Lecture 04: R from Data Analytics Perspective.

KD44103 Big Data Analytics
Faculty of Computing & Informatics,
Universiti Malaysia Sabah

Learning outcomes

- Explain R vectors, factors, data structures, matrices and arrays
- Managing data with R and Exploring & understanding data
- Understand R packages, graphics and plotting
- Explain the main concepts of R for Data Science and data visualization

R Objects

- There are two main concepts behind the R language: objects and functions.
- An object can be seen as a storage space with an associated name. Everything in R is stored in an object. All variables, data, functions, etc. are stored in the memory of the computer in the form of named objects. Functions are a special type of R objects designed to carry out some operation. They usually take some arguments and produce a result by means of executing some set of operations (themselves usually other function calls).

R data structures

- The R data structures used most frequently in machine learning are:
  - Vectors
  - Factors
  - Lists
  - Data frames
  - Matrixes and arrays

Vector

- Vector stores an ordered set of values called **elements.**
  - A vector can contain any number of elements, **but all of the elements must be of the same type of values.**
  - Several vector types are commonly used in data analytics: integer (numbers without decimals), double (numbers with decimals), character (text data), and logical (TRUE or FALSE values).

Vector: Example

- Small vectors can be created by using the c() combine function:
- **For example:**

```
> subject_name <- c("John Doe", "Jane Doe", "Steve Graves")
> temperature <- c(98.1, 98.6, 101.4)
> flu_status <- c(FALSE, FALSE, TRUE)
```

Factors

- A factor is a special case of vector that is solely used to represent categorical or ordinal variables.
- E.g. In the medical dataset we are building, we might use a factor to represent gender, because it uses two categories: MALE and FEMALE.

Factors: Example

- To create a factor from a character vector, simply apply the factor() function.
- **For example:**

```
> gender <- factor(c("MALE", "FEMALE", "MALE"))
> gender
[1] MALE FEMALE MALE
Levels: FEMALE MALE
```

Lists

- A **list** is used for storing an ordered set of elements. Due to this flexibility, lists are often used to store various types of input and output data and sets of configuration parameters for machine learning models.

Lists: Example

- To illustrate lists, consider the medical patient dataset we have been constructing If we want to display all the data on John Doe (subject 1), we will need to enter five R commands:
- > subject_name[1]
- [1] "John Doe"
- > temperature[1]
- [1] 98.1
- > flu_status[1]
- [1] FALSE

Data frames

- data structure utilized in machine learning is the data frame, a structure analogous to a spreadsheet or database, since it has both rows and columns of data.
- **Example**

```
> x <- data.frame(foo = 1:4, bar = c(T, T, F, F))
```

Matrixes and arrays

- A matrix is a data structure that represents a two-dimensional table with rows and columns of data.
- To create a matrix, simply supply a vector of data to the matrix() function along with a parameter specifying the number of rows (nrow) or number of columns (ncol).

Matrixes and arrays: Example

- For example, to create a 2 x 2 matrix storing the numbers one through four, we can use the nrow parameter to request the data to be divided into two rows:
- **> m <- matrix(c(1, 2, 3, 4), nrow = 2)**
- **> m**
- **[,1] [,2]**
- **[1,] 1 3**
- **[2,] 2 4**

Managing data with R

- **Saving, loading, and removing R data structures:**

➢ save(x, y, z, file = "mydata.RData")

➢ **load("mydata.RData")**

➢ **ls()**

➢ **rm(m, subject1)**

Importing and saving data from CSV files

- Perhaps the most common tabular text file format is the **CSV (Comma-Separated**
- **Values**) file, which as the name suggests, uses the comma as a delimiter. The CSV
- files can be imported to and exported from many common applications.

```
> pt_data <- read.csv("pt_data.csv",
stringsAsFactors = FALSE)
```

Exploring and understanding data

- After collecting data and loading it into R's data structures, the next step in the machine learning process involve examining the data in detail.
- **> str:** function provides a method to display the structure of R data structures such as data frames, vectors, or lists.
- **summary():** function displays several common summary statistics

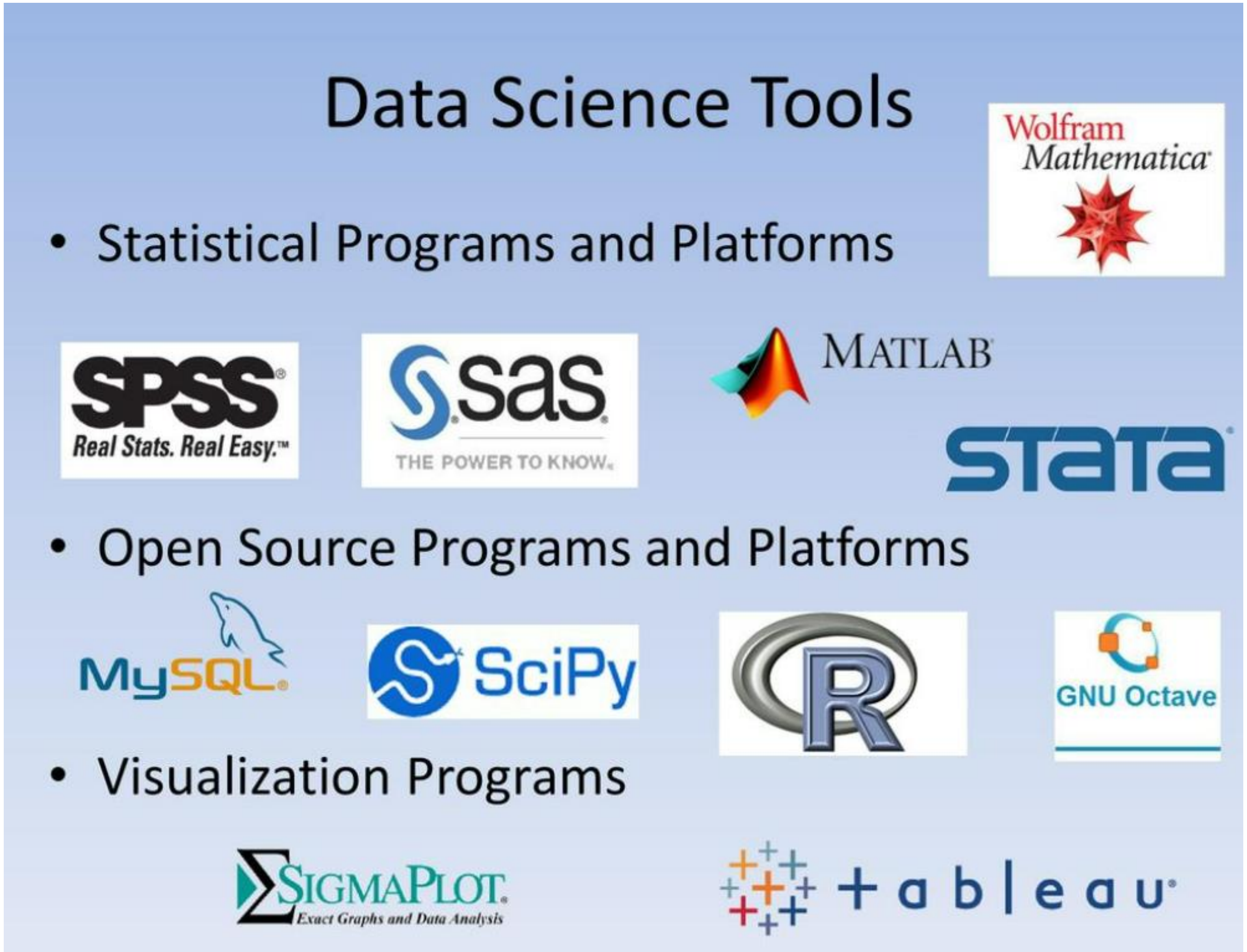
Data Structures

- Supports virtually any type of data
- Numbers, characters, logicals (TRUE/ FALSE)
- Arrays of virtually unlimited sizes
- Simplest: Vectors and Matrices
- Lists: Can Contain mixed type variables
- Data Frame: Rectangular Data Set

Data Structure in R

	Linear	Rectangular
All Same Type	VECTORS	MATRIX*
Mixed	LIST	DATA FRAME

Some data science tools



Data Science Tools

- Statistical Programs and Platforms
- Open Source Programs and Platforms
- Visualization Programs

Wolfram Mathematica
SPSS
Real Stats. Real Easy.™
sas
THE POWER TO KNOW.
MATLAB
STATA
MySQL
SciPy
R
GNU Octave
SigmaPlot
Exact Graphs and Data Analysis
tableau

R Datasets

R comes with a number of sample datasets that you can experiment with. Type

> data( )

to see the available datasets. The results will depend on which packages you have loaded. Type

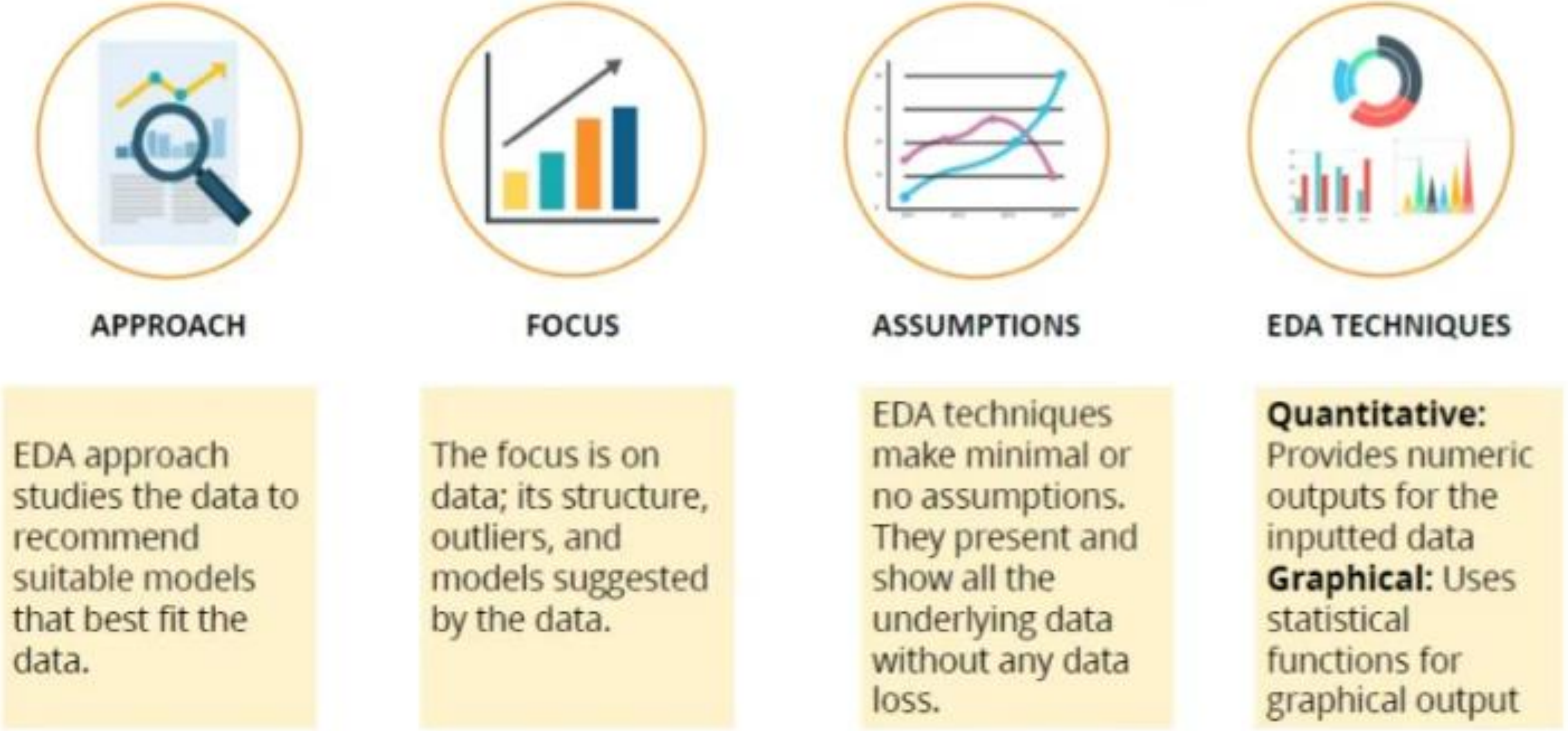
help(*datasetname***)**

for details on a sample dataset.

Importing data in R

You can importing data from four types of files in R

Exploratory data analysis (EDA)



APPROACH

EDA approach studies the data to recommend suitable models that best fit the data.

FOCUS

The focus is on data; its structure, outliers, and models suggested by the data.

ASSUMPTIONS

EDA techniques make minimal or no assumptions. They present and show all the underlying data without any data loss.

EDA TECHNIQUES

Quantitative: Provides numeric outputs for the inputted data

Graphical: Uses statistical functions for graphical output

R Packages

- One of the strengths of R is that the system can easily be extended. The system allows you to write new functions and package those functions in a so called `R package' (or `R library’).
- The R package may also contain other R objects, for example data sets or documentation. There is a lively R user community and many R packages have been written and made available on CRAN for other users.
- Just a few examples, there are packages for portfolio optimization, drawing maps, exporting objects to html, time series analysis, spatial statistics and the list goes on and on.

R Packages

- Around (30) packages are downloaded when you download R.
- To use a function in an R package, that package has to be attached to the system. When you start R not all of the downloaded packages are attached, only around seven packages are attached to the system by default.
- You can use the function search to see a list of packages that are currently attached to the system, this list is also called the search path.

```
> search()
[1] ".GlobalEnv" "package:stats" "package:graphics"
[4] "package:grDevices" "package:datasets" "package:utils"
[7] "package:methods" "Autoloads" "package:base"
```

R Packages

- To attach another package to the system you can use the menu or the library function. Via the menu:

  Select the `Packages' menu and select `Load package...', a list of available packages on your system will be displayed. Select one and click `OK', the package is now attached to your current R session. Via the library function:

  ```
  > library(MASS)
  > shoes
  $A
  [1] 13.2 8.2 10.9 14.3 10.7 6.6 9.5 10.8 8.8 13.3
  $B
  [1] 14.0 8.8 11.2 14.2 11.8 6.4 9.8 11.3 9.3 13.6
  ```

R Packages

- The function library can also be used to list all the available libraries on your system with a short description. Run the function without any arguments

```
> library()
Packages in library 'C:/PROGRA~1/R/R-25~1.0/library':
base                The R Base Package
Boot                 Bootstrap R (S-Plus) Functions  (Canty)
class               Functions for Classification
cluster             Cluster Analysis Extended Rousseeuw  et al.
codetools           Code Analysis Tools for R
datasets            The R Datasets Package
DBI                 R Database Interface
foreign             Read Data Stored by Minitab, S, SAS,
            SPSS, Stata, Systat, dBase, ...
graphics            The R Graphics Package
```

R Packages

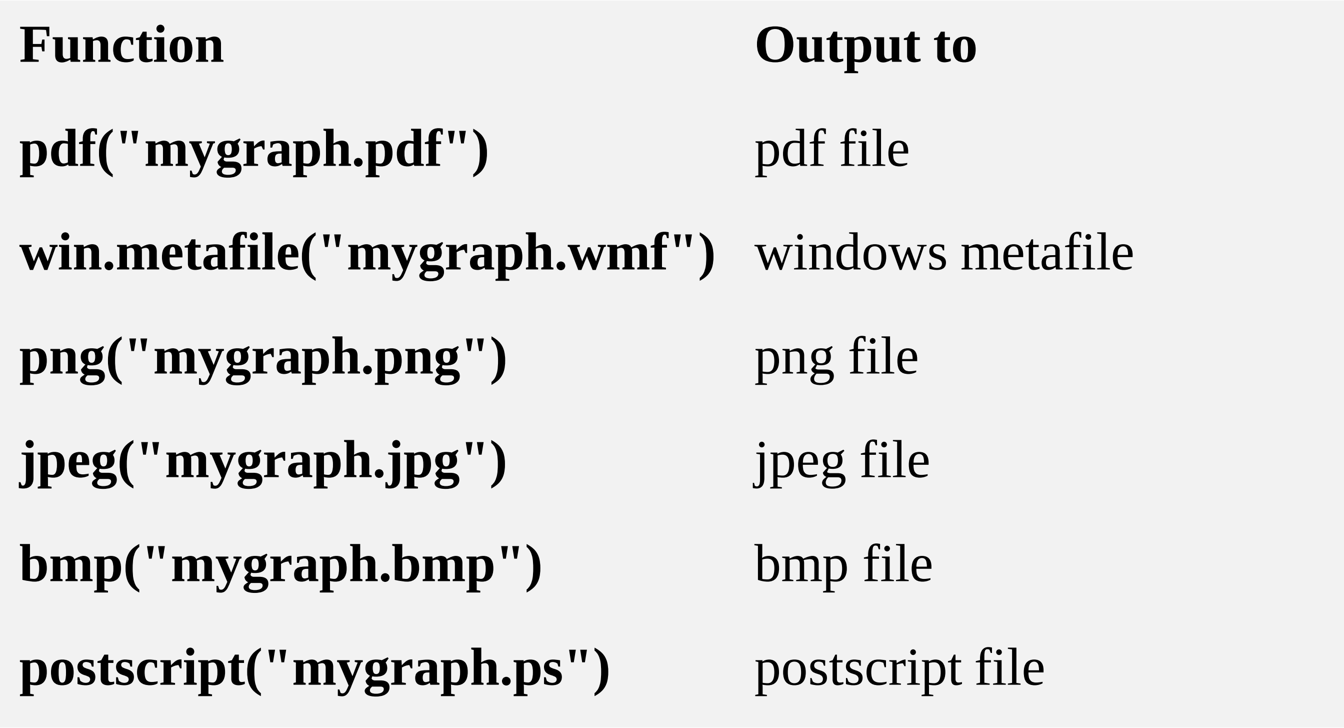
```
install = function() {
install.packages(c("moments","graphics","Rcmdr","hexbin"),
repos="http://lib.stat.cmu.edu/R/CRAN")
}
install()
```

Graphics

- Plot an object, like: plot(num.vec)
  - here plots against index numbers
- Plot sends to graphic devices
  - can specify which graphic device you want
    - postscript, gif, jpeg, etc…
    - you can turn them on and off, like: dev.off()
- Two types of plotting
  - high level: graphs drawn with one call
  - Low Level: add additional information to existing graph

Graphs

To redirect graphic output use one of the following functions. Use **dev.off( )** to return output to the terminal.



Function	**Output to**
pdf("mygraph.pdf")	pdf file
win.metafile("mygraph.wmf")	windows metafile
png("mygraph.png")	png file
jpeg("mygraph.jpg")	jpeg file
bmp("mygraph.bmp")	bmp file
postscript("mygraph.ps")	postscript file

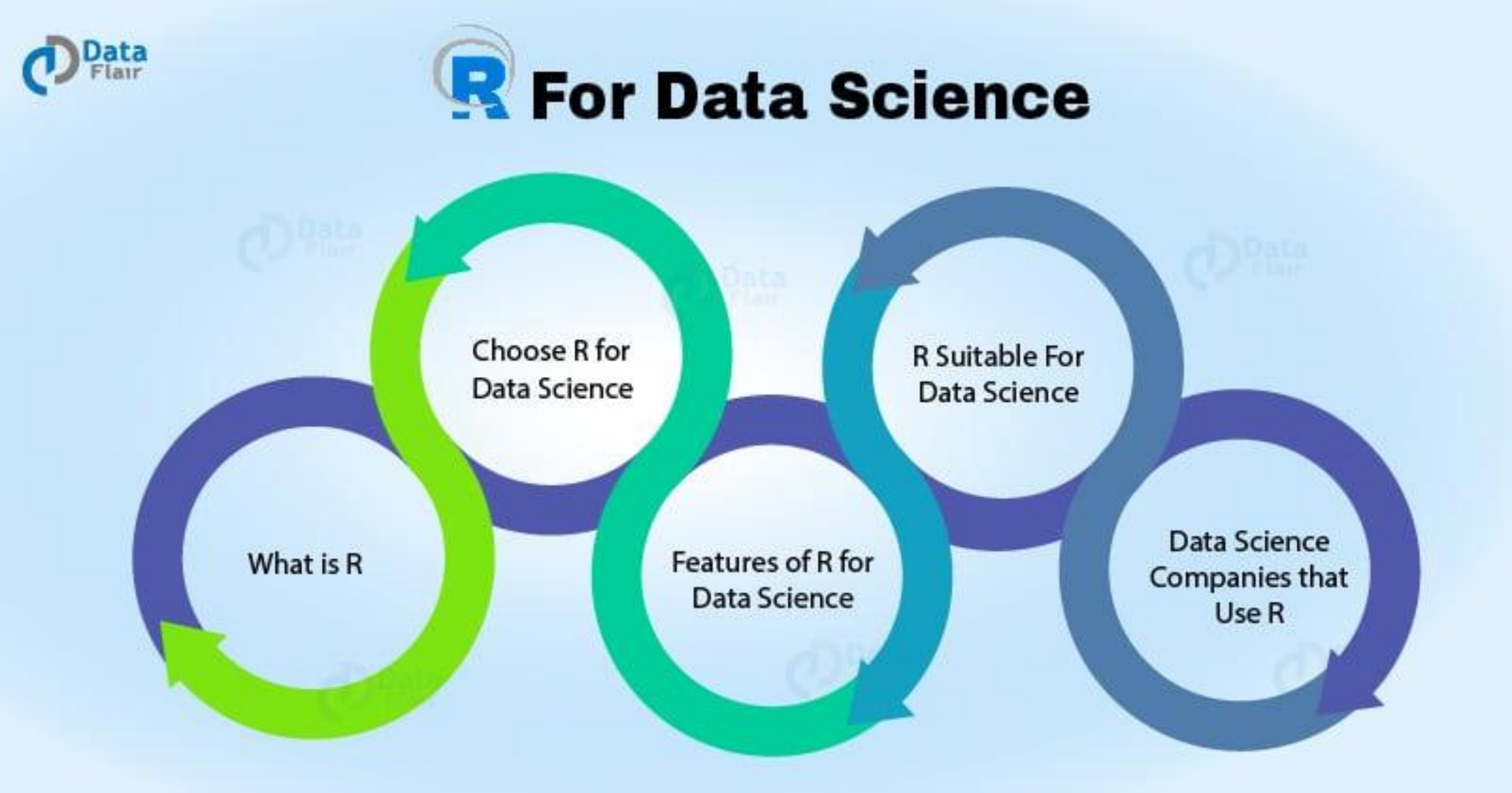
Redirecting Graphs

```
# example - output graph to jpeg file
    jpeg("c:/mygraphs/myplot.jpg")
    plot(x)
    dev.off()
```

Why R for data science

1. R is built for statistics.
2. R is a popular language for data science at top tech firms
3. Learning the data science basics is arguably easier in R.
4. Amazing packages that make your life easier.
5. Inclusive, growing community of data scientists and statisticians.
6. Put another tool in your toolkit.

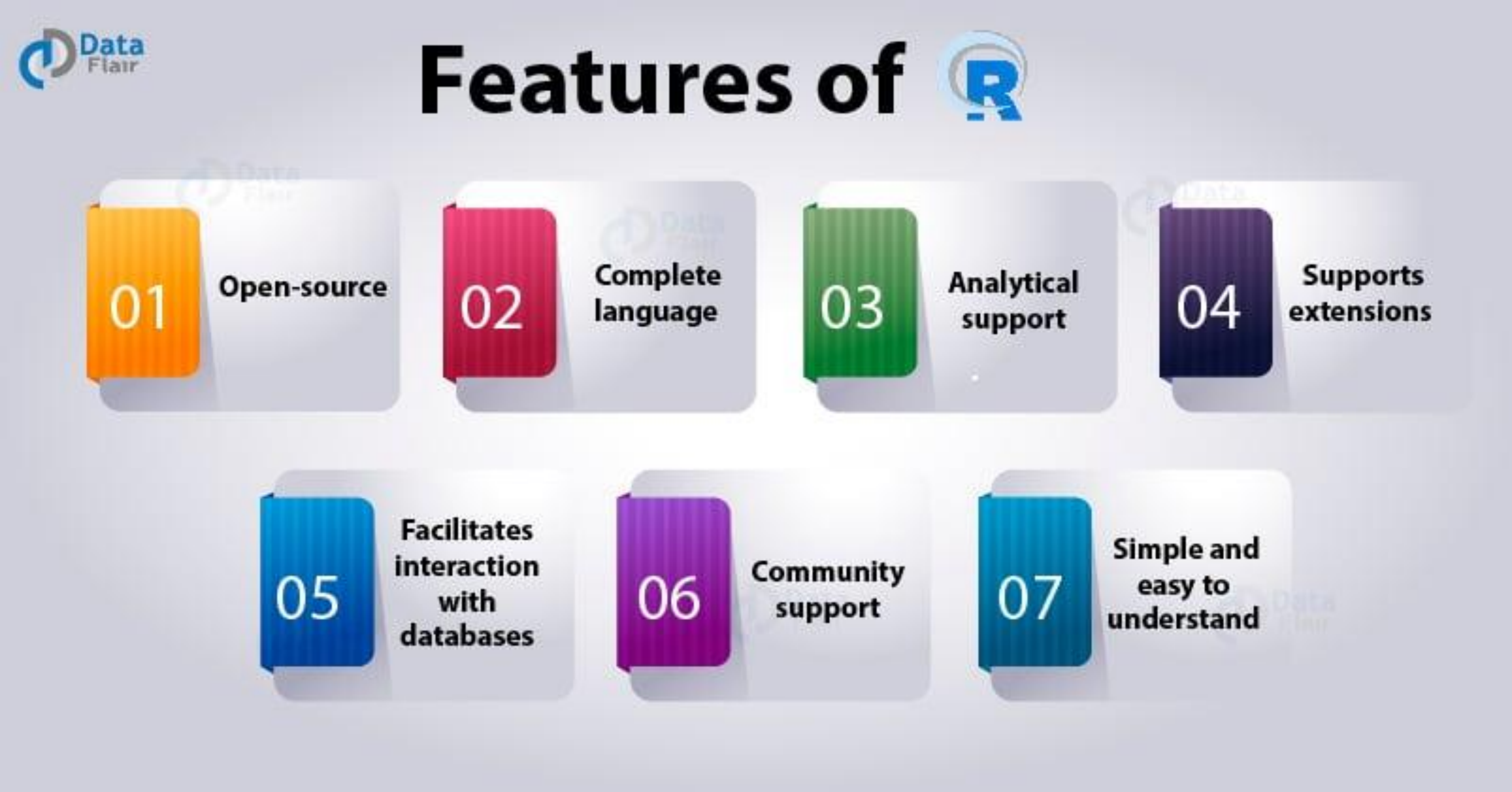
Why R for data science


Data Flair
R For Data Science
What is R
Choose R for Data Science
Features of R for Data Science
R Suitable For Data Science
Data Science Companies that Use R

Why Choose R for Data Science - Discover Top Features and Companies - DataFlair (data-flair.training)

Features of R

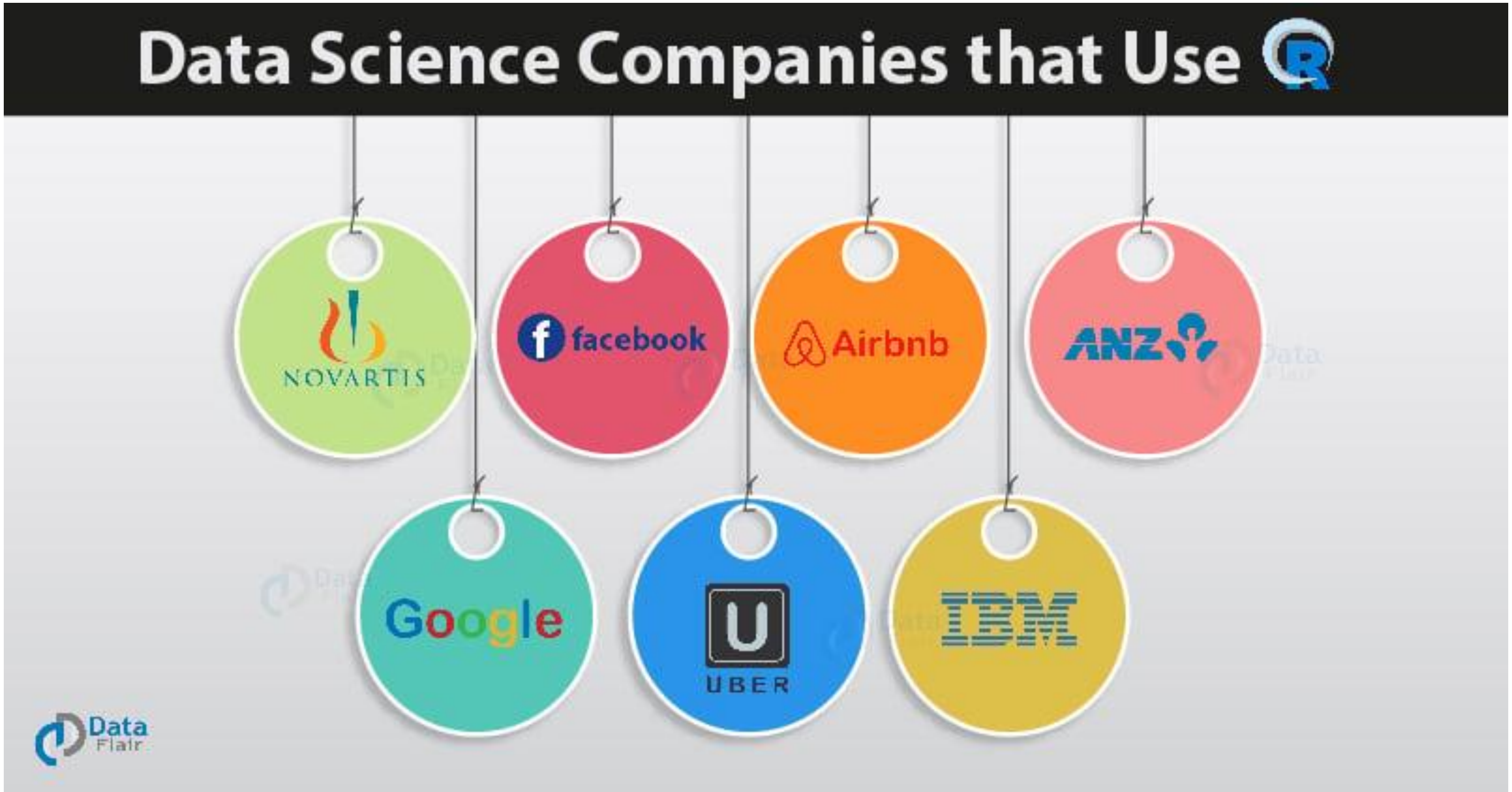
R has important features which we should discuss to understand the role that make R suitable for Data Science.



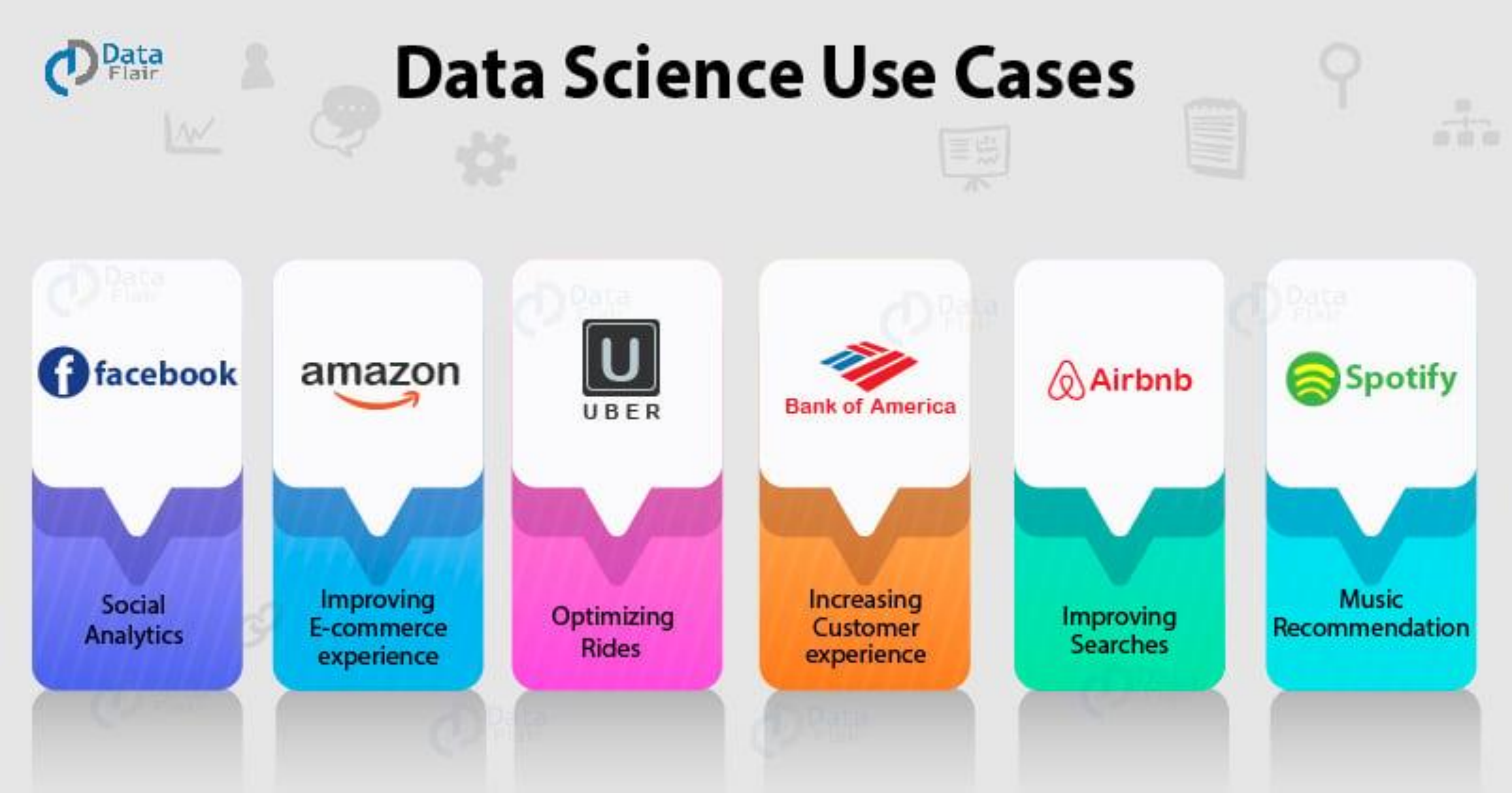
Why Choose R for Data Science - Discover Top Features and Companies - DataFlair (data-flair.training)

Data Science Companies that Use R

Some of the major data science companies that use R analysis and statistical modeling are:



Top 6 Data Science Use Cases that are Changing the World



Top 6 Data Science Use Cases that are Changing the World - DataFlair (data-flair.training)

R for plotting

- R has several systems for making graphs, but ggplot2 is one of the most elegant and most versatile.
- ggplot2 implements the grammar of graphics, a coherent system for describing and building graphs.
- With ggplot2, you can do more faster by learning one system and applying it in many places.
- To access the datasets, help pages, and functions that we will use in this chapter, load the tidyverse

R for plotting

- Content may be stored in objects using the assignment operator. This operator is denoted by an angle bracket followed by a minus sign (<-).
- > x <- 945
- The effect of the previous instruction is thus to store the number 945 on an object named x. By simply entering the name of an object at the R prompt one can see its contents:9 > x [1] 945

R for plotting

- Below you will find other examples of assignment statements. These examples should make it clear that this is a destructive operation as any object can only have a single content at any time t. This means that by assigning some new content to an existing object, you in effect lose its previous content: > y <- 39

```
> y [1] 39
> y <- 43
> y
  [1] 43
```

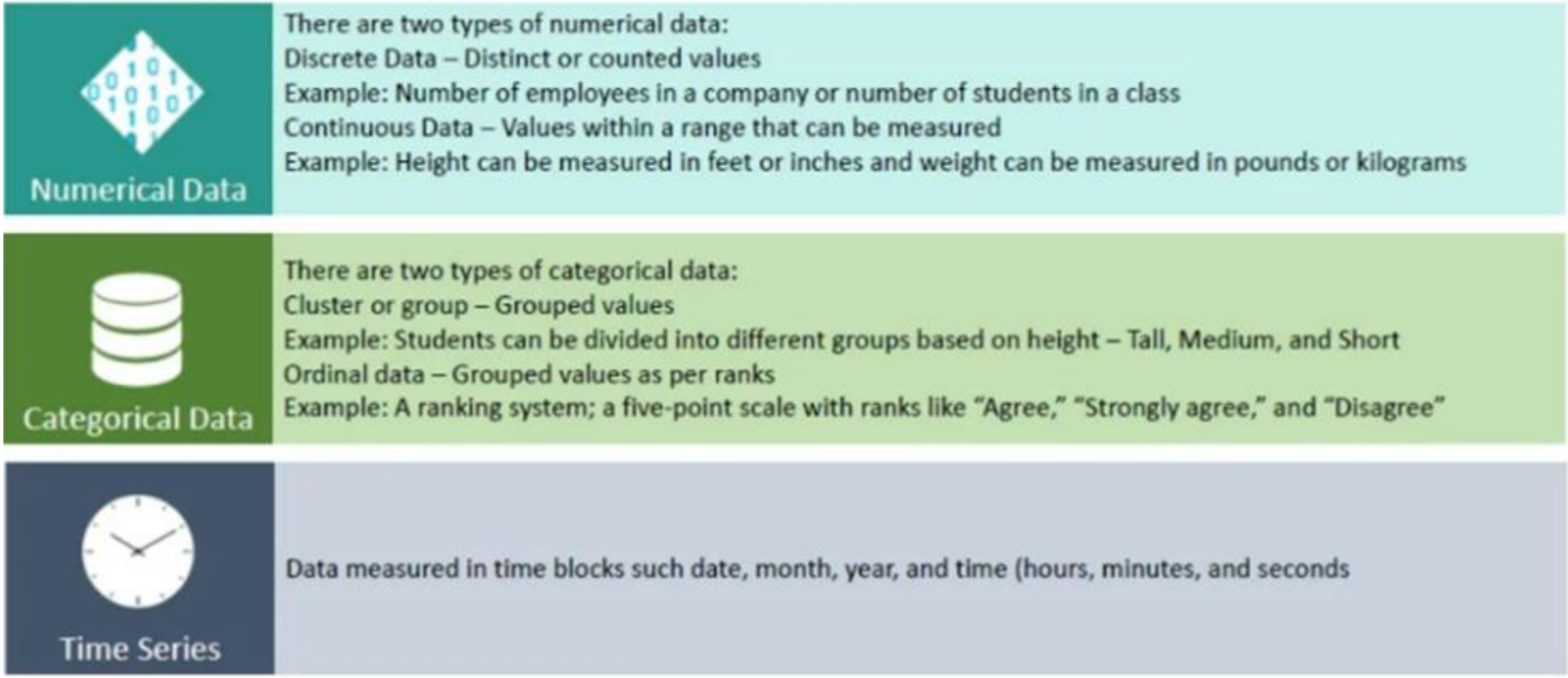
R for plotting

- You can also assign numerical expressions to an object. In this case the object will store the result of the expression:

```
> z <- 5
 > w <- z^2
> w
[1] 25
> i <- (z * 2 + 45)/2
> i
```

Data types for plotting

- There are different types used for plotting



Numerical Data

There are two types of numerical data:
Discrete Data – Distinct or counted values
Example: Number of employees in a company or number of students in a class
Continuous Data – Values within a range that can be measured
Example: Height can be measured in feet or inches and weight can be measured in pounds or kilograms

Categorical Data

There are two types of categorical data:
Cluster or group – Grouped values
Example: Students can be divided into different groups based on height – Tall, Medium, and Short
Ordinal data – Grouped values as per ranks
Example: A ranking system; a five-point scale with ranks like "Agree," "Strongly agree," and "Disagree"

Time Series

Data measured in time blocks such date, month, year, and time (hours, minutes, and seconds

Important Packages of R for Data Science

- Packages in R plays an important role, here are some example of popular and useful Packages:

1- ggplot2

2- Tidyr

3- Dplyr

The Tidyverse

- R package is a collection of functions, data, and documentation that extends the capabilities of base R.
- Using packages is key to the successful use of R.
- Ttidyverse is a coherent system of packages for importing, tidying, transforming, exploring, and visualizing data.
- Tidyverse packages are intended to make statisticians and data scientists more productive by guiding them through workflows that facilitate communication, and result in reproducible work products.

The Tidyverse

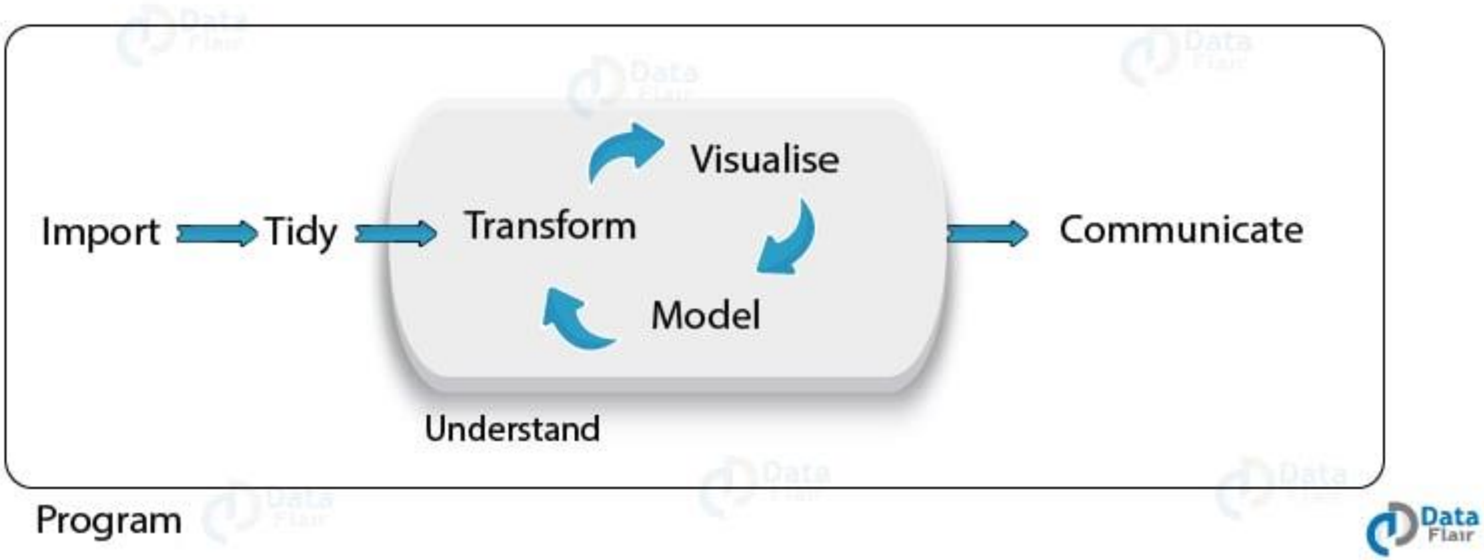
- Tidyverse contains many packages that can help a lot. The packages in the tidyverse share a common philosophy of data and R programming and are designed to work together naturally.
- You can install the complete tidyverse with a single line of code: - *install.packages("tidyverse")*

The Tidyverse

- You will not be able to use the functions, objects, and help files in a package until you load it with library(). Once you have installed a package, you can load it with the library() function:

- *library(tidyverse)*

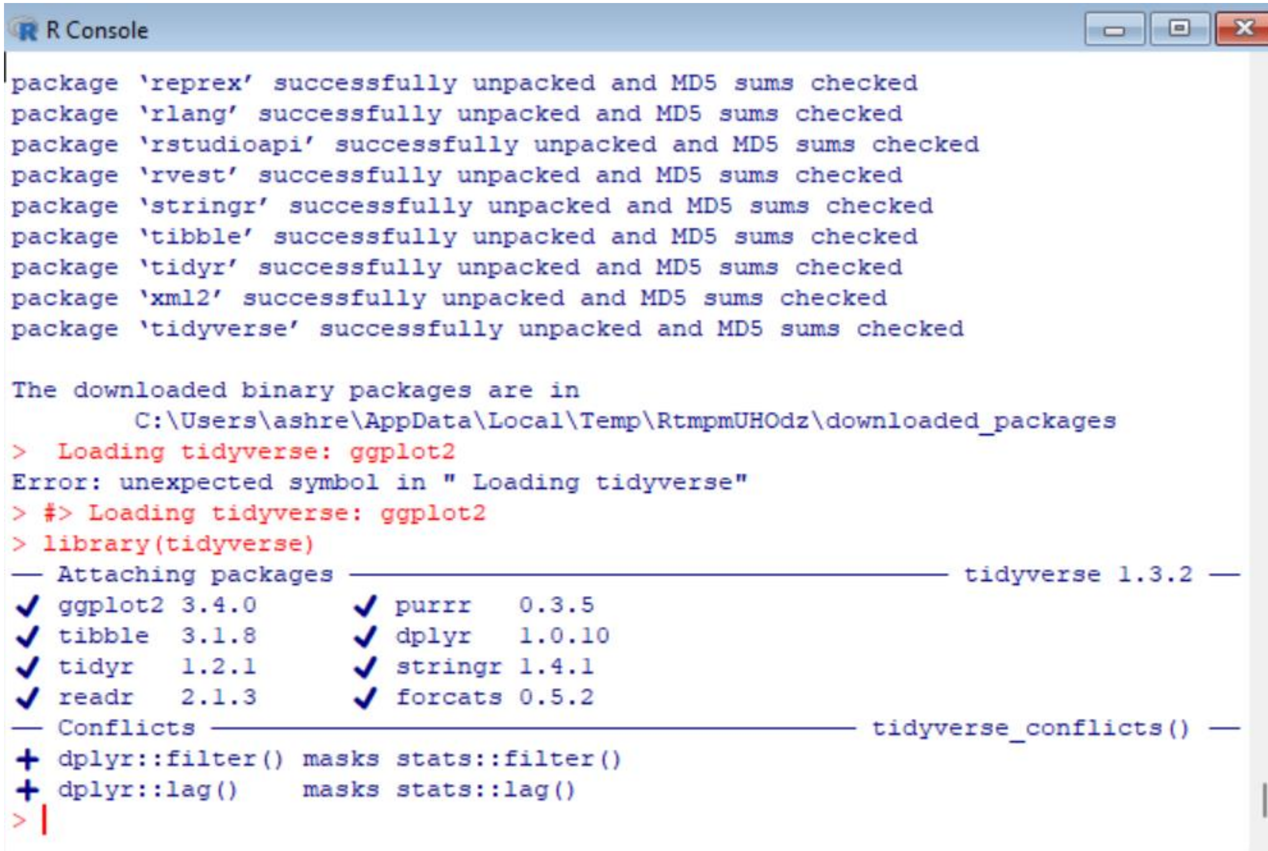
The Tidyverse



The Tidyverse

- This tells you that tidyverse is loading the ggplot2, tibble, tidyr, readr, purrr, and dplyr packages. These are considered to be the core of the tidyverse because you'll use them in almost every analysis.
- Packages in the tidyverse change fairly frequently. You can see if updates are available, and optionally install them, by running:

- *tidy verse_update().*



```
package ‘reprex’ successfully unpacked and MD5 sums checked
package ‘rlang’ successfully unpacked and MD5 sums checked
package ‘rstudioapi’ successfully unpacked and MD5 sums checked
package ‘rvest’ successfully unpacked and MD5 sums checked
package ‘stringr’ successfully unpacked and MD5 sums checked
package ‘tibble’ successfully unpacked and MD5 sums checked
package ‘tidyr’ successfully unpacked and MD5 sums checked
package ‘xml2’ successfully unpacked and MD5 sums checked
package ‘tidyverse’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\ashre\AppData\Local\Temp\RtmpmUHOdz\downloaded_packages
>  Loading tidyverse: ggplot2
Error: unexpected symbol in " Loading tidyverse"
> #> Loading tidyverse: ggplot2
> library(tidyverse)
— Attaching packages ——————————————————————————— tidyverse 1.3.2 —
✔ ggplot2 3.4.0      ✔ purrr   0.3.5
✔ tibble  3.1.8      ✔ dplyr   1.0.10
✔ tidyr   1.2.1      ✔ stringr 1.4.1
✔ readr   2.1.3      ✔ forcats 0.5.2
— Conflicts ————————————————————————————— tidyverse_conflicts() —
+ dplyr::filter() masks stats::filter()
+ dplyr::lag()    masks stats::lag()
> 
```

Other Packages

- There are many other excellent packages that are not part of the tidyverse, because they solve problems in a different domain, or are designed with a different set of underlying principles. This doesn't make them better or worse, just different.
- Example of e data packages from outside the tidyverse:

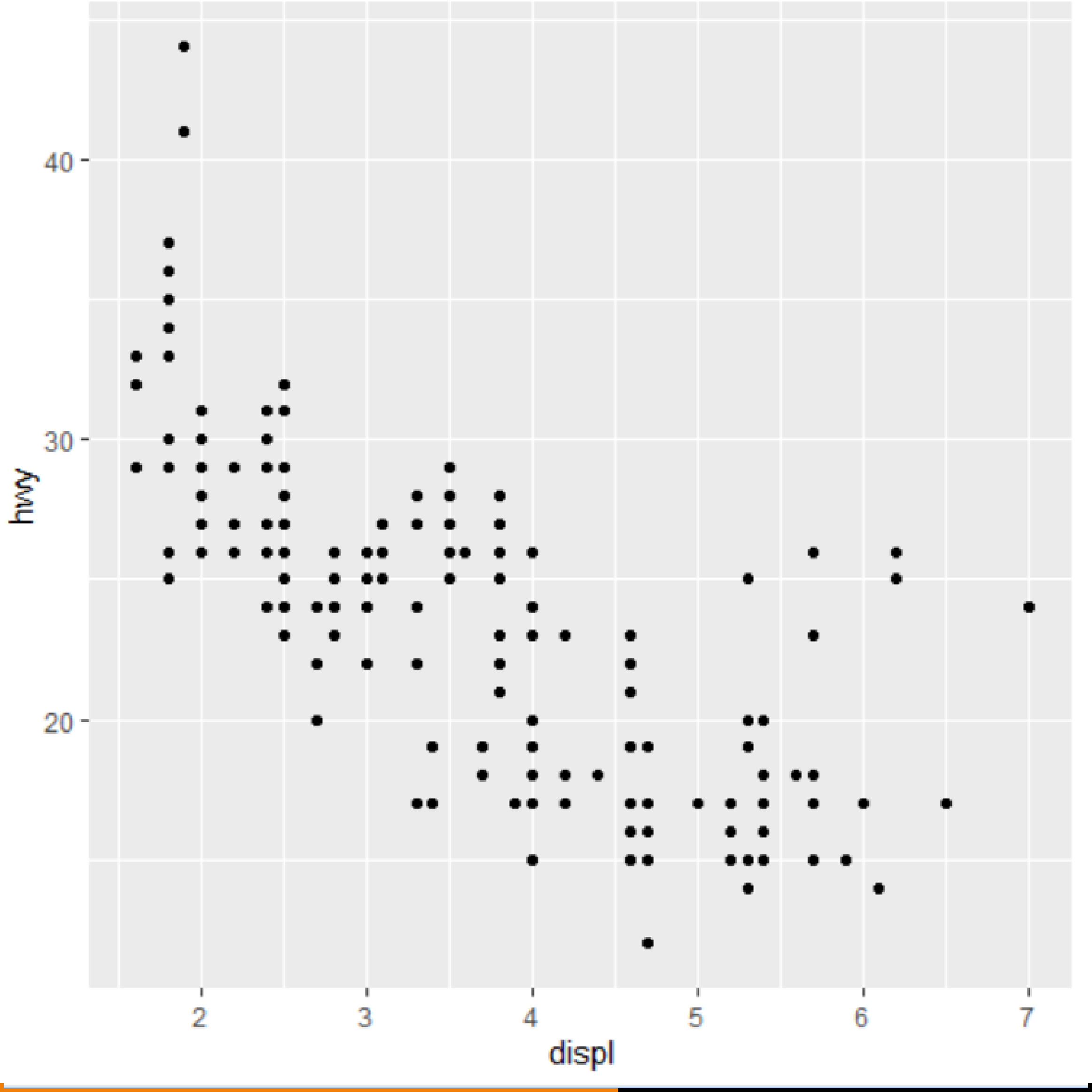
- *install.packages(c("nycflights13", "gapminder", "Lahman"))*

Data Visualization with ggplot2

- ggplot2 is one of the most elegant and most versatile.
- With ggplot2, you can do more faster by learning one system and applying it in many places.
- Creating a *ggplot* To plot mpg, run this code to put displ on the x-axis and *hwy* on the y-axis:

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

Data Visualization with ggplot2



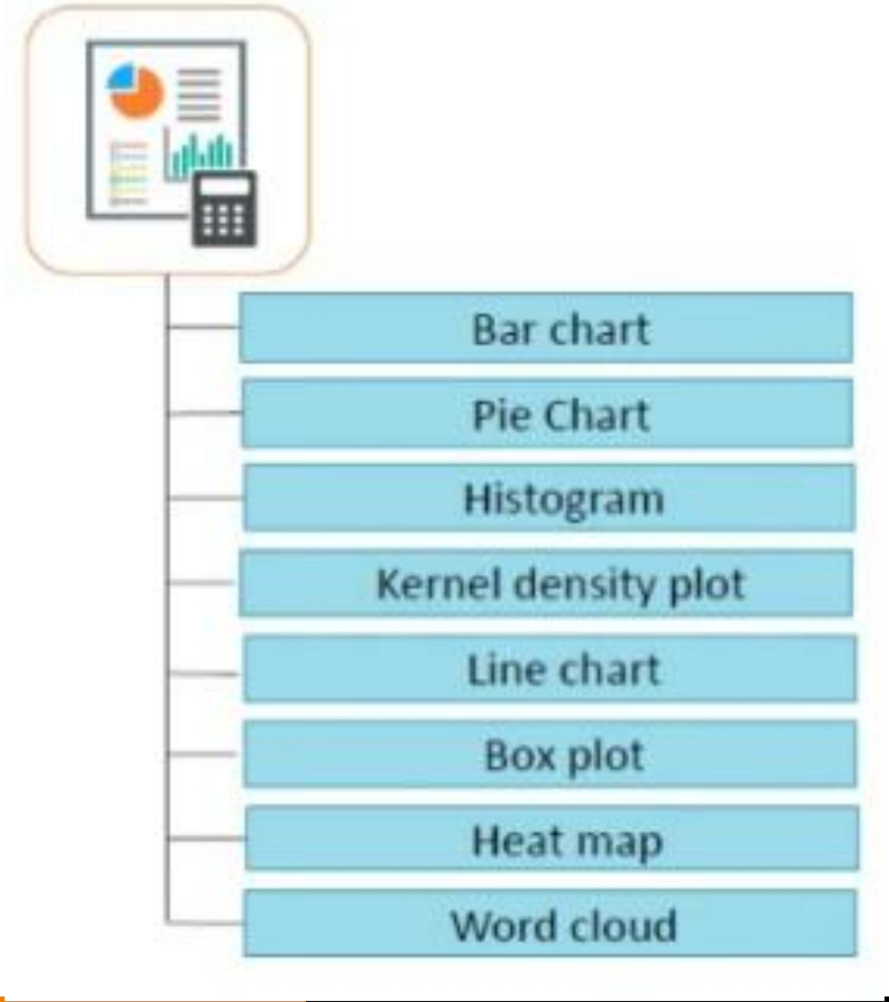
Geometric Objects

- A geom is the geometrical object that a plot uses to represent data. People often describe plots by the type of geom that the plot uses.
- The plot on the left uses the point geom, and the plot on the right uses the smooth geom, a smooth line fitted to the data.

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

Explore data in R

- R can support 8 types of graphics:



Bar chart

Pie Chart

Histogram

Kernel density plot

Line chart

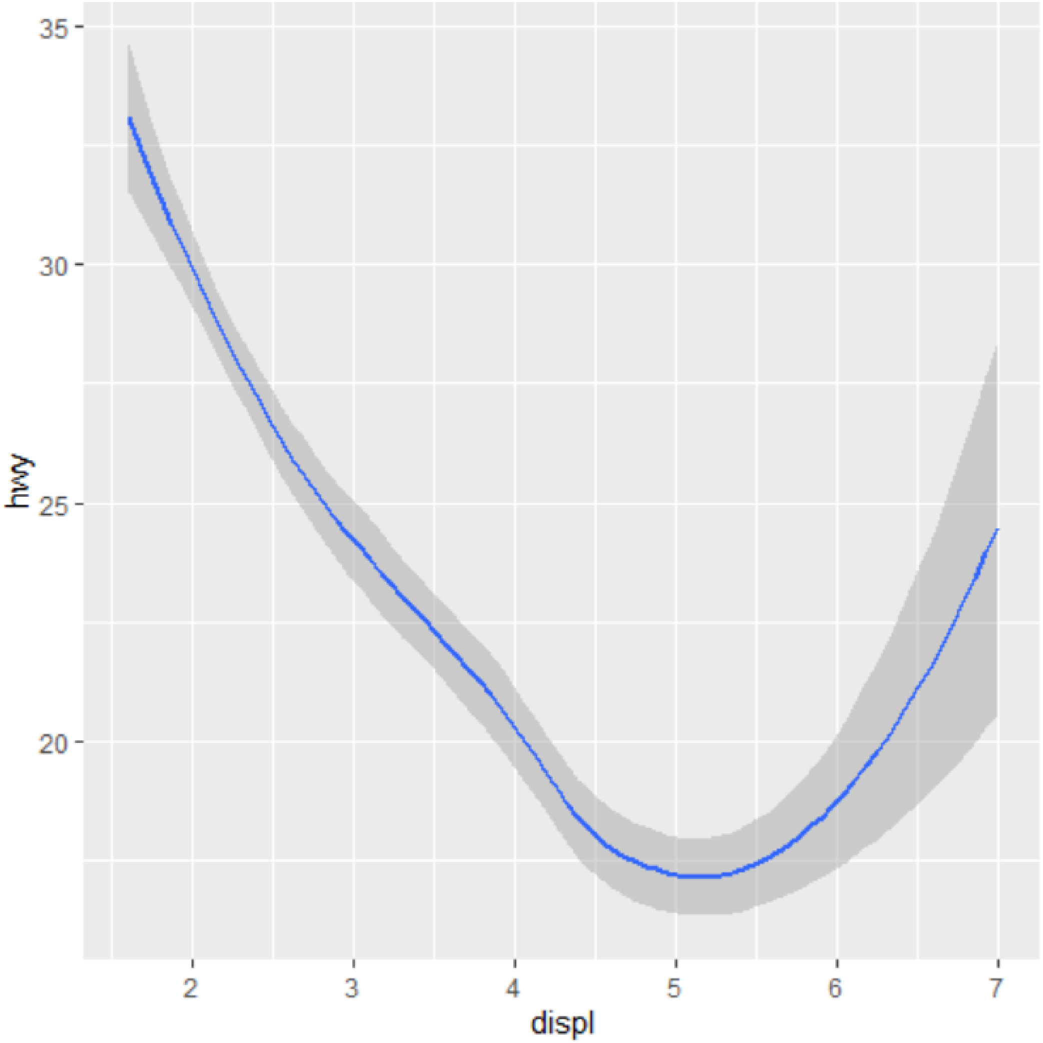
Box plot

Heat map

Word cloud

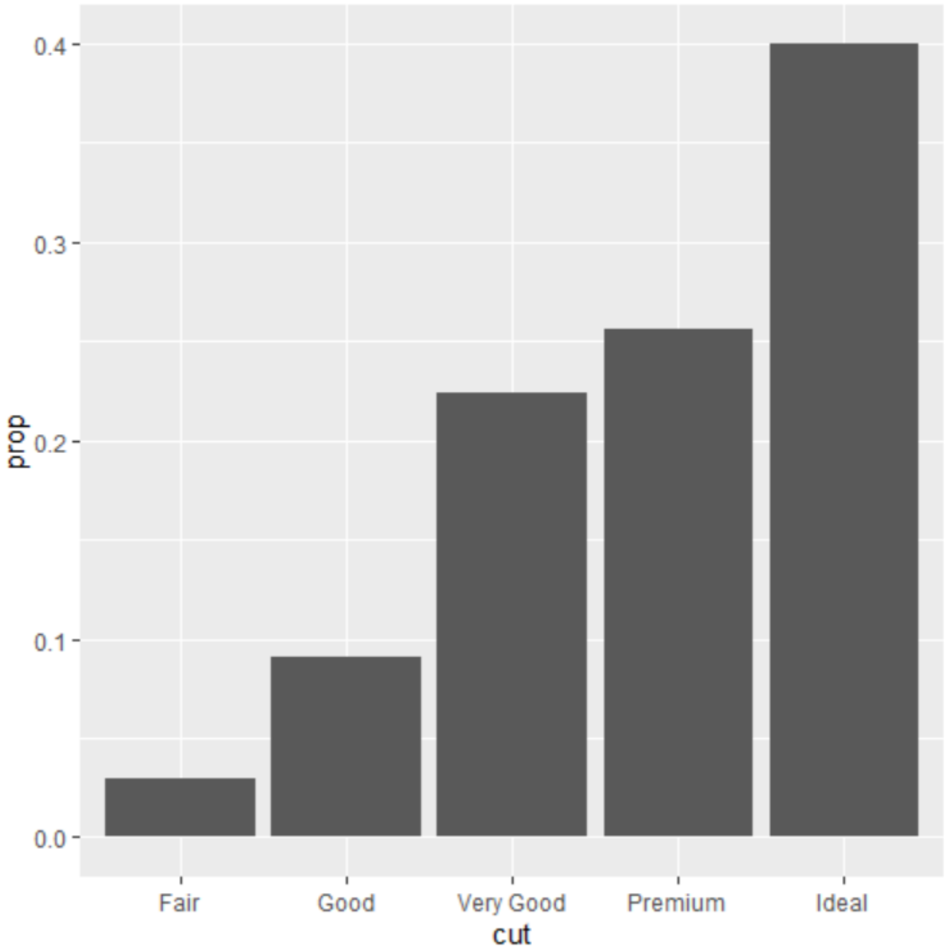
Data Visualization with ggplot2

```
# right
ggplot(data = mpg) +
 geom_smooth(mapping
   = aes(x = displ, y = hwy))
```



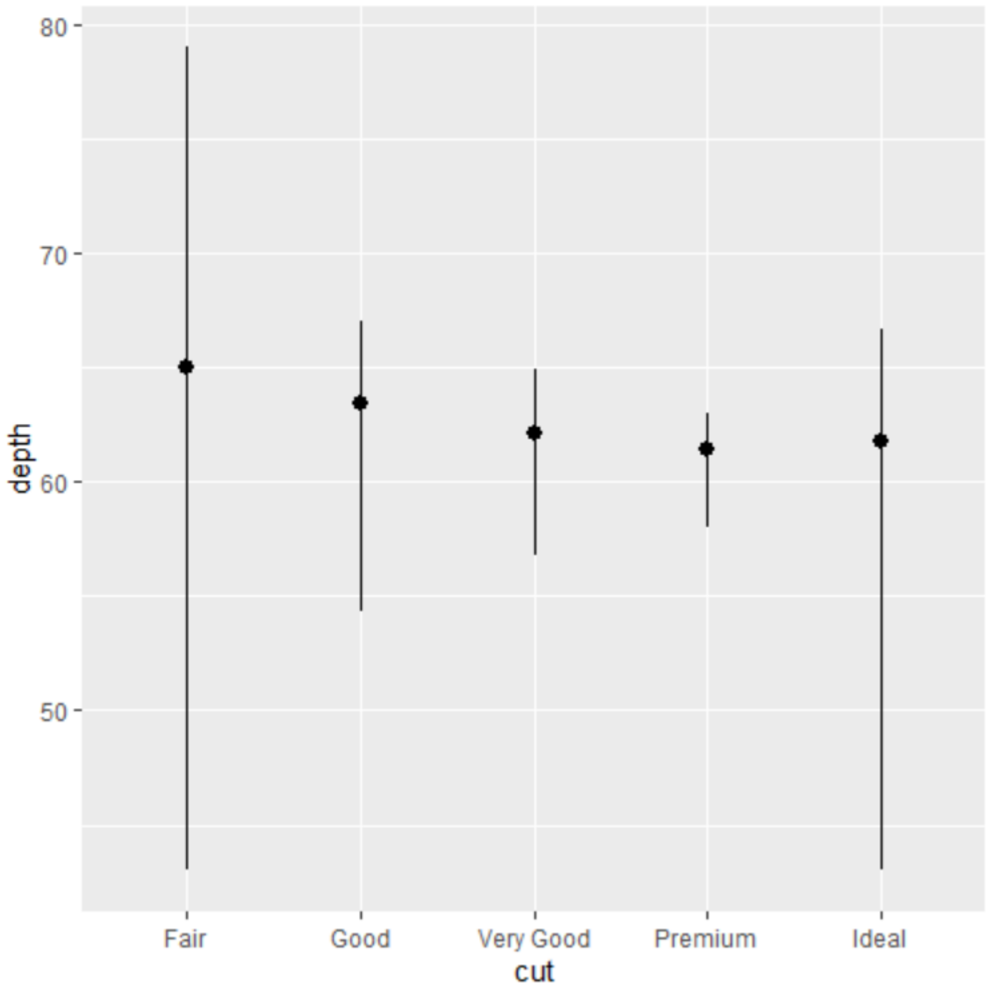
Data Visualization with ggplot2

```
ggplot(data = diamonds) +
geom_bar( mapping =
aes(x = cut, y = ..prop.., group = 1))
```



Data Visualization with ggplot2

```
ggplot(data = diamonds) +
stat_summary(
mapping = aes(x = cut, y = depth),
fun.ymin = min,
fun.ymax = max,
fun.y = median)
```



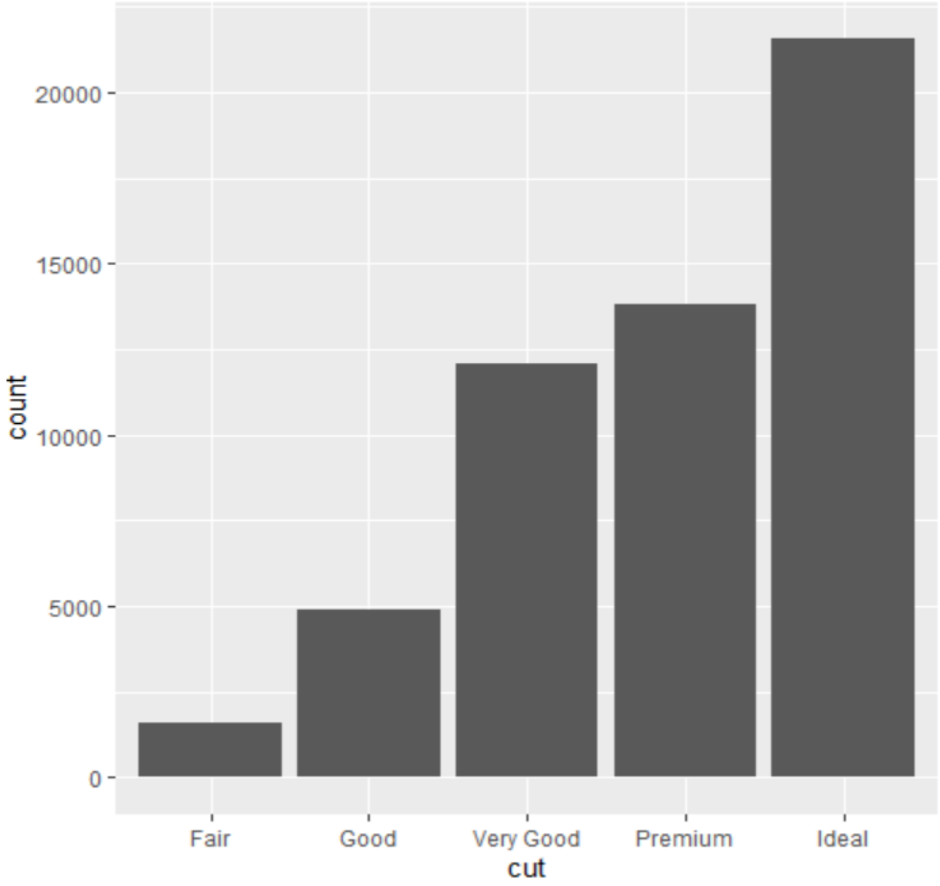
Data Visualization with ggplot2

- ggplot2 provides over 20 stats to use.
- Each stat is a function, can get help in the usual way, e.g., ?*stat_bin*.
- To see a complete list of stats, try the ggplot2 cheatsheet.

Statistical Transformations

- Bar charts seem simple, but they are interesting because they reveal something subtle about plots. Consider a basic bar chart, as drawn with *geom_bar()*.

ggplot(data = diamonds) +
geom_bar(mapping = aes(x = cut))

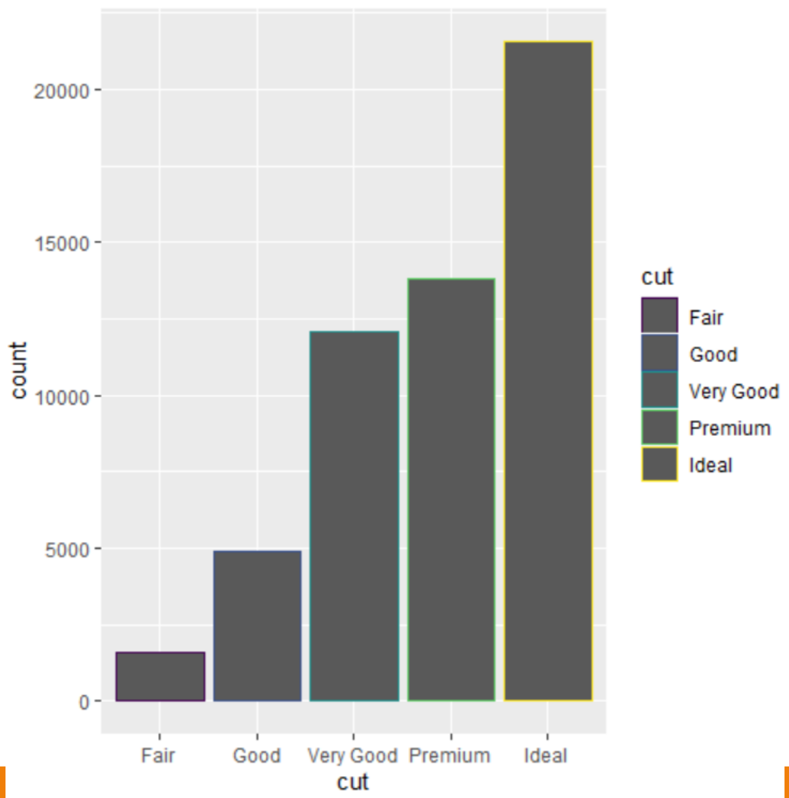


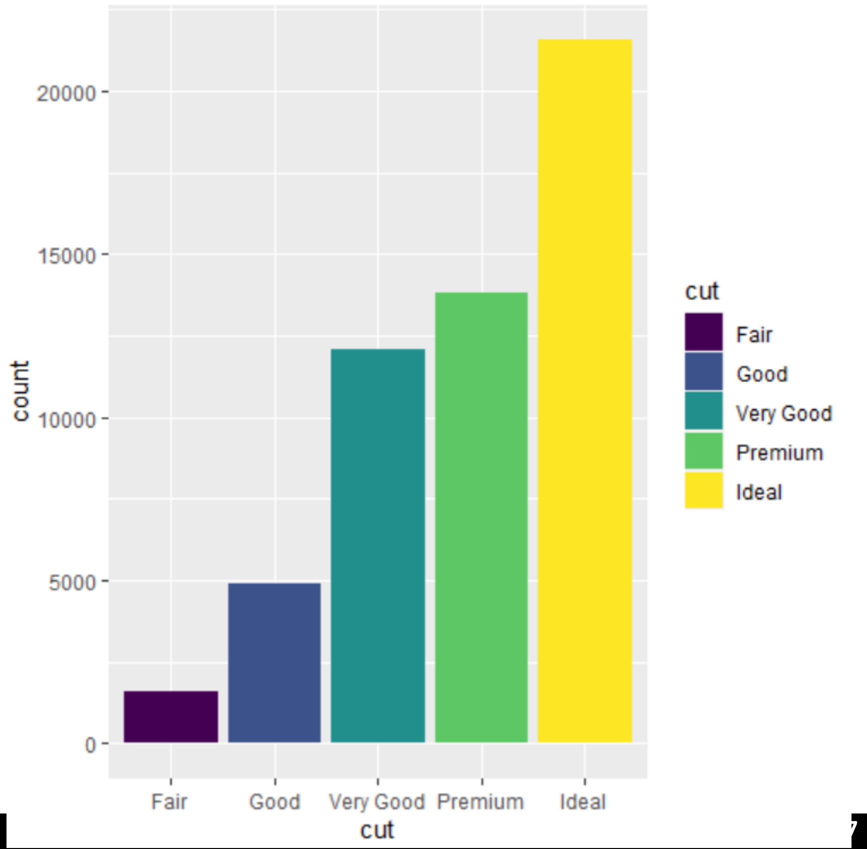
Position Adjustments

- You can color a bar chart using either the color aesthetic, or more usefully, fill:

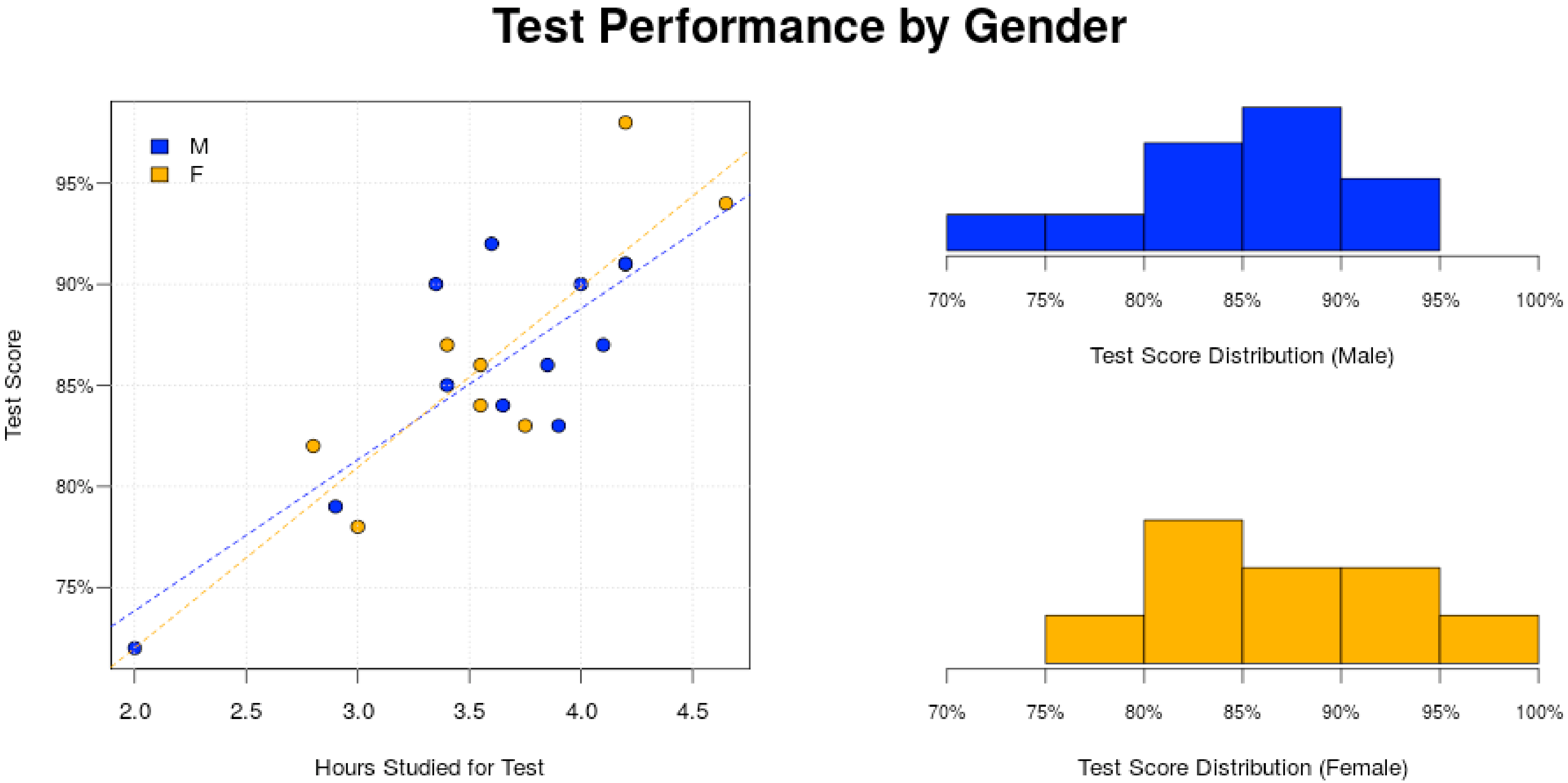
```
ggplot(data = diamonds) +
geom_bar(mapping = aes(x = cut, color = cut))
```

```
ggplot(data = diamonds) +
geom_bar(mapping = aes(x = cut, fill = cut))
```

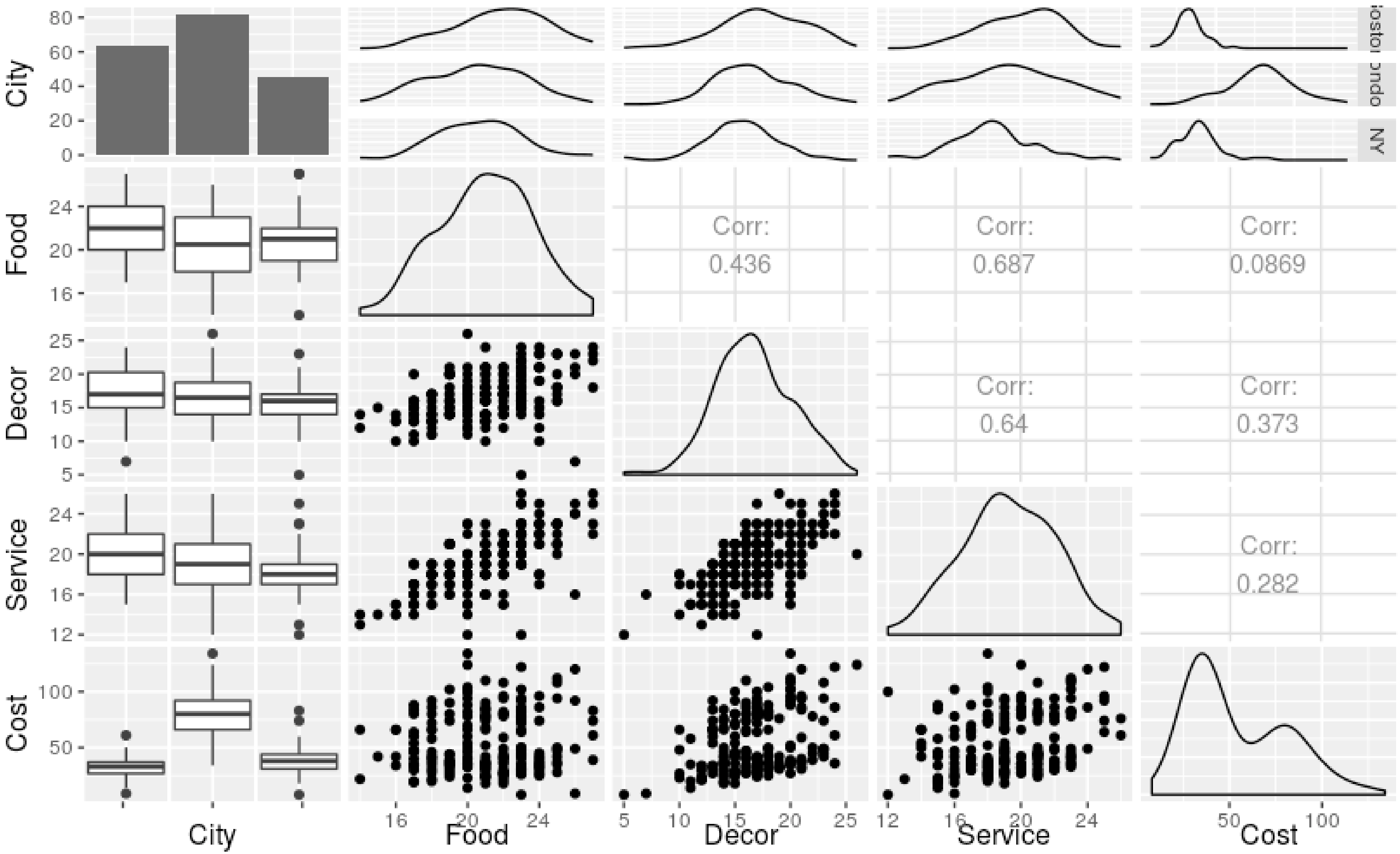




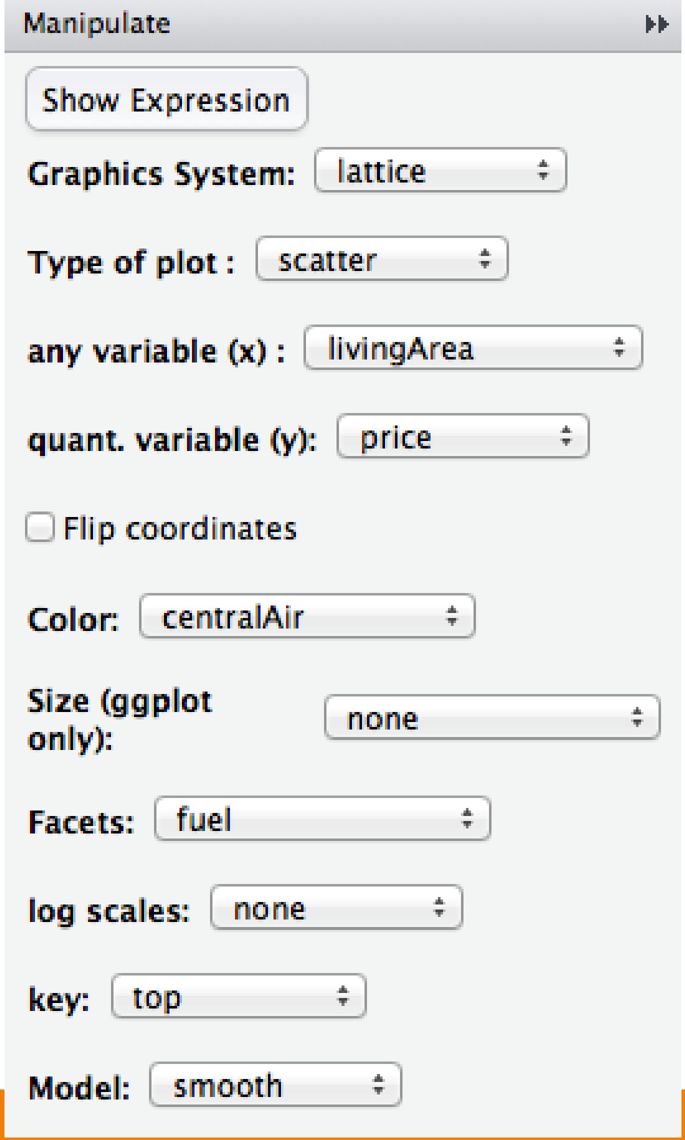
A few simple graphs using the ggplot2 package



An example of graphing using the GGally package in R



The mosaic package mPlot() command makes graphing easy.



Graphics System: lattice

Type of plot : scatter

any variable (x) : livingArea

quant. variable (y): price

☐ Flip coordinates

Color: centralAir

Size (ggplot only): none

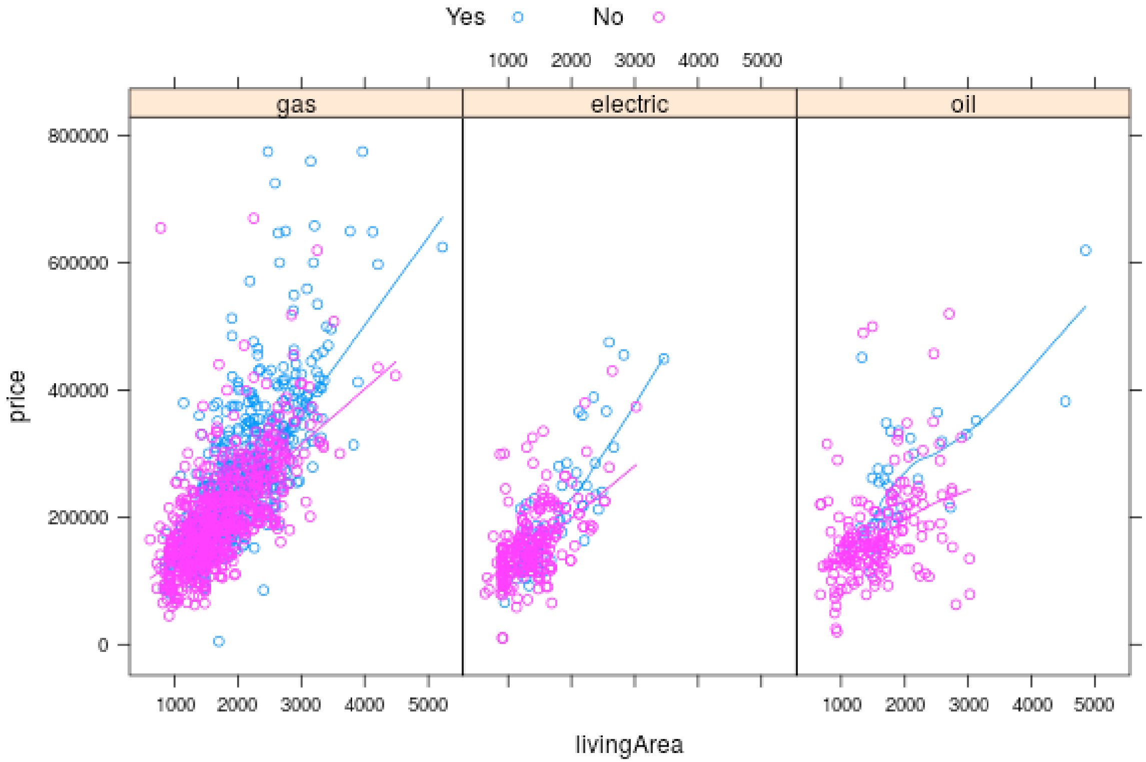
Facets: fuel

log scales: none

key: top

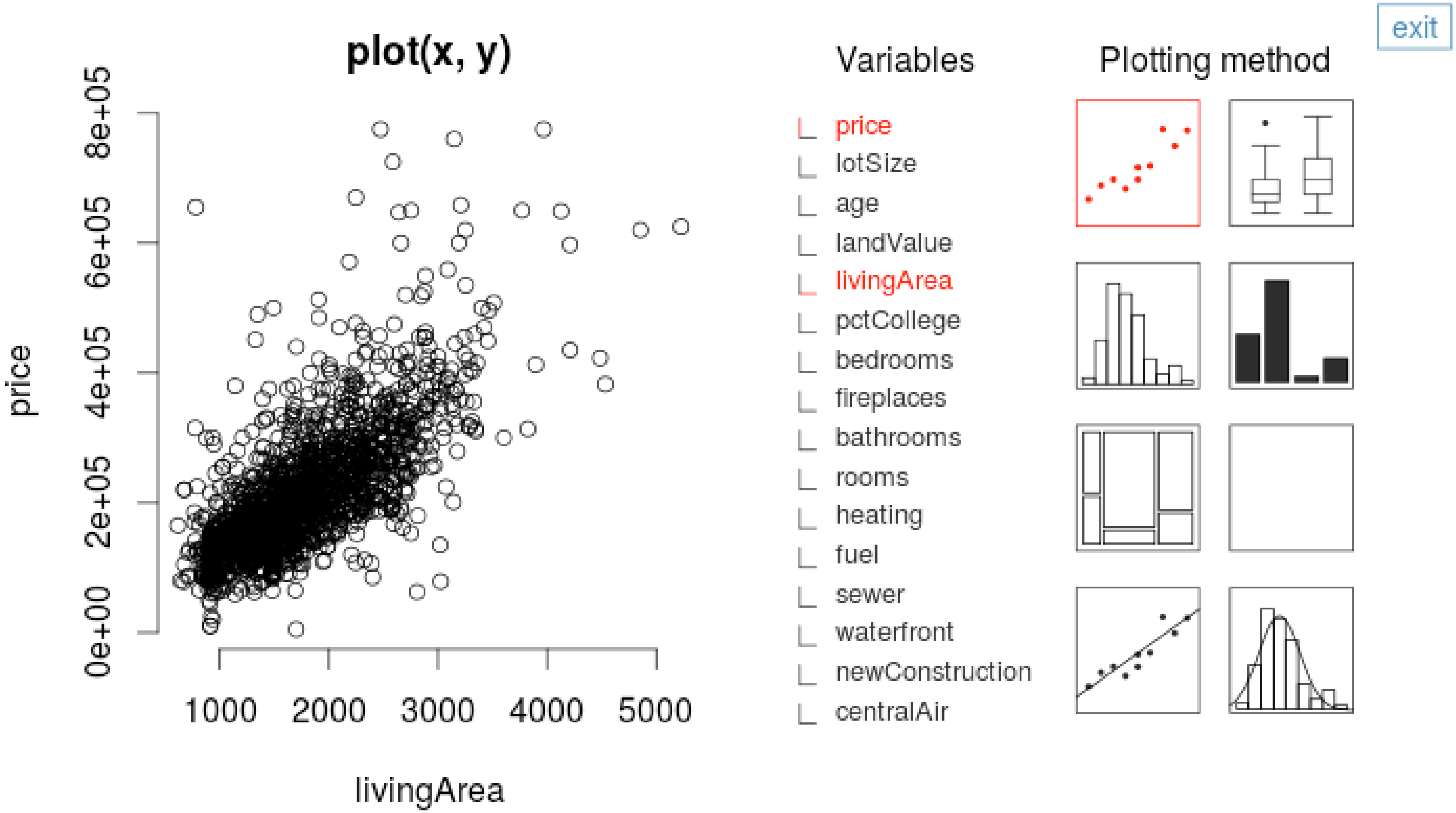
Model: smooth

mPlot(SaratogaHouses)



The openintro package edaPlot() command makes exploring data graphically easy to do

```
edaPlot(SaratogaHouses)
```



The End