**GITHUB Report**

# Starting question

The importance of a national conversation is a cruicial aspect for nation building. However, there is a the lack of reﬁned and summarised data to support this theory is a problem. Based on the news topics of the last decade, which aspects of Australian culture are in need of a conversation?

*Which additional questions might give us insight to the topic development of the last decade?*

***How do we categorize the news topics?***

***How does the result actually contribute to the possibility of national conversation?***

* 1. **Relevant data**

In [6]:

pip install textblob

Requirement already satisfied: textblob in /srv/conda/envs/notebook/ lib/python3.7/site-packages (0.15.3)

Requirement already satisfied: nltk>=3.1 in /srv/conda/envs/noteboo k/lib/python3.7/site-packages (from textblob) (3.5)

Requirement already satisfied: joblib in /srv/conda/envs/notebook/li b/python3.7/site-packages (from nltk>=3.1->textblob) (1.0.0) Requirement already satisfied: click in /srv/conda/envs/notebook/li b/python3.7/site-packages (from nltk>=3.1->textblob) (7.1.2) Requirement already satisfied: tqdm in /srv/conda/envs/notebook/lib/ python3.7/site-packages (from nltk>=3.1->textblob) (4.56.0) Requirement already satisfied: regex in /srv/conda/envs/notebook/li b/python3.7/site-packages (from nltk>=3.1->textblob) (2020.11.13) Note: you may need to restart the kernel to use updated packages.

**import pandas as pd**

**import matplotlib.pyplot import numpy as np**

**import pandas as pd**

**from textblob import** TextBlob **import nltk** nltk.download('punkt') nltk.download('stopwords')

**import os**

**for** dirname, \_, filenames **in** os.walk('/kaggle/input'):

**for** filename **in** filenames: print(os.path.join(dirname, filename))

[nltk\_data] Downloading package punkt to /home/jovyan/nltk\_data... [nltk\_data] Package punkt is already up-to-date!

[nltk\_data] Downloading package stopwords to /home/jovyan/nltk\_dat a...

[nltk\_data] Package stopwords is already up-to-date!

In [8]:

data3 = pd.read\_csv('abcnews-date-text.csv')

In [9]:

data2 = pd.read\_csv('abcnews-date-text.csv',parse\_dates=[0], infer\_datetime\_form at=**True**)

data2.columns = ['date','text']

In [10]:

data3['publish\_year'] = data3['publish\_date'].apply(**lambda** x:int(x/9992))

* 1. **Analysing the data**

In [11]:

data3.shape

Out[11]: (1186018, 3)

data3.shape[0]

Out[12]:

1186018

In [13]:

print("There are **{}** news headlines represented in the data".format(data3.shape[0

]))

There are 1186018 news headlines represented in the data

*How can we address our original question: Which words are common among the headline texts? Maybe it would help if we could view the number of topics for each day?*

In [14]:

daily\_topics = data2.groupby('date')['text'].count() daily\_topics

|  |  |
| --- | --- |
| Out[14]:  date |  |
| 2003-02-19 | 198 |
| 2003-02-20 | 250 |
| 2003-02-21 | 250 |
| 2003-02-22 | 126 |
| 2003-02-23 | 136 |
| 2003-02-24 | 250 |
| 2003-02-25 | 250 |
| 2003-02-26 | 250 |
| 2003-02-27 | 221 |
| 2003-02-28 | 249 |
| 2003-03-01 | 176 |
| 2003-03-02 | 168 |
| 2003-03-03 | 232 |
| 2003-03-04 | 215 |
| 2003-03-05 | 239 |
| 2003-03-06 | 214 |
| 2003-03-07 | 209 |
| 2003-03-08 | 124 |
| 2003-03-09 | 164 |
| 2003-03-10 | 217 |
| 2003-03-11 | 220 |
| 2003-03-12 | 226 |
| 2003-03-13 | 224 |
| 2003-03-14 | 229 |
| 2003-03-15 | 134 |
| 2003-03-16 | 119 |
| 2003-03-17 | 226 |
| 2003-03-18 | 226 |
| 2003-03-19 | 225 |
| 2003-03-20 | 219 |
|  | ... |
| 2019-12-02 | 92 |
| 2019-12-03 | 115 |
| 2019-12-04 | 123 |
| 2019-12-05 | 121 |
| 2019-12-06 | 115 |
| 2019-12-07 | 62 |
| 2019-12-08 | 61 |
| 2019-12-09 | 108 |
| 2019-12-10 | 98 |
| 2019-12-11 | 100 |
| 2019-12-12 | 103 |
| 2019-12-13 | 103 |
| 2019-12-14 | 64 |
| 2019-12-15 | 48 |
| 2019-12-16 | 93 |
| 2019-12-17 | 91 |
| 2019-12-18 | 100 |
| 2019-12-19 | 109 |
| 2019-12-20 | 106 |
| 2019-12-21 | 67 |
| 2019-12-22 | 48 |
| 2019-12-23 | 66 |
| 2019-12-24 | 69 |
| 2019-12-25 | 27 |
| 2019-12-26 | 46 |
| 2019-12-27 | 52 |
| 2019-12-28 | 60 |

|  |  |
| --- | --- |
| 2019-12-29 | 43 |
| 2019-12-30 | 46 |
| 2019-12-31 | 71 |
| Name: text, | Length: 6152, dtype: int64 |
| In [15]: |  |

reindexed\_data = data2['text'] reindexed\_data.index = data2['date'] reindexed\_data.head()

Out[15]:

date

2003-02-19 aba decides against community broadcasting lic... 2003-02-19 act fire witnesses must be aware of defamation 2003-02-19 a g calls for infrastructure protection summit 2003-02-19 air nz staff in aust strike for pay rise 2003-02-19 air nz strike to affect australian travellers Name: text, dtype: object

### Or better yet, let's ﬁnd out the total amount of common words among the headlines

In [16]:

corp = str()

**for** i **in** range(len(data3['headline\_text'])): corp += (' ')+data3['headline\_text'][i]

In [17]:

**import nltk**

words = nltk.word\_tokenize(corp)

In [18]:

**from nltk.corpus import** stopwords stop\_words = set(stopwords.words('english'))

f\_words = [w **for** w **in** words **if not** w **in** stop\_words]

punctuations = '''!()-[]**{}**;:'"\,<>./?@#$%^&\*\_~''' fp\_words = [w **for** w **in** f\_words **if not** w **in** punctuations]

fd = nltk.FreqDist(fp\_words)

df\_fdist = pd.DataFrame.from\_dict(fd, orient='index') df\_fdist.columns = ['Frequency']

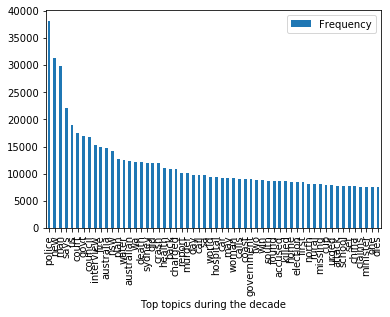
df\_fdist.index.name = 'Top topics during the decade'

freq\_df = df\_fdist[df\_fdist['Frequency']>500] d = freq\_df.to\_dict()['Frequency']

freq\_df1 = df\_fdist[df\_fdist['Frequency']>7500] freq\_df1.sort\_values('Frequency',ascending=**False**).plot(kind='bar')

Out[19]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f62715e1890>



### Now we could see which words are common among the headline texts

* 1. **Visualising the results**

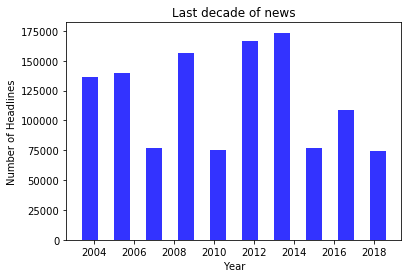
In [20]:

data3['publish\_year'] = data3['publish\_date'].apply(**lambda** x:int(x/10000))

**import matplotlib.pyplot as plt**

plt.hist(data3['publish\_year'], facecolor='blue', alpha=0.8, rwidth = 0.5) plt.xlabel('Year')

plt.ylabel('Number of Headlines') plt.title('Last decade of news') plt.show()



### Above is the amount of news topics generated during 2004 to 2019

In [22]:

pip install wordcloud

Requirement already satisfied: wordcloud in /srv/conda/envs/noteboo k/lib/python3.7/site-packages (1.8.1)

Requirement already satisfied: numpy>=1.6.1 in /srv/conda/envs/noteb ook/lib/python3.7/site-packages (from wordcloud) (1.19.5) Requirement already satisfied: matplotlib in /srv/conda/envs/noteboo k/lib/python3.7/site-packages (from wordcloud) (3.1.3)

Requirement already satisfied: pillow in /srv/conda/envs/notebook/li b/python3.7/site-packages (from wordcloud) (8.1.0)

Requirement already satisfied: kiwisolver>=1.0.1 in /srv/conda/envs/ notebook/lib/python3.7/site-packages (from matplotlib->wordcloud) (1.3.1)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.

0.1 in /srv/conda/envs/notebook/lib/python3.7/site-packages (from ma tplotlib->wordcloud) (2.4.7)

Requirement already satisfied: cycler>=0.10 in /srv/conda/envs/noteb ook/lib/python3.7/site-packages (from matplotlib->wordcloud) (0.10. 0)

Requirement already satisfied: python-dateutil>=2.1 in /srv/conda/en vs/notebook/lib/python3.7/site-packages (from matplotlib->wordcloud) (2.8.1)

Requirement already satisfied: six in /srv/conda/envs/notebook/lib/p ython3.7/site-packages (from cycler>=0.10->matplotlib->wordcloud) (1.15.0)

Note: you may need to restart the kernel to use updated packages.

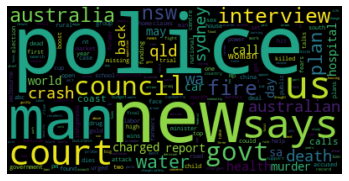
**from wordcloud import** WordCloud **import matplotlib.pyplot as plt** plt.figure(figsize=(40,40)) wordcloud = WordCloud()

wordcloud.generate\_from\_frequencies(frequencies=d) plt.figure()

plt.imshow(wordcloud, interpolation="bilinear") plt.axis("off")

plt.show()

<Figure size 2880x2880 with 0 Axes>



### The wordcloud shows the most popular "words" used during the last decade and beyond

*How do we use categorize this data into groups? Particularly into news topics?*

## 1.5 Insight

**Let's ﬁnd a way to group them into news topics**

**import nltk**

Politics = [0]\*reindexed\_data.shape[0] World = [0]\*reindexed\_data.shape[0] Business = [0]\*reindexed\_data.shape[0] Sport = [0]\*reindexed\_data.shape[0] Law = [0]\*reindexed\_data.shape[0]

Emergency = [0]\*reindexed\_data.shape[0] Health = [0]\*reindexed\_data.shape[0]

**for** i **in** range(reindexed\_data.shape[0]): words = TextBlob(reindexed\_data[i]).words **for** word **in** words:

**if** word == "politics" **or** word == "government" **or** word == "govt" **or** word

== "court": Politics[i]=1

**if** word == "world" **or** word == "finance" **or** word == "real estate" **or** word

== "international": World[i]=1

**if** word == "business" **or** word == "finance": Business[i]=1

**if** word == "sport" **or** word == "rugby": Sport[i]=1

**if** word == "law" **or** word == "police" **or** word == "council" **or** word == "ch arged" **or** word == "accused": Law[i]=1

**if** word == "emergency" **or** word == "fire" **or** word == "crash" **or** word == "murder": Emergency[i]=1

**if** word == "health" **or** word == "death": Health[i]=1

keywords = pd.DataFrame({'text':reindexed\_data,

'Politics':Politics, 'World':World, 'Business':Business, 'Sport':Sport, 'Law':Law, 'Emergency':Emergency, 'Health':Health},

index=reindexed\_data.index)

In [25]:

monthly = keywords.resample('M').sum() print(monthly)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| date | Politics | World | Business | Sport | Law | Emergency | Health |
| 2003-02-28 | 99 | 27 | 11 | 2 | 159 | 103 | 51 |
| 2003-03-31 | 251 | 105 | 13 | 4 | 387 | 183 | 121 |
| 2003-04-30 | 265 | 54 | 24 | 8 | 371 | 172 | 113 |
| 2003-05-31 | 323 | 61 | 29 | 9 | 396 | 167 | 125 |
| 2003-06-30 | 320 | 73 | 25 | 12 | 441 | 150 | 84 |
| 2003-07-31 | 306 | 75 | 21 | 9 | 419 | 172 | 120 |
| 2003-08-31 | 287 | 83 | 26 | 10 | 421 | 218 | 146 |
| 2003-09-30 | 327 | 91 | 26 | 13 | 403 | 188 | 98 |
| 2003-10-31 | 362 | 126 | 36 | 31 | 488 | 185 | 129 |
| 2003-11-30 | 330 | 93 | 26 | 26 | 382 | 160 | 105 |
| 2003-12-31 | 252 | 52 | 27 | 17 | 486 | 174 | 131 |
| 2004-01-31 | 238 | 39 | 23 | 8 | 400 | 179 | 106 |
| 2004-02-29 | 254 | 28 | 36 | 7 | 423 | 184 | 114 |
| 2004-03-31 | 331 | 32 | 19 | 10 | 483 | 194 | 139 |
| 2004-04-30 | 232 | 35 | 15 | 6 | 455 | 181 | 115 |
| 2004-05-31 | 287 | 44 | 22 | 7 | 398 | 158 | 90 |
| 2004-06-30 | 301 | 40 | 22 | 8 | 436 | 175 | 106 |
| 2004-07-31 | 270 | 46 | 29 | 4 | 441 | 174 | 130 |
| 2004-08-31 | 277 | 44 | 25 | 4 | 454 | 207 | 136 |
| 2004-09-30 | 255 | 39 | 25 | 6 | 403 | 194 | 127 |
| 2004-10-31 | 221 | 43 | 23 | 7 | 407 | 210 | 132 |
| 2004-11-30 | 258 | 50 | 20 | 7 | 420 | 196 | 138 |
| 2004-12-31 | 281 | 42 | 17 | 5 | 509 | 221 | 103 |
| 2005-01-31 | 225 | 58 | 21 | 3 | 450 | 250 | 128 |
| 2005-02-28 | 290 | 46 | 31 | 13 | 461 | 188 | 113 |
| 2005-03-31 | 307 | 52 | 21 | 9 | 547 | 199 | 124 |
| 2005-04-30 | 296 | 38 | 23 | 6 | 493 | 226 | 173 |
| 2005-05-31 | 347 | 38 | 43 | 8 | 532 | 195 | 122 |
| 2005-06-30 | 312 | 56 | 23 | 5 | 444 | 160 | 121 |
| 2005-07-31 | 299 | 42 | 25 | 5 | 503 | 160 | 150 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 2017-07-31 | 126 | 70 | 33 | 25 | 204 | 115 | 72 |
| 2017-08-31 | 130 | 80 | 39 | 38 | 213 | 110 | 94 |
| 2017-09-30 | 147 | 70 | 30 | 22 | 150 | 112 | 61 |
| 2017-10-31 | 155 | 64 | 38 | 20 | 203 | 119 | 87 |
| 2017-11-30 | 133 | 71 | 32 | 24 | 150 | 108 | 55 |
| 2017-12-31 | 105 | 38 | 12 | 13 | 168 | 106 | 67 |
| 2018-01-31 | 76 | 37 | 13 | 9 | 154 | 135 | 49 |
| 2018-02-28 | 91 | 36 | 15 | 17 | 187 | 126 | 73 |
| 2018-03-31 | 103 | 77 | 28 | 9 | 164 | 116 | 89 |
| 2018-04-30 | 86 | 41 | 22 | 15 | 162 | 104 | 63 |
| 2018-05-31 | 122 | 57 | 31 | 5 | 178 | 101 | 70 |
| 2018-06-30 | 93 | 194 | 46 | 12 | 146 | 120 | 58 |
| 2018-07-31 | 96 | 136 | 54 | 10 | 194 | 107 | 92 |
| 2018-08-31 | 87 | 52 | 48 | 14 | 154 | 135 | 85 |
| 2018-09-30 | 102 | 51 | 36 | 14 | 153 | 92 | 76 |
| 2018-10-31 | 100 | 59 | 38 | 15 | 120 | 112 | 82 |
| 2018-11-30 | 77 | 76 | 35 | 8 | 171 | 124 | 78 |
| 2018-12-31 | 94 | 39 | 19 | 6 | 157 | 111 | 62 |
| 2019-01-31 | 80 | 24 | 10 | 5 | 124 | 97 | 52 |
| 2019-02-28 | 105 | 38 | 43 | 12 | 146 | 118 | 46 |
| 2019-03-31 | 77 | 60 | 39 | 9 | 138 | 101 | 57 |
| 2019-04-30 | 76 | 35 | 38 | 15 | 146 | 114 | 57 |
| 2019-05-31 | 77 | 57 | 38 | 18 | 144 | 82 | 63 |
| 2019-06-30 | 93 | 106 | 35 | 13 | 159 | 94 | 44 |
| 2019-07-31 | 78 | 100 | 33 | 11 | 154 | 107 | 93 |
| 2019-08-31 | 97 | 40 | 36 | 14 | 179 | 93 | 63 |
| 2019-09-30 | 80 | 68 | 32 | 23 | 133 | 136 | 48 |
| 2019-10-31 | 105 | 82 | 51 | 30 | 188 | 128 | 54 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2019-11-30 | 92 | 48 | 34 | 19 | 172 | 174 | 66 |
| 2019-12-31 | 48 | 31 | 14 | 9 | 113 | 161 | 33 |

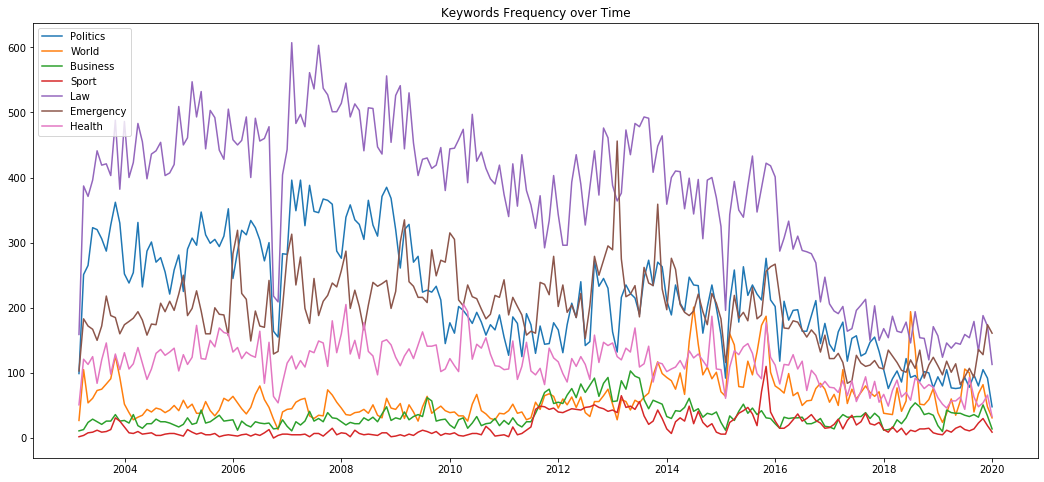
[203 rows x 7 columns]

### Now we see the words grouped into news topics

In [26]:

fig, ax = plt.subplots(figsize=(18,8))

ax.plot(monthly['Politics'], label='Politics'); ax.plot(monthly['World'], label='World'); ax.plot(monthly['Business'], label='Business'); ax.plot(monthly['Sport'], label='Sport'); ax.plot(monthly['Law'], label='Law'); ax.plot(monthly['Emergency'], label='Emergency'); ax.plot(monthly['Health'], label='Health'); ax.set\_title('Keywords Frequency over Time'); ax.legend(loc='upper left');



# Key points from 2010 to 2019 :

News topics on "Law" were the most popular out of all the groups (Keep in mind that keywords for Law include "police", "council", etc.)

These were primarily keywords that focused on legislation or the process of it (et. al "police")

During early 2013, "Emergency" was trending so much that it was almost able to take the top spot for that time period

It could be noted that news on the Victoria bushﬁres were sought after at this point

Topics on "Politics" were surprisingly trending down from 2009 to 2019

It was overtaken numerous times more urgent topics from the "Emergency" section

World news had a spike during early 2016 and 2019, which can be attributed to the US elections (Donald Trump) & China-USA tensions

Topics on "Sports" were the least popular in the last decade

***Based on the presented data above, it is safe to assume that Australia's national conversation were predominantly dominated by topics on legislative matters ("Laws"). It comes to no surprise that matters of the "council" or "police" directly aﬀects the economic status and well-being of the average Aussie family. When push comes to shove, more urgent topics such as "bushﬁres" will come into the limelight as news agencies feel that this is more important. It is worth noting that "Emergency" topics also focus on attention grabbing headlines such as "murder". We may question the media's intention for this: Was the topic made for viewership's sake or from a geniune source of concern? The data also represents the citizen's concern for global topics as they are well aware of international news.***

* 1. **Second question**

According to the WHO, mental health is about the state of a peron's wellnes rather than an illness. There is a diﬀerence between "mental health" and a mental condition. However, there is no oﬃcial international guideline on how to eﬀectively tackle mental health issues. Thus, each country uniquely addresses the problems faced by its citizen thru their own wellness programs. Which leads us to the question:

**How does the frequency of mental health illness and attitudes towards mental health vary by geographic location?**

**What are the health eﬀects of its frequency?**

* 1. **Relevant data**

In [27]:

**import numpy as np import pandas as pd**

**import matplotlib.pyplot as plt import seaborn as sns**

**from collections import** Counter

%**matplotlib** inline

**from subprocess import** check\_output

print(check\_output(["ls", "mentalhealthsurvey.csv"]).decode("utf8"))

mentalhealthsurvey.csv

In [28]:

df = pd.read\_csv('mentalhealthsurvey.csv') df2 = pd.read\_csv('master.csv')

* 1. **Analysing the data**

Let's check the data ﬁrst

df.head()

Out[29]:

**Timestamp Age Gender Country state self\_employed family\_history treatment work\_i**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2014-08-27  11:29:31 | 37 | Female | United States | IL | NaN | No | Yes |
| **1** | 2014-08-27  11:29:37 | 44 | M | United States | IN | NaN | No | No |
| **2** | 2014-08-27  11:29:44 | 32 | Male | Canada | NaN | NaN | No | No |
| **3** | 2014-08-27  11:29:46 | 31 | Male | United Kingdom | NaN | NaN | Yes | Yes |
| **4** | 2014-08-27  11:30:22 | 31 | Male | United States | TX | NaN | No | No |

5 rows × 27 columns

In [30]:

df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 1259 entries, 0 to 1258 Data columns (total 27 columns):

Timestamp 1259 non-null object

Age 1259 non-null int64

Gender 1259 non-null object

Country 1259 non-null object

state 744 non-null object

self\_employed 1241 non-null object

family\_history 1259 non-null object

treatment 1259 non-null object

work\_interfere 995 non-null object

|  |  |  |  |
| --- | --- | --- | --- |
| no\_employees | 1259 | non-null | object |
| remote\_work | 1259 | non-null | object |
| tech\_company | 1259 | non-null | object |
| benefits | 1259 | non-null | object |
| care\_options | 1259 | non-null | object |
| wellness\_program | 1259 | non-null | object |
| seek\_help | 1259 | non-null | object |
| anonymity | 1259 | non-null | object |
| leave | 1259 | non-null | object |
| mental\_health\_consequence | 1259 | non-null | object |
| phys\_health\_consequence | 1259 | non-null | object |
| coworkers | 1259 | non-null | object |
| supervisor | 1259 | non-null | object |
| mental\_health\_interview | 1259 | non-null | object |
| phys\_health\_interview | 1259 | non-null | object |
| mental\_vs\_physical | 1259 | non-null | object |
| obs\_consequence | 1259 | non-null | object |

comments 164 non-null object dtypes: int64(1), object(26)

memory usage: 265.6+ KB

In [32]:

df['Age'] = pd.to\_numeric(df['Age'],errors='coerce')

**def** age\_process(age):

**if** age>=0 **and** age<=100:

**return** age

**else**:

**return** np.nan

df['Age'] = df['Age'].apply(age\_process)

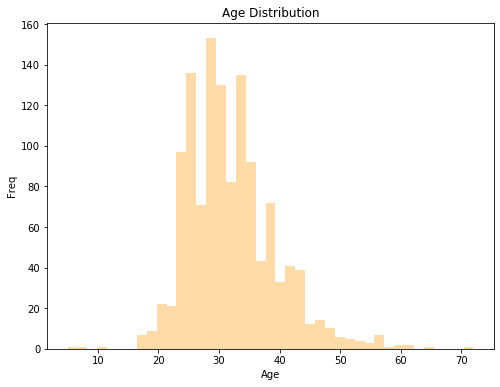
In [33]:

fig,ax = plt.subplots(figsize=(8,6)) sns.distplot(df['Age'].dropna(),ax=ax,kde=**False**,color='#ffa726') plt.title('Age Distribution')

plt.ylabel('Freq')

Out[33]:

Text(0, 0.5, 'Freq')



df['Timestamp'] = pd.to\_datetime(df['Timestamp'],format='%Y-%m-%d') #convert into format df['Year'] = df['Timestamp'].apply(lambda x:x.year)#### Above is the age distribution of mental health responses

Most responses were from workers aged around their early 20's to mid 30's With the most responses from people aged around 26 - 29

Worth noticing that there were responses from people below 10 years old, this could be a discrepency

df['Timestamp'] = pd.to\_datetime(df['Timestamp'],format='%Y-%m-**%d**') df['Year'] = df['Timestamp'].apply(**lambda** x:x.year)

In [35]:

df['Age\_Group'] = pd.cut(df['Age'].dropna(),

[0,18,25,35,45,99],

labels=['<18','18-24','25-34','35-44','45+'])

### Do workers have a family history of mental illness?

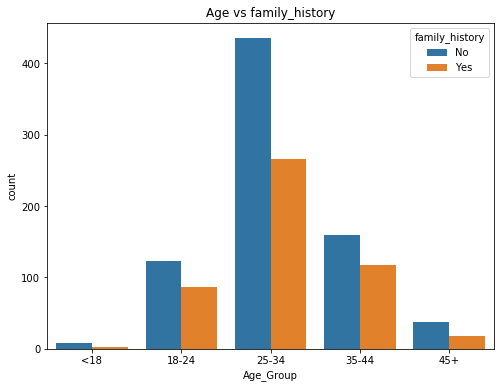
In [36]:

fig,ax = plt.subplots(figsize=(8,6))

sns.countplot(data=df,x = 'Age\_Group',hue= 'family\_history',ax=ax) plt.title('Age vs family\_history')

Out[36]:

Text(0.5, 1.0, 'Age vs family\_history')



### It can be seen that there is a relationship between a person's family history with their mental ilness.

**1.4 Visualising the results**

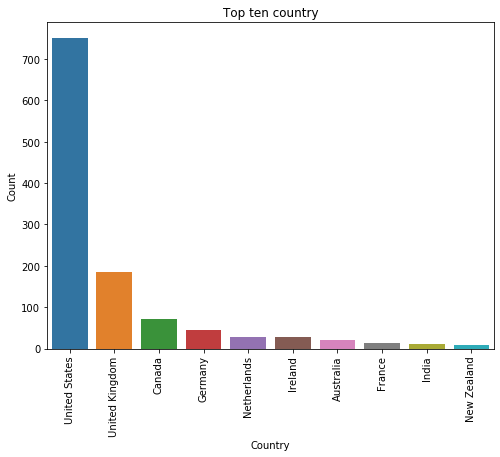
country\_count = Counter(df['Country'].dropna().tolist()).most\_common(10) country\_idx = [country[0] **for** country **in** country\_count]

country\_val = [country[1] **for** country **in** country\_count] fig,ax = plt.subplots(figsize=(8,6))

sns.barplot(x = country\_idx,y=country\_val ,ax =ax) plt.title('Top ten country') plt.xlabel('Country')

plt.ylabel('Count')

ticks = plt.setp(ax.get\_xticklabels(),rotation=90)

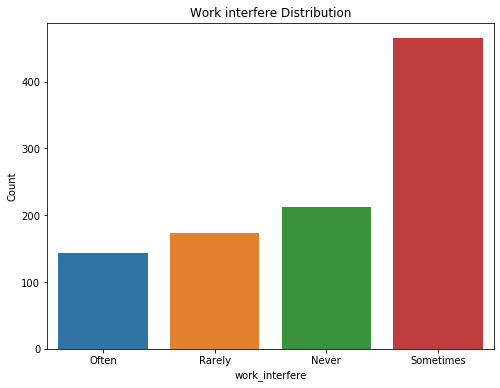


**The graph above shows the top 10 countries with the most recorded number of mental illnesses. USA comes into a distant lead over the next country (UK).**

fig,ax =plt.subplots(figsize=(8,6)) sns.countplot(df['work\_interfere'].dropna(),ax=ax) plt.title('Work interfere Distribution') plt.ylabel('Count')

Out[38]:

Text(0, 0.5, 'Count')



### Above is the frequency of mental health issue interferences for workers

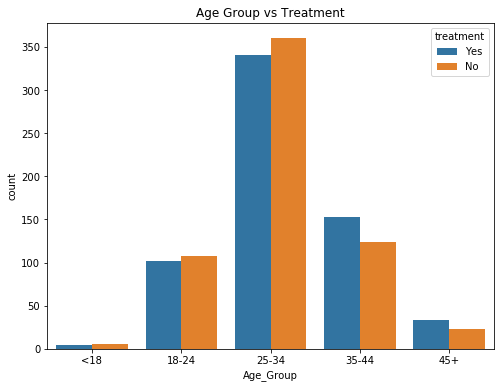
* 1. **Insight**

fig,ax =plt.subplots(figsize=(8,6))

sns.countplot(data = df,x = 'Age\_Group', hue='treatment') plt.title('Age Group vs Treatment')

Out[39]:

Text(0.5, 1.0, 'Age Group vs Treatment')



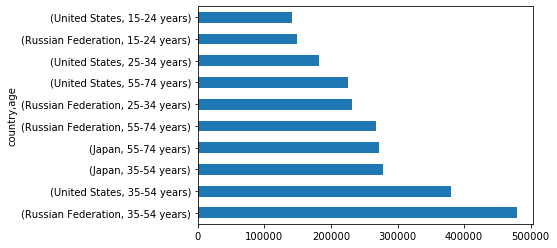
## Less people diagnosed with mental health problems are seeking treatment. This could possibly be either attributed to the fact that there is either a lack of support or society's stigma as a whole.

In [40]:

df2.groupby(['country','age']).suicides\_no.sum().nlargest(10).plot(kind='barh')

Out[40]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f626d189150>



### Above is a chart derived from WHO data. It explains the frequency of diagnosed mental illnesses from developed countries and its suicide rate per million persons.

**Key points from the WHO survey analysis:**

* People with a history of family mental health issues are more likely to suffer the same diagnosis than those who don't have a record.
* Workers from the early 20's to early 30's seem to be the most vulnerable to developing a mental illness
* USA and UK had the most reported cases while other countries were laggin g
* The number of people seeking treatment was less than those who did not
* Majority of mental health related issues only "sometimes" interfered wit h work
* Based on suicide rates, the countries of Japan, Russia and USA were prom inent on the list at different age groups

# Third question

The topic of mental health has lately been trending as an area of interest among the scientiﬁc community, particularly its causes and eﬀects within the workspace. Although novel, we a researchers within the tech world would like to know what are the common atittudes towards mental illnesses and what could be the possible predictors for it?

### Based on the survey, what are the atittudes and predictors for mental illness within the tech space?

In [41]:

In [42]:

**import numpy as np import pandas as pd**

**import random as rnd**

**import seaborn as sns**

sns.set\_palette('Set2')

**import matplotlib.pyplot as plt**

%**matplotlib** inline

In [43]:

data = pd.read\_csv('mental-heath-in-tech-2016\_20161114.csv')

* 1. **Analyze data**

In [44]:

data.describe()

Out[44]:

**Are you**

**self- employed?**

**Is your employer primarily a tech company/organization?**

**Is your primary role within**

**your company related to tech/IT?**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | **treatment of mental**  **health issues?** |  | **professiona** |
| **count** | 1433.000000 | 1146.000000 | 263.000000 | 287.000000 | 1433.000000 | 1433.00000 |
| **mean** | 0.200279 | 0.770506 | 0.942966 | 0.644599 | 0.882066 | 0.58548 |
| **std** | 0.400349 | 0.420691 | 0.232350 | 0.479471 | 0.322643 | 0.49281 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.00000 |
| **25%** | 0.000000 | 1.000000 | 1.000000 | 0.000000 | 1.000000 | 0.00000 |
| **50%** | 0.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.00000 |
| **75%** | 0.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.00000 |
| **max** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.00000 |

**Do you have medical coverage (private insurance or state- provided)**

**which includes**

**Do you have**

**previous employers?**

**Have yo ever soug**

**treatment f**

**a ment health issu**

**from mental healt**

data.head()

Out[45]:

**Are you**

**self- employed?**

**How many employees does your company or organization**

**have?**

**Is your employer primarily a tech company/organization?**

**Is your primary**

**role within your company related**

**to tech/IT?**

**Does your employer provide mental health beneﬁts as part of healthcare coverage?**

**Do you know the options for mental**

**health care available**

**under your employer- provided coverage?**

**empl**

**men (for ex**

**ca oth**

**commu**

**0** 0 26-100 1.0 NaN

Not eligible

for coverage /

N/A

NaN

* + 1. 0 6-25 1.0 NaN No Yes
    2. 0 6-25 1.0 NaN No NaN
    3. 1 NaN NaN NaN NaN NaN

**4** 0 6-25 0.0 1.0 Yes Yes

5 rows × 63 columns

In [46]:

data.loc[(data['What is your age?'] > 90), 'What is your age?'] = 34 data.loc[(data['What is your age?'] < 10), 'What is your age?'] = 34

In [47]:

data['What is your gender?'] = data['What is your gender?'].replace([ 'male', 'Male ', 'M', 'm', 'man', 'Cis male',

'Male.', 'Male (cis)', 'Man', 'Sex is male',

'cis male', 'Malr', 'Dude', "I'm a man why didn't you make this a drop down question. You should of asked sex? And I would of answered yes please. Seriousl

y how much text can this take? ",

'mail', 'M|', 'male ', 'Cis Male', 'Male (trans, FtM)',

'cisdude', 'cis man', 'MALE'], 'Male')

data['What is your gender?'] = data['What is your gender?'].replace([ 'female', 'I identify as female.', 'female ',

'Female assigned at birth ', 'F', 'Woman', 'fm', 'f',

'Cis female', 'Transitioned, M2F', 'Female or Multi-Gender Femme', 'Female ', 'woman', 'female/woman', 'Cisgender Female',

'mtf', 'fem', 'Female (props for making this a freeform field, though)', ' Female', 'Cis-woman', 'AFAB', 'Transgender woman',

'Cis female '], 'Female')

data['What is your gender?'] = data['What is your gender?'].replace([ 'Bigender', 'non-binary,', 'Genderfluid (born female)', 'Other/Transfeminine', 'Androgynous', 'male 9:1 female, roughly', 'nb masculine', 'genderqueer', 'Human', 'Genderfluid',

'Enby', 'genderqueer woman', 'Queer', 'Agender', 'Fluid', 'Genderflux demi-girl', 'female-bodied; no feelings about gender',

'non-binary', 'Male/genderqueer', 'Nonbinary', 'Other', 'none of your busine ss',

'Unicorn', 'human', 'Genderqueer'], 'Genderqueer/Other')

data['What is your gender?'] = data['What is your gender?'].replace(np.NaN, 'Mal e')

data['What is your gender?'].unique()

Out[48]:

array(['Male', 'Female', 'Genderqueer/Other'], dtype=object)

In [49]:

data.drop(['Why or why not?', 'Why or why not?.1'], axis=1, inplace=**True**)

## Visualization

### Show common mental disorders

In [50]:

disorders = {}

disorderCounts = dict(data['If so, what condition(s) were you diagnosed with?']. value\_counts())

**for** i **in** disorderCounts:

disorderList = i.split('|')

**for** j **in** disorderList: j = j.split(' (')[0]

disorders[j] = disorders.get(j, 0) + disorderCounts[i]

tmp = pd.DataFrame()

**for** i **in** disorders:

tmp = tmp.append([i] \* disorders[i])

tmp[0] = tmp[0].replace([

'Autism Spectrum Disorder', 'Autism - while not a "mental illness", still gr eatly affects how I handle anxiety',

'autism spectrum disorder', 'PDD-NOS'], 'Autism')

tmp[0] = tmp[0].replace(['Aspergers', 'Asperger Syndrome'], "Asperger's Syndrom e")

tmp[0] = tmp[0].replace(['posttraumatic stress disourder'], 'Post-traumatic Stre ss Disorder')

tmp[0] = tmp[0].replace(['ADD', 'Attention Deficit Disorder', 'attention deficit disorder'],

'Attention Deficit Hyperactivity Disorder')

tmp[0] = tmp[0].replace(['Schizotypal Personality Disorder'], 'Personality Disor der')

tmp[0] = tmp[0].replace(['Depression'], 'Mood Disorder') tmp[0] = tmp[0].replace([

'Autism', "Asperger's Syndrome", 'Intimate Disorder',

'Seasonal Affective Disorder', 'Burn out', 'Gender Identity Disorder', 'Suicidal Ideation', 'Gender Dysphoria', 'MCD'], 'Others')

g = sns.countplot(y=tmp[0], order=[

'Mood Disorder', 'Anxiety Disorder', 'Attention Deficit Hyperactivity Disord er',

'Post-traumatic Stress Disorder', 'Obsessive-Compulsive Disorder',

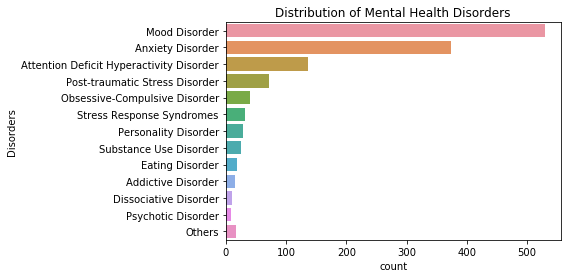
'Stress Response Syndromes', 'Personality Disorder', 'Substance Use Disorde

r',

'Eating Disorder', 'Addictive Disorder', 'Dissociative Disorder', 'Psychotic Disorder', 'Others'])

g.set\_ylabel('Disorders')

g.set\_title('Distribution of Mental Health Disorders') plt.show()



### Above we can see that the most common mental health problems in the tech space are mood and anxiety disorders

**Employee job roles**

In [51]:

labels = ['Self-employed', 'Works at tech company', 'Has a tech role in non-tech company', 'Has a non-tech role at a non-tech company']

sizes = [data['Are you self-employed?'].value\_counts()[1],

data['Is your employer primarily a tech company/organization?'].value\_c ounts()[1],

data['Is your primary role within your company related to tech/IT?'].va lue\_counts()[1],

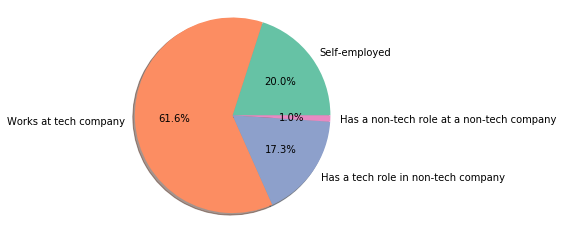
data['Is your primary role within your company related to tech/IT?'].va lue\_counts()[0]

]

fig1, ax1 = plt.subplots()

ax1.pie(sizes, labels=labels, autopct='**%1.1f%%**', shadow=**True**) ax1.axis('equal')

plt.show()



### A lot of the respodents work for a tech company, while the rest are either self-employed or has a tech role in a non-tech company

**Mental health among age and gender groups**

In [52]:

data.rename(columns={'What is your age?': 'Age',

'What is your gender?': 'Gender',

'Do you currently have a mental health disorder?': 'Has MH

D'}, inplace=**True**)

In [53]:

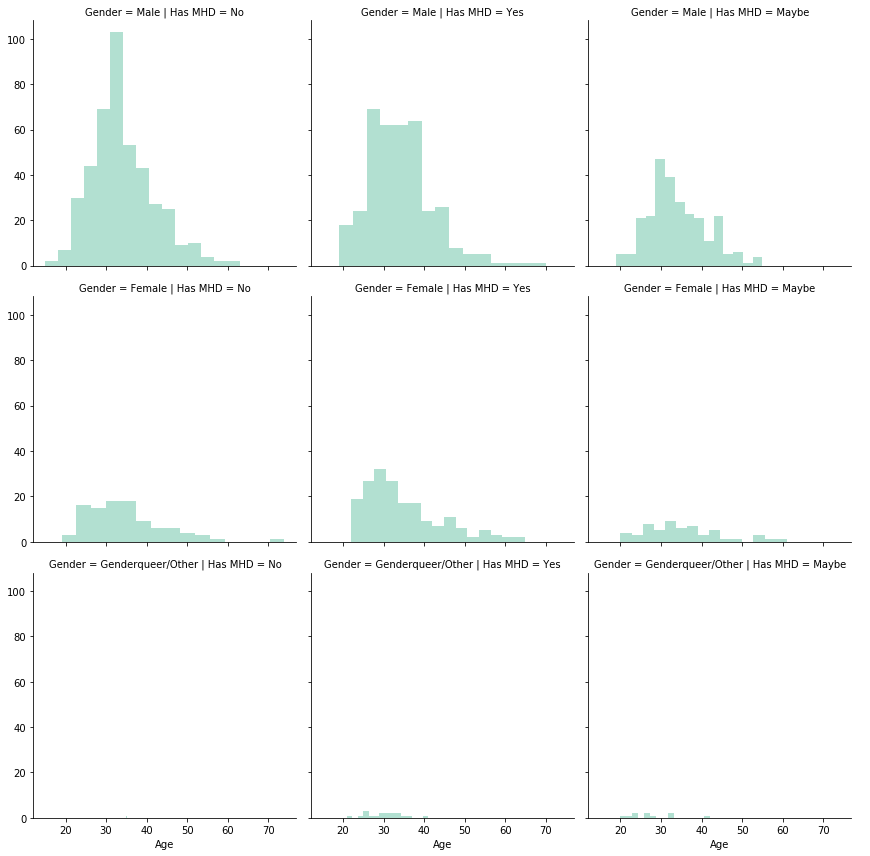
g = sns.FacetGrid(data, row='Gender', col='Has MHD', size=4) g.map(plt.hist, 'Age', alpha=0.5, bins=15)

g.add\_legend()

/srv/conda/envs/notebook/lib/python3.7/site-packages/seaborn/axisgri d.py:230: UserWarning: The `size` paramter has been renamed to `heig ht`; please update your code.

warnings.warn(msg, UserWarning) Out[53]:

<seaborn.axisgrid.FacetGrid at 0x7f625ed73250>



### From the charts above, we could see that male respondents aged in their mid 20s to early 30's are more prone to some sort of mental ilness

**Attitude: Willingness to Bring Up Health Issues in an Interview: Physical vs. Mental**

In [54]:

fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=**True**)

fig.suptitle('Responses to "Would you bring up a health issue**\n**with a potential employer in an interview?"')

g1 = sns.countplot(x='Would you be willing to bring up a physical health issue w ith a potential employer in an interview?',

hue='Gender', data=data, ax=ax1, order=['Yes', 'No', 'Maybe'])

g2 = sns.countplot(x='Would you bring up a mental health issue with a potential employer in an interview?',

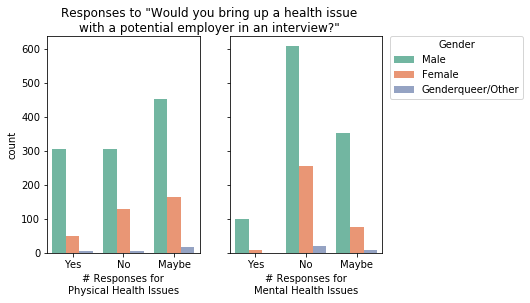
hue='Gender', data=data, ax=ax2, order=['Yes', 'No', 'Maybe'])

g1.legend\_.remove()

plt.legend(bbox\_to\_anchor=(1.05, 1), loc=2, borderaxespad=0., title='Gender') g1.set\_xlabel('# Responses for**\n**Physical Health Issues')

g2.set\_xlabel('# Responses for**\n**Mental Health Issues') g2.set\_ylabel('')

plt.show()



*Based on the survey, less males in the tech world are more willing to bring up mental health issues than physical ones.*

### Attitude: Is Having Mental Health Issues is Harmful to One's Career?

In [55]:

g = sns.countplot(y='Do you feel that being identified as a person with a mental health issue would hurt your career?',

hue='Gender', data=data,

order=['Yes, it has', 'Yes, I think it would',

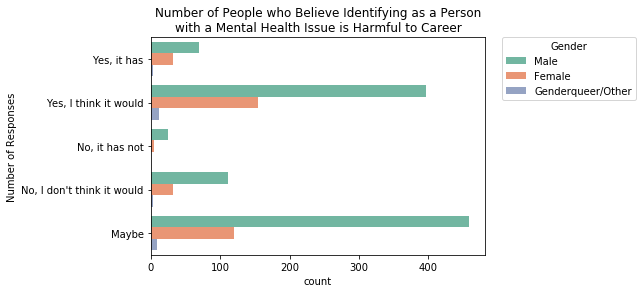
'No, it has not', "No, I don't think it would", 'Maybe'

])

plt.title('Number of People who Believe Identifying as a Person**\n**with a Mental H ealth Issue is Harmful to Career')

plt.legend(bbox\_to\_anchor=(1.05, 1), loc=2, borderaxespad=0., title='Gender') plt.ylabel('Number of Responses')

plt.show()



In [56]:

\*There **is** more stigma **for** the male respondents **as** a lot of them either think men tal health issues are harmful to one's career\*

File "<ipython-input-56-d9dd37497b93>", line 1

\*There is more stigma for the male respondents as a lot of them either think mental health issues are harmful to one's career\*

^

SyntaxError: invalid syntax

### Attitude: Potential Negative Consequences for Discussing Health Issues with Employer: Physical vs. Mental

In [ ]:

fig, (ax1, ax2) = plt.subplots(ncols=2, sharey=**True**)

fig.suptitle('Responses to "Do you think that discussing a health issue**\n**with yo ur employer would have negative consequences?"')

g1 = sns.countplot(x='Do you think that discussing a physical health issue with your employer would have negative consequences?',

hue='Gender', data=data, ax=ax1, order=['Yes', 'No', 'Maybe'])

g2 = sns.countplot(x='Do you think that discussing a mental health disorder with your employer would have negative consequences?',

hue='Gender', data=data, ax=ax2, order=['Yes', 'No', 'Maybe'])

g1.legend\_.remove()

plt.legend(bbox\_to\_anchor=(1.05, 1), loc=2, borderaxespad=0., title='Gender') g1.set\_xlabel('# Responses for**\n**Physical Health Issues')

g2.set\_xlabel('# Responses for**\n**Mental Health Issues') g2.set\_ylabel('')

plt.show()

*From the two graphs, we could see that: having disucussions with employers on physical health issues will have less consequences than mental health topics*

## Insights

**Key points from the Tech survey analysis:**

* + - Mood and anxiety disorders were the most common forms of reported illnes es, followed by ADHD, PTSD, etc.
    - Most repondents came from a tech company or at least serving traditional a tech role in a non-tech company
    - Indicators point that adult males (age 27 - 34) were the most vulnerable out of all the age groups:
      * They had an attitude/belief that mental issues were harmful to one's career
      * Discussion with employers on physical health topics had less negativ e consequences than mental health topics
      * They were less willing to bring up mental health issues in an employ ee interview

### References:

[Szamil. (2018, August 29). WHO Suicide Statistics. Retrieved from https://www.kaggle.com/szamil/who- suicide-statistics/kernels (https://www.kaggle.com/szamil/who-suicide-statistics/kernels)](https://www.kaggle.com/szamil/who-suicide-statistics/kernels)

*Modiﬁed codes from:*

[https://www.kaggle.com/andradaolteanu/preprocess-visualise-model-mental-health-in-tech (https://www.kaggle.com/andradaolteanu/preprocess-visualise-model-mental-health-in-tech)](https://www.kaggle.com/andradaolteanu/preprocess-visualise-model-mental-health-in-tech)

[https://www.kaggle.com/jchen2186/data-visualization-with-python-seaborn (https://www.kaggle.com/jchen2186/data-visualization-with-python-seaborn)](https://www.kaggle.com/jchen2186/data-visualization-with-python-seaborn)

[https://www.kaggle.com/kairosart/machine-learning-for-mental-health-1 (https://www.kaggle.com/kairosart/machine-learning-for-mental-health-1)](https://www.kaggle.com/kairosart/machine-learning-for-mental-health-1)

[https://www.kaggle.com/lostarious/most-used-words-in-a-million-headlines (https://www.kaggle.com/lostarious/most-used-words-in-a-million-headlines)](https://www.kaggle.com/lostarious/most-used-words-in-a-million-headlines)

[https://www.kaggle.com/rcushen/topic-modelling-with-lsa-and-lda (https://www.kaggle.com/rcushen/topic- modelling-with-lsa-and-lda)](https://www.kaggle.com/rcushen/topic-modelling-with-lsa-and-lda)