Recap From Last Time:

- What is R and why should we use it?
- Familiarity with R's basic syntax.
- Familiarity with major R data structures namely **vectors** and **data.frames**.
- Understand the basics of using **functions** (arguments, vectorization and re-cycling).
- Appreciate how you can use R scripts to aid with reproducibility.

DataCamp Homework Reminder!!

Today’s Learning Goals

- Appreciate the major elements of **exploratory data analysis** and why it is important to visualize data.
- Be conversant with **data visualization best practices** and understand how good visualizations optimize for the human visual system.
- Be able to generate informative graphical displays including **scatterplots**, **histograms**, **bar graphs**, **boxplots**, **dendrograms** and **heatmaps** and thereby gain exposure to the extensive graphical capabilities of R.
- Appreciate that you can build even more complex charts with **ggplot** and additional R packages such as **rgl**.
Why visualize at all?

\[
\begin{array}{|c|c|c|}
\hline
x & y \\
\hline
1 & 5.00 & 0.00 \\
2 & 4.18 & 2.75 \\
3 & 1.98 & 4.59 \\
4 & -0.86 & 4.92 \\
5 & -3.43 & 3.64 \\
6 & -4.86 & 1.16 \\
7 & -4.70 & -1.70 \\
8 & -2.99 & -4.01 \\
9 & -0.30 & -4.99 \\
10 & 2.49 & -4.34 \\
11 & 4.46 & -2.25 \\
12 & 4.97 & 0.57 \\
13 & 3.84 & 3.20 \\
14 & 1.45 & 4.79 \\
15 & -1.42 & 4.79 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
x & y \\
\hline
Min. & -4.86 & -4.99 \\
1st Qu. & -2.21 & -1.98 \\
Median & 1.45 & 1.16 \\
Mean & 0.65 & 0.87 \\
3rd Qu. & 4.01 & 4.12 \\
Max. & 5.00 & 4.92 \\
\hline
\end{array}
\]
Exploratory Data Analysis

- ALWAYS look at your data!
- If you can't see it, then don't believe it!
- Exploratory Data Analysis (EDA) allows us to:
  1. Visualize distributions and relationships
  2. Detect errors
  3. Assess assumptions for confirmatory analysis
- EDA is the first step of data analysis!
Exploratory Data Analysis 1977

- Based on insights developed at Bell Labs in the 60's
- Techniques for visualizing and summarizing data
- What can the data tell us? (in contrast to "confirmatory" data analysis)
- Introduced many basic techniques:
  - 5-number summary, box plots, stem and leaf diagrams,...
- 5 Number summary:
  - extremes (min and max)
  - median & quartiles
  - More robust to skewed & longtailed distributions

Side-note: How to summarize data?

Side-note: Mean & standard deviation

Fine for normally distributed data

\[
x \leftarrow \text{rnorm}(1000)
\]

\[
\text{mean}(x)
\]

\[
\text{sd}(x)
\]

Side-note: 5 number summary

Minimum, Q1, Q2, Q3, and maximum

\[
x \leftarrow \text{rnorm}(1000)
\]

\[
\text{mean}(x)
\]

\[
\text{sd}(x)
\]

\[
\text{summary}(x)
\]
Side-note: 5 number summary
Minimum, Q1, Q2, Q3, and maximum

x <- rnorm(1000)
mean(x)
sd(x)
summary(x)

Q1 = first quartile
Q2 = median
Q3 = third quartile
IQR = interquartile range
minimum = smallest value
maximum = largest value
Side-note: boxplot
Graphical form of the 5 number summary!

Also called box-and-whisker plots; See also violin plots etc.

The Trouble with Summary Stats

<table>
<thead>
<tr>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
<th>Set D</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>10</td>
<td>8.04</td>
<td>10</td>
<td>9.14</td>
</tr>
<tr>
<td>8</td>
<td>6.95</td>
<td>8</td>
<td>8.14</td>
</tr>
<tr>
<td>13</td>
<td>7.59</td>
<td>13</td>
<td>8.74</td>
</tr>
<tr>
<td>9</td>
<td>8.81</td>
<td>9</td>
<td>8.77</td>
</tr>
<tr>
<td>11</td>
<td>8.33</td>
<td>11</td>
<td>9.36</td>
</tr>
<tr>
<td>14</td>
<td>9.95</td>
<td>14</td>
<td>8.1</td>
</tr>
<tr>
<td>6</td>
<td>7.24</td>
<td>6</td>
<td>6.13</td>
</tr>
<tr>
<td>4</td>
<td>4.25</td>
<td>4</td>
<td>2.1</td>
</tr>
<tr>
<td>12</td>
<td>10.84</td>
<td>12</td>
<td>9.11</td>
</tr>
<tr>
<td>7</td>
<td>4.82</td>
<td>7</td>
<td>7.26</td>
</tr>
<tr>
<td>5</td>
<td>5.68</td>
<td>5</td>
<td>4.74</td>
</tr>
</tbody>
</table>

Summary Statistics Linear Regression
\[ y = 9.0 + 2.03 x \]
\[ R^2 = 0.67 \]

https://en.wikipedia.org/wiki/Anscombe%27s_quartet

Looking at Data
Today’s Learning Goals

- Appreciate the major elements of **exploratory data analysis** and why it is important to visualize data.

- Be conversant with **data visualization best practices** and understand how good visualizations optimize for the human visual system.

- Be able to generate informative graphical displays including scatterplots, histograms, bar graphs, boxplots, dendrograms and heatmaps and thereby gain exposure to the extensive graphical capabilities of R.

- Appreciate that you can build even more complex charts with ggplot and additional R packages such as rgl.
Key Point: Good visualizations optimize for the human visual system.

Key Point: The most important measurement should exploit the highest ranked encoding possible

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- Angle or Slope
- Area
- Volume or Density or Color saturation/hue
Key Point: The most important measurement should exploit the highest ranked encoding possible

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- Angle or Slope
- Area
- Volume or Density or Color saturation/hue
Observation: Alphabetical is almost never the correct ordering of a categorical variable.
If we did not have the legend would you know which was low or high mpg?

The most important measurement should exploit the highest ranked encoding possible.

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- Angle or Slope
- Area
- Volume or Density or Color saturation/hue
The most important measurement should exploit the highest ranked encoding possible.

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- **Angle or Slope**
- Area
- Volume or Density or Color saturation/hue
If growth (slope) is important, plot it directly.
The most important measurement should exploit the highest ranked encoding possible.

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- **Angle** or Slope
- Area
- Volume or Density or Color saturation/hue

**Observation:** Pie charts are **ALWAYS** a mistake.

Apart from MPAs :-)

Piecharts are the information visualization equivalent of a roofing hammer to the frontal lobe. They have no place in the world of grownups, and occupy the same semiotic space as short pants, a runny nose, and chocolate smeared on one’s face. They are as professional as a pair of assless chaps.


Tables are preferable to graphics for many small data sets. A table is nearly always better than a dumb pie chart; the only thing worse than a pie chart is several of them, for then the viewer is asked to compare quantities located in spatial disarray both within and between pies... Given their low data-density and failure to order numbers along a visual dimension, pie charts should never be used.

-Edward Tufte, The Visual Display of Quantitative Information
Tables are preferable to graphics for many small data sets. A table is nearly always better than a dumb pie chart; the only thing worse than a pie chart is several of them, for then the viewer is asked to compare quantities located in spatial disarray both within and between pies... Given their low data-density and failure to order numbers along a visual dimension, pie charts should never be used.

-Edward Tufte, The Visual Display of Quantitative Information

Who do you think did a better job in tonight’s debate?

<table>
<thead>
<tr>
<th></th>
<th>Clinton</th>
<th>Trump</th>
</tr>
</thead>
<tbody>
<tr>
<td>Among Democrats</td>
<td>99%</td>
<td>1%</td>
</tr>
<tr>
<td>Among Republicans</td>
<td>53%</td>
<td>47%</td>
</tr>
</tbody>
</table>

All good pie charts are jokes...
The most important measurement should exploit the highest ranked encoding possible.

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- Angle or Slope
- Area
- Volume or Density or Color saturation/hue

Stacked anything is nearly always a mistake

This is much better...
The most important measurement should exploit the highest ranked encoding possible.

- Position along a common scale
- Position on identical but nonaligned scales
- Length
- Angle or Slope
- Area
- Volume or Density or Color saturation/hue
Observation: Comparison is trivial on a common scale.

Today’s Learning Goals

- Appreciate the major elements of exploratory data analysis and why it is important to visualize data.
- Be conversant with data visualization best practices and understand how good visualizations optimize for the human visual system.
- Be able to generate informative graphical displays including scatterplots, histograms, bar graphs, boxplots, dendrograms and heatmaps and thereby gain exposure to the extensive graphical capabilities of R.
- Appreciate that you can build even more complex charts with ggplot and additional R packages such as rgl.
Different graphs for different purposes

**Exploratory graphs**: many images for a narrow audience (you!)
**Presentation graphs**: single image for a large audience

---

The R Painters Model

- **Base plot**
- **Overlays**
- **Extras**

*Side-Note:* “Red and green should never be seen”

---

Core R Graph Types

**Cluster Dendrogram**
- `hclust(*, “average”)`
- `dist(USArrests)`

---

Hands-on

**Section 1 only please**

- Create a new **RStudio Project** for this class,
- **Download** the example data files and move them to your project directory,
- Focus on **Sections 1A & 1B** in the handout.
Hands-on
Section 1 only please

- Create a new RStudio Project for this class,
- Download the example data files and move them to your project directory,
- Focus on Sections 1A & 1B in the handout.

Common Options

- Axis scales
  - xlim c(min,max)
  - ylim c(min,max)
- Axis labels
  - xlab(text)
  - ylab(text)
- Plot titles
  - main(text)
  - sub(text)
- Plot characters
  - pch(number)
  - cex(number)

- Local options to change a specific plot
- Global options to affect all graphs

Plot Characters

```
plot(1:5, pch=1:5, cex=1:5)
```
Plot (scatterplots and line graphs)

- Input: Almost anything. 2 x Vectors
- Output: Nothing
- Options:
  ‣ type l=line, p=point, b=line+point
  ‣ lwd line width (thickness)
  ‣ lty line type (1=solid, 2=dashed, 3=dotted etc.)

```
plot(c(1:10)^2, typ="b", lwd=4, lty=3)
```

Hands-on

Section 2 Notes

➡ Focus on Sections 2A & 2B in the lab handout.
➡ Try Section 2C if you have time.
➡ See notes on the following slides...

Section 2B: Barplot (a.k.a. bar graphs)

- Input: Vector (single) or Matrix (stack or group)
- Output: Bar centre positions
- Options:
  ‣ names.arg Bar labels (if not from data)
  ‣ horiz=TRUE Plot horizontally
  ‣ beside=TRUE Plot multiple series as a group not stacked

```
barplot(VADeaths, beside = TRUE)
```
Par

- The `par()` function controls global parameters affecting all plots in the current plot area
- Changes affect all subsequent plots
- Many `par` options can also be passed to individual plots

Par examples

- Reading current value
  - `old.par <- par()$mar`
- Setting a new value
  - `par(mar=c(4,11,2,1))` # Do plot
- Restoring old value after you are done
  - `par(mar=old.par)`

Par options

- Margins
  - `mai` (set margins in inches)
  - `mar` (set margins in number of lines)
  - `mex` (set lines per inch)
  - 4 element vector (bottom, left, top, right)
- Warning
  - `Error in plot.new() : figure margins too large`

```
par( mar=c(2, 10, 10, 1) )
barplot(x)
```
Par options

- Multi-panel
  ```r
  par( mfrow=c(rows,cols) )
  ```

Specifying colors

- Controlled names
  ```r
  col=c("red", "green")
  ```
- Color by number
  ```r
  col=c(1, 2, 3)
  ```
- Hexadecimal strings string
  ```r
  Of the form "#RRGGBB" where each of the pairs RR, GG, BB consists of two hexadecimal digits giving a value in the range 00 to FF:
  ```
  ```r
  #FF0000 (red)
  #0000FF (blue)
  ```

Built in color schemes

- Functions to generate colors
- Pass in the number of colors you want, e.g. to get 7 different colors:
  ```r
  rainbow(7)
  ```
  ```r
  heat.colors(7)
  ```
  ```r
  cm.colors(7)
  ```
  ```r
  terrain.colors(7)
  ```
  ```r
  topo.colors(7)
  ```
  ```r
  Etc.
  ```
Color Packages

- **Color Brewer**
  - Set of pre-defined, optimized palettes
  - `library(RColorBrewer)`
  - `brewer.pal(n_colours, palette)`

- **ColorRamps**
  - Create smooth palettes for ramped color
  - Generates a function to make actual color vectors
  - `colorRampPalette(c(“red”, “white”, “blue”))`
  - `colorRampPalette(c(“red”, “white”, “blue”))(5)`

Applying Color to Plots

- Vector of numbers or specified colors passed to the `col` parameter of a plot function

- Vector of factors used to divide the data
  - Colors will be taken from the set color palette
  - Can read or set using `palette()` function
    - `palette()`
    - `palette(brewer.pal(9,”Set1”))`

    ```r
    plot(1:5, col=1:5, pch=15, cex=2)
    ```

Dynamic use of color

- Coloring by density
  - Pass data and palette to `densCols()`
  - Vector of colors returned

- See Lab Supplement (online):
  - Plotting with color in R

  [https://www.rdocumentation.org/packages/grDevices/versions/3.4.3/topics/densCols](https://www.rdocumentation.org/packages/grDevices/versions/3.4.3/topics/densCols)
Make a lab report!

- Open your previous class05 RStudio project (and your saved R script)

- Can you source your class05.R file to re-generate all your plots without error?

- If so you can now generate a nice HTML report of your work to date...

Homework!

New DataCamp Assignments

- RStudio IDE (Pt 1)
- Intermediate R
  - Conditionals and Control Flow
  - Functions
  - Loops

Useful new website: https://www.data-to-viz.com/