# -\*- coding: utf-8 -\*--

"""

Created on Wed Oct 19 13:18:51 2022

@author: andre

"""

#%% notes

#can only go back to 2018 using the API, 2014-2017 leads to error:

#Cannot load laps, telemetry, weather, and message data because the relevant API is not supported for this session.

#%% installing packages

#pip install "fastf1"

#import fastf1

#installing libraries

import numpy as np

import pandas as pd

import fastf1 as ff1

from fastf1 import plotting

plotting.setup\_mpl()

import matplotlib.pyplot as plt

from matplotlib.collections import LineCollection

from matplotlib import cm

import plotly.express as px

import plotly.graph\_objs as go

from sklearn import linear\_model

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

from patsy import dmatrices

import statsmodels.api as sm

import statsmodels.formula.api as smf

#%%

import os

#%% redirecting OS and caching data for memory

os.getcwd()

#directing to cache folder for data storage

ff1.Cache.enable\_cache("C:\\Users\\andre\\OneDrive\\Desktop\\Thesis\\cache")

#%% getting event schedule by year

schedule18 = ff1.get\_event\_schedule(2018)

schedule19 = ff1.get\_event\_schedule(2019)

schedule20 = ff1.get\_event\_schedule(2020)

schedule21 = ff1.get\_event\_schedule(2021)

schedule22 = ff1.get\_event\_schedule(2022)

#%% loading 2018 data

australia\_race\_18 = ff1.get\_session(2018, 1, 'R')

australia\_race\_18.load()

australia\_race\_18\_laps = australia\_race\_18.laps

australia\_race\_18\_messages = australia\_race\_18.race\_control\_messages

australia\_race\_18\_weather = australia\_race\_18.weather\_data

australia\_race\_18\_results = australia\_race\_18.results

australia\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

australia\_race\_18\_final = pd.merge(australia\_race\_18\_laps, australia\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

australia\_race\_18\_final['TotalTime'] = australia\_race\_18\_final['Time'] - australia\_race\_18.session\_start\_time

australia\_race\_18\_final = australia\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

bahrain\_race\_18 = ff1.get\_session(2018, 2, 'R')

bahrain\_race\_18.load()

bahrain\_race\_18\_laps = bahrain\_race\_18.laps

bahrain\_race\_18\_messages = bahrain\_race\_18.race\_control\_messages

bahrain\_race\_18\_weather = bahrain\_race\_18.weather\_data

bahrain\_race\_18\_results = bahrain\_race\_18.results

bahrain\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

bahrain\_race\_18\_final = pd.merge(bahrain\_race\_18\_laps, bahrain\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

bahrain\_race\_18\_final['TotalTime'] = bahrain\_race\_18\_final['Time'] - bahrain\_race\_18.session\_start\_time

bahrain\_race\_18\_final = bahrain\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

china\_race\_18 = ff1.get\_session(2018, 3, 'R')

china\_race\_18.load()

china\_race\_18\_laps = china\_race\_18.laps

china\_race\_18\_messages = china\_race\_18.race\_control\_messages

china\_race\_18\_weather = china\_race\_18.weather\_data

china\_race\_18\_results = china\_race\_18.results

china\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

china\_race\_18\_final = pd.merge(china\_race\_18\_laps, china\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

china\_race\_18\_final['TotalTime'] = china\_race\_18\_final['Time'] - china\_race\_18.session\_start\_time

china\_race\_18\_final = china\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

azerbaijan\_race\_18 = ff1.get\_session(2018, 4, 'R')

azerbaijan\_race\_18.load()

azerbaijan\_race\_18\_laps = azerbaijan\_race\_18.laps

azerbaijan\_race\_18\_messages = azerbaijan\_race\_18.race\_control\_messages

azerbaijan\_race\_18\_weather = azerbaijan\_race\_18.weather\_data

azerbaijan\_race\_18\_results = azerbaijan\_race\_18.results

azerbaijan\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

azerbaijan\_race\_18\_final = pd.merge(azerbaijan\_race\_18\_laps, azerbaijan\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

azerbaijan\_race\_18\_final['TotalTime'] = azerbaijan\_race\_18\_final['Time'] - azerbaijan\_race\_18.session\_start\_time

azerbaijan\_race\_18\_final = azerbaijan\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

spain\_race\_18 = ff1.get\_session(2018, 5, 'R')

spain\_race\_18.load()

spain\_race\_18\_laps = spain\_race\_18.laps

spain\_race\_18\_messages = spain\_race\_18.race\_control\_messages

spain\_race\_18\_weather = spain\_race\_18.weather\_data

spain\_race\_18\_results = spain\_race\_18.results

spain\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

spain\_race\_18\_final = pd.merge(spain\_race\_18\_laps, spain\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

spain\_race\_18\_final['TotalTime'] = spain\_race\_18\_final['Time'] - spain\_race\_18.session\_start\_time

spain\_race\_18\_final = spain\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

monaco\_race\_18 = ff1.get\_session(2018, 6, 'R')

monaco\_race\_18.load()

monaco\_race\_18\_laps = monaco\_race\_18.laps

monaco\_race\_18\_messages = monaco\_race\_18.race\_control\_messages

monaco\_race\_18\_weather = monaco\_race\_18.weather\_data

monaco\_race\_18\_results = monaco\_race\_18.results

monaco\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

monaco\_race\_18\_final = pd.merge(monaco\_race\_18\_laps, monaco\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

monaco\_race\_18\_final['TotalTime'] = monaco\_race\_18\_final['Time'] - monaco\_race\_18.session\_start\_time

monaco\_race\_18\_final = monaco\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

canada\_race\_18 = ff1.get\_session(2018, 7, 'R')

canada\_race\_18.load()

canada\_race\_18\_laps = canada\_race\_18.laps

canada\_race\_18\_messages = canada\_race\_18.race\_control\_messages

canada\_race\_18\_weather = canada\_race\_18.weather\_data

canada\_race\_18\_results = canada\_race\_18.results

canada\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

canada\_race\_18\_final = pd.merge(canada\_race\_18\_laps, canada\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

canada\_race\_18\_final['TotalTime'] = canada\_race\_18\_final['Time'] - canada\_race\_18.session\_start\_time

canada\_race\_18\_final = canada\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

france\_race\_18 = ff1.get\_session(2018, 8, 'R')

france\_race\_18.load()

france\_race\_18\_laps = france\_race\_18.laps

france\_race\_18\_messages = france\_race\_18.race\_control\_messages

france\_race\_18\_weather = france\_race\_18.weather\_data

france\_race\_18\_results = france\_race\_18.results

france\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

france\_race\_18\_final = pd.merge(france\_race\_18\_laps, france\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

france\_race\_18\_final['TotalTime'] = france\_race\_18\_final['Time'] - france\_race\_18.session\_start\_time

france\_race\_18\_final = france\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

austria\_race\_18 = ff1.get\_session(2018, 9, 'R')

austria\_race\_18.load()

austria\_race\_18\_laps = austria\_race\_18.laps

austria\_race\_18\_messages = austria\_race\_18.race\_control\_messages

austria\_race\_18\_weather = austria\_race\_18.weather\_data

austria\_race\_18\_results = austria\_race\_18.results

austria\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_18\_final = pd.merge(austria\_race\_18\_laps, austria\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_18\_final['TotalTime'] = austria\_race\_18\_final['Time'] - austria\_race\_18.session\_start\_time

austria\_race\_18\_final = austria\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

great\_britain\_race\_18 = ff1.get\_session(2018, 10, 'R')

great\_britain\_race\_18.load()

great\_britain\_race\_18\_laps = great\_britain\_race\_18.laps

great\_britain\_race\_18\_messages = great\_britain\_race\_18.race\_control\_messages

great\_britain\_race\_18\_weather = great\_britain\_race\_18.weather\_data

great\_britain\_race\_18\_results = great\_britain\_race\_18.results

great\_britain\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

great\_britain\_race\_18\_final = pd.merge(great\_britain\_race\_18\_laps, great\_britain\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

great\_britain\_race\_18\_final['TotalTime'] = great\_britain\_race\_18\_final['Time'] - great\_britain\_race\_18.session\_start\_time

great\_britain\_race\_18\_final = great\_britain\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

germany\_race\_18 = ff1.get\_session(2018, 11, 'R')

germany\_race\_18.load()

germany\_race\_18\_laps = germany\_race\_18.laps

germany\_race\_18\_messages = germany\_race\_18.race\_control\_messages

germany\_race\_18\_weather = germany\_race\_18.weather\_data

germany\_race\_18\_results = germany\_race\_18.results

germany\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

germany\_race\_18\_final = pd.merge(germany\_race\_18\_laps, germany\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

germany\_race\_18\_final['TotalTime'] = germany\_race\_18\_final['Time'] - germany\_race\_18.session\_start\_time

germany\_race\_18\_final = germany\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

hungary\_race\_18 = ff1.get\_session(2018, 12, 'R')

hungary\_race\_18.load()

hungary\_race\_18\_laps = hungary\_race\_18.laps

hungary\_race\_18\_messages = hungary\_race\_18.race\_control\_messages

hungary\_race\_18\_weather = hungary\_race\_18.weather\_data

hungary\_race\_18\_results = hungary\_race\_18.results

hungary\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

hungary\_race\_18\_final = pd.merge(hungary\_race\_18\_laps, hungary\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

hungary\_race\_18\_final['TotalTime'] = hungary\_race\_18\_final['Time'] - hungary\_race\_18.session\_start\_time

hungary\_race\_18\_final = hungary\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

belgium\_race\_18 = ff1.get\_session(2018, 13, 'R')

belgium\_race\_18.load()

belgium\_race\_18\_laps = belgium\_race\_18.laps

belgium\_race\_18\_messages = belgium\_race\_18.race\_control\_messages

belgium\_race\_18\_weather = belgium\_race\_18.weather\_data

belgium\_race\_18\_results = belgium\_race\_18.results

belgium\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

belgium\_race\_18\_final = pd.merge(belgium\_race\_18\_laps, belgium\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

belgium\_race\_18\_final['TotalTime'] = belgium\_race\_18\_final['Time'] - belgium\_race\_18.session\_start\_time

belgium\_race\_18\_final = belgium\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_18 = ff1.get\_session(2018, 14, 'R')

italy\_race\_18.load()

italy\_race\_18\_laps = italy\_race\_18.laps

italy\_race\_18\_messages = italy\_race\_18.race\_control\_messages

italy\_race\_18\_weather = italy\_race\_18.weather\_data

italy\_race\_18\_results = italy\_race\_18.results

italy\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_18\_final = pd.merge(italy\_race\_18\_laps, italy\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_18\_final['TotalTime'] = italy\_race\_18\_final['Time'] - italy\_race\_18.session\_start\_time

italy\_race\_18\_final = italy\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

singapore\_race\_18 = ff1.get\_session(2018, 15, 'R')

singapore\_race\_18.load()

singapore\_race\_18\_laps = singapore\_race\_18.laps

singapore\_race\_18\_messages = singapore\_race\_18.race\_control\_messages

singapore\_race\_18\_weather = singapore\_race\_18.weather\_data

singapore\_race\_18\_results = singapore\_race\_18.results

singapore\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

singapore\_race\_18\_final = pd.merge(singapore\_race\_18\_laps, singapore\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

singapore\_race\_18\_final['TotalTime'] = singapore\_race\_18\_final['Time'] - singapore\_race\_18.session\_start\_time

singapore\_race\_18\_final = singapore\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

russia\_race\_18 = ff1.get\_session(2018, 16, 'R')

russia\_race\_18.load()

russia\_race\_18\_laps = russia\_race\_18.laps

russia\_race\_18\_messages = russia\_race\_18.race\_control\_messages

russia\_race\_18\_weather = russia\_race\_18.weather\_data

russia\_race\_18\_results = russia\_race\_18.results

russia\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

russia\_race\_18\_final = pd.merge(russia\_race\_18\_laps, russia\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

russia\_race\_18\_final['TotalTime'] = russia\_race\_18\_final['Time'] - russia\_race\_18.session\_start\_time

russia\_race\_18\_final = russia\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

japan\_race\_18 = ff1.get\_session(2018, 17, 'R')

japan\_race\_18.load()

japan\_race\_18\_laps = japan\_race\_18.laps

japan\_race\_18\_messages = japan\_race\_18.race\_control\_messages

japan\_race\_18\_weather = japan\_race\_18.weather\_data

japan\_race\_18\_results = japan\_race\_18.results

japan\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

japan\_race\_18\_final = pd.merge(japan\_race\_18\_laps, japan\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

japan\_race\_18\_final['TotalTime'] = japan\_race\_18\_final['Time'] - japan\_race\_18.session\_start\_time

japan\_race\_18\_final = japan\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

united\_states\_race\_18 = ff1.get\_session(2018, 18, 'R')

united\_states\_race\_18.load()

united\_states\_race\_18\_laps = united\_states\_race\_18.laps

united\_states\_race\_18\_messages = united\_states\_race\_18.race\_control\_messages

united\_states\_race\_18\_weather = united\_states\_race\_18.weather\_data

united\_states\_race\_18\_results = united\_states\_race\_18.results

united\_states\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

united\_states\_race\_18\_final = pd.merge(united\_states\_race\_18\_laps, united\_states\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

united\_states\_race\_18\_final['TotalTime'] = united\_states\_race\_18\_final['Time'] - united\_states\_race\_18.session\_start\_time

united\_states\_race\_18\_final = united\_states\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

mexico\_race\_18 = ff1.get\_session(2018, 19, 'R')

mexico\_race\_18.load()

mexico\_race\_18\_laps = mexico\_race\_18.laps

mexico\_race\_18\_messages = mexico\_race\_18.race\_control\_messages

mexico\_race\_18\_weather = mexico\_race\_18.weather\_data

mexico\_race\_18\_results = mexico\_race\_18.results

mexico\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

mexico\_race\_18\_final = pd.merge(mexico\_race\_18\_laps, mexico\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

mexico\_race\_18\_final['TotalTime'] = mexico\_race\_18\_final['Time'] - mexico\_race\_18.session\_start\_time

mexico\_race\_18\_final = mexico\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

brazil\_race\_18 = ff1.get\_session(2018, 20, 'R')

brazil\_race\_18.load()

brazil\_race\_18\_laps = brazil\_race\_18.laps

brazil\_race\_18\_messages = brazil\_race\_18.race\_control\_messages

brazil\_race\_18\_weather = brazil\_race\_18.weather\_data

brazil\_race\_18\_results = brazil\_race\_18.results

brazil\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

brazil\_race\_18\_final = pd.merge(brazil\_race\_18\_laps, brazil\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

brazil\_race\_18\_final['TotalTime'] = brazil\_race\_18\_final['Time'] - brazil\_race\_18.session\_start\_time

brazil\_race\_18\_final = brazil\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

united\_arab\_emirates\_race\_18 = ff1.get\_session(2018, 21, 'R')

united\_arab\_emirates\_race\_18.load()

united\_arab\_emirates\_race\_18\_laps = united\_arab\_emirates\_race\_18.laps

united\_arab\_emirates\_race\_18\_messages = united\_arab\_emirates\_race\_18.race\_control\_messages

united\_arab\_emirates\_race\_18\_weather = united\_arab\_emirates\_race\_18.weather\_data

united\_arab\_emirates\_race\_18\_results = united\_arab\_emirates\_race\_18.results

united\_arab\_emirates\_race\_18\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

united\_arab\_emirates\_race\_18\_final = pd.merge(united\_arab\_emirates\_race\_18\_laps, united\_arab\_emirates\_race\_18\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

united\_arab\_emirates\_race\_18\_final['TotalTime'] = united\_arab\_emirates\_race\_18\_final['Time'] - united\_arab\_emirates\_race\_18.session\_start\_time

united\_arab\_emirates\_race\_18\_final = united\_arab\_emirates\_race\_18\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

merge2018\_laps = pd.concat([

australia\_race\_18\_laps,

bahrain\_race\_18\_laps,

china\_race\_18\_laps,

azerbaijan\_race\_18\_laps,

spain\_race\_18\_laps,

monaco\_race\_18\_laps,

canada\_race\_18\_laps,

france\_race\_18\_laps,

austria\_race\_18\_laps,

great\_britain\_race\_18\_laps,

germany\_race\_18\_laps,

hungary\_race\_18\_laps,

belgium\_race\_18\_laps,

italy\_race\_18\_laps,

singapore\_race\_18\_laps,

russia\_race\_18\_laps,

japan\_race\_18\_laps,

united\_states\_race\_18\_laps,

mexico\_race\_18\_laps,

brazil\_race\_18\_laps,

united\_arab\_emirates\_race\_18\_laps],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'UnitedStates',

'Mexico',

'Brazil',

'UnitedArabEmirates'],

names=['RACE'])

merge2018\_messages = pd.concat([

australia\_race\_18\_messages,

bahrain\_race\_18\_messages,

china\_race\_18\_messages,

azerbaijan\_race\_18\_messages,

spain\_race\_18\_messages,

monaco\_race\_18\_messages,

canada\_race\_18\_messages,

france\_race\_18\_messages,

austria\_race\_18\_messages,

great\_britain\_race\_18\_messages,

germany\_race\_18\_messages,

hungary\_race\_18\_messages,

belgium\_race\_18\_messages,

italy\_race\_18\_messages,

singapore\_race\_18\_messages,

russia\_race\_18\_messages,

japan\_race\_18\_messages,

united\_states\_race\_18\_messages,

mexico\_race\_18\_messages,

brazil\_race\_18\_messages,

united\_arab\_emirates\_race\_18\_messages],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'UnitedStates',

'Mexico',

'Brazil',

'UnitedArabEmirates'],

names=['RACE'])

merge2018\_weather = pd.concat([

australia\_race\_18\_weather,

bahrain\_race\_18\_weather,

china\_race\_18\_weather,

azerbaijan\_race\_18\_weather,

spain\_race\_18\_weather,

monaco\_race\_18\_weather,

canada\_race\_18\_weather,

france\_race\_18\_weather,

austria\_race\_18\_weather,

great\_britain\_race\_18\_weather,

germany\_race\_18\_weather,

hungary\_race\_18\_weather,

belgium\_race\_18\_weather,

italy\_race\_18\_weather,

singapore\_race\_18\_weather,

russia\_race\_18\_weather,

japan\_race\_18\_weather,

united\_states\_race\_18\_weather,

mexico\_race\_18\_weather,

brazil\_race\_18\_weather,

united\_arab\_emirates\_race\_18\_weather],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'UnitedStates',

'Mexico',

'Brazil',

'UnitedArabEmirates'],

names=['RACE'])

merge2018\_results = pd.concat([

australia\_race\_18\_results,

bahrain\_race\_18\_results,

china\_race\_18\_results,

azerbaijan\_race\_18\_results,

spain\_race\_18\_results,

monaco\_race\_18\_results,

canada\_race\_18\_results,

france\_race\_18\_results,

austria\_race\_18\_results,

great\_britain\_race\_18\_results,

germany\_race\_18\_results,

hungary\_race\_18\_results,

belgium\_race\_18\_results,

italy\_race\_18\_results,

singapore\_race\_18\_results,

russia\_race\_18\_results,

japan\_race\_18\_results,

united\_states\_race\_18\_results,

mexico\_race\_18\_results,

brazil\_race\_18\_results,

united\_arab\_emirates\_race\_18\_results],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'UnitedStates',

'Mexico',

'Brazil',

'UnitedArabEmirates'],

names=['RACE'])

merge2018\_final = pd.concat([

australia\_race\_18\_final,

bahrain\_race\_18\_final,

china\_race\_18\_final,

azerbaijan\_race\_18\_final,

spain\_race\_18\_final,

monaco\_race\_18\_final,

canada\_race\_18\_final,

france\_race\_18\_final,

austria\_race\_18\_final,

great\_britain\_race\_18\_final,

germany\_race\_18\_final,

hungary\_race\_18\_final,

belgium\_race\_18\_final,

italy\_race\_18\_final,

singapore\_race\_18\_final,

russia\_race\_18\_final,

japan\_race\_18\_final,

united\_states\_race\_18\_final,

mexico\_race\_18\_final,

brazil\_race\_18\_final,

united\_arab\_emirates\_race\_18\_final],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'UnitedStates',

'Mexico',

'Brazil',

'UnitedArabEmirates'],

names=['RACE'])

merge2018\_laps["Year"] = 2018

merge2018\_messages["Year"] = 2018

merge2018\_weather["Year"] = 2018

merge2018\_results["Year"] = 2018

merge2018\_final["Year"] = 2018

#%% loading 2019 data

australia\_race\_19 = ff1.get\_session(2019, 1, 'R')

australia\_race\_19.load()

australia\_race\_19\_laps = australia\_race\_19.laps

australia\_race\_19\_messages = australia\_race\_19.race\_control\_messages

australia\_race\_19\_weather = australia\_race\_19.weather\_data

australia\_race\_19\_results = australia\_race\_19.results

australia\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

australia\_race\_19\_final = pd.merge(australia\_race\_19\_laps, australia\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

australia\_race\_19\_final['TotalTime'] = australia\_race\_19\_final['Time'] - australia\_race\_19.session\_start\_time

australia\_race\_19\_final = australia\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

bahrain\_race\_19 = ff1.get\_session(2019, 2, 'R')

bahrain\_race\_19.load()

bahrain\_race\_19\_laps = bahrain\_race\_19.laps

bahrain\_race\_19\_messages = bahrain\_race\_19.race\_control\_messages

bahrain\_race\_19\_weather = bahrain\_race\_19.weather\_data

bahrain\_race\_19\_results = bahrain\_race\_19.results

bahrain\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

bahrain\_race\_19\_final = pd.merge(bahrain\_race\_19\_laps, bahrain\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

bahrain\_race\_19\_final['TotalTime'] = bahrain\_race\_19\_final['Time'] - bahrain\_race\_19.session\_start\_time

bahrain\_race\_19\_final = bahrain\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

china\_race\_19 = ff1.get\_session(2019, 3, 'R')

china\_race\_19.load()

china\_race\_19\_laps = china\_race\_19.laps

china\_race\_19\_messages= china\_race\_19.race\_control\_messages

china\_race\_19\_weather = china\_race\_19.weather\_data

china\_race\_19\_results = china\_race\_19.results

china\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

china\_race\_19\_final = pd.merge(china\_race\_19\_laps, china\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

china\_race\_19\_final['TotalTime'] = china\_race\_19\_final['Time'] - china\_race\_19.session\_start\_time

china\_race\_19\_final = china\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

azerbaijan\_race\_19 = ff1.get\_session(2019, 4, 'R')

azerbaijan\_race\_19.load()

azerbaijan\_race\_19\_laps = azerbaijan\_race\_19.laps

azerbaijan\_race\_19\_messages = azerbaijan\_race\_19.race\_control\_messages

azerbaijan\_race\_19\_weather = azerbaijan\_race\_19.weather\_data

azerbaijan\_race\_19\_results = azerbaijan\_race\_19.results

azerbaijan\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

azerbaijan\_race\_19\_final = pd.merge(azerbaijan\_race\_19\_laps, azerbaijan\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

azerbaijan\_race\_19\_final['TotalTime'] = azerbaijan\_race\_19\_final['Time'] - azerbaijan\_race\_19.session\_start\_time

azerbaijan\_race\_19\_final = azerbaijan\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

spain\_race\_19 = ff1.get\_session(2019, 5, 'R')

spain\_race\_19.load()

spain\_race\_19\_laps = spain\_race\_19.laps

spain\_race\_19\_messages = spain\_race\_19.race\_control\_messages

spain\_race\_19\_weather = spain\_race\_19.weather\_data

spain\_race\_19\_results = spain\_race\_19.results

spain\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

spain\_race\_19\_final = pd.merge(spain\_race\_19\_laps, spain\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

spain\_race\_19\_final['TotalTime'] = spain\_race\_19\_final['Time'] - spain\_race\_19.session\_start\_time

spain\_race\_19\_final = spain\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

monaco\_race\_19 = ff1.get\_session(2019, 6, 'R')

monaco\_race\_19.load()

monaco\_race\_19\_laps = monaco\_race\_19.laps

monaco\_race\_19\_messages = monaco\_race\_19.race\_control\_messages

monaco\_race\_19\_weather = monaco\_race\_19.weather\_data

monaco\_race\_19\_results = monaco\_race\_19.results

monaco\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

monaco\_race\_19\_final = pd.merge(monaco\_race\_19\_laps, monaco\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

monaco\_race\_19\_final['TotalTime'] = monaco\_race\_19\_final['Time'] - monaco\_race\_19.session\_start\_time

monaco\_race\_19\_final = monaco\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

canada\_race\_19 = ff1.get\_session(2019, 7, 'R')

canada\_race\_19.load()

canada\_race\_19\_laps = canada\_race\_19.laps

canada\_race\_19\_messages = canada\_race\_19.race\_control\_messages

canada\_race\_19\_weather = canada\_race\_19.weather\_data

canada\_race\_19\_results = canada\_race\_19.results

canada\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

canada\_race\_19\_final = pd.merge(canada\_race\_19\_laps, canada\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

canada\_race\_19\_final['TotalTime'] = canada\_race\_19\_final['Time'] - canada\_race\_19.session\_start\_time

canada\_race\_19\_final = canada\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

france\_race\_19 = ff1.get\_session(2019, 8, 'R')

france\_race\_19.load()

france\_race\_19\_laps = france\_race\_19.laps

france\_race\_19\_messages = france\_race\_19.race\_control\_messages

france\_race\_19\_weather = france\_race\_19.weather\_data

france\_race\_19\_results = france\_race\_19.results

france\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

france\_race\_19\_final = pd.merge(france\_race\_19\_laps, france\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

france\_race\_19\_final['TotalTime'] = france\_race\_19\_final['Time'] - france\_race\_19.session\_start\_time

france\_race\_19\_final = france\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

austria\_race\_19 = ff1.get\_session(2019, 9, 'R')

austria\_race\_19.load()

austria\_race\_19\_laps = austria\_race\_19.laps

austria\_race\_19\_messages = austria\_race\_19.race\_control\_messages

austria\_race\_19\_weather = austria\_race\_19.weather\_data

austria\_race\_19\_results = austria\_race\_19.results

austria\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_19\_final = pd.merge(austria\_race\_19\_laps, austria\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_19\_final['TotalTime'] = austria\_race\_19\_final['Time'] - austria\_race\_19.session\_start\_time

austria\_race\_19\_final = austria\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

great\_britain\_race\_19 = ff1.get\_session(2019, 10, 'R')

great\_britain\_race\_19.load()

great\_britain\_race\_19\_laps = great\_britain\_race\_19.laps

great\_britain\_race\_19\_messages = great\_britain\_race\_19.race\_control\_messages

great\_britain\_race\_19\_weather = great\_britain\_race\_19.weather\_data

great\_britain\_race\_19\_results = great\_britain\_race\_19.results

great\_britain\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

great\_britain\_race\_19\_final = pd.merge(great\_britain\_race\_19\_laps, great\_britain\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

great\_britain\_race\_19\_final['TotalTime'] = great\_britain\_race\_19\_final['Time'] - great\_britain\_race\_19.session\_start\_time

great\_britain\_race\_19\_final = great\_britain\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

germany\_race\_19 = ff1.get\_session(2019, 11, 'R')

germany\_race\_19.load()

germany\_race\_19\_laps = germany\_race\_19.laps

germany\_race\_19\_messages = germany\_race\_19.race\_control\_messages

germany\_race\_19\_weather = germany\_race\_19.weather\_data

germany\_race\_19\_results = germany\_race\_19.results

germany\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

germany\_race\_19\_final = pd.merge(germany\_race\_19\_laps, germany\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

germany\_race\_19\_final['TotalTime'] = germany\_race\_19\_final['Time'] - germany\_race\_19.session\_start\_time

germany\_race\_19\_final = germany\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

hungary\_race\_19 = ff1.get\_session(2019, 12, 'R')

hungary\_race\_19.load()

hungary\_race\_19\_laps = hungary\_race\_19.laps

hungary\_race\_19\_messages = hungary\_race\_19.race\_control\_messages

hungary\_race\_19\_weather = hungary\_race\_19.weather\_data

hungary\_race\_19\_results = hungary\_race\_19.results

hungary\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

hungary\_race\_19\_final = pd.merge(hungary\_race\_19\_laps, hungary\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

hungary\_race\_19\_final['TotalTime'] = hungary\_race\_19\_final['Time'] - hungary\_race\_19.session\_start\_time

hungary\_race\_19\_final = hungary\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

belgium\_race\_19 = ff1.get\_session(2019, 13, 'R')

belgium\_race\_19.load()

belgium\_race\_19\_laps = belgium\_race\_19.laps

belgium\_race\_19\_messages = belgium\_race\_19.race\_control\_messages

belgium\_race\_19\_weather = belgium\_race\_19.weather\_data

belgium\_race\_19\_results = belgium\_race\_19.results

belgium\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

belgium\_race\_19\_final = pd.merge(belgium\_race\_19\_laps, belgium\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

belgium\_race\_19\_final['TotalTime'] = belgium\_race\_19\_final['Time'] - belgium\_race\_19.session\_start\_time

belgium\_race\_19\_final = belgium\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_19 = ff1.get\_session(2019, 14, 'R')

italy\_race\_19.load()

italy\_race\_19\_laps = italy\_race\_19.laps

italy\_race\_19\_messages = italy\_race\_19.race\_control\_messages

italy\_race\_19\_weather = italy\_race\_19.weather\_data

italy\_race\_19\_results = italy\_race\_19.results

italy\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_19\_final = pd.merge(italy\_race\_19\_laps, italy\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_19\_final['TotalTime'] = italy\_race\_19\_final['Time'] - italy\_race\_19.session\_start\_time

italy\_race\_19\_final = italy\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

singapore\_race\_19 = ff1.get\_session(2019, 15, 'R')

singapore\_race\_19.load()

singapore\_race\_19\_laps = singapore\_race\_19.laps

singapore\_race\_19\_messages = singapore\_race\_19.race\_control\_messages

singapore\_race\_19\_weather = singapore\_race\_19.weather\_data

singapore\_race\_19\_results = singapore\_race\_19.results

singapore\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

singapore\_race\_19\_final = pd.merge(singapore\_race\_19\_laps, singapore\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

singapore\_race\_19\_final['TotalTime'] = singapore\_race\_19\_final['Time'] - singapore\_race\_19.session\_start\_time

singapore\_race\_19\_final = singapore\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

russia\_race\_19 = ff1.get\_session(2018, 16, 'R')

russia\_race\_19.load()

russia\_race\_19\_laps = russia\_race\_19.laps

russia\_race\_19\_messages = russia\_race\_19.race\_control\_messages

russia\_race\_19\_weather = russia\_race\_19.weather\_data

russia\_race\_19\_results = russia\_race\_19.results

russia\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

russia\_race\_19\_final = pd.merge(russia\_race\_19\_laps, russia\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

russia\_race\_19\_final['TotalTime'] = russia\_race\_19\_final['Time'] - russia\_race\_19.session\_start\_time

russia\_race\_19\_final = russia\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

japan\_race\_19 = ff1.get\_session(2019, 17, 'R')

japan\_race\_19.load()

japan\_race\_19\_laps = japan\_race\_19.laps

japan\_race\_19\_messages = japan\_race\_19.race\_control\_messages

japan\_race\_19\_weather = japan\_race\_19.weather\_data

japan\_race\_19\_results = japan\_race\_19.results

japan\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

japan\_race\_19\_final = pd.merge(japan\_race\_19\_laps, japan\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

japan\_race\_19\_final['TotalTime'] = japan\_race\_19\_final['Time'] - japan\_race\_19.session\_start\_time

japan\_race\_19\_final = japan\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

mexico\_race\_19 = ff1.get\_session(2019, 18, 'R')

mexico\_race\_19.load()

mexico\_race\_19\_laps = mexico\_race\_19.laps

mexico\_race\_19\_messages = mexico\_race\_19.race\_control\_messages

mexico\_race\_19\_weather = mexico\_race\_19.weather\_data

mexico\_race\_19\_results = mexico\_race\_19.results

mexico\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

mexico\_race\_19\_final = pd.merge(mexico\_race\_19\_laps, mexico\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

mexico\_race\_19\_final['TotalTime'] = mexico\_race\_19\_final['Time'] - mexico\_race\_19.session\_start\_time

mexico\_race\_19\_final = mexico\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

united\_states\_race\_19 = ff1.get\_session(2019, 19, 'R')

united\_states\_race\_19.load()

united\_states\_race\_19\_laps = united\_states\_race\_19.laps

united\_states\_race\_19\_messages = united\_states\_race\_19.race\_control\_messages

united\_states\_race\_19\_weather = united\_states\_race\_19.weather\_data

united\_states\_race\_19\_results = united\_states\_race\_19.results

united\_states\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

united\_states\_race\_19\_final = pd.merge(united\_states\_race\_19\_laps, united\_states\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

united\_states\_race\_19\_final['TotalTime'] = united\_states\_race\_19\_final['Time'] - united\_states\_race\_19.session\_start\_time

united\_states\_race\_19\_final = united\_states\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

brazil\_race\_19 = ff1.get\_session(2019, 20, 'R')

brazil\_race\_19.load()

brazil\_race\_19\_laps = brazil\_race\_19.laps

brazil\_race\_19\_messages = brazil\_race\_19.race\_control\_messages

brazil\_race\_19\_weather = brazil\_race\_19.weather\_data

brazil\_race\_19\_results = brazil\_race\_19.results

brazil\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

brazil\_race\_19\_final = pd.merge(brazil\_race\_19\_laps, brazil\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

brazil\_race\_19\_final['TotalTime'] = brazil\_race\_19\_final['Time'] - brazil\_race\_19.session\_start\_time

brazil\_race\_19\_final = brazil\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

abu\_dhabi\_race\_19 = ff1.get\_session(2019, 21, 'R')

abu\_dhabi\_race\_19.load()

abu\_dhabi\_race\_19\_laps = abu\_dhabi\_race\_19.laps

abu\_dhabi\_race\_19\_messages = abu\_dhabi\_race\_19.race\_control\_messages

abu\_dhabi\_race\_19\_weather = abu\_dhabi\_race\_19.weather\_data

abu\_dhabi\_race\_19\_results = abu\_dhabi\_race\_19.results

abu\_dhabi\_race\_19\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

abu\_dhabi\_race\_19\_final = pd.merge(abu\_dhabi\_race\_19\_laps, abu\_dhabi\_race\_19\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

abu\_dhabi\_race\_19\_final['TotalTime'] = abu\_dhabi\_race\_19\_final['Time'] - abu\_dhabi\_race\_19.session\_start\_time

abu\_dhabi\_race\_19\_final = abu\_dhabi\_race\_19\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

merge2019\_laps = pd.concat([

australia\_race\_19\_laps,

bahrain\_race\_19\_laps,

china\_race\_19\_laps,

azerbaijan\_race\_19\_laps,

spain\_race\_19\_laps,

monaco\_race\_19\_laps,

canada\_race\_19\_laps,

france\_race\_19\_laps,

austria\_race\_19\_laps,

great\_britain\_race\_19\_laps,

germany\_race\_19\_laps,

hungary\_race\_19\_laps,

belgium\_race\_19\_laps,

italy\_race\_19\_laps,

singapore\_race\_19\_laps,

russia\_race\_19\_laps,

japan\_race\_19\_laps,

mexico\_race\_19\_laps,

united\_states\_race\_19\_laps,

brazil\_race\_19\_laps,

abu\_dhabi\_race\_19\_laps],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'Mexico',

'UnitedStates',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2019\_messages = pd.concat([

australia\_race\_19\_messages,

bahrain\_race\_19\_messages,

china\_race\_19\_messages,

azerbaijan\_race\_19\_messages,

spain\_race\_19\_messages,

monaco\_race\_19\_messages,

canada\_race\_19\_messages,

france\_race\_19\_messages,

austria\_race\_19\_messages,

great\_britain\_race\_19\_messages,

germany\_race\_19\_messages,

hungary\_race\_19\_messages,

belgium\_race\_19\_messages,

italy\_race\_19\_messages,

singapore\_race\_19\_messages,

russia\_race\_19\_messages,

japan\_race\_19\_messages,

mexico\_race\_19\_messages,

united\_states\_race\_19\_messages,

brazil\_race\_19\_messages,

abu\_dhabi\_race\_19\_messages],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'Mexico',

'UnitedStates',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2019\_weather = pd.concat([

australia\_race\_19\_weather,

bahrain\_race\_19\_weather,

china\_race\_19\_weather,

azerbaijan\_race\_19\_weather,

spain\_race\_19\_weather,

monaco\_race\_19\_weather,

canada\_race\_19\_weather,

france\_race\_19\_weather,

austria\_race\_19\_weather,

great\_britain\_race\_19\_weather,

germany\_race\_19\_weather,

hungary\_race\_19\_weather,

belgium\_race\_19\_weather,

italy\_race\_19\_weather,

singapore\_race\_19\_weather,

russia\_race\_19\_weather,

japan\_race\_19\_weather,

mexico\_race\_19\_weather,

united\_states\_race\_19\_weather,

brazil\_race\_19\_weather,

abu\_dhabi\_race\_19\_weather],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'Mexico',

'UnitedStates',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2019\_results = pd.concat([

australia\_race\_19\_results ,

bahrain\_race\_19\_results ,

china\_race\_19\_results ,

azerbaijan\_race\_19\_results ,

spain\_race\_19\_results ,

monaco\_race\_19\_results ,

canada\_race\_19\_results ,

france\_race\_19\_results ,

austria\_race\_19\_results ,

great\_britain\_race\_19\_results ,

germany\_race\_19\_results ,

hungary\_race\_19\_results ,

belgium\_race\_19\_results ,

italy\_race\_19\_results ,

singapore\_race\_19\_results,

russia\_race\_19\_results ,

japan\_race\_19\_results ,

mexico\_race\_19\_results ,

united\_states\_race\_19\_results ,

brazil\_race\_19\_results ,

abu\_dhabi\_race\_19\_results],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'Mexico',

'UnitedStates',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2019\_final = pd.concat([

australia\_race\_19\_final ,

bahrain\_race\_19\_final ,

china\_race\_19\_final ,

azerbaijan\_race\_19\_final ,

spain\_race\_19\_final ,

monaco\_race\_19\_final ,

canada\_race\_19\_final ,

france\_race\_19\_final ,

austria\_race\_19\_final ,

great\_britain\_race\_19\_final ,

germany\_race\_19\_final ,

hungary\_race\_19\_final ,

belgium\_race\_19\_final ,

italy\_race\_19\_final ,

singapore\_race\_19\_final,

russia\_race\_19\_final ,

japan\_race\_19\_final ,

mexico\_race\_19\_final ,

united\_states\_race\_19\_final ,

brazil\_race\_19\_final ,

abu\_dhabi\_race\_19\_final],

keys = ['Australia',

'Bahrain',

'China',

'Azerbaijan',

'Spain',

'Monaco',

'Canada',

'France',

'Austria',

'GreatBritain',

'Germany',

'Hungary',

'Belgium',

'Italy',

'Singapore',

'Russia',

'Japan',

'Mexico',

'UnitedStates',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2019\_laps["Year"] = 2019

merge2019\_messages["Year"] = 2019

merge2019\_weather["Year"] = 2019

merge2019\_results["Year"] = 2019

merge2019\_final["Year"] = 2019

#%% loading 2020 data

austria\_race\_20a = ff1.get\_session(2020, 1, 'R')

austria\_race\_20a.load()

austria\_race\_20a\_laps = austria\_race\_20a.laps

austria\_race\_20a\_messages = austria\_race\_20a.race\_control\_messages

austria\_race\_20a\_weather = austria\_race\_20a.weather\_data

austria\_race\_20a\_results = austria\_race\_20a.results

austria\_race\_20a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_20a\_final = pd.merge(austria\_race\_20a\_laps, austria\_race\_20a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_20a\_final['TotalTime'] = austria\_race\_20a\_final['Time'] - austria\_race\_20a.session\_start\_time

austria\_race\_20a\_final = austria\_race\_20a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

austria\_race\_20b = ff1.get\_session(2020, 2, 'R')

austria\_race\_20b.load()

austria\_race\_20b\_laps = austria\_race\_20b.laps

austria\_race\_20b\_messages = austria\_race\_20b.race\_control\_messages

austria\_race\_20b\_weather = austria\_race\_20b.weather\_data

austria\_race\_20b\_results = austria\_race\_20b.results

austria\_race\_20b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_20b\_final = pd.merge(austria\_race\_20b\_laps, austria\_race\_20b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_20b\_final['TotalTime'] = austria\_race\_20b\_final['Time'] - austria\_race\_20b.session\_start\_time

austria\_race\_20b\_final = austria\_race\_20b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

hungary\_race\_20 = ff1.get\_session(2020, 3, 'R')

hungary\_race\_20.load()

hungary\_race\_20\_laps = hungary\_race\_20.laps

hungary\_race\_20\_messages = hungary\_race\_20.race\_control\_messages

hungary\_race\_20\_weather = hungary\_race\_20.weather\_data

hungary\_race\_20\_results = hungary\_race\_20.results

hungary\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

hungary\_race\_20\_final = pd.merge(hungary\_race\_20\_laps, hungary\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

hungary\_race\_20\_final['TotalTime'] = hungary\_race\_20\_final['Time'] - hungary\_race\_20.session\_start\_time

hungary\_race\_20\_final = hungary\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

great\_britain\_race\_20a = ff1.get\_session(2020, 4, 'R')

great\_britain\_race\_20a.load()

great\_britain\_race\_20a\_laps = great\_britain\_race\_20a.laps

great\_britain\_race\_20a\_messages = great\_britain\_race\_20a.race\_control\_messages

great\_britain\_race\_20a\_weather = great\_britain\_race\_20a.weather\_data

great\_britain\_race\_20a\_results = great\_britain\_race\_20a.results

great\_britain\_race\_20a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

great\_britain\_race\_20a\_final = pd.merge(great\_britain\_race\_20a\_laps, great\_britain\_race\_20a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

great\_britain\_race\_20a\_final['TotalTime'] = great\_britain\_race\_20a\_final['Time'] - great\_britain\_race\_20a.session\_start\_time

great\_britain\_race\_20a\_final = great\_britain\_race\_20a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

great\_britain\_race\_20b = ff1.get\_session(2020, 5, 'R')

great\_britain\_race\_20b.load()

great\_britain\_race\_20b\_laps = great\_britain\_race\_20b.laps

great\_britain\_race\_20b\_messages = great\_britain\_race\_20b.race\_control\_messages

great\_britain\_race\_20b\_weather = great\_britain\_race\_20b.weather\_data

great\_britain\_race\_20b\_results = great\_britain\_race\_20b.results

great\_britain\_race\_20b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

great\_britain\_race\_20b\_final = pd.merge(great\_britain\_race\_20b\_laps, great\_britain\_race\_20b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

great\_britain\_race\_20b\_final['TotalTime'] = great\_britain\_race\_20b\_final['Time'] - great\_britain\_race\_20b.session\_start\_time

great\_britain\_race\_20b\_final = great\_britain\_race\_20b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

spain\_race\_20 = ff1.get\_session(2020, 6, 'R')

spain\_race\_20.load()

spain\_race\_20\_laps = spain\_race\_20.laps

spain\_race\_20\_messages = spain\_race\_20.race\_control\_messages

spain\_race\_20\_weather = spain\_race\_20.weather\_data

spain\_race\_20\_results = spain\_race\_20.results

spain\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

spain\_race\_20\_final = pd.merge(spain\_race\_20\_laps, spain\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

spain\_race\_20\_final['TotalTime'] = spain\_race\_20\_final['Time'] - spain\_race\_20.session\_start\_time

spain\_race\_20\_final = spain\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

belgium\_race\_20 = ff1.get\_session(2020, 7, 'R')

belgium\_race\_20.load()

belgium\_race\_20\_laps = belgium\_race\_20.laps

belgium\_race\_20\_messages = belgium\_race\_20.race\_control\_messages

belgium\_race\_20\_weather = belgium\_race\_20.weather\_data

belgium\_race\_20\_results = belgium\_race\_20.results

belgium\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

belgium\_race\_20\_final = pd.merge(belgium\_race\_20\_laps, belgium\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

belgium\_race\_20\_final['TotalTime'] = belgium\_race\_20\_final['Time'] - belgium\_race\_20.session\_start\_time

belgium\_race\_20\_final = belgium\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_20a = ff1.get\_session(2020, 8, 'R')

italy\_race\_20a.load()

italy\_race\_20a\_laps = italy\_race\_20a.laps

italy\_race\_20a\_messages = italy\_race\_20a.race\_control\_messages

italy\_race\_20a\_weather = italy\_race\_20a.weather\_data

italy\_race\_20a\_results = italy\_race\_20a.results

italy\_race\_20a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_20a\_final = pd.merge(italy\_race\_20a\_laps, italy\_race\_20a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_20a\_final['TotalTime'] = italy\_race\_20a\_final['Time'] - italy\_race\_20a.session\_start\_time

italy\_race\_20a\_final = italy\_race\_20a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_20b = ff1.get\_session(2020, 9, 'R')

italy\_race\_20b.load()

italy\_race\_20b\_laps = italy\_race\_20b.laps

italy\_race\_20b\_messages = italy\_race\_20b.race\_control\_messages

italy\_race\_20b\_weather = italy\_race\_20b.weather\_data

italy\_race\_20b\_results = italy\_race\_20b.results

italy\_race\_20b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_20b\_final = pd.merge(italy\_race\_20b\_laps, italy\_race\_20b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_20b\_final['TotalTime'] = italy\_race\_20b\_final['Time'] - italy\_race\_20b.session\_start\_time

italy\_race\_20b\_final = italy\_race\_20b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

russia\_race\_20 = ff1.get\_session(2020, 10, 'R')

russia\_race\_20.load()

russia\_race\_20\_laps = russia\_race\_20.laps

russia\_race\_20\_messages = russia\_race\_20.race\_control\_messages

russia\_race\_20\_weather = russia\_race\_20.weather\_data

russia\_race\_20\_results = russia\_race\_20.results

russia\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

russia\_race\_20\_final = pd.merge(russia\_race\_20\_laps, russia\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

russia\_race\_20\_final['TotalTime'] = russia\_race\_20\_final['Time'] - russia\_race\_20.session\_start\_time

russia\_race\_20\_final = russia\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

germany\_race\_20 = ff1.get\_session(2020, 11, 'R')

germany\_race\_20.load()

germany\_race\_20\_laps = germany\_race\_20.laps

germany\_race\_20\_messages = germany\_race\_20.race\_control\_messages

germany\_race\_20\_weather = germany\_race\_20.weather\_data

germany\_race\_20\_results = germany\_race\_20.results

germany\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

germany\_race\_20\_final = pd.merge(germany\_race\_20\_laps, germany\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

germany\_race\_20\_final['TotalTime'] = germany\_race\_20\_final['Time'] - germany\_race\_20.session\_start\_time

germany\_race\_20\_final = germany\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

portugal\_race\_20 = ff1.get\_session(2020, 12, 'R')

portugal\_race\_20.load()

portugal\_race\_20\_laps = portugal\_race\_20.laps

portugal\_race\_20\_messages = portugal\_race\_20.race\_control\_messages

portugal\_race\_20\_weather = portugal\_race\_20.weather\_data

portugal\_race\_20\_results = portugal\_race\_20.results

portugal\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

portugal\_race\_20\_final = pd.merge(portugal\_race\_20\_laps, portugal\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

portugal\_race\_20\_final['TotalTime'] = portugal\_race\_20\_final['Time'] - portugal\_race\_20.session\_start\_time

portugal\_race\_20\_final = portugal\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_20c = ff1.get\_session(2020, 13, 'R')

italy\_race\_20c.load()

italy\_race\_20c\_laps = italy\_race\_20b.laps

italy\_race\_20c\_messages = italy\_race\_20b.race\_control\_messages

italy\_race\_20c\_weather = italy\_race\_20b.weather\_data

italy\_race\_20c\_results = italy\_race\_20b.results

italy\_race\_20c\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_20c\_final = pd.merge(italy\_race\_20c\_laps, italy\_race\_20c\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_20c\_final['TotalTime'] = italy\_race\_20c\_final['Time'] - italy\_race\_20c.session\_start\_time

italy\_race\_20c\_final = italy\_race\_20c\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

turkey\_race\_20 = ff1.get\_session(2020, 14, 'R')

turkey\_race\_20.load()

turkey\_race\_20\_laps = turkey\_race\_20.laps

turkey\_race\_20\_messages = turkey\_race\_20.race\_control\_messages

turkey\_race\_20\_weather = turkey\_race\_20.weather\_data

turkey\_race\_20\_results = turkey\_race\_20.results

turkey\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

turkey\_race\_20\_final = pd.merge(turkey\_race\_20\_laps, turkey\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

turkey\_race\_20\_final['TotalTime'] = turkey\_race\_20\_final['Time'] - turkey\_race\_20.session\_start\_time

turkey\_race\_20\_final = turkey\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

bahrain\_race\_20a = ff1.get\_session(2020, 15, 'R')

bahrain\_race\_20a.load()

bahrain\_race\_20a\_laps = bahrain\_race\_20a.laps

bahrain\_race\_20a\_messages = bahrain\_race\_20a.race\_control\_messages

bahrain\_race\_20a\_weather = bahrain\_race\_20a.weather\_data

bahrain\_race\_20a\_results = bahrain\_race\_20a.results

bahrain\_race\_20a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

bahrain\_race\_20a\_final = pd.merge(bahrain\_race\_20a\_laps, bahrain\_race\_20a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

bahrain\_race\_20a\_final['TotalTime'] = bahrain\_race\_20a\_final['Time'] - bahrain\_race\_20a.session\_start\_time

bahrain\_race\_20a\_final = bahrain\_race\_20a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

bahrain\_race\_20b = ff1.get\_session(2020, 16, 'R')

bahrain\_race\_20b.load()

bahrain\_race\_20b\_laps = bahrain\_race\_20b.laps

bahrain\_race\_20b\_messages = bahrain\_race\_20b.race\_control\_messages

bahrain\_race\_20b\_weather = bahrain\_race\_20b.weather\_data

bahrain\_race\_20b\_results = bahrain\_race\_20b.results

bahrain\_race\_20b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

bahrain\_race\_20b\_final = pd.merge(bahrain\_race\_20b\_laps, bahrain\_race\_20b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

bahrain\_race\_20b\_final['TotalTime'] = bahrain\_race\_20b\_final['Time'] - bahrain\_race\_20b.session\_start\_time

bahrain\_race\_20b\_final = bahrain\_race\_20b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

abu\_dhabi\_race\_20 = ff1.get\_session(2020, 17, 'R')

abu\_dhabi\_race\_20.load()

abu\_dhabi\_race\_20\_laps = abu\_dhabi\_race\_20.laps

abu\_dhabi\_race\_20\_messages = abu\_dhabi\_race\_20.race\_control\_messages

abu\_dhabi\_race\_20\_weather = abu\_dhabi\_race\_20.weather\_data

abu\_dhabi\_race\_20\_results = abu\_dhabi\_race\_20.results

abu\_dhabi\_race\_20\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

abu\_dhabi\_race\_20\_final = pd.merge(abu\_dhabi\_race\_20\_laps, abu\_dhabi\_race\_20\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

abu\_dhabi\_race\_20\_final['TotalTime'] = abu\_dhabi\_race\_20\_final['Time'] - abu\_dhabi\_race\_20.session\_start\_time

abu\_dhabi\_race\_20\_final = abu\_dhabi\_race\_20\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

merge2020\_laps = pd.concat([

austria\_race\_20a\_laps,

austria\_race\_20b\_laps,

hungary\_race\_20\_laps,

great\_britain\_race\_20a\_laps,

great\_britain\_race\_20b\_laps,

spain\_race\_20\_laps,

belgium\_race\_20\_laps,

italy\_race\_20a\_laps,

italy\_race\_20b\_laps,

russia\_race\_20\_laps,

germany\_race\_20\_laps,

portugal\_race\_20\_laps,

italy\_race\_20c\_laps,

turkey\_race\_20\_laps,

bahrain\_race\_20a\_laps,

bahrain\_race\_20b\_laps,

abu\_dhabi\_race\_20\_laps],

keys = ['AustriaA',

'AustriaB',

'Hungary',

'GreatBritainA',

'GreatBritainB',

'Spain',

'Belgium',

'ItalyA',

'ItalyB',

'Russia',

'Germany',

'Portugal',

'ItalyC',

'Turkey',

'BahrainA',

'BahrainB',

'AbuDhabi'],

names=['RACE'])

merge2020\_messages = pd.concat([

austria\_race\_20a\_messages ,

austria\_race\_20b\_messages,

hungary\_race\_20\_messages ,

great\_britain\_race\_20a\_messages ,

great\_britain\_race\_20b\_messages ,

spain\_race\_20\_messages ,

belgium\_race\_20\_messages ,

italy\_race\_20a\_messages ,

italy\_race\_20b\_messages ,

russia\_race\_20\_messages ,

germany\_race\_20\_messages ,

portugal\_race\_20\_messages ,

italy\_race\_20c\_messages ,

turkey\_race\_20\_messages ,

bahrain\_race\_20a\_messages ,

bahrain\_race\_20b\_messages ,

abu\_dhabi\_race\_20\_messages ],

keys = ['AustriaA',

'AustriaB',

'Hungary',

'GreatBritainA',

'GreatBritainB',

'Spain',

'Belgium',

'ItalyA',

'ItalyB',

'Russia',

'Germany',

'Portugal',

'ItalyC',

'Turkey',

'BahrainA',

'BahrainB',

'AbuDhabi'],

names=['RACE'])

merge2020\_weather = pd.concat([

austria\_race\_20a\_weather,

austria\_race\_20b\_weather,

hungary\_race\_20\_weather,

great\_britain\_race\_20a\_weather,

great\_britain\_race\_20b\_weather,

spain\_race\_20\_weather,

belgium\_race\_20\_weather,

italy\_race\_20a\_weather,

italy\_race\_20b\_weather,

russia\_race\_20\_weather,

germany\_race\_20\_weather,

portugal\_race\_20\_weather,

italy\_race\_20c\_weather,

turkey\_race\_20\_weather,

bahrain\_race\_20a\_weather,

bahrain\_race\_20b\_weather,

abu\_dhabi\_race\_20\_weather],

keys = ['AustriaA',

'AustriaB',

'Hungary',

'GreatBritainA',

'GreatBritainB',

'Spain',

'Belgium',

'ItalyA',

'ItalyB',

'Russia',

'Germany',

'Portugal',

'ItalyC',

'Turkey',

'BahrainA',

'BahrainB',

'AbuDhabi'],

names=['RACE'])

merge2020\_results = pd.concat([

austria\_race\_20a\_results ,

austria\_race\_20b\_results ,

hungary\_race\_20\_results ,

great\_britain\_race\_20a\_results ,

great\_britain\_race\_20b\_results ,

spain\_race\_20\_results ,

belgium\_race\_20\_results ,

italy\_race\_20a\_results ,

italy\_race\_20b\_results ,

russia\_race\_20\_results ,

germany\_race\_20\_results ,

portugal\_race\_20\_results ,

italy\_race\_20c\_results ,

turkey\_race\_20\_results ,

bahrain\_race\_20a\_results ,

bahrain\_race\_20b\_results ,

abu\_dhabi\_race\_20\_results ],

keys = ['AustriaA',

'AustriaB',

'Hungary',

'GreatBritainA',

'GreatBritainB',

'Spain',

'Belgium',

'ItalyA',

'ItalyB',

'Russia',

'Germany',

'Portugal',

'ItalyC',

'Turkey',

'BahrainA',

'BahrainB',

'AbuDhabi'],

names=['RACE'])

merge2020\_final = pd.concat([

austria\_race\_20a\_final ,

austria\_race\_20b\_final ,

hungary\_race\_20\_final ,

great\_britain\_race\_20a\_final ,

great\_britain\_race\_20b\_final ,

spain\_race\_20\_final ,

belgium\_race\_20\_final ,

italy\_race\_20a\_final ,

italy\_race\_20b\_final ,

russia\_race\_20\_final ,

germany\_race\_20\_final ,

portugal\_race\_20\_final ,

italy\_race\_20c\_final ,

turkey\_race\_20\_final ,

bahrain\_race\_20a\_final ,

bahrain\_race\_20b\_final ,

abu\_dhabi\_race\_20\_final ],

keys = ['AustriaA',

'AustriaB',

'Hungary',

'GreatBritainA',

'GreatBritainB',

'Spain',

'Belgium',

'ItalyA',

'ItalyB',

'Russia',

'Germany',

'Portugal',

'ItalyC',

'Turkey',

'BahrainA',

'BahrainB',

'AbuDhabi'],

names=['RACE'])

merge2020\_laps["Year"] = 2020

merge2020\_messages["Year"] = 2020

merge2020\_weather["Year"] = 2020

merge2020\_results["Year"] = 2020

merge2020\_final["Year"] = 2020

#%% loading 2021 data

bahrain\_race\_21 = ff1.get\_session(2021, 1, 'R')

bahrain\_race\_21.load()

bahrain\_race\_21\_laps = bahrain\_race\_21.laps

bahrain\_race\_21\_messages= bahrain\_race\_21.race\_control\_messages

bahrain\_race\_21\_weather= bahrain\_race\_21.weather\_data

bahrain\_race\_21\_results = bahrain\_race\_21.results

bahrain\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

bahrain\_race\_21\_final = pd.merge(bahrain\_race\_21\_laps, bahrain\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

bahrain\_race\_21\_final['TotalTime'] = bahrain\_race\_21\_final['Time'] - bahrain\_race\_21.session\_start\_time

bahrain\_race\_21\_final = bahrain\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_21a = ff1.get\_session(2021, 2, 'R')

italy\_race\_21a.load()

italy\_race\_21a\_laps = italy\_race\_21a.laps

italy\_race\_21a\_messages= italy\_race\_21a.race\_control\_messages

italy\_race\_21a\_weather= italy\_race\_21a.weather\_data

italy\_race\_21a\_results= italy\_race\_21a.results

italy\_race\_21a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_21a\_final = pd.merge(italy\_race\_21a\_laps, italy\_race\_21a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_21a\_final['TotalTime'] = italy\_race\_21a\_final['Time'] - italy\_race\_21a.session\_start\_time

italy\_race\_21a\_final = italy\_race\_21a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

portugal\_race\_21 = ff1.get\_session(2021, 3, 'R')

portugal\_race\_21.load()

portugal\_race\_21\_laps = portugal\_race\_21.laps

portugal\_race\_21\_messages= portugal\_race\_21.race\_control\_messages

portugal\_race\_21\_weather= portugal\_race\_21.weather\_data

portugal\_race\_21\_results = portugal\_race\_21.results

portugal\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

portugal\_race\_21\_final = pd.merge(portugal\_race\_21\_laps, portugal\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

portugal\_race\_21\_final['TotalTime'] = portugal\_race\_21\_final['Time'] - portugal\_race\_21.session\_start\_time

portugal\_race\_21\_final = portugal\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

spain\_race\_21 = ff1.get\_session(2021, 4, 'R')

spain\_race\_21.load()

spain\_race\_21\_laps = spain\_race\_21.laps

spain\_race\_21\_messages= spain\_race\_21.race\_control\_messages

spain\_race\_21\_weather= spain\_race\_21.weather\_data

spain\_race\_21\_results= spain\_race\_21.results

spain\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

spain\_race\_21\_final = pd.merge(spain\_race\_21\_laps, spain\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

spain\_race\_21\_final['TotalTime'] = spain\_race\_21\_final['Time'] - spain\_race\_21.session\_start\_time

spain\_race\_21\_final = spain\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

monaco\_race\_21 = ff1.get\_session(2021, 5, 'R')

monaco\_race\_21.load()

monaco\_race\_21\_laps = monaco\_race\_21.laps

monaco\_race\_21\_messages= monaco\_race\_21.race\_control\_messages

monaco\_race\_21\_weather= monaco\_race\_21.weather\_data

monaco\_race\_21\_results= monaco\_race\_21.results

monaco\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

monaco\_race\_21\_final = pd.merge(monaco\_race\_21\_laps, monaco\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

monaco\_race\_21\_final['TotalTime'] = monaco\_race\_21\_final['Time'] - monaco\_race\_21.session\_start\_time

monaco\_race\_21\_final = monaco\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

azerbaijan\_race\_21 = ff1.get\_session(2021, 6, 'R')

azerbaijan\_race\_21.load()

azerbaijan\_race\_21\_laps = azerbaijan\_race\_21.laps

azerbaijan\_race\_21\_messages= azerbaijan\_race\_21.race\_control\_messages

azerbaijan\_race\_21\_weather= azerbaijan\_race\_21.weather\_data

azerbaijan\_race\_21\_results= azerbaijan\_race\_21.results

azerbaijan\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

azerbaijan\_race\_21\_final = pd.merge(azerbaijan\_race\_21\_laps, azerbaijan\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

azerbaijan\_race\_21\_final['TotalTime'] = azerbaijan\_race\_21\_final['Time'] - azerbaijan\_race\_21.session\_start\_time

azerbaijan\_race\_21\_final = azerbaijan\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

france\_race\_21 = ff1.get\_session(2021, 7, 'R')

france\_race\_21.load()

france\_race\_21\_laps = france\_race\_21.laps

france\_race\_21\_messages= france\_race\_21.race\_control\_messages

france\_race\_21\_weather= france\_race\_21.weather\_data

france\_race\_21\_results = france\_race\_21.results

france\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

france\_race\_21\_final = pd.merge(france\_race\_21\_laps, france\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

france\_race\_21\_final['TotalTime'] = france\_race\_21\_final['Time'] - france\_race\_21.session\_start\_time

france\_race\_21\_final = france\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

austria\_race\_21a = ff1.get\_session(2021, 8, 'R')

austria\_race\_21a.load()

austria\_race\_21a\_laps = austria\_race\_21a.laps

austria\_race\_21a\_messages= austria\_race\_21a.race\_control\_messages

austria\_race\_21a\_weather= austria\_race\_21a.weather\_data

austria\_race\_21a\_results = austria\_race\_21a.results

austria\_race\_21a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_21a\_final = pd.merge(austria\_race\_21a\_laps, austria\_race\_21a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_21a\_final['TotalTime'] = austria\_race\_21a\_final['Time'] - austria\_race\_21a.session\_start\_time

austria\_race\_21a\_final = austria\_race\_21a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

austria\_race\_21b = ff1.get\_session(2021, 9, 'R')

austria\_race\_21b.load()

austria\_race\_21b\_laps = austria\_race\_21b.laps

austria\_race\_21b\_messages= austria\_race\_21b.race\_control\_messages

austria\_race\_21b\_weather= austria\_race\_21b.weather\_data

austria\_race\_21b\_results = austria\_race\_21b.results

austria\_race\_21b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_21b\_final = pd.merge(austria\_race\_21b\_laps, austria\_race\_21b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_21b\_final['TotalTime'] = austria\_race\_21b\_final['Time'] - austria\_race\_21b.session\_start\_time

austria\_race\_21b\_final = austria\_race\_21b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

great\_britain\_race\_21 = ff1.get\_session(2021, 10, 'R')

great\_britain\_race\_21.load()

great\_britain\_race\_21\_laps = great\_britain\_race\_21.laps

great\_britain\_race\_21\_messages= great\_britain\_race\_21.race\_control\_messages

great\_britain\_race\_21\_weather= great\_britain\_race\_21.weather\_data

great\_britain\_race\_21\_results = great\_britain\_race\_21.results

great\_britain\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

great\_britain\_race\_21\_final = pd.merge(great\_britain\_race\_21\_laps, great\_britain\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

great\_britain\_race\_21\_final['TotalTime'] = great\_britain\_race\_21\_final['Time'] - great\_britain\_race\_21.session\_start\_time

great\_britain\_race\_21\_final = great\_britain\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

hungary\_race\_21 = ff1.get\_session(2021, 11, 'R')

hungary\_race\_21.load()

hungary\_race\_21\_laps = hungary\_race\_21.laps

hungary\_race\_21\_messages= hungary\_race\_21.race\_control\_messages

hungary\_race\_21\_weather= hungary\_race\_21.weather\_data

hungary\_race\_21\_results = hungary\_race\_21.results

hungary\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

hungary\_race\_21\_final = pd.merge(hungary\_race\_21\_laps, hungary\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

hungary\_race\_21\_final['TotalTime'] = hungary\_race\_21\_final['Time'] - hungary\_race\_21.session\_start\_time

hungary\_race\_21\_final = hungary\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

belgium\_race\_21 = ff1.get\_session(2021, 12, 'R')

belgium\_race\_21.load()

belgium\_race\_21\_laps = belgium\_race\_21.laps

belgium\_race\_21\_messages= belgium\_race\_21.race\_control\_messages

belgium\_race\_21\_weather= belgium\_race\_21.weather\_data

belgium\_race\_21\_results = belgium\_race\_21.results

belgium\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

belgium\_race\_21\_final = pd.merge(belgium\_race\_21\_laps, belgium\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

belgium\_race\_21\_final['TotalTime'] = belgium\_race\_21\_final['Time'] - belgium\_race\_21.session\_start\_time

belgium\_race\_21\_final = belgium\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

netherlands\_race\_21 = ff1.get\_session(2021, 13, 'R')

netherlands\_race\_21 .load()

netherlands\_race\_21\_laps = netherlands\_race\_21.laps

netherlands\_race\_21\_messages= netherlands\_race\_21.race\_control\_messages

netherlands\_race\_21\_weather= netherlands\_race\_21.weather\_data

netherlands\_race\_21\_results = netherlands\_race\_21.results

netherlands\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

netherlands\_race\_21\_final = pd.merge(netherlands\_race\_21\_laps, netherlands\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

netherlands\_race\_21\_final['TotalTime'] = netherlands\_race\_21\_final['Time'] - netherlands\_race\_21.session\_start\_time

netherlands\_race\_21\_final = netherlands\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_21b = ff1.get\_session(2021, 14, 'R')

italy\_race\_21b.load()

italy\_race\_21b\_laps = italy\_race\_21b.laps

italy\_race\_21b\_messages= italy\_race\_21b.race\_control\_messages

italy\_race\_21b\_weather= italy\_race\_21b.weather\_data

italy\_race\_21b\_results = italy\_race\_21b.results

italy\_race\_21b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_21b\_final = pd.merge(italy\_race\_21b\_laps, italy\_race\_21b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_21b\_final['TotalTime'] = italy\_race\_21b\_final['Time'] - italy\_race\_21b.session\_start\_time

italy\_race\_21b\_final = italy\_race\_21b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

russia\_race\_21 = ff1.get\_session(2021, 15, 'R')

russia\_race\_21.load()

russia\_race\_21\_laps = russia\_race\_21.laps

russia\_race\_21\_messages= russia\_race\_21.race\_control\_messages

russia\_race\_21\_weather= russia\_race\_21.weather\_data

russia\_race\_21\_results = russia\_race\_21.results

russia\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

russia\_race\_21\_final = pd.merge(russia\_race\_21\_laps, russia\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

russia\_race\_21\_final['TotalTime'] = russia\_race\_21\_final['Time'] - russia\_race\_21.session\_start\_time

russia\_race\_21\_final = russia\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

turkey\_race\_21 = ff1.get\_session(2021, 16, 'R')

turkey\_race\_21.load()

turkey\_race\_21\_laps = turkey\_race\_21.laps

turkey\_race\_21\_messages= turkey\_race\_21.race\_control\_messages

turkey\_race\_21\_weather= turkey\_race\_21.weather\_data

turkey\_race\_21\_results = turkey\_race\_21.results

turkey\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

turkey\_race\_21\_final = pd.merge(turkey\_race\_21\_laps, turkey\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

turkey\_race\_21\_final['TotalTime'] = turkey\_race\_21\_final['Time'] - turkey\_race\_21.session\_start\_time

turkey\_race\_21\_final = turkey\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

united\_states\_race\_21 = ff1.get\_session(2021, 17, 'R')

united\_states\_race\_21.load()

united\_states\_race\_21\_laps = united\_states\_race\_21.laps

united\_states\_race\_21\_messages= united\_states\_race\_21.race\_control\_messages

united\_states\_race\_21\_weather= united\_states\_race\_21.weather\_data

united\_states\_race\_21\_results = united\_states\_race\_21.results

united\_states\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

united\_states\_race\_21\_final = pd.merge(united\_states\_race\_21\_laps, united\_states\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

united\_states\_race\_21\_final['TotalTime'] = united\_states\_race\_21\_final['Time'] - united\_states\_race\_21.session\_start\_time

united\_states\_race\_21\_final = united\_states\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

mexico\_race\_21 = ff1.get\_session(2021, 18, 'R')

mexico\_race\_21.load()

mexico\_race\_21\_laps = mexico\_race\_21.laps

mexico\_race\_21\_messages= mexico\_race\_21.race\_control\_messages

mexico\_race\_21\_weather= mexico\_race\_21.weather\_data

mexico\_race\_21\_results = mexico\_race\_21.results

mexico\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

mexico\_race\_21\_final = pd.merge(mexico\_race\_21\_laps, mexico\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

mexico\_race\_21\_final['TotalTime'] = mexico\_race\_21\_final['Time'] - mexico\_race\_21.session\_start\_time

mexico\_race\_21\_final = mexico\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

brazil\_race\_21 = ff1.get\_session(2021, 19, 'R')

brazil\_race\_21.load()

brazil\_race\_21\_laps = brazil\_race\_21.laps

brazil\_race\_21\_messages= brazil\_race\_21.race\_control\_messages

brazil\_race\_21\_weather= brazil\_race\_21.weather\_data

brazil\_race\_21\_results = brazil\_race\_21.results

brazil\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

brazil\_race\_21\_final = pd.merge(brazil\_race\_21\_laps, brazil\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

brazil\_race\_21\_final['TotalTime'] = brazil\_race\_21\_final['Time'] - brazil\_race\_21.session\_start\_time

brazil\_race\_21\_final = brazil\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

qatar\_race\_21 = ff1.get\_session(2021, 20, 'R')

qatar\_race\_21.load()

qatar\_race\_21\_laps = qatar\_race\_21.laps

qatar\_race\_21\_messages= qatar\_race\_21.race\_control\_messages

qatar\_race\_21\_weather= qatar\_race\_21.weather\_data

qatar\_race\_21\_results = qatar\_race\_21.results

qatar\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

qatar\_race\_21\_final = pd.merge(qatar\_race\_21\_laps, qatar\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

qatar\_race\_21\_final['TotalTime'] = qatar\_race\_21\_final['Time'] - qatar\_race\_21.session\_start\_time

qatar\_race\_21\_final = qatar\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

saudi\_arabia\_race\_21 = ff1.get\_session(2021, 21, 'R')

saudi\_arabia\_race\_21.load()

saudi\_arabia\_race\_21\_laps = saudi\_arabia\_race\_21.laps

saudi\_arabia\_race\_21\_messages= saudi\_arabia\_race\_21.race\_control\_messages

saudi\_arabia\_race\_21\_weather= saudi\_arabia\_race\_21.weather\_data

saudi\_arabia\_race\_21\_results= saudi\_arabia\_race\_21.results

saudi\_arabia\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

saudi\_arabia\_race\_21\_final = pd.merge(saudi\_arabia\_race\_21\_laps, saudi\_arabia\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

saudi\_arabia\_race\_21\_final['TotalTime'] = saudi\_arabia\_race\_21\_final['Time'] - saudi\_arabia\_race\_21.session\_start\_time

saudi\_arabia\_race\_21\_final = saudi\_arabia\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

abu\_dhabi\_race\_21 = ff1.get\_session(2021, 22, 'R')

abu\_dhabi\_race\_21.load()

abu\_dhabi\_race\_21\_laps = abu\_dhabi\_race\_21.laps

abu\_dhabi\_race\_21\_messages= abu\_dhabi\_race\_21.race\_control\_messages

abu\_dhabi\_race\_21\_weather= abu\_dhabi\_race\_21.weather\_data

abu\_dhabi\_race\_21\_results= abu\_dhabi\_race\_21.results

abu\_dhabi\_race\_21\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

abu\_dhabi\_race\_21\_final = pd.merge(abu\_dhabi\_race\_21\_laps, abu\_dhabi\_race\_21\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

abu\_dhabi\_race\_21\_final['TotalTime'] = abu\_dhabi\_race\_21\_final['Time'] - abu\_dhabi\_race\_21.session\_start\_time

abu\_dhabi\_race\_21\_final = abu\_dhabi\_race\_21\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

merge2021\_laps = pd.concat([

bahrain\_race\_21\_laps,

italy\_race\_21a\_laps,

portugal\_race\_21\_laps,

spain\_race\_21\_laps,

monaco\_race\_21\_laps,

azerbaijan\_race\_21\_laps,

france\_race\_21\_laps,

austria\_race\_21a\_laps,

austria\_race\_21b\_laps,

great\_britain\_race\_21\_laps,

hungary\_race\_21\_laps,

belgium\_race\_21\_laps,

netherlands\_race\_21\_laps,

italy\_race\_21b\_laps,

russia\_race\_21\_laps,

turkey\_race\_21\_laps,

united\_states\_race\_21\_laps,

mexico\_race\_21\_laps,

brazil\_race\_21\_laps,

qatar\_race\_21\_laps,

saudi\_arabia\_race\_21\_laps,

abu\_dhabi\_race\_21\_laps],

keys = ['Bahrain',

'ItalyA',

'Portugal',

'Spain',

'Monaco',

'Azerbaijan',

'France',

'AustriaA',

'AustriaB',

'GreatBritain',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Russia',

'Turkey',

'UnitedStates',

'Mexico',

'Brazil',

'Qatar',

'SaudiArabia',

'AbuDhabi'],

names=['RACE'])

merge2021\_messages = pd.concat([

bahrain\_race\_21\_messages,

italy\_race\_21a\_messages,

portugal\_race\_21\_messages,

spain\_race\_21\_messages,

monaco\_race\_21\_messages,

azerbaijan\_race\_21\_messages,

france\_race\_21\_messages,

austria\_race\_21a\_messages,

austria\_race\_21b\_messages,

great\_britain\_race\_21\_messages,

hungary\_race\_21\_messages,

belgium\_race\_21\_messages,

netherlands\_race\_21\_messages,

italy\_race\_21b\_messages,

russia\_race\_21\_messages,

turkey\_race\_21\_messages,

united\_states\_race\_21\_messages,

mexico\_race\_21\_messages,

brazil\_race\_21\_messages,

qatar\_race\_21\_messages,

saudi\_arabia\_race\_21\_messages,

abu\_dhabi\_race\_21\_messages],

keys = ['Bahrain',

'ItalyA',

'Portugal',

'Spain',

'Monaco',

'Azerbaijan',

'France',

'AustriaA',

'AustriaB',

'GreatBritain',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Russia',

'Turkey',

'UnitedStates',

'Mexico',

'Brazil',

'Qatar',

'SaudiArabia',

'AbuDhabi'],

names=['RACE'])

merge2021\_weather = pd.concat([

bahrain\_race\_21\_weather,

italy\_race\_21a\_weather,

portugal\_race\_21\_weather,

spain\_race\_21\_weather,

monaco\_race\_21\_weather,

azerbaijan\_race\_21\_weather,

france\_race\_21\_weather,

austria\_race\_21a\_weather,

austria\_race\_21b\_weather,

great\_britain\_race\_21\_weather,

hungary\_race\_21\_weather,

belgium\_race\_21\_weather,

netherlands\_race\_21\_weather,

italy\_race\_21b\_weather,

russia\_race\_21\_weather,

turkey\_race\_21\_weather,

united\_states\_race\_21\_weather,

mexico\_race\_21\_weather,

brazil\_race\_21\_weather,

qatar\_race\_21\_weather,

saudi\_arabia\_race\_21\_weather,

abu\_dhabi\_race\_21\_weather],

keys = ['Bahrain',

'ItalyA',

'Portugal',

'Spain',

'Monaco',

'Azerbaijan',

'France',

'AustriaA',

'AustriaB',

'GreatBritain',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Russia',

'Turkey',

'UnitedStates',

'Mexico',

'Brazil',

'Qatar',

'SaudiArabia',

'AbuDhabi'],

names=['RACE'])

merge2021\_results = pd.concat([

bahrain\_race\_21\_results,

italy\_race\_21a\_results,

portugal\_race\_21\_results,

spain\_race\_21\_results,

monaco\_race\_21\_results,

azerbaijan\_race\_21\_results,

france\_race\_21\_results,

austria\_race\_21a\_results,

austria\_race\_21b\_results,

great\_britain\_race\_21\_results,

hungary\_race\_21\_results,

belgium\_race\_21\_results,

netherlands\_race\_21\_results,

italy\_race\_21b\_results,

russia\_race\_21\_results,

turkey\_race\_21\_results,

united\_states\_race\_21\_results,

mexico\_race\_21\_results,

brazil\_race\_21\_results,

qatar\_race\_21\_results,

saudi\_arabia\_race\_21\_results,

abu\_dhabi\_race\_21\_results],

keys = ['Bahrain',

'ItalyA',

'Portugal',

'Spain',

'Monaco',

'Azerbaijan',

'France',

'AustriaA',

'AustriaB',

'GreatBritain',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Russia',

'Turkey',

'UnitedStates',

'Mexico',

'Brazil',

'Qatar',

'SaudiArabia',

'AbuDhabi'],

names=['RACE'])

merge2021\_final = pd.concat([

bahrain\_race\_21\_final,

italy\_race\_21a\_final,

portugal\_race\_21\_final,

spain\_race\_21\_final,

monaco\_race\_21\_final,

azerbaijan\_race\_21\_final,

france\_race\_21\_final,

austria\_race\_21a\_final,

austria\_race\_21b\_final,

great\_britain\_race\_21\_final,

hungary\_race\_21\_final,

belgium\_race\_21\_final,

netherlands\_race\_21\_final,

italy\_race\_21b\_final,

russia\_race\_21\_final,

turkey\_race\_21\_final,

united\_states\_race\_21\_final,

mexico\_race\_21\_final,

brazil\_race\_21\_final,

qatar\_race\_21\_final,

saudi\_arabia\_race\_21\_final,

abu\_dhabi\_race\_21\_final],

keys = ['Bahrain',

'ItalyA',

'Portugal',

'Spain',

'Monaco',

'Azerbaijan',

'France',

'AustriaA',

'AustriaB',

'GreatBritain',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Russia',

'Turkey',

'UnitedStates',

'Mexico',

'Brazil',

'Qatar',

'SaudiArabia',

'AbuDhabi'],

names=['RACE'])

merge2021\_laps["Year"] = 2021

merge2021\_messages["Year"] = 2021

merge2021\_weather["Year"] = 2021

merge2021\_results["Year"] = 2021

merge2021\_final["Year"] = 2021

#%% loading 2022 data

bahrain\_race\_22 = ff1.get\_session(2022, 1, 'R')

bahrain\_race\_22.load()

bahrain\_race\_22\_laps = bahrain\_race\_22.laps

bahrain\_race\_22\_messages = bahrain\_race\_22.race\_control\_messages

bahrain\_race\_22\_weather= bahrain\_race\_22.weather\_data

bahrain\_race\_22\_results= bahrain\_race\_22.results

bahrain\_race\_22\_results.columns

bahrain\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

bahrain\_race\_22\_final = pd.merge(bahrain\_race\_22\_laps, bahrain\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

bahrain\_race\_22\_final['TotalTime'] = bahrain\_race\_22\_final['Time'] - bahrain\_race\_22.session\_start\_time

bahrain\_race\_22\_final = bahrain\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

saudi\_arabia\_race\_22 = ff1.get\_session(2022, 2, 'R')

saudi\_arabia\_race\_22.load()

saudi\_arabia\_race\_22\_laps = saudi\_arabia\_race\_22.laps

saudi\_arabia\_race\_22\_messages = saudi\_arabia\_race\_22.race\_control\_messages

saudi\_arabia\_race\_22\_weather= saudi\_arabia\_race\_22.weather\_data

saudi\_arabia\_race\_22\_results= saudi\_arabia\_race\_22.results

saudi\_arabia\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

saudi\_arabia\_race\_22\_final = pd.merge(saudi\_arabia\_race\_22\_laps, saudi\_arabia\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

saudi\_arabia\_race\_22\_final['TotalTime'] = saudi\_arabia\_race\_22\_final['Time'] - saudi\_arabia\_race\_22.session\_start\_time

saudi\_arabia\_race\_22\_final = saudi\_arabia\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

australia\_race\_22 = ff1.get\_session(2022, 3, 'R')

australia\_race\_22.load()

australia\_race\_22\_laps = australia\_race\_22.laps

australia\_race\_22\_messages = australia\_race\_22.race\_control\_messages

australia\_race\_22\_weather= australia\_race\_22.weather\_data

australia\_race\_22\_results= australia\_race\_22.results

australia\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

australia\_race\_22\_final = pd.merge(australia\_race\_22\_laps, australia\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

australia\_race\_22\_final['TotalTime'] = australia\_race\_22\_final['Time'] - australia\_race\_22.session\_start\_time

australia\_race\_22\_final = australia\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_22a = ff1.get\_session(2022, 4, 'R')

italy\_race\_22a.load()

italy\_race\_22a\_laps = italy\_race\_22a.laps

italy\_race\_22a\_messages = italy\_race\_22a.race\_control\_messages

italy\_race\_22a\_weather= italy\_race\_22a.weather\_data

italy\_race\_22a\_results= italy\_race\_22a.results

italy\_race\_22a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_22a\_final = pd.merge(italy\_race\_22a\_laps, italy\_race\_22a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_22a\_final['TotalTime'] = italy\_race\_22a\_final['Time'] - italy\_race\_22a.session\_start\_time

italy\_race\_22a\_final = italy\_race\_22a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

united\_states\_race\_22a = ff1.get\_session(2022, 5, 'R')

united\_states\_race\_22a.load()

united\_states\_race\_22a\_laps = united\_states\_race\_22a.laps

united\_states\_race\_22a\_messages= united\_states\_race\_22a.race\_control\_messages

united\_states\_race\_22a\_weather= united\_states\_race\_22a.weather\_data

united\_states\_race\_22a\_results= united\_states\_race\_22a.results

united\_states\_race\_22a\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

united\_states\_race\_22a\_final = pd.merge(united\_states\_race\_22a\_laps, united\_states\_race\_22a\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

united\_states\_race\_22a\_final['TotalTime'] = united\_states\_race\_22a\_final['Time'] - united\_states\_race\_22a.session\_start\_time

united\_states\_race\_22a\_final = united\_states\_race\_22a\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

spain\_race\_22 = ff1.get\_session(2022, 6, 'R')

spain\_race\_22.load()

spain\_race\_22\_laps = spain\_race\_22.laps

spain\_race\_22\_messages= spain\_race\_22.race\_control\_messages

spain\_race\_22\_weather= spain\_race\_22.weather\_data

spain\_race\_22\_results= spain\_race\_22.results

spain\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

spain\_race\_22\_final = pd.merge(spain\_race\_22\_laps, spain\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

spain\_race\_22\_final['TotalTime'] = spain\_race\_22\_final['Time'] - spain\_race\_22.session\_start\_time

spain\_race\_22\_final = spain\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

monaco\_race\_22 = ff1.get\_session(2022, 7, 'R')

monaco\_race\_22.load()

monaco\_race\_22\_laps = monaco\_race\_22.laps

monaco\_race\_22\_messages= monaco\_race\_22.race\_control\_messages

monaco\_race\_22\_weather= monaco\_race\_22.weather\_data

monaco\_race\_22\_results= monaco\_race\_22.results

monaco\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

monaco\_race\_22\_final = pd.merge(monaco\_race\_22\_laps, monaco\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

monaco\_race\_22\_final['TotalTime'] = monaco\_race\_22\_final['Time'] - monaco\_race\_22.session\_start\_time

monaco\_race\_22\_final = monaco\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

azerbaijan\_race\_22 = ff1.get\_session(2022, 8, 'R')

azerbaijan\_race\_22.load()

azerbaijan\_race\_22\_laps = azerbaijan\_race\_22.laps

azerbaijan\_race\_22\_messages= azerbaijan\_race\_22.race\_control\_messages

azerbaijan\_race\_22\_weather= azerbaijan\_race\_22.weather\_data

azerbaijan\_race\_22\_results= azerbaijan\_race\_22.results

azerbaijan\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

azerbaijan\_race\_22\_final = pd.merge(azerbaijan\_race\_22\_laps, azerbaijan\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

azerbaijan\_race\_22\_final['TotalTime'] = azerbaijan\_race\_22\_final['Time'] - azerbaijan\_race\_22.session\_start\_time

azerbaijan\_race\_22\_final = azerbaijan\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

canada\_race\_22 = ff1.get\_session(2022, 9, 'R')

canada\_race\_22.load()

canada\_race\_22\_laps = canada\_race\_22.laps

canada\_race\_22\_messages= canada\_race\_22.race\_control\_messages

canada\_race\_22\_weather= canada\_race\_22.weather\_data

canada\_race\_22\_results= canada\_race\_22.results

canada\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

canada\_race\_22\_final = pd.merge(canada\_race\_22\_laps, canada\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

canada\_race\_22\_final['TotalTime'] = canada\_race\_22\_final['Time'] - canada\_race\_22.session\_start\_time

canada\_race\_22\_final = canada\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

great\_britain\_race\_22 = ff1.get\_session(2022, 10, 'R')

great\_britain\_race\_22.load()

great\_britain\_race\_22\_laps = azerbaijan\_race\_22.laps

great\_britain\_race\_22\_messages= azerbaijan\_race\_22.race\_control\_messages

great\_britain\_race\_22\_weather= azerbaijan\_race\_22.weather\_data

great\_britain\_race\_22\_results= azerbaijan\_race\_22.results

great\_britain\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

great\_britain\_race\_22\_final = pd.merge(great\_britain\_race\_22\_laps, great\_britain\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

great\_britain\_race\_22\_final['TotalTime'] = great\_britain\_race\_22\_final['Time'] - great\_britain\_race\_22.session\_start\_time

great\_britain\_race\_22\_final = great\_britain\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

austria\_race\_22 = ff1.get\_session(2022, 11, 'R')

austria\_race\_22.load()

austria\_race\_22\_laps = austria\_race\_22.laps

austria\_race\_22\_messages= austria\_race\_22.race\_control\_messages

austria\_race\_22\_weather= austria\_race\_22.weather\_data

austria\_race\_22\_results= austria\_race\_22.results

austria\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

austria\_race\_22\_final = pd.merge(austria\_race\_22\_laps, austria\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

austria\_race\_22\_final['TotalTime'] = austria\_race\_22\_final['Time'] - austria\_race\_22.session\_start\_time

austria\_race\_22\_final = austria\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

france\_race\_22 = ff1.get\_session(2022, 12, 'R')

france\_race\_22.load()

france\_race\_22\_laps = france\_race\_22.laps

france\_race\_22\_messages= france\_race\_22.race\_control\_messages

france\_race\_22\_weather= france\_race\_22.weather\_data

france\_race\_22\_results= france\_race\_22.results

france\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

france\_race\_22\_final = pd.merge(france\_race\_22\_laps, france\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

france\_race\_22\_final['TotalTime'] = france\_race\_22\_final['Time'] - france\_race\_22.session\_start\_time

france\_race\_22\_final = france\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

hungary\_race\_22 = ff1.get\_session(2022, 13, 'R')

hungary\_race\_22.load()

hungary\_race\_22\_laps = hungary\_race\_22.laps

hungary\_race\_22\_messages= hungary\_race\_22.race\_control\_messages

hungary\_race\_22\_weather= hungary\_race\_22.weather\_data

hungary\_race\_22\_results= hungary\_race\_22.results

hungary\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

hungary\_race\_22\_final = pd.merge(hungary\_race\_22\_laps, hungary\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

hungary\_race\_22\_final['TotalTime'] = hungary\_race\_22\_final['Time'] - hungary\_race\_22.session\_start\_time

hungary\_race\_22\_final = hungary\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

belgium\_race\_22 = ff1.get\_session(2022, 14, 'R')

belgium\_race\_22.load()

belgium\_race\_22\_laps = belgium\_race\_22.laps

belgium\_race\_22\_messages= belgium\_race\_22.race\_control\_messages

belgium\_race\_22\_weather= belgium\_race\_22.weather\_data

belgium\_race\_22\_results= belgium\_race\_22.results

belgium\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

belgium\_race\_22\_final = pd.merge(belgium\_race\_22\_laps, belgium\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

belgium\_race\_22\_final['TotalTime'] = belgium\_race\_22\_final['Time'] - belgium\_race\_22.session\_start\_time

belgium\_race\_22\_final = belgium\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

netherlands\_race\_22 = ff1.get\_session(2022, 15, 'R')

netherlands\_race\_22.load()

netherlands\_race\_22\_laps = netherlands\_race\_22.laps

netherlands\_race\_22\_messages= netherlands\_race\_22.race\_control\_messages

netherlands\_race\_22\_weather= netherlands\_race\_22.weather\_data

netherlands\_race\_22\_results= netherlands\_race\_22.results

netherlands\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

netherlands\_race\_22\_final = pd.merge(netherlands\_race\_22\_laps, netherlands\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

netherlands\_race\_22\_final['TotalTime'] = netherlands\_race\_22\_final['Time'] - netherlands\_race\_22.session\_start\_time

netherlands\_race\_22\_final = netherlands\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

italy\_race\_22b = ff1.get\_session(2022, 16, 'R')

italy\_race\_22b.load()

italy\_race\_22b\_laps = italy\_race\_22b.laps

italy\_race\_22b\_messages= italy\_race\_22b.race\_control\_messages

italy\_race\_22b\_weather= italy\_race\_22b.weather\_data

italy\_race\_22b\_results= italy\_race\_22b.results

italy\_race\_22b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

italy\_race\_22b\_final = pd.merge(italy\_race\_22b\_laps, italy\_race\_22b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

italy\_race\_22b\_final['TotalTime'] = italy\_race\_22b\_final['Time'] - italy\_race\_22b.session\_start\_time

italy\_race\_22b\_final = italy\_race\_22b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

singapore\_race\_22 = ff1.get\_session(2022, 17, 'R')

singapore\_race\_22.load()

singapore\_race\_22\_laps = singapore\_race\_22.laps

singapore\_race\_22\_messages= singapore\_race\_22.race\_control\_messages

singapore\_race\_22\_weather= singapore\_race\_22.weather\_data

singapore\_race\_22\_results= singapore\_race\_22.results

singapore\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

singapore\_race\_22\_final = pd.merge(singapore\_race\_22\_laps, singapore\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

singapore\_race\_22\_final['TotalTime'] = singapore\_race\_22\_final['Time'] - singapore\_race\_22.session\_start\_time

singapore\_race\_22\_final = singapore\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

japan\_race\_22 = ff1.get\_session(2022, 18, 'R')

japan\_race\_22.load()

japan\_race\_22\_laps = japan\_race\_22.laps

japan\_race\_22\_messages= japan\_race\_22.race\_control\_messages

japan\_race\_22\_weather = japan\_race\_22.weather\_data

japan\_race\_22\_results= japan\_race\_22.results

japan\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

japan\_race\_22\_final = pd.merge(japan\_race\_22\_laps, japan\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

japan\_race\_22\_final['TotalTime'] = japan\_race\_22\_final['Time'] - japan\_race\_22.session\_start\_time

japan\_race\_22\_final = japan\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

united\_states\_race\_22b = ff1.get\_session(2022, 19, 'R')

united\_states\_race\_22b.load()

united\_states\_race\_22b\_laps = united\_states\_race\_22b.laps

united\_states\_race\_22b\_messages= united\_states\_race\_22b.race\_control\_messages

united\_states\_race\_22b\_weather= united\_states\_race\_22b.weather\_data

united\_states\_race\_22b\_results= united\_states\_race\_22b.results

united\_states\_race\_22b\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

united\_states\_race\_22b\_final = pd.merge(united\_states\_race\_22b\_laps, united\_states\_race\_22b\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

united\_states\_race\_22b\_final['TotalTime'] = united\_states\_race\_22b\_final['Time'] - united\_states\_race\_22b.session\_start\_time

united\_states\_race\_22b\_final = united\_states\_race\_22b\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

mexico\_race\_22 = ff1.get\_session(2022, 20, 'R')

mexico\_race\_22.load()

mexico\_race\_22\_laps = mexico\_race\_22.laps

mexico\_race\_22\_messages= mexico\_race\_22.race\_control\_messages

mexico\_race\_22\_weather = mexico\_race\_22.weather\_data

mexico\_race\_22\_results= mexico\_race\_22.results

mexico\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

mexico\_race\_22\_final = pd.merge(mexico\_race\_22\_laps, mexico\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

mexico\_race\_22\_final['TotalTime'] = mexico\_race\_22\_final['Time'] - mexico\_race\_22.session\_start\_time

mexico\_race\_22\_final = mexico\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

brazil\_race\_22 = ff1.get\_session(2022, 21, 'R')

brazil\_race\_22.load()

brazil\_race\_22\_laps = brazil\_race\_22.laps

brazil\_race\_22\_messages= brazil\_race\_22.race\_control\_messages

brazil\_race\_22\_weather = brazil\_race\_22.weather\_data

brazil\_race\_22\_results= brazil\_race\_22.results

brazil\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

brazil\_race\_22\_final = pd.merge(brazil\_race\_22\_laps, brazil\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

brazil\_race\_22\_final['TotalTime'] = brazil\_race\_22\_final['Time'] - brazil\_race\_22.session\_start\_time

brazil\_race\_22\_final = brazil\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

abu\_dhabi\_race\_22 = ff1.get\_session(2022, 22, 'R')

abu\_dhabi\_race\_22.load()

abu\_dhabi\_race\_22\_laps = abu\_dhabi\_race\_22.laps

abu\_dhabi\_race\_22\_messages= abu\_dhabi\_race\_22.race\_control\_messages

abu\_dhabi\_race\_22\_weather = abu\_dhabi\_race\_22.weather\_data

abu\_dhabi\_race\_22\_results= abu\_dhabi\_race\_22.results

abu\_dhabi\_race\_22\_results.rename(columns={'Abbreviation': 'Driver'}, inplace=True)

abu\_dhabi\_race\_22\_final = pd.merge(abu\_dhabi\_race\_22\_laps, abu\_dhabi\_race\_22\_results[['Driver', 'GridPosition', 'Position', 'Points']], on='Driver', how='left')

abu\_dhabi\_race\_22\_final['TotalTime'] = abu\_dhabi\_race\_22\_final['Time'] - abu\_dhabi\_race\_22.session\_start\_time

abu\_dhabi\_race\_22\_final = abu\_dhabi\_race\_22\_final.groupby('Driver').tail(1000).sort\_values(by='TotalTime')

merge2022\_laps = pd.concat([

bahrain\_race\_22\_laps,

saudi\_arabia\_race\_22\_laps,

australia\_race\_22\_laps,

italy\_race\_22a\_laps,

united\_states\_race\_22a\_laps,

spain\_race\_22\_laps,

monaco\_race\_22\_laps,

azerbaijan\_race\_22\_laps,

canada\_race\_22\_laps,

great\_britain\_race\_22\_laps,

austria\_race\_22\_laps,

france\_race\_22\_laps,

hungary\_race\_22\_laps,

belgium\_race\_22\_laps,

netherlands\_race\_22\_laps,

italy\_race\_22b\_laps,

singapore\_race\_22\_laps,

japan\_race\_22\_laps,

united\_states\_race\_22b\_laps,

mexico\_race\_22\_laps,

brazil\_race\_22\_laps,

abu\_dhabi\_race\_22\_laps],

keys = ['Bahrain',

'SaudiArabia',

'Australia',

'ItalyA',

'UnitedStatesA',

'Spain',

'Monaco',

'Azerbaijan',

'Canada',

'GreatBritain',

'Austria',

'France',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Singapore',

'Japan',

'UnitedStatesB',

'Mexico',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2022\_messages = pd.concat([

bahrain\_race\_22\_messages,

saudi\_arabia\_race\_22\_messages,

australia\_race\_22\_messages,

italy\_race\_22a\_messages,

united\_states\_race\_22a\_messages,

spain\_race\_22\_messages,

monaco\_race\_22\_messages,

azerbaijan\_race\_22\_messages,

canada\_race\_22\_messages,

great\_britain\_race\_22\_messages,

austria\_race\_22\_messages,

france\_race\_22\_messages,

hungary\_race\_22\_messages,

belgium\_race\_22\_messages,

netherlands\_race\_22\_messages,

italy\_race\_22b\_messages,

singapore\_race\_22\_messages,

japan\_race\_22\_messages,

united\_states\_race\_22b\_messages,

mexico\_race\_22\_messages,

brazil\_race\_22\_messages,

abu\_dhabi\_race\_22\_messages],

keys = ['Bahrain',

'SaudiArabia',

'Australia',

'ItalyA',

'UnitedStatesA',

'Spain',

'Monaco',

'Azerbaijan',

'Canada',

'GreatBritain',

'Austria',

'France',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Singapore',

'Japan',

'UnitedStatesB',

'Mexico',

'Brazil',

'Abu Dhabi'],

names=['RACE'])

merge2022\_weather = pd.concat([

bahrain\_race\_22\_weather,

saudi\_arabia\_race\_22\_weather,

australia\_race\_22\_weather,

italy\_race\_22a\_weather,

united\_states\_race\_22a\_weather,

spain\_race\_22\_weather,

monaco\_race\_22\_weather,

azerbaijan\_race\_22\_weather,

canada\_race\_22\_weather,

great\_britain\_race\_22\_weather,

austria\_race\_22\_weather,

france\_race\_22\_weather,

hungary\_race\_22\_weather,

belgium\_race\_22\_weather,

netherlands\_race\_22\_weather,

italy\_race\_22b\_weather,

singapore\_race\_22\_weather,

japan\_race\_22\_weather,

united\_states\_race\_22b\_weather,

mexico\_race\_22\_weather,

brazil\_race\_22\_weather,

abu\_dhabi\_race\_22\_weather],

keys = ['Bahrain',

'SaudiArabia',

'Australia',

'ItalyA',

'UnitedStatesA',

'Spain',

'Monaco',

'Azerbaijan',

'Canada',

'GreatBritain',

'Austria',

'France',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Singapore',

'Japan',

'UnitedStatesB',

'Mexico',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2022\_results = pd.concat([

bahrain\_race\_22\_results,

saudi\_arabia\_race\_22\_results,

australia\_race\_22\_results,

italy\_race\_22a\_results,

united\_states\_race\_22a\_results,

spain\_race\_22\_results,

monaco\_race\_22\_results,

azerbaijan\_race\_22\_results,

canada\_race\_22\_results,

great\_britain\_race\_22\_results,

austria\_race\_22\_results,

france\_race\_22\_results,

hungary\_race\_22\_results,

belgium\_race\_22\_results,

netherlands\_race\_22\_results,

italy\_race\_22b\_results,

singapore\_race\_22\_results,

japan\_race\_22\_results,

united\_states\_race\_22b\_results,

mexico\_race\_22\_results,

brazil\_race\_22\_results,

abu\_dhabi\_race\_22\_results],

keys = ['Bahrain',

'SaudiArabia',

'Australia',

'ItalyA',

'UnitedStatesA',

'Spain',

'Monaco',

'Azerbaijan',

'Canada',

'GreatBritain',

'Austria',

'France',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Singapore',

'Japan',

'UnitedStatesB',

'Mexico',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2022\_final = pd.concat([

bahrain\_race\_22\_final,

saudi\_arabia\_race\_22\_final,

australia\_race\_22\_final,

italy\_race\_22a\_final,

united\_states\_race\_22a\_final,

spain\_race\_22\_final,

monaco\_race\_22\_final,

azerbaijan\_race\_22\_final,

canada\_race\_22\_final,

great\_britain\_race\_22\_final,

austria\_race\_22\_final,

france\_race\_22\_final,

hungary\_race\_22\_final,

belgium\_race\_22\_final,

netherlands\_race\_22\_final,

italy\_race\_22b\_final,

singapore\_race\_22\_final,

japan\_race\_22\_final,

united\_states\_race\_22b\_final,

mexico\_race\_22\_final,

brazil\_race\_22\_final,

abu\_dhabi\_race\_22\_final],

keys = ['Bahrain',

'SaudiArabia',

'Australia',

'ItalyA',

'UnitedStatesA',

'Spain',

'Monaco',

'Azerbaijan',

'Canada',

'GreatBritain',

'Austria',

'France',

'Hungary',

'Belgium',

'Netherlands',

'ItalyB',

'Singapore',

'Japan',

'UnitedStatesB',

'Mexico',

'Brazil',

'AbuDhabi'],

names=['RACE'])

merge2022\_laps["Year"] = 2022

merge2022\_messages["Year"] = 2022

merge2022\_weather["Year"] = 2022

merge2022\_results["Year"] = 2022

merge2022\_final["Year"] = 2022

#%% merging the yearly data

merge\_total\_laps = pd.concat([

merge2018\_laps,

merge2019\_laps,

merge2020\_laps,

merge2021\_laps,

merge2022\_laps])

merge\_total\_messages = pd.concat([

merge2018\_messages,

merge2019\_messages,

merge2020\_messages,

merge2021\_messages,

merge2022\_messages])

merge\_total\_weather = pd.concat([

merge2018\_weather,

merge2019\_weather,

merge2020\_weather,

merge2021\_weather,

merge2022\_weather])

merge\_total\_results = pd.concat([

merge2018\_results,

merge2019\_results,

merge2020\_results,

merge2021\_results,

merge2022\_results])

merge\_total\_final = pd.concat([

merge2018\_final,

merge2019\_final,

merge2020\_final,

merge2021\_final,

merge2022\_final])

merge\_total\_laps.insert(0, "NewID", range(1, 1 + len(merge\_total\_laps)))

merge\_total\_laps.drop(["Year"], axis=1)

merge\_total\_laps.loc[58170:71068, ["Year"]] = 2022

merge\_total\_messages.insert(0, "NewID", range(1, 1 + len(merge\_total\_messages)))

merge\_total\_messages.drop(["Year"], axis=1)

merge\_total\_messages.loc[58170:71068, ["Year"]] = 2022

merge\_total\_weather.insert(0, "NewID", range(1, 1 + len(merge\_total\_weather)))

merge\_total\_weather.drop(["Year"], axis=1)

merge\_total\_weather.loc[58170:71068, ["Year"]] = 2022

merge\_total\_results.insert(0, "NewID", range(1, 1 + len(merge\_total\_results)))

merge\_total\_results.drop(["Year"], axis=1)

merge\_total\_results.loc[58170:71068, ["Year"]] = 2022

merge\_total\_final.insert(0, "NewID", range(1, 1 + len(merge\_total\_final)))

merge\_total\_final.drop(["Year"], axis=1)

merge\_total\_final.loc[58170:71068, ["Year"]] = 2022

#%% exporting the data to csv

merge\_total\_laps.to\_csv("C:\\Users\\andre\\OneDrive\\Desktop\\laps.csv", na\_rep='NA')

merge\_total\_messages.to\_csv("C:\\Users\\andre\\OneDrive\\Desktop\\messages.csv", na\_rep='NA')

merge\_total\_weather.to\_csv("C:\\Users\\andre\\OneDrive\\Desktop\\weather.csv", na\_rep='NA')

merge\_total\_results.to\_csv("C:\\Users\\andre\\OneDrive\\Desktop\\results.csv", na\_rep='NA')

merge\_total\_final.to\_csv("C:\\Users\\andre\\OneDrive\\Desktop\\final.csv", na\_rep='NA')

#%% pit only dataframe and reducing dimensionality

#making RACE a column

merge\_total\_final.reset\_index(inplace=True)

#calculating gap

merge\_total\_final['Gap'] = merge\_total\_final['TotalTime'] - merge\_total\_final['TotalTime'].shift()

merge\_total\_final['GapSeconds'] = merge\_total\_final['Gap'].dt.total\_seconds()

merge\_total\_final['GapSeconds'] = (merge\_total\_final.GapSeconds.clip(lower=0))

#making pit stop only dataframe and removing the first lap

merge\_total\_final\_pit\_only = merge\_total\_final[pd.notnull(merge\_total\_final['PitInTime'])]

merge\_total\_final\_pit\_only = merge\_total\_final\_pit\_only[merge\_total\_final\_pit\_only['LapNumber'] != 1]

#reducing dimensionality

merge\_total\_final = merge\_total\_final[[

'DriverNumber', 'LapNumber', 'Compound', 'TyreLife', 'FreshTyre', 'Stint', 'Team', 'Year',

'RACE', 'Driver', 'TrackStatus', 'GridPosition', 'Position', 'Points', 'TotalTime', 'GapSeconds']]

merge\_total\_final\_pit\_only = merge\_total\_final\_pit\_only[[

'DriverNumber', 'LapNumber', 'Compound', 'TyreLife', 'FreshTyre', 'Stint', 'Team', 'Year',

'RACE', 'Driver', 'TrackStatus', 'GridPosition', 'Position', 'Points', 'TotalTime', 'GapSeconds']]

merge\_total\_final\_finishers\_only = merge\_total\_final.loc[merge\_total\_final['Points'] != 0]

#transforming string data into categorical/factor https://pbpython.com/categorical-encoding.html

merge\_total\_final\_pit\_only['Team'] = merge\_total\_final\_pit\_only['Team'].astype('category')

merge\_total\_final\_pit\_only["TeamCat"] = merge\_total\_final\_pit\_only["Team"].cat.codes

merge\_total\_final\_pit\_only['Driver'] = merge\_total\_final\_pit\_only['Driver'].astype('category')

merge\_total\_final\_pit\_only["DriverCat"] = merge\_total\_final\_pit\_only["Driver"].cat.codes

merge\_total\_final\_pit\_only['DriverNumber'] = merge\_total\_final\_pit\_only['DriverNumber'].astype('category')

merge\_total\_final\_pit\_only["DriverNumberCat"] = merge\_total\_final\_pit\_only["DriverNumber"].cat.codes

merge\_total\_final\_pit\_only['TrackStatus'] = merge\_total\_final\_pit\_only['TrackStatus'].astype('category')

merge\_total\_final\_pit\_only["TrackStatusCat"] = merge\_total\_final\_pit\_only["TrackStatus"].cat.codes

merge\_total\_final\_pit\_only['Compound'] = merge\_total\_final\_pit\_only['Compound'].astype('category')

merge\_total\_final\_pit\_only["CompoundCat"] = merge\_total\_final\_pit\_only["Compound"].cat.codes

merge\_total\_final\_pit\_only = merge\_total\_final\_pit\_only[[

'DriverNumberCat', 'LapNumber', 'CompoundCat', 'TyreLife', 'FreshTyre', 'Stint', 'TeamCat', 'Year',

'RACE', 'DriverCat', 'TrackStatusCat', 'GridPosition', 'Position', 'Points', 'TotalTime', 'GapSeconds']]

merge\_total\_final\_pit\_only['TotalTime'] = merge\_total\_final\_pit\_only['TotalTime'].dt.total\_seconds()

#%%

#aggregating the data

pit\_in\_time\_lap\_number = merge\_total\_final\_pit\_only.groupby(['Year', 'RACE', 'TeamCat', 'DriverCat']).agg({'LapNumber': ['mean', 'min', 'max', 'count']}).reset\_index()

pit\_in\_time\_lap\_number = pit\_in\_time\_lap\_number.droplevel(0, axis=1)

pit\_in\_time\_lap\_number.reset\_index(inplace=True)

pit\_in\_time\_lap\_number = pit\_in\_time\_lap\_number.astype({'count': 'float'})

pit\_in\_time\_lap\_number.columns = ['Index', 'Year', 'Race', 'Team', 'Driver', 'Mean', 'Min', 'Max', 'NumberOfStops']

#visualizing the normalized value of pit stop frequency

pit\_in\_time\_lap\_number.NumberOfStops.value\_counts(normalize=True).plot(kind='bar')

#visualizing the actual number of pit stop frequency

pit\_in\_time\_lap\_number.NumberOfStops.value\_counts(bins=8).plot(kind='bar')

#analyze tyre life when pit stop is made

pit\_in\_time\_tyre\_life = merge\_total\_final\_pit\_only.groupby(['Year', 'RACE', 'TeamCat', 'DriverCat']).agg({'TyreLife': lambda x: x.tolist(),'LapNumber': ['mean', 'min', 'max', 'count']}).reset\_index()

pit\_in\_time\_tyre\_life = pit\_in\_time\_tyre\_life.droplevel(0, axis=1)

pit\_in\_time\_tyre\_life.reset\_index(inplace=True)

pit\_in\_time\_tyre\_life = pit\_in\_time\_tyre\_life.astype({'count': 'float'})

pit\_in\_time\_tyre\_life.columns = ['Index', 'Year', 'Race', 'Team', 'Driver', 'TyreLife', 'Mean', 'Min', 'Max', 'NumberOfStops']

#pit in time plot

pit\_in\_time\_tyre\_life.TyreLife.value\_counts(normalize=False).plot(kind='bar')

pit\_in\_time\_tyre\_life.TyreLife.value\_counts(normalize=True).plot(kind='line')

#%%

#dataframe showing the tyre life when a pitstop was made

#pit\_stop\_tyre\_life = pd.DataFrame(pit\_in\_time\_tyre\_life.TyreLife.tolist(), index=pit\_in\_time\_tyre\_life.index)

#combine pitStop\_TyreLife and pit\_in\_time\_tyre\_life into same dataframe

#pit\_stop\_df = pd.concat([pit\_in\_time\_tyre\_life, pit\_stop\_tyre\_life], axis=1)

#%%performing the poisson calculation

#https://timeseriesreasoning.com/contents/poisson-regression-model/

poisson\_results\_compound = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['CompoundCat'],

family=sm.families.Poisson()).fit()

poisson\_results\_driver\_number = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['DriverNumberCat'],

family=sm.families.Poisson()).fit()

poisson\_results\_tyre\_life = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['TyreLife'],

family=sm.families.Poisson()).fit()

poisson\_results\_fresh\_tyre = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

pd.get\_dummies(merge\_total\_final\_pit\_only['FreshTyre'], dtype = float),

family=sm.families.Poisson()).fit()

poisson\_results\_stint = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['Stint'],

family=sm.families.Poisson()).fit()

poisson\_results\_team = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['TeamCat'],

family=sm.families.Poisson()).fit()

poisson\_results\_driver = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['DriverCat'],

family=sm.families.Poisson()).fit()

poisson\_results\_track\_status = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['TrackStatusCat'],

family=sm.families.Poisson()).fit()

poisson\_results\_total\_time = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['TotalTime'],

family=sm.families.Poisson()).fit()

poisson\_results\_gap = sm.GLM(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['GapSeconds'],

family=sm.families.Poisson()).fit()

#viewing poisson results per variable

print(poisson\_results\_compound.summary())

print(poisson\_results\_tyre\_life.summary())

print(poisson\_results\_total\_time.summary())

print(poisson\_results\_stint.summary())

print(poisson\_results\_gap.summary())

print(poisson\_results\_fresh\_tyre.summary())

print(poisson\_results\_team.summary())

print(poisson\_results\_driver.summary())

print(poisson\_results\_driver\_number.summary())

print(poisson\_results\_track\_status.summary())

#further research https://bookdown.org/roback/bookdown-BeyondMLR/ch-poissonreg.html

#%% random forest

merge\_total\_final\_rf = merge\_total\_final\_pit\_only[[

'DriverNumberCat', 'LapNumber', 'CompoundCat', 'TyreLife', 'FreshTyre', 'Stint', 'TeamCat', 'Year',

'DriverCat', 'TrackStatusCat', 'GridPosition', 'Position', 'Points', 'TotalTime', 'GapSeconds']]

y = merge\_total\_final\_rf['LapNumber']

X = merge\_total\_final\_rf.drop(['LapNumber'], axis = 1)

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=44)

from sklearn.ensemble import RandomForestClassifier

rf\_model = RandomForestClassifier(n\_estimators=50, max\_features="auto", random\_state=44)

rf\_model.fit(X\_train, y\_train)

predictions = rf\_model.predict(X\_test)

predictions

importances = rf\_model.feature\_importances\_

columns = X.columns

i = 0

while i < len(columns):

print(f" The importance of feature ' {columns[i]}' is {round(importances[i] \* 100, 3)}%.")

i += 1

accuracy\_score(y\_test, predictions)

confusion\_matrix(y\_test, predictions)

#%%#calculate mean/variance of poisson\_results

merge\_total\_final\_pit\_only['LapNumber'].var() #46.567

merge\_total\_final\_pit\_only['LapNumber'].mean() #14.833

#visualize poisson to justify model choice

fig = merge\_total\_final\_pit\_only['LapNumber'].value\_counts(sort=False).plot(kind='barh')

#group dataset by tyre type and get mean and variance to test for heterogeneity

#if variance > mean in group, look at negative binomial

merge\_total\_final\_pit\_only['CompoundCat'].mean() #0.479

merge\_total\_final\_pit\_only['CompoundCat'].var() #0.297

#%%#frequency graph compound tyres

#sort numerically

merge\_total\_final\_pit\_only['CompoundCat'].value\_counts(sort=True).plot(kind='bar')

#%% Building Consul’s Generalized Poison regression model, know as GP-1:

# =============================================================================

# gen\_poisson\_gp1\_cumulative = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['Cumulative'],

# p=1)

# gen\_poisson\_gp1\_cumulative\_results = gen\_poisson\_gp1\_cumulative.fit()

# print(gen\_poisson\_gp1\_cumulative\_results.summary())

#

# gen\_poisson\_gp1\_cumulative\_lap = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['CumulativeLap'],

# p=1)

# gen\_poisson\_gp1\_cumulative\_lap\_results = gen\_poisson\_gp1\_cumulative\_lap.fit()

# print(gen\_poisson\_gp1\_cumulative\_lap\_results.summary())

#

# gen\_poisson\_gp1\_cumulative\_pit = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['CumulativePit'],

# p=1)

# gen\_poisson\_gp1\_cumulative\_pit\_results = gen\_poisson\_gp1\_cumulative\_pit.fit()

# print(gen\_poisson\_gp1\_cumulative\_pit\_results.summary())

#

# gen\_poisson\_gp1\_gap = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['Gap'],

# p=1)

# gen\_poisson\_gp1\_gap\_results = gen\_poisson\_gp1\_gap.fit()

# print(gen\_poisson\_gp1\_gap\_results.summary())

#

# gen\_poisson\_gp1\_track\_status = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['TrackStatusCat'],

# p=1)

# gen\_poisson\_gp1\_track\_status\_results = gen\_poisson\_gp1\_gap.fit()

# print(gen\_poisson\_gp1\_track\_status\_results.summary())

#

# gen\_poisson\_gp1\_compound = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['CompoundCat'],

# p=1)

# gen\_poisson\_gp1\_compound\_results = gen\_poisson\_gp1\_compound.fit()

# print(gen\_poisson\_gp1\_compound\_results.summary())

#

# gen\_poisson\_gp1\_pit\_time = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['PitTime'],

# p=1)

# gen\_poisson\_gp1\_pit\_time\_results = gen\_poisson\_gp1\_pit\_time.fit()

# print(gen\_poisson\_gp1\_pit\_time\_results.summary())

#

# gen\_poisson\_gp1\_pit\_out = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['PitOutTimeSeconds'],

# p=1)

# gen\_poisson\_gp1\_pit\_out\_results = gen\_poisson\_gp1\_pit\_out.fit()

# print(gen\_poisson\_gp1\_pit\_out\_results.summary())

#

# gen\_poisson\_gp1\_pit\_in = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'], ###################

# merge\_total\_final\_pit\_only['PitInTimeSeconds'],

# p=1)

# gen\_poisson\_gp1\_pit\_in\_results = gen\_poisson\_gp1\_pit\_in.fit()

# print(gen\_poisson\_gp1\_pit\_in\_results.summary())

#

# =============================================================================

gen\_poisson\_gp1\_stint = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['Stint'],

p=1)

gen\_poisson\_gp1\_stint\_results = gen\_poisson\_gp1\_stint.fit()

print(gen\_poisson\_gp1\_stint\_results.summary())

gen\_poisson\_gp1\_fresh\_tyre = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'],

pd.get\_dummies(merge\_total\_final\_pit\_only['FreshTyre'], dtype = float), p=1)

gen\_poisson\_gp1\_fresh\_tyre\_results = gen\_poisson\_gp1\_fresh\_tyre.fit()

print(gen\_poisson\_gp1\_fresh\_tyre\_results.summary())

gen\_poisson\_gp1\_tyre\_life = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['TyreLife'],

p=1)

gen\_poisson\_gp1\_tyre\_life\_results = gen\_poisson\_gp1\_tyre\_life.fit()

print(gen\_poisson\_gp1\_tyre\_life\_results.summary())

gen\_poisson\_gp1\_team = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['TeamCat'],

p=1)

gen\_poisson\_gp1\_team\_results = gen\_poisson\_gp1\_team.fit()

print(gen\_poisson\_gp1\_team\_results.summary())

gen\_poisson\_gp1\_driver = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['DriverCat'],

p=1)

gen\_poisson\_gp1\_driver\_results = gen\_poisson\_gp1\_driver.fit()

print(gen\_poisson\_gp1\_driver\_results.summary())

gen\_poisson\_gp1\_driver\_number = sm.GeneralizedPoisson(merge\_total\_final\_pit\_only['LapNumber'],

merge\_total\_final\_pit\_only['DriverNumberCat'],

p=1)

gen\_poisson\_gp1\_driver\_number\_results = gen\_poisson\_gp1\_driver\_number.fit()

print(gen\_poisson\_gp1\_driver\_number\_results.summary())

#%% negative binomial time, holla

#https://timeseriesreasoning.com/contents/negative-binomial-regression-model/

merge\_total\_final\_pit\_only['FreshTyre'] = merge\_total\_final\_pit\_only['FreshTyre'].astype('category')

merge\_total\_final\_pit\_only["FreshTyreCat"] = merge\_total\_final\_pit\_only["FreshTyre"].cat.codes

#create the training and testing data sets

mask = np.random.rand(len(merge\_total\_final\_pit\_only)) < 0.8

merge\_total\_final\_pit\_only\_train = merge\_total\_final\_pit\_only[mask]

merge\_total\_final\_pit\_only\_test = merge\_total\_final\_pit\_only[~mask]

#Setup the regression expression in patsy notation. We are telling patsy that BB\_COUNT is our dependent variable and it depends on the regression variables: DAY, DAY\_OF\_WEEK, MONTH, HIGH\_T, LOW\_T and PRECIP

expr = """LapNumber ~ DriverNumberCat + CompoundCat + TyreLife + FreshTyreCat+

Stint + GapSeconds + TeamCat + DriverCat + TrackStatusCat + Points + TotalTime"""

#Set up the X and y matrices for the training and testing data sets

y\_train, X\_train = dmatrices(expr, merge\_total\_final\_pit\_only\_train, return\_type='dataframe')

y\_test, X\_test = dmatrices(expr, merge\_total\_final\_pit\_only\_test, return\_type='dataframe')

poisson\_training\_results = sm.GLM(y\_train, X\_train, family=sm.families.Poisson()).fit()

print(poisson\_training\_results.summary())

#Add the λ vector as a new column called 'BB\_LAMBDA' to the Data Frame of the training data set

merge\_total\_final\_pit\_only\_train['LapNumber'] = poisson\_training\_results.mu

print(poisson\_training\_results.mu)

print(len(poisson\_training\_results.mu))

#add a derived column called 'AUX\_OLS\_DEP' to the pandas Data Frame. This new column will store the values of the dependent variable of the OLS regression

merge\_total\_final\_pit\_only\_train['AUX\_OLS\_DEP'] = merge\_total\_final\_pit\_only\_train.apply(lambda x: ((x['LapNumber'] - x['LapNumber'])\*\*2 - x['LapNumber']) / x['LapNumber'], axis=1)

#use patsy to form the model specification for the OLSR

ols\_expr = """AUX\_OLS\_DEP ~ LapNumber - 1"""

#Configure and fit the OLSR model

aux\_olsr\_results = smf.ols(ols\_expr, merge\_total\_final\_pit\_only\_train).fit()

print(aux\_olsr\_results.params) #-.06 (why negative, should be positive?)

aux\_olsr\_results.tvalues #test significance of alpha -14.670

#using t-value calculator (https://goodcalculators.com/student-t-value-calculator/))

#positive .06 with 36 degrees of freedom (from len of poisson training.mu) = 2-tailed 1.94 and right tailed 1.59

#should 35 be degrees of freedom (len - 1) or should we use total dataset (47 - 1?)

#either way, t-values are above 1.9 and 1.5

#train the NB2 model on the training data set

nb2\_training\_results = sm.GLM(y\_train, X\_train,family=sm.families.NegativeBinomial(alpha=-aux\_olsr\_results.params[0])).fit()

print(nb2\_training\_results.summary())

#make some predictions using our trained NB2 model

nb2\_predictions = nb2\_training\_results.get\_prediction(X\_test)

predictions\_summary\_frame = nb2\_predictions.summary\_frame()

print(predictions\_summary\_frame)

#plot the predicted counts versus the actual counts for the test data

predicted\_counts=predictions\_summary\_frame['mean']

actual\_counts = y\_test['LapNumber']

fig = plt.figure()

fig.suptitle('Predicted versus actual lap number when pit stop made')

predicted, = plt.plot(X\_test.index, predicted\_counts, 'go-', label='Predicted counts')

actual, = plt.plot(X\_test.index, actual\_counts, 'ro-', label='Actual counts')

plt.legend(handles=[predicted, actual])

plt.show()

#%%#Build Famoye's Restricted Generalized Poison regression model, know as GP-2

# =============================================================================

# gen\_poisson\_gp2 = sm.GeneralizedPoisson(y\_train, X\_train, p=2)

#

# #Fit the model

# gen\_poisson\_gp2\_results = gen\_poisson\_gp2.fit()

#

# #print the results

# print(gen\_poisson\_gp2\_results.summary())

# =============================================================================

#

#%%