Setup

```
In [1]: # Import some useful functions
from numpy import *
from numpy.random import *
from datascience import *
from statsmodels.formula.api import *

# Define some useful functions
def correlation(array_1, array_2):
    return corrcoef(array_1, array_2).item(1)

# Customize look of graphics
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
plt.rcParams['figure.dpi'] = 60
%matplotlib inline

# Force display of all values
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

# Handle some obnoxious warning messages
import warnings
warnings.filterwarnings("ignore")
```

Sales Team Competition

Business Decision

A sales organization has two sales teams. Team #1 comprises 5 representatives and Team #2 comprises 4 representatives. Recently, Team #2 touted that its performance is better because the individual representatives on Team #2 performed better on average than those on Team #1 did.

Data

Sales Team #1 representatives makes sales of these amounts (in $): 1000000, 1500000, 1300000, 1200000, 1600000

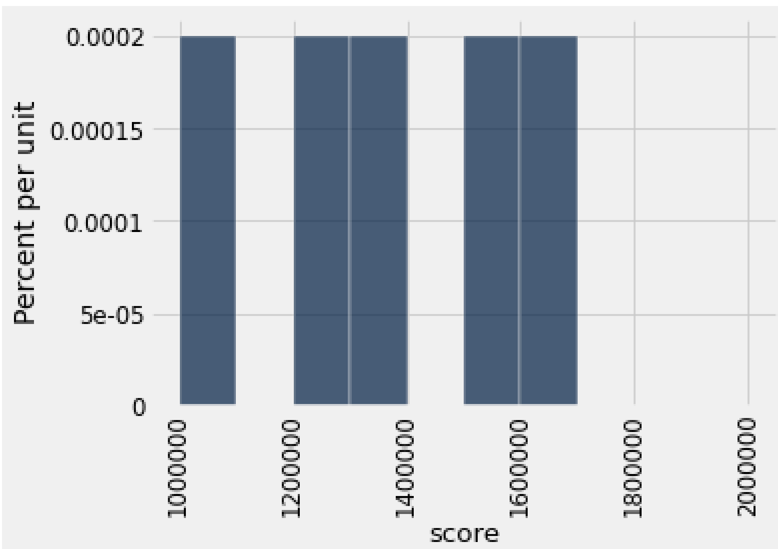
Show the sample size, sample mean, sample standard deviation, and a histogram of the sampled sales (10 bins range 1 million to 2 million)

In [2]:
```
sample_1 = Table().with_column('score', make_array(1000000, 1500000, 1300

size_1 = sample_1.num_rows
mean_1 = mean(sample_1.column('score'))
std_1  = std(sample_1.column('score'))

size_1
mean_1
std_1
sample_1.hist(bins=10, range=make_array(1000000,2000000))
```

Out[2]: 5

Out[2]: 1320000.0

Out[2]: 213541.56504062621



Sales Team #2 representatives make sales of these amounts (in $): 1300000, 1200000, 1700000, 1500000

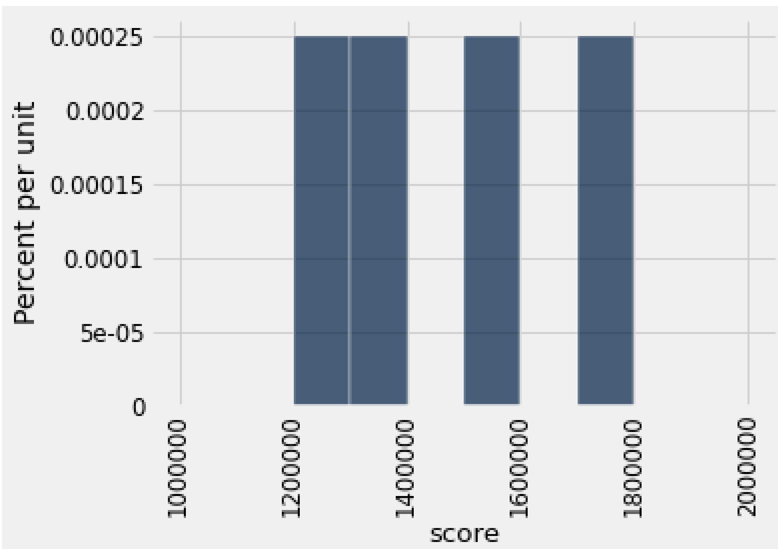
Show the sample size, sample mean, sample standard deviation, and a histogram of the sampled sales (10 bins, range 1 million to 2 million).

In [3]:
```
sample_2 = Table().with_column('score', make_array(1300000, 1200000, 170(

size_2 = sample_2.num_rows
mean_2 = mean(sample_2.column('score'))
std_2  = std(sample_2.column('score'))

size_2
mean_2
std_2
sample_2.hist(bins=10, range=make_array(1000000,2000000))
```

Out[3]: 4

Out[3]: 1425000.0

Out[3]: 192028.64369671521



Analysis

Hypothesize that the difference between population means (pop_mean_1 - pop_mean_2) is ≥ 0. The alternative to this hypothesis is that the difference between population means is < 0.

In [4]:
```
pop_mean_diff_hypo = 0
pop_mean_diff_hypo
```

Out[4]: 0

Calculate and show the two-sample t @ population mean difference 0.

In [5]:
```
two_sample_t = ( (mean_1 - mean_2) - pop_mean_diff_hypo) / sqrt( (std_1*
two_sample_t
```

Out[5]: -0.775361859536156

Calculate and show the degrees of freedom for two samples.

In [6]:
```
df = ((std_1**2 / size_1) + (std_2**2 / size_2))**2 / \
     ((1/(size_1-1))*(std_1**2/size_1)**2 + (1/(size_2-1))*(std_2**2/siz
df
```

Out[6]: 6.846411092302966

Get 1,000,000 values from the standard t distribution for the appropriate degrees of freedom.

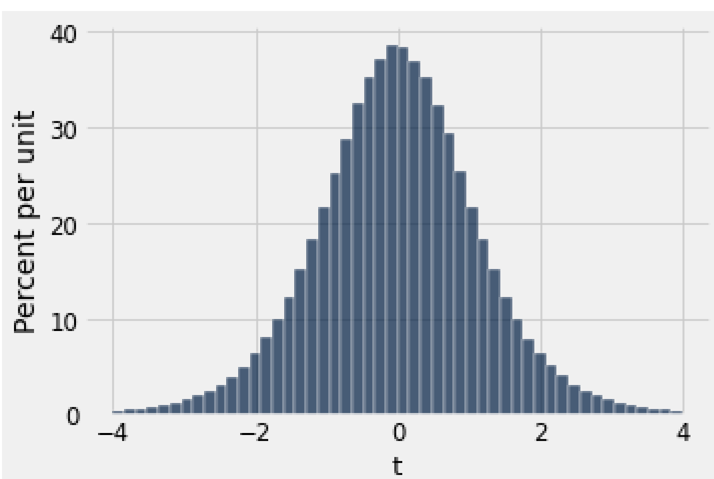
Show a few of the values and a histogram of all the values (50 bins, range -4 to 4).

In [36]:
```
dist_array = standard_t(df, 1000000)
dist = Table().with_column('t', dist_array)

dist
dist.hist(bins=50, range=make_array(-4,4))
```

Out[36]:

t
0.637656
0.278696
0.053153
1.51989
-0.566573
0.935802
-0.0582546
-0.70579
-1.21826
0.248165

... (999990 rows omitted)



Calculate and show the probability that the two-sample t ≤ -0.775 when pop_mean_1 - pop_mean_2 ≥ 0 (this is the p-value).

In [24]:
```
p_value = dist.where('t', are.below_or_equal_to(two_sample_t)).num_rows /
p_value
```

Out[24]: 0.232048

Calculate and show the critical value at significance level 0.05. Note that the suspected difference between population means is LOWER than the hypothesized difference between population means (0).

In [37]:
```
sig_level = 0.05
cv = percentile(sig_level*100, dist.column('t'))
sig_level
cv
```

Out[37]: 0.05

Out[37]: -1.9013319445120884

Calculate and show what you should assume about the hypothesis, at significance level 0.05.

In [16]:
```
p_value > sig_level
two_sample_t > cv
```

Out[16]: True

Out[16]: True

- No, they've just been lucky. The small two-sample t indicates that Team #2 sales on average must be statistically significantly equivalent to Team #1 sales on average.
- No, they've just been lucky. The negative critical value indicates that the Team #2 sales

Breakfast Cereal Focus Groups

Business Decision

A breakfast cereal manufacturer wants to evaluate 5 new versions of a children's breakfast cereal. It runs some focus groups in which children rate the tastes of the 5 versions. The rating scale is 1 (tastes terrible) to 7 (tastes delicious).

Data

Here are 5 samples of ratings:

- cereal 1 ratings: 3,2,3,3,4,2,4,7,7,4,4,1,4
- cereal 2 ratings: 5,5,5,2,1,2,1,5,2,7,7,1,2
- cereal 3 ratings: 6,4,5,6,5,3,3,5,6,4,5,4,4
- cereal 4 ratings: 7,6,3,4,7,4,3,7,5,5,6,5,5
- cereal 5 ratings: 6,6,6,3,6,5,4,3,6,5,4,4,5

Show the treatment count (number of cereal versions), unit count (number of tastings), and the samples (as 5 arrays).

In [38]:
```
x1 = make_array(3,2,3,3,4,2,4,7,7,4,4,1,4)
x2 = make_array(5,5,5,2,1,2,1,5,2,7,7,1,2)
x3 = make_array(6,4,5,6,5,3,3,5,6,4,5,4,4)
x4 = make_array(7,6,3,4,7,4,3,7,5,5,6,5,5)
x5 = make_array(6,6,6,3,6,5,4,3,6,5,4,4,5)

c = 5
n = 5 * 13

c
n
x1
x2
x3
x4
x5
```

Out[38]: 5

Out[38]: 65

Out[38]: array([3, 2, 3, 3, 4, 2, 4, 7, 7, 4, 4, 1, 4], dtype=int64)

Out[38]: array([5, 5, 5, 2, 1, 2, 1, 5, 2, 7, 7, 1, 2], dtype=int64)

Out[38]: array([6, 4, 5, 6, 5, 3, 3, 5, 6, 4, 5, 4, 4], dtype=int64)

Out[38]: array([7, 6, 3, 4, 7, 4, 3, 7, 5, 5, 6, 5, 5], dtype=int64)

Out[38]: array([6, 6, 6, 3, 6, 5, 4, 3, 6, 5, 4, 4, 5], dtype=int64)

Analysis

Hypothesize that the five breakfast cereal population mean ratings are all the same. The alternative to this hypothesis is that the five cereal population mean ratings are not all the same.

Calculate and show the multi-sample f. Calculate and show the upper and lower degrees of freedom associated with the standard f distribution. Calculate and show the p-value. Also show a histogram of the standard f distribution with area corresponding to the p-value highlighted (50 bins, range 0 to 10).

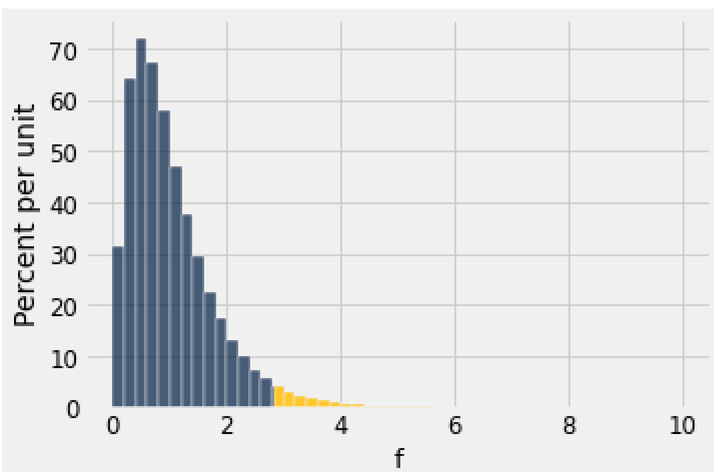
In [53]:
```
en(x1)*mean(x1) + len(x2)*mean(x2) + len(x3)*mean(x3) + len(x4)*mean(x4)
n(x1)*(mean(x1)-mmx)**2 + len(x2)*(mean(x2)-mmx)**2 + len(x3)*(mean(x3)-m
t / (c-1)
m((x1-mean(x1))**2) + sum((x2-mean(x2))**2) + sum((x3-mean(x3))**2) + sum
se / (n-c)
mple_f = mst / mse

 = c-1
 = n-c
ay = f(df_upper, df_lower, 1000000)
able().with_column('f', dist_array)

= dist.where('f', are.above_or_equal_to(multi_sample_f)).num_rows / dist.

mple_f



t(bins=50, range=make_array(0,10), left_end=multi_sample_f, right_end=10)
```

Out[53]: 2.8312883435582812

Out[53]: 4

Out[53]: 60

Out[53]: 0.032118

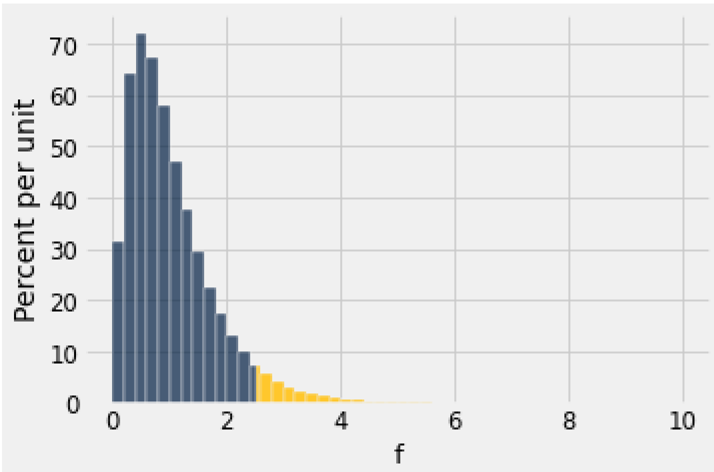


Calculate and show the critical value at significance level 0.05. Also show the significance level and histogram of standard f distribution with the area corresponding to the significance level highlighted.

In [54]:
```
sig_level = 0.05
cv = percentile((1-sig_level)*100, dist.column('f'))

sig_level
cv
dist.hist(bins=50, range=make_array(0,10), left_end=cv, right_end=10)
```

Out[54]: 0.05

Out[54]: 2.5250110904140683



Calculate and show what you should assume about the hypothesis, at significance level 0.05.

In [55]:
```
p_value > sig_level
multi_sample_f < cv
```

Out[55]: False

Out[55]: False

Show the ANOVA table for this analysis.

In [56]:
```
Table().with_columns('source of variation', make_array('between groups',
                     'sum squares', make_array(sst, sse, sst+sse),
                     'df', make_array(df_upper, df_lower, df_upper+df_lo
                     'mean squares', make_array(mst, mse, None),
                     'f', make_array(multi_sample_f, None, None))
```

Out[56]:

source of variation	sum squares	df	mean squares	f
between groups	28.4	4	7.1	2.83129
within groups	150.462	60	2.50769	None
total	178.862	64	None	None