R Cheatsheet

Copyright 2014, Bruce Beauchamp

Contents

[R Cheatsheet 1](#_Toc416580130)

[Best Subset Regression 4](#_Toc416580131)

[Control statements 4](#_Toc416580132)

[Iterating through a vector 4](#_Toc416580133)

[data.frames 5](#_Toc416580134)

[Append a column to a data.frame 5](#_Toc416580135)

[Append a row to a data.frame 5](#_Toc416580136)

[Convert a column to date data type 5](#_Toc416580137)

[Create a data.frame from scratch 6](#_Toc416580138)

[Edit a data.frame in table form 6](#_Toc416580139)

[Expose the fields in a data.frame to the development environment 6](#_Toc416580140)

[Get a cell of data from a data.frame 6](#_Toc416580141)

[Get the column names of a data.frame 6](#_Toc416580142)

[Get the number of rows in a data.frame 7](#_Toc416580143)

[Get a Row of Data from a data.frame 8](#_Toc416580144)

[Import a data.frame column as a formatted date 8](#_Toc416580145)

[Look at the structure of a data.frame 8](#_Toc416580146)

[Select Rows which meet criteria 8](#_Toc416580147)

[Select Columns from a data.frame 8](#_Toc416580148)

[Data Types 9](#_Toc416580149)

[Get the type of an object 9](#_Toc416580150)

[Dates and Times 9](#_Toc416580151)

[Convert a String to a Date 9](#_Toc416580152)

[Environment 9](#_Toc416580153)

[Change the Working Directory 9](#_Toc416580154)

[Get the Working Directory 9](#_Toc416580155)

[List the Files in the Working Directory 9](#_Toc416580156)

[Load a Function into the Environment 9](#_Toc416580157)

[List the Functions Available in the Environment 9](#_Toc416580158)

[Files and Directories 10](#_Toc416580159)

[Create a directory 10](#_Toc416580160)

[Download a file from the web 10](#_Toc416580161)

[See if a directory or file exists 10](#_Toc416580162)

[Libraries 10](#_Toc416580163)

[Loading Libraries 10](#_Toc416580164)

[Linear Regression 11](#_Toc416580165)

[Confidence interval for a regression line fit 11](#_Toc416580166)

[Generate the Regression line 11](#_Toc416580167)

[Get more information about the linear regression 11](#_Toc416580168)

[Plot a regression line on a plot 11](#_Toc416580169)

[Matrices 12](#_Toc416580170)

[Create a matrix 12](#_Toc416580171)

[Misc 12](#_Toc416580172)

[Printing a variable to the console 12](#_Toc416580173)

[Packages 12](#_Toc416580174)

[Installing packages 12](#_Toc416580175)

[Plotting data 13](#_Toc416580176)

[Line Plots 13](#_Toc416580177)

[Histograms 13](#_Toc416580178)

[Line plots with multiple lines 14](#_Toc416580179)

[Plotting to a PNG file 14](#_Toc416580180)

[Scatter plot of y vs x 15](#_Toc416580181)

[Reading Data 16](#_Toc416580182)

[Read a CSV file into a data.frame 16](#_Toc416580183)

[Read from an excel spreadsheet 16](#_Toc416580184)

[Strings 16](#_Toc416580185)

[Concatenate two strings 16](#_Toc416580186)

[Padding a string with zeros 16](#_Toc416580187)

[Vectors 17](#_Toc416580188)

[Concatenate two string vectors 17](#_Toc416580189)

[Creating a vector 17](#_Toc416580190)

[Iterating over a vector 17](#_Toc416580191)

[Length of a vector 17](#_Toc416580192)

[Return valid elements of a vector 17](#_Toc416580193)

# Best Subset Regression

Use the leaps() package.

library(leaps)

regfit.full = regsubsets(Salary~., data=Hitters)

# Control statements

## Iterating through a vector

id = 1:3

for (i in id) {

cat(i, "\n")

}

|  |
| --- |
| 1  2  3 |
|  |
| |  | | --- | |  | |

# data.frames

## Append a column to a data.frame

Note: The column data is inside a vector

|  |
| --- |
| > str(energy\_use\_subset)  'data.frame': 2880 obs. of 9 variables:  $ Date : chr "1/2/2007" "1/2/2007" "1/2/2007" "1/2/2007" ...  $ Time : chr "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...  $ Global\_active\_power : num 0.326 0.326 0.324 0.324 0.322 0.32 0.32 0.32 0.32 0.236 ...  $ Global\_reactive\_power: num 0.128 0.13 0.132 0.134 0.13 0.126 0.126 0.126 0.128 0 ...  $ Voltage : num 243 243 244 244 243 ...  $ Global\_intensity : num 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1 ...  $ Sub\_metering\_1 : num 0 0 0 0 0 0 0 0 0 0 ...  $ Sub\_metering\_2 : num 0 0 0 0 0 0 0 0 0 0 ...  $ Sub\_metering\_3 : num 0 0 0 0 0 0 0 0 0 0 ...  > str(dt)  POSIXlt[1:2880], format: "2007-02-01 00:00:00" "2007-02-01 00:01:00" "2007-02-01 00:02:00" ...  > energy\_use\_subset$date\_time <- dt  > str(energy\_use\_subset)  'data.frame': 2880 obs. of 10 variables:  $ Date : chr "1/2/2007" "1/2/2007" "1/2/2007" "1/2/2007" ...  $ Time : chr "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...  $ Global\_active\_power : num 0.326 0.326 0.324 0.324 0.322 0.32 0.32 0.32 0.32 0.236 ...  $ Global\_reactive\_power: num 0.128 0.13 0.132 0.134 0.13 0.126 0.126 0.126 0.128 0 ...  $ Voltage : num 243 243 244 244 243 ...  $ Global\_intensity : num 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1 ...  $ Sub\_metering\_1 : num 0 0 0 0 0 0 0 0 0 0 ...  $ Sub\_metering\_2 : num 0 0 0 0 0 0 0 0 0 0 ...  $ Sub\_metering\_3 : num 0 0 0 0 0 0 0 0 0 0 ...  $ date\_time : POSIXlt, format: "2007-02-01 00:00:00" "2007-02-01 00:01:00". |
|  |
|  |

## Append a row to a data.frame

> df

id nobs

1 1 2

> df <- rbind(df, data.frame(id=17, nobs=34))

> df

id nobs

1 1 2

2 17 34

## Convert a column to date data type

> energy\_use$Date <- as.Date(energy\_use$Date, format ="%d/%m/%Y")

> str(energy\_use)

'data.frame': 5258 obs. of 9 variables:

$ Date : Date, format: "2007-02-01" "2007-02-01" "2007-02-01" ...

## Create a data.frame from scratch

> df <- data.frame(id=integer(), nobs=integer())

> df <- rbind(df, data.frame(id=1, nobs=2))

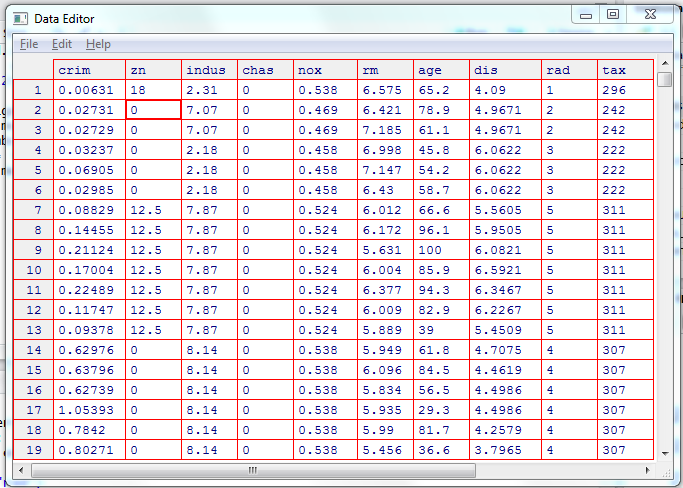
> df

id nobs

1 1 2

## Edit a data.frame in table form

> fix(Boston)



## Expose the fields in a data.frame to the development environment

attach(Boston)

## Get a cell of data from a data.frame

> pollutantData[1,]

Date sulfate nitrate ID

1 2003-01-01 NA NA 1

> pollutantData[1,"ID"]

[1] 1

## Get the column names of a data.frame

> colnames(outcomeData)

## Get the number of rows in a data.frame

|  |
| --- |
| > nrow(data1)  [1] 153 |
|  |
| |  | | --- | |  | |

## Get a Row of Data from a data.frame

|  |
| --- |
| data1[1,]  Ozone Solar.R Wind Temp Month Day  1 41 190 7.4 67 5 1 |
|  |
| |  | | --- | |  | |

## Import a data.frame column as a formatted date

setClass('myDate')

setAs("character","myDate", function(from) as.Date(from, format="%d/%m/%Y") )

energy\_use <- read.csv("data/household\_power\_consumption\_short.txt",

sep=";", colClasses= c("myDate", "character", "numeric",

"numeric","numeric","numeric",

"numeric","numeric","numeric"))

## Look at the structure of a data.frame

> str(energy\_use)

'data.frame': 5258 obs. of 9 variables:

$ Date : Factor w/ 4 levels "1/2/2007","2/2/2007",..: 1 1 1 1 1 1 1 1 1 1 ...

$ Time : Factor w/ 1440 levels "00:00:00","00:01:00",..: 1 2 3 4 5 6 7 8 9 10 ...

$ Global\_active\_power : num 0.326 0.326 0.324 0.324 0.322 0.32 0.32 0.32 0.32 0.236 ...

$ Global\_reactive\_power: num 0.128 0.13 0.132 0.134 0.13 0.126 0.126 0.126 0.128 0 ...

$ Voltage : num 243 243 244 244 243 ...

$ Global\_intensity : num 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1 ...

$ Sub\_metering\_1 : num 0 0 0 0 0 0 0 0 0 0 ...

$ Sub\_metering\_2 : num 0 0 0 0 0 0 0 0 0 0 ...

$ Sub\_metering\_3 : num 0 0 0 0 0 0 0 0 0 0 ...

## Select Rows which meet criteria

> test <- iris[iris$Species=="virginica", ]

> test

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

101 6.3 3.3 6.0 2.5 virginica

102 5.8 2.7 5.1 1.9 virginica

103 7.1 3.0 5.9 2.1 virginica

## Select Columns from a data.frame

outcomeData <- outcomeData[, c(2, 7, column)]

Here, columns 2, 7, and one additional column are retained

# Data Types

## Get the type of an object

|  |
| --- |
| > x <- 4  > class(x)  [1] "numeric" |
|  |
|  |

# Dates and Times

## Convert a String to a Date

> strptime("1/2/2007 00:01:00", "%d/%m/%Y %H:%M:%S")

[1] "2007-02-01 00:01:00 PST"

> class(strptime("1/2/2007 00:01:00", "%d/%m/%Y %H:%M:%S"))

[1] "POSIXlt" "POSIXt"

# Environment

## Change the Working Directory

> setwd("c:/r/Prog3")

> getwd()

[1] "c:/r/Prog3"

## Get the Working Directory

> getwd()

[1] "C:/Users/Bruce/Documents"

>

## List the Files in the Working Directory

dir()

## Load a Function into the Environment

source(“myCode.R”)

## List the Functions Available in the Environment

ls()

# Files and Directories

## Create a directory

> dir.create("data")

## Download a file from the web

download.file()

## See if a directory or file exists

> file.exists("no such dir")

[1] FALSE

# Libraries

## Loading Libraries

> library(ISLR)

# Linear Regression

## Confidence interval for a regression line fit

> confint(fit)

2.5 % 97.5 %

(Intercept) 33.448457 35.6592247

lstat -1.026148 -0.8739505

## Generate the Regression line

> fit = lm(medv~lstat, Boston)

> fit

Coefficients:

(Intercept) lstat

34.55 -0.95

Note generates the regression line for mdev as a function of lstat

## Get more information about the linear regression

> summary(fit)

Residuals:

Min 1Q Median 3Q Max

-15.168 -3.990 -1.318 2.034 24.500

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 34.55384 0.56263 61.41 <2e-16 \*\*\*

lstat -0.95005 0.03873 -24.53 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

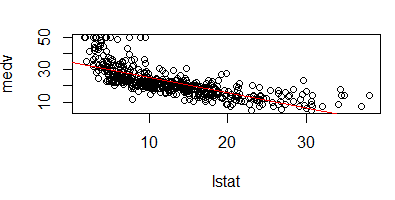
Residual standard error: 6.216 on 504 degrees of freedom

Multiple R-squared: 0.5441, Adjusted R-squared: 0.5432

F-statistic: 601.6 on 1 and 504 DF, p-value: < 2.2e-16

## Plot a regression line on a plot

> abline(fit, col="red")



# Matrices

## Create a matrix

x <- matrix(c(1,2,3,4), 2,2)

> x

[,1] [,2]

[1,] 1 3

[2,] 2 4

# Misc

## Printing a variable to the console

for (i in id) {

**cat(i, "\n")**

}

# Packages

## Installing packages

> install.packages("ISLR")

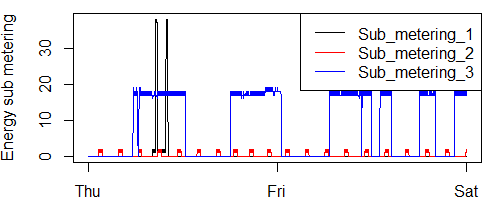
# Plotting data

## Adding a legend to a plot

legend(x='topright',

legend=c("Sub\_metering\_1 ", "Sub\_metering\_2 ", "Sub\_metering\_3 "),

col=c('black', 'red', 'blue'), lty=c(1,1,1), cex=0.5 )



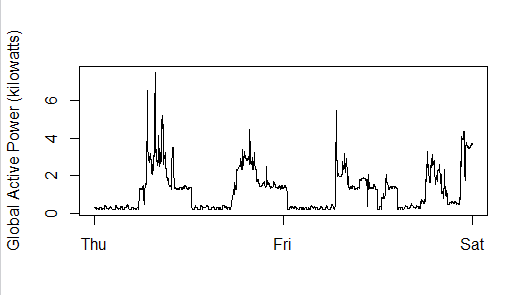
lty = line type. 0=blank, 1=solid

cex : Sets relative size of the legend

## Line Plots

plot(energy\_use\_subset$date\_time, energy\_use\_subset$Global\_active\_power, type="l",

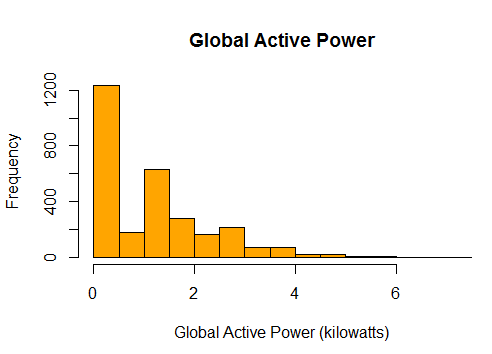
xlab='', ylab='Global Active Power (kilowatts)')



## Histograms

hist(energy\_use\_subset$Global\_active\_power, main='Global Active Power', col='orange',

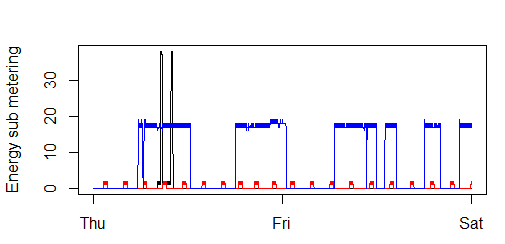
xlab='Global Active Power (kilowatts)')



## Line plots with multiple lines

plot(x,y1,type="l",col="red")

lines(x,y2,col="green")



## Multiple plots in quadrants

par(mfrow=c(2,2), mar=c(4,4,2,1), oma=c(0,0,2,0))

with(energy\_use\_subset, {

plot(energy\_use\_subset$date\_time, energy\_use\_subset$Global\_active\_power, type="l",

xlab='', ylab='Global Active Power (kilowatts)')

plot(energy\_use\_subset$date\_time, energy\_use\_subset$Voltage, type="l",

xlab='datetime', ylab='Voltage')

plot(energy\_use\_subset$date\_time, energy\_use\_subset$Sub\_metering\_1, type="l",

xlab='', ylab='Energy sub metering')

lines(energy\_use\_subset$date\_time, energy\_use\_subset$Sub\_metering\_2, col='red')

lines(energy\_use\_subset$date\_time, energy\_use\_subset$Sub\_metering\_3, col='blue')

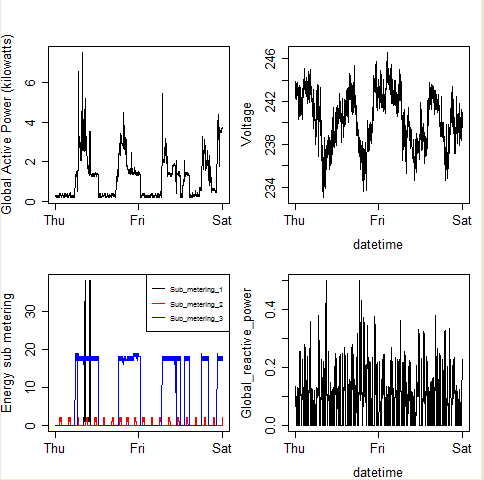
legend(x='topright', legend=c("Sub\_metering\_1 ", "Sub\_metering\_2 ", "Sub\_metering\_3 "),

col=c('black', 'red', 'blue'), lty=c(1,1,1), cex=0.5 )

plot(energy\_use\_subset$date\_time, energy\_use\_subset$Global\_reactive\_power, type="l",

xlab='datetime', ylab='Global\_reactive\_power')

})



## Plotting to a PNG file

hist(energy\_use\_subset$Global\_active\_power, main='Global Active Power', col='orange',

xlab='Global Active Power (kilowatts)')

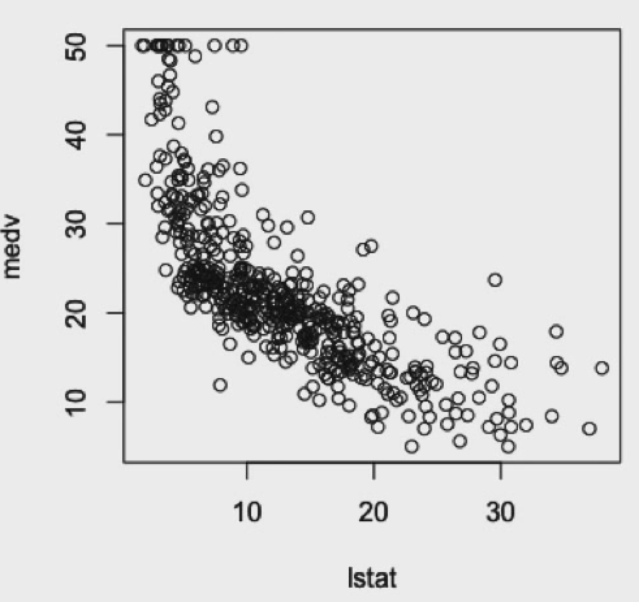
# Now copy the histogram to a png file

dev.copy(png, file='plot1.png', width = 480, height = 480 )

dev.off()

## Scatter plot of y vs x

plot(medv~lstat, Boston)



# Reading Data

## Read a CSV file into a data.frame

> data1 <- read.csv("hw1\_data.csv")

> data1

Ozone Solar.R Wind Temp Month Day

1 41 190 7.4 67 5 1

2 36 118 8.0 72 5 2

3 12 149 12.6 74 5 3

## Read from an excel spreadsheet

library(xlsx)

read.xlsx()

# Strings

## Concatenate two strings

|  |
| --- |
| > paste("a","b", sep = "")  [1] "ab" |
|  |
| |  | | --- | | > |   > paste0("1","2")  [1] "12" |

## Padding a string with zeros

> sprintf("%03d", 17)

[1] "017"

# Vectors

## Concatenate two string vectors

> date\_part <- energy\_use\_subset[, 'Date']

> time\_part <- energy\_use\_subset[, 'Time']

> date\_time\_str <- paste(date\_part, time\_part, sep=' ')

> str(date\_time\_str)

chr [1:2880] "1/2/2007 00:00:00" "1/2/2007 00:01:00" "1/2/2007 00:02:00" ...

## Creating a vector

> a = 1:2

> a

[1] 1 2

|  |
| --- |
| > x <- c(1,3,2,5)  > x  [1] 1 3 2 5 |
|  |
| |  | | --- | |  | |

## Iterating over a vector

for (i in id) {

writeLines(paste(i))

}

|  |
| --- |
| 1  2 |
|  |
| |  | | --- | |  | |

## Length of a vector

> bad <- ozone\_col[is.na(ozone\_col)]

> length(bad)

[1] 37

## Return valid elements of a vector

|  |
| --- |
| > a  [1] 1 2 NA 4  > a[complete.cases(a)]  [1] 1 2 4 |
|  |
| |  | | --- | |  | |