Reproducible Research Project # 1

Pablo Sainz

1 de junio de 2018

Introduction

This markdown file is part of the 1st project of the Reproducible Research course from John Hopkins Data Science Specialization.

Data Loading

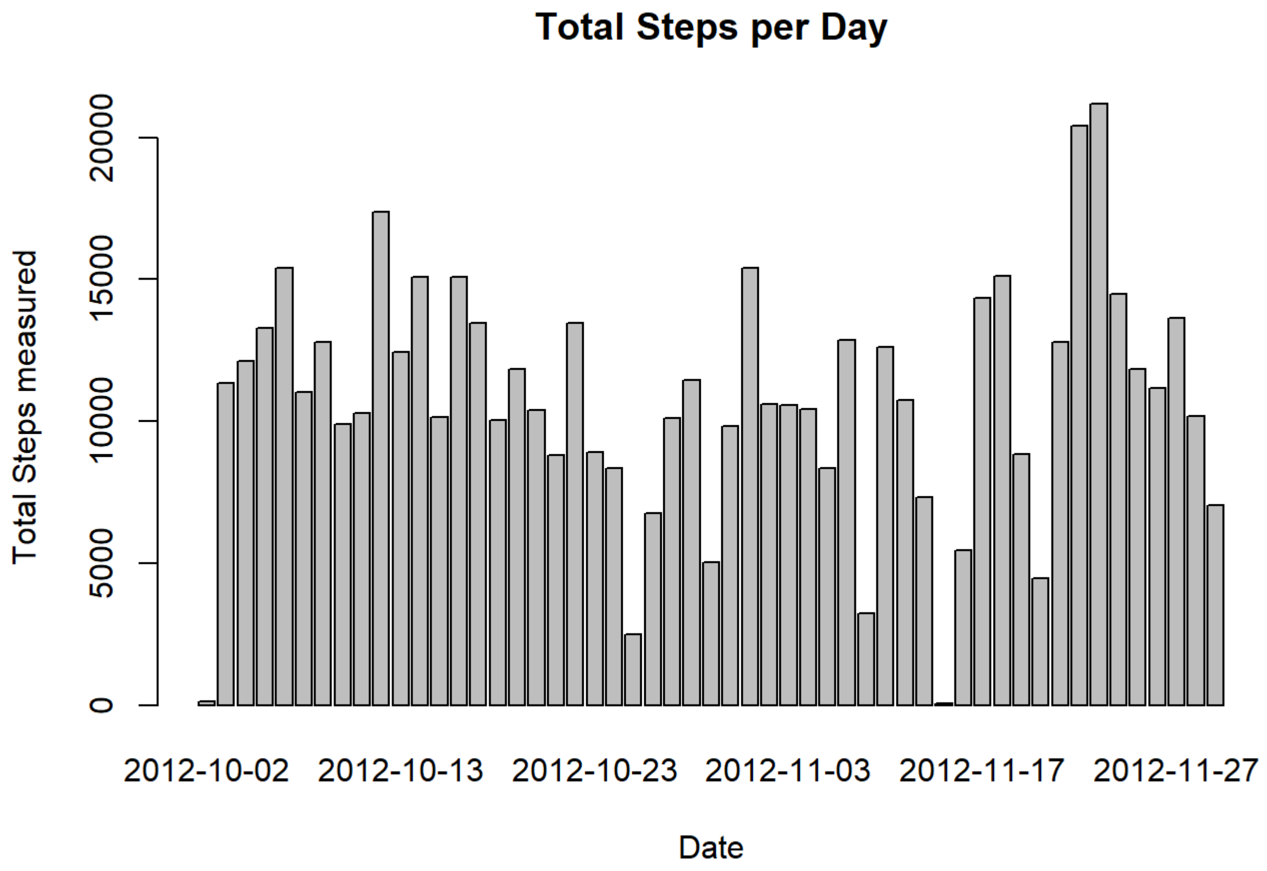
First we load the required data to perform the analysis:

```
rawStepDataSet <- read.csv("C:\\Users\\psainza\\Documents\\Course5Project1\\activity.csv")
filteredStepDataSet <- rawStepDataSet[!is.na(rawStepDataSet$steps),]
```

Steps per Day

The following histogram illustrates the total ammount of steps taken per day:

```
# Obtain the distinct days of the filtered data set
targetDays <- unique(filteredStepDataSet$date)
resultData <- data.frame("Day"=character(),"TotalSteps"=numeric(),stringsAsFactors = FALSE)
colnames(resultData) <- c("Day","Total.Steps")

for (targetDay in targetDays) {
  #Filter the data
  currentDayData <-filteredStepDataSet[filteredStepDataSet$date == targetDay,]
  #Get the current total of emissions on that year
  currentStepsTotal <- sum(currentDayData$steps)
  newRow <- data.frame(targetDay,currentStepsTotal)
  resultData <- rbind(resultData,newRow)
}

barplot(resultData$currentStepsTotal,main = "Total Steps per Day",xlab = "Date",names.arg = uniq
ue(resultData$targetDay),ylab = "Total Steps measured")
```



Indicators of Steps per Day

The mean and median steps per day are shown in the following table:

```r
# Obtain the distinct days of the filtered data set
targetDays <- unique(filteredStepDataSet$date)
resultData2 <- data.frame("Day"=character(),"MeanSteps"=numeric(),"MedianSteps"=numeric(),stringsAsFactors = FALSE)
colnames(resultData2) <- c("Day","Mean.Steps","Median.Steps")

for (targetDay in targetDays) {
  #Filter the data
  currentDayData <-filteredStepDataSet[filteredStepDataSet$date == targetDay,]
  #Get the current total of emissions on that year
  currentStepsMean <- mean(currentDayData$steps)
  currentStepsMedian <- median(currentDayData$steps)
  newRow <- data.frame(targetDay,currentStepsMean,currentStepsMedian)
  resultData2 <- rbind(resultData2,newRow)
}
colnames(resultData2) <- c("Day","Mean.Steps","Median.Steps")
print(resultData2)
```

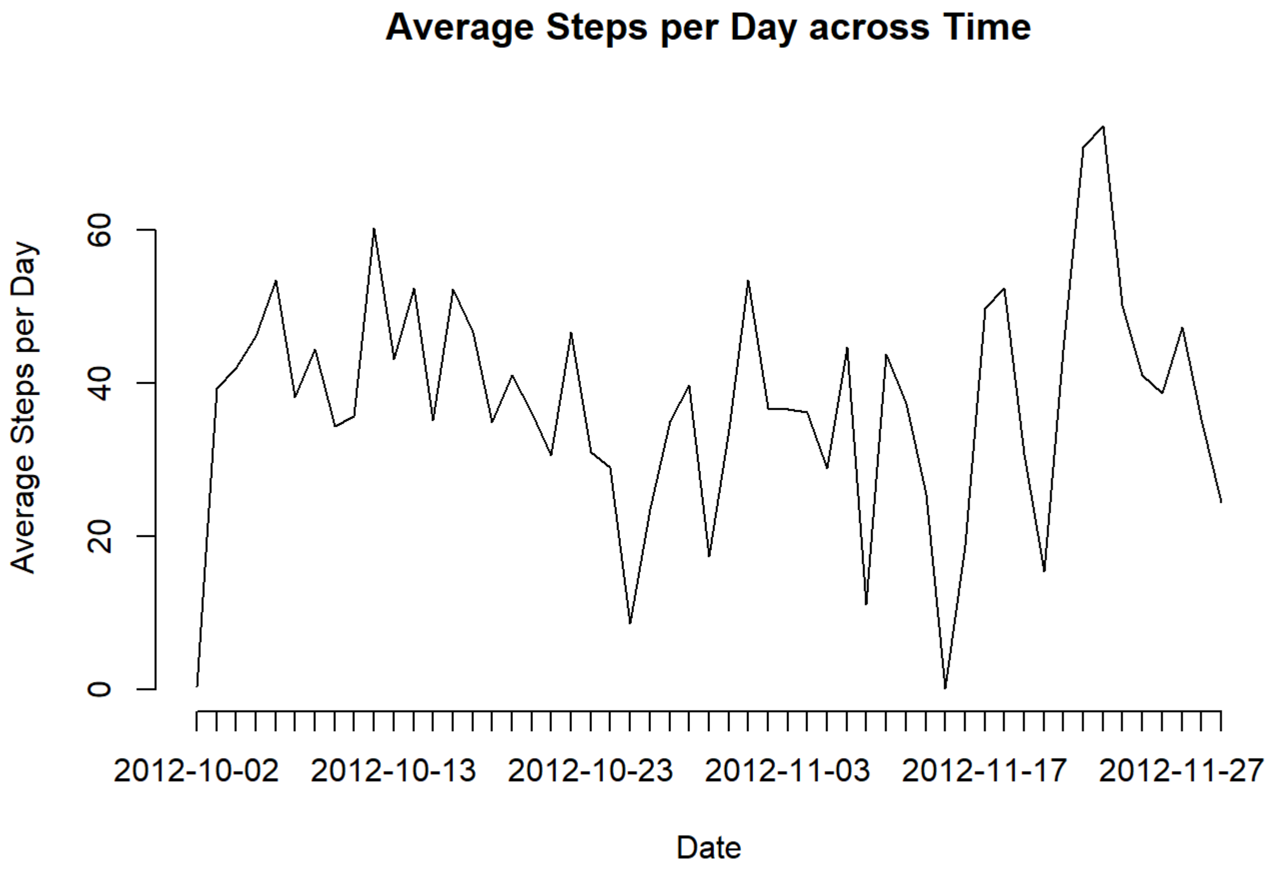
```
##           Day Mean.Steps Median.Steps
## 1  2012-10-02  0.4375000            0
## 2  2012-10-03 39.4166667            0
## 3  2012-10-04 42.0694444            0
## 4  2012-10-05 46.1597222            0
## 5  2012-10-06 53.5416667            0
## 6  2012-10-07 38.2465278            0
## 7  2012-10-09 44.4826389            0
## 8  2012-10-10 34.3750000            0
## 9  2012-10-11 35.7777778            0
## 10 2012-10-12 60.3541667            0
## 11 2012-10-13 43.1458333            0
## 12 2012-10-14 52.4236111            0
## 13 2012-10-15 35.2048611            0
## 14 2012-10-16 52.3750000            0
## 15 2012-10-17 46.7083333            0
## 16 2012-10-18 34.9166667            0
## 17 2012-10-19 41.0729167            0
## 18 2012-10-20 36.0937500            0
## 19 2012-10-21 30.6284722            0
## 20 2012-10-22 46.7361111            0
## 21 2012-10-23 30.9652778            0
## 22 2012-10-24 29.0104167            0
## 23 2012-10-25  8.6527778            0
## 24 2012-10-26 23.5347222            0
## 25 2012-10-27 35.1354167            0
## 26 2012-10-28 39.7847222            0
## 27 2012-10-29 17.4236111            0
## 28 2012-10-30 34.0937500            0
## 29 2012-10-31 53.5208333            0
## 30 2012-11-02 36.8055556            0
## 31 2012-11-03 36.7048611            0
## 32 2012-11-05 36.2465278            0
## 33 2012-11-06 28.9375000            0
## 34 2012-11-07 44.7326389            0
## 35 2012-11-08 11.1770833            0
## 36 2012-11-11 43.7777778            0
## 37 2012-11-12 37.3784722            0
## 38 2012-11-13 25.4722222            0
## 39 2012-11-15  0.1423611            0
## 40 2012-11-16 18.8923611            0
## 41 2012-11-17 49.7881944            0
## 42 2012-11-18 52.4652778            0
## 43 2012-11-19 30.6979167            0
## 44 2012-11-20 15.5277778            0
## 45 2012-11-21 44.3993056            0
## 46 2012-11-22 70.9270833            0
## 47 2012-11-23 73.5902778            0
## 48 2012-11-24 50.2708333            0
## 49 2012-11-25 41.0902778            0
## 50 2012-11-26 38.7569444            0
## 51 2012-11-27 47.3819444            0
## 52 2012-11-28 35.3576389            0
## 53 2012-11-29 24.4687500            0
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Average Steps across Time

The average steps taken per day are shown in the following figure:

```
# Obtain the distinct days of the filtered data set
targetDays <- unique(filteredStepDataSet$date)
resultData3 <- data.frame("Day"=character(),"MeanSteps"=numeric(),stringsAsFactors = FALSE)
colnames(resultData3) <- c("Day","Mean.Steps")

for (targetDay in targetDays) {
  #Filter the data
  currentDayData <-filteredStepDataSet[filteredStepDataSet$date == targetDay,]
  #Get the current total of emissions on that year
  currentStepsMean <- mean(currentDayData$steps,na.rm = TRUE)
  newRow <- data.frame(targetDay,currentStepsMean)
  resultData3 <- rbind(resultData3,newRow)
}
colnames(resultData3) <- c("Day","Mean.Steps")

#print(resultData3)
plot.ts(resultData3$Mean.Steps,main="Average Steps per Day across Time",ylab="Average Steps per
 Day",xlab="Date",axes = F)
axis(2)
axis(1,at=resultData3$Day,labels = resultData3$Day)
```



Time Interval with Maximum Average Steps Taken

The daily interval which in average shows the maximum steps taken in average is the following:

```
# Obtain the distinct days of the filtered data set
targetIntervals <- unique(filteredStepDataSet$interval)
resultData4 <- data.frame("Interval"=numeric(),"Average.Steps"=numeric(),stringsAsFactors = FALS
E)
colnames(resultData4) <- c("Interval","Average.Steps")

for (targetInterval in targetIntervals) {
  #Filter the data
  currentInterval <-filteredStepDataSet[filteredStepDataSet$interval == targetInterval,]
  #Get the current total of emissions on that year
  currentStepsAverage <- mean(currentInterval$steps)
  newRow <- data.frame(targetInterval,currentStepsAverage)
  resultData4 <- rbind(resultData4,newRow)
}
colnames(resultData4) <- c("Interval","Average.Steps")
finalResult = resultData4[max(resultData4$Average.Steps) == resultData4$Average.Steps,]
print(head(finalResult,row.names = FALSE))
```

```
##     Interval Average.Steps
## 104      835      206.1698
```

Replacing Missing Data

One strategy to replace the missing data across the data set. Particularly on the ‘steps’ column would be to replace the ‘NA’ values with zeros, since no safe assumptions can be made in order to interpolate across the time intervals.

Given that strategy, the code to implement such data filling is disclosed as it follows:

```
filledStepDataSet <- read.csv("C:\\Users\\psainza\\Documents\\Course5Project1\\activity.csv")

#Fill the steps column with 0s on missng values
filledStepDataSet[is.na(filledStepDataSet)]<-0
```

Which now displays the missing first day due lack of data captured:

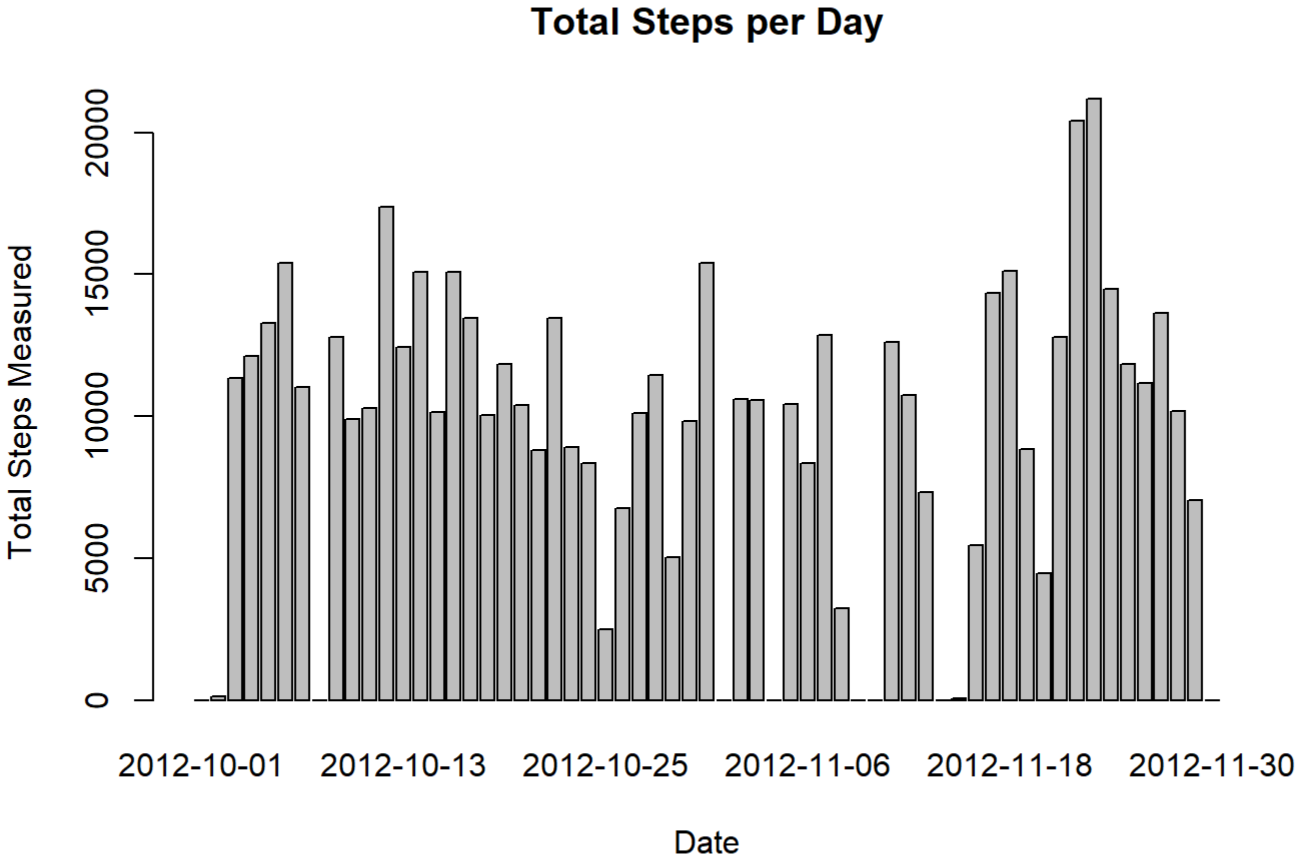
```
head(filledStepDataSet)
```

```
##   steps       date interval
## 1     0 2012-10-01        0
## 2     0 2012-10-01        5
## 3     0 2012-10-01       10
## 4     0 2012-10-01       15
## 5     0 2012-10-01       20
## 6     0 2012-10-01       25
```

Total Steps per Day with Full Data Set

The following figure shows the total steps taken per day now with the full data set considering the previous replacement of NA values:

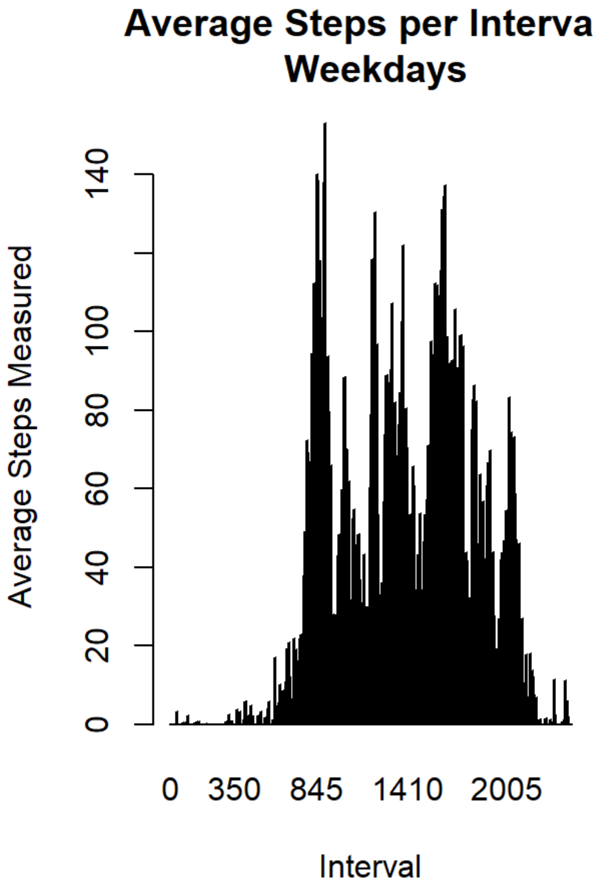
```
# Obtain the distinct days of the filtered data set
targetDays <- unique(filledStepDataSet$date)
resultData5 <- data.frame("Day"=character(),"TotalSteps"=numeric(),stringsAsFactors = FALSE)
colnames(resultData5) <- c("Day","Total.Steps")

for (targetDay in targetDays) {
  #Filter the data
  currentDayData <-filledStepDataSet[filledStepDataSet$date == targetDay,]
  #Get the current total of emissions on that year
  currentStepsTotal <- sum(currentDayData$steps)
  newRow <- data.frame(targetDay,currentStepsTotal)
  resultData5 <- rbind(resultData5,newRow)
}

barplot(resultData5$currentStepsTotal,main = "Total Steps per Day",xlab = "Date",names.arg = uni
que(resultData5$targetDay),ylab = "Total Steps Measured")
```



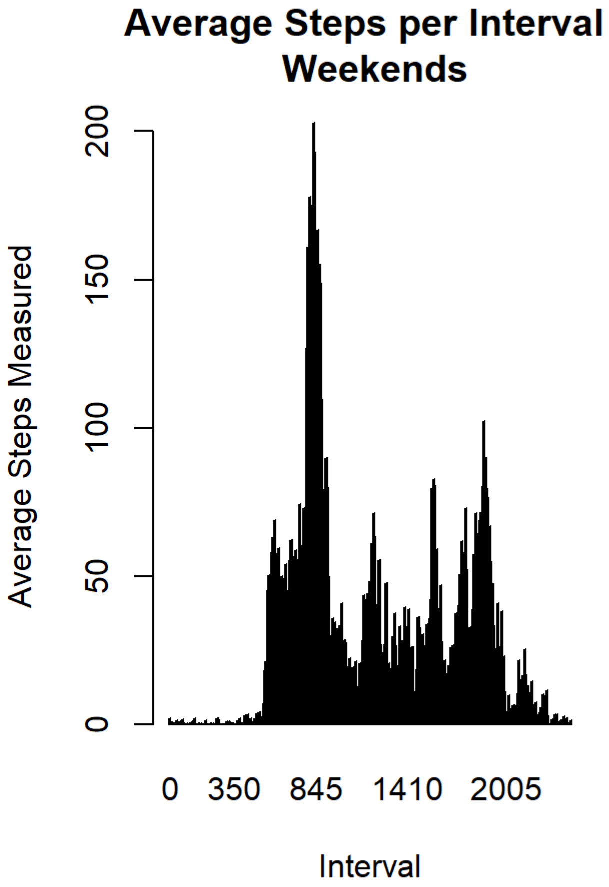
Average Steps Comparison Weekdays vs Weekends

The following plots show a comparison of the average steps taken per interval on weekdays vs weekend days:

```
# Load the required library
library(chron)
targetIntervals <- unique(filledStepDataSet$interval)
resultDataWeekdays <- data.frame("Interval"=character(),"AverageSteps"=numeric(),stringsAsFactor
s = FALSE)
resultDataWeekends <- data.frame("Interval"=character(),"AverageSteps"=numeric(),stringsAsFactor
s = FALSE)


for (targetInterval in targetIntervals) {
  #Filter the data
  currentIntervalData <-filledStepDataSet[filledStepDataSet$interval == targetInterval,]
  #Get the data from weekdays and weekends
  currentWeekdaysData <- currentIntervalData[is.weekend(currentIntervalData$date),]
  currentWeekendsData <- currentIntervalData[!is.weekend(currentIntervalData$date),]
  #Get the averages for each result data set
  currentWeekdaysAverage <- mean(currentWeekdaysData$steps)
  currentWeekendsAverage <- mean(currentWeekendsData$steps)

  newWeekdayRow <- data.frame(targetInterval,currentWeekdaysAverage)
  newWeekendRow <- data.frame(targetInterval,currentWeekendsAverage)

  #Append the result rows
  resultDataWeekdays <- rbind(resultDataWeekdays,newWeekdayRow)
  resultDataWeekends <- rbind(resultDataWeekends,newWeekendRow)
}

colnames(resultDataWeekdays) <- c("Interval","Average.Steps")
colnames(resultDataWeekends) <- c("Interval","Average.Steps")
# Construct the plot
par(bg="white",mfrow=c(1,2))
barplot(resultDataWeekdays$Average.Steps,main = "Average Steps per Interval \n Weekdays",xlab =
"Interval",names.arg = unique(resultDataWeekdays$Interval),ylab = "Average Steps Measured")
barplot(resultDataWeekends$Average.Steps,main = "Average Steps per Interval \n Weekends",xlab =
"Interval",names.arg = unique(resultDataWeekends$Interval),ylab = "Average Steps Measured")
```



Average Steps per Interval
Weekdays



Average Steps per Interval
Weekends