Quantiful Internship Programme Assessment

Josh Atwal

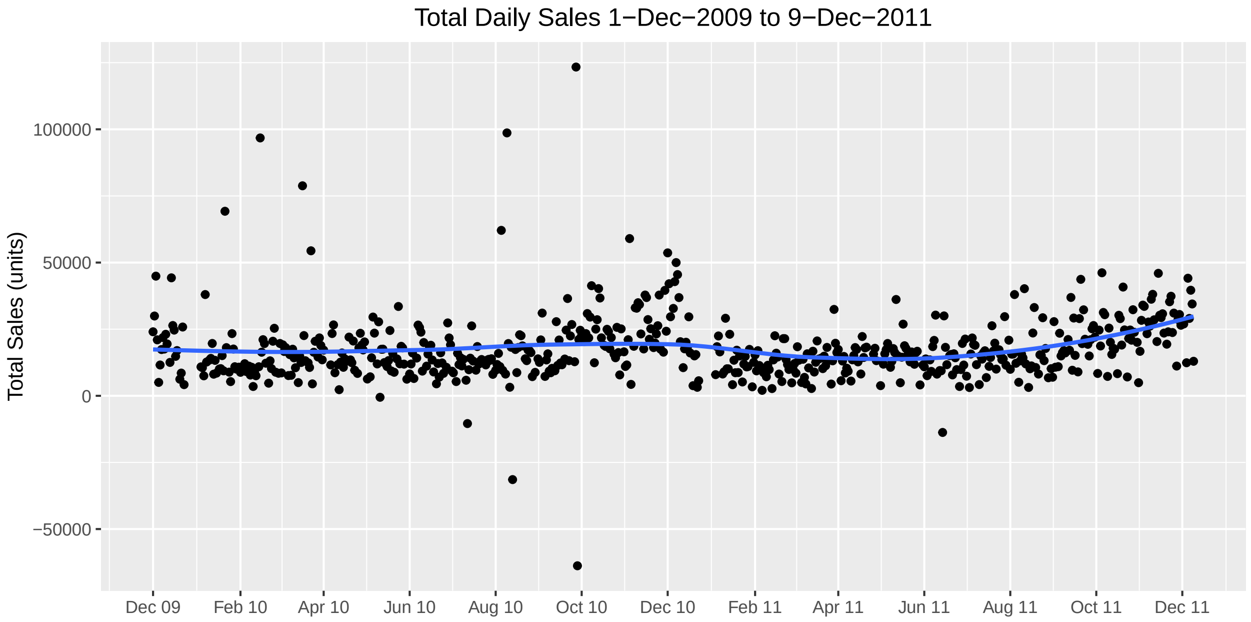
25/08/2020

```r
retail <- read.csv('online_retail_II.csv')
retail$InvoiceDate <- as.Date(retail$InvoiceDate)
```

This analysis concerns a dataset of sales and returns collected over a period December 1st 2009 until December 9th 2011. (This document was originally formatted for HTML viewing).

Exploratory analysis

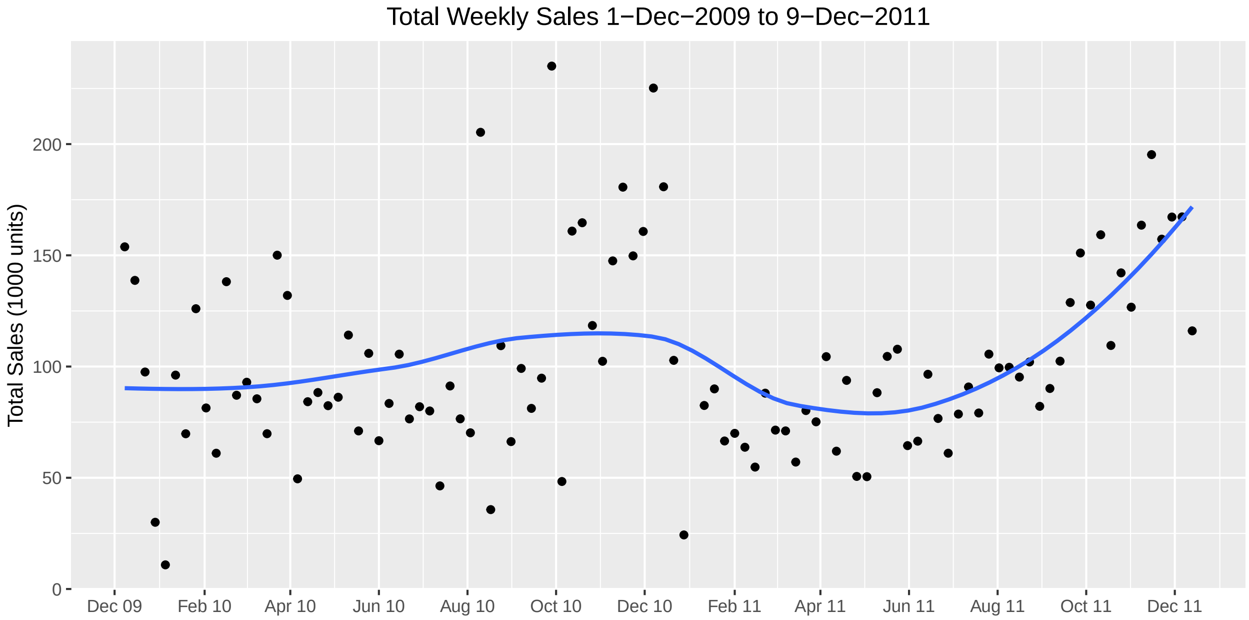
```r
retail %>% group_by(day=date(InvoiceDate), month=month(InvoiceDate)) %>%
        summarise(n=sum(Quantity)) %>%
        ggplot(aes(x=day,y=n)) +
        labs(x="Date",
             y="Total Sales (units)",
             title="Total Daily Sales 1-Dec-2009 to 9-Dec-2011", colour="Year") +
        geom_point() +
        geom_smooth(se = FALSE) +
        theme(plot.title = element_text(hjust = 0.5))+
        scale_x_date(NULL,
                     date_labels = "%b %y",
                     breaks = seq(as.Date("2009-12-1"), as.Date("2011-12-1"), by="2 month"))
```



Daily sales look fairly stable about a mean of 17.5k units per day with surges leading up to the holiday season.

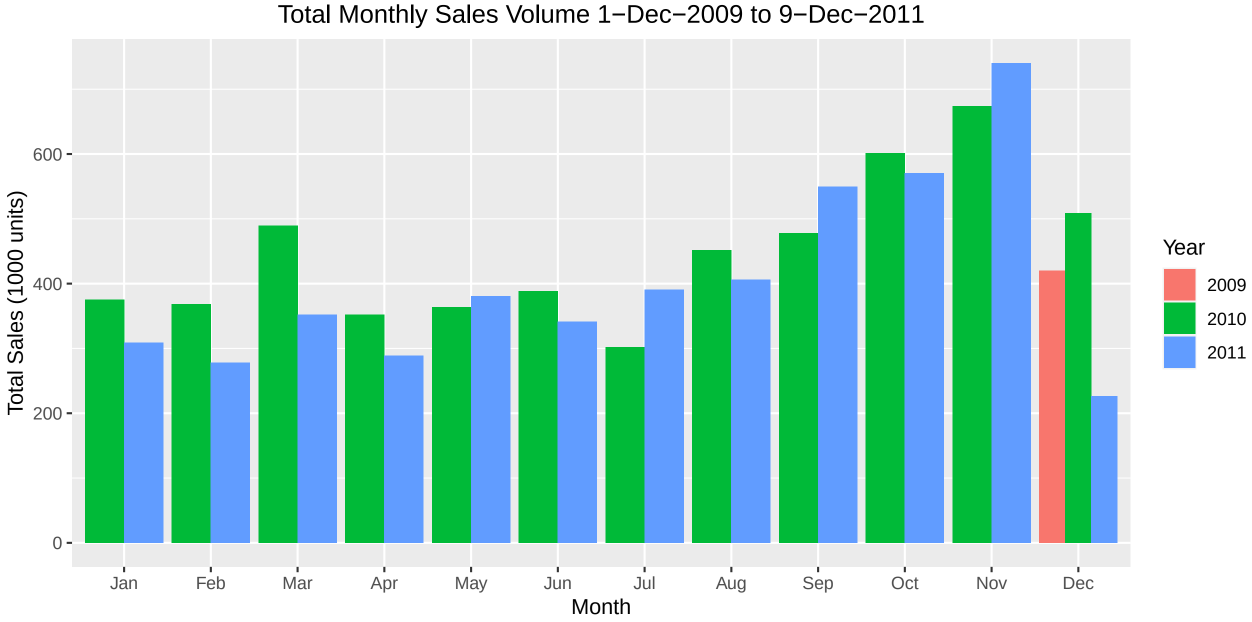
Negative outliers are present, relating to large returns or manual adjustments, but some positive outliers are also present.

```
# First extract the initial and final days of data collection
dateInit <- head(retail$InvoiceDate, 1)
dateFinal <- tail(retail$InvoiceDate, 1)

retail %>%
        # Cumulative week variable
        mutate(cumWeek=interval(InvoiceDate[1], InvoiceDate) %/% weeks(1) + 1) %>%
        group_by(cumWeek) %>%
        summarise(n=sum(Quantity)/1000) %>%
        ggplot(aes(x=retail$InvoiceDate[1] + weeks(cumWeek),y=n)) +
        labs(x="Month, Year",
             y="Total Sales (1000 units)",
             title="Total Weekly Sales 1-Dec-2009 to 9-Dec-2011", colour="Year") +
        geom_point() +
        geom_smooth(se = FALSE) +
        theme(plot.title = element_text(hjust = 0.5)) +
        scale_x_date(NULL, date_labels = "%b %y", breaks = seq(dateInit, dateFinal, by="2 month"))
```



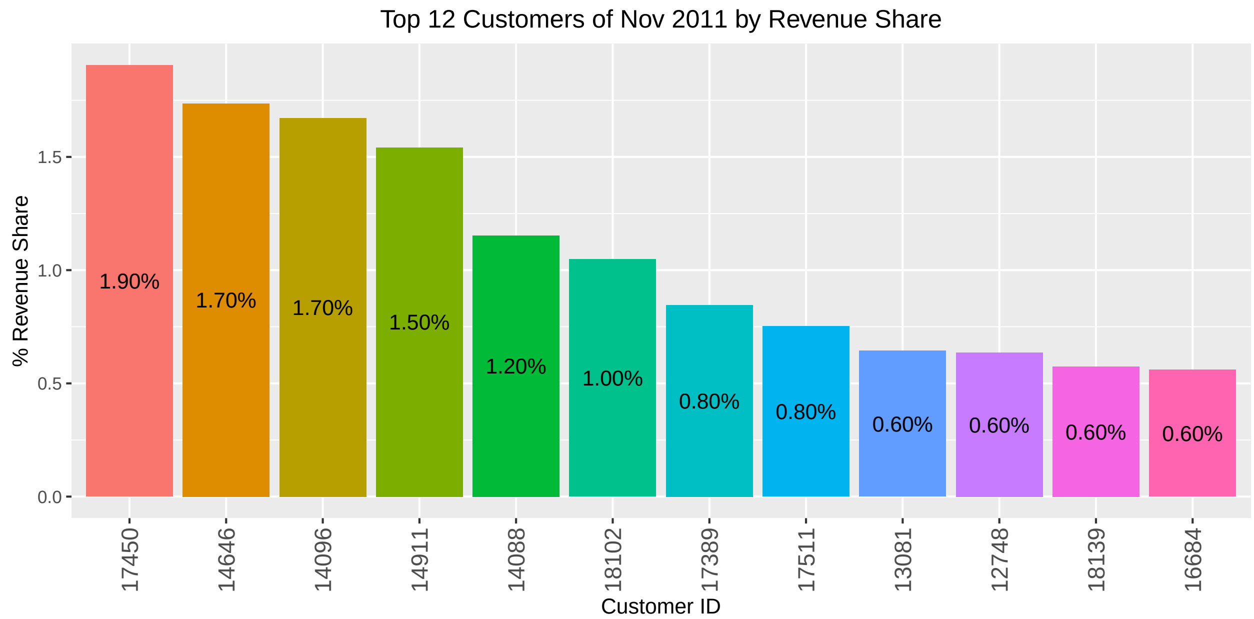
Weekly sales show more clearly the trend of sales, with a surge followed by a dip observed over the 2010-11 holiday season. 2011 recovers from the dip similarly to the previous year at the same mid-year mark, but continues its growth upwards to a level higher than in the previous season.

```
retail %>% group_by(month=month(InvoiceDate, label=T, abbr=T),
                    Year=factor(year(InvoiceDate))) %>%
        summarise(n=sum(Quantity)/1000, .groups="drop") %>%
        ggplot(aes(x=factor(month),y=n, fill=Year)) +
        labs(x="Month",
             y="Total Sales (1000 units)",
             title="Total Monthly Sales Volume 1-Dec-2009 to 9-Dec-2011",
             colour="Year") +
        geom_bar(stat="identity",position=position_dodge()) +
        theme(plot.title = element_text(hjust = 0.5))
```



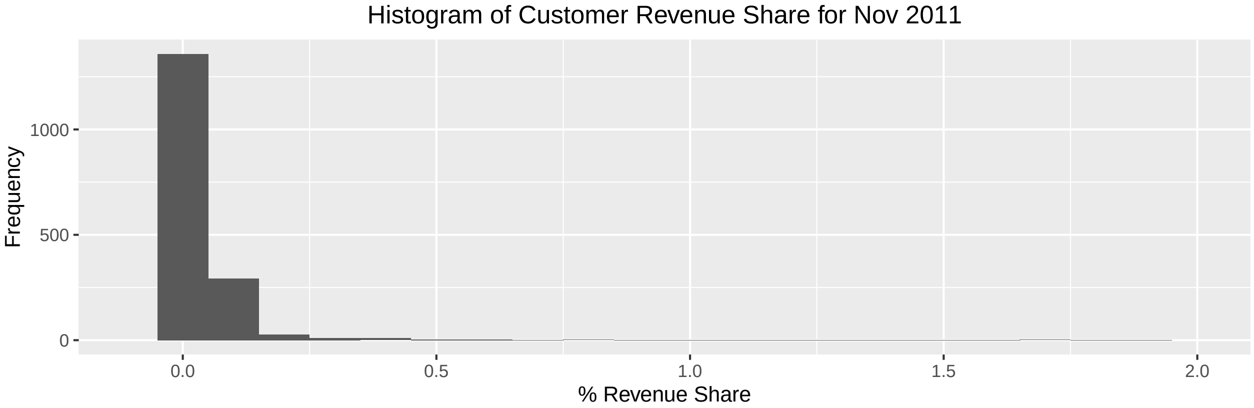
(Note that the dataset does not include sales for the whole month of Dec 2011, explaining the seeming under-performance.) As seen in the weekly sales plot, 2011 begins with a dip in sales below the level of 2010, however recovers to a higher level in the months leading up to December. It seems as though by December, a lot of people have already done their shopping and so this is the month we see sales begin to drop off.

```
currentDate <- tail(retail$InvoiceDate,1)
# Calculate share as proportion of total revenue
shares <- retail %>% filter(month(InvoiceDate)==month(currentDate)-1,
                            year(InvoiceDate)==year(currentDate)) %>%
           group_by(cust=factor(Customer.ID)) %>%
           summarise(Revenue=sum(Quantity*Price)) %>%
           mutate(share=100*Revenue/sum(Revenue))

shares %>%  arrange(desc(share)) %>%
           slice_head(n=13) %>%
           slice_tail(n=12) %>% # Remove missing customer ID
           ggplot(aes(x=reorder(cust, -share), y=share, fill=reorder(cust, -share))) +
           geom_bar(stat="identity") +
           geom_text(aes(label = scales::percent(round(share/100,3))),
                     position = position_stack(vjust = 0.5)) +
           theme(plot.title = element_text(hjust = 0.5),
                 axis.text.x  = element_text(angle=90, vjust=0.5, size=12)) +
           guides(fill=FALSE) +
           labs(x="Customer ID",
                y="% Revenue Share",
                title="Top 12 Customers of Nov 2011 by Revenue Share")
```



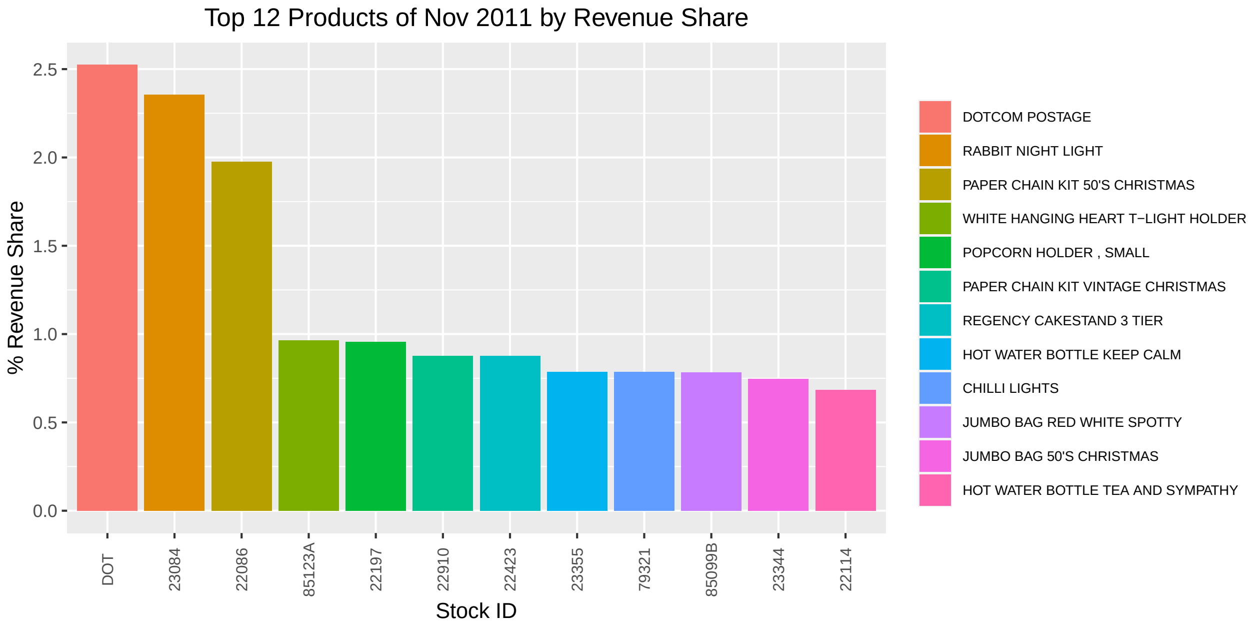
With the largest customer only having a revenue share of 1.9%, last month's revenue seems to be distributed among many customers with only the very top 0.1% of customers having a revenue share greater than 1%.

```
shares %>% ggplot(aes(x=share)) +
                  geom_histogram(binwidth=.1) +
                  xlim(-0.1,2) +
                  labs(x="% Revenue Share",
                       y="Frequency",
                       title="Histogram of Customer Revenue Share for Nov 2011") +
                  theme(plot.title = element_text(hjust = 0.5))
```



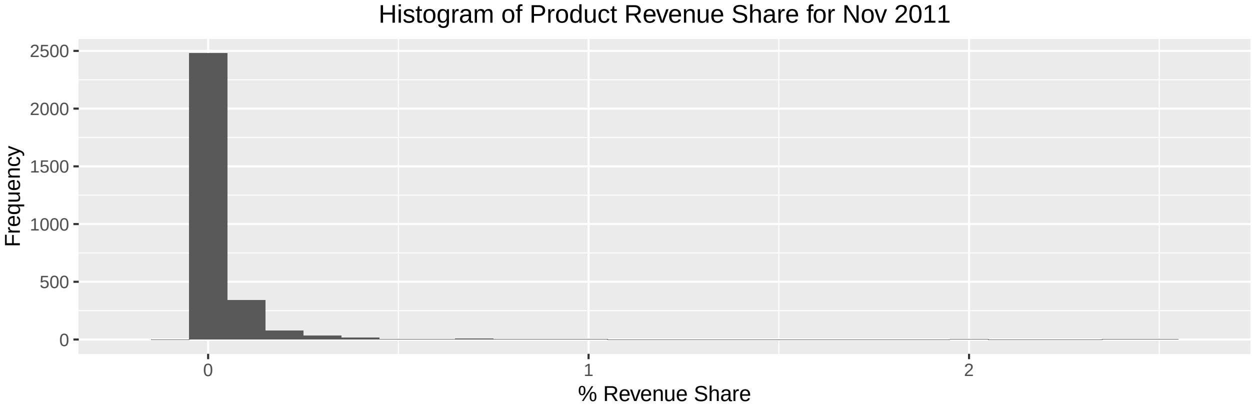
A histogram of % revenue share confirms this, and shows a great bulk customers having only a fractional share of the revenue. Note that these two graphs ignore the fact that 22.5% of the revenue share was not attributable to a customer and was due to things such as manual adjustments.

```
allProducts <- retail %>%
            filter(month(InvoiceDate)==month(currentDate)-1, year(InvoiceDate)==year(currentDate)) %>%
            group_by(item=factor(StockCode)) %>%
            summarise(Revenue=sum(Quantity*Price)) %>%
            mutate(share=100*Revenue/sum(Revenue))

topProducts <- allProducts %>% arrange(desc(share)) %>%
              slice_head(n=12)

descs <- sapply(topProducts$item, function(i){
  d <- retail$Description[as.character(retail$StockCode) == i]
  d[d!=""][1] # First non empty description
})
legend_ord <- levels(reorder(factor(descs), topProducts$share))

topProducts %>% mutate(descs) %>%
            ggplot(aes(x=reorder(item, -share), y=share, fill=reorder(item, -share))) +
            geom_bar(stat="identity") +
            labs(x="Stock ID",
                 y="% Revenue Share",
                 title="Top 12 Products of Nov 2011 by Revenue Share", fill="")+
            theme(plot.title = element_text(hjust = 0.5),
                  legend.text = element_text(size = 7),
                  axis.text.x  = element_text(angle=90, vjust=0.5, size=8)) +
            scale_fill_discrete(labels=descs)
```



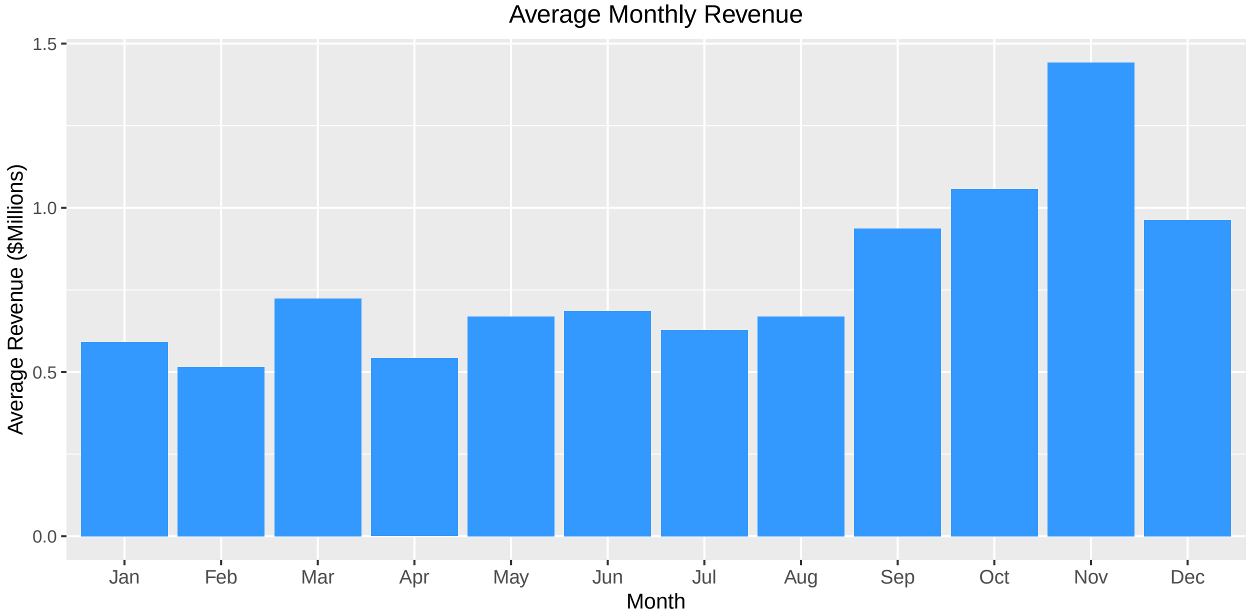
One or two items stood out last month as exceptionally great revenue generators, the “Rabbit Night Light” and “Paper Chain Kit 50’s Christmas” which each held a share of about 2% of the revenue last month. The greatest revenue generator seems to be postage but much profit is unlikely to be generated from the mere shipping of goods.

```
allProducts %>% ggplot(aes(x=share)) +
                  geom_histogram(binwidth=.1) +
                  xlim(-0.2,2.6) +
                  labs(x="% Revenue Share",
                       y="Frequency",
                       title="Histogram of Product Revenue Share for Nov 2011") +
                  theme(plot.title = element_text(hjust = 0.5))
```



A similar distribution is seen in the item revenue share, with a majority of products being sold in low volume and/or at low cost and resulting in a small revenue share. Only a handful of products contribute significantly to the revenue: a total of 19 products have a revenue share greater than 0.5%.

```r
# Weighted average monthly sale price by volume
retail %>% group_by(year=year(InvoiceDate),
                    month=month(InvoiceDate, label=T, abbr=T)) %>%
            summarise(grossRevenue=sum(Price*Quantity), .groups="drop") %>%
            slice_head(n=nrow(.)-1) %>% # Remove Dec 2011
            group_by(month) %>%
            summarise(rev=mean(grossRevenue)/1e6) %>%
            ggplot(aes(x=month, y=rev)) +
            geom_bar(stat="identity", fill="#3399FF") +
            labs(x="Month",
                 y="Average Revenue ($Millions)",
                 title="Average Monthly Revenue") +
            theme(plot.title = element_text(hjust = 0.5),
                  axis.text.x  = element_text(size=10))+
            scale_fill_brewer(palette="Spectral")
```



(Note that the data for December 2011 was not included in this plot because it did not span the entire month). Averaged across the entire the dataset, the monthly revenue shows the expected surge in September

- November leading up to the holiday season. December also maintains some of this effect but it has dropped off completely by January. March shows a small increase in sales, potentially in lead up to Easter.

Handling items purchased before the date of data collection

For each returned item, the dataset is searched in chronological order up until the date of the item return, for another record with the same stock code and customer id which was a purchase (as opposed to another return). If none are found then the return is assumed to have been from before the date of data collection, and filtered out of the dataset.

An obvious limitation to this approach is that the same customer could have bought the same item again since the initial purchase. Leaving the quantity unrestricted (number of returned items does not have to equal number of purchased items) increases the number of false positives but allows for cases with partial return of stock.

```r
# Boolean vector if a returned has a non-missing stock code and customer id
identifiable_returns <- which(as.integer(retail$Quantity) < 0 &
                                !is.na(retail$StockCode) &
                                !is.na(retail$Customer.ID))

# Boolean vector for all returns where there was a previous instance
# of a purchase of that item by that customer
postcollection <- sapply(identifiable_returns, function(i){
  any(retail$Quantity[1:i-1] > 0,
      retail$StockCode[1:i-1] == retail$StockCode[i],
      retail$Customer.ID[1:i-1] == retail$Customer.ID[i], na.rm=T)
})

# Filter dataset accordingly
retail_filtered <- retail[-identifiable_returns[-postcollection],]
```

This approach may be augmented by perhaps considering a company return policy that only allows returns within a separate window, or by recognising the manual adjustments made in the dataset separately from returns as presently anything with a negative quantity is considered a return.

Predicting revenue for December 2011

- **Simple extrapolation**
  - Data is known complete up until and including December 11. From previous year's data, calculate proportion of total December revenue generated up until the 11th, and use this to extrapolate the data for December 2011 to find the estimated revenue for this month.
  - Unclear how indicative the previous years worth of December data are of this year's revenue and how to weight them appropriately.
- **Linear / Time series model**
  - Fit a linear or time series model to the entire data and use it to predict the revenue for December 2011.
  - Include a general trend term as well as coefficients for each month of the year.
  - Easily explainable and verifiable (coefficients of the model come with confidence intervals)
- **Recurrent Neural Network, LSTM, Transformer**
  - Fit a sequence-based neural network model to the data and use it to predict the revenue for December 2011.
  - Difficult to implement and to explain results
  - May be possible to model revenue as a function of each product, and aggregate the revenue for each product into a gross revenue.

Chosen best approach: **Linear / Time series model**. The dataset may contain many rows, but in terms of useful information for future prediction we would hope for many more years of previous data because as of now we can only really claim to have 2 complete observations when it comes to temporal variables like month and year. Despite this, a time series or linear model may still be effective in at least identifying a general trend that is good enough for prediction without needing to incorporate–for example–customer or product sales data. Meanwhile neural networks generally require huge datasets (our current dataset may need to be sub-sampled for this approach to work) and a lot of overhead in setting, training, and optimising them. One other consideration for each approach is to recognise that

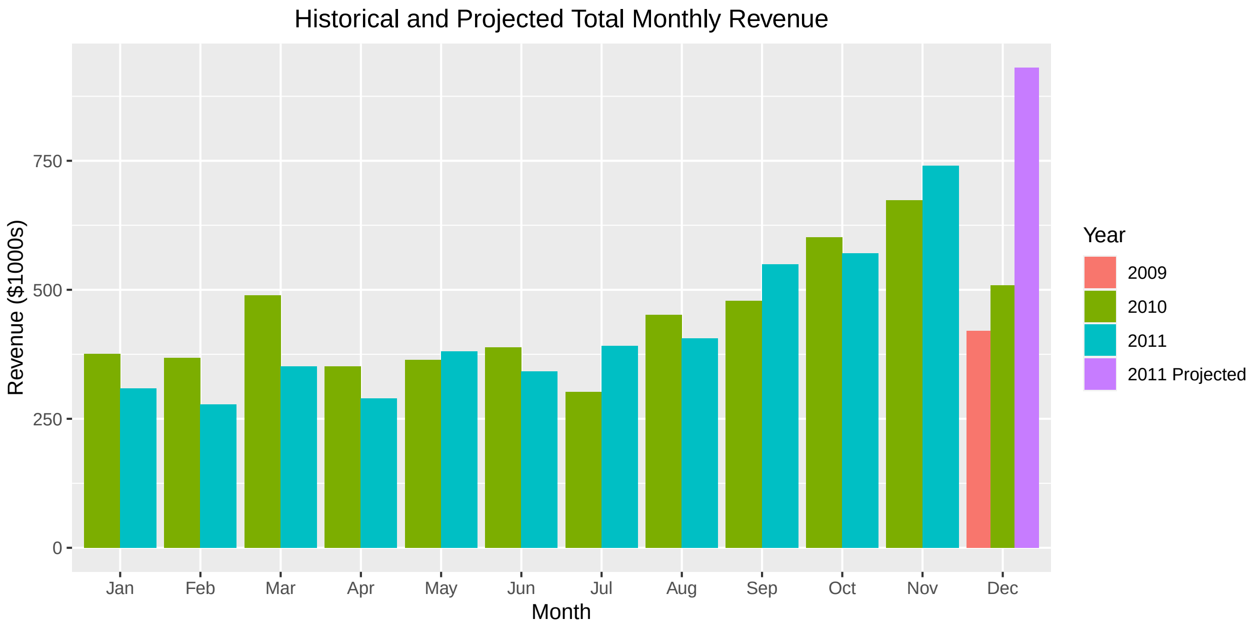
The simple extrapolation approach is low effort, but does not make use of any of the other data and therefore may ignore a long-term trend. A linear / time series model is therefore a good compromise between making use of the data we have, and being relatively simple to implement, while also being easy to explain and verify through statistical tests, and to be extended in the future as more data is collected.

Below I have implemented the simple extrapolation approach, taking the mean between the two previous year's revenue proportion generated in the early part of December and using it to extrapolate this months revenue.

```
# Sum of December revenue for each year prior to the 12th
revenue_11Dec <- retail_filtered %>% filter(month(InvoiceDate) == 12,
                                            day(InvoiceDate) <= 11) %>%
                    group_by(year=year(InvoiceDate)) %>%
                    summarise(revenue=sum(Quantity*Price))

# Sum of december revenue for each year
revenue_Dec <- retail_filtered %>% filter(month(InvoiceDate) == 12) %>%
                    group_by(year=year(InvoiceDate)) %>%
                    summarise(revenue=sum(Quantity*Price)) %>%
                    head(2)

# Proportion of total December revenue that was earned
# prior to December 12 in 2009 and 2010 respectively
revenueProps <- revenue_11Dec$revenue %>% head(2) / revenue_Dec$revenue
# Projected revenue for this month is the current total revenue
# divided by the proportion of revenue it is expected to be
projected <- revenue_11Dec$revenue %>% tail(1)  / mean(revenueProps)
sprintf("$%g", projected)
```

```
## [1] "$930452"
```

```
projectedRetail <- retail %>% group_by(month=month(InvoiceDate, label=T, abbr=T),
                                       Year=factor(year(InvoiceDate))) %>%
                              summarise(n=sum(Quantity)/1000, .groups="drop")

# Insert the projected value into the tibble
projectedRetail$n[nrow(projectedRetail)] <- projected/1000
# Add a level to the factor
levels(projectedRetail$Year) <- c(levels(projectedRetail$Year), "2011 Projected")
projectedRetail$Year[nrow(projectedRetail)] <- "2011 Projected"

projectedRetail %>% ggplot(aes(x=factor(month),y=n, fill=Year)) +
        labs(x="Month",
             y="Revenue ($1000s)",
             title="Historical and Projected Total Monthly Revenue",
             colour="Year") +
        geom_bar(stat="identity",position=position_dodge()) +
        theme(plot.title = element_text(hjust = 0.5))
```



The revenue forecast looks promising (an estimated $930k) and there is some degree of confidence in it as using the mean of the last two years (which are reasonably close to each other) is likely to give a conservative estimate given the seeming upward trend.

While the revenue alone is certainly enough to afford a new Ferrari, it is meaningless to interpret this result without a more complete view of the business, least of all knowing what the operating costs and profit on this revenue is. For example, we identified earlier that the majority share of the last month's revenue was just going towards covering shipping costs.