DATA EXPLORATION AND SUMMARIZATION

Applied Analytics: Frameworks and Methods 1

Outline

- Review of Basics
- Explore Data
- Descriptive Measures
- Visual Summaries

REVIEW OF BASICS

Review of Basics

- Command line operations
- Objects and Assignment
- Names
- Functions
- Data Types
- Data Structures
- R Packages

Command Line Operations

- R Console can be used to do command line operations such as arithmetic. E.g.,
  - *45 * 34*
  - *11223344 / 55667788*

Objects and Assignment

- An object could be a variable, a dataset, a paragraph of text, or a mathematical index.
- We can perform operations on objects.
- Objects have associated attributes.
- R comes with some built-in objects (e.g., pi), but most of them are created by the user

- Assignment is the process of assigning values to an object. E.g.,

  a = 4

  b = 3

  *c = a * b*

- Object names must start with a letter and can only contain letters, numbers, underscore or a period. Common naming formats are:
  - *variable_name,*
  - *variable.name*
  - *VariableName*

Functions

- R has a large collection of built-in functions that have the following syntax:
  - *function_name(arg1 = val1, arg2 = val2, …)*
- For e.g., log(x = 100, base = 10)
- Function arguments can be required or optional.
- If argument names are not stated in a function call, R will assumes argument values are provided in order.
- When using multiple functions, order of operations is innermost to outermost. Consider
- log(exp(1))
- Each function is accompanied with documentation in a standard format and can be found by running
  - *?function_name*

Data Types

- There are a number of supported data types which are referred to as a class. Frequently used classes are listed below.

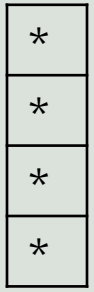
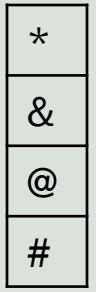
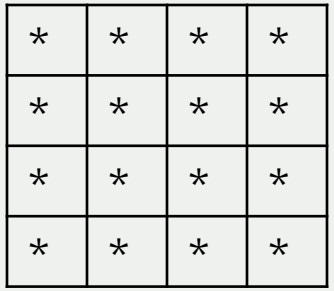
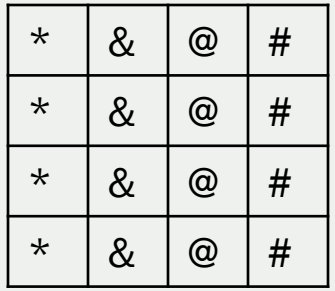
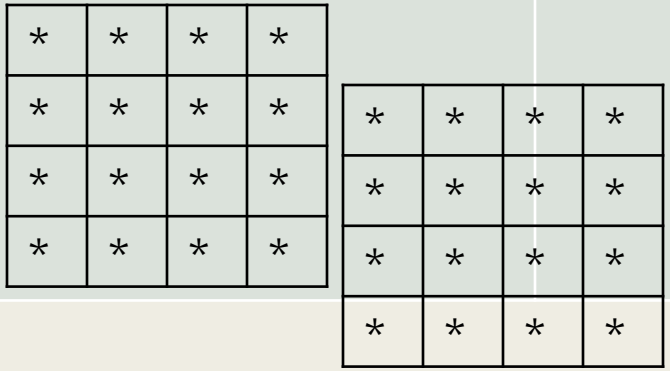
Data Type	Class in R	Class Type
Numbers	numeric, integer	Atomic classes
Text or String	character	
Binary	logical (TRUE/FALSE)	
Complex	complex	
Categorical	factor	
Date and Time	Date class	

Data Types: Change or Coerce

- To find the class of an object: class(object_name) or is.class_name(object_name)
- Class of an object may be changed either automatically to
  - *accommodate an operation (e.g., 4 + T) or*
  - *manually (as.integer(T)*

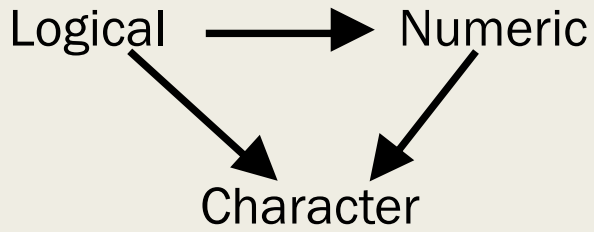
	Examples	Check	Convert to or Coerce
numeric	0, -2, 3.45, 3.1646	is.numeric	as.numeric
character	“foo”, “45”, “TRUE”	is.character	as.character
logical	TRUE, FALSE	is.logical	as.logical
complex	4 + 3i	is.complex	as.complex
factor	male male male female Levels: male female	is.factor	as.factor

Data Structures in R

	Same class	Different class
1-d	vector 	list 
2-d	matrix 	data.frame (tibble, data.table) 
n-d	array 	

Data Structures
Vector

- One-dimensional data structure where all objects belong to same class
- A few vector functions
  - *c()*
  - *is.vector()*
  - *as.vector()*
- Key features
  - *Element-wise operations by default*
  - *Recycling behavior*
- If elements in c() are of different class, all elements are coerced to one class based on the following vector coercion priority



Data Structures
Matrix

- Two- dimensional data structure where all objects belong to same class
- A few matrix functions
  - *matrix()*
  - *is.vector()*
- Just like vectors, they do
  - *Element-wise operations by default*
  - *Same coercion priority as vectors*

Data Structures
List

- Allows elements of different classes to exist
- A few list functions
  - *list()*
  - *is.list()*

Data Structures
Data Frame

- Two-dimensional data structure which resembles a spreadsheet. Elements in a column belong to the same class, but columns can belong to different classes.
- Most datasets resemble a data.frame
- Other closely related structures are tibble and data.table
- A few functions are
  - *data.frame()*
  - *head()*
  - *tail()*
  - *str()*

Data Structures
Read in a Data Frame

- To read in a data frame, you could either include entire file path or set working directory to point to location of file. Here is an illustration of the two options for a file, info.csv
- Method 1: Specify the full path of the file in `read.csv()`
  - *read.csv('c:/my_classes/best_course_ever/info.csv') # on Windows*
  - *read.csv('/my_classes/best_course_ever/info.csv') # on Mac*
- Method 2: Set the working directory to the location of the file. Then specify file name in `read.csv()`
  - *setwd('c:/my_classes/best_course_ever/') # on Windows*
  - *setwd('/my_classes/best_course_ever/') # on Mac*
  - *read.csv('info.csv')*
- To find the file path
  - *For Windows: With Shift pressed, right click on file. Select* "Copy as Path"
  - *For Mac: Right click/secondary click (two-finger click) on file and press Option. Select "Copy xxx as Pathname"*

Helper functions for Data Structures

	Create	Change to	Check	Get Names	Get Dimensions
vector	c	as.vector	is.vector	names	length
matrix	matrix	as.matrix	is.matrix	rownames, colnames	dim, nrow,ncol
array	array	as.array	is.array	dimnames	dim
list	list	as.list	is.list	names	length
data.frame	data.frame	as.data.frame	is.data.frame	names	dim, nrow, ncol

Packages

- The R System involves a Base System and a set of Packages.
- A Package is a collection of code and functions written for the R language
- Usually focuses on a specific task or problem. Implementation is described in the package documentation available on CRAN.
- By only installing packages that one needs, R can be kept light.
- Most useful R applications appear in packages
- To use a package, install it once. Then call the package for each R session. E.g.,

  To install ggplot2

  ```
  install.packages("ggplot2")
  ```

  To load ggplot2 into current R session

  ```
  library(ggplot2)
  ```

Illustration

See Basics.html

EXPLORE

Explore

- head(), tail(): Show first six rows or last six rows
- str(): Structure of the dataset provides basic information about number of rows and columns and class of variables.
- nrow(): Number of rows
- ncol(): Number of columns
- dim(): Number of rows and columns
- names(): Variable names

Subsetting

- A common task in data analysis is to extract a subset the data.
- Data can be extracted by using their index or by conditionals (called logical subsetting).
- But, before that, let us see what an index is.

Index

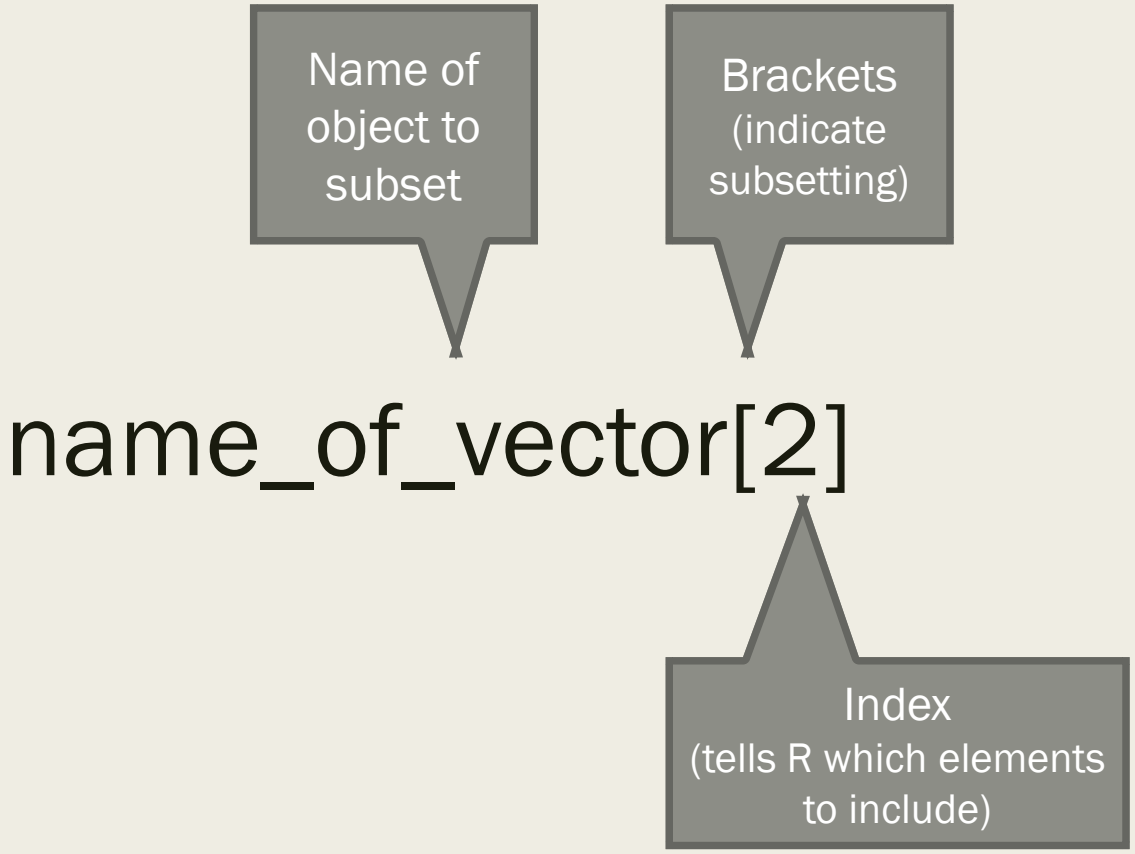
Vector

1	2	3	4	5

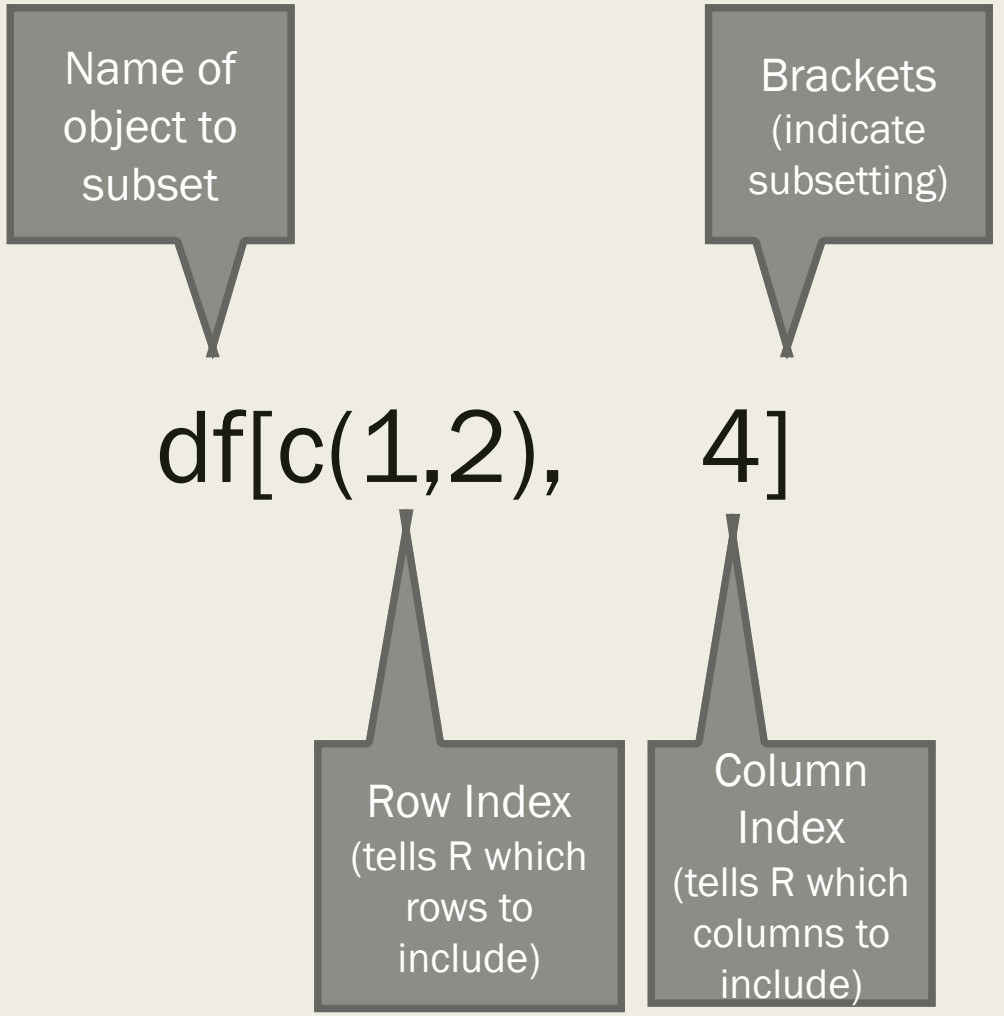
Matrix or Data Frame

1,1	1,2	1,3	1,4
2,1			
		3,3	
			5,4

Subset Notation for Vector



Subset Notation for Data Frame



Index for Subsetting

Type	Description	Example
Integers	Positive: returns specified elements Negative: returns everything but specified elements	df[2,3]
Blank spaces	Returns everything	df[2,] df[,2]
Names	Returns elements or columns with specified names	df[,'var_name']
Logicals	Returns elements that correspond to TRUE	df[,c(F,F,F,T)]
$	Returns a column of a data frame or matrix (as a vector)	df$income

Note: Another way of subsetting from a data frame is using the subset function. E.g. subset(df, var_name=="value")

Subsetting: Exercise

For a data frame, team_salary with 30 rows and 4 columns, spot the error

- team_salary(24,4)
- team_salary[10,5]
- team_salary[“10:14”,]
- team_salary[2,3,4]
- team_salary[4, - c(1,5)]
- team_salary[-c(0:17,19:30), -c(1,3)]
- team_salary[,c(True,False,False,True)]

Logical Tests

- Analysts are generally interested in extracting a subset of the data based on certain selection criteria (e.g., business customers in West Region spending more than $10,000 a month).
- Index-based subsetting is unable to effectively perform SQL like queries.
- To do this, we pass the results of a logical test to a subsetting operation.

Logical Tests

- Uses a relational operator to generate a logical (i.e., where all the elements are TRUE or FALSE).
- For an element, what do you expect from the following?

  4 == 5

  4 > 5

  4 != 5

- Now, for a vector, what do you expect?

  1:5 == 5

  1:5 > 5

  1:5 != 5

Logical Tests

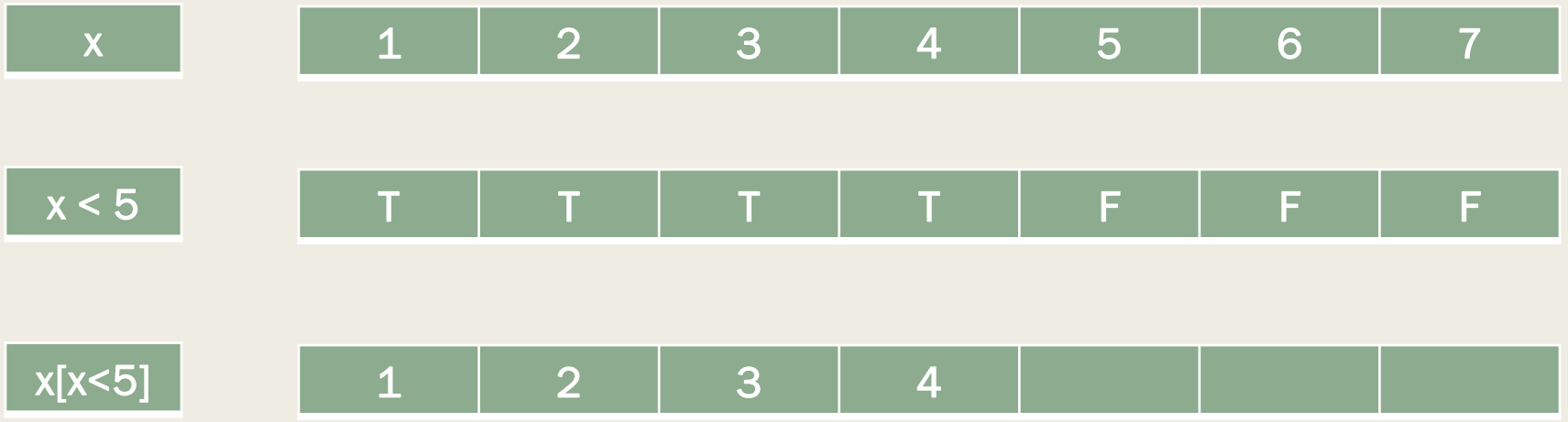
Relational operators are

- Used to compare two R objects
- Comparison yields a logical (so long as the comparison is possible)
- Relational operators are not limited to just integer and numeric classes

c(0, 1, 0, 1) == T

c('Vishal', 'Rohan', 'Nikhil') == 'Nikhil'

Logical Tests - Mechanism



Logical Tests

Smart use of logical tests can make finding a needle in a large dataset easy. But before that, consider the following.

x = 1:7

How many elements in x are …	Logical tests	Coerce logical to numeric
… less than 5	x < 5	sum(x<5)
… less than or equal to 7	x <= 5	sum(x<=5)
… greater than 5	x > 5	sum(x>5)
…. greater than or equal to 5	x >= 5	sum(x>=5)
… equal to 5	x == 5	sum(x==5
… not equal to 5	x != 5	sum(x!=5)

Logical Tests

x = 1:7

How many elements in x are …	Logical tests	Coerce logical to numeric
… less than 5 and greater than 3	x<5 & x>3	sum(x < 5 & x >3)
… less than 2 or greater than 5	x<2 | x>5	sum(x<2 | x>5)
… greater than 5	x > 5	sum(x>5)
…. greater than or equal to 5	x >= 5	sum(x>=5)
… equal to 5	x == 5	sum(x==5
… not equal to 5	x != 5	sum(x!=5)

Logical Tests

x = 1:7

	Logical tests	Coerce logical to numeric
Is 4 present in x	4 %in% x	sum(4 %in% x)
Is 3 present in x	3 %in% x	sum(3 %in% x)
Is 3 present in x; Is 4 present in x	c(3,4) %in% x	sum(c(3,4) %in% x)

Summary of Logical Operators

Operator	Tests
x > y	Is x greater than y
x < y	Is x less than y
x == y	Is x equal to y
x != y	Is x not equal to y
x %in% y	Is x in y
a & b	Both a and b are TRUE
a | b	At least one of a and b is TRUE
xor(a,b)	A is true or b is true but not both
!(a)	Not a
any(a,b,c)	At least one of a, b, c is TRUE
all(a,b,c)	Each of a, b, and c is TRUE

Subsetting using dplyr

- Over the last couple of years, dplyr has become a popular package for subsetting and transforming data.
- library(dplyr) offers two functions for subsetting the data
  - *Select Columns: select()*
  - *Select Rows: filter()*
    - Logical tests can be passed to filter()

Subsetting a List

- Subsetting a list is a bit complicated in part because of its flexible structure.
- List consists of multiple types of classes making it a flexible data storage structure
- Consider the following list

  lst = list(c(1,2),TRUE,c("a","b","c"))
- What will subsetting like before do here?

  lst[1]
- Now, try adding the numbers subsetted using the sum function

  sum(lst[1])

Subsetting a List

- Consider the following metaphor to better understand subsetting from a list
- Consider a train with a number of cars and each car carrying boxes. To call any given box, one has to first address the car and then the box.
- So, to call Box3 in Car5, the code would be:

```
Train[[Car5]][Box3]
```

Subsetting a List

- Single bracket subset always gives back object of same class

  lst[1]

- But, most functions don't work on lists. A work around is to use a double bracket.

  lst[[1]]

  sum(lst[[1]])

- To subset from a list, you have to use a double bracket [[ ]]. What will each of the following return?

  lst[[1]]

  lst[[1]][1]

Illustration

See Explore.html

DESCRIPTIVE MEASURES

Types of Variables

- Nature of the variable dictates the type of analysis
- Metric or numeric: Meaningful arithmetic can be performed
  - *Discrete (e.g., number of children)*
  - *Continuous (e.g., income)*
- Non-metric or categorical (one that is not numeric)
  - *Nominal: Unordered categories (e.g., gender)*
  - *Ordinal: Ordered categories (e.g., rank)*
  - *Dummy: Categorical variables with more than two levels are often dummy coded to simplify analysis. A dummy variable is binary.*
- Date or Time

Variable Type	Class in R
Discrete	integer
Continuous	numeric
Nominal	factor (unordered)
Ordinal	factor (ordered)
Date, Time	Date, POSIXct, POSIXt

Types of Data

- Cross-sectional: Data on a cross-section of the population at a distinct point in time.
- Time-series or longitudinal: Date gathered over time.

Descriptive Measures

- Eyeballing even a moderate sized dataset is unlikely to yield useful insights
- Numerical summaries can yield insights independent of the actual size of the data

Descriptive Measures
For Numeric Variables

- Measures of central tendency
  - *Mean: Average of all values*
  - *Median: Middle observation for sorted data*
  - *Mode: Most frequent value*
- Measures of Dispersion
  - *Range: Difference between maximum and minimum*
  - *pth Percentile: Value such that p% values are below it*
  - *Interquartile range: Difference between values at 25^{th} and 75^{th} percentiles*
  - *Variance: Average of squared deviations from mean*
  - *Standard deviation: Square root of variance*

Descriptive Measures
For Categorical Variables

- Generally involve a cross tabulation to compare categories by
  - *Frequency or Count (e.g., count of males vs. females), or*
  - *Descriptive measure of a numerical variable (e.g., median income of males vs. females)*

Illustration

See DescriptiveMeasures.html

VISUAL SUMMARY

Data Visualization

- Early in the analysis pipeline, visualizing data can aid in
  - *forming an understanding of the data,*
  - *identifying trends, and*
  - *spotting anomalies.*
- Once the analysis has been completed, a good visualization can
  - *get attention,*
  - *keep interest and*
  - *effectively convey analysis insights to stakeholders.*
- Our focus will be on visualizing data early in the analysis process. Accordingly, we will focus on a wide variety of charts but not worry much about presentation.

Visual Summary Benefits

- Visual summaries complement descriptive summaries by
  - *Reinforcing results of descriptive summaries*
  - *Making it possible to inspect a large amount of data quickly*
  - *Helping identify phenomenon overlooked by descriptives as seen in Anscombe's Quartet*

Anscombe's Quartet

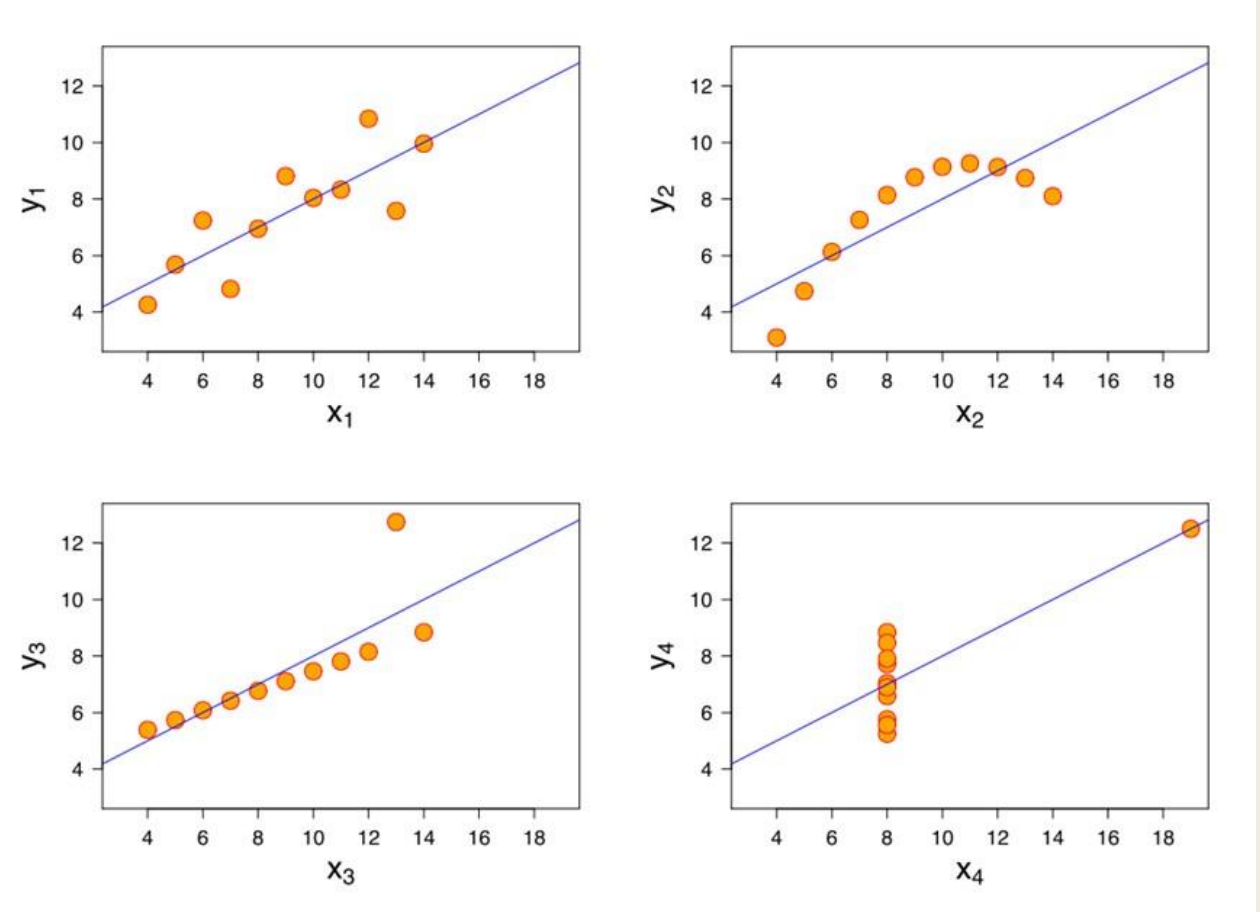
I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Source: Wikipedia

Anscombe's Quartet

Property	Value
Mean of x in each case	9 (exact)
Sample variance of x in each case	11 (exact)
Mean of y in each case	7.50 (to 2 decimal places)
Sample variance of y in each case	4.122 or 4.127 (to 3 decimal places)
Correlation between x and y in each case	0.816 (to 3 decimal places)
Linear regression line in each case	$y = 3.00 + 0.500x$ (to 2 and 3 decimal places, respectively)

Anscombe's Quartet



Visualizing Data in R

- Visualization Packages
  - *graphics is the oldest and since it usually loads automatically is also known as Base Graphics.*
  - *lattice makes better looking charts with less code, through the use of good default arguments.*
  - *ggplot2 attempts to integrate the benefits of `graphics` and `lattice` with a layered approach to generating plots. ggplot2 implements the grammar of graphics, a coherent system for describing and building graphs*
  - *And many other packages*

ggplot2

- Main components of ggplot2
  - *Data*
  - *Aesthetic mapping (aes)*
    - Describes how variables are mapped onto graphical attributes
    - Visual attribute of data including x-y axes, color, fill, shape, and alpha.
  - *Geometric objects (geom)*
    - Determines how values are rendered graphically, as bars (geom_bar), scatterplot (geom_point), line (geom_line), etc.
- Template for graphic in ggplot2 is

  ggplot(data = <Enter Data Here>, mapping = aes(<Enter Aesthetic(s) here>))+

  <Enter geom function here> +

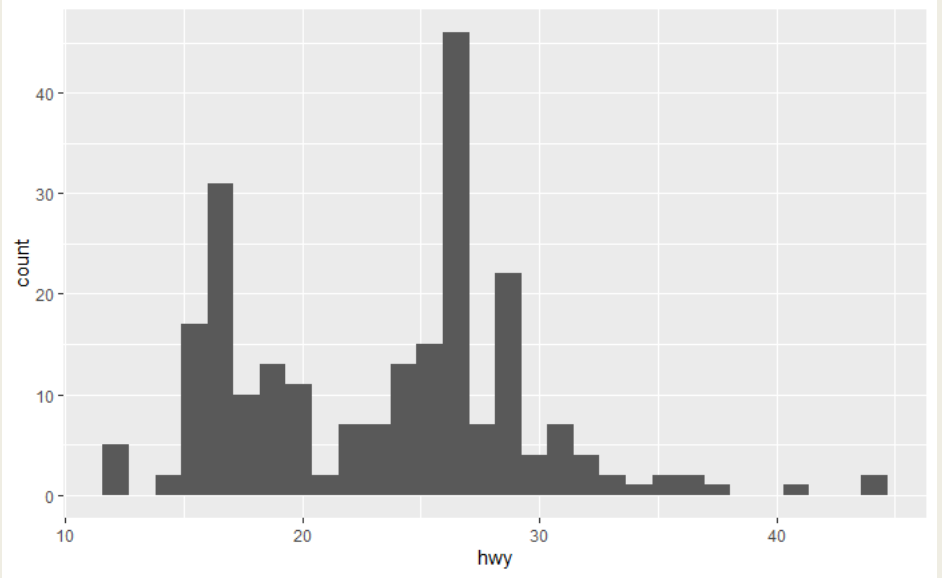
  .....

Visual Summary

- We will look at charts to
  - *examine distribution of a variable, examine variance, and spot outliers.*
  - *exploring relationships between variables.*
  - *expand our view through charts for multiple variables and multiple charts.*

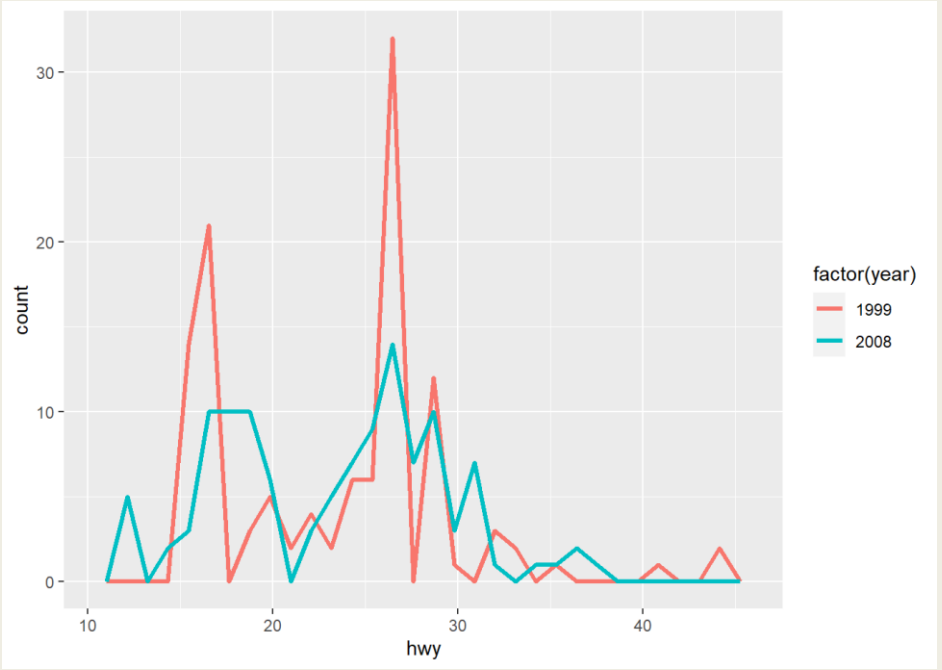
Histogram

- Most common chart for exploring the distribution of a numerical variable.
- Generated by binning the variable into discrete categories and constructing a bar chart of counts of various categories.
- Shape of the distribution may reveal
  - *a distribution that is symmetric or skewed,*
  - *distribution that is flat or peaked, or*
  - *presence of outliers.*



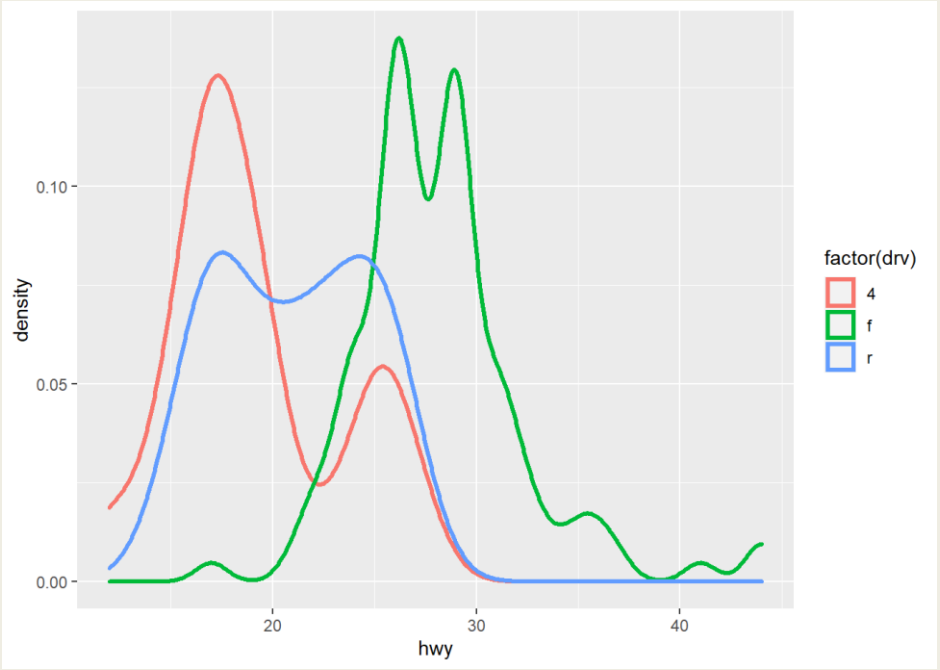
Frequency Polygon

- Outline of a histogram.
- Overcomes the problem of overlaid histograms



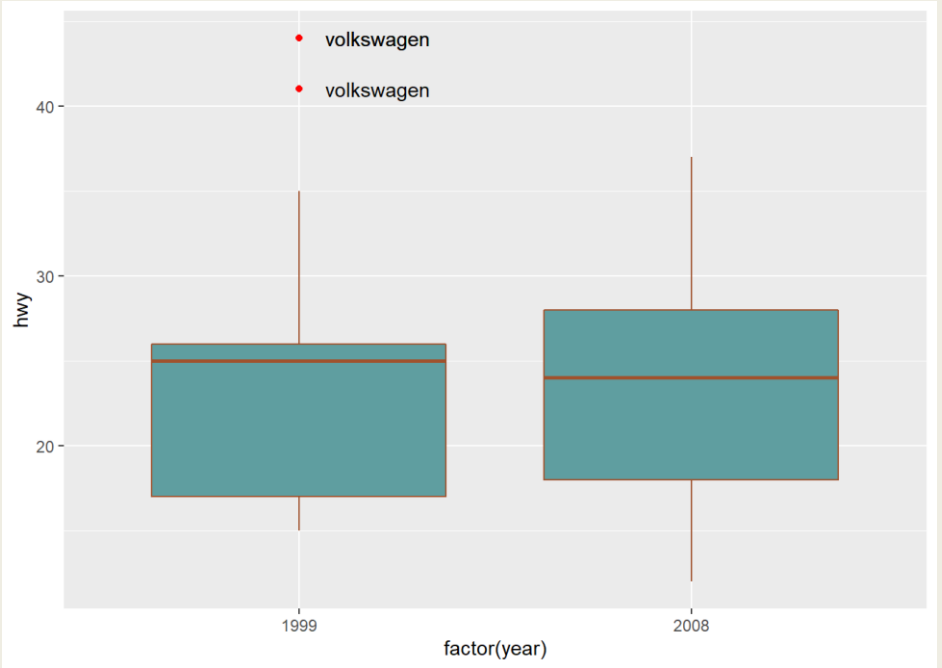
Density curve

- Useful for comparing distributions when there are large differences in sample size
- Standardized frequency distribution



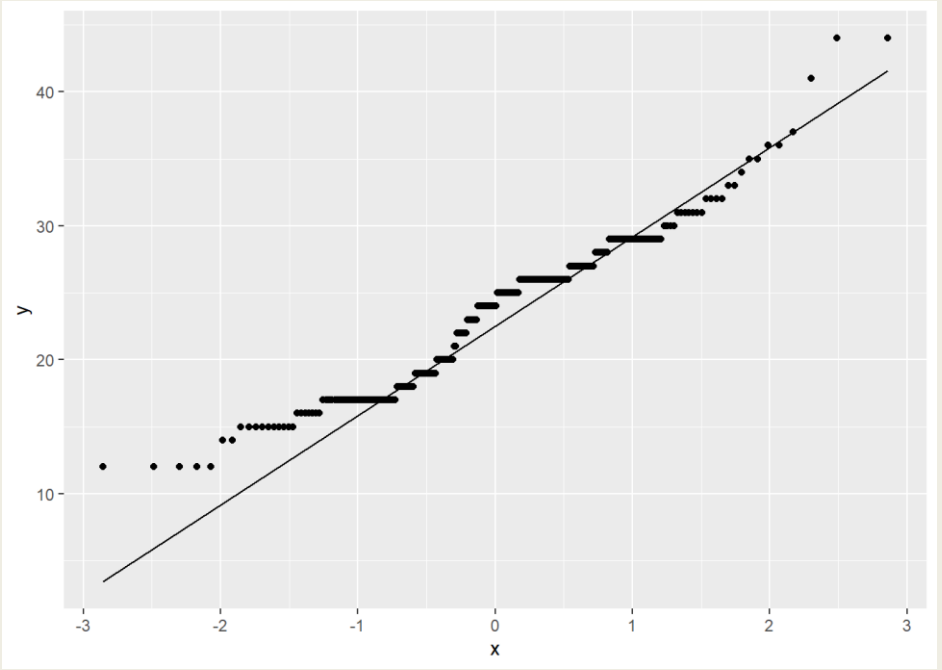
Boxplot

- Also known as a box and whisker plot is a handy chart for examining a distribution.
- However, it is not as popular as a histogram for examining distribution of a single variable. On the other hand, these plots are useful for comparing distributions. They are also useful for spotting outliers.
- Highlights dispersion in the data as well as outliers



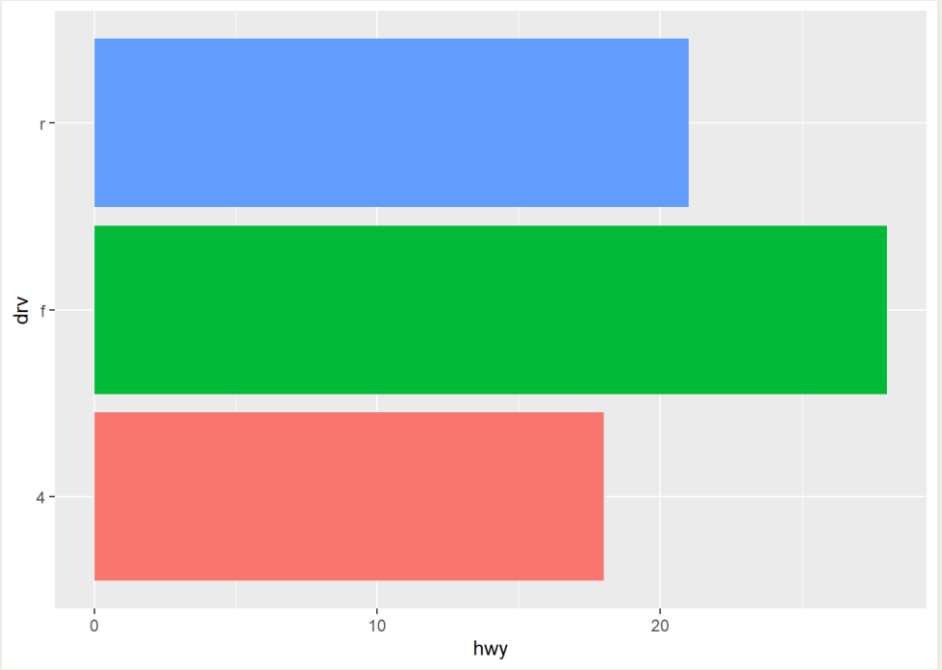
QQ Plot

- Graphical method to evaluate a distribution.
- QQ Plots are a useful way to check data against a distribution the date is thought to come from.
- Most commonly used to see if data comes from a normal distribution by plotting quantiles of data against quantiles expected from a normal distribution.



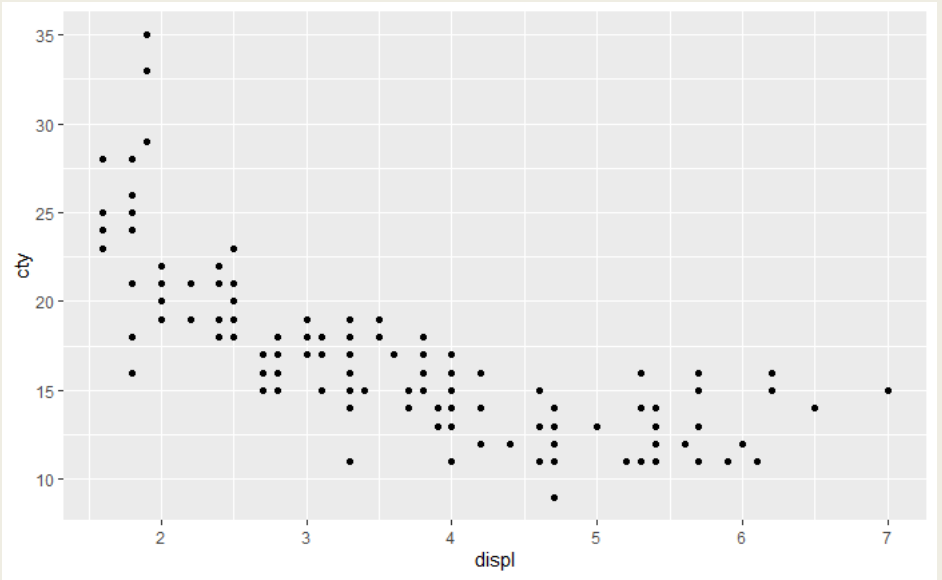
Bar Charts

- To compare numerical summaries across levels of a categorical variable



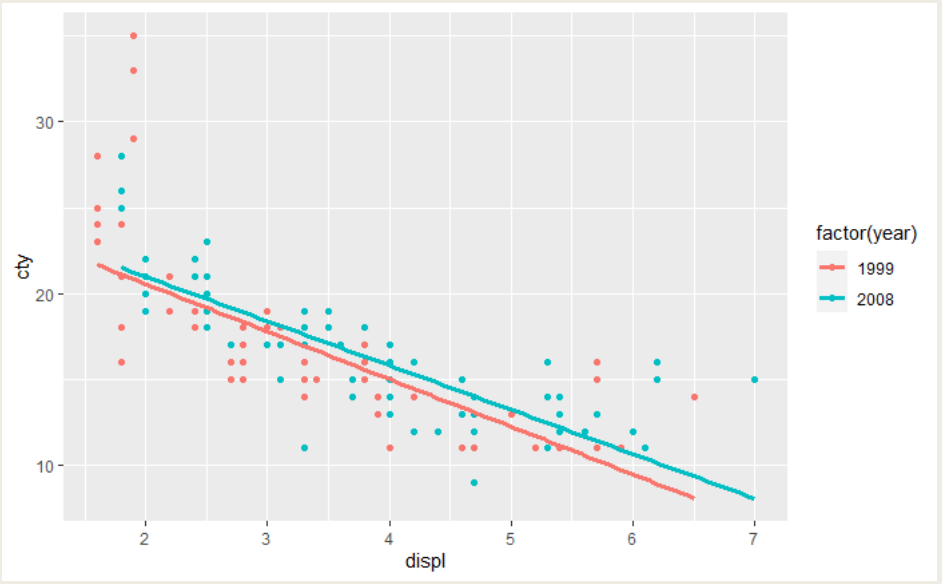
Scatterplots

- A set of dots on a graph where the location of the dots is based on its value for the two numeric variables.
- The pattern of the dots in a scatterplot may reveal the existence and nature of relationship between variables.

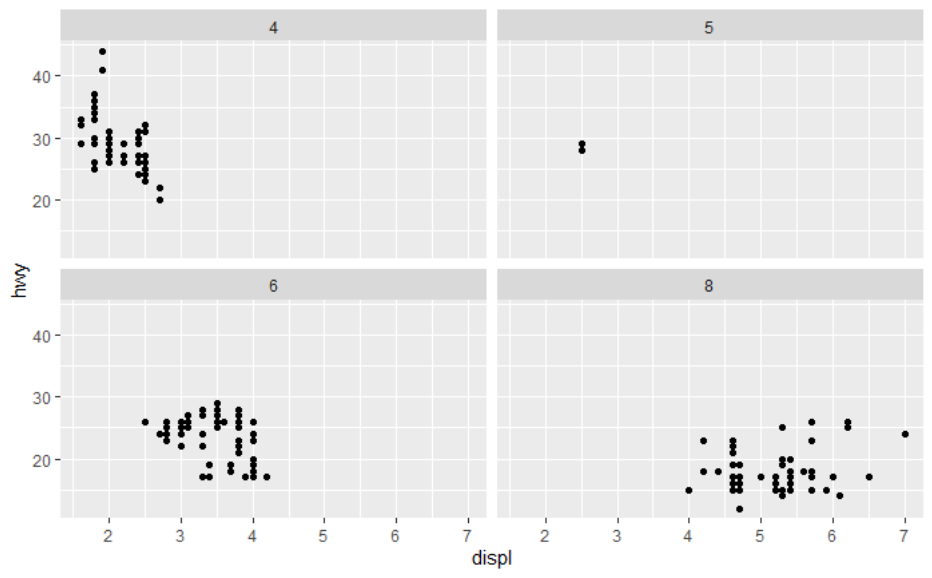


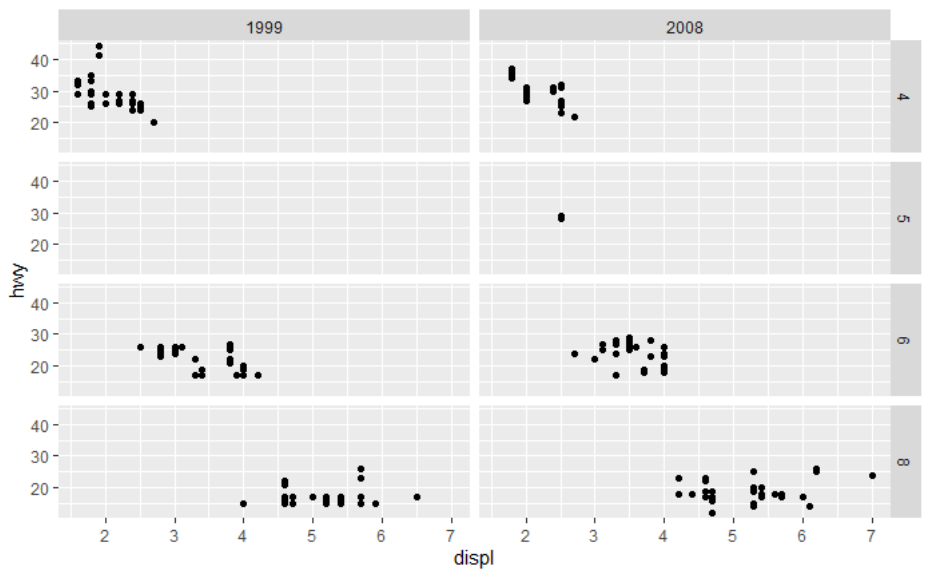
Line Graph

- With large number of points and overplotting, spotting a trend in a scatterplot can be difficult.
- A line graph can clearly depict the trend in the data.



Multiple Charts





Illustration

See VisualSummary.html

Conclusion

- In this module, we
  - *reviewed basics of R*
  - *explored data*
  - *looked at descriptive measures*
  - *examined visual summaries*