24 hours of le mans 2023

August 2, 2023

```
The 24 Hours of Le Mans is a prestigious endurance race in Le Mans, France,
lasting 24 hours non-stop, where teams compete in various classes.
It's one of the most challenging motorsport events globally.
The team that covers the greatest distance in 24 hours emerges as the winner.
```

0.0.1 Web Scrapping using python

- Dataset_source:

https://www.motorsport.com/lemans/results/2023/24-hours-of-le-mans-629901/?st=RACE

Questions:

- How many teams participated in the Le Mans 2023 event?
- How many cars are there in each class?
- What is the average number of laps for each class of car?
- What are the top 3 cars for each class?
- What is the performance comparison between the Chevrolet Corvette C8.R and the hypercar class?(based on the "labs" parameter)

Task:

- Scraping data using the Gazpacho library
- Create a function to scrape team names
- Create dataframe
- Create visualizations using matplotlib.pyplot

Scraping data

[71]:
```
# install library
!pip install gazpacho
import pandas as pd
import numpy as np
```

```
Requirement already satisfied: gazpacho in c:\users\avs_ktb\anaconda3\lib\site-
packages (1.1)
```

[72]:
```
# import gazpacho
from gazpacho import Soup
from requests import get
```

[73]:
```
url = 'https://www.motorsport.com/lemans/results/2023/
 ↪24-hours-of-le-mans-629901/?st=RACE'
response = get(url)
response.status_code
```

[73]: 200

[74]:
```
# find car name
lm = Soup(response.text)
lm.find('td', {'class': 'ms-table_cell ms-table_field--car_model'},␣
 ↪mode='first').strip()
```

[74]: 'Ferrari 499P'

[75]:
```
# using for loop to find the names of cars
cars = lm.find('td', {'class': 'ms-table_cell ms-table_field--car_model'})
cars = [car.strip() for car in cars]
print(cars)
print(len(cars))
```

```
['Ferrari 499P', 'Toyota GR010 - Hybrid', 'Cadillac V-Series.R', 'Cadillac
V-Series.R', 'Ferrari 499P', 'Glickenhaus 007 LMH', 'Glickenhaus 007 LMH',
'Peugeot 9X8', 'Porsche 963', 'Oreca 07', 'Oreca 07', 'Oreca 07', 'Oreca 07',
'Oreca 07', 'Oreca 07', 'Oreca 07', 'Cadillac V-Series.R', 'Oreca 07', 'Oreca
07', 'Oreca 07', 'Oreca 07', 'Porsche 963', 'Oreca 07', 'Oreca 07', 'Oreca 07',
'Chevrolet Corvette C8.R', 'Peugeot 9X8', 'Aston Martin Vantage AMR', 'Porsche
911 RSR - 19', 'Porsche 911 RSR - 19', 'Ferrari 488 GTE EVO', 'Oreca 07', 'Aston
Martin Vantage AMR', 'Oreca 07', 'Porsche 911 RSR - 19', 'Ferrari 488 GTE EVO',
'Oreca 07', 'Ferrari 488 GTE EVO', 'Chevrolet Camaro ZL1', 'Porsche 963',
'Ferrari 488 GTE EVO', 'Porsche 911 RSR - 19', 'Oreca 07', 'Porsche 911 RSR -
19', 'Vanwall Vandervell 680', 'Aston Martin Vantage AMR', 'Oreca 07', 'Porsche
911 RSR - 19', 'Oreca 07', 'Oreca 07', 'Toyota GR010 - Hybrid', 'Ferrari 488 GTE
EVO', 'Oreca 07', 'Porsche 963', 'Aston Martin Vantage AMR', 'Ferrari 488 GTE
EVO', 'Porsche 911 RSR - 19', 'Porsche 911 RSR - 19', 'Aston Martin Vantage
AMR', 'Ferrari 488 GTE EVO', 'Oreca 07', 'Oreca 07']
62
```

[76]:
```
# using for loop to find labs
labs = lm.find('td', {'class': 'ms-table_cell ms-table_field--laps'})
labs = [lab.strip() for lab in labs]
print(labs)
print(len(labs))
```

```
['342', '342', '341', '340', '337', '335', '333', '330', '329', '328', '328',
'327', '327', '327', '327', '325', '324', '323', '322', '322', '321', '320',
'317', '316', '316', '313', '312', '312', '312', '312', '312', '311', '310',
'310', '309', '307', '303', '303', '285', '244', '254', '246', '183', '170',
'165', '163', '158', '118', '117', '113', '103', '89', '87', '84', '58', '33',
```

```
'28', '28', '21', '21', '19', '18']
62
```

[77]:
```
# using for loop to find class of each car
classes = lm.find('span', {'class':'class'})
classes = [x.strip() for x in classes ]
print(classes)
print(len(classes))
```

```
['HYPERCAR', 'HYPERCAR', 'HYPERCAR', 'HYPERCAR', 'HYPERCAR', 'HYPERCAR',
'HYPERCAR', 'HYPERCAR', 'HYPERCAR', 'HYPERCAR', 'LMP2', 'LMP2', 'LMP2', 'LMP2', 'LMP2',
'LMP2', 'LMP2', 'HYPERCAR', 'LMP2', 'LMP2', 'LMP2', 'LMP2', 'HYPERCAR', 'LMP2',
'LMP2', 'LMP2', 'LMGTE AM', 'HYPERCAR', 'LMGTE AM', 'LMGTE AM', 'LMGTE AM',
'LMGTE AM', 'LMP2', 'LMGTE AM', 'LMP2', 'LMGTE AM', 'LMGTE AM', 'LMP2', 'LMGTE
AM', 'INNOVATIVE CAR', 'HYPERCAR', 'LMGTE AM', 'LMGTE AM', 'LMP2', 'LMGTE AM',
'HYPERCAR', 'LMGTE AM', 'LMP2', 'LMGTE AM', 'LMP2', 'LMP2', 'HYPERCAR', 'LMGTE
AM', 'LMP2', 'HYPERCAR', 'LMGTE AM', 'LMGTE AM', 'LMGTE AM', 'LMGTE AM', 'LMGTE
AM', 'LMGTE AM', 'LMP2', 'LMP2']
62
```

[78]:
```
# define fuction to find team name
def transform(soup):
    # Find all div elements with class 'info'
    divs_info = soup.find('div', {'class': 'info'})

    # Initialize a list to store the team names
    team_names = []

    # Loop through each div with class 'info'
    for div_info in divs_info:
        # Find the span element with class 'name' within the div
        span_name = div_info.find('span', {'class': 'name'})

        # Check if the span element is found
        if span_name is not None:
            team_name = span_name.text.strip()
            team_names.append(team_name)

    return team_names

# URL of the web page you want to scrape
url = 'https://www.motorsport.com/lemans/results/2023/
 ↪24-hours-of-le-mans-629901/?st=RACE'

# Send an HTTP request to the URL and get the HTML content
response = get(url)
html_content = response.text
```

```
# Parse the HTML content using gazpacho Soup
soup = Soup(html_content)

# Call the transform function with the soup object
team = transform(soup)

# Print the extracted team names
print(team)
print(len(team))
```

```
['FERRARI AF CORSE', 'Toyota Racing', 'CADILLAC RACING', 'CADILLAC RACING',
'FERRARI AF CORSE', 'GLICKENHAUS RACING', 'GLICKENHAUS RACING', 'PEUGEOT
TOTALENERGIES', 'PORSCHE PENSKE MOTORSPORT', 'INTER EUROPOL COMPETITION', 'TEAM
WRT', 'DUQUEINE TEAM', 'ALPINE ELF TEAM', 'TEAM WRT', 'IDEC SPORT', 'VECTOR
SPORT', 'Action Express Racing', 'United Autosports', 'ALPINE ELF TEAM',
'Algarve Pro Racing Team', 'United Autosports', 'PORSCHE PENSKE MOTORSPORT',
'Cool Racing', 'Jota Sport', 'PANIS RACING', 'Corvette Racing', 'N. Varrone',
'PEUGEOT TOTALENERGIES', 'ORT BY TF', 'GR RACING', 'IRON DAMES', 'AF Corse',
'DKR Engineering', 'NORTHWEST AMR', 'PREMA RACING', 'PROJECT 1 - AO',
'Walkenhorst Motorsport', 'Graff Racing', 'Kessel Racing', 'Hendrick
Motorsports', 'HERTZ TEAM JOTA', 'Kessel Racing', 'Proton Competition', 'AF
Corse', 'Proton Competition', 'FLOYD VANWALL RACING TEAM', "D'Station Racing",
'Cool Racing', 'Proton Competition', 'M. Pedersen', 'INTER EUROPOL COMPETITION',
'PREMA RACING', 'Toyota Racing', 'JMW Motorsport', 'Racing Team Turkey',
'PORSCHE PENSKE MOTORSPORT', 'TF Sport', 'A. Robin', 'M. Robin', 'RICHARD MILLE
AF CORSE', 'IRON LYNX', 'Proton Competition', 'GMB Motorsport', 'AF Corse',
'Tower Motorsports', 'NIELSEN RACING']
66
```

[79]:
```
# drop some item in list
items_to_drop = ['N. Varrone', 'M. Pedersen','A. Robin','M. Robin']

# Using list comprehension to create a new list without the items to drop
team = [item for item in team if item not in items_to_drop]
print(team)
print(len(team))
```

```
['FERRARI AF CORSE', 'Toyota Racing', 'CADILLAC RACING', 'CADILLAC RACING',
'FERRARI AF CORSE', 'GLICKENHAUS RACING', 'GLICKENHAUS RACING', 'PEUGEOT
TOTALENERGIES', 'PORSCHE PENSKE MOTORSPORT', 'INTER EUROPOL COMPETITION', 'TEAM
WRT', 'DUQUEINE TEAM', 'ALPINE ELF TEAM', 'TEAM WRT', 'IDEC SPORT', 'VECTOR
SPORT', 'Action Express Racing', 'United Autosports', 'ALPINE ELF TEAM',
'Algarve Pro Racing Team', 'United Autosports', 'PORSCHE PENSKE MOTORSPORT',
'Cool Racing', 'Jota Sport', 'PANIS RACING', 'Corvette Racing', 'PEUGEOT
TOTALENERGIES', 'ORT BY TF', 'GR RACING', 'IRON DAMES', 'AF Corse', 'DKR
Engineering', 'NORTHWEST AMR', 'PREMA RACING', 'PROJECT 1 - AO', 'Walkenhorst
Motorsport', 'Graff Racing', 'Kessel Racing', 'Hendrick Motorsports', 'HERTZ
TEAM JOTA', 'Kessel Racing', 'Proton Competition', 'AF Corse', 'Proton
```

```
Competition', 'FLOYD VANWALL RACING TEAM', "D'Station Racing", 'Cool Racing',
'Proton Competition', 'INTER EUROPOL COMPETITION', 'PREMA RACING', 'Toyota
Racing', 'JMW Motorsport', 'Racing Team Turkey', 'PORSCHE PENSKE MOTORSPORT',
'TF Sport', 'RICHARD MILLE AF CORSE', 'IRON LYNX', 'Proton Competition', 'GMB
Motorsport', 'AF Corse', 'Tower Motorsports', 'NIELSEN RACING']
62
```

Create dataframe

[80]:
```
# creating a DataFrame from the data that was scraped earlier.
import pandas as pd
df = pd.DataFrame({
    'team': team,
    'car': cars,
    'class_of_car': classes,
    'labs': labs
})

# preview dataframe
df.head(10)
```

[80]:

	team	car	class_of_car	labs
0	FERRARI AF CORSE	Ferrari 499P	HYPERCAR	342
1	Toyota Racing	Toyota GR010 - Hybrid	HYPERCAR	342
2	CADILLAC RACING	Cadillac V-Series.R	HYPERCAR	341
3	CADILLAC RACING	Cadillac V-Series.R	HYPERCAR	340
4	FERRARI AF CORSE	Ferrari 499P	HYPERCAR	337
5	GLICKENHAUS RACING	Glickenhaus 007 LMH	HYPERCAR	335
6	GLICKENHAUS RACING	Glickenhaus 007 LMH	HYPERCAR	333
7	PEUGEOT TOTALENERGIES	Peugeot 9X8	HYPERCAR	330
8	PORSCHE PENSKE MOTORSPORT	Porsche 963	HYPERCAR	329
9	INTER EUROPOL COMPETITION	Oreca 07	LMP2	328

[81]:
```
# cheack missing values
df.isna().sum()
```

[81]:
```
team           0
car            0
class_of_car   0
labs           0
dtype: int64
```

[82]:
```
# check data type
df.dtypes
```

[82]:
```
team           object
car            object
```

```
class_of_car    object
labs            object
dtype: object
```

[83]:
```
# change data type
df['labs'] = df['labs'].astype(float)
df.dtypes
```

[83]:
```
team             object
car              object
class_of_car     object
labs            float64
dtype: object
```

Question 1: How many teams participated in the Le Mans 2023 event?

[84]:
```
total_teams = df['team'].nunique()
print("Total number of teams that have competed in Le Mans 2023:", total_teams)
```

```
Total number of teams that have competed in Le Mans 2023: 43
```

[85]:
```
# Aggregating the count by cars
df['team'].value_counts()
```

[85]:
```
Proton Competition           4
AF Corse                     3
PORSCHE PENSKE MOTORSPORT    3
FERRARI AF CORSE             2
ALPINE ELF TEAM              2
PREMA RACING                 2
Toyota Racing                2
Cool Racing                  2
United Autosports            2
Kessel Racing                2
CADILLAC RACING              2
GLICKENHAUS RACING           2
TEAM WRT                     2
PEUGEOT TOTALENERGIES        2
INTER EUROPOL COMPETITION    2
FLOYD VANWALL RACING TEAM    1
Hendrick Motorsports         1
HERTZ TEAM JOTA              1
Tower Motorsports            1
GMB Motorsport               1
JMW Motorsport               1
Racing Team Turkey           1
TF Sport                     1
```

```
Graff Racing                 1
RICHARD MILLE AF CORSE       1
IRON LYNX                    1
D'Station Racing             1
IRON DAMES                   1
Walkenhorst Motorsport       1
PROJECT 1 - AO               1
NORTHWEST AMR                1
DKR Engineering              1
GR RACING                    1
ORT BY TF                    1
Corvette Racing              1
PANIS RACING                 1
Jota Sport                   1
Algarve Pro Racing Team      1
Action Express Racing        1
VECTOR SPORT                 1
IDEC SPORT                   1
DUQUEINE TEAM                1
NIELSEN RACING               1
Name: team, dtype: int64
```

Question 2: How many cars are there in each class?

[101]:
```
df.groupby('class_of_car')['car'].count().sort_values(ascending = False).
  ↪reset_index()
```

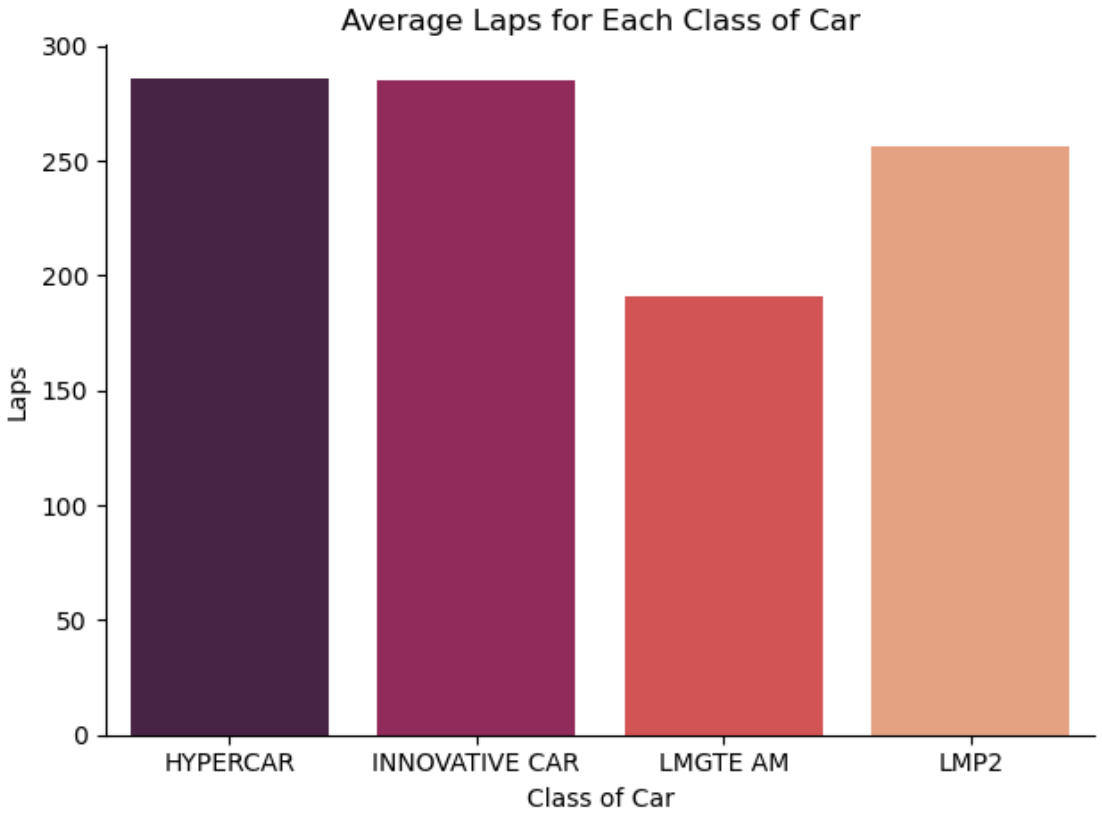
[101]:
```
      class_of_car  car
0             LMP2   24
1         LMGTE AM   21
2         HYPERCAR   16
3  INNOVATIVE CAR     1
```

Question 3: What is the average number of laps for each class of car?

[86]:
```
# find average of labs
average_lab = df.groupby('class_of_car')['labs'].mean().round().reset_index()
average_lab
```

[86]:
```
      class_of_car   labs
0         HYPERCAR  286.0
1  INNOVATIVE CAR  285.0
2         LMGTE AM  191.0
3             LMP2  256.0
```

[87]:
```
import seaborn as sns
import matplotlib.pyplot as plt


sns.barplot(x='class_of_car', y='labs', data=average_lab, palette='rocket')
plt.xlabel("Class of Car")
plt.ylabel("Laps")
plt.title("Average Laps for Each Class of Car")
#plt.xticks(rotation=45, ha='right')
plt.tight_layout()

sns.despine()  # Remove the top and right spines

plt.show()
```



Question 4: What are the top 3 cars for each class?

[88]:
```
# filter HYPERCAR class
df_hyper = df.query("class_of_car == 'HYPERCAR'")[['team','car','labs']].head(3)
df_hyper
```

[88]:
```
               team                     car   labs
0  FERRARI AF CORSE           Ferrari 499P  342.0
1     Toyota Racing  Toyota GR010 - Hybrid  342.0
2   CADILLAC RACING    Cadillac V-Series.R  341.0
```

[89]:
```
# filter LMP2 class
df_LMP2 = df.query("class_of_car == 'LMP2'")[['team','car','labs']].head(3)
df_LMP2
```

[89]:
```
                         team       car   labs
9   INTER EUROPOL COMPETITION  Oreca 07  328.0
10                   TEAM WRT  Oreca 07  328.0
11              DUQUEINE TEAM  Oreca 07  327.0
```

[90]:
```
# filter LMGTE class
df_LMGTE = df.query("class_of_car == 'LMGTE AM'")[['team','car','labs']].head(3)
df_LMGTE
```

[90]:
```
               team                        car   labs
25  Corvette Racing   Chevrolet Corvette C8.R  313.0
27        ORT BY TF  Aston Martin Vantage AMR  312.0
28        GR RACING      Porsche 911 RSR - 19  312.0
```

Question 5: What is the performance comparison between the Chevrolet Corvette C8.R and the hypercar class?(based on the "labs" parameter)

[91]:
```
result = df[(df["car"] == "Chevrolet Corvette C8.R") | (df["class_of_car"] ==␣
 ↪"HYPERCAR")].reset_index()
result
```

[91]:
```
    index                      team                     car class_of_car  \
0       0          FERRARI AF CORSE           Ferrari 499P     HYPERCAR
1       1             Toyota Racing  Toyota GR010 - Hybrid     HYPERCAR
2       2           CADILLAC RACING    Cadillac V-Series.R     HYPERCAR
3       3           CADILLAC RACING    Cadillac V-Series.R     HYPERCAR
4       4          FERRARI AF CORSE           Ferrari 499P     HYPERCAR
5       5       GLICKENHAUS RACING    Glickenhaus 007 LMH     HYPERCAR
6       6       GLICKENHAUS RACING    Glickenhaus 007 LMH     HYPERCAR
7       7    PEUGEOT TOTALENERGIES            Peugeot 9X8     HYPERCAR
8       8  PORSCHE PENSKE MOTORSPORT            Porsche 963     HYPERCAR
9      16    Action Express Racing    Cadillac V-Series.R     HYPERCAR
10     21  PORSCHE PENSKE MOTORSPORT            Porsche 963     HYPERCAR
11     25          Corvette Racing  Chevrolet Corvette C8.R     LMGTE AM
12     26    PEUGEOT TOTALENERGIES            Peugeot 9X8     HYPERCAR
13     39          HERTZ TEAM JOTA            Porsche 963     HYPERCAR
14     44  FLOYD VANWALL RACING TEAM  Vanwall Vandervell 680     HYPERCAR
15     50             Toyota Racing  Toyota GR010 - Hybrid     HYPERCAR
```

```
16     53  PORSCHE PENSKE MOTORSPORT              Porsche 963     HYPERCAR
```

```
     labs
0   342.0
1   342.0
2   341.0
3   340.0
4   337.0
5   335.0
6   333.0
7   330.0
8   329.0
9   324.0
10  320.0
11  313.0
12  312.0
13  244.0
14  165.0
15  103.0
16   84.0
```

[92]:
```
result2 = result.groupby('class_of_car')['labs'].mean().reset_index()
result2
```

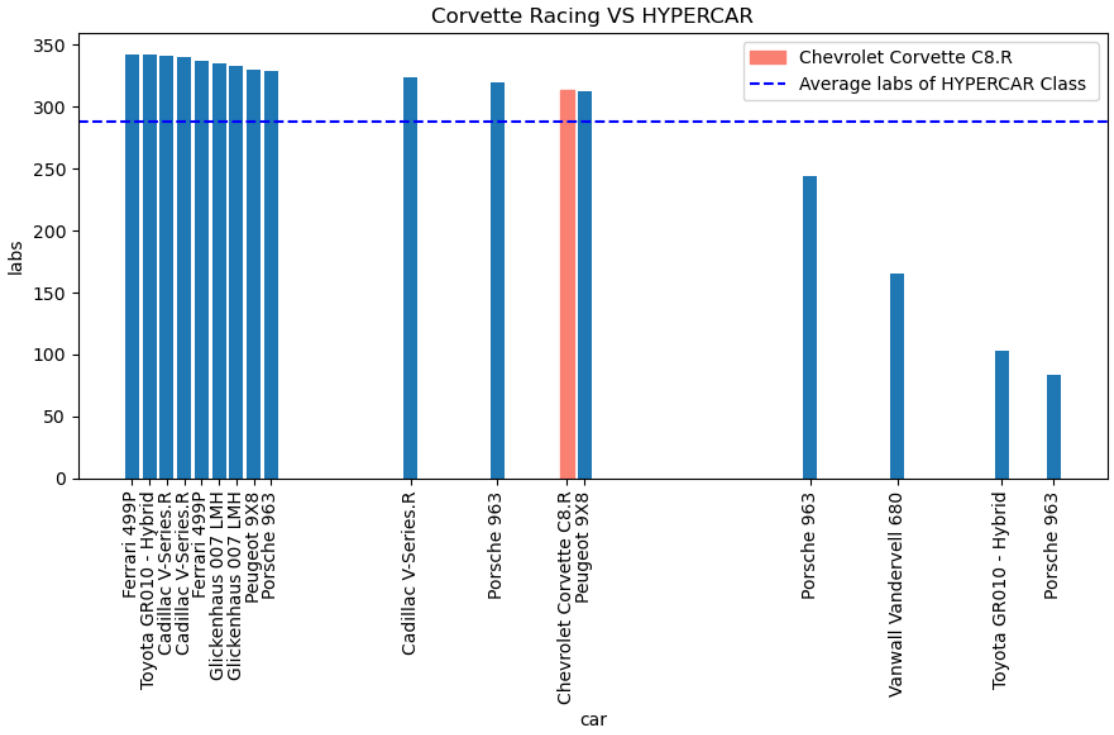
[92]:
```
  class_of_car      labs
0     HYPERCAR  286.3125
1     LMGTE AM  313.0000
```

[93]:
```
print(f"The average labs of 'HYPERCAR Class' is {result2['labs'][0]} labs.\n"
      f"The average labs of 'Corvette Racing' is {result2['labs'][1]} labs.")
```

```
The average labs of 'HYPERCAR Class' is 286.3125 labs.
The average labs of 'Corvette Racing' is 313.0 labs.
```

[94]:
```
# create bar chart
plt.figure(figsize=(9, 6))  # Adjust the width and height as per your preference

bar_width = 0.8  # Adjust the width of each bar (increase/decrease this value␣
 ↪as needed)

bars = plt.bar(result['index'], result['labs'], width=bar_width)  # Use the␣
 ↪'width' parameter

# Use plt.xticks() to set custom labels for each bar
plt.xticks(result['index'], result['car'], rotation=90)

plt.ylabel("labs")
plt.xlabel("car")
```

```
plt.title('Corvette Racing VS HYPERCAR')
bars[11].set_color('salmon')

# Calculate the mean of y-axis values (excluding the value of bars[11])
mean_labs = result[result['index'] != 11]['labs'].mean()

# Plot the mean as a line curve
plt.axhline(mean_labs, color='blue', linestyle='--')
plt.gca().yaxis.set_major_formatter('{:,.0f}'.format)  # y axis number format

# Add a custom legend for the bar in position 11 and the mean line curve
plt.legend(handles=[bars[11], plt.Line2D([], [], color='blue', linestyle='--')],
           labels=['Chevrolet Corvette C8.R', 'Average labs of HYPERCAR Class
 ↪'], loc='upper right')

plt.tight_layout()  # Automatically adjusts subplot parameters for better
 ↪spacing

plt.show()
```



Summmary

1.How many teams participated in the Le Mans 2023 event?

- 43

2.How many cars are there in each class?
- LMP2: 24
- LMGTE AM: 21
- HYPERCAR: 16
- INNOVATIVE CAR: 1

3.What is the average number of laps for each class of car?
- HYPERCAR: 286
- INNOVATIVE CAR: 285
- LMGTE AM: 191
- LMP2: 256

4.What are the top 3 cars for each class?
- HYPERCAR: 1)Ferrari 499P = 342 labs
  2)Toyota GR010 Hybrid = 342 labs
  3)Cadillac V-Series.R = 341 labs

- LMP2: 1)Oreca 07 = 328 labs
  2)Oreca 07 = 328 labs
  3)Oreca 07 = 327 labs

- LMGTE AM: 1)Chevrolet Corvette C8.R = 313 labs
  2)Aston Martin Vantage AMR = 312 labs
  3)Porsche 911 RSR - 19 = 312 labs

5.What is the performance comparison between the Chevrolet Corvette C8.R and the hypercar class?
- HYPERCAR(average labs) 286 labs
- Chevrolet Corvette C8.R 313 labs