**-- Creating the table**

create external table onlineretail\_raw(

InvoiceNo bigint,

StockCode string,

Description string,

Quantity int,

InvoiceDate string,

UnitPrice double,

CustomerID int,

Country string

)

row format delimited fields terminated by '\t'

TBLPROPERTIES ("skip.header.line.count"="1");

**-- Loading the data into table from the text file.**

load data inpath 'OnlineRetail.txt' into table onlineretail\_raw;

**-- Creating the final table with approriate datatypes.**

CREATE external Table RetailSales(

InvoiceNo bigint,

StockCode string,

Description string,

Quantity int,

InvoiceDate timestamp,

UnitPrice double,

TotalPrice double,

CustomerID int,

Country string

);

**-- Inserting into the Final table from the raw table**

INSERT overwrite table RetailSales

select

InvoiceNo,

StockCode,

Description,

Quantity,

CONCAT(InvoiceDate,':00'),

UnitPrice,

ROUND(UnitPrice \* Quantity \* 100)/100 AS TotalPrice,

CustomerID,

Country

from onlineretail\_raw;

**--ANALYSIS**

**-- 1. Revenue Aggregate By Country for top 5 countries.**

select Country, round(sum(UnitPrice\*Quantity)) as Total\_Price from RetailSales

where round(UnitPrice\*Quantity) > 0

group by Country

order by Total\_Price desc

limit 5;

**-- 2. Sales Metrics like NumCustomers, NumTransactions, AvgNumItems, MinAmtPerCustomer, MaxAmtPerCustomer, AvgAmtPerCustomer, StdDevAmtPerCustomer etc by country for top 5 Countries.**

**-- Creating A View with the top 5 countries from the previous question.**

CREATE VIEW InvoiceAmount AS

Select

country,

customerid,

invoiceno,

count(distinct Stockcode) as NumItems,

sum(totalprice) as InvoiceTotal

from retailsales

where totalprice > 0 and Country in ('United Kingdom','Netherlands','EIRE','Germany','France')

group by country,invoiceno,customerid;

**-- Showing the requested metrics from the view**

select

Country,

count(distinct customerid) as NumCustomers,

count(distinct invoiceno) as NumTransactions,

round(avg(NumItems)) as AvgNumItems,

round(min(InvoiceTotal)) as MinAmtperCustomer,

round(max(InvoiceTotal)) as MaxAmtperCustomer,

round(avg(InvoiceTotal)) as AvgAmtperCustomer,

round(std(InvoiceTotal)) as StdDevAmtperCustomer

from InvoiceAmount

group by Country

order by NumCustomers desc;

**-- 3. Daily Sales Activity like NumVisits, TotalAmt etc... per POSIX day of the year**

create view DailySales as

select

date\_format(Invoicedate,'D') as DayOfYear,

Invoiceno,

totalprice

from retailsales

where totalprice > 0 and Invoicedate > '2010-11-31' and Invoicedate < '2011-12-01';

select

DayOfYear,

count(distinct Invoiceno) as NumVisits,

round(sum(totalprice)) as TotalAmt

from DailySales

group by DayOfYear

order by DayOfYear desc;

**-- 4. Hourly Sales Activity like NumVisits, TotalAmt etc... per hour of day**

select

hour(invoicedate) as HourOfDay,

count(distinct Invoiceno) as NumVisits,

round(sum(totalprice)) as TotalAmt

from retailsales

group by hour(invoicedate)

order by HourOfDay desc;

**-- 5. Basket size distribution (Note: Basket size = number of item in a transaction)(in this question, we would like to know that, number of transactions be each basket size i.e. number of transaction with 3 size, number of transaction with 4 size etc)**

create view ItemsDistinct as

select

invoiceno,

count(distinct Stockcode) as NumOfItems

from retailsales

group by invoiceno;

select

NumOfItems,

count(NumOfItems) as CountNumItems

from ItemsDistinct

group by NumOfItems

order by NumOfItems asc;

**-- 6. Top 20 items sold by Frequency**

select

Description,

count(Stockcode) as ItemFrequency

from retailsales

where totalprice > 0

group by Description

order by ItemFrequency desc

limit 20;

-- 7. Customer Lifetime Value distribution by intervals of 1000's (Customer Life Value = total spend

by customer is his/her tenure with the company) (In this question, we would like to calculate

how many customers with CLV between 1-1000, 1000-2000 etc.)? Please note that we don't want calculate bins manually and it required to create bins dynamically.

select customerid,

sum(quantity\*unitprice) Totalspend,

concat(cast((floor((sum(quantity\*unitprice))/1000)\*1000 ) as string),

        '-',

        cast((ceil((sum(quantity\*unitprice))/1000)\*1000) as string) ) bins

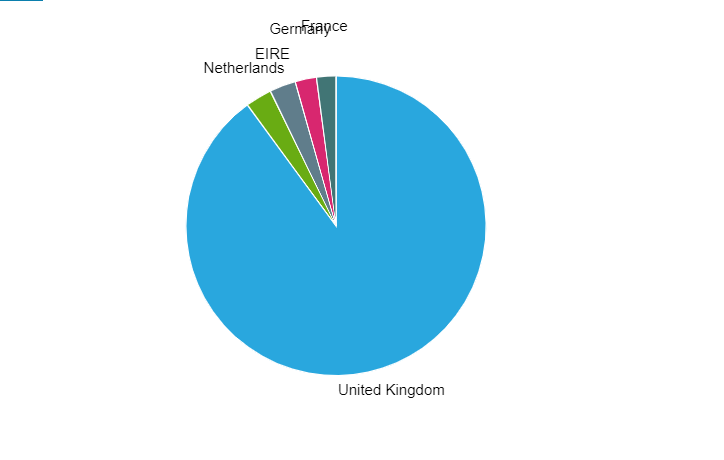
from retailsales

where customerid is not null and quantity>0

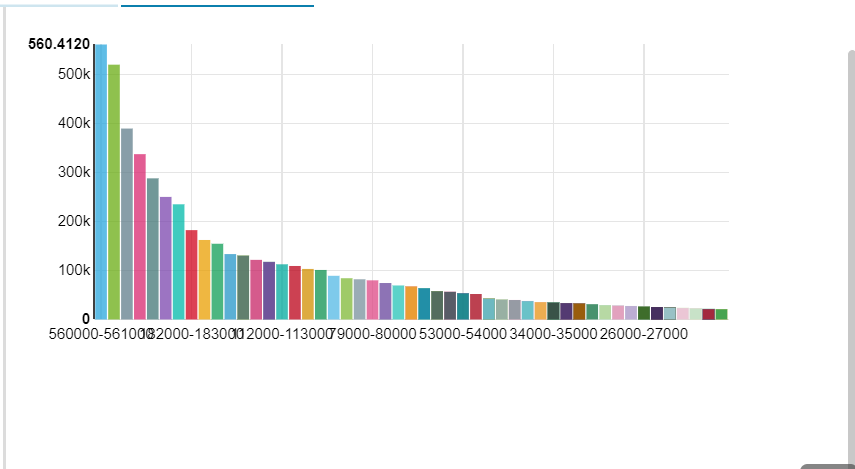
group by customerid

order by Totalspend DESC;

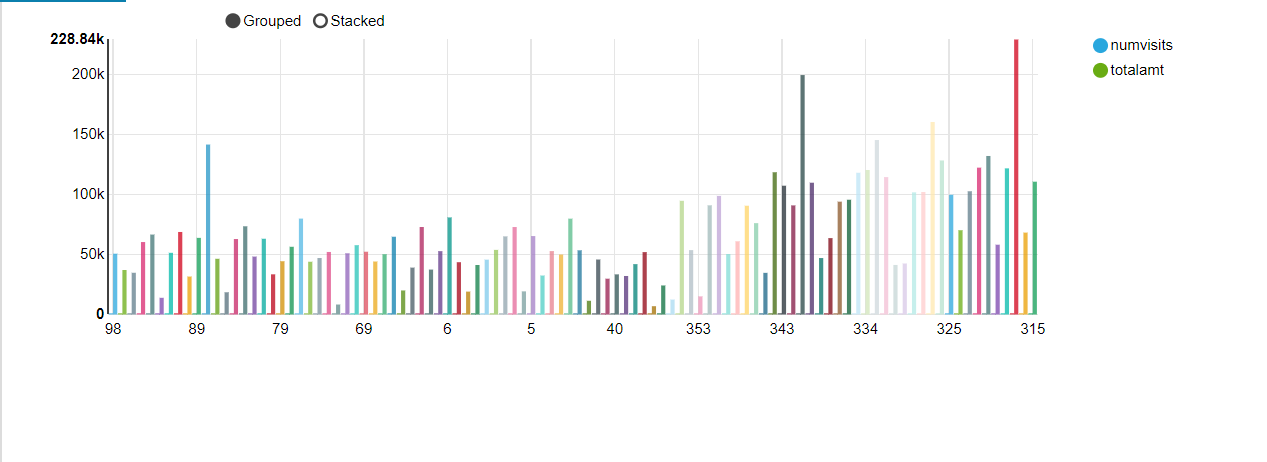
**Revenue by Country**



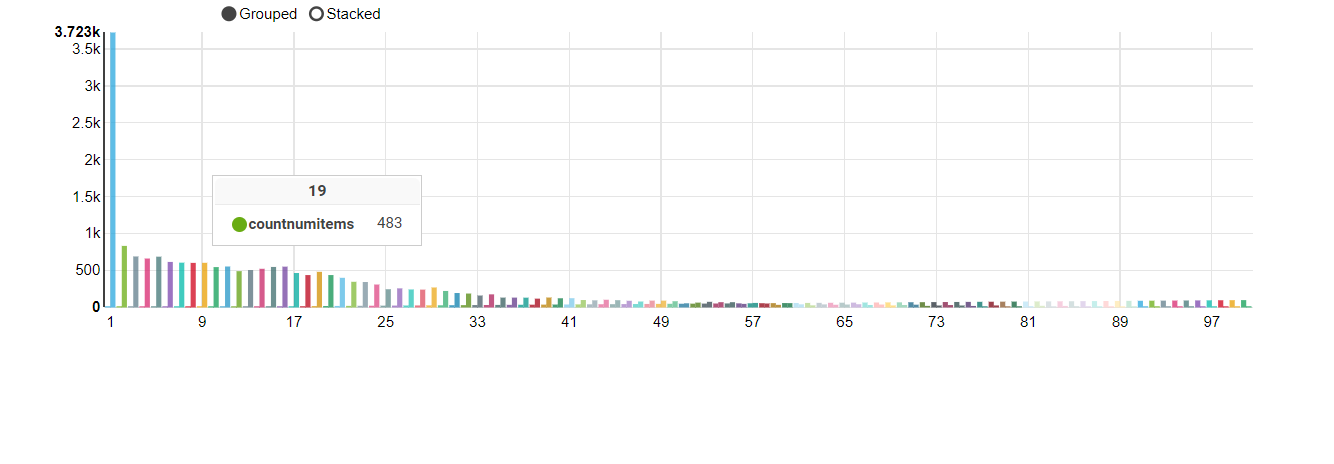
**Distribution of Customer life time value intervals**



**Daily Sales Activity**



**Basket Size Distribution**



**Top 20 Items sold by Frequency**

