PEG Africa Case Study Submission by Marvin Adjei Kojo Lomo

Marvin Adjei Kojo Lomo

3/25/2021

Case Study 1: PEG Africa - Repayment

Scenario

PEG Africa's main clients are consumers who lack both access to reliable electricity and any formal banking services (a.k.a the unbanked). The firm offers a sustainable option to meet residential electricity needs via the Solar Home System (three lights, phone and a radio) to consumers (living on $5 - $10). Consumers can build their credit for additional products and services over time.

Ghana Company Profile

1. 35 Service Centers spread into 7 regions.
2. Region is managed by an ASM covering about 5 Service Centers.
3. Each Service Center has 4 - 6 DSR managed by the Sales Field Manager.

Problem Statement

Despite increase in sales over the past three years, reaching 42000 customers across Ghana, there is a high default rate of solar devices. Bad payer behavior has a huge cost for the company.

Adidome Service Center in the Volta Region

Portfolio - 198 customers from 2015 - 2017. Repayment rate - 36%

Task: An understanding of the External Drivers which could explain such a low repayment rate at Adidome compared to the average repayment rate using data.

What could possibly be the situation here? Could it be that generally socio-economic conditions have plummeted so much that consumers can no longer afford PEG Afica's products or the has become an available alternative from competitors or that the National Grid has connected them online now?

To answer this question I used the Exploratory Data Analysis methodology combined with literature review on loan repayments in Ghana. Due to the paucity of data on District-level demographics, unavailability of Customer-level information at the Adidome Service Center and only 48 Service Center records given, statistical models investigated did not show any significance.

Data preparation

```
# Lets load the required libraries
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(rpart)



# Load the data
repayment_Data <- read_excel("Case 1 Data.xls")


# Converting to tibble Data_frame
repayment_Data <- as_tibble(repayment_Data)

# Lets take a look at the Data
head(repayment_Data)
```

```
## # A tibble: 6 x 4
##   Name                        Region      `Repayment rate` Sales
##   <chr>                       <chr>                  <dbl> <dbl>
## 1 Asamankese Service Center   Eastern                 0.84   333
## 2 Tafo Service Center 2       Ashanti                 0.78    83
## 3 Juaso Service Centre        Ashanti                 0.77    18
## 4 Begoro Service Center       Eastern                 0.72    60
## 5 Juapong - Volta main street Volta South             0.7    802
## 6 Mankessim Central           Central                 0.66   809
```

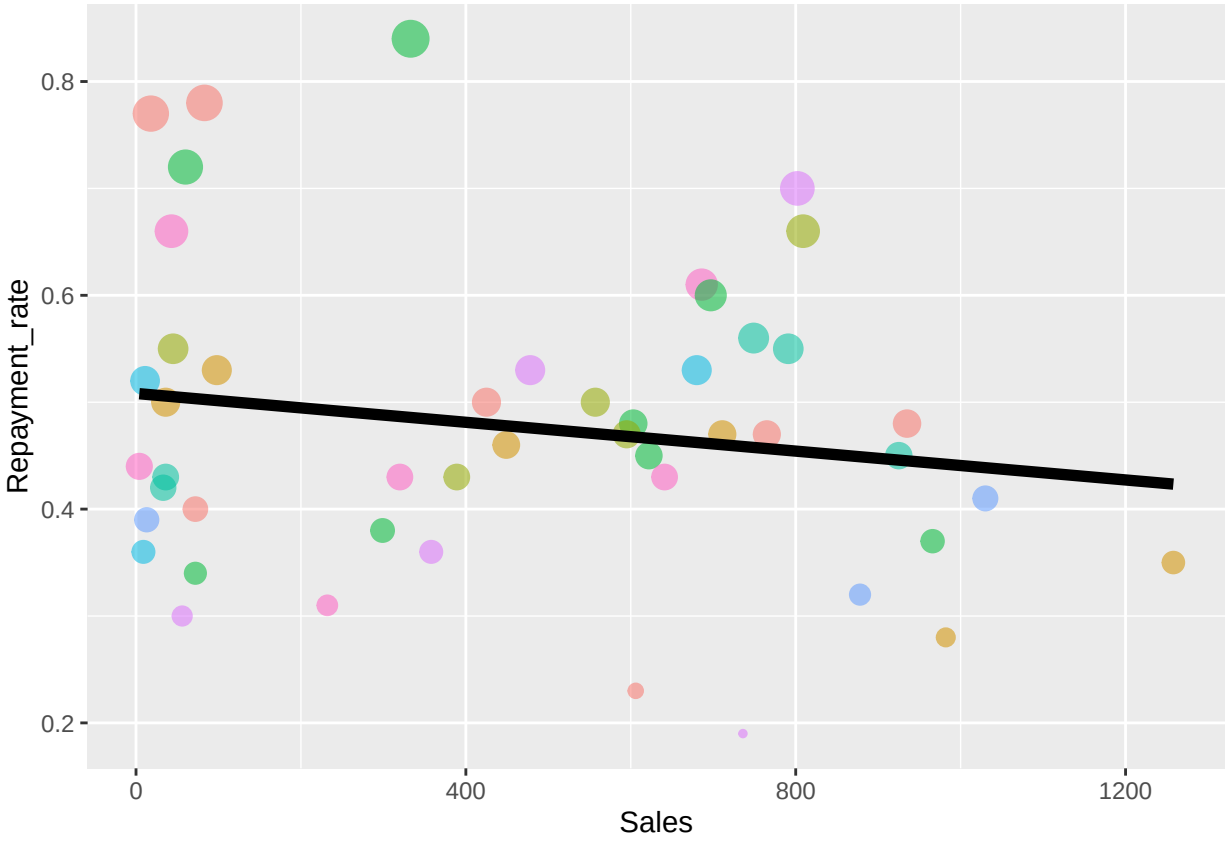
```
# Renaming the Repayment Rate column
repayment_Data$Repayment_rate <- repayment_Data$`Repayment rate`

repayment_Data <- repayment_Data[ , c(1, 2, 4, 5)]


# Converting the region and SC_name to a factor variables

repayment_Data$Region <- as.factor(repayment_Data$Region)
repayment_Data$Name <- as.factor(repayment_Data$Name)

# Focusing in on Adidome_SC
adidome_SC_Data <- repayment_Data %>%
  filter(Region == "Volta South", Name == "Adidome Service Center")
```

Visualizing the Repayment Data

```
reg_line <- repayment_Data %>%
  select("Sales", "Repayment_rate")
```

```r
# Investigating Sales vs Repayment rate
repayment_Data %>%
  ggplot(aes(x = Sales , y = Repayment_rate)) +
  geom_point(aes(size =  Repayment_rate, color = Region, alpha = 2), show.legend = FALSE) +
  geom_smooth(data = reg_line, aes(x = Sales , y = Repayment_rate),
              method = lm ,se = FALSE, color = "black", lwd = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```r
# Lets look at the correlation between repayment rate with Sales
cor(repayment_Data$Sales, repayment_Data$Repayment_rate)
```

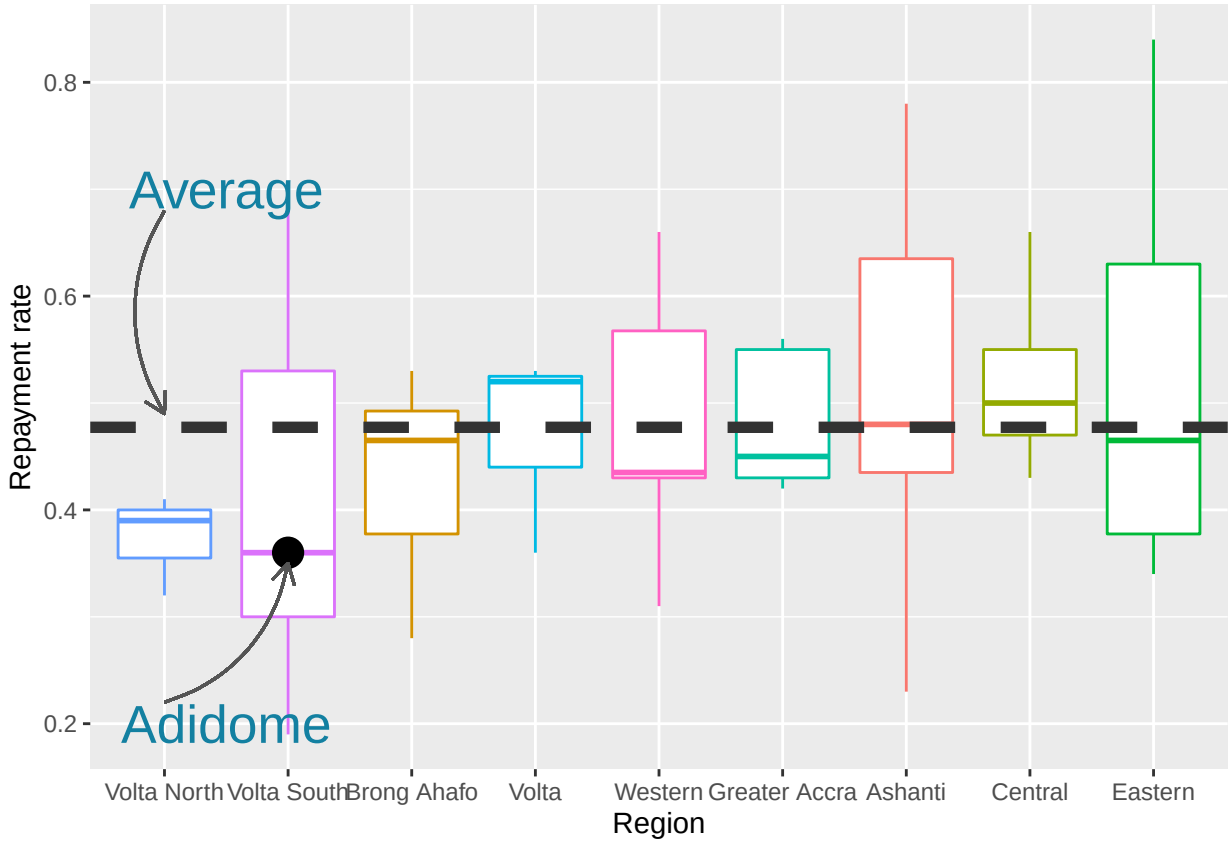
```
## [1] -0.1709478
```

The negative correlation between sales and repayment_rate suggests that Service Centers with high Sales also tend to have a low repayment rate.

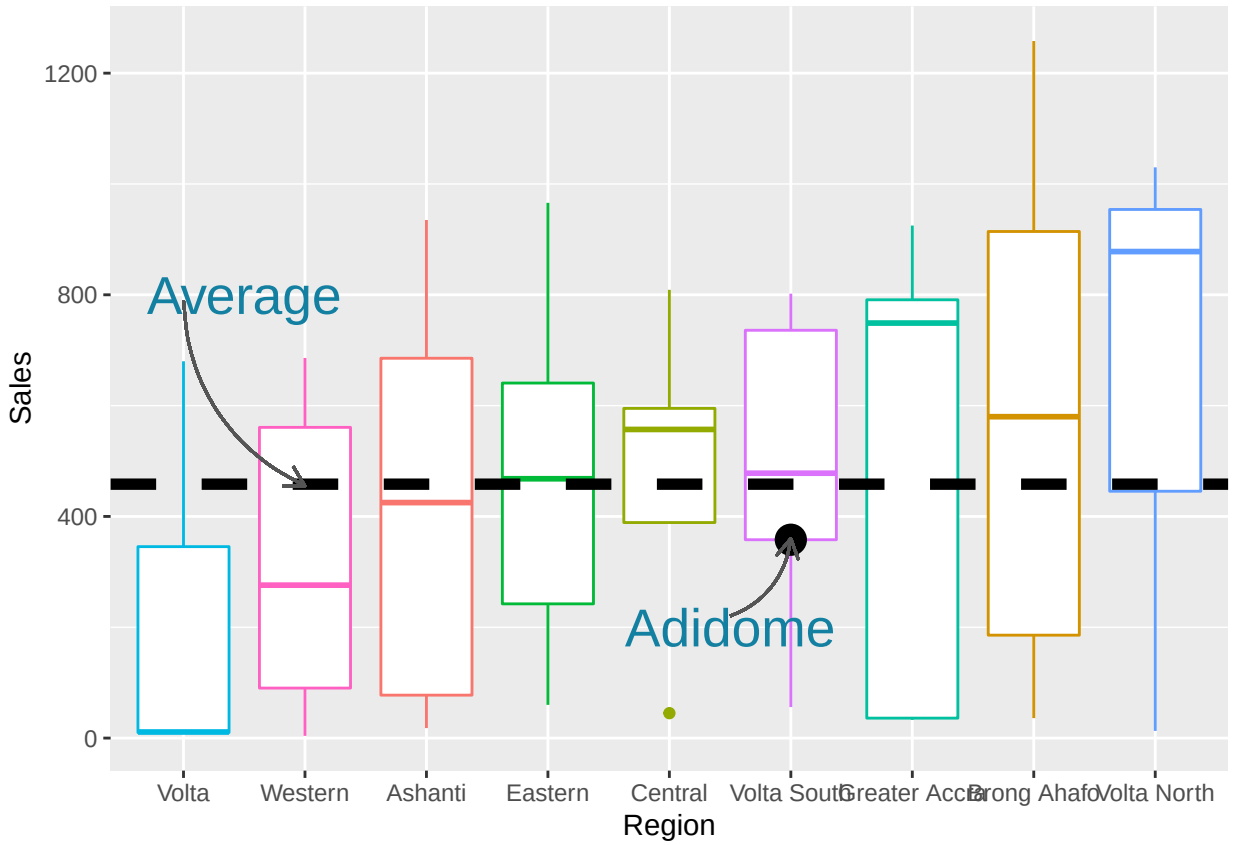
Visualizing the Sales per Region

```r
# Investigating average repayment rate at centers vs adidome_SC
repayment_Data %>%
  ggplot(aes(reorder(x = Region, Repayment_rate), y = Repayment_rate, color = Region)) +
  geom_boxplot(show.legend = FALSE) +
  geom_hline(yintercept = mean(repayment_Data$Repayment_rate), lty = 2, lwd = 2, colour = "#333333") +
  geom_point(data = adidome_SC_Data, aes( x = Region, y = Repayment_rate),
             color = "black", size = 5) +
```

```
  labs(x = "Region", y = "Repayment rate") +
  annotate(geom = "text", label = "Average", x = 1.5, y = 0.7,
           colour = "#1380A1", family = "Helvetica", size = 7) +
  geom_curve(aes(x = 1, y = 0.68, xend = 1, yend = 0.49), colour = "#555555",
             size = 0.5, curvature = 0.3, arrow = arrow(length = unit(0.03, "npc"))) +
  annotate(geom = "text", label = "Adidome", x = 1.5, y = 0.2,
           colour = "#1380A1", family = "Helvetica", size = 7) +
  geom_curve(aes(x = 1, y = 0.22, xend = 2, yend = 0.35), colour = "#555555",
             size = 0.5, curvature = 0.3, arrow = arrow(length = unit(0.03, "npc")))
```




```
# Investigating average Sales at centers vs adidome_SC
repayment_Data %>%
  ggplot(aes(reorder(x = Region, Sales), y = Sales, color = Region)) +
  geom_boxplot(show.legend = FALSE) +
  geom_hline(yintercept = mean(repayment_Data$Sales), lty = 2, lwd = 2) +
  geom_point(data = adidome_SC_Data, aes(x = Region, y = Sales),
             color = "black", size = 5) +
  labs(x = "Region", y = "Sales") +
  annotate(geom = "text", label = "Average", x = 1.5, y = 800,
           colour = "#1380A1", family = "Helvetica", size = 7) +
  geom_curve(aes(x = 1, y = 790, xend = 2, yend = 455), colour = "#555555",
             size = 0.5, curvature = 0.3, arrow = arrow(length = unit(0.03, "npc"))) +
  annotate(geom = "text", label = "Adidome", x = 5.5, y = 200,
           colour = "#1380A1", family = "Helvetica", size = 7) +
  geom_curve(aes(x = 5.5, y = 220, xend = 6, yend = 360), colour = "#555555",
             size = 0.5, curvature = 0.3, arrow = arrow(length = unit(0.03, "npc")))
```




```
# It can be seen above that the adidome SC is also selling below the average SC Sales.
```

The gap between the Repayment rate at the Adidome Service Center and the Average Service Center repayment rate as seen in the Chart 2 can probably be explained by the demographics of the Adidome Area. Various studies on loan repayments conducted have estimated that external drivers that impact loan repayments include

1. Age
2. Education
3. Gender
4. Household size
5. Loan size
6. Purpose of loan
7. Savings

In the context of the Adidome area, other factors such as whether the individual is connected to the National Grid and Household income can be added to the list of factors that may affect the Repayment rate.

In addition, from a recent report released by the statistical service on Multidimensional Poverty Index of regions, as at 2017, the Volta region along with the three Northern regions were ranked as the poorest regions in the country. This could explain why out of the 10 worst performing Service Centers based on Repayment rate, 5 of the 10 are from the Volta Region.

```
sc_ranked <- repayment_Data %>%
  arrange(Repayment_rate)

head(sc_ranked, 10)
```

```
## # A tibble: 10 x 4
##    Name                     Region      Sales Repayment_rate
```

```
##    <fct>                    <fct>       <dbl>          <dbl>
##  1 Dzemeni Service Centre   Volta South   736           0.19
##  2 Agogo Service Centre     Ashanti       606           0.23
##  3 Atebubu Service Centre   Brong Ahafo   982           0.28
##  4 Dzemeni Service Centre 2 Volta South    56           0.3
##  5 Tarkwa                   Western       232           0.31
##  6 Dambai                   Volta North   878           0.32
##  7 Asesewa Service Center 2 Eastern        72           0.34
##  8 Kintampo Service Centre  Brong Ahafo  1258           0.35
##  9 Adidome Service Center   Volta South   358           0.36
## 10 Dambai  2                Volta           9           0.36
```

Given the above, the following Hypothesis can be constructed:

1. Null Hypothesis: The above factors do not affect Repayment rate.
2. Alternative Hypothesis: The above factors affect Repayment rate.

Further data collected from customers, and modeling perfomed to understand the factors influencing customer payment behavior.

Should I be given the opportunity to join PEG Africa, I would love to continue to investigate this negative customer behavior, predict credit score based on payment frequency to reduce the incidence of non-payment of loans.

Case Study 2: PEG Africa - Repayment

Scenario 1:

Given the DB write a SQL Query to get the amount paid by each customer, create a new column called "Total Amount Paid" and use this amount to create a column called outstanding balance in which I will add the customer's outstanding balance after calculating "Total Amount Paid".

```
# Question 1

# Loading the Dataset
db_1 <- read_excel(path = "Case 2 Data.xlsx", sheet = 1)


# Verifying Head Data
head(db_1)
```

```
## # A tibble: 6 x 7
##   ContractId CustomerId Deposit ProductPrice IncomingTransaction~ Amount sc_name
##   <chr>      <chr>      <chr>   <chr>        <chr>                <chr>  <chr>
## 1 1197908    89891      39000   288000       1547963              39000  COCODY
## 2 1197908    89891      39000   288000       1548630              2000   COCODY
## 3 1197908    89891      39000   288000       1548844              2000   COCODY
## 4 1197908    89891      39000   288000       1652933              395    COCODY
## 5 1197908    89891      39000   288000       1745740              1000   COCODY
## 6 1197908    89891      39000   288000       1819734              1000   COCODY
```

```
# Converting Amount and Product_Price to Numeric
db_1$Amount <- as.numeric(db_1$Amount)
db_1$ProductPrice <- as.numeric(db_1$ProductPrice)


# Total Amount Paid by the Customer
```

```r
total_amount_paid_by_Customer <- db_1 %>%
  group_by(CustomerId, ContractId, ProductPrice) %>%
  summarise(total_Payment = sum(Amount))
```

```
## `summarise()` regrouping output by 'CustomerId', 'ContractId' (override with `.groups` argument)
```

```r
# Preview the result of the Query
head(total_amount_paid_by_Customer, 10)
```

```
## # A tibble: 10 x 4
## # Groups:   CustomerId, ContractId [10]
##    CustomerId ContractId ProductPrice total_Payment
##    <chr>      <chr>             <dbl>         <dbl>
##  1 100035     1208028          144500         31550
##  2 100039     1207964          144500         68500
##  3 100152     1208512          144500         72000
##  4 100154     1208433          144500         66900
##  5 100155     1207911          144500         48000
##  6 100273     1228104          144500        163465
##  7 100290     1197913          288000        129400
##  8 100310     1207892          144500        175850
##  9 100328     1207885          144500        153800
## 10 100445     1207867          144500        101000
```

```r
# Clients with More than one Contract
customer_with_more_than_one_Contract <- total_amount_paid_by_Customer %>%
  # Filter by the customerID
  group_by(CustomerId) %>%
  # Count the number of times each Customer ID occurs
  summarise(freq = n()) %>%
  # Show Customers with more than one contract
  filter( freq > 1)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# Preview list of Customers with more than one Contract
head(customer_with_more_than_one_Contract, 10)
```

```
## # A tibble: 10 x 2
##    CustomerId  freq
##    <chr>      <int>
##  1 123469         2
##  2 125261         2
##  3 132151         2
##  4 151680         2
##  5 165438         2
##  6 168073         2
##  7 175217         2
##  8 178057         2
##  9 185187         3
## 10 185826         2
```

```r
# Adding the Outstanding Balances

customer_Outstanding_Balance <- total_amount_paid_by_Customer %>%
  mutate(outstanding_Balance = ProductPrice  - total_Payment)
```

```r
head(customer_Outstanding_Balance, 10)
```

```
## # A tibble: 10 x 5
## # Groups:   CustomerId, ContractId [10]
##    CustomerId ContractId ProductPrice total_Payment outstanding_Balance
##    <chr>      <chr>             <dbl>         <dbl>               <dbl>
##  1 100035     1208028          144500         31550              112950
##  2 100039     1207964          144500         68500               76000
##  3 100152     1208512          144500         72000               72500
##  4 100154     1208433          144500         66900               77600
##  5 100155     1207911          144500         48000               96500
##  6 100273     1228104          144500        163465              -18965
##  7 100290     1197913          288000        129400              158600
##  8 100310     1207892          144500        175850              -31350
##  9 100328     1207885          144500        153800               -9300
## 10 100445     1207867          144500        101000               43500
```

Please Note that the Case Study was not clear on what happens to the Deposit of the Customer. The Query above totals all Amounts paid by the customer for each contract he/she holds without the Customer Down Payment.

If Customer Downpayment is added to amounts paid, the resulting Query is:

```r
# Now, Computing Total Amount Paid with Deposit inclusive

# Loading the Dataset
db_1 <- read_excel(path = "Case 2 Data.xlsx", sheet = 1)


# Verifying Head Data
head(db_1)
```

```
## # A tibble: 6 x 7
##   ContractId CustomerId Deposit ProductPrice IncomingTransaction~ Amount sc_name
##   <chr>      <chr>      <chr>   <chr>        <chr>                <chr>  <chr>
## 1 1197908    89891      39000   288000       1547963              39000  COCODY
## 2 1197908    89891      39000   288000       1548630              2000   COCODY
## 3 1197908    89891      39000   288000       1548844              2000   COCODY
## 4 1197908    89891      39000   288000       1652933              395    COCODY
## 5 1197908    89891      39000   288000       1745740              1000   COCODY
## 6 1197908    89891      39000   288000       1819734              1000   COCODY
```

```r
# Converting Amount and Product_Price to Numeric
db_1$Amount <- as.numeric(db_1$Amount)
db_1$ProductPrice <- as.numeric(db_1$ProductPrice)
db_1$Deposit <- as.numeric(db_1$Deposit)

total_amount_paid_by_Customer_with_Deposit <- db_1 %>%
  group_by(CustomerId, ContractId, ProductPrice, Deposit) %>%
  summarise(total_Payment = sum(Amount))
```

```
## `summarise()` regrouping output by 'CustomerId', 'ContractId', 'ProductPrice' (override with `.groups
```

```r
# Preview of Field
head(total_amount_paid_by_Customer_with_Deposit, 10)
```

```
## # A tibble: 10 x 5
```

```
## # Groups:   CustomerId, ContractId, ProductPrice [10]
##    CustomerId ContractId ProductPrice Deposit total_Payment
##    <chr>      <chr>             <dbl>   <dbl>         <dbl>
##  1 100035     1208028          144500   19500         31550
##  2 100039     1207964          144500   19500         68500
##  3 100152     1208512          144500   19500         72000
##  4 100154     1208433          144500   19500         66900
##  5 100155     1207911          144500   19500         48000
##  6 100273     1228104          144500   19500        163465
##  7 100290     1197913          288000   39000        129400
##  8 100310     1207892          144500   19500        175850
##  9 100328     1207885          144500   19500        153800
## 10 100445     1207867          144500   19500        101000
```

```
# total_amount_paid_plus_Deposit = Deposit + Total_Paid_Amount
total_amount_paid_by_Customer_Plus_Deposit <-
  total_amount_paid_by_Customer_with_Deposit %>%
  mutate(total_amount_paid_plus_Deposit = Deposit + total_Payment)


# Preview
head(total_amount_paid_by_Customer_Plus_Deposit, 10)
```

```
## # A tibble: 10 x 6
## # Groups:   CustomerId, ContractId, ProductPrice [10]
##    CustomerId ContractId ProductPrice Deposit total_Payment total_amount_paid_p~
##    <chr>      <chr>             <dbl>   <dbl>         <dbl>                <dbl>
##  1 100035     1208028          144500   19500         31550                51050
##  2 100039     1207964          144500   19500         68500                88000
##  3 100152     1208512          144500   19500         72000                91500
##  4 100154     1208433          144500   19500         66900                86400
##  5 100155     1207911          144500   19500         48000                67500
##  6 100273     1228104          144500   19500        163465               182965
##  7 100290     1197913          288000   39000        129400               168400
##  8 100310     1207892          144500   19500        175850               195350
##  9 100328     1207885          144500   19500        153800               173300
## 10 100445     1207867          144500   19500        101000               120500
```

```
# Outstanding Payments = Product_Price - (Total_Amount_Paid + Deposit)

customer_Outstanding_Balance_after_Deposit_and_Tot_Payt <- total_amount_paid_by_Customer_Plus_Deposit %
  mutate(Outstanding_Balance_after_Deposit_and_Tot_Payt = ProductPrice - total_amount_paid_plus_Deposit


# Preview
head(customer_Outstanding_Balance_after_Deposit_and_Tot_Payt[ , -3], 10)
```

```
## # A tibble: 10 x 6
## # Groups:   CustomerId, ContractId [10]
##    CustomerId ContractId Deposit total_Payment total_amount_pa~ Outstanding_Bal~
##    <chr>      <chr>        <dbl>         <dbl>            <dbl>            <dbl>
##  1 100035     1208028      19500         31550            51050            93450
##  2 100039     1207964      19500         68500            88000            56500
##  3 100152     1208512      19500         72000            91500            53000
##  4 100154     1208433      19500         66900            86400            58100
```

```
##  5 100155     1207911      19500         48000           67500           77000
##  6 100273     1228104      19500        163465          182965          -38465
##  7 100290     1197913      39000        129400          168400          119600
##  8 100310     1207892      19500        175850          195350          -50850
##  9 100328     1207885      19500        153800          173300          -28800
## 10 100445     1207867      19500        101000          120500           24000
```

Scenario 2:

Write a query to get the sum_paid_to_date and arrears of each contractID at their maximum date of activity.

```
# Load the DB
db_2 <- read_excel(path = "Case 2 Data.xlsx", sheet = 2)


# Lets look at the Data

head(db_2)
```

```
## # A tibble: 6 x 6
##   contractid customerid activation_date     date_of_activity    sum_paid_to_date
##        <dbl>      <dbl> <dttm>              <dttm>                         <dbl>
## 1          1         12 2018-06-01 10:08:53 2019-07-24 00:00:00           433508
## 2          1         12 2018-06-01 10:08:53 2019-07-25 00:00:00           433508
## 3          1         12 2018-06-01 10:08:53 2019-07-26 00:00:00           433508
## 4          1         12 2018-06-01 10:08:53 2019-07-27 00:00:00           433508
## 5          1         12 2018-06-01 10:08:53 2019-07-28 00:00:00           433508
## 6          1         12 2018-06-01 10:08:53 2019-07-30 00:00:00           433508
## # ... with 1 more variable: arrears <dbl>
```

```
# Getting the sum_paid_to_date and arrears at maximum date of activity

sum_Paid_to_Date <- db_2 %>%
  # Grouping by CustomerID and ContractID
  group_by(customerid, contractid) %>%
  # Filtering the Group by maximum activity date
  filter(date_of_activity == max(date_of_activity)) %>%
  arrange(contractid)


head(sum_Paid_to_Date[ , -c(3)], 10)
```

```
## # A tibble: 10 x 5
## # Groups:   customerid, contractid [10]
##    contractid customerid date_of_activity    sum_paid_to_date arrears
##         <dbl>      <dbl> <dttm>                         <dbl>   <dbl>
##  1          1         12 2019-11-24 00:00:00          500728       0
##  2          2         19 2021-03-07 00:00:00          815500   33225.
##  3          3          4 2021-03-07 00:00:00          751300   53323.
##  4          4         20 2021-03-07 00:00:00          785000   10810.
##  5          5         21 2021-03-07 00:00:00          812050  104434.
##  6          6         24 2021-03-07 00:00:00         1058750       0
##  7          7         23 2020-12-22 00:00:00          750489. 190140.
##  8          8         25 2021-03-07 00:00:00          985000   18743.
##  9          9         26 2020-02-20 00:00:00          357000       0
```

```
## 10         10         27 2020-02-28 00:00:00         1195000       0
```

Case Study 3:

Scenario: Demonstrating Business Understanding

For this section, I will break the corresponding KPI metrics into the following:

1. Overall: Total_Sales_by_Country, Product_Performance_per_country, Active_vs_Blocked_Customers_per_Country
2. Sales: Active_Contracts_per_SC, Blocked_Contracts_per_SC, Best_Performing_Products, Product_Sales_per_Country
3. Credit Management: Customer_performance_to_Date, Expected_DailyReceivable_vs_ActualReceived.
4. Finance: Expected_Accounts_Receivable_vs_Actual_Accounts_Receivable, ActualDaily_Deposits_vs_Target, Cash-to-Cash-Cycle.

Group Level Data Analysis

Firstly, lets see which country is performing better:

```
# Loading the dataset
db_3 <- read_excel("Case 3-Sample data.xlsx")

# Preview the data
head(db_3)
```

```
## # A tibble: 6 x 14
##   ContractId Sum_Paid_To_Date Credits_End_Date    dailyAmount Deposit_Amount
##        <dbl>            <dbl> <dttm>                    <dbl>          <dbl>
## 1      32168              534 2019-05-07 00:00:00         2              99
## 2      31942              690 2019-05-07 00:00:00         2              99
## 3      30482              300 2019-05-07 00:00:00         0              50
## 4      29512              350 2019-05-07 00:00:00         2              99
## 5      23911              305 2019-05-07 00:00:00         2.5            99
## 6      23585              432 2019-05-07 00:00:00         2.5            99
## # ... with 9 more variables: Product_Price <dbl>, Activation_Date <dttm>,
## #   `Sales location` <chr>, Outstanding_Balance <dbl>,
## #   sumPaidMinusDeposit <dbl>, customerStatus <chr>, expectedTotalAmount <dbl>,
## #   CountryId <chr>, productTypeGeneral <chr>
```

```
# Converting CustomerStatus to numeric
db_3$customerStatus <- ifelse(db_3$customerStatus == "Blocked", 0, 1)

# Converting Sum_Paid_to_Numeric
db_3$Sum_Paid_To_Date <- as.numeric(db_3$Sum_Paid_To_Date)


db_3 %>%
  group_by(CountryId) %>%
  summarise(Tot_Amt_Received_to_Date_loc_Curr = sum(Sum_Paid_To_Date)) %>%
  arrange(desc(Tot_Amt_Received_to_Date_loc_Curr))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   CountryId Tot_Amt_Received_to_Date_loc_Curr
```

```
##   <chr>                                  <dbl>
## 1 SN                                  10290334.
## 2 CDI                                  3768171.
## 3 GH                                    107472.
```

Looks like PEG Africa's Senegal Office is doing something right!

Next, we find out which Product is selling the most:

```
db_3 %>%
  group_by(productTypeGeneral) %>%
  summarise(sum_paid_per_product = sum(Sum_Paid_To_Date)) %>%
  arrange(desc(sum_paid_per_product))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   productTypeGeneral sum_paid_per_product
##   <chr>                             <dbl>
## 1 TV-X1000                        4957784.
## 2 TV-X740                         4170303.
## 3 TV-X850                         3768390.
## 4 SHS                             1267590
## 5 Add-on                             1910
```

Now, how many Customers does each country have? Lets find out:

```
db_3 %>%
  group_by(CountryId) %>%
  summarise(No_of_Customers = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   CountryId No_of_Customers
##   <chr>               <int>
## 1 CDI                    77
## 2 GH                    156
## 3 SN                    163
```

Finally, lets find out the number of Active customers per group

```
db_3 %>%
  filter(customerStatus == 1) %>%
  group_by(CountryId) %>%
  summarise(Number_of_Active_Cust = sum(customerStatus))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   CountryId Number_of_Active_Cust
##   <chr>                     <dbl>
## 1 GH                          101
## 2 SN                          163
```

It gets interesting as all customers in region CDI are blocked. What could be the reason for this? Also, Senegal has more active clients than the Ghana office. Lets deep-dive into Senegal to find out what is hapenning in there.

A Dive into Senegal

The best performing region is Senegal is: "SC Gandiaye":

```
# Senegal Data
db_3 %>%
  filter(CountryId == "SN", customerStatus == 1) %>%
  group_by(`Sales location`) %>%
  summarise(Earnings_to_Date = sum(Sum_Paid_To_Date)) %>%
  arrange(desc(Earnings_to_Date))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 10 x 2
##    `Sales location` Earnings_to_Date
##    <chr>                       <dbl>
##  1 SC Gandiaye              1628536.
##  2 SC Ziguinchor            1621186.
##  3 SC Kaffrine              1424957.
##  4 SC Nioro                 1316237.
##  5 SC Diouloulou            1228043.
##  6 SC Koungheul              875387.
##  7 SC Kolda                  830894.
##  8 SC Diourbel               765593.
##  9 SC Velingara              527000
## 10 SC Tambacounda             72500
```

Is there a repayment problem? Lets find this out in Senegal:

```
# Repayment in Senegal
db_3 %>%
  filter(CountryId == "SN") %>%
  group_by(`Sales location`) %>%
  summarise(Repayment_rate_Perc = (mean(customerStatus) * 100)) %>%
  arrange(desc(Repayment_rate_Perc))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 10 x 2
##    `Sales location` Repayment_rate_Perc
##    <chr>                          <dbl>
##  1 SC Diouloulou                    100
##  2 SC Diourbel                      100
##  3 SC Gandiaye                      100
##  4 SC Kaffrine                      100
##  5 SC Kolda                         100
##  6 SC Koungheul                     100
##  7 SC Nioro                         100
##  8 SC Tambacounda                   100
##  9 SC Velingara                     100
## 10 SC Ziguinchor                    100
```

Surprisingly, there is no repayment problem in Senegal. I suggest that, management at the group-level investigate strategies being implemented in Senegal's provinces and repeat same at the other locations.

What product sells most in Senegal?

```
# Best performing Products  in Senegal
db_3 %>%
```

```
  filter(CountryId == "SN") %>%
  group_by(productTypeGeneral) %>%
  summarise(Sales_by_Product = n()) %>%
  arrange(desc(Sales_by_Product))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   productTypeGeneral Sales_by_Product
##   <chr>                        <int>
## 1 TV-X740                         93
## 2 TV-X1000                        51
## 3 TV-X850                         19
```

A look at Ghana

Lets take a look at the best performing Service Center in Ghana. Looks like Bogoso leads the pack!

```
db_3 %>%
  filter(CountryId == "GH", customerStatus == 1) %>%
  group_by(`Sales location`) %>%
  summarise(Earnings_to_Date = sum(Sum_Paid_To_Date)) %>%
  arrange(desc(Earnings_to_Date)) %>%
  head(10)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 10 x 2
##    `Sales location`                                Earnings_to_Date
##    <chr>                                                      <dbl>
##  1 Bogoso Service Centre                                      7663.
##  2 Ada-Kasseh Service Centre                                  6423.
##  3 Asankragua (former Assin Fosu - Central Region)            5784.
##  4 Telesales Service Center                                   5571
##  5 Sunyani                                                    5237.
##  6 Dambai                                                     5012
##  7 Kwahu Tafo Service Centre                                  4351
##  8 Lapaz Service Centre                                       3975
##  9 Kpandai                                                    3548
## 10 Atebubu Service Centre                                     3222
```

A look at the percentage of Active clients against total number of clients at the service center (as a measure of repayment since defaulting customers are "blocked") reveals a staggering problem.

```
# Repayment in Ghana
db_3 %>%
  filter(CountryId == "GH") %>%
  group_by(`Sales location`) %>%
  summarise(Repayment_rate_Perc = (mean(customerStatus) * 100)) %>%
  arrange(Repayment_rate_Perc) %>%
  head(10)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 10 x 2
##    `Sales location`                       Repayment_rate_Perc
##    <chr>                                                <dbl>
##  1 Adidome Service Center                                   0
```

```
##  2 Agona Swedru (former Mankessim Central)                  0
##  3 GOASO PARTNER SHOP                                       0
##  4 Ho                                                       0
##  5 Nkwanta Service Centre                                   0
##  6 Partnership Sales                                        0
##  7 Sampa                                                    0
##  8 Takoradi                                                 0
##  9 Tarkwa                                                   0
## 10 Bibiani Service Center                                  25
```

Ghanaian customers have a high default rate and are blocked due to non-repayment of loans. This is a revealing problem I have attempted to answer in case study 1. Demographic data is needed on these customers to understand the attitude of the Ghanaian PEG-Africa customer.

What products sell most in Ghana?

```
# Best performing Products  in Ghana
db_3 %>%
  filter(CountryId == "GH") %>%
  group_by(productTypeGeneral) %>%
  summarise(Sales_by_Product = n()) %>%
  arrange(desc(Sales_by_Product))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   productTypeGeneral Sales_by_Product
##   <chr>                         <int>
## 1 TV-X850                         124
## 2 SHS                              20
## 3 Add-on                            5
## 4 TV-X1000                          4
## 5 TV-X740                           3
```

Sample Visualization

```
library(tidyverse)
library(bbplot)
require(scales)
```

```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'
```

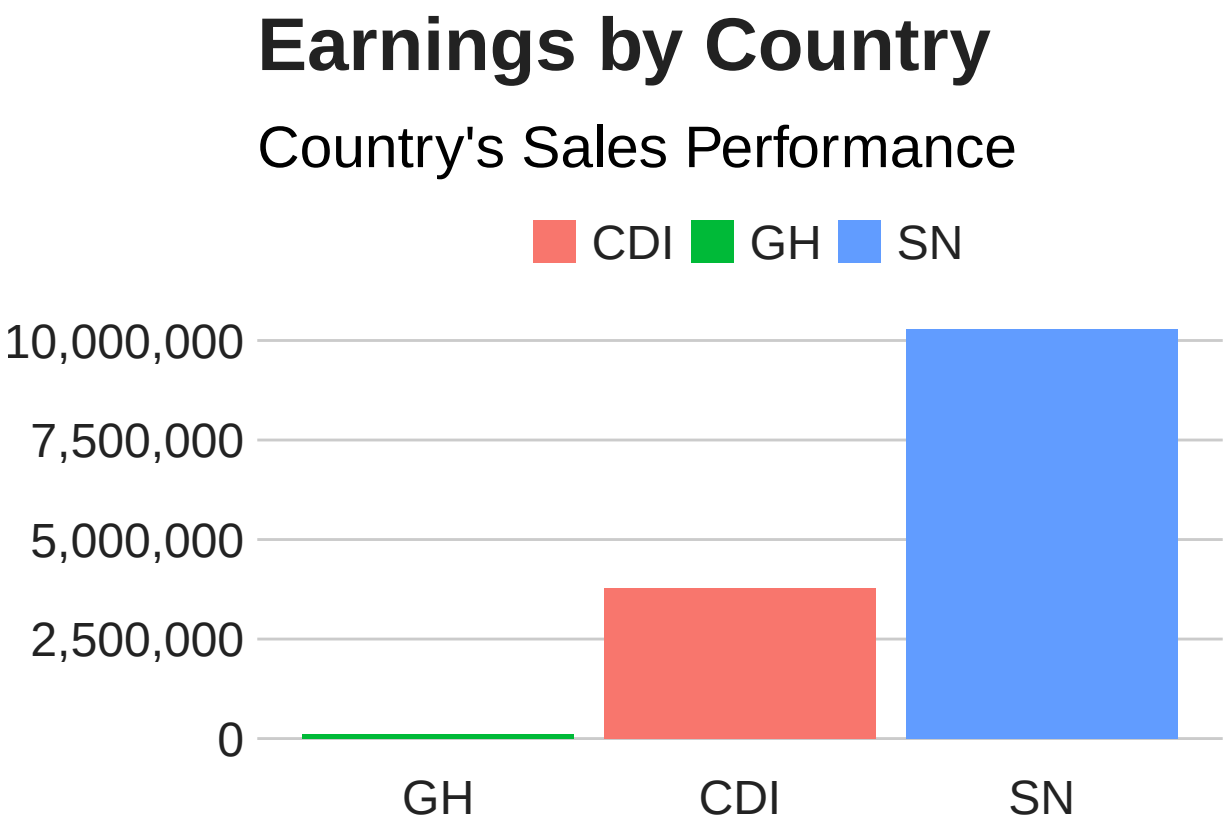
```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
db_3 %>%
  group_by(CountryId) %>%
  summarise(Tot_Amt_Received_to_Date_loc_Curr = sum(Sum_Paid_To_Date)) %>%
  arrange(desc(Tot_Amt_Received_to_Date_loc_Curr)) %>%
  ggplot(aes(reorder(x = CountryId, Tot_Amt_Received_to_Date_loc_Curr), y = Tot_Amt_Received_to_Date_lo
```

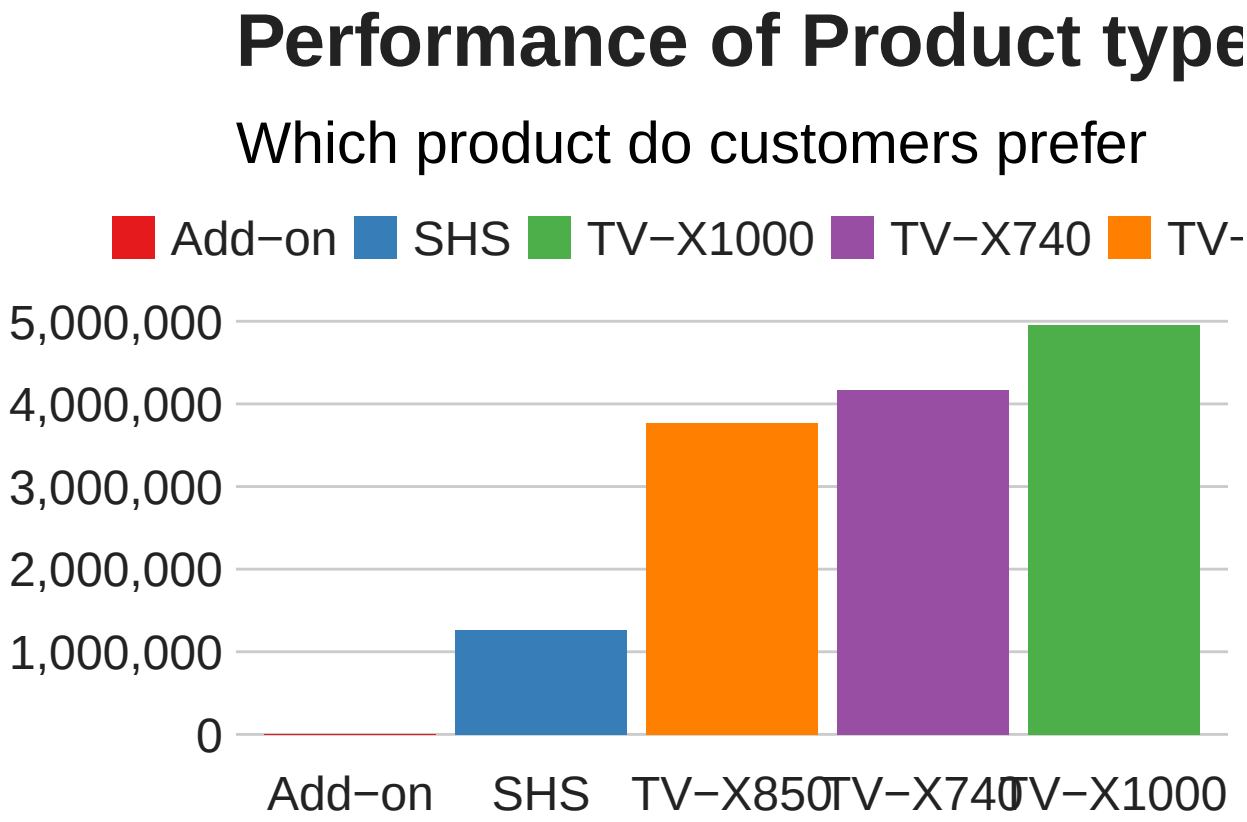
```
  geom_col() +
  scale_y_continuous(labels = comma)+
  labs(title = "Earnings by Country",
       subtitle = "Country's Sales Performance",
       x = "Earnings", y = "Country") +
  bbc_style()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



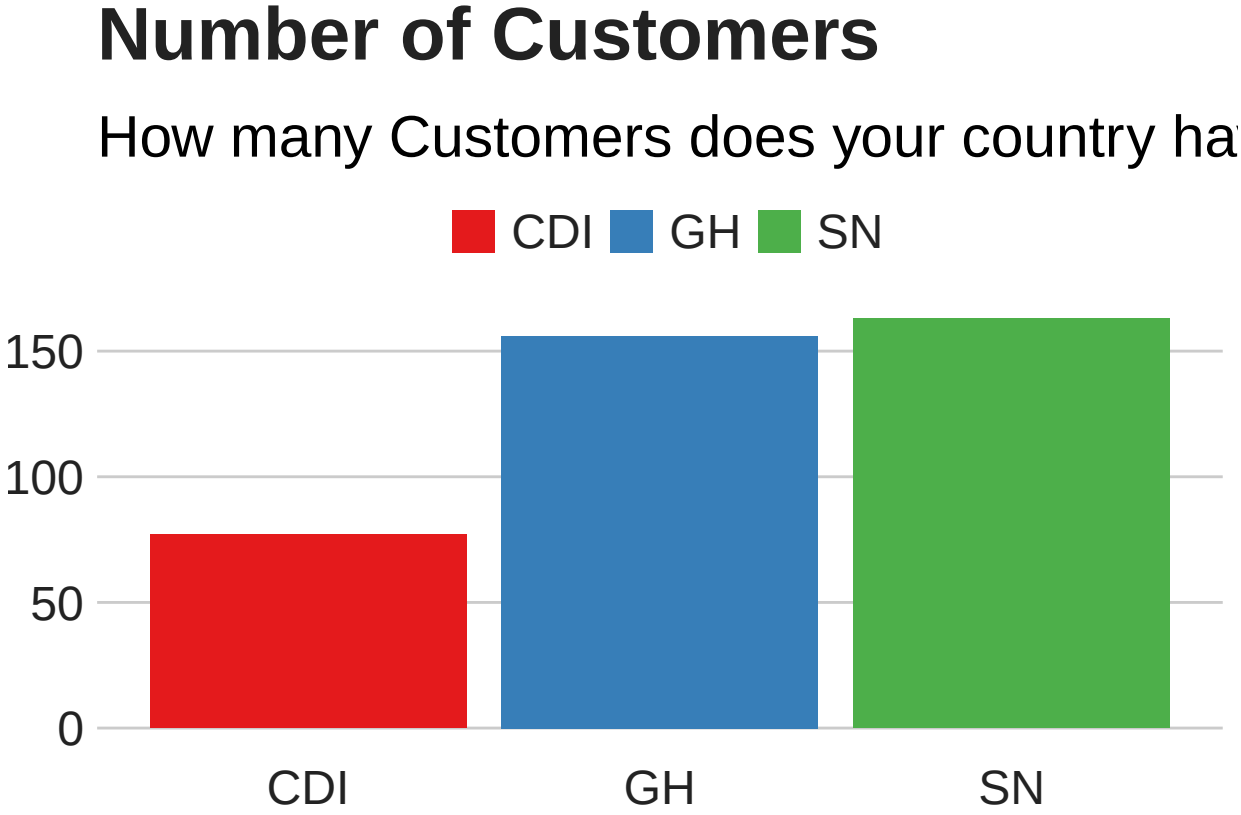

```
db_3 %>%
  group_by(productTypeGeneral) %>%
  summarise(sum_paid_per_product = sum(Sum_Paid_To_Date)) %>%
  arrange(desc(sum_paid_per_product)) %>%
  ggplot(aes(reorder(x = productTypeGeneral, sum_paid_per_product),
             y = sum_paid_per_product, fill = productTypeGeneral)) +
  geom_col() +
  scale_y_continuous(labels = comma) +
  scale_fill_brewer(palette = "Set1") +
  labs(title = "Performance of Product types",
       subtitle = "Which product do customers prefer", x = "Product",
       y = "Earnings by product") +
  bbc_style()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



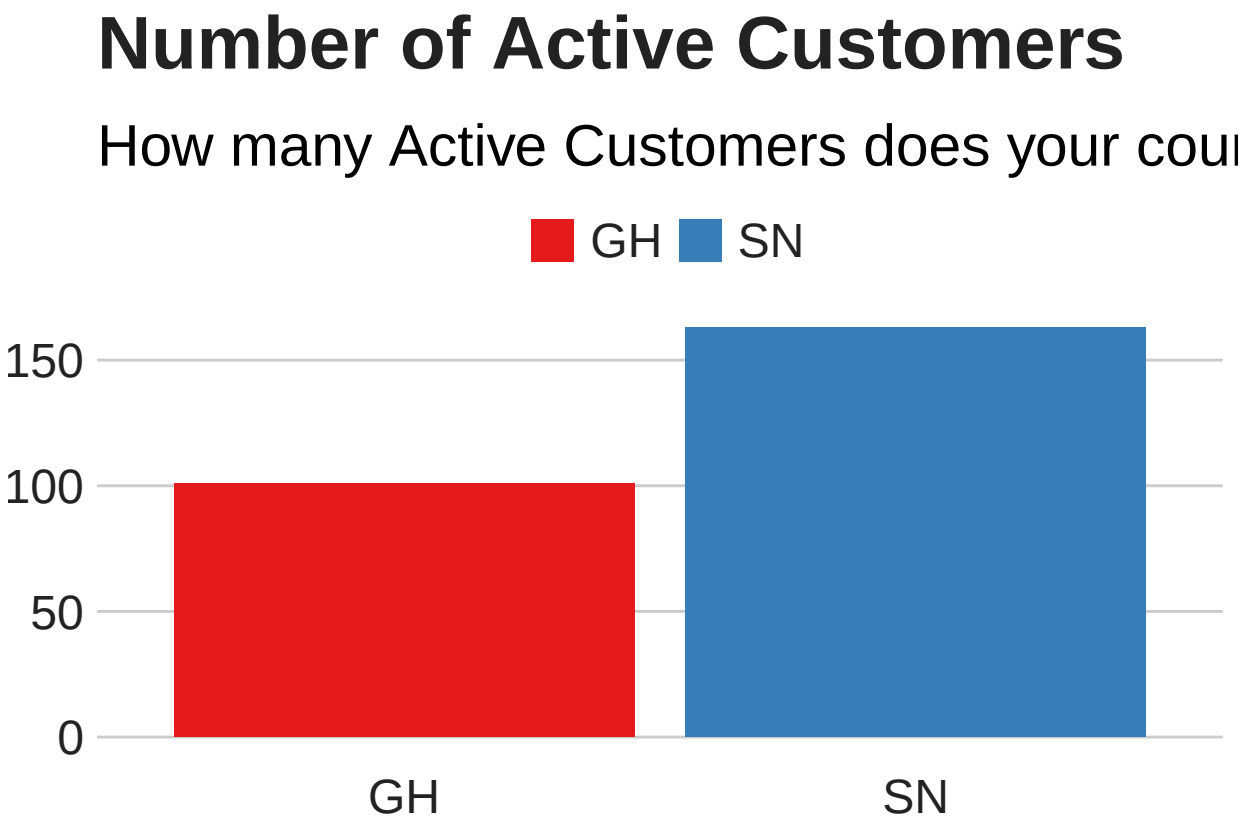

```
db_3 %>%
  group_by(CountryId) %>%
  summarise(No_of_Customers = n()) %>%
  ggplot(aes(x = CountryId, y = No_of_Customers, fill = CountryId)) +
  geom_col() +
  scale_fill_brewer(palette = "Set1") +
  labs(title = "Number of Customers", subtitle = "How many Customers does your country have?",
       x = "Country", y = "Number of Customer") +
  bbc_style()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
db_3 %>%
  filter(customerStatus == 1) %>%
  group_by(CountryId) %>%
  summarise(Number_of_Active_Cust = sum(customerStatus)) %>%
  ggplot(aes(x = CountryId, y = Number_of_Active_Cust, fill = CountryId)) +
  geom_col() +
  scale_fill_brewer(palette = "Set1") +
  labs(title = "Number of Active Customers", subtitle = "How many Active Customers does your country ha
       x = "Country", y = "Number of Customer") +
  bbc_style()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```


Number of Active Customers
How many Active Customers does your cou
GH
SN
150
100
50
0
GH
SN