Client Report - Finding Relationships in Baseball

Course DS 250

Conner Crook

Elevator pitch

Baseball has been around for a long time and has been a very popular sport for just as long. There have been players from all over the Country that play in the Major League including some players from BYUI.

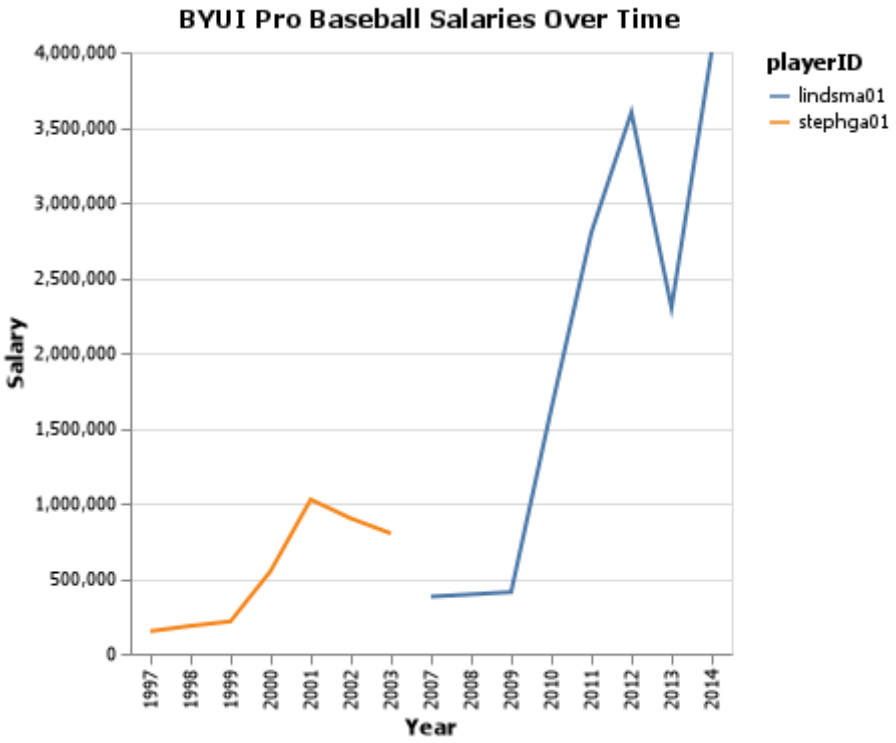
GRAND QUESTION 1

Write an SQL query to create a new dataframe about baseball players who attended BYU-Idaho. The new table should contain five columns: playerID, schoolID, salary, and the yearID/teamID associated with each salary. Order the table by salary (highest to lowest) and print out the table in your report.

There are two players that have come from BYUI. Matt Lindsrom (lindsma01) played the longest and had the highest salary with his highest salary being 4,000,000 in 2014. He started in 2007 with a salary of 380,000. Garrett Stephenson (stephga01) started in 1997 with a salary of 150,000 and had his highest salary in 2001 with a salary of 1,025,000.

TECHNICAL DETAILS

```
byui_baseball_chart = (alt.Chart(byui_baseball).mark_line()
    .encode(
        alt.X('yearID:O', title = "Year"),
        alt.Y('salary', title = "Salary"),
        color='playerID'
    )
    .properties(
        title={
            "text": ["BYUI Pro Baseball Salaries Over Time"]
        }
    ))
```




```
print(byui_baseball
    .head(30)
    .to_markdown(index=False))
```

playerID	schoolID	salary	yearID	teamID
lindsma01	idbyuid	4000000	2014	CHA
lindsma01	idbyuid	2300000	2013	CHA
lindsma01	idbyuid	3600000	2012	BAL
lindsma01	idbyuid	2800000	2011	COL
lindsma01	idbyuid	1625000	2010	HOU
lindsma01	idbyuid	410000	2009	FLO
lindsma01	idbyuid	395000	2008	FLO
lindsma01	idbyuid	380000	2007	FLO
lindsma01	idbyuid	4000000	2014	CHA
lindsma01	idbyuid	2300000	2013	CHA
lindsma01	idbyuid	3600000	2012	BAL

playerID	schoolID	salary	yearID	teamID
lindsma01	idbyuid	2800000	2011	COL
lindsma01	idbyuid	1625000	2010	HOU
lindsma01	idbyuid	410000	2009	FLO
lindsma01	idbyuid	395000	2008	FLO
lindsma01	idbyuid	380000	2007	FLO
stephga01	idbyuid	800000	2003	SLN
stephga01	idbyuid	900000	2002	SLN
stephga01	idbyuid	1025000	2001	SLN
stephga01	idbyuid	550000	2000	SLN
stephga01	idbyuid	215000	1999	SLN
stephga01	idbyuid	185000	1998	PHI
stephga01	idbyuid	150000	1997	PHI
stephga01	idbyuid	800000	2003	SLN
stephga01	idbyuid	900000	2002	SLN
stephga01	idbyuid	1025000	2001	SLN
stephga01	idbyuid	550000	2000	SLN
stephga01	idbyuid	215000	1999	SLN
stephga01	idbyuid	185000	1998	PHI
stephga01	idbyuid	150000	1997	PHI

GRAND QUESTION 2

This three-part question requires you to calculate batting average (number of hits divided by the number of at-bats)

1. Write an SQL query that provides playerID, yearID, and batting average for players with at least one at bat. Sort the table from highest batting average to lowest, and show the top 5 results in your report.

2. Use the same query as above, but only include players with more than 10 “at bats” that year. Print the top 5 results.

3. Now calculate the batting average for players over their entire careers (all years combined). Only include players with more than 100 at bats, and print the top 5 results.

The more times you have at bat, the lower your batting average is. When showing the top 5 batting averages for those people who have had to bat at least once, we get 5 players with a perfect batting score with a batting average of 1. With those players that have had at least 10 chances at bat, that batting average drops with a high of 0.643. This drops even more when showing the top batting averages for those that have had at least 100 chances at bat. The high batting average for these players is 0.492.

TECHNICAL DETAILS

```
print(batting_average_one
    .head(20)
    .to_markdown(index=False))
```

playerID	yearID	batting_average
snowch01	1874	1
baldwki01	1884	1
oconnfr01	1893	1
gumbebi01	1893	1
mccafsp01	1889	1

```
print(batting_average_ten
    .head(20)
    .to_markdown(index=False))
```

playerID	yearID	batting_average
nymanny01	1974	0.642857
carsoma01	2013	0.636364
altizda01	1910	0.6
johnsde01	1975	0.6

playerID	yearID	batting_average
silvech01	1948	0.571429

```
print(batting_average_hundred
    .head(20)
    .to_markdown(index=False))
```

playerID	yearID	batting_average
meyerle01	1871	0.492308
mcveyca01	1871	0.431373
jacksjo01	1911	0.408056
hazlebo01	1957	0.402985
barnero01	1871	0.401274

GRAND QUESTION 3

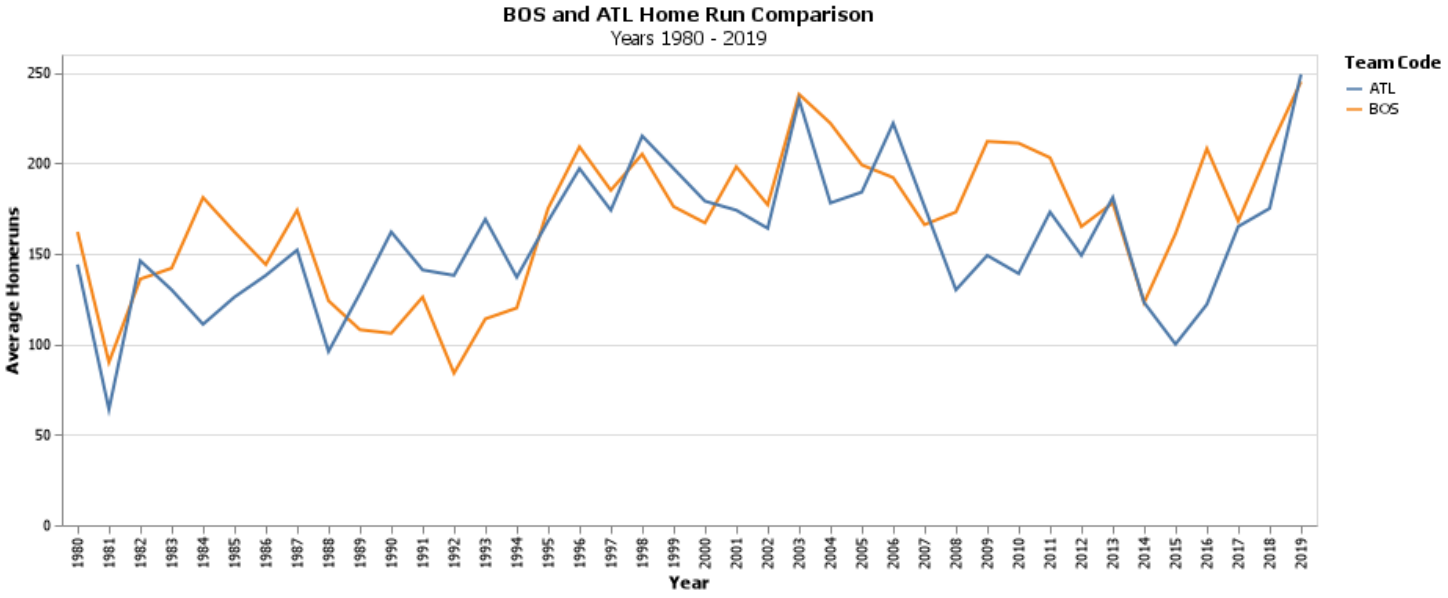
Pick any two baseball teams and compare them using a metric of your choice (average salary, home runs, number of wins, etc.). Write an SQL query to get the data you need. Use Python if additional data wrangling is needed, then make a graph in Altair to visualize the comparison. Provide the visualization and its description.

The Boston Red Sox (BOS) and the Atlanta Braves(ATL) are both Major League teams that are well known in baseball. Both teams show similar trends of how many homeruns they recieve every year since 1980. Similarly, both BOS and ATL had their highest number of homeruns in 2019. ATL had their lowest number of home runs in 1981 and BOS had their lowest number of home runs in 1992.

TECHNICAL DETAILS

```
compare_chart = (alt.Chart(compare_home_runs).mark_line()
    .encode(
        alt.X('yearid:O', title = "Year"),
        alt.Y('Average_Homeruns', title = "Average Homeruns"),
        alt.Color('franchid', title = "Team Code")
    )
    .properties(
        title={
            "text": ["BOS and ATL Home Run Comparison"],
            "subtitle": ["Years 1980 - 2019"]
        }
    ))
```

insert your chart png here



APPENDIX A (PYTHON CODE)

```
#%%
import pandas as pd
import numpy as np
import altair as alt
import datadotworld as dw
#%%
# Testing
results = dw.query('byuidss/cse-250-baseball-database',
    'SELECT playerID FROM Batting').dataframe
results
#%%
##GRAND QUESTION 1
# Pulls playerID, schoolID, salary, and the yearID/teamID associated with each salary
byui_baseball = dw.query('byuidss/cse-250-baseball-database',
    'SELECT p.playerID, c.schoolID, s.salary, s.yearID, s.teamID '
    'FROM People as p INNER JOIN CollegePlaying as c ON p.playerID = c.playerID '
    'INNER JOIN Salaries as s ON p.playerID = s.playerID '
    'INNER JOIN Schools as sc ON c.schoolID = sc.schoolID '
    'WHERE sc.name_full = "Brigham Young University-Idaho"').dataframe
byui_baseball
#%%
# Prints table
print(byui_baseball
    .head(30)
    .to_markdown(index=False))
#%%
# Create Line Chart
byui_baseball_chart = (alt.Chart(byui_baseball).mark_line()
    .encode(
        alt.X('yearID:O', title = "Year"),
        alt.Y('salary', title = "Salary"),
        color='playerID'
    )
    .properties(
        title={
            "text": ["BYUI Pro Baseball Salaries Over Time"]
        }
    ))
byui_baseball_chart.save('byui_baseball_chart.png')
#%%
## GRAND QUESTION 2
# Part 1 - Query playerID, yearID, and calculate the batting average for everyone who has at lea
batting_average_one = dw.query('byuidss/cse-250-baseball-database',
    'SELECT playerID, yearID, h/ab AS batting_average '
    'FROM Batting '
    'WHERE ab >= 1 '
    'ORDER BY batting_average DESC LIMIT 5').dataframe
#%%
#Print to a markdown table
print(batting_average_one
```

```
    .head(20)
    .to_markdown(index=False))
#%%
#part 2 - Same query as above but only include those who have at least 10 at bats.
batting_average_ten = dw.query('byuidss/cse-250-baseball-database',
    'SELECT playerID, yearID, h/ab AS batting_average '
    'FROM Batting '
    'WHERE ab >= 10 '
    'ORDER BY batting_average DESC LIMIT 5').dataframe
#%%
#Print above to markdown table
print(batting_average_ten
    .head(20)
    .to_markdown(index=False))
#%%
#Part 3 - Only include those who have at more than 100 at bats combined over their whole career
batting_average_hundred = dw.query('byuidss/cse-250-baseball-database',
    'SELECT playerID, yearID, h/ab AS batting_average '
    'FROM Batting '
    'WHERE ab > 100 '
    'GROUP BY playerID '
    'ORDER BY batting_average DESC LIMIT 5').dataframe
#%%
# Create Markdown table for above
print(batting_average_hundred
    .head(20)
    .to_markdown(index=False))
#%%
##GRAND QUESTION 3
#query from TeamFranchises and Team tables to find the average homeruns for BOS and ATL
compare_home_runs = dw.query('byuidss/cse-250-baseball-database',
    'SELECT tf.franchname ,t.franchid, t.yearid, SUM(t.hr) as Average_Homeruns '
    'FROM Teams as t INNER JOIN TeamsFranchises as tf '
    'ON t.franchid = tf.franchid '
    'WHERE (t.franchid = "BOS" OR t.franchid = "ATL") AND t.yearid > 1979 '
    'GROUP BY t.franchid, t.yearid').dataframe
#%%
#Create Chart for the above query
compare_chart = (alt.Chart(compare_home_runs).mark_line()
    .encode(
        alt.X('yearid:O', title = "Year"),
        alt.Y('Average_Homeruns', title = "Average Homeruns"),
        alt.Color('franchid', title = "Team Code")
    )
    .properties(
        title={
            "text": ["BOS and ATL Home Run Comparison"],
            "subtitle": ["Years 1980 - 2019"]
        }
    ))
#%%
```