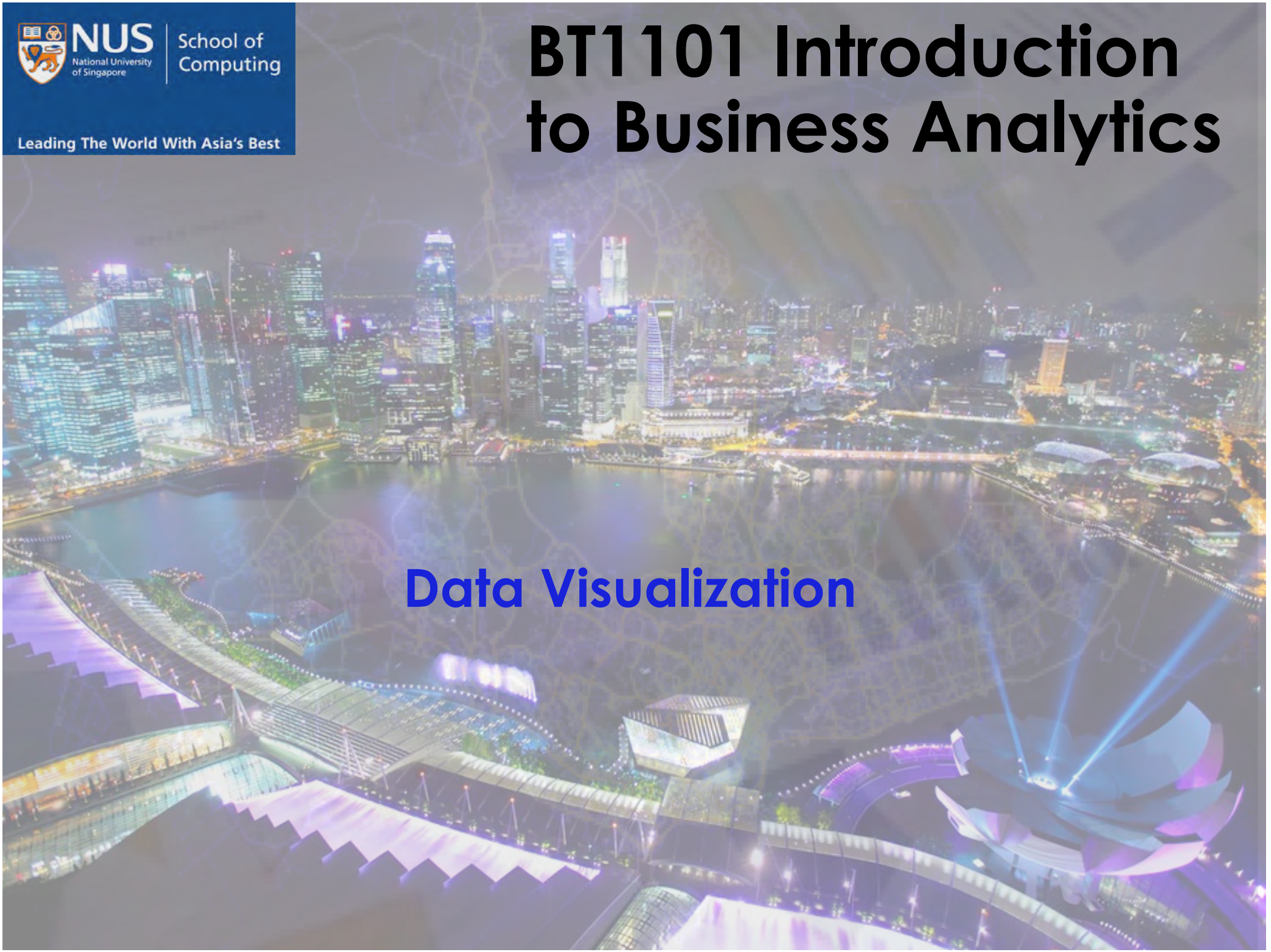
NUS National University of Singapore | School of Computing

Leading The World With Asia's Best

BT1101 Introduction to Business Analytics



Data Visualization

Learning objectives

- Appreciate the importance and role of data visualization
- Be able to describe and summarize data using tabular and visual techniques & to determine the appropriate charts to use to visualize different types of data
- Understand how to use and be able to construct frequency distributions, relative frequency distributions, histogram and to compute cumulative relative frequencies, percentiles and quartiles for a data set

Descriptive Analytics

- Converting data into information to understand past and current performance
- Extracting data from databases, manipulating and summarizing data (descriptive statistical measures),
- Plotting data on charts & data visualizations

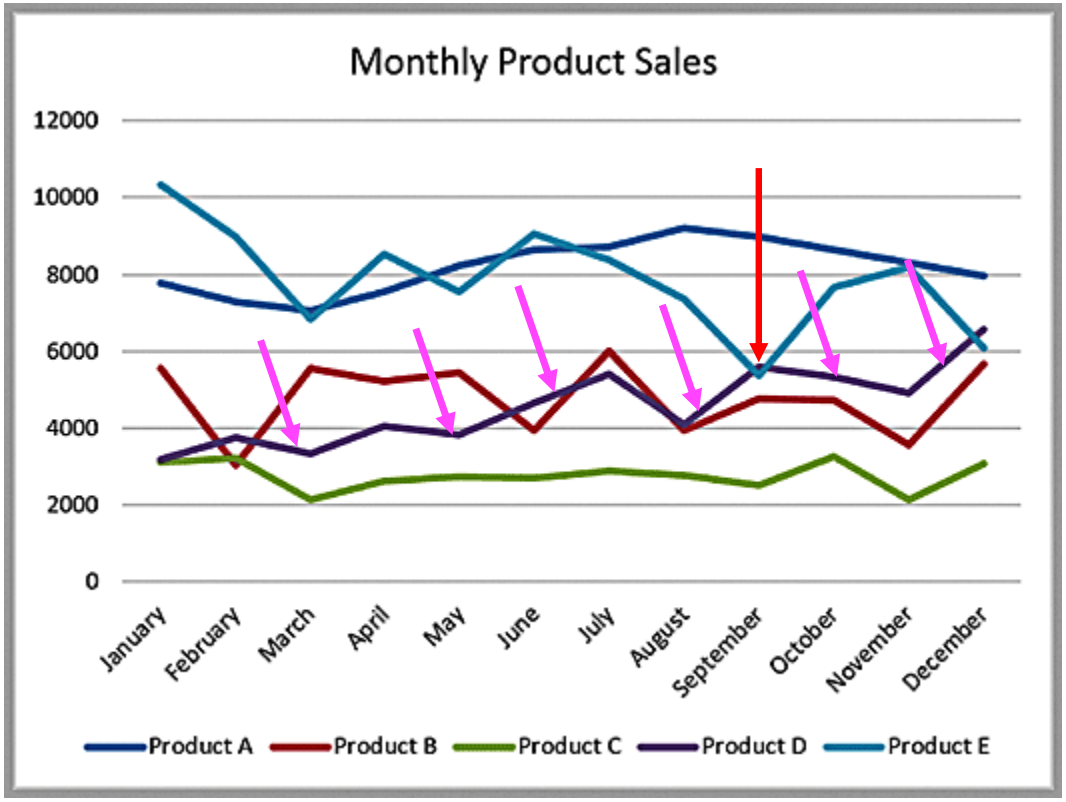
“Why is Data Visualization useful?”

Example of Monthly Product Sales data [tabular form]

	A	B	C	D	E	F
1	Month	Product A	Product B	Product C	Product D	Product E
2	January	7792	5554	3105	3168	10350
3	February	7268	3024	3228	3751	8965
4	March	7049	5543	2147	3319	6827
5	April	7560	5232	2636	4057	8544
6	May	8233	5450	2726	3837	7535
7	June	8629	3943	2705	4664	9070
8	July	8702	5991	2891	5418	8389
9	August	9215	3920	2782	4085	7367
10	September	8986	4753	2524	5575	5377
11	October	8654	4746	3258	5333	7645
12	November	8315	3566	2144	4924	8173
13	December	7978	5670	3071	6563	6088

- How many products were sold in June? 29011
- Which product has least sales in December? C
- Which two products have highest monthly sales overall throughout the year? A & E

Visualization of Monthly Product Sales Data



- Easily compare overall sales of different products
  - Products A & E sell most
  - Product C sells least
- Identify trends
  - sales of Product D are increasing
- Identify patterns
  - sales of Product C is relatively stable while sales of Product B fluctuates more over time
- Identify exceptions
  - Product E's sales fell considerably in September

David McCandless said in his TED talk

[data journalist and information designer]

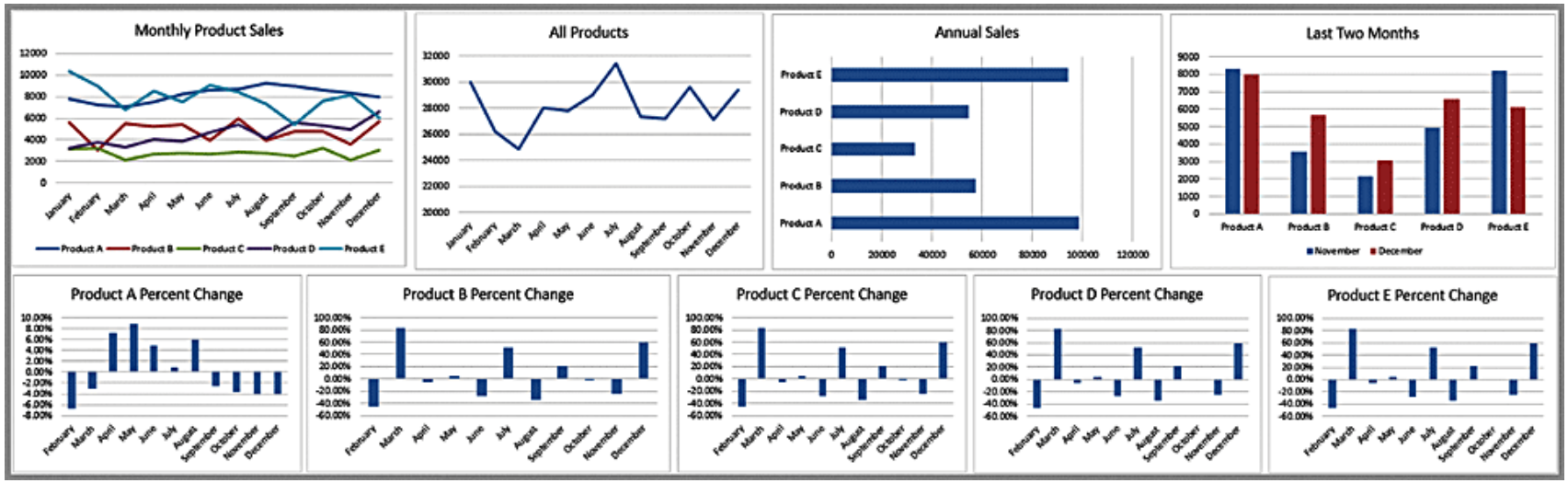
"By visualizing information, we turn it into a landscape that you can explore with your eyes, a sort of information map.

And when you're lost in information, an information map is kind of useful."

Data visualization is the process of displaying data (often in large quantities) in a meaningful fashion to provide insights that will support better decisions.

Dashboard

Dashboard for businesses



- Visual representation of a set of key business measures
- Derived from analogy of an automobile's control panel, which displays speed, gasoline level, temperature, etc.
- Provides important summaries of key business information to help manage a business process or function

Data Visualization

- Pie Charts
- Bar Charts
- Histograms
- Line Charts
- Scatterplots
- Box Plots*

- Area Chart
- Stock chart
- Surface chart
- Doughnut chart
- Radar chart
- Geographic mapping

Visualization Tools

- Excel
- R (basic/ggplot)
- Tableau
- Qlikview
- ChartIO
- For more examples see http://www.creativebloq.com/design-tools/data-visualization-712402

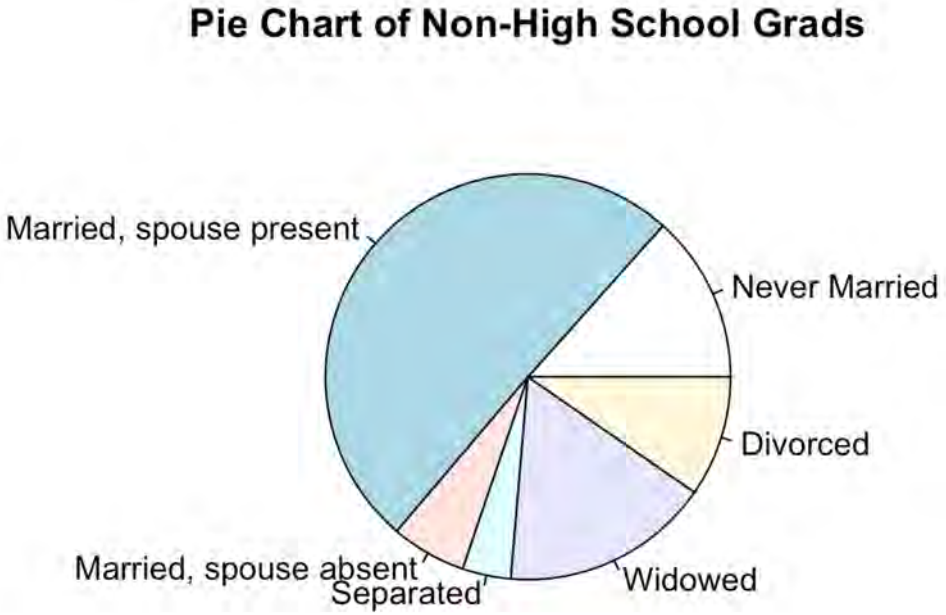
Creating Charts in R

Chart Type	R Functions
Pie Chart	pie(x, labels, radius, main, col, clockwise)
Bar Charts	barplot(H, xlab, ylab, main, names.arg, col)
Box Plots	boxplot(x, data, notch, varwidth, names, main)
Histograms	hist(v, main, xlab, xlim, ylim, breaks, col, border)
Line Graphs	plot(v, type, col, xlab, ylab)
Scatterplots	plot(x, y, main, xlab, ylab, xlim, ylim, axes)

Pie Charts

Census Education Data_pie

	Marital_Status	Not a High School Grad	High School Graduate	Some College No Degree	Associate's Degree	Bachelor's Degree	Advanced Degree
1	Never Married	4120320	7777104	4789872	1828392	5124648	2137416
2	Married, spouse present	15516160	36382720	18084352	8346624	19154432	9523712
3	Married, spouse absent	1847880	2368024	1184012	465392	670712	301136
4	Separated	1188090	1667010	842715	336165	405240	165780
5	Widowed	5145683	4670488	1765010	556657	977544	475195
6	Divorced	2968680	7003040	3806000	1674640	2340690	1217920

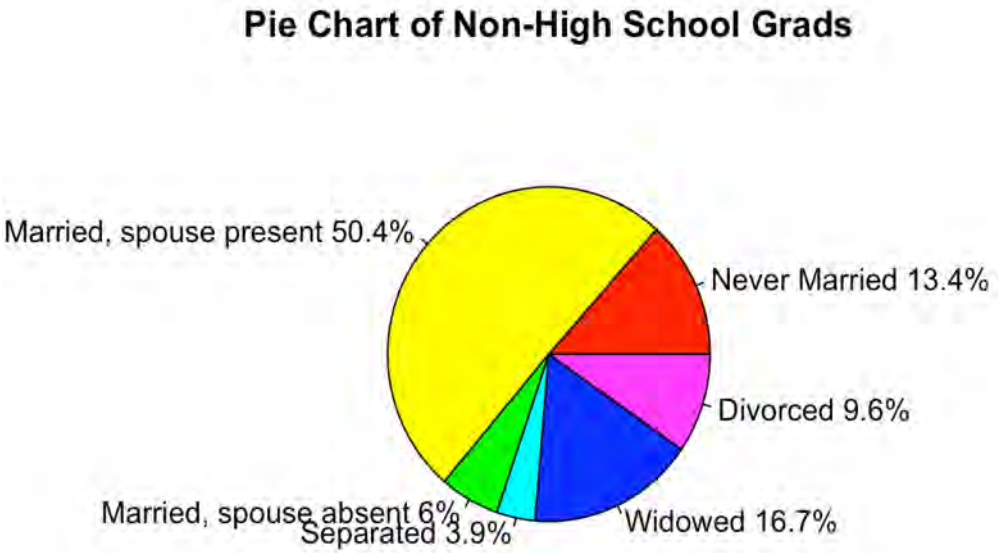



```
> DF1 <- Census_Education_Data_pie
> slices <- DF1$`Not a High School Grad`
> pie(slices,labels=DF1$Marital_Status,main="Pie Chart of Non-High School Grads")
```

- **Pie chart displays relative proportions by partitioning a circle into pie-shaped areas.**
- **How do we interpret this pie chart?**
- **What are the limitations of pie chart?**

Pie Charts

- **Not recommended by data visualization professionals**
- **Difficult to compare relative sizes of areas**



- **When using pie charts**
  - **restrict to small number of categories**
  - **use labels to display category names and actual percentages.**
  - **always ensure numbers add to 100%**
  - **avoid three-dimensional (3-D) pie charts—especially those that are rotated—and keep them simple**

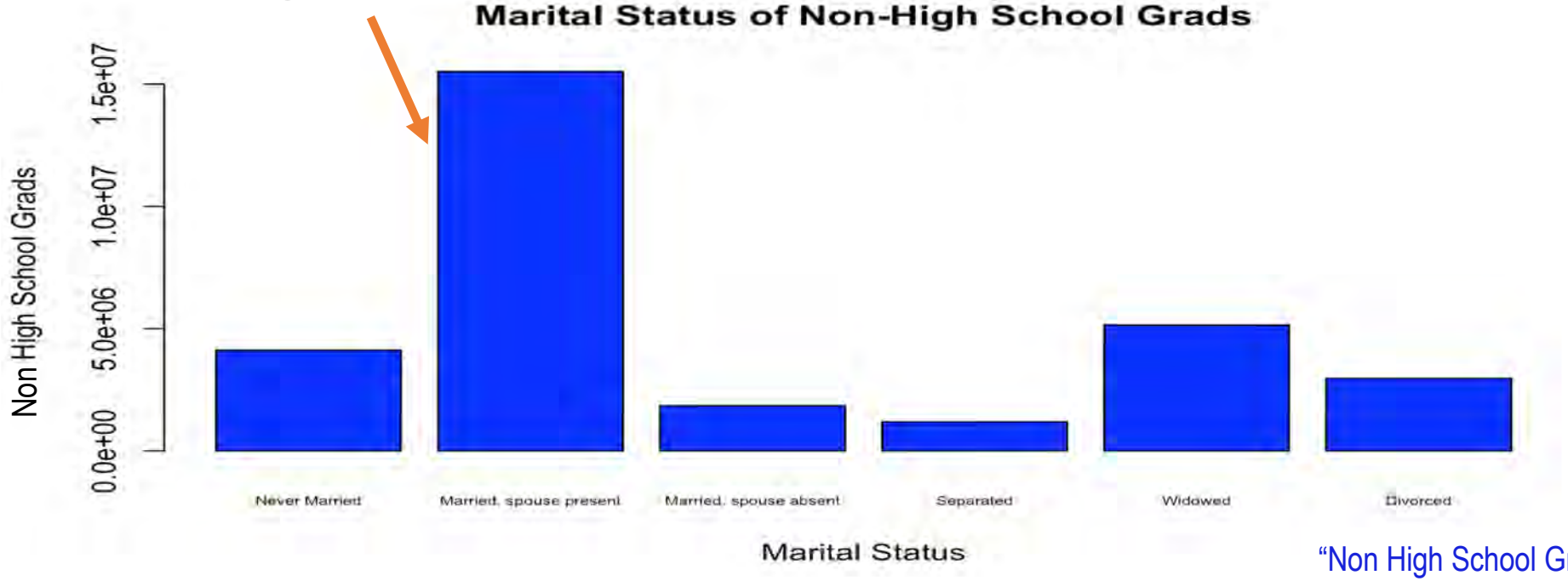
R Script

```
> piepercent <- round(100*DF1$`Not a High School Grad`/sum(DF1$`Not a High School Grad`), 1)
> slices <- DF1$`Not a High School Grad`
> label <- DF1$Marital_Status
> label <- paste (label,piepercent)
> label <- paste(label,"%",sep="")
> pie(slices,labels = label, col=rainbow(length(label)), main="Pie Chart of Non-High School Grads")
```

Bar Charts (bar plot in R)

A **bar chart** represents data in **rectangular** bars with **length** of the bar **proportional** to the **value of the variable**.

Creating a Barplot for Non-High School Grads



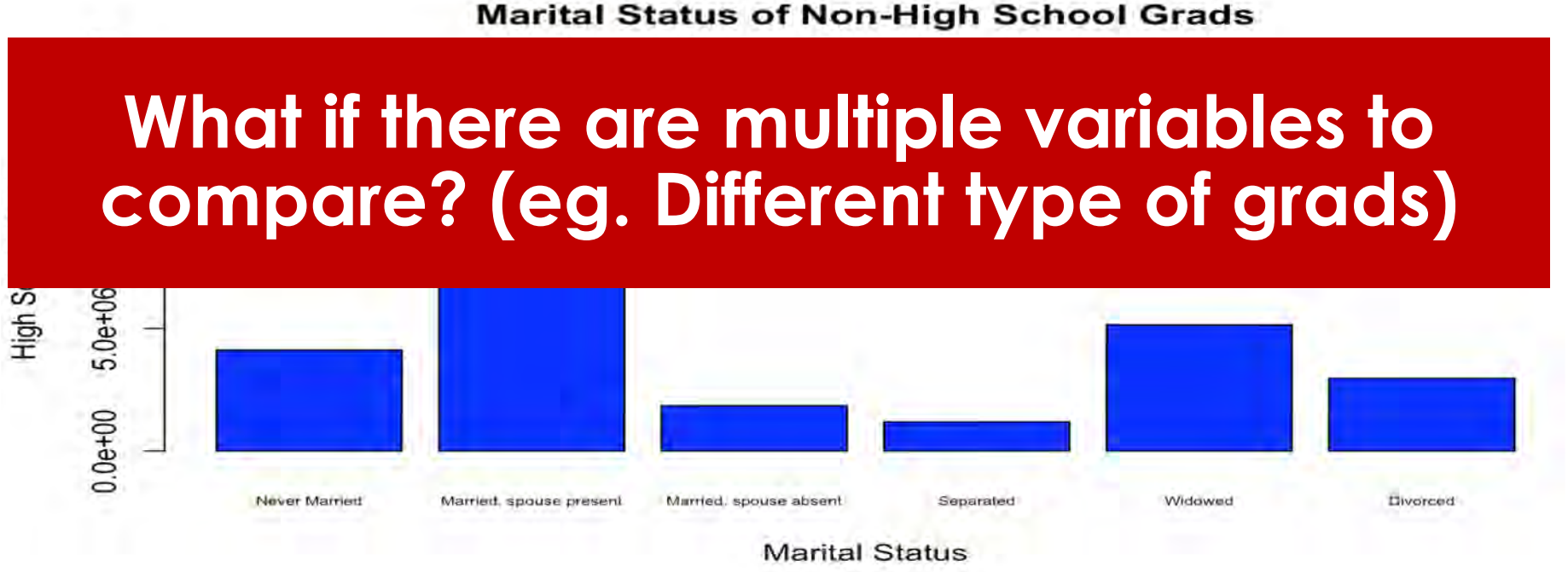
"Non High School Grads",

```
> barplot(DF1$`Not a High
+ School Grad`,names.arg = DF1$Marital_Status, xlab = "Marital Status", ylab = "High School Grads" ,
main="Marital Status of Non-High School Grads",col= "blue", cex.names = 0.5)
```

Bar Charts (bar plot in R)

A **bar chart** represents data in **rectangular** bars with **length** of the bar **proportional** to the **value of the variable**.

Creating a Barplot for Non-High School Grads



What if there are multiple variables to compare? (eg. Different type of grads)

```
> barplot(DF1$`Not a High
+ School Grad`,names.arg = DF1$Marital_Status, xlab = "Marital Status", ylab = "High School Grads" ,
main="Marital Status of Non-High School Grads",col= "blue", cex.names = 0.5)
```

Clustered bar chart (Grouped Barplot in R)

A **clustered bar chart** compares values across categories using vertical rectangles.

Creating Grouped Barplots for Degree Holders

Filter

	Marital_Status	Not a High School Grad	High School Graduate	Some College No Degree	Associate's Degree	Bachelor's Degree	Advanced Degree
1	Never Married	4120320	7777104	4789872	1828392	5124648	2137416
2	Married, spouse present	15516160	36382720	18084352	8346624	19154432	9523712
3	Married, spouse absent	1847880	2368024	1184012	465392	670712	301136
4	Separated	1188090	1667010	842715	336165	405240	165780
5	Widowed	5145683	4670488	1765010	556657	977544	475195
6	Divorced	2968680	7003040	3806000	1674640	2340690	1217920

```
df1 <- Census_Education_Data_pie
# extract columns for degree holders to a new dataframe df2.
df2 <- df1[,c(5,6,7)]
# convert df2 to a matrix called 'value'.
value <- as.matrix(df2)
```

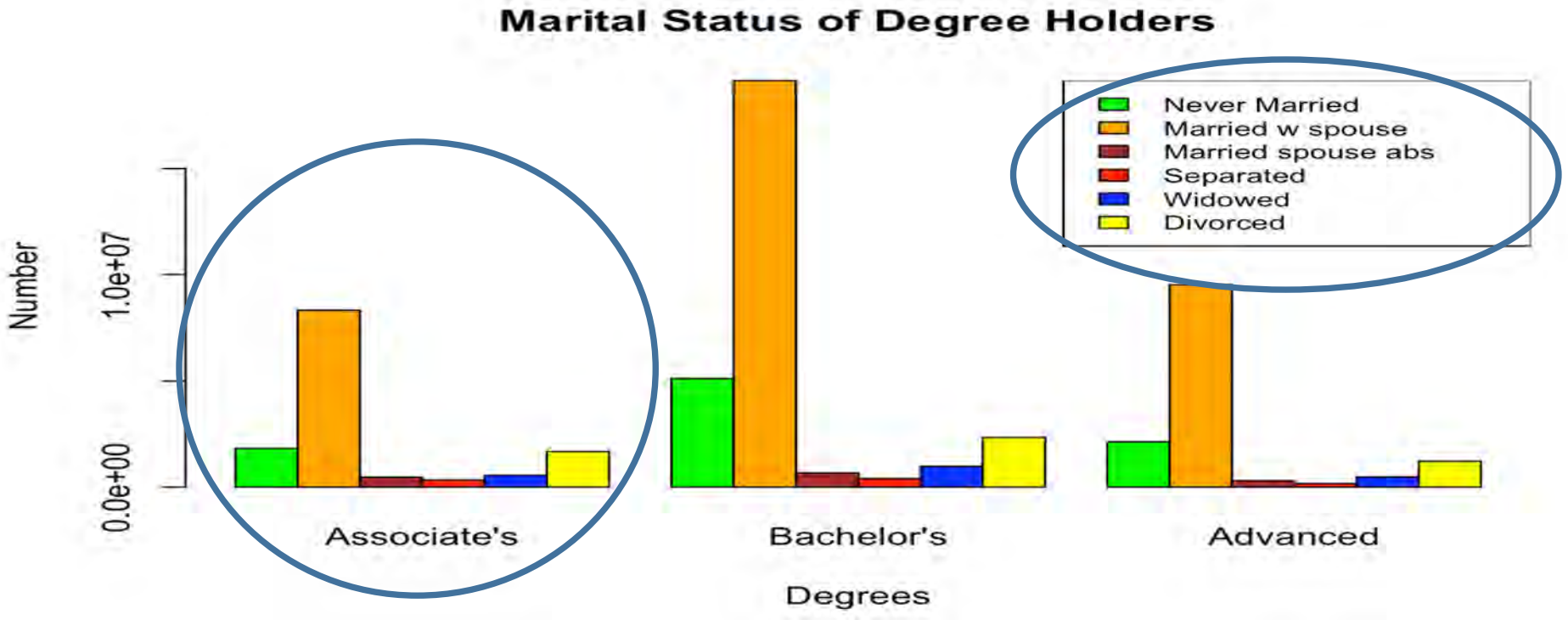
```
> print(value)
     Associate's\r\nDegree Bachelor's\r\nDegree Advanced\r\nDegree
[1,]               1828392                5124648              2137416
[2,]               8346624               19154432              9523712
[3,]                465392                 670712               301136
[4,]                336165                 405240               165780
[5,]                556657                 977544               475195
[6,]               1674640                2340690              1217920
```

Clustered bar chart (Grouped Barplot in R)

Creating Group Barplots for Degree Holders

```
colors <- c("green","orange","brown","red","blue","yellow")
Degrees <- c("Associate's","Bachelor's","Advanced")
MS <- c("Never Married","Married w spouse", "Married spouse abs", "Separated","Widowed","Divorced")

# Create the grouped barplot.
barplot(value,main = "Marital Status of Degree Holders",names.arg = Degrees ,xlab = "Degrees",ylab = "Number",
        col = colors, beside = TRUE)
# Add the legend to the chart.
legend("topright", MS, cex = 0.8, fill = colors)
```

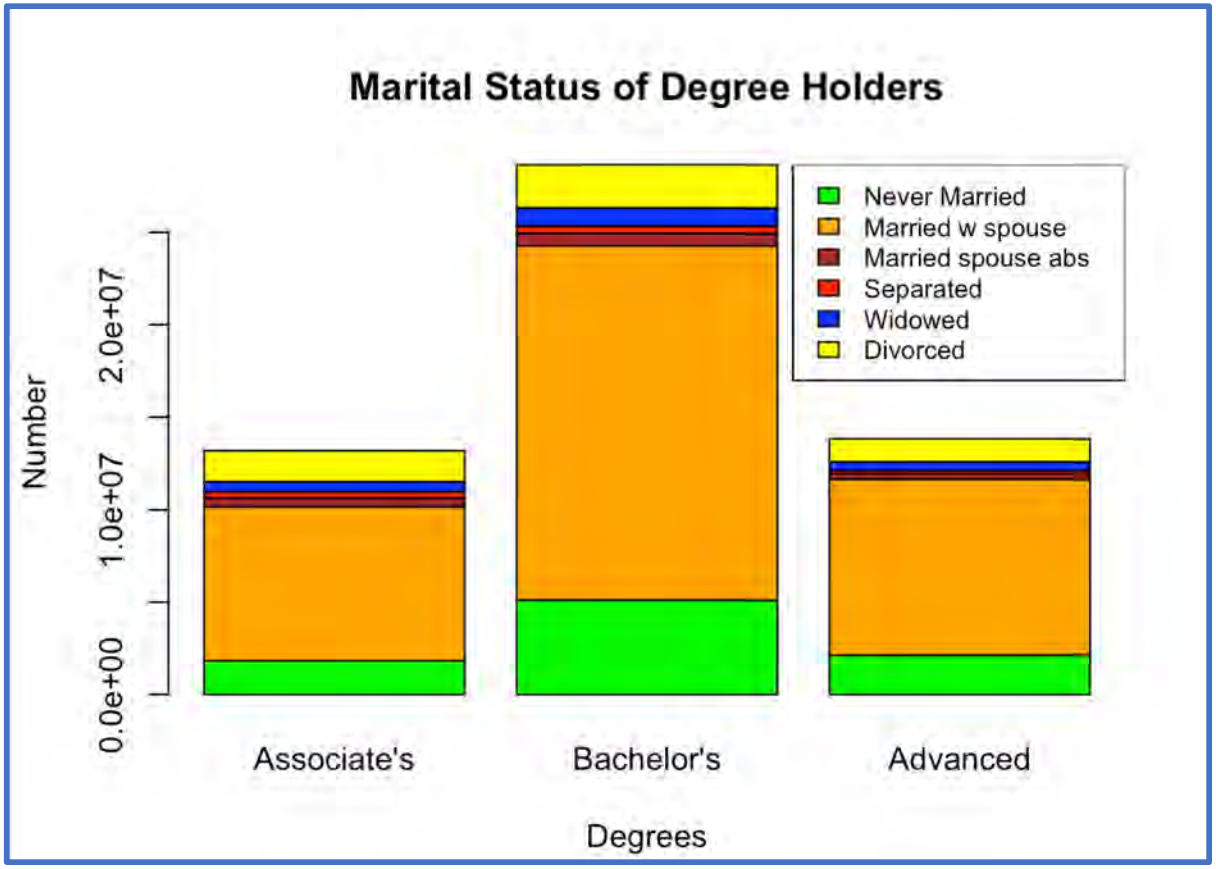
Default for 'beside' is FALSE



Stacked bar chart (Stacked Barplot in R)

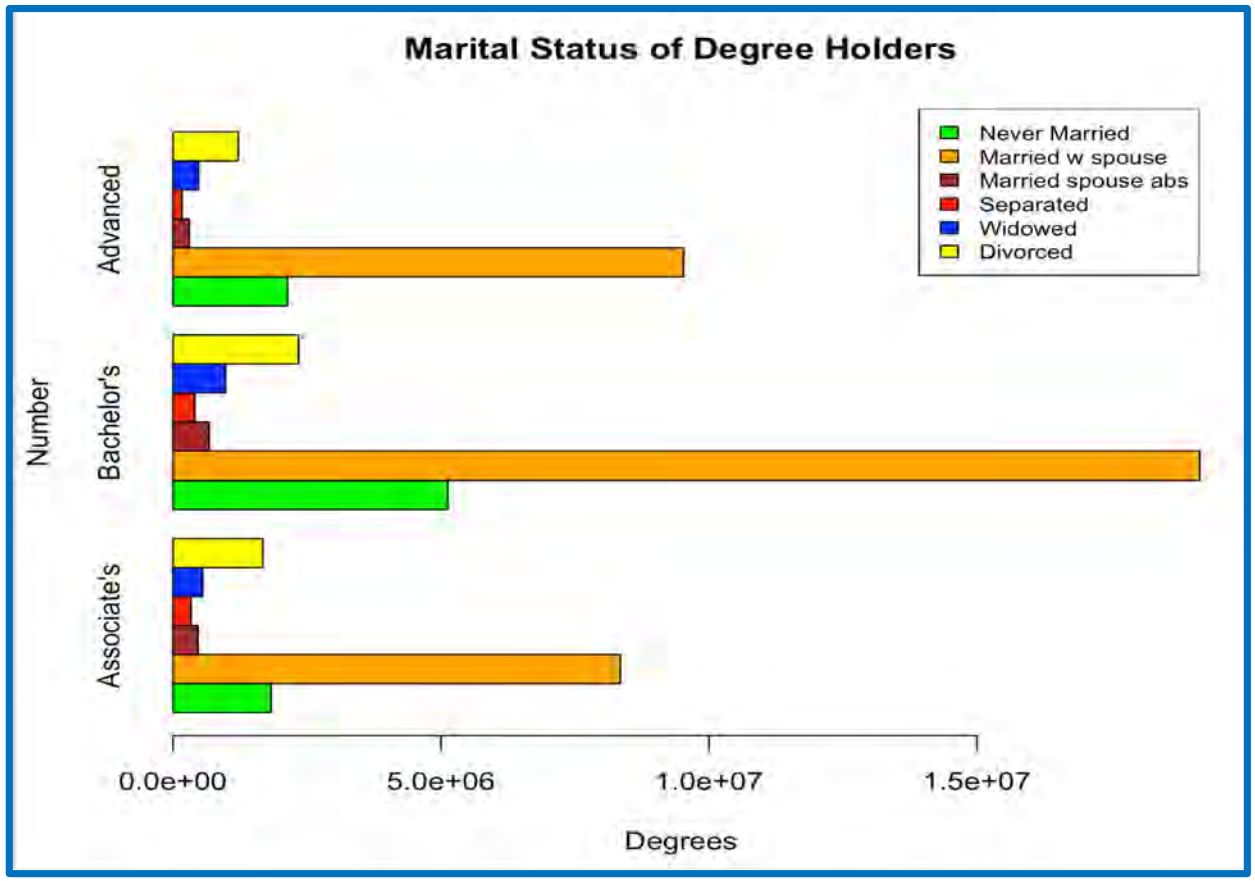
Creating Stacked Barplots for Degree Holders

```
# Create the stacked barplot.
barplot(value,main = "Marital Status of Degree Holders",names.arg = Degrees ,xlab = "Degrees",ylab = "Number",
        col = colors)   beside = FALSE
# Add the legend to the chart.
legend("topright", MS, cex = 0.8, fill = colors)
```

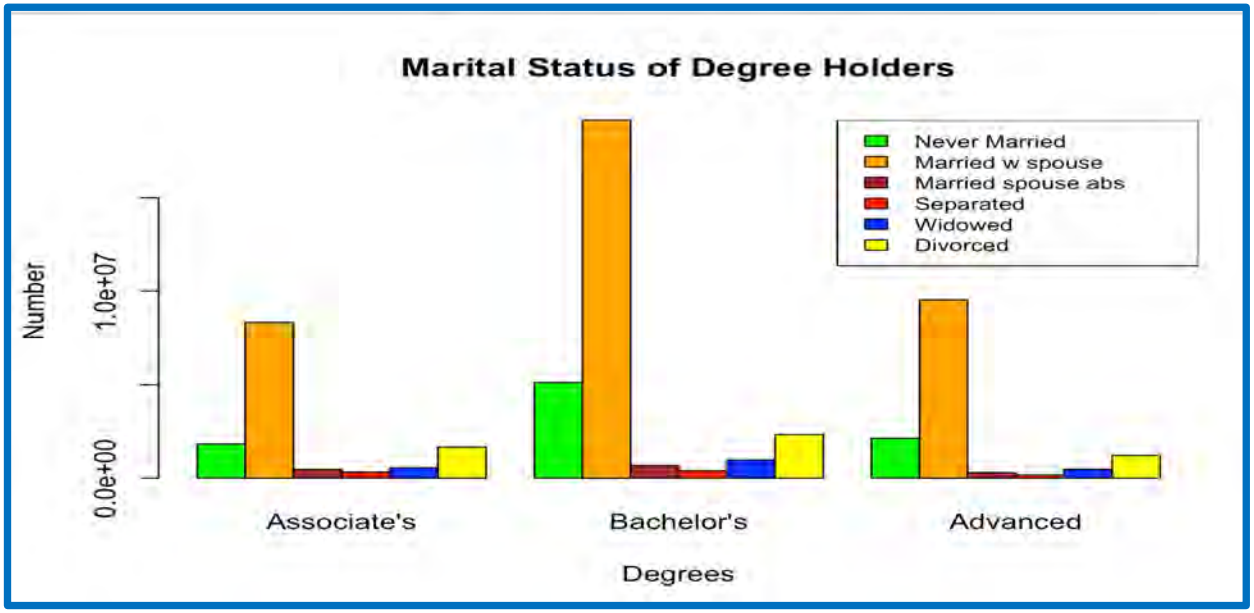


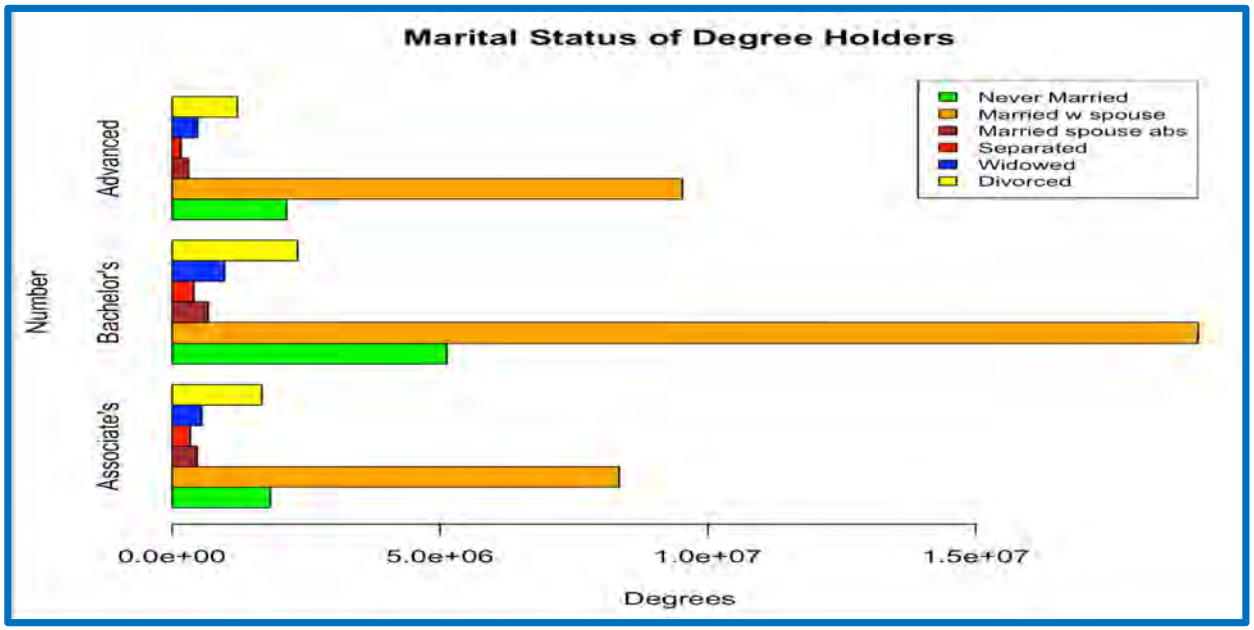
Horizontal Grouped Barplot in R

```
# Create the grouped horizontal barplot.
barplot(value,main = "Marital Status of Degree Holders",names.arg = Degrees ,xlab = "Degrees",ylab = "Number",
        col = colors, beside = TRUE, horiz = TRUE)
# Add the legend to the chart.
legend("topright", MS, cex = 0.8, fill = colors)
```



Horizontal vs Vertical Barplots

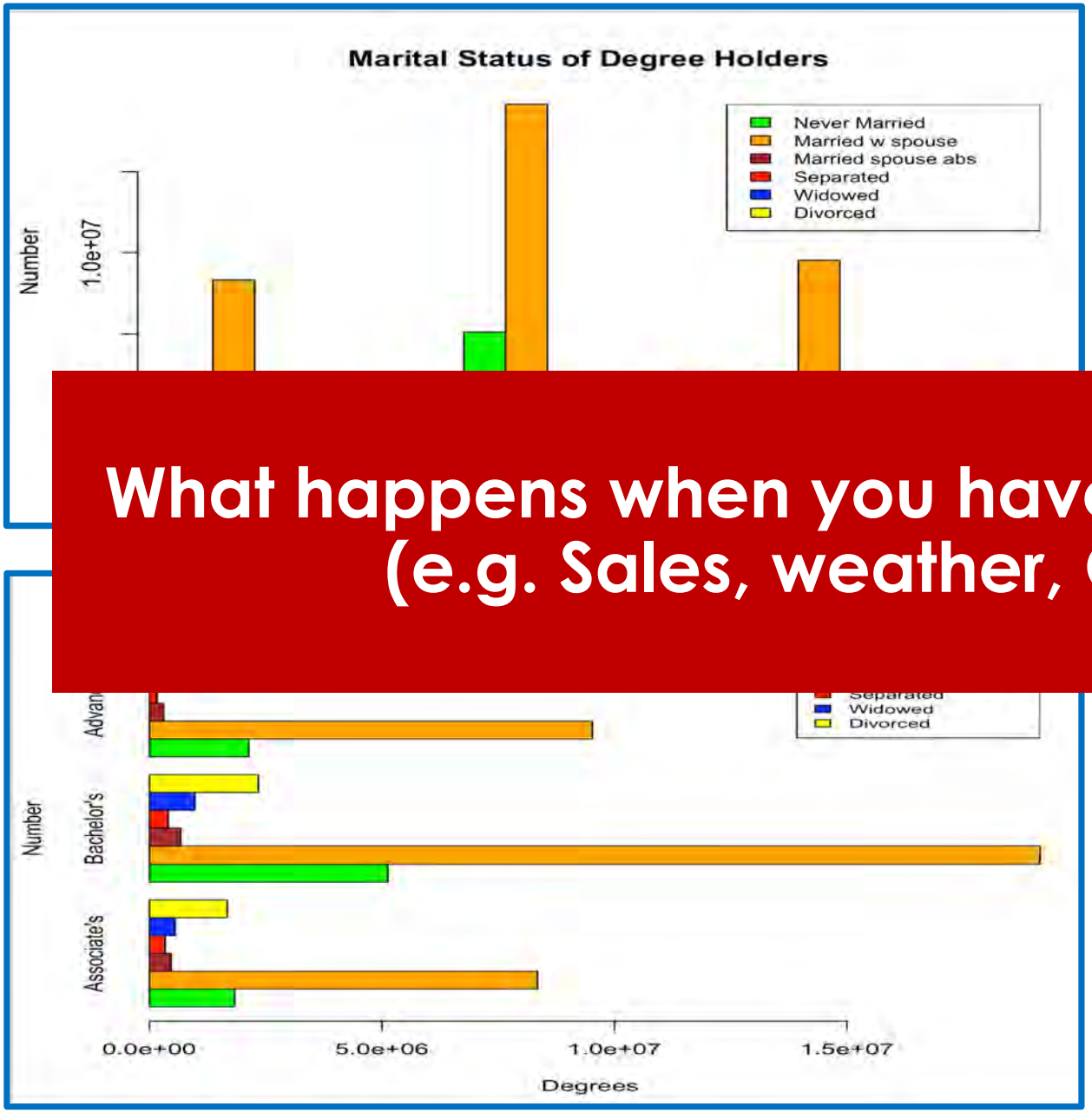




Horizontal & Vertical barplots are useful for:

- **comparing categorical or ordinal data**
- **illustrating differences between sets of values**
- **showing proportions or percentages of a whole**

Horizontal vs Vertical Barplots



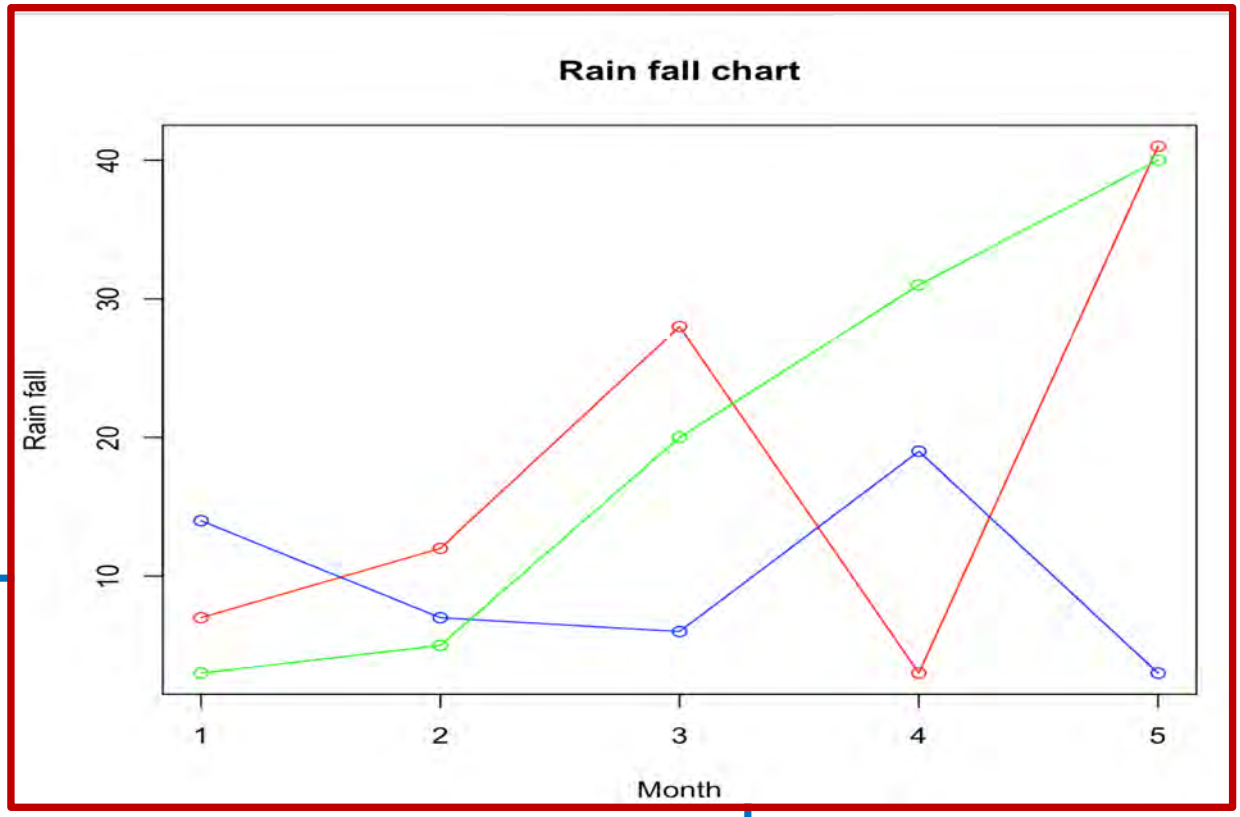
What happens when you have trend data? (e.g. Sales, weather, GDP)

Horizontal & Vertical barplots are useful for:

- **comparing**
- **values**
- **showing proportions or percentages of a whole**

Line Charts

- Useful for displaying data over time.



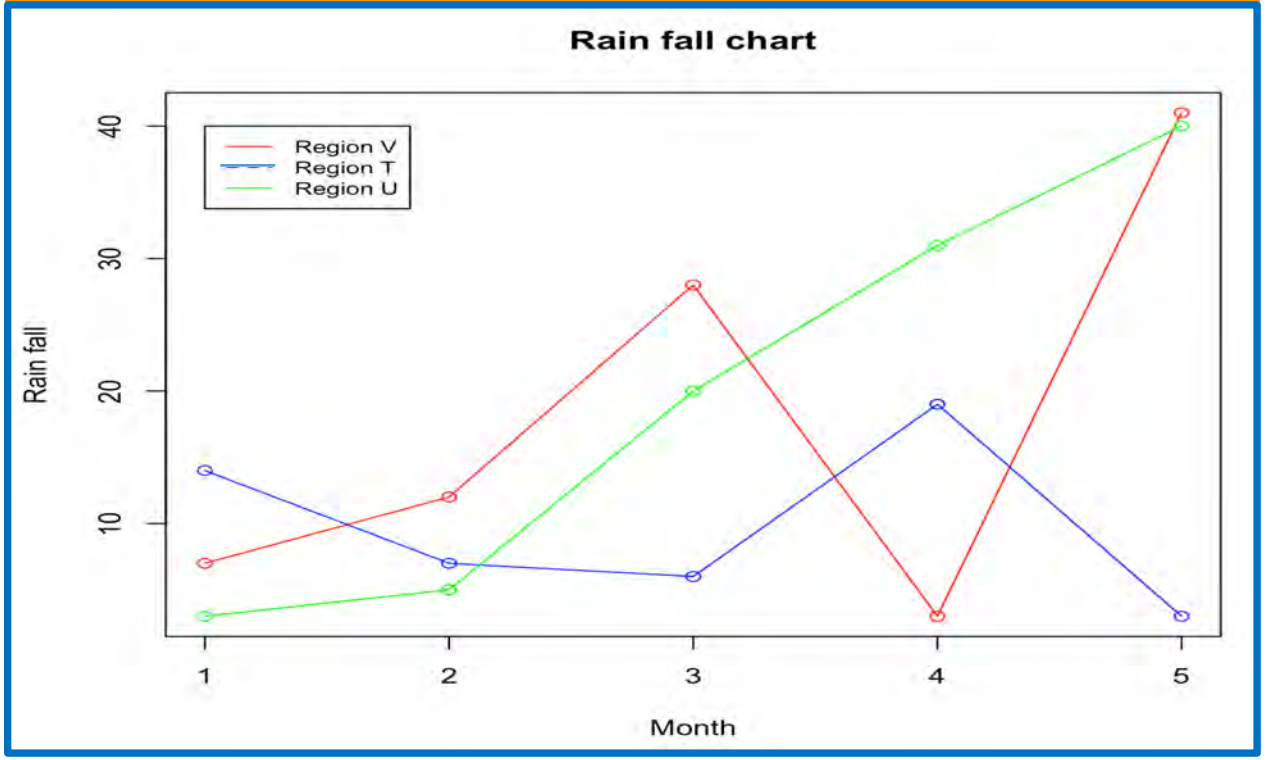

```
# Create Line Charts
# Create the data for the chart.
v <- c(7,12,28,3,41)
t <- c(14,7,6,19,3)
u <- c (3,5,20,31,40)

# Plot the bar chart.
plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall",
     main = "Rain fall chart")
lines(t, type = "o", col = "blue")
lines(u, type = "o", col = "green")
```

Adding a legend to Line Charts

```
legend(x, y=NULL, legend, fill, col, bg)
```

- **x and y** : the x and y co-ordinates to be used to position the legend
- **legend** : the text of the legend
- **fill** : colors to use for filling the boxes beside the legend text
- **col** : colors of lines and points beside the legend text
- **bg** : the background color for the legend box.

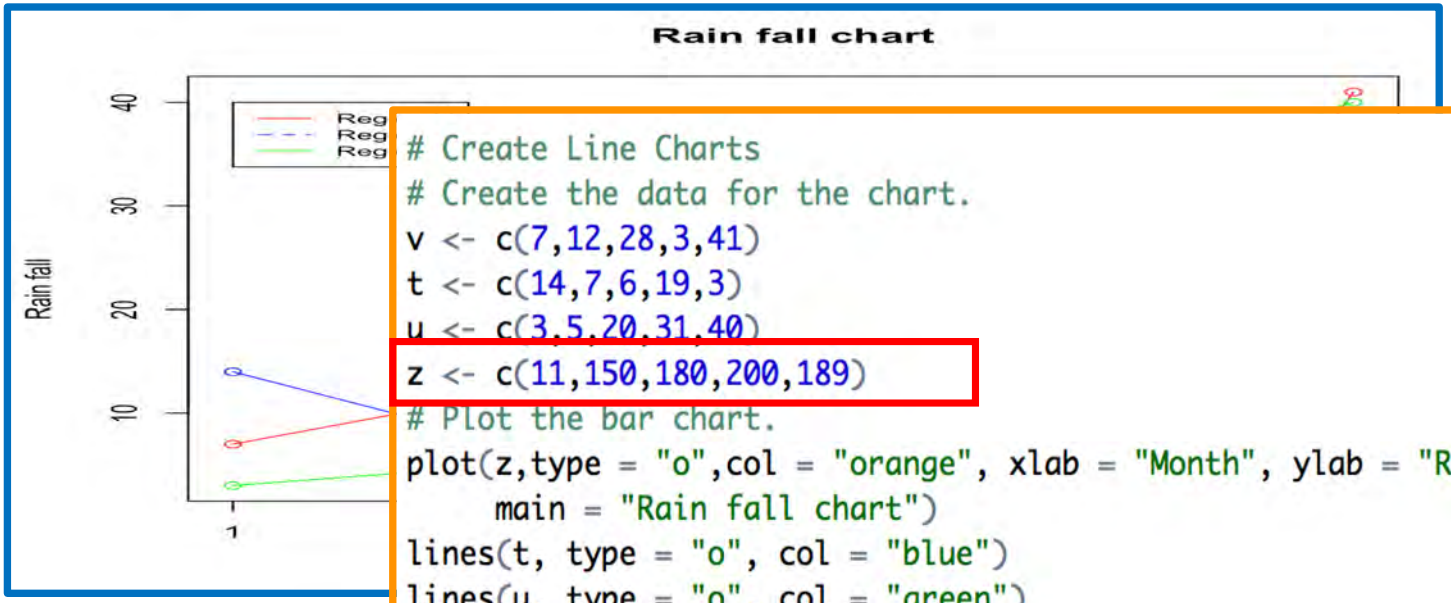
```
# Add a legend
legend(1, 40, legend=c("Region V","Region T","Region U"),
       col=c("red","blue","green"), lty=1 , cex=0.8)
```



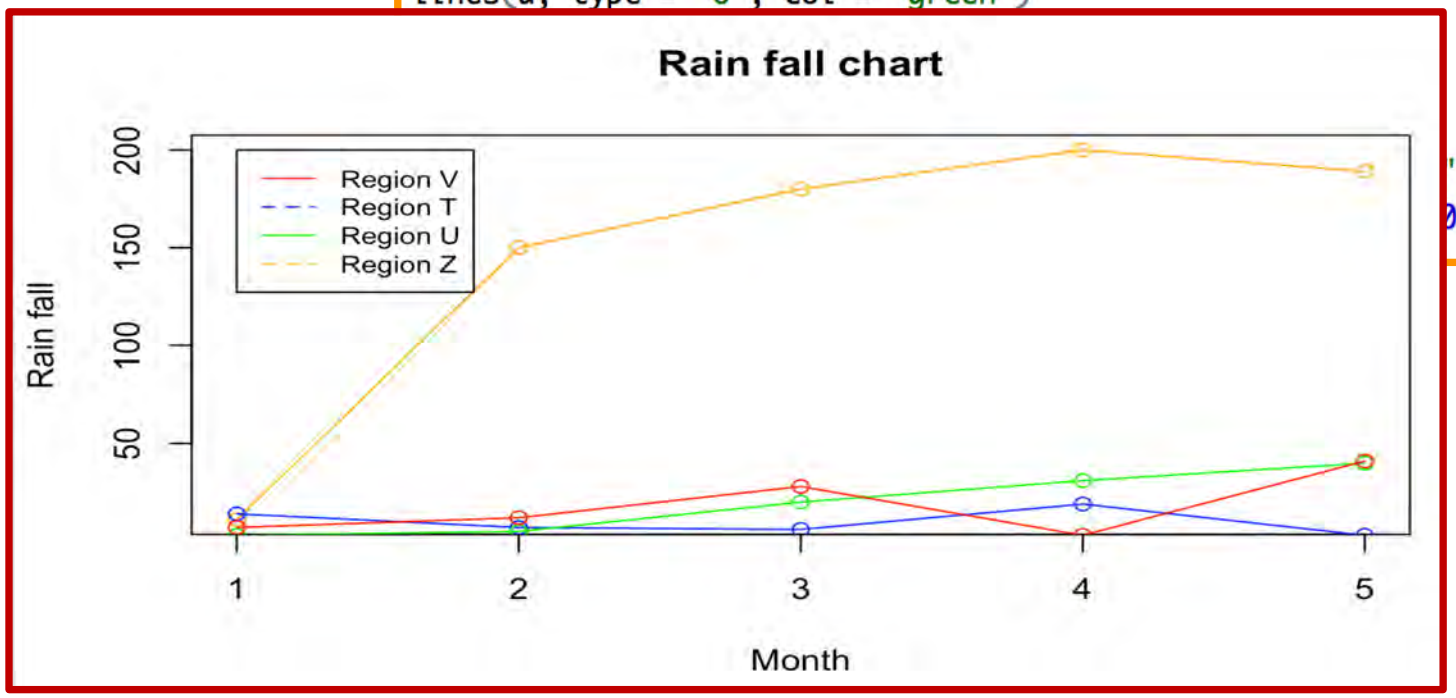
More reference on legends:
http://www.sthda.com/english/wiki/add-legends-to-plots-in-r-software-the-easiest-way

Reference on line types specified by lty
http://www.sthda.com/english/wiki/line-types-in-r-lty

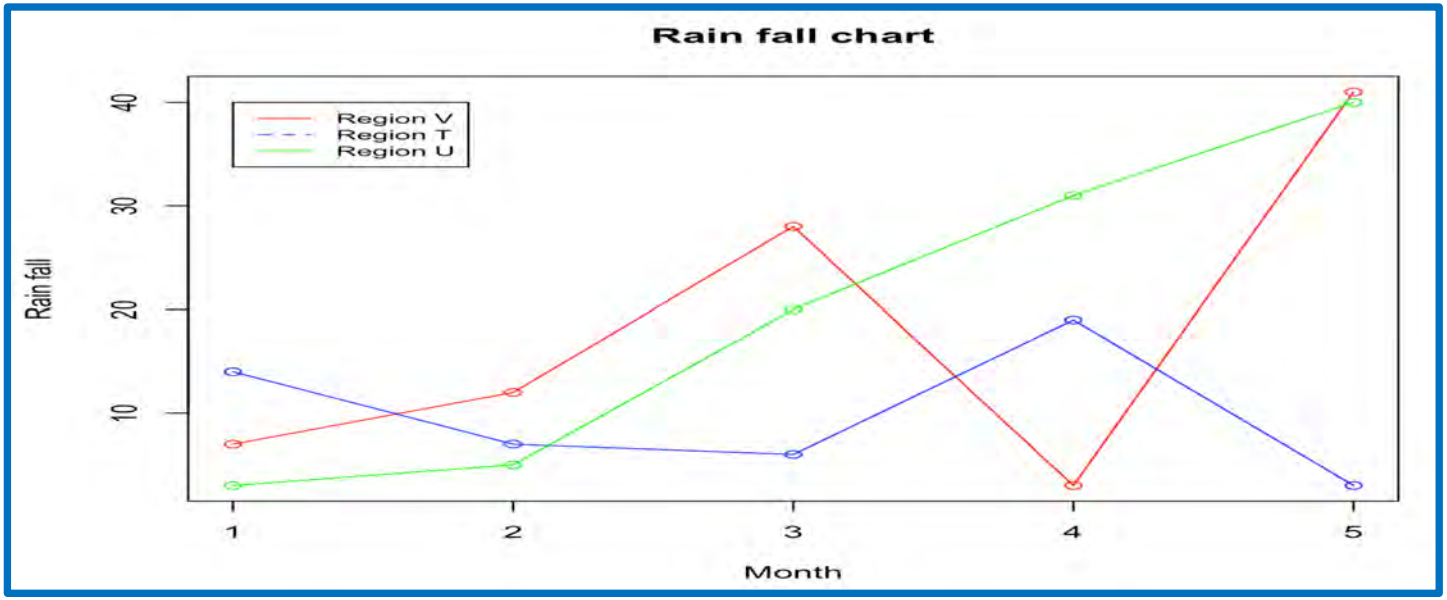
Line Charts

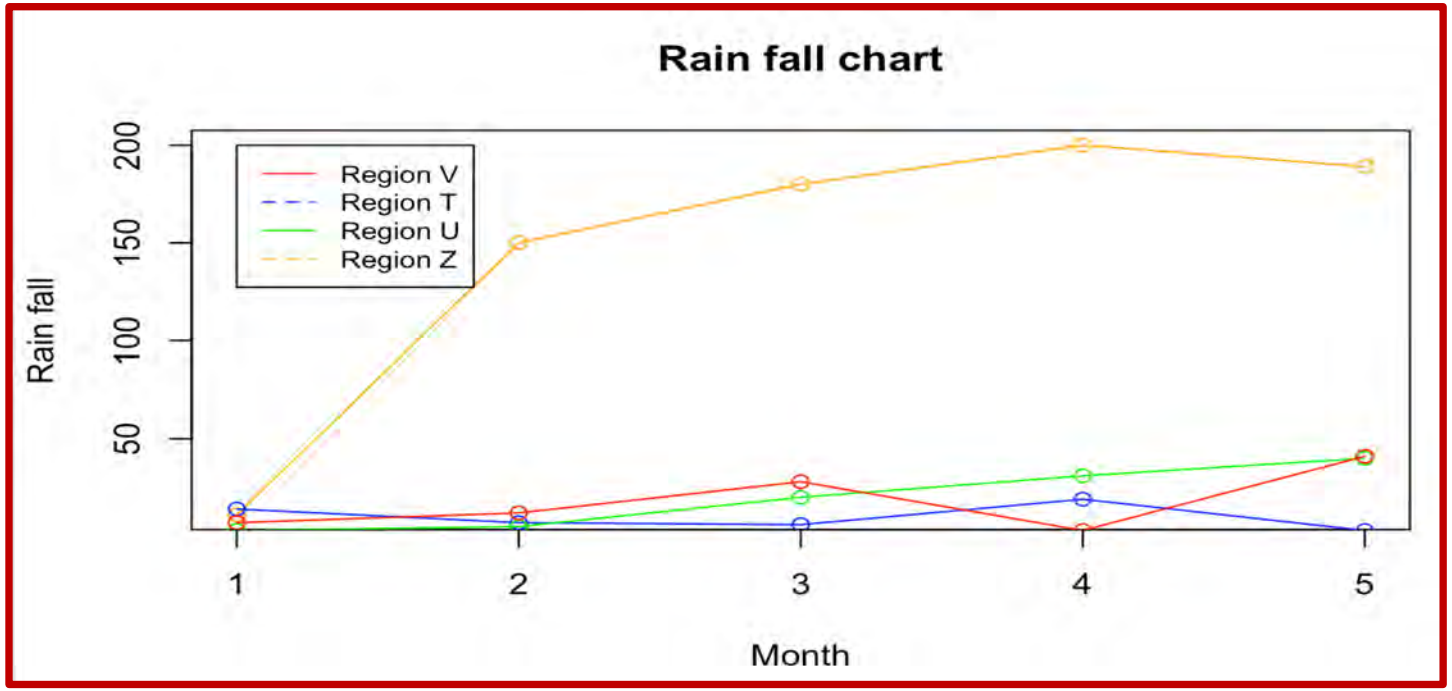



```
# Create Line Charts
# Create the data for the chart.
v <- c(7,12,28,3,41)
t <- c(14,7,6,19,3)
u <- c(3,5,20,31,40)
z <- c(11,150,180,200,189)
# Plot the bar chart.
plot(z,type = "o",col = "orange", xlab = "Month", ylab = "Rain fall",
     main = "Rain fall chart")
lines(t, type = "o", col = "blue")
lines(u, type = "o", col = "green")
```



Line Charts

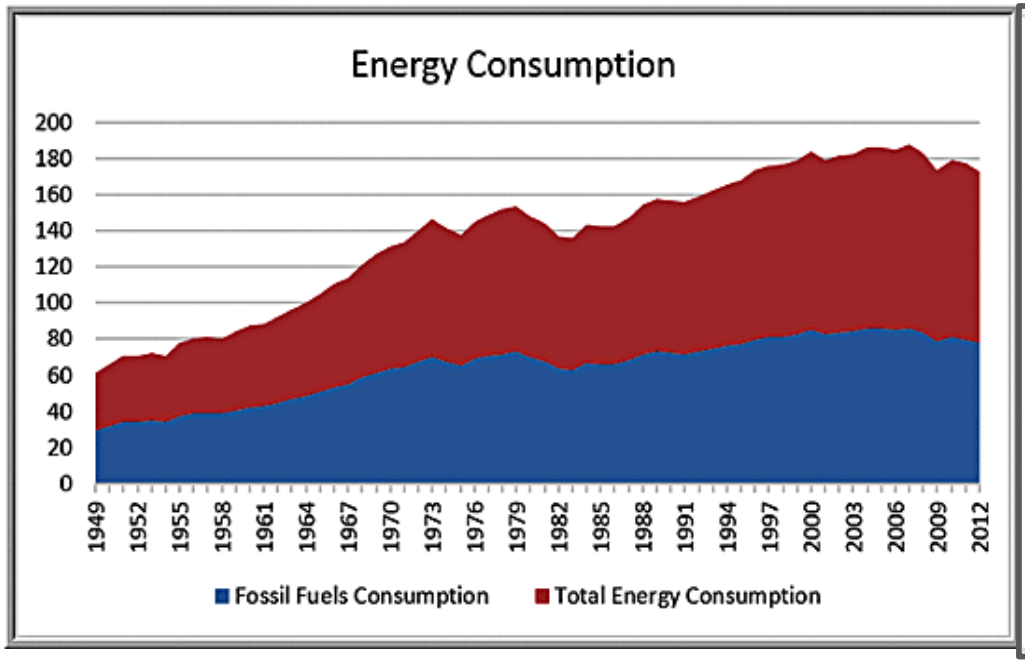


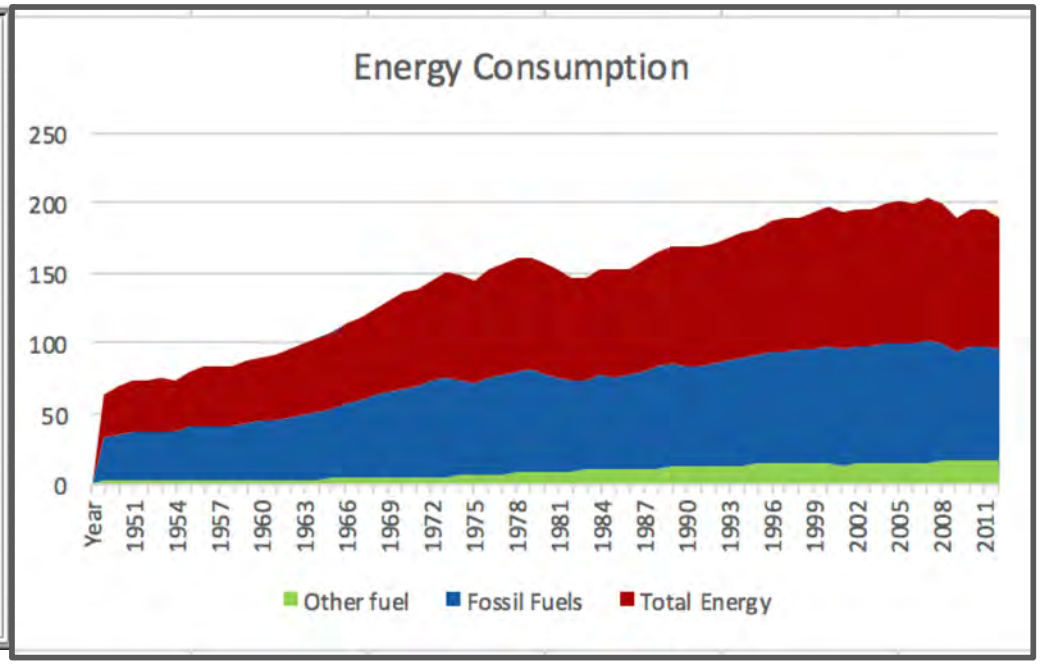


- Multiple data series in line charts
- Magnitude of data values should not differ greatly
- Create separate charts for each data series

Area Charts

An Area Chart for Energy Consumption





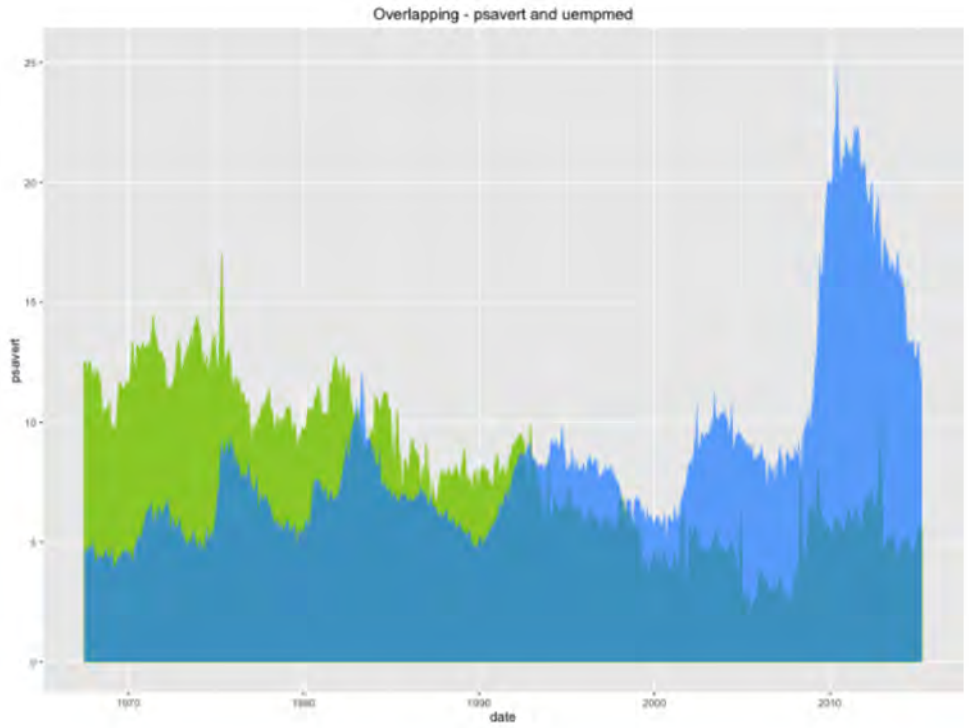
- **Combines features of a pie chart with those of line charts**

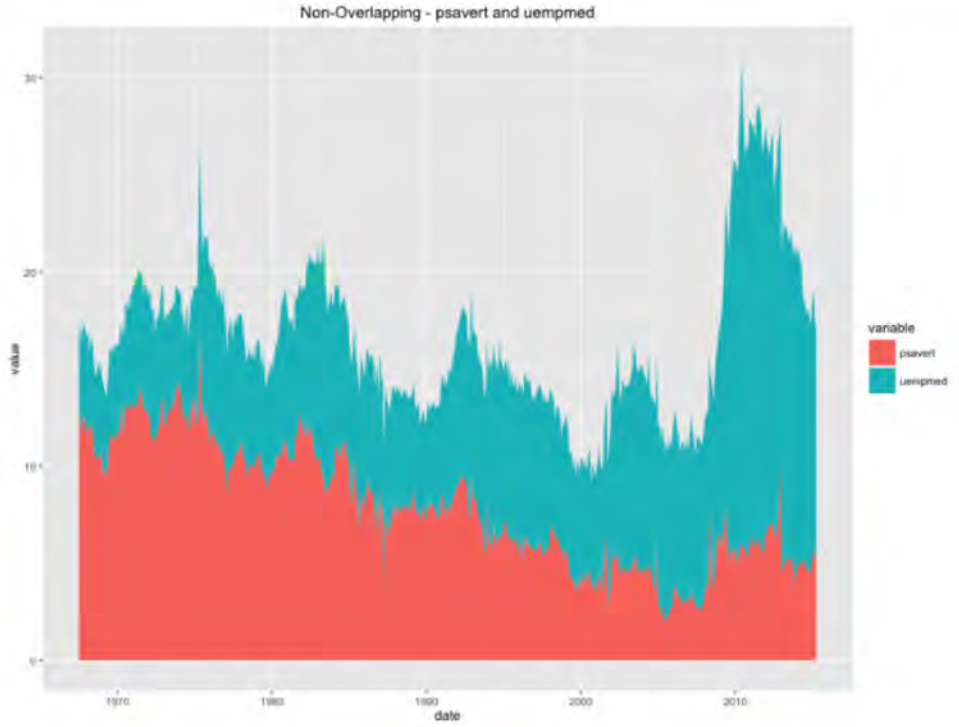
How does it compare with line chart or pie chart?

When should it be avoided?

ggplot2 package required

Reference: http://r-statistics.co/ggplot2-cheatsheet.html#Line%20chart





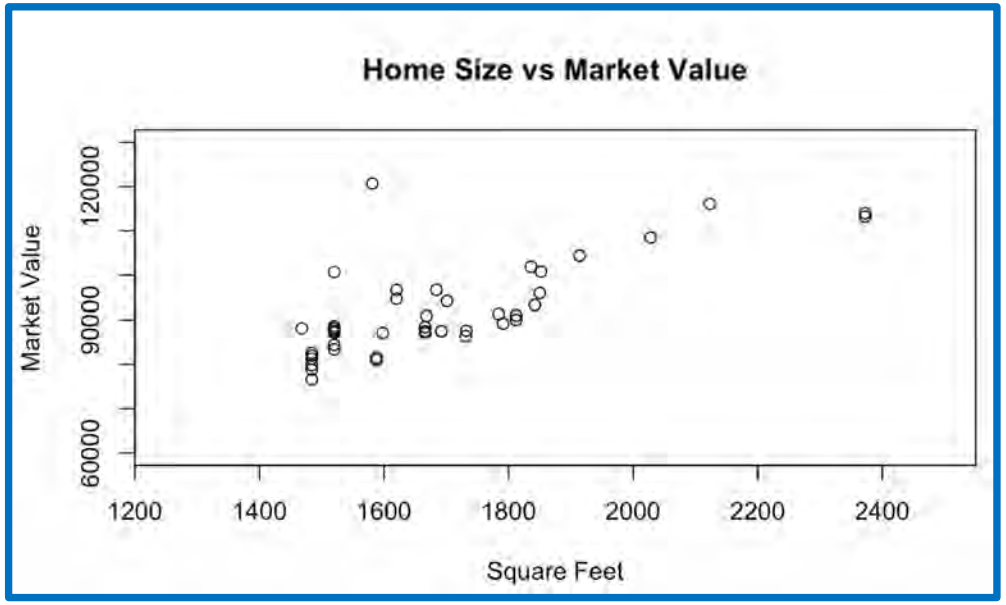
Scatterplots

Using Home Market Value_R data

	House Age	Square Feet	Market Value
1	33	1812	90000
2	32	1914	104400
3	32	1842	93300
4	33	1812	91000
5	32	1836	101900
6	33	2028	108500
7	32	1732	87600
8	33	1850	96000
9	32	1791	89200
10	33	1666	88400
11	32	1852	100800
12	32	1620	96700
13	32	1692	87500
14	32	2372	114000

ving 1 to 14 of 42 entries

Scatter Chart for Real Estate Data



What is the relationship between house size and market value?

- show relationship between two variables
- to construct a scatterplot, we need observations consisting of *pairs* of variables (e.g. house size & market value)

Scatterplots

Using Home Market Value_R data

	House Age	Square Feet	Market Value
1	33	1812	90000
2	32	1914	104400
3	32	1842	93300
4	33	1812	91000
5	32	1836	101900
6	33	2028	108500
7	32	1732	87600
8	33	1850	96000
9	32	1791	89200
10	33	1666	88400
11	32	1852	100800
12	32	1620	96700
13	32	1692	87500
14	32	2372	114000

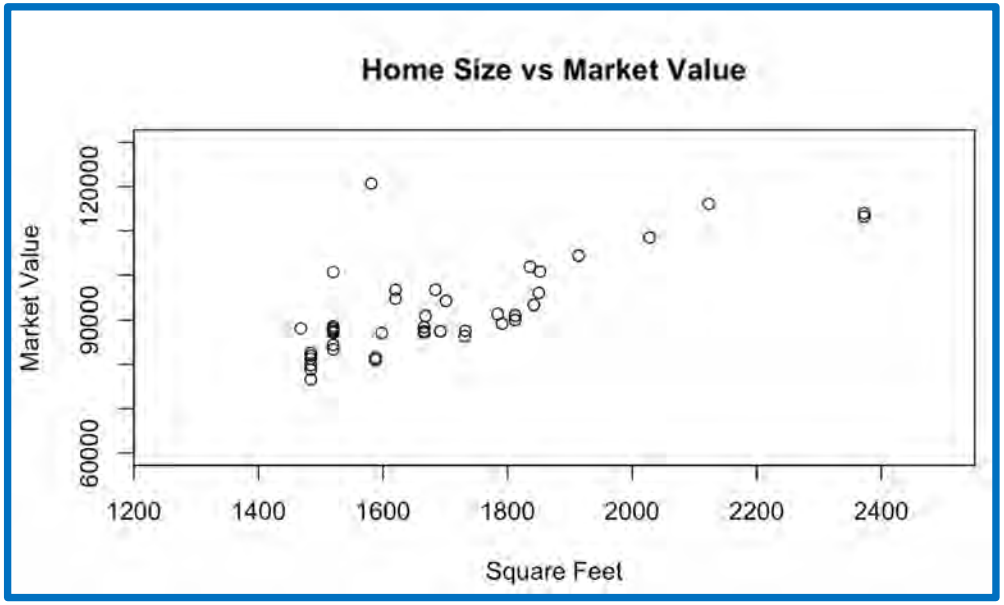
ving 1 to 14 of 42 entries

```
plot(x, y, main, xlab, ylab, xlim, ylim, axes)
```

Following is the description of the parameters used −

- **x** is the data set whose values are the horizontal coordinates.
- **y** is the data set whose values are the vertical coordinates.
- **main** is the tile of the graph.
- **xlab** is the label in the horizontal axis.
- **ylab** is the label in the vertical axis.
- **xlim** is the limits of the values of x used for plotting.
- **ylim** is the limits of the values of y used for plotting.
- **axes** indicates whether both axes should be drawn on the plot

Scatter Chart for Real Estate Data



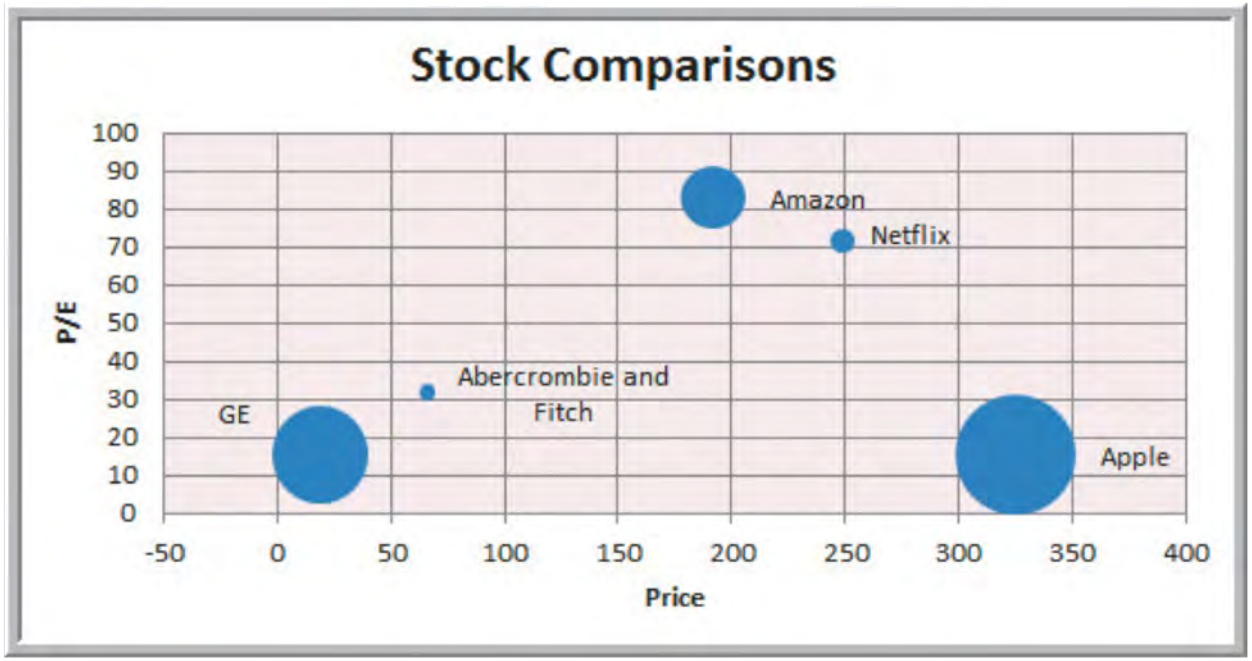
What is the relationship between house size and market value?

```
#create scatter plot.
df3 <- Home_Market_Value_R
input <- df3[,c(2,3)]
plot(input$`Square Feet`, input$`Market Value`, xlab = "Square Feet",ylab = "Market Value ($)", xlim = c(1250,2500),
ylim = c(60000,130000),main = "Home Size vs Market Value")
```

Bubble Charts

Stock Comparisons					
	Netflix	Apple	GE	Amazon	Abercrombie & Fitch
Price	248.66	325	18.6	192	65.9
P/E	71.41	15.5	15.56	83	31.9
Market Cap ($Billions)	13	301	198	86.6	5.77

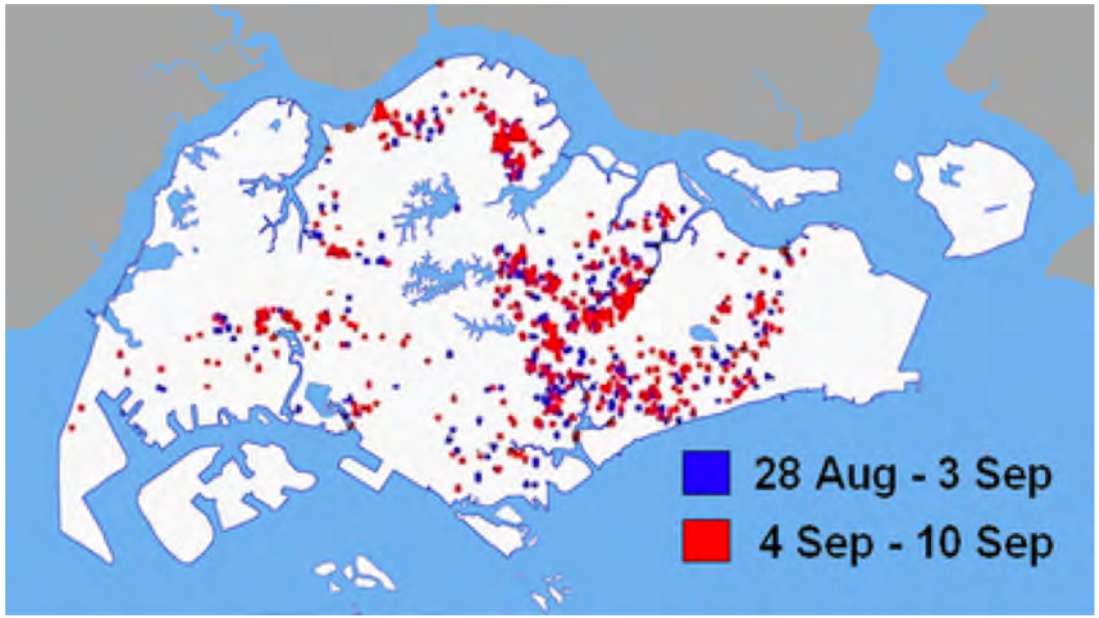
Using the Stock Comparisons file



A Bubble Chart for Stock Comparisons

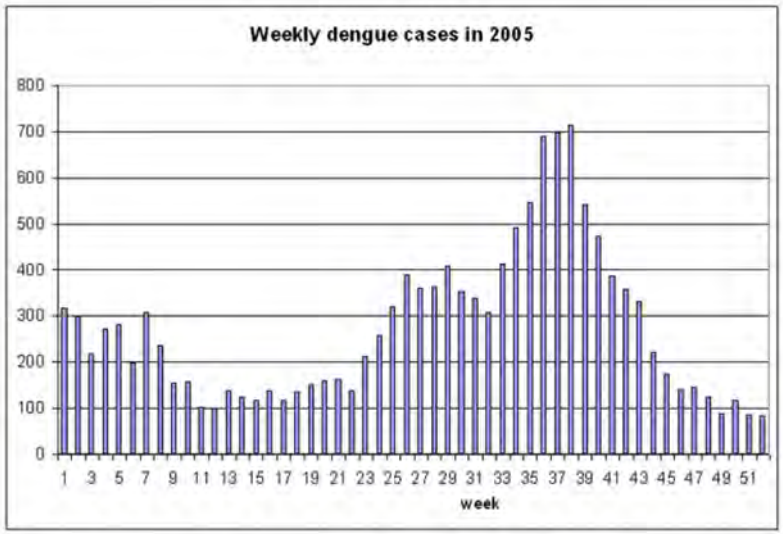
- Type of scatter chart in which size of data marker corresponds to value of a third variable
- Enables plotting of three variables in two dimensions

Geographic Data



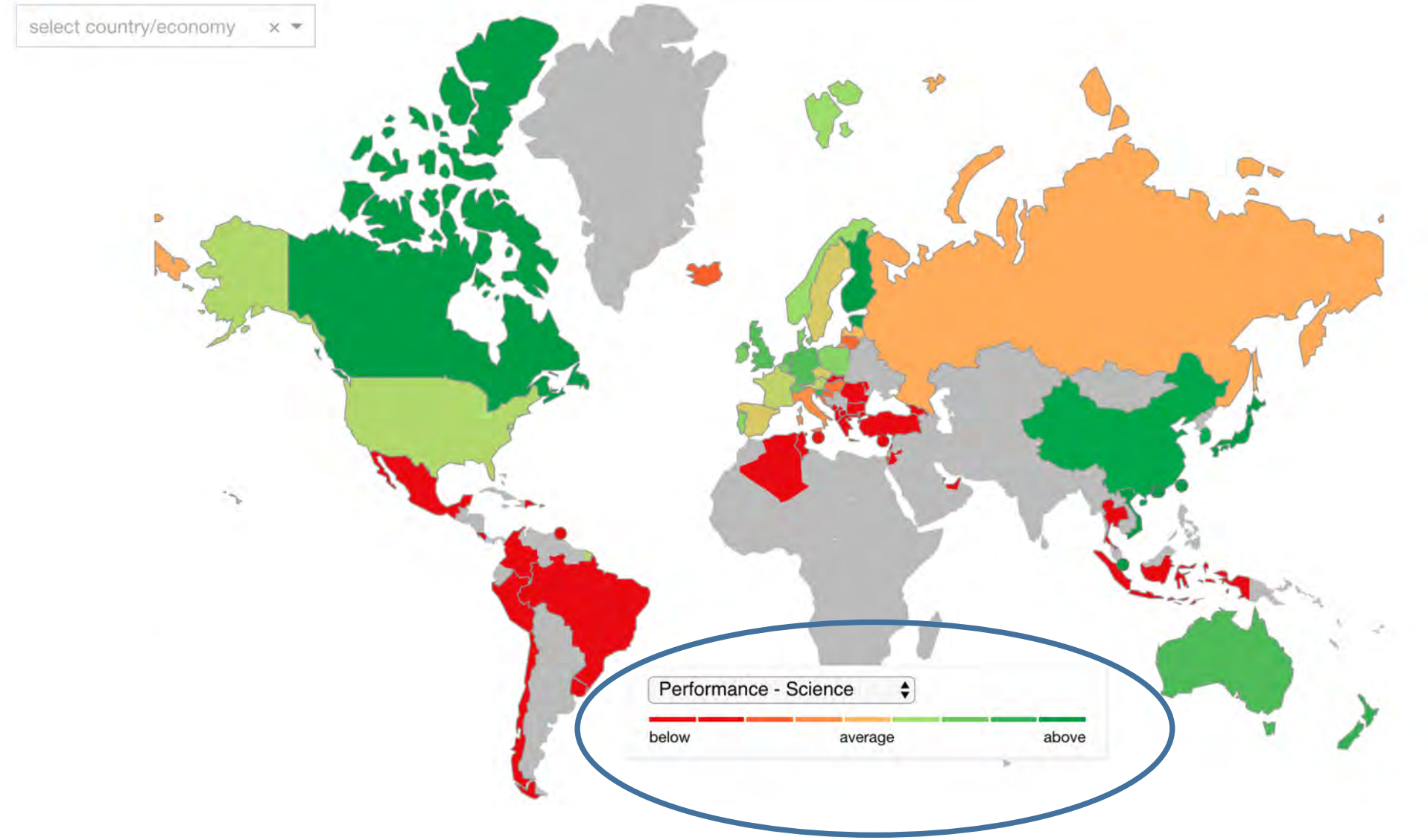
What does this show?

When would geographical data display be useful?



data can highlight key data ends, uncover business opportunities,

PISA 2015


select country/economy
Performance - Science
below
average
above


NUS National University of Singapore
School of Computing
Leading The World With Asia's Best
BT1101 Introduction to Business Analytics
Data Tabulation & Frequencies

Learning objectives

- Appreciate the importance and role of data visualization through tabulation
- Be able to describe and summarize data using tabular techniques (e.g. frequency tables, contingency tables)
- Be able to use and construct frequency distributions, relative frequency distributions, histogram and to compute cumulative relative frequencies, percentiles and quartiles for a data set

Frequency Table

- Frequency distribution - a table that shows number of observations in each of several non-overlapping groups
- Categorical variables naturally define the groups in a frequency distribution.

Frequency Distributions for Categorical Data – An Example

Home_Market_Value_Type

Home_Market_Value_Type

Filter

	House Age	Square Feet	Market Value	Type
1	33	1812	90000	A1
2	32	1914	104400	B1
3	32	1842	93300	A1
4	33	1812	91000	A1
5	32	1836	101900	B2
6	33	2028	108500	B1
7	32	1732	87600	A1
8	33	1850	96000	B2
9	32	1791	89200	A2
10	33	1666	88400	B1
11	32	1852	100800	B1
12	32	1620	96700	A2
13	32	1692	87500	B1
14	32	2372	114000	C
15	32	2372	113200	A1
16	33	1666	87500	A1
17	32	2123	116100	C
18	32	1620	94700	B1
19	32	1731	86400	A1

Showing 1 to 22 of 42 entries

One-way frequency table for house type

Type	Freq
A1	12
A2	10
B1	8
B2	5
C	7

House Type

Frequency or Number of Observations

Frequency Distributions for Categorical Data – An Example

Home_Market_Value_Type

Home_Market_Value_Type

Filter

	House Age	Square Feet	Market Value	Type
1	33	1812	90000	A1
2	32	1914	104400	B1
3	32	1842	93300	A1
4	33	1812	91000	A1
5	32	1836	101900	B2
6	33	2028	108500	B1
7	32	1732	87600	A1
8	33	1850	96000	B2
9	32	1791	89200	A2
10	33	1666	88400	B1
11	32	1852	100800	B1
12	32	1620	96700	A2
13	32	1692	87500	B1
14	32	2372	114000	C
15	32	2372	113200	A1
16	33	1666	87500	A1
17	32	2123	116100	C
18	32	1620	94700	B1
19	32	1731	86400	A1

Showing 1 to 22 of 42 entries

```
> Home<-Home_Market_Value_Type
```

METHOD 1

Using “dplyr” package, “group_by” and “summarise” functions

```
> Freq_Type <- group_by (Home,Type) %>% summarise(Freq = n())
```

Type	Freq
A1	12
A2	10
B1	8
B2	5
C	7

METHOD 2

Using “plyr” package, “table” function

```
> Freq_Type2 <- table(Home$Type)
> Freq_Type2

A1 A2 B1 B2  C
12 10  8  5  7
```

```
> df.freq <- as.data.frame(Freq_Type2)
> df.freq
  Var1 Freq
1   A1   12
2   A2   10
3   B1    8
4   B2    5
5    C    7
```

Frequency Distributions for Categorical Data – An Example

Home_Market_Value_Type

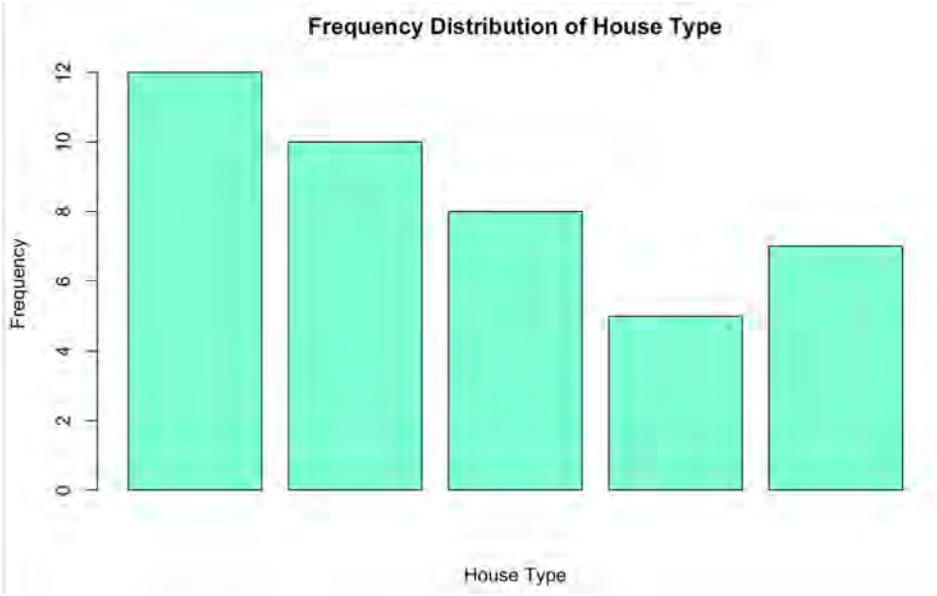
Home_Market_Value_Type

Filter

	House Age	Square Feet	Market Value	Type
1	33	1812	90000	A1
2	32	1914	104400	B1
3	32	1842	93300	A1
4	33	1812	91000	A1
5	32	1836	101900	B2
6	33	2028	108500	B1
7	32	1732	87600	A1
8	33	1850	96000	B2
9	32	1791	89200	A2
10	33	1666	88400	B1
11	32	1852	100800	B1
12	32	1620	96700	A2
13	32	1692	87500	B1
14	32	2372	114000	C
15	32	2372	113200	A1
16	33	1666	87500	A1
17	32	2123	116100	C
18	32	1620	94700	B1
19	32	1731	86400	A1

Showing 1 to 22 of 42 entries

Type	Freq
A1	12
A2	10
B1	8
B2	5
C	7



METHOD 1

Using “dplyr” package, “group_by” and “summarise” functions

```
> Freq_Type <- group_by (Home,Type) %>% summarise(Freq = n())
> barplot(Freq_Type$Freq, main="Frequency Distribution of House Type", xlab="House Type",
ylab="Frequency", col="aquamarine")
```

Plotting Frequency Distributions for Categorical Data – An Example

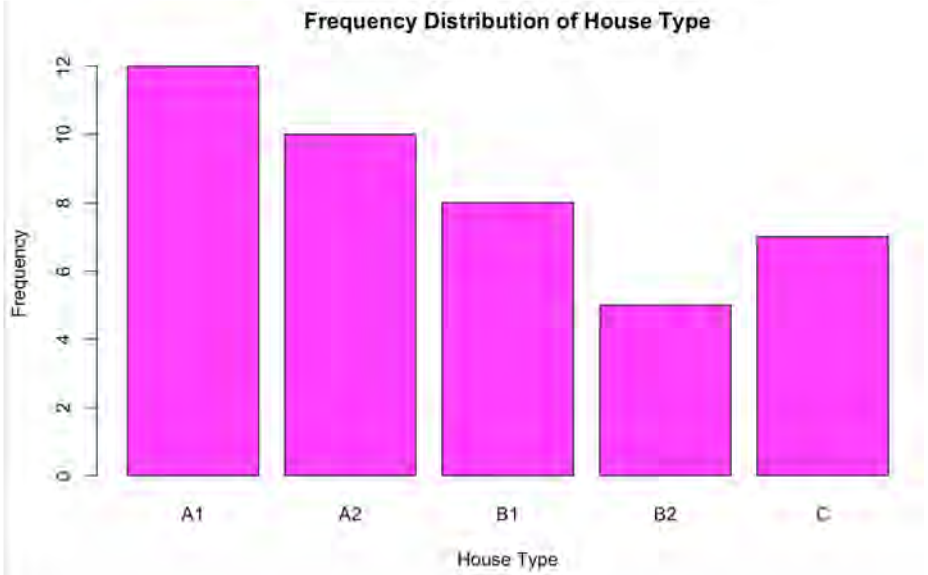
Home_Market_Value_Type

Home_Market_Value_Type

Filter

	House Age	Square Feet	Market Value	Type
1	33	1812	90000	A1
2	32	1914	104400	B1
3	32	1842	93300	A1
4	33	1812	91000	A1
5	32	1836	101900	B2
6	33	2028	108500	B1
7	32	1732	87600	A1
8	33	1850	96000	B2
9	32	1791	89200	A2
10	33	1666	88400	B1
11	32	1852	100800	B1
12	32	1620	96700	A2
13	32	1692	87500	B1
14	32	2372	114000	C
15	32	2372	113200	A1
16	33	1666	87500	A1
17	32	2123	116100	C
18	32	1620	94700	B1
19	32	1731	86400	A1

Showing 1 to 22 of 42 entries



METHOD 2

Using “plyr” package, “table” function

```
> Freq_Type2 <- table(Home$Type)
> Freq_Type2

A1 A2 B1 B2  C
12 10  8  5  7
```

```
> barplot(Freq_Type2, main="Frequency Distribution of House Type", xlab="House Type",ylab="Frequency", col="magenta")
```

Relative Frequency Distributions

- Relative frequency is the fraction, or proportion, of the total.
- If a data set has *n* observations, the relative frequency of category *i* is:

$$\text{relative frequency of category } i = \frac{\text{frequency of category } i}{n}$$

Relative Frequency Distribution

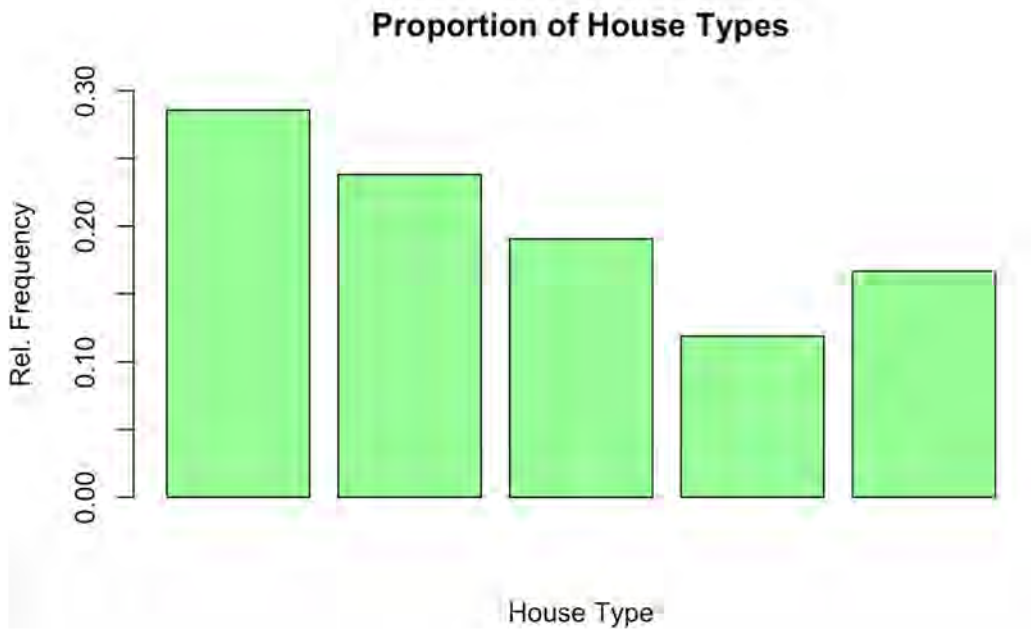
Type	Freq	Rel.Freq
A1	12	0.2857143
A2	10	0.2380952
B1	8	0.1904762
B2	5	0.1190476
C	7	0.1666667

```
> Freq_Type$Rel.Freq <- Freq_Type$Freq/sum(Freq_Type$Freq)
```

Barplot for Relative Frequency of House Type

Relative Frequency Distribution

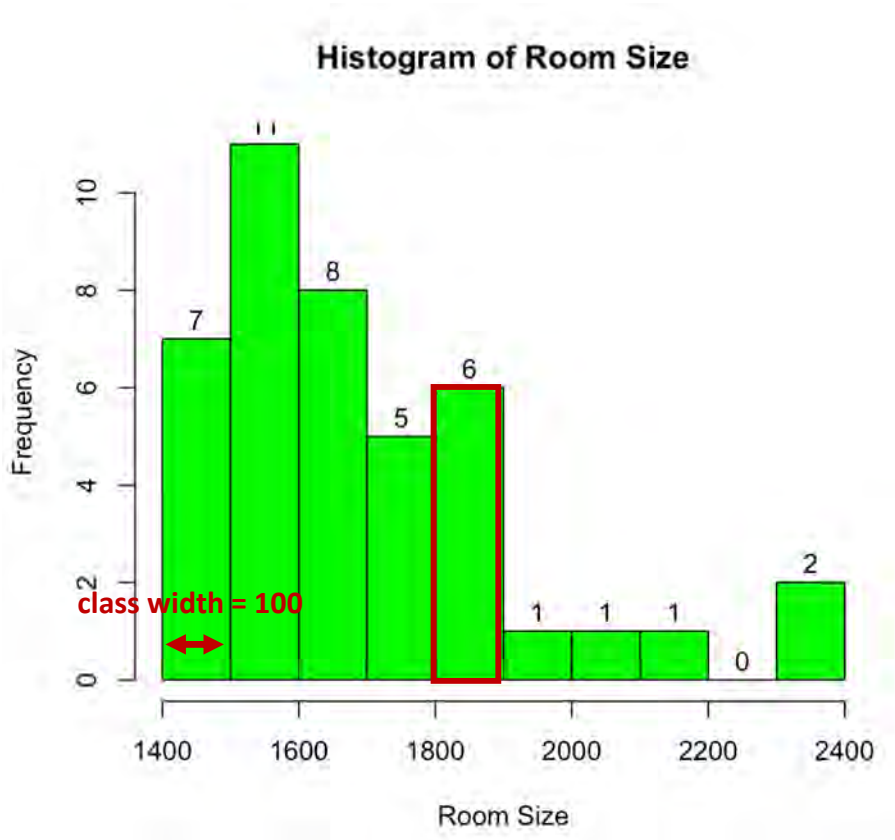
Type	Freq	Rel.Freq
A1	12	0.2857143
A2	10	0.2380952
B1	8	0.1904762
B2	5	0.1190476
C	7	0.1666667




```
> barplot(Freq_Type$Rel.Freq, main="Proportion of House Types", xlab="House Type",ylab="Rel. Frequency",
ylim=c(0,0.3), col="palegreen")
```

Frequency Distributions for Numeric Data

- Histogram: A graphical depiction of a frequency distribution for numerical data in the form of a bar chart
- Terminologies:
  - Class*: a category for grouping data
  - Frequency: Number of data values in a class
  - Density: Relative frequency
  - Upper class limit: largest value that can go in a class
  - Lower class limit: smallest value that can go in a class
  - Class width: Difference between lower class limit of a given class and the lower class limit of the next higher class.
  - Class midpoint: Midpoint of a class



Histogram

- Plotting histogram using “hist” function

```
hist(v,main,xlab,xlim,ylim,breaks,col,border)
```

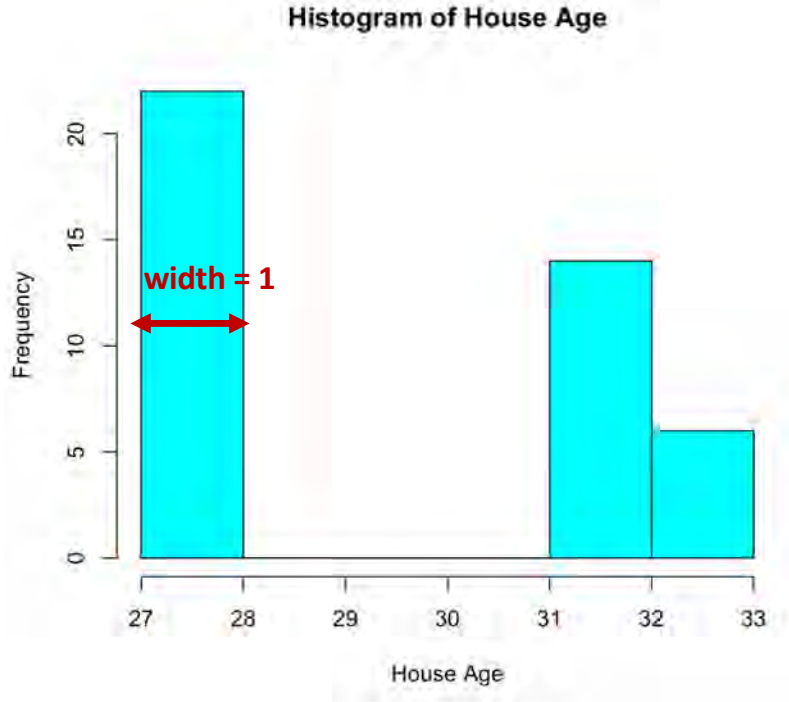
- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.
  (default value is based on Sturges’s rule)

Sturge’s Rule:
k = 1 + 3.322(log n) (k is the number of classes; n is the size of the data)
Eg: k=1+3.322(log 42) = 5.39 --> 6

```
hist(Home$`House Age`,main="Histogram of House Age", col="cyan",xlab="House Age")
```

```
> Home<-Home_Market_Value_Type
```

	House Age	Square Feet	Market Value	Type
1	33	1812	90000	A1
2	32	1914	104400	B1
3	32	1842	93300	A1
4	33	1812	91000	A1
5	32	1836	101900	B2
6	33	2028	108500	B1
7	32	1732	87600	A1
8	33	1850	96000	B2



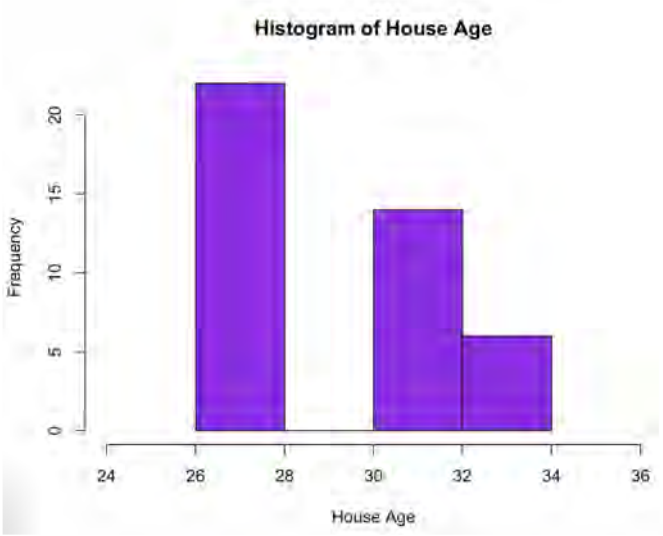
Histograms for Numerical Data

Some rules of thumb:

1. Number of groups - Choose between 5 to 15 groups; more for larger n; range of each should be equal.

2. Choose lower limit of first group (LL) as a whole number smaller than the minimum data value and the upper limit of last group (UL) as a whole number larger than the maximum data value.

3. Group or bar width = $\dfrac{\text{UL} - \text{LL}}{\text{number of groups}}$

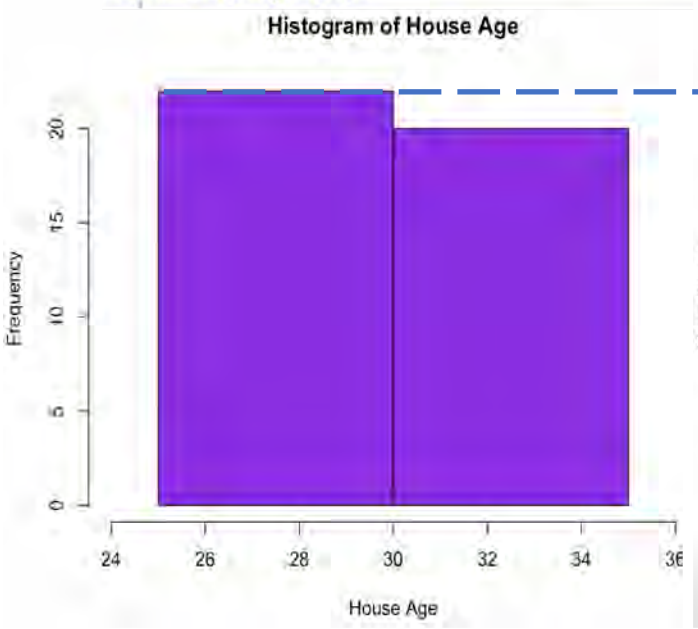


Histograms for Numerical Data

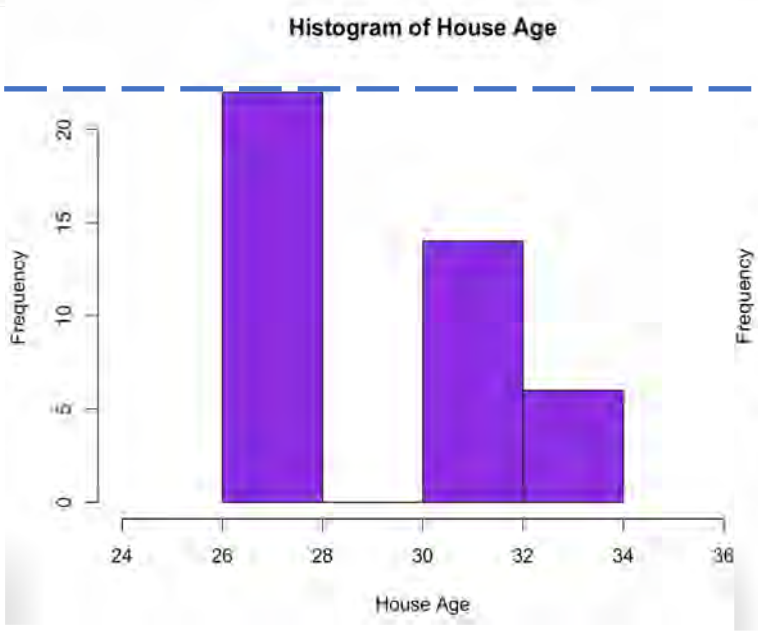
bar width = $\frac{\text{UL} - \text{LL}}{\text{number of groups}}$

- a single number giving the number of cells for the histogram
- a vector giving the breakpoints between histogram cells,
- a function to compute the vector of breakpoints,
- seq (from,to,by)

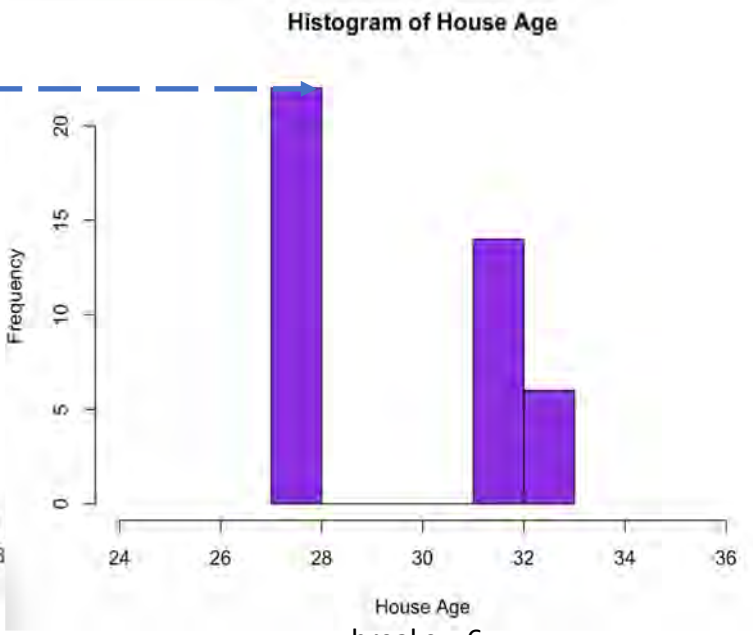
```
> hist(Home$`House Age`,main="Histogram of House Age", col="blueviolet",xlab="House Age", breaks=2,
xlim= c(24,36) )
```



breaks = 2

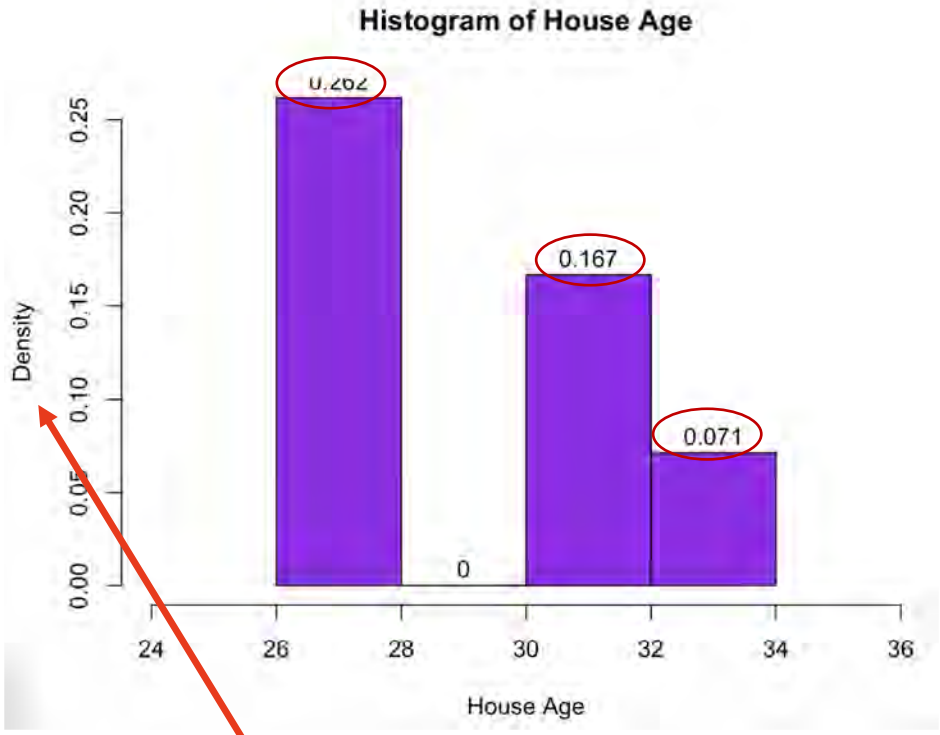


breaks = 4



breaks = 6

Histogram (label and density)




```
> hist(Home$`House Age`,main="Histogram of House Age", col="blueviolet",xlab="House Age", breaks=3,
xlim= c(24,36), labels=TRUE,probability = TRUE )
```

Cumulative Relative Frequency

- **Relative frequency** can also be computed for numerical data.

$$\frac{\text{Frequency of each group}}{\text{Total Number of observations}}$$

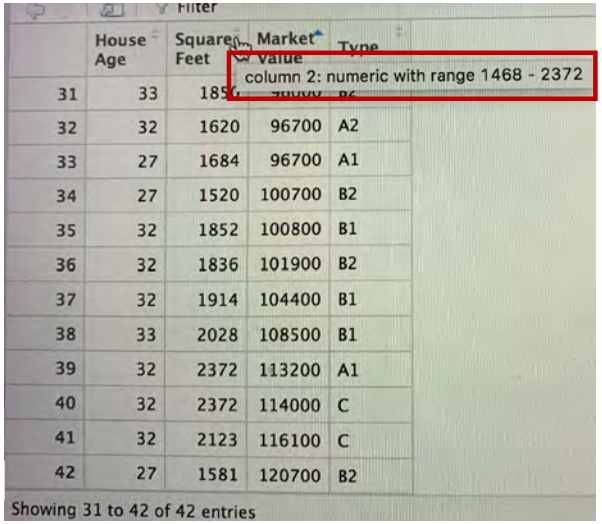
- **Cumulative Relative Frequency**: proportion of total number of observations that fall at or below the upper limit of each group.

Cumulative Relative Frequencies

- Compute Frequency Table

```
> Home<-Home_Market_Value_Type
```

	House Age	Square Feet	Market Value	Type
1	33	1812	90000	A1
2	32	1914	104400	B1
3	32	1842	93300	A1
4	33	1812	91000	A1
5	32	1836	101900	B2
6	33	2028	108500	B1
7	32	1732	87600	A1
8	33	1850	96000	B2




```
> RmSize<-Home$`Square Feet`
> breaks <- seq(1400, 2400, by=100)

> breaks
 [1] 1400 1500 1600 1700 1800 1900 2000 2100 2200 2300 2400

> RmSize.cut <- cut(RmSize, breaks, right=FALSE)
```

cut divides the range of x (RmSize) into intervals and codes the values in x according to which interval they fall. The leftmost interval corresponds to level one, the next leftmost to level two and so on.

```
> RmSize.freq<-table(RmSize.cut)
> freq_table<-transform(RmSize.freq)
> freq_table
          RmSize.cut Freq
1  [1.4e+03,1.5e+03)    7
2  [1.5e+03,1.6e+03)   11
3  [1.6e+03,1.7e+03)    8
4  [1.7e+03,1.8e+03)    5
5  [1.8e+03,1.9e+03)    6
6    [1.9e+03,2e+03)    1
7    [2e+03,2.1e+03)    1
8  [2.1e+03,2.2e+03)    1
9  [2.2e+03,2.3e+03)    0
10 [2.3e+03,2.4e+03)    2
```

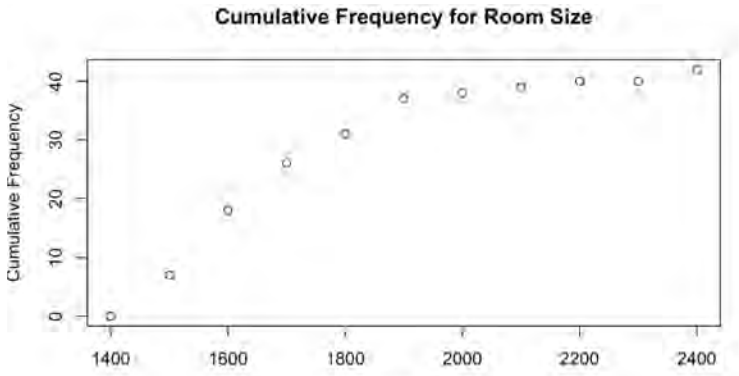
```
> RmSize.cut
 [1] [1.8e+03,1.9e+03) [1.9e+03,2e+03)   [1.8e+03,1.9e+03) [1.8e+03,1.9e+03) [1.8e+03,1.9e+03)
 [6] [2e+03,2.1e+03)   [1.7e+03,1.8e+03) [1.8e+03,1.9e+03) [1.7e+03,1.8e+03) [1.6e+03,1.7e+03)
[11] [1.8e+03,1.9e+03) [1.6e+03,1.7e+03) [1.6e+03,1.7e+03) [2.3e+03,2.4e+03) [2.3e+03,2.4e+03)
[16] [1.6e+03,1.7e+03) [2.1e+03,2.2e+03) [1.6e+03,1.7e+03) [1.7e+03,1.8e+03) [1.6e+03,1.7e+03)
[21] [1.5e+03,1.6e+03) [1.4e+03,1.5e+03) [1.5e+03,1.6e+03) [1.5e+03,1.6e+03) [1.4e+03,1.5e+03)
[26] [1.4e+03,1.5e+03) [1.5e+03,1.6e+03) [1.7e+03,1.8e+03) [1.4e+03,1.5e+03) [1.4e+03,1.5e+03)
[31] [1.5e+03,1.6e+03) [1.5e+03,1.6e+03) [1.4e+03,1.5e+03) [1.5e+03,1.6e+03) [1.6e+03,1.7e+03)
[36] [1.5e+03,1.6e+03) [1.7e+03,1.8e+03) [1.4e+03,1.5e+03) [1.5e+03,1.6e+03) [1.5e+03,1.6e+03)
[41] [1.6e+03,1.7e+03) [1.5e+03,1.6e+03)
10 Levels: [1.4e+03,1.5e+03) [1.5e+03,1.6e+03) [1.6e+03,1.7e+03) ... [2.3e+03,2.4e+03)
```

Cumulative Relative Frequencies

- Compute Cumulative Frequency & Cumulative Relative Frequencies

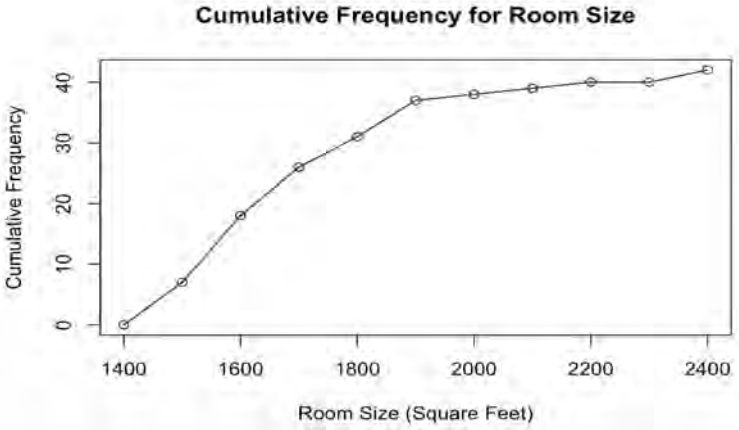
```
> cumfreq_table<-freq_table%>%mutate(cumfreq=cumsum(Freq),cumrelfreq=cumfreq/nrow(Home))
```

	RmSize.cut	Freq	cumfreq	cumrelfreq
1	[1.4e+03,1.5e+03)	7	7	0.1666667
2	[1.5e+03,1.6e+03)	11	18	0.4285714
3	[1.6e+03,1.7e+03)	8	26	0.6190476
4	[1.7e+03,1.8e+03)	5	31	0.7380952
5	[1.8e+03,1.9e+03)	6	37	0.8809524
6	[1.9e+03,2e+03)	1	38	0.9047619
7	[2e+03,2.1e+03)	1	39	0.9285714
8	[2.1e+03,2.2e+03)	1	40	0.9523810
9	[2.2e+03,2.3e+03)	0	40	0.9523810
10	[2.3e+03,2.4e+03)	2	42	1.0000000

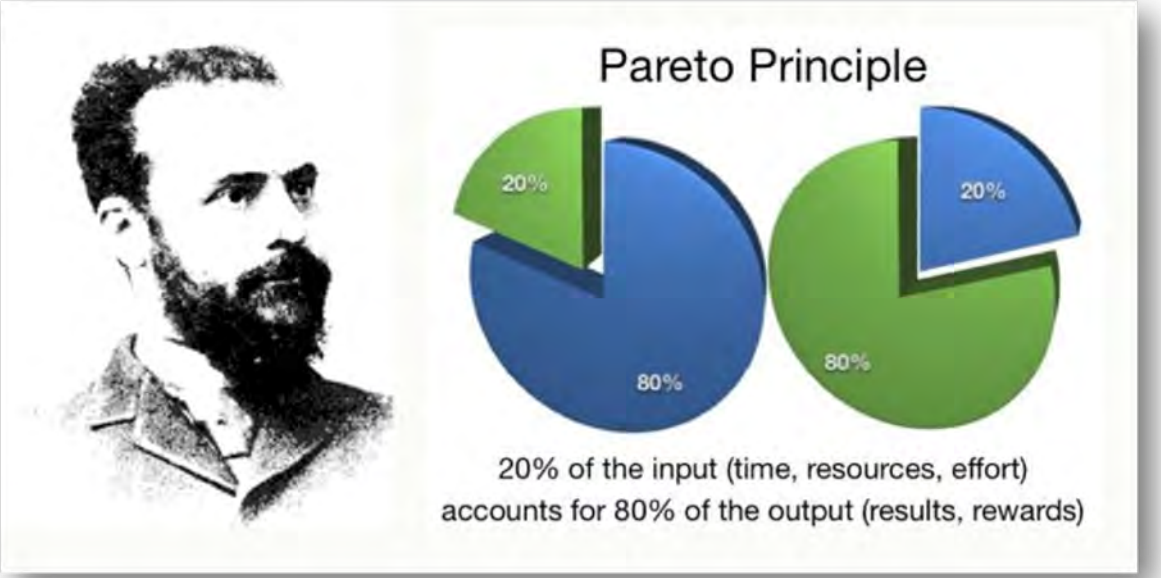


- Plot of Cumulative Frequencies (Ogive)

```
> #create vector of y coordinates to plot
> cumfreq1<-c(0,cumfreq_table$cumfreq) #start with 0
> plot(breaks,cumfreq1,xlab="Room Size (Square Feet)",ylab="Cumulative Frequency",
main="Cumulative Frequency for Room Size")
> lines(breaks, cumfreq1)
```

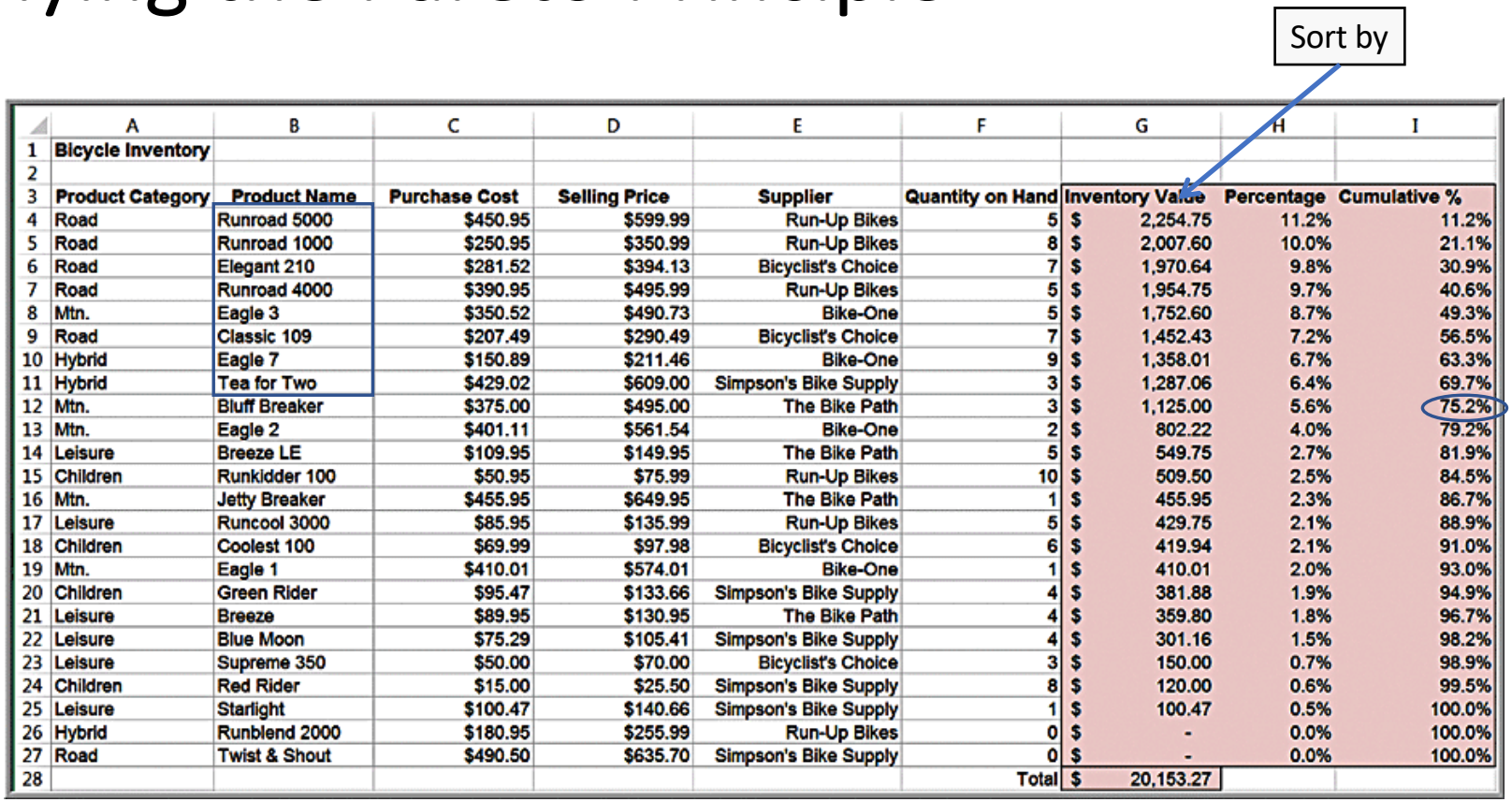


Pareto Analysis



- An Italian economist, Vilfredo Pareto, observed in 1906 that a large proportion of wealth in Italy was owned by a small proportion of people.
- Similarly, businesses often find a large proportion of sales come from a small percentage of customers, a large percentage of quality defects stems from just a couple of sources, or a large percentage of inventory value corresponds to a small percentage of items
- A Pareto analysis involves sorting data and calculating cumulative proportions.

Applying the Pareto Principle



	A	B	C	D	E	F	G	H	I
1	**Bicycle Inventory**								
2									
3	**Product Category**	**Product Name**	**Purchase Cost**	**Selling Price**	**Supplier**	**Quantity on Hand**	**Inventory Value**	**Percentage**	**Cumulative %**
4	Road	Runroad 5000	$450.95	$599.99	Run-Up Bikes	5	$ 2,254.75	11.2%	11.2%
5	Road	Runroad 1000	$250.95	$350.99	Run-Up Bikes	8	$ 2,007.60	10.0%	21.1%
6	Road	Elegant 210	$281.52	$394.13	Bicyclist's Choice	7	$ 1,970.64	9.8%	30.9%
7	Road	Runroad 4000	$390.95	$495.99	Run-Up Bikes	5	$ 1,954.75	9.7%	40.6%
8	Mtn.	Eagle 3	$350.52	$490.73	Bike-One	5	$ 1,752.60	8.7%	49.3%
9	Road	Classic 109	$207.49	$290.49	Bicyclist's Choice	7	$ 1,452.43	7.2%	56.5%
10	Hybrid	Eagle 7	$150.89	$211.46	Bike-One	9	$ 1,358.01	6.7%	63.3%
11	Hybrid	Tea for Two	$429.02	$609.00	Simpson's Bike Supply	3	$ 1,287.06	6.4%	69.7%
12	Mtn.	Bluff Breaker	$375.00	$495.00	The Bike Path	3	$ 1,125.00	5.6%	75.2%
13	Mtn.	Eagle 2	$401.11	$561.54	Bike-One	2	$ 802.22	4.0%	79.2%
14	Leisure	Breeze LE	$109.95	$149.95	The Bike Path	5	$ 549.75	2.7%	81.9%
15	Children	Runkidder 100	$50.95	$75.99	Run-Up Bikes	10	$ 509.50	2.5%	84.5%
16	Mtn.	Jetty Breaker	$455.95	$649.95	The Bike Path	1	$ 455.95	2.3%	86.7%
17	Leisure	Runcool 3000	$85.95	$135.99	Run-Up Bikes	5	$ 429.75	2.1%	88.9%
18	Children	Coolest 100	$69.99	$97.98	Bicyclist's Choice	6	$ 419.94	2.1%	91.0%
19	Mtn.	Eagle 1	$410.01	$574.01	Bike-One	1	$ 410.01	2.0%	93.0%
20	Children	Green Rider	$95.47	$133.66	Simpson's Bike Supply	4	$ 381.88	1.9%	94.9%
21	Leisure	Breeze	$89.95	$130.95	The Bike Path	4	$ 359.80	1.8%	96.7%
22	Leisure	Blue Moon	$75.29	$105.41	Simpson's Bike Supply	4	$ 301.16	1.5%	98.2%
23	Leisure	Supreme 350	$50.00	$70.00	Bicyclist's Choice	3	$ 150.00	0.7%	98.9%
24	Children	Red Rider	$15.00	$25.50	Simpson's Bike Supply	8	$ 120.00	0.6%	99.5%
25	Leisure	Starlight	$100.47	$140.66	Simpson's Bike Supply	1	$ 100.47	0.5%	100.0%
26	Hybrid	Runblend 2000	$180.95	$255.99	Run-Up Bikes	0	$ -	0.0%	100.0%
27	Road	Twist & Shout	$490.50	$635.70	Simpson's Bike Supply	0	$ -	0.0%	100.0%
28						Total	$ 20,153.27		

75% of the bicycle inventory value comes from 40% (9/24) of items.

Percentiles

- k^{th} percentile is a value at or below which at least k percent of the observations lie.
- Most common way to compute the k^{th} percentile is to order the data values from smallest to largest and calculate the rank of the k^{th} percentile using the formula:

$$\frac{nk}{100} + 0.5$$

Computing Percentiles

- Compute the k^{th} percentile for *a variable in sample size n*
- Rank of k^{th} percentile = $nk/100 + 0.5$
  - $n = 94$; $k = 90$
  - For the 90^{th} percentile, rank is

    $= 94(90)/100+0.5 = 85.1$ (round to 85)
  - Value of the 85^{th} observation

Now let's use R to compute the $32^{th,}$ 57^{th}, 98^{th} percentile for *Room Size*

```
> quantile(RmSize, c(.32, .57, .98))
    32%     57%     98%
1527.32 1673.92 2372.00
```

Quartiles

- Quartiles break the data into four parts.
  - 25th percentile is first quartile,Q1;
  - 50th percentile is second quartile, Q2;
  - 75th percentile is third quartile, Q3; and
  - 100th percentile is fourth quartile, Q4.
- One-fourth of the data fall below the first quartile, one-half are below the second quartile, and three-fourths are below the third quartile.

Let's use R to compute the 4 quartiles for Home Size

```
> quantile(RmSize, c(.25, .5, .75, 1))
    25%     50%     75%    100%
1520.00 1666.00 1806.75 2372.00
```

Contingency Tables

- One of most basic statistical tool for summarizing categorical data
- A tabular method that displays number of observations in a data set for different subcategories of two or more categorical variables.
- Contingency tables can accept numerical variables but grouping variable must be categorical.
- Subcategories of variables must be mutually exclusive and exhaustive (i.e. each observation can be classified into only one subcategory, and, taken together over all subcategories, they must constitute the complete data set)

Examples of Contingency Tables

Class rank * Do you live on campus? Crosstabulation

Count

		Do you live on campus?		
		Off-campus	On-campus	Total
Class rank	Freshman	37	100	137
	Sophomore	42	48	90
	Junior	90	8	98
	Senior	62	1	63
Total		231	157	388

When love fails

Percentage of marriages that ended in divorce or annulment before the fifth anniversary

Groom's highest educational qualification	Couples who wed in 2007		Couples who wed in 2008		Couples who wed in 2009	
	Resident marriages*	Singaporeans wed to non-residents**	Resident marriages*	Singaporeans wed to non-residents**	Resident marriages*	Singaporeans wed to non-residents**
Total	6.8	7.1	6.6	7.2	6.4	7.9
Below secondary	8.7	6.5	8.5	5.9	7.8	6.7
Secondary	9.9	7.7	10.1	8.5	9.3	8.5
Post-secondary	7.3	7.5	7.1	8.0	7.5	9.7
University	3.8	5.8	3.5	5.0	3.2	5.3

NOTE: *Involving at least one Singapore citizen or permanent resident
** Foreigners such as those on long-term visit pass

Source: DEPARTMENT OF STATISTICS SUNDAY TIMES GRAPHICS

Constructing a Contingency table for 2 categorical variables

DATA: Home_Market_Value_Type_R (assigned to HomeTR)

Categorical variables

	House Age	Square Feet	Market Value	Type	Region	Sub-Reg
1	33	1812	90000	A1	1	U
2	32	1914	104400	A2	1	X
3	32	1842	93300	B2	1	Z
4	33	1812	91000	A1	1	U
5	32	1836	101900	A1	2	U
6	33	2028	108500	A1	2	U
7	32	1732	87600	A2	2	U
8	33	1850	96000	A2	2	U
9	32	1791	89200	B2	3	U
10	33	1666	88400	A1	3	U
11	32	1852	100800	A1	3	U
12	32	1620	96700	A1	3	U
13	32	1692	87500	A1	4	U
14	32	2372	114000	A1	4	U

Showing 1 to 14 of 42 entries

- Count number of units by type and region

```
> HomeTR<-Home_Market_Value_Type_Reg
> table(HomeTR$Type)
 A1 A2 B1 B2  C
 12 10  8  5  7
> table(HomeTR$Region)
 1  2  3  4
 8 14  9 11
```

row var column var

```
> table(HomeTR$Type, HomeTR$Region)
   1 2 3 4
A1 3 4 3 2
A2 1 4 1 4
B1 1 2 1 4
B2 1 2 2 0
C  2 2 2 1
```

```
> con.table.tol<-NA
> con.table<-table(HomeTR$Type, HomeTR$Region)
> tot.col<-rowSums(con.table)
> con.table.tol<-cbind(con.table,tot.col)
> tot.row<-colSums(con.table.tol)
> con.table.tol<-rbind(con.table.tol,tot.row)
        1  2 3  4 tot.col
A1      3  4 3  2      12
A2      1  4 1  4      10
B1      1  2 1  4       8
B2      1  2 2  0       5
C       2  2 2  1       7
tot.row 8 14 9 11      42
```

Constructing a Contingency table for 2 categorical variables

DATA: Home_Market_Value_Type_R (assigned to HomeTR)

Categorical variables

	House Age	Square Feet	Market Value	Type	Region	Sub-Reg
1	33	1812	90000	A1	1	U
2	32	1914	104400	A2	1	X
3	32	1842	93300	B2	1	Z
4	33	1812	91000	A1	1	U
5	32	1836	101900	A1	2	U
6	33	2028	108500	A1	2	U
7	32	1732	87600	A2	2	U
8	33	1850	96000	A2	2	U
9	32	1791	89200	B2	3	U
10	33	1666	88400	A1	3	U
11	32	1852	100800	A1	3	U
12	32	1620	96700	A1	3	U
13	32	1692	87500	A1	4	U
14	32	2372	114000	A1	4	U

Showing 1 to 14 of 42 entries

- Percentage of units by type and region.

```
> prop.table(table(HomeTR$Type, HomeTR$Region))

             1          2          3          4
A1 0.07142857 0.09523810 0.07142857 0.04761905
A2 0.02380952 0.09523810 0.02380952 0.09523810
B1 0.02380952 0.04761905 0.02380952 0.09523810
B2 0.02380952 0.04761905 0.04761905 0.00000000
C  0.04761905 0.04761905 0.04761905 0.02380952
```

total of all cells equal to 100%

- Percentage of one variable within groups of another variable (column or row)

```
> # Contingency table of distribution of Type across Regions
> prop.table(table(HomeTR$Type, HomeTR$Region),margin=2)

             1          2          3          4
A1 0.37500000 0.28571429 0.33333333 0.18181818
A2 0.12500000 0.28571429 0.11111111 0.36363636
B1 0.12500000 0.14285714 0.11111111 0.36363636
B2 0.12500000 0.14285714 0.22222222 0.00000000
C  0.25000000 0.14285714 0.22222222 0.09090909
```

Constructing Contingency Tables using rPivotTable package

```
rpivotTable(data, rows = NULL, cols = NULL, aggregatorName = NULL,
  vals = NULL, rendererName = NULL, sorter = NULL, exclusions = NULL,
  inclusions = NULL, locale = "en", subtotals = FALSE, ..., width = 800,
  height = 600, elementId = NULL)
```

Arguments

`data`	data.frame or data.table (R>=1.9.6 for safety) with data to use in the pivot table
`rows`	String name of the column in the data.frame to prepopulate the **rows** of the pivot table.
`cols`	String name of the column in the data.frame to prepopulate the **columns** of the pivot table.
`aggregatorName`	String name of the pivottable.js aggregator to prepopulate the pivot table.
`vals`	String name of the column in the data.frame to use with `aggregatorName`. Must be additive (i.e a number).
`rendererName`	List name of the renderer selected, e.g. Table, Heatmap, Treemap etc.

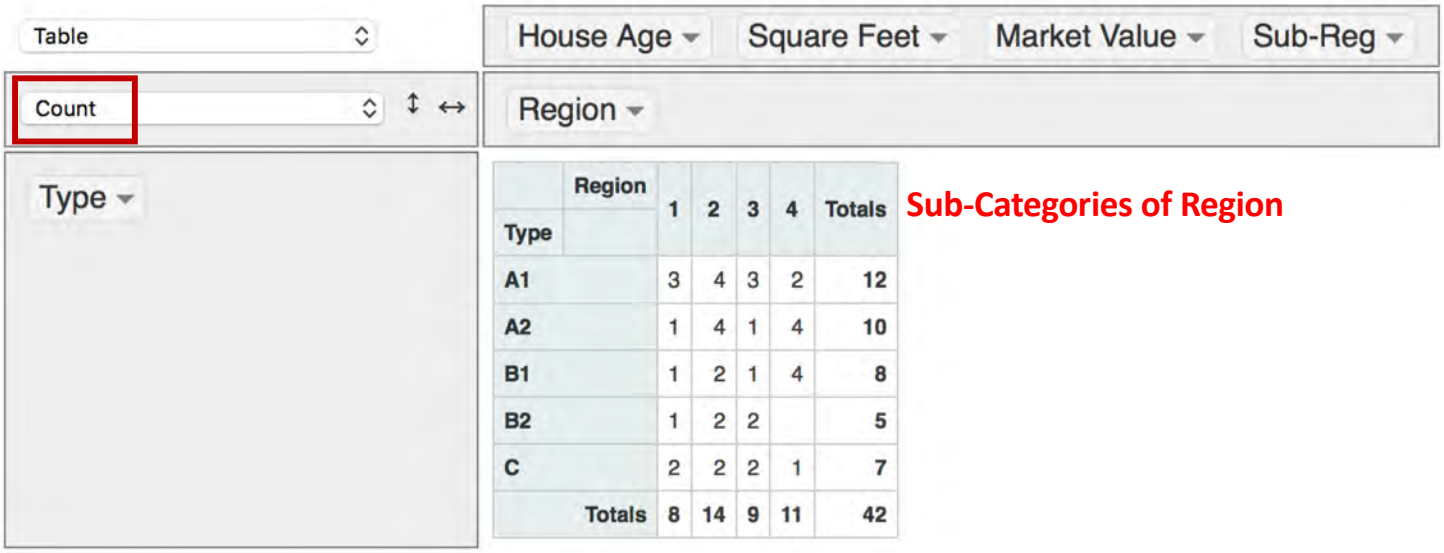
Options: Count, Count Unique Values, List Unique Values, Sum, Integer Sum, Average, Sum over Sum, 80% Upper Bound, 80% Lower Bound, Sum as Fraction of Total, Sum as Fraction of Rows, Sum as Fraction of Columns, Count as Fraction of Total, Count as Fraction of Rows, Count as Fraction of Columns

Reference: Help rpivotTable & https://cran.r-project.org/web/packages/rpivotTable/vignettes/rpivotTableIntroduction.html

Constructing Contingency Tables using rPivotTable package

- Count number of units by type and region.

```
rpivotTable(HomeTR,rows=c("Type"),cols=c("Region"))
```



Type	Region 1	2	3	4	Totals
A1	3	4	3	2	12
A2	1	4	1	4	10
B1	1	2	1	4	8
B2	1	2	2		5
C	2	2	2	1	7
Totals	8	14	9	11	42

Sub-Categories of Region

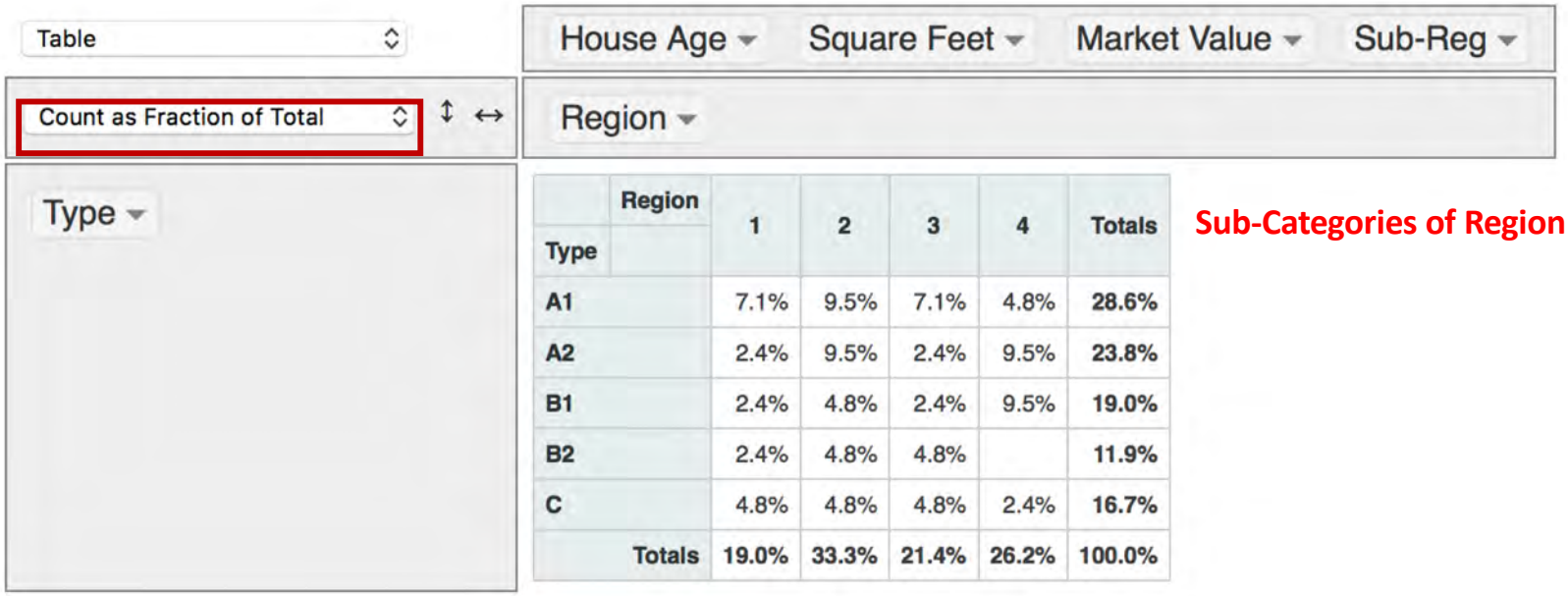
Sub-Categories of Unit Type

https://cran.r-project.org/web/packages/rpivotTable/vignettes/rpivotTableIntroduction.html

Constructing Contingency Tables using rPivotTable package

- Percentage of units over total by type and region.

```
> rpivotTable(HomeTR,rows=c("Type"),cols=c("Region"))
> rpivotTable(HomeTR,rows=c("Type"),cols=c("Region"), aggregatorName = "Count as
Fraction of Total")
```



Type \ Region	1	2	3	4	Totals
A1	7.1%	9.5%	7.1%	4.8%	28.6%
A2	2.4%	9.5%	2.4%	9.5%	23.8%
B1	2.4%	4.8%	2.4%	9.5%	19.0%
B2	2.4%	4.8%	4.8%		11.9%
C	4.8%	4.8%	4.8%	2.4%	16.7%
Totals	19.0%	33.3%	21.4%	26.2%	100.0%

Sub-Categories of Region

Sub-Categories of Unit Type

Constructing a Pivot table for 3 categorical vars

- Count number of units by type, region, and sub-region.

```
> rpivotTable(HomeTR,rows=c("Type"),cols=c("Region", "Sub-Reg"), aggregatorName = "Count as Fraction of Total")
```

Table | House Age | Square Feet | Market Value

Count as Fraction of Total | Region | Sub-Reg

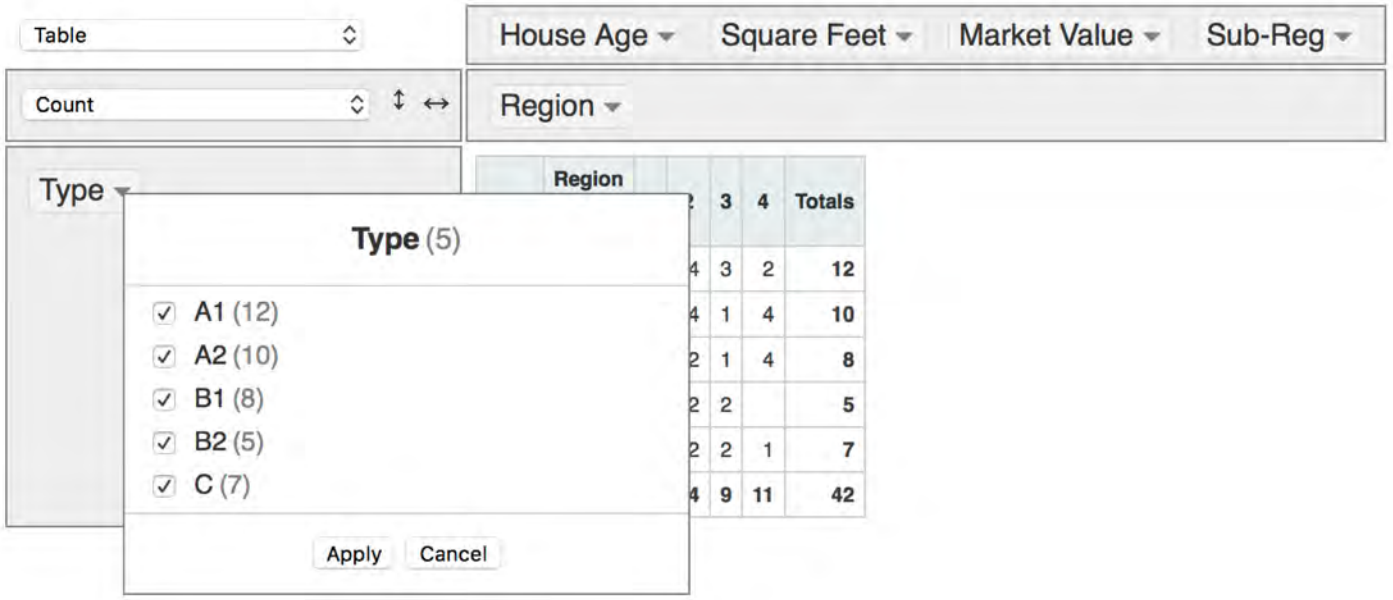
Type

	Region	1			2			3			4		Totals
Type	Sub-Reg	U	X	Z	U	X	Z	U	X	Z	U	Z	
A1		7.1%			7.1%		2.4%	7.1%			4.8%		28.6%
A2			2.4%		4.8%	4.8%			2.4%		4.8%	4.8%	23.8%
B1				2.4%	2.4%	2.4%		2.4%			2.4%	7.1%	19.0%
B2				2.4%			4.8%	2.4%		2.4%			11.9%
C		4.8%					4.8%	2.4%		2.4%		2.4%	16.7%
	Totals	11.9%	2.4%	4.8%	14.3%	7.1%	11.9%	14.3%	2.4%	4.8%	11.9%	14.3%	100.0%

Sub-Categories of sub-region

Slicers

- for drilling down to “slice” a PivotTable and display a subset of data



Slicers

- for drilling down to “slice” a PivotTable and display a subset of data