Coursera: Data Science: Exploratory Data Analysis: Project 1

John W. Tiede

09/03/2014

Introduction

This assignment uses data from the UC Irvine Machine Learning Repository, a popular repository for machine learning datasets. In particular, we will be using the "Individual household electric power consumption Data Set" which I have made available on the course web site:

```
Dataset: Electric power consumption [20Mb]

Description: Measurements of electric power consumption in one household with a one-minute
sampling rate over a period of almost 4 years. Different electrical quantities and some
sub-metering values are available.
```

The following descriptions of the 9 variables in the dataset are taken from the UCI web site:

1. Date: Date in format dd/mm/yyyy
2. Time: time in format hh:mm:ss
3. Global_active_power: household global minute-averaged active power (in kilowatt)
4. Global_reactive_power: household global minute-averaged reactive power (in kilowatt)
5. Voltage: minute-averaged voltage (in volt)
6. Global_intensity: household global minute-averaged current intensity (in ampere)
7. Sub_metering_1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).
8. Sub_metering_2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.
9. Sub_metering_3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

Loading the data

When loading the dataset into R, please consider the following:

The dataset has 2,075,259 rows and 9 columns. First calculate a rough estimate of how much memory the dataset will require in memory before reading into R. Make sure your computer has enough memory (most modern computers should be fine).

```
## [1] "System RAM                 : 4096.00 MBi"
```

```
## [1] "Data Set Memory Requirements:  142.50 MBi"
```

We will only be using data from the dates 2007-02-01 and 2007-02-02. One alternative is to read the data from just those dates rather than reading in the entire dataset and subsetting to those dates.

Looking at the data file, there is a single header line, and the data is separated by the ';' character.

You may find it useful to convert the Date and Time variables to Date/Time classes in R using the strptime() and as.Date() functions.

Note that in this dataset missing values are coded as ?.

```
# Reference:
#  http://stackoverflow.com/questions/5595512/what-is-the-difference-between-require-and-library
suppressPackageStartupMessages(library(xts))

# Read in data:
filename <- "household_power_consumption.txt"
#some.data <- read.table(file=filename, header=TRUE, sep=";",
#                        stringsAsFactor=FALSE, na.strings="?", nrows=10)
#data.classes <- sapply(some.data, class)
#all.data <- read.table(file=filename, header=TRUE, sep=";",
#                       stringsAsFactor=FALSE, na.strings="?",
#                       colClasses=data.classes)
all.data <- read.table(file=filename, header=TRUE, sep=";",
                       stringsAsFactor=FALSE, na.strings="?")
#dim(all.data)

# Create 'zoo' object:
timestamps <- as.POSIXct(paste(all.data$Date, all.data$Time),
                         format="%d/%m/%Y %H:%M:%OS")
HPC <- xts(all.data[, 3:9], timestamps)
#class(HPC)
#colnames(HPC)
#head(HPC)

# Subset for the days '2007-02-01' & '2007-02-02':
t.start <- "2007-02-01"
t.end   <- "2007-02-02"
t.range <- paste0(t.start, "::", t.end)
HPC.sub <- HPC[t.range]
#dim(HPC.sub)
#head(HPC.sub)
```

Making Plots

Our overall goal here is simply to examine how household energy usage varies over a 2-day period in February, 2007. Your task is to reconstruct the following plots below, all of which were constructed using the base plotting system.

First you will need to fork and clone the following GitHub repository: https://github.com/rdpeng/ExData_Plotting1

For each plot you should

- Construct the plot and save it to a PNG file with a width of 480 pixels and a height of 480 pixels.
- Name each of the plot files as plot1.png, plot2.png, etc.
- Create a separate R code file (plot1.R, plot2.R, etc.) that constructs the corresponding plot, i.e. code in plot1.R constructs the plot1.png plot. Your code file should include code for reading the data so that the plot can be fully reproduced. You should also include the code that creates the PNG file.

- Add the PNG file and R code file to your git repository

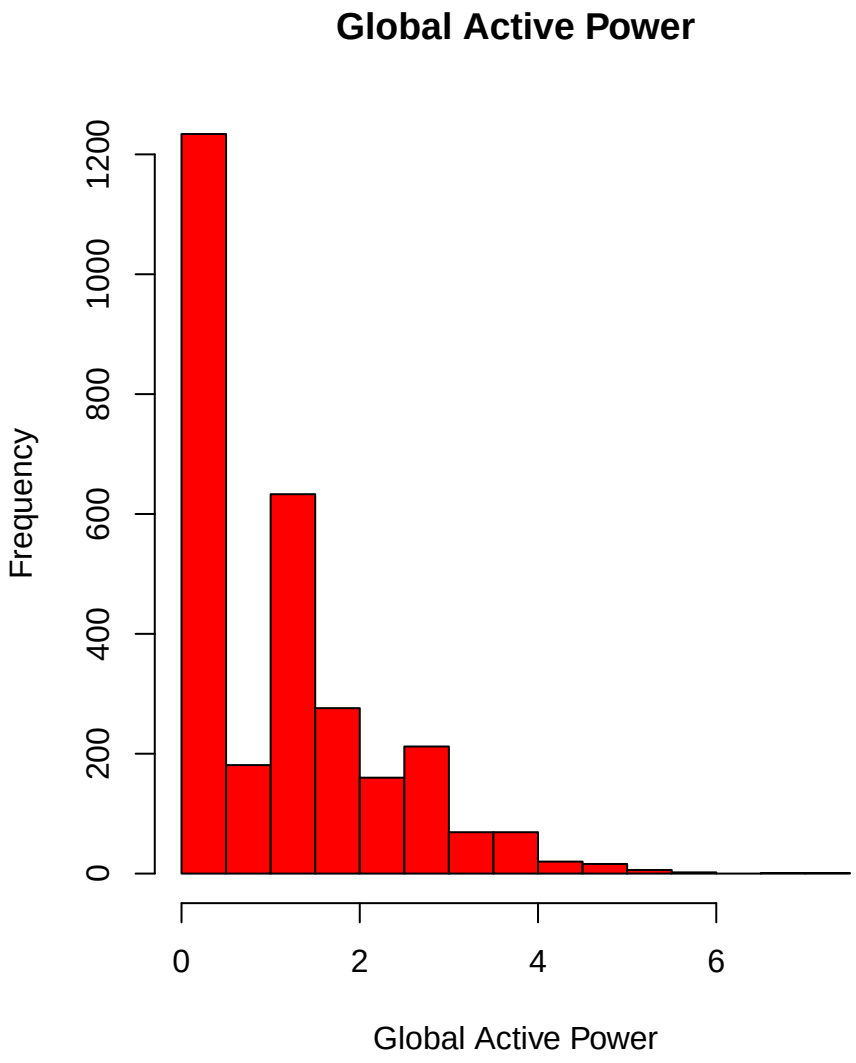
When you are finished with the assignment, push your git repository to GitHub so that the GitHub version of your repository is up to date. There should be four PNG files and four R code files.

The four plots that you will need to construct are shown below.

I will create each in turn.

Plot 1

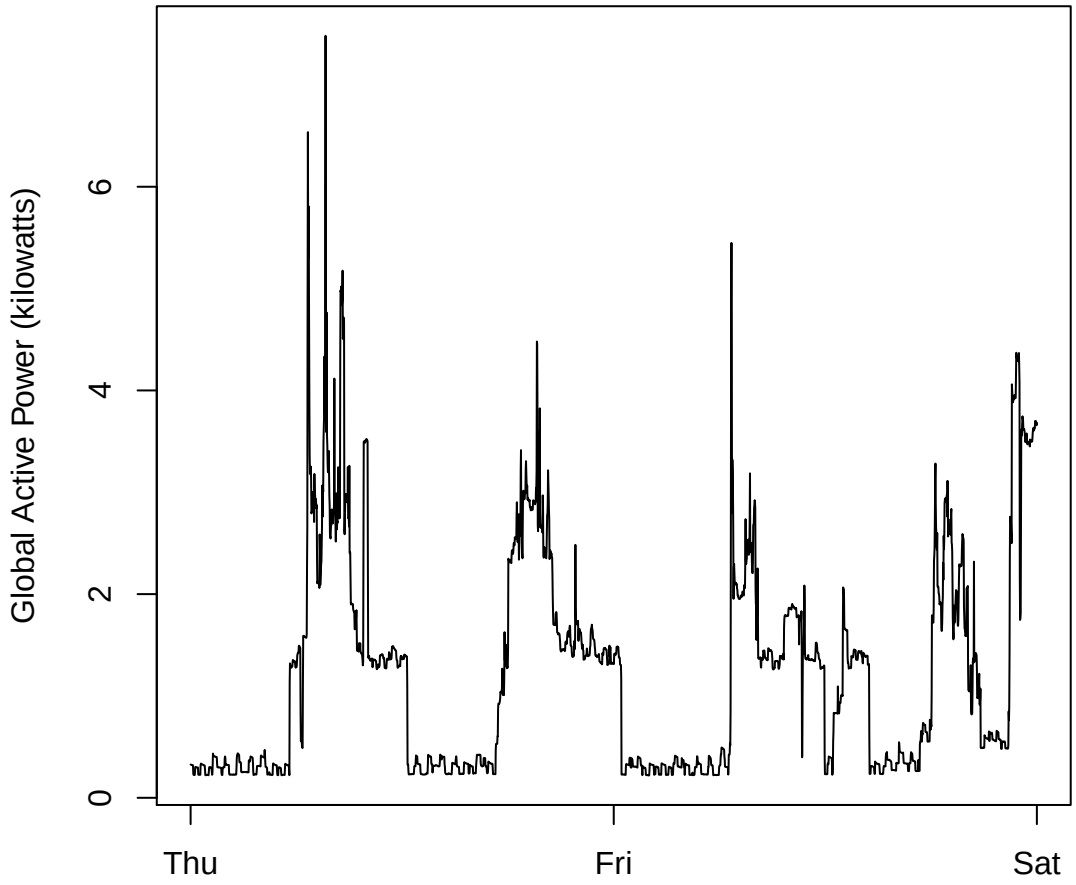
```
if (TRUE) {
  # Create the graph in the 'base' package:
  hist(HPC.sub$Global_active_power,
       main="Global Active Power", xlab="Global Active Power",
       col="Red", border="Black")
} else {
  # Create the 'png' file:
  png(filename="plot1.png")
  hist(HPC.sub$Global_active_power,
       main="Global Active Power", xlab="Global Active Power",
       col="Red", border="Black")
  dev.off()
}
```



Plot 2

```
times <- time(HPC.sub)
ticks <- seq(times[1], times[length(times)], length.out=3)
day.names <- round(seq(times[1], times[length(times)], length.out=3), "hours")
fmt <- "%a"  # format for axis labels ("Day Names")
labs <- format(day.names, fmt)

if (TRUE) {
  # Create the graph in the 'base' package:
  plot.xts(HPC.sub$Global_active_power, auto.grid=FALSE, xaxt="n",
           ylab="Global Active Power (kilowatts)", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)
} else {
  # Create the 'png' file:
  png(filename="plot2.png")
```

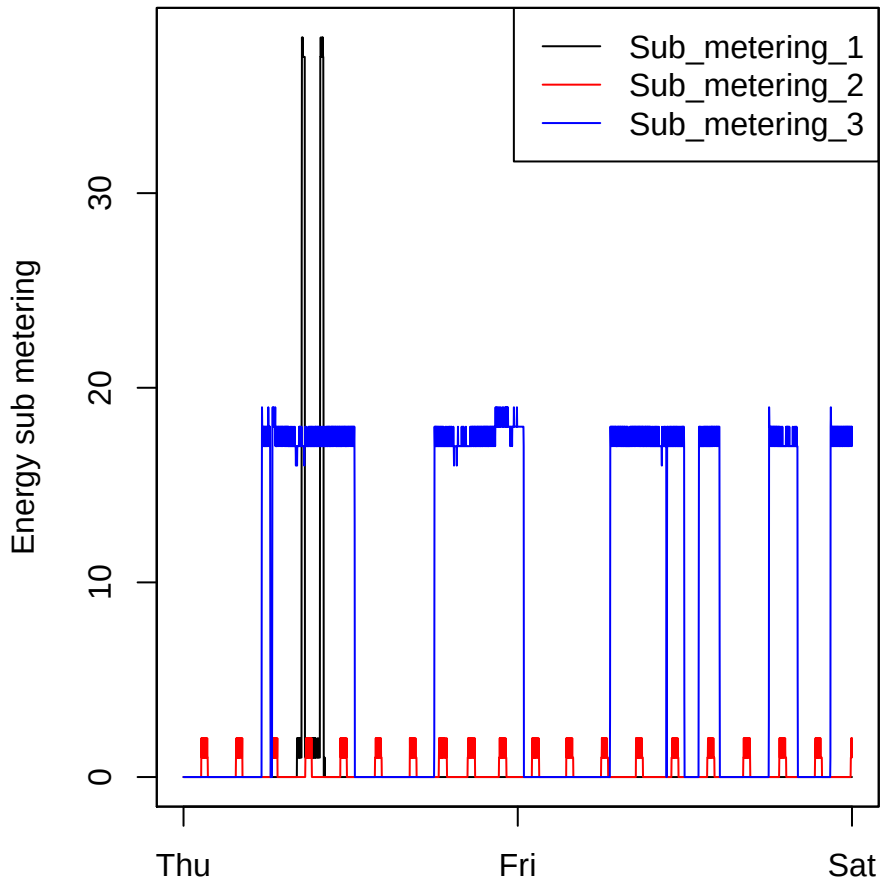
```
  plot.xts(HPC.sub$Global_active_power, auto.grid=FALSE, xaxt="n",
           ylab="Global Active Power (kilowatts)", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)
  dev.off()
}
```



Plot 3

```
# Reference:
# http://blog.revolutionanalytics.com/2014/01/quantitative-finance-applications-in-r-plotting-xts-time-
HPC.sub.zoo <- as.zoo(HPC.sub[, 5:7])
#class(HPC.sub.zoo)
#is.zoo(HPC.sub.zoo)
#dim(HPC.sub.zoo)
#head(HPC.sub.zoo)
my.colors = c("Black", "Red", "Blue")

if (TRUE) {
  # Create the graph in the 'base' package:
  plot(HPC.sub.zoo, xaxt="n", screens=1,
       xlab="", ylab="Energy sub metering", main="", col=my.colors)
```

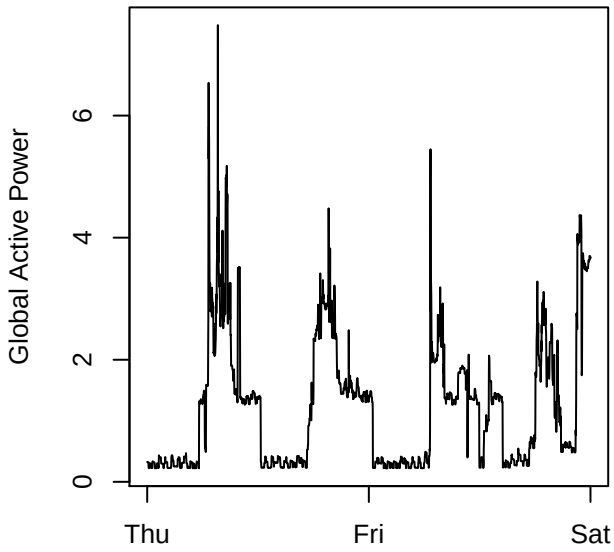
```
  # Use x-axis calculation from the 'Plot 2' code chunk.
  axis(1, at=ticks, labels=labs, tcl=-0.5)
  legend("topright", col=my.colors, lty=1, legend=colnames(HPC.sub.zoo))
} else {
  # Create the 'png' file:
  png(filename="plot3.png")
  plot(HPC.sub.zoo, xaxt="n", screens=1,
       xlab="", ylab="Energy sub metering", main="", col=my.colors)
  # Use x-axis calculation from the 'Plot 2' code chunk.
  axis(1, at=ticks, labels=labs, tcl=-0.5)
  legend("topright", col=my.colors, lty=1, legend=colnames(HPC.sub.zoo))
  dev.off()
}
```

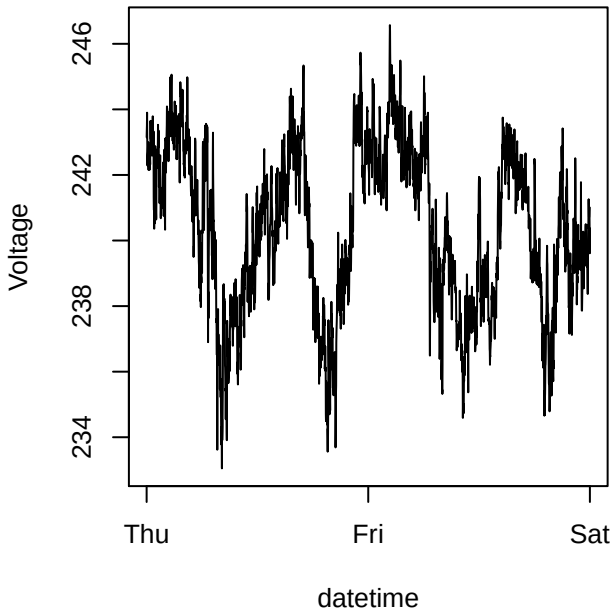


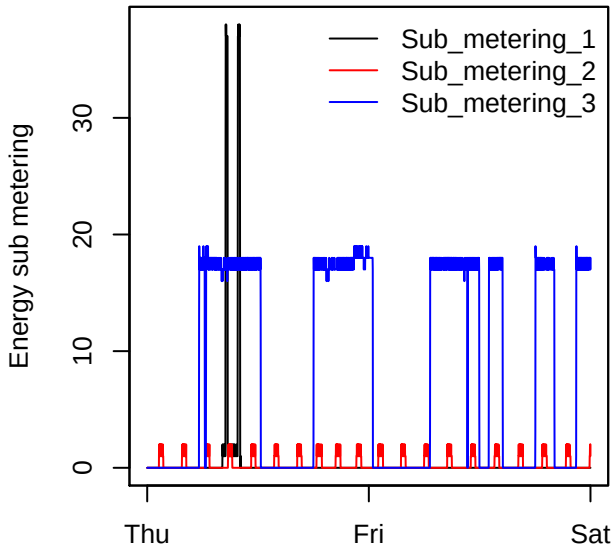
Plot 4

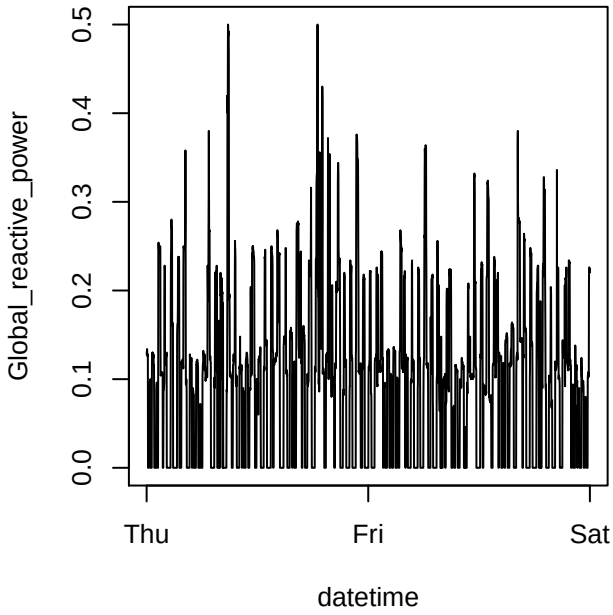
```
if (TRUE) {
  # Create the graph in the 'base' package:
  par(mfrow=c(2,2), mar=c(4,4,1,1), oma=c(1,1,1,1))

  # Plot 1:
```

```
  plot.xts(HPC.sub$Global_active_power, auto.grid=FALSE, xaxt="n",
           ylab="Global Active Power", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)

  # Plot 2:
  plot.xts(HPC.sub$Voltage, auto.grid=FALSE, xaxt="n",
           xlab="datetime", ylab="Voltage", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)

  # Plot 3:
  plot(HPC.sub.zoo, xaxt="n", screens=1,
       xlab="", ylab="Energy sub metering", main="", col=my.colors)
  # Use x-axis calculation from the 'Plot 2' code chunk.
  axis(1, at=ticks, labels=labs, tcl=-0.5)
  legend("topright", col=my.colors, lty=1, legend=colnames(HPC.sub.zoo), bty="n")

  # Plot 4:
  plot.xts(HPC.sub$Global_reactive_power, auto.grid=FALSE, xaxt="n",
           xlab="datetime", ylab="Global_reactive_power", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)
} else {
  # Create the 'png' file:
  png(filename="plot4.png")
  par(mfrow=c(2,2), mar=c(4,4,1,1), oma=c(1,1,1,1))

  # Plot 1:
  plot.xts(HPC.sub$Global_active_power, auto.grid=FALSE, xaxt="n",
           ylab="Global Active Power", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)

  # Plot 2:
  plot.xts(HPC.sub$Voltage, auto.grid=FALSE, xaxt="n",
           xlab="datetime", ylab="Voltage", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)

  # Plot 3:
  plot(HPC.sub.zoo, xaxt="n", screens=1,
       xlab="", ylab="Energy sub metering", main="", col=my.colors)
  # Use x-axis calculation from the 'Plot 2' code chunk.
  axis(1, at=ticks, labels=labs, tcl=-0.5)
  legend("topright", col=my.colors, lty=1, legend=colnames(HPC.sub.zoo), bty="n")

  # Plot 4:
  plot.xts(HPC.sub$Global_reactive_power, auto.grid=FALSE, xaxt="n",
           xlab="datetime", ylab="Global_reactive_power", main="")
  axis(1, at=ticks, labels=labs, tcl=-0.5)
  dev.off()
}
```


Global Active Power
0
2
4
6
Thu
Fri
Sat


Voltage
234
238
242
246
Thu
Fri
Sat
datetime


Energy sub metering
0
10
20
30
Sub_metering_1
Sub_metering_2
Sub_metering_3
Thu
Fri
Sat


Global_reactive_power
0.0
0.1
0.2
0.3
0.4
0.5
Thu
Fri
Sat
datetime