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"

# Hide some unnecessary warning messages
import warnings
warnings.filterwarnings("ignore")
import numpy
numpy.int = numpy.int_
```

Financial Advice

Business Decision

A financial advisor wants to determine the relationship between type of fund and client satisfaction across all its clients. A fund can be made up of either stocks or bonds. Client satisfaction can be high, medium, or low.

Data

Here is a tally of clients reporting various satisfaction levels, grouped by type of fund the client owns:

- 15 clients with stock funds reported high satisfaction.
- 12 clients with stock funds reported medium satisfaction.

- 3 clients with stock funds reported low satisfaction.
- 24 clients with bond funds reported high satisfaction.
- 4 clients with bond funds reported medium satisfaction.
- 2 clients with bond funds reported low satisfaction.

Show the count of each fund type-client satisfaction pair.

In [13]:
```
data_agg = Table().with_columns('fund',  make_array('stocks', 'stocks',
                                'csat',  make_array('high',   'medium',
                                'count', make_array( 15,       12,
data_agg
```

Out[13]:

fund	csat	count
stocks	high	15
stocks	medium	12
stocks	low	3
bonds	high	24
bonds	medium	4
bonds	low	2

Analysis

Calculate and show the count of each fund type.

In [14]:
```
data_agg_fund = data_agg.select('fund', 'count').group('fund', sum)
data_agg_fund
```

Out[14]:

fund	count sum
bonds	30
stocks	30

Calculate and show the count of each client satisfaction level.

In [15]:
```
data_agg_csat = data_agg.select('csat', 'count').group('csat', sum)
data_agg_csat
```

Out[15]:

csat	count sum
high	39
low	5
medium	16

Construct a table that shows fund type, client satisfaction level, count of fund type-client satisfaction level pair, probability of fund type-client satisfaction level pair, count of fund type, and count of client satisfaction level.

In [16]:
```
n = sum(data_agg.column('count'))

data_agg = data_agg.with_column('prob', data_agg.column('count')/n)

data_agg = data_agg.join('fund', data_agg_fund)
data_agg = data_agg.relabeled('count sum', 'count fund')

data_agg = data_agg.join('csat', data_agg_csat)
data_agg = data_agg.relabeled('count sum', 'count csat')

data_agg
```

Out[16]:

csat	fund	count	prob	count fund	count csat
high	bonds	24	0.4	30	39
high	stocks	15	0.25	30	39
low	bonds	2	0.0333333	30	5
low	stocks	3	0.05	30	5
medium	bonds	4	0.0666667	30	16
medium	stocks	12	0.2	30	16

Hypothesize that fund type and client satisfaction level are independent of each other.

In [17]:
```
data_agg = data_agg.with_column('prob fund', data_agg.column('count fund
data_agg = data_agg.with_column('prob csat', data_agg.column('count csat
data_agg = data_agg.with_column('prob hypo', data_agg.column('prob fund'
data_agg
```

Out[17]:

csat	fund	count	prob	count fund	count csat	prob fund	prob csat	prob hypo
high	bonds	24	0.4	30	39	0.5	0.65	0.325
high	stocks	15	0.25	30	39	0.5	0.65	0.325
low	bonds	2	0.0333333	30	5	0.5	0.0833333	0.0416667
low	stocks	3	0.05	30	5	0.5	0.0833333	0.0416667
medium	bonds	4	0.0666667	30	16	0.5	0.266667	0.133333
medium	stocks	12	0.2	30	16	0.5	0.266667	0.133333

Calculate and show the expected count of each fund type-client satisfaction level pair, based on hypothesized probabilities.

In [19]:
```
data_agg = data_agg.with_column('count expected', data_agg.column('prob h
data_agg
```

Out[19]:

csat	fund	count	prob	count fund	count csat	prob fund	prob csat	prob hypo	count expected
high	bonds	24	0.4	30	39	0.5	0.65	0.325	19.5
high	stocks	15	0.25	30	39	0.5	0.65	0.325	19.5
low	bonds	2	0.0333333	30	5	0.5	0.0833333	0.0416667	2.5
low	stocks	3	0.05	30	5	0.5	0.0833333	0.0416667	2.5
medium	bonds	4	0.0666667	30	16	0.5	0.266667	0.133333	8
medium	stocks	12	0.2	30	16	0.5	0.266667	0.133333	8

Calculate and show the sample chi-squared.

In [20]:
```
count = data_agg.column('count')
count_expected = data_agg.column('count expected')

data_agg = data_agg.with_column('rel diff**2', (count - count_expected)**
data_agg

sample_chisquared = sum(data_agg.column('rel diff**2'))
sample_chisquared
```

Out[20]:

csat	fund	count	prob	count fund	count csat	prob fund	prob csat	prob hypo	count expected	rel diff**2
high	bonds	24	0.4	30	39	0.5	0.65	0.325	19.5	1.03846
high	stocks	15	0.25	30	39	0.5	0.65	0.325	19.5	1.03846
low	bonds	2	0.0333333	30	5	0.5	0.0833333	0.0416667	2.5	0.1
low	stocks	3	0.05	30	5	0.5	0.0833333	0.0416667	2.5	0.1
medium	bonds	4	0.0666667	30	16	0.5	0.266667	0.133333	8	2
medium	stocks	12	0.2	30	16	0.5	0.266667	0.133333	8	2

Out[20]:
```
6.2769230769230777
```

Get 1,000,000 values from the standard chi-squared distribution for the appropriate degrees of freedom. Show the degrees of freedom, a few of the values, and a histogram of all the values (50 bins, range 0 to 25).

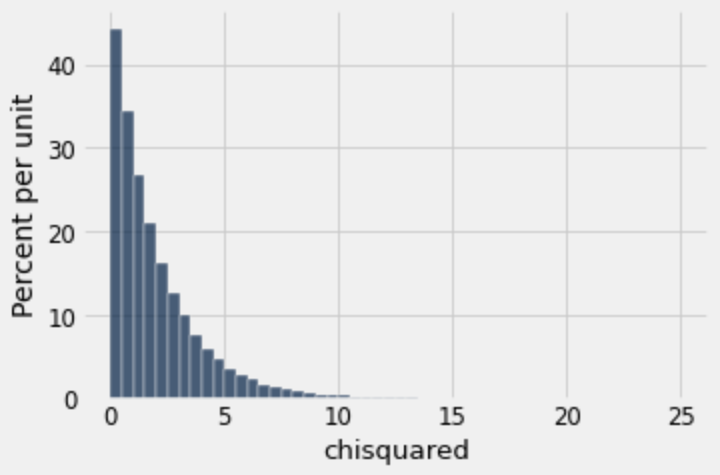
In [21]:
```
r = len(unique(data_agg.column('fund')))
c = len(unique(data_agg.column('csat')))
df = (r-1)*(c-1)
df

dist_array = chisquare(df, 1000000)
dist = Table().with_column('chisquared', dist_array)

dist
dist.hist(bins=50, range=make_array(0,25))
```

Out[21]: 2

Out[21]:

chisquared
0.0118468
6.41271
0.225082
0.919671
4.43848
2.32522
0.0672102
5.04554
5.13512
3.02556

... (999990 rows omitted)

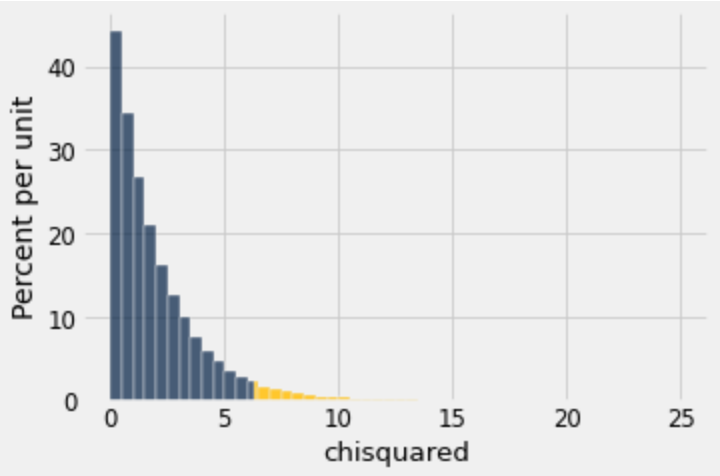


Calculate and show the probability of getting a sample with the sample chi-squared (or above), given that the hypothesis is correct (this is the p-value). Also show the sample chi-squared and a histogram of the standard chi-squared distribution with the area corresponding to the probability highlighted.

In [27]:
```
p_value = dist.where('chisquared', are.above_or_equal_to(sample_chisquare

sample_chisquared
p_value
dist.hist(bins=50, range=make_array(0,25), left_end=sample_chisquared, ri
```

Out[27]: 6.2769230769230777

Out[27]: 0.043116

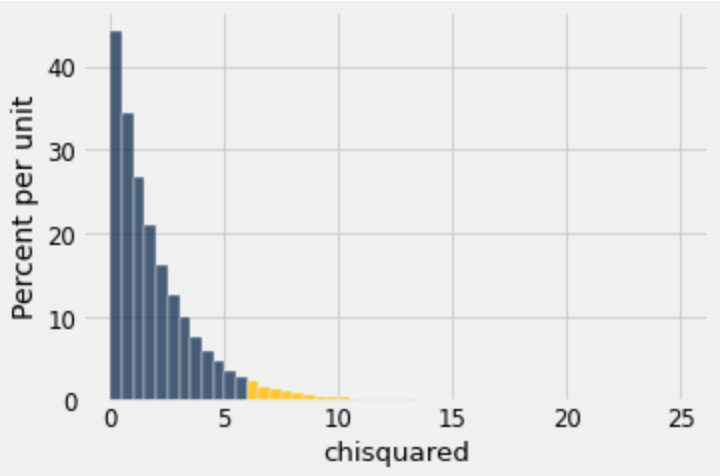


Calculate and show the critical value at significance level 0.05. Also show the significance level and a histogram of the standard chi-squared distribution with the area corresponding to the significance level highlighted.

In [28]:
```
sig_level = 0.05
cv = percentile((1-sig_level)*100, dist.column('chisquared'))

sig_level
cv
dist.hist(bins=50, range=make_array(0,25), left_end=cv, right_end=25)
```

Out[28]: 0.05

Out[28]: 5.9836262402967355



Calculate and show what to conclude about the hypothesis at significance level 0.05.

In [29]:
```
p_value > sig_level
sample_chisquared < cv
```

Out[29]: False

Out[29]: False