Final-Project

##Submited by:-

- **Ashish Pandya(000543403)**
- **Dhruvil Patel(000544320)**

```
# import matlab and numpy
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

# Load the dataset into a DataFrame
file_path = '/content/COVID_Parcel_Business.csv'
data = pd.read_csv(file_path)

data.head(12)
```

```
{"type":"dataframe","variable_name":"data"}
```

```
data.shape
```

```
(190719, 4)
```

```
data['FakeCustomerID'].unique()
```

```
array([718117, 317344, 631279, ..., 994361, 811549, 228893])
```

```
data['THE_YEAR'].unique()
```

```
array([2021, 2020, 2019, 2018])
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 190719 entries, 0 to 190718
Data columns (total 4 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   FakeCustomerID  190719 non-null  int64
 1   THE_YEAR        190719 non-null  int64
 2   THE_WEEK        190719 non-null  int64
 3   VOLUME          190719 non-null  int64
dtypes: int64(4)
memory usage: 5.8 MB
```

```
missing_value = data.isnull().sum()
missing_value
```

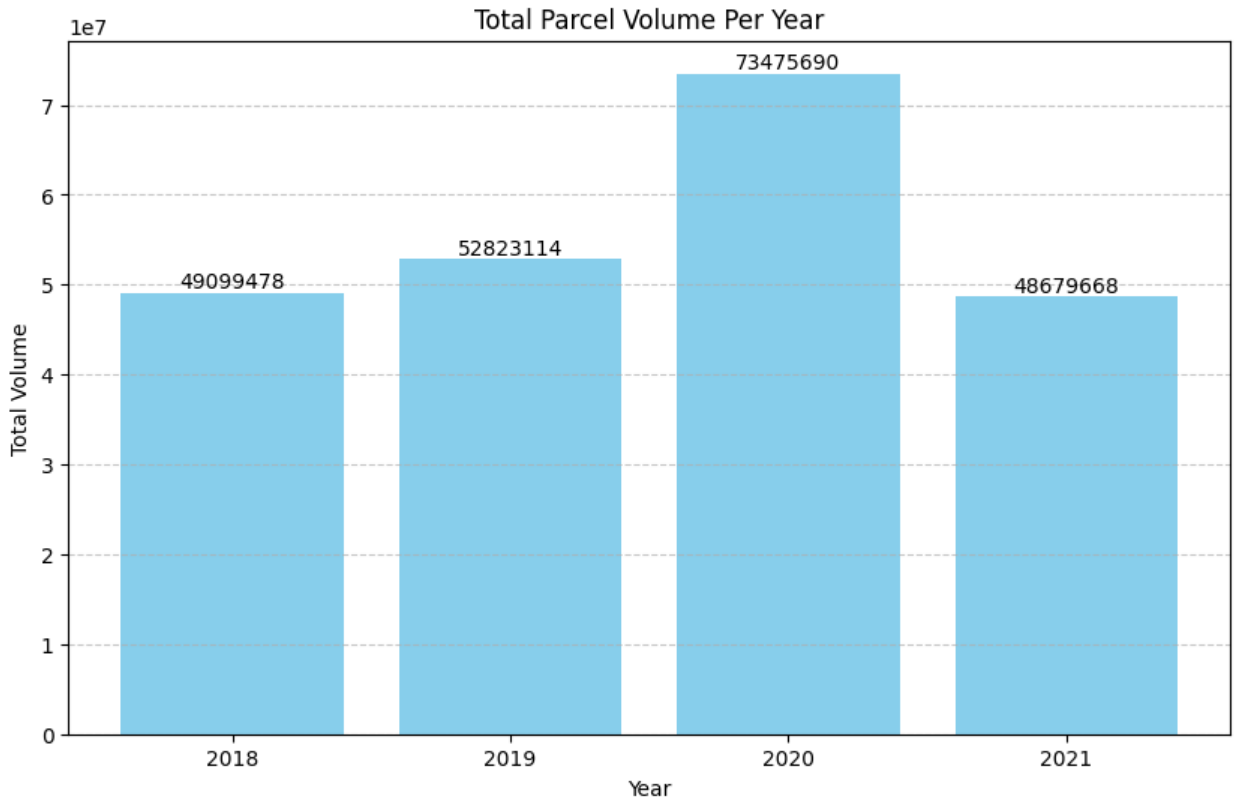
```
FakeCustomerID    0
THE_YEAR          0
```

```
THE_WEEK          0
VOLUME            0
dtype: int64

data.drop_duplicates(inplace=True)

data.shape

(190719, 4)
```

Total Volume per year

```
# Total volume per year
volume_per_year = data.groupby('THE_YEAR')
['VOLUME'].sum().reset_index()
print(volume_per_year)

# Bar chart to visualize trends
plt.figure(figsize=(10, 6))
plt.bar(volume_per_year['THE_YEAR'], volume_per_year['VOLUME'],
color='skyblue')
plt.title('Total Parcel Volume Per Year')
plt.xlabel('Year')
plt.ylabel('Total Volume')
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Optional: Add
gridlines for better readability

# Set the x-axis ticks to match the years
plt.xticks(volume_per_year['THE_YEAR'], rotation=0)  # Ensures only
the year values appear

# Add data labels
for index, value in enumerate(volume_per_year['VOLUME']):
    plt.text(volume_per_year['THE_YEAR'][index], value, str(value),
ha='center', va='bottom')

# Save the plot
plt.savefig('Total_Parcel_Volume_Per_Year_Bar.png',
bbox_inches='tight')
plt.show()
```

```
   THE_YEAR    VOLUME
0      2018  49099478
1      2019  52823114
2      2020  73475690
3      2021  48679668
```



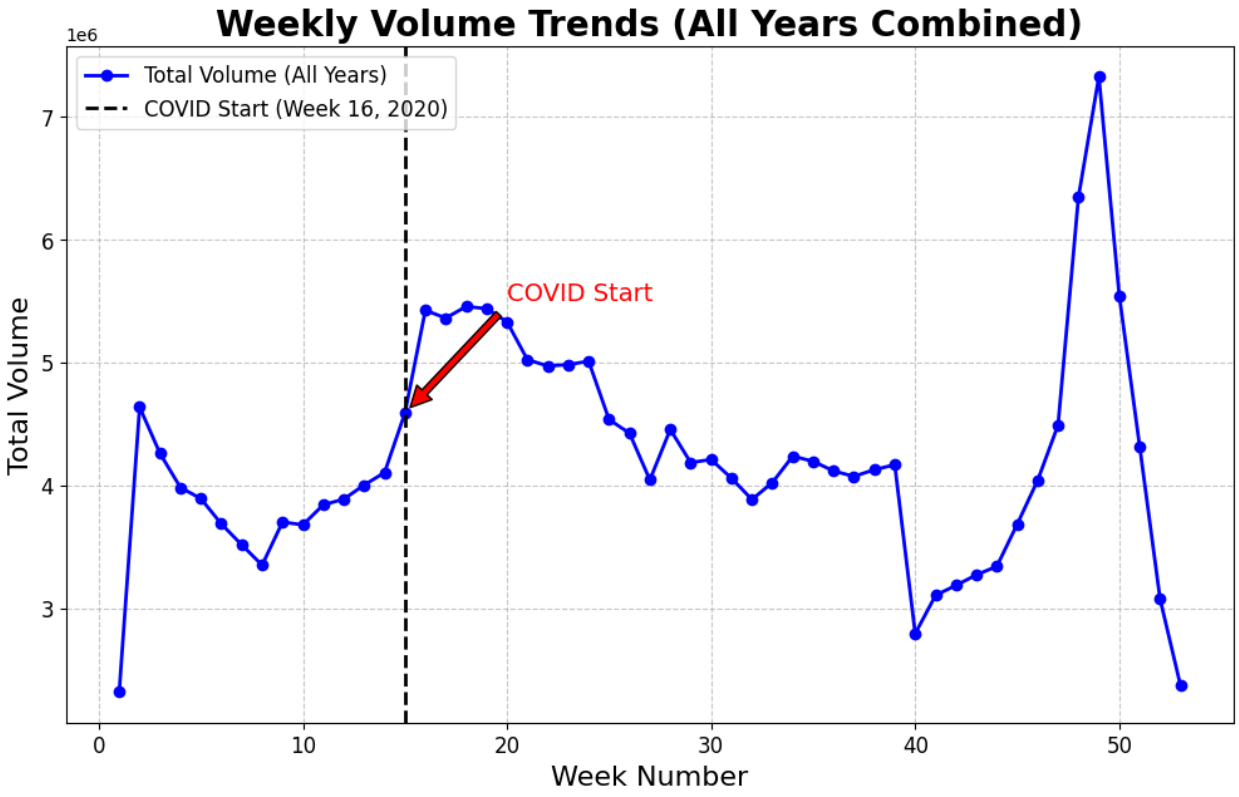
Weekly Parcel Volume Comparison (2019-2021)

###1.When were customer volumes first impacted by COVID-19?

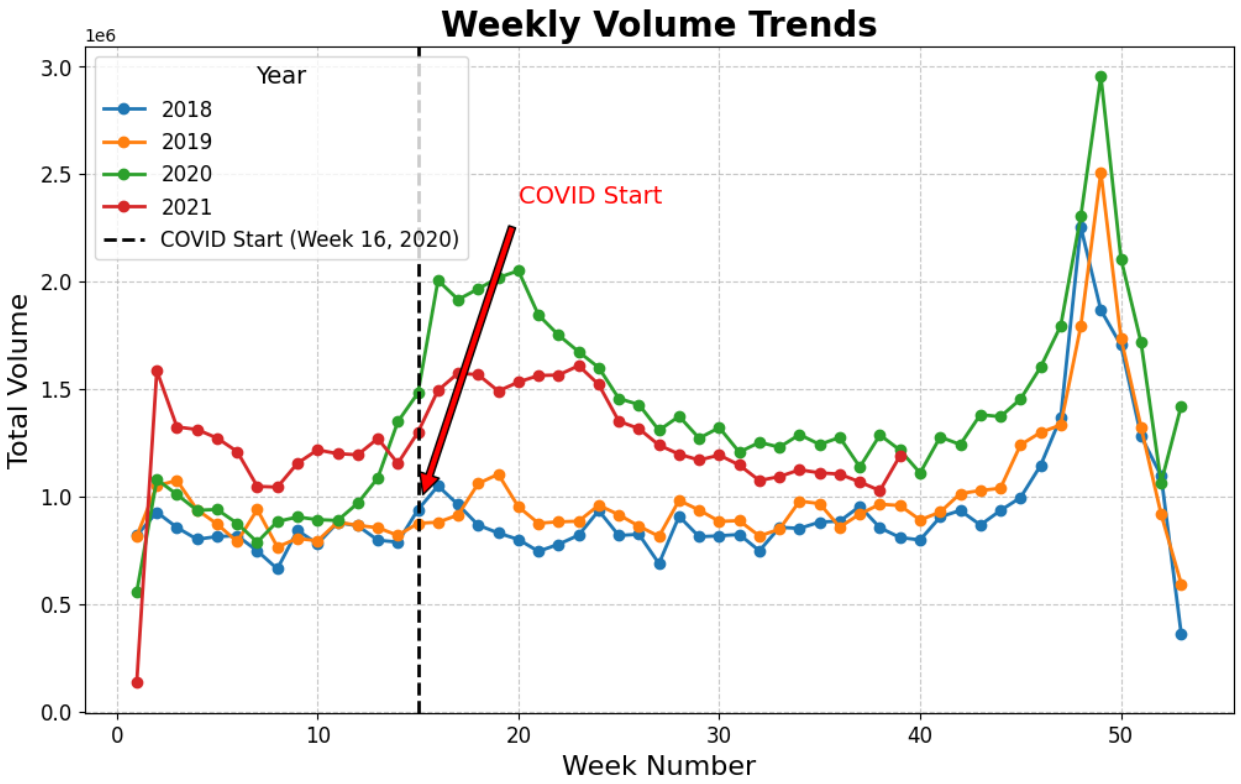
###2.What events within the COVID timeline may have contributed to the change?

```
# Aggregate total volume by week across all years
weekly_total_volume = data.groupby('THE_WEEK')
['VOLUME'].sum().reset_index()

plt.figure(figsize=(12, 7))  # Increase figure size for better
visibility

# Plot a single line for total weekly volume
plt.plot(weekly_total_volume['THE_WEEK'],
weekly_total_volume['VOLUME'],
         marker='o', linestyle='-', linewidth=2, markersize=6,
color='blue', label='Total Volume (All Years)')

# Add vertical line for COVID start
plt.axvline(x=15, color='black', linestyle='--', linewidth=2,
```

```
label='COVID Start (Week 16, 2020)')

# Customize the plot
plt.title('Weekly Volume Trends (All Years Combined)', fontsize=20,
fontweight='bold')
plt.xlabel('Week Number', fontsize=16)
plt.ylabel('Total Volume', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(fontsize=12, loc='upper left')
plt.grid(True, linestyle='--', alpha=0.7)

# Annotate COVID start point
covid_start_volume =
weekly_total_volume[weekly_total_volume['THE_WEEK'] == 15]
['VOLUME'].values[0]
plt.annotate('COVID Start', xy=(15, covid_start_volume),
             xytext=(20, covid_start_volume * 1.2),
             arrowprops=dict(facecolor='red', shrink=0.05),
fontsize=14, color='red')

# Save the plot
plt.savefig('Weekly_Volume_Trends_All_Years.png', bbox_inches='tight')

plt.show()
```



```
# Weekly Trends
volume_trends = data.groupby(['THE_YEAR', 'THE_WEEK'])
['VOLUME'].sum().reset_index()

plt.figure(figsize=(12, 7))  # Increase figure size for better
visibility

# Plot volumes for each year separately
for year in volume_trends['THE_YEAR'].unique():
    yearly_data = volume_trends[volume_trends['THE_YEAR'] == year]
    plt.plot(yearly_data['THE_WEEK'], yearly_data['VOLUME'],
marker='o', linestyle='-', linewidth=2, markersize=6, label=f'{year}')

# Add vertical line for COVID start
plt.axvline(x=15, color='black', linestyle='--', linewidth=2,
label='COVID Start (Week 16, 2020)')

# Customize the plot
plt.title('Weekly Volume Trends', fontsize=20, fontweight='bold')
plt.xlabel('Week Number', fontsize=16)
plt.ylabel('Total Volume', fontsize=16)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(title='Year', fontsize=12, title_fontsize=14, loc='upper
left')
plt.grid(True, linestyle='--', alpha=0.7)

# Annotate COVID start point
plt.annotate('COVID Start', xy=(15,
volume_trends[volume_trends['THE_WEEK'] == 15]['VOLUME'].values[0]),
             xytext=(20, volume_trends['VOLUME'].max() * 0.8),
             arrowprops=dict(facecolor='red', shrink=0.05),
fontsize=14, color='red')

# Save the plot
plt.savefig('Weekly_Volume_Trends.png', bbox_inches='tight')

plt.show()
```

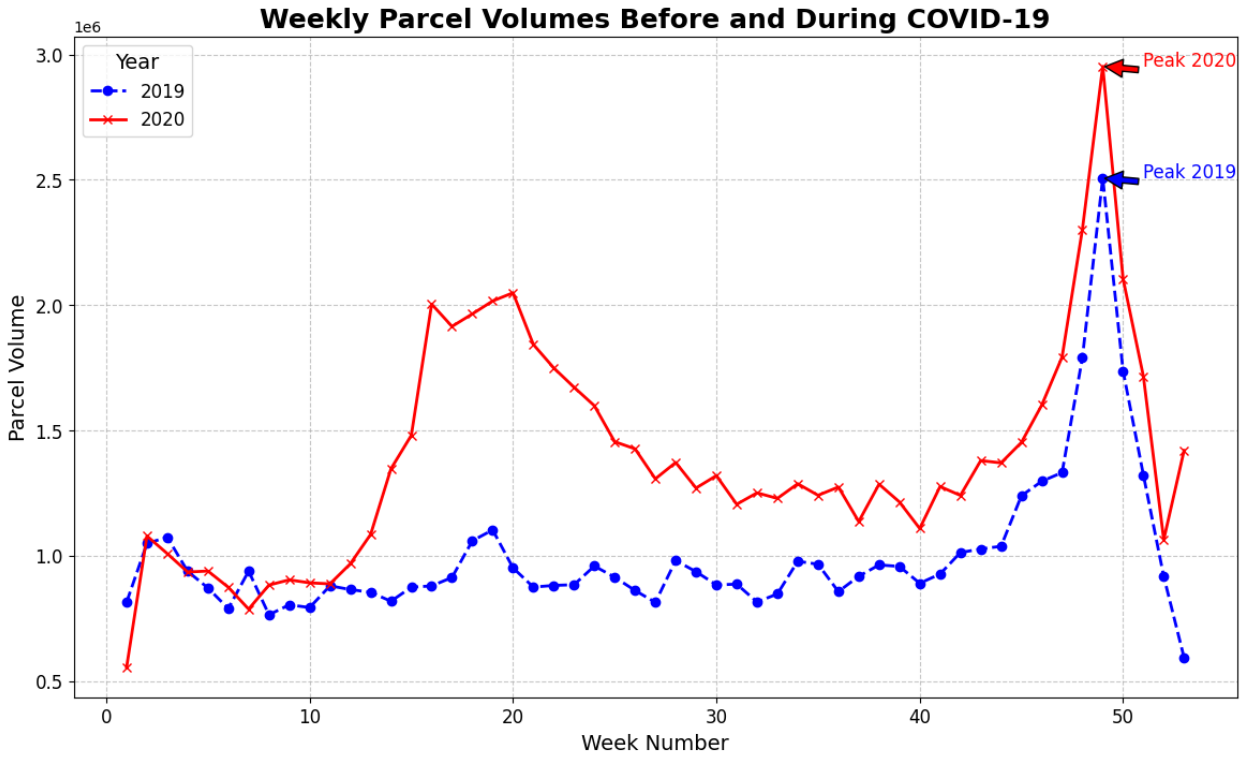


##How did the COVID-19 pandemic impact ABC Company's parcel business?

How did COVID-19 impact peak season in 2020?

```
# Compute yearly parcel volumes before and during COVID-19
pre_covid = data[data['THE_YEAR'] == 2019]
during_covid = data[data['THE_YEAR'] == 2020]

weekly_pre_covid = pre_covid.groupby('THE_WEEK').sum()
weekly_during_covid = during_covid.groupby('THE_WEEK').sum()

# Plot the volumes
plt.figure(figsize=(14, 8))
plt.plot(weekly_pre_covid.index, weekly_pre_covid['VOLUME'],
label='2019', color='blue', marker='o', linestyle='--', linewidth=2)
plt.plot(weekly_during_covid.index, weekly_during_covid['VOLUME'],
label='2020', color='red', marker='x', linestyle='-', linewidth=2)

# Customize the plot
plt.title('Weekly Parcel Volumes Before and During COVID-19',
fontsize=18, fontweight='bold')
plt.xlabel('Week Number', fontsize=14)
plt.ylabel('Parcel Volume', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(title='Year', fontsize=12, title_fontsize=14, loc='upper
```

```
left')
plt.grid(True, linestyle='--', alpha=0.7)

# Annotate significant points
max_pre_covid = weekly_pre_covid['VOLUME'].max()
max_during_covid = weekly_during_covid['VOLUME'].max()
plt.annotate('Peak 2019', xy=(weekly_pre_covid['VOLUME'].idxmax(),
max_pre_covid), xytext=(weekly_pre_covid['VOLUME'].idxmax()+2,
max_pre_covid+2000),
             arrowprops=dict(facecolor='blue', shrink=0.05),
fontsize=12, color='blue')
plt.annotate('Peak 2020', xy=(weekly_during_covid['VOLUME'].idxmax(),
max_during_covid), xytext=(weekly_during_covid['VOLUME'].idxmax()+2,
max_during_covid+2000),
             arrowprops=dict(facecolor='red', shrink=0.05),
fontsize=12, color='red')

# Save the plot
plt.savefig('Weekly_Parcel_Volumes_Before_and_During_COVID-19.png',
bbox_inches='tight')
plt.show()
```




```
# Detect significant changes in weekly volumes
volume_change = weekly_during_covid['VOLUME'].pct_change()
significant_change = volume_change[volume_change.abs() > 0.1]  #
```

```
Define a threshold for significant change
print(significant_change)
```

```
THE_WEEK
2     0.943265
8     0.122049
13    0.121344
14    0.241488
16    0.353829
21   -0.100594
37   -0.108158
38    0.131244
41    0.151064
43    0.111986
46    0.102511
47    0.119903
48    0.284395
49    0.282333
50   -0.287217
51   -0.185115
52   -0.379688
53    0.334871
Name: VOLUME, dtype: float64
```

```
# Define Pre-COVID and COVID Periods with week
pre_covid = data[(data['THE_YEAR'] == 2019) & (data['THE_WEEK'] <=
15)]
covid_period = data[(data['THE_YEAR'] == 2020) & (data['THE_WEEK'] >=
16)]

# Calculate Industry Standard Growth Rate (ISGR)
pre_covid_2020 = data[(data['THE_YEAR'] == 2020) & (data['THE_WEEK']
<= 15)]
pre_covid_2019 = data[(data['THE_YEAR'] == 2019) & (data['THE_WEEK']
<= 15)]

pre_covid_2020_volume = pre_covid_2020['VOLUME'].sum()
pre_covid_2019_volume = pre_covid_2019['VOLUME'].sum()

ISGR = ((pre_covid_2020_volume - pre_covid_2019_volume) /
pre_covid_2019_volume) * 100
print(f"Industry Standard Growth Rate (ISGR): {ISGR:.2f}%")
```

```
Industry Standard Growth Rate (ISGR): 11.40%
```

```
# Aggregate Data by Customer
pre_covid_agg = pre_covid.groupby(['FakeCustomerID']).agg(
    Pre_COVID_Volume=('VOLUME', 'sum')
).reset_index()

covid_period_agg = covid_period.groupby(['FakeCustomerID']).agg(
```

```
    COVID_Volume=('VOLUME', 'sum')
).reset_index()

# Merge Pre-COVID and COVID Data
customer_data = pd.merge(pre_covid_agg, covid_period_agg,
on='FakeCustomerID', how='outer').fillna(0)

customer_data.head()
```


```
{"summary":"{\n  \"name\": \"customer_data\",\n  \"rows\": 1055,\n
\"fields\": [\n    {\n      \"column\": \"FakeCustomerID\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
264912,\n        \"min\": 100771,\n        \"max\": 999362,\n
\"num_unique_values\": 1055,\n        \"samples\": [\n
308228,\n          823286,\n          835653\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Pre_COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
109331.10868730562,\n        \"min\": 0.0,\n        \"max\":
2269733.0,\n        \"num_unique_values\": 843,\n        \"samples\":
[\n          260.0,\n          448.0,\n          760.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
516772.94818821957,\n        \"min\": 0.0,\n        \"max\":
12420500.0,\n        \"num_unique_values\": 991,\n        \"samples\":
[\n          2593.0,\n          3780.0,\n          213907.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"customer_data"}
```


```
# Calculate Growth Rates and Categorize Customers
customer_data['Volume_Growth_Rate'] = (
    (customer_data['COVID_Volume'] -
customer_data['Pre_COVID_Volume']) /
customer_data['Pre_COVID_Volume'].replace(0, np.nan)
) * 100

def categorize(row):
    if row['Pre_COVID_Volume'] == 0 and row['COVID_Volume'] > 0:
        return 'New Customer'
    elif row['Volume_Growth_Rate'] > ISGR:
        return 'High Growth'
    elif row['Volume_Growth_Rate'] == ISGR:
        return 'Stable'
    elif row['Volume_Growth_Rate'] < 0:
        return 'Declining'
    elif row['Pre_COVID_Volume'] > 0 and row['COVID_Volume'] == 0:
        return 'Lost'
    else:
```

```
        return 'Moderately Growing'

customer_data['Customer_Category'] = customer_data.apply(categorize,
axis=1)

# Add a 'Capped_Growth' column to limit Volume_Growth_Rate to a
maximum of 100%
customer_data['Capped_Growth'] =
customer_data['Volume_Growth_Rate'].apply(lambda x: min(x, 100) if x >
0 else x)
customer_data['Volume_Growth_Rate'] = (
    (customer_data['COVID_Volume'] -
customer_data['Pre_COVID_Volume']) /
customer_data['Pre_COVID_Volume'].replace(0, np.nan)
) * 100
# Cap growth for understanding purposes
customer_data['Capped_Growth'] =
customer_data['Volume_Growth_Rate'].apply(lambda x: min(x, 100) if x >
0 else x)

customer_data.head()
```


```
{"summary":"{\n  \"name\": \"customer_data\",\n  \"rows\": 1055,\n
\"fields\": [\n    {\n      \"column\": \"FakeCustomerID\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
264912,\n        \"min\": 100771,\n        \"max\": 999362,\n
\"num_unique_values\": 1055,\n        \"samples\": [\n
308228,\n          823286,\n          835653\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Pre_COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
109331.10868730562,\n        \"min\": 0.0,\n        \"max\":
2269733.0,\n        \"num_unique_values\": 843,\n        \"samples\":
[\n          260.0,\n          448.0,\n          760.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
516772.94818821957,\n        \"min\": 0.0,\n        \"max\":
12420500.0,\n        \"num_unique_values\": 991,\n        \"samples\":
[\n          2593.0,\n          3780.0,\n          213907.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Volume_Growth_Rate\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 22852.284886302372,\n        \"min\": -
100.0,\n        \"max\": 691833.3333333333,\n
\"num_unique_values\": 1011,\n        \"samples\": [\n
51.86846038863976,\n          147.81021897810217,\n
307.9003181336161\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Customer_Category\",\n      \"properties\": {\n        \"dtype\":
```

\"category\",\n   \"num_unique_values\": 4,\n   \"samples\":
[\n   \"Declining\",\n   \"Moderately Growing\",\n
\"High Growth\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Capped_Growth\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 54.136164656890735,\n   \"min\": -
100.0,\n   \"max\": 100.0,\n   \"num_unique_values\": 250,\n
\"samples\": [\n   10.667752442996743,\n   -
26.16155988857939,\n   96.39278557114228\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"customer_data"}

```
# Peak Season Analysis
peak_season_2020 = data[(data['THE_YEAR'] == 2020) &
(data['THE_WEEK'].between(40, 52))]
peak_season_2019 = data[(data['THE_YEAR'] == 2019) &
(data['THE_WEEK'].between(40, 52))]

peak_2020_volume = peak_season_2020['VOLUME'].sum()
peak_2019_volume = peak_season_2019['VOLUME'].sum()
print(f"Peak Season 2019 Volume: {peak_2019_volume}")
print(f"Peak Season 2020 Volume: {peak_2020_volume}")
```

```
Peak Season 2019 Volume: 17031369
Peak Season 2020 Volume: 21356413
```

```
#Volume Tier Analysis
customer_data['Volume_Tier'] = pd.cut(
    customer_data['COVID_Volume'],
    bins=[-1, 100, 500, np.inf],
    labels=['Low Volume', 'Medium Volume', 'High Volume']
)

volume_tier_impact = customer_data.groupby('Volume_Tier').agg(
    Pre_COVID_Volume=('Pre_COVID_Volume', 'sum'),
    COVID_Volume=('COVID_Volume', 'sum'),
    Customer_Count=('FakeCustomerID', 'count')
).reset_index()
```

```
<ipython-input-24-47976a92a19d>:8: FutureWarning: The default of
observed=False is deprecated and will be changed to True in a future
version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
  volume_tier_impact = customer_data.groupby('Volume_Tier').agg(
```

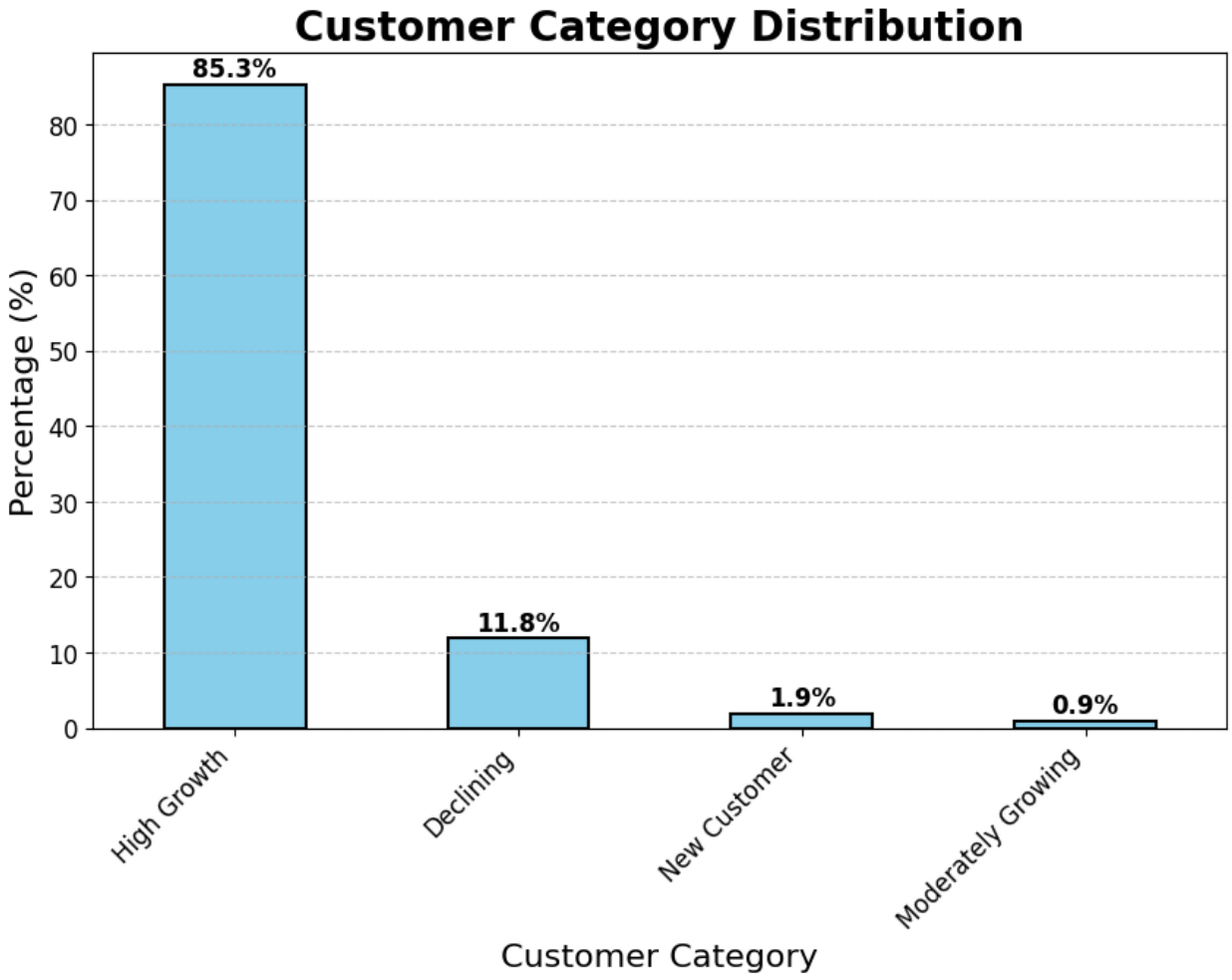
```
# Define customer groups based on volume thresholds and discounts
def assign_customer_size(volume):
    if volume > 500000:
        return 'Enterprise'
    elif 200000 <= volume <= 500000:
        return 'Large'
```

```
    elif 10000 <= volume < 200000:
        return 'Medium'
    elif 1000 <= volume < 10000:
        return 'Small'
    else:
        return 'Other'

customer_data['Customer_Size'] =
customer_data['COVID_Volume'].apply(assign_customer_size)

# Calculate discount impact
discounts = {'Enterprise': 22, 'Large': 17, 'Medium': 10, 'Small': 4,
'Other': 0}
customer_data['Discount'] =
customer_data['Customer_Size'].map(discounts)

# Group data by customer size
grouped_discounts = customer_data.groupby('Customer_Size').agg(
    Total_Volume=('COVID_Volume', 'sum'),
    Average_Discount=('Discount', 'mean'),
    Customer_Count=('FakeCustomerID', 'count')
).reset_index()
```

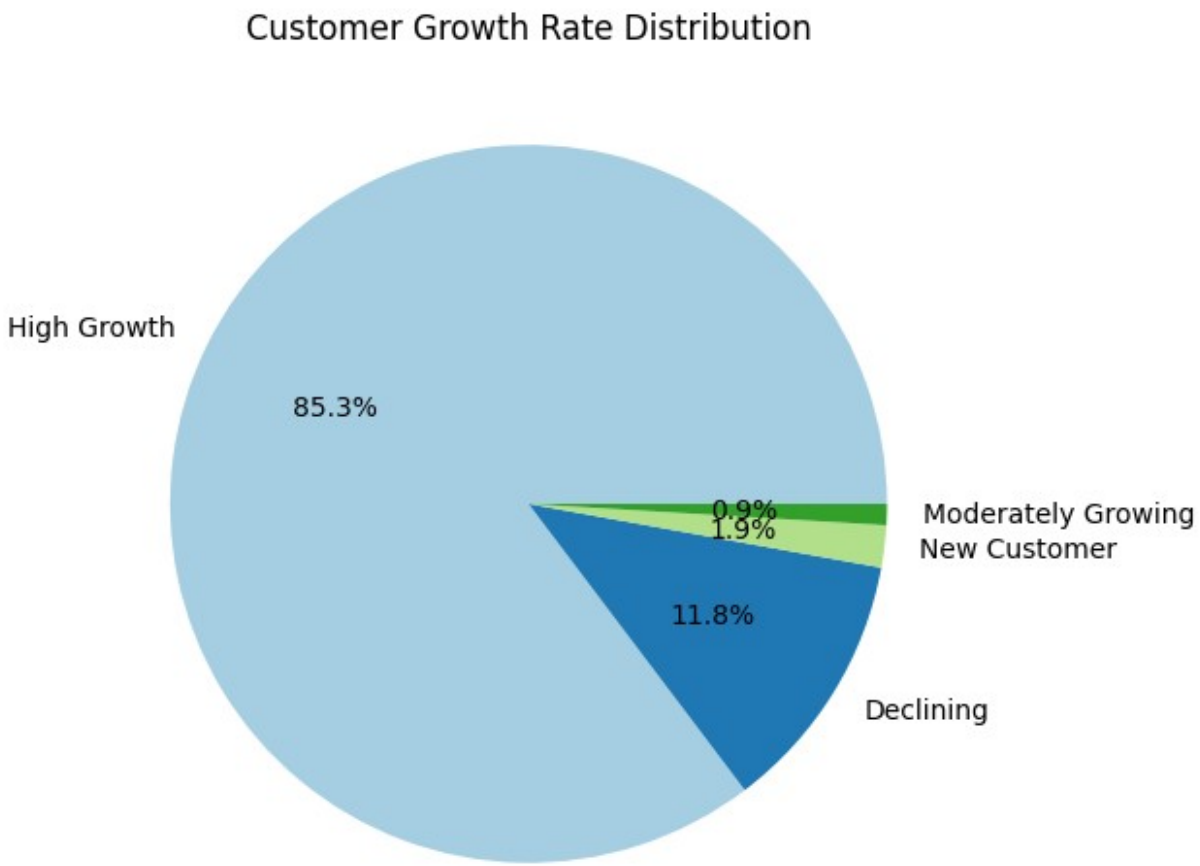
What percent of each customer group is growing, moderately growing, and declining during the COVID observation period?

```
# Customer Category Distribution
customer_category_distribution =
customer_data['Customer_Category'].value_counts(normalize=True) * 100

plt.figure(figsize=(10, 6))  # Increase figure size for better
visibility
customer_category_distribution.plot(kind='bar', color='skyblue',
edgecolor='black', linewidth=1.5)

# Customize the plot
plt.title('Customer Category Distribution', fontsize=20,
fontweight='bold')
plt.xlabel('Customer Category', fontsize=16)
plt.ylabel('Percentage (%)', fontsize=16)
plt.xticks(rotation=45, fontsize=12, ha='right')  # Rotate x-axis
labels for better readability
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotate bars with percentage values
for index, value in enumerate(customer_category_distribution):
```

```
    plt.text(index, value + 1, f'{value:.1f}%', ha='center',
fontsize=12, fontweight='bold')

# Save the plot
plt.savefig('Customer_Category_Distribution.png', bbox_inches='tight')

plt.show()
```




```
# Pie Chart: Growth Rate Distribution
plt.figure(figsize=(8, 6))
customer_data['Customer_Category'].value_counts().plot(kind='pie',
autopct='%1.1f%%', colors=plt.cm.Paired.colors)
plt.title('Customer Growth Rate Distribution')
plt.ylabel('')
plt.savefig('Growth_Rate_Distribution_Pie.png')
plt.show()
```




```
# Filter rows where Customer_Category is 'New Customer'
new_customers = customer_data[customer_data['Customer_Category'] ==
'New Customer']

# Print the rows for New Customers
print("Rows for New Customers:")
print(new_customers)

# Save the new customers to a CSV file for review
new_customers.to_csv('New_Customers.csv', index=False)
```

```
Rows for New Customers:
      FakeCustomerID  Pre_COVID_Volume  COVID_Volume
Volume_Growth_Rate  \
14            108454               0.0        1955.0
NaN
111           190192               0.0       12609.0
NaN
139           212169               0.0           3.0
NaN
```

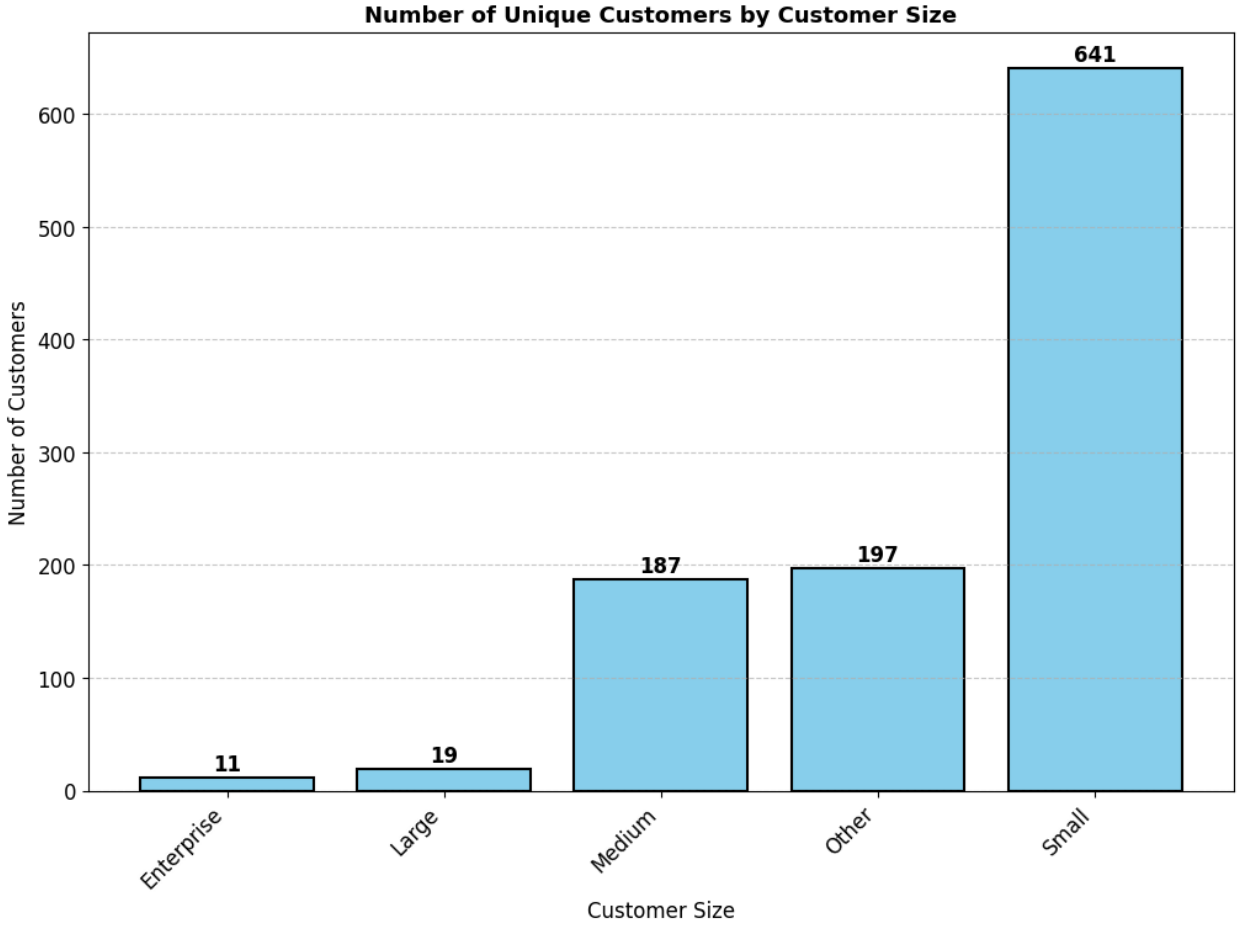
```
171            239408                0.0        3481.0
NaN
238            295997                0.0        5732.0
NaN
324            359427                0.0       57302.0
NaN
416            434228                0.0        4739.0
NaN
461            475022                0.0        2287.0
NaN
658            650813                0.0        5377.0
NaN
712            709440                0.0        5194.0
NaN
724            719389                0.0        3792.0
NaN
727            720407                0.0        9306.0
NaN
807            792169                0.0        1878.0
NaN
889            866497                0.0        1127.0
NaN
942            904643                0.0           2.0
NaN
1005           958309                0.0        2699.0
NaN
1011           965584                0.0        7567.0
NaN
1020           973720                0.0       22395.0
NaN
1025           976180                0.0        2507.0
NaN
1043           990926                0.0       10247.0
NaN

     Customer_Category  Capped_Growth  Volume_Tier Customer_Size
Discount
14         New Customer            NaN  High Volume          Small
4
111        New Customer            NaN  High Volume         Medium
10
139        New Customer            NaN   Low Volume          Other
0
171        New Customer            NaN  High Volume          Small
4
238        New Customer            NaN  High Volume          Small
4
324        New Customer            NaN  High Volume         Medium
10
```

```
416       New Customer             NaN  High Volume          Small
4
461       New Customer             NaN  High Volume          Small
4
658       New Customer             NaN  High Volume          Small
4
712       New Customer             NaN  High Volume          Small
4
724       New Customer             NaN  High Volume          Small
4
727       New Customer             NaN  High Volume          Small
4
807       New Customer             NaN  High Volume          Small
4
889       New Customer             NaN  High Volume          Small
4
942       New Customer             NaN   Low Volume          Other
0
1005      New Customer             NaN  High Volume          Small
4
1011      New Customer             NaN  High Volume          Small
4
1020      New Customer             NaN  High Volume         Medium
10
1025      New Customer             NaN  High Volume          Small
4
1043      New Customer             NaN  High Volume         Medium
10
```

##How has COVID affected companies in the different customer groups (Enterprise, Large, etc.)?

```
# Count the number of unique FakeCustomerID in each Customer_Size
customer_size_counts = customer_data.groupby('Customer_Size')
['FakeCustomerID'].nunique().reset_index()
customer_size_counts.columns = ['Customer_Size', 'Customer_Count']

# Plot the bar graph
plt.figure(figsize=(12, 8))  # Increase figure size for better
visibility
plt.bar(customer_size_counts['Customer_Size'],
customer_size_counts['Customer_Count'], color='skyblue',
edgecolor='black', linewidth=1.5)

# Customize the plot
plt.title('Number of Unique Customers by Customer Size', fontsize=13,
fontweight='bold')
plt.xlabel('Customer Size', fontsize=12)
plt.ylabel('Number of Customers', fontsize=12)
```

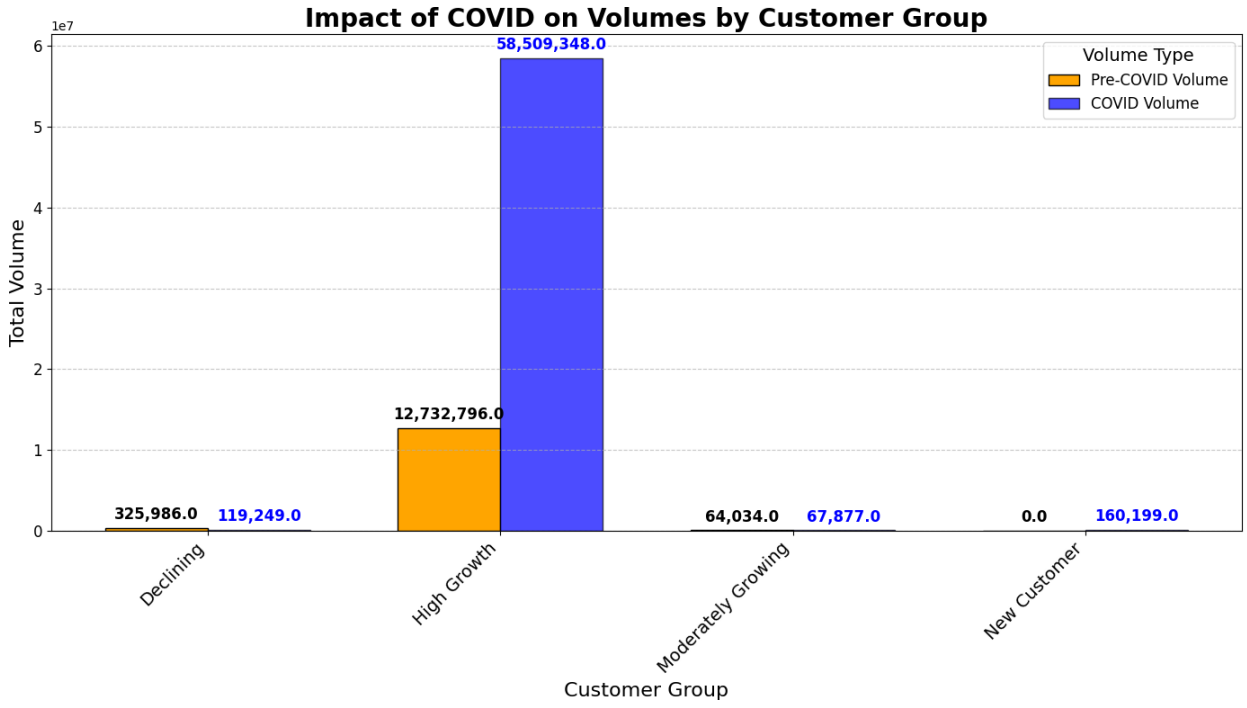
```
plt.xticks(rotation=45, fontsize=12, ha='right')  # Rotate x-axis
labels for better readability
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotate bars with count values
for index, value in enumerate(customer_size_counts['Customer_Count']):
    plt.text(index, value +
max(customer_size_counts['Customer_Count']) * 0.01, f'{value}',
ha='center', fontsize=12, fontweight='bold')

# Save the plot
plt.savefig('Customer_Size_Unique_Counts.png', bbox_inches='tight')

plt.show()
```



What was the overall impact of COVID on volumes and revenue by customer group

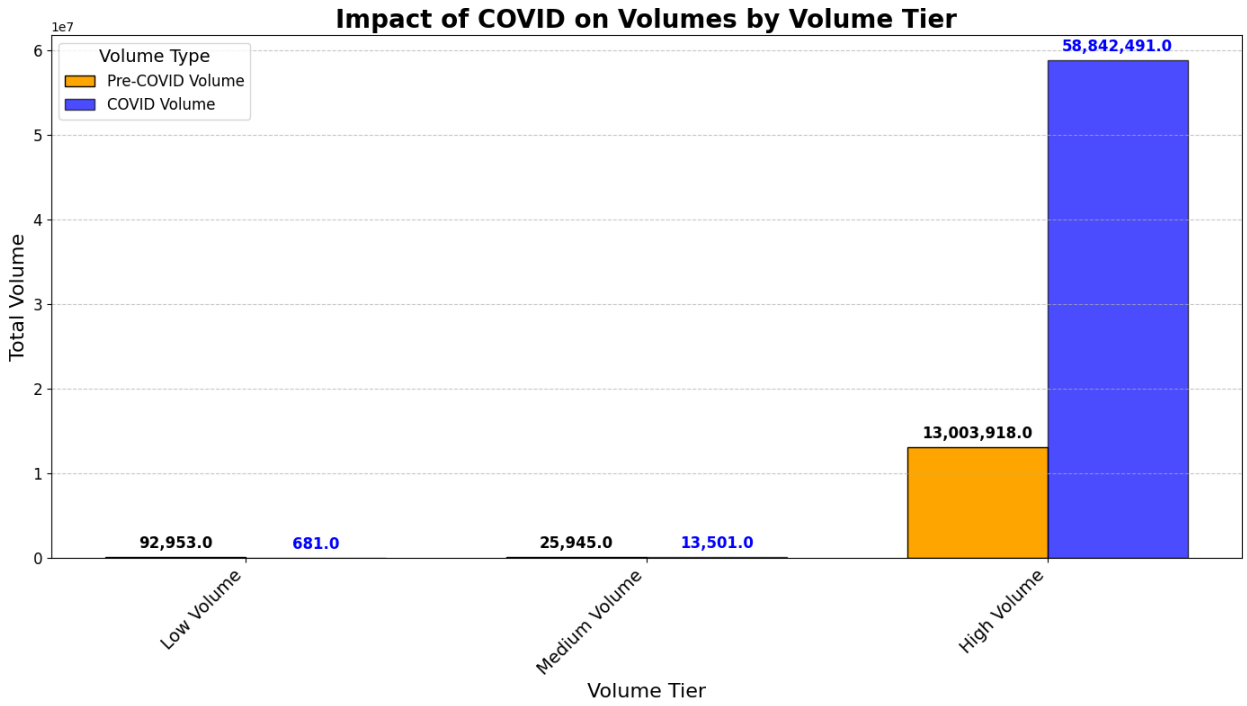
```
# Overall Impact of COVID on Volumes and Revenue by Customer Group
overall_impact = customer_data.groupby('Customer_Category').agg(
    Total_Pre_COVID_Volume=('Pre_COVID_Volume', 'sum'),
    Total_COVID_Volume=('COVID_Volume', 'sum'),
    Average_Growth_Rate=('Volume_Growth_Rate', 'mean'),
    Customer_Count=('FakeCustomerID', 'count')
).reset_index()

# Add Total Revenue Impact Placeholder (Assuming 'Revenue' Calculation
Exists)
# Example Revenue Logic: Replace 'Revenue_Per_Unit' with actual
column/constant
Revenue_Per_Unit = 1.2  # Example: $1.2 revenue per unit volume
overall_impact['Pre_COVID_Revenue'] =
overall_impact['Total_Pre_COVID_Volume'] * Revenue_Per_Unit
overall_impact['COVID_Revenue'] = overall_impact['Total_COVID_Volume']
* Revenue_Per_Unit
overall_impact['Revenue_Change'] = overall_impact['COVID_Revenue'] -
overall_impact['Pre_COVID_Revenue']

# Visualization: COVID Impact on Volumes and Revenue
plt.figure(figsize=(14, 8))  # Increase figure size for better
visibility

bar_width = 0.35
index = np.arange(len(overall_impact['Customer_Category']))

# Plot side-by-side bars
plt.bar(index, overall_impact['Total_Pre_COVID_Volume'], bar_width,
color='orange', label='Pre-COVID Volume', edgecolor='black')
plt.bar(index + bar_width, overall_impact['Total_COVID_Volume'],
bar_width, color='blue', label='COVID Volume', edgecolor='black',
alpha=0.7)

# Customize the plot
plt.title('Impact of COVID on Volumes by Customer Group', fontsize=20,
fontweight='bold')
plt.xlabel('Customer Group', fontsize=16)
plt.ylabel('Total Volume', fontsize=16)
plt.xticks(index + bar_width / 2, overall_impact['Customer_Category'],
fontsize=14, rotation=45, ha='right')
plt.yticks(fontsize=12)
plt.legend(title='Volume Type', fontsize=12, title_fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotate bars with volume values
for i in range(len(overall_impact)):
```

```
    plt.text(index[i], overall_impact['Total_Pre_COVID_Volume'][i] +
0.05 * max(overall_impact['Total_Pre_COVID_Volume']),
f'{overall_impact["Total_Pre_COVID_Volume"][i]:,}', ha='center',
va='bottom', fontsize=12, fontweight='bold')
    plt.text(index[i] + bar_width,
overall_impact['Total_COVID_Volume'][i] + 0.05 *
max(overall_impact['Total_Pre_COVID_Volume']),
f'{overall_impact["Total_COVID_Volume"][i]:,}', ha='center',
va='bottom', fontsize=12, fontweight='bold', color='blue')

plt.tight_layout()

# Save the plot
plt.savefig('COVID_Impact_Volumes.png', bbox_inches='tight')

plt.show()
```

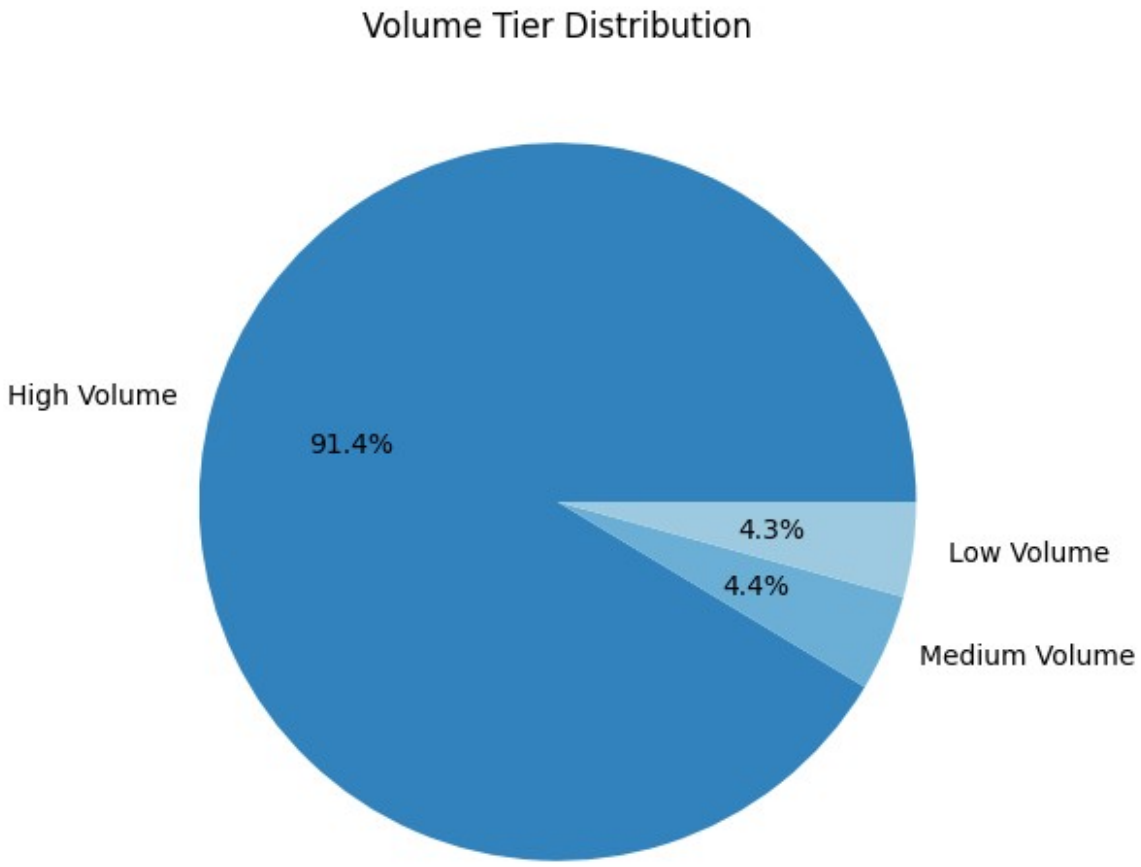



```
# Visualization: Volume Tier Impact
plt.figure(figsize=(14, 8))  # Increase figure size for better
visibility

bar_width = 0.35
index = np.arange(len(volume_tier_impact['Volume_Tier']))

# Plot side-by-side bars
plt.bar(index, volume_tier_impact['Pre_COVID_Volume'], bar_width,
color='orange', label='Pre-COVID Volume', edgecolor='black')
plt.bar(index + bar_width, volume_tier_impact['COVID_Volume'],
```

```
bar_width, color='blue', label='COVID Volume', edgecolor='black',
alpha=0.7)

# Customize the plot
plt.title('Impact of COVID on Volumes by Volume Tier', fontsize=20,
fontweight='bold')
plt.xlabel('Volume Tier', fontsize=16)
plt.ylabel('Total Volume', fontsize=16)
plt.xticks(index + bar_width / 2, volume_tier_impact['Volume_Tier'],
fontsize=14, rotation=45, ha='right')
plt.yticks(fontsize=12)
plt.legend(title='Volume Type', fontsize=12, title_fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotate bars with volume values
for i in range(len(volume_tier_impact)):
    plt.text(index[i], volume_tier_impact['Pre_COVID_Volume'][i] +
0.05 * max(volume_tier_impact['Pre_COVID_Volume']),
f'{volume_tier_impact["Pre_COVID_Volume"][i]:,}', ha='center',
va='bottom', fontsize=12, fontweight='bold')
    plt.text(index[i] + bar_width, volume_tier_impact['COVID_Volume']
[i] + 0.05 * max(volume_tier_impact['Pre_COVID_Volume']),
f'{volume_tier_impact["COVID_Volume"][i]:,}', ha='center',
va='bottom', fontsize=12, fontweight='bold', color='blue')

plt.tight_layout()

# Save the plot
plt.savefig('Volume_Tier_Impact.png', bbox_inches='tight')

plt.show()
```




```
# Pie Chart: Volume Tier Distribution
plt.figure(figsize=(8, 6))
customer_data['Volume_Tier'].value_counts().plot(kind='pie',
autopct='%1.1f%%', colors=plt.cm.tab20c.colors)
plt.title('Volume Tier Distribution')
plt.ylabel('')
plt.savefig('Volume_Tier_Distribution_Pie.png')
plt.show()
```



Final Output Summary

```
print("Customer Data Analysis:")
customer_data.head()
```

```
Customer Data Analysis:
```


```
{"summary":"{\n  \"name\": \"customer_data\",\n  \"rows\": 1055,\n
\"fields\": [\n    {\n      \"column\": \"FakeCustomerID\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
264912,\n        \"min\": 100771,\n        \"max\": 999362,\n
\"num_unique_values\": 1055,\n        \"samples\": [\n
308228,\n          823286,\n          835653\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Pre_COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
109331.10868730562,\n        \"min\": 0.0,\n        \"max\":
2269733.0,\n        \"num_unique_values\": 843,\n        \"samples\":
[\n          260.0,\n          448.0,\n          760.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"COVID_Volume\",\n
```

\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
516772.94818821957,\n   \"min\": 0.0,\n   \"max\":
12420500.0,\n   \"num_unique_values\": 991,\n   \"samples\":
[\n   2593.0,\n   3780.0,\n   213907.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Volume_Growth_Rate\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 22852.284886302372,\n   \"min\": -
100.0,\n   \"max\": 691833.3333333333,\n
\"num_unique_values\": 1011,\n   \"samples\": [\n
51.86846038863976,\n   147.81021897810217,\n
307.9003181336161\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Customer_Category\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 4,\n   \"samples\":
[\n   \"Declining\",\n   \"Moderately Growing\",\n
\"High Growth\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Capped_Growth\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 54.136164656890735,\n   \"min\": -
100.0,\n   \"max\": 100.0,\n   \"num_unique_values\": 250,\n
\"samples\": [\n   10.667752442996743,\n   -
26.16155988857939,\n   96.39278557114228\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Volume_Tier\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n   \"samples\": [\n   \"High
Volume\",\n   \"Medium Volume\",\n   \"Low Volume\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Customer_Size\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n   \"samples\": [\n
\"Other\",\n   \"Enterprise\",\n   \"Medium\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Discount\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 3,\n   \"min\": 0,\n
\"max\": 22,\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   0,\n   22,\n   10\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"customer_data"}

```
print("\nPeak Season Volumes:")
print(f"2019: {peak_2019_volume}, 2020: {peak_2020_volume}")
```

```
Peak Season Volumes:
2019: 17031369, 2020: 21356413
```

```
print("\nVolume Tier Impact:")
volume_tier_impact
```

```
Volume Tier Impact:
```


```
{"summary":"{\n  \"name\": \"volume_tier_impact\",\n  \"rows\": 3,\n
\"fields\": [\n    {\n      \"column\": \"Volume_Tier\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n          \"Low
Volume\",\n          \"Medium Volume\",\n          \"High Volume\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Pre_COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7473567.761250513,\n        \"min\": 25945.0,\n        \"max\":
13003918.0,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          92953.0,\n          25945.0,\n          13003918.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"COVID_Volume\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
33968634.63267401,\n        \"min\": 681.0,\n        \"max\":
58842491.0,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          681.0,\n          13501.0,\n          58842491.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Customer_Count\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
530,\n        \"min\": 45,\n        \"max\": 964,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          45,\n
46,\n          964\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"volume_tier_impact"}
```


```
print("\nCustomer Category Distribution:")
customer_category_distribution
```

```
Customer Category Distribution:

Customer_Category
High Growth           85.308057
Declining             11.848341
New Customer           1.895735
Moderately Growing     0.947867
Name: proportion, dtype: float64
```

```
print("\nCustomer Size and Discounts:")
grouped_discounts
```

```
Customer Size and Discounts:
```

{"summary":"{\n \"name\": \"grouped_discounts\",\n \"rows\": 5,\n
\"fields\": [\n   {\n   \"column\": \"Customer_Size\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n   \"samples\": [\n
\"Large\",\n   \"Small\",\n   \"Medium\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Total_Volume\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
17900200.294905633,\n   \"min\": 97240.0,\n   \"max\":
43397844.0,\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   5940352.0,\n   2318933.0,\n   7102304.0\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Average_Discount\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
9.044335243676011,\n   \"min\": 0.0,\n   \"max\": 22.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   17.0,\n
4.0,\n   10.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Customer_Count\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 256,\n   \"min\": 11,\n
\"max\": 641,\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   19,\n   641,\n   187\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\
n}","type":"dataframe","variable_name":"grouped_discounts"}

```
# Display the Impact Table
print("\nOverall Impact of COVID on Volumes and Revenue by Customer Group:")
overall_impact
```

```
Overall Impact of COVID on Volumes and Revenue by Customer Group:
```

{"summary":"{\n \"name\": \"overall_impact\",\n \"rows\": 4,\n
\"fields\": [\n   {\n   \"column\": \"Customer_Category\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 4,\n   \"samples\": [\n   \"High
Growth\",\n   \"New Customer\",\n   \"Declining\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Total_Pre_COVID_Volume\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
6302972.480253635,\n   \"min\": 0.0,\n   \"max\":
12732796.0,\n   \"num_unique_values\": 4,\n   \"samples\":
[\n   12732796.0,\n   0.0,\n   325986.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Total_COVID_Volume\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 29196810.93063991,\n   \"min\":
67877.0,\n   \"max\": 58509348.0,\n   \"num_unique_values\":

```
4,\n        \"samples\": [\n          58509348.0,\n
160199.0,\n          119249.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Average_Growth_Rate\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1246.9798630344735,\n
\"min\": -58.2649746467416,\n        \"max\": 2133.453762556908,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          -
58.2649746467416,\n          2133.453762556908,\n
6.986287455210548\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Customer_Count\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 427,\n        \"min\": 10,\n
\"max\": 900,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          900,\n          20,\n          125\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Pre_COVID_Revenue\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7563566.976304362,\n        \"min\": 0.0,\n        \"max\":
15279355.2,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          15279355.2,\n          0.0,\n          391183.2\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"COVID_Revenue\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 35036173.11676789,\n        \"min\":
81452.4,\n        \"max\": 70211217.6,\n        \"num_unique_values\":
4,\n        \"samples\": [\n          70211217.6,\n
192238.8,\n          143098.8\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Revenue_Change\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 27475062.549097225,\n
\"min\": -248084.40000000002,\n        \"max\": 54931862.39999999,\n
\"num_unique_values\": 4,\n        \"samples\": [\n
54931862.39999999,\n          192238.8,\n          -
248084.40000000002\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"overall_impact"}
```