Homework1_620

Zihan Wang

2024-02-04

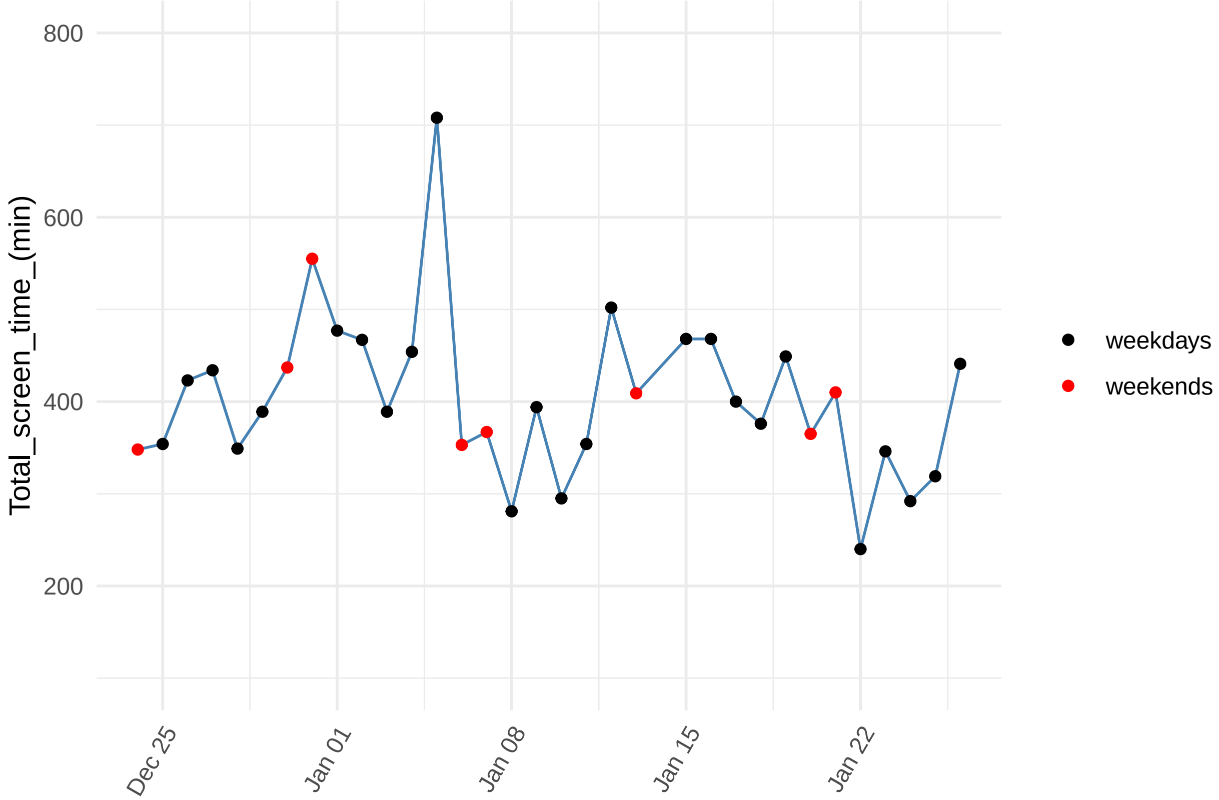
```r
# 1(a)

# Hypothesis: High amount of screen time is associated with high levels of
# depressive symptoms

# Tang, S., Werner-Seidler, A., Torok, M., Mackinnon, A. J., & Christensen, H.
# (2021). The relationship between screen time and mental health in young
# people: A systematic review of longitudinal studies. Clinical psychology
# review, 86, 102021. https://doi.org/10.1016/j.cpr.2021.102021

# 1(b)

# Informed consent form ensures that participants are fully aware of the
# studies they are invoveld in regard of their purposes, procedures,risks and
# benefits and also the right to withdraw from the study and the
# confidentiality of the raw research data.


# 1(c)

# Screen activity is recorded in real-time by the mobile device. We collected
# daily entries of total screen time (Total.ST: total screen time in HH-MM
# format, and Total.ST.min: total screen time in MM format), social app screen
# time (Social.ST: social app screen time in HH-MM format, and Social.ST.min:
# social app screen time in MM format),total number of times the user picked up
# the phone (Pickups), and the time of the first pick-up (Pickup.1st). We are
# using the screen time data from 12/24/2023 to 01/26/2024 stored in the mobile
# device.
```

```r
Dataframe <- read_excel("Screen_Time.xlsx")
```

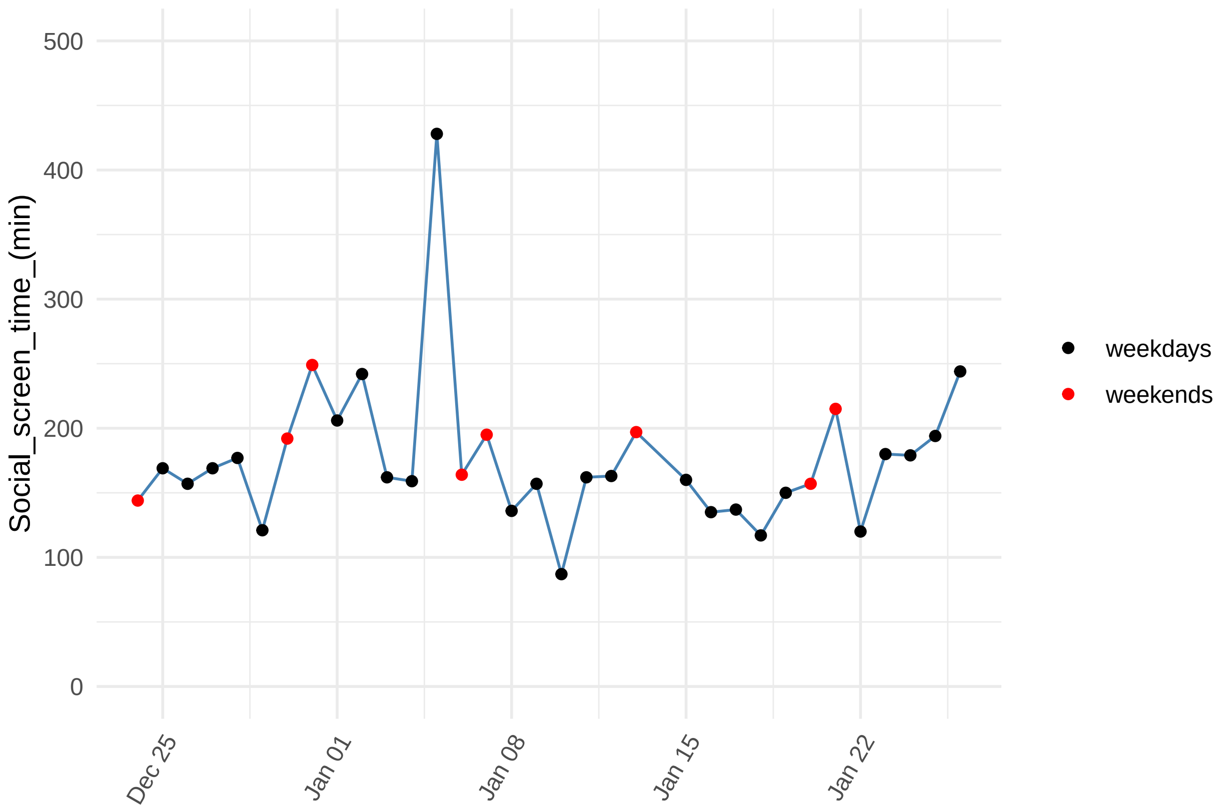
```r
Dataframe$Pickup.1st <- sapply(Dataframe$Pickup.1st, function(x) format(x, "%H:%M"))

Dataframe <- na.omit(Dataframe)
```

```r
# 1(d)
Dataframe$Proportion_Social = Dataframe$Social.ST.min/Dataframe$Total.ST.min
Dataframe$Duration = Dataframe$Total.ST.min/Dataframe$Pickups
Dataframe$Date <- as.Date(Dataframe$Date, format = "%m/%d/%Y")
Dataframe$if_weekend <- ifelse(weekdays(Dataframe$Date) %in% c("Saturday", "Sunday"),
    "Weekend", "Weekday")
```

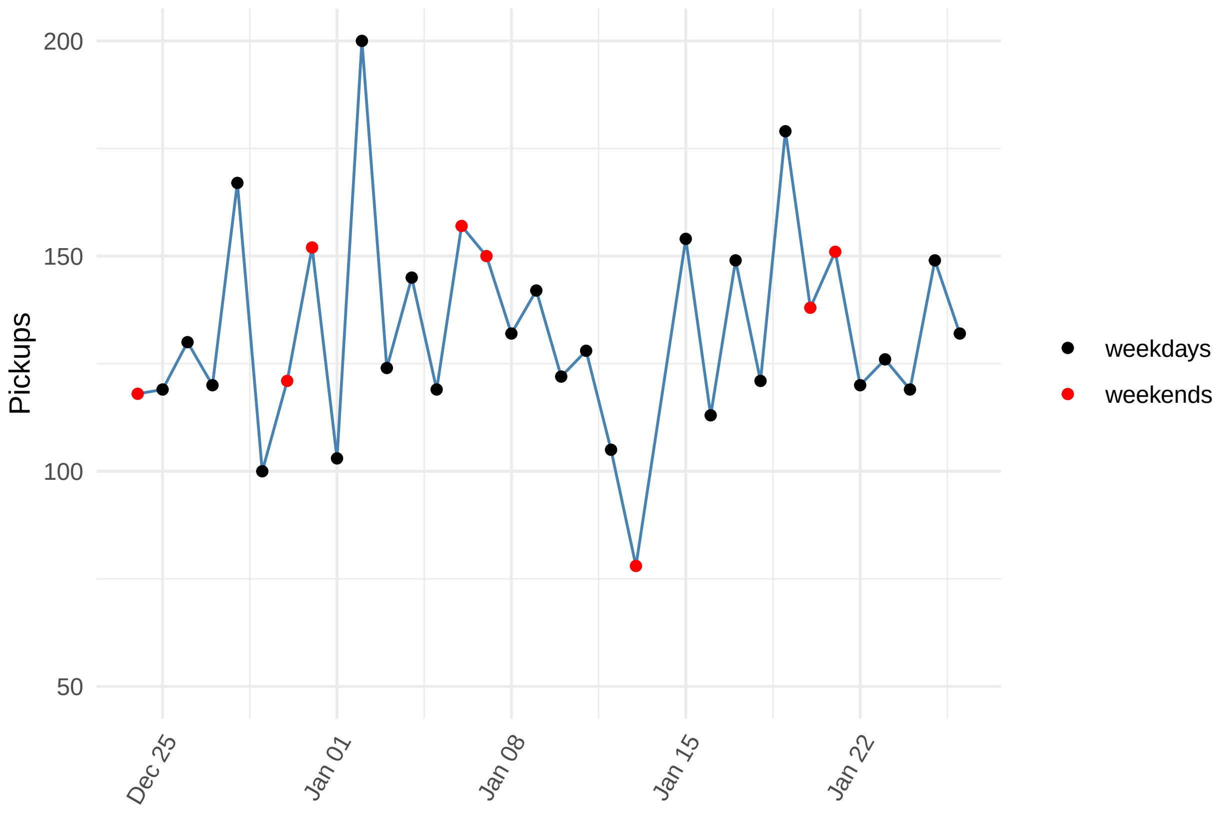
```
# 2(a)

# The data collected during weekend days are colored in red in order to
# visualize any differences of mobile device use between weekdays and weekends.
# It seems that there is no obvious difference of the temporal pattern between
# weekdays and weekends in regard of five variables.


total = ggplot(Dataframe, aes(x = Date, y = Total.ST.min, color = if_weekend)) +
    geom_line(color = "steelblue") + geom_point() + xlab("") + ylab("Total_screen_time_(min)") +
    ylim(100, 800) + scale_color_manual(labels = c("weekdays", "weekends"), values = c("black",
    "red")) + theme_minimal() + theme(axis.text.x = element_text(angle = 60, hjust = 1),
    legend.title = element_blank())
print(total)
```



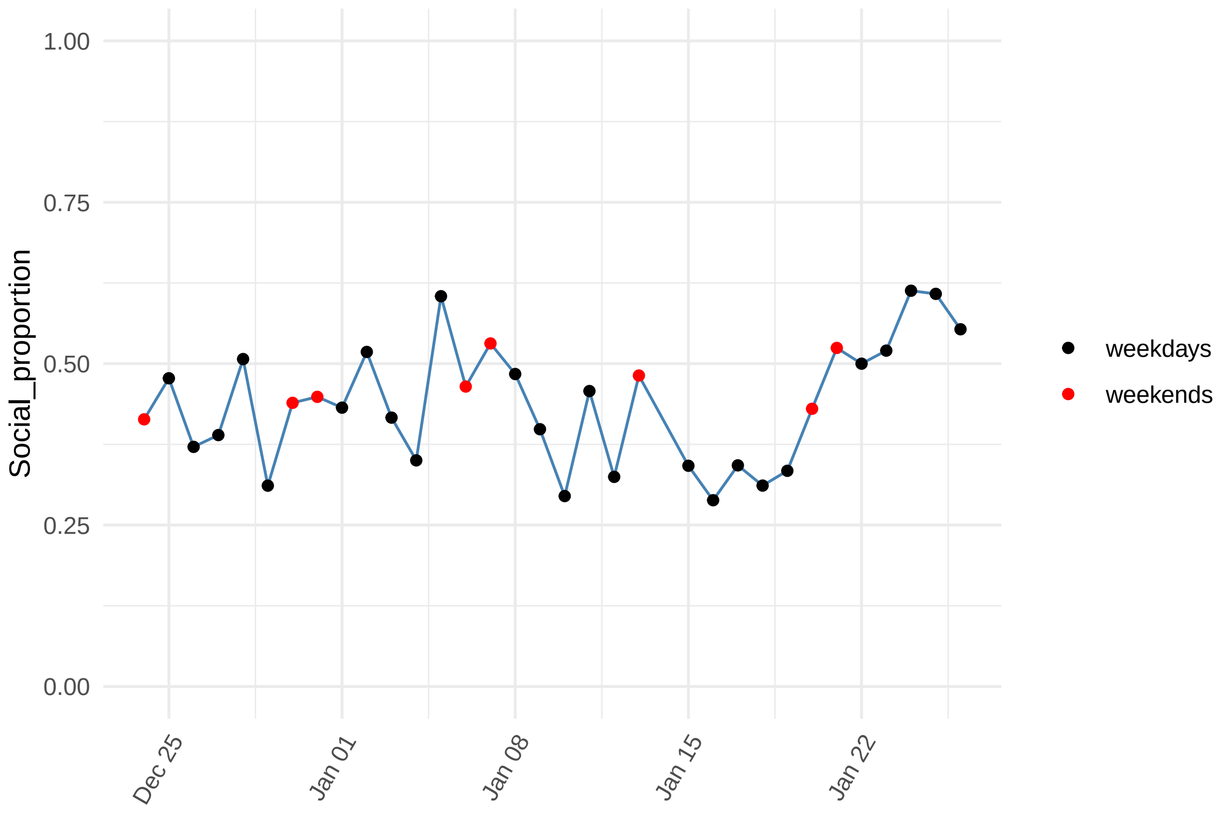
```
social = ggplot(Dataframe, aes(x = Date, y = Social.ST.min, color = if_weekend)) +
    geom_line(color = "steelblue") + geom_point() + xlab("") + ylab("Social_screen_time_(min)") +
    ylim(0, 500) + scale_color_manual(labels = c("weekdays", "weekends"), values = c("black",
    "red")) + theme_minimal() + theme(axis.text.x = element_text(angle = 60, hjust = 1),
    legend.title = element_blank())
print(social)
```



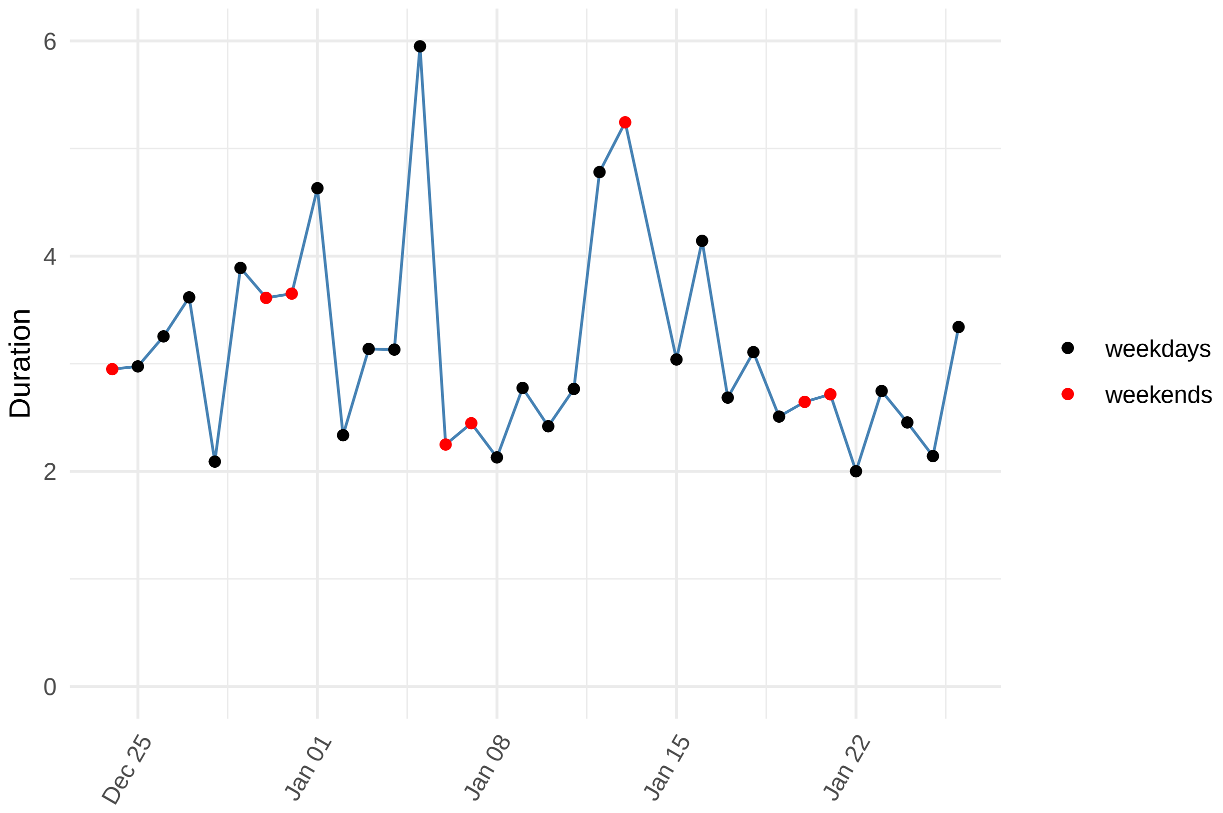
```
pickup = ggplot(Dataframe, aes(x = Date, y = Pickups, color = if_weekend)) + geom_line(color = "steelblu
    geom_point() + xlab("") + ylab("Pickups") + ylim(50, 200) + scale_color_manual(labels = c("weekdays
    "weekends"), values = c("black", "red")) + theme_minimal() + theme(axis.text.x = element_text(angle
    hjust = 1), legend.title = element_blank())
print(pickup)
```



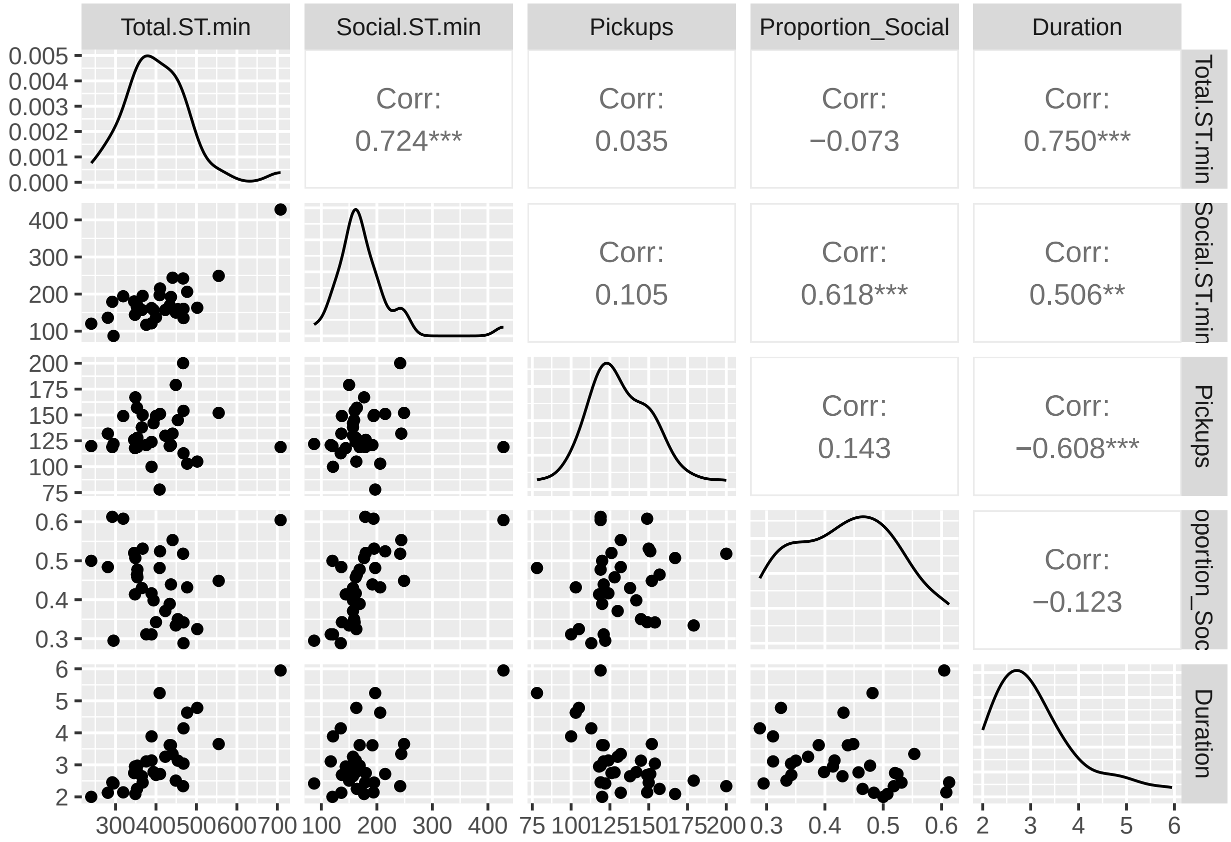
```
social_proportion = ggplot(Dataframe, aes(x = Date, y = Proportion_Social, color = if_weekend)) +
    geom_line(color = "steelblue") + geom_point() + xlab("") + ylab("Social_proportion") +
    ylim(0, 1) + scale_color_manual(labels = c("weekdays", "weekends"), values = c("black",
    "red")) + theme_minimal() + theme(axis.text.x = element_text(angle = 60, hjust = 1),
    legend.title = element_blank())
print(social_proportion)
```



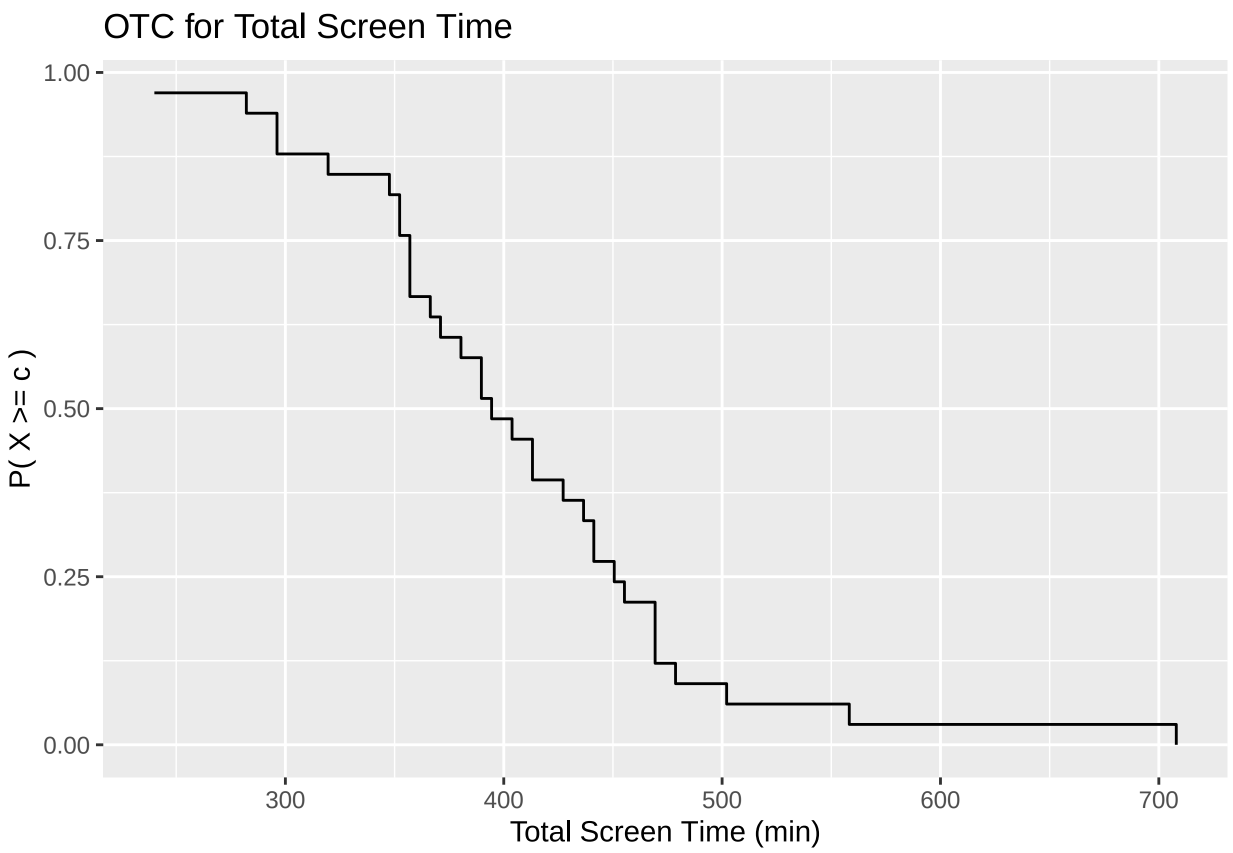
```
duration = ggplot(Dataframe, aes(x = Date, y = Duration, color = if_weekend)) + geom_line(color = "steel
    geom_point() + xlab("") + ylab("Duration") + ylim(0, 6) + scale_color_manual(labels = c("weekdays",
    "weekends"), values = c("black", "red")) + theme_minimal() + theme(axis.text.x = element_text(angle
    hjust = 1), legend.title = element_blank())
print(duration)
```



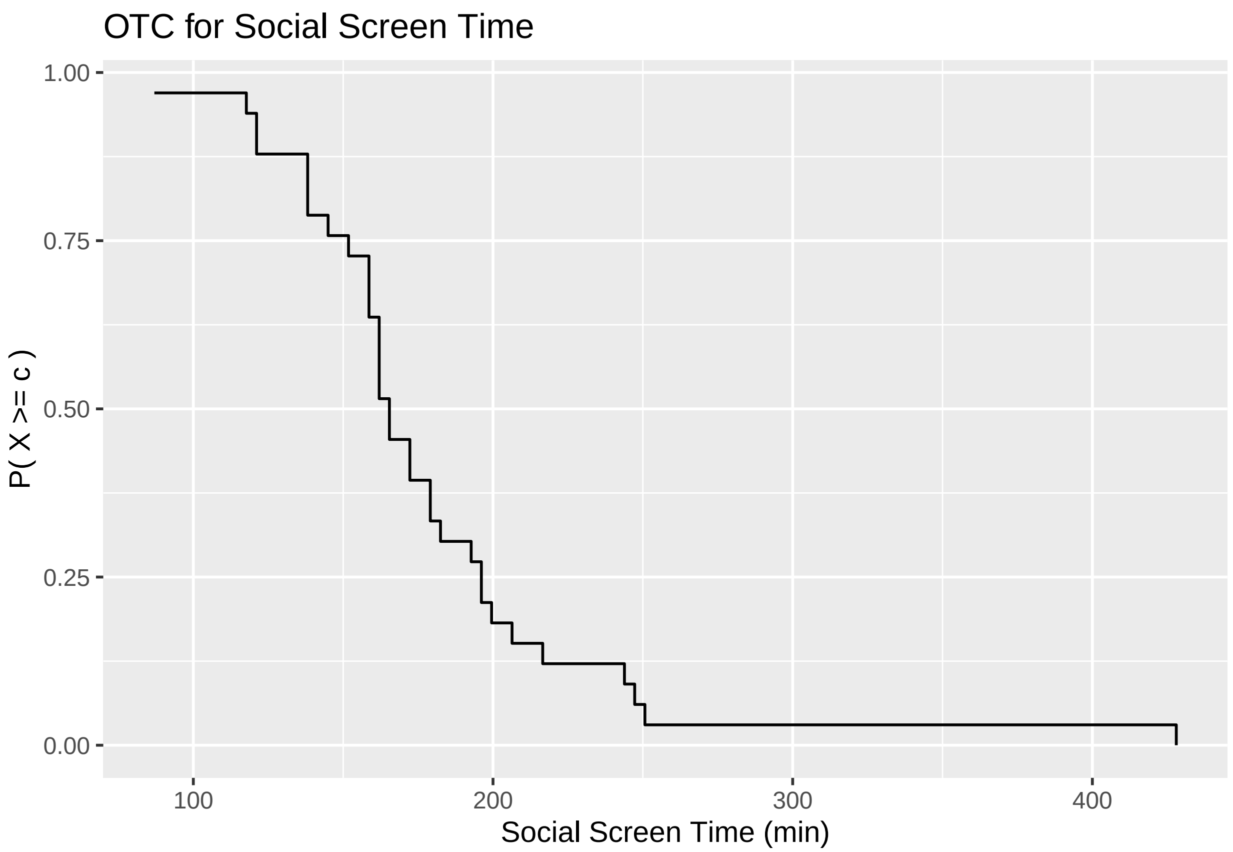
```
# 2(b)

# The pair of daily duration per use and total screen time has the highest
# correlation.It is also of worth to point out that the daily proportion of
# social screen time and total screen time are nearly uncorrelated, the number
# of pickup times and social screen time are nearly uncorrelated, and the daily
# duration per use and the daily proportion of social screen time are nealy
# uncorrelated.

ggpairs(Dataframe, columns = c("Total.ST.min", "Social.ST.min", "Pickups", "Proportion_Social",
    "Duration"))
```



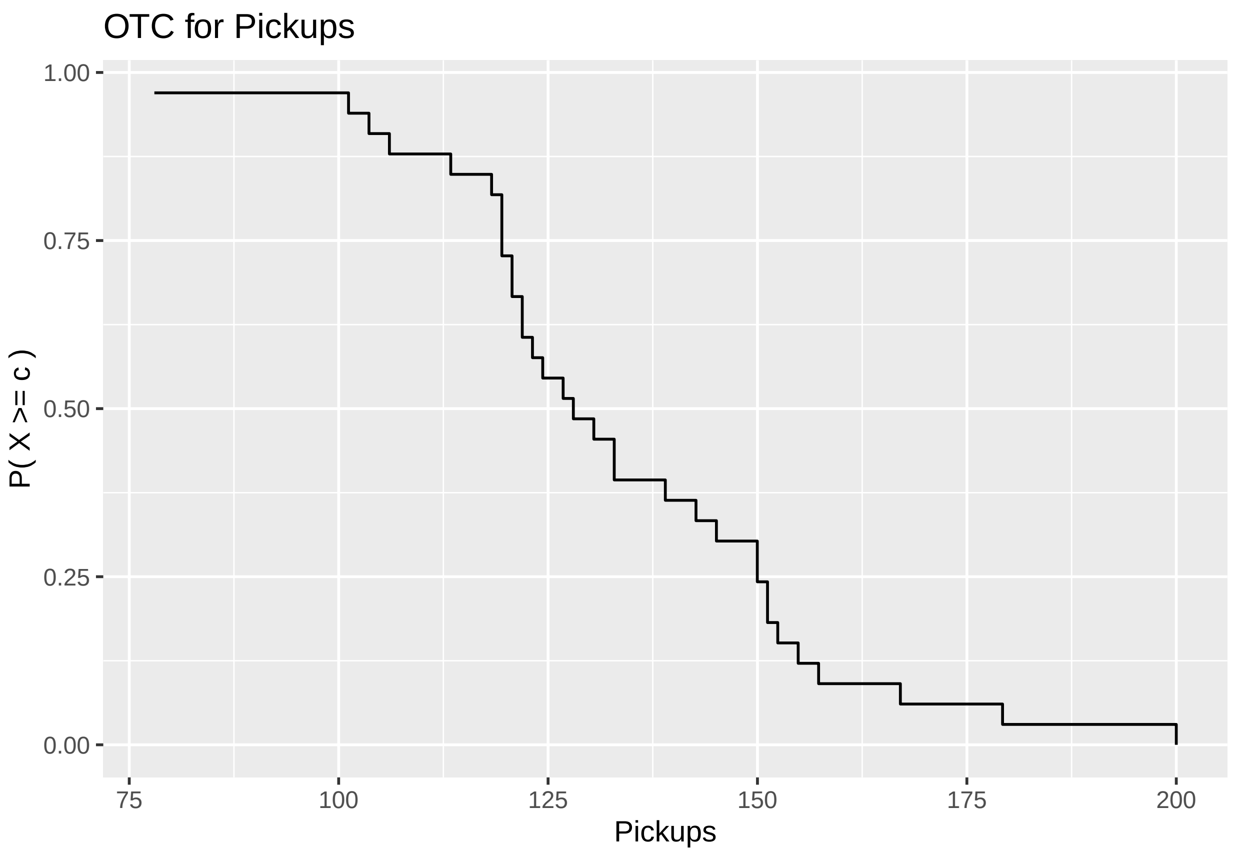
```
# 2(c)

# For total screen time, social screen time and pickups after the patterns are
# quite similar, that is after a certain point the curves drop dramatically,
# indicating that high screen time, high social screen time and high pickups
# are relatively less common.While for the daily proportion of the social
# screen time and daily duration per use the curves are relatively smooth.

ggplot(Dataframe, aes(x = Total.ST.min)) + stat_function(fun = function(x) 1 - ecdf(Dataframe$Total.ST.
    geom = "step") + labs(title = "OTC for Total Screen Time", x = "Total Screen Time (min)",
    y = "P( X >= c )")
```



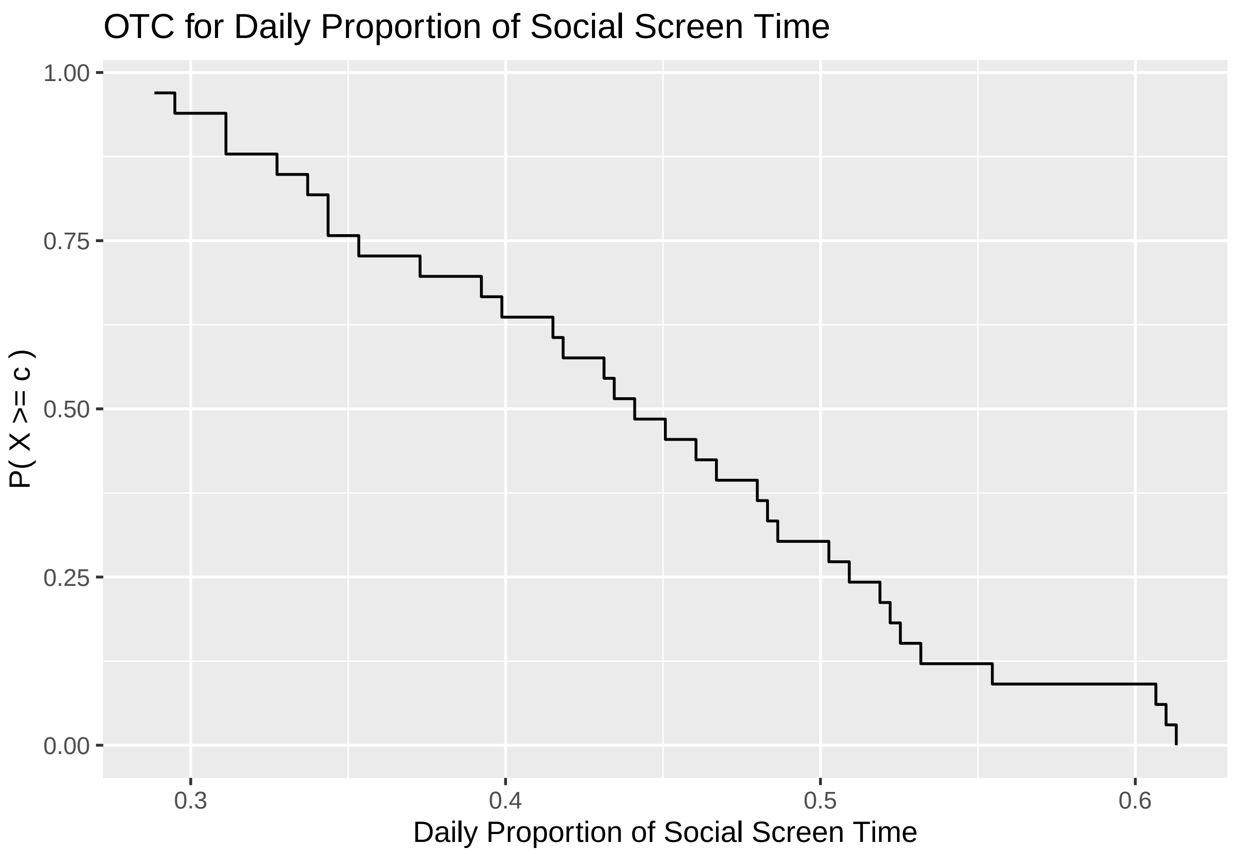
```
ggplot(Dataframe, aes(x = Social.ST.min)) + stat_function(fun = function(x) 1 - ecdf(Dataframe$Social.S
    geom = "step") + labs(title = "OTC for Social Screen Time", x = "Social Screen Time (min)",
    y = "P( X >= c )")
```



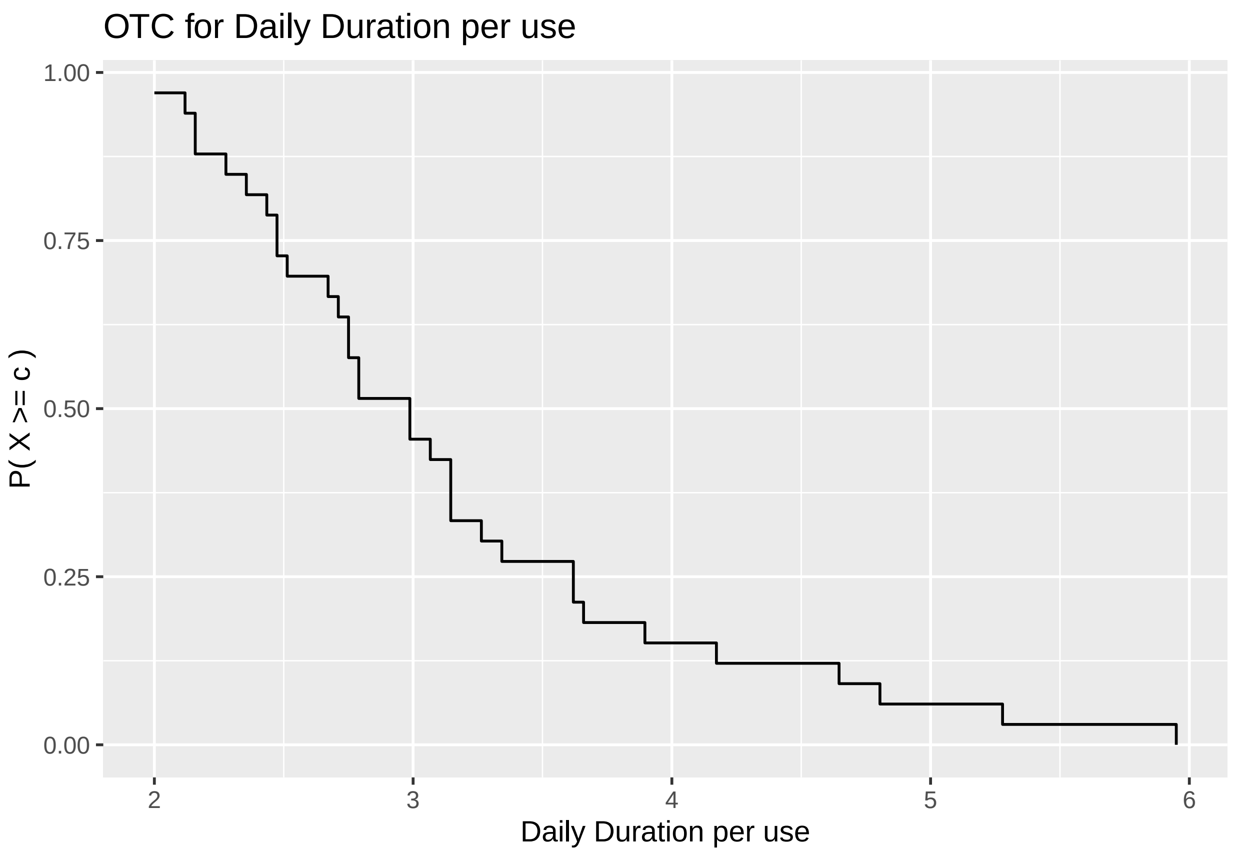
```
ggplot(Dataframe, aes(x = Pickups)) + stat_function(fun = function(x) 1 - ecdf(Dataframe$Pickups)(x),
    geom = "step") + labs(title = "OTC for Pickups", x = "Pickups", y = "P( X >= c )")
```



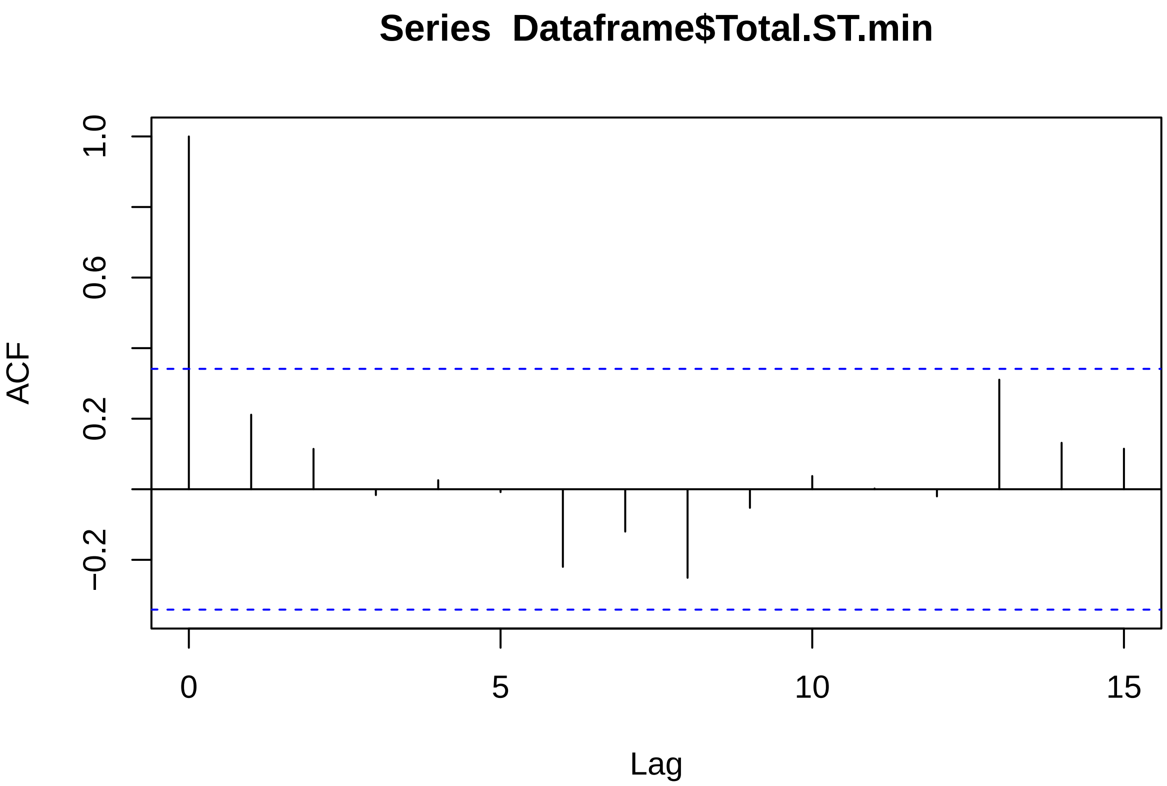
```
ggplot(Dataframe, aes(x = Proportion_Social)) + stat_function(fun = function(x) 1 -
    ecdf(Dataframe$Proportion_Social)(x), geom = "step") + labs(title = "OTC for Daily Proportion of Soc
    x = "Daily Proportion of Social Screen Time", y = "P( X >= c )")
```



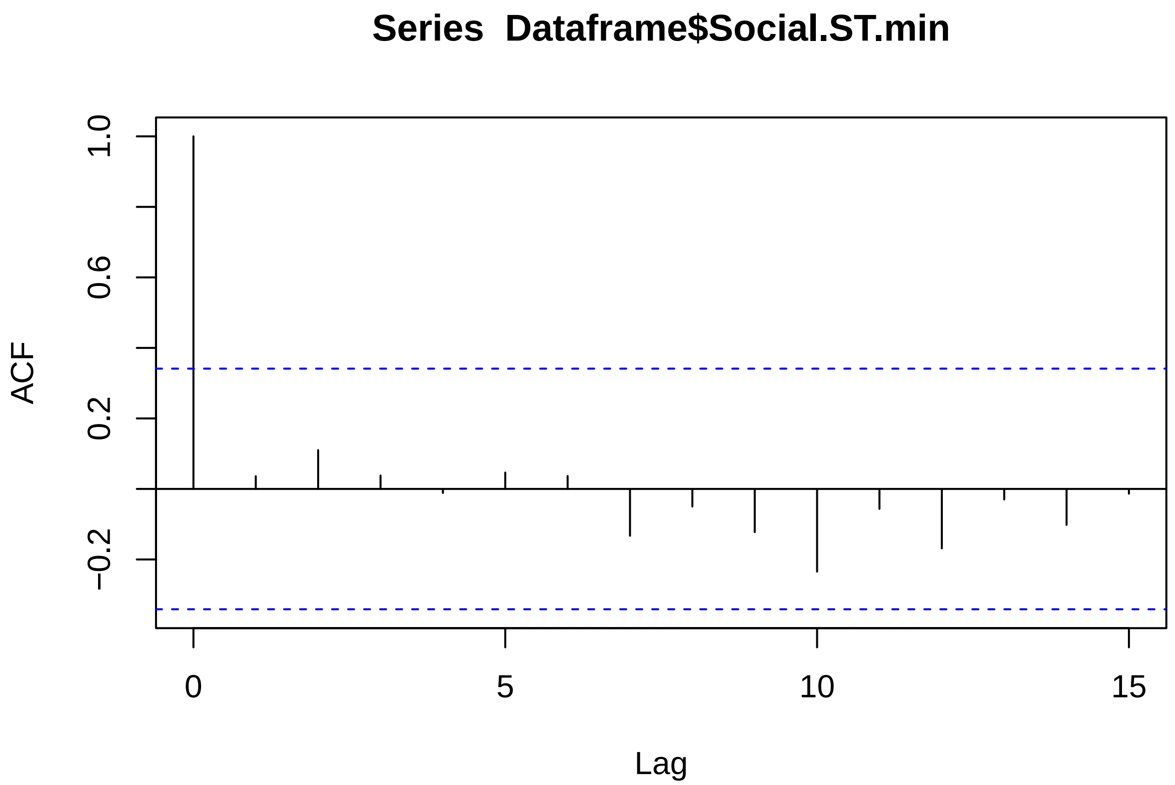
```
ggplot(Dataframe, aes(x = Duration)) + stat_function(fun = function(x) 1 - ecdf(Dataframe$Duration)(x),
    geom = "step") + labs(title = "OTC for Daily Duration per use", x = "Daily Duration per use",
    y = "P( X >= c )")
```



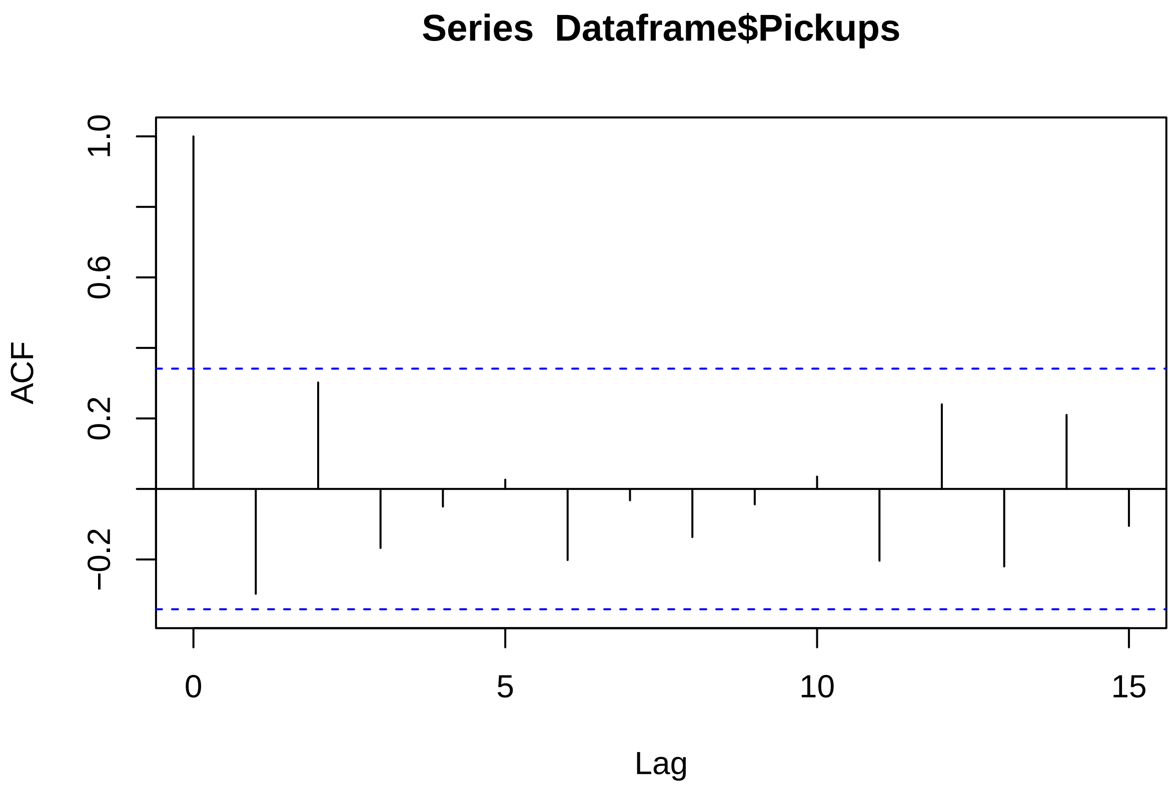
```
# 2(d)

# From the acf plot we can infer the only possible autocorrelation is in regard
# of daily proportion of the social screen time.

acf(Dataframe$Total.ST.min)
```



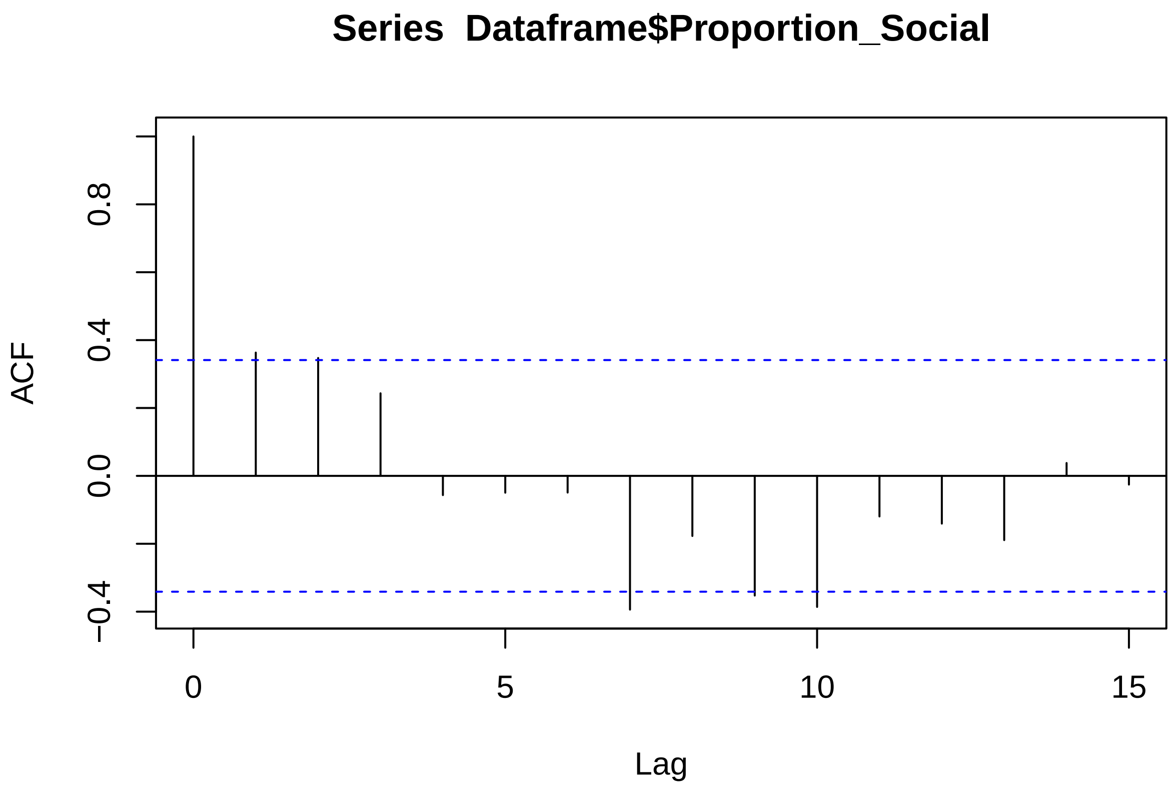
```
acf(Dataframe$Social.ST.min)
```



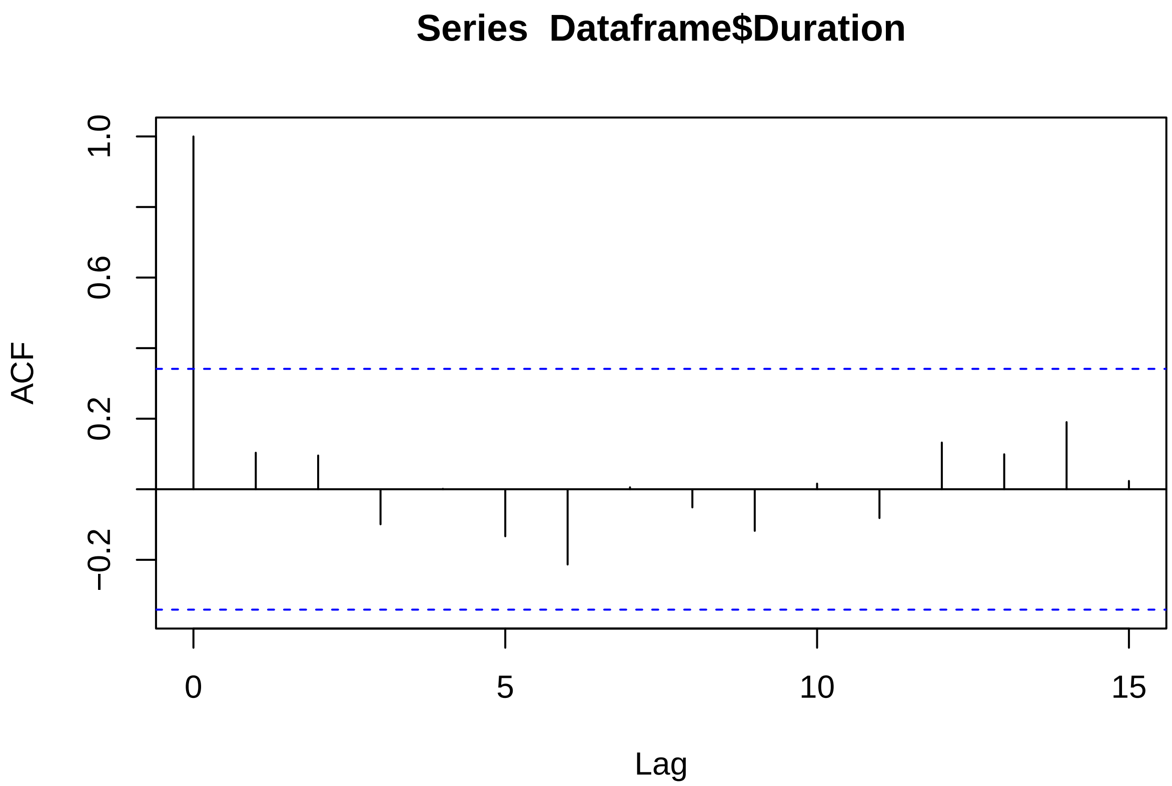
```
acf(Dataframe$Pickups)
```



```
acf(Dataframe$Proportion_Social)
```



```
acf(Dataframe$Duration)
```



```
acf(Dataframe$Total.ST.min, plot = FALSE)
```

```
##
## Autocorrelations of series 'Dataframe$Total.ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.211  0.114 -0.017  0.026 -0.008 -0.220 -0.120 -0.251 -0.053  0.038
##     11     12     13     14     15
##  0.002 -0.020  0.311  0.132  0.115
```

```
acf(Dataframe$Social.ST.min, plot = FALSE)
```

```
##
## Autocorrelations of series 'Dataframe$Social.ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.036  0.110  0.038 -0.011  0.046  0.037 -0.133 -0.050 -0.122 -0.235
##     11     12     13     14     15
## -0.056 -0.168 -0.030 -0.102 -0.014
```

```
acf(Dataframe$Pickups, plot = FALSE)
```

```
##
## Autocorrelations of series 'Dataframe$Pickups', by lag
```

```
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000 -0.297  0.302 -0.168 -0.050  0.026 -0.202 -0.032 -0.137 -0.044  0.035
##     11     12     13     14     15
## -0.204  0.240 -0.220  0.210 -0.105
```

```
acf(Dataframe$Proportion_Social, plot = FALSE)
```

```
##
## Autocorrelations of series 'Dataframe$Proportion_Social', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.363  0.347  0.243 -0.057 -0.050 -0.049 -0.394 -0.177 -0.353 -0.386
##     11     12     13     14     15
## -0.120 -0.141 -0.189  0.038 -0.026
```

```
acf(Dataframe$Duration, plot = FALSE)
```

```
##
## Autocorrelations of series 'Dataframe$Duration', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.104  0.096 -0.099  0.001 -0.133 -0.213  0.005 -0.051 -0.118  0.016
##     11     12     13     14     15
## -0.082  0.132  0.099  0.190  0.024
```

```
# 3(a)

Dataframe$Pickup.1st <- as.POSIXct(Dataframe$Pickup.1st, format = "%H:%M")

Dataframe = Dataframe %>%
    mutate(Pickup.1st.angular = (hour(Pickup.1st) * 60 + minute(Pickup.1st))/(24 *
        60) * 360)

# 3(b)

# The first pickup time is mainly concentrated between 12am and 1am.

first.pickup.cir = circular(Dataframe$Pickup.1st.angular, units = "degrees", template = "clock24")
png("HD1_C3(b).png")
plot(first.pickup.cir, col = "blue")
dev.off()
```

```
## pdf
##   2
```

```
# 3(c)

# I choose bin size to be 48, which corresponds to an interval of half an hour.

png("HD1_C3(c).png")
plot(first.pickup.cir, stack = TRUE, bins = 48, col = "red")
dev.off()
```

```
## pdf 
##   2
```

```
# 4(a)

# The Yt is the daily number of pickups, but the rate lambda is the expected
# hourly rate, there exists an inconsistency between the unit.

# 4(b)
Dataframe$Total.ST.Hour = Dataframe$Total.ST.min/60

model = glm(Pickups ~ offset(log(Total.ST.Hour)), family = "poisson", data = Dataframe)

lambda = exp(coef(model))
print(lambda)
```

```
## (Intercept) 
##    19.75362
```

```
# 4(c)

# (c.1) the p-value for Xt is 0.9139, so there is no evidence of significantly
# different behavior of daily pickups between weekdays and weekends using the
# significance level alpha=0.05

# (c.2)the p-value for Zt is 0.0506, so there is no evidence of significantly
# different behavior of daily pickups between whether or not being days on
# winter break.

Dataframe$if_break = c(rep(0, 17), rep(1, 16))
Dataframe$if_weekend_dummy = ifelse(Dataframe$if_weekend == "Weekday", 1, 0)

model_ <- glm(Pickups ~ if_break + if_weekend_dummy, family = poisson(link = "log"),
    data = Dataframe, offset = log(Total.ST.Hour))

summary(model_)
```

```
## 
## Call:
## glm(formula = Pickups ~ if_break + if_weekend_dummy, family = poisson(link = "log"), 
##     data = Dataframe, offset = log(Total.ST.Hour))
## 
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       2.95839    0.03274  90.361   <2e-16 ***
## if_break          0.05949    0.03043   1.955   0.0506 .  
## if_weekend_dummy -0.00383    0.03543  -0.108   0.9139    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 306.43  on 32  degrees of freedom
## Residual deviance: 302.60  on 30  degrees of freedom
## AIC: 530.11
##
## Number of Fisher Scoring iterations: 4
```

```
# 5(a)

Dataframe$Pickup.1st.pi = (Dataframe$Pickup.1st.angular/180) * pi
mle_result <- mle.vonmises(Dataframe$Pickup.1st.pi)
```

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

```
print(mle_result)
```

```
##
## Call:
## mle.vonmises(x = Dataframe$Pickup.1st.pi)
##
## mu: 0.09784  ( 0.04751 )
##
## kappa: 13.94  ( 3.363 )
```

```
# 5(b)
pvonmises_result <- pvonmises((127.5/180) * pi, mu = 0.09784, kappa = 13.94)
```

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

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularmuradians0counter
```

```
# The return value of the pvonmises is nearly one, which means the probability
# of my first pickup being 8:30 AM or later is nearly 0
```