GENERAL ASSEMBLY

R: AN INTRODUCTION

Week 8

DSI-NYC Triassics, General Assembly

LEARNING OBJECTIVES

- You will be able to…
  - Compare R and Python
  - Write R scripts in the RStudio IDE
  - Load a dataset, describe it and visualize it

LESSON GUIDE

10 min	Opening	Contextualizing R
20 min	Introduction	Syntax, base functions, key libraries
20 min	Guided practice	Codealong of basic commands
30 min	Independent practice	Load a new dataset, explore and visualize
5 min	Conclusion	

WHY R?

- High-level, interpreted, dynamically typed language (like Python)
- Powerful for statistics, excellent for ad hoc analyses
- Common alternative to Python in data science world
- Open source, vibrant ecosystem
- Hadleyverse

WHY NOT MORE R, THEN?

- Less suited to production code
- If you need it, you can learn it

THIS ISN'T JUST ABOUT R

- You're becoming self-sufficient learners within the data science and programming world
- It will take you less time to 'get' R than it took for Python
- This pattern will hold with other new technologies

INTRO TO R

THE BASICS

THE ENVIRONMENT

- R is the language/software
- RStudio is an integrated development environment (IDE). You'll love it.

RStudio

example.R | w8d4_intro_to_r.R | WA_business_licenses.R

```
33  ##----Load and shape data----
34
35  # Load and combine scraped business license data from multiple files.
36  # UBIs are IDs; they are numeric but may have leading 0s, so must be read as strings.
37
38  load.data <- function(dir = data_dir){
39
40  setwd(dir)
41
42  df <- data.frame(UBI=character(),NAICS=character(),OPEN.DATE=character(),CLOSE.DATE=ch
43  for(dataFile in list.files()){
44    if (substr(dataFile,1,5)=="batch"){
45    temp <- read.csv(dataFile,sep='\t',header=TRUE,stringsAsFactors=FALSE,colClasses=c(r
46    df<-rbind(df,temp)}
47  }
48  rm(temp)
49
50  # Drop observations with NAs
51  df<-df[complete.cases(df),]
52
```

32:3 (Top Level) R Script

Console ~/WA-historical-business-license-patterns/data/

```
Coefficients:
          ar1      ar2     ma1     ma2
      -0.4789  -0.2648  0.0525  0.0372
s.e.   0.2489   0.2169  0.2565  0.2405

sigma^2 estimated as 190477:  log likelihood = -2061.95,  aic = 4133.9

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set -5.813825 435.6456 201.7525 -32.84621 54.20367 0.9110635 0.0008826697
          used (Mb) gc trigger (Mb) max used (Mb)
Ncells  824154 44.1    1442291 77.1  1442291 77.1
Vcells 4290822 32.8    8910808 68.0  8910808 68.0
>
```

Environment | History

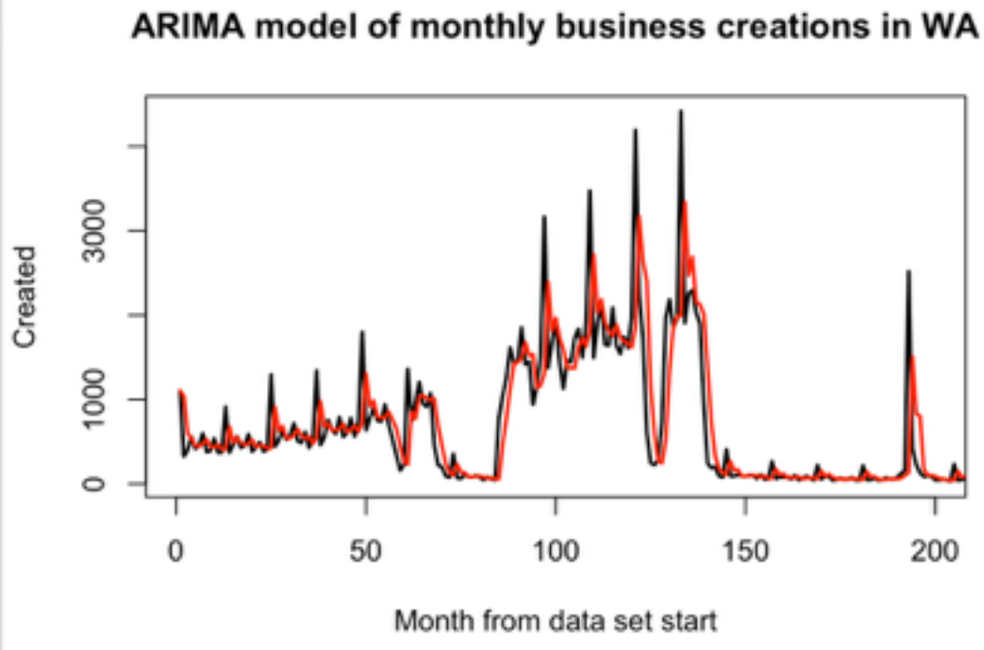
Import Dataset | List

Global Environment

yearValues	161 obs. of 4 variables
Values	
arimaFitted	Time-Series [1:276] from 1 to 276: 1106 1039 600 539…
arimaPredictions	List of 2
data_dir	"~/WA-historical-business-license-patterns/data/"
endYear	147L
f	List of 13
g	List of 13
i	3

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish



THE ENVIRONMENT

- Panels include the console, a text editor, an environment GUI, image viewer and more.

Check: try some arithmetic in the console. Then write it in the editor and run a line with command-enter.

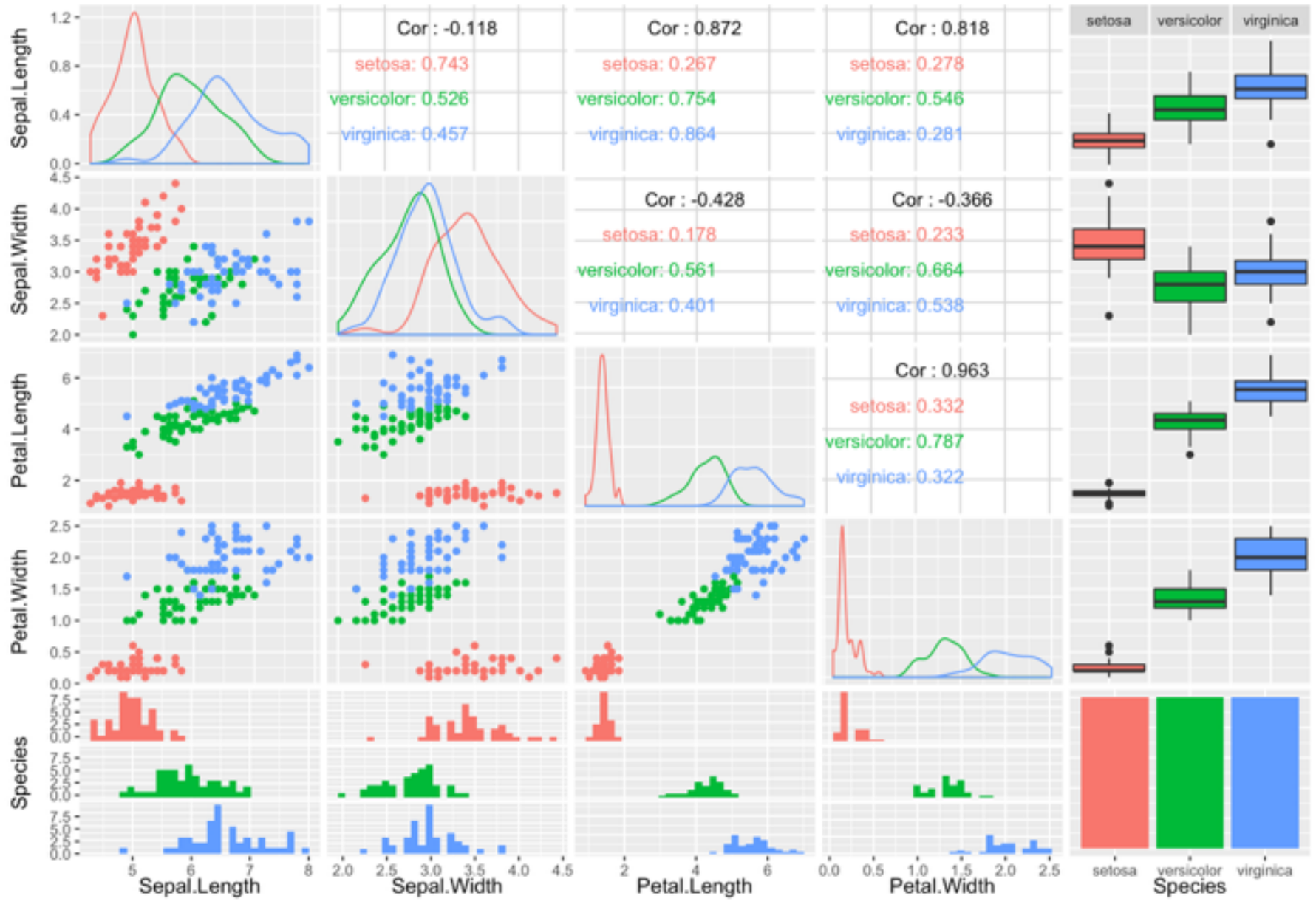
THE ENVIRONMENT

- Of the many wonderful RStudio features, its inline documentation might be the best:
- Type '?str' (or ?whatever) into the console

INTRO TO R

- Some motivation:
  - Load packages
    - install.packages(ggplot2)
    - install.packages(GGally)
    - library(ggplot2)
    - library(GGally)
  - Explore data
    - head(iris)
    - ggpairs(iris, mapping = aes(colour=Species))

INTRO TO R


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
Cor : -0.118
setosa: 0.743
versicolor: 0.526
virginica: 0.457
Cor : 0.872
setosa: 0.267
versicolor: 0.754
virginica: 0.864
Cor : 0.818
setosa: 0.278
versicolor: 0.546
virginica: 0.281
Cor : -0.428
setosa: 0.178
versicolor: 0.561
virginica: 0.401
Cor : -0.366
setosa: 0.233
versicolor: 0.664
virginica: 0.538
Cor : 0.963
setosa: 0.332
versicolor: 0.787
virginica: 0.322
setosa
versicolor
virginica

INTRO TO R

SYNTAX

INTRO TO R

- Whitespace
  - Not syntactical, as in most languages
  - Except newlines
- Assignment operator
  - <-, traditionally
  - = to set function parameters
- Indexing
  - Starts at 1
  - No negative indexing (a[-1] returns a without the first element)
- Case sensitive, variables names can include dots
  - ‘my.R.variable.name’

INTRO TO R

OBJECTS

EVERYTHING IS ONE

- class()
- vectors
- lists
- matrices
- arrays
- data.frame

TYPES

- typeof()
- integer
- numeric
- logical
- factor
- character
- complex

INTRO TO R

CONTROL FLOW

CONTROL FLOW

- if (condition) {true_expression else true_expression}
- for (variable in list) { expression }
- while (condition) {expression}

USER-DEFINED FUNCTIONS

- function.name <- function() {}

```
myfunction <- function(arg1, arg2, ... ){
  statements
  return(object)
}
```

INTRO TO R

DATAFRAMES

INTRO TO R

- Same tabular concept as pandas' dataframes - rows as observations, columns as attributes
- Indexing and reference
  - $
    - [1,]
    - [1]
- df functions
  - read_csv()
  - head(), tail()
  - summary()

INTRO TO R

- ‘apply’ and its variants are similar to using ‘map’ with functions in pandas / numpy
- A justifiably famed SO answer: http://stackoverflow.com/questions/3505701/r-grouping-functions-sapply-vs-lapply-vs-apply-vs-tapply-vs-by-vs-aggrega

INTRO TO R

GGPLOT2

INTRO TO R

- Based on a ‘grammar of graphics’
  - ggplot() … geom_() … scale_() … theme_()
- Real smooth
- Created by Hadley Wickham, prolific creator of the ‘Hadleyverse’ (presumably not a term he invented) of R packages

INTRO TO R

GUIDED PRACTICE

INTRO TO R

- Open up the codealong script

INTRO TO R

INDEPENDENT PRACTICE

CONCLUSION

FURTHER READING

- https://cran.r-project.org/doc/contrib/Short-refcard.pdf
- http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html
- http://www.cookbook-r.com/Graphs/
- http://www.noamross.net/blog/2014/4/16/vectorization-in-r--why.html

INTRO TO R

LAB

INTRO TO R

- 1. There are ML packages galore - fit a random forest to the iris dataset and assess the results
- 2. There is a package called ‘data.table’ which offers speed improvements for large datasets - install it, load the liquor sales dataset from project 3 into a data.table object, and tabulate some aggregate sales totals
  - https://www.datacamp.com/community/tutorials/data-table-r-tutorial