In [1]:
```
from IPython.core.display import HTML, Markdown, display

import numpy.random as npr
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
import statsmodels.formula.api as smf
import pingouin as pg

import ipywidgets as widgets

# Enable plots inside the Jupyter Notebook
%matplotlib inline
```

Differences between means (review)

Authored by *Todd Gureckis* with input from *Matt Crump*.

Exercise 1: Bootstrapping the t-distribution

Exercise 1

Create a plot of the sampling distribution of the t statistic for 10,000 random normal samples of size 6 and 500.

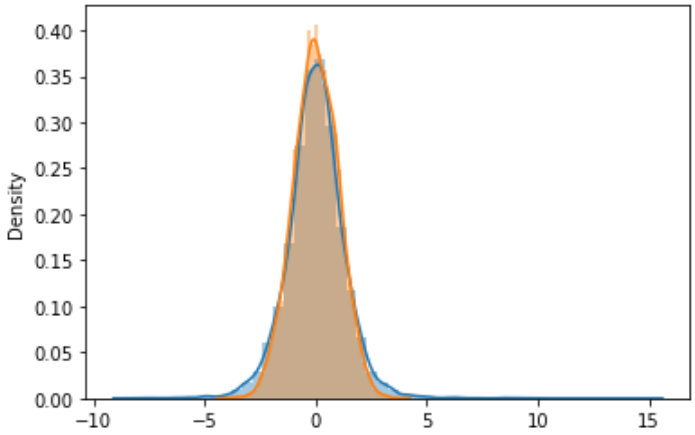
In [2]:
```
ts=[]
for _ in range(10000): # repeat 10000 times
    r_sample = np.random.normal(0,1,size=6)
    sem = np.std(r_sample,ddof=1)/np.sqrt(len(r_sample))
    t_stat = np.mean(r_sample)/sem
    ts.append(t_stat)

ts2=[]
for _ in range(10000):
    r_sample = np.random.normal(0,1,size=500)
    sem = np.std(r_sample,ddof=1)/np.sqrt(len(r_sample))
    t_stat = np.mean(r_sample)/sem
    ts2.append(t_stat)

sns.distplot(ts)
sns.distplot(ts2)
```

```
/Users/peilingjiang/Documents/cognition_env/lib/python3.8/site-packages/seaborn/distributio
ns.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/Users/peilingjiang/Documents/cognition_env/lib/python3.8/site-packages/seaborn/distributio
ns.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a fut
ure version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[2]:
```
<AxesSubplot:ylabel='Density'>
```



Stop and think
Do these distibutions look identical? What is different and why?

Exercise 2: Relationship between p and t values

Exercise 2
Using the following interactive widget, explore the critical value of t shown for different degrees of freedom (sample sizes) and alpha levels. Report in a cell below the critical p values for a t-distribution with 9 degrees of freedom and alpha is 0.05, a t-distribution with 50 degrees of freedom and alpha 0.05 and a the critical value of a t-distribution with 25 degrees of freedom and alpha = 0.4.

In [3]:

```
@widgets.interact(dof=widgets.IntSlider(min=1, max=53, step=1, value=10), alpha=widgets.Flo
def plot_t_onsided(dof, alpha):
    fix, ax = plt.subplots(1,1,figsize=(10,6))

    x=np.linspace(-3.5,3.5,100)
    y=stats.t.pdf(x,df=dof)
    t_crit=stats.t.ppf(1.0-alpha, df=dof)
    print(t_crit)
    ax.plot(x,y)
    ax.set_ylabel("probability")
    ax.set_xlabel("value of t statistic")
    ax.set_title("One Sided Test")
    ax.fill_between(x,y,where=x>t_crit,interpolate=True,facecolor='lightblue',alpha=0.2,hat
    ax.set_xticks([0, t_crit])
    #ax.set_yticklabels([])


    sns.despine(top=True, right=True, left=True)

    plt.show()
```

$$t_{\text{crit}} = 1.8331(\text{df} = 9,\ \alpha = 0.05)$$

$$t_{\text{crit}} = 0.2561(\text{df} = 25,\ \alpha = 0.40)$$

Stop and think
How does the critical p-value change based on sample size?

As sample size increases, the critical p-value decreases. Also, as α value increases, the critical p-value also

decreases.

Exercise 3: Computing a one sample t-test by hand

Exercise 3
The following cell defines a small dataset as a numpy array. Compute the t-value for this array under the null hypothesis that the true mean is 0.25. You will find the functions np.mean(), np.std(), and np.sqrt() useful. Print the t-value out in a cell by itself. Then use the stats.t.cdf() function to compute the p-value associated with that t using a one sided test.

In [4]:
```
scores=np.array([.5,.56,.76,.8,.9]) # here is your data
# compute the "effect" (i.e., difference between the mean of the values and the null hypoth
effect = np.mean(scores) - 0.25
# compute the error (i.e., the standard error of the mean)
error = np.std(scores, ddof=1) / np.sqrt(len(scores))
# Pay attention to the degrees of freedom!!
# compute the t-value
t = effect / error
print('t = ', t)
# use stats.t.cdf() to compute the area in the dail of the correct t-distribution for a one
p = 1 - stats.t.cdf(t, df=len(scores) - 1)
print('p = ', p)
```

```
t =  6.03672166714376
p =  0.0018983152219906874
```

Stop and think
If you have trouble try Googling these functions to find information about the arguments!

Exercise 4: Using `pingouin` to do a on sample `ttest()`

Exercise 4
The following cell shows how to do a one sample t-test from a numpy array. Repeat this for the null hypothesis of 0.25, 0.5, and 0.75. All tests should be one-sided. Write one sentence below each t-test to describe how you would report the test in a paper.

In [5]:
```
import pingouin as pg
scores=np.array([.5,.56,.76,.8,.9])
pg.ttest(x=scores, y=0.25, tail='one-sided')
```

Out[5]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	6.036722	4	greater	0.001898	[0.54, inf]	2.699704	28.44	0.999169

The scores we obtained have a extremely significantly higher mean than the population mean of 0.25, $t(4) = 6.04$, one-tailed $p < 0.01$. Thus, we reject the null hypothesis that the scores were drawn from that population.

In [6]:
```
pg.ttest(x=scores, y=0.5, tail='one-sided')
```

Out[6]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	2.712536	4	greater	0.026699	[0.54, inf]	1.213083	4.235	0.719802

The scores we obtained have a significantly higher mean than the population mean of 0.5, $t(4) = 2.71$, one-tailed $p < 0.05$. Thus, we reject the null hypothesis that the scores were drawn from that population.

In [7]:
```
pg.ttest(x=scores, y=0.75, tail='one-sided')
```

Out[7]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	-0.61165	4	less	0.286913	[-inf, 0.86]	0.273538	0.923	0.12922

The mean of the scores we obtained is not significantly different from the population mean of 0.75, $t(4) = 0.61$, one-tailed $p > 0.05$. Thus, we fail to reject the null hypothesis that the scores were drawn from that population.

Exercise 5: Melting data

Exercise 5
The cell below sets up a simple pandas dataframe like the one in my slides. Is this data frame tidy or wide?

In [8]:
```
exp_df = pd.DataFrame({"subjects": [1,2,3,4,5], "level_A": [1,4,3,6,5], "level_B": [4,8,7,9
exp_df
```

Out[8]:

	subjects	level_A	level_B
0	1	1	4
1	2	4	8
2	3	3	7
3	4	6	9
4	5	5	10

The pandas $me < ()$ function convert from a wide representation to the tidy representation. Try playing with different settings of the id_vars, var_name, and value_name.

In [9]:
```
tidy_exp_df=exp_df.melt(id_vars='subjects', var_name='level', value_name='test_value')
tidy_exp_df
```

Out[9]:

	subjects	level	test_value
0	1	level_A	1
1	2	level_A	4
2	3	level_A	3
3	4	level_A	6
4	5	level_A	5
5	1	level_B	4
6	2	level_B	8
7	3	level_B	7
8	4	level_B	9
9	5	level_B	10

The pandas $\pi vot()$ function convert from a tidy representation to the wide representation. Try playing with different settings of the index, columns, and values to see what they do.

In [10]:
```
tidy_exp_df.pivot(index='subjects',columns='level', values='test_value').reset_index()
```

Out[10]:

level	subjects	level_A	level_B
0	1	1	4
1	2	4	8
2	3	3	7
3	4	6	9
4	5	5	10

Using the wide format add a new column to the dataframe called 'differences' that is the difference between level B and level A for each subject. Show this new dataframe below.

In [11]:
```
exp_df['differences'] = exp_df['level_A'] - exp_df['level_B']
exp_df
```

Out[11]:

	subjects	level_A	level_B	differences
0	1	1	4	-3
1	2	4	8	-4
2	3	3	7	-4
3	4	6	9	-3
4	5	5	10	-5

Follow along with the steps in the lecture to compute the paired t-test.

In [12]:
```
pg.ttest(x=exp_df['differences'], y=0, tail='two-sided')
```

Out[12]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	-10.155927	4	two-sided	0.000529	[-4.84, -2.76]	4.541869	60.047	1.0

In [13]:
```
pg.ttest(x=exp_df['level_A'], y=exp_df['level_B'], paired=True, tail='two-sided')
```

Out[13]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	-10.155927	4	two-sided	0.000529	[-4.84, -2.76]	1.791337	60.047	0.84422

Exercise 6: Paired t-test example

STUDY DESCRIPTION

Parents often sing to their children and, even as infants, children listen to and look at their parents while they are singing. Research by Mehr, Song, and Spelke (2016) sought to explore the psychological function that

music has for parents and infants, by examining the hypothesis that particular melodies convey important social information to infants. Specifically, melodies convey information about social affiliation.

The authors argue that melodies are shared within social groups. Whereas children growing up in one culture may be exposed to certain songs as infants (e.g., "Rock-a-bye Baby"), children growing up in other cultures (or even other groups within a culture) may be exposed to different songs. Thus, when a novel person (someone who the infant has never seen before) sings a familiar song, it may signal to the infant that this new person is a member of their social group.

To test this hypothesis, the researchers recruited 32 infants and their parents to complete an experiment. During their first visit to the lab, the parents were taught a new lullaby (one that neither they nor their infants had heard before). The experimenters asked the parents to sing the new lullaby to their child every day for the next 1-2 weeks.

Following this 1-2 week exposure period, the parents and their infant returned to the lab to complete the experimental portion of the study. Infants were first shown a screen with side-by-side videos of two unfamiliar people, each of whom were silently smiling and looking at the infant.The researchers recorded the looking behavior (or gaze) of the infants during this 'baseline' phase. Next, one by one, the two unfamiliar people on the screen sang either the lullaby that the parents learned or a different lullaby (that had the same lyrics and rhythm, but a different melody). Finally, the infants saw the same silent video used at baseline, and the researchers again recorded the looking behavior of the infants during this 'test' phase.For more details on the experiment's methods, please refer to Mehr et al. (2016) Experiment 1.

The first thing to do is download the .csv formatted data file, using the link above, or just click here.

In [14]:
```
# get the baby data frame
baby_df = pd.read_csv('http://gureckislab.org/courses/fall19/labincp/data/MehrSongSpelke201
# filter to only have the data from experiment 1
experiment_one_df = baby_df[baby_df['exp1']==1]
experiment_one_df.head()
```

Out[14]:

	id	study_code	exp1	exp2	exp3	exp4	exp5	dob	dot1	dot2	...	dtword13	dtnowc
0	101	"LUL"	1	0	0			09oct2012	29mar2013	05apr2013	...	0	
1	102	"LUL"	1	0	0			16nov2012	10may2013	17may2013	...	0	
2	103	"LUL"	1	0	0			26nov2012	11may2013	20may2013	...	0	
3	104	"LUL"	1	0	0			19nov2012	11may2013	18may2013	...	0	
4	105	"LUL"	1	0	0			29nov2012	15may2013	29may2013	...	0	

5 rows × 153 columns

Baseline phase: Conduct a one sample t-test

You first want to show that infants' looking behavior did not differ from chance during the baseline trial. The baseline trial was 16 seconds long. During the baseline, infants watched a video of two unfamiliar people, one of the left and one on the right. There was no sound during the baseline. Both of the actors in the video smiled directly at the infant.

The important question was to determine whether the infant looked more or less to either person. If they showed no preference, the infant should look at both people about 50% of the time. How could we determine whether the infant looked at both people about 50% of the time?

The `experiment_one_df` data frame has a column called `Baseline_Proportion_Gaze_to_Singer`. All of these values show how the proportion of time that the infant looked to the person who would later sing the familiar song to them. If the average of these proportion is .5 across the infants, then we would have some evidence that the infants were not biased at the beginning of the experiment. However, if the infants on average had a bias toward the singer, then the average proportion of the looking time should be different than .5.

Using a one-sample t-test, we can test the hypothesis that our sample mean for the `Baseline_Proportion_Gaze_to_Singer` was not different from .5.

Exercise 6
The cell below shows how to get the looking time proportions from the baseline phase. Conduct a one-sample t-test using pinguoin to see if this data is different than a null hypothesis of 0.5 (no preference).

In [15]:
```
# here is how to get the column
experiment_one_df['Baseline_Proportion_Gaze_to_Singer']
```

Out[15]:
```
0     0.437126
1     0.412533
2     0.754491
3     0.438878
4     0.474645
5     0.870902
6     0.236715
7     0.759259
8     0.416335
9     0.799534
10    0.378677
11    0.697892
12    0.593407
13    0.614907
14    0.614907
15    0.316832
16    0.310417
17    0.504367
18    0.469340
19    0.504082
20    0.564033
21    0.256637
22    0.700000
23    0.382353
24    0.371859
25    0.284464
26    0.767816
27    0.473786
28    0.821218
29    0.590164
30    0.422037
31    0.435484
Name: Baseline_Proportion_Gaze_to_Singer, dtype: float64
```

In [16]:
```
pg.ttest(x=experiment_one_df['Baseline_Proportion_Gaze_to_Singer'], y=0.5, tail='two-sided'
```

Out[16]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	0.674375	31	two-sided	0.505071	[0.46, 0.58]	0.119214	0.233	0.100209

The infants' baseline proportion gaze to singers is not significantly different from 0.5, $t(31) = 0.67$, $p > 0.05$. Thus, we conclude that infants show no initial preference on the woman captured in the testing video.

Remember how the experiment went. Infants watched silent video recordings of two women (Baseline). Then

each person sung a song, one was familiar to the infant (their parents sung the song to them many times), and one was unfamiliar (singing phase). After the singing phase, the infants watched the silent video of the two singers again (test phase). The critical question was whether the infants would look more to the person who sung the familiar song compared to the person who sun the unfamiliar song. If the infants did this, they should look more than 50% of the time to the singer who sang the familiar song. We have the data, we can do another one sample t-test to find out.

The cell below shows how to get the looking time proportions from the test phase. First conduct a one sample t-test on this test data compared to a null hypothesis of 0.5. What do you find? Finally, conduct a paired t-test between the baseline and test phase using pinguoin to see if this data is different than a null hypothesis of 0.0 (no difference).

In [17]:
```
# here is how to get the column
experiment_one_df['Test_Proportion_Gaze_to_Singer']
```

Out[17]:
```
0     0.602740
1     0.683027
2     0.724138
3     0.281654
4     0.498542
5     0.950920
6     0.417755
7     0.938202
8     0.500000
9     0.586294
10    0.472623
11    0.508380
12    0.811189
13    0.571801
14    0.777448
15    0.262846
16    0.507937
17    0.436975
18    0.542105
19    0.600897
20    0.418675
21    0.789474
22    0.760108
23    0.623894
24    0.366412
25    0.461539
26    0.899521
27    0.531100
28    0.541899
29    0.700389
30    0.762963
31    0.460274
Name: Test_Proportion_Gaze_to_Singer, dtype: float64
```

In [18]:
```
pg.ttest(
    x=experiment_one_df['Baseline_Proportion_Gaze_to_Singer'],
    y=experiment_one_df['Test_Proportion_Gaze_to_Singer'],
    paired=True,
    tail='two-sided'
)
```

Out[18]:

	T	dof	tail	p-val	CI95%	cohen-d	BF10	power
T-test	-2.41643	31	two-sided	0.021753	[-0.13, -0.01]	0.407103	2.297	0.606906

In [19]:
```
experiment_one_df['Test_Proportion_Gaze_to_Singer'].mean()
```

Out[19]: 0.59349125

Alright. What did we find? You should take a stab at writing down what we found. You can use the same kind of language that I used from the first one sample-test. You should state the mean proportion, the t-value, the dfs, and the p-value. You should be able to answer the question, did the infants look longer at the singer who sang the familiar song? And, did they look longer than would be consist with chance at 50%.

During the testing phase, infants look significantly longer at the singer who sang the familiar song in the singing phase, $M = 0.59$, $t(31) = 2.42$, $p < 0.05$. Thus, we conclude that singing familiar melodies convey important social information to infants.