HW1 620

Chongwei Shi

Problem 1

```
library(readxl)
screen_activity = read_excel("Screen Time.xlsx")
head(screen_activity)
```

```
## # A tibble: 6 x 8
##   Date       ‘total ST‘ ‘total ST min‘ ‘Social ST‘ ‘Social ST min‘ ‘Pick up‘
##   <chr>      <chr>               <dbl> <chr>                 <dbl>     <dbl>
## 1 12/24/2023 5h                    298 2.5h                    144        58
## 2 12/25/2023 9h                    539 4.75h                   285       101
## 3 12/26/2023 14.417h               865 6.733h                  264       104
## 4 12/27/2023 12.5h                 748 7h                      422       178
## 5 12/28/2023 13h                   789 3.5h                    211       133
## 6 12/29/2023 11h                   652 5.717h                  343       121
## # i 2 more variables: ‘first Pick up‘ <dttm>, ‘weeday or not‘ <dbl>
```

```
# Data cleansing
colnames(screen_activity)[2] <- c("total ST (h)")
colnames(screen_activity)[4] <- c("Social ST (h)")
screen_activity$`total ST (h)` <- gsub("h", "", screen_activity$`total ST (h)`)
screen_activity$`Social ST (h)` <- gsub("h", "", screen_activity$`Social ST (h)`)
screen_activity$`first Pick up` <- gsub("1899-12-31", "", screen_activity$`first Pick up`
colnames(screen_activity) <- c("Date", "total_ST_h", "total_ST_min", "Social_ST_h", "Soci


screen_activity$Date <- as.Date(screen_activity$Date, format = "%m/%d/%Y")
screen_activity$total_ST_h <- as.numeric(screen_activity$total_ST_h)
```

a

The purpose of the data collection is to analyze the impact of screen time and social behavior, and daily routine. The hypothesis is that the increase of screen time, espeicially on social media platforms has a relationship to the later first pick up times in the morning. A prior study that is related to this topic includes the research by Woods and Scott, which suggested that excesive screen time can delay time of sleep and quality of sleep.

b

This is important as the data collection includes private data of individuals. Although it is anonymous, the participants still have the right to understand the purpose of the data collection, the process of the experiment, potential harm and benefits, and what the process will do to protect participants to the largest extent.

c

The data is collected between the time period of 12/24/2023 to 1/26/2024 with a duration of 34 days. The variables collected include:

Date: The date of each data entry.

Total Screen Time: The total duration of screen time for each day in hours.

Total Screen Time Minutes: The total duration of screen time for each day in minutes.

Social Screen Time: The duration of screen time spent on social activities for each day in hours.

Social Screen Time Minutes: The duration of screen time spent on social activities for each day in minutes.

Pick up: The number of times the device was picked up during each day.

First Pick up: The first pickup time of the device for each day.

Weekday or Not: A binary variable indicating whether the day is a weekday (1) or not (0).

The data is collected from the participant's screen activity recorded in the mobile device. The data collection ends on the end of the day of 1/26/24 which is Friday.

d

```r
# Calculate daily proportion of social screen time (as a percentage)
screen_activity$daily_proportion_social_ST <- (screen_activity$Social_ST_min / screen_act

# Calculate daily duration per use (in minutes)
screen_activity$daily_duration_per_use <- screen_activity$total_ST_min / screen_activity$
head(screen_activity)
```

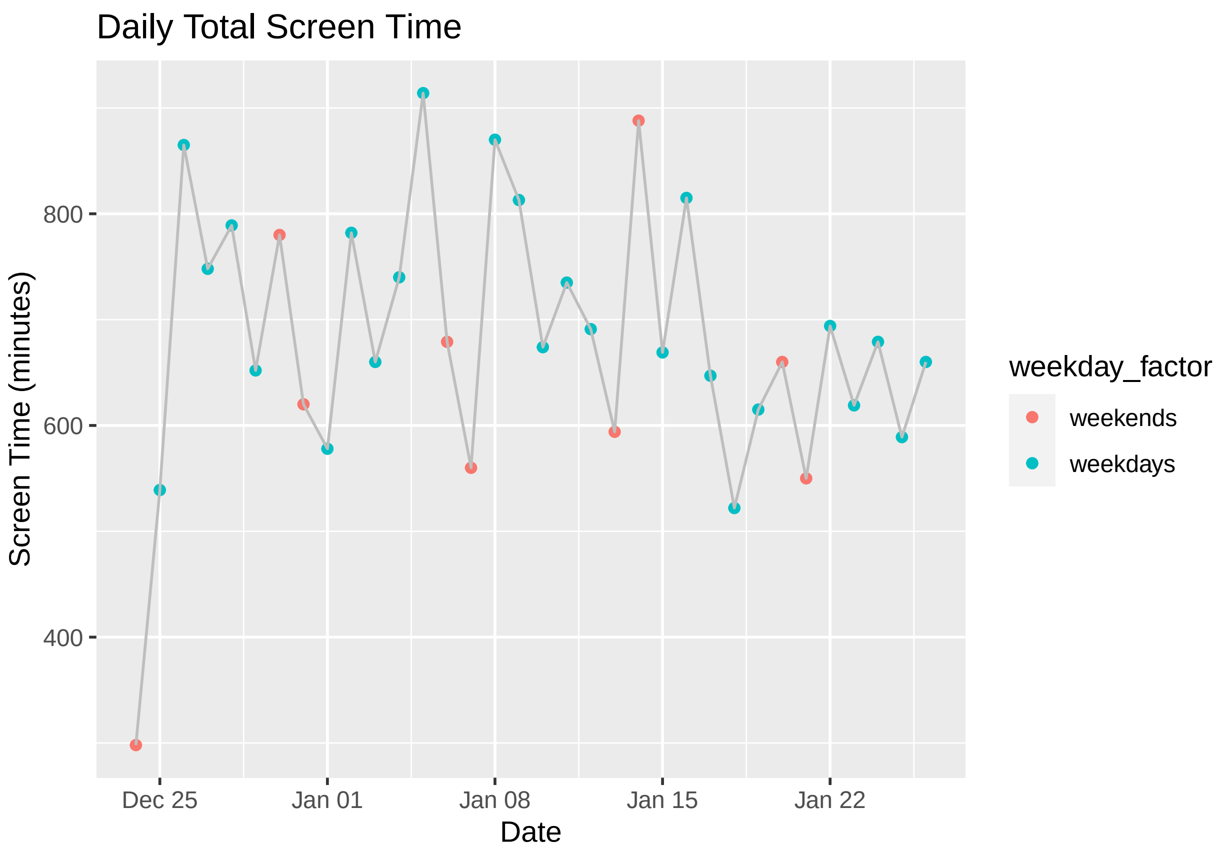
```
## # A tibble: 6 x 10
##   Date       total_ST_h total_ST_min Social_ST_h Social_ST_min Pick_up
##   <date>          <dbl>        <dbl> <chr>               <dbl>   <dbl>
## 1 2023-12-24        5            298 2.5                   144      58
## 2 2023-12-25        9            539 4.75                  285     101
## 3 2023-12-26       14.4          865 6.733                 264     104
## 4 2023-12-27       12.5          748 7                     422     178
## 5 2023-12-28       13            789 3.5                   211     133
## 6 2023-12-29       11            652 5.717                 343     121
## # i 4 more variables: first_Pick_up <chr>, weekday_or_not <dbl>,
## #   daily_proportion_social_ST <dbl>, daily_duration_per_use <dbl>
```

Problem 2

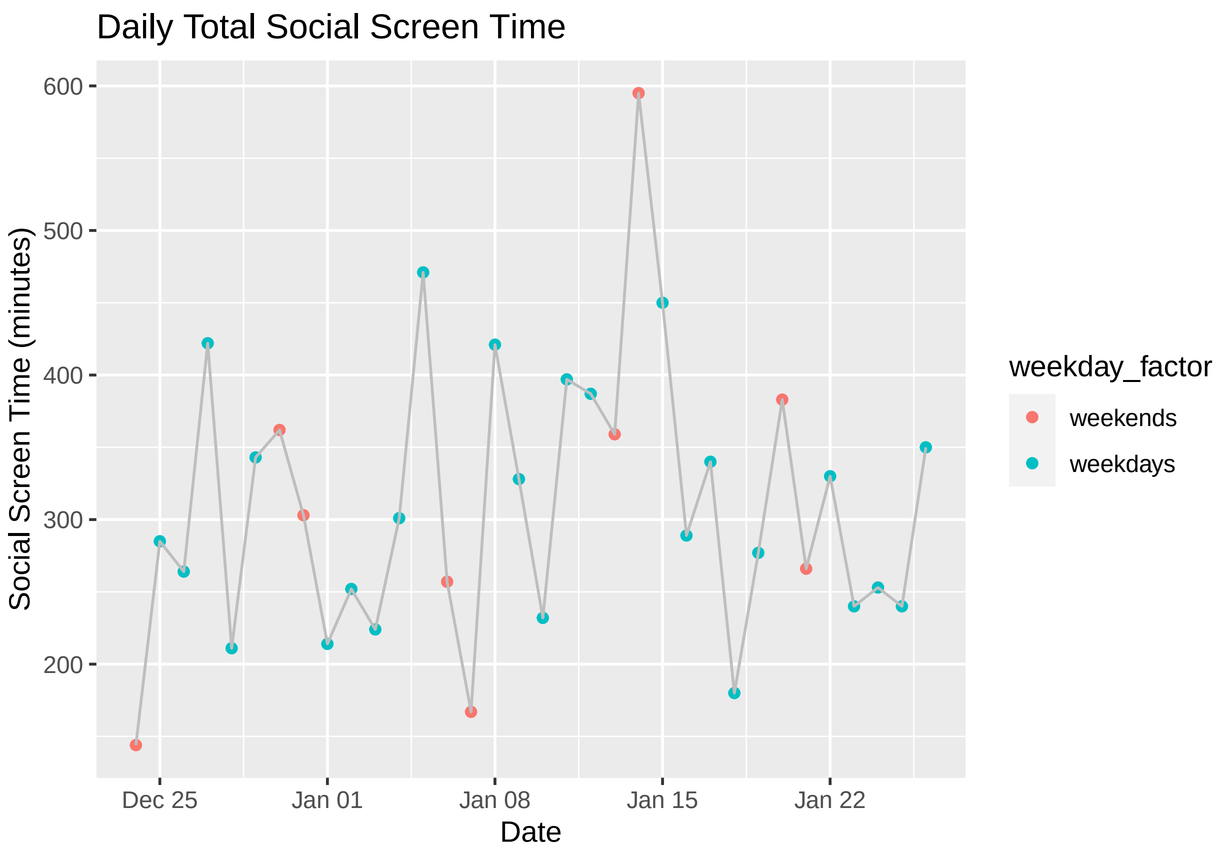
a

Daily total screen time doesn't have much difference between weekend or weekday. The total minutes seem to be around 600-800minutes throughout the entire period. Compared to the total screen time, the total social screentime seem to be a lot lower with around 200-450minutes. There is one outlier that is during weekend. Similarly, it does not seem that weekend or weekday effects the daily number of pickups. The variability seems to be a lot higher espcicially during the week of Dec 25 and Jan 15. The variability of the daily proportion of social screen time is higher varying from 25% to almost 70%. Interestingly, there is no increasing or decreasing pattern. Differently, the daily duration per use has less variability. There may be one outlier that is during weekend.

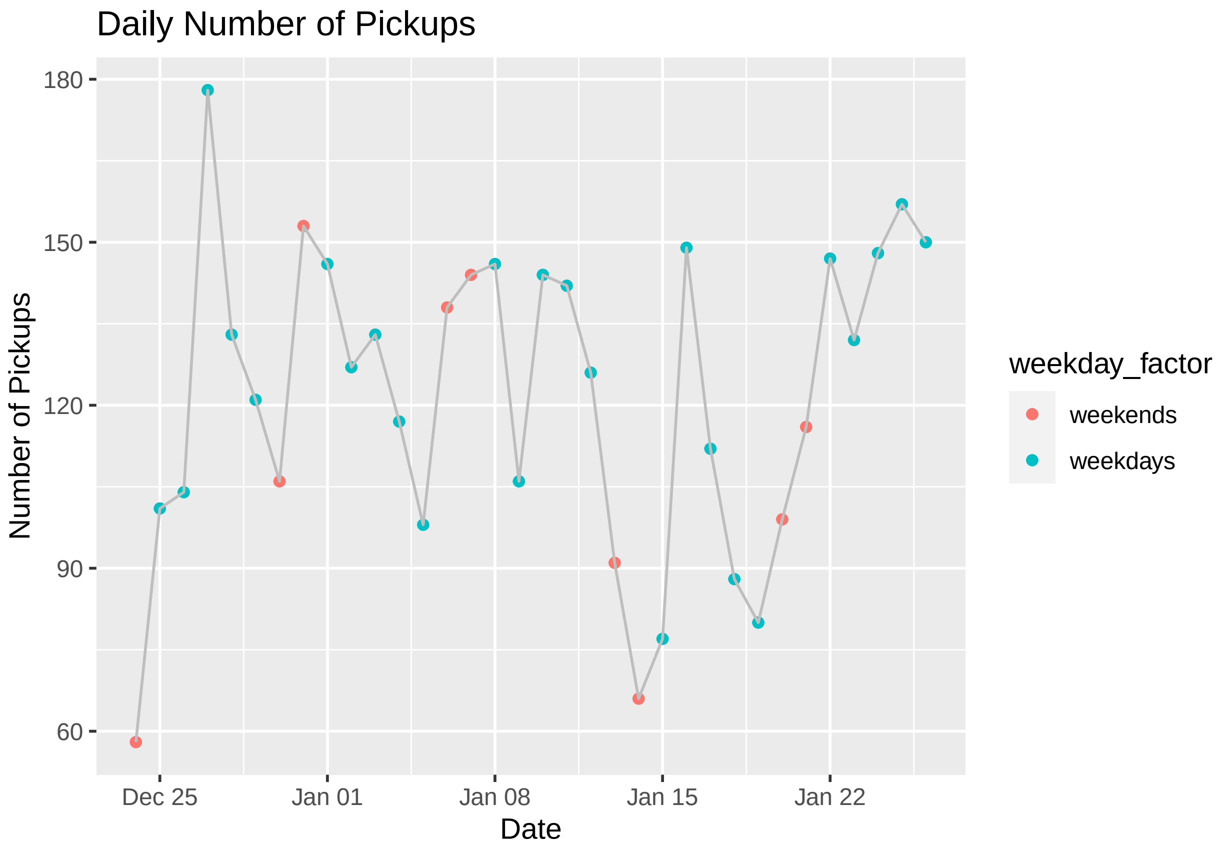
```
library(ggplot2)
par(mfrow = c(2, 3))

screen_activity$weekday_factor <- factor(screen_activity$weekday_or_not, labels = c("week

# Daily Total Screen Time
ggplot(screen_activity, aes(x = Date, y = total_ST_min, color = weekday_factor)) +
  geom_point() +
  geom_line(color = "grey") +
  labs(title = "Daily Total Screen Time",
       x = "Date",
       y = "Screen Time (minutes)")
```



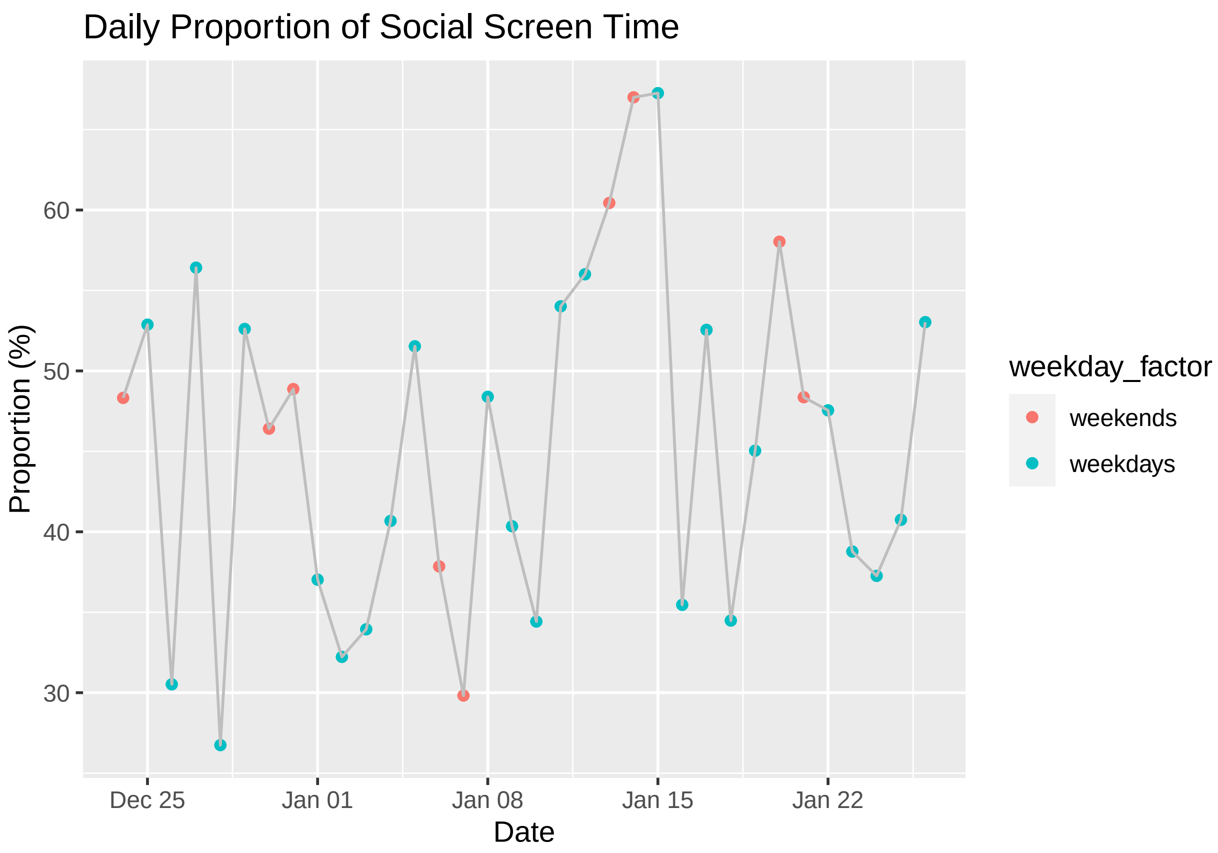
```
# Daily Total Social Screen Time
ggplot(screen_activity, aes(x = Date, y = Social_ST_min, color = weekday_factor)) +
  geom_point() +
  geom_line(color = "grey") +
  labs(title = "Daily Total Social Screen Time", x = "Date", y = "Social Screen Time (min
```



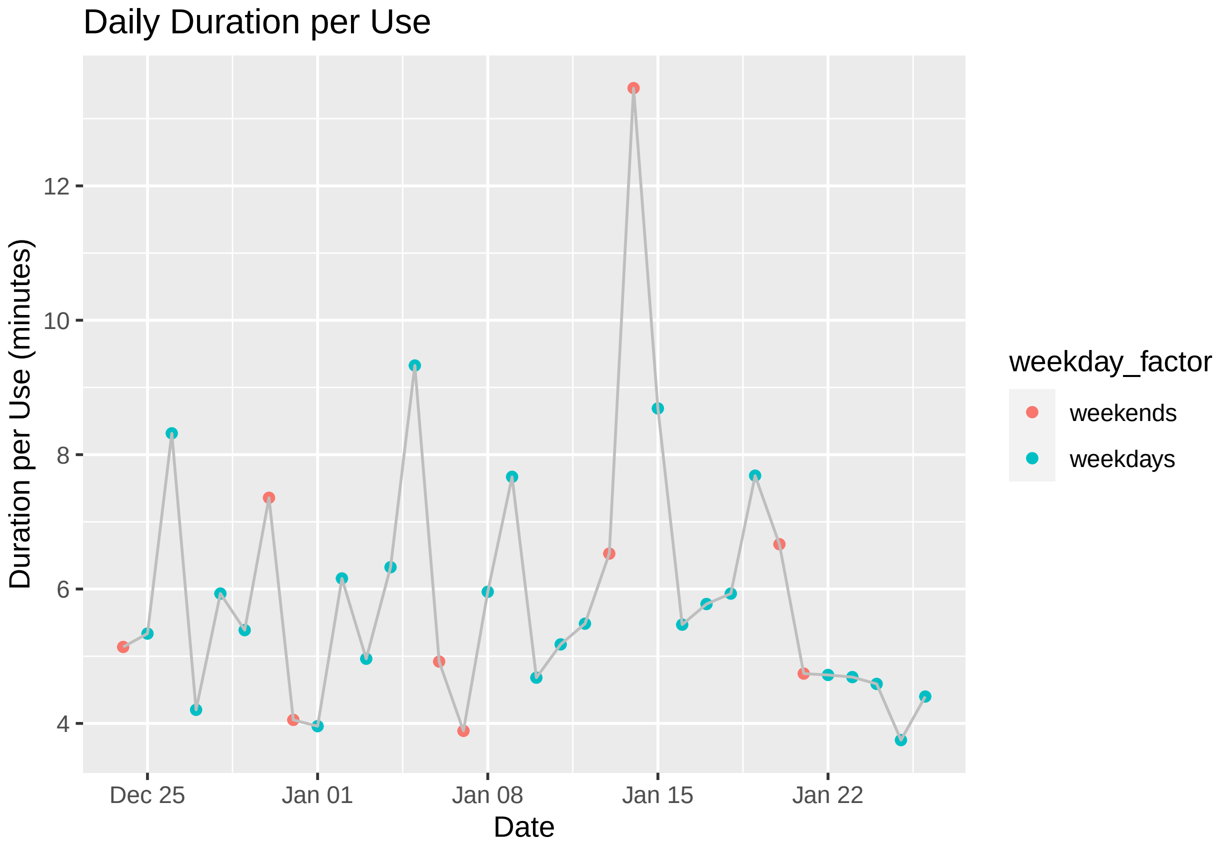
```
# Plot of Daily Number of Pickups
ggplot(screen_activity, aes(x = Date, y = Pick_up, color = weekday_factor)) +
  geom_point() +
  geom_line(color = "grey") +
  labs(title = "Daily Number of Pickups", x = "Date", y = "Number of Pickups")
```



```
# Plot of Daily Proportion of Social Screen Time
ggplot(screen_activity, aes(x = Date, y = daily_proportion_social_ST, color = weekday_fac
  geom_point() +
  geom_line(color = "grey") +
  labs(title = "Daily Proportion of Social Screen Time", x = "Date", y = "Proportion (%)"
```



```
# Plot of Daily Duration per Use
ggplot(screen_activity, aes(x = Date, y = daily_duration_per_use, , color = weekday_facto
  geom_point() +
  geom_line(color = "grey") +
  labs(title = "Daily Duration per Use", x = "Date", y = "Duration per Use (minutes)")
```



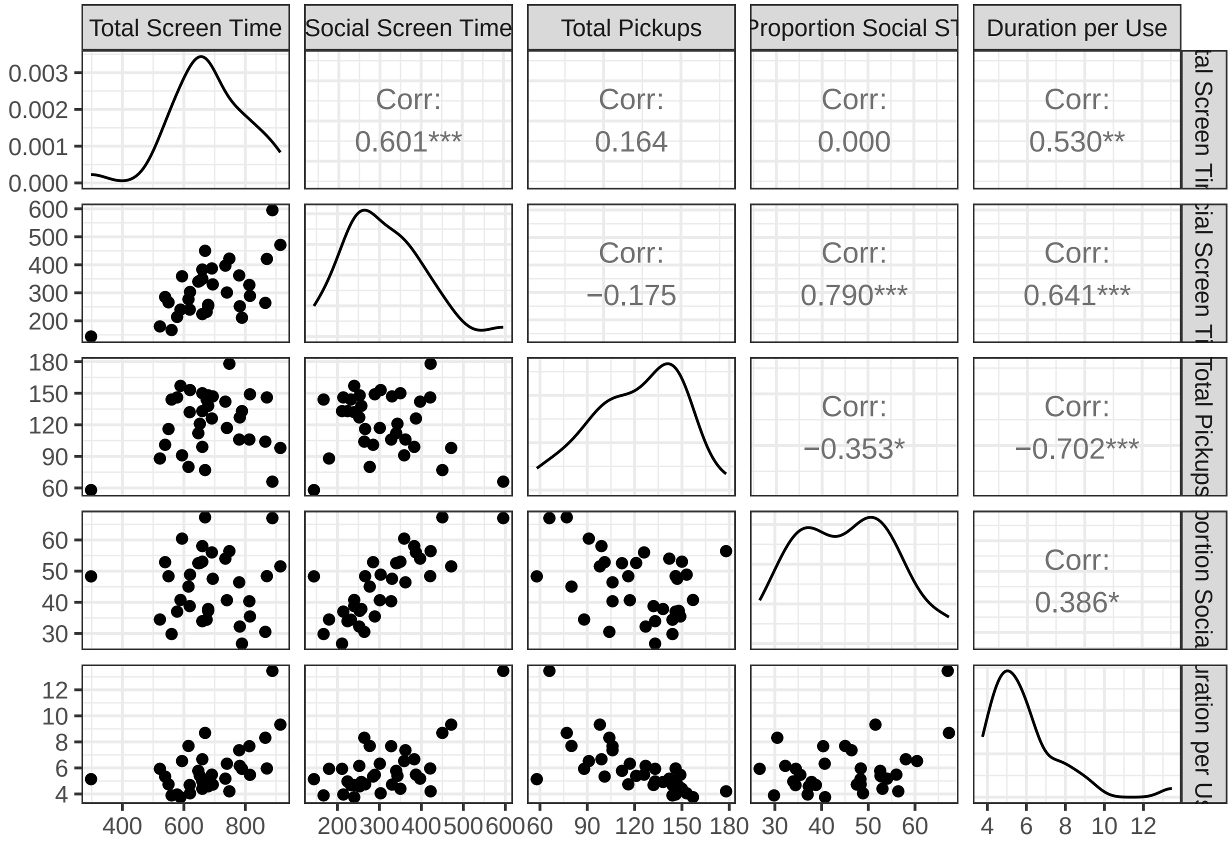
b

Proportion of total screen time vs social screen time has the highest positive corrleation and duration per use vs total pick up has the second highiest with a negative correlation.

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
ggpairs(screen_activity,
        columns = c("total_ST_min", "Social_ST_min", "Pick_up", "daily_proportion_social_
        columnLabels = c("Total Screen Time", "Social Screen Time", "Total Pickups", "Pro
  theme_bw()
```

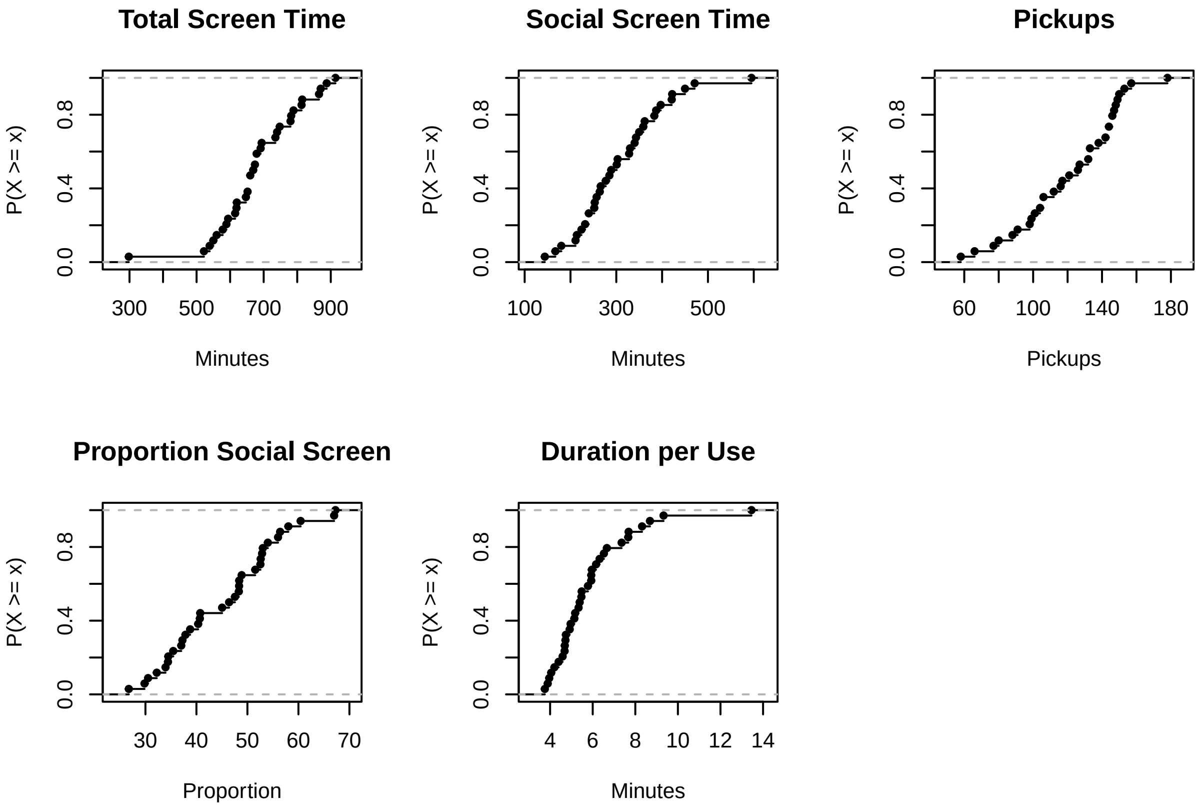


c

In general, it seems like screen time is very active from the five plots. There is higher probability for x-axis to be greater for all 5 plots.

```r
par(mfrow = c(2, 3))
# Daily Total Screen Time
cdf_total <- ecdf(screen_activity$total_ST_min)
plot(cdf_total, main="Total Screen Time",
     xlab="Minutes", ylab="P(X >= x)")

# Daily Social Screen Time
cdf_social <- ecdf(screen_activity$Social_ST_min)
plot(cdf_social, main="Social Screen Time",
     xlab="Minutes", ylab="P(X >= x)")

# Daily Pickups
cdf_pickups <- ecdf(screen_activity$Pick_up)
plot(cdf_pickups, main="Pickups",
     xlab="Pickups", ylab="P(X >= x)")
```

```
# Daily Proportion Social Screen Time
cdf_prop <- ecdf(screen_activity$daily_proportion_social_ST)
plot(cdf_prop, main="Proportion Social Screen",
     xlab="Proportion", ylab="P(X >= x)")

# Daily Duration per Use
cdf_duration <- ecdf(screen_activity$daily_duration_per_use)
plot(cdf_duration, main="Duration per Use",
     xlab="Minutes", ylab="P(X >= x)")
```

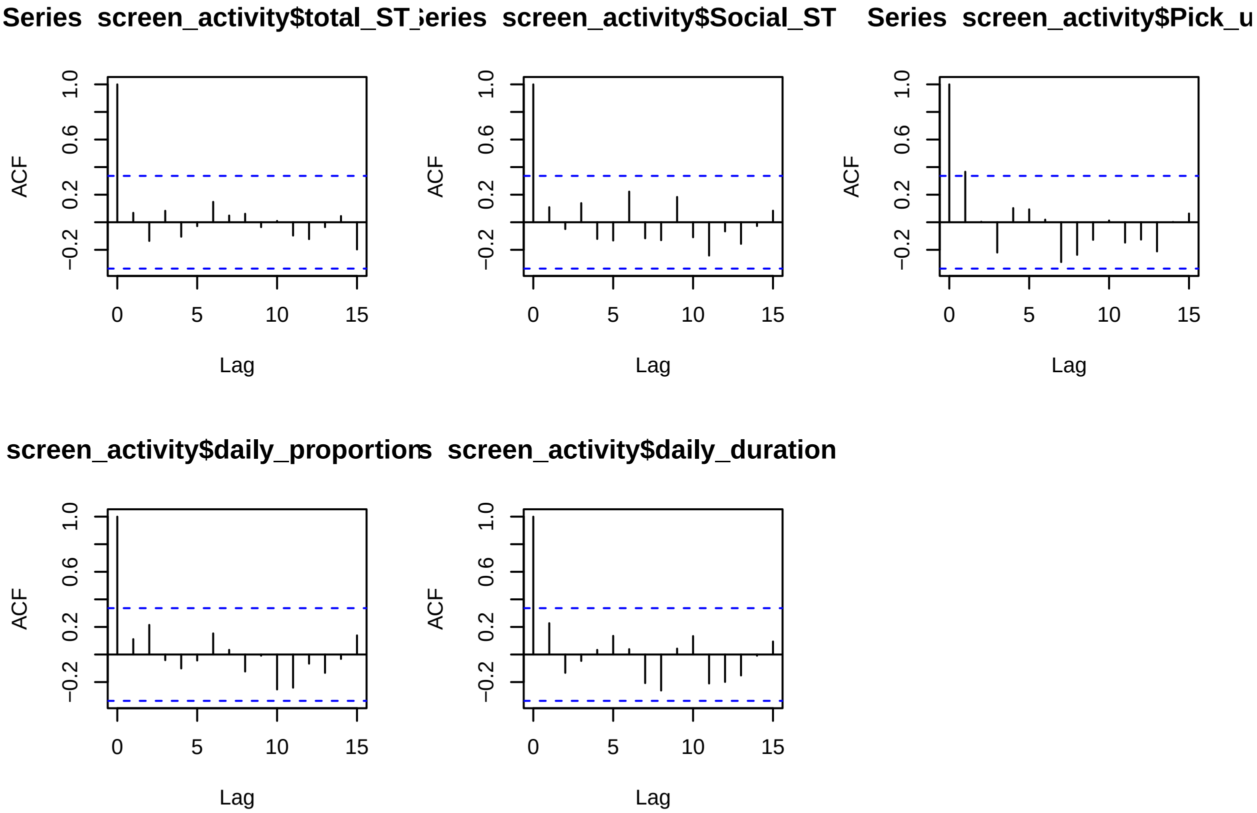


d

It seems like the only significant autocorrelation is with Pick_up with lag 1 indicating that there is a positive autocorrelation at a 1 day lag. This can indicate that high pickups on one day are likely to be followed by high pickups the following day.

```
par(mfrow = c(2, 3))

acf(screen_activity$total_ST_min)
acf(screen_activity$Social_ST_min)
acf(screen_activity$Pick_up)
acf(screen_activity$daily_proportion_social_ST)
acf(screen_activity$daily_duration_per_use)


acf(screen_activity$Pick_up, plot = FALSE)
```

```
##
## Autocorrelations of series 'screen_activity$Pick_up', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.366  0.004 -0.220  0.103  0.094  0.019 -0.289 -0.237 -0.128  0.013
##     11     12     13     14     15
## -0.148 -0.126 -0.212  0.002  0.064
```



Problem 3

a

```
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library(circular)
```

```
##
## Attaching package: ’circular’

## The following objects are masked from ’package:stats’:
##
##     sd, var
```

```
library(lubridate)
```

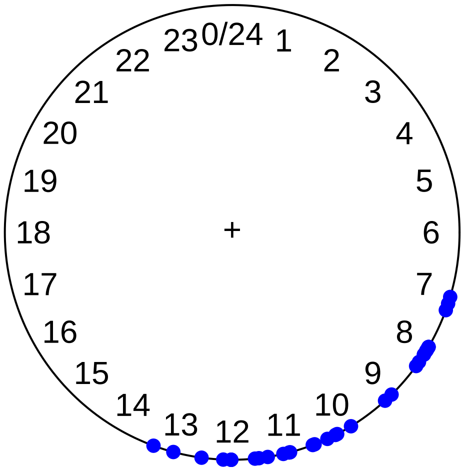
```
##
## Attaching package: ’lubridate’

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

```
library(ggplot2)

# Convert 'first_Pick_up' to a POSIXct object, assuming the dates are on the same day
screen_activity <- screen_activity %>%
  mutate(
    first_Pick_up = hms(first_Pick_up),
    Pickup_1st_angular = (hour(first_Pick_up) * 60 + minute(first_Pick_up)) / (24 * 60) *
  )

# Create a circular object
first_pickup_cir <- circular(screen_activity$Pickup_1st_angular, units = "degrees", templ
```

b

The first pickup time varies from around 7am to 1pm. This person seem to have a wide first pickup pattern. If first pickup time can somewhat reflect the wake up pattern, this indicates that this person does not seem to have a routine wake up time.

```
# Scatterplot
plot(first_pickup_cir, col = "blue")
```



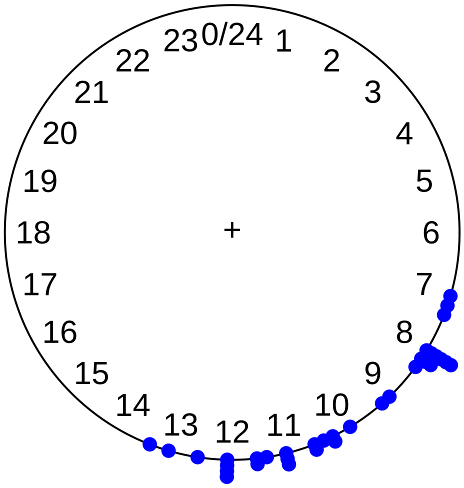
```
dev.off()
```

```
## null device
##           1
```

c

We chose a bin size of 30 minutes interval with 144 bins to reflect more details since the variation is quite large for the time frame. As shown in the graph, the person's first pick up time is mostly at 8am. And if it is not 8am, then the pickup time would vary mostly likely to be around 10am to 1pm.

```
# histogram
plot(first_pickup_cir, stack = TRUE, bins = 144, col = "blue") # 48 bins for 30-minute i
```



```
dev.off()
```

```
## null device
##           1
```

Problem 4

a

S_t is needed to capture the variability of the daily total screen time. It provides a scaling effect assuming that with the days of more screen time, we might expect more pickups.

b

```
poisson_model <- glm(Pick_up ~ offset(log(total_ST_h)), family = poisson, data = screen_a

summary(poisson_model)
```

```
##
## Call:
## glm(formula = Pick_up ~ offset(log(total_ST_h)), family = poisson,
##     data = screen_activity)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.3128  -1.4422   0.3082   2.0248   4.7251
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.36976    0.01555   152.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 291.51  on 33  degrees of freedom
## Residual deviance: 291.51  on 33  degrees of freedom
## AIC: 518.21
##
## Number of Fisher Scoring iterations: 4
```

c

```
screen_activity <- screen_activity %>%
  mutate(
    Xt = as.numeric(weekday_factor == "weekdays"),
    Zt = as.numeric(Date >= "2024-01-10")
  )
screen_activity$total_ST_h <- as.numeric(screen_activity$total_ST_h)
log_linear_model <- glm(Pick_up ~ Xt + Zt + offset(log(total_ST_h)), family = poisson, da

summary(log_linear_model)
```

```
##
## Call:
## glm(formula = Pick_up ~ Xt + Zt + offset(log(total_ST_h)), family = poisson,
##     data = screen_activity)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.9974  -1.3419   0.1841   1.9332   4.5587
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  2.33210    0.03540  65.877   <2e-16 ***
## Xt           0.04364    0.03669   1.189    0.234    
## Zt           0.00910    0.03112   0.292    0.770    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 291.51  on 33  degrees of freedom
## Residual deviance: 289.99  on 31  degrees of freedom
## AIC: 520.69
## 
## Number of Fisher Scoring iterations: 4
```

(c1)

The p-value for Xt is 0.234 which is not smaller than 0.05, so we would not say that there is a statistically significant difference in daily pickups between weekdays and weekends.

(c2)

The p-value for Zt is 0.77 which is not smaller than 0.05, so we would not say that there is a statistically significant change in daily pickups after January 10th, the start of the winter semester.

Problem 5

a

```r
screen_activity <- screen_activity %>%
  mutate(
    Pickup_1st_hours = hour(first_Pick_up) + minute(first_Pick_up) / 60,
    Pickup_1st_radians = Pickup_1st_hours * (2 * pi / 24)
  )

estimates <- mle.vonmises(screen_activity$Pickup_1st_radians)
```

```
## Warning in as.circular(x): an object is coerced to the class ’circular’ using default
##   type: ’angles’
##   units: ’radians’
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularxradians0counter2pi
```

```r
print(estimates)
```

```
##
## Call:
## mle.vonmises(x = screen_activity$Pickup_1st_radians)
##
## mu: 2.6  ( 0.08324 )
##
## kappa: 4.78  ( 1.071 )
```

```r
mu = estimates$mu
lambda = estimates$kappa
```

b

```r
cutoff <- (8 + 30/60) * (pi/12) # 8:30AM in radians
probability_8_30_or_later <- 1 - pvonmises(cutoff, mu, lambda)
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularqradians0counter
```

```r
print(probability_8_30_or_later)
```

```
## [1] 0.7850864
```