Project 2: Wrangling and Exploratory Data Analysis

Mukun Guo 29 March 2020

In [1]:
```
import pandas as pd
import sqlite3
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Connect to the database
con = sqlite3.connect(r'data/lahman2016.sqlite')
```

Wrangling

Problem 1

In [3]:
```
# We use the following sql query to select data we need from the database, then map it to a pandas dat
aframe.
query = """ select S.yearID, S.teamID, S.lgID, playerID, franchID, W, G, sum(salary) as payroll, cast
(W as float)/cast(G as float)*100 as winRate
            from Salaries S join Teams T on T.yearID = S.yearID and T.teamID = S.teamID
            where S.yearID <= 2014 and S.yearID >= 1990
            group by S.teamID, S.yearID
            order by winRate desc; """
df = pd.read_sql_query(query, con)
df.head(3)
```

Out[3]:

	yearID	teamID	lgID	playerID	franchID	W	G	payroll	winRate
0	2001	SEA	AL	abbotpa01	SEA	116	162	74720834.0	71.604938
1	1998	NYA	AL	bankswi01	NYY	114	162	66806867.0	70.370370
2	1995	CLE	AL	alomasa02	CLE	100	144	37937835.0	69.444444

Exploratory data analysis

Payroll distribution

Problem 2

In [4]:
```
df_plot = df[['yearID', 'teamID', 'payroll']]
df_plot['payroll'] = df_plot['payroll'] / 1e8
df_plot.head(3)
```

Out[4]:

	yearID	teamID	payroll
0	2001	SEA	0.747208
1	1998	NYA	0.668069
2	1995	CLE	0.379378

In [5]:
```
from plotnine import *
from matplotlib import pyplot as plt

# We choose a boxplot to capture the distribution of payrolls.
# Here we set y-axis to be the payroll (in 10^8) and x-axis to be year.
# Each box represent the distribution of payroll in a year

p = (ggplot(df_plot, aes(x='factor(yearID)', y='payroll', fill='factor(yearID)')) +
     geom_boxplot() +
     labs(y="Payroll (1e8)", x = "Year") +
     ggtitle('Distribution of payroll over the years')
)

p
```

```
<Figure size 640x480 with 1 Axes>
```

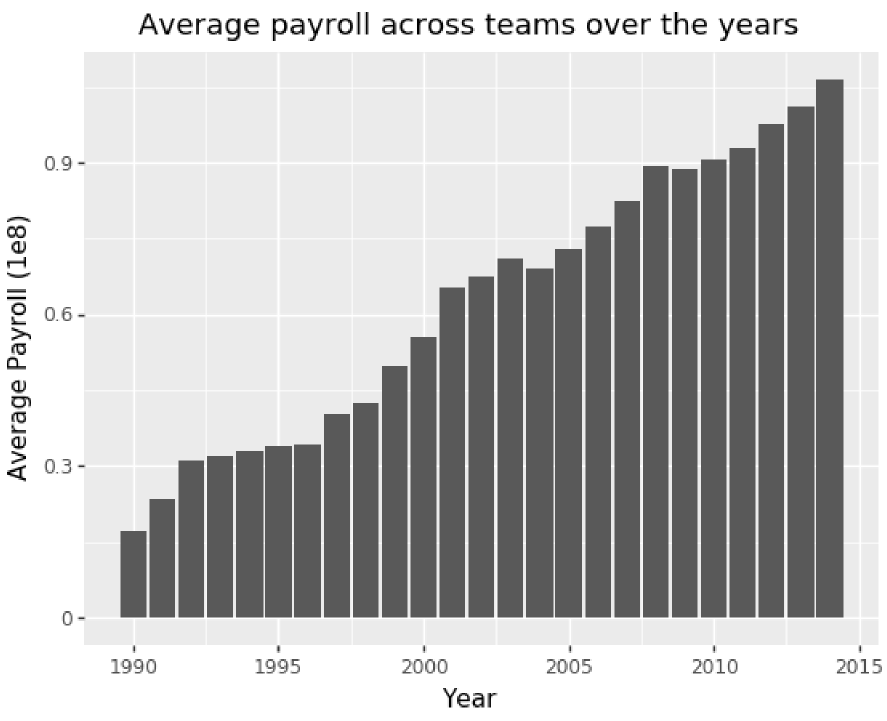
Out[5]: `<ggplot: (7555695601)>`

Question 1

The above plot shows the distribution of payroll over the eyars. Intuitively, we can see that teams are spending more and more over the years and the center of data is increasing. i.e. the average payroll of each year increases. We can going to explore further whether this statement is true by plotting the data in a different way.

Problem 3

In [6]:
```
# Here we use a barplot to capture the central tendency of data over the years
# We can see that even there are a few exceptions, generally the average payroll is increasing over th
e years
df_plot_mean = df_plot.groupby('yearID', as_index=False).mean()
p = (ggplot(df_plot_mean, aes(x='yearID', weight='payroll')) +
     geom_bar() +
     labs(y="Average Payroll (1e8)", x = "Year") +
     ggtitle('Average payroll across teams over the years'))

p
```



Out[6]: `<ggplot: (7555698273)>`

Correlation between payroll and winning percentage

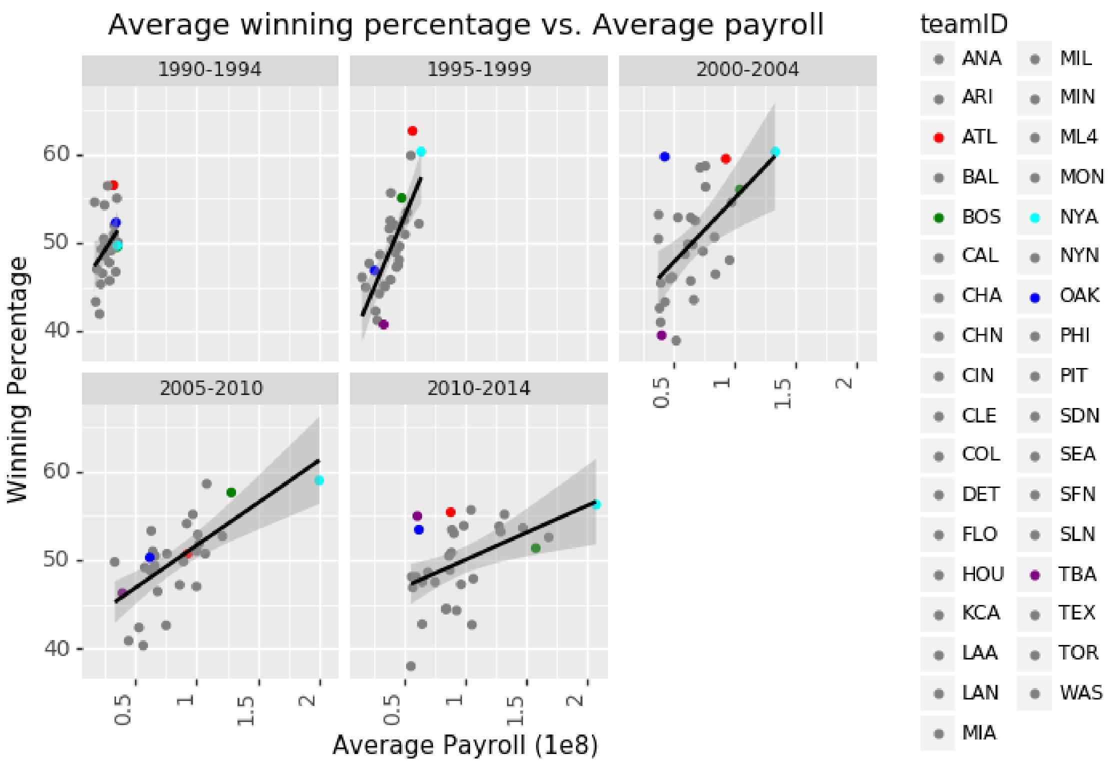
Problem 4

In [7]:
```python
# We create a new column called 'binned' to capture which time period the record belongs to
bins = 5
labels = ['1990-1994','1995-1999','2000-2004','2005-2010','2010-2014']
df['binned']= pd.cut(df['yearID'], bins=bins, labels=labels)
df_p4 = df.groupby(['binned', 'teamID'], as_index=False)['payroll', 'winRate'].mean()
df_p4['payroll'] = df_p4['payroll'] / 1e8

df_p4.head(3) # Some field is NaN because there is no record for that team in a certain time period
```

Out[7]:

	binned	teamID	payroll	winRate
0	1990-1994	ANA	NaN	NaN
1	1990-1994	ARI	NaN	NaN
2	1990-1994	ATL	0.317219	56.497726

In [8]:
```python
# We mark five teams ['OAK', 'ATL', 'NYA', 'BOS', 'TBA'] with colors to facilitate our analysis for Qu
estion2
cols = {}
for team in df['teamID']:
    cols[team] = 'grey'
cols['OAK'] = 'blue'
cols['ATL'] = 'red'
cols['NYA'] = 'cyan'
cols['BOS'] = 'green'
cols['TBA'] = 'purple'

# Here we use faceting to generate 5 plots of different time period, so that we can have a first impre
ssion
# of how a team performs over the years. The regression line shows the average performance of teams in
a time period.
p = (ggplot(df_p4, aes(x='payroll', y='winRate')) +
     geom_point(aes(color='teamID')) +
     scale_color_manual(cols) +
     geom_smooth(method='lm') +
     facet_wrap(['binned']) +
     theme(axis_text_x = element_text(size=10, angle=90, hjust=1)) +
     theme(axis_text_y = element_text(size=10)) +
     labs(y="Winning Percentage", x = "Average Payroll (1e8)")  +
     ggtitle('Average winning percentage vs. Average payroll')
    )
p
```



Out[8]: <ggplot: (7556382469)>

Question 2

The above plot shows the relation between the winning percentage and the average payroll of every team in a time period. For most teams, their performance in each time period varies. However, there are some teams that performs very good at paying for wins consistently, for example, ATL consistently stands out in paying for winning (or at least hit the average performance). ATL is indicating with blue in the above chart. OAK performs well consistently throughout 1990-2014 as well, and it stands out significantly in the period of 2000-2004, with a payroll of less than 5000000 and win rate of around 60% (indicated with red)

Data transformation

Standardization across years

Problem 5

In [9]:

```
# for every record in the dataframe, standardized_payroll = (payroll - average_payroll) / (std_payrol
l),
# where the mean and standard deviation is calculated using all the record of that year.
df['standardized_payroll'] = df.groupby('yearID')['payroll'].transform(lambda x: (x - x.mean()) / x.st
d())

# Here we also show the mean and standard deviation of each year
print('average payroll of each year\n-----------------------')
print(df.groupby('yearID')['payroll'].mean())
print('-----------------------')
print()
print('standard deviation of payroll of each year\n-----------------------')
print(df.groupby('yearID')['payroll'].std())
print('-----------------------')
print()

df.head(3)
```

```
average payroll of each year
-----------------------
yearID
1990    1.707235e+07
1991    2.357879e+07
1992    3.098244e+07
1993    3.220500e+07
1994    3.313701e+07
1995    3.398105e+07
1996    3.417798e+07
1997    4.026021e+07
1998    4.260943e+07
1999    4.980762e+07
2000    5.553784e+07
2001    6.535544e+07
2002    6.746925e+07
2003    7.094207e+07
2004    6.902220e+07
2005    7.295711e+07
2006    7.738242e+07
2007    8.255630e+07
2008    8.949529e+07
2009    8.882423e+07
2010    9.071200e+07
2011    9.281684e+07
2012    9.775804e+07
2013    1.011509e+08
2014    1.064106e+08
Name: payroll, dtype: float64
-----------------------

standard deviation of payroll of each year
-----------------------
yearID
1990    3.771834e+06
1991    6.894669e+06
1992    9.150607e+06
1993    9.232485e+06
1994    8.528749e+06
1995    9.447998e+06
1996    1.068853e+07
1997    1.306073e+07
1998    1.538081e+07
1999    2.056133e+07
2000    2.141622e+07
2001    2.470771e+07
2002    2.469219e+07
2003    2.801196e+07
2004    3.282411e+07
2005    3.417478e+07
2006    3.226495e+07
2007    3.390705e+07
2008    3.780200e+07
2009    3.385709e+07
2010    3.811503e+07
2011    4.081197e+07
2012    3.681754e+07
2013    4.883029e+07
2014    4.250538e+07
Name: payroll, dtype: float64
-----------------------
```

Out[9]:

	yearID	teamID	lgID	playerID	franchID	W	G	payroll	winRate	binned	standardized_payroll
0	2001	SEA	AL	abbotpa01	SEA	116	162	74720834.0	71.604938	2000-2004	0.379047
1	1998	NYA	AL	bankswi01	NYY	114	162	66806867.0	70.370370	1995-1999	1.573223
2	1995	CLE	AL	alomasa02	CLE	100	144	37937835.0	69.444444	1995-1999	0.418796

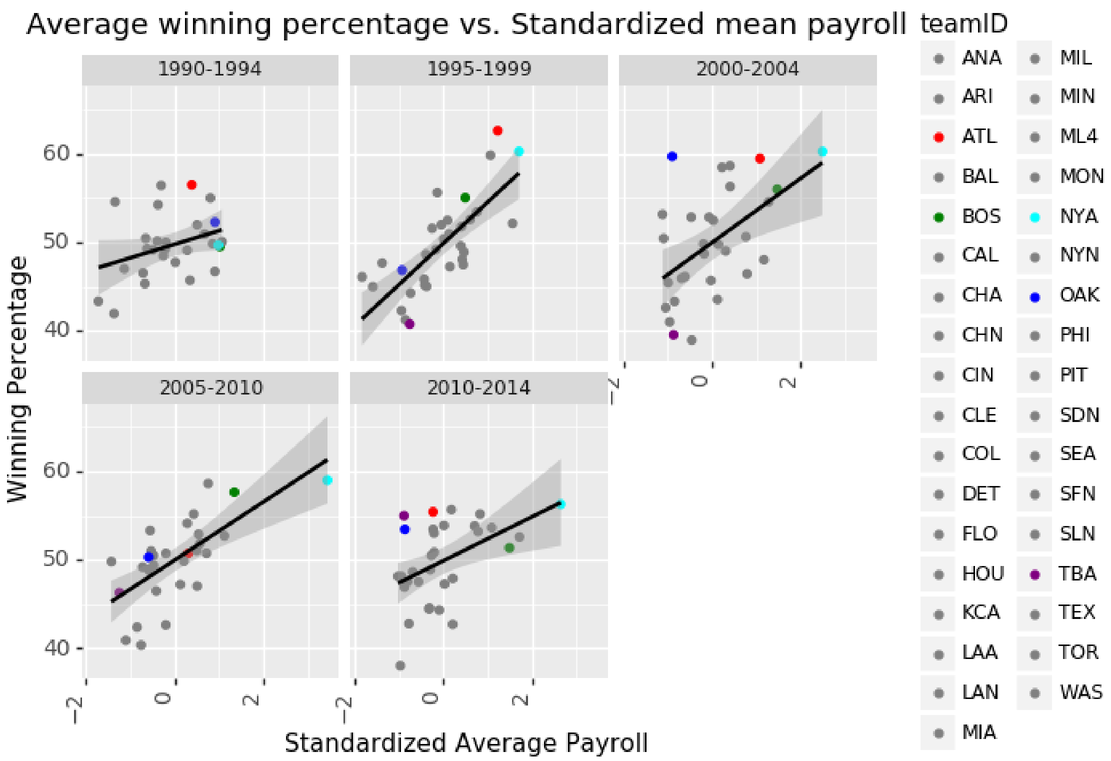
Problem 6

In [10]:
```
df_p6 = df.groupby(['binned', 'teamID'], as_index=False)['standardized_payroll', 'winRate'].mean()

df_p6.head(3)
```

Out[10]:

	binned	teamID	standardized_payroll	winRate
0	1990-1994	ANA	NaN	NaN
1	1990-1994	ARI	NaN	NaN
2	1990-1994	ATL	0.381441	56.497726

In [11]:
```
# Similar to Problem4, we mark some teams with color to facilitate our analysis
cols = {}
for team in df['teamID']:
    cols[team] = 'grey'

cols['OAK'] = 'blue'
cols['ATL'] = 'red'
cols['NYA'] = 'cyan'
cols['BOS'] = 'green'
cols['TBA'] = 'purple'

p = (ggplot(df_p6, aes(x='standardized_payroll', y='winRate')) +
     geom_point(aes(color='teamID')) +
     scale_color_manual(cols) +
     geom_smooth(method='lm') +
     facet_wrap(['binned']) +
     theme(axis_text_x = element_text(size=10, angle=90, hjust=1)) +
     theme(axis_text_y = element_text(size=10)) +
     labs(y="Winning Percentage", x = "Standardized Average Payroll") +
     ggtitle('Average winning percentage vs. Standardized mean payroll')
    )
p
```



Out[11]: <ggplot: (7557044041)>

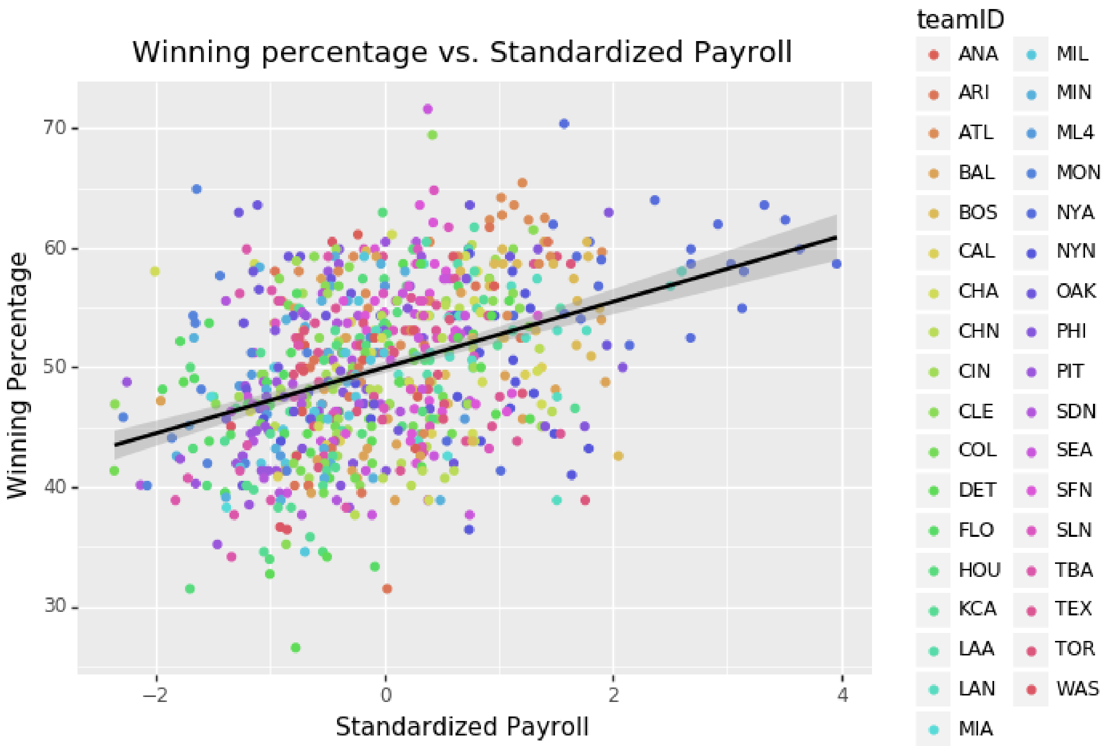
Question 3

The above plot shows the relation between winning percentage and standardized average payroll of a team in a time period. After the standardization, the *data range* of payroll changes from **the actual range of payroll** to **the number of standard deviation away from the data center (mean)**. And the *center of data* (mean) is changed to **0**. The *spread of the data* **remain the same**. Visually, the distribution of data points remains the same on the plot, but the range and unit of x-axis has changed. Moreover, standardization **conteract the impact of different range of payroll in different bins**, which can be seen by compare the "1990-1994" plot in Question 2 and Question 3

Expected wins

Problem 7

In [12]:
```
# We make a single scatter plot where the x-axis is the standardized payroll and y-axis is the winning
percentage
# Then we generate the regression line to see how are the two attributes correlated.
df_p7 = df[['standardized_payroll', 'winRate', 'teamID']]
p = (ggplot(df_p7, aes(x='standardized_payroll', y='winRate')) +
     geom_point(aes(color='teamID')) +
     geom_smooth(method='lm') +
     labs(y="Winning Percentage", x = "Standardized Payroll")  +
     ggtitle('Winning percentage vs. Standardized Payroll')
)

p
```



Out[12]: <ggplot: (7556371669)>

Spending efficiency

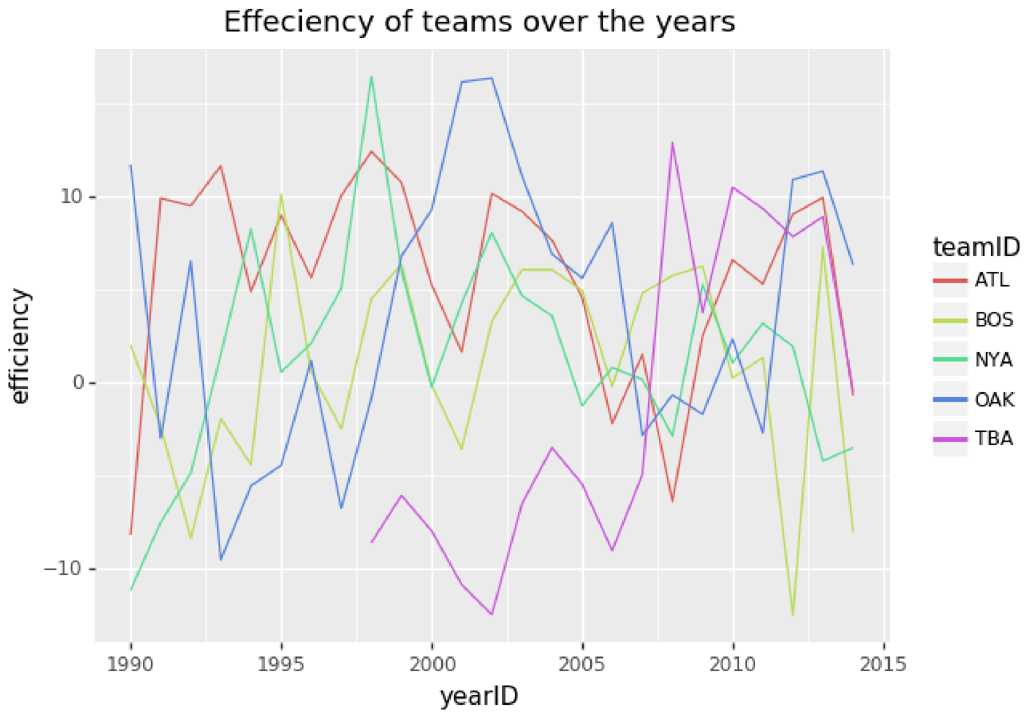
Problem 8

In [13]:
```
# We create a new column called 'effeciency to store the calculated effeciency of a  team in a certain
year'
expected_win_pct = lambda std_payroll: 50 + 2.5 * std_payroll
df['efficiency'] = df['winRate'] - df['standardized_payroll'].map(expected_win_pct)

df.head(3)
```

Out[13]:

	yearID	teamID	lgID	playerID	franchID	W	G	payroll	winRate	binned	standardized_payroll	efficiency
0	2001	SEA	AL	abbotpa01	SEA	116	162	74720834.0	71.604938	2000-2004	0.379047	20.657320
1	1998	NYA	AL	bankswi01	NYY	114	162	66806867.0	70.370370	1995-1999	1.573223	16.437314
2	1995	CLE	AL	alomasa02	CLE	100	144	37937835.0	69.444444	1995-1999	0.418796	18.397454

In [14]:
```
# We use geom_line instead of geom_smooth to have a more precise observation of the effeciency of ever
y team in a year.
# We choose the following 5 teams to see how good is their effeciency over the years.
df_p8 = df[df['teamID'].isin(['OAK', 'BOS', 'NYA', 'ATL', 'TBA'])][['yearID', 'teamID', 'efficiency']]

p = (ggplot(df_p8, aes(x='yearID', y='efficiency')) +
     geom_line(aes(color = 'teamID')) +
     ggtitle('Effeciency of teams over the years')
)

p
```



Out[14]: <ggplot: (7557303125)>

Question 4

The above plot a teams effeciency over the years. It explicitly shows how good a team performs given their total payroll. Basically, every team aims to perform as good as possible with a fixed amount of money, and the "effeciency" we defined help us to quantify it. By comparing the above plot with the ones in Question2 and Question3, we can see that teams with possible effeciency are typically those which lies above the regression line in the plot of Question2 and Question3. During the Moneyball period, Oakland's effeciency is roughly 16-17 as we can tell from the plot.