```python
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr

# Load & read the dataset
from google.colab import drive
drive.mount("/content/gdrive")
df = pd.read_csv('/content/gdrive/My Drive/Colab
Notebooks/F1Drivers_Dataset.csv')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly
remount, call drive.mount("/content/gdrive", force_remount=True).
```

```python
# Data Exploration

# Print the shape (number of rows, number of columns)
print(df.shape)
# Check the first few rows of the dataset
print(df.head())
# Check summary statistics for numerical columns and round them out
print(df.describe().round())
# Check data types and missing values
print(df.info())
```

```
(868, 22)
              Driver     Nationality
Seasons  \
0        Carlo Abate           Italy                          [1962, 1963]

1   George Abecassis  United Kingdom                          [1951, 1952]

2      Kenny Acheson  United Kingdom                          [1983, 1985]

3  Andrea de Adamich           Italy  [1968, 1970, 1971, 1972, 1973]

4     Philippe Adams         Belgium                                [1994]


   Championships  Race_Entries  Race_Starts  Pole_Positions  Race_Wins
\
0            0.0           3.0          0.0             0.0        0.0

1            0.0           2.0          2.0             0.0        0.0

2            0.0          10.0          3.0             0.0        0.0

3            0.0          36.0         30.0             0.0        0.0
```

```
4              0.0            2.0           2.0                0.0        0.0


   Podiums  Fastest_Laps  ...  Championship Years  Decade Pole_Rate  \
0      0.0           0.0  ...                 NaN    1960       0.0
1      0.0           0.0  ...                 NaN    1950       0.0
2      0.0           0.0  ...                 NaN    1980       0.0
3      0.0           0.0  ...                 NaN    1970       0.0
4      0.0           0.0  ...                 NaN    1990       0.0

   Start_Rate  Win_Rate  Podium_Rate  FastLap_Rate
Points_Per_Entry  \
0    0.000000       0.0          0.0           0.0          0.000000

1    1.000000       0.0          0.0           0.0          0.000000

2    0.300000       0.0          0.0           0.0          0.000000

3    0.833333       0.0          0.0           0.0          0.166667

4    1.000000       0.0          0.0           0.0          0.000000


   Years_Active  Champion
0             2     False
1             2     False
2             2     False
3             5     False
4             1     False

[5 rows x 22 columns]
       Championships  Race_Entries  Race_Starts  Pole_Positions
Race_Wins  \
count          868.0         868.0        868.0           868.0
868.0
mean             0.0          30.0         28.0             1.0
1.0
std              1.0          54.0         53.0             6.0
6.0
min              0.0           1.0          0.0             0.0
0.0
25%              0.0           2.0          1.0             0.0
0.0
50%              0.0           7.0          5.0             0.0
0.0
75%              0.0          29.0         26.0             0.0
0.0
max              7.0         359.0        356.0           103.0
103.0
```

```
       Podiums  Fastest_Laps  Points  Decade  Pole_Rate  Start_Rate
Win_Rate  \
count    868.0         868.0   868.0   868.0      868.0       868.0
868.0
mean       4.0           1.0    56.0  1972.0        0.0         1.0
0.0
std       14.0           5.0   266.0    20.0        0.0         0.0
0.0
min        0.0           0.0     0.0  1950.0        0.0         0.0
0.0
25%        0.0           0.0     0.0  1960.0        0.0         1.0
0.0
50%        0.0           0.0     0.0  1970.0        0.0         1.0
0.0
75%        0.0           0.0     8.0  1982.0        0.0         1.0
0.0
max      191.0          77.0  4416.0  2020.0        1.0         1.0
0.0

       Podium_Rate  FastLap_Rate  Points_Per_Entry  Years_Active
count        868.0         868.0             868.0         868.0
mean           0.0           0.0               0.0           4.0
std            0.0           0.0               1.0           4.0
min            0.0           0.0               0.0           1.0
25%            0.0           0.0               0.0           1.0
50%            0.0           0.0               0.0           2.0
75%            0.0           0.0               0.0           5.0
max            1.0           0.0              14.0          19.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 868 entries, 0 to 867
Data columns (total 22 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Driver              868 non-null    object
 1   Nationality         868 non-null    object
 2   Seasons             868 non-null    object
 3   Championships       868 non-null    float64
 4   Race_Entries        868 non-null    float64
 5   Race_Starts         868 non-null    float64
 6   Pole_Positions      868 non-null    float64
 7   Race_Wins           868 non-null    float64
 8   Podiums             868 non-null    float64
 9   Fastest_Laps        868 non-null    float64
 10  Points              868 non-null    float64
 11  Active              868 non-null    bool
 12  Championship Years  34 non-null     object
 13  Decade              868 non-null    int64
 14  Pole_Rate           868 non-null    float64
 15  Start_Rate          868 non-null    float64
```

```
 16  Win_Rate            868 non-null    float64
 17  Podium_Rate         868 non-null    float64
 18  FastLap_Rate        868 non-null    float64
 19  Points_Per_Entry    868 non-null    float64
 20  Years_Active        868 non-null    int64
 21  Champion            868 non-null    bool
dtypes: bool(2), float64(14), int64(2), object(4)
memory usage: 137.4+ KB
None
```

```python
#Check null and duplicate values
print(df.isnull().sum())
print(df.duplicated().sum())
```

```
Driver                    0
Nationality               0
Seasons                   0
Championships             0
Race_Entries              0
Race_Starts               0
Pole_Positions            0
Race_Wins                 0
Podiums                   0
Fastest_Laps              0
Points                    0
Active                    0
Championship Years      834
Decade                    0
Pole_Rate                 0
Start_Rate                0
Win_Rate                  0
Podium_Rate               0
FastLap_Rate              0
Points_Per_Entry          0
Years_Active              0
Champion                  0
dtype: int64
0
```

```python
# Check data types
data_types = df.dtypes
print(data_types)
```

```
Driver                   object
Nationality              object
Seasons                  object
Championships           float64
Race_Entries            float64
Race_Starts             float64
Pole_Positions          float64
```

```
Race_Wins                   float64
Podiums                     float64
Fastest_Laps                float64
Points                      float64
Active                         bool
Championship Years           object
Decade                        int64
Pole_Rate                   float64
Start_Rate                  float64
Win_Rate                    float64
Podium_Rate                 float64
FastLap_Rate                float64
Points_Per_Entry            float64
Years_Active                  int64
Champion                       bool
dtype: object
```

Seasons data type is object, Extract years and convert to integers

Load the dataset with custom parsing for 'Season' column

```
def parse_seasons(seasons_str):
    # Remove square brackets and split by comma, then convert to
integers
    return [int(year.strip()) for year in
seasons_str.strip('[]').split(',')]

# Load the CSV file with custom parsing for 'Season' column
data = pd.read_csv ('/content/gdrive/My Drive/Colab
Notebooks/F1Drivers_Dataset.csv', converters={'Seasons':
parse_seasons})
```

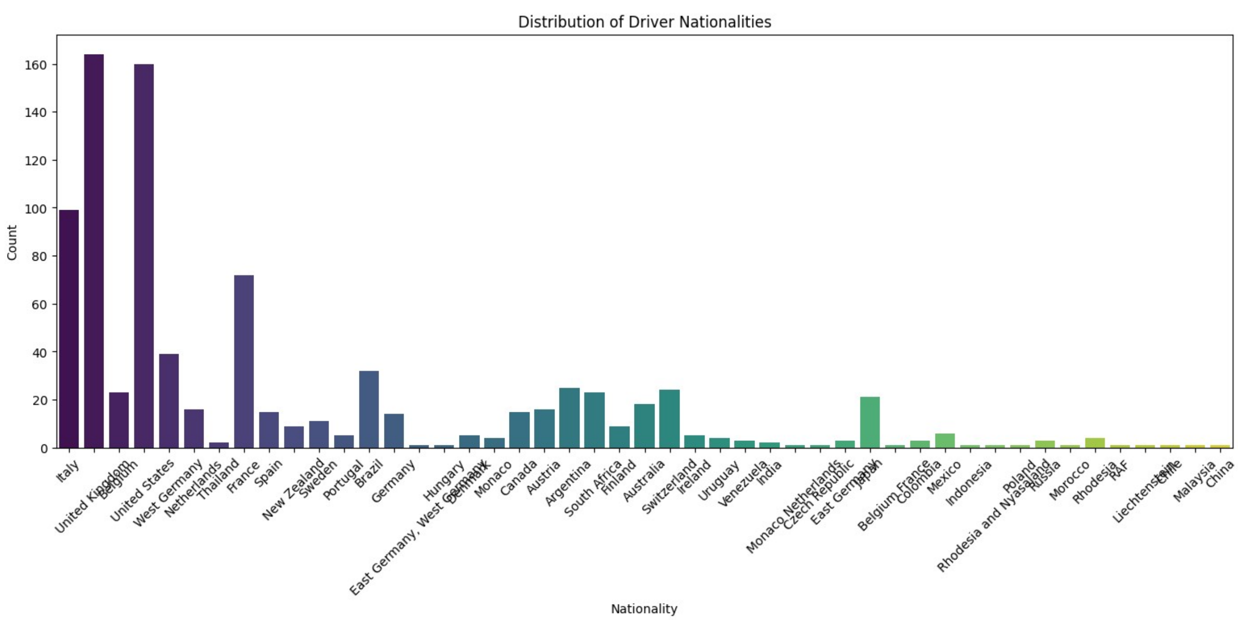
Now, the 'Seasons' column contains lists of integers instead of strings(objects)

```
print(data['Seasons'].head())

# Display the DataFrame
print(data)
```

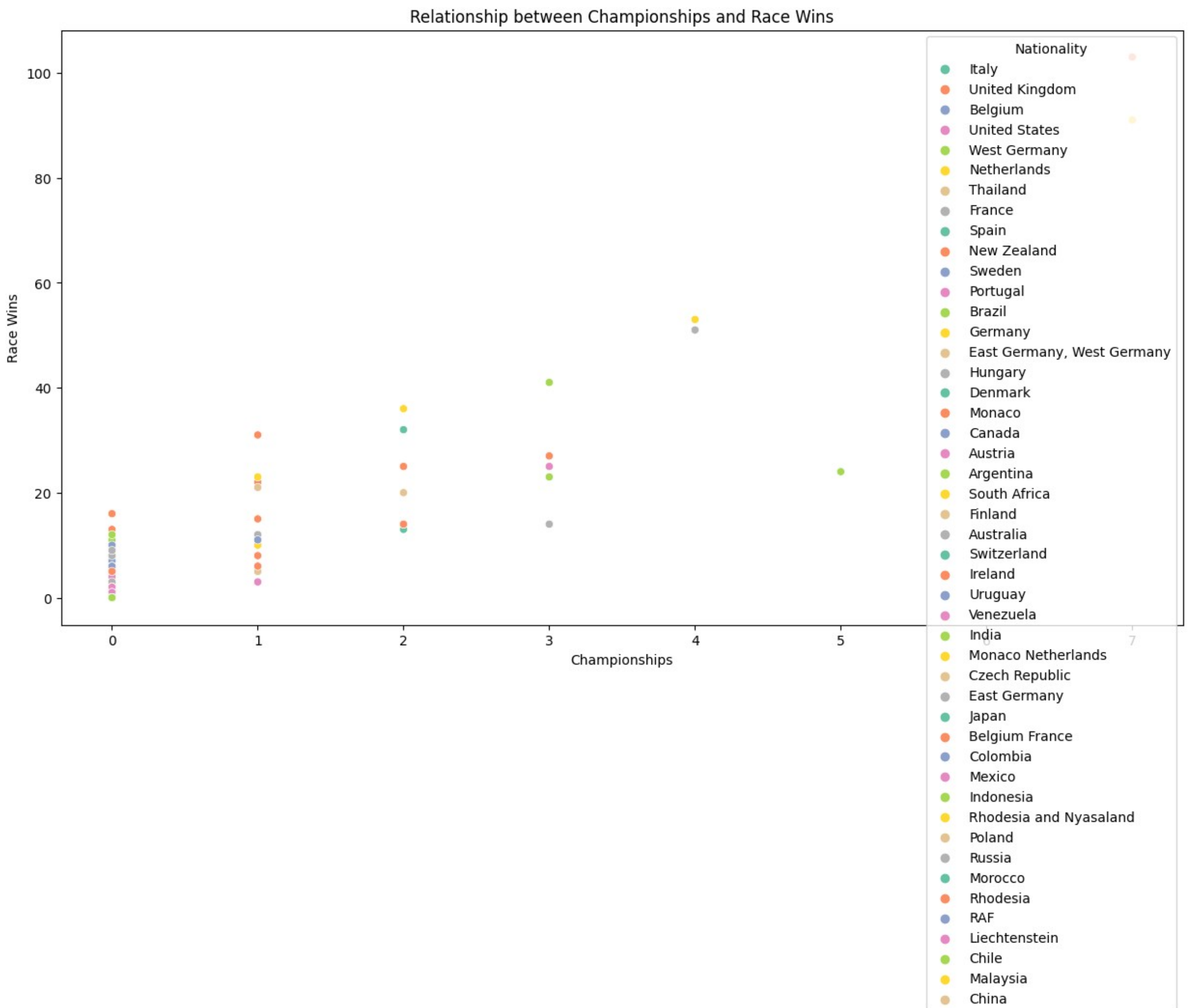
```
0                      [1962, 1963]
1                      [1951, 1952]
2                      [1983, 1985]
3    [1968, 1970, 1971, 1972, 1973]
4                            [1994]
Name: Seasons, dtype: object
                Driver     Nationality                         Seasons  
\
0          Carlo Abate           Italy                    [1962, 1963]   

1     George Abecassis  United Kingdom                    [1951, 1952]   

2        Kenny Acheson  United Kingdom                    [1983, 1985]   

3    Andrea de Adamich           Italy  [1968, 1970, 1971, 1972, 1973]   

4       Philippe Adams         Belgium                          [1994]   

..                 ...             ...                             ...   

863      Emilio Zapico           Spain                          [1976]   

864        Zhou Guanyu           China                          [2022]   

865      Ricardo Zonta          Brazil  [1999, 2000, 2001, 2004, 2005]   

866        Renzo Zorzi           Italy              [1975, 1976, 1977]   

867     Ricardo Zunino       Argentina              [1979, 1980, 1981]   


     Championships  Race_Entries  Race_Starts  Pole_Positions  
Race_Wins  \
0              0.0           3.0          0.0             0.0        
0.0   
1              0.0           2.0          2.0             0.0        
0.0   
2              0.0          10.0          3.0             0.0        
0.0   
3              0.0          36.0         30.0             0.0        
0.0   
4              0.0           2.0          2.0             0.0        
0.0   
..             ...           ...          ...             ...        .
..   
863            0.0           1.0          0.0             0.0        
0.0   
864            0.0          23.0         23.0             0.0        
0.0   
865            0.0          37.0         36.0             0.0        
0.0   
```

```
866            0.0          7.0          7.0              0.0
0.0
867            0.0         11.0         10.0              0.0
0.0

     Podiums  Fastest_Laps  ...  Championship Years  Decade Pole_Rate
\
0        0.0           0.0  ...                 NaN    1960       0.0

1        0.0           0.0  ...                 NaN    1950       0.0

2        0.0           0.0  ...                 NaN    1980       0.0

3        0.0           0.0  ...                 NaN    1970       0.0

4        0.0           0.0  ...                 NaN    1990       0.0

..       ...           ...  ...                 ...     ...       ...

863      0.0           0.0  ...                 NaN    1980       0.0

864      0.0           2.0  ...                 NaN    2020       0.0

865      0.0           0.0  ...                 NaN    2000       0.0

866      0.0           0.0  ...                 NaN    1980       0.0

867      0.0           0.0  ...                 NaN    1980       0.0


     Start_Rate  Win_Rate  Podium_Rate  FastLap_Rate  Points_Per_Entry
\
0      0.000000       0.0          0.0      0.000000          0.000000

1      1.000000       0.0          0.0      0.000000          0.000000

2      0.300000       0.0          0.0      0.000000          0.000000

3      0.833333       0.0          0.0      0.000000          0.166667

4      1.000000       0.0          0.0      0.000000          0.000000

..          ...       ...          ...           ...               ...

863    0.000000       0.0          0.0      0.000000          0.000000

864    1.000000       0.0          0.0      0.086957          0.260870

865    0.972973       0.0          0.0      0.000000          0.081081

866    1.000000       0.0          0.0      0.000000          0.142857
```

```
867    0.909091       0.0          0.0      0.000000          0.000000

     Years_Active  Champion
0               2     False
1               2     False
2               2     False
3               5     False
4               1     False
..            ...       ...
863             1     False
864             1     False
865             5     False
866             3     False
867             3     False

[868 rows x 22 columns]

# Preliminary Analysis

# Explore the distribution of driver nationalities
plt.figure(figsize=(17, 6))
sns.countplot(data=df, x='Nationality', palette='viridis')
plt.title('Distribution of Driver Nationalities')
plt.xlabel('Nationality')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



```python
# Relationship between Championships and Race Wins
plt.figure(figsize=(15, 8))
sns.scatterplot(x='Championships', y='Race_Wins', hue='Nationality',
data=df, palette='Set2')
plt.title('Relationship between Championships and Race Wins')
plt.xlabel('Championships')
plt.ylabel('Race Wins')
plt.legend(title='Nationality', loc='upper right')
plt.show()
```

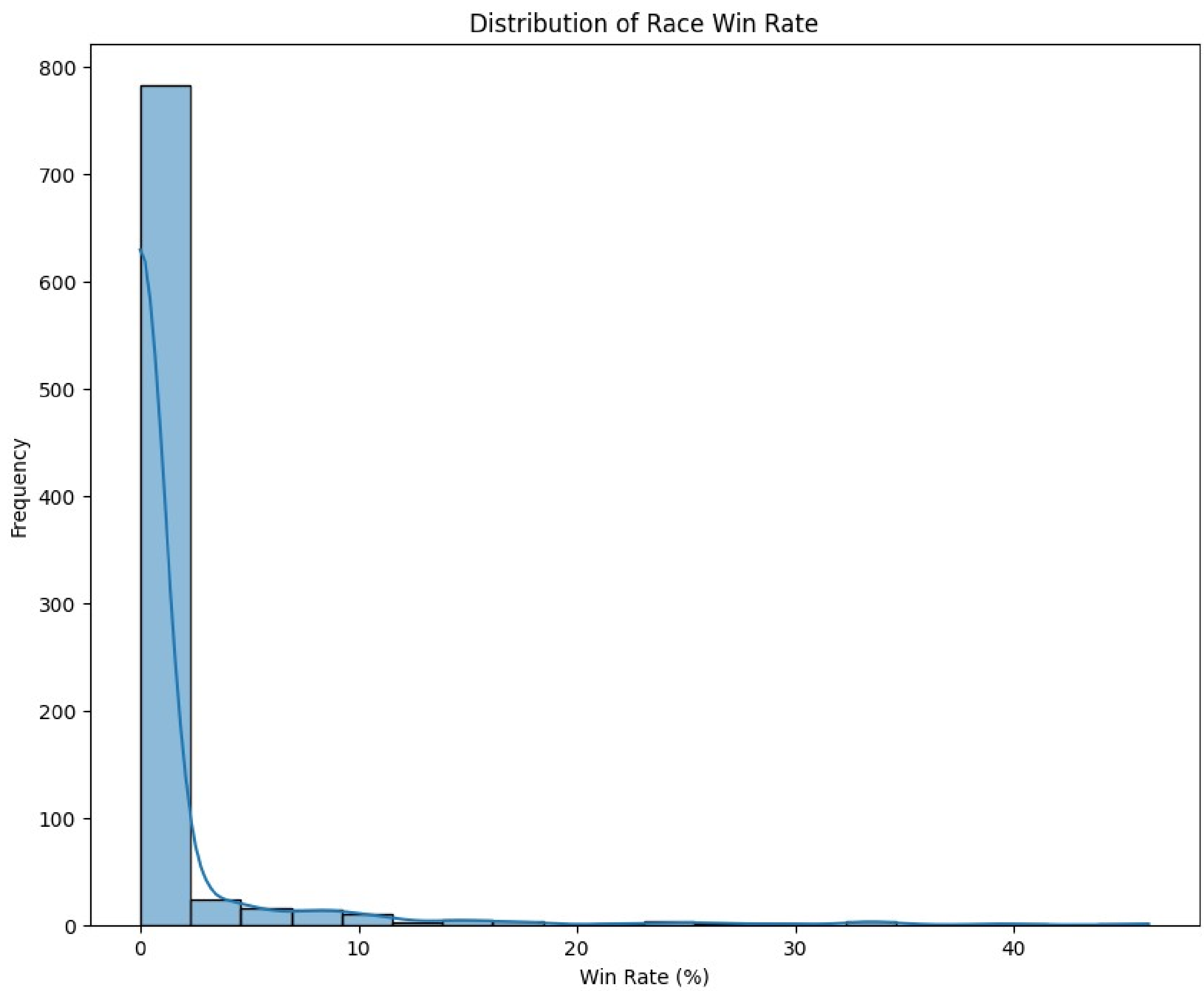



```python
# Further insight
# Calculate the average win rate for champions
avg_win_rate_champions = df[df['Champion'] == 1]['Win_Rate'].mean()
print(f'Average Win Rate for Champions: {avg_win_rate_champions:.2%}')
```

```
Average Win Rate for Champions: 15.55%
```

Average Win Rate for Champions: 15.55%

```
# Use Case 3: Analysis of Race Win Rate
data['Win_Rate'] = (data['Race_Wins'] / data['Race_Entries']) * 100
plt.figure(figsize=(10, 8))
sns.histplot(data['Win_Rate'], bins=20, kde=True)
plt.title("Distribution of Race Win Rate")
plt.xlabel("Win Rate (%)")
plt.ylabel("Frequency")
plt.show()
```



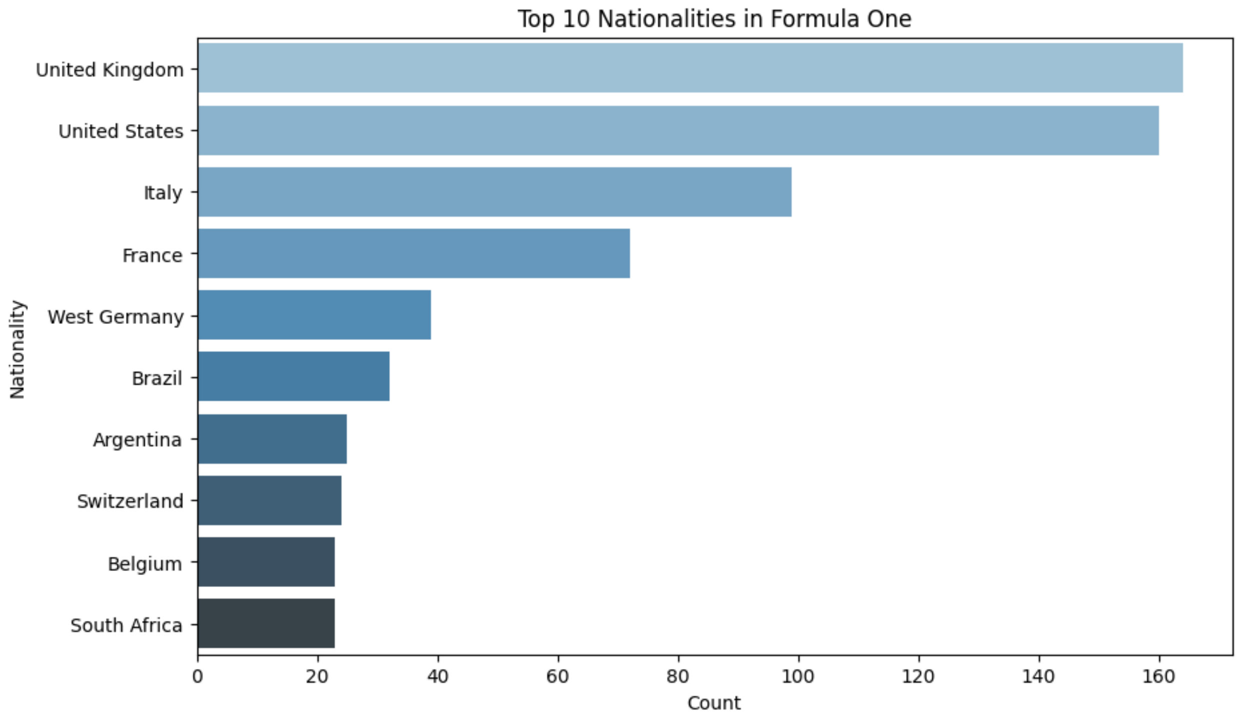
Deductions for Preliminary Analysis

1. Most drivers do not have championships, indicating that winning a championship is rare.

2. There is a positive correlation between championships and race wins.

3. The distribution of race win rates is right-skewed, with a few drivers having high win rates.

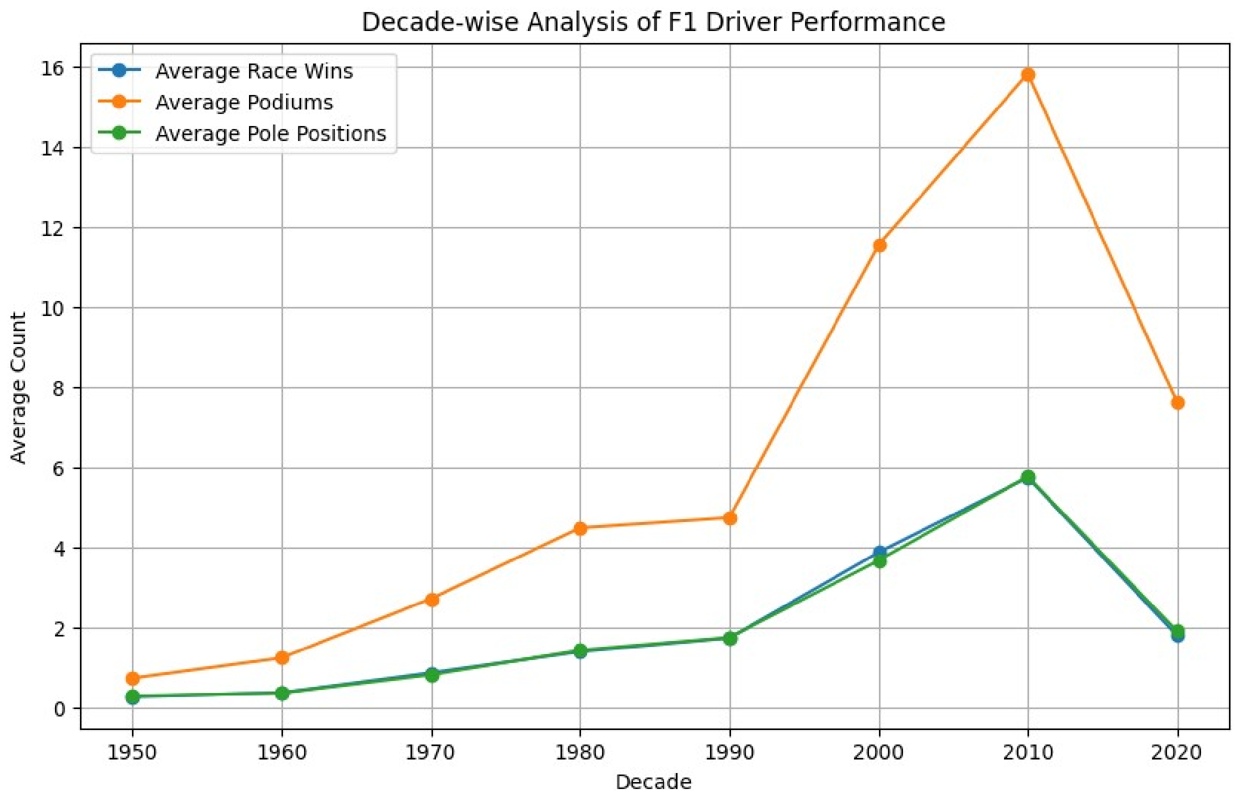
```
# Use Case 1: Nationality Analysis
#Nationality Analysis: Analyzing the distribution of the top 10
nationalities among Formula One drivers.

nationality_counts = data['Nationality'].value_counts().head(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=nationality_counts.values, y=nationality_counts.index,
palette='Blues_d')
plt.title("Top 10 Nationalities in Formula One")
plt.xlabel("Count")
plt.ylabel("Nationality")
plt.show()
```



UK and US nationalities are the most common- by a significant amount- among Formula One drivers in the dataset. This might indicate the popularity of the sport is highest in those two countries, along with the availability of infrastructure to train F1 drivers.

```
# Use Case 2: Decade-wise Analysis
# Decade-wise Analysis
#Decade-wise Analysis: Examining how race wins, pole positions, and
podiums have evolved over the decades.


# Calculate the average number of Race Wins, Podiums, and Pole
Positions for each decade
decade_stats = data.groupby('Decade').agg({
    'Race_Wins': 'mean',
    'Podiums': 'mean',
    'Pole_Positions': 'mean'
}).reset_index()
```

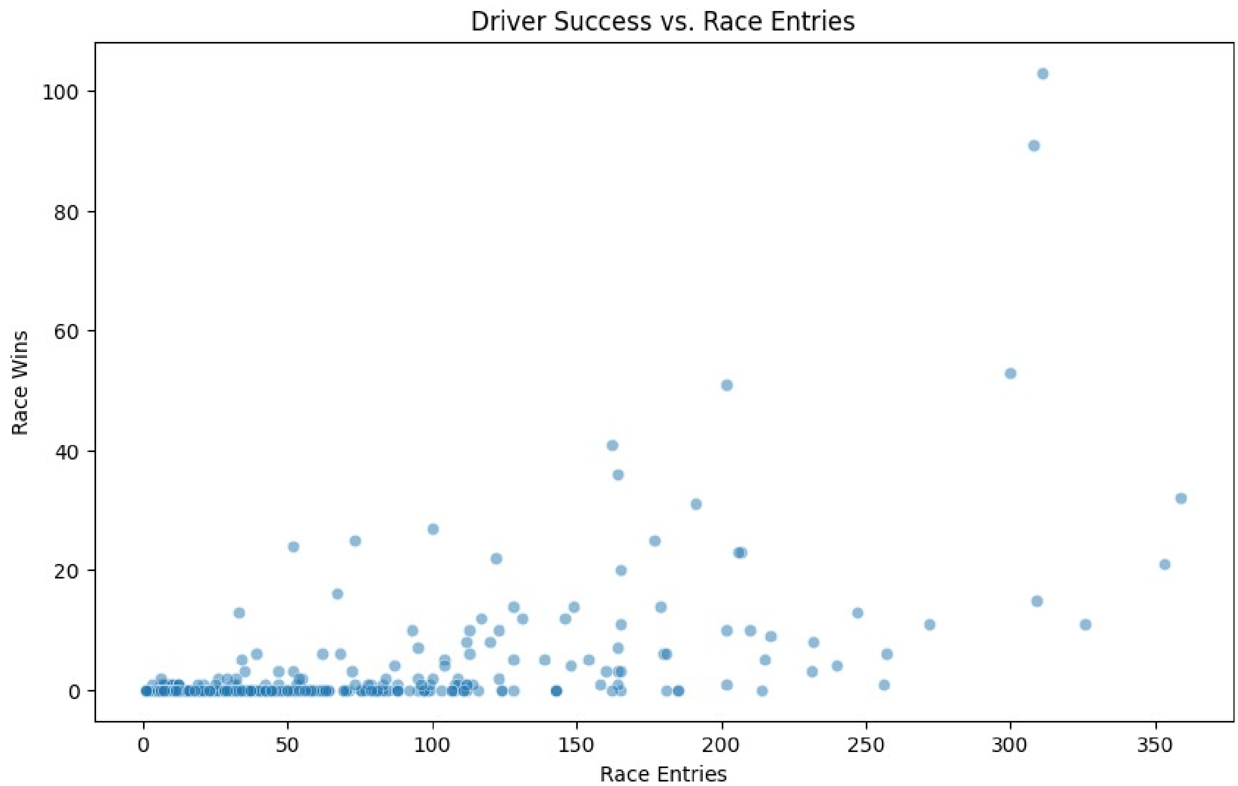
```
# Data Visualization
plt.figure(figsize=(10, 6))
plt.plot(decade_stats['Decade'], decade_stats['Race_Wins'],
marker='o', label='Average Race Wins')
plt.plot(decade_stats['Decade'], decade_stats['Podiums'], marker='o',
label='Average Podiums')
plt.plot(decade_stats['Decade'], decade_stats['Pole_Positions'],
marker='o', label='Average Pole Positions')
plt.xlabel('Decade')
plt.ylabel('Average Count')
plt.title('Decade-wise Analysis of F1 Driver Performance')
plt.legend()
plt.grid(True)
plt.show()
```



The analysis reveals a historical trend of continuous performance improvement among Formula 1 drivers from the 1950s to the early 2010s. This sustained growth in race wins, podiums, and pole positions reflects the cumulative effect of advancements in car technology, training regimens, and race strategies.

#Additionally,there is an observable decline in performance between the 2010s and 2020. This decline may be attributed to various factors, including regulations aimed at reducing car performance advantages, stricter engine and aerodynamic rules, and the emergence of dominant teams and drivers who temporarily reduced competitiveness across the field.

```
# Use Case 3: Driver Success vs. Race Entries
#Driver Success vs. Race Entries: Investigating whether there's a
clear relationship between the number of race entries and
championships won.

# Data Cleaning
data['Championships'] = data['Championships'].fillna(0).astype(int)
data['Win_Rate'] = (data['Race_Wins'] / data['Race_Entries']) * 100

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Race_Entries', y='Race_Wins', data=data, alpha=0.5)
plt.title("Driver Success vs. Race Entries")
plt.xlabel("Race Entries")
plt.ylabel("Race Wins")
plt.show()

# Calculate Pearson correlation between Race Entries and Race Wins
correlation, p_value = pearsonr(data['Race_Entries'],
data['Race_Wins'])
print(f"Pearson Correlation Coefficient: {correlation:.2f}")
```

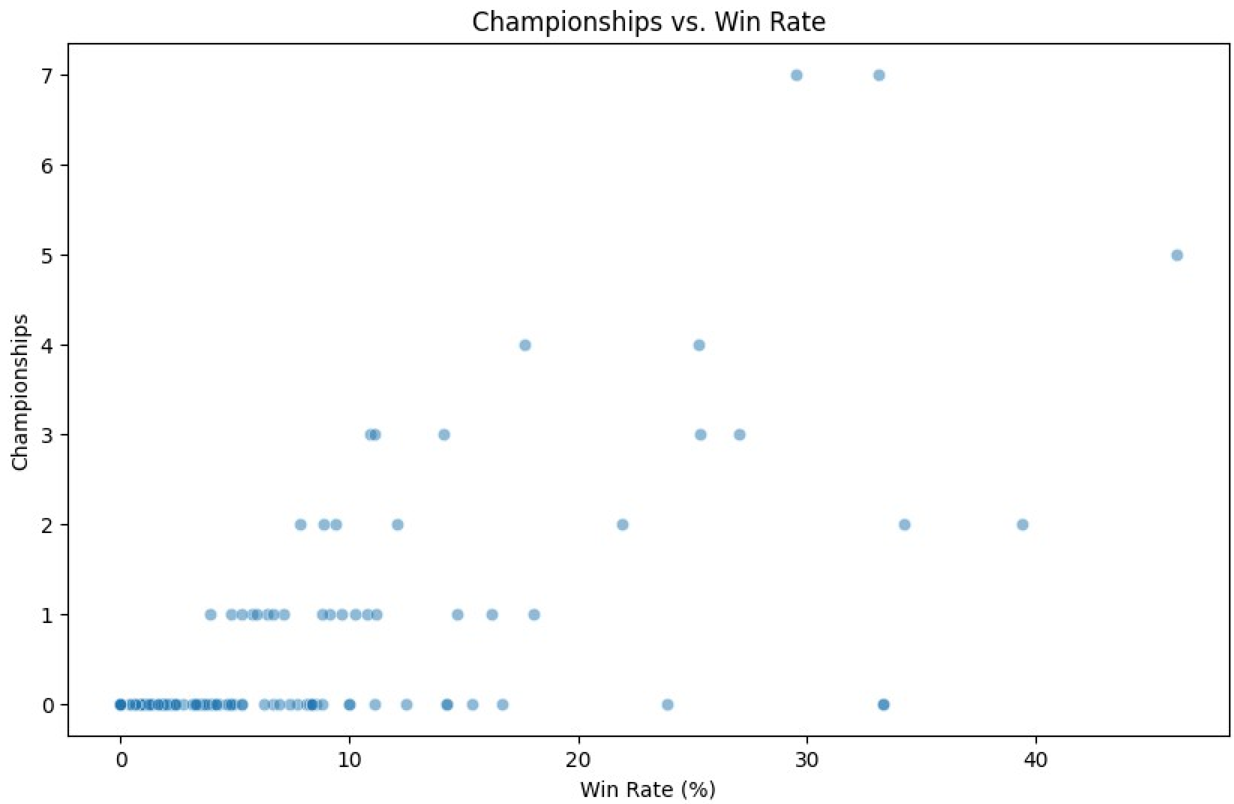



```
Pearson Correlation Coefficient: 0.59
```

#The correlation coefficient of 0.6 suggests that there is a moderate positive relationship between the number of race entries and the number of race wins.

As drivers participate in more races (higher race entries), they tend to achieve a higher number of race wins, on average.

```
# Use Case 4: Championship and Win Rate Analysis
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Win_Rate', y='Championships', data=data, alpha=0.5)
plt.title("Championships vs. Win Rate")
plt.xlabel("Win Rate (%)")
plt.ylabel("Championships")
plt.show()
```




```
# 4.1 - Calculate Pearson correlation between Win Rate and Championships
correlation, p_value = pearsonr(data['Win_Rate'], data['Championships'])
print(f"Pearson Correlation: {correlation:.2f}")
print(f"P-Value: {p_value:.2f}")

Pearson Correlation: 0.73
P-Value: 0.00
```

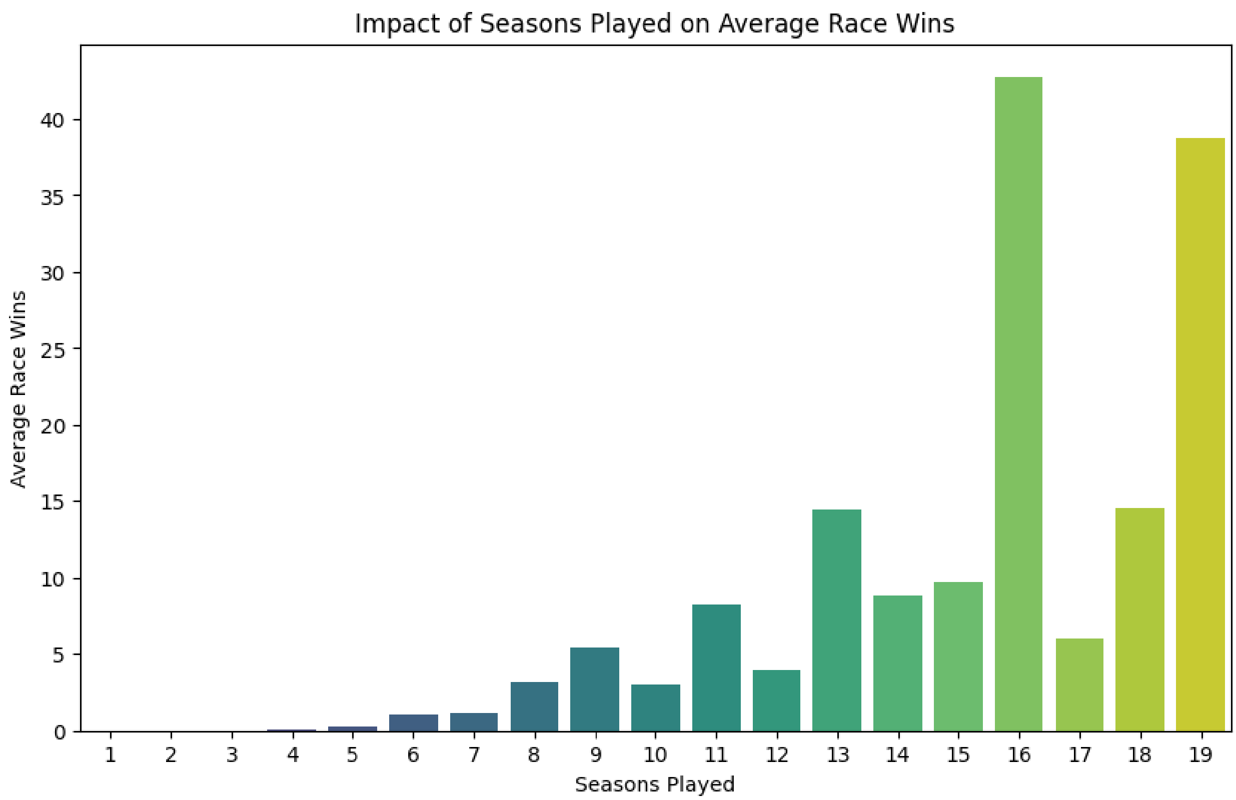
#In this case, the correlation coefficient is positive and relatively strong (0.73), which means there is a positive linear relationship between the two variables.

#Win Rate and Championships: The scatter plot shows a positive correlation between a driver's win rate and the number of championships they have won. This suggests that drivers with higher win rates are more likely to achieve championships. The Pearson correlation coefficient confirms this positive relationship.

#ie. Drivers with higher win rates are more likely to become champions, indicating the importance of consistent race performance in winning championships.

```
# More Data Cleaning
data['Championships'] = data['Championships'].fillna(0).astype(int)
data['Win_Rate'] = (data['Race_Wins'] / data['Race_Entries']) * 100
```

```python
# Use Case #5: Impact of Seasons Played on Performance
data['Seasons_Played'] = data['Seasons'].apply(len)

# Calculate average performance metrics based on the number of seasons played
seasons_performance = data.groupby('Seasons_Played').agg({
    'Race_Wins': 'mean',
    'Podiums': 'mean',
    'Championships': 'mean'
}).reset_index()

# Plotting the impact of seasons played on performance
plt.figure(figsize=(10, 6))
sns.barplot(x='Seasons_Played', y='Race_Wins', data=seasons_performance, palette='viridis')
plt.title("Impact of Seasons Played on Average Race Wins")
plt.xlabel("Seasons Played")
plt.ylabel("Average Race Wins")
plt.show()
```




```python
#5-1 # Calculate Pearson correlation coefficient
correlation, p_value = pearsonr(seasons_performance['Seasons_Played'], seasons_performance['Race_Wins'])
print(f"Pearson Correlation Coefficient: {correlation:.2f}")
```

```
Pearson Correlation Coefficient: 0.71
```

#The analysis shows a positive correlation between the number of seasons played and the average race wins for drivers.

Drivers who have competed in more seasons tend to have a higher average number of race wins.