Sales Analysis

Lawrence May

22/08/2020

1.) Read in the data

```
defaultW <- getOption("warn")
options(warn = -1)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```
#Reading in the data
retail <- read.csv("/Users/lawrence/Downloads/sales/online_retail_II.csv")
head(retail)
```

```
##   Invoice StockCode                         Description Quantity
## 1  489434     85048 15CM CHRISTMAS GLASS BALL 20 LIGHTS       12
## 2  489434    79323P                  PINK CHERRY LIGHTS       12
## 3  489434    79323W                 WHITE CHERRY LIGHTS       12
## 4  489434     22041        RECORD FRAME 7" SINGLE SIZE        48
## 5  489434     21232      STRAWBERRY CERAMIC TRINKET BOX       24
## 6  489434     22064          PINK DOUGHNUT TRINKET POT        24
##           InvoiceDate Price Customer.ID        Country
## 1 2009-12-01 07:45:00  6.95       13085 United Kingdom
## 2 2009-12-01 07:45:00  6.75       13085 United Kingdom
## 3 2009-12-01 07:45:00  6.75       13085 United Kingdom
## 4 2009-12-01 07:45:00  2.10       13085 United Kingdom
## 5 2009-12-01 07:45:00  1.25       13085 United Kingdom
## 6 2009-12-01 07:45:00  1.65       13085 United Kingdom
```

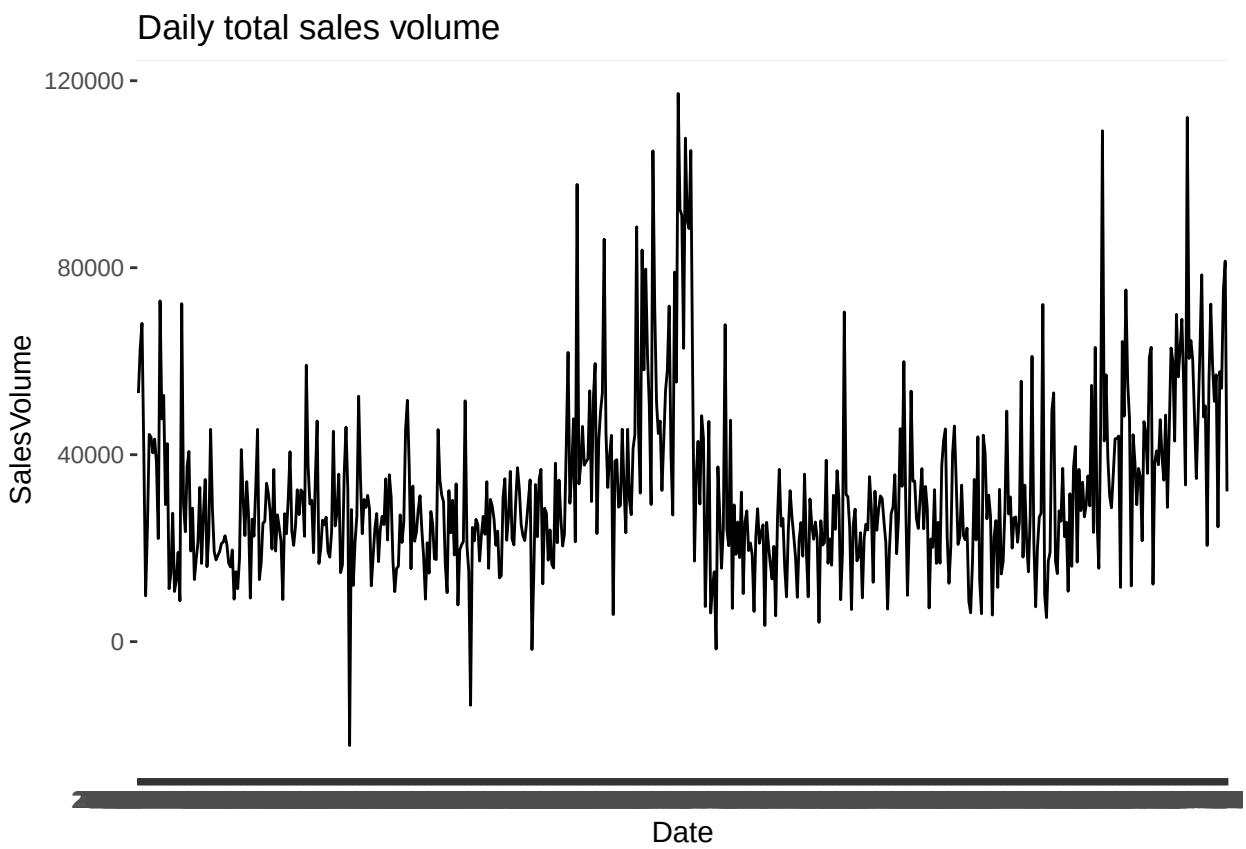
2.) Prepare and comment on the following exploratory plots: Total daily, weekly and monthly sales volumes

```r
#Preparing the data
retail %>%
  separate(InvoiceDate, into=c('Date'), sep = " ", remove=F) %>%
  separate(InvoiceDate, into=c('MonthYear'), sep = c(7), remove=F) %>%
  mutate(Sales = retail$Quantity * retail$Price)-> retail

aggregate(Sales ~ MonthYear, data=retail, FUN=sum) %>% separate(MonthYear, into=c('Year'), sep = c(4), r
aggregate(Sales ~ Date, data=retail, FUN=sum) -> daily

val<-c(mean(daily$Sales),median(daily$Sales),max(daily$Sales),min(daily$Sales))
title<-c('Mean','Median','Max','Min')
rbind(title,val)
```

```
##       [,1]               [,2]         [,3]         [,4]
## title "Mean"             "Median"     "Max"        "Min"
## val   "31932.5340529801" "27481.625" "117271.12" "-22212.609"
```

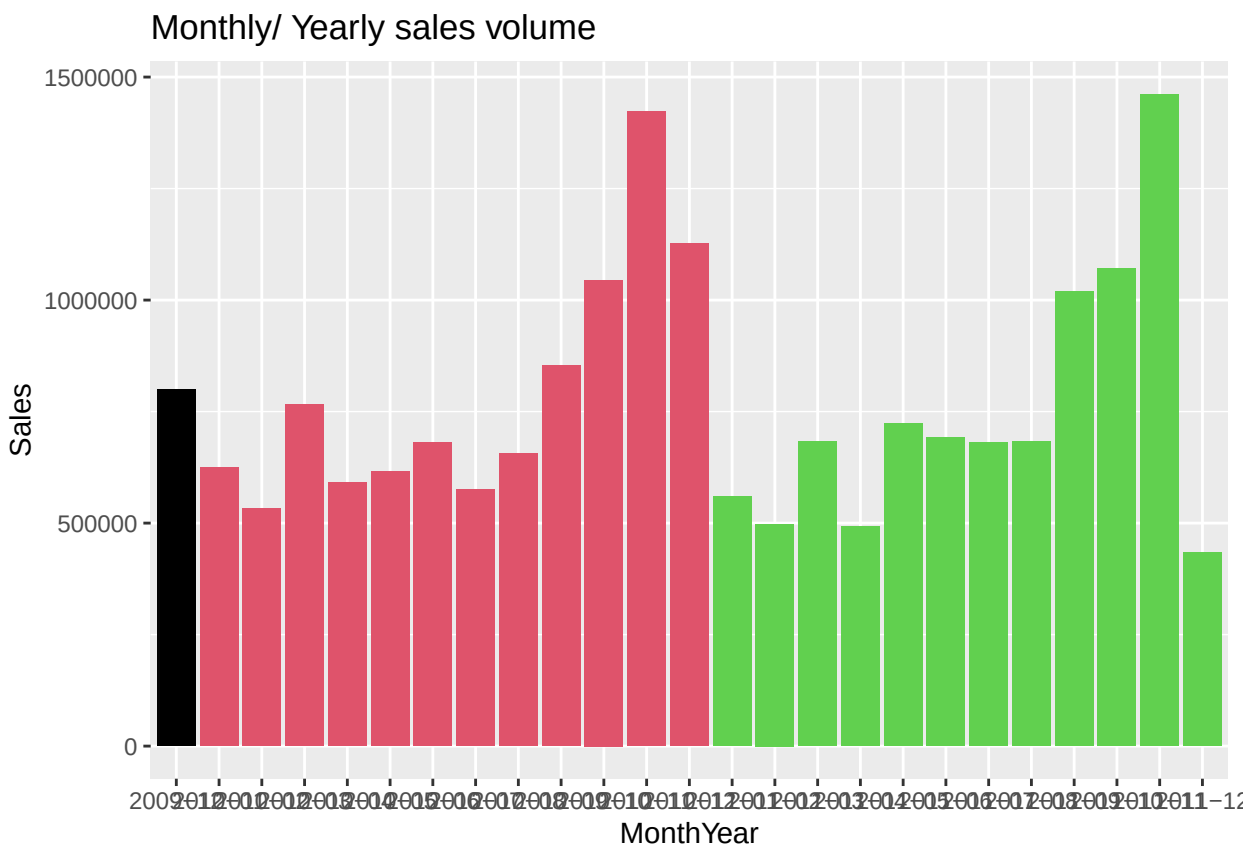
```r
colnames(daily)<-c("Date","SalesVolume")
```

```r
#Daily sales volume plot
ggplot(daily, aes(Date, SalesVolume, group = 1)) + geom_line() + ggtitle("Daily total sales volume")
```



Sales volume appears to be highly volatile by day, with the mean being at 32,000 and median at 27,000 pounds per day, a few outliers however increasing to up to 117,000 pounds per day (maybe a sale?) and -22000, maybe just after Christmas when everyone is returning their presents.

```
#Sales volume by month and year
ggplot(monthly_yearly, aes(x=MonthYear, y=Sales)) + geom_bar(stat="identity", fill = monthly_yearly$Yea
```



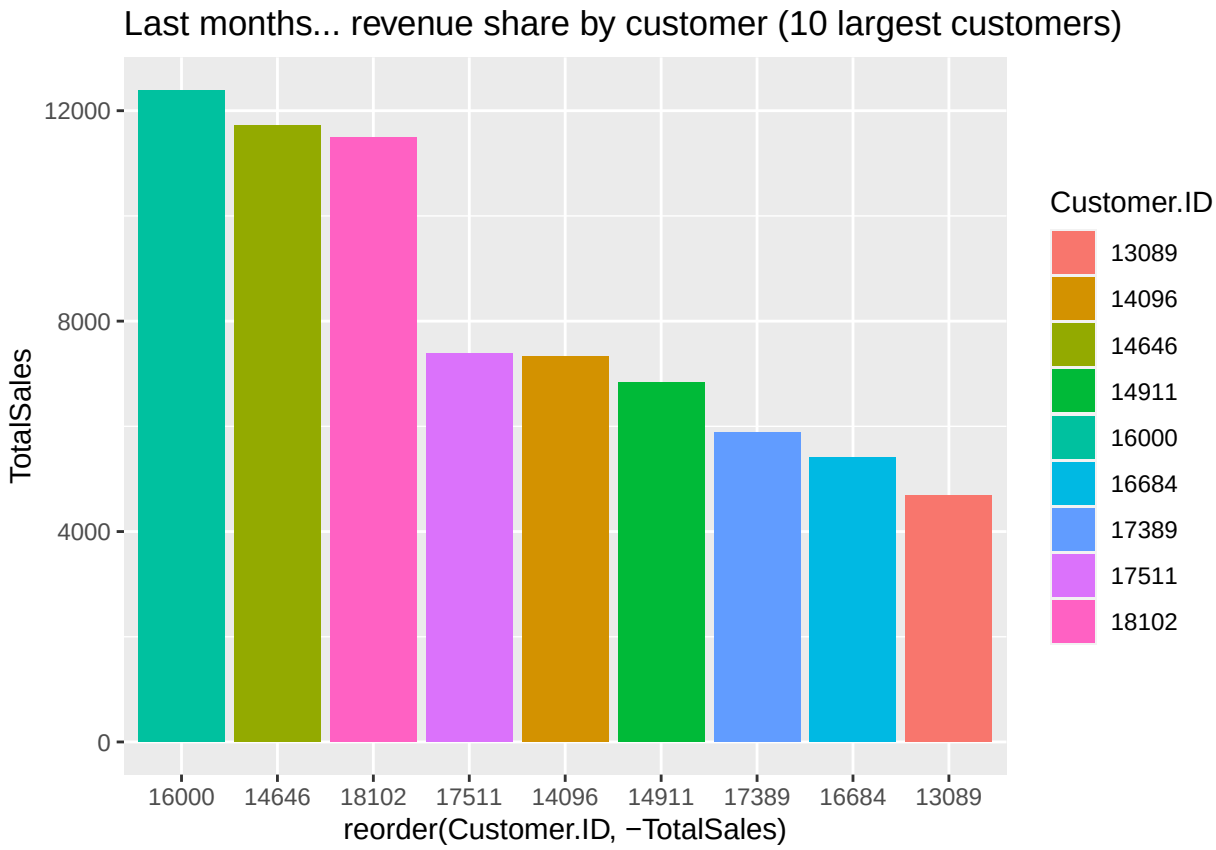
There appears to be a clear monthly pattern with increasing sales towards the end of the year (and the Christmas season).

Last months' revenue share by product and by customer

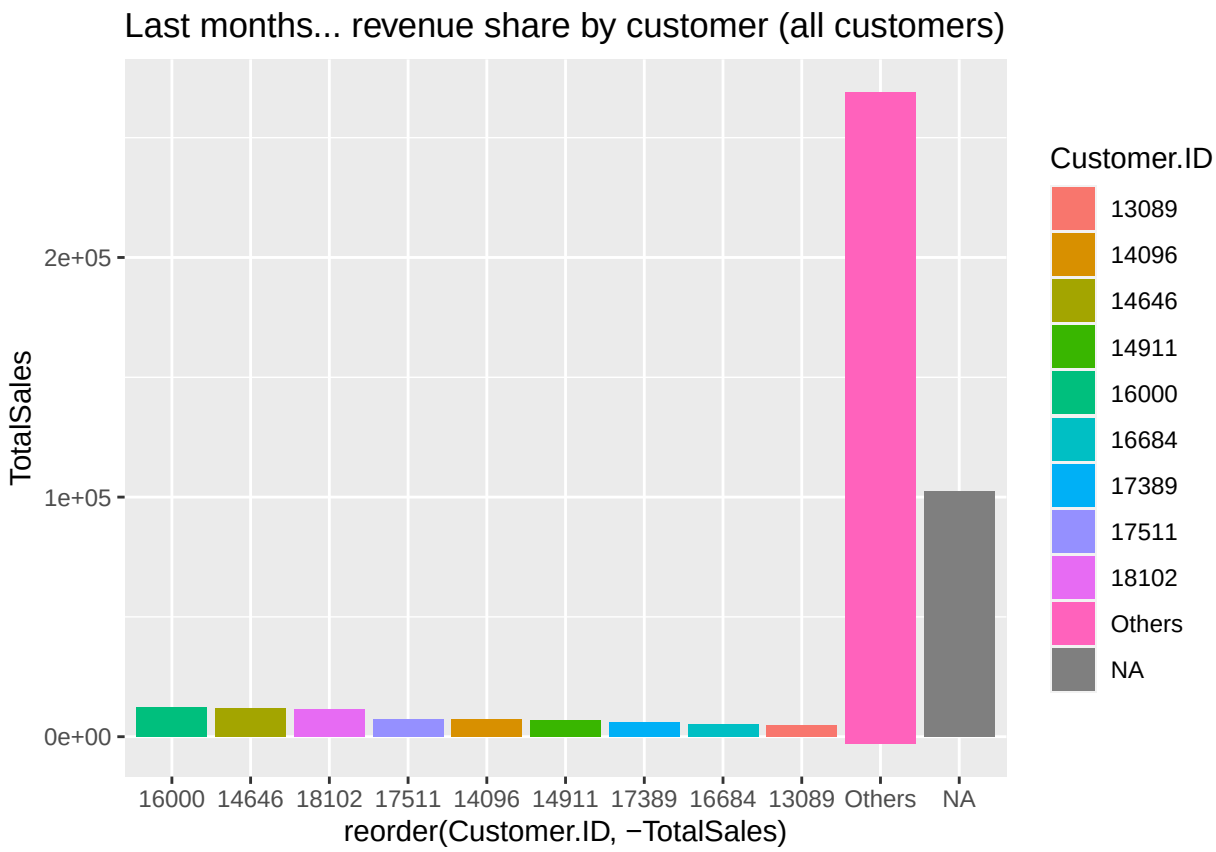
```
retail[retail$MonthYear=='2011-12',] ->lastMonth
lastMonth<-lastMonth[!(lastMonth$StockCode=="DOT" | lastMonth$StockCode=="CRUK" | lastMonth$StockCode==

by_cust <- lastMonth %>%
    group_by(Customer.ID) %>%
    summarise(TotalSales=sum(Sales)) %>%
    arrange(desc(TotalSales)) %>%
    mutate(Index = 1:n(), Customer.ID = ifelse(Index > 10, "Others", Customer.ID))
```

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

```
#10 largest customers
ggplot(by_cust[!(by_cust$Customer.ID=="Others" | is.na(by_cust$Customer.ID)),], aes(x = reorder(Custome
    ggtitle("Last months' revenue share by customer (10 largest customers)")
```




```
#all customers
ggplot(by_cust, aes(x = reorder(Customer.ID, -TotalSales), y = TotalSales)) +
    geom_bar(stat = "identity", aes(fill = Customer.ID)) +
    ggtitle("Last months' revenue share by customer (all customers)")
```

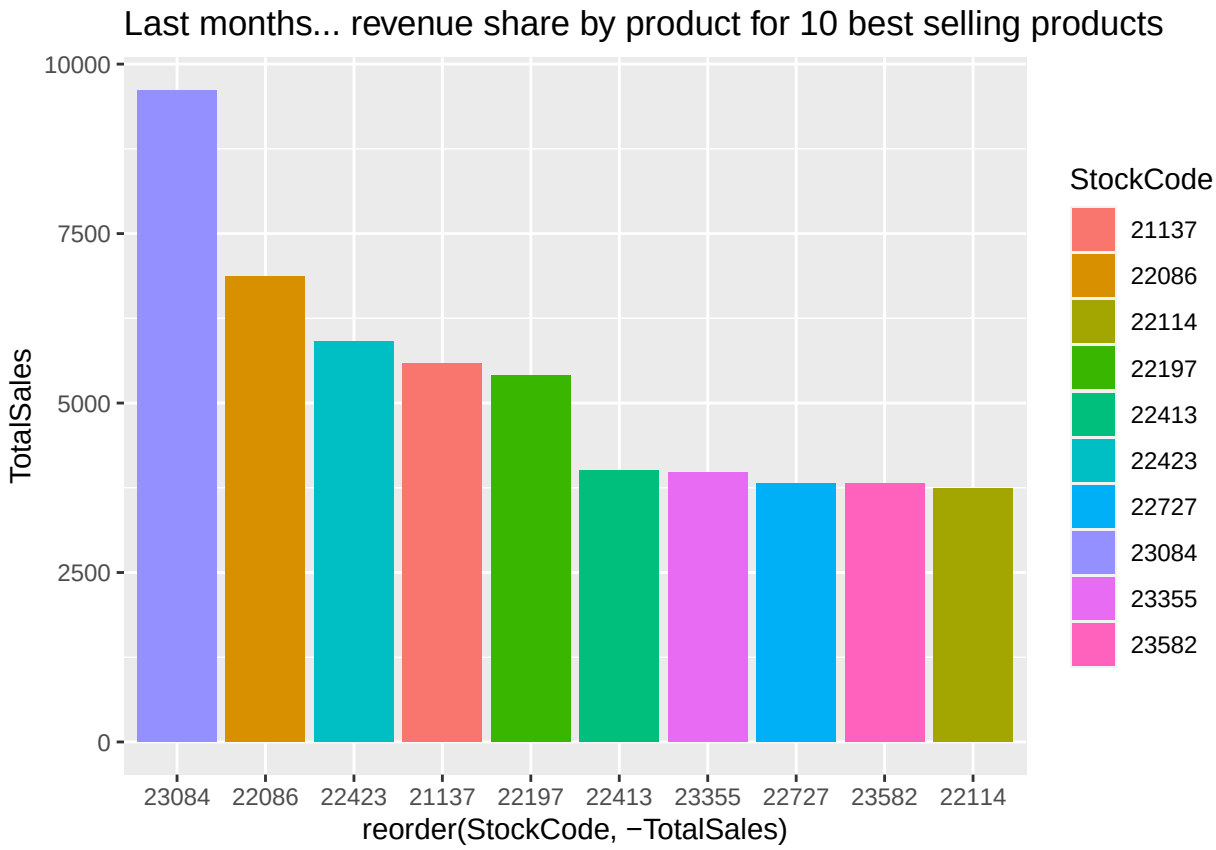


Sales appear to be highly fragmented, there is no one big customer.

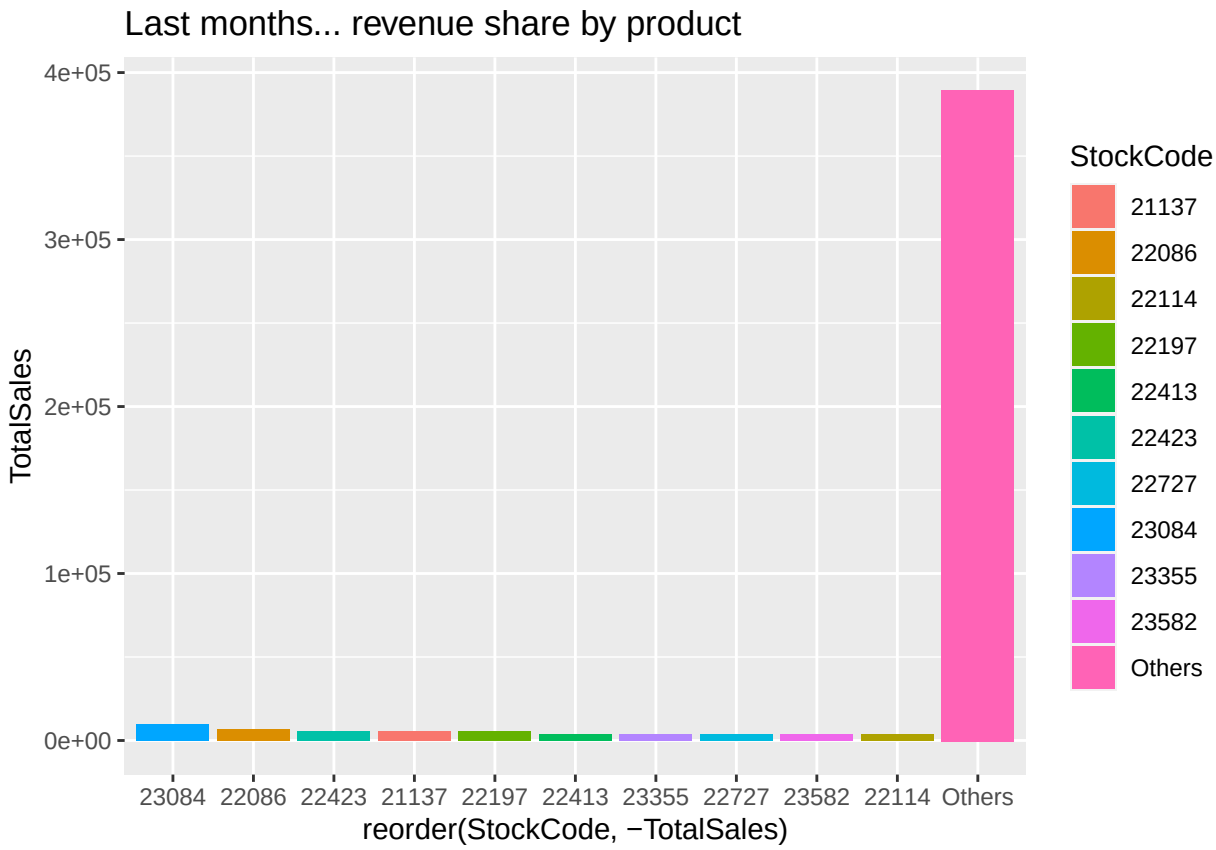
```
by_prod <- lastMonth %>%
    group_by(StockCode) %>%
    summarise(TotalSales=sum(Sales)) %>%
    arrange(desc(TotalSales)) %>% mutate(Index = 1:n(), StockCode = ifelse(Index > 10, "Others", StockC
```

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

```
#10 products with highest revenue share
ggplot(by_prod[!(by_prod$StockCode=="Others"),], aes(x = reorder(StockCode, -TotalSales), y = TotalSale
  geom_bar(stat = "identity", aes(fill = StockCode))+
  ggtitle("Last months' revenue share by product for 10 best selling products")
```




```
#All products
ggplot(by_prod, aes(x = reorder(StockCode, -TotalSales), y = TotalSales)) +
  geom_bar(stat = "identity", aes(fill = StockCode))+
  ggtitle("Last months' revenue share by product")
```




```
sum(by_prod$TotalSales)  #Total sales this month
```

```
## [1] 441393.9
```

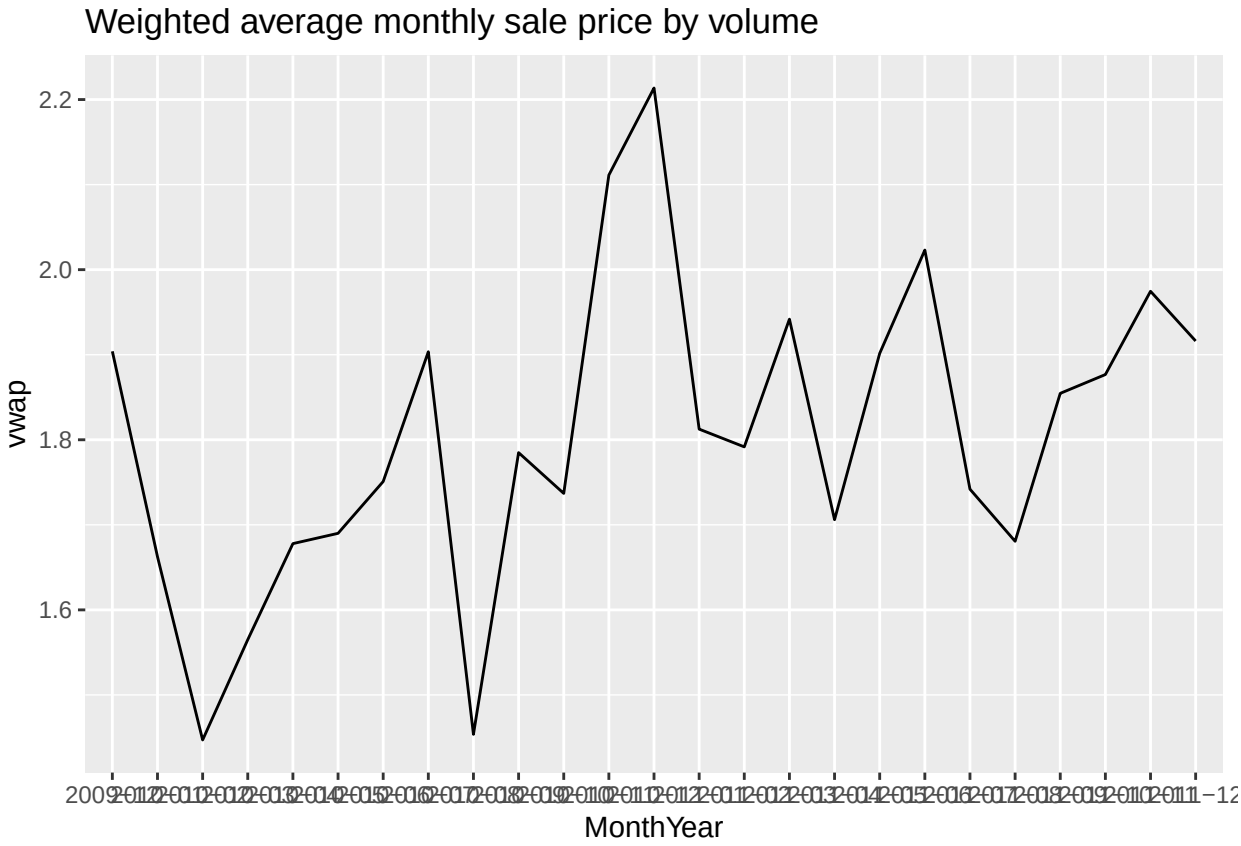
Again, sales appear to be highly fragmented by product type, each having monthly sales between 100-10000 pounds, out of total sales of 441393.9 pounds.

Weighted average monthly sale price by volume

```
retail %>% group_by(MonthYear) %>% summarise(vwap = sum(Sales)/sum(Quantity)) -> VWAP
```

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

```
ggplot(VWAP, aes(MonthYear, vwap, group = 1)) + geom_line() + ggtitle("Weighted average monthly sale pri
```



There is no obvious trend in the weighted average monthly sale price by volume, it fluctuates between 1.60 to 2.20 pounds.

3. You'll note the dataset contains negative volumes, relating to sales returns. Some of these returns relate to products sold before the data collection date, thus should be filtered from the dataset before we use it for modelling. Describe and implement a logical way of performing this task

```
#The code does not appear to be working on markdown running on my laptop, I have instead run the below

retail <-read.csv('/Users/lawrence/Downloads/sales/retail.csv')
# #Select returned items
# retail %>% filter(Sales < 0) %>% select(StockCode,Customer.ID,Sales,Invoice,MonthYear) %>% mutate (Sa
# retail %>% filter(Sales > 0) -> no_returns
#
# #Only consider returns within 3 months after the dataset collection started which is reasonable as mo
# returns <- returns[(returns$MonthYear=='2009-12' | returns$MonthYear=='2010-01' | returns$MonthYear==
#
# #Check if the original purchase with the same metrics (Customer.ID, StockCode and +ive Sales) exists,
# apply(returns,MARGIN = 1, FUN = function(x){
#   if(nrow(no_returns[(no_returns$StockCode==x[1] & no_returns$Customer.ID==as.double(x[2]) & no_retur
#     retail<<-retail[!(retail$Invoice==x[4]),] #Accesses and modifies global variable retail by removi
#   }
# })
```

A logical to filter these returns out would be to match them with the initial sales order. This cannot be found directly from the invoice, therefore we would need to match it by customer id, stockcode, absolute value of sales and roughly the same time period.

a)The owner of the online retailer wants to know how much revenue to expect for this month (12/2011), to help him decide what sports car he buys his partner for Christmas.

Outline a few different solutions you might suggest that solve this problem. Include in your description:

What metrics/values you might want to use:

I would suggest using combined total sales volume, be it monthly, weekly or daily, as the response variable. This is far better, easier and more accurate to model (due to CLT and large numbers) than individual purchases by specific customers or products. Doing so would result in a far more complicated model that will not necessarily result in more accurate predictions, especially considering the highly fragmented nature of sales by individual customers or products. I will therefore only look at aggregate sales volume as a response variable, and use the Date as an explanatory variable to capture the underlying trend, and Month to capture recurring monthly sales patterns.

How you would aggregate those metrics:

The aggregation could be done using the aggregate function, or summarize using tidyverse.

What model/algorithm or logic you would use to make a prediction on them:

I would use the linear model lm() function in r with totalsales (monthly, daily or weekly) as the response and Date and as.factor(Month) as the explanatory variables.

What uncertainties you might need to explore I would then look at confidence intervals, r^2, residuals and other goodness of fit measures to determine if this is a good fit. If not, another option would be to use the time series functionalities in R with ts() or holtwinters() functions, or potentially to use a glm() as maybe a poisson or other distribution is better suited, or taking the log of the response variable might be an option to improve it as well given that it is sales data.

Looking at the monthly patterns in total sales volume observed in the two previous years, December was the month with the second highest sales in 2010, unless there is a strong reason to suggest otherwise I would expect a similar pattern in 2011. Additionally, overall sales growth will need to be taken into account to come up with an accurate estimation for this years December sales. We also have data for this year's December of up to the 9th of December. This could be useful for estimation by comparing it with how the last two years full-December sales data compared with sales up until the 9th of December.

In addition, something like a profit margin, COGS and tax estimation would be helpful to determine if the owner can afford the Ferrari or not as just simple sales revenue does not say too much about the owners' bottom line.

b) Select the method you think is the best approach, and explain why. Your justification should weigh up the effort required and expected accuracy.

For starters, I think the lm() is the best approach to start as it is the most simple and easy to implement one, while likely still yielding reasonably good accuracy.

c) Show an implementation one of your solutions (doesn't need to be your selected method), and show the final forecast alongside the historical time series.

```
aggregate(Sales ~ MonthYear, data=retail, FUN=sum) %>% separate(MonthYear,into=c('Year','Month')) %>% s
sales <- sales[-25,] #Remove last incomplete December observation
monthly.lm <- lm(Sales ~ Month, data=sales)
summary(monthly.lm)
```

```
##
## Call:
```

```
## lm(formula = Sales ~ Month, data = sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -151925  -39145       0   39145  151925
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   598430      59587  10.043 3.42e-07 ***
## Month02       -75703      84269  -0.898 0.386669
## Month03       126128      84269   1.497 0.160297
## Month04       -56536      84269  -0.671 0.514989
## Month05        70898      84269   0.841 0.416615
## Month06        87025      84269   1.033 0.322109
## Month07        29838      84269   0.354 0.729422
## Month08        71298      84269   0.846 0.414063
## Month09       338239      84269   4.014 0.001719 **
## Month10       459506      84269   5.453 0.000147 ***
## Month11       843775      84269  10.013 3.53e-07 ***
## Month12       376091      84269   4.463 0.000775 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 84270 on 12 degrees of freedom
## Multiple R-squared:  0.949,  Adjusted R-squared:  0.9022
## F-statistic:  20.3 on 11 and 12 DF,  p-value: 4.548e-06
```

```
monthly.pred <- predict(monthly.lm, newdata = data.frame(Month='12'), se.fit = T)

lower<-monthly.pred$fit-1.96*monthly.pred$se.fit
upper<-monthly.pred$fit+1.96*monthly.pred$se.fit
cbind(monthly.pred$fit,lower,upper)
```

```
##               lower   upper
## 1 974520.9 857729.8 1091312
```

Just looking at historical monthly averages, we'd expect December sales to lie between 857729.8 and 1091312 pounds with 95% confidence.

```
daily.sales <- retail %>%
  select(Sales, MonthYear, Date) %>%
  separate(MonthYear,into=c('Year','Month')) %>%
  select(-c(Year)) %>%
  group_by(Date) %>%
  summarise(DailySales=sum(Sales), Month) #Creates Month variable and aggregates daily sales
```

```
## `summarise()` regrouping output by 'Date' (override with `.groups` argument)
```

```
daily.sales <- unique(daily.sales) #Remove redundant rows
daily.sales$Date<- as.Date(daily.sales$Date) #Convert date to Date object
daily.sales<-daily.sales[-604,] #Exclude the last day as sales don't appear to be complete for that day
```

```r
daily.lm<- lm(DailySales ~ Date + Month, data=daily.sales)
summary(daily.lm)
```

```
##
## Call:
## lm(formula = DailySales ~ Date + Month, data = daily.sales)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45880  -7574  -1019   5934  72140
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -63907.183  48197.721  -1.326  0.18537
## Date             5.999      3.251   1.845  0.06551 .
## Month02      -3322.281   3089.215  -1.075  0.28262
## Month03       1554.431   3006.742   0.517  0.60536
## Month04       -783.133   3167.975  -0.247  0.80484
## Month05       1650.536   3097.753   0.533  0.59436
## Month06        533.205   3066.587   0.174  0.86202
## Month07      -1848.220   3083.735  -0.599  0.54917
## Month08       -436.114   3103.992  -0.141  0.88831
## Month09       9642.432   3128.046   3.083  0.00215 **
## Month10      14124.396   3154.274   4.478 9.05e-06 ***
## Month11      28722.031   3183.312   9.023  < 2e-16 ***
## Month12      23803.944   3090.429   7.702 5.67e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15130 on 590 degrees of freedom
## Multiple R-squared:  0.3323, Adjusted R-squared:  0.3187
## F-statistic: 24.47 on 12 and 590 DF,  p-value: < 2.2e-16
```

Looking at this model, we can see that there appears to be a (weak) sales growth trend, on average sales tend to increase by about 5 pounds on every calendar day. This trend is not very significant however (P-value = 0.08). Using this model to predict remaining sales in December:

```r
month<-c(rep('12',23))
date<-c('2011-12-09','2011-12-10','2011-12-11','2011-12-12','2011-12-13','2011-12-14','2011-12-15','201
dec<-data.frame(date,month)
names(dec) <- c('Date', 'Month')
dec$Date<-as.Date(dec$Date)

daily.pred <- predict(daily.lm, newdata = dec, se.fit = T)

lower<-daily.pred$fit-1.96*daily.pred$se.fit
upper<-daily.pred$fit+1.96*daily.pred$se.fit


prev.sales<- retail %>% filter((Date>'2011-11-30' & Date < '2011-12-09')) %>% summarise(TotalSales = su
prev.sales<-prev.sales[[1]]

cbind(sum(daily.pred$fit)+prev.sales,sum(lower)+prev.sales,sum(upper)+prev.sales) #Predictions for tota
```

```
##          [,1]    [,2]    [,3]
## [1,] 1594171 1473128 1715213
```

```
last.dec<- retail %>% filter((Date>'2010-11-30' & Date < '2011-01-01')) %>% summarise(TotalSales = sum(S
last.dec[[1]]
```

```
## [1] 1126445
```

This model taking into account the growth trend would predict total sales to lie between 1461205 and 1699748 pounds. This figure seems quite high, especially considering last years total December sales were only 1126445. The model is likely putting too much weight on the sales increase from December 2009 to December 2010.

```
this.dec.frac <-retail %>% filter((Date>'2011-11-30' & Date < '2011-12-09')) %>% summarise(TotalSales =
last.dec.frac <-retail %>% filter((Date>'2010-11-30' & Date < '2010-12-09')) %>% summarise(TotalSales =
this.dec.frac[[1]]  #Total sales this year in the first eight December days
```

```
## [1] 401536.5
```

```
last.dec.frac[[1]]  #Total sales last year in the first eight December days
```

```
## [1] 649912.6
```

```
frac<- last.dec.frac[[1]]/last.dec[[1]] #Fraction of total December sales in the first eight days
frac
```

```
## [1] 0.5769588
```

```
this.dec.frac[[1]]/frac   #Extrapolating this percentage on this years first eight days sales.
```

```
## [1] 695953.5
```

Another interesting point is that total sales in the first eight days of December last year were about 650,000 pounds, while this year this figure is only at about 401,000 pounds. Given that last year 57% of December sales occured in these first eight days, this may be cause for concern in accuracy of the model, especially the second one. Looking purely at these numbers, we would only expect about 700,000 pounds of total December sales this year. However, this might just be a coincidence, maybe Christmas shopping is just left a bit late in 2011 compared to 2010.

How confident are you of this forecast - do you back your prediction enough to recommend the new Ferrari?

Based on these estimates, I would give a conservative estimate of Sales to be between 700,000 to 900,000 pounds. Depending on the owners profit margin, this could be enough for the Ferrari, but to be on the cautious side maybe a Toyota would be a better choice for this year.