STAT 450: Case Studies in Statistics
Seminar 1: January 4, 2018

Kevin Multani

Department of Statistics, UBC

Introduction to R and RStudio, R workflow, R basics

Have R and RStudio installed on your computer!

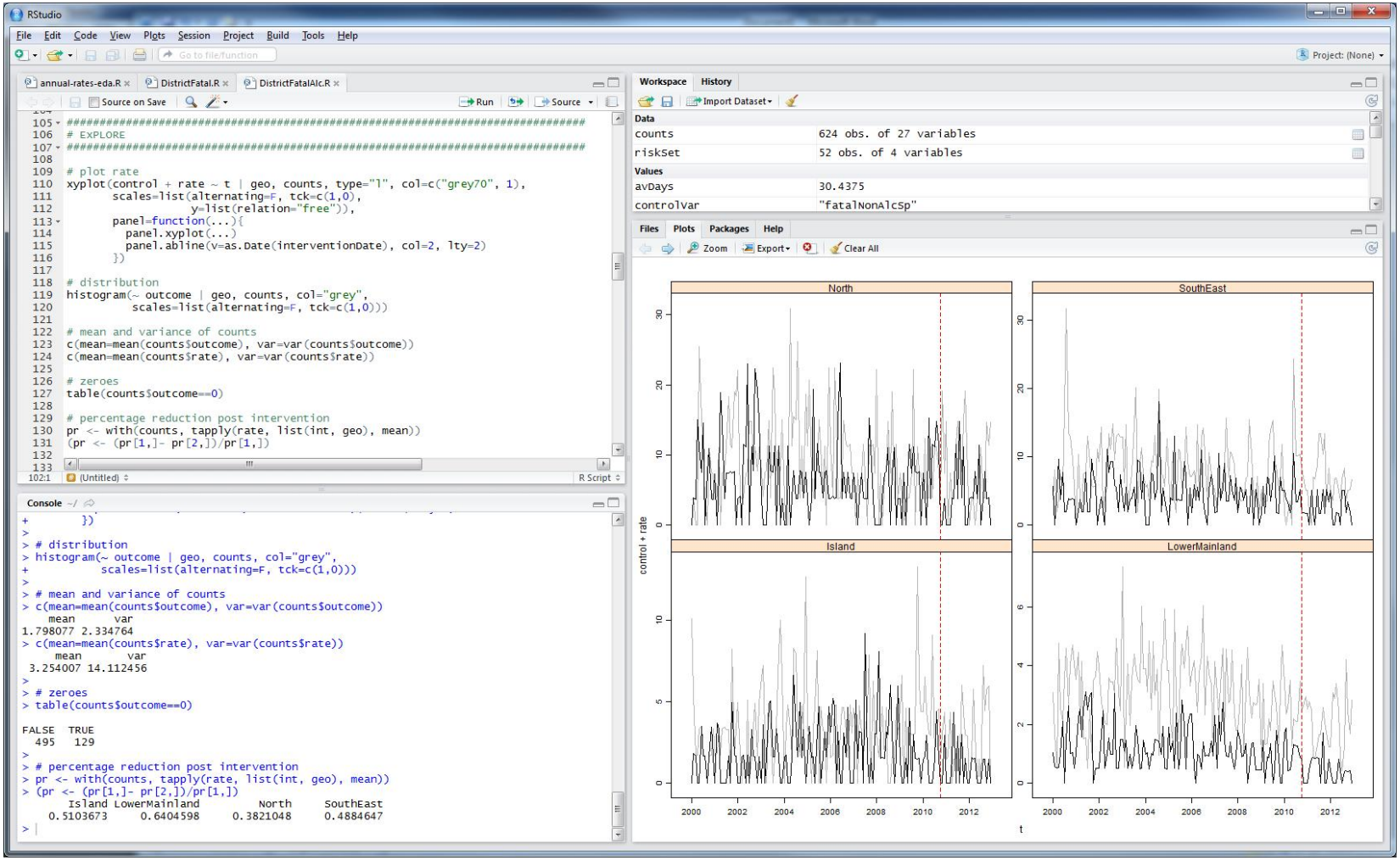
Ask for help if you run into problems or check out: http://stat545-ubc.github.io/block000_r-rstudio-install.html.

Programming with R and RStudio

The best way to learn how to program is to do something useful, so this is what we will do to review R basics: data analysis.

We recommend using RStudio to run R. RStudio is a more user-friendly interface for R. It allows you to write code, run programs, and view outputs all in one place.

RStudio


RStudio
File Edit Code View Plots Session Project Build Tools Help
annual-rates-eda.R
DistrictFatal.R
DistrictFatalAlc.R
Source on Save
Run
Source
105 ##########
106 # EXPLORE
107 ##########
109 # plot rate
110 xyplot(control + rate ~ t | geo, counts, type="l", col=c("grey70", 1),
111 scales=list(alternating=F, tck=c(1,0),
112 y=list(relation="free")),
113 panel=function(...){
114 panel.xyplot(...)
115 panel.abline(v=as.Date(interventionDate), col=2, lty=2)
116 })
118 # distribution
119 histogram(~ outcome | geo, counts, col="grey",
120 scales=list(alternating=F, tck=c(1,0)))
122 # mean and variance of counts
123 c(mean=mean(counts$outcome), var=var(counts$outcome))
124 c(mean=mean(counts$rate), var=var(counts$rate))
126 # zeroes
127 table(counts$outcome==0)
129 # percentage reduction post intervention
130 pr <- with(counts, tapply(rate, list(int, geo), mean))
131 (pr <- (pr[1,]- pr[2,])/pr[1,])
102:1 (Untitled) R Script
Console ~/
> # distribution
> histogram(~ outcome | geo, counts, col="grey",
+ scales=list(alternating=F, tck=c(1,0)))
> # mean and variance of counts
> c(mean=mean(counts$outcome), var=var(counts$outcome))
mean var
1.798077 2.334764
> c(mean=mean(counts$rate), var=var(counts$rate))
mean var
3.254007 14.112456
> # zeroes
> table(counts$outcome==0)
FALSE TRUE
495 129
> # percentage reduction post intervention
> pr <- with(counts, tapply(rate, list(int, geo), mean))
> (pr <- (pr[1,]- pr[2,])/pr[1,])
Island LowerMainland North SouthEast
0.5103673 0.6404598 0.3821048 0.4884647
Workspace History
Import Dataset
Data
counts 624 obs. of 27 variables
riskSet 52 obs. of 4 variables
Values
avDays 30.4375
controlVar "fatalNonAlcSp"
Files Plots Packages Help
Zoom Export Clear All
North
SouthEast
Island
LowerMainland
control + rate
t
2000 2002 2004 2006 2008 2010 2012

More RStudio features

RStudio has some very nice features that the usual R interface does not:

- "Syntax highlighting, code completion, and smart indentation" (from RStudio)
- "Quickly jump to function definitions" (from RStudio)
- "Easily manage multiple working directories using projects" (from RStudio) **We will briefly talk about this next week
- Communication with Github **We will talk about this next week

The Gapminder data

Now let's look at some of the data from the Gapminder project to review how to program in R (or more precisely to analyze data in R). Here is an exercept prepared by Prof Jenny Bryan: http://tiny.cc/gapminder.

Loading data into R

For rectangular spreadsheet-like (and "clean") data in plain text, use `read.table` (or `read.delim` or `read.csv`):

```
# Loading the data
gDat <- read.table(file = "http://tiny.cc/gapminder", sep = "\t",
    header = TRUE)

gDat <- read.delim(file = "http://tiny.cc/gapminder")
```

Data frames

When loading rectangular spreadsheet-like data into R, R automaticaly stores it as a `data.frame` object. `Data.frame` object is perhaps the most important object you will be dealing with because...

- most functions for inference, modelling, and graphing usually take `data.frame` objects as input
- it can store variables of different "types" (will talk about this more shortly), such as character data, numerical data, categorical data

Overviewing data frames

```
# structure of the data
str(gDat)
```

```
## 'data.frame': 1704 obs. of  6 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ pop      : num  8425333 9240934 10267083 11537966 13079460 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ gdpPercap: num  779 821 853 836 740 ...
```

Overviewing data frames

```
# look at the first few and last few of the data
head(gDat)
```

```
##       country year      pop continent lifeExp gdpPercap
## 1 Afghanistan 1952  8425333      Asia   28.80     779.4
## 2 Afghanistan 1957  9240934      Asia   30.33     820.9
## 3 Afghanistan 1962 10267083      Asia   32.00     853.1
## 4 Afghanistan 1967 11537966      Asia   34.02     836.2
## 5 Afghanistan 1972 13079460      Asia   36.09     740.0
## 6 Afghanistan 1977 14880372      Asia   38.44     786.1
```

```
tail(gDat)
```

```
##       country year      pop continent lifeExp gdpPercap
## 1699 Zimbabwe 1982  7636524    Africa   60.36     788.9
## 1700 Zimbabwe 1987  9216418    Africa   62.35     706.2
## 1701 Zimbabwe 1992 10704340    Africa   60.38     693.4
## 1702 Zimbabwe 1997 11404948    Africa   46.81     792.4
## 1703 Zimbabwe 2002 11926563    Africa   39.99     672.0
## 1704 Zimbabwe 2007 12311143    Africa   43.49     469.7
```

Overviewing data frames

```
# more useful functions
dim(gDat)  # dimension

## [1] 1704    6

nrow(gDat)  # number of rows

## [1] 1704

ncol(gDat)  # number of cols

## [1] 6
```

Overviewing data frames

```
# statistical overview
summary(gDat)
```

```
##         country           year            pop                continent
##  Afghanistan:  12   Min.   :1952   Min.   :6.00e+04   Africa  :624
##  Albania    :  12   1st Qu.:1966   1st Qu.:2.79e+06   Americas:300
##  Algeria    :  12   Median :1980   Median :7.02e+06   Asia    :396
##  Angola     :  12   Mean   :1980   Mean   :2.96e+07   Europe  :360
##  Argentina  :  12   3rd Qu.:1993   3rd Qu.:1.96e+07   Oceania : 24
##  Australia  :  12   Max.   :2007   Max.   :1.32e+09
##  (Other)    :1632
##     lifeExp        gdpPercap
##  Min.   :23.6   Min.   :   241
##  1st Qu.:48.2   1st Qu.:  1202
##  Median :60.7   Median :  3532
##  Mean   :59.5   Mean   :  7215
##  3rd Qu.:70.8   3rd Qu.:  9325
##  Max.   :82.6   Max.   :113523
##
```

Manipulating data frames

Sometimes we may want to have only little pieces of the data, e.g., certain column(s) or certain row(s) or both!

To extract one variable only:

```
gDat$country  # a categorical variable
gDat$pop  # a numerical variable
```

Manipulating data frames

To extract the rows which correspond to the year 2007:

```
subset(gDat, year == 2007)
```

To extract the rows which correspond to the year 2007 and select only the variables `country` and `pop`:

```
subset(gDat, year == 2007, select = c(country, pop))
```

Note: We can still accomplish the above using `[rows, cols]` command, but the code will be less self-documenting.

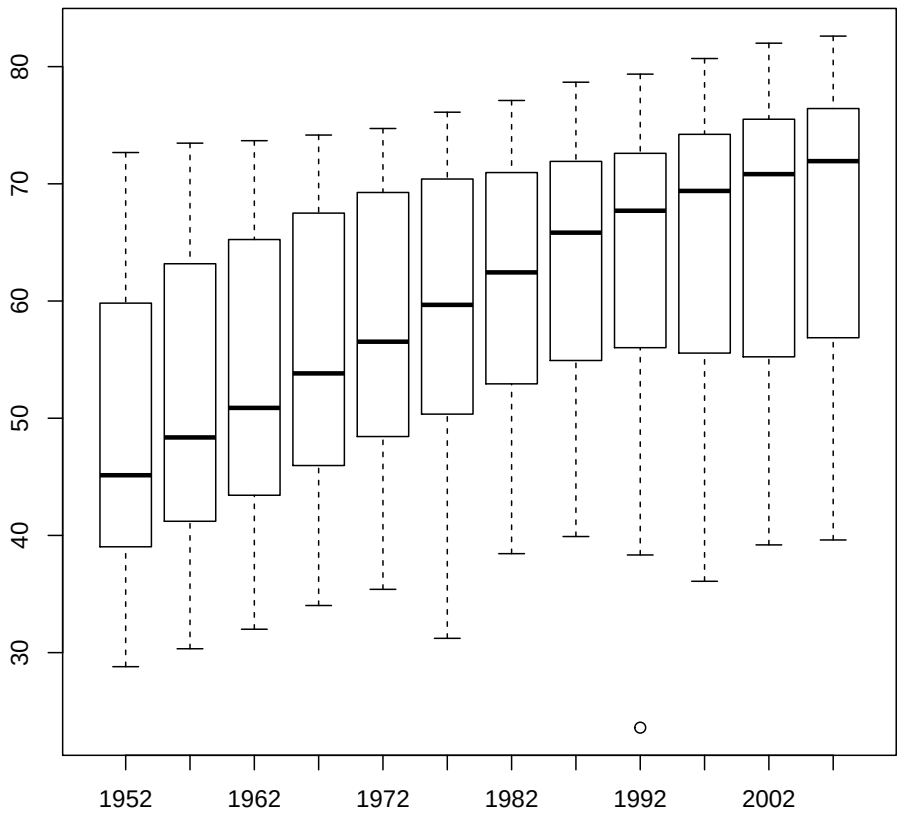
Plotting data frames

Since visualization of our data is a critical step before starting any type of low volume statistical analyses, we will do some simple plots using base R graphics.

There are other advanced graphical functions like `ggplot` but we won't be talking about this today.

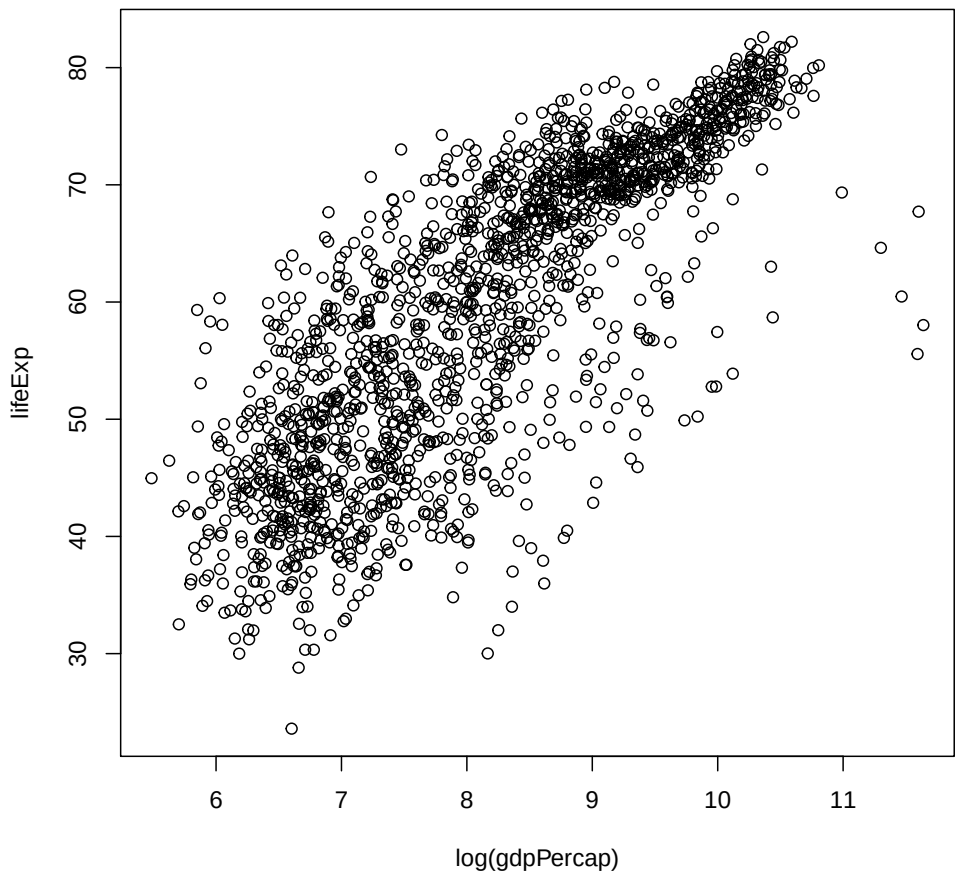
Plotting data frames

```
boxplot(lifeExp ~ year, data = gDat)
```



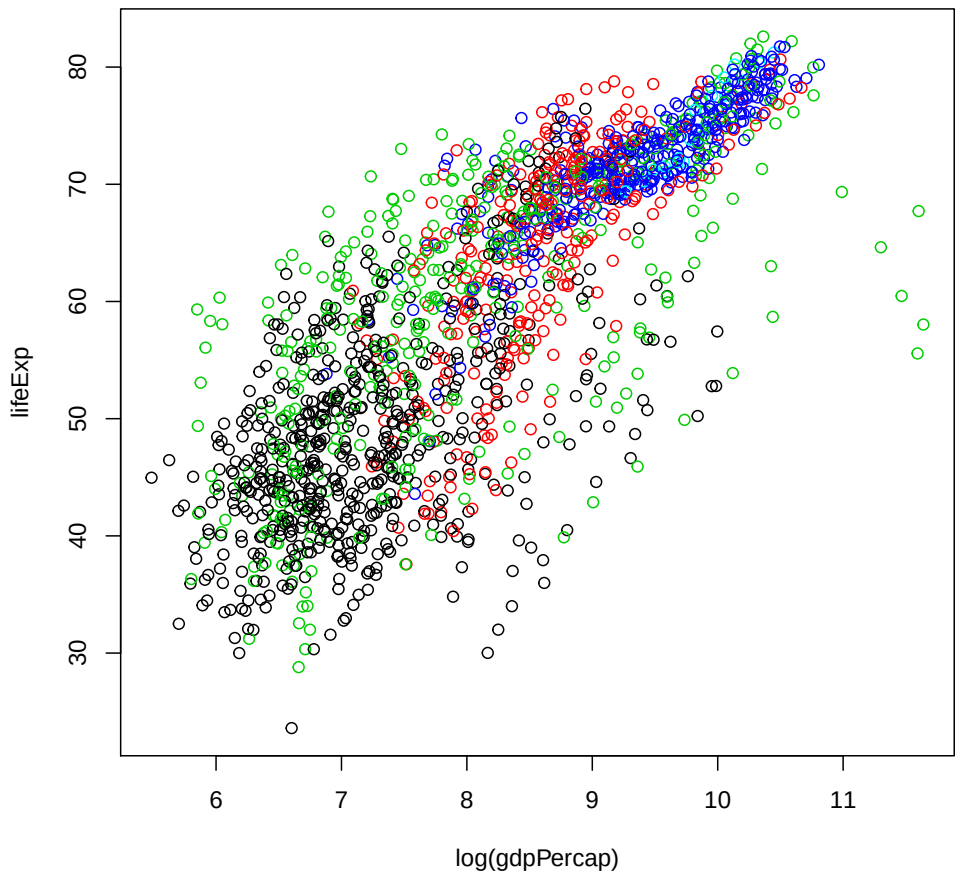
Plotting data frames

```
plot(lifeExp ~ log(gdpPercap), data = gDat)
```



Plotting data frames

```
plot(lifeExp ~ log(gdpPercap), col = continent, data = gDat)
```



Exporting graphical output

To export graphical output you can use the following approaches:

- use the `Export` button on the figure panel in RStudio
- use the following commands (one example of many):

```
pdf(file = "lifeExp_vs_gdp.pdf")  ## can be replaced by jpeg, png,...
plot(lifeExp ~ log(gdpPercap), col = continent, data = gDat)
dev.off()
```

Workspace

When you quit R/Rstudio, you'll always get a prompt like this:

```
Save workspace image to ~/.Rdata?
```

The workspace is your current R working environment and includes all objects you created. You can choose to save an image of the current workspace that is automatically reloaded the next time R is started.

Note: As a beginner, it is okay to save your workspace, but I hope you can consider saving your R script instead. You can always recreate all the objects using the same script.

Working directory

The working directory is the place where the R console looks for files when R initially starts up. This is also where R, by default, will save files to.

You can explicitly check your working directory with:

```
getwd()
```

It is also displayed at the top of the RStudio console.

Sometimes you would like to have your working directory set to be the same place as where your R script locates. RStudio does that!! (i.e., when you open a .R file using RStudio) Otherwise, you need to use `setwd(dir)`...

Other stuff: R objects and "types"

`Data.frame` objects are simply a list of "variables" (columns in the data matrix) that are formally called "atomic vectors" (fancy term but you don't need to really know).

It is important to learn a little about atomic vectors, especially their types, because all functions in R will not work properly if the input data have the wrong type. For example, plotting and summarizing categorical data and numerical data can be different!

Other stuff: R objects and "types"

Informally speaking atomic vectors usually come in the following "types":

- character: variable of "strings" (or "words")
- logical: binary variable that only takes 0 and 1
- numeric: quantitative variable
- factor: categorical variable with a **pre-defined** and **fixed** list of categories

Note: Always always double check the variables "types" in your data by using `str`.
Note: Dealing with factors can be a pain. Master it! They are useful data types in analyses where you have both numeric and categorical information. (We will talk about this more, later)

Other stuff: Packages

Sometimes you will be using some functions that do not belong to the default packages that come with your R installation. For example, I want to do forecasting for time series models using the package `forecast`.

To install an R package you can either do it through `Tools` on the menu bar in RStudio or type the following on the console:

```
install.packages("forecast")
```

Every time you wish to use a package that you have installed, you must load it into your library:

```
library(forecast)
```

Other stuff: Functions and documentation, getting help

There are millions of R functions out there. It is fine that you don't know how to use most of them! But it is not fine that you don't know how to look for its documentation/manual! (a good package should have good documention for its functions)

Some good general advice:

- Use the `help` panel in RStudio or use the following commands:

```
?function
```

  Then try out the examples provided in the documentation *(this is what I usually do first, even before checking its arguments)*.
- Google! *(am I good in R coding, or just a good Googler?)*

Reference

- "Programming with R" by Software Carpentry, http://software-carpentry.org/v5/novice/r/index.html
- STAT545 lecture notes by Prof Jenny Bryan, http://stat545-ubc.github.io/topics.html. See topics:
  - Basic R and RStudio, workspace, working directory, RStudio Project
  - Basic care and feeding of data in R
  - R objects (beyond data.frames) and indexing

R style guide

"Good coding style is like using correct punctuation. You can manage without it, but it sure makes things easier to read." *Hadley Wickham*

R style guide

Pick a coding style and stick to it! There are plenty of coding styles out there. Here is one from Hadley Wickham I recommend: http://adv-r.had.co.nz/Style.html

Summary

1. File names: end in `.R`
2. Object names: `variable.name` or `variable_name` (be concise but meaningful)
3. Spacing: Spaces around operators like `=`, `+`, `-`, `<-`, etc.
4. Indentation: two spaces, no tabs
5. Line length: Max 80 characters
6. Assignment: Use `<-` not `=`
7. Commenting: `# Comment on why not the what`

References

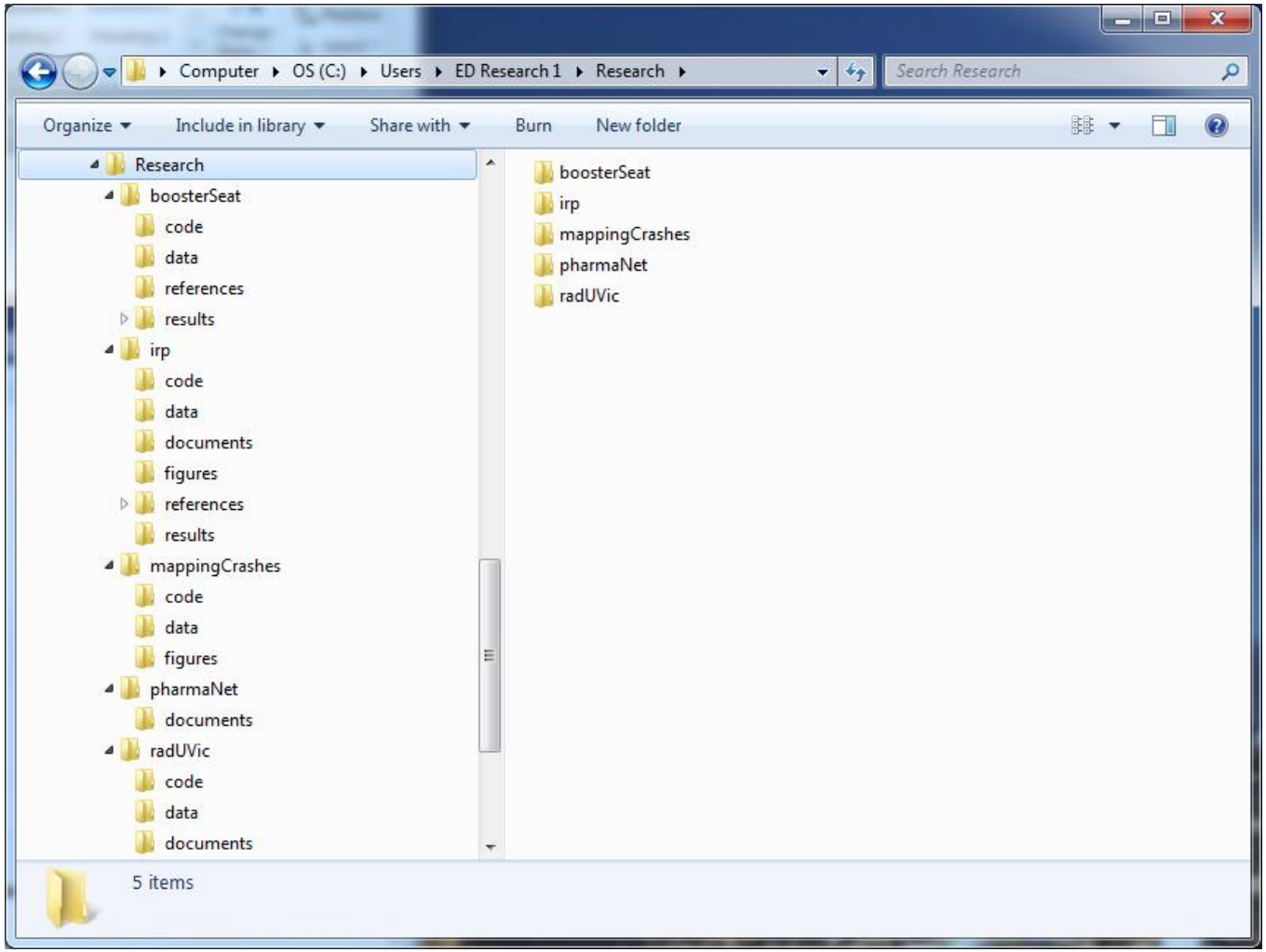
- Advanced R by Hadley Wickham – http://adv-r.had.co.nz/Style.html
- Google's R Style Guide – http://google-styleguide.googlecode.com/svn/trunk/Rguide.xml

Project organization

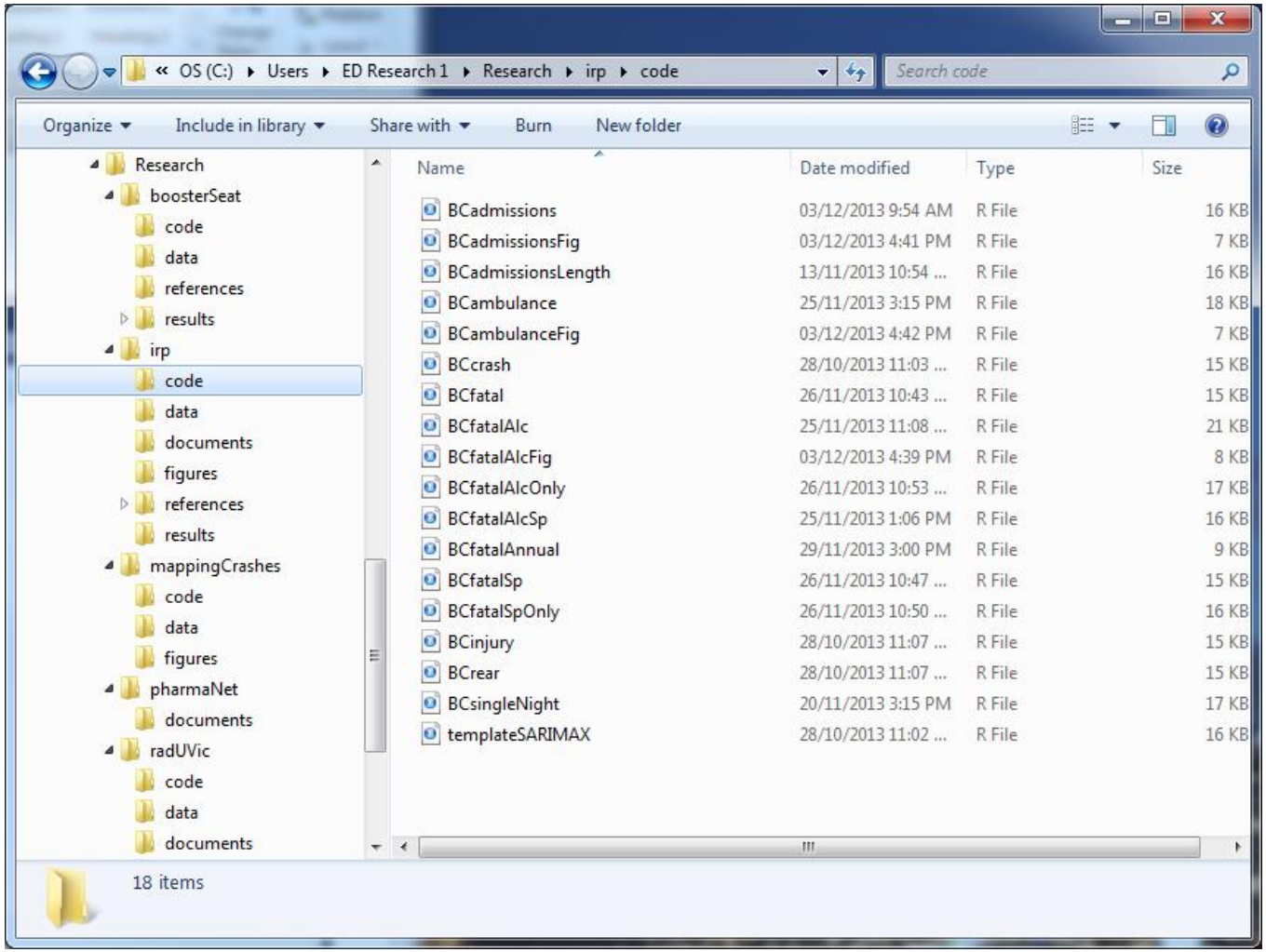
Organizing your workflow will make your life (and your colleague's) easier! Most projects are likely to have the following components:

- code
- data
- documents
- figures
- references
- results

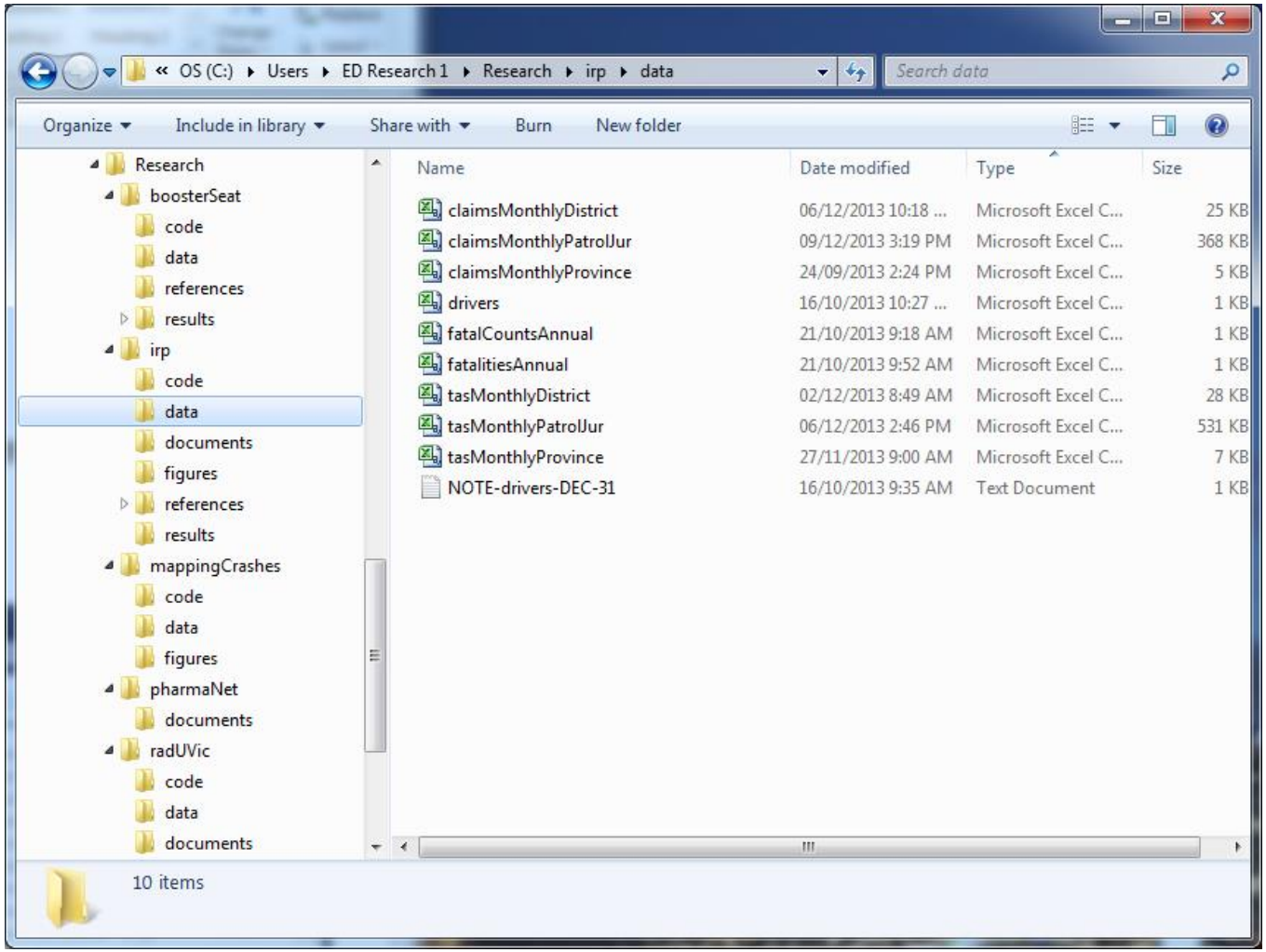
Directories



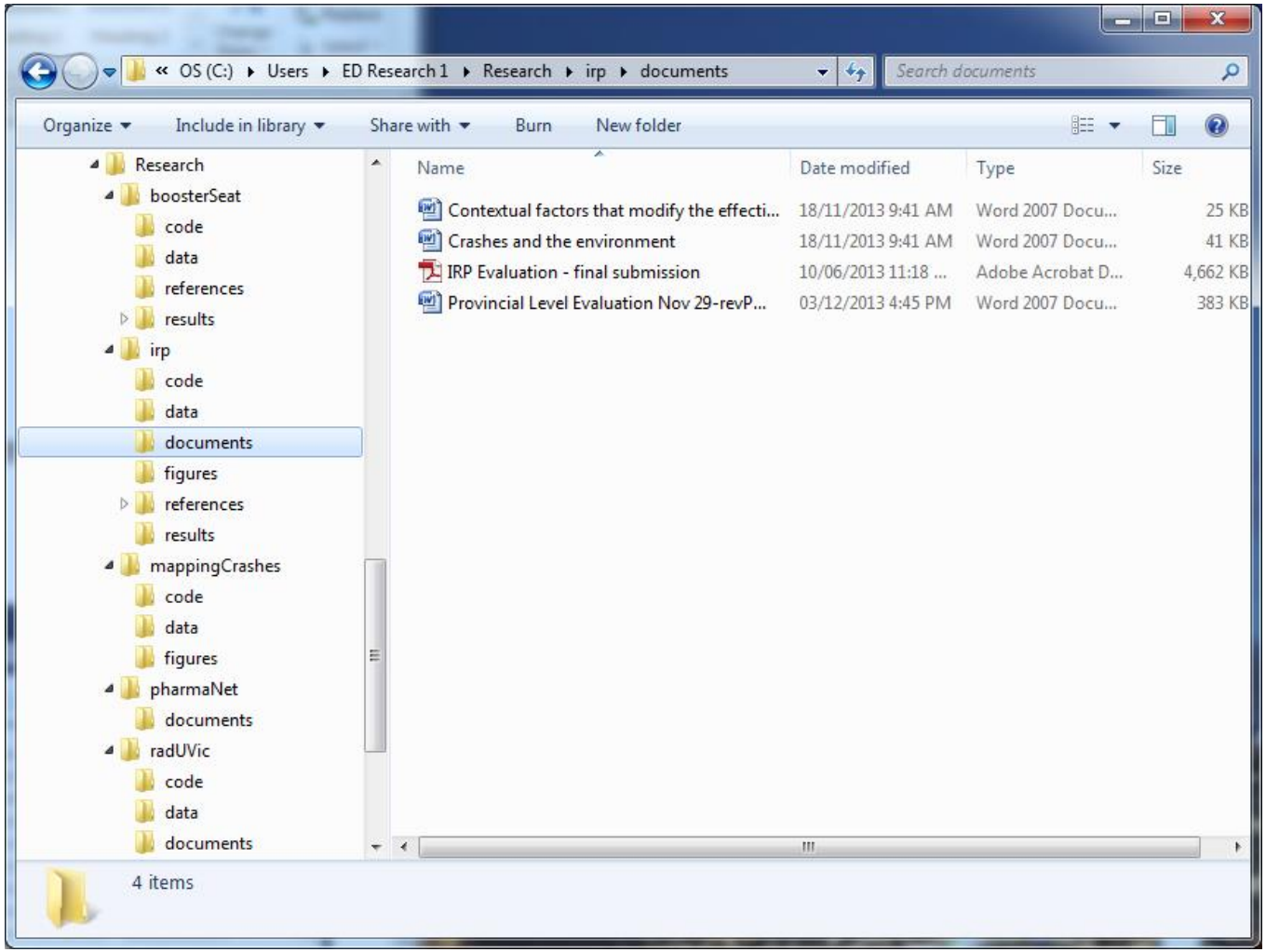
Code


« OS (C:) ▸ Users ▸ ED Research 1 ▸ Research ▸ irp ▸ code
Search code
Organize
Include in library
Share with
Burn
New folder
Research
boosterSeat
code
data
references
results
irp
code
data
documents
figures
references
results
mappingCrashes
code
data
figures
pharmaNet
documents
radUVic
code
data
documents
Name
Date modified
Type
Size
BCadmissions 03/12/2013 9:54 AM R File 16 KB
BCadmissionsFig 03/12/2013 4:41 PM R File 7 KB
BCadmissionsLength 13/11/2013 10:54 ... R File 16 KB
BCambulance 25/11/2013 3:15 PM R File 18 KB
BCambulanceFig 03/12/2013 4:42 PM R File 7 KB
BCcrash 28/10/2013 11:03 ... R File 15 KB
BCfatal 26/11/2013 10:43 ... R File 15 KB
BCfatalAlc 25/11/2013 11:08 ... R File 21 KB
BCfatalAlcFig 03/12/2013 4:39 PM R File 8 KB
BCfatalAlcOnly 26/11/2013 10:53 ... R File 17 KB
BCfatalAlcSp 25/11/2013 1:06 PM R File 16 KB
BCfatalAnnual 29/11/2013 3:00 PM R File 9 KB
BCfatalSp 26/11/2013 10:47 ... R File 15 KB
BCfatalSpOnly 26/11/2013 10:50 ... R File 16 KB
BCinjury 28/10/2013 11:07 ... R File 15 KB
BCrear 28/10/2013 11:07 ... R File 15 KB
BCsingleNight 20/11/2013 3:15 PM R File 17 KB
templateSARIMAX 28/10/2013 11:02 ... R File 16 KB
18 items

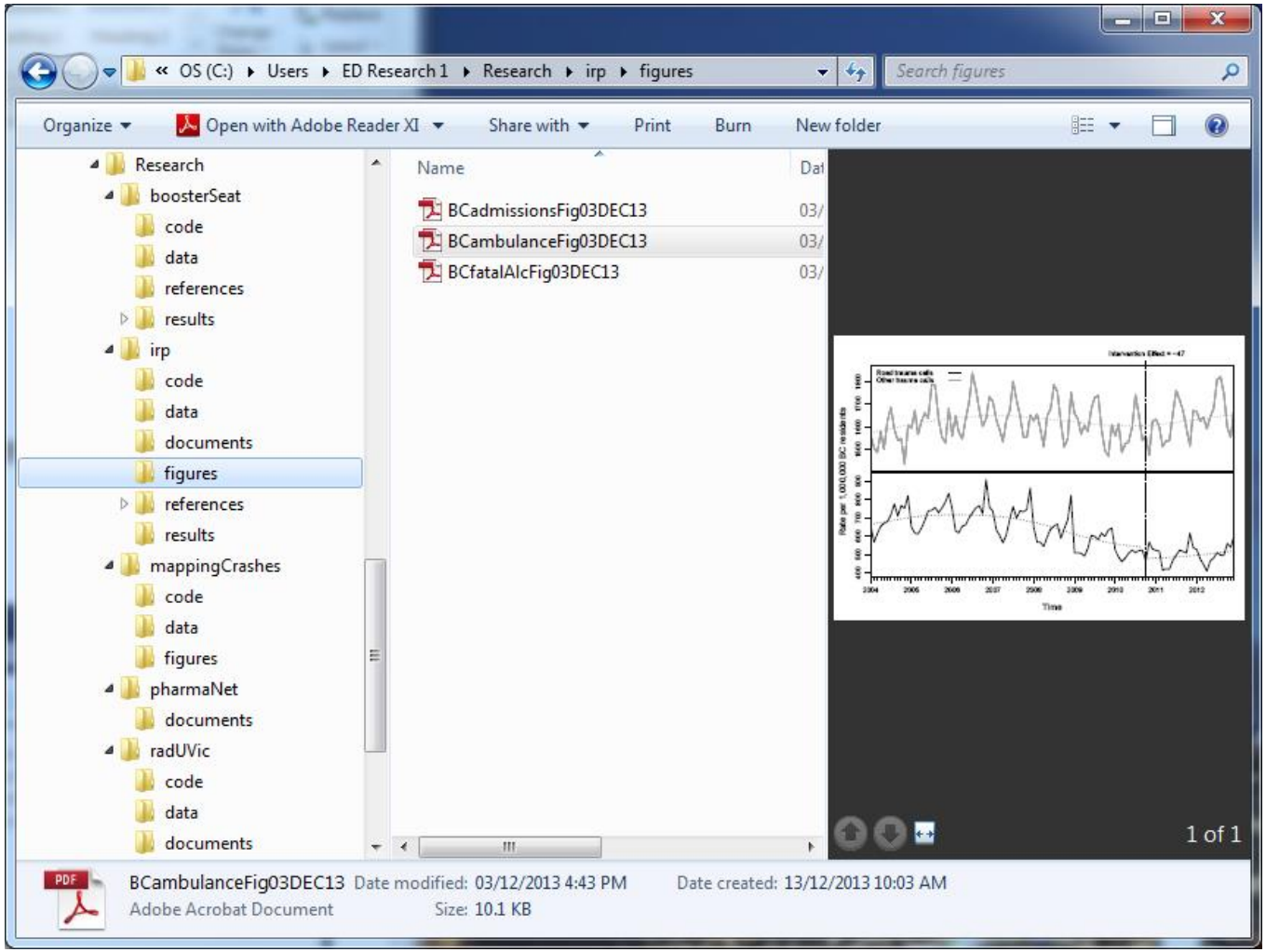
Data


« OS (C:) ▸ Users ▸ ED Research 1 ▸ Research ▸ irp ▸ data
Search data
Organize
Include in library
Share with
Burn
New folder
Research
boosterSeat
code
data
references
results
irp
code
data
documents
figures
references
results
mappingCrashes
code
data
figures
pharmaNet
documents
radUVic
code
data
documents
Name
Date modified
Type
Size
claimsMonthlyDistrict 06/12/2013 10:18 ... Microsoft Excel C... 25 KB
claimsMonthlyPatrolJur 09/12/2013 3:19 PM Microsoft Excel C... 368 KB
claimsMonthlyProvince 24/09/2013 2:24 PM Microsoft Excel C... 5 KB
drivers 16/10/2013 10:27 ... Microsoft Excel C... 1 KB
fatalCountsAnnual 21/10/2013 9:18 AM Microsoft Excel C... 1 KB
fatalitiesAnnual 21/10/2013 9:52 AM Microsoft Excel C... 1 KB
tasMonthlyDistrict 02/12/2013 8:49 AM Microsoft Excel C... 28 KB
tasMonthlyPatrolJur 06/12/2013 2:46 PM Microsoft Excel C... 531 KB
tasMonthlyProvince 27/11/2013 9:00 AM Microsoft Excel C... 7 KB
NOTE-drivers-DEC-31 16/10/2013 9:35 AM Text Document 1 KB
10 items

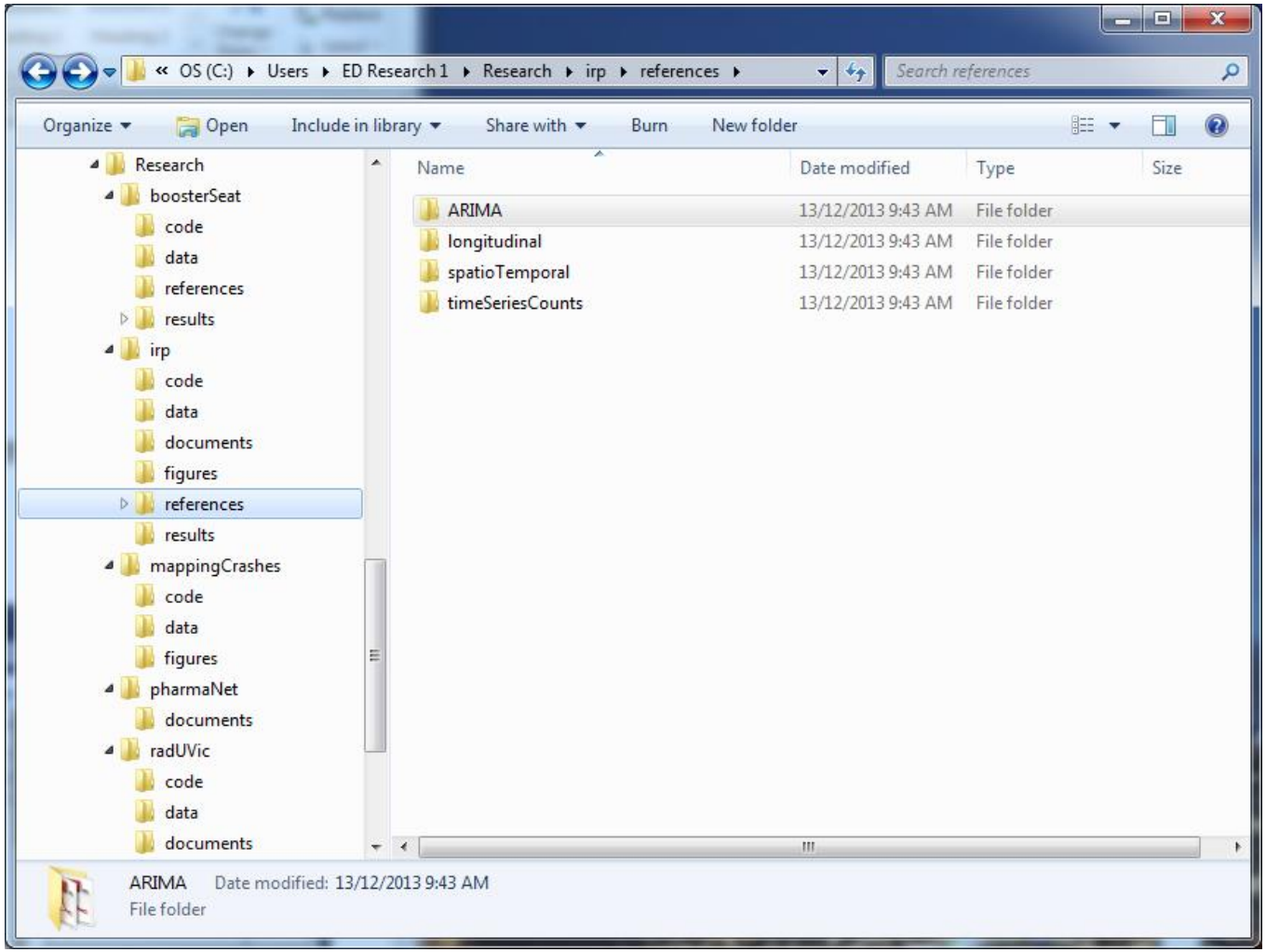
Documents


« OS (C:) ▸ Users ▸ ED Research 1 ▸ Research ▸ irp ▸ documents
Search documents
Organize
Include in library
Share with
Burn
New folder
Research
boosterSeat
code
data
references
results
irp
code
data
documents
figures
references
results
mappingCrashes
code
data
figures
pharmaNet
documents
radUVic
code
data
documents
Name
Date modified
Type
Size
Contextual factors that modify the effecti... 18/11/2013 9:41 AM Word 2007 Docu... 25 KB
Crashes and the environment 18/11/2013 9:41 AM Word 2007 Docu... 41 KB
IRP Evaluation - final submission 10/06/2013 11:18 ... Adobe Acrobat D... 4,662 KB
Provincial Level Evaluation Nov 29-revP... 03/12/2013 4:45 PM Word 2007 Docu... 383 KB
4 items

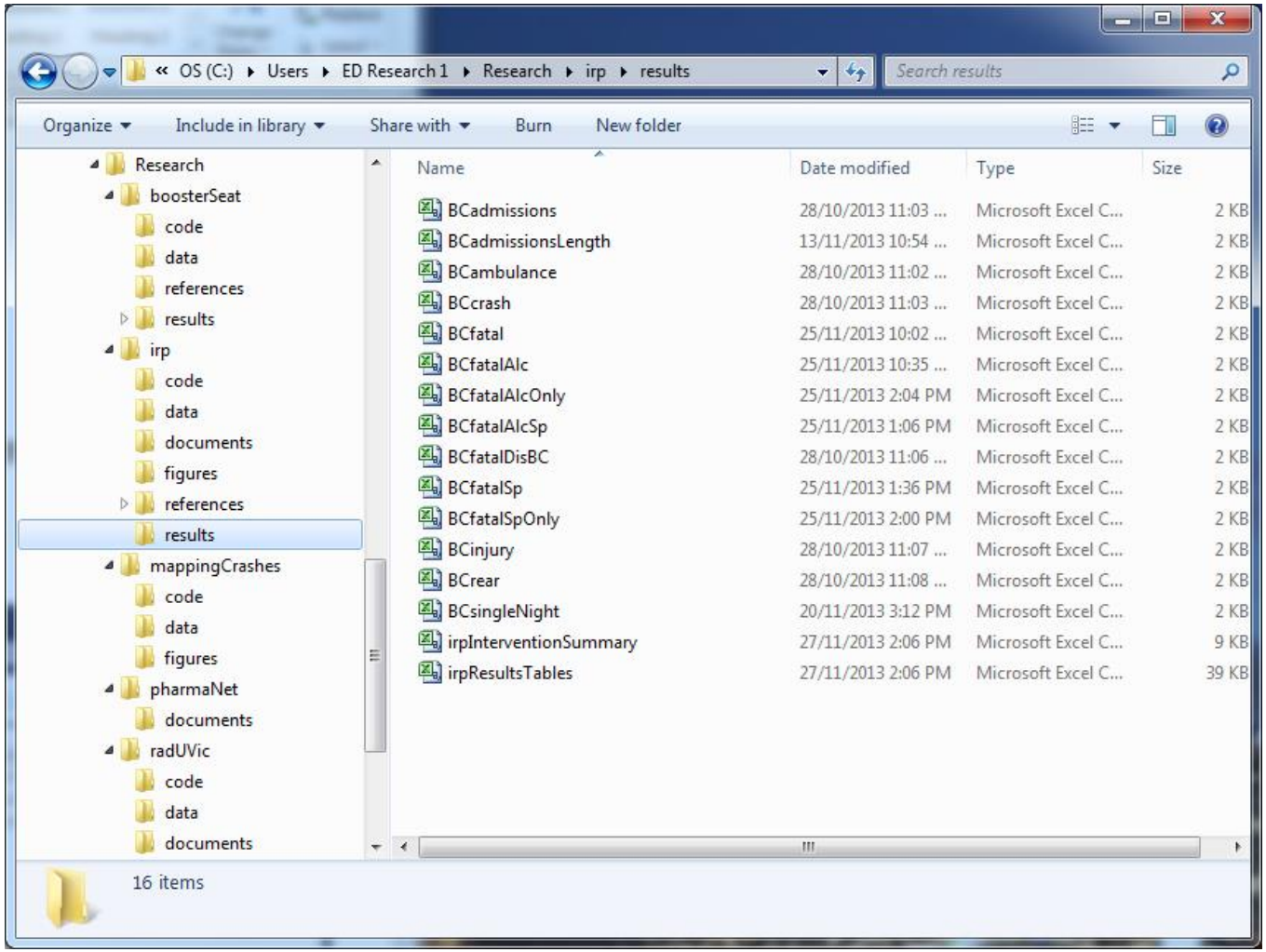
Figures


OS (C:) ▸ Users ▸ ED Research 1 ▸ Research ▸ irp ▸ figures
Search figures
Organize
Open with Adobe Reader XI
Share with
Print
Burn
New folder
Research
boosterSeat
code
data
references
results
irp
code
data
documents
figures
references
results
mappingCrashes
code
data
figures
pharmaNet
documents
radUVic
code
data
documents
Name
BCadmissionsFig03DEC13
BCambulanceFig03DEC13
BCfatalAlcFig03DEC13
1 of 1
BCambulanceFig03DEC13 Date modified: 03/12/2013 4:43 PM Date created: 13/12/2013 10:03 AM
Adobe Acrobat Document Size: 10.1 KB

References


« OS (C:) ▸ Users ▸ ED Research 1 ▸ Research ▸ irp ▸ references ▸
Search references
Organize
Open
Include in library
Share with
Burn
New folder
Research
boosterSeat
code
data
references
results
irp
code
data
documents
figures
references
results
mappingCrashes
code
data
figures
pharmaNet
documents
radUVic
code
data
documents
Name
Date modified
Type
Size
ARIMA 13/12/2013 9:43 AM File folder
longitudinal 13/12/2013 9:43 AM File folder
spatioTemporal 13/12/2013 9:43 AM File folder
timeSeriesCounts 13/12/2013 9:43 AM File folder
ARIMA Date modified: 13/12/2013 9:43 AM
File folder

Results


« OS (C:) ▸ Users ▸ ED Research 1 ▸ Research ▸ irp ▸ results
Search results
Organize
Include in library
Share with
Burn
New folder
Research
boosterSeat
code
data
references
results
irp
code
data
documents
figures
references
results
mappingCrashes
code
data
figures
pharmaNet
documents
radUVic
code
data
documents
Name
Date modified
Type
Size
BCadmissions 28/10/2013 11:03 ... Microsoft Excel C... 2 KB
BCadmissionsLength 13/11/2013 10:54 ... Microsoft Excel C... 2 KB
BCambulance 28/10/2013 11:02 ... Microsoft Excel C... 2 KB
BCcrash 28/10/2013 11:03 ... Microsoft Excel C... 2 KB
BCfatal 25/11/2013 10:02 ... Microsoft Excel C... 2 KB
BCfatalAlc 25/11/2013 10:35 ... Microsoft Excel C... 2 KB
BCfatalAlcOnly 25/11/2013 2:04 PM Microsoft Excel C... 2 KB
BCfatalAlcSp 25/11/2013 1:06 PM Microsoft Excel C... 2 KB
BCfatalDisBC 28/10/2013 11:06 ... Microsoft Excel C... 2 KB
BCfatalSp 25/11/2013 1:36 PM Microsoft Excel C... 2 KB
BCfatalSpOnly 25/11/2013 2:00 PM Microsoft Excel C... 2 KB
BCinjury 28/10/2013 11:07 ... Microsoft Excel C... 2 KB
BCrear 28/10/2013 11:08 ... Microsoft Excel C... 2 KB
BCsingleNight 20/11/2013 3:12 PM Microsoft Excel C... 2 KB
irpInterventionSummary 27/11/2013 2:06 PM Microsoft Excel C... 9 KB
irpResultsTables 27/11/2013 2:06 PM Microsoft Excel C... 39 KB
16 items

Exercise

Create directories and perform the analysis for the following project:

- Project: Black Cherry Trees
- Goal: Describe the relationship between cherry tree diameter (girth) and height using a graph
- Input: Measurements of the girth, height, and volume of timber in 31 felled black cherry trees
- Output: An appropriate graphical summary of tree diameter for short and tall trees

```
head(trees, n = 3)  ## the R data set is called trees
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
```

Some Useful R Commands

- Assign a name to an object with <-. eg. x <- 5
- Create a sequence of equally spaced numbers: seq(from,to,by)
- Create a matrix: matrix(vector, nrow, ncol)
- Extract a variable from a data set: dataset$variable
- Plotting: boxplot(x∼y), plot(x,y), hist(x)

Confused about a function? Type ? then the name of the function to see the help menu. eg. ?plot