Importing all the Libraries

In [1]:
```
%matplotlib inline
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
import json
from pandas.io.json import json_normalize
from wordcloud import WordCloud, STOPWORDS
```

In [2]:
```
month_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Se
p',
               'Oct', 'Nov', 'Dec']
day_order = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
```

In [3]:
```
df = pd.read_csv('../TED_TALKS/Data/ted_main.csv')
df.columns
```

Out[3]:
```
Index(['comments', 'description', 'duration', 'event', 'film_date',
       'languages', 'main_speaker', 'name', 'num_speaker', 'published_date
',
       'ratings', 'related_talks', 'speaker_occupation', 'tags', 'title',
       'url', 'views'],
      dtype='object')
```

Features Available

- name: The official name of the TED Talk. Includes the title and the speaker
- title: The title of the talk
- description: A blurb of what the talk is about
- main_speaker: The first named speaker of the talk
- speaker_occupation: The occupation of the main speaker
- num_speaker: The number of speakers in the talk
- duration: The duration of the talk in seconds
- event: The TED/TEDx event where the talk took place
- film_date: The Unix timestamp of the filming
- published_date: The Unix timestamp of the publication of the talk on TED.com
- comments: The number of first level comments made on the talk
- tags: The themes associated with the talk
- languages: The number of languages in which the talk is available
- ratings: A stringified dictionary of the various ratings given to the talk(For ex: inspiring, fascinating, jaw dropping etc.)
- related_talks: A list of dictionaries of recommended talks to watch next
- url: The URL of the talk
- views: Number of views on the talk

In [4]:
```
df = df[['name', 'title', 'description', 'main_speaker', 'speaker_occupation',
       'num_speaker', 'duration', 'event', 'film_date', 'published_date',
'comments',
       'tags', 'languages', 'ratings', 'related_talks', 'url', 'views']]
```

In [5]:
```
import datetime
df['film_date'] = df['film_date'].apply(lambda x: datetime.datetime.
                                        fromtimestamp(int(x)).strftime('%d-%
m-%Y'))
df['published_date'] = df['published_date'].apply(lambda x: datetime.datet
ime.
                                                  fromtimestamp(int(x)).st
rftime('%d-%m-%Y'))
```

In [6]:
```
df.head()
```

Out[6]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	durat
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	1164
1	Al Gore: Averting the climate crisis	Averting the climate crisis	With the same humor and humanity he exuded in ...	Al Gore	Climate advocate	1	977
2	David Pogue: Simplicity sells	Simplicity sells	New York Times columnist David Pogue takes aim...	David Pogue	Technology columnist	1	1286
3	Majora Carter: Greening the ghetto	Greening the ghetto	In an emotionally charged talk, MacArthur-winn...	Majora Carter	Activist for environmental justice	1	1116
4	Hans Rosling: The best stats	The best stats you've	You've never seen data presented	Hans Rosling	Global health expert; data visionary	1	1190

	you've ever seen	ever seen	like this. Wi...				

In [7]:
```
len(df)
```

Out[7]: 2550

In [8]:
```
pop_talks = df[['title', 'main_speaker', 'views', 'comments',
                'film_date']].sort_values('views', ascending=False)[:15]
pop_talks
```

Out[8]:

	title	main_speaker	views	comments	film_date
0	Do schools kill creativity?	Ken Robinson	47227110	4553	25-02-2006
1346	Your body language may shape who you are	Amy Cuddy	43155405	2290	26-06-2012
677	How great leaders inspire action	Simon Sinek	34309432	1930	17-09-2009
837	The power of vulnerability	Brené Brown	31168150	1927	06-06-2010
452	10 things you didn't know about orgasm	Mary Roach	22270883	354	06-02-2009
1776	How to speak so that people want to listen	Julian Treasure	21594632	297	10-06-2013
201	My stroke of insight	Jill Bolte Taylor	21190883	2877	27-02-2008
5	Why we do what we do	Tony Robbins	20685401	672	02-02-2006
2114	This is what happens when you reply to spam email	James Veitch	20475972	150	08-12-2015
1416	Looks aren't everything. Believe me, I'm a model.	Cameron Russell	19787465	846	27-10-2012
500	The puzzle of motivation	Dan Pink	18830983	1094	24-07-2009
1163	The power of introverts	Susan Cain	17629275	1155	28-02-2012
1036	How to spot a liar	Pamela Meyer	16861578	561	13-07-2011
2109	What makes a good life? Lessons from the longe...	Robert Waldinger	16601927	527	14-11-2015
					11-05-

1129	The happy secret to better work	Shawn Achor	16209727	754	2011

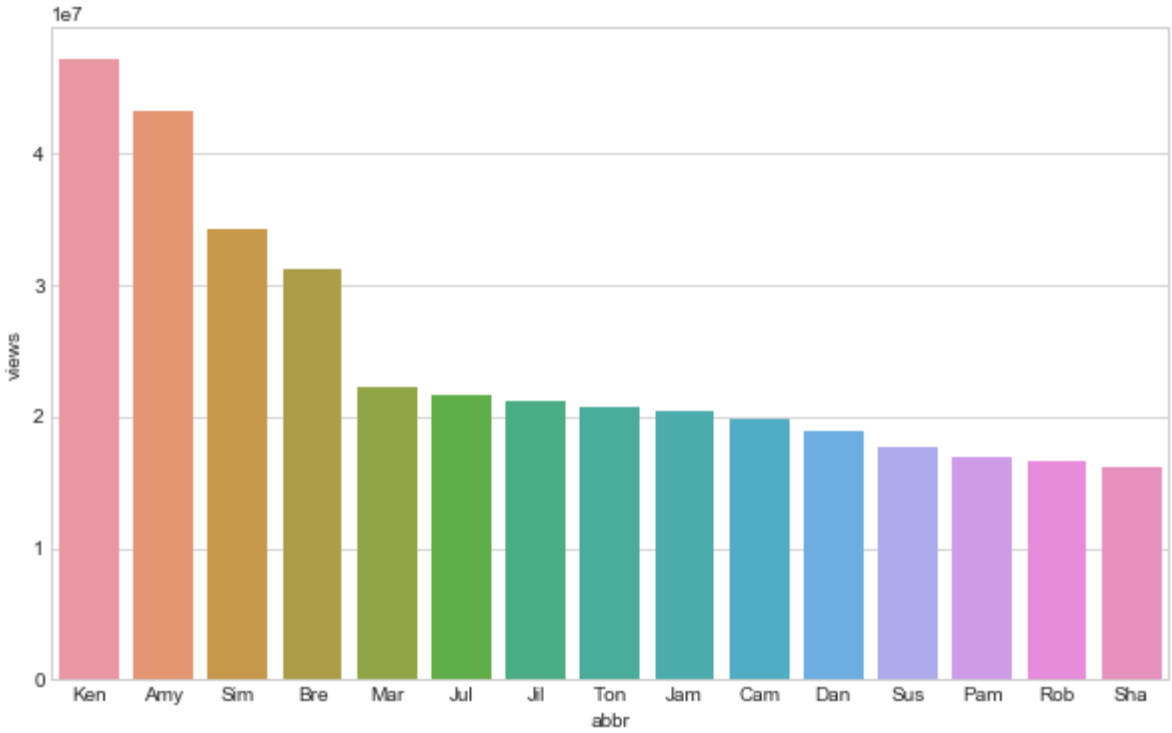
Observations

- Ken Robinson's talk on **Do schools kill creativity?** is the most popular TED Talk with 47.2 million views
- Also, it is the first ever talk to be uploaded on TED
- Robinson's talk is closely followed by Amy Cuddy's talk on **Your body language may shape who you are**
- There are only 2 talks above 40 million views abd 4 talks above 30 million

Lets create a bar chart to visualize these 15 talks in terms of the number of views they garnered

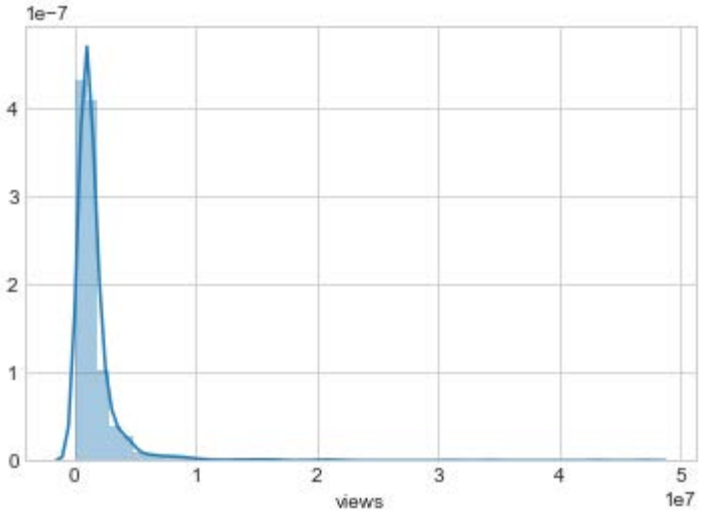
In [9]:
```
pop_talks['abbr'] = pop_talks['main_speaker'].apply(lambda x: x[:3])
sns.set_style("whitegrid")
plt.figure(figsize=(10,6))
sns.barplot(x='abbr', y='views', data=pop_talks)
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x9cf07b0>



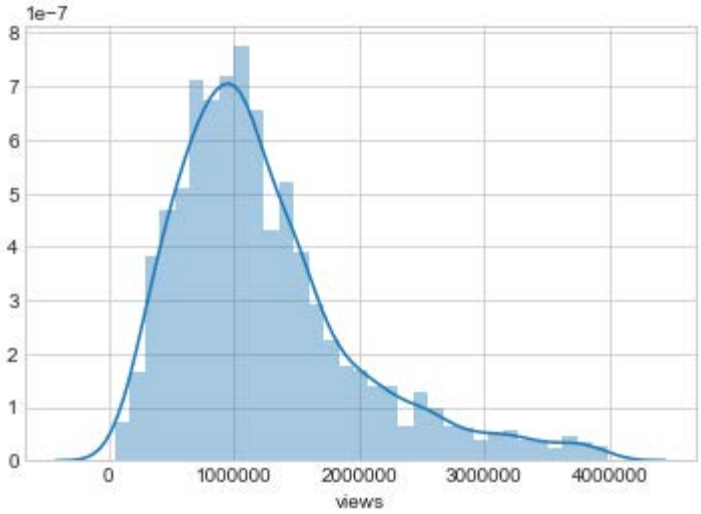
In [10]:
```
sns.distplot(df['views'])
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1484fcd0>



In [11]:
```
sns.distplot(df[df['views'] < 4e6]['views'])
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x148a1570>
```



In [12]:
```
df['views'].describe()
```

Out[12]:
```
count    2.550000e+03
mean     1.698297e+06
std      2.498479e+06
min      5.044300e+04
25%      7.557928e+05
50%      1.124524e+06
75%      1.700760e+06
max      4.722711e+07
Name: views, dtype: float64
```

Comments

In [13]:
```
df['comments'].describe()
```

Out[13]:
```
count    2550.000000
mean      191.562353
```

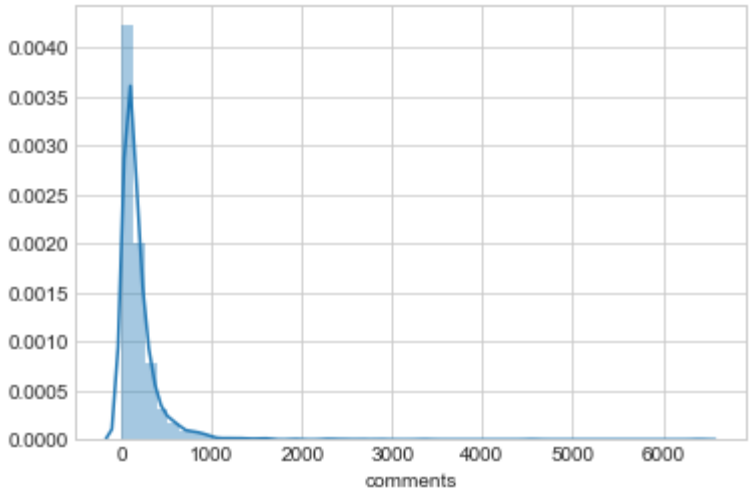
```
std       282.315223
min         2.000000
25%        63.000000
50%       118.000000
75%       221.750000
max      6404.000000
Name: comments, dtype: float64
```

Observations

- On average, there are 191.5 comments on every TED talk. Assuming the comments are constructive criticism, we can conclude that the TED Online Community is highly involved in discussions revolving TED Talks
- There is a huge standard deviation associated with the comments. In fact, it is even larger than the mean suggesting that the measures may be sensitive to outliers. We shall plot this to check the nature of the distribution
- The minimum number of comments on a talk is **2** and the maximum is **6404**. The range is 6404. The min number, though, may be as a result of the talk being posted extremely recently

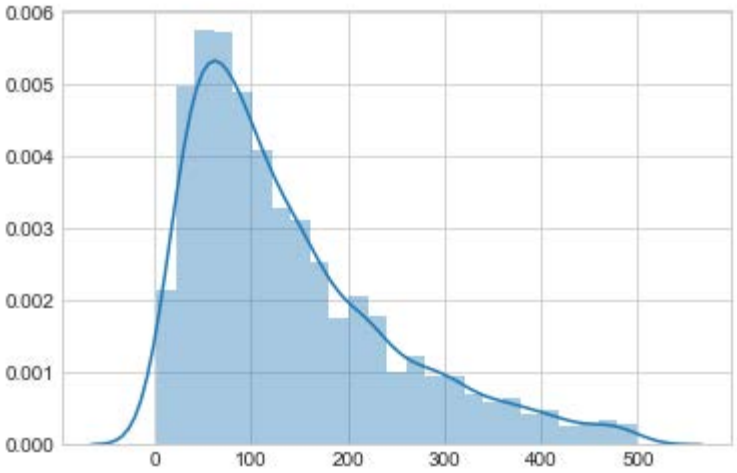
In [14]:
```
sns.distplot(df['comments'])
```

Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x148fd5b0>
```



In [15]:
```
sns.distplot(df[df['comments'] < 500][['comments']])
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x148f1bf0>
```

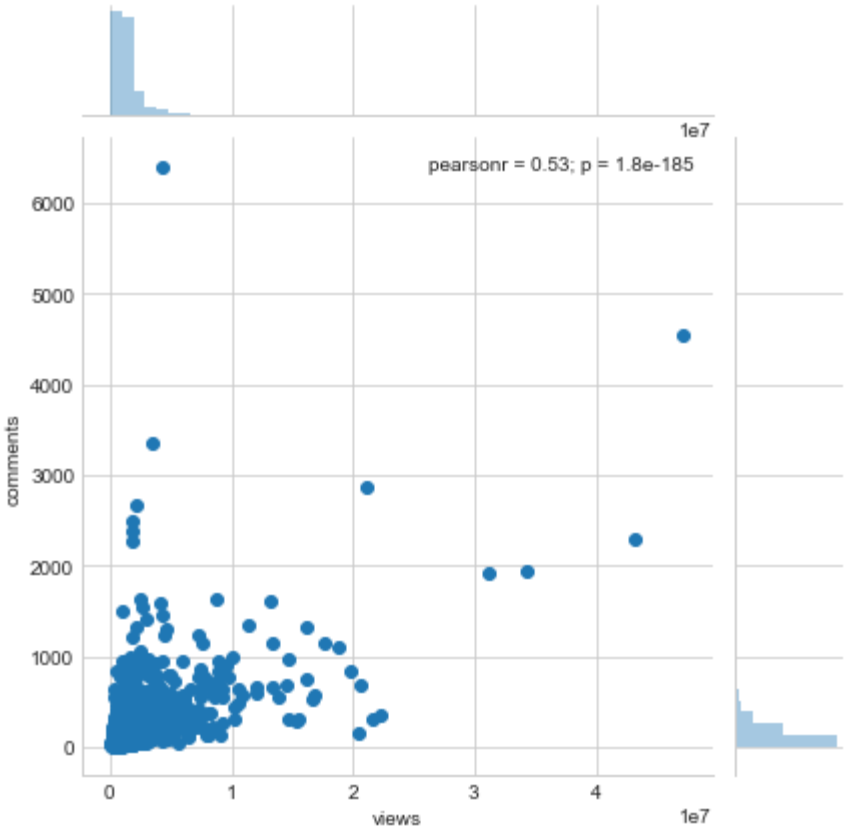


From the plot above, we can see that the bulk of the talks have fewer than 500 comments. This clearly suggests that the mean obtained above has been heavily influenced by outliers. This is possible because the number of samples is only 2550 talks.

Another question emerges that if the number of views is correlated with the number of comments then as more popular the videos tend to be then do they have more comments?

In [16]:
```
sns.jointplot(x='views', y='comments', data=df)
```

Out[16]:
```
<seaborn.axisgrid.JointGrid at 0x15b062b0>
```



In [17]:
```
df[['views', 'comments']].corr()
```

Out[17]:

	views	comments
views	1.000000	0.530939
comments	0.530939	1.000000

As the scatterplot and the correlation matrix show, the pearson coefficient is slightly more than 0.5. This suggests a medium to strong correlation between the two quantities. This result was pretty expected as mentioned above. Let us now check the number of views and comments on the 10 most commented TED Talks of all time.

In [18]:
```
df[['title', 'main_speaker', 'views', 'comments']].sort_values('comments',
 ascending = False).head(10)
```

Out[18]:

	title	main_speaker	views	comments
96	Militant atheism	Richard Dawkins	4374792	6404
0	Do schools kill creativity?	Ken Robinson	47227110	4553
644	Science can answer moral questions	Sam Harris	3433437	3356
201	My stroke of insight	Jill Bolte Taylor	21190883	2877
1787	How do you explain consciousness?	David Chalmers	2162764	2673
954	Taking imagination seriously	Janet Echelman	1832930	2492
840	On reading the Koran	Lesley Hazleton	1847256	2374
1346	Your body language may shape who you are	Amy Cuddy	43155405	2290
661	The danger of science denial	Michael Specter	1838628	2272
677	How great leaders inspire action	Simon Sinek	34309432	1930

As we can see, Richard Dawkins'talk on Militant Atheism generated the greatest amount of discussion and opinions despite having significantly lesser views than Ken Robinson's talk, which is second in the list.

So, Which talks tend to attract the largest amount of discussion?

To answer this question, lets define a new feature discussion quotient which is simply the ratio of the number of comments to the number of views. We will then check which talks have the largest discussion quotient.

In [19]:
```
df['dis_quo'] = df['comments']/df['views']
```

In [20]:
```
df[['title', 'main_speaker', 'views', 'comments', 'dis_quo', 'film_date']]
.sort_values('dis_quo', ascending=False).head(10)
```

Out[20]:

	title	main_speaker	views	comments	dis_quo	film_date
744	The case for same-sex	Diane J. Savino	292395	649	0.002220	02-12-

	marriage					2009
803	E-voting without fraud	David Bismark	543551	834	0.001534	14-07-2010
96	Militant atheism	Richard Dawkins	4374792	6404	0.001464	02-02-2002
694	Inside a school for suicide bombers	Sharmeen Obaid-Chinoy	1057238	1502	0.001421	10-02-2010
954	Taking imagination seriously	Janet Echelman	1832930	2492	0.001360	03-03-2011
840	On reading the Koran	Lesley Hazleton	1847256	2374	0.001285	10-10-2010
876	Curating humanity's heritage	Elizabeth Lindsey	439180	555	0.001264	08-12-2010
1787	How do you explain consciousness?	David Chalmers	2162764	2673	0.001236	18-03-2014
661	The danger of science denial	Michael Specter	1838628	2272	0.001236	11-02-2010
561	Dance to change the world	Mallika Sarabhai	481834	595	0.001235	04-11-2009

As we can see, the discussion quotient for **The Case for Same Sex Marriage** is 0.00222 which has relatively less views and comments as compared to Militant atheism which is highest number of comments.

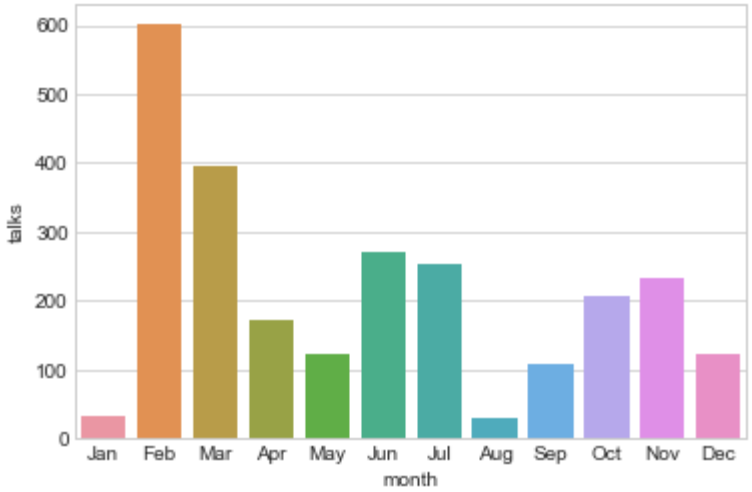
Ananlysing TED Talks by the month and the year

TED (especially TEDx) Talks tend to occur all throughout the year. Is there a hot month as far is TED is concerned? In other words, how are the talks distributed throughout the months since its inception? Let us find out.

In [21]:
```
df['month'] = df['film_date'].apply(lambda x: month_order[int(x.split('-')
[1]) - 1])
month_df = pd.DataFrame(df['month'].value_counts()).reset_index()
month_df.columns = ['month', 'talks']
```

In [22]:
```
sns.barplot(x='month', y='talks', data = month_df, order = month_order)
```

Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15d78d90>
```

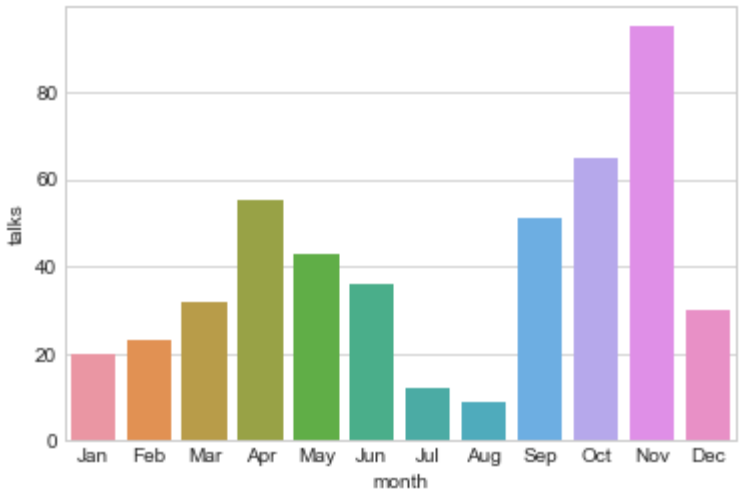


February is clearly the most popular month for TED Conferences wheresas August and January are the least popular. February's popularity is largely due to the official TED Conferences which are held in this month. Let us check the distribution for TEDx talks only.

In [23]:
```
df_x = df[df['event'].str.contains('TEDx')]
x_month_df = pd.DataFrame(df_x['month'].value_counts().reset_index())
x_month_df.columns = ['month', 'talks']
```

In [24]:
```
sns.barplot(x='month', y='talks',data = x_month_df, order = month_order)
```

Out[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15ea5eb0>
```



As far as TEDx talks are concerned, **November** is the most popular month. However, we cannot take this result at face value as very few of the TEDx talks are actually uploaded to the TED website and therefore, it is entirely possible that the sample in our dataset is not at all representative of all TEDx talks. A slightly more accurate statement would be that **the most popular TEDx talks take place the most in October and November**.

The next question that arises is the most popular days for conducting TED and TEDx conferences. The tools applied are very sensible to the procedure applied for months.

In [25]:
```
def getday(x):
```

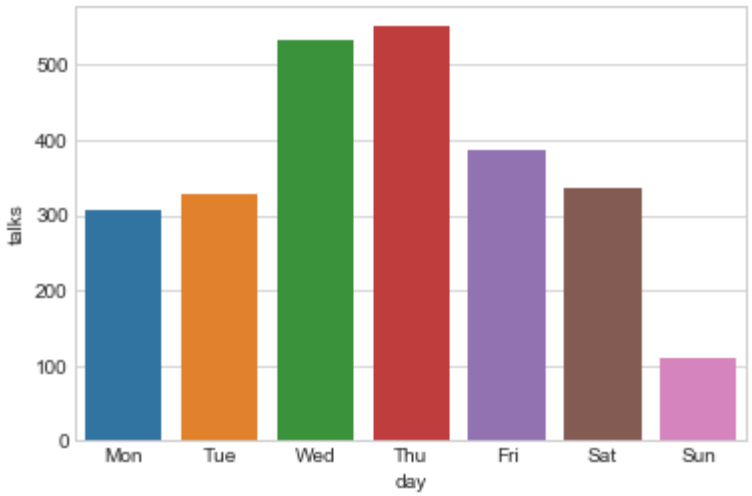
```
    day, month, year = (int(i) for i in x.split('-'))
    answer = datetime.date(year, month, day).weekday()
    return day_order[answer]
```

In [26]:
```
df['day'] = df['film_date'].apply(getday)
```

In [27]:
```
day_df = pd.DataFrame(df['day'].value_counts()).reset_index()
day_df.columns = ['day', 'talks']
```

In [28]:
```
sns.barplot(x = 'day', y ='talks', data=day_df, order = day_order)
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15f0d910>
```



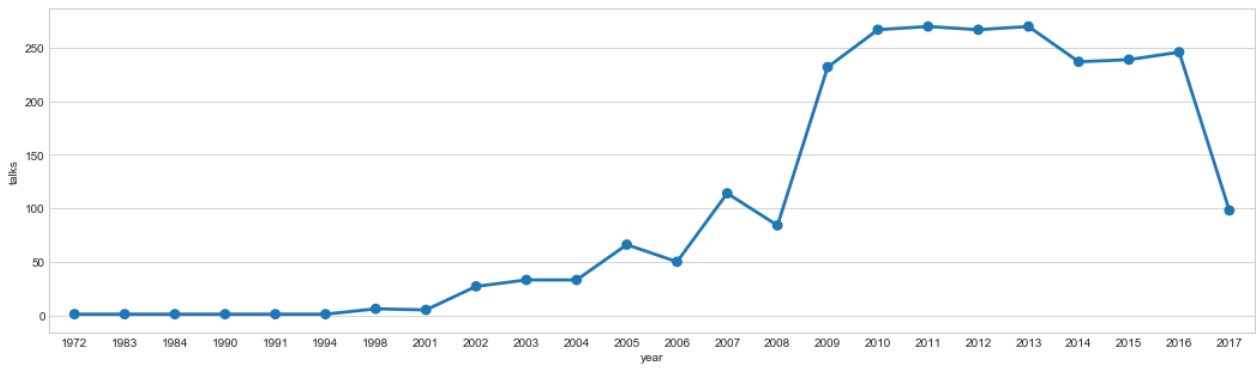
The distribution of days is almost a bell curve with **Wednesday and Thursday** being the most popular days and Sunday being the least popular. This is pretty interesting because most TED Conferences would happen sometime in the weekend.

Let's visualize the number of TED talks through the years and check the integrity of our analysis.

In [29]:
```
df['year'] = df['film_date'].apply(lambda x: x.split('-')[2])
year_df = pd.DataFrame(df['year'].value_counts().reset_index())
year_df.columns = ['year', 'talks']
```

In [30]:
```
plt.figure(figsize=(18,5))
sns.pointplot(x = 'year', y ='talks', data=year_df)
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15f603d0>
```



Observations

- As expected, the number of TED Talks have gradually increased over the years since its inception in 1984
- There was sharp increase in the number of talks in 2009. It might be interesting to know the reasons behind 2009 being the tipping point wher the number of talks increased more than twofold.
- The number of talks have been pretty much constant since 2009.

Finally, to put it all together, lets construct a heat map that shows the number of talks by month and year. This will give us a good summary of the distribution of talks.

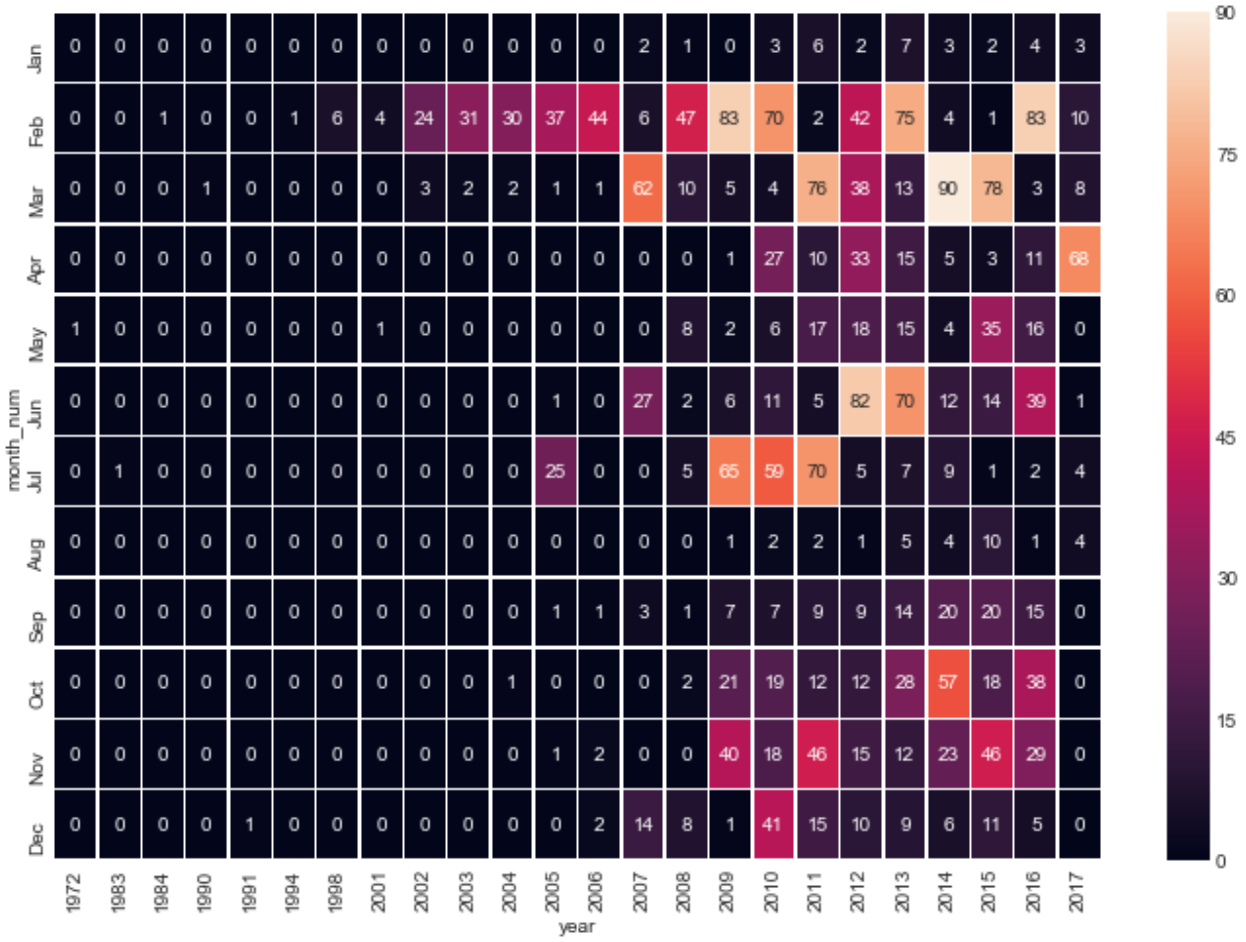
Heat Map

In [31]:
```
months = {'Jan': 1, 'Feb': 2, 'Mar': 3, 'Apr': 4, 'May': 5, 'Jun': 6,
          'Jul': 7, 'Aug': 8, 'Sep': 9, 'Oct': 10, 'Nov': 11, 'Dec': 12}
```

In [32]:
```
hmap_df = df.copy()
hmap_df['film_date'] = hmap_df['film_date'].apply(lambda x: month_order[in
t(x.split('-')[1]) - 1] + " " + str(x.split('-')[2]))
hmap_df = pd.pivot_table(hmap_df[['film_date', 'title']], index='film_date
', aggfunc='count').reset_index()
hmap_df['month_num'] = hmap_df['film_date'].apply(lambda x: months[x.split
()[0]])
hmap_df['year'] = hmap_df['film_date'].apply(lambda x: x.split()[1])
hmap_df = hmap_df.sort_values(['year', 'month_num'])
hmap_df = hmap_df[['month_num', 'year', 'title']]
hmap_df = hmap_df.pivot('month_num', 'year', 'title')
hmap_df = hmap_df.fillna(0)
```

In [33]:
```
f, ax = plt.subplots(figsize=(12,8))
sns.heatmap(hmap_df, annot = True, linewidths=.5, ax=ax, fmt='n', yticklab
els= month_order)
```

Out[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15f21430>
```



TED Speakers

In this section, we will check the most popular TED speakers and who has given the maximum number of talks

In [34]:
```
speaker_df = df.groupby('main_speaker').count().reset_index()[['main_speaker', 'comments']]
speaker_df.columns = ['main_speaker', 'appearences']
speaker_df = speaker_df.sort_values('appearences', ascending = False)
speaker_df.head(10)
```

Out[34]:

	main_speaker	appearences
770	Hans Rosling	9
1066	Juan Enriquez	7
1693	Rives	6
1278	Marco Tempest	6
397	Clay Shirky	5
1487	Nicholas Negroponte	5
1075	Julian Treasure	5

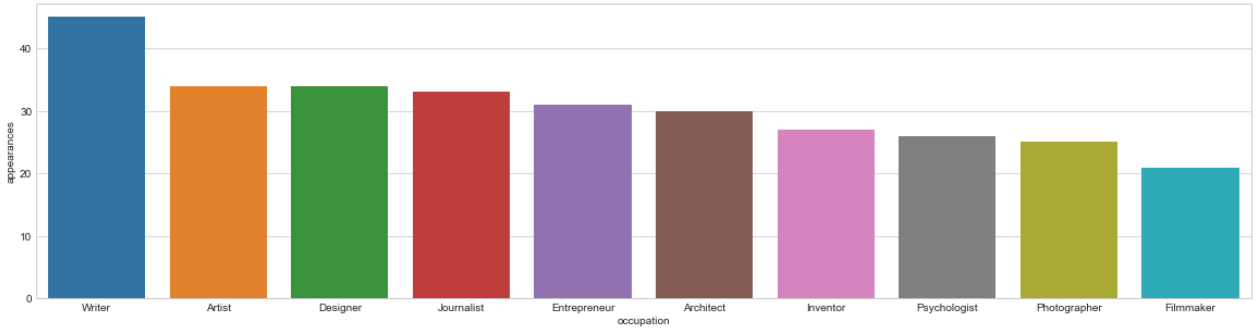
424	Dan Ariely	5
850	Jacqueline Novogratz	5
248	Bill Gates	5

Hans Rosling, the Swiss Health Professor is clearly the most popular TED speaker, with 9 appearances on the TED Forum. Followed by Juan Enriquez with 7 appearances. Rives and Macro Tempest have graced the TED platform 6 times.

Let's check which occupation you should choose to become a TED speaker. Lets have a look at what kind of people TED is most interested in inviting to its events.

In [35]:
```
occupation_df = df.groupby('speaker_occupation').count().reset_index()[['speaker_occupation', 'comments']]
occupation_df.columns = ['occupation', 'appearances']
occupation_df = occupation_df.sort_values('appearances', ascending = False)
```

In [36]:
```
plt.figure(figsize=(20,5))
sns.barplot(x='occupation', y='appearances', data=occupation_df.head(10))
plt.show()
```



Observations

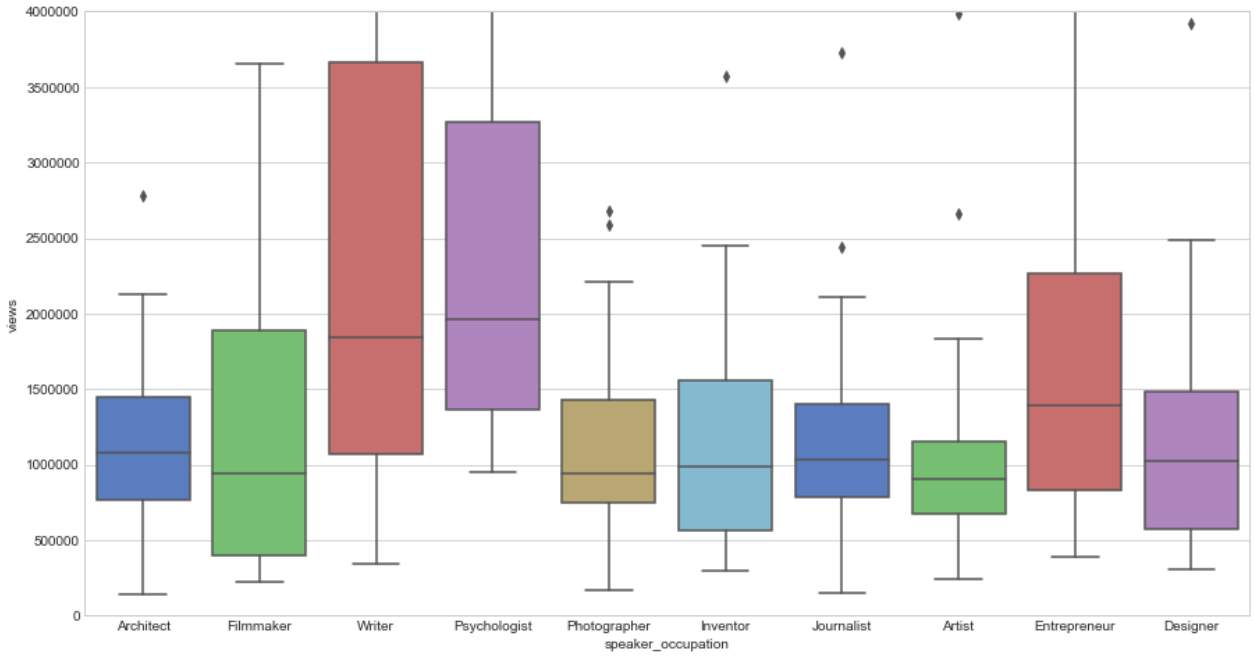
- **Writers** are the most popular with more than 45 speakers identifying themselves as the aforementioned
- **Artists** and **Designers** come a distant second with around 35 speakers in each category.
- This result must be taken with a pinch of salt as a considerable number of speakers identify themeselves with multiple professions (for example, writer/entrepreneur). Performing an analysis taking this intro consideration is left as an exercise to the reader.

Do some professions tend to attract a larger number of viewers? Lets visualize the relationship the relationship between the top 10 most popular professions and the views that generate in the form of a box plot.

In [37]:
```
fig, ax = plt.subplots(nrows = 1, ncols = 1, figsize = (15, 8))
sns.boxplot(x = 'speaker_occupation', y = 'views', data = df[df['speaker_occupation'].isin(occupation_df.head(10)['occupation'])], palette='muted', ax=ax)
```

```
ax.set_ylim([0, 0.4e7])
```

Out[37]: (0, 4000000.0)



On average, out of the top 10 most popular professions, **Psychologists** tend to garner the most views. **Writers** have the greatest range of views between the first an the third quartile.

Finally, let us check the number of talks which have had more than one speaker.

In [38]:
```
df['num_speaker'].value_counts()
```

Out[38]:
```
1    2492
2      49
3       5
4       3
5       1
Name: num_speaker, dtype: int64
```

Almost every talk has one speaker. There are close to 50 talks where two people shared the stage. The max number of speakers to share a single stage was 5. I suspect this was a dance performance. Lets have a look.

In [39]:
```
df[df['num_speaker'] == 5][['title', 'description', 'main_speaker', 'event
']]
```

Out[39]:

	title	description	main_speaker	event
2507	A dance to honor Mother Earth	Movement artists Jon Boogz and Lil Buck debut ...	Jon Boogz and Lil Buck	TED2017

I was correct. It is a talk titled **A dance to honor Mother Earth** by Jon Boogz and Lil Buck at the TED 2017 COnference.

TED Events

Which TED Events tend to hold the most number of TED.com upload worthy events? We will try to answer that question in this section.

In [40]:
```
events_df = df[['title', 'event']].groupby('event').count().reset_index()
events_df.columns = ['event', 'talks']
events_df = events_df.sort_values('talks', ascending = False)
events_df.head(10)
```

Out[40]:

	event	talks
64	TED2014	84
59	TED2009	83
63	TED2013	77
66	TED2016	77
65	TED2015	75
99	TEDGlobal 2012	70
61	TED2011	70
60	TED2010	68
98	TEDGlobal 2011	68
57	TED2007	68

TED Languages

In [41]:
```
df['languages'].describe()
```

Out[41]:
```
count    2550.000000
mean       27.326275
std         9.563452
min         0.000000
25%        23.000000
50%        28.000000
75%        33.000000
max        72.000000
Name: languages, dtype: float64
```

On average, a TED talk is available in 27 different languages. The maximum number of languages a TED talk is available is a staggering 72. Let us check which talk this is.

In [42]:
```
df[df['languages'] == 72]
```

Out[42]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	du
	Matt		Is there				

973	Cutts: Try something new for 30 days	Try something new for 30 days	something you've always meant to do, ...	Matt Cutts	Technologist	1	20

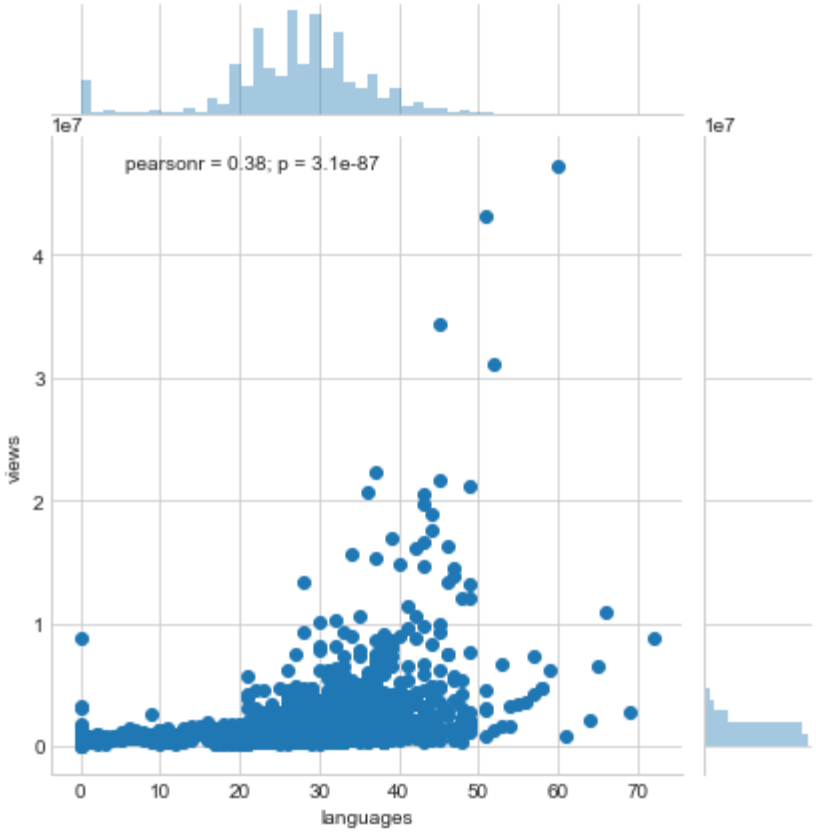
1 rows × 21 columns

The most translated TED talk of all time is **Matt Cutts: Try something new for 30 days**. The talk does have a very universal theme of exploration. The sheer number of languages it's available in demands a little more inspection though as it has just over 80 million views, far fewer than the most popular TED Talk.

Finally, let us check there is a correlation between the number of views and the number of languages a talk is available in. We would think that this should be the case since the talk is more accessible to a larger number of people but as Matt Cutt's talk shows, it may be not really be the case.

In [43]:
```
sns.jointplot(x= 'languages', y='views', data=df)
```

Out[43]:
```
<seaborn.axisgrid.JointGrid at 0x16faf7f0>
```



The pearson coefficient is 0.38 suggesting a medium correlation between the aforementioned quatities.

TED Themes

In this section, we will try to find out the most popular themes in the TED conferences. Although TED started out as a conference about technology, entertainment and design, it has since diversified into virtually every field of study and walk of life. It will be interesting to see if this conference with Silicon Valley origins has a bias towards certain topics.

To answer this question, we need to wrangle our data in a way that it is suitable for analysis. More specifically, we need to split the related_tags list into separate rows.

In [44]:
```
import ast
df['tags'] = df['tags'].apply(lambda x: ast.literal_eval(x))
```

In [45]:
```
s = df.apply(lambda x: pd.Series(x['tags']), axis=1).stack().reset_index(l
evel=1,drop=True)
s.name = 'theme'
```

In [46]:
```
theme_df = df.drop('tags', axis=1).join(s)
theme_df.head()
```

Out[46]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	durat
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	1164
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	1164
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	1164
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	1164
0	Ken Robinson: Do	Do schools	Sir Ken Robinson makes an	Ken Robinson	Author/educator	1	1164

	schools kill creativity?	kill creativity?	entertaining and pro...				

5 rows × 21 columns

In [47]:
```
len(theme_df['theme'].value_counts())
```

Out[47]: 416

TED defines a staggering **416 different categories** for its talks. Let us now check the most popular themes.

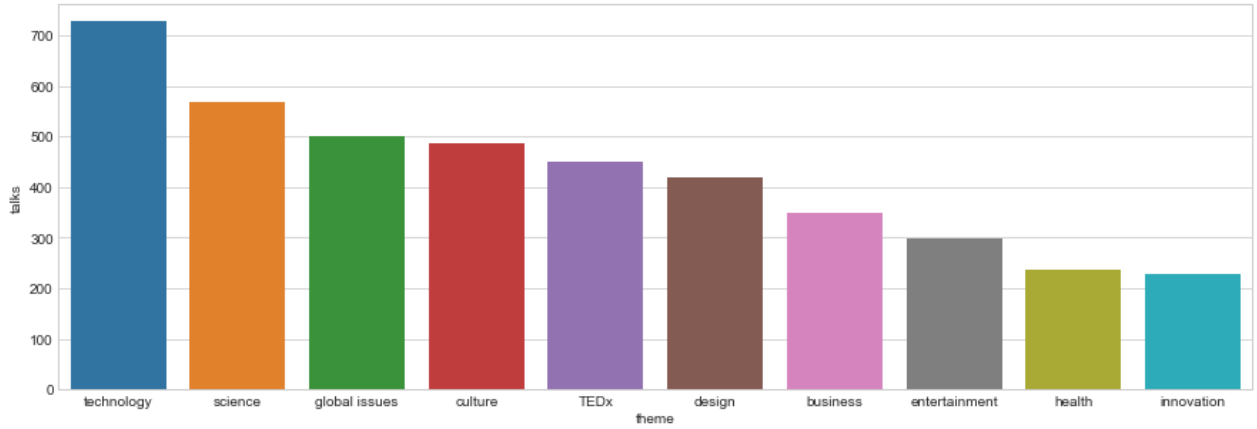
In [48]:
```
pop_themes = pd.DataFrame(theme_df['theme'].value_counts()).reset_index()
pop_themes.columns = ['theme', 'talks']
pop_themes.head(10)
```

Out[48]:

	theme	talks
0	technology	727
1	science	567
2	global issues	501
3	culture	486
4	TEDx	450
5	design	418
6	business	348
7	entertainment	299
8	health	236
9	innovation	229

In [49]:
```
plt.figure(figsize =(15,5))
sns.barplot(x='theme', y='talks', data=pop_themes.head(10))
```

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x1737f690>



As may have been expected, Technology is the most popular topic for talks. The other two original factions, Design and Entertainment, also make it to the list of top 10 themes. Science and Global Issues are the second and the third most popular themes respectively.

The next question arises is the trends in the share of topics of TED Talks across the world. Has the demand for Technology talks increased? Do certain years have a disproportionate share of talks related to global issues? Let's find out!

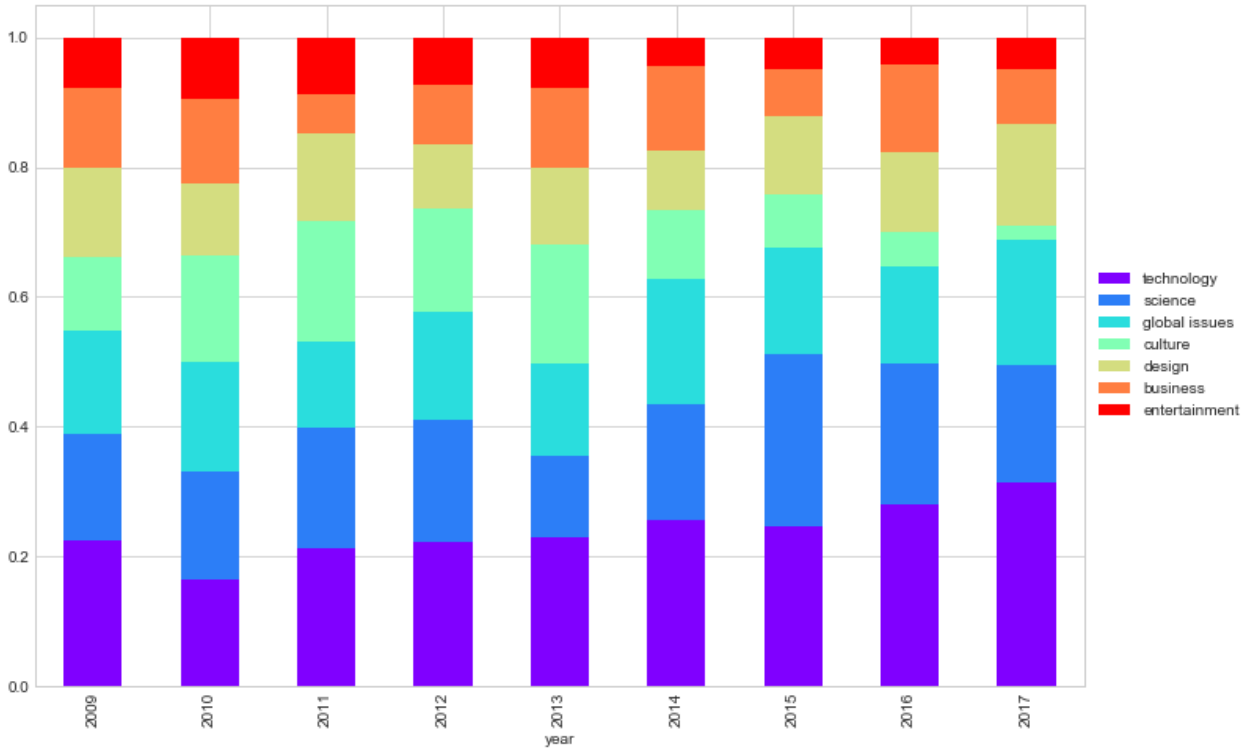
We will only consider the top 7 themes, excluding TEDx and talks after 2009, the year when the number of TED talks really peaked.

In [50]:
```
pop_theme_talks = theme_df[(theme_df['theme'].isin(pop_themes.head(8)['theme'])) &
                           (theme_df['theme'] != 'TEDx')]
pop_theme_talks['year'] = pop_theme_talks['year'].astype('int')
pop_theme_talks = pop_theme_talks[pop_theme_talks['year'] > 2008]
```

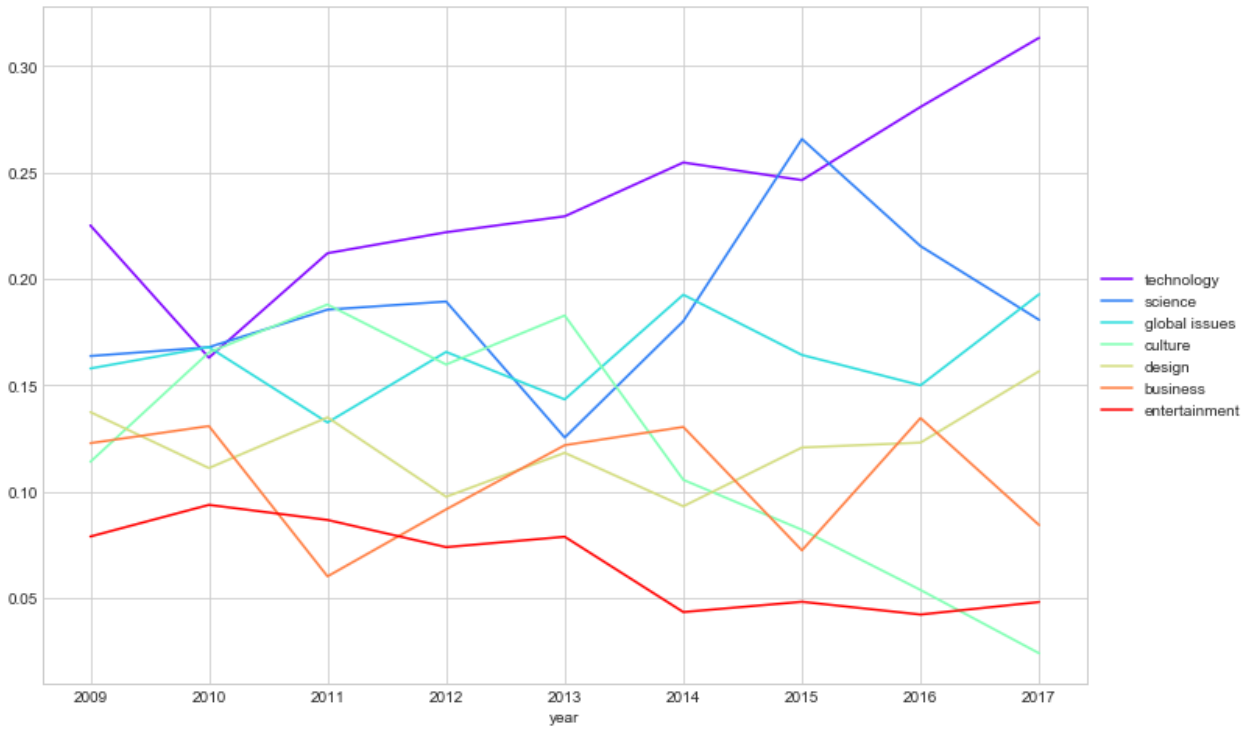
```
c:\users\arnab tarwani\appdata\local\programs\python\python36-32\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [51]:
```
themes = list(pop_themes.head(8)['theme'])
themes.remove('TEDx')
ctab = pd.crosstab([pop_theme_talks['year']], pop_theme_talks['theme']).apply(
lambda x: x/x.sum(), axis=1)
ctab[themes].plot(kind='bar', stacked=True, colormap='rainbow', figsize=(12,8)).legend(loc='center left', bbox_to_anchor=(1, 0.5))
```

Out[51]:
```
<matplotlib.legend.Legend at 0x17363970>
```



In [52]:
```
ctab[themes].plot(kind='line', stacked=False, colormap='rainbow', figsize=
(12,8)).legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



The proportion of technology talks has steadily increased over the years with a slight dip in 2010. This is understandable considering the boom of technologies such as blockchain, deep learning and augmented reality capturing people's imagination.

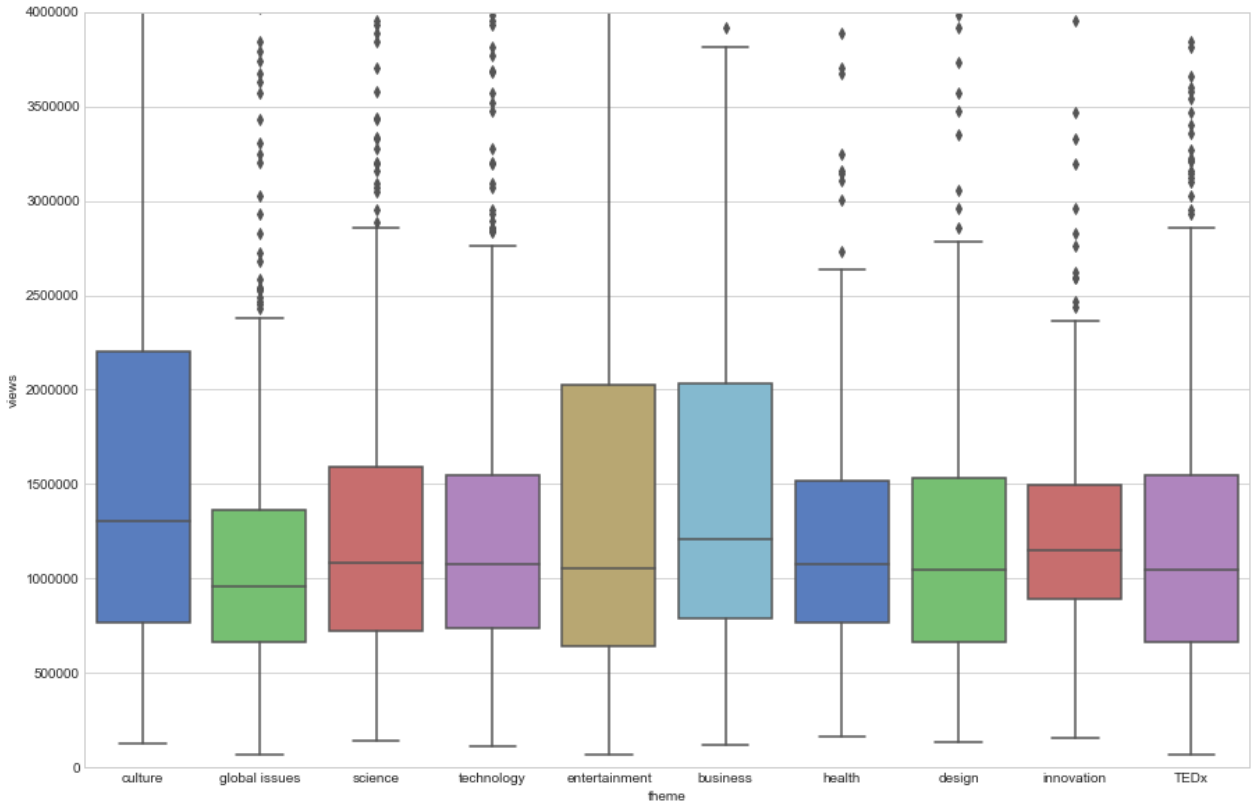
Talks on culture have witnessed a dip, decreasing steadily starting 2013. The share of culture talks has been the least in 2017. Entertainment talks seem to have witnessed a slight decline in popularity

since 2009.

Like with the speaker occupations, let us investigate if certain topics tend to garner more views than a certain other topics. We will be doing this analysis for the top ten categories that we discovered in an earlier cell. As with the speaker occupations, the box plot will be used to deduce this relation.

In [53]:
```
pop_theme_talks = theme_df[theme_df['theme'].isin(pop_themes.head(10)['theme'])]
fig, ax = plt.subplots(nrows=1, ncols = 1, figsize = (15,10))
sns.boxplot(x= 'theme', y='views', data=pop_theme_talks, palette='muted', ax=ax)
ax.set_ylim([0,0.4e7])
```

Out[53]:
```
(0, 4000000.0)
```



Although culture has lost share in the number of TED talks over the years, they garner the highest number of views.

TED Duration and Word Counts

In [54]:
```
# Convert to minutes
df['duration'] = df['duration']/60
df['duration'].describe()
```

Out[54]:
```
count    2550.000000
mean       13.775170
std         6.233486
min         2.250000
25%         9.616667
50%        14.133333
```

```
75%        17.445833
max        87.600000
Name: duration, dtype: float64
```

TED Talks, on an average are 13.7 minutes long. Whis is surprising as TED Talks are often 18 minutes or longer and the average is a good 3 minutes shorter than that.

The shortest TED talk on record is 2.25 minutes long whereas the longest talk is 87.6 minutes. The longest talk was not actually a TED talk. Lets take a look at both the shortest and longest talks.

In [55]:
```
df[df['duration'] == 2.25]
```

Out[55]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	dura
239	Murray Gell-Mann: The ancestor of language	The ancestor of language	After speaking at TED2007 on elegance in physi...	Murray Gell-Mann	Physicist	1	2.25

1 rows × 21 columns

In [56]:
```
df[df['duration'] == 87.6]
```

Out[56]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	dura
640	Douglas Adams: Parrots, the universe and every...	Parrots, the universe and everything	Blind river dolphins, reclusive lemurs, a parr...	Douglas Adams	Author, satirist	1	87.6

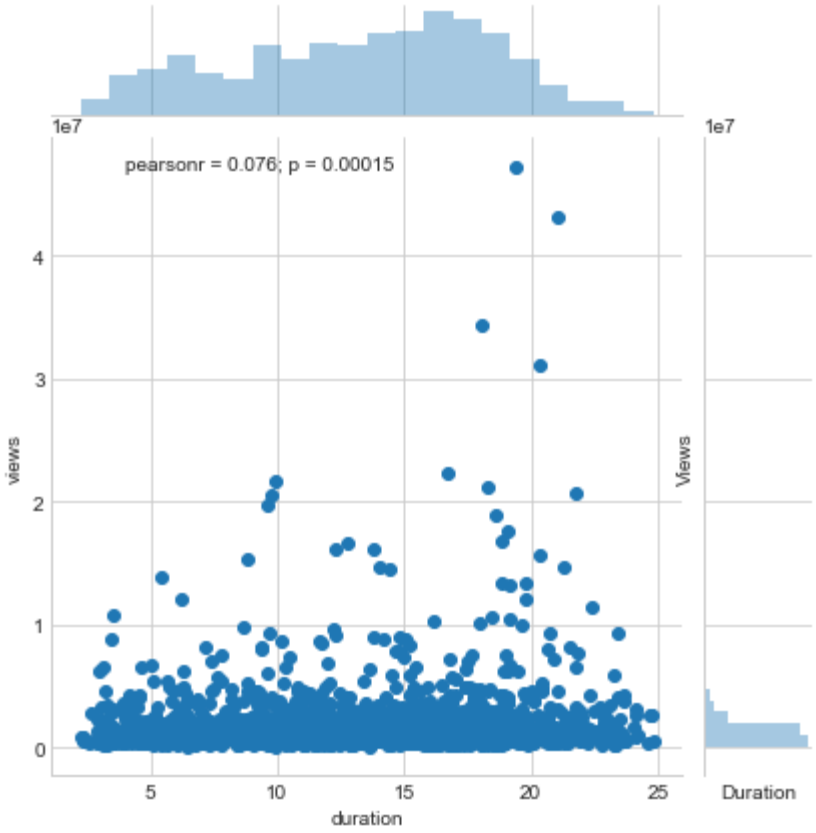
1 rows × 21 columns

The shortest talk was at TED2007 titled **The ancestor of language** by Murray Gell-Mann. The longest talk on TED.com, as we had guessed, it is not a TED Talk at all. Rather, it was a talk titled **Parrots, the universe and everything** delivered by Douglas Adams at the University of California in 2001.

Let us now check for any correlation between the popularity and the duration of TED Talk. To make sure we only include TED Talks, we will consider only those talks which have a duration less than 25 minutes.

In [57]:
```
sns.jointplot(x= 'duration',y='views', data= df[df['duration'] < 25])
plt.xlabel('Duration')
plt.ylabel('Views')
```

Out[57]:
```
Text(336.971,0.5,'Views')
```



There seems to be almost no correlation between these two quantities. This strongly suggests that there is no tangible correlation between the length of and the length and the popularity of a TED Talk. Content is king at TED.

Next, we'll look at transcripts to get an idea of word count. For this, we'll introduce our second dataset, the one which contains all the transcripts.

In [58]:
```
df2=pd.read_csv('../TED_TALKS/Data/transcripts.csv')
df2.head()
```

Out[58]:

	transcript	url
0	Good morning. How are you?(Laughter)It's been ...	https://www.ted.com/talks/ken_robinson_says_sc...
1	Thank you so much, Chris. And it's truly a gre...	https://www.ted.com/talks/al_gore_on_averting_...
2	(Music: "The Sound of Silence," Simon & Garfun...	https://www.ted.com/talks/david_pogue_says_sim...
3	If you're here today — and I'm very happy that...	https://www.ted.com/talks/majora_carter_s_tale...

4	About 10 years ago, I took on the task to teac...	https://www.ted.com/talks/hans_rosling_shows_t...

In [59]:
```
len(df2)
```

Out[59]: 2467

It seems that we have data available for 2467 talks. Lets perform a join of the two dataframes on the url feature to include word counts for every talk.

In [60]:
```
df3 = pd.merge(left=df, right=df2, how='left', left_on='url', right_on='ur
l')
df3.head()
```

Out[60]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	dur
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	19.40
1	Al Gore: Averting the climate crisis	Averting the climate crisis	With the same humor and humanity he exuded in ...	Al Gore	Climate advocate	1	16.28
2	David Pogue: Simplicity sells	Simplicity sells	New York Times columnist David Pogue takes aim...	David Pogue	Technology columnist	1	21.43
3	Majora Carter: Greening the ghetto	Greening the ghetto	In an emotionally charged talk, MacArthur-winn...	Majora Carter	Activist for environmental justice	1	18.60
4	Hans Rosling: The best stats you've ever seen	The best stats you've ever seen	You've never seen data presented like this. Wi...	Hans Rosling	Global health expert; data visionary	1	19.83

5 rows × 22 columns

In [61]:
```
df3['transcript'] = df3['transcript'].fillna('')
df3['wc'] = df3['transcript'].apply(lambda x: len(x.split()))
```

In [62]:
```
df3['wc'].describe()
```

Out[62]:
```
count    2553.000000
mean     1971.550725
std      1009.494329
min         0.000000
25%      1235.000000
50%      1983.000000
75%      2681.000000
max      9044.000000
Name: wc, dtype: float64
```

We can see that the average TED Talk has around 1971 words and there is a significantly large standard deviation of a 1009 words. The longest talk is about 9044 words on length.

Like duration, there shouldn;t be any correlation betweem number of words and views. We will proceed to look at a more interesting statistic: the number of words per minute.

In [63]:
```
df3['wpm'] = df3['wc']/df3['duration']
df3['wpm'].describe()
```

Out[63]:
```
count    2553.000000
mean      142.147752
std        39.635348
min         0.000000
25%       131.069182
50%       149.018182
75%       164.984615
max       247.364865
Name: wpm, dtype: float64
```

The average words per minute a TED speaker enunciates are about 142. The fastest speaker spoke a staggering 247 words per minute which is much higher than the average 125-150 words per minute in English. Let us see who this is!

In [64]:
```
df3[df3['wpm'] > 245]
```

Out[64]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	durat
441	Mae Jemison: Teach arts and sciences together	Teach arts and sciences together	Mae Jemison is an astronaut, a doctor, an art ...	Mae Jemison	Astronaut, engineer, entrepreneur, physician a...	1	14.8

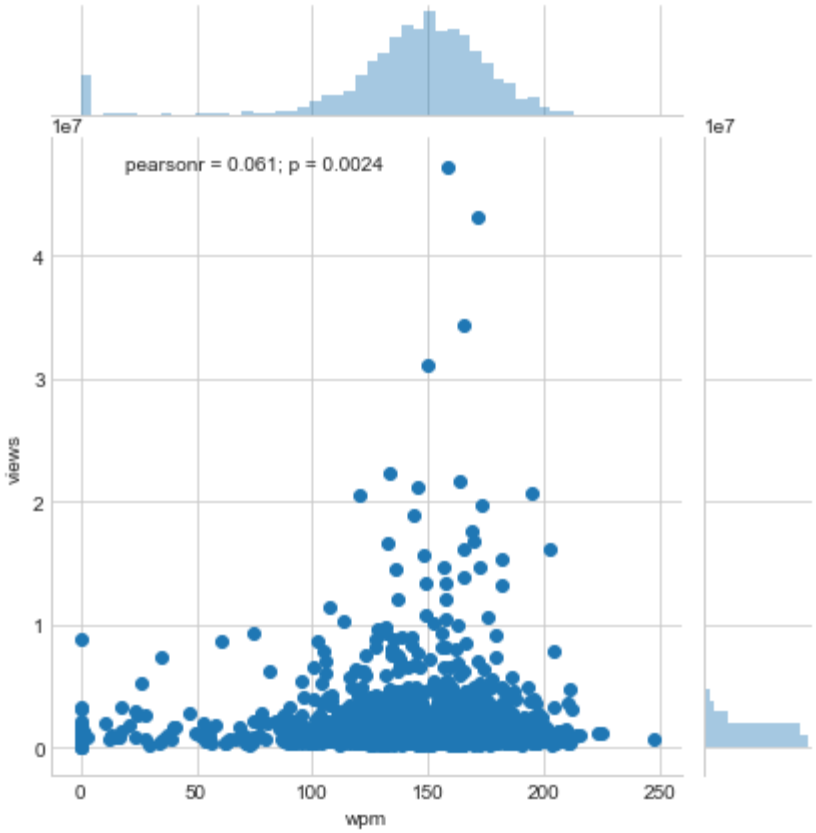
1 rows × 24 columns

The person is **Mae Jemison** with a talk on **Teach arts and sciences together** at the TED2002 conference. We should take this result with a pinch of salt as the speaker wasn't quite as fast as it seems.

Finally, in this section, we'll see the correlation between words per minute and popularity.

In [65]:
```
sns.jointplot(x='wpm', y='views', data=df3[df3['duration'] < 25])
```

Out[65]:
```
<seaborn.axisgrid.JointGrid at 0x17404170>
```



Again, there is practically no correlation.If you are going to give a TED Talk, you probably shouldn't worry if you're speaking a little faster or slower than usual.

TED Ratings

In [66]:
```
df.iloc[1]['ratings']
```

Out[66]:
```
"[{'id': 7, 'name': 'Funny', 'count': 544}, {'id': 3, 'name': 'Courageous'
, 'count': 139}, {'id': 2, 'name': 'Confusing', 'count': 62}, {'id': 1, 'n
ame': 'Beautiful', 'count': 58}, {'id': 21, 'name': 'Unconvincing', 'count
': 258}, {'id': 11, 'name': 'Longwinded', 'count': 113}, {'id': 8, 'name':
 'Informative', 'count': 443}, {'id': 10, 'name': 'Inspiring', 'count': 41
3}, {'id': 22, 'name': 'Fascinating', 'count': 132}, {'id': 9, 'name': 'In
genious', 'count': 56}, {'id': 24, 'name': 'Persuasive', 'count': 268}, {'
id': 23, 'name': 'Jaw-dropping', 'count': 116}, {'id': 26, 'name': 'Obnoxi
```

```
ous', 'count': 131}, {'id': 25, 'name': 'OK', 'count': 203}]"
```

In [68]:
```
df['ratings'] = df['ratings'].apply(lambda x: ast.literal_eval(x))
```

In this talk we'll find which talks were rated the funniest, the most beautiful, the most confusing and most jaw dropping of all time. The rest is left to the reader to explore. We now need to define three extra features to accomplish this task.

In [72]:
```
df['funny'] = df['ratings'].apply(lambda x: x[0]['count'])
df['jawdrop'] = df['ratings'].apply(lambda x: x[-3]['count'])
df['beautiful'] = df['ratings'].apply(lambda x: x[3]['count'])
df['confusing'] = df['ratings'].apply(lambda x: x[2]['count'])
df.head()
```

Out[72]:

	name	title	description	main_speaker	speaker_occupation	num_speaker	dur
0	Ken Robinson: Do schools kill creativity?	Do schools kill creativity?	Sir Ken Robinson makes an entertaining and pro...	Ken Robinson	Author/educator	1	19.40
1	Al Gore: Averting the climate crisis	Averting the climate crisis	With the same humor and humanity he exuded in ...	Al Gore	Climate advocate	1	16.28
2	David Pogue: Simplicity sells	Simplicity sells	New York Times columnist David Pogue takes aim...	David Pogue	Technology columnist	1	21.43
3	Majora Carter: Greening the ghetto	Greening the ghetto	In an emotionally charged talk, MacArthur-winn...	Majora Carter	Activist for environmental justice	1	18.60
4	Hans Rosling: The best stats you've ever seen	The best stats you've ever seen	You've never seen data presented like this. Wi...	Hans Rosling	Global health expert; data visionary	1	19.83

5 rows × 25 columns

Funniest Talks of all time

In [73]:
```
df[['title', 'main_speaker', 'views', 'published_date', 'funny']].sort_values('funny', ascending = False)[:10]
```

Out[73]:

	title	main_speaker	views	published_date	funny
837	The power of vulnerability	Brené Brown	31168150	23-12-2010	21444
0	Do schools kill creativity?	Ken Robinson	47227110	27-06-2006	19645
1030	How to live before you die	Steve Jobs	8744428	06-10-2011	17290
201	My stroke of insight	Jill Bolte Taylor	21190883	12-03-2008	14447
1129	The happy secret to better work	Shawn Achor	16209727	01-02-2012	11213
1940	The price of shame	Monica Lewinsky	11443190	21-03-2015	8668
2109	What makes a good life? Lessons from the longe...	Robert Waldinger	16601927	23-12-2015	8590
1747	Why good leaders make you feel safe	Simon Sinek	6803938	19-05-2014	8569
553	The thrilling potential of SixthSense technology	Pranav Mistry	16097077	16-11-2009	8416
176	Underwater astonishments	David Gallo	13926113	11-01-2008	8328

Most beautiful talks of all time

In [75]:
```
df[['title', 'main_speaker', 'views', 'published_date', 'beautiful']].sort_values('beautiful', ascending= False)[:10]
```

Out[75]:

	title	main_speaker	views	published_date	beautiful
201	My stroke of insight	Jill Bolte Taylor	21190883	12-03-2008	9437
677	How great leaders inspire action	Simon Sinek	34309432	04-05-2010	8845
381	Your elusive creative genius	Elizabeth Gilbert	13155478	09-02-2009	8130
2161	Inside the mind of a master procrastinator	Tim Urban	14745406	16-03-2016	7445
1129	The happy secret to better work	Shawn Achor	16209727	01-02-2012	7315

1779	Which country does the most good for the world?	Simon Anholt	4548276	02-07-2014	6390
1346	Your body language may shape who you are	Amy Cuddy	43155405	01-10-2012	6217
500	The puzzle of motivation	Dan Pink	18830983	24-08-2009	4797
614	Teach every child about food	Jamie Oliver	7638978	11-02-2010	4779
553	The thrilling potential of SixthSense technology	Pranav Mistry	16097077	16-11-2009	4702

Most Jaw Dropping Talks of all time

In [76]:
```
df[['title', 'views', 'main_speaker', 'published_date', 'jawdrop']].sort_values('jawdrop', ascending=False)[:10]
```

Out[76]:

	title	views	main_speaker	published_date	jawdrop
4	The best stats you've ever seen	12005869	Hans Rosling	28-06-2006	2542
1163	The power of introverts	17629275	Susan Cain	03-03-2012	2467
381	Your elusive creative genius	13155478	Elizabeth Gilbert	09-02-2009	2093
1030	How to live before you die	8744428	Steve Jobs	06-10-2011	1368
0	Do schools kill creativity?	47227110	Ken Robinson	27-06-2006	1174
677	How great leaders inspire action	34309432	Simon Sinek	04-05-2010	1161
29	The surprising science of happiness	14689301	Dan Gilbert	26-09-2006	1047
117	New insights on poverty	3243784	Hans Rosling	25-06-2007	828
500	The puzzle of motivation	18830983	Dan Pink	24-08-2009	825
1170	Why you will fail to have a great career	5917201	Larry Smith	11-03-2012	752

Most Confusing Talks of all time

In [77]:
```
df[['title', 'views', 'main_speaker', 'published_date', 'confusing']].sort_values('confusing', ascending=False)[:10]
```

Out[77]:

	title	views	main_speaker	published_date	confusing
1346	Your body language may shape who you are	43155405	Amy Cuddy	01-10-2012	11111

201	My stroke of insight	21190883	Jill Bolte Taylor	12-03-2008	10464
1163	The power of introverts	17629275	Susan Cain	03-03-2012	10218
246	The transformative power of classical music	9315483	Benjamin Zander	25-06-2008	8108
837	The power of vulnerability	31168150	Brené Brown	23-12-2010	7942
972	Building a park in the sky	704205	Robert Hammond	30-06-2011	6685
0	Do schools kill creativity?	47227110	Ken Robinson	27-06-2006	6073
919	3 things I learned while my plane crashed	6636475	Ric Elias	22-04-2011	5834
176	Underwater astonishments	13926113	David Gallo	11-01-2008	5201
1776	How to speak so that people want to listen	21594632	Julian Treasure	27-06-2014	5167

Related Videos

In [80]:
```
df['related_talks'] = df['related_talks'].apply(lambda x: ast.literal_eval(x))
```

In [82]:
```
s = df.apply(lambda x: pd.Series(x['related_talks']), axis=1).stack().reset_index(level=1, drop= True)
s.name = 'related'
```

In [83]:
```
related_df = df.drop('related_talks', axis = 1).join(s)
related_df['related'] = related_df['related'].apply(lambda x: x['title'])
```

In [85]:
```
d = dict(related_df['title'].drop_duplicates())
d = {v: k for k, v in d.items()}
```

In [86]:
```
related_df['title'] = related_df['title'].apply(lambda x: d[x])
related_df['related'] = related_df['related'].apply(lambda x: d[x])
```

In [87]:
```
related_df = related_df[['title', 'related']]
related_df.head()
```

Out[87]:

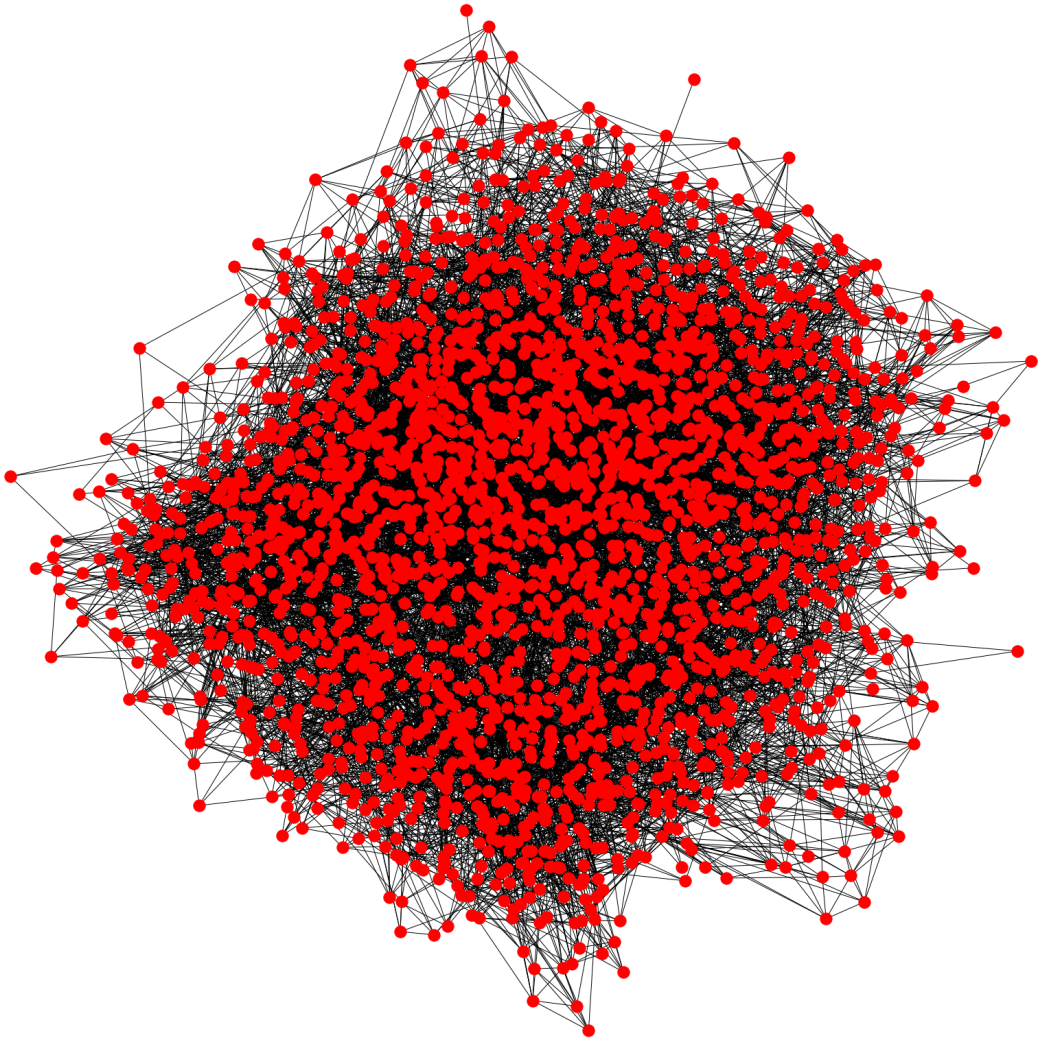
	title	related
0	0	692
0	0	1502
0	0	1991
0	0	715

0	0	1023

In [88]:
```
edges = list(zip(related_df['title'], related_df['related']))
```

In [89]:
```
import networkx as nx
G = nx.Graph()
G.add_edges_from(edges)
```

In [90]:
```
plt.figure(figsize=(25,25))
nx.draw(G, with_labels=False)
```



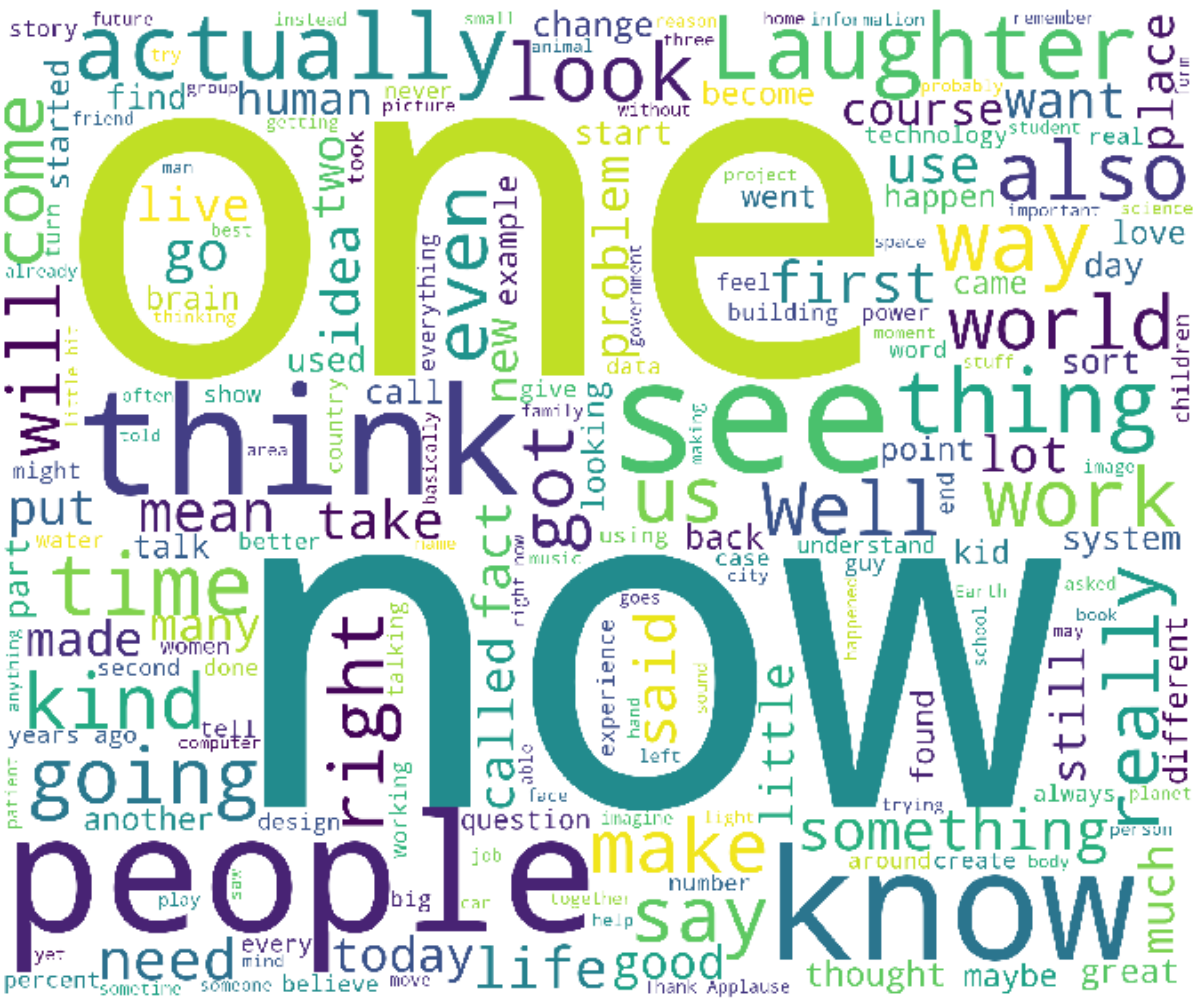
The graph is reasonably dense with every node connected to at least 3 other nodes. This shows us the real beauty of the conference. No matter how different the subjects of the talks are, the common theme of spreading ideas and inspiring people seems to be a adhesive force between them.

The TED Word Cloud

In [91]:
```
corpus = ' '.join(df2['transcript'])
corpus = corpus.replace('.',', ')
```

In [97]:
```
wordcloud = WordCloud(stopwords = STOPWORDS, background_color = 'white', w
idth = 2400,
                      height= 2000).generate(corpus)
plt.figure(figsize=(12,15))
plt.imshow(wordcloud)
plt.axis('off')
```

Out[97]: (-0.5, 2399.5, 1999.5, -0.5)



The word **One** is the most popular across the corpus of all TED Transcripts which I think encapsulates the idea of TED pretty well. **Now, Think, See, People, Laughter** and **Now** are among the most popular words used in TED speeches. TED speeches seem to play a lot emphasis on knowledge, insight, the present and of course, the people.