codecademy

Visualizing Data With Seaborn

World Cup

Visualizing Data With Matplotlib

Alex Ricciardi

Date: 06/16/2020

Project Overview

Using the Python graphing module, Matplotlib and the Seaborn library visualize the World Cup data from the Fifa World Cup from 1930-2014 to analyze trends and discover insights about the world's game, fútbol!

The project:

1. Import the modules that you'll be using in this project.
2. Inspect the raw CSV files.
3. Load WorldCupMatches.csv into a DataFrame called df.
4. Inspect the DataFrame using .head().
5. Create a new column in df named Total Goals, and set it equal to the sum of the columns Home Team Goals and Away Team Goals.
6. You are going to create a bar chart visualizing how many goals were scored each year the World Cup was held between 1930-2014. Set the style of your plot to be whitegrid.
7. To make the text in your visualization bigger and easier to read, set the context to be "poster".
8. Create a figure and axes for your plot using the syntax: f, ax = plt.subplots()
9. Using the data in df and the syntax: ax = sns.barplot(), visualize the columns Year and Total Goals as a bar chart.
10. Render your bar chart so you can see it.
11. Effective visualizations include a clear title.
12. Load goals.csv into a DataFrame called df_goals.
13. Experimenting with different contexts and font scales can help you decide on the best context and font scale for the particular visualization.
14. Set ax2 equal to a box plot with the color palette Spectral that visualizes the data in the DataFrame df_goals with the column year on the x-axis and goals on the y-axis.
15. Give your box plot a meaningful and clear title.

Task-1:
Import the libraries

1 Importing Libraries

Matplotlib: Visualization with Python
https://matplotlib.org/

Pandas: Data manipulation and analysis
https://pandas.pydata.org/

Seaborn: Statistical data visualization
Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
https://seaborn.pydata.org/

Code

```
# Matplotlib
from matplotlib import pyplot as plt
# Pandas
import pandas as pd
# Seaborn
import seaborn as sns
```

Tack-3:
Load WorldCupMatches.csv

2.1 overview

Load WorldCupMatches.csv into a DataFrame called df.
This will allow you to eventually plot the DataFrame with Seaborn.

2.2 Load WorldCupMatches.csv

The pandas I/O API is a set of top level reader functions accessed like pandas.read_csv() that generally return a pandas object. The corresponding writer functions are object methods that are accessed like DataFrame.to_csv(). Below is a table containing available readers and writers.
https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html

pandas.read_csv
Read a comma-separated values (csv) file into DataFrame.
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Code

```
df = pd.read_csv('WorldCupMatches.csv')
```

Task-4:
Inspect the DataFrame

4.1 overview

It is usually a good idea to check any new DataFrame to make sure the results are as expected.

Inspect the DataFrame using .head(). Make sure to use print() to wrap any output you want to inspect.

4.2 Inspect the DataFrame using .head()

Code

```
print(df.head())
```

Output

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	Attendance	Half-time Home Goals	Half-time Away Goals	Referee	Assistant 1	Assistant 2	RoundID	MatchID	Home Team Initials	Away Team Initial
0	1930	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4	1	Mexico		4444.0	3	0	LOMBARDI Domingo (URU)	CRISTOPHE Henry (BEL)	REGO Gilberto (BRA)	201	1096	FRA	MEX
1	1930	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3	0	Belgium		18346.0	2	0	MACIAS Jose (ARG)	MATEUCCI Francisco (URU)	WARNKEN Alberto (CHI)	201	1090	USA	BEL
2	1930	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2	1	Brazil		24059.0	2	0	TEJADA Anibal (URU)	VALLARINO Ricardo (URU)	BALWAY Thomas (FRA)	201	1093	YUG	BRA
3	1930	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3	1	Peru		2549.0	1	0	WARNKEN Alberto (CHI)	LANGENUS Jean (BEL)	MATEUCCI Francisco (URU)	201	1098	ROU	PER

Task-5:
Create a new column in df named Total Goals

5.1 overview

The data in WorldCupMatches.csv has the goals scored in each match broken up by goals for the home team and goals for the away team. We want to visualize the total number of goals scored in each match.

Create a new column in df named Total Goals, and set it equal to the sum of the columns Home Team Goals and Away Team Goals.

Print the results of df.head() to confirm your new column.

5.2 Create a new column in df named Total Goals

Code

```
df['Total Goals'] = df['Home Team Goals'] + df['Away Team Goals']
print(df.head(10))
```

Output

	Year	Datetime	Stage	Stadium	City	Home Team Name	Home Team Goals	Away Team Goals	Away Team Name	Win conditions	Attendance	Half-time Home Goals	Half-time Away Goals	Referee	Assistant 1	Assistant 2	RoundID	MatchID	Home Team Initials	Away Team Initials	Total Goals
0	1930	13 Jul 1930 - 15:00	Group 1	Pocitos	Montevideo	France	4	1	Mexico		4444.0	3	0	LOMBARDI Domingo (URU)	CRISTOPHE Henry (BEL)	REGO Gilberto (BRA)	201	1096	FRA	MEX	5
1	1930	13 Jul 1930 - 15:00	Group 4	Parque Central	Montevideo	USA	3	0	Belgium		18346.0	2	0	MACIAS Jose (ARG)	MATEUCCI Francisco (URU)	WARNKEN Alberto (CHI)	201	1090	USA	BEL	3
2	1930	14 Jul 1930 - 12:45	Group 2	Parque Central	Montevideo	Yugoslavia	2	1	Brazil		24059.0	2	0	TEJADA Anibal (URU)	VALLARINO Ricardo (URU)	BALWAY Thomas (FRA)	201	1093	YUG	BRA	3
3	1930	14 Jul 1930 - 14:50	Group 3	Pocitos	Montevideo	Romania	3	1	Peru		2549.0	1	0	WARNKEN Alberto (CHI)	LANGENUS Jean (BEL)	MATEUCCI Francisco (URU)	201	1098	ROU	PER	4
4	1930	15 Jul 1930 - 16:00	Group 1	Parque Central	Montevideo	Argentina	1	0	France		23409.0	0	0	REGO Gilberto (BRA)	SAUCEDO Ulises (BOL)	RADULESCU Constantin (ROU)	201	1085	ARG	FRA	1

Task-6:
Goals scored per year between 1930-2014.

6.1 overview

You are going to create a bar chart visualizing how many goals were scored each year the World Cup was held between 1930-2014.

Set the style of your plot to be whitegrid . This will add gridlines to the plot which will make it easier to read the visualization.

6.2 Goals scored per year between 1930-2014

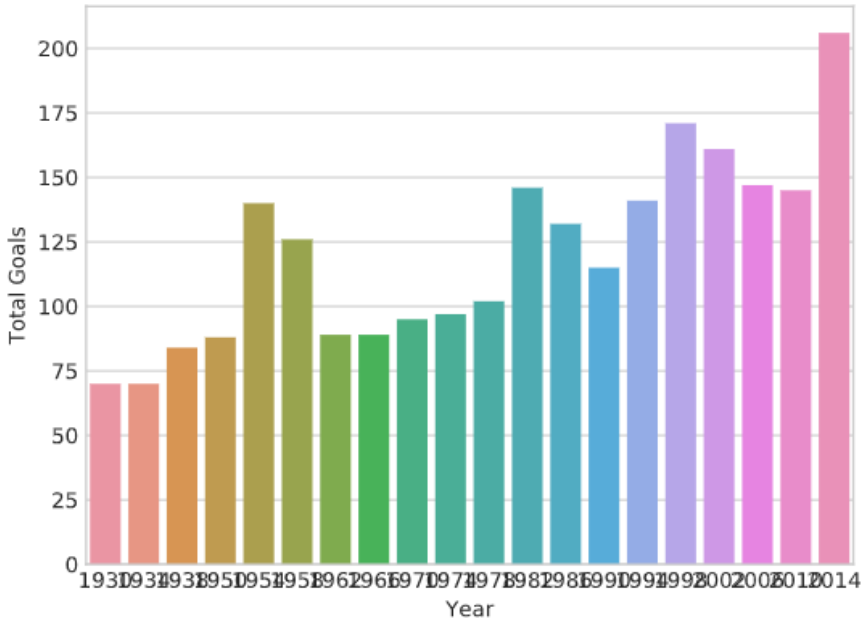
Code

```
sns.set_style("whitegrid")
sns.barplot(data=df, x='Year', y='Total Goals', ci=None, estimator=sum)
plt.show()
```

The x axis ticks label, the years, are not legible.

Graph style adjustments are needed to make the text easier to read

Output



Task-7:
Style Adjustments.

7.1 overview

To make the text in your visualization bigger and easier to read, set the context to be "poster".

If you would like to further adjust the font size of your plot, you can pass sns.set_context() a second optional argument using the keyword font_scale.

6.2 Graph style adjustments

Code

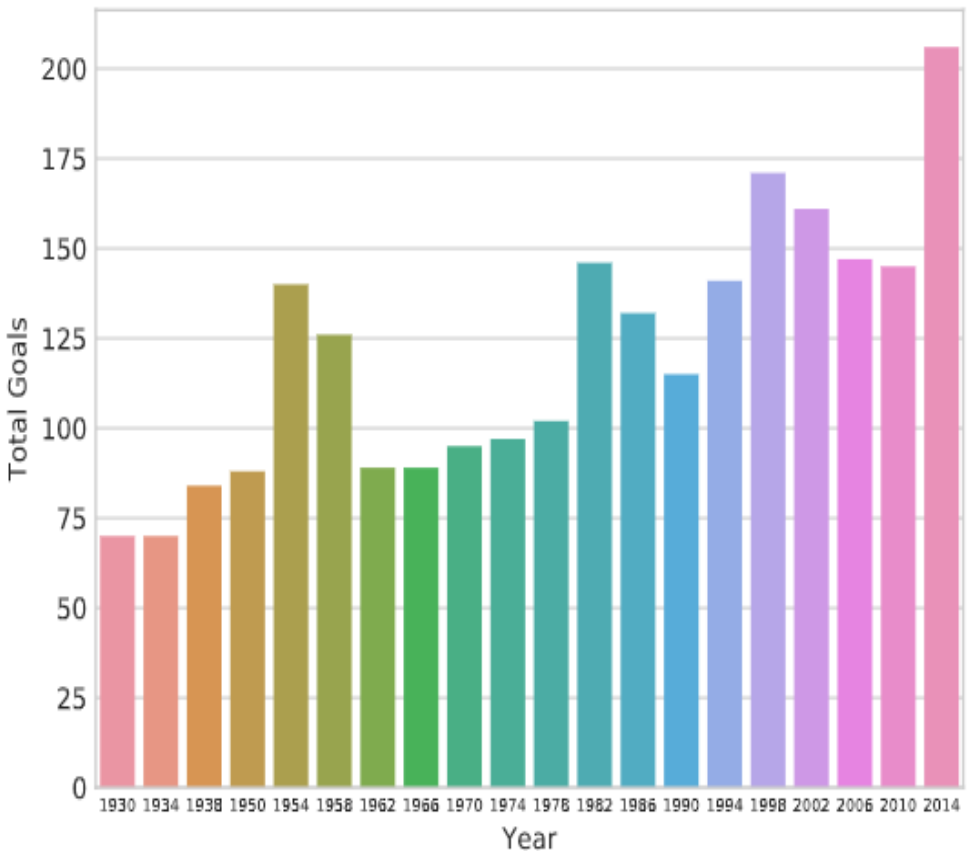
```
sns.set_style("whitegrid")
sns.set_context( rc={'font.size': 10, 'xtick.labelsize': 6})
sns.barplot(data=df, x='Year', y='Total Goals', ci=None,
estimator=sum)
plt.show()
```

Using sns.set_context("poster") did not achieved a graph balanced visual, the x and y labels were oversized compared to the rest of the graph.

I used the rc parameters within the sns.set_context() function to achieve a more graph balanced visual.

https://github.com/mwaskom/seaborn/blob/master/seaborn/rcmod.py

Output



Task-8:
Create a figure

8.1 overview

Create a figure and axes for your plot using the syntax:
f, ax = plt.subplots()

Inside of plt.subplots(), set the size of the figure to be 12 inches wide and 7 inches tall.

6.2 Graph style adjustments

Code

```
f, ax = plt.subplots(figsize=(12, 7))
```

The code synthase f, is short for fig, the figure parameter of the Matplotlib module.

https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.subplots.html

Task-9, Task-10 and Task-11: Create axes

9-10-11.1 overview

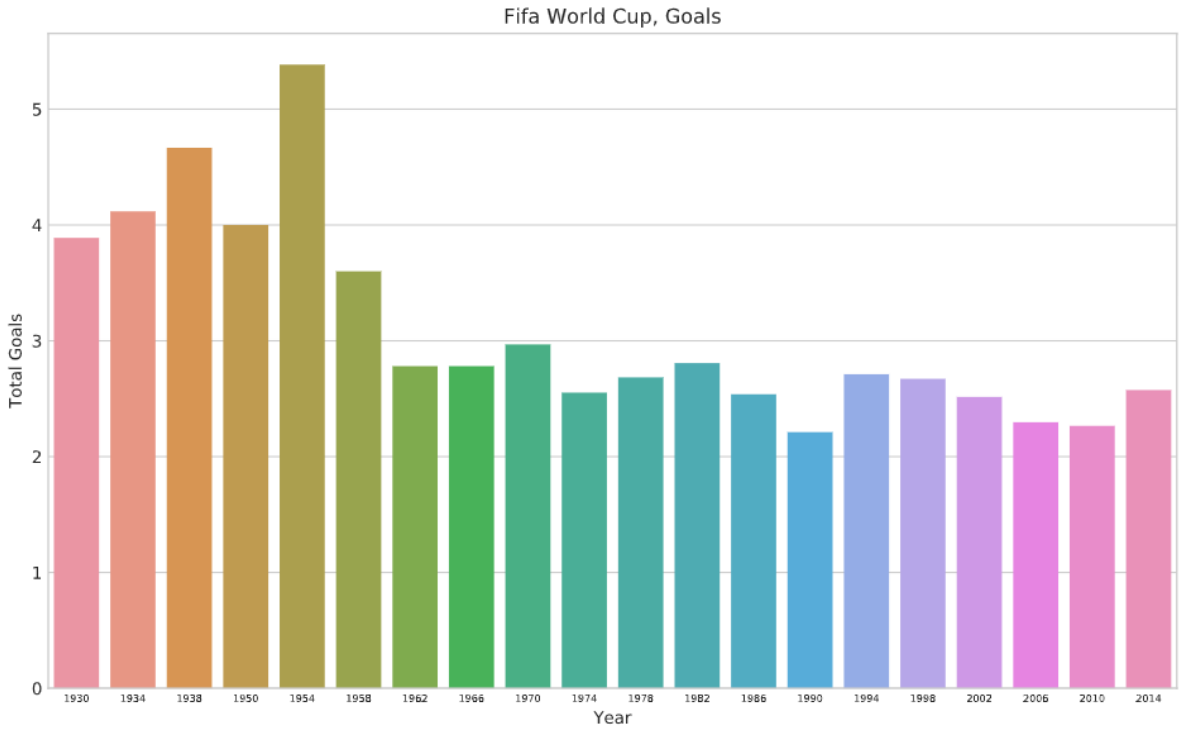
9. Using the data in df and the syntax:
   ax = sns.barplot()
   visualize the columns Year and Total Goals as a bar chart.
   Year should be on the x-axis, and Total Goals should be on the y-axis.

10. Render your bar chart so you can see it.

11. Effective visualizations include a clear title.
    Give your bar chart a meaningful title using ax.set_title().

9-10-11.2 Graph

Output



Code

```
# ----------------- 9.
ax = sns.barplot(data=df, x='Year',
y='Total Goals', ci=None)
# ----------------- 11.
ax.set_title('Fifa World Cup, Goals')
# ----------------- 10.
plt.show()
```

Task-12:
Load goals.csv

12.1 overview

Now you are going to create a box plot so you can visualize the distribution of the goals data instead of just the average with a bar chart.

Load goals.csv into a DataFrame called df_goals, and take a quick look at the DataFrame using .head().

12.2 Load goals.csv into a DataFrame called df_goals

Code

```
df_goals = pd.read_csv('goals.csv')
print(df_goals.head())
```

Output

	goals	home/away	year
0	4	home	1930
1	3	home	1930
2	2	home	1930
3	3	home	1930
4	1	home	1930

Task-13 to Task-17: Visualize df_goals data distribution

13 to 17.1 overview

13. Experimenting with different contexts and font scales can help you decide on the best context and font scale for the particular visualization.Try setting the context of the plot to be notebook and the font_scale to be 1.25.

14. Create a figure for your second plot. Set the variables f, ax2 and instantiate a figure that is 12 inches wide and 7 inches tall.

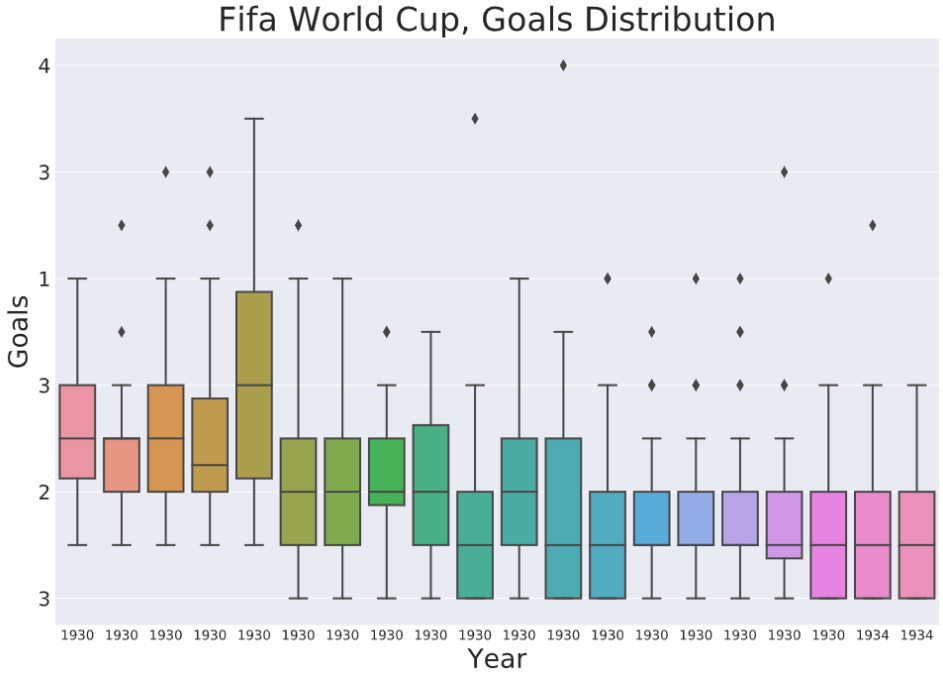
15. Set ax2 equal to a box plot with the color palette Spectral that visualizes the data in the DataFrame df_goals with the column year on the x-axis and goals on the y-axis.

16. Give your box plot a meaningful and clear title.

17. Render your box plot so you can see it.

13 to 17.2 Data Distribution Graph

Output



Code

```
sns.set_style("darkgrid")
f, ax2 = plt.subplots(figsize=(12, 8))

ax2 = sns.boxplot(data=df_goals, x='year', y='goals’)

ax2.set_xlabel('Year', fontsize=20)
ax2.set_ylabel('Goals', fontsize=20)

ax2.set_title('Fifa World Cup, Goals Distribution',
fontsize=25)

ax2.set_xticklabels(df_goals.year, fontsize=10)
ax2.set_yticklabels(df_goals.goals, fontsize=15)

plt.show(ax2)
```