Reproducible Research Course Project 2

Tochi Okeke

December 8, 2018

This project will analyze data from the Activity monitoring dataset and create a report using R Markdown and tidyr.

First, I will set my working directory and download the data using the data.table package

```
library(data.table)#for subsetting
library(dplyr) #for data cleaning
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(ggplot2) #for visualization
data <- fread("activity.csv")
data
```

```
##        steps       date interval
##     1:    NA 2012-10-01        0
##     2:    NA 2012-10-01        5
##     3:    NA 2012-10-01       10
##     4:    NA 2012-10-01       15
##     5:    NA 2012-10-01       20
##    ---
## 17564:    NA 2012-11-30     2335
## 17565:    NA 2012-11-30     2340
## 17566:    NA 2012-11-30     2345
## 17567:    NA 2012-11-30     2350
## 17568:    NA 2012-11-30     2355
```

Taking a quick look at the data with summary

```
summary(data)
```

```
##      steps            date              interval
##  Min.   :  0.00   Length:17568       Min.   :   0.0
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8
##  Median :  0.00   Mode  :character   Median :1177.5
##  Mean   : 37.38                      Mean   :1177.5
##  3rd Qu.: 12.00                      3rd Qu.:1766.2
##  Max.   :806.00                      Max.   :2355.0
##  NA's   :2304
```

Checking the class of the 'date' column

```
class(data$date)
```

```
## [1] "character"
```

Downloading the lubridate library to convert the date column into 'date' class

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year

## The following object is masked from 'package:base':
##
##     date
```

```
data$date <- ymd(data$date)
print(class(data$date))
```

```
## [1] "Date"
```

What is the mean total number of steps taken per day?

Calculate the total number of steps per day.

Using the data.table package, I will aggregate the data by date and find the total steps per day.

```
totalsteps <- data[,sum(steps),date]
totalsteps
```

```
##           date    V1
##  1: 2012-10-01    NA
##  2: 2012-10-02   126
##  3: 2012-10-03 11352
##  4: 2012-10-04 12116
##  5: 2012-10-05 13294
##  6: 2012-10-06 15420
##  7: 2012-10-07 11015
##  8: 2012-10-08    NA
##  9: 2012-10-09 12811
## 10: 2012-10-10  9900
## 11: 2012-10-11 10304
## 12: 2012-10-12 17382
## 13: 2012-10-13 12426
## 14: 2012-10-14 15098
## 15: 2012-10-15 10139
## 16: 2012-10-16 15084
## 17: 2012-10-17 13452
## 18: 2012-10-18 10056
## 19: 2012-10-19 11829
## 20: 2012-10-20 10395
## 21: 2012-10-21  8821
## 22: 2012-10-22 13460
## 23: 2012-10-23  8918
## 24: 2012-10-24  8355
## 25: 2012-10-25  2492
## 26: 2012-10-26  6778
## 27: 2012-10-27 10119
## 28: 2012-10-28 11458
## 29: 2012-10-29  5018
## 30: 2012-10-30  9819
## 31: 2012-10-31 15414
## 32: 2012-11-01    NA
## 33: 2012-11-02 10600
## 34: 2012-11-03 10571
## 35: 2012-11-04    NA
## 36: 2012-11-05 10439
## 37: 2012-11-06  8334
## 38: 2012-11-07 12883
## 39: 2012-11-08  3219
## 40: 2012-11-09    NA
## 41: 2012-11-10    NA
## 42: 2012-11-11 12608
## 43: 2012-11-12 10765
## 44: 2012-11-13  7336
## 45: 2012-11-14    NA
## 46: 2012-11-15    41
## 47: 2012-11-16  5441
## 48: 2012-11-17 14339
## 49: 2012-11-18 15110
## 50: 2012-11-19  8841
## 51: 2012-11-20  4472
## 52: 2012-11-21 12787
## 53: 2012-11-22 20427
## 54: 2012-11-23 21194
## 55: 2012-11-24 14478
## 56: 2012-11-25 11834
## 57: 2012-11-26 11162
## 58: 2012-11-27 13646
## 59: 2012-11-28 10183
## 60: 2012-11-29  7047
## 61: 2012-11-30    NA
##           date    V1
```

The mean is

```
mean(totalsteps$V1,na.rm=TRUE)
```

```
## [1] 10766.19
```

The median is

```
median(totalsteps$V1,na.rm=TRUE)
```

```
## [1] 10765
```

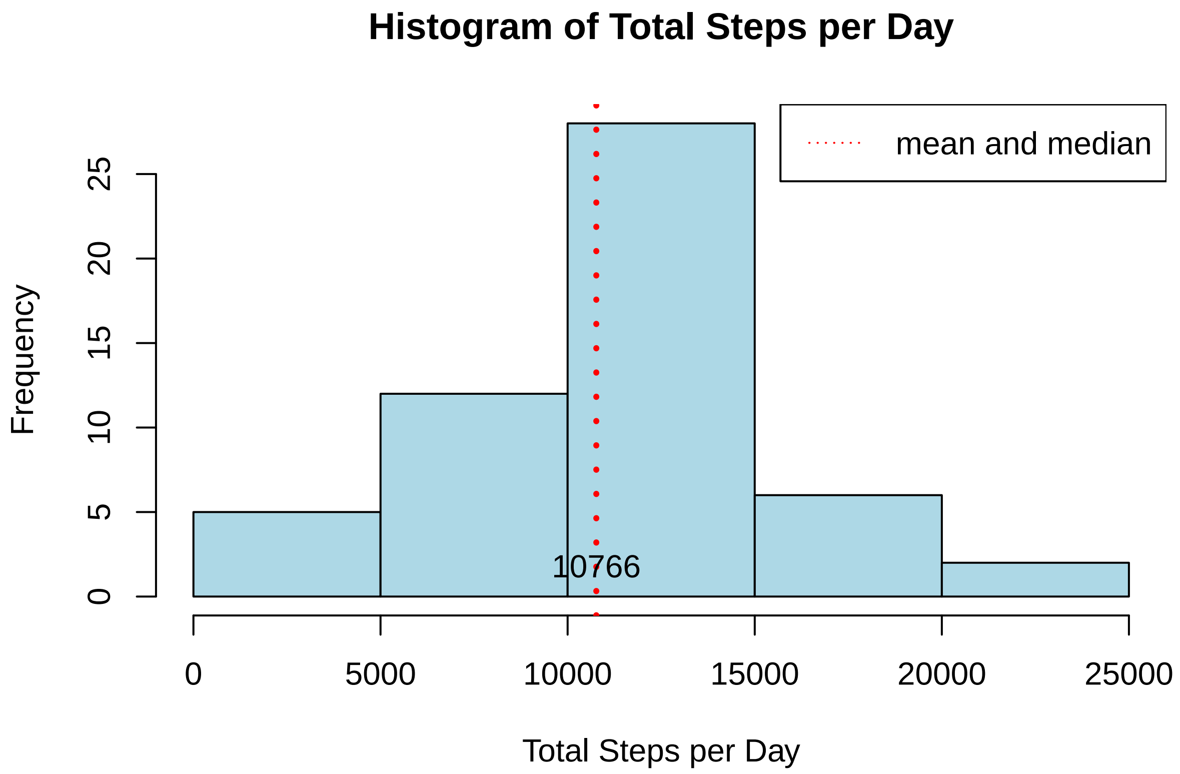
```
#Creating the histogram and adding labels
 hist(totalsteps$V1,xlab="Total Steps per Day",main="Histogram of Total Steps per Day",col="lightblue")

#Adding a vertical line with to indicate the mean
```

```
abline(v=c(mean(totalsteps$V1,na.rm=TRUE)), col=c("red"),lwd=3,lty=3)

text(x=mean(totalsteps$V1,na.rm=TRUE),y=0,"10766",pos=3)

#Adding the legend for ease of interpretation
legend("topright","mean and median",col="red",lty=3)
```



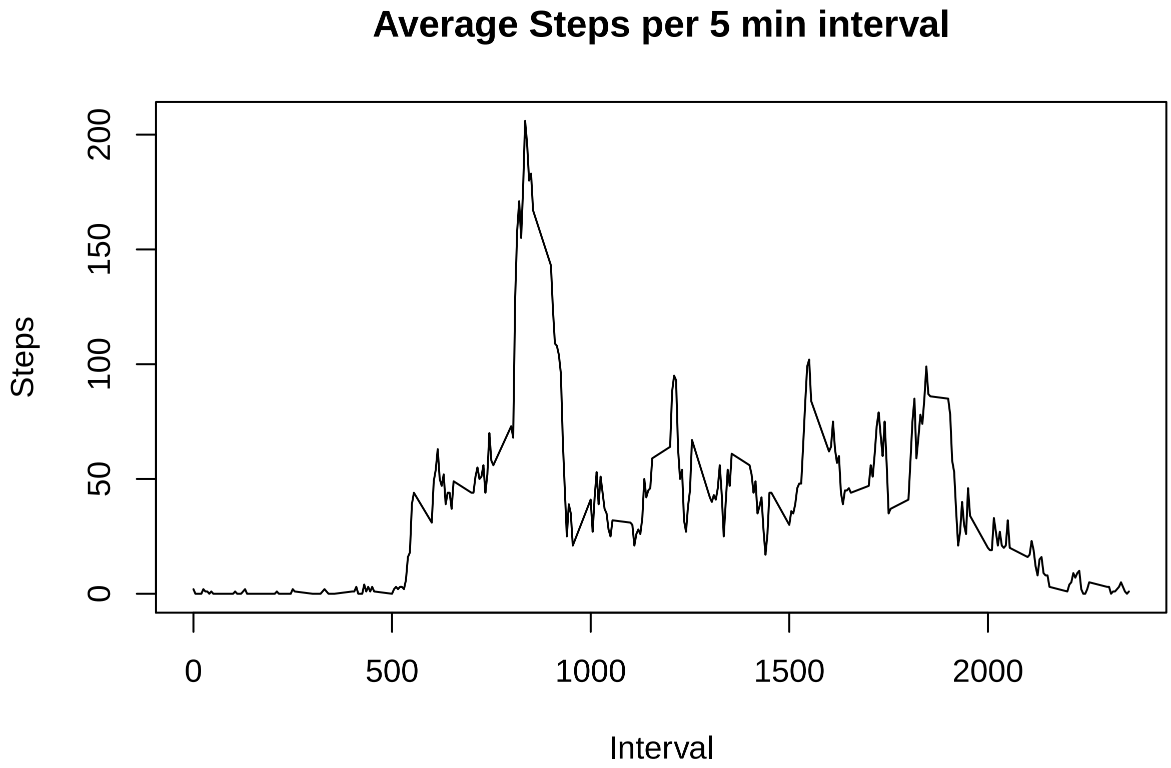
#### What is the average daily activity patternn?

I will use the data.table package to get the average steps per interval.

```
avgsteps <- data[,round(mean(steps,na.rm=TRUE)),interval]
avgsteps
```

```
##      interval V1
##   1:        0  2
##   2:        5  0
##   3:       10  0
##   4:       15  0
##   5:       20  0
##  ---
## 284:     2335  5
## 285:     2340  3
## 286:     2345  1
## 287:     2350  0
## 288:     2355  1
```

```r
plot(x = avgsteps$interval,y=avgsteps$V1,type='l', xlab="Interval",ylab="Steps",main="Average Steps per
```

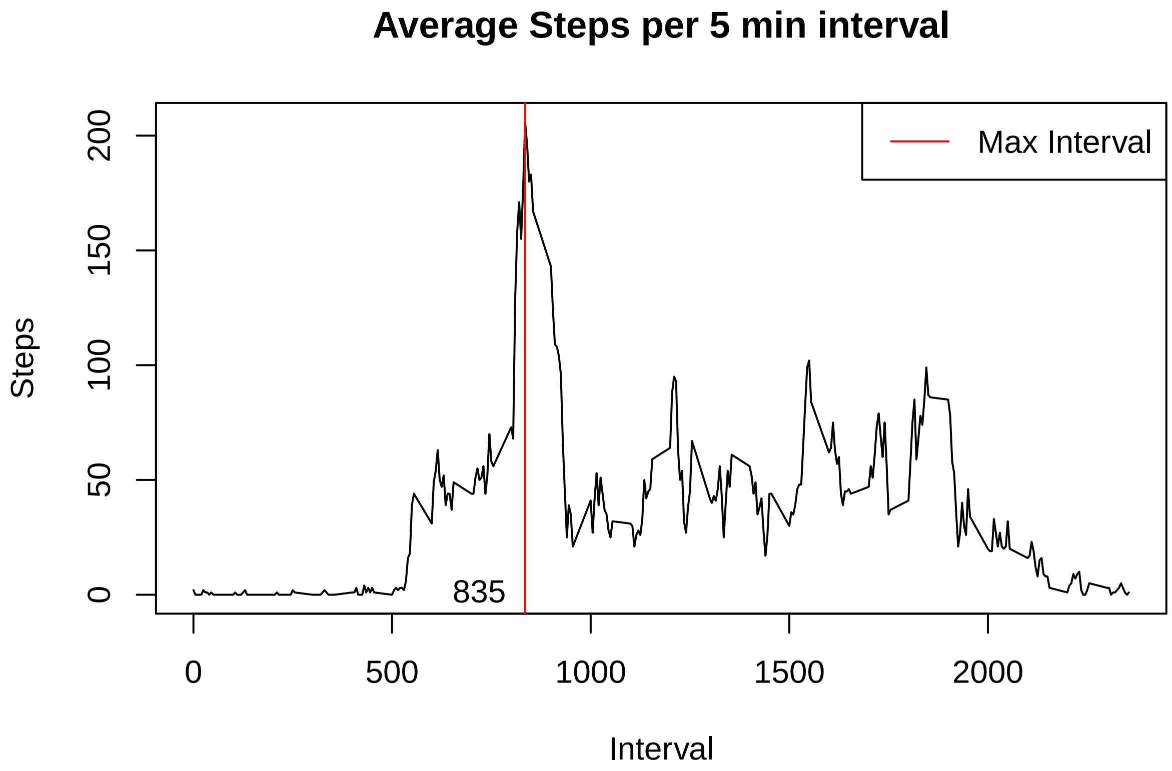


Using a vertical line at max(steps), we can find the 5 minute interval that contains the maximum number of steps

```r
#subset the avgsteps dataset to get the maximum interval
max_interval <- avgsteps[which(avgsteps$V1==max(avgsteps$V1))]
max_interval
```

```
##    interval  V1
## 1:      835 206
```

Plotting the vertical line to show that the interval with the maximum number of steps is 835.



Imputing Missing Values

Total missing values in the dataset

```
#Total missing values in the dataset
sum(is.na(data))
```

```
## [1] 2304
```

```
#Total missing values in the interval column
(sum(is.na(data$interval)))
```

```
## [1] 0
```

```
#Total missing values in the steps column
print(sum(is.na(data$steps)))
```

```
## [1] 2304
```

To impute the values in the steps column I will fill in the missing values using the average steps for that interval.

I will use the avgsteps table as a lookup table. NA values in the ‘steps’ column of ‘data’ are replaced with the average steps for its corresponding interval.

```
#Using dplyr, add the replace_steps column with filled in values
#If the value in the steps column is missing, use the match function to find its appropriate average st
#Pass the new dataset to the mutate function using the pipe operator
#Drop the old steps column
no_na <- mutate(data,replace_steps = ifelse(is.na(steps),avgsteps$V1[match(interval,avgsteps$interval)]
#Compare the two datasets (imputed data(no_na) vs missing data(data))
print(head(no_na,5))
```

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

```
print(head(data,5))
```

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

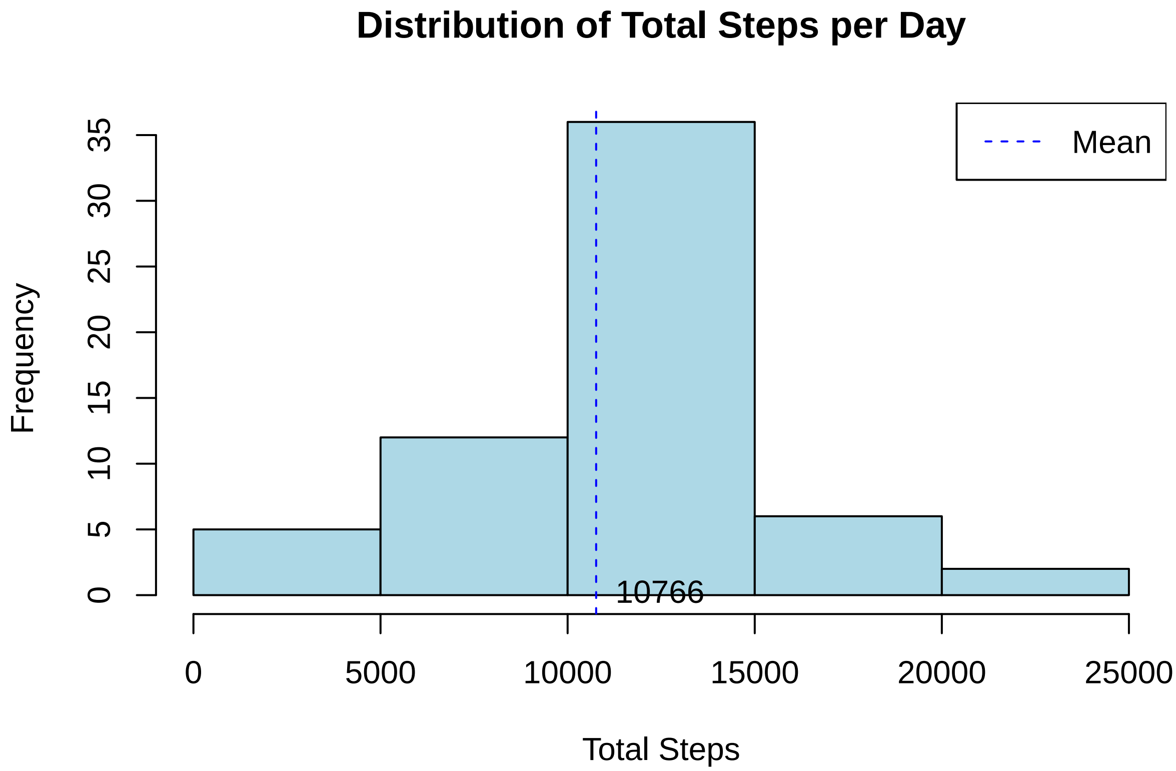
Make a histogram of the total number of steps taken each day

```
no_na <- as.data.table(no_na) #Converting the no_na from a data.frame object to a data.table for ease o

plot_data <- no_na[,sum(replace_steps),date] # Select all rows and sum the replace_steps column while g

hist(plot_data$V1,xlab="Total Steps",col="lightblue", main="Distribution of Total Steps per Day") #Crea
abline(v=c(mean(plot_data$V1,mean(totalsteps$V1,na.rm=TRUE))),col=c("blue","red"),lty=2)
text(x=mean(plot_data$V1),y=0,"10766",pos=4)

legend("topright","Mean",col="blue",lty=2)# Adding the legend for ease of interpretation
```



```
print(mean(plot_data$V1,na.rm = TRUE))
```

```
## [1] 10765.64
```

```
print(median(plot_data$V1))
```

```
## [1] 10762
```

There is little to no difference between the means of the data with and without na values.

```
mean(plot_data$V1) #Imputed mean
```

```
## [1] 10765.64
```

```
mean(totalsteps$V1,na.rm=TRUE) #Mean without na
```

```
## [1] 10766.19
```

Are there differences in activity patterns between weekdays and weekends?

```
days <- c("Monday","Tuesday","Wednesday","Thursday","Friday") #Create a list of weekdays
no_na <- mutate(no_na,type_of_day = ifelse(weekdays(no_na$date) %in% days,"weekday","weekend")) #Check
head(no_na,5)
```

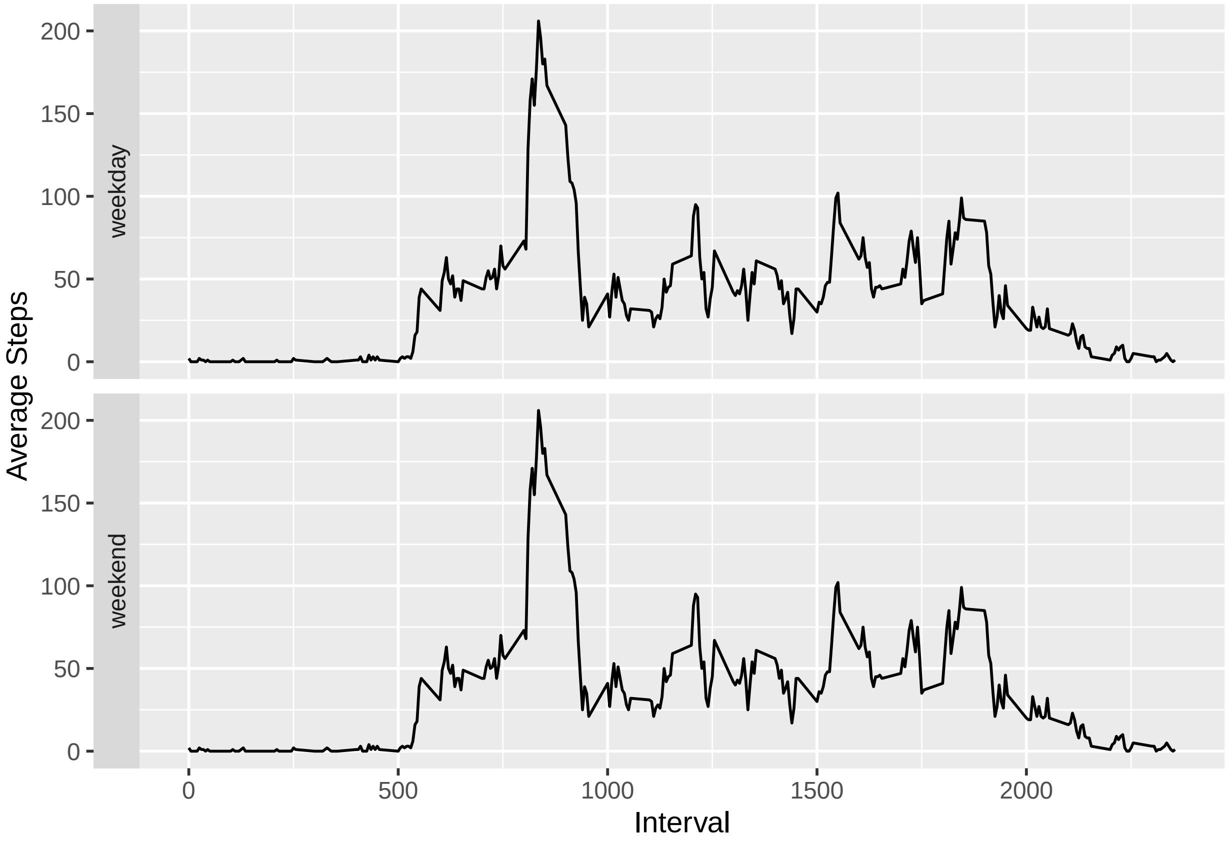
```
##         date interval replace_steps type_of_day
## 1 2012-10-01        0             2     weekday
## 2 2012-10-01        5             0     weekday
## 3 2012-10-01       10             0     weekday
## 4 2012-10-01       15             0     weekday
## 5 2012-10-01       20             0     weekday
```

```
no_na <- as.data.table(no_na)
plotting_data <- no_na[,.(date,type_of_day,round(mean(replace_steps))),.(interval)] #Subset no_na by se
head(plotting_data,5)
```

```
##    interval       date type_of_day V3
## 1:        0 2012-10-01     weekday  2
## 2:        0 2012-10-02     weekday  2
## 3:        0 2012-10-03     weekday  2
## 4:        0 2012-10-04     weekday  2
## 5:        0 2012-10-05     weekday  2
```

Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
p <- plotting_data %>% ggplot(aes(interval,V3))
p + geom_line() + facet_grid(type_of_day~.,switch = "y") + ylab("Average Steps") + xlab("Interval")
```


Average Steps
weekday
weekend
Interval