```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
from google.colab import files
uploaded = files.upload()

import io
filename = list(uploaded.keys())[0]
customer_orders = pd.read_csv(io.BytesIO(uploaded[filename]))

customer_orders.head()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving customer_orders.csv to customer_orders (5).csv
```

	order_id	customer_id	order_date	order_amount	shipping_address	order_status
0	d27d8139-a252-4402-9fd0-23a85592285e	2824	2023-05-28	22.26	32181 Johnson Course Apt. 389, New Jamesside, ...	pending
1	31d3bfde-52dd-4d35-bca6-ee6bddf1eb60	5012	2022-05-23	119.37	65423 Garcia Light, West Melanieview, AS 06196	delivered
2	c381a67d-68e6-4d40-912d-77cdc343036f	2679	2023-11-19	341.58	84959 Janet Cape Apt. 413, South Joshuastad, G...	delivered

```
customer_orders['order_date'] = pd.to_datetime(customer_orders['order_date'])
customer_orders['cohort_month'] = customer_orders.groupby('customer_id')['order_date'].transform('min').dt.to_period('M
customer_orders['order_month'] = customer_orders['order_date'].dt.to_period('M')
customer_orders['cohort_index'] = (customer_orders['order_month'].dt.year - customer_orders['cohort_month'].dt.year) *
customer_orders[['customer_id', 'order_date', 'cohort_month', 'order_month', 'cohort_index']].head()
```

	customer_id	order_date	cohort_month	order_month	cohort_index
0	2824	2023-05-28	2021-03	2023-05	26
1	5012	2022-05-23	2022-05	2022-05	0
2	2679	2023-11-19	2021-10	2023-11	25
3	2424	2021-11-21	2021-05	2021-11	6
4	1488	2021-03-21	2021-03	2021-03	0

```
cohort_data = customer_orders.groupby(['cohort_month', 'cohort_index'])['customer_id'].nunique().reset_index()
cohort_counts = cohort_data.pivot(index='cohort_month', columns='cohort_index', values='customer_id')
cohort_counts
```

cohort_index	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61	62	63
cohort_month																					
2020-01	241.0	6.0	2.0	7.0	8.0	7.0	9.0	8.0	4.0	9.0	...	11.0	4.0	3.0	4.0	3.0	4.0	6.0	4.0	4.0	6.0
2020-02	228.0	4.0	8.0	6.0	6.0	5.0	5.0	11.0	10.0	3.0	...	4.0	7.0	11.0	8.0	6.0	4.0	5.0	7.0	9.0	NaN
2020-03	207.0	NaN	5.0	7.0	8.0	2.0	5.0	9.0	7.0	5.0	...	6.0	4.0	5.0	4.0	5.0	8.0	2.0	6.0	NaN	NaN
2020-04	200.0	2.0	7.0	10.0	5.0	2.0	5.0	6.0	9.0	3.0	...	3.0	9.0	4.0	6.0	1.0	4.0	5.0	NaN	NaN	NaN
2020-05	209.0	4.0	10.0	3.0	5.0	7.0	8.0	6.0	8.0	5.0	...	6.0	6.0	5.0	6.0	7.0	4.0	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2024-12	50.0	2.0	2.0	2.0	1.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-01	49.0	1.0	1.0	1.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-02	39.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-03	58.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-04	39.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

64 rows × 64 columns

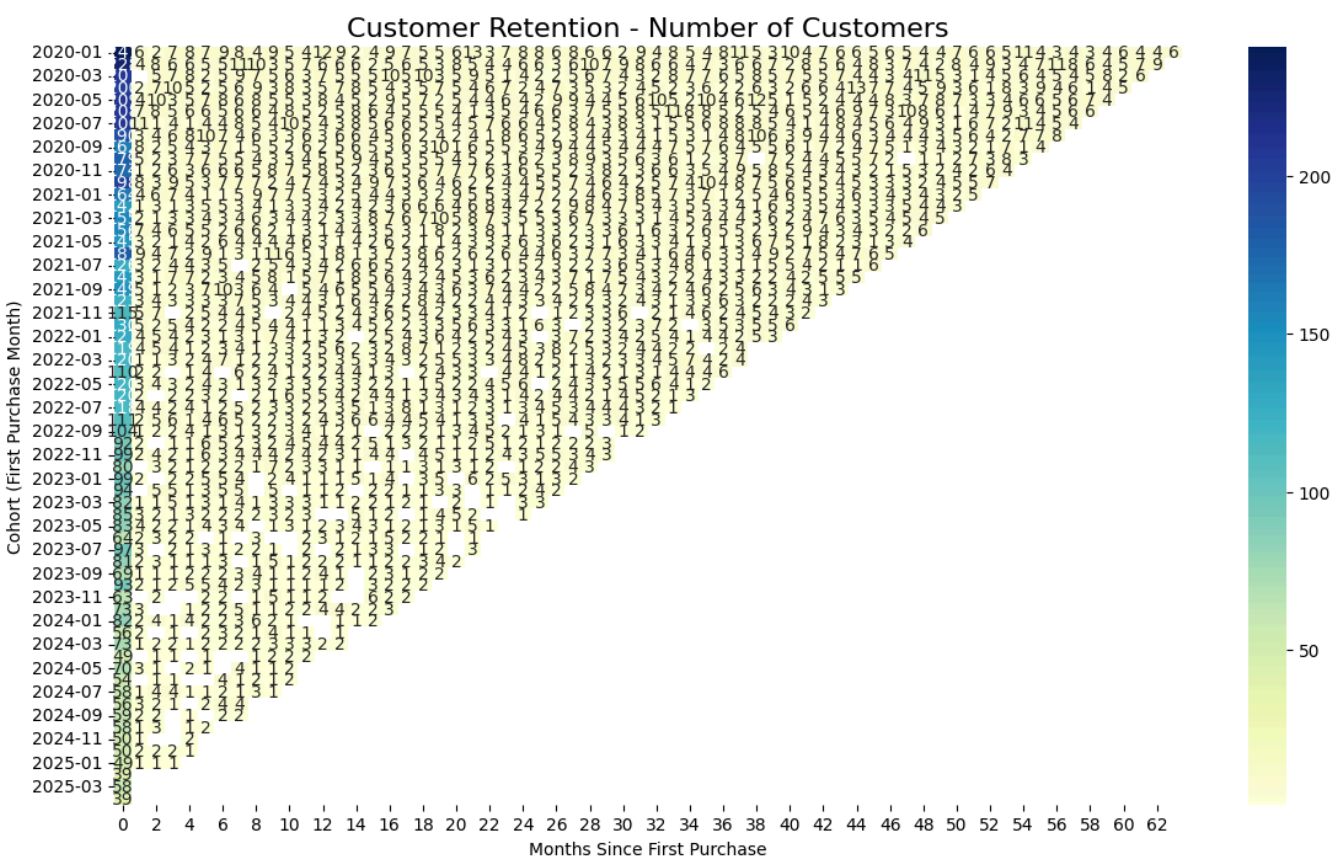
```
cohort_sizes = cohort_counts.iloc[:,0]
retention = cohort_counts.divide(cohort_sizes, axis=0).round(3) * 100

retention
```

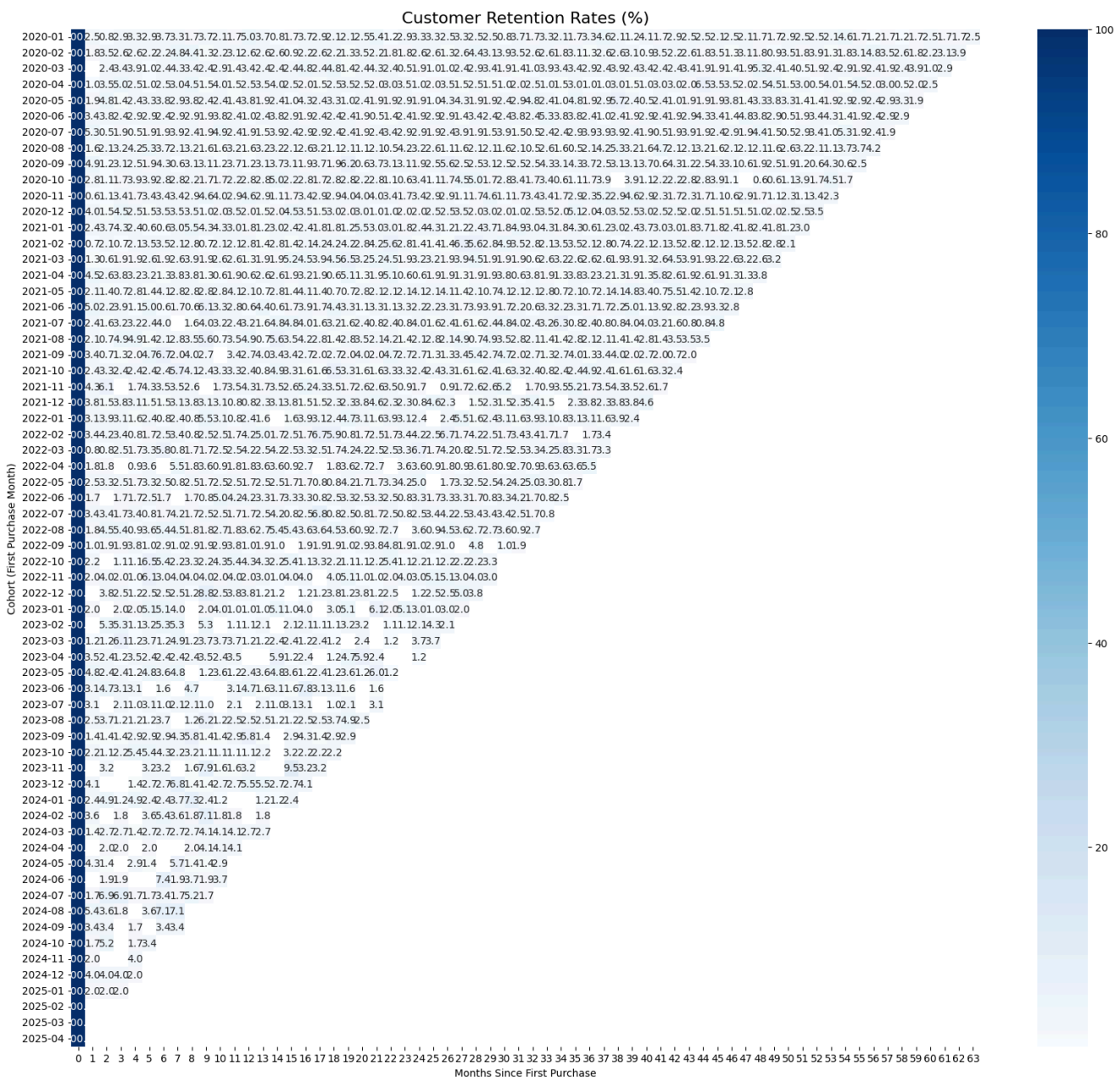
cohort_index	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61	62	63
cohort_month																					
2020-01	100.0	2.5	0.8	2.9	3.3	2.9	3.7	3.3	1.7	3.7	...	4.6	1.7	1.2	1.7	1.2	1.7	2.5	1.7	1.7	2.5
2020-02	100.0	1.8	3.5	2.6	2.6	2.2	2.2	4.8	4.4	1.3	...	1.8	3.1	4.8	3.5	2.6	1.8	2.2	3.1	3.9	NaN
2020-03	100.0	NaN	2.4	3.4	3.9	1.0	2.4	4.3	3.4	2.4	...	2.9	1.9	2.4	1.9	2.4	3.9	1.0	2.9	NaN	NaN
2020-04	100.0	1.0	3.5	5.0	2.5	1.0	2.5	3.0	4.5	1.5	...	1.5	4.5	2.0	3.0	0.5	2.0	2.5	NaN	NaN	NaN
2020-05	100.0	1.9	4.8	1.4	2.4	3.3	3.8	2.9	3.8	2.4	...	2.9	2.9	2.4	2.9	3.3	1.9	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2024-12	100.0	4.0	4.0	4.0	2.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-01	100.0	2.0	2.0	2.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-02	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-03	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2025-04	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

64 rows × 64 columns

```
plt.figure(figsize=(14,8))
sns.heatmap(cohort_counts, annot=True, fmt='.0f', cmap='YlGnBu')
plt.title('Customer Retention - Number of Customers', fontsize=16)
plt.xlabel('Months Since First Purchase')
plt.ylabel('Cohort (First Purchase Month)')
plt.show()
```



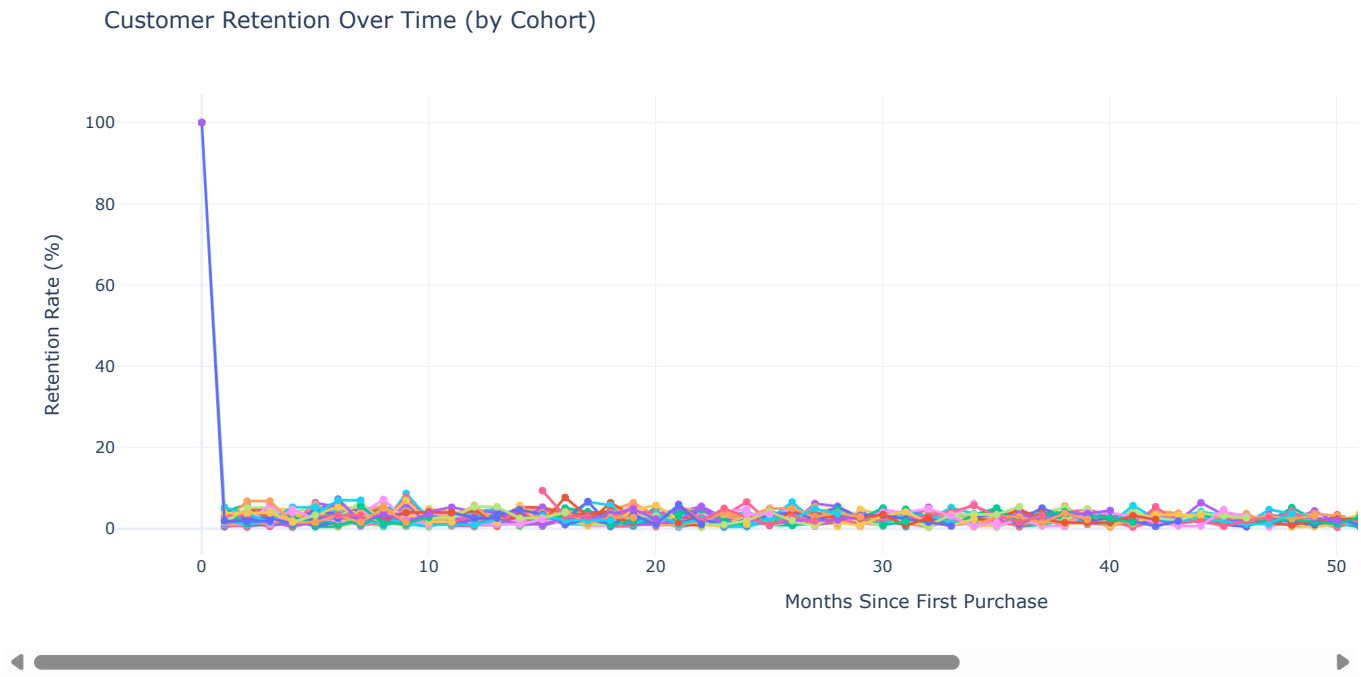

```
plt.figure(figsize=(20,18))
sns.heatmap(retention, annot=True, fmt='.1f', cmap='Blues')
plt.title('Customer Retention Rates (%)', fontsize=16)
plt.xlabel('Months Since First Purchase')
plt.ylabel('Cohort (First Purchase Month)')
plt.show()
```




```
retention_long = retention.reset_index().melt(id_vars='cohort_month')
retention_long.columns = ['CohortMonth', 'MonthsSinceFirstPurchase', 'RetentionRate']

fig = px.line(retention_long, x='MonthsSinceFirstPurchase', y='RetentionRate', color='CohortMonth', markers=True,
              title="Customer Retention Over Time (by Cohort)",
              labels={"MonthsSinceFirstPurchase": "Months Since First Purchase", "RetentionRate": "Retention Rate (%)"})

fig.update_layout(template="plotly_white")
fig.show()
```



Customer Retention Analysis Explanation. This visualization shows:

- Each row represents customers who made their first purchase in a specific month.
- Each column shows what percentage of those customers returned in subsequent months.
- A higher value means stronger customer loyalty.

```
Key Observations:
```

- The darker the color, the more customers stayed active over time.
- Rapid drops suggest customer churn issues after initial purchase.
- Trends can inform Alt Mobility's customer engagement and loyalty strategies.

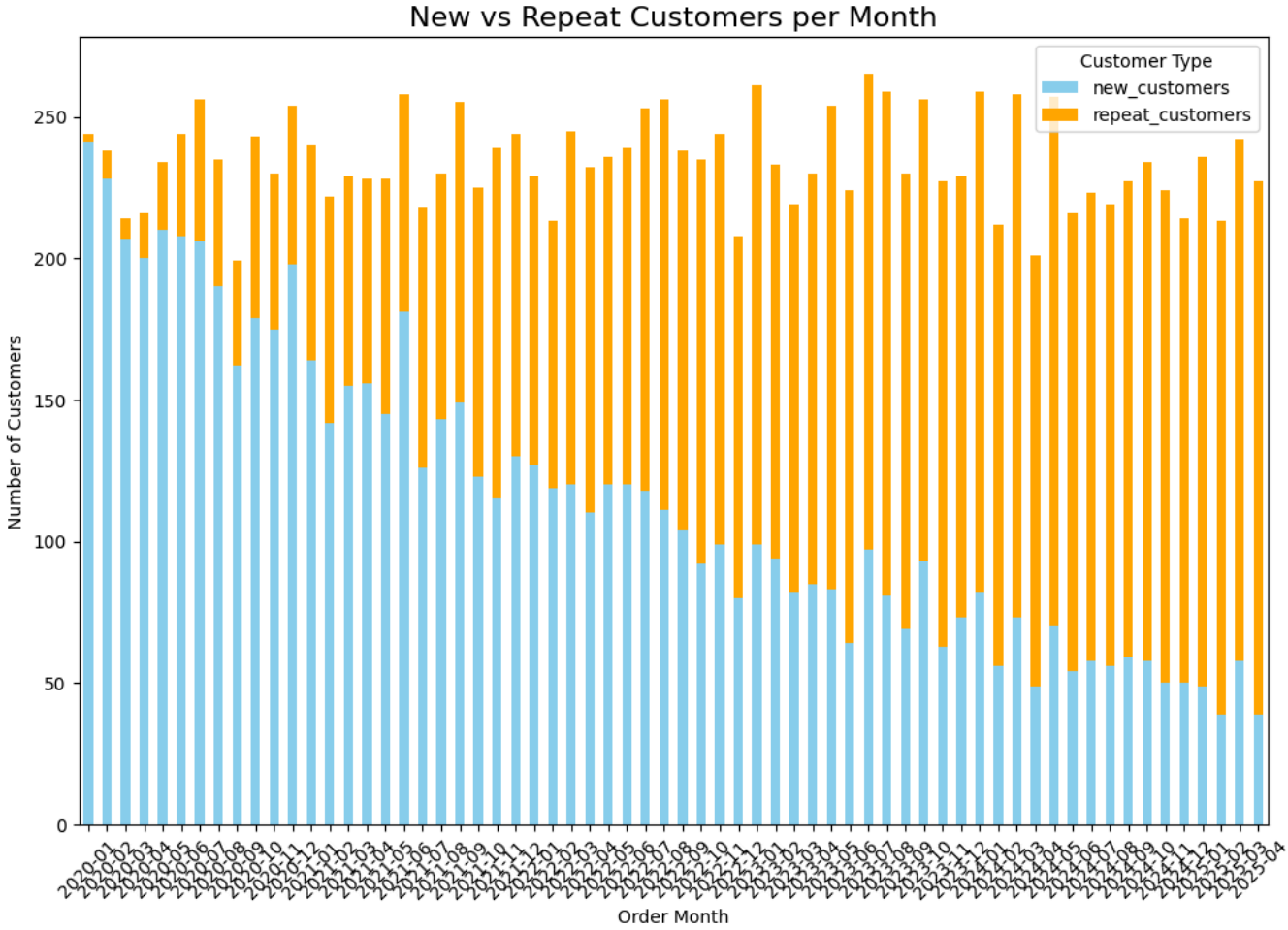
Why Heatmap + Line Chart Together:

- Heatmap gives a fast snapshot across all cohorts.
- Line chart shows how each cohort behaves over time.

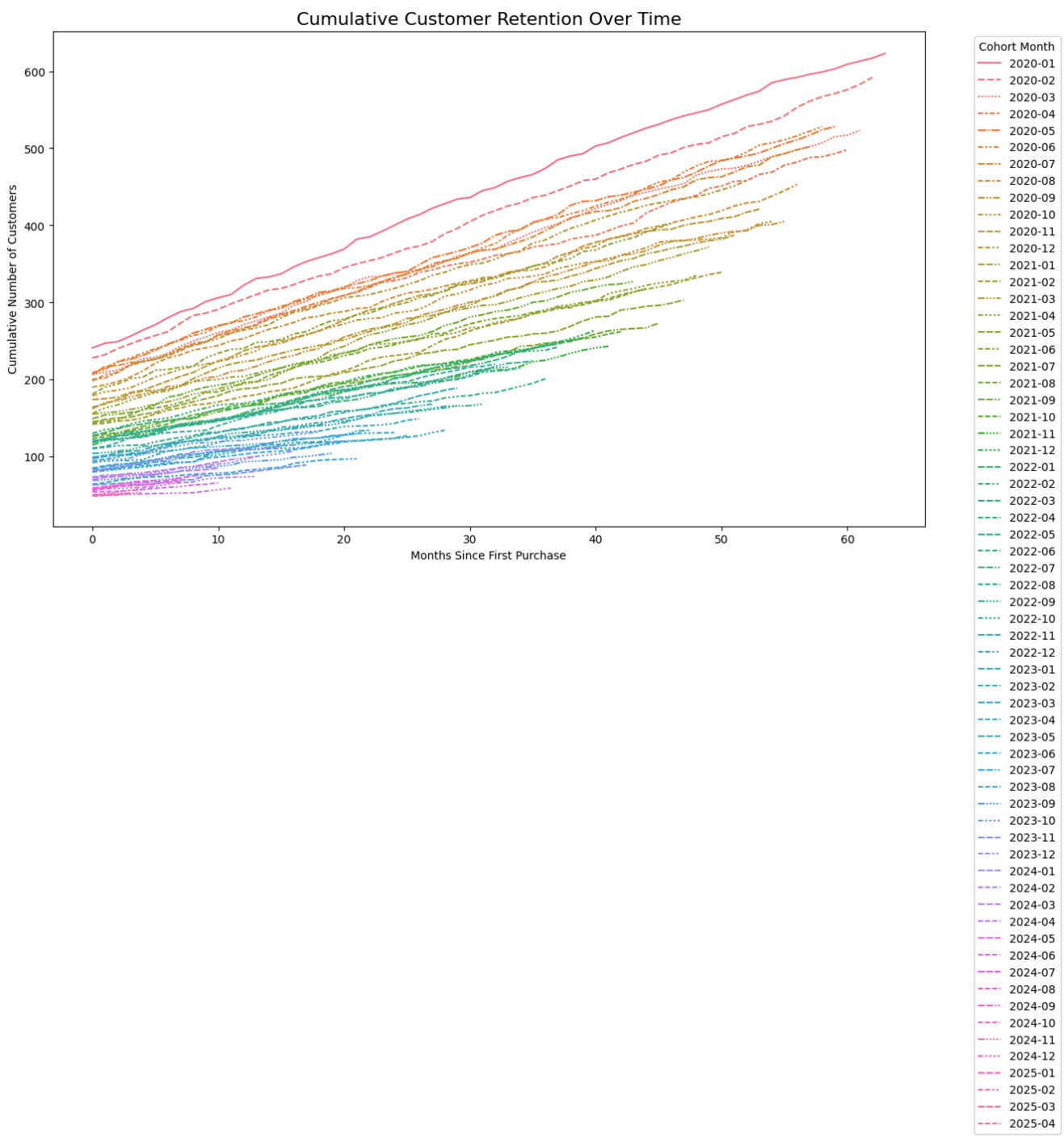
```
first_orders = customer_orders.groupby('customer_id')['order_date'].min().reset_index()
customer_orders['is_first_order'] = customer_orders['order_date'] == customer_orders['customer_id'].map(first_orders.se

monthly = customer_orders.groupby(customer_orders['order_month']).agg(
    new_customers = ('is_first_order', lambda x: x.sum()),
    repeat_customers = ('is_first_order', lambda x: (~x).sum())
).reset_index()

monthly.plot(x='order_month', y=['new_customers', 'repeat_customers'], kind='bar', stacked=True, figsize=(12,8), color=
plt.title('New vs Repeat Customers per Month', fontsize=16)
plt.ylabel('Number of Customers')
plt.xlabel('Order Month')
plt.xticks(rotation=45)
plt.legend(title='Customer Type')
plt.show()
```



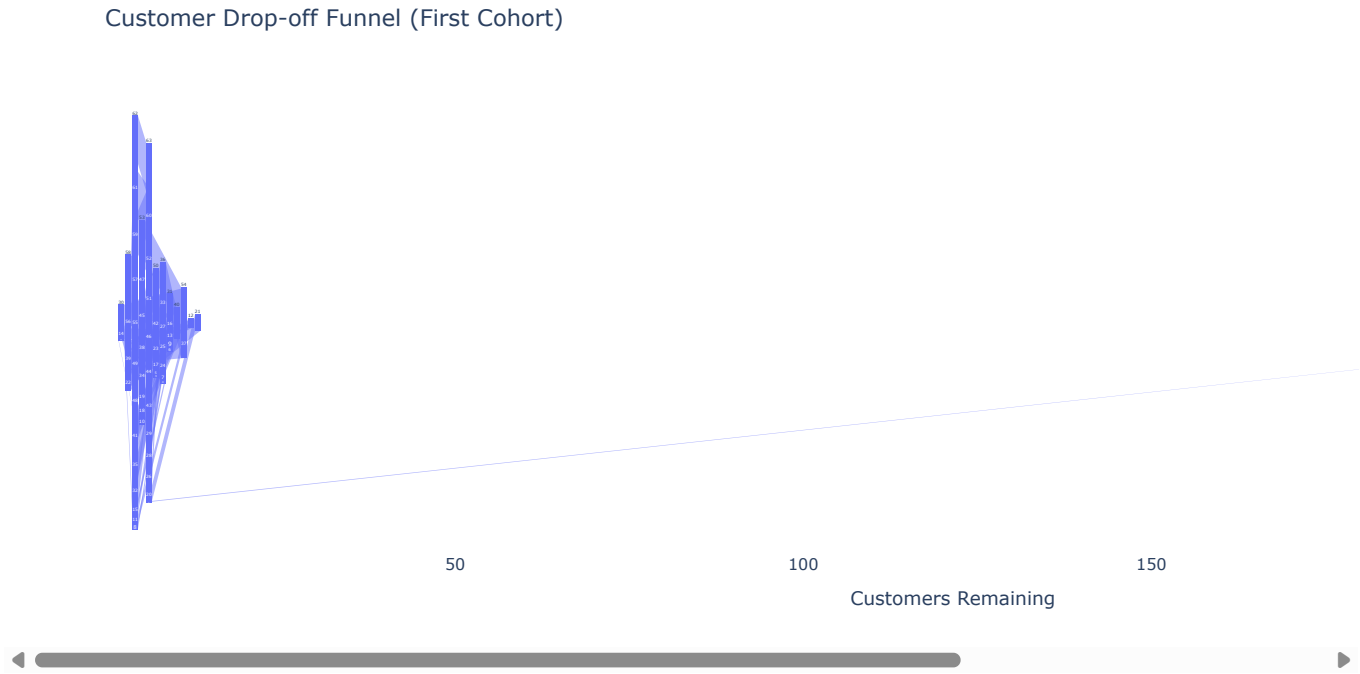

```
cumulative_retention = cohort_counts.cumsum(axis=1)
plt.figure(figsize=(14,8))
sns.lineplot(data=cumulative_retention.transpose())
plt.title('Cumulative Customer Retention Over Time', fontsize=16)
plt.xlabel('Months Since First Purchase')
plt.ylabel('Cumulative Number of Customers')
plt.legend(title='Cohort Month', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



Start coding or generate with AI.

```
sample_cohort = cohort_counts.iloc[0]
sample_cohort = sample_cohort.dropna()
```

```
fig = px.funnel(sample_cohort.reset_index(),
                x=sample_cohort.values,
                y=sample_cohort.index,
                title='Customer Drop-off Funnel (First Cohort)',
                labels={'y':'Months Since First Purchase', 'x':'Customers Remaining'})
fig.update_layout(template="plotly_white")
fig.show()
```




```
cohort_size = customer_orders.groupby('cohort_month')['customer_id'].nunique().reset_index()

cohort_size['cohort_month'] = cohort_size['cohort_month'].astype(str)

fig = px.area(cohort_size,
              x='cohort_month',
              y='customer_id',
              title='New Customers Acquired per Month (Cohort Size)',
              labels={'cohort_month': 'First Purchase Month', 'customer_id': 'Number of Customers'})
fig.update_layout(template="plotly_white")
fig.show()
```