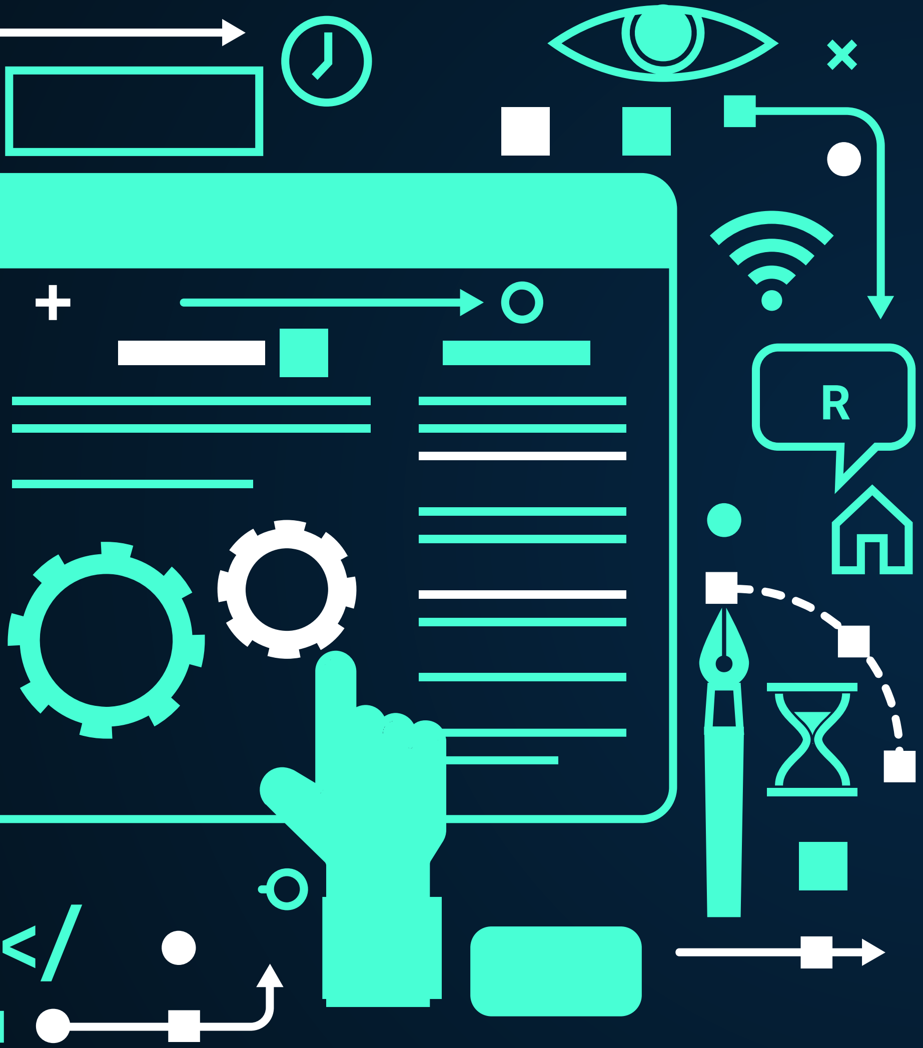


Statistical Bioinformatics Lab: ggplot2

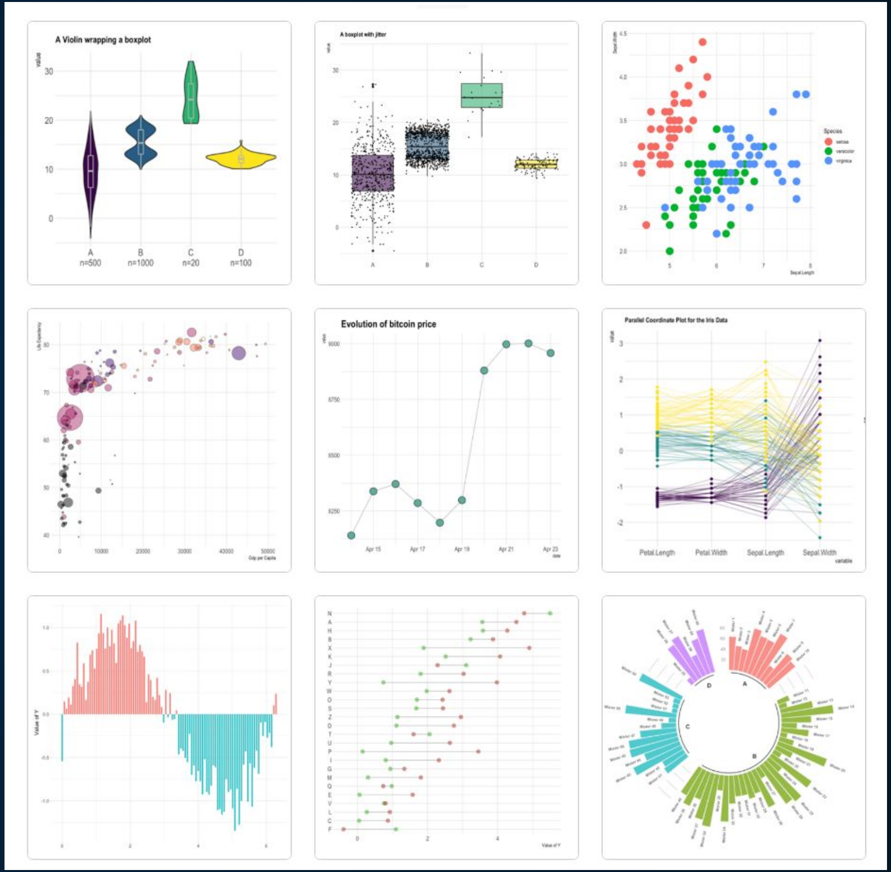
Feb 1, 2023

Mini Activity from Lab 3

- Read in the “gene_data.csv” and “sig_genes.csv” files.
- Replace the “_” in the colnames of gene_data with an empty space.
- Match the entrez ids (rownames of gene_data) with only genes from the sig_genes dataset that are significant (adj-pval < 0.1 and logFC > 2).
- Extract gene_data information for only the matches from sig_genes.
- Perform t-tests on all genes between the S and C groups.

GGPLOT2

- R package that enables you to create beautiful data visualizations.
- Can build almost any type of graph.
- https://ggplot2-book.org/



https://www.r-graph-gallery.com/ggplot2-package.html

GGPLOT2

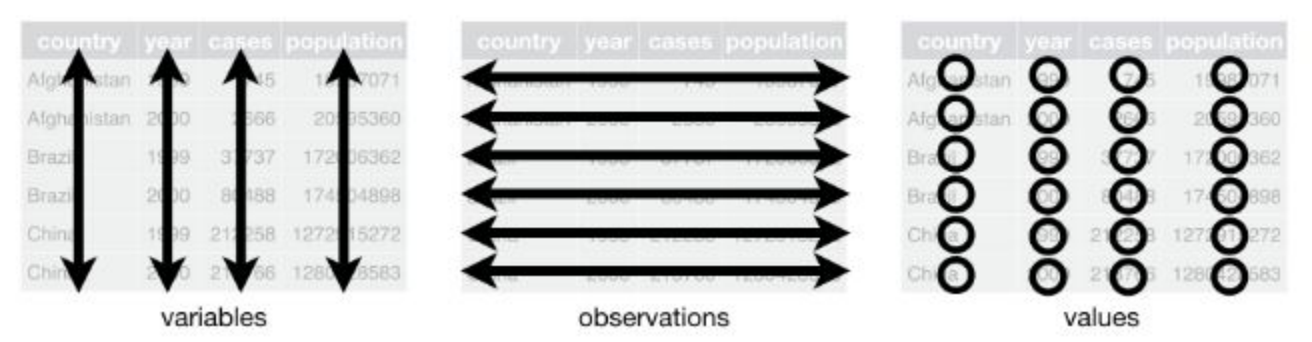
- Even art!

1
0
-1
y
-1.0
-0.5
0.0
0.5
1.0
x

https://www.r-graph-gallery.com/ggplot2-package.html

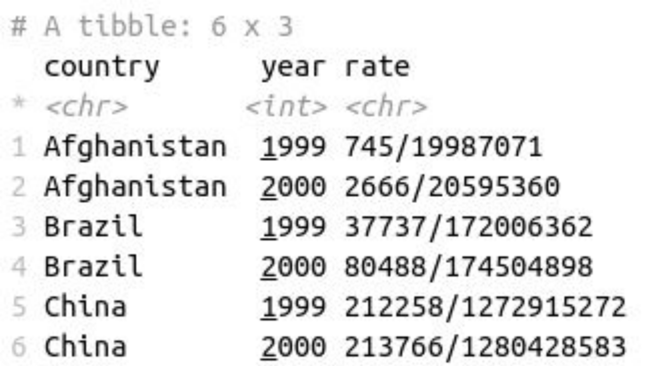
Data + ggplot2

- Works on “tidy” dataframes:
  - Each variable has its own column
  - Each observation has its own row
  - Each value has its own cell



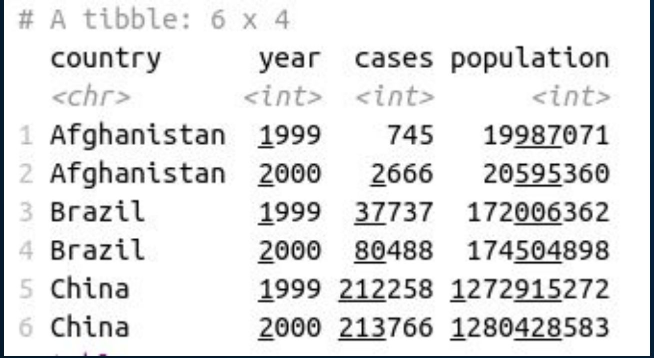
Tidy data

1



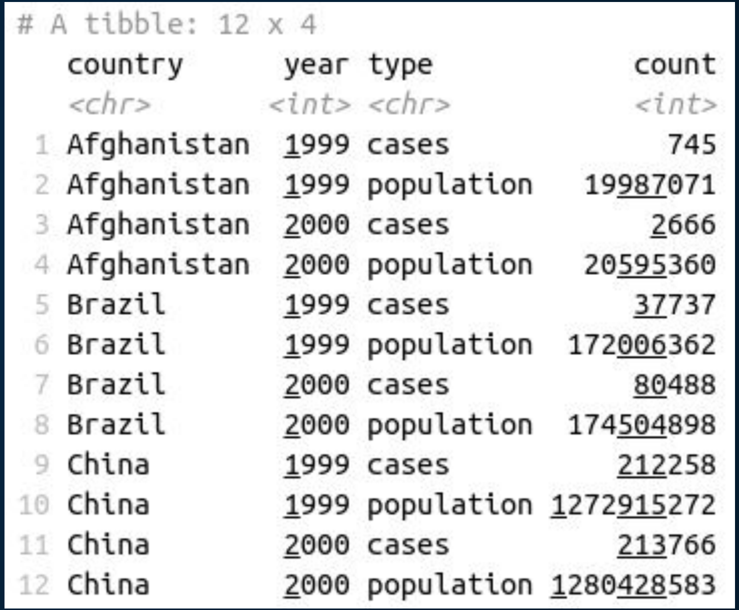
```
# A tibble: 6 x 3
  country      year rate
* <chr>       <int> <chr>
1 Afghanistan  1999 745/19987071
2 Afghanistan  2000 2666/20595360
3 Brazil       1999 37737/172006362
4 Brazil       2000 80488/174504898
5 China        1999 212258/1272915272
6 China        2000 213766/1280428583
```

2



```
# A tibble: 6 x 4
  country      year  cases population
  <chr>       <int>  <int>      <int>
1 Afghanistan  1999    745   19987071
2 Afghanistan  2000   2666   20595360
3 Brazil       1999  37737  172006362
4 Brazil       2000  80488  174504898
5 China        1999 212258 1272915272
6 China        2000 213766 1280428583
```

3



```
# A tibble: 12 x 4
   country      year type            count
   <chr>       <int> <chr>           <int>
 1 Afghanistan  1999 cases             745
 2 Afghanistan  1999 population   19987071
 3 Afghanistan  2000 cases            2666
 4 Afghanistan  2000 population   20595360
 5 Brazil       1999 cases           37737
 6 Brazil       1999 population  172006362
 7 Brazil       2000 cases           80488
 8 Brazil       2000 population  174504898
 9 China        1999 cases          212258
10 China        1999 population 1272915272
11 China        2000 cases          213766
12 China        2000 population 1280428583
```

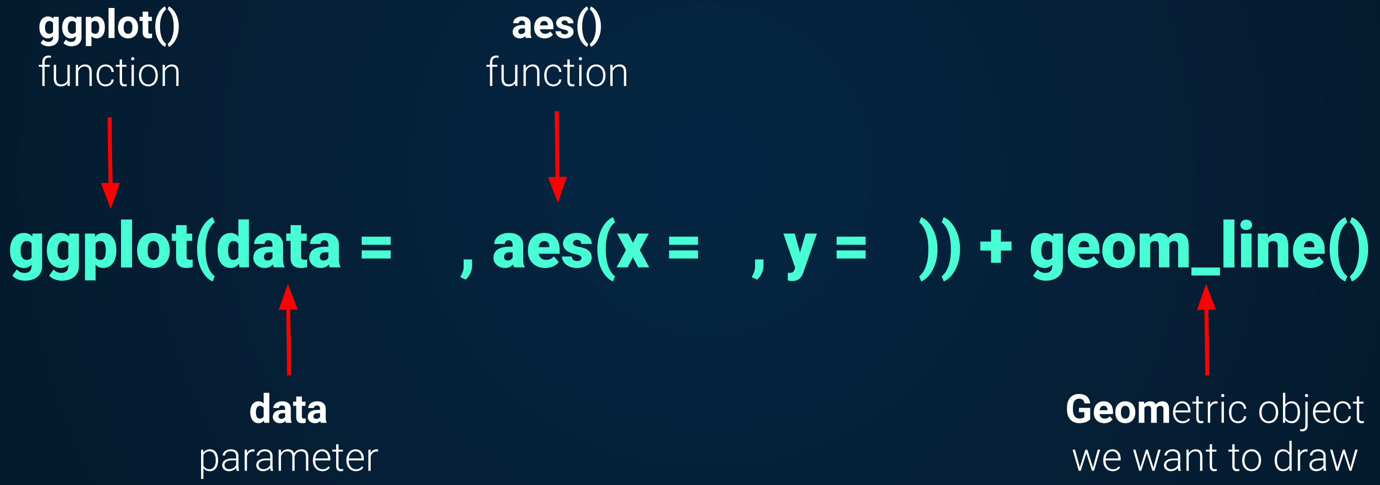
Which dataset is tidy?

Understanding ggplot2

- Built on the grammar of graphics - the idea that any plot can be created from the same set of components.
    - A **dataset**
    - A **coordinate system**
    - A set of **geoms** (visual representation of data points).

- Key to ggplot2?
    - Think of a figure in terms of layers.

Syntax of ggplot2



ggplot() function

aes() function

```
ggplot(data =    , aes(x =    , y =    )) + geom_line()
```

data parameter

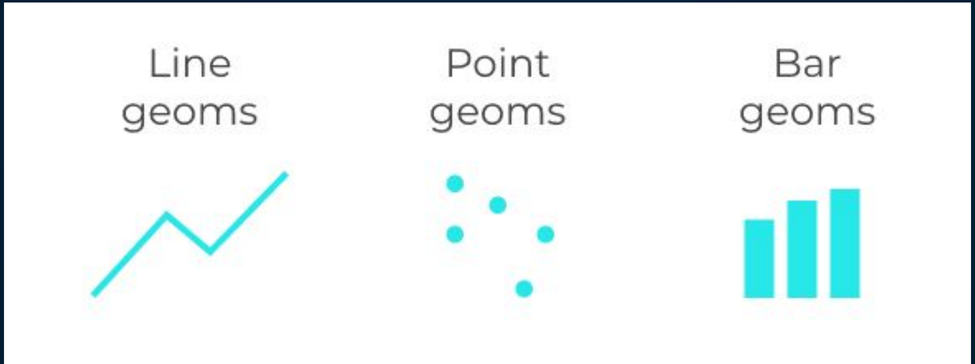
Geometric object we want to draw

Understanding ggplot2

- First step is to call the ggplot function.
- **ggplot(data = happy, aes(x = Country.or.region, y = Freedom.to.make.life.choices))**
    - Tells R we're creating a new plot.
    - Any arguments are the global options for the plot - apply to ALL layers.
    - First, tells ggplot what **data** to show (specifies the data.frame object).
    - Second, tells how variables in the data map to **aesthetic** properties of the figure (X and Y coordinates)
    - ggplot will look for the variables in the data.

Understanding ggplot2

- Need to tell ggplot how we want to visualize the data by adding a **geom** layer.
- **ggplot(data = happy, mapping = aes(x = Country.or.region, y = Freedom.to.make.life.choices)) + geom_point()**



Geoms 101

- Geometric objects including lines, points, boxes, polygons, etc.
- Scatter plot: geom_point()
- Line chart: geom_line()
- What about a bar chart?
  - Bar chart: geom_bar()
- THE TYPE OF GEOM USED DETERMINES THE TYPE OF VISUALIZATION!

Geoms 101

- Geoms have attributes.
- A position in a coordinate system (x/y)
- Color
- Size
- Shape, etc.
- These are all *aesthetic* attributes

The aes() function

- Maps the **data** to the visual objects (geoms) ~ map the **variables** from your data frame to the aesthetic attributes of the geometric objects in your plot.

Mini Activity Time!

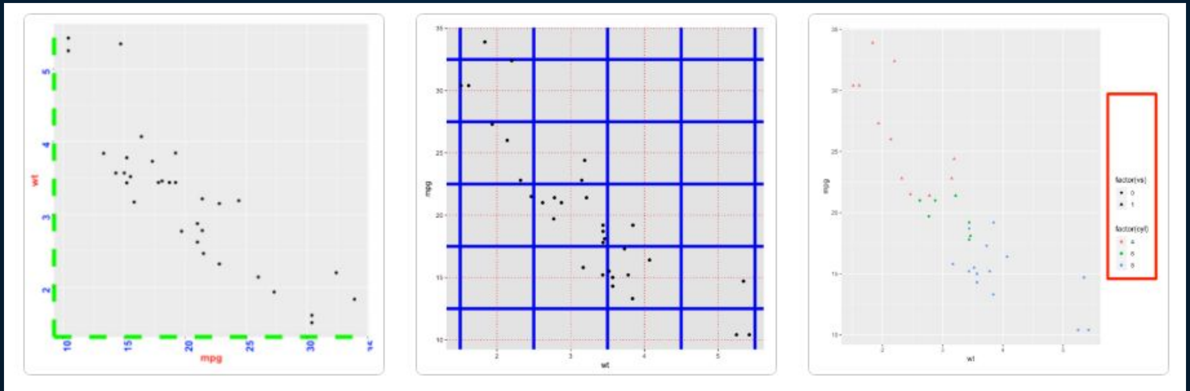
1. Read in the World Happiness Index data from 2019 (download from here https://drive.google.com/file/d/1CYZNhP9phRogwya5uCKUqbaHZqw9pjIA/view?usp=share_link).
2. Subset the data to the top 15 countries.
3. Inspect the data.
4. Make a preliminary scatterplot.

Modifying text in ggplot2

- For publication purposes, need to change up the plot.
- X axis should be “Country”, Y axis should be “Freedom” and the plot needs a title!
- Do this by adding some more layers with the **theme()** function - controls the axis text and overall text size + the **labs()** function - add labels for axes, plot title, and legend.

Modifying text in ggplot2

- theme() function allows for chart customization:
  - Axis ~ title, label, line and ticks
  - Background ~ background color and major and minor gridlines
  - Legend ~ position, text, symbols and more



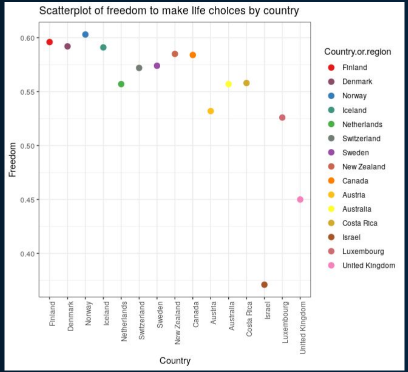
https://www.r-graph-gallery.com/ggplot2-package.html

Modify titles

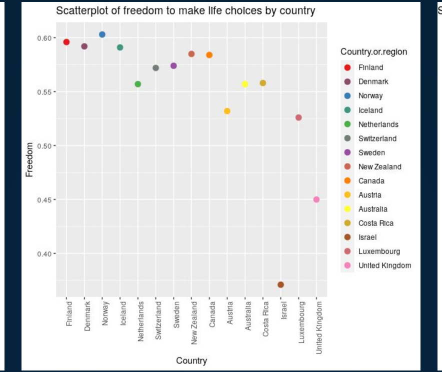
- ggplot(data = happy, mapping = aes(x = Country.or.region, y = Freedom.to.make.life.choices)) + geom_point() + **labs(x = "Country", y = "Freedom", title = "Figure 1") + theme(axis.text.x = element_text(angle=90, hjust = 1))**

Themes

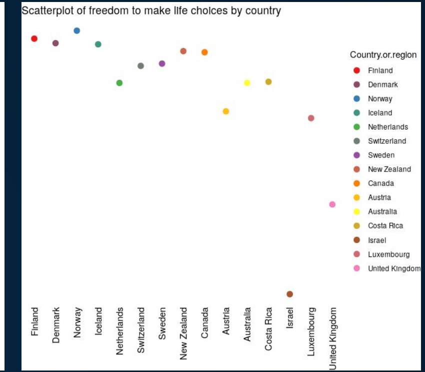
theme_bw()



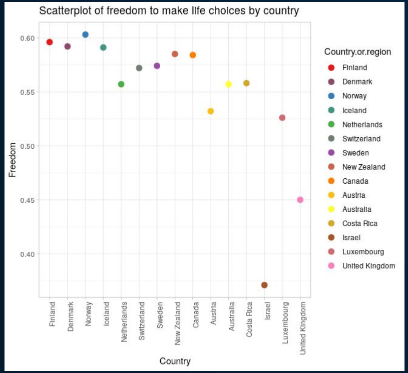
default

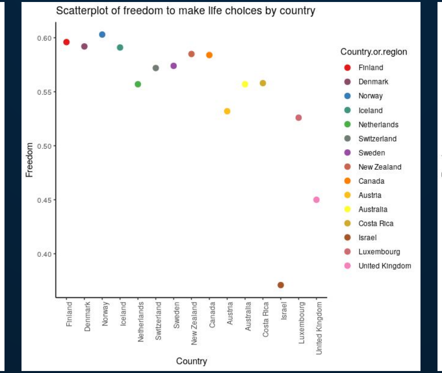


theme_void()



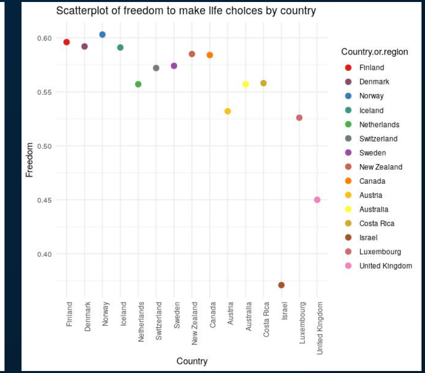
theme_light()





theme_classic()

theme_minimal()



Add themes

- ggplot(data = happy, mapping = aes(x = Country.or.region, y = Freedom.to.make.life.choices)) + geom_point() + labs(x = "Country", y = "Freedom", title = "Figure 1") + theme(axis.text.x = element_text(angle=90, hjust = 1)) + **theme_bw()**

Make it interactive

- Linked with the “plotly” R package
- Super easy to make your ggplot chart into an interactive experience!
- ggplotly()

```
library(plotly)

p <- ggplot(data = happy, aes(x = Country.or.region, y = Freedom.to.make.life.choices, color = Country.or.region)) +
  geom_point(size = 3)

int_plot <- ggplotly(p)

htmlwidgets::saveWidget(int_plot, "interactive_plot.html")
```

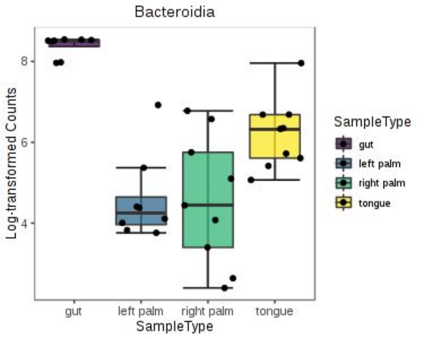
Mini Activity Time!

- Add a new column to the data frame specifying which continent the country belongs to using the “countrycode” R package
    - Tips: ?countrycode ?codelist
- Color the scatterplot by continent!
- Change the font size of the axis labels to 15 and colored blue.
- Change the size of the points according to any variable in the dataframe.
- Save the plot as a png using ggsave.
- Now make it interactive and save it!
- Make a bar-chart of any other variable you find interesting versus the country.

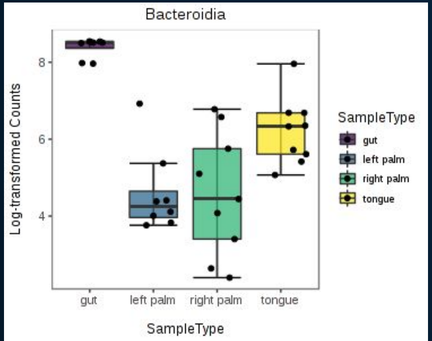
Now you know R! Sorta...

- Comments are **very** helpful – history of what you did and why.
- Use spaces and tabs! **Don’t make ugly code.**
  - Essential style guide: https://google.github.io/styleguide/Rguide.xml
- R Studio cheat sheets: RStudio Cheat Sheets
- If you’ve retained nothing: R for cats · and cat lovers

Axis spacing in ggplot2



labs(y="Log-transformed Counts", x=variable)



Add a new line before the X axis label and after the Y axis label.
labs(y="Log-transformed Counts\n", x=paste0("\n",variable))