VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

Project Title: ONLINE RETAIL DATA ANALYSIS

Course Name : BIG DATA ANALYTICS

Course Code : SWE2011

Slot : G1 +TG1

Faculty : SATHIYAMOORTHY E

Submitted By:

M.Naga Lakshmi-19MIS0358

B.Rohith – 19MIS0370

Review

Abstract:

This project involves the analysis of online retail data using a combination of R, tidyverse, sparklyr, and Spark. The goal of the project is to gain insights into customer behavior, sales trends, and product performance to inform business decisions and improve overall profitability. The data is first preprocessed and cleaned using R and the tidyverse, which provides a user-friendly and efficient approach to data manipulation and transformation. Then, sparklyr and Spark are used to handle big data and perform distributed computing, enabling the analysis of large datasets with ease. The project includes various data analysis techniques, such as data visualization, exploratory data analysis, and predictive modeling. The results of the analysis are presented in a clear and concise manner, providing valuable insights and actionable recommendations for the business. Overall, this project demonstrates the power of using R, tidyverse, sparklyr, and Spark in combination for online retail data analysis, highlighting the importance of leveraging big data tools and techniques for business success.

Introduction:

Introducing our latest project - online retail data analysis using R, tidyverse, sparklyr, and Spark. The world of online retail is rapidly growing, and it is becoming increasingly important for businesses to gain insights into customer behavior, sales trends, and product performance to stay ahead of the competition. Our project aims to provide these insights by analyzing online retail data using a powerful combination of R, tidyverse, sparklyr, and Spark.

The project begins by preprocessing and cleaning the data using R and the tidyverse, which is a popular set of packages for data manipulation and transformation. Once the data is cleaned, we use sparklyr and Spark to handle big data and perform distributed computing, enabling the analysis of large datasets with ease. This approach allows us to analyze vast amounts of online retail data in a short amount of time, and with minimal computational resources.

The analysis itself involves a range of techniques, including data visualization, exploratory data analysis, and predictive modeling. By leveraging these techniques, we gain valuable insights into customer behavior, sales trends, and product performance, which can inform business decisions and improve overall profitability.

The results of our analysis are presented in a clear and concise manner, with actionable recommendations for the business. Our project demonstrates the power of using R, tidyverse, sparklyr, and Spark in combination for online retail data analysis, highlighting the importance of leveraging big data tools and techniques for business success.

Dataset descriptions:

The dataset contains 541,909 records of online retail transactions, each uniquely identified by a 6-digit invoice number. The dataset includes information about each product sold, including its stock code and product name, as well as the quantity of each product sold and its unit price in sterling. Additionally, each transaction is associated with a specific date and time, as well as a customer ID and the name of the country where the customer resides.

It is important to note that if the invoice number begins with the letter 'c', it indicates a cancellation of the transaction. This information can be used to analyze customer behavior and trends in product returns.

Overall, this dataset provides a rich source of information for analyzing online retail sales, customer behavior, and trends in product performance. Its size and complexity make it an ideal candidate for big data analysis techniques such as distributed computing using tools like sparklyr and Spark.

Column	Description
InvoiceNo	Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter ‘c’, it indicates a cancellation.
StockCode	Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description	Product (item) name. Nominal.
Quantity	The quantities of each product (item) per transaction. Numeric.
InvoiceDate	Invice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice	Unit price. Numeric, Product price per unit in sterling.
CustomerID	Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country	Country name. Nominal, the name of the country where each customer resides.

Literature Survey:

1.Total retail goods consumption, industry structure, urban population growth and pollution intensity: an application of panel data analysis for China

There has been a growing concern regarding the regulation of environmental pollution in the face of a growing population, global warming, and climate change. Governments around the world have devised various mechanisms and policy strategies to ameliorate the worsening condition of natural environment around the world. Similar to the developed world, in China, the government is also aware of deteriorating environmental conditions. Hence, the existing abatement instruments include pollution discharge fees and several other policy strategies. This research is conducted to investigate the association between pollution intensity and its determinants, i.e., pollutant discharge fees and urban population, third industry structure, and total retail goods consumption. The secondary data of 29 provinces is used for empirical analysis. The principal component analysis is used to develop a single index called pollution intensity, and panel autoregressive distributed lags model (ARDL), or pooled mean group (PMG) analysis, is employed to find long-run and short-run relationship.

2.Impact of COVID-19 on the Customer End of Retail Supply Chains: A Big Data Analysis of Consumer Satisfaction

The COVID-19 pandemic has been one of the biggest disruptive events of recent decades and has had a global effect on society and the economy. The political regulations resulting from COVID-19 also led to significant changes in physical grocery shopping. However, the specific impact of COVID-19 on consumer satisfaction at the customer end of retail supply chains, i.e., the point-of-sale (PoS), has not yet been addressed. By gathering and analyzing consumer satisfaction data (ratings) and sentiments (evaluation comments) available on the open web, the current study evaluates the impact of COVID-19 on consumer satisfaction at the PoS. Focusing on the five biggest retail chains in Austria, the results show that there was a general and significant decline in consumer satisfaction due to the pandemic. The results also show a high impact of political regulations on consumer satisfaction. Furthermore, the text-mining based analysis of evaluation comments indicate that store layout and facilities, as well as product availability and waiting time had a great impact on consumer satisfaction. In total, over 533,000 consumer satisfaction ratings and over 153,000 textual comments have been analyzed, Future research could focus on applying the used data analysis technique and the adapted consumer sentiment dimensions in different settings, such as countries other than Austria or smaller retail chains

3.Exploratory space data analysis of spatial patterns of large-scale retail commercial facilities: The case of Gulou District, Nanjing, China

This study uses methods, such as a nearest proximity index, nuclear density, spatial interpolation, buffering zone, and overlay analysis, based on an exploratory spatial data analysis tool. It focuses on a large commercial facility in which a mathematical analysis is conducted on its spatial patterns. In the study, 45 large-scale retail commercial facilities (LSRCFs) in the Gulou District, Nanjing, China, were chosen, and the spatial concentration, density, and structure of the LSRCFs in this area were analyzed. Three additional factors, namely, population, transportation, and consumption, were examined to determine their impact on the spatial patterns of the LSRCFs. Finally, this study recommends a spatial layout for the future of the Gulou District according to the analysis results.

4.Consumer choice of store brands across store formats: A panel data analysis under crisis periods

This paper investigates the effect of marketing variables and consumer personal characteristics on store brand choice over national brands in times of crisis in France. It also seeks to clarify how store formats affect consumption strategies towards brands in turbulent times. Based on a large sample (panel of 4500 households, *N*=79,789), we used a binary logit model to assess consumer choice of store brands over national brands across two different store formats (hypermarket and supermarket). Results show that, overall, marketing variables and consumer characteristics affect significantly store brand choice over national brands. However, while crisis intensity clearly moderates the relationships between marketing policy variables and store brand choice, it does not affect so extensively the relationships between consumer characteristics and store brand choice over national brands. Furthermore, hypermarket and supermarket formats are not affected similarly by crisis. This research highlights the diversity of consumer strategies developed to cope with economic crises. Theoretical and managerial implications of these findings are discussed.

5.Revolution of Retail Industry: From Perspective of Retail 1.0 to 4.0

When Industry 4.0 was first introduced in 2010, it also brought the retail industry into the fourth revolution. Retail 4.0, on the other hand, appears to be a novel concept for retailers worldwide. When Industry 4.0 technologies such as the Artificial Intelligent (AI), Internet of Things (IoT), Cloud Computing, Big Data Analytical (BDA), and Augmented Reality (AR) were implemented in the retail industry, the term Retail 4.0 arose from Industry 4.0. This paper examines Retail 4.0 technologies and their application in the retail industry. The retail industry's revolution is also discussed in this paper. The final section examines the extent of implementation of retail 4.0

technology in various nations.

6.Challenges of Big Data analysis

Big Data bring new opportunities to modern society and challenges to data scientists. On the one hand, Big Data hold great promises for discovering subtle population patterns and heterogeneities that are not possible with small-scale data. On the other hand, the massive sample size and high dimensionality of Big Data introduce unique computational and statistical challenges, including scalability and storage bottleneck, noise accumulation, spurious correlation, incidental endogeneity and measurement errors. These challenges are distinguished and require new computational and statistical paradigm. This paper gives overviews on the salient features of Big Data and how these features impact on paradigm change on statistical and computational methods as well as computing architectures. We also provide various new perspectives on the Big Data analysis and computation. In particular, we emphasize on the viability of the sparsest solution in high-confidence set and point out that exogenous assumptions in most statistical methods for Big Data cannot be validated due to incidental endogeneity. They can lead to wrong statistical inferences and consequently wrong scientific conclusions.

7. IMPACT OF BIG DATA ON THE RETAIL INDUSTRY

With the recent emergence of Big Data with its Volume, Variety and Velocity (3V's), data analysis has emerged as a crucial area of study for both practitioners and researchers, reflecting the magnitude and impact of data-related problems to be resolved in business organizations, including the retail industry. This study has methodically identified and analysed four factors, namely, data source, data analysis tools, financial and economic outcomes and data security and data privacy, to gauge their influence on the impact of Big Data in the retail industry. This research analyses the impact of big data analysis on retail firms that use data and business analytics to make decisions, termed a data-driven decision-making (DDD) approach. The new finding is arrived that financial and economic outcome showed a strong support and have direct relationship with data analysis tools of retail industry. Data for the study were collected using a survey of various business practices and investments in information technology by retail organizations. The data analysis showed that retail organizations which use DDD have higher output and productivity. Using SMART PLS data analysis methods with solid support of review from ISI Journals, the relationship between DDD and performance is also evident in aspects of

organization such as the utilization of inventory, customer engagement and market value in the retail industry.

8.Data Analysis and Application of Retail Enterprises Based on Knime

With the rapid development of information technology, the application of cross-discipline has shown explosive growth. Data mining, big data and other technologies are rapidly entering all walks of life. Research on data mining technology and its application in data analysis of related enterprises have certain theoretical significance and practical value. This paper starts with a cross-border sales data set and uses Knime visual data analysis tool to illustrate that the widespread application of visual data mining technology in small and medium-sized enterprises is a feasible way to improve the decision-making ability of enterprises.

9. Retail Data Measurement Tools, Cognitive Artificial Intelligence Algorithms, and Metaverse Live Shopping Analytics in Immersive Hyper-Connected Virtual Spaces

Based on an in-depth survey of the literature, the purpose of the paper is to explore retail data measurement tools, cognitive artificial intelligence algorithms, and metaverse live shopping analytics in immersive hyper-connected virtual spaces. In this research, previous findings were cumulated showing that artificial intelligence chatbot customer service can boost customer engagement and hyper-realistic personalized interactive experiences by use of visual analytics, shopper behavioral data, and location data, and we contribute to the literature by indicating that digital inter- active performance in relation to virtual products and possessions shapes customer habits by harnessing location data. Throughout February 2022, a quantitative literature review of the Web of Science, Scopus, and ProQuest databases was performed, with search terms including “metaverse” + “live shopping analytics,” “retail data measurement tools,” “cognitive artificial intelligence algorithms,” and “immersive hyper-connected virtual spaces.” As research published in 2022 was inspected, only 87 articles satisfied the eligibility criteria. By taking out controversial or ambiguous findings (insufficient/irrelevant data), outcomes unsubstantiated by replication, too general material, or studies with nearly identical titles, we selected 18 mainly empirical sources. Data visualization tools: Dimensions (bibliometric mapping) and VOSviewer (layout algorithms). Reporting quality assessment tool: PRISMA. Methodological quality assessment tools include: AMSTAR, Dedoose, Distiller SR, and SRDR.

10.Big data for business management in the retail industry

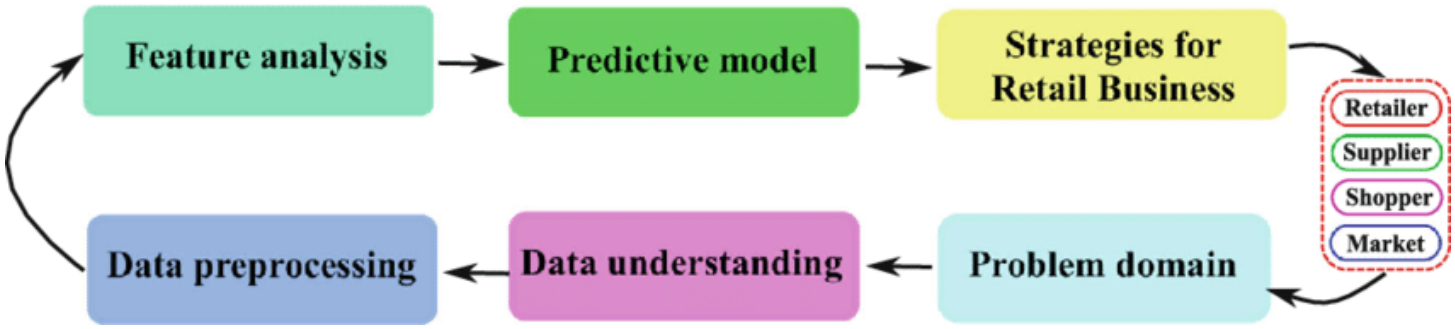
he purpose of this paper is to shed light on how big data deployment transforms organizational practices, thereby generating potential benefits, in a specific industry: retail. To achieve the paper's goal, the authors have conducted several semi-structured interviews with marketing managers of four retailers in Italy, and researched secondary data to get a broader picture of big data deployment in the organizations. Data analysis helped identify specific aspects related to big data deployment, data gathering methods, required competences and data sharing approaches.Despite the growing interest in big data in various fields of research, there are still few empirical studies on big data deployment in organizations in the management field, and even fewer on specific sectors. This research provides evidence of specific areas of analysis concerning big data in the retail industry.

S/W tools used:

Tools used for the development are as follows:

- ❖ R Studios
- ❖ R language
- ❖ Tidyverse, sparklyr and readxl

Proposed System Control Flow:



Implementation:

R-Code:

```
library(tidyverse)
library(sparklyr)
library(readxl)

Sys.setenv(JAVA_HOME="C:\\Program Files\\Java\\jdk1.8.0_202\\")
library("sparklyr")

sc <- spark_connect(master = "local",
                    spark_home = "C:\\Users\\msi\\AppData\\Local\\spark\\spark-2.3.1-bin-hadoop2.7",
                    version = "2.3.1",app_name = "retail_data_analysis")


# skip this step if it's been done in a previous analysis run
if(!file.exists("sales-transactions")){
  # read the dataset into R
  sales_transactions <- read_excel("C:/Users/msi/Downloads/r/Online Retail.xlsx")

  # basic check to see the data has been properly read
  head(sales_transactions)
  dim(sales_transactions)

  # pushing the data into Spark DataFrame
  sales_transactions_tbl <- copy_to(sc, sales_transactions, "sales_transactions", overwrite = TRUE)
  # mark the cancelled invoices, correct the country name, create InvoicePrefix indicator
  sales_transactions_tbl <- sales_transactions_tbl %>%
    mutate(InvoiceStatus = ifelse(InvoiceNo %in% c("581483", "541431", "556444"), "Cancelled", NA),
           Country = ifelse(Country == "EIRE", "Ireland", Country),
           InvoicePrefix = ifelse(substr(InvoiceNo,1,1) %in% letters | substr(InvoiceNo,1,1) %in% LETTERS, substr(InvoiceNo,1,1), NA))
  # check if the table is available on Spark
  src_tbls(sc)
  # save the Spark DataFrame to a Parquet file(s) for persistence across sessions (Spark applications) - to a local filesystem, current project Data/spark-warehouse
subdirectory
  spark_write_parquet(sales_transactions_tbl, str_c("file:///", getwd(), "/sales-transactions"),mode = "overwrite")
}
# read from Parquet into in-memory Spark DataFrame
sales_transactions_tbl <- spark_read_parquet(sc, "sales_transactions", str_c("file:///", getwd(), "/sales-transactions"), mode = "overwrite")


# read from Parquet into in-memory Spark DataFrame
sales_transactions_tbl <- spark_read_parquet(sc, "sales_transactions", str_c("file:///", getwd(), "/sales-transactions"), mode = "overwrite")


sales_transactions_tbl %>%
  sample_n(10)


sales_transactions_tbl %>%
  summarise(n_records = n(),
            n_invoices = n_distinct(InvoiceNo),
            n_missing_inv_rec = sum(as.integer(is.na(InvoiceNo)),na.rm = TRUE),
            n_customers = n_distinct(CustomerID),
            n_missing_cust_rec = sum(as.integer(is.na(CustomerID)),na.rm = TRUE))


sales_transactions_tbl %>%
  summarise(n_dist_stocks = n_distinct(StockCode),
            n_missing_stocks = sum(as.integer(is.na(StockCode)),na.rm = TRUE),
            n_dist_desc = n_distinct(Description),
            n_missing_desc = sum(as.integer(is.na(Description)),na.rm = TRUE),
            n_missing_quant = sum(as.integer(is.na(Quantity)),na.rm = TRUE),
            n_missing_prices = sum(as.integer(is.na(UnitPrice)),na.rm = TRUE))


# skip this step if it's been done in a previous analysis run
if(!file.exists("Data/spark-warehouse/invoices")){
  # create a temp table on Spark which holds the aggregation result
  invoices_tbl <- sales_transactions_tbl %>%
    group_by(InvoiceNo, InvoiceStatus, CustomerID, Country) %>%
    summarise(InvoiceDate = max(InvoiceDate, na.rm = TRUE),
              LineItems = n_distinct(StockCode),
              ItemQuantity = sum(Quantity, na.rm = TRUE),
              InvoiceAmount = sum(Quantity * UnitPrice, na.rm = TRUE)) %>%
    mutate(InvoicePrefix = ifelse(substr(InvoiceNo,1,1) %in% letters | substr(InvoiceNo,1,1) %in% LETTERS, substr(InvoiceNo,1,1), NA))
```

```
  # register the temp table in Spark
  sdf_register(invoices_tbl, "invoices")
  # save the Spark DataFrame to a Parquet file(s) for persistence across sessions (Spark applications) - to a local filesystem, current project Data/spark-warehouse
subdirectory
  spark_write_parquet(invoices_tbl,str_c("file:///", getwd(), "/invoices"),mode = "overwrite")
}


# read from Parquet into in-memory Spark DataFrame
invoices_tbl <- spark_read_parquet(sc, "invoices", str_c("file:///", getwd(), "/invoices"), mode = "overwrite")


invoices_tbl %>%
  summarise(n_rows = n(),
            n_distinct_invoices = n_distinct(InvoiceNo))


invoices_tbl %>%
  group_by(InvoiceNo) %>%
  summarise(n_rows = n()) %>%
  filter(n_rows > 1) %>%
  inner_join(invoices_tbl, by = "InvoiceNo") %>%
  arrange(desc(n_rows), InvoiceNo)


# calculate summary stats in Spark and pull the result into R
cancel_summary <- invoices_tbl %>%
  group_by(InvoicePrefix) %>%
  summarise(invoices = n(),
            amounts = sum(InvoiceAmount, na.rm = TRUE)) %>%
  mutate(InvoiceType = ifelse(is.na(InvoicePrefix), "Regular", ifelse(InvoicePrefix == "C", "Cancellation", InvoicePrefix))) %>%
  collect()

# use ggplot2 for data display - number of invoices by invoice type
ggplot(cancel_summary) +
  geom_bar(aes(x = InvoiceType, y = invoices), stat = "identity", position = "dodge", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Invoice type", y = "Number of invoices")


# invoice amounts by invoice type
ggplot(cancel_summary) +
  geom_bar(aes(x = InvoiceType, y = amounts), stat = "identity", position = "dodge", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Invoice type", y = "Invoice amounts (GBP)")


invoices_tbl %>%
  filter(InvoicePrefix == "C") %>%
  arrange(desc(InvoiceAmount)) %>%
  filter(rank(InvoiceAmount) <= 10)


invoices_tbl %>%
  filter(InvoicePrefix == "C") %>%
  select(InvoiceNo, InvoiceDate, InvoiceAmount) %>%
  transmute(CancelNo = InvoiceNo,
            CancelDate = InvoiceDate,
            CancelAmount = InvoiceAmount,
            InvoiceNo = substr(InvoiceNo, 2, 7)) %>%
  left_join(invoices_tbl, by = "InvoiceNo") %>%
  select(CancelNo, CancelDate, CancelAmount, InvoiceNo, InvoiceDate, InvoiceAmount) %>%
  arrange(CancelAmount) %>%
  filter(!is.na(InvoiceAmount))


invoices_tbl %>%
  filter(abs(InvoiceAmount) %in% c(168469.60, 77183.60, 38970.00, 22998.40, 17836.46, 16888.02, 16453.71)) %>%
  arrange(desc(abs(InvoiceAmount)))


invoices_tbl <- invoices_tbl %>%
  filter(is.na(InvoicePrefix) & is.na(InvoiceStatus))


invoices_tbl %>%
  mutate(SuspCustGroup = ifelse(is.na(CustomerID), "Suspected retail", "Suspected wholesale")) %>%
  group_by(LineItems, SuspCustGroup) %>%
  summarise(num_invoices = n(), invoice_amounts = sum(InvoiceAmount, na.rm = TRUE)) %>%
```

```
collect() %>%
  ggplot() +
  geom_bar(aes(x = LineItems, y = num_invoices), stat = "identity", fill = "#08306b") +
  coord_cartesian(xlim = c(0, 100), ylim = c(0, 2500)) +
  labs(x = "Line items on invoice",
       y = "Number of invoices",
       fill = "Customer group") +
  scale_fill_brewer(palette = "Set1") +
  facet_grid(SuspCustGroup ~ .)


invoices_tbl <- invoices_tbl %>%
  mutate(CustomerGroup = ifelse(is.na(CustomerID), "Retail", "Wholesale"))


cust_group_stats <- invoices_tbl %>%
  group_by(CustomerGroup) %>%
  summarise(n_invoices = n(),
            invoice_amount = sum(InvoiceAmount, na.rm = TRUE)) %>%
  collect()

ggplot(cust_group_stats) +
  geom_bar(aes(x = CustomerGroup, y = n_invoices), stat = "identity", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Customer group",
       y = "N of invoices",
       title = "Number of invoices per customer group") +
  theme(plot.title = element_text(hjust = 0.5))


ggplot(cust_group_stats) +
  geom_bar(aes(x = CustomerGroup, y = invoice_amount), stat = "identity", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Customer group",
       y = "Invoice amounts (GBP)",
       title = "Invoice amounts per customer group") +
  theme(plot.title = element_text(hjust = 0.5))




library(leaflet)
library(rworldmap)

# aggregate revenue by country
sales_by_country <- invoices_tbl %>%
  group_by(Country) %>%
  summarise(Amount = sum(InvoiceAmount, na.rm = TRUE)) %>%
  collect

# join the revenue by country to the world map of countries
sPDF <- joinCountryData2Map(sales_by_country
                            ,joinCode = "NAME"
                            ,nameJoinColumn = "Country", verbose = FALSE)


# select only the countries which generated revenue
existing_countries <- subset(sPDF, !is.na(Amount))

# create spending classes for revenues per country
bins <- c(0, 50000, 100000, 150000, 200000, 250000, 300000, Inf)
# assign a color to each of the classes
pal <- colorBin("YlOrRd", domain = existing_countries$Amount, bins = bins)

# create labels with actual revenue amounts per country, for hover info
labels <- paste0("<strong>", existing_countries$Country, "</strong><br/>",
                 format(existing_countries$Amount, digits = 1, big.mark = ".", decimal.mark = ",", scientific = FALSE),
                 " GBP") %>% lapply(htmltools::HTML)

# create the choropleth map
leaflet(existing_countries) %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addPolygons(
    fillColor = ~pal(Amount),
    weight = 1,
    opacity = 1,
    color = "white",
    dashArray = "3",
```

```
fillOpacity = 0.7,
    highlight = highlightOptions(
      weight = 2,
      color = "#666",
      dashArray = "",
      fillOpacity = 0.7,
      bringToFront = TRUE),
    label = labels,
    labelOptions = labelOptions(
      style = list("font-weight" = "normal", padding = "3px 8px"),
      textsize = "15px",
      direction = "auto")) %>%
  addLegend(pal = pal, values = ~Amount, opacity = 0.7, title = NULL,
            position = "topright") %>%
  setView(17,34,2)



library(ggplot2)


invoices_tbl %>%
  filter(CustomerGroup == "Wholesale") %>%
  group_by(CustomerID) %>%
  summarise(NumInvoices = n()) %>%
  ungroup() %>%
  group_by(NumInvoices) %>%
  summarise(NumCustomers = n()) %>%
  arrange(NumInvoices) %>%
  collect() %>%
  ggplot() +
  geom_freqpoly(aes(x = NumInvoices, y = NumCustomers), stat = "identity", colour = "#08306b") +
  labs(x = "Number of purchases",
       y = "Number of customers",
       title = "Wholesale customers by number of purchases") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  theme(plot.title = element_text(hjust = 0.5))


invoices_tbl %>%
  filter(CustomerGroup == "Wholesale") %>%
  group_by(CustomerID) %>%
  summarise(NumInvoices = as.double(n())) %>%
  ungroup() %>%
  ft_bucketizer(input_col = "NumInvoices", output_col = "NumInvoicesDisc", splits = c(1,2,3,4,5,6,11,16,21,Inf)) %>%
  group_by(NumInvoicesDisc) %>%
  summarise(NumCustomers = n()) %>%
  arrange(NumInvoicesDisc) %>%
  collect() %>%
  mutate(NumInvoicesDisc = ordered(NumInvoicesDisc,
                                   levels = c(0,1,2,3,4,5,6,7,8),
                                   labels = c("1","2","3","4","5","6-10","11-15","16-20","21+")),
         Group = "Purchases") %>%
  ggplot() +
  geom_bar(aes(x = Group, fill = reorder(NumInvoicesDisc, desc(NumInvoicesDisc)), y = NumCustomers), stat = "identity", position = "stack") +
  labs(x = "",
       y = "Number of customers",
       title = "Wholesale customers by number of purchases",
       fill = "Number of purchases") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette = "Blues")



invoices_tbl %>%
  filter(CustomerGroup == "Wholesale") %>%
  group_by(CustomerID) %>%
  summarise(NumInvoices = n(),
            AmountSpent = sum(InvoiceAmount, na.rm = TRUE),
            FirstPurchase = min(InvoiceDate, na.rm = TRUE),
            AvgPurchaseValue = sum(InvoiceAmount, na.rm = TRUE) / n()) %>%
  ungroup() %>%
  filter(NumInvoices >= 4 & rank(desc(AmountSpent)) <= 10)
```

```
sales_transactions_tbl <- sales_transactions_tbl %>%
  filter(is.na(InvoicePrefix) & is.na(InvoiceStatus))


sales_transactions_tbl %>%
  summarise(n_dist_products = n_distinct(StockCode))


multiple_descriptions <- sales_transactions_tbl %>%
  group_by(StockCode) %>%
  summarise(n_desc = n_distinct(Description)) %>%
  filter(n_desc > 1)

sales_transactions_tbl %>%
  inner_join(multiple_descriptions, by = "StockCode") %>%
  select(n_desc, StockCode, Description) %>%
  group_by(n_desc, StockCode, Description) %>%
  summarise(n_rows = n()) %>%
  arrange(desc(n_desc), StockCode, desc(n_rows)) %>%
  head(100)


products_tbl <- sales_transactions_tbl %>%
  group_by(StockCode, Description) %>%
  summarise(n_rows = n()) %>%
  filter(rank(desc(n_rows)) == 1) %>%
  select(StockCode, Description)

# register the temp table in Spark
sdf_register(products_tbl, "products")


# save the Spark DataFrame to a Parquet file(s) for persistence across sessions (Spark applications) - to a local filesystem, current project Data/spark-warehouse
subdirectory
spark_write_parquet(products_tbl, str_c("file:///", getwd(), "/products"), mode = "overwrite")


sales_transactions_tbl <- sales_transactions_tbl %>%
  select(-Description) %>%
  inner_join(products_tbl, by = "StockCode")


products_tbl %>%
  arrange(StockCode) %>%
  head(100)


products_tbl %>%
  filter(substr(StockCode, 1, 5) %in% c("15056", "16161")) %>%
  arrange(StockCode)


products_tbl %>%
  filter(StockCode %in% c("16014", "16015", "16016"))


products_tbl %>%
  mutate(ProductType = ifelse(substr(StockCode, -1, 1) %in% letters | substr(StockCode, -1, 1) %in% LETTERS, "Variation", "Regular")) %>%
  group_by(ProductType) %>%
  summarise(n_rows = n()) %>%
  collect %>%
  ggplot() +
  geom_bar(aes(x = ProductType, y = n_rows), stat = "identity", position = "dodge", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Product type", y = "Number of products")


sales_transactions_tbl %>%
  mutate(ProductType = ifelse(substr(StockCode, -1, 1) %in% letters | substr(StockCode, -1, 1) %in% LETTERS, "Variation", "Regular")) %>%
  group_by(ProductType) %>%
  summarise(SalesAmount = sum(Quantity * UnitPrice, na.rm = TRUE)) %>%
  collect %>%
  ggplot() +
  geom_bar(aes(x = ProductType, y = SalesAmount), stat = "identity", position = "dodge", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Product type", y = "Sales amount (GBP)")
```

```
sales_transactions_tbl %>%
  group_by(StockCode, Description) %>%
  summarise(SalesAmount = sum(Quantity * UnitPrice, na.rm = TRUE)) %>%
  filter(rank(desc(SalesAmount)) <= 100) %>%
  arrange(desc(SalesAmount)) %>%
  head(100)


sales_transactions_tbl <- sales_transactions_tbl %>%
  filter(!StockCode %in% c("DOT", "POST", "M", "AMAZONFEE"))

# I'm looking at average price, because the price of a product can change during the year.
sales_transactions_tbl %>%
  group_by(StockCode, Description) %>%
  summarise(SalesAmount = sum(Quantity * UnitPrice, na.rm = TRUE),
            SalesQuantity = sum(Quantity, na.rm = TRUE),
            AvgPrice = mean(UnitPrice, na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(SalesAmtPercOfTotal = round(SalesAmount / sum(SalesAmount, na.rm = TRUE) * 100, 2)) %>%
  arrange(desc(SalesAmount)) %>%
  mutate(CumRevenue = cumsum(SalesAmount),
         CumRevPerc = cumsum(SalesAmtPercOfTotal)) %>%
  filter(rank(desc(SalesAmount)) <= 100)


top_10_products <- sales_transactions_tbl %>%
  mutate(CustomerGroup = ifelse(is.na(CustomerID), "Retail", "Wholesale")) %>%
  group_by(Description, CustomerGroup) %>%
  summarise(SalesAmount = sum(Quantity * UnitPrice, na.rm = TRUE)) %>%
  ungroup() %>%
  group_by(CustomerGroup) %>%
  filter(rank(desc(SalesAmount)) <= 10) %>%
  collect

top_10_products %>%
  filter(CustomerGroup == "Wholesale") %>%
  ggplot() +
  geom_bar(aes(x = reorder(factor(Description), SalesAmount), y = SalesAmount), stat = "identity", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Product",
       y = "Sales amount (GBP)",
       title = "Top 10 sold products - wholesale") +
  coord_flip()


top_10_products %>%
  filter(CustomerGroup == "Retail") %>%
  ggplot() +
  geom_bar(aes(x = reorder(factor(Description), SalesAmount), y = SalesAmount), stat = "identity", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Product",
       y = "Sales amount (GBP)",
       title = "Top 10 sold products - retail") +
  coord_flip()


itemsets_tbl <- sales_transactions_tbl %>%
  select(InvoiceNo, Description) %>%
  distinct() %>%
  group_by(InvoiceNo) %>%
  summarise(items = collect_list(Description))

# let's look at how the prepared data looks like - one row, one invoice, and a list of its line items
head(itemsets_tbl)


# run the FPGrowth
fp_model <- ml_fpgrowth(itemsets_tbl, min_confidence = 0.5, min_support = 0.025)

# extract the derived frequent itemsets and reformat the data
freq_itemsets <- ml_freq_itemsets(fp_model) %>%
  collect %>%
  mutate(list_length = map_int(items, length)) %>%
  filter(list_length > 1) %>%
  arrange(desc(freq)) %>%
  mutate(itemset = map_chr(items, str_c, sep = "-", collapse = "-")) %>%
  select(-items, -list_length)

# display the itemsets
freq_itemsets
```

```
library(wordcloud)

wordcloud(freq_itemsets$itemset, freq_itemsets$freq, max.words = 20, scale=c(0.1, 3.0),rot.per = 0,
          colors=brewer.pal(8, "Dark2"), random.order = FALSE, random.color = FALSE, fixed.asp = FALSE)


library(networkD3)

# extract association rules
assoc_rules <- ml_association_rules(fp_model) %>%
  collect %>%
  mutate(antecedent = map_chr(antecedent, str_c, sep = " + ", collapse = " + ")) %>%
  mutate(consequent = map_chr(consequent, str_c, sep = " + ", collapse = " + "))

# create a list of distinct antecedents
ante <- assoc_rules %>%
  distinct(antecedent) %>%
  transmute(name = antecedent)

# create a list of distinct consequents, combine them with distinct antecedents to create a list of network nodes
# add a unique id to every node
nodes <- assoc_rules %>%
  distinct(consequent) %>%
  transmute(name = consequent) %>%
  bind_rows(ante) %>%
  distinct() %>%
  mutate(group = "1") %>%
  mutate(row_id = seq(from = 0, length.out = length(name)), size = 20)

# extract directed link information from association rules, and add corresponding node IDs
links <- assoc_rules %>%
  left_join(nodes, by = c("antecedent" = "name")) %>%
  mutate(antecedent_row_id = row_id) %>%
  select(-row_id) %>%
  left_join(nodes, by = c("consequent" = "name")) %>%
  mutate(consequent_row_id = row_id) %>%
  select(-row_id,-group.x,-group.y)

# create the network visual using the nodes and links
forceNetwork(Links = as.data.frame(links), Nodes = as.data.frame(nodes), Source = "antecedent_row_id",
             Target = "consequent_row_id", Value = "confidence", NodeID = "name",
             Group = "group", opacity = 0.9, arrows = TRUE, linkWidth = JS("function(d) { return d.value * 4; }"),
             Nodesize = "size", fontSize = 15, fontFamily = "arial", linkDistance = 100, charge = -30, bounded = TRUE,
             opacityNoHover = 0.5)


ratings_tbl <- sales_transactions_tbl %>%
  filter(!is.na(CustomerID)) %>%
  select(CustomerID, StockCode, Description, Quantity) %>%
  group_by(CustomerID, StockCode, Description) %>%
  summarise(Quantity = sum(Quantity, na.rm = TRUE)) %>%
  ungroup() %>%
  mutate(CustomerID = as.integer(CustomerID),
         StockID = as.integer(rank(StockCode)))

# let's look at how the prepared data looks like - one row, one item, and a quantity transformed into 1-10 rating
head(ratings_tbl, 250)


# How many ratings do I have?
ratings_tbl %>%
  summarise(row_count = n())


# create a stock ID and names table - to join it back to results
product_names_tbl <- ratings_tbl %>%
  select(StockID, StockCode, Description) %>%
  distinct()


# train the ALS. I'll set the regularization parameter to 0.1, set implicit preference to true to indicate to ALS that ratings are actually derived from other information, and set the cold start to drop, to get only the results where the recommender returns a recommendation.
als_model <- ml_als(ratings_tbl, rating_col = "Quantity", user_col = "CustomerID",
                    item_col = "StockID", reg_param = 0.1,
                    implicit_prefs = TRUE, alpha = 1, nonnegative = FALSE,
                    max_iter = 10, num_user_blocks = 10, num_item_blocks = 10,
                    checkpoint_interval = 10, cold_start_strategy = "drop")
```

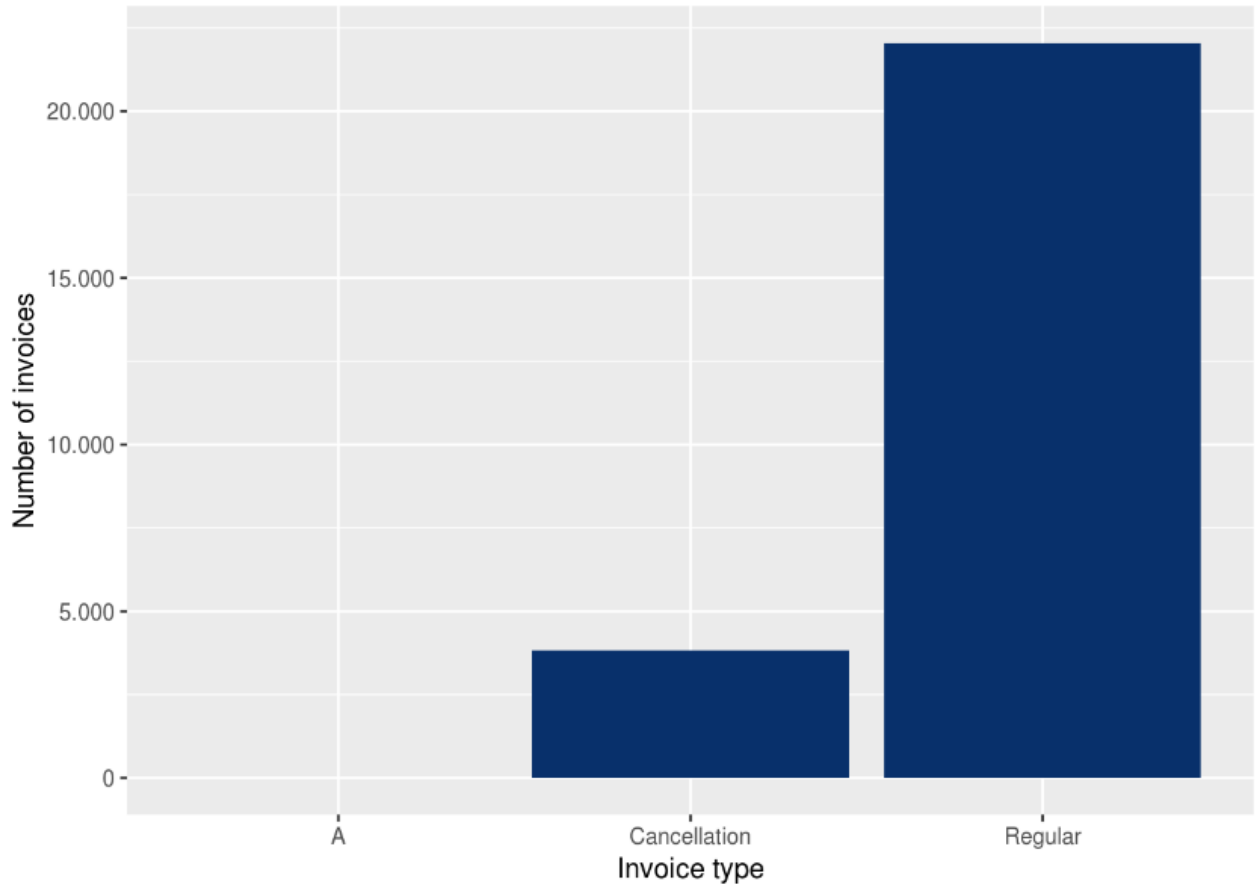
```
top_5_recommended_products <- ml_recommend(als_model, type = "items", 5) %>%
  inner_join(product_names_tbl, by = "StockID") %>%
  select(-recommendations) %>%
  arrange(CustomerID, desc(rating))

top_5_recommended_products %>%
  head(100)


ratings_tbl %>%
  filter(CustomerID == 12353) %>%
  arrange(CustomerID, desc(Quantity)) %>%
  select(CustomerID, StockCode, Description, Quantity)


top_5_recommended_products %>%
  filter(CustomerID == 12353) %>%
  arrange(CustomerID, desc(rating)) %>%
  select(CustomerID, StockCode, Description, rating)


ratings_tbl %>%
  filter(CustomerID == 12361) %>%
  arrange(CustomerID, desc(Quantity)) %>%
  select(CustomerID, StockCode, Description, Quantity)


top_5_recommended_products %>%
  filter(CustomerID == 12361) %>%
  arrange(CustomerID, desc(rating)) %>%
  select(CustomerID, StockCode, Description, rating)


ratings_tbl %>%
  filter(CustomerID == 12367) %>%
  arrange(CustomerID, desc(Quantity)) %>%
  select(CustomerID, StockCode, Description, Quantity)


top_5_recommended_products %>%
  filter(CustomerID == 12367) %>%
  arrange(CustomerID, desc(rating)) %>%
  select(CustomerID, StockCode, Description, rating)


ratings_tbl %>%
  filter(CustomerID == 12401) %>%
  arrange(CustomerID, desc(Quantity)) %>%
  select(CustomerID, StockCode, Description, Quantity)


top_5_recommended_products %>%
  filter(CustomerID == 12401) %>%
  arrange(CustomerID, desc(rating)) %>%
  select(CustomerID, StockCode, Description, rating)


ratings_tbl %>%
  filter(CustomerID == 12441) %>%
  arrange(CustomerID, desc(Quantity)) %>%
  select(CustomerID, StockCode, Description, Quantity)


top_5_recommended_products %>%
  filter(CustomerID == 12441) %>%
  arrange(CustomerID, desc(rating)) %>%
  select(CustomerID, StockCode, Description, rating)


spark_disconnect(sc)
```

Output of Graphs and tables:

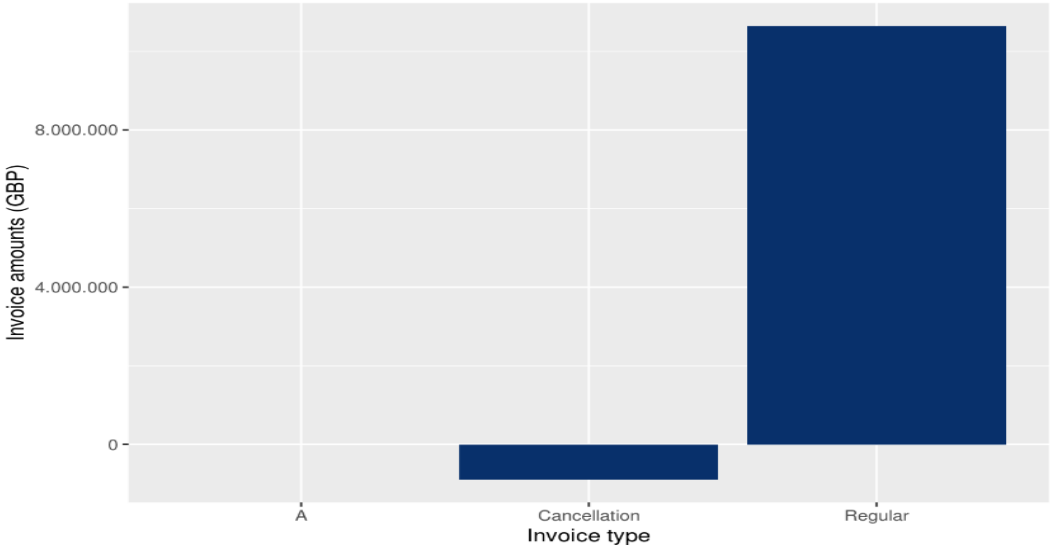
```
sales_transactions_tbl %>%
   sample_n(10)
```

InvoiceNo <chr>	StockCode <chr>	Description <chr>	Quantity <dbl>	InvoiceDate <S3: POSIXct>	UnitPrice <dbl>
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55
536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25
536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85
536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69

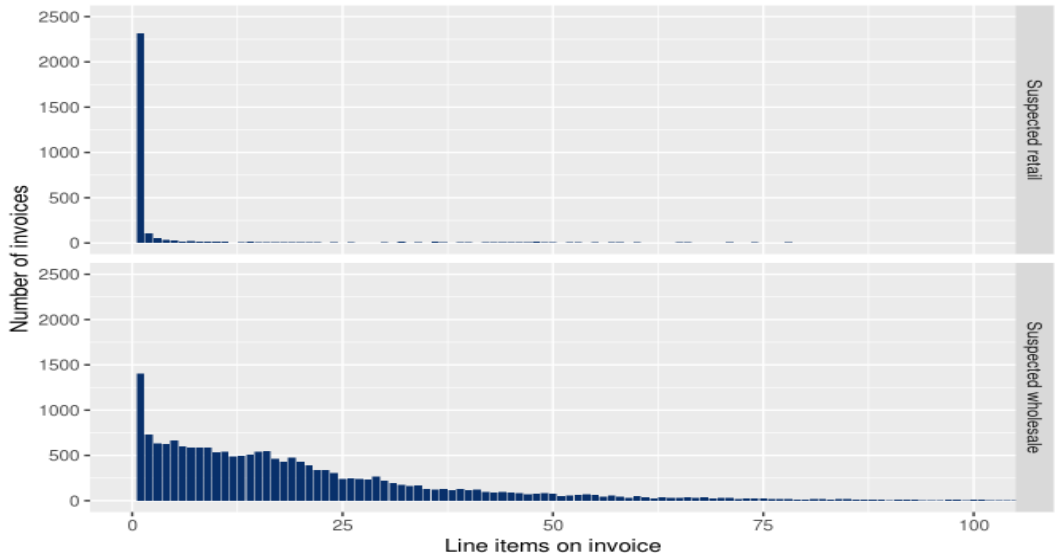
1-10 of 10 rows | 1-6 of 10 columns

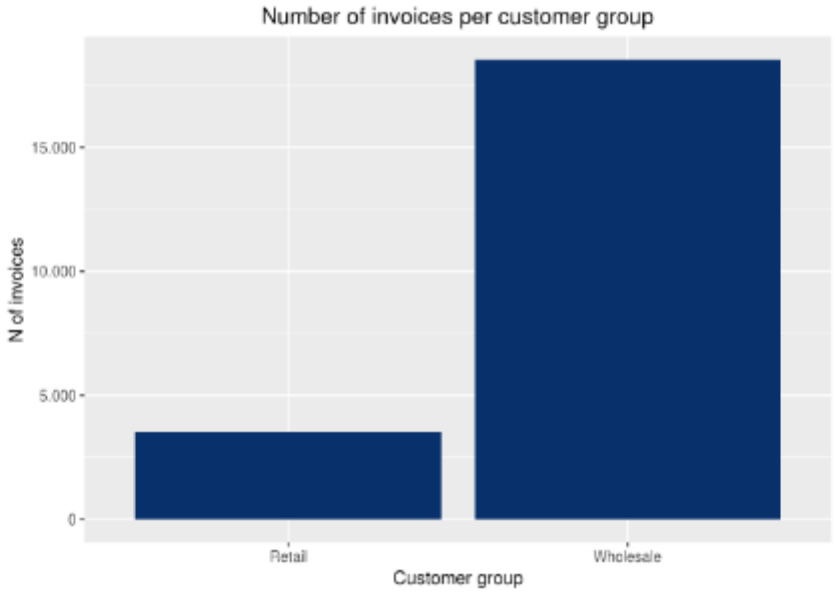


```r
# invoice amounts by invoice type
ggplot(cancel_summary) +
  geom_bar(aes(x = InvoiceType, y = amounts), stat = "identity", position = "dodge", fill = "#08306b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Invoice type", y = "Invoice amounts (GBP)")
```



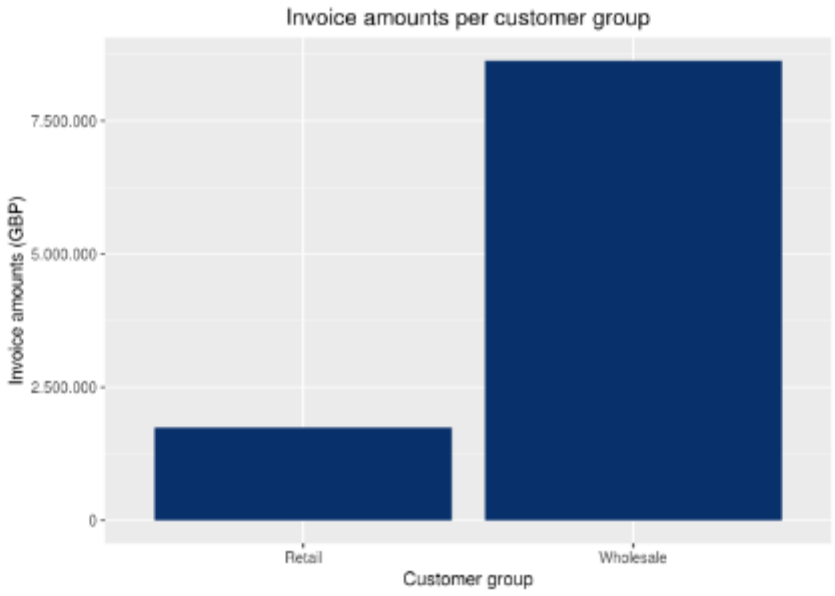

```r
invoices_tbl %>%
  mutate(SuspCustGroup = ifelse(is.na(CustomerID), "Suspected retail", "Suspected wholesale")) %>%
  group_by(LineItems, SuspCustGroup) %>%
  summarise(num_invoices = n(), invoice_amounts = sum(InvoiceAmount, na.rm = TRUE)) %>%
  collect() %>%
  ggplot() +
  geom_bar(aes(x = LineItems, y = num_invoices), stat = "identity", fill = "#08306b") +
    coord_cartesian(xlim = c(0, 100), ylim = c(0, 2500)) +
    labs(x = "Line items on invoice",
         y = "Number of invoices",
         fill = "Customer group") +
    scale_fill_brewer(palette = "Set1") +
  facet_grid(SuspCustGroup ~ .)
```

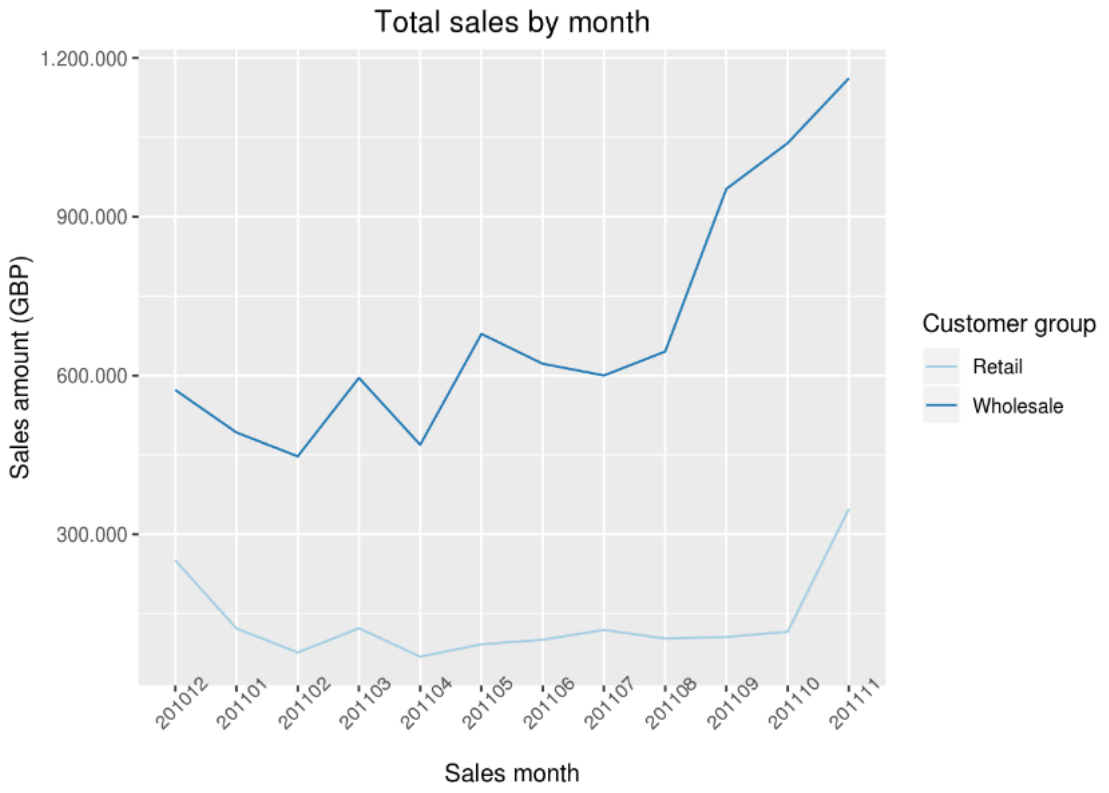


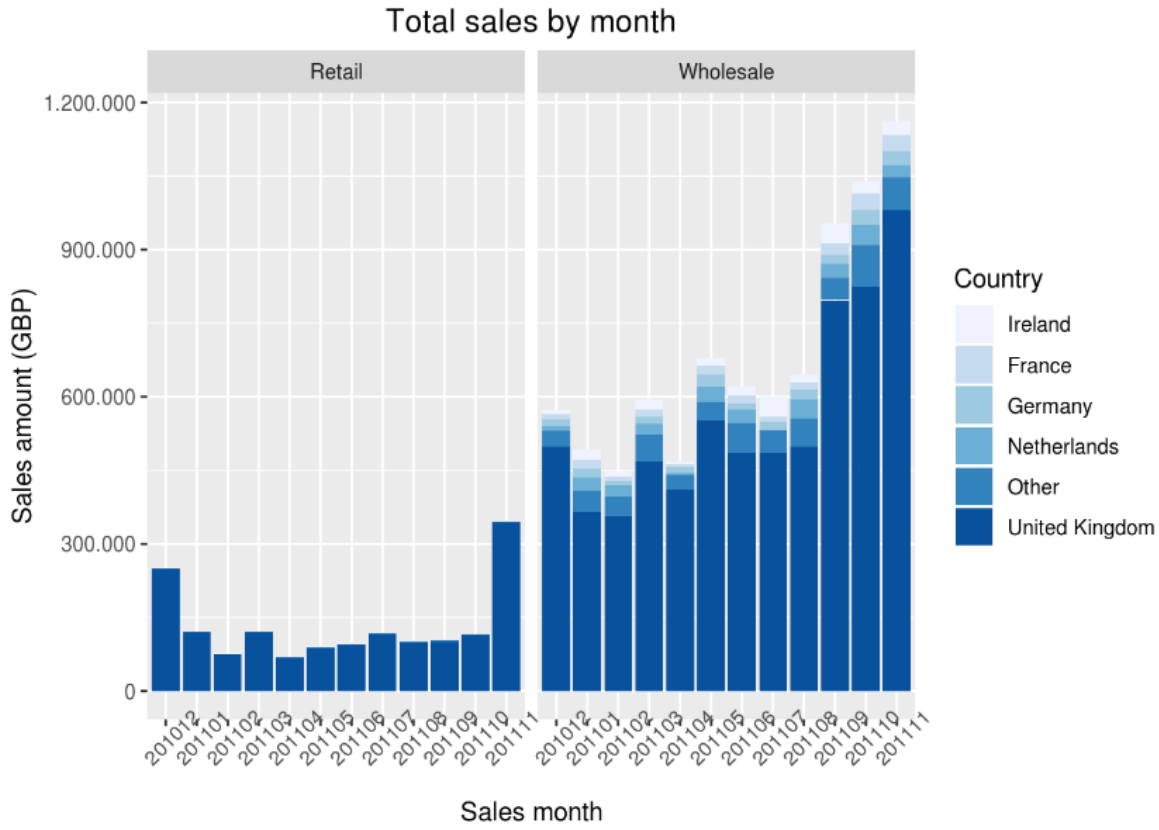


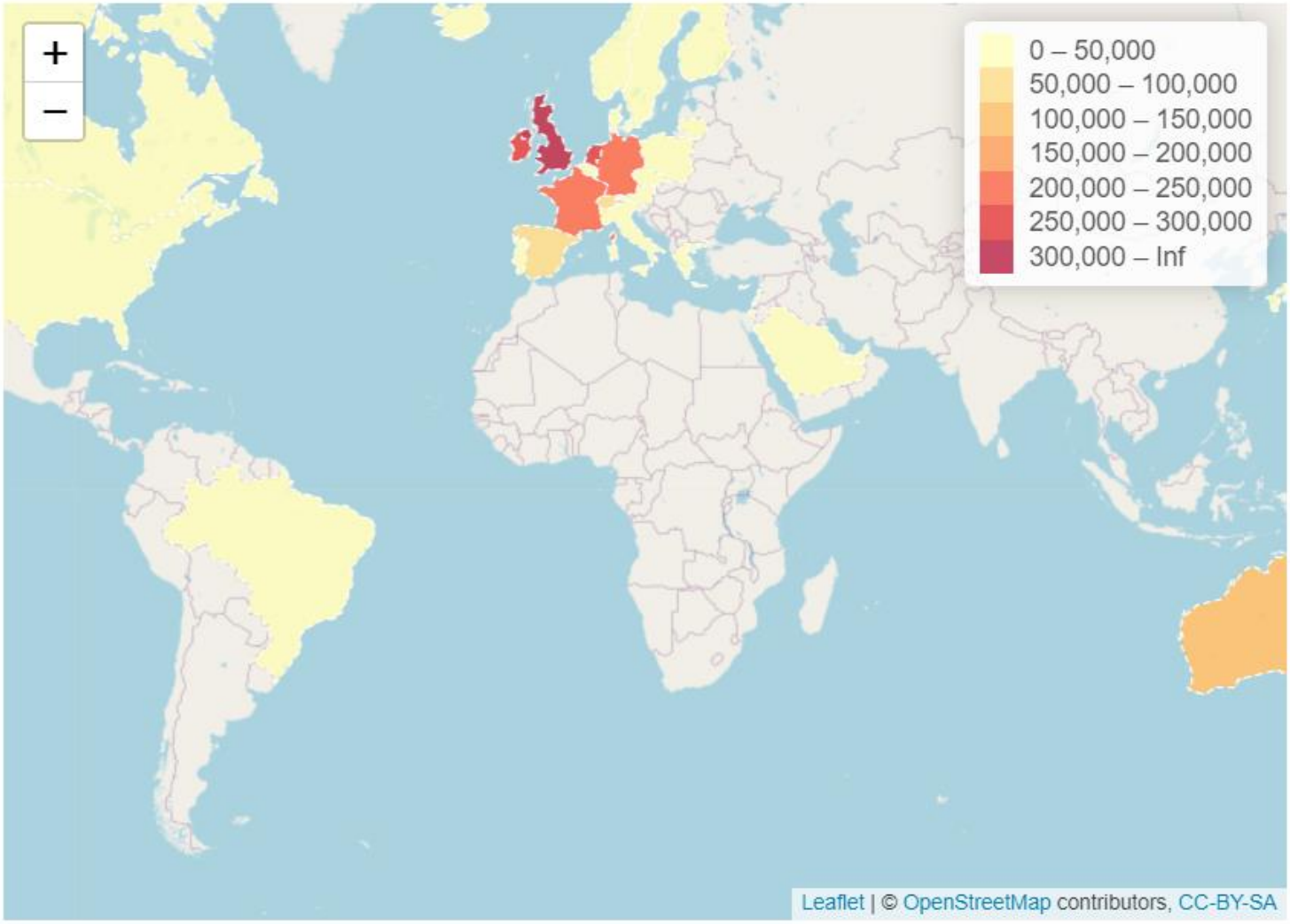
Hide

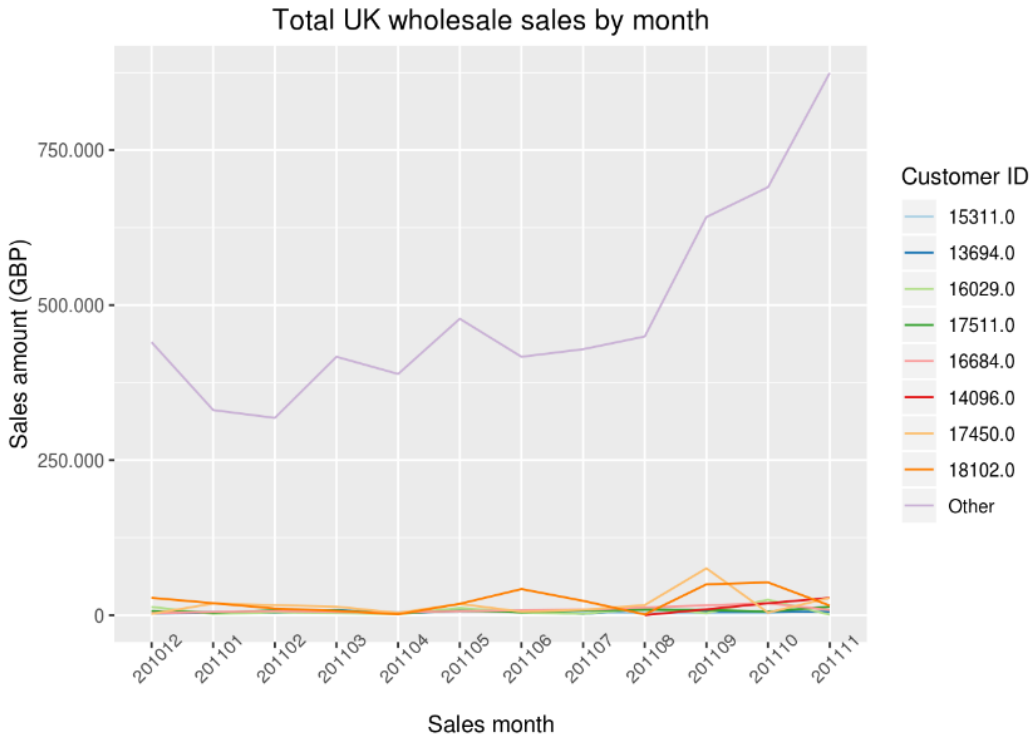
```
ggplot(cust_group_stats) +
  geom_bar(aes(x = CustomerGroup, y = invoice_amount), stat = "identity", fill = "#083066") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Customer group",
       y = "Invoice amounts (GBP)",
       title = "Invoice amounts per customer group") +
  theme(plot.title = element_text(hjust = 0.5))
```

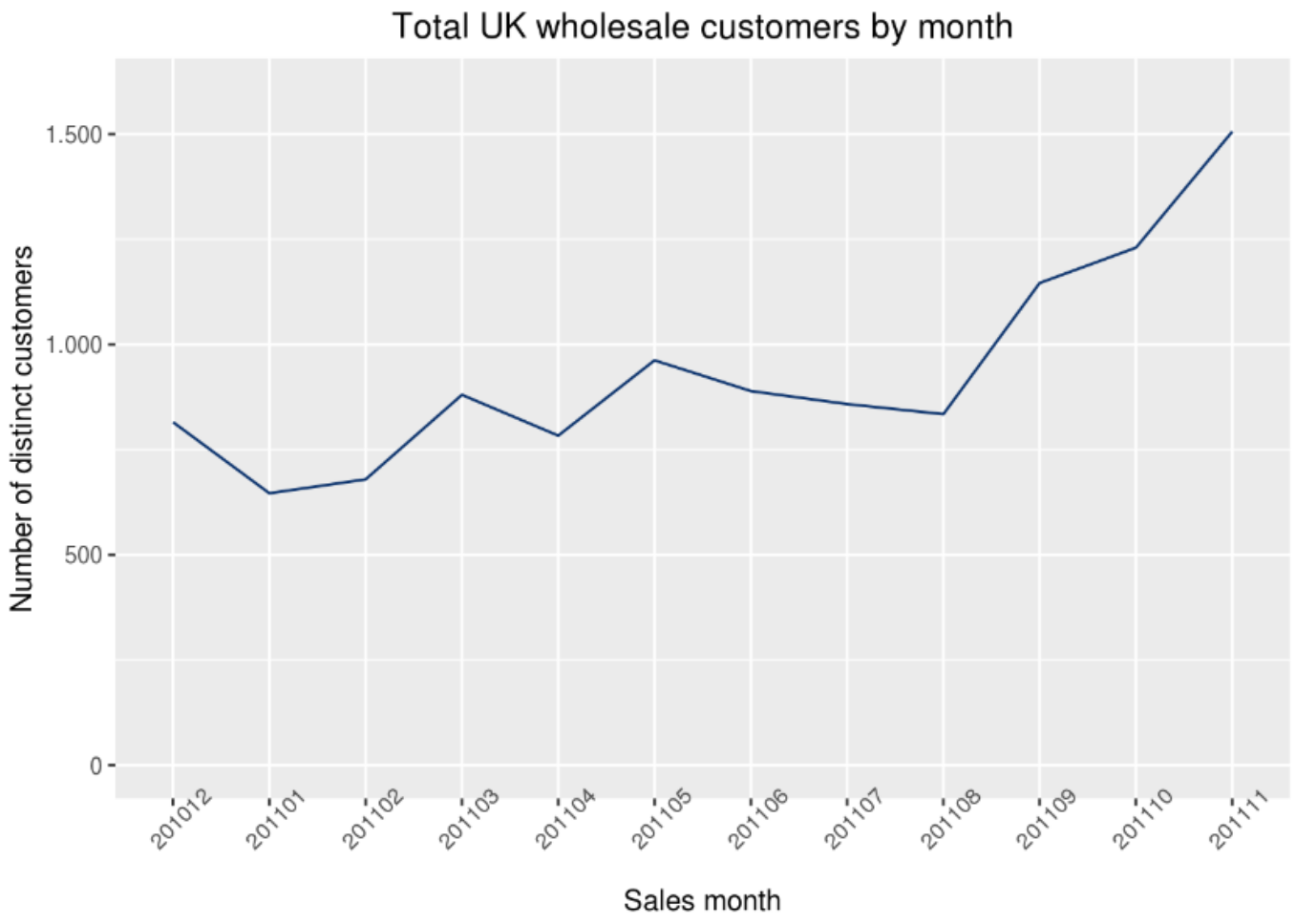


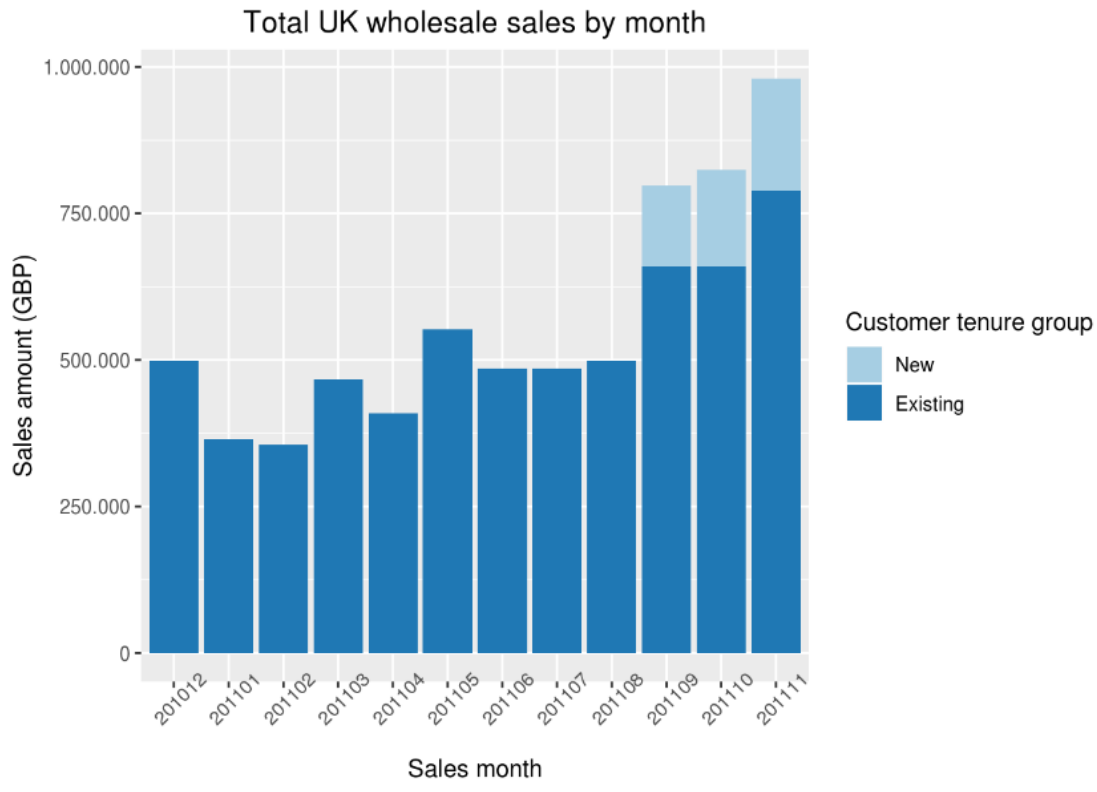

Total sales by month
1.200.000
900.000
600.000
300.000
Sales amount (GBP)
201012
201101
201102
201103
201104
201105
201106
201107
201108
201109
201110
201111
Sales month
Customer group
Retail
Wholesale

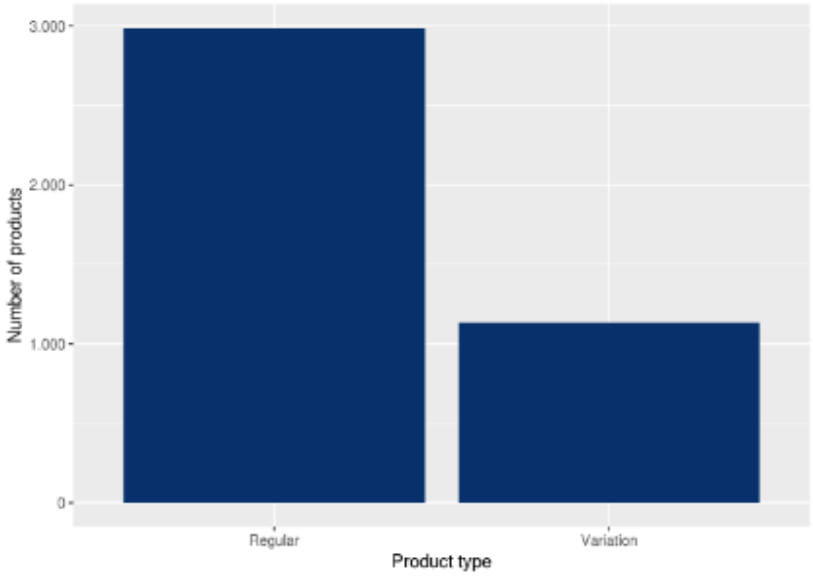

Total sales by month
Retail
Wholesale
1.200.000
900.000
600.000
300.000
0
Sales amount (GBP)
201012
201101
201102
201103
201104
201105
201106
201107
201108
201109
201110
201111
Sales month
Country
Ireland
France
Germany
Netherlands
Other
United Kingdom


+
−
0 – 50,000
50,000 – 100,000
100,000 – 150,000
150,000 – 200,000
200,000 – 250,000
250,000 – 300,000
300,000 – Inf
Leaflet | © OpenStreetMap contributors, CC-BY-SA


Total UK wholesale sales by month
Sales amount (GBP)
750.000
500.000
250.000
0
201012
201101
201102
201103
201104
201105
201106
201107
201108
201109
201110
201111
Sales month
Customer ID
15311.0
13694.0
16029.0
17511.0
16684.0
14096.0
17450.0
18102.0
Other

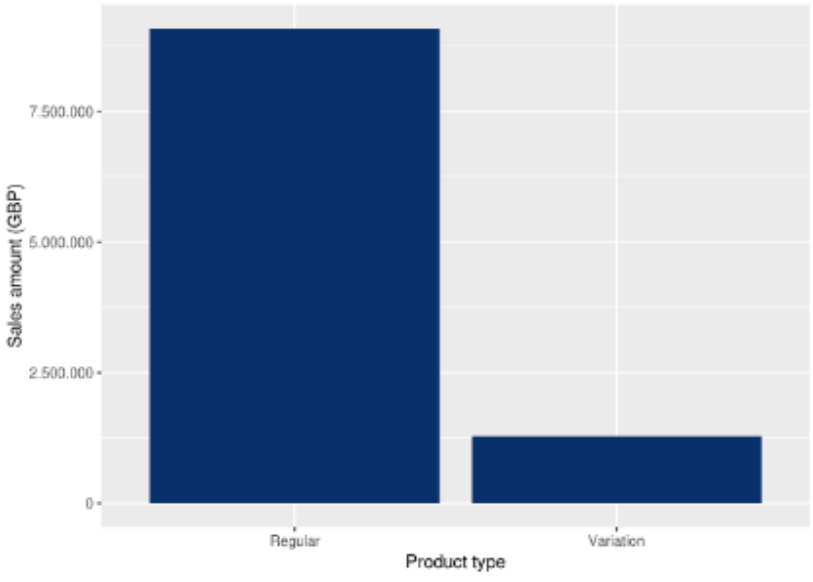

Total UK wholesale customers by month
Number of distinct customers
1.500
1.000
500
0
201012
201101
201102
201103
201104
201105
201106
201107
201108
201109
201110
201111
Sales month


Total UK wholesale sales by month
Sales amount (GBP)
1.000.000
750.000
500.000
250.000
0
201012
201101
201102
201103
201104
201105
201106
201107
201108
201109
201110
201111
Sales month
Customer tenure group
New
Existing



So, roughly a quarter of all products are variations. Is the relationship the same if I look at sales amounts, instead of product counts?

```
sales_transactions_tbl %>%
  mutate(ProductType = ifelse(substr(StockCode, -1, 1) %in% letters | substr(StockCode, -1, 1) %in% LETTERS, "Variat
egular")) %>%
  group_by(ProductType) %>%
  summarise(SalesAmount = sum(Quantity * UnitPrice, na.rm = TRUE)) %>%
  collect %>%
  ggplot() +
  geom_bar(aes(x = ProductType, y = SalesAmount), stat = "identity", position = "dodge", fill = "#083d6b") +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark = ",", scientific = FALSE)) +
  labs(x = "Product type", y = "Sales amount (GBP)")
```



```
library(wordcloud)

wordcloud(freq_itemsets$itemset, freq_itemsets$freq, max.words = 20, scale=c(0.1, 3.0),rot.per = 0,
              colors=brewer.pal(8, "Dark2"), random.order = FALSE, random.color = FALSE, fixed.asp = FALSE)
```


```
library(networkD3)
```

Conclusion and Future Work:

- In conclusion, our project on online retail data analysis using R, tidyverse, sparklyr, and Spark has provided valuable insights into customer behavior, sales trends, and product performance for online retail businesses. By leveraging the power of big data tools and techniques, we were able to analyze a large and complex dataset of over 541,000 transactions with ease.

- Our analysis revealed several key findings, including trends in customer spending and product popularity, as well as patterns in product returns and cancellations. These insights can be used by businesses to inform their marketing strategies, inventory management, and customer service efforts, ultimately leading to improved profitability and customer satisfaction.

- The combination of R, tidyverse, sparklyr, and Spark proved to be an effective approach for handling big data and performing distributed computing, allowing us to analyze vast amounts of data quickly and efficiently. This project serves as a demonstration of the power of big data tools and techniques in the field of online retail, highlighting the importance of leveraging data analytics for business success.

References:

[1.] Acquisti, A. (2004), Privacy in electronic commerce and the economics of immediate gratification, in EC '04 Proceedings of the 5th ACM conference on Electronic commerce, pp. 21–29.

[2.] Ahmed, S.R. (2004), "Applications of data mining in retail business", IEEE Information Technology: Coding and Computing, Proceedings. ITCC 2004, International Conference, Vol. 2

[3.] Ashley, C. and Noble, S.M. (2013), "It's closing time: territorial behaviours from customers in response to front line employees", Journal of Retailing, Vol. 90 No. 1, pp. 74–92.

[4.] Bagozzi, R.P. (1980), "Causal Models in Marketing", Wiley, New York.

[5.] Baier, D. and Decker, R. (2012), "Special issue on data analysis and classification in marketing – preface by the guest editors", Advances in Data Analy