Biostat620hw1-RongJi

RongJi

2024-02-05

https://github.com/rongjiiiii/620HW1.git (https://github.com/rongjiiiii/620HW1.git)

Problem 1: Data Collection and Processing

a. purpose of data collection

We are interested in the relationship between social median screen time per day and the sleep times .We hypothesize that a higher proportion of social media screen time is associated with later sleep times in individuals.

[^1]: Hjetland, G. J., Skogen, J. C., Hysing, M., & Sivertsen, B. (2021). The Association Between Self-Reported Screen Time, Social Media Addiction, and Sleep Among Norwegian University Students. Frontiers in public health, 9, 794307. https://doi.org/10.3389/fpubh.2021.794307 (https://doi.org/10.3389/fpubh.2021.794307)

b. Explain the role of Informed Consent Form in connection to the planned study and data collection

This is a document that participants in a study sign to acknowledge that they understand the nature of the research and agree to participate. It’s a critical component of ethical research, especially when personal data is being collected. You need to explain how the consent form is related to your study and data collection, indicating that participants are made aware of what data is being collected, why, and how it will be used. Perhaps most importantly, an Informed Consent Form also includes information about confidentiality and the right to withdraw from the study at any time. These measures ensure that participants are fully informed about their rights as participants, and that they are able to make an informed decision about whether to participate in the study.

c. Data collection plan:

1. Data are collected on January 26, 2024, the data freeze day.

2. Variables collected include total screen time per day, social media screen time per day, number of pickups per day and first pick up time.

3. Data are collected from each participant’s mobile phone.

4. About 34 students’ data who enrolled in Biostat620 will be collected,and each participation has about 3 weeks’ data, ending at January 26,2024.

d.

```
rm(list = ls())
# install.packages("gridExtra")
library(readxl)
library(lubridate)
```

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

```
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(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
hw1data <- read_excel("/Users/jiniuniu/Downloads/screentimedata.xlsx")
hw1data$Pickup.1st <- format(hw1data$Pickup.1st, "%H:%M")
hw1data$Date <- as.Date(hw1data$Date, format = "%m/%d/%Y")
convert_time <- function(x) {
  x <- as.period(hm(x))
  return(60*hour(x) + minute(x))
}

hw1data$Total.ST.min <- convert_time(hw1data$Total.ST)
hw1data$Social.ST.min <- convert_time(hw1data$Social.ST)

hw1data$Social.ST.prop <- hw1data$Social.ST.min / hw1data$Total.ST.min
hw1data$Duration.per.use <- hw1data$Total.ST.min / hw1data$Pickups
hw1data
```

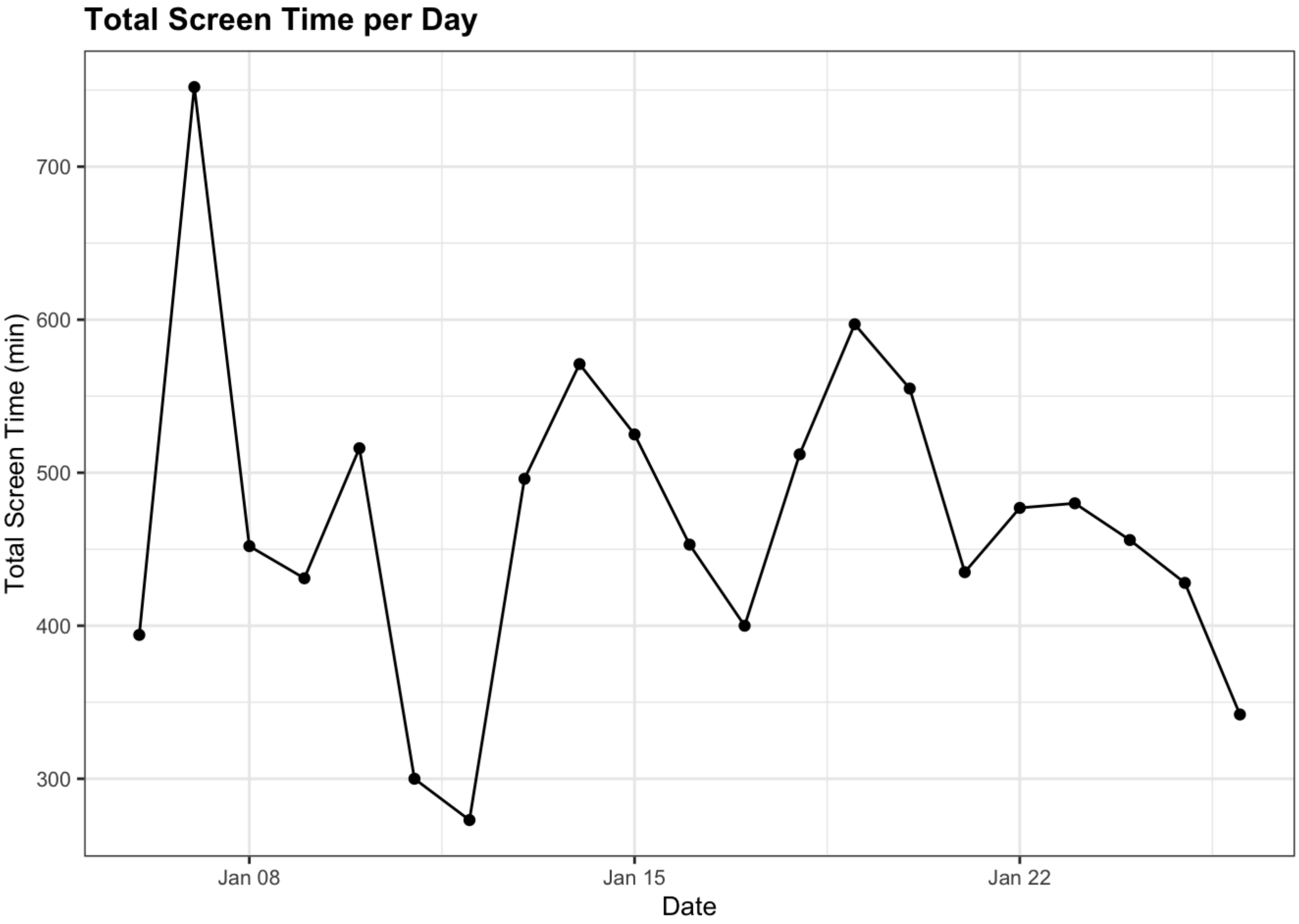
```
## # A tibble: 21 × 9
##    Date       Total.ST Total.ST.min Social.ST Social.ST.min Pickups Pickup.1st
##    <date>     <chr>           <dbl> <chr>             <dbl>   <dbl> <chr>
##  1 2024-01-06 6h34min           394 5h28min             328     112 01:03
##  2 2024-01-07 12h32min          752 9h55min             595      62 02:47
##  3 2024-01-08 7h32min           452 5h12min             312     156 00:38
##  4 2024-01-09 7h11min           431 5h28min             328     171 01:19
##  5 2024-01-10 8h36min           516 6h34min             394     172 00:00
##  6 2024-01-11 5h0min            300 4h2min              242      80 08:58
##  7 2024-01-12 4h33min           273 3h37min             217      78 00:15
##  8 2024-01-13 8h16min           496 6h43min             403      58 00:38
##  9 2024-01-14 9h31min           571 8h38min             518     114 00:18
## 10 2024-01-15 8h45min           525 5h24min             324     141 07:15
## # ℹ 11 more rows
## # ℹ 2 more variables: Social.ST.prop <dbl>, Duration.per.use <dbl>
```

Problem 2

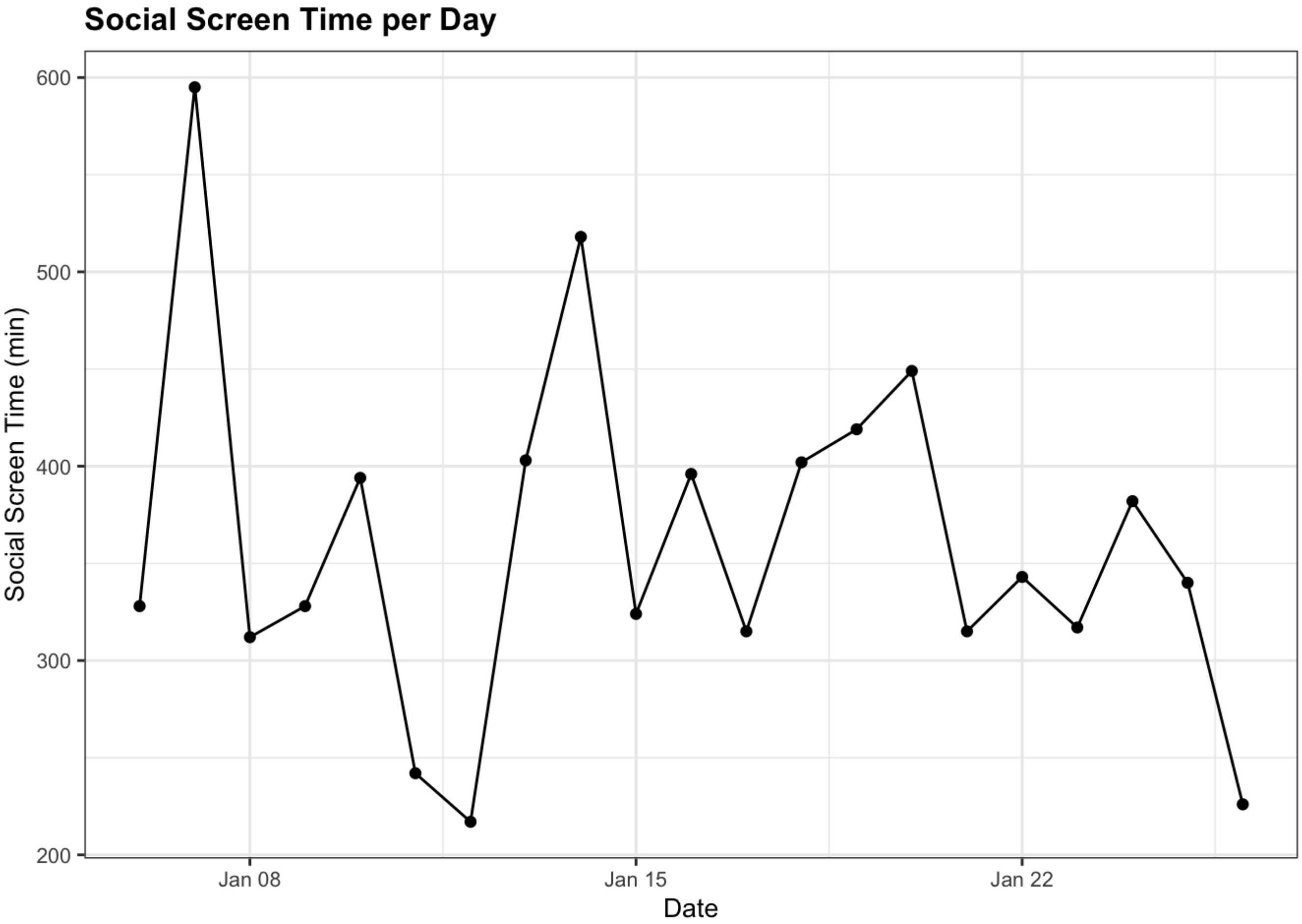
(a).Make a time series plot of each of the five variables in your data. Describe temporal patterns from these time series plots.

```
library(ggplot2)
library(gridExtra)


tot.st.min.plot <- ggplot(hw1data, aes(x = Date, y = Total.ST.min)) +
  geom_line() +
  geom_point() +
  labs(title = "Total Screen Time per Day", x = "Date", y = "Total Screen Time (min)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))

social.st.min.plot <- ggplot(hw1data, aes(x = Date, y = Social.ST.min)) +
  geom_line() +
  geom_point() +
  labs(title = "Social Screen Time per Day", x = "Date", y = "Social Screen Time (min)")
+
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))

pickups.plot <- ggplot(hw1data, aes(x = Date, y = Pickups)) +
  geom_line() +
  geom_point() +
  labs(title = "Number of Pickups per Day", x = "Date", y = "Number of Pickups") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))

social.st.prop.plot <- ggplot(hw1data, aes(x = Date, y = Social.ST.prop)) +
  geom_line() +
  geom_point() +
  labs(title = "Proportion of Social Screen Time per Day", x = "Date", y = "Proportion o
f Social Screen Time") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))

duration.per.use.plot <- ggplot(hw1data, aes(x = Date, y = Duration.per.use)) +
  geom_line() +
  geom_point() +
  labs(title = "Duration per Use per Day", x = "Date", y = "Duration per Use (min)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))
```

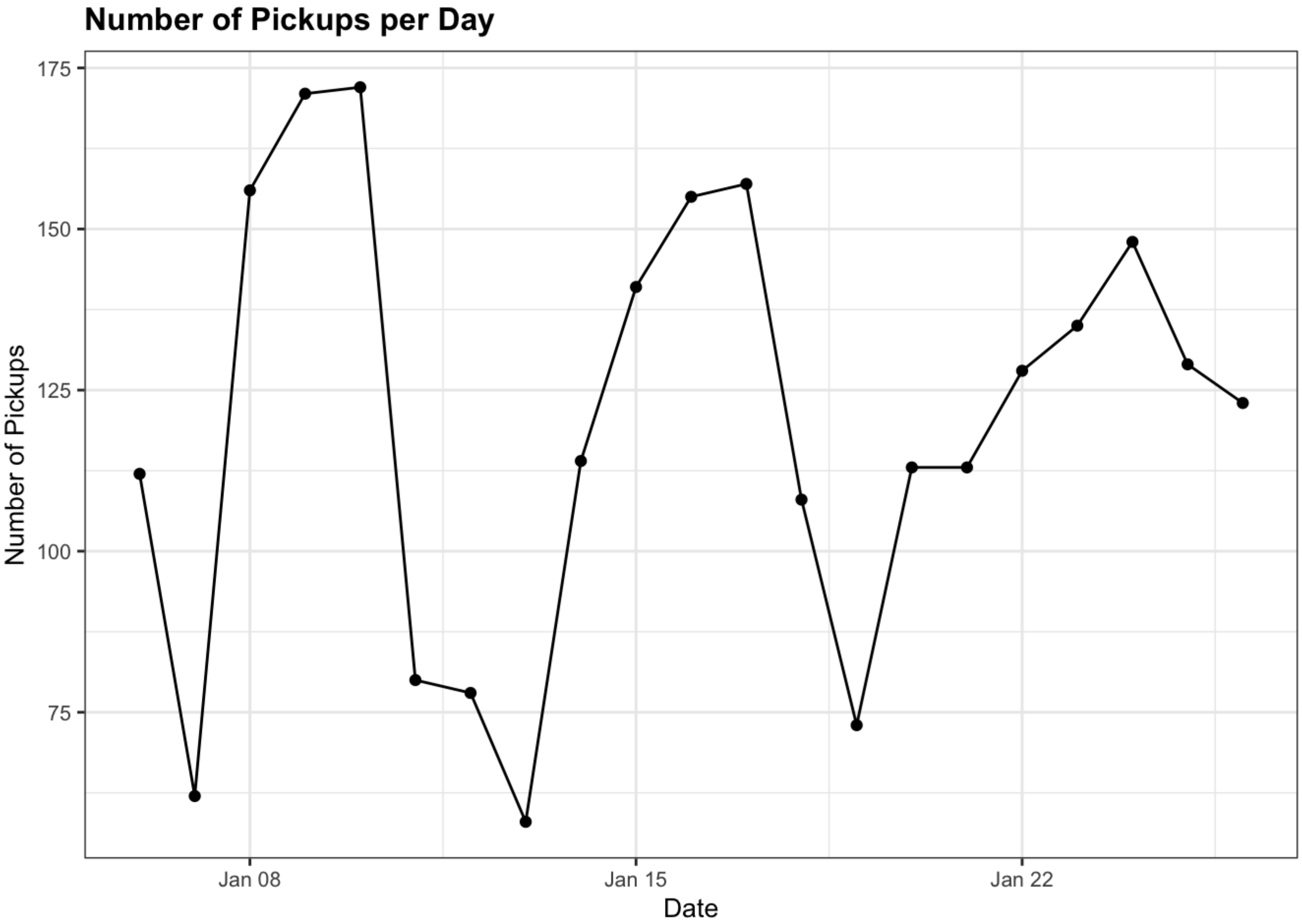
```
tot.st.min.plot
```



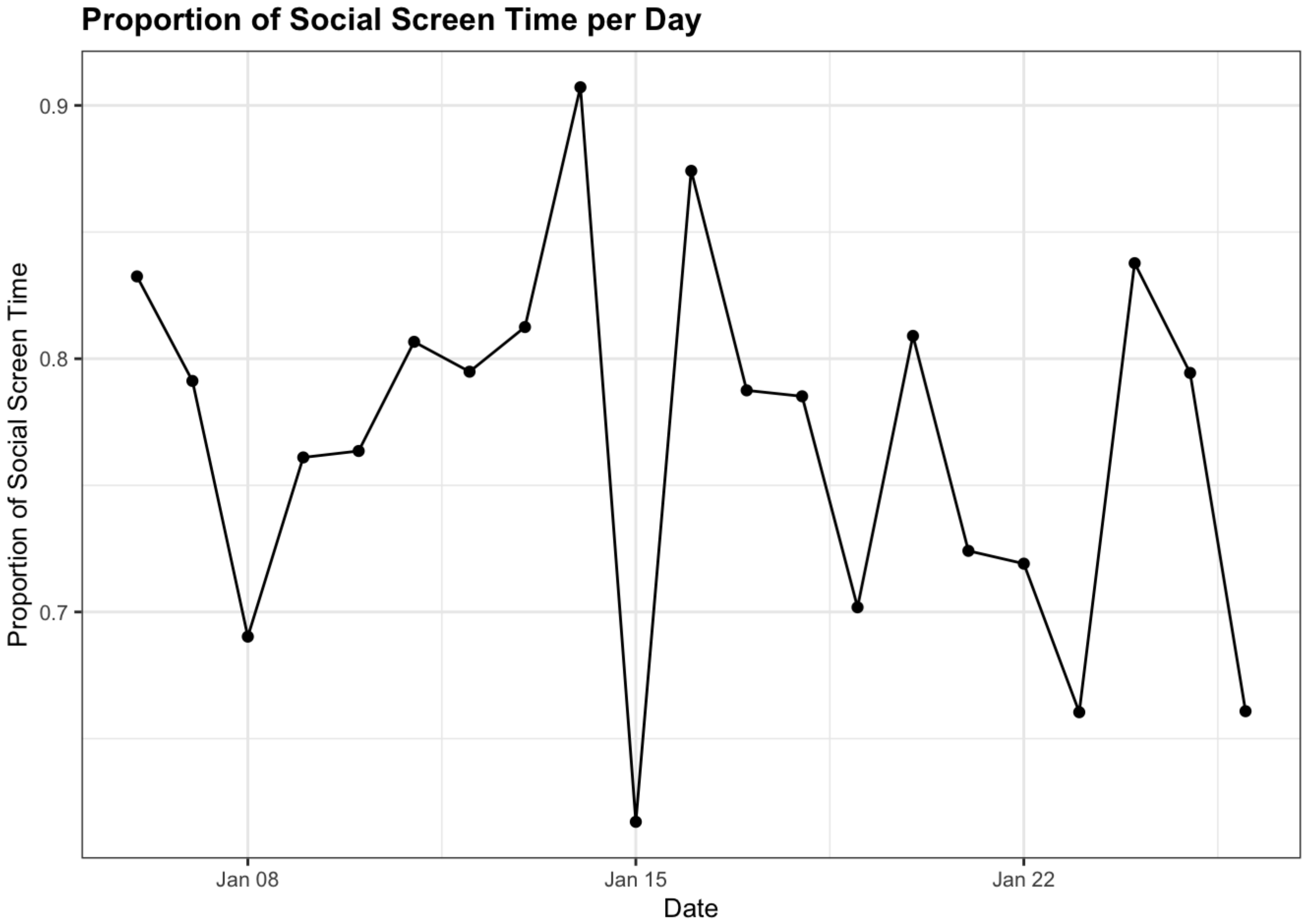
```
social.st.min.plot
```



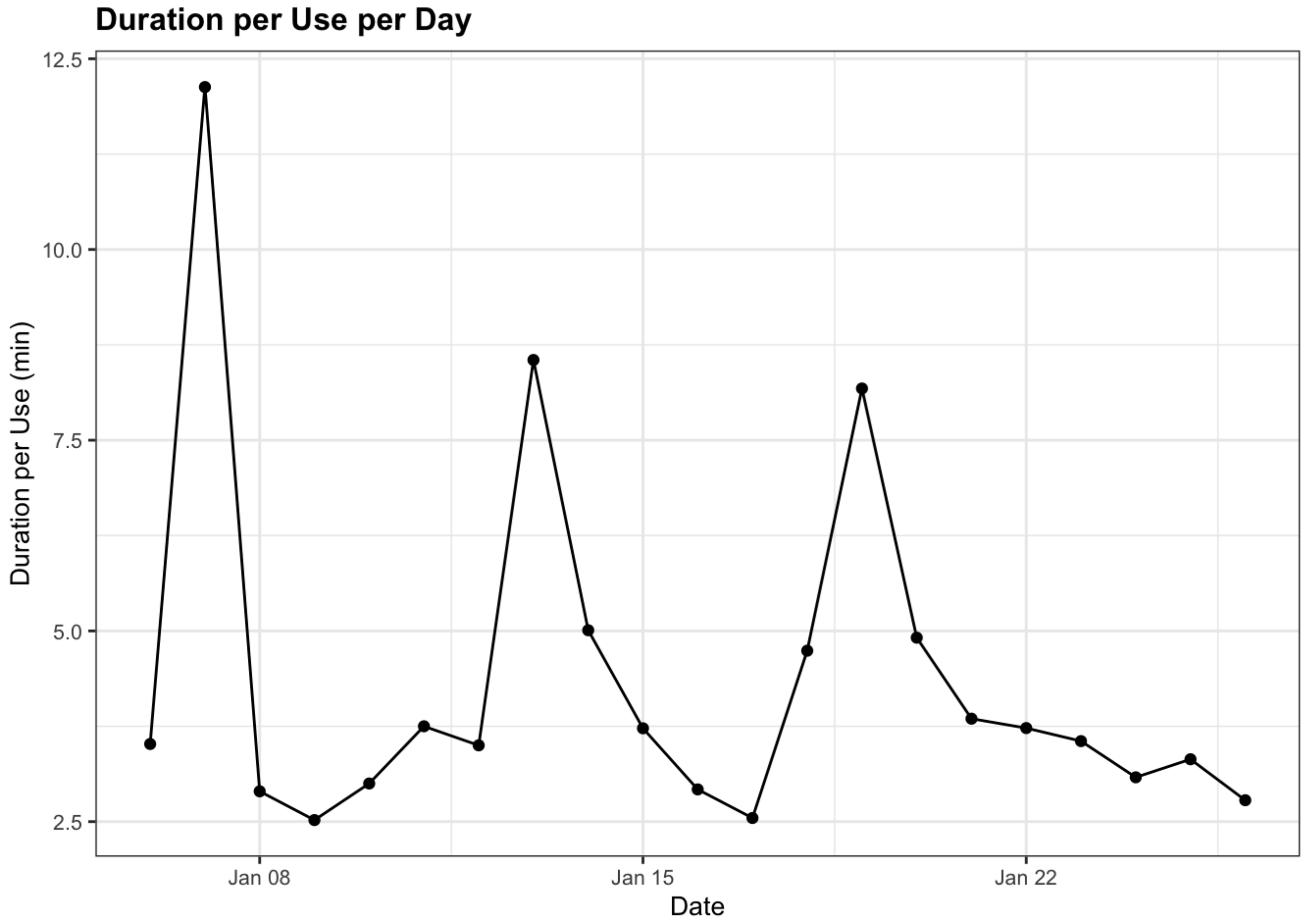
```
pickups.plot
```



```
social.st.prop.plot
```



```
duration.per.use.plot
```

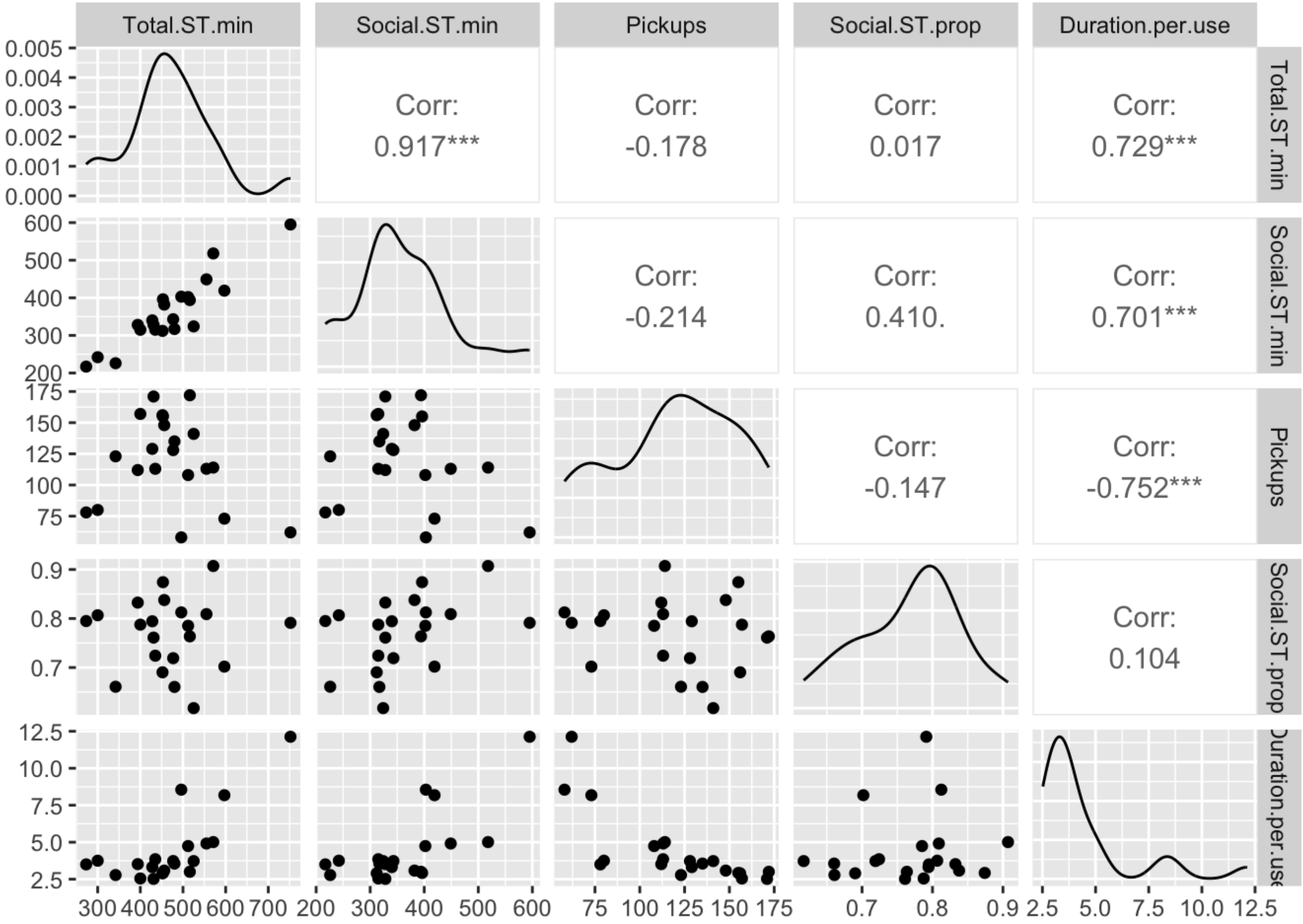


b.

```
library(GGally)
```

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

```
ggpairs(hw1data[, c(
  "Total.ST.min",
  "Social.ST.min",
  "Pickups",
  "Social.ST.prop",
  "Duration.per.use"
)])
```

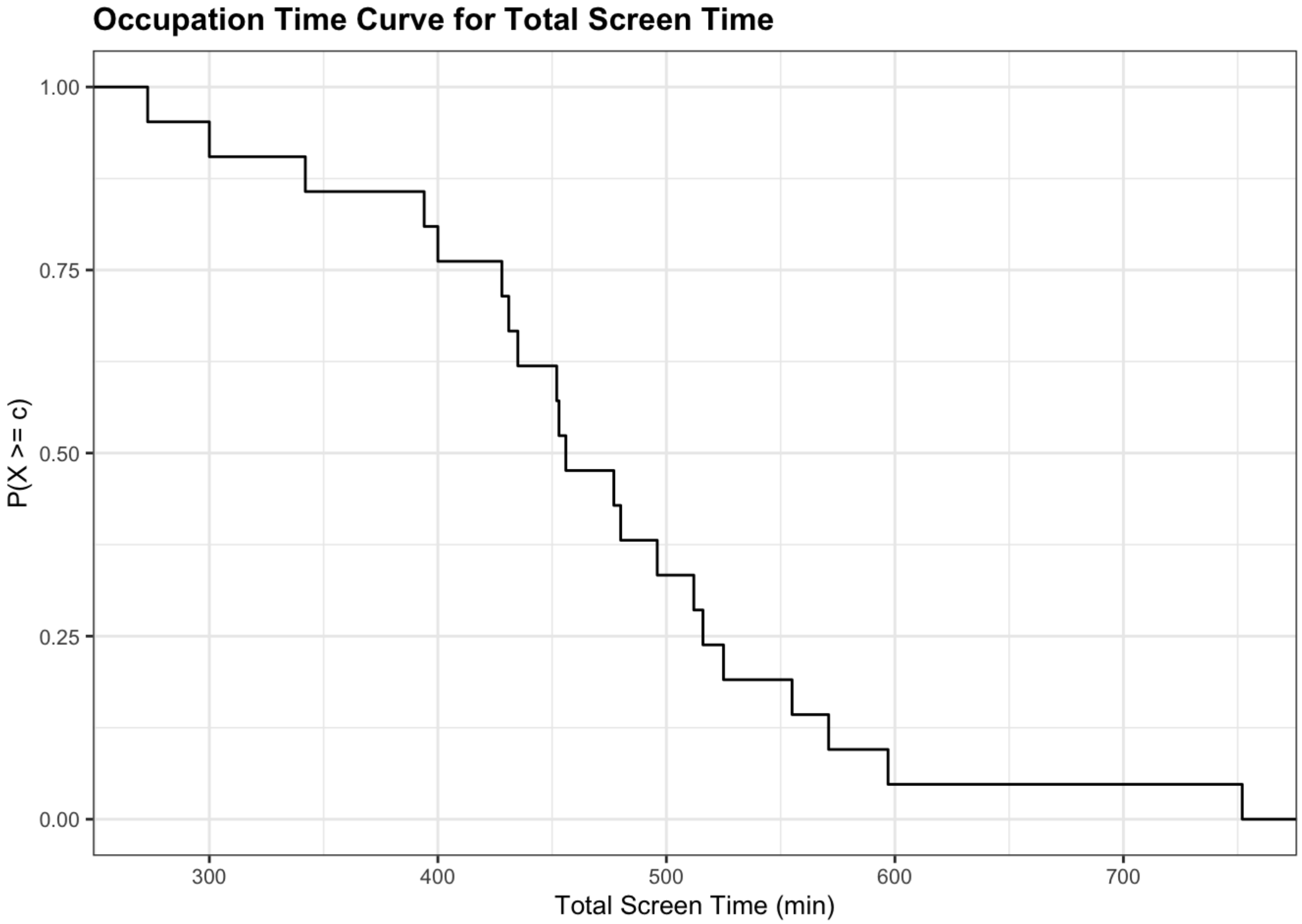


Total Screen Time per day and Social Screen Time per day has the highest correlation.

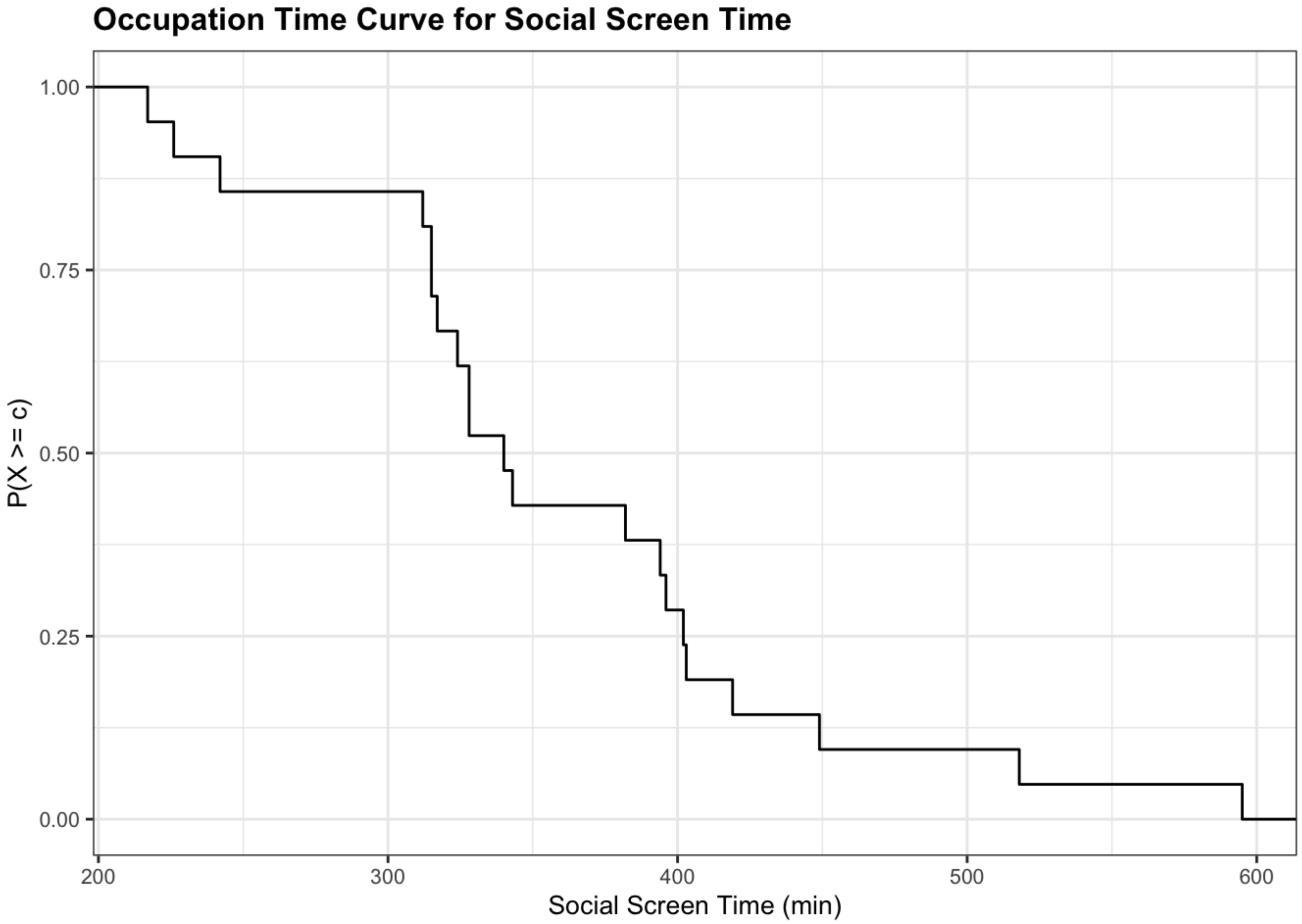
c.

```
occupation_time_curve_TST <- ggplot(hw1data, aes(x = Total.ST.min)) +
  labs(title = "Occupation Time Curve for Total Screen Time",
       x = "Total Screen Time (min)",
       y = "P(X >= c)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))
occupation_time_curve_TST + stat_ecdf(geom = "step", aes(y = 1 - ..y..))
```

```
## Warning: The dot-dot notation (`..y..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(y)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

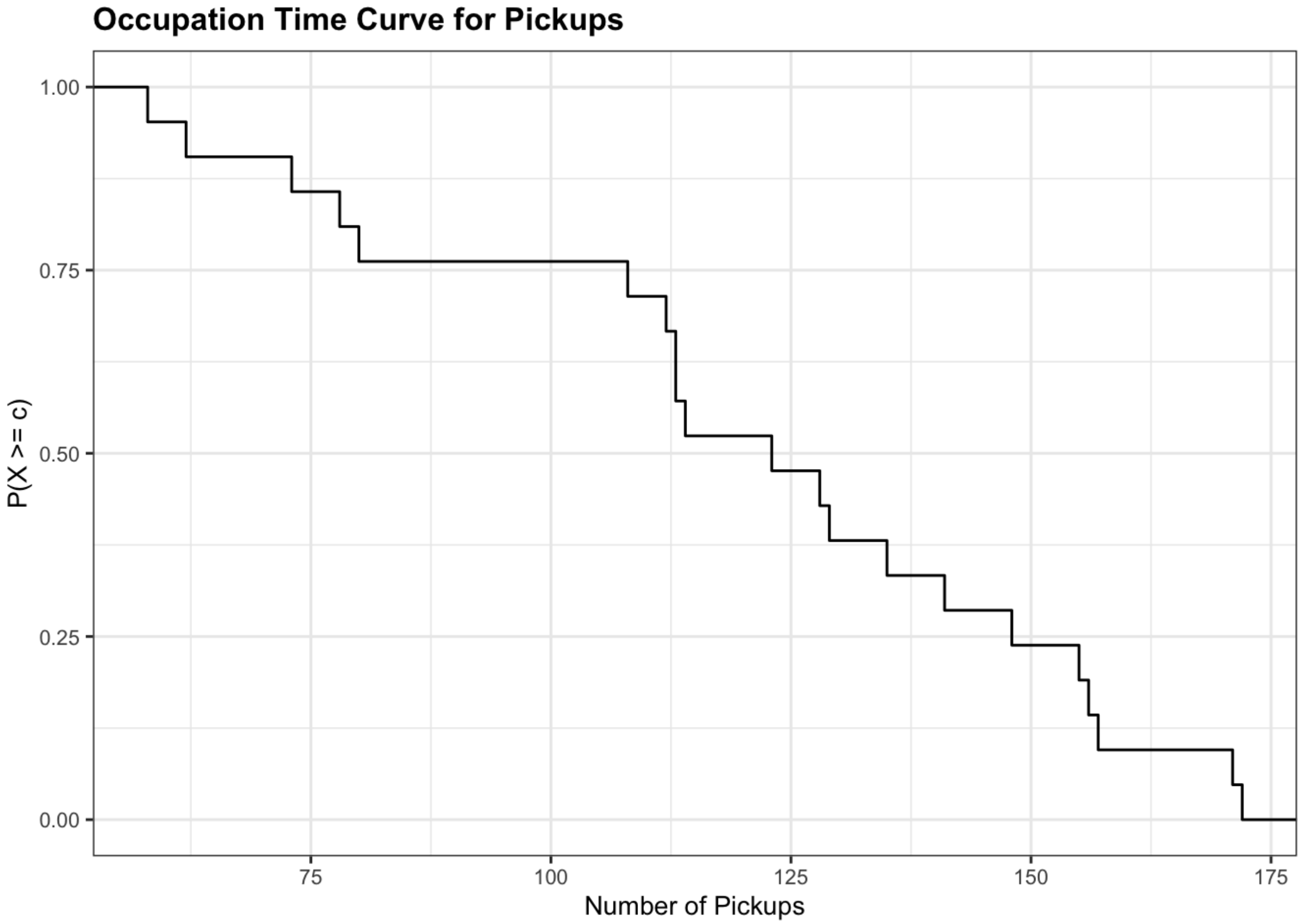


```
occupation_time_curve_SST <- ggplot(hw1data, aes(x = Social.ST.min)) +
  labs(title = "Occupation Time Curve for Social Screen Time",
       x = "Social Screen Time (min)",
       y = "P(X >= c)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))
occupation_time_curve_SST + stat_ecdf(geom = "step", aes(y = 1 - ..y..))
```

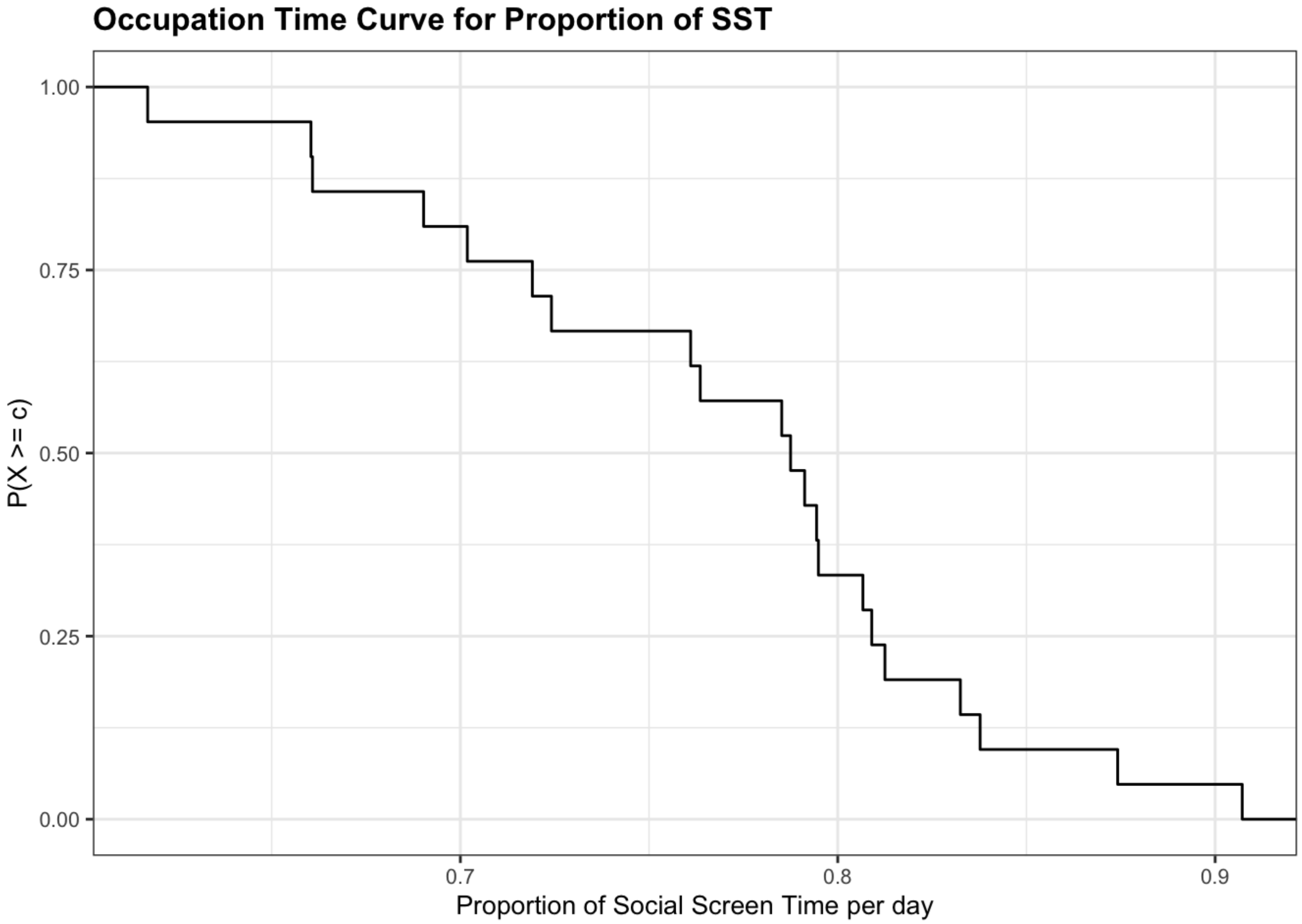


Occupation Time Curve for Social Screen Time

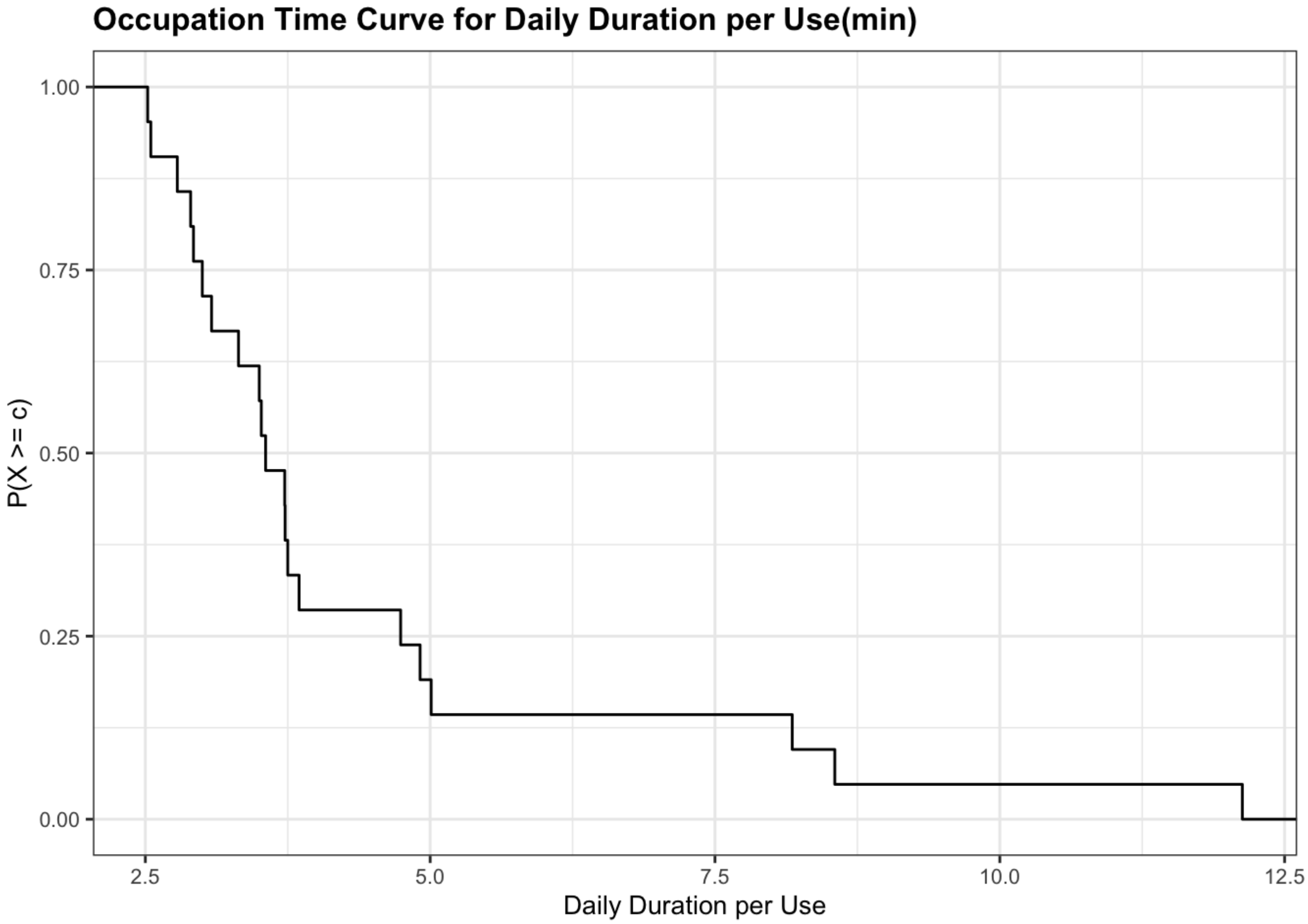
```
occupation_time_curve_pks <- ggplot(hw1data, aes(x = Pickups)) +
  labs(title = "Occupation Time Curve for Pickups",
       x = "Number of Pickups",
       y = "P(X >= c)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))
occupation_time_curve_pks + stat_ecdf(geom = "step", aes(y = 1 - ..y..))
```



```
occupation_time_curve_prop <- ggplot(hw1data, aes(x = Social.ST.prop)) +
  labs(title = "Occupation Time Curve for Proportion of SST",
       x = "Proportion of Social Screen Time per day",
       y = "P(X >= c)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))
occupation_time_curve_prop + stat_ecdf(geom = "step", aes(y = 1 - ..y..))
```



```
occupation_time_curve_duration <- ggplot(hw1data, aes(x = Duration.per.use)) +
  labs(title = "Occupation Time Curve for Daily Duration per Use(min)",
       x = "Daily Duration per Use",
       y = "P(X >= c)") +
  theme_bw() +
  theme(plot.title = element_text(size = 12, face = "bold"),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 8))
occupation_time_curve_duration + stat_ecdf(geom = "step", aes(y = 1 - ..y..))
```



d.

```
acf(hw1data$Total.ST.min, plot = FALSE, lag.max = 21)
```

```
##
## Autocorrelations of series 'hw1data$Total.ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.079 -0.228 -0.065 -0.176 -0.044  0.062 -0.122 -0.123 -0.005 -0.077
##     11     12     13     14     15     16     17     18     19     20
## -0.022  0.090  0.101  0.066  0.111 -0.008  0.005 -0.038 -0.149  0.043
```

```
acf(hw1data$Social.ST.min, plot = FALSE, lag.max = 21)
```

```
##
## Autocorrelations of series 'hw1data$Social.ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000 -0.054 -0.201 -0.021 -0.056 -0.155  0.099  0.045 -0.240  0.033 -0.045
##     11     12     13     14     15     16     17     18     19     20
##  0.091 -0.068  0.091  0.068  0.086 -0.089  0.073  0.006 -0.190  0.027
```

```
acf(hw1data$Pickups, plot = FALSE, lag.max = 21)
```

```
## 
## Autocorrelations of series 'hw1data$Pickups', by lag
## 
##      0      1      2      3      4      5      6      7      8      9     10 
##  1.000  0.360 -0.212 -0.665 -0.423 -0.094  0.356  0.407  0.229 -0.123 -0.271 
##     11     12     13     14     15     16     17     18     19     20 
## -0.196  0.013  0.002  0.103  0.079  0.027 -0.054 -0.027 -0.010 -0.001
```

```
acf(hw1data$Social.ST.prop, plot = FALSE, lag.max = 21)
```

```
## 
## Autocorrelations of series 'hw1data$Social.ST.prop', by lag
## 
##      0      1      2      3      4      5      6      7      8      9     10 
##  1.000 -0.304 -0.014  0.012  0.227 -0.157 -0.085  0.152  0.084 -0.325 -0.067 
##     11     12     13     14     15     16     17     18     19     20 
##  0.230 -0.177 -0.004  0.037 -0.003 -0.100 -0.061  0.124 -0.007 -0.064
```

```
acf(hw1data$Duration.per.use, plot = FALSE, lag.max = 21)
```

```
## 
## Autocorrelations of series 'hw1data$Duration.per.use', by lag
## 
##      0      1      2      3      4      5      6      7      8      9     10 
##  1.000 -0.016 -0.172 -0.231 -0.187 -0.084  0.434  0.029 -0.019 -0.102 -0.188 
##     11     12     13     14     15     16     17     18     19     20 
## -0.047  0.230 -0.008  0.016  0.014  0.003 -0.042 -0.041 -0.100  0.012
```

There are no obvious autocorrelation.

Problem 3

a.

```
hw1data <- hw1data %>%
  mutate(Pickup.1st = as.POSIXct(Pickup.1st, format = "%H:%M")) %>%
  mutate(Pickup.1st.angular=(hour(Pickup.1st)*60+minute(Pickup.1st))/(24*60)*360)
hw1data$Pickup.1st.angular
```

```
##  [1]  15.75  41.75   9.50  19.75   0.00 134.50   3.75   9.50   4.50 108.75
## [11]  98.75 109.75 120.50   7.75   4.00   4.75   9.75 119.00 111.00   5.50
## [21]  23.25
```

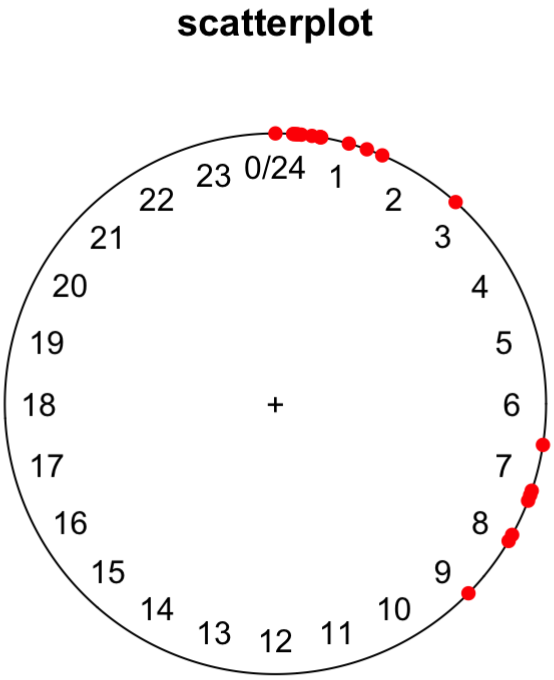
b.

```
library(circular)
```

```
##
## Attaching package: 'circular'
```

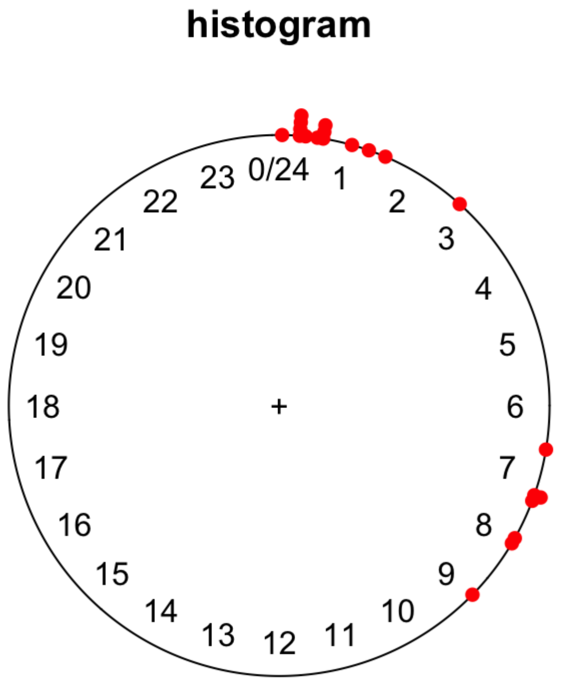
```
## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
first.pickup.circle <- circular(hw1data$Pickup.1st.angular,
                                units = "degrees",
                                template = "clock24")
plot(first.pickup.circle ,col = "red", main = 'scatterplot')
```



c.

```
plot(first.pickup.circle, stack = TRUE, bins = 288, col = "red", main = 'histogram')
```



histogram

Problem 4

a. St acts as a standardization here, when the daily total screen time is to low or to high, it will convert it to other unit, which will narrow the scale, and satisfies the assumption of Poisson Distribution.

b.

```
model <- glm(Pickups ~ offset(log(hw1data$Total.ST.min/60)), family = "poisson", data =
hw1data)
summary(model)
```

```
##
## Call:
## glm(formula = Pickups ~ offset(log(hw1data$Total.ST.min/60)),
##     family = "poisson", data = hw1data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -11.0053   -2.1042    0.9332    3.2889    5.3142
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.7340     0.0199   137.4   <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: 354.47  on 20  degrees of freedom
## Residual deviance: 354.47  on 20  degrees of freedom
## AIC: 494.72
##
## Number of Fisher Scoring iterations: 4
```

```
lambda <- exp(coef(model)[1])
```

c.

```
library(lubridate)
hw1data$Xt <- ifelse(wday(hw1data$Date) %in% c(2,3,4,5,6), 1, 0)
hw1data$Zt <- ifelse(hw1data$Date >= as.Date('2024-01-10'), 1, 0)
model2 <- glm(Pickups ~ Xt + Zt + offset(log(Total.ST.min/60)), family = poisson, data =
hw1data)
summary(model2)
```

```
##
## Call:
## glm(formula = Pickups ~ Xt + Zt + offset(log(Total.ST.min/60)),
##     family = poisson, data = hw1data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.6184  -0.8953   0.9110   1.9982   4.0945
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.43453    0.05258  46.302   <2e-16 ***
## Xt           0.52216    0.04907  10.641   <2e-16 ***
## Zt          -0.09972    0.05151  -1.936   0.0529 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 354.47  on 20  degrees of freedom
## Residual deviance: 230.77  on 18  degrees of freedom
## AIC: 375.02
##
## Number of Fisher Scoring iterations: 4
```

(c.1) There is significant evidence that I have more pickups on weekdays than weekends because p<2e-16

(c.2) There is no significant evidence that I have had more pickups after the winter semester began since p = 0.0529 > 0.05

5

a.

```
parameter <- mle.vonmises((hw1data$Pickup.1st.angular*pi)/180)
```

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

```
print(parameter)
```

```
##
## Call:
## mle.vonmises(x = (hw1data$Pickup.1st.angular * pi)/180)
##
## mu: 0.6997  ( 0.2005 )
##
## kappa: 1.793  ( 0.4942 )
```

b.

```
x = (8*60+30)/(24*60)*360
x_pi = x*pi/180
1 - pvonmises(x_pi, mu = parameter$mu, kappa = parameter$kappa)
```

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

```
## [1] 0.05112026
```

The probability that the first pickup is later than 8:30AM is 0.0511