The solutions have been derived using RStudio. Here I provide explanations to what I have done, the code I have used, plots, and analyzes of the plots. The codes are marked here with >.

First the excel dataset should be saved as three separate csv tables so that they can be imported to RStudio following the steps File > Import Dataset > From text (base) > browse > choose the right file. When the files have been imported succesfully they are renamed.

```
>customers <- Aktia_challenge_data_customers
>events <- Aktia_challenge_data_events
>products <- Aktia_challenge_data_products
```

Install the needed packages and load them so that we can use SQL query, plot etc.

```
>install.packages("ggplot2")
>install.packages("dplyr")
>install.packages("lubridate")
>install.packages("sqldf")
>install.packages("stats")
```

```
>library(ggplot2)
>library(dplyr)
>library(lubridate)
>library(sqldf)
>library(stats)
```

Change the format of the column Date in events from character to date.

```
>date <- dmy(events$Date)
>events$Date <- date
```

Change the names of the sixth and seventh columns so they are easier to handle. Print to see the changes.

```
>names(events)[6] <- "Discount_percent"
>names(events)[7] <- "Payment_eur"
>events
```

Modify the header of the products table to make it easier to handle.

```
>names(products) <- as.matrix(products[1, ])
>products <- products[-1, ]
>products[] <- lapply(products, function(x) type.convert(as.character(x)))
```

Change the names of the products table so the names of the columns with prices are more simple. Denote the fourth column with list prices from 1.1.2017-31.12.2018 as old_prices, and the fifth column with list prices from 1.1.2019-31.12.2020 as new_prices.

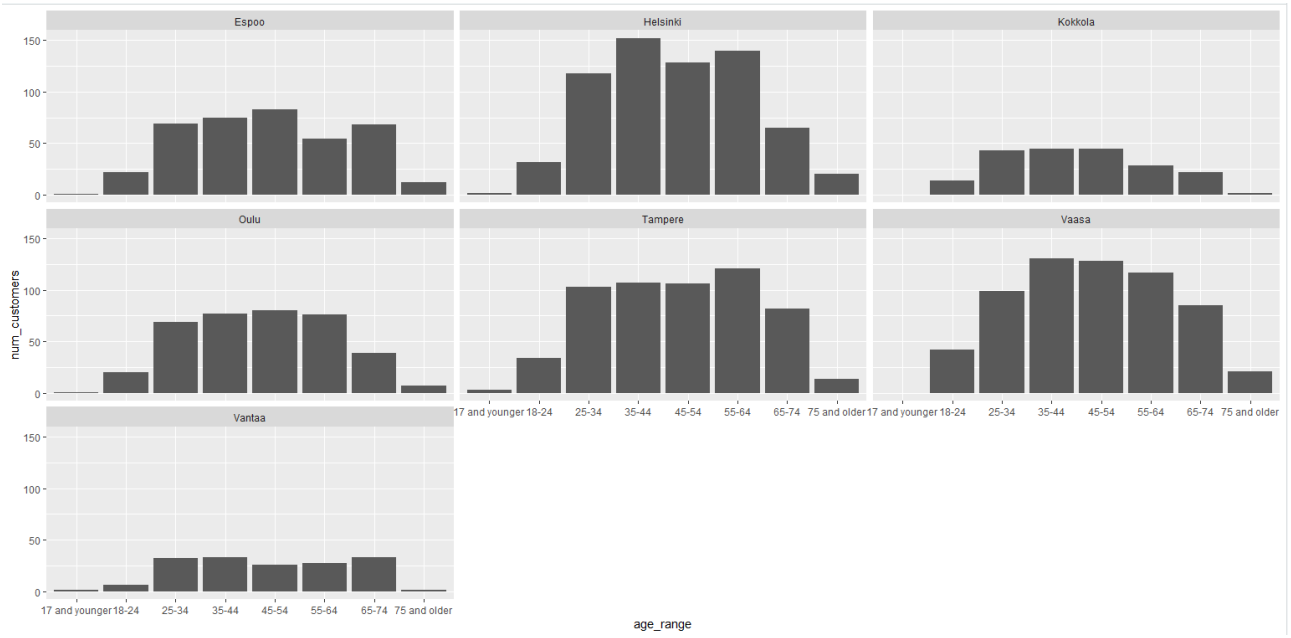
```
>names(products)[4] <- "old_prices"
>names(products)[5] <- "new_prices"
>products
```

Who are our customers? Let's find out how old our customers are and what branch they are customers in.

```
>age_customers <- sqldf("SELECT COUNT(CustomerID) AS num_customers,
                        Branch,
                        CASE WHEN Age < 18 THEN '17 and younger'
                         WHEN AGE BETWEEN 18 AND 24 THEN '18-24'
                         WHEN AGE BETWEEN 25 AND 34 THEN '25-34'
                         WHEN AGE BETWEEN 35 AND 44 THEN '35-44'
                         WHEN AGE BETWEEN 45 AND 54 THEN '45-54'
                         WHEN AGE BETWEEN 55 AND 64 THEN '55-64'
                         WHEN AGE BETWEEN 65 AND 74 THEN '65-74'
                         WHEN AGE >= 75 THEN '75 and older'
                         END AS age_range
                         FROM customers
                        GROUP BY age_range,
                        Branch
                        ORDER BY age_range")
```

```
>age_customers
```

```
>ggplot(age_customers, aes(x = age_range, y= num_customers, Fill = age_range))+
 geom_col() +
 facet_wrap(~Branch)
```



Aktia has mainly customers who are in their 30's to 60's. The age distribution might be of interest in case we want to consider what new products might have adequate demand. We can also see that the Helsinki branch has the most customers whereas Vantaa has the least customers.

How are the assets under management (AUM) distributed?

```
>customer_assets <- sqldf("SELECT AVG(AUM) AS avg_aum,
                          SUM(AUM) AS total_aum,
                          COUNT(CustomerID) AS num_customers,
                          Segment
```

```
        FROM customers
        GROUP BY Segment")
```

Print the table customer_assets to see the average amount of assets per customer per segment (avg_aum), the total amount of assets of all the customers per segment (total_aum), the number of customers per segment (num_customers), and the segment.

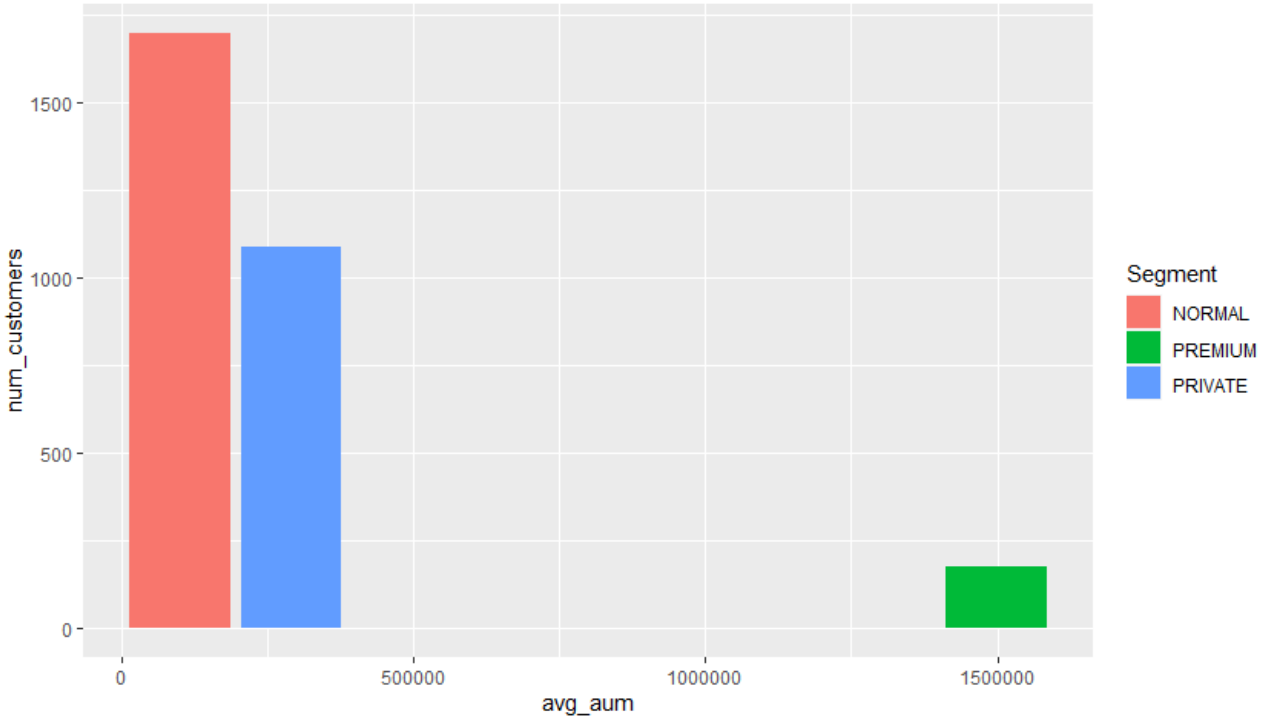
>customer_assets

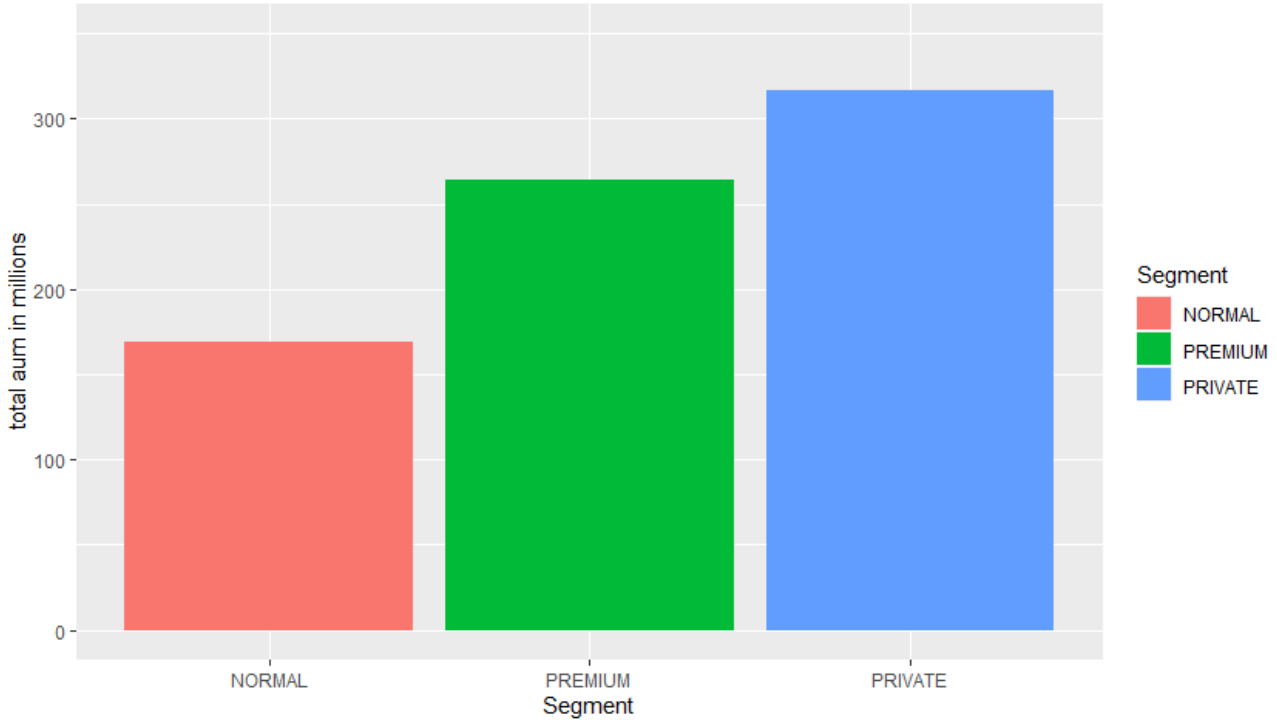
```
     avg_aum total_aum num_customers Segment
1   99468.09 168896825          1698  NORMAL
2 1498261.66 263694052           176 PREMIUM
3  291165.12 316496484          1087 PRIVATE
```

Make plots with the table customer_assets to see the distribution of the assets and the amount of customers in the different segments.

```
>ggplot(customer_assets, aes(x = avg_aum, y = num_customers, fill = Segment))+
    geom_col()
```

```
>ggplot(customer_assets, aes(x = Segment, y = total_aum/1000000, fill = Segment)) +
    geom_col()+
    scale_y_continuous(name= "total aum in millions", limits=c(0, 350))
```





57 % (1698) of Aktia's customers are in the normal segment, 6 % (176) are in the premium segment and 37 % (1087) are in the private segment. The average assets under management for normal customers are just under 100k, for premium customers just under 1,5m. and for private customers over 290k. However the normal customers bring the least assets to Aktia (22 %) whereas premium customers bring in 35 % and private customers bring in 42 % of all the assets that are managed in Aktia.

We might also want to know how much the customers age affect the amount of assets under management (AUM). This can be examined using linear regression. Our null hypothesis here is that age does not affect the AUM, h_0: Age=0.

>age_lm <- lm(customers$AUM ~ customers$Age, data=customers)
>summary(age_lm)

```
Call:
lm(formula = customers$AUM ~ customers$Age, data = customers)

Residuals:
    Min      1Q  Median      3Q     Max
-365761 -170661  -89027   27552 7919254

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    21846.9    26936.8   0.811    0.417
customers$Age   4874.7      540.7   9.016   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 449700 on 2959 degrees of freedom
Multiple R-squared:  0.02673,   Adjusted R-squared:  0.02641
F-statistic: 81.28 on 1 and 2959 DF,  p-value: < 2.2e-16
```

As we can see from the summary, the AUM grows 4874,7 euros when the customer gets one year older. The p-value is also very small (< 2.e-16) so the null hypothesis that age does not affect the amount of AUM can be rejected at confidence levels 10, 5, and 1 percent. However the R-squared is close to zero (0.02673) which means that the regression is not very well fitted and most of the variation in the AUM cannot be explained solely by age.

Next we shall examine whether there are differences in the amount of service payments before and after the change in list price, and how the change in prices affects the total income of service payments in different branches.

Divide the events into two tables according to the time when list prices changed.

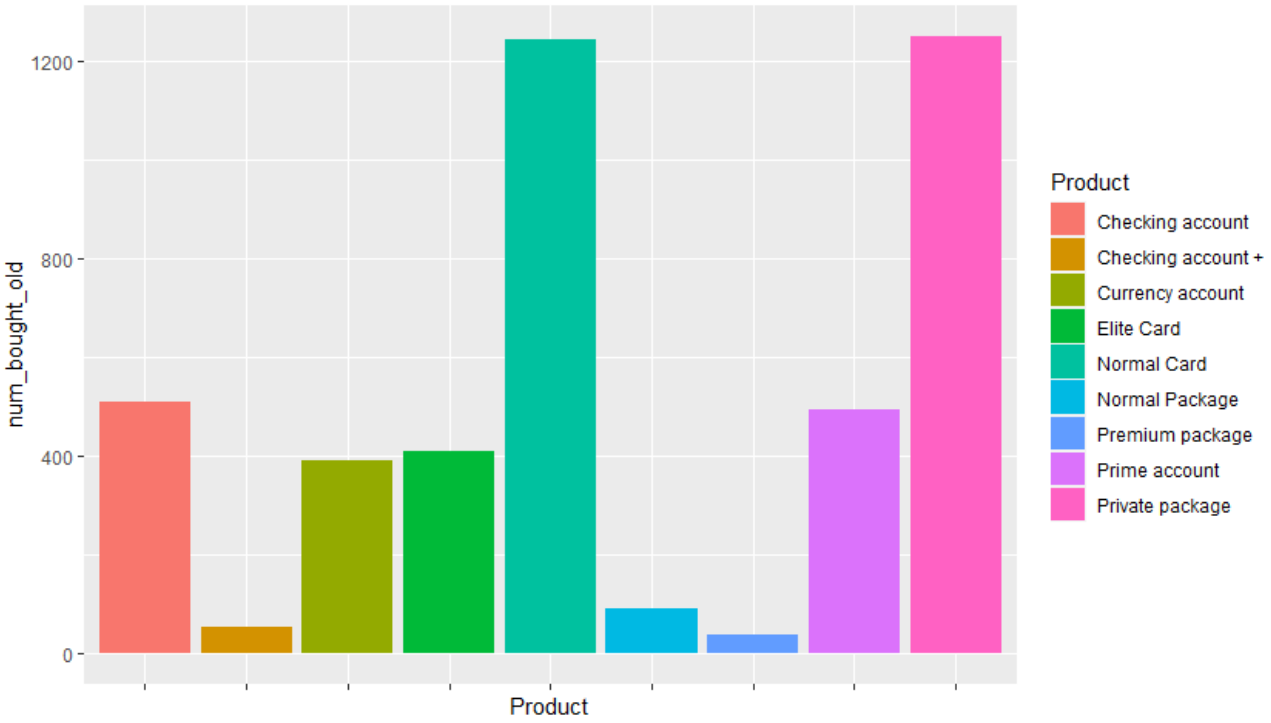
```
>events_new <- events %>%
      filter(Date >= as.Date("2019-01-01"))

>events_old <- events %>%
      filter(Date < as.Date("2019-01-01"))
```

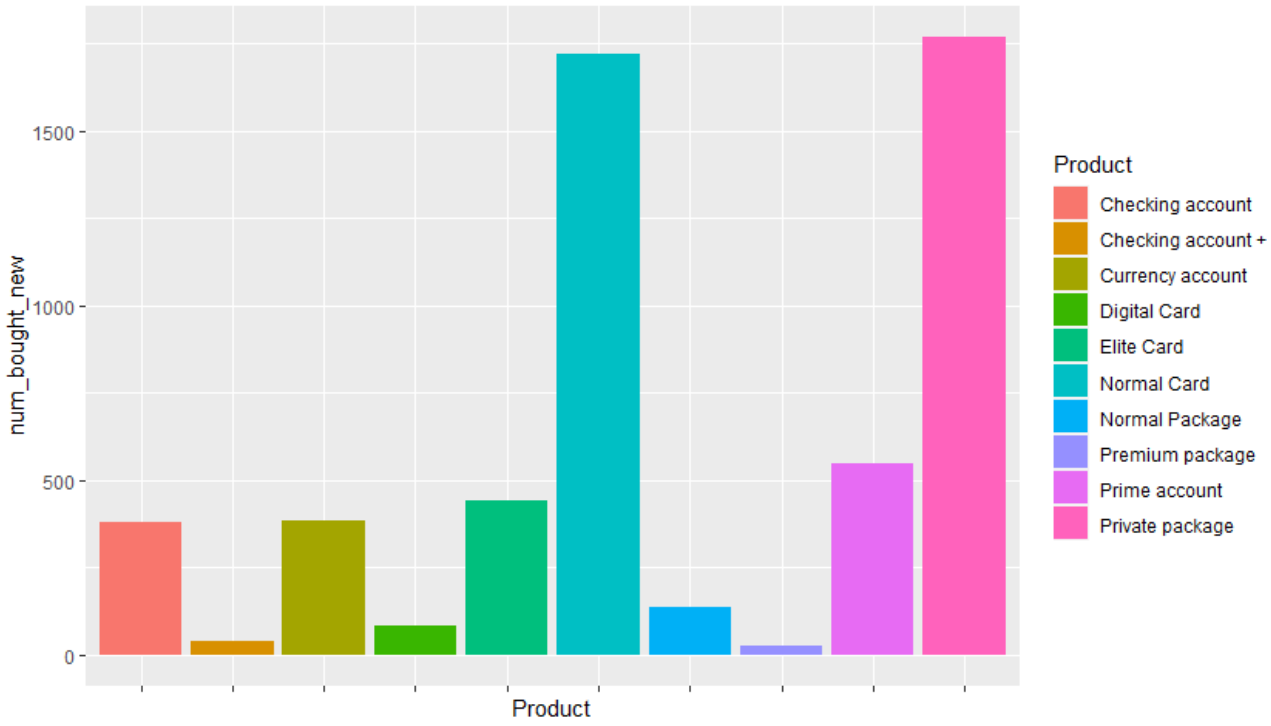
What products do customers pay for? Make two tables, one with bought products during the old list prices, and one with bought products during the new list prices.

```
>products_bought_old <- sqldf("SELECT COUNT(e.ProductID) AS num_bought_old,
                        p.Product,
                        e.ProductID
                        FROM events_old AS e
                        LEFT JOIN products AS p
                        ON e.ProductID = p.ProductID
                        GROUP BY e.ProductID
                        ORDER BY num_bought_old")

>products_bought_new <- sqldf("SELECT COUNT(e.ProductID) AS num_bought_new,
                        p.Product,
                        e.ProductID
                        FROM events_new AS e
                        LEFT JOIN products AS p
                        ON e.ProductID = p.ProductID
                        GROUP BY e.ProductID
                        ORDER BY num_bought_new")
```

Plot the amount of products that were bought.

```
>ggplot(products_bought_old, aes(x=Product, y= num_bought_old, fill=Product))+
                    geom_col()+
                    theme(axis.text.x=element_blank())

>ggplot(products_bought_new, aes(x=Product, y= num_bought_new, fill=Product))+
                    geom_col()+
                    theme(axis.text.x=element_blank())
```





Print the tables to get exact numbers.

```
> products_bought_old
  num_bought_old            Product ProductID
1             36    Premium package   AC13PRI
2             53 Checking account +   AC18CHE
3             89     Normal Package   CU15PRE
4            390   Currency account   AC16CUR
5            408         Elite Card   CA11NOR
6            494      Prime account   CA12DIG
7            508   Checking account   AC16CHE
8           1242        Normal Card   CU14NOR
9           1249    Private package   CU15PRI
> products_bought_new
   num_bought_new            Product ProductID
1              25    Premium package   AC13PRI
2              38 Checking account +   AC18CHE
3              84       Digital Card   CA10ELI
4             136     Normal Package   CU15PRE
5             380   Checking account   AC16CHE
6             385   Currency account   AC16CUR
7             441         Elite Card   CA11NOR
8             549      Prime account   CA12DIG
9            1718        Normal Card   CU14NOR
10           1769    Private package   CU15PRI
```

After the price change the amount of the Normal and Private packages, the Elite and Normal Cards, and the Prime accounts increased, and the Digital Card was introduced. The amount of Premium packages, Checking accounts and Checking accounts +, and Currency accounts decreased after the change in prices.

Form tables that have the needed columns to analyze the development of collected service fees per branch. Exclude the payments that did not happen.

```
>payments_made_old <- sqldf("SELECT e.Payment,
                e.ProductID,
                p.Product,
                e.Discount_percent,
                e.Payment_eur,
                c.Branch,
                e.Date,
                p.old_prices
                FROM events_old AS e
                LEFT JOIN customers AS c
                ON e.CustomerID = c.CustomerID
                LEFT JOIN products AS p
                ON e.ProductID = p.ProductID
                WHERE Payment IS 'TOSI'
                GROUP BY c.Branch,
                e.ProductID,
                e.Discount_percent,
                e.Payment
                ORDER BY Branch")
```

```
>payments_made_new <- sqldf("SELECT e1.Payment,
                e1.ProductID,
                p.Product,
                e1.Discount_percent,
                e1.Payment_eur,
                c.Branch,
                e1.Date,
                p.new_prices
                FROM events_new AS e1
                LEFT JOIN customers AS c
                ON e1.CustomerID = c.CustomerID
                LEFT JOIN products AS p
                ON e1.ProductID = p.ProductID
                WHERE Payment IS 'TOSI'
                GROUP BY c.Branch,
                e1.ProductID,
                e1.Discount_percent,
                e1.Payment
                ORDER BY Branch")
```

Now we have the two tables we need in order to check for differences in the amount of service fees paid to each branch. We can see from the products table that the Digital card fee was introduces 1.1.2019 so the Digital card is not included in the table for old list prices. Let's print a table with the total fees collected to see whether the branches collected more fees before or after the price change. After that plot tables from which we can see the total amount of fees collected by branches per product.

```
>fees_collected_new <- sqldf("SELECT SUM(Payment_eur) AS total_fees_new,
                  Product,
                  Branch
                  FROM payments_made_new
                  GROUP BY Branch, Product")

>fees_collected_old <- sqldf("SELECT SUM(Payment_eur) AS total_fees_old,
                  Product,
                  Branch
                  FROM payments_made_old
                  GROUP BY Branch, Product")

>total_fees <- sqldf("SELECT  fcn.Branch,
                              SUM(total_fees_old),
                              SUM(total_fees_new)
                FROM fees_collected_new AS fcn
                LEFT JOIN fees_collected_old AS fco
                ON fcn.Product=fco.Product AND fcn.Branch=fco.Branch
                GROUP BY fcn.Branch")
```

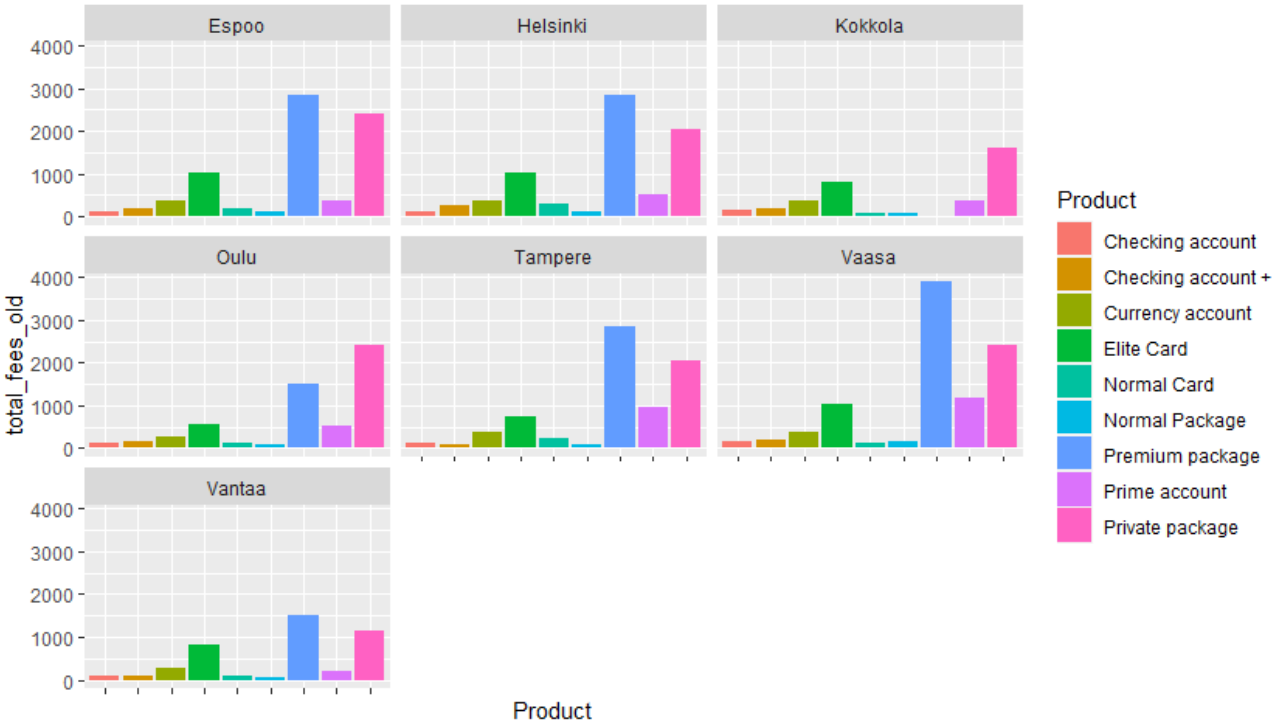
	Branch	SUM(total_fees_old)	SUM(total_fees_new)
1	Espoo	4804	4909
2	Helsinki	7608	9580
3	Kokkola	3740	5954
4	Oulu	5809	5522
5	Tampere	7570	8080
6	Vaasa	9484	10296
7	Vantaa	4293	5075

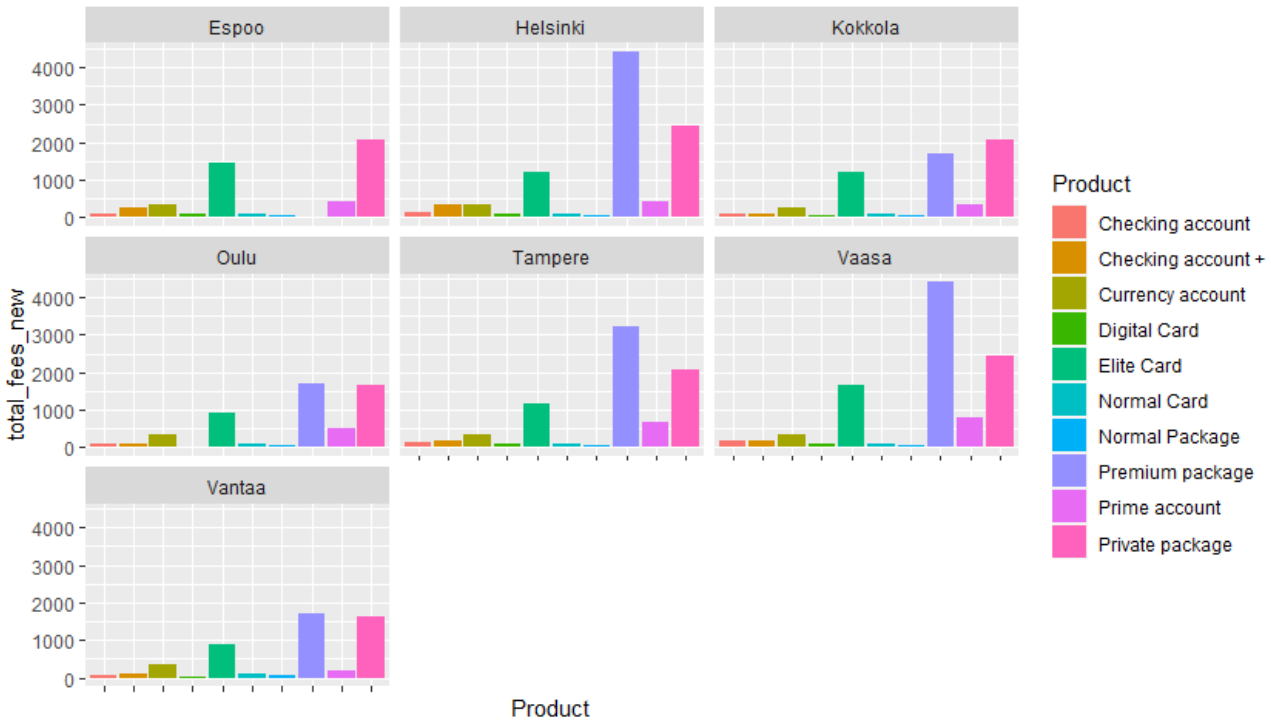
All braches except for Oulu collected more product fees after the price change than before.

Plot to see how the fees were distributed among products.

```
>ggplot(fees_collected_new, aes(x=Product, y= total_fees_new, fill=Product))+
                geom_col()+
                theme(axis.text.x=element_blank())+
                facet_wrap(~Branch)
```

```
>ggplot(fees_collected_old, aes(x=Product, y= total_fees_old, fill=Product))+
                geom_col()+
                theme(axis.text.x=element_blank())+
                facet_wrap(~Branch)
```





In almost all branches and with both prices, the products that yield the most fees to the branches are the Premium and Private Packages. The Prime Account was the most profitable account and the Elite Card was the most profitable card both before and after the price change. The Digital Card was approximately as profitable as the Normal Card so it did not yield much additional value compared to the total fees collected.

In Espoo the biggest changes in the collected fees were an increase of the Elite Card fees and a decrease from almost 3000 eur to 0 eur in Premium Package fees.

In Helsinki the biggest change was a 1500 eur increase in Premium Package fees.

In Kokkola the biggest changes were an increase from 0 to over 1500 eur in Premium Package fees, and a 500 eur decrease in Private Package fees.

In Oulu the biggest changes were a small increase in the Elite Card fees and an almost 1000 eur decrease in Private Package fees.

In Tampere there was an increase of some hundreds of euros in the Prime Account fees.

In Vaasa the Elite Card fees increased 500 eur and the Prime Account fees decreased some hundred euros.

In Vantaa the Private Package fees increased from 1000 to 1500 eur.

There were also other changes that were small compared to the total amount of collected fees.