In [45]:
```
# Import the libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# read the csv files

customers_df = pd.read_csv("Customers.csv")
subscriptions_df = pd.read_csv("Subscriptions.csv")
transactions_df = pd.read_csv("Transactions.csv")
churn_df = pd.read_csv("Churn.csv")
```

In [4]:
```
customers_df.head(10)
```

Out[4]:

	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
0	1	John	Doe	john.doe@example.com	123-456-7890	10-01-2022	Active	North America
1	2	Jane	Smith	jane.smith@example.com	098-765-4321	15-12-2021	Inactive	Europe
2	3	Alice	Johnson	alice.j@example.com	567-890-1234	20-03-2020	Active	Asia
3	4	Bob	Brown	bob.brown@example.com	234-567-8901	25-06-2019	Inactive	North America
4	5	Charlie	Davis	charlie.d@example.com	345-678-9012	14-07-2021	Active	Europe
5	6	Diana	Clark	diana.c@example.com	456-789-0123	30-08-2018	Active	Asia
6	7	Eva	Harris	eva.h@example.com	567-890-1234	22-12-2020	Inactive	North America
7	8	Frank	Garcia	frank.g@example.com	678-901-2345	11-03-2019	Active	Europe
8	9	Grace	Miller	grace.m@example.com	789-012-3456	18-02-2022	Inactive	Asia
9	10	Hank	Wilson	hank.w@example.com	890-123-4567	27-05-2020	Active	North America

In [6]:
```
customers_df.shape
```

Out[6]:
```
(300, 8)
```

In [8]:
```
customers_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   CustomerID   300 non-null    int64 
 1   FirstName    299 non-null    object
 2   LastName     299 non-null    object
 3   Email        298 non-null    object
 4   PhoneNumber  298 non-null    object
 5   JoinDate     298 non-null    object
 6   Status       299 non-null    object
 7   Region       299 non-null    object
dtypes: int64(1), object(7)
memory usage: 18.9+ KB
```

In [9]:
```
customers_df.describe()
```

Out[9]:

	CustomerID
count	300.000000
mean	150.500000
std	86.746758
min	1.000000
25%	75.750000
50%	150.500000
75%	225.250000
max	300.000000

In [10]:
```
transactions_df.describe()
```

Out[10]:

	TransactionID	CustomerID	Amount
count	216.000	216.000000	216.000000
mean	108.500	153.388889	112.199074
std	62.498	71.834931	53.175289
min	1.000	11.000000	20.000000
25%	54.750	103.750000	64.500000
50%	108.500	156.500000	117.500000
75%	162.250	214.250000	157.250000
max	216.000	270.000000	200.000000

In [12]:
```
# 1. Print Customer First Name and Last name

customer_names = customers_df[["FirstName", "LastName"]]

customer_names.head(10)
```

Out[12]:

	FirstName	LastName
0	John	Doe
1	Jane	Smith
2	Alice	Johnson
3	Bob	Brown
4	Charlie	Davis
5	Diana	Clark
6	Eva	Harris
7	Frank	Garcia
8	Grace	Miller
9	Hank	Wilson

In [13]:
```
#2. Find Customers from a North America Region

customers_NorthAmerica = customers_df[customers_df["Region"]=="North America"]

customers_NorthAmerica.head(20)
```

Out[13]:

	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
0	1	John	Doe	john.doe@example.com	123-456-7890	10-01-2022	Active	North America
3	4	Bob	Brown	bob.brown@example.com	234-567-8901	25-06-2019	Inactive	North America
6	7	Eva	Harris	eva.h@example.com	567-890-1234	22-12-2020	Inactive	North America
9	10	Hank	Wilson	hank.w@example.com	890-123-4567	27-05-2020	Active	North America
12	13	Kelly	Lewis	kelly.l@example.com	123-456-7890	15-04-2018	Active	North America
15	16	Noah	Allen	noah.a@example.com	456-789-0123	19-06-2021	Inactive	North America
18	19	Quinn	Scott	quinn.s@example.com	789-012-3456	05-01-2018	Active	North America
21	22	Tina	Perez	tina.p@example.com	012-345-6789	09-08-2020	Inactive	North America
24	25	Wendy	Cook	wendy.c@example.com	345-678-9012	21-07-2019	Active	North America
27	28	Zane	Evans	zane.e@example.com	678-901-2345	13-12-2019	Inactive	North America
30	31	Leo	Morgan	leo.m@example.com	321-654-0987	10-01-2023	Active	North America
33	34	Penny	Ford	penny.f@example.com	654-321-0987	30-10-2021	Inactive	North America
36	37	Tom	Wood	tom.w@example.com	987-321-6540	03-03-2023	Active	North America
39	40	Will	Stone	will.s@example.com	321-654-9870	20-07-2019	Active	North America
40	41	Anna	Moore	anna.m@example.com	123-456-7890	01-02-2023	Active	North America
43	44	David	James	david.j@example.com	456-789-0123	05-05-2020	Inactive	North America
46	47	Grace	Morris	grace.m@example.com	789-012-3456	15-11-2021	Active	North America
49	50	Jasmine	Taylor	jasmine.t@example.com	012-345-6789	10-05-2021	Inactive	North America
50	51	Kyle	Lee	kyle.l@example.com	123-456-7891	05-04-2023	Active	North America
53	54	Nora	Hughes	nora.h@example.com	456-789-0124	30-06-2020	Inactive	North America

In [15]:
```
#3. Count the number of customers

total_customers = customers_df.shape[0]

print(f"Total nnumber of customers : {total_customers}")
```

```
Total nnumber of customers : 300
```

In [18]:
```
#4. Show me all the details of the "Active" customers. Also give the count.

active_customers = customers_df[customers_df["Status"] == "Active"]
print(active_customers)
count = active_customers.shape[0]


active_customers.head(152)
```

```
     CustomerID FirstName LastName                    Email   PhoneNumber  \
0             1      John      Doe      john.doe@example.com  123-456-7890
2             3     Alice  Johnson       alice.j@example.com  567-890-1234
4             5   Charlie    Davis     charlie.d@example.com  345-678-9012
5             6     Diana    Clark       diana.c@example.com  456-789-0123
7             8     Frank   Garcia       frank.g@example.com  678-901-2345
..          ...       ...      ...                       ...           ...
290         291     Lucas   Taylor  lucas.taylor@example.com  901-234-5678
292         293      Jake    Young    jake.young@example.com  123-456-7890
294         295     Aiden    Davis   aiden.davis@example.com  345-678-9012
296         297      Liam   Wilson   liam.wilson@example.com  567-890-1234
298         299      Ella  Johnson  ella.johnson@example.com  789-012-3456

       JoinDate  Status         Region
0    10-01-2022  Active  North America
2    20-03-2020  Active           Asia
4    14-07-2021  Active         Europe
5    30-08-2018  Active           Asia
7    11-03-2019  Active         Europe
..          ...     ...            ...
290  19-08-2019  Active  North America
292  29-05-2021  Active           Asia
294  03-11-2022  Active         Europe
296  10-08-2020  Active  North America
298  18-05-2022  Active           Asia

[152 rows x 8 columns]
```

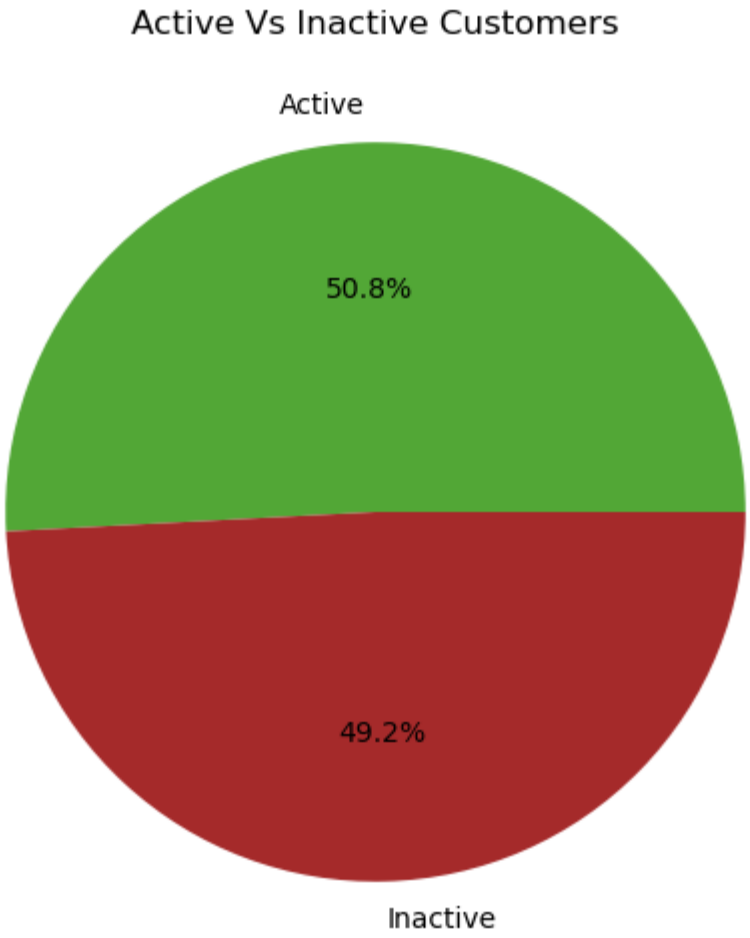
Out[18]:

	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
0	1	John	Doe	john.doe@example.com	123-456-7890	10-01-2022	Active	North America
2	3	Alice	Johnson	alice.j@example.com	567-890-1234	20-03-2020	Active	Asia
4	5	Charlie	Davis	charlie.d@example.com	345-678-9012	14-07-2021	Active	Europe
5	6	Diana	Clark	diana.c@example.com	456-789-0123	30-08-2018	Active	Asia
7	8	Frank	Garcia	frank.g@example.com	678-901-2345	11-03-2019	Active	Europe
...	...	...	...	...	...	...	...	...
290	291	Lucas	Taylor	lucas.taylor@example.com	901-234-5678	19-08-2019	Active	North America
292	293	Jake	Young	jake.young@example.com	123-456-7890	29-05-2021	Active	Asia
294	295	Aiden	Davis	aiden.davis@example.com	345-678-9012	03-11-2022	Active	Europe
296	297	Liam	Wilson	liam.wilson@example.com	567-890-1234	10-08-2020	Active	North America
298	299	Ella	Johnson	ella.johnson@example.com	789-012-3456	18-05-2022	Active	Asia

152 rows × 8 columns

In [26]:

```
#5. Visualise to show the percentage of Active & Inactive customers

# Calculate the number of active & inactive customers

status_count = customers_df['Status'].value_counts()

# create a pie chart

plt.figure(figsize=(10,6))

plt.pie(status_count,labels=status_count.index, autopct = '%1.1f%%',colors=['#52A736','#A52A2A'])

plt.title("Active Vs Inactive Customers")
plt.show()
```



In [33]:
```
# list down all those customers who joined after 01-01-2021.


after_2021_customers = customers_df[customers_df["JoinDate"]>"01-01-2021"]
after_2021_customers.head(30)
```

Out[33]:

	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
0	1	John	Doe	john.doe@example.com	123-456-7890	2022-01-10	Active	North America
1	2	Jane	Smith	jane.smith@example.com	098-765-4321	2021-12-15	Inactive	Europe
4	5	Charlie	Davis	charlie.d@example.com	345-678-9012	2021-07-14	Active	Europe
8	9	Grace	Miller	grace.m@example.com	789-012-3456	2022-02-18	Inactive	Asia
11	12	Jack	White	jack.w@example.com	012-345-6789	2021-11-20	Inactive	Asia
14	15	Mia	Hall	mia.h@example.com	345-678-9012	2022-03-02	Active	Asia
15	16	Noah	Allen	noah.a@example.com	456-789-0123	2021-06-19	Inactive	North America
19	20	Rachel	Adams	rachel.a@example.com	890-123-4567	2021-10-27	Inactive	Europe
22	23	Uma	Collins	uma.c@example.com	123-456-7890	2021-01-03	Active	Europe
25	26	Xander	Cooper	xander.c@example.com	456-789-0123	2021-03-15	Inactive	Europe
28	29	Abby	Flores	abby.f@example.com	789-012-3456	2021-02-22	Active	Europe
30	31	Leo	Morgan	leo.m@example.com	321-654-0987	2023-01-10	Active	North America
31	32	Nina	Gray	nina.g@example.com	456-123-7890	2022-11-22	Inactive	Europe
33	34	Penny	Ford	penny.f@example.com	654-321-0987	2021-10-30	Inactive	North America
34	35	Ray	Hill	ray.h@example.com	123-789-4560	2021-09-18	Active	Europe
35	36	Sara	Green	sara.g@example.com	789-654-1230	2022-02-14	Inactive	Asia
36	37	Tom	Wood	tom.w@example.com	987-321-6540	2023-03-03	Active	North America
38	39	Vera	Scott	vera.s@example.com	654-098-3210	2021-08-10	Active	Asia
40	41	Anna	Moore	anna.m@example.com	123-456-7890	2023-02-01	Active	North America
41	42	Ben	Reed	ben.r@example.com	234-567-8901	2021-12-01	Inactive	Europe
42	43	Clara	Bennett	clara.b@example.com	345-678-9012	2022-03-10	Active	Asia
45	46	Frank	Adams	frank.a@example.com	678-901-2345	2022-07-30	Inactive	Asia
46	47	Grace	Morris	grace.m@example.com	789-012-3456	2021-11-15	Active	North America
48	49	Ian	Stewart	ian.s@example.com	901-234-5678	2022-09-25	Active	Asia
49	50	Jasmine	Taylor	jasmine.t@example.com	012-345-6789	2021-05-10	Inactive	North America

	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
50	51	Kyle	Lee	kyle.l@example.com	123-456-7891	2023-04-05	Active	North America
51	52	Laura	Baker	laura.b@example.com	234-567-8902	2021-11-12	Inactive	Europe
52	53	Mark	Watson	mark.w@example.com	345-678-9013	2022-08-23	Active	Asia
54	55	Oscar	Kim	oscar.k@example.com	567-890-1235	2021-02-18	Active	Europe
56	57	Quincy	Adams	quincy.a@example.com	789-012-3457	2022-12-01	Active	North America

In [29]:
```
after_2021_customers.shape[0]
```

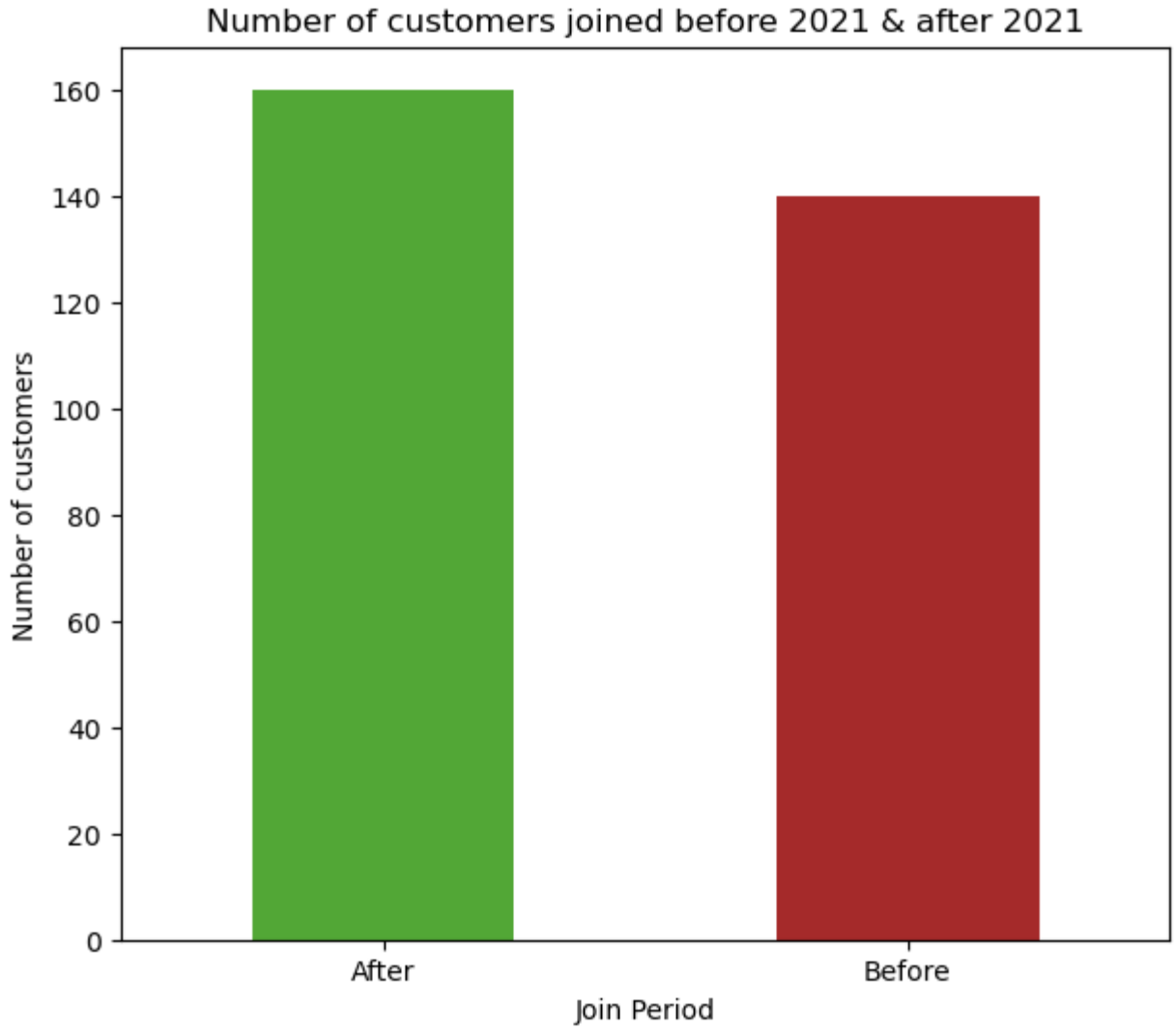
Out[29]: 298

In [47]:
```
# Convert the "Join Date" column to date time format

customers_df["JoinDate"]= pd.to_datetime(customers_df["JoinDate"],format="%d-%m-%Y")

customers_df.head(10)
```

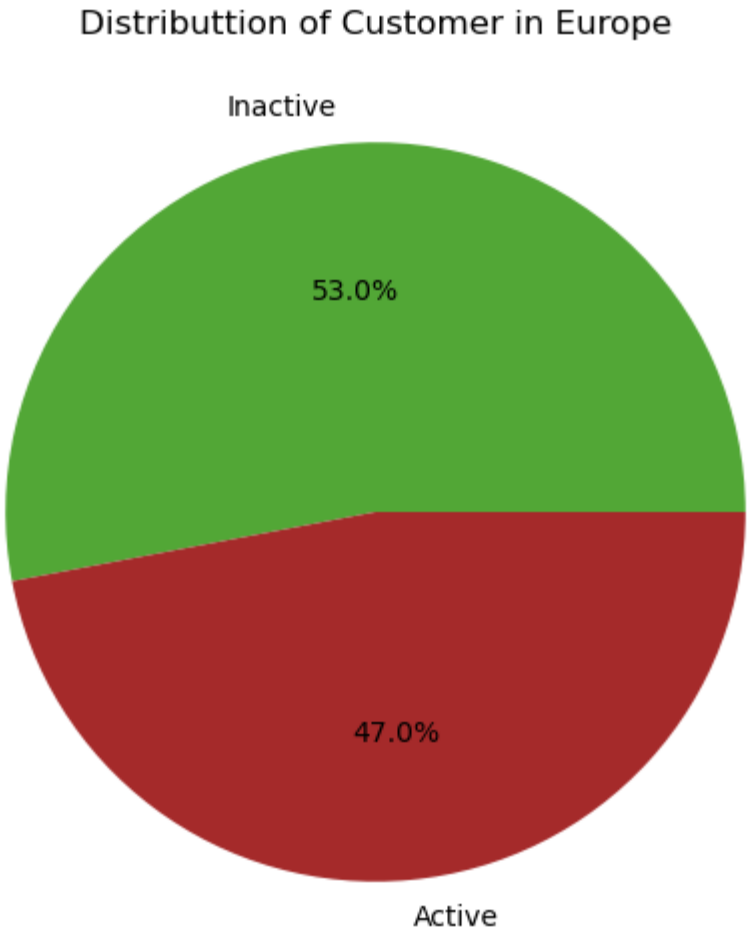
Out[47]:

	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region
0	1	John	Doe	john.doe@example.com	123-456-7890	2022-01-10	Active	North America
1	2	Jane	Smith	jane.smith@example.com	098-765-4321	2021-12-15	Inactive	Europe
2	3	Alice	Johnson	alice.j@example.com	567-890-1234	2020-03-20	Active	Asia
3	4	Bob	Brown	bob.brown@example.com	234-567-8901	2019-06-25	Inactive	North America
4	5	Charlie	Davis	charlie.d@example.com	345-678-9012	2021-07-14	Active	Europe
5	6	Diana	Clark	diana.c@example.com	456-789-0123	2018-08-30	Active	Asia
6	7	Eva	Harris	eva.h@example.com	567-890-1234	2020-12-22	Inactive	North America
7	8	Frank	Garcia	frank.g@example.com	678-901-2345	2019-03-11	Active	Europe
8	9	Grace	Miller	grace.m@example.com	789-012-3456	2022-02-18	Inactive	Asia
9	10	Hank	Wilson	hank.w@example.com	890-123-4567	2020-05-27	Active	North America

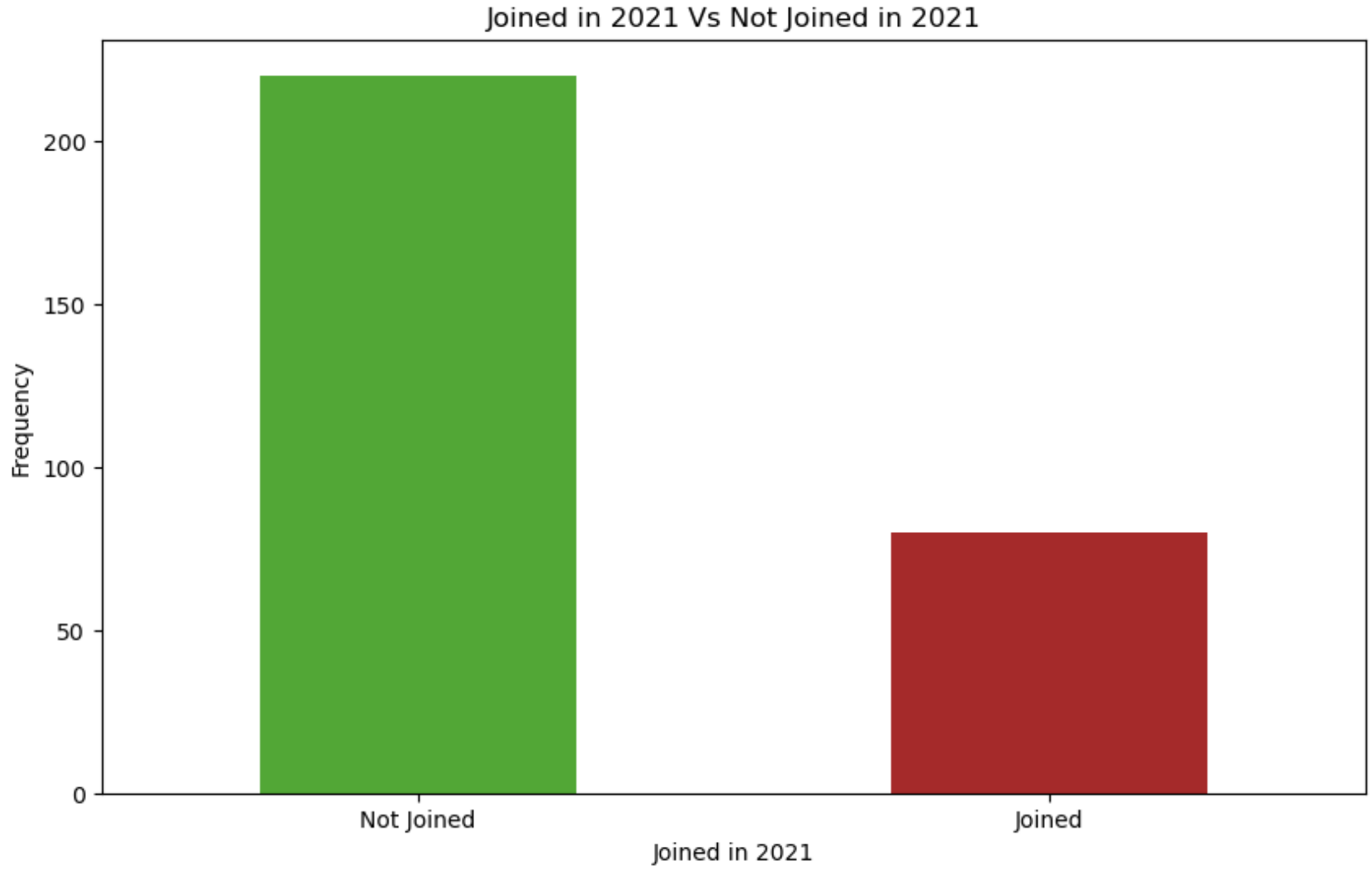
In [55]:
```
# Create a visualisation to show how many customers joined after and before 01-01-2021


# define the date for comparision

date_threshold = pd.Timestamp("2021-01-01")

# create a new column

customers_df["JoinPeriod"] = customers_df['JoinDate'].apply(lambda x:"After" if x>date_threshold else "Bef

# Count the number of customers in each category

join_period_counts = customers_df["JoinPeriod"].value_counts()

# Plot the bar chart


plt.figure(figsize=(7,6))

join_period_counts.plot(kind="bar",color=['#52A736','#A52A2A'])
plt.title("Number of customers joined before 2021 & after 2021")
plt.xlabel("Join Period")
plt.ylabel("Number of customers")
plt.xticks(rotation=0)
plt.show()
```


Number of customers joined before 2021 & after 2021
Number of customers
0
20
40
60
80
100
120
140
160
After
Before
Join Period

In [58]:
```
# Visualise the percentage of Active & Inactive customers in Europe

customers_europe = customers_df[customers_df['Region']=="Europe"]

status_count_europe = customers_europe["Status"].value_counts()

plt.figure(figsize=(10,6))

plt.pie(status_count_europe ,labels=status_count_europe.index, autopct = '%1.1f%%',colors=['#52A736','#A52

plt.title("Distributtion of Customer in Europe")
plt.show()
```


Distributtion of Customer in Europe
Inactive
53.0%
47.0%
Active

In [61]:
```
# Visualise the count of customers who joined in 2021 Vs not joined in 2021

customers_df["Joined2021"]= pd.to_datetime(customers_df['JoinDate']).dt.year==2021

joined_2021_counts = customers_df["Joined2021"].value_counts()

# Create a bar chart

plt.figure(figsize=(10,6))
joined_2021_counts.plot(kind="bar",color=['#52A736','#A52A2A'])
plt.title("Joined in 2021 Vs Not Joined in 2021")

plt.xlabel("Joined in 2021")
plt.ylabel("Frequency")
plt.xticks(ticks=[0,1],labels=["Not Joined","Joined"],rotation=0)
plt.show()
```



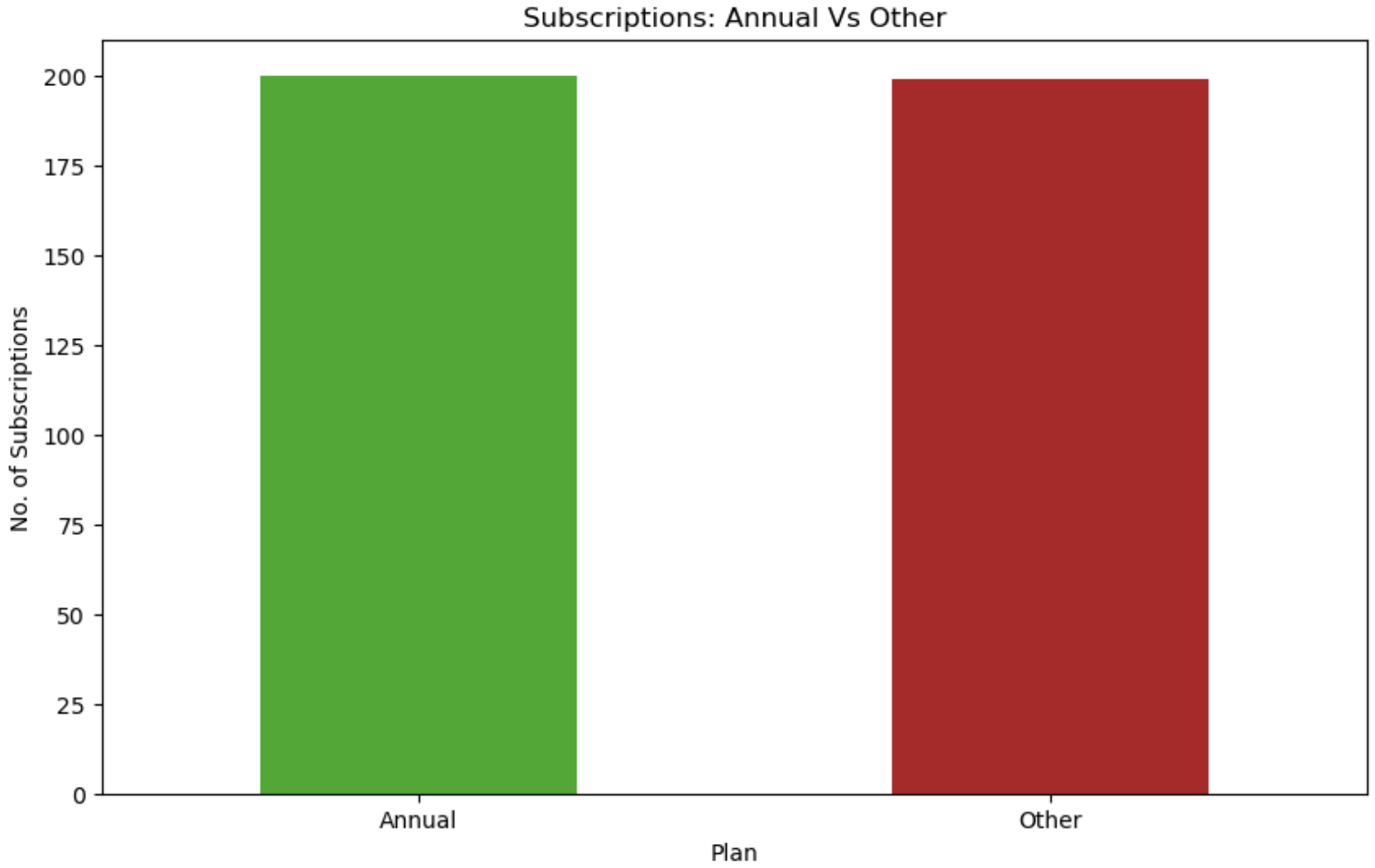
In [65]:
```
# List down all the customers with email containing "Example"

customers_with_example = customers_df[customers_df["Email"].str.contains("example",na=False)]

customers_with_example.head(300)
```

Out[65]:

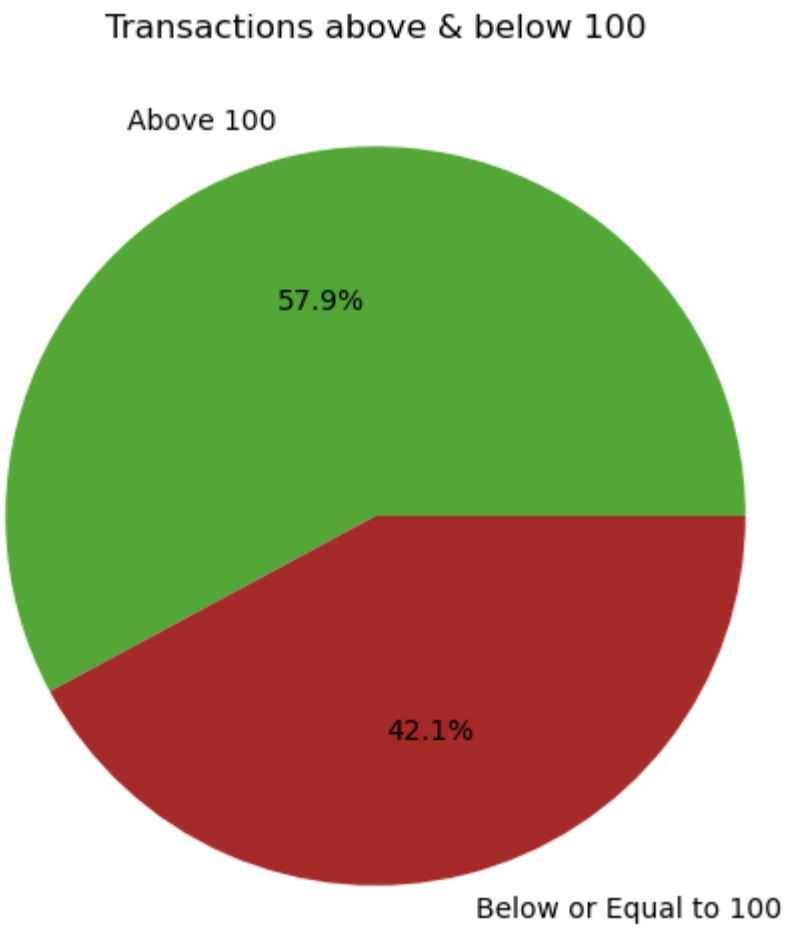
	CustomerID	FirstName	LastName	Email	PhoneNumber	JoinDate	Status	Region	JoinPeriod	Joine
0	1	John	Doe	john.doe@example.com	123-456-7890	2022-01-10	Active	North America	After	
1	2	Jane	Smith	jane.smith@example.com	098-765-4321	2021-12-15	Inactive	Europe	After	
2	3	Alice	Johnson	alice.j@example.com	567-890-1234	2020-03-20	Active	Asia	Before	
3	4	Bob	Brown	bob.brown@example.com	234-567-8901	2019-06-25	Inactive	North America	Before	
4	5	Charlie	Davis	charlie.d@example.com	345-678-9012	2021-07-14	Active	Europe	After	
...	...	...	...	...	...	...	...	...	...	
295	296	Emily	Brown	emily.brown@example.com	456-789-0123	2021-03-19	Inactive	Asia	After	
296	297	Liam	Wilson	liam.wilson@example.com	567-890-1234	2020-08-10	Active	North America	Before	
297	298	Olivia	Perry	olivia.perry@example.com	678-901-2345	2019-06-12	Inactive	Europe	Before	
298	299	Ella	Johnson	ella.johnson@example.com	789-012-3456	2022-05-18	Active	Asia	After	
299	300	Lucas	Lee	lucas.lee@example.com	890-123-4567	2021-11-01	Inactive	North America	After	

298 rows × 10 columns

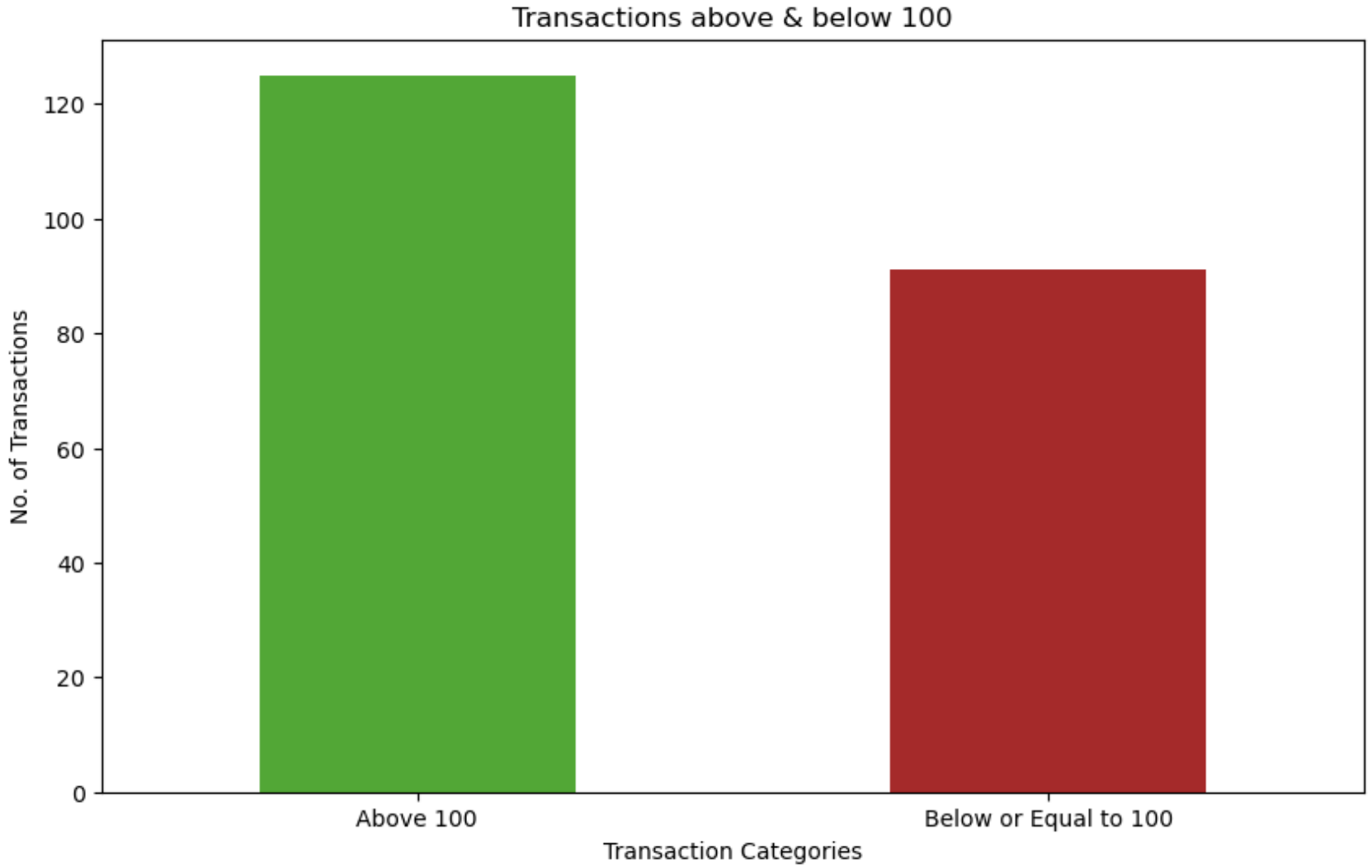
In [68]:
```
# Show the details where the subscriptions are Annual Plan. Then Visualise the subscriptions with Annual P
# and without plan

subscriptions_df['PlanCategory'] = subscriptions_df["PlanType"].apply(lambda x: "Annual" if x=="Annual" el

annual_counts=subscriptions_df['PlanCategory'].value_counts()


plt.figure(figsize=(10,6))
annual_counts.plot(kind="bar",color=['#52A736','#A52A2A'])
plt.title("Subscriptions: Annual Vs Other")

plt.xlabel("Plan")
plt.ylabel("No. of Subscriptions")
plt.xticks(rotation=0)
plt.show()
```


Subscriptions: Annual Vs Other
No. of Subscriptions
0
25
50
75
100
125
150
175
200
Annual
Other
Plan

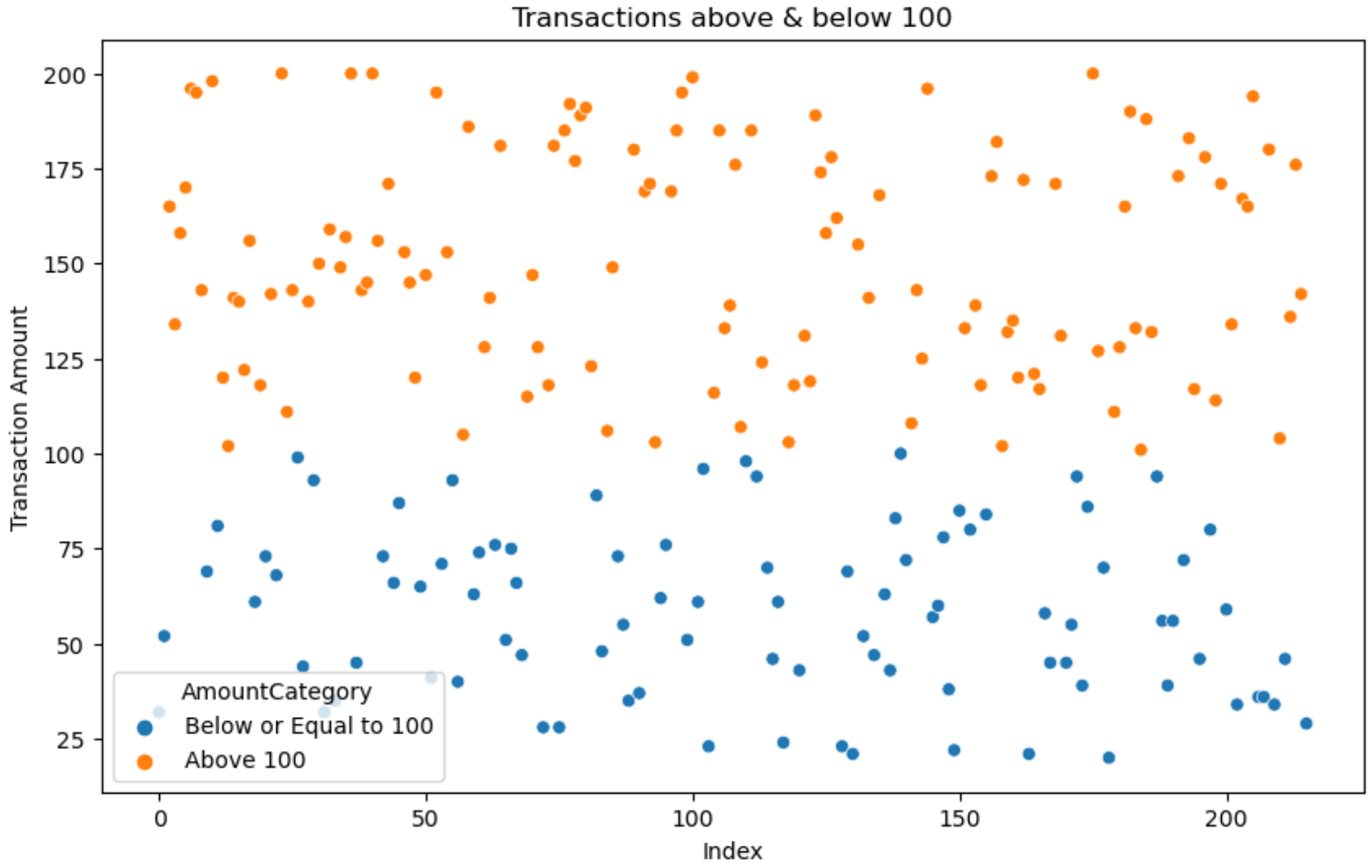
In [83]:
```
# Visualise the percentage of transactions above & below $100. Create pie chart & bar chart

#Create two categories : "Above 100" & "Below or Equal 100"

transactions_df["AmountCategory"] = transactions_df["Amount"].apply(lambda x:"Above 100" if x > 100 else "

category_counts = transactions_df["AmountCategory"].value_counts()

# Pie Chart

plt.figure(figsize=(10,6))

plt.pie(category_counts,labels=category_counts.index, autopct= "%1.1f%%", colors=['#52A736','#A52A2A'])
plt.title("Transactions above & below 100")
plt.show()
```



In [87]:
```
# Bar Chart

plt.figure(figsize=(10,6))

category_counts.plot(kind="bar",color=['#52A736','#A52A2A'])

plt.title("Transactions above & below 100")
plt.xlabel("Transaction Categories")
plt.ylabel("No. of Transactions")
plt.xticks(rotation=0)
plt.show()
```



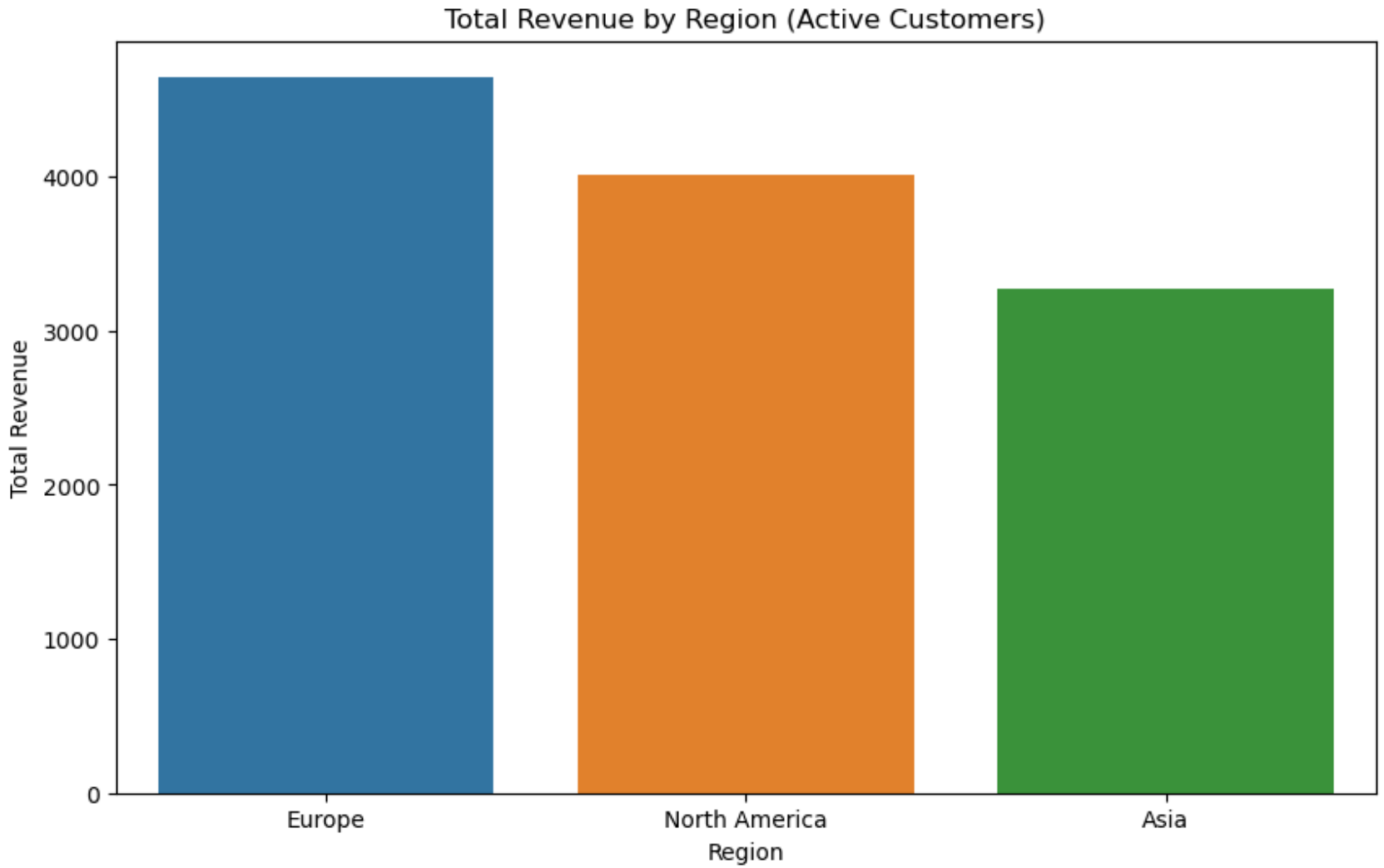
In [88]:
```
# Scatter Plot

plt.figure(figsize=(10,6))
sns.scatterplot(x= transactions_df.index, y="Amount", hue="AmountCategory", data=transactions_df)
plt.title("Transactions above & below 100")
plt.xlabel("Index")
plt.ylabel("Transaction Amount")

plt.show()
```



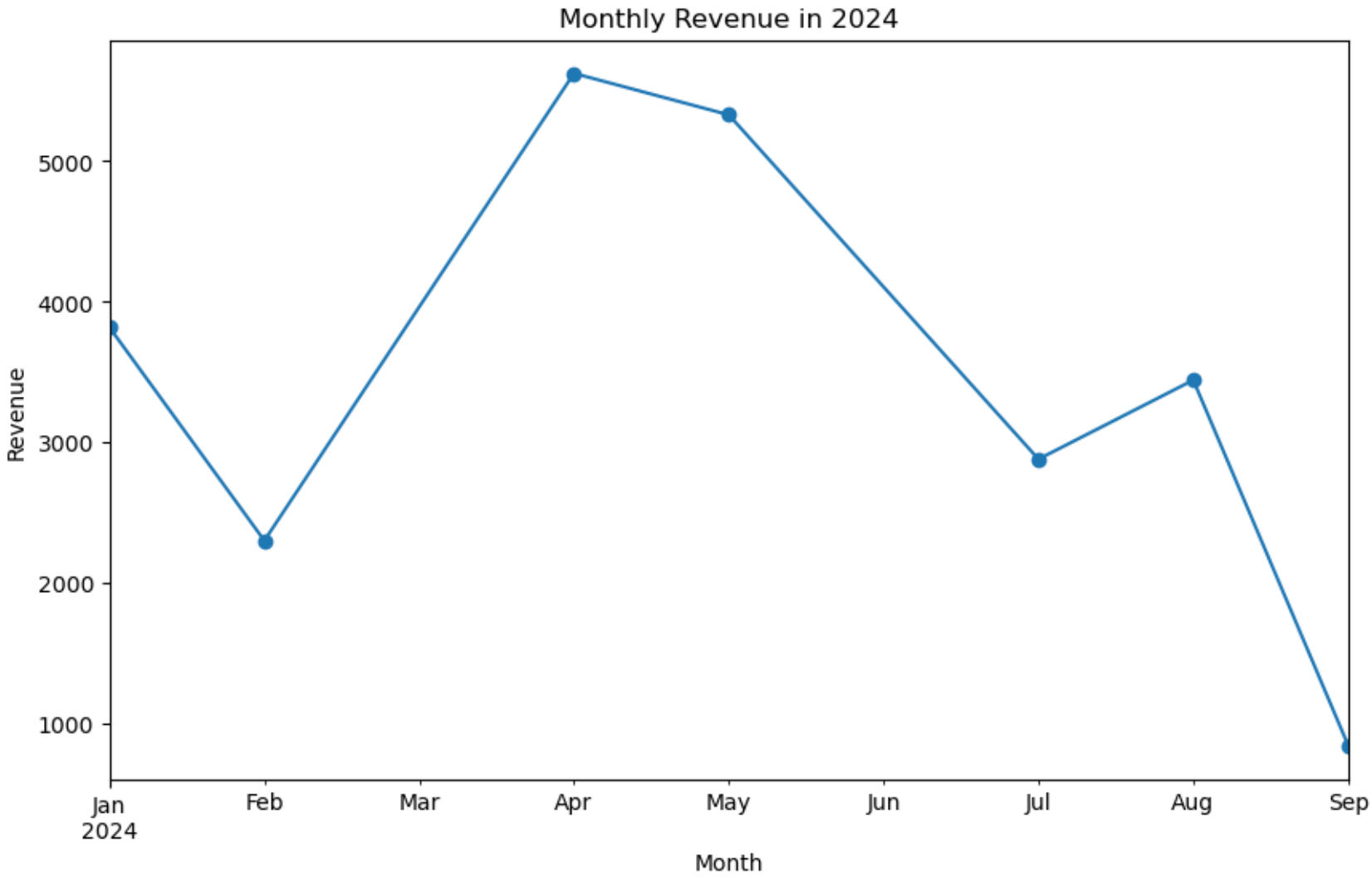
In [ ]:
```
# Find the 5 most recent transactions
```

In [94]:
```
#Which region has the total revenue generated from active customers ?

#Hint: Aggreate total transactions amount by region for active customers only.

active_customers = customers_df[customers_df["Status"]=="Active"]

active_transactions  = pd.merge(active_customers, transactions_df, on="CustomerID")

revenue_by_region = active_transactions.groupby("Region")["Amount"].sum().sort_values(ascending=False)

print(revenue_by_region)

#Visualization

plt.figure(figsize=(10,6))
sns.barplot(x=revenue_by_region.index,y=revenue_by_region.values)

plt.title("Total Revenue by Region (Active Customers)")
plt.xlabel("Region")
plt.ylabel("Total Revenue")
plt.show()
```

```
Region
Europe           4650
North America    4017
Asia             3278
Name: Amount, dtype: int64
```


Total Revenue by Region (Active Customers)
Total Revenue
0
1000
2000
3000
4000
Europe
North America
Asia
Region

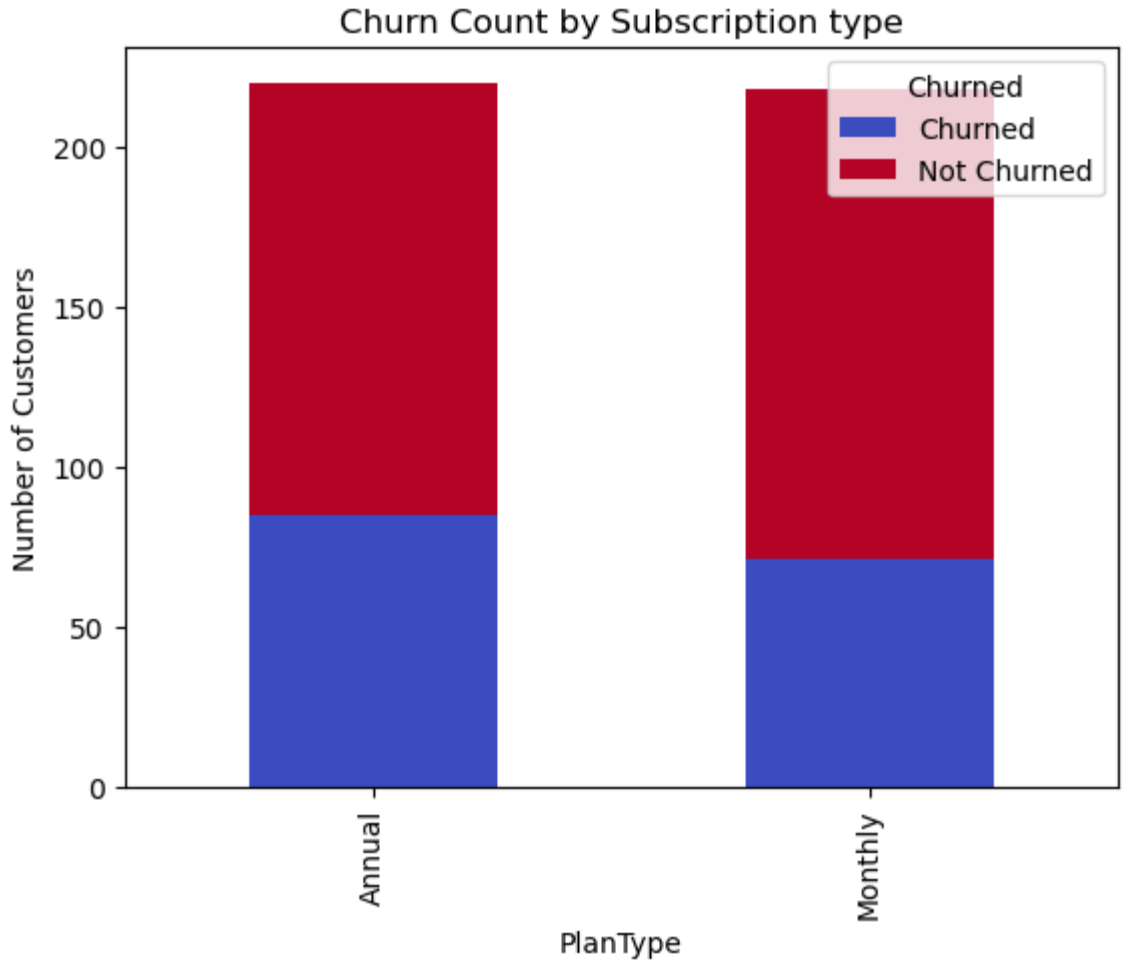
In [98]:
```
# Plot month-wise revenue trends for 2024

# Steps 1: Ensure transaction date is datetime format

transactions_df["TransactionDate"]= pd.to_datetime(transactions_df["TransactionDate"],errors="coerce", forı


# Step 2: Filter for year 2024 only

transactions_2024=transactions_df[transactions_df["TransactionDate'].dt.year==2024]


# step 3: Extract "Month" as (Year-Month)

transactions_2024["Month"] = transactions_2024["TransactionDate"].dt.to_period("M")


#step 4: group and sum amount by Amount

monthly_revenue = transactions_2024.groupby("Month")["Amount"].sum().sort_index()


#step 5: plotting

plt.figure(figsize=(10,6))

monthly_revenue.plot(kind="line", marker="o")
plt.title("Monthly Revenue in 2024")
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.show()
```


Monthly Revenue in 2024
Revenue
5000
4000
3000
2000
1000
Jan
2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Month

In [101…

```
# Find out the correlation between subscription type & churn.

#Hint : Find churn rate for Annual Vs Monthly subscribers

subscription_churn = pd.merge(subscriptions_df,churn_df, on = "CustomerID", how="left")

#We want to know which subscribed customers have churned by adding churn info beside their subscription.

subscription_churn["Churned"]= np.where (subscription_churn["ChurnID"].isna(), "Not Churned", "Churned")

#creates a new column named as "Churned"
#using where function to check: if "Churn ID" is missing (Nan), didn't churn ("Not Churned"), else "Churne


churn_by_plan = subscription_churn.groupby(["PlanType","Churned"]).size().unstack().fillna(0)

#size(): counts how many customers are in each group
#unstack(): reshapes the tables : churned values become column headers
#fillna(0): replaces any missing values with 0.

plt.figure(figsize=(10,6))
churn_by_plan.plot(kind="bar",stacked=True,colormap="coolwarm")

plt.title("Churn Count by Subscription type")

plt.ylabel("Number of Customers")

plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```


Churn Count by Subscription type
Churned
Churned
Not Churned
Number of Customers
0
50
100
150
200
Annual
Monthly
PlanType