

# INTRODUCTION TO DATA SCIENCE

## Lecture 9

Dr. Ibrahim Radwan

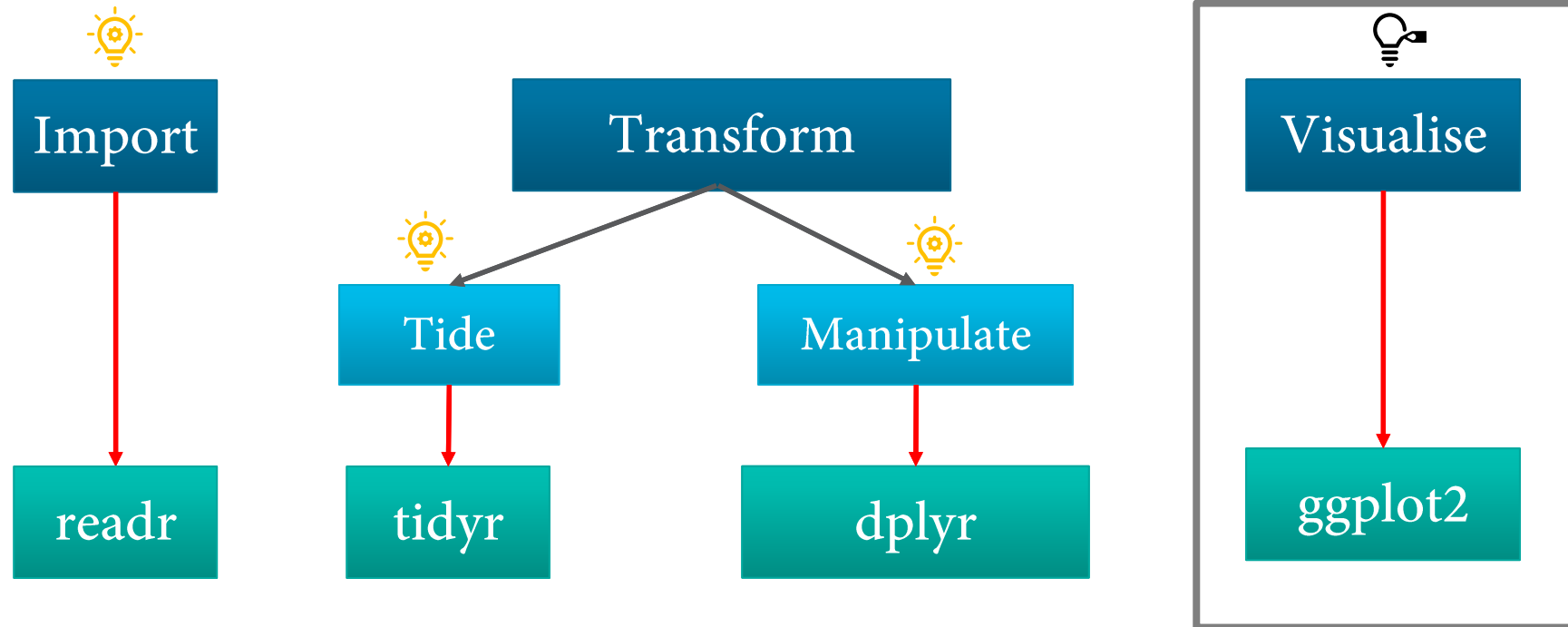
# OUTLINE

---

- Data Wrangling, a recap
- Exploratory Data Analysis
- Data Visualisation
- Grammar of Graphics
- EDA; Univariate Analysis

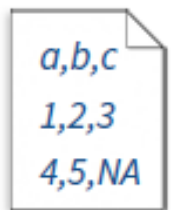
# DATA WRANGLING

- Practically, we have three main processes to wrangle the data



# DATA IMPORT (RECAP)

```
read_(file, col_names = TRUE, col_types = NULL, locale = default_locale(), na = c("", "NA"),  
      quoted_na = TRUE, comment = "", trim_ws = TRUE, skip = 0, n_max = Inf, guess_max = min(1000,  
      n_max), progress = interactive())
```

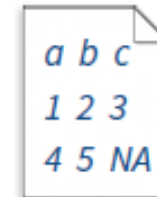


a,b,c  
1,2,3  
4,5,NA



A	B	C
1	2	3
4	5	NA

**Comma Delimited Files**  
`read_csv("file.csv")`

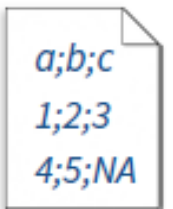


a b c  
1 2 3  
4 5 NA



A	B	C
1	2	3
4	5	NA

**Tab Delimited Files**  
`read_tsv("file.tsv")`  
Also `read_table()`

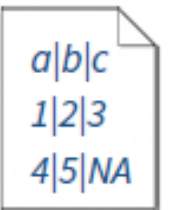


a;b;c  
1;2;3  
4;5;NA



A	B	C
1	2	3
4	5	NA

**Semi-colon Delimited Files**  
`read_csv2("file2.csv")`



a|b|c  
1|2|3  
4|5|NA



A	B	C
1	2	3
4	5	NA

**Files with Any Delimiter**  
`read_delim("file.txt", delim = "|")`

**To save data into csv or txt file**

**Comma delimited file**

`write_csv(x, path, na = "NA", append = FALSE,  
col_names = !append)`

**File with arbitrary delimiter**

`write_delim(x, path, delim = " ", na = "NA",  
append = FALSE, col_names = !append)`

# DATA MANIPULATION (RECAP)

- The `dplyr` package in `tidyverse` library presents five *verbs* for manipulating the data in data frames:
  1. `filter()` extracts a subset of the rows (i.e., observations) based on some criteria
  2. `select()` extracts a subset of the columns (i.e., features, variables) based on some criteria
  3. `mutate()` adds or modifies existing columns
  4. `arrange()` sorts the rows
  5. `summarise()` aggregates the data across rows (e.g., group them according to some criteria)
- Each of these functions takes a data frame as its first argument and returns a data frame.

# TIDY DATA (RECAP)

- There are three interrelated rules which make a dataset tidy:
  1. Each variable must have its own column.
  2. Each observation must have its own row.
  3. Each value must have its own cell.

Having your data in a tidy format is crucial for data manipulation and exploring

country	year	cases	population
Afghanistan	1999	18145	19807071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	17206362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	216766	128042583

variables

country	year	cases	population
Afghanistan	1999	18145	19807071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	17206362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	216766	128042583

observations

country	year	cases	population
Afghanistan	99	725	98071
Afghanistan	00	666	59360
Brazil	99	737	206362
Brazil	00	488	504898
China	99	2258	27291272
China	00	6766	42583

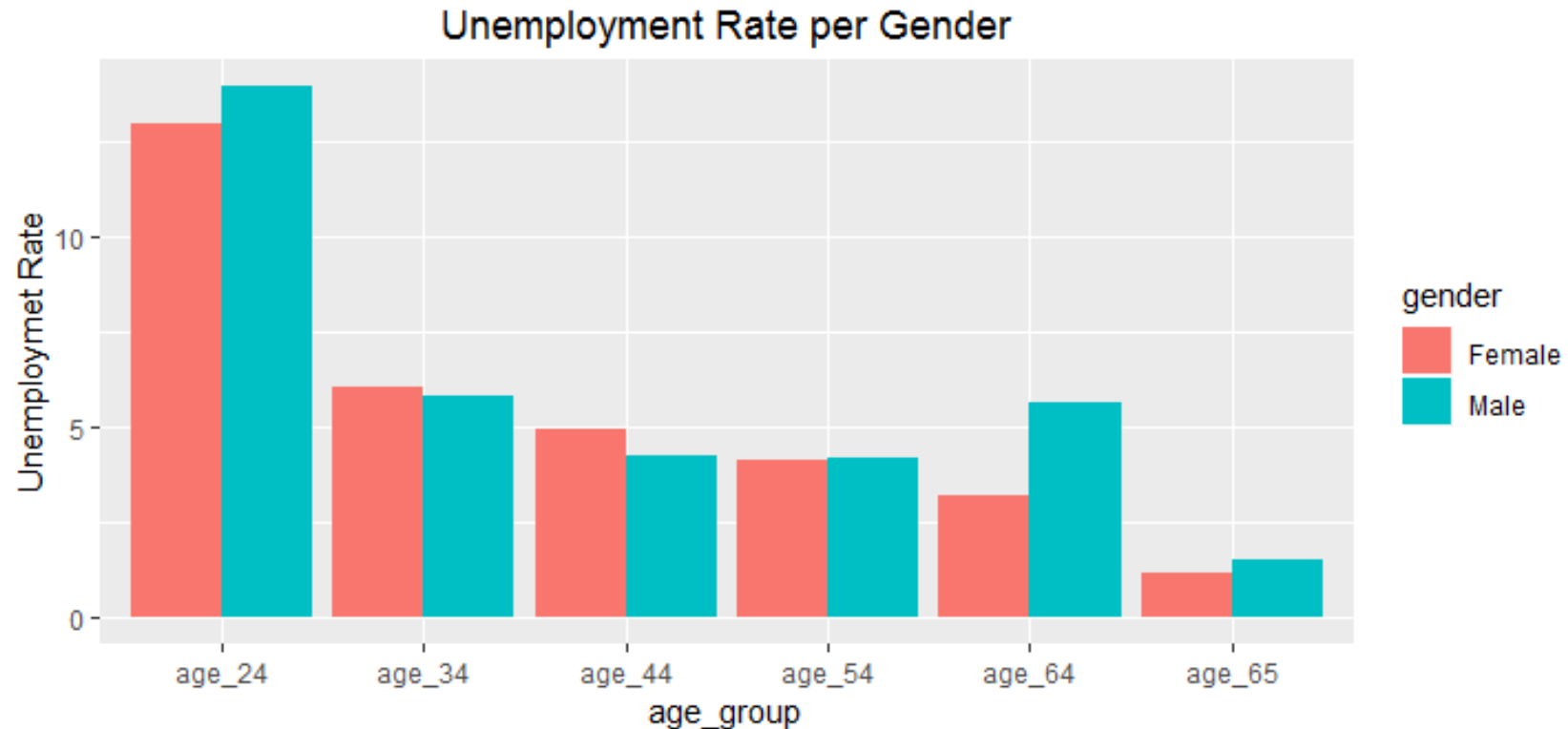
values

# TIDY DATA (RECAP) – 2

- The `tidyr` package presents four main *verbs/functions* to tidy up the data:
  1. `gather()` collapses multiple columns into key-value pairs. It produces a “long” data format from a “wide” one.
  2. `spread()` takes two columns (key & value), and spreads into multiple columns: it makes “long” data wider. This is the reverse of `gather`.
  3. `unite()` unites multiple columns into one
  4. `separate()` takes a column and divides it into multiple columns
- Each of these functions takes a data frame as its first argument and returns a data frame.

# DATA VISUALISATION

- *“The simple graph has brought more information to the data analyst’s mind than any other device.” — John Tukey*





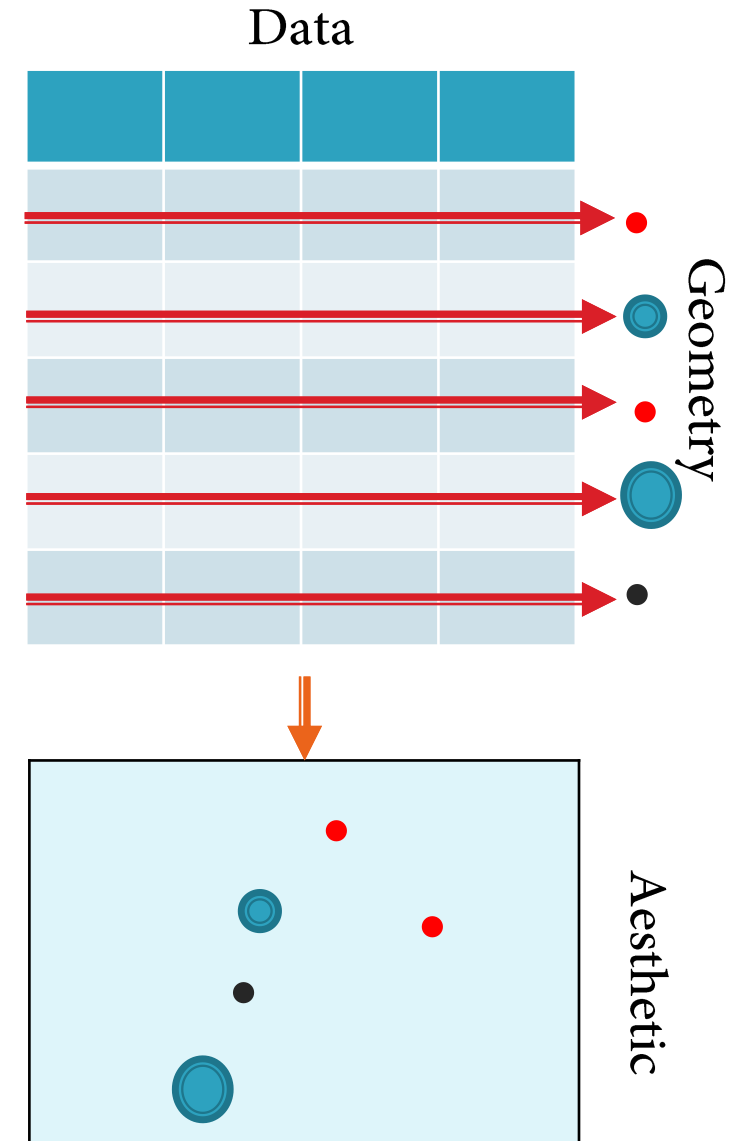
# DATA VISUALISATION (2)

---

- Presenting the data variables into a pictorial or a graphical format
- Visualising data provides a guide to:
  - Check changes in variables
  - Inspect the differences or relations between variables
  - Find patterns in the data
  - Grasp new concepts or insights from the data
- Data visualisation should be easy to the stakeholders
- Data visualisation aids data modelling processes

# GRAPH COMPONENTS

- To build a graph in R , you will need to specify three components:
  1. **Data:** the set of records/variables that we need to represent with a graph
  2. **Geometry:** the type of the plot, which will be generated, usually it is a function such as (scatterplot, boxplot, barplot, histogram, smooth density, etc.)
  3. **Aesthetic mapping:** the coordinate map and the other visual cues, such as size, scale and color.



# BUILD GRAPHS IN R

- There are two functions in ggplot2 library to build a graph in R:

`qplot()`

Quick plot that encapsulates the three graph components all in the call of the function.

`ggplot()`

Build the graph layer by layer. Start by defining an empty frame and then add the subsequent operations.

We will be focusing on using this method to build the graphs

# BUILD GRAPHS IN R (2)

- *Graph grammar* is an elegant way to use few functions to be able to build many graphs and plots layer by layer by combining these functions together.
- The template for building a ggplot graph:

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping=aes(<MAPPINGS>))
```

- The ggplot2 is one of the core members of the *'tidyverse'* library, so you will be able to use its functions when loading the *'tidyverse'* library.
  - *library(tidyverse)*

# GRAPHS IN R - EXAMPLE

- Let us start by looking at this dataset (mpg), which provides information about “Fuel economy data from 1999 to 2008 for 38 popular models of car”
- Then, we need to build a plot to answer some questions such as:
  - Do cars with big engines use more fuel than the car with small engines?
- To answer this question, let us have a look to the variables of this dataset:

```
> mpg
# A tibble: 234 x 11
  manufacturer model   displ year   cyl trans      drv   cty   hwy fl   class
  <chr>         <chr> <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
1 audi         a4      1.8  1999     4 auto(l5) f       18     29 p     compa~
2 audi         a4      1.8  1999     4 manual(m~ f       21     29 p     compa~
3 audi         a4      2    2008     4 manual(m~ f       20     31 p     compa~
4 audi         a4      2    2008     4 auto(av) f       21     30 p     compa~
5 audi         a4      2.8  1999     6 auto(l5) f       16     26 p     compa~
6 audi         a4      2.8  1999     6 manual(m~ f       18     26 p     compa~
7 audi         a4      3.1  2008     6 auto(av) f       18     27 p     compa~
8 audi         a4 quatt~ 1.8  1999     4 manual(m~ 4       18     26 p     compa~
9 audi         a4 quatt~ 1.8  1999     4 auto(l5) 4       16     25 p     compa~
10 audi        a4 quatt~ 2    2008     4 manual(m~ 4       20     28 p     compa~
# ... with 224 more rows
```

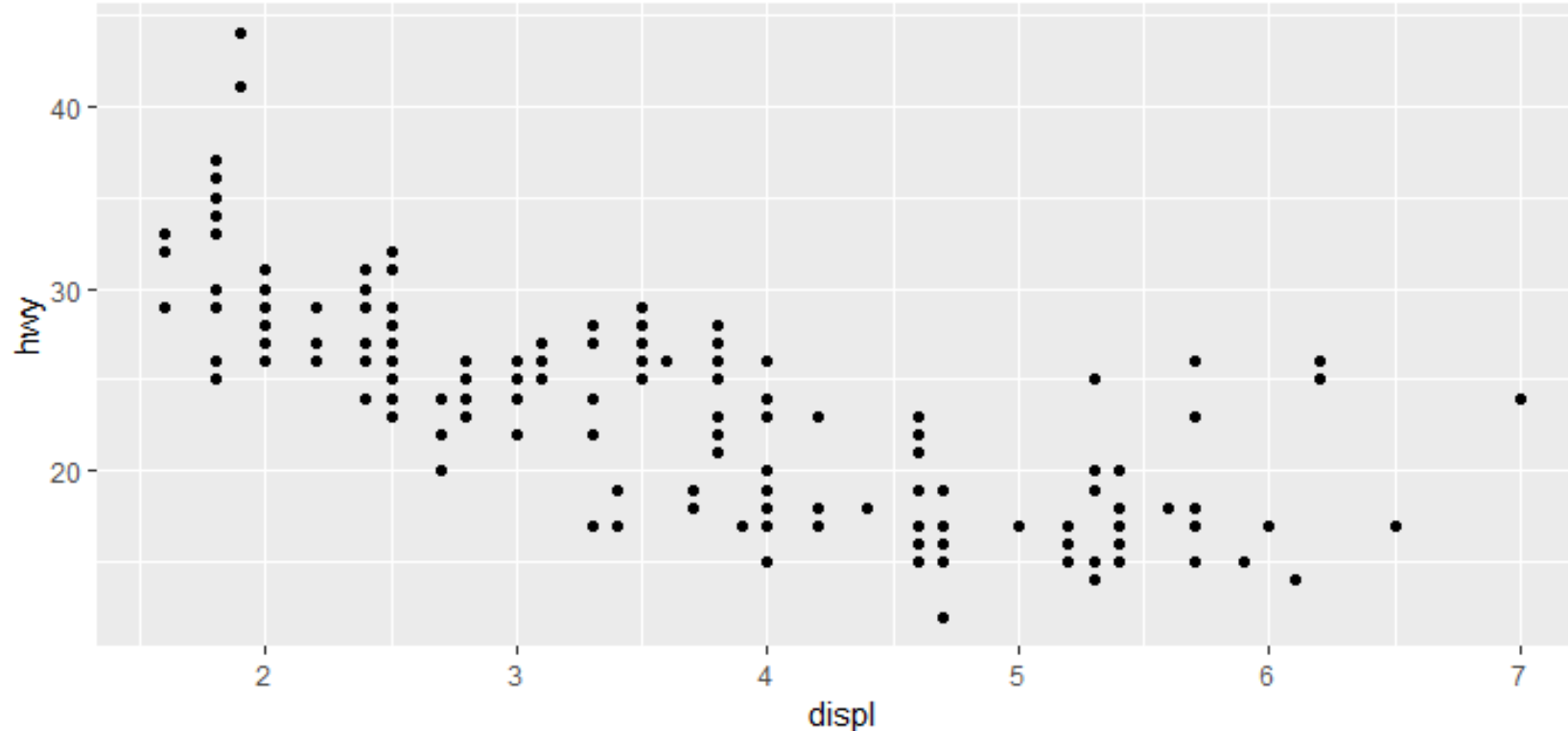
# GRAPHS IN R – EXAMPLE (2)

Create an empty  
coordinate system

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

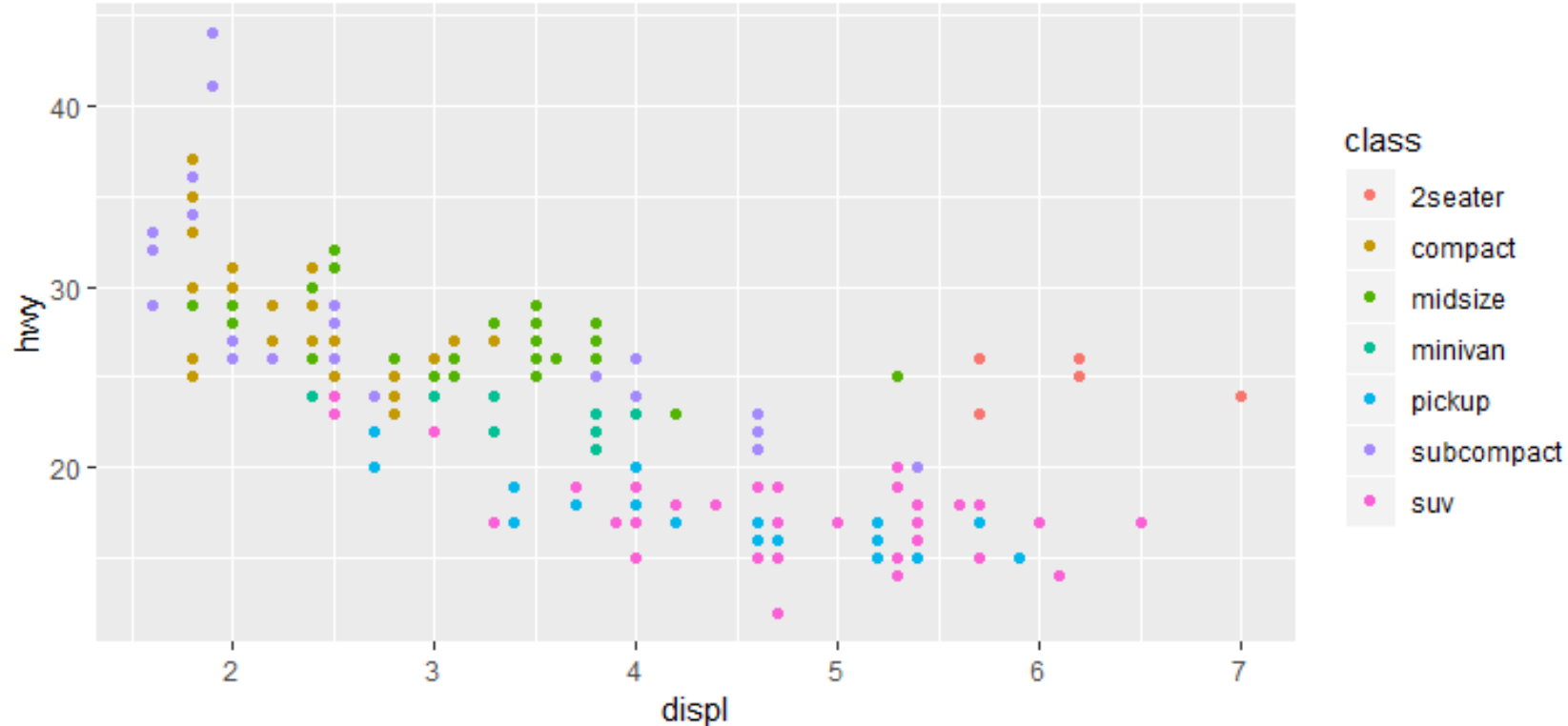
Add a layer of scatter  
plot to represent the  
relationship between  
two variables

You may add as many  
layers as needed



# GRAPHS IN R – EXAMPLE (3)

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

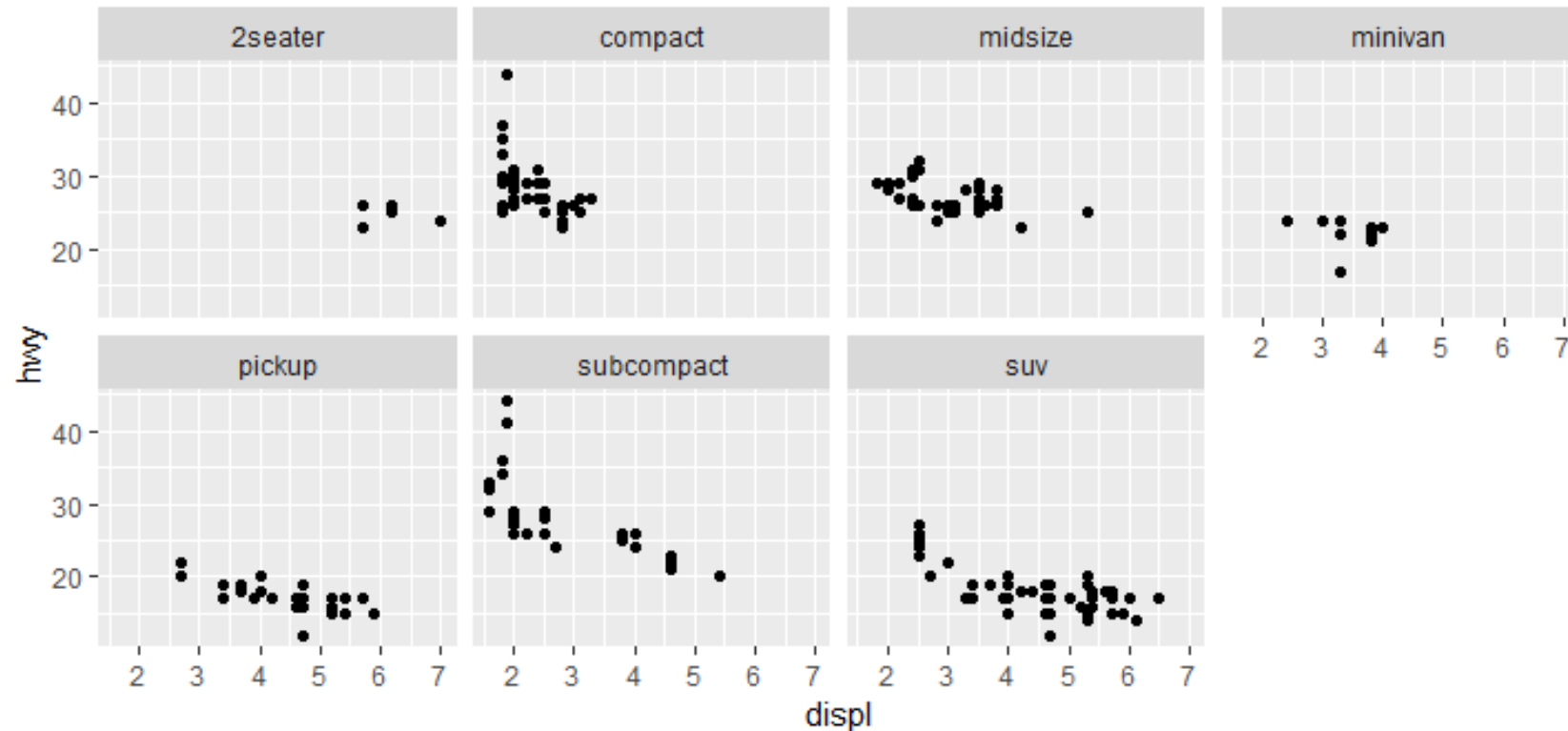


Three variables are represented in the 2D graph

# GRAPHS IN R – EXAMPLE (4)

You may add additional variables by using facets (e.g. subplots)

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_wrap(~class, nrow = 2)
```

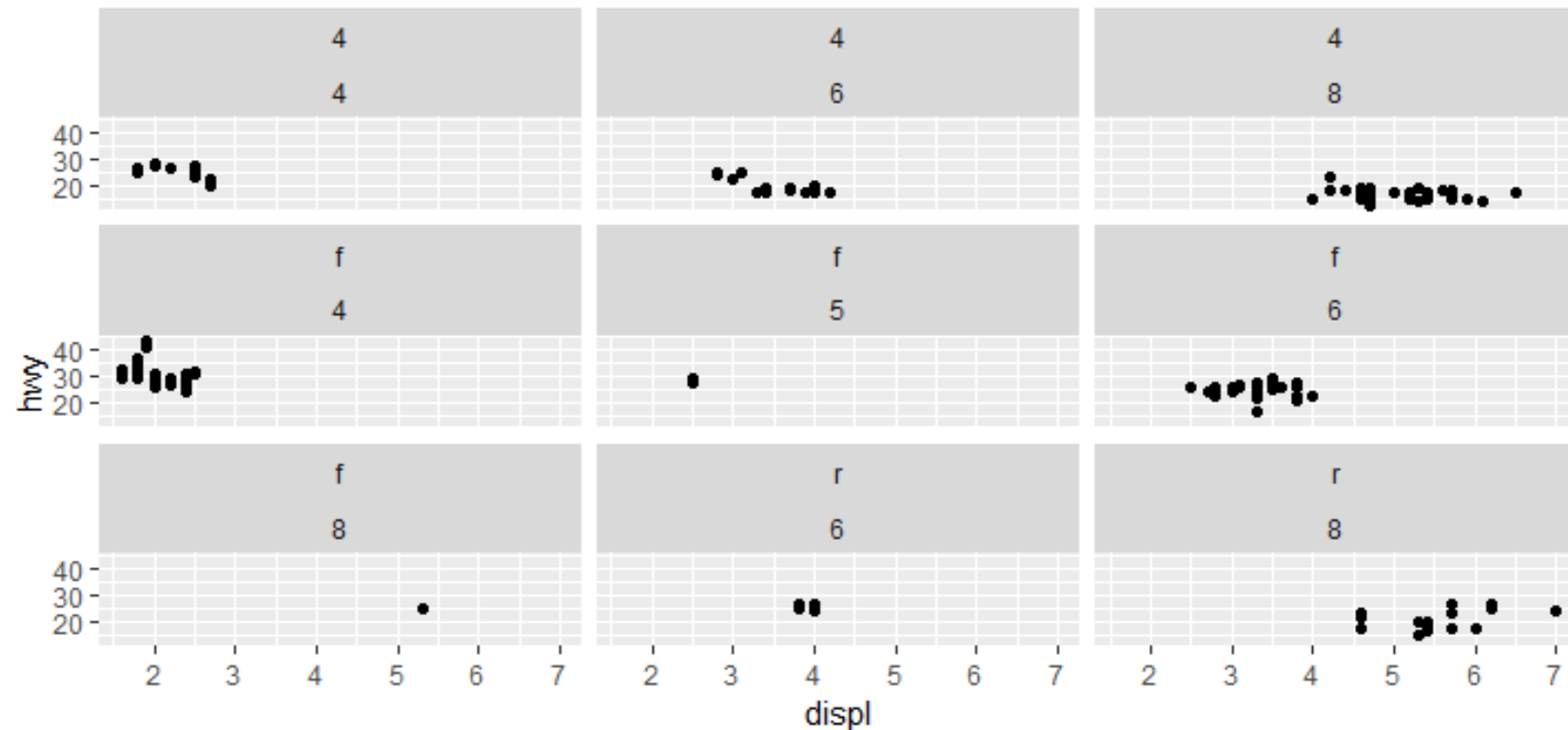




# GRAPHS IN R – EXAMPLE (5)

The variables () should  
be discrete when using  
facets

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_grid(drv ~ cyl)
```



# EXPLORATORY DATA ANALYSIS (EDA)

---

- EDA: *is the process of exploring the data variables toward discovering some trends or patterns from the data. This leads the modelling step toward fixing issues or guiding the decision making.*
- To understand the variables in a dataset, we may transform these variables into other format or extract their summaries (e.g. mean, variance, etc.) or to get insights about the distribution of these variables.
- The most elegant way to understand relationships with-in a variable or between variables is by *visualising* these relationships.

- To extract the relationships between variables or to discover the patterns/distributions of the variables, we need to check on the type of these variables.
- To conduct the data analysis on variables for sake of understanding their relationships, this analysis can be either:
  - Uni-variate analysis
    - Discover the variations of the data into **one** variable
  - Multi-variate analysis
    - Discover the co-variation of **multiple** variables
    - Bi-variate analysis is a special type of this analysis with only **two** variables.

# EXPLORATORY DATA ANALYSIS (EDA) – 3

- EDA
  - Uni-variate analysis
    - Discrete
    - Continuous
  - Bi-variate analysis (can be extended to multi-variate analysis)
    - Discrete
    - Continuous

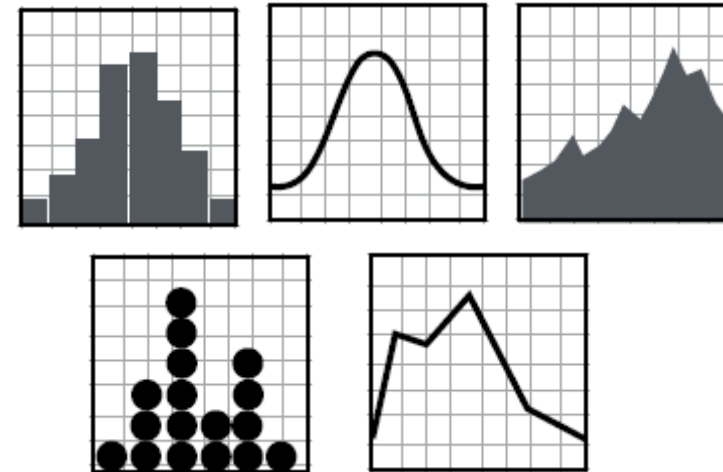
We will start by using the visualisation to do the EDA for both of the univariate and bi-variate analysis.

- There are two types of visualization-based univariate analysis:
  - Visualising variation of continuous variable
  - Visualising variation of discrete variable
- Examples of univariate continuous :
  - Histograms, etc.
- Examples of univariate discrete:
  - bar plots, etc.

# EDA – UNIVARIATE, CONTINUOUS

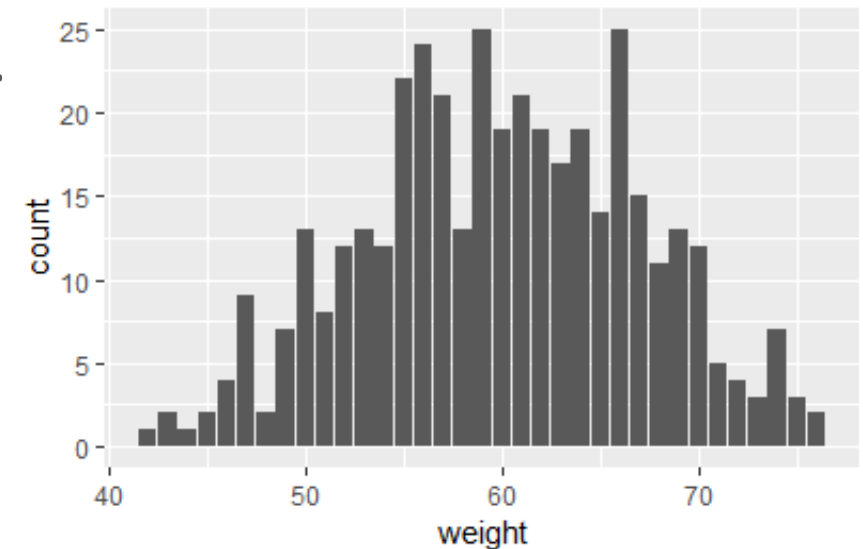
- The analysis is done based on just one variable, where it is a ‘numerical’ continuous variable.
- ggplot2 provides many functions to plot the variation of the univariate continuous variable such as:

- `geom_histogram()`, for a histogram plot
- `geom_density()`, for a density plot
- `geom_area()`, for an area plot
- `geom_dotplot()`, for a dot plot
- `geom_freqpoly()`, for a frequency polygon



# HISTOGRAMS

- Summarise the variable by extracting a 1-D distribution of the variation in the data within this variable.
- The data are summarised using suitable binwidths.
- A histogram plot shows properties, such as:
  - center (i.e., the location) of the data;
  - spread (i.e., the scale) of the data;
  - skewness (i.e., left or right) of the data and
  - presence of outliers



# HISTOGRAMS (2)

---

- A Histogram plot of a variable can answer some questions on the data such as:
  - What kind of distribution do the data come from?
  - How spread out are the data?
  - Are the data symmetric or skewed?
  - Are there outliers in the data?



# BUILDING A HISTOGRAM

1- a) First layer contains the data specification  
b) Also specifies the variable

2- a) Second layer specifies the type of plot;  
the geometric function

3- a) Third and following layers specify axis  
labels and further visual cues

```
ggplot(data,aes(x = variable_name))+
```

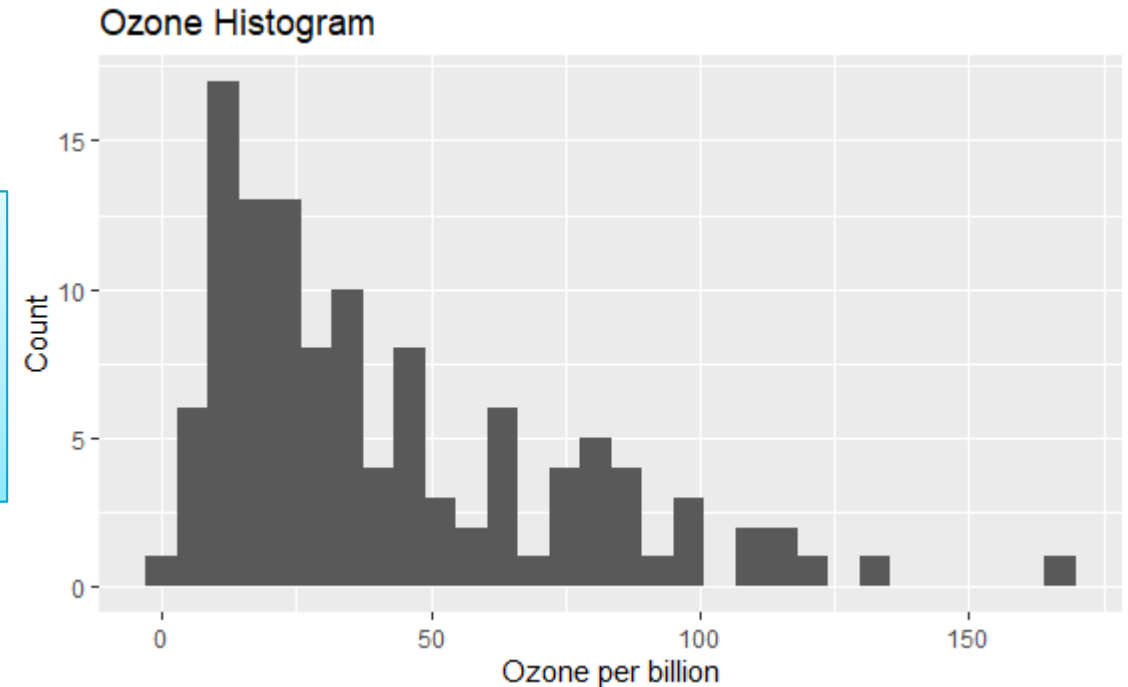
```
geom_histogram() +
```

```
xlab("x axis label") +  
ylab("y axis label")+  
ggtitle("plot title")
```

# EXAMPLE

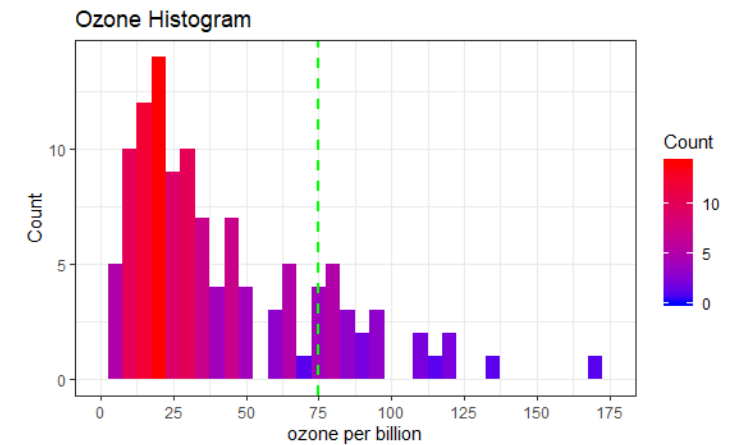
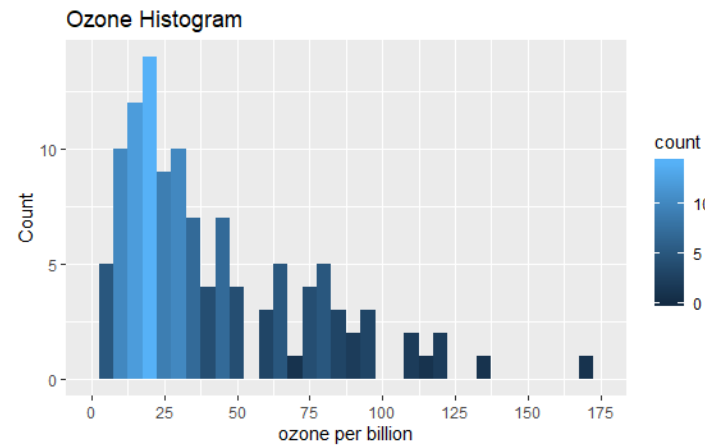
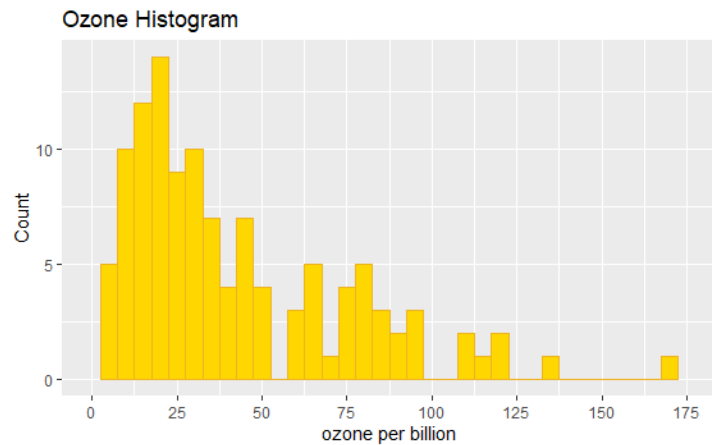
## Basic histogram visualization

```
ggplot(data= airquality, aes(x = Ozone)) +  
  geom_histogram() +  
  xlab("Ozone per billion") +  
  ylab("Count") +  
  ggtitle("Ozone Histogram")
```



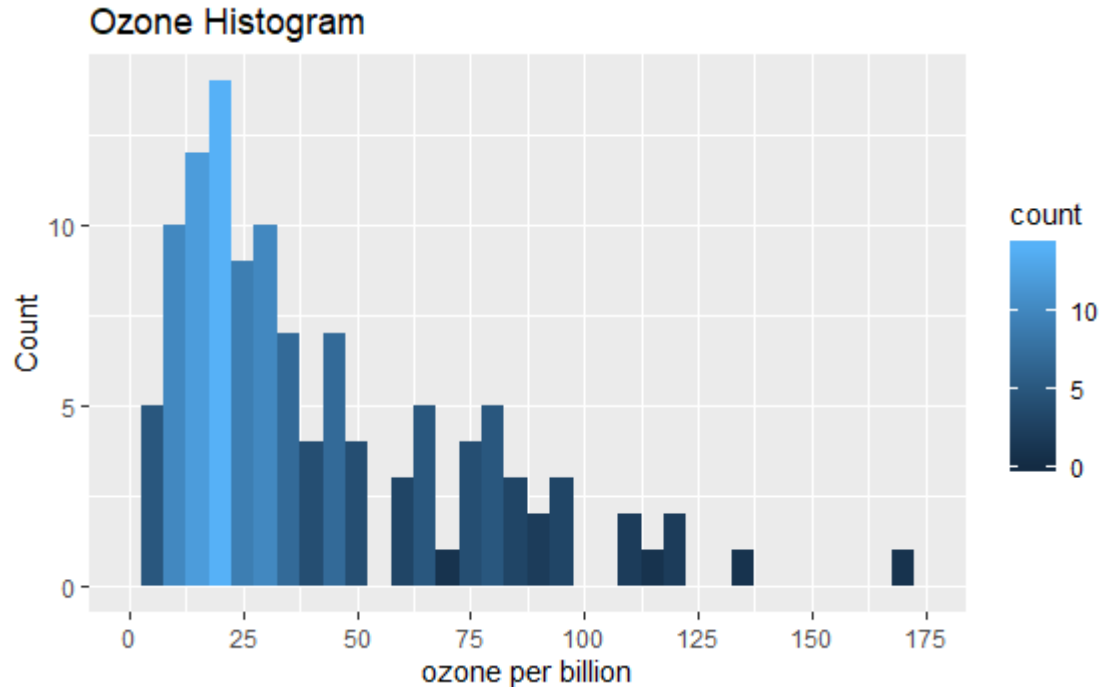
# EXAMPLE (2)

Adding extra visual cues and more options



The full code is shared under week 10 on Canvas

# HISTOGRAM INTERPRETATION



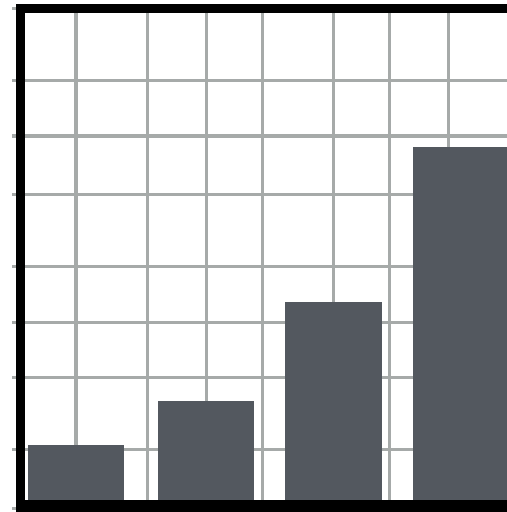
## Back to our questions...

- What kind of distribution do the data come from?
- How spread out are the data?
- Are the data symmetric or skewed?
- Are there outliers in the data?

The data is right skewed and there are clearly outliers as the graph stretches on the scale to 175 Ozone per billion. It seems the median is around the 20 Ozone per billion.

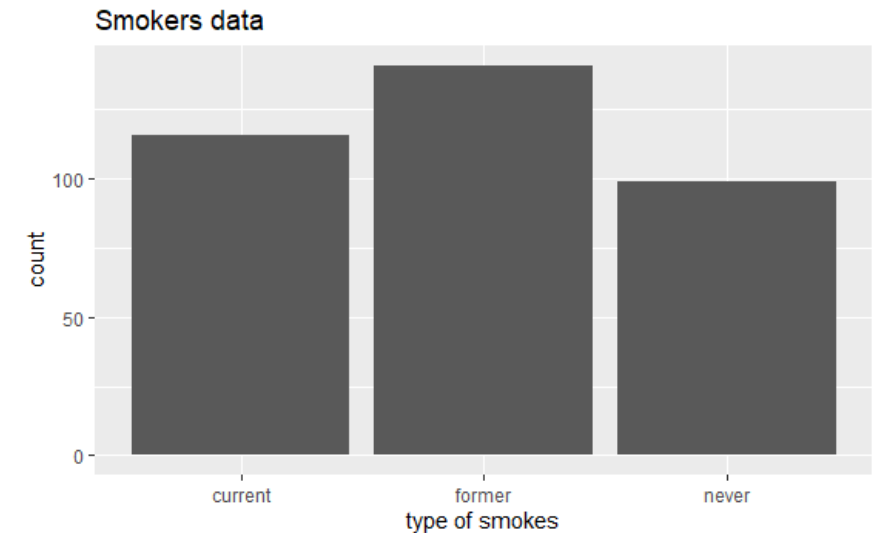
# EDA – UNIVARIATE, DISCRETE

- The analysis is done based on just one variable, where it is a discrete (i.e. categorical) variable.
- ggplot2 provides one functions to plot the variation of the univariate discrete, which is:
  - *geom\_bar()*, for a bar plot



# UNIVARIATE BAR PLOT

- Summarise the variable by extracting a frequency of each level/category of the univariate variable.
- The x-axis represents the levels/categories of the data
- The y-axis becomes the counts of each category.
- Bar plots helps answering the following questions:
  - What is the spread of observations between data levels?
  - Which level has the most observations or least observations etc.



# BUILDING A BAR PLOT

1- a) First layer contains the data specification  
b) Also specifies the variable

2- a) Second layer specifies the type of plot;  
the geometric function

3- a) Third and following layers specify axis  
labels and further visual cues

```
ggplot(data,aes(x = variable_name))+
```

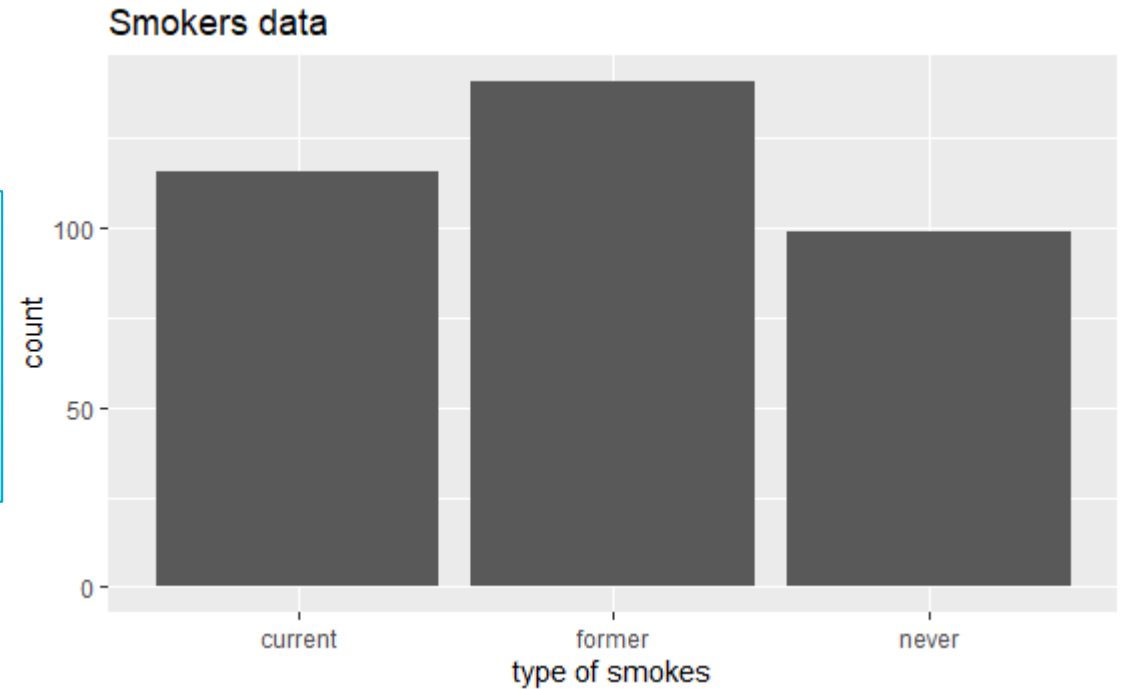
```
geom_bar() +
```

```
xlab("x axis label") +  
ylab("y axis label")+  
ggtitle("plot title")
```

# EXAMPLE

## Basic bar plot visualization

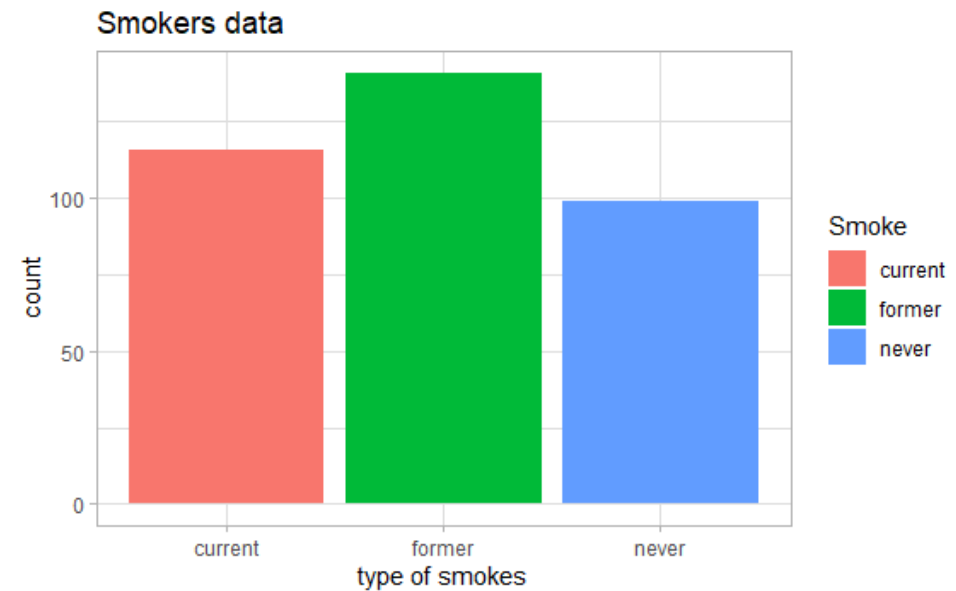
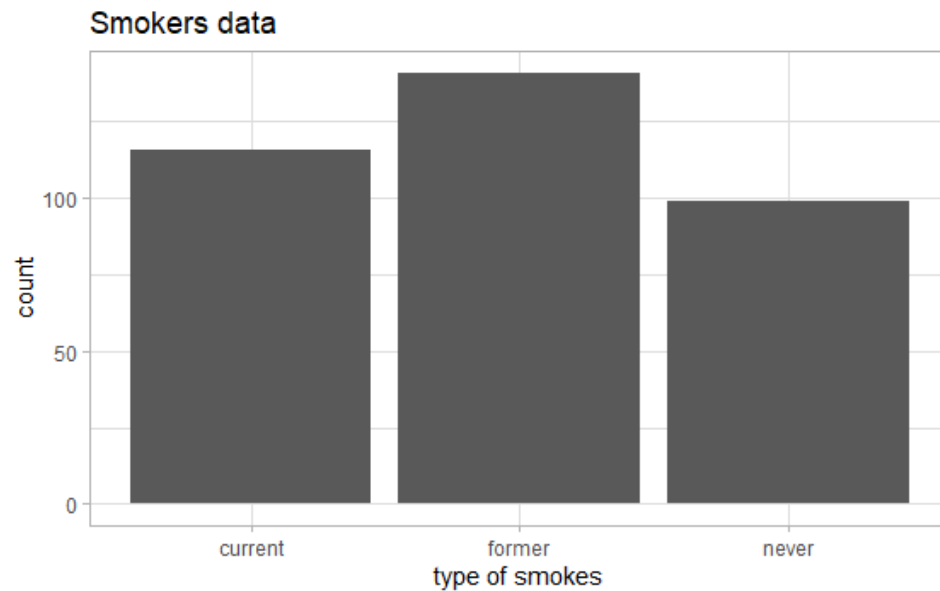
```
ggplot(data, aes(x = Smoke)) +  
  geom_bar()+  
  xlab("type of smokes") +  
  ylab("count") +  
  ggtitle("Smokers data")
```





# EXAMPLE (2)

Adding extra visual cues and more options



The full code is shared under week 10 on Canvas

# BAR PLOT INTERPRETATION



## Back to our questions...

What is the spread of observations between levels of the data?  
Which level has the most observations or least observations  
etc. ?

Former smokers are more than the current, etc.

# RECOMMENDED READING

---

- You are recommended to read chapters 3 from the “*R for Data Science*” book:
  - <https://r4ds.had.co.nz/data-visualisation.html>

# ANNOUNCEMENT

---

- This week online test has been released since yesterday and is due this Sunday.
  - One attempt
  - 60 minutes long
  - Weights 20%