playground_jan2022

#reading in the files

```
train_data = read.csv("train.csv")
test_data = read.csv("test.csv")
submission = read.csv("sample_submission.csv")
gdp_data = read.csv("GDP_data_2015_to_2019_Finland_Norway_Sweden.csv")
```

```
summary(train_data)
```

```
##      row_id          date              country             store          
##  Min.   :    0   Length:26298       Length:26298       Length:26298      
##  1st Qu.: 6574   Class :character   Class :character   Class :character  
##  Median :13148   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :13148                                                           
##  3rd Qu.:19723                                                           
##  Max.   :26297                                                           
##    product             num_sold      
##  Length:26298       Min.   :  70.0  
##  Class :character   1st Qu.: 190.0  
##  Mode  :character   Median : 315.0  
##                     Mean   : 387.5  
##                     3rd Qu.: 510.0  
##                     Max.   :2884.0
```

```
#Looking at the summary statistics, the things I need to change is the date to datetime and then the rest of the independent variables into categorical variables
```

#fixing the columns for the train set

```
train_data$date = ymd(train_data$date)
train_data$country = as.factor(train_data$country)
train_data$store = as.factor(train_data$store)
train_data$product = as.factor(train_data$product)
train_data$month = month(train_data$date)
```

#feature engineering on train set

```
#1. adding a column representing day of the week
dayofweek = weekdays(train_data$date)

#2. adding a column that represents if it's a weekend or weekday, 1 represents weekend and 0 rep
resents weekday
weekend=ifelse(dayofweek == "Sunday",1,ifelse(dayofweek=="Saturday",1,0))

#3. adding a column that represents GDP for country for given years

#My approach to achieving this: A left join on both the year and country column

#a. create a year column
year = year(train_data$date)

#a2. we need to concatenate the first 3 columns with the train set first
train_data1=cbind(train_data,dayofweek,weekend,year)

#b. change the column names of the gdp dataframe
colnames(gdp_data) = c("year","Finland","Norway","Sweden")

#c. pivot the dataframe into the long form
gdp_longer = gdp_data %>% pivot_longer(Finland:Sweden,names_to = "country",values_to = "GDP")

#d. left join to make the final dataframe
train_final = train_data1 %>% left_join(gdp_longer,by=c("year","country"))
```

#repeat the entire process with the testing set

#fixing the test set variables

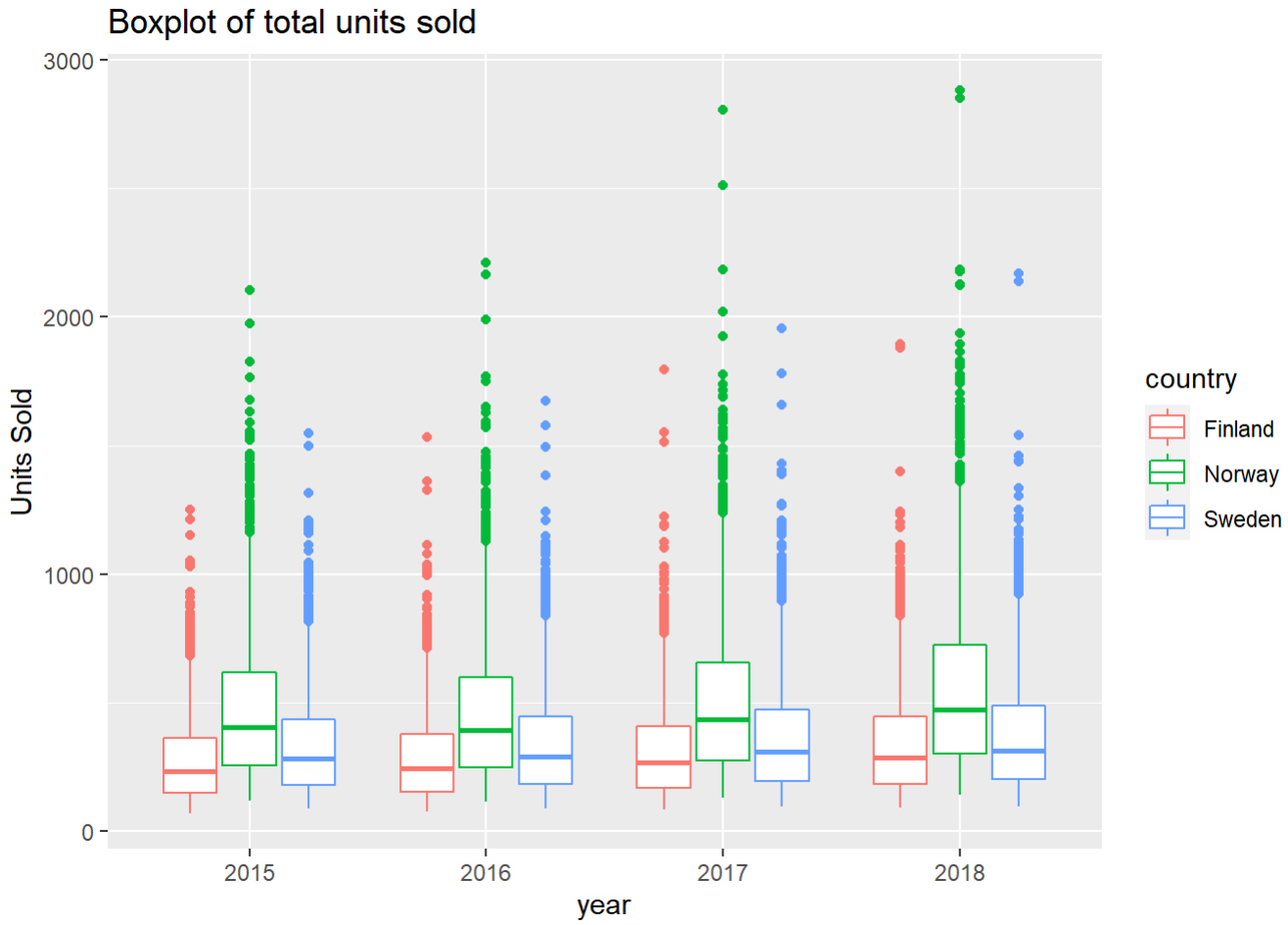
```
test_data$date = ymd(test_data$date)
test_data$country = as.factor(test_data$country)
test_data$store = as.factor(test_data$store)
test_data$product = as.factor(test_data$product)
test_data$month = month(test_data$date)
```

#feature engineering on test set

```
#only difference from train set is that we dont have to deal with the transformation of the gdp
 dataframe
dayofweek = weekdays(test_data$date)

weekend=ifelse(dayofweek == "Sunday",1,ifelse(dayofweek=="Saturday",1,0))


year = year(test_data$date)

test_data1=cbind(test_data,dayofweek,weekend,year)


test_final = test_data1 %>% left_join(gdp_longer,by=c("year","country"))
```

#Data visualization: Only for train set because we don't have response variable for test set.

```
train_final %>% mutate(year = factor(year),country = as.factor(country)) %>% ggplot(aes(year,num_sold,colour=country)) +geom_boxplot() + ylab("Units Sold") + ggtitle("Boxplot of total units sold")
```



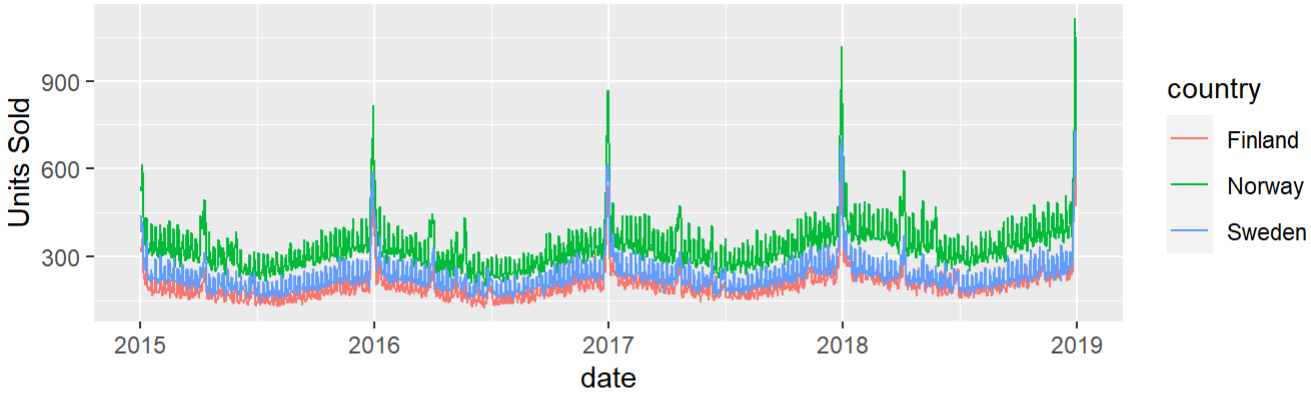

```
#This boxplot shows that more units are being sold every year
```

#Line plot of the number of units sold group by country, store, and product.

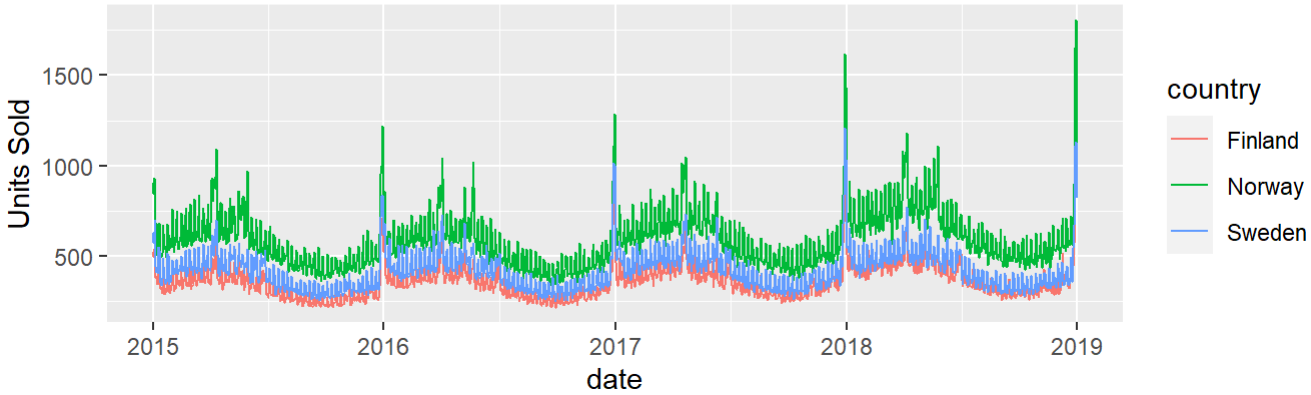
```
products = c("Kaggle Mug","Kaggle Hat","Kaggle Sticker")
stores = c("KaggleMart","KaggleRama")

params = expand.grid(products = products,stores = stores)

lineplot_list = list()

for(i in 1:nrow(params)){
  lineplot_list[[i]] = train_final %>% mutate(country = as.factor(country)) %>% filter(product==
params[i,1] & store==params[i,2]) %>% ggplot(aes(date,num_sold)) +geom_line(aes(color=country),l
wd=0.5) + ylab("Units Sold") + ggtitle(paste0("Line plot of ",params[i,1],"s sold at ",params[i,
2])) + theme(plot.title = element_text(size = 10))

}

marrangeGrob(lineplot_list,nrow=2,ncol=1)
```

page 1 of 3

Line plot of Kaggle Mugs sold at KaggleMart

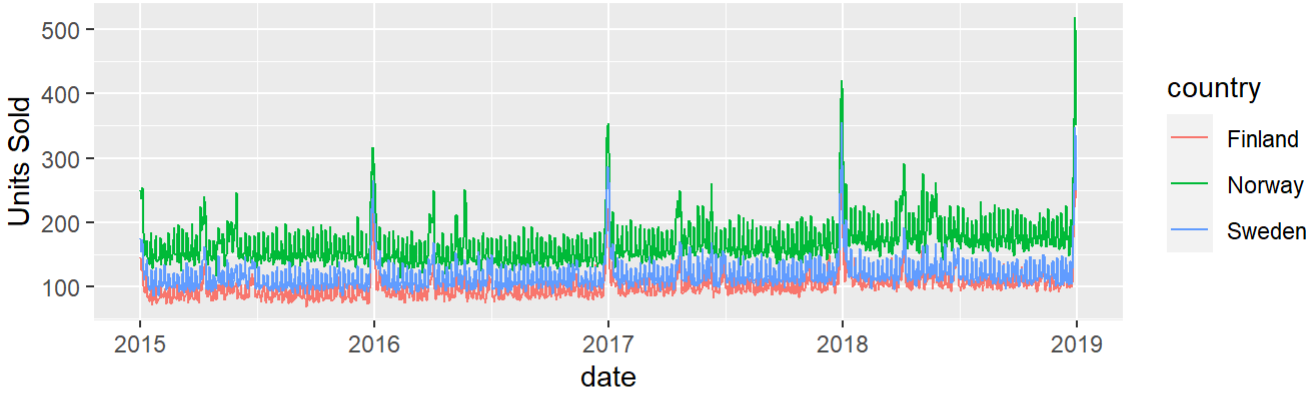


Line plot of Kaggle Hats sold at KaggleMart

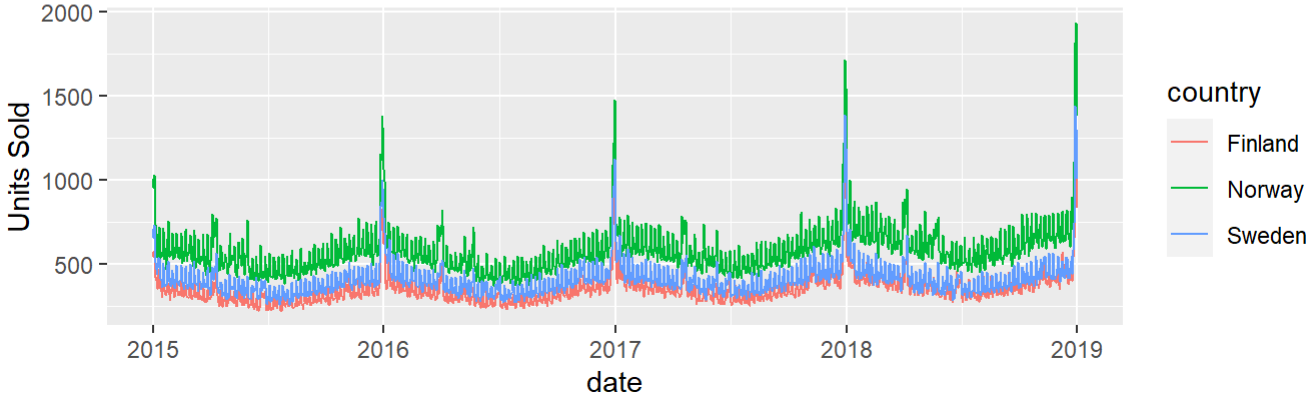


page 2 of 3

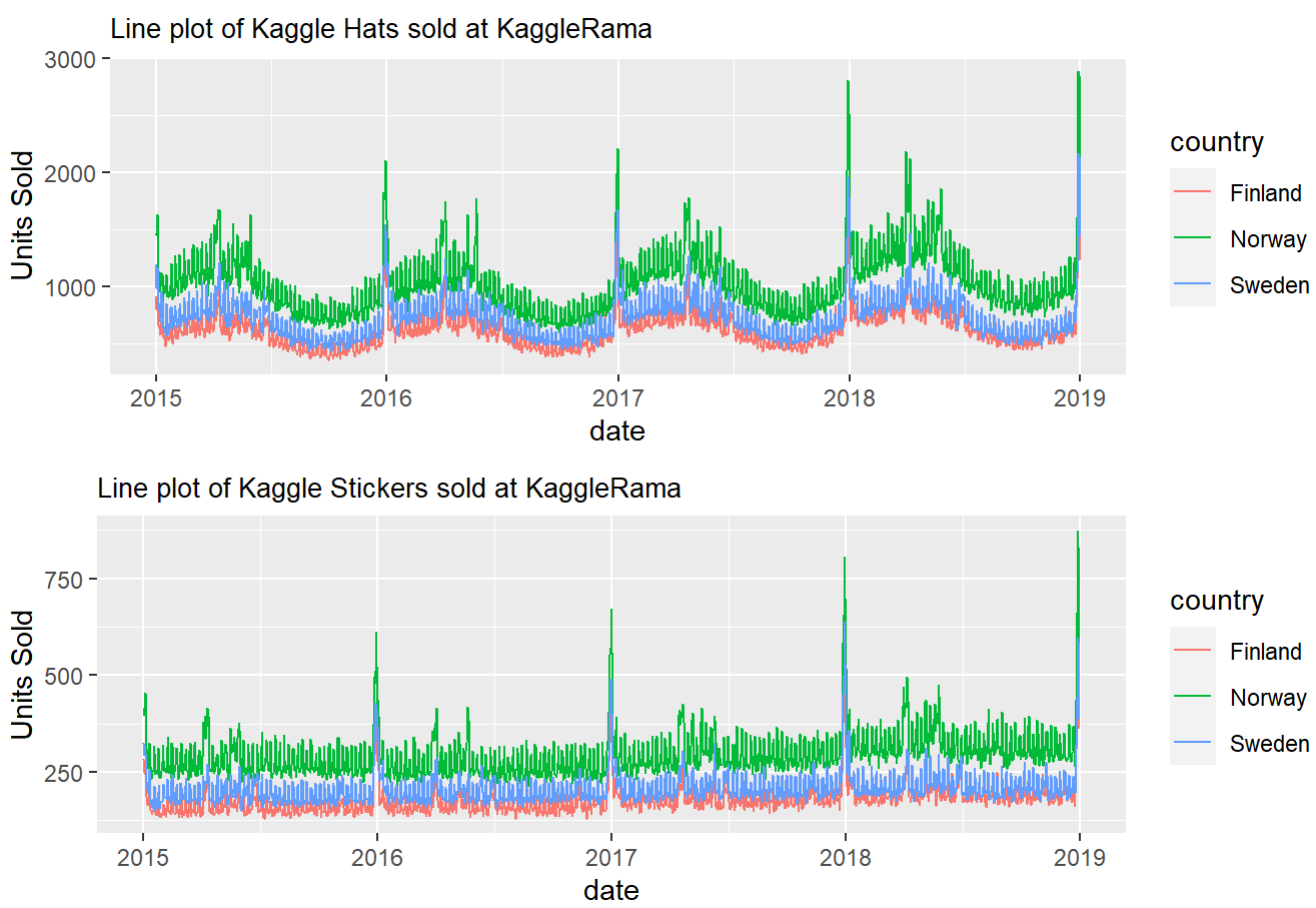
Line plot of Kaggle Stickers sold at KaggleMart



Line plot of Kaggle Mugs sold at KaggleRama



page 3 of 3

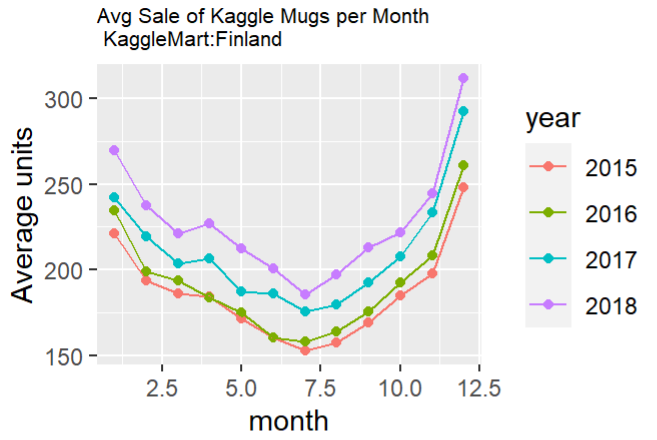


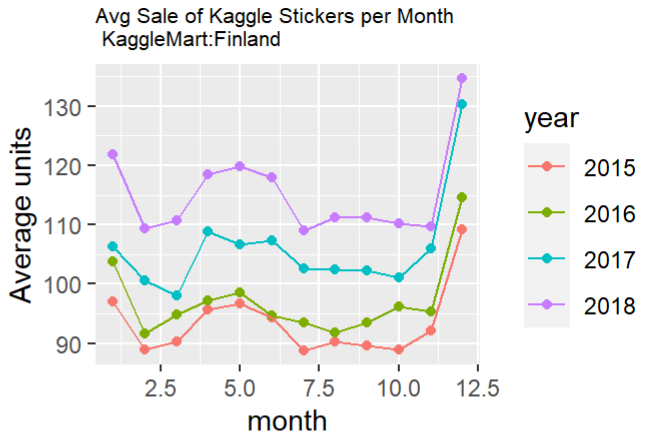

```
#Looking at these plots we can see that there's definitely seasonality, we can zoom in for each
 product to get a clearer picture.
```

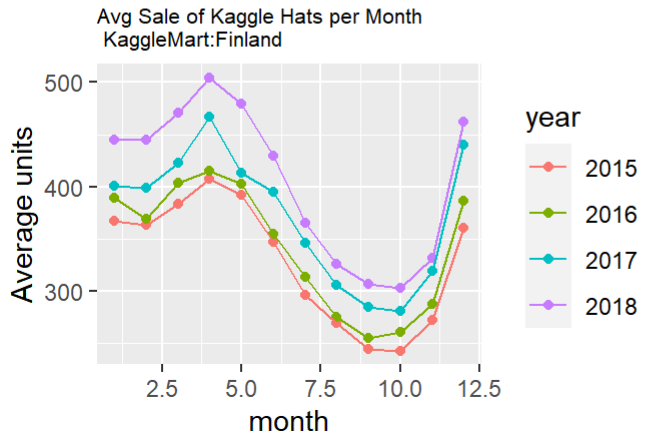
#Zoomed in line plot for average unit

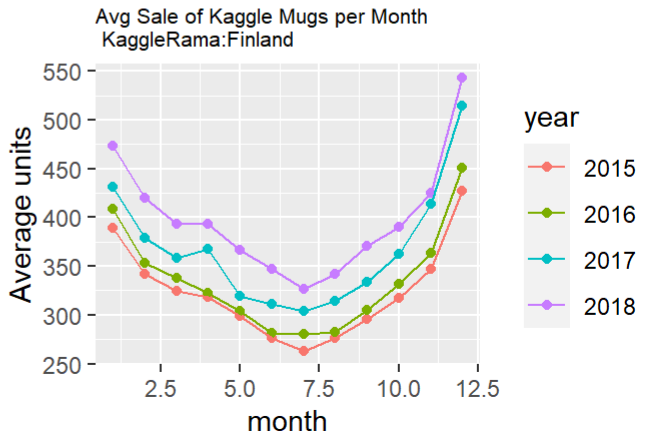
```
products = c("Kaggle Mug","Kaggle Hat","Kaggle Sticker")
stores = c("KaggleMart","KaggleRama")
country = c("Finland","Norway","Sweden")

params = expand.grid(product = products,store = stores,country = country)

another_list = list()

for(i in 1:nrow(params)){

another_list[[i]] = train_final %>% filter(product == params[i,1] & store== params[i,2] & countr
y ==params[i,3]) %>% group_by(month,year) %>% summarise(avg_sales = mean(num_sold)) %>% mutate(y
ear = as.factor(year)) %>% ggplot(aes(month,avg_sales,color=year)) + geom_point() + geom_line()
 + ylab("Average units") +ggtitle(paste0("Avg Sale of ",params[i,1],"s per Month \n ",params[i,2
],":",params[i,3])) + theme(plot.title = element_text(size = 8))
}

marrangeGrob(another_list,nrow=2,ncol=2)
```

page 1 of 5

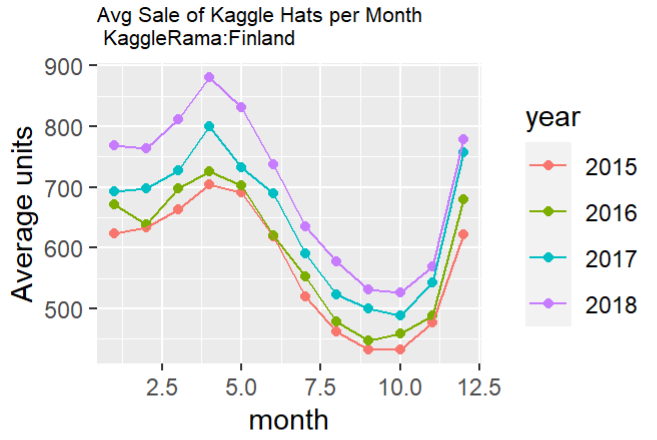


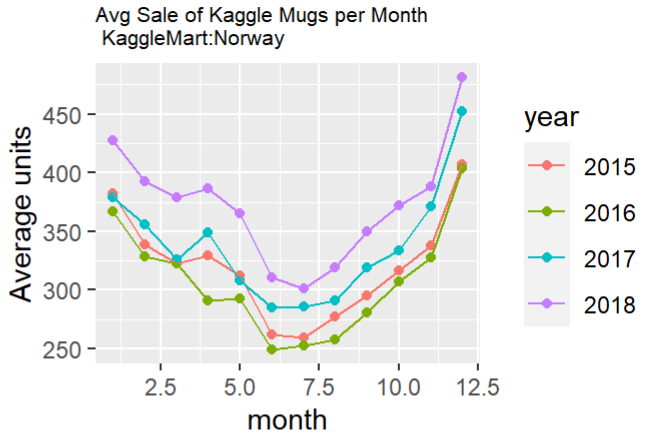


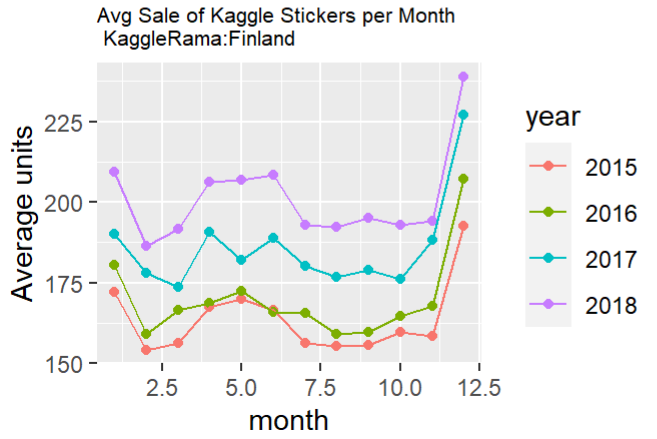


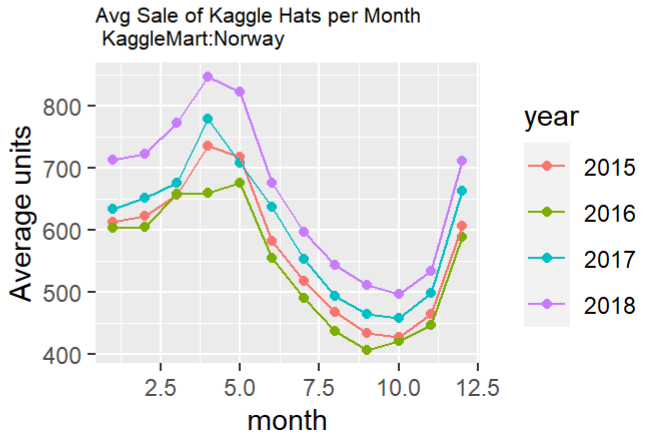


page 2 of 5

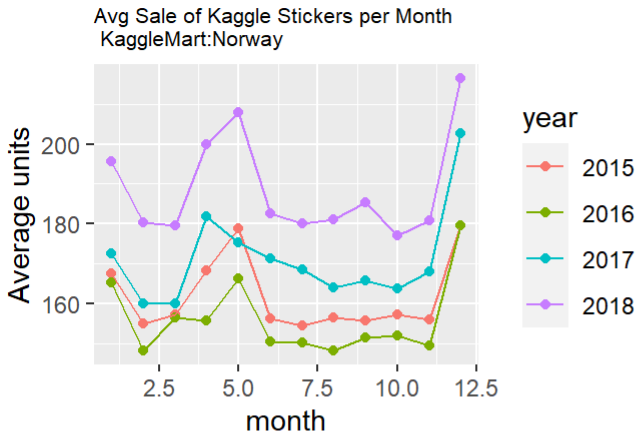


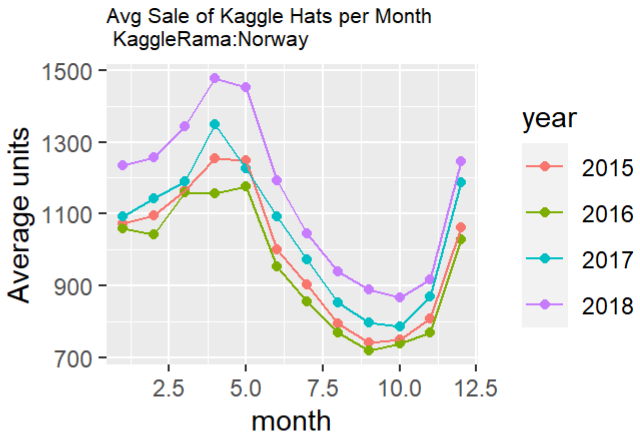


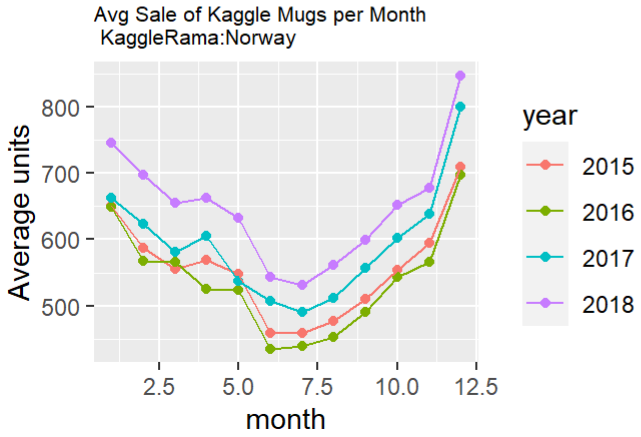


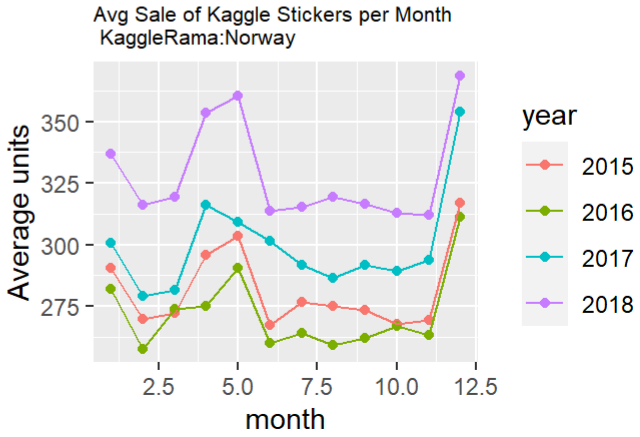


page 3 of 5

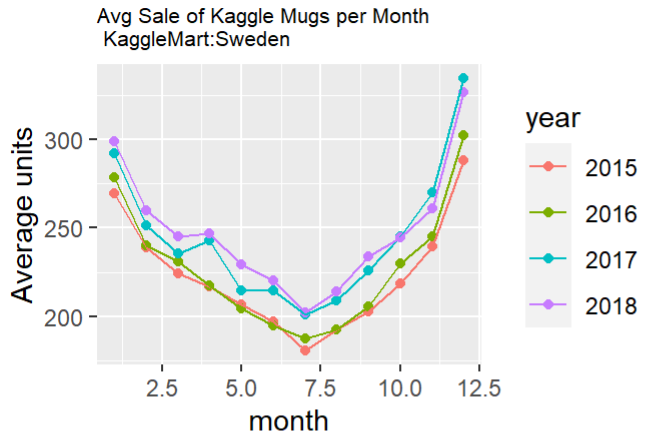

Avg Sale of Kaggle Stickers per Month
KaggleMart:Norway
Average units
200
180
160
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

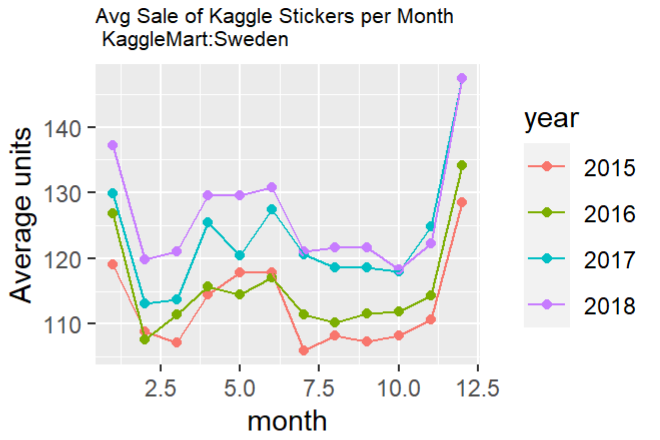

Avg Sale of Kaggle Hats per Month
KaggleRama:Norway
Average units
1500
1300
1100
900
700
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

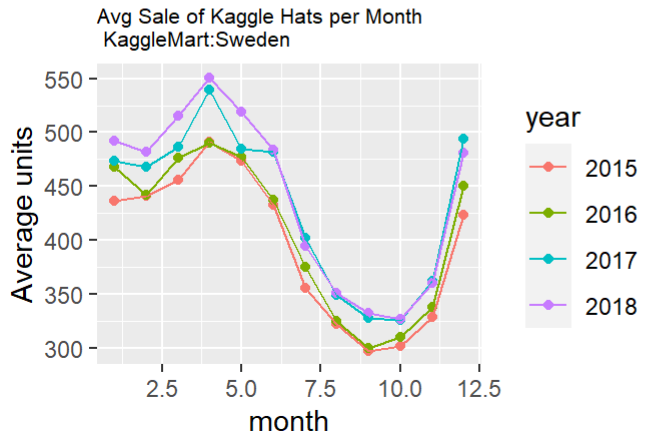

Avg Sale of Kaggle Mugs per Month
KaggleRama:Norway
Average units
800
700
600
500
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

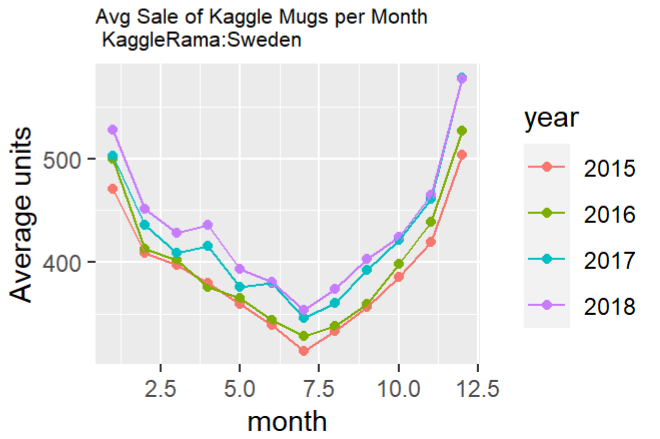

Avg Sale of Kaggle Stickers per Month
KaggleRama:Norway
Average units
350
325
300
275
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

page 4 of 5

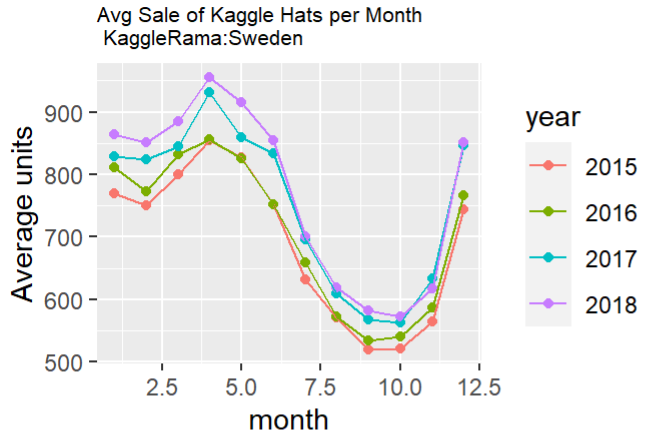


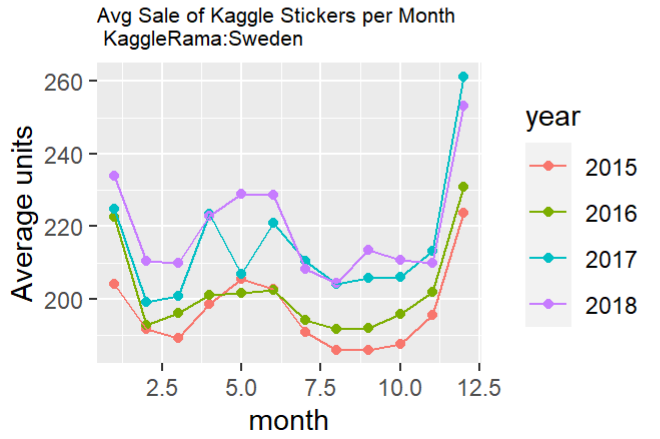






page 5 of 5



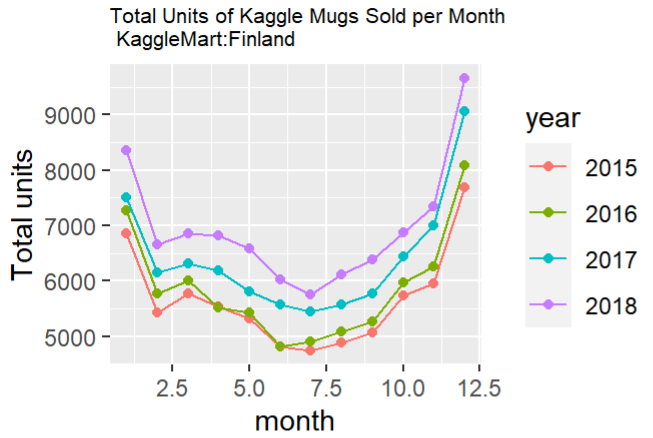


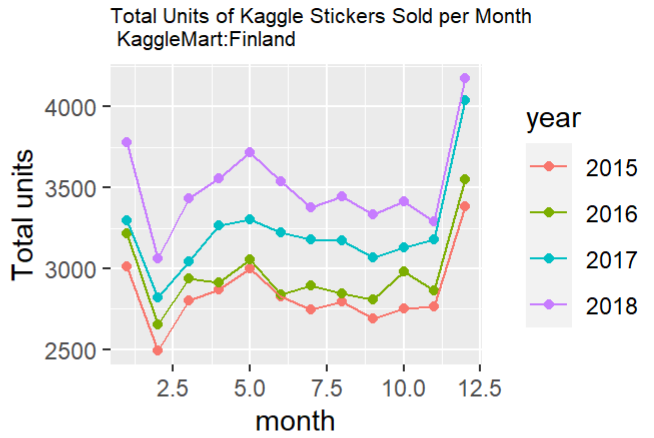
```
#I can see that there's a different seasonality only for the different products, the country and store doesn't change the seasonality.
```

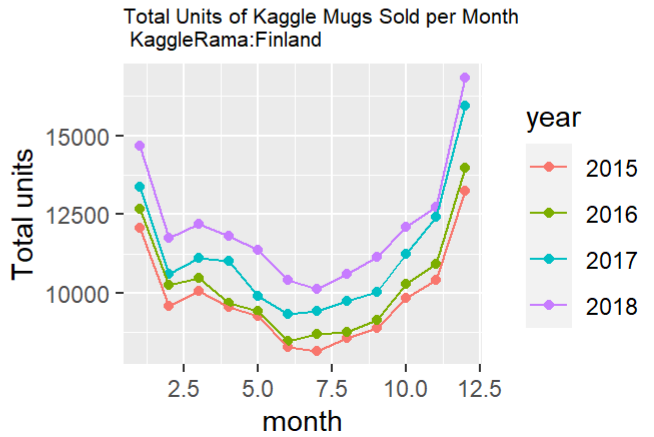
#Zoomed in line plot for total unit

```
another_list2 = list()

for(i in 1:nrow(params)){

another_list2[[i]] = train_final %>% filter(product == params[i,1] & store== params[i,2] & country ==params[i,3]) %>% group_by(month,year) %>% summarise(avg_sales = sum(num_sold)) %>% mutate(year = as.factor(year)) %>% ggplot(aes(month,avg_sales,color=year)) + geom_point() + geom_line() + ylab("Total units") +ggtitle(paste0("Total Units of ",params[i,1],"s Sold per Month \n ",params[i,2],":",params[i,3])) + theme(plot.title = element_text(size = 8))
}

marrangeGrob(another_list2,nrow=2,ncol=2)
```

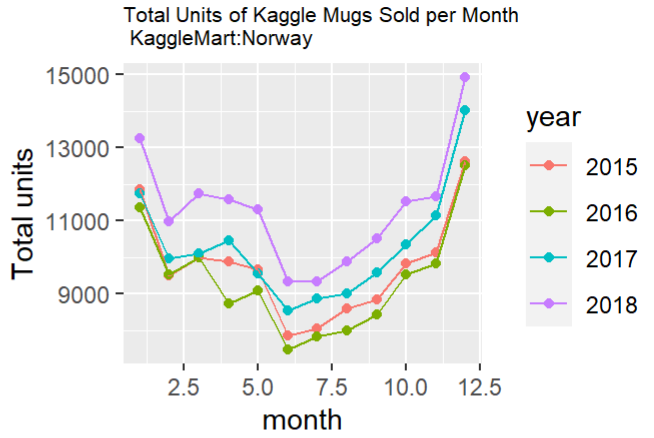
page 1 of 5

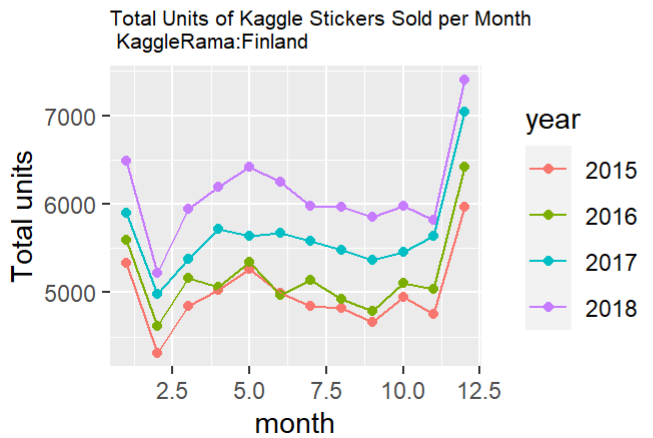




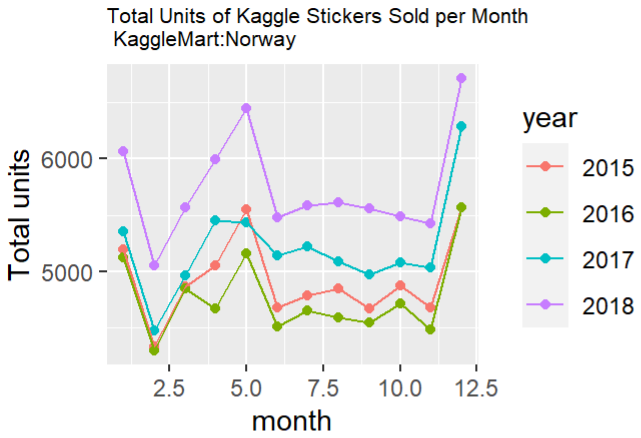


page 2 of 5

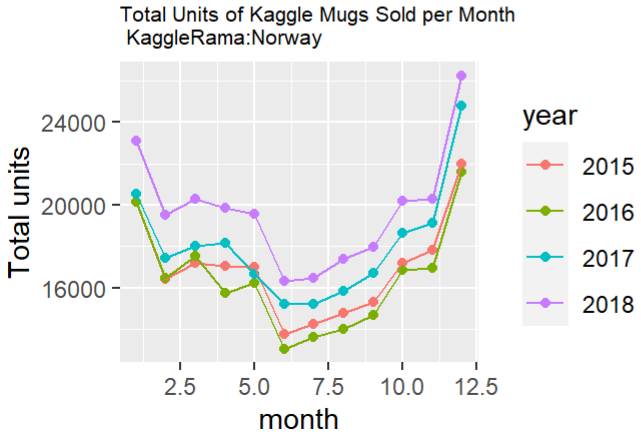


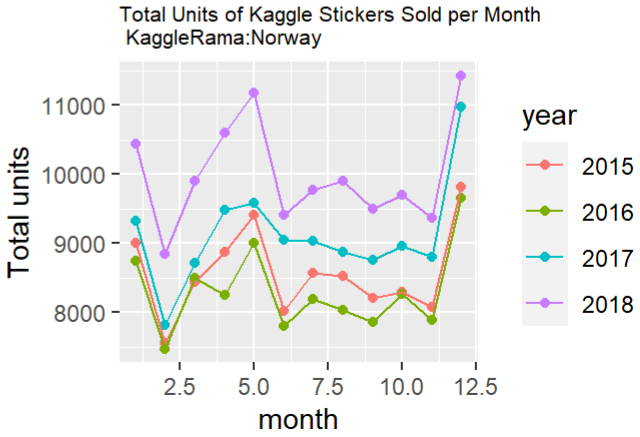


page 3 of 5

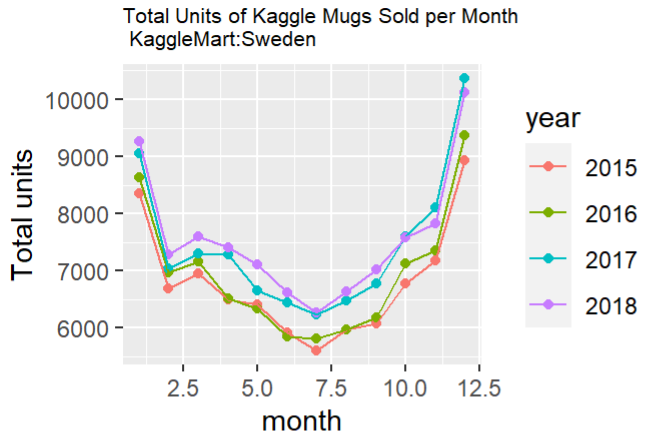

Total Units of Kaggle Stickers Sold per Month
KaggleMart:Norway
Total units
6000
5000
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

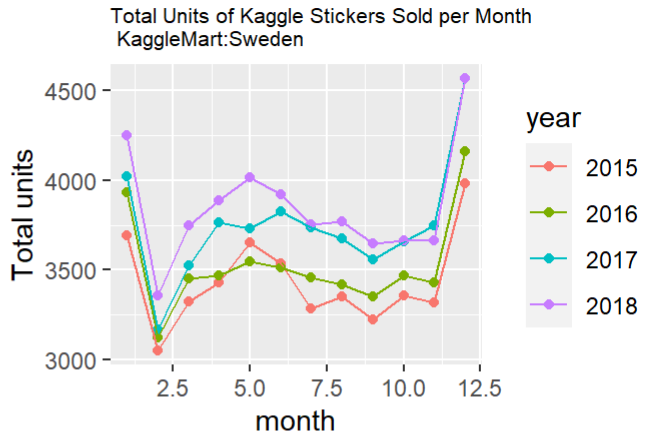

Total Units of Kaggle Hats Sold per Month
KaggleRama:Norway
Total units
45000
40000
35000
30000
25000
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

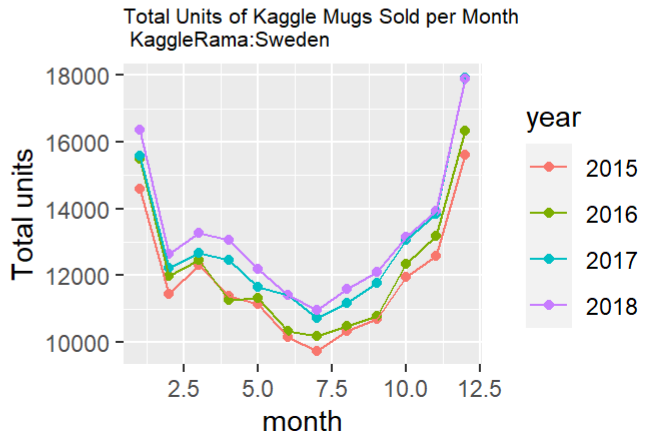

Total Units of Kaggle Mugs Sold per Month
KaggleRama:Norway
Total units
24000
20000
16000
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018


Total Units of Kaggle Stickers Sold per Month
KaggleRama:Norway
Total units
11000
10000
9000
8000
2.5
5.0
7.5
10.0
12.5
month
year
2015
2016
2017
2018

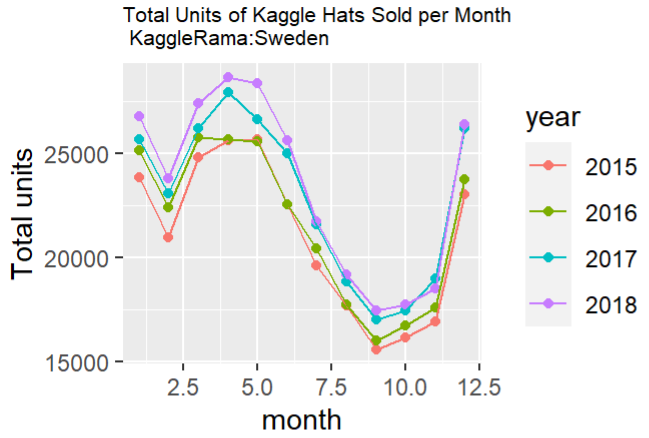
page 4 of 5

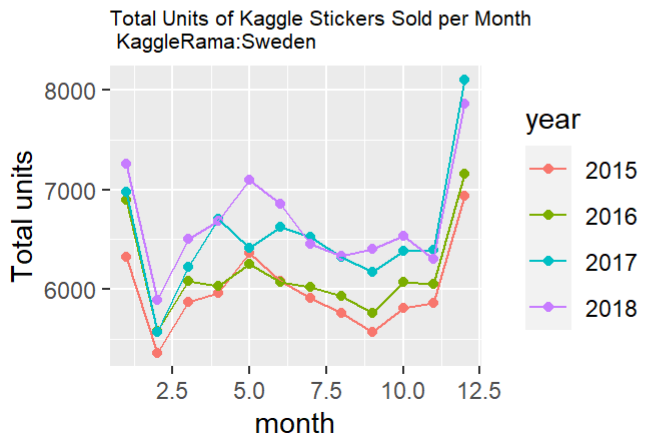






page 5 of 5





```
#These plots also show that there's a different seasonality for differing products. Now because
 there's seasonality, I think it's worthwhile to find out if there's any relevant holidays to in
clude in the feature engineering process.
```