Biostat 620- hw01 code

wenfei mao

2024-02-04

problem 2

(a) Make a time series plot of each of the five variables in your data

```
#Time series plot of daily total screen time
Sys.setlocale("LC_TIME", "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8"
```

```
library(xts)
```

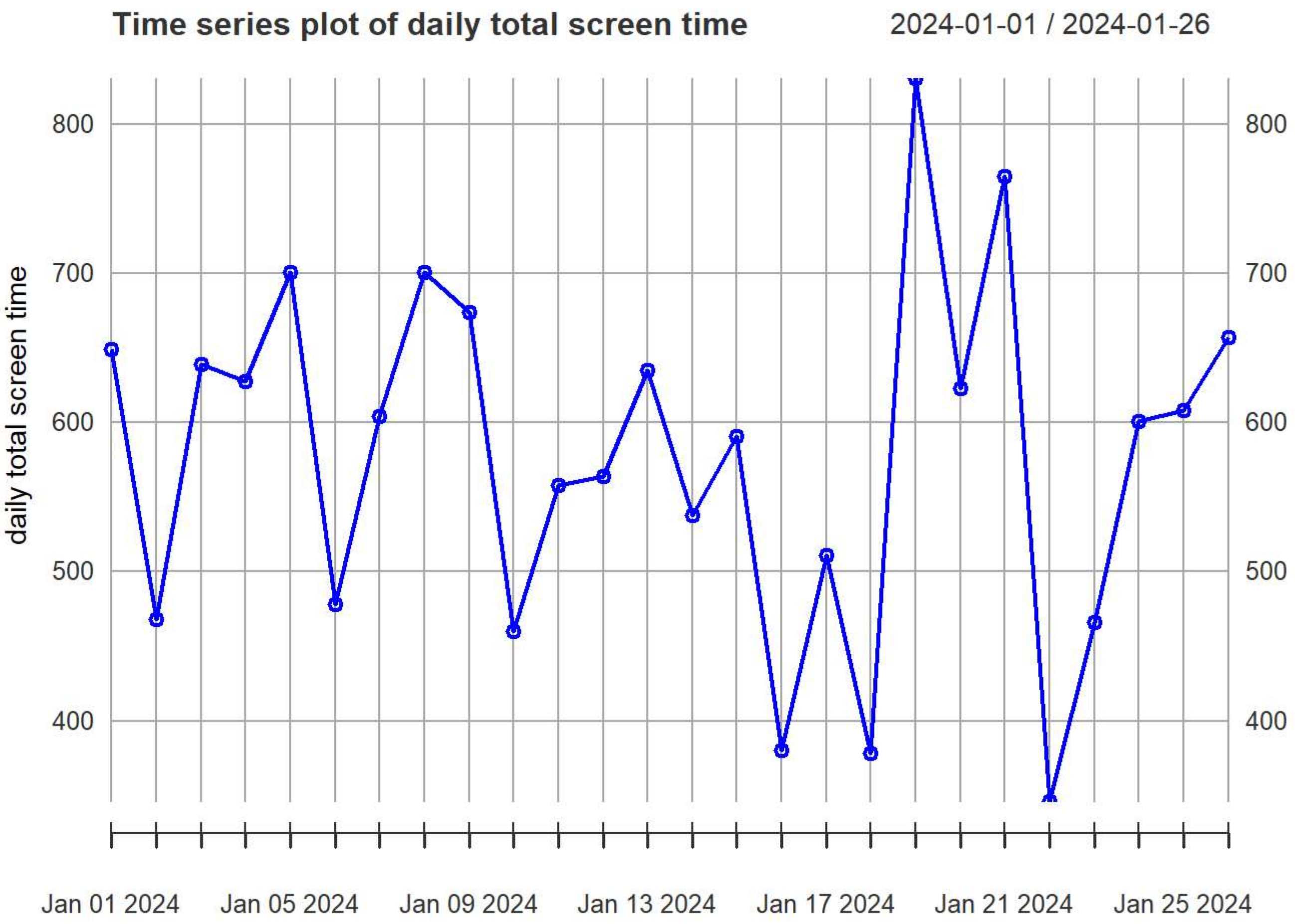
```
## Warning: 程辑包'xts'是用R版本4.3.2 来建造的
```

```
## 载入需要的程辑包：zoo
```

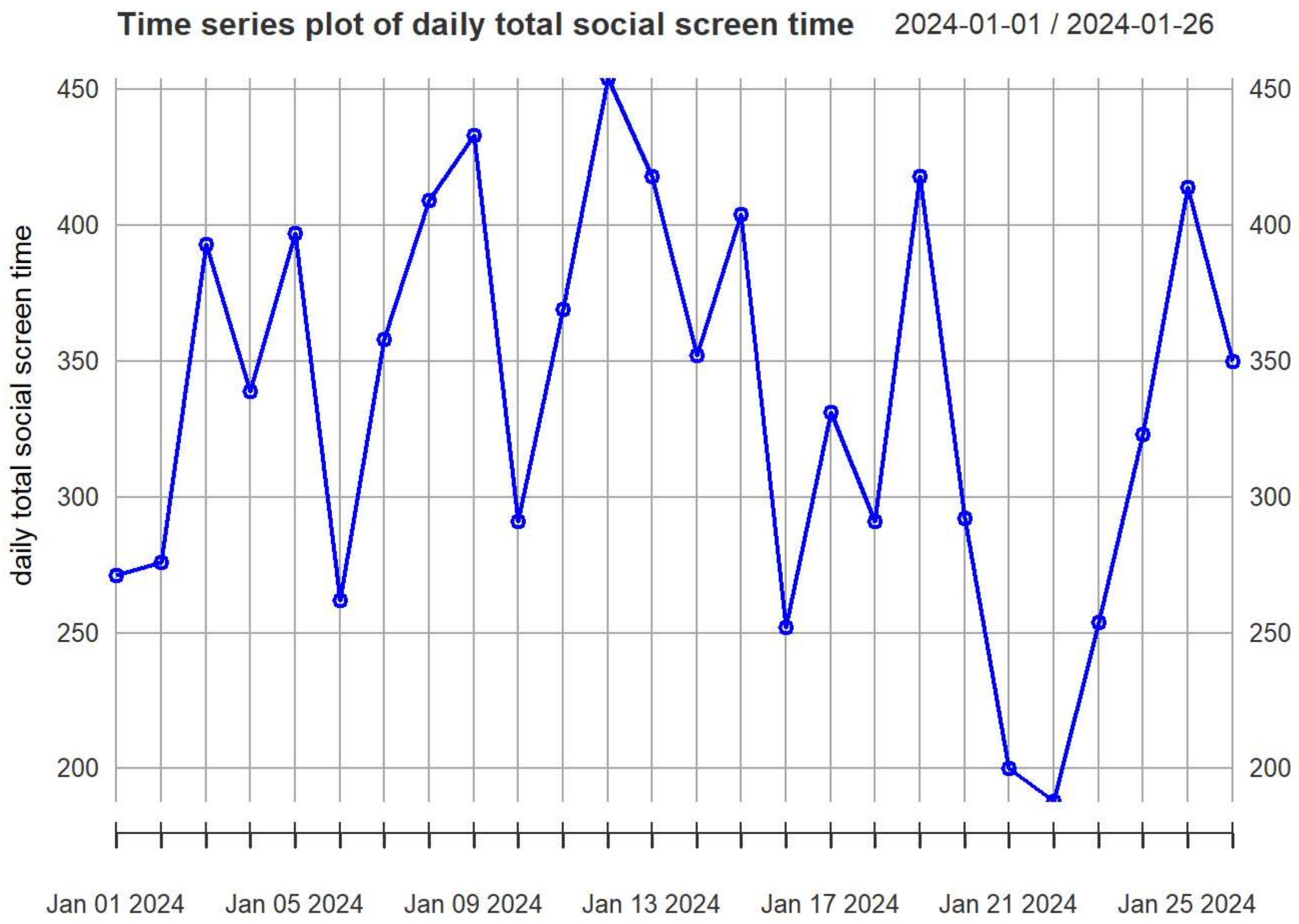
```
##
## 载入程辑包：'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

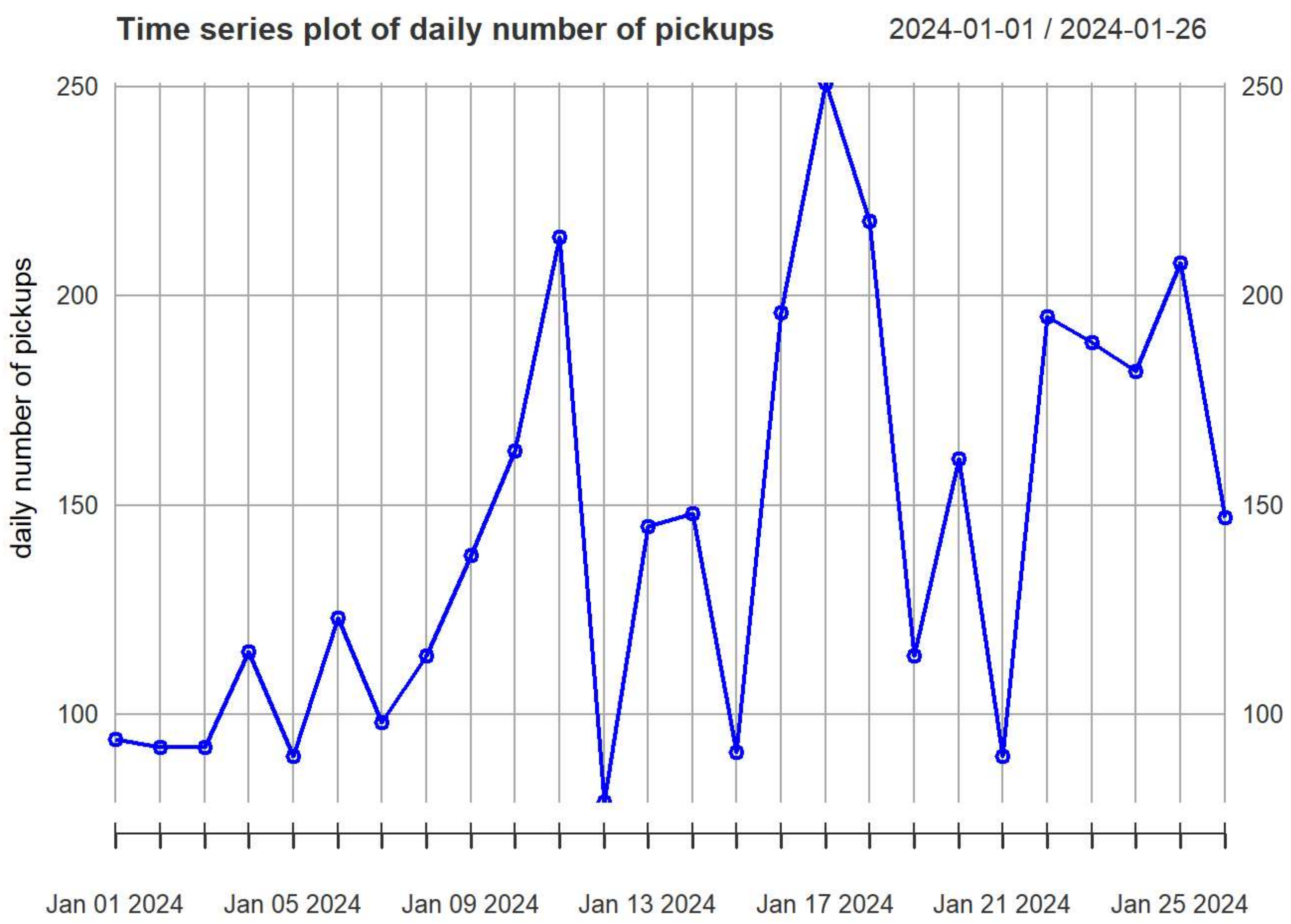
```
Total.ST.min <- c(649,468,639,628,701,478,604,701,674,460,558,564,635,538,591,380,511,378,831,623,765,346,
                  466,601,608,657)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
time_series_ST_xts <- xts(Total.ST.min, order.by = date_seq)
plot.xts(time_series_ST_xts, main = "Time series plot of daily total screen time", xlab = "Date", ylab =  "daily total screen time", type = "o", col = "blue", format = "%Y-%m-%d")
```



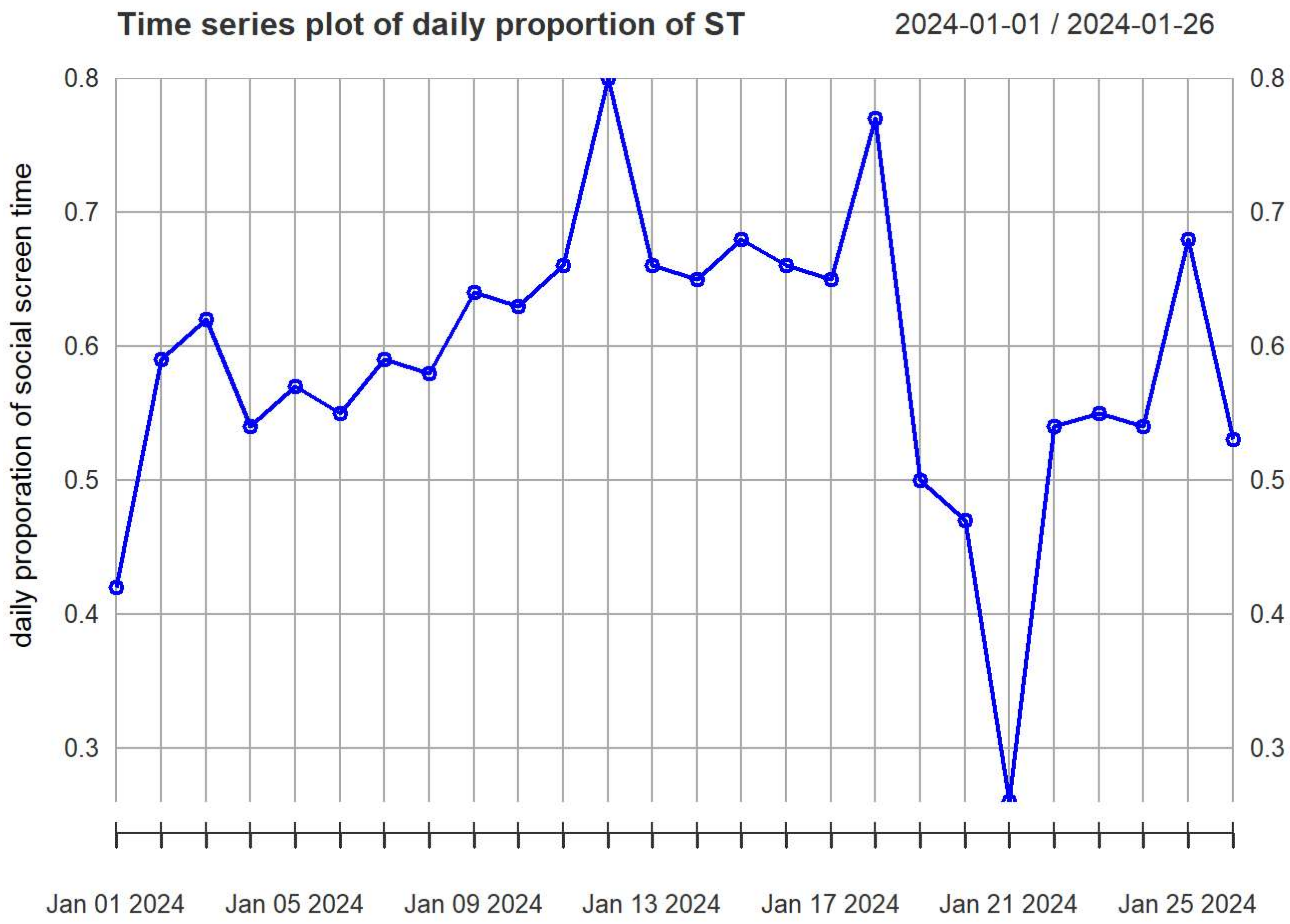

```
#Time series plot of daily total social screen time
Social.ST.min <- c(271,276,393,339,397,262,358,409,433,291,369,454,418,352,404,252,331,291,418,
292,200,188,254,323,414,350)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
time_series_social_ST_xts <- xts(Social.ST.min, order.by = date_seq)
plot.xts(time_series_social_ST_xts, main = "Time series plot of daily total social screen tim
e", xlab = "Date", ylab = "daily total social screen time", type = "o", col = "blue", format =
"%Y-%m-%d")
```



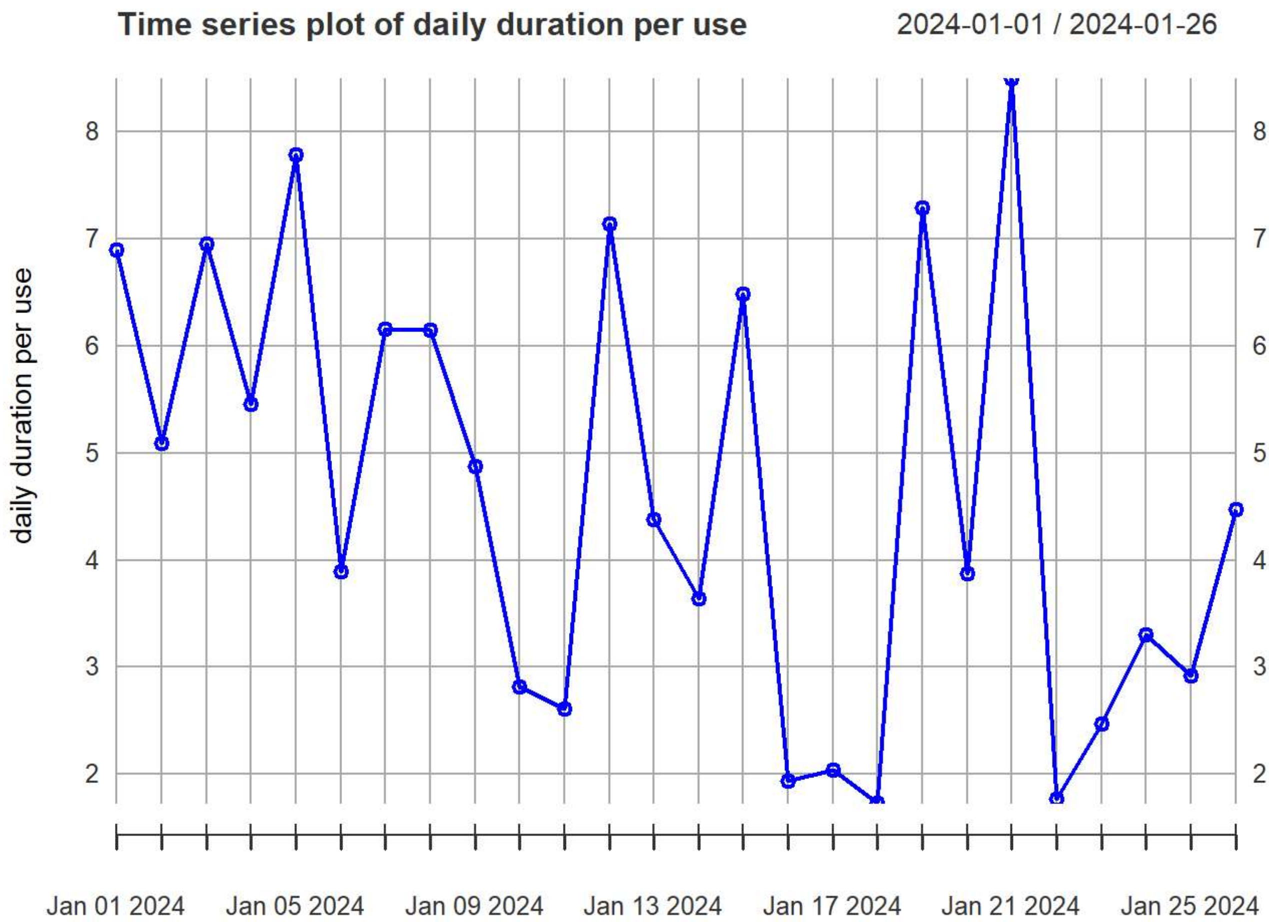

```
#Time series plot of daily number of pickups
daily_num_pickups <- c(94,92,92,115,90,123,98,114,138,163,214,79,145,148,91,196,251,218,114,16
1,90,195,189,182,208,147)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
daily_num_pickups_xts <- xts(daily_num_pickups, order.by = date_seq)
plot.xts(daily_num_pickups_xts, main = "Time series plot of daily number of pickups", xlab = "D
ate", ylab = "daily number of pickups", type = "o", col = "blue", format = "%Y-%m-%d")
```




```
#Time series plot of daily proportion of social screen time
daily_proportion_ST.min <- c(0.42,0.59,0.62,0.54,0.57,0.55,0.59,0.58,0.64,0.63,0.66,0.80,0.66,
0.65,0.68,
                             0.66, 0.65,0.77,0.50,0.47,0.26,0.54,0.55,0.54,0.68,0.53)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
daily_proportion_ST.min_xts <- xts(daily_proportion_ST.min, order.by = date_seq)
plot.xts(daily_proportion_ST.min_xts, main = "Time series plot of daily proportion of ST", xlab
= "Date", ylab = "daily proporation of social screen time", type = "o", col = "blue", format =
"%Y-%m-%d")
```




```
#Time series plot of daily duration per use
daily_duration_per_use <- c(6.90,5.09,6.95,5.46,7.79,3.89,6.16,6.15,4.88,2.82,2.61,7.14,4.38,3.64,6.49,1.94,2.04,1.73,7.29,3.87, 8.50,1.77,2.47,3.30,2.92,4.47)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
daily_duration_per_use_xts <- xts(daily_duration_per_use, order.by = date_seq)
plot.xts(daily_duration_per_use_xts, main = "Time series plot of daily duration per use ", xlab = "Date", ylab = "daily duration per use", type = "o", col = "blue", format = "%Y-%m-%d")
```



b.Make pairwise scatterplots of five variables.

```
#install.packages("GGally")
```

```
library("GGally")
```

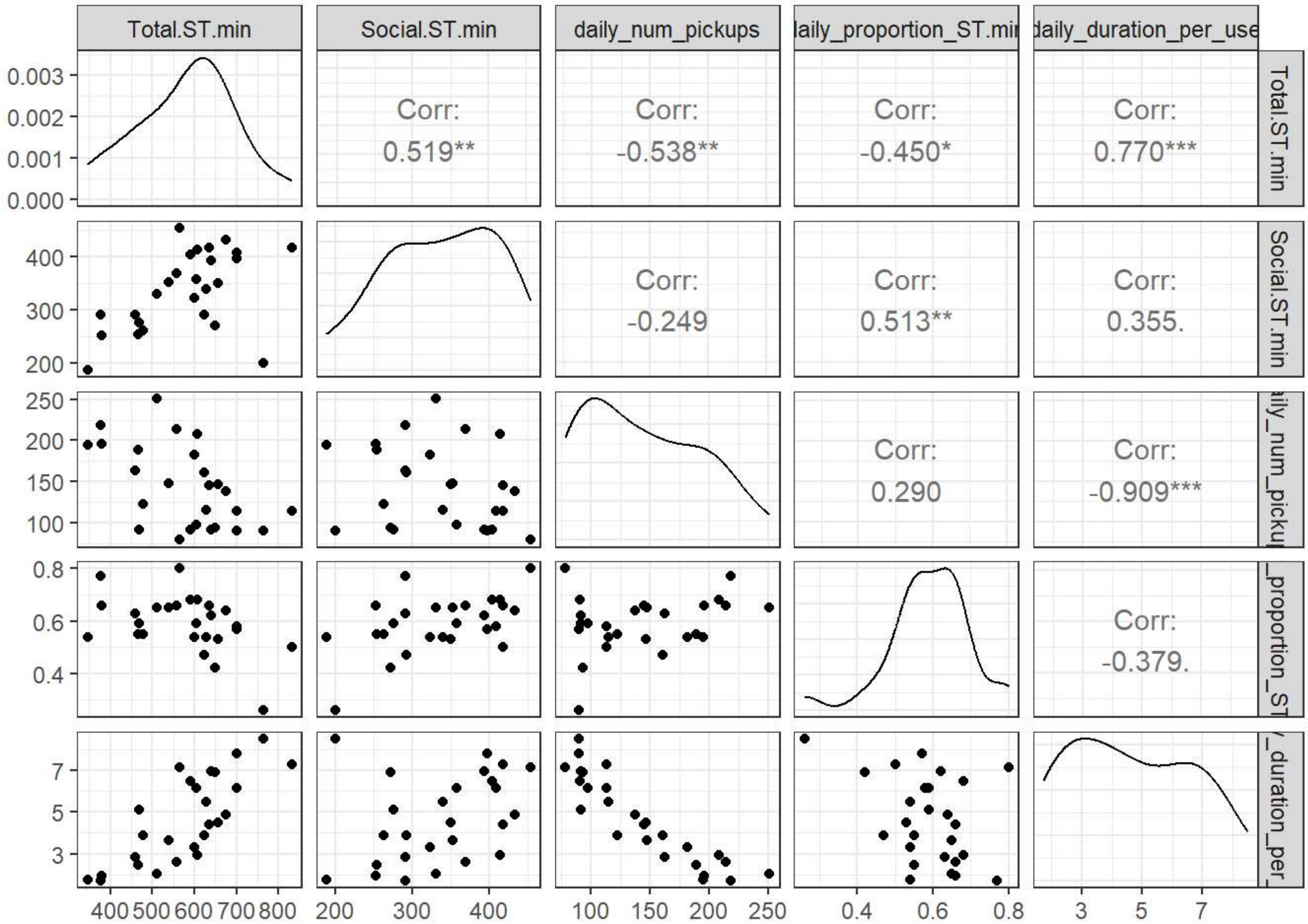
```
## Warning: 程辑包'GGally'是用R版本4.3.2 来建造的
```

```
## 载入需要的程辑包：ggplot2
```

```
## Warning: 程辑包'ggplot2'是用R版本4.3.2 来建造的
```

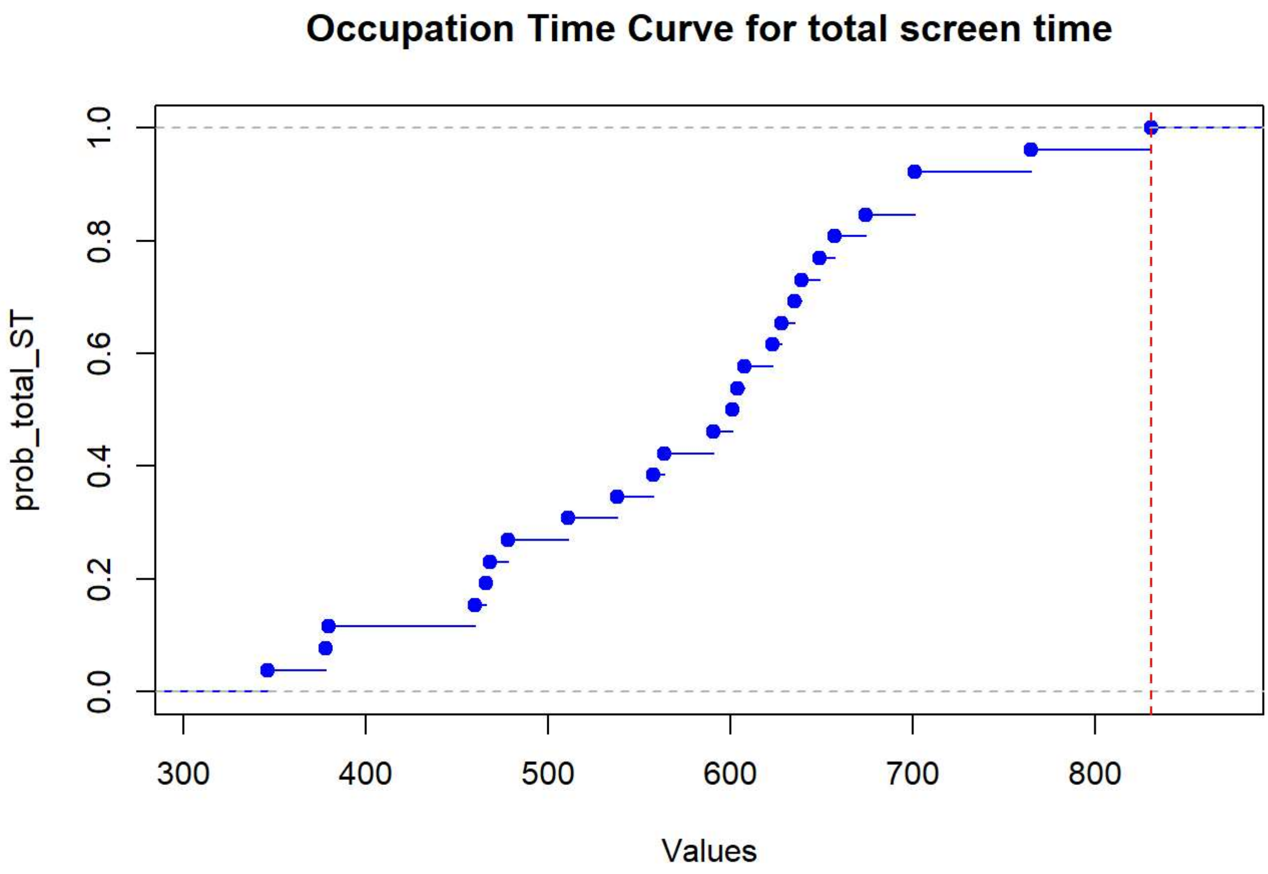
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
data <- data.frame(Total.ST.min= c(649,468,639,628,701,478,604,701,674,460,558,564,635,538,591,
380,511,378,831,623,765,346,466,601,608,657), Social.ST.min = c(271,276,393,339,397,262,358,40
9,433,291,369,454,418,352,404,252,331,291,418,292,200,188,
                   254,323,414,350), daily_num_pickups = c(94,92,92,115,90,123,98,114,138,163,2
14,79,145,148,91,196,251,218,114,161,90,195,189,
                                                           182,208,147),daily_proportion_ST.min
=  c(0.42,0.59,0.62,0.54,0.57,0.55,0.59,0.58,0.64,0.63,0.66,0.80,0.66,0.65,0.68,
                             0.66,  0.65,0.77,0.50,0.47,0.26,0.54,0.55,0.54,0.68,0.53),daily_dur
ation_per_use = c(6.90,5.09,6.95,5.46,7.79,3.89,6.16,6.15,4.88,2.82,2.61,7.14,4.38,3.64,6.49,1.
94,2.04,1.73,7.29,3.87, 8.50,1.77,2.47,3.30,2.92,4.47))
ggpairs(data, columns = c("Total.ST.min","Social.ST.min","daily_num_pickups","daily_proportion_
ST.min","daily_duration_per_use"), columnLabels =  c("Total.ST.min","Social.ST.min","daily_num_
pickups","daily_proportion_ST.min","daily_duration_per_use")) + theme_bw()
```

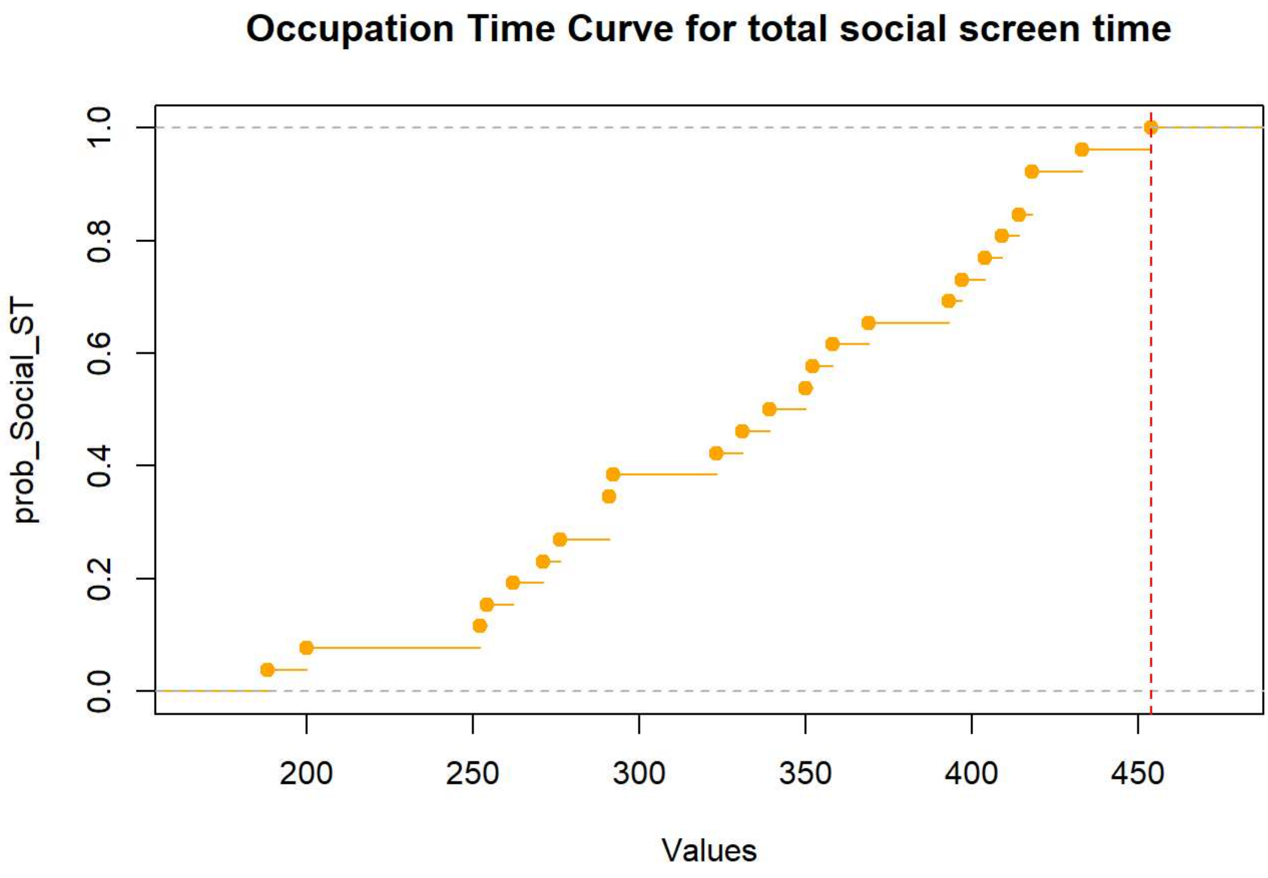


c.Make an occupation time curve for each of the five time series

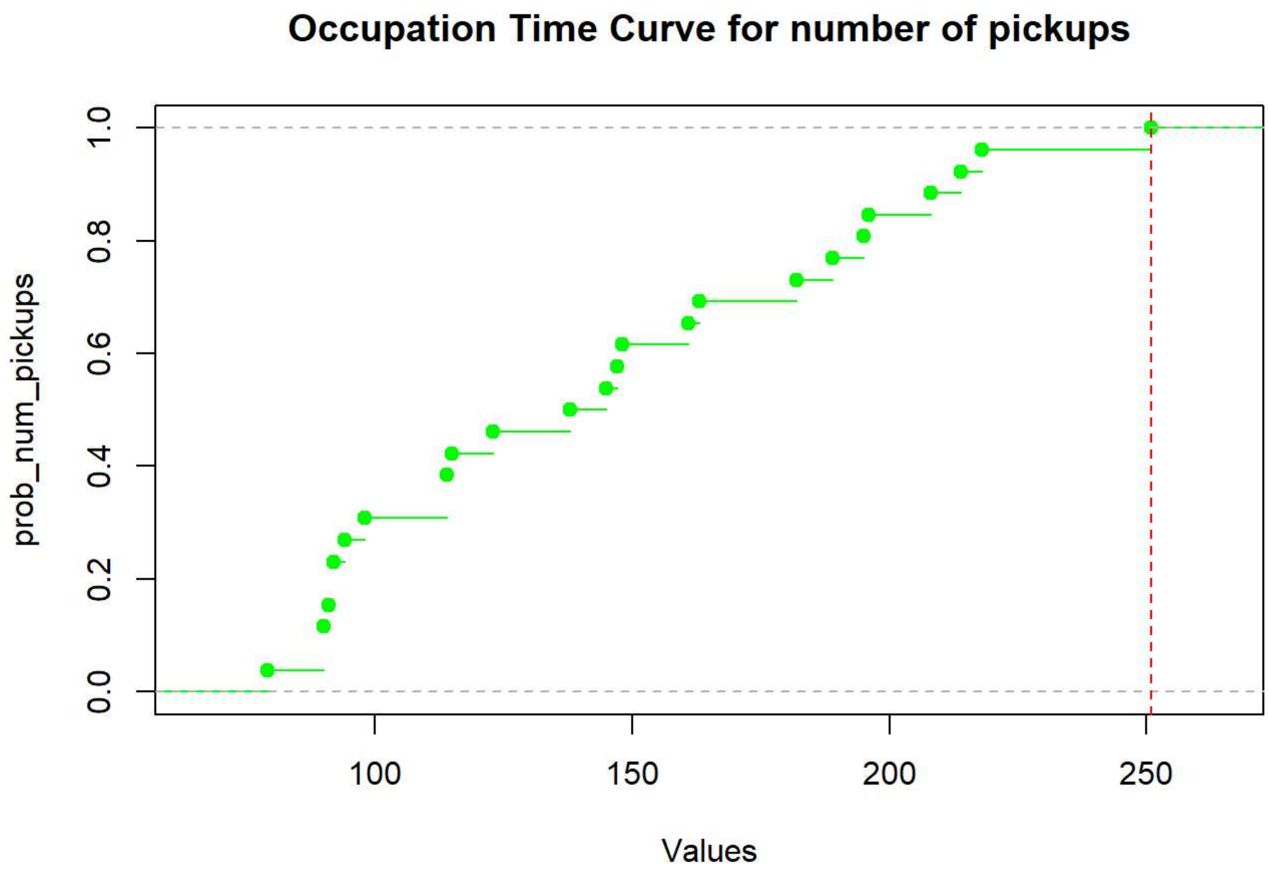
```
#Occupation time curve for daily total screen time
Total.ST.min <- c(649,468,639,628,701,478,604,701,674,460,558,564,635,538,591,380,511,378,831,6
23,765,346,
                  466,601,608,657)
ecdf_total_ST <- ecdf(Total.ST.min)
c_value_total_ST <- 831
prob_total_ST <- 1 - ecdf_total_ST(c_value_total_ST)
par(mfrow = c(1,1))
plot(ecdf_total_ST, col = "blue", main = "Occupation Time Curve for total screen time", xlab =
"Values", ylab = "prob_total_ST")
abline(v = c_value_total_ST, col = "red", lty = 2)
```



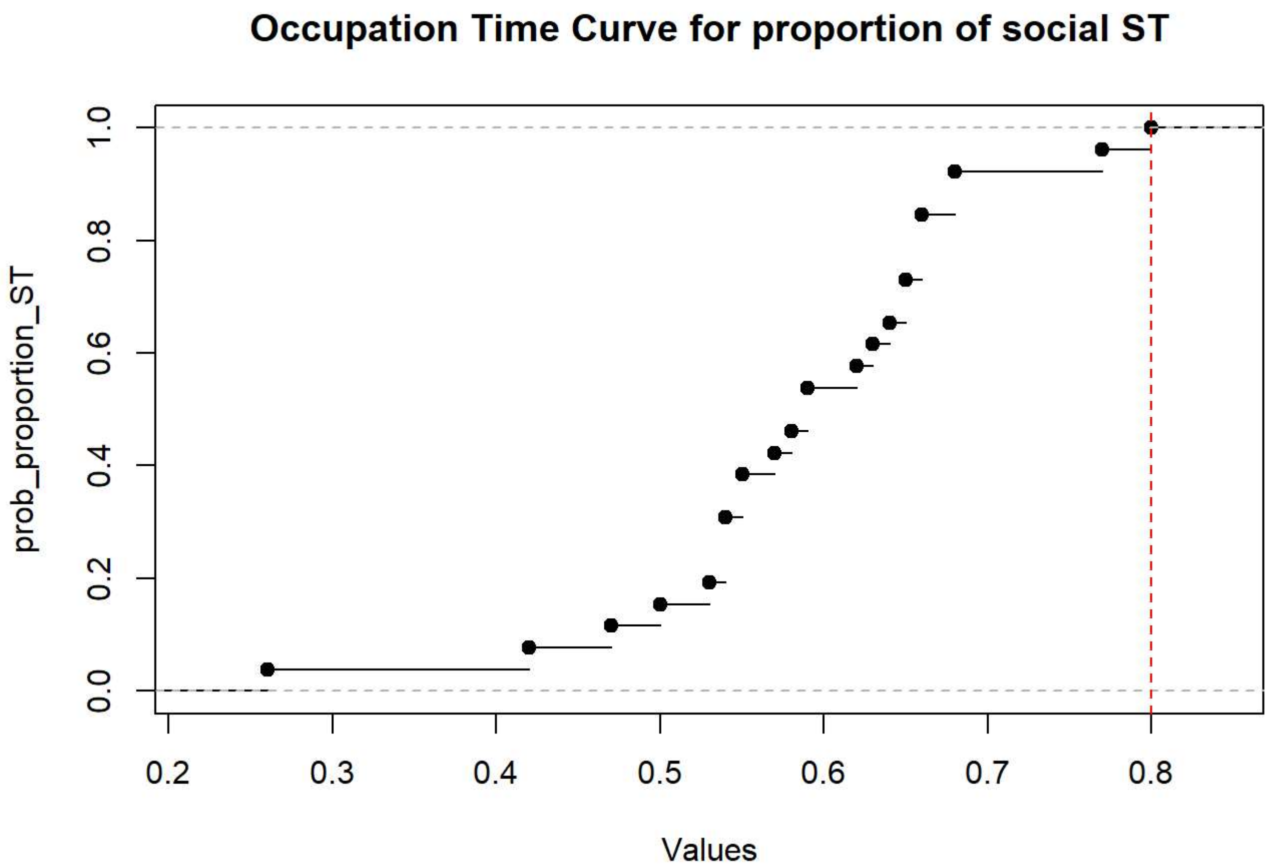

```
#Occupation time curve for daily total social screen time
Social.ST.min <- c(271,276,393,339,397,262,358,409,433,291,369,454,418,352,404,252,331,291,418,
292,200,188,254,323,414,350)
ecdf_Social_ST <- ecdf(Social.ST.min)
c_value_Social_ST <- 454
prob_Social_ST <- 1 - ecdf_Social_ST(c_value_Social_ST)
par(mfrow = c(1,1))
plot(ecdf_Social_ST, col = "orange", main = "Occupation Time Curve for total social screen tim
e", xlab = "Values", ylab = "prob_Social_ST")
abline(v = c_value_Social_ST, col = "red", lty = 2)
```



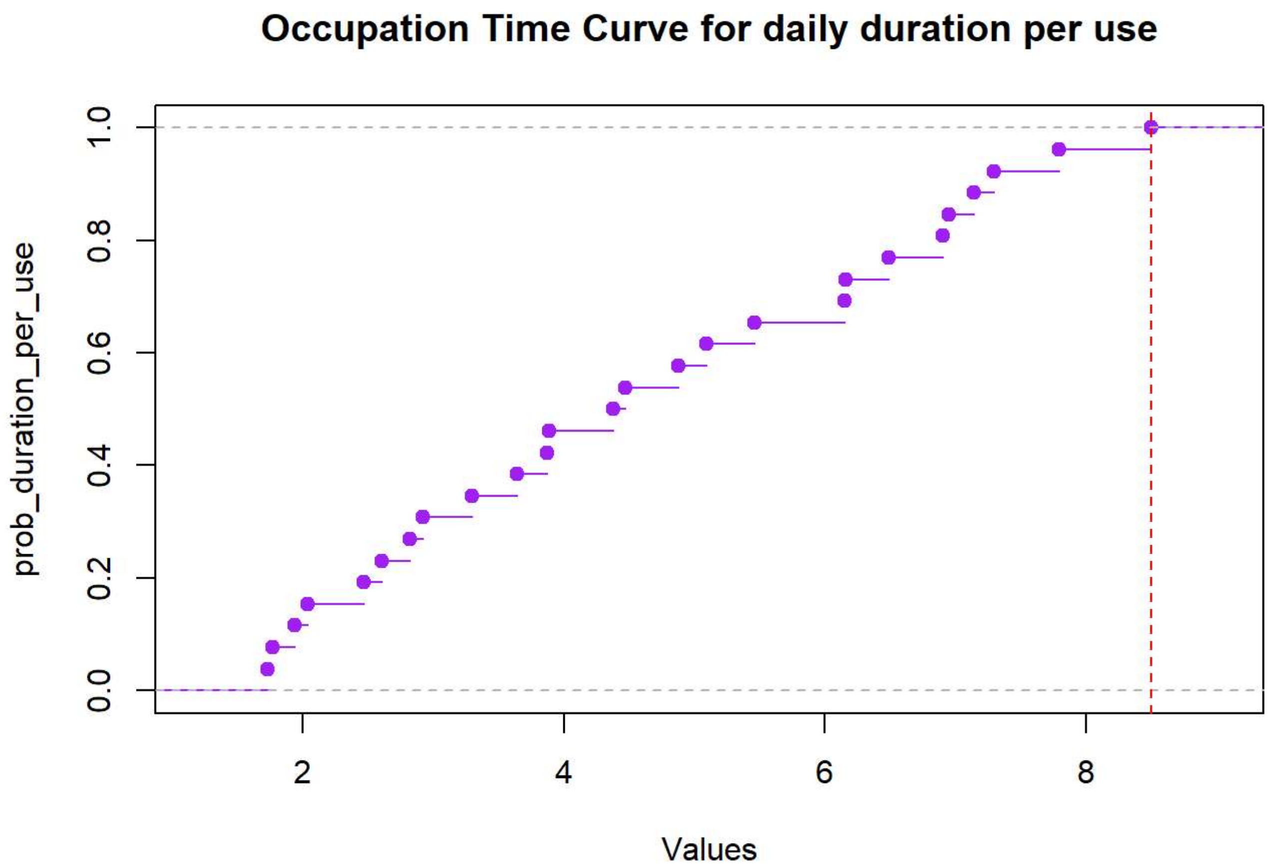

```
#Occupation time curve for daily number of pickups
daily_num_pickups <- c(94,92,92,115,90,123,98,114,138,163,214,79,145,148,91,196,251,218,114,16
1,90,195,189,182,208,147)
ecdf_daily_num_pickups <- ecdf(daily_num_pickups)
c_value_num_pickups <- 251
prob_num_pickups <- 1 - ecdf_daily_num_pickups(c_value_num_pickups)
par(mfrow = c(1,1))
plot(ecdf_daily_num_pickups, col = "green", main = "Occupation Time Curve for number of pickup
s", xlab = "Values", ylab = "prob_num_pickups")
abline(v = c_value_num_pickups, col = "red", lty = 2)
```




```
#Occupation time curve for daily proportion Of social screen time
daily_proportion_ST.min <- c(0.42,0.59,0.62,0.54,0.57,0.55,0.59,0.58,0.64,0.63,0.66,0.80,0.66,0.65,0.68,
                             0.66, 0.65,0.77,0.50,0.47,0.26,0.54,0.55,0.54,0.68,0.53)
ecdf_daily_proportion_ST <- ecdf(daily_proportion_ST.min)
c_value_proportion_ST <- 0.80
prob_proportion_ST <- 1 - ecdf_daily_proportion_ST(c_value_proportion_ST)
par(mfrow = c(1,1))
plot(ecdf_daily_proportion_ST, col = "black", main = "Occupation Time Curve for proportion of social ST", xlab = "Values", ylab = "prob_proportion_ST")
abline(v = c_value_proportion_ST, col = "red", lty = 2)
```

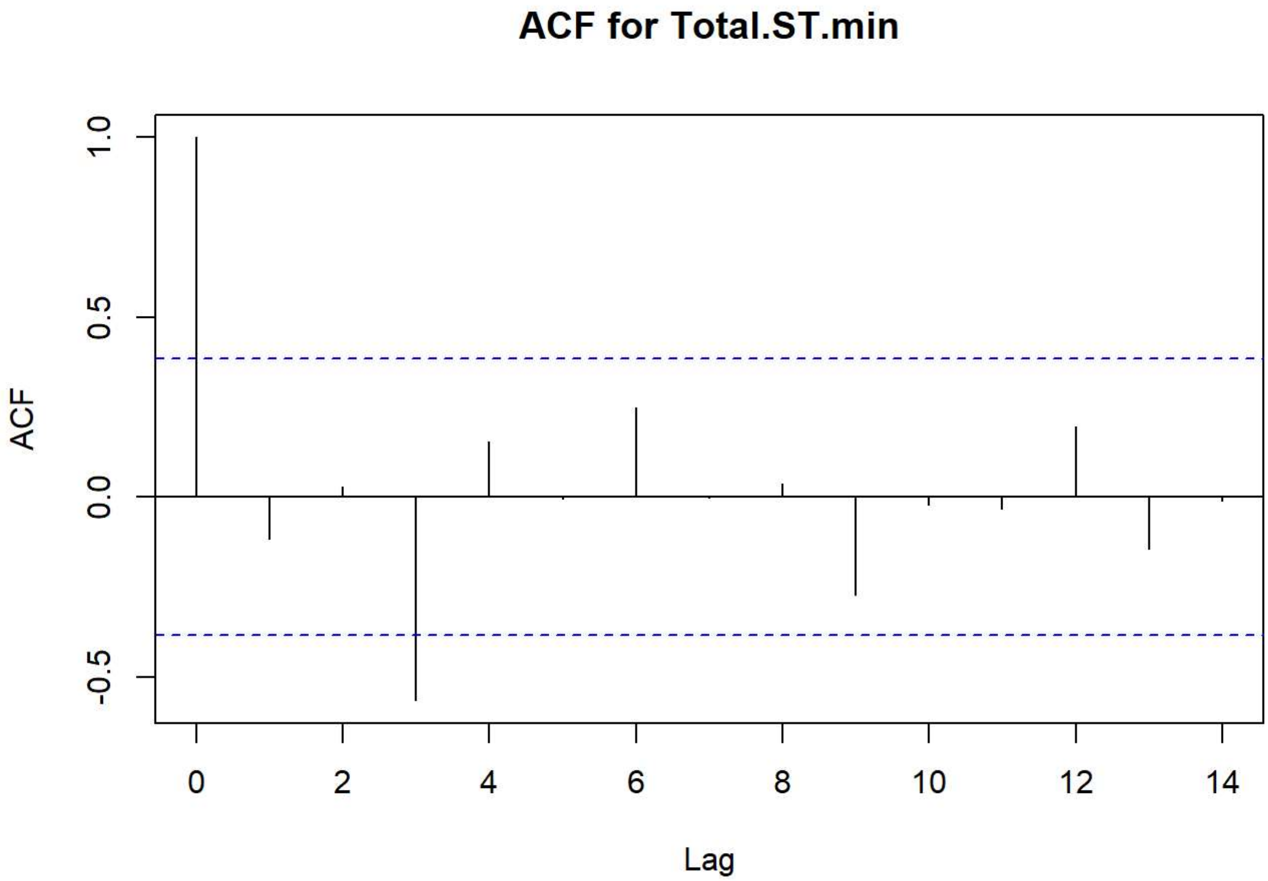



```
#Occupation time curve for daily duration per use
daily_duration_per_use <- c(6.90,5.09,6.95,5.46,7.79,3.89,6.16,6.15,4.88,2.82,2.61,7.14,4.38,3.
64,6.49,1.94,2.04,1.73,7.29,3.87, 8.50,1.77,2.47,3.30,2.92,4.47)
ecdf_daily_duration_per_use <- ecdf(daily_duration_per_use)
c_value_duration_per_use <- 8.50
prob_duration_per_use <- 1 - ecdf_daily_duration_per_use(c_value_duration_per_use)
par(mfrow = c(1,1))
plot(ecdf_daily_duration_per_use, col = ″purple″, main = ″Occupation Time Curve for daily durat
ion per use″, xlab = ″Values″, ylab = ″prob_duration_per_use″)
abline(v = c_value_duration_per_use, col = ″red″, lty = 2)
```

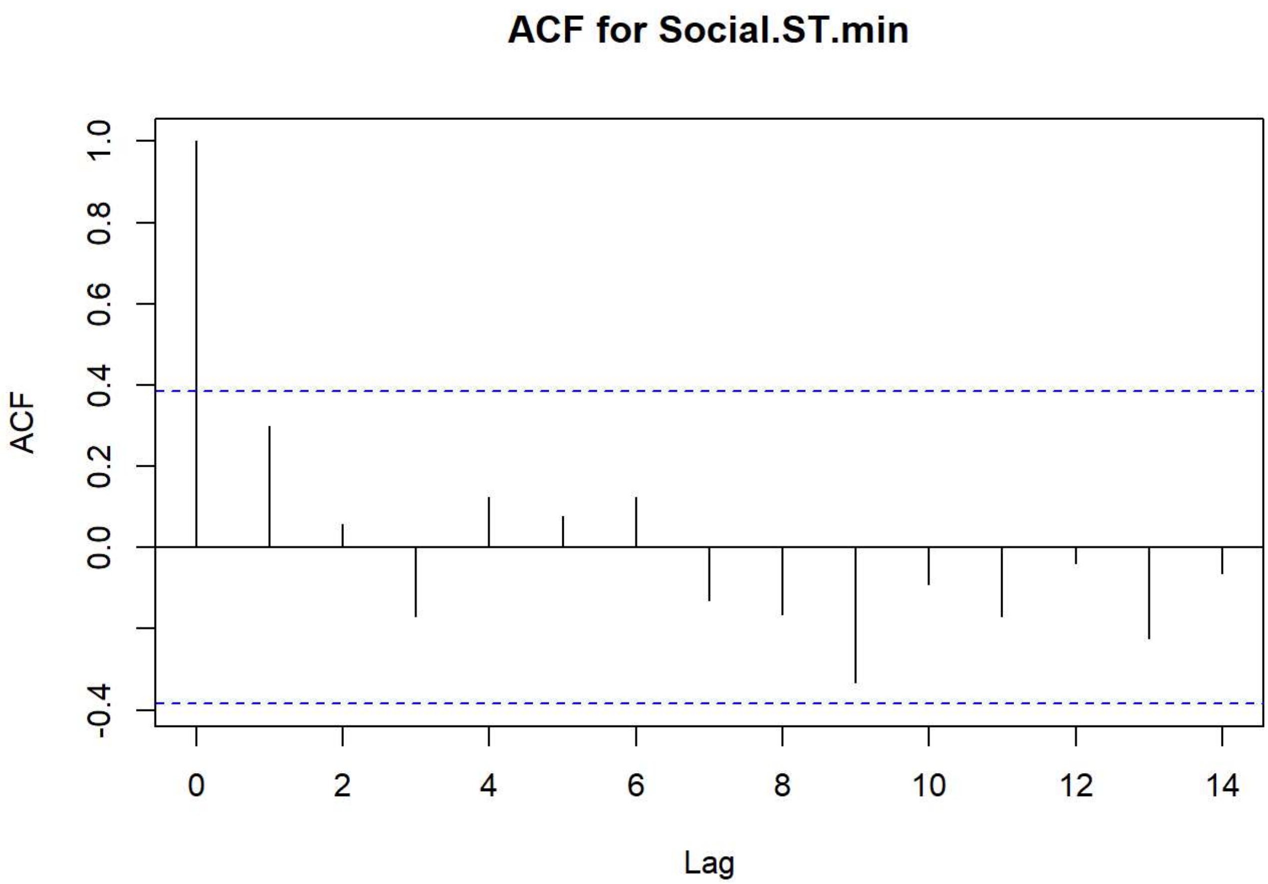


(d) Use the R function acf to display the serial dependence for each of the five time series

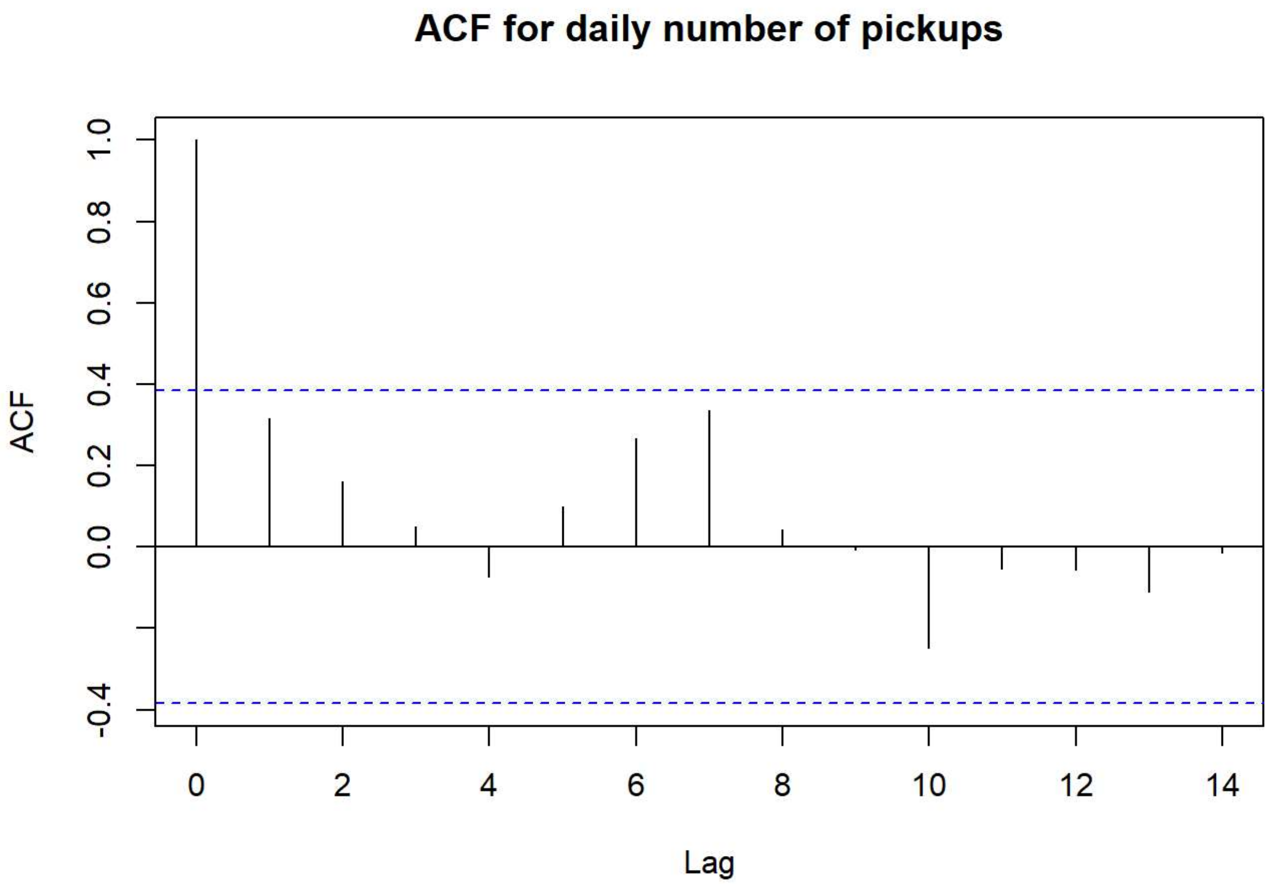
```
Total.ST.min <- c(649,468,639,628,701,478,604,701,674,460,558,564,635,538,591,380,511,378,831,6
23,765,346,
                  466,601,608,657)
Social.ST.min <- c(271,276,393,339,397,262,358,409,433,291,369,454,418,352,404,252,331,291,418,
292,200,188,
                   254,323,414,350)
daily_num_pickups <- c(94,92,92,115,90,123,98,114,138,163,214,79,145,148,91,196,251,218,114,16
1,90,195,189,
                       182,208,147)
daily_proportion_ST.min <- c(0.42,0.59,0.62,0.54,0.57,0.55,0.59,0.58,0.64,0.63,0.66,0.80,0.66,
0.65,0.68,
                             0.66,  0.65,0.77,0.50,0.47,0.26,0.54,0.55,0.54,0.68,0.53)
daily_duration_per_use <- c(6.90,5.09,6.95,5.46,7.79,3.89,6.16,6.15,4.88,2.82,2.61,7.14,4.38,3.
64,6.49,1.94,2.04,1.73,7.29,3.87,  8.50,1.77,2.47,3.30,2.92,4.47)
acf(Total.ST.min, main = "ACF for Total.ST.min")
```



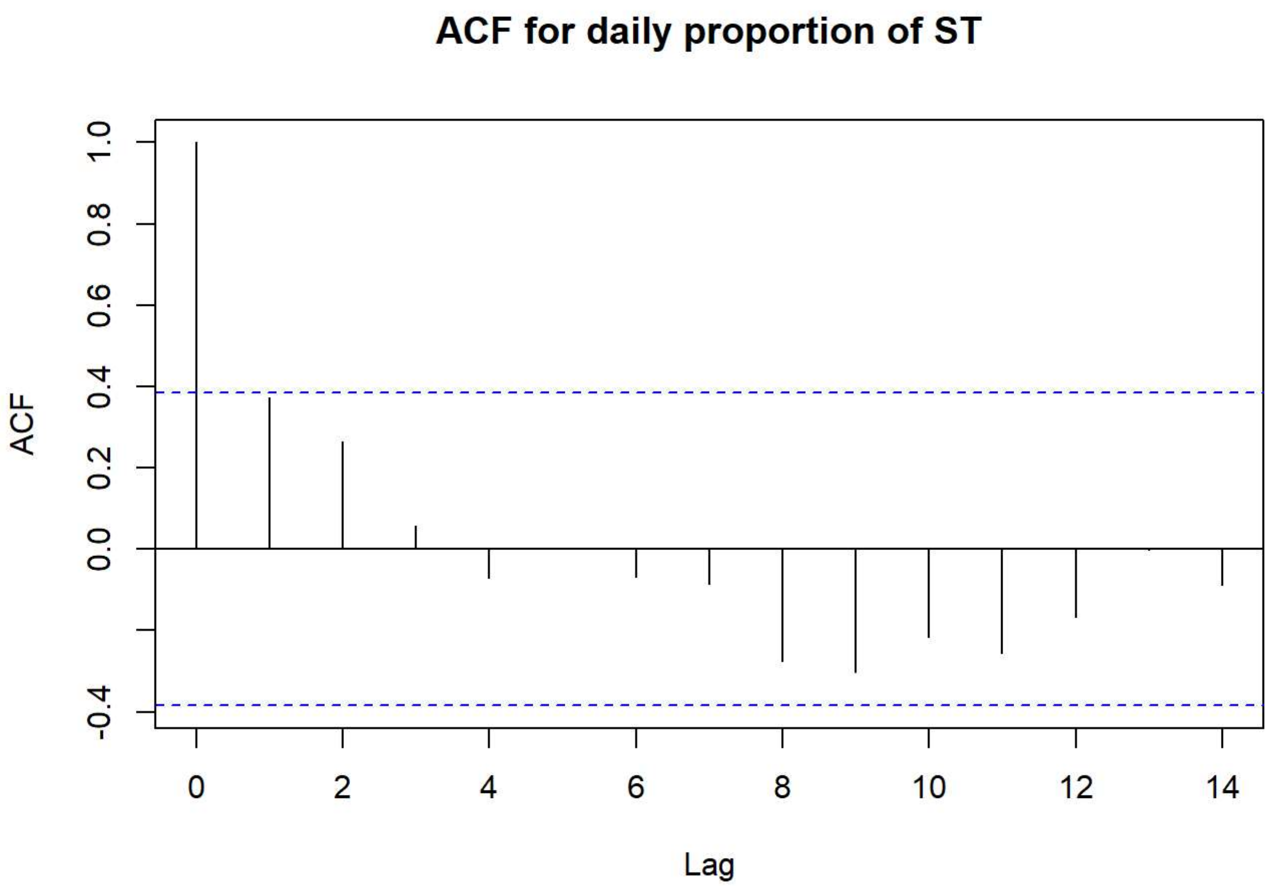

```
acf(Social.ST.min, main = "ACF for Social.ST.min")
```



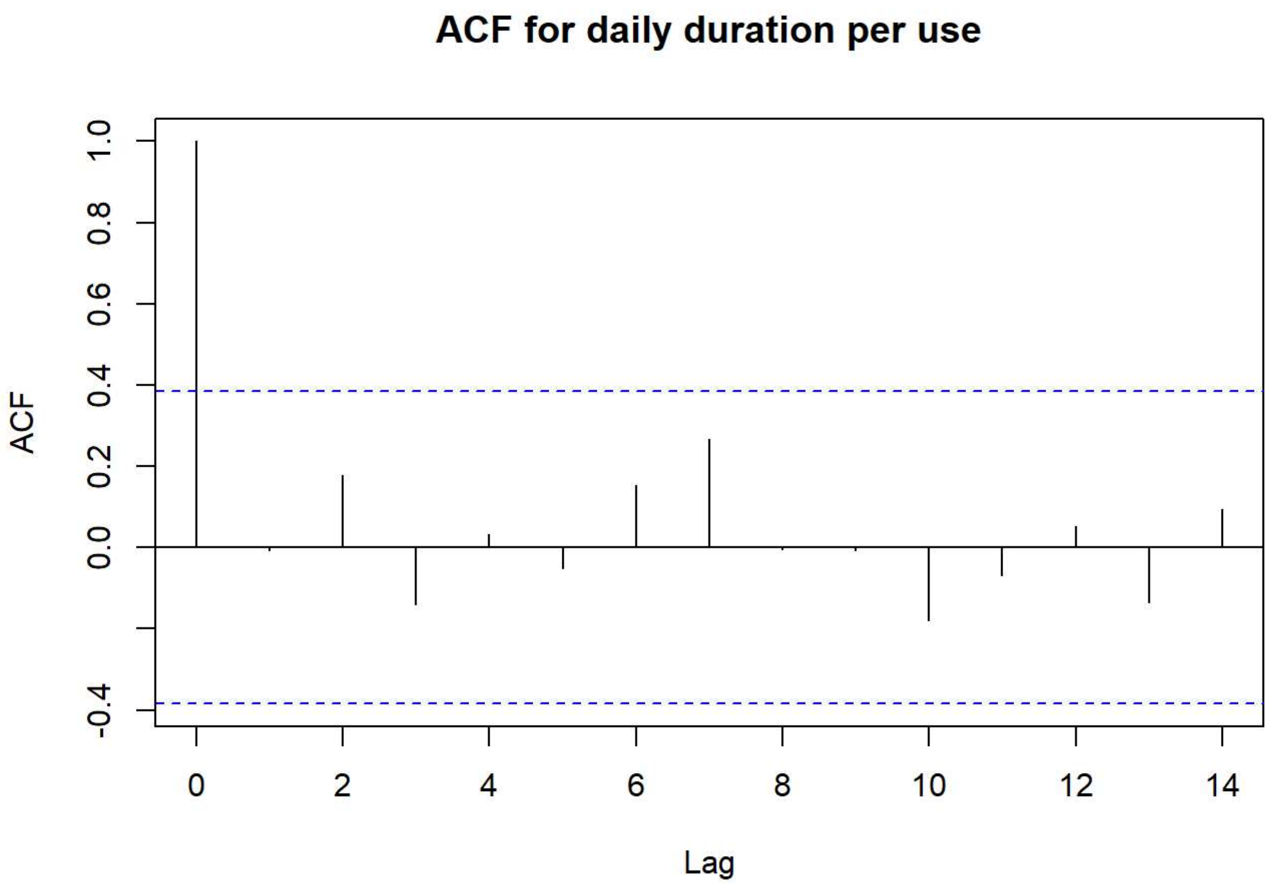
```
acf(daily_num_pickups, main = "ACF for daily number of pickups")
```




```
acf(daily_proportion_ST.min, main = "ACF for daily proportion of ST")
```




```
acf(daily_duration_per_use, main = "ACF for daily duration per use")
```



```
acf(Total.ST.min, main = "ACF for Total.ST.min", plot = FALSE)
```

```
##
## Autocorrelations of series 'Total.ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000 -0.117  0.028 -0.566  0.154 -0.006  0.249 -0.002  0.037 -0.272 -0.021
##     11     12     13     14
## -0.032  0.196 -0.145 -0.010
```

```
acf(Social.ST.min, main = "ACF for Social.ST.min", plot = FALSE)
```

```
##
## Autocorrelations of series 'Social.ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.299  0.057 -0.170  0.124  0.078  0.123 -0.130 -0.164 -0.331 -0.091
##     11     12     13     14
## -0.169 -0.038 -0.224 -0.063
```

```
acf(daily_num_pickups, main = "ACF for daily number of pickups", plot = FALSE)
```

```
##
## Autocorrelations of series 'daily_num_pickups', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.316  0.161  0.049 -0.072  0.100  0.266  0.337  0.043 -0.007 -0.249
##     11     12     13     14
## -0.055 -0.056 -0.110 -0.013
```

```
acf(daily_proportion_ST.min, main = "ACF for daily proportion of ST", plot = FALSE)
```

```
##
## Autocorrelations of series 'daily_proportion_ST.min', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.373  0.265  0.057 -0.072  0.002 -0.069 -0.087 -0.275 -0.302 -0.216
##     11     12     13     14
## -0.257 -0.166 -0.002 -0.087
```

```
acf(daily_duration_per_use, main = "ACF for daily duration per use", plot = FALSE)
```

```
##
## Autocorrelations of series 'daily_duration_per_use', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000 -0.008  0.177 -0.140  0.034 -0.052  0.153  0.266 -0.004 -0.008 -0.179
##     11     12     13     14
## -0.068  0.053 -0.136  0.093
```

problem 3 b. Make a scatterplot of the first pickup data on a 24-hour clock circle.

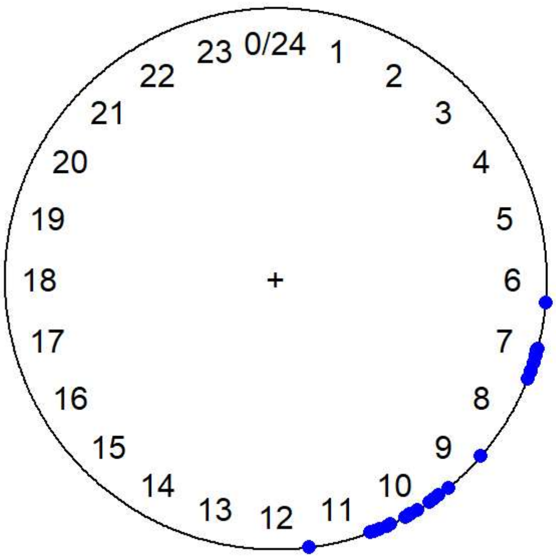
```
#install.packages("circular")
library(circular)
```

```
## Warning: 程辑包'circular'是用R版本4.3.2 来建造的
```

```
##
## 载入程辑包：'circular'
```

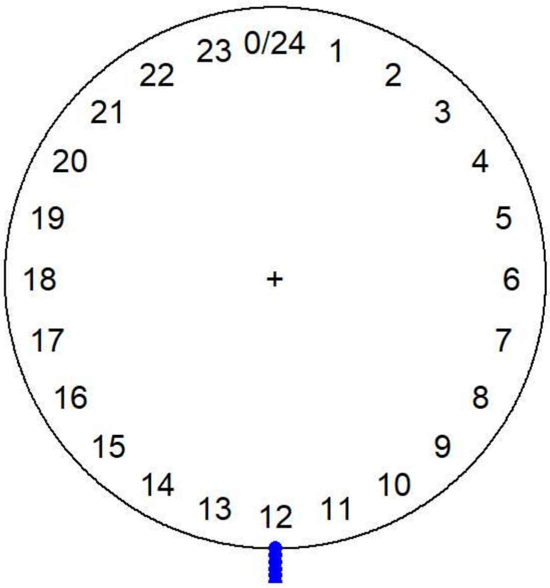
```
## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
pickup_1st_degree <- c(155, 157.5, 173, 144.5, 159.5, 131, 145.5, 155.75, 148.5, 95, 108, 140.
5, 143, 158.75, 106.25, 111.75, 105, 110, 150.5, 151, 150.25, 109.75, 105.25, 106.5, 105.25, 15
1.5)
first_pickup_circular = circular(pickup_1st_degree, units = "degrees", template = "clock24")
plot(first_pickup_circular, col = "blue")
```

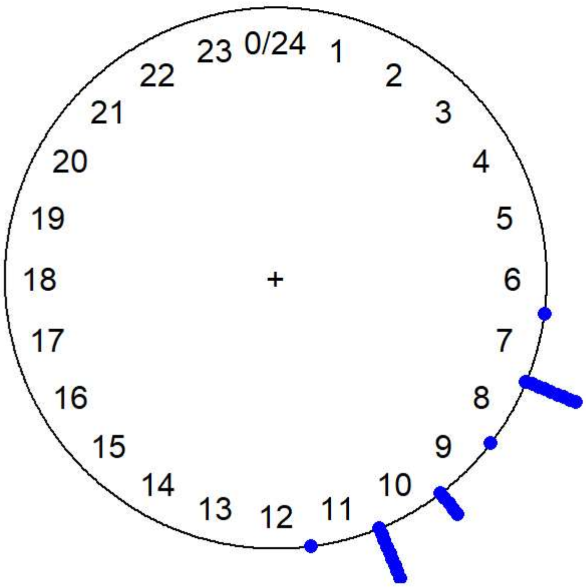


c. Make a histogram plot on the circle in that you may choose a suitable bin size to create stacking.For example, you may set a bin size at 2.5 degree, which corresponds an interval of 10 minutes.Adjust the bin size to create different forms of histogram, and explain the reason that you choose a particular value to report your final histogram plot.

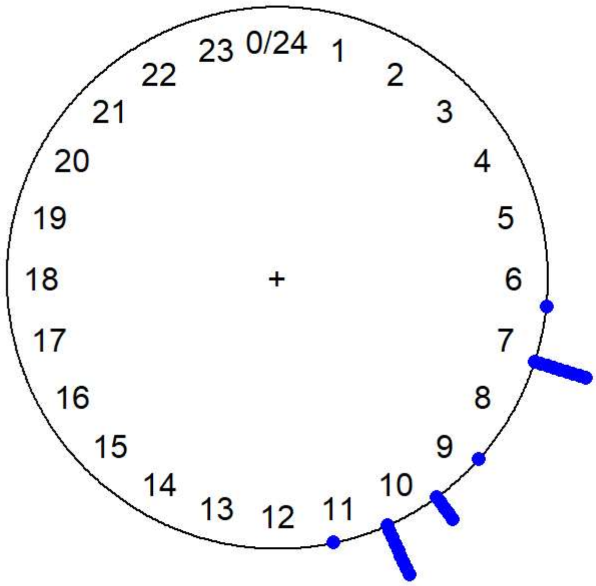
```
plot(first_pickup_circular, stack = TRUE, bins = 2.5, col = "blue")
```



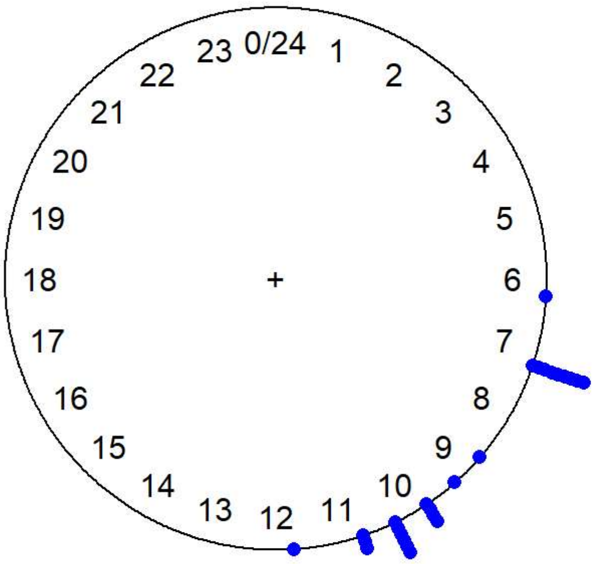

```
plot(first_pickup_circular, stack = TRUE, bins = 24, col = "blue")
```



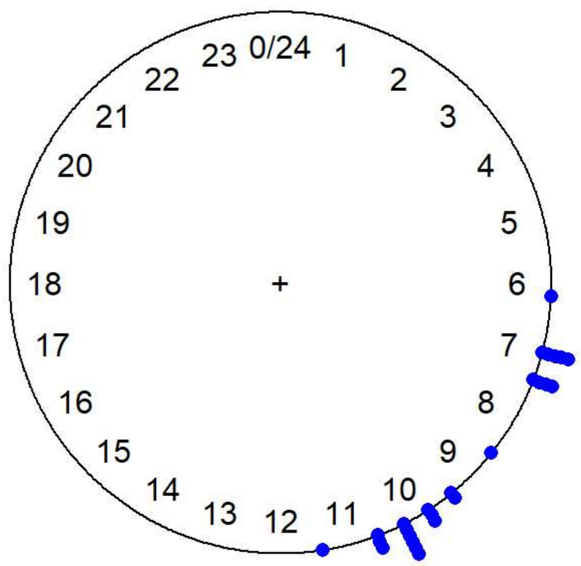
```
plot(first_pickup_circular, stack = TRUE, bins = 30, col = "blue")
```




```
plot(first_pickup_circular, stack = TRUE, bins = 48, col = "blue")
```




```
plot(first_pickup_circular, stack = TRUE, bins = 60, col = "blue")
```



problem 4 b. Use the R function glm to estimate the rate parameter λ in which ln(St) is included in the model as an offset.

```
Yt_pickups <- c(94,92,92,115,90,123,98,114,138,163,214,79,145,148,91,196,251,218,114,161,90,19
5,189,
                182,208,147)
St <- c(10.82, 10.8, 10.65, 10.47, 11.68, 7.97, 10.07, 11.68, 11.23, 7.67, 9.3, 9.4, 10.58, 8.9
7, 9.85, 6.33, 8.52, 6.3, 13.85, 10.38, 12.75, 5.77, 7.77, 10.02, 10.13, 10.95)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
df <- data.frame(Date = date_seq, Yt = Yt_pickups)
model <- glm(Yt ~ offset(log(St)), family = poisson, data = df)
summary(model)
```

```
##
## Call:
## glm(formula = Yt ~ offset(log(St)), family = poisson, data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.69173    0.01634   164.8   <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: 903.44  on 25  degrees of freedom
## Residual deviance: 903.44  on 25  degrees of freedom
## AIC: 1081
##
## Number of Fisher Scoring iterations: 4
```

c.Define two dummy variables: Xt = 1 for day t being a weekday and 0 for day t being a weekend day; and Zt = 1 for day t being January, 10 (the first day of the winter semester) or after, and 0 for day t before January, 10 (the winter holiday day).Repeat part (b) for a model ln(λ(t)) = β0+β1Xt+β2Zt, under which the rate parameter λ differs between weekdays and weekends as well as between the winter semester and the winter holiday. This model is called log-linear model. Cleary, this rate parameter depends on day t. Use the R function glm to estimate the regression coefficients and answer the following questions.(c.1) Is there data evidence for significantly different behavior of daily pickups between weekdays and weekends? Justify your answer using the significance level α = 0.05. (c.2) Is there data evidence for a significant change on the behavior of daily pickups after the winter semester began? Justify your answer using the significance level α = 0.05

```
Xt_values <- c(1,1,1,1,1,0,0,1,1,1,1,1,0,0,1,1,1,1,1,0,0,1,1,1,1,1)
Zt_values <- c(0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
lambda_t_values <- c(8.69,8.52,8.64,10.98,7.71,15.43,9.73,9.76,12.29,21.25,23.01,8.40,13.71,16.50,9.24,
                     30.96,29.46,34.60,8.23,15.51,7.06,33.80,24.32,18.16,20.53,13.42)
log_lambda = log(lambda_t_values)
start_date <- as.Date("2024-01-01")
end_date <- as.Date("2024-01-26")
date_seq <- seq(from = start_date, to = end_date, by = "day")
df <- data.frame(Date = date_seq, Xt = Xt_values, Zt = Zt_values)
model_log_linear <- glm(log_lambda ~ Xt + Zt, family = poisson, data = df)
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.162173
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.142416
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.156403
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.396075
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.042518
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.736314
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.275214
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.278292
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.508786
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.056357
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.135929
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.128232
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.618125
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.803360
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.223542
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.432696
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.383033
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.543854
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.107786
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.741485
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 1.954445
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.520461
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.191299
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.899221
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 3.021887
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 2.596746
```

```
summary(model_log_linear)
```

```
##
## Call:
## glm(formula = log_lambda ~ Xt + Zt, family = poisson, data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.77757    0.31798   2.445   0.0145 *
## Xt           0.07049    0.29096   0.242   0.8086
## Zt           0.21351    0.26269   0.813   0.4163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2.3605  on 25  degrees of freedom
## Residual deviance: 1.6280  on 23  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 4
```

problem 5 a. Use the R function mle.vonmises from the R package circular to obtain the estimates of the two model parameters μ and λ from your data of first pickups.

```
library(circular)
pickup_1st_degree <- c(155, 157.5, 173, 144.5, 159.5, 131, 145.5, 155.75, 148.5, 95, 108, 140.
5, 143, 158.75, 106.25, 111.75, 105, 110, 150.5, 151, 150.25, 109.75, 105.25, 106.5, 105.25, 15
1.5)
mle_vonmises_result <- mle.vonmises(circular(pickup_1st_degree, units = "degrees"))
mu = mle_vonmises_result$mu
lambda = 1/mle_vonmises_result$kappa
print(mu)
```

```
## Circular Data:
## Type = angles
## Units = degrees
## Template = none
## Modulo = asis
## Zero = 0
## Rotation = counter
## [1] 133.9388
```

```
print(lambda)
```

```
## [1] 0.1520858
```

b. Based on the estimated parameters from part (a), use the R function pvonmises from the R package circular to calculate the probability that your first pickup is 8:30AM or later.

```
library(circular)
mle_vonmises_result <- mle.vonmises(circular(pickup_1st_degree, units = "degrees"))
mu <- mle_vonmises_result$mu
kappa <- 1/0.1520858
value <- circular(127.5, units = "degrees")
prob_8_30_AM_later <- 1 - pvonmises(value, mu = mu, kappa = kappa)
print(prob_8_30_AM_later)
```

```
## [1] 0.6110587
```