BIG DATA MANAGEMENT

BLACK FRIDAY ANALİZİ

Grup Üyeleri

- Büşra Nur Aydemir
- Leyla Yiğit
- Mehmet Ak
- Mercan Karacabey
- Merve Özen Şahin

Projede kullanılacak datayı Kaggle'dan indirdik. (https://www.kaggle.com/mehdidag/black-friday/version/1)

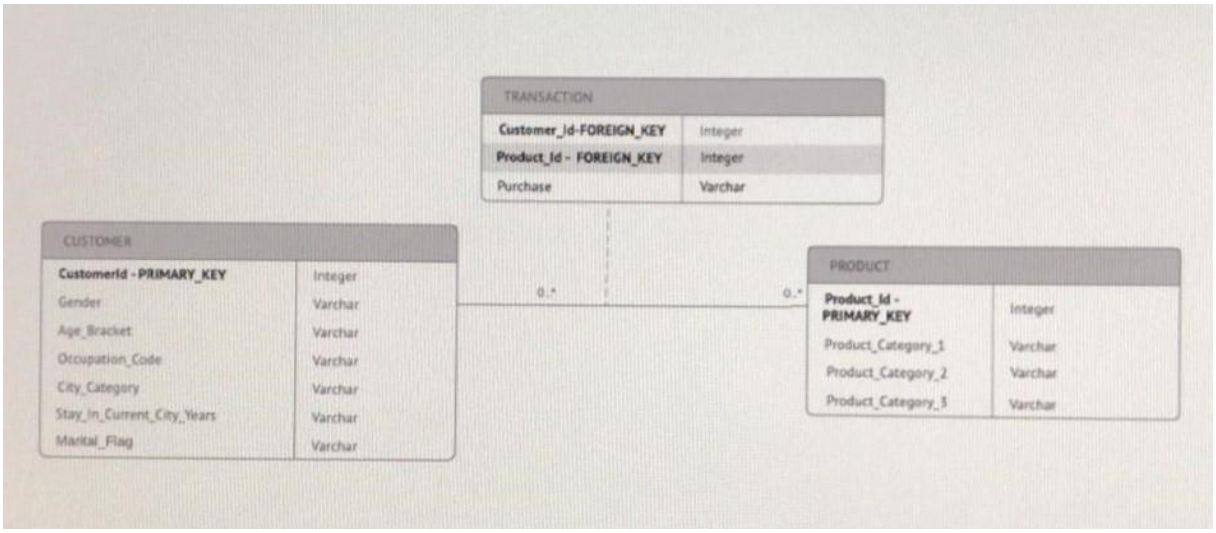
İndirilen datayı Veritabanında Black_Friday isimli bir tablo yarattık ve csv datayı PostgreSQL'e yükledik. Datanın toplam satır sayısı: 1075153.

Datalar local db üzerinde tutulmuştur.

PostgreSQL'de 3 adet tablo oluşturduk. Bu şekilde normalize bir veritabanı tasarımının olması bizim sorgu performanslarımızı da iyileştirecek, gereksiz veri tekrarını azaltacaktı. Transaction, Customer ve Product. Bu tablolara datayı Black_Friday tablosundan aşağıdaki komutlar ile aktardık. Bu aktarım sürecinden sonra ilk yüklediğimiz tabloyu da drop ettik.

```
INSERT INTO PRODUCT
SELECT DISTINCT PRODUCT_ID, PRODUCT_CATEGORY_1, PRODUCT_CATEGORY_2,
PRODUCT_CATEGORY_3 FROM BLACK_FRIDAY;

INSERT INTO TRANSACTION
SELECT USER_ID, PRODUCT_ID , PURCHASE FROM BLACK_FRIDAY;

INSERT INTO CUSTOMER
SELECT DISTINCT USER_ID, GENDER, AGE, OCCUPATION, CITY_CATEGORY,
Stay_In_Current_City_Years, TO_NUMBER(MARITAL_STATUS, '9')  FROM TABLO;
```

ER Diagram



With ile yazdığımız ilk cte(Common table expression) yapısında müşteri ve ürün kategori_1 bazında; toplam alışveriş tutarını hesaplıyor. Sonrasında müşteri tablosu ile join kuruyoruz. Buradan müşteriye ait cinsiyet, yaş aralığı, meslek kodu, şehir kodu gibi bilgileri de alarak (ki bu bilgiler müşteri bazında unique olduğu için datayı çoklatmıyor) kaç ürün kategorisinde alışveriş yaptığını hesaplayarak, toplam tutara göre büyükten küçüğe sıralıyor. With koymadan bu işlemin aynısını 3 tabloyu joinleyerek de elde edebilirdik. Count(tp.Product_Category_1) kısmındaki kullanımı Count(distinct tp.Product_Category_1) diyerek aynı sonuca ulaşırdık.

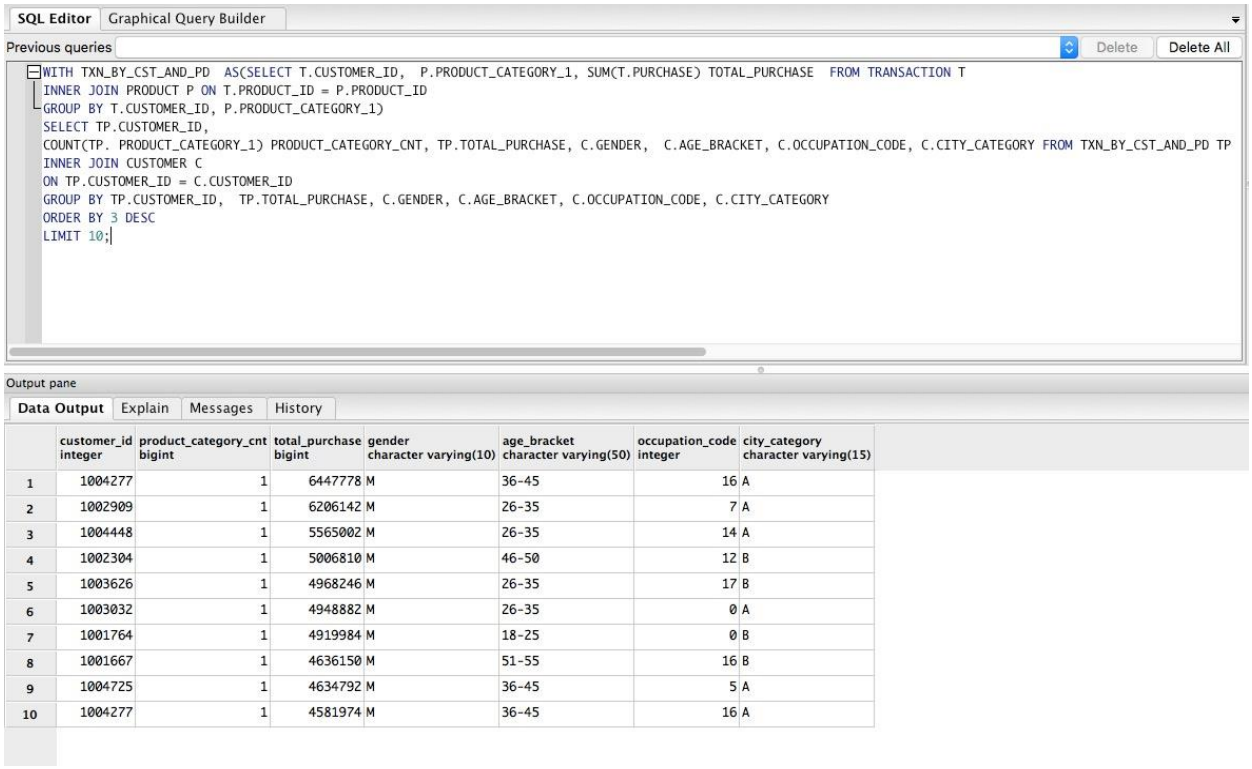
```
WITH TXN_BY_CST_AND_PD  AS(SELECT T.CUSTOMER_ID,  P.PRODUCT_CATEGORY_1,
SUM(T.PURCHASE) TOTAL_PURCHASE  FROM TRANSACTION T INNER JOIN PRODUCT P
ON T.PRODUCT_ID = P.PRODUCT_ID
GROUP BY T.CUSTOMER_ID, P.PRODUCT_CATEGORY_1)
SELECT TP.CUSTOMER_ID,
COUNT(TP. PRODUCT_CATEGORY_1) PRODUCT_CATEGORY_CNT, TP.TOTAL_PURCHASE,
C.GENDER,  C.AGE_BRACKET, C.OCCUPATION_CODE, C.CITY_CATEGORY FROM
TXN_BY_CST_AND_PD TP INNER JOIN CUSTOMER C
ON TP.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY TP.CUSTOMER_ID,  TP.TOTAL_PURCHASE, C.GENDER, C.AGE_BRACKET,
C.OCCUPATION_CODE, C.CITY_CATEGORY ORDER BY 3 DESC;
```

Üstteki ile aynı sonuç:

```
SELECT T.CUSTOMER_ID,
COUNT(DISTINCT P.PRODUCT_CATEGORY_1) PRODUCT_CATEGORY_CNT,
SUM(T.PURCHASE) TOTAL_PURCHASE,
C.GENDER,  C.AGE_BRACKET,  C.OCCUPATION_CODE, C.CITY_CATEGORY FROM
TRANSACTION T INNER JOIN PRODUCT P
ON T.PRODUCT_ID = P.PRODUCT_ID
INNER JOIN CUSTOMER C
ON T.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY T.CUSTOMER_ID,  T.PURCHASE, C.GENDER, C.AGE_BRACKET,
C.OCCUPATION_CODE, C.CITY_CATEGORY ORDER BY 3 DESC;
```

Yukarıdaki sql'in çıktısı :

(Sql'in çıktısı csv file olarak da proje dosyasında yer almaktadır.)



SQL Editor | Graphical Query Builder

Previous queries | Delete | Delete All

```
WITH TXN_BY_CST_AND_PD  AS(SELECT T.CUSTOMER_ID,  P.PRODUCT_CATEGORY_1, SUM(T.PURCHASE) TOTAL_PURCHASE  FROM TRANSACTION T
INNER JOIN PRODUCT P ON T.PRODUCT_ID = P.PRODUCT_ID
GROUP BY T.CUSTOMER_ID, P.PRODUCT_CATEGORY_1)
SELECT TP.CUSTOMER_ID,
COUNT(TP. PRODUCT_CATEGORY_1) PRODUCT_CATEGORY_CNT, TP.TOTAL_PURCHASE, C.GENDER,  C.AGE_BRACKET, C.OCCUPATION_CODE, C.CITY_CATEGORY FROM TXN_BY_CST_AND_PD TP
INNER JOIN CUSTOMER C
ON TP.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY TP.CUSTOMER_ID,  TP.TOTAL_PURCHASE, C.GENDER, C.AGE_BRACKET, C.OCCUPATION_CODE, C.CITY_CATEGORY
ORDER BY 3 DESC
LIMIT 10;
```

Output pane

Data Output | Explain | Messages | History

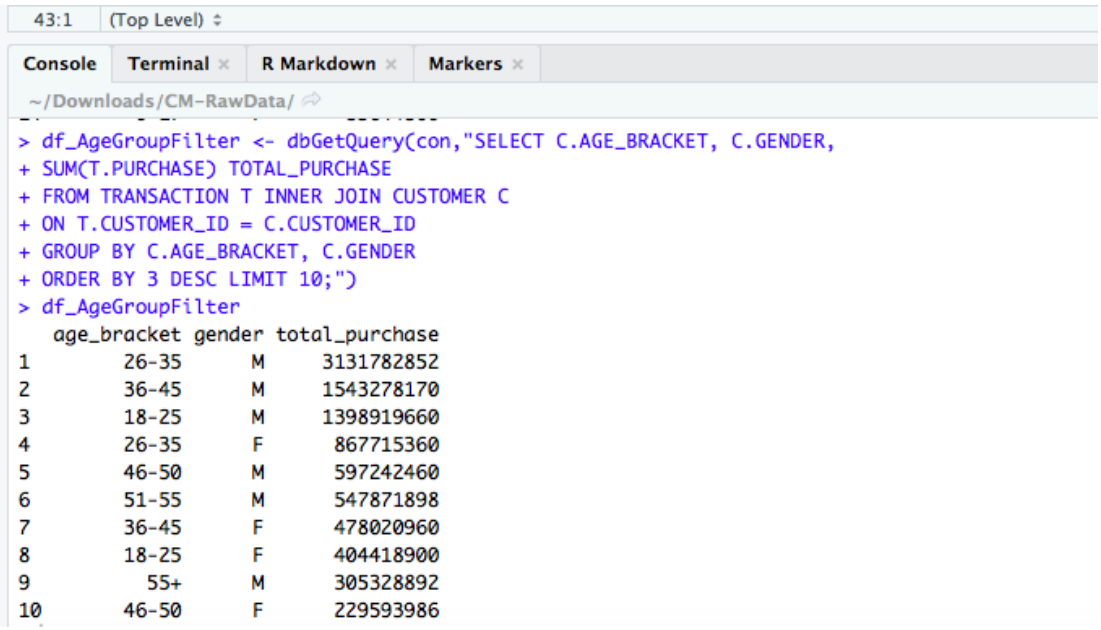
	customer_id integer	product_category_cnt bigint	total_purchase bigint	gender character varying(10)	age_bracket character varying(50)	occupation_code integer	city_category character varying(15)
1	1004277	1	6447778	M	36-45	16	A
2	1002909	1	6206142	M	26-35	7	A
3	1004448	1	5565002	M	26-35	14	A
4	1002304	1	5006810	M	46-50	12	B
5	1003626	1	4968246	M	26-35	17	B
6	1003032	1	4948882	M	26-35	0	A
7	1001764	1	4919984	M	18-25	0	B
8	1001667	1	4636150	M	51-55	16	B
9	1004725	1	4634792	M	36-45	5	A
10	1004277	1	4581974	M	36-45	16	A

R Üzerinden Görselleştirilme Yapılmıştır.

R kodları aşağıdaki gibidir :

```
install.packages("RPostgreSQL")

remove.packages("DBI")

install.packages("DBI")

require("RPostgreSQL")

# create a connection

# save the password that we can "hide" it as best as we can by collapsing it

pw <- {"mk1234"}

# loads the PostgreSQL driver

drv <- dbDriver("PostgreSQL")

# creates a connection to the postgres database

# note that "con" will be used later in each connection to the database
```

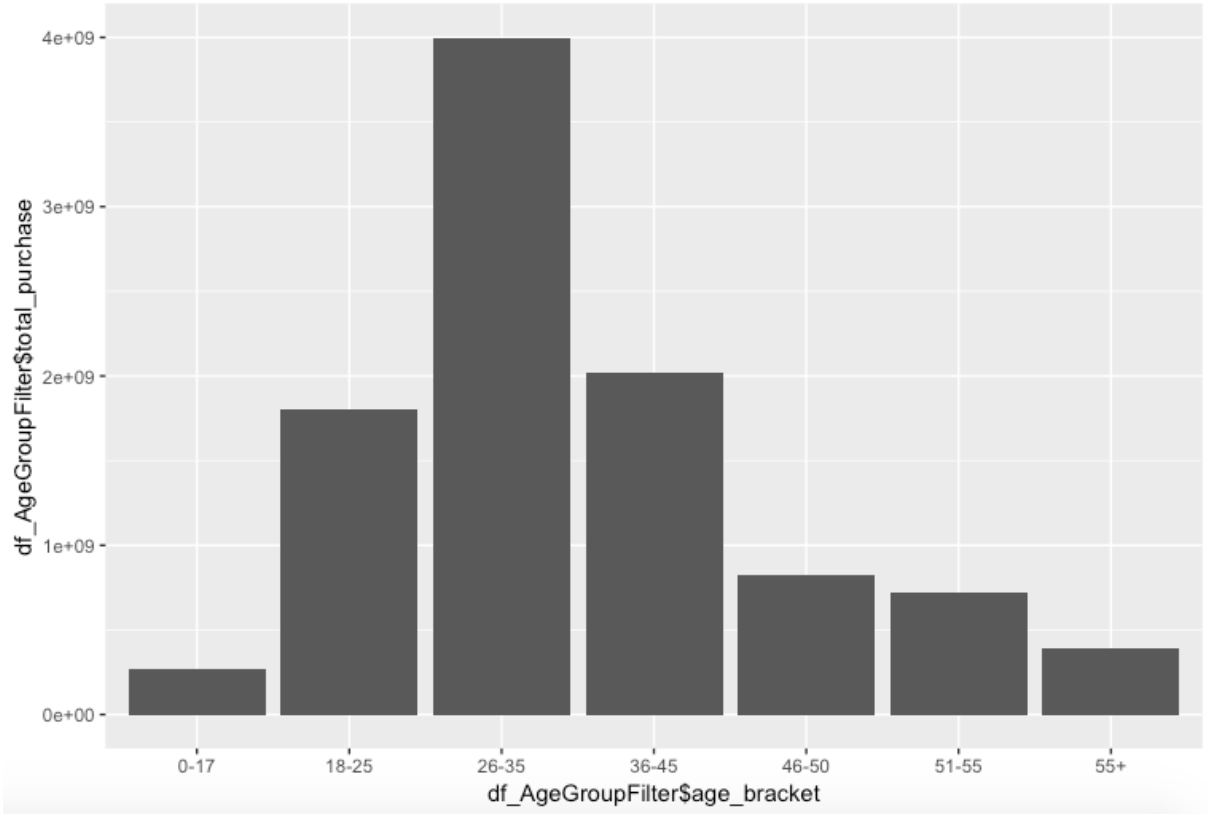
```
con <- dbConnect(drv, dbname = "postgres",
                 host = "localhost", port = 5432,
                 user = "postgres", password = pw)
rm(pw) # removes the password
# check for the black_friday
dbExistsTable(con, "black_friday")
# query the data from postgreSQL
df_customer <- dbGetQuery(con, "SELECT * from customer")
df_product <- dbGetQuery(con, "SELECT * from product")
df_transaction <- dbGetQuery(con,"SELECT * from product Transaction limit 10")
df_AgeGroupFilter <- dbGetQuery(con,"SELECT C.AGE_BRACKET, C.GENDER,
SUM(T.PURCHASE) TOTAL_PURCHASE
FROM TRANSACTION T INNER JOIN CUSTOMER C
ON T.CUSTOMER_ID = C.CUSTOMER_ID
GROUP BY C.AGE_BRACKET, C.GENDER
ORDER BY 3 DESC
LIMIT 10;")
df_AgeGroupFilter
```



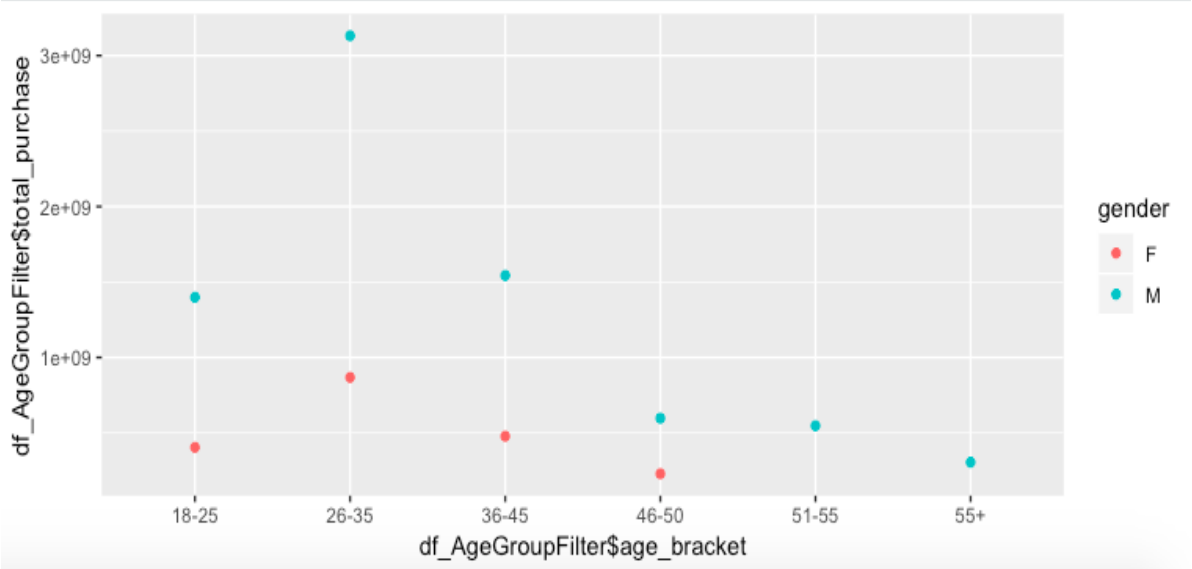
```
ggplot(df_AgeGroupFilter,aes(x=df_AgeGroupFilter$age_bracket,
       y=Home.Value,
       color=gender))+
 geom_point()
```

R'da Görselleştirmenin Çıktısı:

- ***Yaş grubuna ve toplam değerlere göre görselleştirme yapılmıştır.***

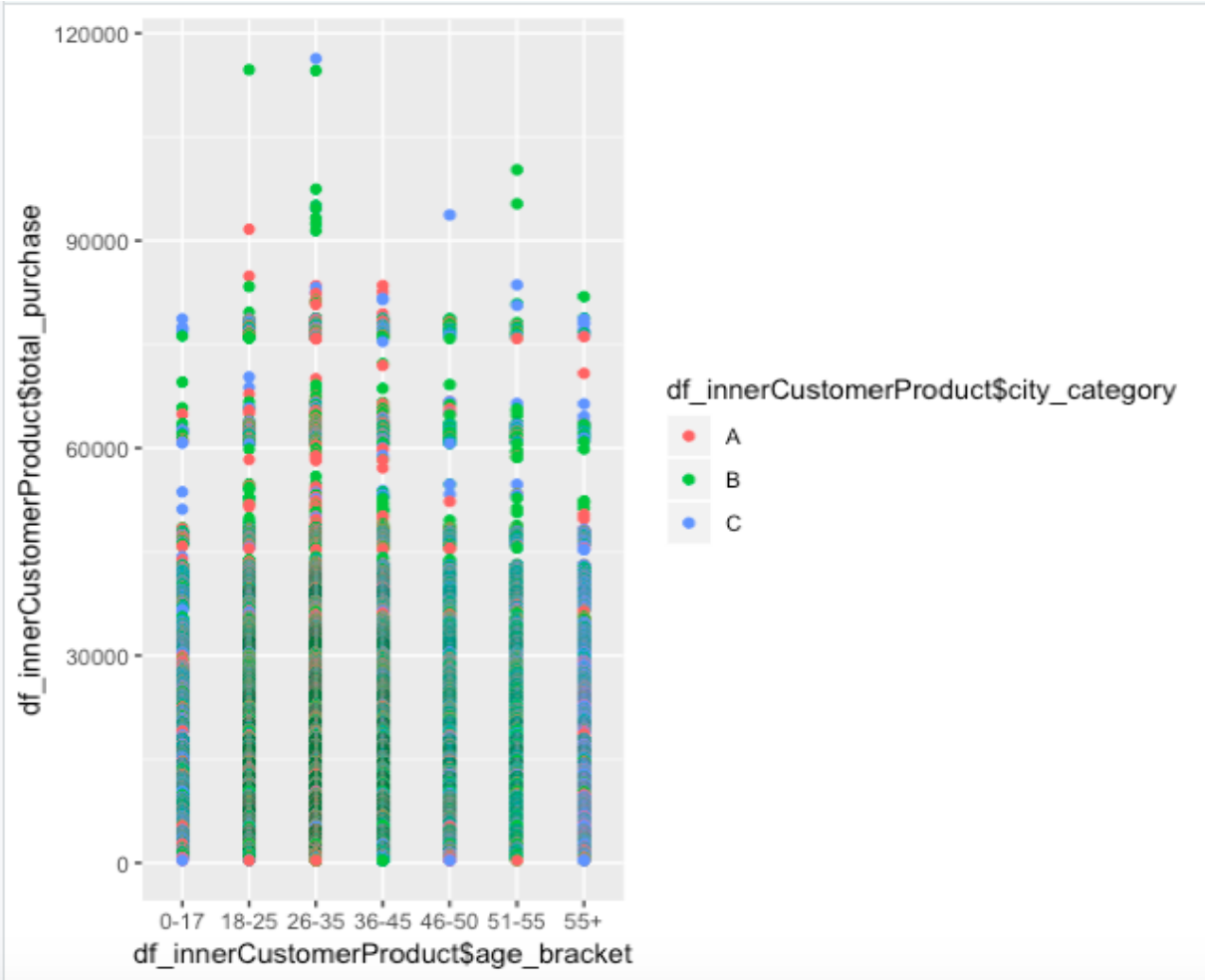


- ***Yaş grubuna ve cinsiyete göre görselleştirme yapılmıştır.***



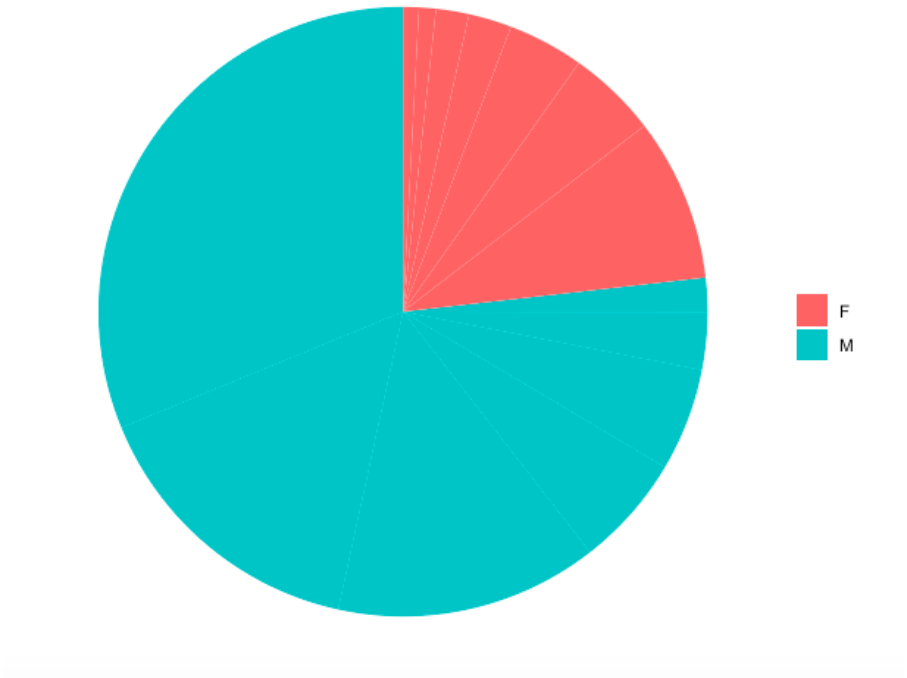
- ***City Category'sine göre renklendirilerek :***

```
pc1 <- ggplot(df_innerCustomerProduct, aes(x =
df_innerCustomerProduct$age_bracket, y =
df_innerCustomerProduct$total_purchase, color =
df_innerCustomerProduct$city_category))

pc1 + geom_point()


pc2 <- pc1 +

 geom_smooth(mapping = aes(linetype = "r2"),

          method = "lm",

          formula = y ~ x + log(x), se = FALSE,

          color = "red")

pc2 + geom_point()
```

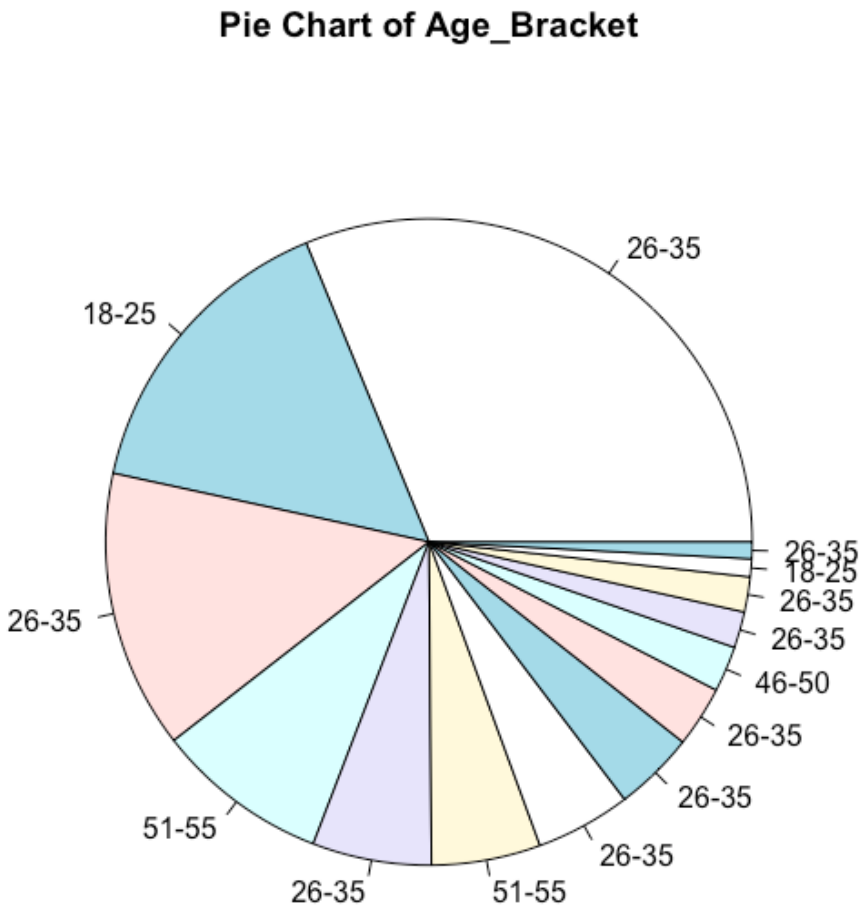


- *KADIN/ERKEK Alışveriş Oranları Piechart olarak ifade edilmiştir:*

```
# Gender - Total Purchase

ggplot(data=df_AgeGroupFilter,aes(x="",y =
df_AgeGroupFilter$total_purchase,fill=df_AgeGroupFilter$gender)) +

 geom_bar(stat="identity",width=1) +

 coord_polar("y",direction=-1) +

 theme_void() +

 labs(fill="")
```



Yaş Kategorilerine Göre Pie Chart Eklenmiştir.



#Age Bracket - Total Purchase
pie(df_AgeGroupFilter$total_purchase, labels =
df_innerCustomerProduct$age_bracket, main="Pie Chart of Age_Bracket")