DataFrame Import

Xander Palermo

This file extracts an existing the f1db from the Postgres dump provided by the F1DB project. This file then saves a handful of pickled DataFrames to be used in other notebooks

```
import json
import pickle

import sqlalchemy
import pandas as pd
```

Give access to notebook for Postgres Database actions

```
# Replace these with your actual credentials
db_username = "postgres"
db_password = input("Database Password: ")
db_host = "localhost"
db_port = "5432"
db_name = "f1db"

# Connect to postgres database
engine =
sqlalchemy.create_engine(f"postgresql+psycopg2://{db_username}:
{db_password}@{db_host}:{db_port}/{db_name}")
```

Import Info

Functionality of these tools are saved to *DataFrameImport.py* so that they can be utilized in other Notebooks.

```
from DataFrameImport import *
```

Table Look Up Functionality

Create a function that can be used to look up specific tables, or list all table

```
schema_info_path = "resources/f1db/"
schema_file = "f1db.schema.json"

with open(schema_info_path + schema_file, 'r') as f:
    info_json = json.load(f)

schema_names = list(dict(info_json["properties"].items()).keys())
schema_descriptions = [x["description"] for x in
list(dict(info_json["properties"].items()).values())]
```

```
schema_info = dict(zip(schema_names, schema_descriptions))

with open("resources/info/schema_info_dump.pkl", "wb") as file:
    pickle.dump(schema_info, file)

# Prints alias of every table
def list_schemas():
    for s in schema_info.keys():
        print(s)

#Prints table alias and description, or provide specific table alias
to get single description
def get_schema_info(t: str = ""):
    if t in schema_info.keys():
        print(t)
        print(schema_info[t])
        print()
    else:
        for t in schema_info.keys():
            get_schema_info(t)

# ex:
# list_schemas()
# get_schema_info()
```

Column Look Up Functionality

Create an object that holds information of all schema descriptions and data types (accessed using dot notation)

```
# Placeholder class that is used to attach dynamically named
attributes
class Null:
    def __str__(self):
        return "\n".join(f"{k} = {v}" for k, v in vars(self).items())
+ "\n"

column_info = info_json['definitions']

with engine.connect() as c:
    query = c.execute(sqlalchemy.text(""
                                      "SELECT table_name "
                                      "FROM information_schema.tables
"
                                      "WHERE table_schema = 'public' "
                                      "    AND table_type = 'BASE
TABLE';"))
    table_names = query.all()
table_names = [list(i)[0] for i in table_names]

table_var_names = list(map(lambda x: x.replace('_', '
').title().replace(' ', ''), table_names))

schema = Null()

for table in table_var_names:
    try:
        #Create column object
        attr_names = list(column_info[table]['properties'].keys())
        exec(f"schema.{table} = Null()")


        for attr_name in attr_names:
            # Assign type and description name
            exec(f"schema.{table}.{attr_name} = Null()")

            try:
                attr_type = column_info[table]['properties']
[attr_name]["type"]
                exec(f"schema.{table}.{attr_name}.type = attr_type")
            except KeyError:
                pass

            try:
                attr_desc = column_info[table]['properties']
[attr_name]["description"]
                exec(f"schema.{table}.{attr_name}.description =
attr_desc")
            except KeyError:
                pass
    except KeyError:
        pass

with open("resources/info/schema_columns_info_dump.pkl", "wb") as
file:
    pickle.dump(schema, file)


# ex:
# schema.{table_name}.{column_name(optional)}
# print(schema.Continent)            #Gives info on all columns in
DataFame Continent
# print(schema.Continent.id)         #Gives info on specific column
(id) in DataFrame Continent
```

Pickle Tables

Saving tables as pickled DataFrames so that work can be independently from existence of a postgres database.

```
path = "resources/pickled_tables/"
extension = ".plk"

for table in table_names:
    with engine.connect() as conn:
        query = f"SELECT * FROM {table}"

        dataFrame = pd.read_sql_query(query, conn)
        dataFrame.to_pickle(f'{path}{table}{extension}')

# Read using the following code
# table = dataFrame you want to load
# with open(f"resources/pickled_tables/{table}.pkl","rb") as file:
#   dataFrame = pickle.load(file)
```

Driver Analysis

Analyzing lap times, podiums, wins, championships, career length, and races to determine best drivers.

```
import pandas as pd

import seaborn as sns

sns.set_palette('bright')
```

Reading DataFrames

```
driver = pd.read_pickle("driver.plk")
driver

                         id                    name first_name  \
0              adderly-fong            Adderly Fong    Adderly
1              adolf-brudes            Adolf Brudes      Adolf
2       adolfo-schwelm-cruz     Adolfo Schwelm Cruz     Adolfo
3             adrian-campos           Adrián Campos     Adrián
4              adrian-sutil            Adrian Sutil     Adrian
..                      ...                     ...        ...
907                yuji-ide                Yuji Ide       Yuji
908             yuki-tsunoda            Yuki Tsunoda       Yuki
909  yves-giraud-cabantous  Yves Giraud-Cabantous       Yves
910            zak-osullivan           Zak O'Sullivan        Zak
911       zsolt-baumgartner       Zsolt Baumgartner      Zsolt

            last_name                         full_name
abbreviation  \
0                Fong          Adderly Fong Cheun-yue
FON
1              Brudes          Adolf Brudes von Breslau
BRU
2        Schwelm Cruz      Adolfo Julio Carlos Schwelm Cruz
SCH
3             Campos             Adrián Campos Suñer
CAM
4               Sutil                    Adrian Sutil
SUT
..                ...                             ...
...
907               Ide                        Yuji Ide
IDE
908           Tsunoda                    Yuki Tsunoda
TSU
```

```
909  Giraud-Cabantous  Marius Aristide Yves Giraud-Cabantous
CAB
910        O'Sullivan                         Zak O'Sullivan
OSU
911       Baumgartner                      Zsolt Baumgartner
BAU

    permanent_number gender date_of_birth date_of_death  ...  \
0               None   MALE    1990-03-02          None  ...
1               None   MALE    1899-10-15    1986-11-05  ...
2               None   MALE    1923-06-28    2012-02-10  ...
3               None   MALE    1960-06-17    2021-01-27  ...
4               None   MALE    1983-01-11          None  ...
..               ...    ...           ...           ...  ...
907             None   MALE    1975-01-21          None  ...
908               22   MALE    2000-05-11          None  ...
909             None   MALE    1904-10-08    1973-03-30  ...
910             None   MALE    2005-02-06          None  ...
911             None   MALE    1981-01-01          None  ...

    total_race_starts total_race_wins total_race_laps total_podiums  \
0                   0               0               0             0
1                   1               0               5             0
2                   1               0              20             0
3                  17               0             433             0
4                 128               0            6022             0
..                ...             ...             ...           ...
907                 4               0             145             0
908               105               0            5653             0
909                13               0             522             0
910                 0               0               0             0
911                20               0             959             0

     total_points  total_championship_points  total_pole_positions  \
0             0.0                        0.0                     0
1             0.0                        0.0                     0
2             0.0                        0.0                     0
3             0.0                        0.0                     0
4           124.0                      124.0                     0
..            ...                        ...                   ...
907           0.0                        0.0                     0
908         111.0                      111.0                     0
909           5.0                        5.0                     0
910           0.0                        0.0                     0
911           1.0                        1.0                     0

     total_fastest_laps  total_driver_of_the_day  total_grand_slams
0                     0                        0                  0
1                     0                        0                  0
2                     0                        0                  0
```

```
3                      0                         0                  0
4                      1                         0                  0
..                   ...                       ...                ...
907                    0                         0                  0
908                    1                         2                  0
909                    0                         0                  0
910                    0                         0                  0
911                    0                         0                  0

[912 rows x 29 columns]
```

```
driver_by_season = pd.read_pickle("season_driver.plk")
driver_by_season
```

```
      year           driver_id  position_number position_text  \
0     1950  juan-manuel-fangio              2.0             2
1     1950      luigi-fagioli              3.0             3
2     1950        nino-farina              1.0             1
3     1950        reg-parnell              9.0             9
4     1950     consalvo-sanesi              NaN          None
...    ...                 ...              ...           ...
3374  2025      lewis-hamilton              6.0             6
3375  2025      dino-beganovic              NaN          None
3376  2025   gabriel-bortoleto             18.0            18
3377  2025     nico-hulkenberg             10.0            10
3378  2025        isack-hadjar              9.0             9

      best_starting_grid_position  best_race_result
total_race_entries  \
0                             1.0               1.0
6
1                             2.0               2.0
6
2                             1.0               1.0
6
3                             4.0               3.0
2
4                             4.0               NaN
1
...                           ...               ...                  ..
.
3374                          4.0               4.0
18
3375                          NaN               NaN
0
3376                          7.0               6.0
18
3377                         11.0               3.0
18
3378                          4.0               3.0
```

```
18

      total_race_starts  total_race_wins  total_race_laps
total_podiums  \
0                     6                3              317
3
1                     6                0              291
5
2                     6                3              282
3
3                     2                0               80
1
4                     1                0               11
0
...                 ...              ...              ...              .
..
3374                 18                0             1030
0
3375                  0                0                0
0
3376                 18                0              986
0
3377                 17                0              963
1
3378                 18                0              983
1

      total_points  total_pole_positions  total_fastest_laps  \
0             27.0                     4                   3
1             28.0                     0                   0
2             30.0                     2                   3
3              4.0                     0                   0
4              0.0                     0                   0
...            ...                   ...                 ...
3374         125.0                     0                   1
3375           0.0                     0                   0
3376          18.0                     0                   0
3377          37.0                     0                   0
3378          39.0                     0                   0

      total_driver_of_the_day  total_grand_slams
0                           0                  0
1                           0                  0
2                           0                  0
3                           0                  0
4                           0                  0
...                       ...                ...
3374                        2                  0
3375                        0                  0
3376                        2                  0
```

```
3377                        1                  0
3378                        1                  0

[3379 rows x 16 columns]
```

```
race_data = pd.read_pickle("race_data.plk")
race_data
```

```
        race_id                       type  position_display_order  \
0           290     PRE_QUALIFYING_RESULT                        1
1           290     PRE_QUALIFYING_RESULT                        2
2           290     PRE_QUALIFYING_RESULT                        3
3           290     PRE_QUALIFYING_RESULT                        4
4           290     PRE_QUALIFYING_RESULT                        5
...         ...                        ...                      ...
183627     1143  DRIVER_OF_THE_DAY_RESULT                        1
183628     1143  DRIVER_OF_THE_DAY_RESULT                        2
183629     1143  DRIVER_OF_THE_DAY_RESULT                        3
183630     1143  DRIVER_OF_THE_DAY_RESULT                        4
183631     1143  DRIVER_OF_THE_DAY_RESULT                        5

        position_number position_text driver_number
driver_id  \
0                   1.0             1            40   gilles-
villeneuve
1                   2.0             2            23      patrick-
tambay
2                   3.0             3            34  jean-pierre-
jarier
3                   4.0             4            30        brett-
lunger
4                   5.0             5            38        brian-
henton
...                 ...           ...           ...                  ..
.
183627              1.0             1            14     fernando-
alonso
183628              2.0             2            63      george-
russell
183629              3.0             3             1      max-
verstappen
183630              4.0             4             4        lando-
norris
183631              5.0             5            44      lewis-
hamilton

       constructor_id engine_manufacturer_id tyre_manufacturer_id  ...
\
0             mclaren                   ford             goodyear  ...
```

```
1             ensign                   ford            goodyear  ...
2             penske                   ford            goodyear  ...
3            mclaren                   ford            goodyear  ...
4              march                   ford            goodyear  ...
...              ...                    ...                 ...  ...
183627  aston-martin               mercedes             pirelli  ...
183628      mercedes               mercedes             pirelli  ...
183629      red-bull             honda-rbpt             pirelli  ...
183630       mclaren               mercedes             pirelli  ...
183631       ferrari                ferrari             pirelli  ...

       fastest_lap_time_millis  fastest_lap_gap fastest_lap_gap_millis
\
0                          NaN             None                    NaN
1                          NaN             None                    NaN
2                          NaN             None                    NaN
3                          NaN             None                    NaN
4                          NaN             None                    NaN
...                        ...              ...                    ...
183627                     NaN             None                    NaN
183628                     NaN             None                    NaN
183629                     NaN             None                    NaN
183630                     NaN             None                    NaN
183631                     NaN             None                    NaN

        fastest_lap_interval fastest_lap_interval_millis
pit_stop_stop  \
0                       None                         NaN
NaN
1                       None                         NaN
```

```
NaN
2                        None                          NaN
NaN
3                        None                          NaN
NaN
4                        None                          NaN
NaN
...                       ...                          ...               ..
.
183627                   None                          NaN
NaN
183628                   None                          NaN
NaN
183629                   None                          NaN
NaN
183630                   None                          NaN
NaN
183631                   None                          NaN
NaN

        pit_stop_lap pit_stop_time  pit_stop_time_millis  \
0                NaN          None                   NaN
1                NaN          None                   NaN
2                NaN          None                   NaN
3                NaN          None                   NaN
4                NaN          None                   NaN
...              ...           ...                   ...
183627           NaN          None                   NaN
183628           NaN          None                   NaN
183629           NaN          None                   NaN
183630           NaN          None                   NaN
183631           NaN          None                   NaN

       driver_of_the_day_percentage
0                               NaN
1                               NaN
2                               NaN
3                               NaN
4                               NaN
...                             ...
183627                         22.5
183628                         16.4
183629                         14.5
183630                          8.7
183631                          7.6

[183632 rows x 71 columns]
```

Cleaning DataFrames

driver DataFrame

```
driver.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 912 entries, 0 to 911
Data columns (total 29 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   id                               912 non-null    object 
 1   name                             912 non-null    object 
 2   first_name                       912 non-null    object 
 3   last_name                        912 non-null    object 
 4   full_name                        912 non-null    object 
 5   abbreviation                     912 non-null    object 
 6   permanent_number                 47 non-null     object 
 7   gender                           912 non-null    object 
 8   date_of_birth                    912 non-null    object 
 9   date_of_death                    501 non-null    object 
 10  place_of_birth                   912 non-null    object 
 11  country_of_birth_country_id      912 non-null    object 
 12  nationality_country_id           912 non-null    object 
 13  second_nationality_country_id    7 non-null      object 
 14  best_championship_position       384 non-null    float64
 15  best_starting_grid_position      791 non-null    float64
 16  best_race_result                 678 non-null    float64
 17  total_championship_wins          912 non-null    int64  
 18  total_race_entries               912 non-null    int64  
 19  total_race_starts                912 non-null    int64  
 20  total_race_wins                  912 non-null    int64  
 21  total_race_laps                  912 non-null    int64  
 22  total_podiums                    912 non-null    int64  
 23  total_points                     912 non-null    float64
 24  total_championship_points        912 non-null    float64
 25  total_pole_positions             912 non-null    int64  
 26  total_fastest_laps               912 non-null    int64  
 27  total_driver_of_the_day          912 non-null    int64  
 28  total_grand_slams                912 non-null    int64  
dtypes: float64(5), int64(10), object(14)
memory usage: 206.8+ KB
```

```
driver = 
driver.drop(columns=['full_name','permanent_number','gender','date_of_
birth','date_of_death','place_of_birth',

'country_of_birth_country_id','second_nationality_country_id','total_r
ace_entries',
'total_championship_points','total_driver_of_the_day'])
driver

                          id                   name first_name  \
0               adderly-fong           Adderly Fong    Adderly
1              adolf-brudes           Adolf Brudes      Adolf
2        adolfo-schwelm-cruz    Adolfo Schwelm Cruz     Adolfo
3              adrian-campos          Adrián Campos     Adrián
4               adrian-sutil           Adrian Sutil     Adrian
..                       ...                    ...        ...
907                 yuji-ide               Yuji Ide       Yuji
908             yuki-tsunoda           Yuki Tsunoda       Yuki
909  yves-giraud-cabantous  Yves Giraud-Cabantous       Yves
910            zak-osullivan          Zak O'Sullivan        Zak
911        zsolt-baumgartner      Zsolt Baumgartner      Zsolt

            last_name abbreviation nationality_country_id  \
0                Fong          FON              hong-kong
1              Brudes          BRU                germany
2        Schwelm Cruz          SCH              argentina
3              Campos          CAM                  spain
4               Sutil          SUT                germany
..                ...          ...                    ...
907               Ide          IDE                  japan
908           Tsunoda          TSU                  japan
909  Giraud-Cabantous          CAB                 france
910         O'Sullivan          OSU         united-kingdom
911        Baumgartner          BAU                hungary

     best_championship_position  best_starting_grid_position  \
0                           NaN                          NaN
1                           NaN                         19.0
2                           NaN                         13.0
3                           NaN                         16.0
4                           9.0                          2.0
..                          ...                          ...
907                        25.0                         18.0
908                        12.0                          3.0
909                        14.0                          5.0
910                         NaN                          NaN
911                        20.0                         17.0

     best_race_result  total_championship_wins  total_race_starts  \
0                 NaN                        0                  0
1                 NaN                        0                  1
2                 NaN                        0                  1
3                14.0                        0                 17
4                 4.0                        0                128
..                ...                      ...                ...
907              13.0                        0                  4
```

```
908            4.0                       0               105
909            4.0                       0                13
910            NaN                       0                 0
911            8.0                       0                20

     total_race_wins  total_race_laps  total_podiums  total_points  \
0                  0                0              0           0.0
1                  0                5              0           0.0
2                  0               20              0           0.0
3                  0              433              0           0.0
4                  0             6022              0         124.0
..               ...              ...            ...           ...
907                0              145              0           0.0
908                0             5653              0         111.0
909                0              522              0           5.0
910                0                0              0           0.0
911                0              959              0           1.0

     total_pole_positions  total_fastest_laps  total_grand_slams
0                       0                   0                  0
1                       0                   0                  0
2                       0                   0                  0
3                       0                   0                  0
4                       0                   1                  0
..                    ...                 ...                ...
907                     0                   0                  0
908                     0                   1                  0
909                     0                   0                  0
910                     0                   0                  0
911                     0                   0                  0

[912 rows x 18 columns]
```

driver_by_season DataFrame

```
driver_by_season.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3379 entries, 0 to 3378
Data columns (total 16 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   year                         3379 non-null   int64  
 1   driver_id                    3379 non-null   object 
 2   position_number              1657 non-null   float64
 3   position_text                1658 non-null   object 
 4   best_starting_grid_position  3041 non-null   float64
 5   best_race_result             2652 non-null   float64
 6   total_race_entries           3379 non-null   int64  
 7   total_race_starts            3379 non-null   int64  
 8   total_race_wins              3379 non-null   int64
 9   total_race_laps              3379 non-null   int64
 10  total_podiums                3379 non-null   int64
 11  total_points                 3379 non-null   float64
 12  total_pole_positions         3379 non-null   int64
 13  total_fastest_laps           3379 non-null   int64
 14  total_driver_of_the_day      3379 non-null   int64
 15  total_grand_slams            3379 non-null   int64
dtypes: float64(4), int64(10), object(2)
memory usage: 422.5+ KB

driver_by_season = 
driver_by_season.drop(columns=['position_number','total_race_entries',
'total_driver_of_the_day'])
driver_by_season = 
driver_by_season.rename(columns={'position_text':'position'})
driver_by_season

      year           driver_id position  
best_starting_grid_position  \
0     1950  juan-manuel-fangio        2                          1.0   

1     1950       luigi-fagioli        3                          2.0   

2     1950         nino-farina        1                          1.0   

3     1950         reg-parnell        9                          4.0   

4     1950      consalvo-sanesi     None                          4.0   

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

3374  2025      lewis-hamilton        6                          4.0   

3375  2025       dino-beganovic     None                          NaN   

3376  2025   gabriel-bortoleto       18                          7.0   

3377  2025     nico-hulkenberg       10                         11.0   

3378  2025         isack-hadjar        9                          4.0   


      best_race_result  total_race_starts  total_race_wins  
total_race_laps  \
0                  1.0                  6                3
317   
1                  2.0                  6                0
291   
2                  1.0                  6                3
282   
```

```
3                    3.0                   2                 0
80
4                    NaN                   1                 0
11
...                  ...                 ...               ...
...
3374                 4.0                  18                 0
1030
3375                 NaN                   0                 0
0
3376                 6.0                  18                 0
986
3377                 3.0                  17                 0
963
3378                 3.0                  18                 0
983

      total_podiums  total_points  total_pole_positions
total_fastest_laps  \
0                 3          27.0                     4
3
1                 5          28.0                     0
0
2                 3          30.0                     2
3
3                 1           4.0                     0
0
4                 0           0.0                     0
0
...             ...           ...                   ...
...
3374              0         125.0                     0
1
3375              0           0.0                     0
0
3376              0          18.0                     0
0
3377              1          37.0                     0
0
3378              1          39.0                     0
0

      total_grand_slams
0                     0
1                     0
2                     0
3                     0
4                     0
...                 ...
```

```
3374                  0
3375                  0
3376                  0
3377                  0
3378                  0

[3379 rows x 13 columns]
```

race_data DataFrame

```
race_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183632 entries, 0 to 183631
Data columns (total 71 columns):
 #   Column                                                Non-Null
Count   Dtype
---  ------
--------------   -----
 0   race_id                                               183632 non-
null  int64
 1   type                                                  183632 non-
null  object
 2   position_display_order                                183632 non-
null  int64
 3   position_number                                       172468 non-
null  float64
 4   position_text                                         183632 non-
null  object
 5   driver_number                                         183632 non-
null  object
 6   driver_id                                             183632 non-
null  object
 7   constructor_id                                        183632 non-
null  object
 8   engine_manufacturer_id                                183632 non-
null  object
 9   tyre_manufacturer_id                                  183632 non-
null  object
 10  practice_time                                         47260 non-
null   object
 11  practice_time_millis                                  47260 non-
null   float64
 12  practice_gap                                          45124 non-
null   object
 13  practice_gap_millis                                   45124 non-
null   float64
 14  practice_interval                                     45124 non-
null   object
 15  practice_interval_millis                              45124 non-
```

```
null   float64
 16  practice_laps                                          38322 non-
null   float64
 17  qualifying_time                                        33926 non-
null   object
 18  qualifying_time_millis                                 33926 non-
null   float64
 19  qualifying_q1                                          8470 non-
null    object
 20  qualifying_q1_millis                                   8470 non-
null    float64
 21  qualifying_q2                                          6216 non-
null    object
 22  qualifying_q2_millis                                   6216 non-
null    float64
 23  qualifying_q3                                          3952 non-
null    object
 24  qualifying_q3_millis                                   3952 non-
null    float64
 25  qualifying_gap                                         36049 non-
null   object
 26  qualifying_gap_millis                                  36049 non-
null   float64
 27  qualifying_interval                                    36036 non-
null   object
 28  qualifying_interval_millis                             36036 non-
null   float64
 29  qualifying_laps                                        17016 non-
null   float64
 30  starting_grid_position_qualification_position_number  25680 non-
null   float64
 31  starting_grid_position_qualification_position_text    25809 non-
null   object
 32  starting_grid_position_grid_penalty                    573 non-
null     object
 33  starting_grid_position_grid_penalty_positions          500 non-
null     float64
 34  starting_grid_position_time                            25258 non-
null   object
 35  starting_grid_position_time_millis                     25258 non-
null   float64
 36  race_shared_car                                        27591 non-
null   object
 37  race_laps                                              25664 non-
null   float64
 38  race_time                                              8318 non-
null    object
 39  race_time_millis                                       8318 non-
null    float64
```

```
 40  race_time_penalty                                    274 non-
null     object
 41  race_time_penalty_millis                             274 non-
null     float64
 42  race_gap                                             14822 non-
null   object
 43  race_gap_millis                                      7154 non-
null    float64
 44  race_gap_laps                                        7668 non-
null    float64
 45  race_interval                                        7136 non-
null    object
 46  race_interval_millis                                 7136 non-
null    float64
 47  race_reason_retired                                  9998 non-
null    object
 48  race_points                                          8505 non-
null    float64
 49  race_pole_position                                   27591 non-
null   object
 50  race_qualification_position_number                   26872 non-
null   float64
 51  race_qualification_position_text                     27009 non-
null   object
 52  race_grid_position_number                            25584 non-
null   float64
 53  race_grid_position_text                              25815 non-
null   object
 54  race_positions_gained                                16626 non-
null   float64
 55  race_pit_stops                                       12676 non-
null   float64
 56  race_fastest_lap                                     27571 non-
null   object
 57  race_driver_of_the_day                               4601 non-
null    object
 58  race_grand_slam                                      27591 non-
null   object
 59  fastest_lap_lap                                      16689 non-
null   float64
 60  fastest_lap_time                                     16736 non-
null   object
 61  fastest_lap_time_millis                              16736 non-
null   float64
 62  fastest_lap_gap                                      15593 non-
null   object
 63  fastest_lap_gap_millis                               15593 non-
null   float64
 64  fastest_lap_interval                                 15593 non-
```

```
null   object
 65  fastest_lap_interval_millis                           15593 non-
null   float64
 66  pit_stop_stop                                         21889 non-
null   float64
 67  pit_stop_lap                                          21889 non-
null   float64
 68  pit_stop_time                                         21888 non-
null   object
 69  pit_stop_time_millis                                  21888 non-
null   float64
 70  driver_of_the_day_percentage                          720 non-
null     float64
dtypes: float64(34), int64(2), object(35)
memory usage: 99.5+ MB
```

```
race_data =
race_data.drop(columns=['position_display_order','position_number','dr
iver_number',

'engine_manufacturer_id','tyre_manufacturer_id','practice_time','pract
ice_time_millis',

'practice_gap','practice_gap_millis','qualifying_time_millis','qualify
ing_q1_millis',

'qualifying_q2_millis','qualifying_q3_millis','qualifying_interval',

'qualifying_interval_millis','starting_grid_position_qualification_pos
ition_number',

'starting_grid_position_time','starting_grid_position_time_millis','ra
ce_time_millis',

'race_gap','race_gap_millis','race_qualification_position_number',

'race_driver_of_the_day','fastest_lap_time_millis','fastest_lap_gap',

'fastest_lap_gap_millis','fastest_lap_interval','fastest_lap_interval_
millis',

'pit_stop_time_millis','driver_of_the_day_percentage'])
race_data = race_data.rename(columns={'position_text':'position'})
race_data
```

```
        race_id                      type position           driver_id
\
0           290     PRE_QUALIFYING_RESULT        1   gilles-villeneuve

1           290     PRE_QUALIFYING_RESULT        2      patrick-tambay
```

```
2            290     PRE_QUALIFYING_RESULT        3  jean-pierre-jarier
3            290     PRE_QUALIFYING_RESULT        4        brett-lunger
4            290     PRE_QUALIFYING_RESULT        5        brian-henton
...          ...                       ...      ...                 ...
183627      1143  DRIVER_OF_THE_DAY_RESULT        1     fernando-alonso
183628      1143  DRIVER_OF_THE_DAY_RESULT        2      george-russell
183629      1143  DRIVER_OF_THE_DAY_RESULT        3      max-verstappen
183630      1143  DRIVER_OF_THE_DAY_RESULT        4        lando-norris
183631      1143  DRIVER_OF_THE_DAY_RESULT        5      lewis-hamilton

       constructor_id practice_interval  practice_interval_millis  \
0             mclaren              None                       NaN   
1              ensign              None                       NaN   
2              penske              None                       NaN   
3             mclaren              None                       NaN   
4               march              None                       NaN   
...               ...               ...                       ...   
183627   aston-martin              None                       NaN   
183628       mercedes              None                       NaN   
183629       red-bull              None                       NaN   
183630        mclaren              None                       NaN   
183631        ferrari              None                       NaN   

        practice_laps qualifying_time qualifying_q1  ...  \
0                 NaN        1:19.480          None  ...   
1                 NaN        1:19.550          None  ...   
2                 NaN        1:19.630          None  ...   
3                 NaN        1:19.720          None  ...   
4                 NaN        1:19.820          None  ...   
...               ...             ...           ...  ...   
183627            NaN            None          None  ...   
183628            NaN            None          None  ...   
183629            NaN            None          None  ...   
183630            NaN            None          None  ...   
183631            NaN            None          None  ...   

       race_grid_position_text race_positions_gained race_pit_stops  \
0                         None                   NaN            NaN   
1                         None                   NaN            NaN   
2                         None                   NaN            NaN   
```

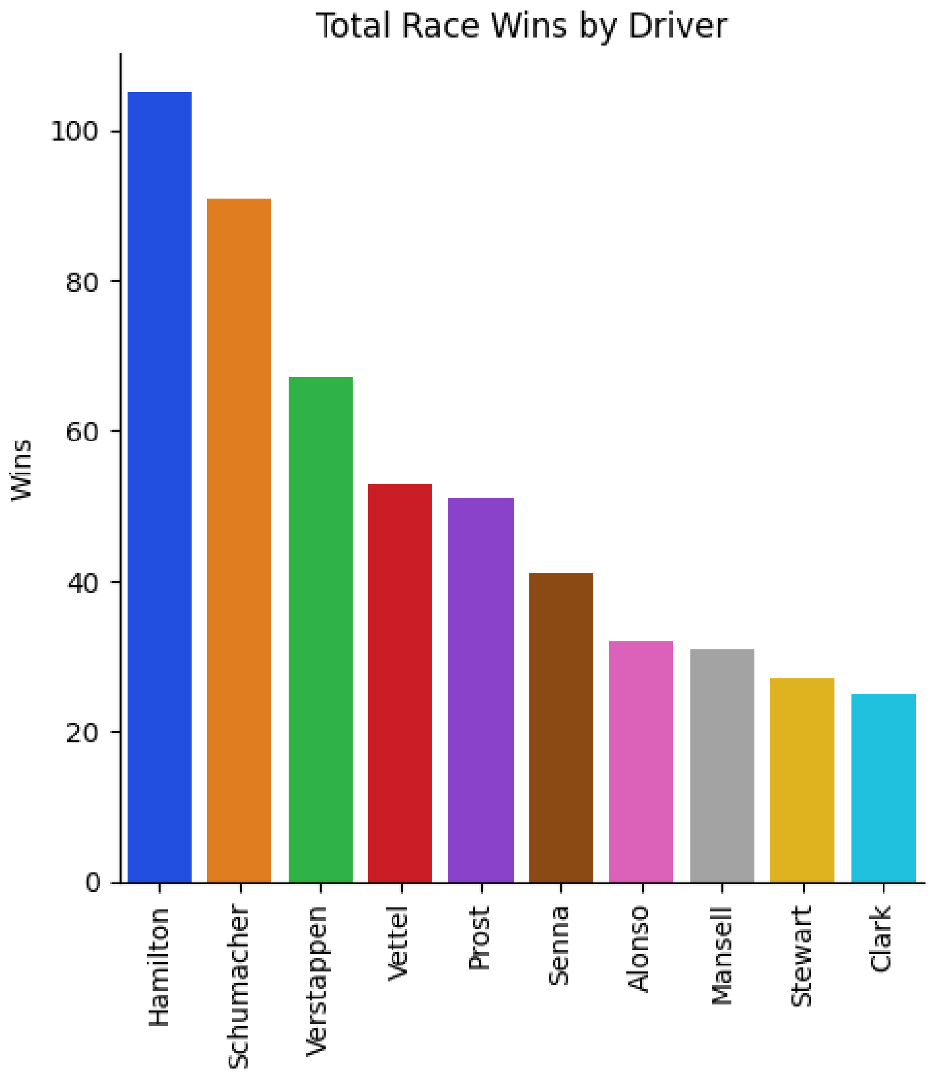
```
3                           None                   NaN            NaN
4                           None                   NaN            NaN
...                          ...                   ...            ...
183627                      None                   NaN            NaN
183628                      None                   NaN            NaN
183629                      None                   NaN            NaN
183630                      None                   NaN            NaN
183631                      None                   NaN            NaN

        race_fastest_lap  race_grand_slam fastest_lap_lap
fastest_lap_time  \
0                   None             None             NaN
None
1                   None             None             NaN
None
2                   None             None             NaN
None
3                   None             None             NaN
None
4                   None             None             NaN
None
...                  ...              ...             ...
...
183627              None             None             NaN
None
183628              None             None             NaN
None
183629              None             None             NaN
None
183630              None             None             NaN
None
183631              None             None             NaN
None

        pit_stop_stop pit_stop_lap  pit_stop_time
0                 NaN          NaN           None
1                 NaN          NaN           None
2                 NaN          NaN           None
3                 NaN          NaN           None
4                 NaN          NaN           None
...               ...          ...            ...
183627            NaN          NaN           None
183628            NaN          NaN           None
183629            NaN          NaN           None
183630            NaN          NaN           None
183631            NaN          NaN           None

[183632 rows x 41 columns]
```

Analyzing Drivers

```
driver['win_rank'] = driver.total_race_wins.rank(method='max',
ascending=False)
top_10_wins = driver.sort_values('win_rank').head(10)
top_10_wins

                     id               name first_name   last_name  \
558     lewis-hamilton     Lewis Hamilton      Lewis    Hamilton
619  michael-schumacher  Michael Schumacher    Michael  Schumacher
613     max-verstappen     Max Verstappen        Max  Verstappen
816    sebastian-vettel    Sebastian Vettel  Sebastian      Vettel
10          alain-prost        Alain Prost      Alain       Prost
70         ayrton-senna       Ayrton Senna     Ayrton       Senna
280     fernando-alonso    Fernando Alonso   Fernando      Alonso
659       nigel-mansell      Nigel Mansell      Nigel     Mansell
412      jackie-stewart     Jackie Stewart     Jackie     Stewart
448            jim-clark          Jim Clark        Jim       Clark

    abbreviation nationality_country_id  best_championship_position  \
558          HAM         united-kingdom                         1.0
619          MSC                germany                         1.0
613          VER            netherlands                         1.0
816          VET                germany                         1.0
10           PRO                 france                         1.0
70           SEN                 brazil                         1.0
280          ALO                  spain                         1.0
659          MAN         united-kingdom                         1.0
412          STE         united-kingdom                         1.0
448          CLA         united-kingdom                         1.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
558                          1.0               1.0
7
619                          1.0               1.0
7
613                          1.0               1.0
4
816                          1.0               1.0
4
10                           1.0               1.0
4
70                           1.0               1.0
3
280                          1.0               1.0
2
659                          1.0               1.0
1
412                          1.0               1.0
```

```
3
448                          1.0                1.0
2

     total_race_starts  total_race_wins  total_race_laps
total_podiums  \
558                374              105            21325
202
619                306               91            16825
155
613                227               67            12329
121
816                299               53            16426
122
10                 199               51            10540
106
70                 161               41             8219
80
280                420               32            22758
106
659                187               31             8750
59
412                 99               27             5225
43
448                 72               25             3877
32

     total_points  total_pole_positions  total_fastest_laps  \
558        4987.5                   104                  68
619        1566.0                    68                  77
613        3296.5                    46                  35
816        3098.0                    57                  38
10          798.5                    33                  41
70          614.0                    65                  19
280        2373.0                    22                  26
659         482.0                    32                  30
412         360.0                    17                  15
448         274.0                    33                  28

     total_grand_slams  win_rank
558                  6       1.0
619                  5       2.0
613                  6       3.0
816                  4       4.0
10                   0       5.0
70                   4       6.0
280                  0       7.0
659                  4       8.0
412                  4       9.0
448                  8      11.0
```

```
g = sns.catplot(data=top_10_wins, kind='bar', x='last_name',
y='total_race_wins', errorbar=None, hue='last_name')
for ax in g.axes.flat:
    ax.set_title('Total Race Wins by Driver')
    ax.set_xlabel('')
    ax.set_ylabel('Wins')
    ax.tick_params('x', labelrotation=90)
```



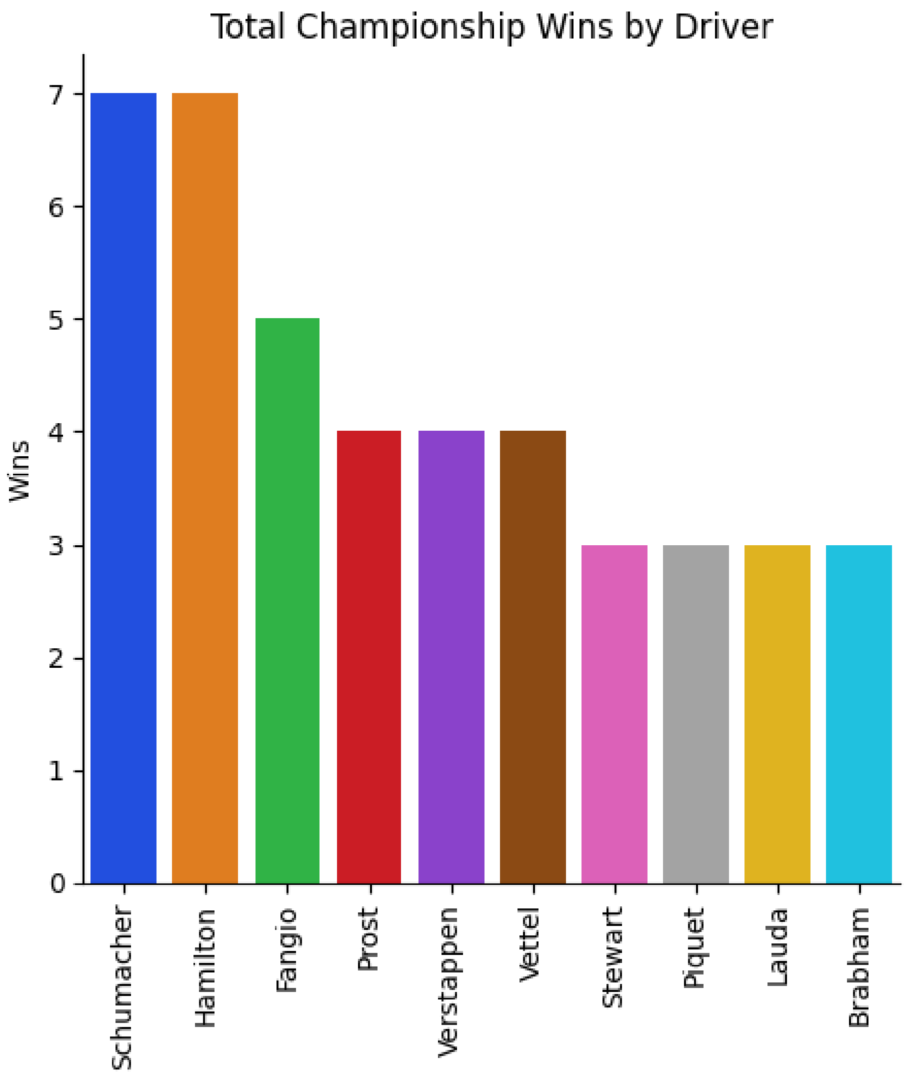
```
driver['championship_rank'] =
driver.total_championship_wins.rank(method='max', ascending=False)
top_10_champs = driver.sort_values('championship_rank').head(10)
top_10_champs
```

```
                     id                  name   first_name
last_name  \
619  michael-schumacher  Michael Schumacher      Michael   Schumacher
```

```
558      lewis-hamilton      Lewis Hamilton        Lewis    Hamilton

511  juan-manuel-fangio  Juan Manuel Fangio  Juan Manuel      Fangio

10          alain-prost         Alain Prost        Alain       Prost

613      max-verstappen      Max Verstappen          Max  Verstappen

816    sebastian-vettel    Sebastian Vettel    Sebastian      Vettel

412      jackie-stewart      Jackie Stewart       Jackie     Stewart

651        nelson-piquet       Nelson Piquet       Nelson      Piquet

660          niki-lauda          Niki Lauda         Niki       Lauda

403        jack-brabham        Jack Brabham         Jack     Brabham


    abbreviation nationality_country_id  best_championship_position  \
619          MSC                germany                         1.0
558          HAM         united-kingdom                         1.0
511          FAN              argentina                         1.0
10           PRO                 france                         1.0
613          VER            netherlands                         1.0
816          VET                germany                         1.0
412          STE         united-kingdom                         1.0
651          PIQ                 brazil                         1.0
660          LAU                austria                         1.0
403          BRA              australia                         1.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
619                          1.0               1.0
7
558                          1.0               1.0
7
511                          1.0               1.0
5
10                           1.0               1.0
4
613                          1.0               1.0
4
816                          1.0               1.0
4
412                          1.0               1.0
3
651                          1.0               1.0
3
```

```
660                          1.0               1.0
3
403                          1.0               1.0
3

     total_race_starts  total_race_wins  total_race_laps
total_podiums  \
619                306               91            16825
155
558                374              105            21325
202
511                 51               24             2960
35
10                 199               51            10540
106
613                227               67            12329
121
816                299               53            16426
122
412                 99               27             5225
43
651                203               23             9870
60
660                171               25             8213
54
403                126               14             6124
31

     total_points  total_pole_positions  total_fastest_laps  \
619       1566.00                    68                  77
558       4987.50                   104                  68
511        277.64                    29                  23
10         798.50                    33                  41
613       3296.50                    46                  35
816       3098.00                    57                  38
412        360.00                    17                  15
651        485.50                    24                  23
660        420.50                    24                  24
403        261.00                    13                  12

     total_grand_slams  win_rank  championship_rank
619                  5       2.0                2.0
558                  6       1.0                2.0
511                  0      12.0                3.0
10                   0       5.0                6.0
613                  6       3.0                6.0
816                  4       4.0                6.0
412                  4       9.0               11.0
651                  3      14.0               11.0
```

```
660                     0      11.0                   11.0
403                     0      22.0                   11.0

p = sns.catplot(data=top_10_champs, kind='bar', x='last_name',
y='total_championship_wins', errorbar=None, hue='last_name')
for ax in p.axes.flat:
    ax.set_title('Total Championship Wins by Driver')
    ax.set_xlabel('')
    ax.set_ylabel('Wins')
    ax.tick_params('x', labelrotation=90)
```



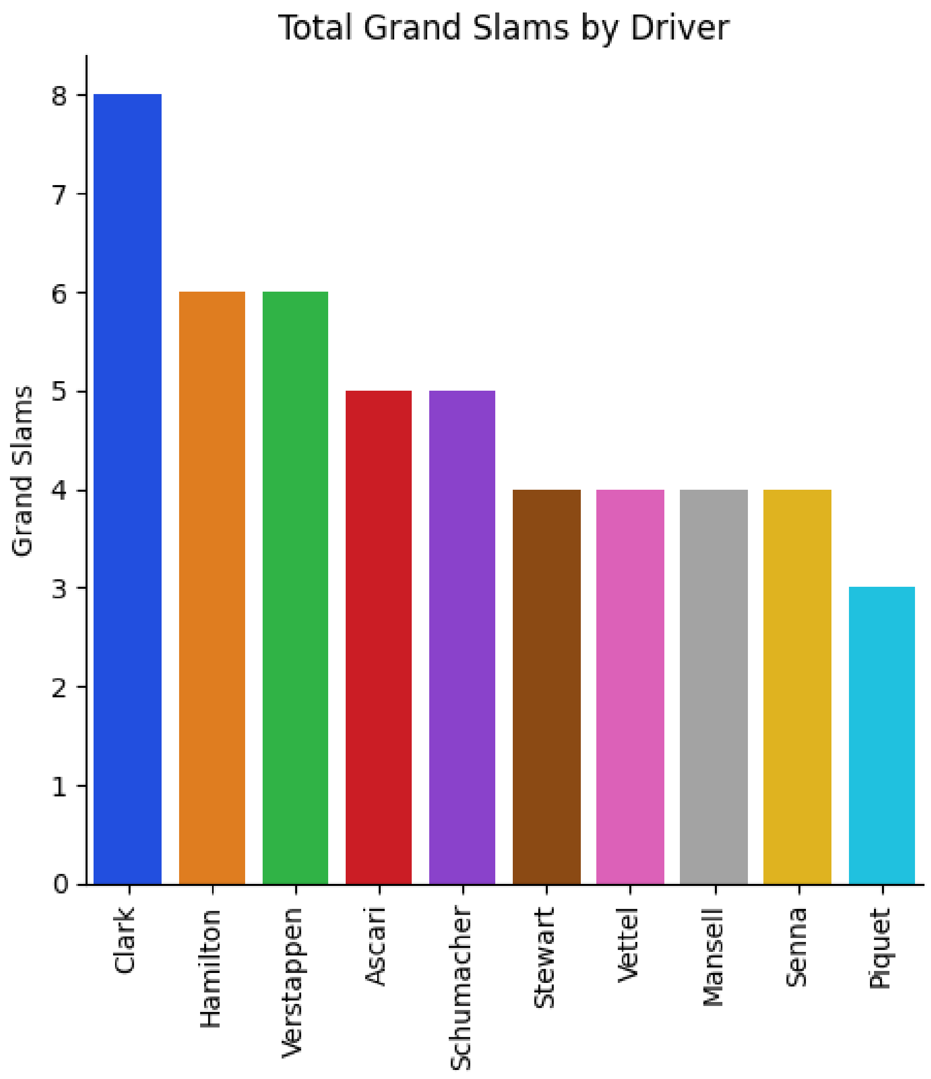
```
# This is grand slams (where the driver gets pole position, leads
every lap, sets the fastest lap, and wins the race)
driver['slam_rank'] = driver.total_grand_slams.rank(method='max',
ascending=False)
```

```
top_10_slams = driver.sort_values('slam_rank').head(10)
top_10_slams
```

```
                     id                name first_name   last_name  \
448           jim-clark           Jim Clark        Jim       Clark
558     lewis-hamilton      Lewis Hamilton      Lewis    Hamilton
613     max-verstappen      Max Verstappen        Max  Verstappen
17      alberto-ascari      Alberto Ascari    Alberto      Ascari
619  michael-schumacher  Michael Schumacher    Michael  Schumacher
412      jackie-stewart      Jackie Stewart     Jackie     Stewart
816    sebastian-vettel    Sebastian Vettel  Sebastian      Vettel
659       nigel-mansell       Nigel Mansell      Nigel     Mansell
70         ayrton-senna        Ayrton Senna     Ayrton       Senna
651        nelson-piquet       Nelson Piquet     Nelson      Piquet

    abbreviation nationality_country_id  best_championship_position  \
448          CLA         united-kingdom                         1.0
558          HAM         united-kingdom                         1.0
613          VER            netherlands                         1.0
17           ASC                  italy                         1.0
619          MSC                germany                         1.0
412          STE         united-kingdom                         1.0
816          VET                germany                         1.0
659          MAN         united-kingdom                         1.0
70           SEN                 brazil                         1.0
651          PIQ                 brazil                         1.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
448                          1.0               1.0
2
558                          1.0               1.0
7
613                          1.0               1.0
4
17                           1.0               1.0
2
619                          1.0               1.0
7
412                          1.0               1.0
3
816                          1.0               1.0
4
659                          1.0               1.0
1
70                           1.0               1.0
3
651                          1.0               1.0
3
```

```
     ...  total_race_wins  total_race_laps  total_podiums
total_points  \
448  ...               25             3877             32
274.00
558  ...              105            21325            202
4987.50
613  ...               67            12329            121
3296.50
17   ...               13             1609             17
140.14
619  ...               91            16825            155
1566.00
412  ...               27             5225             43
360.00
816  ...               53            16426            122
3098.00
659  ...               31             8750             59
482.00
70   ...               41             8219             80
614.00
651  ...               23             9870             60
485.50

     total_pole_positions  total_fastest_laps  total_grand_slams
win_rank  \
448                    33                  28                  8
11.0
558                   104                  68                  6
1.0
613                    46                  35                  6
3.0
17                     14                  13                  5
24.0
619                    68                  77                  5
2.0
412                    17                  15                  4
9.0
816                    57                  38                  4
4.0
659                    32                  30                  4
8.0
70                     65                  19                  4
6.0
651                    24                  23                  3
14.0

     championship_rank  slam_rank
448               17.0        1.0
558                2.0        3.0
613                6.0        3.0
```

```
17                  17.0        5.0
619                  2.0        5.0
412                 11.0        9.0
816                  6.0        9.0
659                 34.0        9.0
70                  11.0        9.0
651                 11.0       10.0

[10 rows x 21 columns]

w = sns.catplot(data=top_10_slams, kind='bar', x='last_name',
y='total_grand_slams', errorbar=None, hue='last_name')
for ax in w.axes.flat:
    ax.set_title('Total Grand Slams by Driver')
    ax.set_xlabel('')
    ax.set_ylabel('Grand Slams')
    ax.tick_params('x', labelrotation=90)
```



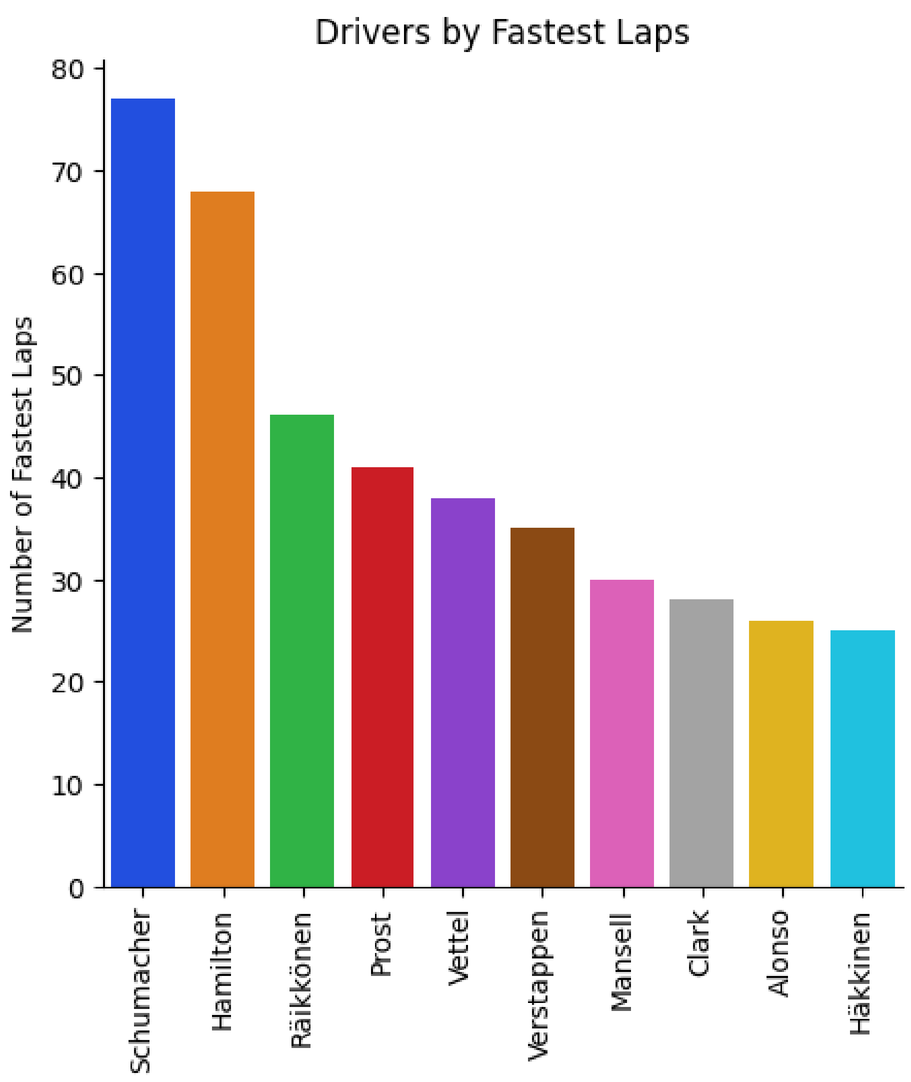
```
driver['fastest_lap_rank'] =
driver.total_fastest_laps.rank(method='max', ascending=False)
top_10_fast_laps = driver.sort_values('fastest_lap_rank').head(10)
top_10_fast_laps
```

```
                     id              name first_name   last_name  \
619  michael-schumacher  Michael Schumacher    Michael  Schumacher
558      lewis-hamilton      Lewis Hamilton      Lewis    Hamilton
537      kimi-raikkonen      Kimi Räikkönen       Kimi   Räikkönen
10          alain-prost         Alain Prost      Alain       Prost
816    sebastian-vettel    Sebastian Vettel  Sebastian      Vettel
613      max-verstappen      Max Verstappen        Max  Verstappen
659       nigel-mansell       Nigel Mansell      Nigel     Mansell
448           jim-clark           Jim Clark        Jim       Clark
280     fernando-alonso     Fernando Alonso   Fernando      Alonso
624        mika-hakkinen       Mika Häkkinen       Mika    Häkkinen

    abbreviation nationality_country_id  best_championship_position  \
619          MSC                germany                         1.0
558          HAM         united-kingdom                         1.0
537          RAI                finland                         1.0
10           PRO                 france                         1.0
816          VET                germany                         1.0
613          VER            netherlands                         1.0
659          MAN         united-kingdom                         1.0
448          CLA         united-kingdom                         1.0
280          ALO                  spain                         1.0
624          HAK                finland                         1.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
619                          1.0               1.0
7
558                          1.0               1.0
7
537                          1.0               1.0
1
10                           1.0               1.0
4
816                          1.0               1.0
4
613                          1.0               1.0
4
659                          1.0               1.0
1
448                          1.0               1.0
2
280                          1.0               1.0
2
624                          1.0               1.0
```

```
2

     ...  total_race_laps  total_podiums  total_points
total_pole_positions  \
619  ...            16825            155        1566.0
68
558  ...            21325            202        4987.5
104
537  ...            18621            103        1873.0
18
10   ...            10540            106         798.5
33
816  ...            16426            122        3098.0
57
613  ...            12329            121        3296.5
46
659  ...             8750             59         482.0
32
448  ...             3877             32         274.0
33
280  ...            22758            106        2373.0
22
624  ...             7719             51         420.0
26

     total_fastest_laps  total_grand_slams  win_rank
championship_rank  \
619                  77                  5       2.0
2.0
558                  68                  6       1.0
2.0
537                  46                  0      16.0
34.0
10                   41                  0       5.0
6.0
816                  38                  4       4.0
6.0
613                  35                  6       3.0
6.0
659                  30                  4       8.0
34.0
448                  28                  8      11.0
17.0
280                  26                  0       7.0
17.0
624                  25                  0      17.0
17.0

     slam_rank  fastest_lap_rank
619        5.0               1.0
```

```
558       3.0                 2.0
537     912.0                 3.0
10      912.0                 4.0
816       9.0                 5.0
613       3.0                 6.0
659       9.0                 7.0
448       1.0                 8.0
280     912.0                 9.0
624     912.0                10.0

[10 rows x 22 columns]
```

```
f = sns.catplot(data=top_10_fast_laps, kind='bar', x='last_name',
y='total_fastest_laps', errorbar=None, hue='last_name')
for ax in f.axes.flat:
    ax.set_title('Drivers by Fastest Laps')
    ax.set_xlabel('')
    ax.set_ylabel('Number of Fastest Laps')
    ax.tick_params('x', labelrotation=90)
```

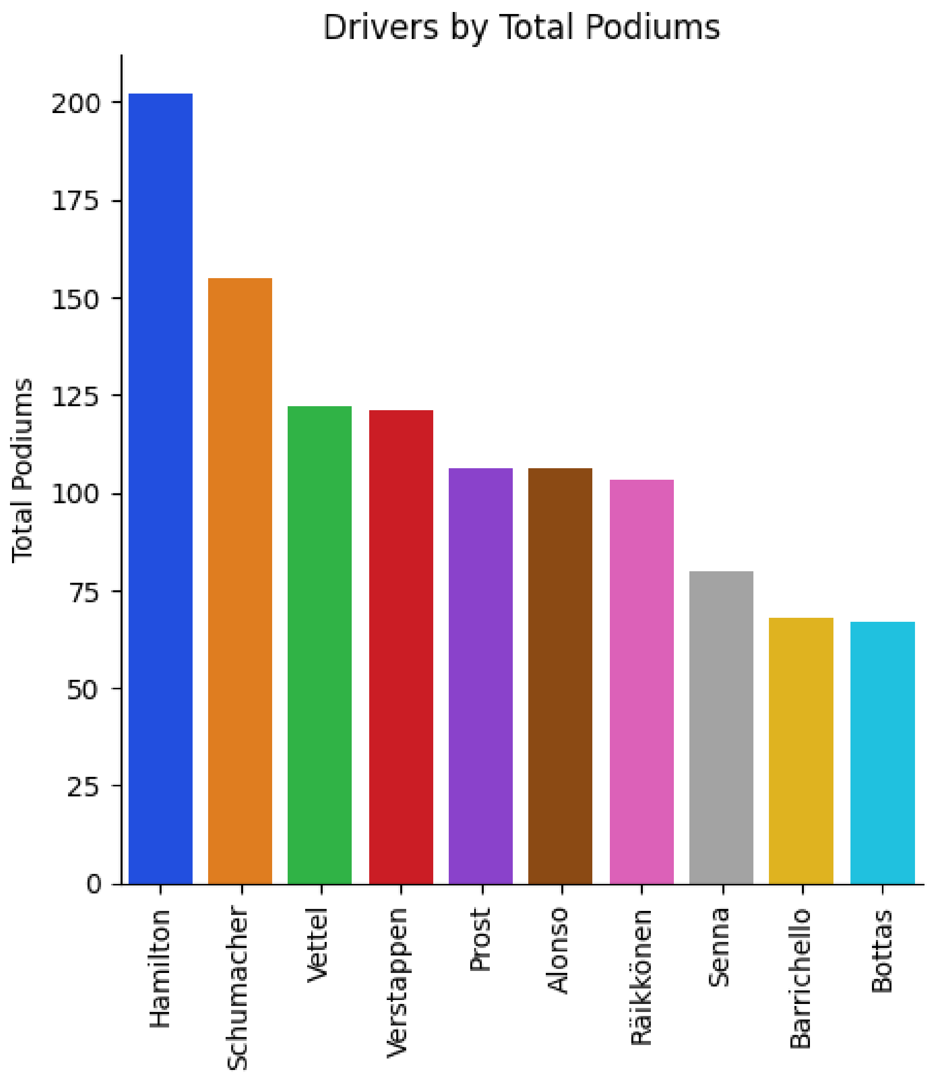


```python
# This is by podiums which means first, second, or third
driver['podium_rank'] = driver.total_podiums.rank(method='max',
ascending=False)
top_10_podiums = driver.sort_values('podium_rank').head(10)
top_10_podiums
```

```
                     id                name first_name    last_name  \
558      lewis-hamilton      Lewis Hamilton      Lewis     Hamilton
619  michael-schumacher  Michael Schumacher    Michael   Schumacher
816    sebastian-vettel    Sebastian Vettel  Sebastian       Vettel
613      max-verstappen      Max Verstappen        Max   Verstappen
10          alain-prost         Alain Prost      Alain        Prost
280     fernando-alonso     Fernando Alonso   Fernando       Alonso
537      kimi-raikkonen      Kimi Räikkönen       Kimi    Räikkönen
70         ayrton-senna        Ayrton Senna     Ayrton        Senna
801  rubens-barrichello  Rubens Barrichello     Rubens  Barrichello
881     valtteri-bottas     Valtteri Bottas   Valtteri       Bottas
```

```
    abbreviation nationality_country_id  best_championship_position  \
558          HAM         united-kingdom                         1.0
619          MSC                germany                         1.0
816          VET                germany                         1.0
613          VER            netherlands                         1.0
10           PRO                 france                         1.0
280          ALO                  spain                         1.0
537          RAI                finland                         1.0
70           SEN                 brazil                         1.0
801          BAR                 brazil                         2.0
881          BOT                finland                         2.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
558                          1.0               1.0
7
619                          1.0               1.0
7
816                          1.0               1.0
4
613                          1.0               1.0
4
10                           1.0               1.0
4
280                          1.0               1.0
2
537                          1.0               1.0
1
70                           1.0               1.0
3
801                          1.0               1.0
0
881                          1.0               1.0
0

     ...  total_podiums  total_points  total_pole_positions  \
558  ...            202        4987.5                   104
619  ...            155        1566.0                    68
816  ...            122        3098.0                    57
613  ...            121        3296.5                    46
10   ...            106         798.5                    33
280  ...            106        2373.0                    22
537  ...            103        1873.0                    18
70   ...             80         614.0                    65
801  ...             68         658.0                    14
881  ...             67        1797.0                    20

     total_fastest_laps  total_grand_slams  win_rank
championship_rank  \
558                     68                    6         1.0
2.0
619                     77                    5         2.0
2.0
816                     38                    4         4.0
6.0
613                     35                    6         3.0
6.0
10                      41                    0         5.0
6.0
280                     26                    0         7.0
17.0
537                     46                    0        16.0
34.0
70                      19                    4         6.0
11.0
801                     17                    0        30.0
912.0
881                     19                    0        35.0
912.0

     slam_rank  fastest_lap_rank  podium_rank
558        3.0               2.0          1.0
619        5.0               1.0          2.0
816        9.0               5.0          3.0
613        3.0               6.0          4.0
10       912.0               4.0          6.0
280      912.0               9.0          6.0
537      912.0               3.0          7.0
70         9.0              20.0          8.0
801      912.0              24.0          9.0
881      912.0              20.0         10.0

[10 rows x 23 columns]
```

```python
po = sns.catplot(data=top_10_podiums, kind='bar', x='last_name',
y='total_podiums', errorbar=None, hue='last_name')
for ax in po.axes.flat:
    ax.set_title('Drivers by Total Podiums')
    ax.set_xlabel('')
    ax.set_ylabel('Total Podiums')
    ax.tick_params('x', labelrotation=90)
```

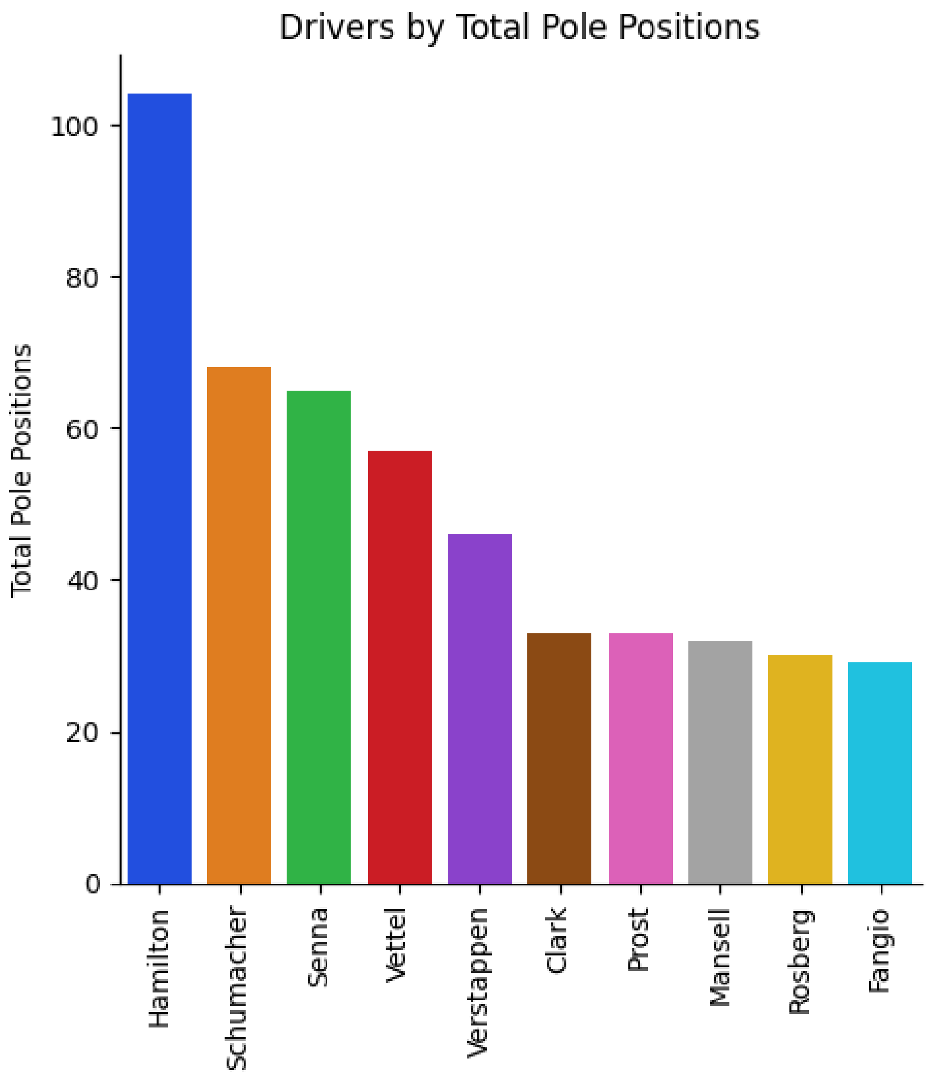


```
driver['pole_position_rank'] =
driver.total_pole_positions.rank(method='max', ascending=False)
top_10_pole_positions =
driver.sort_values('pole_position_rank').head(10)
top_10_pole_positions

                     id                name  first_name
last_name  \
558      lewis-hamilton      Lewis Hamilton       Lewis    Hamilton

619  michael-schumacher  Michael Schumacher     Michael  Schumacher

70         ayrton-senna        Ayrton Senna      Ayrton       Senna

816    sebastian-vettel    Sebastian Vettel   Sebastian      Vettel

613      max-verstappen      Max Verstappen         Max  Verstappen
```

```
448             jim-clark           Jim Clark          Jim       Clark

10            alain-prost         Alain Prost        Alain       Prost

659         nigel-mansell       Nigel Mansell        Nigel     Mansell

656           nico-rosberg        Nico Rosberg         Nico     Rosberg

511  juan-manuel-fangio  Juan Manuel Fangio  Juan Manuel      Fangio


    abbreviation nationality_country_id  best_championship_position  \
558          HAM         united-kingdom                         1.0
619          MSC                germany                         1.0
70           SEN                 brazil                         1.0
816          VET                germany                         1.0
613          VER            netherlands                         1.0
448          CLA         united-kingdom                         1.0
10           PRO                 france                         1.0
659          MAN         united-kingdom                         1.0
656          ROS                germany                         1.0
511          FAN              argentina                         1.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
558                          1.0               1.0
7
619                          1.0               1.0
7
70                           1.0               1.0
3
816                          1.0               1.0
4
613                          1.0               1.0
4
448                          1.0               1.0
2
10                           1.0               1.0
4
659                          1.0               1.0
1
656                          1.0               1.0
1
511                          1.0               1.0
5

     ...  total_points  total_pole_positions  total_fastest_laps  \
558  ...       4987.50                   104                  68
619  ...       1566.00                    68                  77
```

```
70   ...       614.00                   65                   19
816  ...      3098.00                   57                   38
613  ...      3296.50                   46                   35
448  ...       274.00                   33                   28
10   ...       798.50                   33                   41
659  ...       482.00                   32                   30
656  ...      1594.50                   30                   20
511  ...       277.64                   29                   23

     total_grand_slams  win_rank  championship_rank  slam_rank  \
558                  6       1.0                2.0        3.0
619                  5       2.0                2.0        5.0
70                   4       6.0               11.0        9.0
816                  4       4.0                6.0        9.0
613                  6       3.0                6.0        3.0
448                  8      11.0               17.0        1.0
10                   0       5.0                6.0      912.0
659                  4       8.0               34.0        9.0
656                  0      14.0               34.0      912.0
511                  0      12.0                3.0      912.0

     fastest_lap_rank  podium_rank  pole_position_rank
558               2.0          1.0                 1.0
619               1.0          2.0                 2.0
70               20.0          8.0                 3.0
816               5.0          3.0                 4.0
613               6.0          4.0                 5.0
448               8.0         36.0                 7.0
10                4.0          6.0                 7.0
659               7.0         13.0                 8.0
656              15.0         14.0                 9.0
511              13.0         30.0                10.0

[10 rows x 24 columns]
```

```
pole = sns.catplot(data=top_10_pole_positions, kind='bar',
x='last_name', y='total_pole_positions', errorbar=None,
                  hue='last_name')
for ax in pole.axes.flat:
    ax.set_title('Drivers by Total Pole Positions')
    ax.set_xlabel('')
    ax.set_ylabel('Total Pole Positions')
    ax.tick_params('x', labelrotation=90)
```



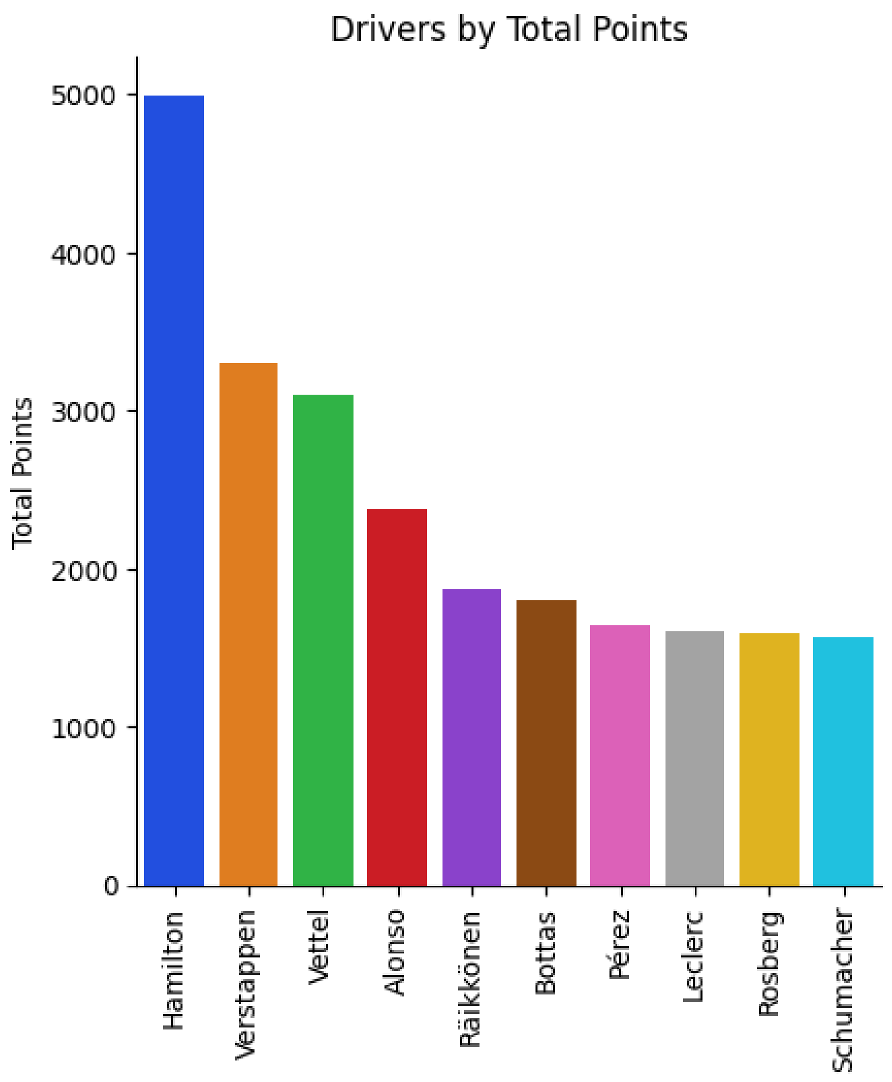
```
driver['points_rank'] = driver.total_points.rank(method='max',
ascending=False)
top_10_points = driver.sort_values('points_rank').head(10)
top_10_points
```

```
                     id               name first_name   last_name  \
558      lewis-hamilton     Lewis Hamilton      Lewis    Hamilton
613      max-verstappen     Max Verstappen        Max  Verstappen
816    sebastian-vettel   Sebastian Vettel  Sebastian      Vettel
280     fernando-alonso    Fernando Alonso   Fernando      Alonso
537      kimi-raikkonen     Kimi Räikkönen       Kimi   Räikkönen
881     valtteri-bottas    Valtteri Bottas   Valtteri      Bottas
821        sergio-perez       Sergio Pérez     Sergio       Pérez
145     charles-leclerc    Charles Leclerc    Charles     Leclerc
656         nico-rosberg       Nico Rosberg       Nico     Rosberg
619  michael-schumacher  Michael Schumacher    Michael  Schumacher
```

```
    abbreviation nationality_country_id  best_championship_position  \
558          HAM         united-kingdom                         1.0   
613          VER            netherlands                         1.0   
816          VET                germany                         1.0   
280          ALO                  spain                         1.0   
537          RAI                finland                         1.0   
881          BOT                finland                         2.0   
821          PER                 mexico                         2.0   
145          LEC                 monaco                         2.0   
656          ROS                germany                         1.0   
619          MSC                germany                         1.0   

     best_starting_grid_position  best_race_result  
total_championship_wins  \
558                          1.0               1.0  
7   
613                          1.0               1.0  
4   
816                          1.0               1.0  
4   
280                          1.0               1.0  
2   
537                          1.0               1.0  
1   
881                          1.0               1.0  
0   
821                          1.0               1.0  
0   
145                          1.0               1.0  
0   
656                          1.0               1.0  
1   
619                          1.0               1.0  
7   

     ...  total_pole_positions  total_fastest_laps  total_grand_slams  
\
558  ...                   104                  68                  6   

613  ...                    46                  35                  6   

816  ...                    57                  38                  4   

280  ...                    22                  26                  0   

537  ...                    18                  46                  0   

881  ...                    20                  19                  0   

821  ...                     3                  12                  0   
```

```
145  ...                   27                  10                  0

656  ...                   30                  20                  0

619  ...                   68                  77                  5


     win_rank  championship_rank  slam_rank  fastest_lap_rank
podium_rank  \
558       1.0                2.0        3.0               2.0
1.0
613       3.0                6.0        3.0               6.0
4.0
816       4.0                6.0        9.0               5.0
3.0
280       7.0               17.0      912.0               9.0
6.0
537      16.0               34.0      912.0               3.0
7.0
881      35.0              912.0      912.0              20.0
10.0
821      52.0              912.0      912.0              35.0
26.0
145      42.0              912.0      912.0              40.0
19.0
656      14.0               34.0      912.0              15.0
14.0
619       2.0                2.0        5.0               1.0
2.0

     pole_position_rank  points_rank
558                 1.0          1.0
613                 5.0          2.0
816                 4.0          3.0
280                15.0          4.0
537                20.0          5.0
881                17.0          6.0
821                68.0          7.0
145                11.0          8.0
656                 9.0          9.0
619                 2.0         10.0

[10 rows x 25 columns]
```

```python
point = sns.catplot(data=top_10_points, kind='bar', x='last_name',
y='total_points', errorbar=None, hue='last_name')
for ax in point.axes.flat:
    ax.set_title('Drivers by Total Points')
    ax.set_xlabel('')
    ax.set_ylabel('Total Points')
    ax.tick_params('x', labelrotation=90)
```



Amount of Seasons and Races Driver has Competed in

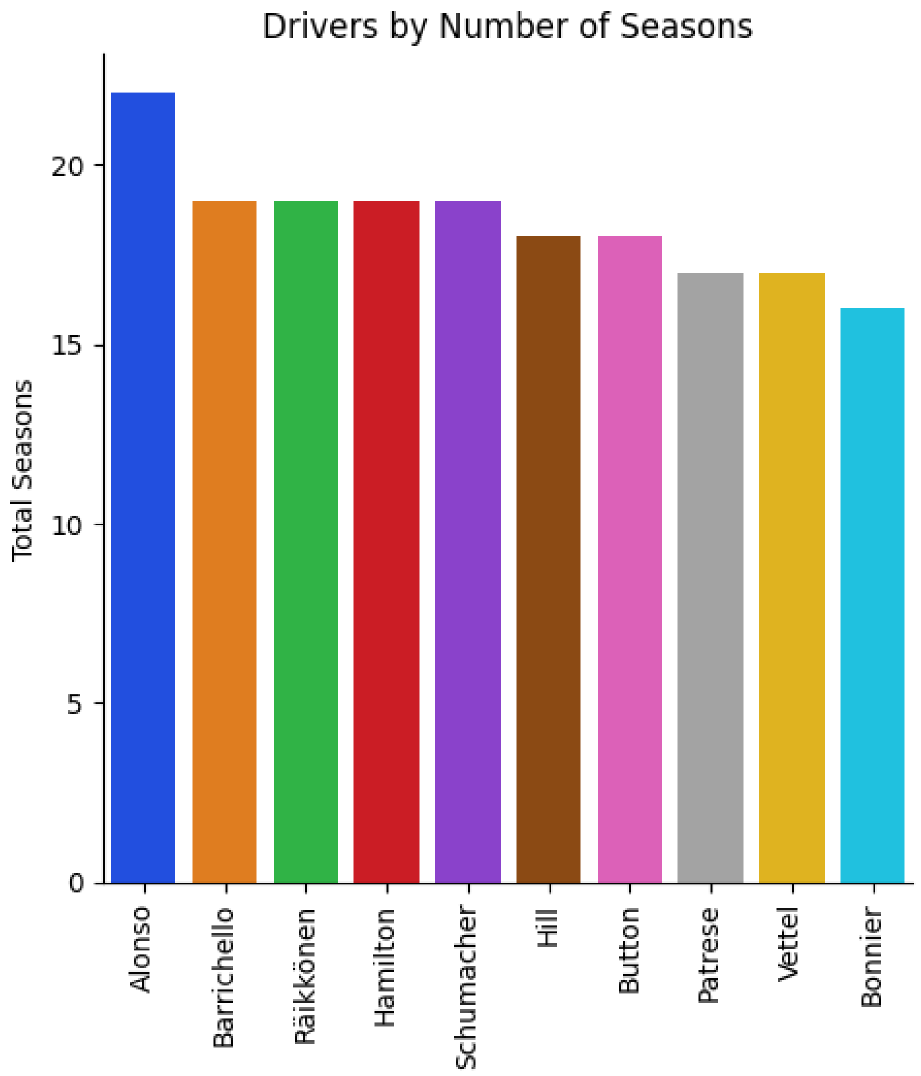
```
season_group = driver_by_season.groupby('driver_id')
driver_season = season_group[['year']].count()
driver_season = driver_season.reset_index()
driver_season =
driver_season.rename(columns={'year':'seasons','driver_id':'id'})
driver_season

                        id  seasons
0             adderly-fong        1
1             adolf-brudes        1
2      adolfo-schwelm-cruz        1
```

```
3           adrian-campos        2
4            adrian-sutil        8
..                    ...      ...
907              yuji-ide        1
908           yuki-tsunoda       5
909  yves-giraud-cabantous       4
910          zak-osullivan       1
911      zsolt-baumgartner       2

[912 rows x 2 columns]
```

```
# Adding the driver names to the driver_season DataFrame
driver_names = driver[['name','first_name','last_name','id']]
driver_merged = driver_season.merge(driver_names, on='id')
driver_merged
```

```
                        id  seasons                   name first_name  \
0             adderly-fong        1           Adderly Fong    Adderly   
1             adolf-brudes        1           Adolf Brudes      Adolf   
2      adolfo-schwelm-cruz        1    Adolfo Schwelm Cruz     Adolfo   
3            adrian-campos        2         Adrián Campos     Adrián   
4             adrian-sutil        8           Adrian Sutil     Adrian   
..                     ...      ...                    ...        ...   
907               yuji-ide        1               Yuji Ide       Yuji   
908           yuki-tsunoda        5           Yuki Tsunoda       Yuki   
909  yves-giraud-cabantous        4  Yves Giraud-Cabantous       Yves   
910          zak-osullivan        1          Zak O'Sullivan        Zak   
911      zsolt-baumgartner        2      Zsolt Baumgartner      Zsolt   

             last_name  
0                 Fong  
1               Brudes  
2         Schwelm Cruz  
3               Campos  
4                Sutil  
..                 ...  
907                Ide  
908            Tsunoda  
909   Giraud-Cabantous  
```

```
910        O'Sullivan
911       Baumgartner

[912 rows x 5 columns]

driver_merged['seasons_ranked'] =
driver_merged.seasons.rank(method='max', ascending=False)
top_10_seasons = driver_merged.sort_values('seasons_ranked').head(10)
top_10_seasons

                     id  seasons                name first_name
last_name  \
280     fernando-alonso       22     Fernando Alonso   Fernando
Alonso
801  rubens-barrichello       19  Rubens Barrichello     Rubens
Barrichello
536      kimi-raikkonen       19       Kimi Räikkönen       Kimi
Räikkönen
557      lewis-hamilton       19      Lewis Hamilton      Lewis
Hamilton
619  michael-schumacher       19  Michael Schumacher    Michael
Schumacher
345         graham-hill       18         Graham Hill     Graham
Hill
442        jenson-button       18       Jenson Button     Jenson
Button
761    riccardo-patrese       17    Riccardo Patrese   Riccardo
Patrese
816    sebastian-vettel       17    Sebastian Vettel  Sebastian
Vettel
461          jo-bonnier       16          Jo Bonnier         Jo
Bonnier

     seasons_ranked
280             1.0
801             5.0
536             5.0
557             5.0
619             5.0
345             7.0
442             7.0
761             9.0
816             9.0
461            11.0

se = sns.catplot(data=top_10_seasons, kind='bar', x='last_name',
y='seasons', errorbar=None, hue='last_name')
for ax in se.axes.flat:
    ax.set_title('Drivers by Number of Seasons')
    ax.set_xlabel('')
    ax.set_ylabel('Total Seasons')
    ax.tick_params('x', labelrotation=90)
```



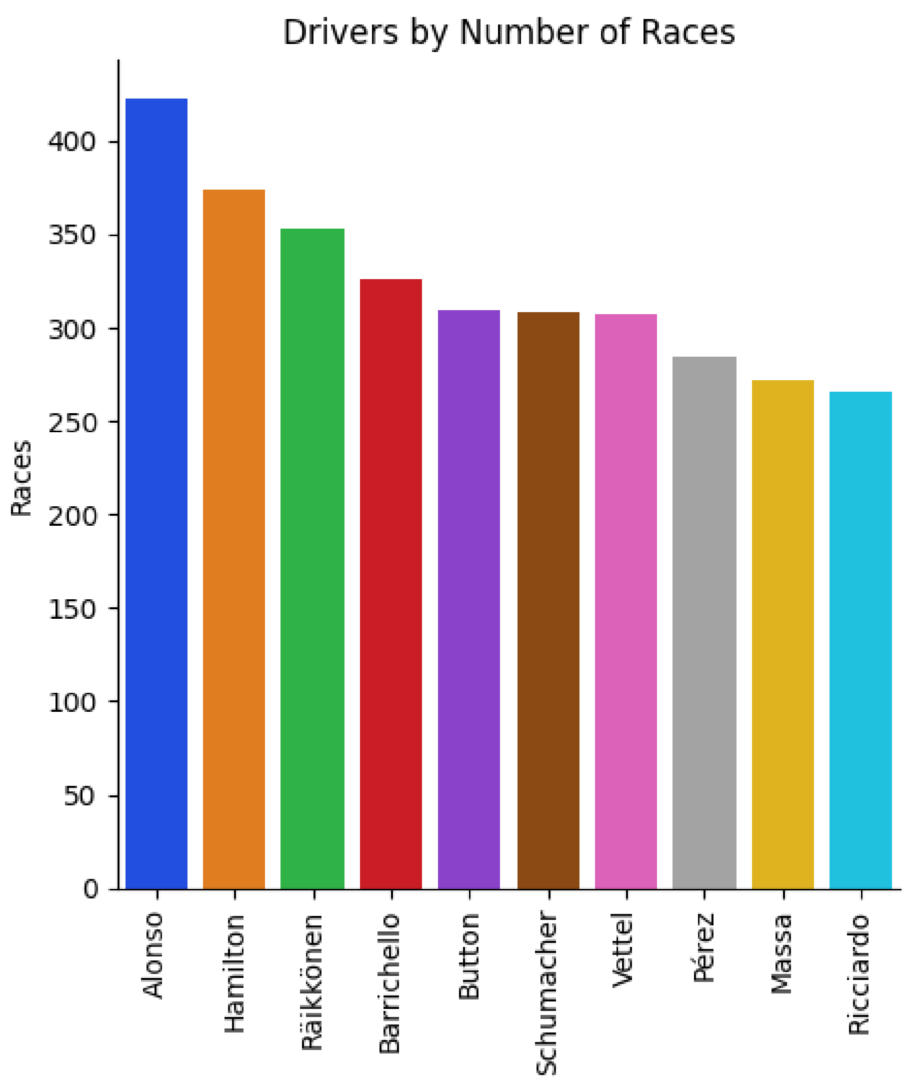
It is worth noting that Alonso and Hamilton are current drivers.

```
races_grouped = race_data.groupby('driver_id')
career_races = races_grouped[['race_id']].nunique()
career_races = career_races.reset_index()
career_races =
career_races.rename(columns={'driver_id':'id','race_id':'races'})
career_races

                        id  races
0             adderly-fong      1
1             adolf-brudes      1
2      adolfo-schwelm-cruz      1
```

```
3            adrian-campos     21
4             adrian-sutil    131
..                     ...    ...
904               yuji-ide      4
905            yuki-tsunoda   108
906  yves-giraud-cabantous     13
907          zak-osullivan      1
908       zsolt-baumgartner    20

[909 rows x 2 columns]

races_merged = career_races.merge(driver_names, on='id')
races_merged

                        id  races                   name first_name  \
0             adderly-fong      1           Adderly Fong    Adderly
1             adolf-brudes      1           Adolf Brudes      Adolf
2       adolfo-schwelm-cruz      1    Adolfo Schwelm Cruz     Adolfo
3            adrian-campos     21          Adrián Campos     Adrián
4             adrian-sutil    131            Adrian Sutil     Adrian
..                     ...    ...                    ...        ...
904               yuji-ide      4               Yuji Ide       Yuji
905            yuki-tsunoda   108           Yuki Tsunoda       Yuki
906  yves-giraud-cabantous     13  Yves Giraud-Cabantous       Yves
907          zak-osullivan      1          Zak O'Sullivan        Zak
908       zsolt-baumgartner    20       Zsolt Baumgartner      Zsolt

            last_name
0                Fong
1              Brudes
2         Schwelm Cruz
3              Campos
4               Sutil
..                ...
904               Ide
905           Tsunoda
906  Giraud-Cabantous
907         O'Sullivan
908        Baumgartner

[909 rows x 5 columns]

races_merged['races_rank'] = races_merged.races.rank(method='max',
ascending=False)
top_10_races = races_merged.sort_values('races_rank').head(10)
top_10_races

                     id  races                name first_name
last_name  \
279     fernando-alonso    422     Fernando Alonso   Fernando
```

```
Alonso
556      lewis-hamilton    374       Lewis Hamilton       Lewis
Hamilton
535      kimi-raikkonen    353       Kimi Räikkönen        Kimi
Räikkönen
799  rubens-barrichello    326  Rubens Barrichello      Rubens
Barrichello
441       jenson-button    309        Jenson Button      Jenson
Button
617  michael-schumacher    308  Michael Schumacher     Michael
Schumacher
813    sebastian-vettel    307     Sebastian Vettel   Sebastian
Vettel
818        sergio-perez    284         Sergio Pérez      Sergio
Pérez
277        felipe-massa    272         Felipe Massa      Felipe
Massa
184    daniel-ricciardo    266    Daniel Ricciardo      Daniel
Ricciardo

     races_rank
279         1.0
556         2.0
535         3.0
799         4.0
441         5.0
617         6.0
813         7.0
818         8.0
277         9.0
184        10.0
```

```
re = sns.catplot(data=top_10_races, kind='bar', x='last_name',
y='races', errorbar=None, hue='last_name')
for ax in re.axes.flat:
    ax.set_title('Drivers by Number of Races')
    ax.set_xlabel('')
    ax.set_ylabel('Races')
    ax.tick_params('x', labelrotation=90)
```



Top 10 Drivers Analysis

I chose our top 10 drivers based on how many times they appeared in the top 10 of the previous statistics. Lewis Hamilton, Michael Schumacher, and Sebastian Vettel with 9 each. Max Verstappen with 7. Fernando Alonso with 6. Alain Prost and Kimi Räikkönen with 5 each. Aryton Senna, Nigel Mansell, and Jim Clark with 4 each.

```
# Reclean Driver
driver_cleaned =
driver.drop(columns=['win_rank','championship_rank','slam_rank','faste
st_lap_rank','podium_rank',

'pole_position_rank','points_rank'])

# new_driver_name DataFrame
new_driver_name = driver_cleaned.query('id == "lewis-hamilton" or id
== "michael-schumacher" or id == "sebastian-vettel" or \
id == "max-verstappen" or id == "fernando-alonso" or id == "alain-
prost" or id == "kimi-raikkonen" or id == "ayrton-senna" \
or id == "nigel-mansell" or id == "jim-clark"')
new_driver_name

                     id                name first_name   last_name  \
10          alain-prost         Alain Prost      Alain       Prost
70         ayrton-senna        Ayrton Senna     Ayrton       Senna
280     fernando-alonso     Fernando Alonso   Fernando      Alonso
448           jim-clark           Jim Clark        Jim       Clark
537      kimi-raikkonen      Kimi Räikkönen       Kimi   Räikkönen
558      lewis-hamilton      Lewis Hamilton      Lewis    Hamilton
613      max-verstappen      Max Verstappen        Max  Verstappen
619  michael-schumacher  Michael Schumacher    Michael  Schumacher
659       nigel-mansell       Nigel Mansell      Nigel     Mansell
816    sebastian-vettel    Sebastian Vettel  Sebastian      Vettel

    abbreviation nationality_country_id  best_championship_position  \
10           PRO                 france                         1.0
70           SEN                 brazil                         1.0
280          ALO                  spain                         1.0
448          CLA         united-kingdom                         1.0
537          RAI                finland                         1.0
558          HAM         united-kingdom                         1.0
613          VER            netherlands                         1.0
619          MSC                germany                         1.0
659          MAN         united-kingdom                         1.0
816          VET                germany                         1.0

     best_starting_grid_position  best_race_result
total_championship_wins  \
10                           1.0               1.0
4
70                           1.0               1.0
3
280                          1.0               1.0
2
448                          1.0               1.0
2
537                          1.0               1.0
1
558                          1.0               1.0
7
613                          1.0               1.0
4
619                          1.0               1.0
7
659                          1.0               1.0
1
```

```
816                          1.0                1.0
4

     total_race_starts  total_race_wins  total_race_laps
total_podiums  \
10                 199               51            10540
106
70                 161               41             8219
80
280                420               32            22758
106
448                 72               25             3877
32
537                350               21            18621
103
558                374              105            21325
202
613                227               67            12329
121
619                306               91            16825
155
659                187               31             8750
59
816                299               53            16426
122

     total_points  total_pole_positions  total_fastest_laps
total_grand_slams
10          798.5                    33                  41
0
70          614.0                    65                  19
4
280        2373.0                    22                  26
0
448         274.0                    33                  28
8
537        1873.0                    18                  46
0
558        4987.5                   104                  68
6
613        3296.5                    46                  35
6
619        1566.0                    68                  77
5
659         482.0                    32                  30
4
816        3098.0                    57                  38
4
```

```
# new_seasons DataFrame
seasons_cleaned =
driver_merged.drop(columns=['name','first_name','last_name'])
new_seasons = seasons_cleaned.query('id == "lewis-hamilton" or id ==
"michael-schumacher" or id == "sebastian-vettel" or \
id == "max-verstappen" or id == "fernando-alonso" or id == "alain-
prost" or id == "kimi-raikkonen" or id == "ayrton-senna" \
or id == "nigel-mansell" or id == "jim-clark"')
new_seasons = new_seasons.drop(columns=['seasons_ranked'])
new_seasons
```

```
                     id  seasons
10          alain-prost       13
70         ayrton-senna       11
280     fernando-alonso       22
447           jim-clark        9
536      kimi-raikkonen       19
557      lewis-hamilton       19
613      max-verstappen       12
619  michael-schumacher       19
659       nigel-mansell       15
816    sebastian-vettel       17
```

```
# new_races DataFrame
races_cleaned =
races_merged.drop(columns=['name','first_name','last_name','races_rank
'])
new_races = races_cleaned.query('id == "lewis-hamilton" or id ==
"michael-schumacher" or id == "sebastian-vettel" or \
id == "max-verstappen" or id == "fernando-alonso" or id == "alain-
prost" or id == "kimi-raikkonen" or id == "ayrton-senna" \
or id == "nigel-mansell" or id == "jim-clark"')
new_races
```

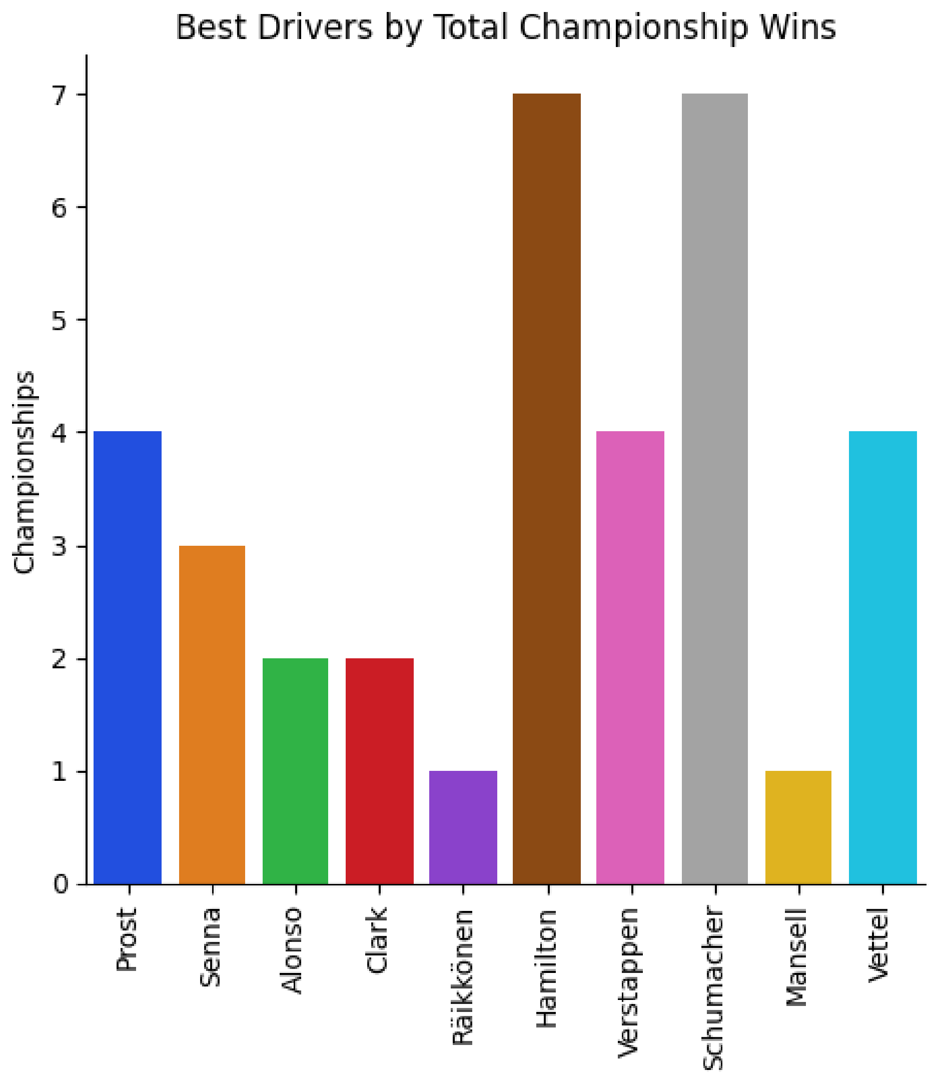
```
                     id  races
10          alain-prost    202
70         ayrton-senna    162
279     fernando-alonso    422
446           jim-clark     73
535      kimi-raikkonen    353
556      lewis-hamilton    374
611      max-verstappen    230
617  michael-schumacher    308
657       nigel-mansell    191
813    sebastian-vettel    307
```

```
# top_10_drivers DataFame
name_and_seasons = new_driver_name.merge(new_seasons, on='id')
top_10_drivers = name_and_seasons.merge(new_races, on='id')
top_10_drivers
```

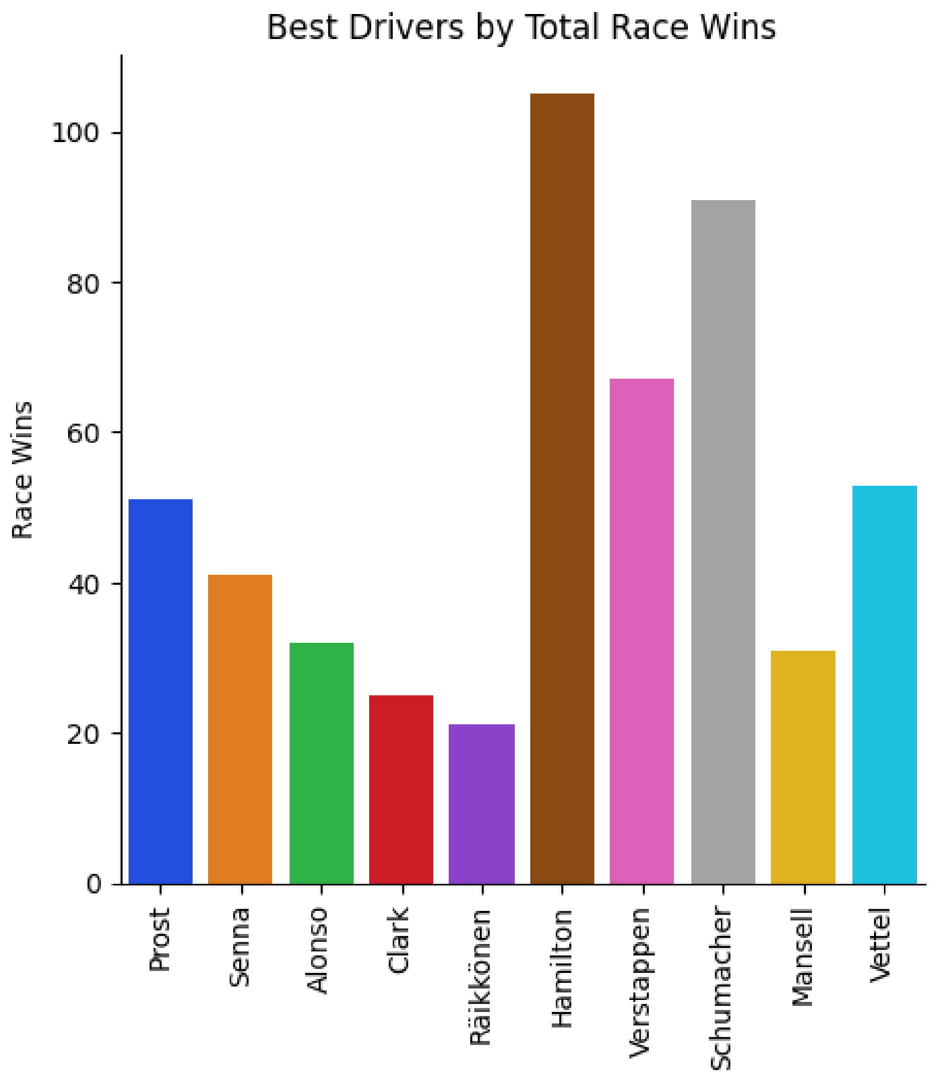
```
                   id                name first_name   last_name
abbreviation  \
0         alain-prost         Alain Prost      Alain       Prost
PRO
1        ayrton-senna        Ayrton Senna     Ayrton       Senna
SEN
2     fernando-alonso     Fernando Alonso   Fernando      Alonso
ALO
3           jim-clark           Jim Clark        Jim       Clark
CLA
4      kimi-raikkonen      Kimi Räikkönen       Kimi   Räikkönen
RAI
5      lewis-hamilton      Lewis Hamilton      Lewis    Hamilton
HAM
6      max-verstappen      Max Verstappen        Max  Verstappen
VER
7  michael-schumacher  Michael Schumacher    Michael  Schumacher
MSC
8        nigel-mansell        Nigel Mansell      Nigel     Mansell
MAN
9    sebastian-vettel    Sebastian Vettel  Sebastian      Vettel
VET

  nationality_country_id  best_championship_position  \
0                 france                         1.0
1                 brazil                         1.0
2                  spain                         1.0
3         united-kingdom                         1.0
4                finland                         1.0
5         united-kingdom                         1.0
6            netherlands                         1.0
7                germany                         1.0
8         united-kingdom                         1.0
9                germany                         1.0

   best_starting_grid_position  best_race_result
total_championship_wins  \
0                          1.0               1.0
4
1                          1.0               1.0
3
2                          1.0               1.0
2
3                          1.0               1.0
2
4                          1.0               1.0
1
5                          1.0               1.0
7
6                          1.0               1.0
```

```
4
7                         1.0               1.0
7
8                         1.0               1.0
1
9                         1.0               1.0
4

   total_race_starts  total_race_wins  total_race_laps  total_podiums
\
0                199               51            10540            106

1                161               41             8219             80

2                420               32            22758            106

3                 72               25             3877             32

4                350               21            18621            103

5                374              105            21325            202

6                227               67            12329            121

7                306               91            16825            155

8                187               31             8750             59

9                299               53            16426            122


   total_points  total_pole_positions  total_fastest_laps
total_grand_slams  \
0         798.5                    33                  41
0
1         614.0                    65                  19
4
2        2373.0                    22                  26
0
3         274.0                    33                  28
8
4        1873.0                    18                  46
0
5        4987.5                   104                  68
6
6        3296.5                    46                  35
6
7        1566.0                    68                  77
5
8         482.0                    32                  30
4
```

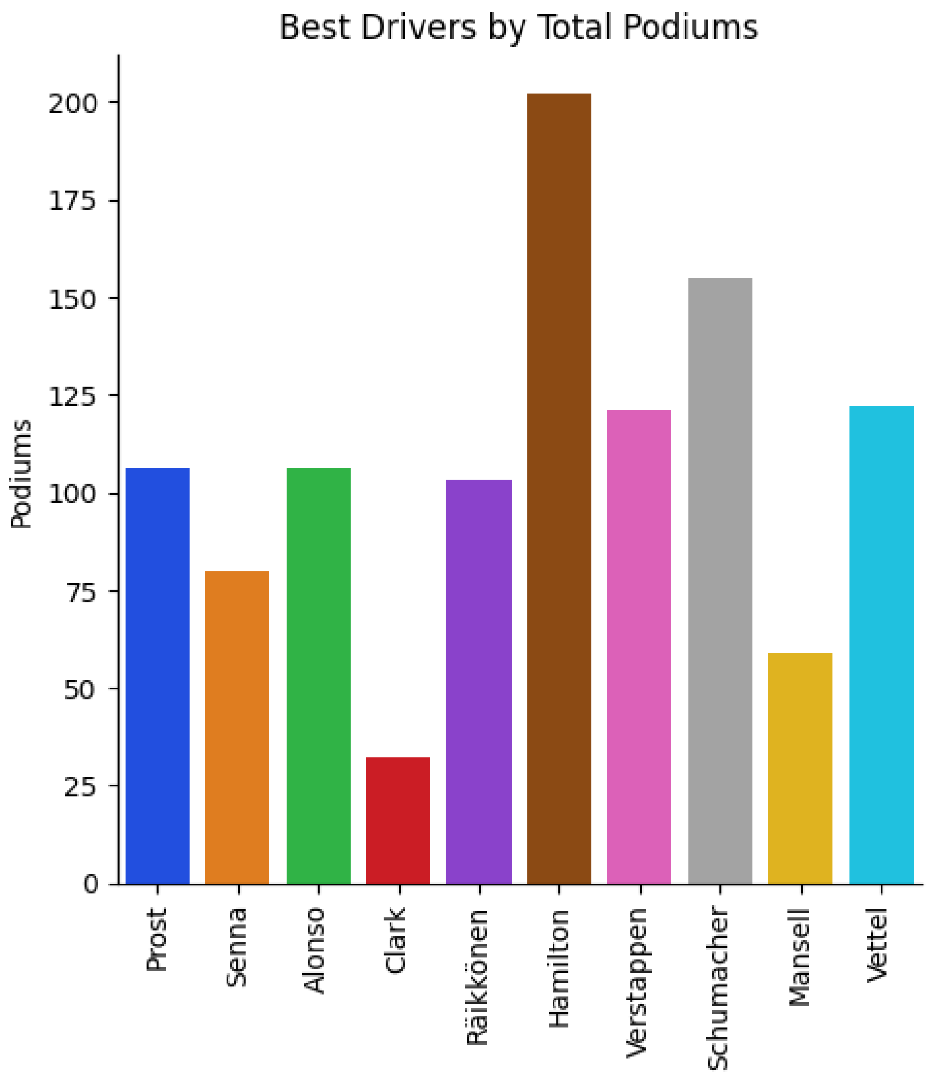
```
9        3098.0                    57                    38
4

   seasons  races
0       13    202
1       11    162
2       22    422
3        9     73
4       19    353
5       19    374
6       12    230
7       19    308
8       15    191
9       17    307

champ = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_championship_wins', errorbar=None,
                   hue='last_name')
for ax in champ.axes.flat:
    ax.set_title('Best Drivers by Total Championship Wins')
    ax.set_xlabel('')
    ax.set_ylabel('Championships')
    ax.tick_params('x', labelrotation=90)
```



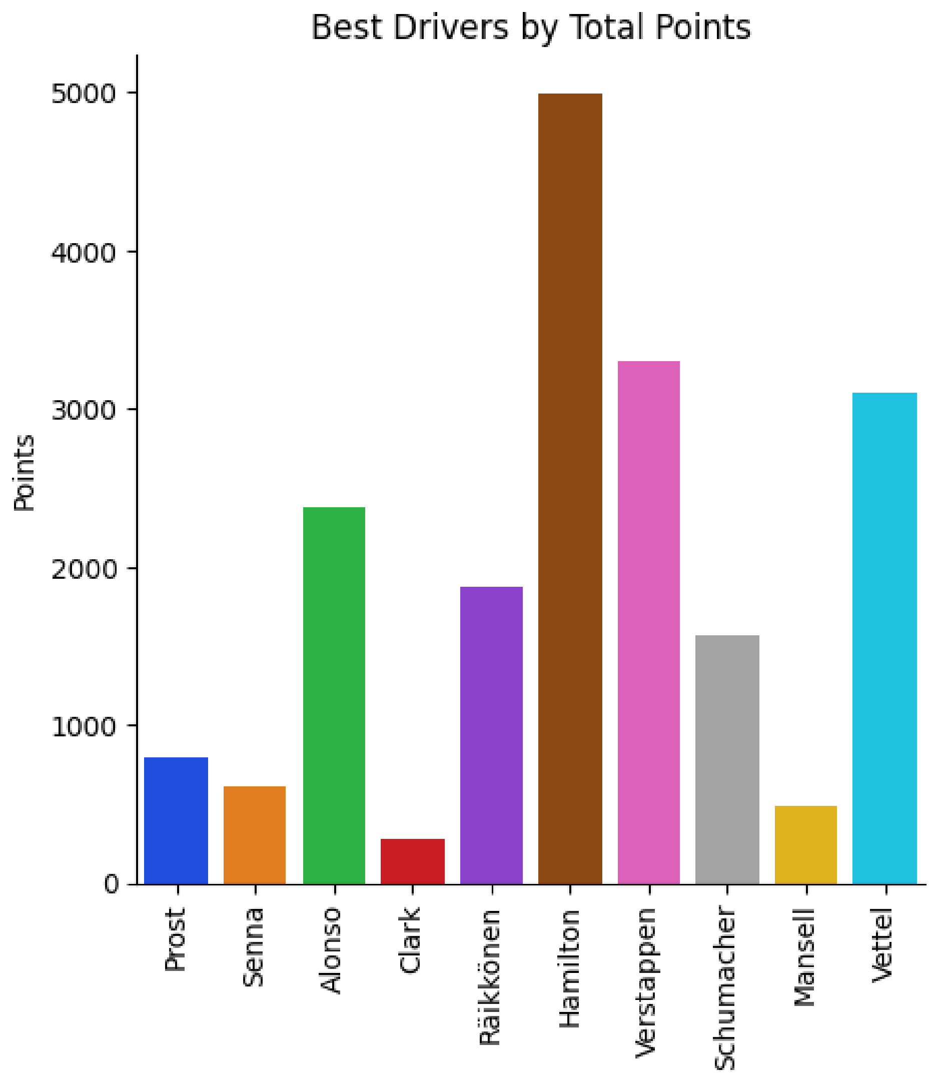
```python
wins = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_race_wins', errorbar=None, hue='last_name')
for ax in wins.axes.flat:
    ax.set_title('Best Drivers by Total Race Wins')
    ax.set_xlabel('')
    ax.set_ylabel('Race Wins')
    ax.tick_params('x', labelrotation=90)
```



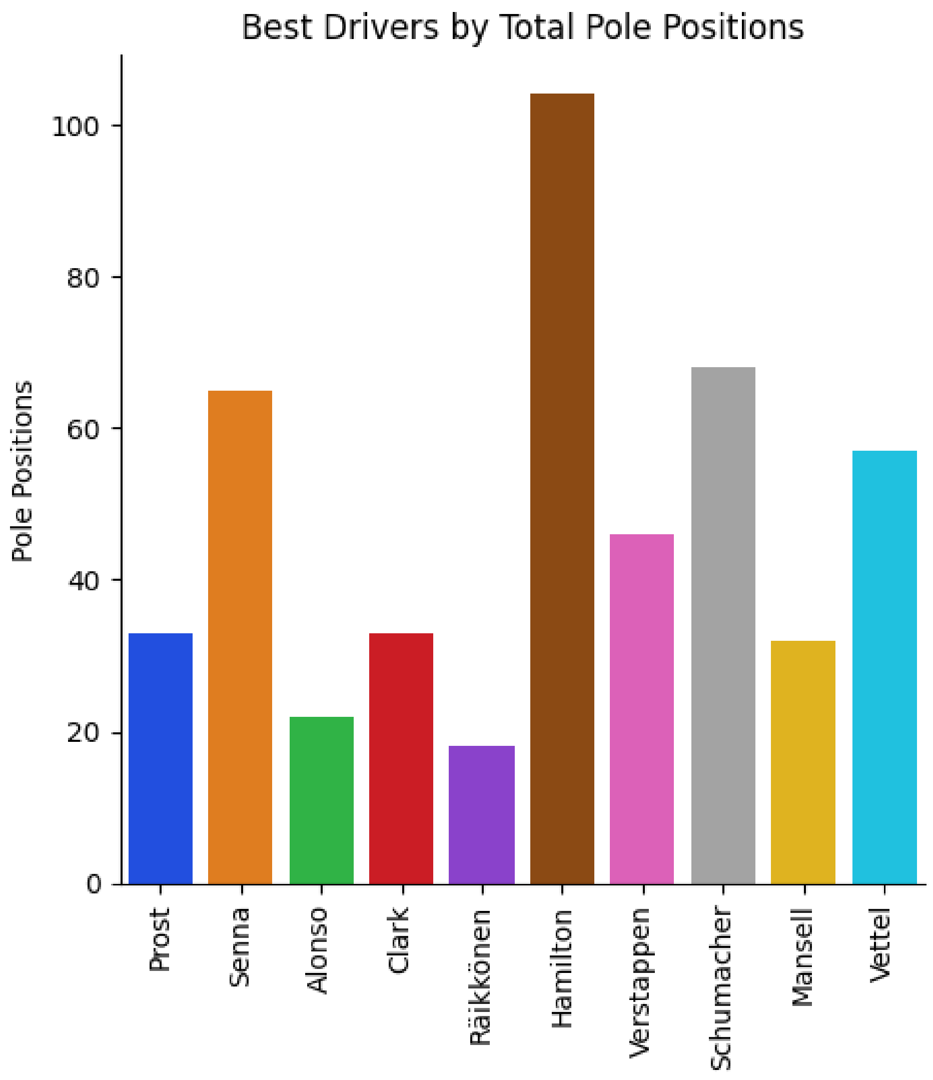
```python
pods = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_podiums', errorbar=None, hue='last_name')
for ax in pods.axes.flat:
    ax.set_title('Best Drivers by Total Podiums')
    ax.set_xlabel('')
    ax.set_ylabel('Podiums')
    ax.tick_params('x', labelrotation=90)
```



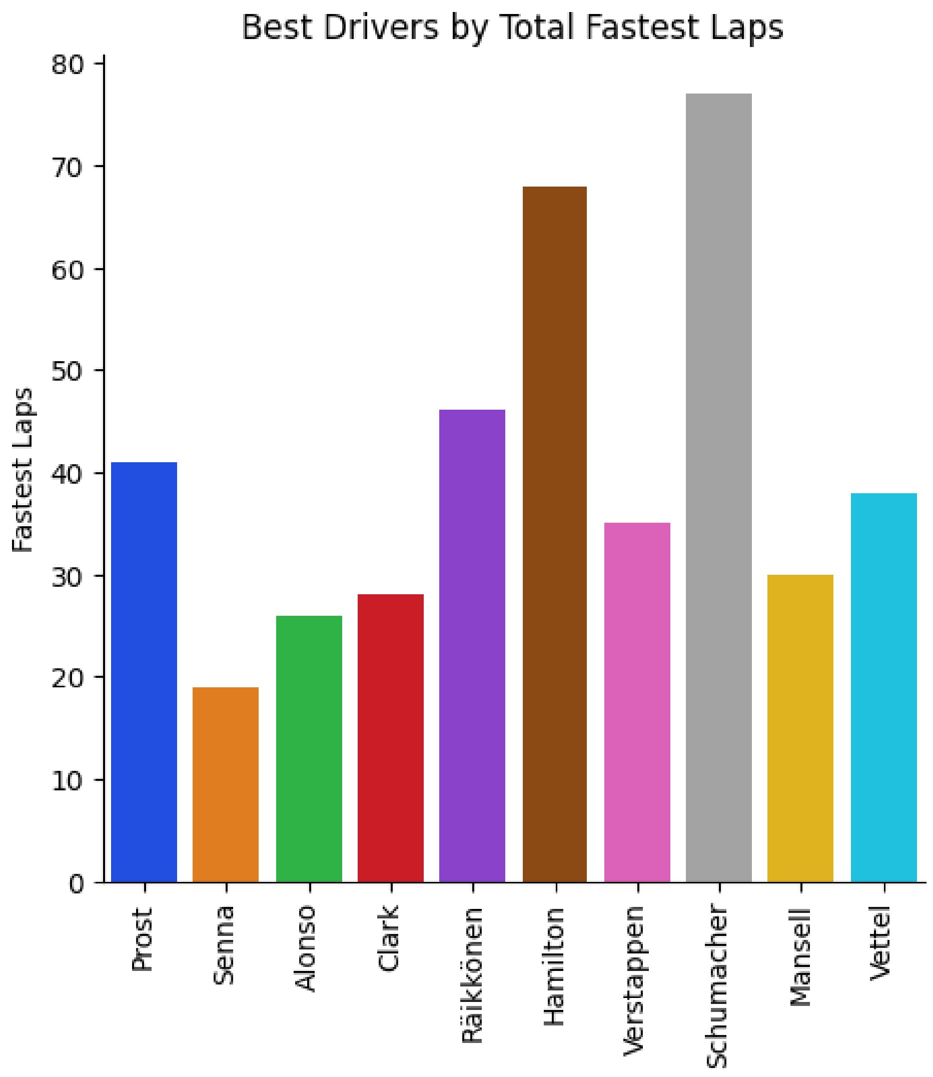
```
point = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_points', errorbar=None, hue='last_name')
for ax in point.axes.flat:
    ax.set_title('Best Drivers by Total Points')
    ax.set_xlabel('')
    ax.set_ylabel('Points')
    ax.tick_params('x', labelrotation=90)
```



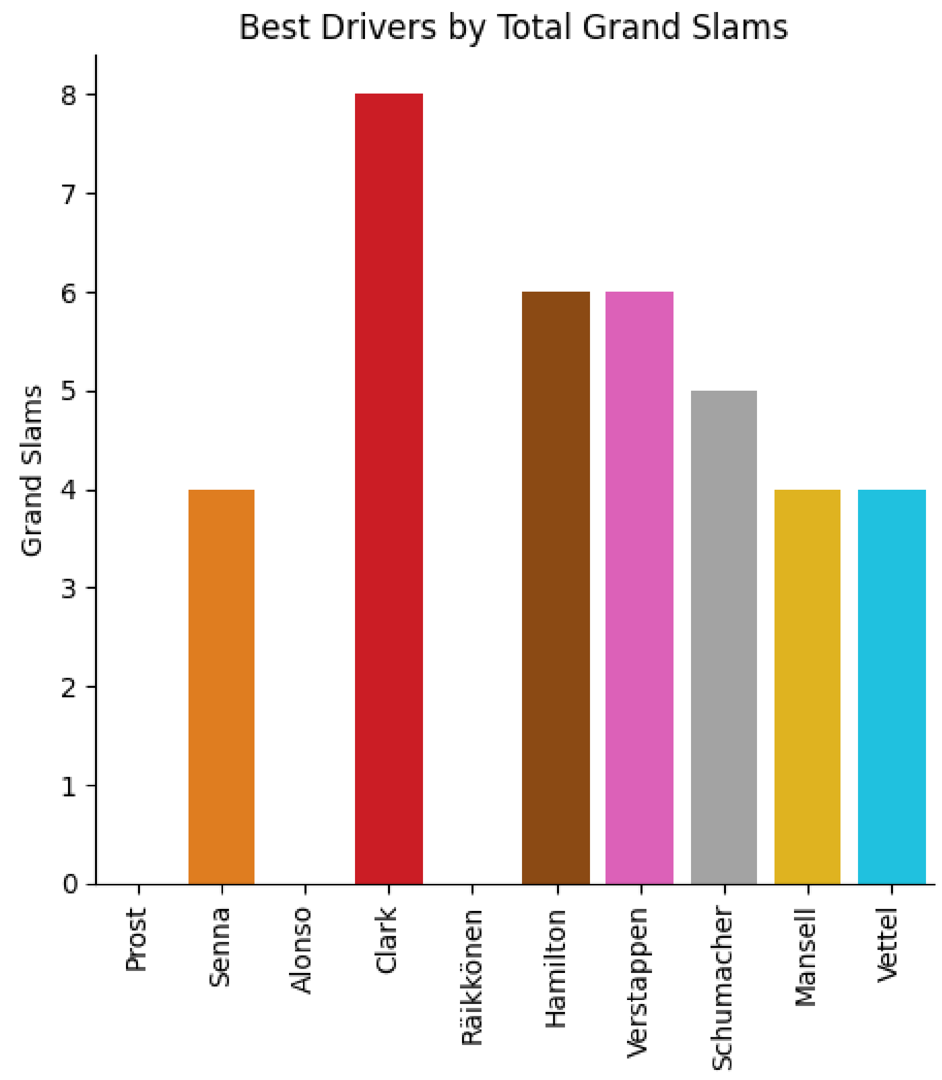
```
pole = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_pole_positions', errorbar=None, hue='last_name')
for ax in pole.axes.flat:
    ax.set_title('Best Drivers by Total Pole Positions')
    ax.set_xlabel('')
    ax.set_ylabel('Pole Positions')
    ax.tick_params('x', labelrotation=90)
```



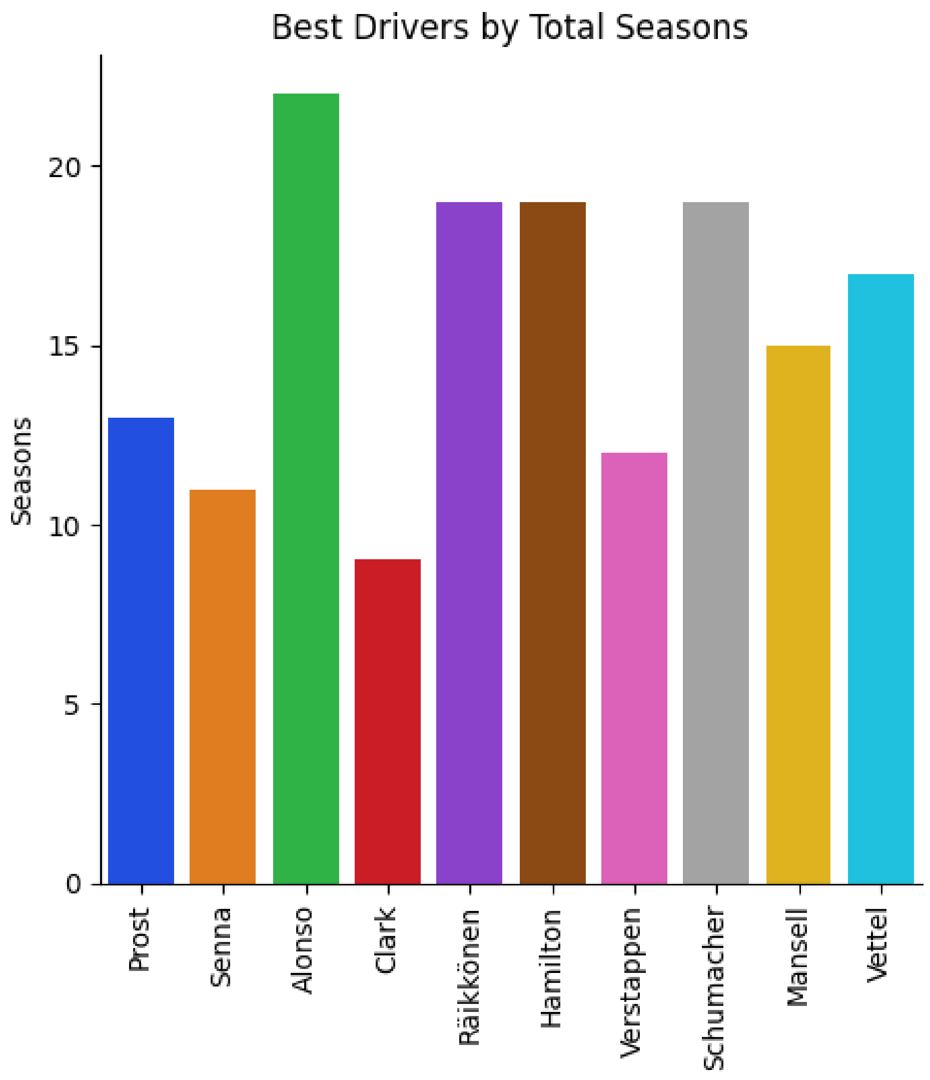
```python
laps = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_fastest_laps', errorbar=None, hue='last_name')
for ax in laps.axes.flat:
    ax.set_title('Best Drivers by Total Fastest Laps')
    ax.set_xlabel('')
    ax.set_ylabel('Fastest Laps')
    ax.tick_params('x', labelrotation=90)
```



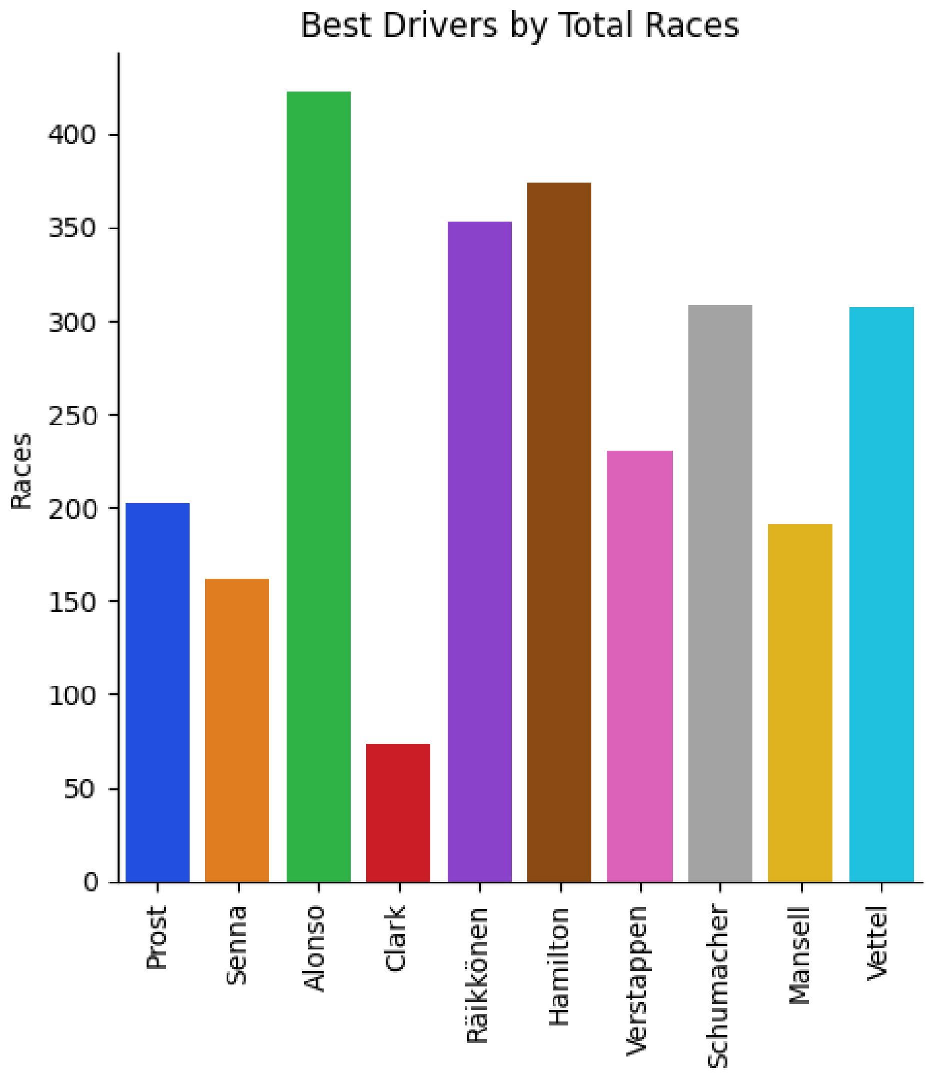
```
slam = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='total_grand_slams', errorbar=None, hue='last_name')
for ax in slam.axes.flat:
    ax.set_title('Best Drivers by Total Grand Slams')
    ax.set_xlabel('')
    ax.set_ylabel('Grand Slams')
    ax.tick_params('x', labelrotation=90)
```



```
sea = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='seasons', errorbar=None, hue='last_name')
for ax in sea.axes.flat:
    ax.set_title('Best Drivers by Total Seasons')
    ax.set_xlabel('')
    ax.set_ylabel('Seasons')
    ax.tick_params('x', labelrotation=90)
```



```
rac = sns.catplot(data=top_10_drivers, kind='bar', x='last_name',
y='races', errorbar=None, hue='last_name')
for ax in rac.axes.flat:
    ax.set_title('Best Drivers by Total Races')
    ax.set_xlabel('')
    ax.set_ylabel('Races')
    ax.tick_params('x', labelrotation=90)
```



Best Drivers by Total Races
Races
0
50
100
150
200
250
300
350
400
Prost
Senna
Alonso
Clark
Räikkönen
Hamilton
Verstappen
Schumacher
Mansell
Vettel

import modules

```
conda install psycopg2
```

```
2 channel Terms of Service accepted
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 25.5.1
    latest version: 25.11.0

Please update conda by running

    $ conda update -n base -c defaults conda



# All requested packages already installed.


Note: you may need to restart the kernel to use updated packages.
```

```python
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import pickle
import os

# Set up plotting style
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (12, 6)

# Load pickled data files
pickled_path = "./resources/pickled_tables/"

# Load engine and engine_manufacturer data
with open(os.path.join(pickled_path, "engine.plk"), "rb") as f:
    engine_df = pickle.load(f)

with open(os.path.join(pickled_path, "engine_manufacturer.plk"), "rb")
as f:
    engine_manufacturer_df = pickle.load(f)

# Load related tables for comprehensive analysis
```

```
with open(os.path.join(pickled_path,
"season_engine_manufacturer.plk"), "rb") as f:
    season_engine_manufacturer_df = pickle.load(f)

with open(os.path.join(pickled_path, "season_entrant_engine.plk"),
"rb") as f:
    season_entrant_engine_df = pickle.load(f)

with open(os.path.join(pickled_path, "race.plk"), "rb") as f:
    race_df = pickle.load(f)

with open(os.path.join(pickled_path, "season.plk"), "rb") as f:
    season_df = pickle.load(f)

print(f"Engine Data Shape: {engine_df.shape}")
print(f"Engine Manufacturer Data Shape:
{engine_manufacturer_df.shape}")
print(f"Season Engine Manufacturer Data Shape:
{season_engine_manufacturer_df.shape}")
print(f"Season Entrant Engine Data Shape:
{season_entrant_engine_df.shape}")

Engine Data Shape: (419, 7)
Engine Manufacturer Data Shape: (76, 17)
Season Engine Manufacturer Data Shape: (555, 15)
Season Entrant Engine Data Shape: (2016, 5)
```

Display engine manufacturer data

```
print("Engine Manufacturer DataFrame:")
print(engine_manufacturer_df.head(10))
print("\nEngine Manufacturer Columns:",
engine_manufacturer_df.columns.tolist())
print("\nData Info:")
print(engine_manufacturer_df.info())

Engine Manufacturer DataFrame:
             id         name      country_id
best_championship_position  \
0          acer         Acer          taiwan
9.0
1    alfa-romeo   Alfa Romeo           italy
3.0
2          alta         Alta  united-kingdom
NaN
3        arrows       Arrows  united-kingdom
7.0
4      asiatech     Asiatech          france
```

```
9.0
5  aston-martin  Aston Martin  united-kingdom
NaN
6           ats           ATS           italy
NaN
7           bmw           BMW         germany
2.0
8      borgward      Borgward         germany
NaN
9           bpm           BPM           italy
NaN

   best_starting_grid_position  best_race_result
total_championship_wins  \
0                          4.0               5.0
0
1                          1.0               1.0
0
2                          6.0               3.0
0
3                          6.0               4.0
0
4                         13.0               5.0
0
5                          2.0               6.0
0
6                         13.0              11.0
0
7                          1.0               1.0
0
8                         16.0              10.0
0
9                          NaN               NaN
0

   total_race_entries  total_race_starts  total_race_wins
total_race_laps  \
0                  17                 17                0
1707
1                 225                215               12
17979
2                  29                 26                0
2610
3                  32                 32                0
2254
4                  34                 33                0
2826
5                   6                  5                0
518
```

```
6                    7                  7                 0
303
7                  273                270                20
31993
8                    2                  1                 0
139
9                    1                  0                 0
0

   total_podiums  total_podium_races  total_points
total_championship_points  \
0              0                   0           4.0
4.0
1             40                  30         166.0
148.0
2              1                   1           0.0
0.0
3              0                   0           7.0
7.0
4              0                   0           3.0
3.0
5              0                   0           0.0
0.0
6              0                   0           0.0
0.0
7             86                  76        1021.0
1021.0
8              0                   0           0.0
0.0
9              0                   0           0.0
0.0

   total_pole_positions  total_fastest_laps
0                     0                   0
1                    15                  20
2                     0                   0
3                     0                   0
4                     0                   0
5                     0                   0
6                     0                   0
7                    33                  33
8                     0                   0
9                     0                   0

Engine Manufacturer Columns: ['id', 'name', 'country_id',
'best_championship_position', 'best_starting_grid_position',
'best_race_result', 'total_championship_wins', 'total_race_entries',
'total_race_starts', 'total_race_wins', 'total_race_laps',
'total_podiums', 'total_podium_races', 'total_points',
'total_championship_points', 'total_pole_positions',
'total_fastest_laps']

Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76 entries, 0 to 75
Data columns (total 17 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   id                            76 non-null     object 
 1   name                          76 non-null     object 
 2   country_id                    76 non-null     object 
 3   best_championship_position    45 non-null     float64
 4   best_starting_grid_position   72 non-null     float64
 5   best_race_result              65 non-null     float64
 6   total_championship_wins       76 non-null     int64  
 7   total_race_entries            76 non-null     int64  
 8   total_race_starts             76 non-null     int64  
 9   total_race_wins               76 non-null     int64  
 10  total_race_laps               76 non-null     int64  
 11  total_podiums                 76 non-null     int64  
 12  total_podium_races            76 non-null     int64  
 13  total_points                  76 non-null     float64
 14  total_championship_points     76 non-null     float64
 15  total_pole_positions          76 non-null     int64  
 16  total_fastest_laps            76 non-null     int64  
dtypes: float64(5), int64(9), object(3)
memory usage: 10.2+ KB
None
```

Sort by wins and display top performers

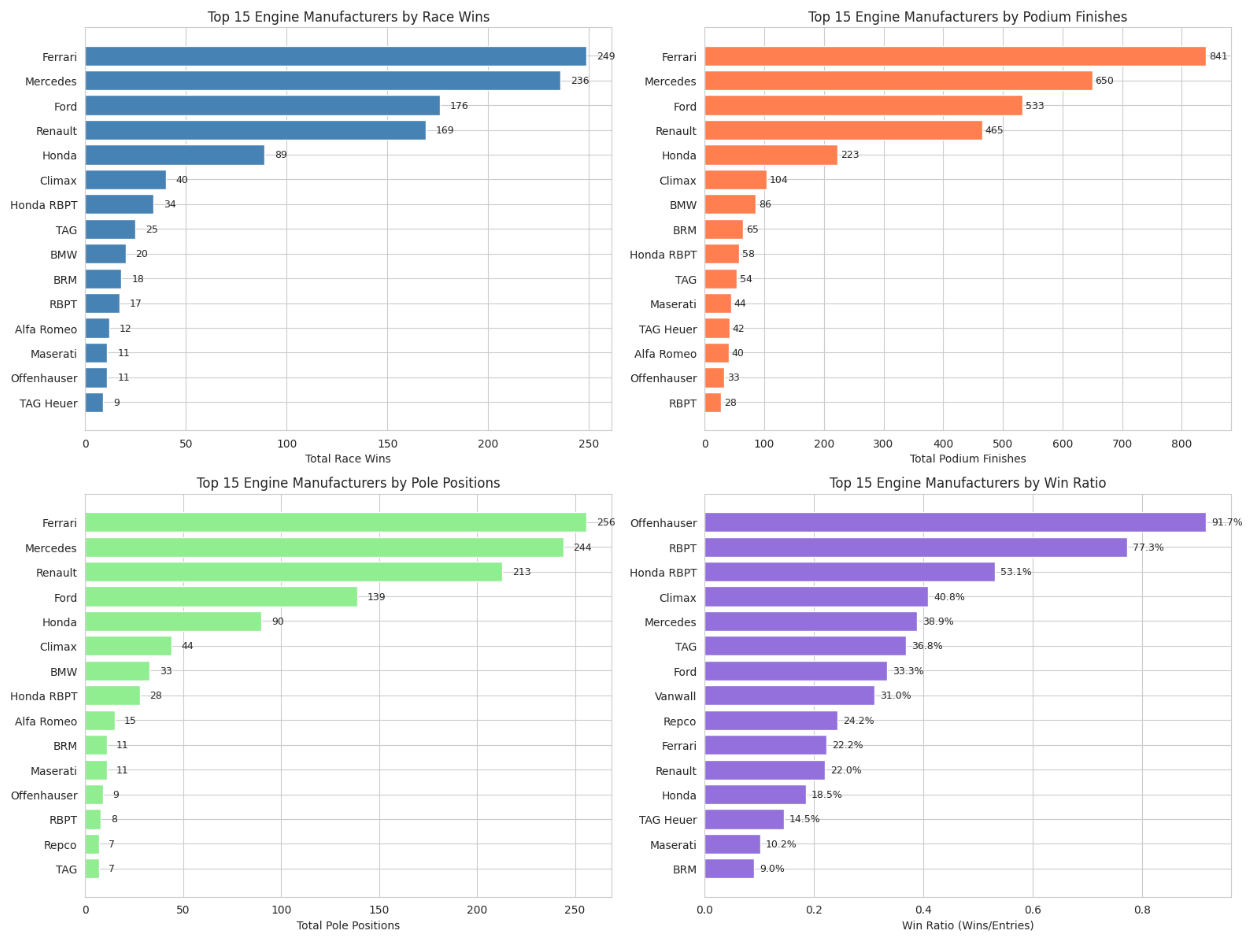
```
# Sort by wins and display top performers
top_engines = engine_manufacturer_df.nlargest(15, 'total_race_wins')[
    ['id', 'name', 'country_id', 'total_race_entries',
'total_race_wins',
     'total_podiums', 'total_pole_positions', 'total_fastest_laps']
].reset_index(drop=True)

print("Top 15 Engine Manufacturers by Wins:")
print(top_engines)

Top 15 Engine Manufacturers by Wins:
             id          name                country_id
total_race_entries  \
0       ferrari      Ferrari                      italy
1120
1      mercedes     Mercedes                    germany
607
```

```
2          ford          Ford  united-states-of-america
528
3       renault       Renault                    france
768
4         honda         Honda                     japan
482
5        climax        Climax            united-kingdom
98
6    honda-rbpt   Honda RBPT                     japan
64
7           tag           TAG                luxembourg
68
8           bmw           BMW                   germany
273
9           brm           BRM            united-kingdom
200
10         rbpt          RBPT                     japan
22
11   alfa-romeo   Alfa Romeo                     italy
225
12     maserati      Maserati                     italy
108
13  offenhauser  Offenhauser  united-states-of-america
12
14    tag-heuer    TAG Heuer               switzerland
62

    total_race_wins  total_podiums  total_pole_positions
total_fastest_laps
0               249            841                   256
275
1               236            650                   244
236
2               176            533                   139
162
3               169            465                   213
177
4                89            223                    90
76
5                40            104                    44
45
6                34             58                    28
19
7                25             54                     7
18
8                20             86                    33
33
9                18             65                    11
14
```

```
10                17             28                      8
8
11                12             40                     15
20
12                11             44                     11
19
13                11             33                      9
10
14                 9             42                      3
13
```

Create subplot visualizations

```
fig, axes = plt.subplots(2, 2, figsize=(16, 12))

top_15_wins = engine_manufacturer_df.nlargest(15, 'total_race_wins')
axes[0, 0].barh(top_15_wins['name'], top_15_wins['total_race_wins'],
color='steelblue')
axes[0, 0].set_xlabel('Total Race Wins')
axes[0, 0].set_title('Top 15 Engine Manufacturers by Race Wins')
axes[0, 0].invert_yaxis()
for i, wins in enumerate(top_15_wins['total_race_wins']):
    axes[0, 0].text(wins + 5, i, f'{int(wins):,}', va='center',
fontsize=9)

top_15_podiums = engine_manufacturer_df.nlargest(15, 'total_podiums')
axes[0, 1].barh(top_15_podiums['name'],
top_15_podiums['total_podiums'], color='coral')
axes[0, 1].set_xlabel('Total Podium Finishes')
axes[0, 1].set_title('Top 15 Engine Manufacturers by Podium Finishes')
axes[0, 1].invert_yaxis()
for i, podiums in enumerate(top_15_podiums['total_podiums']):
    axes[0, 1].text(podiums + 5, i, f'{int(podiums):,}', va='center',
fontsize=9)

top_15_poles = engine_manufacturer_df.nlargest(15,
'total_pole_positions')
axes[1, 0].barh(top_15_poles['name'],
top_15_poles['total_pole_positions'], color='lightgreen')
axes[1, 0].set_xlabel('Total Pole Positions')
axes[1, 0].set_title('Top 15 Engine Manufacturers by Pole Positions')
axes[1, 0].invert_yaxis()
for i, poles in enumerate(top_15_poles['total_pole_positions']):
    axes[1, 0].text(poles + 5, i, f'{int(poles):,}', va='center',
fontsize=9)

engine_manufacturer_df['win_ratio'] =
(engine_manufacturer_df['total_race_wins'] /
engine_manufacturer_df['total_race_entries']).fillna(0)
top_15_ratio = engine_manufacturer_df.nlargest(15, 'win_ratio')
axes[1, 1].barh(top_15_ratio['name'], top_15_ratio['win_ratio'],
color='mediumpurple')
axes[1, 1].set_xlabel('Win Ratio (Wins/Entries)')
axes[1, 1].set_title('Top 15 Engine Manufacturers by Win Ratio')
axes[1, 1].invert_yaxis()
for i, ratio in enumerate(top_15_ratio['win_ratio']):
    axes[1, 1].text(ratio + 0.01, i, f'{ratio:.1%}', va='center',
fontsize=9)

plt.tight_layout()
plt.savefig('engine_manufacturers_summary.png', dpi=300,
bbox_inches='tight')
plt.show()
print("Engine manufacturer summary visualization saved!")
```



```
Engine manufacturer summary visualization saved!
```

Calculate efficiency metrics

```
# Calculate efficiency metrics
engine_stats = engine_manufacturer_df[
    ['name', 'total_race_entries', 'total_race_wins', 'total_podiums',

     'total_pole_positions', 'total_fastest_laps']
].copy()

# Calculate ratios
engine_stats['podium_ratio'] =
(engine_manufacturer_df['total_podiums'] /

engine_manufacturer_df['total_race_entries']).round(3)
engine_stats['win_ratio'] = (engine_manufacturer_df['total_race_wins']
/

engine_manufacturer_df['total_race_entries']).round(3)
engine_stats['pole_ratio'] =
(engine_manufacturer_df['total_pole_positions'] /

engine_manufacturer_df['total_race_entries']).round(3)
engine_stats['fastest_lap_ratio'] =
(engine_manufacturer_df['total_fastest_laps'] /

engine_manufacturer_df['total_race_entries']).round(3)

# Sort by podium ratio and display top performers
top_performers = engine_stats.sort_values('podium_ratio',
ascending=False).head(15)
print("\nTop 15 Engine Manufacturers by Efficiency (Podium Ratio):")
print(top_performers.to_string())
```

```
Top 15 Engine Manufacturers by Efficiency (Podium Ratio):
           name  total_race_entries  total_race_wins  total_podiums
total_pole_positions  total_fastest_laps  podium_ratio  win_ratio
pole_ratio  fastest_lap_ratio
48  Offenhauser                  12               11             33
9                  10         2.750      0.917      0.750
0.833
57         RBPT                  22               17             28
8                   8         1.273      0.773      0.364
0.364
43     Mercedes                 607              236            650
244                 236         1.071      0.389      0.402
0.389
16       Climax                  98               40            104
44                  45         1.061      0.408      0.449
0.459
```

```
25          Ford                   528              176            533
139                  162            1.009      0.333       0.263
0.307
28   Honda RBPT                     64               34             58
28                   19            0.906      0.531       0.438
0.297
66           TAG                    68               25             54
7                   18            0.794      0.368       0.103
0.265
59         Repco                    33                8             25
7                    4            0.758      0.242       0.212
0.121
23       Ferrari                  1120              249            841
256                  275            0.751      0.222       0.229
0.246
65    TAG Heuer                     62                9             42
3                   13            0.677      0.145       0.048
0.210
58       Renault                   768              169            465
213                  177            0.605      0.220       0.277
0.230
29         Honda                   482               89            223
90                   76            0.463      0.185       0.187
0.158
71       Vanwall                    29                9             13
7                    6            0.448      0.310       0.241
0.207
39      Maserati                   108               11             44
11                   19            0.407      0.102       0.102
0.176
11           BRM                   200               18             65
11                   14            0.325      0.090       0.055
0.070

print(season_df.columns.tolist())
print(season_df.head())

['year']
   year
0  1950
1  1951
2  1952
3  1953
4  1954
```

Generate comprehensive summary

```
# Generate comprehensive summary
summary_stats = engine_manufacturer_df[
    ['total_race_entries', 'total_race_wins', 'total_podiums',
     'total_pole_positions', 'total_fastest_laps']
].describe()

print("\nSummary Statistics for Engine Manufacturers:")
print(summary_stats.round(2))

# Key insights
print("\n=== KEY INSIGHTS ===")
print(f"Total Manufacturers: {len(engine_manufacturer_df)}")
print(f"Total Race Entries:
{engine_manufacturer_df['total_race_entries'].sum()}")
print(f"Total Races Won:
{engine_manufacturer_df['total_race_wins'].sum()}")
print(f"Total Podiums:
{engine_manufacturer_df['total_podiums'].sum()}")

# Top performers
print("\nMost Successful Engine Manufacturer:")
top =
engine_manufacturer_df.loc[engine_manufacturer_df['total_race_wins'].i
dxmax()]
print(f"  {top['name']}: {int(top['total_race_wins'])} wins")

print("\nHighest Win Ratio:")
engine_manufacturer_df['win_ratio'] =
(engine_manufacturer_df['total_race_wins'] /

engine_manufacturer_df['total_race_entries'])
top_ratio =
engine_manufacturer_df.loc[engine_manufacturer_df['win_ratio'].idxmax(
)]
print(f"  {top_ratio['name']}: {top_ratio['win_ratio']:.3f} wins per
entry")
```

```

Summary Statistics for Engine Manufacturers:
       total_race_entries  total_race_wins  total_podiums  \
count               76.00            76.00          76.00
mean                89.43            15.04          45.11
std                183.62            48.03         144.38
min                  1.00             0.00           0.00
25%                  6.00             0.00           0.00
50%                 21.00             0.00           1.00
75%                 77.00             1.50          14.50
max               1120.00           249.00         841.00
```

```
       total_pole_positions  total_fastest_laps
count                 76.00               76.00
mean                  15.04               15.25
std                   49.70               49.29
min                    0.00                0.00
25%                    0.00                0.00
50%                    0.00                0.00
75%                    2.25                3.25
max                  256.00              275.00

=== KEY INSIGHTS ===
Total Manufacturers: 76
Total Race Entries: 6797
Total Races Won: 1143
Total Podiums: 3428

Most Successful Engine Manufacturer:
  Ferrari: 249 wins

Highest Win Ratio:
  Offenhauser: 0.917 wins per entry
```

Race Data Analysis

Goal: Develop graphs of WR statistics of certain tracks over time

Test the hypothesis: World Record Times have been cut down as new technology in racing becomes better

Import Data and Modules

```
from seaborn import FacetGrid

from DataFrameImport import get_schema_info, list_schemas, schema

import pandas as pd
import seaborn as sns

path = 'resources/pickled_tables/'
extension = '.plk'

race_data_table = 'race_data'
race_table = 'race'
circuit_table = 'circuit'

race_data_file = path + race_data_table + extension
race_file = path + race_table + extension
circuit_file = path + circuit_table + extension

race_data = pd.read_pickle(race_data_file)
race = pd.read_pickle(race_file)
circuit = pd.read_pickle(circuit_file)

race_data
```

```
        race_id                       type  position_display_order  \
0           290     PRE_QUALIFYING_RESULT                       1
1           290     PRE_QUALIFYING_RESULT                       2
2           290     PRE_QUALIFYING_RESULT                       3
3           290     PRE_QUALIFYING_RESULT                       4
4           290     PRE_QUALIFYING_RESULT                       5
...         ...                       ...                     ...
183627     1143   DRIVER_OF_THE_DAY_RESULT                      1
183628     1143   DRIVER_OF_THE_DAY_RESULT                      2
183629     1143   DRIVER_OF_THE_DAY_RESULT                      3
183630     1143   DRIVER_OF_THE_DAY_RESULT                      4
183631     1143   DRIVER_OF_THE_DAY_RESULT                      5

        position_number position_text driver_number
driver_id  \
0                   1.0             1            40   gilles-
```

```
villeneuve
1                   2.0            2            23      patrick-
tambay
2                   3.0            3            34  jean-pierre-
jarier
3                   4.0            4            30        brett-
lunger
4                   5.0            5            38        brian-
henton
...                 ...          ...           ...              ..
.
183627              1.0            1            14     fernando-
alonso
183628              2.0            2            63      george-
russell
183629              3.0            3             1      max-
verstappen
183630              4.0            4             4        lando-
norris
183631              5.0            5            44      lewis-
hamilton

       constructor_id engine_manufacturer_id tyre_manufacturer_id  ...
\
0             mclaren                   ford             goodyear  ...

1              ensign                   ford             goodyear  ...

2              penske                   ford             goodyear  ...

3             mclaren                   ford             goodyear  ...

4               march                   ford             goodyear  ...

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

183627   aston-martin               mercedes              pirelli  ...

183628       mercedes               mercedes              pirelli  ...

183629       red-bull             honda-rbpt              pirelli  ...

183630        mclaren               mercedes              pirelli  ...

183631        ferrari                ferrari              pirelli  ...


       fastest_lap_time_millis  fastest_lap_gap fastest_lap_gap_millis
\
0                          NaN             None                    NaN
```

```
1                                NaN              None                        NaN

2                                NaN              None                        NaN

3                                NaN              None                        NaN

4                                NaN              None                        NaN

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

183627                           NaN              None                        NaN

183628                           NaN              None                        NaN

183629                           NaN              None                        NaN

183630                           NaN              None                        NaN

183631                           NaN              None                        NaN


        fastest_lap_interval fastest_lap_interval_millis
pit_stop_stop  \
0                       None                         NaN
NaN
1                       None                         NaN
NaN
2                       None                         NaN
NaN
3                       None                         NaN
NaN
4                       None                         NaN
NaN
...                      ...                         ...              ..
.
183627                  None                         NaN
NaN
183628                  None                         NaN
NaN
183629                  None                         NaN
NaN
183630                  None                         NaN
NaN
183631                  None                         NaN
NaN

        pit_stop_lap pit_stop_time  pit_stop_time_millis  \
0                NaN          None                   NaN
1                NaN          None                   NaN
2                NaN          None                   NaN
```

```
3                 NaN          None                       NaN
4                 NaN          None                       NaN
...               ...           ...                       ...
183627            NaN          None                       NaN
183628            NaN          None                       NaN
183629            NaN          None                       NaN
183630            NaN          None                       NaN
183631            NaN          None                       NaN

       driver_of_the_day_percentage
0                               NaN
1                               NaN
2                               NaN
3                               NaN
4                               NaN
...                             ...
183627                         22.5
183628                         16.4
183629                         14.5
183630                          8.7
183631                          7.6

[183632 rows x 71 columns]

race

        id  year  round        date   time  grand_prix_id  \
0        1  1950      1  1950-05-13   None  great-britain
1        2  1950      2  1950-05-21   None         monaco
2        3  1950      3  1950-05-30   None   indianapolis
3        4  1950      4  1950-06-04   None    switzerland
4        5  1950      5  1950-06-18   None        belgium
...    ...   ...    ...         ...    ...            ...
1144  1145  2025     20  2025-10-26  20:00         mexico
1145  1146  2025     21  2025-11-09  17:00      sao-paulo
1146  1147  2025     22  2025-11-23  04:00      las-vegas
1147  1148  2025     23  2025-11-30  16:00          qatar
1148  1149  2025     24  2025-12-07  13:00      abu-dhabi

                                          official_name
qualifying_format  \
0                         1950 RAC British Grand Prix
TWO_SESSION
1                           Grand Prix de Monaco 1950
TWO_SESSION
2                               1950 Indianapolis 500
FOUR_LAPS
3                        Grosser Preis der Schweiz 1950
TWO_SESSION
4                             1950 Belgian Grand Prix
```

```
TWO_SESSION
...                                                  ...
...
1144  Formula 1 Gran Premio de la Ciudad de México 2025
KNOCKOUT
1145  Formula 1 MSC Cruises Grande Prêmio de São Pau...
KNOCKOUT
1146       Formula 1 Heineken Las Vegas Grand Prix 2025
KNOCKOUT
1147      Formula 1 Qatar Airways Qatar Grand Prix 2025
KNOCKOUT
1148  Formula 1 Etihad Airways Abu Dhabi Grand Prix ...
KNOCKOUT

     sprint_qualifying_format         circuit_id  ...
qualifying_2_date  \
0                        None        silverstone  ...
None
1                        None             monaco  ...
None
2                        None       indianapolis  ...
None
3                        None         bremgarten  ...
None
4                        None  spa-francorchamps  ...
None
...                       ...                ...  ...                ..
.
1144                     None        mexico-city  ...
None
1145          SPRINT_SHOOTOUT         interlagos  ...
None
1146                     None          las-vegas  ...
None
1147          SPRINT_SHOOTOUT             lusail  ...
None
1148                     None         yas-marina  ...
None

     qualifying_2_time  qualifying_date  qualifying_time  \
0                 None             None             None
1                 None             None             None
2                 None             None             None
3                 None             None             None
4                 None             None             None
...                ...              ...              ...
1144              None       2025-10-25            21:00
1145              None       2025-11-08            18:00
1146              None       2025-11-22            04:00
```

```
1147              None      2025-11-29          18:00
1148              None      2025-12-06          14:00

      sprint_qualifying_date  sprint_qualifying_time  sprint_race_date
\
0                       None                    None              None

1                       None                    None              None

2                       None                    None              None

3                       None                    None              None

4                       None                    None              None

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

1144                    None                    None              None

1145              2025-11-07                   18:30        2025-11-08

1146                    None                    None              None

1147              2025-11-28                   17:30        2025-11-29

1148                    None                    None              None


      sprint_race_time  warming_up_date  warming_up_time
0                 None             None             None
1                 None             None             None
2                 None             None             None
3                 None             None             None
4                 None             None             None
...                ...              ...              ...
1144              None             None             None
1145             14:00             None             None
1146              None             None             None
1147             14:00             None             None
1148              None             None             None

[1149 rows x 42 columns]

circuit

            id                name                        full_name  \
0     adelaide            Adelaide          Adelaide Street Circuit
1         aida                Aida  Okayama International Circuit
2     ain-diab            Ain-Diab                 Ain-Diab Circuit
3      aintree             Aintree   Aintree Motor Racing Circuit
4   anderstorp  Anderstorp Raceway               Anderstorp Raceway
```

```
..          ...             ...                           ...
72  yas-marina          Yas Marina            Yas Marina Circuit
73     yeongam               Korea    Korea International Circuit
74   zandvoort           Zandvoort        Circuit Park Zandvoort
75     zeltweg             Zeltweg                       Zeltweg
76      zolder              Zolder                 Circuit Zolder

          previous_names    type       direction      place_name  \
0                   None  STREET       CLOCKWISE        Adelaide
1        TI Circuit Aida    RACE       CLOCKWISE            Aida
2                   None    ROAD       CLOCKWISE      Casablanca
3                   None    ROAD       CLOCKWISE         Aintree
4   Scandinavian Raceway    RACE       CLOCKWISE      Anderstorp
..                   ...     ...             ...             ...
72                  None    RACE  ANTI_CLOCKWISE       Abu Dhabi
73                  None    RACE  ANTI_CLOCKWISE         Yeongam
74                  None    RACE       CLOCKWISE       Zandvoort
75                  None    ROAD       CLOCKWISE         Zeltweg
76                  None    RACE       CLOCKWISE  Heusden-Zolder

              country_id   latitude   longitude  length  turns  \
0              australia -34.927222  138.617222   3.780     16
1                  japan  34.915000  134.221111   3.703     13
2                morocco  33.578611   -7.687500   7.618     18
3         united-kingdom  53.476944   -2.940556   4.828      8
4                 sweden  57.264167   13.601389   4.031      8
..                   ...        ...         ...     ...    ...
72  united-arab-emirates  24.467222   54.603056   5.281     16
73           south-korea  34.733333  126.416667   5.615     18
74           netherlands  52.388819    4.540922   4.259     14
75               austria  47.202222   14.742222   3.186      4
76               belgium  50.988889    5.255556   4.262     15

    total_races_held
0                 11
1                  2
2                  1
3                  5
4                  6
..               ...
72                16
73                 4
74                35
75                 1
76                10

[77 rows x 13 columns]
```

Clean Data

Data Documentation Look-up

Display helper tools that describe the information contained in the table

```
get_schema_info('races')
```

```
races
The list of races, each representing detailed information about
individual races, including results, participants, and statistics.
```

```
get_schema_info('circuits')
```

```
circuits
The list of circuits, each representing a specific racing track,
including geographical location and race history.
```

```
print(schema.Race)
```

```
id = type = integer
description = The unique identifier of the race.

year = type = integer
description = The year of the season.

round = type = integer
description = The round number of the race in the season.

date = type = string
description = The date of the race in UTC.

time = type = ['string', 'null']
description = The start time of the race in UTC.

grandPrixId = type = string
description = The identifier of the Grand Prix associated with the
race.

officialName = type = string
description = The official name of the race.

qualifyingFormat = description = The qualifying format of the race.

sprintQualifyingFormat = description = The sprint qualifying format of
the race.

circuitId = type = string
description = The identifier of the circuit where the race takes
```

```
place.

circuitType = description = The type of the circuit.

direction = description = The direction of the circuit.

courseLength = type = number
description = The length of the circuit (race course) in kilometers.

turns = type = integer
description = The number of turns (corners) in the configuration of
the circuit.

laps = type = integer
description = The total number of laps of the race.

distance = type = number
description = The total distance of the race in kilometers.

scheduledLaps = type = ['integer', 'null']
description = The scheduled number of laps of the race.

scheduledDistance = type = ['number', 'null']
description = The scheduled distance of the race in kilometers.

driversChampionshipDecider = type = boolean
description = Whether this race was the decider of the World Drivers'
Championship.

constructorsChampionshipDecider = type = boolean
description = Whether this race was the decider of the World
Constructors' Championship.

preQualifyingDate = type = ['string', 'null']
description = The date of the pre-qualifying session in UTC.

preQualifyingTime = type = ['string', 'null']
description = The start time of the pre-qualifying session in UTC.

preQualifyingResults = type = ['array', 'null']
description = The results of the pre-qualifying session.

freePractice1Date = type = ['string', 'null']
description = The date of the 1st free practice session in UTC.

freePractice1Time = type = ['string', 'null']
description = The start time of the 1st free practice session in UTC.

freePractice1Results = type = ['array', 'null']
description = The results of the 1st free practice session.
```

```
freePractice2Date = type = ['string', 'null']
description = The date of the 2nd free practice session in UTC.

freePractice2Time = type = ['string', 'null']
description = The start time of the 2nd free practice session in UTC.

freePractice2Results = type = ['array', 'null']
description = The results of the 2nd free practice session.

freePractice3Date = type = ['string', 'null']
description = The date of the 3rd free practice session in UTC.

freePractice3Time = type = ['string', 'null']
description = The start time of the 3rd free practice session in UTC.

freePractice3Results = type = ['array', 'null']
description = The results of the 3rd free practice session.

freePractice4Date = type = ['string', 'null']
description = The date of the 4th free practice session UTC.

freePractice4Time = type = ['string', 'null']
description = The start time of the 4th free practice session in UTC.

freePractice4Results = type = ['array', 'null']
description = The results of the 4th free practice session.

qualifying1Date = type = ['string', 'null']
description = The date of the 1st qualifying session in UTC.

qualifying1Time = type = ['string', 'null']
description = The start time of the 1st qualifying session in UTC.

qualifying1Results = type = ['array', 'null']
description = The results of the 1st qualifying session.

qualifying2Date = type = ['string', 'null']
description = The date of the 2nd qualifying session UTC.

qualifying2Time = type = ['string', 'null']
description = The start time of the 2nd qualifying session UTC.

qualifying2Results = type = ['array', 'null']
description = The results of the 2nd qualifying session.

qualifyingDate = type = ['string', 'null']
description = The date of the qualifying session UTC.

qualifyingTime = type = ['string', 'null']
description = The start time of the qualifying session UTC.
```

```
qualifyingResults = type = ['array', 'null']
description = The results of the qualifying session.

sprintQualifyingDate = type = ['string', 'null']
description = The date of the sprint qualifying session in UTC.

sprintQualifyingTime = type = ['string', 'null']
description = The start time of the sprint qualifying session in UTC.

sprintQualifyingResults = type = ['array', 'null']
description = The results of the sprint qualifying session.

sprintStartingGridPositions = type = ['array', 'null']
description = The starting grid positions for the sprint race.

sprintRaceDate = type = ['string', 'null']
description = The date of the sprint race in UTC.

sprintRaceTime = type = ['string', 'null']
description = The start time of the sprint race in UTC.

sprintRaceResults = type = ['array', 'null']
description = The results of the sprint race.

warmingUpDate = type = ['string', 'null']
description = The date of the warming-up session in UTC.

warmingUpTime = type = ['string', 'null']
description = The start time of the warming-up session in UTC.

warmingUpResults = type = ['array', 'null']
description = The results of the warming-up session.

startingGridPositions = type = ['array', 'null']
description = The starting grid positions for the race.

raceResults = type = ['array', 'null']
description = The results of the race.

fastestLaps = type = ['array', 'null']
description = The fastest laps recorded during the race..

pitStops = type = ['array', 'null']
description = The pit stops made during the race.

driverOfTheDayResults = type = ['array', 'null']
description = The results of the Driver of the Day vote.

driverStandings = type = ['array', 'null']
description = The driver standings after the race.
```

```
constructorStandings = type = ['array', 'null']
description = The constructor standings after the race.
```

Initial Analysis

```
print(schema)
```

```
Continent = id = type = string
description = The unique identifier for the continent.

code = type = string
description = The unique code of the continent.

name = type = string
description = The name of the continent.

demonym = type = string
description = The demonym used for people from the continent.


Country = id = type = string
description = The unique identifier for the country.

alpha2Code = type = string
description = The unique ISO 3166-1 alpha-2 code of the country.

alpha3Code = type = string
description = The unique ISO 3166-1 alpha-3 code of the country.

iocCode = type = ['string', 'null']
description = The unique International Olympic Committee (IOC) code of
the country.

name = type = string
description = The name of the country.

demonym = type = ['string', 'null']
description = The demonym for citizens of the country.

continentId = type = string
description = The identifier for the continent where the country is
located.


Driver = id = type = string
description = The unique identifier of the driver.

name = type = string
description = The name of the driver, typically used for display
```

```
purposes.

firstName = type = string
description = The given name or first name of the driver.

lastName = type = string
description = The family name or last name of the driver.

fullName = type = string
description = The full name of the driver, usually a combination of
first, middle and last names.

abbreviation = type = string
description = The three-letter abbreviation for the driver, consisting
of uppercase letters (e.g., 'SEN' for Ayrton Senna).

permanentNumber = type = ['string', 'null']
description = The permanent racing number chosen by the driver.

gender = description = The gender of the driver.

dateOfBirth = type = string
description = The birth date of the driver.

dateOfDeath = type = ['string', 'null']
description = The death date of the driver, if applicable.

placeOfBirth = type = string
description = The place of birth of the driver.

countryOfBirthCountryId = type = string
description = The identifier of the country where the driver was born.

nationalityCountryId = type = string
description = The identifier of the nationality of the driver.

secondNationalityCountryId = type = ['string', 'null']
description = The identifier of the second nationality of the driver,
if applicable.

familyRelationships = type = ['array', 'null']
description = The family relationships involving the driver, such as
parent or sibling relationships.

bestChampionshipPosition = type = ['integer', 'null']
description = The best finishing position achieved by the driver in a
World Drivers' Championship.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the driver
in a race.
```

```
bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the driver in a
race.

totalChampionshipWins = type = integer
description = The total number of World Drivers' Championship titles
won by the driver.

totalRaceEntries = type = integer
description = The total number of races entered by the driver.

totalRaceStarts = type = integer
description = The total number of races started by the driver.

totalRaceWins = type = integer
description = The total number of races won by the driver.

totalRaceLaps = type = integer
description = The total number of laps completed by the driver.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the
driver.

totalPoints = type = number
description = The total number of points accumulated by the driver.

totalChampionshipPoints = type = number
description = The total number of World Drivers' Championship points
accumulated by the driver.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the
driver.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the driver.

totalDriverOfTheDay = type = integer
description = The total number of Driver of the Day awards received by
the driver.

totalGrandSlams = type = integer
description = The total number of Grand Slams achieved by the driver,
defined as pole position, fastest lap, and leading every lap of the
race.


DriverFamilyRelationship = positionDisplayOrder = type = integer
description = The display order of the family relationship relative to
```

```
other relationships of the parent driver.

driverId = type = string
description = The identifier of the related driver.

type = description = The type of the family relationship.


Constructor = id = type = string
description = The unique identifier of the constructor.

name = type = string
description = The name of the constructor, typically used for display
purposes.

fullName = type = string
description = The full name of the constructor.

countryId = type = string
description = The identifier of the country of origin of the
constructor.

chronology = type = ['array', 'null']
description = The chronology of the constructor.

bestChampionshipPosition = type = ['integer', 'null']
description = The best finishing position achieved by the constructor
in a World Constructors' Championship.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the
constructor in a race.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the constructor
in a race.

totalChampionshipWins = type = integer
description = The total number of World Constructors' Championship
titles won by the constructor.

totalRaceEntries = type = integer
description = The total number of races entered by the constructor.

totalRaceStarts = type = integer
description = The total number of races started by the constructor.

totalRaceWins = type = integer
description = The total number of races won by the constructor.

total1And2Finishes = type = integer
```

```
description = The total number of races in which the constructor
finished in both 1st and 2nd place.

totalRaceLaps = type = integer
description = The total number of laps completed by the constructor.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the
constructor.

totalPodiumRaces = type = integer
description = The total number of races in which the constructor
finished on the podium.

totalPoints = type = number
description = The total number of points accumulated by the
constructor.

totalChampionshipPoints = type = number
description = The total number of World Constructors' Championship
points accumulated by the constructor.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the
constructor.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the constructor.


ConstructorChronology = positionDisplayOrder = type = integer
description = The display order of the constructor within the
chronological sequence of the parent constructor.

constructorId = type = string
description = The identifier of the constructor.

yearFrom = type = integer
description = The year from.

yearTo = type = ['integer', 'null']
description = The year to, or null if it is still active.


Chassis = id = type = string
description = The unique identifier of the chassis.

constructorId = type = string
description = The identifier of the constructor associated with the
chassis.
```

```
name = type = string
description = The name of the chassis.

fullName = type = string
description = The full name of the chassis.


EngineManufacturer = id = type = string
description = The unique identifier of the engine manufacturer.

name = type = string
description = The name of the engine manufacturer.

countryId = type = string
description = The identifier of the country of origin of the engine
manufacturer.

bestChampionshipPosition = type = ['integer', 'null']
description = The best finishing position achieved by the engine
manufacturer in a World Constructors' Championship.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the engine
manufacturer in a race.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the engine
manufacturer in a race.

totalChampionshipWins = type = integer
description = The total number of World Constructors' Championship
titles won by the engine manufacturer.

totalRaceEntries = type = integer
description = The total number of races entered by the engine
manufacturer.

totalRaceStarts = type = integer
description = The total number of races started by the engine
manufacturer.

totalRaceWins = type = integer
description = The total number of races won by the engine
manufacturer.

totalRaceLaps = type = integer
description = The total number of laps completed by the engine
manufacturer.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the
```

```
engine manufacturer.

totalPodiumRaces = type = integer
description = The total number of races in which the engine
manufacturer finished on the podium.

totalPoints = type = number
description = The total number of points accumulated by the engine
manufacturer.

totalChampionshipPoints = type = number
description = The total number of World Constructors' Championship
points accumulated by the engine manufacturer.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the
engine manufacturer.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the engine
manufacturer.


Engine = id = type = string
description = The unique identifier of the engine.

engineManufacturerId = type = string
description = The identifier of the engine manufacturer associated
with the engine.

name = type = string
description = The name of the engine.

fullName = type = string
description = The full name of the engine.

capacity = type = ['number', 'null']
description = The capacity of the engine, measured in liters.

configuration = description = The configuration of the engine, such as
V6, V8, etc.

aspiration = description = The aspiration of the engine, such as
naturally aspirated or turbocharged.


TyreManufacturer = id = type = string
description = The unique identifier of the tyre manufacturer.

name = type = string
description = The name of the tyre manufacturer.
```

```
countryId = type = string
description = The identifier of the country of origin of the tyre
manufacturer.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the tyre
manufacturer in a race.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the tyre
manufacturer in a race.

totalRaceEntries = type = integer
description = The total number of races entered by the tyre
manufacturer.

totalRaceStarts = type = integer
description = The total number of races started by the tyre
manufacturer.

totalRaceWins = type = integer
description = The total number of races won by the tyre manufacturer.

totalRaceLaps = type = integer
description = The total number of laps completed by the tyre
manufacturer.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the tyre
manufacturer.

totalPodiumRaces = type = integer
description = The total number of races in which the tyre manufacturer
finished on the podium.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the tyre
manufacturer.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the tyre
manufacturer.


Circuit = id = type = string
description = The unique identifier of the circuit.

name = type = string
description = The name of the circuit, typically used for display
purposes.
```

```
fullName = type = string
description = The full official name of the circuit.

previousNames = type = ['array', 'null']
description = The previous names used for the circuit.

type = description = The type of the circuit.

direction = description = The direction of the current or most
recently used configuration of the circuit.

placeName = type = string
description = The place name where the circuit is located.

countryId = type = string
description = The identifier of the country where the circuit is
located.

latitude = type = number
description = The latitude coordinate where the circuit is located.

longitude = type = number
description = The longitude coordinate where the circuit is located.

length = type = number
description = The length of the current or most recently used
configuration of the circuit in kilometers.

turns = type = integer
description = The number of turns (corners) in the current or most
recently used configuration of the circuit.

totalRacesHeld = type = integer
description = The total number of races held at the circuit.


GrandPrix = id = type = string
description = The unique identifier of the Grand Prix.

name = type = string
description = The name of the Grand Prix, typically used for display
purposes.

fullName = type = string
description = The full name of the Grand Prix.

shortName = type = string
description = The short name of the Grand Prix.

abbreviation = type = string
```

```
description = The three-character abbreviation of the Grand Prix.

countryId = type = ['string', 'null']
description = The identifier of the country where the Grand Prix is
held.

totalRacesHeld = type = integer
description = The total number of races held for this Grand Prix.


SeasonEntrant = entrantId = type = string
description = The identifier of the entrant.

countryId = type = string
description = The identifier of the country of the entrant.

constructors = type = array
description = The constructors associated with the entrant.


Entrant = id = type = string
description = The unique identifier of the entrant.

name = type = string
description = The name of the entrant.


Season = year = type = integer
description = The year of the season.

entrants = type = ['array', 'null']
description = The entrants competing in the season.

constructors = type = ['array', 'null']
description = The constructors competing in the season.

engineManufacturers = type = ['array', 'null']
description = The engine manufacturers competing in the season.

tyreManufacturers = type = ['array', 'null']
description = The tyre manufacturers competing in the season.

drivers = type = ['array', 'null']
description = The drivers competing in the season.

driverStandings = type = ['array', 'null']
description = The driver standings of the season.

constructorStandings = type = ['array', 'null']
description = The constructor standings of the season.
```

```
SeasonEntrantConstructor = constructorId = type = string
description = The identifier of the constructor.

engineManufacturerId = type = string
description = The identifier of the engine manufacturer.

chassis = type = array
description = The chassis used by the constructor.

engines = type = array
description = The engines used by the constructor.

tyreManufacturers = type = array
description = The tyre manufacturers used by the constructor.

drivers = type = ['array', 'null']
description = The drivers who drove for the constructor.


SeasonEntrantChassis = chassisId = type = string
description = The identifier of the chassis.


SeasonEntrantEngine = engineId = type = string
description = The identifier of the engine.


SeasonEntrantTyreManufacturer = tyreManufacturerId = type = string
description = The identifier of the tyre manufacturer.


SeasonEntrantDriver = driverId = type = string
description = The identifier of the driver.

rounds = type = ['array', 'null']
description = The rounds in which the driver participated.

roundsText = type = ['string', 'null']
description = The textual representation of the rounds in which the
driver participated.

testDriver = type = boolean
description = Whether the driver was a test / free practice driver.


SeasonConstructor = year = type = integer
description = The year of the season.

constructorId = type = string
description = The identifier of the constructor.
```

```
positionNumber = type = ['integer', 'null']
description = The numerical position of the constructor in the season
standings.

positionText = type = ['string', 'null']
description = The textual representation of the constructor's position
in the season standings, including special statuses.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the
constructor during the season.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the constructor
during the season.

totalRaceEntries = type = integer
description = The total number of races entered by the constructor
during the season.

totalRaceStarts = type = integer
description = The total number of races started by the constructor
during the season.

totalRaceWins = type = integer
description = The total number of races won by the constructor during
the season.

total1And2Finishes = type = integer
description = The total number of races in which the constructor
finished in both 1st and 2nd place during the season.

totalRaceLaps = type = integer
description = The total number of laps completed by the constructor
during the season.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the
constructor during the season.

totalPodiumRaces = type = integer
description = The total number of races in which the constructor
finished on the podium during the season.

totalPoints = type = number
description = The total number of points accumulated by the
constructor during the season.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the
```

```
constructor during the season.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the constructor
during the season.


SeasonEngineManufacturer = year = type = integer
description = The year of the season.

engineManufacturerId = type = string
description = The identifier of the engine manufacturer.

positionNumber = type = ['integer', 'null']
description = The numerical position of the engine manufacturer in the
season standings.

positionText = type = ['string', 'null']
description = The textual representation of the engine manufacturer's
position in the season standings, including special statuses.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the engine
manufacturer during the season.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the engine
manufacturer during the season.

totalRaceEntries = type = integer
description = The total number of races entered by the engine
manufacturer during the season.

totalRaceStarts = type = integer
description = The total number of races started by the engine
manufacturer during the season.

totalRaceWins = type = integer
description = The total number of races won by the engine manufacturer
during the season.

totalRaceLaps = type = integer
description = The total number of laps completed by the engine
manufacturer during the season.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the
engine manufacturer during the season.

totalPodiumRaces = type = integer
description = The total number of races in which the engine
```

```
manufacturer finished on the podium during the season.

totalPoints = type = number
description = The total number of points accumulated by the engine
manufacturer during the season.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the
engine manufacturer during the season.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the engine
manufacturer during the season.


SeasonTyreManufacturer = year = type = integer
description = The year of the season.

tyreManufacturerId = type = string
description = The identifier of the tyre manufacturer.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the tyre
manufacturer during the season.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the tyre
manufacturer during the season.

totalRaceEntries = type = integer
description = The total number of races entered by the tyre
manufacturer during the season.

totalRaceStarts = type = integer
description = The total number of races started by the tyre
manufacturer during the season.

totalRaceWins = type = integer
description = The total number of races won by the tyre manufacturer
during the season.

totalRaceLaps = type = integer
description = The total number of laps completed by the tyre
manufacturer during the season.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the tyre
manufacturer during the season.

totalPodiumRaces = type = integer
description = The total number of races in which the tyre manufacturer
```

```
finished on the podium during the season.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the tyre
manufacturer during the season.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the tyre
manufacturer during the season.


SeasonDriver = year = type = integer
description = The year of the season.

driverId = type = string
description = The identifier of the driver.

positionNumber = type = ['integer', 'null']
description = The numerical position of the driver in the season
standings.

positionText = type = ['string', 'null']
description = The textual representation of the driver's position in
the season standings, including special statuses.

bestStartingGridPosition = type = ['integer', 'null']
description = The best starting grid position achieved by the driver
during the season.

bestRaceResult = type = ['integer', 'null']
description = The best finishing position achieved by the driver
during the season.

totalRaceEntries = type = integer
description = The total number of races entered by the driver during
the season.

totalRaceStarts = type = integer
description = The total number of races started by the driver during
the season.

totalRaceWins = type = integer
description = The total number of races won by the driver during the
season.

totalRaceLaps = type = integer
description = The total number of laps completed by the driver during
the season.

totalPodiums = type = integer
description = The total number of podium finishes achieved by the
```

```
driver during the season.

totalPoints = type = number
description = The total number of points accumulated by the driver
during the season.

totalPolePositions = type = integer
description = The total number of pole positions achieved by the
driver during the season.

totalFastestLaps = type = integer
description = The total number of fastest laps set by the driver
during the season.

totalDriverOfTheDay = type = integer
description = The total number of Driver of the Day awards received by
the driver during the season.

totalGrandSlams = type = integer
description = The total number of Grand Slams achieved by the driver
during the season, defined as pole position, fastest lap, and leading
every lap of the race.


SeasonDriverStanding = positionDisplayOrder = type = integer
description = The display order of the driver's position in the
standings.

positionNumber = type = ['integer', 'null']
description = The numerical position of the driver in the standings.

positionText = type = string
description = The textual representation of the driver's position in
the standings, including special statuses.

driverId = type = string
description = The identifier of the driver.

points = type = number
description = The total number of points determining the driver's
position in the standings.


SeasonConstructorStanding = positionDisplayOrder = type = integer
description = The display order for the constructor's position in the
standings.

positionNumber = type = ['integer', 'null']
description = The numerical position of the constructor in the
standings.
```

```
positionText = type = string
description = The textual representation of the constructor's position
in the standings, including special statuses.

constructorId = type = string
description = The identifier of the constructor.

engineManufacturerId = type = string
description = The identifier of the engine manufacturer.

points = type = number
description = The total number of points determining the constructor's
position in the standings.


Race = id = type = integer
description = The unique identifier of the race.

year = type = integer
description = The year of the season.

round = type = integer
description = The round number of the race in the season.

date = type = string
description = The date of the race in UTC.

time = type = ['string', 'null']
description = The start time of the race in UTC.

grandPrixId = type = string
description = The identifier of the Grand Prix associated with the
race.

officialName = type = string
description = The official name of the race.

qualifyingFormat = description = The qualifying format of the race.

sprintQualifyingFormat = description = The sprint qualifying format of
the race.

circuitId = type = string
description = The identifier of the circuit where the race takes
place.

circuitType = description = The type of the circuit.

direction = description = The direction of the circuit.

courseLength = type = number
```

```
description = The length of the circuit (race course) in kilometers.

turns = type = integer
description = The number of turns (corners) in the configuration of
the circuit.

laps = type = integer
description = The total number of laps of the race.

distance = type = number
description = The total distance of the race in kilometers.

scheduledLaps = type = ['integer', 'null']
description = The scheduled number of laps of the race.

scheduledDistance = type = ['number', 'null']
description = The scheduled distance of the race in kilometers.

driversChampionshipDecider = type = boolean
description = Whether this race was the decider of the World Drivers'
Championship.

constructorsChampionshipDecider = type = boolean
description = Whether this race was the decider of the World
Constructors' Championship.

preQualifyingDate = type = ['string', 'null']
description = The date of the pre-qualifying session in UTC.

preQualifyingTime = type = ['string', 'null']
description = The start time of the pre-qualifying session in UTC.

preQualifyingResults = type = ['array', 'null']
description = The results of the pre-qualifying session.

freePractice1Date = type = ['string', 'null']
description = The date of the 1st free practice session in UTC.

freePractice1Time = type = ['string', 'null']
description = The start time of the 1st free practice session in UTC.

freePractice1Results = type = ['array', 'null']
description = The results of the 1st free practice session.

freePractice2Date = type = ['string', 'null']
description = The date of the 2nd free practice session in UTC.

freePractice2Time = type = ['string', 'null']
description = The start time of the 2nd free practice session in UTC.

freePractice2Results = type = ['array', 'null']
```

```
description = The results of the 2nd free practice session.

freePractice3Date = type = ['string', 'null']
description = The date of the 3rd free practice session in UTC.

freePractice3Time = type = ['string', 'null']
description = The start time of the 3rd free practice session in UTC.

freePractice3Results = type = ['array', 'null']
description = The results of the 3rd free practice session.

freePractice4Date = type = ['string', 'null']
description = The date of the 4th free practice session UTC.

freePractice4Time = type = ['string', 'null']
description = The start time of the 4th free practice session in UTC.

freePractice4Results = type = ['array', 'null']
description = The results of the 4th free practice session.

qualifying1Date = type = ['string', 'null']
description = The date of the 1st qualifying session in UTC.

qualifying1Time = type = ['string', 'null']
description = The start time of the 1st qualifying session in UTC.

qualifying1Results = type = ['array', 'null']
description = The results of the 1st qualifying session.

qualifying2Date = type = ['string', 'null']
description = The date of the 2nd qualifying session UTC.

qualifying2Time = type = ['string', 'null']
description = The start time of the 2nd qualifying session UTC.

qualifying2Results = type = ['array', 'null']
description = The results of the 2nd qualifying session.

qualifyingDate = type = ['string', 'null']
description = The date of the qualifying session UTC.

qualifyingTime = type = ['string', 'null']
description = The start time of the qualifying session UTC.

qualifyingResults = type = ['array', 'null']
description = The results of the qualifying session.

sprintQualifyingDate = type = ['string', 'null']
description = The date of the sprint qualifying session in UTC.

sprintQualifyingTime = type = ['string', 'null']
```

```
description = The start time of the sprint qualifying session in UTC.

sprintQualifyingResults = type = ['array', 'null']
description = The results of the sprint qualifying session.

sprintStartingGridPositions = type = ['array', 'null']
description = The starting grid positions for the sprint race.

sprintRaceDate = type = ['string', 'null']
description = The date of the sprint race in UTC.

sprintRaceTime = type = ['string', 'null']
description = The start time of the sprint race in UTC.

sprintRaceResults = type = ['array', 'null']
description = The results of the sprint race.

warmingUpDate = type = ['string', 'null']
description = The date of the warming-up session in UTC.

warmingUpTime = type = ['string', 'null']
description = The start time of the warming-up session in UTC.

warmingUpResults = type = ['array', 'null']
description = The results of the warming-up session.

startingGridPositions = type = ['array', 'null']
description = The starting grid positions for the race.

raceResults = type = ['array', 'null']
description = The results of the race.

fastestLaps = type = ['array', 'null']
description = The fastest laps recorded during the race..

pitStops = type = ['array', 'null']
description = The pit stops made during the race.

driverOfTheDayResults = type = ['array', 'null']
description = The results of the Driver of the Day vote.

driverStandings = type = ['array', 'null']
description = The driver standings after the race.

constructorStandings = type = ['array', 'null']
description = The constructor standings after the race.


RaceDriverStanding = positionDisplayOrder = type = integer
description = The display order of the driver's position in the
standings.
```

```
positionNumber = type = ['integer', 'null']
description = The numerical position of the driver in the standings.

positionText = type = string
description = The textual representation of the driver's position in
the standings, including special statuses.

driverId = type = string
description = The identifier of the driver.

points = type = number
description = The total number of points determining the driver's
position in the standings.

positionsGained = type = ['integer', 'null']
description = The positions gained in the standings since the previous
race.


RaceConstructorStanding = positionDisplayOrder = type = integer
description = The display order for the constructor's position in the
standings.

positionNumber = type = ['integer', 'null']
description = The numerical position of the constructor in the
standings.

positionText = type = string
description = The textual representation of the constructor's position
in the standings, including special statuses.

constructorId = type = string
description = The identifier of the constructor.

engineManufacturerId = type = string
description = The identifier of the engine manufacturer.

points = type = number
description = The points.

positionsGained = type = ['integer', 'null']
description = The total number of points determining the constructor's
position in the standings.



race.info(verbose=True, memory_usage='deep', show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1149 entries, 0 to 1148
Data columns (total 42 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   id                                1149 non-null   int64  
 1   year                              1149 non-null   int64  
 2   round                             1149 non-null   int64  
 3   date                              1149 non-null   object 
 4   time                              48 non-null     object 
 5   grand_prix_id                     1149 non-null   object 
 6   official_name                     1149 non-null   object 
 7   qualifying_format                 1149 non-null   object 
 8   sprint_qualifying_format          18 non-null     object 
 9   circuit_id                        1149 non-null   object 
 10  circuit_type                      1149 non-null   object 
 11  direction                         1149 non-null   object 
 12  course_length                     1149 non-null   float64
 13  turns                             1149 non-null   int64  
 14  laps                              1149 non-null   int64  
 15  distance                          1149 non-null   float64
 16  scheduled_laps                    80 non-null     float64
 17  scheduled_distance                80 non-null     float64
 18  drivers_championship_decider      1149 non-null   bool   
 19  constructors_championship_decider 1149 non-null   bool   
 20  pre_qualifying_date               0 non-null      object 
 21  pre_qualifying_time               0 non-null      object 
 22  free_practice_1_date              48 non-null     object 
 23  free_practice_1_time              48 non-null     object 
 24  free_practice_2_date              36 non-null     object 
 25  free_practice_2_time              36 non-null     object 
 26  free_practice_3_date              36 non-null     object 
 27  free_practice_3_time              36 non-null     object 
 28  free_practice_4_date              0 non-null      object 
 29  free_practice_4_time              0 non-null      object 
 30  qualifying_1_date                 0 non-null      object 
 31  qualifying_1_time                 0 non-null      object 
 32  qualifying_2_date                 0 non-null      object 
 33  qualifying_2_time                 0 non-null      object 
 34  qualifying_date                   48 non-null     object 
 35  qualifying_time                   48 non-null     object 
 36  sprint_qualifying_date            12 non-null     object 
 37  sprint_qualifying_time            12 non-null     object 
 38  sprint_race_date                  12 non-null     object 
 39  sprint_race_time                  12 non-null     object 
 40  warming_up_date                   0 non-null      object 
 41  warming_up_time                   0 non-null      object 
dtypes: bool(2), float64(4), int64(5), object(31)
memory usage: 1.2 MB
```

```
race_data.info(verbose=True, memory_usage='deep', show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183632 entries, 0 to 183631
Data columns (total 71 columns):
 #   Column                                               Non-Null
Count   Dtype
---  ------
--------------   -----
 0   race_id                                              183632 non-
null  int64
 1   type                                                 183632 non-
null  object
 2   position_display_order                               183632 non-
null  int64
 3   position_number                                      172468 non-
null  float64
 4   position_text                                        183632 non-
null  object
 5   driver_number                                        183632 non-
null  object
 6   driver_id                                            183632 non-
null  object
 7   constructor_id                                       183632 non-
null  object
 8   engine_manufacturer_id                               183632 non-
null  object
 9   tyre_manufacturer_id                                 183632 non-
null  object
 10  practice_time                                        47260 non-
null   object
 11  practice_time_millis                                 47260 non-
null   float64
 12  practice_gap                                         45124 non-
null   object
 13  practice_gap_millis                                  45124 non-
null   float64
 14  practice_interval                                    45124 non-
null   object
 15  practice_interval_millis                             45124 non-
null   float64
 16  practice_laps                                        38322 non-
null   float64
 17  qualifying_time                                      33926 non-
null   object
 18  qualifying_time_millis                               33926 non-
null   float64
 19  qualifying_q1                                        8470 non-
null    object
 20  qualifying_q1_millis                                 8470 non-
```

```
null    float64
 21  qualifying_q2                                         6216 non-
null    object
 22  qualifying_q2_millis                                  6216 non-
null    float64
 23  qualifying_q3                                         3952 non-
null    object
 24  qualifying_q3_millis                                  3952 non-
null    float64
 25  qualifying_gap                                        36049 non-
null   object
 26  qualifying_gap_millis                                 36049 non-
null   float64
 27  qualifying_interval                                   36036 non-
null   object
 28  qualifying_interval_millis                            36036 non-
null   float64
 29  qualifying_laps                                       17016 non-
null   float64
 30  starting_grid_position_qualification_position_number  25680 non-
null   float64
 31  starting_grid_position_qualification_position_text    25809 non-
null   object
 32  starting_grid_position_grid_penalty                   573 non-
null     object
 33  starting_grid_position_grid_penalty_positions         500 non-
null     float64
 34  starting_grid_position_time                           25258 non-
null   object
 35  starting_grid_position_time_millis                    25258 non-
null   float64
 36  race_shared_car                                       27591 non-
null   object
 37  race_laps                                             25664 non-
null   float64
 38  race_time                                             8318 non-
null    object
 39  race_time_millis                                      8318 non-
null    float64
 40  race_time_penalty                                     274 non-
null     object
 41  race_time_penalty_millis                              274 non-
null     float64
 42  race_gap                                              14822 non-
null   object
 43  race_gap_millis                                       7154 non-
null    float64
 44  race_gap_laps                                         7668 non-
null    float64
```

```
 45  race_interval                                   7136 non-
null    object 
 46  race_interval_millis                            7136 non-
null    float64
 47  race_reason_retired                             9998 non-
null    object 
 48  race_points                                     8505 non-
null    float64
 49  race_pole_position                              27591 non-
null   object 
 50  race_qualification_position_number              26872 non-
null   float64
 51  race_qualification_position_text                27009 non-
null   object 
 52  race_grid_position_number                       25584 non-
null   float64
 53  race_grid_position_text                         25815 non-
null   object 
 54  race_positions_gained                           16626 non-
null   float64
 55  race_pit_stops                                  12676 non-
null   float64
 56  race_fastest_lap                                27571 non-
null   object 
 57  race_driver_of_the_day                          4601 non-
null    object 
 58  race_grand_slam                                 27591 non-
null   object 
 59  fastest_lap_lap                                 16689 non-
null   float64
 60  fastest_lap_time                                16736 non-
null   object 
 61  fastest_lap_time_millis                         16736 non-
null   float64
 62  fastest_lap_gap                                 15593 non-
null   object 
 63  fastest_lap_gap_millis                          15593 non-
null   float64
 64  fastest_lap_interval                            15593 non-
null   object 
 65  fastest_lap_interval_millis                     15593 non-
null   float64
 66  pit_stop_stop                                   21889 non-
null   float64
 67  pit_stop_lap                                    21889 non-
null   float64
 68  pit_stop_time                                   21888 non-
null   object 
 69  pit_stop_time_millis                            21888 non-
```

```
null    float64
 70  driver_of_the_day_percentage                          720 non-
null     float64
dtypes: float64(34), int64(2), object(35)
memory usage: 253.4 MB
```

```
circuit.info(verbose=True, memory_usage='deep', show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77 entries, 0 to 76
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 77 non-null     object
 1   name               77 non-null     object
 2   full_name          77 non-null     object
 3   previous_names     15 non-null     object
 4   type               77 non-null     object
 5   direction          77 non-null     object
 6   place_name         77 non-null     object
 7   country_id         77 non-null     object
 8   latitude           77 non-null     float64
 9   longitude          77 non-null     float64
 10  length             77 non-null     float64
 11  turns              77 non-null     int64
 12  total_races_held   77 non-null     int64
dtypes: float64(3), int64(2), object(8)
memory usage: 37.4 KB
```

```
circuit.id.unique()
```

```
array(['adelaide', 'aida', 'ain-diab', 'aintree', 'anderstorp',
'austin',
       'avus', 'bahrain', 'baku', 'brands-hatch', 'bremgarten',
'buddh',
       'buenos-aires', 'bugatti', 'caesars-palace', 'catalunya',
       'clermont-ferrand', 'dallas', 'detroit', 'dijon', 'donington',
       'east-london', 'estoril', 'fuji', 'hockenheimring',
'hungaroring',
       'imola', 'indianapolis', 'interlagos', 'istanbul',
'jacarepagua',
       'jarama', 'jeddah', 'jerez', 'kyalami', 'las-vegas', 'long-
beach',
       'lusail', 'magny-cours', 'marina-bay', 'melbourne', 'mexico-
city',
       'miami', 'monaco', 'monsanto', 'mont-tremblant', 'montjuic',
       'montreal', 'monza', 'mosport', 'mugello', 'nivelles',
       'nurburgring', 'paul-ricard', 'pedralbes', 'pescara',
'phoenix',
       'portimao', 'porto', 'reims', 'riverside', 'rouen', 'sebring',
       'sepang', 'shanghai', 'silverstone', 'sochi', 'spa-
francorchamps',
       'spielberg', 'suzuka', 'valencia', 'watkins-glen', 'yas-
marina',
       'yeongam', 'zandvoort', 'zeltweg', 'zolder'], dtype=object)

circuit.sort_values(by='total_races_held', ascending=False)
[['id','total_races_held']].head()

# Circuits that have been repetitively raced on will show the most
conclusive results

                   id  total_races_held
48              monza                75
43             monaco                71
65        silverstone                60
67   spa-francorchamps               58
47           montreal                44
```

Initial Notes:

- constructors_championship_decider is used for postgres functionality, and therefore can be dropped from the dataframe for analysis

Circuit.previous_name is arbitrary info and can be dropped from the dataframe Circuit.name and index are the same values, names can be dropped Circuit.total_races_held can be simplified to be called 'races_count'

```
circuit =
circuit.drop(columns=['name','previous_names','direction','latitude','
longitude'])
circuit = circuit.rename(columns = {
    'total_races_held' : 'race_count'
})
```

Preparing the Data

```
tracks = ['monza', 'monaco', 'silverstone', 'spa-francorchamps',
'montreal'] # Circuits of interest; have the most amount of racing
data to analyze

race[['date']] = race[['date']].apply(pd.to_datetime, format="%Y-%m-
%d")


def query_circuit(circuit_name: str, agg_method : str = 'min') ->
pd.DataFrame:

    selected_data = (race.query(f'circuit_id == "{circuit_name}"')
                    .rename(columns={'id' : 'race_id'}))
```

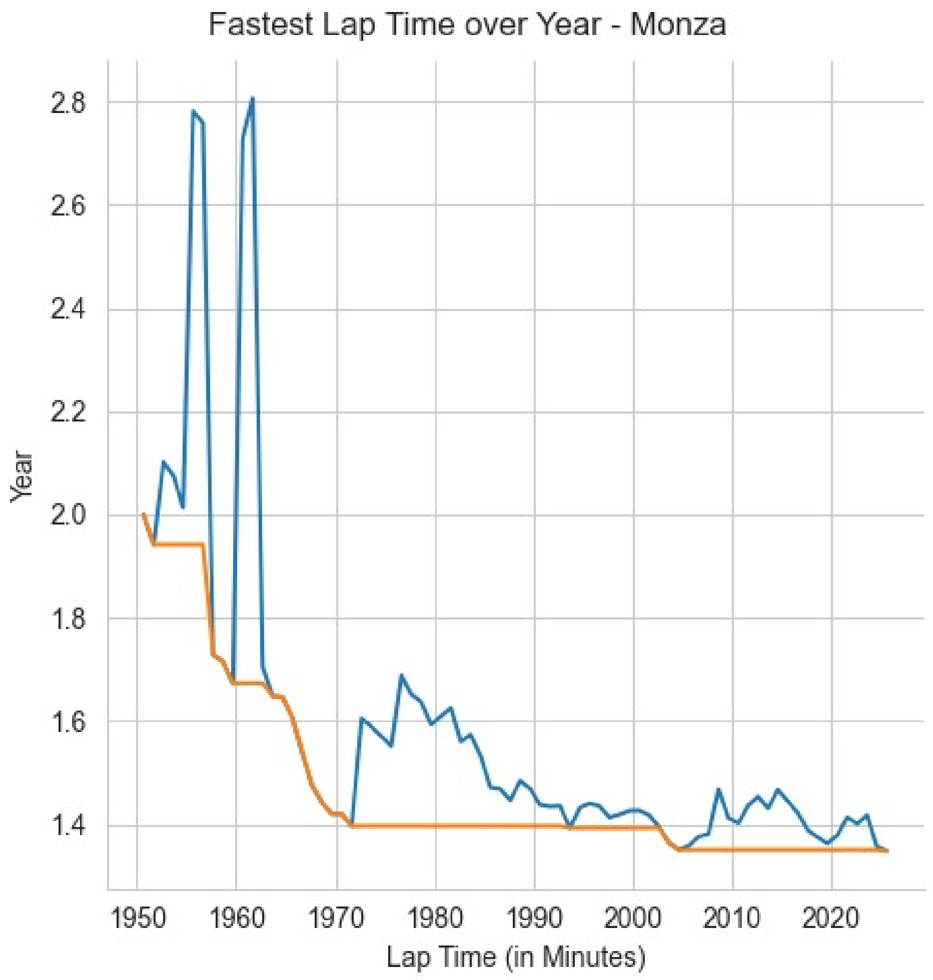
```
# Change id name to match with race_data for merge

    selected_data = (
        selected_data
            .drop(columns=['time','grand_prix_id',
'circuit_id' ,'circuit_type', 'direction'])      # Remove redundant
information (same for all rows)
            .dropna(axis=1, how='all')
# Remove non-applicable information (null for all rows)

            .merge(right=race_data, on='race_id', how='left')
# Merge selected data and race_data to obtain lap time data
            .rename(columns={'fastest_lap_time_millis' :
'fastest_time'})

            .groupby('race_id')[['fastest_time']]
            .agg(agg_method)
# apply aggregation

            .merge(right=selected_data[['race_id', 'date',
'circuit_id']].copy(), on='race_id', how='left' )      # Merge
selected data back to maintain date column
            .drop(columns='race_id')
        )

    selected_data['fastest_time'] = selected_data['fastest_time'] /
60000                           # Convert to minutes
    selected_data = selected_data.sort_values(by='date')
    # selected_data =
selected_data.set_index('date').resample('YE').agg(agg_method).reset_i
ndex()
    selected_data['record'] =
selected_data['fastest_time'].expanding().min()
# Calculate running minimum

    selected_data = selected_data.melt(id_vars=['circuit_id', 'date'],
value_vars=['fastest_time','record'])        # melt data for plotting

    return selected_data

lap_data_min = pd.concat([query_circuit(table) for table in tracks])
# lap_data_avg = pd.concat([query_circuit(table, 'mean') for table in
tracks])

lap_data_min.head()
```

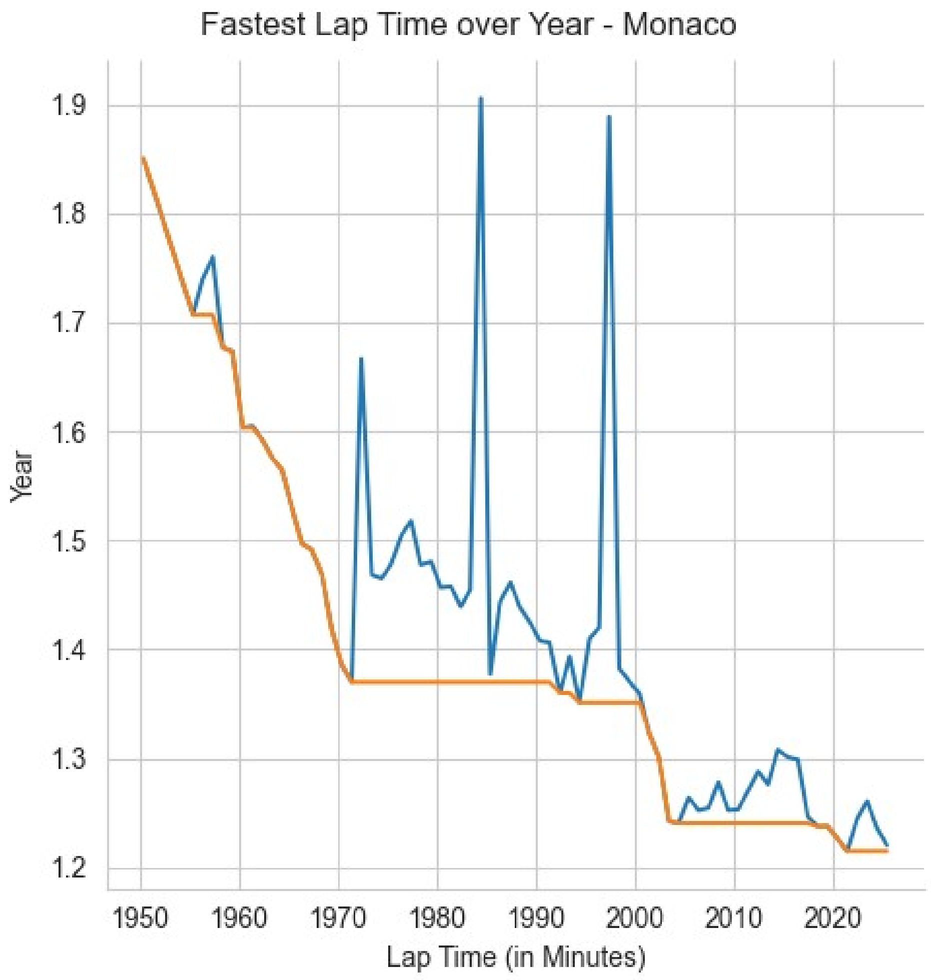
```
  circuit_id       date      variable     value
0      monza 1950-09-03  fastest_time  2.000000
1      monza 1951-09-16  fastest_time  1.941667
2      monza 1952-09-07  fastest_time  2.101667
```

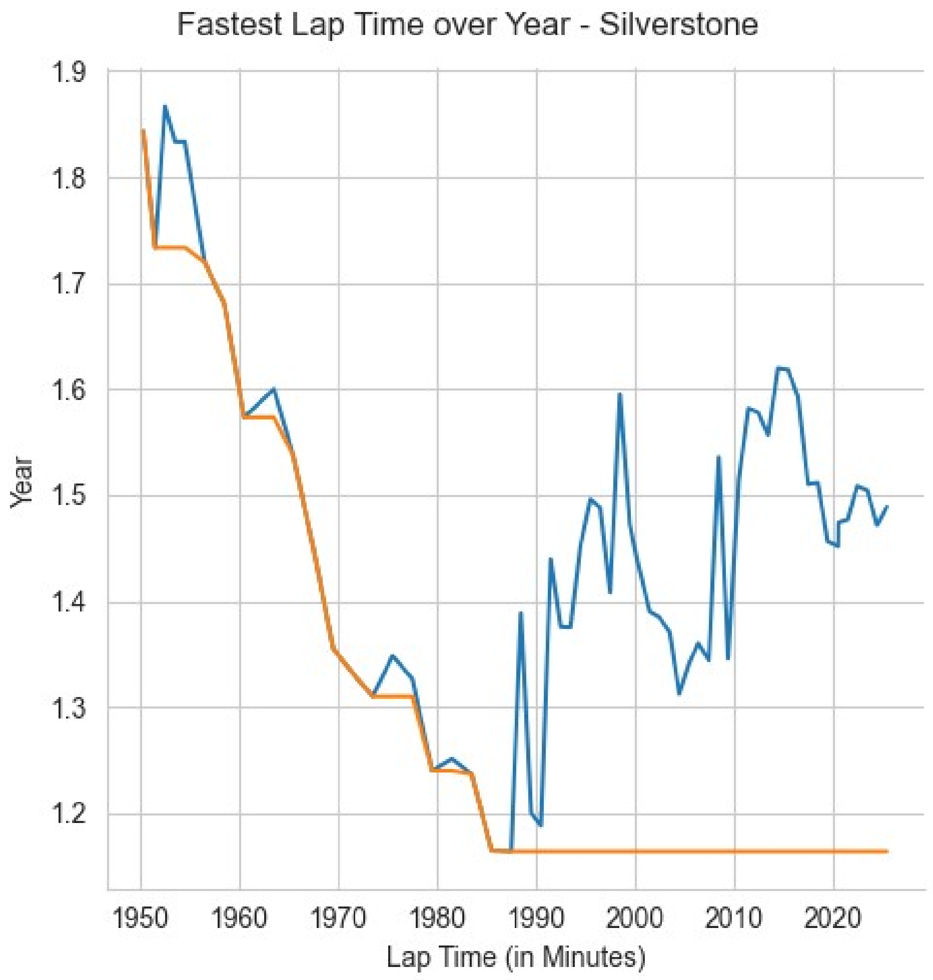
```
3      monza 1953-09-13  fastest_time  2.075000
4      monza 1954-09-05  fastest_time  2.013333

# lap_data_avg.head()

data_percent_change = (
    lap_data_min
    .query('variable == "record"')
    .set_index(['circuit_id', 'date'])
    .groupby(level=0)['value']
    .pct_change()                       # Apply percentage
    .apply(lambda x: -1 * x)            # Reverse so all data is
positive
    .to_frame()
    .reset_index()
    .set_index('date',)
)

data_percent_change.head()

df = []

# Downsize data to yearly for clarity

for track in tracks:
    temp = data_percent_change.query(f'circuit_id == "{track}"')
    temp = temp.value.resample('YS').mean().to_frame()
    temp['circuit_id'] = track
    temp = temp.reset_index()
    df.append(temp)

data_percent_change = pd.concat(df) # Recompile into a single
DataFrame (so graphing functions can be used)
data_percent_change.head()

        date     value circuit_id
0 1950-01-01       NaN      monza
1 1951-01-01  0.029167      monza
2 1952-01-01  0.000000      monza
3 1953-01-01  0.000000      monza
4 1954-01-01  0.000000      monza
```

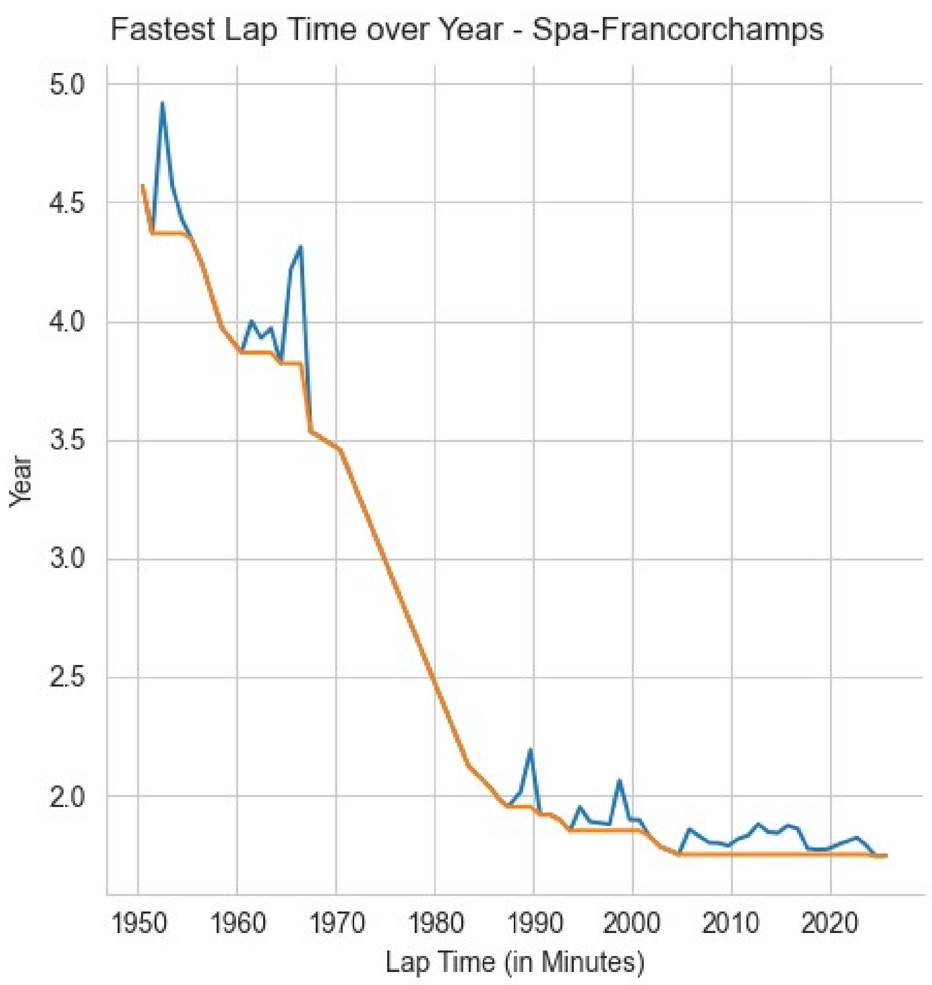
Plotting the data

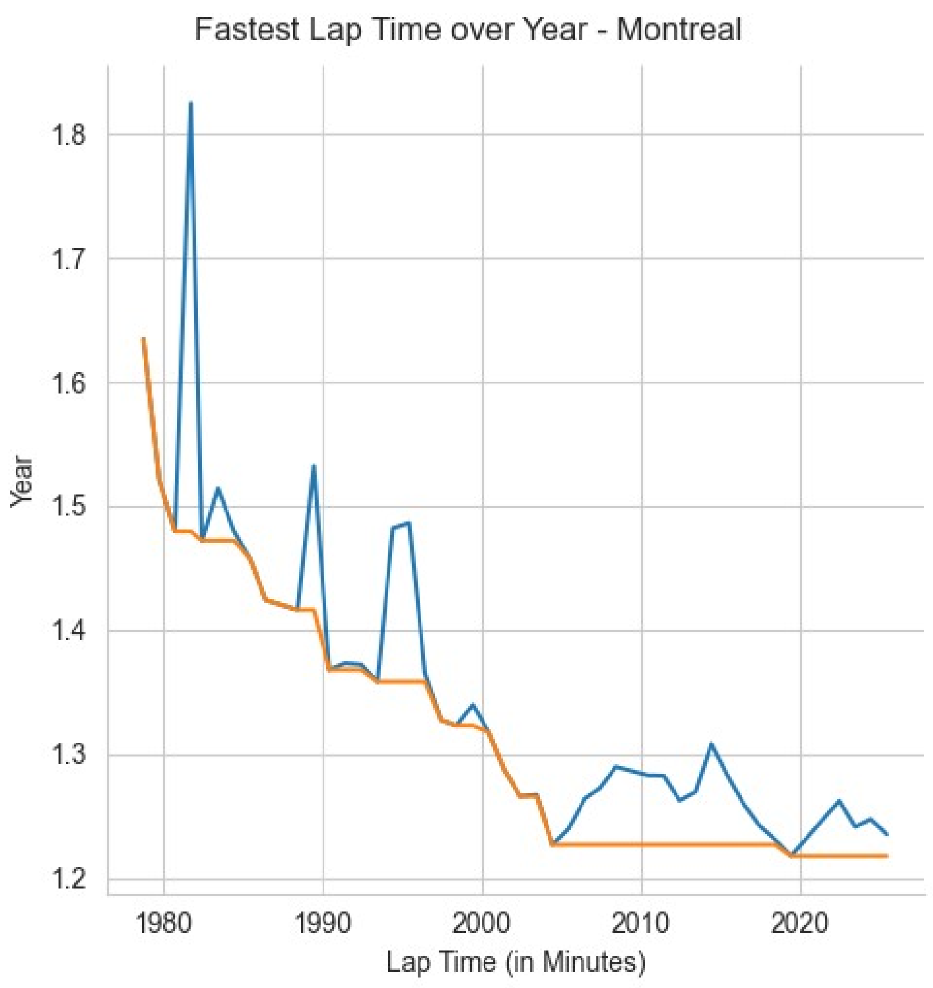
```
def create_plot ( data_set : pd.DataFrame, super_title : str ) ->
sns.FacetGrid:
    # Create single figure with multiple plots
    g = sns.relplot(data=data_set,
                kind="line", x='date', y='value', hue='variable',
                col='circuit_id', facet_kws={ 'sharex' : False,
'sharey' : False }
    )
    g.set_axis_labels('Year', 'Lap Time (in minutes)')
    g.figure.suptitle(super_title, y = 1.02)

    for ax in g.axes.flat:
            ax.set_title(ax.get_title()[13:].title())
    return g

path = 'charts/circuit_data/'
extension = '.png'

def save_plot (
        data_set : pd.DataFrame,
        super_title : str,
        subfolder : str = '',
        x_label : str = "Lap Time (in Minutes)",
        y_label : str = "Year",
        y_lim : float = None,
):
    # Save individual plots to directory

    sns.set_style('whitegrid')

    for cir in lap_data_min.circuit_id.unique().tolist():

        g = sns.relplot(
            data=data_set.query(f'circuit_id == "{cir}"'),
            kind='line',
            x='date',
            y='value',
            hue= 'variable' if 'variable' in data_set.columns else
None,
            # legend=False      # Used for presentation
        )

        g.figure.suptitle(f'{super_title} - {cir.title()}', y = 1.02)

        g.set_axis_labels(
            x_label,
            y_label
        )

        for ax in g.axes.flat:
            ax.set(ylim = (0, y_lim) if y_lim is not None else None)

        file = path + subfolder + cir + extension
        g.savefig(file)

save_plot(lap_data_min, "Fastest Lap Time over Year")
```


Fastest Lap Time over Year - Monza
Year
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1950
1960
1970
1980
1990
2000
2010
2020
Lap Time (in Minutes)

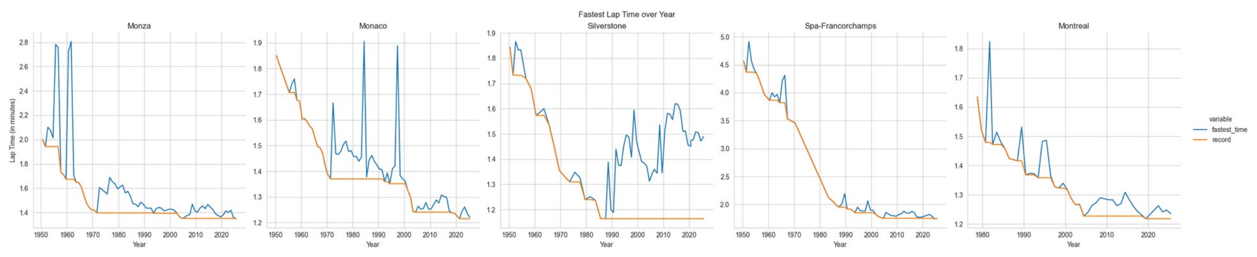

Fastest Lap Time over Year - Monaco
Year
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1950
1960
1970
1980
1990
2000
2010
2020
Lap Time (in Minutes)


Fastest Lap Time over Year - Silverstone
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
Year
1950
1960
1970
1980
1990
2000
2010
2020
Lap Time (in Minutes)

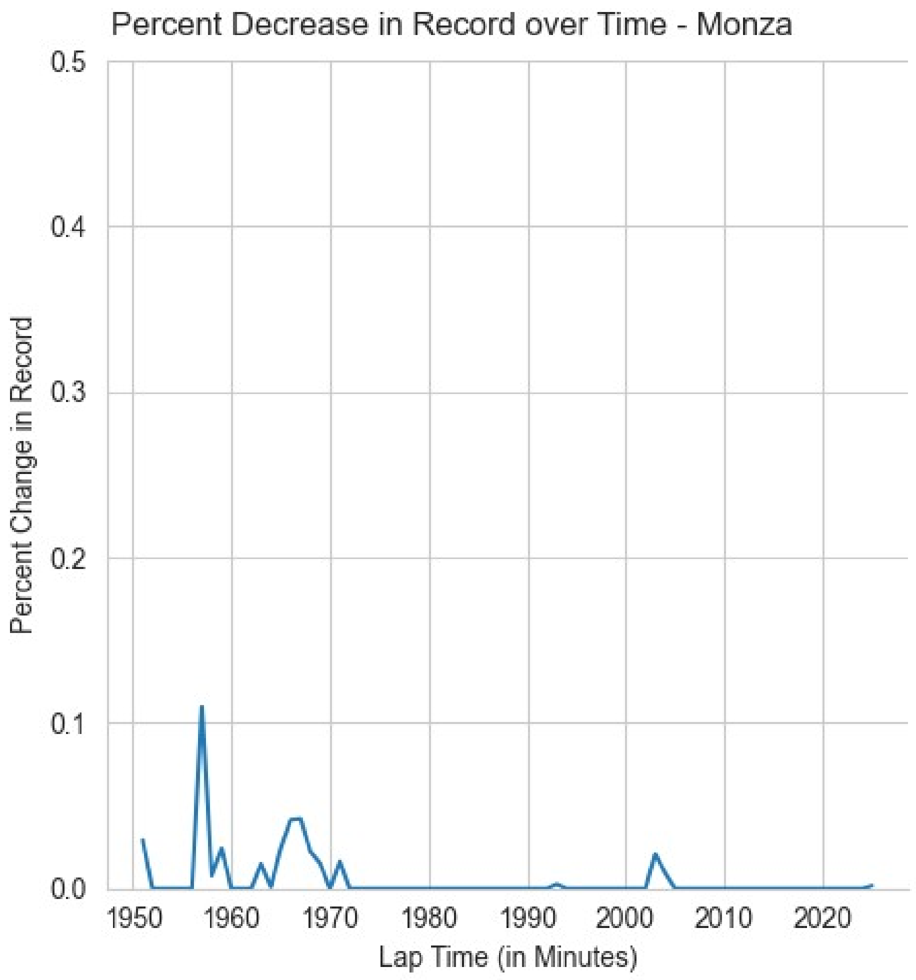

Fastest Lap Time over Year - Spa-Francorchamps
Year
Lap Time (in Minutes)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1950
1960
1970
1980
1990
2000
2010
2020

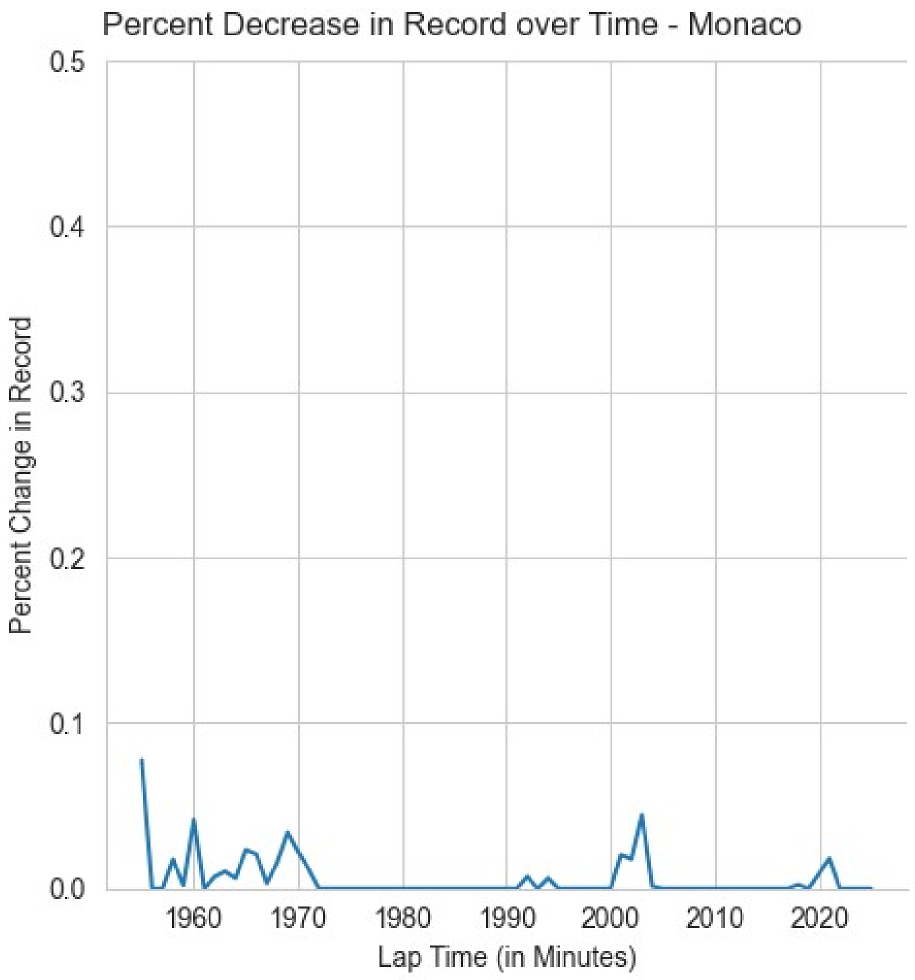


```
create_plot(lap_data_min, "Fastest Lap Time over Year")

<seaborn.axisgrid.FacetGrid at 0x16b3a3750>
```

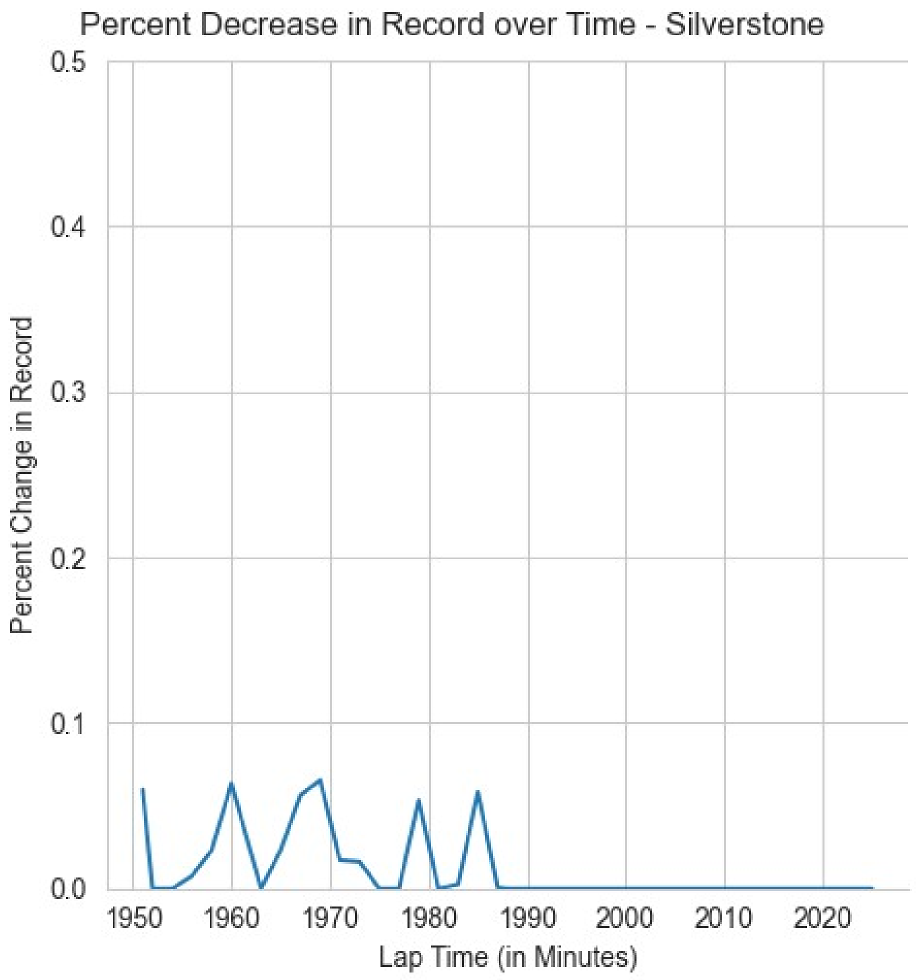


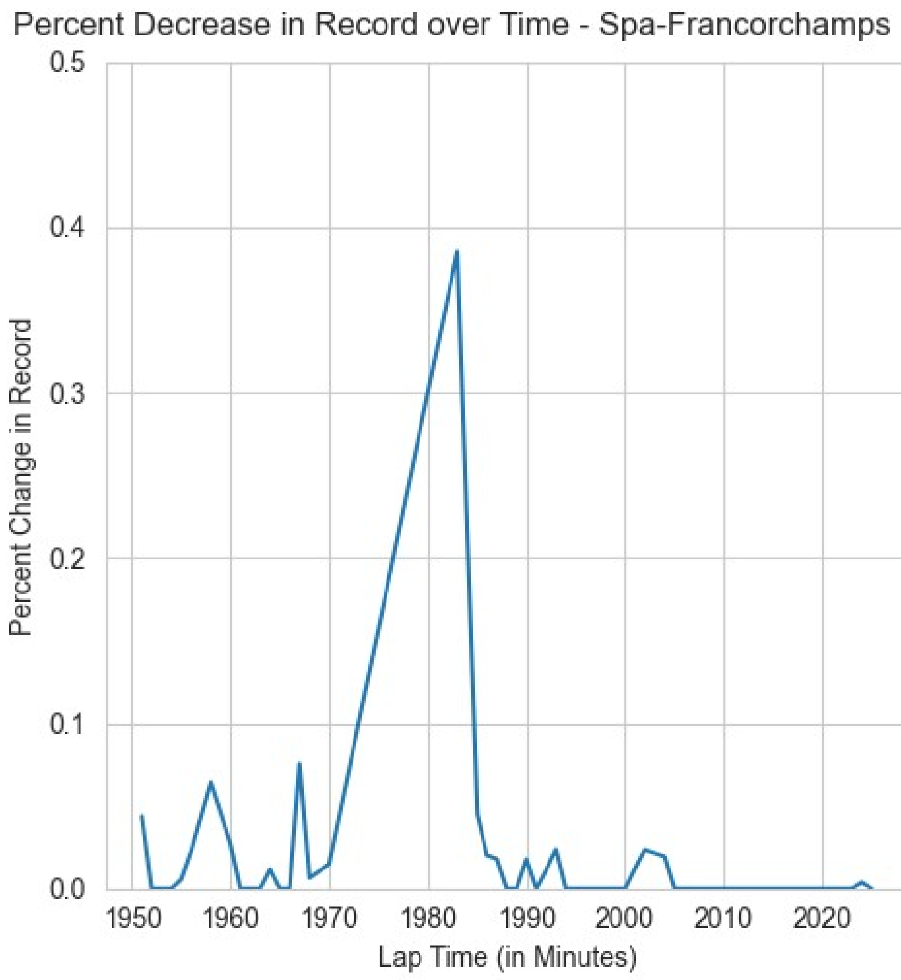
```
save_plot(
    data_set=data_percent_change,
    super_title="Percent Decrease in Record over Time",
    subfolder='pct_change/',
    y_label='Percent Change in Record',
    y_lim = .5
)
```

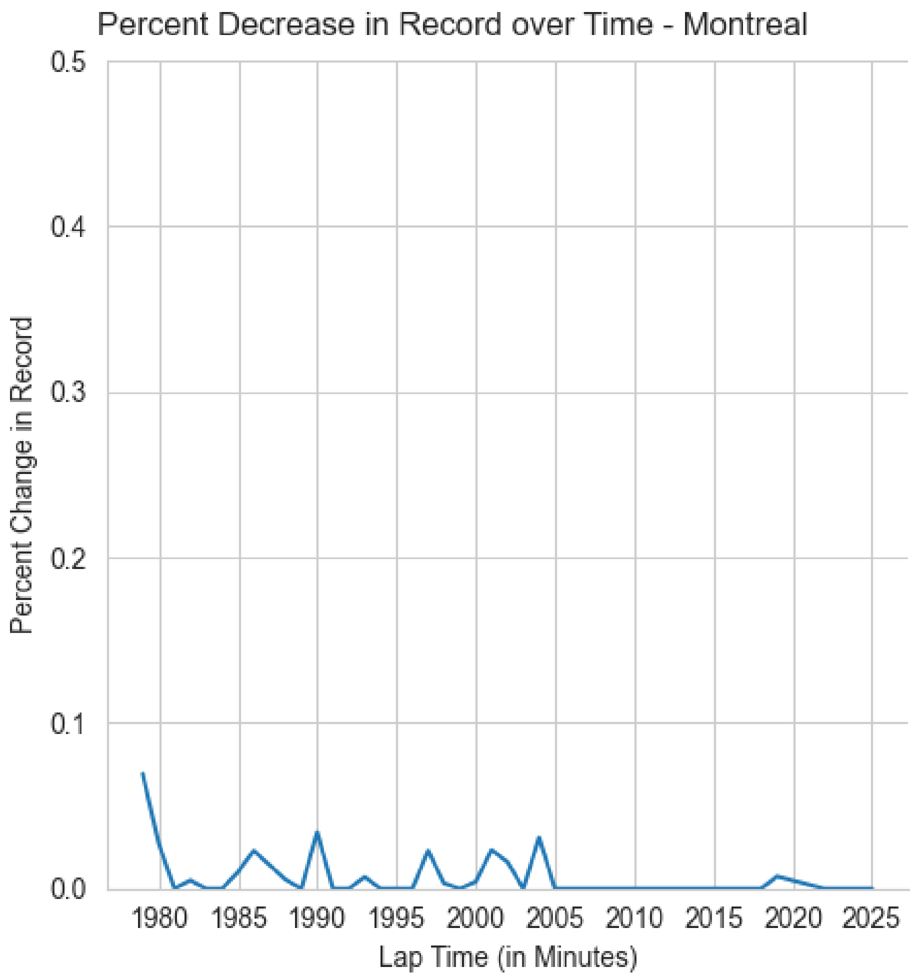

Percent Decrease in Record over Time - Monza
0.5
0.4
0.3
0.2
0.1
0.0
Percent Change in Record
1950
1960
1970
1980
1990
2000
2010
2020
Lap Time (in Minutes)



Percent Decrease in Record over Time - Monaco
Percent Change in Record
Lap Time (in Minutes)
0.0
0.1
0.2
0.3
0.4
0.5
1960
1970
1980
1990
2000
2010
2020


Percent Decrease in Record over Time - Silverstone
0.5
0.4
0.3
0.2
0.1
0.0
Percent Change in Record
1950
1960
1970
1980
1990
2000
2010
2020
Lap Time (in Minutes)


Percent Decrease in Record over Time - Spa-Francorchamps
Percent Change in Record
Lap Time (in Minutes)
0.0
0.1
0.2
0.3
0.4
0.5
1950
1960
1970
1980
1990
2000
2010
2020


Percent Decrease in Record over Time - Montreal
0.5
0.4
0.3
0.2
0.1
0.0
Percent Change in Record
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
Lap Time (in Minutes)

Tyre Data Analysis

Goal: analyze all races, lap times (via results/fastest laps), pit stops, and all tyre manufacturers.

Import data and Modules

```
conda install psycopg2
```

```
2 channel Terms of Service accepted
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 25.5.1
    latest version: 25.11.0

Please update conda by running

    $ conda update -n base -c defaults conda



# All requested packages already installed.


Note: you may need to restart the kernel to use updated packages.
```

```
from numpy.ma.extras import unique
import matplotlib.pyplot as plt


import pandas as pd
import seaborn as sns

path = 'resources/pickled_tables/'
extension = '.plk'

tire_data_table = "tyre_manufacturer"
tyre_data_file = path + tire_data_table + extension
tire_data = pd.read_pickle(tyre_data_file)

print(f"Loaded data shape: {tire_data.shape}")
tire_data.head()
```

```
Loaded data shape: (9, 13)

            id         name      country_id
best_starting_grid_position  \
0         avon         Avon  united-kingdom
2
1  bridgestone  Bridgestone           japan
1
2  continental  Continental         germany
1
3       dunlop       Dunlop  united-kingdom
1
4    englebert    Englebert         belgium
1

   best_race_result  total_race_entries  total_race_starts
total_race_wins  \
0                 5                  32                 28
0
1                 1                 244                244
175
2                 1                  13                 13
10
3                 1                 177                175
84
4                 1                  60                 60
8

   total_race_laps  total_podiums  total_podium_races
total_pole_positions  \
0             2961              0                   0
0
1           173435            482                 209
168
2             2232             18                  11
8
3            84697            241                 104
77
4            11015             40                  26
11

   total_fastest_laps
0                   0
1                 170
2                   9
3                  83
4                  12

tire_data.head(100)
```

```
            id         name                country_id  \
0         avon         Avon            united-kingdom
1  bridgestone  Bridgestone                     japan
2  continental  Continental                   germany
3       dunlop       Dunlop            united-kingdom
4    englebert    Englebert                   belgium
5    firestone    Firestone  united-states-of-america
6     goodyear     Goodyear  united-states-of-america
7     michelin     Michelin                    france
8      pirelli      Pirelli                     italy

   best_starting_grid_position  best_race_result
total_race_entries  \
0                            2                 5                  32

1                            1                 1                 244

2                            1                 1                  13

3                            1                 1                 177

4                            1                 1                  60

5                            1                 1                 122

6                            1                 1                 493

7                            1                 1                 217

8                            1                 1                 509


   total_race_starts  total_race_wins  total_race_laps  total_podiums
\
0                 28                0             2961              0

1                244              175           173435            482

2                 13               10             2232             18

3                175               84            84697            241

4                 60                8            11015             40

5                122               48            96610            138

6                493              368           376316           1139

7                215              102            99137            317

8                504              348           398748           1053
```

```
   total_podium_races  total_pole_positions  total_fastest_laps
0                   0                     0                   0
1                 209                   168                 170
2                  11                     8                   9
3                 104                    77                  83
4                  26                    11                  12
5                  77                    59                  52
6                 459                   358                 364
7                 179                   111                 108
8                 374                   351                 361
```

Show basic stats

```
print(f"Number of tyre manufacturers: {len(tire_data)}")
```

```
Number of tyre manufacturers: 9
```

Sort by total wins to see the best manufacturers

```
sorted_by_wins = tire_data.sort_values('total_race_wins',
ascending=False)
print("\nTyre Manufacturers sorted by total wins:")
print(sorted_by_wins[['name', 'total_race_wins',
'total_race_entries']].head(10))
```

```
Tyre Manufacturers sorted by total wins:
          name  total_race_wins  total_race_entries
6     Goodyear              368                 493
8      Pirelli              348                 509
1  Bridgestone              175                 244
7     Michelin              102                 217
3       Dunlop               84                 177
5    Firestone               48                 122
2  Continental               10                  13
4    Englebert                8                  60
0         Avon                0                  32
```

Calculate win percentage

```
tire_data['win_percentage'] = (tire_data['total_race_wins'] /
tire_data['total_race_entries'] * 100)
sorted_by_win_percentage = tire_data.sort_values('win_percentage',
ascending=False)
print("\nTyre Manufacturers sorted by win percentage:")
print(sorted_by_win_percentage[['name', 'win_percentage',
'total_race_wins', 'total_race_entries']].head(10))
```

```
Tyre Manufacturers sorted by win percentage:
          name  win_percentage  total_race_wins  total_race_entries
2  Continental       76.923077               10                  13
6     Goodyear       74.645030              368                 493
1  Bridgestone       71.721311              175                 244
8      Pirelli       68.369352              348                 509
3       Dunlop       47.457627               84                 177
7     Michelin       47.004608              102                 217
5    Firestone       39.344262               48                 122
4    Englebert       13.333333                8                  60
0         Avon        0.000000                0                  32
```

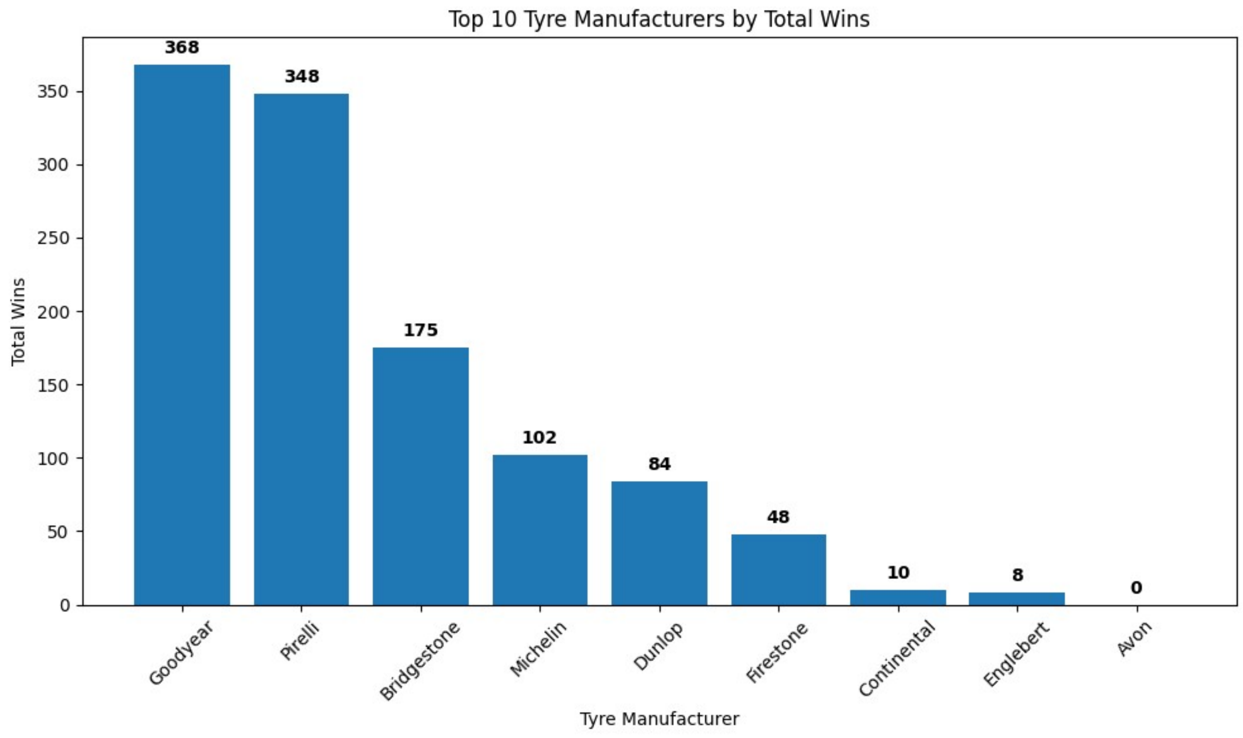
Sort by total races

```
sorted_by_races = tire_data.sort_values('total_race_entries',
ascending=False)
print("\nTyre Manufacturers sorted by total race entries:")
print(sorted_by_races[['name', 'total_race_entries',
'total_race_wins']].head(10))
```

```
Tyre Manufacturers sorted by total race entries:
          name  total_race_entries  total_race_wins
8      Pirelli                 509              348
6     Goodyear                 493              368
1  Bridgestone                 244              175
7     Michelin                 217              102
3       Dunlop                 177               84
5    Firestone                 122               48
4    Englebert                  60                8
0         Avon                  32                0
2  Continental                  13               10
```

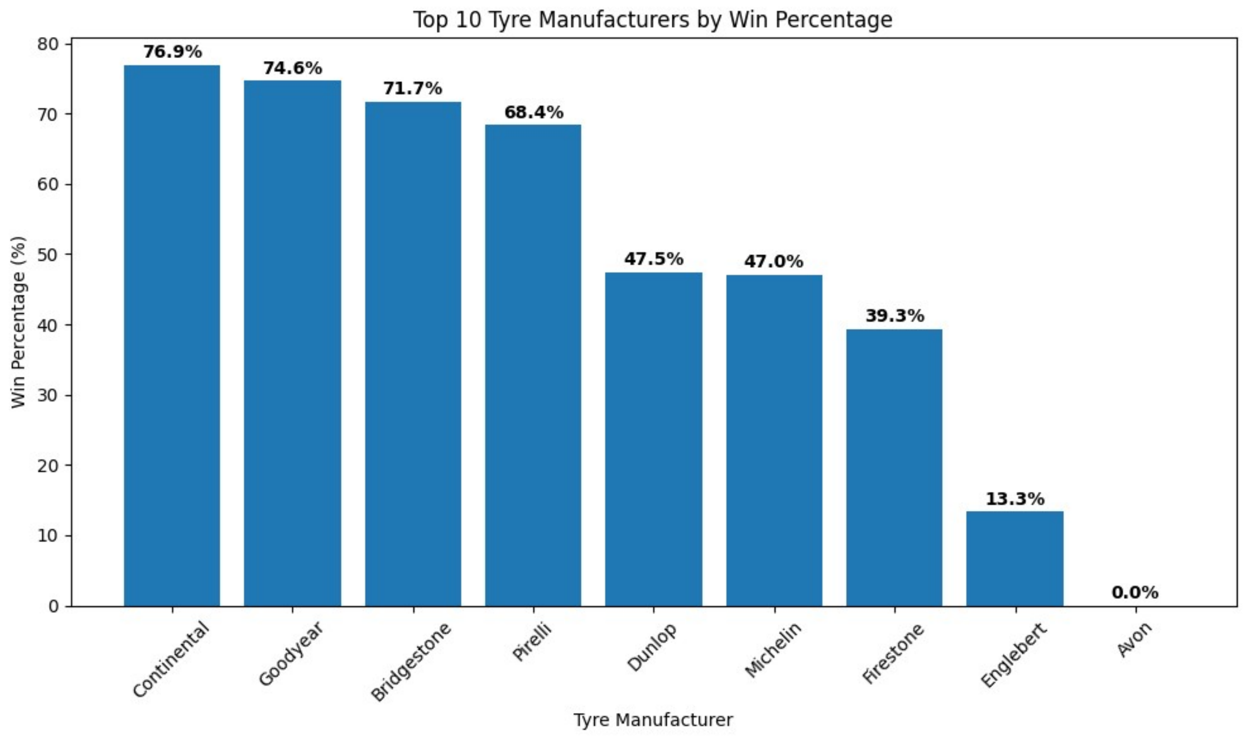
Simple Visualization 1: Total Wins Comparison

```
# Set figure size (similar to Ch9 plotting examples)
plt.figure(figsize=(10, 6))

# Get top 10 using nlargest() method from Chapter 8 slides
top_10_wins = sorted_by_wins.nlargest(10, 'total_race_wins')

# Create bar plot (similar to Ch8 plotting examples)
bars = plt.bar(top_10_wins['name'], top_10_wins['total_race_wins'])

# Set titles and labels (similar to Ch9 plot customization)
plt.title('Top 10 Tyre Manufacturers by Total Wins')
plt.xlabel('Tyre Manufacturer')
plt.ylabel('Total Wins')
plt.xticks(rotation=45)  # Rotate x-axis labels like in Ch9 examples

# Add value labels on bars (enhanced version with formatting)
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height + 5,
             f'{int(height):,}',  # Format with commas for thousands
             ha='center', va='bottom', fontweight='bold')

# Adjust layout and show (standard pattern from slides)
plt.tight_layout()
plt.show()
```



Simple Visualization 2: Win Percentage

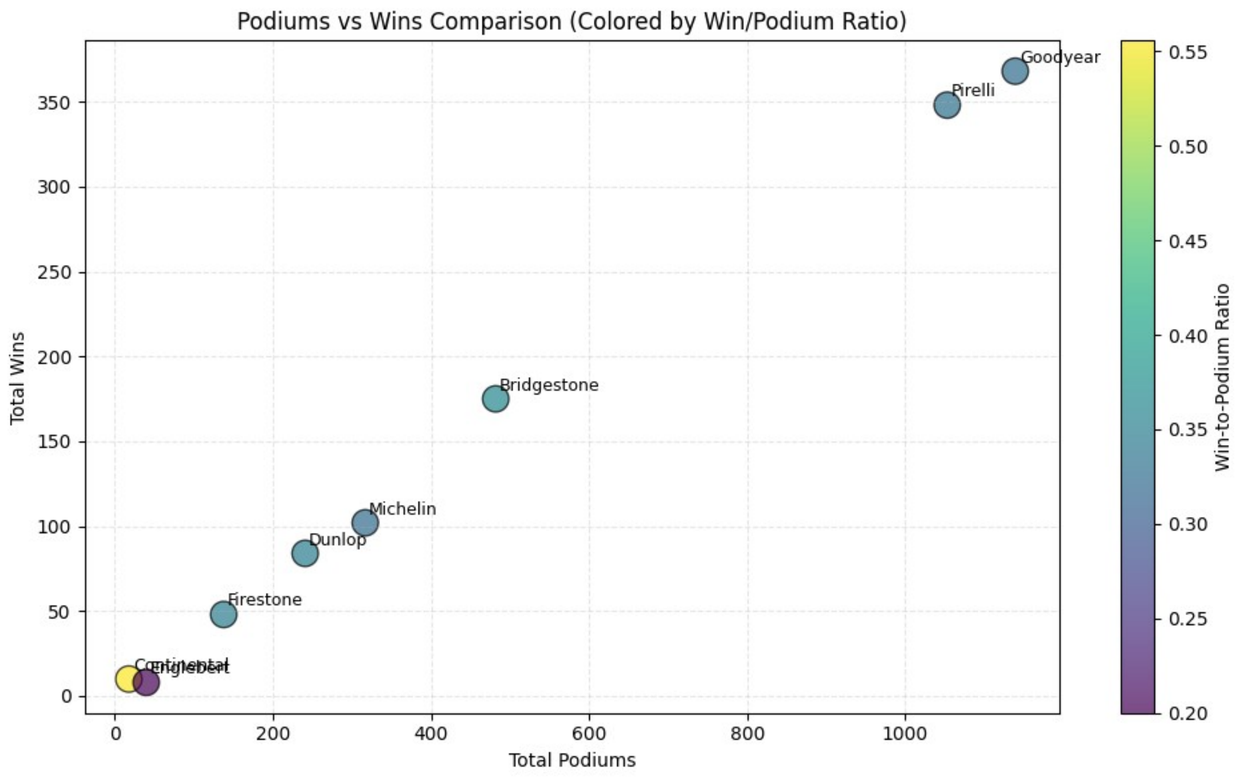
```
# Set figure size (similar to Ch9 plotting examples)
plt.figure(figsize=(10, 6))

# Get top 10 using nlargest() method from Chapter 8 slides
top_10_percentage = sorted_by_win_percentage.nlargest(10,
'win_percentage')

# Create bar plot (similar to Ch8 plotting examples)
bars = plt.bar(top_10_percentage['name'],
top_10_percentage['win_percentage'])

# Set titles and labels (similar to Ch9 plot customization)
plt.title('Top 10 Tyre Manufacturers by Win Percentage')
plt.xlabel('Tyre Manufacturer')
plt.ylabel('Win Percentage (%)')
plt.xticks(rotation=45)  # Rotate x-axis labels like in Ch9 examples

# Add value labels on bars with formatting
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height + 0.5,
             f'{height:.1f}%',  # Format as percentage with 1 decimal
             ha='center', va='bottom', fontweight='bold')

# Adjust layout and show (standard pattern from slides)
plt.tight_layout()
plt.show()
```



Simple Visualization 3: Podiums vs Wins

```
# Get top 8
top_8 = sorted_by_wins.nlargest(8, 'total_race_wins')

# Calculate win-to-podium ratio for coloring
```

```
top_8['win_podium_ratio'] = top_8['total_race_wins'] /
top_8['total_podiums']

# Create scatter plot with color gradient
plt.figure(figsize=(10, 6))
scatter = plt.scatter(top_8['total_podiums'],
top_8['total_race_wins'],
                     s=200, alpha=0.7,
                     c=top_8['win_podium_ratio'],  # Color by ratio
                     cmap='viridis',
                     edgecolors='black', linewidth=1)

# Add colorbar
plt.colorbar(scatter, label='Win-to-Podium Ratio')

# Add labels for each point
for idx, row in top_8.iterrows():
    plt.text(row['total_podiums'] + 5, row['total_race_wins'] + 5,
             row['name'], fontsize=9, fontweight='medium')

# Set titles and labels
plt.title('Podiums vs Wins Comparison (Colored by Win/Podium Ratio)')
plt.xlabel('Total Podiums')
plt.ylabel('Total Wins')
plt.grid(True, alpha=0.3, linestyle='--')

plt.tight_layout()
plt.show()
```



```python
# Calculate some summary statistics
print("=== Summary Statistics ===")
print(f"Average wins per manufacturer:
{tire_data['total_race_wins'].mean():.1f}")
print(f"Average race entries per manufacturer:
{tire_data['total_race_entries'].mean():.1f}")
print(f"Total wins across all manufacturers:
{tire_data['total_race_wins'].sum()}")
print(f"Total race entries across all manufacturers:
{tire_data['total_race_entries'].sum()}")

# Show manufacturers with perfect or near-perfect records
print("\n=== Manufacturers with High Win Rates ===")
high_win_rate = tire_data[tire_data['win_percentage'] > 50]
print(high_win_rate[['name', 'win_percentage', 'total_race_wins',
'total_race_entries']].sort_values('win_percentage', ascending=False))
```

```
=== Summary Statistics ===
Average wins per manufacturer: 127.0
Average race entries per manufacturer: 207.4
Total wins across all manufacturers: 1143
Total race entries across all manufacturers: 1867

=== Manufacturers with High Win Rates ===
          name  win_percentage  total_race_wins  total_race_entries
2  Continental       76.923077               10                  13
```

```
6     Goodyear       74.645030                  368                  493
1  Bridgestone       71.721311                  175                  244
8      Pirelli       68.369352                  348                  509
```