange is the time period when I contract COVID. Interestingly, I tend to work the most on the weekends (9/11, 9/17, 9/18, and 9/25), likely because I not only have more time in the day, but also because I have assignments due every Monday. Furthermore, the amount of work I do generally decreases throughout the week until the weekend, most clearly seen from 9/11 - 9/16. I believe this trend would've been clearer had I not gotten COVID. Because I was in my room all day during quarantine, there was little else to do throughout the day other than work. My excitement over being released from quarantine was best exemplified in the 4.5 hour drop in work time on the day I was released.

```
g <- ggplot(work_time_per_day,
            aes(x = start_date,
                y = hours_per_day)) +
  annotate("rect",
```

```
           fill = "orange",
           alpha = 0.3,
           xmin = as_date("2022-09-16"), xmax = as_date("2022-09-21"),
           ymin = -Inf, ymax = Inf) +
  geom_line() +
  geom_point() +
  labs(title = "Time Spent on Coursework Per Day",
       subtitle = "9/11 - 9/26",
       x = "Date",
       y = "Duration (Hours)") +
  scale_x_date(date_breaks = "2 days") +
  annotate("text",
           x = as_date("2022-09-18"),
           y = 4,
           label = "In Quarantine",
           hjust = 0.3)

g
```

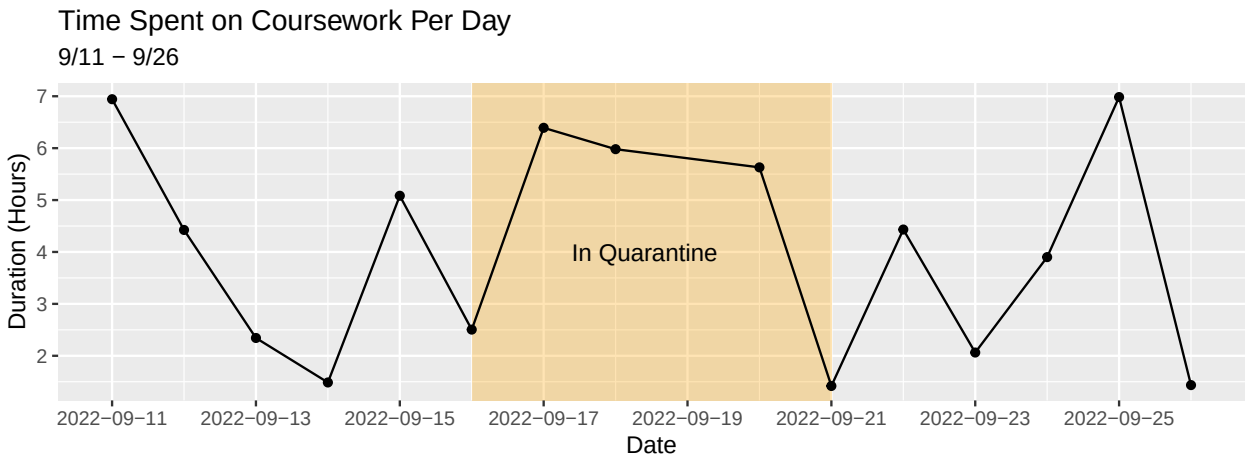

Figure 3: A time-series graph displaying how much time I spent on work per day. The highlighted area represents the time I was in quarantine for a positive COVID-19 result.

```
kable(coursework_summary,
      col.names = c("Variable",
                    "Observations",
                    "Total",
                    "Minimum",
                    "Median",
                    "Maximum",
                    "SD"),
      caption = "Summary of Time Spent on Work (Actual and Estimate)",
      align = "lcc",
```

```
    booktabs = TRUE) %>%
kable_styling(latex_options = "HOLD_position") %>%
add_header_above(header = c(" " = 2,
                            "Hours Worked " = 5),
                 bold = TRUE) %>%
# Adding three levels of indentation for section headers
add_indent(c(1, 11)) %>%
add_indent(c(2, 7, 12, 17),
           level_of_indent = 2) %>%
add_indent(c(3:6, 8:10, 13:16, 18:20),
           level_of_indent = 3) %>%
row_spec(c(1, 2, 7, 11, 12, 17),
         bold = TRUE)
```

Table 1: Summary of Time Spent on Work (Actual and Estimate)

		Hours Worked				
Variable	Observations	Total	Minimum	Median	Maximum	SD
Actual						
Class						
ARCH	5	8.28	1.12	1.42	2.49	0.60
ENST	7	7.56	0.45	1.08	1.63	0.45
PHIL	12	21.19	0.48	1.65	3.90	1.20
STAT	13	23.99	0.50	1.42	4.43	1.28
Assignment Type						
PSET	6	15.01	0.69	2.37	4.43	1.59
Reading	20	26.70	0.45	1.30	3.90	0.84
Writing	11	19.32	0.72	1.87	3.78	0.90
Estimate						
Class						
ARCH	4	8.22	1.25	1.98	3.00	0.72
ENST	5	8.40	0.77	1.83	2.50	0.79
PHIL	10	18.43	1.00	1.25	5.00	1.27
STAT	5	19.40	3.00	3.75	4.87	0.90
Assignment Type						
PSET	2	8.62	3.75	4.31	4.87	0.79
Reading	14	31.50	0.77	1.98	5.00	1.33
Writing	8	14.33	0.97	1.54	3.00	0.83

Conclusions

When it comes to estimating how long work will take me to complete, I am not the best. Although I tend to overestimate more than I underestimate, my underestimates are off by a much larger margin. This tends to be the case with problem sets and more complex readings especially, like the ones in my philosophy class. In the future, rather than jumping into assignments, I should preview them more and try to gauge its length and difficulty. This will help me be more realistic with the time it takes me to do work.

While problem sets take me the most amount of time to complete on average, reading assignments are my most frequently assigned piece of work, and thus comprise a majority of my work hours. Sometimes, long readings are unavoidable; however, in general, I should be more intentional about what I am reading rather than reading every word. For example, looking at the next class' theme could provide a helpful lens through which to appraoch the readings, making it more efficient and more effective.

The trends found in the amount of time I spend on work per day were somewhat troubling. Not only are they wildly volatile, but I also tend to spend the most amount of time working on the weekends. While it is understandable that weekends have the most free time, I would ideally like the treat the weekend as a time to relax, not a time to be productive. Of course there will be times where working on the weekends will be necessary, but working a majority of my hours on the weekend is not sustainable. Going forward, I should look to spread my work time across the week more equitably, so as to avoid devoting my weekends to work.

Reflection

I thought this process was incredibly rewarding, not only because of the data visualizations skills, but also what I learned about myself. In terms of the questions I posed in the beginning, I think I satisfied my curiosity. From a data planning standpoint, I thought estimating how long it would take me to finish work and then actually timing how long it took me to finish was a great. In retrospect, I would've chosen a different method of data collection for my estimates (a spreadsheet, perhaps); however, given the bounds of the assignment, I though Google Calendar was a perfectly fine collection method.

The first difficulty actually came before data collection even started during the proposal phase. Planning out the questions I was going to answer and the data collection methods I was going to use while trying to keep in mind how the data will look in R and how it would be reflected in various graphs was incredibly difficult. However, I am happy I did so, as it made this process much easier than if I had just winged it throughout the two weeks.

The second difficulty came during the collection. There was a logistical difficulty, since I wasn't used to recording my work times. Forgetting to stop the timer was a common occurrence. There was also a conceptual difficulty in the first couple of days with how I recorded my estimates. After the first day of data recording, I realized I was actually estimating how long each work session would take, not how long each assignment would take. It was more natural to record the former way in Google Calendar since each estimation was a block of time, which would sometimes bleed into other events. For example, a problem set would of course take more than an hour, but blocking off a 5 hours chunk in a calendar did not feel intuitive at first. Luckily, I caught the mistake sooner rather than later.

For future data collection and analysis projects, I need to keep in mind that planning is everything. The more time spent on thinking through a detailed research question, how data will be defined, how it will be collected, how it will look on a table, how it will look on a plot, and the conclusions that could be drawn from it, the easier everything will be later down the line.