CS 111: Program Design I

Lecture 16: Legal Analytics; Files concluded; More Lists

Robert H. Sloan & Richard Warner
University of Illinois at Chicago
Tuesday, October 22, 2019

UIC

LEGAL DATA ANALYTICS

The Prophecy Business

- "Our business is prophecy, and if prophecy were certain, there would not be much credit in prophesying."
  - Max Radin, 1925 (a legal theorist, UC Berkeley)
- *But*: there *is* certainty—and a lot of it.
  - The outcome in 90%+ of legal cases is highly predictable.
- *But*: the remaining unpredictable ones tend to matter—because they resolve open moral and political questions.

So Why So Much Litigation?

- A lot of litigation occurs over issues with a highly predictable outcome. Why?

a) Motivations other than winning—to delay, gain tax advantages (from losing), to seek revenge, and so on.

b) Misunderstandings of the law.

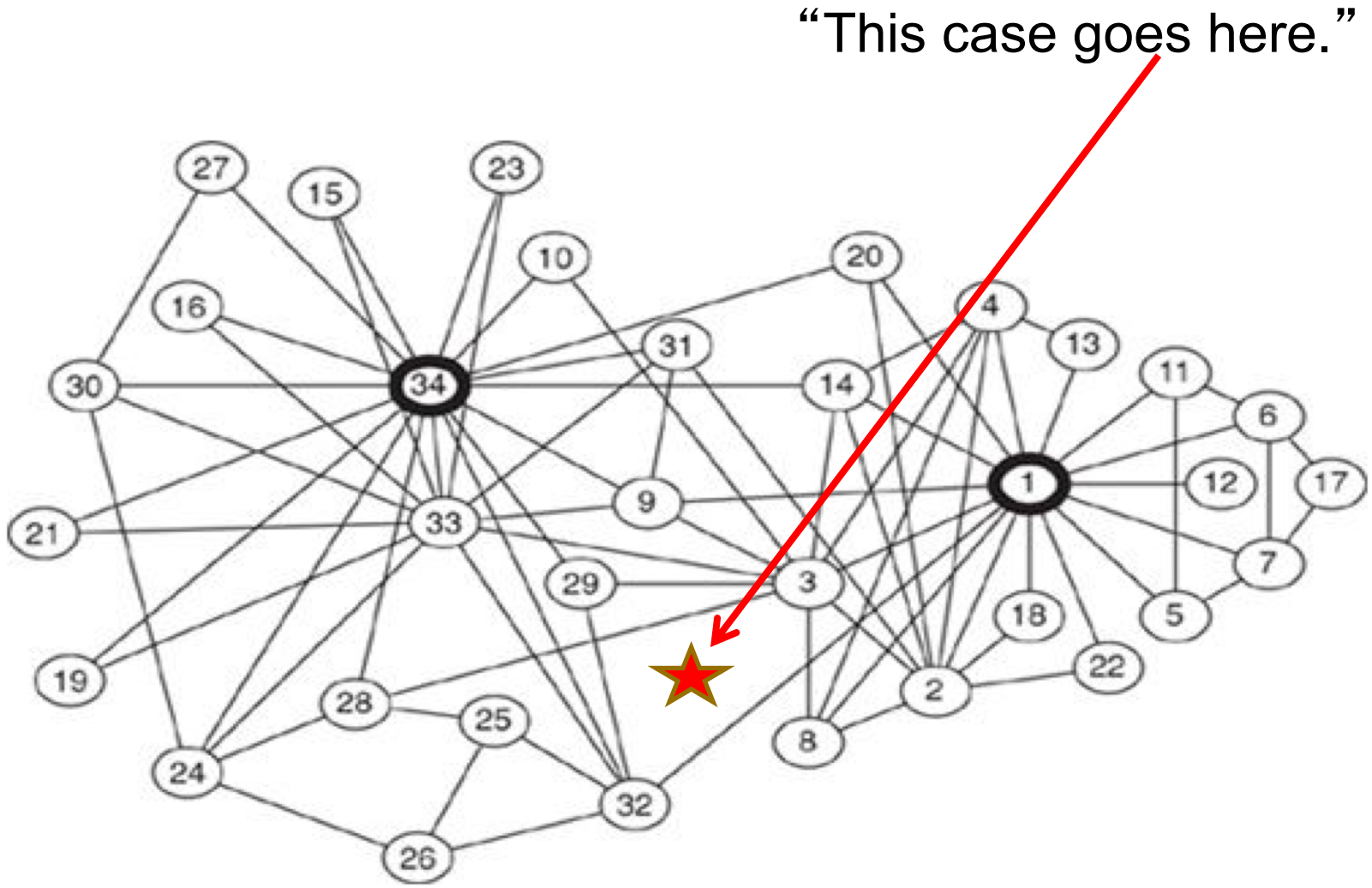
c) Disagreements over facts.

d) All of the above.

Traditional Prediction

- Identify one or more patterns.
- Predict the court will adhere to or depart from the pattern.
- How do you identify the patterns?
  - Training (law school) and experience.

Network and Prediction



Actually, a graph of friendships in a karate club, but makes the point.

Easy To Be Wrong About Patterns

- In 90% of the cases, traditional prediction works very well. But it is easy to be wrong about the patterns in the remaining 10%.
- An example: Most commentators and the 9th Circuit are currently wrong about the pattern of offer and acceptance in standard form contracting.
- So: “Our business is prophecy, and if prophecy were certain, there would not be much credit in prophesying.”

Why The Concern With Other Cases?

- Why lawyers and judges spend “much of the time . . . discussing how far the ruling should go and what its impact would be on other cases”?

a) That is what lawyers do.

b) To figure out where the case belongs in the space of relevantly similar cases.

c) To figure out where the case belongs in the space of relevantly similar cases, and to show that their decision is not arbitrary.

Data Analytics

- Dan Katz's model: "The team analyzed more than 60 years of Supreme Court data between 1953 and 2013, a total of 7,700 cases and more than 68,000 justice votes. The model predicts the behavior of 30 justices appointed by 13 presidents through six decades."

Why do this?

- Better predictions.
- Dan Katz's view:
  - "Many lawyers have years of expertise and knowledge that a computer simply cannot replicate . . . However, there are actually three ways to forecast something – experts, crowds and algorithms. A combination of these three methods is usually the most powerful and accurate."

More Reasons

- Discover patterns we did not know.
- Discover we are wrong about patterns we think we know.
- Remove the mystery.

LIMITATIONS OF LEGAL ANALYSIS WITH DATA ANALYTICS


Pay no attention to that
man behind the curtain!

Behind The Curtain

"Behind the curtain," you use groupby to rearrange the data--like this:

Ginsburg, 1

case

case

***

Ginsburg, 2

case

case

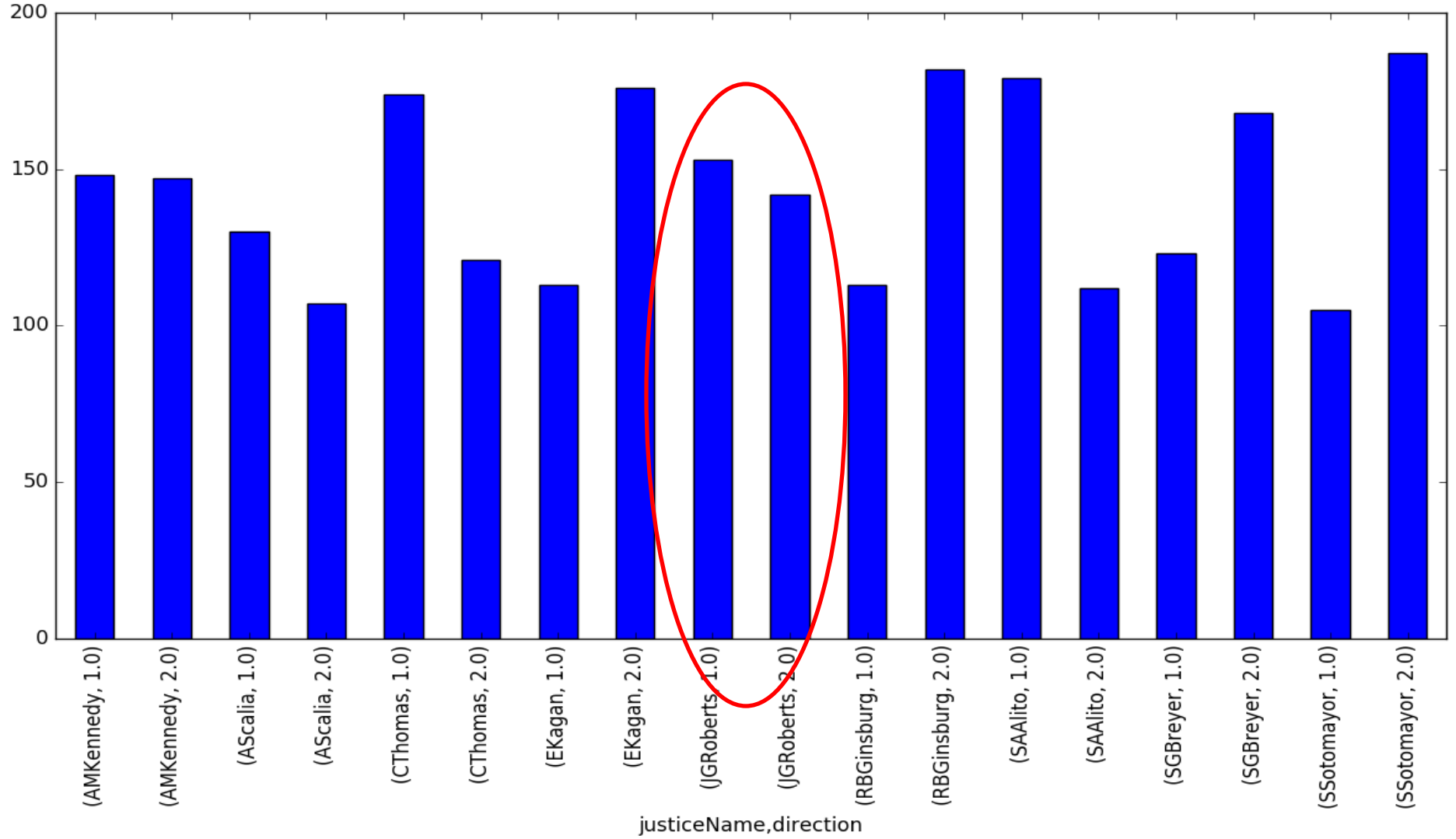
***

You can print this structure:

```
grouped = scdb.groupby(['justiceName', 'direction'])

for item in grouped:
    print (item[0])
    print (item[1])
```

Directions From 2012



Outcomes Or Arguments?

- “For each justice, calculate and plot the number of votes in each "direction" (total, not by term).”
- You are adding up

a) Outcomes of cases

b) Arguments for those outcomes.

What Do Outcomes Show?

- From 2005 – 2012, Roberts voted with the conservatives. Starting in 2012, he sided with the liberals in some cases.
- The explanation is that

a) Roberts has changed his views to be more liberal.

b) His views are what they have always been, but arguments in particular cases led him to side with the liberals.

c) Hard to tell if it is (a) or (b) or a combination.

Context Matters

- Suppose you know that Victoria recently read *Sense and Sensibility*, has been married to Victor for thirty years, voted for Obama in 2012, Clinton in 2016, and protested in Chicago against immigration policy.
- Can you reliably infer that she liked *Sense and Sensibility*, will remain married to Victor, approved of Obama's 2012 policies, and dislikes the current immigration policy?

a) Yes

b) No

Context Example

- Sally defaults on a $50,000 credit card debt.
  - incurred the debt to pay for lifesaving treatment for her eight-year old daughter
  - she has been paying what she can, she cannot afford the minimum payment.
  - When the credit card company begins collection procedures, she declares bankruptcy.
- Roger defaults on a $50,000 credit card debt.
  - incurred through compulsive gambling.
  - He declares bankruptcy.
- The credit reports of the bankruptcies provide no indication of the different contexts.

Cleaning, Organizing, Removing Context

	caseId	docketId	caselssuesId …
0	1946-001	1946-001-01	1946-001-01-01 . . .
1	1946-001	1946-001-01	1946-001-01-01 . . .
2	1946-001	1946-001-01	1946-001-01-01 . . .
3	1946-001	1946-001-01	1946-001-01-01 . . .
4	1946-001	1946-001-01	1946-001-01-01 . . .

Accurate Prediction

- Would expect accurate predictions from data without the relevant context?

a) Yes

b) No

Indeed, predictive data analysis (predictive analytics) can be poor at prediction.

When Is It Useful?

- When humans are even worse at prediction.
- There is significantly increased benefit from improved prediction accuracy.
- False positives and false negatives either decrease from whatever approach would otherwise be taken or are low in an absolute sense

FILES CONTINUED

Open arguments recapitulated

- Open
  - first argument: filename
  - Second argument, mode, 'r' read assumed if omitted
  - Optional encoding argument; irrelevant most of the time
    - But as you know, may need it if you are working with Spanish, Italian, Albanian, Tagalog, etc.
    - Or legal materials using section symbol §
    - (But probably not Arabic, Hebrew, Mandarin, Russian, etc. These would use UTF-8 encoding of Unicode, which is assumed)

```
f = open('SCDB_2019_01_justiceCentered_Citation.csv', 'r', encoding='ISO-
  8859-1')
```

(Text) File reading, a little more slowly

- Recall text file = sequence of lines
- Line = sequence of characters *up to and including* the special newline character \n
  - (Special case: probably last set of characters at end of file will work okay even if text file doesn't end with newline as it should.)
  - (How could we find out?)

Speaking of text

- afile.txt:

```
1234
Can I have a little more?
5678910
I love you!
ABCD
Can I bring my friend to
tea?
```

```
f = open("afile.txt", "r")
line = f.readline()
```

What is len(line)?

A. 0
B. 1
C. 4
D. 5
E. 6

Can iterate over text file reference (*not* in book)

```
fileref = open('afile.txt', 'r')

for line in fileref:    # process each line
    process line as we wish in this block
rest of program

fileref.close()
```

- Perhaps easiest way to read text file, all other things being equal

Reminder: Creating & *writing to* file

```
[In 1]: justices = 'Neil Gorsuch, Clarence Thomas,
Ruth Bader Ginsburg, Stephen G. Breyer, John G.
Roberts, Samuel A. Alito, Sonia Sotomayor, Elena
Kagan, Brett Kavanaugh'
[In 2]: file = open('justices.txt', 'w')
[In 3]: file.write('These are the justices of the
Supreme Court As of Oct 1 2019\n')
[In 4]: file.write(justices)
[In 5]: file.close()
```

Close the ~~door~~ file already

- after `f = open('some_file')` should eventually have statement

  `f.close()`
- If omitted, often Python system will ensure nothing bad happens but:
  - Bad style
  - Not guaranteed not to have wild bugs, e.g., files getting erased (rare in practice but can and does happen)
  - We will take off points if we spot it on labs or exams

Less likely to make mistake with with

```
with open('afile.txt', 'r') as fileref:
    for line in fileref:     # process each line
        process line as we wish in this block
rest of program

# No need to remember to close!
```

MODULES: REVIEW, 1 MORE EXAMPLE (RANDOM)

Modules

- import to make it available
- access it through the dot notation (like methods)
- We'll briefly look at one interesting module that is from the standard library: random

An interesting module: random

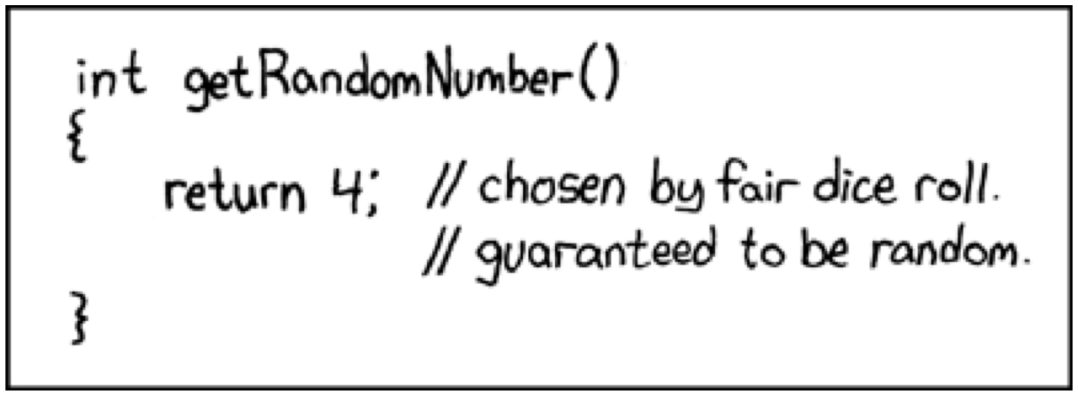
```
>>> import random
>>> for i in range(5):
...     print(random.random())
...
0.12636664029165268
0.2821272889535512
0.6160031940187543
0.2860900698190852 5
0.6277074518401735
```

- Notice: We're using *function* named random from *module* named random, hence random.random()

Randomly choosing words from a list

```
>>> for i in range(5):
...     print(random.choice(["Here", "is", "a", "list", "of", "words", "in", "random", "order"]))
...
list
words
in
Here
list
```

Exactly what does random mean?



How often do you look at xkcd

- A Usually or always
- B Once in a while
- C Never
- D I've never heard of xkcd

Randomly generating language

- Given a list of nouns, verbs that agree in tense and number, and object phrases that all match the verbs
- We can randomly take one from each to make sentences.

```
import random

def excuse():
  excuse = ["I didn't know I was in this class", "I thought
I already graduated", "I got stuck in a blizzard"]
  bigNum = ["4", "17", "like a billion", "mega", "tons of"]
  lottaWork = ["midterms", "Ph.D. theses", "programs"]
  print ("I need an extension because",
random.choice(excuse), "and I had", random.choice(bigNum),
random.choice(lottaWork), "to do.")
```

Side note: Good example of a function that *should* have 0 inputs and no return value.

Running random sentence generator

```
>>> excuse()
I need an extension because I thought I already graduated and I had like a billion programs to do.
>>> excuse()
I need an extension because I got stuck in a blizzard and I had 4 programs to do.
>>> excuse()
I need an extension because I got stuck in a blizzard and I had 17 programs to do.
>>> excuse()
I need an extension because I thought I already graduated and I had tons of programs to do.
>>> excuse()
I need an extension because I didn’t know I was in this class and I had 17 Ph.D. theses to do.
```