Prob Set 6: due Mon night 24 Apr

Prob Set 7: to be issued tonight, due mid next week

Prob Set 8: due 11 May (end of classes)
\[ \text{Cov}(X,Y) = E[(X-E[X])(Y-E[Y])] \]

\[ \text{Cov}(X,X) = \text{Var}(X) \]

\[ X>E[X], \ Y>E[Y] \]
\[ X<E[X], \ Y<E[Y] \]
\[ \text{Cov}(X,Y) > 0 \]

\[ X<E[X], \ Y>E[Y] \]
\[ X>E[X], \ Y<E[Y] \]
\[ \text{Cov}(X,Y) < 0 \]
Correlation coefficient only measures association. Correlation does not imply causation.

Though the correlation between the weight and the math ability of children in a school district may be positive, that does not mean that doing math makes children heavier or that putting on weight improves the children's math skills.

Age is a confounding variable: older children are both heavier and better at math than younger children, on average.
Figure 1. Correlation between Countries’ Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

see also http://www.tylervigen.com/spurious-correlations

https://www.inferentialthinking.com/chapters/13/1/correlation.html
Recall that if we rescale the data, \( x \rightarrow x/c \), then that divides the standard deviation \( \sigma[x] \) by the same \( c \).

In particular, we can divide by \( c= \sigma[x] \), which gives a new distribution with standard deviation normalized to 1 (as is done to calculate z values).

Note that \( \text{Pearson}(x,y) = \frac{\text{Cov}[x,y]}{\sigma[x] \sigma[y]} \) is unchanged by rescaling both \( x \) and \( y \).

On the other hand, if we rescale \( x\rightarrow x/\sigma[x] \) and \( y\rightarrow y/\sigma[y] \), then the linear regression slope \( \frac{\text{Cov}[x,y]}{\text{Var}[x]} \rightarrow \text{Cov}[x,y] \).

Therefore the \( \text{Pearson}(x,y) \) is just the linear regression slope when the variables are rescaled to have standard deviation equal to 1.
[20,20,15,10,25]
1) naive bayes, 2) simple probab, 3) wikipedia, 4) python, 5) research lit
Dependences are not always linear

(Except in … )
(e) The Division of Coastal Management shall be the only State agency authorized to
develop rates of sea-level rise and shall do so only at the request of the Commission. These
rates shall only be determined using historical data, and these data shall be limited to the time
period following the year 1900. Rates of sea-level rise may be extrapolated linearly to estimate
future rates of rise but shall not include scenarios of accelerated rates of sea-level rise. Rates of
sea-level rise shall not be one rate for the entire coast but, rather, the Division shall consider
separately oceanfront and estuarine shorelines. For oceanfront shorelines, the Division shall use
no fewer than the four regions defined in the April 2011 report entitled "North Carolina Beach
Spearman correlation=1
Pearson correlation=0.88

https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient
The Pearson correlation coefficient misses non-linear relationships and is also sensitive to outliers — the Spearman correlation can sometimes find correlations that Pearson misses.

It is defined as the Pearson correlation of the rank order of the data. That means it also varies from $-1$ (perfectly anti-correlated) to $+1$ (perfectly correlated), with $0$ meaning uncorrelated.

If the data has $x = [.6, .4, .2, .1, .5]$ then the ranks are $r = [5, 3, 2, 1, 4]$.

For data $y = [403, 54, 7, 2, 148]$, the ranks $s = [5, 3, 2, 1, 4]$ are the same.*

so the Spearman correlation is $1$, whereas the Pearson is less than one. Both functions are available in scipy.stats (as pearsonr() and spearmanr()).

[*Actually the second was generated from the first by taking the integer part of $\exp(10x)$*]
Defined as the Pearson correlation for the ranks, the Spearman correlation is written

\[ \rho = \frac{\text{Cov}[r, s]}{\sigma[r] \sigma[s]} , \]  

where \( \text{Cov}[r, s] = E[(r - E[r])(s - E[s])] \)

(generalizing the \( \text{Var}[x] = E[(x - E[x])^2] \), with \( \text{Cov}[x, x] = \text{Var}[x] \)).

The formula for the Spearman correlation coefficient is given at http://en.wikipedia.org/wiki/Spearman’s_rank_correlation_coefficient in terms of the difference \( d_i = r_i - s_i \) between ranks, in this easily calculable form:

\[ \rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} . \]  

(2)

It is straightforward to verify that (1) reduces to (2) (see linked notes)
roughly elliptically distributed and there are no prominent outliers: same

less sensitive than the Pearson correlation to strong outliers that are in the tails of both samples (limited to value of rank)

https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient
The Democrats Show Some Spunk

Two days of obstruction on Capitol Hill.

By Jim Newell

We are on Day 12 of the Trump administration, and Senate Democrats have begun slowing the pace of Capitol Hill down to a crawl.

It started Monday night with a surprise procedural move, just as Congress was returning to session. The Senate Finance Committee was scheduled to meet at 6 p.m. to vote on a GOP budget resolution that included a nearly $1 trillion cut to Medicare, Medicaid, and other entitlement programs. The committee had agreed to the plan, but the Senate was not scheduled to vote. Democrats refused to vote on the plan, and the meeting was delayed until 9 p.m. When Republican Majority Leader Mitch McConnell finally called the vote, 53 of 54 Senate Republicans were present, but they did not have enough votes to approve the plan. After almost an hour of discussion, McConnell announced that the Senate would not vote on the resolution. Democrats hailed the move as a victory for the party, as the procedure was used to avoid a vote on a policy they opposed. Republicans accused the Democrats of delaying the budget process, but Democrats argued that the budget resolution was a bad idea and should not have been voted on. The move was seen as a sign of the Democratic Party's growing frustration with the Trump administration's policies.
allsoc = []
for url in urls:
    driver.get(url)
    social = driver.find_element_by_class_name('header_social')
    soc = []  # temporary list for this url
    for cl in ('facebook', 'twitter', 'comments'):
        elt = social.find_element_by_class_name(cl)
        subelt = elt.find_element_by_class_name('count')
        soc.append(subelt.get_attribute('innerHTML'))
    allsoc.append(soc)