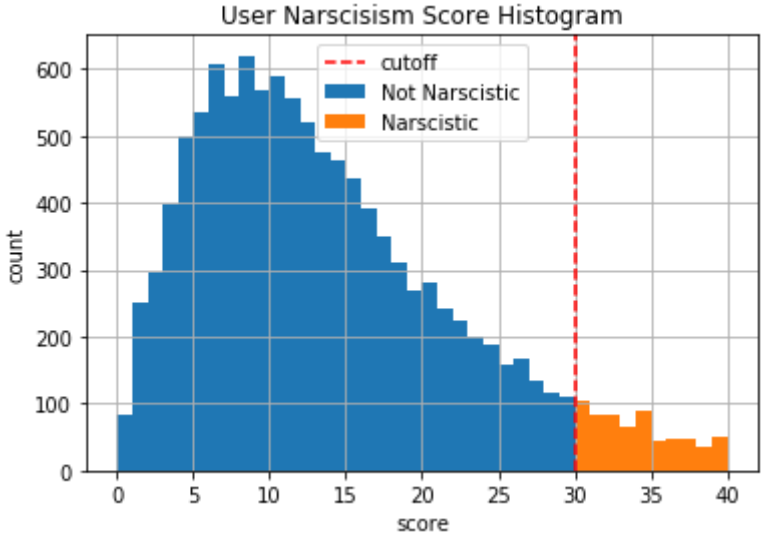
In [8]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Looking at the Narscisism data

In [9]:
```
# Read in the dataset
df = pd.read_csv('data.csv')

#plot the scores
idx = df.score < 30
ax = df.score[idx].hist(bins = range(0,31),label = 'Not Narscistic')
idx = df.score >= 30
ax.hist(df.score[idx],label = 'Narscistic')
ax.set_xlabel('score')
ax.set_ylabel('count')
ax.set_title('User Narscisism Score Histogram')
ax.axvline(30,0,600,c = 'r',ls = '--',label ='cutoff')
ax.legend();
```



From above, we can identify users as narscistic if they answer enough questons that push their score above 30.

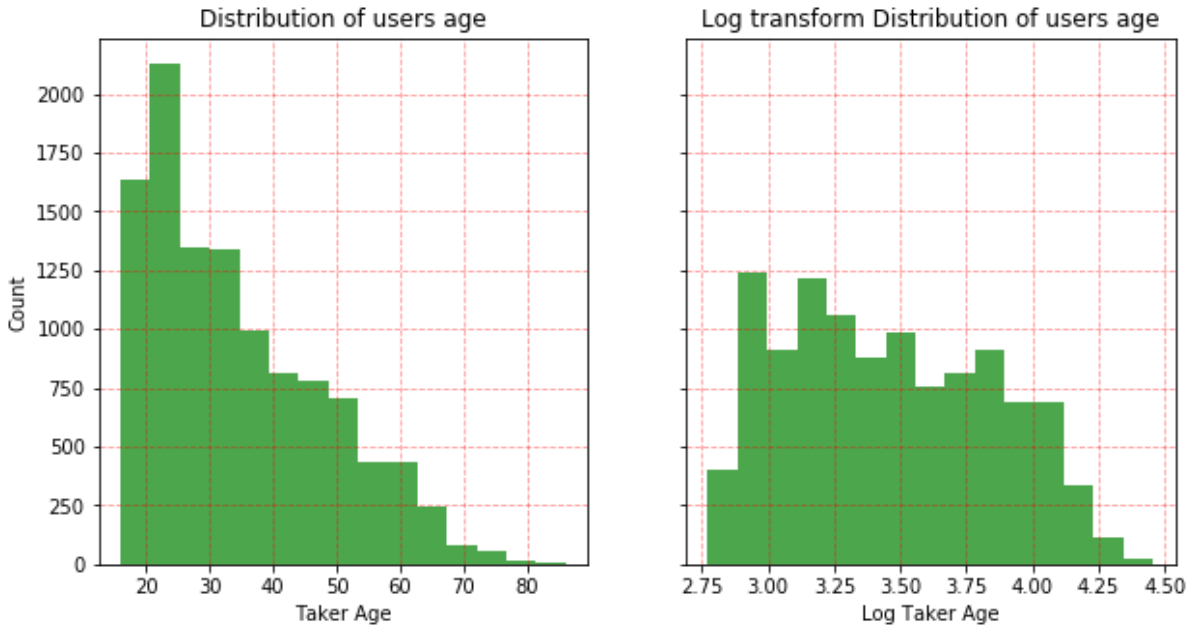
My goal is to find the questions that are answered the most differently between users that are narscistic vs not narscistic.

Initially, I am going to get rid of some of the data in this set that is bad. There are 2 defining factors that I identified as making the data bad.

1. Remove all test takers that said their age was **greater** than 100 or **less than** 15
2. Remove all test takers who took longer than 10,000 seconds to take the test

Histogram of user age -- Before and After a log transformation

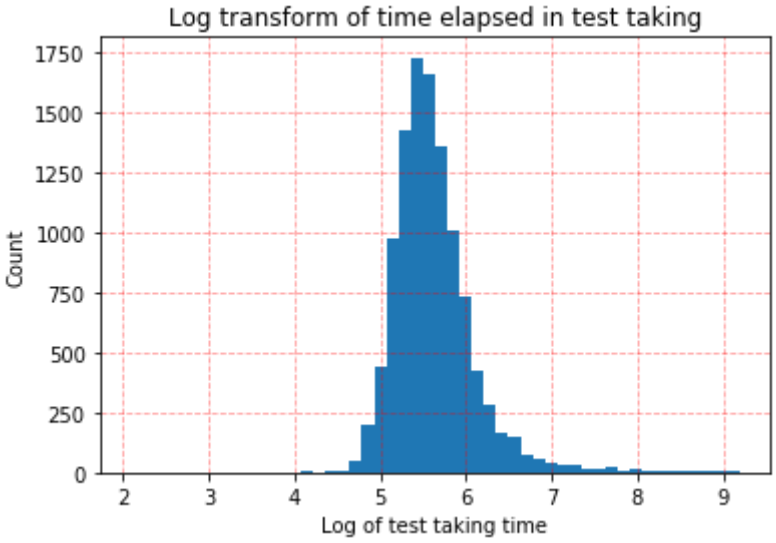
In [10]:
```
# Find the Index of the nonoutliers
df_idx = (df['age'] < 100) & (df['age'] > 15)

# Only save the good data
df = df.loc[df_idx,:]

#Create a subplots object
_,ax = plt.subplots(1,2,figsize = (10,5),sharey = True)

#plot the two histograms
ax[0].hist(df['age'], bins = 15,alpha = .7,color = 'g');
ax[1].hist(np.log(df['age']), bins = 15,alpha = .7,color = 'g');

#Set X and Y labels
ax[0].set_ylabel('Count')
ax[0].set_xlabel('Taker Age')
ax[1].set_xlabel('Log Taker Age')

# Set the titles
ax[0].set_title('Distribution of users age')
ax[1].set_title('Log transform Distribution of users age')

#add the grid
ax[1].grid(alpha = .4, c = 'r',ls = '--')
ax[0].grid(alpha = .4, c = 'r',ls = '--')
```



Test time elapsed removal:

Now lets take a look at the Distribution of the elapsed time. Lets exclude users who took more than 10,000 seconds. As well, going to take the log of these numbers so that the distribution looks a litle better

In [11]:
```
# Find the index of the non outliers in time to take the test
df_idx = df['elapse'] < 10000

# Get rid of the outliers
df = df.loc[df_idx,:]

# plot the histogram
plt.hist(np.log(df['elapse']), bins = 50);

# set the xlabel, ylabel and the title
plt.xlabel('Log of test taking time')
plt.ylabel('Count')
plt.title('Log transform of time elapsed in test taking')

# add a grid
plt.grid(alpha = .4, c = 'r',ls = '--')
```



Data is cleaned, Ready for analysis

Now that we have the data cleaned, we are ready to start looking at what identifies people as narcissitic the most.

The way these questions are organized is that there are *40* questions, and within these questions the test taker is required to choose a statement that they identify most with. Then at the end of the test, the test sums the number of questions that are identified with narcissistic personality. I will reoder the questions into a dictionary where the first value **always** is identified with being more narcissistic.

Below is the code to do this

In [12]:
```
# Questions identified as being narcissitic with answer A
Questions_A = [1, 2, 3, 6, 8, 11, 12, 13, 14, 16,
               21, 24, 25, 27, 29, 30, 31, 33, 34,
              36, 37, 38, 39]
# Questions identified as being narcissitic with answer B
Questions_B = [4, 5, 7, 9, 10, 15, 17, 18, 19, 20,
               22, 23, 26, 28, 32, 35, 40]
#create the dictionary, and go through the text file containing the ques
tions and answers
questions = {}
with open('codebook.txt') as f:
    for i in range(4):
        f.readline()
    for i,line in enumerate(f):
        if i == 40:
            break
        data = line.split('.')
        if i + 1 in Questions_A:
            questions[data[0]] = [data[1].replace("1=",'').strip(),
                                                  data[2].replace("2=",'').s
trip()]
        else:
            questions[data[0]] = [data[2].replace("2=",'').strip(),
                                                  data[1].replace("1=",'').s
trip()]
```

Questions that imply narcissism:

Now that I have taken these questions and organized them accordingly, We can take a look at the questions that are identified with people being narcissitic:

In [14]:
```
# Build the table (requires a couple imports)
import textwrap
from prettytable import PrettyTable

# Set up the table that we will be adding each to
t = PrettyTable(["Q's that impply Narcissism", "Q's that imply not Narci
ssism"])

# Go through each question and add each to the table
for i, quest in enumerate(questions.values()):

    # Wrap the text questions so that we have nice formatting
    q1 = textwrap.fill(quest[0], 30)
    q2 = textwrap.fill(quest[1], 30)

    # add the question to the table
    t.add_row([q1,q2])
    t.add_row(['--','--'])
print(t)
```

Q's that impply Narcissism	Q's that imply not Narcissism
I have a natural talent for influencing people	I am not good at influencing people
Modesty doesn't become me	I am essentially a modest person
I would do almost anything on a dare	I tend to be a fairly cautious person
I know that I am good because everybody keeps telling me so	When people compliment me I sometimes get embarrassed
If I ruled the world it would be a better place	The thought of ruling the world frightens the hell out of me
I can usually talk my way out of anything	I try to accept the consequences of my behavior
I like to be the center of attention	I prefer to blend in with the crowd
I will be a success	I am not too concerned about success
I think I am a special person	I am no better or worse than most people
I see myself as a good leader	I am not sure if I would make a good leader
I am assertive	I wish I were more assertive
I like to have authority over other people	I don't mind following orders
I find it easy to manipulate people	I don't like it when I find myself manipulating people
I insist upon getting the respect that is due me	I usually get the respect that I deserve
I like to show off my body	I don't particularly like to show off my body
I can read people like a book	People are sometimes hard to understand
I like to take responsibility for making decisions	If I feel competent I am willing to take responsibility for making decisions
I want to amount to something in the eyes of the world	I just want to be reasonably happy

--	--
I like to look at my body	My body is nothing special
--	--
I will usually show off if I get the chance	I try not to be a show off
--	--
I always know what I am doing	Sometimes I am not sure of what I am doing
--	--
I rarely depend on anyone else to get things done	I sometimes depend on people to get things done
--	--
Everybody likes to hear my stories	Sometimes I tell good stories
--	--
I expect a great deal from other people	I like to do things for other people
--	--
I will never be satisfied until I get all that I deserve	I take my satisfactions as they come
--	--
I like to be complimented	Compliments embarrass me
--	--
I have a strong will to power	Power for its own sake doesn't interest me
--	--
I like to start new fads and fashions	I don't care about new fads and fashions
--	--
I like to look at myself in the mirror	I am not particularly interested in looking at myself in the mirror
--	--
I really like to be the center of attention	It makes me uncomfortable to be the center of attention
--	--
I can live my life in any way I want to	People can't always live their lives in terms of what they want
--	--
People always seem to recognize my authority	Being an authority doesn't mean that much to me
--	--
I would prefer to be a leader	It makes little difference to me whether I am a leader or not
--	--
I am going to be a great person	I hope I am going to be successful
--	--
I can make anybody believe anything I want them to	People sometimes believe what I tell them
--	--
I am a born leader	Leadership is a quality that takes a long time to develop
--	--
I wish somebody would someday	I don't like people to pry

```
|       write my biography        |  into my life for any reason    |
|                --               |                --               |
| I get upset when people don't   | I don't mind blending into the  |
|  notice how I look when I go    | crowd when I go out in public   |
|         out in public           |                                 |
|                --               |                --               |
|  I am more capable than other   |   There is a lot that I can     |
|             people              |    learn from other people      |
|                --               |                --               |
|  I am an extraordinary person   | I am much like everybody else   |
|                --               |                --               |
+---------------------------------+---------------------------------+
```

Scaling the answers to make the analysis simplier:

I am going to apply a scaling function to the answers so that the analysis works a little better and is slightly cleaner

In [15]:
```
# Import the scaling Function
from sklearn.preprocessing import StandardScaler as SS

#Initialize the scaler
scaler = SS()

#apply the scaling to the whole dataset
sc_df = pd.DataFrame(scaler.fit_transform(df),columns = df.columns)
```

Now that we have have removed the outliers, and scaled the answers, lets go ahead and start doing some real analysis.

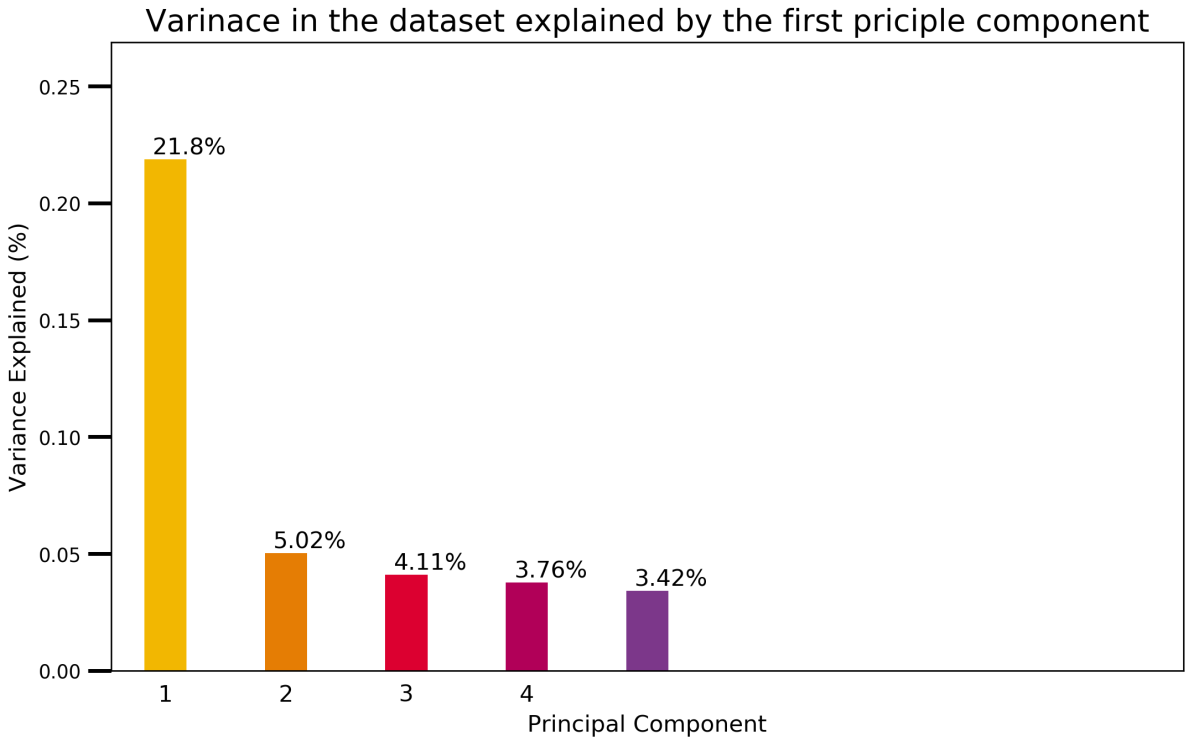
PCA will identify the components of the data - set where there is the most variation in answers. Using this information, I will then identify the questions that seperate users the most. Since PCA gives some of the components as negative values, I will later on explore **NMF** but for now this is a good start

In [16]:
```
# Take only the columns from our dataset that correspond to question Ans
wers
nar_df = sc_df.loc[:,'Q1':'Q40']

# Import the PCA module
from sklearn.decomposition import PCA

#create the PCA object and apply it to the dataset
model = PCA(n_components=5)
model.fit(nar_df);
```

Lets take a look at how much variation in the Dataset is explained by the first couple priciple components that where Identified:

In [18]:
```
def scree_plot(pca, title=None):
    num_components = pca.n_components_
    ind = np.arange(num_components)
    vals = pca.explained_variance_ratio_
    plt.figure(figsize=(10, 6), dpi=250)
    ax = plt.subplot(111)
    ax.bar(ind, vals, 0.35,
           color=[(0.949, 0.718, 0.004),
                  (0.898, 0.49, 0.016),
                  (0.863, 0, 0.188),
                  (0.694, 0, 0.345),
                  (0.486, 0.216, 0.541),
                  (0.204, 0.396, 0.667),
                  (0.035, 0.635, 0.459),
                  (0.486, 0.722, 0.329),
                 ])

    for i in range(num_components):
        ax.annotate(r"%s%%" % ((str(vals[i]*100)[:4])), (ind[i]+0.2, val
s[i]), va="bottom", ha="center", fontsize=12)

    ax.set_xticklabels(ind,
                       fontsize=12)

    ax.set_ylim(0, max(vals)+0.05)
    ax.set_xlim(0-0.45, 8+0.45)

    ax.xaxis.set_tick_params(width=0)
    ax.yaxis.set_tick_params(width=2, length=12)

    ax.set_xlabel("Principal Component", fontsize=12)
    ax.set_ylabel("Variance Explained (%)", fontsize=12)

    if title is not None:
        plt.title(title, fontsize=16)
```

In [19]:
```
scree_plot(model,title = 'Varinace in the dataset explained by the first
 priciple component')
```



PCA explained variance

We can see from the above plot that over 20% of the variance in the dataset is explained by the first principle component. Using this information, we can also look at the questions that are giving this component the most signal.

- Later on I will aggragate the first 5 components, and see what questions are most important in all 5. For now, I will be just looking at the first component

Below is a list of the most influencial questions in terms of explaining variance in the first PCA Component

In [21]:
```
np.argsort(np.abs(model.components_[0]))
```

Out[21]:
```
array([21, 16, 10,  2, 13, 14, 36, 23, 18, 25,  0, 27, 15, 28, 20,  7,
9,
        1, 17, 37, 33,  4, 30, 22, 38, 19, 32,  6,  8,  5,  3, 11, 24,
31,
       12, 29, 34, 39, 35, 26])
```

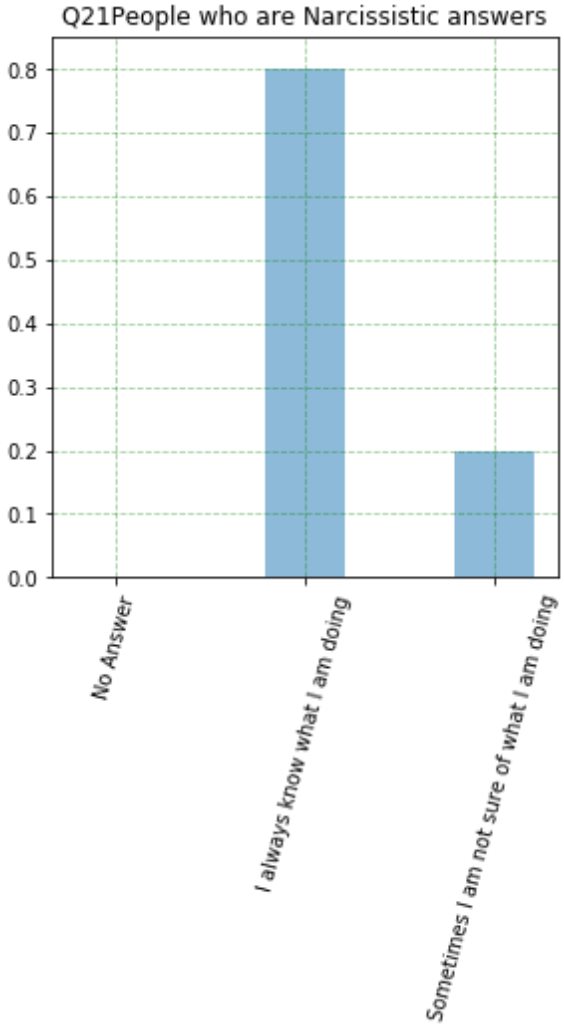
What where users answers from these questions?

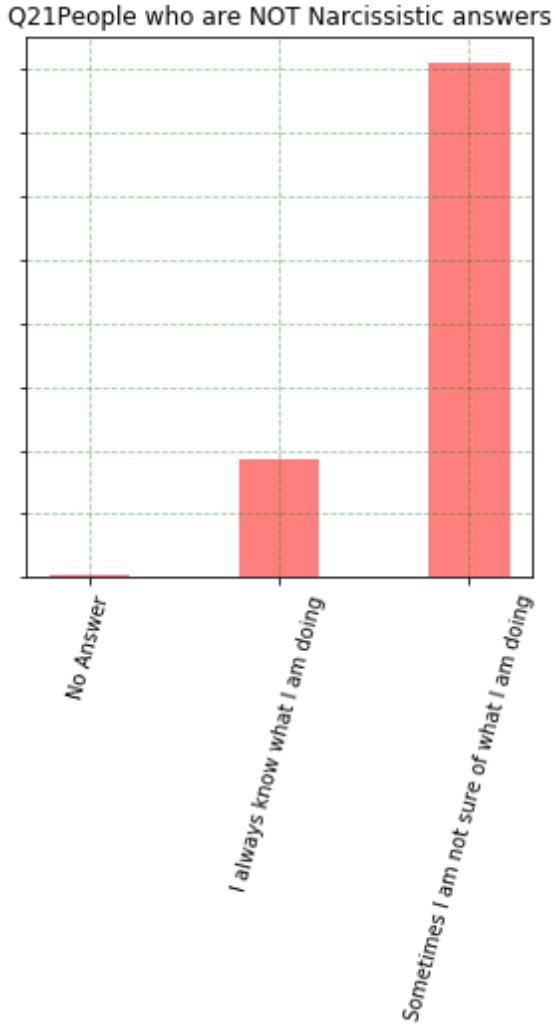
- Lets see how users responded to these questions and what kind of split they give

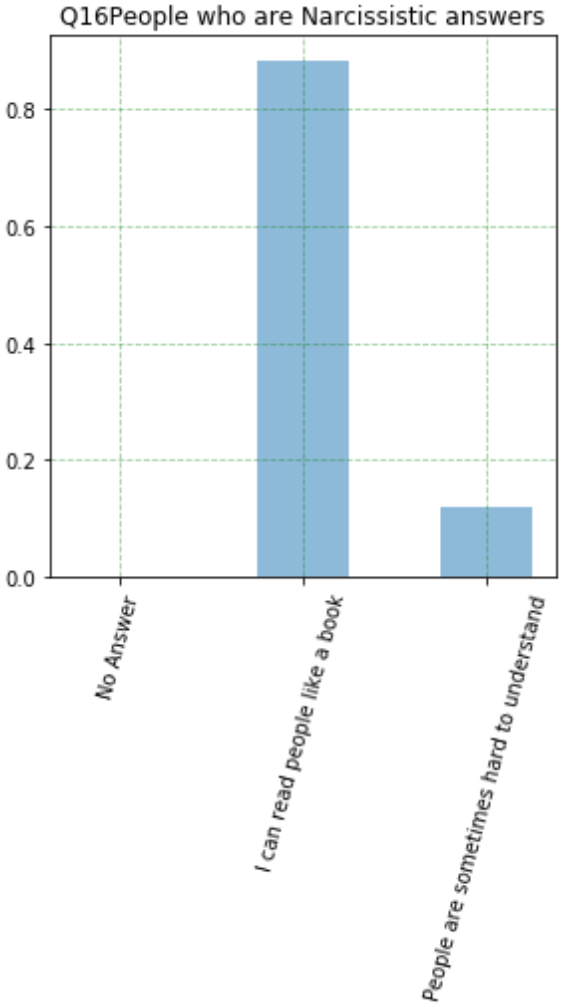
In [23]:
```
# Define the people who are narcissistic
nar_idx = df.score > 30
nar_idx = nar_idx.reset_index()
nar_idx = nar_idx['score']

# Define the people who are not narcissistic
not_idx = df.score < 30
not_idx = not_idx.reset_index()
not_idx = not_idx['score']
def plot_top_n(components,scaled_df,nar_idx, not_idx,num_comp):

    # Go through the top 'n questions
    for i in range(num_comp):
        # Create the plot
        _,ax = plt.subplots(1,2,figsize = (10,5),sharey = True)

        # define the question we are looking at
        name = 'Q'+str(components[i])
        data = scaled_df[name].reset_index()

        # Get the tick values (since we scaled them they are a little st
range)
        tick_vals = sorted(list(set(scaled_df[name])))

        # create the bins for plotting
        b = []
        for val in tick_vals:
            b.append(val-.5)
            b.append(val+.5)

        # plot the histograms
        data[nar_idx][name].hist(density = 1,alpha = .5,
                                 ax = ax[0],bins = b)
        data[not_idx][name].hist(density = 1,alpha = .5,
                                 ax = ax[1],bins = b,
                                 color = 'r')

        # Set the tick marks
        ax[0].set_xticks(tick_vals)
        ax[1].set_xticks(tick_vals)

        #Code to get the right labels
        if components[i] in Questions_A:
            nar = 0
            not_nar = 1
        else:
            nar = 1
            not_nar = 0
        ax[1].set_xticklabels(['No Answer',
                               questions[name][nar],
                               questions[name][not_nar]],
                              rotation=75)
        ax[0].set_xticklabels(['No Answer',
                               questions[name][nar],
                               questions[name][not_nar]],
                              rotation=75)
```

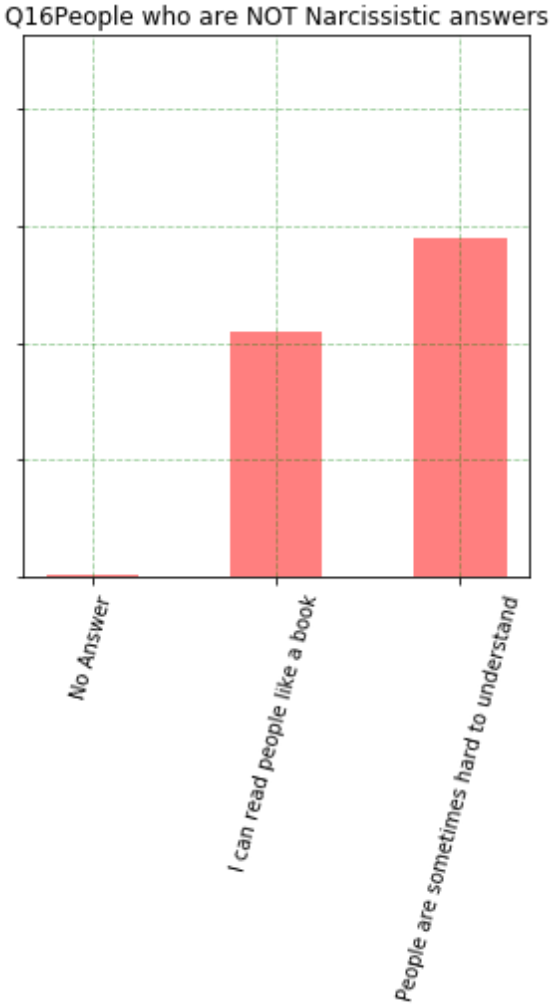
```
        # Add a grid
        ax[0].grid(alpha = .4, c= 'g',ls = '--')
        ax[1].grid(alpha = .4, c= 'g',ls = '--')

        #add a title
        ax[0].set_title(name + 'People who are Narcissistic answers')
        ax[1].set_title(name + 'People who are NOT Narcissistic answers'
)
        plt.show()
```

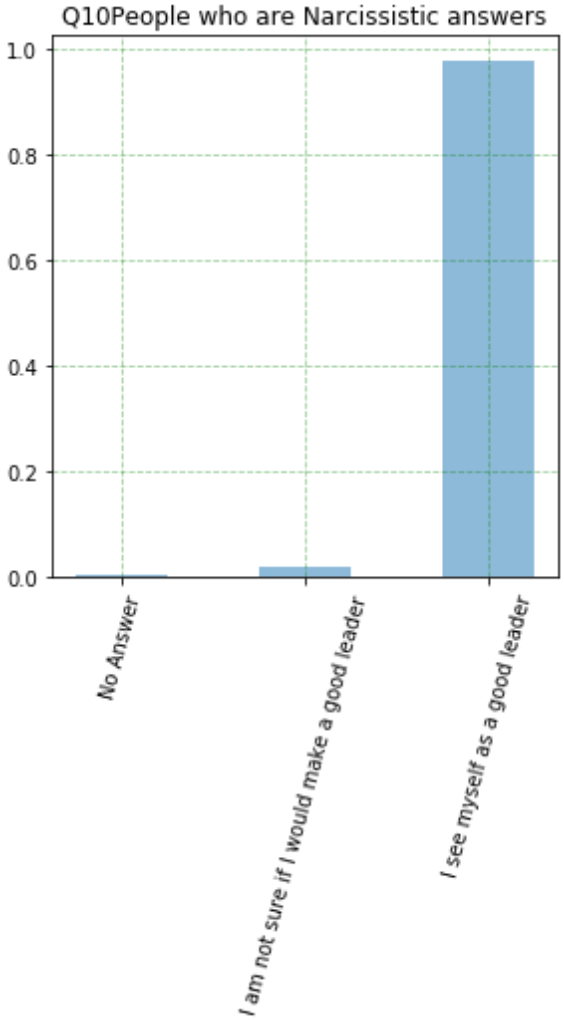
In [24]:
```
components = np.argsort(np.abs(model.components_[0]))
plot_top_n(components,sc_df,nar_idx, not_idx,5)
```

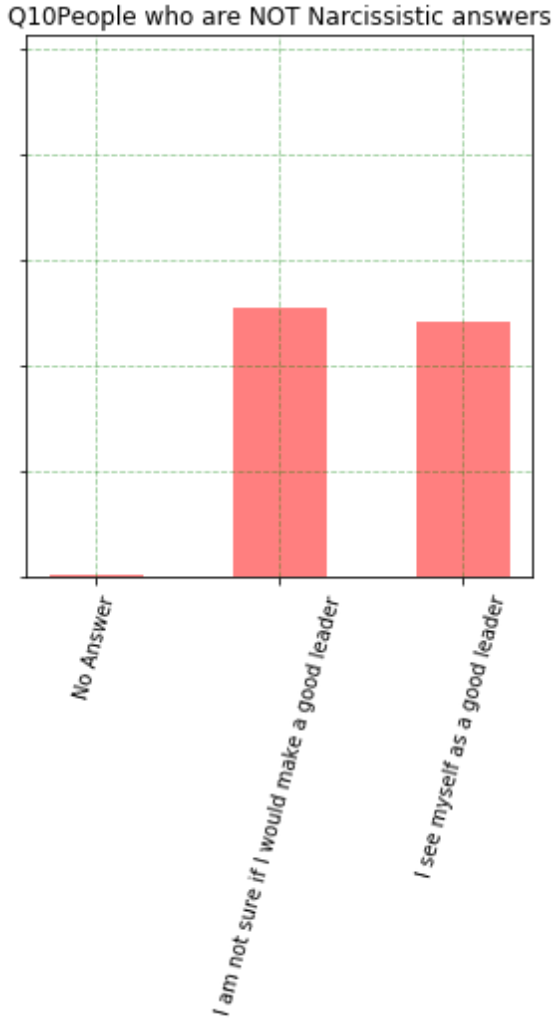

Q21People who are Narcissistic answers
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
No Answer
I always know what I am doing
Sometimes I am not sure of what I am doing

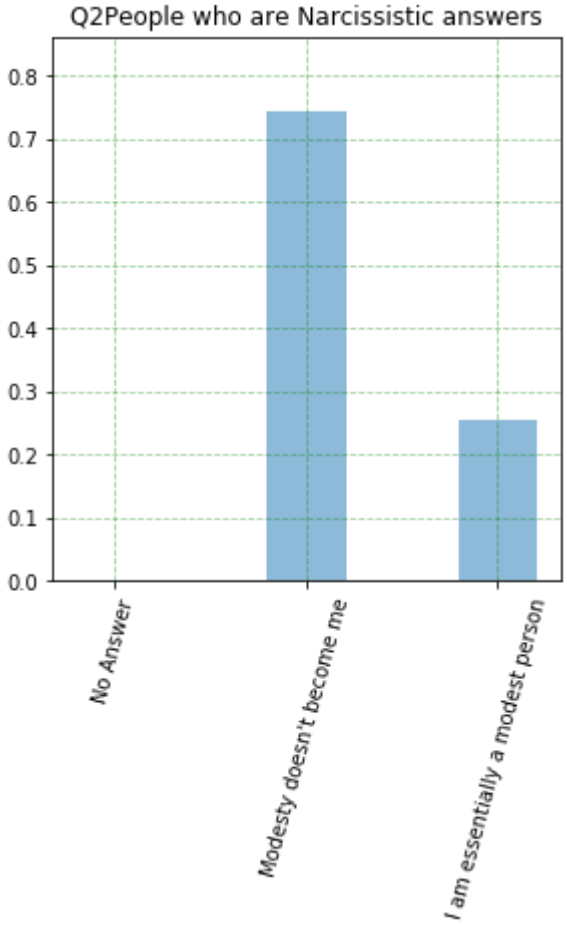

Q21People who are NOT Narcissistic answers
No Answer
I always know what I am doing
Sometimes I am not sure of what I am doing

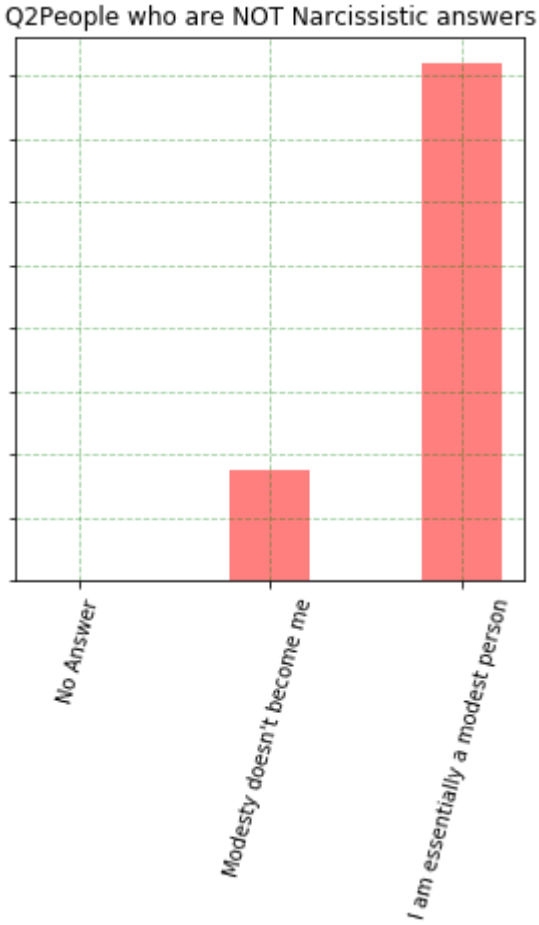

Q16People who are Narcissistic answers
0.8
0.6
0.4
0.2
0.0
No Answer
I can read people like a book
People are sometimes hard to understand

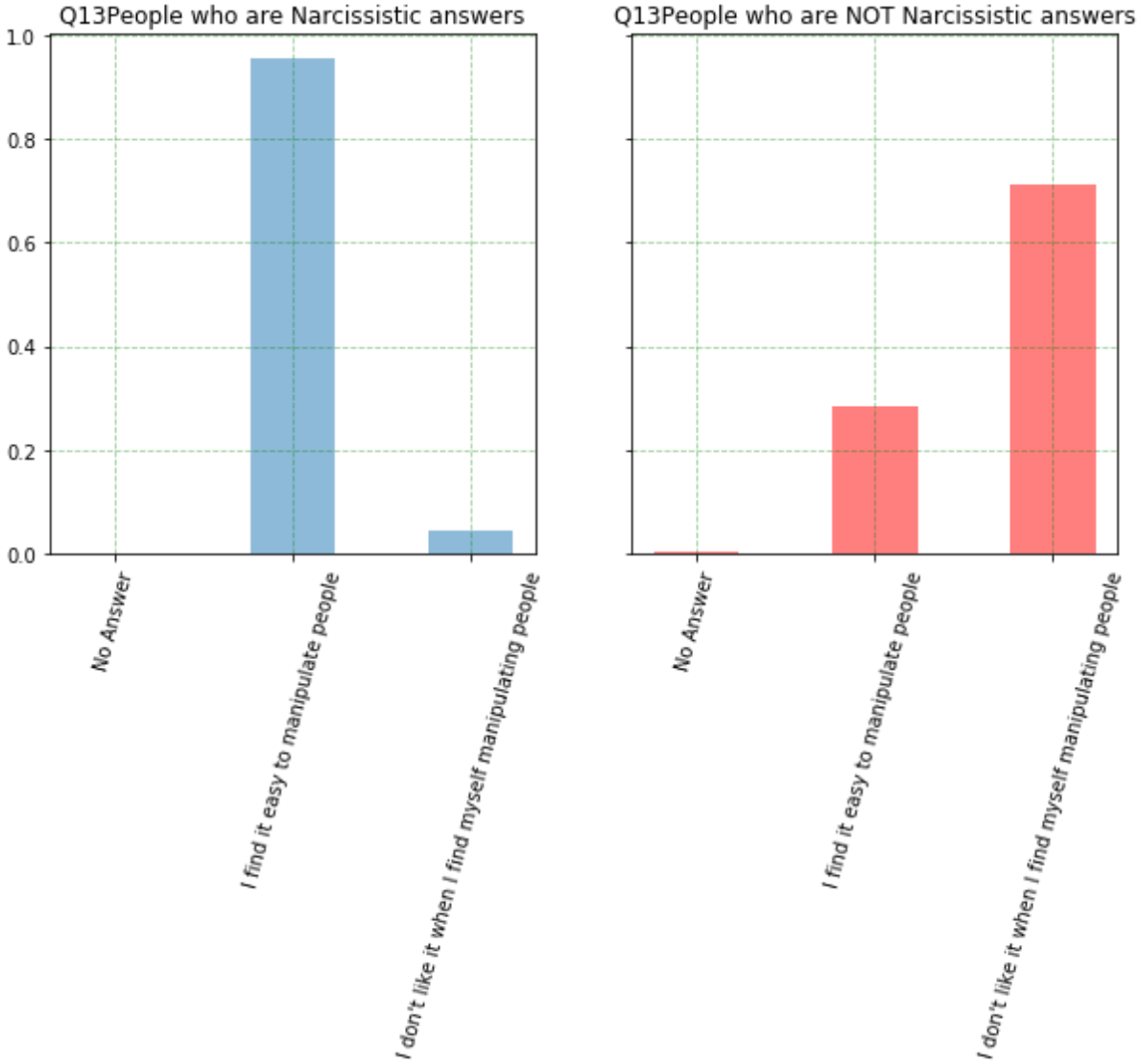

Q16People who are NOT Narcissistic answers
No Answer
I can read people like a book
People are sometimes hard to understand


Q10People who are Narcissistic answers
1.0
0.8
0.6
0.4
0.2
0.0
No Answer
I am not sure if I would make a good leader
I see myself as a good leader


Q10People who are NOT Narcissistic answers
No Answer
I am not sure if I would make a good leader
I see myself as a good leader

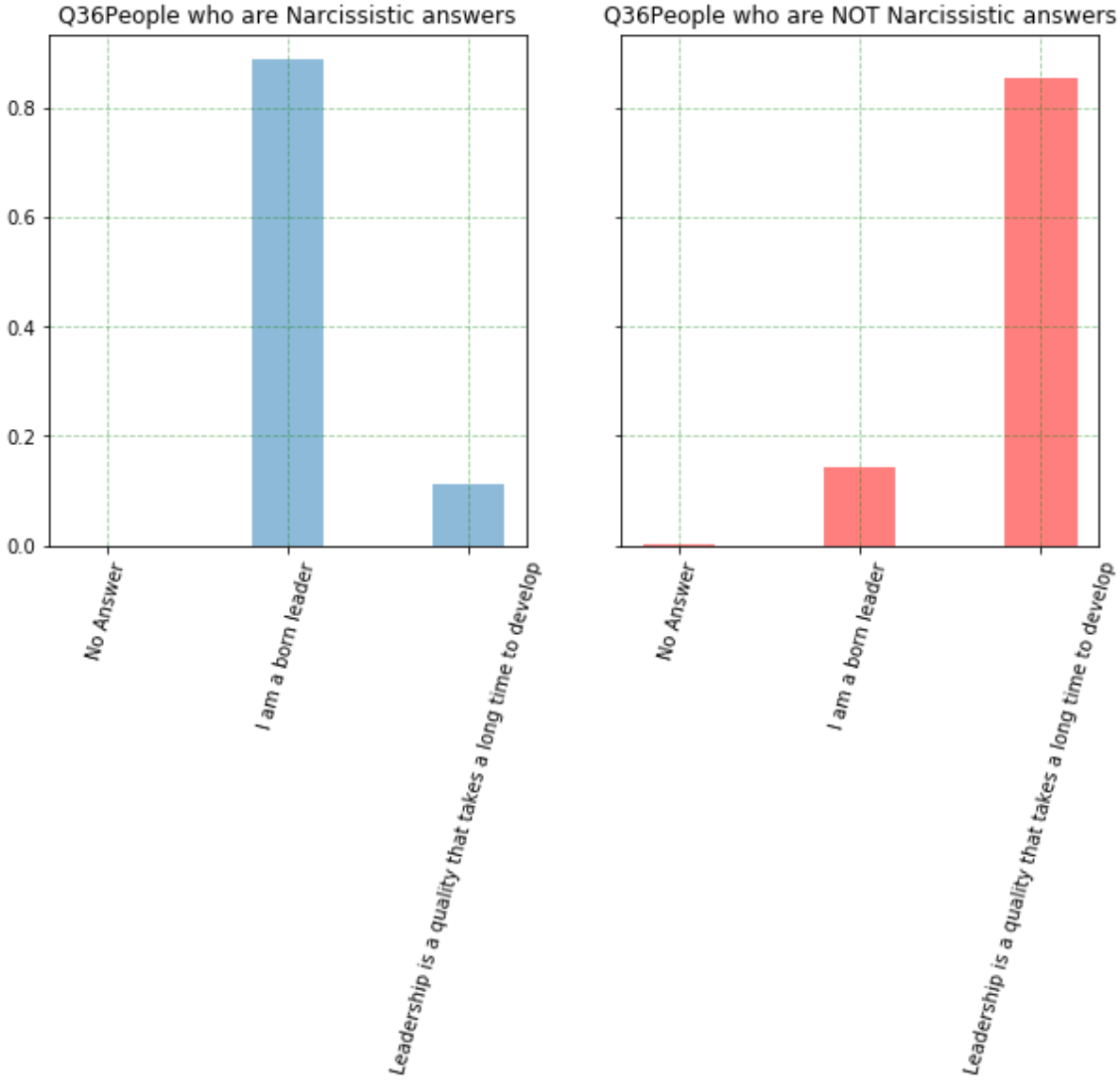

Q2People who are Narcissistic answers
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
No Answer
Modesty doesn't become me
I am essentially a modest person

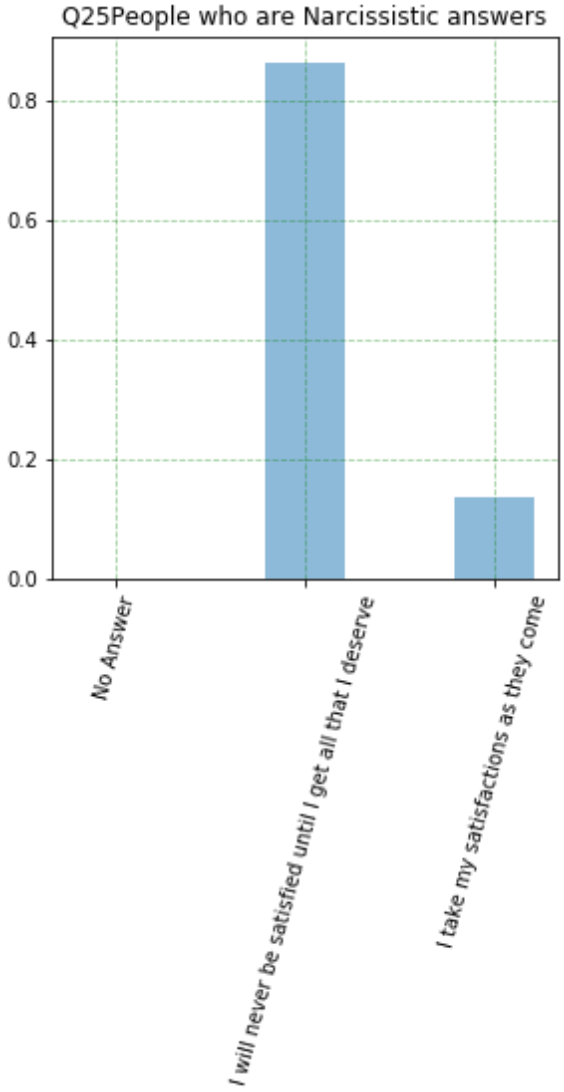

Q2People who are NOT Narcissistic answers
No Answer
Modesty doesn't become me
I am essentially a modest person

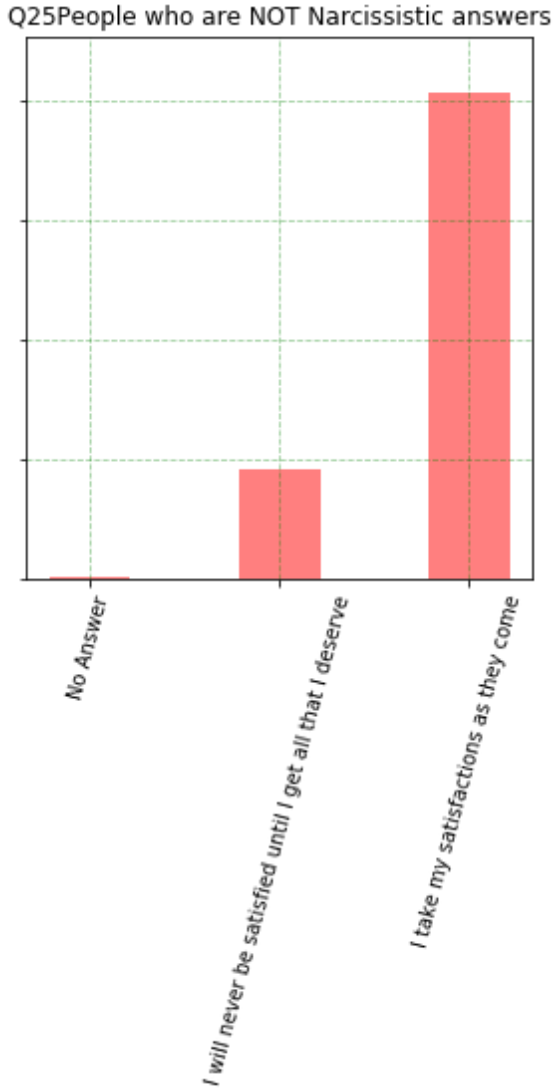


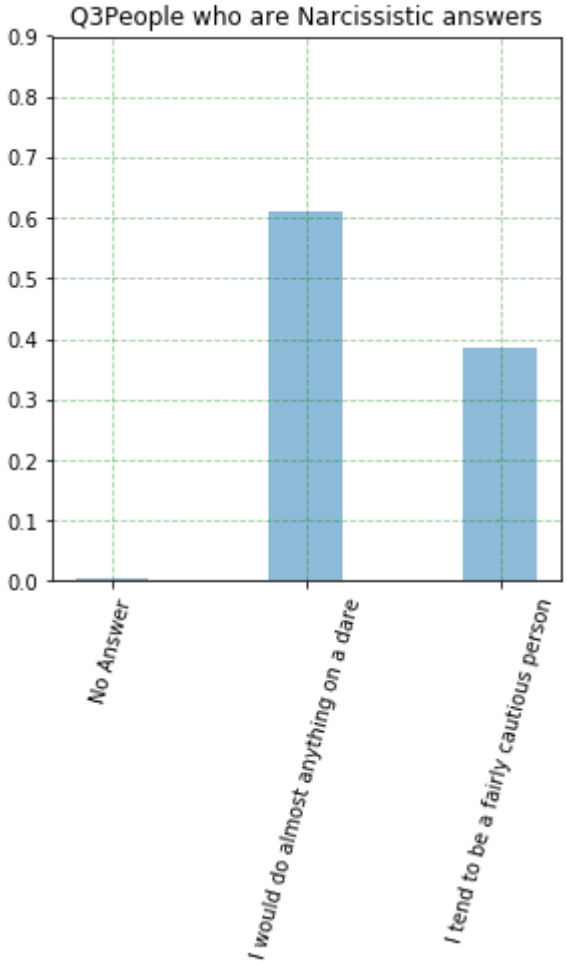
Finally, I can look at the questions that are identified the most throughout all of the principle components

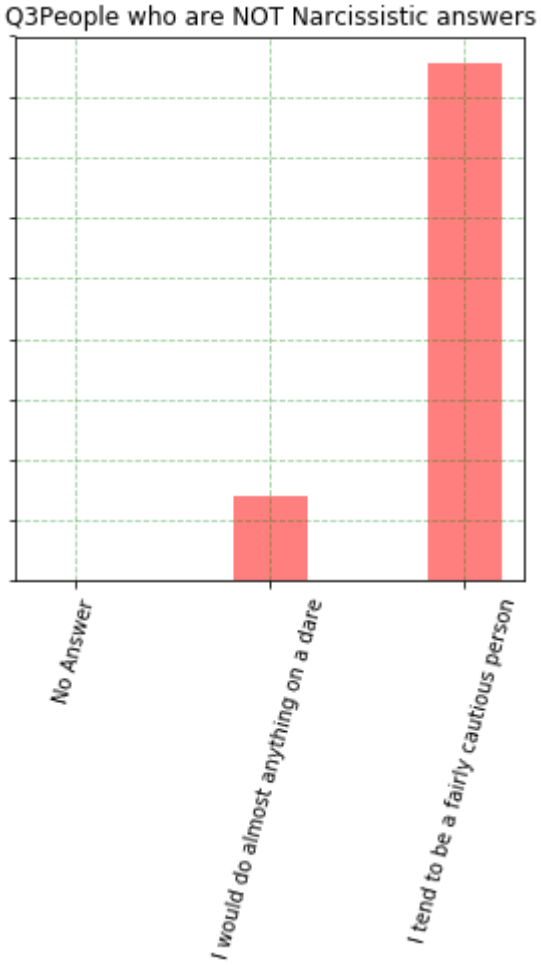
In [25]:
```
components = []
for i in range(5):
    components.append(np.argsort(np.abs(model.components_[i]))[0:10])
from collections import Counter
important_qs = Counter(np.array(components).flatten()).most_common(5)
components = [x[0] for x in important_qs]
plot_top_n(components,sc_df,nar_idx, not_idx,3)
```


Q36People who are Narcissistic answers
0.0
0.2
0.4
0.6
0.8
No Answer
I am a born leader
Leadership is a quality that takes a long time to develop
Q36People who are NOT Narcissistic answers
No Answer
I am a born leader
Leadership is a quality that takes a long time to develop


Q25People who are Narcissistic answers
0.0
0.2
0.4
0.6
0.8
No Answer
I will never be satisfied until I get all that I deserve
I take my satisfactions as they come


Q25People who are NOT Narcissistic answers
No Answer
I will never be satisfied until I get all that I deserve
I take my satisfactions as they come


Q3People who are Narcissistic answers
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
No Answer
I would do almost anything on a dare
I tend to be a fairly cautious person


Q3People who are NOT Narcissistic answers
No Answer
I would do almost anything on a dare
I tend to be a fairly cautious person