DDBJ

June 7, 2018

1 Black Jack (Double decks black jack)

1.0.1 Would make sure if paople can make money at Casino. If yes, want to know how much money I can make in a year.

What I did is that 1, Made black jack sumilator which is exact same rules as Parq Casino where is in down town in Vancouver by my own using python to collect datas to make dataframe. 2, Made sure if counting cards really works to win or not. 3, Figured out how much money should be buyin and goal in a day. 4, Sumilated 30times(days) and one year. In this case one year is 360 days.

```
In [14]: import pandas as pd
         import csv
         import matplotlib.pyplot as plt
         %pylab inline
         pylab.rcParams['figure.figsize'] = (15.0, 12.0)
         import mpl_toolkits.mplot3d as m3d
         from scipy import stats
```

```
Populating the interactive namespace from numpy and matplotlib
```

1.1 First of all, need to explain what is counting cards.

To be honest, Casinos dont allow players to count cards. So you must act that you dont counting cards.

1.1.1 What is card counting?

- Count 2 < cards <= 6 as +1
- Count 7<= cards <= 9 as 0
- Count 10 <= cards <= A as -1

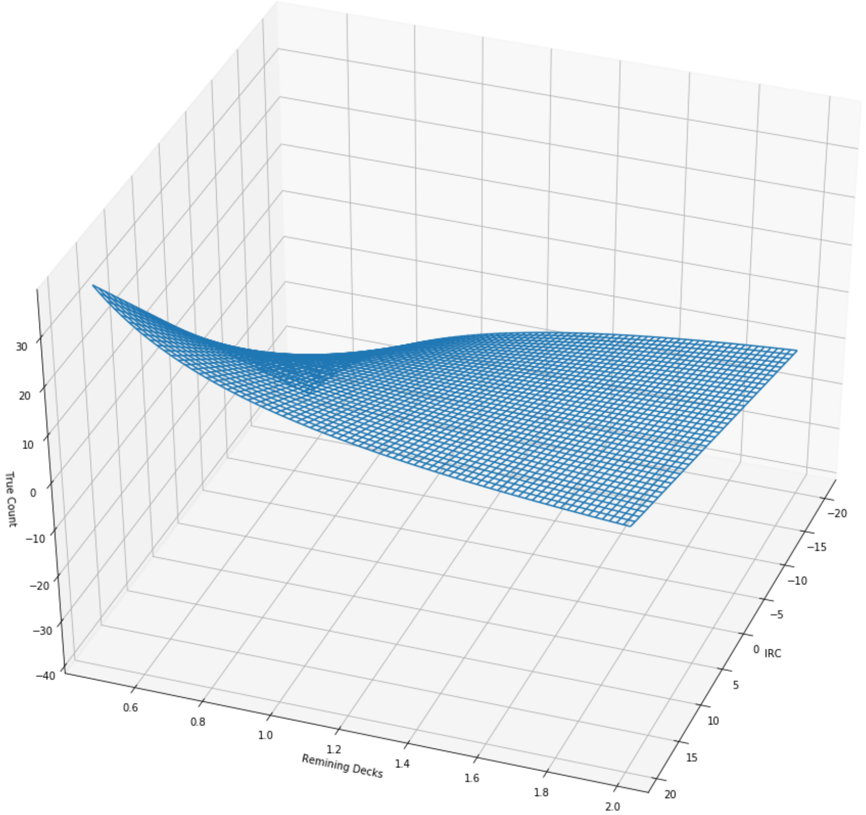
For instance, Player has 10 and 9, dealer has 10 and 10. The counting is going to be -3 which means player has more disadvantage to win than the counting is 0 and we call it IRC stands for Initial Running Count .

1.1.2 What is true count?

Made 3d plot about true count. True Count = IRC/Remaining Decks

For examples, When the IRC is 3 and remaining decks is 3/2(78cards), True Count is 2.

```
In [16]: fig3d = m3d.Axes3D(plt.figure())
         x = np.arange(-20,20,1)
         y = np.arange(26/52,2,1/52)
         xx,yy = np.meshgrid(x,y)
         Z = xx/yy
         fig3d.plot_wireframe(xx,yy,Z,rstride=1,cstride=1);
         fig3d.view_init(40,20)
         fig3d.set_xlabel("IRC");
         fig3d.set_ylabel("Remining Decks");
         fig3d.set_zlabel("True Count");
```



1.1.3 First of all, made 5 data frames which have 5000 hands datas as following rule

- Betting System: True count >= 5.5 bet 10 units, True count >= 1.5 bet math.floor(True count * 2 - 1), Others bet 1 unit (1 unit 25bucks)

```
In [17]: df1 = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Project/D
         df1.head()
```

```
Out[17]:   Player First Hand Player Action  Player Hand Player Sum  Dealer Show Card  \
         0            [8, 7]             H    [8, 7, 8]         23                 7
         1            [8, 9]             S       [8, 9]         17                 5
         2          [10, 10]             S     [10, 10]         20               111
         3          [10, 10]             S     [10, 10]         20                10
         4           [10, 5]             H  [10, 5, 10]         25               111

              Dealer Hand  Dealer Sum Result  Total Winning  Running Count  True Count
         0           [7.]           7      L          -25.0              0         0.0
         1  [ 5. 10.  2.]          17      P          -25.0              2         1.0
         2    [111.  10.]          21      L          -50.0              3         1.6
         3      [10. 10.]          20      P          -50.0             -1        -0.6
         4         [111.]          11      L          -75.0             -5        -3.0
```

```
In [18]: df2 = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Project/D
         df2.head()
```

```
Out[18]:   Player First Hand Player Action  Player Hand Player Sum  Dealer Show Card  \
         0         [111, 10]           PBJ    [111, 10]         21                10
         1            [3, 5]             H   [3, 5, 10]         18                 7
         2          [111, 4]             D  [111, 4, 2]         17                 5
         3           [3, 10]             S      [3, 10]         13                 2
         4            [4, 5]             D    [4, 5, 8]         17                 3

                  Dealer Hand  Dealer Sum Result  Total Winning  Running Count  \
         0          [10. 10.]          20      W           37.5              0
         1      [7. 2. 5. 3.]          17      W           62.5             -4
         2      [ 5. 10.  7.]          22      W          112.5              1
         3  [ 2.  9.  4. 10.]          25      W          162.5              3
         4   [  3. 111.   6.]          20      L          -37.5              4

            True Count
         0         0.0
         1        -2.1
         2         0.6
         3         1.8
         4         2.6
```

```
In [19]: df3 = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Project/D
         df3.head()
```

```
Out[19]:   Player First Hand Player Action Player Hand Player Sum  Dealer Show Card  \
         0            [7, 9]             S       [7, 9]         16                 3
         1            [4, 6]             H  [4, 6, 10]         20                10
         2         [10, 111]           PBJ   [10, 111]         21                 8
         3           [9, 10]             S     [9, 10]         19                10
         4          [10, 10]             S    [10, 10]         20                  4

              Dealer Hand  Dealer Sum Result  Total Winning  Running Count  True Count
         0     [3. 5. 9.]          17      L          -25.0              0         0.0
         1    [ 10. 111.]          21      L          -50.0              3         1.6
         2      [ 8. 10.]          18      W          -12.5              2         1.1
         3  [10.  5. 10.]          25      W           12.5             -1        -0.6
         4     [4. 7. 7.]          18      W           37.5             -3        -1.8
```

```
In [20]: df4 = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Project/D
         df4.head()
```

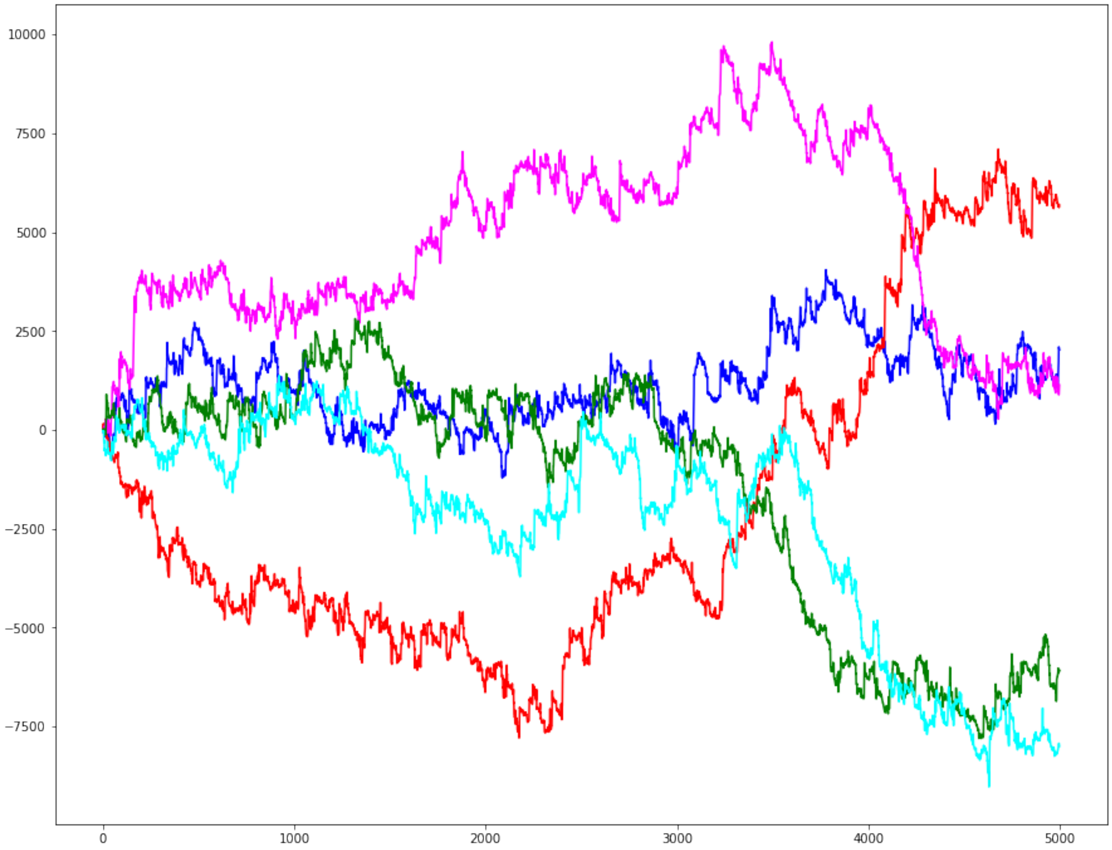
```
Out[20]:   Player First Hand Player Action  Player Hand Player Sum  Dealer Show Card  \
         0            [8, 9]             S       [8, 9]         17                10
         1          [4, 111]             H  [4, 111, 2]         17                 3
         2           [10, 9]             S      [10, 9]         19                 2
         3            [7, 3]             H    [7, 3, 8]         18                10
         4            [8, 4]             H    [8, 4, 6]         18                10

                           Dealer Hand  Dealer Sum Result  Total Winning  \
         0               [10.  5.  6.]          21      L          -25.0
         1               [ 3. 10. 10.]          23      W            0.0
         2  [  2. 111. 111.   2. 111.]          17      W           25.0
         3                   [10. 10.]          20      L            0.0
         4               [10.  6.  4.]          20      L          -25.0

            Running Count  True Count
         0              0         0.0
         1              1         0.5
         2              1         0.6
         3             -1        -0.6
         4             -1        -0.6
```

```
In [21]: df5 = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Project/D
         df5.head()
```

```
Out[21]:   Player First Hand Player Action     Player Hand Player Sum  \
         0           [3, 10]             H  [3, 10, 2, 10]         25
         1           [2, 10]             H  [2, 10, 2, 10]         24
         2           [4, 10]             H      [4, 10, 6]         20
         3            [2, 6]             H    [2, 6, 3, 8]         19
         4            [9, 8]             S          [9, 8]         17

            Dealer Show Card        Dealer Hand  Dealer Sum Result  Total Winning  \
         0                 7               [7.]           7      L          -25.0
         1                10              [10.]          10      L          -50.0
         2                 7        [  7. 111.]          18      W          -25.0
         3                 7         [7. 4. 9.]          20      L          -50.0
         4                 5  [ 5.  6.  5. 10.]          26      W          -25.0
```

```
   Running Count  True Count
0              0         0.0
1              1         0.5
2              0         0.0
3              1         0.6
4              6         3.8
```

```
In [22]: plt.plot(df1.index,df1["Total Winning"], color = "blue");
         plt.plot(df2.index,df2["Total Winning"], color = "red");
         plt.plot(df3.index,df3["Total Winning"], color = "magenta");
         plt.plot(df4.index,df4["Total Winning"], color = "green");
         plt.plot(df5.index,df5["Total Winning"], color = "cyan");
```



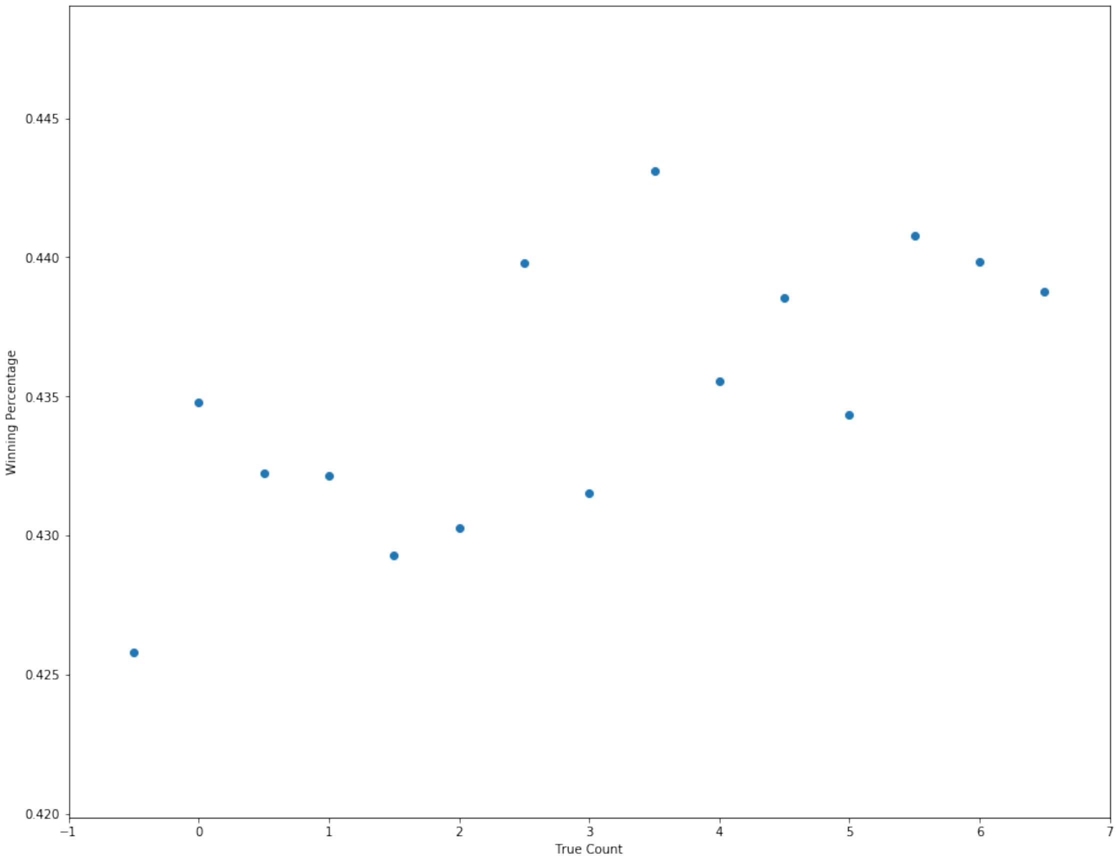
Right now, not sure that it’s possible to make money or not.

From now, investigated that winning percentage related to true count. So what I did is that run 20000 times for each true count which is from -0.5 to 6.5 step by 0.5.

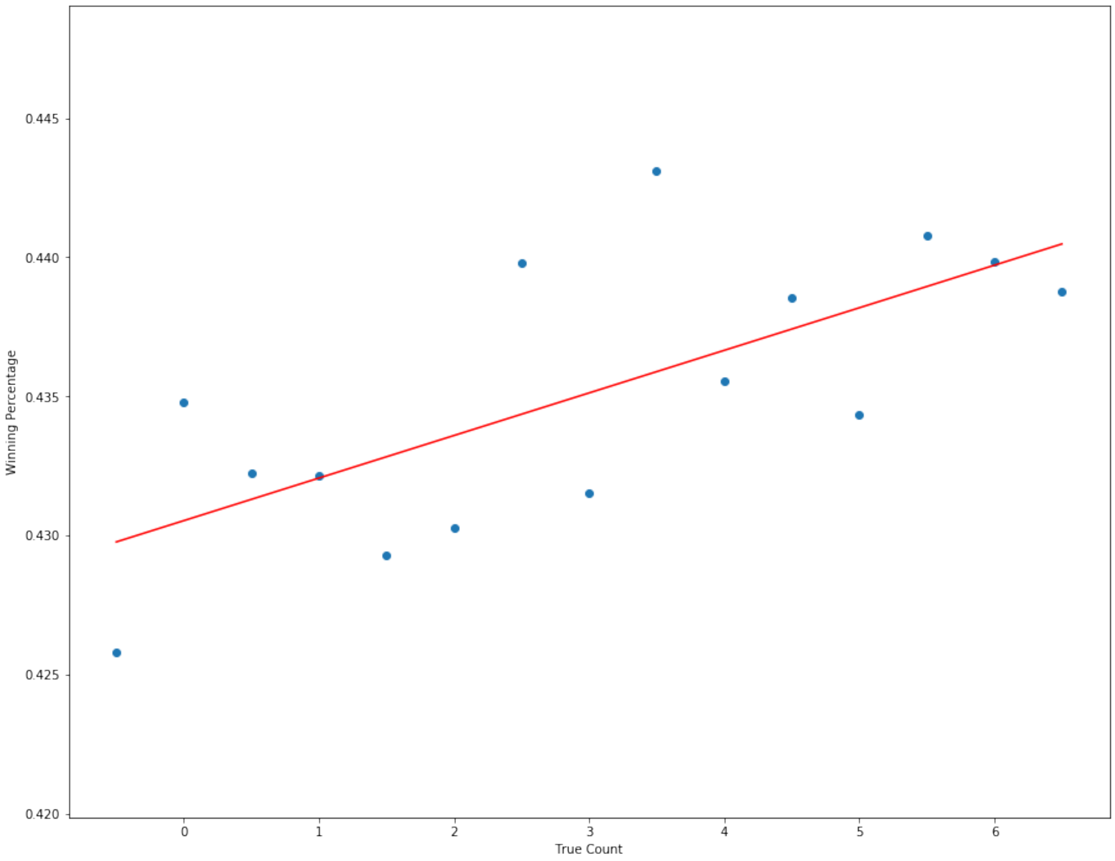
```
In [23]: dftcs = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Project/
         dftcs.head()
```

```
Out[23]:   Player First Hand Player Action    Player Hand Player Sum  Dealer Show Card  \
        0          [10, 5]             H     [10, 5, 2]         17                 7
        1           [7, 5]             S         [7, 5]         12                 4
        2          [9, 10]             S        [9, 10]         19                 8
        3          [10, 3]             S        [10, 3]         13                 5
        4         [111, 6]             H  [111, 6, 111]         18                 2

                     Dealer Hand  Dealer Sum Result  Total Winning  Running Count  \
        0              [ 7. 10.]          17      P            0.0             -1
        1  [  4. 111.  10.   2.]          17      L          -25.0             -1
        2             [8. 7. 6.]          21      L          -50.0             -1
        3          [ 5. 10. 10.]          25      W          -25.0             -1
        4      [ 2. 10.  3.  4.]          19      L          -50.0             -1

           True Count
        0        -0.5
        1        -0.5
        2        -0.5
        3        -0.5
        4        -0.5
```

```
In [25]: true_count = np.arange(-0.5,7,0.5)
         win_rate = list(map(lambda i: len(dftcs[(dftcs["True Count"] == i) & (dftcs["Result"]
         scatter(true_count,win_rate);
         plt.xlim(-1,7);
         plt.ylabel("Winning Percentage");
         plt.xlabel("True Count");
```

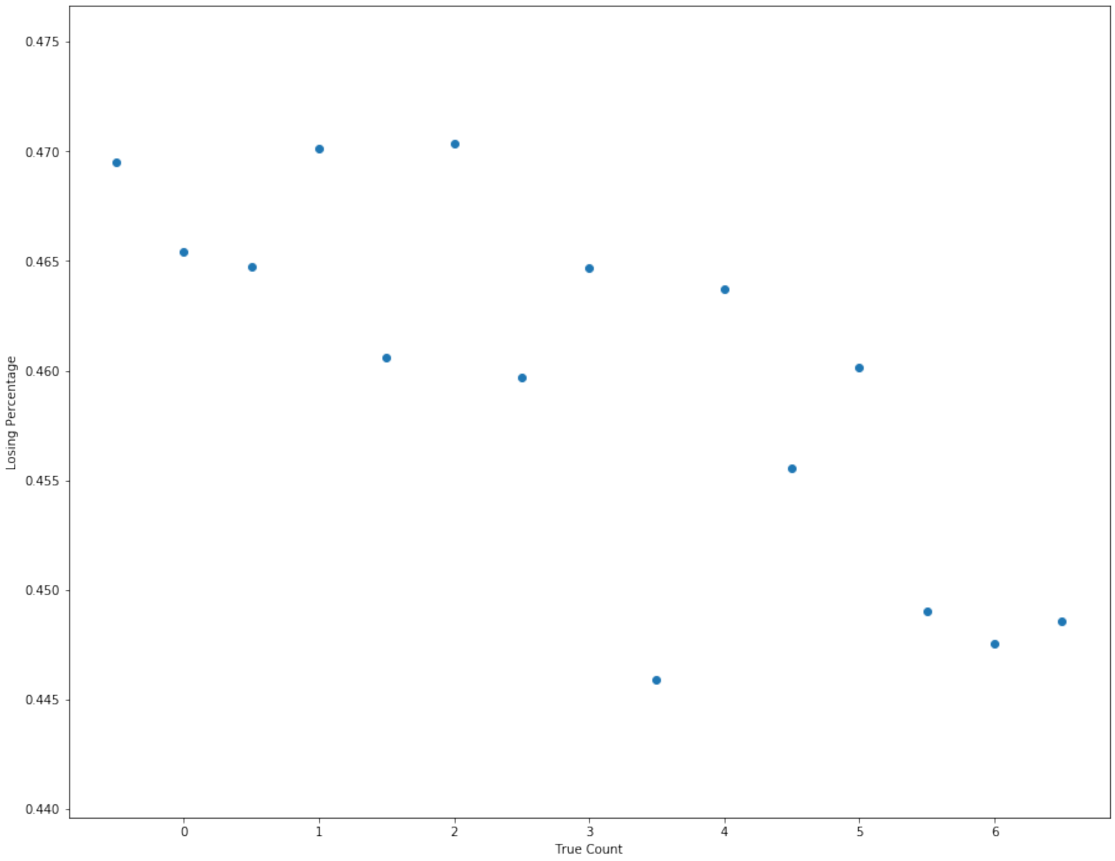


```
In [27]: np.corrcoef(true_count,win_rate)

Out[27]: array([[1.        , 0.69393464],
                [0.69393464, 1.        ]])

In [28]: slope_w, intercept_w, rvalue_w, pvalue_w, stderr_w = stats.linregress(true_count,win_

In [29]: scatter(true_count,win_rate);
         plt.plot(true_count,intercept_w + slope_w*true_count, color = "red");
         plt.ylabel("Winning Percentage");
         plt.xlabel("True Count");
```

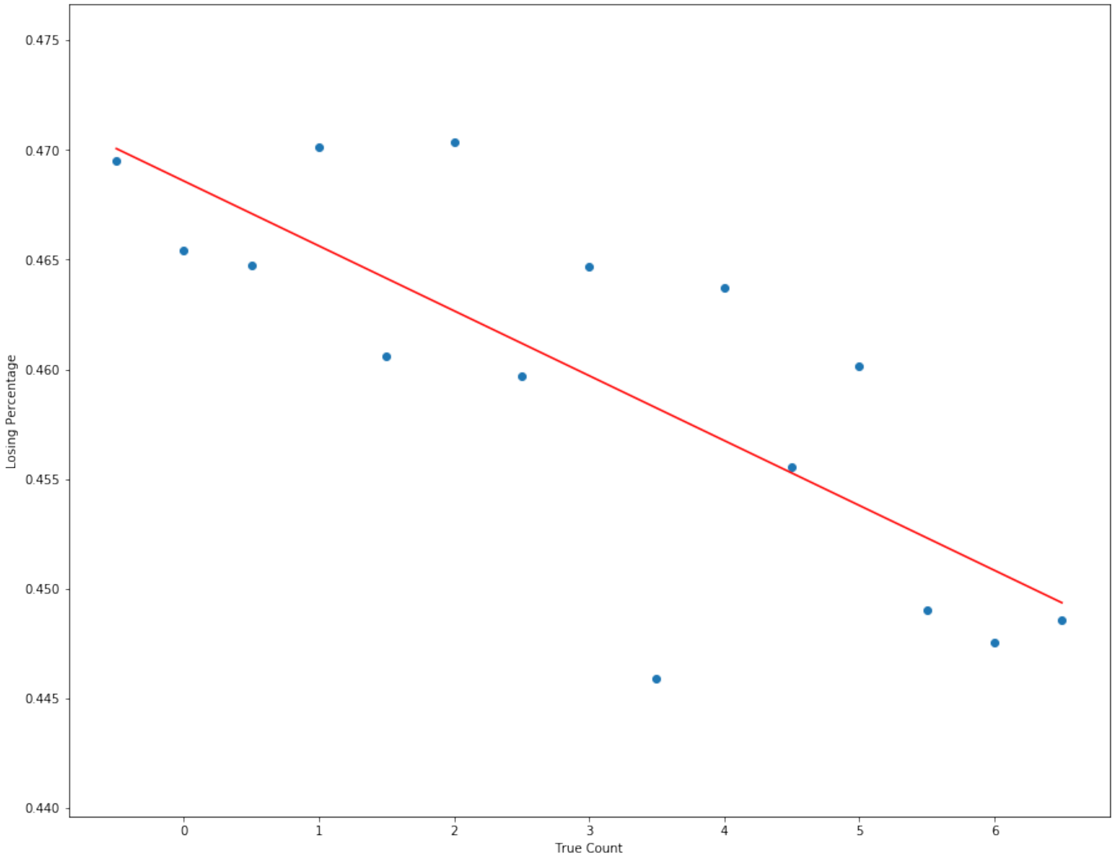


Now we can say there is relashionship between winning percentage and card counting. Let's see the other relationships.

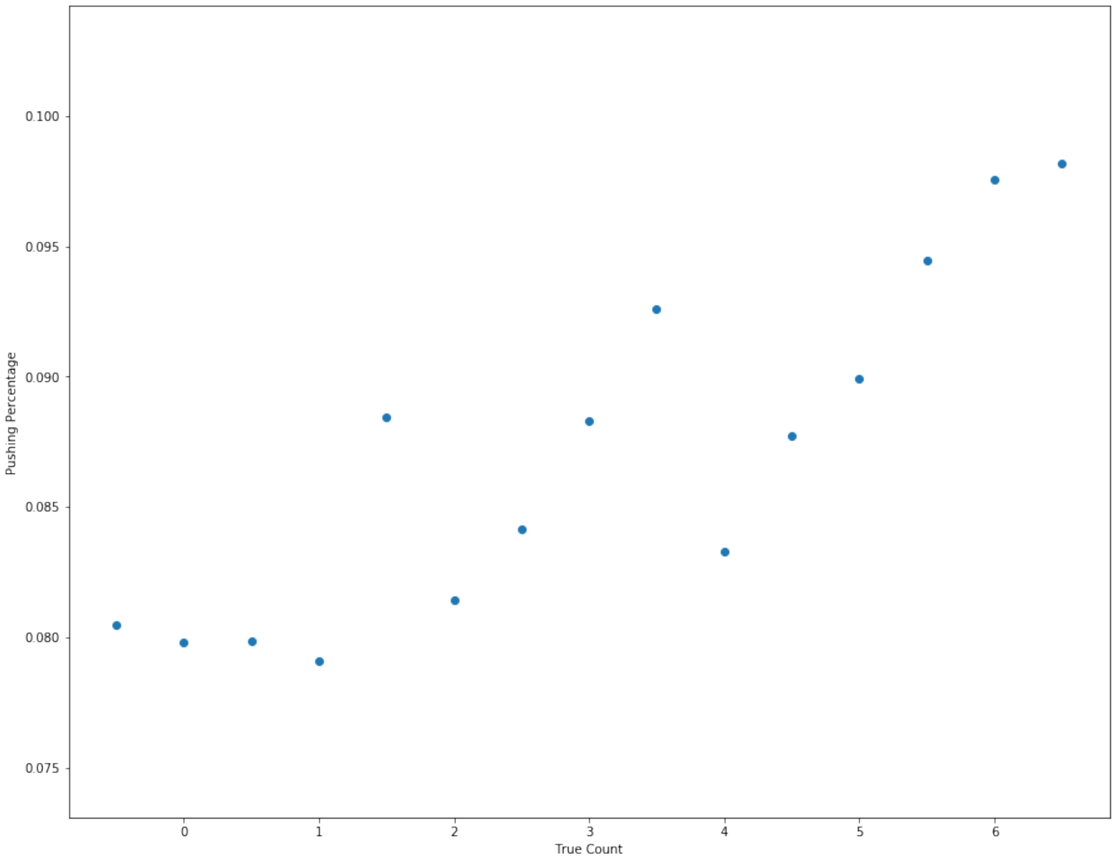
```
In [31]: true_count = np.arange(-0.5,7,0.5)
         lose_rate = list(map(lambda i: len(dftcs[(dftcs["True Count"] == i) & (dftcs["Result"]
         scatter(true_count,lose_rate);
         plt.ylabel("Losing Percentage");
         plt.xlabel("True Count");
```



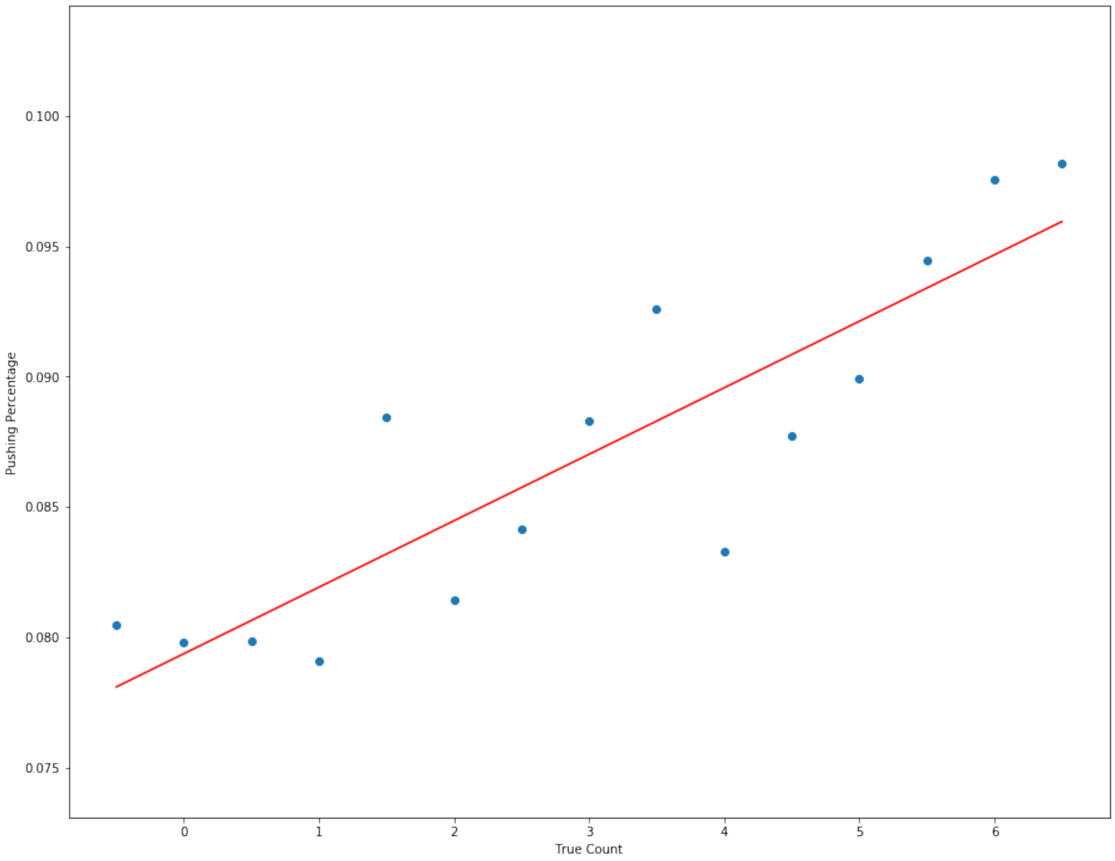
```
In [32]: np.corrcoef(true_count,lose_rate)

Out[32]: array([[ 1.        , -0.77821736],
                [-0.77821736,  1.        ]])

In [33]: slope_l, intercept_l, rvalue_l, pvalue_l, stderr_l = stats.linregress(true_count,lose_

In [34]: scatter(true_count,lose_rate);
         plt.plot(true_count,intercept_l + slope_l*true_count, color = "red");
         plt.ylabel("Losing Percentage");
         plt.xlabel("True Count");
```



```
In [35]: true_count = np.arange(-0.5,7,0.5)
         push_rate = list(map(lambda i: len(dftcs[(dftcs["True Count"] == i) & (dftcs["Result"]
         scatter(true_count,push_rate);
         plt.ylabel("Pushing Percentage");
         plt.xlabel("True Count");
```

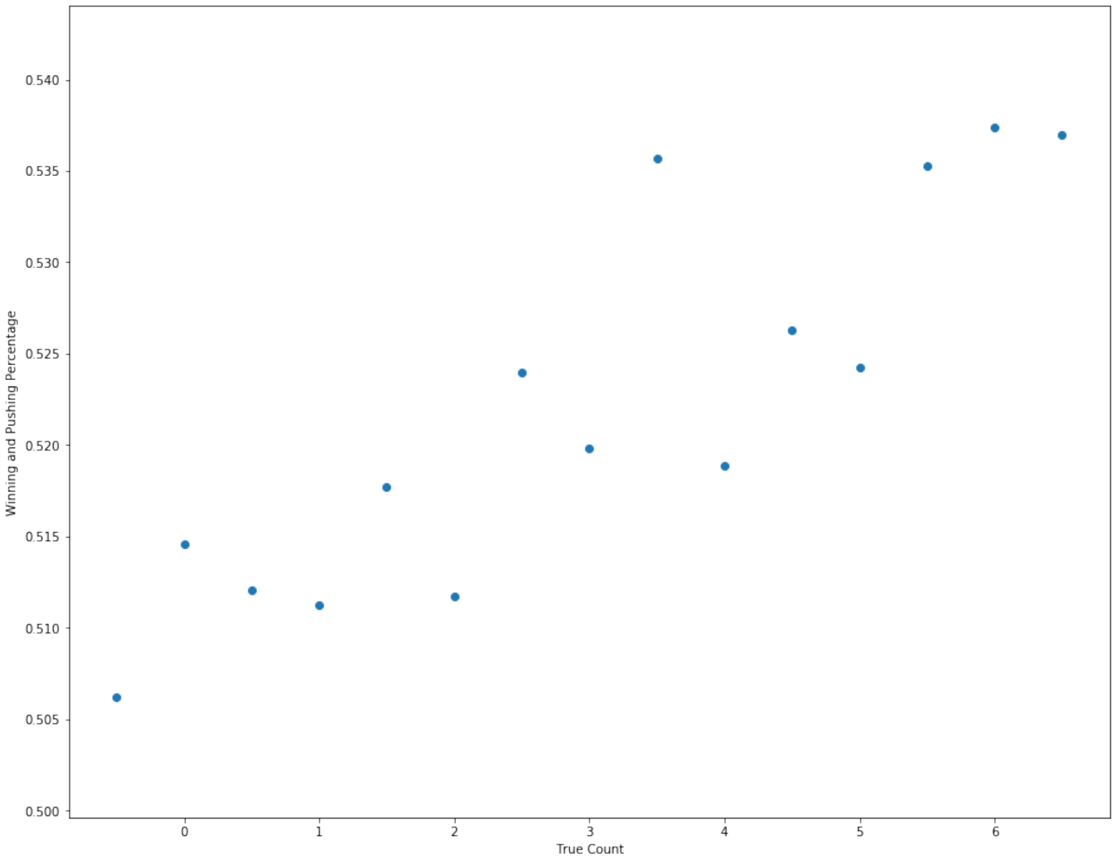


```
In [36]: np.corrcoef(true_count,push_rate)

Out[36]: array([[1.       , 0.8729908],
                [0.8729908, 1.       ]])

In [37]: slope_p, intercept_p, rvalue_p, pvalue_p, stderr_p = stats.linregress(true_count,push_

In [38]: scatter(true_count,push_rate);
         plt.plot(true_count,intercept_p + slope_p*true_count, color = "red");
         plt.ylabel("Pushing Percentage");
         plt.xlabel("True Count");
```



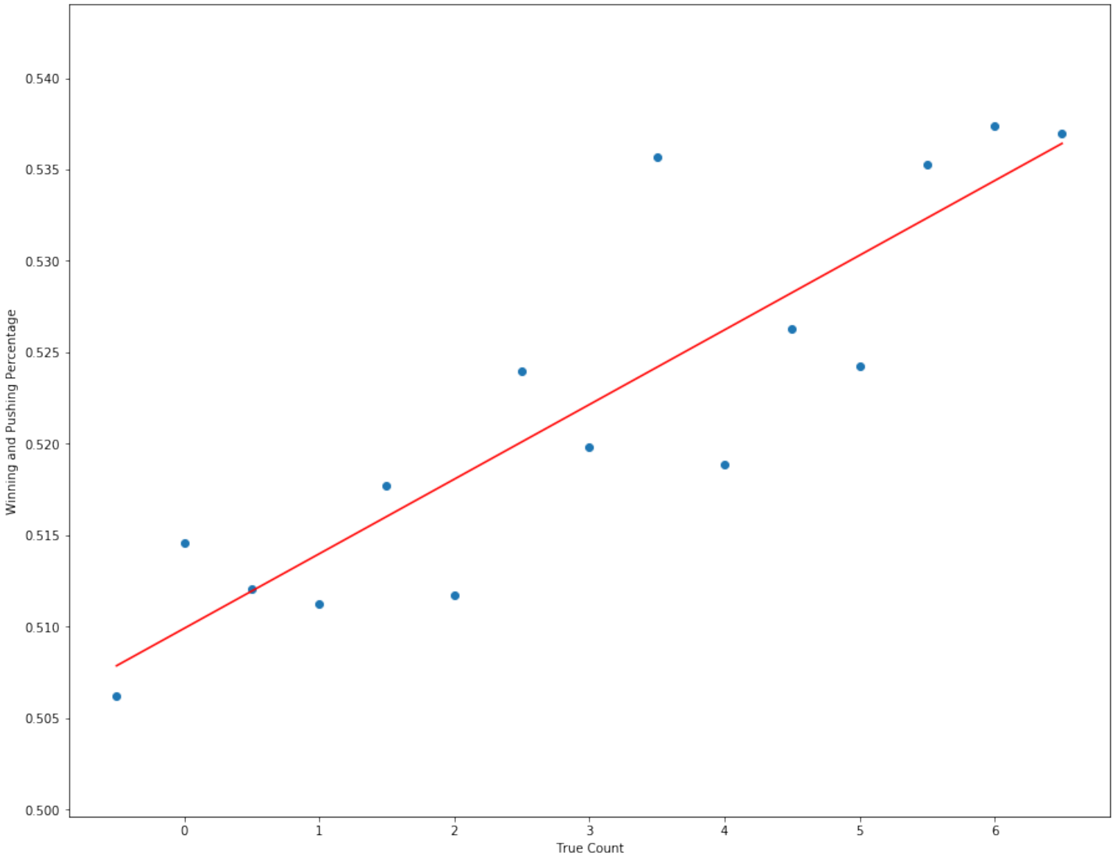
More true counting increase,more losing percentage decrease.
And we can say same thing as winning percentage to the pushing percentage.

```
In [39]: true_count = np.arange(-0.5,7,0.5)
         win_and_push_rate = list(map(lambda i: len(dftcs[(dftcs["True Count"] == i) & ((dftcs
         scatter(true_count,win_and_push_rate);
         plt.ylabel("Winning and Pushing Percentage");
         plt.xlabel("True Count");
```

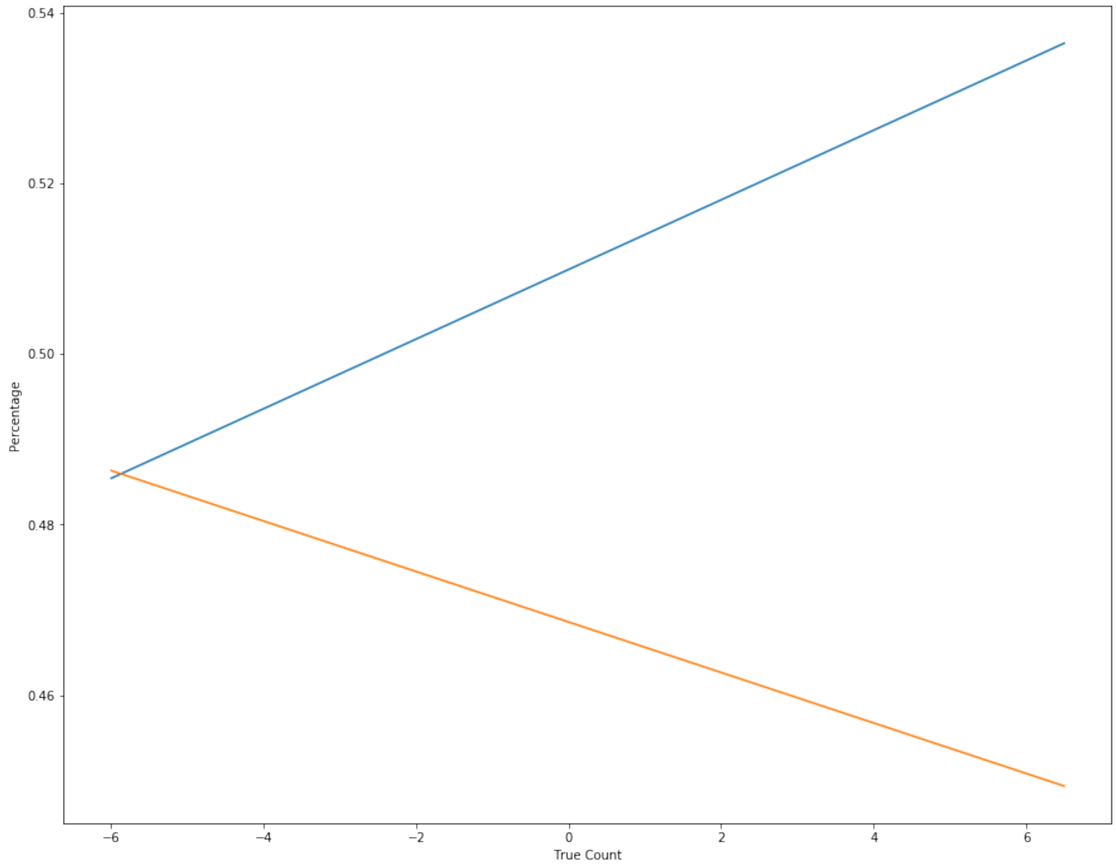


Then check the relationship between winning and pushing percentage and losing percentage

```
In [40]: slope_wp, intercept_wp, rvalue_wp, pvalue_wp, stderr_wp = stats.linregress(true_count

In [41]: scatter(true_count,win_and_push_rate);
         plt.plot(true_count,intercept_wp + slope_wp*true_count, color = "red");
         plt.ylabel("Winning and Pushing Percentage");
         plt.xlabel("True Count");
```



```
In [42]: count = np.arange(-6,7,0.5)
         plt.plot(count,intercept_wp + slope_wp*count);
         plt.plot(count,intercept_l + slope_l*count);
         plt.ylabel("Percentage");
         plt.xlabel("True Count");
```



When true count is greater than -6, winning + pushing percentage is bigger than losing percentage. It means P(lose money) < P(not to lose money), therefore we can bet more than usual. But perhaps there might be some error in values so change bet size when the true count >= 1.5

1.1.4 Betting Systems

There are some famous betting systems such as #### a, 1-2-1.... After you win,next bet is going to be 2 units. Whenever you lose, go back to 1 unit.

```
For instance,
lose  win  lose  win  win  win  lose.....
1uni1unit2units1unit2units1unit2units.....
```

b, 1-3-2-5

```
When you are winnig, chenge bet size.

lose  win  win  lose  win  win  win  win  lose....
1u    1u    3u    2u    1u    3u   2u    5u    1u
```

You can search other betting systems, if you are interested in.

1.1.5 Sumilated with following rules

- When true count is greater than equal to 1.5, 1-2-1... betting system.
- Buyin 200 dollers. If player lose 200 dollers in a day, stop the game at the day.
- Goal 100 dollers. If player made 100 dollers in a day, stop the game at the day.
- Play for 10 years(3600 days)

```
In [12]: df_result = pd.read_csv("/Users/koheisuzuki/Desktop/Bob_Suzuki Brainstation_Final_Pro
         df_result.head()
```

```
Out[12]:    Todays Amount  Todays Result  Total Amount
         0         -200.0              0        -200.0
         1         -200.0              0        -400.0
         2          100.0              1        -300.0
         3          112.5              1        -187.5
         4          112.5              1         -75.0
```

1.1.6 Check the winning percentage

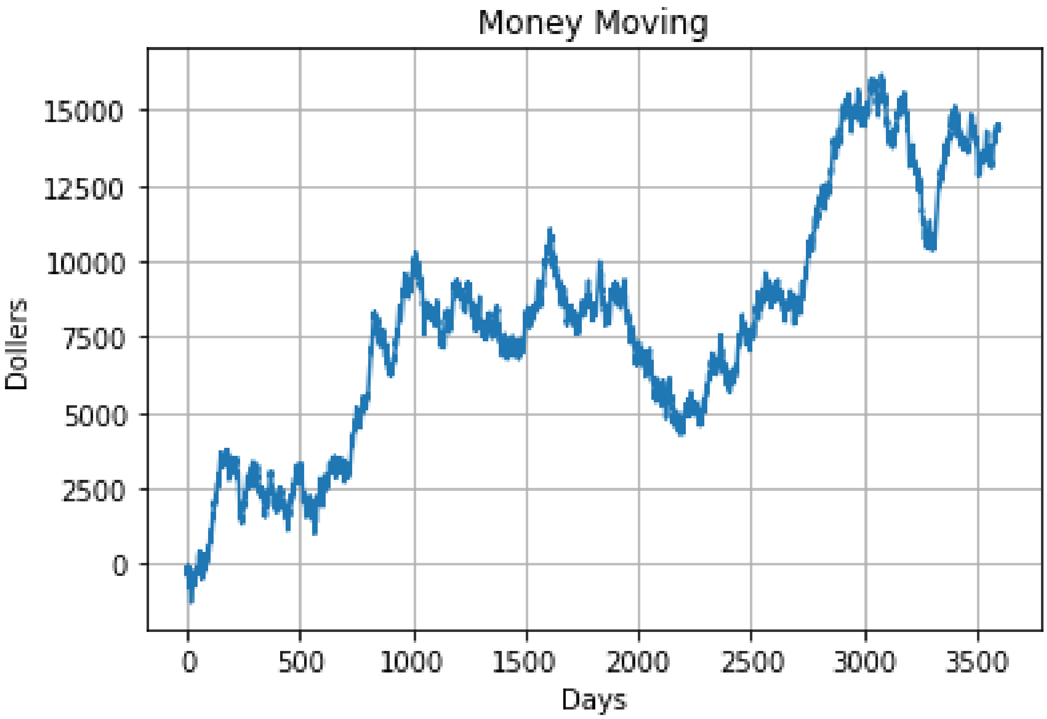
When player buyin is 200 dollers and goal is 100 dollers, player needs 66.66% for win. It means 2399.8days / 10years(3600days).

```
In [4]: len(df_result[df_result["Todays Result"] == 1])
```

```
Out[4]: 2402
```

So this means player has made money for 10 years
This is the result of player

```
In [5]: plt.plot(df_result.index, df_result["Total Amount"]);
        plt.title("Money Moving");
        plt.xlabel("Days");
        plt.ylabel("Dollers");
        plt.grid();
```



```
In [1451]: df_result.tail()

Out[1451]:       Todays Amount  Todays Result  Total Amount
           3595         -200.0              0       14300.0
           3596          100.0              1       14400.0
           3597          100.0              1       14500.0
           3598         -200.0              0       14300.0
           3599          100.0              1       14400.0
```

1.1.7 Result

When use following rules, *Player's buyin is 200 dollers / day* Player's goal is 100 dollers / day *When the true count is greater then equal to 1.5, use 1-2-1 betting system.

Reduction Rate (14400 + 200 * 3600) / 200 * 3600 = 1.019444....
So we can make money with Double Decks Black Jack at Parq Casino