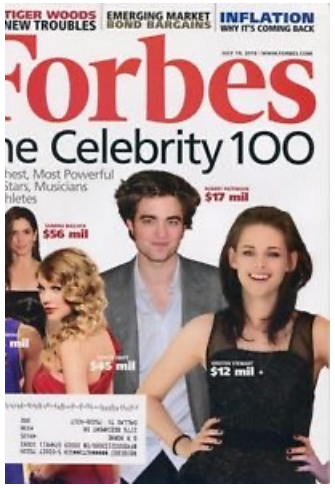
Forbes List vs Wikipedia Pageviews

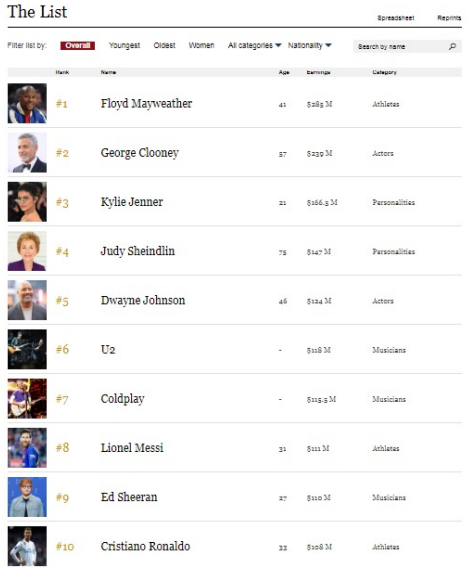
An exploration in correlation

The Forbes Celebrity 100

- Forbes compiles an annual list of the world's highest paid celebrities.
- The list has been published since 1999.



The World's Highest-Paid Celebrities



The Best of the Best

To be cited in the Forbes list is already a great honor for any celebrity. But even in this elite group, there are over-achievers.

From the annual list of 100, let's look at only the top 10.

And from the annual top 10, let's see who's been included the most.

Celebrities most often included in the Forbes Top 10

```
StartYear = 1999
dfForbesCount = dfForbes[dfForbes["year"] >= StartYear].groupby("recipient")["recipie
dfForbesCount.columns = ["ForbesCount"]
dfForbesCount = dfForbesCount.reset_index()
dfForbesCount = dfForbesCount.rename(columns={'recipient':'Celebrity'})
dfForbesCount = dfForbesCount.sort_values(by = "ForbesCount", ascending = False)
dfForbesCount = dfForbesCount.reset_index(drop = True)
dfForbesCount.head(10)
```

	Celebrity	ForbesCount
0	Oprah Winfrey	16
1	Tiger Woods	12
2	Steven Spielberg	11
3	Tom Cruise	6
4	Madonna	6
5	Beyonce	6
6	The Rolling Stones	6
7	U2	5
8	Johnny Depp	5
9	Michael Jordan	5

Wikipedia Pageviews

Wikipedia was launched in 2001 and quickly became the default free encyclopedia.

Anyone can access and contribute freely.

Since 2016, Wikipedia has made it's page view statistics available.

Forbes List as a Predictor of Wikipedia Page Views?

Throughout the Forbes List history, there have been 82 celebrities (or groups of celebrities) that made it to the top 10.

One convenient trait they share is that they ALL have dedicated Wikipedia pages.

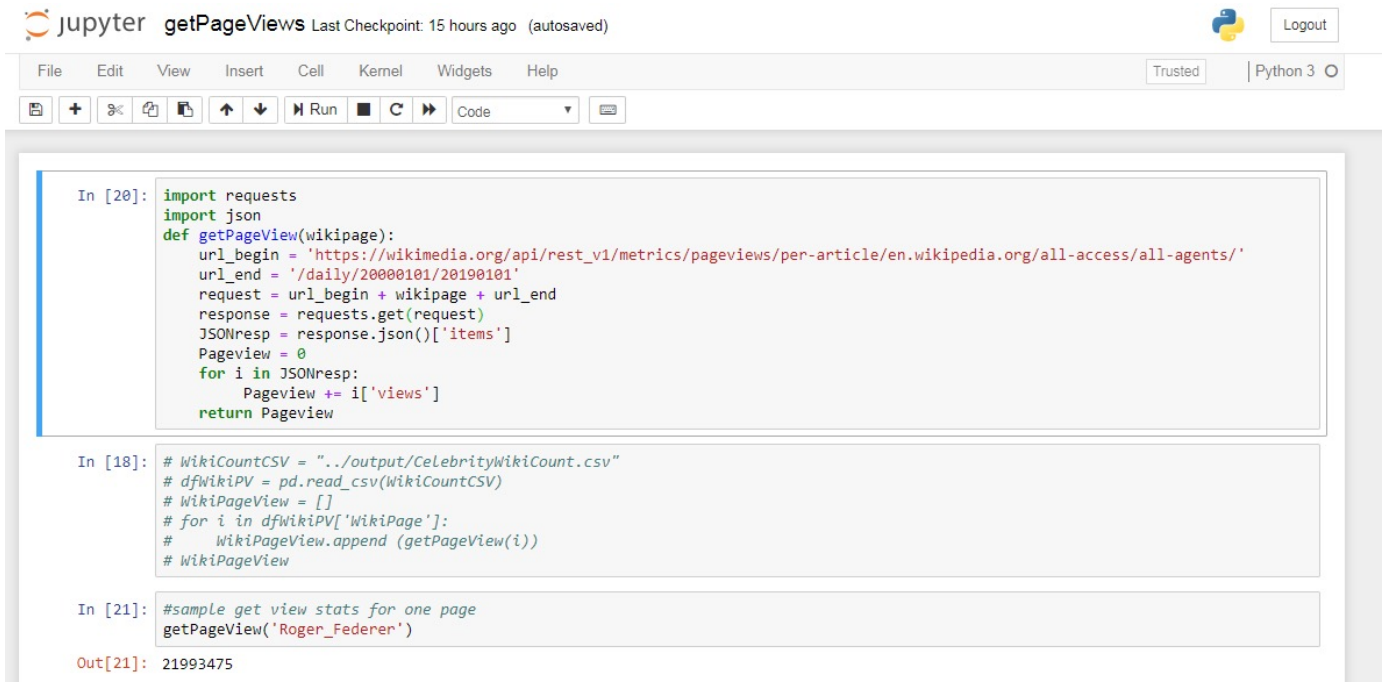
While the Forbes list is a good indicator of a celebrity's financial standing, it is interesting to ask if the Forbes List predicts their Wikipedia page views.

We can state this hypothesis more simply:

Is it true that the higher the celebrity's "Forbes Count", the higher his or her page view in Wikipedia?

Getting the Page View Counts

We use a Jupyter notebook running on Python to harvest Wikipedia page statistics.



Ranking the Same List of 82 Celebrities

Celebrities included in the Forbes Top and Bottom 10, ranked by number of times included

In [25]:
```
StartYear = 1999
dfForbesCount = dfForbes[dfForbes["year"] >= StartYear].groupby("recipient")["recipient"].agg([len])
dfForbesCount.columns = ["ForbesCount"]
dfForbesCount = dfForbesCount.reset_index()
dfForbesCount = dfForbesCount.rename(columns={'recipient':'Celebrity'})
dfForbesCount = dfForbesCount.sort_values(by = "ForbesCount", ascending = False)
dfForbesCount = dfForbesCount.reset_index(drop = True)
print (dfForbesCount.head(10))
print ('-----------------------------------')
print ('-----------------------------------')
print (dfForbesCount.tail(10))
```

```
            Celebrity  ForbesCount
0       Oprah Winfrey           16
1         Tiger Woods           12
2    Steven Spielberg           11
3          Tom Cruise            6
4             Madonna            6
5             Beyonce            6
6  The Rolling Stones            6
7                  U2            5
8         Johnny Depp            5
9      Michael Jordan            5
-----------------------------------
-----------------------------------
            Celebrity  ForbesCount
72      James Cameron            1
73     Jerry Seinfeld            1
74     Judy Sheindlin            1
75              Adele            1
76         Kevin Hart            1
77     Kim Kardashian            1
78        Kobe Bryant            1
79       Kylie Jenner            1
80  Leonardo DiCaprio            1
81      Julia Roberts            1
```

Celebrities included in the Forbes Top and Bottom 10, ranked by Wikipedia Page Views

In [30]:
```
dfWikiPV = dfWikiPV.sort_values(by = "PageView", ascending = False)
dfWikiPV = dfWikiPV.reset_index(drop = True)
#dfWikiPV
print (dfWikiPV.head(10))
print ('-----------------------------------------------------------------------------')
print ('-----------------------------------------------------------------------------')
print (dfWikiPV.tail(10))
```

```
           Celebrity           WikiPage  WikiEditCount  PageView
0  Cristiano Ronaldo  Cristiano_Ronaldo          16652  44265738
1       Lionel Messi       Lionel_Messi          16272  38824431
2     Dwayne Johnson     Dwayne_Johnson          16097  37396145
3       LeBron James       LeBron_James          14642  29929582
4             Eminem             Eminem          19899  28742857
5  Leonardo DiCaprio  Leonardo_DiCaprio           7654  27873482
6     Kim Kardashian     Kim_Kardashian           5945  26101487
7     Michael Jordan     Michael_Jordan          12040  25275868
8         Tom Cruise         Tom_Cruise           7769  25135433
9      Justin Bieber      Justin_Bieber           8593  24775359
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
          Celebrity         WikiPage  WikiEditCount  PageView
72         Bon Jovi         Bon_Jovi           8163   4962838
73     Garth Brooks     Garth_Brooks           4160   4911917
74  Dr. Phil McGraw      Phil_McGraw           2638   4237066
75            NSYNC            NSYNC           6130   3485107
76        Dan Brown        Dan_Brown           3382   3139622
77       The Police       The_Police           4429   2535558
78    Rush Limbaugh    Rush_Limbaugh           8685   2319879
79    Judy Sheindlin    Judy_Sheindlin           2762   2312679
80  James Patterson  James_Patterson           2543   2078324
81       Paula Deen       Paula_Deen           2563   1546522
```

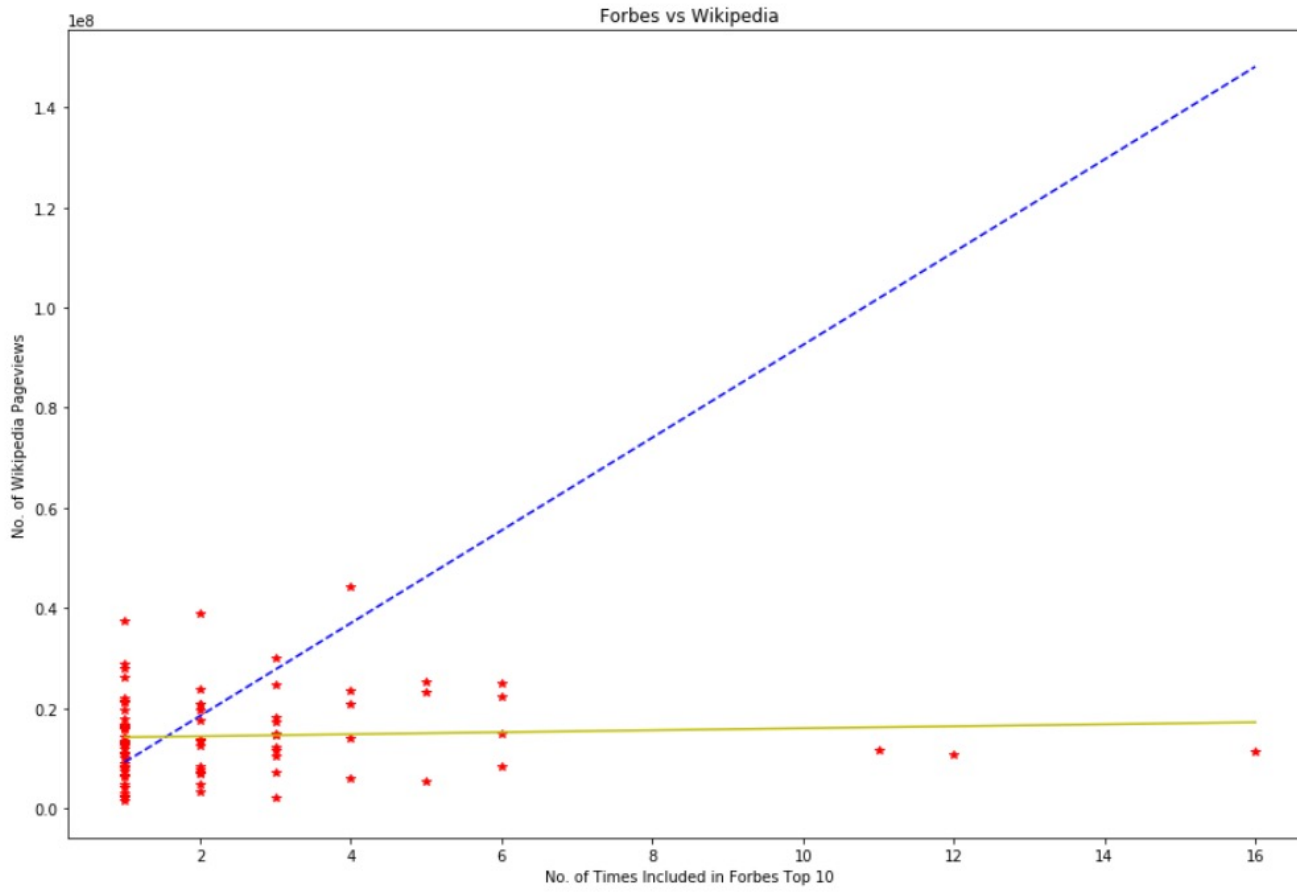
Making a Scatter Plot with Linear Regression

At the extremes, the two lists are ordered differently. This already gives an instinctive feeling of no correlation.

Exactly how false our hypothesis is, is something we can determine through analysis.

We create a scatter plot of all 82 celebrities (red), and determine the best straight line with minimal distance from each point (in yellow). This represents our actual correlation.

For contrast, ideal positive correlation is shown in blue.



R-value, P-value, Correlation, Catch

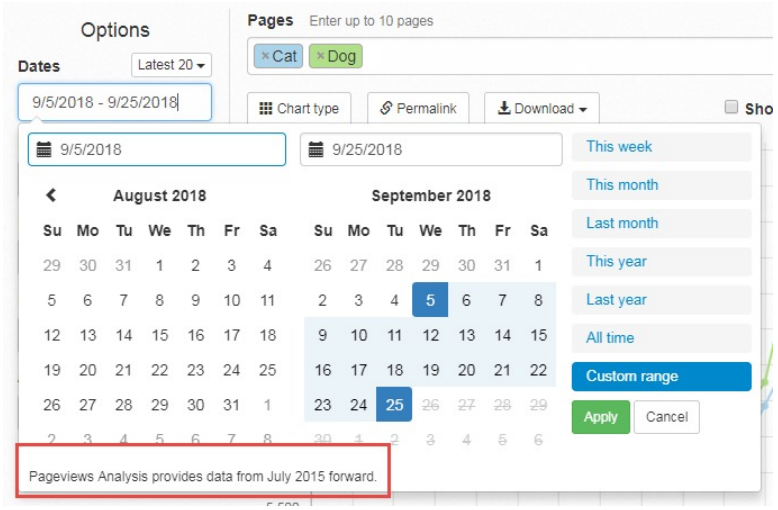
```
r_value: 0.0585844451932 p_value: 0.60110659818 std_err: 377818.895678
```

- R-value (-1 to +1) measures how close the two variables (Forbes count and Page view) follow each other. An r-value close to zero indicates no correlation.
- P-value (0 to 1) indicates how likely our data was a product of randomness. The higher the p-value, the more likely it is by pure chance.

Both our computed r and p values suggest that our hypothesis is false.

In other words, the number of times a celebrity has been included in the Forbes Top 10 list does not predict his or her Wikipedia page views.

However, there is a catch. An inherent bias in our data may be affecting our results. The next slides explore what's wrong.



Data Bias

The Pageviews API from Wikipedia only provides data from July 2015 forward.

This creates a bias in favor of more recent celebrities because the pageviews from 1999 to 2014 are not counted.

Determining Trend of R-value Over Years

If our data and results are not valid for year range (1999 to present), are there any year ranges when they start becoming valid?

We create a function that iterates over the data, increasing the "StartYear" with each step.

Create table of statistical results for all inclusive years

In [9]:
```
def getGraphYearly():
    Year = []
    R_value = []
    P_value = []
    Std_error = []
    global dfYearly
    for StartYear in range(1999,2018):
        getArr = getyearlystat(StartYear)
        #print (getArr)
        Year.append(getArr[1])
        R_value.append(getArr[3])
        P_value.append(getArr[5])
        Std_error.append(getArr[7])
    dfYearly =  pd.DataFrame({
     'Year': Year,
     'R_value': R_value,
     'P_value': P_value,
     'Std_error': Std_error
    })
    dfYearly
```

In [10]:
```
getGraphYearly()
dfYearly
```

Out[10]:

	P_value	R_value	Std_error	Year
0	0.601107	0.058584	3.778189e+05	1999
1	0.583370	0.063908	4.090965e+05	2000
2	0.511049	0.077604	4.458255e+05	2001
3	0.547700	0.072544	4.848378e+05	2002
4	0.548515	0.073474	5.331233e+05	2003
5	0.427619	0.100088	5.865397e+05	2004
6	0.289193	0.134535	6.441705e+05	2005
7	0.293220	0.136773	7.465471e+05	2006
8	0.242880	0.155795	8.451343e+05	2007
9	0.206719	0.169785	9.470498e+05	2008
10	0.286149	0.147806	1.044345e+06	2009
11	0.160536	0.203616	1.242632e+06	2010
12	0.092623	0.250877	1.437247e+06	2011
13	0.120871	0.240145	1.762115e+06	2012
14	0.086788	0.277845	1.935074e+06	2013
15	0.085961	0.303515	2.273205e+06	2014
16	0.184312	0.258387	2.706294e+06	2015
17	0.134428	0.314521	4.521360e+06	2016
18	0.316306	0.250377	9.535031e+06	2017

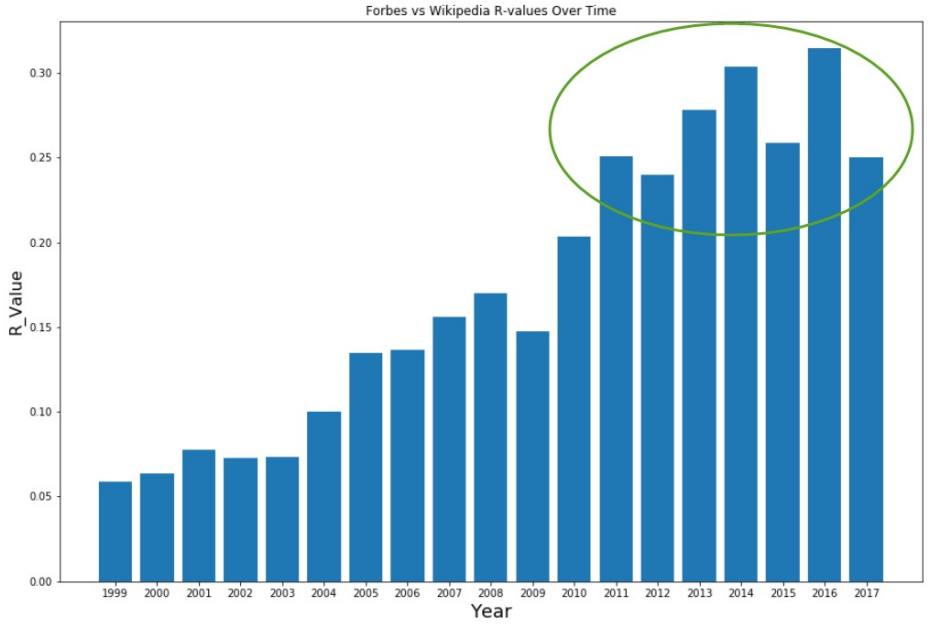
Increasing Correlation Over Time

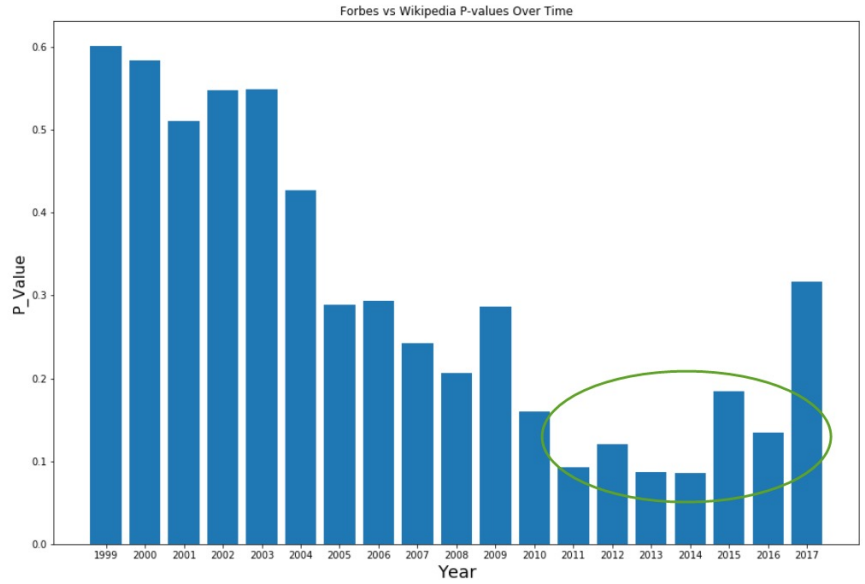
The results seem positive. We see an increasing correlation of Forbes and Wikipedia data over time.

More significantly, from "no correlation", our graph achieves "weak correlation" on year 2010.

We also see a significant decrease in P-value around the same year.

Correcting for data, our results now suggest weak correlation.





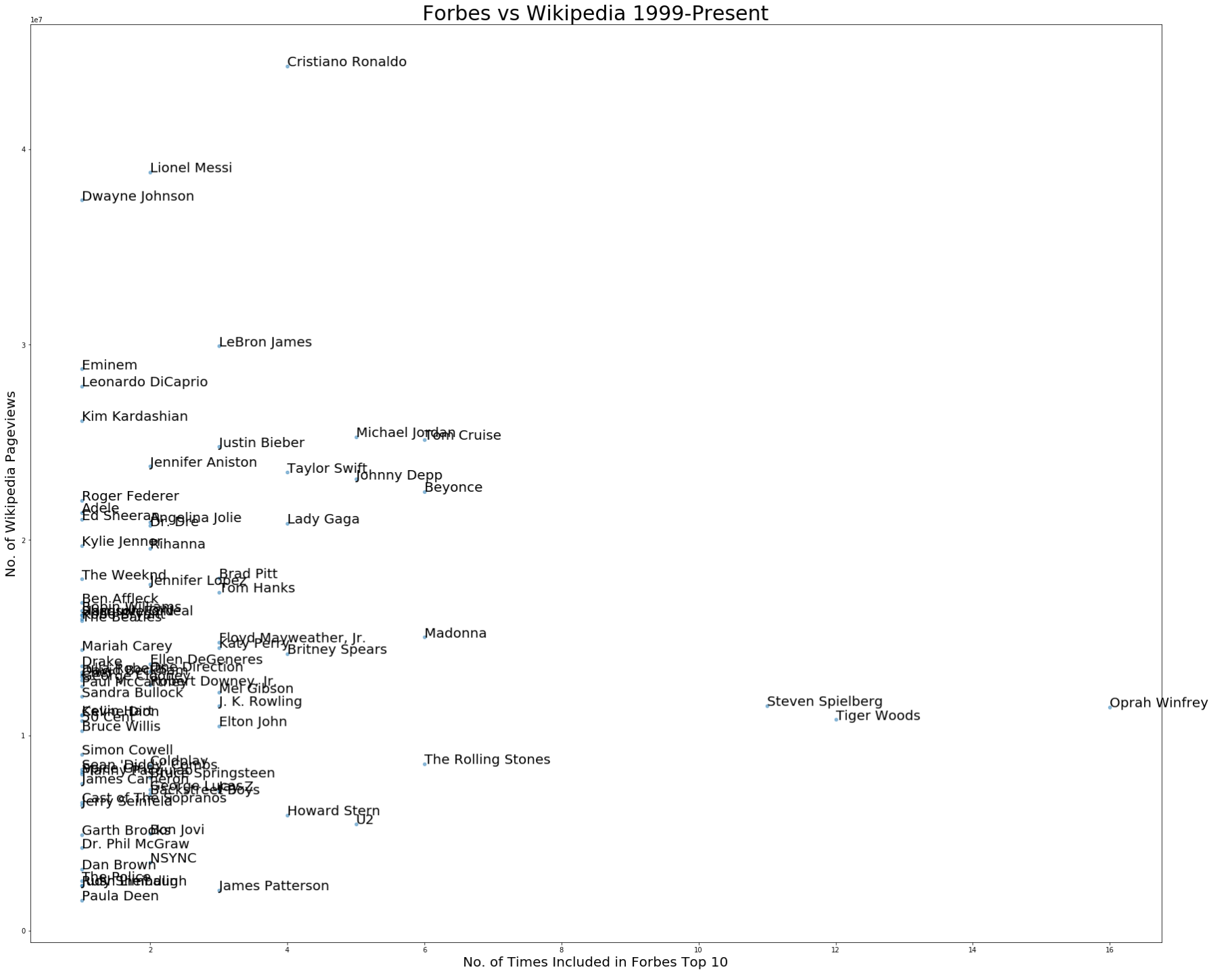
Acknowledgements

- This small presentation was created with advise and inspiration from DIY Data Workshop facilitated by Dr. Reina Reyes, 2018, Shaw Boulevard, Mandaluyong, Metro Manila.
- Forbes List data prepared by Reina Reyes.
- Pageview data prepared by Jesus Chua Jr.

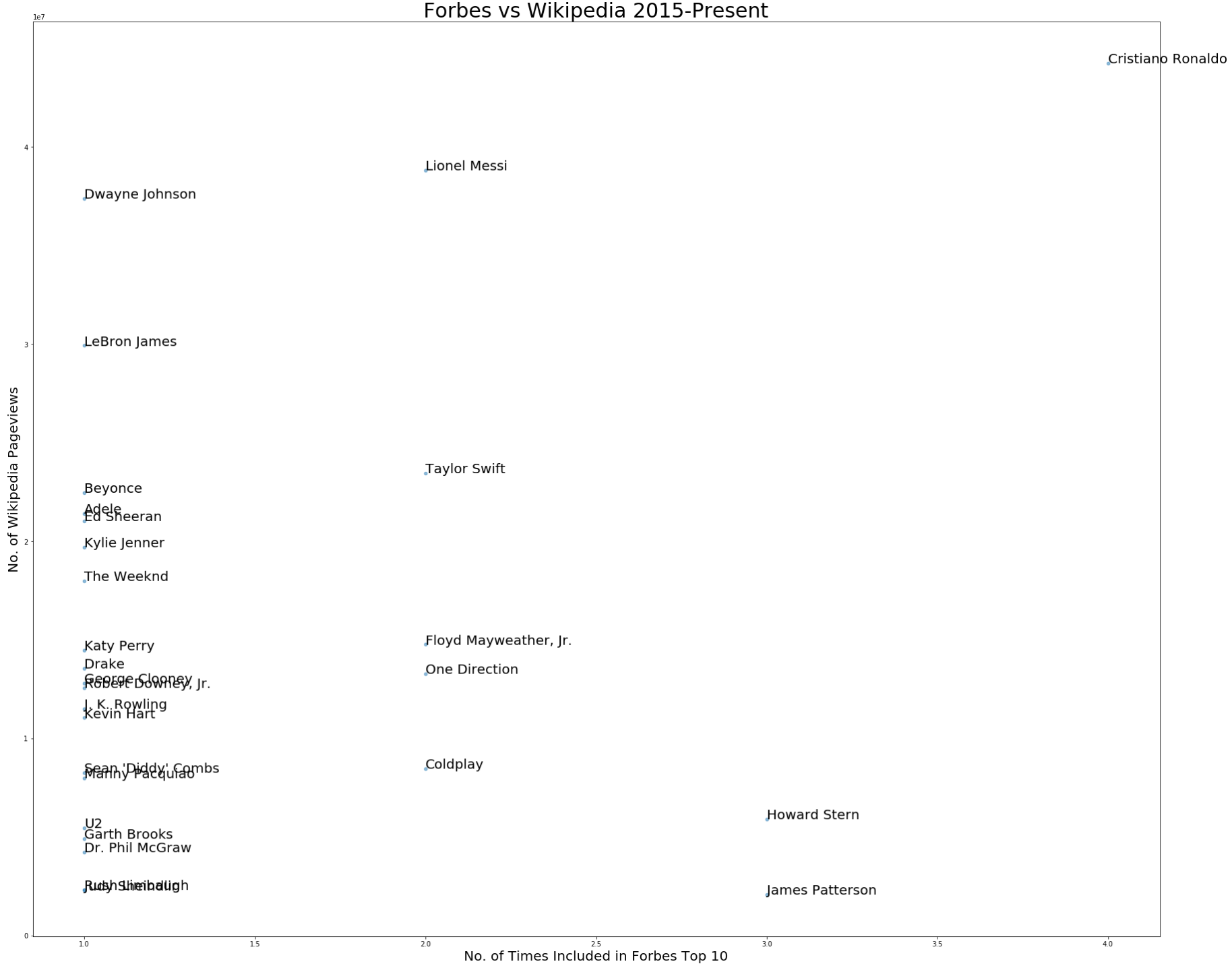
- Subjective rule-of-thumb used in categorizing R-value correlation:
  - 0.00-0.19: very weak
  - 0.20-0.39: weak
  - 0.40-0.59: moderate
  - 0.60-0.79: strong
  - 0.80-1.00: very strong

  Source: https://www.researchgate.net/post/What_is_the_minimum_value_of_correlation_coefficient_to_prove_the_existence_of_the_accepted_relationship_between_scores_of_two_of_more_tests
- There is a noticeable decrease in R-value in year 2017, and a corresponding spike in P-value. This is most likely caused by the sample size being too small. At year range 2017-present, there are only 18 celebrities at play.

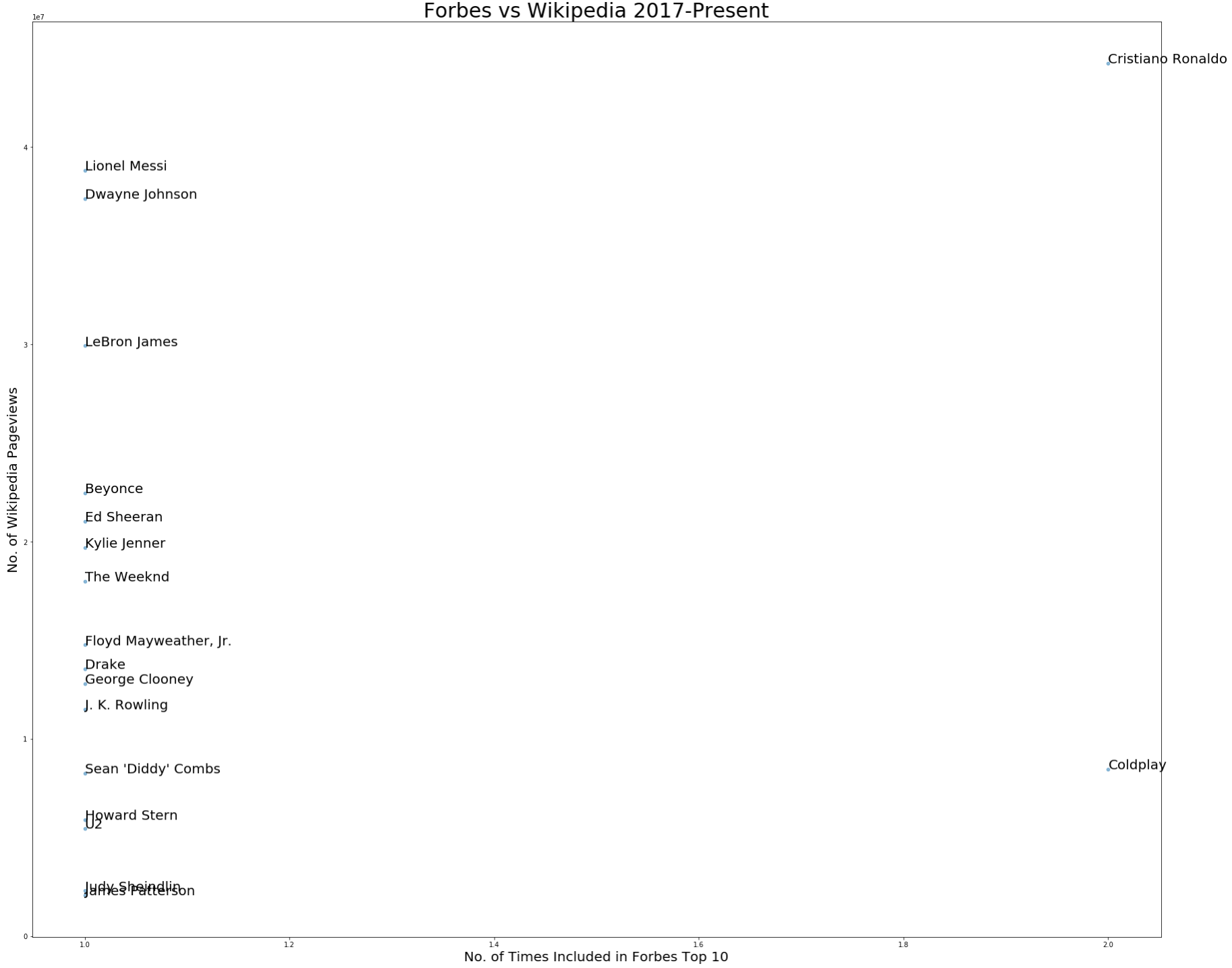
Original Scatter Plot with Names



Data-Corrected Scatter Plot



Scatter Plot with too small sample size



Thank you for your time