# Introduction to R

## SETAC Workshop

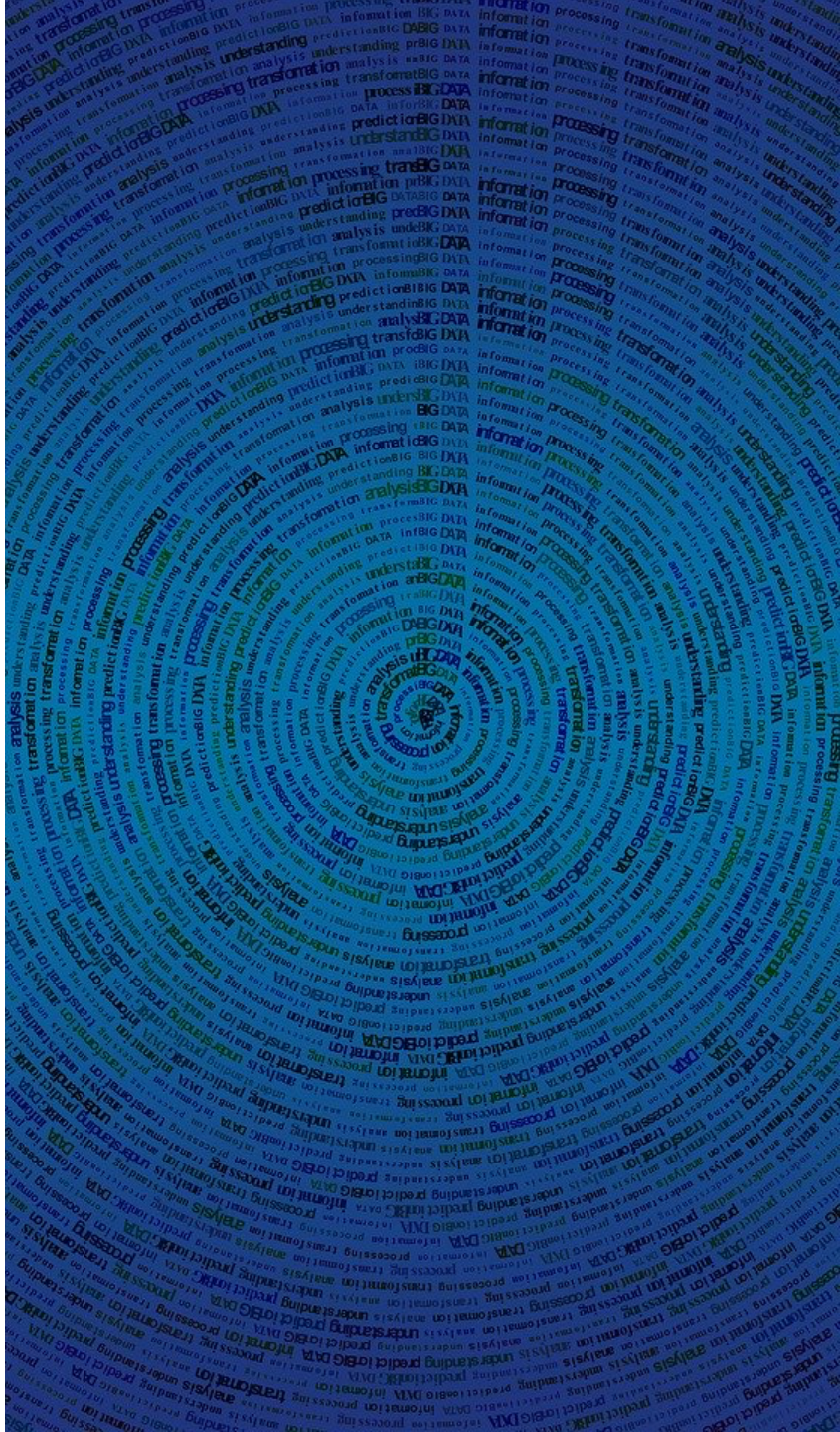
## November 15th, 2020

**Dr. Kristin Eccles**

kristin.eccles@utoronto.ca

@kristineccles

https://github.com/kristineccles/setac_intro_to_r_2020



# Overview

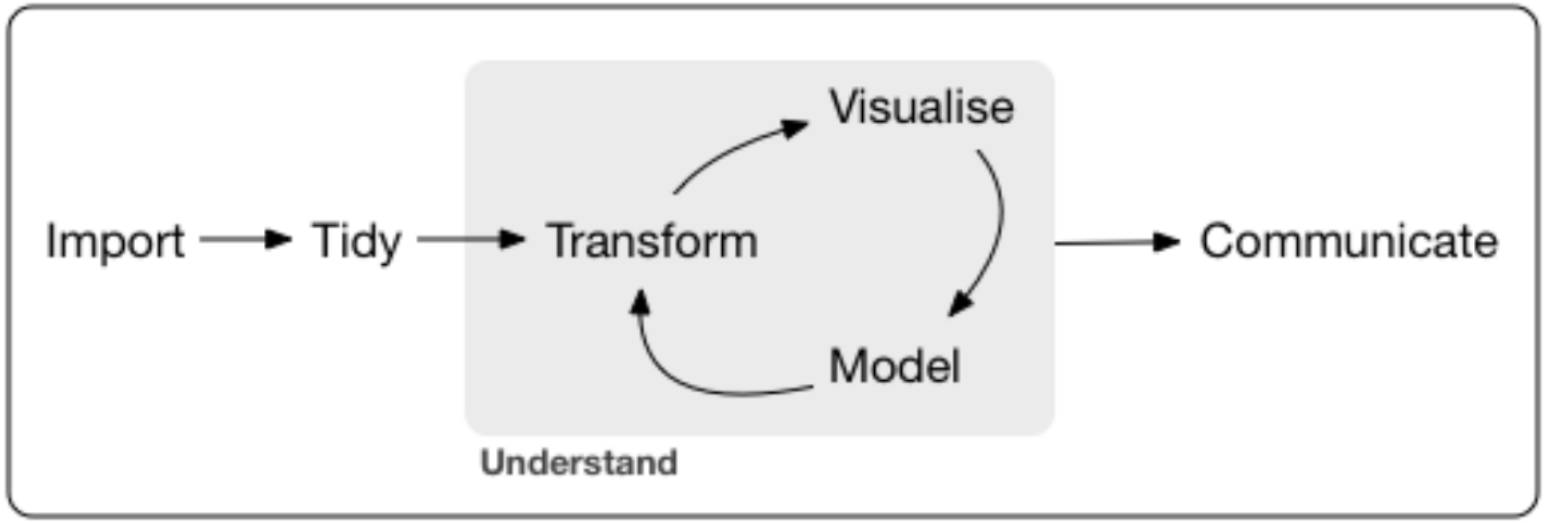
- Why and when would you want to use R?
- How to get help
- Intro to RStudio
- Core Language
  - Data types, functions, operations, loading data, saving data
  - R Packages (installing, loading, using)
- Exploratory Data Analysis
  - Graphs (base, ggplot2)
- Statistics
  - Inferential statistics (t-test)
  - Relational Statistics (linear regression)

## Schedule

| Time | Item |
|---|---|
| 8-10:15 CST | Into to R and Guided R examples |
| 10:15-10:30 CST | Break |
| 10:30-12:00 CST | Appy your knowledge- R workbook |

# What is R?

- "R is a free software environment for statistical computing and graphics" - http://www.r-project.org




- Why the name "R"?
  - First letter of two originators: Ross Ihaka and Robert Gentleman
  - Built on a earlier language called "S"
  - (S-Plus)

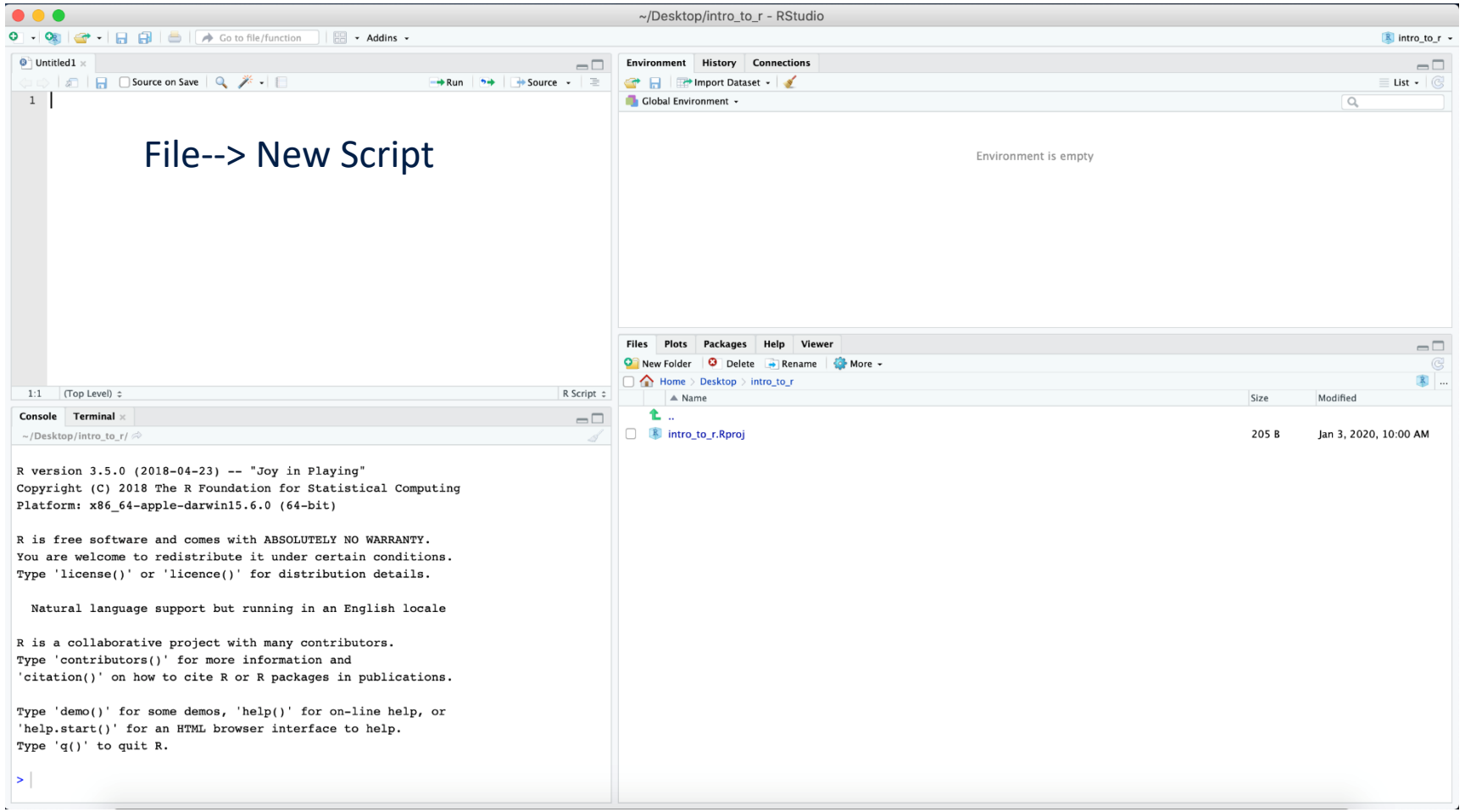
# Why use R?

- R is free to use
- R is **open source**
  - "denoting software for which the original source code is made freely available and may be redistributed and modified."
- Runs on all operating systems (Windows, OSX, Linux)
- R is very versatile
  - huge library of user-contributed packages (over 6,000 on Comprehensive R Archive Network (CRAN))
- Facilitates reproducible research
- Popular in academia and industry
  - A lot of free online resources (stack overflow, r stats, etc.)

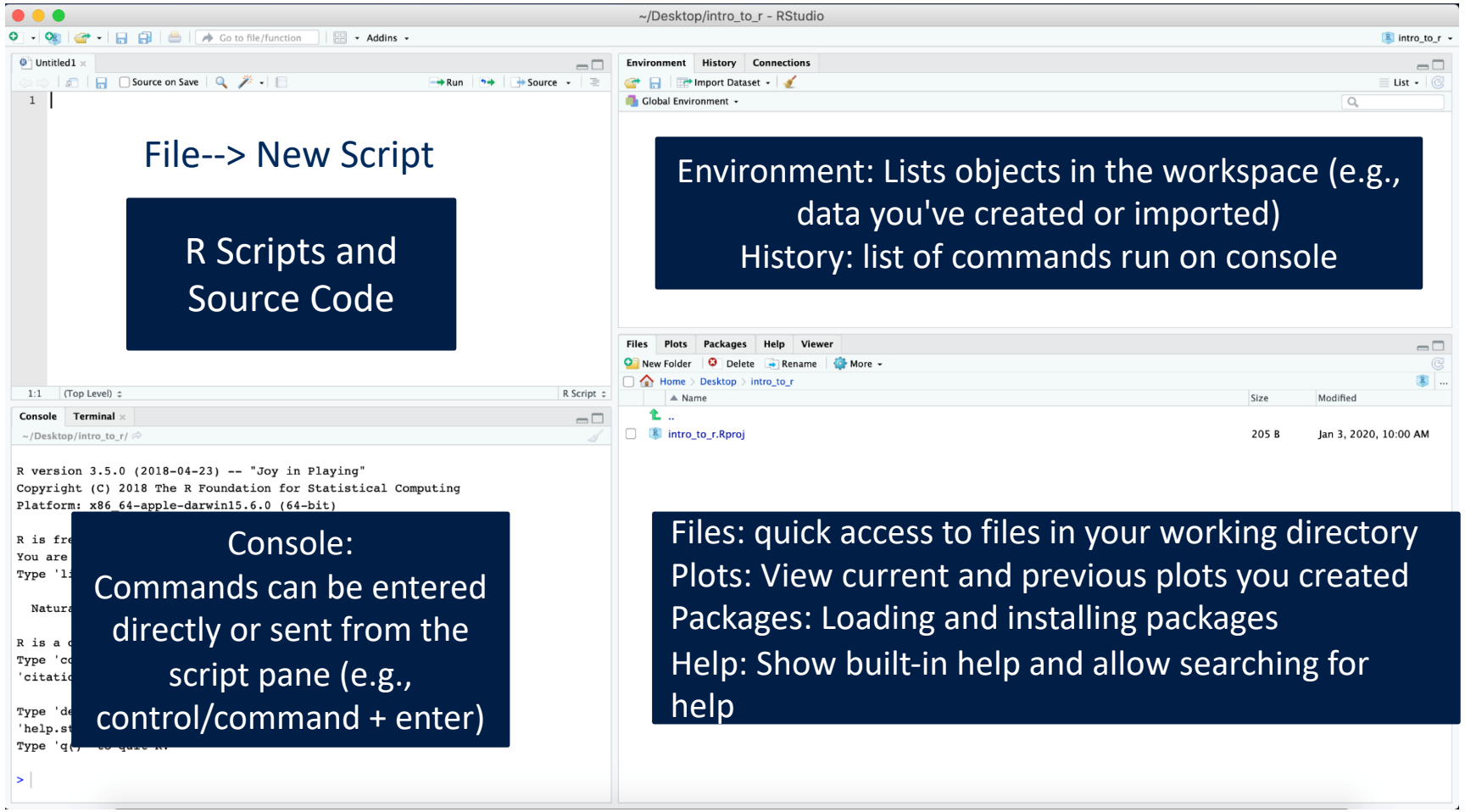
# Challenges of using R

- R involves writing scripts
- It does not have a GUI like SPSS, SAS, Stata, etc.
  - But RStudio is a user friendly Integrated Developer Environment (IDE)
- R is more interactive
  - In SPSS and SAS you choose a command and get piles of output which you wade through
  - R is a conversation: You interactively request relevant output

# A Guided Tour of RStudio



# A Guided Tour of RStudio



# RStudio Projects

- It is good practice to store all files related to a particular analysis project in a single directory on your computer
  - I.e. scripts, data files, configuration files, figures, exported tables, etc.
- Rstudio makes this easy to do
  - (Go to: `File → New Project → New Directory → New Project → Create Project`)
  - The directory name:  want to call it
  - Create project of a subdirectory of:  where on your computer you want it stored
- This generates a folder and a file with an "Rproj" extension (e.g., `projectname.Rproj`)
  - In the future, double click on this file to open the project
  - R studio will open the previous working environment

# Overview of common file extensions

- R Specific file formats
  - .r : R script files
  - .rmd : RMarkdown files
  - .Rproj : RStudio project files
  - .rdata : Native format in r for saving R objects
- Other relevant formats
  - .md : Markdown file
  - .csv : comma separated value data file

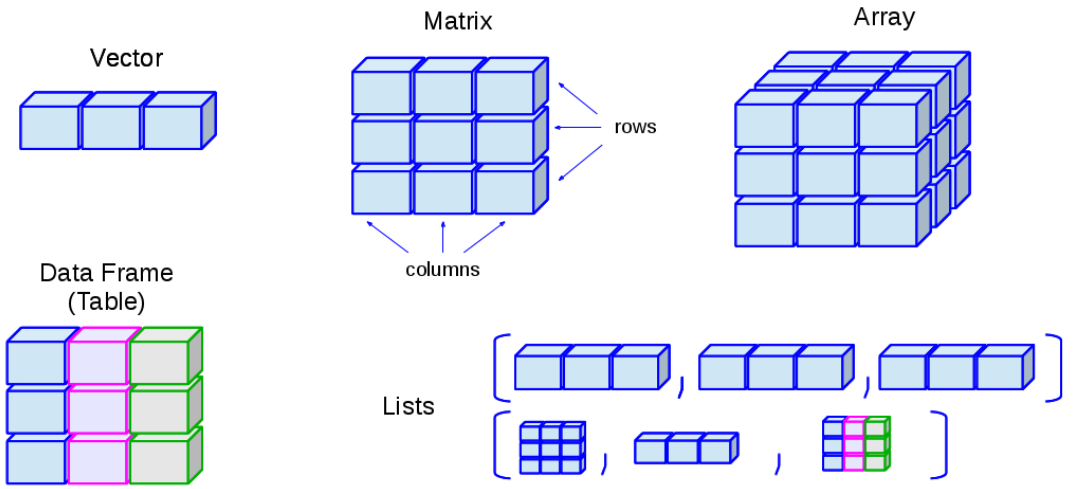
# Objects and Classes

- R is an object-oriented language
- Everything in R is an object: functions, symbols, and even R expressions.
- Objects may have attributes, such as name, dimension, and class R is an object-oriented language
  - Every object in R has a type
  - Every object in R is a member of a class
  - i.e. vectors, numeric vectors, data frames, lists, and arrays
- All R code manipulates objects

For more reading: Chapter 7 and 10 of R in a Nutshell

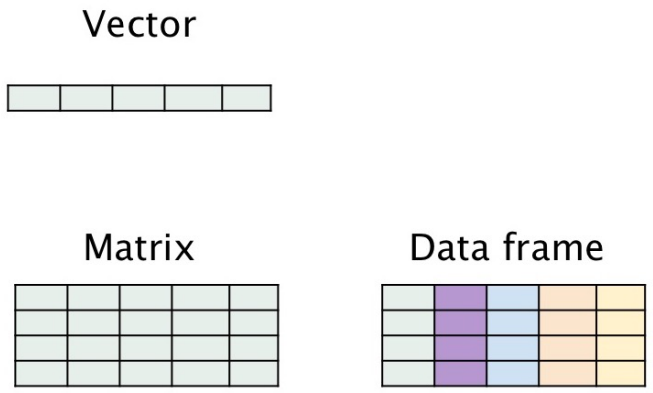
# Data structure and types in R

- Data structures: vectors, matrices, arrays, data frames (similar to tables, and lists




- Data types: integer, numeric, logical, and factor

| Variables | Example |
|---|---|
| integer | 100 |
| numeric | 0.05 |
| character | “hello” |
| logical | TRUE |
| factor | “Green” |



# Introducing R Commands

OPEN R

- R is an interpreted language
- Accessed through a command-line interpreter
  - This requires the user knowledge of commands and their parameters, and the syntax of the language
- Upon starting there is a “>” in the console. R is prompting you to type something, so this is called a prompt.
- The commands that you type into the console are called expressions

For more reading: R in a Nutshell (Chapter 2 and 3)

# Vectors

- Vectors are one dimensional sequences of values
- In R, any number that you enter in the console is interpreted as a vector (numeric or character).
- A vector is an ordered collection of numbers.
  - The “[1]” means that the index of the first item displayed in the row is 1.

```
> # Basic Operations
> 2 + 2 # addition
[1] 4
> 3 - 5 # subtraction
[1] -2
> 3 * 2 # multiplication
[1] 6
> (2 + 2)^(3 / 3.5) # exponents and brackets
[1] 3.281341
```

```
> "Hello world."
[1] "Hello world."
> #This is called a character vector in R.
> c("Hello world", "Hello R interpreter")
[1] "Hello world"          "Hello R interpreter"
```

# Functions

- Functions are the workhorses of R
- They take arguments as inputs and return objects as outputs.
- May modify objects in the environment or cause effects outside the R environment
  - I.e. plotting graphics, saving files, or sending data over the network.
- Functions provide information about vectors
- There are probably hundreds of thousands of functions in R.
- E.g.
  - length(x), mean(x), sd(x)

For more reading: Chapter 5 R in a Nutshell

# Indexing

- The $ sign is used to reference a column by name
  - df$teams
- Reference a column
  - df[,2:3]
- Reference a row
  - df[2:3,]
- Reference rows and columns
  - df[1:2,1:2]
- R functions work better on columns than rows
  - Try calculating the average of a column
  - How would calculate the average of a row?

```
> df
  teams wins loses
1   PHI   92    70
2   NYM   89    73
3   FLA   94    77
4   ATL   72    90
5   WSN   59   102
```

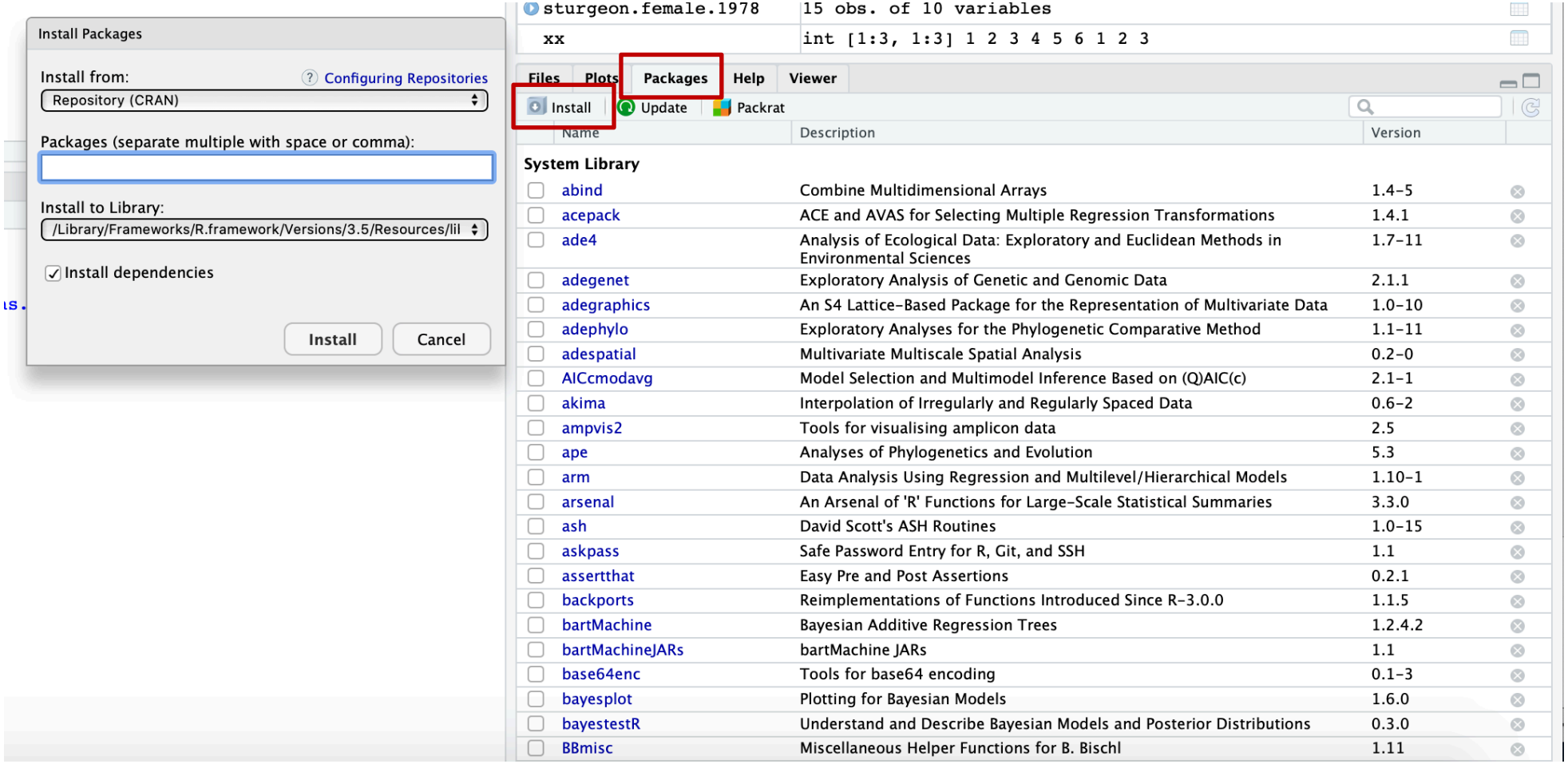
# Loading Packages

- A package is a related set of functions, help files, and data files that have been bundled together.
- Typically, all the functions in the package are related
  - i.e. the stats package contains functions for doing statistical analysis
- You first need to make sure that it has been installed into a local library
  - R comes with a number of different packages

*Table 4-1. Packages included with R*

| Package name | Loaded by default | Description |
|---|---|---|
| base | ✓ | Basic functions of the R language, including arithmetic, I/O, programming support |
| boot | | Bootstrap resampling |
| class | | Classification algorithms, including nearest neighbors, self-organizing maps, and learning vector quantization |
| cluster | | Clustering algorithms |
| codetools | | Tools for analyzing R code |
| compiler | | Byte code compiler for R |
| datasets | ✓ | Some famous data sets |
| foreign | | Tools for reading data from other formats, including Stata, SAS, and SPSS files |
| graphics | ✓ | Functions for base graphics |
| grDevices | ✓ | Device support for base and grid graphics, including system-specific functions |
| grid | | Tools for building more sophisticated graphics than the base graphics |
| KernSmooth | | Functions for kernel smoothing |
| lattice | | An implementation of Trellis graphics for R: prettier graphics than the default graphics |
| MASS | | Functions and data used in the book *Modern Applied Statistics with S* by Venables and Ripley; contains a lot of useful |

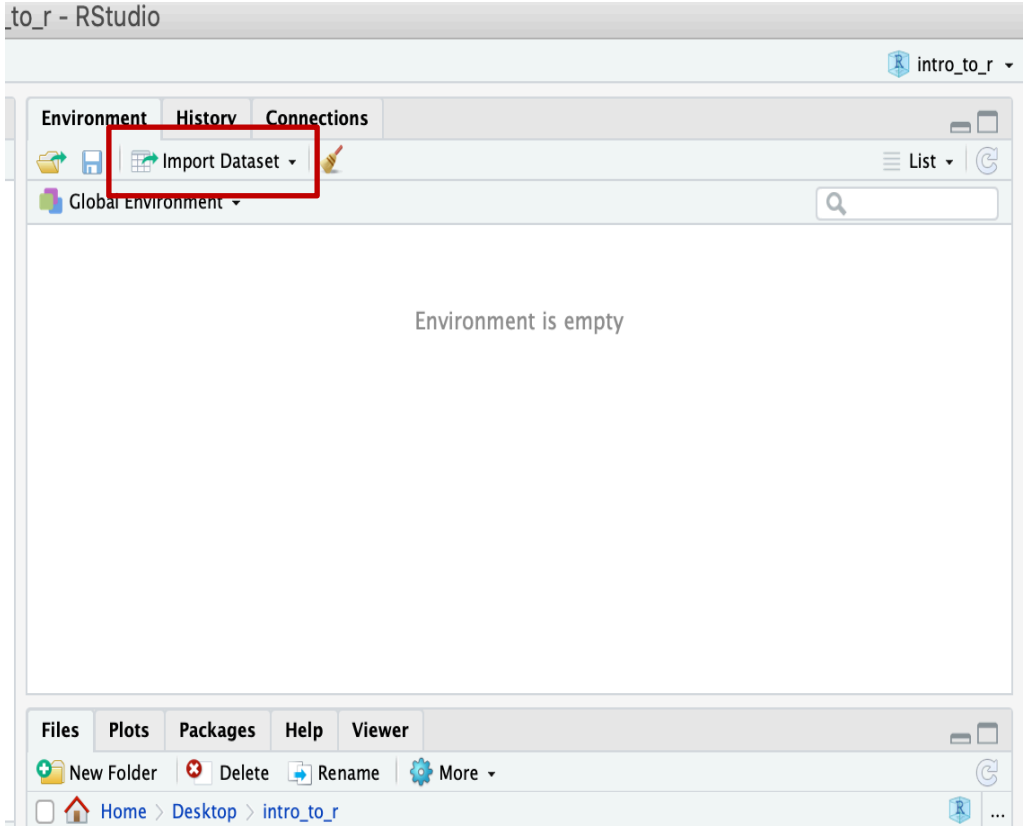
For more info see chapter 4 of R in a Nutshell




Or you can use the command install.packages(package)
Then you must call the library using library(package)

# Read data into R

- You can import a variety of data file types, including from other statistics programs like SPSS, Stata, SAS, Minitab
- Common file formats like .xlsx and .csv
- The easiest way to import data is using the Import Dataset button in the Environment window.
- Better to use the command read.csv()



# Useful Commands/ Tips

- To bring up help file,  in the command line type:
  - ?commandname (searches only installed packages) OR help(commandname)
  - ??commandname (searches whole CRAN repository)
    - i.e. ?ggplot or help(ggplot)
- R is always case sensitive
- Always use a script and work from the editor
- Save your own annotated copy of the script.
- The # symbol means that the line will not be executed in R (useful to annotate scripts)

# Descriptive Statistics

- Characteristics of a normal (Gaussian) distribution:
  - Mean, median, mode are all the same & in the middle of the distribution
- Symmetrical, smooth, continuous distribution
- Is my data normal?
  - Make a histogram
  - Look at mean, median, and modal values
  - Check the shape of the distribution:
    - Skewness and Kurtosis

**summary(df)**

- **no st.dev in summary**
- **sd(df)**

**library(psych)**

- **describe(df)**

# Inferential Statistics: Unpaired T-Test

- Objective: Test if there is a difference between the **mean** of two groups
  - Boxplots show the median
- Data type: one continuous and one categorical variable
- Assumptions: variable normality, equal variance between the groups
  - For unequal variance use Welch's T-Test

# Inferential Statistics: ANOVA

- Test: If there is a difference between the **mean** of 3 or more groups
  - This only tells you if there is a different NOT where the difference is
- Data: Three or more groups and a continuous variable
- Assumptions (F-test):variable normality, equal variance between the groups (ish)

# Inferential Statistics: ANOVA Post-hoc test (Tukey)

- Test: Multiple comparison to determine the differences in means after an ANOVA
- Data: Three or more groups and a continuous variable
- Assumptions (t-test):variable normality, equal variance between the groups
- Note the p adj: This p-value is adjusted for multiple comparison
  - By default R uses Bonferroni (α /n), where n= number of comparisons

# Relational Statistics: Correlation (Pearson Product-Moment)

- Test: Association between variables
  - correlation coefficient ρ (Greek letter rho)
- Data: Two continuous variables
  - Pearson product-moment correlation
  - Spearman rank and Kendall Tau are the non-parametric version
  - Contingency analysis (categorical data)
- Assumptions (t-test): variable normality, linear association between variables

# Relational Statistics: Linear Regression (Univariate)

- Test: the slope coefficient is significant
- Data: two continuous variables
  - Generalized linear model is used for non-parametric (e.g. logit)
- Assumptions (*t*-test): , Test statistic: *t*, Assumptions: Variable normality , Random/probabilistic sample of paired variables, Variables have a linear association

- Test: Goodness of fit ($R^2$)
  - test that the model predicts a significant amount of the variance in y
  - Coefficient of determination (not the same as ρ )
- Assumptions ( *F*-test): Variable normality ,Random/probabilistic sample of paired variables, Variables have a linear association

# Assumptions of the residuals

- e = y - Y-hat
- In linear regression we have assumptions on the residuals too
  - Normality (Shapiro-Wilk)
  - Linearity (RESET test)
  - Lack of serial autocorrelation (Durbin-Watson test)
  - Homogeneity (Breusch-Pagan test)
- Can use individual test or plot(lm1)

# Relational Statistics: Multivariate

- lm( y ~ v1 + v2 + v3…)
- Assumptions and test the same as univariate linear regression
- One additional
  - Multicollinearity
    - Cannot have a high correlation between independent variables
    - > +/- 0.70 it typically the threshold
    - Correlation matrices are helpful
  - Can test this using a variance inflation factor (VIF)
    - >5

# Reading Outputs

- Assumption tests

```
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   2  1.2814 0.282
      103
```

Pass ✓

Testing assumptions- does the data/ model meet the criteria for a non-parametric model

**We want p > 0.05-** we want to accept the null that the data is normal, linear, homoscedastic, etc.

But if we do fail the assumption tests we can use our judgement…is it normal enough

```
        Shapiro-Wilk normality test

data:  resid(lm2)
W = 0.97749, p-value = 0.05726
```

Pass ✓

```
> # homoscedasticity of the residuals
> bptest(lm2) #pass

        studentized Breusch-Pagan test

data:  lm2
BP = 18.549, df = 3, p-value = 0.0003388
```

Fail X

```
> # lack of serial autocorrelation in residuals
> durbinWatsonTest(lm2) #pass
 lag Autocorrelation D-W Statistic p-value
   1      0.08870798       1.80676   0.256
 Alternative hypothesis: rho != 0
> # linearity of residuals
> resettest(lm2) #pass

        RESET test

data:  lm2
RESET = 3.1741, df1 = 2, df2 = 105, p-value = 0.04587
```

Pass ✓

Fail X

# Reading outputs

- T-test

```
        Welch Two Sample t-test

data:  fklngth by sex
t = 3.8594, df = 85.684, p-value = 0.0002198
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.299322 7.184765
sample estimates:
mean in group FEMALE   mean in group MALE
            45.43530             40.69326
```

P<0.05 so we reject the null- there is a statistically significant difference between the two groups

By how much?

Difference between groups = 45.43 - 40.69

= 4.74

# Reading outputs

- ANOVA

P > 0.05 so we accept the null- there is no statistically significant difference between the groups

```
Analysis of Variance Table

Response: fklngth
                  Df  Sum Sq Mean Sq F value Pr(>F)
as.factor(year)    2   90.71  45.357  1.9656 0.1453
Residuals        103 2376.74  23.075
>
```

If p < 0.05 we would do a Tukey post-hoc follow up to determine where the difference is (remember ANOVA only tells you if there is a difference, not where it is)

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = fklngth ~ as.factor(year), data = sturgeon.female)

$`as.factor(year)`
                 diff       lwr       upr     p adj
1979-1978 -2.5911136 -6.027194 0.8449667 0.1769060
1980-1978 -2.7007874 -6.071661 0.6700862 0.1423109
1980-1979 -0.1096738 -2.511791 2.2924432 0.9935222
```

Group comparison

Confidence Interval around the difference

P-value is adjusted for multiple comparisons using Bonferroni method Alpha/# of combinations

In this case:
0.05/ 3= 0.0167

# Reading outputs

- Correlation

```
        Pearson's product-moment correlation

data:  sturgeon_complete$age and sturgeon_complete$fklngth
t = 16.761, df = 97, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8011121 0.9054703
sample estimates:
      cor
0.8621723
```

Confidence Interval around correlation

P < 0.05 so we reject the null- there is a statistically significant correlation between the two variables

Correlation ranges from -1 to 1
0.86 is a strong positive relationship between the two variables

# Reading outputs

- Linear Regression

```
Call:
lm(formula = y ~ x, data = dataset)
Residuals:
    Min      1Q  Median      3Q     Max
-212.04  -62.52   30.87   86.77  121.88
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   28.149    180.756   0.156  0.88010
x             39.057      9.172   4.258  0.00277 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 121.8 on 8 deg. of freedom
Multiple R-squared: 0.69, Adjusted R-squared: 0.65
F-statistic: 18.13 on 1 and 8 DF,  p-value: 0.002767
```

**Model coefficients** (aka parameters)

- cols are the coefficient **estimates**, the **standard error** of the coefficient, the ***t* value** for the coefficient & the ***p*-value** for this *t* value
- y-intercept (intercept)
- slope (Estimate)
- legend explaining level of significance codes (α=0.05)

# Reading outputs

- Linear Regression

```
Call:
lm(formula = y ~ x, data = dataset)
Residuals:
    Min      1Q  Median      3Q     Max
-212.04  -62.52   30.87   86.77  121.88
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   28.149    180.756   0.156  0.88010
x             39.057      9.172   4.258  0.00277 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 121.8 on 8 deg. of freedom
Multiple R-squared: 0.69, Adjusted R-squared: 0.65
F-statistic: 18.13 on 1 and 8 DF,  p-value: 0.002767
```

**Model coefficients** (aka parameters)

- cols are the coefficient **estimates**, the **standard error** of the coefficient, the ***t* value** for the coefficient & the ***p*-value** for this *t* value
- y-intercept (intercept)
- slope (Estimate)
- legend explaining level of significance codes (α=0.05)

# Reading outputs

- Linear Regression

```
Call:
lm(formula = y ~ x, data = dataset)
Residuals:
    Min      1Q  Median      3Q     Max
-212.04  -62.52   30.87   86.77  121.88
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   28.149    180.756   0.156  0.88010
x             39.057      9.172   4.258  0.00277 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 121.8 on 8 deg. of freedom
Multiple R-squared: 0.69, Adjusted R-squared: 0.65
F-statistic: 18.13 on 1 and 8 DF,  p-value: 0.002767
```

- Overall model performance
- Standard error of the residuals (SSE)
- $R^2$ and the adjusted $R^2$
- Use the adj. R2 since it is more conservative
- F statistic and the p-value (prob. of getting a bigger F)