**2**

**Grammar of graphics and visual components**

**Show** Slide 1

**Lesson Objectives**

**Show Slide 1**

By the end of this lesson, you will be able to:

1. Topic A: More on grammar of graphics
   1. Layers
   2. Scales, Legends, and Axes
   3. Coordinates
   4. Geoms and Statistical Summaries
2. Topic B: Faceting to make multiplots and divide the data into subplots.
3. Topic C: Using and changing styles and colors
   1. Using Colors in plots
   2. Themes and changing the appearance of graphs

**Lesson time: 2 hrs 30 mins**

**Logistics**

Start with summarizing the parts that were learnt in the previous lessons. Remind them about what the

terms geometric objects, grammar of graphics etc. means, and what kind of plots we created in the first

lesson. Mention that we will be going into greater details on some of the aspects touched upon in the

previous lesson and introduce some more new features.

**Topic A: More on Grammar of graphics**

**Time: 50 mins**

**Show** Slide X

Grammar of graphics is the language used to describe various components of a graphic which represents the data in a visualization. In this topic, we learn more about this and then use it to make plots. We encountered some of the terms used in Grammar of graphics in the previous lessons.

**Discussion**

What are some of the grammar of graphics we learnt and used in the previous class?

**Answer**

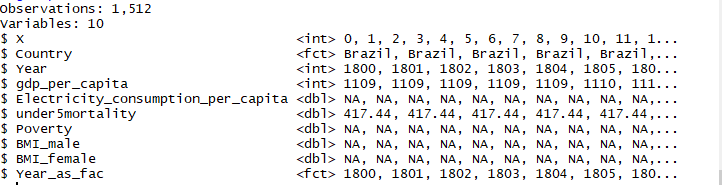
Rebinning, theme

**Show** Slide XX

**Subtopic: Layers:**

Every plot in ggplot2 is built up as a layer. Layers are made of geometric objects (geoms), their statistical transformations(stats), and thematic aspects. Hence, each plot can be thought of as a separate variable in itself. Aesthetic mappings, defined with aes (), describe how variables are mapped to visual properties or **aesthetics**.

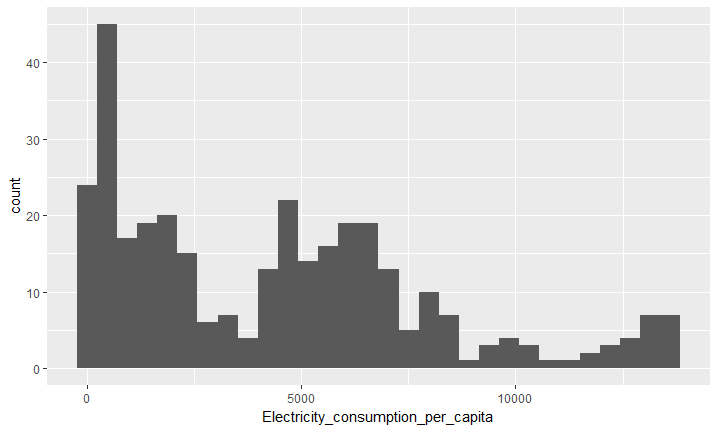
Let’s see an example. Let’s use the dataset “gapminder”. You can see the variables available in the snippet below. It has different variables for different countries, for example, “GDP per Capita”, “Electricity consumption per capita”, and so on.



Let’s look at the histogram for Electricity consumption per capita. We are not analyzing this histogram so don’t worry about the features.

p1 <- ggplot(df,aes(x=Electricity\_consumption\_per\_capita))

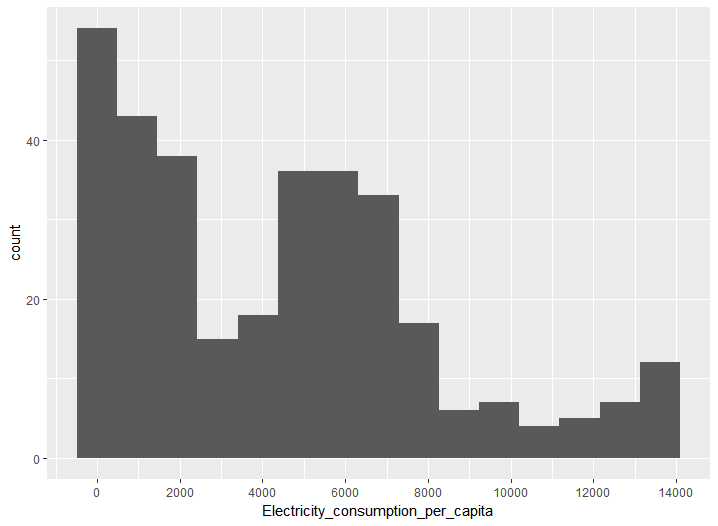
p2 <- p1+geom\_histogram()



Now, lets add another layer and rebin the histogram, saving it in another plot p3 and display it.

p3 <- p1+geom\_histogram(bins=15)

p3



**Exercise**: Using more Layers

**Aim**: To use layers to customize a histogram

**Steps for completion**:

Use the codes above

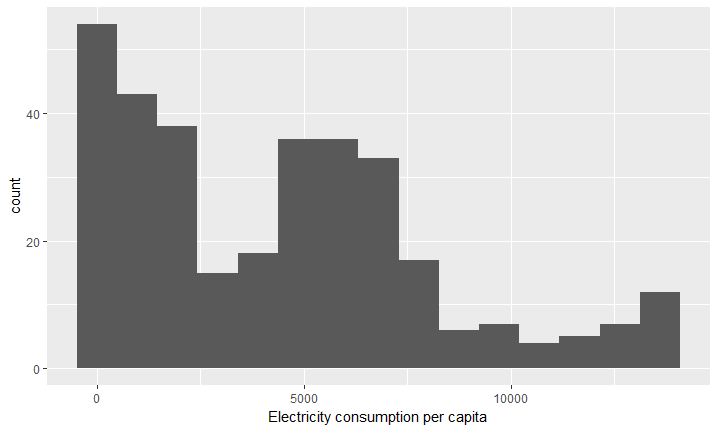
Add another layer to change the labelling of the axes (remove the “\_” in the variable name) and save it in a plot - p4.

Try plotting them individually as “p1”, “p2”, “p3”, “p4” on the command line.

Outcome:

Four plots.

Final plot should look like this:



**Discussion:** Why can’t we display “p1”?

**Answer:** Because it is just the command which loads the data and we haven’t defined a geometric object yet, which we do when we define “p2”.

The takeaway from this exercise is that all the 4 plots, p2,p3,p4 are variables and can be displayed independent of each other. This is what makes layers very powerful, as the individual layers are also saved, and one can keep track of the changes and go back to edit one of the layers if needed. We shall continue to use layers in our coding exercises which follow.

1. **Scales,Axes and Legends:** Scales map values in the data space to values in an aesthetic space, whether its color, shape or size. Scales are used to change legend or axes, providing an inverse mapping and enable us to understand the data from the graphic itself. In the previous example, when we plotted the histogram, what ggplot actually did was to apply a default scale to describe the x and y-axes. But we can modify that.

To modify scales the following commands are used:

Continuous variables: **scale\_x\_continuous** (x-axis), **scale\_y\_continuous**(y-axis)

Discrete variables: **scale\_x\_discrete**(x-axis); **scale\_y\_discrete**(y-axis).

Some of the common things we may like to change are:

* **name:** the first argument gives the axis or legend title
* **limits:** the minimum and maximum of the scale
* **breaks:** the points along the scale where labels should appear
* **labels:** the labels that appear at each break

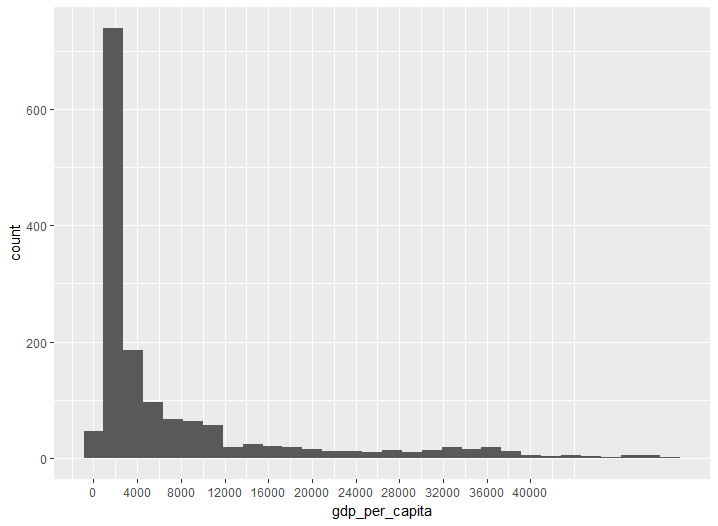
Let’s use these commands.

Exercise: Explore “scale\_x\_continuous” and “scale\_x\_discrete”

Aim: To explore some of the options available in “scale\_x\_continuous” and “scale\_x\_discrete”.

Steps for completion:

* 1. Use the same dataset as before and plot the “gdp\_per\_capita” as a histogram
  2. Plot p1. Can you tell where the maximum occurs? No. We need a finer labelling.
  3. Study and view the options in scale\_x\_continuous(Note: Use ?scale\_x\_continuous on the command line, to see the options)
  4. Add the layer: scale\_x\_continuous(breaks=seq(0,40000,4000) ) (See Slide). This command defines a range between 0, 40000, with breaks of 4000.
  5. What is the maximum GDP per capita? What type of histogram is this?



Outcome :

Code, Histogram, and Answer to Q6:

Maximum “GDP per Capita” is about 2000. It’s a right skewed histogram

**Instructor Note:** See the documentation of the scales package for more details.

Highlight the importance of scales, and why “breaks” is important as brought up in Q3.

**URL:** [**https://cran.r-project.org/web/packages/scales/scales.pdf**](https://cran.r-project.org/web/packages/scales/scales.pdf)

**Subtopic: Coordinates**

* Cartesian- **coord\_cartesian():** By default ggplot uses a cartesian coordinate system but sometimes one might want to use polar coordinates, which is another way to look at fractions. So one use could be to transform a bar chart into a pie chart.
* Polar- **coord\_polar():** In ggplot2, the cartesian coordinates x,y become polar coordinates theta, and r. You can specify which coordinate is theta and which r.

Let’s do an exercise to understand polar coordinates.

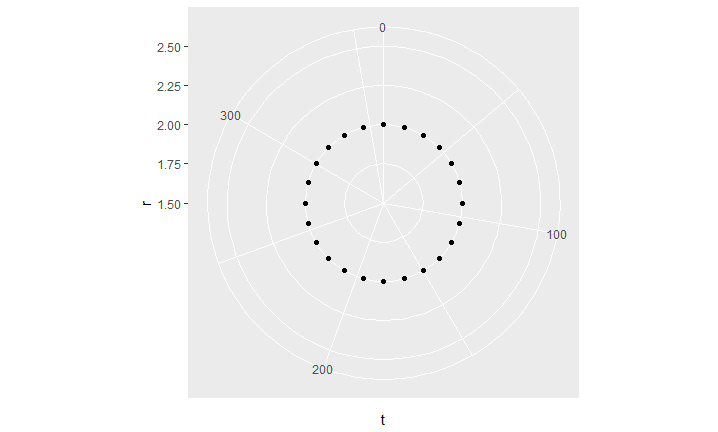
Exercise: Understanding Polar Coordinates

Aim: To generate some numbers and plot them in cartesian and polar coordinates

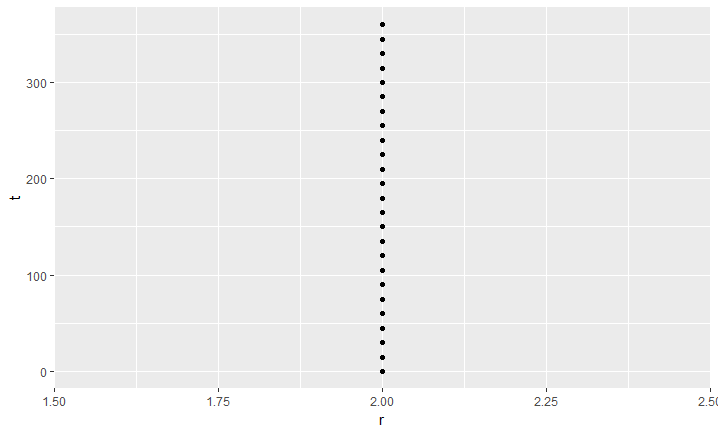
Steps for completion:

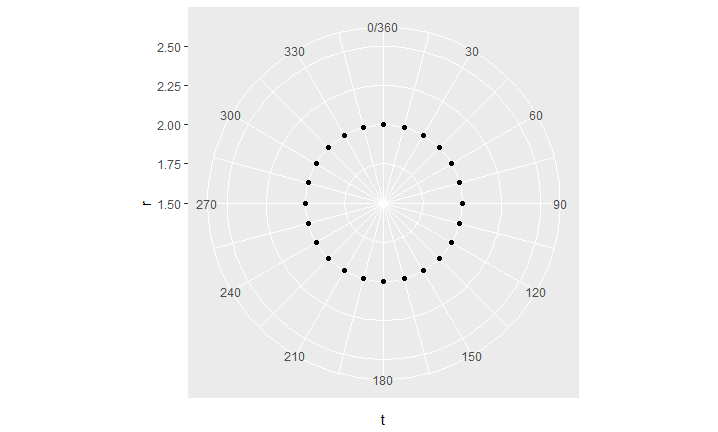
1. Generate angles between 0-360 in intervals of 15.
2. Generate another variable the radius r.
3. Use qplot to plot the vectors in cartesian coordinates (Plot1). What do you see? Write your answer in the code in comments.
4. Use qplot to plot the vectors in polar coordinates. What do you see? Write your answer.
5. Your first plot should look like Plot 1.
6. Use scale to change the labelling to look like Plot 2.

Outcome:

1. Plots 1,2 and Final Plot
2. Answers in comments in the code. Plot2

Plot 1



Final Plot:

Discussion: As one can see angles are much better viewed in polar coordinates and the distance of the points from the centre show us that the radius is a constant for this data. It can be used to also depict fractions, which we shall also work on. The center corresponds to 1.5, and the circles represent the different radii, 1.72, 2.0 and so on.

Changing the formatting of the labels: Various label formats are available and one can change the label formats as well. The following label formats are available:

* scales::comma\_format() adds commas to make it easier to read large numbers.
* scales::unit\_format(unit, scale) adds a unit suffix, optionally scaling.
* scales::dollar\_format(prefix, suffix) displays currency values, rounding to two decimal places and adding a prefix or suffix.
* scales::wrap\_format() wraps long labels into multiple lines.

We shall combine all the above commands to make a single plot:

Activity A: Applying Grammar of graphics to create a complex visualization

Scenario: Sometimes one needs to customize a plot and change scales and coordinate types. Say, you have profits in billions or trillions of dollars, you would want to represent a big number as 1M, or 2M and so on. Or, you might want to view percentages of profits per month as a pie-chart. In that case you would need to change your cartesian to polar co-ordinates.

Aim:

To use Grammar of graphics to create the given histogram

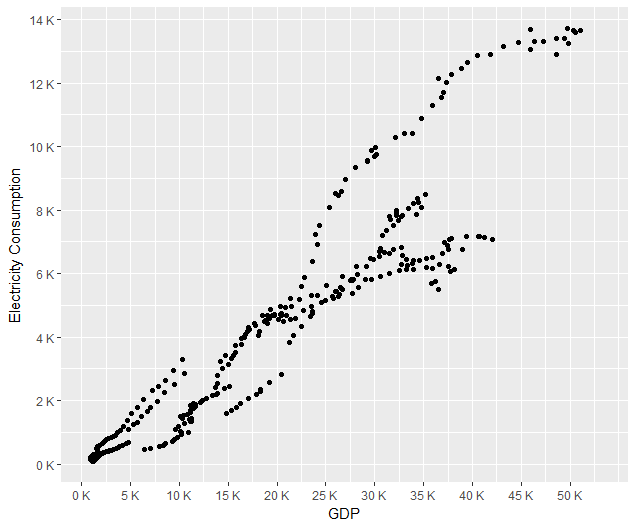
Steps for completion:

1. Use the commands we just explored to create the histogram on the slide.
2. You can use the help command to explore the options.
3. For changing scales you have to use one of the label formats above.

(Use labels=scales::unit\_format("K", 1e-3)) for the labelling

Outcome:

A code and scatter plot which matches this plot:



**Topic B: Facets**

**Time: 30 mins**

In data visualization, we sometimes need to compare different groups, and look at data alongside each other for the different groups. One way of do this is to create a subplot for each group. These kinds of plots are known as Trellis displays. In ggplot2, they’re called facets. Facets divide the data by some discrete or categorical variable and displays the same type of a graph for each data subset.

Let’s look at Electricity consumption vs GDP for different countries, which we just did in the previous activity.

Discussion: What kind of correlation did you see in that plot?

Answer: The Electricity consumption is higher as the GDP grows.

Discussion: But we don’t know which Country has the highest GDP or Electricity consumption. Let’s split the data now.

Exercise: Using facets to split data

Aim:

To plot subsets of the data as separate subplots

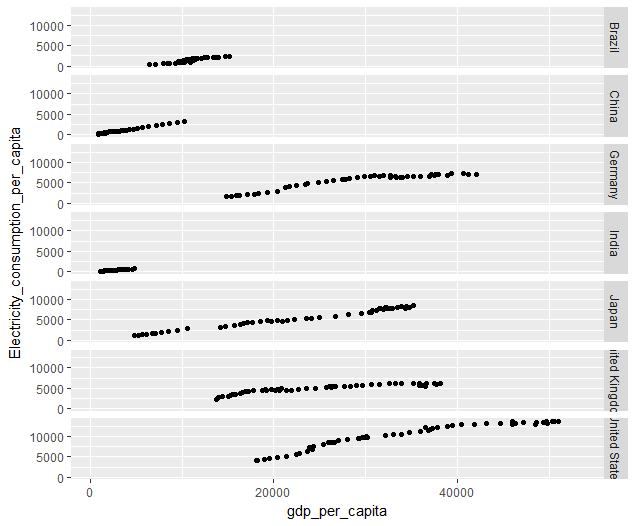
Steps for completion:

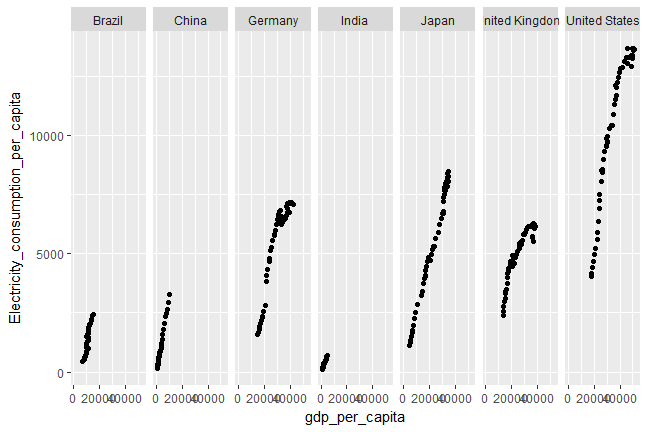
* 1. Use dataset “gapminder.csv”
  2. Create a subset of the data
  3. Use facet\_grid() or facet\_wrap(), and specify the variables on which to split.
  4. We have to plot Electricity Consumption vs GDP for each country separately. So our split variable is “Country”.

p <- ggplot(df, aes(x=gdp\_per\_capita, y=Electricity\_consumption\_per\_capita)) +

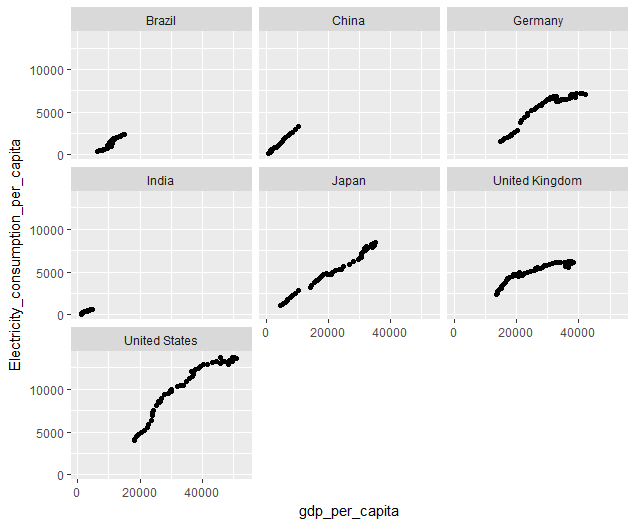
geom\_point()

* 1. p + facet\_grid(Country ~ .) - Horizontally split
  2. p + facet\_grid(. ~ Country) - Vertically Arranged





Discussion: Another command can be used which is facet\_wrap(). With facet\_wrap(), the subplots are laid out horizontally and wrap around, like words on a page. The plot would look like this will facet\_wrap()



What does one notice about the trend? Ask students first .

**Answer:** The Electricity consumption is higher as the GDP grows for all countries, but now we can see that the electricity consumption is highest for United States. We also see that the growth is different for different countries and one can look at more details if required.

**Activity for Topic B.**

1. **Topic C: Using and changing style and colors**

**Time: 1hr**

**Subtopic: using colors in plots**

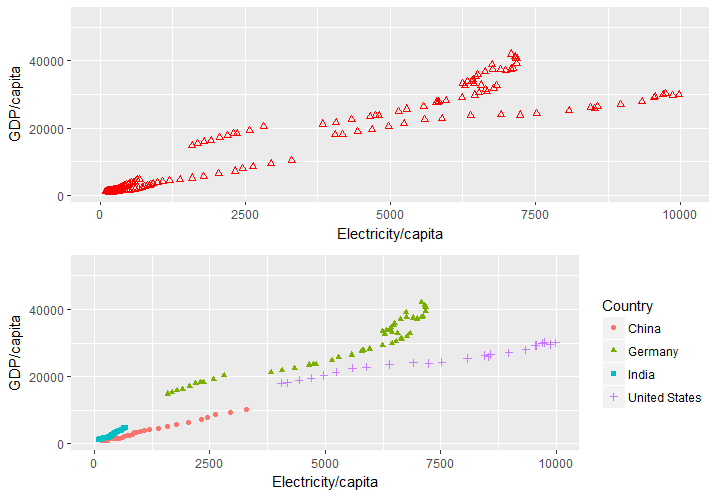
Instead of faceting, we could also produce a color differentiated plot. Say, we plot the Electricity consumption vs GDP but use different colors or shapes for the countries.

Exercise: Use color to group points by a variable

Aim: To produce a color differentiated scatter plot with resepct to a third variable.

Steps for completion:

1. Choose a subset of data set 1 (gapminder) and select a few countries.
2. Do a scatter plot of the two variables and change the X and Y titles.
3. Then change the color and shape of the points – Plot 1
4. Now group points by “Country” mapped by colour and shape – Plot 3 and Plot 4



**Acitivity C:**

Scenario: A loan company has given loan amounts to people with different features, for example, employment status, home ownership, credit grade, etc. You want to see the relationship of some of those variables.

Aim: To view the distribution of loan amount vs home ownership color differentiated by Credit grade.

Steps for Completion:

1. Use the dataset : LoanStatus
2. Clean the dataset slightly (we want to remove the “NONE” and “NA” cases)

dfn <- na.omit(dfn)

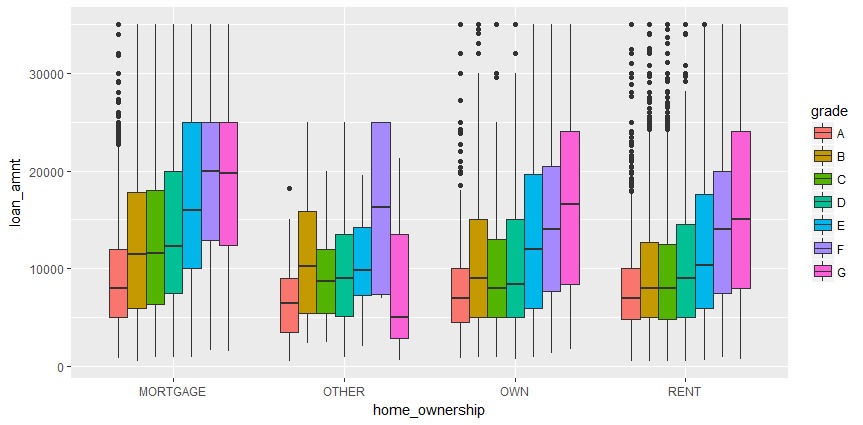
dfn <- subset(dfn, !dfn$home\_ownership %in% c("NONE"))

1. Create a box plot showing loan amount vs. home\_ownership.
2. Answer the following questions:
   1. Which Credit grades have taken highest median loan amount? Which credit grades correspond to lowest loan amount.
   2. For credit grade F, are the loan amounts higher or lower if the person has a mortgage vs. the person who rents a home?
   3. By how much does the loan amount differ for people who own a mortgage for credit grades A vs credit grade G.

(Note: You have to do a finer labelling to answer this question)

Outcome:

Code, Plot and answers to Q4.



**Subtopic: Themes and changing the appearance of graphs**

The grammar of graphics that underlies ggplot2 is concerned with how data is processed and displayed—it’s not concerned with things like fonts, background colors, and so on. To tune the appearance of these things, ggplot2’s theming system provides control over the appearance of non-data elements. We will be touching upon a few thematic aspects in this course. It would not be possible to go over all thematic options. For a more complete list of options use the help guide, by typing ?theme or look at the ggplot2 manual.

Themes can be changed in multiple ways:

* Change it individually for each plot
* Use a pre-defined Theme
* Define your own theme and use it for all your plots.

Note: The exercise below is intended to introduce you to different themes elements. The plot we produce at the end is not necessarily the best visual, but is just to illustrate the options.

Exercise:

Using theme to customize a plot

Aim:

To use thematic commands to customize and improve a plot appearance.

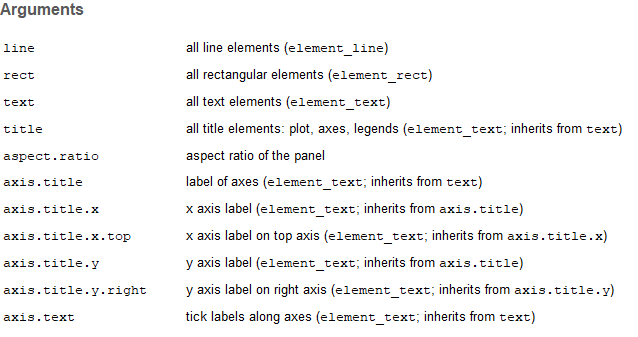
Steps for Completion:

1. Lets use the “HollywoodMovies” data and do a bar chart of movie titles and world Gross amount. Since we have too many we make a subset:

dfn <- subset(HollywoodMovies, Genre %in% c("Action","Adventure","Comedy","Drama","Romance")

& LeadStudio %in% c("Fox","Sony","Columbia","Paramount","Disney"))

1. Let’s pre-define a theme.
2. Type ?theme (Alternatively while you type “theme” R shows you the options available - See snapshots below)
3. Some of the options available.

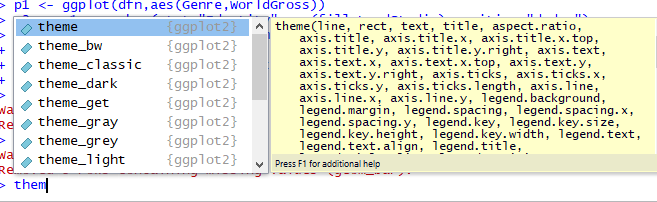


1. Note that axis.title is of type “element\_text”, which in turn inherits from “text”
2. Hence, we will change aspects of element\_text. Type ? element\_text to bring up the options.

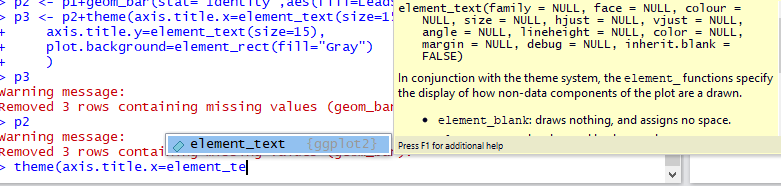
Use the table below to change some of the thematic aspects:

|  |  |  |
| --- | --- | --- |
| Argument | Type | Values |
| axis.title.x | Element\_text | size=15, family=”Helvetica”, angle=45 |
| axis.title.y | Element\_text | size=15, family=”Helvetica”, angle=45 |
| Panel.grid.major | Element\_line | color=”gray87” |
| Panel.background | Element\_rect | Fill=“Beige” |
| plot.background | Element\_rect | Fill=”Gray”,size=20 |

R will automatically autofill and display options before you finish typing:



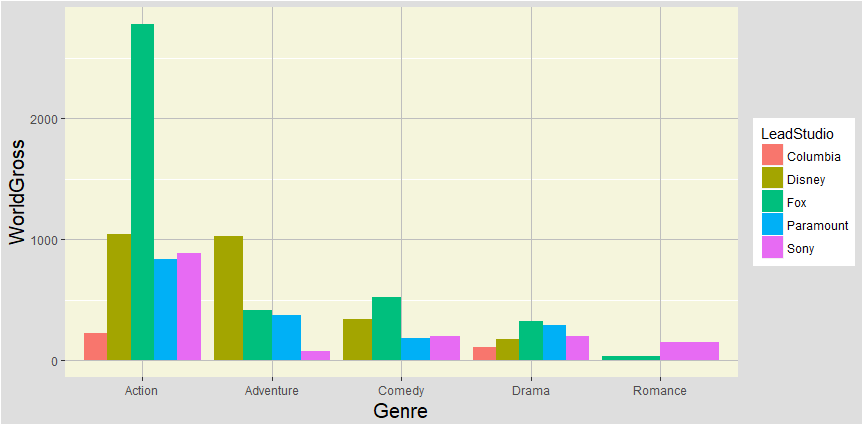
Setting the axis.title options brings up these options for element\_text



Discussion:

This plot shows use the World Grossed amount for the different Genres for the different production houses. One can see that Action has the maximum earnings, and most of them are produced by Fox. Disney also earns more than other houses. Disney earns more than Fox for the Adventure Genre.

Outcome:



Discussion:

List some good and bad features about this plot

Answer:

**Good:** The X and Y titles are visible. The plot is color differentiated making it easy to distinguish between the different studios. The grid lines are visible allowing us to read the values.

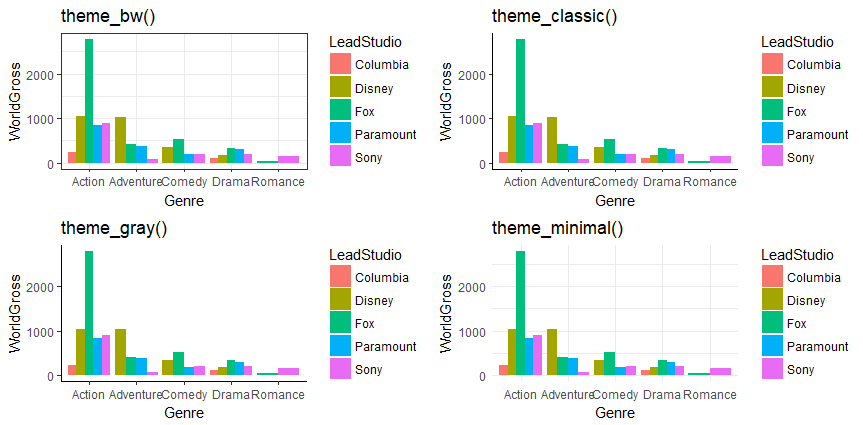
**Bad:** This plot has too many bright colors which is distracting. Panel background is better visualized as white. The labels are not very visible and would be nicer to have bigger font sizes.

Now, you have the tools to customize and improve the plot.

**Instructor Note:**

Color palette can be found here : <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>

* **Using a predefined Theme**: There are some predefined themes that one can use. The same plots produced by using different themes are as below:



**Exercise:** Using or setting your own theme globally

Aim:To set your own theme for an individual plot or set it globally

Steps for completion:

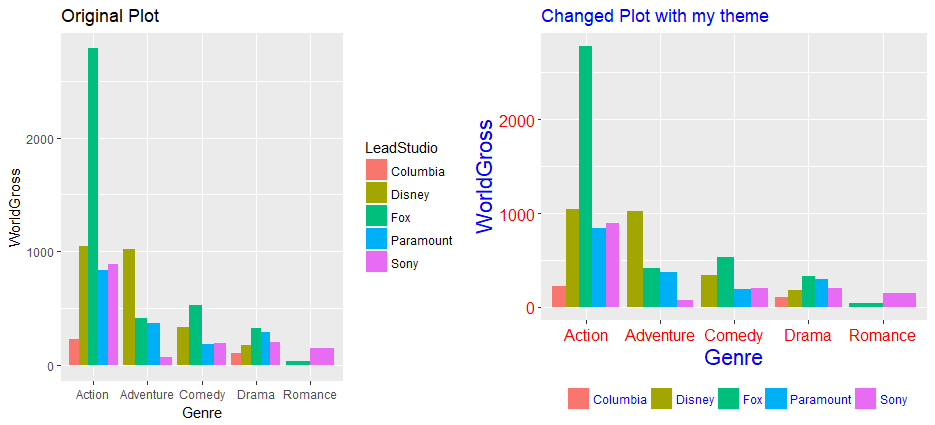
1. Setup your theme:
   1. Change the legend title, and position
   2. Change the axes title colors, and relative size (1.5). Using “text” changes the colors of the axes titles and legend text, but not the label text.
   3. Change the axes label text sizes: Use “axis.text”. (Note: If you want to change the x and y axes separately use, axis.text.x and axis.text.y)

mytheme <- theme(text = element\_text(colour="Blue"), axis.text = element\_text(size=12),axis.title = element\_text(size = rel(1.5)))

1. Use your theme with an individual plot: p2 <- p1+mytheme
2. Or set it globally:

theme\_set(mytheme)

Outcome: Code and plot with mytheme



Discussion: List some of the differences between the original and changed plot.

Answer: Students should list the differences mentioning the name of the variable changed and what the change is, e.g., “axis labels are changed from black to red” or “the legend in the new plot is at the bottom and does not have a title”

Instructor Note:

Point to Github Code and explain the parts of each.

**Logistics**

**Summary**

**Practice Questions**

**Show**

**Answers**