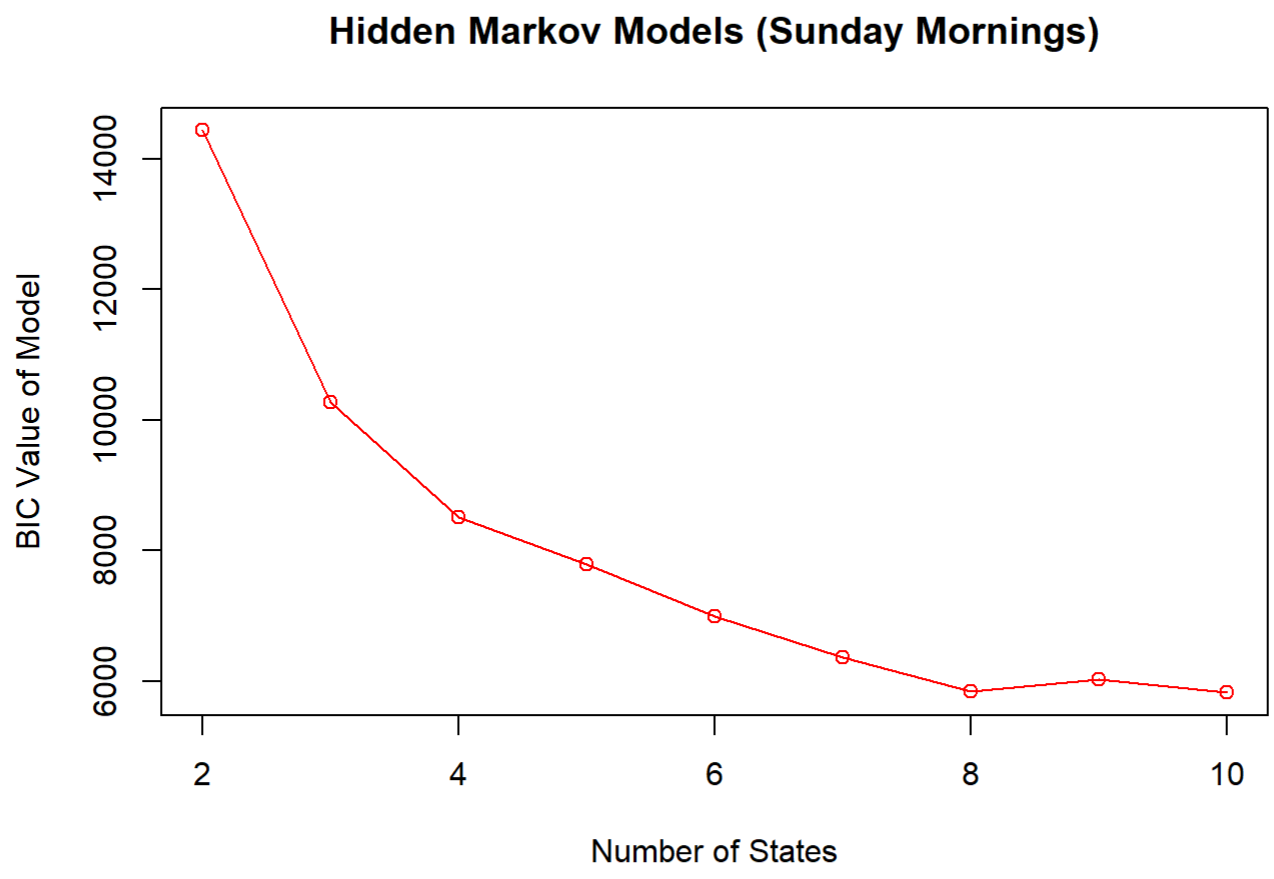
CMPT318-A2

Group 10

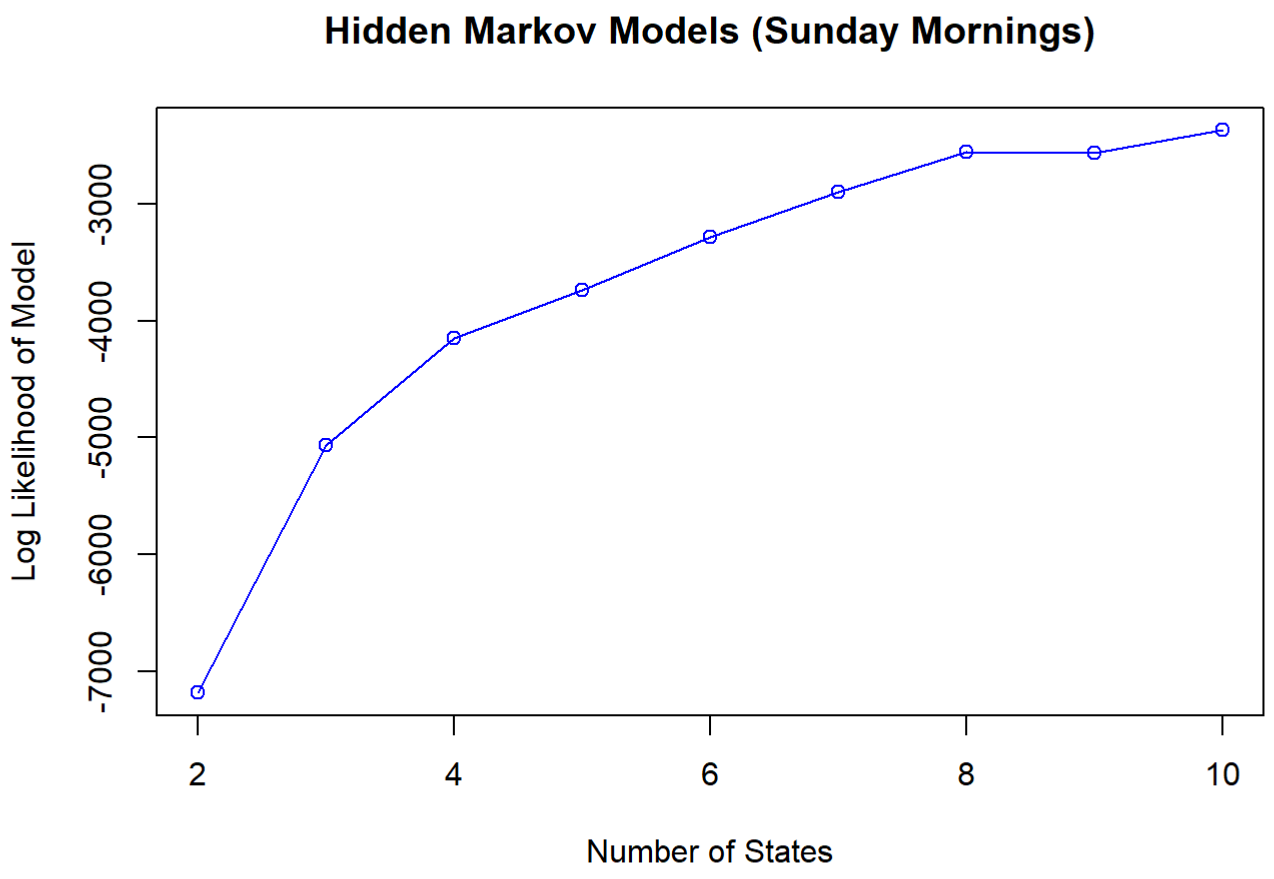
October 26, 2018

Question 1

```
# Question 1
# Set the variables
GAP <- sundayMornings$Global_active_power
GRP <- sundayMornings$Global_reactive_power
Vol <- sundayMornings$Voltage
GI  <- sundayMornings$Global_intensity

set.seed(1)
BICvector <- vector()
logLikVector <- vector()
states <- c(2:10)
vectorntimes = rep(181, nrow(sundayMornings)/181)
for (state in states) {
  print(paste("Number of States: ", state))
  mod <- depmix(response = Global_active_power ~ 1, data = sundayMornings, nstates = state, ntim
es = vectorntimes)
  fm <- fit(mod)
  summary(fm)
  BICvector <- c(BICvector, BIC(fm))
  logLikVector <- c(logLikVector, logLik(fm))
}
```



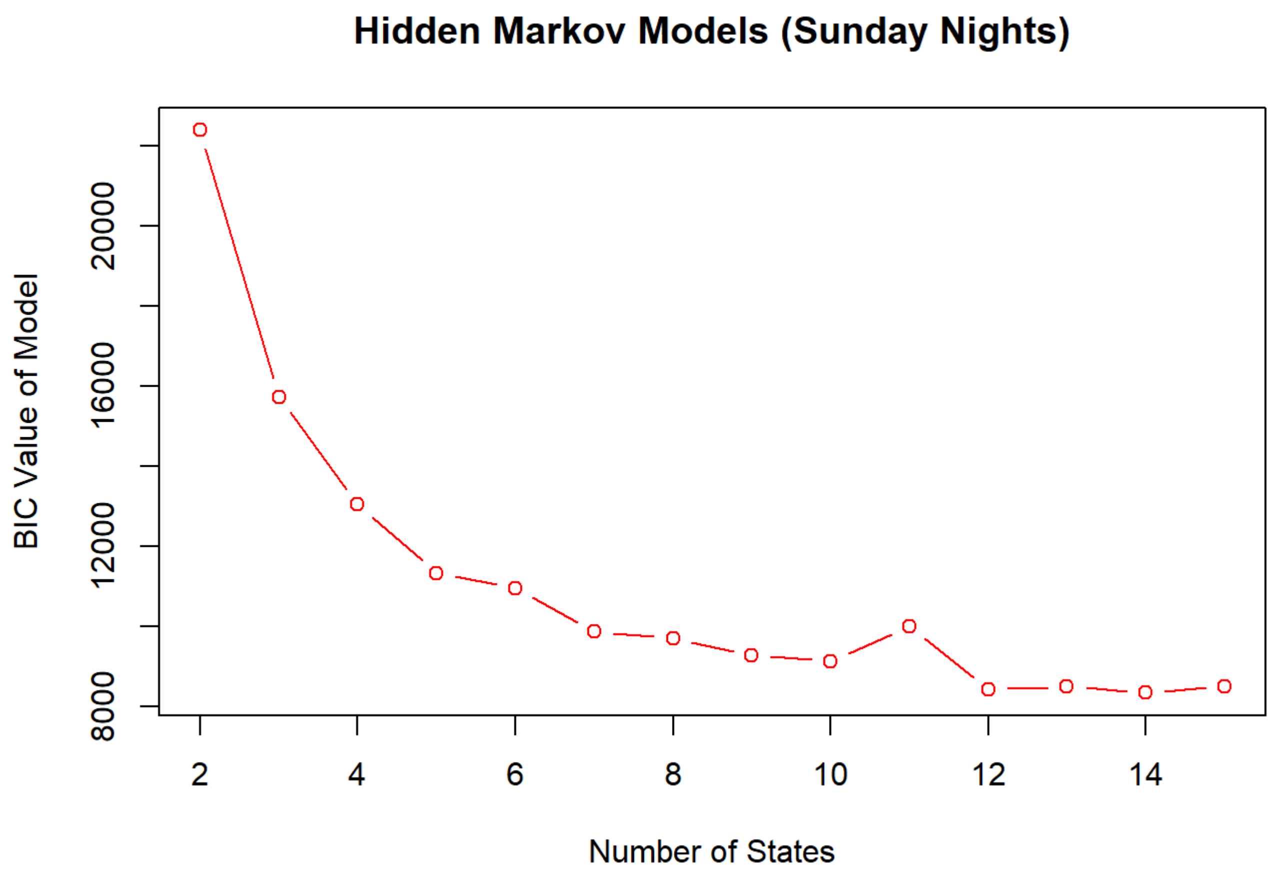
```
plot(states, logLikVector, type="o", col="blue", main="Hidden Markov Models (Sunday Mornings)",
 xlab = "Number of States", ylab = "Log Likelihood of Model")
```



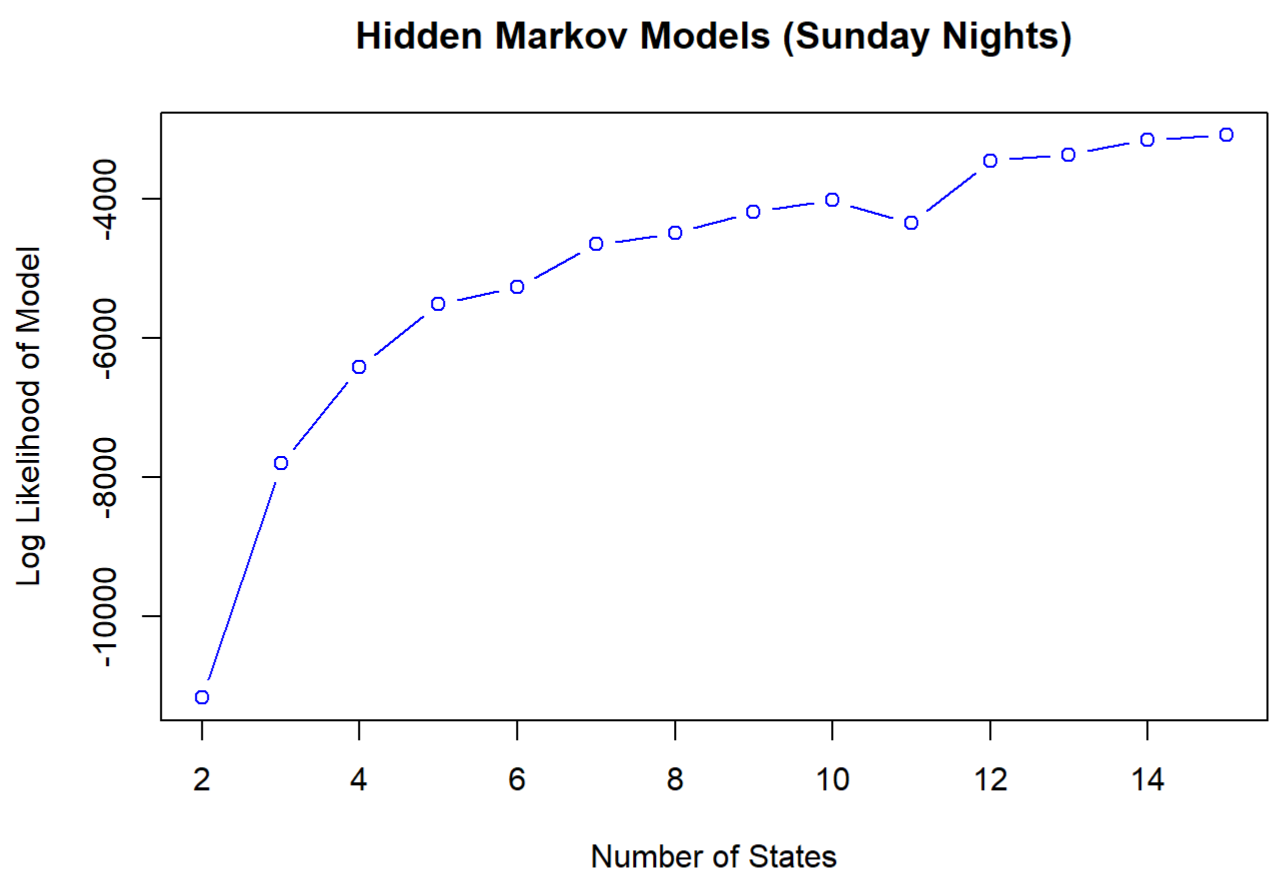
Results: We split the data set into the time frame of Sunday Mornings which are 8 AM to 11 AM (inclusive). Once the we have the dataset to perform the analysis on, we run the depmix function with 2 to 10 states and plot the results on a graph with the respective BIC values and the negative Log Likelihood values. From the plots we have obtained, we see that the ideal number of states for the sundayMorning data set is 8. The reason we chose 8 states is because the BIC value goes up with 9 states. Any number of states higher than 8 seem to be overfitting the model. The exact BIC value when the model has 8 states is: 5833.805 The exact Log Likelihood when the model has 8 states is: -2555.488

Question 2

```
# Question 2

set.seed(1)
BICvector <- vector()
logLikVector <- vector()
states <- c(2:15)
vectorntimes = rep(181, nrow(sundayNights)/181)

for (state in states) {
  print(paste("Number of States: ", state))
  mod <- depmix(response = Global_active_power ~ 1, data = sundayNights, nstates = state, ntimes
 = vectorntimes)
  fm <- fit(mod)
  summary(fm)
  BICvector <- c(BICvector, BIC(fm))
  logLikVector <- c(logLikVector, logLik(fm))
}
```



```
plot(states, logLikVector, type="b", col="blue", main="Hidden Markov Models (Sunday Nights)", xl
ab = "Number of States", ylab = "Log Likelihood of Model")
```



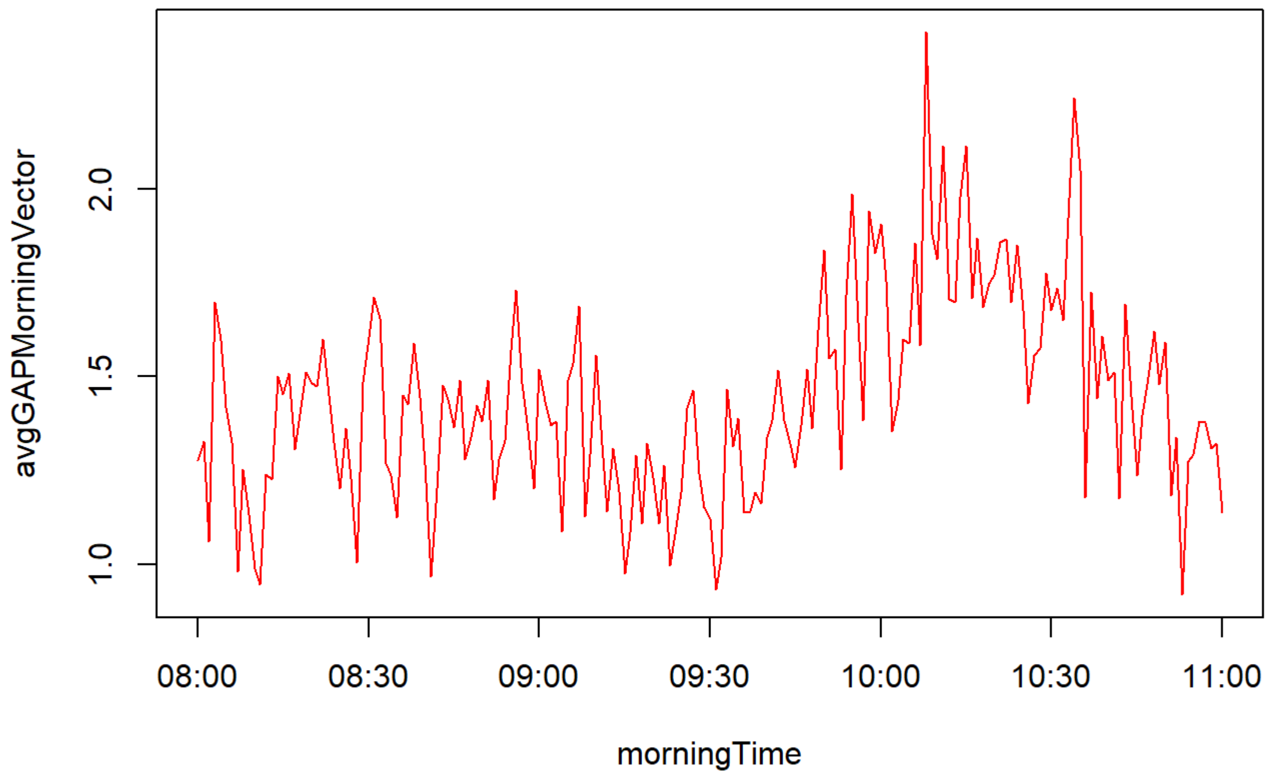
Results: We split the data set into the time frame of Sunday Nights which are 9 PM to 12 AM (inclusive). Once we have the dataset to perform the analysis on, we run the depmix function with 2 to 15 states and plot the results on a graph with the respective BIC values and the Log Likelihood values. From the plots we have obtained, we see that the ideal number of states for the Sunday Nights data set is 10. The reason we chose 10 states is because the BIC value goes up with 11 states. Any number of states higher than 10 seem to be overfitting the model. The exact BIC value when the model has 10 states is: 9133.780 The exact Log Likelihood when the model has 10 states is: -4022.480

Question 3 part a

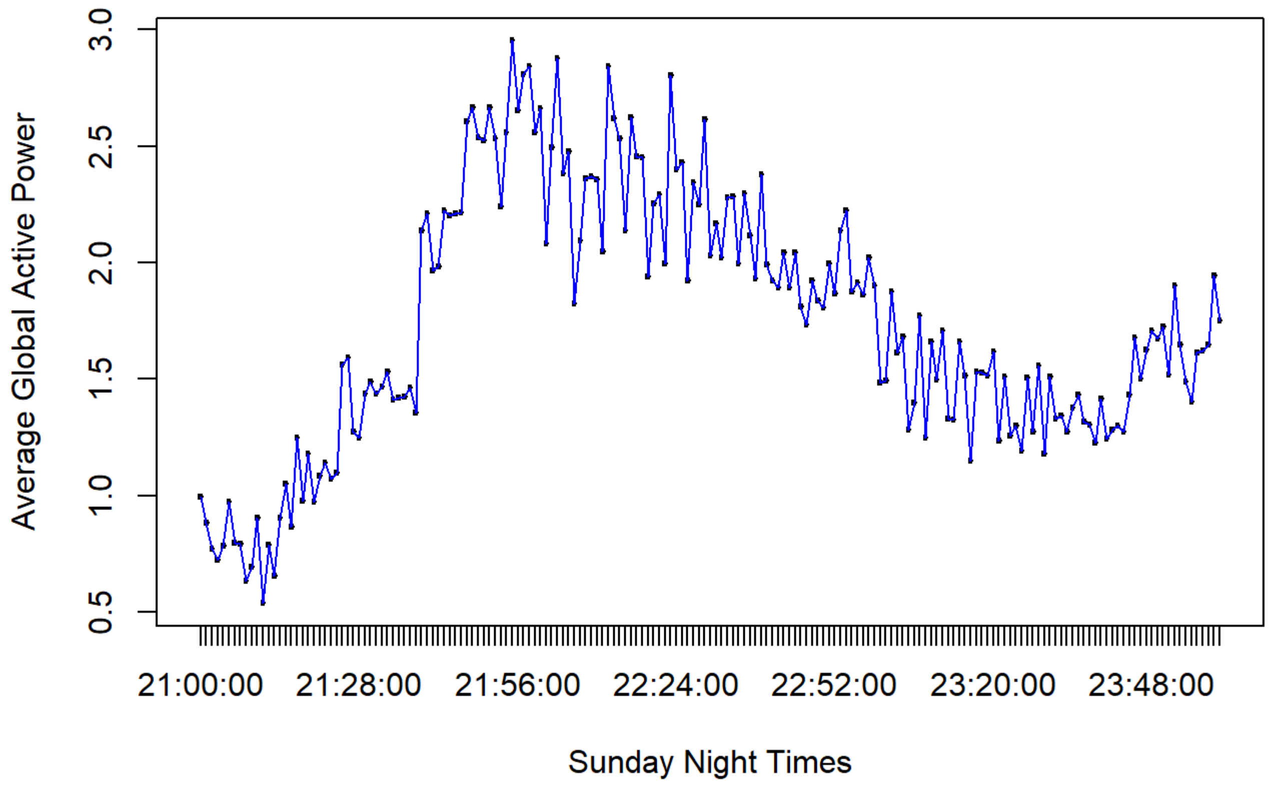
```
c <- 0
avgGAPMorningVector <- vector()
morningTime <- unique(format(sundayMornings$Time, na.rm = T, format = "%H:%M:%S"))
for (time in morningTime) {
  c <- c + 1
  avgGAPMorning <- mean(sundayMornings$Global_active_power[times == time], na.rm = T)
  avgGAPMorningVector <- c(avgGAPMorningVector, avgGAPMorning)
  print(paste(c, "Average Global_active_power for Sunday mornings at time", time, "=", avgGAPMor
ning))
}
```

```
## [1] "162 Average Global_active_power for Sunday nights at time 23:41:00 = 1.28415833333333"
## [1] "163 Average Global_active_power for Sunday nights at time 23:42:00 = 1.29998333333333"
## [1] "164 Average Global_active_power for Sunday nights at time 23:43:00 = 1.27534166666667"
## [1] "165 Average Global_active_power for Sunday nights at time 23:44:00 = 1.431675"
## [1] "166 Average Global_active_power for Sunday nights at time 23:45:00 = 1.6761"
## [1] "167 Average Global_active_power for Sunday nights at time 23:46:00 = 1.50066666666667"
## [1] "168 Average Global_active_power for Sunday nights at time 23:47:00 = 1.62681666666667"
## [1] "169 Average Global_active_power for Sunday nights at time 23:48:00 = 1.707025"
## [1] "170 Average Global_active_power for Sunday nights at time 23:49:00 = 1.6717"
## [1] "171 Average Global_active_power for Sunday nights at time 23:50:00 = 1.727075"
## [1] "172 Average Global_active_power for Sunday nights at time 23:51:00 = 1.51685833333333"
## [1] "173 Average Global_active_power for Sunday nights at time 23:52:00 = 1.90211666666667"
## [1] "174 Average Global_active_power for Sunday nights at time 23:53:00 = 1.64964166666667"
## [1] "175 Average Global_active_power for Sunday nights at time 23:54:00 = 1.4895"
## [1] "176 Average Global_active_power for Sunday nights at time 23:55:00 = 1.40423333333333"
## [1] "177 Average Global_active_power for Sunday nights at time 23:56:00 = 1.61224166666667"
## [1] "178 Average Global_active_power for Sunday nights at time 23:57:00 = 1.62295833333333"
## [1] "179 Average Global_active_power for Sunday nights at time 23:58:00 = 1.6475"
## [1] "180 Average Global_active_power for Sunday nights at time 23:59:00 = 1.94539166666667"
## [1] "181 Average Global_active_power for Sunday nights at time 00:00:00 = 1.74920714285714"
```

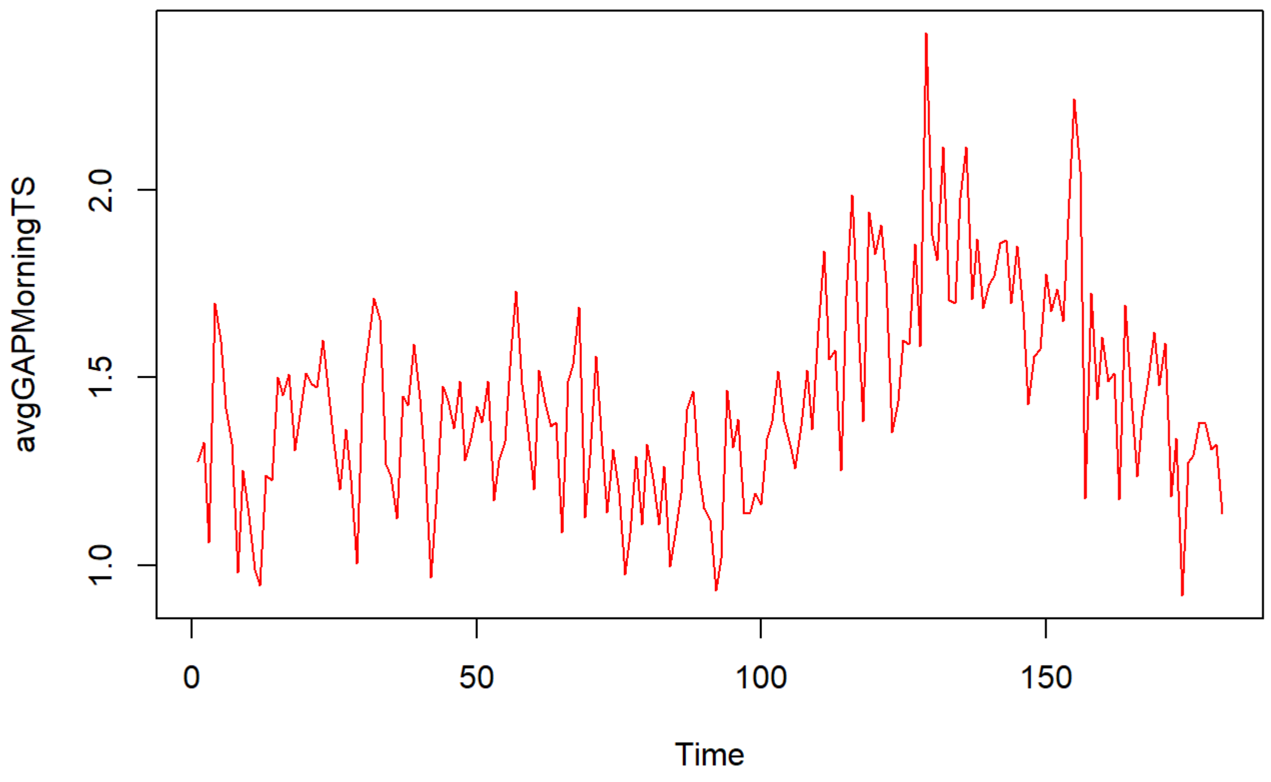
```
# Plot Average Global_active_power for Sunday mornings and nights
morningTime <- strptime(morningTime, format = "%H:%M:%S")
plot(morningTime, avgGAPMorningVector, ty="l", col="red")
```



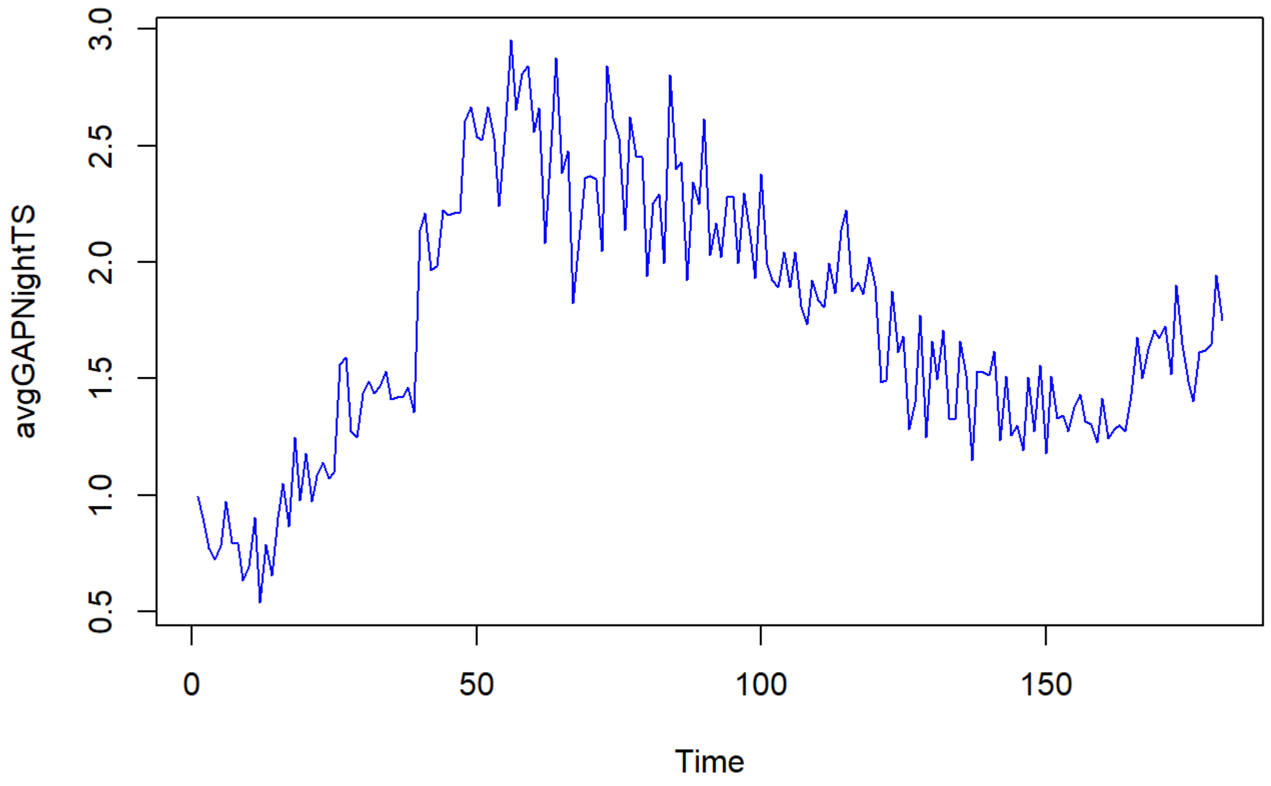
```
nightTime <- factor(nightTime, levels=sundayNights[0:181,]$Time)
plot(nightTime, avgGAPNightVector, type="n", col="blue", xlab="Sunday Night Times", ylab="Averag
e Global Active Power")
lines(nightTime, avgGAPNightVector, type="l", col="blue", xlab="Sunday Night Times", ylab="Avera
ge Global Active Power")
```



```
# Plot Time Series for average Global_active_power for Sunday mornings and nights
avgGAPMorningTS <- ts(avgGAPMorningVector)
avgGAPNightTS <- ts(avgGAPNightVector)
plot.ts(avgGAPMorningTS, col="red")
```



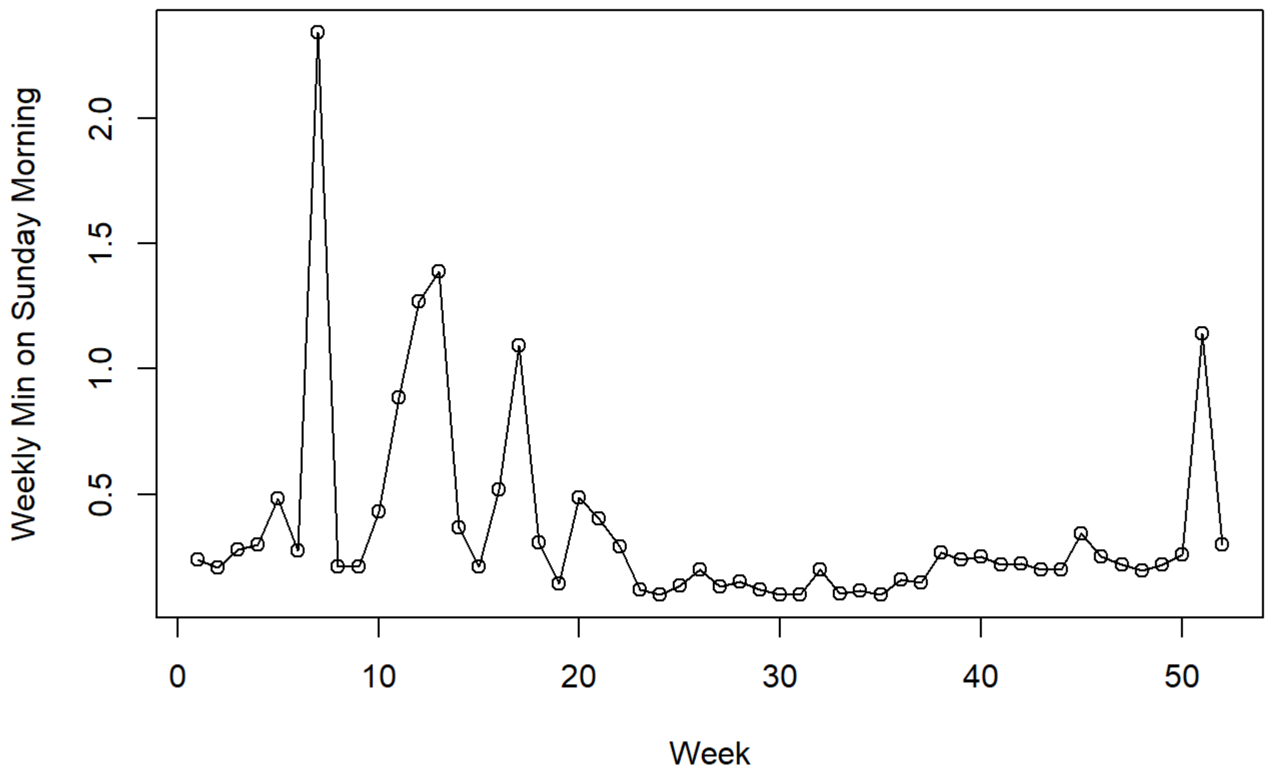
```
plot.ts(avgGAPNightTS, col="blue")
```



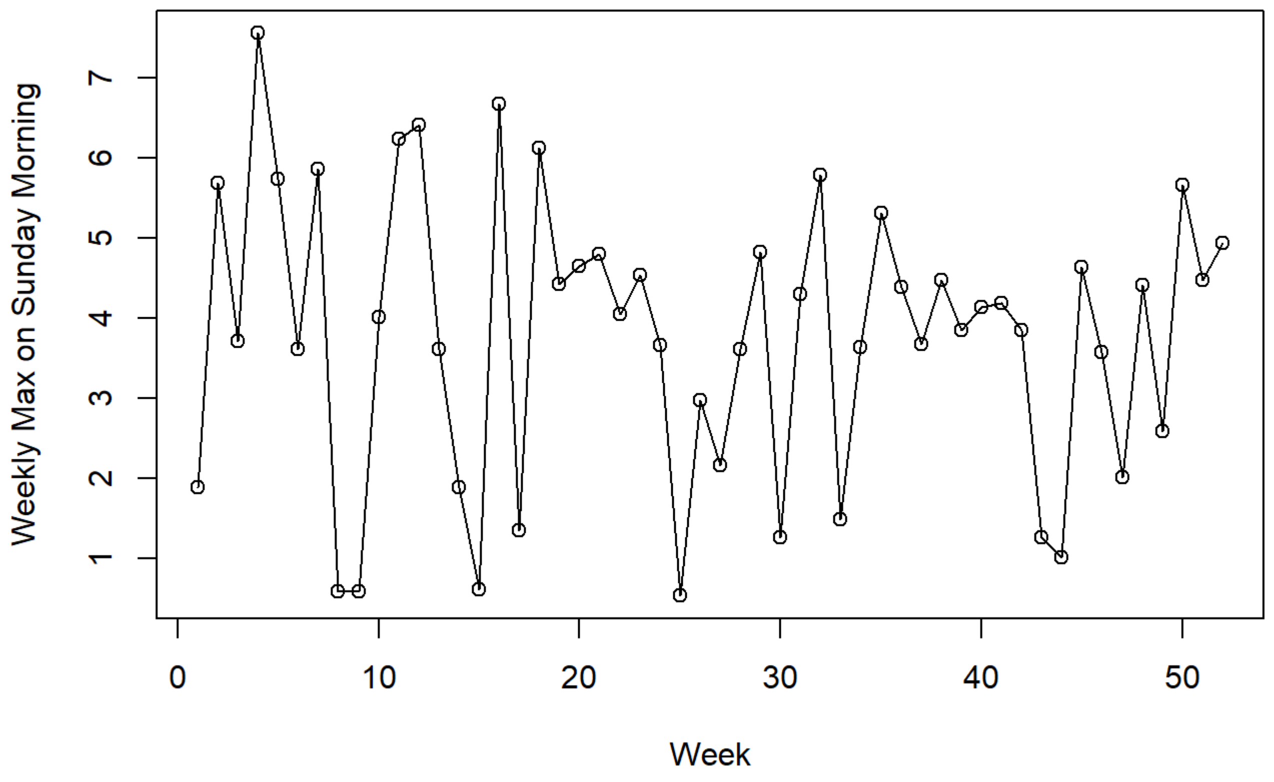
Explanation: The red plots represent the average global active power plotted against morning times. The blue plots are for Sunday night times. We see that from 9:30 PM to 11 PM the most electricity was consumed. We see that for the Sunday morning times the electricity consumption is high from 9:30 AM to 11 AM.

Question 3 part b

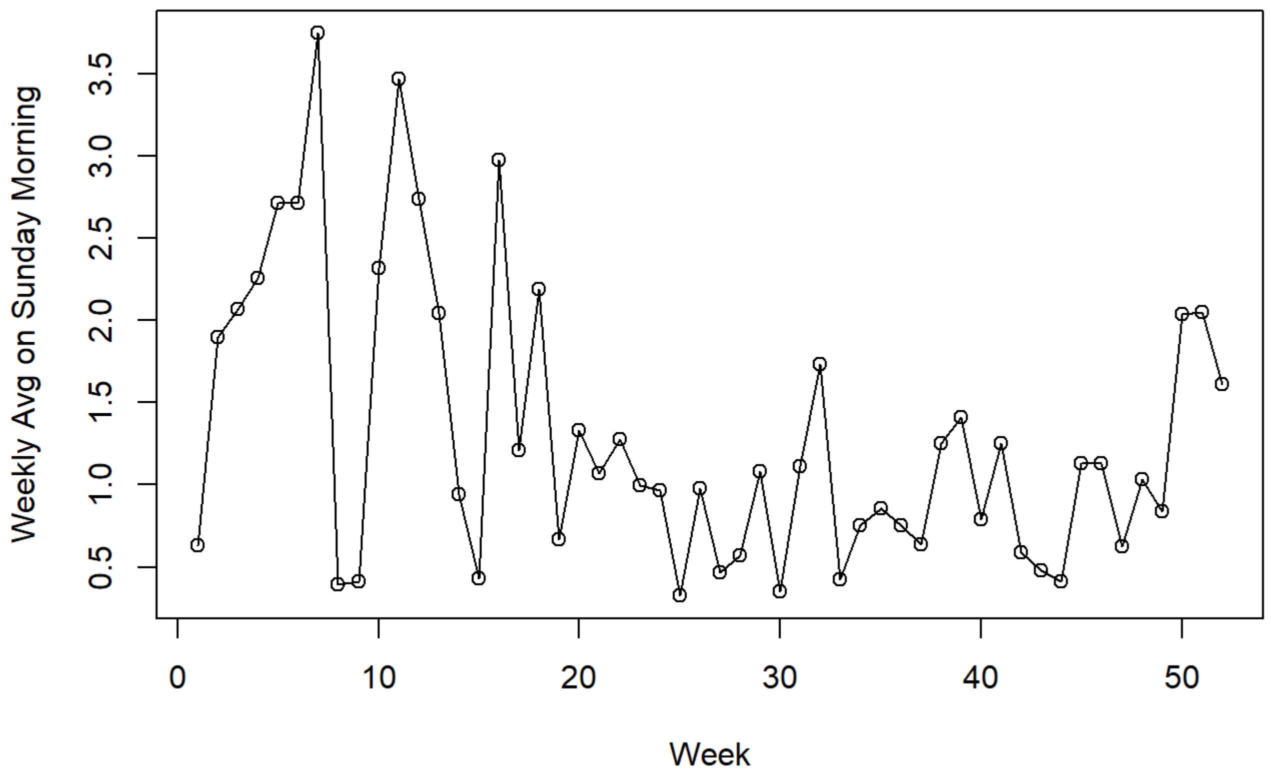
```
sundaydate <- unique(sundayMornings$Date)

# Defining weekly statistical measurements
weekeveningmin <- vector()
weekeveningmax <- vector()
weekeveningavg <- vector()
weekeveningsd  <- vector()
weekmorningmin <- vector()
weekmorningmax <- vector()
weekmorningavg <- vector()
weekmorningsd  <- vector()

# Defining monthly statistical measurements
monthmorningmin <- vector()
monthmorningmax <- vector()
monthmorningavg <- vector()
monthmorningsd  <- vector()
montheveningmin <- vector()
montheveningmax <- vector()
montheveningavg <- vector()
montheveningsd  <- vector()

# Defining seasonal statistical measurements
seasonmorningmin <- vector()
seasonmorningmax <- vector()
seasonmorningavg <- vector()
seasonmorningsd  <- vector()
seasoneveningmin <- vector()
seasoneveningmax <- vector()
seasoneveningavg <- vector()
seasoneveningsd  <- vector()

i = 0
for (Date in sundaydate){
  i <- i + 1
  minlist <- sort(sundayMornings$Global_active_power[which(sundayMornings$Date == Date)])
  maxlist <- sort(minlist,decreasing = T)
  weekmorningmin[i] <- minlist[1]
  weekmorningmax[i] <- maxlist[1]
  weekmorningavg[i] <- mean(minlist)
  weekmorningsd[i]  <- sd(minlist)

  minlist <- sort(sundayNights$Global_active_power[((181*i)-180):(181*i)])
  maxlist <- sort(minlist,decreasing = T)
  weekeveningmin[i] <- minlist[1]
  weekeveningmax[i] <- maxlist[1]
  weekeveningavg[i] <- mean(minlist)
  weekeveningsd[i]  <- sd(minlist)
}

months <- (1:12)

for (i in 1:12){
  targetmon <- subset(sundayMornings, month(sundayMornings$Date) == i)
  minlist <- sort(targetmon$Global_active_power)
  maxlist <- sort(minlist,decreasing = T)
  monthmorningmin[i] <- minlist[1]
  monthmorningmax[i] <- maxlist[1]
  monthmorningavg[i] <- mean(minlist)
  monthmorningsd[i]  <- sd(minlist)

  targetmon <- subset(sundayNights, month(sundayNights$Date) == i)
  minlist <- sort(targetmon$Global_active_power)
  maxlist <- sort(minlist,decreasing = T)
  montheveningmin[i] <- minlist[1]
  montheveningmax[i] <- maxlist[1]
  montheveningavg[i] <- mean(minlist)
  montheveningsd[i]  <- sd(minlist)
}


for (i in 1:4){

  if (((3*i)+2) < 12){
    targetseason <- subset(sundayMornings, month(sundayMornings$Date) >= (3*i) & month(sundayMor
nings$Date) <= ((3*i)+2))
    minlist <- sort(targetseason$Global_active_power)
    maxlist <- sort(minlist,decreasing = T)
    seasonmorningmin[i] <- minlist[1]
    seasonmorningmax[i] <- maxlist[1]
    seasonmorningavg[i] <- mean(minlist)
    seasonmorningsd[i]  <- sd(minlist)

    targetseason <- subset(sundayNights, month(sundayNights$Date) >= (3*i) & month(sundayNights
$Date) <= ((3*i)+2))
    minlist <- sort(targetseason$Global_active_power)
    maxlist <- sort(minlist,decreasing = T)
    seasoneveningmin[i] <- minlist[1]
    seasoneveningmax[i] <- maxlist[1]
    seasoneveningavg[i] <- mean(minlist)
    seasoneveningsd[i]  <- sd(minlist)
  }else{
    targetseason <- subset(sundayMornings, month(sundayMornings$Date) == 1 | month(sundayMorning
s$Date) == 12 | month(sundayMornings$Date) == 2)
    minlist <- sort(targetseason$Global_active_power)
    maxlist <- sort(minlist,decreasing = T)
    seasonmorningmin[i] <- minlist[1]
    seasonmorningmax[i] <- maxlist[1]
    seasonmorningavg[i] <- mean(minlist)
    seasonmorningsd[i]  <- sd(minlist)

    targetseason <- subset(sundayNights, month(sundayNights$Date) == 1 | month(sundayNights$Dat
e) == 12 | month(sundayNights$Date) == 2)
    minlist <- sort(targetseason$Global_active_power)
    maxlist <- sort(minlist,decreasing = T)
    seasoneveningmin[i] <- minlist[1]
    seasoneveningmax[i] <- maxlist[1]
    seasoneveningavg[i] <- mean(minlist)
    seasoneveningsd[i]  <- sd(minlist)
  }
}

# Weekly
plot(1:52,weekmorningmin, xlab ='Week', ylab = 'Weekly Min on Sunday Morning', type = "o")
```



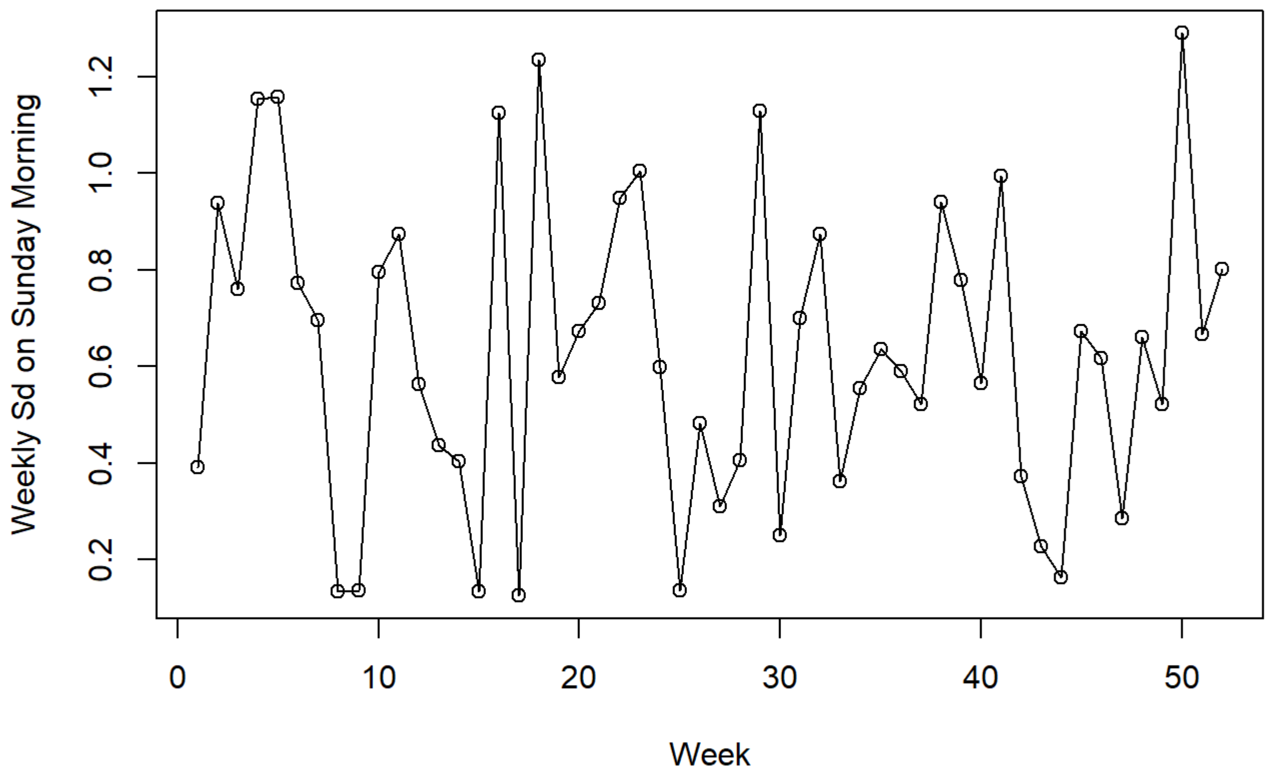
```
plot(1:52,weekmorningmax, xlab ='Week', ylab = 'Weekly Max on Sunday Morning', type = "o")
```



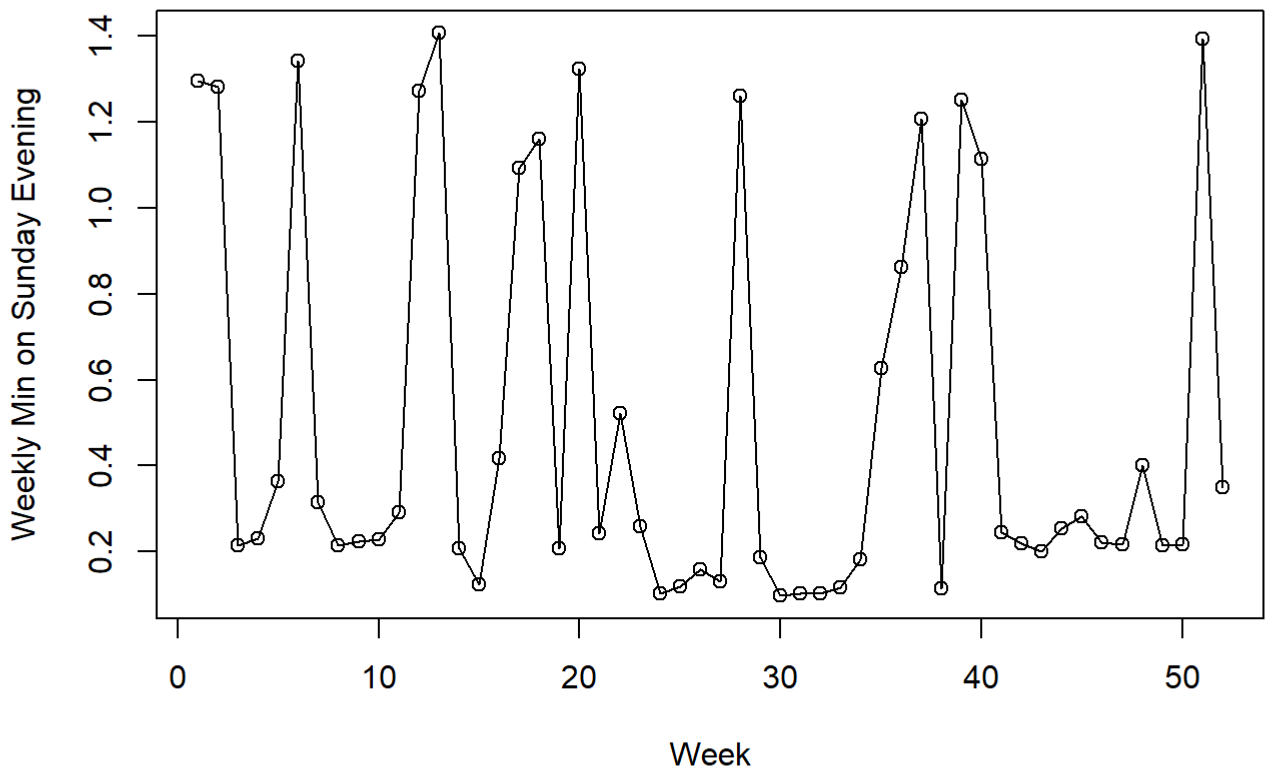
```
plot(1:52,weekmorningavg, xlab ='Week', ylab = 'Weekly Avg on Sunday Morning', type = "o")
```



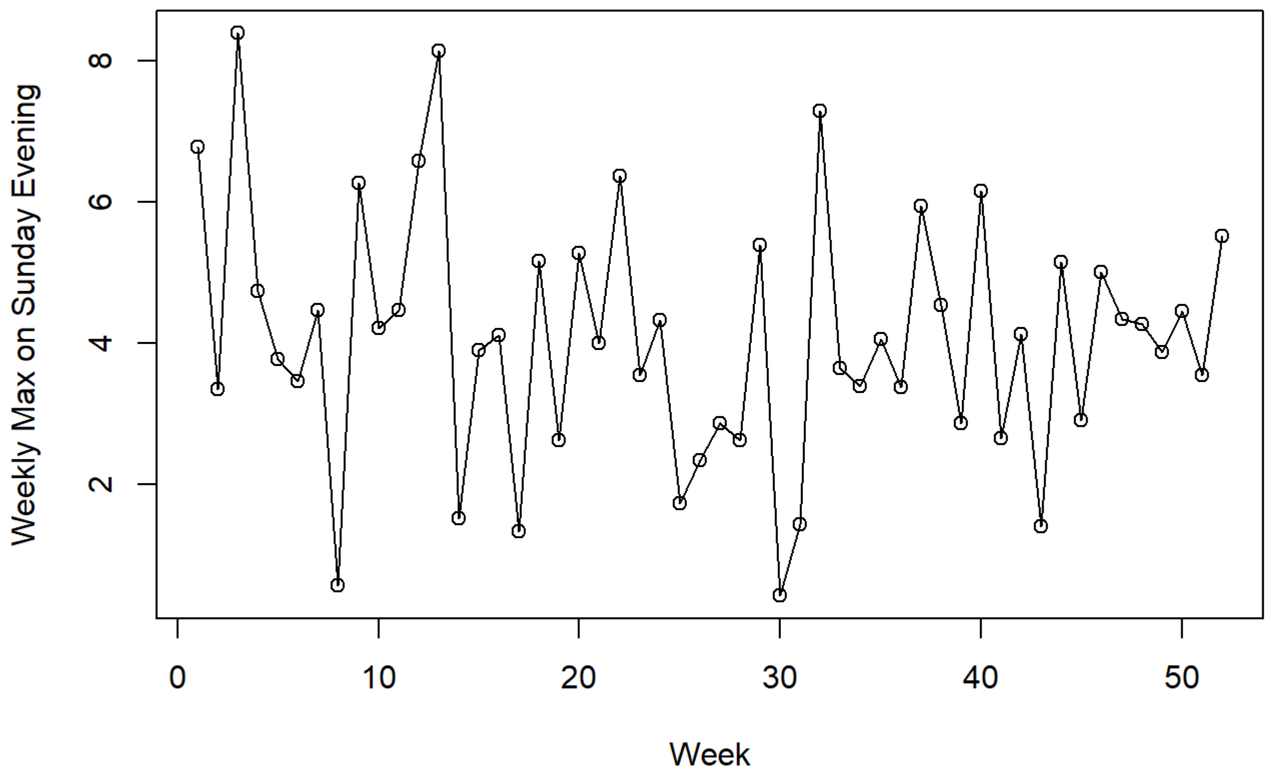
```
plot(1:52,weekmorningsd,  xlab ='Week', ylab = 'Weekly Sd on Sunday Morning',  type = "o")
```



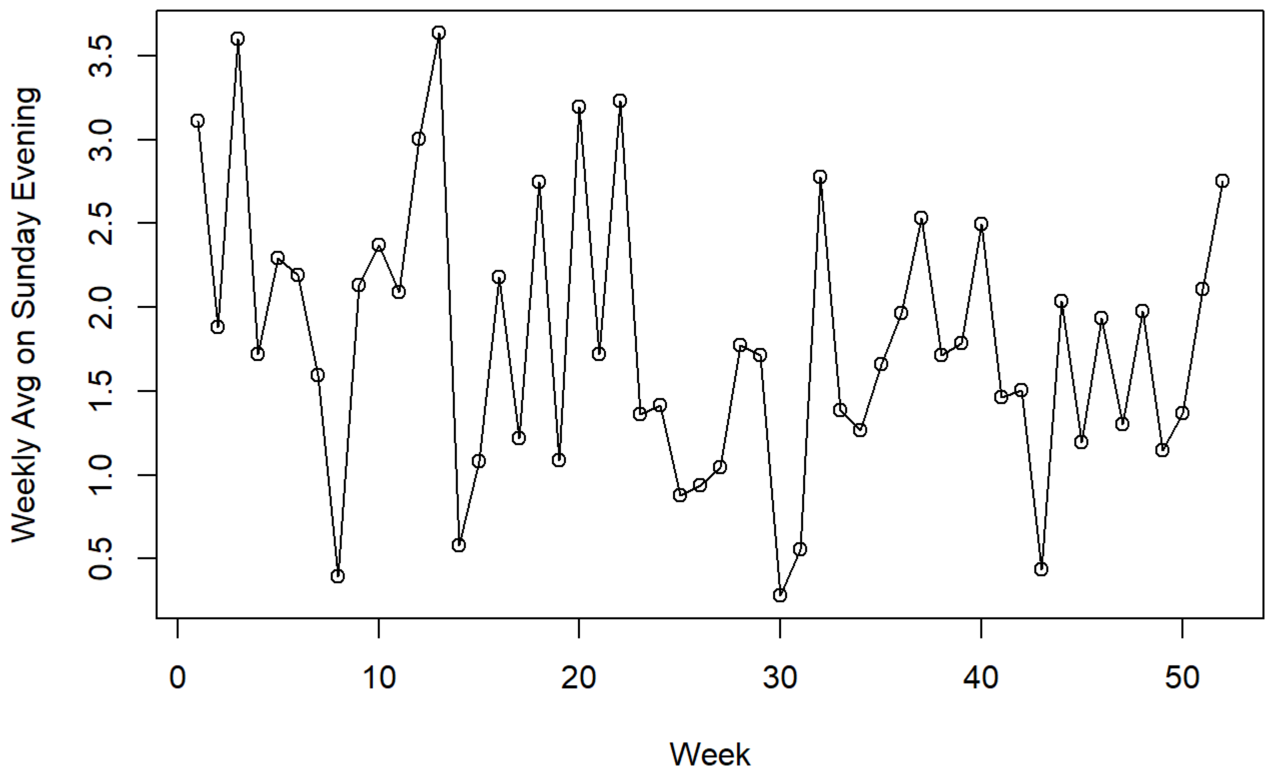
```
plot(1:52,weekeveningmin, xlab ='Week', ylab = 'Weekly Min on Sunday Evening', type = "o")
```



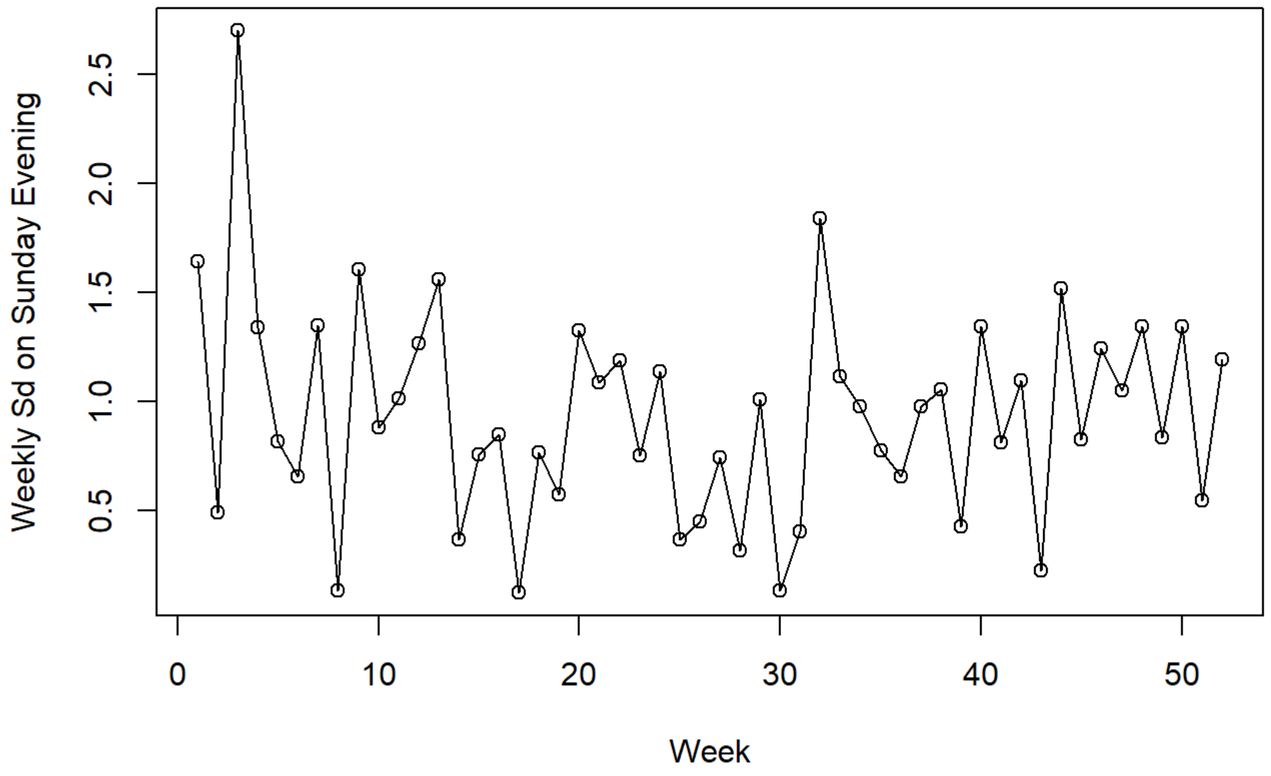
```
plot(1:52,weekeveningmax, xlab ='Week', ylab = 'Weekly Max on Sunday Evening', type = "o")
```



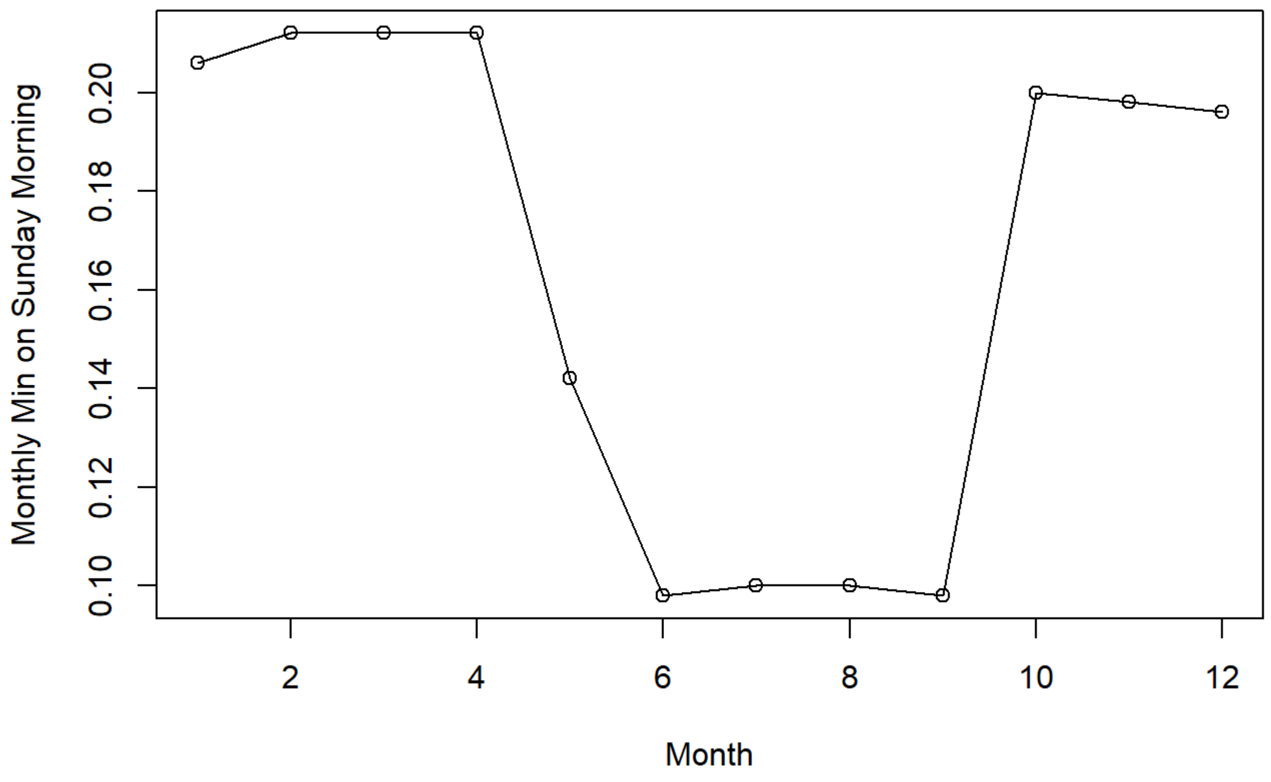
```
plot(1:52,weekeveningavg, xlab ='Week', ylab = 'Weekly Avg on Sunday Evening', type = "o")
```



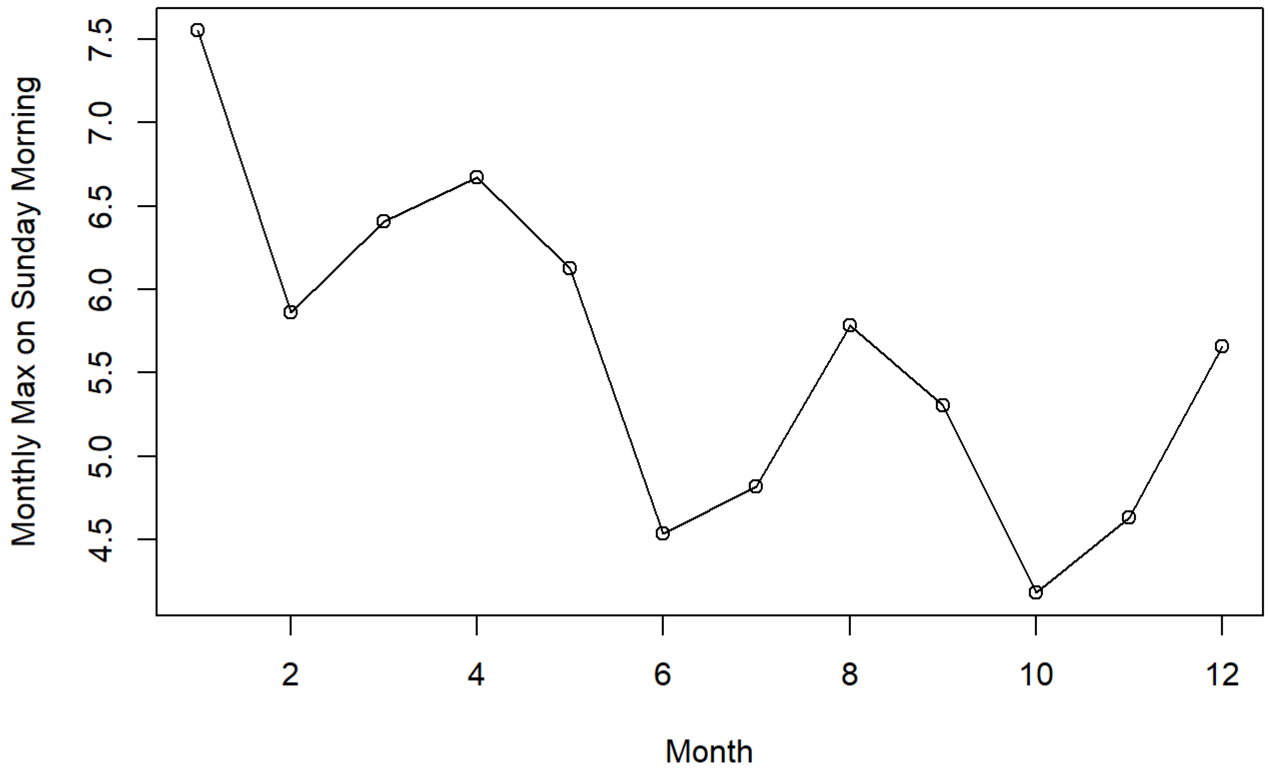
```
plot(1:52,weekeveningsd,  xlab ='Week', ylab = 'Weekly Sd on Sunday Evening',  type = "o")
```



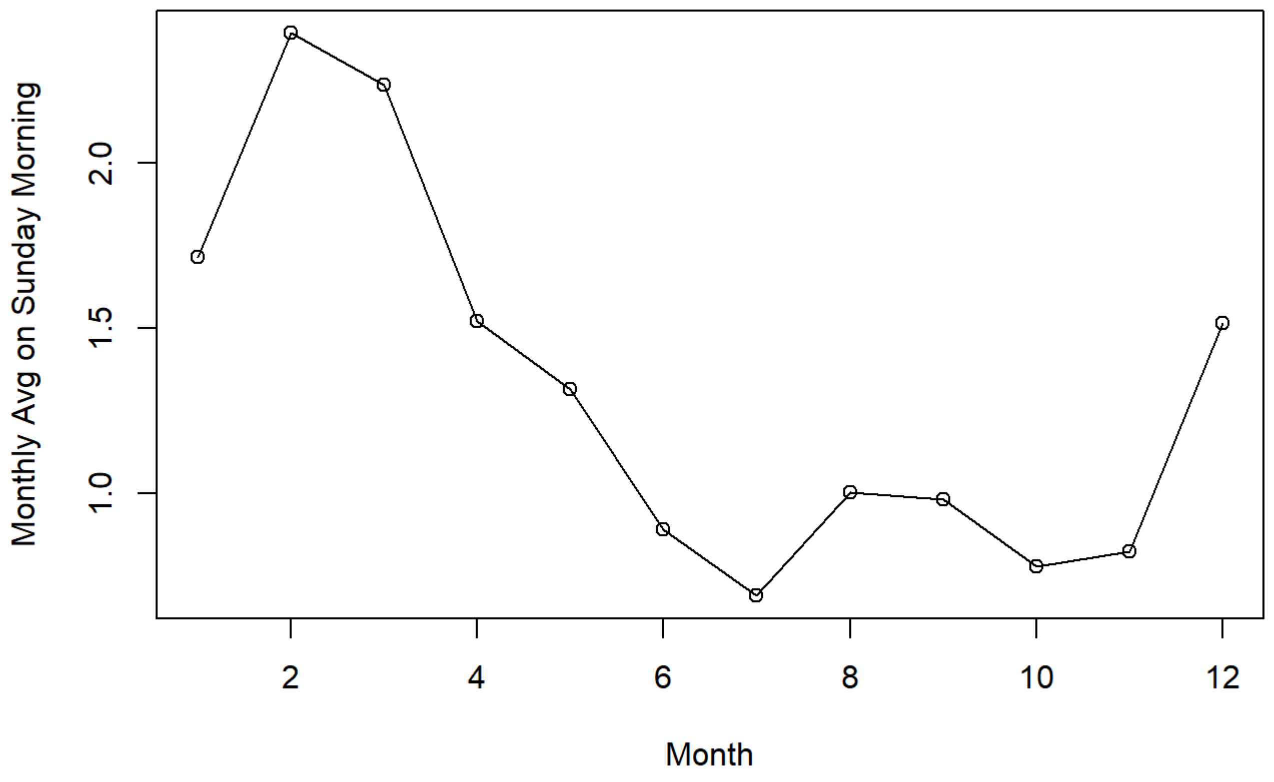
```
# Monthly
plot(1:12,monthmorningmin, xlab ='Month', ylab = 'Monthly Min on Sunday Morning', type = "o")
```



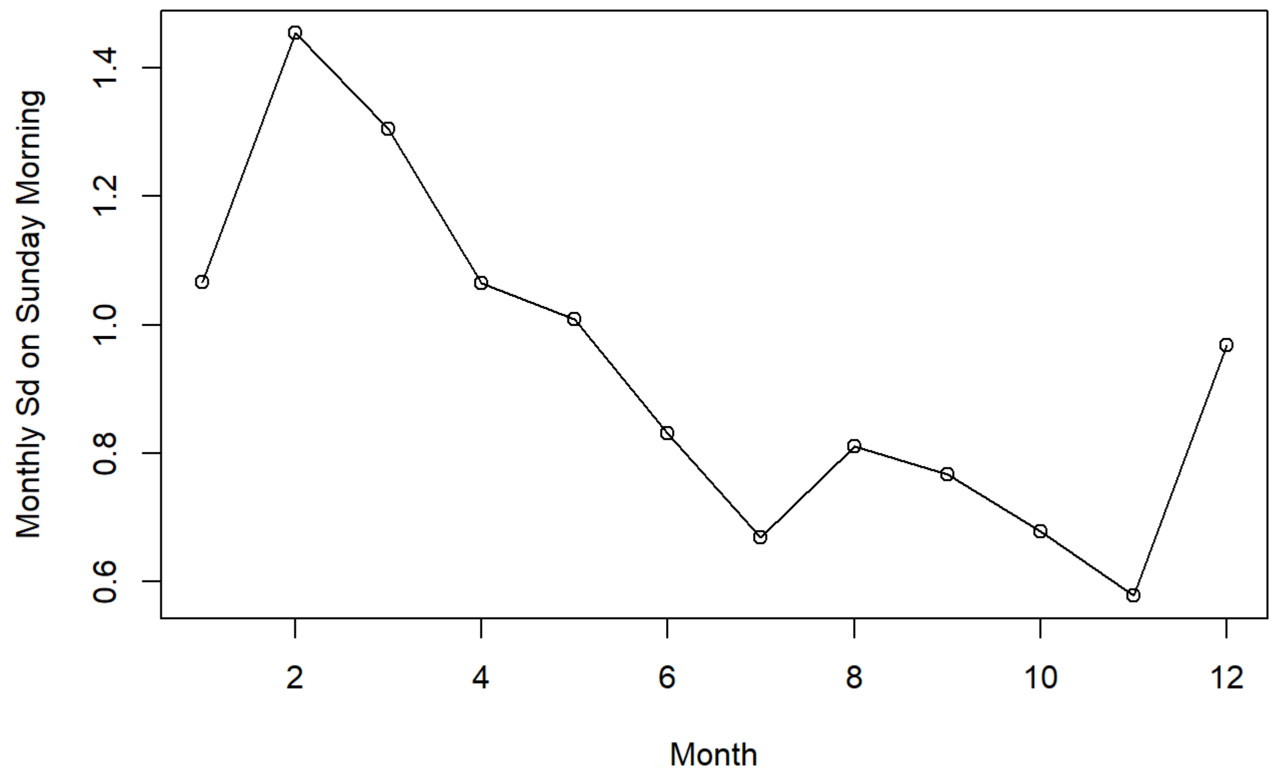
```
plot(1:12,monthmorningmax, xlab ='Month', ylab = 'Monthly Max on Sunday Morning', type = "o")
```



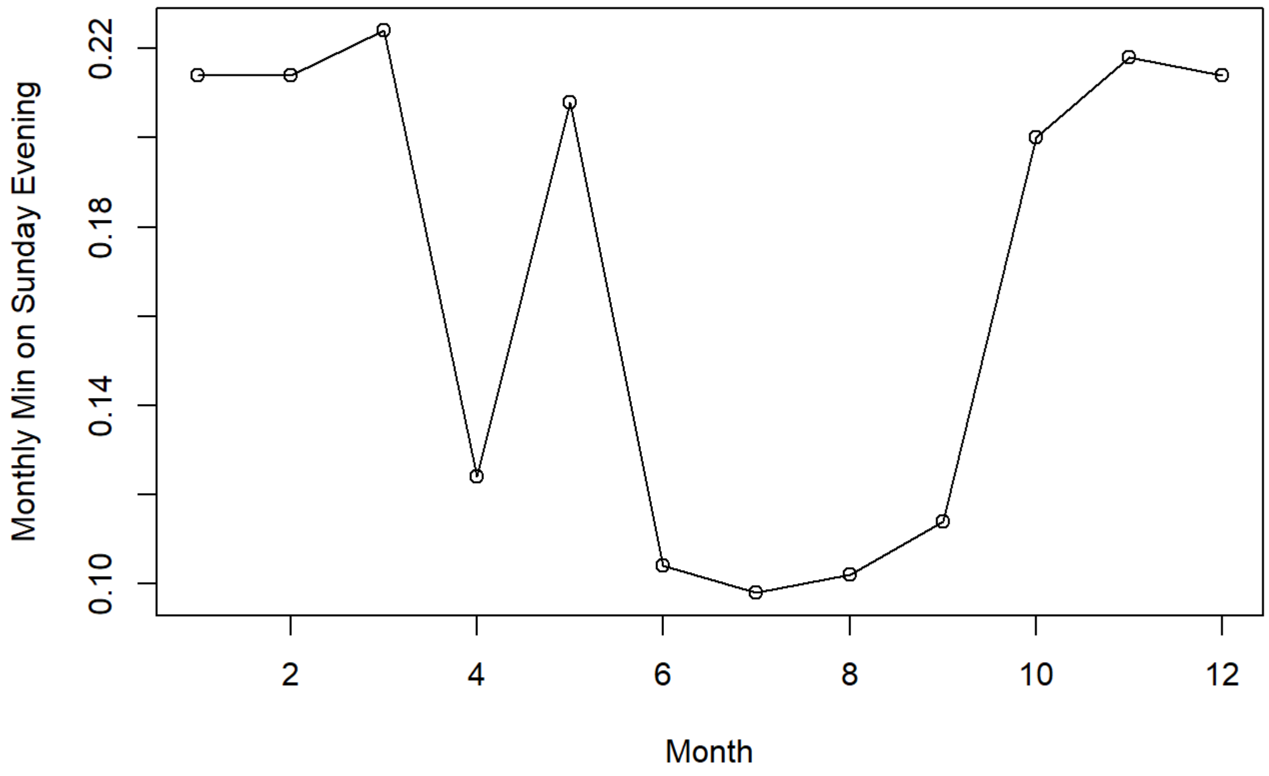
```
plot(1:12,monthmorningavg, xlab ='Month', ylab = 'Monthly Avg on Sunday Morning', type = "o")
```



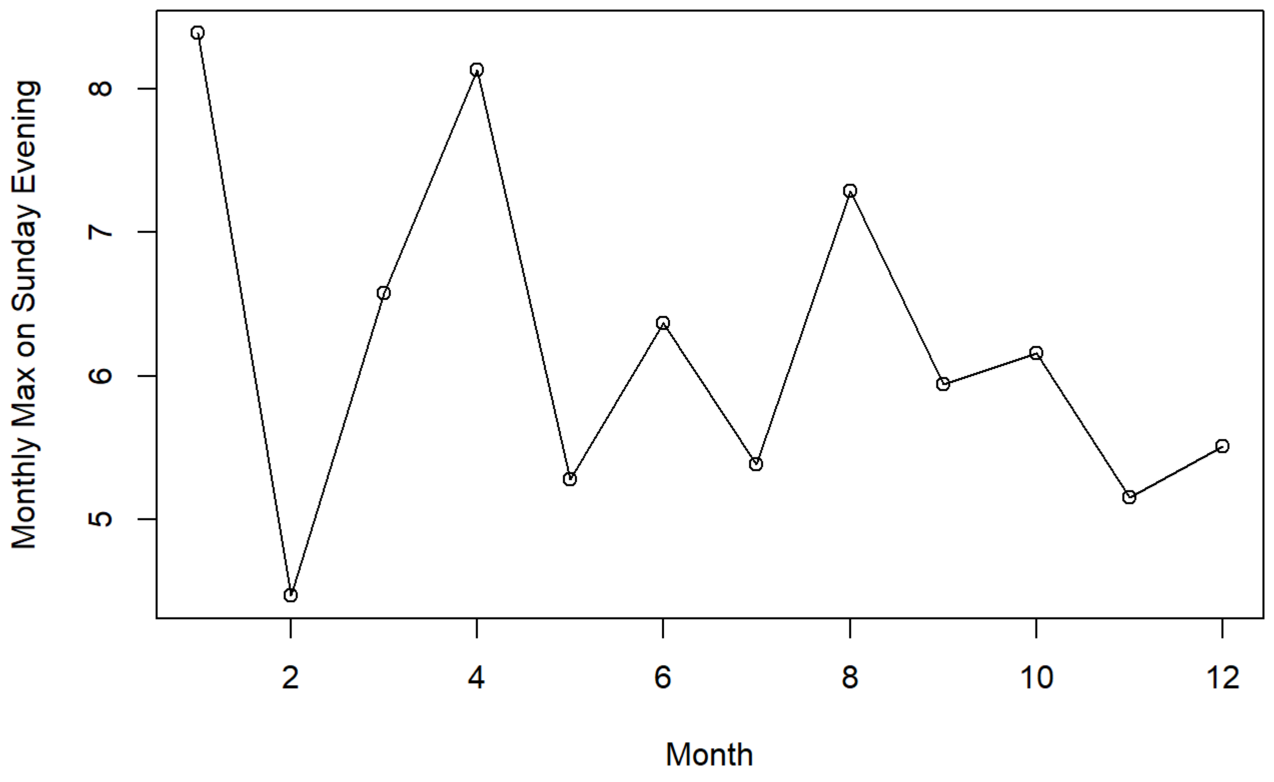
```
plot(1:12,monthmorningsd,  xlab ='Month', ylab = 'Monthly Sd on Sunday Morning',  type = "o")
```



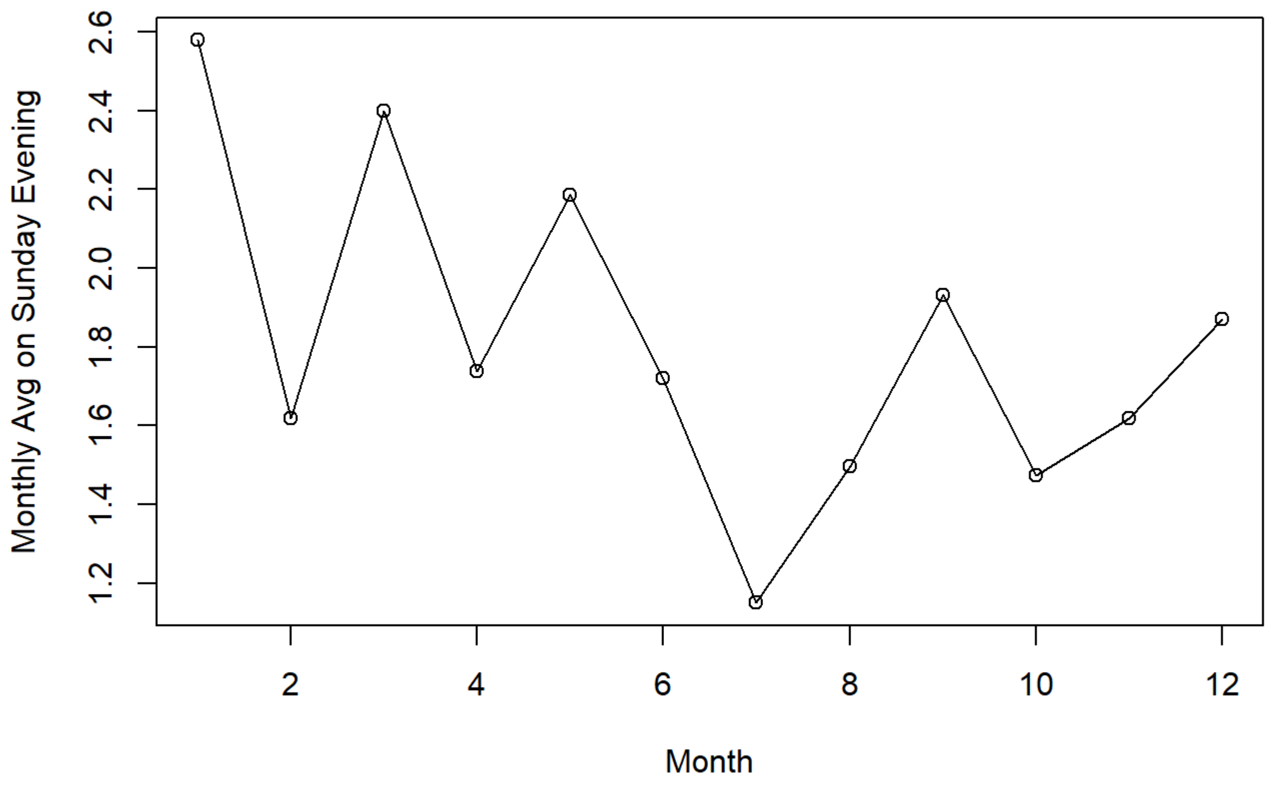
```
plot(1:12,montheveningmin, xlab ='Month', ylab = 'Monthly Min on Sunday Evening', type = "o")
```



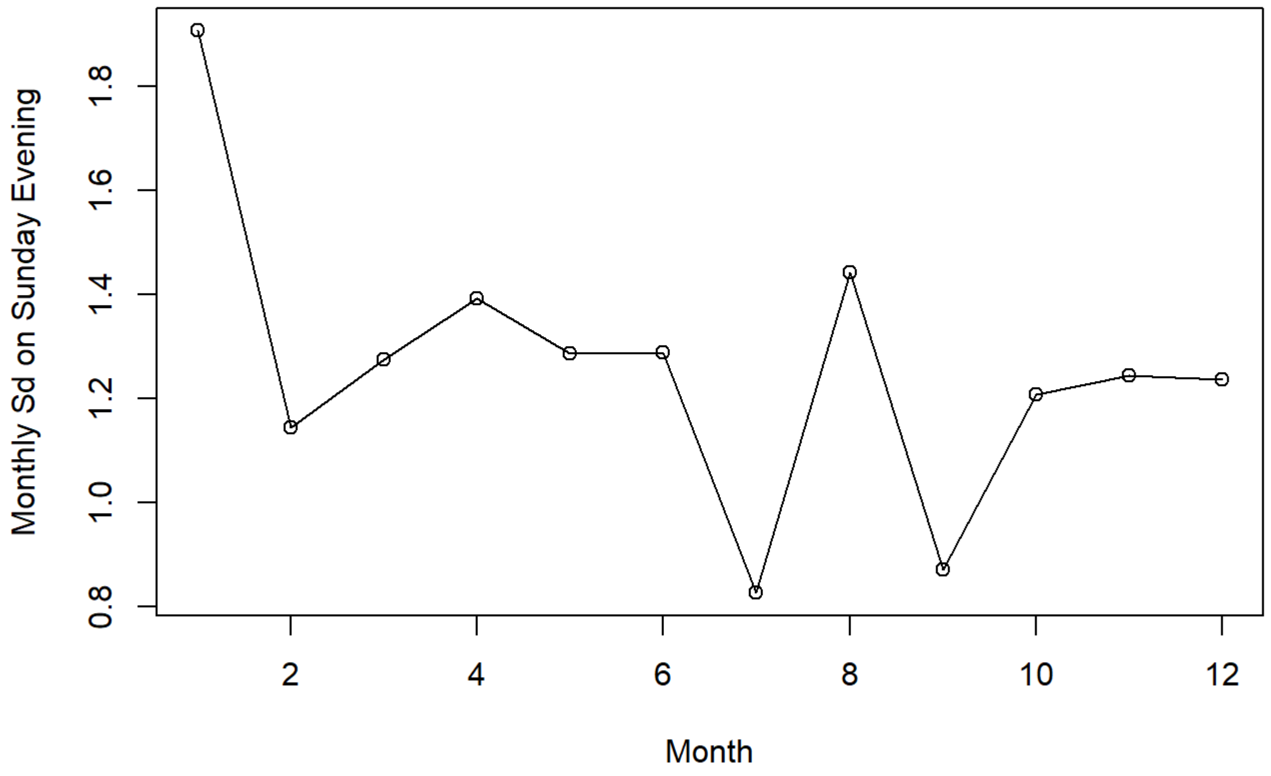
```
plot(1:12,montheveningmax, xlab ='Month', ylab = 'Monthly Max on Sunday Evening', type = "o")
```



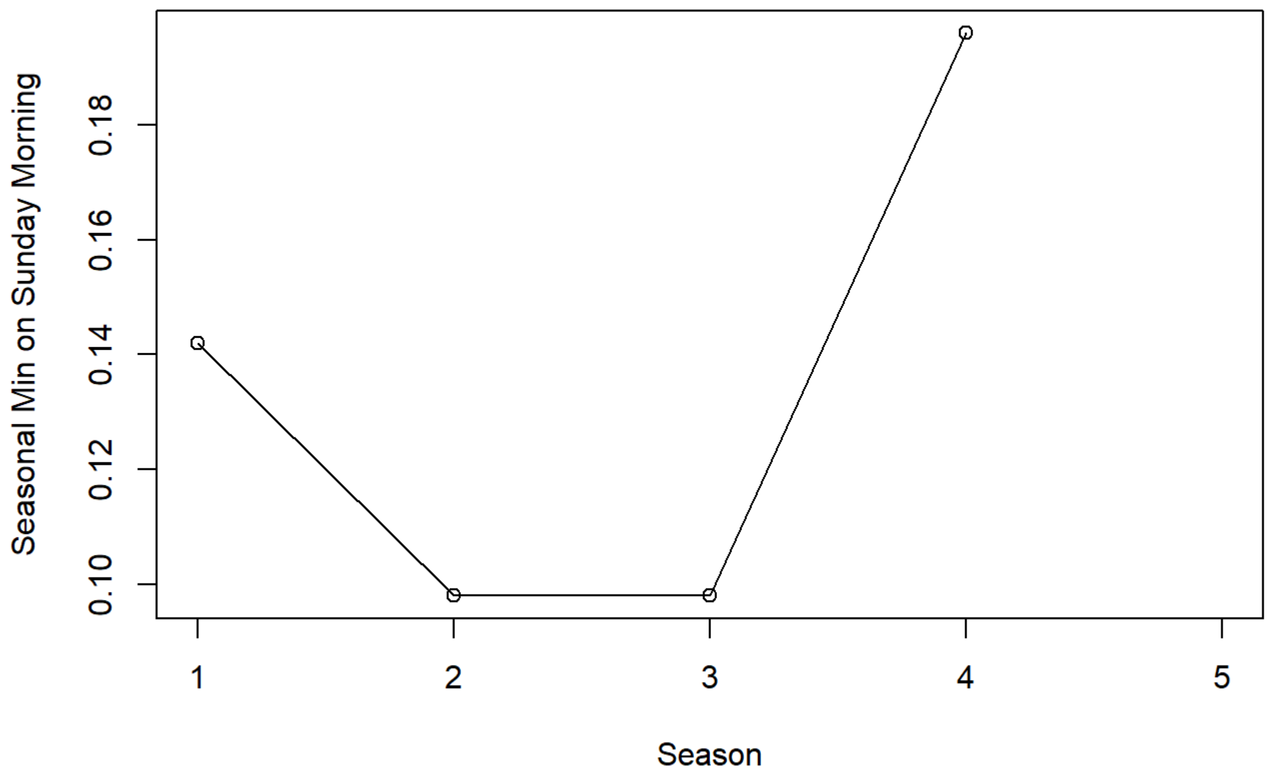
```
plot(1:12,montheveningavg, xlab ='Month', ylab = 'Monthly Avg on Sunday Evening', type = "o")
```



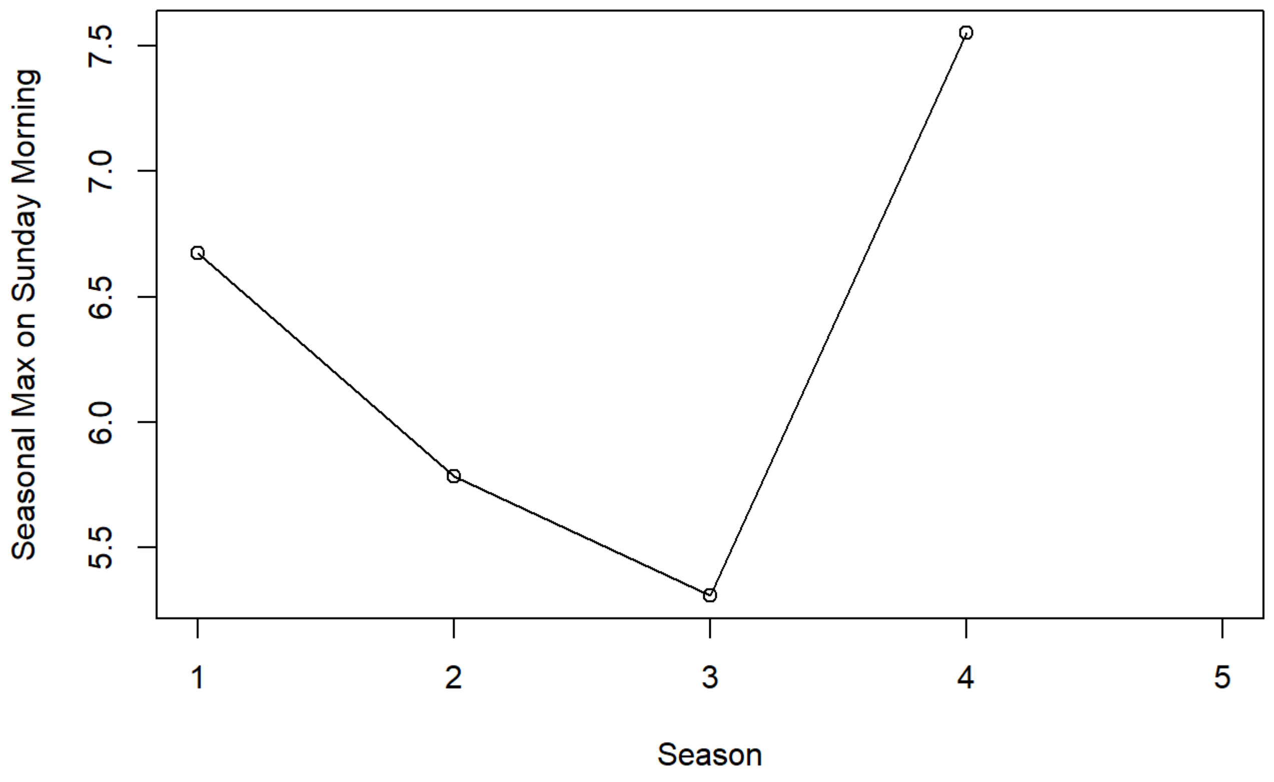
```
plot(1:12,montheveningsd,  xlab ='Month', ylab = 'Monthly Sd on Sunday Evening',  type = "o")
```



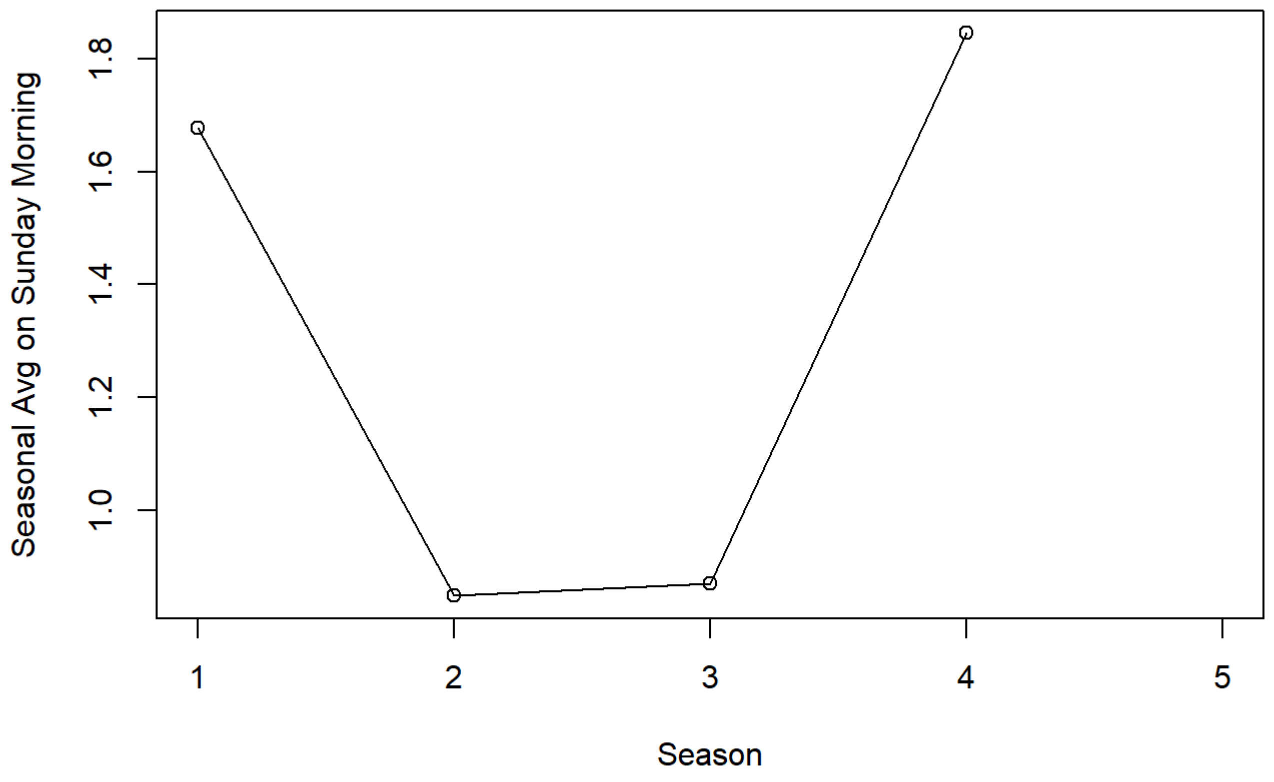
```
# Seasonal
plot(xlim=c(1,5),seasonmorningmin, xlab ='Season', ylab = 'Seasonal Min on Sunday Morning', type
 = "o")
```



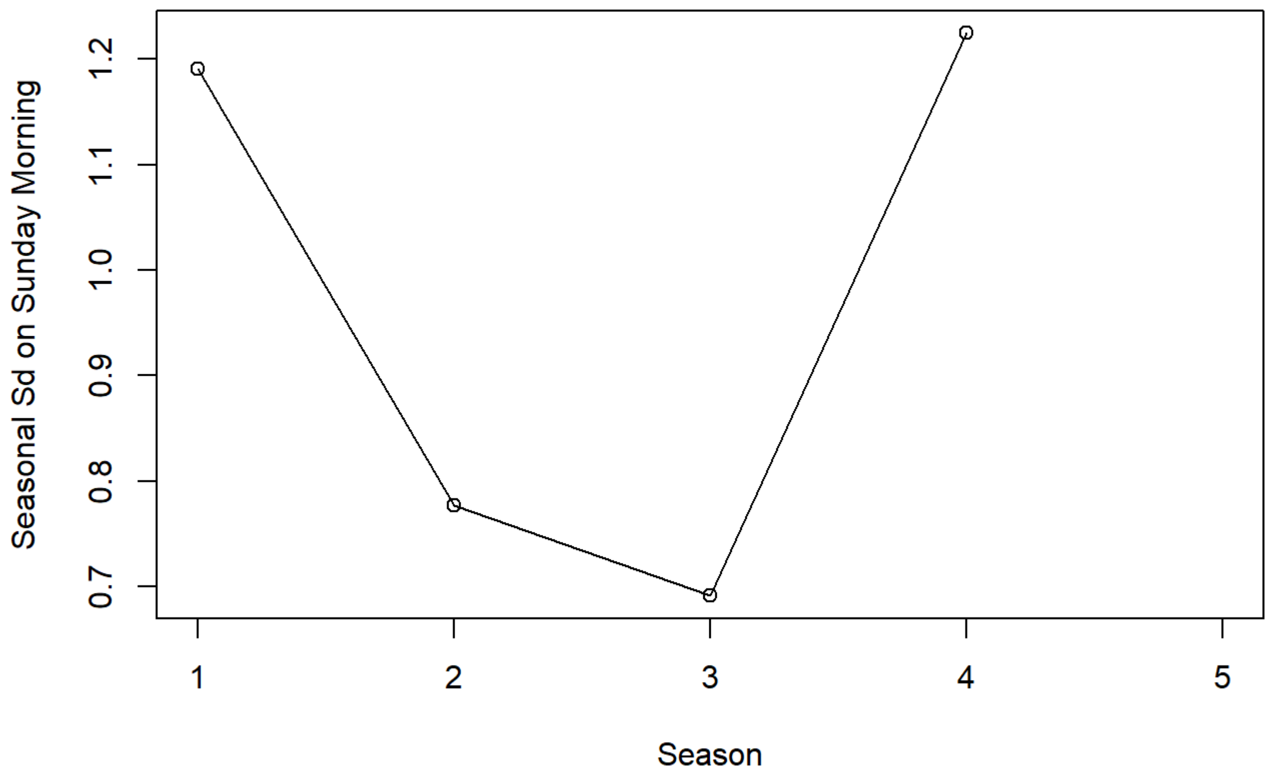
```
plot(xlim=c(1,5),seasonmorningmax, xlab ='Season', ylab = 'Seasonal Max on Sunday Morning', type
 = "o")
```



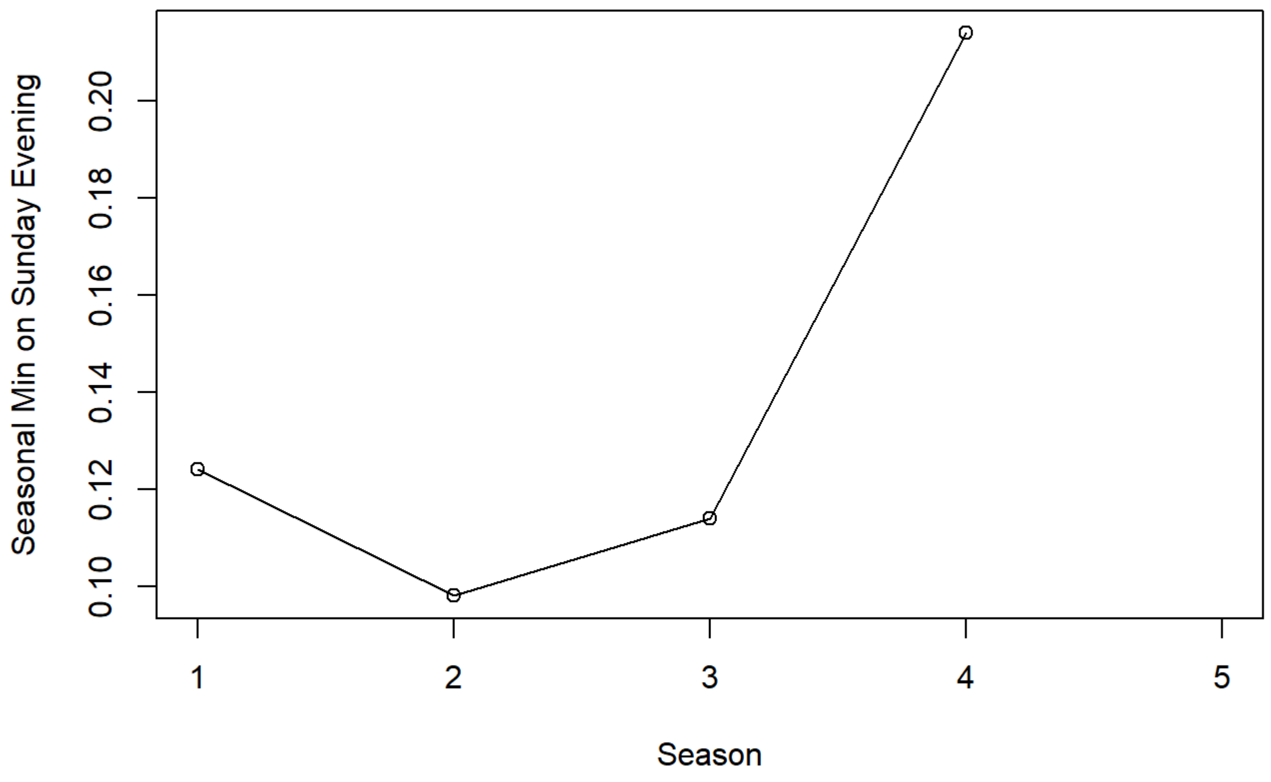
```
plot(xlim=c(1,5),seasonmorningavg, xlab ='Season', ylab = 'Seasonal Avg on Sunday Morning', type
 = "o")
```



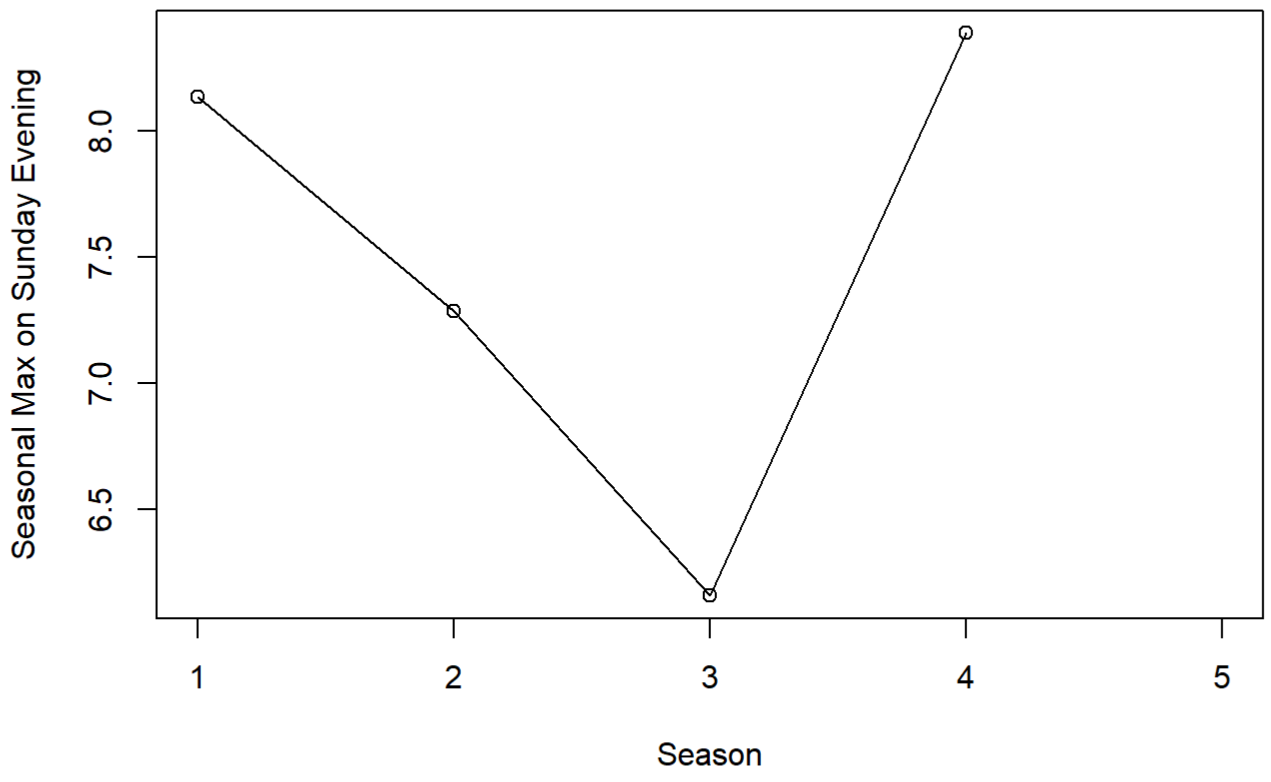
```
plot(xlim=c(1,5),seasonmorningsd,  xlab ='Season', ylab = 'Seasonal Sd on Sunday Morning',  type
 = "o")
```



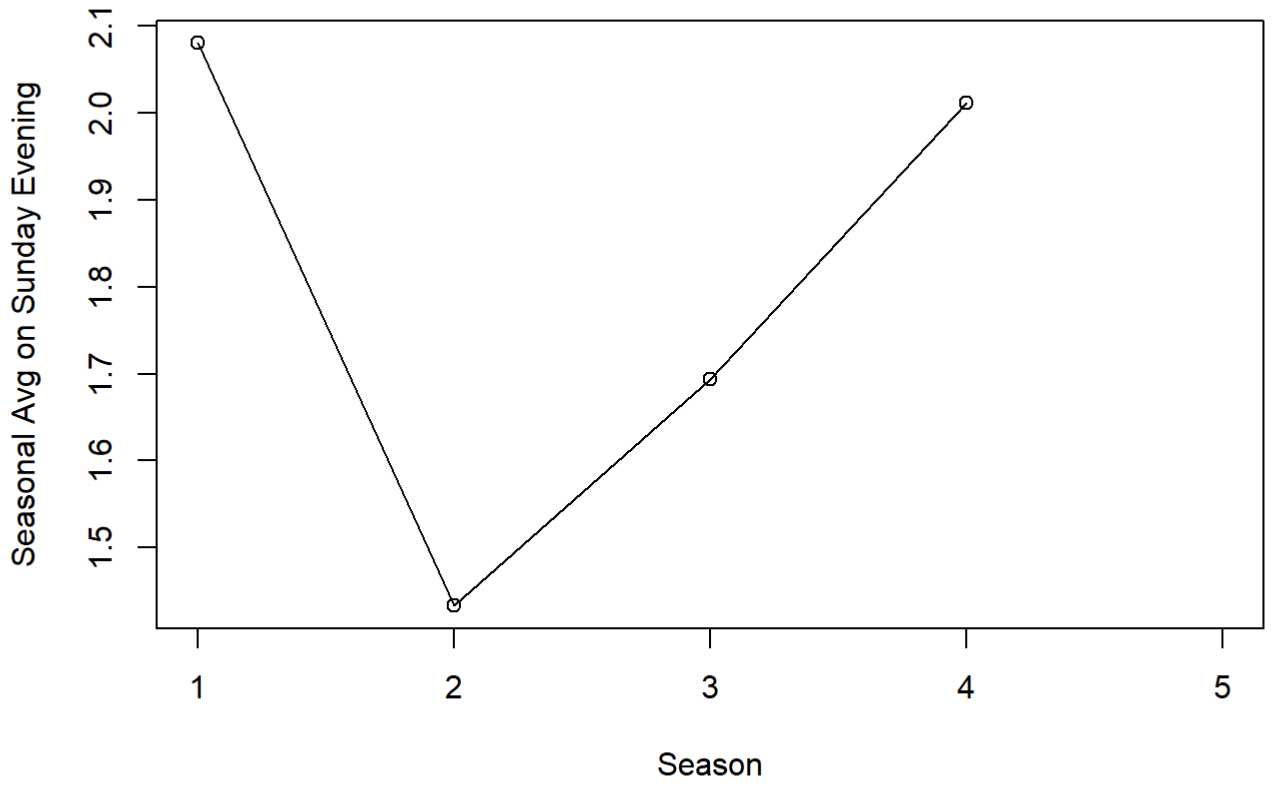
```
plot(xlim=c(1,5),seasoneveningmin, xlab ='Season', ylab = 'Seasonal Min on Sunday Evening', type
 = "o")
```



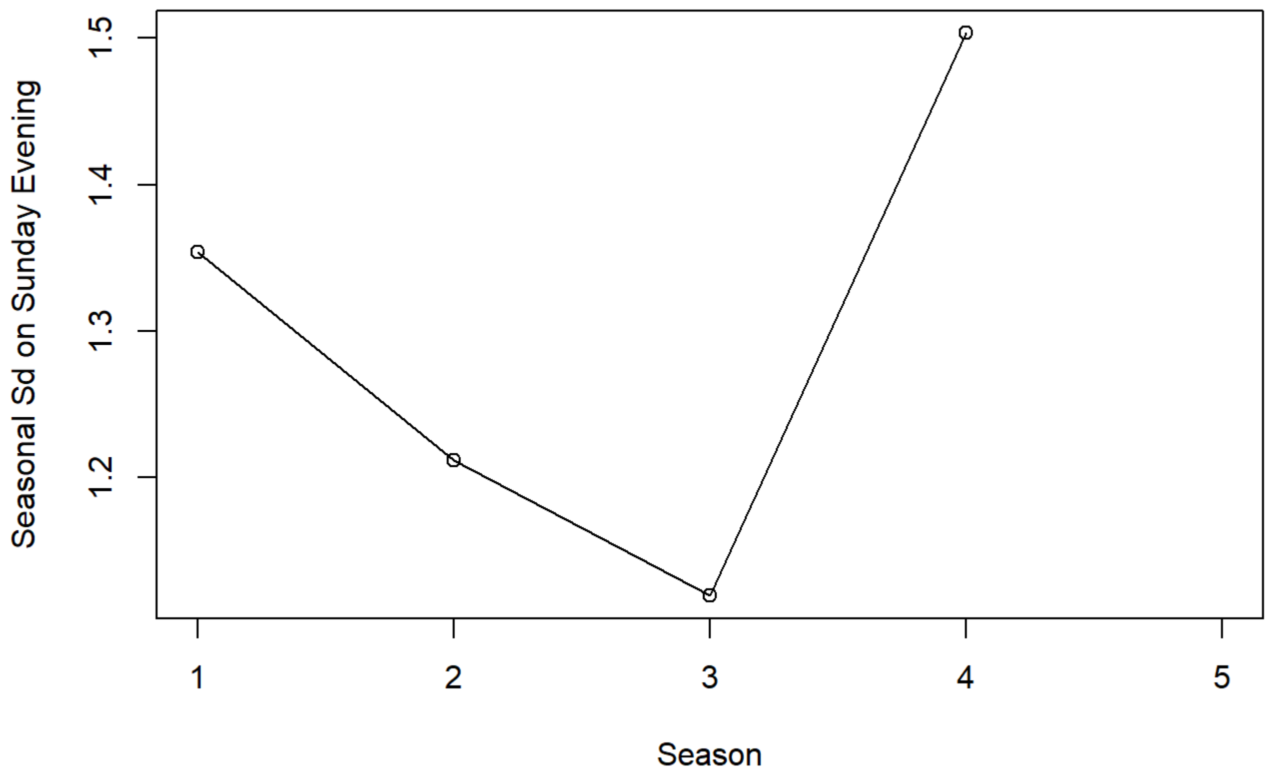
```
plot(xlim=c(1,5),seasoneveningmax, xlab ='Season', ylab = 'Seasonal Max on Sunday Evening', type
 = "o")
```



```
plot(xlim=c(1,5),seasoneveningavg, xlab ='Season', ylab = 'Seasonal Avg on Sunday Evening', type
 = "o")
```



```
plot(xlim=c(1,5),seasoneveningsd,  xlab ='Season', ylab = 'Seasonal Sd on Sunday Evening',  type
 = "o")
```



Explanation: Season 1 - Spring Season 2 - Summer Season 3 - Fall Season 4 - Winter

For seasonal plots, we can see that in Winter the SD in Global Electric Power consumption is high. The electricity deviates largely in Winter. Looking at the plots for mean, minimum, maximum and standard deviation, we have come to the conclusion that electricity consumption is high during Winter and the lowest in the Summer. Looking at a broader picture allows us to see more general trends of electricity consumption.