Travel Agency Case Study - Code

September 14, 2024

0.1 Homework 2 Case Study - BANA 6610

0.2 Briana, Edda, Jigna, Pratik, Srikanth

A travel agency has faced a decline in sales amid the COVID-19 pandemic. In response, the management is eager to delve into the booking data to gain a deeper understanding of past booking patterns and sales trends. This analysis will serve as a foundational step toward formulating effective strategies aimed at revitalizing and boosting sales. _____

Objective: Analyze past booking patterns and sales trends to provide insight and strategies for enhancing sales.

In-House Experts: Clarified some terms with Professor Du and Google

Domain Knowledge: Technofunk: Components of Tourism Industry (useful website)

Population of Interest: Travelers from this agency's records from 2009 until 2022. Representative Sample.

Relevant Independent Variables: Days of Trip, Month, Destination Packages, Category, Groups Size, Booking Source

---

0.3 Data Cleansing Notes

We imported traveldata.csv to excel and cleaned it there. - We verified each column had the same number of rows. - We hid the original columns clientID since it wasn't needed. - We trimmed white spaces - For the null/NaN/UK cells we replaced with "Other." - We checked the consistency of the data types, making all categorical variables string and numerical variables either float_64 or int. - Checked for consistent entries - Replaced US States listed as countries to US - Clarified column titles and capitalized - Checked and fixed Trip IDs with multiple destinations - Checked and fixed the variations of spelling in each destination_package

We imported the Excel-cleaned data into JupyterLab and verified the data was still clean.

Other things to note: - Since businesses should focus on maximizing profit, not revenue, we focused on profit as the key dependent variable - There were many errors in the Destination_Countries column (i.e. duplicate names, incorrect countries, states listed as their own country, etc.) Because of this, we focused our analysis on Destination Packages. We verified unique package names. - A booking source "UK" needs to be clarified. Is this "Unknown" or "United Kingdom" perhaps? We left it as "UK." - We created the column "Trip_Days" by subtracting the "Date_of_trip_to" from

"Date_of_trip_to" column. - We corrected a date that initially listed the trip as -21. Suspecting this might be a typo, we adjusted the trip date to match the date listed in the "trip to" column.

[560]:
```
df=pd.read_csv('travel_data_3.csv')
df.head()
```

[560]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	Category	Date_of_trip_from	Date_of_trip_to	Trip_days	Year	Month_No	Month	bkgsource	Revenue	Profit	Groupsize
0	B16-42027	805	Galapagos-FIT	Ecuador	FIT	4/7/2017	4/16/2017	9	2017	4	Apr	Repeat Client	20910.0	2322.0	3
1	B11-25115	211	Deep Water Cay	Bahamas	SW	4/13/2011	4/17/2011	4	2011	4	Apr	Company referral	3295.0	659.0	1
2	B11-27267	211	Deep Water Cay	Bahamas	SW	4/23/2012	4/27/2012	4	2012	4	Apr	Company referral	3900.0	696.9	1
3	B11-25462	211	Deep Water Cay	Bahamas	SW	4/28/2011	5/2/2011	4	2011	4	Apr	Company referral	2805.0	503.0	1
4	B14-36987	235	North Riding Point	Bahamas	SW	4/5/2015	4/11/2015	6	2015	4	Apr	Repeat Client	9600.0	1920.0	2

[561]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import describe
```

[562]:
```
df['Bkgsource']=df['bkgsource'].fillna(value='Other').astype('str')
```

[563]:
```
df.isnull().sum()
```

[563]:
```
bkgref-test            0
Trip ID                0
Destination_Package    0
Destination_Country    0
Category               0
Date_of_trip_from      0
Date_of_trip_to        0
Trip_days              0
Year                   0
Month_No               0
```

```
Month                    0
bkgsource               25
Revenue                  0
Profit                   0
Groupsize                0
Bkgsource                0
dtype: int64
```

[564]:
```
df['Bkgsource'].unique()
```

[564]:
```
array(['Repeat Client', 'Company referral', 'Other', 'Friend',
       'Web/Internet/Mail', 'Agent/Outfitter', 'UK', 'Internet Chat',
       'Organizations', 'Trade Shows', 'Frontiers, Ltd.', 'Magazine'],
      dtype=object)
```

[565]:
```
df = df.drop('bkgsource', axis=1)
df.head()
```

[565]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	Category	\
0	B16-42027	805	Galapagos-FIT	Ecuador	FIT	
1	B11-25115	211	Deep Water Cay	Bahamas	SW	
2	B11-27267	211	Deep Water Cay	Bahamas	SW	
3	B11-25462	211	Deep Water Cay	Bahamas	SW	
4	B14-36987	235	North Riding Point	Bahamas	SW	

	Date_of_trip_from	Date_of_trip_to	Trip_days	Year	Month_No	Month	Revenue	\
0	4/7/2017	4/16/2017	9	2017	4	Apr	20910.0	
1	4/13/2011	4/17/2011	4	2011	4	Apr	3295.0	
2	4/23/2012	4/27/2012	4	2012	4	Apr	3900.0	
3	4/28/2011	5/2/2011	4	2011	4	Apr	2805.0	
4	4/5/2015	4/11/2015	6	2015	4	Apr	9600.0	

	Profit	Groupsize	Bkgsource
0	2322.0	3	Repeat Client
1	659.0	1	Company referral
2	696.9	1	Company referral
3	503.0	1	Company referral
4	1920.0	2	Repeat Client

[566]:
```
df.isnull().sum()
```

[566]:
```
bkgref-test            0
Trip ID                0
Destination_Package    0
Destination_Country    0
Category               0
Date_of_trip_from      0
```

```
Date_of_trip_to        0
Trip_days              0
Year                   0
Month_No               0
Month                  0
Revenue                0
Profit                 0
Groupsize              0
Bkgsource              0
dtype: int64
```

[567]:
```
df.isin(['?']).sum()
```

[567]:
```
bkgref-test            0
Trip ID                0
Destination_Package    0
Destination_Country    0
Category               0
Date_of_trip_from      0
Date_of_trip_to        0
Trip_days              0
Year                   0
Month_No               0
Month                  0
Revenue                0
Profit                 0
Groupsize              0
Bkgsource              0
dtype: int64
```

[568]:
```
per_missing = df.isnull().sum()*100/len(df)
per_missing
```

[568]:
```
bkgref-test            0.0
Trip ID                0.0
Destination_Package    0.0
Destination_Country    0.0
Category               0.0
Date_of_trip_from      0.0
Date_of_trip_to        0.0
Trip_days              0.0
Year                   0.0
Month_No               0.0
Month                  0.0
Revenue                0.0
Profit                 0.0
Groupsize              0.0
```

```
Bkgsource              0.0
dtype: float64
```

[569]:
```
df.dropna(how='any',inplace=True)
```

[570]:
```
df.shape
```

[570]:
```
(16681, 15)
```

[571]:
```
dup = df.duplicated().any()
```

[572]:
```
dup
```

[572]:
```
False
```

[573]:
```
column_data_types = df.dtypes
column_data_types
```

[573]:
```
bkgref-test             object
Trip ID                  int64
Destination_Package     object
Destination_Country     object
Category                object
Date_of_trip_from       object
Date_of_trip_to         object
Trip_days                int64
Year                     int64
Month_No                 int64
Month                   object
Revenue                float64
Profit                 float64
Groupsize                int64
Bkgsource               object
dtype: object
```

[574]:
```
df['Destination_Package'] = df['Destination_Package'].astype(str)
df['Destination_Country'] = df['Destination_Country'].astype(str)
df['Category'] = df['Category'].astype(str)
df['Month'] = df['Month'].astype(str)
df['Bkgsource'] = df['Bkgsource'].astype(str)
df.dtypes
```

[574]:
```
bkgref-test             object
Trip ID                  int64
Destination_Package     object
Destination_Country     object
Category                object
Date_of_trip_from       object
```

```
Date_of_trip_to         object
Trip_days                int64
Year                     int64
Month_No                 int64
Month                   object
Revenue                float64
Profit                 float64
Groupsize                int64
Bkgsource               object
dtype: object
```

[575]:
```
unique_pkgs = sorted(df['Destination_Package'].unique())
unique_pkgs
```

[575]:
```
['Abaco Lodge',
 'Africa - Shooting',
 'Africa Fishing',
 'Agua Boa',
 'Air-FIT',
 'Alaska FIT',
 'Alaska Fishing',
 'Alta - Norway',
 'Amer West Other',
 "Angler's Edge Outfitters-MT",
 'Antarctica-Cruise/Dive',
 'Argentina Fishing-NW Dorado',
 'Argentina Fishing-Patagonia',
 'Argentina Fishing-Rio Grande',
 'Argentina WS',
 'Argentina WS Various',
 'Around the World',
 'Asia FIT',
 'Aust/NZ FIT',
 'Australia Fishing',
 'BGH Africa',
 'BGH North America',
 'BGH South America',
 'BGH South Pacific',
 'Bahamas Misc Saltwater',
 'Bair Lodge',
 'Belize Misc.',
 'Belize River Lodge',
 'Belmond',
 'Big Hole Lodge',
 'Boca Paila',
 'Bolivia WS',
 'Brazil Peacock Bass',
```

```
'Brevyeni',
'Bristol Bay Lodge',
'British Isles',
'Cabo San Lucas/Baja',
'Canada FIT',
'Canada Fishing',
'Cape Santa Maria',
'Caribbean FIT',
'Casa Blanca/Playa Blanca',
'Central America-FIT',
'Chile Misc',
'Christmas Island',
'Classical Cruises',
'Copal Tree or Belcampo Lodge',
'Cordoba WS',
'Costa Rica FIT',
'Crocodile Bay-CR',
'Czech Republic/Hungary',
'Deep Water Cay',
'Denmark Mallards',
'Denmark Pheasants',
'Desroches Island',
'Disney',
'Dufflocq Properties',
'East Africa',
'Eastern-Central Europe',
'Egypt',
'El Pescador',
'Elegant Journeys',
'England WS',
'Europe FIT',
'Flamingo Bay Pacific Charters',
'France',
'France WS',
'French Country Waterways',
'Galapagos-FIT',
'Grand Slam Lodge',
'Great American Steamboat Company',
'Great Britain-Chalk Streams',
'Guatemala',
'H2O Bonefishing',
'Healing Waters Lodge',
'Iceland Trout',
'Iceland-Salmon',
'Ikari House',
'India',
'Intnl. Cruise',
```

```
'Italy',
'Kamalame Cay',
'Los Roques',
'Mangrove Cay',
'Martin Pescador Lodge',
'Merlo',
'Misc. Barges',
'Misc. Central America',
'Misc. Freshwater Fishing',
'Misc. International Fishing',
'Mongolia',
'New Zealand Fishing',
'North Riding Point',
'Norway Fishing-Various',
'Paraguay WS',
'Pittstown Point',
'Ponoi',
'Rio Parismina',
'River Cruises',
'River Plate Wingshooting Entre Rios',
'Russia',
'S A Arg. Misc',
'S. Africa FIT',
'S. Pacific',
'Saltwater Fishing',
'Scandinavia',
'Scotland Pheasants',
'Seychelles - Misc',
'Seychelles Fishing',
'Silver King Lodge-CR',
'South America - FIT',
'South America Fishing',
'Spain-Portugal',
'Spain-WS',
'Spring Ridge Club',
'Tecka Lodge',
'The Delphi Club',
'Three Forks Ranch',
'Three Rivers Ranch',
'Tres Valles',
'Tropic Star',
'Tsimane Lodge',
'Turkey/MiddleEast/Morocco',
'Turneffe Flats Lodge',
'Turneffe Island Lodge',
'UK Barges',
'US FIT',
```

```
'US WS',
'Uruguay WS',
'Yellowstone Valley Ranch',
'Yokanga',
'Yucatan Fishing Misc']
```

0.4 Find Outliers or Mistakes

[576]:
```
df['z_rev'] = stats.zscore(df['Revenue'])
outliers = df[df['z_rev'].abs() > 5]
outliers
```

[576]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	\
1185	B18-47568	825	Around the World	Vari	
3100	B16-40891	841	Aust/NZ FIT	Aust/NZ	
3131	B20-54006	806	Caribbean FIT	Caribbean	
3241	B21-54262	800	US FIT	US	
5174	B11-27597	233	Bahamas Misc Saltwater	Bahamas	
5654	B18-46449	225	Seychelles Fishing	Seychelles	
7850	B12-28054	651	S. Africa FIT	S. Africa	
7851	B13-31322	651	S. Africa FIT	S. Africa	
7965	B15-39926	800	US FIT	US	
13024	B17-43376	800	US FIT	US	
13973	B13-31307	650	East Africa	East Africa	
14589	B18-49047	555	Scotland Pheasants	Scotland	
15080	B15-37106	550	Denmark Pheasants	Denmark	
15081	B16-40510	550	Denmark Pheasants	Denmark	
15082	B17-43799	550	Denmark Pheasants	Denmark	
15083	B18-46943	550	Denmark Pheasants	Denmark	
15084	B19-50036	550	Denmark Pheasants	Denmark	
15143	B18-47915	804	Intnl. Cruise	Vari	
15977	B18-47473	842	Scandinavia	Scandinavia	

	Category	Date_of_trip_from	Date_of_trip_to	Trip_days	Year	Month_No	\
1185	FIT	4/15/2019	5/6/2019	21	2019	4	
3100	FIT	12/17/2016	1/2/2017	16	2016	12	
3131	FIT	12/18/2020	1/29/2021	42	2020	12	
3241	FIT	12/19/2021	1/2/2022	14	2021	12	
5174	SW	2/7/2012	2/14/2012	7	2012	2	
5654	SW	1/26/2019	2/2/2019	7	2019	1	
7850	AFR	7/11/2012	8/21/2012	41	2012	7	
7851	AFR	7/12/2013	8/27/2013	46	2013	7	
7965	FIT	7/22/2017	7/29/2017	7	2017	7	
13024	FIT	5/9/2017	5/25/2017	16	2017	5	
13973	AFR	11/20/2013	12/21/2013	31	2013	11	
14589	WS	10/4/2019	10/10/2019	6	2019	10	
15080	WS	10/10/2015	10/16/2015	6	2015	10	

15081	WS	10/9/2016	10/14/2016	5	2016	10
15082	WS	10/22/2017	10/27/2017	5	2017	10
15083	WS	10/21/2018	10/26/2018	5	2018	10
15084	WS	10/19/2019	10/25/2019	6	2019	10
15143	FIT	9/13/2019	9/22/2019	9	2019	9
15977	FIT	9/22/2018	9/30/2018	8	2018	9

	Month	Revenue	Profit	Groupsize	Bkgsource	z_rev
1185	Apr	253776.0	25376.0	2	Repeat Client	5.714877
3100	Dec	241490.0	44891.0	2	Repeat Client	5.424062
3131	Dec	630000.0	63000.0	14	Repeat Client	14.620260
3241	Dec	312192.0	46827.0	9	Repeat Client	7.097608
5174	Feb	273205.0	262.0	2	Company referral	6.174769
5654	Jan	233496.0	30456.0	1	UK	5.234840
7850	Jul	315435.0	34590.0	1	Company referral	7.174372
7851	Jul	263830.0	26325.0	1	Repeat Client	5.952859
7965	Jul	393805.0	47256.6	1	Repeat Client	9.029423
13024	May	255997.0	25599.0	7	Repeat Client	5.767449
13973	Nov	408064.0	5000.0	1	Repeat Client	9.366940
14589	Oct	236296.0	47259.0	1	Repeat Client	5.301118
15080	Oct	231140.0	24050.0	13	Repeat Client	5.179073
15081	Oct	414250.0	29600.0	16	Repeat Client	9.513365
15082	Oct	283952.0	29600.0	17	Repeat Client	6.429156
15083	Oct	351175.0	29600.0	17	Repeat Client	8.020353
15084	Oct	303086.0	25900.0	1	Repeat Client	6.882066
15143	Sep	265096.0	29908.0	33	Repeat Client	5.982826
15977	Sep	4900000.0	427.4	2	Repeat Client	115.692992

0.4.1 Row #15977 is suspicious –> high z-score and low groupsize

---

0.4.2 Change 4,900,000 to 49,000 which matches the revenue magnitude of other Scandinavia trips booked for 8 days with 2 people more closely.

[546]:
```
filtered_df = df[(df['Destination_Package'] == 'Scandinavia') &
                 (df['Trip_days'] == 8) &
                 (df['Groupsize'] == 2)]

# Display the result
print("Filtered DataFrame:")
filtered_df
```

```
Filtered DataFrame:
```

[546]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	Category
1998	B21-55654	842	Scandinavia	Scandinavia	FIT
4371	B21-56600	842	Scandinavia	Scandinavia	FIT
6557	B18-48623	842	Scandinavia	Scandinavia	FIT
7107	B21-55333	842	Scandinavia	Scandinavia	FIT
7227	B19-50118	842	Scandinavia	Scandinavia	FIT
9044	B19-49556	842	Scandinavia	Scandinavia	FIT
9561	B19-49800	842	Scandinavia	Scandinavia	FIT
10037	B19-49831	842	Scandinavia	Scandinavia	FIT
15977	B18-47473	842	Scandinavia	Scandinavia	FIT

	Date_of_trip_from	Date_of_trip_to	Trip_days	Year	Month_No	Month
1998	8/21/2021	8/29/2021	8	2021	8	Aug
4371	2/12/2022	2/20/2022	8	2022	2	Feb
6557	7/13/2019	7/21/2019	8	2019	7	Jul
7107	7/22/2021	7/30/2021	8	2021	7	Jul
7227	7/25/2019	8/2/2019	8	2019	7	Jul
9044	6/29/2019	7/7/2019	8	2019	6	Jun
9561	6/21/2019	6/29/2019	8	2019	6	Jun
10037	6/29/2019	7/7/2019	8	2019	6	Jun
15977	9/22/2018	9/30/2018	8	2018	9	Sep

	Revenue	Profit	Groupsize	Bkgsource	z_rev
1998	11544.0	1730.0	2	Repeat Client	-0.018859
4371	5070.0	506.0	2	Repeat Client	-0.172101
6557	16490.0	1757.0	2	Repeat Client	0.098215
7107	13300.0	1596.0	2	Repeat Client	0.022707
7227	0.0	0.0	2	Repeat Client	-0.292110
9044	12000.0	2800.0	2	Repeat Client	-0.008065
9561	14780.0	2216.0	2	Web/Internet/Mail	0.057739
10037	13000.0	2400.0	2	Repeat Client	0.015606
15977	49000.0	427.4	2	Repeat Client	115.692992

[545]:
```
df.loc[15977, 'Revenue'] = 49000
df.loc[15977]
```

[545]:
```
bkgref-test                B18-47473
Trip ID                          842
Destination_Package      Scandinavia
Destination_Country      Scandinavia
Category                         FIT
Date_of_trip_from          9/22/2018
Date_of_trip_to            9/30/2018
Trip_days                          8
Year                            2018
Month_No                           9
Month                            Sep
```

```
Revenue                      49000.0
Profit                         427.4
Groupsize                          2
Bkgsource             Repeat Client
z_rev                    115.692992
Name: 15977, dtype: object
```

[548]:
```
no_revprof = df[(df['Profit'] == 0) & (df['Revenue'] == 0)]
no_revprof
```

[548]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	\
85	B20-53726	209	Yucatan Fishing Misc	Mexico	
109	B22-57197	207	Casa Blanca/Playa Blanca	Mexico	
127	B11-26655	206	Boca Paila	Mexico	
145	B18-47493	804	Intnl. Cruise	Vari	
168	B18-47566	210	Grand Slam Lodge	Mexico	
…	…	…	…	…	
16660	B14-35612	380	Ponoi	Russia	
16661	B15-38792	380	Ponoi	Russia	
16662	B16-41756	380	Ponoi	Russia	
16663	B17-44685	380	Ponoi	Russia	
16664	B19-50885	380	Ponoi	Russia	

	Category	Date_of_trip_from	Date_of_trip_to	Trip_days	Year	Month_No	\
85	SW	4/23/2021	5/2/2021	9	2021	4	
109	SW	4/1/2022	4/9/2022	8	2022	4	
127	SW	4/4/2012	4/7/2012	3	2012	4	
145	FIT	4/24/2018	5/23/2018	29	2018	4	
168	SW	4/8/2019	4/13/2019	5	2019	4	
…	…	…	…	…	…	…	
16660	FO	9/13/2014	9/20/2014	7	2014	9	
16661	FO	9/12/2015	9/19/2015	7	2015	9	
16662	FO	9/17/2016	9/24/2016	7	2016	9	
16663	FO	9/16/2017	9/23/2017	7	2017	9	
16664	FO	9/14/2019	9/21/2019	7	2019	9	

	Month	Revenue	Profit	Groupsize	Bkgsource	z_rev	z_prof
85	Apr	0.0	0.0	1	Agent/Outfitter	-0.29211	-0.68316
109	Apr	0.0	0.0	1	Agent/Outfitter	-0.29211	-0.68316
127	Apr	0.0	0.0	1	Company referral	-0.29211	-0.68316
145	Apr	0.0	0.0	4	Repeat Client	-0.29211	-0.68316
168	Apr	0.0	0.0	2	UK	-0.29211	-0.68316
…	…	…	…	…	…	…	…
16660	Sep	0.0	0.0	1	Repeat Client	-0.29211	-0.68316
16661	Sep	0.0	0.0	1	UK	-0.29211	-0.68316
16662	Sep	0.0	0.0	1	UK	-0.29211	-0.68316
16663	Sep	0.0	0.0	1	UK	-0.29211	-0.68316

```
16664   Sep      0.0     0.0          1                UK -0.29211 -0.68316

[795 rows x 17 columns]
```

0.4.3 973 rows have $0 profit recorded! 795 rows have Profit = 0 = Revenue!

0.4.4 Recommendation: Find the profit and revenue imputs to make the analysis more accurate.

For now, we will create a filtered data frame that excludes the rows where profit AND revenue equal 0. This only excludes 4.8% of our data set and we will only use this dataframe for analysis that considers profit as a main factor.

[556]:
```
df_norevprof = df[(df['Profit'] != 0) & (df['Revenue'] != 0)]
df_norevprof.describe()
```

[556]:

	Trip ID	Trip_days	Year	Month_No	Revenue
count	15684.000000	15684.000000	15684.000000	15684.000000	15684.000000
mean	498.158824	9.412841	2016.662586	6.036598	12715.398298
std	266.014408	15.910917	3.135593	3.082895	19111.999427
min	101.000000	0.000000	2010.000000	1.000000	94.820000
25%	238.000000	6.000000	2014.000000	4.000000	4500.000000
50%	380.000000	7.000000	2017.000000	6.000000	7600.000000
75%	806.000000	10.000000	2019.000000	8.000000	14122.500000
max	853.000000	386.000000	2022.000000	12.000000	630000.000000

	Profit	Groupsize	z_rev	z_prof
count	15684.000000	15684.000000	15684.000000	15684.000000
mean	1829.099154	1.986483	0.016190	0.042703
std	2560.307319	1.788133	1.028537	1.016036
min	-400.000000	1.000000	-0.289866	-0.841896
25%	675.000000	1.000000	-0.185593	-0.415292
50%	1145.000000	2.000000	-0.112215	-0.228776
75%	2090.165000	2.000000	0.042176	0.146305
max	63000.000000	37.000000	115.692992	24.317860

We cannot fully rely on the accuracy of revenue and profit for analysis because of these instances above, but we did what we could to make it more reliable.

0.4.5 Checking profit outliers

[547]:
```
df['z_prof'] = stats.zscore(df['Profit'])
outliers = df[df['z_prof'].abs() > 10]
outliers
```

[547]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	Category	\
1861	B17-45684	651	S. Africa FIT	S. Africa	AFR	

1870	B17-43864	800	US FIT	US	FIT
3100	B16-40891	841	Aust/NZ FIT	Aust/NZ	FIT
3131	B20-54006	806	Caribbean FIT	Caribbean	FIT
3228	B11-27224	651	S. Africa FIT	S. Africa	AFR
3240	B20-53540	800	US FIT	US	FIT
3241	B21-54262	800	US FIT	US	FIT
5654	B18-46449	225	Seychelles Fishing	Seychelles	SW
6408	B17-44019	805	Galapagos-FIT	Ecuador	FIT
6751	B15-37444	651	S. Africa FIT	S. Africa	AFR
7183	B13-33668	650	East Africa	East Africa	AFR
7212	B12-28791	150	Alaska Fishing	Alaska	FW
7495	B20-53198	650	East Africa	East Africa	AFR
7850	B12-28054	651	S. Africa FIT	S. Africa	AFR
7965	B15-39926	800	US FIT	US	FIT
8369	B11-26062	804	Intnl. Cruise	Vari	FIT
9805	B21-55283	808	Europe FIT	Europe	FIT
10657	B13-33112	651	S. Africa FIT	S. Africa	AFR
14589	B18-49047	555	Scotland Pheasants	Scotland	WS
15058	B21-56212	843	British Isles	England	FIT
15081	B16-40510	550	Denmark Pheasants	Denmark	WS
15082	B17-43799	550	Denmark Pheasants	Denmark	WS
15083	B18-46943	550	Denmark Pheasants	Denmark	WS
15143	B18-47915	804	Intnl. Cruise	Vari	FIT

	Date_of_trip_from	Date_of_trip_to	Trip_days	Year	Month_No	Month	\
1861	8/3/2018	8/18/2018	15	2018	8	Aug	
1870	8/13/2017	8/26/2017	13	2017	8	Aug	
3100	12/17/2016	1/2/2017	16	2016	12	Dec	
3131	12/18/2020	1/29/2021	42	2020	12	Dec	
3228	12/19/2012	1/1/2013	13	2012	12	Dec	
3240	12/20/2020	12/27/2020	7	2020	12	Dec	
3241	12/19/2021	1/2/2022	14	2021	12	Dec	
5654	1/26/2019	2/2/2019	7	2019	1	Jan	
6408	1/8/2018	1/15/2018	7	2018	1	Jan	
6751	7/20/2015	8/4/2015	15	2015	7	Jul	
7183	7/22/2014	8/12/2014	21	2014	7	Jul	
7212	7/31/2013	8/7/2013	7	2013	7	Jul	
7495	7/9/2021	7/25/2021	16	2021	7	Jul	
7850	7/11/2012	8/21/2012	41	2012	7	Jul	
7965	7/22/2017	7/29/2017	7	2017	7	Jul	
8369	6/17/2012	6/25/2012	8	2012	6	Jun	
9805	6/10/2022	6/24/2022	14	2022	6	Jun	
10657	3/20/2014	4/4/2014	15	2014	3	Mar	
14589	10/4/2019	10/10/2019	6	2019	10	Oct	
15058	10/28/2021	11/7/2021	10	2021	10	Oct	
15081	10/9/2016	10/14/2016	5	2016	10	Oct	
15082	10/22/2017	10/27/2017	5	2017	10	Oct	

```
15083        10/21/2018      10/26/2018          5  2018         10   Oct
15143         9/13/2019       9/22/2019          9  2019          9   Sep

        Revenue   Profit  Groupsize         Bkgsource      z_rev     z_prof
1861   110506.0  27037.0          5     Repeat Client   2.323614  10.046246
1870   172356.0  30276.0         12                UK   3.787630  11.331616
3100   241490.0  44891.0          2     Repeat Client   5.424062  17.131456
3131   630000.0  63000.0         14     Repeat Client  14.620260  24.317860
3228   152555.0  30465.0         11  Company referral   3.318932  11.406619
3240   205796.0  32476.0          8     Repeat Client   4.579170  12.204667
3241   312192.0  46827.0          9     Repeat Client   7.097608  17.899741
5654   233496.0  30456.0          1                UK   5.234840  11.403047
6408   196889.0  27080.0          9     Repeat Client   4.368337  10.063310
6751   132455.0  28964.0          7     Repeat Client   2.843157  10.810960
7183   222676.0  41251.0          8     Repeat Client   4.978726  15.686952
7212   153125.0  30625.0         32  Company referral   3.332424  11.470114
7495   152680.0  27680.0          5     Repeat Client   3.321891  10.301415
7850   315435.0  34590.0          1  Company referral   7.174372  13.043591
7965   393805.0  47256.6          1     Repeat Client   9.029423  18.070224
8369   130738.0  31474.0         22  Company referral   2.802514  11.807032
9805   182412.0  38592.0         18     Repeat Client   4.025660  14.631751
10657  143910.0  30516.0          2     Repeat Client   3.114301  11.426858
14589  236296.0  47259.0          1     Repeat Client   5.301118  18.071177
15058  223483.0  29151.0          1                UK   4.997828  10.885169
15081  414250.0  29600.0         16     Repeat Client   9.513365  11.063351
15082  283952.0  29600.0         17     Repeat Client   6.429156  11.063351
15083  351175.0  29600.0         17     Repeat Client   8.020353  11.063351
15143  265096.0  29908.0         33     Repeat Client   5.982826  11.185578
```

[549]:
```
suspicious_profit = df[df['Profit'] < 0]
suspicious_profit
```

[549]:
```
     bkgref-test  Trip ID Destination_Package Destination_Country Category  \
8846   B16-42376      305       Iceland Trout             Iceland       FW
8847   B16-42377      305       Iceland Trout             Iceland       FW
9796   B16-42375      305       Iceland Trout             Iceland       FW

     Date_of_trip_from Date_of_trip_to  Trip_days  Year  Month_No Month  \
8846          6/8/2017       6/14/2017          6  2017         6   Jun
8847          6/8/2017       6/14/2017          6  2017         6   Jun
9796          6/8/2017       6/16/2017          8  2017         6   Jun

      Revenue  Profit  Groupsize      Bkgsource     z_rev    z_prof
8846   5275.0  -400.0          1  Repeat Client -0.167249 -0.841896
8847   5275.0  -400.0          1  Repeat Client -0.167249 -0.841896
9796   5275.0  -400.0          1  Repeat Client -0.167249 -0.841896
```

0.4.6 -$400 profit for three rows - ask management

Leave alone for now. It is possible they lost money on this particular trip

[550]:
```
df.describe()
```

[550]:

	Trip ID	Trip_days	Year	Month_No	Revenue
count	16681.000000	16681.000000	16681.000000	16681.000000	16681.000000
mean	499.590732	9.689527	2016.672622	6.053534	12049.907307
std	265.996124	17.955906	3.156520	3.084707	18780.003225
min	101.000000	-21.000000	2010.000000	1.000000	0.000000
25%	241.000000	6.000000	2014.000000	4.000000	4050.000000
50%	380.000000	7.000000	2017.000000	6.000000	7190.000000
75%	806.000000	10.000000	2019.000000	8.000000	13666.000000
max	900.000000	386.000000	2022.000000	12.000000	630000.000000

	Profit	Groupsize	z_rev	z_prof
count	16681.000000	16681.000000	1.668100e+04	1.668100e+04
mean	1721.492028	1.970146	4.823989e-17	1.331123e-17
std	2519.972780	1.764510	1.000030e+00	1.000030e+00
min	-400.000000	1.000000	-2.921101e-01	-8.418962e-01
25%	590.000000	1.000000	-1.962448e-01	-4.490231e-01
50%	1072.000000	2.000000	-1.219197e-01	-2.577454e-01
75%	2000.000000	2.000000	3.137001e-02	1.105235e-01
max	63000.000000	37.000000	1.156930e+02	2.431786e+01

[552]:
```
select_metric_columns = ['Trip_days', 'Groupsize']
df[select_metric_columns].describe()
```

[552]:

	Trip_days	Groupsize
count	16681.000000	16681.000000
mean	9.689527	1.970146
std	17.955906	1.764510
min	-21.000000	1.000000
25%	6.000000	1.000000
50%	7.000000	2.000000
75%	10.000000	2.000000
max	386.000000	37.000000

0.4.7 Minimum trip day shouldn't be negative. Find and switch the "end" date with the "start" date.

[554]:
```
suspicious_days = df[df['Trip_days'] < 0]
suspicious_days
```

[554]:

	bkgref-test	Trip ID	Destination_Package	Destination_Country	Category
1252	B12-28997	225	Seychelles Fishing	Seychelles	SW

```
     Date_of_trip_from Date_of_trip_to  Trip_days  Year  Month_No Month  \
1252         4/20/2013       3/30/2013        -21  2013         4   Apr

      Revenue  Profit  Groupsize        Bkgsource     z_rev    z_prof
1252   7295.0  1550.0          1  Frontiers, Ltd. -0.119434 -0.068055
```

[555]:
```
df.loc[1252, 'Date_of_trip_from'] = '3/30/2013'
df.loc[1252, 'Date_of_trip_to'] = '4/20/2103'
df.loc[1252, 'Trip_days'] = 21
df.loc[1252]
```

[555]:
```
bkgref-test                    B12-28997
Trip ID                              225
Destination_Package    Seychelles Fishing
Destination_Country            Seychelles
Category                               SW
Date_of_trip_from                3/30/2013
Date_of_trip_to                  4/20/2103
Trip_days                               21
Year                                  2013
Month_No                                 4
Month                                  Apr
Revenue                             7295.0
Profit                              1550.0
Groupsize                                1
Bkgsource                  Frontiers, Ltd.
z_rev                             -0.119434
z_prof                            -0.068055
Name: 1252, dtype: object
```

0.5 UNIVARIATE ANALYSIS

0.6 Important Findings:

1. Average vacation length (days) = 9.7
2. Most often booked vacation length (days) = 7
3. Average group size = 1.97
4. The most frequent/popular destination package was Ponoi, with a frequency of 1068. The 10 most popular destination packages were Ponoi, US FIT, Europe FIT, Italy, Alaska Fishing, Amer West Other, Intnl. Cruise, Carribean FIT, and France (over 345 counts).
5. Most popular category = FIT
6. Most popular month to travel = June
7. Most popular booking source = Repeat clients

8. The top 10 destination packages faced a decline in bookings in 2020 (during COVID-19). Bookings for the destination packages increased again in 2021 (probably once COVID-19 restrictions started lifting). Before 2020, there was increase in bookings for the destination packages over time, except for Europe FIT in 2018.
9. Ponoi is booked the most during the months of June and September. Alaska Fishing and Amer West Other is popular during August. Europe FIT has highest frequency in May and September. US FIT is consistent throughout all the months.
10. After a spike in company referrals, the most popular booking source (repeat clients) increased and maintained the highest booking source
11. Despite the epidemic, the typical length of the trips hasn't altered much This shows that, while the frequency and type of travel have changed, the people's interest in duration of trip did not change
12. The top three trip categories remained the same before and after covid (FIT, FO, and SW)

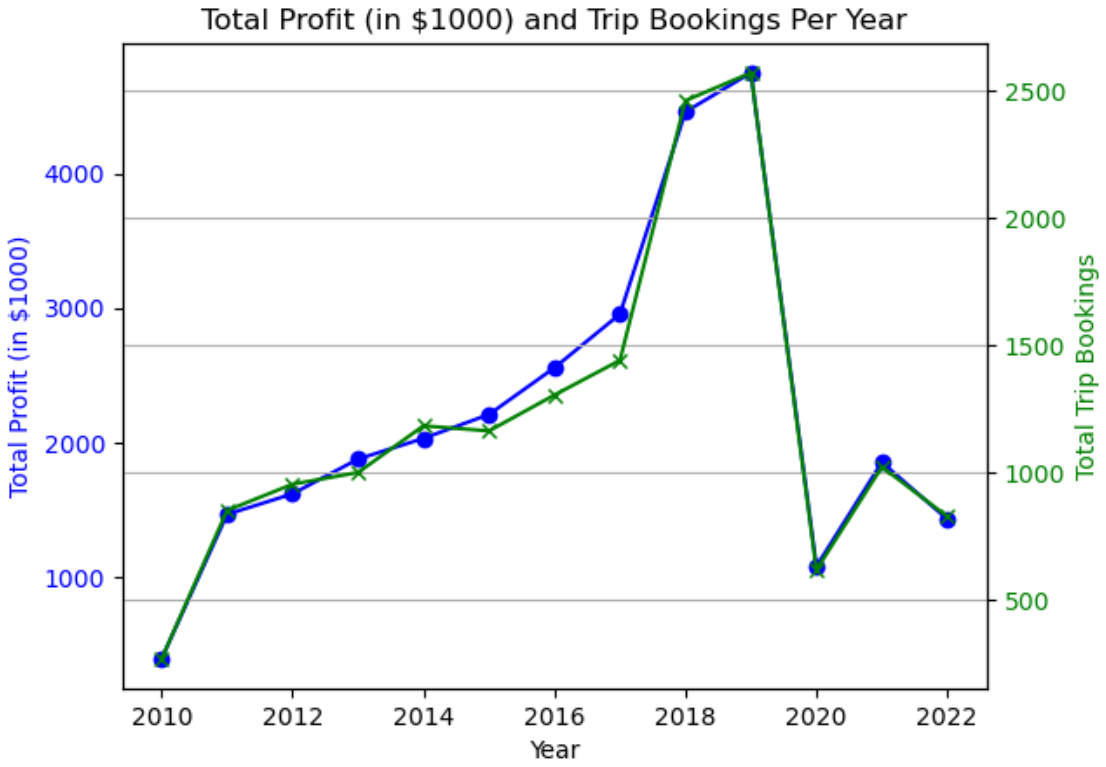
[553]:
```
select_columns = ['Destination_Package', 'Category', 'Trip_days', 'Month',␣
 ↪'Groupsize', 'Bkgsource']
mode_values = df[select_columns].mode().iloc[0]
mode_counts = df[select_columns].apply(lambda col: (col == mode_values[col.
 ↪name]).sum())
total_rows = len(df)
mode_percentages = (mode_counts / total_rows) * 100


modes_and_counts = pd.DataFrame({
    'Mode': mode_values,
    'Count': mode_counts,
    'Percentage (%)': mode_percentages
})


from IPython.display import display, HTML

display(HTML("<h3>Most Popular Categories</h3>"))
display(modes_and_counts)
```

```
<IPython.core.display.HTML object>
```

	Mode	Count	Percentage (%)
Destination_Package	Ponoi	1098	6.582339
Category	FIT	6073	36.406690
Trip_days	7	3798	22.768419
Month	Jun	1944	11.653978
Groupsize	1	7691	46.106349
Bkgsource	Repeat Client	9214	55.236497

[522]:
```
total_profit_per_year = df_norevprof.groupby('Year')['Profit'].sum()
total_profit_per_year_in_thousands = total_profit_per_year / 1000
total_trips_per_year = df_norevprof.groupby('Year').size()
```

```
fig, ax1 = plt.subplots()

ax1.set_xlabel('Year')
ax1.set_ylabel('Total Profit (in $1000)', color='blue')
ax1.plot(total_profit_per_year_in_thousands.index,␣
 ↪total_profit_per_year_in_thousands.values, marker='o', color='blue',␣
 ↪label='Total Profit (in $1000)')
ax1.tick_params(axis='y', labelcolor='blue')

ax2 = ax1.twinx()
ax2.set_ylabel('Total Trip Bookings', color='green')
ax2.plot(total_trips_per_year.index, total_trips_per_year.values, marker='x',␣
 ↪color='green', label='Total Trip Bookings')
ax2.tick_params(axis='y', labelcolor='green')
plt.title('Total Profit (in $1000) and Trip Bookings Per Year')

plt.grid(True)
plt.show()
```



[558]:
```
total_profit_2019 = df[df['Year'] == 2019]['Profit'].sum()
total_profit_2020 = df[df['Year'] == 2020]['Profit'].sum()
total_profit_2022 = df[df['Year'] == 2022]['Profit'].sum()

if total_profit_2019 != 0:
    percentage_loss_1920 = ((total_profit_2019 - total_profit_2020) /␣
 ↪total_profit_2019) * 100
else:
    percentage_loss = 0

if total_profit_2020 != 0:
    percentage_gain_2220 = ((total_profit_2022 - total_profit_2020) /␣
 ↪total_profit_2020) * 100
else:
    percentage_gain_2220 = 0

if total_profit_2019 != 0:
    percentage_loss_1922 = ((total_profit_2019 - total_profit_2022) /␣
 ↪total_profit_2019) * 100
else:
    percentage_loss = 0

print(f"The percentage of profits lost from 2019 to 2020 is␣
 ↪{percentage_loss_1920:.2f}%.")
print(f"The percentage of profits lost from 2019 to 2022 is␣
 ↪{percentage_loss_1922:.2f}%.")
print(f"The percentage of profits gained from 2020 to 2022 is␣
 ↪{percentage_loss_2220:.2f}%.")
print(f'The total profit in 2019 is {total_profit_2019}')
print(f'The total profit in 2020 is {total_profit_2020}')
print(f'The total profit in 2022 is {total_profit_2022}')
```

```
The percentage of profits lost from 2019 to 2020 is 77.17%.
The percentage of profits lost from 2019 to 2022 is 69.80%.
The percentage of profits gained from 2020 to 2022 is 32.29%.
The total profit in 2019 is 4747299.284
The total profit in 2020 is 1083698.2600000002
The total profit in 2022 is 1433646.9899999998
```

[523]:
```
metric_columns = ['Trip_days', 'Revenue', 'Profit', 'Groupsize']
df[metric_columns].describe()
```

[523]:

	Trip_days	Revenue	Profit	Groupsize
count	16681.000000	16681.000000	16681.000000	16681.000000
mean	9.692045	12049.907307	1721.492028	1.970146
std	17.954547	18780.003225	2519.972780	1.764510
min	0.000000	0.000000	-400.000000	1.000000

25%	6.000000	4050.000000	590.000000	1.000000
50%	7.000000	7190.000000	1072.000000	2.000000
75%	10.000000	13666.000000	2000.000000	2.000000
max	386.000000	630000.000000	63000.000000	37.000000

[524]:
```
total_profit_per_year = df.groupby('Year')['Profit'].sum()

# Find the year with the highest total profit
year_with_highest_profit = total_profit_per_year.idxmax()
highest_profit_value = total_profit_per_year.max()

# Display the result
print(f"The year with the highest total profit is {year_with_highest_profit}␣
 ↪with a profit of {highest_profit_value:.2f}.")
```

The year with the highest total profit is 2019 with a profit of 4747299.28.

[525]:
```
total_profit_2020 = df[df['Year'] == 2020]['Profit'].sum()

# Display the result
print(f"The total profit in 2020 was {total_profit_2020:.2f}.")
```

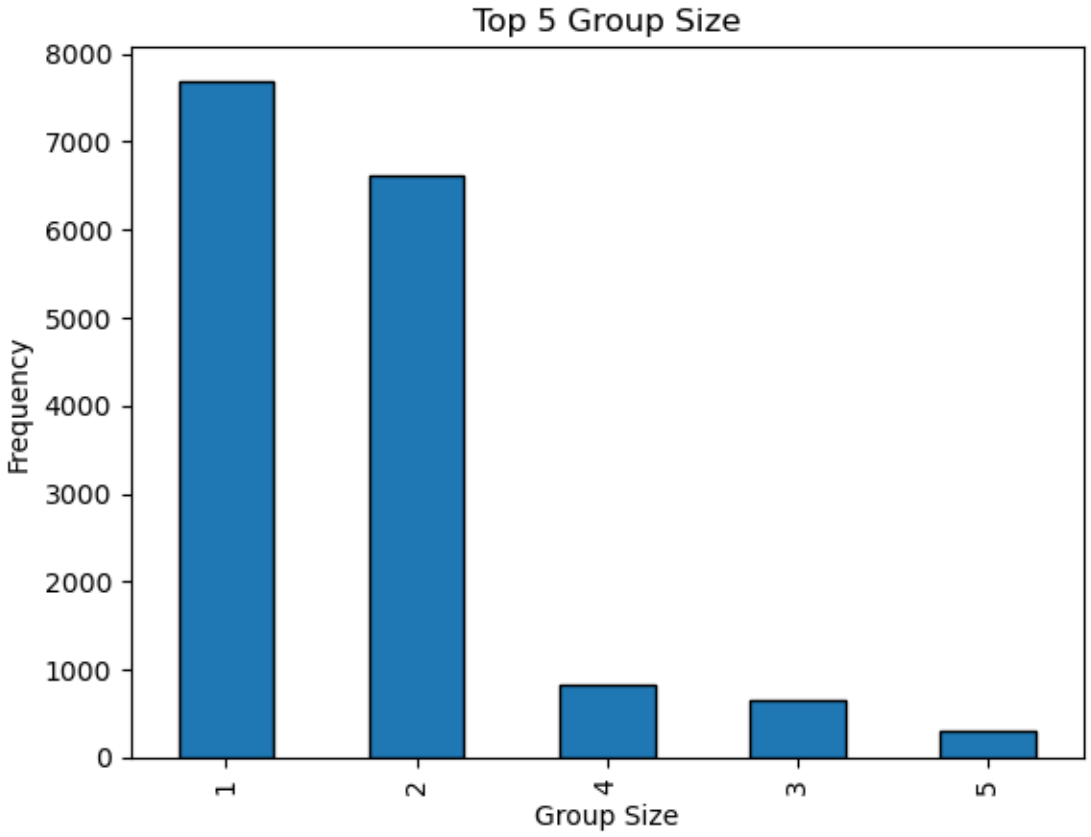
The total profit in 2020 was 1083698.26.

[526]:
```
total_profit_2019 = df[df['Year'] == 2019]['Profit'].sum()
total_profit_2020 = df[df['Year'] == 2020]['Profit'].sum()

# Calculate the percentage loss
if total_profit_2019 != 0:
    percentage_loss = ((total_profit_2019 - total_profit_2020) /␣
 ↪total_profit_2019) * 100
else:
    percentage_loss = 0

# Display the result
print(f"The percentage of profits lost from 2019 to 2020 is {percentage_loss:.
 ↪2f}%.")
```

The percentage of profits lost from 2019 to 2020 is 77.17%.

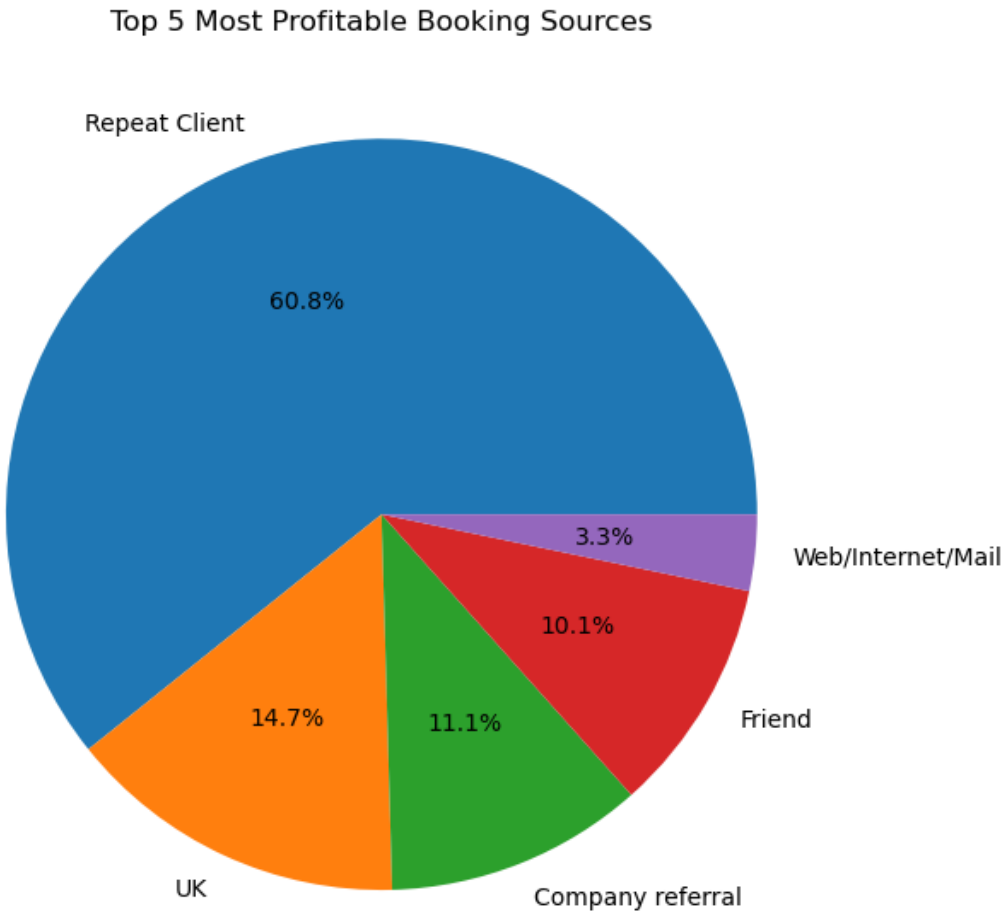
[527]:
```
group_size_counts = df['Groupsize'].value_counts().head()

# Plotting the bar chart
group_size_counts.plot(kind='bar', edgecolor='black')

# Adding labels and title
plt.title('Top 5 Group Size')
plt.xlabel('Group Size')
plt.ylabel('Frequency')
```

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



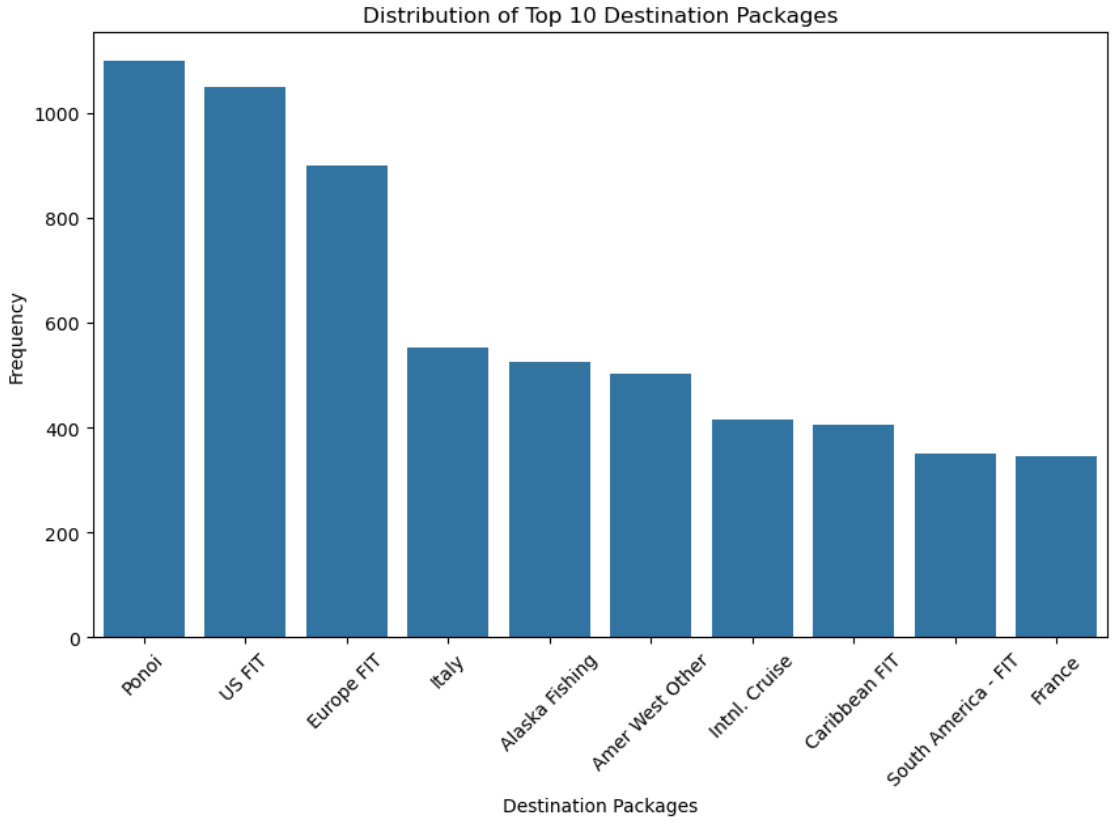
[482]:
```
# Group by 'Destination_Package' and sum the total profit for each package␣
 ↪pre-COVID and post-COVI
bkgsource_profits = df_norevprof.groupby('Bkgsource')['Profit'].sum().
 ↪reset_index()

# Sort by the highest profit and select the top 5 destination packages for␣
 ↪pre-COVID and post-COVID
top_5_sources = bkgsource_profits.nlargest(5, 'Profit')

# Plotting the pie charts for pre-COVID and post-COVID
plt.figure(figsize=(14, 6))

# Pre-COVID Pie Chart
plt.subplot(1, 2, 1)
plt.pie(top_5_sources['Profit'], labels=top_5_sources['Bkgsource'], autopct='%1.
 ↪1f%%')
```

```
plt.title('Top 5 Most Profitable Booking Sources')

# Show both pie charts
plt.tight_layout()
plt.show()
```



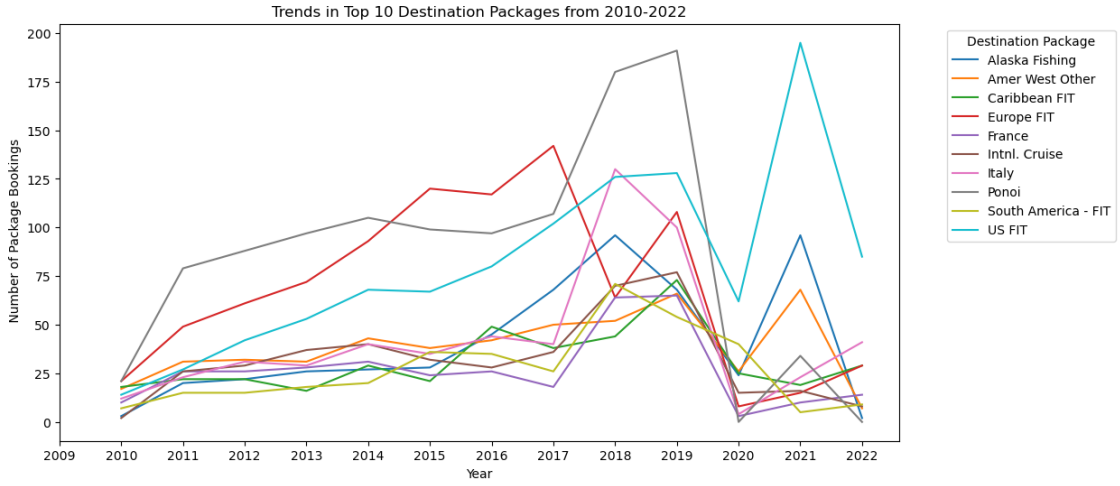
0.7 Most Popular Package

[483]:
```
top_packages = df['Destination_Package'].value_counts().nlargest(10).index
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Destination_Package', order=top_packages)
plt.xlabel('Destination Packages')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.title('Distribution of Top 10 Destination Packages')
plt.show()
```



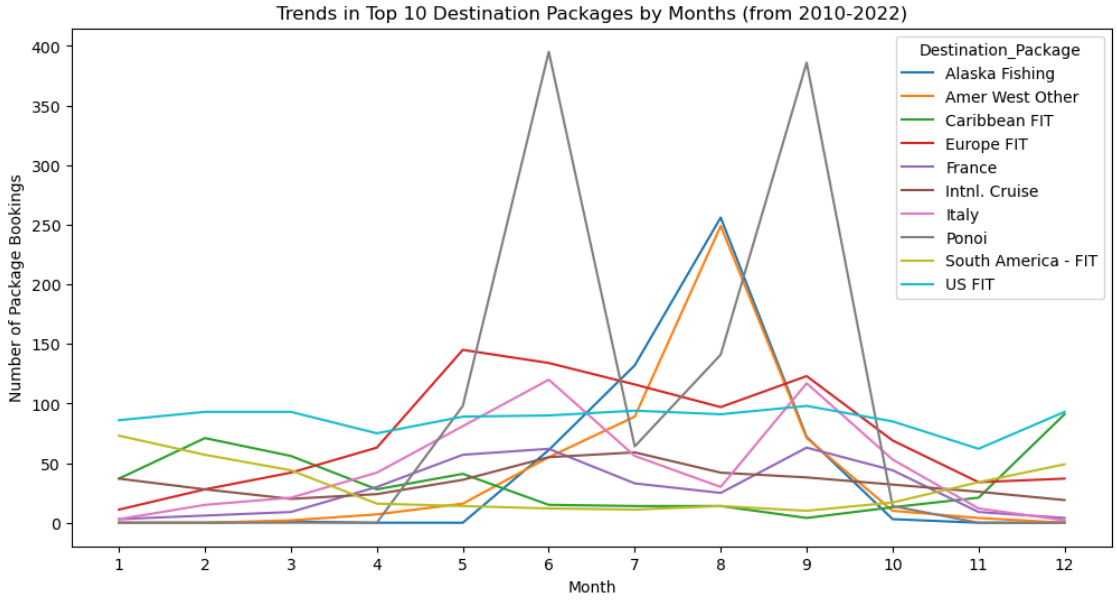
[484]:
```
top_10_packages = df['Destination_Package'].value_counts().nlargest(10).index
df_top_10 = df[df['Destination_Package'].isin(top_10_packages)]

top_packages_over_time = df_top_10.groupby(['Year', 'Destination_Package']).
 ↪size().unstack().fillna(0)
top_packages_over_time.plot(figsize=(12, 6), kind='line')
plt.title('Trends in Top 10 Destination Packages from 2010-2022')
plt.xlabel('Year')
plt.ylabel(' Number of Package Bookings')
plt.legend(title='Destination Package', bbox_to_anchor=(1.05, 1), loc='upper␣
 ↪left')
```

```
plt.xticks(ticks= list(range(2009,2023)))
plt.show()
```



All the top 10 destination packages faced a decline in bookings during 2020 (when COVID-19 happened). Number of bookings for the packages increased again in 2021 for Ponoi (grey), Amer West Other (orange), Alaska Fishing (dark blue) and US FIT (light blue). In general, there was an increase in bookings for the top 10 destination over the years, until 2020.

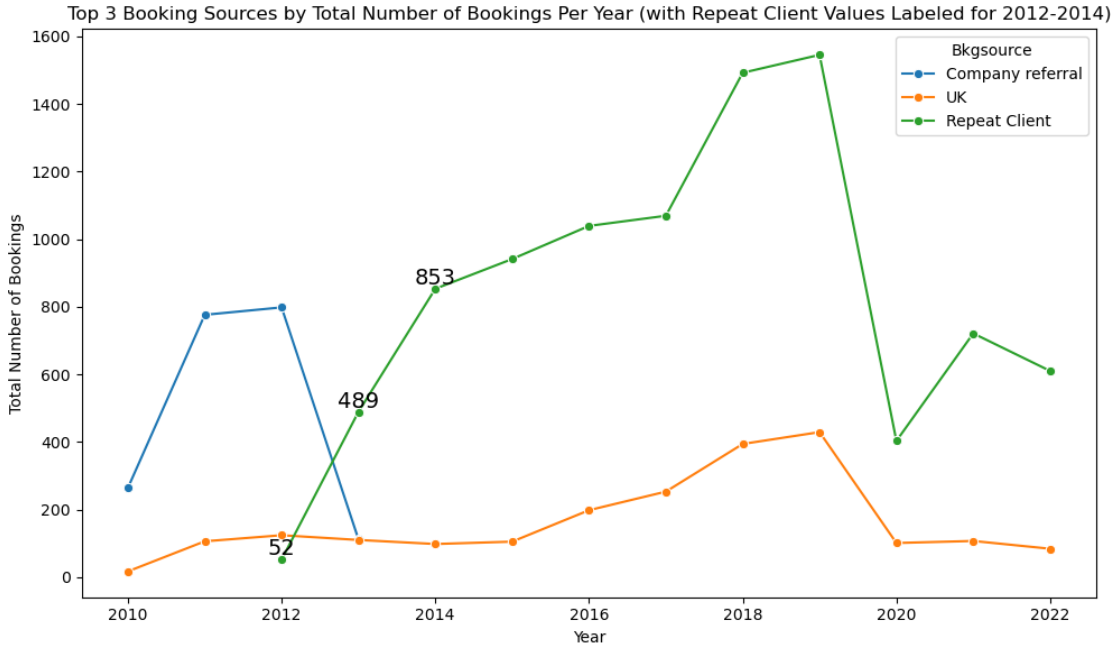
[485]:
```
top_packages_over_month = df_top_10.groupby(['Month_No',␣
↪'Destination_Package']).size().unstack().fillna(0)
top_packages_over_month.plot(figsize=(12, 6), kind='line')
plt.title('Trends in Top 10 Destination Packages by Months (from 2010-2022)')
plt.xlabel('Month')
plt.ylabel('Number of Package Bookings')
plt.xticks(ticks= list(range(1,13)))
plt.show()
```



Ponoi (grey) is popular during the months of June and September. Alaska Fishing (dark blue) and Amer West Other (orange) is popular during August. Europe FIT (red) has highest frequency in May and September. US FIT (light blue) is consistent throughout all months. Italy (pink) also had more bookings in June and September, as Ponoi.

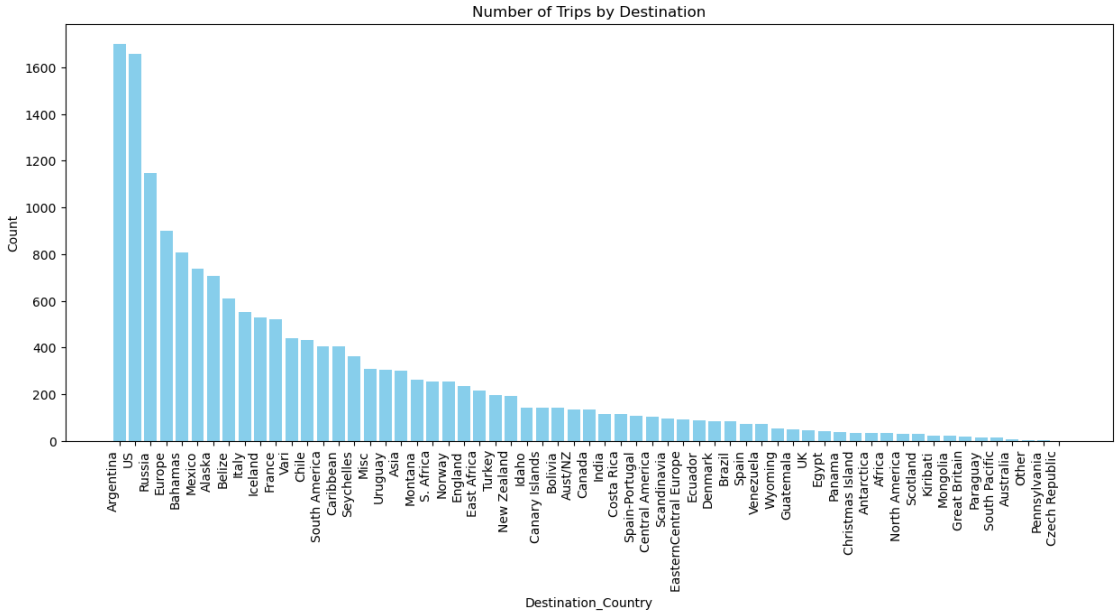
[486]:
```
df['IsRepeatClient'] = df['Bkgsource'].apply(lambda x: 1 if x == 'Repeat␣
 ↪Client' else 0)
bookings_per_year = df.groupby(['Year', 'Bkgsource']).size().
 ↪reset_index(name='Total_Bookings')
top_3_sources = bookings_per_year.groupby('Bkgsource')['Total_Bookings'].sum().
 ↪nlargest(3).index
top_3_data = bookings_per_year[bookings_per_year['Bkgsource'].
 ↪isin(top_3_sources)]
repeat_clients = df[(df['Bkgsource'] == 'Repeat Client') & (df['Year'].
 ↪between(2012, 2014))]
repeat_clients_per_year = repeat_clients.groupby('Year').size().
 ↪reset_index(name='Repeat_Bookings')
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year', y='Total_Bookings', hue='Bkgsource', data=top_3_data,␣
 ↪marker='o')
for _, row in repeat_clients_per_year.iterrows():
    plt.text(row['Year'], row['Repeat_Bookings'], f"{row['Repeat_Bookings']}",␣
 ↪color='black', size=14, ha='center', va='bottom')
plt.title('Top 3 Booking Sources by Total Number of Bookings Per Year (with␣
 ↪Repeat Client Values Labeled for 2012-2014)')
plt.xlabel('Year')
```

```
plt.ylabel('Total Number of Bookings')
plt.tight_layout()
plt.show()
```



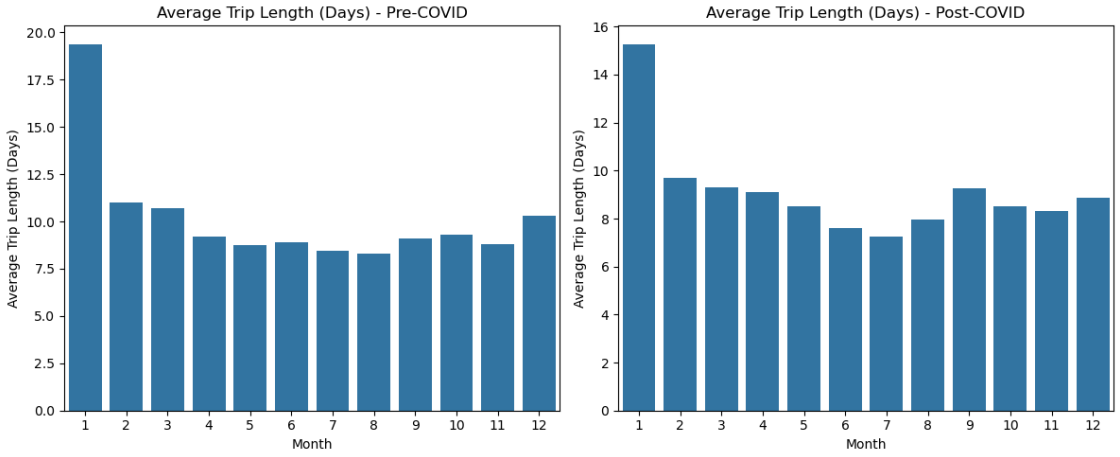
[487]:
```
df = pd.read_csv("travel_data_final.csv")
def create_bar_plot(data, title, x_variable, y_variable, save_file):


    counts = data[x_variable].value_counts()


    plt.figure(figsize=(15, 6))  # Adjust the figure size as needed
    plt.bar(counts.index, counts.values, color='skyblue')


    plt.xlabel(x_variable)
    plt.ylabel(y_variable)
    plt.title(title)


    plt.xticks(rotation=90, ha='right')

    plt.savefig(f'{save_file}.png')
    plt.show()
```

```
create_bar_plot(df, 'Number of Trips by Destination', 'Destination_Country',␣
 ↪'Count', 'destination_trips')
```



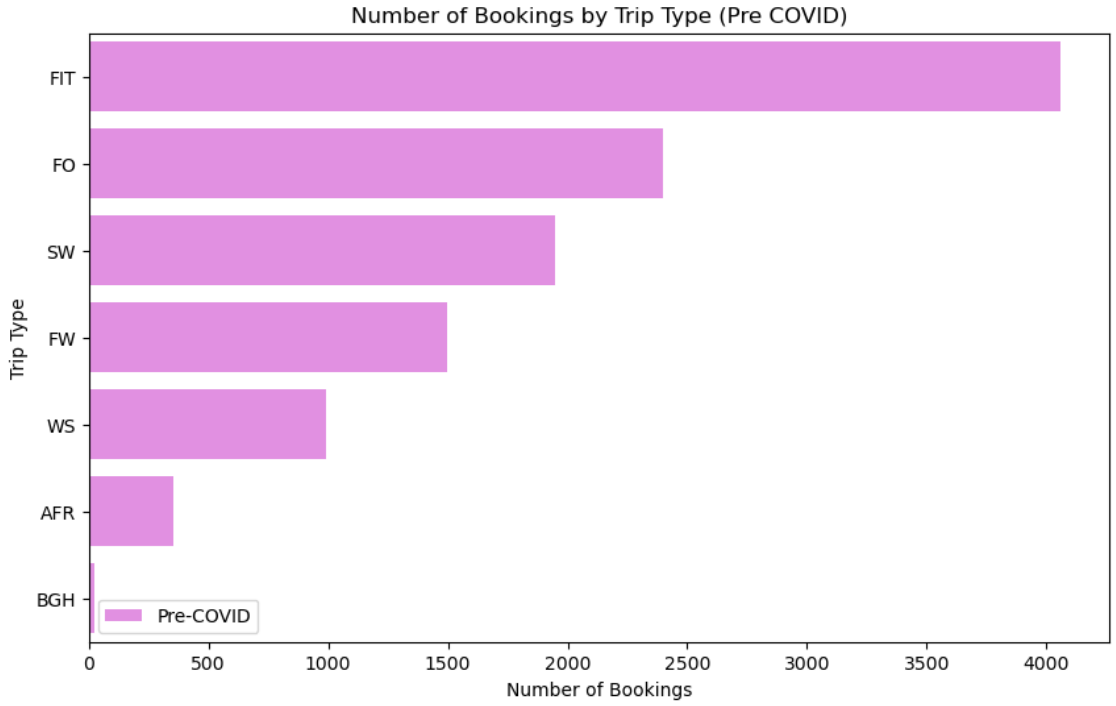
[577]:
```
df['Year'] = pd.to_datetime(df['Date_of_trip_from']).dt.year
df['Month'] = pd.to_datetime(df['Date_of_trip_from']).dt.month


pre_covid_data = df[df['Year'] < 2019]

post_covid_data = df[df['Year'] >= 2019]

pre_covid_avg_trip_length = pre_covid_data.groupby('Month')['Trip_days'].mean()
post_covid_avg_trip_length = post_covid_data.groupby('Month')['Trip_days'].
 ↪mean()

fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Pre-COVID Data
sns.barplot(x=pre_covid_avg_trip_length.index, y=pre_covid_avg_trip_length.
 ↪values, ax=axes[0])
axes[0].set_title('Average Trip Length (Days) - Pre-COVID')
axes[0].set_xlabel('Month')
axes[0].set_ylabel('Average Trip Length (Days)')
```

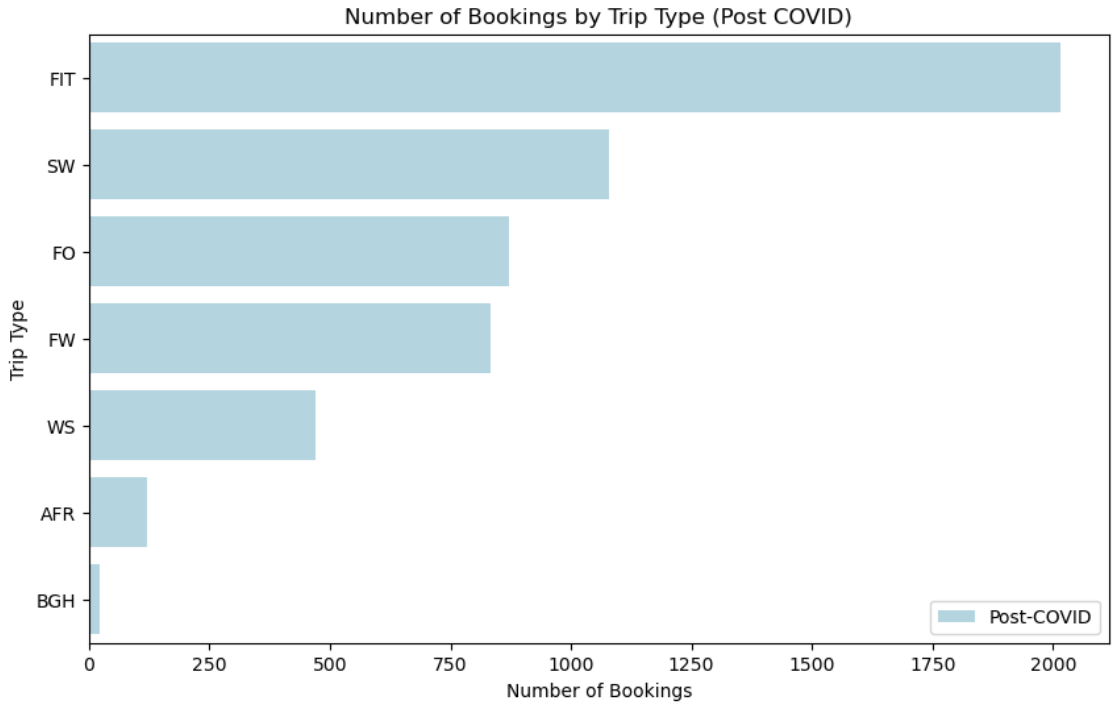
```
# Post-COVID Data
sns.barplot(x=post_covid_avg_trip_length.index, y=post_covid_avg_trip_length.
 ↪values, ax=axes[1])
axes[1].set_title('Average Trip Length (Days) - Post-COVID')
axes[1].set_xlabel('Month')
axes[1].set_ylabel('Average Trip Length (Days)')

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



[440]:
```
df['Year'] = pd.to_datetime(df['Date_of_trip_from']).dt.year
df['Month'] = pd.to_datetime(df['Date_of_trip_from']).dt.month


pre_covid_data = df[df['Year'] < 2019]

post_covid_data = df[df['Year'] >= 2019]

pre_covid_bookings = pre_covid_data.groupby('Category')['Trip ID'].count().
 ↪reset_index()
post_covid_bookings = post_covid_data.groupby('Category')['Trip ID'].count().
 ↪reset_index()



# Pre COVID Chart
fig, ax = plt.subplots(figsize=(10, 6))
pre_covid_bookings = pre_covid_bookings.sort_values(by='Trip ID',␣
 ↪ascending=False)
```

```
sns.barplot(x='Trip ID', y='Category', data=pre_covid_bookings,␣
 ↪label='Pre-COVID', color='violet', ax=ax)

ax.set_xlabel('Number of Bookings')
ax.set_ylabel('Trip Type')
ax.set_title('Number of Bookings by Trip Type (Pre COVID)')
ax.legend()
plt.show()

# Post COVID Chart
fig, ax = plt.subplots(figsize=(10, 6))
post_covid_bookings = post_covid_bookings.sort_values(by='Trip ID',␣
 ↪ascending=False)
sns.barplot(x='Trip ID', y='Category', data=post_covid_bookings,␣
 ↪label='Post-COVID', color='lightblue', ax=ax)

ax.set_xlabel('Number of Bookings')
ax.set_ylabel('Trip Type')
ax.set_title('Number of Bookings by Trip Type (Post COVID)')
ax.legend()
plt.show()
```





---

1 BIVARIATE ANALYSIS

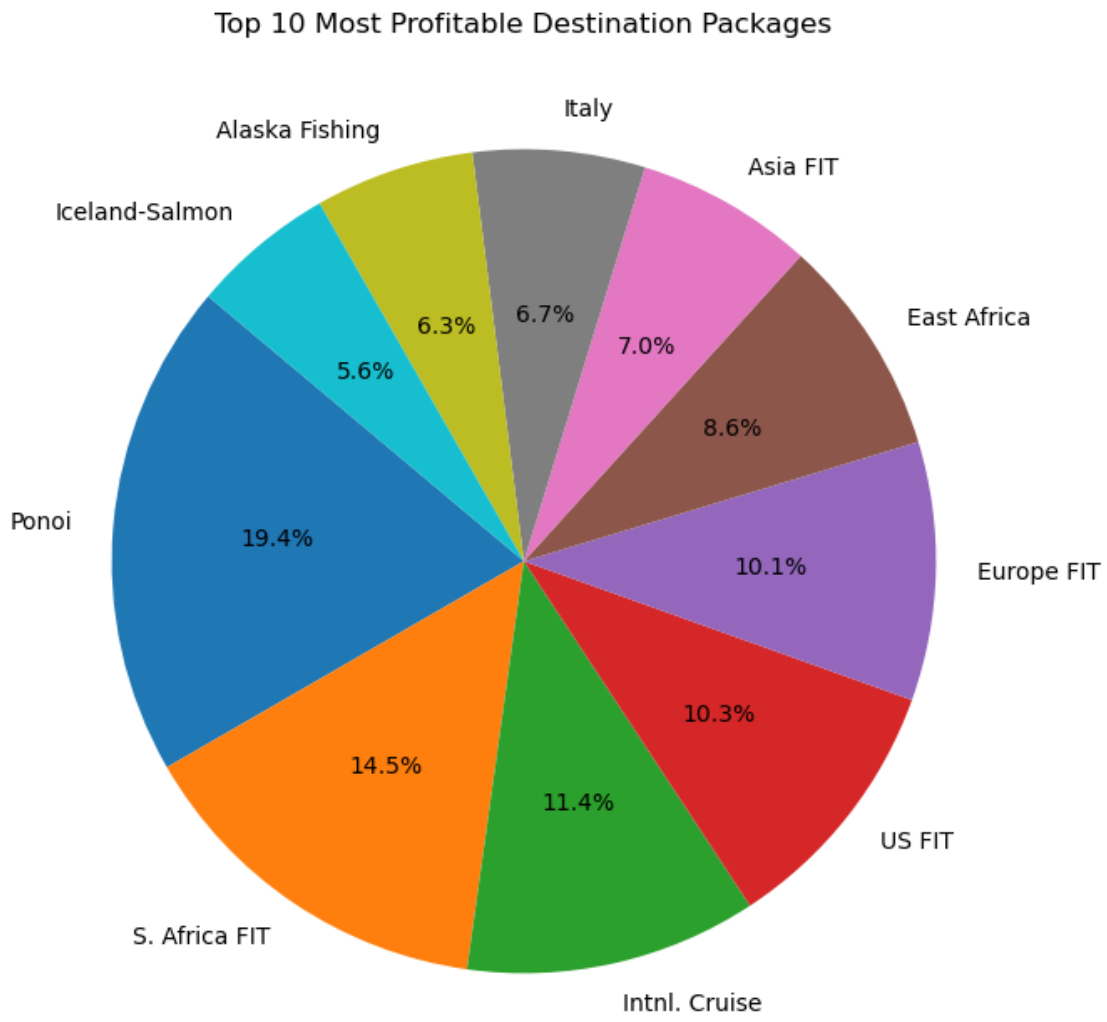
1.1 Important Findings

1. Most profitable package = Ponoi
2. Most profitable booking source = Repeat Clients
3. Most popular Destination Package of Repeat Clients = US FIT
4. Most profitable category is AFR (Africa?)
5. Same day trips are preferred with groups of two or fewer

1.1.1 Profit vs Destination Package

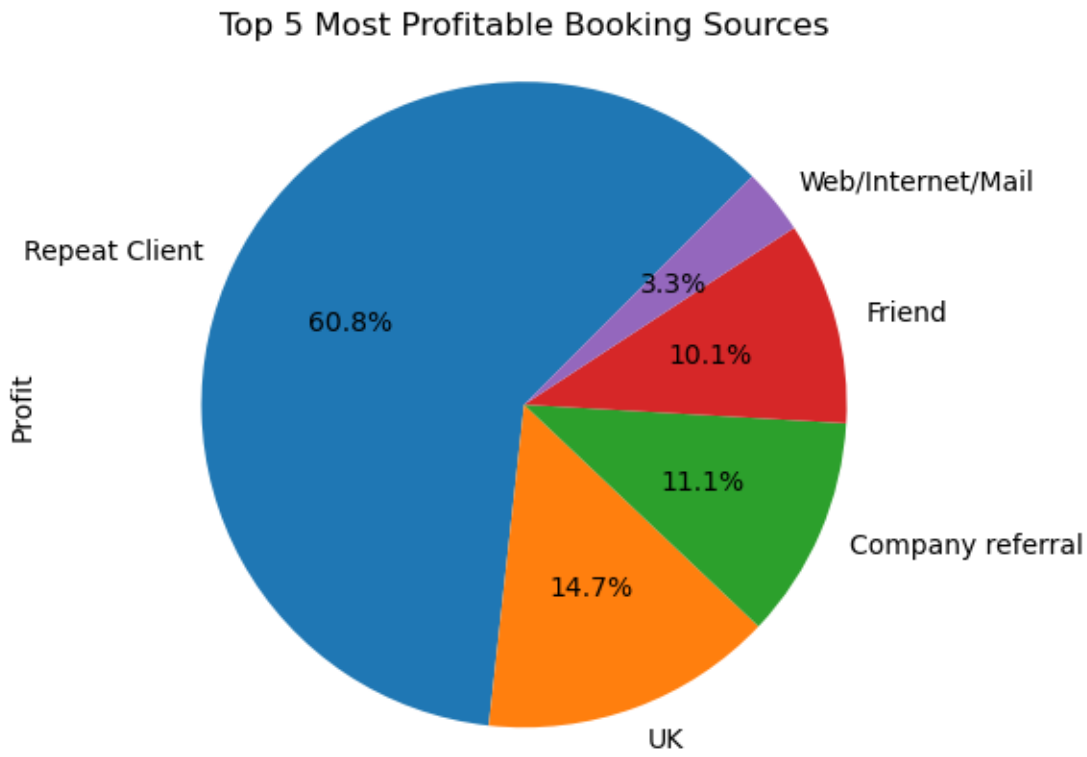
[441]:
```
profit_by_package = df_norevprof.groupby('Destination_Package')['Profit'].sum()

# Sort the packages by profit in descending order and get the top 10
top_10_profitable_packages = profit_by_package.sort_values(ascending=False).
  ↪head(10)

# Create a pie chart
plt.figure(figsize=(8, 8))
```

```
plt.pie(top_10_profitable_packages, labels=top_10_profitable_packages.index,␣
↪autopct='%1.1f%%', startangle=140)
plt.title('Top 10 Most Profitable Destination Packages')
plt.show()
```



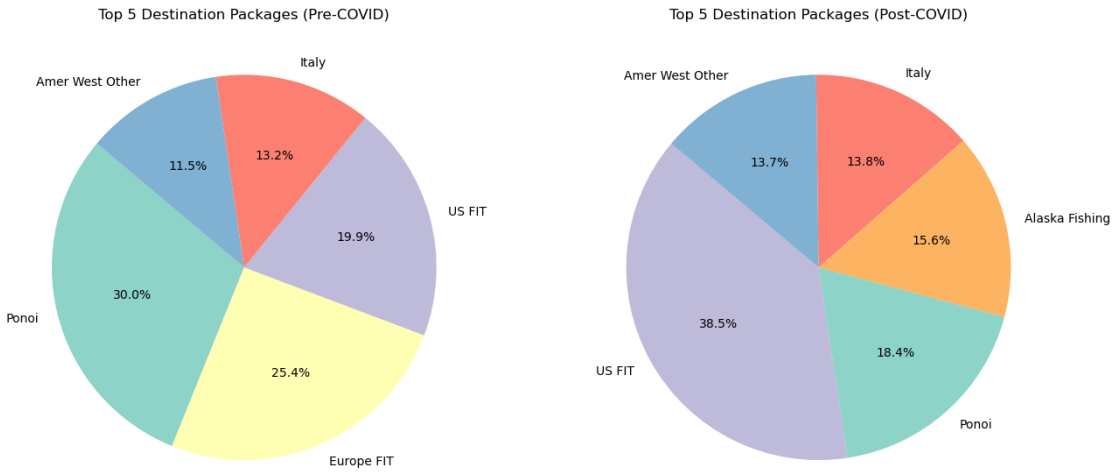
[442]:
```
bkgsource_profit = df_norevprof.groupby('Bkgsource')['Profit'].sum()

top_5_bkgsource_profit = bkgsource_profit.sort_values(ascending=False).head(5)
top_5_bkgsource_profit.plot(
    kind='pie', autopct='%1.1f%%', startangle=45)

plt.title('Top 5 Most Profitable Booking Sources')
plt.axis('equal')
```

```
plt.show()
```



[443]:
```
df['Year'] = pd.to_datetime(df['Date_of_trip_from']).dt.year
pre_covid_data = df[df['Year'] < 2019]
post_covid_data = df[df['Year'] >= 2019]
pre_covid_trips = pre_covid_data.groupby('Destination_Package').size().
 ↪reset_index(name='Trips_PreCOVID')
post_covid_trips = post_covid_data.groupby('Destination_Package').size().
 ↪reset_index(name='Trips_PostCOVID')
top_5_pre_covid = pre_covid_trips.nlargest(5, 'Trips_PreCOVID')
top_5_post_covid = post_covid_trips.nlargest(5, 'Trips_PostCOVID')
all_top_destinations = pd.concat([top_5_pre_covid,␣
 ↪top_5_post_covid])['Destination_Package'].unique()

colors = plt.get_cmap('Set3')(range(len(all_top_destinations)))
color_map = {destination: colors[i] for i, destination in␣
 ↪enumerate(all_top_destinations)}
pre_covid_colors = [color_map[dest] for dest in␣
 ↪top_5_pre_covid['Destination_Package']]
post_covid_colors = [color_map[dest] for dest in␣
 ↪top_5_post_covid['Destination_Package']]
```

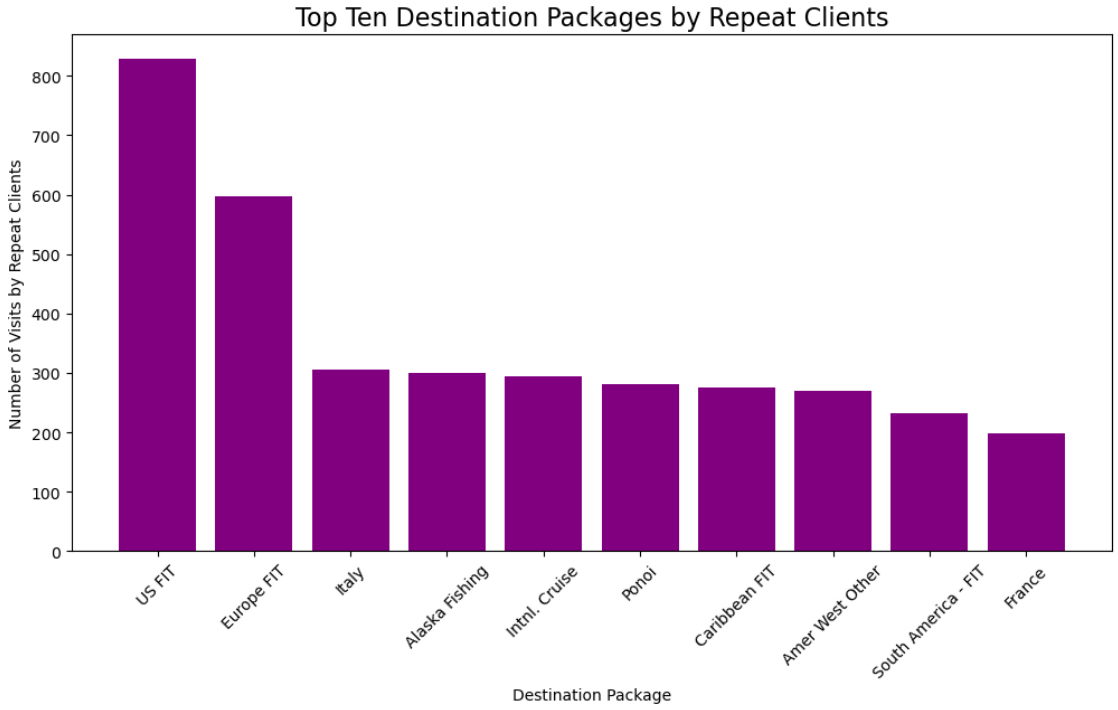
```
plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
plt.pie(top_5_pre_covid['Trips_PreCOVID'],␣
 ↪labels=top_5_pre_covid['Destination_Package'],
        autopct='%1.1f%%', startangle=140, colors=pre_covid_colors)
plt.title('Top 5 Destination Packages (Pre-COVID)')
plt.subplot(1, 2, 2)
plt.pie(top_5_post_covid['Trips_PostCOVID'],␣
 ↪labels=top_5_post_covid['Destination_Package'],
        autopct='%1.1f%%', startangle=140, colors=post_covid_colors)
plt.title('Top 5 Destination Packages (Post-COVID)')
plt.tight_layout()
plt.show()
```



1.1.2 Top Booking Source (Repeat Clients) vs Destination Package

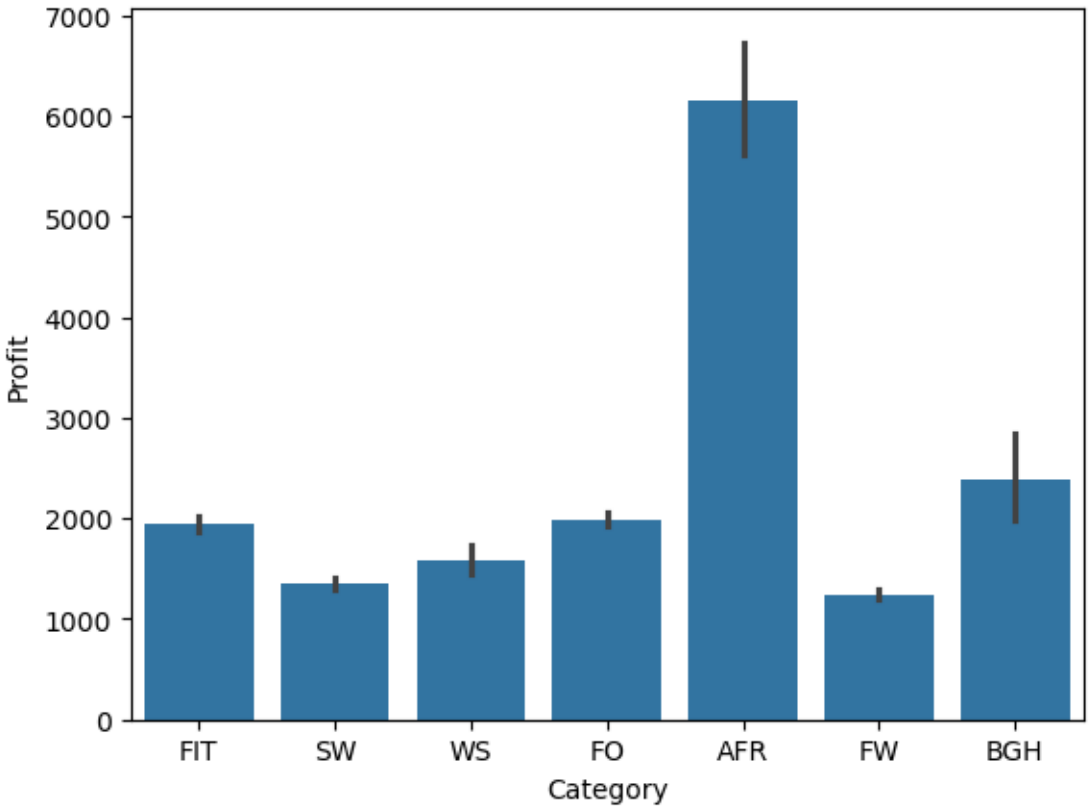
[578]:
```
repeat_clients_df = df[df['Bkgsource'] == 'Repeat Client']

destinationpkg_counts = repeat_clients_df['Destination_Package'].value_counts().
 ↪head(10)

plt.figure(figsize=(12, 6))
plt.bar(destinationpkg_counts.index, destinationpkg_counts.values,␣
 ↪color='purple')

plt.xlabel('Destination Package')
plt.ylabel('Number of Visits by Repeat Clients')
plt.title('Top Ten Destination Packages by Repeat Clients', size=16)
plt.xticks(rotation=45)
```

```
plt.show()
```

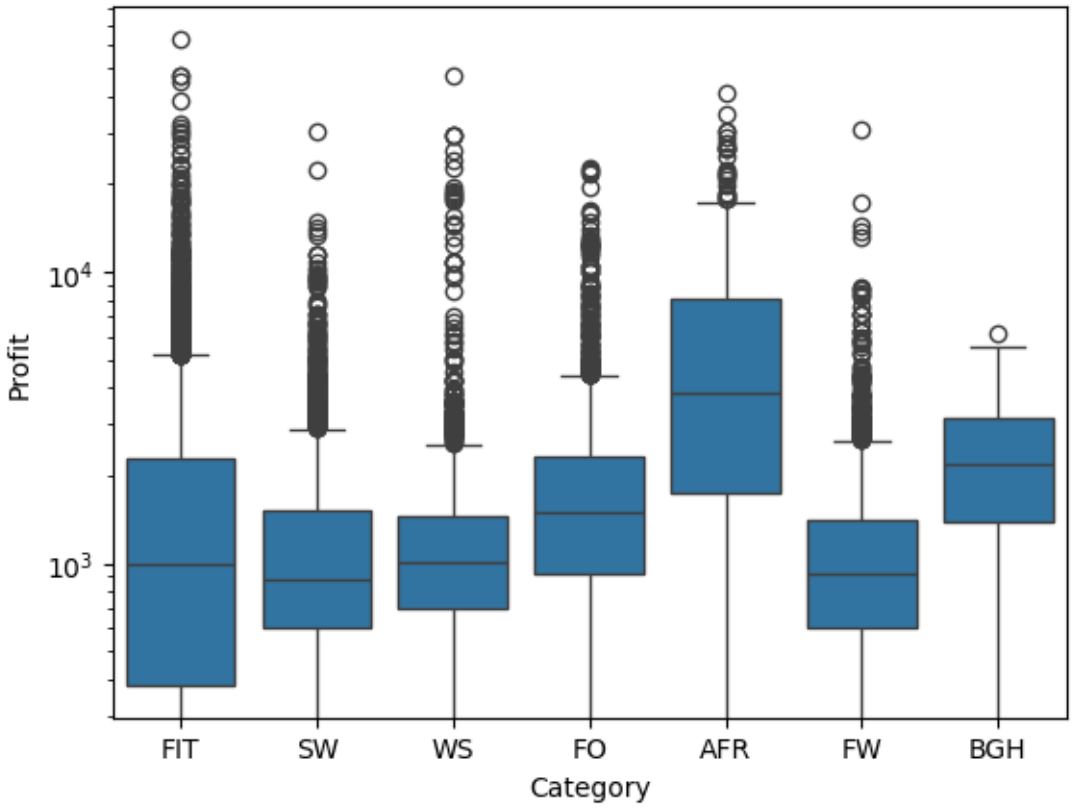


1.1.3 Profit vs. Category

[445]:
```
sns.barplot(x='Category', y='Profit', data=df_norevprof)
plt.show()
```

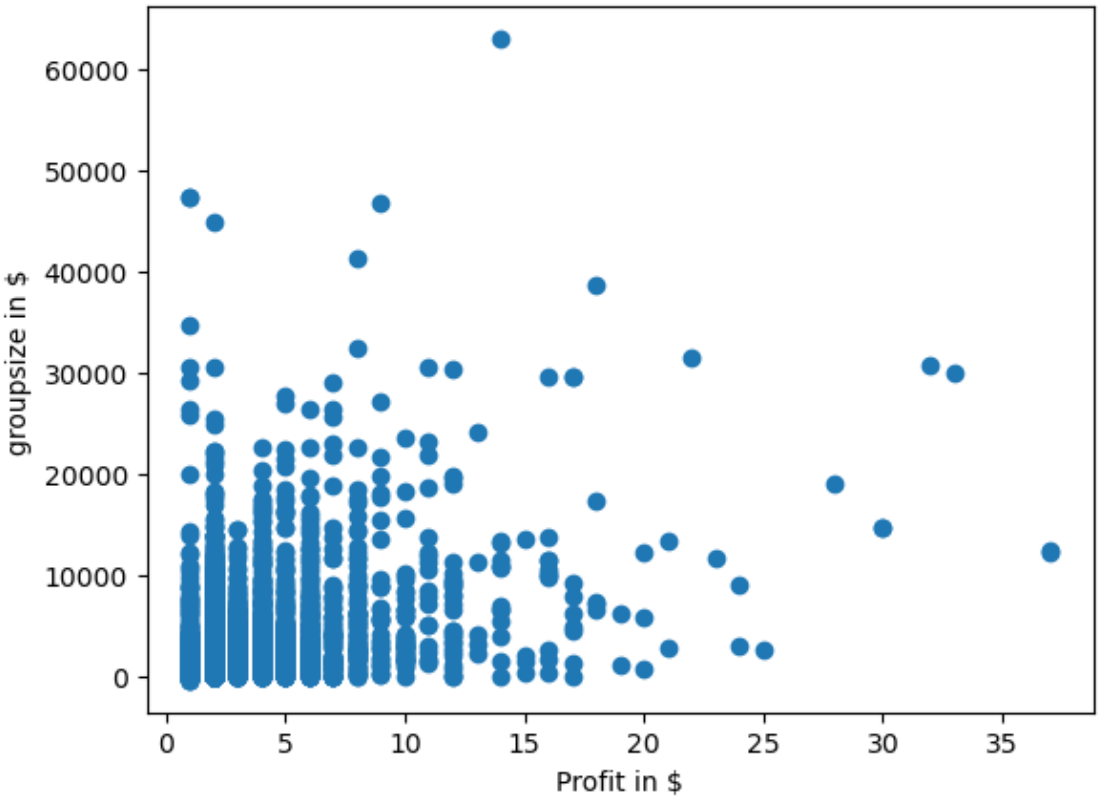


[446]:
```
sns.boxplot(x='Category', y='Profit', data=df)
plt.yscale('log')
plt.show()
```

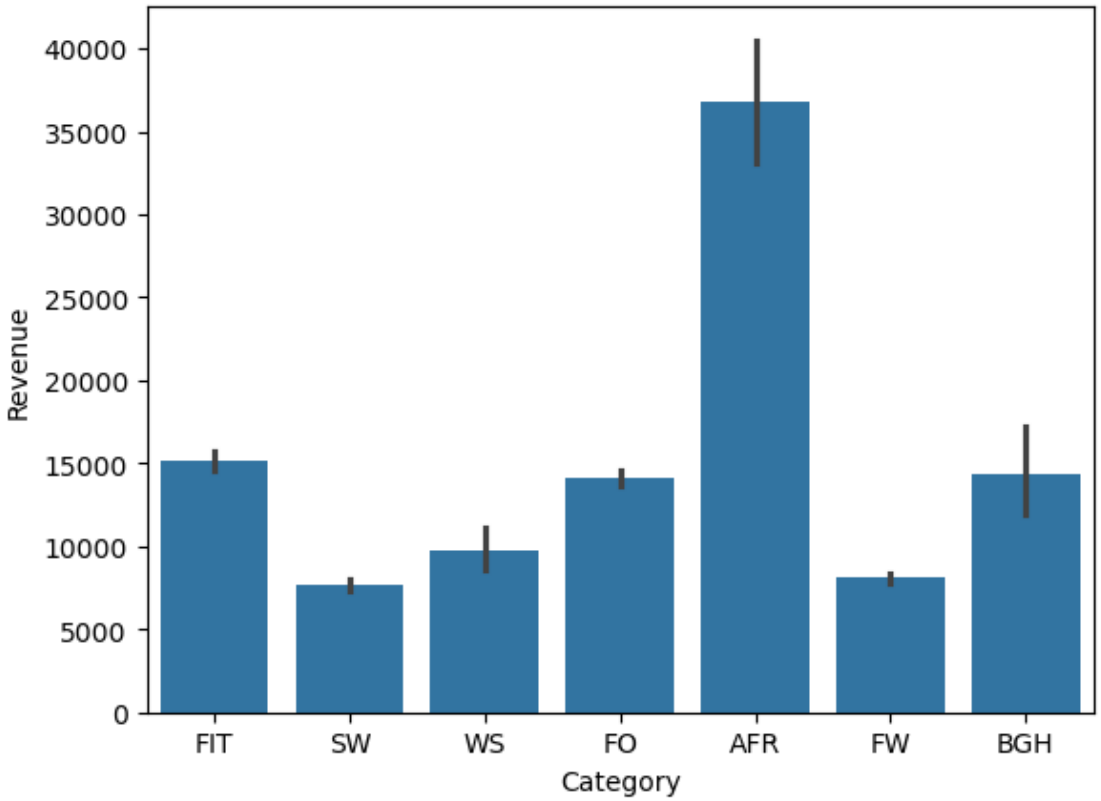


[107]:
```
plt.scatter(df['groupsize'],df['Profit'],)
plt.xlabel('Profit in $')
plt.ylabel('groupsize in $')
```

[107]:
```
Text(0, 0.5, 'groupsize in $')
```

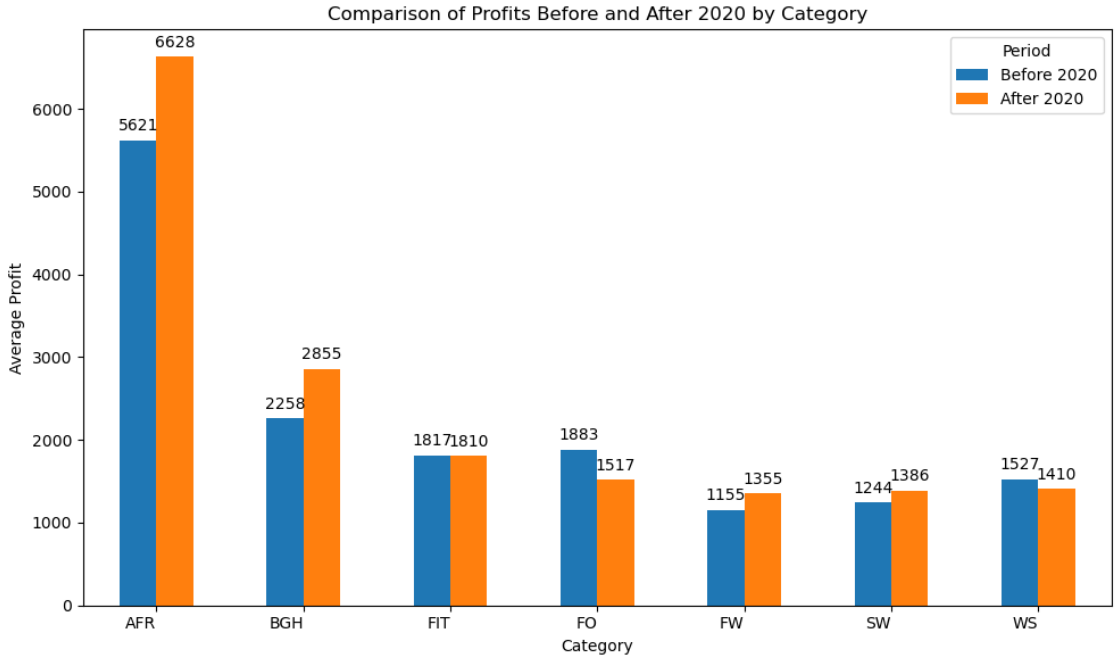


[261]:
```
sns.barplot(x='Category', y='Revenue', data=df_norevprof)
plt.show()
```

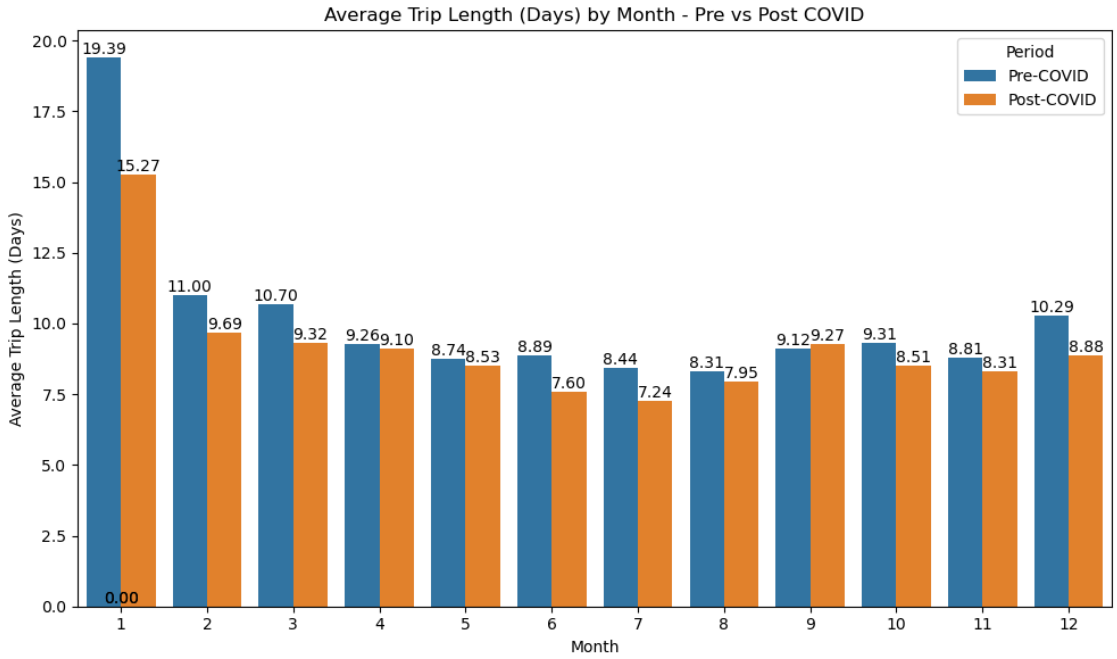


[48]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
df['Period'] = df['Year'].apply(lambda x: 'Before 2020' if x <= 2020 else␣
 ↪'After 2020')
avg_profits = df.groupby(['Category', 'Period'])['Profit'].mean().reset_index()
pivot_table = avg_profits.pivot(index='Category', columns='Period',␣
 ↪values='Profit').fillna(np.nan)
period_order = ['Before 2020', 'After 2020']
pivot_table = pivot_table[period_order]
category_profit_bar = pivot_table.plot(kind='bar', figsize=(10, 6))
for p in category_profit_bar.patches:
    height = p.get_height()
    if not np.isnan(height):
        category_profit_bar.annotate(format(height, '.0f'),
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='center', xytext=(0, 9), textcoords='offset␣
 ↪points', fontsize=10)
plt.title('Comparison of Profits Before and After 2020 by Category')
plt.xlabel('Category')
plt.ylabel('Average Profit')
```

```
plt.xticks(rotation=0, ha='right')
plt.legend(title='Period')
plt.tight_layout()
plt.show()
```

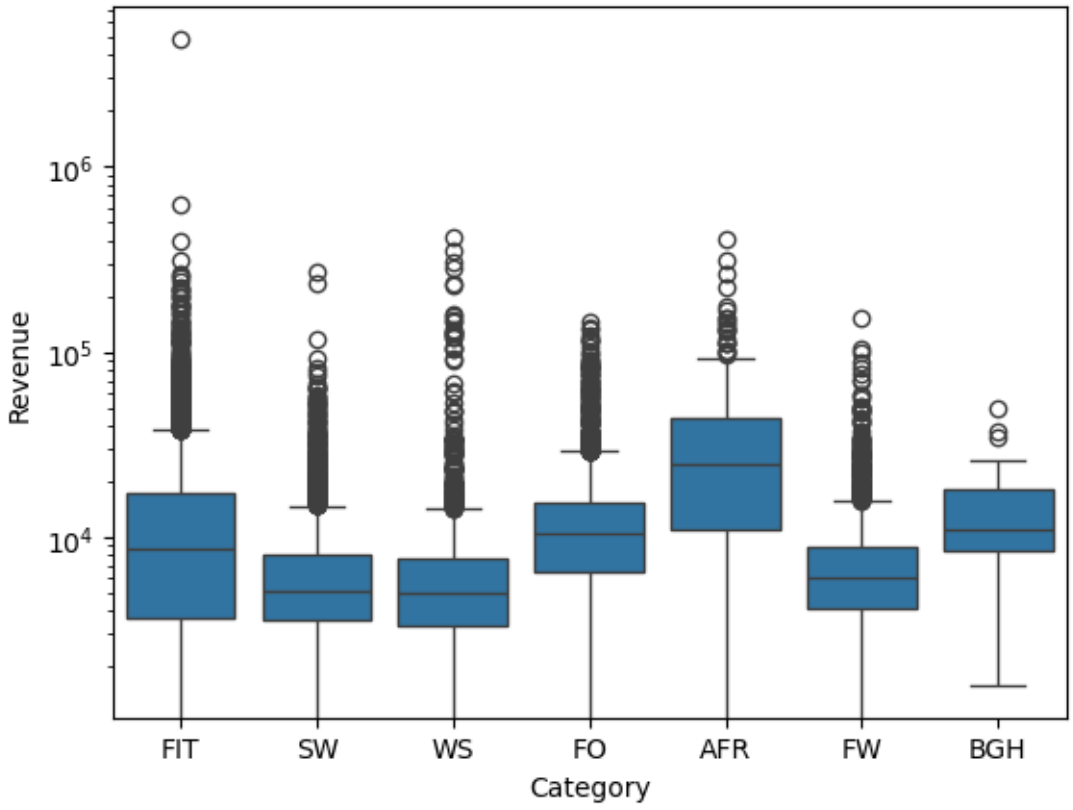


[84]:
```
combined_data = pd.DataFrame({
    'Month': list(pre_covid_avg_trip_length.index) +␣
 ↪list(post_covid_avg_trip_length.index),
    'Average Trip Length': list(pre_covid_avg_trip_length.values) +␣
 ↪list(post_covid_avg_trip_length.values),
    'Period': ['Pre-COVID'] * len(pre_covid_avg_trip_length) + ['Post-COVID'] *␣
 ↪len(post_covid_avg_trip_length)
})
plt.figure(figsize=(10, 6))
sns.barplot(x='Month', y='Average Trip Length', hue='Period',␣
 ↪data=combined_data)
for p in plt.gca().patches:
    plt.text(p.get_x() + p.get_width() / 2, p.get_height(),
             '{:.2f}'.format(p.get_height()), ha='center', va='bottom')
plt.title('Average Trip Length (Days) by Month - Pre vs Post COVID')
plt.xlabel('Month')
plt.ylabel('Average Trip Length (Days)')
plt.tight_layout()

plt.show()
```



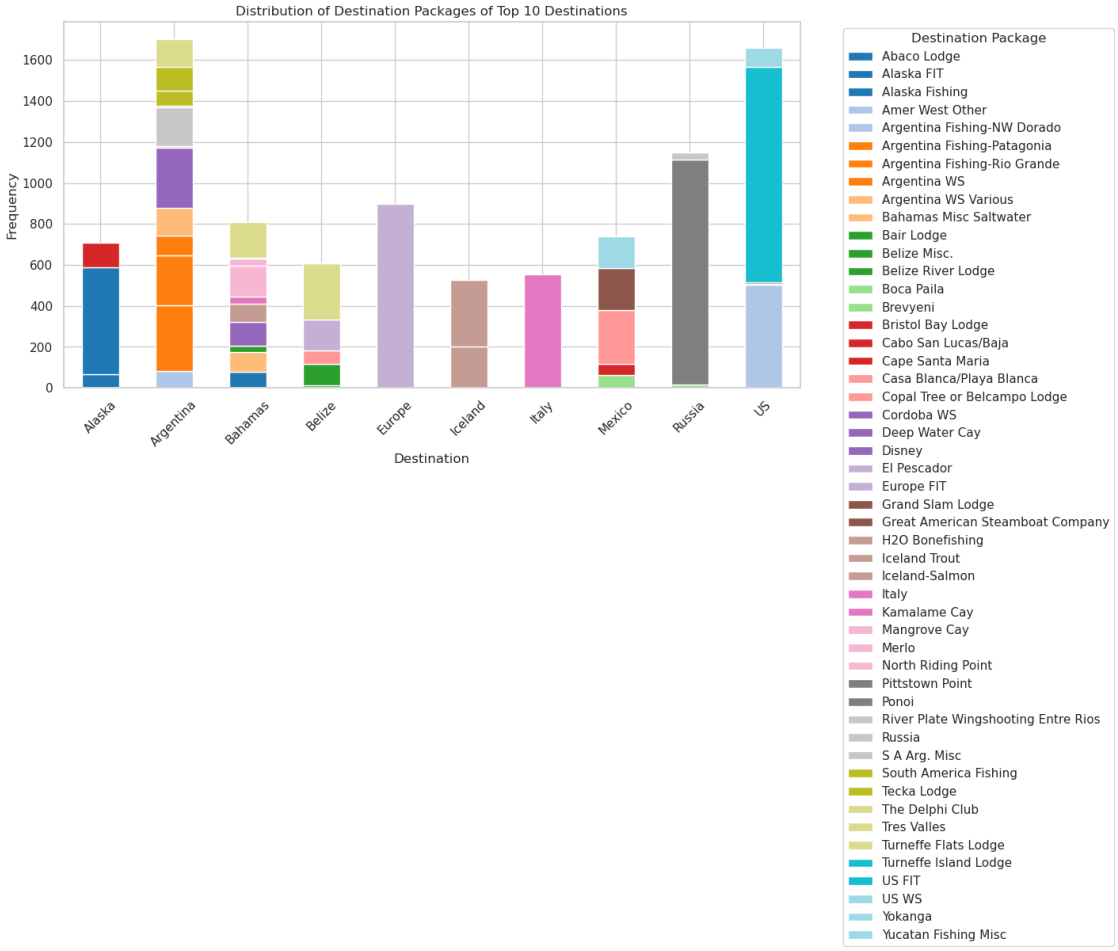
1.1.4 *I didn't think we were going to use revenue as a considering factor since we're focusing on profit. Thoughts?*

[100]:
```
sns.boxplot(x='Category', y='Revenue', data=df)
plt.yscale('log')
plt.show()
```



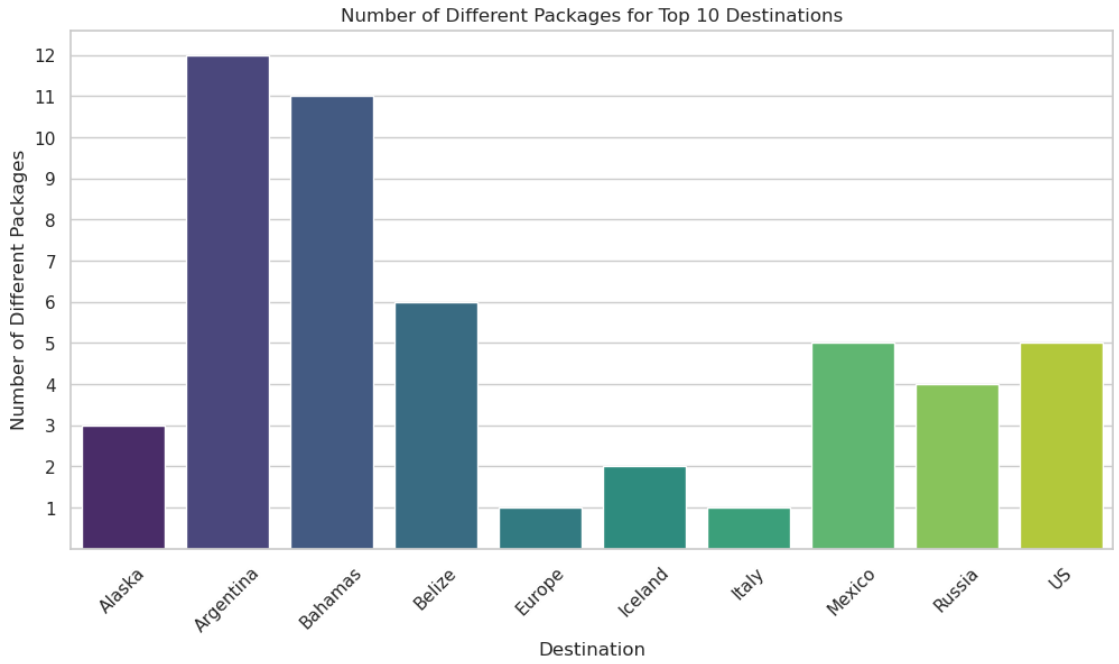
1.2 Packages among Destinations

[286]:
```
package_destination = df.groupby('Destination_Country')['Destination_Package'].
 ↪count().nlargest(10).index
top_10_destination_df = df[df['Destination_Country'].isin(package_destination)]

pivot_table = top_10_destination_df.pivot_table(index='Destination_Country',␣
 ↪columns='Destination_Package', aggfunc='size', fill_value=0)


stacked_plot = pivot_table.plot(kind='bar', stacked=True, figsize=(12, 6),␣
 ↪colormap='tab20')
plt.title('Distribution of Destination Packages of Top 10 Destinations')
plt.xlabel('Destination')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.legend(title='Destination Package', bbox_to_anchor=(1.05, 1), loc='upper␣
 ↪left')
plt.show()
```



[262]:
```
package_dest_counts = top_10_destination_df.
  ↪groupby('Destination_Country')['Destination_Package'].nunique()

package_dest_counts_df = package_destination_counts.reset_index()
package_dest_counts_df.columns = ['Destination_Country', 'Number_of_Packages']


plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
sns.barplot(data=package_counts_df, x='Destination_Country',␣
  ↪y='Number_of_Packages', palette='viridis')
plt.title('Number of Different Packages for Top 10 Destinations')
plt.xlabel('Destination')
plt.ylabel('Number of Different Packages')
plt.yticks(ticks= list(range(1,13)))
plt.xticks(rotation=45)
plt.show()
```

```
/tmp/ipykernel_30188/4060735050.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(data=package_counts_df, x='Destination_Country',
y='Number_of_Packages', palette='viridis')
```

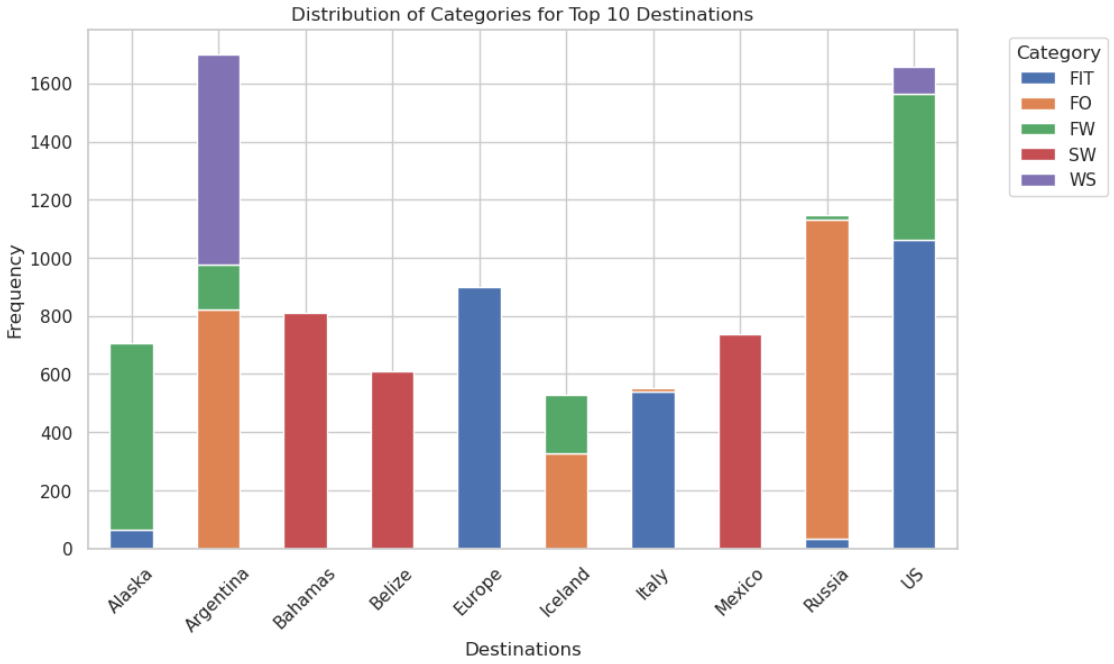


Argentina has the most number of different destination packages (12 packages), followed by Bahamas (11 packages). The number of bookings for these different travel packages for these two countries vary. The package that is mostly booked in Russia is Ponoi. Europe FIT is the mostly booked package in Europe and US fit for United States. Italy FIT is the most booked package for Italy. For these four places, they have asmall number of different packages (less than 5), indicating that alot of their profit for each place must have come from these individual packages that are receiving the most bookings.

1.3 Categories among top 10 destinations

[285]:
```
top_destinations = df['Destination_Country'].value_counts().nlargest(10).index
df_top_10_dest = df[df['Destination_Country'].isin(top_destinations)]

pivot_table = df_top_10_dest.pivot_table(index='Destination_Country',␣
↪columns='Category', aggfunc='size', fill_value=0)
```

```
pivot_table.plot(kind='bar', stacked=True, figsize=(10, 6))
plt.title('Distribution of Categories for Top 10 Destinations')
plt.xlabel('Destinations')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.legend(title='Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



FIT is the most popular trip category for Europe, Italy, and US .

[290]:
```
travel_data = pd.read_csv("travel_data_final.csv")


travel_data['Year'] = pd.to_datetime(travel_data['Date_of_trip_from']).dt.year
travel_data['Month'] = pd.to_datetime(travel_data['Date_of_trip_from']).dt.month


pre_covid_data = travel_data[travel_data['Year'] < 2019]

post_covid_data = travel_data[travel_data['Year'] >= 2019]
```

[291]:
```
print("Pre-COVID Data:")
print(pre_covid_data)

print("\nPost-COVID Data:")
```

```
print(post_covid_data)
```

```
Pre-COVID Data:
          bkgref  Trip ID           destination   Destination_Package  \
0      B16-42027      805         Galapagos-FIT         Galapagos-FIT
1      B11-25115      211        Deep Water Cay        Deep Water Cay
2      B11-27267      211        Deep Water Cay        Deep Water Cay
3      B11-25462      211        Deep Water Cay        Deep Water Cay
4      B14-36987      235    North Riding Point    North Riding Point
…          …      …                   …                   …
16670  B15-39633      380                Ponoi                 Ponoi
16671  B16-42572      380                Ponoi                 Ponoi
16672  B17-45391      380                Ponoi                 Ponoi
16678  B15-38845      209  Yucatan Fishing Misc  Yucatan Fishing Misc
16679  B17-44666      359         Tsimane Lodge         Tsimane Lodge

      Destination Comparision Destination_Country Category Date_of_trip_from  \
0                       Match             Ecuador      FIT          4/7/2017
1                       Match             Bahamas       SW         4/13/2011
2                       Match             Bahamas       SW         4/23/2012
3                       Match             Bahamas       SW         4/28/2011
4                       Match             Bahamas       SW          4/5/2015
…                       …                 …      …              …
16670                   Match              Russia       FO         9/10/2016
16671                   Match              Russia       FO          9/9/2017
16672                   Match              Russia       FO          9/8/2018
16678                   Match              Mexico       SW          9/8/2015
16679                   Match             Bolivia       FW         9/14/2017

      Date_of_trip_to  Period_of_Trip  Year  month  Month  Revenue  Profit  \
0           4/16/2017               9  2017      4      4  20910.0  2322.0
1           4/17/2011               4  2011      4      4   3295.0   659.0
2           4/27/2012               4  2012      4      4   3900.0   696.9
3            5/2/2011               4  2011      4      4   2805.0   503.0
4           4/11/2015               6  2015      4      4   9600.0  1920.0
…               …              …   …    …     …       …      …
16670       9/17/2016               7  2016      9      9  15490.0  2323.5
16671       9/16/2017               7  2017      9      9  15490.0  2323.5
16672       9/15/2018               7  2018      9      9  15490.0  2323.5
16678       9/13/2015               5  2015      9      9   2995.0   599.0
16679       9/24/2017              10  2017      9      9   7600.0  1140.0

       groupsize
0              3
1              1
2              1
3              1
4              2
```

```
…          …
16670          1
16671          1
16672          1
16678          1
16679          1

[11270 rows x 16 columns]

Post-COVID Data:
          bkgref  Trip ID           destination   Destination_Package  \
5      B19-49576      230          Mangrove Cay          Mangrove Cay
8      B19-49577      230          Mangrove Cay          Mangrove Cay
12     B18-47163      224  Turneffe Flats Lodge  Turneffe Flats Lodge
19     B18-48185      220    Belize River Lodge    Belize River Lodge
20     B21-54739      215       The Delphi Club       The Delphi Club
…          …      …                   …                     …
16674  B18-48526      380               Ponoi                 Ponoi
16675  B19-50032      380               Ponoi                 Ponoi
16676  B20-53811      380               Ponoi                 Ponoi
16677  B20-53891      380               Ponoi                 Ponoi
16680  B20-53898      380               Ponoi                 Ponoi

      Destination Comparision Destination_Country Category Date_of_trip_from  \
5                       Match             Bahamas       SW          4/3/2019
8                       Match             Bahamas       SW          4/4/2019
12                      Match              Belize       SW         4/27/2019
19                      Match              Belize       SW         4/27/2019
20                      Match             Bahamas       SW         4/17/2021
…                       …                 …      …              …
16674                   Match              Russia       FO          9/7/2019
16675                   Match              Russia       FO         9/21/2019
16676                   Match              Russia       FO         9/18/2021
16677                   Match              Russia       FO         9/11/2021
16680                   Match              Russia       FO         9/18/2021

      Date_of_trip_to  Period_of_Trip  Year  month  Month  Revenue   Profit  \
5           4/7/2019               4  2019      4      4   3808.0   711.20
8           4/7/2019               3  2019      4      4   3808.0   711.20
12          5/4/2019               7  2019      4      4   4736.2   878.00
19          5/2/2019               5  2019      4      4   4071.0   366.00
20         4/22/2021               5  2021      4      4   7590.0  1518.00
…                …             …   …    …    …      …       …       …
16674      9/14/2019               7  2019      9      9  16590.0  2488.50
16675      9/28/2019               7  2019      9      9  14490.0  1948.50
16676      9/25/2021               7  2021      9      9  13990.0  2098.50
16677      9/18/2021               7  2021      9      9  16590.0  2488.50
16680      9/25/2021               7  2021      9      9  10990.0  1648.58
```

```
       groupsize
5              1
8              1
12             1
19             1
20             1
…            …
16674          1
16675          1
16676          1
16677          1
16680          1

[5411 rows x 16 columns]
```

2 Finding Recommendations Based on Client's Usual Group Size for Top 10 Countries

As per our finding the mode of groupsize is 2 which indicates most of clients going for travel are couples,hence recommending the couples the top 10 such countries which tend to give the client the best profit.

[304]:
```
client_group_size = 2

# Filter data for group size 2 and top 10 countries
recommended_packages_top =␣
 ↪group_recommendation_top[group_recommendation_top['groupsize'] ==␣
 ↪client_group_size]

print(f"Recommended packages for group size {client_group_size} in top 10␣
 ↪destination countries:")
print(recommended_packages_top)
```

```
Recommended packages for group size 2 in top 10 destination countries:
    groupsize Destination_Country Category       Profit       Revenue
19          2              Alaska      FIT  2074.855667  20181.812000
20          2              Alaska       FW  2004.568973  13109.859384
21          2           Argentina       FO  1978.433210  10896.621399
22          2           Argentina       FW  1300.031333   8256.800000
23          2           Argentina       WS  1708.866265   7363.846988
24          2             Bahamas       SW  1363.854691   8628.769342
25          2        East Africa      AFR  5018.289730  31015.554054
26          2              Europe      FIT  1073.726433   8263.906433
27          2              France      FIT  1842.660873  11762.673746
28          2              France       WS  1480.777778  11250.333333
29          2              Russia      FIT  1593.636364  10726.467273
```

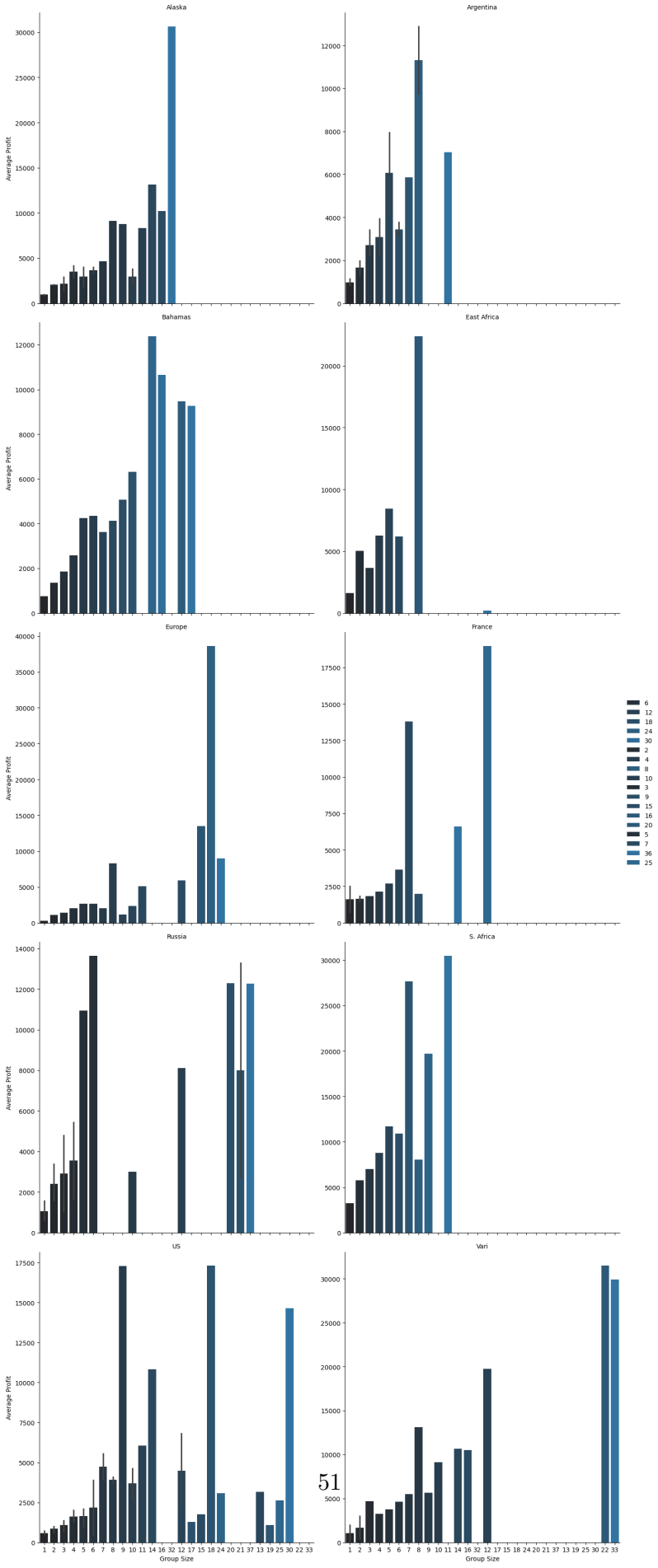
```
30          2              Russia       FO  3380.222549  22527.911765
31          2              Russia       FW  2247.000000  14980.000000
32          2           S. Africa      AFR  5758.300140  29766.537343
33          2                  US      FIT   629.715988   6206.407461
34          2                  US       FW   994.279947   6042.024043
35          2                  US       WS   943.863077   6749.246154
36          2                Vari      FIT  2993.070312  26230.599688
37          2                Vari       FW   390.025000   3732.500000
```

[305]:
```
# Sort recommended packages by profit for top 10 countries
final_recommendations_top = recommended_packages_top.sort_values(by='Profit',␣
 ↪ascending=False)

print("Top recommendations for the client based on group size in top 10␣
 ↪destination countries:")
print(final_recommendations_top)
```

```
Top recommendations for the client based on group size in top 10 destination
countries:
    groupsize Destination_Country Category       Profit       Revenue
32          2           S. Africa      AFR  5758.300140  29766.537343
25          2         East Africa      AFR  5018.289730  31015.554054
30          2              Russia       FO  3380.222549  22527.911765
36          2                Vari      FIT  2993.070312  26230.599688
31          2              Russia       FW  2247.000000  14980.000000
19          2              Alaska      FIT  2074.855667  20181.812000
20          2              Alaska       FW  2004.568973  13109.859384
21          2           Argentina       FO  1978.433210  10896.621399
27          2              France      FIT  1842.660873  11762.673746
23          2           Argentina       WS  1708.866265   7363.846988
29          2              Russia      FIT  1593.636364  10726.467273
28          2              France       WS  1480.777778  11250.333333
24          2             Bahamas       SW  1363.854691   8628.769342
22          2           Argentina       FW  1300.031333   8256.800000
26          2              Europe      FIT  1073.726433   8263.906433
34          2                  US       FW   994.279947   6042.024043
35          2                  US       WS   943.863077   6749.246154
33          2                  US      FIT   629.715988   6206.407461
37          2                Vari       FW   390.025000   3732.500000
```

[306]:
```
import seaborn as sns
import matplotlib.pyplot as plt

g = sns.FacetGrid(group_recommendation_top, col='Destination_Country',␣
 ↪col_wrap=2, height=7, sharey=False)
plt.figure(figsize=(15,10 ))
g.map_dataframe(sns.barplot, x='groupsize', y='Profit',␣
 ↪hue='groupsize',palette='dark:#1f77b4')
```

```
g.add_legend()
g.set_axis_labels('Group Size', 'Average Profit')
g.set_titles(col_template="{col_name}")
plt.show()
```


Alaska
Argentina
Bahamas
East Africa
Europe
France
Russia
S. Africa
US
Vari
Average Profit
Group Size
6
12
18
24
30
2
4
8
10
3
9
15
16
20
5
7
36
25

```
<Figure size 1500x1000 with 0 Axes>
```

2.1 Key Insights:

Destination-Specific Patterns:

Alaska: The average profit seems to increase with group size, with a significant spike around group size 10.

Argentina: The average profit is relatively stable across group sizes, with a slight increase for larger groups.

Bahamas: The average profit shows a similar pattern to Argentina, with a slight increase for larger groups.

East Africa: The average profit is generally lower than other destinations, with a few outliers for larger group sizes.

Europe: There's a clear trend of increasing average profit with larger group sizes.

France: Similar to Europe, the average profit increases with group size, with a few outliers.

Russia: The average profit is relatively high, with a slight decrease for larger group sizes.

South Africa: The average profit is relatively stable across group sizes, with a few outliers.

US: The average profit is generally lower than other destinations, with a few outliers for larger group sizes.

Spain: The average profit is relatively high, with a slight decrease for larger group sizes

General Trends:

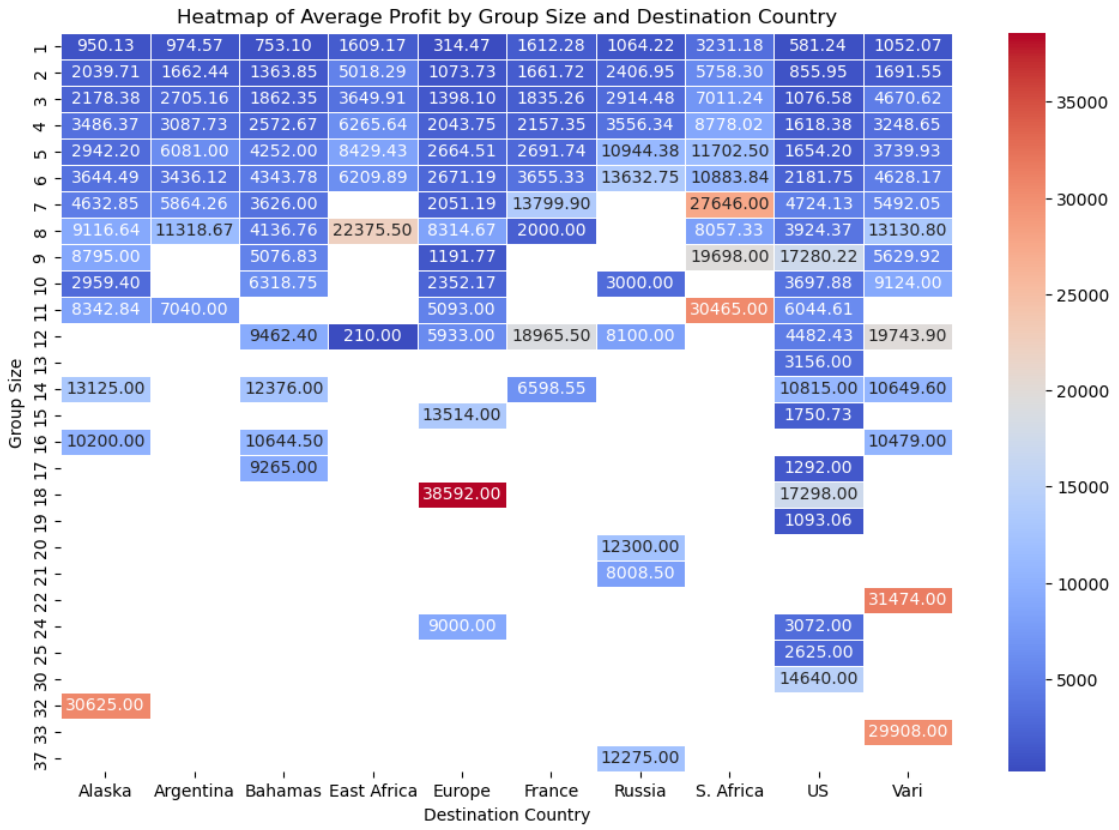
Group Size Impact: In general, larger group sizes tend to have higher average profits, especially for destinations like Europe and Spain. Destination Variation: The average profit varies significantly across destinations, indicating that some destinations are more profitable than others.

Additional Observations:

Outliers: There are a few outliers in the data, especially for larger group sizes. These could be due to factors like special packages, seasonal variations, or other external influences.

[308]:
```
import seaborn as sns
import matplotlib.pyplot as plt

pivot_table = group_recommendation_top.pivot_table(index='groupsize',␣
 ↪columns='Destination_Country', values='Profit', aggfunc='mean')
plt.figure(figsize=(12, 8))
sns.heatmap(pivot_table, annot=True, cmap='coolwarm', fmt='.2f', linewidths=.5)
plt.title('Heatmap of Average Profit by Group Size and Destination Country')
plt.xlabel('Destination Country')
plt.ylabel('Group Size')
```

```
plt.show()
```

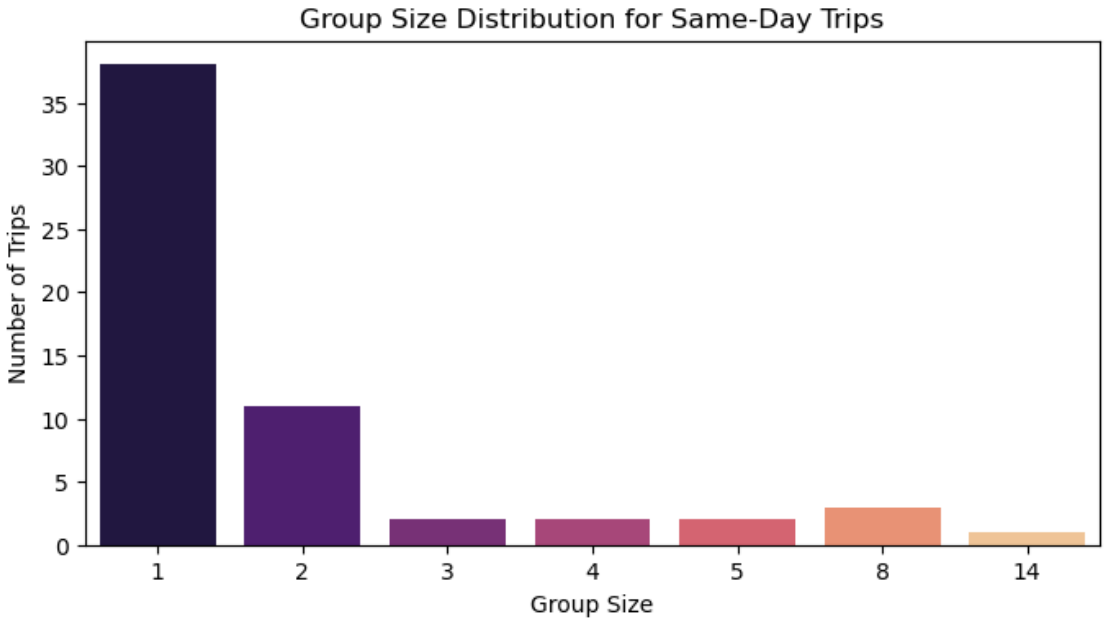


[296]:
```
zero_trip_period = df[df['Trip_days'] == 0]

# Count the number of zero-period trips and calculate the percentage
zero_trip_count = len(zero_trip_period)
total_count = len(df)
percentage_zero_trip = (zero_trip_count / total_count) * 100
destinations_zero_trip = zero_trip_period['Destination_Country'].value_counts()
groupsize_zero_trip = zero_trip_period['Groupsize'].value_counts()
plt.figure(figsize=(8,4))
sns.barplot(x=groupsize_zero_trip.index, y=groupsize_zero_trip.values,␣
 ↪palette='magma')
plt.title('Group Size Distribution for Same-Day Trips')
plt.ylabel('Number of Trips')
plt.xlabel('Group Size')
plt.show()
```

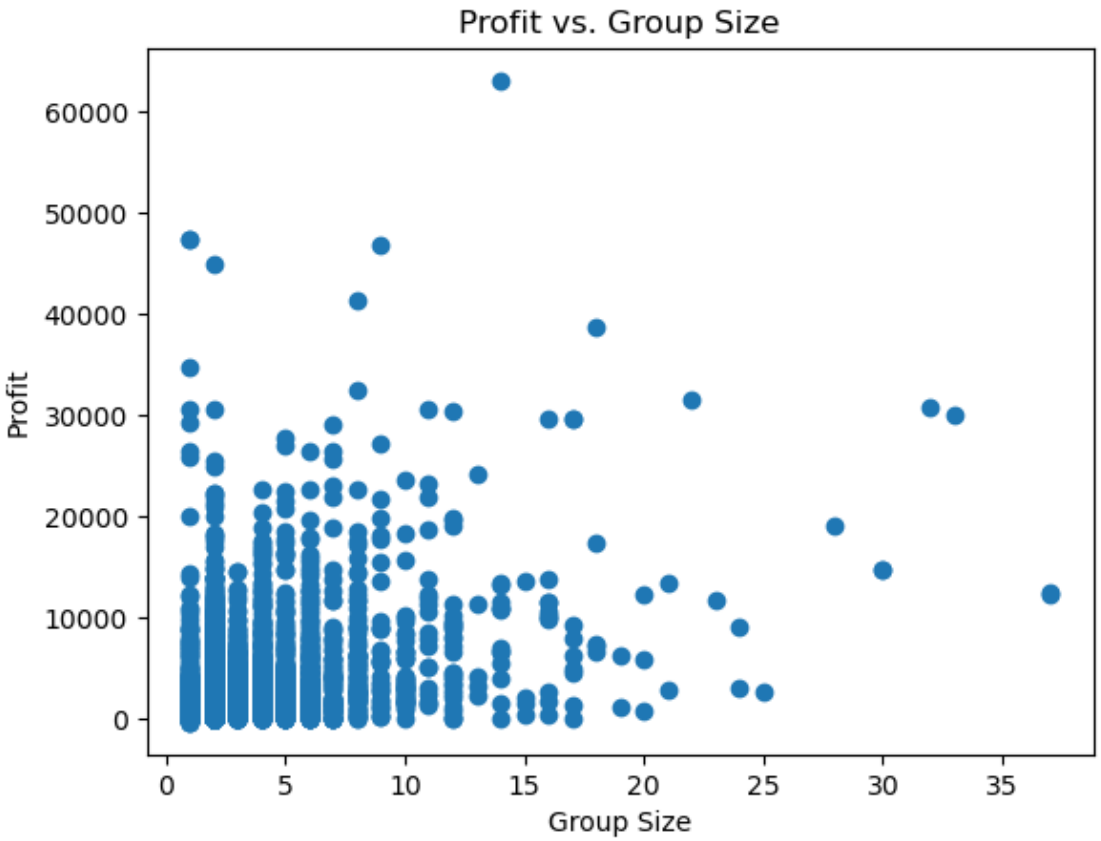
```
/tmp/ipykernel_442/745638395.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
```

v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
  sns.barplot(x=groupsize_zero_trip.index, y=groupsize_zero_trip.values,
palette='magma')
```



[179]:
```
# Create the scatter plot
plt.scatter(df['groupsize'], df['Profit'])

# Add labels and title
plt.xlabel('Group Size')
plt.ylabel('Profit')
plt.title('Profit vs. Group Size')

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



[180]:
```
# Calculate the correlation coefficient
correlation_coefficient = df['Profit'].corr(df['groupsize'])

print("Correlation coefficient:", correlation_coefficient)
```

```
Correlation coefficient: 0.42769527484290326
```

[215]:
```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.11/site-
packages (1.3.1)
Requirement already satisfied: numpy<2.0,>=1.17.3 in
/opt/conda/lib/python3.11/site-packages (from scikit-learn) (1.24.4)
Requirement already satisfied: scipy>=1.5.0 in /opt/conda/lib/python3.11/site-
packages (from scikit-learn) (1.11.3)
Requirement already satisfied: joblib>=1.1.1 in /opt/conda/lib/python3.11/site-
packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.11/site-packages (from scikit-learn) (3.2.0)
Note: you may need to restart the kernel to use updated packages.
```

[216]:
```
from sklearn.linear_model import LinearRegression
```

[224]:
```
# X = df[['Trip_days', 'Revenue', 'Groupsize']]  # Independent variables
# y = df['Profit']  # Dependent variable (Profit)
X = df[(df['Profit'] != 0) & (df['Revenue'] != 0)]
X = X[['Trip_days', 'Revenue', 'Groupsize']]  # Independent variables
y = df[(df['Profit'] != 0) & (df['Revenue'] != 0)]
y = y['Profit']  # Dependent variable (Profit)

# Create a linear regression model
model = LinearRegression()

# Fit the model
model.fit(X, y)

# Get the coefficients for each variable
coefficients = pd.DataFrame({'Feature': X.columns, 'Coefficient': model.coef_})

# Display the coefficients
print(coefficients)
```

```
     Feature  Coefficient
0  Trip_days     0.860627
1    Revenue     0.116442
2  Groupsize    61.593882
```

[ ]:
```

```