12.2) Confidence Intervals

Vitor Kamada

December 2019

Reference

Tables, Graphics, and Figures from

Computational and Inferential Thinking: The Foundations of Data Science

Adhikari & DeNero (2019): Ch 13.3 Confidence Intervals

https://www.inferentialthinking.com/

Stat Labs by Deborah Nolan and Terry Speed

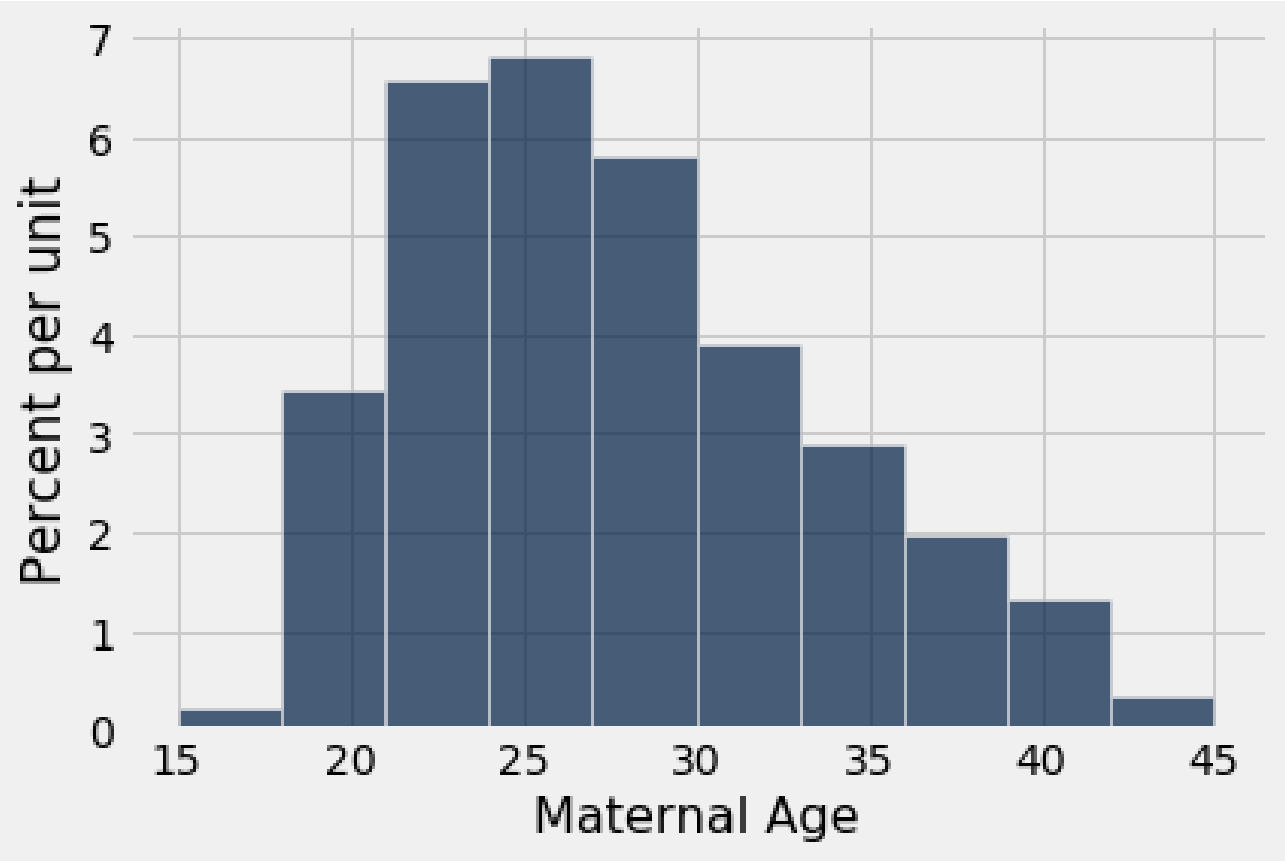
```
from datascience import *
path_data = 'https://github.com/data-8/textbook/raw/gh-pages/data/'
baby = Table.read_table(path_data + 'baby.csv')
```

Maternal Age	Maternal Height	Maternal Pregnancy Weight	Maternal Smoker
27	62	100	False
33	64	135	False
28	64	115	True

```
np.mean(baby.column('Maternal Age'))
```

```
27.228279386712096
```

```
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
baby.select('Maternal Age').hist()
```



Bootstrap Method

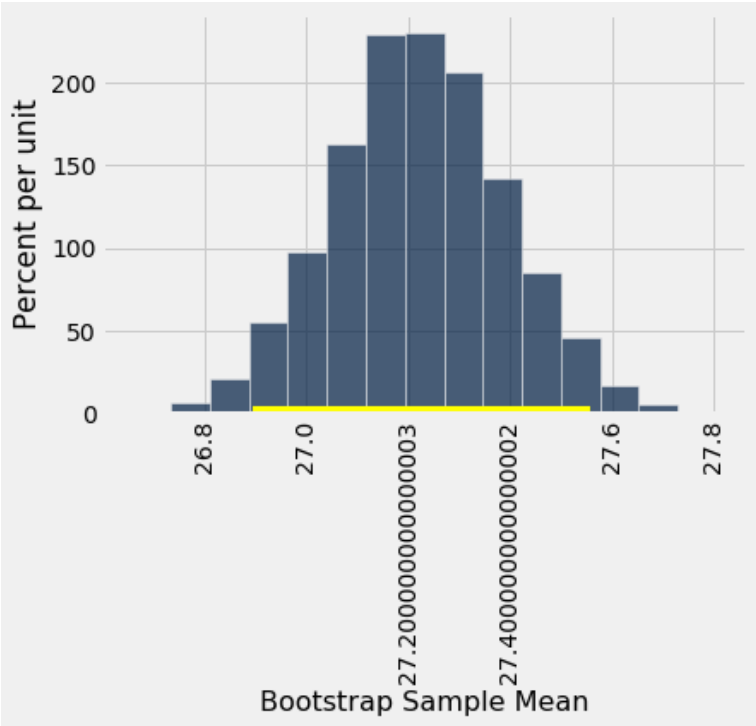
```
def bootstrap_mean(original_sample, label, replications):
    """Returns an array of bootstrapped sample means:
    original_sample: table containing the original sample
    label: label of column containing the variable
    replications: number of bootstrap samples
    """
    just_one_column = original_sample.select(label)
    means = make_array()
    for i in np.arange(replications):
        bootstrap_sample = just_one_column.sample()
        resampled_mean = np.mean(bootstrap_sample.column(0))
        means = np.append(means, resampled_mean)
    return means
```

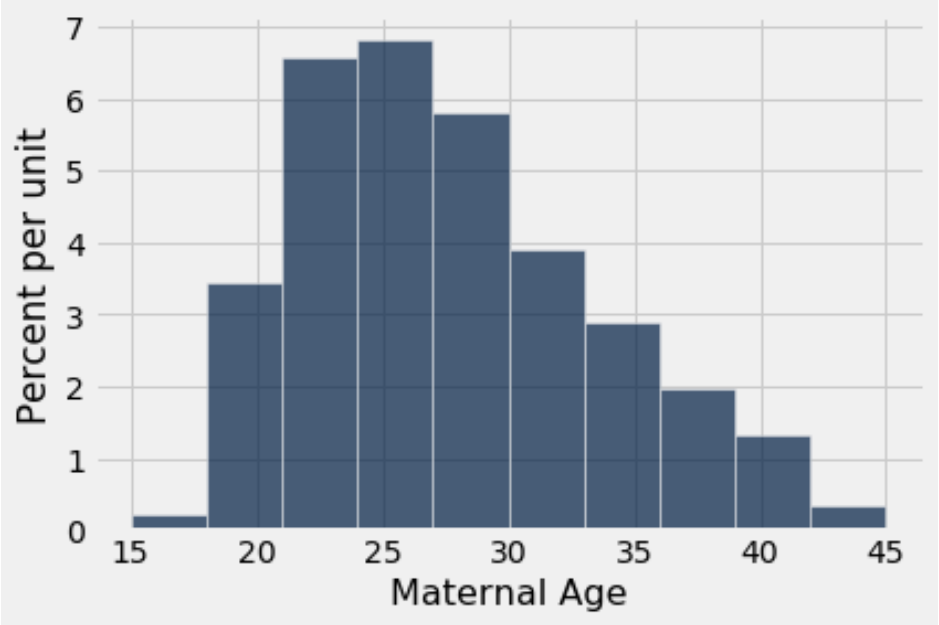
```
bstrap_means = bootstrap_mean(baby, 'Maternal Age', 5000)
left = percentile(2.5, bstrap_means)
right = percentile(97.5, bstrap_means)
make_array(left, right)
```

```
array([26.89437819, 27.55792164])
```

```
resampled_means = Table().with_column(
    'Bootstrap Sample Mean', bstrap_means
)
resampled_means.hist(bins=15)
plots.plot(make_array(left, right),
  make_array(0, 0), color='yellow', lw=8);
```

```
baby.select('Maternal Age').hist()
```

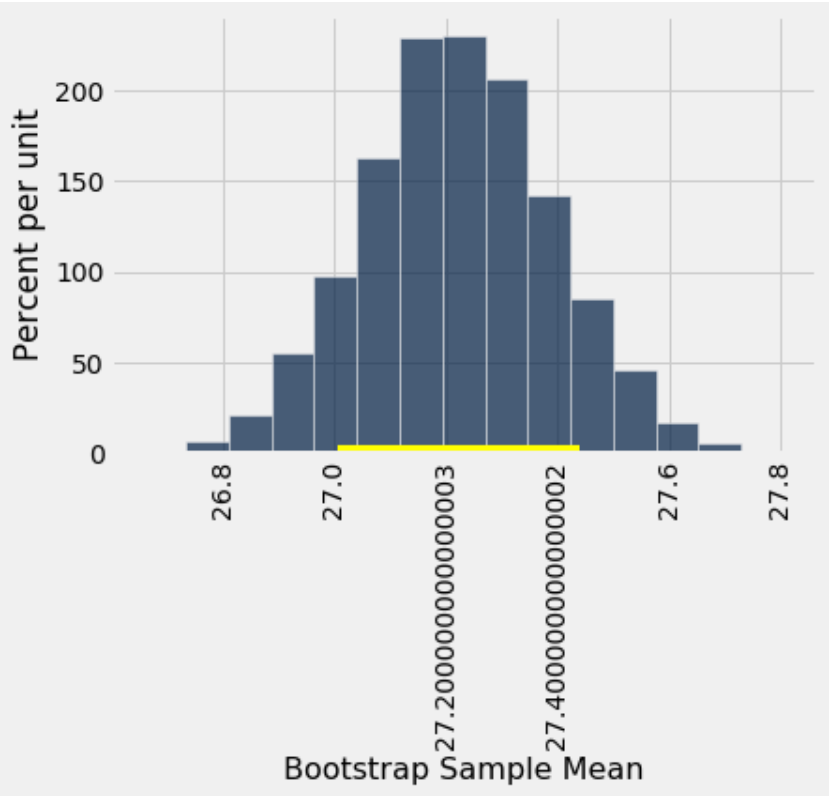




```
left_80 = percentile(10, bstrap_means)
right_80 = percentile(90, bstrap_means)
make_array(left_80, right_80)
```

```
array([27.0076661 , 27.44037479])
```

```
resampled_means.hist(bins=15)
plots.plot(make_array(left_80, right_80),
  make_array(0, 0), color='yellow', lw=8);
```



Bootstrap Percentile Method

```
def bootstrap_proportion(original_sample, label, replications):
    """Returns an array of bootstrapped sample proportions:
    original_sample: table containing the original sample
    label: label of column containing the Boolean variable
    replications: number of bootstrap samples
    """
    just_one_column = original_sample.select(label)
    proportions = make_array()
    for i in np.arange(replications):
        bootstrap_sample = just_one_column.sample()
        resample_array = bootstrap_sample.column(0)
        resampled_proportion = np.count_nonzero\
        (resample_array)/len(resample_array)
        proportions = np.append(proportions, resampled_proportion)
    return proportions
```

Confidence Interval for a Population Proportion

```
bstrap_props = bootstrap_proportion(baby, 'Maternal Smoker', 5000)
left = percentile(2.5, bstrap_props)
right = percentile(97.5, bstrap_props)
make_array(left, right)
```

```
array([0.3637138 , 0.41993186])
```

```
smoking = baby.column('Maternal Smoker')
np.count_nonzero(smoking)/len(smoking)
```

```
0.3909710391822828
```

```
resampled_proportions = Table().with_column(
    'Bootstrap Sample Proportion', bstrap_props
)
resampled_proportions.hist(bins=15)
plots.plot(make_array(left, right),
  make_array(0, 0), color='yellow', lw=8);
```