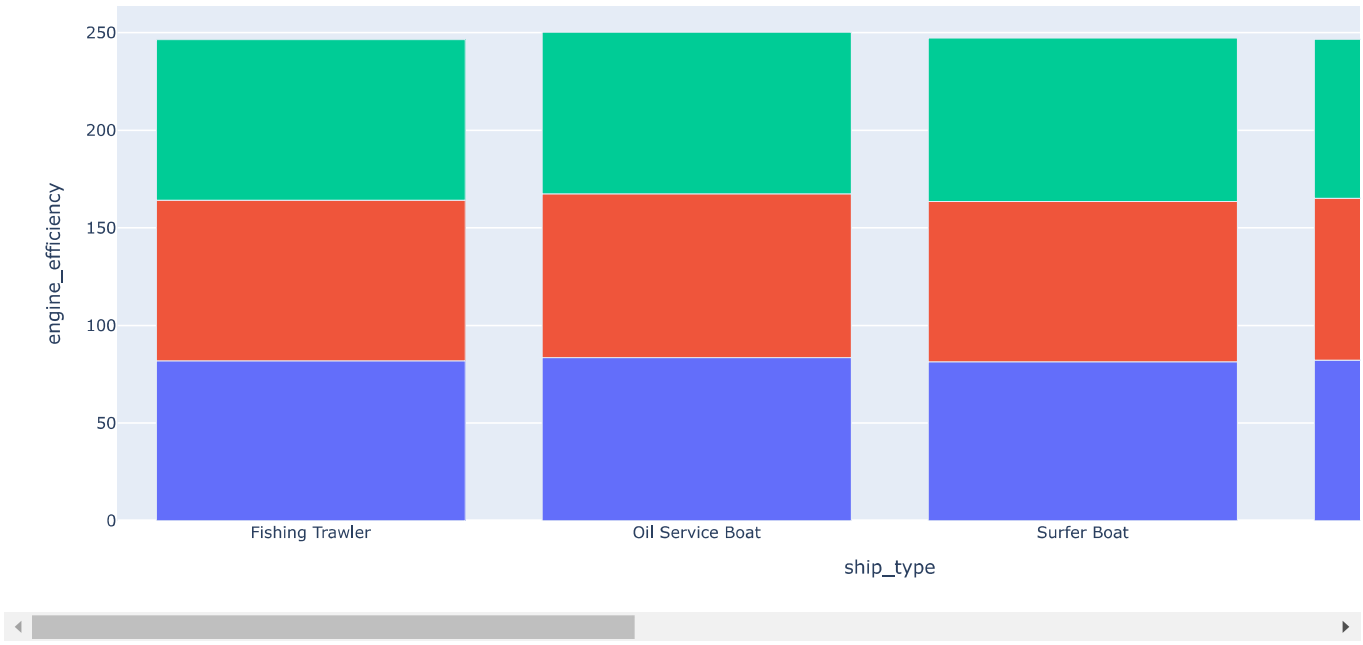
```
# MCQ's

# Answers :
# 1) A
# 2) B
# 3) C
# 4) B
# 5) A
# 6) D
# 7) C
# 8) B
# 9) D
# 10) A,C and D
# 11) A and D
# 12) B
# 13) B
# 14) A
# 15) C
# 16) C
# 17) C
# 18) B
# 19) C
# 20) C



# Q.1)
import pandas as pd
import seaborn as sns
import plotly.express as px
import plotly
ship=pd.read_csv("/content/ship_fuel_efficiency.csv")


ship.head(5)
```

	ship_id	ship_type	route_id	month	distance	fuel_type	fuel_consumption	CO2_emissions	weather_conditions	engine_efficienc
0	NG001	Oil Service Boat	Warri-Bonny	January	132.26	HFO	3779.77	10625.76	Stormy	92.1
1	NG001	Oil Service Boat	Port Harcourt-Lagos	February	128.52	HFO	4461.44	12779.73	Moderate	92.9
2	NG001	Oil Service	Port	March	67.20	HFO	1967.73	5358.01	Calm	87.0

```
ship1=ship.groupby(['ship_type','weather_conditions'])['engine_efficiency'].mean().reset_index()
ship1
```

	ship_type	weather_conditions	engine_efficiency
0	Fishing Trawler	Calm	81.936373
1	Fishing Trawler	Moderate	82.262745
2	Fishing Trawler	Stormy	82.220625
3	Oil Service Boat	Calm	83.531635
4	Oil Service Boat	Moderate	83.858810
5	Oil Service Boat	Stormy	82.824715
6	Surfer Boat	Calm	81.435446
7	Surfer Boat	Moderate	82.098000
8	Surfer Boat	Stormy	83.684159
9	Tanker Ship	Calm	82.265195
10	Tanker Ship	Moderate	82.894597
11	Tanker Ship	Stormy	81.406923

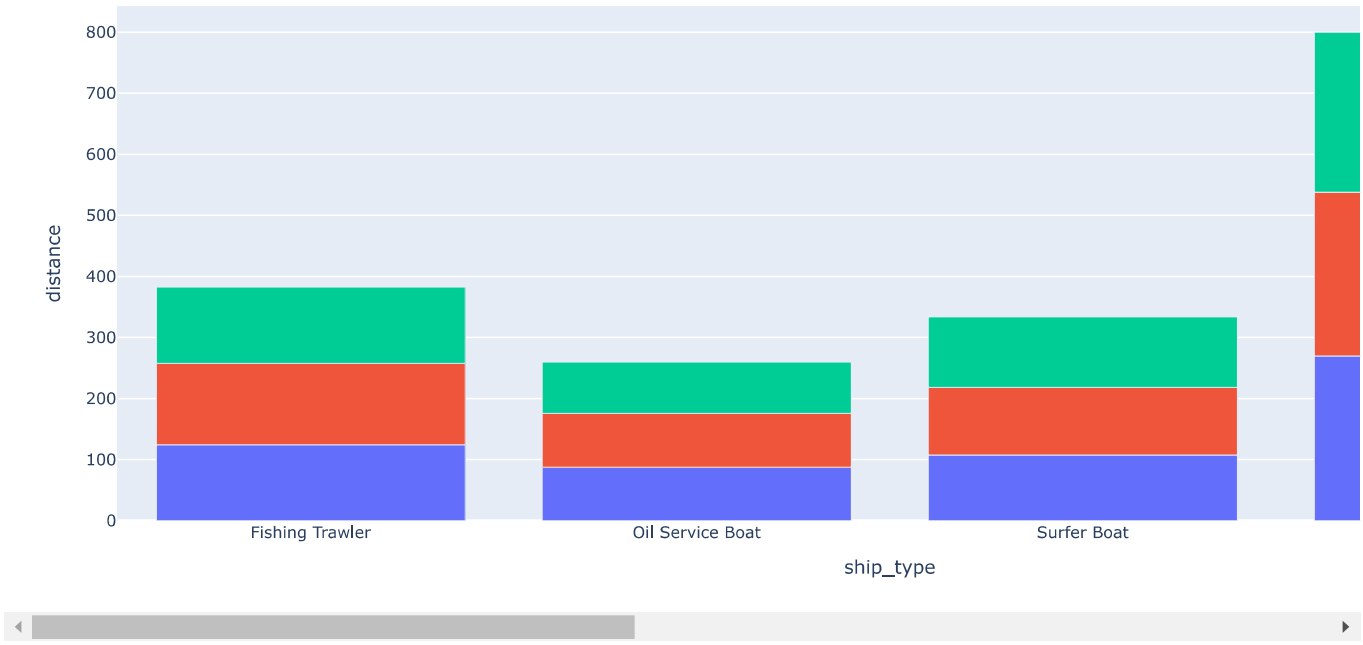
```
px.bar(ship1,x='ship_type',y='engine_efficiency',color='weather_conditions')
```




```
ship2=ship.groupby(['ship_type','weather_conditions'])['distance'].mean().reset_index()
ship2
```

	ship_type	weather_conditions	distance
0	Fishing Trawler	Calm	124.128627
1	Fishing Trawler	Moderate	133.637549
2	Fishing Trawler	Stormy	125.008229
3	Oil Service Boat	Calm	87.397925
4	Oil Service Boat	Moderate	88.427619
5	Oil Service Boat	Stormy	83.673415
6	Surfer Boat	Calm	107.604455
7	Surfer Boat	Moderate	110.391455
8	Surfer Boat	Stormy	115.502832
9	Tanker Ship	Calm	269.752857
10	Tanker Ship	Moderate	268.255403
11	Tanker Ship	Stormy	261.795154

```
px.bar(ship2,x='ship_type',y='distance',color='weather_conditions')
```



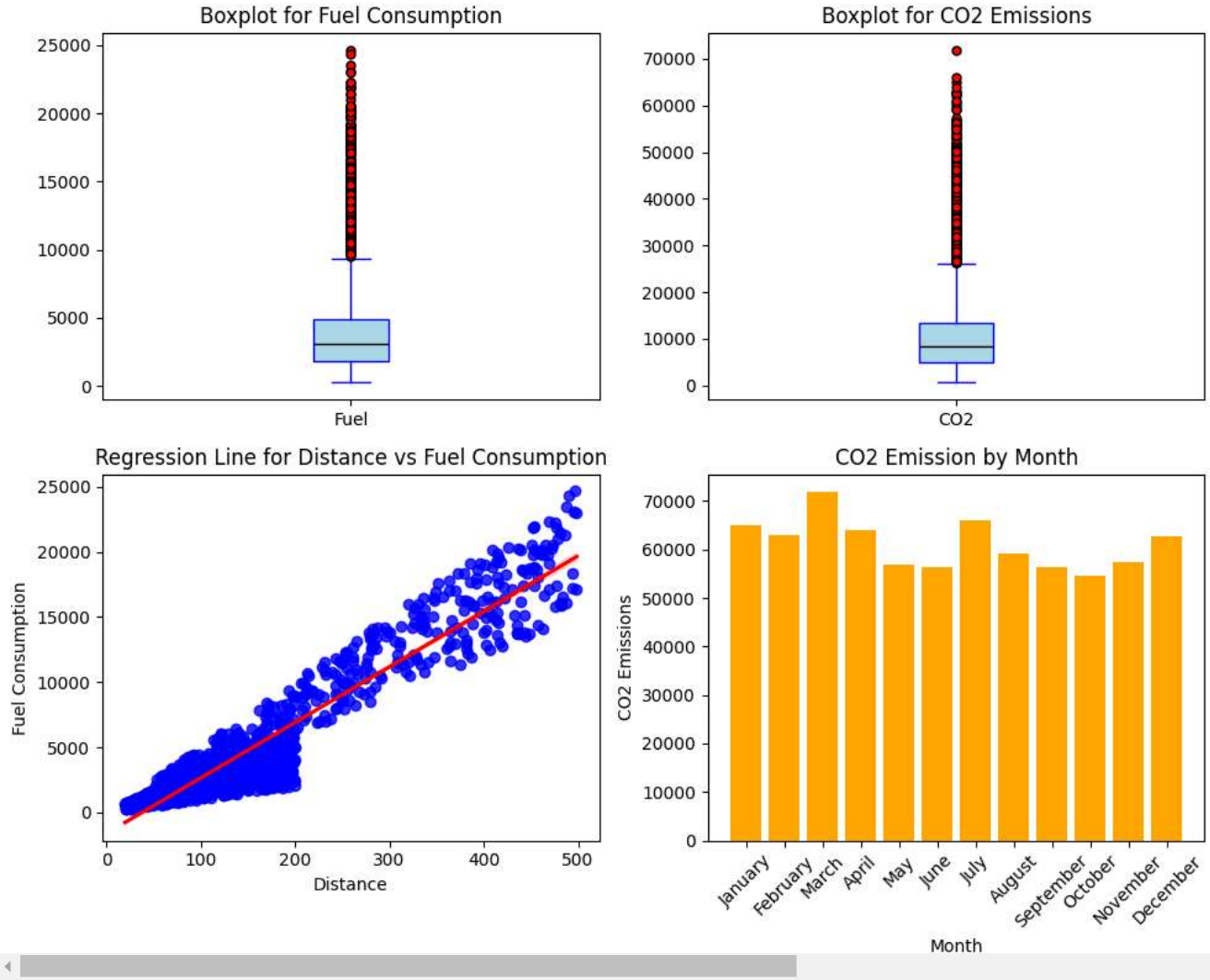

```
# or
import pandas as pd
ship3 = ship.groupby(['ship_type', 'weather_conditions'])[['engine_efficiency', 'distance']].mean().reset_index()

pd.DataFrame(ship3)
```

	ship_type	weather_conditions	engine_efficiency	distance
0	Fishing Trawler	Calm	81.936373	124.128627
1	Fishing Trawler	Moderate	82.262745	133.637549
2	Fishing Trawler	Stormy	82.220625	125.008229
3	Oil Service Boat	Calm	83.531635	87.397925
4	Oil Service Boat	Moderate	83.858810	88.427619
5	Oil Service Boat	Stormy	82.824715	83.673415
6	Surfer Boat	Calm	81.435446	107.604455
7	Surfer Boat	Moderate	82.098000	110.391455
8	Surfer Boat	Stormy	83.684159	115.502832
9	Tanker Ship	Calm	82.265195	269.752857
10	Tanker Ship	Moderate	82.894597	268.255403
11	Tanker Ship	Stormy	81.406923	261.795154

```
fig = px.bar(ship3, x='ship_type', y='engine_efficiency', color='weather_conditions',text='distance')
fig
```


```
import matplotlib.pyplot as plt
import seaborn as sns  # Don't forget to import seaborn for regplot
import numpy as np

fuel = ship['fuel_consumption']
CO2 = ship['CO2_emissions']
month = ship['month']
distance = ship['distance']

fig, axs = plt.subplots(2, 2, figsize=(10, 8))

# a) Boxplot for fuel consumption
axs[0, 0].boxplot(
    [fuel],
    patch_artist=True,  # Enables color fill inside the boxes
    boxprops=dict(facecolor='lightblue', color='blue'),  # Box color
    capprops=dict(color='blue'),  # Caps color
    whiskerprops=dict(color='blue'),  # Whisker color
    flierprops=dict(markerfacecolor='red', marker='o', markersize=5, linestyle='none'),  # Outliers
    medianprops=dict(color='black')  # Median line color
)
axs[0, 0].set_xticklabels(['Fuel'])
axs[0, 0].set_title('Boxplot for Fuel Consumption')

# b) Boxplot for CO2 emissions
axs[0, 1].boxplot(
    [CO2],
    patch_artist=True,
    boxprops=dict(facecolor='lightblue', color='blue'),
    capprops=dict(color='blue'),
    whiskerprops=dict(color='blue'),
    flierprops=dict(markerfacecolor='red', marker='o', markersize=5, linestyle='none'),
    medianprops=dict(color='black')
)
axs[0, 1].set_xticklabels(['CO2'])
axs[0, 1].set_title('Boxplot for CO2 Emissions')

# c) Bar plot for month-wise CO2 emissions
axs[1, 1].bar(month, CO2, color='orange')
axs[1, 1].set_title('CO2 Emission by Month')
axs[1, 1].set_xlabel('Month')
axs[1, 1].set_ylabel('CO2 Emissions')
axs[1, 1].tick_params(axis='x', rotation=45)  # Rotate labels for better visibility

# d) Regression plot for distance vs fuel consumption
sns.regplot(x='distance', y='fuel_consumption', data=ship, scatter_kws={'color': 'blue'}, line_kws={'color': 'red'}, ax=axs[1, 0])

# Set title and labels
axs[1, 0].set_title('Regression Line for Distance vs Fuel Consumption')
axs[1, 0].set_xlabel('Distance')
axs[1, 0].set_ylabel('Fuel Consumption')

# Adjust layout for better spacing
```

```
plt.tight_layout()

# Show the plot
plt.show()
```




```
# Q.3) a)which is the longest route    b)which ship id is done maximum service

ship.head(5)
```

	ship_id	ship_type	route_id	month	distance	fuel_type	fuel_consumption	CO2_emissions	weather_conditions	engine_efficienc
0	NG001	Oil Service Boat	Warri-Bonny	January	132.26	HFO	3779.77	10625.76	Stormy	92.1
1	NG001	Oil Service Boat	Port Harcourt-Lagos	February	128.52	HFO	4461.44	12779.73	Moderate	92.9
		Oil Service	Port							

```
# Find the row with the longest distance (route)

longest_route = ship.loc[ship['distance'].idxmax()]
print(longest_route)
```

```
ship_id                  NG067
ship_type          Tanker Ship
route_id           Lagos-Apapa
month                 February
distance                498.55
fuel_type               Diesel
fuel_consumption       22973.21
CO2_emissions          62936.17
weather_conditions       Stormy
engine_efficiency         70.49
Name: 793, dtype: object
```

```
# b) b)which ship id is done maximum service

# Group by 'ship_id' and sum the 'distance' for each ship

max_service_ship = ship.groupby('ship_id')['distance'].sum().idxmax()

print(f"The ship with the maximum service (total distance traveled) is: {max_service_ship}")
```

```
The ship with the maximum service (total distance traveled) is: NG048
```

```
# Q.4) Perform the descriptive statisics and conclude the data

descriptive_statistics=ship.describe()
descriptive_statistics
```

	distance	fuel_consumption	CO2_emissions	engine_efficiency
count	1440.000000	1440.000000	1440.000000	1440.000000
mean	151.753354	4844.246535	13365.454882	82.582924
std	108.472230	4892.352813	13567.650118	7.158289
min	20.080000	237.880000	615.680000	70.010000