

Online Retail Analysis

Sharn-konet Reitsma

[Task One] Loading the Data

```
data <- read_csv("online_retail_II.csv")
summary(data)
```

```
##      Invoice      StockCode      Description      Quantity
## Length:1067371 Length:1067371 Length:1067371 Min.    :-80995.00
## Class :character Class :character Class :character 1st Qu.:    1.00
## Mode  :character Mode  :character Mode  :character Median :    3.00
##                                     Mean  :    9.94
##                                     3rd Qu.:   10.00
##                                     Max.   : 80995.00
##
##      InvoiceDate      Price      Customer ID
## Min.    :2009-12-01 07:45:00 Min.    : -53594.36 Min.    :12346
## 1st Qu.:2010-07-09 09:46:00 1st Qu.:    1.25 1st Qu.:13975
## Median :2010-12-07 15:28:00 Median :    2.10 Median :15255
## Mean   :2011-01-02 21:13:55 Mean   :    4.65 Mean   :15325
## 3rd Qu.:2011-07-22 10:23:00 3rd Qu.:    4.15 3rd Qu.:16797
## Max.   :2011-12-09 12:50:00 Max.   : 38970.00 Max.   :18287
##                                     NA's   :243007
##
##      Country
## Length:1067371
## Class :character
## Mode  :character
##
##
##
```

From the summary above, we see:

- Negative Prices
- Negative Quantities
- Missing Data (NA's)

It's hard to interpret what this data means. From this, it's clear that we need to look at the data quality.

[Task Three] Handling Returns and Data Cleaning

In this section I will investigate the anomalous data identified in the summary above. First I'll be looking into the data for which there are negative quantities.¹

```
data %>% filter(Quantity <= 0)
```

Invoice	Description	Quantity	Price	Customer ID
C489449	PAPER BUNTING WHITE LACE	-12	2.95	16321
C489449	CREAM FELT EASTER EGG BASKET	-6	1.65	16321
C489449	POTTING SHED SOW 'N' GROW SET	-4	4.25	16321
C489449	POTTING SHED TWINE	-6	2.10	16321
C489449	PAPER CHAIN KIT RETRO SPOT	-12	2.95	16321
C489449	SAVE THE PLANET MUG	-12	1.25	16321

It seems like there are a large amount of quantities which are negative. From the context for the CSV, these are cancelled transactions indicated by the "C" in the invoice. If not removed, these cancelled transactions will count as revenue as they have a positive price associated with them. Further investigation shows that these invoice codes have no regular counter part (ie. no double-counting is occurring), we can just safely remove all of the cancelled invoice codes.

Along with these are some observations of returns, negative quantities with no "C". Because some objects may be purchased before the data collection period, we want to remove their corresponding returns. We could do this by searching for matches of particular quantity, customer ID, and stock code, however, because I'm limited on time, and because this is a very small portion of the data set, **I am going to remove these returns instead.**

Investigating Other Anomalous Data

Next I investigate what observations are strictly below zero.

```
data %>% filter(Price < 0)
```

Invoice	Description	Quantity	Price	Customer ID
A506401	Adjust bad debt	1	-53594.36	NA
A516228	Adjust bad debt	1	-44031.79	NA
A528059	Adjust bad debt	1	-38925.87	NA
A563186	Adjust bad debt	1	-11062.06	NA
A563187	Adjust bad debt	1	-11062.06	NA

Data which is less than zero is there to adjust for bad debt. It's unrelated to revenue and, due to the large numbers, may influence any statistics we generate about the data. As such I will remove these observations from the dataset.

Next we will check on the missing Customer ID data.

```
data %>% filter(is.na(`Customer ID`), Quantity > 0)
```

¹Task Three has been integrated into my data cleaning, as this is where I originally dealt with the returned items and due to the effects on task two.

Invoice	Description	Quantity	Price	Customer ID
489525	BLUE PULL BACK RACING CAR	1	0.55	NA
489525	SET/6 3D KIT CARDS FOR KIDS	1	0.85	NA
489548	FELTCRAFT DOLL ROSIE	1	2.95	NA
489548	FELT TOADSTOOL LARGE	12	1.25	NA
489548	FELTCRAFT DOLL MOLLY	3	2.95	NA
489548	LARGE HEART MEASURING SPOONS	1	1.65	NA

It seems as though this data is still useful, it should only be filtered out if we explicitly need to use the customer ID in our analysis.

Clean Data

From the above investigation, it's clear that some of the data needs to be removed before we can summarise and visualise the data.

```
# Remove the cancelled invoices
cancelled_invoices <- data %>% filter(grepl("C", Invoice))
cancelled_invoices <- cancelled_invoices$Invoice %>% unique()

filtered_data <- data %>% filter(!(Invoice %in% cancelled_invoices))

# Remove all returns
filtered_data <- filtered_data %>% filter(Quantity > 0)

# Remove invoices with negative pricing
filtered_data <- filtered_data %>% filter(Price > 0)
```

[Task Two] Plotting the Data

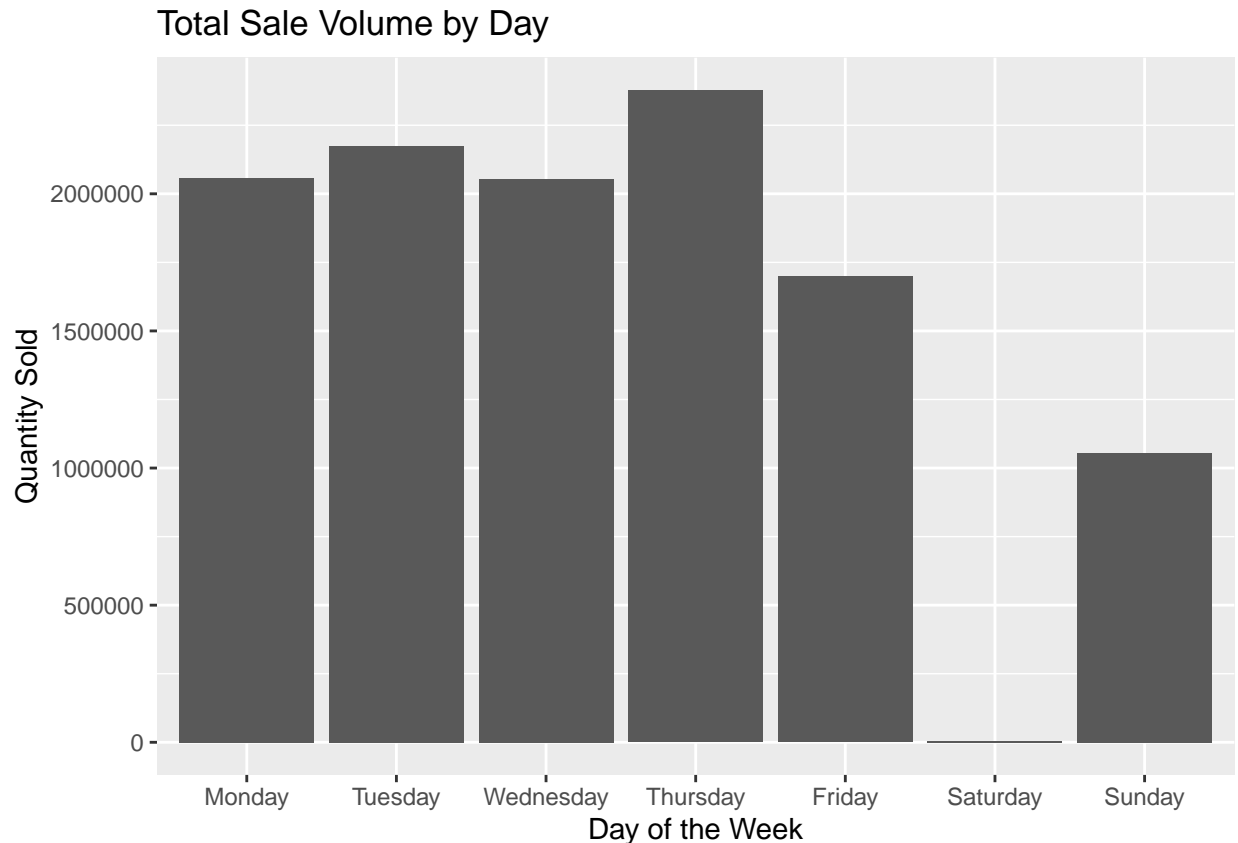
Before modelling the data, it's a good idea to do some exploratory analysis and investigate what relationships lie in the data.

Plotting Sales Volumes

```
filtered_data <- filtered_data %>% mutate(dow = weekdays(InvoiceDate))

filtered_data$dow <- factor(filtered_data$dow,
                           levels = c("Monday", "Tuesday", "Wednesday",
                                       "Thursday", "Friday", "Saturday", "Sunday"),
                           ordered = TRUE)

ggplot(data = filtered_data, aes(x = dow)) +
  geom_bar(aes(weight = Quantity)) +
  labs(title = "Total Sale Volume by Day",
       x = "Day of the Week",
       y = "Quantity Sold")
```



There's an abnormally small number of sales on Saturdays. Because their customers are mostly wholesalers, there is likely a reduced number of customers toward the weekend which we can see in the Friday and Sunday total. This also means that it's likely the low number of sales on Saturday is due to the store being closed. This should be confirmed with the owner.

Plotting Revenue Share

Unfortunately, I was not able to finish this plot. Given more time, I would have used the top 10 customer IDs and products to limit the number of groupings there were in the plot. I have however summarised the data, which conveys much of the same information as the plot would have.

```

lastest_date <- filtered_data$InvoiceDate %>%
  unique() %>%
  .[order(., decreasing = TRUE)] %>%
  .[[2]]

prev_month_data <- filtered_data %>% filter(year(InvoiceDate) == year(lastest_date),
                                           month(InvoiceDate) == month(lastest_date - dmonths(1)))

prev_month_data <- prev_month_data %>% mutate(revenue = Quantity * Price)

# Grab the top 10 customers
customer_id_summ <- prev_month_data %>%
  group_by(`Customer ID`) %>%
  summarise(total_rev = sum(revenue)) %>%
  arrange(desc(total_rev)) %>% head(11)

```

```

top_10_customers <- customer_id_summ$`Customer ID`[2:11]

stock_code_summ <- prev_month_data %>%
  group_by(`StockCode`) %>%
  summarise(total_rev = sum(revenue)) %>%
  arrange(desc(total_rev)) %>% head(10)

top_10_products <- stock_code_summ$StockCode

#
# Set all labels except for the top 10 of customer ID and Stock code to "Other"
#
# Plot, given the correctly labelled data:
#
# ggplot(data = prev_month_data, aes(x = StockCode, fill = `Customer ID`)) +
#   geom_bar(aes(weight = Price)) +
#   labs(title = "Last Month's Revenue by Product and Customer",
#         x = "Product",
#         y = "Total Revenue (£)",
#         legend = "Prodduct")

```

Table 4: Top 10 Customers by Revenue

Customer	Total Revenue
17450	27869.45
14096	27827.78
14646	25375.41
14911	22720.73
14088	16851.61
18102	15331.08
17511	13522.22
17389	12361.36
12748	10639.23
13081	9467.26

Table 5: Top 10 Products Sold by Revenue

Product	Total Revenue
DOT	36905.40
23084	34556.72
22086	28985.04
22197	14195.60
85123A	14136.70
22423	13799.58
22910	12944.49
23355	11824.42
85099B	11641.37
79321	11485.77

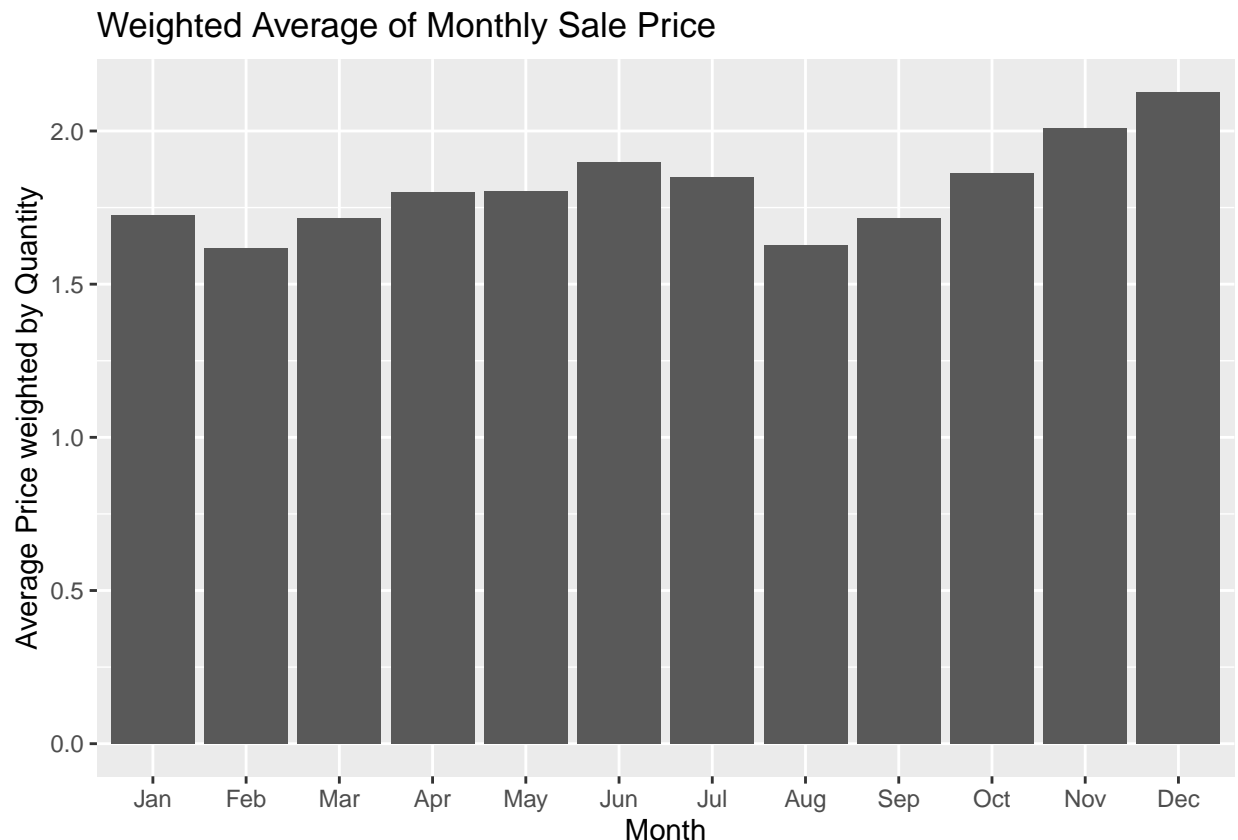
Plotting Average Price Weighted by Quantity Sold

```
monthly_data <- filtered_data %>% mutate(month = months(InvoiceDate, abbreviate = TRUE),
                                          year = year(InvoiceDate))

monthly_summary <- monthly_data %>%
  group_by(month) %>%
  summarise(weighted_average = weighted.mean(Price, w = Quantity))

monthly_summary$month <- factor(monthly_summary$month,
                                levels = c("Jan", "Feb", "Mar",
                                             "Apr", "May", "Jun",
                                             "Jul", "Aug", "Sep",
                                             "Oct", "Nov", "Dec"),
                                ordered = TRUE)

ggplot(data = monthly_summary, aes(x = month)) +
  geom_bar(aes(weight = weighted_average)) +
  labs(title = "Weighted Average of Monthly Sale Price",
       x = "Month",
       y = "Average Price weighted by Quantity")
```



From this we can see that there's some seasonal trends in the data. It slowly climbs to a peak in January-July and in August-December. The peak in December is likely caused by a higher demand in gift-ware during Christmas, especially because the store sells Christmas decorations. I'm unsure what the July peak is caused by, it might be worth asking a subject matter expert, such as the owner, who might have more intuition.

[Task Four] Predicting Monthly Revenue

Linear Regression

Create a linear model using the price, quantity, and timestamps within the data. The response variable of interest would be Total Revenue, calculated through $Price \times Quantity$. Revenue could then be aggregated over each month, and both the month and year could be used as explanatory variables.

It would be important to convey the uncertainty of the prediction through confidence intervals.

To evaluate model predictions, I would use cross-validation to get an unbiased estimate of the error.

The metrics which will be of interest include:

- Price
- Quantity
- Timestamps

ARIMA Model

Because the intention is to forecast this month's revenue, and because we have timeseries data, an ARIMA model would also be suitable here. It should be fairly reliable for forecasting only one month ahead, so long as there are no significant events which would throw off the trend.

Similarly to above, I would aggregate the total revenue monthly and use this as the data that's fed into the ARIMA model.

While experimenting with different models in the ARIMA family, I would also investigate integrating a seasonal component through a SARIMA model as there is likely seasonality in this kind of data (higher sales during certain periods), as commented on previously.

The metrics which will be of interest include:

- Price
- Quantity
- Timestamps

Uncertainties to explore:

For each of the above models, the below uncertainties would need to be taken into account:

- What plans the retailer has for discounts during the Christmas period, and if this is different from previous years
- What new items are being sold on the store? Will any be in particularly high demand for Christmas?

And generally, it will be worth asking the retailer if they're making any changes to how they're running the business in the coming month (new website, new membership program, etc.). Past data won't give a great forecast if it isn't representative of what's happening this month.

Other Models

Given more time, I would do a literature review and investigate what models are explored in "Forecasting: Principles and Practice" by Rob J Hyndman and George Athanasopoulos. The text is openly available <https://otexts.com/fpp2/> link and is a great resource to help interpret and utilise timeseries data.

[4b] Best Approach

Of the two models above, I would recommend the ARIMA model as the best approach. It takes advantage of the timeseries nature of the data and, while more complex than the linear model approach, is not overly complex for the increase in accuracy it would provide (like something to the tune of a neural network). The methodology is well established, and there are existing libraries in R (namely `<forecast>` link) to support the development of this kind of model.

[4c] Implementation of Chosen Solution

For this I'm going to choose to go with the ARIMA model. Largely, this is because it provides some quick functionality for comparing the historical data to the prediction. If given more time, I would choose to compare a handful of researched methods and compare them, or would ask for advice from peers, since I have limited experience in the area.

```
library(forecast)

revenueTS <- monthly_data %>%
  mutate(Revenue = Price*Quantity,
         date = floor_date(as.Date(InvoiceDate), "month")) %>%
  group_by(date) %>%
  summarise(total_revenue = sum(Revenue)) %>%
  arrange(date)

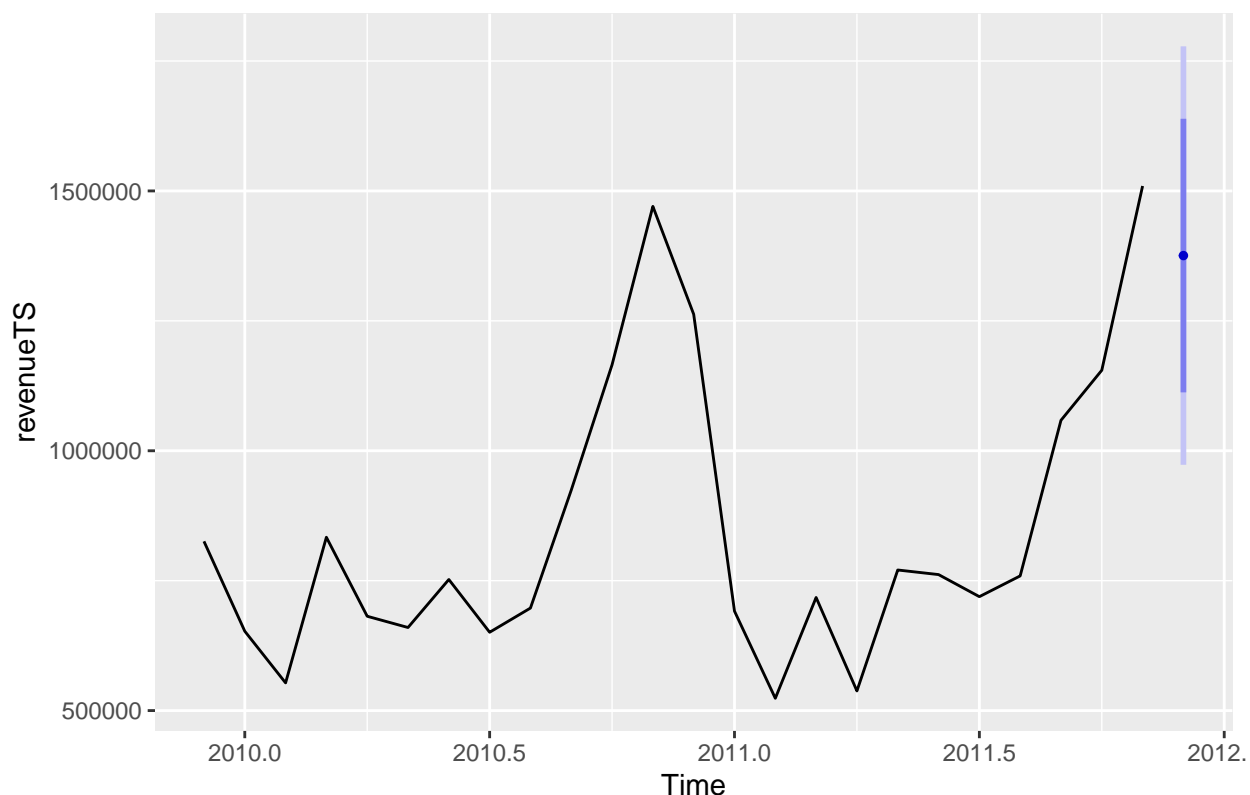
# Remove the current days we have from that month
revenueTS <- revenueTS %>% filter(date < "2011-12-01")

revenueTS <- revenueTS %>% select(total_revenue) %>% ts(start = c(2009, 12),
                                                         end = c(2011, 11),
                                                         deltat = 1/12)

model <- auto.arima(revenueTS, lambda = "auto")

model %>% forecast(h = 1) %>% autoplot()
```


Forecasts from ARIMA(2,0,0) with non-zero mean



The model above shows the prediction for the total amount of revenue earned during December of 2011, along with confidence intervals. The darker confidence interval is an 80% confidence interval, and the lighter band is a 95% confidence interval.

I'm confident in the forecast, however this is more due to the large uncertainty in the prediction, rather than how robust my method was.

Given more time I would do a more thorough analysis where I look into autocorrelations and am more involved in the modelling process. I would look into exploring daily/weekly summaries of the data, as the monthly data here is quite limiting and does not give a good prediction of total revenue. It may also be worth exploring the results of linear regression and seeing how they compare to the estimate here.

As to whether or not I would recommend a new ferrari based on this forecast, that's a different matter altogether. In 2011, the conversion rate from sterling to USD was around $\text{£}1 \text{ GBP} = \1.6025 USD . Around then a new ferrari was about \$225,325; so around $\text{£}140,000$. Of course, whether or not he could buy the ferrari entirely depends what proportion of the money he would reinvest into his business, and how much he gets to take to the bank. Regardless, spending $\text{£}140,000$ based on an analysis I did in 30 minutes seems ill-advised. I would not recommend it.