# Data100 Sp22 Disc 4 Regex/Viz!

**Attendance**:

**https://tinyurl.com/disc4michelle**

# Announcements

**Due Dates**

-Homework 4 due Feb 17

-Lab 4 due Feb 15

-Weekly Check 4 due Feb 14

**Other**

-Review session Feb19-20! (Weekend, will be recorded)

-Midterm Feb. 24

# Regex

# Basic Regex

| operation | example | matches | does not match |
|---|---|---|---|
| or | `TEA\|BOBA` | `TEA`<br>`BOBA` | every other string |
| character class | `[A-Za-z]*` | `helloWorld`<br>`BoBaTEa` | `b0b4T`<br>`AB_AB` |
| beginning of line (carat) | `^word.*` | `word`<br>`wordle` | `sword` |
| end of line | `.*word$` | `word`<br>`sword` | `wordle` |
| character class negation (carat within []) | `[^a-z]+` | `PEPPERS3982`<br>`17211!↑å` | `porch`<br>`CLAmS` |

# Repeating Characters

| operation | example | matches | does not match |
|---|---|---|---|
| Zero or more | `BOB*A` | BOA<br>BOBBBBA | BOOA<br>BBBA |
| One or more | `BOB+A` | BOBA<br>BOBBBBA | BOA<br>BBBA |
| zero or one | `cats?` | cat<br>cats | any other string |
| repeated exactly {a} times | `c[ao]{2}t` | coat<br>caat | caoat<br>cat |
| repeated from a to b times: {a,b} | `c[ao]{1,2}t` | coat<br>cat | caoat<br>ct |
| Parenthesis (also used in catching groups) | `(CA)+T` | CACACAT<br>CAT | CAAT<br>T |

# Character Groups

1. **. (period):** any character apart from a '\n' (newline)
2. **\w:** a "word" character (letter, digit, or underscore) [a-zA-Z0-9_]
3. **\s:** matches a single whitespace character
4. **\d:** decimal digit [0-9]
5. **\:** escapes a special regex character.  e.g. \( means match a opening parenthesis **(**

# Regex in Python

1. re.match(pattern, string, ...): match if zero or more characters at the beginning of string match given pattern

2. re.search(pattern, string, ...): scan through string looking for the first location where the regular expression pattern produces a match

3. re.findall(pattern, string, ...): returns all non-overlapping matches of pattern in string as a list of strings

4. re.sub(pattern, repl, string, ...): return the string obtained by replacing the non-overlapping occurrences of pattern in string by the replacement repl

# Other features

1. **Greedy vs Lazy**
    a. **Greedy: match as many times as possible**
    b. **Lazy: match as few times as possible**
    c. **+? (lazy): a+? only ever matches a max of 1 a**
2. **Order of operations**

| | ERE Precedence (from high to low) | |
|---|---|---|
| 1 | Collation-related bracket symbols | [==] [::] [..] |
| 2 | Escaped characters | \<special character> |
| 3 | Bracket expression | [] |
| 4 | Grouping | () |
| 5 | Single-character-ERE duplication | * + ? {m,n} |
| 6 | Concatenation | |
| 7 | Anchoring | ^ $ |
| 8 | Alternation | \| |

# Quickfire Regex

Which of these regex strings:

Matches: jossh, jossssh

Doesn’t match: josh, joh, bjosssh

1. jos[s]+h
2. ^jos[s]+h
3. ^jos[s]*h
4. jos[s]*h

# Quickfire Regex

Which of these regex strings:

Matches: jossh, jossssh

Doesn’t match: josh, joh, bjosssh

1. jos[s]+h: matches b(josssh)
2. ^jos[s]+h
3. ^jos[s]*h: matches josh
4. jos[s]*h: matches b(josssh)

# Quickfire Regex

Which strings would this Regex match?

[\w]?[\d]{2,3}[\w]+[A-Z]

1. j05hl15A
2. j0shli5A
3. 05lisA
4. j055shhhlisa

# Quickfire Regex

Which strings would this Regex match?

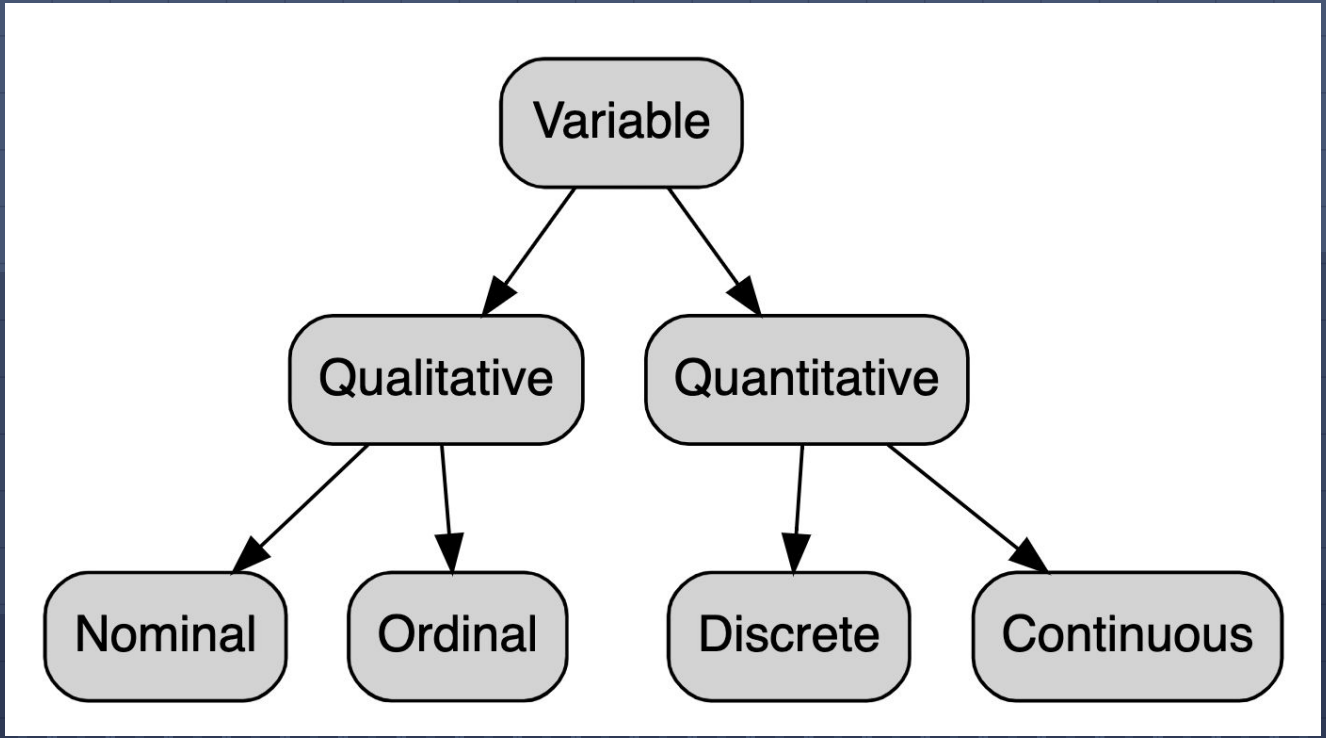
[\w]?[\d]{2,3}[\w]+[A-Z]

1. j05hl15A
2. j0shli5A : atleast 2 digits needed for [\d]{2,3}
3. 05lisA
4. J055shhhlisa: needs to end with an uppercase letter

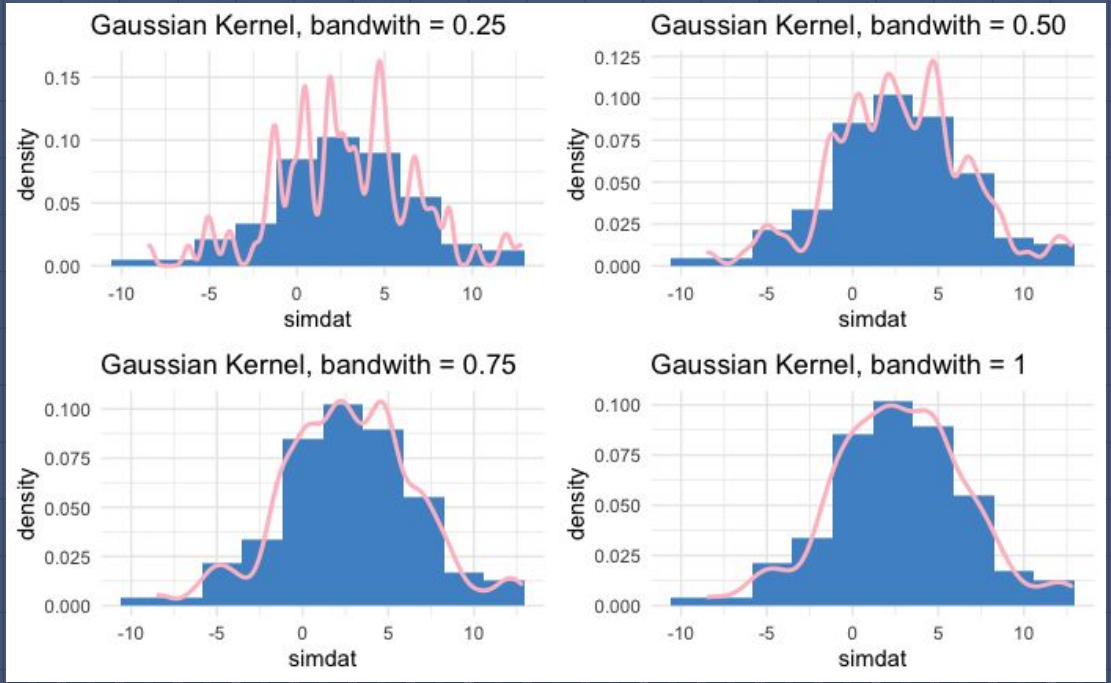
# Visualizations

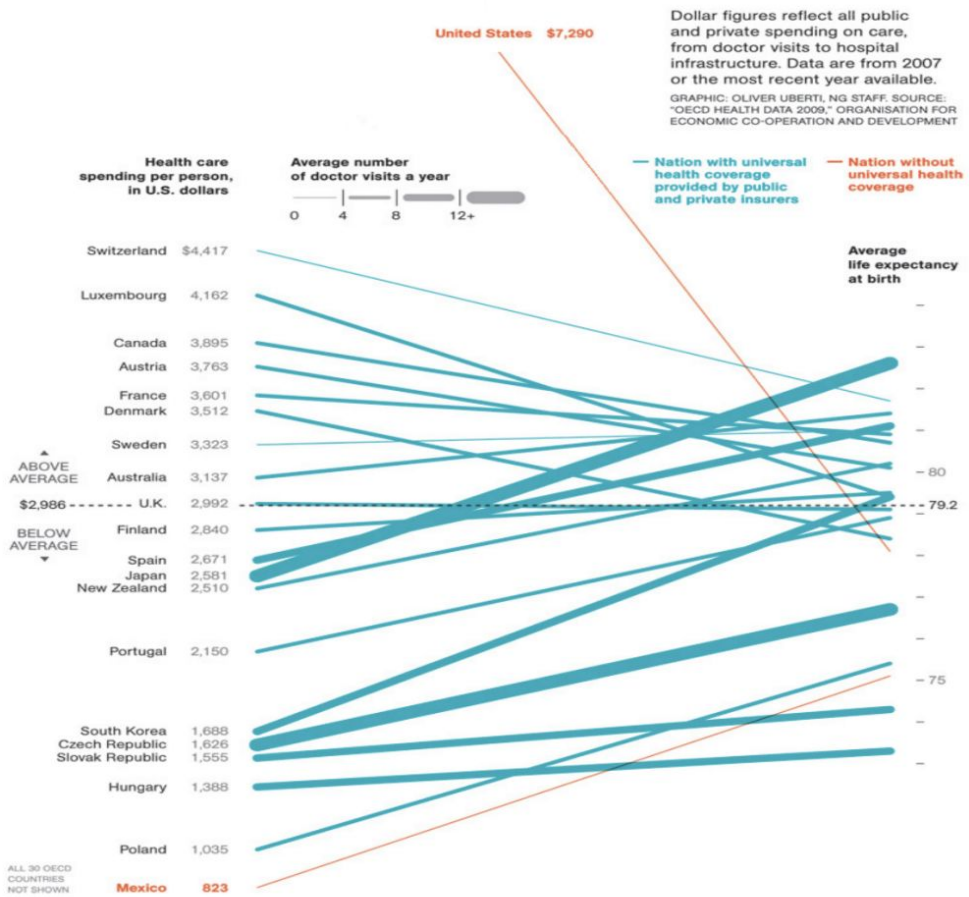
2

# Types of Variables



# KDEs


Gaussian Kernel, bandwith = 0.25
Gaussian Kernel, bandwith = 0.50
Gaussian Kernel, bandwith = 0.75
Gaussian Kernel, bandwith = 1
density
simdat
0.15
0.10
0.05
0.00
0.125
0.100
0.075
0.050
0.025
0.000
-10
-5
0
5
10


United States $7,290
Dollar figures reflect all public and private spending on care, from doctor visits to hospital infrastructure. Data are from 2007 or the most recent year available.
GRAPHIC: OLIVER UBERTI, NG STAFF. SOURCE: "OECD HEALTH DATA 2009," ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT
Health care spending per person, in U.S. dollars
Average number of doctor visits a year
0 4 8 12+
Nation with universal health coverage provided by public and private insurers
Nation without universal health coverage
Average life expectancy at birth
Switzerland $4,417
Luxembourg 4,162
Canada 3,895
Austria 3,763
France 3,601
Denmark 3,512
Sweden 3,323
ABOVE AVERAGE
Australia 3,137
$2,986 U.K. 2,992
BELOW AVERAGE
Finland 2,840
Spain 2,671
Japan 2,581
New Zealand 2,510
Portugal 2,150
South Korea 1,688
Czech Republic 1,626
Slovak Republic 1,555
Hungary 1,388
Poland 1,035
Mexico 823
ALL 30 OECD COUNTRIES NOT SHOWN
80
79.2
75

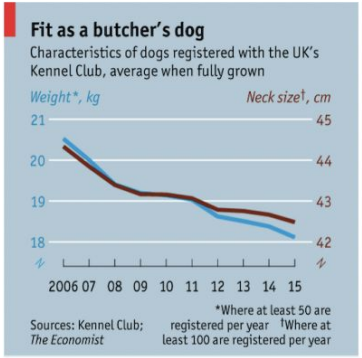
(a) Five variables are being represented visually in this graphic. What are they and what are their feature types (ie qualitative, quantitative, nominal, ordinal)?

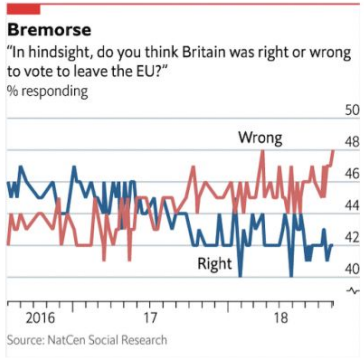
(c) How can we figure out how to interpret the visual qualities of the plot, e.g., how do we know what a color represents?

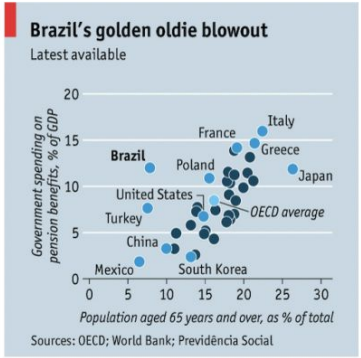
(e) Make 3 observations about the figure. Describe the feature that you are basing your observation on.

For example, South Korea's expenditure on health care is comparable to Eastern European countries (and among the lowest of all countries plotted), but the life expectancy is much higher than the Eastern European countries. In the plot we see that the left endpoint of South Korea's line segment is near the Eastern European countries, but the slope of the line segment is much steeper.

9. Creating visualizations that represent data accurately and that support the narrative we wish to create is no easy task. Even the journalists and editors at *The Economist*, a newspaper known for it's compelling, data-driven articles, have been known to make blunders. Three of their ill-thought-out plots are presented below. Consider what aspects of the visualizations are misleading, and think of ways in which you can remedy them.




*Hint:* The datapoints in the rightmost plot are shaded based on whether or not they are labeled.