Chapter 2
Location, Variability and Process

Section 2.4
Analysis of Processes over Time:
The Time Series Visualization

David W. Gerbing

The School of Business
Portland State University

- Time Series Visualization
  - Create the Time Series Visualization
  - A Single Time Series with Color Customizations
  - HCL Colors for Data Visualization
  - Multiple Time Series on a Single Panel
  - Multiple Time Series on Different Panels

2.4a
Create the Time Series Visualization

Run Chart and Time Series Chart

Graphical display of variation over time

- **Time Series chart**: Plotted sequence of values against the corresponding dates and/or times at which the values were recorded, usually at regular intervals
- The time series is one of the basic concepts for forecasting
- Ex: ship times, inventory levels
- Goal:
  - Discover: The underlying structure, disentangled from the random variation
  - Forecast: Extrapolate this structure into the future

Accessing Historical Share Prices

Go to the web, `finance.yahoo.com`

- Many business data sources, including time series data such as share prices, are available on the web, such as Link: `http://finance.yahoo.com`, and access as . . .

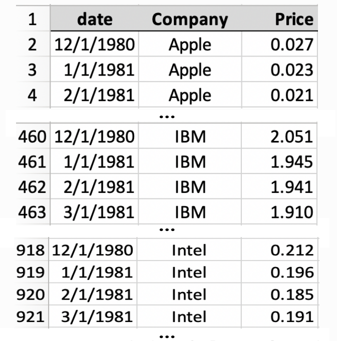
1. To left of the `Search` button near top-left of the home page, enter stock symbol, such as AAPL, then click button
2. At the center of the horizontal list of choices, located below the large font stock price, click link: `Historical Data`
3. Choose a Time Period and Frequency, such as `Monthly`,
4. Under the Apply button, click: `Download Data`
5. The result is the file `AAPL.csv` with variable Adj Close, which reflects actions such as stock splits and dividends

- The monthly share prices from Apple are used in the following examples, available from Link: Apple share Price History

Time Series Data in Excel

Apple, IBM, Intel monthly from 12/1/1980 through 1/1/2019

- Price per share adjusted for stock splits
- date column in Excel displayed as an Excel date format, which translates into an R Date type when read by lessR `Read()`
- Company has three values: Apple, IBM, and Intel



1	date	Company	Price
2	12/1/1980	Apple	0.027
3	1/1/1981	Apple	0.023
4	2/1/1981	Apple	0.021
		...	
460	12/1/1980	IBM	2.051
461	1/1/1981	IBM	1.945
462	2/1/1981	IBM	1.941
463	3/1/1981	IBM	1.910
		...	
918	12/1/1980	Intel	0.212
919	1/1/1981	Intel	0.196
920	2/1/1981	Intel	0.185
921	3/1/1981	Intel	0.191
		...	

Time Series Data

Apple, IBM, Intel monthly from 12/1/1980 through 1/1/2019

```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
```

or, if off the Internet, already downloaded with lessR,

```
d <- Read("StockPrice")
```

```
  Data Types
------------------------------------------------
character: Non-numeric data values
Date: Date with year, month and day
double: Numeric data values with decimal digits
------------------------------------------------


   Variable                  Missing  Unique
       Name      Type  Values  Values  Values  First and ...
---------------------------------------------------------------
 1     date      Date    1374       0     458  1980-12-01 ...
 2  Company character    1374       0       3  Apple  Apple ...
 3    Price    double    1374       0    1259  0.027  0.023 ...
---------------------------------------------------------------
```

Two Forms of the Data to Plot a Time Series

I. Plot Y vs. date

- Y is a numeric variable, such as share price, plot on y-axis
- X: is an R variable of type Date
  - If data from an Excel file, and the date variable is formatted as an Excel date, then Read() reads the Excel date as an R date (see previous slide)
  - If date variable in Excel formatted as a character string, or read from a text file such as a csv file, then the date field is read as a character string and must be converted to an R Date type with as.Date (explained later)

II. Plot Y as an R type ts variable

- Express the variable Y as an R variable type ts (for time series) with the ts function
- This expression of the data is needed for the widely used forecast package to generate forecasts

2.4b
A Single Time Series with Color Customizations

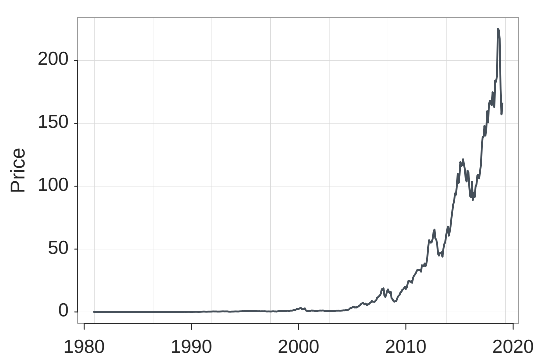
Set Up the Time Series Visualization

Plot share price vs. date

- function: Plot with the lessR `Plot()` function
- x-variable: date
- y-variable: Price
- `rows`: Select specified rows from the data frame for analysis according to a logical condition
  - The R double equal sign, == means is equal to
  - The == does *not* set to equality, it evaluates equality, resulting in a value that is either `TRUE` or `FALSE`
  - (Company=="Apple") evaluates to `TRUE` only for those rows of data for which the data value for the variable Company equals "Apple"

Time Series Plot - Apple Only

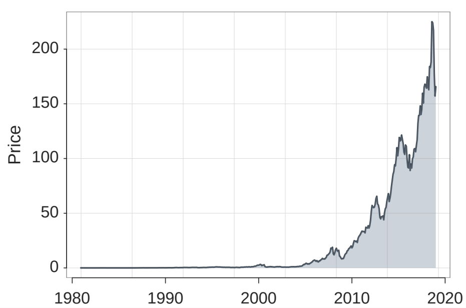
```
Plot(date, Price, rows=(Company=="Apple"))
```



Time Series Plot - The `area_fill` Parameter

- The **fill** parameter in general, and `area_fill` in particular, refer to interior regions, here the area under the curve, how the object is *viewed from the inside*
- `fill`: set to `"on"` indicates to fill the area under the time series curve with the default color of the current color theme

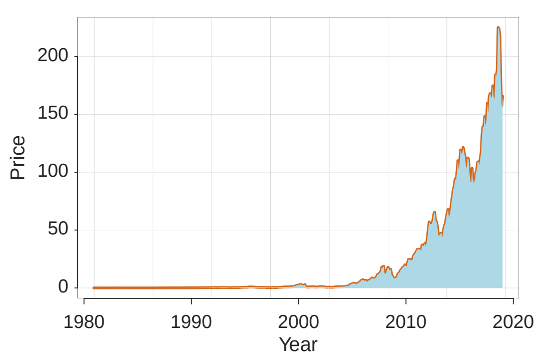
```
Plot(date, Price, rows=(Company=="Apple"), area_fill="on")
```



Time Series Plot - The color Parameter

- The **color** parameter sets the color of a line, edge, or border, how the object is *viewed from the outside*
- The **lwd** parameter sets the line width of the line

```
Plot(date, Price, rows=(Company=="Apple"),
      color="chocolate", lwd=3, area_fill="lightblue")
```

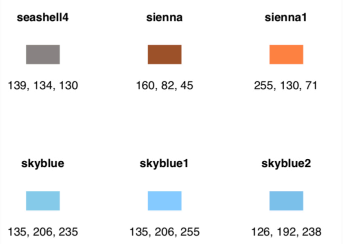


R Named Colors

- colors() is the R function for listing 100's of named colors

```
> colors()
  [1] "white"             "aliceblue"           "antiquewhite"
  [4] "antiquewhite1"     "antiquewhite2"       "antiquewhite3"
  [7] "antiquewhite4"     "aquamarine"          "aquamarine1"
 [10] "aquamarine2"       "aquamarine3"         "aquamarine4"  ...
```

- showColors() is the lessR function for listing the named colors plus the color display, and red, green, blue components

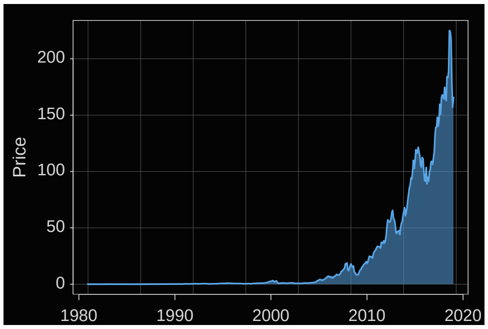


Time Series Plot - WSJ Blue Color Theme

Begin with a numeric variable, then specify dates

- Set trans to set the transparency level as a proportion

```
style(sub_theme="black")
Plot(date, Price, rows=(Company=="Apple"),
      color="steelblue2", area_fill="steelblue3", trans=.55)
```



2.4c
HCL Colors for Data Visualization

A Problem for Data Visualizations

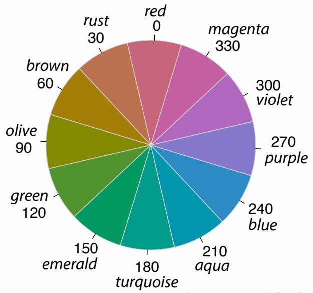
- Unless carefully chosen, different colors have different levels of intensity and brightness
- These different brightnesses lead to biased perception of different aspects of a visualization, such as the bars in a bar chart, which become more readily apparent than comparable features of the visualization that are darker or less visible
- Comparable visual aesthetics should display at the same brightness, with the same tone of gray if converted to grayscale
- This problem is often realized because computer monitors display colors according to levels of the three primary colors – red, green, and blue, but ...
  - Unfortunately the red, green, and blue components do not evenly match in terms of the perception of brightness
  - For example, using the R function `rgb` to define a mostly red color, rgb(.9, .1, .1), does not display at the same level of brightness of a comparable specification of a mostly blue color, rgb(.1, .1, .9)

Colors for Data Visualizations: HCL

Human perception dimensions of color from R `hcl()` function

- Hue, `h`: Color name from the rainbow, from 0 to 360 degrees
- Chroma, `c`: Intensity of color from gray to colorful, 0 to 100
- Luminance, `l`: Brightness, 0 to 100

```
getColors(shape="wheel", n=12) # lessR function
```

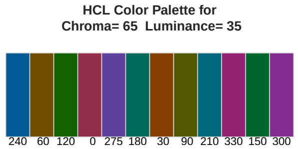


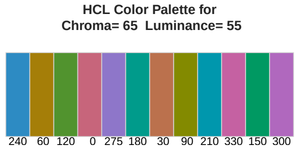
HCL Qualitative Palette: Chroma=65, Luminance Varies

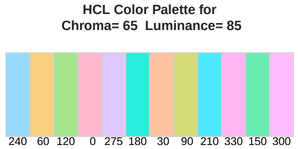
getColors(c=65, l=35)

getColors(c=65, l=55)

getColors(c=65, l=85)





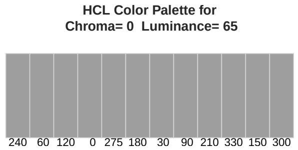


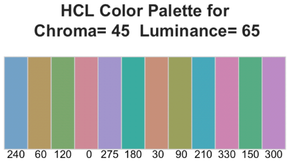
HCL Qualitative Palette: Chroma Varies, Luminance=65

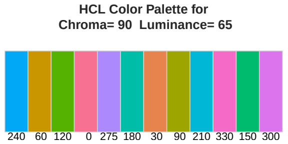
getColors(c=0, l=65)

getColors(c=45, l=65)

getColors(c=90, l=65)







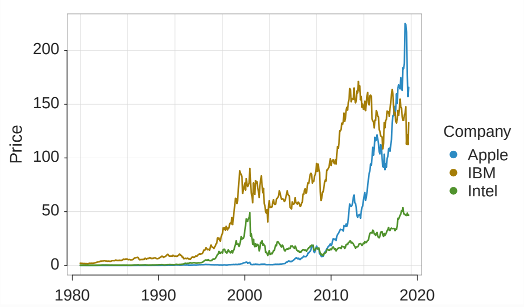
2.4d
Multiple Time Series on a Single Panel

Multiple Time Series Plot

Begin with a numeric variable, then specify dates

- The date in the input Excel file already formatted as a date
- The **by** parameter signals multiple plots on the same panel

```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
Plot(date, Price, by=Company)
```

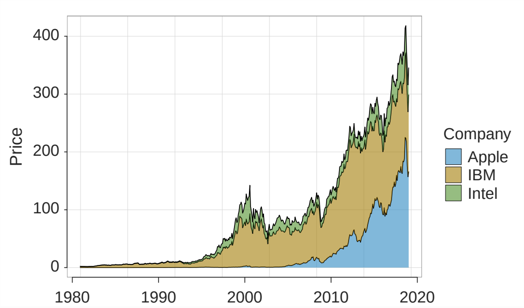


Stacked Multiple Time Series Plot

Begin with a numeric variable, then specify dates

- Set **stack** to TRUE to stack the plots on top of each other

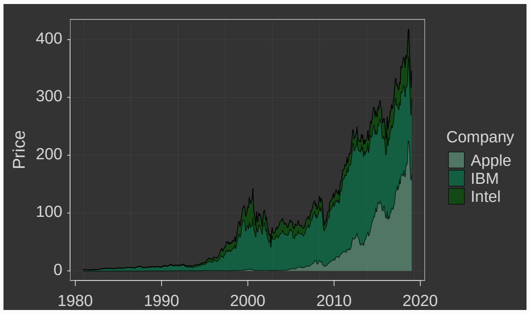
```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
Plot(date, Price, by=Company, stack=TRUE, trans=0.4)
```



Stacked Multiple Time Series Plot, Greens

Define a sequential color palette

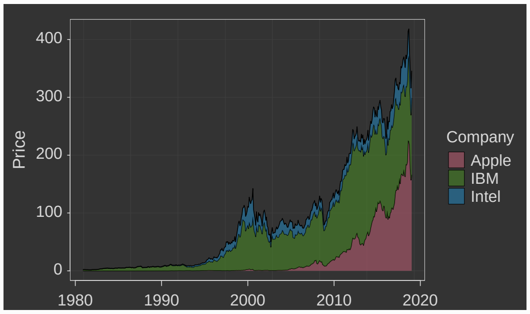
```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
Plot(date, Price, by=Company, stack=TRUE, trans=0.4,
     fill="greens")
```



Stacked Multiple Time Series Plot, Bright Colors

Qualitative color palette with black background

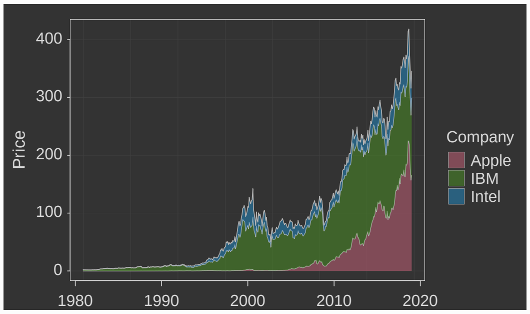
```
Plot(date, Price, by=Company, stack=TRUE, trans=0.4)
```



Stacked Multiple Time Series Plot, Light Lines

Customize the time series lines

```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
Plot(date, Price, by=Company, stack=TRUE, trans=0.4,
     color="gray85")
```

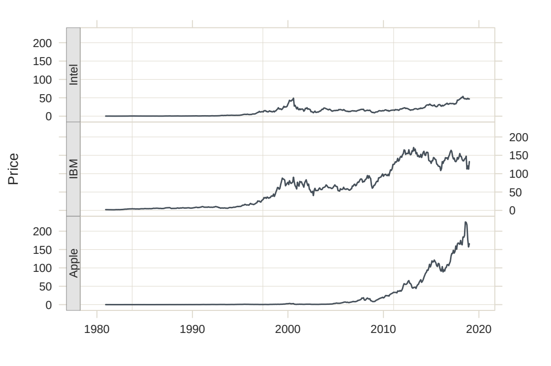


2.4e
Multiple Time Series on Different Panels

Trellis Time Series Plot

Plot parameter by1 creates separate panels for Trellis plots

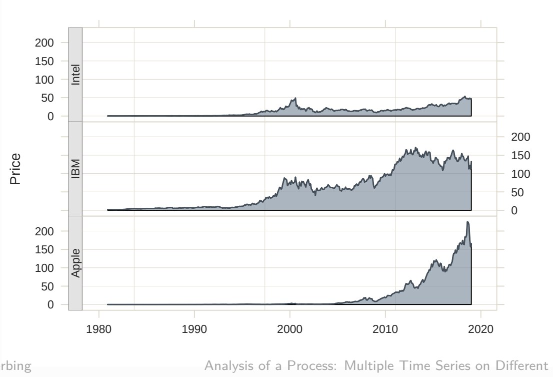
```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
Plot(date, Price, by1=Company)
```



Trellis Time Series Plot

Can also add a fill color

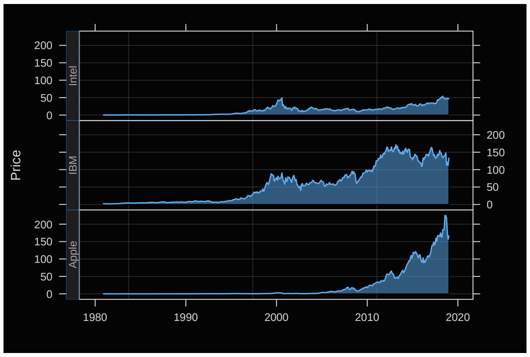
```
d <- Read("http://lessRstats.com/data/PPStechLong.xlsx")
Plot(date, Price, by1=Company, area_fill="on", trans=0.4)
```



Trellis Time Series Plot - WSJ Blue Theme

Follow the same style function call as for just Apple

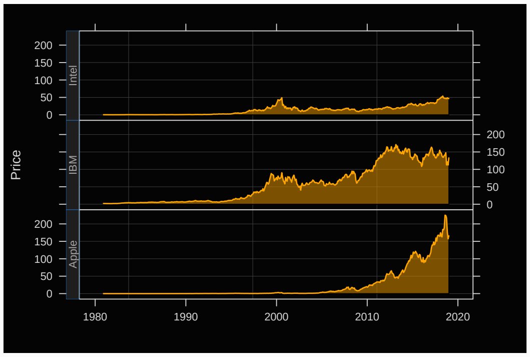
```
style(sub_theme="black")
Plot(date, Price, by1=Company,
       color="steelblue2", area_ fill="steelblue3", trans=.55)
```



Trellis Time Series Plot - Orange Theme

Can customize to any set of colors

```
style(sub_theme="black")
Plot(date, Price, by1=Company,
      color="orange", area_fill="orange3", trans=.55)
```



Index

Subtract 2 from each listed value to get the Slide #

- The End