lec08

September 13, 2021

[1]:
```
from datascience import *
import numpy as np
import warnings
warnings.filterwarnings("ignore")

%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
plots.rcParams["patch.force_edgecolor"] = True
```

0.1 Lecture 8

0.2 Categorical Distribution

[4]:
```
top = Table.read_table('top_movies_2017.csv')
top
```

[4]:
```
Title                           | Studio          | Gross     | Gross
(Adjusted) | Year
Gone with the Wind              | MGM             | 198676459 | 1796176700
| 1939
Star Wars                       | Fox             | 460998007 | 1583483200
| 1977
The Sound of Music              | Fox             | 158671368 | 1266072700
| 1965
E.T.: The Extra-Terrestrial     | Universal       | 435110554 | 1261085000
| 1982
Titanic                         | Paramount       | 658672302 | 1204368000
| 1997
The Ten Commandments            | Paramount       | 65500000  | 1164590000
| 1956
Jaws                            | Universal       | 260000000 | 1138620700
| 1975
Doctor Zhivago                 | MGM             | 111721910 | 1103564200
| 1965
The Exorcist                   | Warner Brothers | 232906145 | 983226600
| 1973
Snow White and the Seven Dwarves | Disney          | 184925486 | 969010000
```

```
| 1937
… (190 rows omitted)
```

[5]:
```
studios = top.select('Studio')
studios
```

[5]:
```
Studio
MGM
Fox
Fox
Universal
Paramount
Paramount
Universal
MGM
Warner Brothers
Disney
… (190 rows omitted)
```

[6]:
```
studio_distribution = studios.group('Studio')
```

[7]:
```
studio_distribution
```

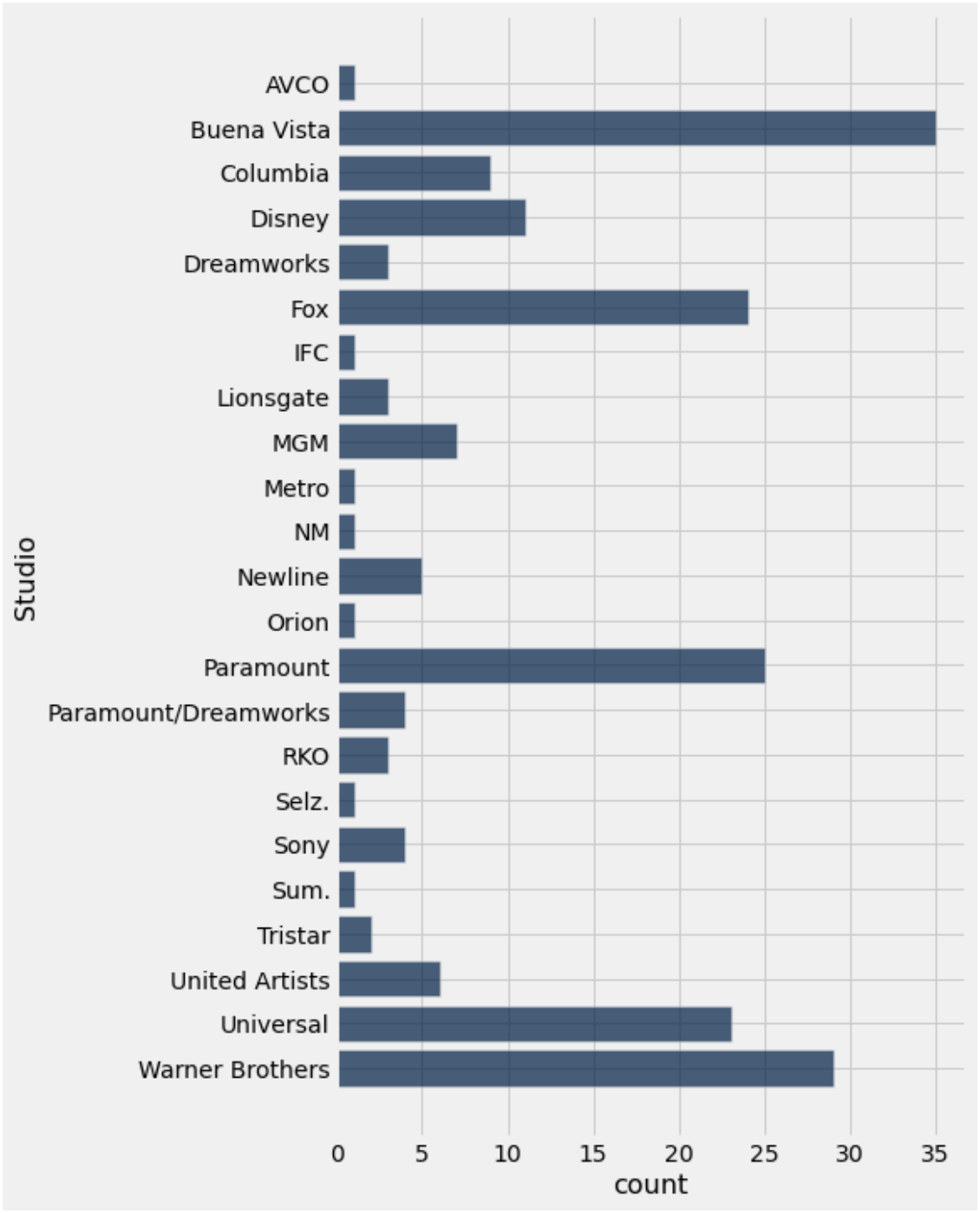
[7]:
```
Studio      | count
AVCO        | 1
Buena Vista | 35
Columbia    | 9
Disney      | 11
Dreamworks  | 3
Fox         | 24
IFC         | 1
Lionsgate   | 3
MGM         | 7
Metro       | 1
… (13 rows omitted)
```

[8]:
```
sum(studio_distribution.column(1))
```

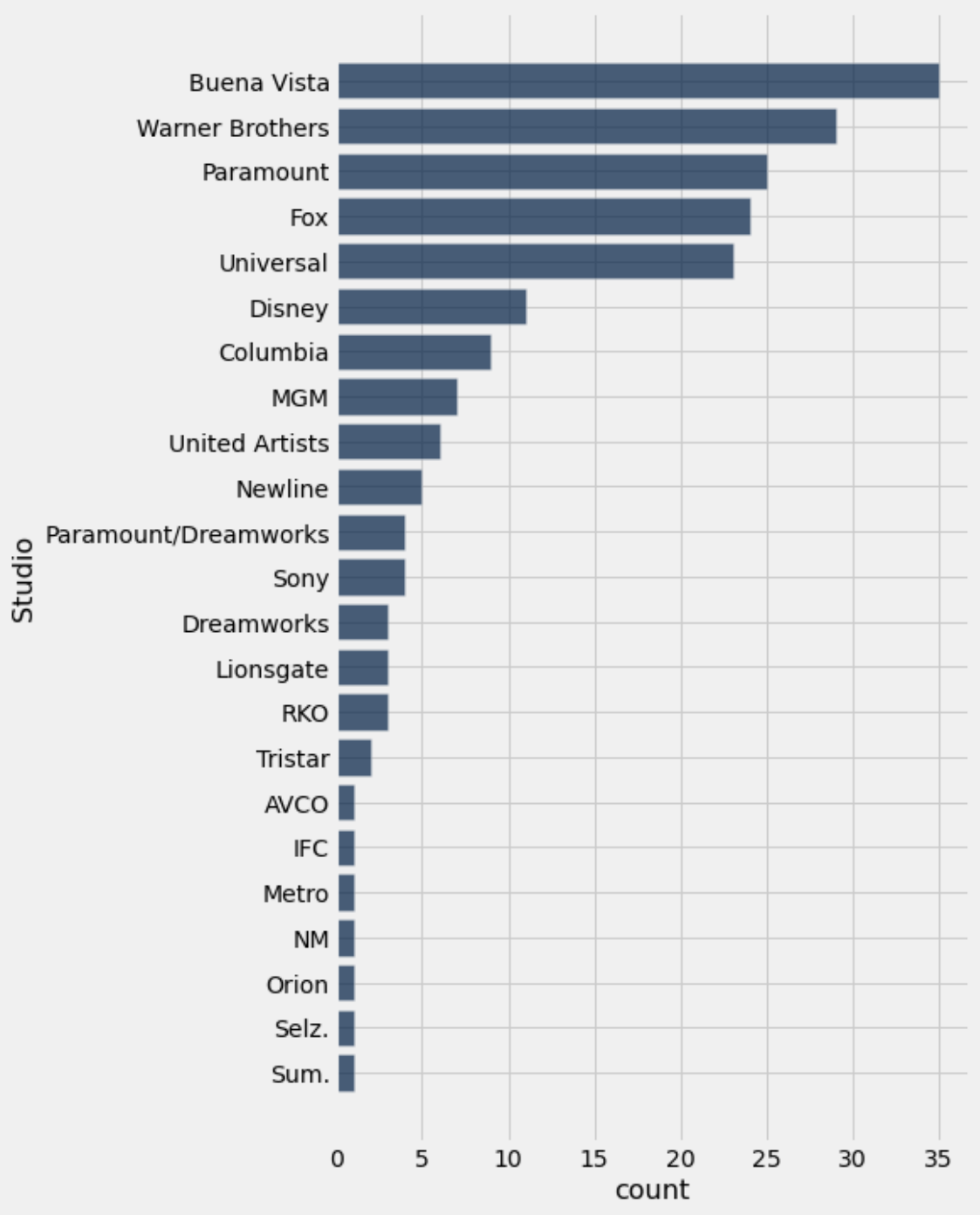
[8]:
```
200
```

0.3 Bar Charts

[9]:
```
studio_distribution.barh('Studio')
```



[10]:
```
studio_distribution.sort(1, descending=True).barh(0)
```



0.4 Numerical Distribution

[11]:
```
ages = 2018 - top.column('Year')
top = top.with_column('Age', ages)
```

[12]:
```
top
```

[12]:
```
Title                             | Studio          | Gross     | Gross
(Adjusted) | Year | Age
Gone with the Wind                | MGM             | 198676459 | 1796176700
| 1939 | 79
Star Wars                         | Fox             | 460998007 | 1583483200
| 1977 | 41
The Sound of Music                | Fox             | 158671368 | 1266072700
| 1965 | 53
E.T.: The Extra-Terrestrial       | Universal       | 435110554 | 1261085000
| 1982 | 36
Titanic                           | Paramount       | 658672302 | 1204368000
| 1997 | 21
The Ten Commandments              | Paramount       | 65500000  | 1164590000
| 1956 | 62
Jaws                              | Universal       | 260000000 | 1138620700
| 1975 | 43
Doctor Zhivago                    | MGM             | 111721910 | 1103564200
| 1965 | 53
The Exorcist                      | Warner Brothers | 232906145 | 983226600
| 1973 | 45
Snow White and the Seven Dwarves | Disney          | 184925486 | 969010000
| 1937 | 81
… (190 rows omitted)
```

0.5 Binning

[13]:
```
[min(ages), max(ages)]
```

[13]:
```
[1, 97]
```

[14]:
```
my_bins = make_array(0, 5, 10, 15, 25, 40, 65, 100)
```

[15]:
```
top.bin('Age', bins = my_bins)
```

[15]:
```
bin  | Age count
0    | 21
5    | 17
10   | 18
15   | 37
25   | 40
40   | 52
65   | 15
100  | 0
```

[16]:
```
sum(top.bin('Age', bins = my_bins).column(1))
```

[16]:
```
200
```

[17]:
```
top.bin('Age', bins = np.arange(0, 101, 25))
```

[17]:
```
bin  | Age count
0    | 93
25   | 66
50   | 34
75   | 7
100  | 0
```

[18]:
```
top.bin('Age', bins = np.arange(0, 60, 25))
```

[18]:
```
bin  | Age count
0    | 93
25   | 68
50   | 0
```

[19]:
```
top.where('Age', 50)
```

[19]:
```
Title                 | Studio   | Gross    | Gross (Adjusted) | Year | Age
2001: A Space Odyssey | MGM      | 56954992 | 385261600        | 1968 | 50
Funny Girl            | Columbia | 52223306 | 355950700        | 1968 | 50
```

0.6 Histograms

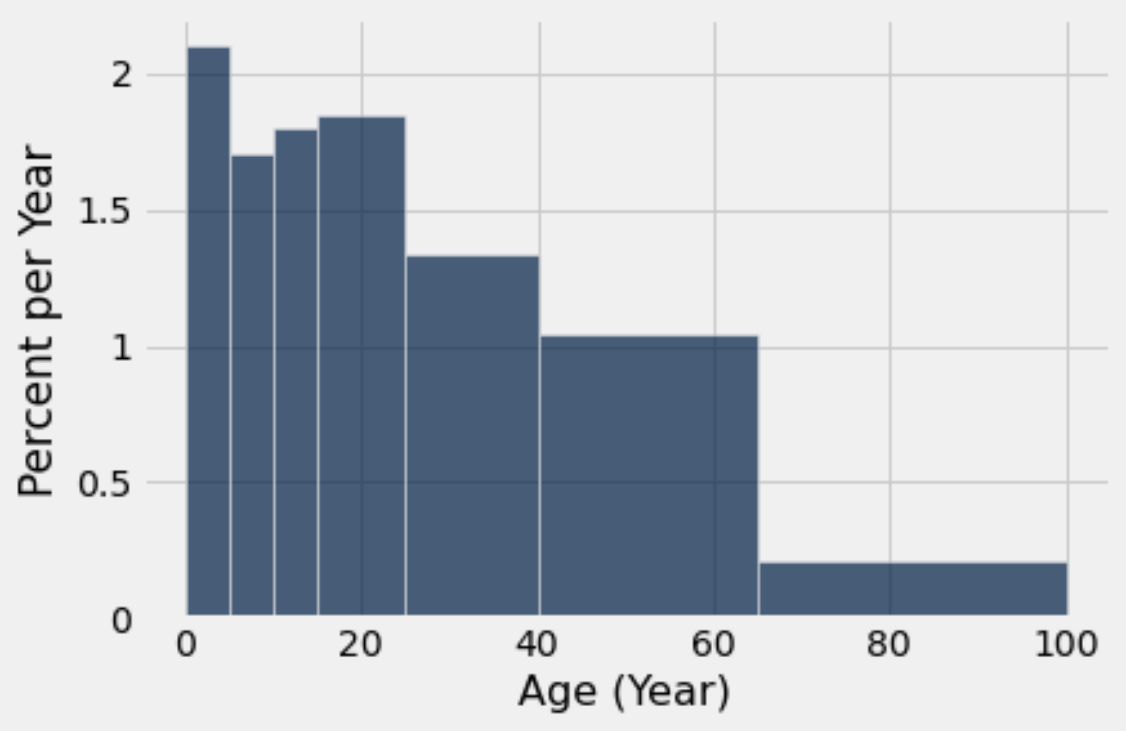
[20]:
```
my_bins
```

[20]:
```
array([  0,   5,  10,  15,  25,  40,  65, 100], dtype=int64)
```

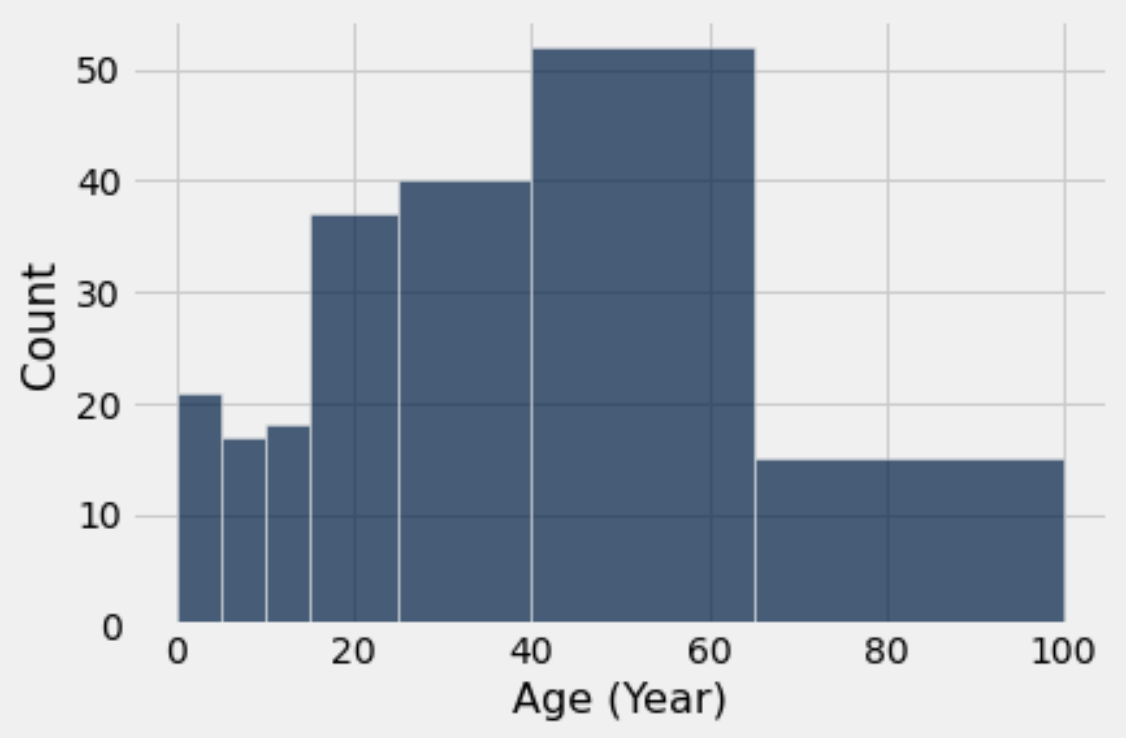
[21]:
```
top.bin('Age', bins = my_bins)
```

[21]:
```
bin  | Age count
0    | 21
5    | 17
10   | 18
15   | 37
25   | 40
40   | 52
65   | 15
100  | 0
```

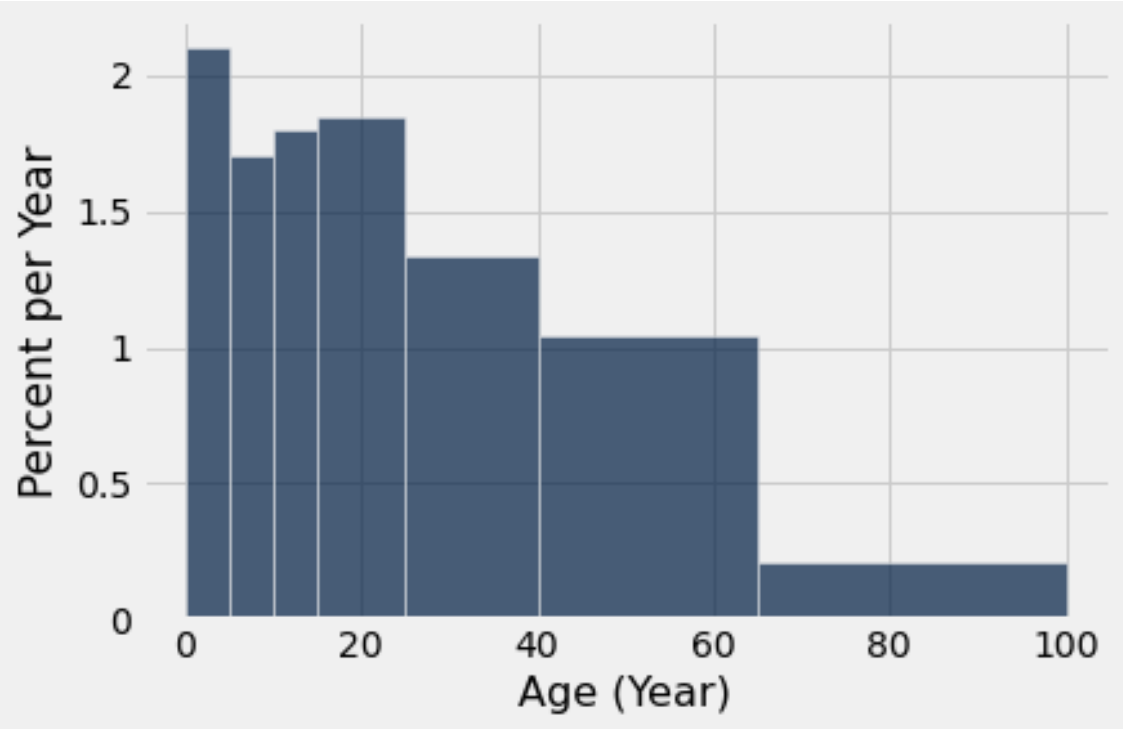
[22]:
```
top.hist('Age', bins = my_bins, unit = 'Year')
```



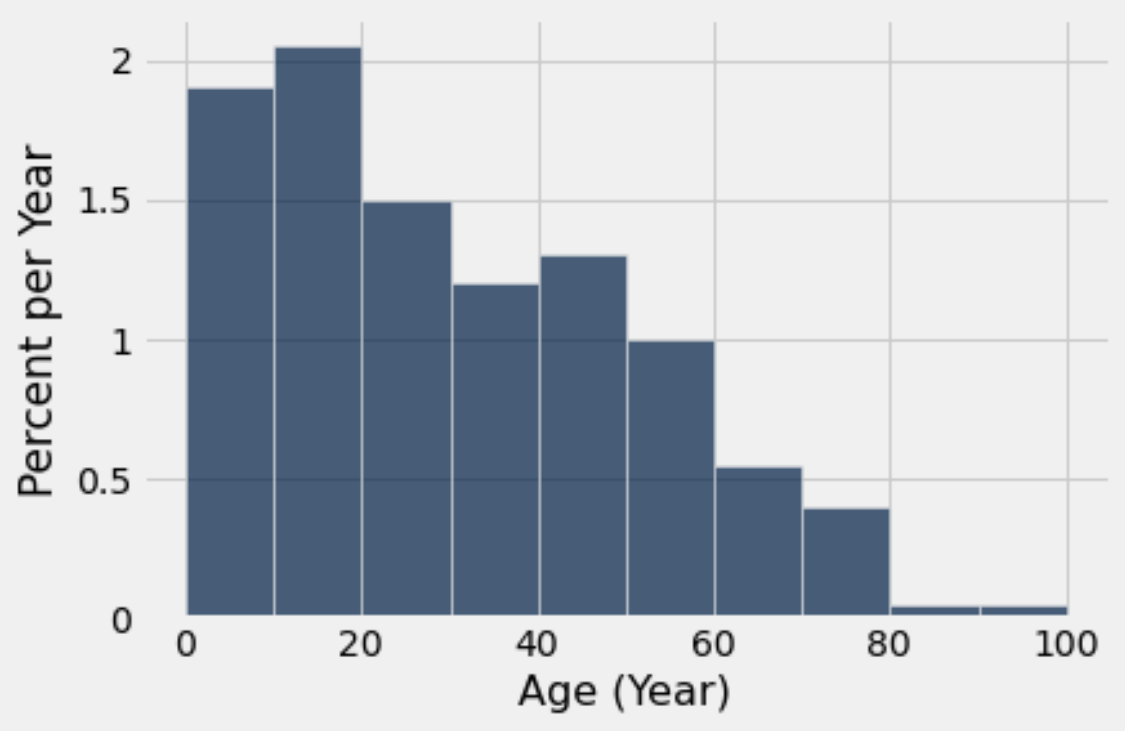
[23]:
```
# What *not* to do:
top.hist('Age', bins = my_bins, unit = 'Year', normed = False)
```



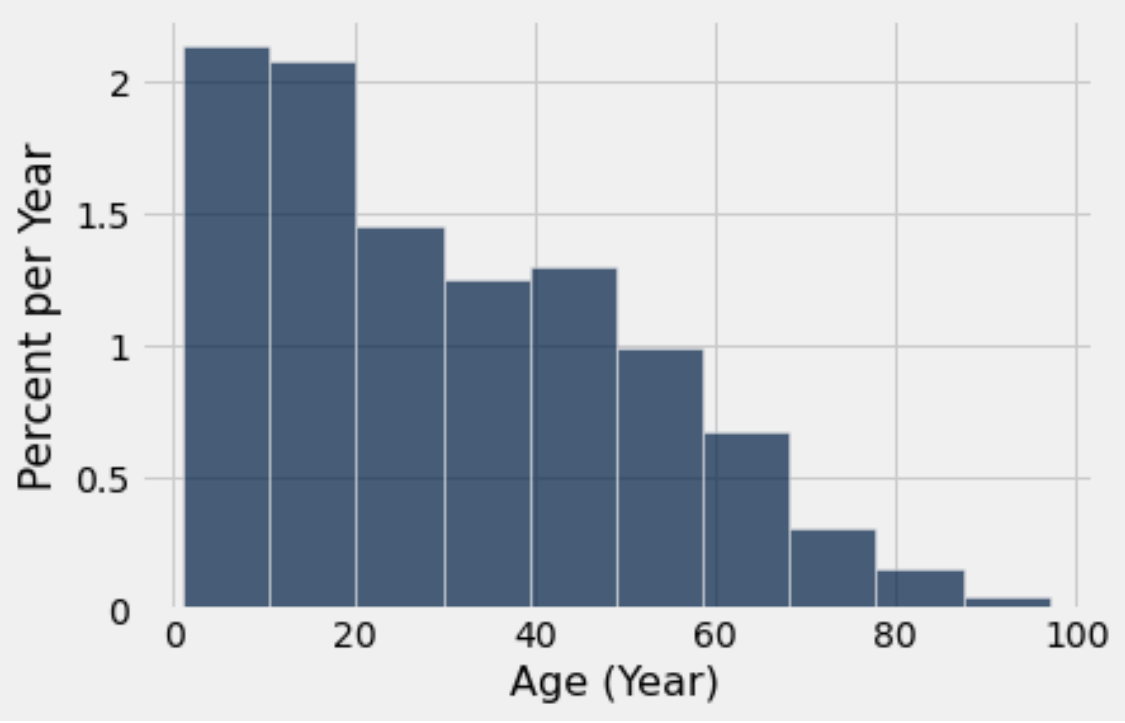
[24]:
```
top.hist('Age', bins = my_bins, unit = 'Year')
```



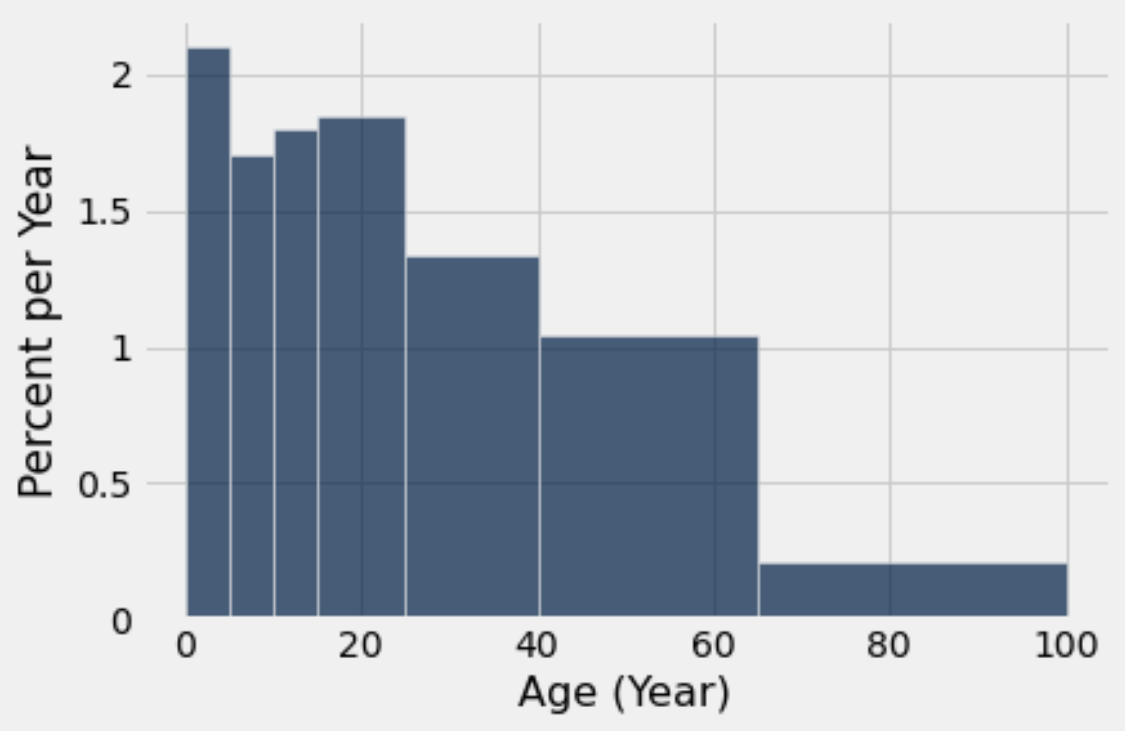
[25]:
```
top.hist('Age', bins = np.arange(0, 110, 10), unit = 'Year')
```



[26]:
```
top.hist('Age', unit='Year')
```



[27]:
```
top.hist('Age', bins = my_bins, unit = 'Year')
```



[28]:
```
distribution = top.bin('Age', bins = my_bins)
```

[29]:
```
distribution
```

[29]:
```
bin  | Age count
0    | 21
5    | 17
10   | 18
15   | 37
25   | 40
40   | 52
65   | 15
100  | 0
```

[30]:
```
# 52 out of 200 movies in the [40, 65) bin

percent = (52/200) * 100
percent
```

[30]:
```
26.0
```

[31]:
```
width = 65 - 40
width
```

[31]: 25

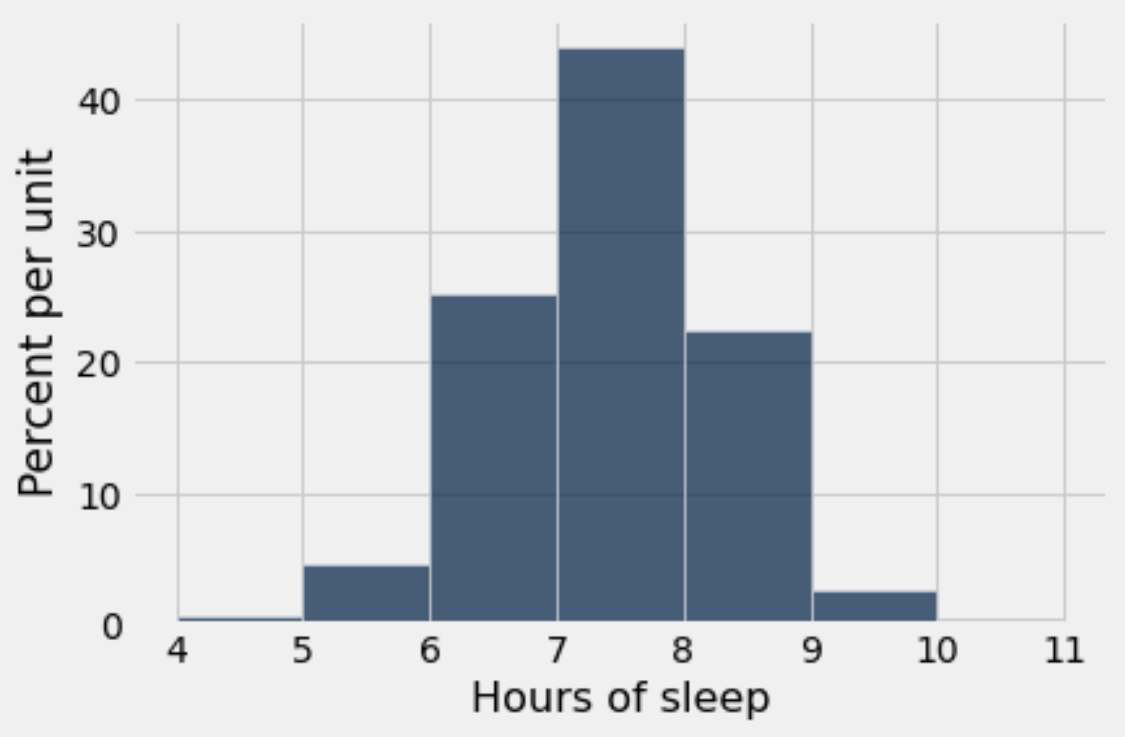
[32]:
```
height = percent / width
height
```

[32]: 1.04

[34]:
```
survey = Table.read_table('welcome_survey.csv')
survey.show(5)
```

<IPython.core.display.HTML object>

[35]:
```
survey.hist('Hours of sleep', bins=np.arange(4, 12))
```

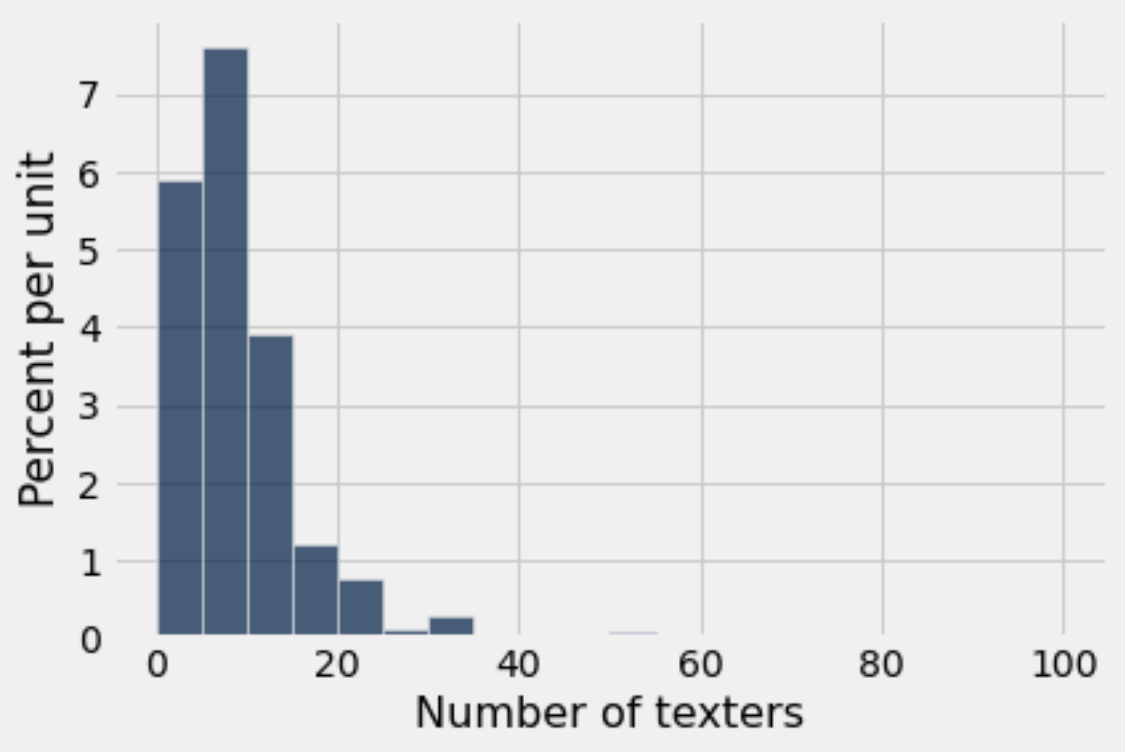


[36]:
```
texters = survey.column('Number of texters')
```

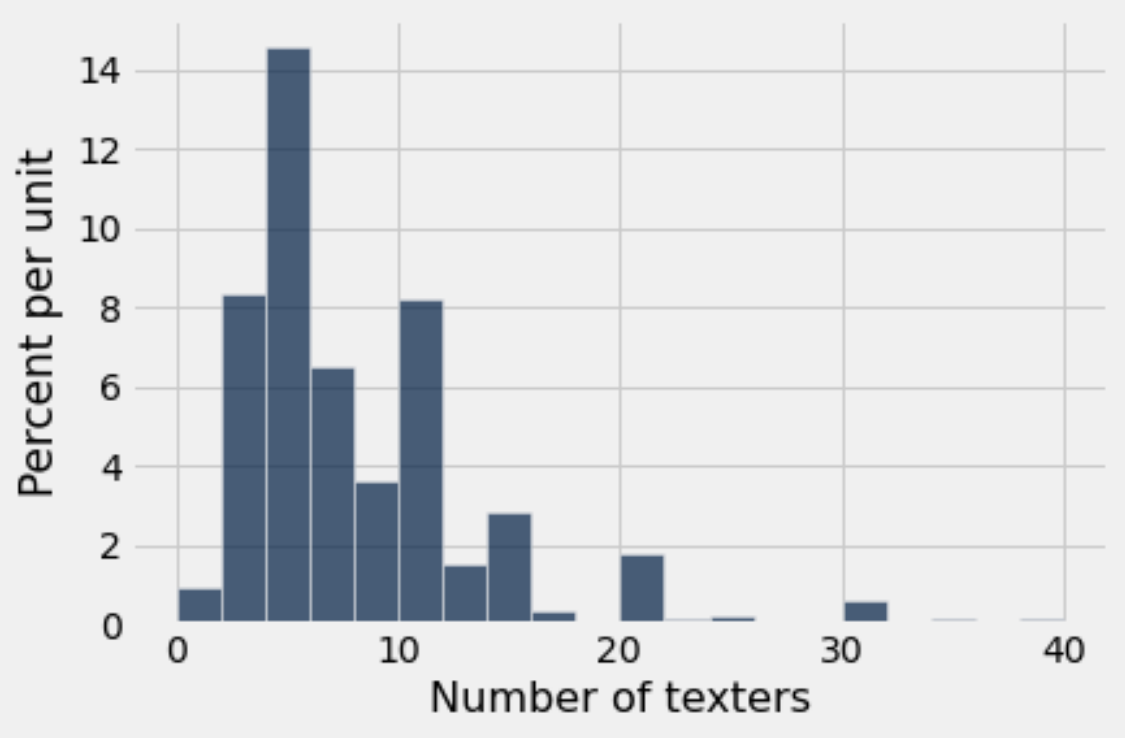
[37]:
```
[min(texters), max(texters)]
```

[37]: [0, 100]

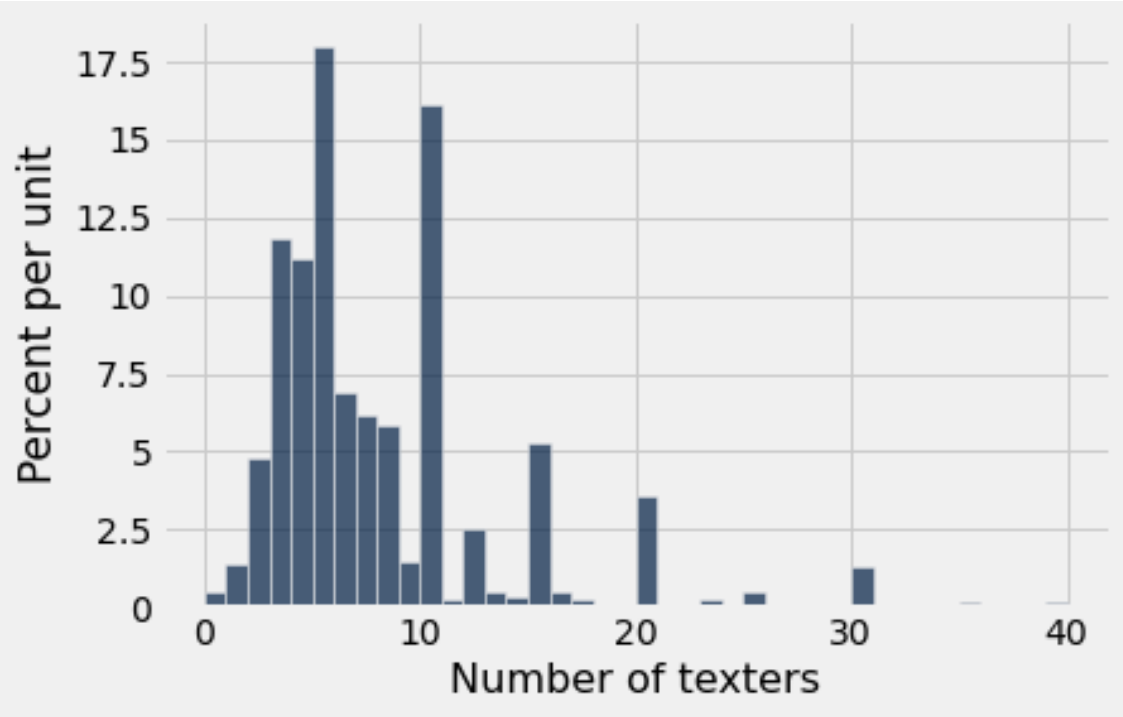
[38]:
```
survey.hist('Number of texters', bins=np.arange(0, 101, 5))
```



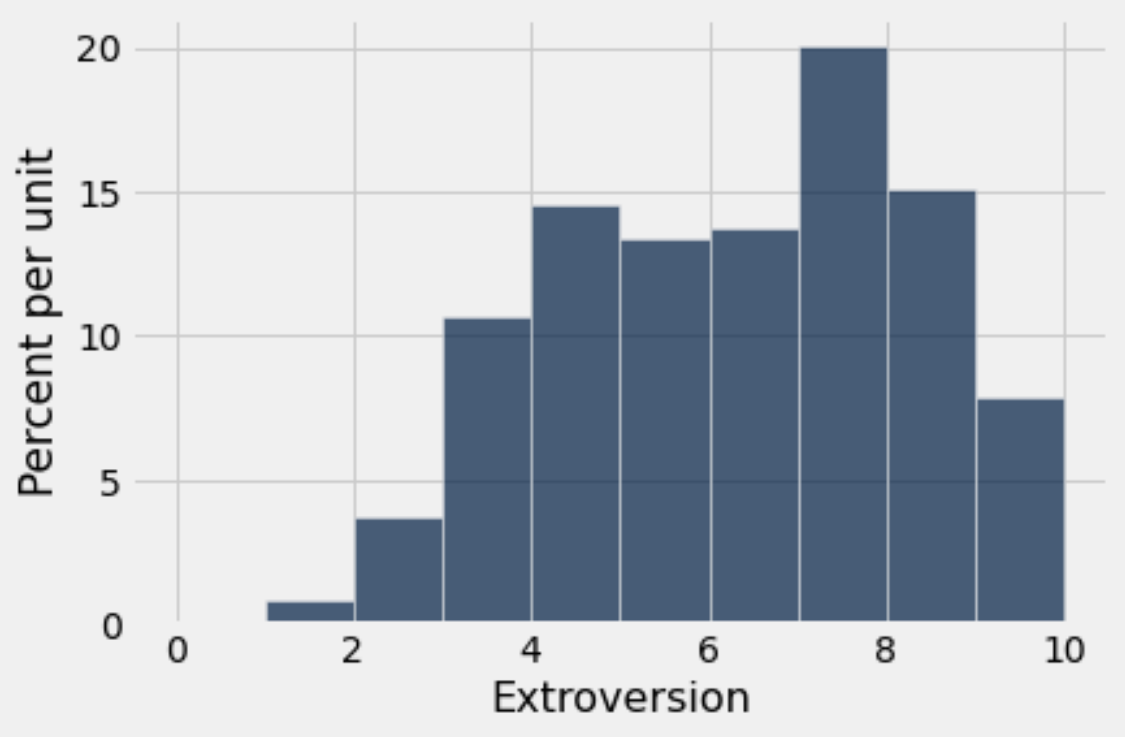
[39]:
```
survey.hist('Number of texters', bins=np.arange(0, 41, 2))
```



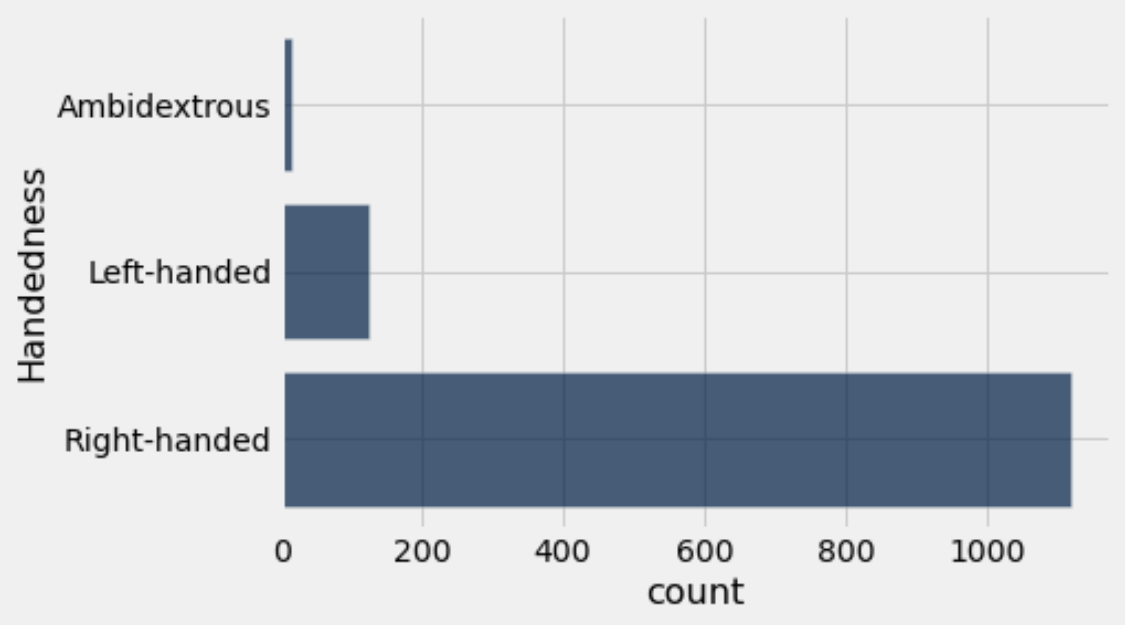
[40]:
```
survey.hist('Number of texters', bins=np.arange(0, 41, 1))
```



[41]:
```
survey.hist('Extroversion', bins=np.arange(0, 11))
```



[42]:
```
hand_distribution = survey.group('Handedness')
hand_distribution.barh('Handedness')
```



[43]:
```
galton = Table.read_table('galton.csv')

galton
```

[43]:
```
family | father | mother | midparentHeight | children | childNum | gender |
childHeight
1      | 78.5   | 67     | 75.43           | 4        | 1        | male   | 73.2
1      | 78.5   | 67     | 75.43           | 4        | 2        | female | 69.2
1      | 78.5   | 67     | 75.43           | 4        | 3        | female | 69
1      | 78.5   | 67     | 75.43           | 4        | 4        | female | 69
2      | 75.5   | 66.5   | 73.66           | 4        | 1        | male   | 73.5
2      | 75.5   | 66.5   | 73.66           | 4        | 2        | male   | 72.5
2      | 75.5   | 66.5   | 73.66           | 4        | 3        | female | 65.5
2      | 75.5   | 66.5   | 73.66           | 4        | 4        | female | 65.5
3      | 75     | 64     | 72.06           | 2        | 1        | male   | 71
3      | 75     | 64     | 72.06           | 2        | 2        | female | 68
… (924 rows omitted)
```

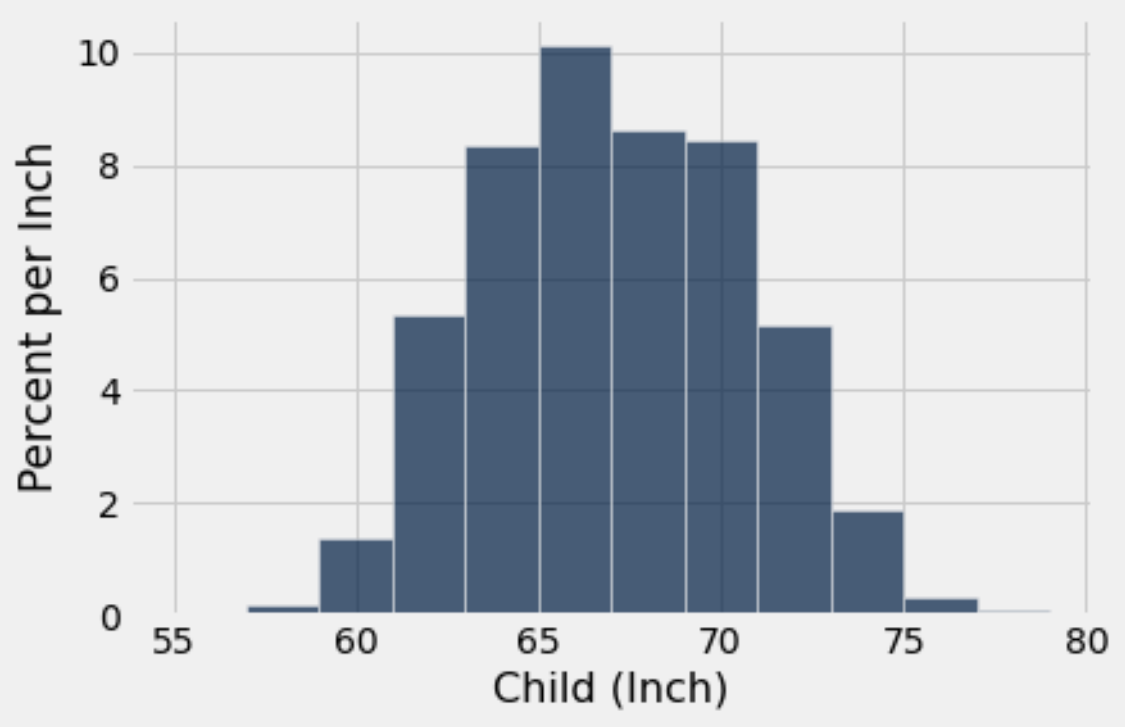
[44]:
```
heights = galton.select(3, 7).relabeled(0, 'MidParent').relabeled(1, 'Child')

heights
```

[44]:
```
MidParent | Child
75.43     | 73.2
75.43     | 69.2
75.43     | 69
75.43     | 69
73.66     | 73.5
73.66     | 72.5
73.66     | 65.5
73.66     | 65.5
72.06     | 71
72.06     | 68
… (924 rows omitted)
```

[45]:
```
my_bins = np.arange(55, 80, 2)
```

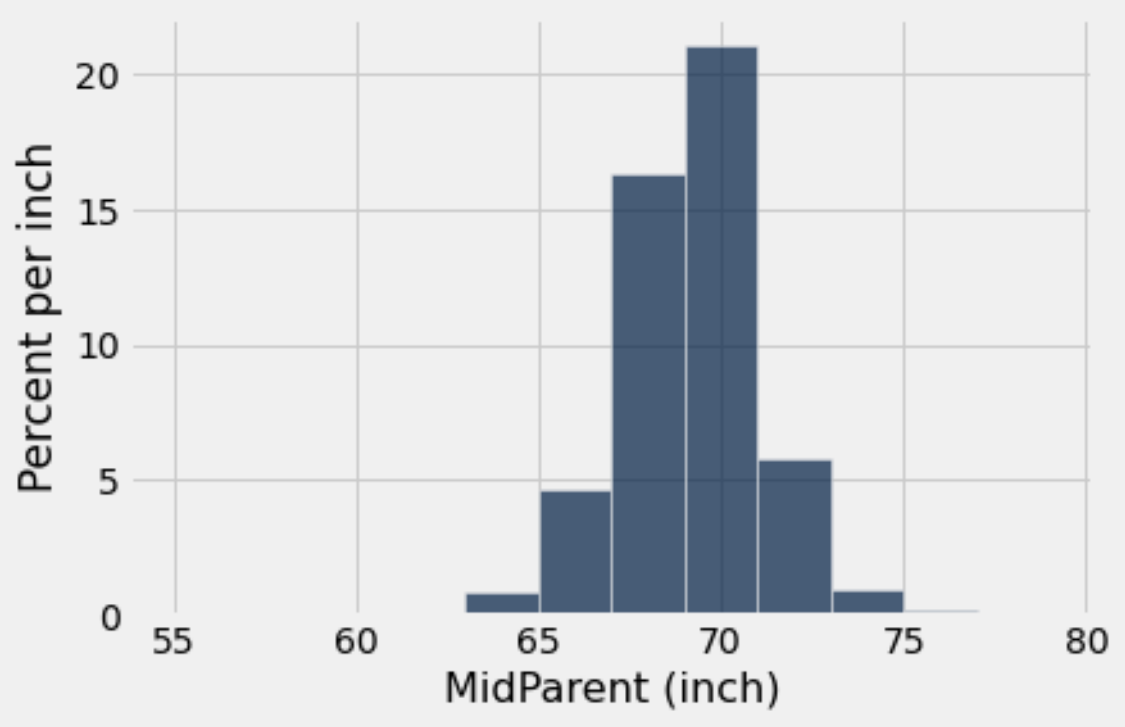
[46]:
```
heights.hist('Child', bins = my_bins, unit='Inch')
```



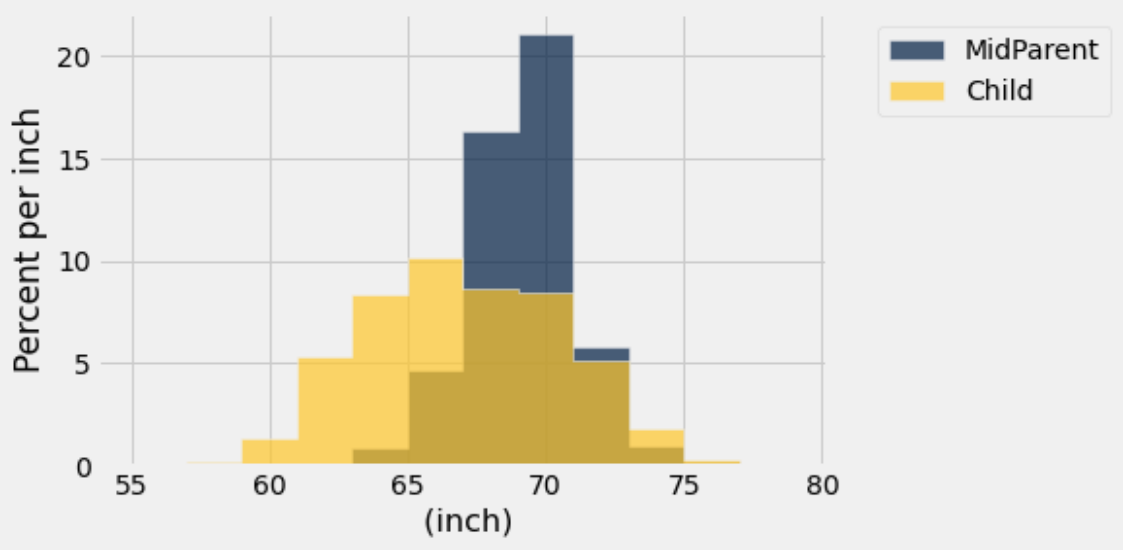
[47]:
```
heights.where('Child', are.between(65, 67)).num_rows / heights.num_rows
```

[47]: 0.20235546038543897

[48]:
```
heights.hist('MidParent', bins=my_bins, unit='inch')
```



[49]:
```
heights.hist(bins=my_bins, unit='inch')
```



[ ]:

[ ]: