EXPLORATORY DATA ANALYSIS (EDA)

Introducción a la Ciencia de Datos

Coral del Val Muñoz

Dept. Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada
Dept. Molecular Biophysics, German Cancer Research Center Heidelberg, Alemania

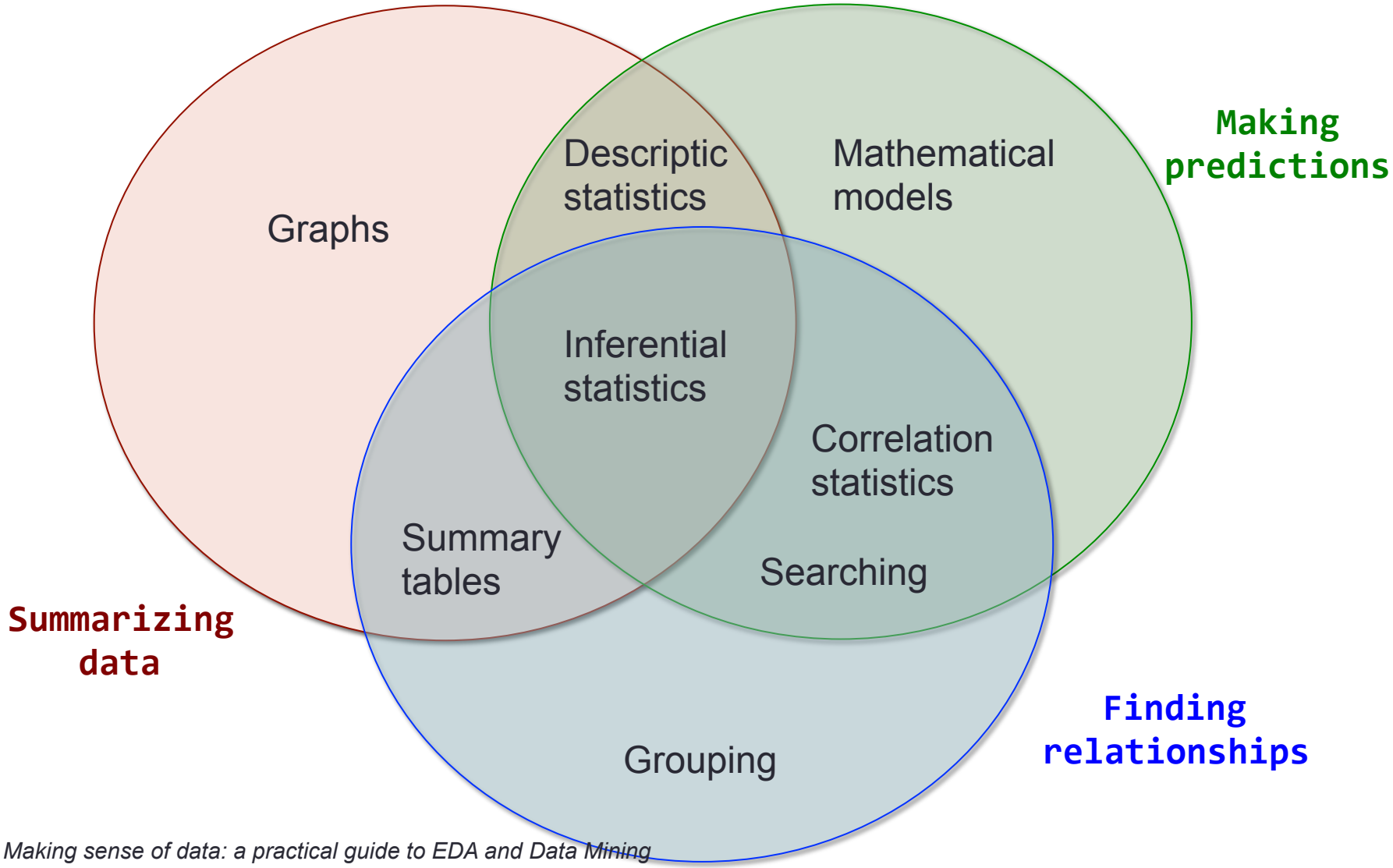
Index

What is EDA?

- In statistics, exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods.

  - Gain insight into a data set
  - Discover patterns in the data
  - Extract important variables
  - Detect outliers and anomalies
  - Identify transforms (e.g. log(x))
  - Generate hypothesis

Data Analysis tasks and methods



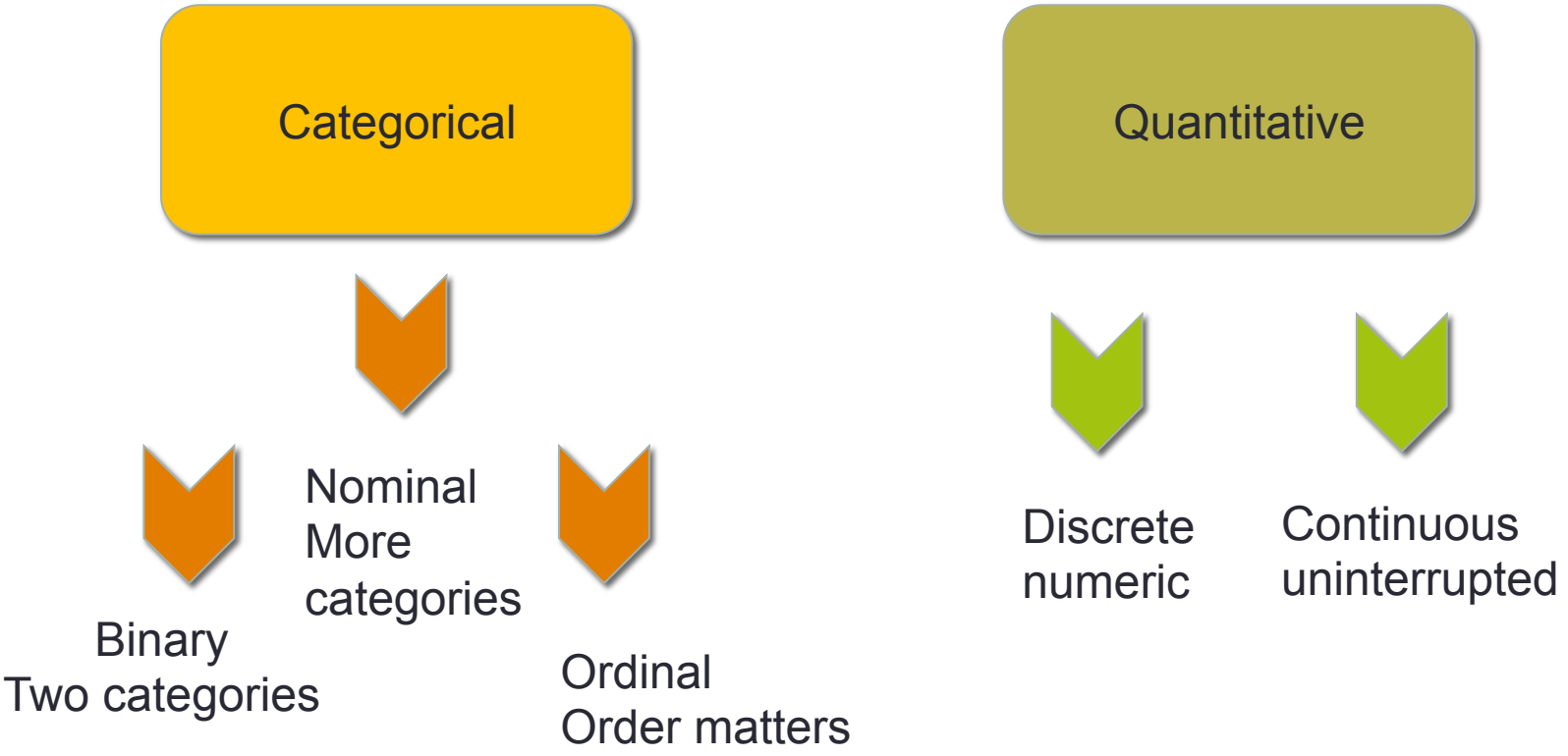
Source: Making sense of data: a practical guide to EDA and Data Mining

EDA techniques

Most EDA techniques consist of:

- Plotting raw data (e.g. histograms, probability plots, scatter plots).
- Plotting simple statistics (e.g. mean plots, standard deviation plots, box plots)
- Use those plots to maximize our natural pattern-recognition abilities.

Types of data

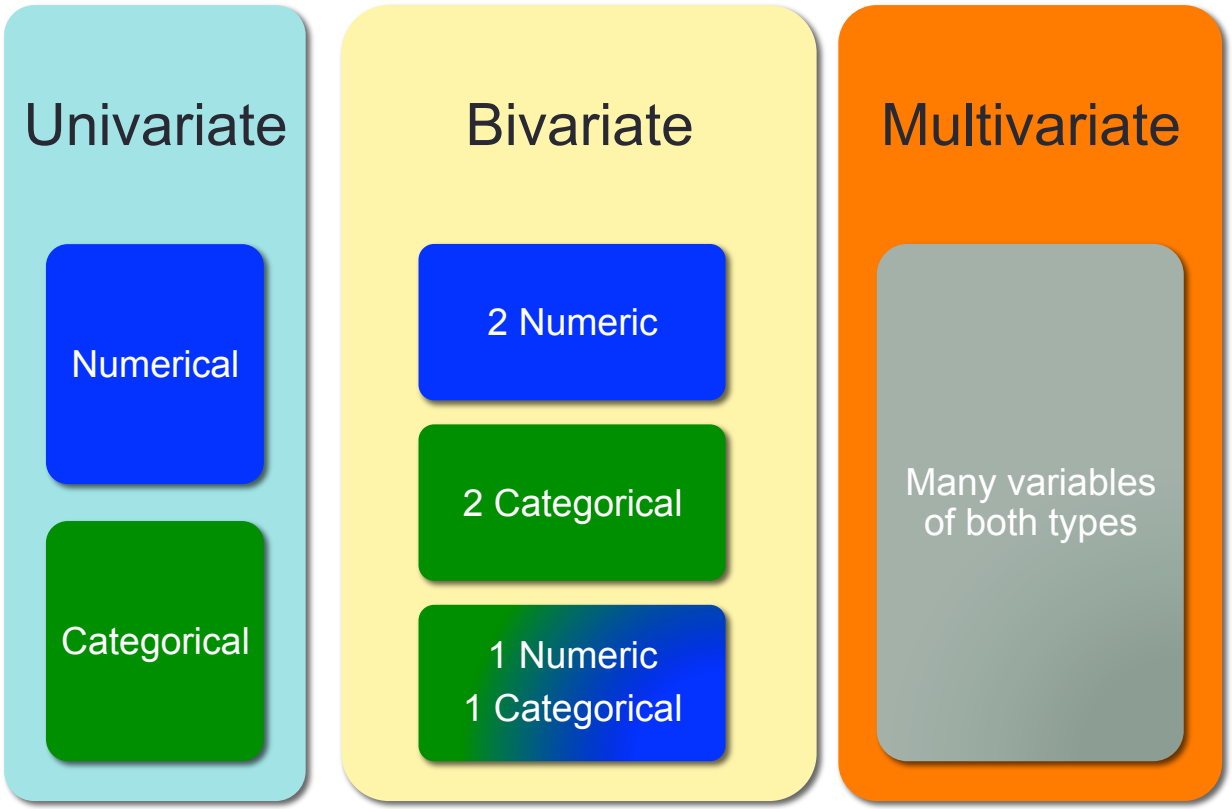


Data dimensionality

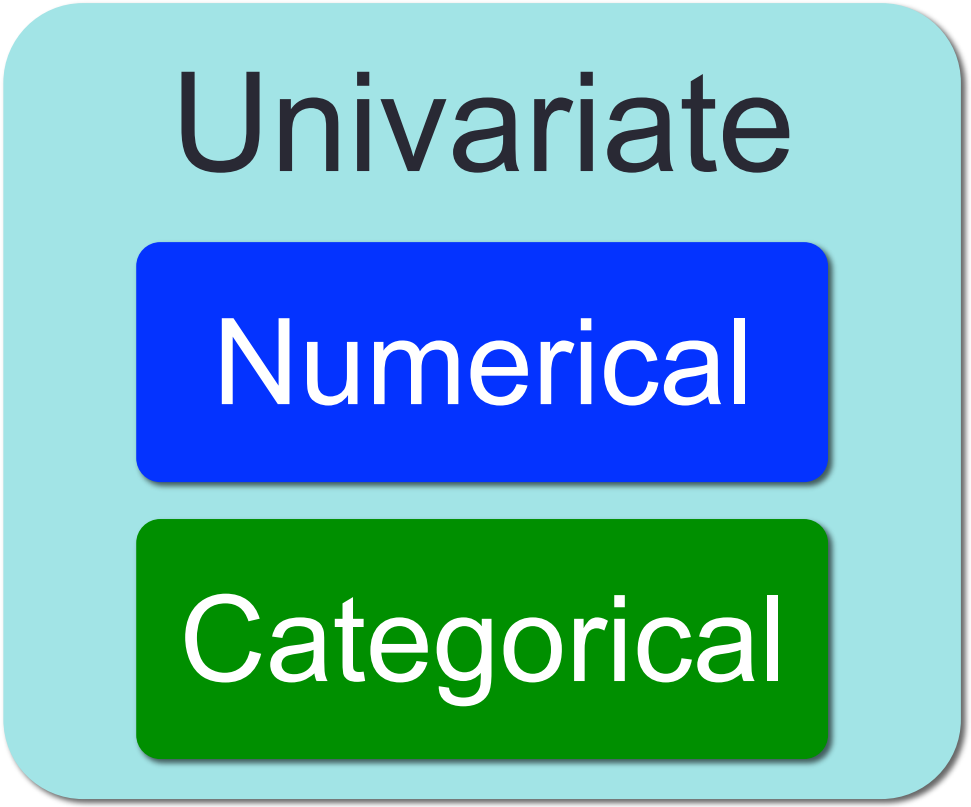
Univariate:
Measurement made on one variable per Subject

Bivariate:
Measurement made on two variables per Subject

Multivariate:
Measurement made on many variables per subject



Univariate quantitative



Examining distributions

- In order to convert raw data into useful information we need to:

'quantify' a data set, using a set of *summary statistics* (e.g. mean, median, variance, standard deviation)

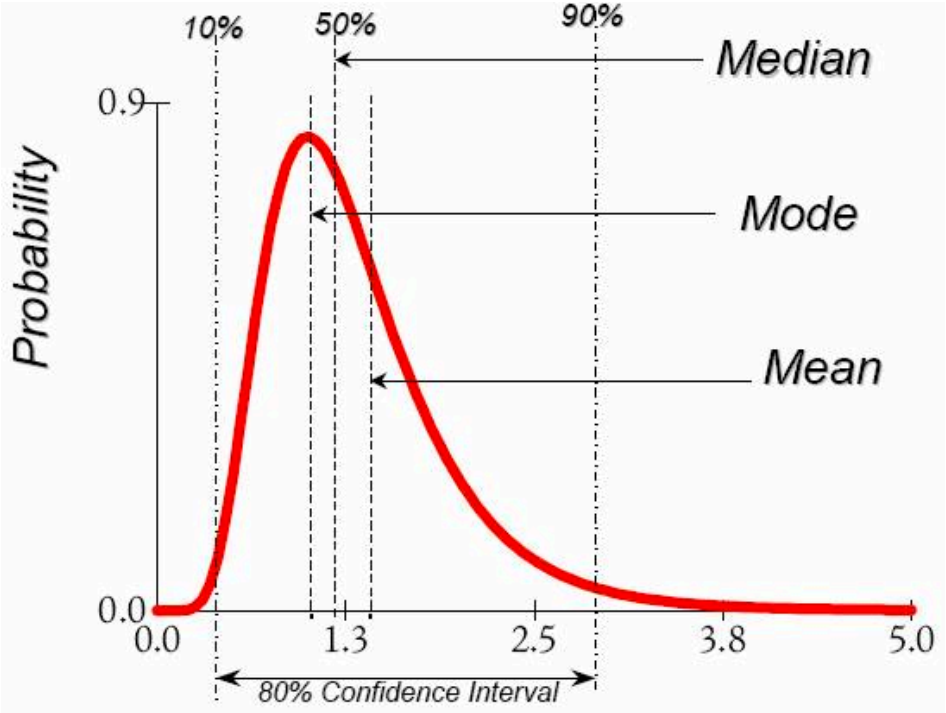
Numerical summaries of functions

- **Central Tendency measures.** They are computed to give a “center” around which the measurements in the data are distributed.
- **Variation or Variability measures**. They describe “data spread” or how far away the measurements are from the center.
- **Relative Standing measures**. They describe the relative position of specific measurements in the data.

Univariate analysis: Central tendency measures

- `mean()`
- `median()`
- `mode()`

- When using most of these functions remember to use argument **`na.rm = T`**

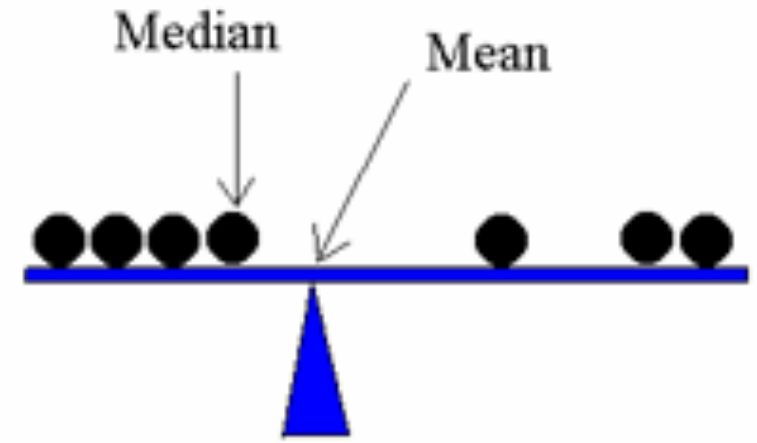


source http://herdingcats.typepad.com/photos/uncategorized/statistics.jpg

Mean or Median?

- Mean is best for symmetric distributions without outliers.
- The mean is not a robust tool since it is largely influenced by outliers.

- Median is useful for skewed distributions or data with outliers. It derives at central tendency since it is much more robust and sensible.

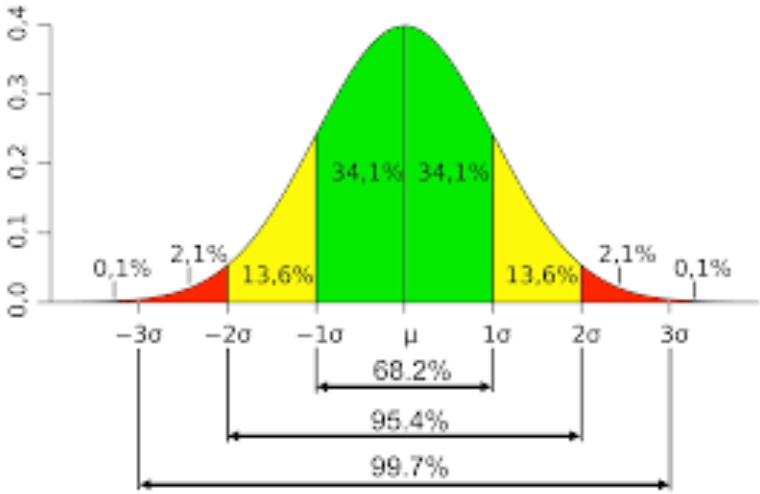


Source: https://onlinecourses.science.psu.edu

Univariate analysis: dispersion measures

- **Standard deviation:** The average amount the the scores deviate from the mean. `std()`
- **Variance:** The variance is the mean of the squares of the individual deviations. `var()`
- **Minimum**: `min()`
- **Maximum:** `max()`
- **Range:** The maximun difference in the data. `range()`

- When using most of these functions remember to use argument **`na.rm = T`**



source http://projectmanager.com.au/can-you-use-standard-deviation-in-project-management/

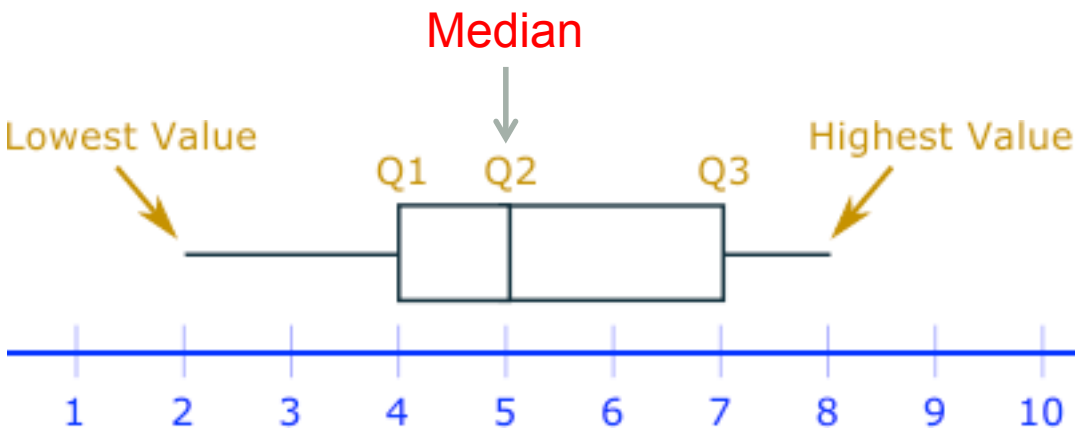
Normally distributed data, approximately 95% of the values lie within 2 sd of the mean.

Short Description of Descriptive Statistics and R Functions

Parameter	Description	R function
Mean	arithmetic average	`mean()`
Median	middle value, 50% quantile	`median()`
Mode	most frequent value	`sort(table(), decreasting = TRUE)[1]`
Standard Deviation	variation around mean	`sd()`
Quantiles	percent rank of values, such that all values are <= p	`quantile()`

Univariate analysis: quartiles

- The quartiles of a population or a sample are the three values which divide the distribution or observed data into even fourths. `quantiles()`
- remember to use argument **`na.rm = T`**



source https://www.mathsisfun.com/data/quartiles.html

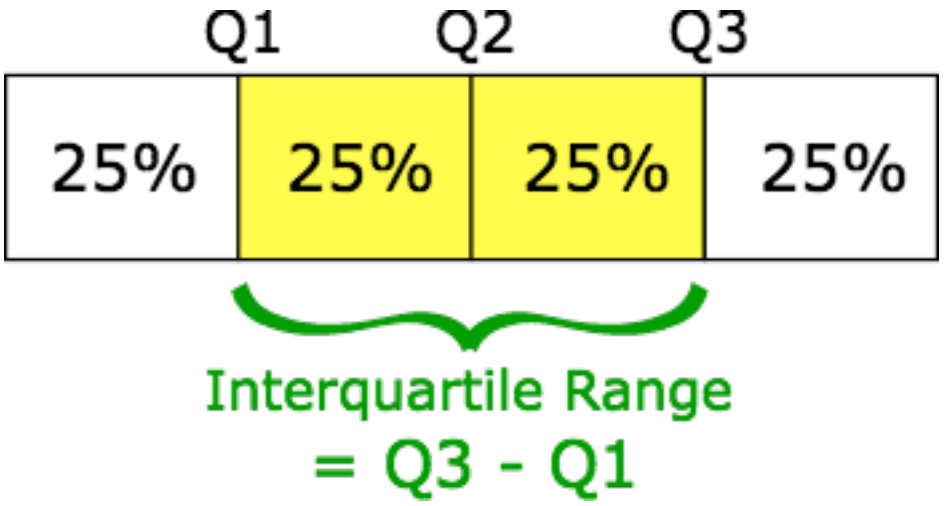
Univariate analysis: percentiles (aka quantiles)

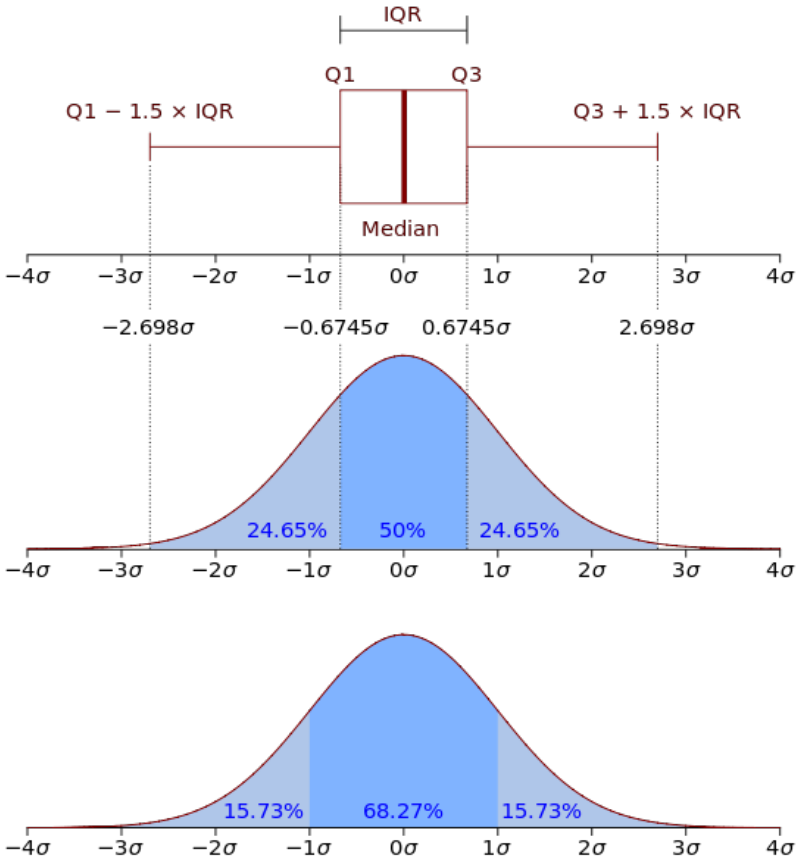
The *p*-quantile has the property that *p*% of the observations are less than or equal to it.

```
> set.seed(100)
> x <- rnorm(100, mean=0, sd=1)
> quantile(x)
        0%        25%        50%        75%       100% 
-2.2719255 -0.6088466 -0.0594199  0.6558911  2.5819589 
> quantile(x, probs=c(0.1, 0.2, 0.9))
       10%        20%        90% 
-1.1744996 -0.8267067  1.3834892 
```

Univariate analysis: inter quartile range

- The inter quartile range (IQR) is a more robust measure of spread `IQR()`
- remember to use argument **`na.rm = T`**





source wikipedia

Other measures: Skewness & Kurtosis

In comparison to a Gaussian model:

- Skewness is a measure of asymmetry.
- Kurtosis is a measure of “peakedness”

- Values for skewness and kurtosis close to zero indicate that the shape of a frequency distribution for a variable approximates a normal distribution which is important for checking assumptions in certain data analysis methods.

```
#Simulation
n.sample <- rnorm(n = 10000,
mean = 55, sd = 4.5)

#Skewness and Kurtosis
library(moments)

skewness(n.sample)

kurtosis(n.sample)
```

Contingency tables

- For categorical data we can use contingency tables.

```
table(iris$Species)

    setosa versicolor  virginica
        50         50         50
```

Univariate analysis: Five numbers summary

The `summary()` function prints some basic descriptive statistics (including the count of missing values) for not only one, but also multiple variables, for example:

```
> summary(rage)

      rage
 Min.   :18.00
 1st Qu.:35.00
 Median :48.00
 Mean   :49.62
 3rd Qu.:64.00
 Max.   :97.00
 NA's   :3
```

Hmisc package

- In order to know more about the dataset such as the missing values, distribution of numerical variables, and distinct values of categorical variables, we can use an additional package called Hmisc

```
> library (Hmisc)
> describe(iris)

5  Variables      150  Observations
---------------------------------------------------------------------------------------------
Sepal.Length
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75
     150        0       35    0.998    5.843   0.9462    4.600    4.800    5.100    5.800    6.400
     .90      .95
   6.900    7.255

lowest : 4.3 4.4 4.5 4.6 4.7, highest: 7.3 7.4 7.6 7.7 7.9
Species
       n  missing distinct
     150        0        3

Value          setosa versicolor  virginica
Frequency          50         50         50
Proportion      0.333      0.333      0.333
---------------------------------------------------------------------------------------------
```

Descriptive Statistics

```
> set.seed(100)
> x <- rnorm(100, mean=0, sd=1)
> mean(x)
[1] 0.002912563
> median(x)
[1] -0.0594199

# measure of statistical dispersion, IQR = Q3 -  Q1
> IQR(x)
[1] 1.264738
> var(x)
[1] 1.04185
> summary(x)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.272000 -0.608800 -0.059420  0.002913  0.655900  2.582000
```

Exploring data

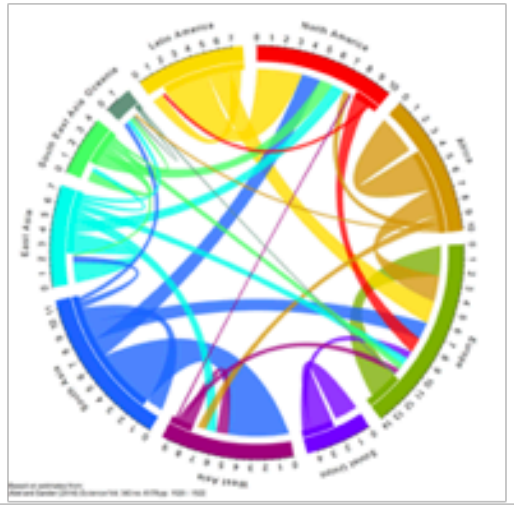
```
# Provides basic descriptive statistics and frequencies.
summary(mydata)
# Open data editor
edit(mydata)
# Provides the structure of the dataset
str(mydata)
# Lists variables in the dataset
names(mydata)
# First 6 rows of dataset
head(mydata)
# All rows but the last 10
head(mydata, n= -10)
# Last 6 rows
tail(mydata)
# First 10 rows
mydata[1:10, ]
# First 10 rows of data of the first 3 variables
mydata[1:10,1:3]
```

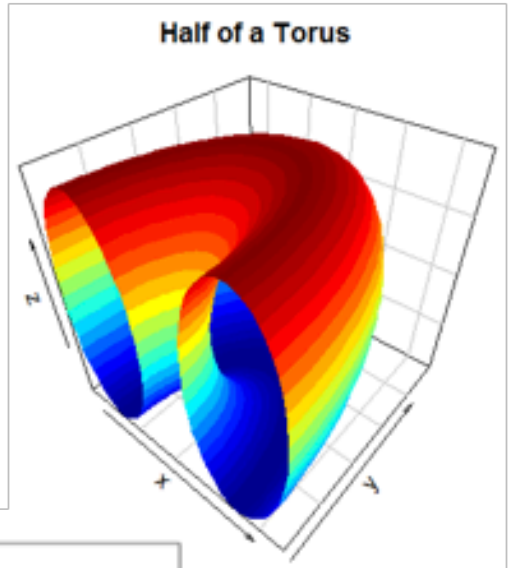
Data analysis workflow

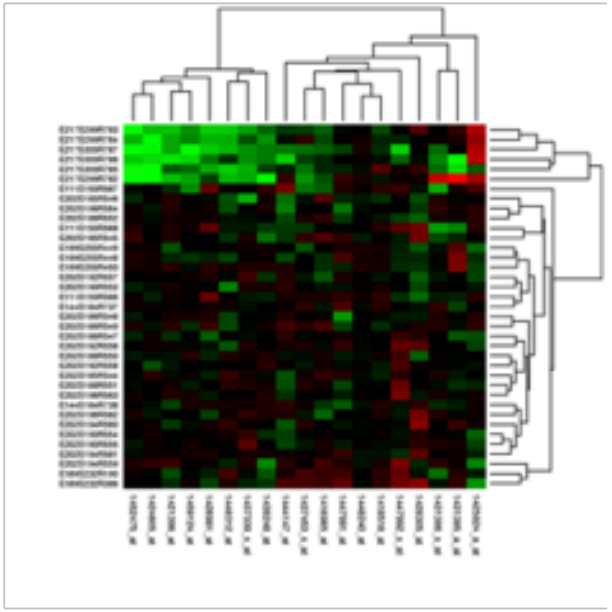
From R data analysis cookbok 2nd edition

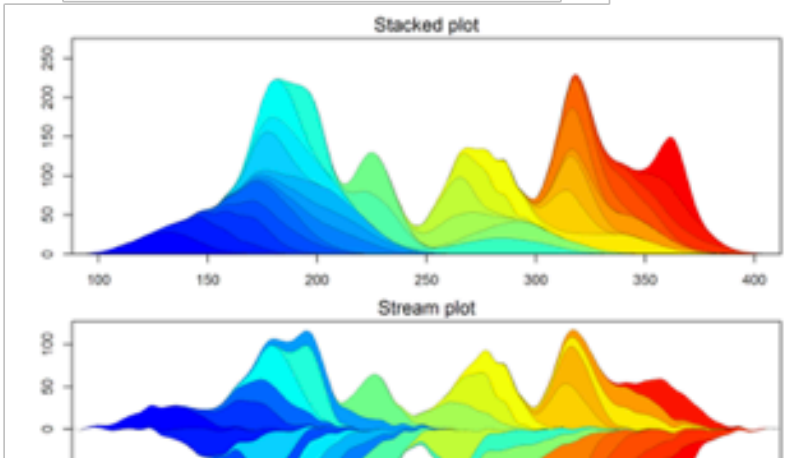
Data Visualization

Gallery of plots









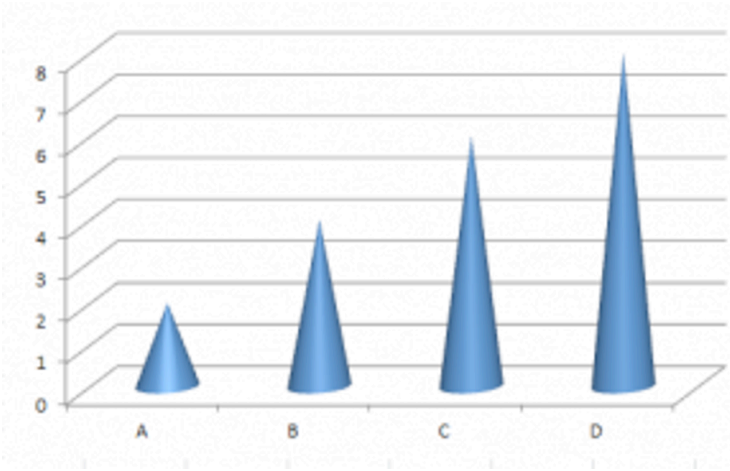
R plots

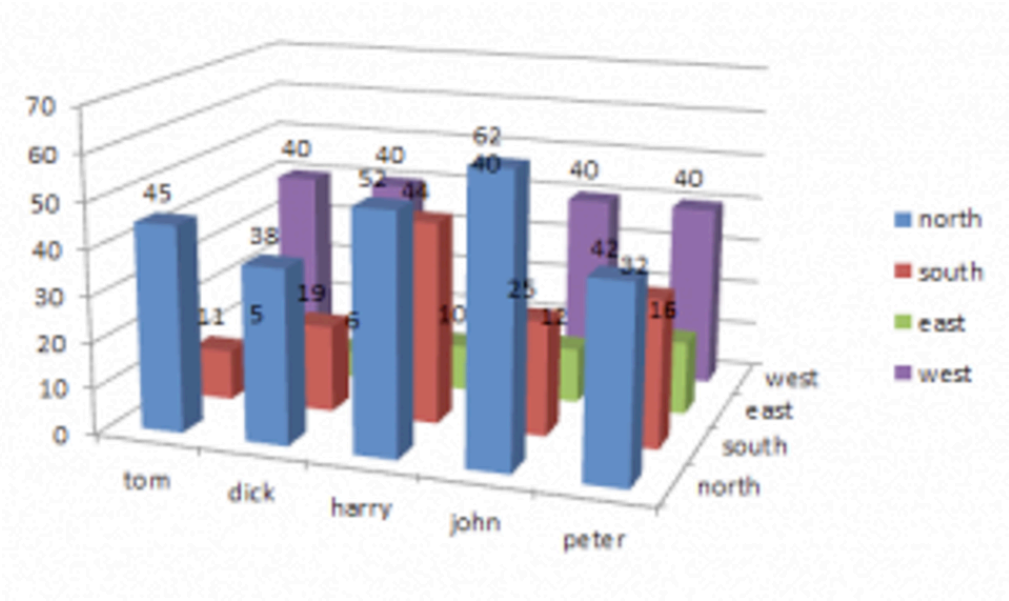
- We will see a small set of them:
- Histograms
- Boxplots
- Barplots
- Pies

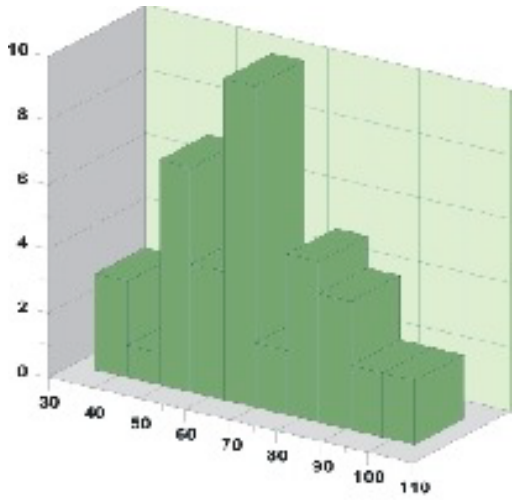
Data Visualization

- A graphic should display as much information as it can
- Strive for clarity.
- Make the data stand out:
  - Avoid too many superimposed elements, such as too many curves in the same graphing space.
  - Find the right aspect ratio and scaling to properly bring out the details of the data.
  - Avoid having the data all skewed to one side or the other of your graph.

Examples of bad plots







https://www.forbes.com/sites/naomirobbins/2012/05/30/winner-of-the-bad-graph-contest-announced-2/#35e9d5632e06

Univariate data: Graphical analyisis

- Histogram
- Box plots
- Bar plots

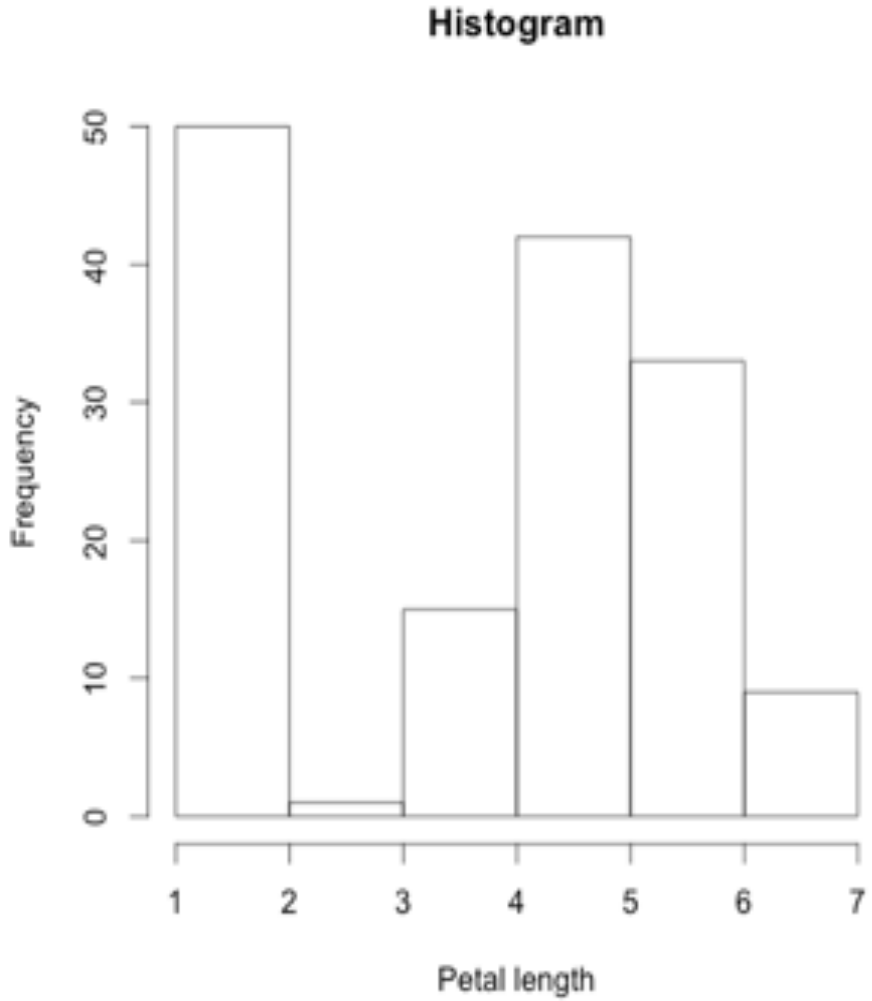
Histogram

- Each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values.
- It gives a general impression of the distribution's shape

- To construct a histogram, define the range of data for each bar (called a **bin**).
- Generally one will choose between about 5 and 30 bins, depending on the amount of data and the shape of the distribution.
- It is often worthwhile to try a few different bin sizes/ numbers especially with small samples

Histogram

```
hist(iris$Petal.Length,
main="Histogram",
xlab="Petal length”)
```

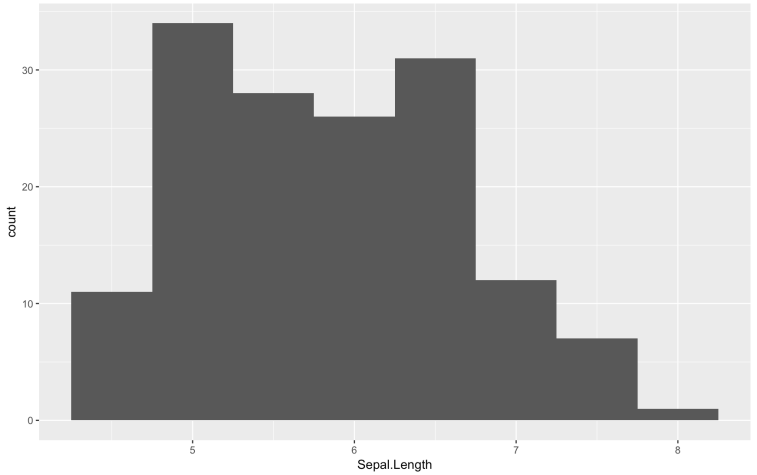


Histogram

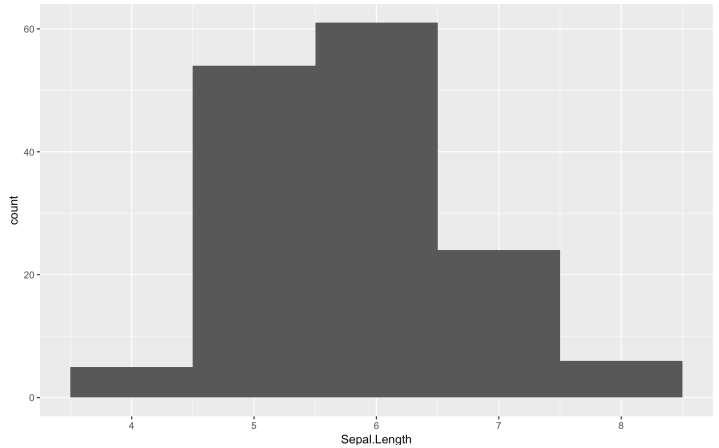
- With practice, histograms are one of the best ways to quickly learn a lot about your data, including central tendency, spread, modality, shape and outliers.

```
> ggplot(iris, aes(Sepal.Length))
+ geom_histogram(binwidth = 0.5)
> ggplot(iris, aes(Sepal.Length))
+ geom_histogram(binwidth = 1)
> ggplot(iris, aes(Sepal.Length))
+ geom_histogram(binwidth = 2)
```

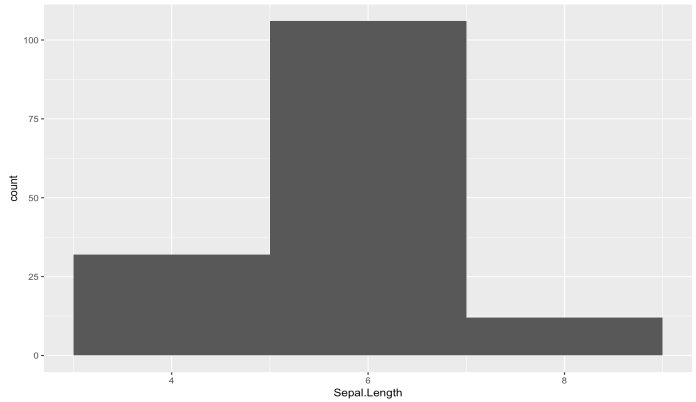
Bin=0.5



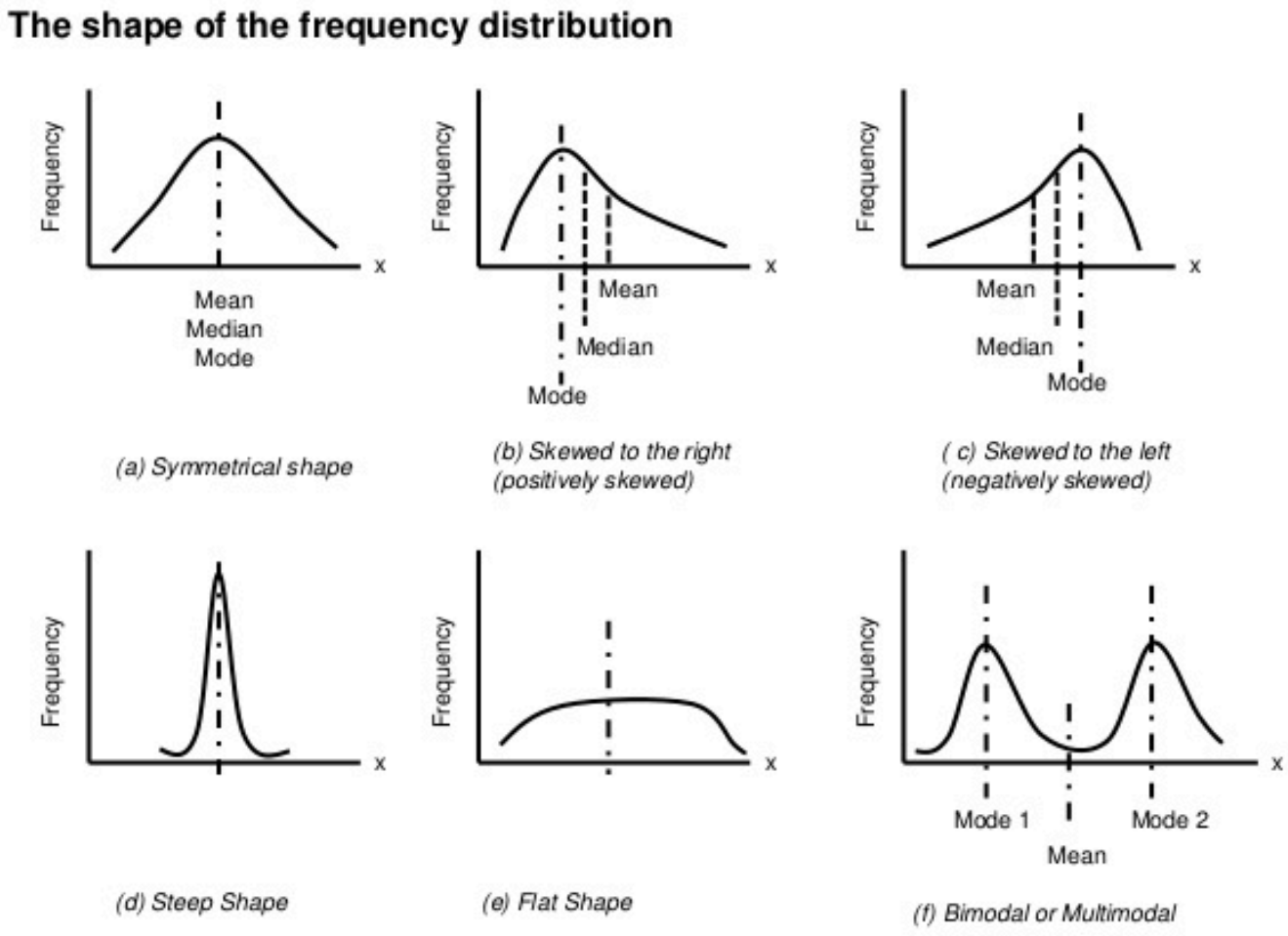
Bin=1



Bin=2



Common histogram distributions



Common Shapes of Frequency Distribution 53

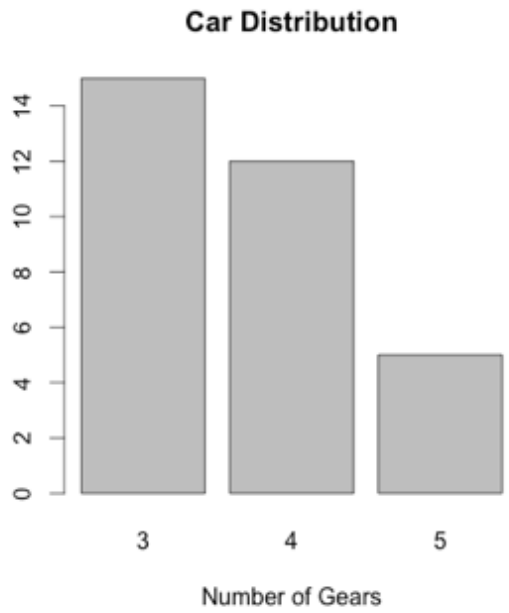
Bar graphs

- Bar charts are a common visual tool for displaying a single categorical variable.
- Categories are listed on the x-axis, and frequencies or proportions on the y-axis.
- The height of each bar represents either counts or percentages
- Easier to compare categories with bar graph than with pie chart

Bar graphs

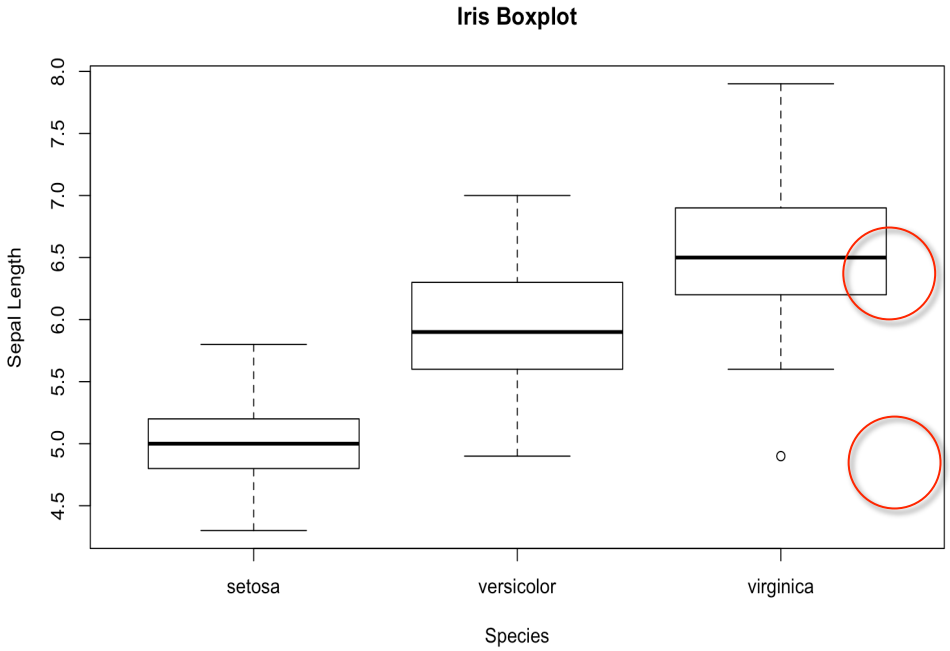
```
> mtcars[1:5,]
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
```

```
counts <- table(mtcars$gear)
barplot(counts, main="Car
Distribution", xlab="Number of Gears")
```



Boxplot

- Boxplots show robust measures of location and spread as well as providing information about symmetry of a frequency distribution and outliers.
- Is the way to visualize the five-number summary.




```
boxplot(Sepal.Length~Species,
data=iris, xlab="Species",
ylab="Sepal Length",
main="Iris Boxplot")
```

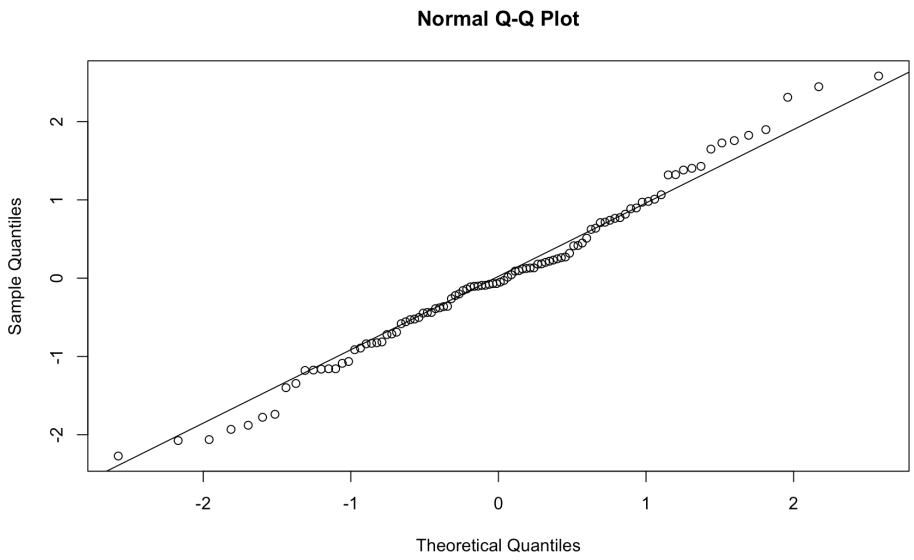
QQ-plot

- Many statistical methods make some assumption about the distribution of the data (e.g. normality).
- The quantile-quantile are often used to determine whether a dataset is normally distributed
- The QQ-plot shows the theoretical quantiles versus the empirical quantiles. If the distribution assumed (theoretical one) is indeed the correct one, we should observe a straight line.

QQ-plot

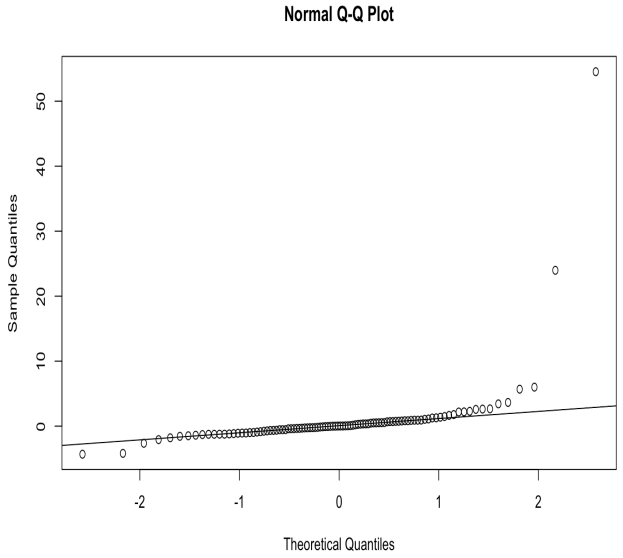
Distribucion normal

```
x<-rnorm(100, mean=0, sd=1)
qqnorm(x)
qqline(x)
```

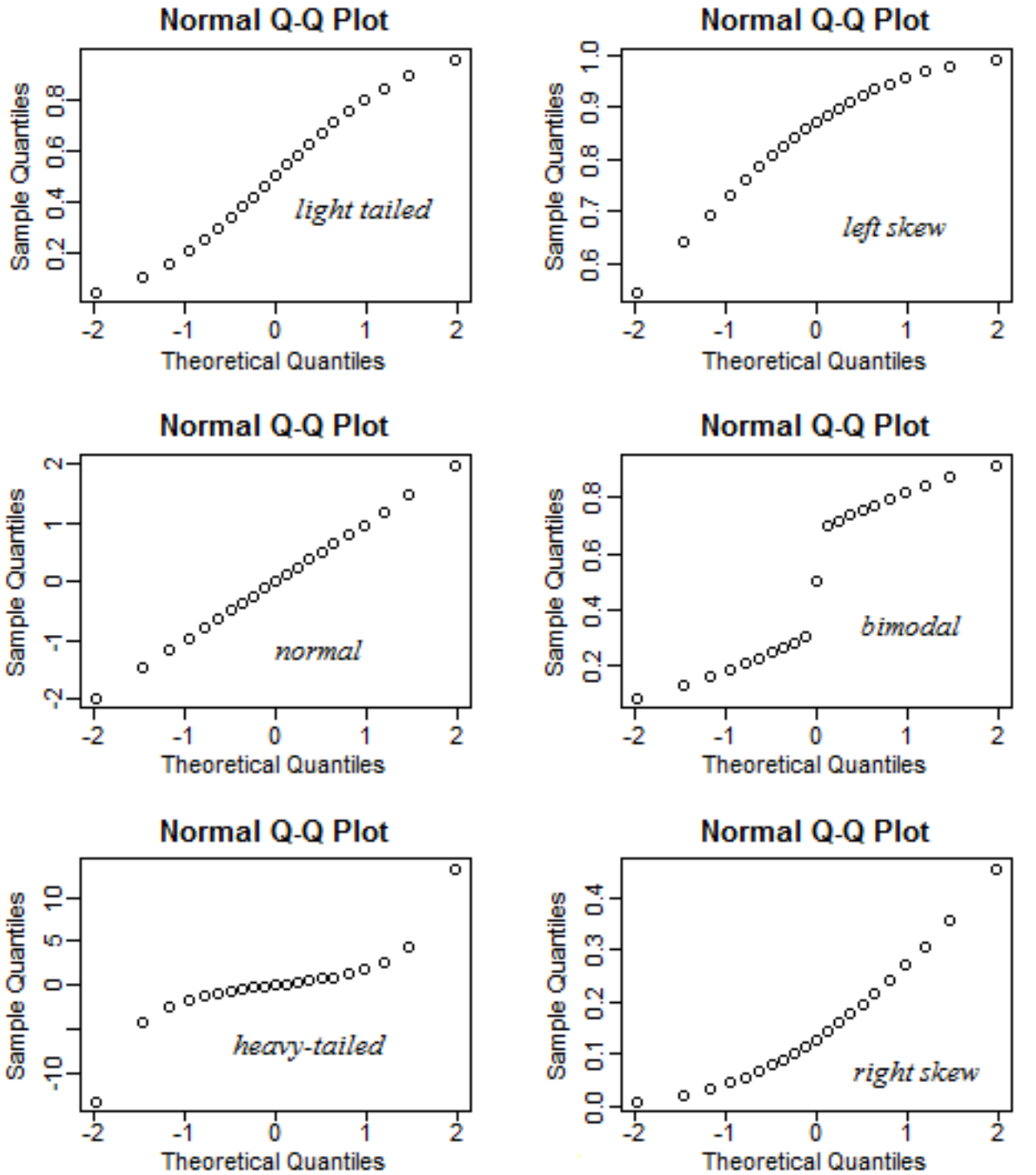


T Distribucion with 2 df

```
set.seed(100)
x<-rt(100,df=2)
qqnorm(x)
qqline(x)
```



QQ-plot

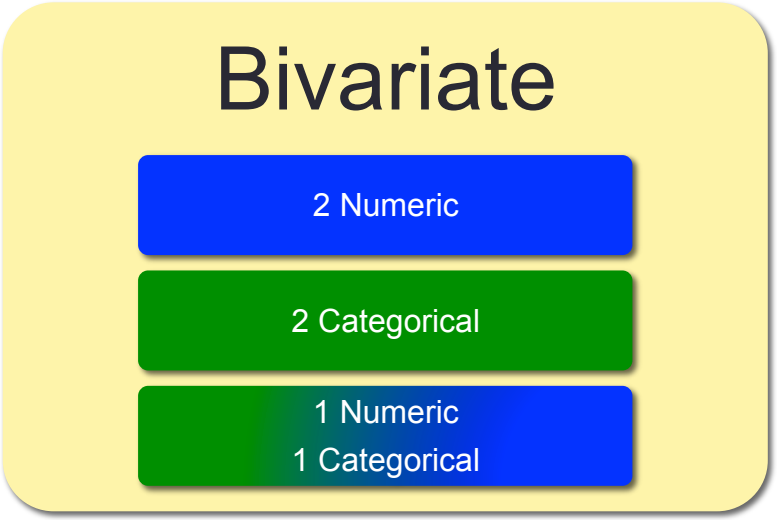


Source: https://stats.stackexchange.com/questions/101274/how-to-interpret-a-qq-plot

Bivariate Quantitative

Bivariate analysis include:

- Crosstabs
- Covariance
- Correlation



Advanced techniques include:

- Cluster analysis
- Analysis of variance (ANOVA)
- Factor analysis
- Principal component analysis (PCA)

Bivariate Quantitative: Contingency tables

Two categorical variables

- **Contingency tables** provide a way to display the frequencies and relative frequencies of observations classified according to two categorical variables.
- The elements of one category are displayed across the columns; the elements of the other category are displayed over the rows.

- The basic table types supported by crosstab() are:
  - *f*requency - frequency count
  - *r*ow.pct - proportion within row
  - *c*ol.pct - proportion within column
  - *j*oint.pct - proportion within final 2 dimensions of table
  - *t*otal.pct - proportion of entire table

Bivariate Quantitative: Contingency tables

```
library(gmodels)
data(infert, package = "datasets")
CrossTable(infert$education, infert$induced, prop.t=TRUE, prop.r=TRUE,
prop.c=TRUE)
```

```
 Cell Contents
|-------------------------|
|                       N |
|              Expected N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
```

infert$education	infert$induced: 0	1	2	Row Total
0-5yrs	4	2	6	12
	1.232	0.506	9.898	
	0.333	0.167	0.500	0.048
	0.028	0.029	0.162	
	0.016	0.008	0.024	
6-11yrs	78	27	15	120
	1.121	1.059	0.471	
	0.650	0.225	0.125	0.484
	0.545	0.397	0.405	
	0.315	0.109	0.060	
12+ yrs	61	39	16	116
	0.518	1.627	0.099	
	0.526	0.336	0.138	0.468
	0.427	0.574	0.432	
	0.246	0.157	0.065	
Column Total	143	68	37	248
	0.577	0.274	0.149	

Bivariate Quantitative: Covariation

Two numerical variables

- The covariance expresses how much two numeric variables "change together" and the nature of that relationship, whether it is positive or negative

- The R commands cov() is used for the sample covariance; you need only to supply the two corresponding vectors of data.

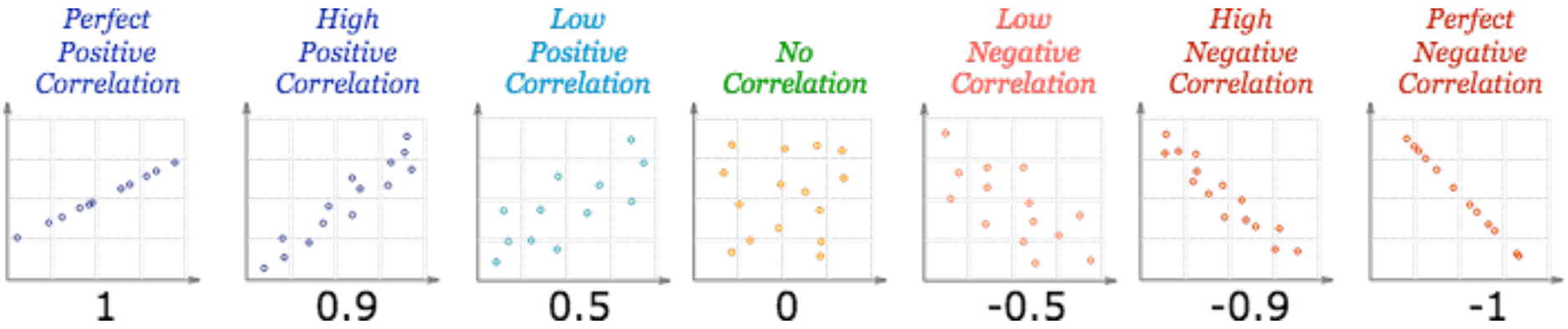
```
xdata <- c(2,4.4,3,3,2,2.2,2,4)
ydata <- c(1,4.4,1,3,2,2.2,2,7)
cov(xdata,ydata)
[1] 1.479286
```

- A positive `cov()` result shows a positive linear relationship—as x increases, y increases.
- A negative result, it shows a negative linear relationship
- A covariance = 0 indicates that there is no linear relationship

Bivariate Quantitative: Correlation

Two numerical variables

- Correlation allows you to interpret the covariance further by identifying both the direction and the strength of any association.



Technically, independence implies zero correlation, but the reverse is not necessarily true.

Source: https://www.mathsisfun.com/data/correlation.html

Bivariate Quantitative: Correlation

Two numerical variables

- `cor()` computes the correlation coefficient
- `cor.test()` test for association/correlation between paired samples. It returns both the correlation coefficient and the significance level(or p-value) of the correlation .

```
cor(x, y, method = c("pearson", "kendall", "spearman"))
cor.test(x, y, method=c("pearson", "kendall", "spearman"))
```

Bivariate Quantitative: Correlation

Two numerical variables

```
my_data <- mtcars
res <- cor.test(my_data$wt, my_data$mpg,
                method = "pearson")
```

```
data:  my_data$wt and my_data$mpg
t = -9.559, df = 30, p-value = 1.294e-10
alternative hypothesis: true correlation
is not equal to 0
95 percent confidence interval:
 -0.9338264 -0.7440872
sample estimates:
       cor
-0.8676594
```

```
# Extract the p.value
res$p.value
# Extract the correlation coefficient
res$estimate
```

In the result above :

t is the t-test statistic value (t = -9.559),
df is the degrees of freedom (df= 30),
p-value is the significance level of the t-test (p-value = 1.294e-10.
conf.int is the confidence interval of the correlation coefficient at 95% (-0.9338264 ,-0.7440872)

Bivariate Quantitative

- Especially for a categorical explanatory variable and a quantitative outcome variable, it is useful to produce a variety of univariate statistics for the quantitative variable at each level of the categorical variable

  - Comparing the means is an informal version of ANOVA.
  - Comparing medians is a robust informal version of one-way ANOVA.
  - Comparing measures of spread is a good informal test of the assumption of equal variances needed for valid analysis of variance.

Bivariate data: Graphical analyisis

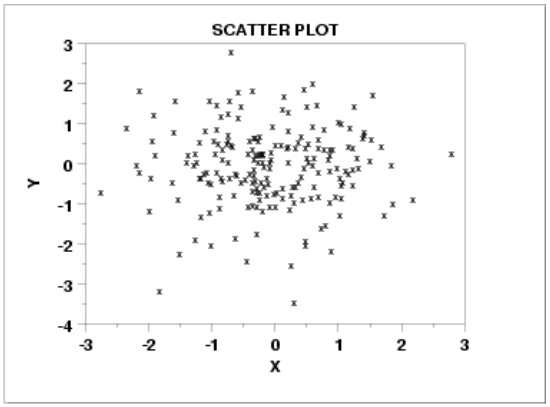
- Scatterplot
- Box plot
- Barplot

Scatter plot

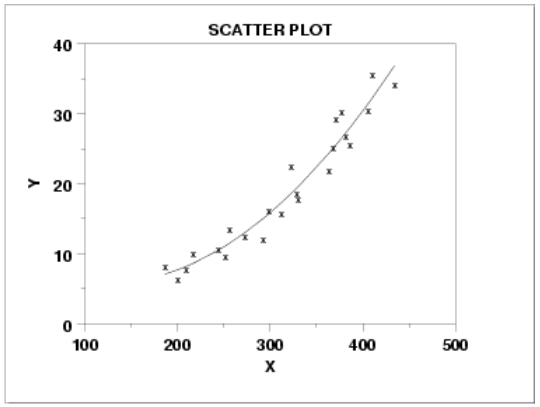
- Scatter plots reveal relationships or association between two numerical variables
- They can provide answers to the following questions:
  - Are variables X and Y related?
  - Are variables X and Y linearly related?
  - Are variables X and Y non-linearly related?
  - Does the variation in Y change depending on X?
  - Are there outliers?

Scatter plot

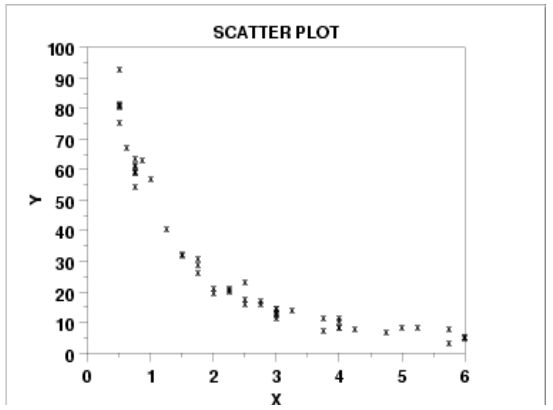
No Relationship



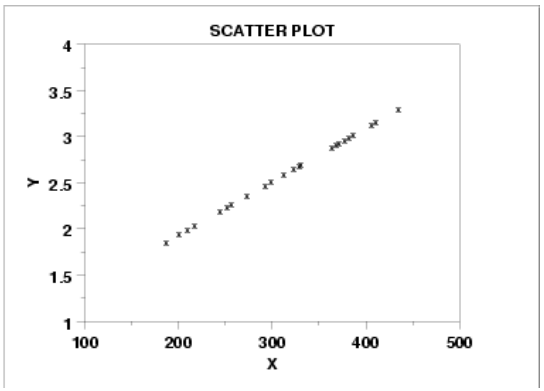
cuadratic



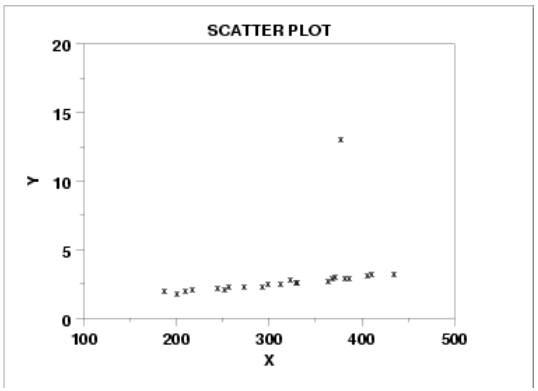
exponential



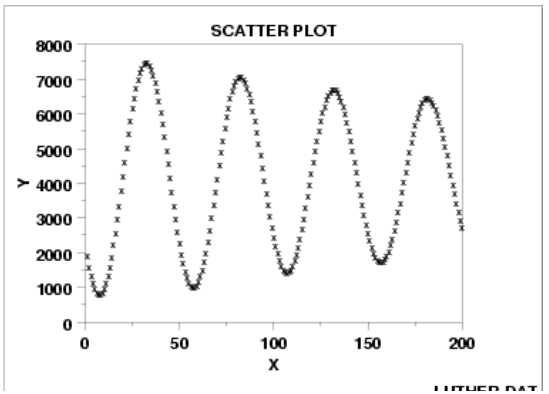
perfect lineal correlation



outliers

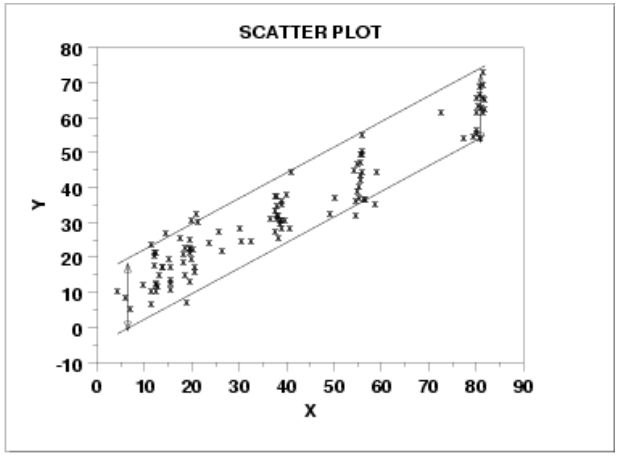


sinusoidal



Scatter plot

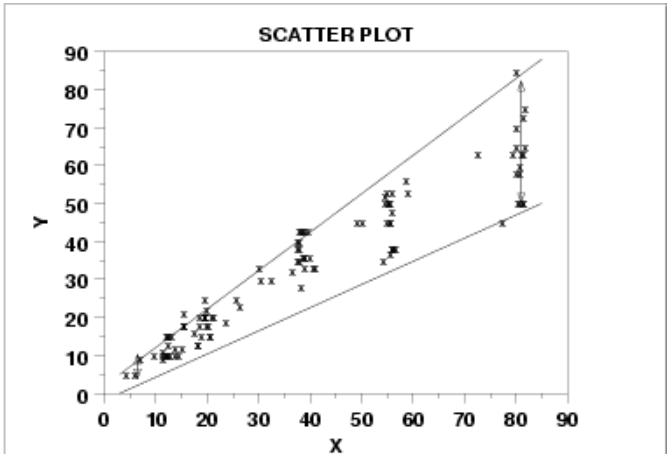
Homoscedastic



Variation of Y does Not depend on X

- Y (+- 10 units) regardless X
- Important: assumption from regression

Heteroscedastic



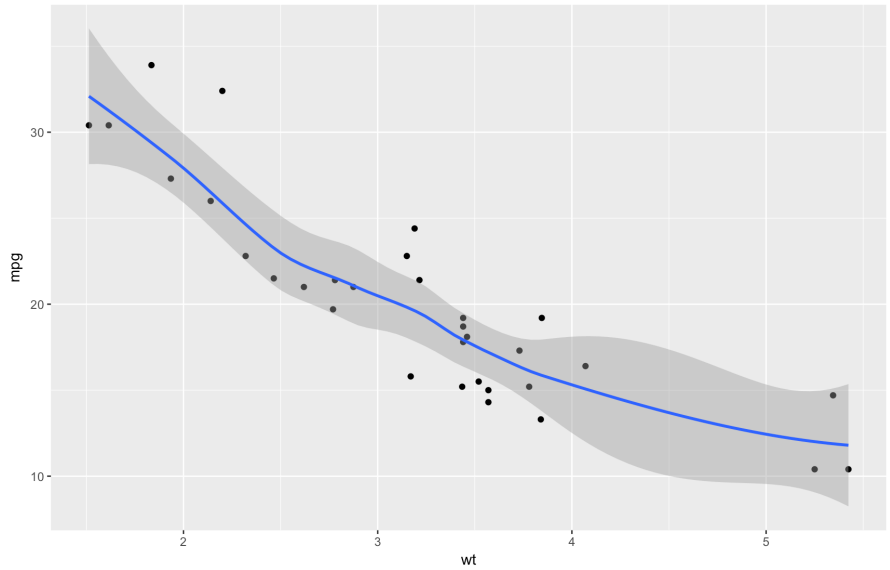
Variation of Y does depend on X

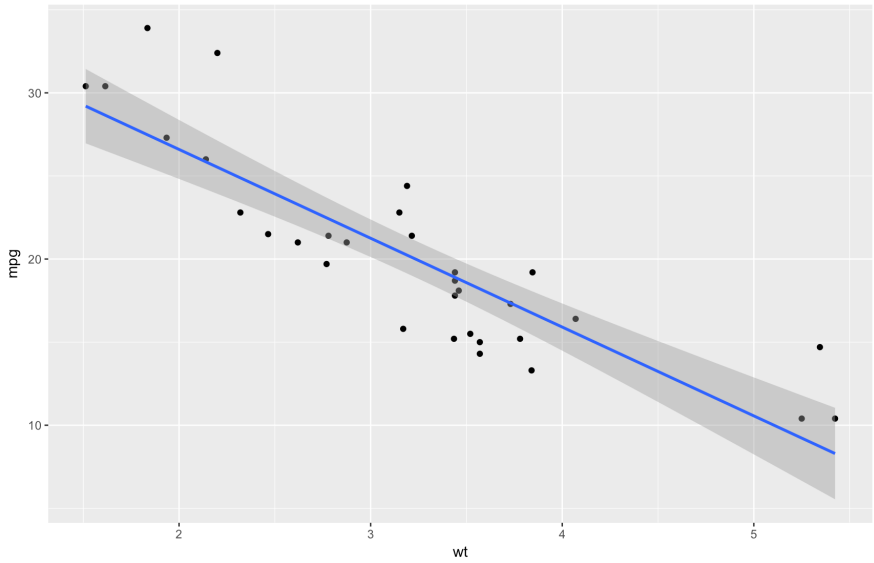
- The variation of Y is not constant,
- Implies necessity of proper weighting or Y transformation

Plotting two continuous variables: smoothers (ggplot2)

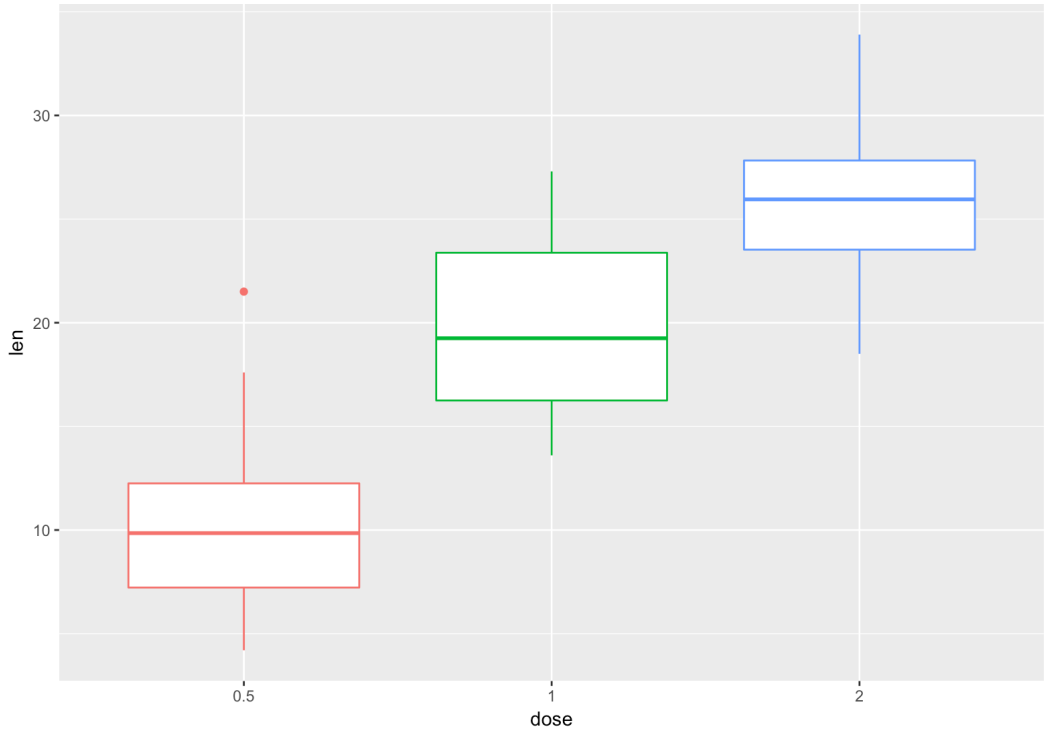
```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
# Add the regression line with
# confidence interval
ggplot(mtcars, aes(x=wt, y=mpg))
+ geom_point()
+ geom_smooth(method=lm)
```





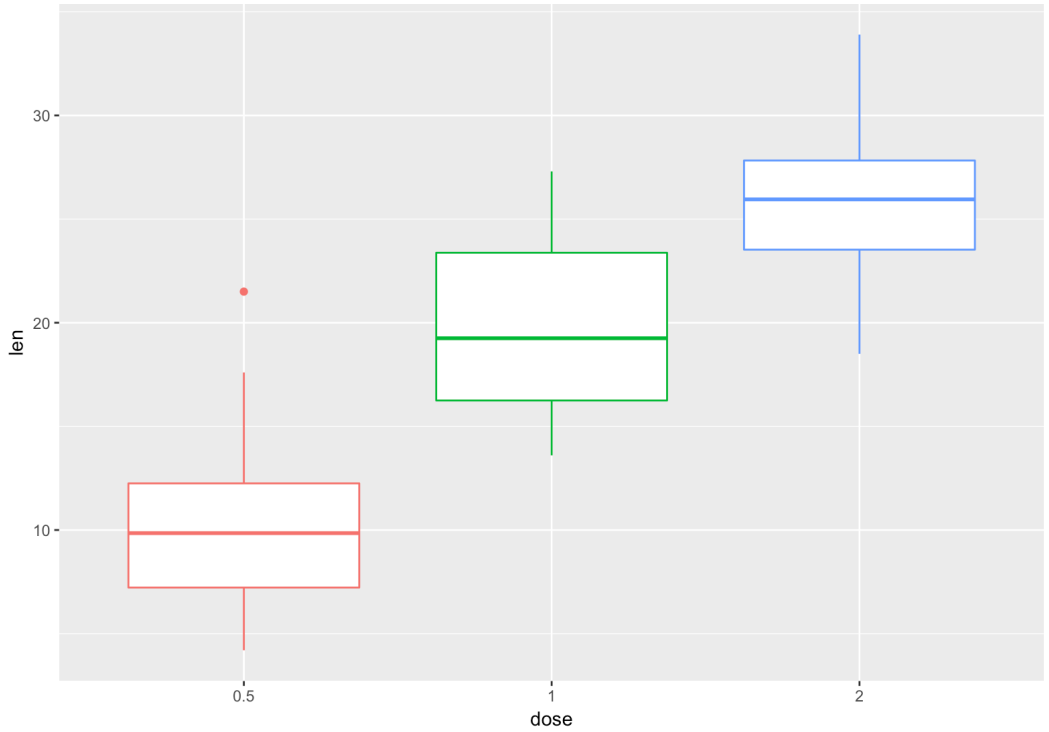
Boxplot: two variables




```
ToothGrowth$dose<-
as.factor(ToothGrowth$dose)
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

Two variables: 1 continuous, 1 ordinal => factor conversion (example)

Boxplot: two variables




```
ToothGrowth$dose<-
as.factor(ToothGrowth$dose)
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

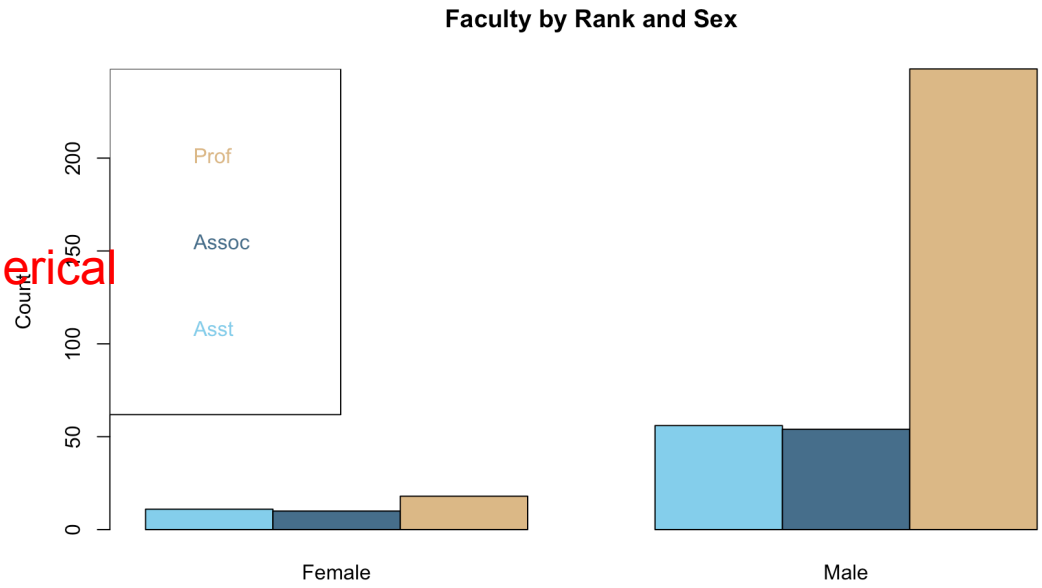
Two variables: 1 continuous, 1 ordinal => factor conversion (example)
1 numerical, 1 categorical

Bar plots, stacked plots

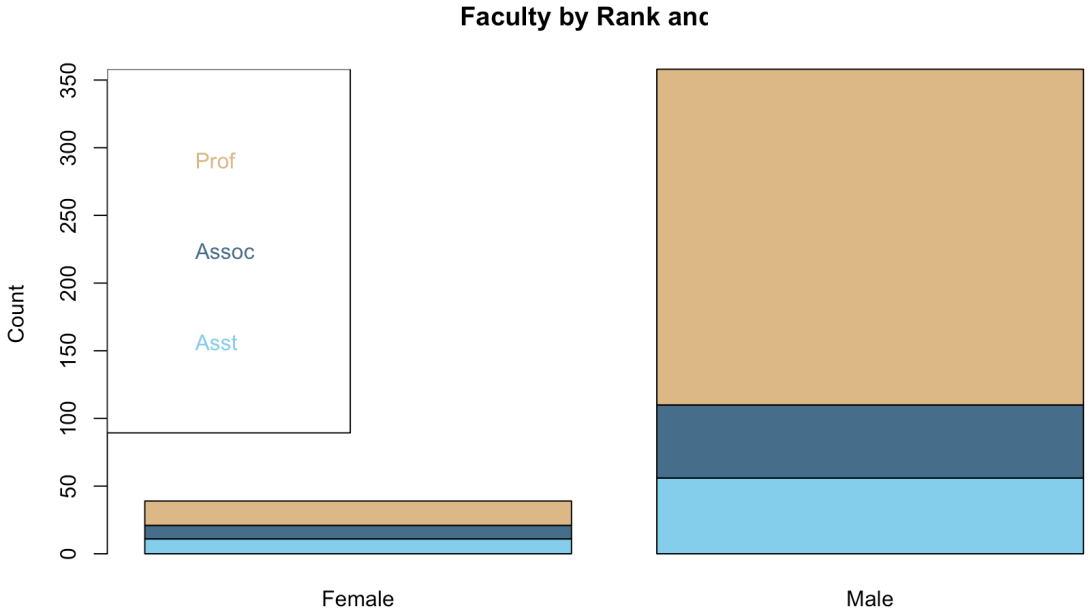
```
library(car)
attach(Salaries)
rankcount = table(rank) #get counts & save in vector rankcount
rank2 = table(rank,sex)
barplot(rank2, ylab = "Count", names.arg = c("Female","Male"),
        main = "Faculty by Rank and Sex",
        col = c("skyblue","skyblue4","burlywood"),
        sub = "c. Stacked plot")
legend("topleft", c("Prof","Assoc","Asst"),
       text.col = c("burlywood","skyblue4","skyblue"))

barplot(rank2, ylab = "Count", names.arg = c("Female","Male"),
        main = "Faculty by Rank and Sex",
        col = c("skyblue","skyblue4","burlywood"),
        sub = "d. Grouped plot", beside = T)
legend("topleft", c("Prof","Assoc","Asst"),
       text.col = c("burlywood","skyblue4","skyblue"))
```

Bar plots

Two variables categorical y numerical



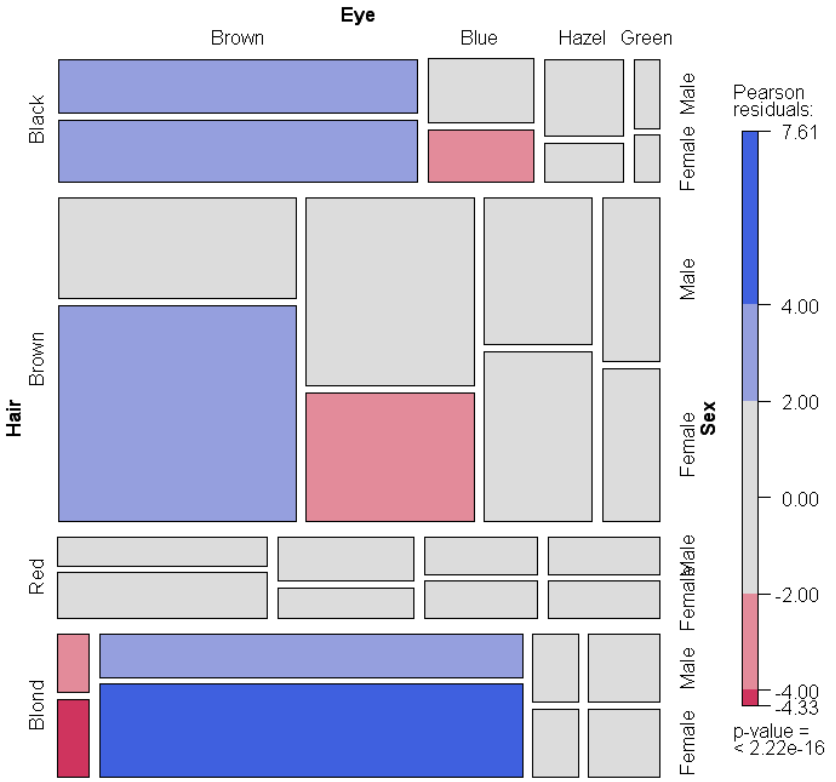
d. Grouped plot



c. Stacked plot

Visualization of two categorical data: mosaic plot

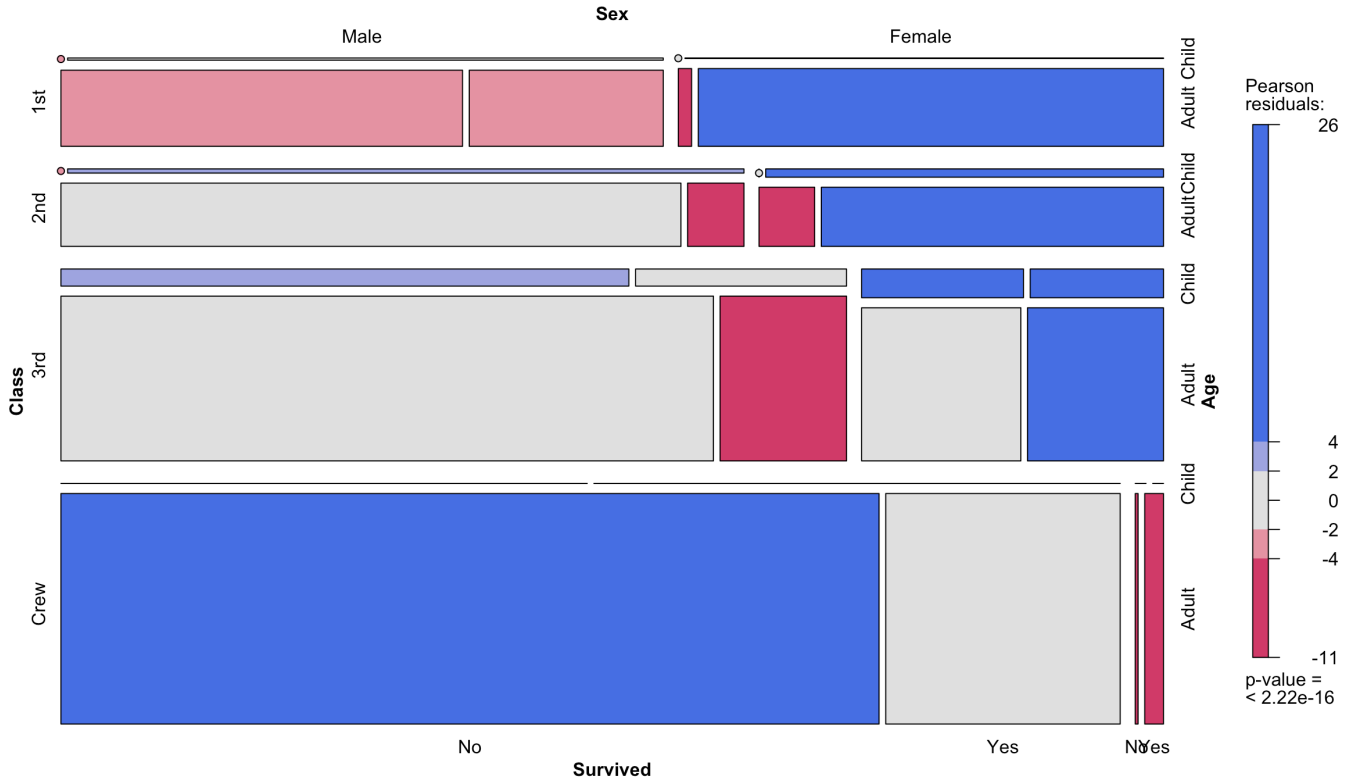
- Mosaic plots provide an ideal method both for visualizing contingency tables
- At each stage of plot creation, the rectangles are split parallel to one of the two axes.
- The important encoding is length.



Visualization of two categorical data: mosaic plot

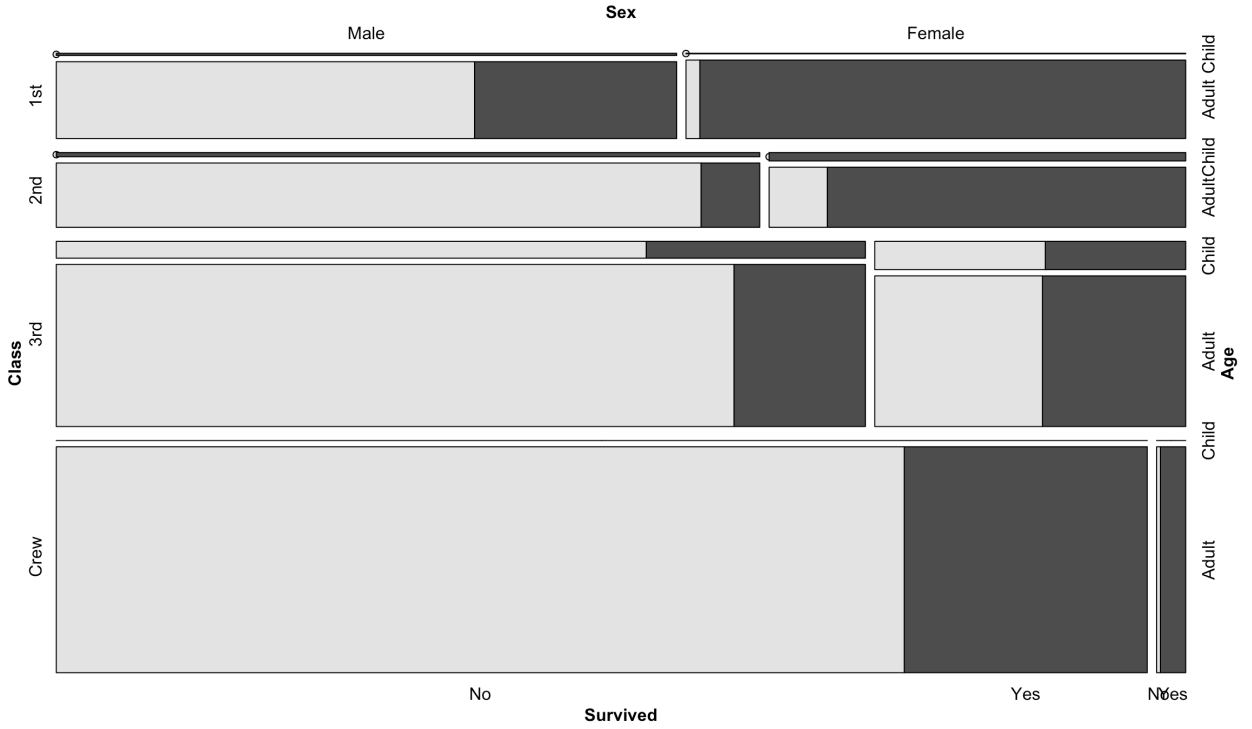
- In order to produce a mosaic plot it is necessary to have:
  - A contingency table containing the data.
  - A preferred ordering of the variables, with the “response” variable last.

```
library(vcd)
mosaic(Titanic
,shade=TRUE)
```



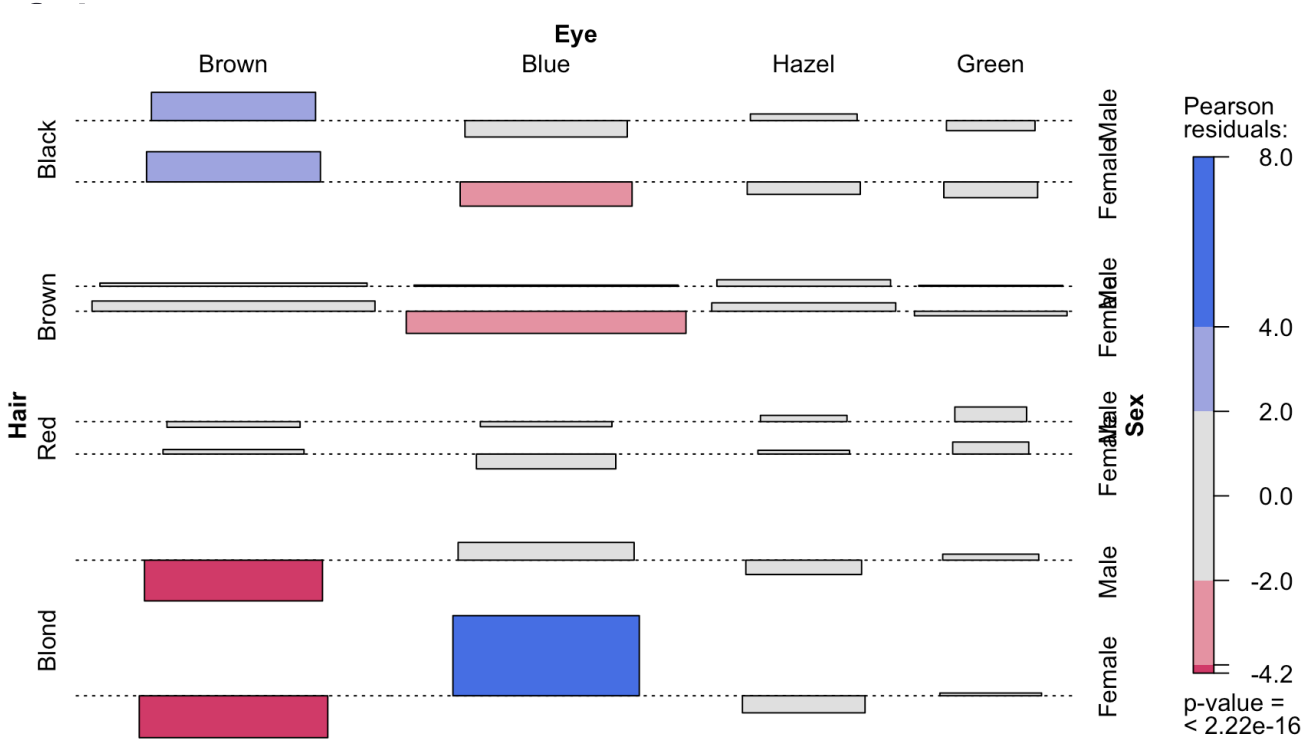
Visualization of two categorical data: mosaic plot

```
library(vcd)
## Highlighting:
mosaic(Survived ~ ., data = Titanic, shade=TRUE)
```



Visualization of two categorical data: Association plot

- Counts are represented by rectangles proportional in size to the counts combinations



Multivariate Quantitative

Bivariate analysis include:

- Crosstabs
- Covariance
- Correlation

Advanced techniques include:

- Cluster analysis
- Analysis of variance (ANOVA)
- Factor analysis
- Principal component analysis (PCA)

Multivariate Graphical

- Matrix Scatterplot
- Profile Plot
- Correlations for Multivariate Data

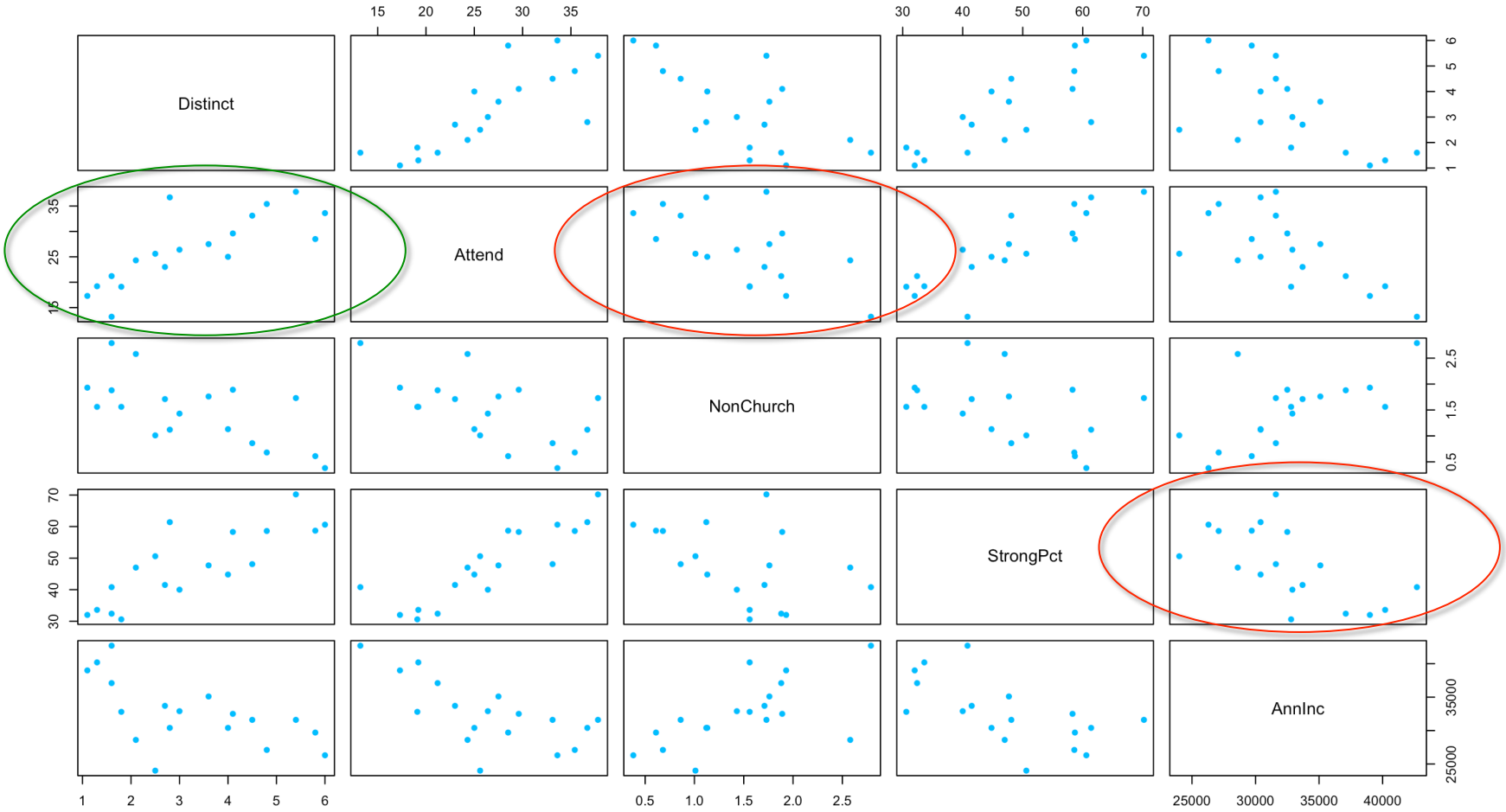
Scatterplot matrix for numerical variables

- Sometimes helps to look at the relationships of each of the possible pairs of variables first. R provides a shortcut command, pairs()

```
install.packages("Sleuth2")
library(Sleuth2)
attach(ex1713)
head(ex1713)
pairs(~ Distinct + Attend + NonChurch + StrongPct + AnnInc,
      pch = 16, col = "deepskyblue")
```

the variable names are typed as a formula, beginning with the ~ symbol

Scatterplot matrix for numerical variables: pairs()



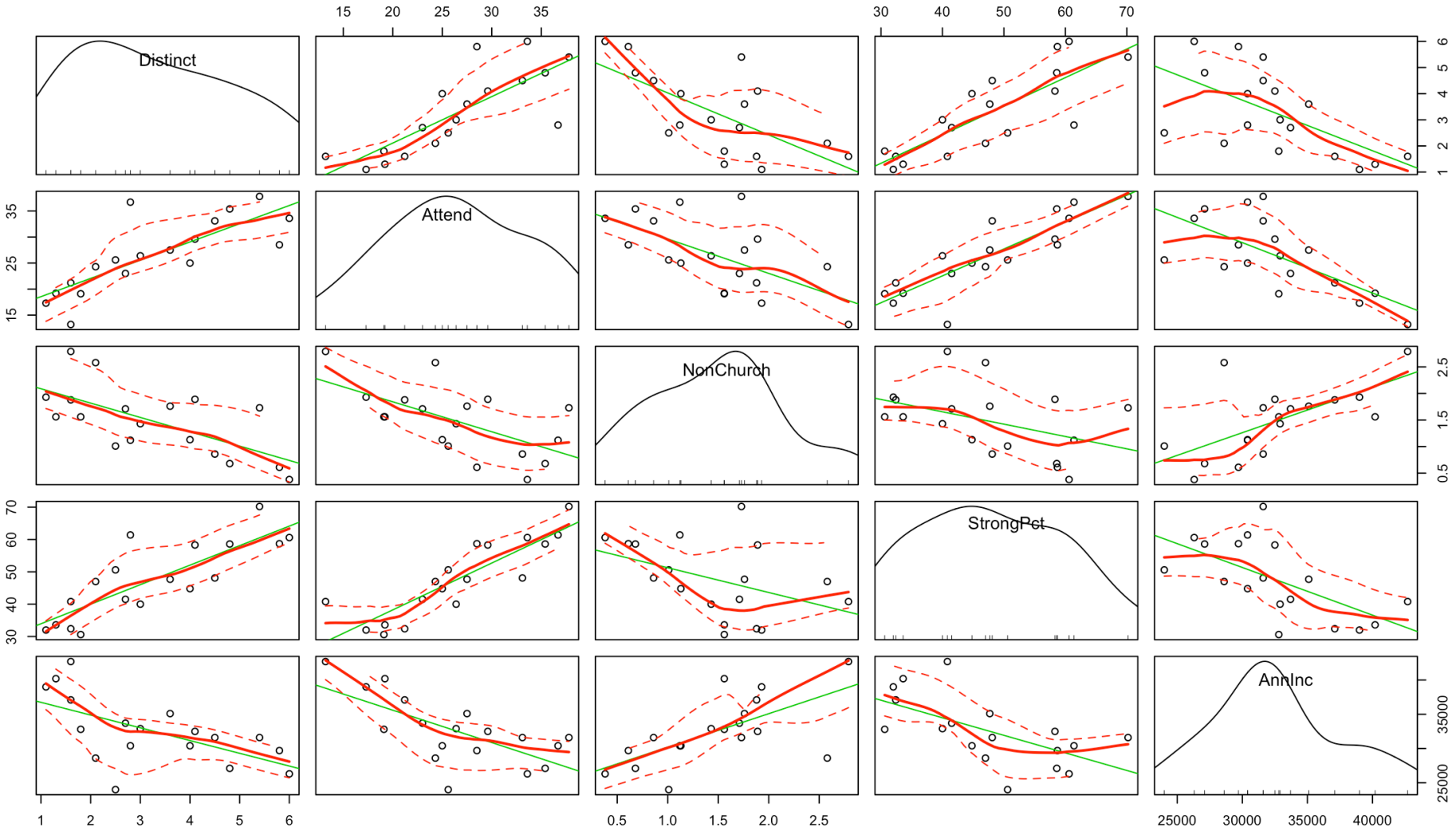
Scatterplot matrix for numerical variables: cars()

- The car package has a function called scatterplotMatrix() with useful features:

  - each of the variables can go on the diagonal of the matrix as a histogram, density plot, box plot, QQ plot, or 1D (diagonal) strip chart..
  - Add smoothers
  - easily add a least-squares line to each plot.

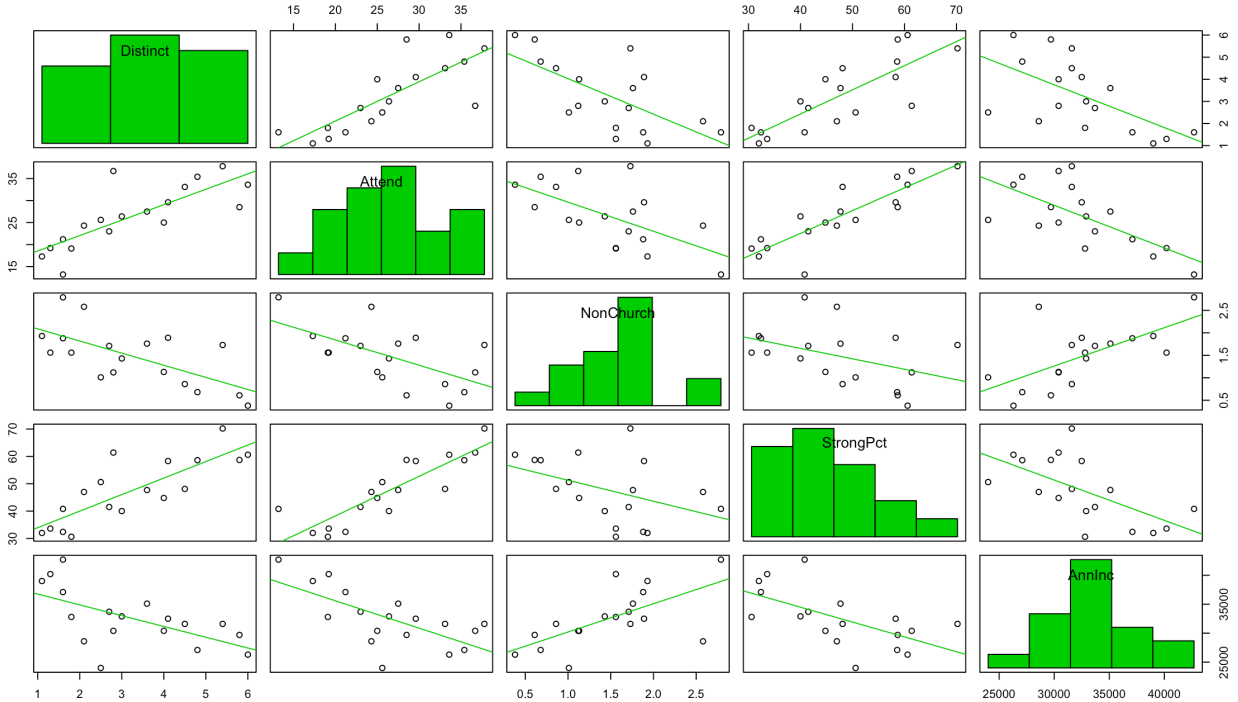
```
library(car)
scatterplotMatrix(~Distinct + Attend + NonChurch +
StrongPct + AnnInc)
```

Scatterplot matrix for numerical variables: cars()



Scatterplot matrix for numerical variables: cars()

```
library(car)
scatterplotMatrix(~Distinct + Attend + NonChurch + StrongPct
+ AnnInc, diagonal = "histogram", smoother = NULL))
```



Corrgram/Correlogram

- Type of graph related to the scatter plot matrix.
- The individual scatter plots are replaced by symbols that represent numbers measuring the amount of linear correlation between two quantitative variables.
- first necessary to make a correlation matrix by using the `cor()` function:

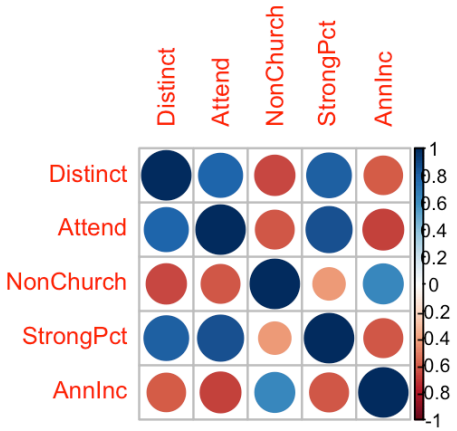
```
> install.packages("corrplot")
> library(Sleuth2)
> attach(ex1713)
> y = cor(ex1713[, 2:6]) # use all rows and columns 2-6
```

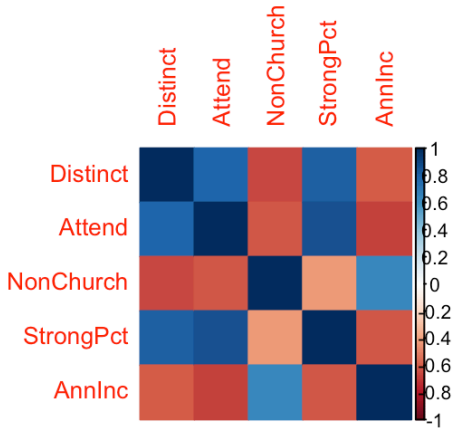
```
            Distinct     Attend  NonChurch  StrongPct     AnnInc
Distinct   1.0000000  0.7891067 -0.6585883  0.8127124 -0.6003892
Attend     0.7891067  1.0000000 -0.6107342  0.8649691 -0.6766143
NonChurch -0.6585883 -0.6107342  1.0000000 -0.4218525  0.6458747
StrongPct  0.8127124  0.8649691 -0.4218525  1.0000000 -0.6146261
AnnInc    -0.6003892 -0.6766143  0.6458747 -0.6146261  1.0000000
```

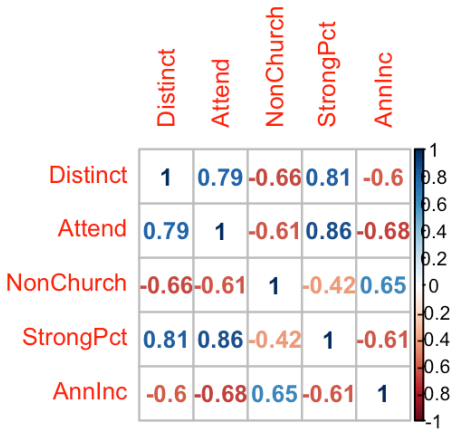
Corrgram/Correlogram

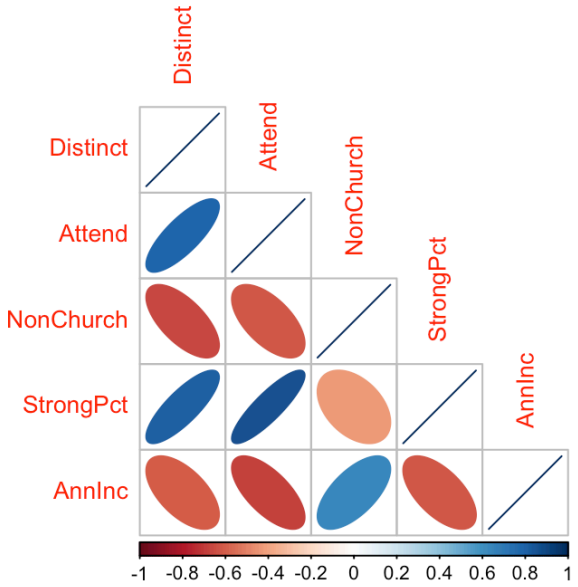
```
> install.packages("corrplot")
> library(Sleuth2)
> library("corrplot")
> attach(ex1713)
# use all rows and columns 2-6
> y = cor(ex1713[, 2:6])
> par(mfrow = c(2,2))
> corrplot(y)  # default method is "circle"
> corrplot(y, method = "color")
> corrplot(y, method = "number")
> corrplot(y, method = "ellipse", type = "lower")
```

Corrgram/Correlogram







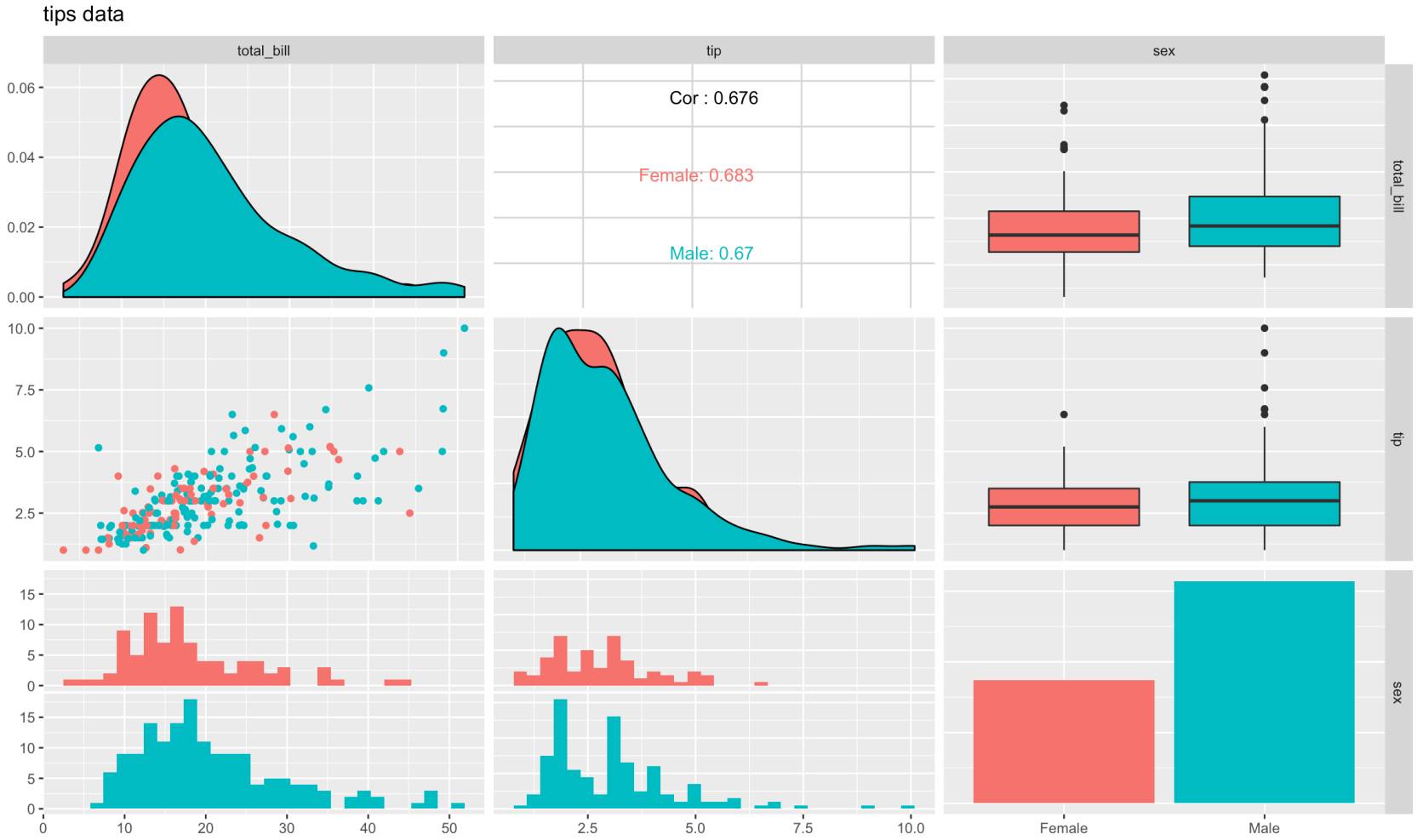


Generalized Pairs Matrix with Mixed Quantitative and Categorical Variables

- Datasets with both quantitative and categorical variables are quite common.
- it is still possible to produce a meaningful display of all the pairwise plots of variables with `ggpairs()` from the GGally package or `gpairs()` from the gpairs package.

```
library(GGally)
library(ggplot2)
data(tips, package="reshape")
ggpairs(data=tips, # data.frame with variables
        columns=1:3, # columns to plot, default to all.
        title="tips data", # title of the plot)
```

Generalized Pairs Matrix with Mixed Quantitative and Categorical Variables



Facetted graphics: Lattice vs. ggplot

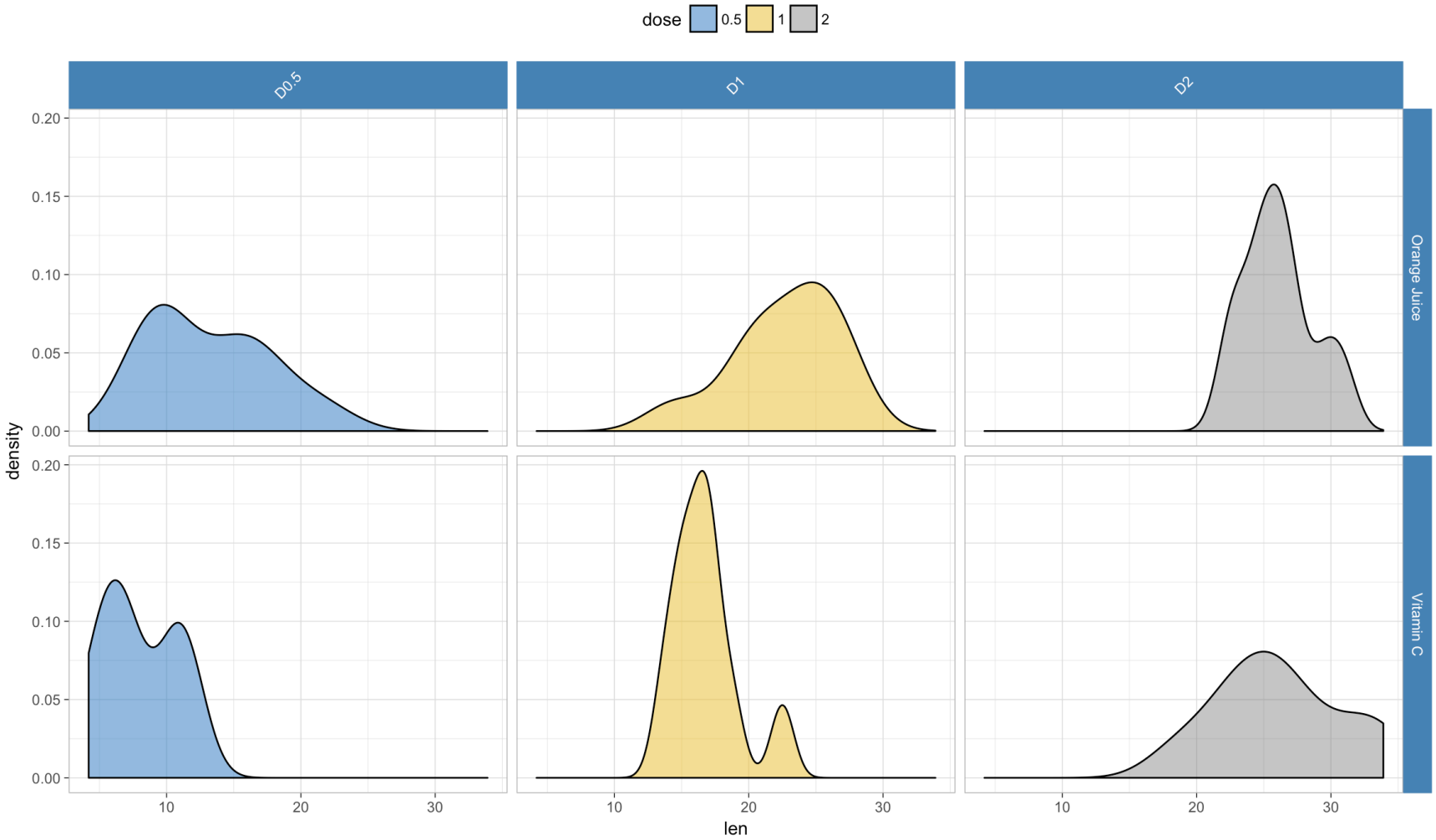
Pros for Lattice

- Intuitive structure for controlled data at a group / subgroup level
- Achieve simple panelled graphics very quickly
- Well documented
- Extensions available (latticeExtra, nlme)
- A lot faster than ggplot2

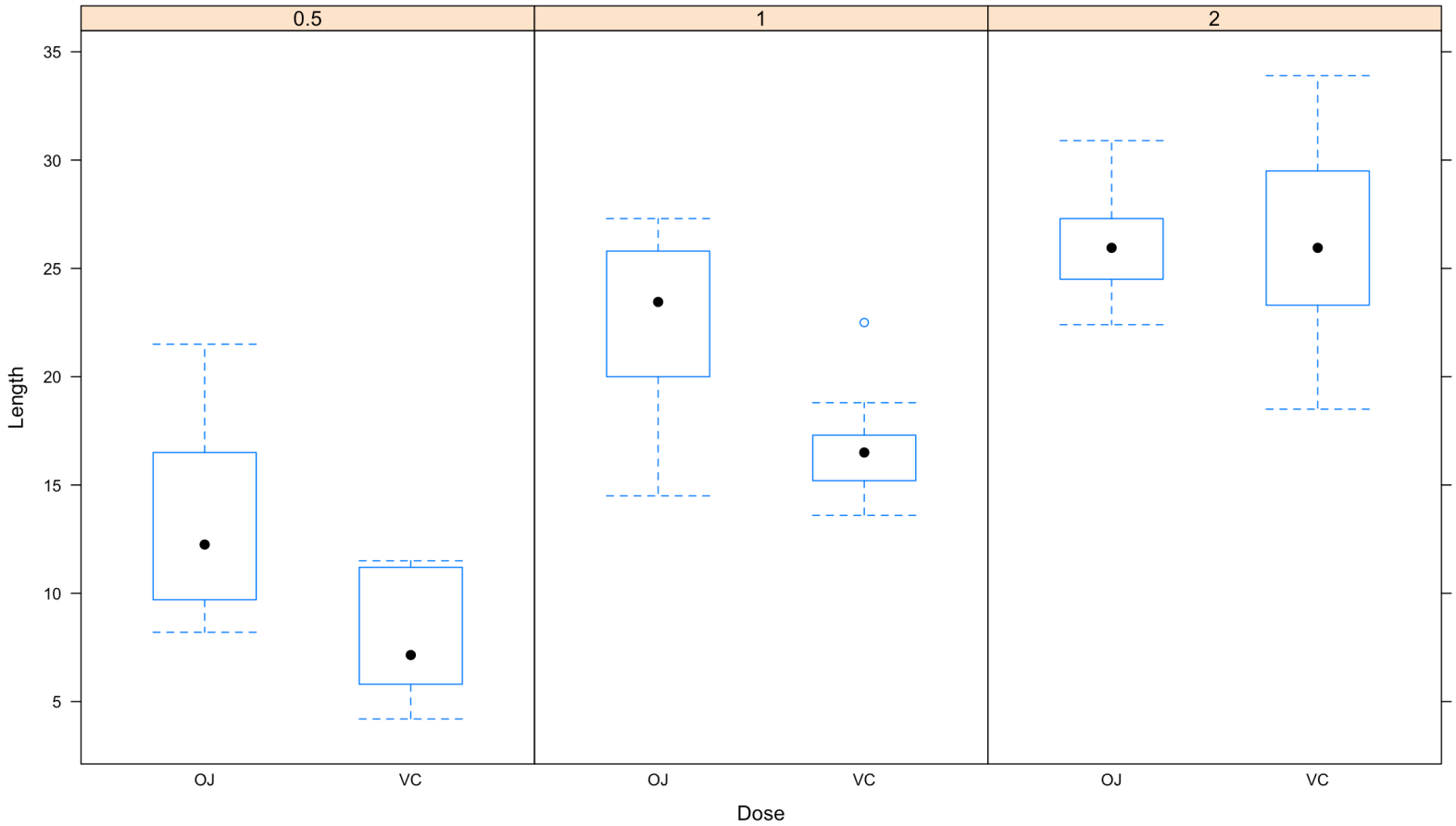
Cons for Lattice

- Default options can be frustrating
- Default styling doesn't look great
- Making good use of the panel / panel.groups structure needs lots of "function" knowledge
- Some "tricks" needed to do more than 2 levels of nested grouping

ggplot example

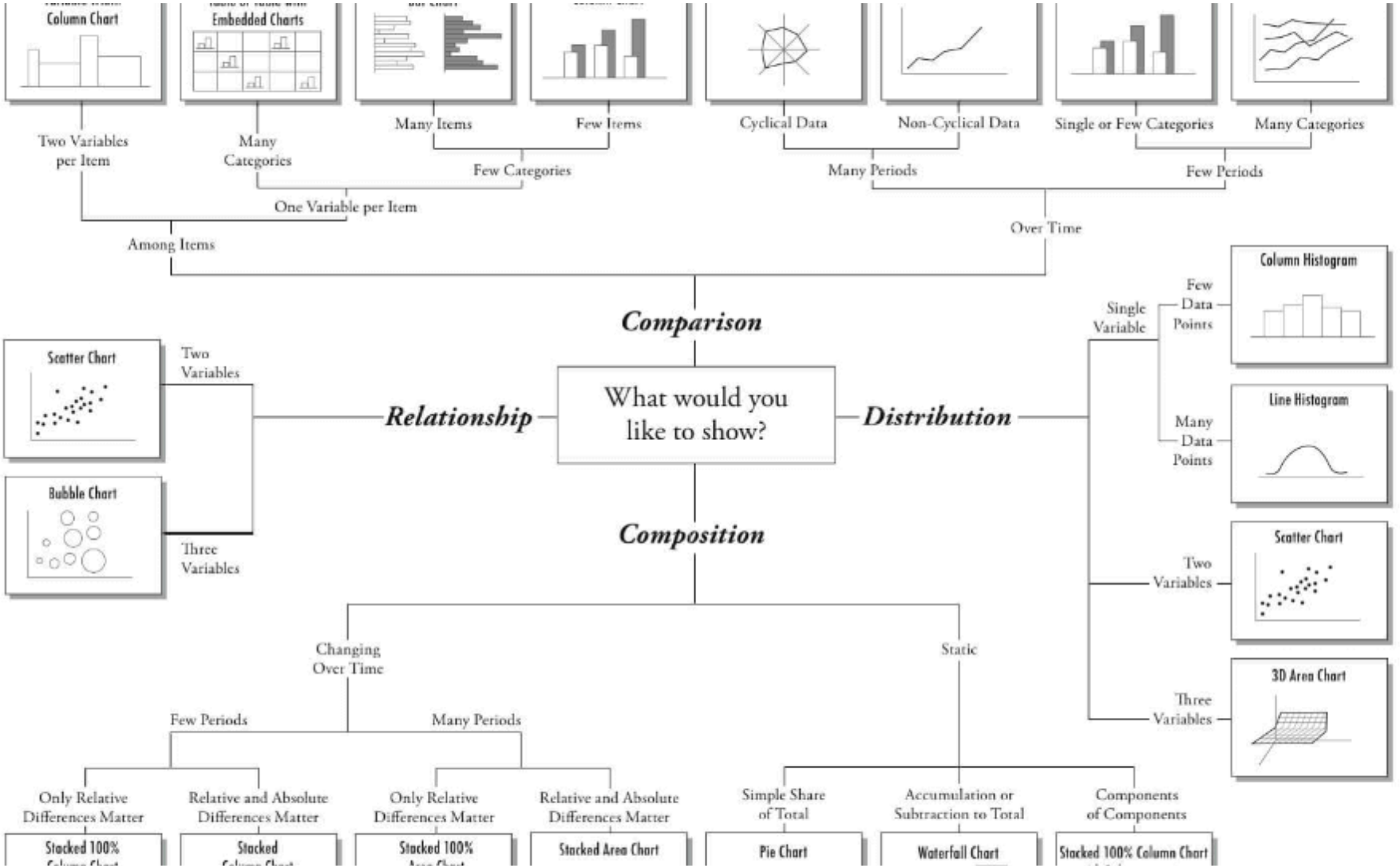


Lattice example



To determine the best graph…

- How many variables do you want to show in a single chart?
- How many data points will you display for each variable?
- Will you display values over a period of time, or among items or groups?


What would you like to show?
Comparison
Relationship
Distribution
Composition
Column Chart
Embedded Charts
Two Variables per Item
Many Categories
Many Items
Few Items
Few Categories
One Variable per Item
Among Items
Cyclical Data
Non-Cyclical Data
Many Periods
Single or Few Categories
Many Categories
Few Periods
Over Time
Scatter Chart
Bubble Chart
Two Variables
Three Variables
Single Variable
Few Data Points
Many Data Points
Column Histogram
Line Histogram
Scatter Chart
3D Area Chart
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Stacked 100%
Stacked
Stacked 100%
Stacked Area Chart
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components
Pie Chart
Waterfall Chart
Stacked 100% Column Chart