P4DS Summative Assignment 2

Data Analysis Project

Formula 1 Drivers' Results Data Analysis

Student ID: 201912075

Project Plan

The Data (15 marks)

The data being used is gathered from Kaggle and includes information about Formula One which is a motorsport where drivers compete against each other in open-wheel single-seater formula racing cars. Every year (season) consists of a given number of races where the first few drivers (the numbers may differ from year to year) win points toward the World Drivers' Championship (the higher a driver is placed, the more points they get). At the end of every season, the driver with the most points is named the World Champion.

The dataset being explored in this project consists of 14 files with detailed information about every season from 1950 to 2022. However, only three of these files are of interest in order to achieve the objectives of the project.

The first file is **starting_grids.csv** which includes information about the starting grids (starting race positions of the drivers) for every Grand Prix (race). The dataset has 9 columns:

- Car - The name of the car manufacturer the driver raced for
- Detail - Contains **Starting-Grid** value among all entries, showing this dataset contains information about the starting grids
- Driver - The name of the driver
- DriverCode - The name abbreviation of the driver
- Grand Prix - The name of the Grand Prix
- No - The racing number of the driver
- Pos - The starting position of the driver for the given Grand Prix

- Time - The qualifying time of the driver (drivers' grid places are determined by a qualifying session before the race and any potential penalties)
- Year - The year the given Grand Prix happened

The second file is **race_details.csv** which consists of information about the race results for all Grands Prix. The dataset has 11 columns:

- Pos - The position the driver finished the given Grand Prix at
- No - The racing number of the driver
- Driver - The name of the driver
- Car - The name of the car manufacturer the driver raced for
- Laps - The total amount of laps completed by the driver in the given Grand Prix
- Time/Retired - The total race time for the winner and the time difference to the winner for the rest of the drivers in the given Grand Prix
- PTS - Points won by the driver from the given Grand Prix
- Year - The year the given Grand Prix happened
- Grand Prix - The name of the Grand Prix
- Detail - Contains **Race-Result** value among all entries, showing this dataset contains information about the race results
- DriverCode - The name abbreviation of the driver

The last file of interest is **driver_standings.csv**, containing information about the number of points earned by every driver and their corresponding positions in the Drivers' Championship at the end of every season (i.e. Driver Standings). The dataset has 7 columns:

- Pos - The position of the driver in the Driver Standings at the end of the given year
- Driver - The name of the driver
- Nationality - The abbreviation of the country the driver raced for
- Car - The name of the car manufacturer the driver raced for
- PTS - Total amount of points earned by the driver through the year
- DriverCode - The name abbreviation of the driver
- Year - The year the Driver Standings is for

The author of the dataset states that the information in the dataset is taken from the official website of Formula One. Based on this statement, the information should be the most accurate possible. However, it cannot be trusted that the information comes from the official website of the sport indeed and even if this is the case, errors are still possible when transferring the data. Additionally, the information on the official Formula One website might be incorrect or missing in places itself, even though it should be the most reliable

source. One quick glance at the **starting_grid.csv** and the **race_details.csv** files shows that the first one contains information about 5 races happened in 1950 while the second file contains information about 7 races happened in 1950, which shows that there is missing information in the datasets, indeed.

When looking at the dataset on the Kaggle website, its usability is marked as 9.71 by the system. The dataset received a score of 100% on all the three main characteristics - Completeness, Credibility, Compatibility. All these measurements show that the dataset is well-organised, well-documented and easy to understand and work with. Furthermore, short useful descriptions are included for every of the 14 files in the dataset and for the dataset itself, making the understanding of the purpose of the dataset very clear. Additionally, the features in every of the 14 datasets corresponds very well to their corresponding descriptions, which makes the work with them much easier. Overall, the dataset is well-structured and well-described, making it easy to work with.

Project Aim and Objectives (5 marks)

The beginning of Formula One dates back to 1950 and since then the sport has seen numerous Grand Prix winners and World Driver Champions. The aim of this project is to go through the rich sport's history and provide data analysis about some of the important aspects of the sport. Formula One has developed a lot during the years and is considered the pinnacle of the motorsports. The sport is very complicated and consists of many aspects. On one hand, the sport is one of the most technological sports in the world and teams consists of hundreds of people all contributing to constructing the fastest possible car. Usually the car performance is the biggest difference between the overall teams' and drivers' performances on track. On the other hand, however, the drivers' condition both physical and mental is also of huge importance and is often what makes the difference between teammates as they usually drive the same car in terms of performance. The motivation behind this data analysis project is to investigate whether there are some connections between drivers' race start positions and the following race results as well as between the number of race wins and the Drivers' Championship. Because the sport is so complex, the focus of this project is just on the pure race results and the points earned by the drivers toward the World Drivers' Championship. More precisely, the project takes into consideration the starting grid and the final race results for every Grand Prix as well as the final World Drivers' Championship for every season in order to achieve the project's objectives described below.

Specific Objective(s)

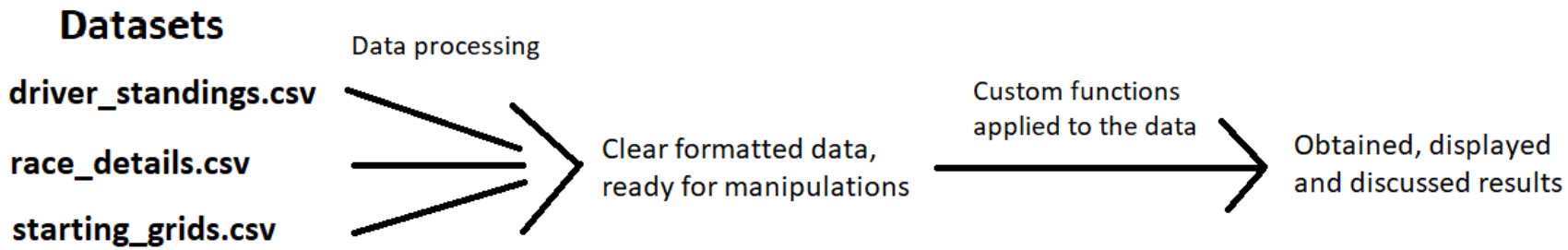
The current data analysis project aims to answer the following three questions (the project's objectives):

- **Objective 1:** Did one of the drivers starting a Grand Prix on the front row (1^{st} or 2^{nd} place) always win the race?
- **Objective 2:** Did the driver with the most wins during a season always win the World Drivers' Championship at the end of the year (the champion is the driver with the most points)?

- **Objective 3:** Did the number of races per season increase through the years? If so, did this lead to more race wins by the Champion or to having more race winners? Who is the driver won the most World Drivers' Championships and who is the driver won the most Grands Prix? Is this the same person?

System Design (5 marks)

Architecture



Three datasets are used in the current data analysis project, namely **driver_standings.csv**, **race_details.csv** and **starting_grids.csv**. Firstly, the data in all datasets goes through a data processing stage where data details are shown and discussed and unnecessary data columns are dropped as well as some data manipulations are made where needed. Then, it is noticed that some important information is missing in **starting_grids.csv** and the important part of it is displayed in a table format, which is made by the help of the **display_html_table** function. Then the results for **Objective 1** are calculated, displayed and discussed.

After **Objective 1**, the work on **Objective 2** is provided. There, the information about all the Champions and all the drivers with the most race wins for a season is gathered. Then, it is checked whether the Champion(s) for a given season is/are the driver or among the drivers with the most race wins for the same season. Results are shown and discussed.

Finally, the project includes the work on **Objective 3**. The data needed in this section is based on already gathered information and only transformations with it were required. Then, the data is plotted and discussed. At the end, conclusions are made about the last two questions of **Objective 3**.

Processing Modules and Algorithms

- Data cleaning by dropping unnecessary columns and duplicates removal

- Data value types transformation from *object* to *int* for *Pos* columns, representing a driver's position, as well as making sure there is only one space between driver's names and race's name
- Creation of **display_html_table** function which displays a table of values in a good table format
- Using various types of charts such as a bar chart, a pie chart and a line graph, to display results and discuss them

Program Code (25 marks)

Initially, all libraries being used are being imported

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
from IPython.display import HTML, Image
from collections import Counter
```

Reading the CSV files into Pandas DataFrames

In [2]:
```
df_Driver_Standings = pd.read_csv('driver_standings.csv')
df_Race_Details = pd.read_csv('race_details.csv')
df_Starting_Grids = pd.read_csv('starting_grids.csv')
```

Data Preprocessing for all DataFrames

Some of the columns in all of the three datasets are not needed for achieving the objectives of this project. Hence, these columns are dropped as this makes the datasets cleaner, containing important information only.

In [3]:
```
try:
    df_Driver_Standings = df_Driver_Standings.drop(["Nationality", "Car", "DriverCode"], axis=1)
except KeyError as e:
    print(e, end=". ")
    print("These columns have already been removed from the Driver Standings DataFrame.")

try:
    df_Race_Details = df_Race_Details.drop(["No", "Car", "Laps", "Time/Retired", "Detail", "DriverCode"], axis=1)
except KeyError as e:
    print(e, end=". ")
    print("These columns have already been removed from the Race Details DataFrame.")

try:
```

```
    df_Starting_Grids = df_Starting_Grids.drop(["Car", "Detail", "DriverCode", "No", "Time"], axis=1)
except KeyError as e:
    print(e, end=". ")
    print("These columns have already been removed from the Starting Grids DataFrame.")
```

In [4]:

```
# General information and descriptive statistics about the DataFrames
print("Driver Standings DataFrame:\n")
display(df_Driver_Standings.info())
display(df_Driver_Standings.describe())

print("\nRace Details DataFrame:\n")
display(df_Race_Details.info())
display(df_Race_Details.describe())

print("\nStarting Grids DataFrame:\n")
display(df_Starting_Grids.info())
display(df_Starting_Grids.describe())
```

```
Driver Standings DataFrame:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1618 entries, 0 to 1617
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Pos     1618 non-null   object
 1   Driver  1618 non-null   object
 2   PTS     1618 non-null   float64
 3   Year    1618 non-null   int64
dtypes: float64(1), int64(1), object(2)
memory usage: 50.7+ KB
None
```

	PTS	Year
count	1618.000000	1618.000000
mean	29.898331	1986.158220
std	58.039108	21.307326
min	0.000000	1950.000000
25%	3.000000	1967.000000
50%	9.000000	1986.000000

	PTS	Year
75%	30.375000	2005.000000
max	454.000000	2022.000000

```
Race Details DataFrame:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23978 entries, 0 to 23977
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Pos         23978 non-null  object
 1   Driver      23978 non-null  object
 2   PTS         23978 non-null  float64
 3   Year        23978 non-null  int64
 4   Grand Prix  23978 non-null  object
dtypes: float64(1), int64(1), object(3)
memory usage: 936.8+ KB
None
```

	PTS	Year
count	23978.000000	23978.000000
mean	2.026716	1991.014096
std	4.300269	19.609931
min	0.000000	1950.000000
25%	0.000000	1976.000000
50%	0.000000	1992.000000
75%	2.000000	2008.000000
max	50.000000	2022.000000

```
Starting Grids DataFrame:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22529 entries, 0 to 22528
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Driver      22529 non-null  object
 1   Grand Prix  22529 non-null  object
 2   Pos         22529 non-null  int64
```

```
 3   Year        22529 non-null  int64
dtypes: int64(2), object(2)
memory usage: 704.2+ KB
None
```

	Pos	Year
count	22529.000000	22529.000000
mean	11.867682	1993.403702
std	6.768206	17.674670
min	1.000000	1950.000000
25%	6.000000	1979.000000
50%	12.000000	1994.000000
75%	17.000000	2009.000000
max	33.000000	2022.000000

In [5]:

```
# Check for any duplicates

print("Driver Standings Duplicates:")
print(df_Driver_Standings.duplicated().value_counts())

print("\nRace Details Duplicates:")
print(df_Race_Details.duplicated().value_counts())

print("\nStarting Grids Duplicates:")
print(df_Starting_Grids.duplicated().value_counts())
```

```
Driver Standings Duplicates:
False    1618
dtype: int64

Race Details Duplicates:
False    23958
True        20
dtype: int64

Starting Grids Duplicates:
False    22529
dtype: int64
```

From the observations above, there are 20 duplicated entries in the Race Details DataFrame. The duplicated entries represent exactly one race, so they are not needed and are dropped

In [6]:
```
# Check which are the duplicated values
display(df_Race_Details[df_Race_Details.duplicated()==True])

# Check the number of entries before removing duplicates
print("Number of entries before removing duplicates: {}".format(df_Race_Details.shape[0]))

df_Race_Details.drop_duplicates(inplace = True)

# Check the number of entries to confirm that the duplicates were removed
print("Number of entries after removing duplicates: {}".format(df_Race_Details.shape[0]))
```

	Pos	Driver	PTS	Year	Grand Prix
23918	1	Max Verstappen	25.0	2022	Hungary
23919	2	Lewis Hamilton	19.0	2022	Hungary
23920	3	George Russell	15.0	2022	Hungary
23921	4	Carlos Sainz	12.0	2022	Hungary
23922	5	Sergio Perez	10.0	2022	Hungary
23923	6	Charles Leclerc	8.0	2022	Hungary
23924	7	Lando Norris	6.0	2022	Hungary
23925	8	Fernando Alonso	4.0	2022	Hungary
23926	9	Esteban Ocon	2.0	2022	Hungary
23927	10	Sebastian Vettel	1.0	2022	Hungary
23928	11	Lance Stroll	0.0	2022	Hungary
23929	12	Pierre Gasly	0.0	2022	Hungary
23930	13	Zhou Guanyu	0.0	2022	Hungary
23931	14	Mick Schumacher	0.0	2022	Hungary
23932	15	Daniel Ricciardo	0.0	2022	Hungary
23933	16	Kevin Magnussen	0.0	2022	Hungary

	Pos	Driver	PTS	Year	Grand Prix
23934	17	Alexander Albon	0.0	2022	Hungary
23935	18	Nicholas Latifi	0.0	2022	Hungary
23936	19	Yuki Tsunoda	0.0	2022	Hungary
23937	20	Valtteri Bottas	0.0	2022	Hungary

```
Number of entries before removing duplicates: 23978
Number of entries after removing duplicates: 23958
```

Note: The number of entries in both Starting_Grids and Race_Details DataFrames should be the same as the number of drivers starting a race should be the same as the number of drivers having some type of classification after the race.

In [7]:
```
print("Number of entries in Starting_Grids DataFrame: {}".format(df_Starting_Grids.shape[0]))
print("Number of entries in Race_Details DataFrame: {}".format(df_Race_Details.shape[0]))
```

```
Number of entries in Starting_Grids DataFrame: 22529
Number of entries in Race_Details DataFrame: 23958
```

From the check above, both numbers are not the same which indicates that there is missing information for some races (as mentioned earlier), potentially about the drivers' starting positions as the Starting_Grids DataFrame has less entries than the Race_Details DataFrame. However, Race_Details DataFrame may also have missing information. Detailed analysis of this situation follows later in this project.

In [8]:
```
# Drivers' positions need to be of a numerical type
df_Driver_Standings['Pos'] = pd.to_numeric(df_Driver_Standings['Pos'], errors='coerce')
df_Race_Details['Pos'] = pd.to_numeric(df_Race_Details['Pos'], errors='coerce')
```

In [9]:
```
# Making sure string data is as clear as possible and there is only one space between
# driver's names as well as between Grand Prix's names.
# Otherwise, the results might be incorrect. However, typos in the data are still possible.

df_Driver_Standings['Driver'] = df_Driver_Standings['Driver'].apply(lambda x: " ".join(x.split()))

df_Race_Details['Driver'] = df_Race_Details['Driver'].apply(lambda x: " ".join(x.split()))
df_Race_Details['Grand Prix'] = df_Race_Details['Grand Prix'].apply(lambda x: " ".join(x.split()))

df_Starting_Grids['Driver'] = df_Starting_Grids['Driver'].apply(lambda x: " ".join(x.split()))
df_Starting_Grids['Grand Prix'] = df_Starting_Grids['Grand Prix'].apply(lambda x: " ".join(x.split()))
```

Drivers' might not be assigned with a specific position and they may also be classified as NC (not classified) or DQ (disqualified), for example. For the purposes of this project, these values are of no interest and are marked as -1 for simplicity (after being marked as NaNs).

In [10]:
```
df_Driver_Standings['Pos'].fillna(-1, inplace = True)
df_Driver_Standings['Pos'] = df_Driver_Standings['Pos'].astype(int)

df_Race_Details['Pos'].fillna(-1, inplace = True)
df_Race_Details['Pos'] = df_Race_Details['Pos'].astype(int)
```

In [11]:
```
# Check for any missing values

print("Driver Standings Missing Values:")
print(df_Driver_Standings.isnull().sum())

print("\nRace Details Missing Values:")
print(df_Race_Details.isnull().sum())

print("\nStarting Grids Missing Values:")
print(df_Starting_Grids.isnull().sum())
```

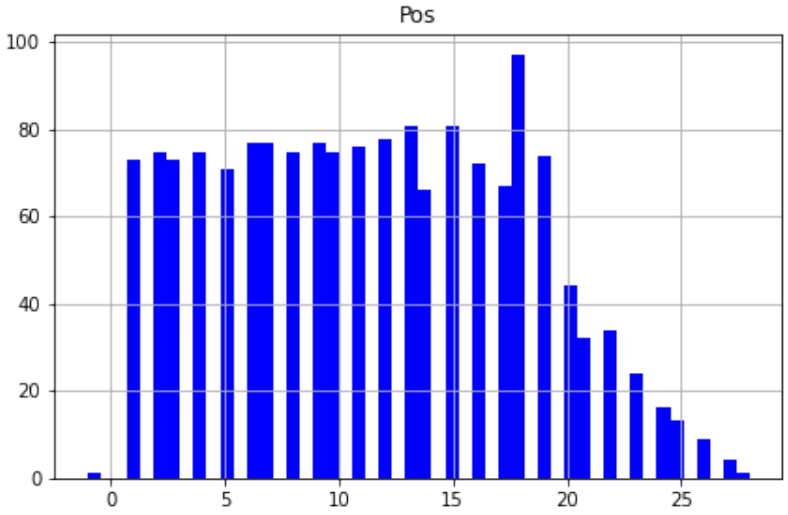
```
Driver Standings Missing Values:
Pos       0
Driver    0
PTS       0
Year      0
dtype: int64

Race Details Missing Values:
Pos           0
Driver        0
PTS           0
Year          0
Grand Prix    0
dtype: int64

Starting Grids Missing Values:
Driver        0
Grand Prix    0
Pos           0
Year          0
dtype: int64
```

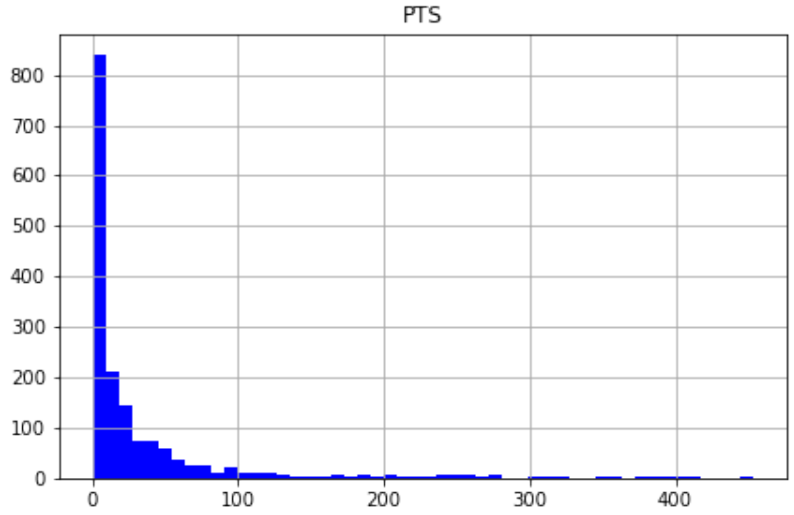
The check for missing values (NA values) shows that there are no such values

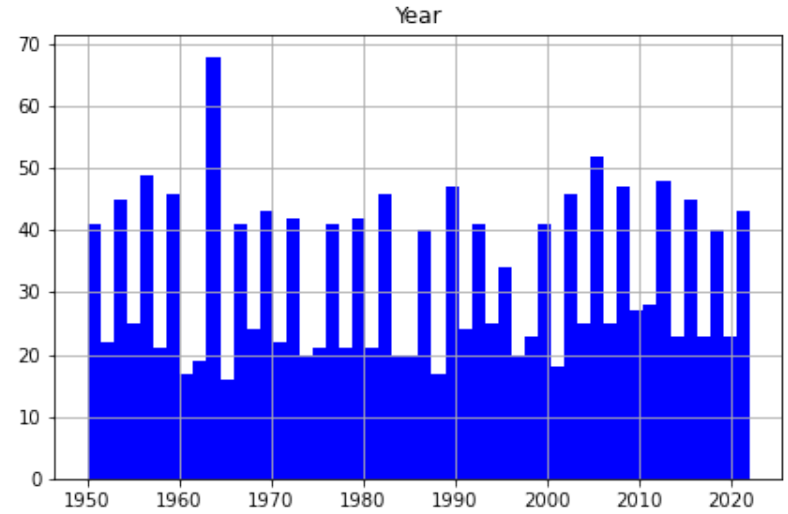
Univariate analysis on data distrbution

In [12]:
```
print("Driver Standings numerical data distribution")
df_Driver_Standings.hist(bins=50, figsize=(16,10),color='b')
plt.show()
```

```
Driver Standings numerical data distribution
```





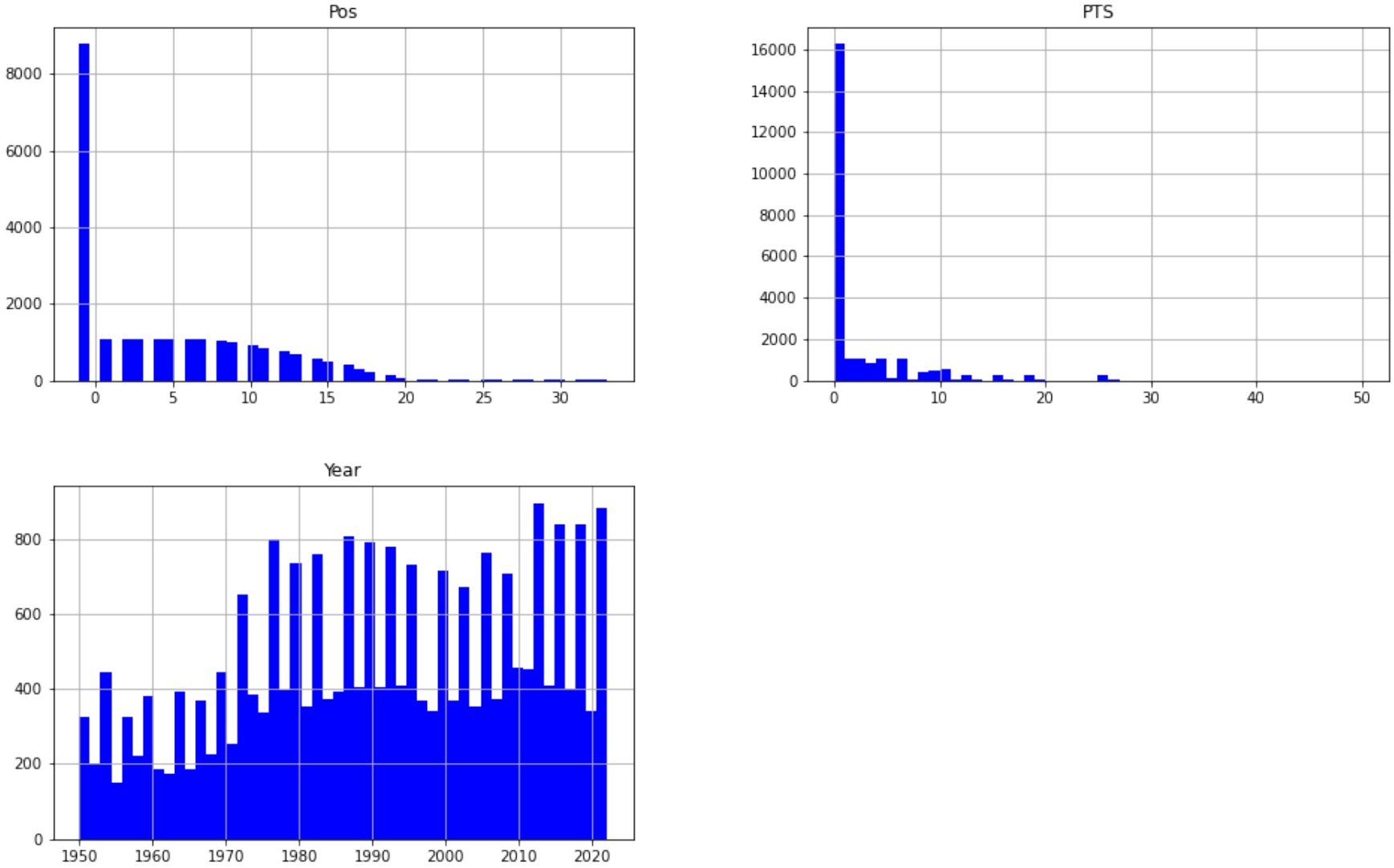


From the data above, the Pos attribute is relatively equally distributed until values 17/18, possibly showing that there were more rarely driver classifications after those positions. The PTS attribute distribution is skewed toward smaller values, possibly meaning that the

majority of drivers scored very small amount of points in a season. The last attribute distribution - the year one, is relatively equal with a value around 1963/1965 a lot higher than the others, possibly meaning more drivers participated in that time.

In [13]:
```
print("Race Details numerical data distribution")
df_Race_Details.hist(bins=50, figsize=(16,10),color='b')
plt.show()
```

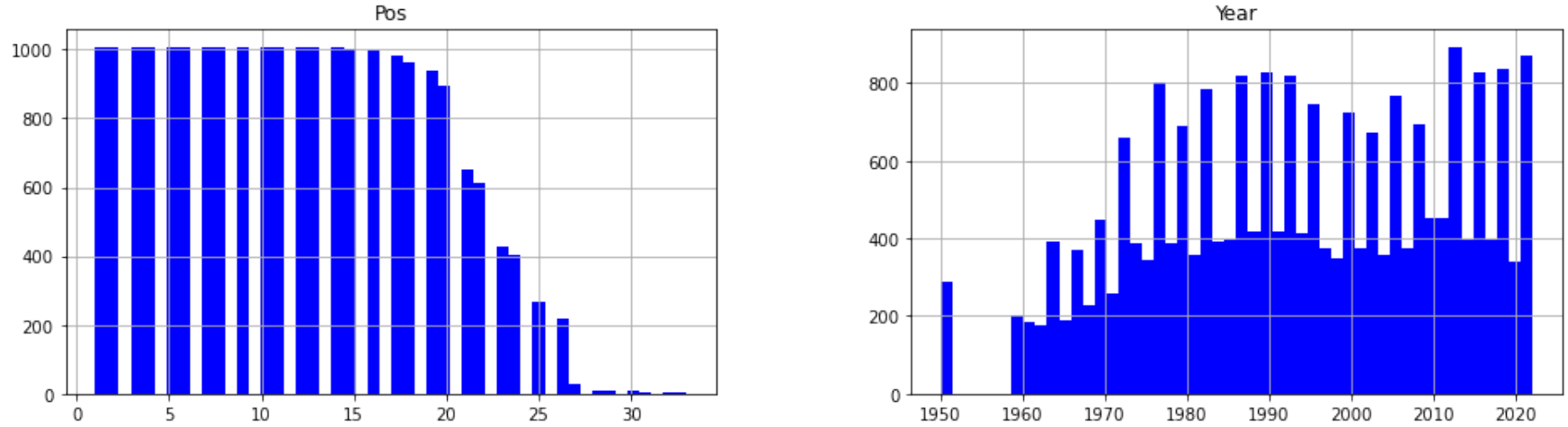
```
Race Details numerical data distribution
```



From the data above, the Pos attribute is relatively equally distributed until values around 10 and decreases afterwards, with the exception of the negative value (-1). This information possibly means that, the majority of the races were finished by at least 10 drivers as well as that possibly the majority of the drivers were not classified a position. The PTS attribute distribution has a peak at 0, possibly meaning the majority of the drivers received 0 points from a race. The Year attribute distribution increases with the time, possibly meaning the number of races in a season increased through the years.

In [14]:
```
print("Starting Grids numerical data distribution")
df_Starting_Grids.hist(bins=50, figsize=(16,4),color='b')
plt.show()
```

```
Starting Grids numerical data distribution
```



The Pos attribute distribution is relatively equal between 1 and 20 and drops down afterwards, possibly showing that at least 20 drivers participated in a race. The Year attribute distribution increases with the time, meaning the number of starting grids increased with the time, possibly because the number of races increased. Interestingly, there is a gap in the Year attribute's data roughly between 1950 and 1960. This means there is no information in the data about the starting grids during this period.

Objective 1: Did one of the drivers starting a Grand Prix on the front row (1^{st} or 2^{nd} place) always win the race?

In order to answer the question asked in **Objective 1**, the Starting Grids and the Race Details DataFrames are needed. Firstly, it must be ensured that there is information about the starting grid position of the winner for every Grand Prix. This is where the problem about the inequality between the entries in the Starting Grids and the Race Details DataFrames detected earlier is tackled.

In [15]:
```
# Set with (race, year) tuples for all races with information about the starting grid
all_races_with_grid = set((row_data['Grand Prix'], row_data['Year']) for row_name, row_data in df_Starting_Grids.iterrows
```

```
all_races_with_grid = sorted(all_races_with_grid, key=lambda x:x[1])

# Set with (race, year) tuples for all races with information about the final race results
all_races_with_results = set((row_data['Grand Prix'], row_data['Year']) for row_name, row_data in df_Race_Details.iterrow
all_races_with_results = sorted(all_races_with_results, key=lambda x:x[1])

print("Total races with grid information: " + str(len(all_races_with_grid)))
print("Total races with final results information: " + str(len(all_races_with_results)))
```

```
Total races with grid information: 1008
Total races with final results information: 1079
```

From the information above, there are at least 71 Grands Prix where there is information about the final race results but there is no information about the corresponding race starting grid. However, there is no guarantee that for every race starting grid there are corresponding final race results in the data. This is what the following code checks.

In [16]:

```
missing_race_results = set((race, year) for (race, year) in all_races_with_grid if (race, year) not in all_races_with_res
print(f"There are {len(missing_race_results)} races where there is information about the starting grid but there is no in

missing_grid = set((race, year) for (race, year) in all_races_with_results if (race, year) not in all_races_with_grid)
print(f"There are {len(missing_grid)} races where there is information about the final race results but there is no infor
```

```
There are 0 races where there is information about the starting grid but there is no information about the final race res
ults

There are 71 races where there is information about the final race results but there is no information about the starting
grid
```

From the test above, there are not any races with missing final results but with information about the corresponding starting grid information. There are exactly 71 races with information about the final race results but with no information about the corresponding race starting grid. These races are displayed below.

In [17]:

```
# A help function to display a table from a set in a nice HTML format
def html_table_from_set(caption, th_values, set_values, caption_style = "", th_style = "", td_style = ""):

    html_string = "<table>\n"
    html_string += "<caption {}>".format(caption_style)
    html_string += caption
    html_string += "</caption>"
    html_string += "<tr>\n"

    for th_value in th_values:
        html_string += "<th {}>".format(th_style) + th_value +"</th>\n"
```

```
    html_string += "</tr>\n"

    for values in set_values:
        html_string += "<tr>"
        for value in values:

            if isinstance(value, tuple) or isinstance(value, list) or isinstance(value, set):
                cell_value = ""
                for v in value:
                    cell_value += str(v) + ", "
                html_string += "<td {}>".format(td_style) + cell_value[:-2] + "</td>"

            else:
                html_string += "<td {}>".format(td_style) + str(value) + "</td>"

        html_string += "<tr>\n"
    html_string += "</table>\n"

    return html_string

# Function to display a table in a nice HTML format
def display_html_table(caption, header_values, cell_values):

    caption_style = ('style="color:#010000;' +
                     'text-align:center;' +
                     'font-size:1.5em;' +
                     'font-weight:bold;"')

    header_style = ('style="background-color:#ff582f;' +
                    'font-size:1.25em;' +
                    'text-align:center;' +
                    'border:1px solid black;"')

    cell_style = ( 'style="background-color:#fbad9a;' +
                   'font-weight:bold;' +
                   'text-align:center;' +
                   'border:1px solid black;"' )

    html_table = html_table_from_set(caption, header_values, cell_values, caption_style, header_style, cell_style)
    display( HTML("<center>" + html_table + "</center>") )


table_caption = "Races without starting grid information"
table_column_names = ["Race", "Year"]
```

```
# Sort the races with missing grid information by the year they happened,
# even though they may not have happened in the same order for the given year
missing_grid = sorted(missing_grid, key=lambda x:x[1])

display_html_table(table_caption, table_column_names, missing_grid)
```

Races without starting grid information

Race	Year
Great Britain	1950
Switzerland	1950
Belgium	1952
Switzerland	1952
Great Britain	1952
Indianapolis 500	1952
Italy	1952
Netherlands	1952
Germany	1952
France	1952
Great Britain	1953
Indianapolis 500	1953
Italy	1953
Netherlands	1953
Germany	1953
France	1953
Belgium	1953
Argentina	1953

Switzerland	**1953**
Germany	**1954**
France	**1954**
Belgium	**1954**
Argentina	**1954**
Spain	**1954**
Switzerland	**1954**
Great Britain	**1954**
Indianapolis 500	**1954**
Italy	**1954**
Belgium	**1955**
Argentina	**1955**
Great Britain	**1955**
Indianapolis 500	**1955**
Italy	**1955**
Monaco	**1955**
Netherlands	**1955**
Great Britain	**1956**
Indianapolis 500	**1956**
Italy	**1956**
Monaco	**1956**
Germany	**1956**
France	**1956**
Belgium	**1956**
Argentina	**1956**

Germany	**1957**
Pescara	**1957**
France	**1957**
Argentina	**1957**
Great Britain	**1957**
Indianapolis 500	**1957**
Monaco	**1957**
Italy	**1957**
Belgium	**1958**
Argentina	**1958**
Portugal	**1958**
Great Britain	**1958**
Italy	**1958**
Indianapolis 500	**1958**
Morocco	**1958**
Monaco	**1958**
Netherlands	**1958**
Germany	**1958**
France	**1958**
Italy	**1959**
Indianapolis 500	**1959**
Monaco	**1959**
Great Britain	**1959**
Netherlands	**1959**
United States	**1959**

Germany	**1959**
France	**1959**
Portugal	**1959**

All races with information about their corresponding starting grid are used to do the analysis for **Objective 1**

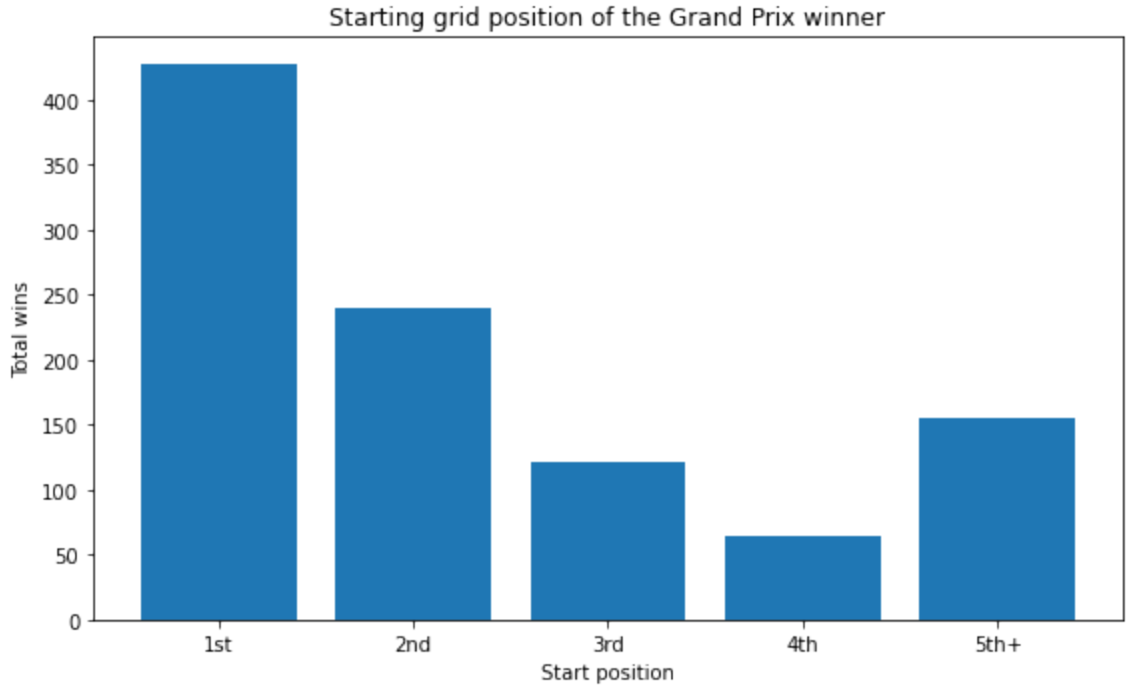
In [18]:
```
# Winners of all races with starting grid information
winners_with_grid = set((row_data['Driver'], row_data['Grand Prix'], row_data['Year'])
                        for row_name, row_data in df_Race_Details.iterrows()
                        if (row_data['Grand Prix'], row_data['Year']) in all_races_with_grid
                        and row_data['Pos'] == 1)

# The starting grid position of all winners
winners_grid_pos = [row_data['Pos'] for row_name, row_data in df_Starting_Grids.iterrows()
                    if (row_data['Driver'], row_data['Grand Prix'], row_data['Year']) in winners_with_grid]

print(f"Total number of races with grid information: {len(all_races_with_grid)}")
print(f"Total number of winners in races with grid information: {len(winners_with_grid)}")
```

```
Total number of races with grid information: 1008
Total number of winners in races with grid information: 1009
```

Interesting fact, there are 1009 winners in the dataset from 1008 races with information about their starting grid. This is like that because there were three races in the Formula 1 history which were won by two drivers sharing a car. There is information in this project's dataset about only one of these races (France 1951), which explains the difference between the aforementioned numbers.

In [19]:
```
# Maximum number of drivers competed in a season
max_num_drivers = df_Driver_Standings['Pos'].max()

start_pos_for_win = [0 for _ in range(max_num_drivers)]

for pos in winners_grid_pos:
    start_pos_for_win[pos-1] += 1


fig, ax = plt.subplots(figsize=(8,5))

x_values = [i for i in range(1, 6)]
# From 1st to 5th+
after_fourth = sum(start_pos_for_win[4:])
y_values = start_pos_for_win[:4] + [after_fourth]
```

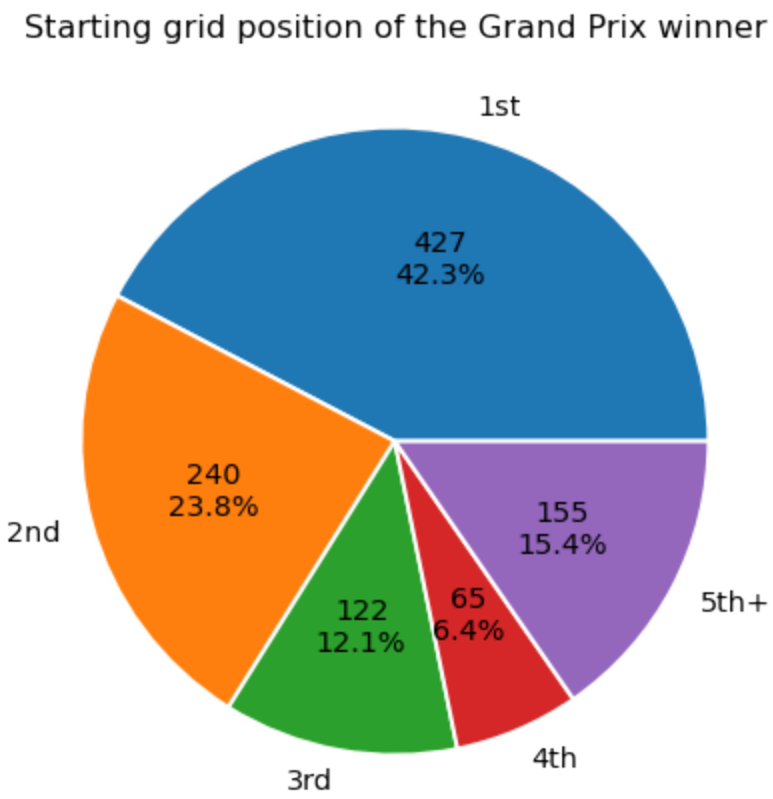
```
ax.set_xlabel('Start position')
ax.set_ylabel('Total wins')
ax.set_title('Starting grid position of the Grand Prix winner')

x_labels = ["1st", "2nd", "3rd", "4th", "5th+"]

ax.set_xticks(range(1,6))
ax.set_xticklabels(x_labels)

ax.bar(x_values, y_values)

plt.tight_layout()

fig.savefig('start_winning_position_bar.png', bbox_inches = 'tight')
```



In [20]:

```
fig, ax = plt.subplots(figsize=(8,6))
ax.pie(y_values, labels=x_labels, autopct=lambda x: '{:.0f}\n{:.1f}%'.format(x*sum(y_values)/100, x), wedgeprops={'linewi
       textprops={'size': 'x-large'})
ax.set_title('Starting grid position of the Grand Prix winner', fontsize = 16)
plt.tight_layout()
```

```
fig.savefig('start_winning_position_pie.png', bbox_inches = 'tight')
```



From the data above, 42.3% or 427 of the races were won by the driver who started the race first and by 23.8% or 240 by the driver who started second. Hence, in 66.1% or 667 of the Grands Prix the winner started on the front row.

Objective 2: Did the driver with the most wins during a season always win the World Drivers' Championship at the end of the year (the champion is the driver with the most points)?

The total number of wins for every driver for a given season is needed.

In [21]:
```
# Winners of all races no matter whether there is starting grid information
winners = set((row_data['Driver'], row_data['Grand Prix'], row_data['Year'])
              for row_name, row_data in df_Race_Details.iterrows() if row_data['Pos'] == 1)
```

In [22]:
```
# The most common driver's name in a given data for a given year
def most_common_driver(data, year):
    return Counter(race[0] for race in data if race[2] == year)

def drivers_with_most_season_wins(data, year):
    mcd = most_common_driver(data, year)
    most_wins = max(mcd.values())
    return set(key for key, value in mcd.items() if value == most_wins)

# The driver(s) with the most wins for every season
drivers_with_most_wins = set((tuple(drivers_with_most_season_wins(winners, year)), year)
                             for year in range(1950, 2023))

drivers_with_most_wins = sorted(drivers_with_most_wins, key=lambda x:x[1])
```

In [23]:
```
# World Drivers' Champions for every year
champions = set((row_data['Driver'], row_data['Year'])
                for row_name, row_data in df_Driver_Standings.iterrows() if row_data['Pos'] == 1)

champions = sorted(champions, key=lambda x:x[1])

# In case there are multiple champions in a season, they are put together in a list (similar idea to the drivers with the
champs = list()
years = set()

for champion in champions:
    if champion[1] in years:
        champs[champion[1] - 1950][0].append(champion[0])
    else:
        champs.append(([champion[0]], champion[1]))
        years.add(champion[1])
```

In [24]:
```
table_caption = "Champion(s) and Driver(s) with the most wins in a season"
table_column_names = ["Year", "Champion(s)", "Driver(s) with the most race wins"]

table_data = sorted((year, sorted(champs[year - 1950][0]), sorted(drivers_with_most_wins[year - 1950][0]))
                 for year in range(1950, 2023))

display_html_table(table_caption, table_column_names, table_data)
```

Champion(s) and Driver(s) with the most wins in a season

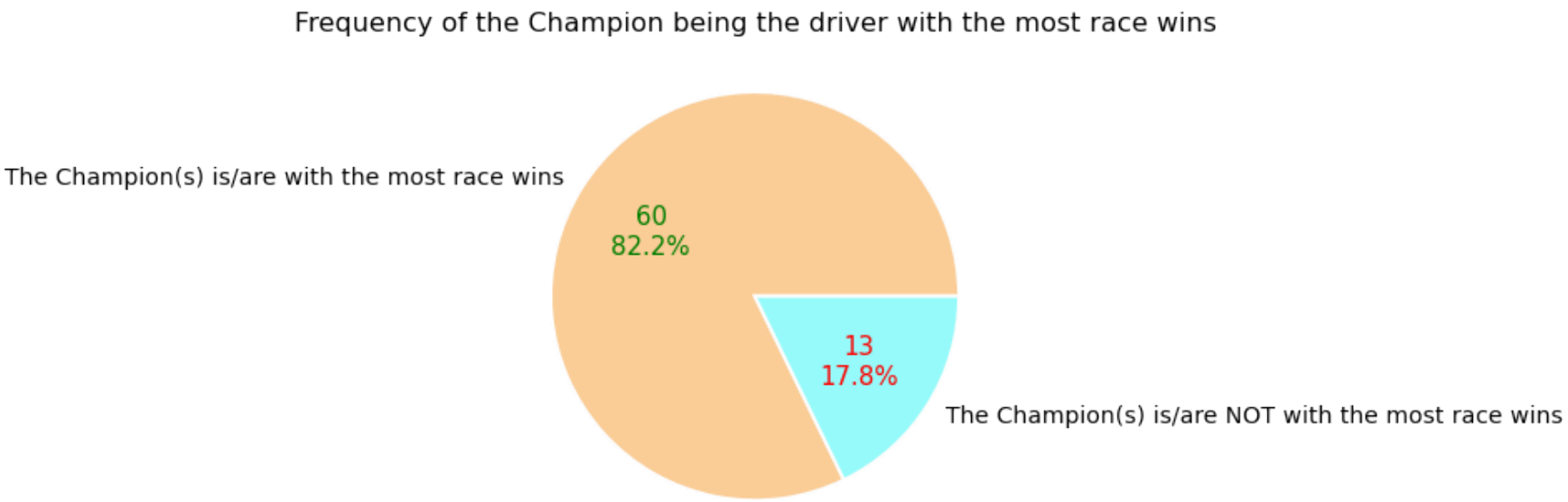
Year	Champion(s)	Driver(s) with the most race wins
1950	Nino Farina	Juan Manuel Fangio, Nino Farina
1951	Juan Manuel Fangio	Juan Manuel Fangio
1952	Alberto Ascari	Alberto Ascari
1953	Alberto Ascari	Alberto Ascari
1954	Juan Manuel Fangio	Juan Manuel Fangio
1955	Juan Manuel Fangio	Juan Manuel Fangio
1956	Juan Manuel Fangio	Juan Manuel Fangio
1957	Juan Manuel Fangio	Juan Manuel Fangio
1958	Mike Hawthorn	Stirling Moss
1959	Jack Brabham	Jack Brabham, Stirling Moss, Tony Brooks
1960	Jack Brabham	Jack Brabham
1961	Phil Hill	Phil Hill, Stirling Moss, Wolfgang von Trips
1962	Graham Hill	Graham Hill
1963	Jim Clark	Jim Clark
1964	John Surtees	Jim Clark
1965	Jim Clark	Jim Clark
1966	Jack Brabham	Jack Brabham
1967	Denny Hulme	Jim Clark
1968	Graham Hill	Graham Hill, Jackie Stewart
1969	Jackie Stewart	Jackie Stewart
1970	Jochen Rindt	Jochen Rindt
1971	Jackie Stewart	Jackie Stewart
1972	Emerson Fittipaldi	Emerson Fittipaldi

1973	**Jackie Stewart**	**Jackie Stewart**
1974	**Emerson Fittipaldi**	**Carlos Reutemann, Emerson Fittipaldi, Ronnie Peterson**
1975	**Niki Lauda**	**Niki Lauda**
1976	**James Hunt**	**James Hunt**
1977	**Niki Lauda**	**Mario Andretti**
1978	**Mario Andretti**	**Mario Andretti**
1979	**Jody Scheckter**	**Alan Jones**
1980	**Alan Jones**	**Alan Jones**
1981	**Nelson Piquet**	**Alain Prost, Nelson Piquet**
1982	**Keke Rosberg**	**Alain Prost, Didier Pironi, John Watson, Niki Lauda, Rene Arnoux**
1983	**Nelson Piquet**	**Alain Prost**
1984	**Niki Lauda**	**Alain Prost**
1985	**Alain Prost**	**Alain Prost**
1986	**Alain Prost**	**Nigel Mansell**
1987	**Nelson Piquet**	**Nigel Mansell**
1988	**Ayrton Senna**	**Ayrton Senna**
1989	**Alain Prost**	**Ayrton Senna**
1990	**Ayrton Senna**	**Ayrton Senna**
1991	**Ayrton Senna**	**Ayrton Senna**
1992	**Nigel Mansell**	**Nigel Mansell**
1993	**Alain Prost**	**Alain Prost**
1994	**Michael Schumacher**	**Michael Schumacher**
1995	**Michael Schumacher**	**Michael Schumacher**
1996	**Damon Hill**	**Damon Hill**
1997	**Jacques Villeneuve**	**Jacques Villeneuve**

1998	**Mika Hakkinen**	**Mika Hakkinen**
1999	**Mika Hakkinen**	**Mika Hakkinen**
2000	**Michael Schumacher**	**Michael Schumacher**
2001	**Michael Schumacher**	**Michael Schumacher**
2002	**Michael Schumacher**	**Michael Schumacher**
2003	**Michael Schumacher**	**Michael Schumacher**
2004	**Michael Schumacher**	**Michael Schumacher**
2005	**Fernando Alonso**	**Fernando Alonso, Kimi Räikkönen**
2006	**Fernando Alonso**	**Fernando Alonso, Michael Schumacher**
2007	**Kimi Räikkönen**	**Kimi Räikkönen**
2008	**Lewis Hamilton**	**Felipe Massa**
2009	**Jenson Button**	**Jenson Button**
2010	**Sebastian Vettel**	**Fernando Alonso, Sebastian Vettel**
2011	**Sebastian Vettel**	**Sebastian Vettel**
2012	**Sebastian Vettel**	**Sebastian Vettel**
2013	**Sebastian Vettel**	**Sebastian Vettel**
2014	**Lewis Hamilton**	**Lewis Hamilton**
2015	**Lewis Hamilton**	**Lewis Hamilton**
2016	**Nico Rosberg**	**Lewis Hamilton**
2017	**Lewis Hamilton**	**Lewis Hamilton**
2018	**Lewis Hamilton**	**Lewis Hamilton**
2019	**Lewis Hamilton**	**Lewis Hamilton**
2020	**Lewis Hamilton**	**Lewis Hamilton**
2021	**Max Verstappen**	**Max Verstappen**
2022	**Max Verstappen**	**Max Verstappen**

In [25]:
```
# Years when the champion(s) is/are also the driver(s) with the most wins. In case
# there were miltiple champions, at least one of them should be the driver with the most wins
years_with_champ_most_wins = set(year for year in range(1950, 2023)
                                 for champ in champs[year-1950][0]
                                 if champ in drivers_with_most_wins[year-1950][0])
```

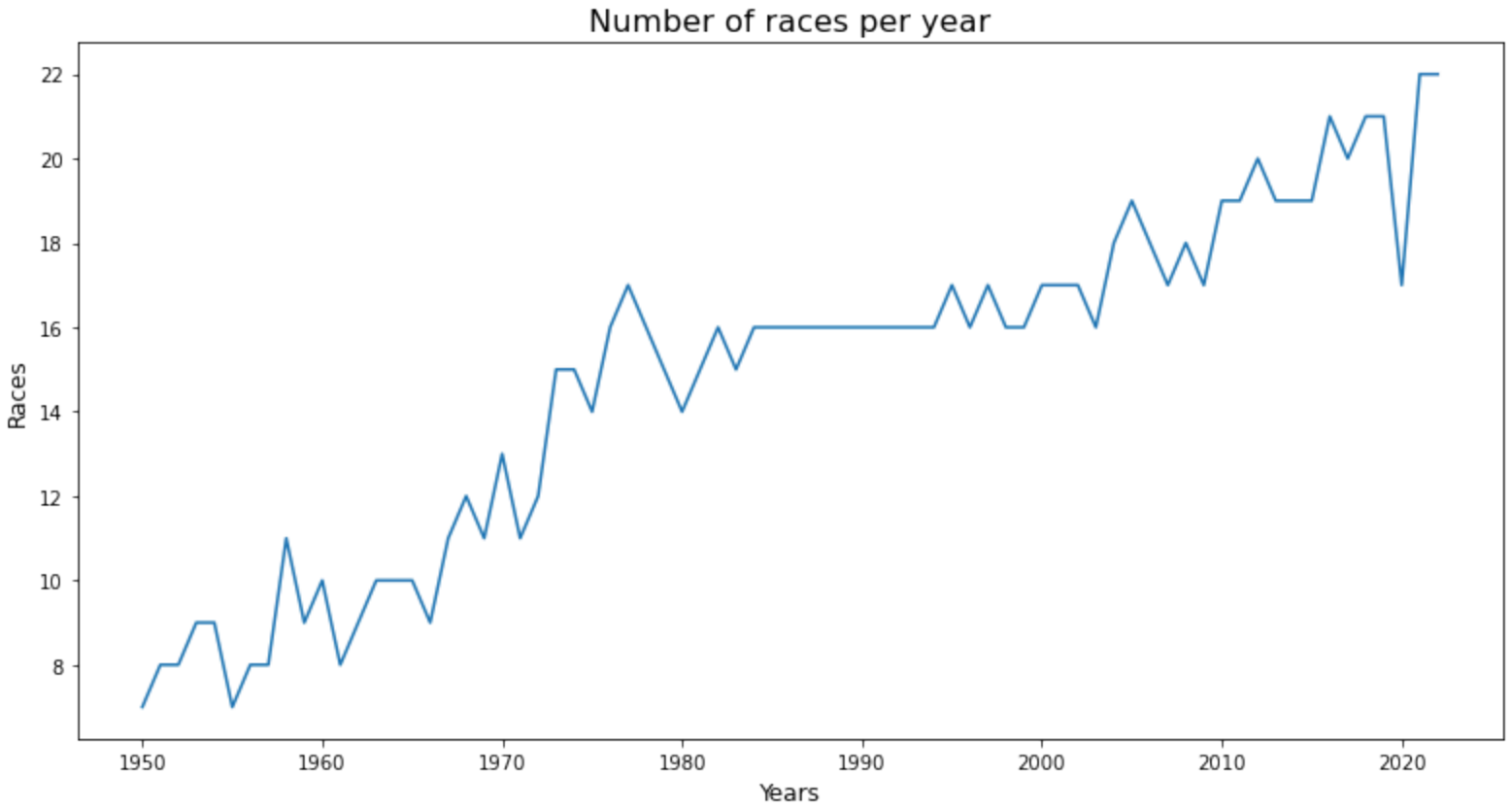
In [26]:
```
fig, ax = plt.subplots(figsize=(10,6))

x_values = [len(years_with_champ_most_wins), 73 - len(years_with_champ_most_wins)]
patches, texts, autotexts = ax.pie(x_values,
       labels=["The Champion(s) is/are with the most race wins", "The Champion(s) is/are NOT with the most race wins"],
       autopct=lambda x: '{:.0f}\n{:.1f}%'.format(x*sum(x_values)/100, x),
       wedgeprops={'linewidth': 2.0, 'edgecolor': 'white'},
       textprops={'fontsize': 14}, colors = ["#fdce98", "#98fdfb"])#, 'color': 'w'})

autotexts[0].set_color('green')
autotexts[1].set_color('red')
[autotext.set_size(15) for autotext in autotexts]

ax.set_title('Frequency of the Champion being the driver with the most race wins', fontsize = 16)
plt.tight_layout()

fig.savefig("champion_with_most_wins.png", bbox_inches = 'tight')
```



From the data above, in the majority of the seasons (60 out of 73), the eventual Champion was also the driver with the most race wins.

Objective 3: Did the number of races per season increase through the years? If so, did this lead to more race wins by the Champion or to having more race winners? Who is the driver won the most World Drivers' Championships and who is the driver won the most Grands Prix? Is this the same person?

In [27]:
```
# Number of races every year
races_yearly = Counter(race[1] for race in all_races_with_results)

fig, ax = plt.subplots(figsize=(11,6))

x_values = [year for year in range(1950, 2023)]
y_values = [races_yearly[year] for year in range(1950, 2023)]

ax.set_xlabel('Years', fontsize = 12)
ax.set_ylabel('Races', fontsize = 12)
ax.set_title('Number of races per year', fontsize = 16)

ax.plot(x_values, y_values)

plt.tight_layout()

fig.savefig('races_per_season.png', bbox_inches = 'tight')
```

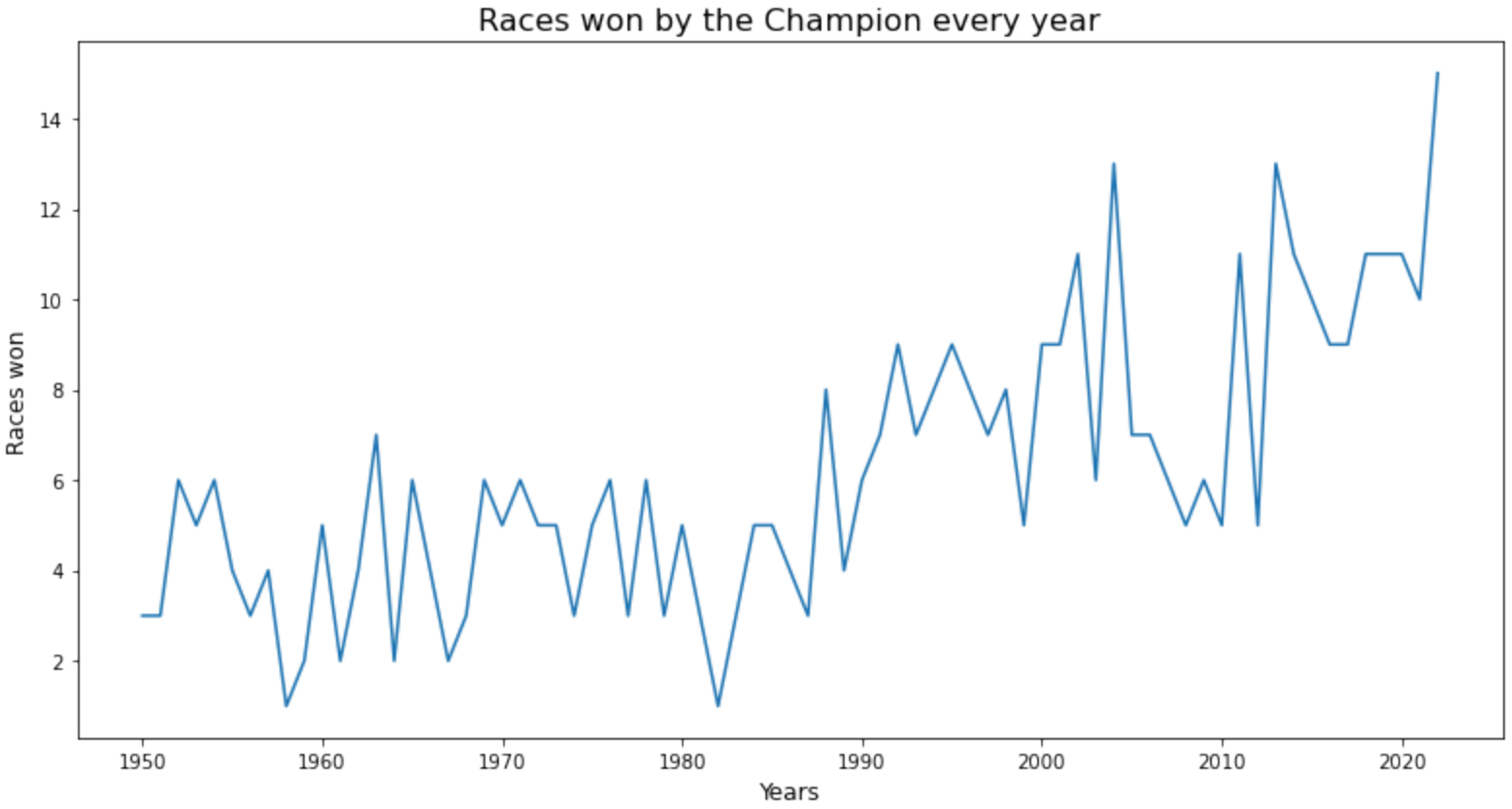


From the data above, the number of races has not been constant over the years with an overall steady increase.

In [28]:

```
# The races won by the champion every year
# In case there are multiple champions in a season, the average amount of wins is taken
races_won_by_champ = list()

for y in range(1950, 2023):
    total_wins = 0
    for i in champs[y-1950][0]:
        total_wins += most_common_driver(winners, y)[i]
    races_won_by_champ.append(total_wins/len(champs[y-1950][0]))

fig, ax = plt.subplots(figsize=(11,6))

x_values = [year for year in range(1950, 2023)]
y_values = races_won_by_champ
```

```
ax.set_xlabel('Years', fontsize = 12)
ax.set_ylabel('Races won', fontsize = 12)
ax.set_title('Races won by the Champion every year', fontsize = 16)

ax.plot(x_values, y_values)

plt.tight_layout()

fig.savefig('champion_wins.png', bbox_inches = 'tight')
```

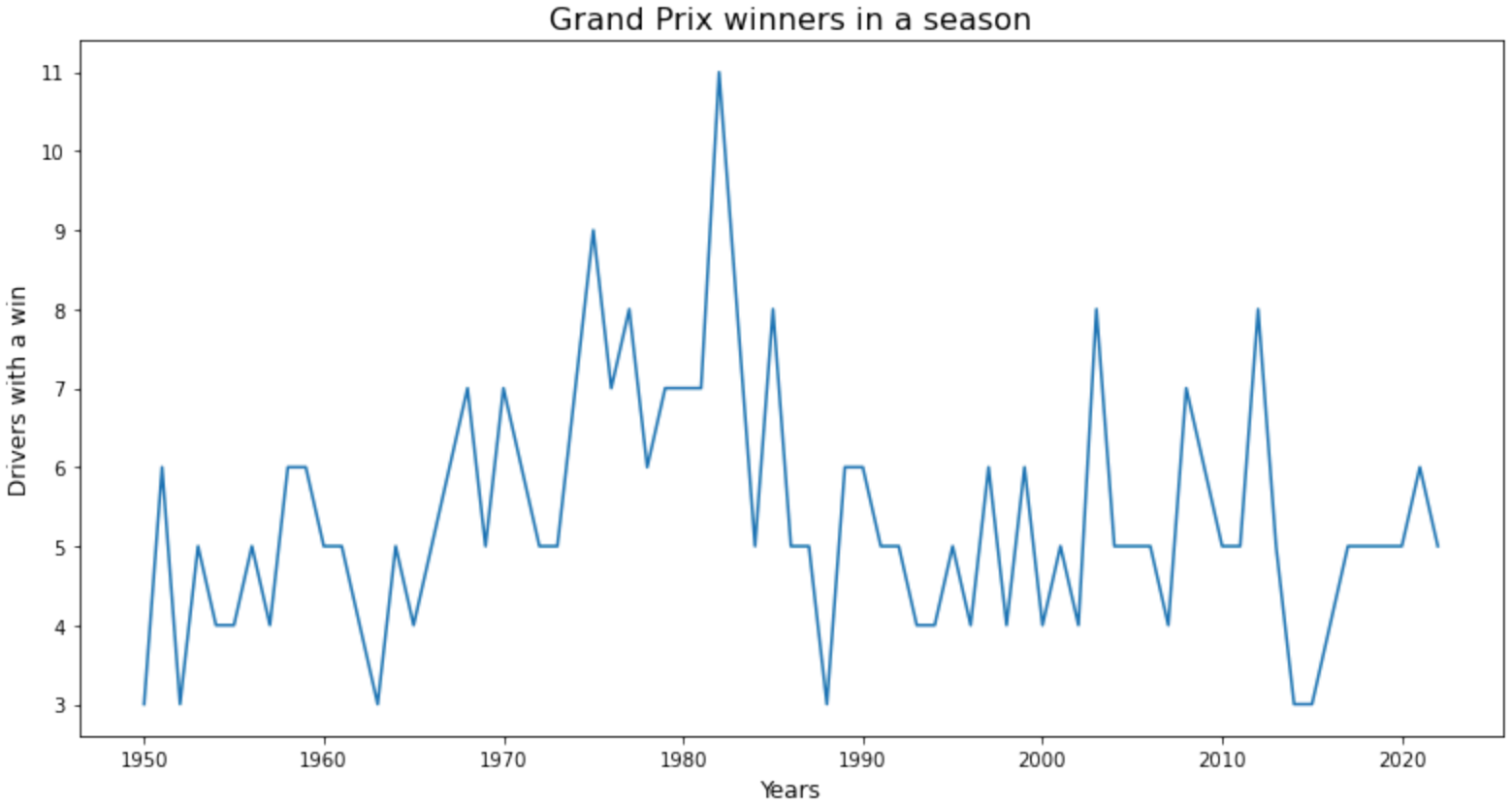


From the data above, the number of races won by the Champion in a given season fluctuates from a year to year but increases rapidly after around 1990.

In [29]:

```
# Number of drivers won a race in a season
winners_in_season = [len(most_common_driver(winners, year))
    for year in range(1950, 2023)]
```

```
fig, ax = plt.subplots(figsize=(11,6))

x_values = [year for year in range(1950, 2023)]
y_values = winners_in_season

ax.set_xlabel('Years', fontsize = 12)
ax.set_ylabel('Drivers with a win', fontsize = 12)
ax.set_title('Grand Prix winners in a season', fontsize = 16)

ax.plot(x_values, y_values)

plt.tight_layout()

fig.savefig('winners_per_season.png', bbox_inches = 'tight')
```



From the data above, with the exception of 12 years when there were at least 7 Grand Prix winners, there no more than 6 Grand Prix winners in the majority of the seasons.

In [30]:
```
# Total number of Grands Prix wins for all Grand Prix winners, sorted by Grands Prix won (from the most Grands Prix won t
total_grands_prix = Counter(win[0] for win in winners).most_common()

# Total championships for all champions, sorted by championships won (from the most championships to the least)
total_championships = Counter(ch for cham in champs for ch in cham[0]).most_common()

most_grand_prix_wins = [driver[0] for driver in total_grands_prix
                        if driver[1] == total_grands_prix[0][1]]

most_championships = [driver[0] for driver in total_championships
                        if driver[1] == total_championships[0][1]]

print("Driver(s) won the most Grands Prix is/are {}".format(", ".join(most_grand_prix_wins)))

print("Driver(s) won the most World Drivers' Championships is/are {}".format(", ".join(most_championships)))
```

```
Driver(s) won the most Grands Prix is/are Lewis Hamilton
Driver(s) won the most World Drivers' Championships is/are Michael Schumacher, Lewis Hamilton
```

Project Outcome (10 + 10 marks)

Overview of Results

Objective 1

Did one of the drivers starting a Grand Prix on the front row (1^{st} or 2^{nd} place) always win the race?

Explanation of Results

Before the start of every Grand Prix, Formula One drivers line up on the starting grid in a certain order. The first two positions, also known as the front row (drivers line up in two columns behind each other), are usually considered the ones giving the best chances for an eventual win. This data analysis project investigated whether drivers starting from the front row always win the race at the end.

Results show that in 667 or 66.1% of the cases a Grand Prix was won by a driver who started the same Grand Prix from the front row. In further details, 427 or 42.3% of the races were won by the driver who started from first position (the pole position) and 240 or 23.8% of the races were won by the driver who started from second position. In comparison, 122 of the races were won by the driver starting from third position, 65 from the driver starting from fourth position and the remaining 155 races were won by drivers starting fifth or further back the

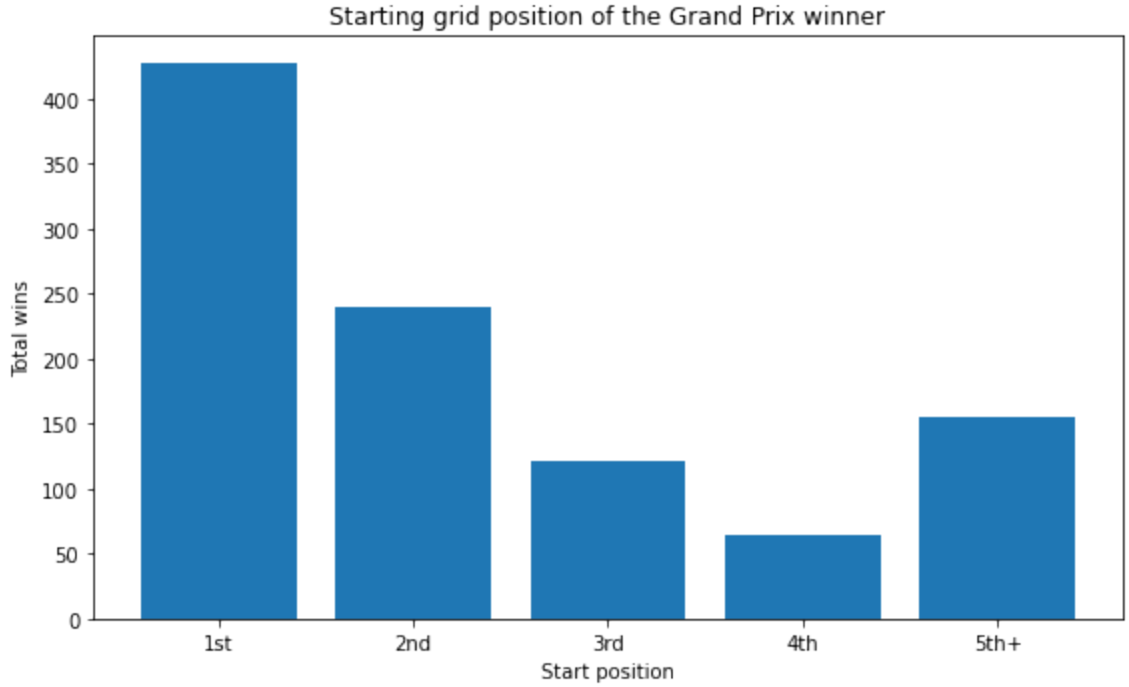
grid. Hence, the starting grid position turns out to be very important in Formula One as nearly two thirds of the Grands Prix were won by drivers who started from the front row. However, more than a third of the Grands Prix were won by drivers who started third or behind, meaning that a driver still has good chances for a Grand Prix even if they do not start first or second.

Visualisation

The following bar and pie charts give a vivid representation of the distribution of Grand Prix winners' grid starting positions, in which the importance of the grid position is well illustrated.

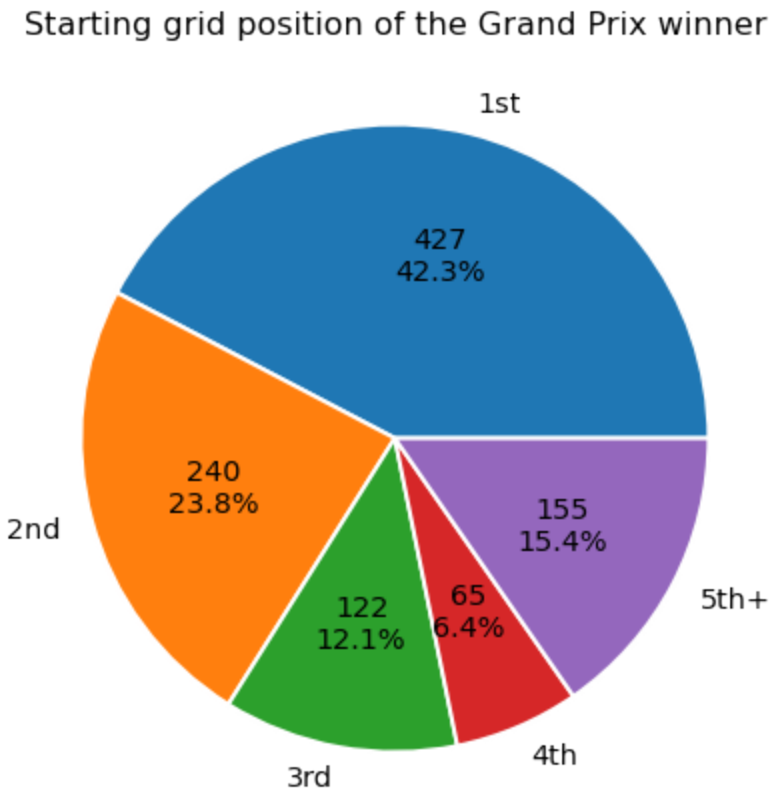
In [31]:
```
Image(filename="start_winning_position_bar.png")
```

Out[31]:



In [32]:
```
Image(filename="start_winning_position_pie.png")
```

Out[32]:



Objective 2

Did the driver with the most wins during a season always win the World Drivers' Championship at the end of the year (the champion is the driver with the most points)?

Explanation of Results

In every Grand Prix the first few drivers receive points toward the World Drivers' Championship. The higher the driver is classified in a race, the more points they get. At the end of the season the driver with the most points is crowned the World Champion. This data analysis project investigated how often the champion was also the driver with the most Grand Prix wins.

Information about all the 73 World Drivers' Championships was extracted and the results showed that in 60 or 82.2% of the Championships the champion was also the driver with the most race wins or was among the drivers with the most race wins during the given season. In other words, only 13 or 17.8% of the Championships were not won by the driver(s) with the most race wins during the given season.

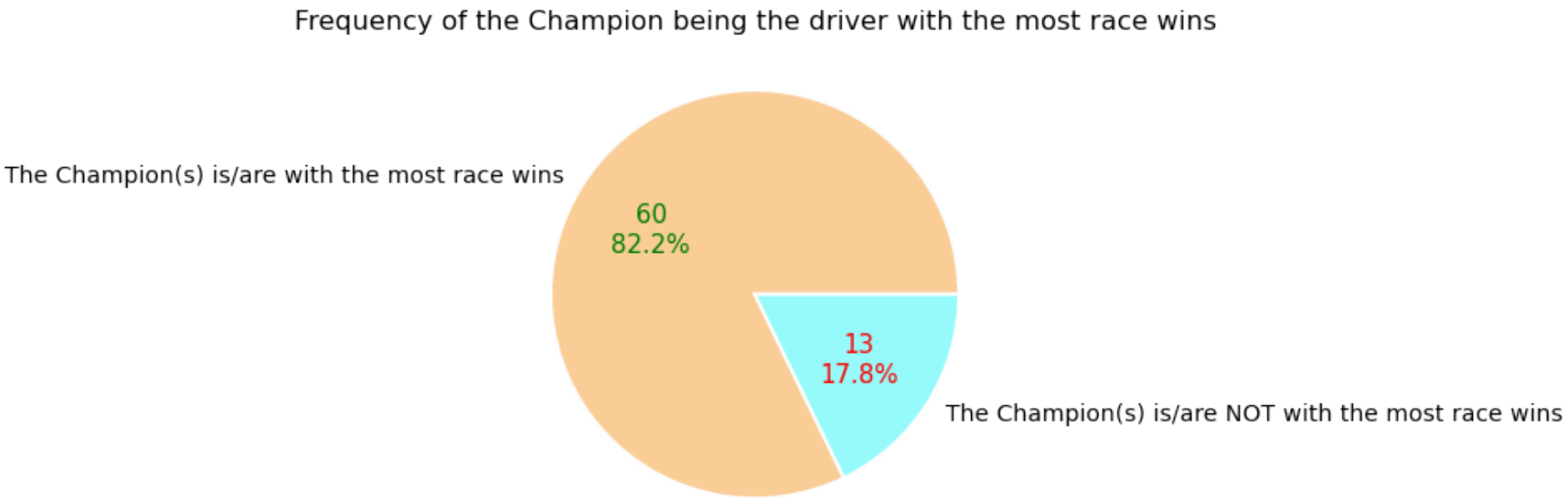
Consequently, it may be deduced that the probability the Champion to also be the driver with the most race wins is very high. However, even though it is much more rare, figures show that the Championship may still be won by a driver without this being the driver with the most race wins. So, no firm conclusions can be made.

Visualisation

The following pie chart gives a vivid representation of the frequency the World Drivers' Championship being won by the driver with the most Grand Prix wins during the season. Based on the chart, it is clear how more frequently the driver with the most wins was also the Champion.

In [33]:
```
Image(filename="champion_with_most_wins.png")
```

Out[33]:



Objective 3

Did the number of races per season increase through the years? If so, did this lead to more race wins by the Champion or to having more race winners? Who is the driver won the most World Drivers' Championships and who is the driver won the most Grands Prix? Is this the same person?

Explanation of Results

The focus of **Objective 3** was to see whether the number of races per season increased through the years and whether this led to any consequences in terms of number of race winners/wins. Additionally, information about the driver with the most World Drivers' Championships and the driver with the most Grand Prix wins was exctracted, too.

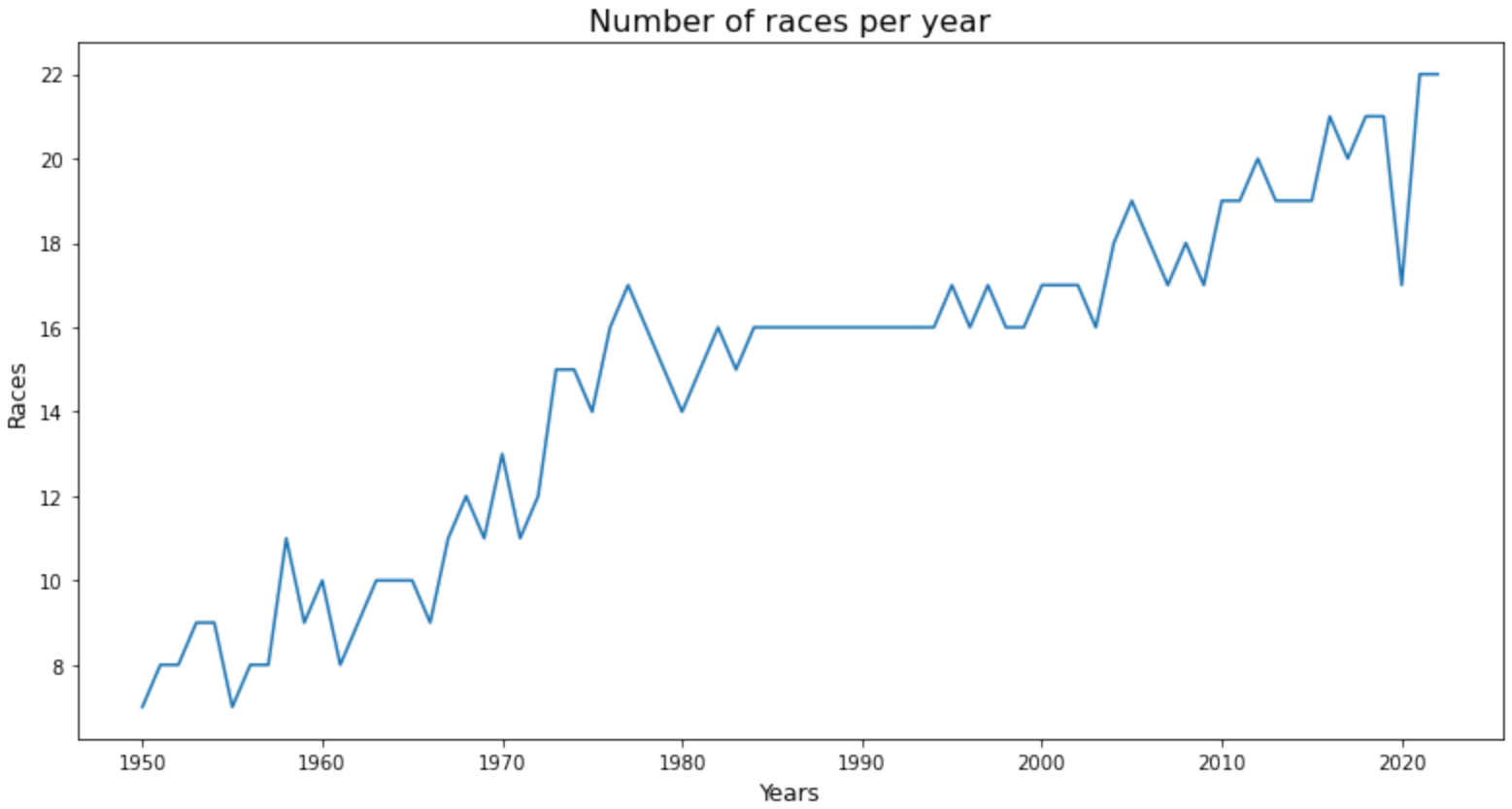
The data analysis conducted in **Objective 3** showed clearly that the number of races per season increased overall through the years. However, there were not so clear tendencies neither regarding the number of race wins by the Champion nor the number of race winners. Still, there was noticeable increase in the number of races won by the Champion after around 1990. In terms of the number of race winners, there were at most six winners per season for all seasons in the data with the exception of 12, with 11 being the highest number of winners for a season.

Finally, it was discovered that there were two drivers with the most World Drivers' Championships, namely Michael Schumacher and Lewis Hamilton. Furthermore, one of them, Lewis Hamilton, was also found to be the driver with the most Grands Prix wins.

Visualisation

In [34]:
```
Image(filename="races_per_season.png")
```

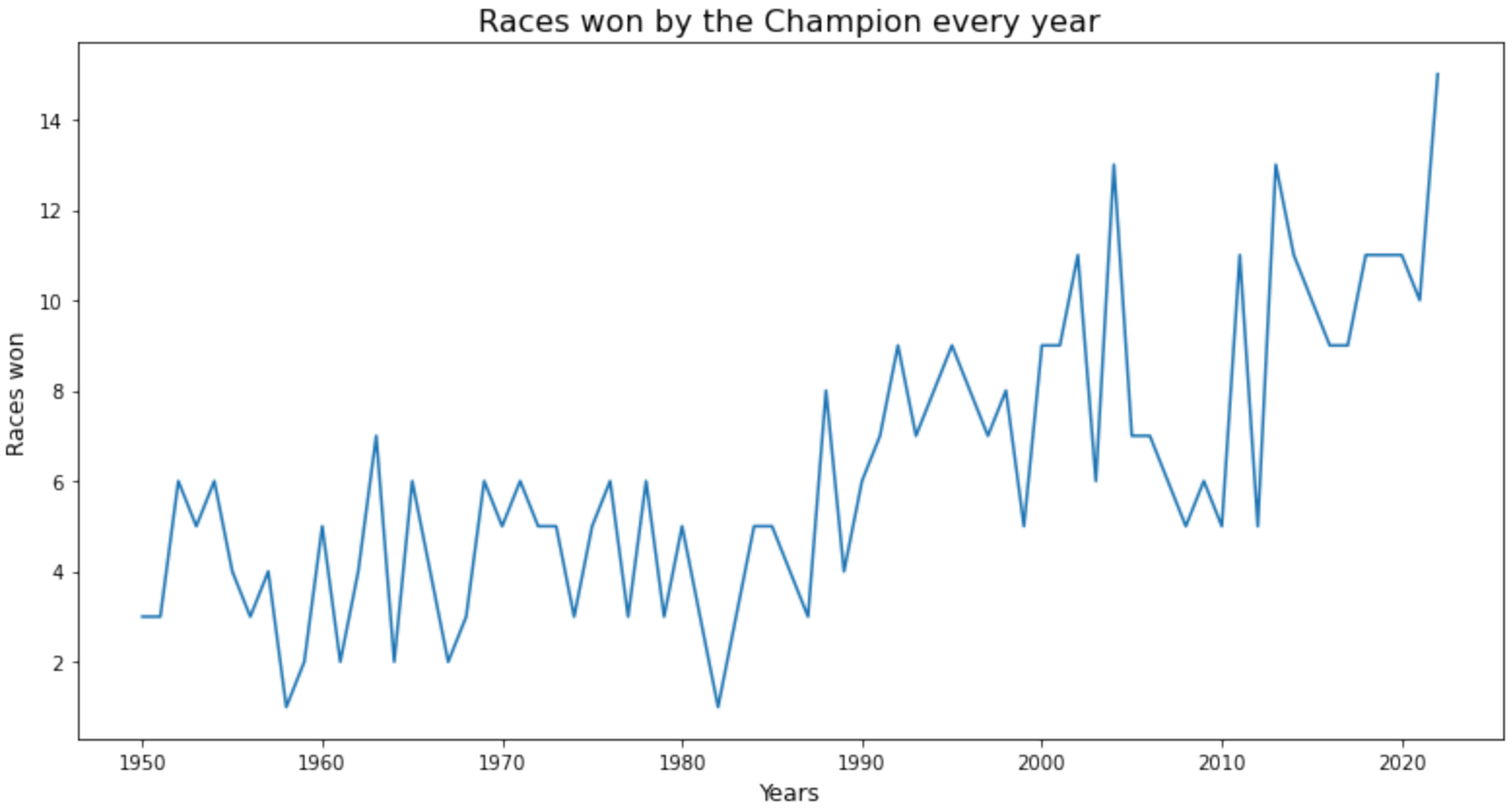
Out[34]:



In [35]:

```
Image(filename="champion_wins.png")
```

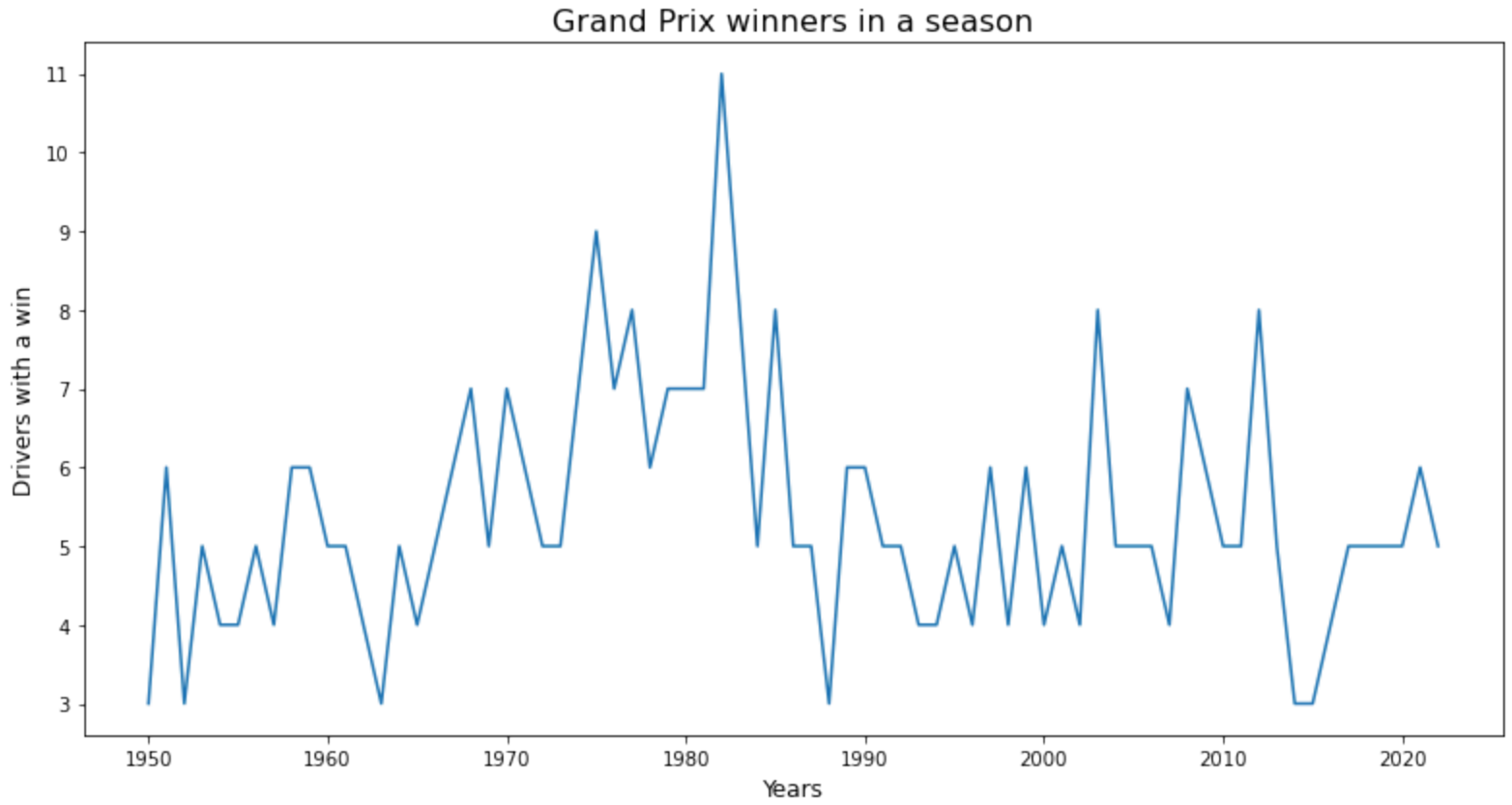
Out[35]:



In [36]:

```
Image(filename="winners_per_season.png")
```

Out[36]:



In [37]:

```
print("Driver(s) won the most Grands Prix is/are {}".format(", ".join(most_grand_prix_wins)))

print("Driver(s) won the most World Drivers' Championships is/are {}".format(", ".join(most_championships)))
```

```
Driver(s) won the most Grands Prix is/are Lewis Hamilton
Driver(s) won the most World Drivers' Championships is/are Michael Schumacher, Lewis Hamilton
```

Conclusion and presentation (10 marks)

Achievements

The data analysis conducted in this project managed to answer the questions asked in the three objectives of the project. Firstly, the results showed that the starting position of the driver for a Grand Prix is very important for the eventual win. It was shown that nearly two thirds of the races were won by a driver starting from the front row. Secondly, the analysis discover that the probability the World Drivers' Champion

also to be the driver with the most Grand Prix wins during the same year is very high. More precisely, this happend in 82.2% of the seasons in the history of the sport. Thirdly, it was shown that there was a tendency of the number of races being increased through the years, which did not lead to constant increases neither in races being won by the eventual season Champion, nor in the number of race winners per season. However, there was still an increase in the number of races won by the Champion after around 1990. Finally, it was found that two drivers are with the most World Drivers' Championships, one of them also being the driver with the most Grand Prix wins, which is expected.

Limitations

Formula One is a very complicated motorsport which involves numerous aspects. In the current data analysis only a small portion of those aspects were taken into consideration, such as a Grand Prix starting grid, Grand Prix final results and the World Drivers' Championship. Additionally, the used dataset includes all seasons until 2022, while the sport continues developing and the 2023 and the current 2024 season need to be included in the analysis, as well.

Future Work

In future work, an up-to-date dataset should be used as well as all aspects of Formula One need to be taken into consideration to make the most complete analysis possible. This include details such as qualifying performance, sprint race results, pit stops perforamnce, car performance and many more. The hypothesis is that every single aspect of the sport, especially in combination with the other apects, has an impact on the final results.

Video Presentation

The video presentation is in the folder with all other files, named P4DS_A2_Data_Analysis_Project.mp4