**SQL Queries**

**price\_elasticity**

**Code:**

with overall\_product as

(

select \* from (

select distinct extract(year from date) as year,PARSE\_DATE("%Y%m",format\_date("%Y%m",date)) as month\_year,item\_description,

round(sum(sale\_dollars)/sum(bottles\_sold),2) as retail\_price\_new,

sum(bottles\_sold) as total\_sold,

count(distinct invoice\_and\_item\_number) as total\_invoices,

round(sum(sale\_dollars),2) as total\_sales\_dollar

from `bigquery-public-data.iowa\_liquor\_sales.sales`

group by 1,2,3

order by 3,1 asc,2 asc

) where year>=2022 and total\_sold is not null and total\_sold!=0

)

Select \*,

case when elasticity\_quotient is null then null

when elasticity\_quotient>1 then "elastic"

when elasticity\_quotient=1 then "unitary"

when elasticity\_quotient=0 then "perfectly inelastic"

when elasticity\_quotient<1 then "inelastic"

end as elastic\_group

from (

Select \*, case when retail\_percent\_diff>=0.10 then 1 else 0 end as greater\_than\_10\_percent,

case when retail\_percent\_diff>=0.10 and sales\_percent\_diff>=0 then 1

when retail\_percent\_diff>=0.10 and sales\_percent\_diff<=0 then 2 else 0 end as difference\_groupings,

case when retail\_percent\_diff = 0 then null

else round((quantity\_percent\_diff/retail\_percent\_diff),2) end as elasticity\_quotient

from(

select \*,round(retail\_price\_new - lag\_retail\_price\_new,2) as retail\_dollar\_diff,

round((retail\_price\_new - lag\_retail\_price\_new)/ lag\_retail\_price\_new,2) as retail\_percent\_diff,

round(total\_sales\_dollar - lag\_total\_sales\_dollar,2) as sales\_dollar\_diff,

round((total\_sales\_dollar - lag\_total\_sales\_dollar)/ lag\_total\_sales\_dollar,2) as sales\_percent\_diff,

round((total\_sold - lag\_total\_sold)/ lag\_total\_sold,2) as quantity\_percent\_diff

from

( select \*,lag(retail\_price\_new) over (partition by item\_description order by year asc,month\_year asc) as lag\_retail\_price\_new,

lag(total\_sales\_dollar) over (partition by item\_description order by year asc,month\_year asc) as lag\_total\_sales\_dollar,

lag(total\_sold) over (partition by item\_description order by year asc,month\_year asc) as lag\_total\_sold

from overall\_product

)

)

)

**quantity\_forecast**

**ARIMA Model:**

CREATE OR REPLACE MODEL `absolute-keel-377719.google\_hackathon.arima\_liquor\_item`

OPTIONS(MODEL\_TYPE = 'ARIMA\_PLUS'

, TIME\_SERIES\_TIMESTAMP\_COL = 'date'

, TIME\_SERIES\_DATA\_COL = 'bottles\_sold'

, TIME\_SERIES\_ID\_COL = 'item\_description'

, HORIZON = 12

, AUTO\_ARIMA = TRUE

, AUTO\_ARIMA\_MAX\_ORDER = 5

, DATA\_FREQUENCY = 'MONTHLY'

, HOLIDAY\_REGION = 'US'

, DECOMPOSE\_TIME\_SERIES = TRUE

, CLEAN\_SPIKES\_AND\_DIPS = TRUE

, ADJUST\_STEP\_CHANGES = TRUE

, TREND\_SMOOTHING\_WINDOW\_SIZE = 11)

AS

SELECT

date, bottles\_sold, item\_description

FROM(

select year, month, DATETIME(year, month, 1,0,0,0) as date, sum(sale\_dollars) as total\_sales, sum(bottles\_sold) as bottles\_sold, item\_description

from

(SELECT extract(year FROM date) as year,extract(month FROM date) as month, \* from `bigquery-public-data.iowa\_liquor\_sales.sales`)

group by year, month, item\_description

order by year, month

)

**NOTE:** Trend smoothing window determined by STL decomposition in R Studio.

**Code:**

SELECT

\*

FROM ML.EXPLAIN\_FORECAST(MODEL `absolute-keel-377719.google\_hackathon.arima\_liquor\_item`,

STRUCT(12 AS horizon, 0.9 AS confidence\_level))

**sales\_dashboard**

**Code:**

select \* from

(

select sales.\*,loc.store\_location from

(SELECT date, state\_bottle\_retail, sum(sale\_dollars) as total\_sales, sum(bottles\_sold) as total\_quantity, avg(pack\*state\_bottle\_cost) as cpc, item\_description, store\_number, store\_name, vendor\_number, vendor\_name, city, county

FROM bigquery-public-data.iowa\_liquor\_sales.sales

GROUP BY date, item\_description, store\_number, store\_name, vendor\_number, vendor\_name, city, county, state\_bottle\_retail) as sales

left join (select \* from (select \*,row\_number() over(partition by store\_number) as row\_num from (select store\_number, store\_location from `bigquery-public-data.iowa\_liquor\_sales.sales` ))where row\_num = 1) as loc

on loc.store\_number = sales.store\_number

)

**sales\_forecast**

**ARIMA Model:**

CREATE OR REPLACE MODEL `absolute-keel-377719.google\_hackathon.arima\_liquor`

OPTIONS(MODEL\_TYPE = 'ARIMA\_PLUS'

, TIME\_SERIES\_TIMESTAMP\_COL = 'date'

, TIME\_SERIES\_DATA\_COL = 'total\_sales'

, TIME\_SERIES\_ID\_COL = 'store\_name'

, HORIZON = 12

, AUTO\_ARIMA = TRUE

, AUTO\_ARIMA\_MAX\_ORDER = 5

, DATA\_FREQUENCY = 'MONTHLY'

, HOLIDAY\_REGION = 'US'

, DECOMPOSE\_TIME\_SERIES = TRUE

, CLEAN\_SPIKES\_AND\_DIPS = TRUE

, ADJUST\_STEP\_CHANGES = TRUE

, TREND\_SMOOTHING\_WINDOW\_SIZE = 11)

AS

SELECT

date, total\_sales, store\_name

FROM(

select year, month, DATETIME(year, month, 1,0,0,0) as date, sum(sale\_dollars) as total\_sales, sum(bottles\_sold) as bottles\_sold, store\_name

from

(SELECT extract(year FROM date) as year,extract(month FROM date) as month, \* from `bigquery-public-data.iowa\_liquor\_sales.sales`)

group by year, month, store\_name

order by year, month

)

**NOTE:** Trend smoothing window determined by STL decomposition in R Studio.

**Code:**

SELECT

\*

FROM ML.EXPLAIN\_FORECAST(MODEL `absolute-keel-377719.google\_hackathon.arima\_liquor`,

STRUCT(12 AS horizon, 0.9 AS confidence\_level))

**sales\_dashboard\_forecast**

**Code:**

SELECT date, store\_name, county, city, total\_sales, 'Historical' as `Type`

FROM absolute-keel-377719.google\_hackathon.sales\_dashboard m1

UNION ALL

SELECT CAST(time\_series\_timestamp AS DATE) as date, ml.store\_name, county, city, time\_series\_data, 'Forecast' as `Type`

FROM ML.EXPLAIN\_FORECAST(MODEL `absolute-keel-377719.google\_hackathon.arima\_liquor`,

STRUCT(12 AS horizon, 0.9 AS confidence\_level)) ml

LEFT JOIN (select distinct store\_name,max(city) as city,max(county) as county from`bigquery-public-data.iowa\_liquor\_sales.sales`

group by store\_name

order by 1) ls ON ls.store\_name = ml.store\_name

WHERE CAST(time\_series\_timestamp AS DATE) >= DATE(2023,01,01)