FRAME WORK MINI PROJECT

TITTLE: EDA ON FIFA MEN AND WOMEN PLAYER DATASET

GENRE: SPORTS

NAME: GOKUL PRASATH G.K

DEPT: AIML

YEAR: 3RD

SEM: 5TH

AIM:

To perform exploratory data visualization on the FIFA dataset using Python (Matplotlib, Seaborn, Plotly), Power BI, and Tableau, and compare their visualization capabilities.

OBJECTIVES:

- Load and analyze the FIFA player dataset.
- Perform data cleaning and transformation.
- Create visualizations using Python.
- Build dashboards in Power BI and Tableau.
- Interpret insights from all tools.

DATASET DESCRIPTION:

Dataset Name: FIFA Data for EDA and Stats (Kaggle)
Source: https://www.kaggle.com/datasets/mukeshmanral/fifa-data-for-eda-and-stats

Attribute	Description
Name	Player name
Age	Player age
Nationality	Country of origin
Overall	Overall rating
Potential	Potential rating
Club	Team/Club name
Value	Market value in Euros
Wage	Player weekly wage
Position	Playing position
Work Rate	Player work intensity
Preferred Foot	Right or Left foot

TOOLS USED:

- **Python Libraries:** Pandas, Matplotlib, Seaborn, Plotly, NumPy
- **Data Visualization Tools:** Power BI, Tableau

Python Implementation :

CODE:

```
!pip install kagglehub pandas matplotlib seaborn plotly --quiet

import kagglehub
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import numpy as np
import os

sns.set_style("whitegrid")

dataset_dir = kagglehub.dataset_download("mukeshmanral/fifa-data-for-eda-and-stats")
print("Dataset downloaded to:", dataset_dir)

files = os.listdir(dataset_dir)
csv_file = os.path.join(dataset_dir, files[0])
df = pd.read_csv(csv_file)

def convert_value(x):
    if isinstance(x, str):
        x = x.replace('€','').replace('K','000').replace('M','000000')
        return float(x)
    return x

for col in ['Value','Wage']:
    if col in df.columns:
        df[col] = df[col].apply(convert_value)



plt.figure(figsize=(10,5))
top_players = df[['Name', 'Overall']].sort_values(by='Overall', ascending=False).head(10)
sns.barplot(x='Overall', y='Name', data=top_players, palette="viridis")
plt.title("Top 10 Players: Comparing Overall Ratings", fontsize=14)
plt.xlabel("Overall Rating")
plt.ylabel("Player Name")
plt.show()

plt.figure(figsize=(10,5))
sns.histplot(df['Age'], bins=20, kde=True, color='skyblue')
plt.title("Player Age Distribution", fontsize=14)
plt.xlabel("Age")
plt.ylabel("Number of Players")
plt.show()

plt.figure(figsize=(12,6))
sns.boxplot(x='Position', y='Overall', data=df, palette="Set2")
plt.xticks(rotation=45)
plt.title("Overall Rating by Position", fontsize=14)
plt.xlabel("Position")
plt.ylabel("Overall Rating")
plt.show()

plt.figure(figsize=(10,5))
sns.scatterplot(x='Overall', y='Potential', data=df, hue='Age', palette='coolwarm',
alpha=0.7)
plt.title("Overall vs Potential by Age", fontsize=14)
plt.xlabel("Overall Rating")
plt.ylabel("Potential Rating")
plt.legend(title='Age', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

plt.figure(figsize=(12,10))
corr = df.select_dtypes(include='number').corr()
mask = np.triu(np.ones_like(corr, dtype=bool))
sns.heatmap(corr, mask=mask, annot=False, cmap="coolwarm", linewidths=0.5,
cbar_kws={"shrink": 0.8}, square=True)
plt.title("Correlation Matrix: Relationships Between Numeric Attributes", fontsize=14)
plt.show()

plt.figure(figsize=(12,6))
top_nationalities = df['Nationality'].value_counts().head(15)
sns.barplot(x=top_nationalities.values, y=top_nationalities.index, palette="magma")
plt.title("Top 15 Nationalities: Player Count Comparison", fontsize=14)
plt.xlabel("Number of Players")
plt.ylabel("Nationality")
plt.show()

if 'Preferred Foot' in df.columns:
    foot_counts = df['Preferred Foot'].value_counts()
    plt.figure(figsize=(6,6))
    plt.pie(foot_counts, labels=foot_counts.index, autopct='%1.1f%%',
colors=['#ff9999','#66b3ff'], startangle=90)
    plt.title("Preferred Foot Distribution", fontsize=14)
    plt.show()

avg_overall_age = df.groupby('Age')['Overall'].mean().reset_index()
plt.figure(figsize=(10,5))
sns.lineplot(x='Age', y='Overall', data=avg_overall_age, marker='o', color='darkgreen')
plt.title("Average Overall Rating by Age", fontsize=14)
plt.xlabel("Age")
plt.ylabel("Average Overall")
plt.show()

plt.figure(figsize=(10,5))
sns.kdeplot(df['Potential'], fill=True, color='green')
plt.title("Distribution of Player Potential", fontsize=14)
plt.xlabel("Potential")
plt.ylabel("Density")
plt.show()

plt.figure(figsize=(12,6))
sns.stripplot(x='Position', y='Overall', data=df, size=3, jitter=True, palette="Set3")
plt.xticks(rotation=45)
plt.title("Overall Rating by Position (Strip Plot)", fontsize=14)
plt.xlabel("Position")
plt.ylabel("Overall Rating")
plt.show()

if 'Work Rate' in df.columns:
    plt.figure(figsize=(12,6))
    sns.violinplot(x='Work Rate', y='Overall', data=df, palette="Pastel1")
    plt.xticks(rotation=45)
    plt.title("Overall Rating by Work Rate", fontsize=14)
    plt.xlabel("Work Rate")
    plt.ylabel("Overall Rating")
    plt.show()

from pandas.plotting import scatter_matrix
numeric_cols = df.select_dtypes(include='number').columns.tolist()
scatter_matrix(df[numeric_cols], figsize=(15,15), diagonal='kde', color='purple')
plt.suptitle("Scatter Matrix: Comparing Key Numeric Metrics", fontsize=16)
plt.show()
```

```
if 'Club' in df.columns:
    top_clubs = df['Club'].value_counts().head(10)
    plt.figure(figsize=(10,5))
    sns.barplot(x=top_clubs.values, y=top_clubs.index, palette='coolwarm')
    plt.title("Top 10 Clubs by Number of Players", fontsize=14)
    plt.xlabel("Number of Players")
    plt.ylabel("Club")
    plt.show()

if 'Nationality' in df.columns and 'Club' in df.columns:
    sunburst_df = df.groupby(['Nationality','Club']).size().reset_index(name='PlayerCount')
    fig = px.sunburst(sunburst_df, path=['Nationality','Club'], values='PlayerCount',
                      color='PlayerCount', color_continuous_scale='RdBu',
hover_data=['PlayerCount'])
    fig.update_traces(textinfo='label')
    fig.update_layout(title="Players by Nationality and Club (Clean View)")
    fig.show()

fig = px.scatter(df, x='Age', y='Overall', size='Potential', color='Nationality',
hover_name='Name',
                 size_max=60, opacity=0.6)
fig.update_layout(title="Age vs Overall with Potential as Bubble Size")
fig.show()
```

Pythonoutput:

Power bi output:

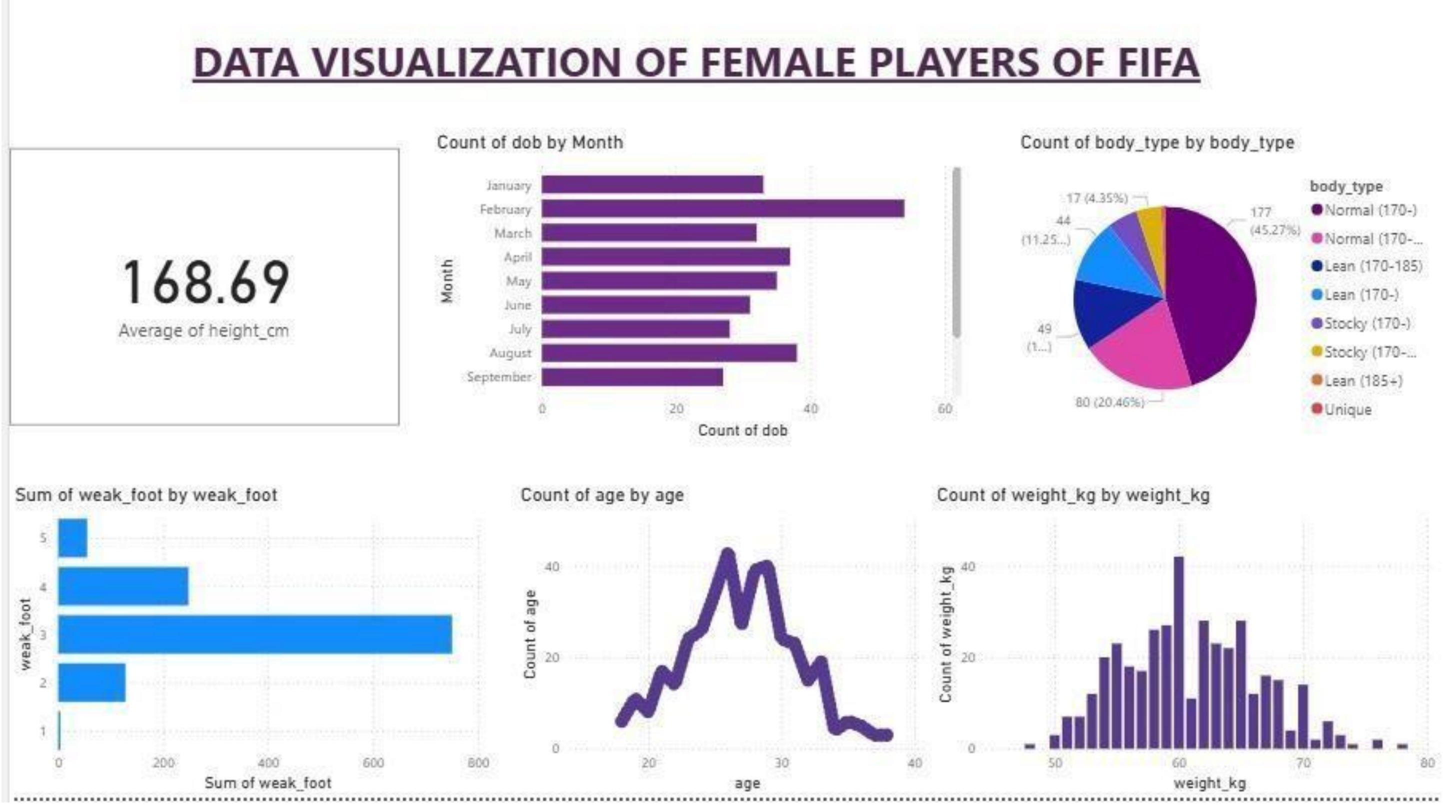
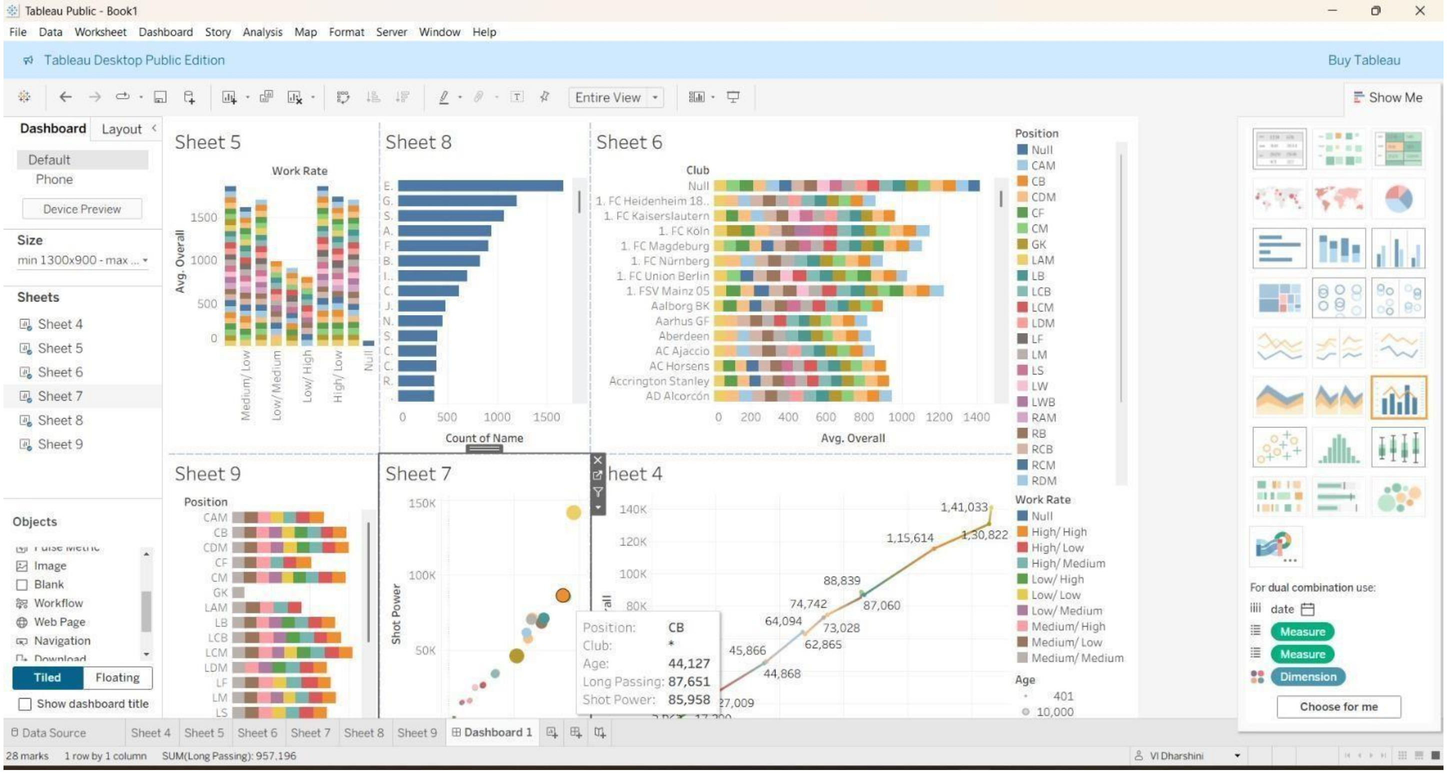


Tableau ouput:



Comparison Between Visualization Tools

Feature	Python	Power BI	Tableau
Customization	High	Medium	High
Interactivity	Moderate	Very High	Very High
Ease of Use	Medium	Easy	Easy
Suitable For	Analysts	Business Users	Analysts C Managers

Conclusion

- Python allows flexible and detailed control over visuals.
- Power BI and Tableau offer interactive and dynamic dashboards.
- Combining all three gives complete analytical and visual coverage.