R for Data Science - 2

BS3033 Data Science for Biologists

Dr Wilson Goh
School of Biological Sciences

Learning Objectives

By the end of this topic, you should be able to:

- Read the R syntax.
- Use the data structures/ objects of R.
- Deploy programming concepts in the syntax (for example, recursion, loops, variable assignment etc.).

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Importing Data into R

BS3033 Data Science for Biologists

Dr Wilson Goh
School of Biological Sciences

Importing Data (Excel)

Quite frequently, the sample data is in Excel format, and needs to be imported into R prior to use. For this, we can use the function read.xls from the gdata package. It reads from an Excel spreadsheet and returns a data frame. The following shows how to load an Excel spreadsheet named "mydata.xls". This method requires Perl runtime to be present in the system.

```
> library(gdata)                    # load gdata package
> help(read.xls)                    # documentation
> mydata = read.xls("mydata.xls")   # read from first sheet
```

Importing Data

The read.table() function is one of the most common ways of loading data into R workspace. Save the following in a text file separated by space as “mydata.txt” with a text editor.

```
100   a1   b1
200   a2   b2
300   a3   b3
400   a4   b4
```

In the R console or as a script, load the data into a data frame called mydata.

```
mydata = read.table("mydata.txt")  # read text file
mydata                             #see contents
```

To find out more about the read.table function and its arguments, type help(read.table).

Importing Data

Another way is to store the data as comma separated values (CSV) format in which case, we may use the read.csv() function.

```
Col1, Col2,Col3
100, a1, b1
200, a2, b2
300, a3, b3
```

Copy and paste the data above in a file named "mydata.csv" with a text editor, we can read the data with the function read.csv.

```
mydata = read.csv("mydata.csv")  # read csv file
mydata
```

The first row of the data file should contain the column names instead of the actual data.

Working Directory

Finally, the code samples above assume the data files are located in the R working directory, which can be found with the function getwd.

```
getwd()             # get current working directory
```

You can select a different working directory with the function setwd(), and thus avoid entering the full path of the data files.

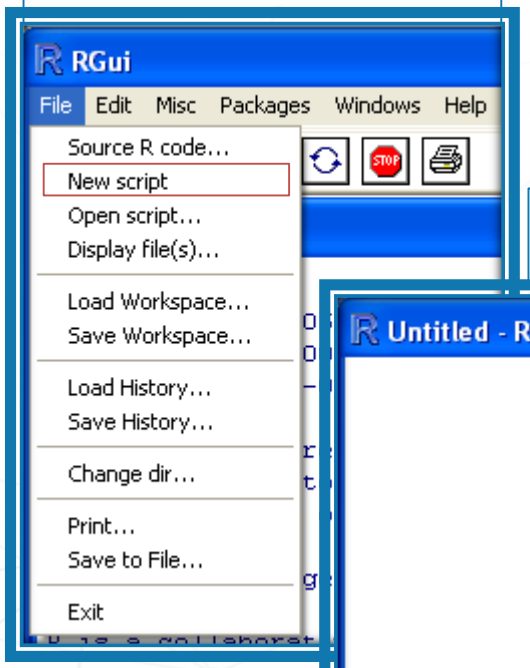
```
setwd("<new path>")   # set working directory
```

Note that the forward slash should be used as the path separator even on Windows platform.

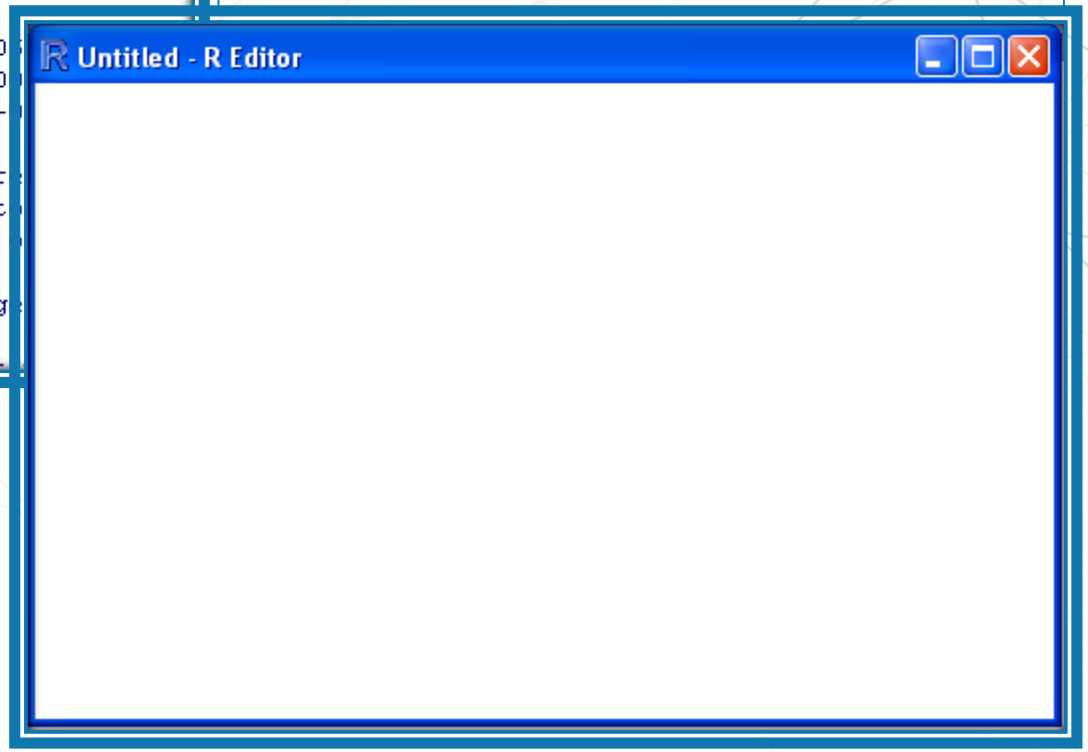
```
setwd("C:/MyDoc")
```

R Scripts

Opening a script.



This gives you a script window.



NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Basic Data Representation with R

BS3033 Data Science for Biologists

Dr Wilson Goh
School of Biological Sciences

Graphs

A graph communicates visual information about the data. It is an integral part of descriptive statistical analysis because it allows us to visualise the information in various ways, allowing us to draw insights.

Complements summary statistics (e.g. measures of central tendency and dispersion).

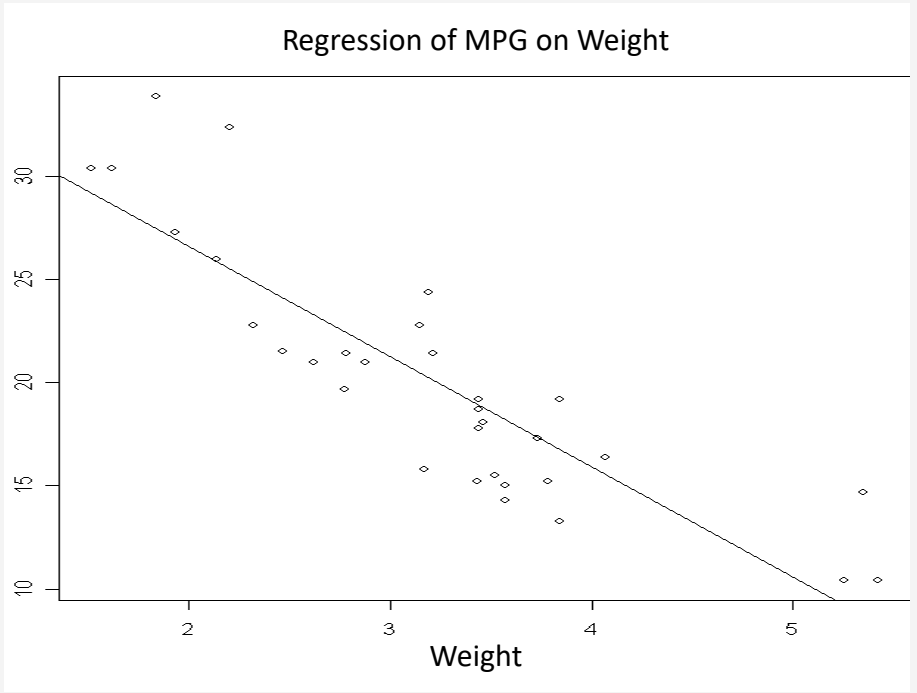
Sometimes summary statistics are too simple.

Graphs in R

In R, graphs are typically created interactively.

```
# Creating a Graph
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
```

The plot( ) function opens a graph window and plots weight vs. miles per gallon. The next line of code adds a regression line to this graph. The final line adds a title.

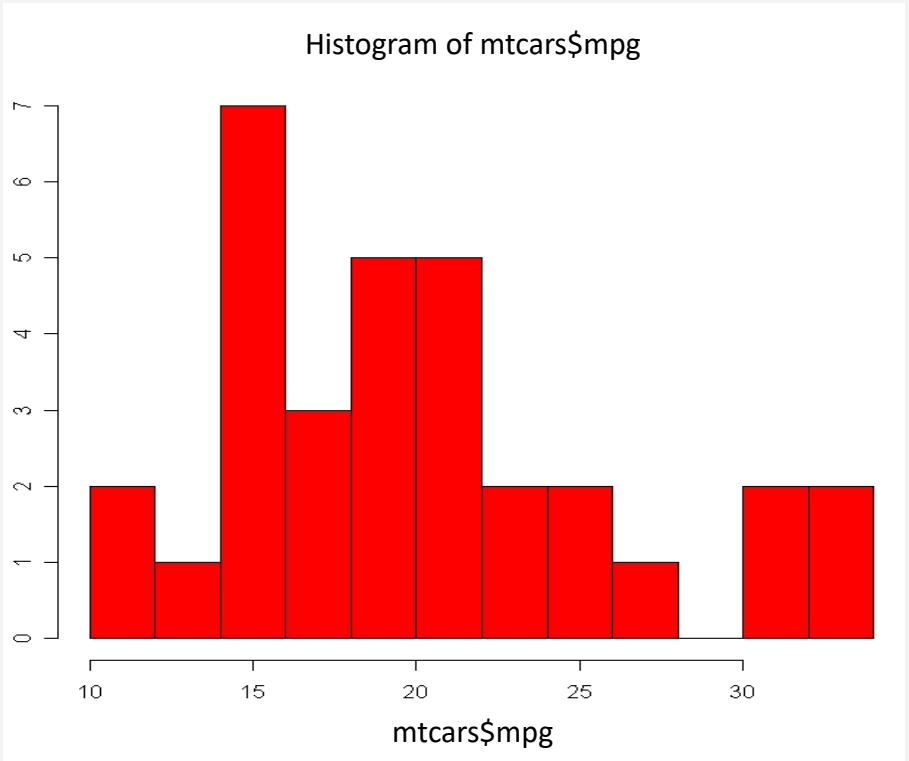


Histograms

A histogram shows the frequency distribution of a given variable.

You can create histograms with the function hist(x) where x is a numeric vector of values to be plotted.

# Colored Histogram with Different Number of Bins
hist(mtcars$mpg, breaks=12, col="red")



Histograms

Disadvantages

- Can only be used with continuous data.
- Cannot be used for easily comparing two datasets.
- The shape of the histogram may change depending on the interval of each bin(this can be changed but hard to optimise).
- Only looks at one variable at a time.

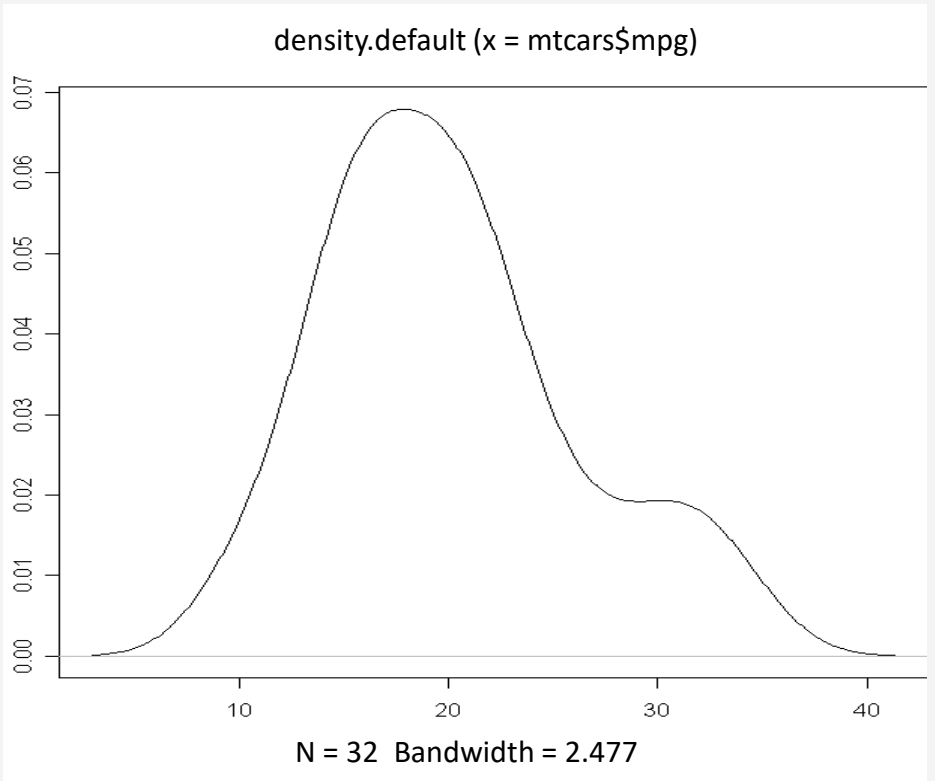
Advantages

- Simple summary showing the shape of the distribution for a large dataset.

Kernal Density Plots

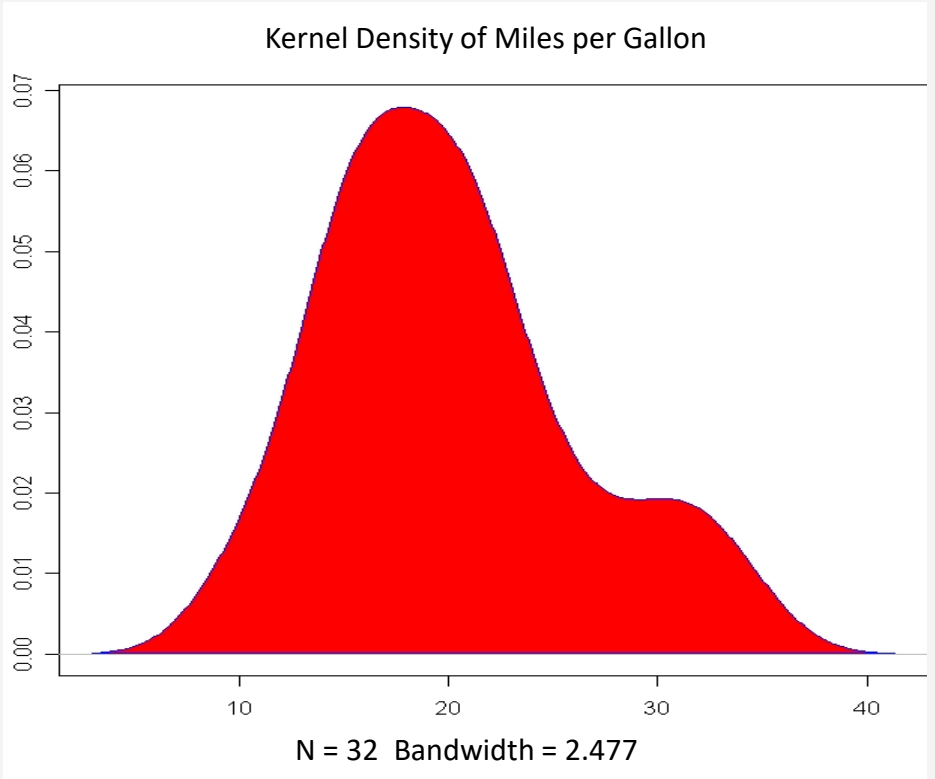
Kernal density plots are usually a much more effective way to view the distribution of a variable.
Create the plot using plot(density(x)) where x is a numeric vector.

```
# Kernel Density Plot
d <- density(mtcars$mpg) # returns the density data
plot(d) # plots the results
```



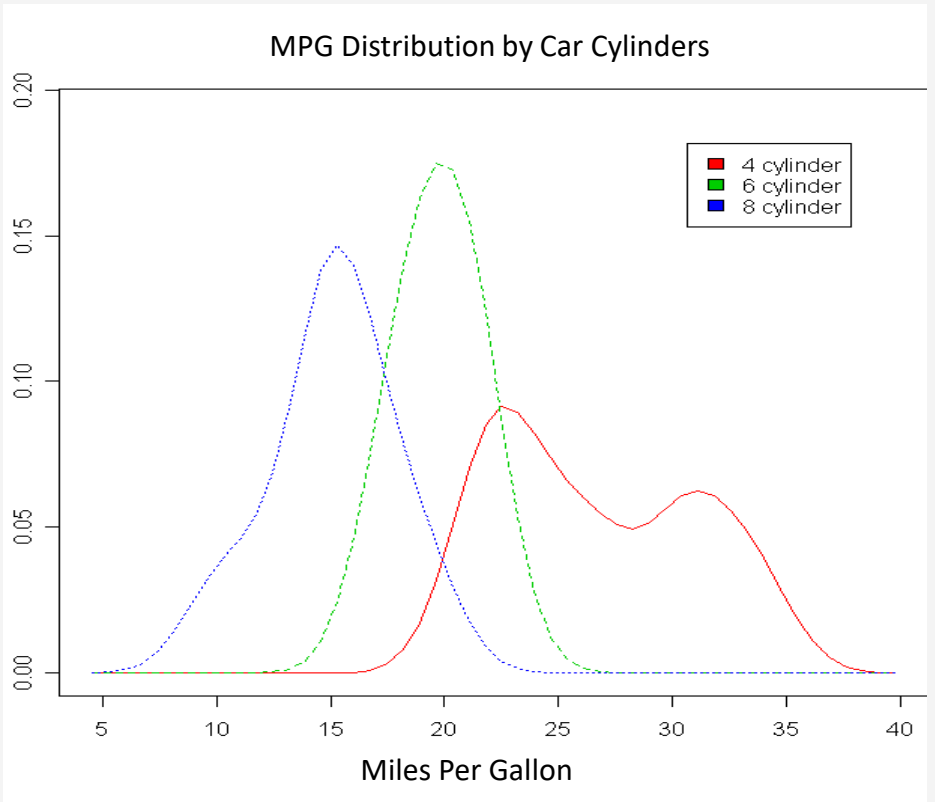
Kernal Density Plots

```
# Filled Density Plot
d <- density(mtcars$mpg)
plot(d, main="Kernel Density of Miles Per Gallon")
polygon(d, col="red", border="blue")
```



Comparing Groups Using Kernal Density

The sm.density.compare( ) function in the sm package allows you to superimpose the kernal density plots of two or more groups. The format is sm.density.compare(x, factor) where x is a numeric vector and factor is the grouping variable.



Kernal Density Plots

Disadvantages

- Can only be used with continuous data.
- The kernal density estimation is a smoothing function applied on the data and is inferential (may not be accurate fit to data).

Advantages

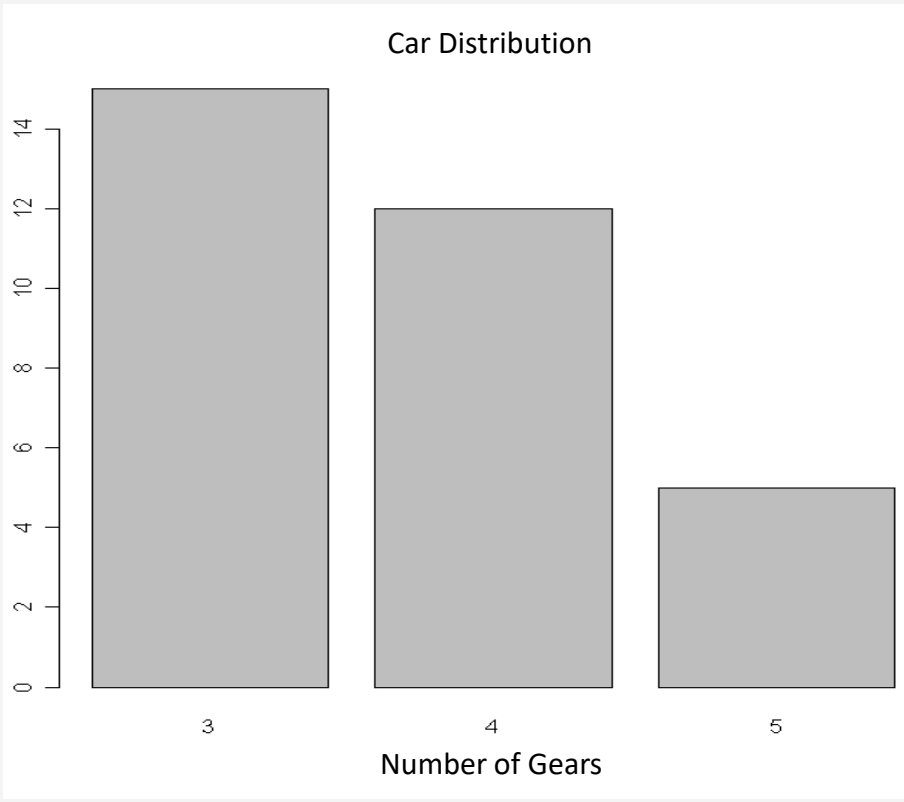
- Shows the shape of the distribution for a large dataset.
- Does not require bin optimisation.
- Can be used for easily comparing multiple datasets.

Barplots

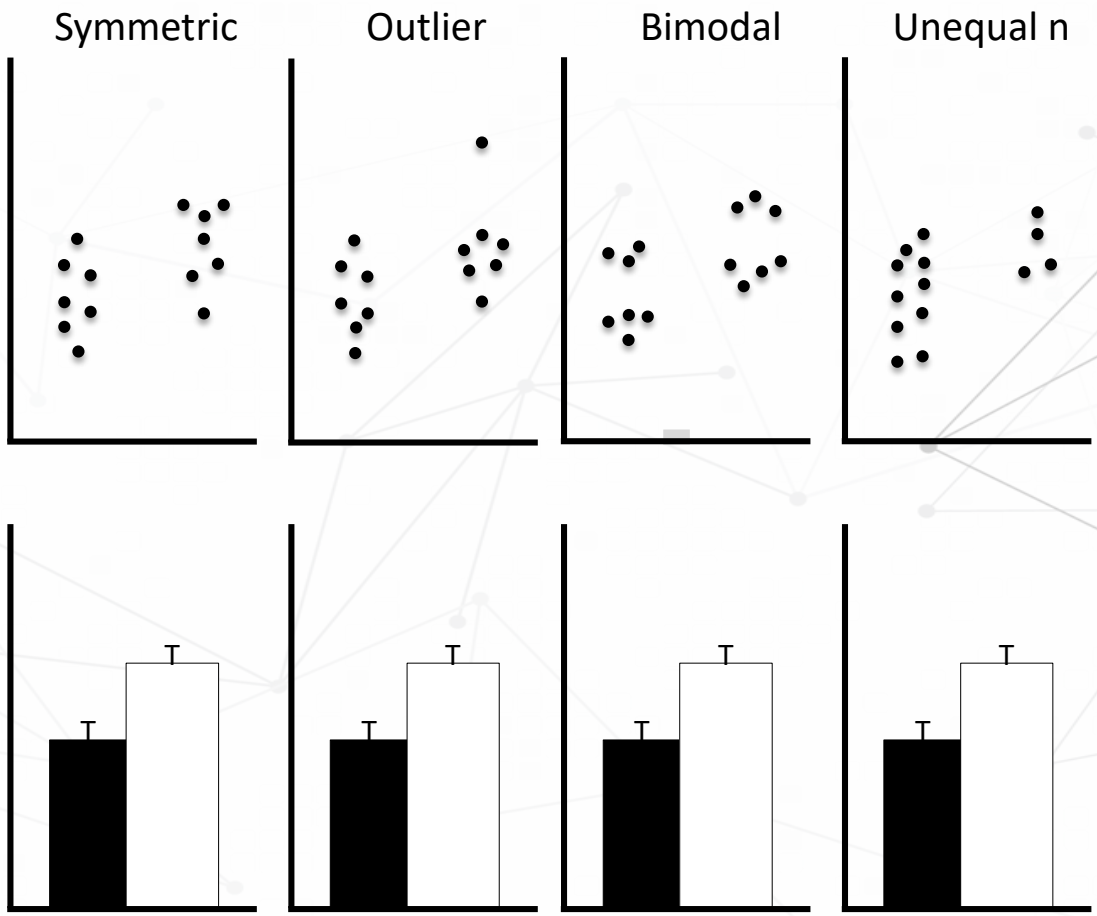
A bar chart or bar graph is a chart that presents grouped data with rectangular bars with lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column bar chart.

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

Very commonly used and easy to do in MS Excel. But did you know...



Consider the Following Data Distribution

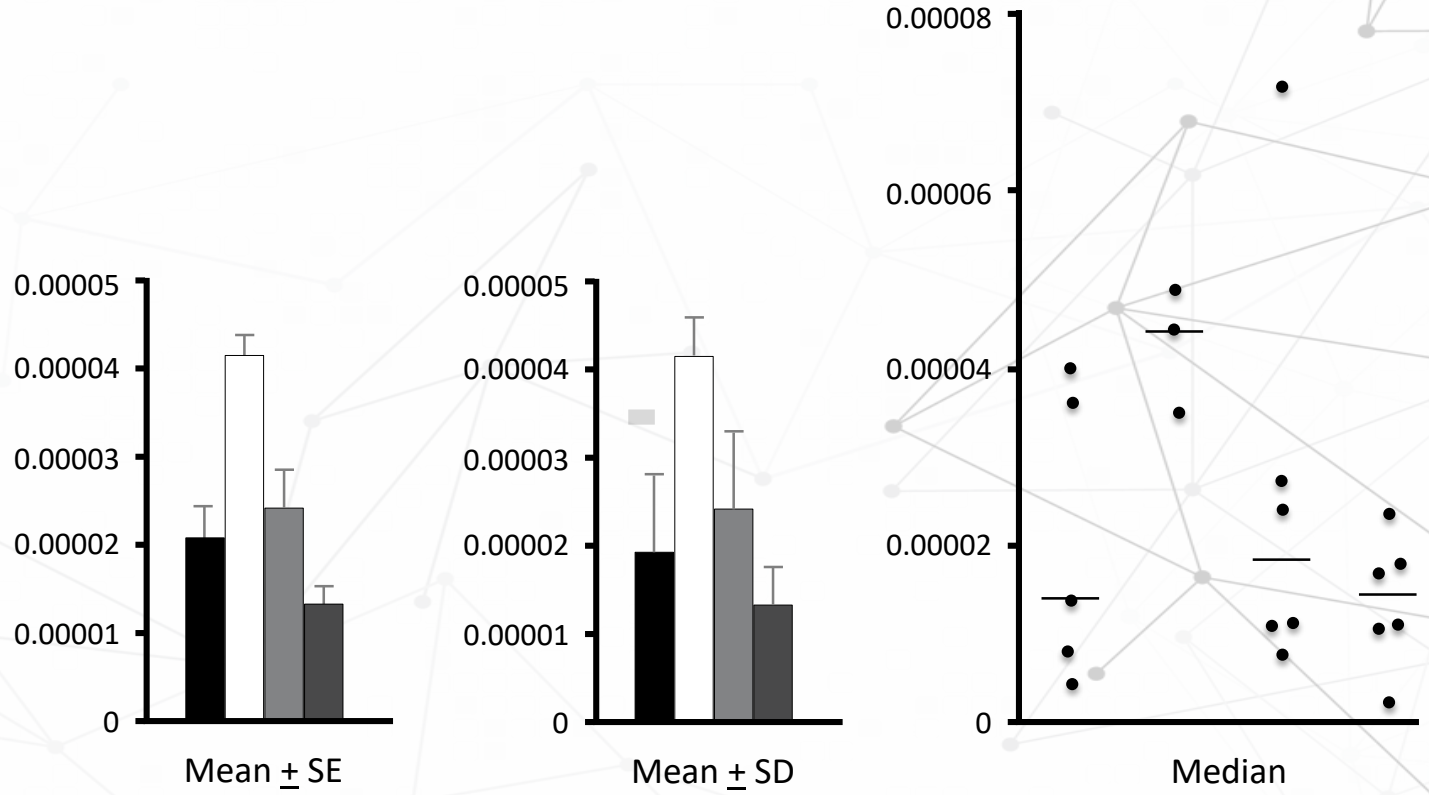


What do you think the barplots look like?

Your barplots are hiding the true data distribution.

Recreated from source: Beyond bar and line graphs: time for a new data presentation paradigm.
Weissgerber TL, Milic NM, Winham SJ, Garovic VD.
PLoS Biol. 2015 Apr 22;13(4):e1002128. doi: 10.1371/journal.pbio.1002128. eCollection 2015 Apr.

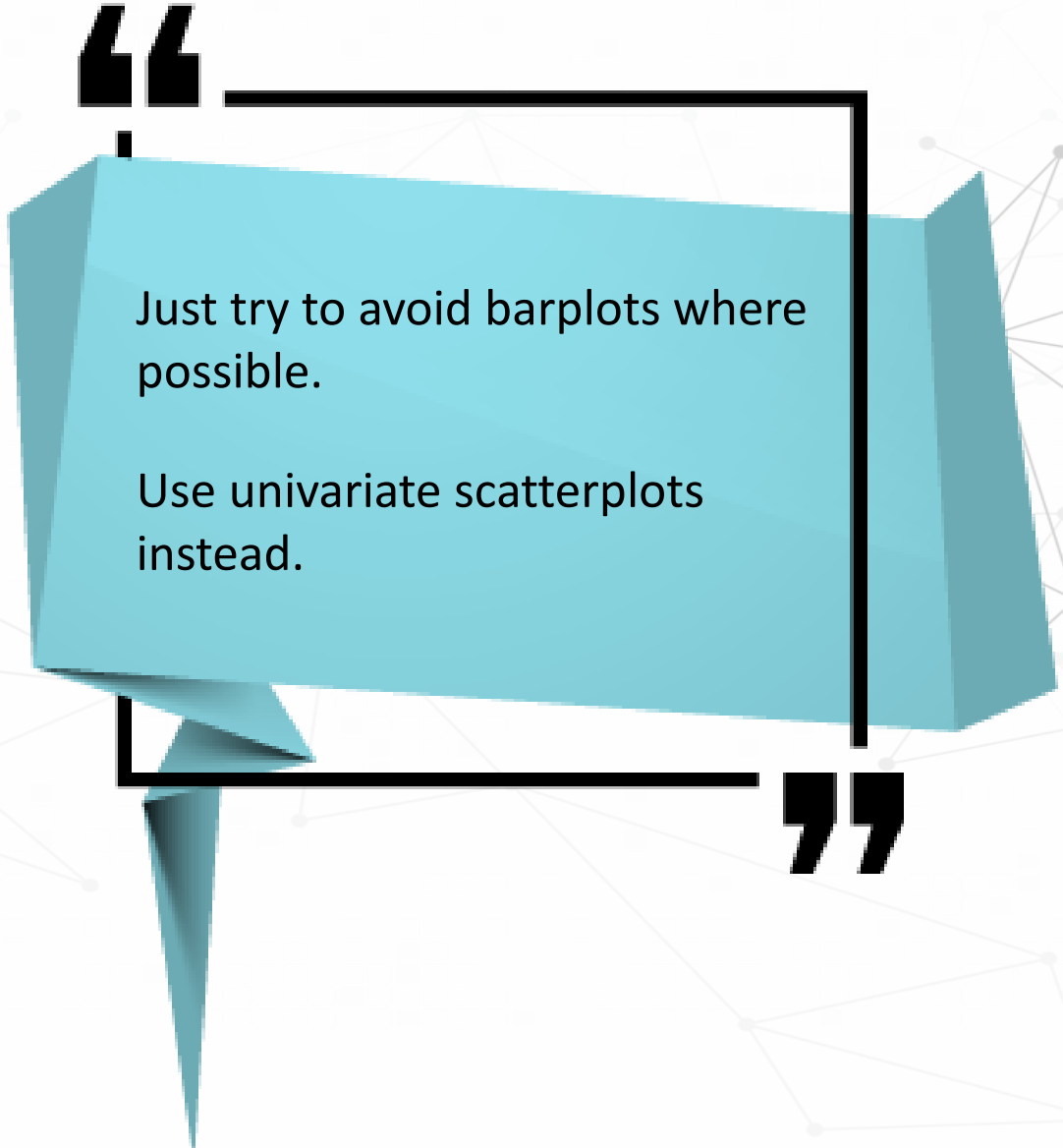
Where is your Data Center?



Again, your barplots are lying to you about your data centers.

Recreated from source: Beyond bar and line graphs: time for a new data presentation paradigm.
Weissgerber TL, Milic NM, Winham SJ, Garovic VD.
PLoS Biol. 2015 Apr 22;13(4):e1002128. doi: 10.1371/journal.pbio.1002128. eCollection 2015 Apr.

Barplots

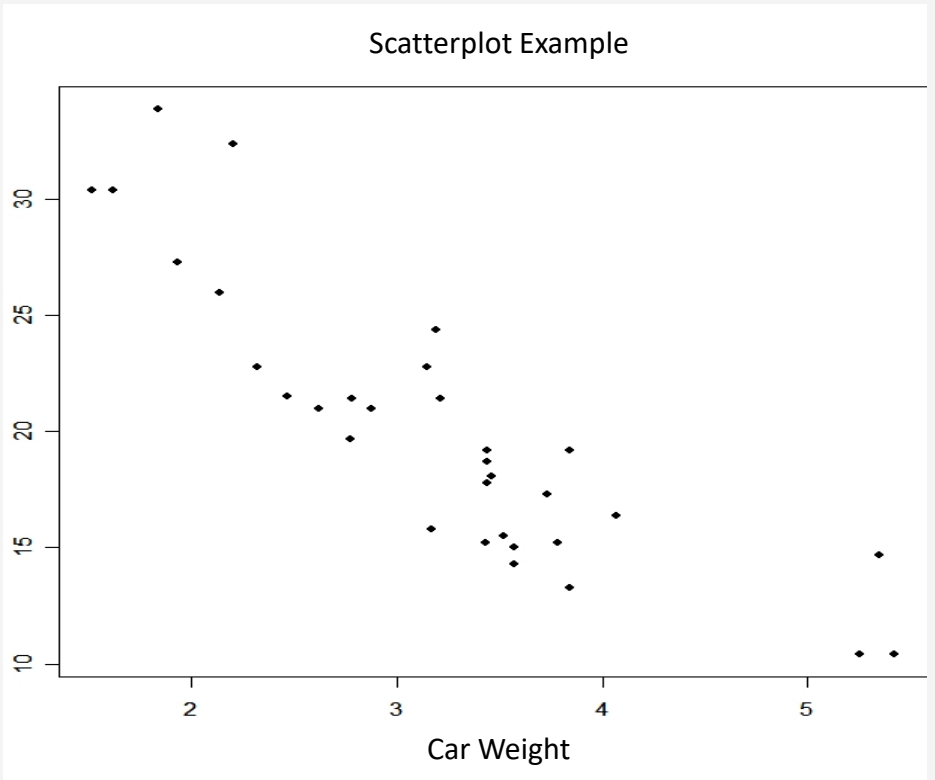

Just try to avoid barplots where possible.
Use univariate scatterplots instead.

Scatterplots

A scatterplot is a graph using Cartesian coordinates (e.g. x and y axis) to display values for typically two variables for a set of data.

They are usually used for checking if two variables are correlated.

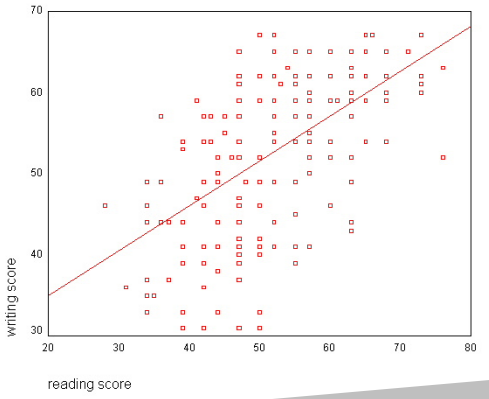
Use the plot function in R, e.g. plot(wt, mpg) after calling attach(mtcars).



Scatterplots

Disadvantages

- Commonly abused to identify or report non-existent relationships between two variables as shown below.



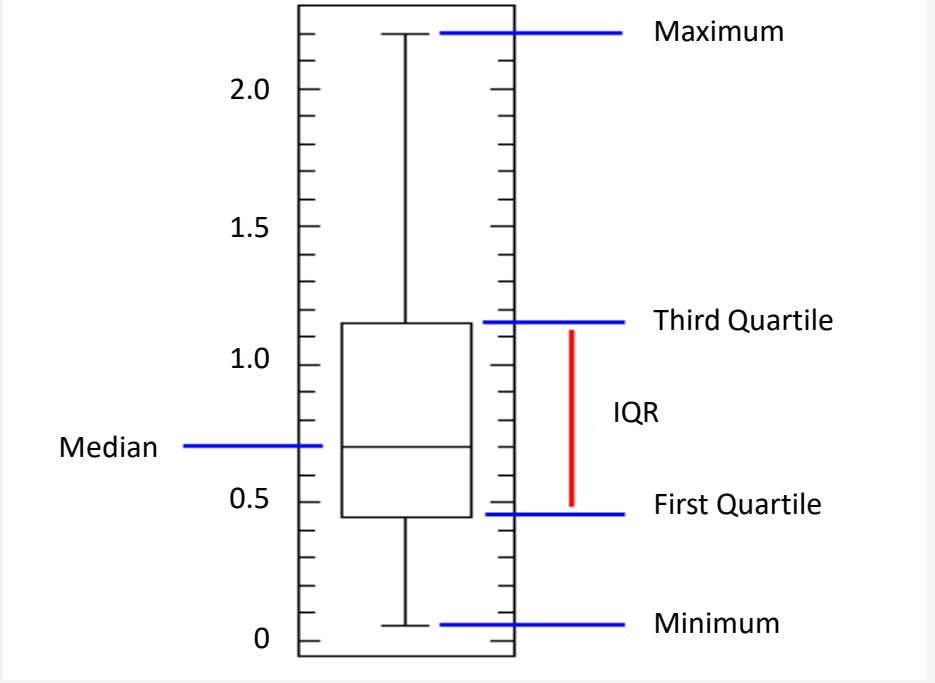
- Simple way of showing the relationship between two variables.

Advantages

Boxplots

The simplest possible box plot displays the full range of variation (from min to max), the likely range of variation (the IQR), and a typical value (the median).

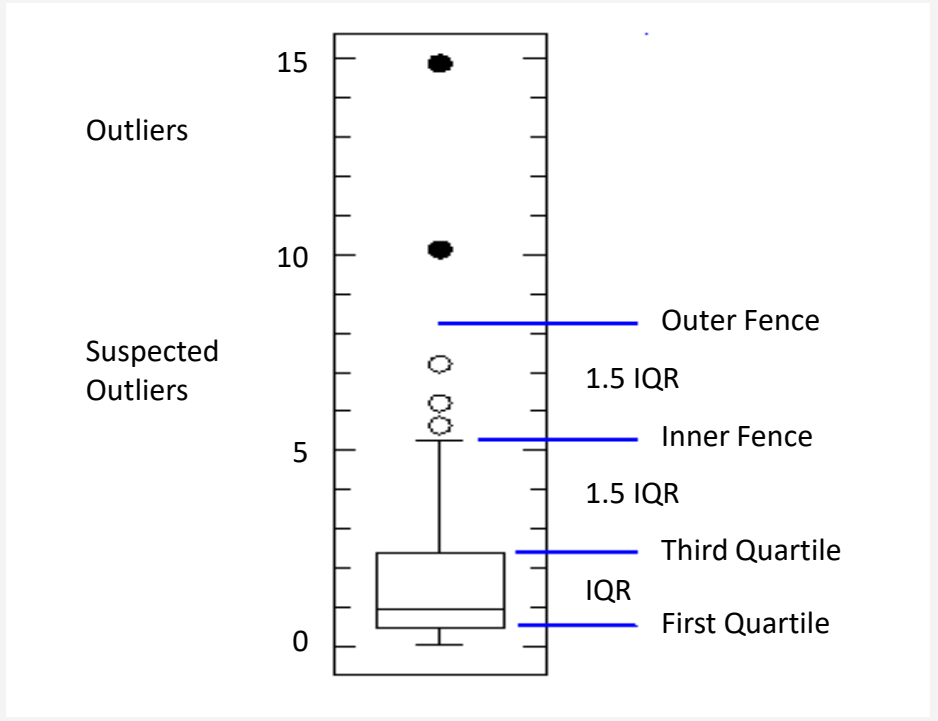
Not uncommonly real datasets will display surprisingly high maximums or surprisingly low minimums called outliers.



Boxplots

The boxplot can also be modified to define outliers, as follows:

- Outliers are either 3×IQR or more above the third quartile or 3×IQR or more below the first quartile.
- Suspected outliers are slightly more central versions of outliers. Either 1.5×IQR or more above the third quartile or 1.5×IQR or more below the first quartile.

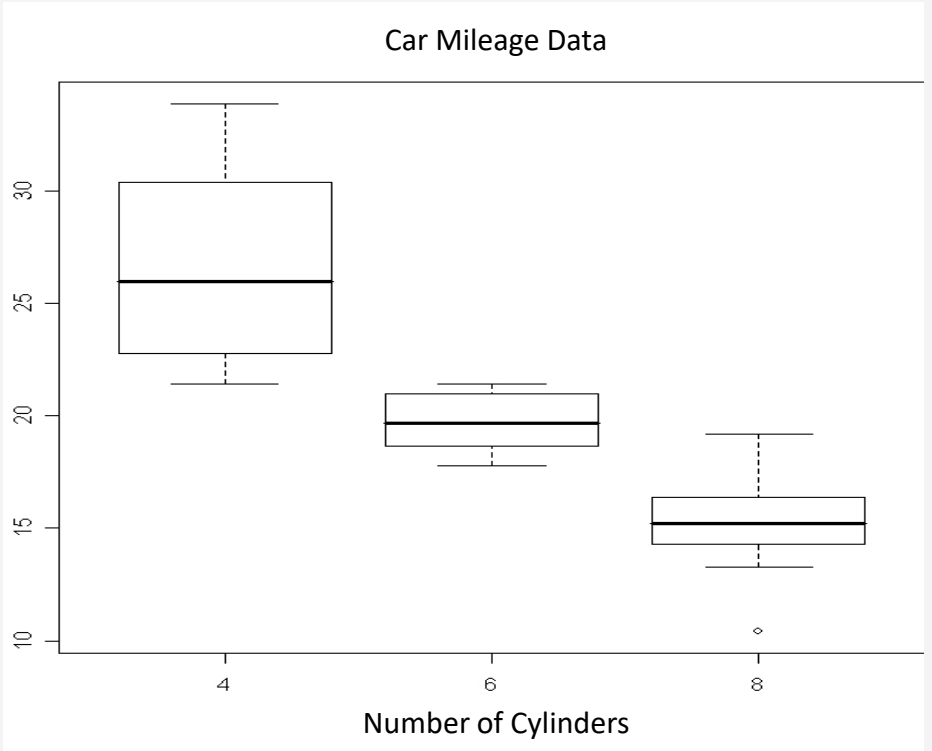


Boxplots

Boxplots can be created for individual variables or for variables by group.

The format is boxplot(x, data=), where x is a formula and data= denotes the data frame providing the data. An example of a formula is y~group where a separate boxplot for numeric variable y is generated for each value of group.

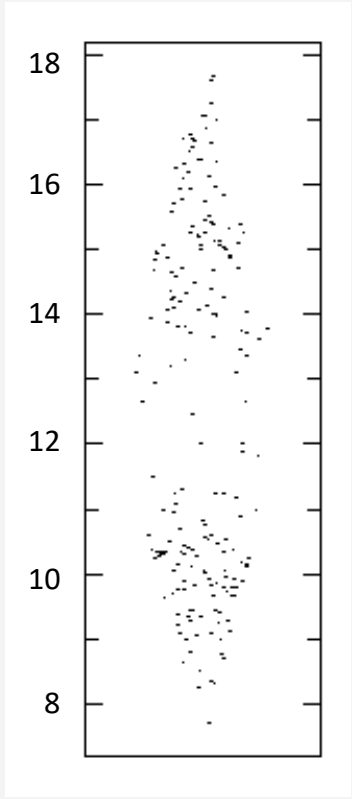
```
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data", xlab="Number of Cylinders", ylab="Miles Per Gallon") # Boxplot of MPG by Car Cylinders
```



Boxplot

Boxplots, like the barplots, can also be deceptive.

In this example, there are obviously two clusters, but the boxplot will not tell you this information.



Boxplots

Disadvantages

- May hide true data distribution (e.g. bimodal data).
- Use with a histogram may be helpful.

Advantages

- Handles large data easily.
- Easy inference of the median, IQR, and potential outliers.

Some Good Analytical Practices (GAPs)

For **small sample size** (< 5), summary statistics are not meaningful. Use scatterplots.

Use the **median** rather than the mean to identify the center of your data.

Never apply **statistical tests** before checking the data distribution.

Check the **actual distribution** of individual data points (do not skip right to summary statistics).

Always check for **outliers, non-symmetry, hidden subpopulations**, and handle them accordingly.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Summary

BS3033 Data Science for Biologists

Dr Wilson Goh
School of Biological Sciences

Key Takeaways from this Topic

1. R graphics are powerful. But be mindful that each graphics has advantages and disadvantages. One should be mindful in using the appropriate graphic to meet the requirements of the task at hand.