In [1]:

```
import requests
from tabulate import tabulate
import numpy_financial
import pandas as pd
import math

import scipy.stats

import matplotlib
import matplotlib.pyplot as pp
```

In [2]:

```
url = 'https://nylottery.ny.gov/drupal-api/api/v2/scratch_off_data?_format=json'
r = requests.get(url)
data  = r.json()
```

In [3]:

```
Dataset = []
Spending_amount = 20

for TicketType in data['rows']:
    Array_of_Tickets = []
    try:
        Title = TicketType['title']
        Title = Title.replace('  ','')
        game_number = TicketType['game_number']
        ticket_price = float(TicketType['ticket_price'])

        # Looping through the tiers of prizes
        All_tier_prices = TicketType['odds_prizes']
        Total_Profit = 0
        Sum_of_remaining_prizes = 0
        Sum_of_remaining_and_paid_out_prizes = 0
        for Different_tier_prices in All_tier_prices:
            Tier_prizes_remaining = int(Different_tier_prices['prizes_remaining'])
            Tier_prizes_paidout = int(Different_tier_prices['prizes_paid_out'])

            Tier_overall_odds = Different_tier_prices['overall_odds']
            Tier_overall_odds = Tier_overall_odds.replace('1 in ','')
            Tier_overall_odds = Tier_overall_odds.replace(',','')
            Tier_overall_odds = float(Tier_overall_odds)

            prize_amount = Different_tier_prices['prize_amount']
            prize_amount = prize_amount.replace(',','')
            prize_amount = prize_amount.replace('$','')
            prize_amount = prize_amount.replace(' ','')

            try:
                prize_amount = float(prize_amount)
            except Exception as err:
                try:
                    Parsed_prize_amount = prize_amount[:4]
                    Parsed_prize_amount = Parsed_prize_amount.replace('$','')
                    Parsed_prize_amount = Parsed_prize_amount.replace('/','')
                    Parsed_prize_amount = Parsed_prize_amount.replace('A','')
                    Parsed_prize_amount = Parsed_prize_amount.replace('W','')
                    if 'K' in Parsed_prize_amount:
                        Parsed_prize_amount = Parsed_prize_amount.replace('K','')
                        Parsed_prize_amount = int(Parsed_prize_amount)*1000
                    if 'WK' in Different_tier_prices['prize_amount']:
                        Parsed_prize_amount = int(Parsed_prize_amount)*52
                    elif "WEEK" in Different_tier_prices['prize_amount']:
                        Parsed_prize_amount = int(Parsed_prize_amount)*52
                    Cashflow = []
                    for num in range(50):
                        Cashflow.append(Parsed_prize_amount)
                    NPV = round(numpy_financial.npv(.10, Cashflow),2)
                    prize_amount = NPV

                except Exception as err:
                    if Tier_overall_odds >= 1000000:
                        Cashflow = []
                        for num in range(50):
                            Cashflow.append(ticket_price*15000)
                        NPV = round(numpy_financial.npv(.15, Cashflow),2)
                        prize_amount = NPV
                    else:
                        prize_amount = 0

            # Adding the Remaining Tickets into an array
            for num in range(Tier_prizes_remaining):
                rounded = int(round(prize_amount,0))
                Array_of_Tickets.append(rounded)

            Total_Profit = Total_Profit + (Tier_prizes_remaining*prize_amount)
            Sum_of_remaining_prizes = Sum_of_remaining_prizes + Tier_prizes_remaining
            Sum_of_remaining_and_paid_out_prizes = Sum_of_remaining_and_paid_out_prizes + Tier_prizes_remaining + Tier_prizes_paidout

        # overall odds of pulling a winning ticket
        overall_odds_num = TicketType['overall_odds']
        overall_odds_num = overall_odds_num.replace(':','')
        overall_odds_num = overall_odds_num.replace('\t','')
        overall_odds_num = overall_odds_num.replace(' ','')
        overall_odds_num = overall_odds_num.replace(',','')
        overall_odds_num = overall_odds_num.replace('1in','')
        overall_odds_num = overall_odds_num.replace('OddsofWinningCashPrize','')
        overall_odds_num = overall_odds_num.replace('CashOdds','')
        try:
            overall_odds_num = float(overall_odds_num)
        except:
            Rough_Estimate_of_Remaining_Tickets = (Tier_prizes_remaining + Tier_prizes_paidout)*Tier_overall_odds
            overall_odds_num = round(1/(Sum_of_remaining_and_paid_out_prizes/Rough_Estimate_of_Remaining_Tickets),2)
        overall_odds_str = "1 in {}".format(overall_odds_num)

        # Amount of tickets bought for excepted return
#         amount = round(overall_odds_num)

        amount = math.floor(Spending_amount/ticket_price)
        if amount == 0:
            continue
        Cost_of_Tickets = amount*ticket_price
        # Calculations
        Total_remaining_tickets = round(float(overall_odds_num)*int(Sum_of_remaining_prizes),0)
        Total_amount_of_ticket_ever_made = round(float(overall_odds_num)*int(Sum_of_remaining_and_paid_out_prizes),0)
        Total_Cost = Total_remaining_tickets * ticket_price
        Total_return = round(((Total_Profit-Total_Cost)/Total_Cost) * 100,2)
        perc_remaining = round((Total_remaining_tickets/Total_amount_of_ticket_ever_made)*100,2)

        # Adding Losing Tickets to the array
        Remaining_Losing_Tickets = int(round(Total_remaining_tickets - len(Array_of_Tickets),0))
        for num in range(Remaining_Losing_Tickets):
            Array_of_Tickets.append(0)

        df = pd.DataFrame(Array_of_Tickets, columns = ['Remaining_Tickets'])
        bootstrap = pd.DataFrame({'sumOfTickets': [df.sample(amount, replace=True).Remaining_Tickets.sum() for i in range(1000)]})
        # Chance of breaking even or better
        Chance_BEOB = round(100-scipy.stats.percentileofscore(bootstrap.sumOfTickets, Cost_of_Tickets),2)
        # Return %
        Return_per = round(((bootstrap.sumOfTickets.mean() - Cost_of_Tickets)/Cost_of_Tickets)*100,2)

        # Sorting Volume Label
        if Total_remaining_tickets>=1000000000:
            Total_remaining_tickets = round(Total_remaining_tickets / 1000000000,0)
            Total_remaining_tickets = "{} Bill".format(Total_remaining_tickets)
        elif Total_remaining_tickets>= 1000000:
            Total_remaining_tickets = round(Total_remaining_tickets / 1000000,0)
            Total_remaining_tickets = "{} Mill".format(Total_remaining_tickets)
        elif Total_remaining_tickets>=1000:
            Total_remaining_tickets = round(Total_remaining_tickets / 1000,0)
            Total_remaining_tickets = "{} Thous".format(Total_remaining_tickets)

        Dataset.append([Title, Chance_BEOB, Return_per, Total_return, overall_odds_str, ticket_price, Total_remaining_tickets, perc_remaining, game_number]
        remaininglist = len(data['rows']) - len(Dataset)
        string = "There is {} remaining tickets to be checked                                                             ".format(remaininglist)
        print(string, end='\r')
        #     except ZeroDivisionError:
    except Exception as err:
        x=1
        print(Title, game_number, err)

Dataset.sort(key = lambda i: i[1])
Dataset.reverse()
Header = ['Title', '>=BE %', 'Return %', 'Total Return %','Odds', 'Price', 'Remaining', '%', 'Ticket Num']

print(tabulate(Dataset, headers = Header))
```

```
Title                             >=BE %    Return %    Total Return %  Odds          Price  Remaining        %    Ticket Num
------------------------------  --------  ----------  ----------------  ---------  -------  -----------  -----  ------------
$5,000,000 MEGA MULTIPLIER          26        -32.85            -35.89  1 in 3.33       20  1.0 Mill      5.8           1439
ALL CASH                            25.8       -1.55            -34.68  1 in 4.64       20  8.0 Mill     74.64          1571
$300,000,000 CASH PAYOUT            23.9      -26.95            -27.93  1 in 3.55       20  11.0 Mill    52.62          1528
SET FOR LIFE                        23.7      -35.3             -32.93  1 in 3.96       10  31.0 Mill    62.89          1548
CASH X50                            23.35     -28.9             -29.13  1 in 3.91       10  6.0 Mill     28.65          1554
$5,000,000 CASH ROYALE              22.9        7.95            -27.46  1 in 3.9        20  16.0 Mill    89.1           1579
CASH X100                           22.85     -31.75            -26.29  1 in 3.68       20  15.0 Mill    64.88          1558
TRIPLE JACKPOT 777                  22.85     -35.25            -32.42  1 in 4.23       10  12.0 Mill    63.21          1567
$50 OR $100                         22.4      -28               -25.1   1 in 8.0        10  2.0 Mill     34.34          1575
X SERIES: 100X                      21.95     -37.8              -8.61  1 in 3.68       20  1.0 Mill      4.72          1509
$5,000,000 RICHES                   21.85     -46.45             35.93  1 in 3.35       20  319.0 Thous   1.81          1483
```

$10,000 A WEEK FOR LIFE	21.55	-45.23	-42.98	1 in 3.39	20	1.0 Mill	5.92	1478
FAST $500!	21.35	-25.45	-32.96	1 in 4.17	10	4.0 Mill	20.4	1524
HOLIDAY $1,000,000!	21.15	-37	-31.1	1 in 3.46	10	2.0 Mill	30.42	1543
HIT IT BIG!	21.1	-31.03	-34.58	1 in 4.55	5	6.0 Mill	55.14	1559
MULTIPLIER CRAZE	21	-33.8	-33.48	1 in 3.92	10	17.0 Mill	93.89	1582
MILLIONAIRE MAKER	20.75	-29.95	-27.89	1 in 3.91	20	14.0 Mill	63.54	1540
MULTIPLIER MONEY	20.7	-31.55	-29.02	1 in 4.26	5	6.0 Mill	50.49	1537
CASH X20	20.15	-32.58	-36.99	1 in 4.11	5	7.0 Mill	40.96	1555
DOUBLE TRIPLE CASHWORD	20.15	-43.25	-36.22	1 in 5.14	5	26.0 Mill	62.22	1549
$1,000,000 BONUS WORD CASHWORD	19.85	-38.3	-36.64	1 in 4.35	10	3.0 Mill	14.19	1551
POWER SPOT	19.75	-42.07	-32.93	1 in 4.27	5	3.0 Mill	25.81	1522
TRIPLE RED 777	19.7	-41.5	-43.79	1 in 4.24	10	652.0 Thous	3.52	1534
ELECTRIC 10X	19.2	-19.15	-34.93	1 in 3.94	10	5.0 Mill	27.99	1517
INSTANT $500	19	-34.83	-32.02	1 in 4.19	10	12.0 Mill	64.87	1561
CASH X20 BINGO	19	-41.85	-34.84	1 in 4.31	5	9.0 Mill	69.77	1572
LINE 'EM UP	19	-40.82	-37.9	1 in 4.33	3	7.0 Mill	64.44	1578
MAGIC 8 BALL	19	-34.65	-34.74	1 in 4.02	5	9.0 Mill	92.04	1583
EXTREME 8S	18.8	-44.92	-34.41	1 in 4.36	5	5.0 Mill	46.44	1533
MYSTERY MULTIPLIER CASHWORD	18.75	-39.54	-38.12	1 in 4.18	3	5.0 Mill	27.98	1530
CASH TO GO!	18.65	-36.92	-37.22	1 in 4.04	5	5.0 Mill	42.53	1538
SET FOR LIFE	18.55	-50.28	-46.46	1 in 3.97	10	1.0 Mill	2.89	1472
7-11-21 TRIPLER	18.55	-37.85	-40.12	1 in 4.54	1	4.0 Mill	42.62	1581
WILD CASH MULTIPLIER	18.45	-37.15	-34.47	1 in 4.13	5	6.0 Mill	67.38	1577
LOOSE CHANGE MULTIPLIER	17.75	-37.3	-34.91	1 in 4.01	5	4.0 Mill	29.13	1510
BIG BUCKS	17.35	-41.95	-38.02	1 in 4.29	5	2.0 Mill	19.67	1458
DOUBLE YOUR DOLLARS	17.15	-31.65	-36.23	1 in 4.14	5	5.0 Mill	55.73	1576
BINGO X20	17.05	-46.77	-41.63	1 in 4.31	5	1.0 Mill	8.36	1536
CASHWORD DOUBLER	17	-36.92	-40.56	1 in 5.09	2	26.0 Mill	32.12	1531
FIND THE 7S	16.85	-37.05	-37.58	1 in 4.88	2	9.0 Mill	92.38	1569
$1,000,000 GOLDEN FORTUNE	16.7	-46	-38.64	1 in 4.06	5	6.0 Mill	56.78	1547
$1,000,000 BONUS WORD CASHWORD	16.3	-48.85	-48.07	1 in 4.35	10	794.0 Thous	4.33	1502
RUBY RED 7S	16.3	-41.17	-37.16	1 in 4.15	5	7.0 Mill	57.95	1568
DIAMOND 25X	16.2	-37.47	-33.76	1 in 4.23	5	6.0 Mill	64.94	1580
STRIKE IT RICH	16.15	-36.75	-35.16	1 in 4.07	5	690.0 Thous	6.65	1513
FAST $250!	16.1	-40.2	-36.12	1 in 4.12	5	4.0 Mill	34.96	1525
LOTERIA	15.9	-37.98	-37.81	1 in 4.4	3	6.0 Mill	36.02	1541
TREASURE HUNT	15.8	-41.95	-44.23	1 in 4.12	5	1.0 Mill	12.16	1496
HOLIDAY FUN!	15.7	-43.75	-38.25	1 in 3.98	5	3.0 Mill	39.12	1544
LOOSE CHANGE	15.7	-37.46	-36.62	1 in 4.7	1	79.0 Mill	51.54	1552
WINNER$ GALORE	15.05	-41.24	-42.63	1 in 4.8	2	2.0 Mill	15.76	1514
LUCKY 7S	15.05	-40.35	-36.8	1 in 4.96	1	20.0 Mill	79.51	1570
CASH X5	14.95	-41.8	-39.69	1 in 4.82	1	11.0 Mill	41.86	1557
$25K CASH BONUS	14.45	-46.04	-43.3	1 in 4.8	2	2.0 Mill	18.24	1535
X SERIES: 20X	14.25	-46.15	-44.25	1 in 4.12	5	1.0 Mill	8.7	1506
X SERIES: 10X	14.2	-42.61	-45.69	1 in 4.8	2	1.0 Mill	7.72	1507
CASH X10	14.05	-48.77	-38.3	1 in 4.8	2	14.0 Mill	53.95	1556
$1,000,000 PREMIERE	13.95	-42	-36.98	1 in 3.92	5	1.0 Mill	14.07	1467
BREAK THE BANK	13.9	-47	-38.31	1 in 4.16	5	8.0 Mill	64.06	1562
FAST $100!	13.85	-41.48	-42.89	1 in 4.9	2	2.0 Mill	13.77	1526
HOLIDAY LUCK	13.85	-40.11	-41.26	1 in 4.51	2	3.0 Mill	30.71	1545
CASHWORD X15	13.85	-44.62	-41.15	1 in 4.2	3	3.0 Mill	20.97	1560
$500 BLOWOUT	13.8	-47.1	-47.97	1 in 4.19	10	357.0 Thous	1.93	1493
$60,000 DIAMONDS / RUBIES	13.4	-42.85	-45.01	1 in 4.14	3	2.0 Mill	10.16	1523
$2,500 A WEEK FOR LIFE	13.3	-52.87	-53.96	1 in 4.75	5	3.0 Mill	10.01	1447
LOOSE CHANGE TRIPLER	13.15	-45.44	-41.19	1 in 4.86	2	2.0 Mill	13.45	1542
TRIPLE WINNING 7'S	13.05	-45.8	-42.11	1 in 4.84	2	1.0 Mill	13.72	1563
ONE-WORD CASHWORD	12.8	-43.53	-41.27	1 in 4.23	3	771.0 Thous	3.93	1481
WIN FOR LIFE	12.65	-49.52	-45.09	1 in 3.75	2	25.0 Mill	54.28	1532
DOUBLE TRIPLE CASHWORD	12.6	-55.27	-42.34	1 in 5.14	5	720.0 Thous	1.79	1492
DOUBLE BONUS 7S	12.4	-52.9	-37.81	1 in 4.4	5	2.0 Mill	22.16	1529
777 MULTIPLIER	12.15	-46.88	-43.82	1 in 4.84	2	678.0 Thous	6.57	1519

```
X SERIES: 15X CASHWORD             12.05      -51.16            -47.52  1 in 4.12        3  1.0 Mill      7.65          1511
$100 OR $200                       12         -30               -29.35  1 in 8.0        20  120.0 Thous   1.94          1521
$1,000,000 LUCKY DOG               11.85      -49               -39.05  1 in 3.9         5  2.0 Mill     18.36          1518
CASHWORD DOUBLER                   11.75      -49.09            -45.47  1 in 5.09        2  2.0 Mill      2.8           1477
CASHWORD                           11.2         2.03            -47.83  1 in 5.02        2  6.0 Mill     15.18          1515
CASHWORD                           10.95      -47               -46.09  1 in 5.02        2  2.0 Mill      4.01          1475
MATCH 2 WIN                        10.7       -45.32            -41.7   1 in 5.22        1  4.0 Mill     30.07          1566
IT TAKES 2                         10.5       -28.24            -38.62  1 in 4.87        2  6.0 Mill     60.68          1573
$5,000 CASH!                       10.2       -45.93            -48.68  1 in 5.34        1  1.0 Mill     10.81          1539
FAST $50!                          10.1       -45.67            -46.28  1 in 4.82        1  2.0 Mill     11.1           1527
BEAT THE HOUSE                      9.85      -51.78            -50.94  1 in 5.01        1  2.0 Mill     11.65          1446
LUCKY 7S                            9.75      -51.76            -49.98  1 in 4.96        1  3.0 Mill     10.88          1520
LUCKY 7'S                           9.65      -51.51            -49.87  1 in 4.67        1  4.0 Mill      8.38          1454
DOUBLE DOUBLER                      9.45      -28.37            -46.09  1 in 4.87        1  1.0 Mill      8.7           1498
HGTV'S MY LOTTERY DREAM HOME        9.15      -54.12            -42.62  1 in 4.09        5  1.0 Mill     19.66          1564
HOLIDAY 7S / LUCKY 7S               8.6       -53.52            -49.72  1 in 4.96        1  3.0 Mill     13.14          1550
LOOSE CHANGE                        7.6       -53.75            -55.22  1 in 4.7         1  5.0 Mill      3.51          1476
WIN FOR LIFE                        3.95      -75.03            -68.27  1 in 3.75        2  5.0 Mill     11.97          1453
INSTANT TAKE 5                      3.35      -78.6             -76.67  1 in 4.69        1  115.0 Mill   83.8           1574
INSTANT TAKE 5                      2.5       -81.29            -81.32  1 in 4.69        1  7.0 Mill      5.2           1516
INSTANT TAKE 5                      1.45      -82.71            -81.05  1 in 4.69        1  7.0 Mill      4.82          1455
```

In [4]:

```
Ticket_num = '1439'
amount = 1

Array_of_Tickets = []
for TicketType in data['rows']:
    if TicketType['game_number'] == Ticket_num:
        Title = TicketType['title']
        game_number = TicketType['game_number']
        ticket_price = float(TicketType['ticket_price'])
        Cost_of_Tickets = amount*ticket_price

        # Looping through the tiers of prizes
        All_tier_prices = TicketType['odds_prizes']
        Total_Profit = 0
        Sum_of_remaining_prizes = 0
        Sum_of_remaining_and_paid_out_prizes = 0
        for Different_tier_prices in All_tier_prices:
            Tier_prizes_remaining = int(Different_tier_prices['prizes_remaining'])
            Tier_prizes_paidout = int(Different_tier_prices['prizes_paid_out'])

            Tier_overall_odds = Different_tier_prices['overall_odds']
            Tier_overall_odds = Tier_overall_odds.replace('1 in ','')
            Tier_overall_odds = Tier_overall_odds.replace(',','')
            Tier_overall_odds = float(Tier_overall_odds)

            prize_amount = Different_tier_prices['prize_amount']
            prize_amount = prize_amount.replace(',','')
            prize_amount = prize_amount.replace('$','')
            prize_amount = prize_amount.replace(' ','')

            try:
                prize_amount = float(prize_amount)
            except Exception as err:
                try:
                    Parsed_prize_amount = prize_amount[:4]
                    Parsed_prize_amount = Parsed_prize_amount.replace('$','')
                    Parsed_prize_amount = Parsed_prize_amount.replace('/','')
                    Parsed_prize_amount = Parsed_prize_amount.replace('A','')
                    Parsed_prize_amount = Parsed_prize_amount.replace('W','')
                    if 'K' in Parsed_prize_amount:
                        Parsed_prize_amount = Parsed_prize_amount.replace('K','')
                        Parsed_prize_amount = int(Parsed_prize_amount)*1000
                    if 'WK' in Different_tier_prices['prize_amount']:
                        Parsed_prize_amount = int(Parsed_prize_amount)*52
                    elif "WEEK" in Different_tier_prices['prize_amount']:
                        Parsed_prize_amount = int(Parsed_prize_amount)*52
                    Cashflow = []
                    for num in range(50):
                        Cashflow.append(Parsed_prize_amount)
                    NPV = round(numpy_financial.npv(.10, Cashflow),2)
                    prize_amount = NPV

                except Exception as err:
                    if Tier_overall_odds >= 1000000:
                        Cashflow = []
                        for num in range(50):
                            Cashflow.append(ticket_price*15000)
                        NPV = round(numpy_financial.npv(.15, Cashflow),2)
                        prize_amount = NPV
                    else:
                        prize_amount = 0
            # Adding the Remaining Tickets into an array
            for num in range(Tier_prizes_remaining):
                rounded = int(round(prize_amount,0))
                Array_of_Tickets.append(rounded)

            Total_Profit = Total_Profit + (Tier_prizes_remaining*prize_amount)
            Sum_of_remaining_prizes = Sum_of_remaining_prizes + Tier_prizes_remaining
            Sum_of_remaining_and_paid_out_prizes = Sum_of_remaining_and_paid_out_prizes + Tier_prizes_remaining + Tier_prizes_paidout

        # overall odds of pulling a winning ticket
        overall_odds_num = TicketType['overall_odds']
        overall_odds_num = overall_odds_num.replace(':','')
        overall_odds_num = overall_odds_num.replace('\t','')
        overall_odds_num = overall_odds_num.replace(' ','')
        overall_odds_num = overall_odds_num.replace(',','')
        overall_odds_num = overall_odds_num.replace('1in','')
        overall_odds_num = overall_odds_num.replace('OddsofWinningCashPrize','')
        overall_odds_num = overall_odds_num.replace('CashOdds','')
        try:
            overall_odds_num = float(overall_odds_num)
        except:
            Rough_Estimate_of_Remaining_Tickets = (Tier_prizes_remaining + Tier_prizes_paidout)*Tier_overall_odds
            overall_odds_num = round(1/(Sum_of_remaining_and_paid_out_prizes/Rough_Estimate_of_Remaining_Tickets),2)
        overall_odds_str = "1 in {}".format(overall_odds_num)

        # Calculations
        Total_remaining_tickets = round(float(overall_odds_num)*int(Sum_of_remaining_prizes),0)
        Remaining_Losing_Tickets = int(round(Total_remaining_tickets - len(Array_of_Tickets),0))
        for num in range(Remaining_Losing_Tickets):
            Array_of_Tickets.append(0)
```

```
df = pd.DataFrame(Array_of_Tickets, columns = ['Remaining_Tickets'])
df.head()

bootstrap = pd.DataFrame({'sumOfTickets': [df.sample(amount, replace=True).Remaining_Tickets.sum() for i in range(1000)]})
round(100-scipy.stats.percentileofscore(bootstrap.sumOfTickets, Cost_of_Tickets),2)
```

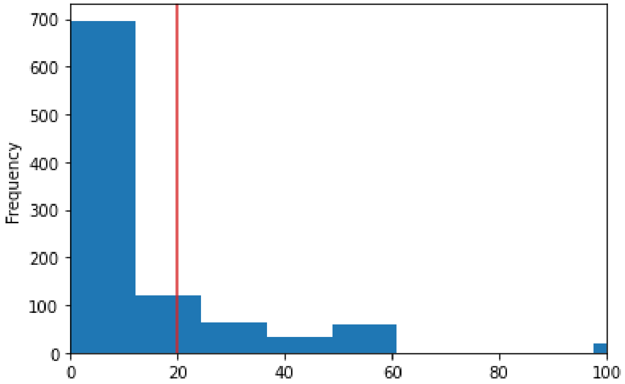
```
Out[4]: 24.3
```

```
In [5]: round(((bootstrap.sumOfTickets.mean() - Cost_of_Tickets)/Cost_of_Tickets)*100,2)
```

```
Out[5]: -34.9
```

```
In [6]: bins_count  = int(bootstrap.sumOfTickets.count()/24)
        bootstrap.sumOfTickets.plot.hist(bins = bins_count, xlim=(0,Cost_of_Tickets*5))
        pp.axvline(Cost_of_Tickets, c='C3')
        bootstrap.sumOfTickets.mean(), Cost_of_Tickets
```

```
Out[6]: (13.02, 20.0)
```



```
In [7]: bootstrap.sumOfTickets.max()
```

```
Out[7]: 500
```

```
In [8]: round(scipy.stats.percentileofscore(bootstrap.sumOfTickets, 300),2)
```

```
Out[8]: 99.6
```

```
In [ ]:
```