Python Notes

Steven Wang

March 28, 2016

```
#The parts of the Marin videos that have been noted:
#0
#1,2,3,4,5,6,7,8,9
#2.3,
#3.3,3.4,
#,4.9
#5.1,5.2,5.3


R is case sensitive

x = 11
y <- 11

ls() # show everything stored in workspace
rm(y) # delete y

x.1 = 14 # variable name can have period and number, but it could not start with number

xx = "marin" # assign characters to variable
yy = "1" # if include numbers in quotations, R will treat them as characters

sqrt(y)
y^(1/2)
log(y) # ln(y)
exp(y) # anti-log

log2(y) # log based on 2
abs(-14) # absolute value

# if you input an incomplete command, R would return with a "+" sign

c # concatenate
x1 = c(1,3,5,7,9)
```

```r

gender = c("male","female")

2   7  # integer values
seq(from=1, to=7,by=1)
seq(from=1, to=7,by=1/3)
rep(1,times=10)

rep(c("m","f"),times=5)

x = 1:5
x + 10
x * 10

# if two vectors are of the same length, we may add/subtract/muit/div corresponding elements // element-wise

y[3] #extract the 3rd element of y
y[-3] #extract all elements except the 3rd element of y
y[1:3] # extract 1st to 3rd elements
y[c(1,5)] # extract 1st and 5th
y[-c(1,5)] # extract all except 1st and 5th
y[y<6] # extract the elements that are less than 6

mat=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,byrow=TRUE) # "nrow": the number of rows; "byrow=TRUE": elements will be entered in a row-wise //"TRUE" has to be capital letters

mat[1,2] # extract element from row 1, column 2
mat[c(1,3),2] # extract elements from row 1 and 3, column 2
mat[2,] # extract row 2, all columns // leaving row or column blank to extract all columns or rows

mat*10 # multiply element-wise

#IMPORTING DATA AND WORKING WITH DATA
help(read.table) #help(command you want to know more about)
?read.table #show help
Data1 <- read.table(file="/DirectoryPath/data.txt", header=TRUE, sep="\t") #"header=TRUE" tells R that 1st row is header; "sep=\t": seperate by table(since the data in the example is table delimited). or "sep=",""","sep="""
read.csv(file, header = TRUE, sep = ",", quote = "\"", dec = ".", fill = TRUE, comment.char = "", ...)

Data2 = read.table(file.choose(), header=TRUE, sep = "\t") #press enter and R will let you choose your data source file
```

```
#if using R-studio, GUI, just click some buttons
#in Europe, it's common to use comma representing a decimal point

dim(mat) # show dimensions of the data, rows and columns

head(Data1) # show the 1st 6 rows of the data
tail(Data1) # show the last 6 rows of the data

Data1[c(5,6,7,8,9),] #square brackets: subset
Data1[5:9,]
Data[-(4:722)   ]

#subsetting data
names(LungCapData) #show the names of the variables
mean(LungCapDate$Age) #extract variable "Age" from LungCapData
LungCapData$Age

attaching data #pros:able to call variables by there names without "$" cons:put data in R's memory


mean(Age) #this would not work

attach(LungCapData)
detach(LungCapData)

class(Age) # return the type of the variable: integer, numeric, factor(/categorical)

levels(smoke) # show factors of a factor variable, like("yes" "no")

summary(LungCapData)

x = (c(0,1,1,1,0,0,0,0,0))
class(x)
x = as.factor(x)
class(x)

length(Age) # show how many obeservations are there under Age
Age[11:14]

mean(Age[Gender=="female"])

FemData = LungCapData[Gender=="female",] # create a subset of data containing only female and include all columns

```

```
MaleOver15 = LungCapData[Gender=="male" & Age>15,]

#Logic Statements and some other
temp = Age>15
temp[1:5] #return FALSE TRUE TRUE FALSE FALSE

temp2 = as.numeric(Age>15)
temp2[1:5] #return 0 1 1 0 0

FemSmoke <- Gender=="female" & Smoke=="yes"

MoreData <- cbind(LungCapData, FemSmoke) #cbind: add the new data in each row

rm(list=ls()) # remove all the thing in the workspace

#setting up working directory
getwd() #show working directory

setwd("/home/coupe")
setwd("~")

projectWD <- "/home/coupe/"
setwd(projectWD)

save.image("Marin.Rdata") #save this session .Rdata
load("Marin.Rdata")
load(file.choose())

#using scripts  .R
#select the commands and Run it


#histograms
hist(LungCap)
hist(LungCap, freq=FLASE)
hist(LungCap, freq=F)

hist(LungCap, prob=T)


hist(LungCap, prob=T, ylim=c(0, 0.2))
hist(LungCap, prob=T, ylim=c(0, 0.2), breaks=7)
hist(LungCap, prob=T, ylim=c(0, 0.2), breaks=seq(from=0, to=16, by=2))
hist(LungCap, prob=T, ylim=c(0, 0.2), breaks=seq(from=0, to=16, by=2), main="Boxplot of Lung Capacity", xlab="Lung Capacity", las=1) #las=1: rotate the y axis
```

```
lines(density(lungCap))
lines(density(lungCap), col=2, lwd=3) #"col=2": set color as 2//2 is red; "lwd=3": the width od the line

#3.3 Normal Distribution, Z Scores, and Normal Probabilities in R
#X~N(75,5^2)
help(pnorm)
pnorm(q=70, mean=75, sd=5, lower.tail=T) #by lower.tail, we mean less than 70
pnorm(q=70, mean=75, sd=5, lower.tail=T) #In R, by default, lower.tail=T)
pnorm(q=85, mean=75, sd=5, lower.tail=F)

pnorm(q=1, mean=0, sd=1, lower.tail=T) #calculate the probability of Z, the standard normal

qnorm() #calculate quantile or precentagetile
qnorm(p=0.25, mean=75, sd=5, lower.tail=T) #find Q1

dnorm() #probability function
x <- seq(from=55, to=95, by=0.25)
dens <- dnorm(x, mean=75, sd=5)

plot(x, dens) #"x" work as variable of "dens"
plot(x, dens, type="l") #change the plot from dots to a line

abline(v=75) #vertical line

rnorm() #draw a random sample from a normally distributed population
rand <- rnorm(n=40, mean=75, sd=5) #with 40 observations

#3.4 t Distribution and t Scores in R
#mean=0, sd=1, tf=25 //25 degrees of freedom
help(pt)
#P(t > 2.3)
pt(q=2.3, df=25, lower.tail=F) # one-sided p-value(the tail area which is greater than 2.3 //higher tail)
pt(q=2.3, df=25, lower.tail=F) + pt(q=-2.3, df=25, lower.tail=T) #two-sided p-value
pt(q=2.3, df=25, lower.tail=F)*2 #two-sided p-value

#find t-value for 95% confidence
#value of t with 2.5% in each tail
qt(p=0.025, df=25, lower.tail=T)

help(pf) #F probability
help(pexp) #exponetional probability

```

```r
#4.9 Correlation and Covariance in R
#LungCap and Age
help(cor.test)

(Age, LungCap, main="Scatterplot", las=1)

cor(Age, LungCap, method="pearson") #pearson is the default; the order of "Age" and "LungCap" does not matter)
cor(Age, LungCap, method="kendall")
cor.test(Age, LungCap, method="pearson")
cor.test(Age, LungCap, method="spearman") #R will return a warning
cor.test(Age, LungCap, method="spearman". exact=F)

cor.test(Age, LungCap, method="pearson", alt="greater", conf.level=0.99) #"alt":by default, the alternative is a two-sided test

cov(Age, LungCap) #coveriance

pairs(LungCapData) #produce all possible pair-wise plots
pairs(LungCapData[,1:3])

cor(LungCapData[,1:3]) #matrix of correlation
cor(LungCapData[,1:3], method="spearman")

cov(LungCapData[,1:3]) #matrix of covariance

#5.1 Linear Regression in R
#LungCap is the outcome or dependent (Y) variable
plot(Age, LungCap, main="Scatterplot")
cor(Age, LungCap)

help(lm)
?lm

mod <- lm(LungCap ~ Age) #1st variable is Y variableA, 2nd is X variable
summary(mod) #we have got understand this

attributes(mod) #return what's stored in mod
mod$coefficients
mod$coef
coef(mod)

abline(mod)

confint(mod) #confidence interval
confint(mod, level=0.99) # change confidence interval
```

```r
anova(mod) #this ANOVA table corresponds to the f-test presented in the last row of the linear regression summary

#5.2 Checking Linear Regression Assumptions in R
plot(Age, LungCap)
mod <- lm(LungCap ~ Age)

#intercept slop
#standard deviation of residual errors is called Residual Standard Error

#R has a set of built in regression diagnostic plots to check regression through residuals, r of the 4 assumptions
plot(mod) # actually shows 4 plots
par(mfrow=c(2,2)) #if you want to show all plots on the screen at the same time
plot(mod)

#How non-constant variance will show up in a residual plot

#5.3 Multiple Linear Regression in R
help(lm)
model1 <- lm(LungCap ~ Age + Height)
summary(model1)

cor(Age, Height, method="pearson")
#return shows high collinearity, not good
confint(model1, conf.level=0.95) # set confidence interval

model2 <- lm(LungCap ~ Age + Height + Smoke + Gender + Caesarean)
summary(model2)

# ############################################################################
##not from marin
# ############################################################################


edit(file = "test.R")
file.edit('foo.R')

20150203
as.POSXITlt("YYYY-MM-DD")->A
as.Date(as.character("19990101"), "%Y%m%d")
# "%M": minute, "%m": month

```

```
%d  day as a number (0-31)
%a  abbreviated weekday
%A  unabbreviated weekday
%m  month (00-12)
%b  abbreviated month
%B  unabbreviated month
%y  2-digit year
%Y  4-digit year


my_list<-list()
my_list<-list(vec1,vec2,mat1)
mylist[[1]]
#call the first element in the list

mylist<-list(number=vec1, people=vec2,m1=mat1)
mylist$number
#call the first element in the list

while~
{
    if~
         next
    ~
}

while~
{
    if~
         break
    ~
}

mat[logical mat]
mat[mat<1]
a=sort(a)

sum(data)
rowSums(data)
colSums(data)
as.matrix(data)

#20150328

Normal Probability Plot(QQ plot)
-The normal probability plot is a graphical technique for assessing whether or not a data set is approximately normally distributed.
```

```
-The data are plotted against a theoretical normal distribution in such a way that the points should form an approximate straight line.
-Departures from  this straight line indicate departures from normality.
-The relevant R functions are qqnorm() and qqline().

qqnorm(x)
qqline(x)

Time Series
The ts() function will convert a numeric vector into an R time series object. The format is ts(vector, start=, end=, frequency=) where start and end are the times of the first and last observation and frequency is the number of observations per unit time (1=annual, 4=quartly, 12=monthly, etc.).
a=ts(b)
plot(a)

diff(x)
#takes differences between current value and its previous one.
```

R