Final Project: Age of Aquiring Words with Different Sentiments

Name: `Lexy Li`

SID: `3033785746`

I. Introduction

1. Age of Acquisition database

Kuperman et al. (2012) collected ratings of "Age of Acquisition" for 30,121 English words (nouns, verbs, and adjectives). Subjects were asked to consider at what age they had acquired a certain word, and these ages are represented as means ("Rating.Mean") and standard deviations (Rating.SD). This information is widely used in child language development/language acquisition research.

You can read more about it [here (http://crr.ugent.be/archives/806)](http://crr.ugent.be/archives/806) and [here (http://crr.ugent.be/papers/Kuperman%20et%20al%20AoA%20ratings.pdf)](http://crr.ugent.be/papers/Kuperman%20et%20al%20AoA%20ratings.pdf).

In [1]:
```
!pip install -q textblob
```

In [2]:
```
from datascience import *
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import string
import re
from scipy.stats import pearsonr
from scipy import stats

#install textblob in terminal
#pip install -U textblob
#python -m textblob.download_corpora
from textblob import TextBlob
```

In [3]:
```
d = Table.read_table("datasets/aoa.csv")
d.show(10)
```

Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno
a	22	22	20415.3	2.89	1.21	1
aardvark	18	18	0.41	9.89	3.66	1
abacus	20	13	0.24	8.69	3.77	0.65
abalone	18	13	0.51	12.23	3.54	0.72
abandon	19	19	8.1	8.32	2.75	1
abandoner	19	18	0.02	11.89	3.36	0.95
abandonment	22	22	0.96	10.27	2.57	1
abase	19	14	0.06	14.57	4.29	0.74
abasement	19	12	nan	15.13	5.37	0.63
abate	19	18	0.1	14.44	3.57	0.95

... (31308 rows omitted)

2. Research Question

Children laugh, cry, and express their love and hatred in straightforward, explicit ways. Growing up, I came to admire children a lot as I sometimes find it hard to love and say love. When I realize that my expression of emotions and opinions start to carry weight because it would potentially affect others or even hurt myself back, I have to think twice before I speak, hoping to best convey objective and neutral ideas. This does not only apply to me solely. When we leave childhood for adulthood, what we choose to say and what others want us to hear create a new language environment. Granted, an increasing age not only allows us to intellectually understand more complicated words, but also urges us to go beyond the most simple, emotional expressions and adapt to new social contexts.

Do we deal with more words with extreme subjectivity and polarity when we are little? Do we tend to acquire more objective and neutral words as our age increases? With this database, I would like to look into the pattern of how we acquire words with different sentiments in different stages of life.

3. Defining the sentiment of a word

In this project I will be using an imported library to help me identify word sentiments. **TextBlob** is a Python library for processing textual data. It is a great tool for sentiment analysis and more.

The sentiment function of textblob returns two properties, polarity, and subjectivity. Polarity is float which lies in the range of [-1, 1], where 1 means positive statement and -1 means a negative statement. The subjectivity is a float within the range [0.0, 1.0], where 0.0 is very objective and 1.0 is very subjective.

In [4]:
```
print("polarity")
print("beautiful: ", TextBlob("beautiful").sentiment.polarity)
print("ugly: ", TextBlob("ugly").sentiment.polarity)

print("subjectivity")
print("hate: ", TextBlob("hate").sentiment.subjectivity)
print("understand: ", TextBlob("understand").sentiment.subjectivity)
```

```
polarity
beautiful:  0.85
ugly:  -0.7
subjectivity
hate:  0.9
understand:  0.0
```

II. Data Summary

1. A closer look: choosing "Word" as our primary key

In [5]:
```
print("number of columns: ", d.num_columns)
print("number of rows: ", d.num_rows)
```

```
number of columns:  7
number of rows:  31318
```

In [6]:
```
list_of_words = np.unique(d.column('Word'))
list_of_words
print("number of unique words: ", len(list_of_words))
print("number of non-unique words: ", d.num_rows - len(list_of_words))
```

```
number of unique words:  31124
number of non-unique words:  194
```

In [7]:
```
d.group('Word').where('count', are.above(1))
```

Out[7]:

Word	count
nan	195

From the above cells we notice that there are some rows that fail to record the testing words. We will need to remove 195 rows with "nan" values, which should leave us with 31123 valid records.

2. A closer look: Statistics in other columns/variables

1) Rating.SD

In [8]:
```
stat_data = stats.describe(d.column('Rating.SD'), nan_policy = "omit")
print("mean: ", stat_data.mean)
print("variance: ", stat_data.variance)
d.where('Rating.SD', are.above(7)).column('Word')
```

```
mean:  3.1876976099980676
variance:  0.8174913778438608
```

Out[8]:
```
array(['app', 'borzoi', 'bumbershoot', 'casaba', 'chorister', 'connubial',
       'consubstantiation', 'contravention', 'culottes', 'effulgent',
       'enfilade', 'folderol', 'gawp', 'glebe', 'locoweed', 'mashie',
       'moll', 'neomycin', 'opprobrious', 'pachinko', 'pilchard',
       'quotidian', 'samovar', 'slugabed', 'terrazzo', 'twee', 'tweet',
       'tweeting', 'wold'], dtype='<U22')
```

2) Rating Mean

In [9]:
```
stat_data = stats.describe(d.column('Rating.Mean'), nan_policy = "omit")
print("mean: ", stat_data.mean)
print("variance: ", stat_data.variance)
d.where('Rating.Mean', are.above(19)).column('Word')
#d.where('Rating.Mean', are.below(3)).column('Word')
```

```
mean:  11.000036007072818
variance:  9.267008332036780
```

Out[9]:
```
array(['architrave', 'berkelium', 'calceolaria', 'cortices',
       'cytomegalovirus', 'diethylamide', 'diphthongization', 'edematous',
       'eisteddfod', 'furfural', 'guerdon', 'hyperparathyroidism',
       'kendo', 'majuscule', 'maxilla', 'methadone', 'mortmain',
       'odalisque', 'oubliette', 'pederasty', 'penury', 'psychosexual',
       'sadomasochist', 'schottische', 'scopolamine', 'streptomycin',
       'thrombocytopenia', 'velar', 'vermiform', 'vicuna', 'yakitori'],
      dtype='<U22')
```

3) Dunno

In [10]:
```
stat_data = stats.describe(d.column('Dunno'), nan_policy = "omit")
print("mean: ", stat_data.mean)
print("variance: ", stat_data.variance)
d.where('Dunno', are.below(0.5))
```

```
mean:  0.8732219509060533
variance:  0.03938896323500758
```

Out[10]:

Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno
abattoir	19	6	0.14	15.17	2.71	0.32
abbacy	21	2	nan	14.5	0.71	0.1
abbess	19	7	0.04	15.43	2.37	0.37
abeyance	20	7	0.04	15	2.58	0.35
abjuration	21	8	nan	17.12	2.59	0.38
ablation	18	7	0.02	13.29	3.45	0.39
aborning	17	4	0.02	13.25	3.77	0.24
abrade	18	8	0.02	14	4.54	0.44
abrogate	19	9	0.02	15.22	2.99	0.47
abut	21	9	0.12	13.11	1.9	0.43

... (2287 rows omitted)

III. Data manipulation

1. Data cleaning

- Removing rows that don't have their words recorded.

In [11]:
```
d = d.where('Word', are.not_equal_to('nan'))
d.num_rows
```

Out[11]: 31123

In [12]:
```
d
```

Out[12]:

Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno
a	22	22	20415.3	2.89	1.21	1
aardvark	18	18	0.41	9.89	3.66	1
abacus	20	13	0.24	8.69	3.77	0.65
abalone	18	13	0.51	12.23	3.54	0.72
abandon	19	19	8.1	8.32	2.75	1
abandoner	19	18	0.02	11.89	3.36	0.95
abandonment	22	22	0.96	10.27	2.57	1
abase	19	14	0.06	14.57	4.29	0.74
abasement	19	12	nan	15.13	5.37	0.63
abate	19	18	0.1	14.44	3.57	0.95

... (31113 rows omitted)

2. get the sentiment score of all words from TextBlob

In [13]:
```
def get_sentiment_polarity(word):
    return round(TextBlob(word).sentiment.polarity, 5)

def get_sentiment_subjectivity(word):
    return round(TextBlob(word).sentiment.subjectivity, 5)


d = d.with_columns('Polarity', d.apply(get_sentiment_polarity, 'Word'))
d = d.with_columns('Polarity Mag', d.apply(abs, 'Polarity'))
d = d.with_columns('Subjectivity', d.apply(get_sentiment_subjectivity, 'Wor
d'))
d
```

Out[13]:

Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno	Polarity	Polarity Mag	Su
a	22	22	20415.3	2.89	1.21	1	0	0	
aardvark	18	18	0.41	9.89	3.66	1	0	0	
abacus	20	13	0.24	8.69	3.77	0.65	0	0	
abalone	18	13	0.51	12.23	3.54	0.72	0	0	
abandon	19	19	8.1	8.32	2.75	1	0	0	
abandoner	19	18	0.02	11.89	3.36	0.95	0	0	
abandonment	22	22	0.96	10.27	2.57	1	0	0	
abase	19	14	0.06	14.57	4.29	0.74	0	0	
abasement	19	12	nan	15.13	5.37	0.63	0	0	
abate	19	18	0.1	14.44	3.57	0.95	0	0	

... (31113 rows omitted)

In [14]:
```
##Very Positive Words
d.where('Polarity', are.above(0.8))
```

Out[14]:

Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno	Polarity	Polarity Mag	Sul
awesome	18	18	31.37	7.33	3.24	1	1	1	
beautiful	18	18	279.73	5.72	1.84	1	0.85	0.85	
best	21	21	404.37	4.09	1.66	1	1	1	
breathtaking	18	18	1.24	11	2.52	1	1	1	
brilliant	20	20	35.8	7.95	1.76	1	0.9	0.9	
consummate	19	18	0.98	13.78	2.53	0.95	0.95	0.95	
cushy	20	18	0.71	8.83	3.05	0.9	0.9	0.9	
dainty	18	17	1.2	8.76	3.01	0.94	0.9	0.9	
delicious	20	20	21.53	6.5	2.26	1	1	1	
delightful	18	18	9.2	9.72	3.41	1	1	1	

... (17 rows omitted)

In [15]:
```
##Very Subjective Words
d.where('Subjectivity', are.above(0.8))
```

Out[15]:

Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno	Polarity	Polarity Mag	Subj
abrupt	22	22	1.14	10.95	3.47	1	-0.125	0.125	
absolute	19	19	11.31	8.53	3.44	1	0.2	0.2	
absolutely	21	21	113.12	7.08	2.58	1	0.2	0.2	
absurd	1917	1901	9.71	10.5	3.1	0.99	-0.5	0.5	
abundant	19	19	0.57	12.84	2.57	1	0.6	0.6	
acute	20	20	2.94	11.6	3.65	1	0.6	0.6	
addictive	18	18	1.12	10.33	3.01	1	0	0	
addled	19	15	0.22	12	2.83	0.79	-0.46667	0.46667	0
adept	18	14	0.65	12.5	3.08	0.78	0.6	0.6	
adorable	18	18	10.53	6.94	2.62	1	0.5	0.5	

... (412 rows omitted)

3. Categorize words with non-zero polarity and sentiment

In [16]:
```
def categorize_polarity(val):
    if val < 0:
        return -1
    if val > 0:
        return 1
    else:
        return 0

def categorize_subjectivity(val):
    if val > 0:
        return 1
    else:
        return 0

d = d.with_columns('Polar', d.apply(categorize_polarity, 'Polarity'))
d = d.with_columns('Subject', d.apply(categorize_subjectivity, 'Subjectivit
y'))

d.where('Subjectivity', are.above(0.8))
```

Out[16]:

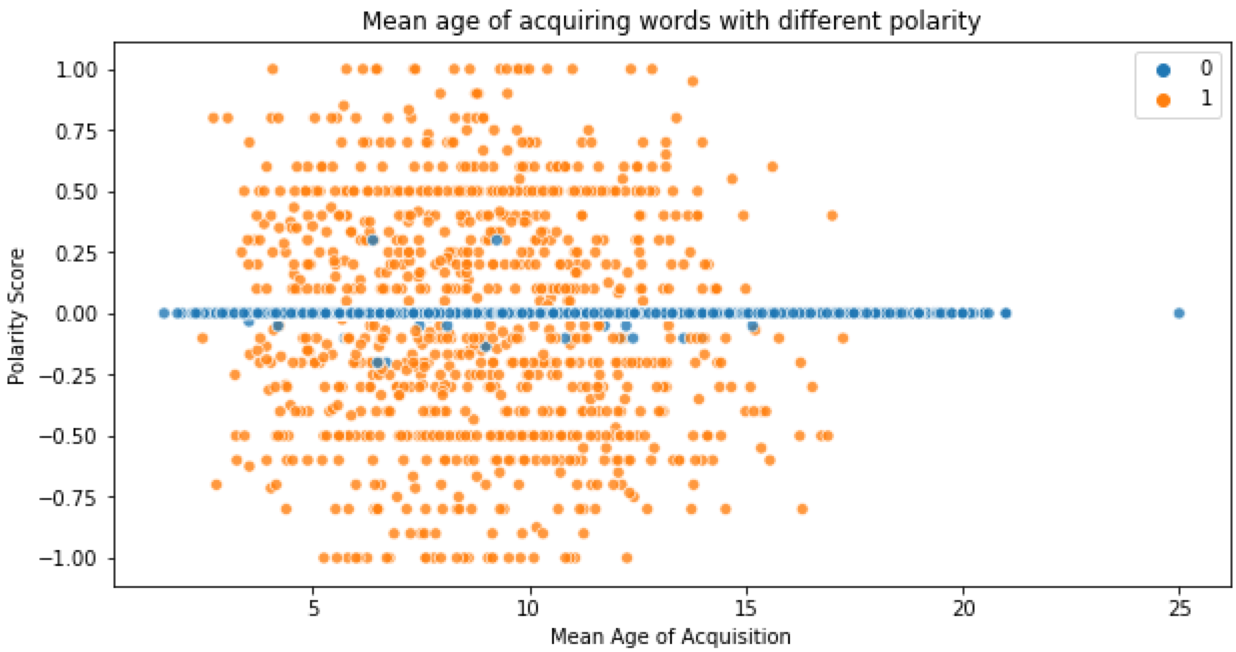
Word	OccurTotal	OccurNum	Freq_pm	Rating.Mean	Rating.SD	Dunno	Polarity	Polarity Mag	Subj
abrupt	22	22	1.14	10.95	3.47	1	-0.125	0.125	
absolute	19	19	11.31	8.53	3.44	1	0.2	0.2	
absolutely	21	21	113.12	7.08	2.58	1	0.2	0.2	
absurd	1917	1901	9.71	10.5	3.1	0.99	-0.5	0.5	
abundant	19	19	0.57	12.84	2.57	1	0.6	0.6	
acute	20	20	2.94	11.6	3.65	1	0.6	0.6	
addictive	18	18	1.12	10.33	3.01	1	0	0	
addled	19	15	0.22	12	2.83	0.79	-0.46667	0.46667	0
adept	18	14	0.65	12.5	3.08	0.78	0.6	0.6	
adorable	18	18	10.53	6.94	2.62	1	0.5	0.5	

... (412 rows omitted)

IV. Data visualization

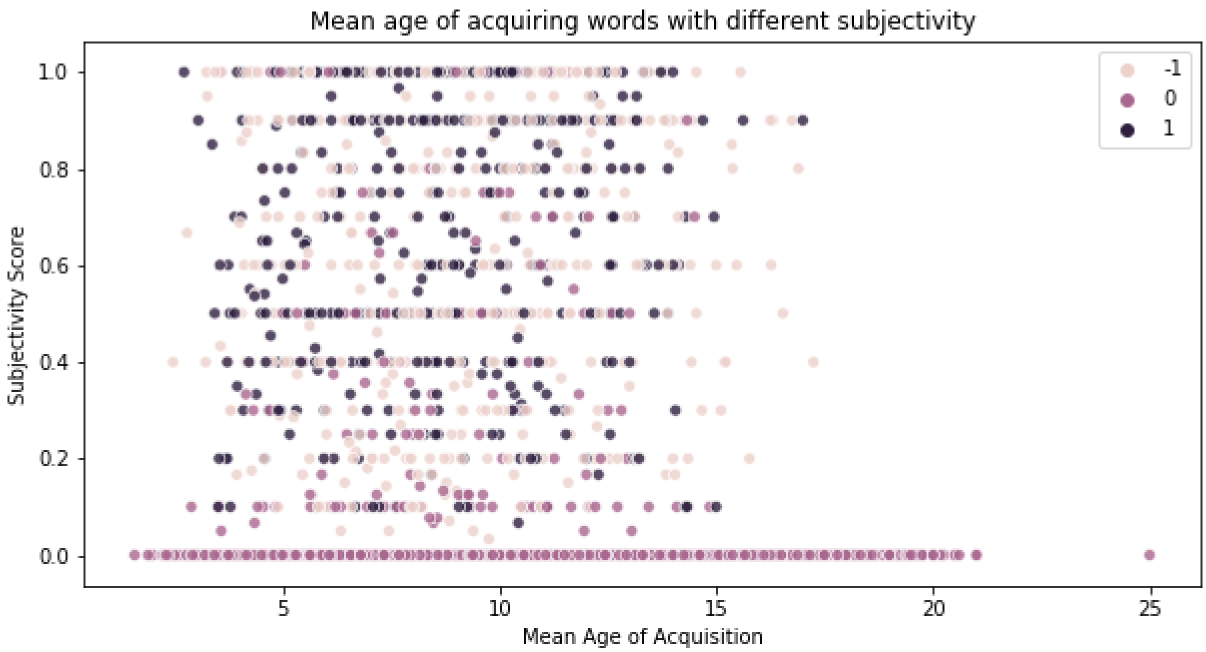
1.1 Overview : Polarity vs Age

In [17]:
```
fig = plt.gcf()
fig.set_size_inches(10, 5)

sns.scatterplot(d.column('Rating.Mean'), d.column('Polarity'), hue = d.colum
n('Subject'), alpha = 0.8);
plt.xlabel('Mean Age of Acquisition');
plt.ylabel('Polarity Score');
plt.title('Mean age of acquiring words with different polarity');
plt.show();
```



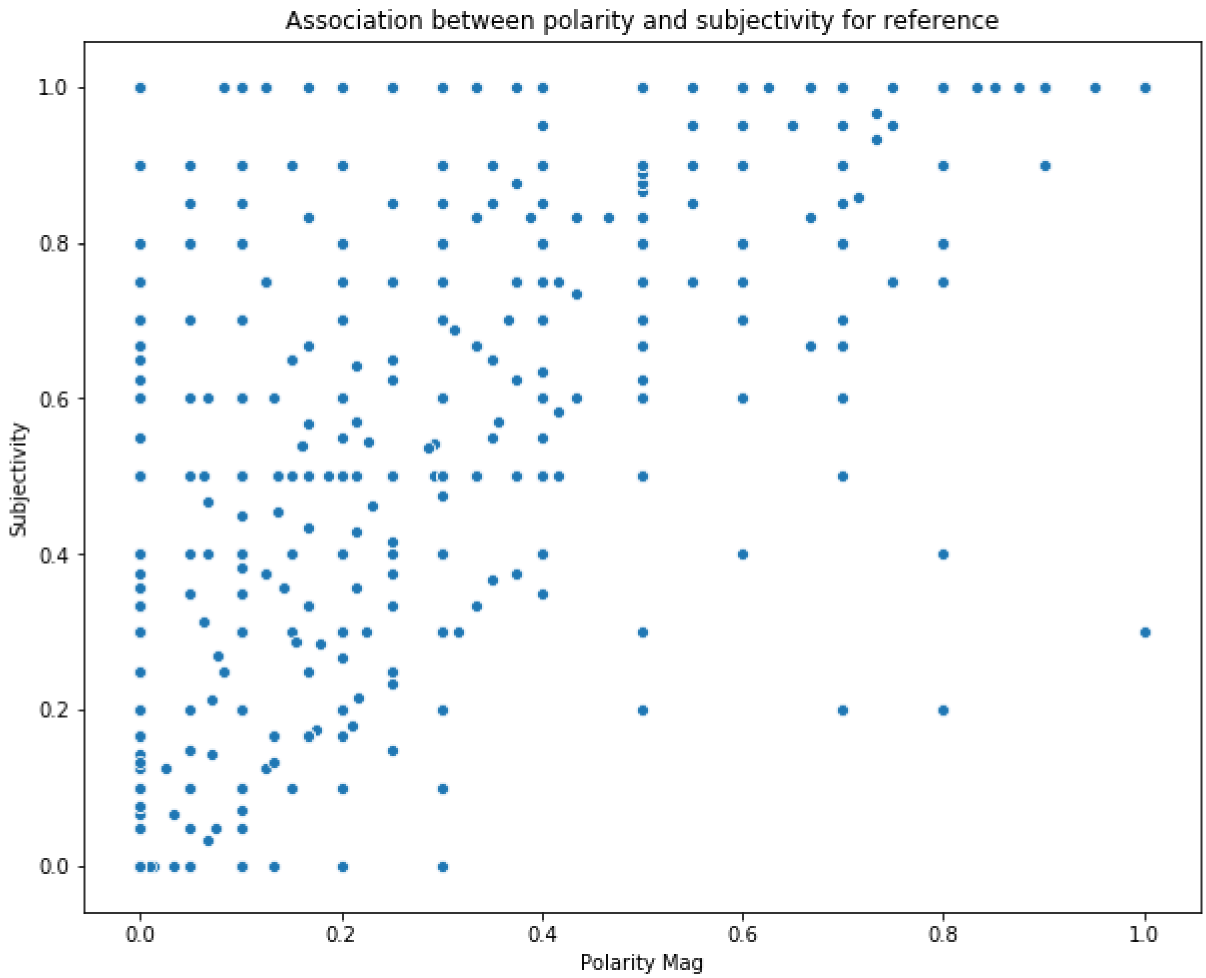
1.2 Overview : Subjectivity vs Age

In [18]:
```
fig = plt.gcf()
fig.set_size_inches(10, 5)

sns.scatterplot(d.column('Rating.Mean'), d.column('Subjectivity'), hue = d.c
olumn('Polar'), alpha = 0.8);
plt.xlabel('Mean Age of Acquisition');
plt.ylabel('Subjectivity Score');
plt.title('Mean age of acquiring words with different subjectivity');
plt.show();
```



1.3 Overview : Caveat - Subjectivity vs Polarity

In [19]:
```
fig = plt.gcf()
fig.set_size_inches(10, 8)

sns.scatterplot(d.column('Polarity Mag'), d.column('Subjectivity'));
plt.xlabel('Polarity Mag');
plt.ylabel('Subjectivity');
plt.title('Association between polarity and subjectivity for reference');
plt.show();
```



In [20]:
```
pearsonr(d.column('Polarity Mag'), d.column('Subjectivity'))
```

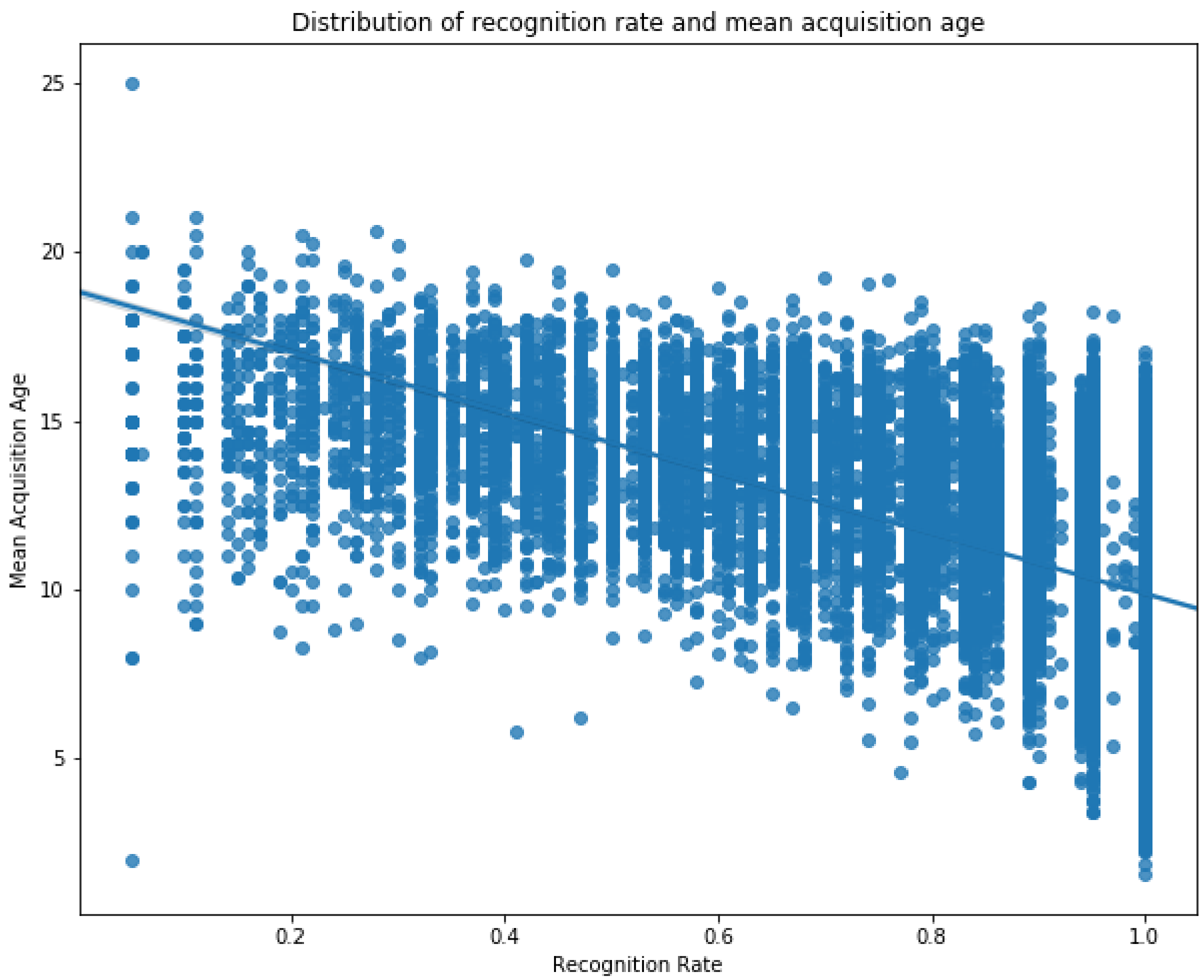
Out[20]: (0.8814526971975205, 0.0)

- It's important for us to realize that it is not because of the database itself but because of the model we import, that the resulting Subjectivity and Polarity Magnitudes are dependent. That it, words with a higher Polarity Maginitude tend to be more subjective by themselves. This decides our following way of looking and exploring the correlation between either of these two variables and Acquisition Age in the statictical testing section.

2.1 Sentiment vs Recognition Rate

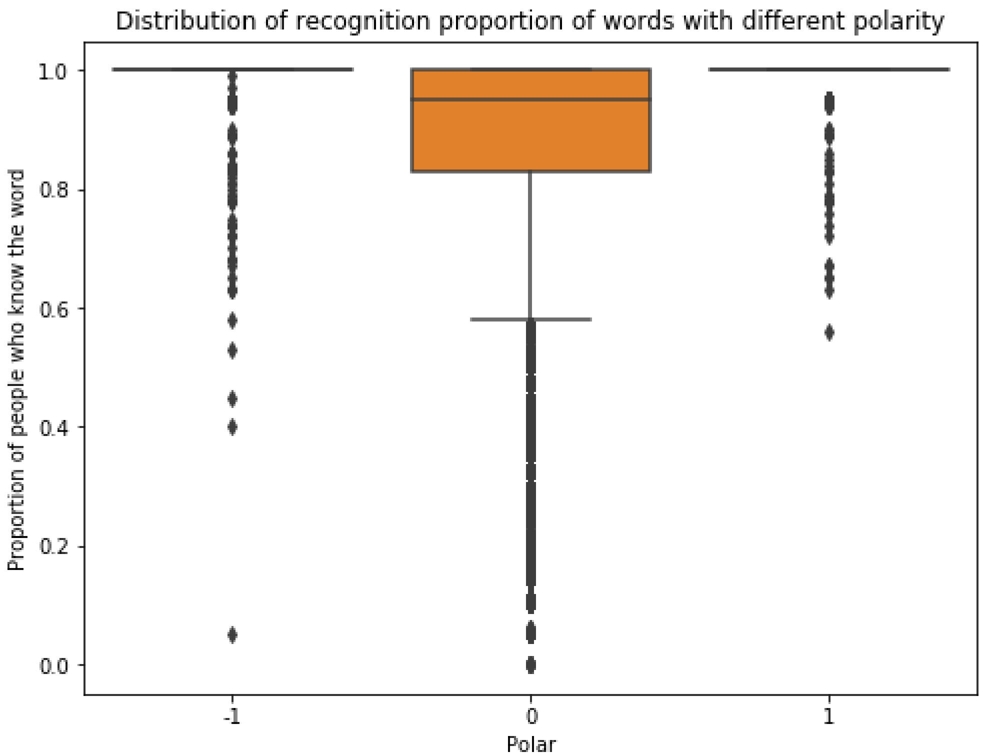
1) Precondition: Recognition seems to be negatively correlated with Mean Acquisition Age

In [21]:
```
fig = plt.gcf()
fig.set_size_inches(10, 8)

sns.regplot(d.column('Dunno'), d.column('Rating.Mean'));
plt.xlabel('Recognition Rate');
plt.ylabel('Mean Acquisition Age');
plt.title('Distribution of recognition rate and mean acquisition age');
plt.show();
```

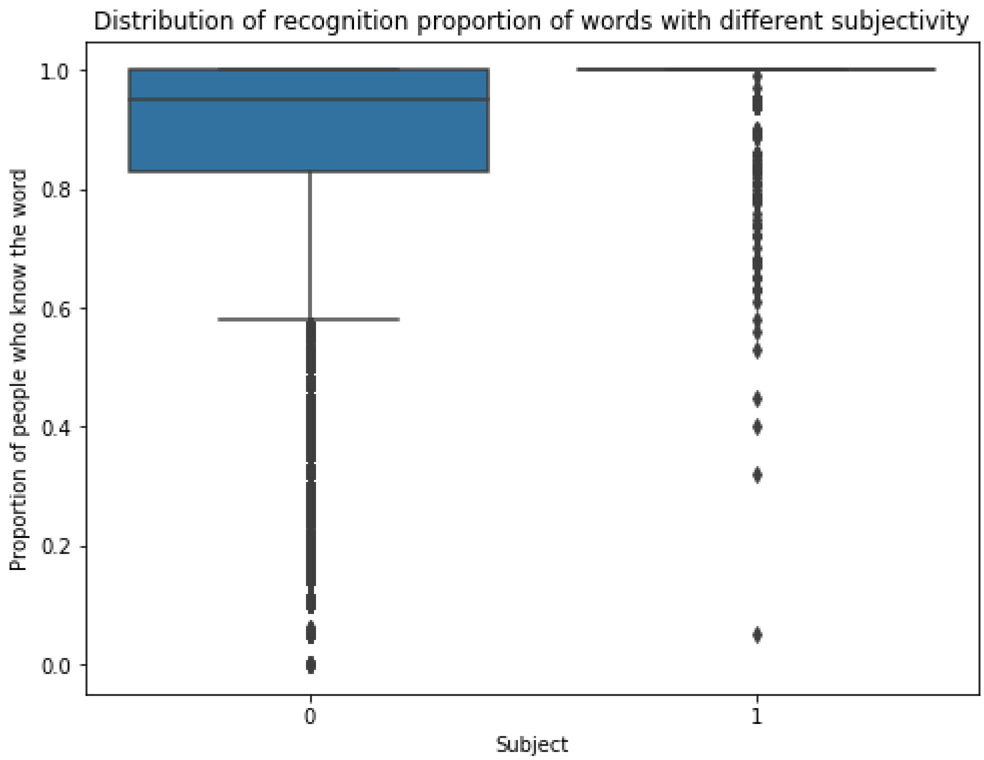


2) Sentimental words seem to be focused on higher recognition rates.

In [22]:
```
fig = plt.gcf()
fig.set_size_inches(8, 6)

sns.boxplot(d.column('Polar'), d.column('Dunno'));
plt.xlabel('Polar');
plt.ylabel('Proportion of people who know the word');
plt.title('Distribution of recognition proportion of words with different po
larity');
plt.show();
```



In [23]:
```
fig = plt.gcf()
fig.set_size_inches(8, 6)

sns.boxplot(d.column('Subject'), d.column('Dunno'));
plt.xlabel('Subject');
plt.ylabel('Proportion of people who know the word');
plt.title('Distribution of recognition proportion of words with different su
bjectivity');
plt.show();
```



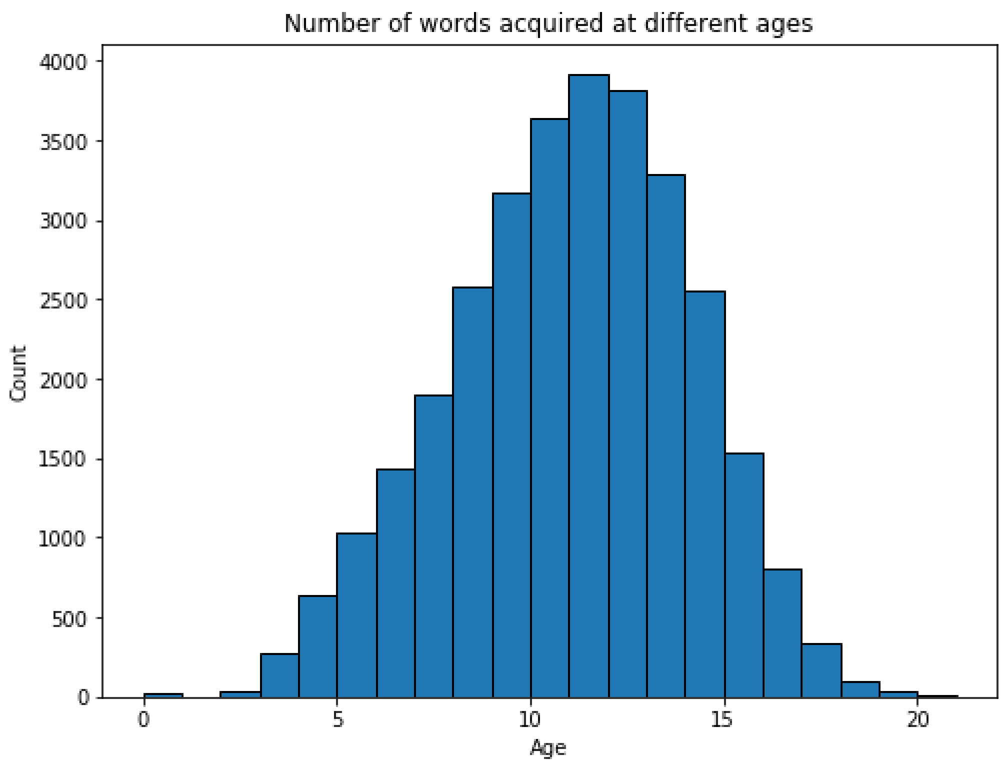
V. Statistical Testing

In this section I will test my hypothesis using correlation coefficient and fit a linear regression model on my data.

- **H1**: As age increases, the proportion of words with extreme polarity and subjectivity that we learned in a certian period decreases.
- **H0(Null Hypothesis)**: The change of proportion of polar and subject words we learn along the change of age is due to chance.

0. Counts of Words Learned at Different Ages

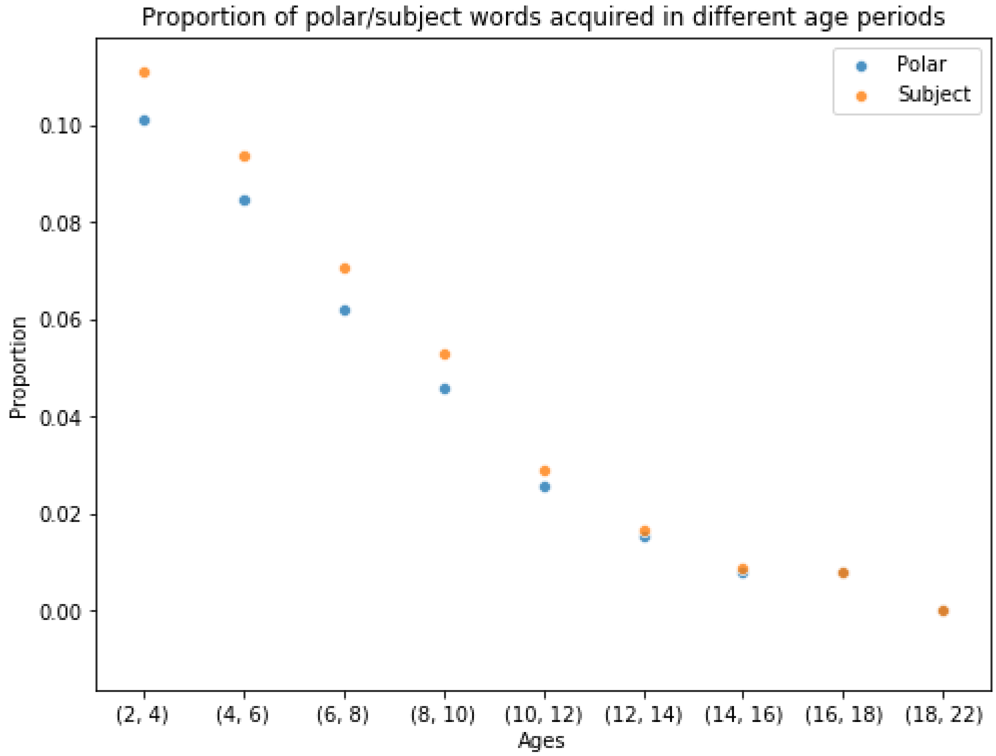
In [24]:
```
fig = plt.gcf()
fig.set_size_inches(8, 6)

new_col = np.nan_to_num(d.column('Rating.Mean'), np.nanmean(d.column('Ratin
g.Mean')))
plt.hist(new_col, bins = range(0, 22));

plt.xlabel('Age');
plt.ylabel('Count');
plt.title('Number of words acquired at different ages');
plt.show();
```



1. Discrete Stages (Age Periods)

We first split the range of acquisition age into small periods: we will calculate the proportion of polar/subject words learned in every two years.

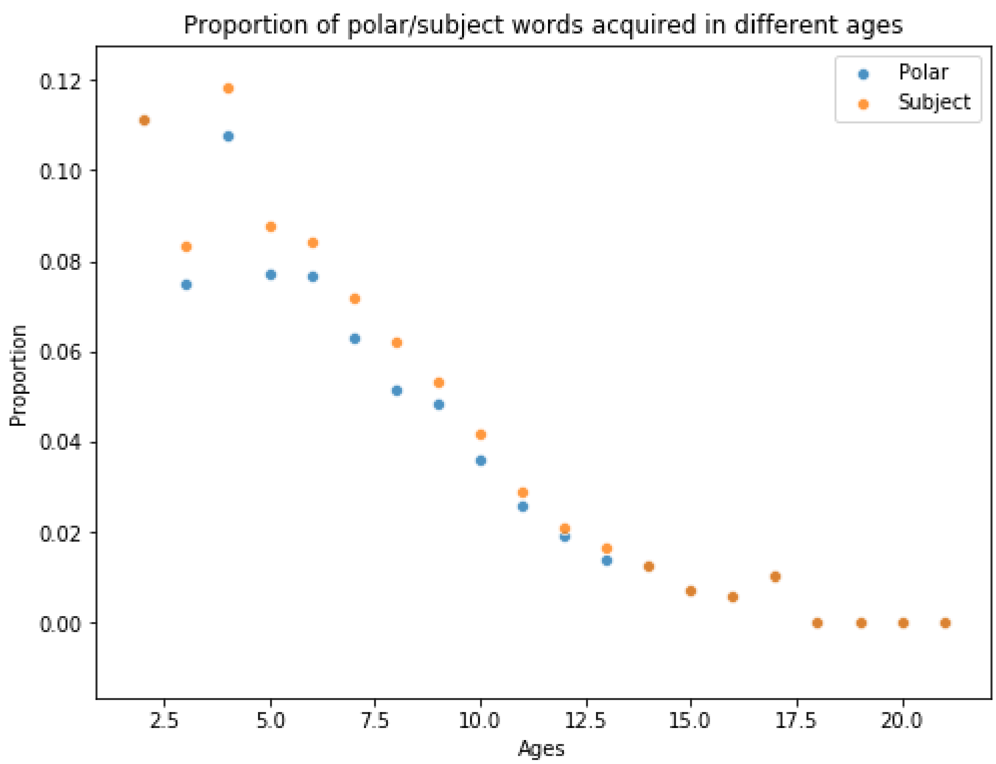
In [25]:
```
stages = [(2, 4), (4, 6), (6, 8), (8, 10), (10, 12), (12, 14), (14, 16), (16
, 18), (18, 22)]
stages_str = ["(" + str(t[0]) + ", "+ str(t[1]) + ")"for t in stages]
polar_props = []
subject_props = []


def calc_polar_prop(t1, t2):
    tb = d.where('Rating.Mean', are.above(t1)).where('Rating.Mean', are.belo
w(t2))
    if tb.num_rows == 0:
        return 0
    return np.sum(np.abs(tb.column('Polar')))/ tb.num_rows

def calc_subject_prop(t1, t2):
    tb = d.where('Rating.Mean', are.above(t1)).where('Rating.Mean', are.belo
w(t2))
    if tb.num_rows == 0:
        return 0
    return np.sum(tb.column('Subject'))/ tb.num_rows




for t1, t2 in stages:
    polar_props.append(calc_polar_prop(t1, t2))


for t1, t2 in stages:
    subject_props.append(calc_subject_prop(t1, t2))




fig = plt.gcf()
fig.set_size_inches(8, 6)

sns.scatterplot(stages_str, polar_props, label = "Polar", alpha = 0.8);
sns.scatterplot(stages_str, subject_props, label = "Subject", alpha = 0.8);
plt.xlabel('Ages');
plt.ylabel('Proportion');
plt.title('Proportion of polar/subject words acquired in different age perio
ds');
plt.show();
```



2. Continuous Stages (Age)

To get more "continuous" data, for each data point of age, we take a very small period around it and calculate the proportions with regard to the period.

In [26]:
```
ages = range(2, 22)
polar_props = []
subject_props = []

for a in ages:
    polar_props.append(calc_polar_prop(a-0.5, a+0.5))
    subject_props.append(calc_subject_prop(a-0.5, a+0.5))

fig = plt.gcf()
fig.set_size_inches(8, 6)

sns.scatterplot(ages, polar_props, label = "Polar", alpha = 0.8);
sns.scatterplot(ages, subject_props, label = "Subject", alpha = 0.8);
plt.xlabel('Ages');
plt.ylabel('Proportion');
plt.title('Proportion of polar/subject words acquired in different ages');
plt.show();
```



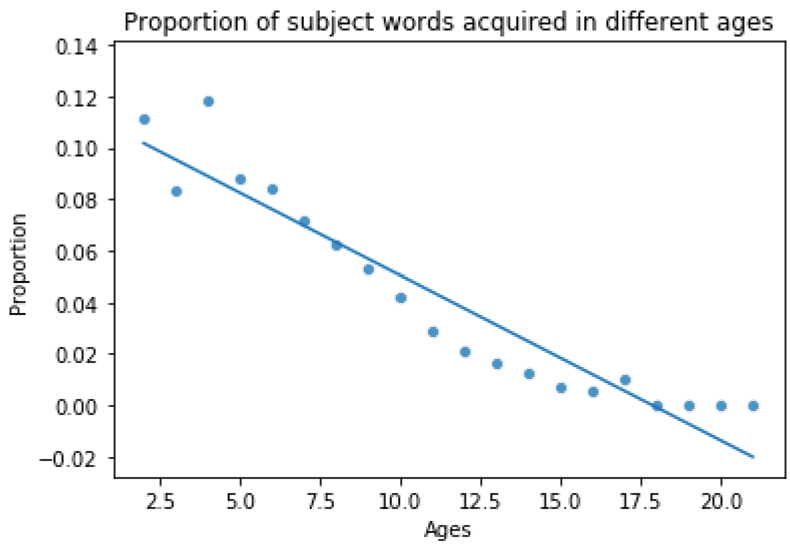
3. Hypothesis Testing on Subject Proportion vs Age

Since the distributions of polar words and subject words appear to be very similar in the plot above, proving the association between age and either one of the two variables will suffice to prove the other.

In [27]:
```
r, p = pearsonr(ages, subject_props)
print("Correlation Coef: ", r)
print("Probability of H0: ", p)
```

```
Correlation Coef:  -0.949213464992743
Probability of H0:  1.774294799792572e-10
```

In [28]:
```
slope, intercept = np.polyfit(ages, subject_props, deg = 1)
modeled_y = np.multiply(slope, ages) + intercept
sns.scatterplot(ages, subject_props, alpha = 0.8);
plt.plot(ages, modeled_y)
plt.xlabel('Ages');
plt.ylabel('Proportion');
plt.title('Proportion of subject words acquired in different ages');
plt.show();
```

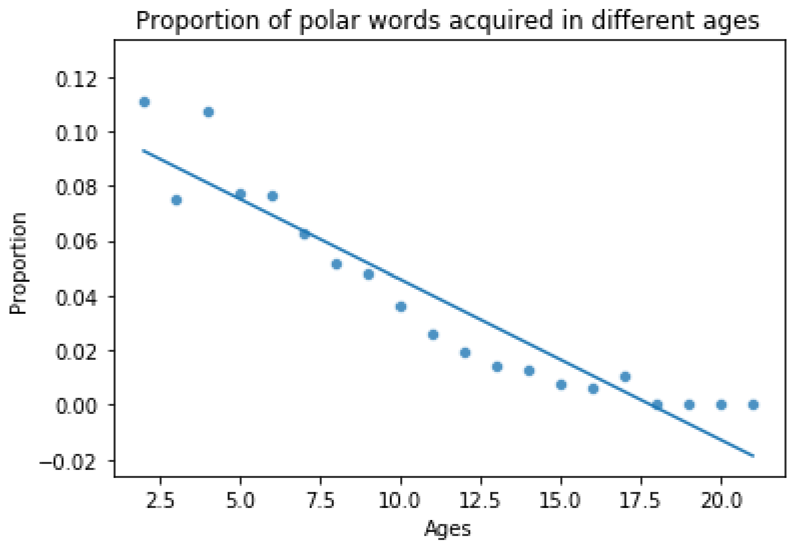


4. Hypothesis Testing on Polar Proportion vs Age

In [29]:
```
r, p = pearsonr(ages, polar_props)
print("Correlation Coef: ", r)
print("Probability of H0: ", p)
```

```
Correlation Coef:  -0.9435547788392432
Probability of H0:  4.496594152360724e-10
```

In [30]:
```
slope, intercept = np.polyfit(ages, polar_props, deg = 1)
modeled_y = np.multiply(slope, ages) + intercept
sns.scatterplot(ages, polar_props, alpha = 0.8);
plt.plot(ages, modeled_y)
plt.xlabel('Ages');
plt.ylabel('Proportion');
plt.title('Proportion of polar words acquired in different ages');
plt.show();
```



VI. Conclusion

1. Results From Statistical Testing

- In the plot above, we have fit a very good linear regression model to the dataset which closely predicts the proportion of subject words acquired at a certain age.
- Using the function of Pearsonr, we find that the linear correlation coefficient between the two variables(age and proportion of subject words) is very close to -1, the high magnitude of which testifies that our hypothesis is very likely to be true. It is very unlikely (prob is almost 0) for us to reject our hypothesis and conclude with the null hypothesis.
- As our age increases, the proportion of sentimental words learned decreases in the process of word acquisition.
- Although the number of words we learn at different ages shows an almost "normal" distribution, the proportion of sentimental words we learn still shows a nearly drastically decreasing trend. This is a very strong support for the validity of our hypothesis.

2. Future Research Queations and Implications From the Results

- Social Behavioral Significance: Human's learning experience is dependent on our application need. As our result shows that a greater proportion of sentimental words is learned at an earlier age, would it prove that we are in a higher demand of straightforward/explicit expressions at early life stages?
- Educational Significance: It is also believed that human's choice of learning is dependent on our perception of the world and choice of interest. With that being said and according to our result, are children more open to learning and understanding sentiment-driven behaviors at an earlier age?

3. Potential Drawbacks in This Project

- The choice of words in the data base might affect our result. If our choice of subject/polar words is biased in terms of their frequencies and counts as compared to other words, our result might become less reliable. For example, we might know more sentimental words when we are younger simply because these words are shorter and easier to learn.
- Potential bias in using TextBlob's complex model: Since polarity score and subjectivity score are continuous, our choice of less proper threshold for categorization could cause bias. For example, a word with polarity score = 0.2 might not be very "polar" in some people's opinion, and so simply calling a word "polar" as long as its score is not zero might not be good enought. As a matter of fact, when I looked into the scores returned by TextBlob, only words with score magnitudes greater than 0.7 could be interpreted to be obviously polar/subject from my intuition.

4. Credits and Thanks

- BIG THANKS to Andrew and Geoff for this great, great semester! I really had fun taking this class!

In [ ]:

```

```