**Expository Data Analysis w/ R**

**ECOG 314 – 001 and ECON-181-001**

**First Course Meeting is September 2**

**(Revised: 6:31 PM, September 15, 2016)**

**Introduction to Data Exploration and Analysis with R**

**About this Course**

Conducting data and econometric analysis requires not only an understanding of theoretical concepts, but practical knowledge of how to carry out empirical work. Statistical programming packages are the means by which empirical work is conducted. In this course, students will learn how to use one such language, R, as a means of building their empirical toolkit. R has become one of the leading languages in data science and statistics. The program, which is free, is the tool of choice for data science professionals in academia, research, and industry. R users include full-time number crunchers, data curators, data visualization experts, and occasional data analysts.

The course will expose you to the basics of R as applied to cleaning up messy data, breaking up large datasets into manageable pieces, uncovering patterns, deriving insights, making predictions using statistical methods, and clearly communicating statistical findings. As a result, students will expand upon and put into practice many of the concepts they covered in econometrics.

This introductory course will meet weekly on Fridays from 9:00 am to 12:00 p.m., and will include lectures, labs with help and meeting with Federal Reserve Board research assistants. **Class will meet in the Federal Reverse Board’s building at 1801 K-Street, NW. Washington, DC.** (*We are offering Metro reimbursement to and from Shaw/Howard stop to Farragut North to students, which will be paid in a lump sum in December*). We will provide instructions to registered students for clearing security and entering the building.

William Ampeh, a Lead Technology Analyst at the Federal Reserve Board has developed the course content with a team of other Federal Reserve staff, and they will run the course meetings. Andrew Cohen, an Assistant Director at the Federal Reserve Board and Visiting Professor in the Economics Department will coordinate logistics for the course.

**Aim of Course**

This course, informally titled, “*Introduction to Data Exploration and Analysis with R*” provides a supportive, hands-on environment for students to learn R and apply and expand their existing knowledge of econometrics by conducting statistical analysis in R.

You will be introduced to concepts and techniques that will help you learn how to program in R and master the basic syntax. A range of vocabulary will be introduced to help you solve common statistical problems. You will be encouraged to continuously practice and expand on sample programs presented in class by solving weekly assignments and by participating in a group coding project. Additional help will be available both in and out of class if needed.

**What You Will Learn**

Starting with variables and basic operations, you will learn how to handle data structures such as vectors, matrices, data frames and lists. You will then learn to load data from a variety of formats (including SAS, Excel and text) into R, cleanup and manipulate data (including locating missing data and transforming data), and store R datasets for future use. Next you will learn to describe and examine measurement data (descriptive statistics), fit regression models, setup simulations, construct contingency tables, and implement sampling techniques.. Finally, you will learn about the graphical capabilities of R, how to create and manage your packages with Git version control, and how to publish your package on GitHub.

After completing this course, you will be able to use R as a data analysis tool. Specifically, you will be able to:

* Create, read, modify and store R datasets
* Use available R packages and write functions in R
* Create figures and plots using R
* Perform and interpret parametric one- and two-sample tests using R
* Perform and interpret multiple linear regression using R
* Perform and interpret one-way ANOVA using R
* Create, manage and publish R packages

**Course Prerequisite**

All applicants must have completed a college level course in Econometrics with a grade of **B** **or higher**. No prior training in programming or data science is required.

**Required Text (free)**

1: An Introduction to R, by W. N. Venables, D. M. Smith and the R Core Team

URL: <https://cran.r-project.org/doc/manuals/R-intro.pdf>:

**Recommended Optional Texts and Online Reference Materials**

1: Statistical Analysis with R

“This introduction to the freely available statistical software package R is primarily intended for people already familiar with common statistical concepts.”

URL: <http://www.statoek.wiso.uni-goettingen.de/mitarbeiter/ogi/pub/r_workshop.pdf>

2: Getting Started in Data Analysis: Stata, R, SPSS, Excel: R

A self-guided tour to help you find and analyze data using Stata, R, Excel and SPSS. The goal is to provide basic learning tools for classes, research and/or professional development.

URL: <http://libguides.princeton.edu/dss/R>

3: Gareth James, et al. 2013. An Introduction to Statistical Learning with Applications in R.

Springer site to download the corrected 6th printing pdf with access to slides and 15 hours of lecture videos.

URL: <http://www-bcf.usc.edu/~gareth/ISL/>

4: There are many online resources for R. Here is a Twitter feed of posts by R bloggers.

URL: <https://twitter.com/Rbloggers>

**Computer**

A Windows or Mac laptop is required with the following minimum configuration: 4 GB RAM or higher; 320 GB hard disk; configured to allow the installation of R and RStudio software. A limited number of loaner laptops will be made available if needed, **for in-class use only**.

**Software**

R and selected R packages will be the primary software for this class. R is free. People around the world use and contribute to R. Prior knowledge of R is helpful but not required. Substantial instruction will be provided in lecture notes and assignments, and additional instructions will also be available in the online reference materials. R documentation comes with R. Many books and free online materials address R and/or R packages.

You may use either R or Revolution R Open (now Microsoft R open)

Link to Download R: [Comprehensive R Archive](http://cran.us.r-project.org/)

Link to Download Microsoft R Open: [Revolution R Open now Microsoft R Open](https://mran.revolutionanalytics.com/)

RStudio is the recommended R integrated development environment

RStudio Download: See <https://www.rstudio.com/home/>

RStudio is easy to install and the installation does not require any instruction. However, the following links provide additional setup and navigation guidance:

<http://web.cs.ucla.edu/~gulzar/rstudio/index.html>

<http://dss.princeton.edu/training/RStudio101.pdf>

<https://support.rstudio.com/hc/en-us/sections/200107586-Using-RStudio>

**Course Work**

Assignments (approx. weekly); Midterm project (take home); Final project (take home).

**Grading**

Numerical class grades will be based on the homework (30%), midterm project (30%) and the final project (40%). The instructor reserves the right to amend weighting.

**Midterm Project**

This will be an individual (solo) programming project to be presented in class. You will be given two weeks to do this project. You will submit your presentation slides, a write-up containing both your R code and its results, and an explanation of how you approached the problem and why you chose that approach.

**End of Semester Project**

Statistical data analysis of your choice using data from UCI data repository <https://archive.ics.uci.edu/ml/datasets.html>

**Course Syllabus**

**Class Notes/Assignment**

**1: Introduction to Basics**  
Part 1: Install R and RStudio; Start RStudio, explore the features, menus and windows in RStudio, and take your first steps with R.

Part 2: Introduce the basic R data types including vectors, arrays, lists, matrices, data frame and factors. Explore basic operations on the basic R data types.

Part 3: Load the *mosaic* package, and display the functions in the *mosaic* package (using Google search or *ls(“package:mosaic”*). Use *help (?)* examine the *summary* function.

Part 4: Comment your work, save your workspace, and exit your R session.

Homework 1 assigned.

**2: Data Input, Management and Output**  
Part 1: Read external files, keep only the variables needed, display a few lines of dataset, add comments to help later users understand what is in the dataset, and save the dataset into a native format for future use.

Part 2: Clean the R workspace, load and display the saved dataset. Use *tally()* and *favstat()* from the *mosaic* package to display the distribution and relevant information on the dataset.

Part 3: Select variables in your dataset (by subset, column name, using logic, string search, using *$* notation and by simple name).

Part 4: Transform variables using the *DPlyr* package, handle missing values, rename variables, keep and drop variables, remove duplicate observations, create summarized or aggregated datasets.

Part 5: Export your dataset to some other format (SAS, MATLAB, CSV).

Homework 2 assigned.

**3: Descriptive Statistics and Exploratory Data Analysis (EDA) in R** (revised, 09/15/2016)  
Part 1: Calculating summary statistics (min, max, mean, median, quantiles, skewness). Centering, normalizing and scaling data.

Part 2: EDA graphs (*histogram()*, *boxplot()*, *densityplot()*, *qqnorm()*).

Part 3: Test for continuous variables (test for normality, student’s *t* test, equal variances test, non-parametric tests), P-values, confidence interval estimation, power of a test.

Homework 5 assigned.

**4: Generating Data**   
Part 1: Generate numeric sequences, factors, repetitious patterns, text to create filenames, and simple loops (iteration).

**Part 2: Generate r**andom data and simulate data that satisfy specific constraints.

Part 3: Sample data and compute statistics.

Part 4: Large dataset considerations.

Part 5: Manage R dataset, files and workspace.

Homework 4 assigned.

**5: R Programming and Operating System Interface --** (revised, 09/15/2016)  
Part 1: Sequences and simple loops (iteration), conditional execution.

**Part 2:** Writing functions, specifying function arguments and output, writing for loops, and testing variable scope.

**Part 3: I**mplement functions for selected descriptive statistics.

Part 4: Interactions with the operating system (*getwd()*, *setwd()*, *list.files()*).

Homework 3 assigned.

**Project 1: \* \* \* Midterm Project Presentation \* \* \***

**6: Simple Linear Regression and ANOVA in R**  
Part 1: Model fitting (linear regression, linear regression with categorical covariates, linear regression with interactions, predicted values, residuals).

**Part 2: C**onduct and interpret analysis of variance (ANOVA).

Homework 6 assigned.

**7: Random Samples in R**  
Part 1: Generating data samples of a specified size in R.

Part 2: Random sampling a dataset in R using the *sample()* function.

Part 3: Speeding up processing by replacing *for loops* with matrices.

Homework 7 assigned.

**8: Date and Time Variables in R**  
Part 1: Create and access Date-Time variables.

**Part 2: Date and time operations.**

**Part 3: Quick introduction to time series data management in R.**

**Part 4: Time series plotting.**

Homework 8 assigned.

**9: Producing Graphs in R**  
Part 1: Traditional graphs (line charts, bar charts, histograms, dotplots).

**Part 2: R graphics packages (ggplot, lattice) and graphics devices.**

**Part 3: R graphics parameters and plotting style (single and multi-plots).**

**Part 4: Scatter plots with large datasets (jittering, small pints and binning).**

**Part 5: Introduction to ggplot.**

Homework 9 assigned.

**10: R Package, Managing R Programs with Git and GitHub**  
Part 1: Build your first R package.

**Part 2:** Git(Hub) and (R) Markdown crash course.

Homework 9 assigned.

**Project 2: \* \* \* Final Class Project Presentation\* \* \***

Understand data analysis via EDA as a journey and a way to explore data

Explore data at multiple levels using appropriate visualizations

Acquire statistical knowledge for summarizing data

Demonstrate curiosity and skepticism when performing data analysis

Develop intuition around a data set and understand how the data was generated.

1: Introduction to R

1. Hello World using Base-R
2. Language Basics using Base R
3. Efficient and reproduce-able coding using RStudio IDE
4. Walking through the RStudio IDE

2: Transforming and cleaning data

3: Calculating descriptive statistics

4: Visualizing data

5: Moving beyond R and exploratory data analysis

Introduction to R and Exploratory data analysis

Gavin Simpson

November 2006

Summary

In this practical class we will introduce you to working with R. You will complete an

introductory session with R and then use a data set of Spheroidal Carbonaceous Particle

(SCP) surface chemistry and demonstrate some of the Exploratory Data Analysis (EDA)

functions in R. Finally, we introduce the concept of statistical tests in R through a small

study of fecudity in a predatory gastropod on intertidal shores.

1 Your first R session

R can be used as a glorified calculator; you enter calculations at the prompt, R works out the

answer and displays it to you. Try the following sequence of commands. Remember, only type

the commands on lines starting with the prompt “

>

” below. In each case, the output from R is

shown.

> 3

\*

5

[1] 15

> 7 + 2

[1] 9

> 100/5

[1] 20

> 99 - 1

[1] 98

1

After each sum R prints out the result with a

[1]

at the start of the line. This means the result

is a vector of length 1. As you will see later, this string shows you which entry in the vector is

displayed at the beginning of each line.

This isn’t much use if you can’t store results though. To store the result of a calculation, the

assignment operator “

<-

” is used.

> a <- 3

\*

5

Notice that this time R does not print the answer to the console. You have just created an object

in R with the name “

a

”. To print out the object to the console you just have to type the name of

the object and press

Enter

.

> a

[1] 15

We can now use this to do sums like you would on a calculator using the memory functions.

> b <- 7 + 2

> a + b

[1] 24

> a

\*

b

[1] 135

> c <- a

\*

b

> c

[1] 135

To see a list of the objects in the current workspace use the

ls()

function:

> ls()

[1] "a" "b" "c"

2

2 Exploratory Data Analysis

Use R’s EDA functions to examine the SCP data with a view to answering the following ques-

tions:

1. Suggest which chemical elements give the best discrimination between coal and oil par-

ticles;

2. Suggest which variables are highly correlated and may be dropped from the analysis;

3. Suggest which particles (either coal or oil) have an unusual chemical composition – i. e.,

are outliers;

SCP’s are produced by the high temperature combustion of fossil fuels in oil and coal-fired

power stations. Since SCP’s and sulphur (as

SO

2

) are emitted from the same sources and are

dispersed in the same way, the record of SCP’s in lake sediments may be used to infer the spa-

tial and temporal patterns of sulphur deposition, an acid loading on lake systems. In addition,

SCP’s produced by the combustion of coal and oil have different chemical signatures, so char-

acterization of particles in a sediment core can be used to partition the pollution loading into

distinct sources.

The data set consists of a sample of 100 SCP’s from two power stations (50 SCP’s from each).

One, Pembroke, is an oil-fired station, the other, Drax, is coal-fired. The data form a training

set that was used to generate a discriminant function for classifying SCP’s derived from lake

sediments into fuel type. The samples of fly-ash from power station flues were analysed using

Energy Dispersive Spectroscopy (EDS) and the chemical composition was measured.

These data are available in the file

scp.txt

, a standard comma-delimited ASCII text file. You

will use R in this practical class, learn how to read the data into R and how to use the software

to perform standard EDA and graphics.

3 Reading the data into R

Firstly, start R on the system you are using in the

prac1

directory that contains the files for

this practical, instructions on how to do this have been circulated on a separate handout. Your

screen should look similar (but not the same) to the one shown in Figure 1. The R prompt is

a “

>

” character and commands are entered via the keyboard and subsequently evaluated after

the user presses the return key.

The following commands read the data from the

scp.txt

file into a R object called

scp.dat

and perform some rudimentary manipulation of the data:

> scp.dat <- read.csv(file = "scp.txt", row.names = 1)

> scp.dat[, 18] <- as.factor(scp.dat[, 18])

> levels(scp.dat[, 18]) <- c("Coal", "Oil")

The

read.csv()

function reads the data into an object,

scp.dat

. The argument

row.names

instructs R to use the first column of data as the row names.

3

Figure 1: The R interface; in this case the R GUI interface under MicroSoft Windows

By default R assigns names to each observation. As the

scp.txt

file contains a column la-

belled

"SampleID"

, which contains the sample names, we now convert the first column of

the

scp.dat

object to characters (text) and then assign these as the sample names using the

rownames

function. Note that we subset the

scp.dat

object using the following notation:

object[

r

,

c

]

Here

r

and

c

represent the row and column required, respectively. To select the first column of

scp.dat

we use

scp.dat[,1]

), leaving the row identifier blank.

The last variable in

scp.dat

is

FuelType

. This variable is currently coded as

"1"

,

"0"

. R has a

special way of dealing with “factors”, but we need to tell R that

FuelType

is a factor (

FuelType

is currently a numeric variable). The function

as.factor()

is used to

coerce

a vector of data

from one type into a factor (

scp.dat[,18] <- as.factor(scp.dat[,18])

. The “levels” of

FuelType

are still coded as

1

,

0

. We can replace the current levels with something more use-

ful using

levels(scp.dat[,18]) <- c("Coal", "Oil")

. Note the use of the concatenate

function,

c()

, to create a character vector of length 2.

Simply typing the name of an object followed by return prints the contents of that object to

the screen (increase the size of your R console window before doing this otherwise the printed

contents will extend over many pages).

> scp.dat

> names(scp.dat)

> str(scp.dat)

The

names()

function prints out the names describing the contents of the object. The output

4

from names depends on the

class

of the object passed as an argument

1

to

names()

. For a data

frame like

scp.dat

,

names()

prints out the labels for the columns, the variables. The

str()

function prints out the

str

ucture of the object. The output from

str()

shows that

scp.dat

contains 17 numeric variables and one (

FuelType

) factor variable with the levels

"Coal"

and

"Oil"

.

4 Summary statistics

Simple summary statistics can be generated using

summary()

:

> summary(scp.dat)

summary()

is a generic function, used to summarize an object passed as an argument to it. For

data frames, numeric matrices and vectors,

summary()

prints out the minimum and maximum

values, the mean, the median and the upper and lower quartiles for each variable in the object.

For factor variables, the number of observations of each level is displayed.

Another way of generating specific summary information is to use the corresponding function,

e.g.

mean(), min(), range()

or

median()

:

> mean(scp.dat[, 1])

[1] -12.0025

Doing this for all 17 variables and for each of the descriptive statistical functions could quickly

become tedious. R contains many functions that allow us to quickly apply functions across

each variable in a data frame. For example, to get the mean for each variable in the

scp.dat

data frame you would type:

> apply(scp.dat[, -18], 2, mean)

Na Mg Al Si P S Cl

-12.0025 -7.4240 9.8417 6.9593 3.5364 56.2859 -1.0291

K Ca Ti V Cr Mn Fe

1.5607 -0.4613 0.5897 0.8021 -0.0081 -0.3536 3.9651

Ni Cu Zn

0.2438 0.3103 0.1268

As its name suggests,

apply()

takes as it argument a vector or array and applies a named func-

tion on either the rows, or the columns of that vector or array (the 2 in the above call indicate

that we want to work on the columns). Apply can be used to run any appropriate function

including a users own custom functions. As an example, we now calculate the trimmed mean:

1

Arguments are the values entered between the brackets of a call to a function, and indicate a variety of options,

such as the name of an object to work on or which method to use. A function’s arguments are documented in the

help pages for that function

5

> apply(scp.dat[, -18], 2, mean, trim = 0.1)

Na Mg Al Si P

-11.462000 -6.672250 9.327625 6.749750 3.650125

S Cl K Ca Ti

57.069125 -2.711250 1.527250 -0.712250 0.460250

V Cr Mn Fe Ni

0.853000 -0.168125 -0.377375 2.768500 0.244750

Cu Zn

0.129500 0.117875

In the above code snippet, we pass an additional argument to

mean()

,

trim = 0.1

, which

calculates the trimmed mean, by trimming 10% of the data off each end of the distribution

and then calculating the mean of the remaining data points. How this works is quite simple.

apply()

is told to work on columns 1 through 17 of the

scp.dat

object. Each column in turn

is passed to our function and

mean()

calculates the trimmed mean.

Try using

apply()

to calculate the range (

range()

), the variance (

var()

) and the standard

deviation (

sd()

) for variables 1 through 17 in

scp.dat

.

Having now looked at the summary statistics for the entire data set, we can now start to look

for differences in SCP surface chemistry between the two fuel types. This time we need to

subset the data and calculate the summary statistics for each level in

FuelType

. The are many

ways of subsetting a data frame, but the easiest way is to use the

aggregatey()

function.

First, attach

scp.dat

to the search path

2

using:

> attach(scp.dat)

Now try the following code snippet:

> aggregate(scp.dat[, -18], list(FuelType), mean)

Group.1 Na Mg Al Si P S Cl

1 Coal -15.482 -6.6932 20.0824 13.2710 4.0560 40.4590 4.2242

2 Oil -8.523 -8.1548 -0.3990 0.6476 3.0168 72.1128 -6.2824

K Ca Ti V Cr Mn Fe Ni

1 2.6146 -1.2518 1.0392 0.3276 0.1092 -0.2356 0.3748 -0.1174

2 0.5068 0.3292 0.1402 1.2766 -0.1254 -0.4716 7.5554 0.6050

Cu Zn

1 0.2378 0.2198

2 0.3828 0.0338

> aggregate(scp.dat[, -18], list(FuelType), mean, trim = 0.1)

2

Attaching an object to the search path tells R to look inside that object for variables as well as in the standard

environment. By attaching

scp.dat

to the search path we can use the variable names directly.

6