Introduction to R Workshop

Session 4

Sean Nguyen

MSU BEST

Broadening Experiences in Scientific Training

Rstudio keyboard shortcuts

- alt + ‘-‘ **<-**
- cmd/ctrl + shift + m **%>%**
- cmd/ctrl + shift + r **# new section**
- cmd/ctrl + shift + c **# comment**

Session 4: Goals

- Statistical tests
- R markdown
- R notebooks
- GitHub

Power analysis

- pwr - package
- Determine sample size to detect effect given sample size and degree of confidence
- Need three to calculate the fourth
  - **sample size = n**
  - **effect size = d**
  - **significance level (P value) = sig.level**
  - **power 1-P = power**

(ANOVA)

number of groups = k

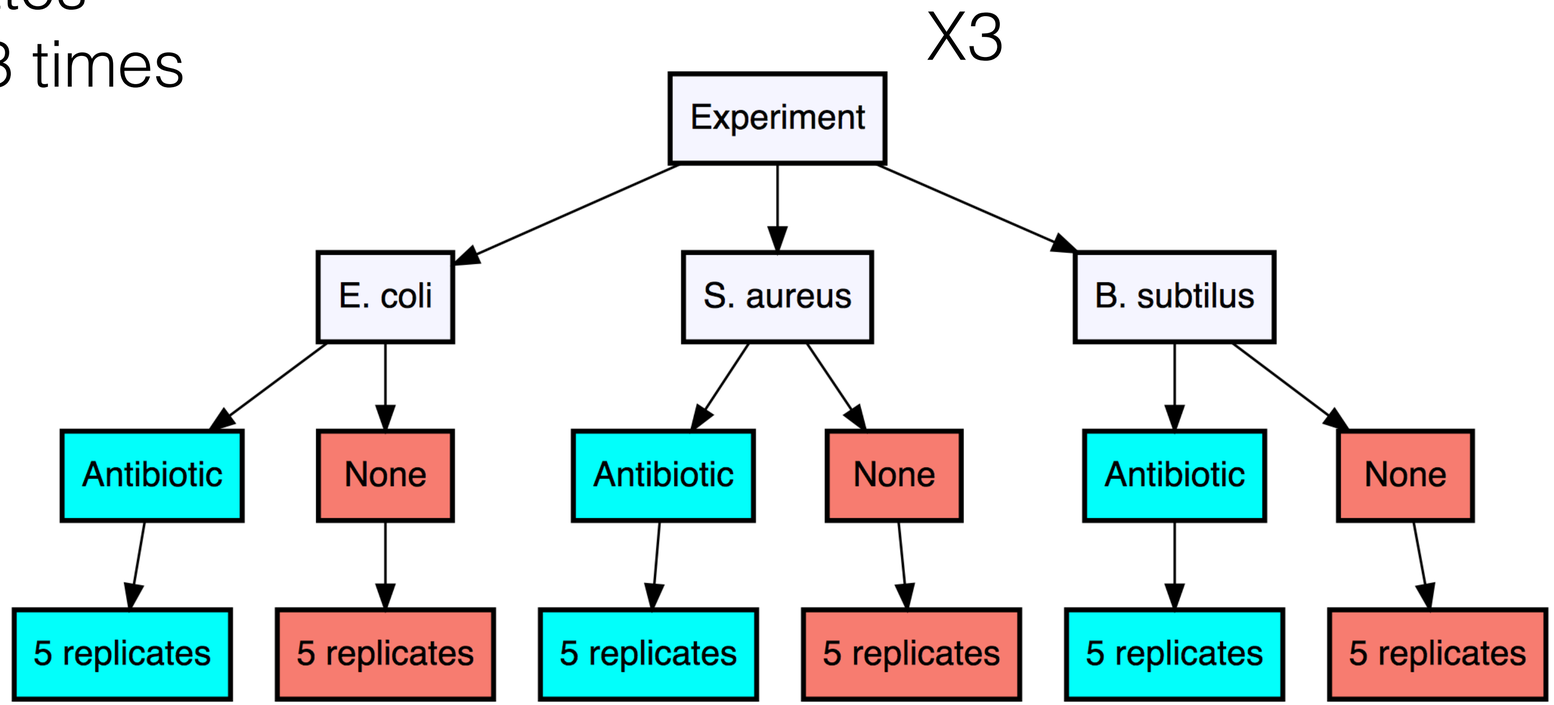
effect size = f (0.1, 0.25, 0.4)

```
pwr.anova.test(k = ,n = ,f = , sig.level =, power = )
```

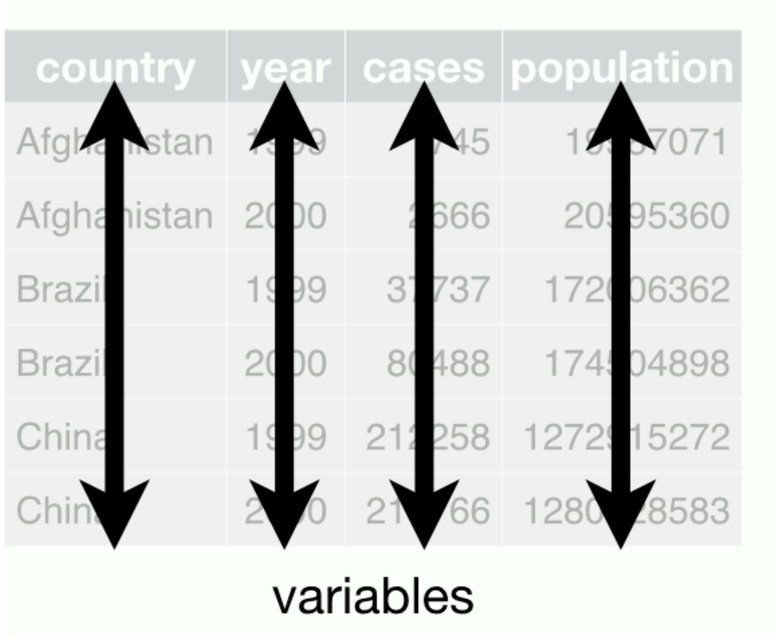
```
pwr.t.test(n = , d = , sig.level =, power = ,
           type = c("two.sample",
                    "one.sample",
                    "paired"))
```

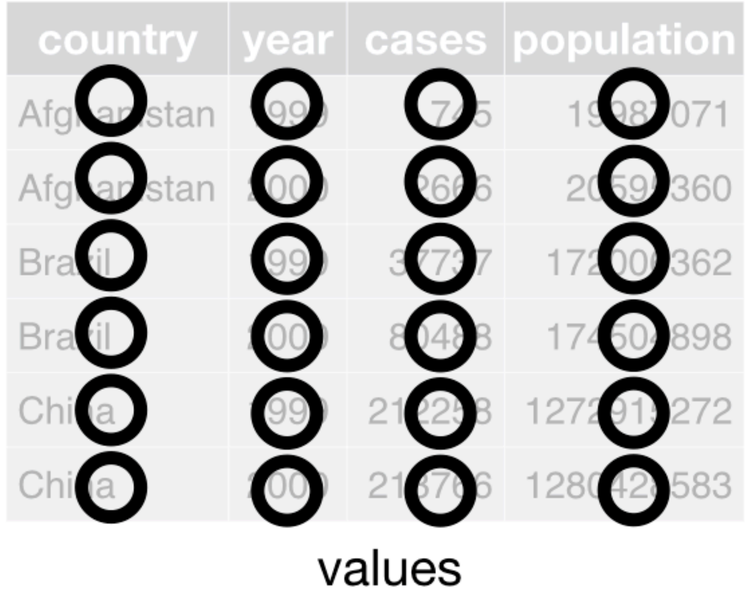
Experimental Design

- **Three organisms** – E. coli, S. aureus, B. subtilus
- **Two treatments** - Antibiotic, None
- **Experiment** - 5 replicates
- **Repeat experiment** - 3 times



Refresher on tidy data





Wide format

Treatment	1_Ecoli	1_Saureus	1_Bsubtilis	2_Ecoli	2_Saureus	2_Bsubtilis
Antibiotic	285	240	312	362	244	415
Antibiotic	345	371	461	368	375	315
Antibiotic	298	337	352	287	228	370
Antibiotic	286	394	494	378	302	314
Antibiotic	354	213	311	363	349	303
None	146	286	340	228	284	363
None	180	300	285	246	262	381
None	137	279	271	166	266	325
None	179	253	355	226	270	398
None	168	272	424	175	258	336

Long format (tidy)

Treatment	Experiment	Organism	Count
Antibiotic	1	Ecoli	285
Antibiotic	1	Ecoli	345
Antibiotic	1	Ecoli	298
Antibiotic	1	Ecoli	286
Antibiotic	1	Ecoli	354
None	1	Ecoli	146
None	1	Ecoli	180
None	1	Ecoli	137
None	1	Ecoli	179
None	1	Ecoli	168

Tidy data

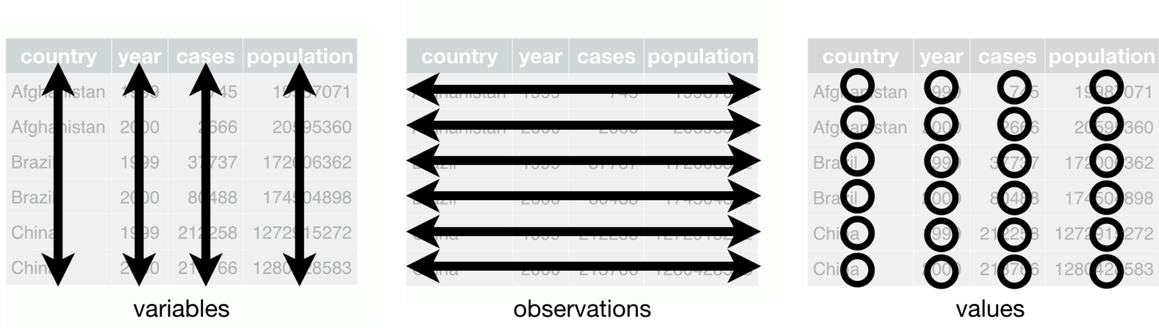
Treatment	Experiment	Organism	Count
Antibiotic	1	Ecoli	285
Antibiotic	1	Ecoli	345
Antibiotic	1	Ecoli	298
Antibiotic	1	Ecoli	286
Antibiotic	1	Ecoli	354
None	1	Ecoli	146
None	1	Ecoli	180
None	1	Ecoli	137
None	1	Ecoli	179
None	1	Ecoli	168

Organism	Treatment	Experiment	N	mean	sd	se
Ecoli	Antibiotic	1	5	313.6	33.32116445	14.90167776
Ecoli	Antibiotic	2	5	351.6	36.66469692	16.39695094
Ecoli	Antibiotic	3	5	346.2	44.80736547	20.03846301
Ecoli	None	1	5	162	19.55760722	8.746427842
Ecoli	None	2	5	208.2	35.42880184	15.84424186
Ecoli	None	3	5	177.6	40.14722905	17.95438665

Statistical tests

- shapiro.test - normal distribution
- t.test - T test
- aov - ANOVA
- TukeyHSD - Tukey post hoc test
- wilcox.test - Mann Whitney U test
- kruskal.test - Kruskal Wallis test

Much easier to run on ‘tidy data’



Shapiro Test

- Tests for normal distribution of data
- Need all values in a single column
- Tests the null hypothesis that data is normally distributed

Formula: shapiro.test(dataframe$column)

```
shapiro <- shapiro.test(data3$Count)
```

Student's t-Test

- Data normally distributed
- Compare differences between two means

Formula: t.test(y~x, data= dataframe) # where y is numeric and x is a binary factor

Formula: t.test(y1, y2, data= dataframe) # where y1 and y2 are numeric

Formula: t.test(y1, y2, paired = TRUE) # where y1 and y2 are numeric

```
tt <- t.test(mpg~am, data = mtcars)
```

Analysis of Variance (ANOVA)

- Data normally distributed
- Determine a significant difference between a group of means

Formula: aov(numerical~factor*factor2*factor3, data = dataframe)

ANOVA <- aov(mean~Organism*Treatment, data = data4)

Tukey's HSD

- Post hoc test
- Single step multiple comparison
- Determine means that differ significantly

Formula: TukeyHSD(aov_output)

```
ANOVA <- aov(mean~Organism*Treatment, data = data4)
  tukey <- TukeyHSD(ANOVA)
```

Wilcoxon/Mann-Whitney U test

- Non-parametric test
- Mann-Whitney U test
- Wilcoxon Signed Rank test

Formula: wilcox.test(y~A) # where y is numeric, A is binary factor

Formula: wilcox.test(y, x) # where x and y are numeric

Formula: wilcox.test(y1, y2) # where y1 and y2 are numeric

Kruskal Wallis test

- Non-parametric test
- one way ANOVA by ranks

Formula: kruskal.test(y~A) # where y is numeric and A is a factor

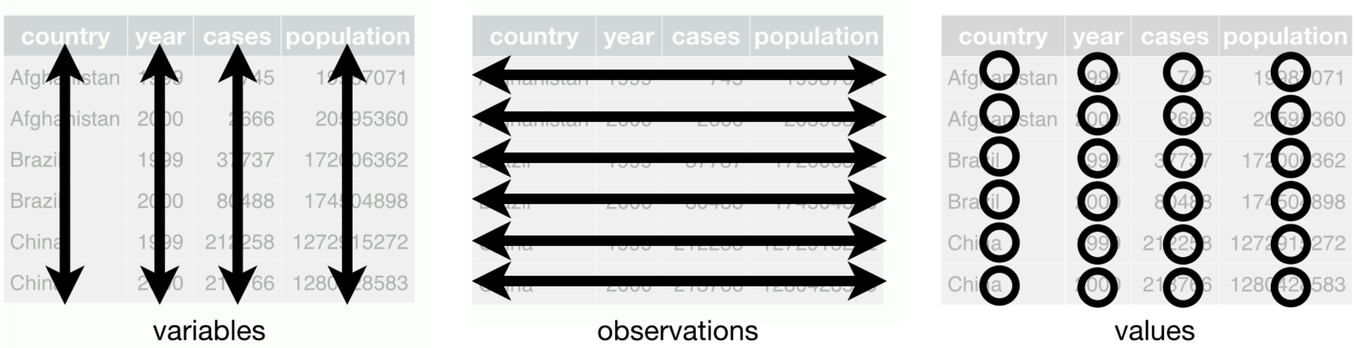
Transforming data

- mutate()
  - creates a new column from existing data

Formula: data1 <- data %>%
mutate(new_column = log2(column))

broom

- Makes statistical outputs ‘tidy’
  - export stats as a data frame
  - save as .csv



tidy_stats <- tidy(statisical_analysis)

Demo!

Markdown

- Lightweight markup language
- Easy formatting
- Easy to read
- Simple syntax

```
**Bold text**
*italics*
Plain text

#Big Header
##Smaller Header
###Smaller
####Even maller

Easily create lists

- item one
- item two
- item three

It's really easy to make tables

|header|header|header|
|---|---|---|
|value1|value2|value3|
```

Bold text

italics

Plain text

Big Header

Smaller Header

Smaller

Even maller

Easily create lists

- item one
- item two
- item three

It's really easy to make tables

header	header	header
value1	value2	value3

Rmarkdown

- Markdown
- LaTeX
- R code
- renders to .md, .pdf, .html
- Great for formatting dissertation

```
---
title: "Untitled"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

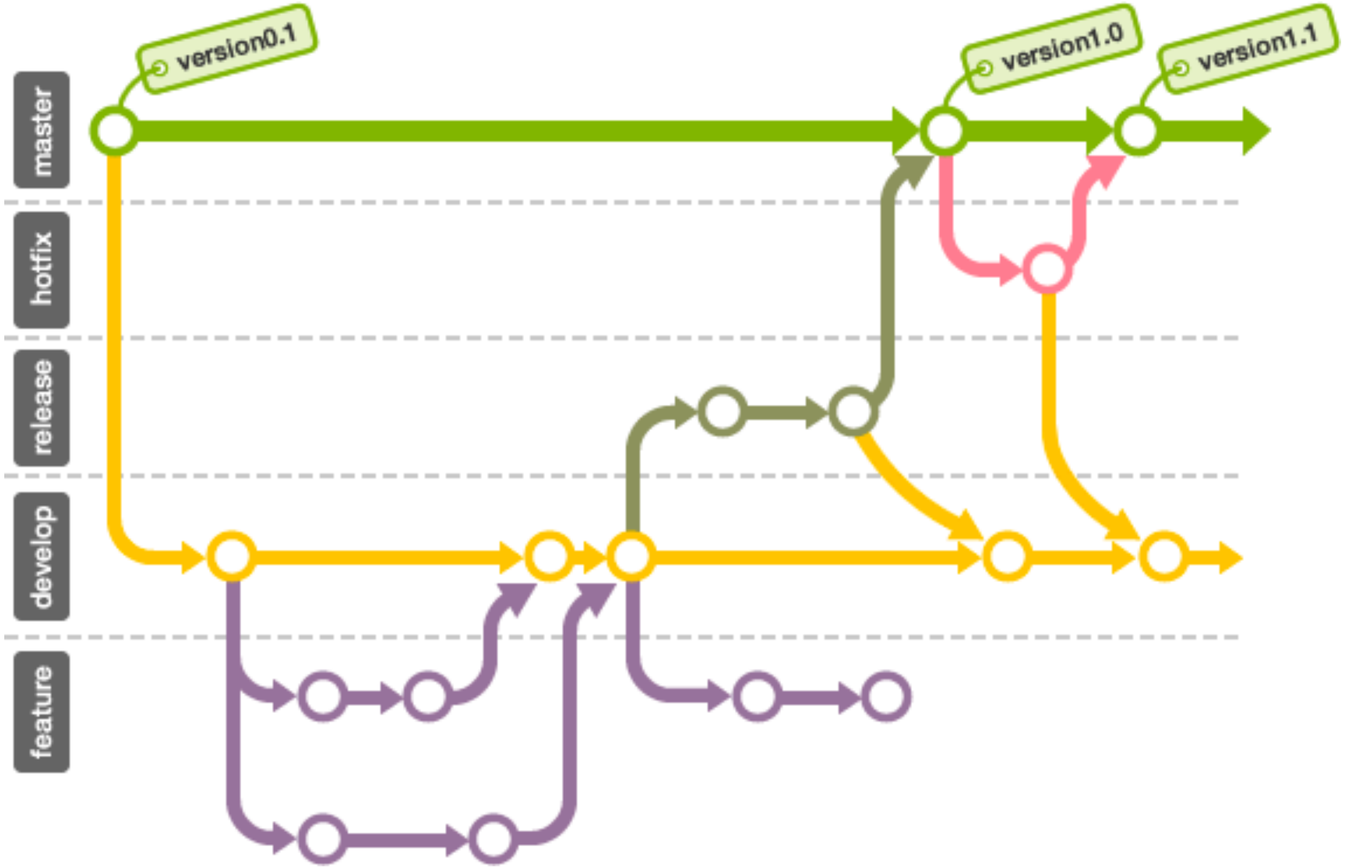
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
```

```
     speed           dist       
 Min.   : 4.0   Min.   :  2.00  
 1st Qu.:12.0   1st Qu.: 26.00  
 Median :15.0   Median : 36.00  
 Mean   :15.4   Mean   : 42.98  
 3rd Qu.:19.0   3rd Qu.: 56.00  
 Max.   :25.0   Max.   :120.00  
```

Git

- Version control
- Great for collaboration
- Graph theory tree model



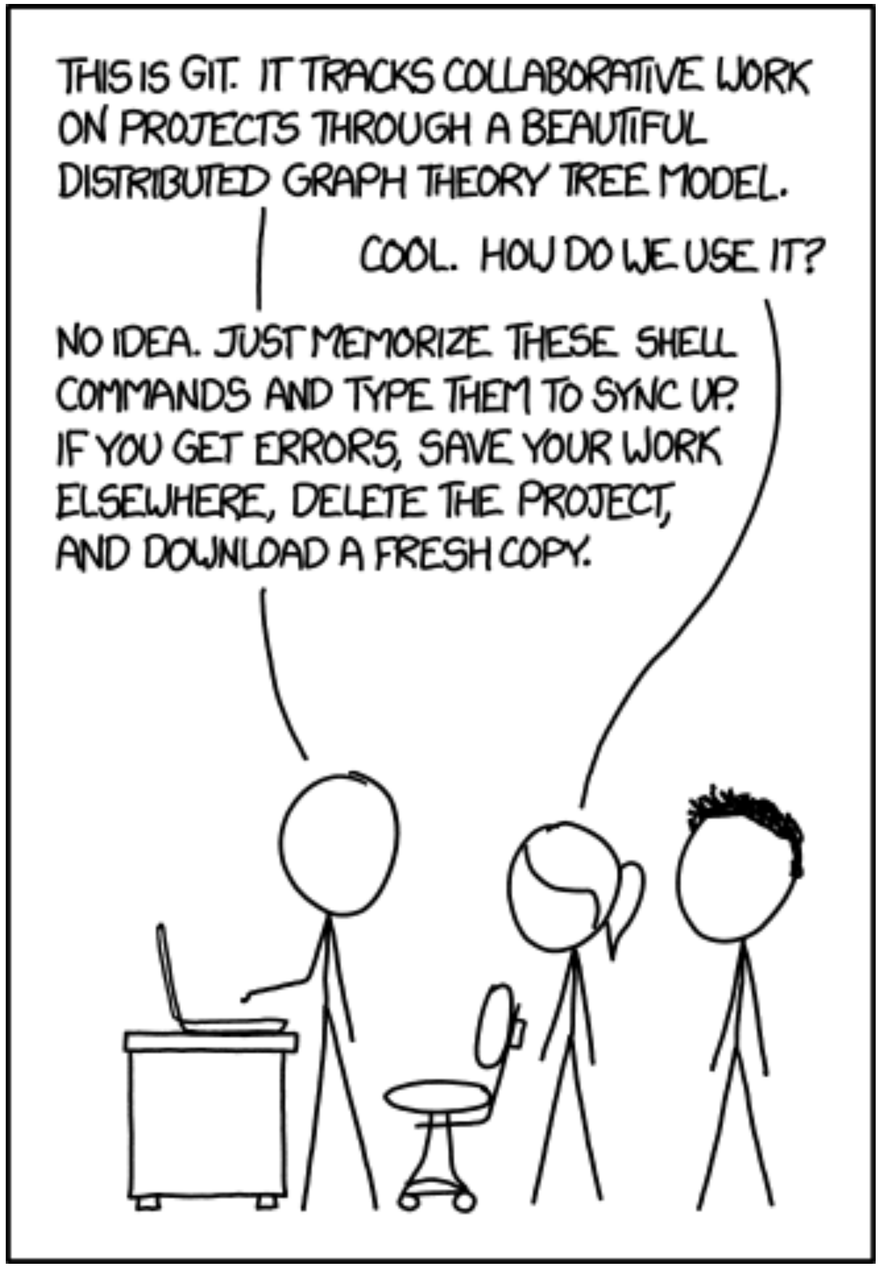
Companies & Projects Using Git

Google facebook Microsoft twitter LinkedIn NETFLIX PostgreSQL

android RAILS Qt GNOME eclipse

Git commands

- git clone <repository url>
- git pull
- git add -A
- git status
- git commit -m “insert memo here”
- git status
- git push



GitHub

- GitHub **IS NOT** Git
- Place to store your code
- Easy to track changes
- Showcase your work
- Collaboration

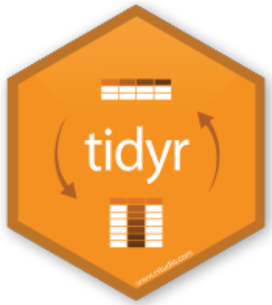
```
 p1 <- data %>%
   ggplot(aes(x = gdpPercap, y = lifeExp, size = pop, color =
continent, frame = year))+
-  geom_point(aes(text = paste ("country:",country)))+

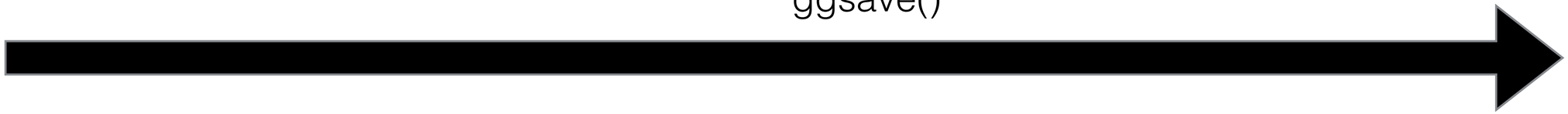
-  scale_x_log10()

-p1 + facet_wrap(~year)
```

```
65  p1 <- data %>%
66    ggplot(aes(x = gdpPercap, y = lifeExp, size = pop, color =
continent, frame = year))+
67  +  geom_point(aes(text = paste ("country:",country)))+ # add
country 'text"
68  +  scale_x_log10()+
69  +  ggtitle("Life Expectancy vs. GDP per Capita") +
70  +  xlab("GDP per Capita") +
71  +  ylab("Average Life Expectancy")
72  +
73  +p1
74
75  +# Saving plots
76  +ggsave(plot = p1, "gapminder.png", dpi = 600,
77  +       height = 5, width = 7, units = "in")
78
```

Data Analysis in the Tidyverse

Import	Tidy	Wrangle	Visualize	Stats	Communicate
					
read_csv() write_csv()	gather() spread() separate() unite()	filter() rename() select() mutate() group_by() summarise()	ggplot() geom_bar() geom_point() geom_boxplot() geom_hist() geom_violin() ggsave()	t.test() aov() TukeyHSD() tidy()	.md .Rmd .pdf .html



Thank you!