# 5) Visualization

Vitor Kamada

December 2019

# Reference

Tables, Graphics, and Figures from

**Computational and Inferential Thinking: The Foundations of Data Science**

Adhikari & DeNero (2019): Ch 7. Visualization

https://www.inferentialthinking.com

# Internet Movie Database (IMDB)

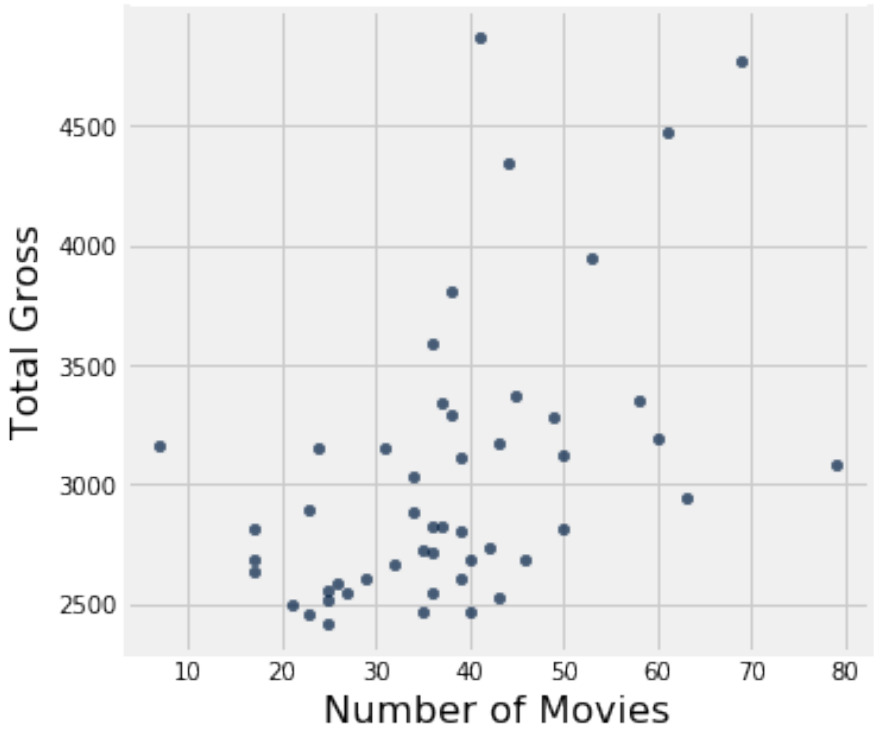
```
from datascience import *
path_data = 'https://github.com/data-8/textbook/raw/gh-pages/data/'
actors = Table.read_table(path_data + 'actors.csv')
```

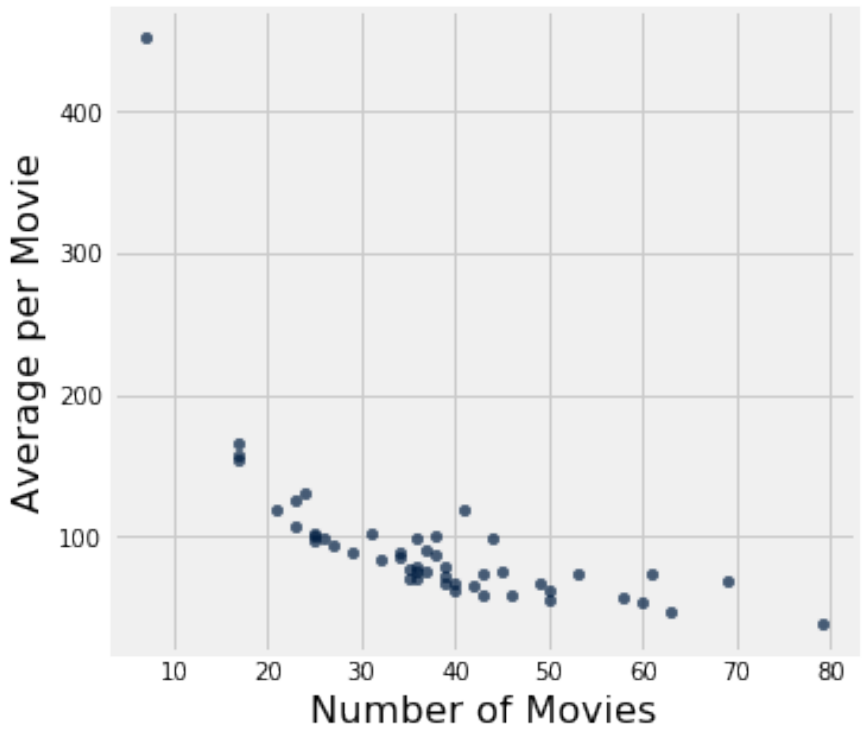
| Actor | Total Gross | Number of Movies | Average per Movie |
|---|---|---|---|
| Harrison Ford | 4871.7 | 41 | 118.8 |
| Samuel L. Jackson | 4772.8 | 69 | 69.2 |
| Morgan Freeman | 4468.3 | 61 | 73.3 |
| Tom Hanks | 4340.8 | 44 | 98.7 |

50 top grossing actors

# Scatter Plots

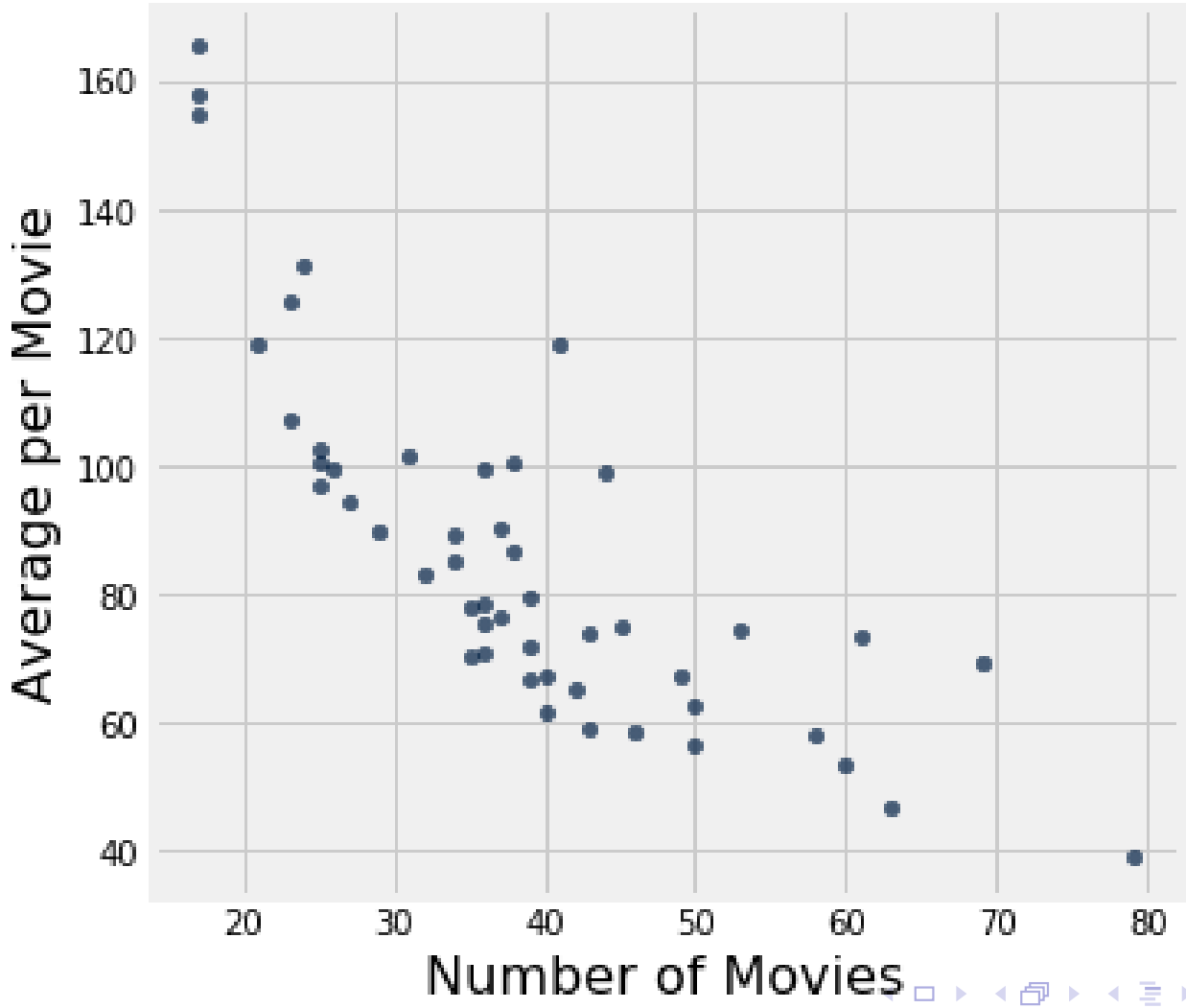
```
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
%matplotlib inline
actors.scatter('Number of Movies', 'Total Gross')

actors.scatter('Number of Movies', 'Average per Movie')
```



# Plot without Outlier

```
no_outlier = actors.where('Number of Movies', are.above(10))
no_outlier.scatter('Number of Movies', 'Average per Movie')
```



# Explaining Outlier: Droid C-3PO in Star Wars

```
actors.where('Number of Movies', are.above(60))
```

| Actor | Total Gross | Number of Movies | Average per Movie |
|---|---|---|---|
| Samuel L. Jackson | 4772.8 | 69 | 69.2 |
| Morgan Freeman | 4468.3 | 61 | 73.3 |
| Robert DeNiro | 3081.3 | 79 | 39 |

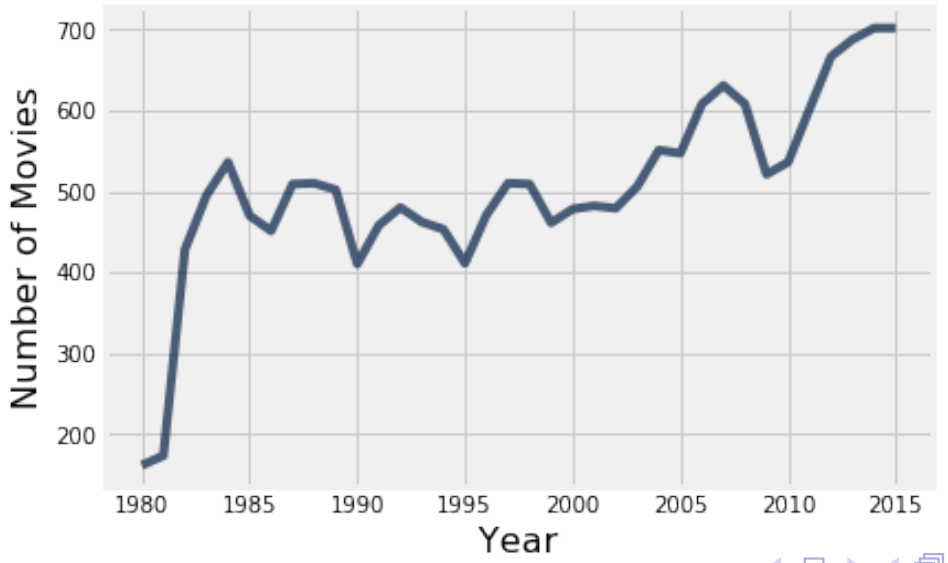
```
actors.where('Number of Movies', are.below(10))
```

| Actor | Total Gross | Number of Movies | Average per Movie | #1 Movie | Gross |
|---|---|---|---|---|---|
| Anthony Daniels | 3162.9 | 7 | 451.8 | Star Wars: The Force Awakens | 936.7 |

```
movies_by_year = Table.read_table(path_data + 'movies_by_year.csv')
```

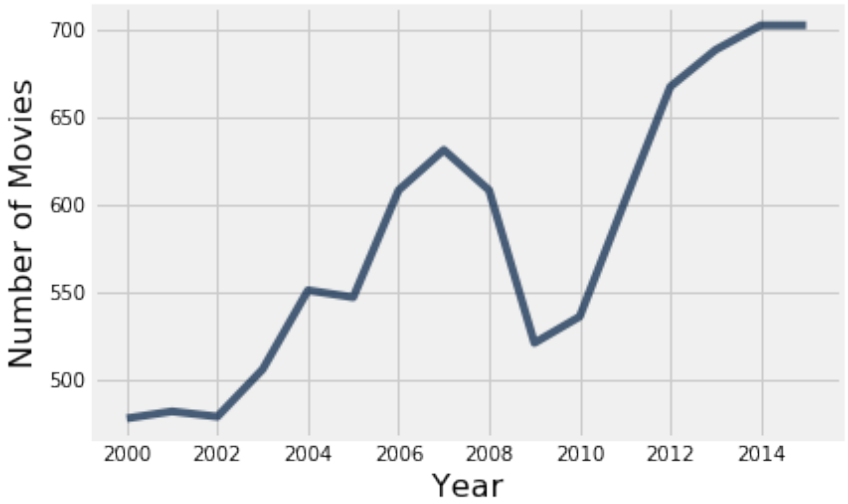
| Year | Total Gross | Number of Movies | #1 Movie |
|---|---|---|---|
| 2015 | 11128.5 | 702 | Star Wars: The Force Awakens |
| 2014 | 10360.8 | 702 | American Sniper |
| 2013 | 10923.6 | 688 | Catching Fire |
| 2012 | 10837.4 | 667 | The Avengers |
| 2011 | 10174.3 | 602 | Harry Potter / Deathly Hallows (P2) |

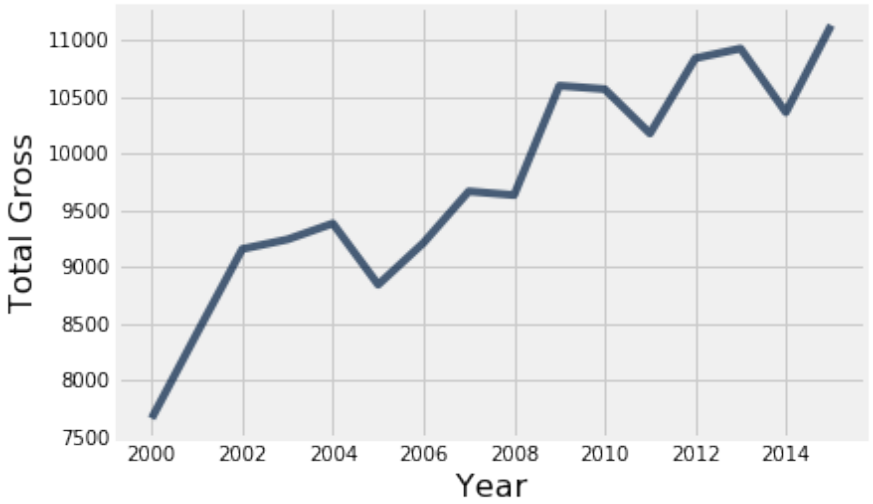
```
movies_by_year.plot('Year', 'Number of Movies')
```



```
century_21 = movies_by_year.where('Year', are.above(1999))
century_21.plot('Year', 'Number of Movies')
```

```
century_21.plot('Year', 'Total Gross')
```




```
century_21.where('Year', are.equal_to(2009))
```

| Year | Total Gross | Number of Movies | #1 Movie |
|---|---|---|---|
| 2009 | 10595.5 | 521 | Avatar |

# USA Top Grossing Movies of All Time

```
top = Table.read_table(path_data + 'top_movies.csv')
# Make the numbers in the Gross and Gross
# (Adjusted) columns look nicer:
top.set_format([2, 3], NumberFormatter)
```

| Title | Studio | Gross | Gross (Adjusted) | Year |
|---|---|---|---|---|
| Star Wars: The Force Awakens | Buena Vista (Disney) | 906,723,418 | 906,723,400 | 2015 |
| Avatar | Fox | 760,507,625 | 846,120,800 | 2009 |
| Titanic | Paramount | 658,672,302 | 1,178,627,900 | 1997 |
| Jurassic World | Universal | 652,270,625 | 687,728,000 | 2015 |

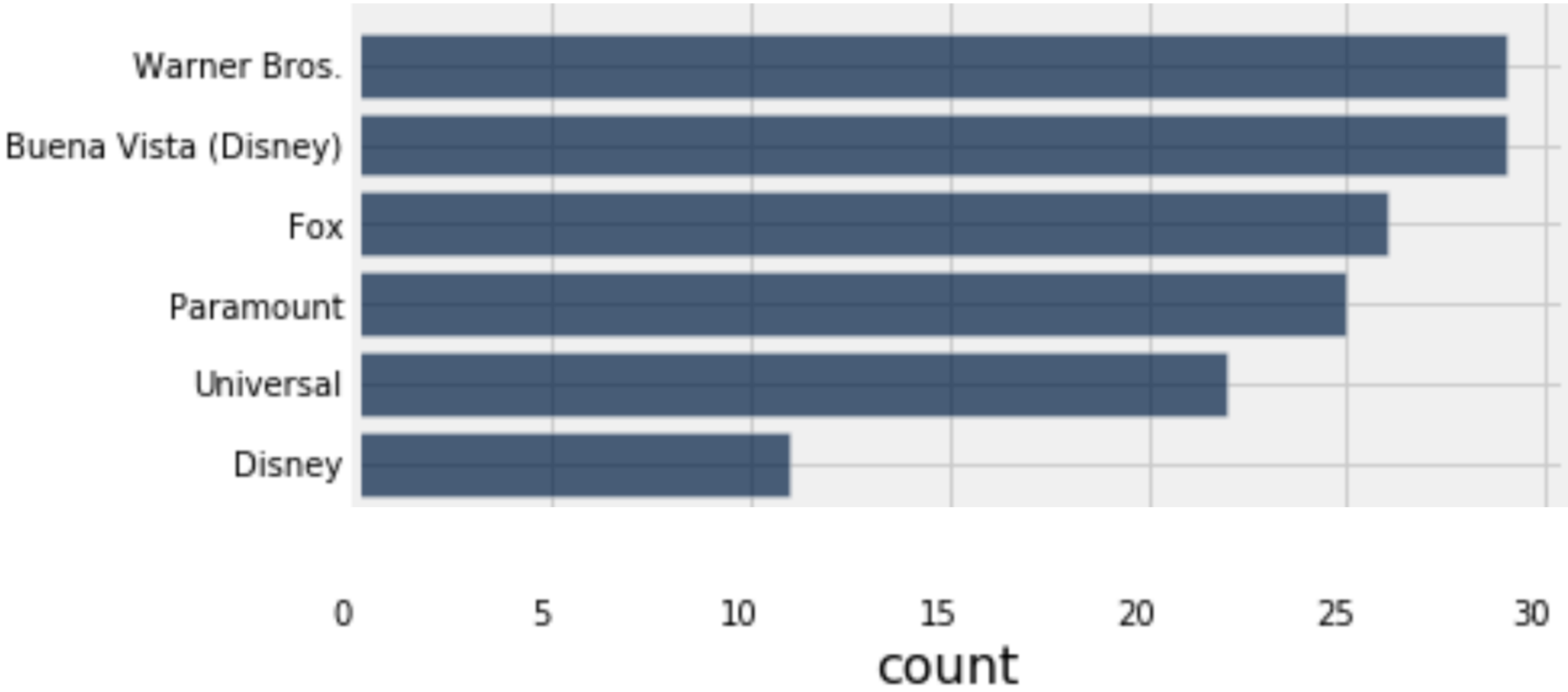
# Counts of Rows in each Category

```
movies_and_studios = top.select('Title', 'Studio')
movies_and_studios.group('Studio')
```

| Studio | count |
|---|---|
| AVCO | 1 |
| Buena Vista (Disney) | 29 |
| Columbia | 10 |
| Disney | 11 |
| Dreamworks | 3 |

# Bar Chart with Count

```
studio_distribution = movies_and_studios.group('Studio')
studio_distribution.sort('count', descending=True).barh('Studio')
```



# Measure the Adjusted Gross in U$ Millions

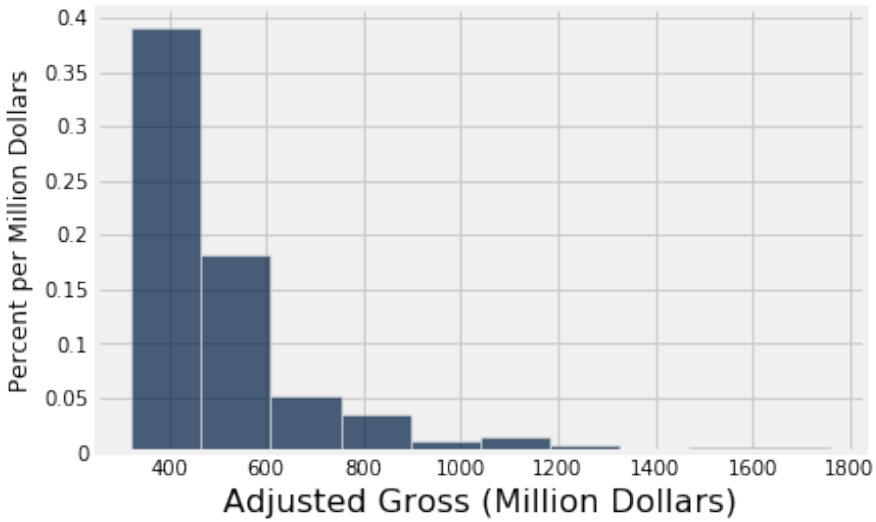
```
import numpy as np
millions = top.select(0).with_column('Adjusted Gross',
                                     np.round(top.column(3)/1e6, 2))
```

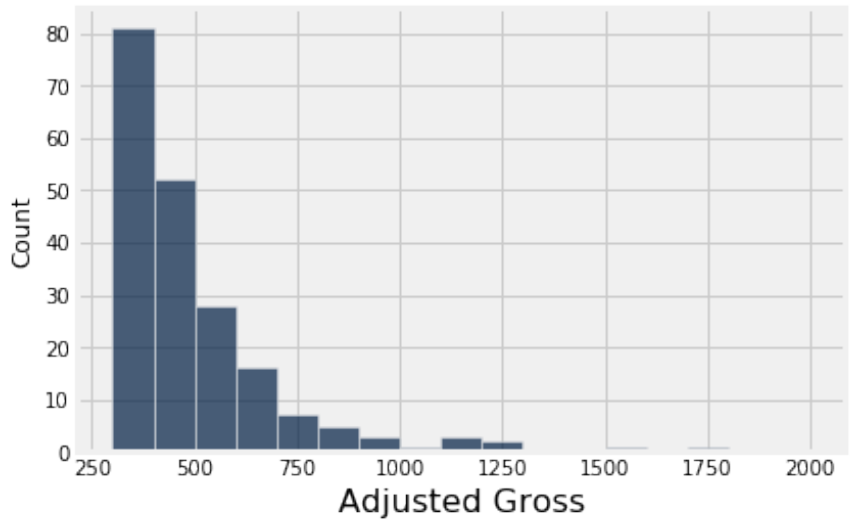
| Title | Adjusted Gross |
|---|---|
| Star Wars: The Force Awakens | 906.72 |
| Avatar | 846.12 |
| Titanic | 1178.63 |
| Jurassic World | 687.73 |

# Histogram

```
millions.hist('Adjusted Gross', unit="Million Dollars")
```

```
millions.hist('Adjusted Gross',
  bins=np.arange(300,2001,100), normed=False)
```



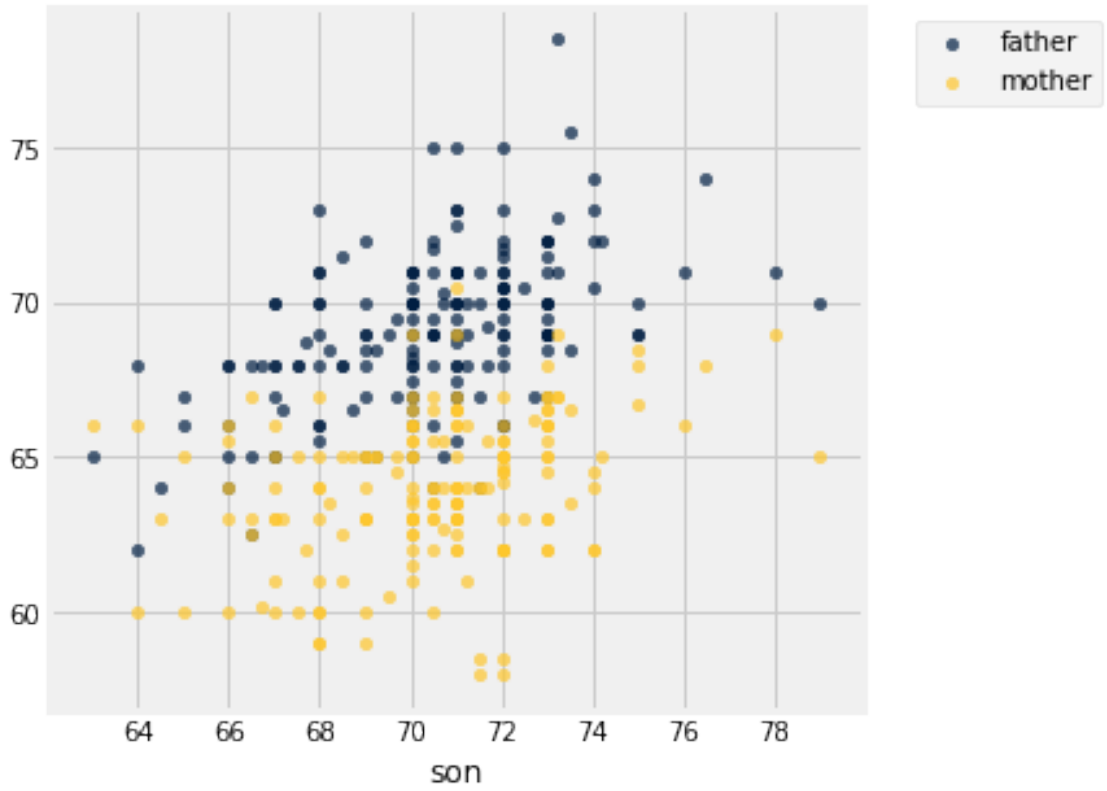



# Francis Galton (1822-1911)

```
heights = Table.read_table(path_data + 'galton_subset.csv')
```

```
heights.scatter('son')
```

| father | mother | son |
|---|---|---|
| 78.5 | 67 | 73.2 |
| 75.5 | 66.5 | 73.5 |
| 75 | 64 | 71 |
| 75 | 64 | 70.5 |
| 75 | 58.5 | 72 |



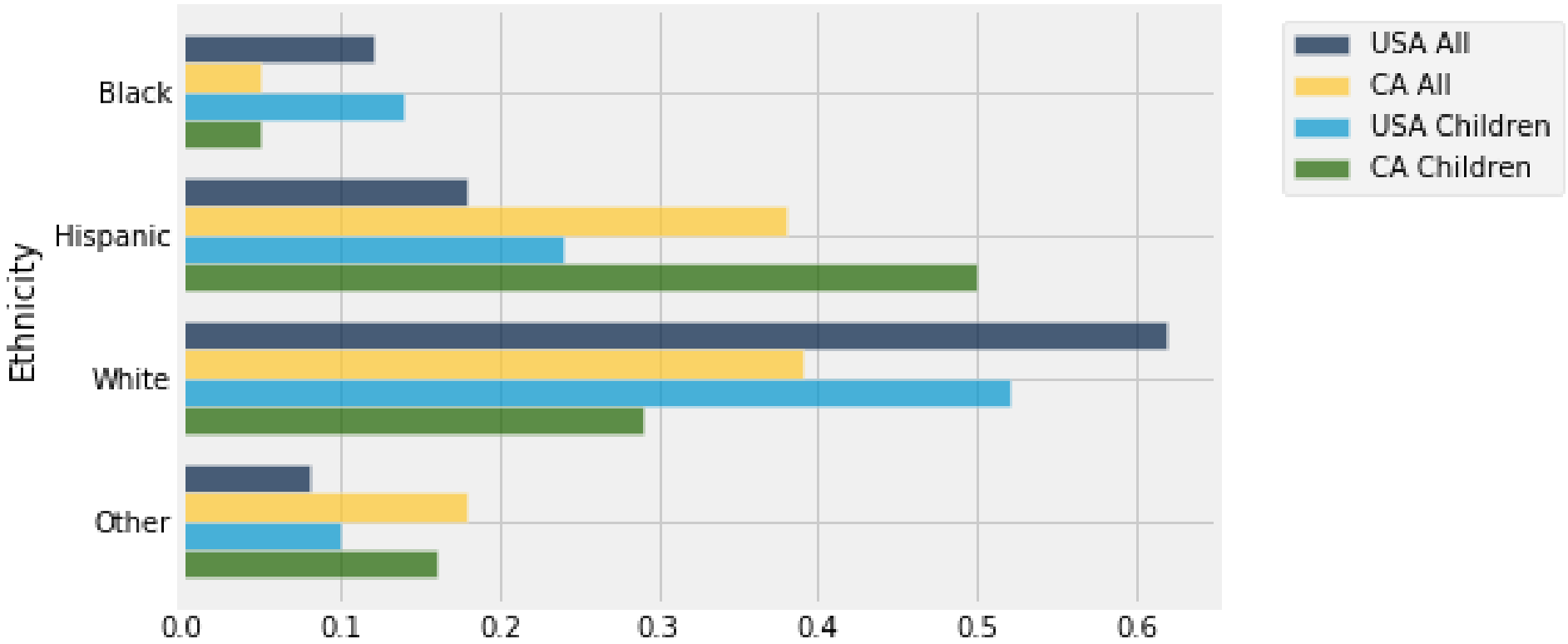
# Census data

```
usa_ca = Table.read_table(path_data + 'usa_ca_2014.csv')
```

| Ethnicity | USA All | CA All | USA Children | CA Children |
|---|---|---|---|---|
| Black | 0.12 | 0.05 | 0.14 | 0.05 |
| Hispanic | 0.18 | 0.38 | 0.24 | 0.5 |
| White | 0.62 | 0.39 | 0.52 | 0.29 |
| Other | 0.08 | 0.18 | 0.1 | 0.16 |

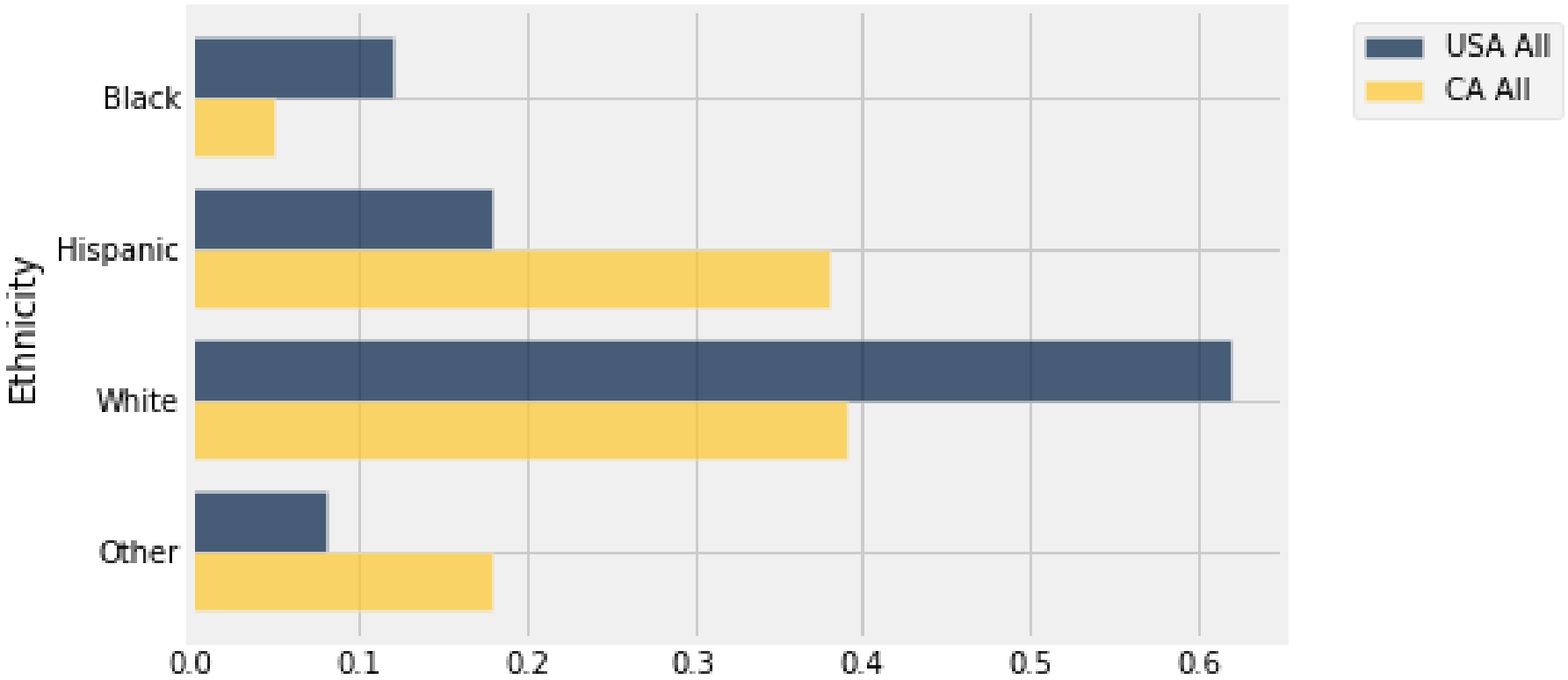
# Too Much Information on this Graph

```
usa_ca.barh('Ethnicity')
```



# Entire Populations of the USA vs California

```
usa_ca.select('Ethnicity', 'USA All', 'CA All').barh('Ethnicity')
```



# Californian Population vs Children

```
usa_ca.select('Ethnicity', 'CA All', 'CA Children').barh('Ethnicity')
```