CHURN COHORT ANALYSIS

Using a sample data of 541,909 records

About The Dataset

Context

This is a transnational dataset, which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based online retail business.

The company mainly sells unique all-occasion gifts.

Many customers of the company are wholesalers.

Data Source:

The data was downloaded from University of California Irvine (UCI) website, curtesy of Dr. Daqing Chen, Director of Public Analytics group. chend '@' lsbu.ac.uk, School of Engineering, London South Bank University, London SE1 0AA, UK.

The project based on guide by Angelina Frimpong's YouTube teaching with link here.

Data Attributes:

- ***InvoiceNo:*** Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- ***StockCode:*** Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- ***Description:*** Product (item) name. Nominal.
- ***Quantity:*** The quantities of each product (item) per transaction. Numeric.
- ***InvoiceDate:*** Invoice Date and time. Numeric, the day and time when each transaction was generated.
- ***UnitPrice:*** Unit price. Numeric, Product price per unit in sterling.
- ***CustomerID:*** Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

- Country: Country name. Nominal, the name of the country where each customer resides.

The dataset was an Excel file, which I converted to CSV for easy manipulation using data tools. There were a total of 541,909 records.

Here is a peep look at the dataset structure:

	A	B	C	D	E	F	G	H	I
1	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	
2	536365	85123A	WHITE HANG	6	12/1/10	2.55	17850	United Kingdom	
3	536365	71053	WHITE META	6	12/1/10	3.39	17850	United Kingdom	
4	536365	84406B	CREAM CUPI	8	12/1/10	2.75	17850	United Kingdom	
5	536365	84029G	KNITTED UNI	6	12/1/10	3.39	17850	United Kingdom	
6	536365	84029E	RED WOOLLY	6	12/1/10	3.39	17850	United Kingdom	
7	536365	22752	SET 7 BABUS	2	12/1/10	7.65	17850	United Kingdom	
8	536365	21730	GLASS STAR	6	12/1/10	4.25	17850	United Kingdom	
9	536366	22633	HAND WARM	6	12/1/10	1.85	17850	United Kingdom	
10	536366	22632	HAND WARM	6	12/1/10	1.85	17850	United Kingdom	
11	536367	84879	ASSORTED C	32	12/1/10	1.69	13047	United Kingdom	
12	536367	22745	POPPY'S PLA	6	12/1/10	2.1	13047	United Kingdom	
13	536367	22748	POPPY'S PLA	6	12/1/10	2.1	13047	United Kingdom	
14	536367	22749	FELTCRAFT P	8	12/1/10	3.75	13047	United Kingdom	

Objective Of The Analysis

The main objective of this churn analysis project is to evaluate a company's customer **churn** (**loss) rate** in order **to reduce it**.

What is App Churn Rate?

Churn rate, also known as the rate of attrition, is the percentage of users who stopped patronizing a company within a given period.

Why Churn Rate Matters

Churn rate suppresses growth. It's the leaky bucket analogy: as customers drop off, the company will keep struggling to refill its bucket by adding new (acquisition) customers. Facts to note:

- Acquiring a new customer is 5 to 25 times more expensive than retaining one;
- Reducing churn by just 5% can boost profitability by 75%;
- Improving retention has about 4 times greater impact on growth than acquisition;
- The probability of selling to an existing customer is 60 to 70%, but only an average of 15% for a new prospect.

This is why it is important to be mindful of churn rate and retentions rate.

Data Analytic Tool

For this project, **SQL - CTE** (**via BigQuery platform**) and **Tableau** were used.

Data Preparations

The downloaded dataset named "online_retail.xlsx" was first converted to a CSV file called "online_retail.csv".

The csv file was imported into Google cloud-based BigQuery, where SQL was used inspect and clean the dataset. The following are some of the cleaning done:

- Checking the records, 135, 080 were found without CustomerID, which is very vital, and 406,829 records had the needed CustomerID. These records with no CustomerID were removed.
- Also, out of these 406,829 records, 397,884 records had Quantity and Unit Price, which were also needed.
- Finally, some records were duplicated. These were a total 5, 215 records and there were removed, leaving us with 392, 669 clean records.

The SQL codes used to clean these records are:

```
---
#####CLEAN RECORD
WITH retails AS
(
 -- First we get the records that has CustomerID
 SELECT InvoiceNo
    ,StockCode
    ,Description
    ,Quantity
    ,InvoiceDate
    ,UnitPrice
    ,CustomerID
    ,Country
 FROM `churn-370413.online_retail.retail`
```

```
 WHERE CustomerID != 0
),
wtQnty_n_price AS
(
 --Let us get records that have customerID, Quantity and Unit Price
 SELECT *
 FROM retails
 WHERE Quantity > 0 AND UnitPrice > 0
),
check_duplicate AS
(
 -- Let us check and remove the duplicated records
 SELECT * , ROW_NUMBER() OVER (PARTITION BY InvoiceNo, StockCode, Quantity
 ORDER BY InvoiceDate)dup_flag
 FROM wtQnty_n_price
)

SELECT *
FROM check_duplicate
WHERE dup_flag = 1
---
```

Data Processing & Analysis

Generating Cohort Values

"Cohort analysis is an analytical technique that categorizes and divides data into groups with common characteristics prior to analysis. "

We need 4 main values to successfully generate cohort analysis values. These are:

1. Unique Identifier (CustomerID)
2. Initial Start Date (First Invoice Date)
3. Revenue Data
4. Cohort Index

Out of these, we have the CustomerID and Revenue Data in place. Using SQL we need derive the remaining 2 values.

We need the **cohort index**, this is the number of months since the first transaction for each customer.

```
---
#####BEGIN COHORT ANALYSIS

SELECT
     CustomerID,
     min(InvoiceDate) first_purchase_date,
     DATEFROMPARTS(year(min(InvoiceDate)),
month(min(InvoiceDate)), 1) Cohort_Date
INTO #cohort_table
FROM #clean_records
GROUP BY CustomerID


---Create Cohort Index
SELECT
 mmm.*,
 (year_diff * 12 + month_diff + 1) AS cohort_index
FROM
 (
  SELECT
   mm.*,
   (invoice_year - cohort_year) AS year_diff,
   (invoice_month - cohort_month) AS month_diff ,
  FROM
   (
    SELECT
     m.*,
     c.Cohort_Date,
     extract(YEAR FROM date(m.InvoiceDate)) invoice_year,
     extract(MONTH FROM date(m.InvoiceDate)) invoice_month,
     extract(YEAR FROM date(c.Cohort_Date)) cohort_year,
     extract(MONTH FROM date(c.Cohort_Date)) cohort_month
    FROM `churn-370413.online_retail.clean_record` m
    LEFT JOIN `churn-370413.online_retail.cohort_table` c
     ON m.CustomerID = c.CustomerID
   )mm
 )mmm
---
```

Here is the output:

		InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	dup_flag	Cohort_Date	invoice_year	invoice_mon	cohort_year	cohort_mon	year_diff	month_diff	cohort_index
2	0	540016	21080	SET/20 RED f	8	1/4/11	0.85	16282	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
3	1	540019	22139	RETROSPOT	3	1/4/11	4.95	12957	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
4	2	540059	21609	SET 12 LAVEI	96	1/4/11	2.55	17457	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
5	3	540188	72741	GRAND CHO(	9	1/5/11	1.45	16771	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
6	4	540247	21377	SMALL CAMI	6	1/5/11	1.65	15464	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
7	5	540247	21416	CLAM SHELL	1	1/5/11	3.75	15464	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
8	6	540267	21238	RED RETROS	96	1/6/11	0.72	12415	Australia	1	1/1/11	2011	1	2011	1	0	0	1
9	7	540267	84228	HEN HOUSE	120	1/6/11	0.37	12415	Australia	1	1/1/11	2011	1	2011	1	0	0	1
10	8	540278	20973	12 PENCIL SM	2	1/6/11	0.65	15719	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
11	9	540279	22910	PAPER CHAI	12	1/6/11	2.95	13368	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
12	10	540353	21931	JUMBO STOF	2	1/6/11	1.95	13764	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
13	11	540353	22113	GREY HEART	1	1/6/11	3.75	13764	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
14	12	540362	21669	BLUE STRIPE	2	1/6/11	1.25	13656	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
15	13	540373	84378	SET OF 3 HEA	3	1/6/11	1.25	13280	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
16	14	540458	21238	RED RETROS	8	1/7/11	0.85	12501	Germany	1	1/1/11	2011	1	2011	1	0	0	1
17	15	540458	21245	GREEN POLK	8	1/7/11	1.69	12501	Germany	1	1/1/11	2011	1	2011	1	0	0	1
18	16	540458	21980	PACK OF 12 f	24	1/7/11	0.29	12501	Germany	1	1/1/11	2011	1	2011	1	0	0	1
19	17	540469	22561	WOODEN SC	6	1/7/11	1.65	12484	Spain	1	1/1/11	2011	1	2011	1	0	0	1
20	18	540469	84251B	GREETING CA	12	1/7/11	0.19	12484	Spain	1	1/1/11	2011	1	2011	1	0	0	1
21	19	540472	21361	LOVE LARGE	2	1/7/11	12.75	18179	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1
22	20	540473	22469	HEART OF W	4	1/7/11	1.65	17284	United Kingd	1	1/1/11	2011	1	2011	1	0	0	1

Our Cohort Index shows a range from 1 to 13.
Next we need to generate Cohort Pivot table to show retentions.

PIVOT TABLE

```
---Pivot Data to see the cohort table
SELECT  *
FROM (
  SELECT DISTINCT
    CustomerID,
    Cohort_Date,
    cohort_index
  FROM `churn-370413.online_retail.cohort_index`
)tbl
PIVOT(
  Count(CustomerID)
  FOR Cohort_Index IN (1,2,3,4,5,6,7,8,9,10,11,12,13)
) AS pivot_table;
```

SQL Output:

w	Cohort_Date	_1	_2	_3	_4	_5	_6	_7	_8	_9	_10	_11	_12	_13
1	2010-12-1	885	324	286	340	321	352	321	309	313	350	331	445	235
2	2011-1-1	417	92	111	96	134	120	103	101	125	136	152	49	0
3	2011-2-1	380	71	71	108	103	94	96	106	94	116	26	0	0
4	2011-3-1	452	68	114	90	101	76	121	104	126	39	0	0	0
5	2011-4-1	300	64	61	63	59	68	65	78	22	0	0	0	0
6	2011-5-1	284	54	49	49	59	66	75	27	0	0	0	0	0
7	2011-6-1	242	42	38	64	56	81	23	0	0	0	0	0	0
8	2011-7-1	188	34	39	42	51	21	0	0	0	0	0	0	0
9	2011-8-1	169	35	42	41	21	0	0	0	0	0	0	0	0
10	2011-9-1	299	70	90	34	0	0	0	0	0	0	0	0	0
11	2011-10-1	358	86	41	0	0	0	0	0	0	0	0	0	0
12	2011-11-1	323	36	0	0	0	0	0	0	0	0	0	0	0

Tableau Output:

Retention Table

	Cohort Index												
Cohort Period	1	2	3	4	5	6	7	8	9	10	11	12	13
2010-12-01	885	324	286	340	321	352	321	309	313	350	331	445	235
2011-01-01	417	92	111	96	134	120	103	101	125	136	152	49	
2011-02-01	380	71	71	108	103	94	96	106	94	116	26		
2011-03-01	452	68	114	90	101	76	121	104	126	39			
2011-04-01	300	64	61	63	59	68	65	78	22				
2011-05-01	284	54	49	49	59	66	75	27					
2011-06-01	242	42	38	64	56	81	23						
2011-07-01	188	34	39	42	51	21							
2011-08-01	169	35	42	41	21								
2011-09-01	299	70	90	34									
2011-10-01	358	86	41										
2011-11-01	323	36											
2011-12-01	40												

PIVOT RATE

-- We will convert the output of the above to Percentages:

-- (each ROW Value divided by First Row Value multiplied by 100)

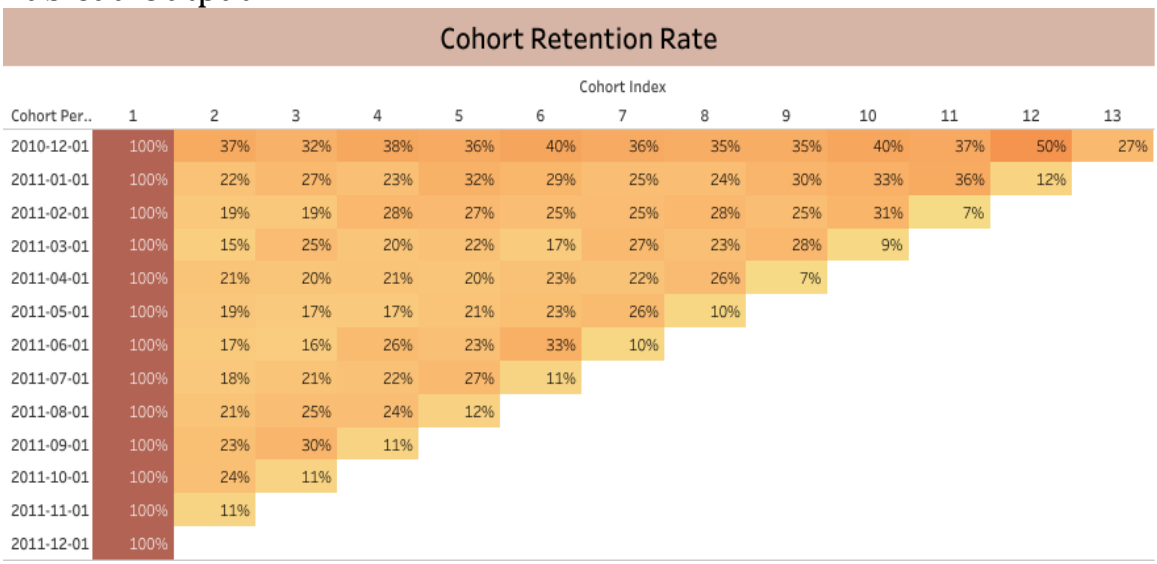
```
SELECT Cohort_Date,
   ROUND((_1/_1 * 100),2) as i1,
   ROUND((_2/_1 * 100),2) as i2,
   ROUND((_3/_1 * 100),2) as i3,
   ROUND((_4/_1 * 100),2) as i4,
   ROUND((_5/_1 * 100),2) as i5,
   ROUND((_6/_1 * 100),2) as i6,
   ROUND((_7/_1 * 100),2) as i7,
   ROUND((_8/_1 * 100),2) as i8,
```

```
    ROUND((_9/_1 * 100),2) as i9,
    ROUND((_10/_1 * 100),2) as i10,
    ROUND((_11/_1 * 100),2) as i11,
    ROUND((_12/_1 * 100),2) as i12,
    ROUND((_13/_1 * 100),2) as i13
FROM `churn-370413.online_retail.pivot`
```

SQL Output:

Row	Cohort_Date	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13
1	2010-12-1	100.0	36.61	32.32	38.42	36.27	39.77	36.27	34.92	35.37	39.55	37.4	50.28	26.55
2	2011-1-1	100.0	22.06	26.62	23.02	32.13	28.78	24.7	24.22	29.98	32.61	36.45	11.75	0.0
3	2011-2-1	100.0	18.68	18.68	28.42	27.11	24.74	25.26	27.89	24.74	30.53	6.84	0.0	0.0
4	2011-3-1	100.0	15.04	25.22	19.91	22.35	16.81	26.77	23.01	27.88	8.63	0.0	0.0	0.0
5	2011-4-1	100.0	21.33	20.33	21.0	19.67	22.67	21.67	26.0	7.33	0.0	0.0	0.0	0.0
6	2011-5-1	100.0	19.01	17.25	17.25	20.77	23.24	26.41	9.51	0.0	0.0	0.0	0.0	0.0
7	2011-6-1	100.0	17.36	15.7	26.45	23.14	33.47	9.5	0.0	0.0	0.0	0.0	0.0	0.0
8	2011-7-1	100.0	18.09	20.74	22.34	27.13	11.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2011-8-1	100.0	20.71	24.85	24.26	12.43	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	2011-9-1	100.0	23.41	30.1	11.37	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	2011-10-1	100.0	24.02	11.45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	2011-11-1	100.0	11.15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tableau Output:



Generating Churn Values

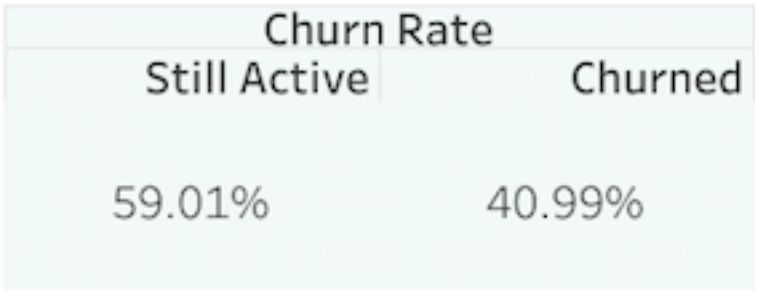
Finally, we need to generate values that will enable us calculate the churn rate.

```
  SELECT
   CustomerID,
   min(InvoiceDate) first_purchase_date,
   max(InvoiceDate) last_purchase_date,
   EXTRACT(MONTH FROM (max(InvoiceDate))) invoice_month,
   DATETIME_DIFF(DATETIME(TIMESTAMP(max(InvoiceDate))),
DATETIME(TIMESTAMP(min(InvoiceDate))), MONTH) as
months_active
 FROM clean_record
 GROUP BY CustomerID
 ORDER BY max(InvoiceDate) DESC
```

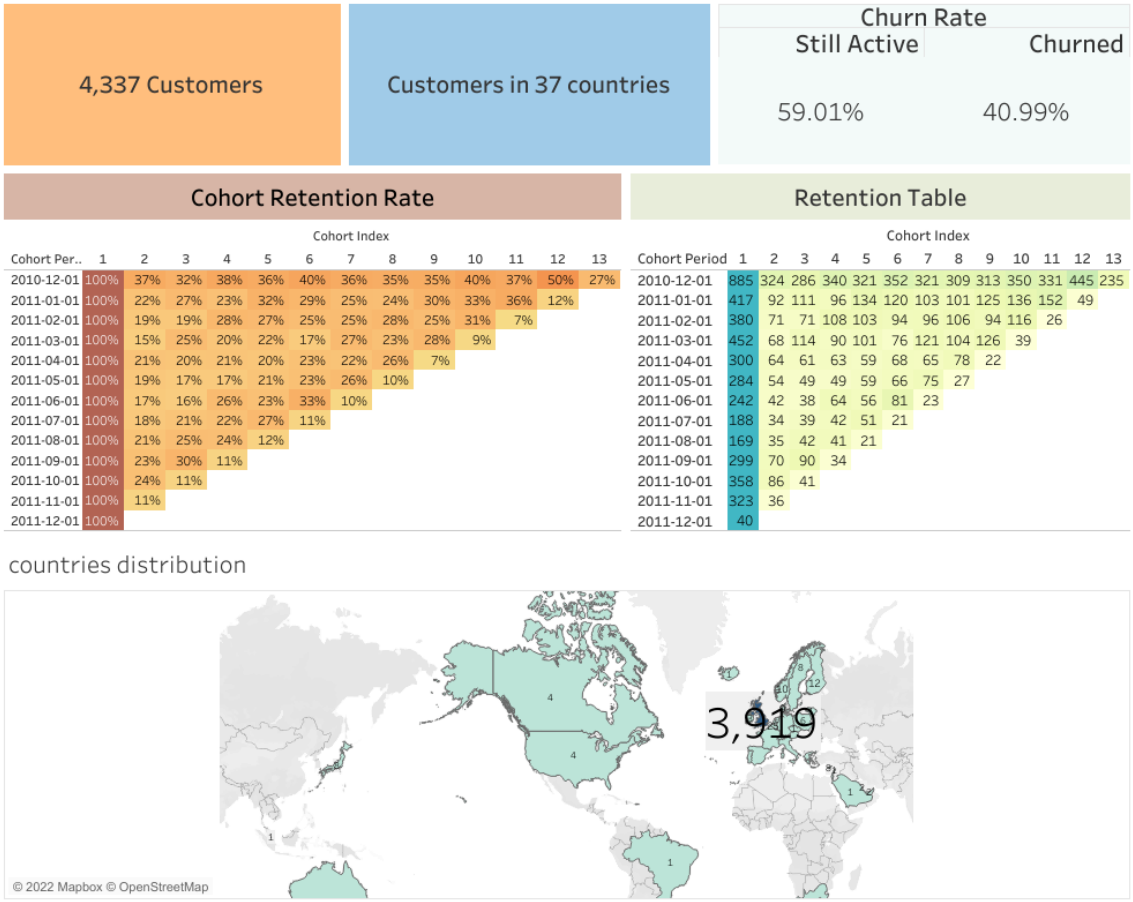
SQL output:

Row	CustomerID	first_purchase_d	last_purchase_d	invoice_month	months_active
351	14114	2011-01-13	2011-03-11	3	2
352	15057	2011-01-28	2011-03-09	3	2
353	14779	2011-01-20	2011-03-04	3	2
354	14479	2010-12-09	2011-03-31	3	3
355	15894	2010-12-05	2011-03-31	3	3
356	15353	2010-12-07	2011-03-30	3	3
357	15032	2010-12-08	2011-03-28	3	3
358	13649	2010-12-08	2011-03-28	3	3
359	18071	2010-12-09	2011-03-27	3	3
360	14256	2010-12-10	2011-03-24	3	3
361	16250	2010-12-01	2011-03-23	3	3

Tableau Output



Data Visualization



Conclusion

As we can see, the churn rate is 40.99%. It is high, but the most important thing is to see how to gain them back and reduce these rate. This can be done by first finding out why patronages stop.