Calender Assignment

Lorraine Oloo

3/16/2021

#Data wrangling

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ical)
library(readxl)
```

```
#calling the data set and setting start time an end time for my times
my_calendar0 <- ical_parse_df("C:/Users/Lorraine/Dropbox/Data Science/Data-Science/Calender/loloo23@amh
  mutate(start_datetime = with_tz(start, tzone = "America/New_York")
    , end_datetime = with_tz(end, tzone = "America/New_York")
    , length_hour = end_datetime - start_datetime
    , date = floor_date(start_datetime, unit = "day"))

my_calendar1 <- arrange(my_calendar0, date)
```

```
#the names in my calender
names(my_calendar1)
```

```
##  [1] "uid"            "summary"        "start"          "end"
##  [5] "description"    "last.modified"  "status"         "start_datetime"
##  [9] "end_datetime"   "length_hour"    "date"
```

```
#shows variables in a specific column
unique(my_calendar1$summary)
```

```
## [1] "R-DS"  "A-DS"  "R-MB"  "A-DS " "A-EC"  "A-EC " "A-MB"
```

```
#combining the variables in sunmmary
my_calendar1 <- mutate(my_calendar1, summary_correct = str_trim(summary))
unique(my_calendar1$summary_correct)
```

```
## [1] "R-DS" "A-DS" "R-MB" "A-EC" "A-MB"
```

```
#creating day of the week
my_calendar1 <- mutate(my_calendar1, day_of_week = weekdays(date))
```

```
#Distinguishing between assignment and reading and the specific courses

#Reading
my_calendar1 <- my_calendar1 %>%
  mutate(study0 = case_when(str_detect(summary_correct, "R") ~ "Reading"
                            ,TRUE ~ summary_correct))
#Assignment
my_calendar1 <- my_calendar1 %>%
  mutate(study1 = case_when(str_detect(study0, "A") ~ "Assignment"
                            ,TRUE ~ study0))
#Data Science
my_calendar1 <- my_calendar1 %>%
  mutate(course = case_when(str_detect(summary_correct, "D") ~ "Data Science"
                            ,TRUE ~ summary_correct))
#Econometrics
my_calendar1 <- my_calendar1 %>%
  mutate(course1 = case_when(str_detect(course, "E") ~ "Econometrics"
                            ,TRUE ~ course))
#Money and Banking
my_calendar1 <- my_calendar1 %>%
  mutate(course2 = case_when(str_detect(course1, "M") ~ "Money and Banking"
                            ,TRUE ~ course1))
```

```
#Weekday or weekend
my_calendar1 <- my_calendar1 %>%
  mutate(type_day = ifelse(day_of_week %in% c("Saturday", "Sunday"), "Weekend", "Weekday"))
```

```
#selecting variables I'll be working with
my_calendar2 <- my_calendar1 %>%
  select(summary_correct, date, day_of_week, type_day,course2, study1, length_hour)
```

```
social_media <- read_excel("Social Media.xlsx")
social_media1 <- arrange(social_media,date)
#remove PU from the data set
social_media1 <- filter(social_media1, summary_correct != "PU")
#Adding day of the week
social_media1 <- mutate(social_media1, day_of_week = weekdays(date))
#Weekday or weekend
social_media1 <- social_media1 %>%
  mutate(type_day = ifelse(day_of_week %in% c("Saturday", "Sunday"), "Weekend", "Weekday"))
```

(http://www.sthda.com/english/wiki/ggplot2-barplots-quick-start-guide-r-software-and-data-visualization#bar-plot-with-labels) (to find the summary of my graphs)

###Graphs for my studies

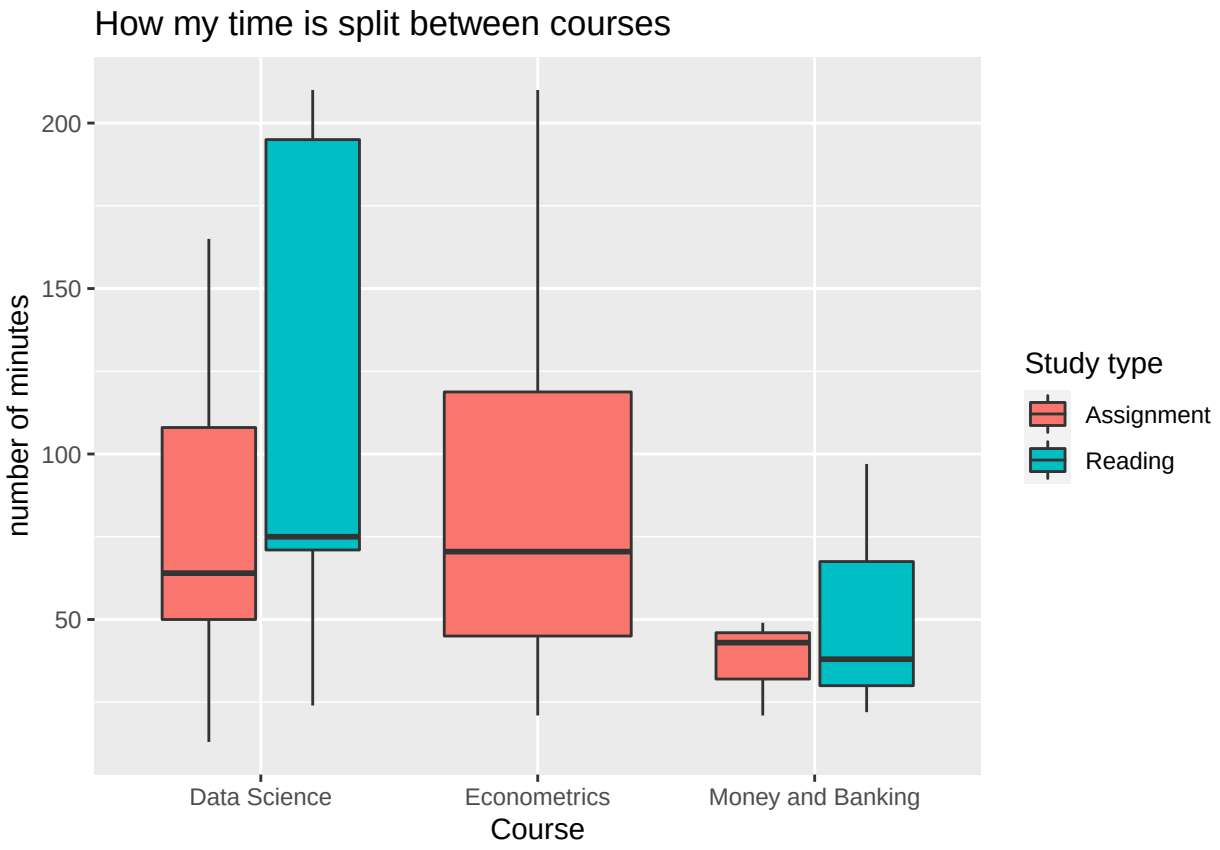
```
#A bar graph to show how all the time I spent studying was distributed between my three courses, in rea

# Stacked barplot with multiple groups
ggplot(data=my_calendar2, aes(x=course2, y=length_hour, fill=study1)) +  geom_bar(stat="identity") + lab
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

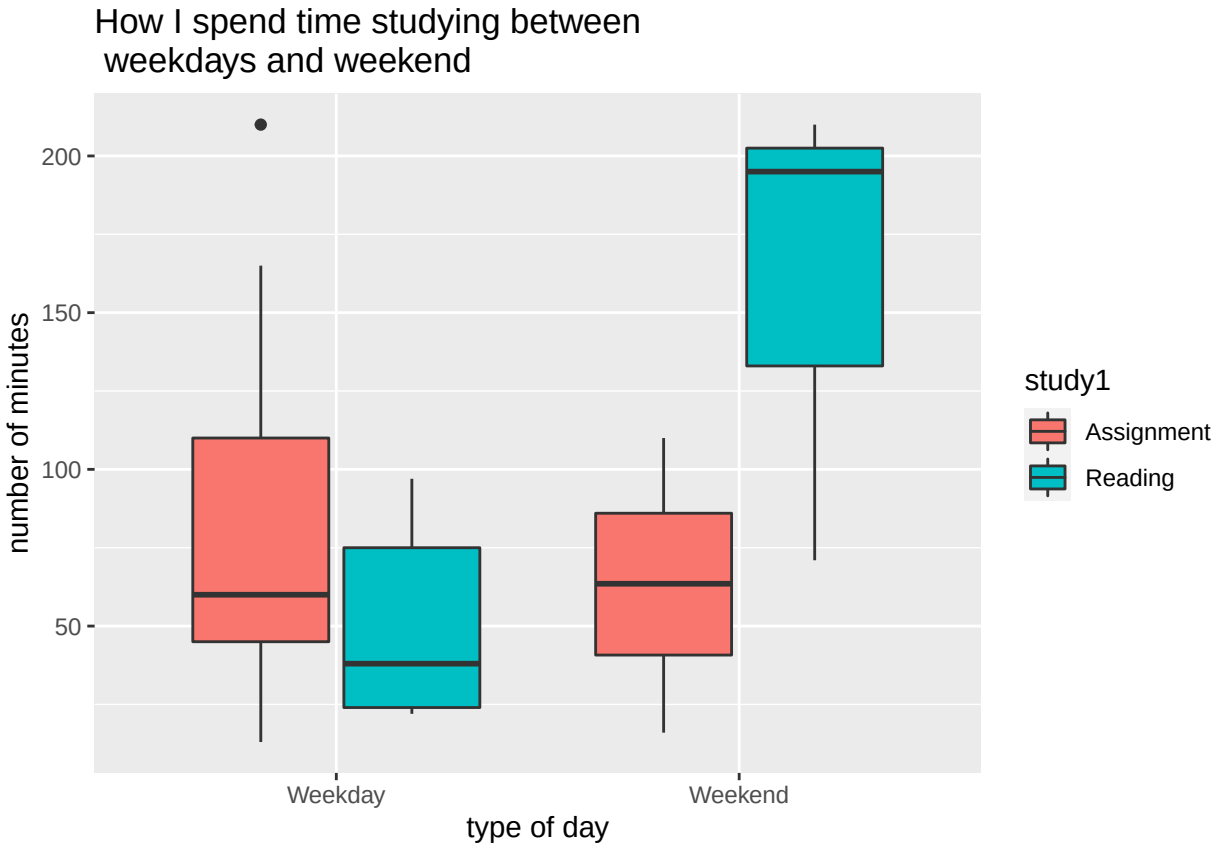

```
#Boxplot to show the distribution of time spent in courses
ggplot(data=my_calendar2, aes(x=course2, y=length_hour, fill=study1)) + geom_boxplot() + labs(title = "
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

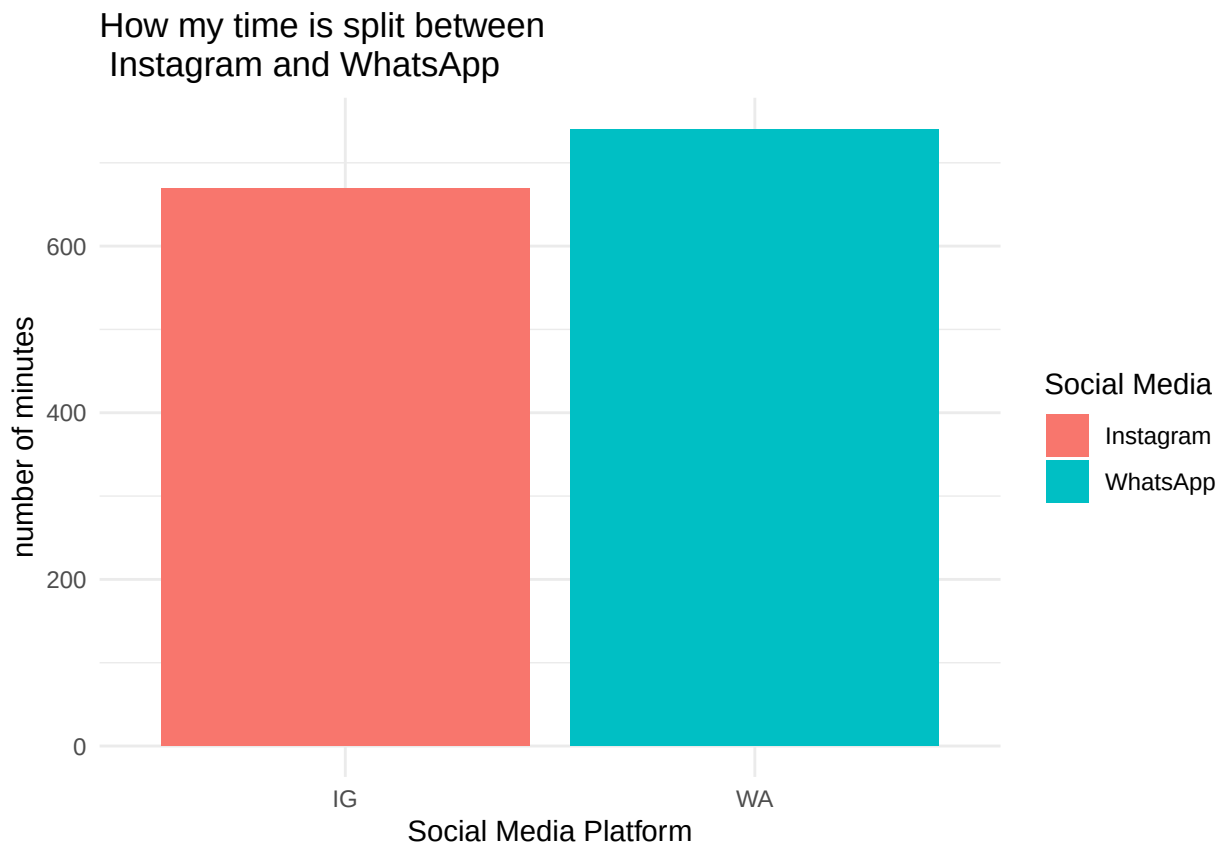



```
#weekday and weekend boxplot
ggplot(data=my_calendar2, aes(x=type_day, y=length_hour, fill=study1)) +  geom_boxplot() + labs(title =
```

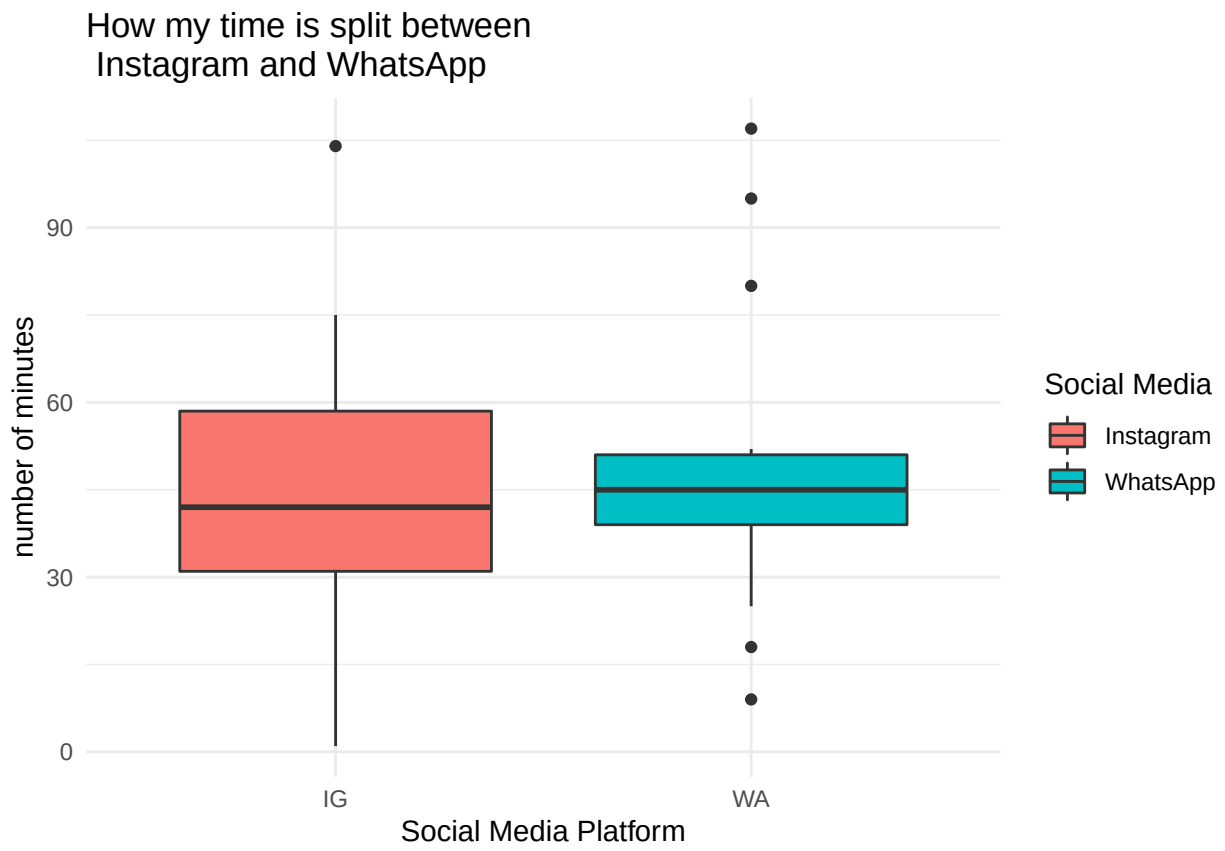
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



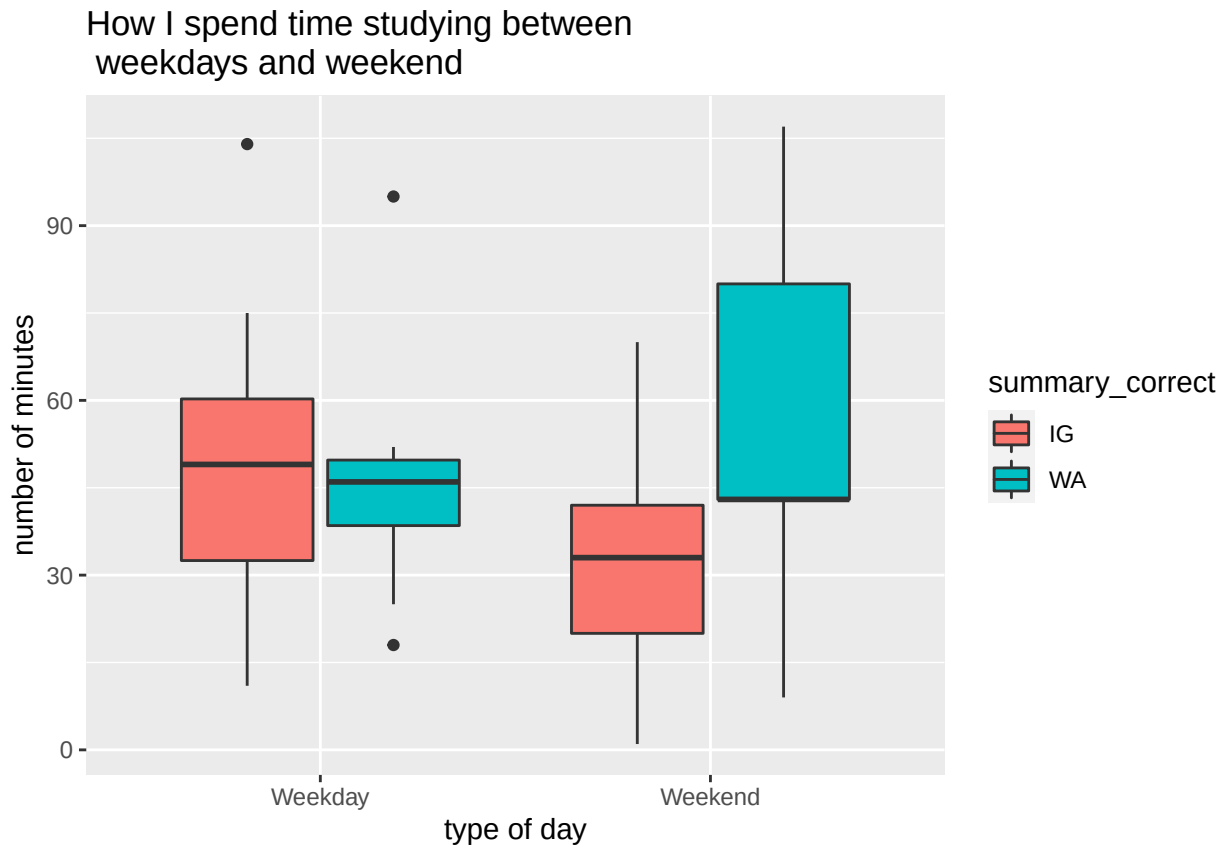

```
#social media
ggplot(data=social_media1, aes(x=summary_correct, y=length_hour, fill=summary_correct)) +  geom_bar(sta
```




```
#Boxplot to show the distribution of social media usage
ggplot(data=social_media1, aes(x=summary_correct, y=length_hour, fill=summary_correct)) + geom_boxplot()
```




```
#weekday and weekend boxplot
ggplot(data=social_media1, aes(x=type_day, y=length_hour, fill=summary_correct)) +  geom_boxplot() + lab
```




```
social_media1 %>%
  group_by(type_day,summary_correct)%>%
  summarize(avg_min = mean(length_hour))
```

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

```
## # A tibble: 4 x 3
## # Groups:   type_day [2]
##   type_day summary_correct avg_min
##   <chr>    <chr>              <dbl>
## 1 Weekday  IG                  50.4
## 2 Weekday  WA                  45.9
## 3 Weekend  IG                  33.2
## 4 Weekend  WA                  56.4
```