

LEARNING THE LINGO: AN R NEWBIE'S PERSPECTIVE



KATE DODDS
PHD STUDENT MACQUARIE UNIVERSITY

...."hello world?"

?Kate... visual basic



Microsoft Visual Basic for Applications - techonthenet_2013.xlsm - [Module1 (Code)]

File Edit View Insert Format Debug Run Tools Add-Ins Window Help

Project - VBAProject

(General) CheckIfFileExists

```
' Example provided by techonthenet.com

Sub CheckIfFileExists()

    Dim LRow As Integer
    Dim LPath As String
    Dim LExtension As String
    Dim LContinue As Boolean

    'Initialize variables
    LContinue = True
    LRow = 2
    LPath = "R:\Symbols\"
    LExtension = ".sym"

    'Loop through all column A values until a blank
    While LContinue

        'Found a blank cell, do not continue
        If Cells(LRow, 1).Value = "" Then
            LContinue = False
        Else
            'Do something with the file
            '...
        End If
        LRow = LRow + 1
    Wend

End Sub
```

Properties - Module1

Module1 Module

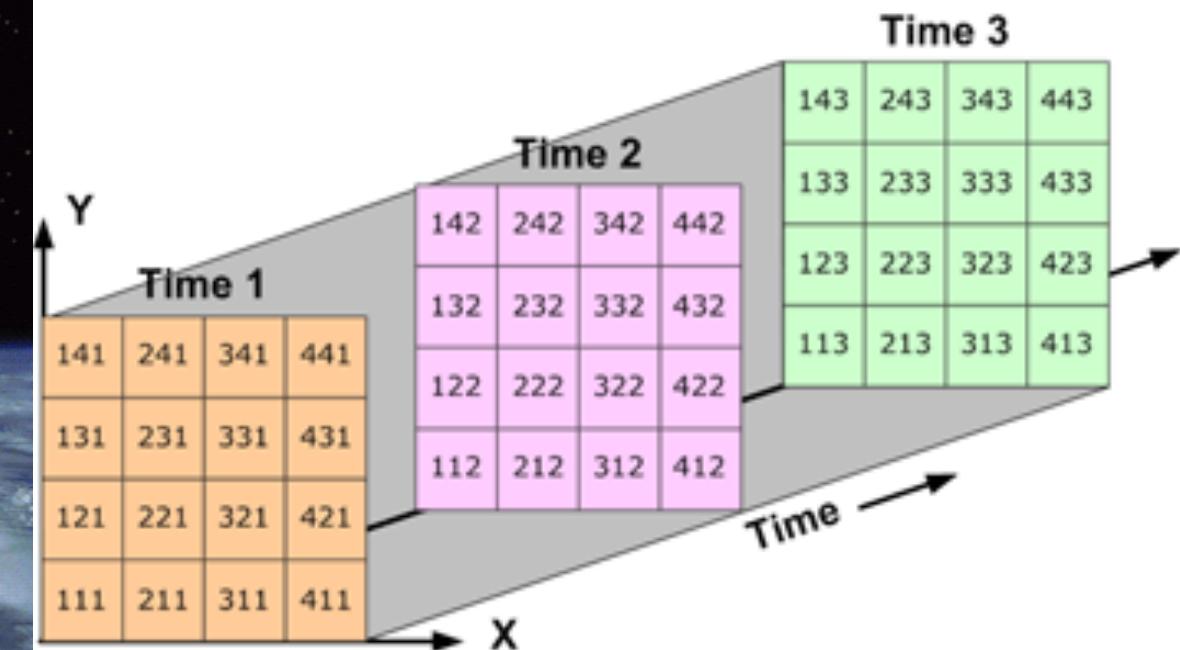
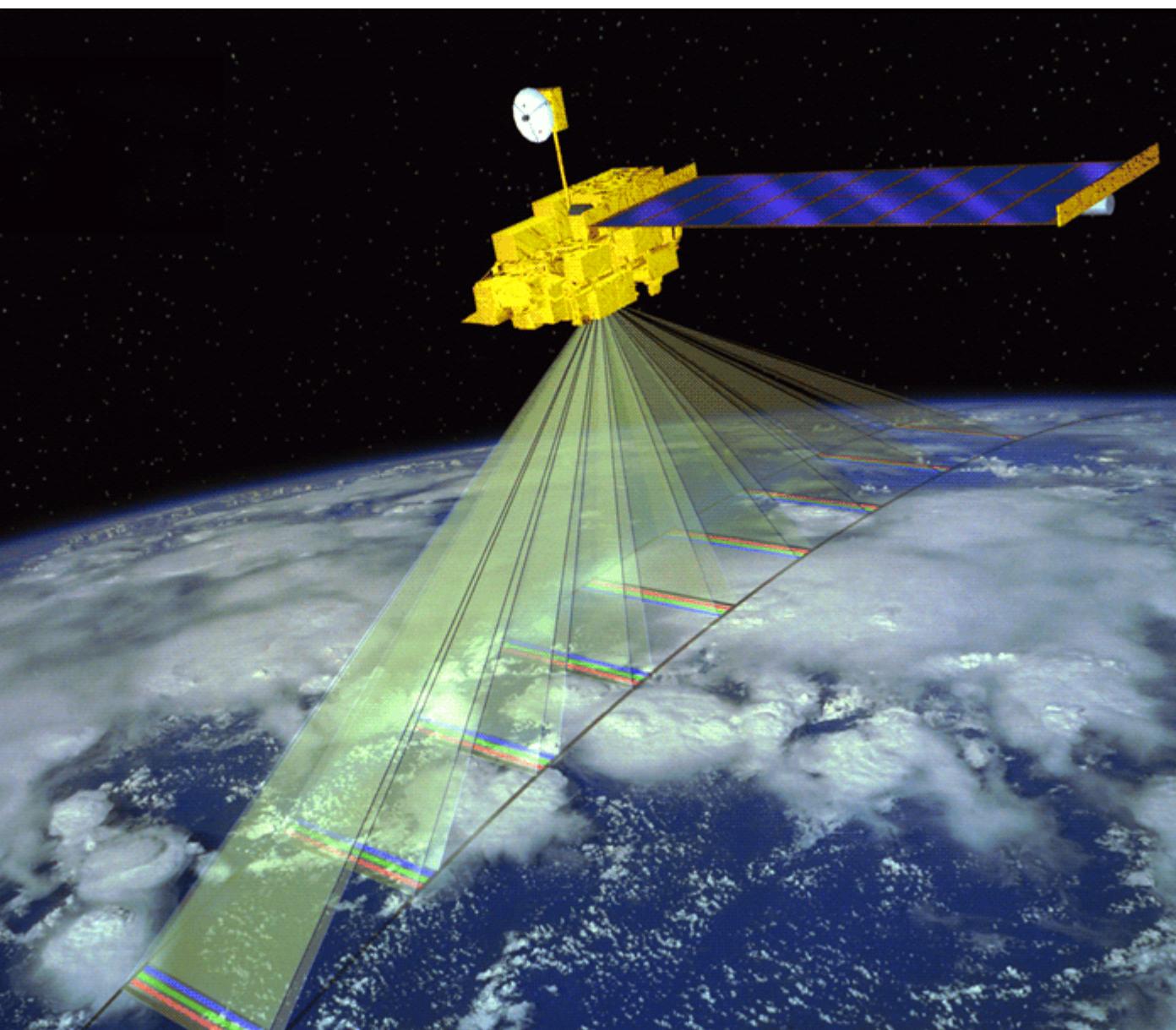
Alphabetic Categorized

(Name) Module1

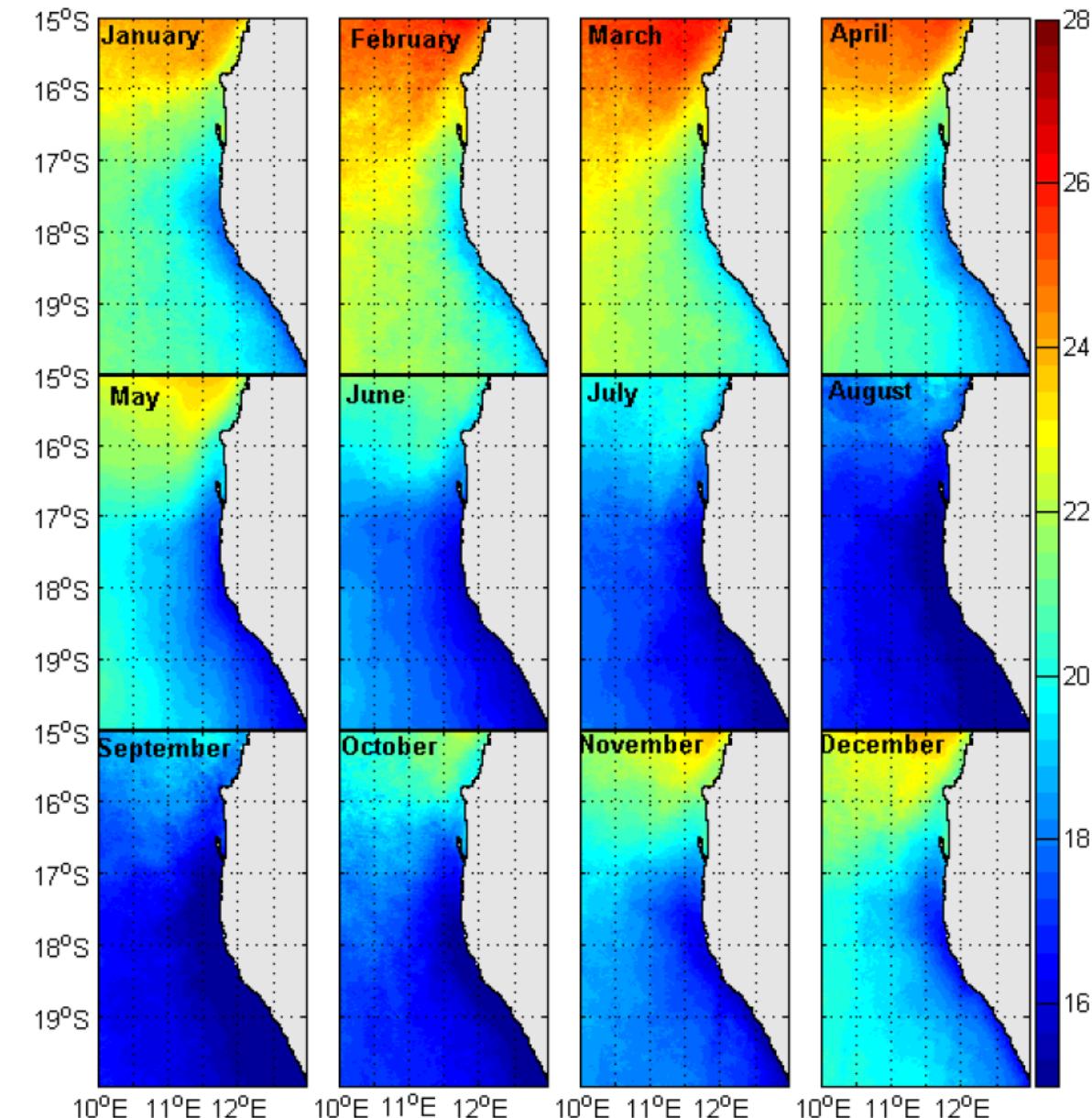
Immediate

TechOnTheNet 2018

?Kate... matlab



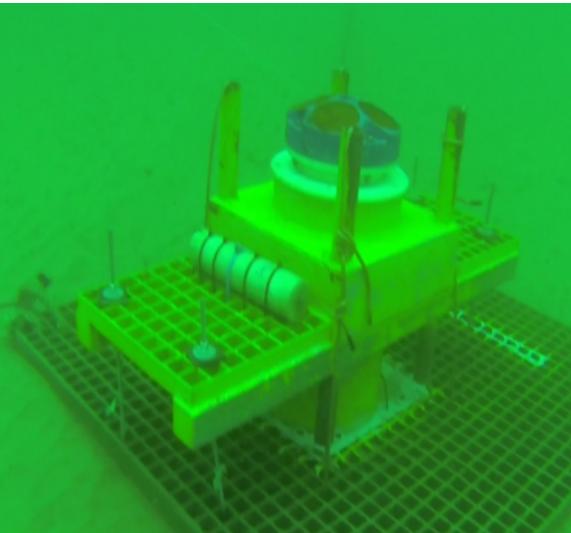
?Kate... matlab



```
44 %% indices from pressure data
45 %Load peak data - inputed manually to xls file
46 [peaks,txt]=xlsread('peak_details.xlsx')
47 %%run loops for all files
48 for i=1:length(peaks(:,1))
49     stationname=txt{i+1,7}
50     % stationname='CTA';
51     station.name= stationname;
52     down = [round(peaks(i,2)):round(peaks(i,3))];
53     up = [round(peaks(i,4)):round(peaks(i,5))];
54     %%down = [round(peaks(19,2)):round(peaks(19,3))];
55     %%up = [round(peaks(19,4)):round(peaks(19,5))];
56 %% keeping some time info
57 start_time_down = datestr(data.time(down(1)));
58 end_time_down = datestr(data.time(down(end)));
59 %%start_time_up = datestr(data.time(up(1)));
60 %%end_time_up = datestr(data.time(up(end)));
61 %% separating upcast and downcast
62
63 %%downcast
64
65 names = fieldnames(data);
66 for i = 1:length(names)
67     downcast.(names{i}) = data.(names{i})(down);
68 end
69
70 %%upcast
71
72 %%for i = 1:length(names)
```

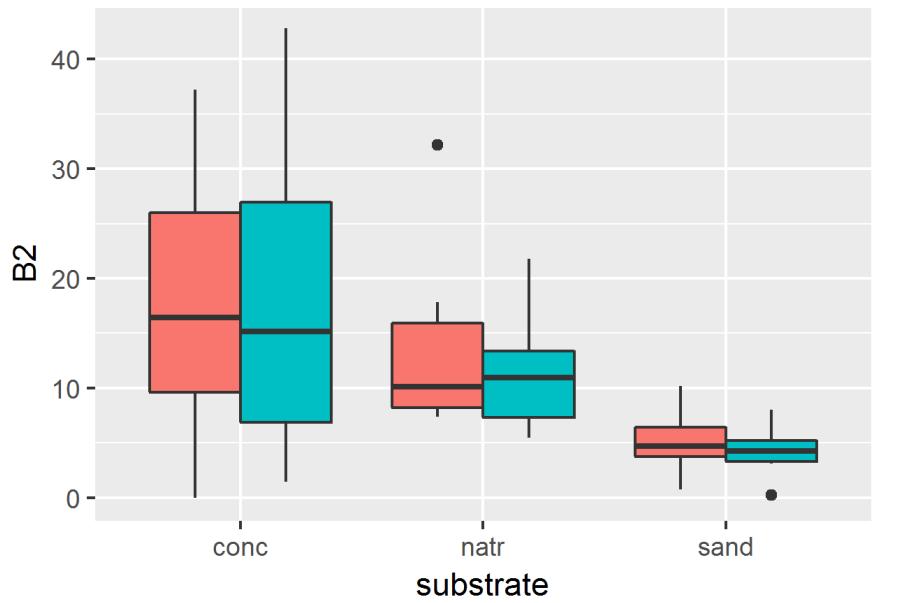
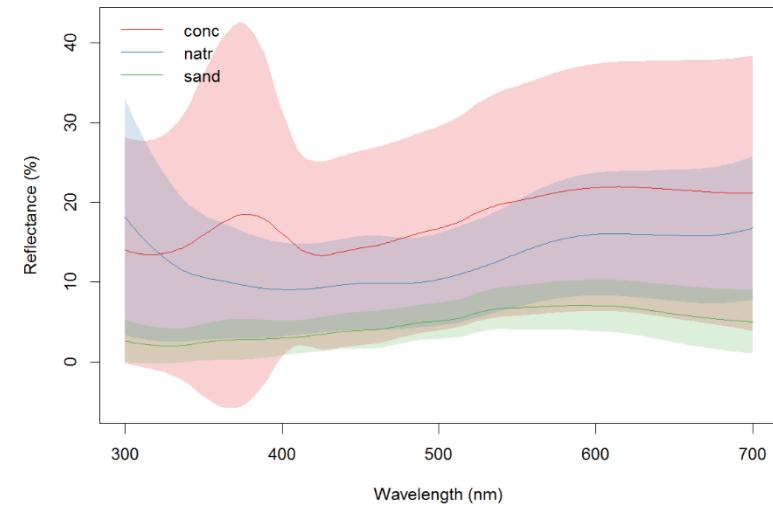


?Kate... python



```
15 # 4. Compare data
16
17 """ Step 1"""
18 import pandas as pd
19 import pylab as py
20 import numpy as np
21 import matplotlib as plt
22 import seaborn as sns
23
24 csv_file=pd.read_csv('C:/Users/bio-student/OneDrive - Macquarie University/
25 datafile=csv_file[12:] #check that row 12 is the start of the data on all f
26 datafile=datafile.reset_index()
27 datafile=datafile.drop(['index'],axis=1)
28 new_names={'Data from kate_Reflection_023.txt Node':'Wavelength', 'Unnamed:
29 datafile=datafile.rename(index=str,columns=new_names)
30 datafile['Wavelength']=pd.to_numeric(datafile['Wavelength'])
31
32 # removing outliers - using + and - 100 as reflectance max/min
33 ind = [i for i,x in enumerate(datafile['Reflectance']) if x < -100]
34 datafile.replace([datafile['Reflectance'].iloc[ind]],np.nan)
35 ind = [i for i,x in enumerate(datafile['Reflectance']) if x > 100]
36 datafile.replace([datafile['Reflectance'].iloc[ind]],np.nan)
37
38 # removing data before 200 nm and above 800 nm wavelength
39 datafile=(datafile.Wavelength > 199) & (datafile.Wavelength < 801)
40
41 #Saving stats file
42 stats=datafile.describe()
43 stats.to_csv('C:/Users/bio-student/OneDrive - Macquarie University/PhD/DATA
44
45 #Savig Xls file
46 datafile.to_csv('C:/Users/bio-student/OneDrive - Macquarie University/PhD/D
47
48 #Plotting and saving plot
49 pic=datafile.plot('Wavelength','Reflectance',ylim=[-100,100])
50 fig=pic.get_figure()
51 fig.savefig('C:/Users/bio-student/OneDrive - Macquarie University/PhD/DATA
52
```

?Kate... R

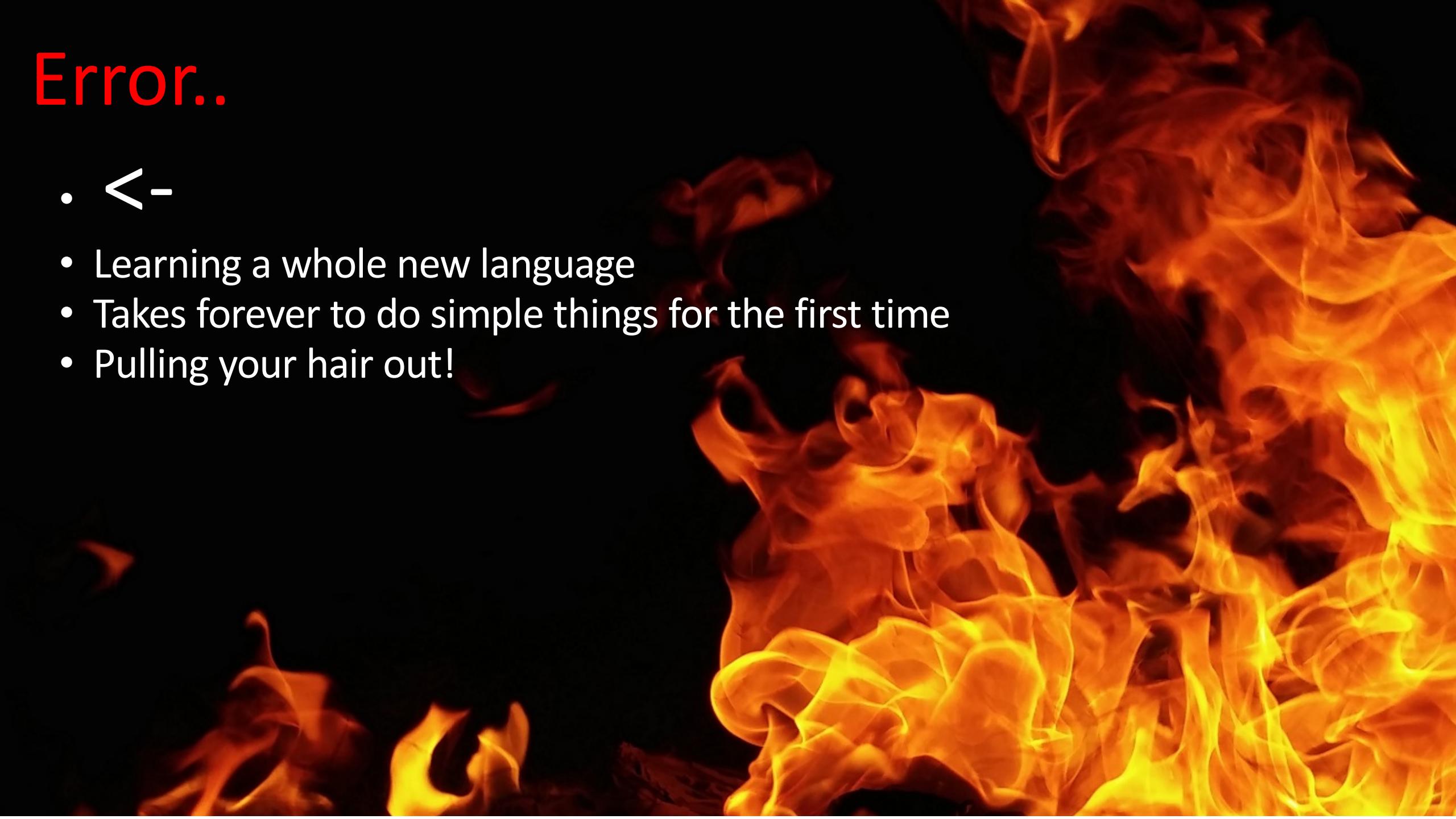


water
dry
wet

```
20 # - PLOT: Boxplots of B2 by substrate
21 # run all = ctrl+alt+R
22
23 rm(list=ls())
24 graphics.off()
25
26 library(plyr)
27 library(dplyr)
28 library(pavo)
29 library(tibble)
30 library(tidyr)
31 library(ggplot2)
32 library(stringr)
33
34 path<- 'C:/users/bio-student/OneDrive - Macquarie University/PhD/5_DATA 1_Colour/5_Data'
35 setwd(path)
36 getwd()
37
38 #####
39 ## IMPORTING ALL SPECS
40 #####
41 specs<-read.csv("./outputs/specs_all.csv")
42 specs<-as.rspec(specs)
43 is.rspec(specs)
44
45 #####
46 ## VISUALISING DATA - before fixing negatives
47 #####
48 # rmv in file names means outliers have been removed.
49 # can compare to below section plots if needed, or comment out [ctrl+shift+c]
50 plot(specs,main="specs_rmv",ylim=c(-50,130))
51 dev.copy(tiff,"./outputs/specs_all_rmv.tiff",width=8,height=6,units="in",res=150)
52 dev.off()
53 graphics.off()
54 plot(specs[,grep("wl|conc",colnames(specs))],main="conc_rmv",ylim=c(-50,130))
55 dev.copy(tiff,"./outputs/specs_conc_rmv.tiff",width=8,height=6,units="in",res=150)
56 dev.off()
57 graphics.off()
58 plot(specs[,grep("wl|natr",colnames(specs))],main="natr_rmv",ylim=c(-50,130))
59 dev.copy(tiff,"./outputs/specs_natr_rmv.tiff",width=8,height=6,units="in",res=150)
60 dev.off()
61 graphics.off()
```

Error..

- <-
- Learning a whole new language
- Takes forever to do simple things for the first time
- Pulling your hair out!



R



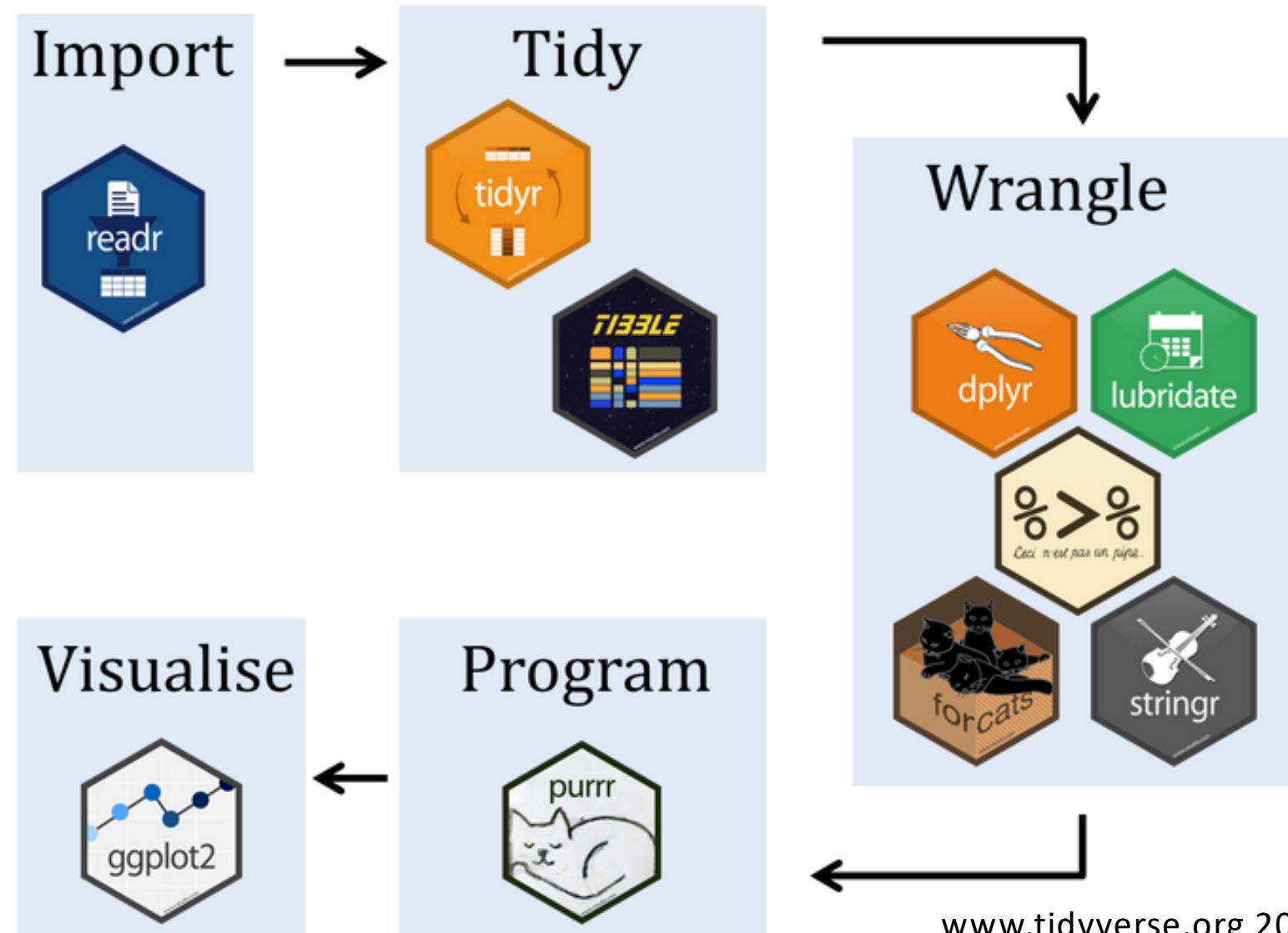
R Loves

- Tidyverse



R Loves

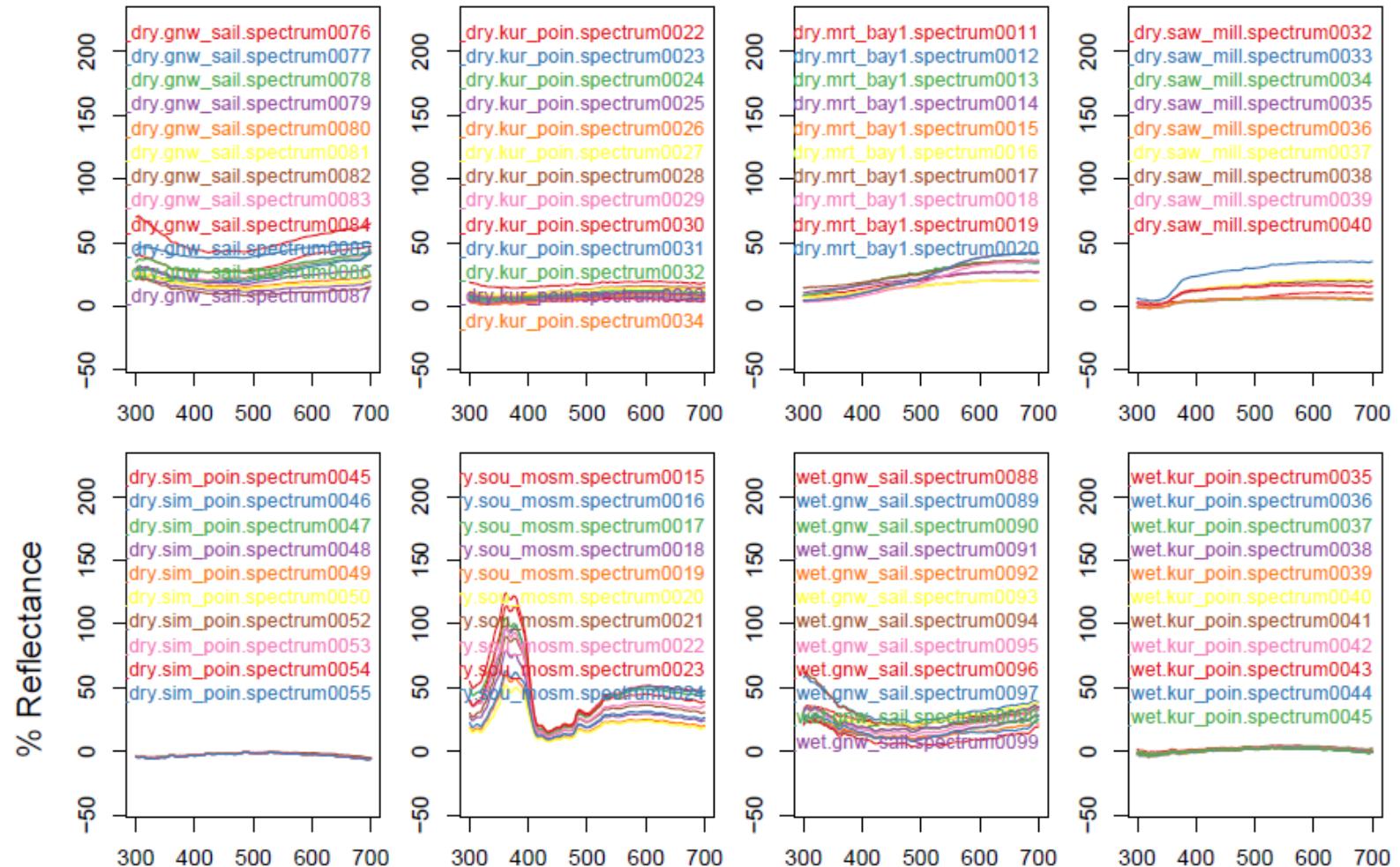
- Tidyverse



R Loves

- Tidyverse
- Plug and play packages

`install.packages("pavo")`



R Loves

- Tidyverse
- Plug and play packages
- Vignettes

HOW TO....

vignette("pavo")

pavo

Rafael Maia, Thomas White, Pierre-Paul Bitton, Chad Eliason

2017-10-02

- Introduction
- Dataset Description
- Organizing Spectral Data
 - Importing Data
 - Subsetting and Merging Data
- Processing Spectral Data
 - Averaging Spectra
 - Normalizing and Smoothing Spectra
 - Binning and PCA Analysis of Spectral Shape
 - Dealing With Negative Values in Spectra
- Visualizing Spectral Data

R Loves

- Tidyverse
 - Plug and play packages
 - Vignettes
 - Cheat sheets

Data Transformation with dplyr :: CHEAT SHEET



dplyr functions work with pipes and expect tidy data. In tidy data:

 Each variable is in its own column	 Each observation, or case, is in its own row	 pipes <code>x %>% f(y)</code> becomes <code>f(x, y)</code>
--	--	--

Summarise Cases

These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

summary function →

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

-  `filter(.data, ...)` Extract rows that meet logical criteria. `filter(iris, Sepal.Length > 7)`
-  `distinct(.data, ..., .keep_all = FALSE)` Remove rows with duplicate values. `distinct(iris, Species)`
-  `sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame())` Randomly select fraction of rows.

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

-  `pull(data, var = -1)` Extract column values as a vector. Choose by name or index. `pull(iris, Sepal.Length)`
-  `select(.data, ...)` Extract columns as a table. Also `select_if()`. `select(iris, Sepal.Length, Species)`

Use these helpers with `select()`, e.g. `select(iris, starts_with("Sepal"))`

RStudio IDE Cheat Sheet

- Data Transformation with dplyr
- Data Visualization with ggplot2
- Package Development with devtools
- Web Applications with shiny
- Interfacing Spark with sparklyr
- R Markdown Cheat Sheet
- R Markdown Reference Guide

R Loves

- Tidyverse
- Plug and play packages
- Vignettes
- Cheat sheets
- Similarity to Python
- It's a whole new language – there's so much to learn.
.....Talk R to others.

....vignette('how to get better at r')

- Spatial R – integrating R with QGIS and Python
- Object orientated programming
- Version control and GIT
- Being more efficient and effective

? What's Next...

... get flying

