Unit 2: Exploratory Data Analysis

Exploratory Data Analysis (EDA)

- EDA is the first key step in analyzing the data and can be considered an initial investigation of a data set.

- EDA can entail:
  1. visualizing data and discovering patterns and structures,
  2. identifying abnormalities and outliers,
  3. identifying important variables / optimal factor settings,
  4. checking assumptions,

EDA for different types of data

The tools used in EDA depend on the type of the data.

- Categorical (Qualitative) data
  - Graphical summaries: Pie and bar charts
  - Numerical summaries: Counts, frequency tables, cross tabulations
- Numerical (Quantitative) data
  - Graphical summaries: Histogram, boxplot, comparative plots
  - Numerical summaries: Mean, median, mode, standard deviation, quartiles and others
- Bivariate numerical data
  - Graphical summaries: Scatterplot
  - Numerical summaries: Correlation
- Time series data
  - Graphical summaries: Time series plot
  - Numerical summaries: Auto-correlation
- High-dimensional and complex data

When you summarize data...

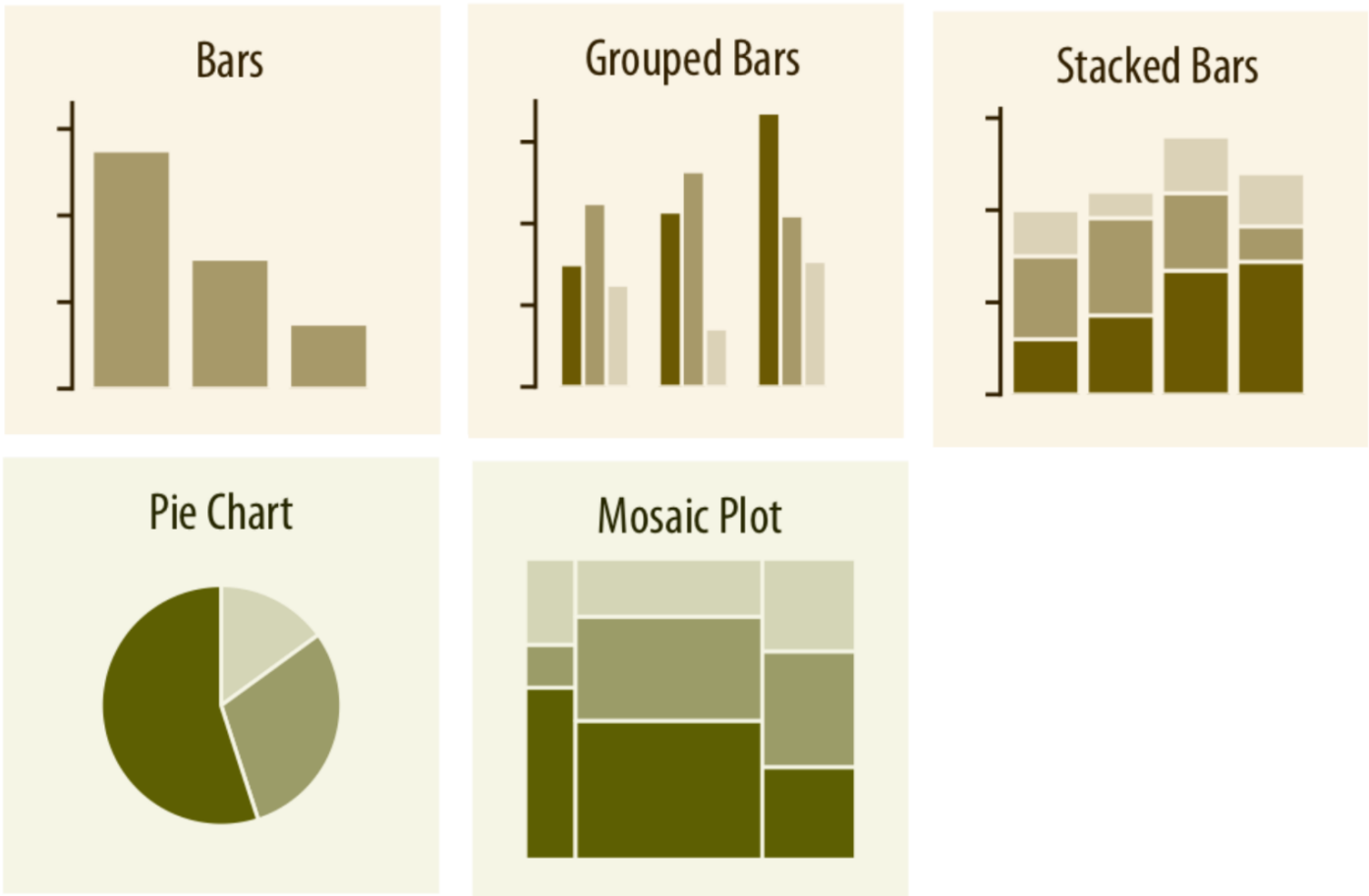
- Data is typically comprised of observations on variables; the two types of variables are quantitative (numerical) and qualitative (categorical).
- Depending on the type, use appropriate graphical summaries.
- Depending on the type, use appropriate numerical summaries.
- Use graphical summaries to decide which measures to use for central tendency and dispersion.
- Finally, it is important to describe what these measures tell us about the data.

Categorical data

- Categorical data represent values of categorical variables (also called *qualitative variables*) that place individuals onto one of several groups.
- This type of data is recorded as labels.
- Consider a data set with observations about individuals; categorical data could consist of hair colour, eye colour and blood type.
- Recall that categorical variables can be nominal and ordinal, where, in the latter, ordering makes sense (has some meaning).

Graphical displays for categorical data

- Graphical displays of categorical data focus on counts and proportions, such as:



Graphical displays for categorical data: pie charts

- In a pie chart, slices of the pie represent categories and the size of the slices reflect the proportions.

- Consider the distribution of colours in bags of M&M's

An example

- It is said that the distribution of colours is: brown (14%), yellow (14%), red (13%), blue (24%), green (15%) and orange (20%) We can use R's function pie to make a pie chart of the colours:

```
> pie(myMMs$value, labels = myMMs$group,col=c("tan4","yellow","r
ed","blue","springgreen4","orange"))
```

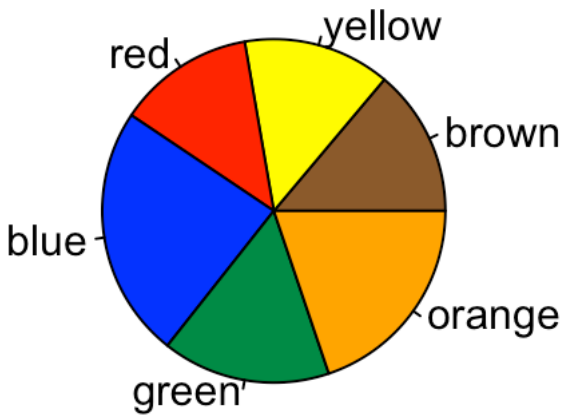
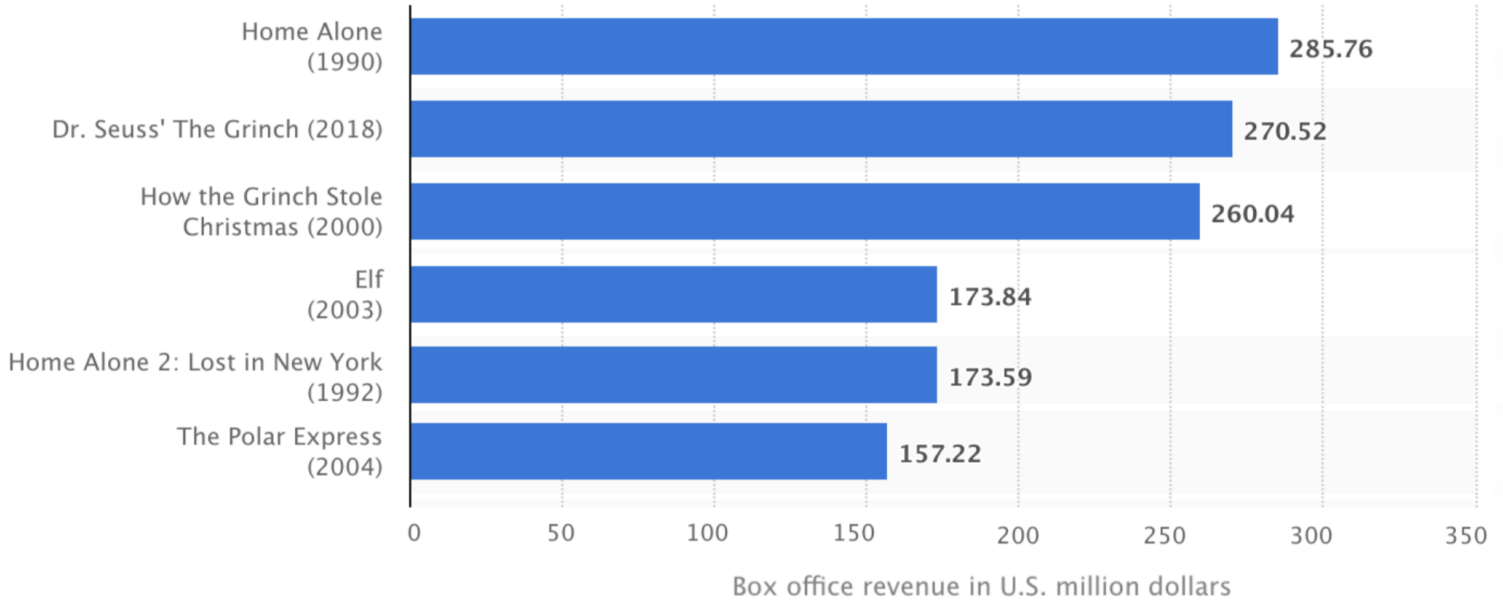


*Image and proportions taken from `https://joshmadison.com/2007/12/02/mms-color-distribution-analysis/`.

Graphical displays for categorical data: bar charts

- In a bar chart, the bars represent categories and the size of the bars reflect the counts (frequencies) or proportions.
- Here we see the US box office revenue of Christmas movies (as of February 2019) in the form of a bar chart (bar graph):



*Taken from statista.

- To make a simple bar chart in R, see the help file on the function barplot.

Cross tabulation for categorical data

- Cross tabulation, also known as contingency tables, are a way of displaying counts/frequencies for two variables across *several* categories.
- For a data set containing information on n individuals/units, a simple example is shown below:

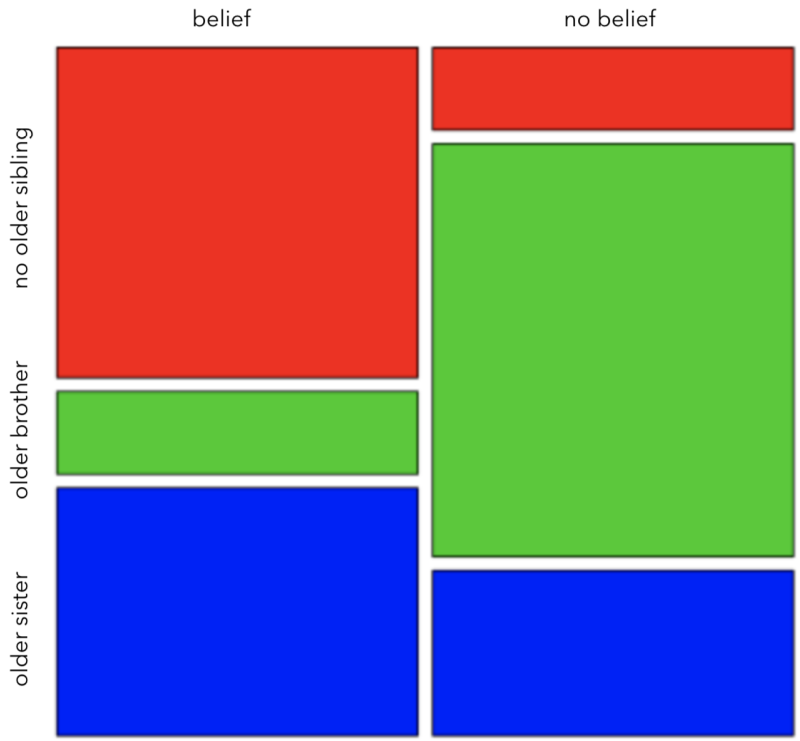
Variable 1 / Variable 2	Category 1	Category 2	Category 3	Total
Category 1	a	b	c	a+b+c
Category 2	d	e	f	d+e+f
Category 3	g	h	i	g+h+i
Total	a+d+g	b+e+h	c+f+i	a+b+c+d+e+f+g+h+i=n

Contingency tables

- More precisely, these tables are made up of what are called cells and each cell contains the count/frequency; that is, cell (i, j) contains the count for those falling in Category i for Variable 1 and in Category j for Variable 2.
- In making such tables, one would may wish to determine if there is independence between the variables;
- For this purpose, there is what is called the chi-squared test.
- These tests will be discussed in detail in a subsequent unit.

Graphical displays for categorical data: mosaic plot

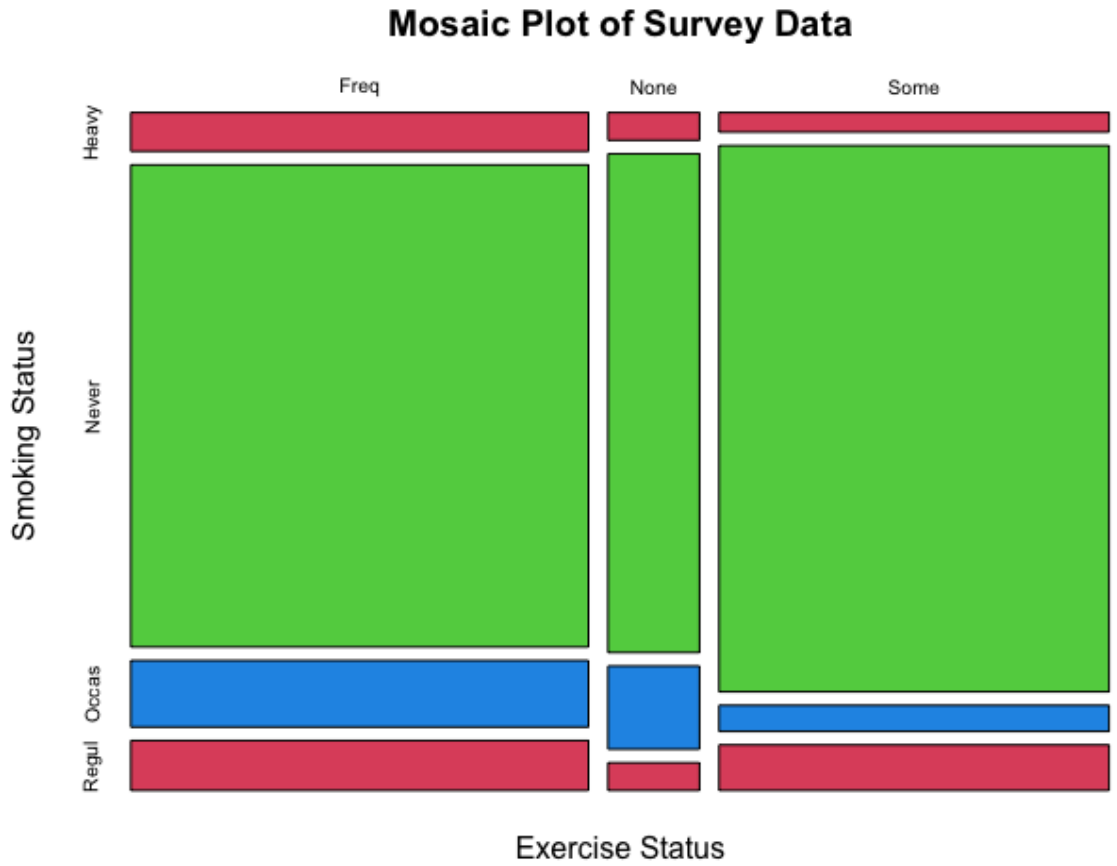
- A mosaic plot is a way of displaying proportions in a contingency table (cell relative frequencies).
- Here we see the proportions for two variables relating to belief in Santa: sibling status (3 categories) and belief (2 categories):



*Taken from `https://www.cyclismo.org/tutorial/R/intermediatePlotting.html`.

An example

- Consider the built-in data set survey in the package MASS; two of its variables are categorical: smoking status and exercise practice.
- We can create a mosaic plot in R using the function mosaicplot.

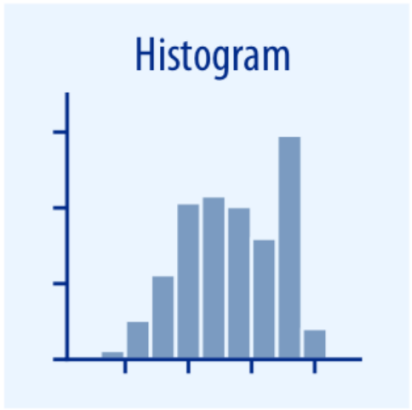


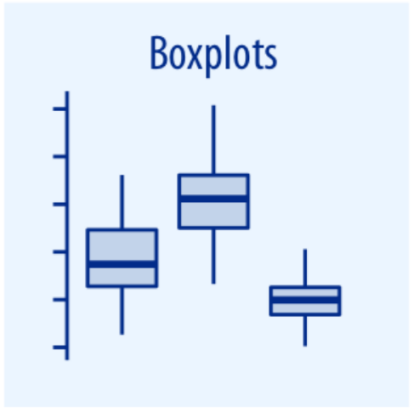
Quantitative data

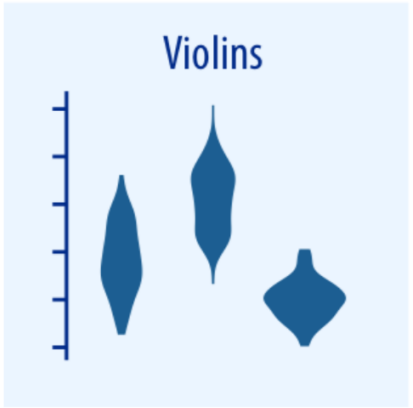
- Quantitative data represent values of quantitative variables for which arithmetic operations such as adding and averaging make sense.
- This type of data is recorded as numbers.
- Recall that quantitative variables can be continuous or discrete.
- Consider a data set with observations about individuals; quantitative data could consist of cholesterol level, weight and height.

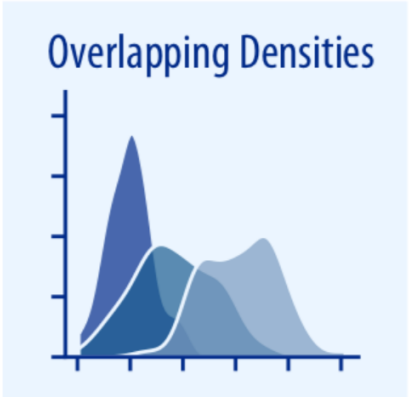
Graphical displays for numerical data

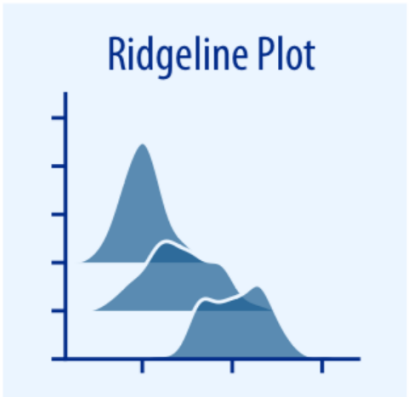
- Graphical displays of numerical data focus on the distribution of the data, such as:







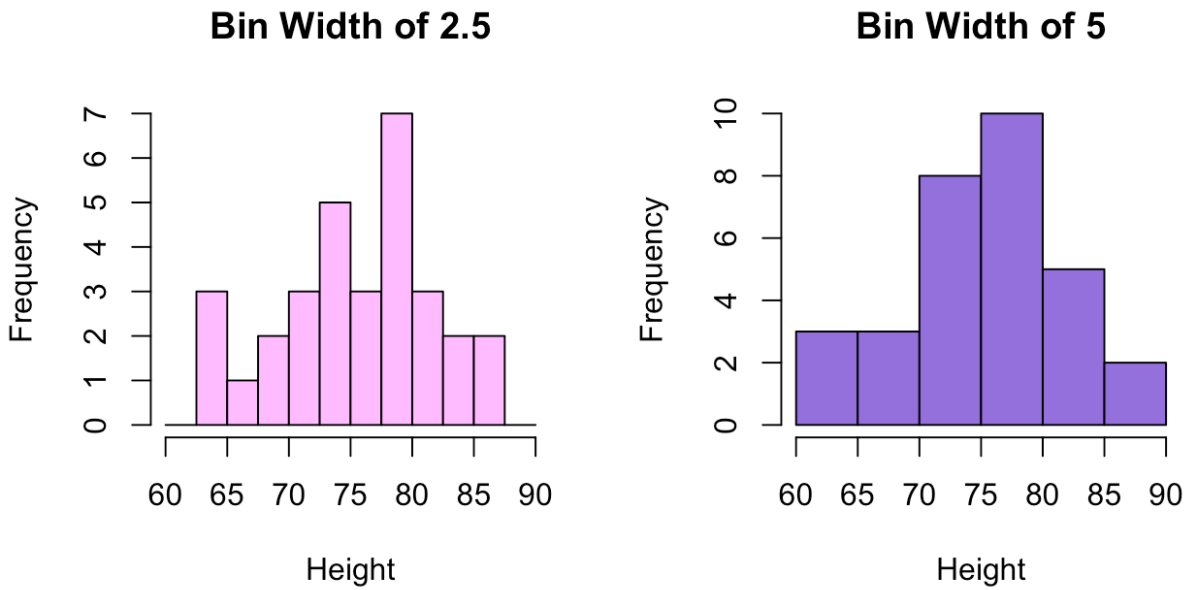




Graphical displays for numerical data: histograms

- Histograms are like bar charts except they are for quantitative data and the bars touch to reflect continuity in the data.
- Consider the in-built data set trees and making histograms of one of the variables (Height) using R's function hist:

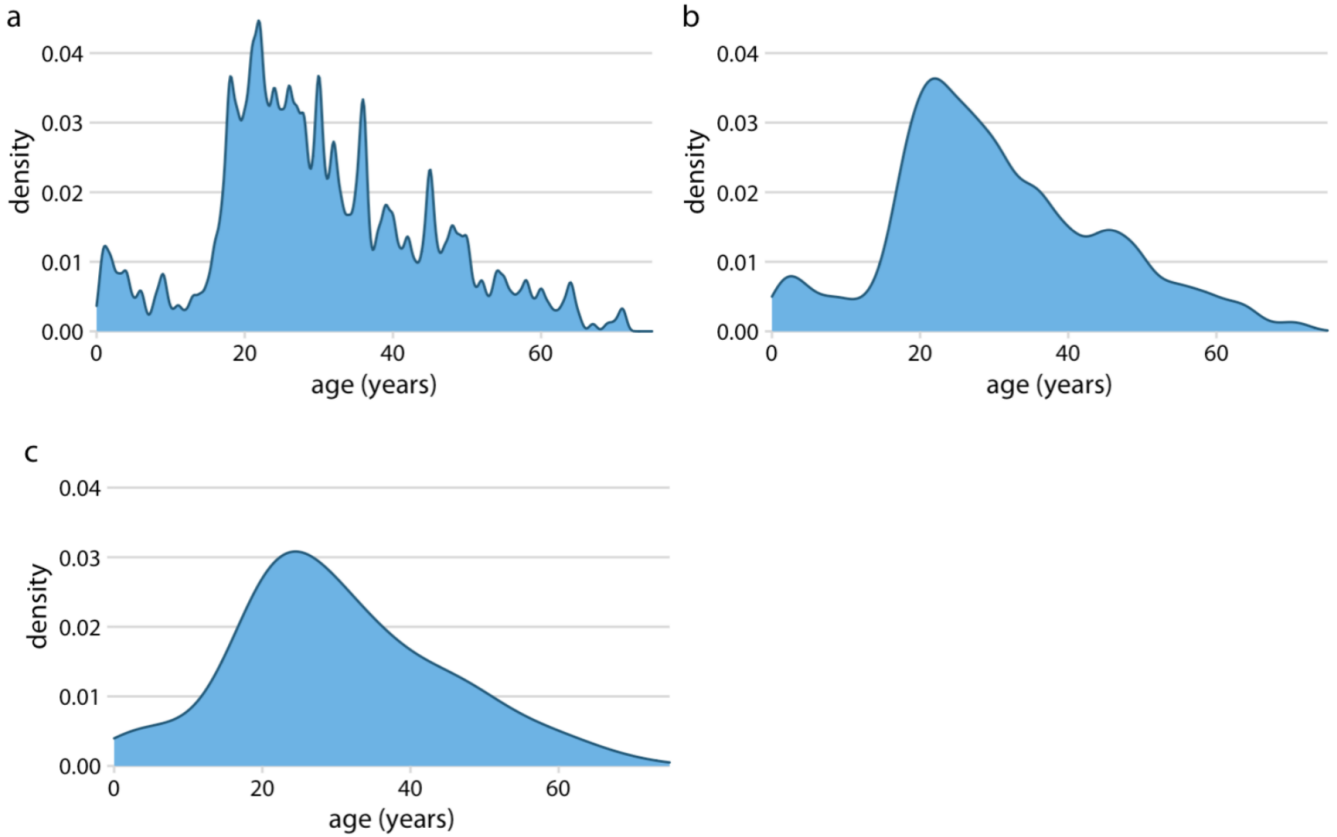
```
> hist(trees$Height,breaks=seq(60,90,2.5),col="plum1",main="Bin
 Width of 2.5",xlab="Height")
> hist(trees$Height,breaks=seq(60,90,5),col="mediumpurple",main
="Bin Width of 5",xlab="Height")
```



- Notice it's a different picture depending on how many bins are used.

Graphical displays for numerical data: density plots

- When dealing with data on a continuous variable, we can smooth histograms by creating a density curve (density plot), which in these cases, are estimates of the probability density function (PDF) of the variable:



- PDFs will be discussed in more detail in Unit 3.

Kernel density plots

- R's function density can create such densities, with the normal (gaussian) being the default PDF being fit.

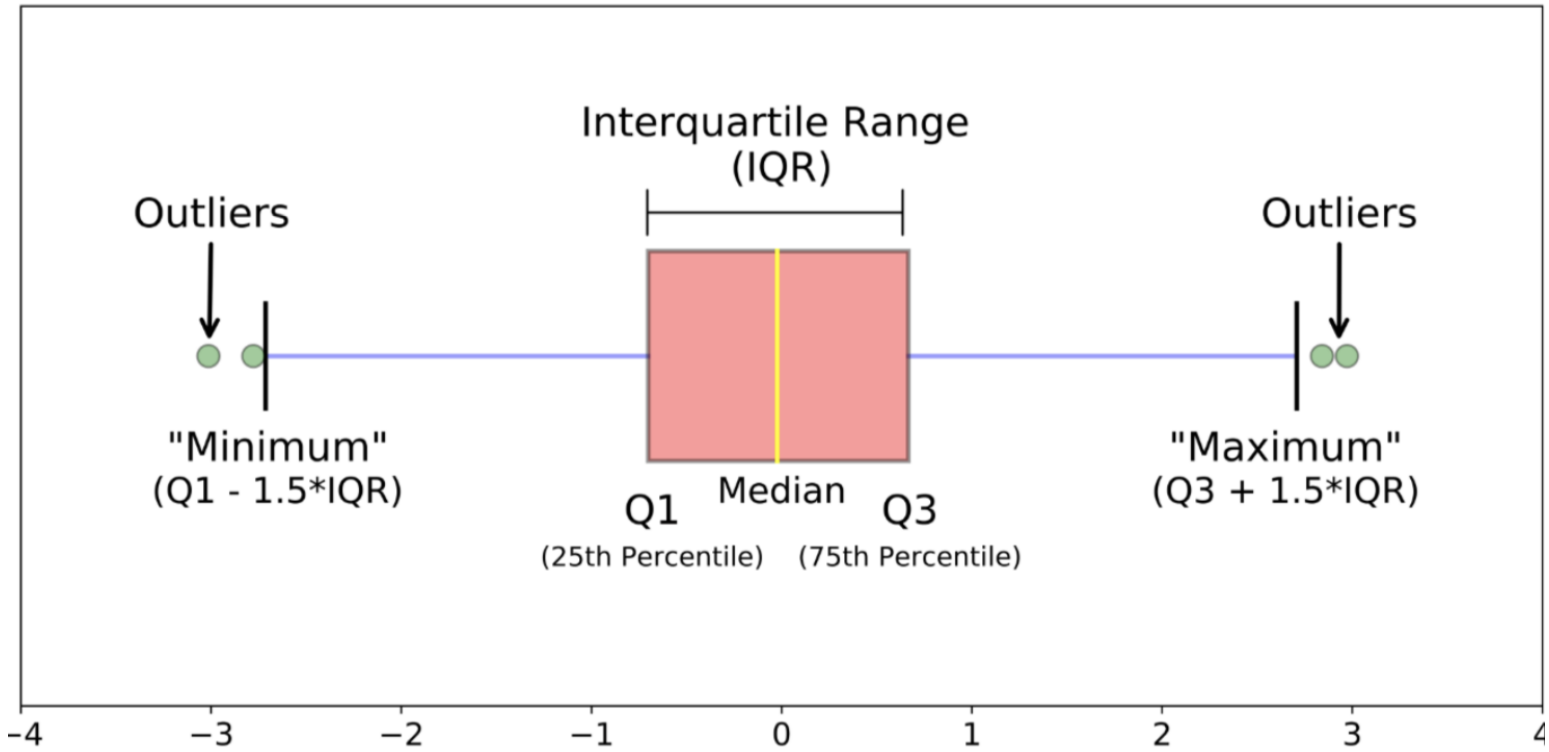
- Some details (that you **don't need to know**) about density plots:
  - The density function creates what is called a kernel density, which is defined as:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

    where K is called the kernel and h is called the bandwidth.
  - Kernels are special PDFs in that they have to be even functions, such as the Uniform(-1,1) PDF and the Normal(0,1) PDF; a kernel can be chosen through the argument kernel.
  - The bandwidth is like a *smoothing* parameter and it can be specified through the argument bw.

Graphical displays for numerical data: boxplots

- Another plot displaying the distribution of a data is the boxplot:

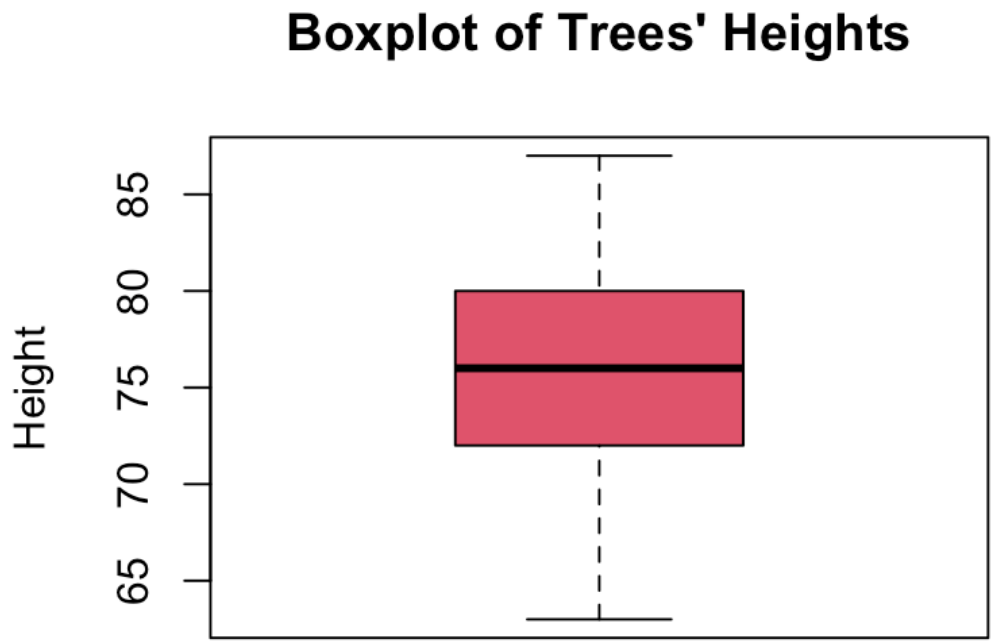


- Boxplots can be vertical or horizontal.
- From a boxplot, one can comment on such things as the center, spread and shape; also, with a *modified boxplot*, outliers can be shown.

An example

- Consider again the trees data set and their heights:

```
> boxplot(trees$Height,main="Boxplot of Trees' Heights",
col=2,ylab="Height")
```



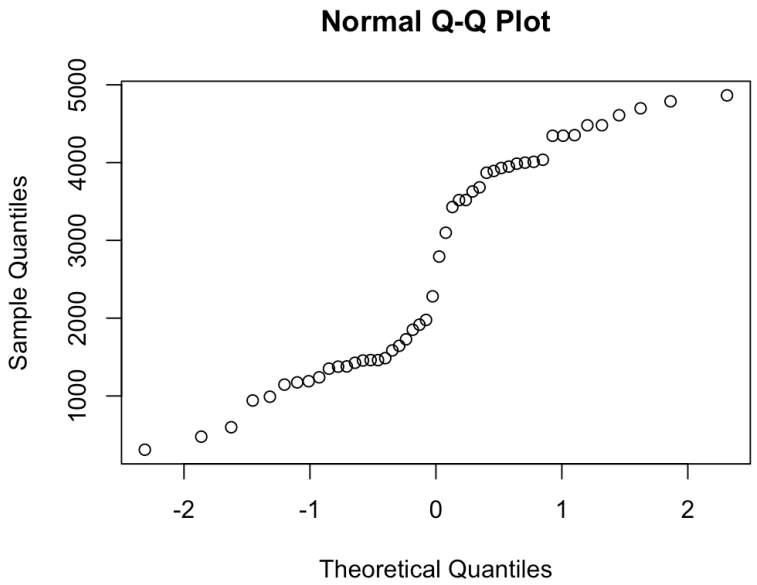
- From the boxplot, it appears the distribution is fairly symmetric.

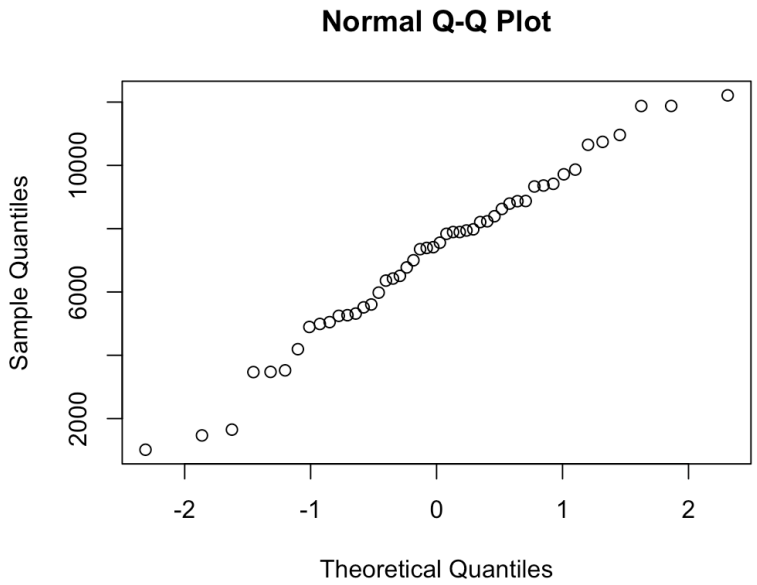
Assessing a model: the Q-Q plot (quantile-quantile plot)

- The Q-Q plot is a graph which allow us to assess if numerical data has come from a distribution, such as Normal.
- The plot allows us to see if the distributional assumption is plausible, and if not, how the assumption is violated and what data points contribute to the violation.
- A QQ-plot is simply a scatterplot (to be discussed soon) created by plotting two sets of quantiles against one another.
- If both sets of quantiles have come from the same distribution, we should see the points forming a straight line (approximately).
- In R, there are two functions to create Q-Q plots: qqnorm() and qqplot(); the difference between the two is that in using the latter, you can specify which distribution you are assessing goodness-of-fit.

The Q-Q plot: an example

- Let us consider R's data set rock.
- Suppose that we want to assess whether the variables perimeter and area follow Normal distributions.





- From the plot on the left, it seems that perimeter is not normally distributed but from the QQ plot on the right, it is evident that area is approximately normally distributed.

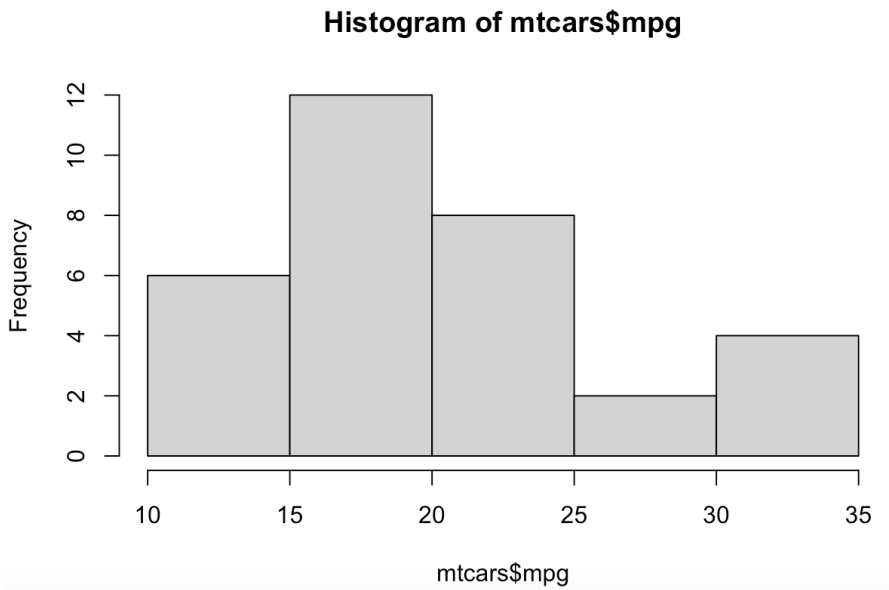
Review: some key summaries for numerical data

Summary	Type of Sample	Computing	Remarks
Sample mean, $\hat{\mu}, \bar{X}$	Continuous	$\frac{1}{n}\sum_{i=1}^{n} X_i$	• Summarizes the "center" of the data • Sensitive to outliers
Sample variance, $\widehat{\sigma^2}, S^2$	Continuous	$\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$	• Summarizes the "spread" of the data • Outliers may inflate this value
Order statistic, $X_{(i)}$	Continuous	ith largest value of the sample	• Summarizes the order/rank of the data
Sample median, $X_{0.5}$	Continuous	If n is even: $\frac{\left(X_{\left(\frac{n}{2}\right)} + X_{\left(\frac{n}{2}+1\right)}\right)}{2}$ If n is odd: $X_{\left(\frac{n}{2}+0.5\right)}$	• Summarizes the "center" of the data • Robust to outliers
Sample α quartiles, X_α $0 \le \alpha \le 1$	Continuous	If $\alpha = \frac{i}{n+1}$ for $i = 1, \ldots, n$: $X_\alpha = X_{(i)}$ Otherwise, do linear interpolation	• Summarizes the order/rank of the data • Robust to outliers
Sample Interquartile Range (Sample IQR)	Continuous	$X_{0.75} - X_{0.25}$	• Summarizes the "spread" of the data • Robust to outliers
Correlation	Bivariate	$r = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right)\left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}}$	• Measures the strength of linear relationship between X and Y

A simple example

- Looking at the mtcars data set, and in particular, exploring the variable mpg, we can use R to find the sample mean, variance and make a histogram:

```
> mean(mtcars$mpg)
[1] 20.09062
> var(mtcars$mpg)
[1] 36.3241
> hist(mtcars$mpg)
```



Some measures of shape

- For numerical data, many statistical tests rely on the assumption that the data come from a normal population, i.e., are normally distributed; this assumption would imply a nice bell shape to the data (when looking at, say, a histogram).

- Just like sample variance or IQR are used to summarize spread, we can also look at the shape of the data set through different numerical measures.

- To determine the shape of a data set, two common measures are skewness and kurtosis.

Shape statistics: skewness and kurtosis

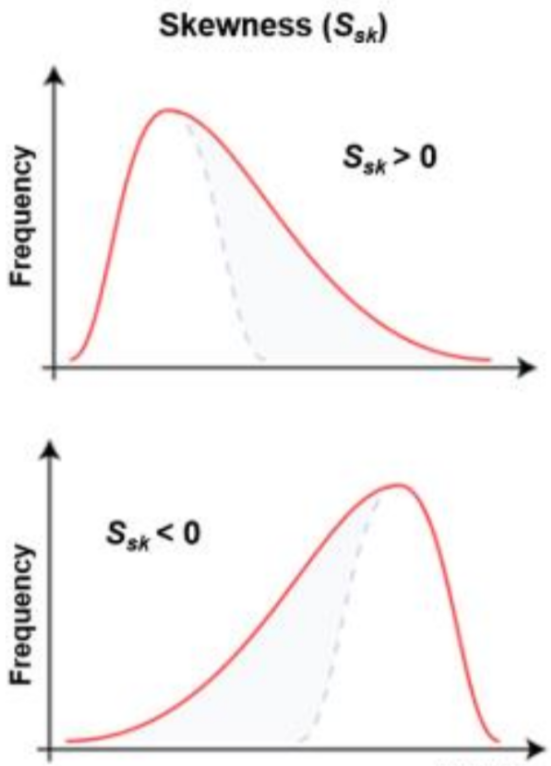
- Skewness is a measure of the *assymmetry* of the distribution, i.e., the lack of symmetry it possesses.

- A histogram can indicate skewness but numerically, for a sample of size n with observations $x_1, \ldots, x_n$, skewness can be quantified with the moment coefficient of skewness which is computed as

$$S_{sk} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left(\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}\right)^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2\right]^{3/2}}.$$

Skewness

- Any distribution that is symmetric has a skewness of 0, whereas a distribution that is skewed to the right is positively skewed and a distribution that is skewed to the left is negatively skewed.



Kurtosis

- Kurtosis is measure of how heavy-tailed the distribution is.
- A high kurtosis implies heavy-tailed (leptokurtic) and a low kurtosis implies light-tailed (platykurtic); the normal distribution has a kurtosis of 3 and is called mesokurtic.
- For a sample of size n, the sample excess kurtosis is computed as

$$\boxed{S_{ku} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^2} - 3}.$$

- The above definition (formula) of kurtosis is often used so that the excess kurtosis of the normal distribution is 0.

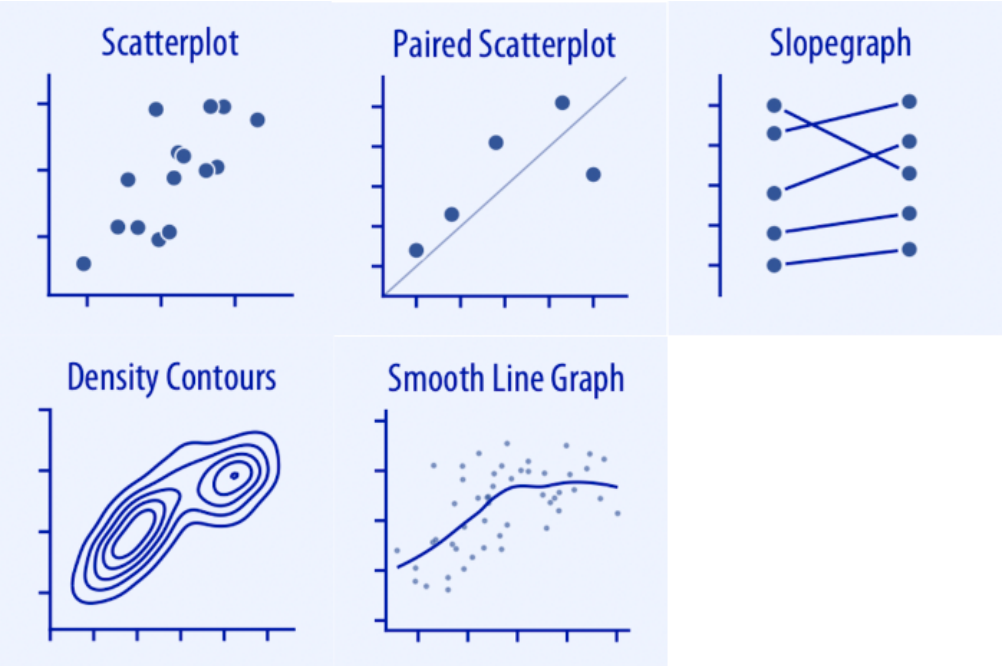
A simple example

- Using the package fBasics, R can calculate the sample skewness and kurtosis using the library timeDate by calling the functions skewness and kurtosis; read their help files for the method used.
- Other packages have other versions of the two functions.
- Looking at the mtcars data set, and in particular, exploring the variables mpg and disp:

```
> skewness(mtcars$mpg)
[1] 0.610655
attr(,"method")
[1] "moment"
> skewness(mtcars$disp)
[1] 0.381657
attr(,"method")
[1] "moment"
> kurtosis(mtcars$mpg)
[1] -0.372766
attr(,"method")
[1] "excess"
> kurtosis(mtcars$disp)
[1] -1.207212
attr(,"method")
[1] "excess"
```

Bivariate numerical data

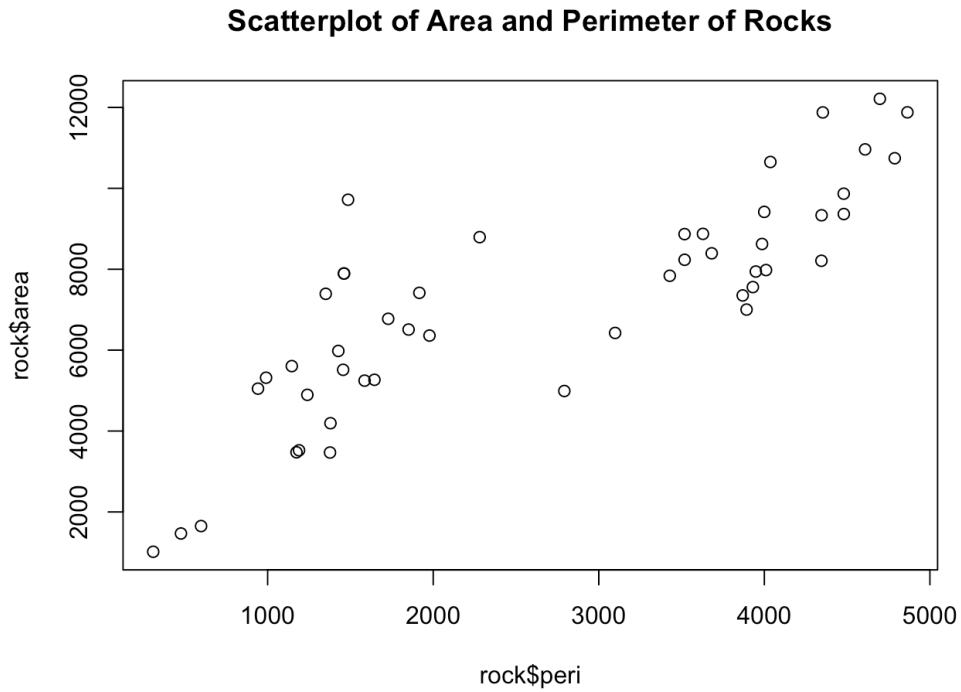
- Bivariate data contains observations on two (numerical) variables; some common displays of such bivariate data are:



- The most common is probably the scatterplot which shows the *relationship* between the two variables.

Scatterplots

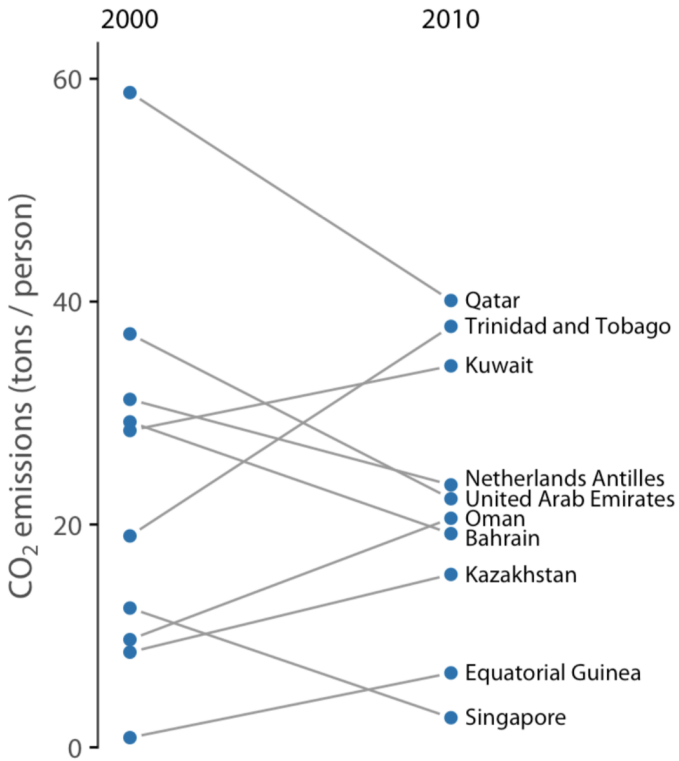
- Bivariate data typically arises when two variables are measured on each unit/subject.
- Looking at the rock data set once again, we can type plot(rock$peri,rock$area,main="Scatterplot of Area and Perimeter of Rocks") to get:



From the plot it appears there is a positive linear relationship between the two variables.

Slope graph

- When data consists of what may be thought of as *before* and *after* data, slope graphs can be very informative; they show the change in a variable between the two times using lines (slopes).
- Consider this slope graph showing the CO_2 emissions in the 10 countries with the largest difference per year:



Numerical measures for bivariate data

- The most common numerical tool for looking at the association between two numerical variables is correlation as defined in the last row of the table on Slide 23.

- Recall that correlation measures the direction and strength of the linear relationship between two quantitative variables (if it exists).

- In R, for the cars data set, looking at the variables mpg and disp, we use the in-built function cor:

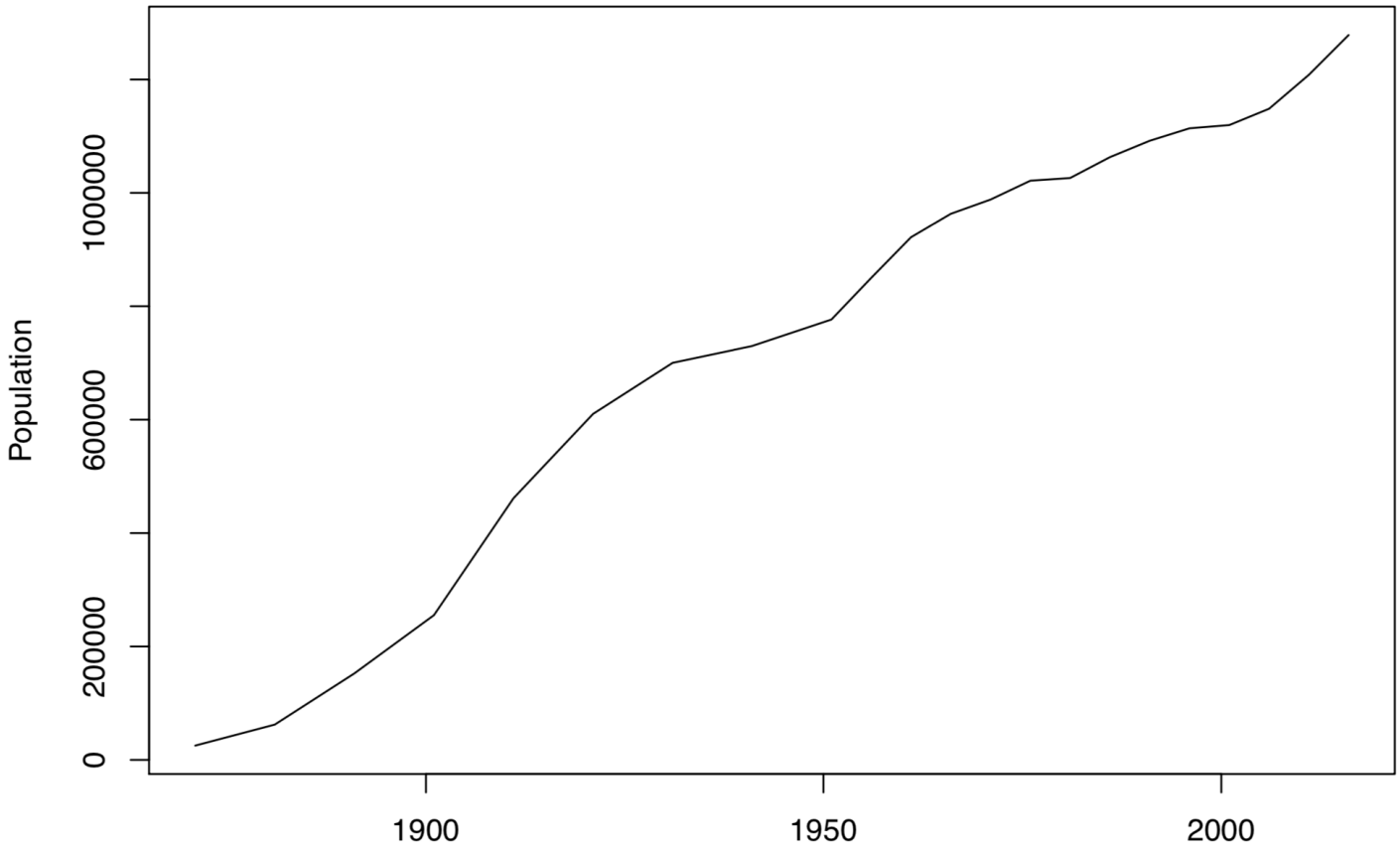
```
> cor(mtcars$mpg,mtcars$disp)
[1] -0.8475514
```

Time series data and time series plots

- A time series is a series of data points collected over time.
- Typically, data are collected at successive *equally spaced* points in time.
- Examples include
  - the daily air temperature or monthly precipitation in a specific location (say Winnipeg)
  - the annual yield of canola in a state
  - annual Manitoba population data
  - daily closing stock prices
  - weekly interest rates of a Canadian bank.
- A time series plot is a graph that we can use to evaluate patterns and behaviour in data over time.
- A time series plot displays observations on the y-axis against equally spaced time intervals on the x-axis.

An example

- Consider a time series consisting of Manitoba's population over time from 1871-2016.



Auto-correlation

- To investigate similarities amongst observations over time, one can look at the autocorrelation function.
- The autocorrelation function of a time series can be used to
  - detect non-randomness in data,
  - identify an appropriate time series model if the data are not random.
- Some details (that you **don't need to know**) about auto-correlation:
  - For a given time series $Y_1, Y_2, \ldots, Y_N$ collected at equally spaced time points, the lag-k auto-correlation function is defined as

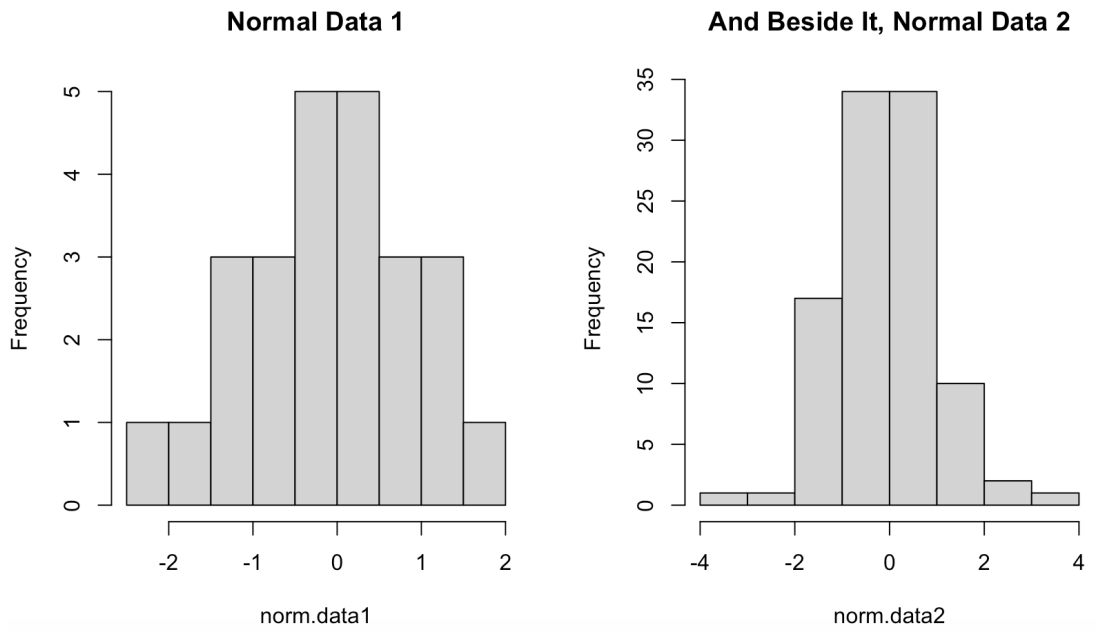
$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}$$

  - Note that autocorrelation is a correlation coefficient; instead of correlation between two different variables, the correlation is between two values of the same variable at time points X_i and X_{i+k}.

Some notes on plots

- If producing more than one plot, they can be displayed in an array; read ?par for details on adjusting graphical parameters.
- Consider the following code and resulting output:

```
> par(mfrow=c(1,2))
> hist(norm.data1,main="Normal Data 1")
> hist(norm.data2,main="And Beside It, Normal Data 2")
```



- Important note: to clear plots and reset graphical parameters, one needs to type dev.off().

Some more notes on plots

- There are many different plots available in R, as well as packages which can create more sophisticated plots.
- Whatever plotting function that is used, the appearance of the plot is often flexible, i.e., the colour, the size, etc.
- For example, in a simple scatterplot (using R's plot), the shape and colour of the points can be changed using the additional arguments pch and col, respectively.
- One popular package these days which is devoted to data visualization is ggplot2.

High dimensional data

- High dimensional data consists of observations on many variables, which can be in the hundreds and thousands.
- Visualizing high dimensional data can be challenging, but with the emergence of large amounts of data in recent years, new tools are being created.
- In R, in addition to ggplot2, many packages have been made, such as:
    - scatterplot3d
    - plot3D
    - tabplot
    - lattice
- There is also a collection of packages called tidyverse which includes packages such as ggplot2, dplyr and tidyr which focus on data visualization, manipulation and exploration.

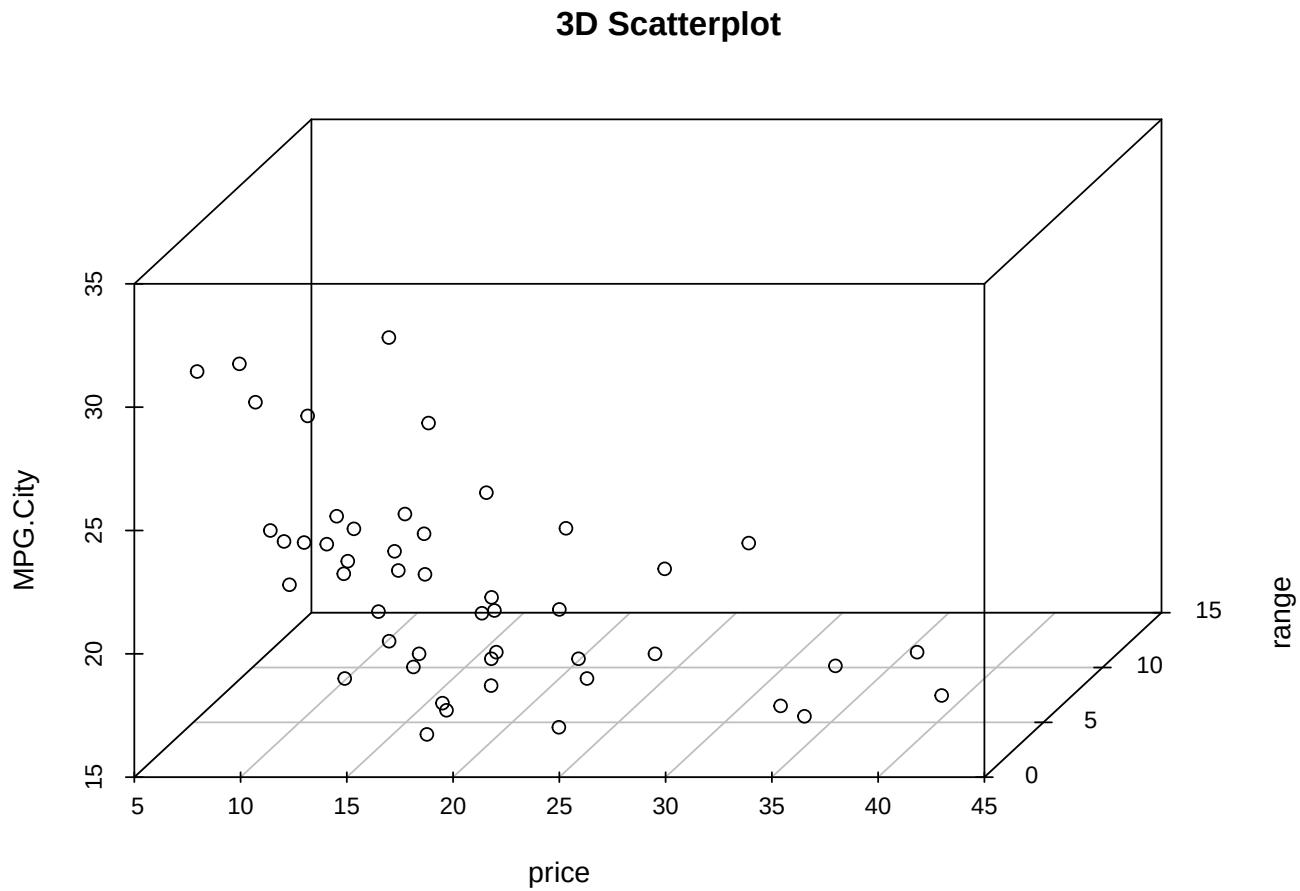
3D Scatterplots

- In a 3D scatterplot, data points on three axes are plotted in order to show the relationship between three variables.
- Each row in the data table is represented, where position depends on its values in the columns set on the x, y, and z axes.
- If the points are close to a straight line in any direction in the three-dimensional space of the 3D scatterplot, the correlation between the corresponding variables is high.
- If the markers are equally distributed in the 3D scatterplot, the correlation is low, or zero.

An example

- Consider a data set on US car prices

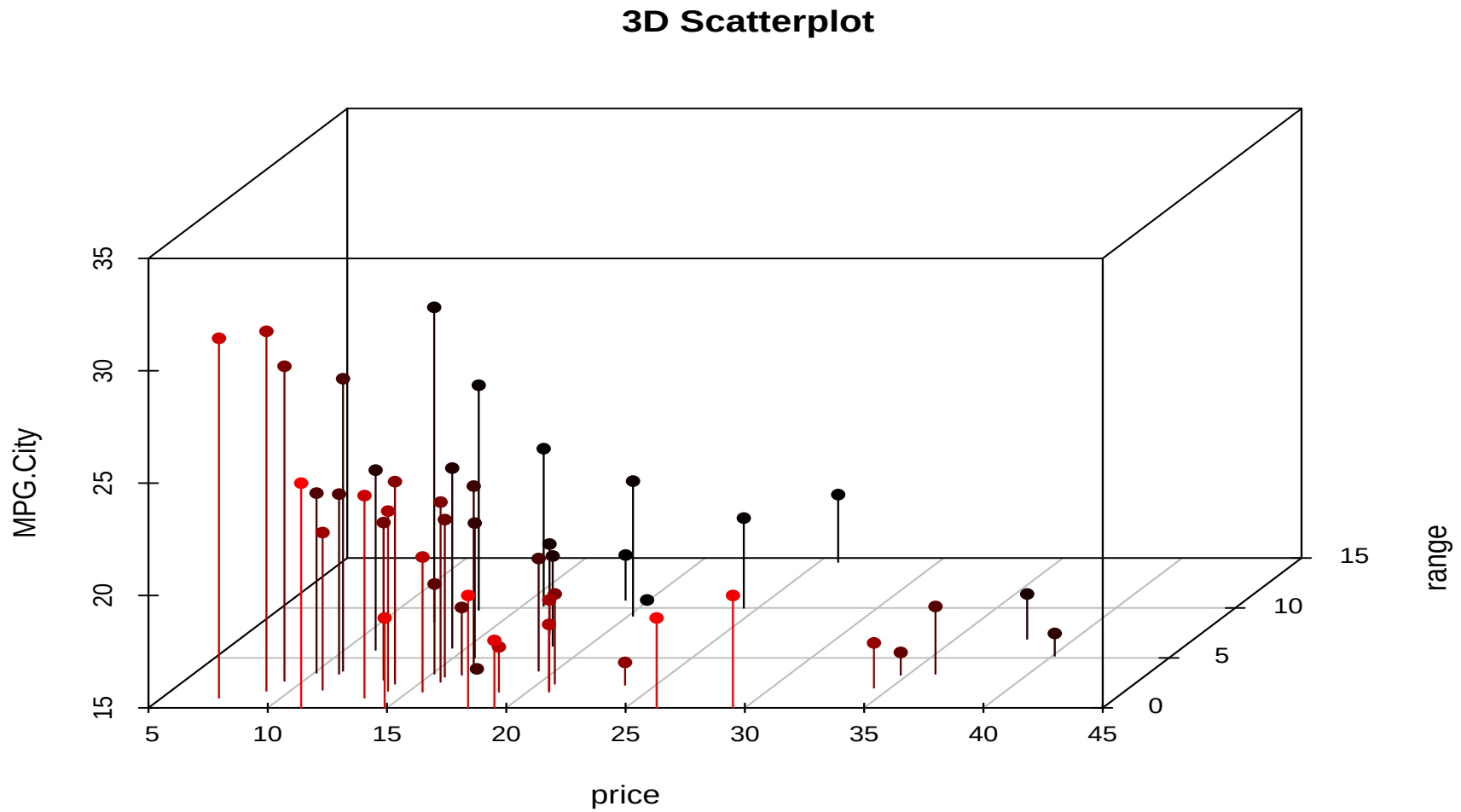
```
> library(scatterplot3d)
> scatterplot3d(car.data[,3],car.data[,5],car.data[,8], main="3D Scatterplot",xlab="price",
ylab="range",zlab="MPG.City")
```



An example

- We can change the look of the scatterplot by changing some of the values of the arguments:

```
> scatterplot3d(car.data[,3],car.data[,5],car.data[,8], pch=16, highlight.3d=TRUE, type
="h", main="3D Scatterplot",xlab="price",ylab="range",zlab="MPG.City")
```

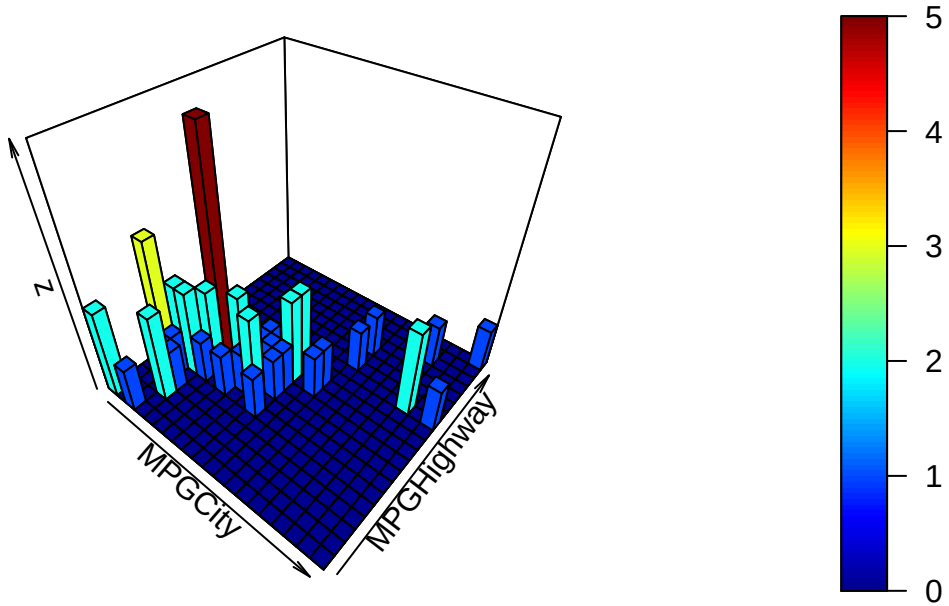


3D histograms

- In a 3D histogram, there are two variables X and Y and bivariate frequencies are plotted on the z-axis.
- That is, data are from two variables and the number of occurrences of the data in a two dimensional grid are represented by 3D bars.
- The grid and bin are defined by the user.

An example

```
> library(plot3D)
> x = car.data[,8]
> y = car.data[,9]
> xbin = cut(x,20)
> ybin = cut(y,20)
> bincount = table(xbin,ybin)
> hist3D(z=bincount,border="black",xlab="MPGCity",ylab="MPGHighway")
```



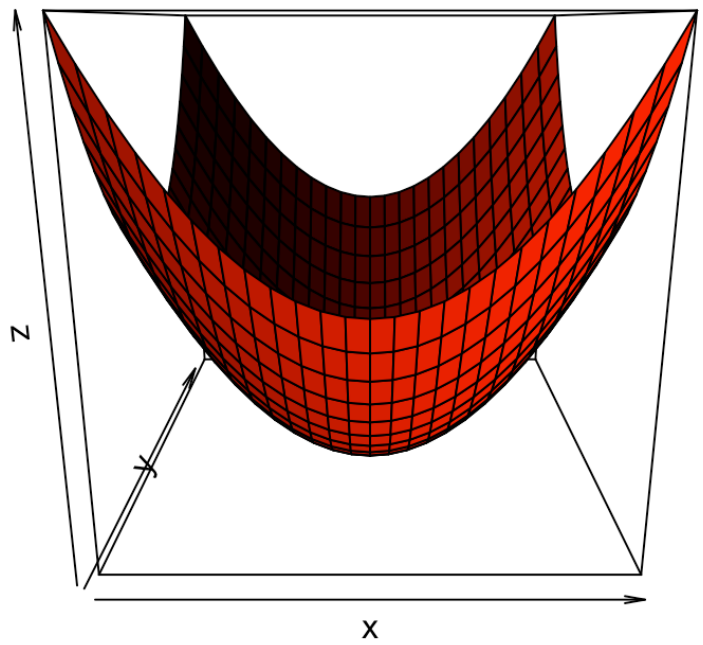
Contour plots

- A contour plot is a graphical tool for representing a 3D surface by plotting constant z slices, called contours, on a 2-dimensional grid.
- Given a value for z, lines are drawn for connecting the (x, y) coordinates where that z value occurs.
- The contour plot can be used to asses how the variable Z changes as a function of X and Y.

An example

- Consider visualizing the shape of the following three-dimensional function:
$$z = f(x, y) = x^2 + y^2; -1 < x, y < 1.$$
- Using R's functions outer and persp:

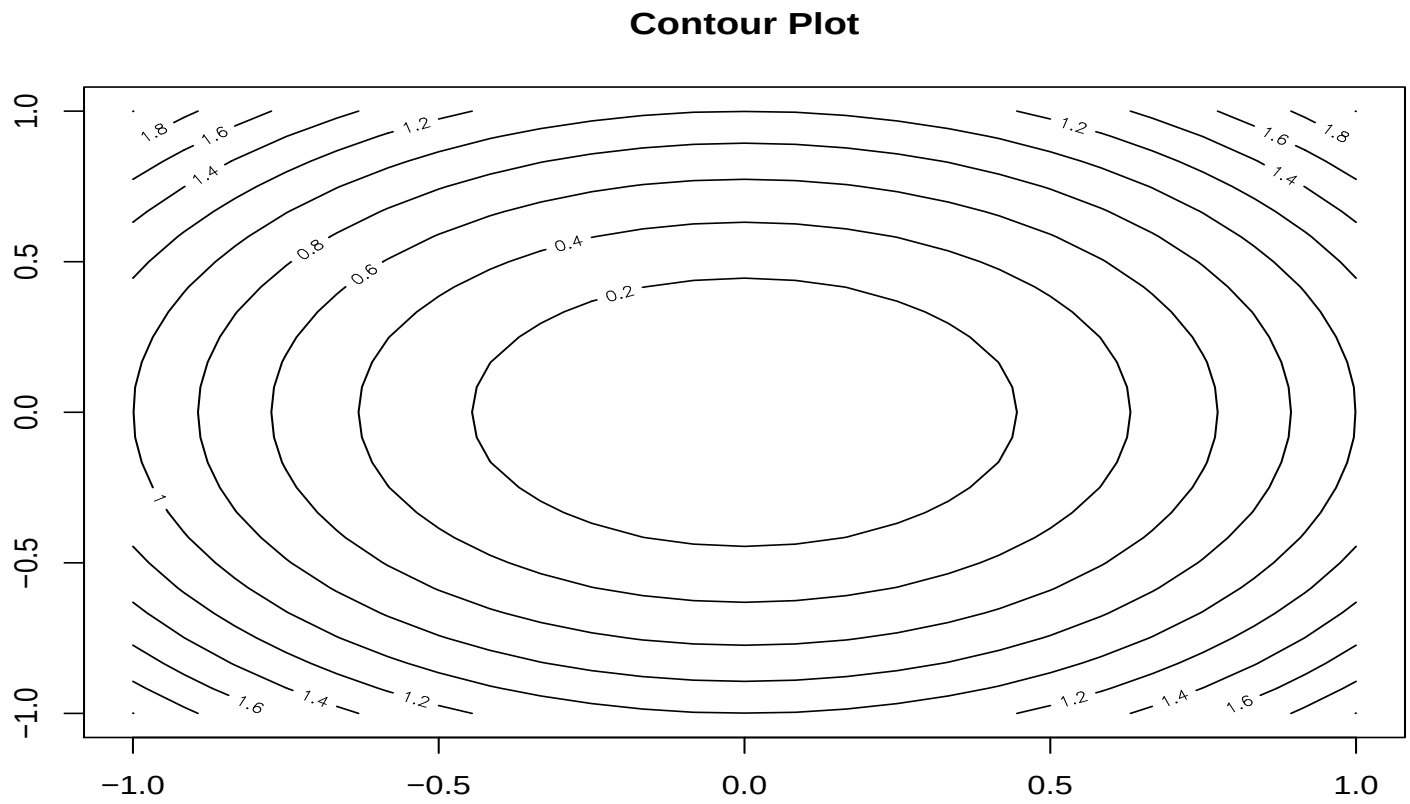
```
>
>
>
>
>
> x <- seq(-1,1,len=25)
> y <- seq(-1,1,len=25)
> z <- outer(x, y, FUN=function(x,y) x^2 +y^2)
> persp(x,y,z, shade=.75, col="red")
>
>
>
>
>
>
>
>
>
>
>
>
>
```



- We can further visualize the 3D surface with a contour plot.

An example

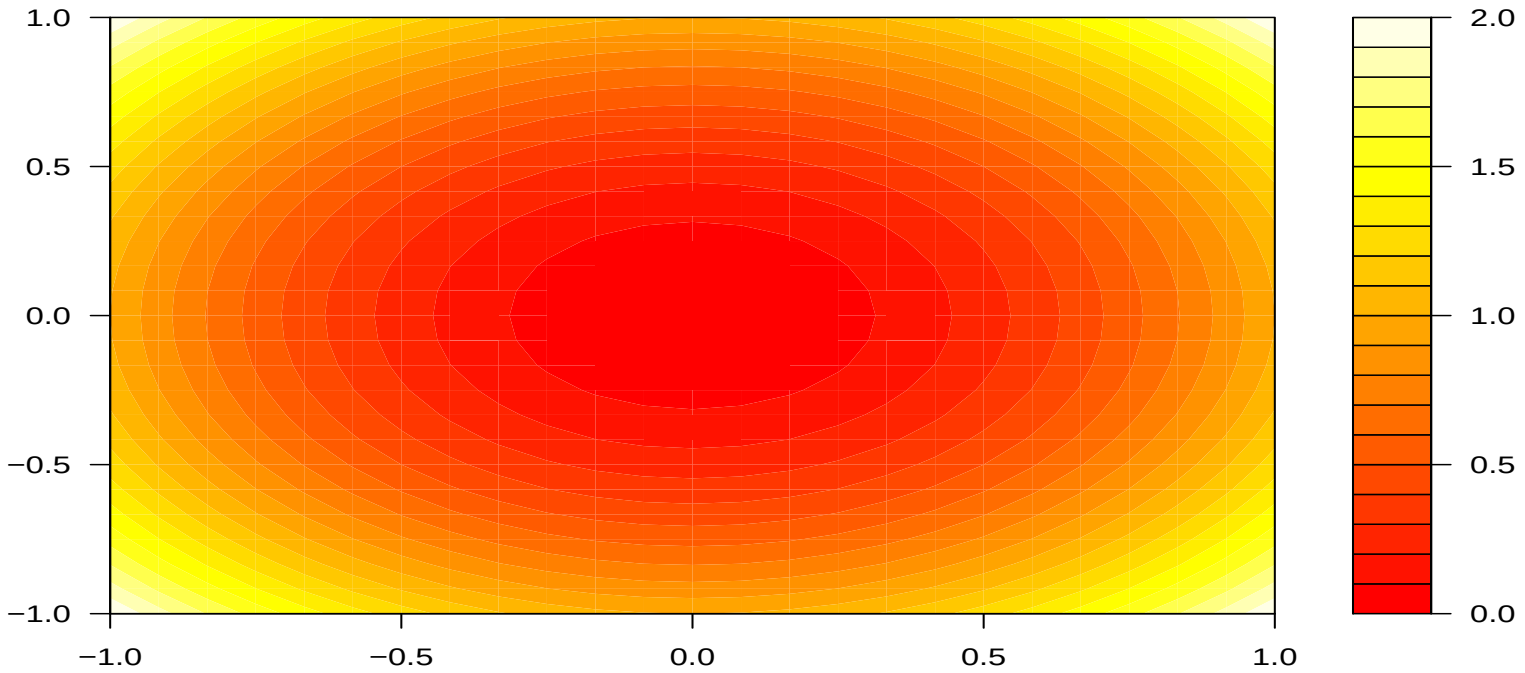
```
> contour(x,y,z, main="Contour Plot")
```



An example

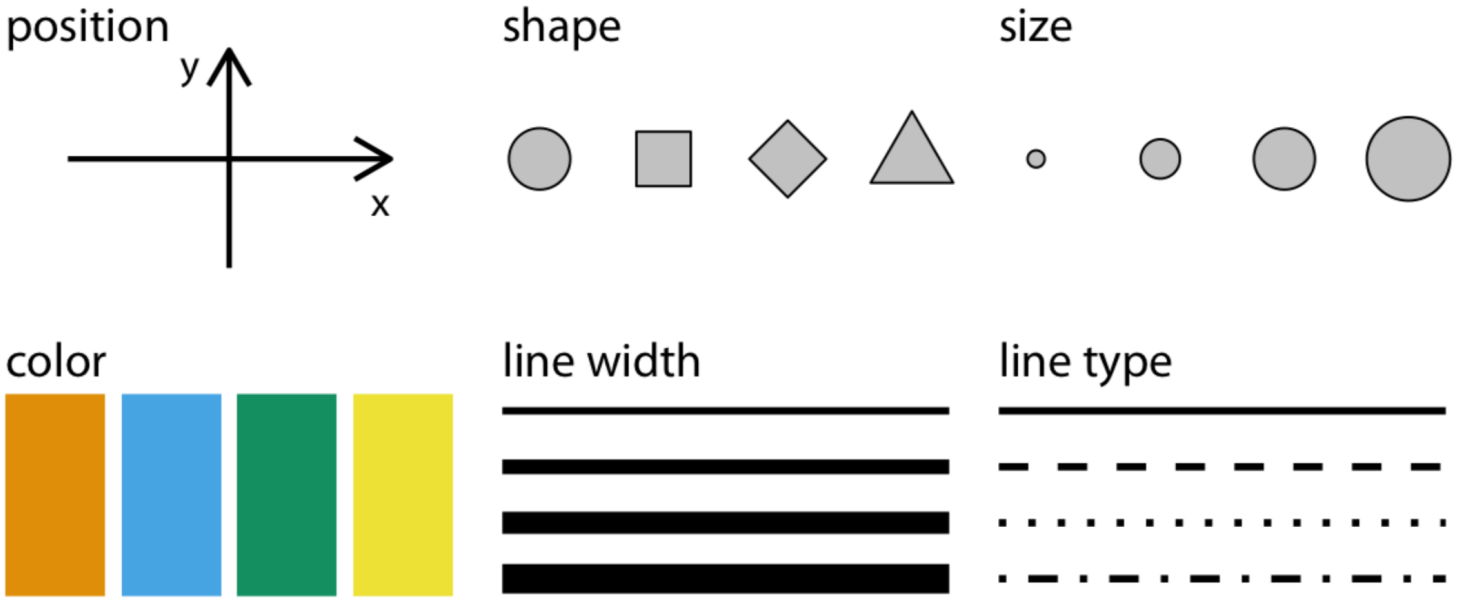
- We can also plot a different type of contour plot:

```
> contour(x,y,z, main="Contour Plot")
> filled.contour(x,y,z, color.palette = heat.colors)
```



Visualization goals

- There are some goals (rules) to follow when visualizing data:
  1. Clarity - present the data in a way that is meaningful and appropriate.
  2. Precision - represent data accurately.
  3. Aesthetics - choose visualizations that are easy to perceive and interpret.



  4. Be efficient - present only that which needs presenting.