Midterm 1

March 10, 2022

[235]:
```
from datascience import *
import numpy as np
import warnings
warnings.filterwarnings("ignore")

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

1 Question 1

[236]:
```
food = Table.read_table('caloriesMcD.csv')
food
```

[236]:
```
Category  | Item                                                  |
Calories | Calories from Fat | Total Fat | Saturated Fat | Saturated Fat (%
Daily Value) | Trans Fat | Cholesterol | Sodium | Carbohydrates | Dietary Fiber
| Sugars | Protein
Breakfast | Egg McMuffin                                          | 300
| 120              | 13        | 5             | 25
| 0         | 260         | 750    | 31            | 4             | 3      | 17
Breakfast | Egg White Delight                                     | 250
| 70               | 8         | 3             | 15
| 0         | 25          | 770    | 30            | 4             | 3      | 18
Breakfast | Sausage McMuffin                                      | 370
| 200              | 23        | 8             | 42
| 0         | 45          | 780    | 29            | 4             | 2      | 14
Breakfast | Sausage McMuffin with Egg                             | 450
| 250              | 28        | 10            | 52
| 0         | 285         | 860    | 30            | 4             | 2      | 21
Breakfast | Sausage McMuffin with Egg Whites                      | 400
| 210              | 23        | 8             | 42
| 0         | 50          | 880    | 30            | 4             | 2      | 21
Breakfast | Steak & Egg McMuffin                                  | 430
| 210              | 23        | 9             | 46
```

```
| 1         | 300         | 960    | 31            | 4             | 3      | 26
Breakfast | Bacon, Egg & Cheese Biscuit (Regular Biscuit)                | 460
| 230               | 26        | 13            | 65
| 0         | 250         | 1300   | 38            | 2             | 3      | 19
Breakfast | Bacon, Egg & Cheese Biscuit (Large Biscuit)                  | 520
| 270               | 30        | 14            | 68
| 0         | 250         | 1410   | 43            | 3             | 4      | 19
Breakfast | Bacon, Egg & Cheese Biscuit with Egg Whites (Regular Bis … | 410
| 180               | 20        | 11            | 56
| 0         | 35          | 1300   | 36            | 2             | 3      | 20
Breakfast | Bacon, Egg & Cheese Biscuit with Egg Whites (Large Biscuit)  | 470
| 220               | 25        | 12            | 59
| 0         | 35          | 1420   | 42            | 3             | 4      | 20
… (250 rows omitted)
```

[237]:
```
#A
sum(food.column('Calories'))
```

[237]:
```
95750
```

[238]:
```
#B
fat = food.where('Calories from Fat',are.above(200))
fat
```

[238]:
```
Category  | Item                                                         |
Calories | Calories from Fat | Total Fat | Saturated Fat | Saturated Fat (%
Daily Value) | Trans Fat | Cholesterol | Sodium | Carbohydrates | Dietary Fiber
| Sugars | Protein
Breakfast | Sausage McMuffin with Egg                                    | 450
| 250               | 28        | 10            | 52
| 0         | 285         | 860    | 30            | 4             | 2      | 21
Breakfast | Sausage McMuffin with Egg Whites                             | 400
| 210               | 23        | 8             | 42
| 0         | 50          | 880    | 30            | 4             | 2      | 21
Breakfast | Steak & Egg McMuffin                                         | 430
| 210               | 23        | 9             | 46
| 1         | 300         | 960    | 31            | 4             | 3      | 26
Breakfast | Bacon, Egg & Cheese Biscuit (Regular Biscuit)                | 460
| 230               | 26        | 13            | 65
| 0         | 250         | 1300   | 38            | 2             | 3      | 19
Breakfast | Bacon, Egg & Cheese Biscuit (Large Biscuit)                  | 520
| 270               | 30        | 14            | 68
| 0         | 250         | 1410   | 43            | 3             | 4      | 19
Breakfast | Bacon, Egg & Cheese Biscuit with Egg Whites (Large Biscuit) | 470
| 220               | 25        | 12            | 59
| 0         | 35          | 1420   | 42            | 3             | 4      | 20
Breakfast | Sausage Biscuit (Regular Biscuit)                            | 430
```

```
| 240                | 27         | 12             | 62
| 0          | 30           | 1080    | 34             | 2              | 2       | 11
Breakfast | Sausage Biscuit (Large Biscuit)                                 | 480
| 280                | 31         | 13             | 65
| 0          | 30           | 1190    | 39             | 3              | 3       | 11
Breakfast | Sausage Biscuit with Egg (Regular Biscuit)                      | 510
| 290                | 33         | 14             | 71
| 0          | 250          | 1170    | 36             | 2              | 2       | 18
Breakfast | Sausage Biscuit with Egg (Large Biscuit)                        | 570
| 330                | 37         | 15             | 74
| 0          | 250          | 1280    | 42             | 3              | 3       | 18
… (50 rows omitted)
```

[239]:
```
#C
sugar = food.sort("Sugars", descending = True)
sugar
```

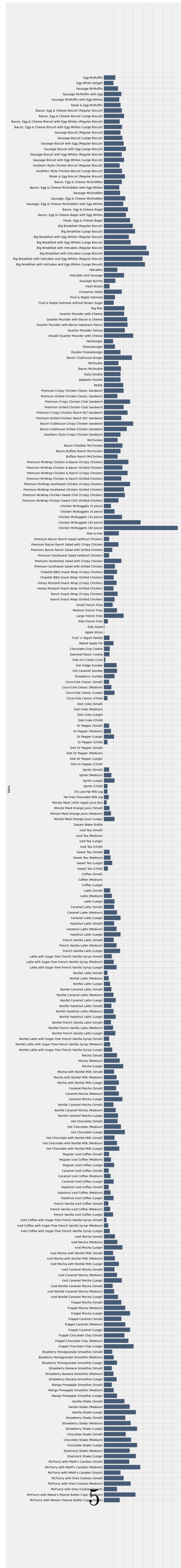
[239]:
```
Category           | Item                                                 |
Calories | Calories from Fat | Total Fat | Saturated Fat | Saturated Fat (%
Daily Value) | Trans Fat | Cholesterol | Sodium | Carbohydrates | Dietary Fiber
| Sugars | Protein
Smoothies & Shakes | McFlurry with M&M’s Candies (Medium)              | 930
| 290                | 33         | 20             | 102
| 1          | 75           | 260     | 139            | 2              | 128     | 20
Smoothies & Shakes | Strawberry Shake (Large)                             | 850
| 210                | 24         | 15             | 75
| 1          | 90           | 260     | 140            | 0              | 123     | 18
Smoothies & Shakes | Chocolate Shake (Large)                              | 850
| 210                | 23         | 15             | 74
| 1          | 85           | 380     | 141            | 2              | 120     | 19
Smoothies & Shakes | Shamrock Shake (Large)                               | 820
| 210                | 23         | 15             | 73
| 1          | 90           | 260     | 135            | 0              | 115     | 18
Smoothies & Shakes | McFlurry with Reese's Peanut Butter Cups (Medium) | 810
| 290                | 32         | 15             | 76
| 1          | 60           | 400     | 114            | 2              | 103     | 21
Smoothies & Shakes | Vanilla Shake (Large)                                | 820
| 210                | 23         | 15             | 73
| 1          | 90           | 260     | 135            | 0              | 101     | 18
Smoothies & Shakes | Strawberry Shake (Medium)                            | 690
| 180                | 20         | 13             | 63
| 1          | 75           | 210     | 114            | 0              | 100     | 15
Coffee & Tea       | Frappé Chocolate Chip (Large)                        | 760
| 280                | 31         | 20             | 101
| 1.5        | 95           | 200     | 111            | 1              | 99      | 12
Smoothies & Shakes | Chocolate Shake (Medium)                             | 700
| 180                | 20         | 12             | 62
```

```
| 1          | 75           | 300     | 114            | 2              | 97      | 15
Smoothies & Shakes | Shamrock Shake (Medium)                          | 660
| 170                | 19         | 12             | 61
| 1          | 75           | 210     | 109            | 0              | 93      | 14
… (250 rows omitted)
```

[240]:
```
#D
a = food.group('Category', np.max).select(0,3)
a
```

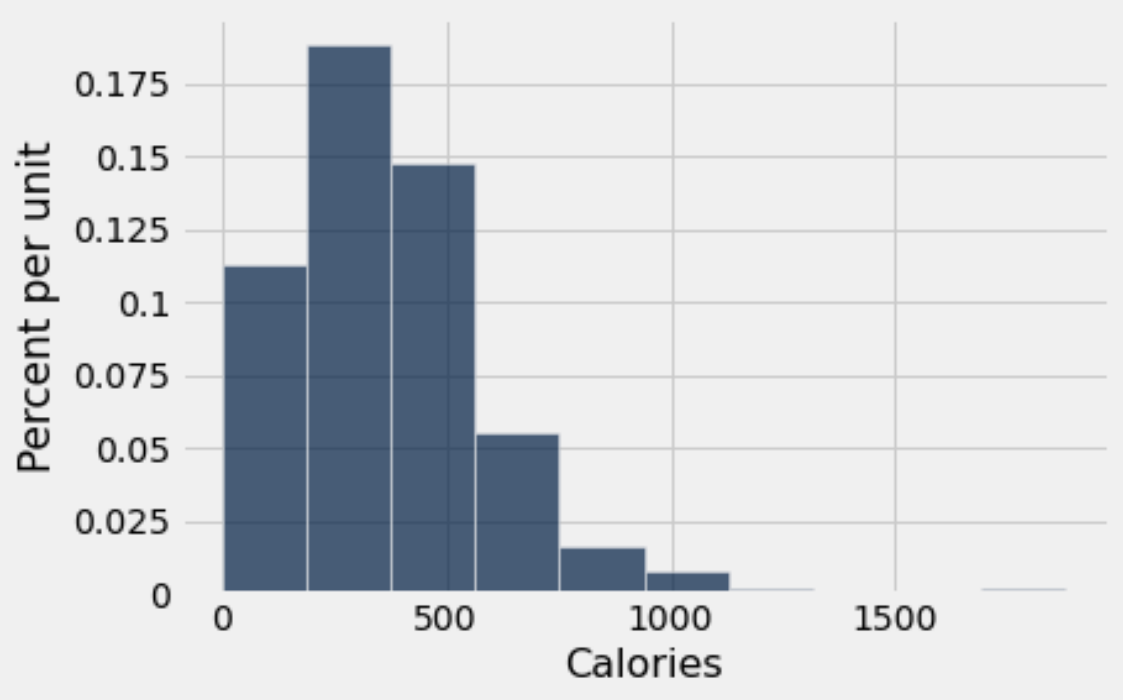
[240]:
```
Category           | Calories from Fat amax
Beef & Pork        | 380
Beverages          | 20
Breakfast          | 540
Chicken & Fish     | 1060
Coffee & Tea       | 280
Desserts           | 110
Salads             | 190
Smoothies & Shakes | 290
Snacks & Sides     | 220
```

[241]:
```
#E
food.barh('Item', 'Calories')
```


Item
Calories
0
250
500
750
1000
1250
1500
1750

Chicken Mcnuggets(40 piece) has the most calories and the diet sodas and ice tea have no calories

[242]:
```
food.hist(2)
```



[243]:
```
#F
category_item = food.select('Category', 'Item')
```

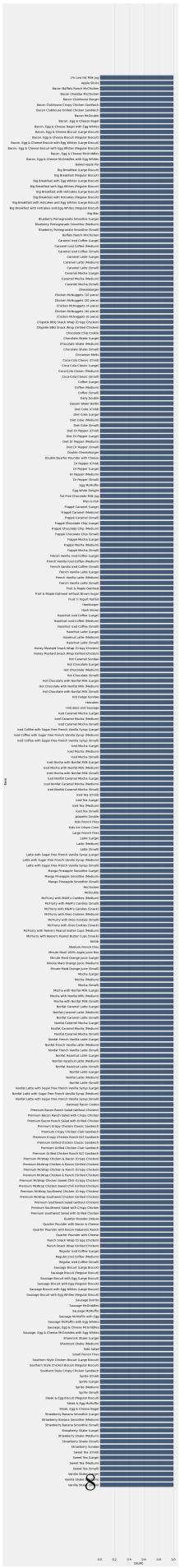
[244]:
```
item_distribution = category_item.group('Item')
item_distribution
```

[244]:
```
Item                                       | count
1% Low Fat Milk Jug                        | 1
Apple Slices                               | 1
Bacon Buffalo Ranch McChicken              | 1
Bacon Cheddar McChicken                    | 1
Bacon Clubhouse Burger                     | 1
Bacon Clubhouse Crispy Chicken Sandwich    | 1
Bacon Clubhouse Grilled Chicken Sandwich   | 1
Bacon McDouble                             | 1
Bacon, Egg & Cheese Bagel                  | 1
Bacon, Egg & Cheese Bagel with Egg Whites  | 1
… (250 rows omitted)
```

[245]:
```
sum(item_distribution.column('count'))
```

[245]: 260

[246]:
```
item_distribution.sort('count', descending=True).barh('Item')
```


1% Low Fat Milk Jug
Apple Slices
Bacon Buffalo Ranch McChicken
Bacon Cheddar McChicken
Bacon Clubhouse Burger
Bacon Clubhouse Crispy Chicken Sandwich
Bacon Clubhouse Grilled Chicken Sandwich
Bacon McDonald
Bacon, Egg & Cheese Bagel
Bacon, Egg & Cheese Bagel with Egg Whites
Bacon, Egg & Cheese Biscuit (Large Biscuit)
Bacon, Egg & Cheese Biscuit (Regular Biscuit)
Bacon, Egg & Cheese Biscuit with Egg Whites (Large Biscuit)
Bacon, Egg & Cheese Biscuit with Egg Whites (Regular Biscuit)
Bacon, Egg & Cheese McGriddles
Bacon, Egg & Cheese McGriddles with Egg Whites
Baked Apple Pie
Big Breakfast (Large Biscuit)
Big Breakfast (Regular Biscuit)
Big Breakfast with Egg Whites (Large Biscuit)
Big Breakfast with Egg Whites (Regular Biscuit)
Big Breakfast with Hotcakes (Large Biscuit)
Big Breakfast with Hotcakes (Regular Biscuit)
Big Breakfast with Hotcakes and Egg Whites (Large Biscuit)
Big Breakfast with Hotcakes and Egg Whites (Regular Biscuit)
Big Mac
Blueberry Pomegranate Smoothie (Large)
Blueberry Pomegranate Smoothie (Medium)
Blueberry Pomegranate Smoothie (Small)
Buffalo Ranch McChicken
Caramel Iced Coffee (Large)
Caramel Iced Coffee (Medium)
Caramel Iced Coffee (Small)
Caramel Latte (Large)
Caramel Latte (Medium)
Caramel Latte (Small)
Caramel Mocha (Large)
Caramel Mocha (Medium)
Caramel Mocha (Small)
Cheeseburger
Chicken McNuggets (10 piece)
Chicken McNuggets (20 piece)
Chicken McNuggets (4 piece)
Chicken McNuggets (40 piece)
Chicken McNuggets (6 piece)
Chipotle BBQ Snack Wrap (Crispy Chicken)
Chipotle BBQ Snack Wrap (Grilled Chicken)
Chocolate Chip Cookie
Chocolate Shake (Large)
Chocolate Shake (Medium)
Chocolate Shake (Small)
Cinnamon Melts
Coca-Cola Classic (Child)
Coca-Cola Classic (Large)
Coca-Cola Classic (Medium)
Coca-Cola Classic (Small)
Coffee (Large)
Coffee (Medium)
Coffee (Small)
Dairy Double
Dasani Water Bottle
Diet Coke (Child)
Diet Coke (Large)
Diet Coke (Medium)
Diet Coke (Small)
Diet Dr Pepper (Child)
Diet Dr Pepper (Large)
Diet Dr Pepper (Medium)
Diet Dr Pepper (Small)
Double Cheeseburger
Double Quarter Pounder with Cheese
Dr Pepper (Child)
Dr Pepper (Large)
Dr Pepper (Medium)
Dr Pepper (Small)
Egg McMuffin
Egg White Delight
Fat Free Chocolate Milk
Fish (Child)
Frappé Caramel (Large)
Frappé Caramel (Medium)
Frappé Caramel (Small)
Frappé Chocolate Chip (Large)
Frappé Chocolate Chip (Medium)
Frappé Chocolate Chip (Small)
Frappé Mocha (Large)
Frappé Mocha (Medium)
Frappé Mocha (Small)
French Vanilla Iced Coffee (Large)
French Vanilla Iced Coffee (Medium)
French Vanilla Iced Coffee (Small)
French Vanilla Latte (Large)
French Vanilla Latte (Medium)
French Vanilla Latte (Small)
Fruit & Maple Oatmeal without Brown Sugar
Fruit 'n Yogurt Parfait
Hamburger
Hash Brown
Hazelnut Iced Coffee (Large)
Hazelnut Iced Coffee (Medium)
Hazelnut Iced Coffee (Small)
Hazelnut Latte (Large)
Hazelnut Latte (Medium)
Hazelnut Latte (Small)
Honey Mustard Snack Wrap (Crispy Chicken)
Honey Mustard Snack Wrap (Grilled Chicken)
Hot Caramel Sundae
Hot Chocolate (Large)
Hot Chocolate (Medium)
Hot Chocolate (Small)
Hot Chocolate with Nonfat Milk (Large)
Hot Chocolate with Nonfat Milk (Medium)
Hot Chocolate with Nonfat Milk (Small)
Hot Fudge Sundae
Hotcakes
Hotcakes and Sausage
Iced Caramel Mocha (Large)
Iced Caramel Mocha (Medium)
Iced Caramel Mocha (Small)
Iced Coffee with Sugar Free French Vanilla Syrup (Large)
Iced Coffee with Sugar Free French Vanilla Syrup (Medium)
Iced Coffee with Sugar Free French Vanilla Syrup (Small)
Iced Mocha (Large)
Iced Mocha (Medium)
Iced Mocha (Small)
Iced Mocha with Nonfat Milk (Large)
Iced Mocha with Nonfat Milk (Medium)
Iced Mocha with Nonfat Milk (Small)
Iced Nonfat Caramel Mocha (Large)
Iced Nonfat Caramel Mocha (Medium)
Iced Nonfat Caramel Mocha (Small)
Iced Tea (Child)
Iced Tea (Large)
Iced Tea (Medium)
Iced Tea (Small)
Jalapeño Double
Kids French Fries
Kids Ice Cream Cone
Large French Fries
Latte (Large)
Latte (Medium)
Latte (Small)
Latte with Sugar Free French Vanilla Syrup (Large)
Latte with Sugar Free French Vanilla Syrup (Medium)
Latte with Sugar Free French Vanilla Syrup (Small)
Mango Pineapple Smoothie (Large)
Mango Pineapple Smoothie (Medium)
Mango Pineapple Smoothie (Small)
McDonaldland Cookies
McDouble
McFlurry with M&M's Candies (Medium)
McFlurry with M&M's Candies (Small)
McFlurry with Oreo Cookies (Medium)
McFlurry with Oreo Cookies (Small)
McFlurry with Oreo Cookies (Snack)
McFlurry with Reese's Peanut Butter Cups (Medium)
McFlurry with Reese's Peanut Butter Cups (Snack)
McRib
Medium French Fries
Minute Maid 100% Apple Juice Box
Minute Maid Orange Juice (Large)
Minute Maid Orange Juice (Medium)
Minute Maid Orange Juice (Small)
Mocha (Large)
Mocha (Medium)
Mocha (Small)
Mocha with Nonfat Milk (Large)
Mocha with Nonfat Milk (Medium)
Mocha with Nonfat Milk (Small)
Nonfat Caramel Latte (Large)
Nonfat Caramel Latte (Medium)
Nonfat Caramel Latte (Small)
Nonfat Caramel Mocha (Large)
Nonfat Caramel Mocha (Medium)
Nonfat Caramel Mocha (Small)
Nonfat French Vanilla Latte (Large)
Nonfat French Vanilla Latte (Medium)
Nonfat French Vanilla Latte (Small)
Nonfat Hazelnut Latte (Large)
Nonfat Hazelnut Latte (Medium)
Nonfat Hazelnut Latte (Small)
Nonfat Latte (Large)
Nonfat Latte (Medium)
Nonfat Latte (Small)
Nonfat Latte with Sugar Free French Vanilla Syrup (Large)
Nonfat Latte with Sugar Free French Vanilla Syrup (Medium)
Nonfat Latte with Sugar Free French Vanilla Syrup (Small)
Oatmeal Raisin Cookie
Premium Bacon Ranch Salad (without Chicken)
Premium Bacon Ranch Salad with Crispy Chicken
Premium Bacon Ranch Salad with Grilled Chicken
Premium Crispy Chicken Classic Sandwich
Premium Crispy Chicken Club Sandwich
Premium Crispy Chicken Ranch BLT Sandwich
Premium Grilled Chicken Classic Sandwich
Premium Grilled Chicken Club Sandwich
Premium Grilled Chicken Ranch BLT Sandwich
Premium McWrap Chicken & Bacon (Crispy Chicken)
Premium McWrap Chicken & Bacon (Grilled Chicken)
Premium McWrap Chicken & Ranch (Crispy Chicken)
Premium McWrap Chicken & Ranch (Grilled Chicken)
Premium McWrap Chicken Sweet Chili (Crispy Chicken)
Premium McWrap Chicken Sweet Chili (Grilled Chicken)
Premium McWrap Southwest Chicken (Crispy Chicken)
Premium McWrap Southwest Chicken (Grilled Chicken)
Premium Southwest Salad (without Chicken)
Premium Southwest Salad with Crispy Chicken
Premium Southwest Salad with Grilled Chicken
Quarter Pounder (Deluxe)
Quarter Pounder with Bacon & Cheese
Quarter Pounder with Bacon Habanero Ranch
Quarter Pounder with Cheese
Ranch Snack Wrap (Crispy Chicken)
Ranch Snack Wrap (Grilled Chicken)
Regular Iced Coffee (Large)
Regular Iced Coffee (Medium)
Regular Iced Coffee (Small)
Sausage Biscuit (Large Biscuit)
Sausage Biscuit (Regular Biscuit)
Sausage Biscuit with Egg (Large Biscuit)
Sausage Biscuit with Egg (Regular Biscuit)
Sausage Biscuit with Egg Whites (Large Biscuit)
Sausage Biscuit with Egg Whites (Regular Biscuit)
Sausage Burrito
Sausage McGriddles
Sausage McMuffin
Sausage McMuffin with Egg
Sausage McMuffin with Egg Whites
Sausage, Egg & Cheese McGriddles
Sausage, Egg & Cheese McGriddles with Egg Whites
Shamrock Shake (Large)
Shamrock Shake (Medium)
Side Salad
Small French Fries
Southern Style Chicken Biscuit (Large Biscuit)
Southern Style Chicken Biscuit (Regular Biscuit)
Southern Style Crispy Chicken Sandwich
Sprite (Child)
Sprite (Large)
Sprite (Medium)
Sprite (Small)
Steak & Egg Biscuit (Regular Biscuit)
Steak & Egg McMuffin
Steak, Egg & Cheese Bagel
Strawberry Banana Smoothie (Large)
Strawberry Banana Smoothie (Medium)
Strawberry Banana Smoothie (Small)
Strawberry Shake (Large)
Strawberry Shake (Medium)
Strawberry Shake (Small)
Strawberry Sundae
Sweet Tea (Child)
Sweet Tea (Large)
Sweet Tea (Medium)
Sweet Tea (Small)
Vanilla Shake (Large)
Vanilla Shake (Medium)
Vanilla Shake (Small)
Item
0.0
0.2
0.4
0.6
0.8
1.0
count

[ ]:

2 Question 2

[339]:
```
comp = Table.read_table('raw_compensation.csv')
comp
```

[339]:
```
Rank | Name                | Company (Headquarters)          | Total Pay | % Change           | Cash Pay | Equity Pay | Other Pay | Ratio of CEO pay to average industry worker pay
1    | Mark V. Hurd*       | Oracle (Redwood City)           | $53.25    | (No previous year) | $0.95    | $52.27     | $0.02     | 362
2    | Safra A. Catz*      | Oracle (Redwood City)           | $53.24    | (No previous year) | $0.95    | $52.27     | $0.02     | 362
3    | Robert A. Iger      | Walt Disney (Burbank)           | $44.91    | -3%                | $24.89   | $17.28     | $2.74     | 477
4    | Marissa A. Mayer    | Yahoo! (Sunnyvale)              | $35.98    | -15%               | $1.00    | $34.43     | $0.55     | 342
5    | Marc Benioff        | salesforce.com (San Francisco)  | $33.36    | -16%               | $4.65    | $27.26     | $1.45     | 338
6    | John H. Hammergren  | McKesson (San Francisco)        | $24.84    | -4%                | $12.10   | $12.37     | $0.37     | 222
7    | John S. Watson      | Chevron (San Ramon)             | $22.04    | -15%               | $4.31    | $14.68     | $3.05     | 183
8    | Jeffrey Weiner      | LinkedIn (Mountain View)        | $19.86    | 27%                | $2.47    | $17.26     | $0.13     | 182
9    | John T. Chambers**  | Cisco Systems (San Jose)        | $19.62    | 19%                | $5.10    | $14.51     | $0.01     | 170
10   | John G. Stumpf      | Wells Fargo  (San Francisco)    | $19.32    | -10%               | $6.80    | $12.50     | $0.02     | 256
… (92 rows omitted)
```

[340]:
```
#A
comp5 = Table.read_table('raw_compensation.csv').show(5)
comp5
```

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

[341]:
```
#B
def string_num(Total_Pay):
    return int(float(Total_Pay.strip('$')))
```

[342]:
```
string_num('$2')
```

[342]: 2

[343]:
```
Total_Pay = comp.apply(string_num,"Total Pay")
Total_Pay
```

[343]:
```
array([53, 53, 44, 35, 33, 24, 22, 19, 19, 19, 18, 18, 18, 18, 17, 16, 16,
       16, 16, 16, 15, 14, 14, 14, 14, 14, 14, 14, 14, 13, 12, 12, 12, 12,
       12, 11, 11, 11, 11, 11, 11, 10, 10, 10, 10, 10, 10, 10, 10,  9,  9,
        9,  9,  9,  9,  9,  8,  8,  8,  8,  8,  8,  8,  7,  7,  7,
        7,  7,  7,  7,  6,  6,  6,  6,  6,  5,  5,  5,  5,  5,  5,  4,  4,
        4,  4,  4,  3,  3,  2,  2,  2,  2,  1,  1,  1,  0,  0,  0,  0,  0])
```

[344]:
```
#C
comp = comp.with_column("Total_Pay", Total_Pay)
comp
```

[344]:
```
Rank | Name               | Company (Headquarters)         | Total Pay | %
Change           | Cash Pay | Equity Pay | Other Pay | Ratio of CEO pay to
average industry worker pay | Total_Pay
1    | Mark V. Hurd*      | Oracle (Redwood City)          | $53.25    | (No
previous year) | $0.95    | $52.27     | $0.02     | 362
| 53
2    | Safra A. Catz*     | Oracle (Redwood City)          | $53.24    | (No
previous year) | $0.95    | $52.27     | $0.02     | 362
| 53
3    | Robert A. Iger     | Walt Disney (Burbank)          | $44.91    | -3%
| $24.89   | $17.28     | $2.74     | 477
| 44
4    | Marissa A. Mayer   | Yahoo! (Sunnyvale)             | $35.98    | -15%
| $1.00    | $34.43     | $0.55     | 342
| 35
5    | Marc Benioff       | salesforce.com (San Francisco) | $33.36    | -16%
| $4.65    | $27.26     | $1.45     | 338
| 33
6    | John H. Hammergren | McKesson (San Francisco)       | $24.84    | -4%
| $12.10   | $12.37     | $0.37     | 222
| 24
7    | John S. Watson     | Chevron (San Ramon)            | $22.04    | -15%
| $4.31    | $14.68     | $3.05     | 183
| 22
8    | Jeffrey Weiner     | LinkedIn (Mountain View)       | $19.86    | 27%
| $2.47    | $17.26     | $0.13     | 182
| 19
9    | John T. Chambers** | Cisco Systems (San Jose)       | $19.62    | 19%
| $5.10    | $14.51     | $0.01     | 170
| 19
10   | John G. Stumpf     | Wells Fargo  (San Francisco)   | $19.32    | -10%
```

```
| $6.80    | $12.50     | $0.02     | 256
| 19
… (92 rows omitted)
```

[345]:
```
compnew = comp.with_column("Total Pay", comp.apply(string_num,"Total Pay"))
compnew
```

[345]:
```
Rank | Name               | Company (Headquarters)         | Total Pay | %
Change           | Cash Pay | Equity Pay | Other Pay | Ratio of CEO pay to
average industry worker pay | Total_Pay
1    | Mark V. Hurd*      | Oracle (Redwood City)          | 53        | (No
previous year) | $0.95    | $52.27     | $0.02     | 362
| 53
2    | Safra A. Catz*     | Oracle (Redwood City)          | 53        | (No
previous year) | $0.95    | $52.27     | $0.02     | 362
| 53
3    | Robert A. Iger     | Walt Disney (Burbank)          | 44        | -3%
| $24.89   | $17.28     | $2.74     | 477
| 44
4    | Marissa A. Mayer   | Yahoo! (Sunnyvale)             | 35        | -15%
| $1.00    | $34.43     | $0.55     | 342
| 35
5    | Marc Benioff       | salesforce.com (San Francisco) | 33        | -16%
| $4.65    | $27.26     | $1.45     | 338
| 33
6    | John H. Hammergren | McKesson (San Francisco)       | 24        | -4%
| $12.10   | $12.37     | $0.37     | 222
| 24
7    | John S. Watson     | Chevron (San Ramon)            | 22        | -15%
| $4.31    | $14.68     | $3.05     | 183
| 22
8    | Jeffrey Weiner     | LinkedIn (Mountain View)       | 19        | 27%
| $2.47    | $17.26     | $0.13     | 182
| 19
9    | John T. Chambers** | Cisco Systems (San Jose)       | 19        | 19%
| $5.10    | $14.51     | $0.01     | 170
| 19
10   | John G. Stumpf     | Wells Fargo  (San Francisco)   | 19        | -10%
| $6.80    | $12.50     | $0.02     | 256
| 19
… (92 rows omitted)
```

[346]:
```
#D
average_total_pay = np.average(comp.column("Total_Pay"))
average_total_pay
```

[346]:
```
10.970588235294118
```

Because of the $ sign being a string it wont let us compute the average.

[364]:
```
#E
CEOavg = comp.sort(Total_Pay,are.above(15)).show(21)
CEOavg
```

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

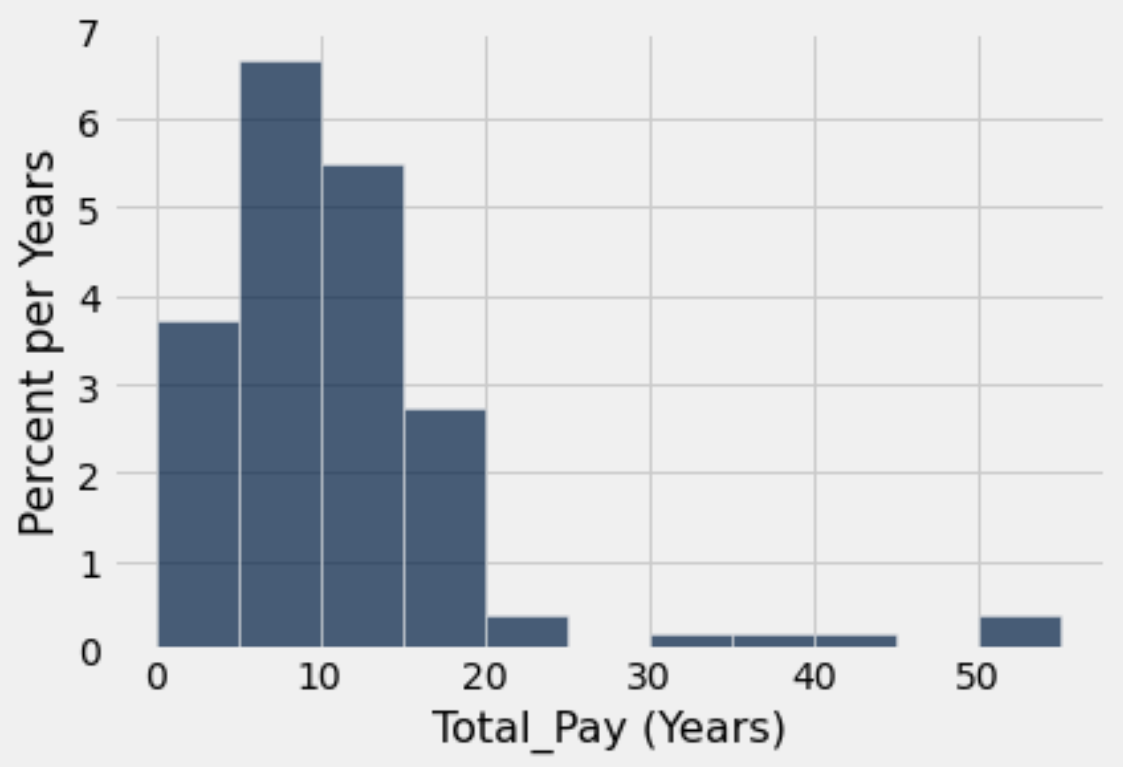
21 CEO made more then 15 million

[365]:
```
#F
my_bins = make_array(0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55)
```

[366]:
```
comp.bin('Total_Pay', bins = my_bins)
```

[366]:
```
bin  | Total_Pay count
0    | 19
5    | 34
10   | 28
15   | 14
20   | 2
25   | 0
30   | 1
35   | 1
40   | 1
45   | 0
… (2 rows omitted)
```

[367]:
```
comp.hist('Total_Pay', bins = my_bins, unit = 'Years')
```



I would say the histogram is skewed to the left since most of the data is in the begining of the histogram

[387]:
```
#G
(50-30)*0.25
```

[387]: 5.0

[388]:
```
#H
comp.where('Total_Pay', are.between(30, 50)).num_rows
```

[388]: 3

[389]:
```
comp.num_rows
```

[389]: 102

[390]:
```
3*100/(102)
```

[390]: 2.9411764705882355

[351]:
```
comp.where(Total_Pay,are.above(30))
```

[351]:
```
Rank | Name              | Company (Headquarters)          | Total Pay | % Change
| Cash Pay | Equity Pay | Other Pay | Ratio of CEO pay to average industry
worker pay | Total_Pay
1    | Mark V. Hurd*     | Oracle (Redwood City)           | $53.25    | (No
previous year) | $0.95    | $52.27     | $0.02     | 362
| 53
2    | Safra A. Catz*    | Oracle (Redwood City)           | $53.24    | (No
previous year) | $0.95    | $52.27     | $0.02     | 362
| 53
3    | Robert A. Iger    | Walt Disney (Burbank)           | $44.91    | -3%
| $24.89   | $17.28     | $2.74     | 477
| 44
4    | Marissa A. Mayer  | Yahoo! (Sunnyvale)              | $35.98    | -15%
| $1.00    | $34.43     | $0.55     | 342
| 35
5    | Marc Benioff      | salesforce.com (San Francisco) | $33.36    | -16%
| $4.65    | $27.26     | $1.45     | 338
| 33
```

[352]:
```
#I
type(comp.column("Cash Pay")[0])
```

[352]:
```
numpy.str_
```

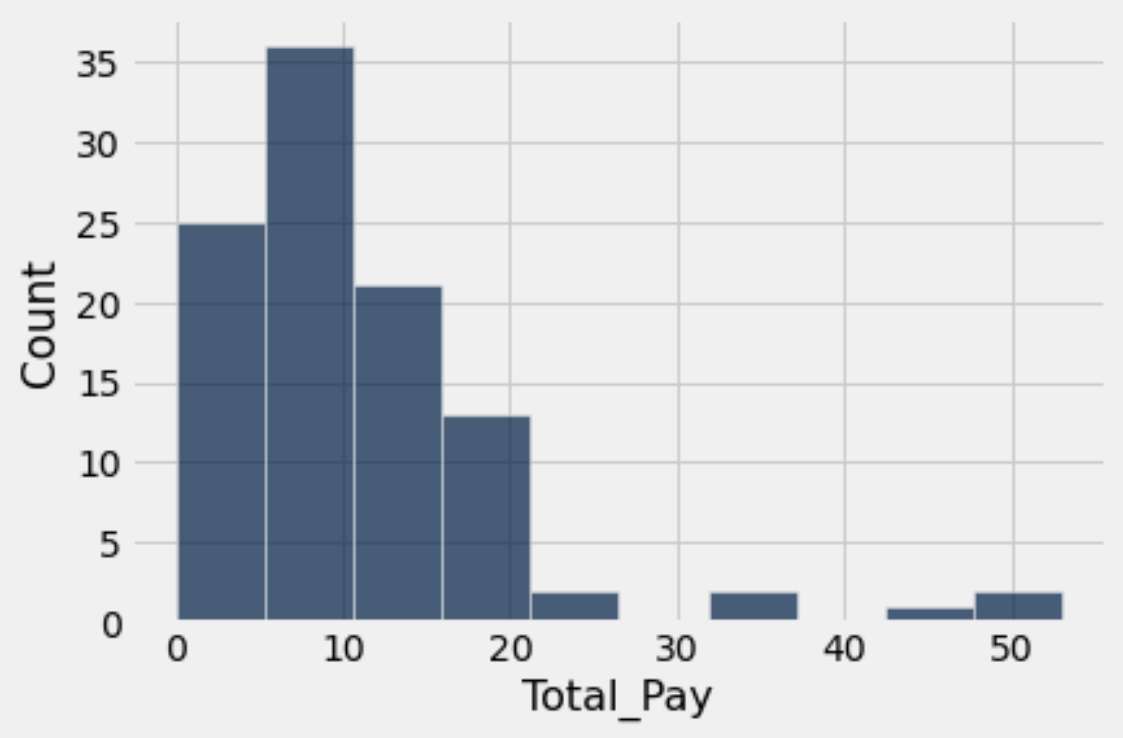
[353]:
```
cash_proportion = comp.apply(string_num, "Cash Pay") / comp.column("Total_Pay")
cash_proportion
```

[353]:
```
array([0.        , 0.        , 0.54545455, 0.02857143, 0.12121212,
       0.5       , 0.18181818, 0.10526316, 0.26315789, 0.31578947,
       0.27777778, 0.22222222, 0.11111111, 0.16666667, 0.17647059,
       0.0625    , 0.125     , 0.25      , 0.3125    , 0.125     ,
       0.26666667, 0.21428571, 0.28571429, 0.21428571, 0.21428571,
       0.14285714, 0.28571429, 0.21428571, 0.21428571, 0.15384615,
       0.33333333, 0.25      , 0.16666667, 0.41666667, 0.41666667,
       0.36363636, 0.18181818, 0.18181818, 0.36363636, 0.18181818,
       0.18181818, 0.1       , 0.3       , 0.1       , 0.2       ,
       1.        , 0.2       , 0.2       , 0.3       , 0.22222222,
       0.77777778, 0.33333333, 0.22222222, 0.22222222, 0.11111111,
       0.22222222, 0.33333333, 0.33333333, 0.5       , 0.125     ,
       0.375     , 0.25      , 0.25      , 0.125     , 0.375     ,
       0.28571429, 0.42857143, 0.28571429, 0.14285714, 1.        ,
       0.14285714, 0.85714286, 0.33333333, 0.83333333, 0.16666667,
       0.16666667, 0.16666667, 0.4       , 0.2       , 0.2       ,
       0.6       , 0.2       , 0.        , 0.5       , 0.5       ,
       0.25      , 0.        , 0.25      , 0.33333333, 0.33333333,
       1.        , 0.5       , 0.5       , 0.5       , 0.        ,
       1.        , 1.        ,       nan,       nan,       nan,
```

```
                nan,        nan])
```

[354]:
```
def percent_string_to_num(percent_string):return float(percent_string.
 ↪strip("%"))
```

[355]:
```
comp.hist("Total_Pay", normed= False)
```



3 Question 3

[415]:
```
imdb = Table.read_table('imdb.csv')
imdb
```

[415]:

Votes	Rating	Title	Year	Decade
88355	8.4	M	1931	1930
132823	8.3	Singin' in the Rain	1952	1950
74178	8.3	All About Eve	1950	1950
635139	8.6	Léon	1994	1990
145514	8.2	The Elephant Man	1980	1980
425461	8.3	Full Metal Jacket	1987	1980
441174	8.1	Gone Girl	2014	2010
850601	8.3	Batman Begins	2005	2000
37664	8.2	Judgment at Nuremberg	1961	1960
46987	8	Relatos salvajes	2014	2010

… (240 rows omitted)

[229]:
```
#A
forties = imdb.where('Decade', are.equal_to(1940))
forties
```

[229]:
```
Votes  | Rating | Title                           | Year | Decade
55793  | 8.1    | The Grapes of Wrath             | 1940 | 1940
86715  | 8.3    | Double Indemnity                | 1944 | 1940
101754 | 8.1    | The Maltese Falcon              | 1941 | 1940
71003  | 8.3    | The Treasure of the Sierra Madre | 1948 | 1940
35983  | 8.1    | The Best Years of Our Lives     | 1946 | 1940
81887  | 8.3    | Ladri di biciclette             | 1948 | 1940
66622  | 8      | Notorious                       | 1946 | 1940
350551 | 8.5    | Casablanca                      | 1942 | 1940
59578  | 8      | The Big Sleep                   | 1946 | 1940
78216  | 8.2    | Rebecca                         | 1940 | 1940
… (4 rows omitted)
```

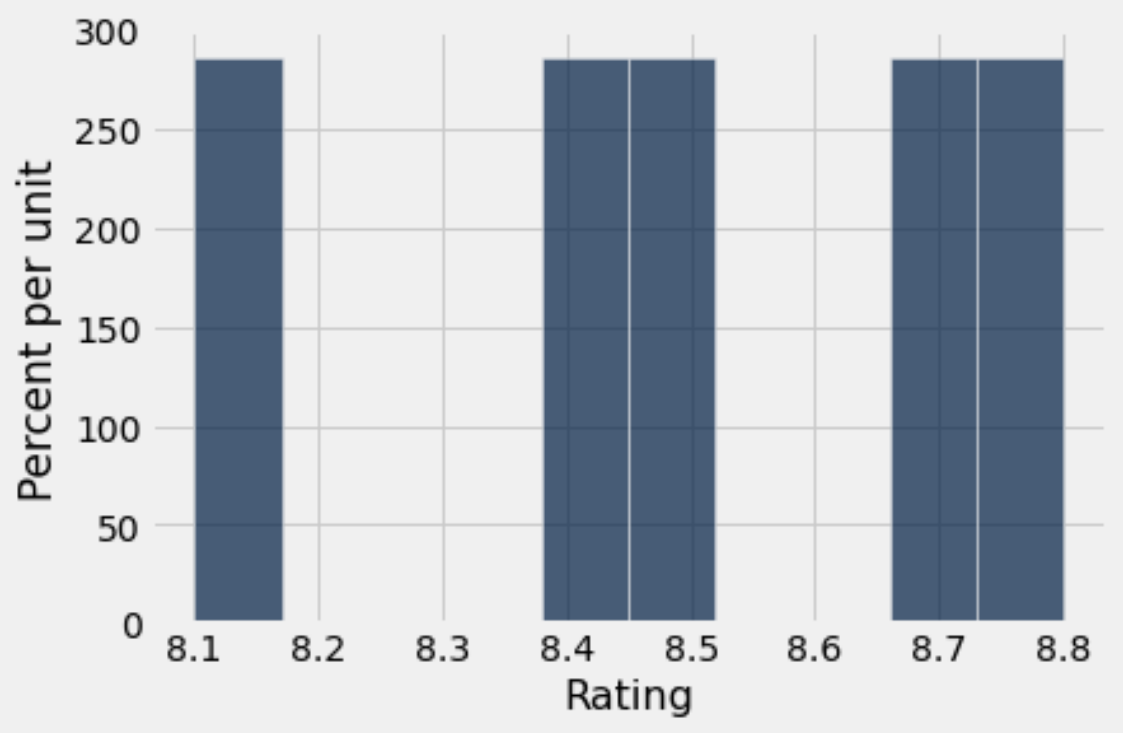
[230]:
```
#B
average_rating_in_forties = np.average(forties.column("Rating"))
average_rating_in_forties
```

[230]:
```
8.257142857142856
```

[231]:
```
#C
ninety_nine = imdb.where('Year', are.equal_to(1999))
ninety_nine
```

[231]:
```
Votes   | Rating | Title           | Year | Decade
1177098 | 8.8    | Fight Club      | 1999 | 1990
735056  | 8.4    | American Beauty | 1999 | 1990
630994  | 8.1    | The Sixth Sense | 1999 | 1990
1073043 | 8.7    | The Matrix      | 1999 | 1990
672878  | 8.5    | The Green Mile  | 1999 | 1990
```

[232]:
```
ninety_nine.hist('Rating')
```



yes but we need frequences of rating if we are to make a bar chart for Ratings

[233]:
```
#D
really_highly_rated = imdb.where('Rating',are.above(8.5))
really_highly_rated
```

[233]:
```
Votes   | Rating | Title                                  | Year | Decade
635139  | 8.6    | Léon                                   | 1994 | 1990
1027398 | 9.2    | The Godfather                          | 1972 | 1970
767224  | 8.6    | The Silence of the Lambs               | 1991 | 1990
1498733 | 9.2    | The Shawshank Redemption               | 1994 | 1990
447875  | 8.9    | Il buono, il brutto, il cattivo (1966) | 1966 | 1960
967389  | 8.7    | The Lord of the Rings: The Two Towers  | 2002 | 2000
689541  | 8.6    | Interstellar                           | 2014 | 2010
1473049 | 8.9    | The Dark Knight                        | 2008 | 2000
192206  | 8.6    | C'era una volta il West                | 1968 | 1960
1271949 | 8.7    | Inception                              | 2010 | 2010
… (19 rows omitted)
```

[394]:
```
#E
rating = imdb.column("Rating")
rating
```

[394]:
```
array([8.4, 8.3, 8.3, 8.6, 8.2, 8.3, 8.1, 8.3, 8.2, 8. , 8.1, 8.2, 8.3,
       8.3, 8.1, 8.4, 8.5, 8.2, 8.1, 8.4, 8.1, 8.1, 9.2, 8. , 8.2, 8.1,
       8.2, 8.5, 8. , 8.3, 8.1, 8. , 8. , 8.3, 8.1, 8. , 8. , 8.3, 8.4,
       8.1, 8.1, 8.5, 8.5, 8. , 8.3, 8.1, 8. , 8.6, 8.5, 8.3, 8.3, 8. ,
       8.2, 9.2, 8.2, 8.5, 8. , 8.9, 8.4, 8.2, 8.1, 8.3, 8.1, 8.1, 8.1,
       8.3, 8.2, 8.3, 8.7, 8.3, 8.6, 8. , 8.1, 8.2, 8.5, 8.3, 8.9, 8. ,
       8.6, 8.3, 8.1, 8.7, 8.4, 8.1, 8.4, 8. , 8.5, 8.8, 8.2, 8.2, 8.5,
       9. , 8. , 8. , 8.3, 8.4, 8.6, 8.5, 8.7, 8.4, 8.1, 8.1, 8.1, 8.7,
       8.4, 8.9, 8.1, 8.2, 8. , 8.5, 8.5, 8. , 8. , 8.4, 8.1, 8.1, 8. ,
       8. , 8.3, 8.1, 8. , 8.3, 8. , 8. , 8. , 8. , 8. , 8. , 8. , 8.7,
       8.3, 8. , 8. , 8.5, 8. , 8.1, 8.1, 8.1, 8.3, 8.2, 8.3, 8.9, 8.2,
       8.2, 8. , 8.3, 8.2, 8.9, 8.5, 8.5, 8.1, 8.1, 8.5, 8.3, 8. , 8.2,
       8.7, 8.3, 8.5, 8.1, 8.3, 8.2, 8.4, 8.1, 8.1, 8.1, 8. , 8.2, 8. ,
       8.6, 8.3, 8.2, 8. , 8.3, 8. , 8.2, 8. , 8.2, 8.8, 8.1, 8. , 8.1,
       8. , 8.2, 8.5, 8.1, 8.4, 8.1, 8.1, 8.7, 8.2, 8. , 8. , 8. , 8.3,
       8.4, 8. , 8.5, 8.1, 8.1, 8.2, 8.2, 8.4, 8.3, 8.6, 8.2, 8. , 8.1,
       8.2, 8.1, 8.3, 8.4, 8.5, 8.6, 8. , 8.3, 8.5, 8.5, 8.3, 8.5, 8.4,
       8. , 8.1, 8.7, 8.9, 8.3, 8.1, 8.1, 8. , 8.2, 8.4, 8.4, 8.1, 8.3,
       8.4, 8.2, 8.5, 8. , 8.2, 8.1, 8.4, 8.1, 8.6, 8.4, 8.1, 8.7, 8.1,
       8.2, 8.1, 8.3])
```

[ ]:

[393]:
```
imdb.sort("Rating", descending=True)
```

[393]:

Votes	Rating	Title	Year	Decade
1027398	9.2	The Godfather	1972	1970
1498733	9.2	The Shawshank Redemption	1994	1990
692753	9	The Godfather: Part II	1974	1970
447875	8.9	Il buono, il brutto, il cattivo (1966)	1966	1960
1473049	8.9	The Dark Knight	2008	2000
384187	8.9	12 Angry Men	1957	1950
1074146	8.9	The Lord of the Rings: The Return of the King	2003	2000
761224	8.9	Schindler's List	1993	1990
1166532	8.9	Pulp Fiction	1994	1990
1177098	8.8	Fight Club	1999	1990

… (240 rows omitted)

[400]:
```
def rating_year(Rating):
    return int(float(Rating))
```

[404]:
```
rating_year(9.2)
```

[404]:
```
9
```

[409]:
```
Total_Pay = imdb.apply(rating_year,"Rating")
Total_Pay
```

[409]:
```
array([8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       9, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 9, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8])
```

[417]:
```
compnew = imdb.with_column("Rating", imdb.apply(rating_year,"Rating"))
compnew
```

[417]:
```
Votes  | Rating | Title                  | Year | Decade
88355  | 8      | M                      | 1931 | 1930
132823 | 8      | Singin' in the Rain    | 1952 | 1950
74178  | 8      | All About Eve          | 1950 | 1950
635139 | 8      | Léon                   | 1994 | 1990
145514 | 8      | The Elephant Man       | 1980 | 1980
425461 | 8      | Full Metal Jacket      | 1987 | 1980
441174 | 8      | Gone Girl              | 2014 | 2010
850601 | 8      | Batman Begins          | 2005 | 2000
37664  | 8      | Judgment at Nuremberg  | 1961 | 1960
46987  | 8      | Relatos salvajes       | 2014 | 2010
… (240 rows omitted)
```

[419]:
```
type(compnew.column("Rating")[0])
```

[419]:
```
numpy.int64
```

[422]:
```
compnew.barh('Year', 'Rating')
```


Title
Rating
0
2
4
6
8

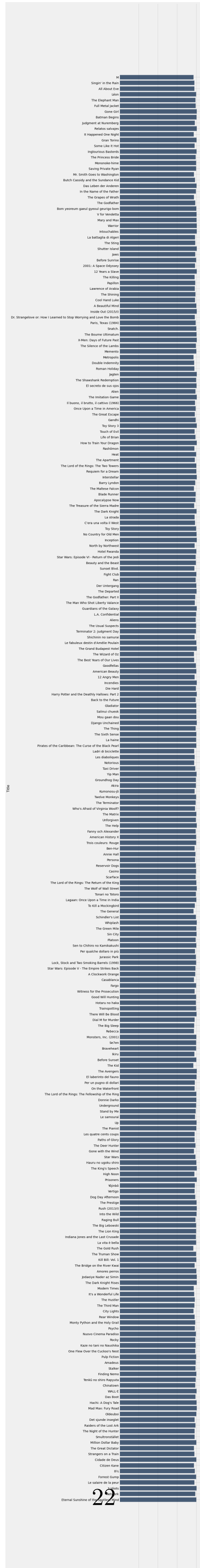
#F

The plot shows that most of the years recieved an 8 rating but 1974, 1994 and 1972 received a rating higher then 8

[431]:
```
#G
imdb.sort("Title", descending=True)
```

[431]:
```
Votes  | Rating | Title                           | Year | Decade
65370  | 8.3    | Yôjinbô                         | 1961 | 1960
138240 | 8      | Yip Man                         | 2008 | 2000
427099 | 8      | X-Men: Days of Future Past      | 2014 | 2010
53186  | 8.3    | Witness for the Prosecution     | 1957 | 1950
50208  | 8      | Who's Afraid of Virginia Woolf? | 1966 | 1960
264333 | 8.5    | Whiplash                        | 2014 | 2010
287727 | 8.2    | Warrior                         | 2011 | 2010
618914 | 8.4    | WALL·E                          | 2008 | 2000
218430 | 8.4    | Vertigo                         | 1958 | 1950
700999 | 8.2    | V for Vendetta                  | 2005 | 2000
… (240 rows omitted)
```

[430]:
```
imdb.barh('Title', 'Year')
```


Title
Year
0
500
1000
1500
2000

[ ]: