

GG606

Intro & Data Vis

Background

- Not a lot of data skills/courses
- Various ways to teach
 - Teach language
 - Teach stats

Why

- Why grad courses?
- Thesis vs project
- Pedagogy

Introductions

- Why are you in GG606
- Undergrad degree
- Research
- Experience with R, other languages

Caveats

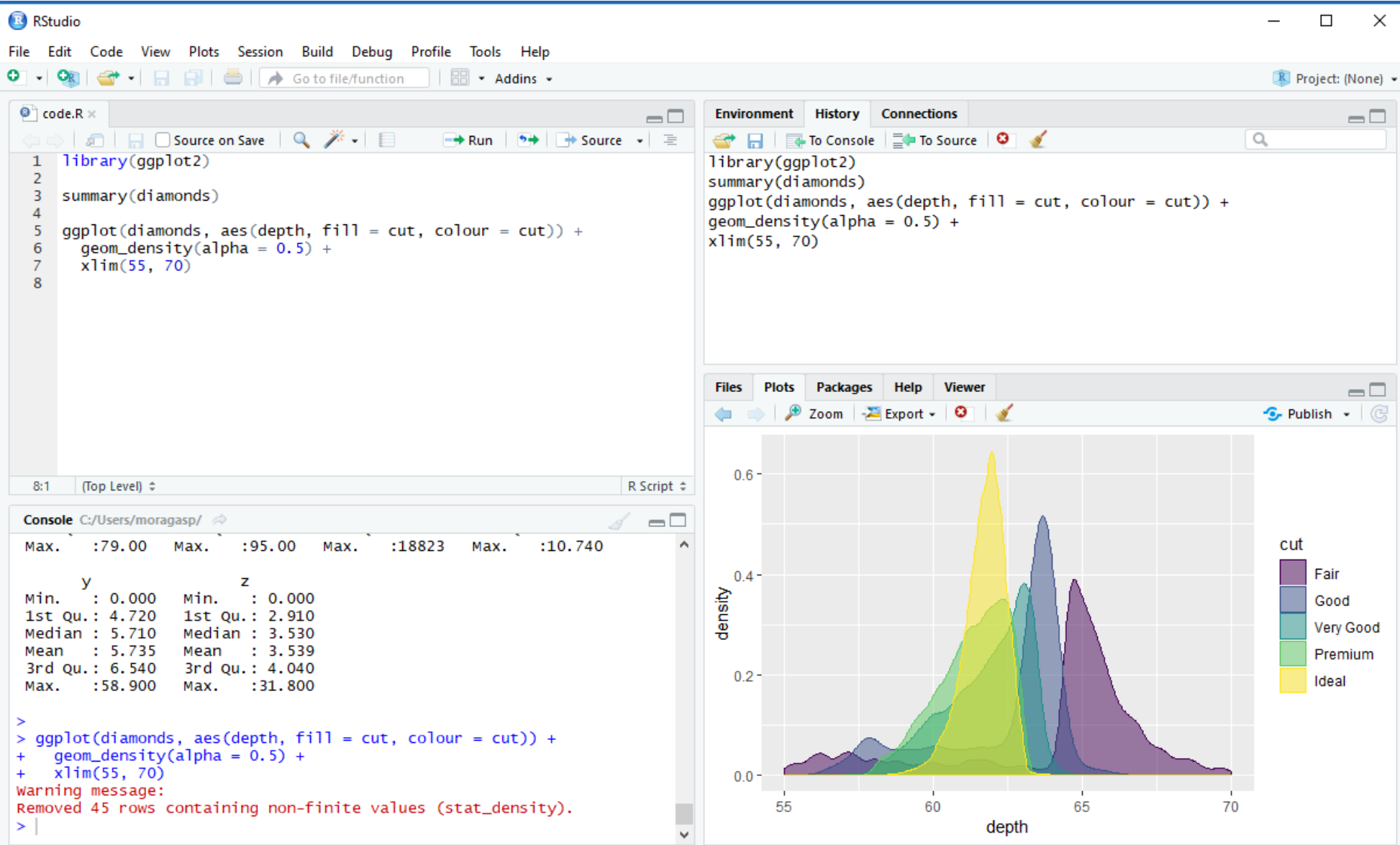
- Not a 'Learn R' course per se
 - Use R to discuss concepts
 - Pros + Cons to R
- Based around *R4DS* book
 - Also the IDS book from UBC
 - Ideally, you will bring new data most weeks

Structure of Grad Courses

- Topics & Discussions
- Lectures & Assignments
- Professional Development
- Skill & Practicum

R & RStudio

- Use R & RStudio example platforms
 - R language
 - RStudio integrated development environment IDE
 - posit.cloud



RStudio

Project: (None)

Untitled1

Source on Save

Run

Source

1

Environment

History

Import Dataset

Global Environment

Environment is empty

Files

Plots

Packages

Help

Viewer

New Folder

Delete

Rename

More

Home > Documents > SoftwareCarpentry > r_gapminder

	Name	Size	Modified
	..		
<input type="checkbox"/>	r-novice-gapminder		

Console

R version 3.5.2 (2018-12-13) -- "Sincere Rumpkin Patch"

Copyright (C) 2016 The R Foundation for Statistical Computing

Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

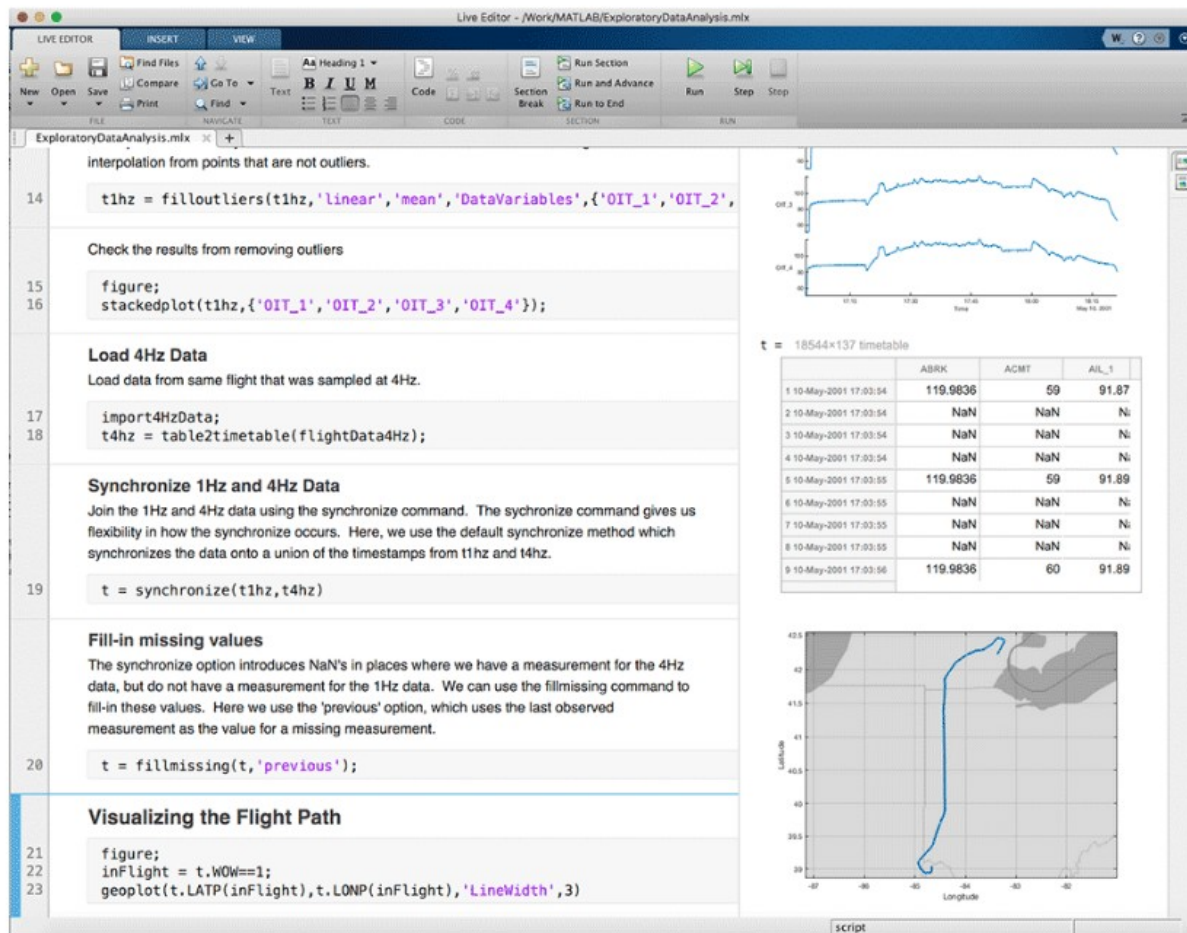
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

Why Use MATLAB for Data Science?

Exploratory Data Analysis

Spend less time preprocessing data. From time-series sensor data to images to text, MATLAB datatypes significantly reduce the time required to preprocess data. High-level functions make it easy to synchronize disparate time series, replace outliers with interpolated values, filter noisy signals, split raw text into words, and much more. Quickly visualize your data to understand trends and identify data quality issues with plots and the Live Editor.



HOME PLOTS APPS EDITOR PUBLISH VIEW

Find Files Compare Print Find Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Editor - /home/jvenkiteswaran/github/PoRGy/PoRGy_Matlab.m

ammonia_degassing_inverse_model.m x degassingModel.m x PoRGy_Matlab.m x +

```
1 function [p, bounds, q] = PoRGy_Matlab(filename)
2 % Single-file MATLAB implementation of the PoRGy model of O2 saturation
3 % and d18O-O2.
4 % This is complete but needs more comments and optimising.
5 % The idea was to include all necessary functions into one M-file so that
6 % it was (1) easier to pass around and (2) would export all the relevant
7 % data with one command.
8 %
9 % [p, bounds] = PoRGy_Matlab(filename)
10 % p is the struct of fitted variables
11 % bounds is the struct of bounds placed on p
12 % and filename is the base filename of the field data (n.b. 'ticks around filename')
13 %
14 % An example of how to call this function:
15 %     [p, bounds] = PoRGy_Matlab('s_sask_hague');
16 %
17 % The r2 values and SSE value will scroll by as MATLAB find a best-fit.
18 % Three output files are created with filenames like these:
19 % s_sask_hague_PoRGy-matlab_output_20070514T111545.csv
20 % s_sask_hague_PoRGy-matlab_output_20070514T111545.png
21 % s_sask_hague_PoRGy-matlab_output_summary_20070514T111545.csv
22 %
23 % Copyright 2007-2008 Jason J. Venkiteswaran & Kevin K. Venkiteswaran
24 %
25 % Revision 1.04 2008-12-11
26 % added code to report cell q with all output data/results required when
```

Current Folder

Name	Size	Date Modified	Type
appdata		2015-06-15 09:41:...	Folder
bin		2015-06-15 09:41:...	Folder
bugreport		2015-06-15 09:40:...	Folder
etc		2015-06-15 09:41:...	Folder
examples		2015-06-15 09:40:...	Folder
extern		2015-06-15 09:41:...	Folder
help		2015-06-15 09:41:...	Folder
java		2015-06-15 09:40:...	Folder
licenses		2015-06-15 09:45:...	Folder
resources		2015-06-15 09:40:...	Folder
sys		2015-06-15 09:41:...	Folder
toolbox		2015-06-15 09:40:...	Folder
ui		2015-06-15 09:40:...	Folder
license_agreement.txt	82 KB	2015-01-14 05:46:...	TXT File
patents.txt	8 KB	2015-02-03 06:39:...	TXT File
trademarks.txt	1 KB	2013-12-28 02:08:...	TXT File

Details

Workspace

Name	Value
------	-------

Command Window

Initializing...

Ln 1 Col 1

Mode: Command Ln 1, Col 1 Lorenz.ipynb

R

- S (John Chambers et al.) at Bell Labs, 1976
 - Implemented as fortran libs
- Rewritten in C, 1988 as v3, v4 in 1998
- R, 1993 (Ross Ihaka & Robert Gentleman)
- 1995 GNU GPL, paper 1996
- ‘packages’ to expand capabilities

R + 'tidyverse'

- Hadley Wickham PhD thesis 2008
- 'ggplot' 2005/7, Grammar of Graphics (Leland Wilkinson)
- 'plyr' 2008, tools 'split-apply-combine'
- 'tidy data'
- *opinionated*



Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II.

<http://www.jstatsoft.org/>

Tidy Data

Hadley Wickham
RStudio

Abstract

A huge amount of effort is spent cleaning data to get it ready for analysis, but there has been little research on how to make data cleaning as easy and effective as possible. This paper tackles a small, but important, component of data cleaning: data tidying. Tidy datasets are easy to manipulate, model and visualise, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table. This framework makes it easy to tidy messy datasets because only a small set of tools are needed to deal with a wide range of un-tidy datasets. This structure also makes it easier to develop tidy tools for data analysis, tools that both input and output tidy datasets. The advantages of a consistent data structure and matching tools are demonstrated with a case study free from mundane data manipulation chores.

Tidy vs Messy Data

- each variable is a column
- each observation (or case) is a row
- *Give an example where you didn't follow this in the past*

country	year	cases	population
Afghanistan	1999	37745	19987071
Afghanistan	2000	3666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	214258	1272915272
China	2000	216766	1280425583

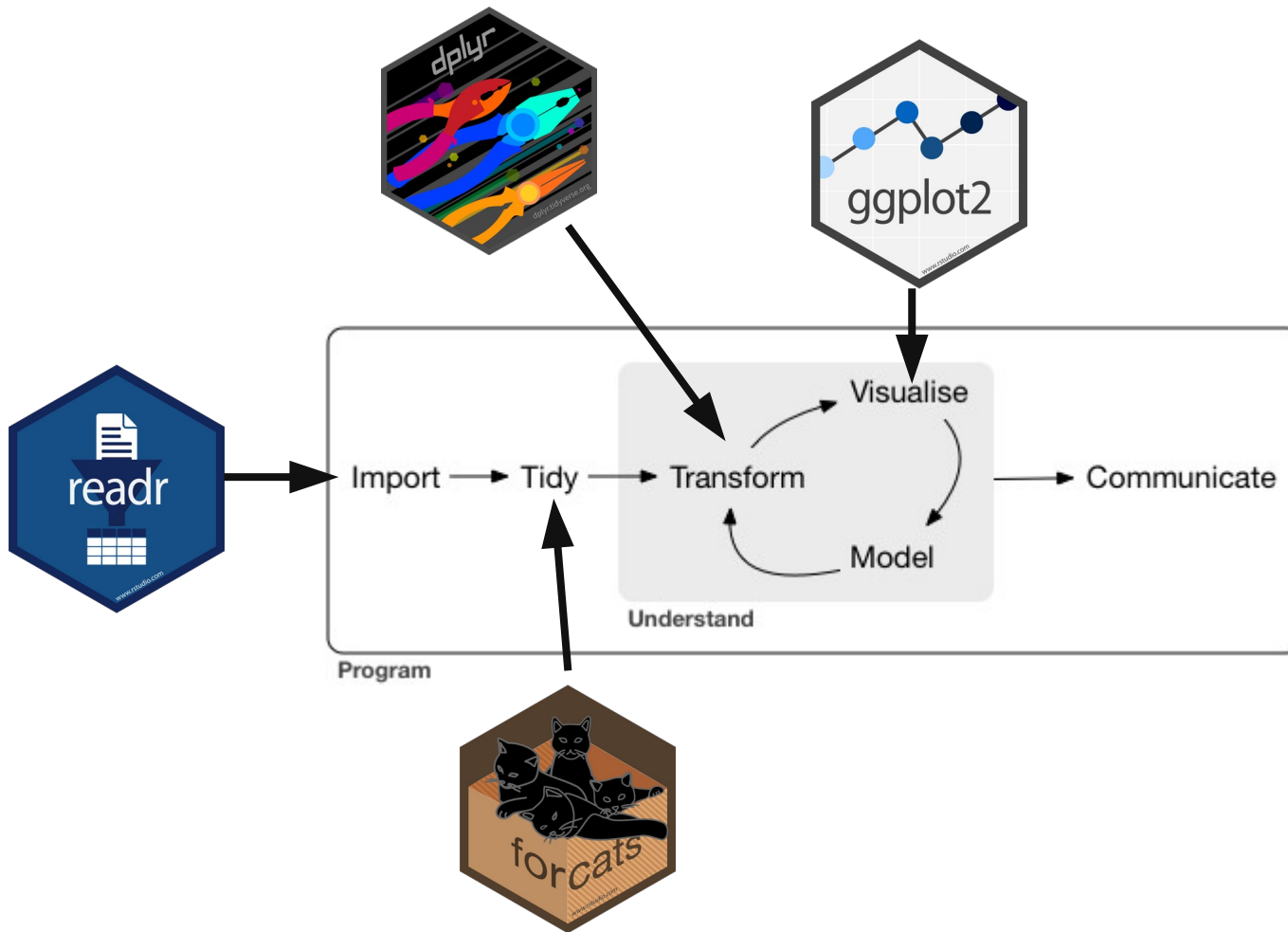
variables

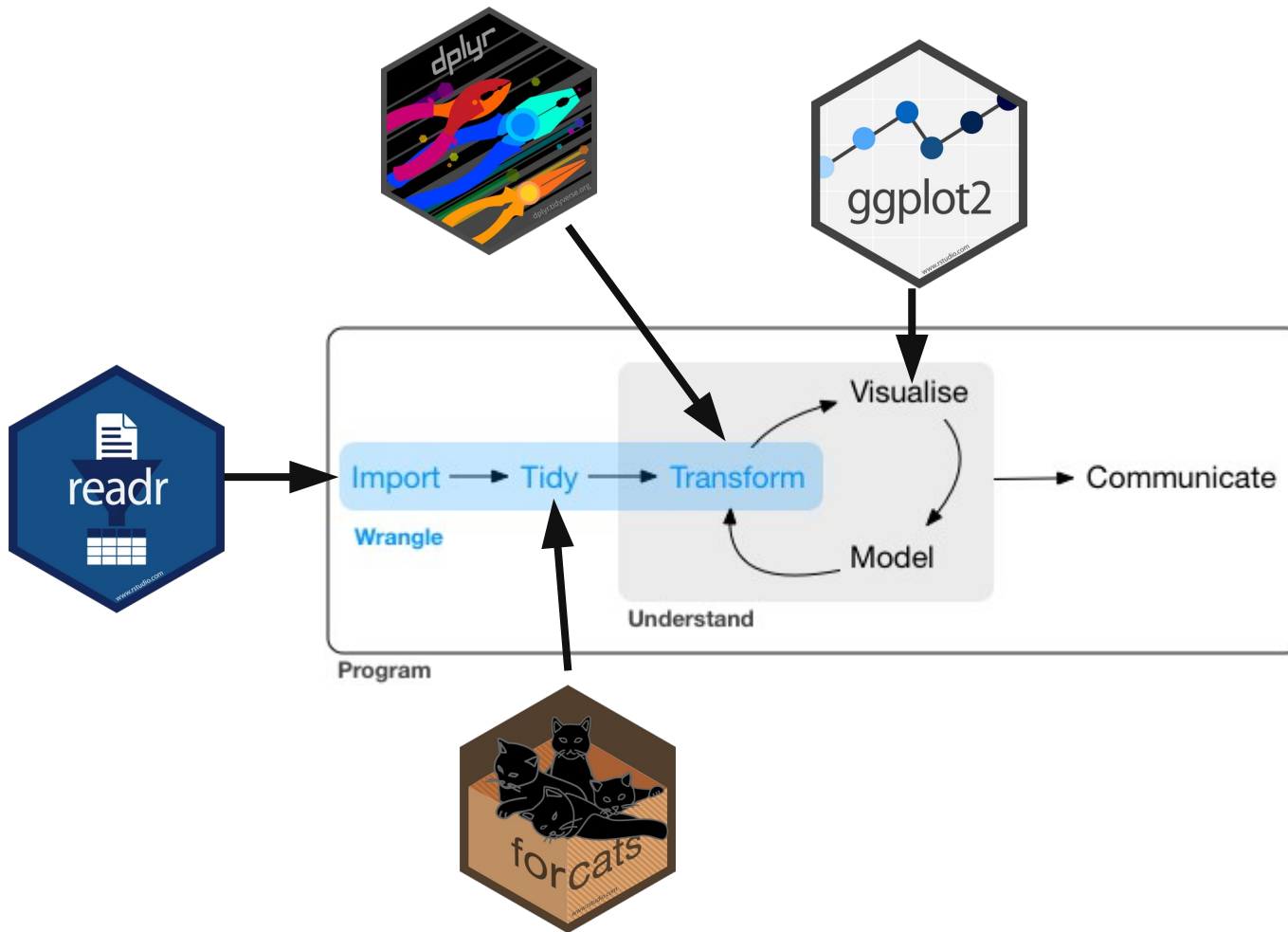
country	year	cases	population
Afghanistan	1999	37745	19987071
Afghanistan	2000	3666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	214258	1272915272
China	2000	216766	1280425583

observations

country	year	cases	population
Afghanistan	1999	37745	19987071
Afghanistan	2000	3666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	214258	1272915272
China	2000	216766	1280425583

values









Break?

Putting Little Things Together

- RStudio IDE
- Project Management
- R Intro



Vince Buffalo
@vsbuffalo



Managing your projects in a reproducible fashion doesn't just make your science reproducible, it makes your life easier.

11:26 PM · Apr 14, 2013 · TweetDeck

RStudio IDE

- Default 4 pane layout

RStudio Project

- Click the “File” menu button, then “New Project”.
- Click “New Directory”.
- Click “Empty Project”.
- Type in the name of the directory to store your project, e.g. “my_project”.
- *If available, select the checkbox for “Create a git repository.”*
- Click the “Create Project” button.

Matlab projects

Create Projects

What Are Projects?

A project is a scalable environment where you can manage MATLAB® files, data files, requirements, reports, spreadsheets, tests, and generated files together in one place.

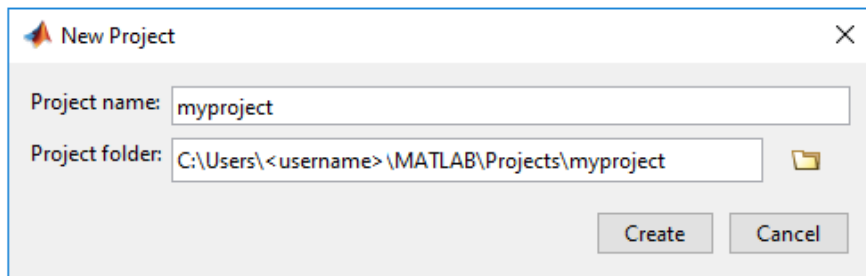
Projects can help you organize your work and collaborate. Projects promote productivity and teamwork by helping you with common tasks.

- Find all the files that belong with your project.
- Create standard ways to set up and shut down the MATLAB environment across a team.
- Create, store, and easily access common operations.
- View and label modified files for peer review workflows.
- Share projects using built-in integration with Git™, Subversion® (SVN), or using external source control tools.

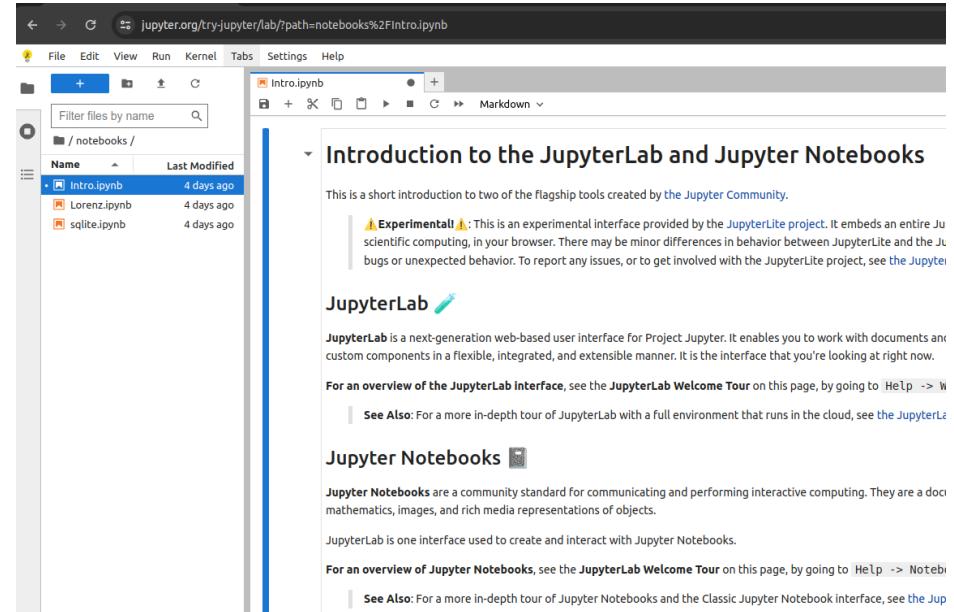
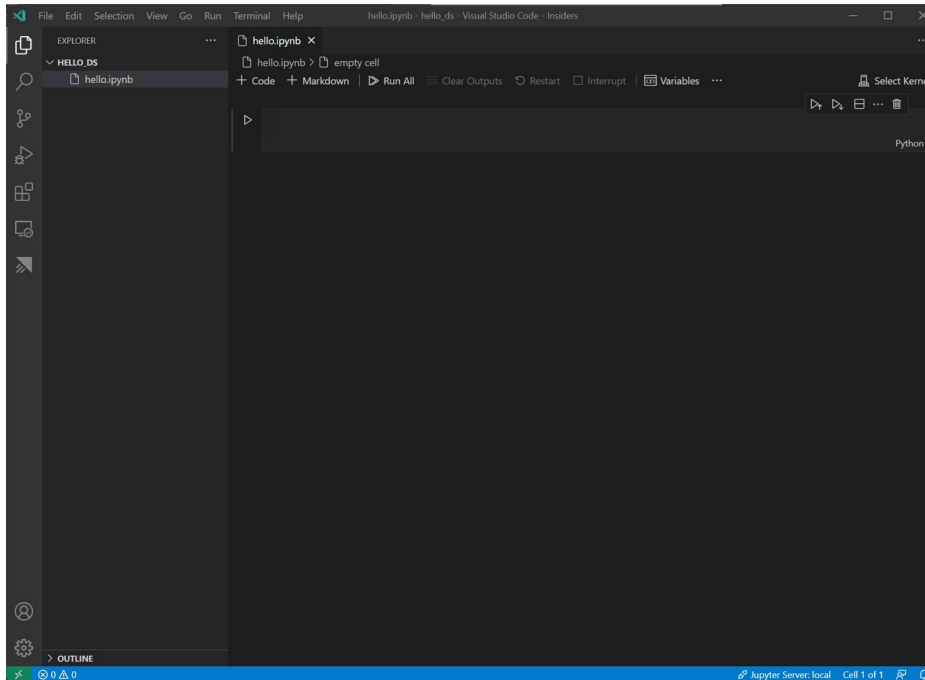
Create Project

To create a blank project, on the **Home** tab, click **New > Project > Blank Project**. To create a project from an existing folder, on the **Home** tab, click **New > Project > From Folder**.

The New Project dialog box opens. Enter a project name, select a project folder, and click **Create**.



Jupyter notebooks



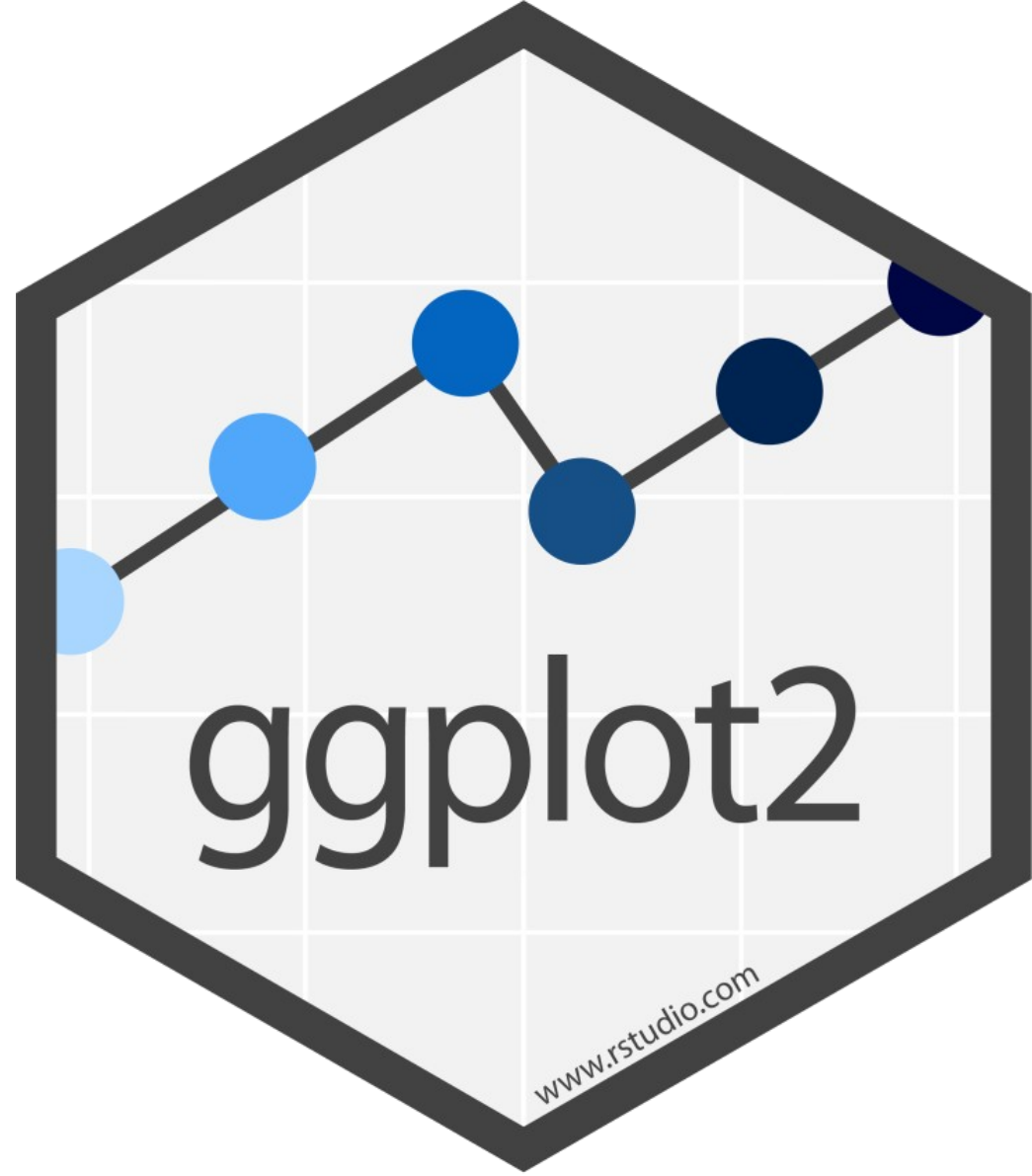
R Intro 1

- Calculator
 - Functions
 - Comparisons
 - Assignment
 - *R-Scripts*
- $1+2$
 - $\sin(1)$
 - $1 = 1$
 - $x \leftarrow 3$
 - File, New File, R Script

R Intro 2

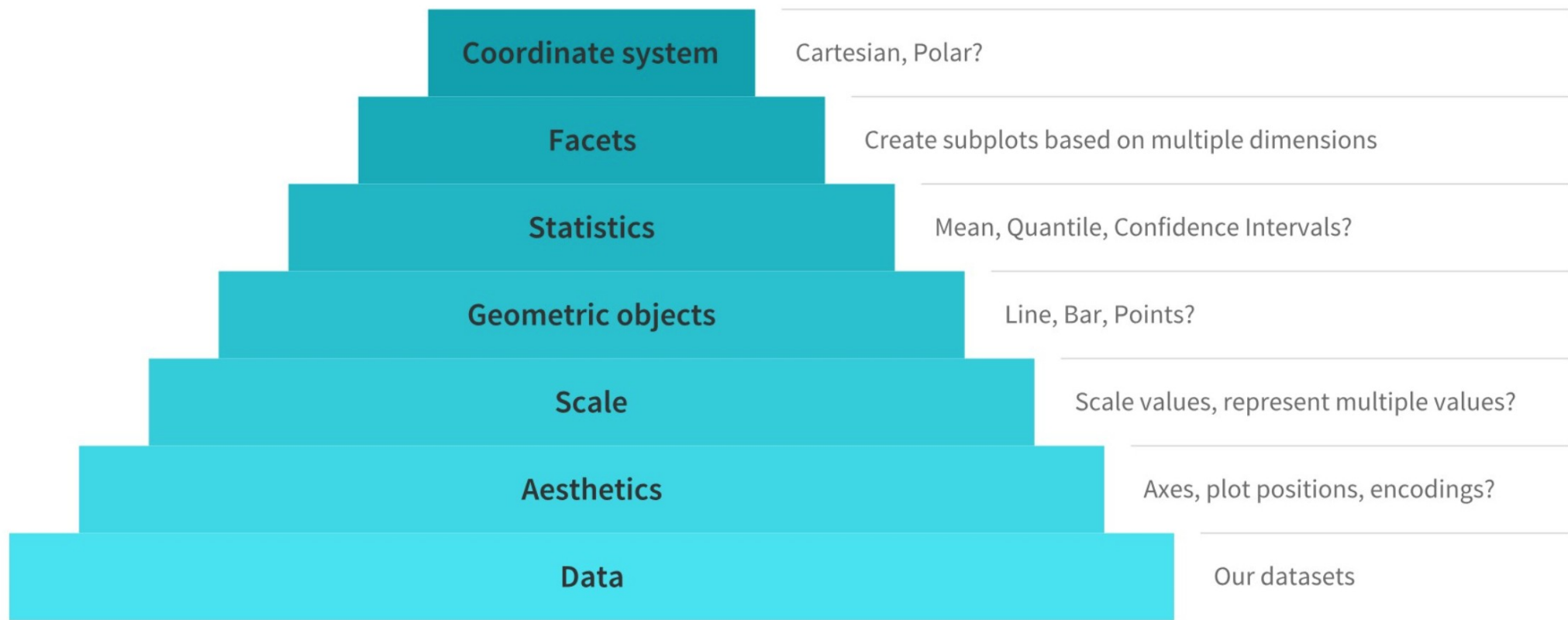
- Write commands in an R Script and save it
- periods.between.words
- underscores_between_words
- camelCaseForWords
- *Do you want to create a variable if that name is already in use?*

```
install.packages("tidyverse")
```



ggplot2

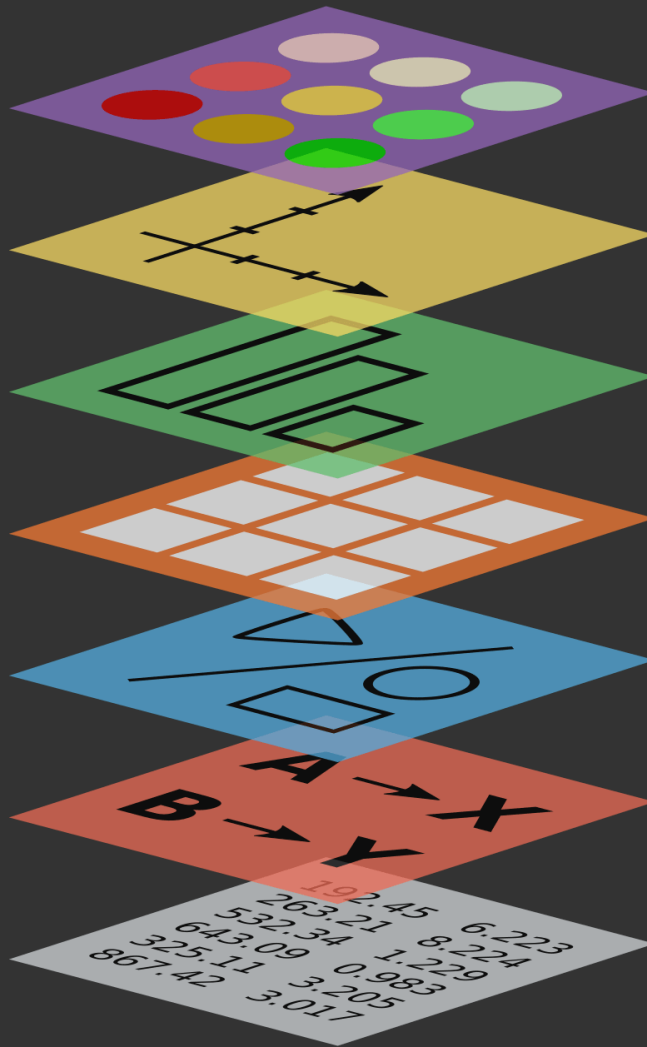
- *grammar is about structure*
 - data, variables mapped to aesthetics
 - layer (geom elements, stat transforms)
 - mapping (aesthetics)
 - scale (data space to aes space, colour, size, shape)
 - coord (Cartesian, polar, map)
 - facet (small multiples, latticing, trellising)
 - theme (details)
- *What does this not do?*



The major elements of the grammar of graphics. Source: Towards Data Science



Theme
Coordinates
Statistics
Facets
Geometries
Aesthetics
Data



Start

- `library(tidyverse)`
- `mpg`
- `ggplot(mpg) + geom_point(aes(x = displ, y = hwy))`
- *Discuss*

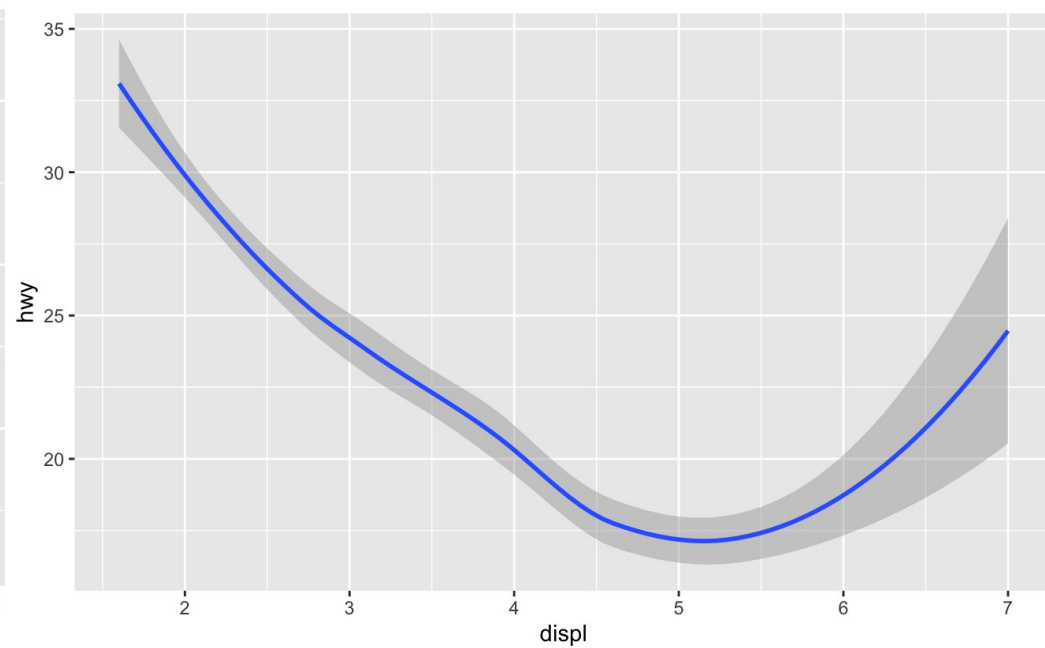
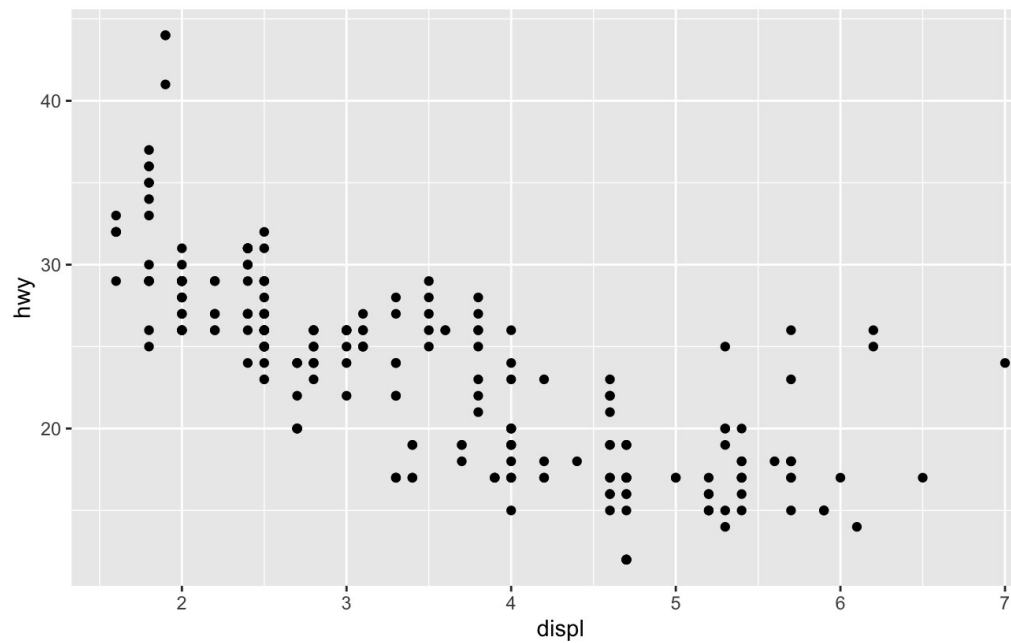
Start

- `library(tidyverse)`
- `mpg`
- `ggplot(mpg) + geom_point(aes(x = displ, y = hwy))`
- *Discuss*
- `colour`, `size`, `shape`, `alpha`
- Inside vs outside the `aes()`

Facets

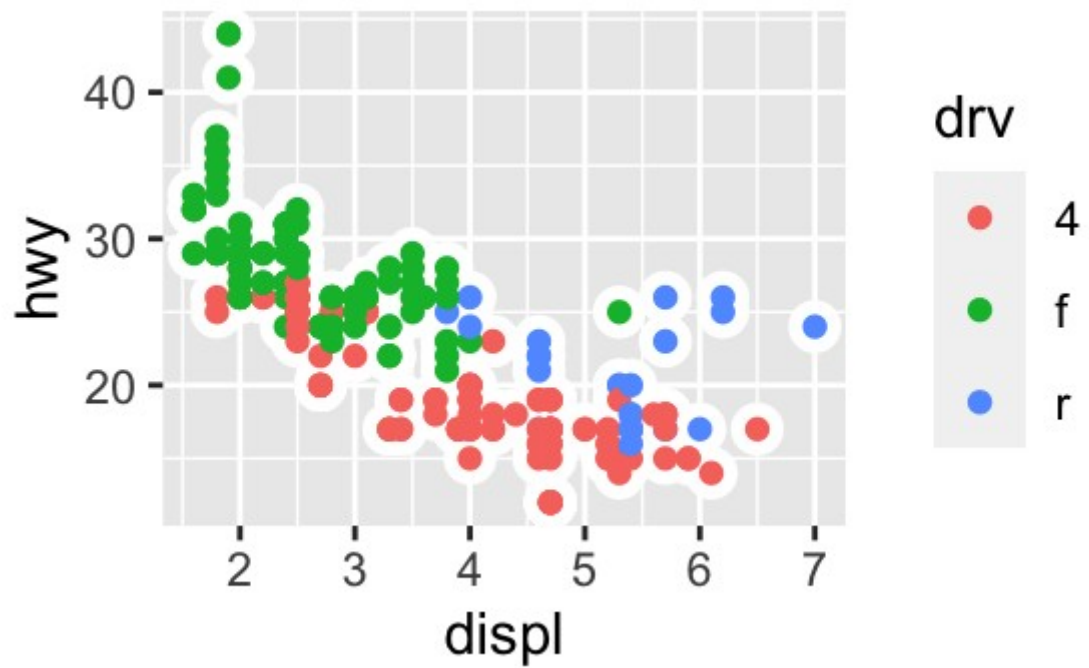
- Splitting plot by a variable
- `ggplot(data = mpg) +
 geom_point(mapping = aes(x = displ, y = hwy)) +
 facet_wrap(~ class, nrow = 2)`
- `ggplot(data = mpg) +
 geom_point(mapping = aes(x = displ, y = hwy)) +
 facet_grid(drv ~ cyl)`
- *Discuss*

Other geom



Other geoms

- `ggplot(data = mpg) +
 geom_point(mapping = aes(x = displ, y = hwy)) +
 geom_smooth(mapping = aes(x = displ, y = hwy))`
- `ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
 geom_point(mapping = aes(colour = class)) +
 geom_smooth()`
- *Discuss*



Break?

Stats

- `ggplot(diamonds) +
 geom_bar(aes(cut))`

1. `geom_bar()` begins with the **diamonds** data set

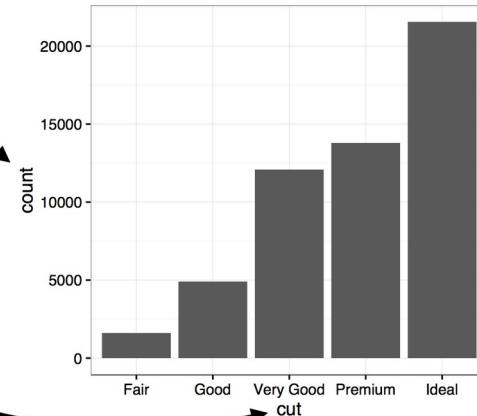
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...

2. `geom_bar()` transforms the data with the "count" stat, which returns a data set of cut values and counts.

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

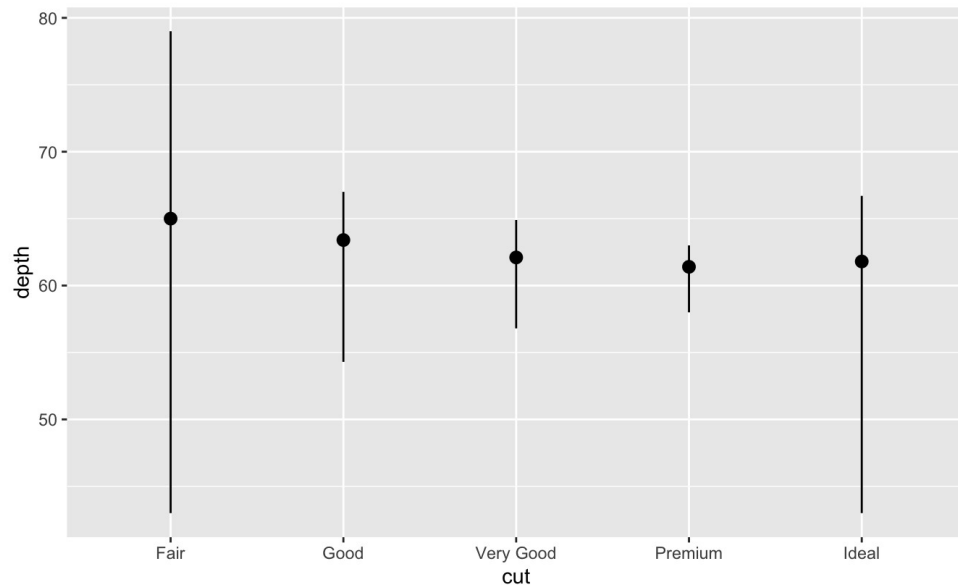
3. `geom_bar()` uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.



?geom_bar

Stats

- `ggplot(data = diamonds) +
 stat_summary(
 mapping = aes(x = cut, y = depth),
 fun.min = min,
 fun.max = max,
 fun = median
)`



Position

- Colour vs fill
- `ggplot(data = diamonds) +
 geom_bar(aes(cut, X = clarity))`

Position

- Colour vs fill
- `ggplot(data = diamonds) +
 geom_bar(aes(cut, X = clarity))`
- `identity`
*special for bars, stacked, overlapping,
proportion, side-by-side*

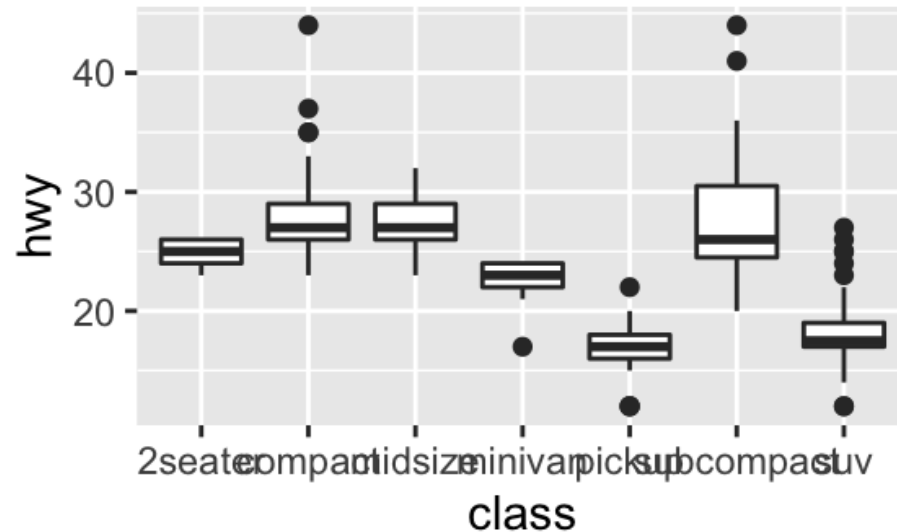
Position

- `ggplot(mpg, aes(x = cty, y = hwy)) +
 geom_point()`
- 234 rows in mpg, where are they?
- `position = "jitter"`
or
`geom_jitter()`
- *Discuss how to improve figure*

?position_

Coords

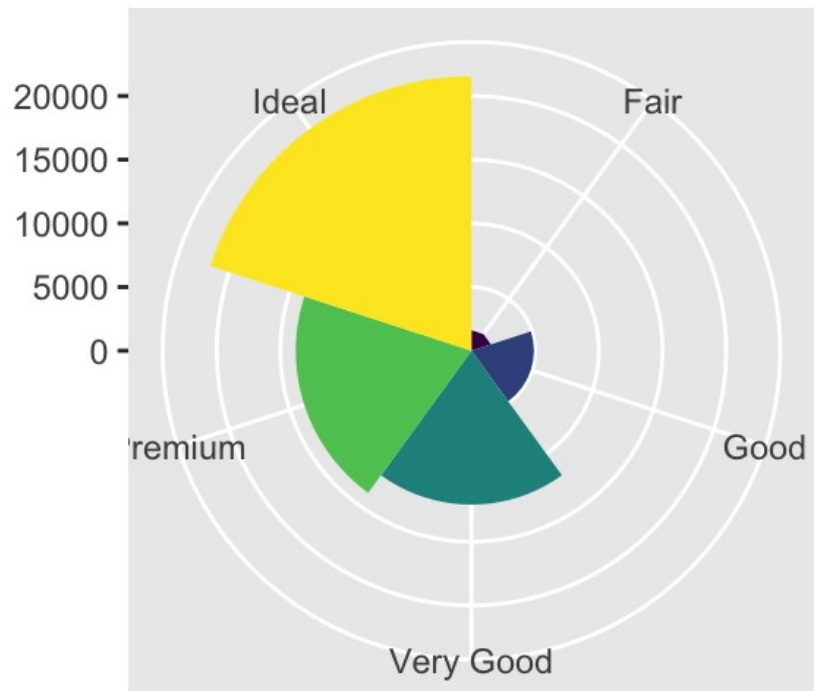
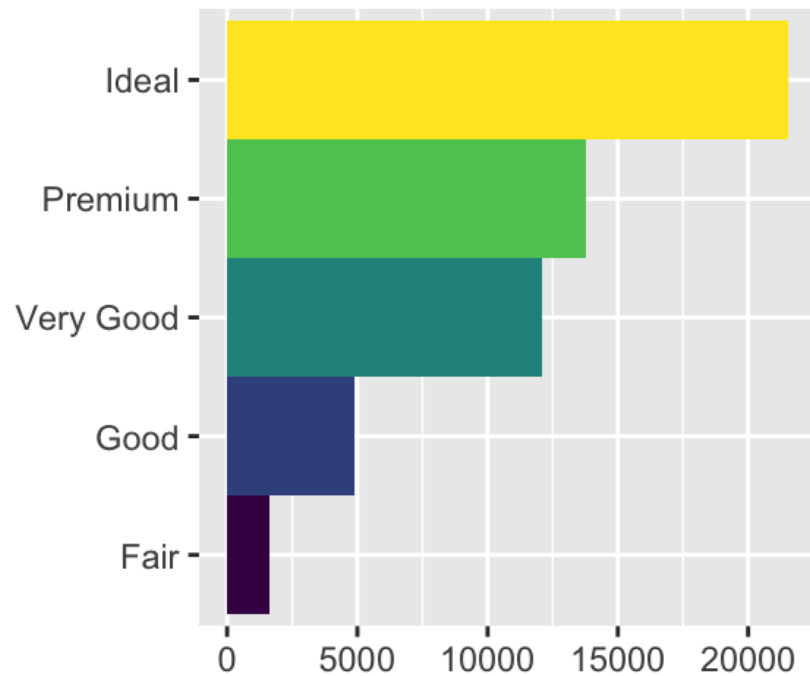
- *Which coordinate system?*
- `ggplot(mpg, aes(class, hwy)) +
 geom_boxplot() +
 coord_flip()`
- *What will this do? Useful?*



Coords

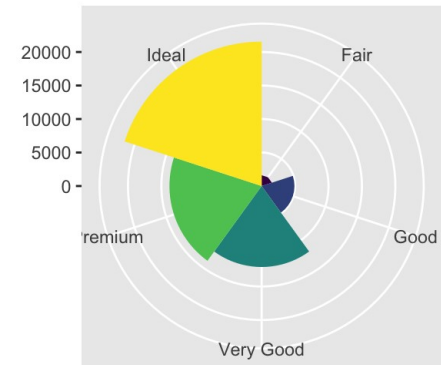
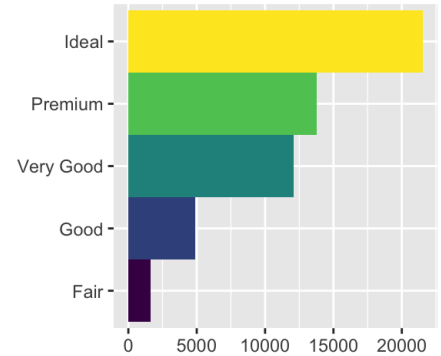
- When would other coordinate systems be useful?
- `?coord_polar`

Coords



Coords

- `bar ← ggplot(diamonds) +
 geom_bar(aes(x = cut, fill = cut),
 show.legend = FALSE,
 width = 1) +
 theme(aspect.ratio = 1) +
 labs(x = NULL, y = NULL)`
- `bar + coord_flip()`
- `bar + coord_polar()`



Summary

- `ggplot(data = <DATA>) +
 <GEOM_FUNCTION>(
 mapping = aes(<MAPPINGS>),
 stat = <STAT>,
 position = <POSITION>
) +
 <COORDINATE_FUNCTION> +
 <FACET_FUNCTION>`

Homework

- `install.packages("palmerpenguins")`

```
> library(palmerpenguins)
> penguins
# A tibble: 344 x 8
  species island bill_length_mm bill_depth_mm flipper_length... body_mass_g
  <fct>    <fct>      <dbl>        <dbl>        <int>        <int>
1 Adelie  Torge...    39.1         18.7         181         3750
2 Adelie  Torge...    39.5         17.4         186         3800
3 Adelie  Torge...    40.3          18         195         3250
4 Adelie  Torge...    NA           NA           NA           NA
5 Adelie  Torge...    36.7         19.3         193         3450
6 Adelie  Torge...    39.3         20.6         190         3650
7 Adelie  Torge...    38.9         17.8         181         3625
8 Adelie  Torge...    39.2         19.6         195         4675
9 Adelie  Torge...    34.1         18.1         193         3475
10 Adelie Torge...    42           20.2         190         4250
# ... with 334 more rows, and 2 more variables: sex <fct>, year <int>
> 
```

Homework

- `install.packages("palmerpenguins")`
- Make some interesting plots with new `geom_` and `theme_`
- ggplot2.tidyverse.org
- Try out `ggsave`
- pangaea.de & www.frdr-dfdr.ca & borealisdata.ca
- allisonhorst.github.io/palmerpenguins/

