Customer Life time value Descriptive

based on cohort

- CLV_Churn
- CLV

In [ ]:

```
import pandas as pd
import plotly.express as px
```

In [ ]:

```
df=pd.read_csv("./Dataset/Cleaned_Online_retail_dec2010-dec2011.csv",dtype_backend="pyarrow")
df.head()
```

Out[ ]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Revenue
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.30
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397885 entries, 0 to 397884
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      397885 non-null  int64[pyarrow]
 1   StockCode    397885 non-null  string[pyarrow]
 2   Description  397885 non-null  string[pyarrow]
 3   Quantity     397885 non-null  int64[pyarrow]
 4   InvoiceDate  397885 non-null  string[pyarrow]
 5   Price        397885 non-null  double[pyarrow]
 6   Customer ID  397885 non-null  int64[pyarrow]
 7   Country      397885 non-null  string[pyarrow]
 8   Revenue      397885 non-null  double[pyarrow]
dtypes: double[pyarrow](2), int64[pyarrow](3), string[pyarrow](4)
memory usage: 45.5 MB
```

In [ ]:

```
df["InvoiceDate"]=pd.to_datetime(df["InvoiceDate"])
```

In [ ]:

```
#Adding customers first transaction date
df.loc[:,"Start_month"]=df.groupby("Customer ID")["InvoiceDate"].transform(lambda x: x.min())
```

In [ ]:

```
#finding difference of every transaction from startdate of customers in months
from dateutil.relativedelta import relativedelta
```

```
def diffdates(x,y):
    return int(abs((relativedelta(x,y).years*12)+relativedelta(x,y).months))


df.loc[:,"Months_Since_Join"]=df.apply(lambda x:diffdates(x.Start_month,x.InvoiceDate ),axis=1)
df.head()
```

Out[ ]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Revenue	Start_month	Months_Since_Join
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.30	2010-12-01 08:26:00	0
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34	2010-12-01 08:26:00	0
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	22.00	2010-12-01 08:26:00	0
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34	2010-12-01 08:26:00	0
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34	2010-12-01 08:26:00	0

In [ ]:

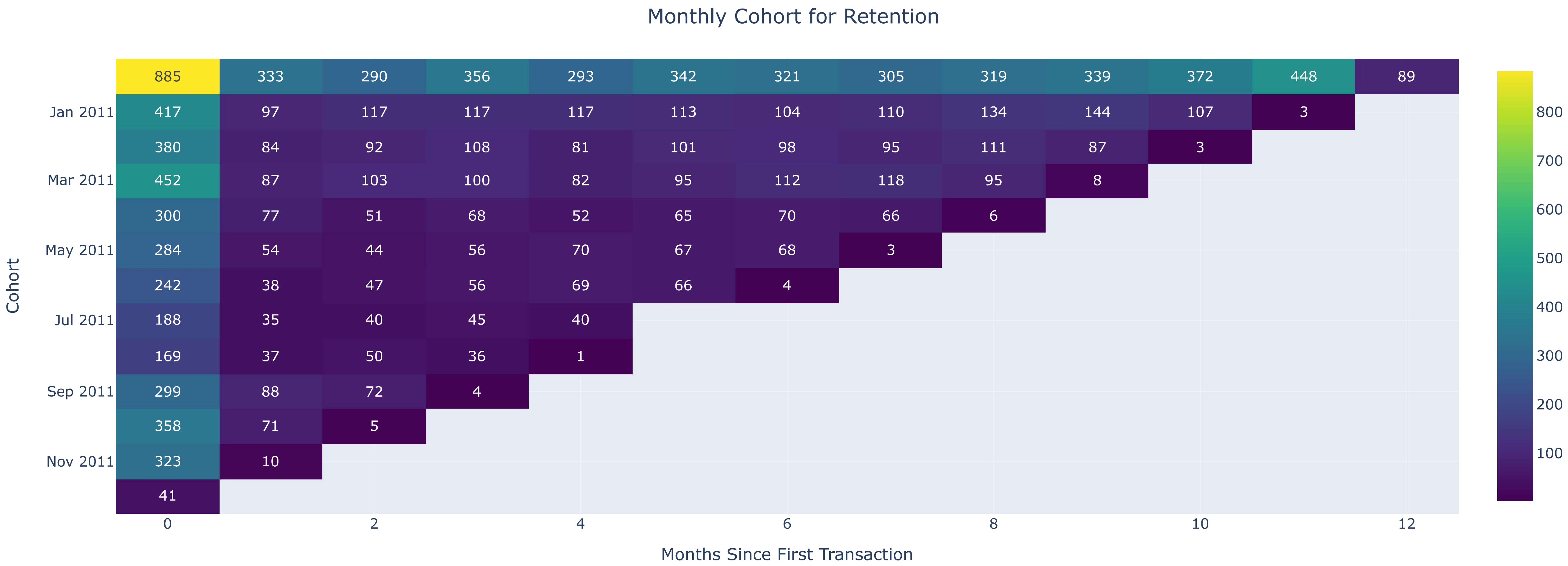
```
df.loc[:,"Start_month"]=df["Start_month"].transform(lambda x: x.strftime("%Y-%m"))
df.head()
```

Out[ ]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Revenue	Start_month	Months_Since_Join
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.30	2010-12-01	0
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34	2010-12-01	0
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	22.00	2010-12-01	0
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34	2010-12-01	0
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34	2010-12-01	0

In [ ]:

```
#Cohort matrix with retention values
cohort_visual_df=df.groupby(["Start_month","Months_Since_Join"])["Customer ID"].apply(pd.Series.nunique).reset_index()
fig=px.imshow(cohort_visual_df.pivot(index="Start_month",columns="Months_Since_Join",values="Customer ID"),color_continuous_scale="viridis",text_auto=True)
fig.update_layout(xaxis_title="Months Since First Transaction",yaxis_title="Cohort",title={
            'text' : 'Monthly Cohort for Retention',
            'x':0.5,
            'xanchor': 'center'
        })
fig.show("notebook")
```



In [ ]:

```
cohort=df.groupby(["Start_month","Customer ID"]).agg(lifespan=('InvoiceDate',lambda x:(x.max()-x.min()).days),
                                                      Frequency=('Invoice','nunique'),
                                                      Total_sales=('Revenue','sum'),
                                                      Avg_sales=('Revenue','mean'),



).reset_index()
cohort.head()
```

Out[ ]:

	Start_month	Customer ID	lifespan	Frequency	Total_sales	Avg_sales
0	2010-12-01	12347	365	7	4310.00	23.681319
1	2010-12-01	12348	282	4	1797.24	57.975484
2	2010-12-01	12370	309	4	3545.69	21.231677
3	2010-12-01	12377	39	2	1628.12	21.144416
4	2010-12-01	12383	167	5	1850.56	18.692525

In [ ]:

```
cohort_summary=cohort.groupby("Start_month").agg(cohort_size=('Customer ID','nunique'),
                                                 Avg_sales=('Avg_sales','mean'),
                                                 Avg_Frequency=('Frequency','mean'),
                                               Churn_rate=('Frequency',lambda x:(1-(x>1).sum()/len(x))),
                                               Avg_lifespan=('lifespan','mean')
                                               ).reset_index()

cohort_summary
```

Out[ ]:

	Start_month	cohort_size	Avg_sales	Avg_Frequency	Churn_rate	Avg_lifespan
0	2010-12-01	885	36.422175	9.398870	0.125424	267.748023
1	2011-01-01	417	229.024754	5.163070	0.184652	208.083933
2	2011-02-01	380	36.631088	4.107895	0.228947	171.002632
3	2011-03-01	452	28.076714	3.564159	0.278761	140.685841
4	2011-04-01	300	27.500523	3.080000	0.333333	113.050000
5	2011-05-01	284	241.196904	2.883803	0.306338	99.954225
6	2011-06-01	242	83.920088	2.731405	0.355372	82.946281
7	2011-07-01	188	29.477879	2.351064	0.393617	57.186170
8	2011-08-01	169	32.904727	2.118343	0.455621	42.366864
9	2011-09-01	299	35.094298	1.989967	0.521739	26.655518
10	2011-10-01	358	25.035718	1.698324	0.653631	11.645251
11	2011-11-01	323	25.476888	1.359133	0.736842	3.597523
12	2011-12-01	41	153.916796	1.048780	0.975610	0.000000

Churn CLV= (Avg Sales * Avg Frquency) / Churn Rate

Assume Profit Margin= 0.10 (10%)

In [ ]:

```
cohort_summary["CLV_Churn"]=((cohort_summary["Avg_sales"]*cohort_summary["Avg_Frequency"])/cohort_summary["Churn_rate"])*0.10
```

Lifetime CLV = (Avg Sales * Avg Frequency per year * Customer Lifetime)

since we have only 1year data if we expect customer lifetime = 2years

In [ ]:

```
cohort_summary["CLV_Lifetime"]=cohort_summary["Avg_sales"]*cohort_summary["Avg_Frequency"]*2*0.10
cohort_summary
```

Out[ ]:

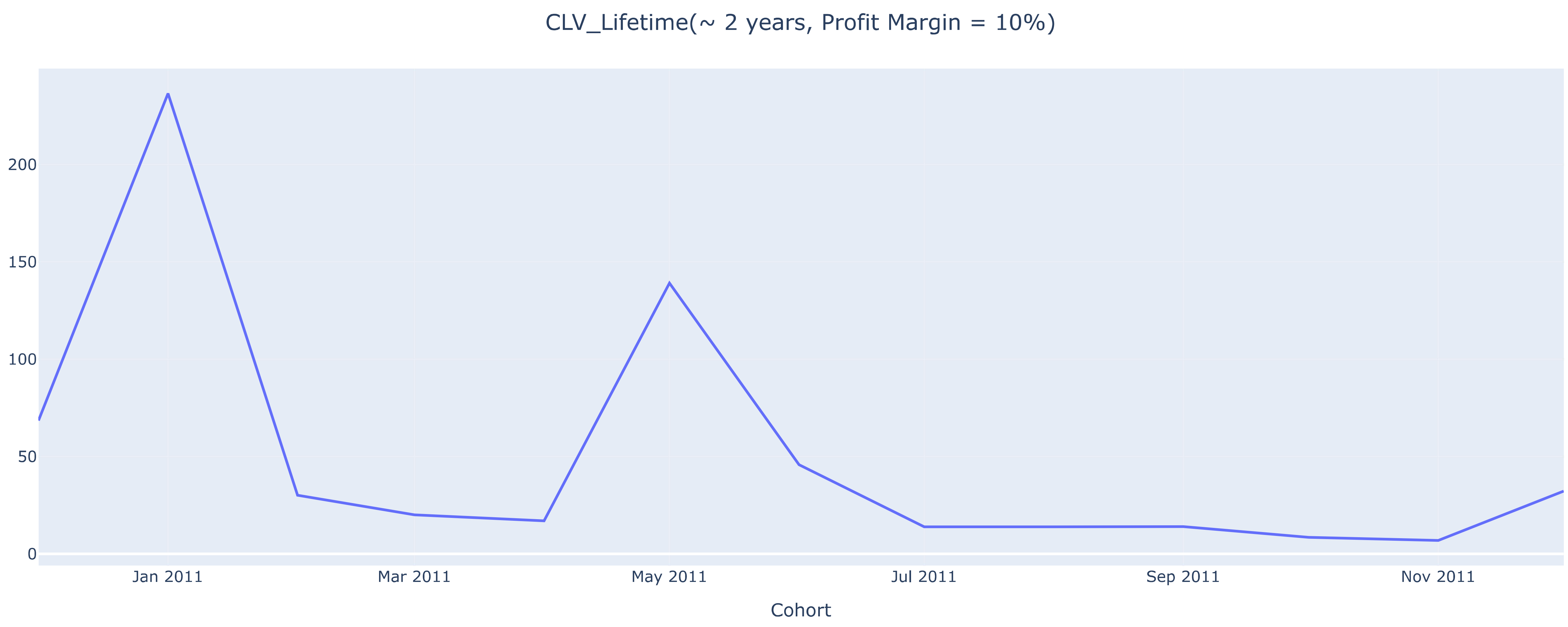
	Start_month	cohort_size	Avg_sales	Avg_Frequency	Churn_rate	Avg_lifespan	CLV_Churn	CLV_Lifetime
0	2010-12-01	885	36.422175	9.398870	0.125424	267.748023	272.936620	68.465457
1	2011-01-01	417	229.024754	5.163070	0.184652	208.083933	640.377007	236.494146
2	2011-02-01	380	36.631088	4.107895	0.228947	171.002632	65.725435	30.095331
3	2011-03-01	452	28.076714	3.564159	0.278761	140.685841	35.898084	20.013976
4	2011-04-01	300	27.500523	3.080000	0.333333	113.050000	25.410484	16.940322
5	2011-05-01	284	241.196904	2.883803	0.306338	99.954225	227.057775	139.112862
6	2011-06-01	242	83.920088	2.731405	0.355372	82.946281	64.501370	45.843949
7	2011-07-01	188	29.477879	2.351064	0.393617	57.186170	17.607058	13.860875
8	2011-08-01	169	32.904727	2.118343	0.455621	42.366864	15.298561	13.940701
9	2011-09-01	299	35.094298	1.989967	0.521739	26.655518	13.385325	13.967296
10	2011-10-01	358	25.035718	1.698324	0.653631	11.645251	6.505007	8.503752
11	2011-11-01	323	25.476888	1.359133	0.736842	3.597523	4.699308	6.925296
12	2011-12-01	41	153.916796	1.048780	0.975610	0.000000	16.546056	32.284987

from above data

- CLV_Lifetime= expected avg CLV from each Customer for 2years
- CLV_Churn= Avg CLV From Each Customers with Churn

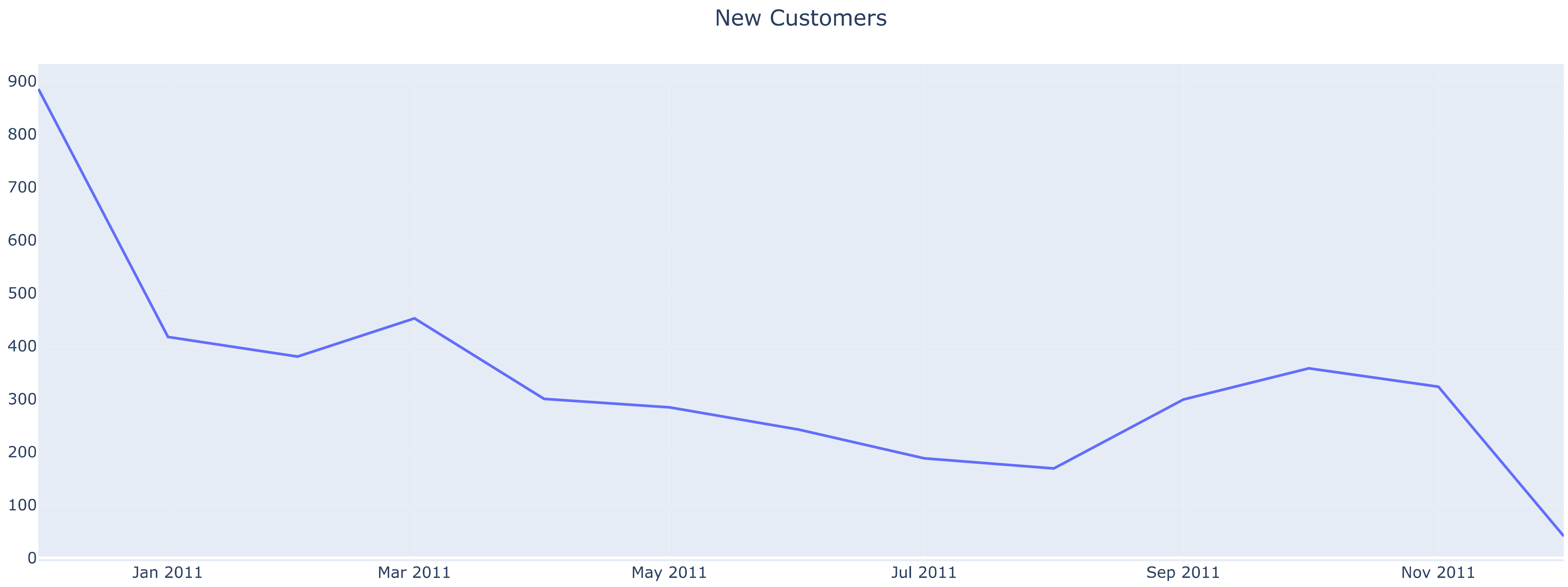
In [ ]:

```
fig=px.line(cohort_summary,x="Start_month",y="CLV_Lifetime")
fig.update_layout(xaxis_title="Cohort",yaxis_title="",title={
            'text' : 'CLV_Lifetime(~ 2 years, Profit Margin = 10%) ',
            'x':0.5,
            'xanchor': 'center'
        })
fig.show("notebook")
```



In [ ]:

```
fig=px.line(cohort_summary,x="Start_month",y="cohort_size")
fig.update_layout(xaxis_title="",yaxis_title="",title={
            'text' : 'New Customers',
            'x':0.5,
            'xanchor': 'center'
        })
fig.show("notebook")
```


New Customers
900
800
700
600
500
400
300
200
100
0
Jan 2011
Mar 2011
May 2011
Jul 2011
Sep 2011
Nov 2011