lec15

October 4, 2021

[1]:
```
from datascience import *
import numpy as np

%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
```

0.1 Lecture 15

0.2 Rules of Probability

Three tickets: R, G, B

Two draws at random without replacement Sample Space is: …

P(first draw is G) = 2/6

Multiplication P(first two draws are GR) = P(first draw is G)*P(second draw is R | first draw is G) = (2/6)*(1/2) = 1/6

Addition P(second draw is G) = P(first draw is R and second is G) + P(first draw is B and second is G) (1/6) + (1/6)

[2]:
```
# P(at least one Head in 3 tosses of coin)
# P(at least one Head in 10 tosses)
# P(Mo and Jo both appear)
# P(neither Mo nor Jo appears)
# see slides
```

0.3 Distributions

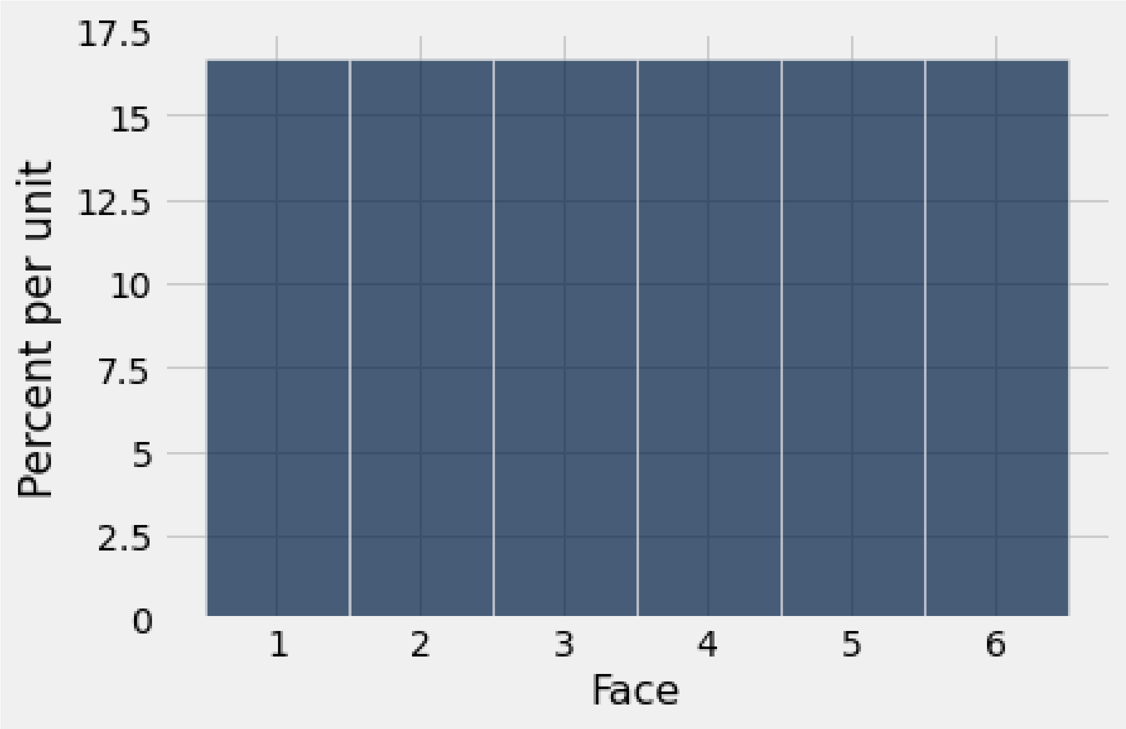
[3]:
```
die = Table().with_column('Face', np.arange(1, 7))
die
```

[3]:
```
Face
1
2
3
4
5
```

```
6
```

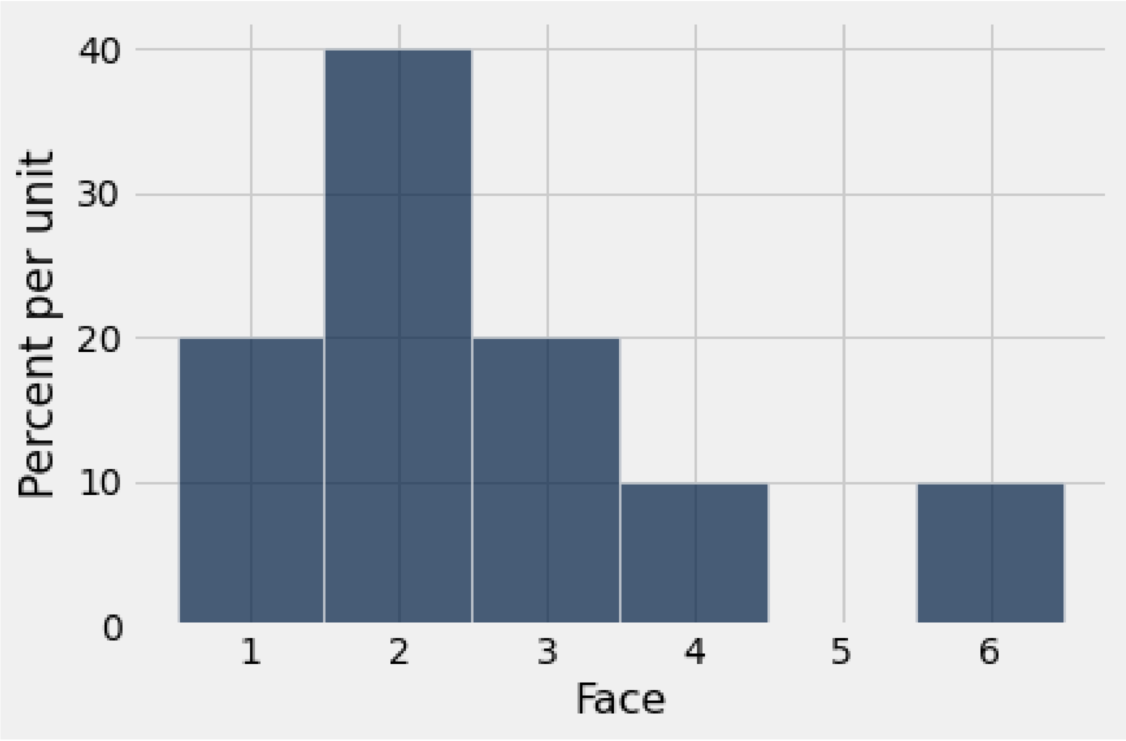
```
[4]: roll_bins = np.arange(0.5, 6.6, 1)
```

```
[5]: die.hist(bins = roll_bins)
```

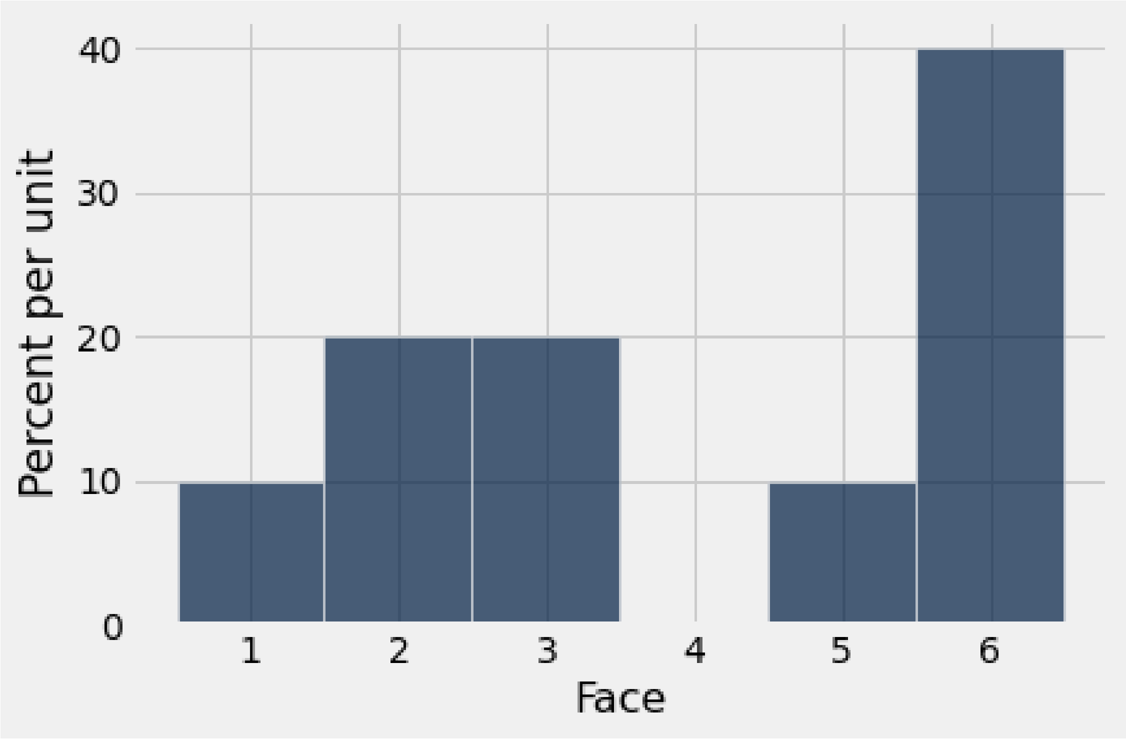


0.4 Random Samples

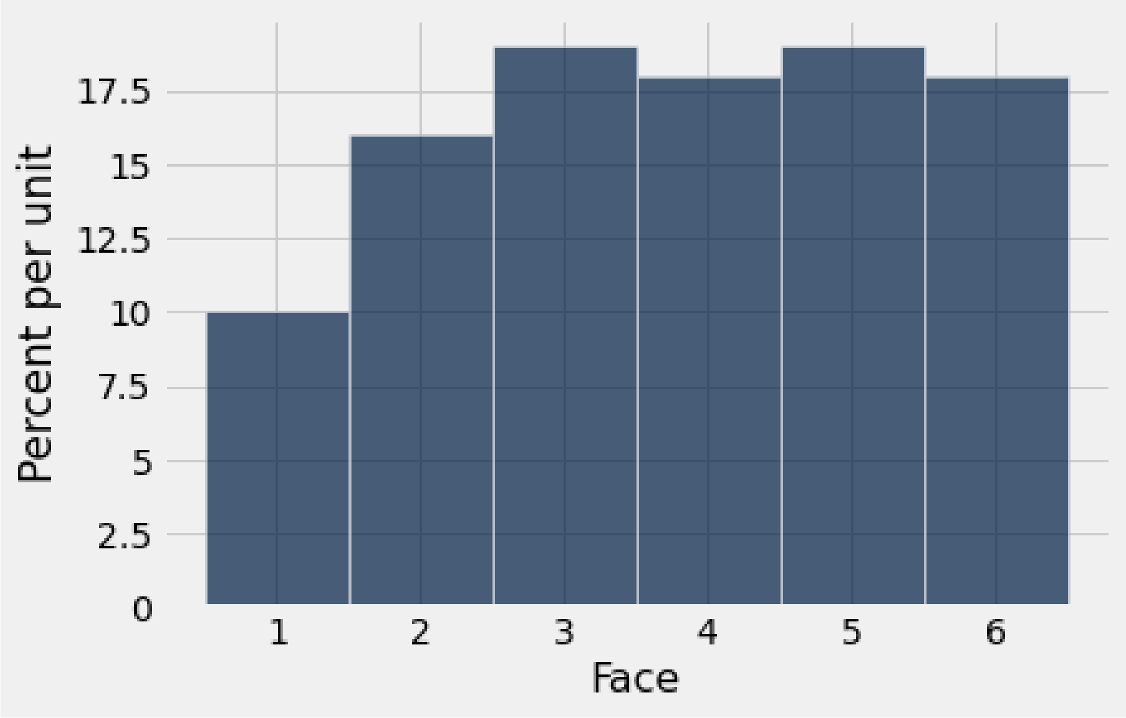
```
[6]: die.sample(10).hist(bins = roll_bins)
```



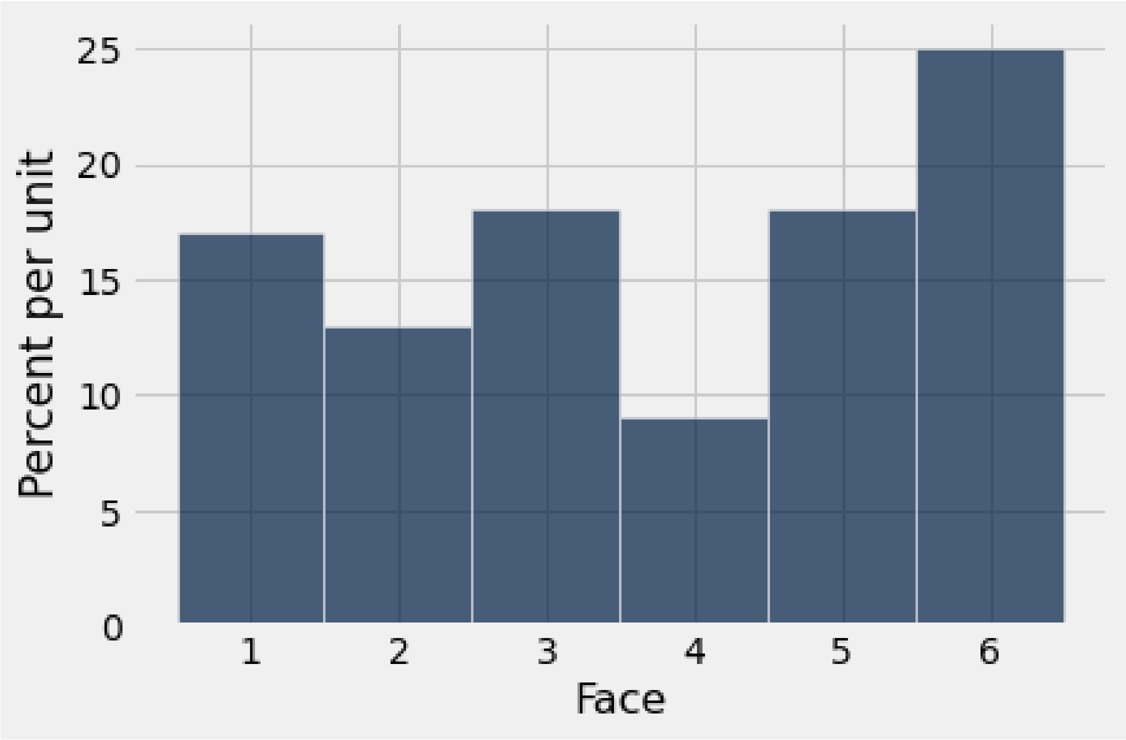
[7]:
```
die.sample(10).hist(bins = roll_bins)
```



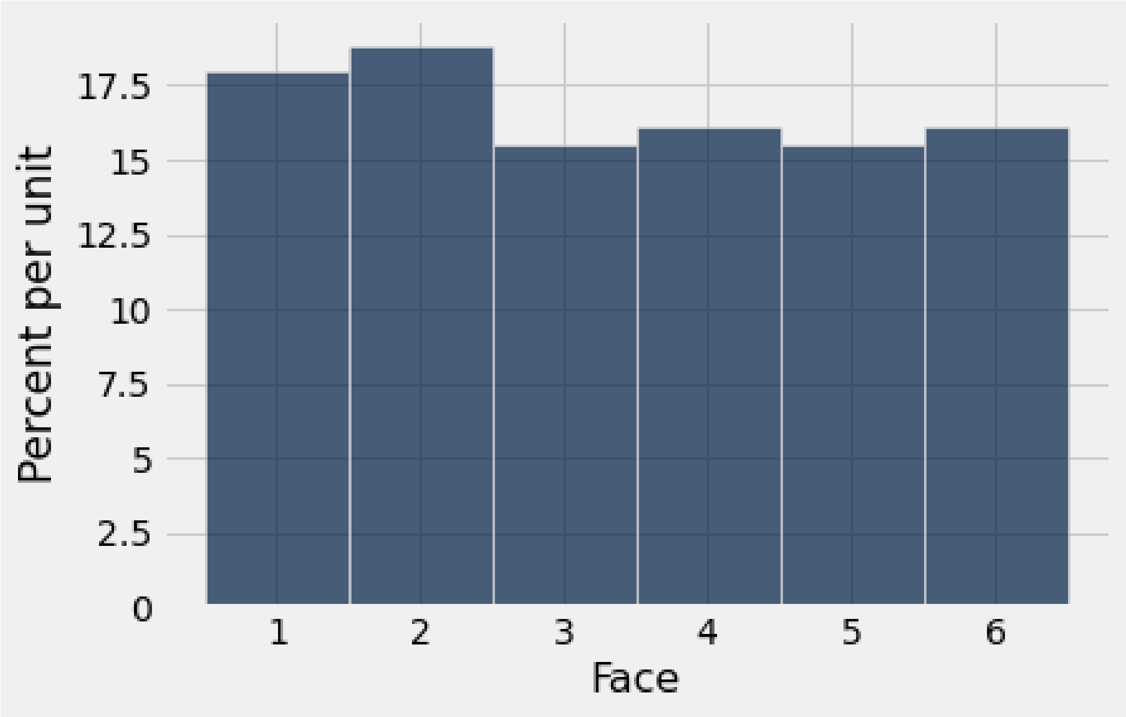
[8]:
```
die.sample(100).hist(bins = roll_bins)
```



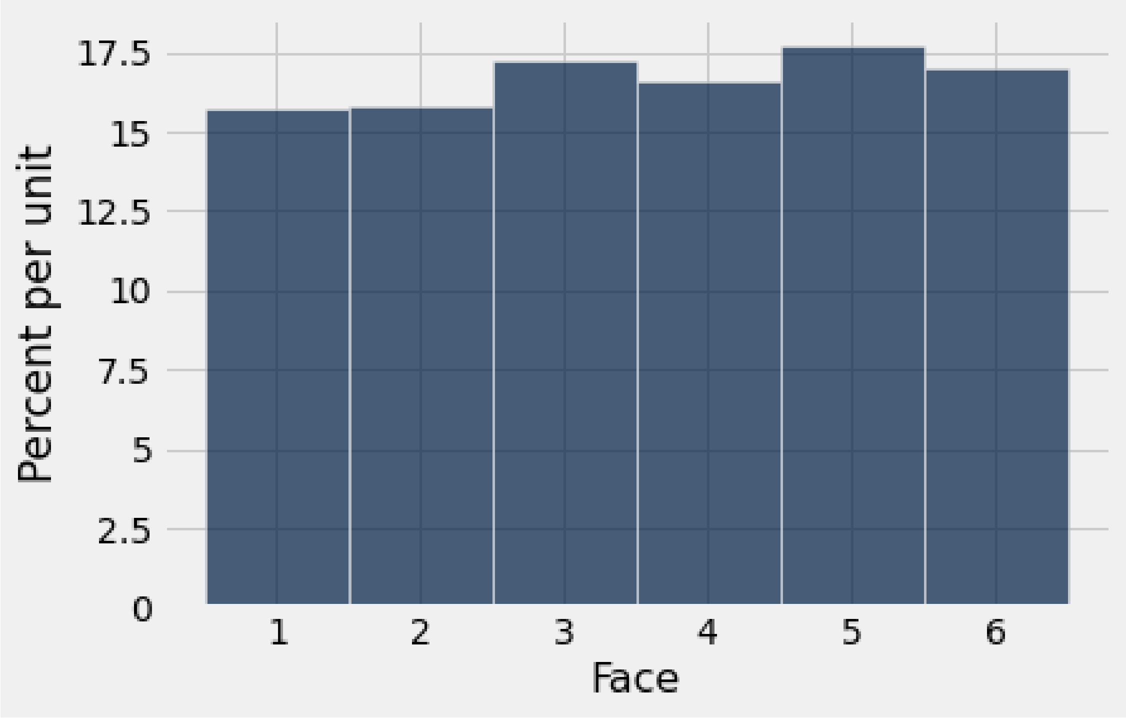
[9]:
```
die.sample(100).hist(bins = roll_bins)
```



[10]:
```
die.sample(1000).hist(bins = roll_bins)
```



[11]:
```
die.sample(1000).hist(bins = roll_bins)
```



[12]:
```
united = Table.read_table('united_summer2015.csv')
united = united.with_column('Row', np.arange(united.num_rows)).
 ↪move_to_start('Row')
```

[13]:
```
united
```

[13]:
```
Row   | Date    | Flight Number | Destination | Delay
0     | 6/1/15 | 73            | HNL         | 257
1     | 6/1/15 | 217           | EWR         | 28
2     | 6/1/15 | 237           | STL         | -3
3     | 6/1/15 | 250           | SAN         | 0
4     | 6/1/15 | 267           | PHL         | 64
5     | 6/1/15 | 273           | SEA         | -6
6     | 6/1/15 | 278           | SEA         | -8
7     | 6/1/15 | 292           | EWR         | 12
8     | 6/1/15 | 300           | HNL         | 20
9     | 6/1/15 | 317           | IND         | -10
… (13815 rows omitted)
```

[14]:
```
united.take(make_array(999, 1000, 1001))
```

[14]:

```
Row  | Date   | Flight Number | Destination | Delay
999  | 6/7/15 | 1684          | LIH         | -3
1000 | 6/7/15 | 1692          | EWR         | 7
1001 | 6/7/15 | 1699          | ATL         | 6
```

[15]:

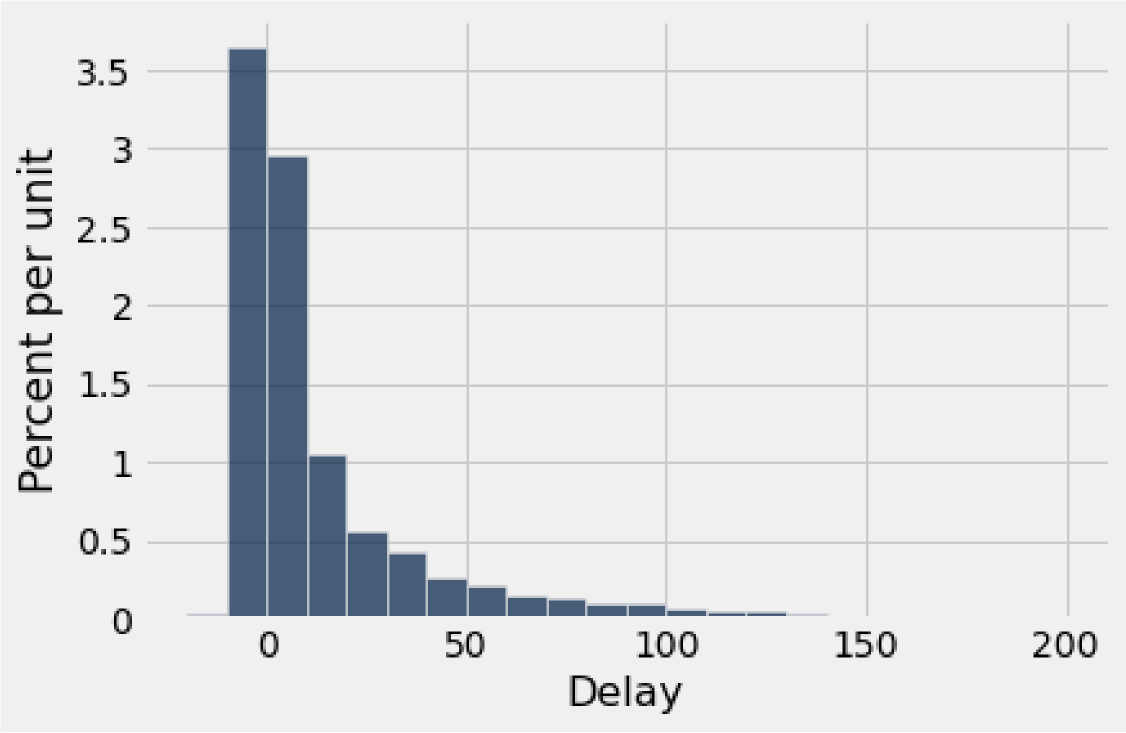
```
united

start = np.random.choice(np.arange(1000))
systematic_sample = united.take(np.arange(start, united.num_rows, 1000))
systematic_sample.show()
```

```
<IPython.core.display.HTML object>
```

[16]:

```
united.hist('Delay', bins = np.arange(-20, 201, 10))
```



[17]:

```
min(united.column('Delay')), max(united.column('Delay'))
```

[17]:

```
(-16, 580)
```

[18]:

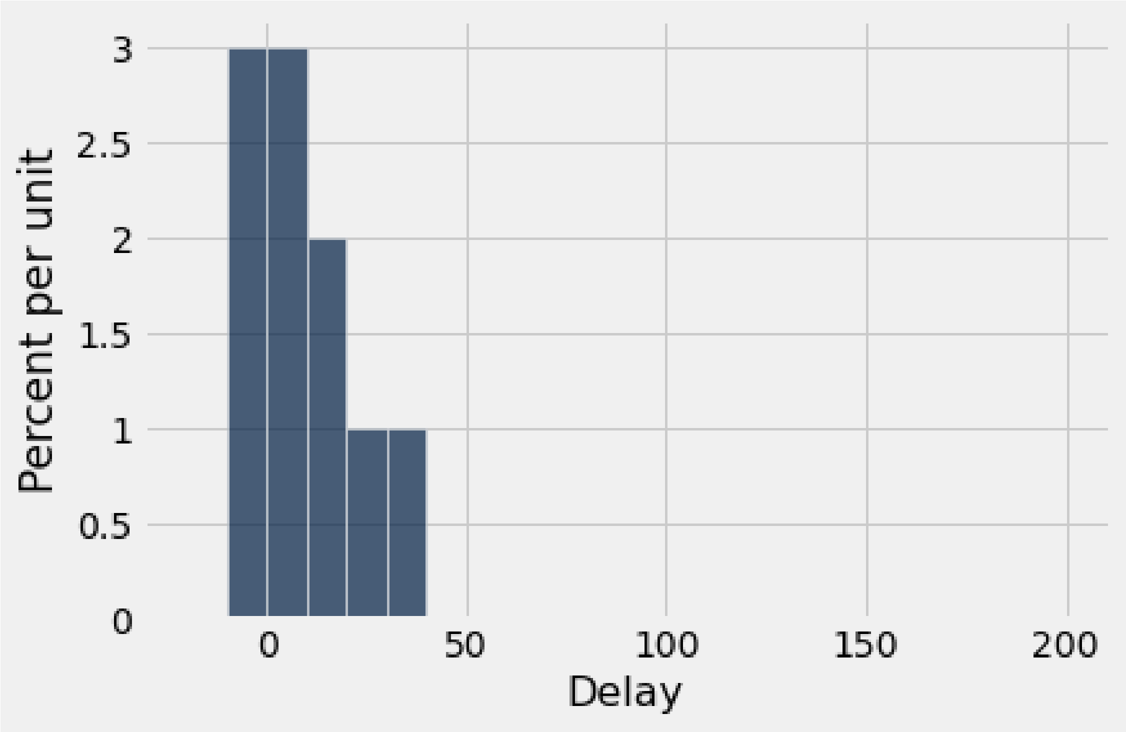
```
united.where('Delay', 580)
```

[18]:

```
Row  | Date    | Flight Number | Destination | Delay
3140 | 6/21/15 | 1964          | SEA         | 580
```

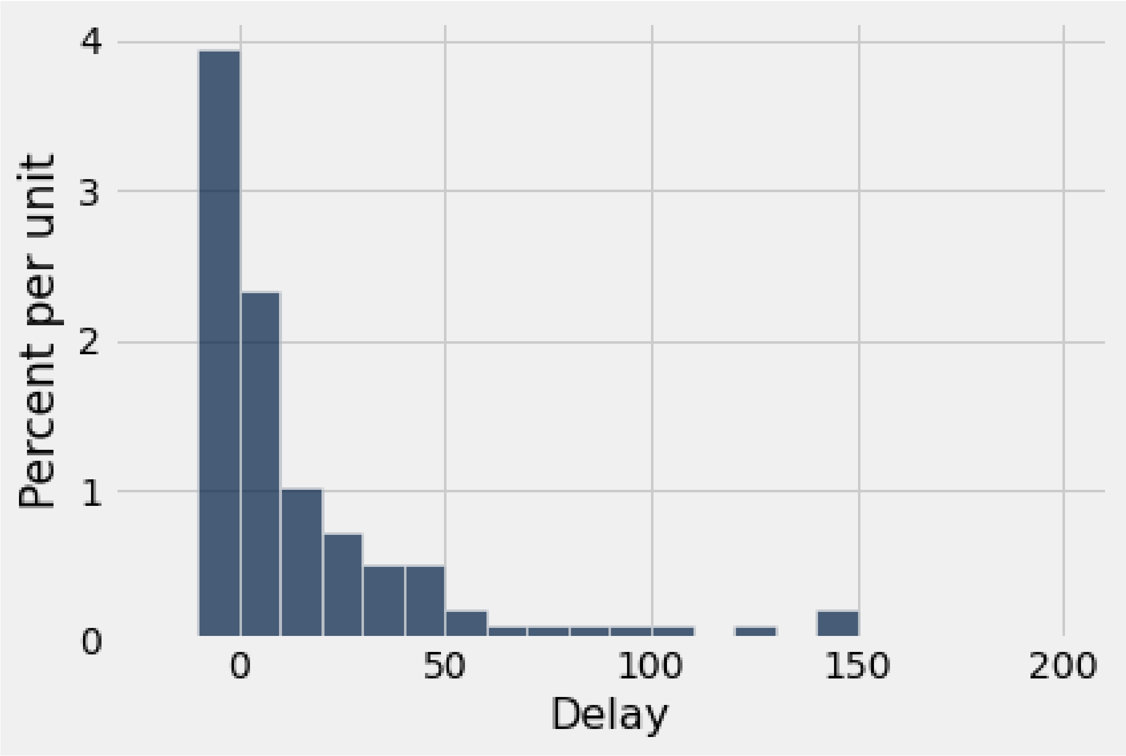
[19]:

```
united.sample(10).hist('Delay', bins = np.arange(-20, 201, 10))
```

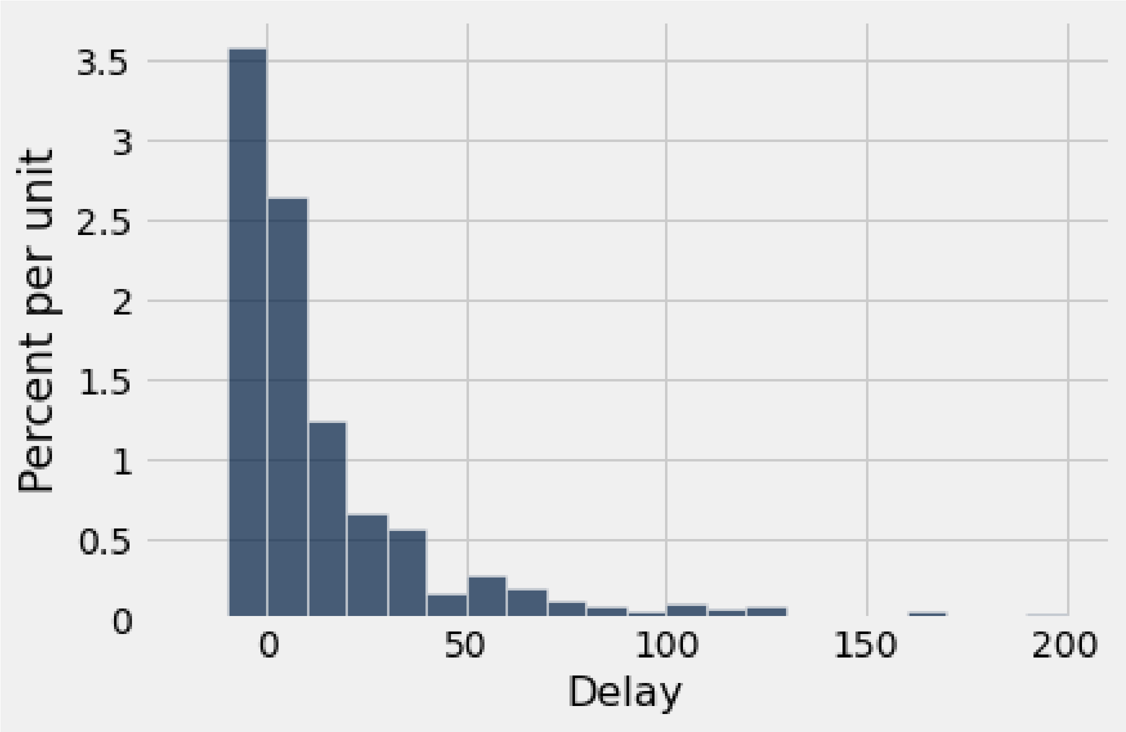


[20]:

```
united.sample(100).hist('Delay', bins = np.arange(-20, 201, 10))
```



[21]:
```
united.sample(1000).hist('Delay', bins = np.arange(-20, 201, 10))
```



0.5 Simulating Statistics

[22]:

```
np.median(united.column('Delay'))
```

[22]:

```
2.0
```

[23]:

```
united.where('Delay', are.below_or_equal_to(2)).num_rows / united.num_rows
```

[23]:

```
0.5018444846292948
```

[24]:

```
np.median(united.sample(10).column('Delay'))
```

[24]:

```
5.5
```

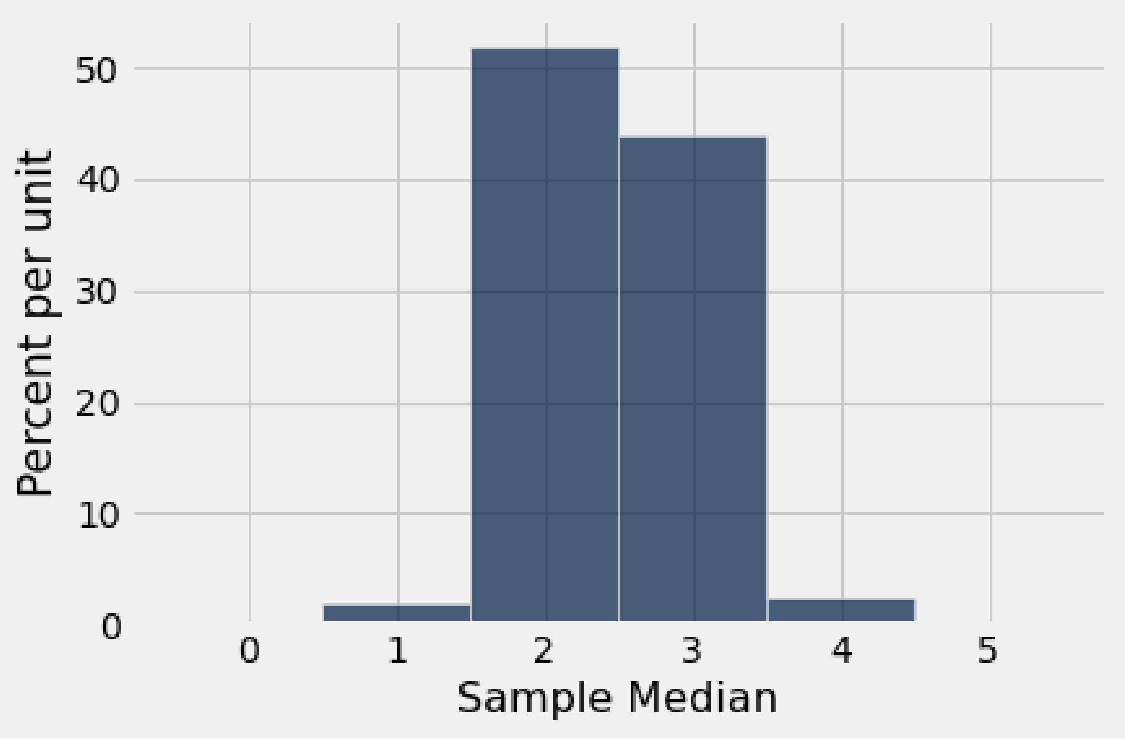
[25]:

```
medians = make_array()

for i in np.arange(10000):
    new_median = np.median(united.sample(1000).column('Delay'))
    medians = np.append(medians, new_median)
```

[26]:

```
Table().with_column('Sample Median', medians).hist(bins = np.arange(-0.5, 5.6,␣
 ↪1))
```



[27]:

```
np.mean(united.column('Delay'))
```

[27]:

```
16.658155515370705
```

[28]:

```
np.mean(united.sample(10).column('Delay'))
```

[28]:

```
1.1
```

[29]:

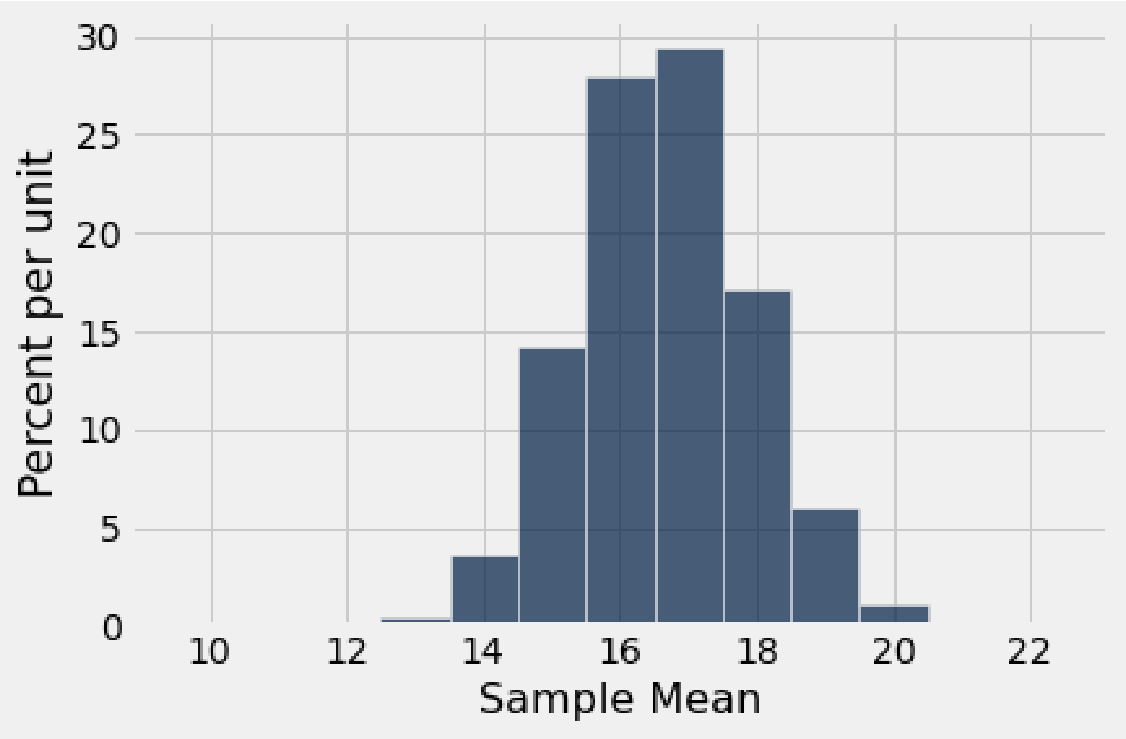
```
means = make_array()
```

[30]:

```
for i in np.arange(10000):
    new_mean = np.mean(united.sample(1000).column('Delay'))
    means = np.append(means, new_mean)
```

[31]:

```
Table().with_column('Sample Mean', means).hist(bins = np.arange(9.5, 22.6, 1))
```



[ ]:

```
```

[ ]:

```
```