Food Delivery Sales Analysis

```
library(readr)
library(ggplot2)
library(hms)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
rm(list=ls())
```

```
df<- read_csv("sample_orders.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   time_opened = col_datetime(format = ""),
##   customer_id = col_double(),
##   subtotal = col_double()
## )
```

```
#Cleanup
str(df)
```

```
## spec_tbl_df [2,074 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ time_opened: POSIXct[1:2074], format: "2021-03-21 14:59:00" "2021-03-21 14:57:00" ...
##  $ customer_id: num [1:2074] 6.82e+18 -3.80e+18 1.70e+18 -5.97e+18 -7.49e+16 ...
##  $ subtotal   : num [1:2074] 1425 1945 3545 1350 3195 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   time_opened = col_datetime(format = ""),
##   ..   customer_id = col_double(),
##   ..   subtotal = col_double()
##   .. )
```

```
head(df)
```

```
## # A tibble: 6 x 3
##   time_opened         customer_id subtotal
##   <dttm>                    <dbl>    <dbl>
## 1 2021-03-21 14:59:00     6.82e18     1425
## 2 2021-03-21 14:57:00    -3.80e18     1945
## 3 2021-03-21 14:44:00     1.70e18     3545
## 4 2021-03-21 14:03:00    -5.97e18     1350
## 5 2021-03-21 14:01:00    -7.49e16     3195
## 6 2021-03-21 14:02:00    -7.49e18     8400
```

```
#summary(df$time_opened)
df_2<-df[!is.na(df$time_opened),]
summary(df_2$time_opened)
```

```
##                  Min.               1st Qu.                Median
## "2021-01-23 16:05:00" "2021-02-15 13:57:00" "2021-02-28 12:25:30"
##                  Mean               3rd Qu.                  Max.
## "2021-02-27 06:59:54" "2021-03-11 18:31:15" "2021-03-21 14:59:00"
```

```
summary(df_2$customer_id)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -9.189e+18 -4.712e+18 -3.805e+17 -1.670e+17  4.302e+18  9.209e+18
```

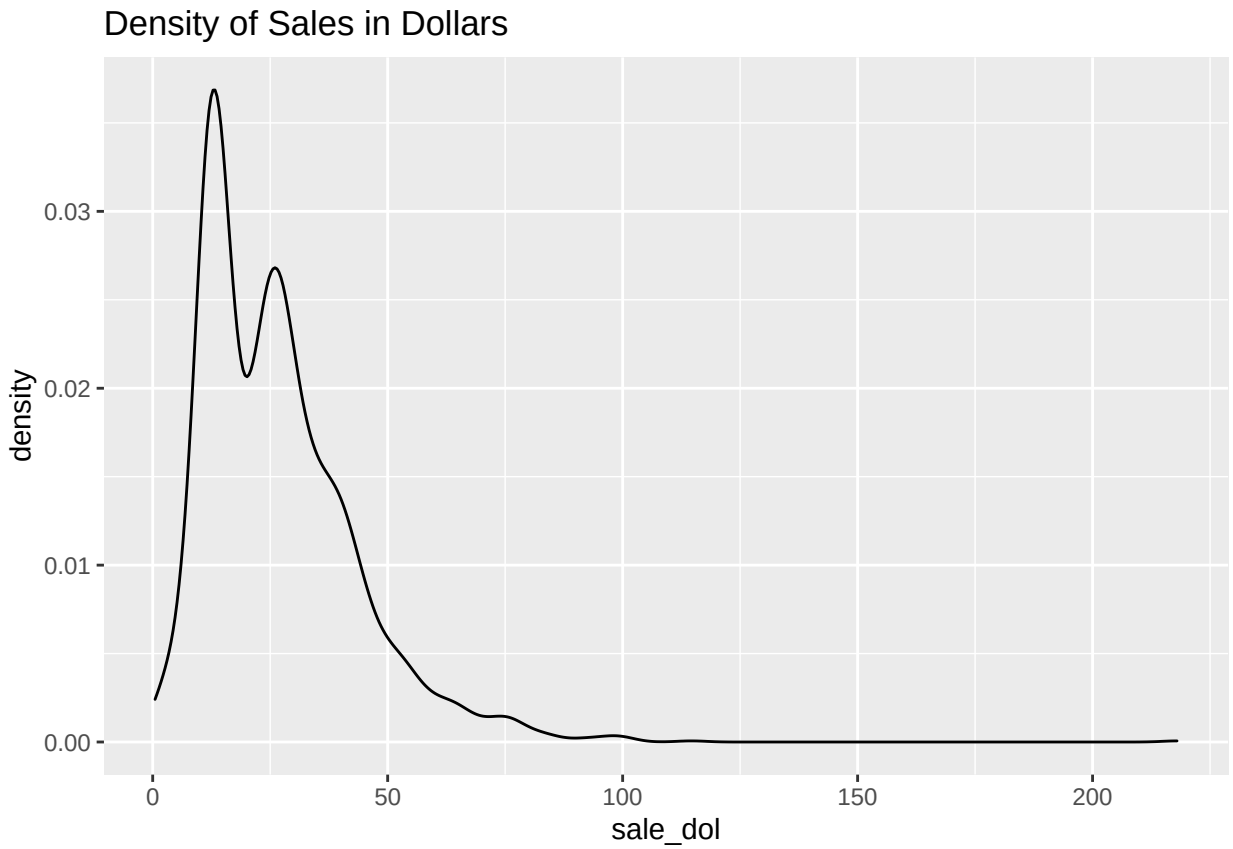
```
summary(df_2$subtotal)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1350    2442    2681    3523   21797
```

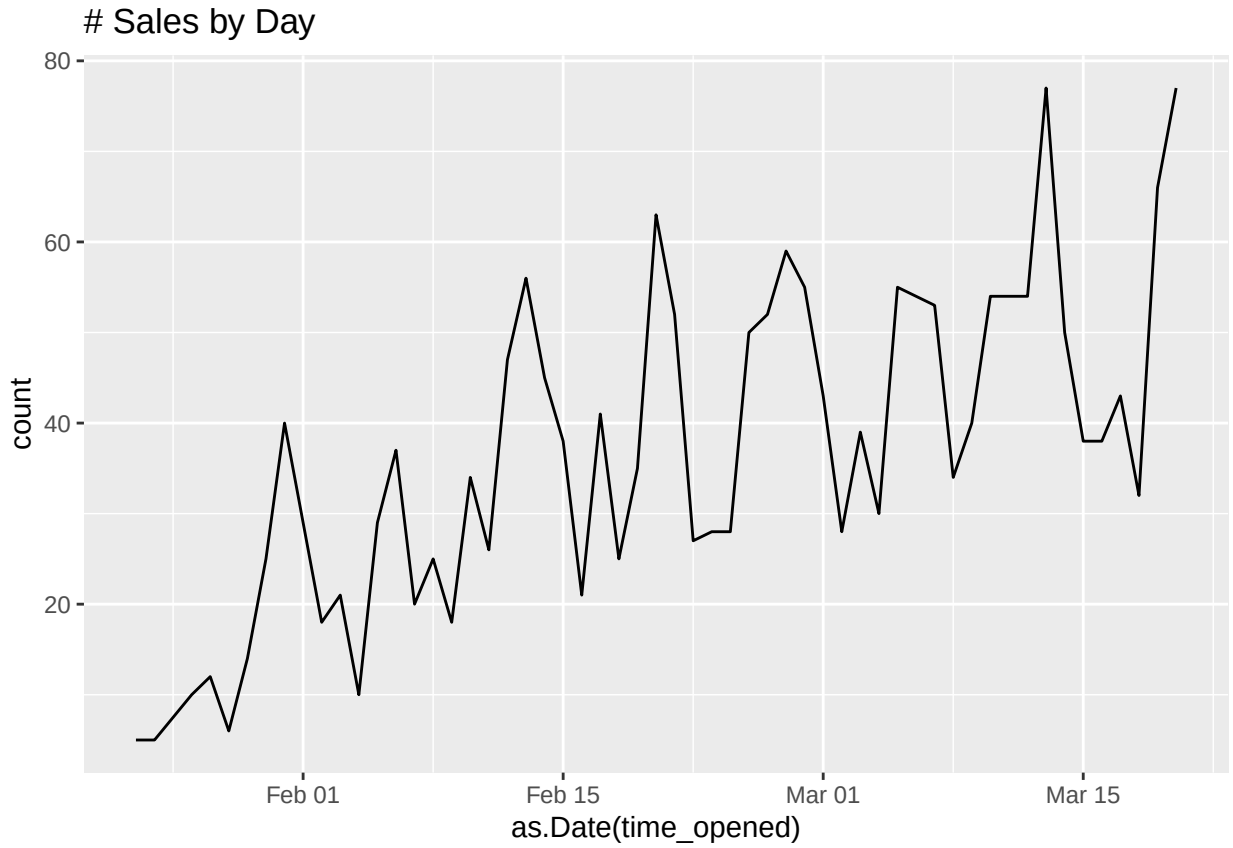
```
df_3<-df_2[df_2$subtotal!=0,]

df_3$customer_id<-as.factor(df_3$customer_id)

#Remove partial day
df_4<-df_3[as.Date(df_3$time_opened)<'2021-03-21',]

df_f<-df_4
```

```
#Feature Construction
df_f$sale_dol<-df_f$subtotal/100
df_f$dow<-weekdays(as.Date(df_f$time_opened))
df_f$time_only<-as.numeric(as_hms(df_f$time_opened))/3600
```

```
#EDA
ggplot(data=df_f,aes(x=sale_dol))+
         geom_density()+
  ggtitle("Density of Sales in Dollars")
```



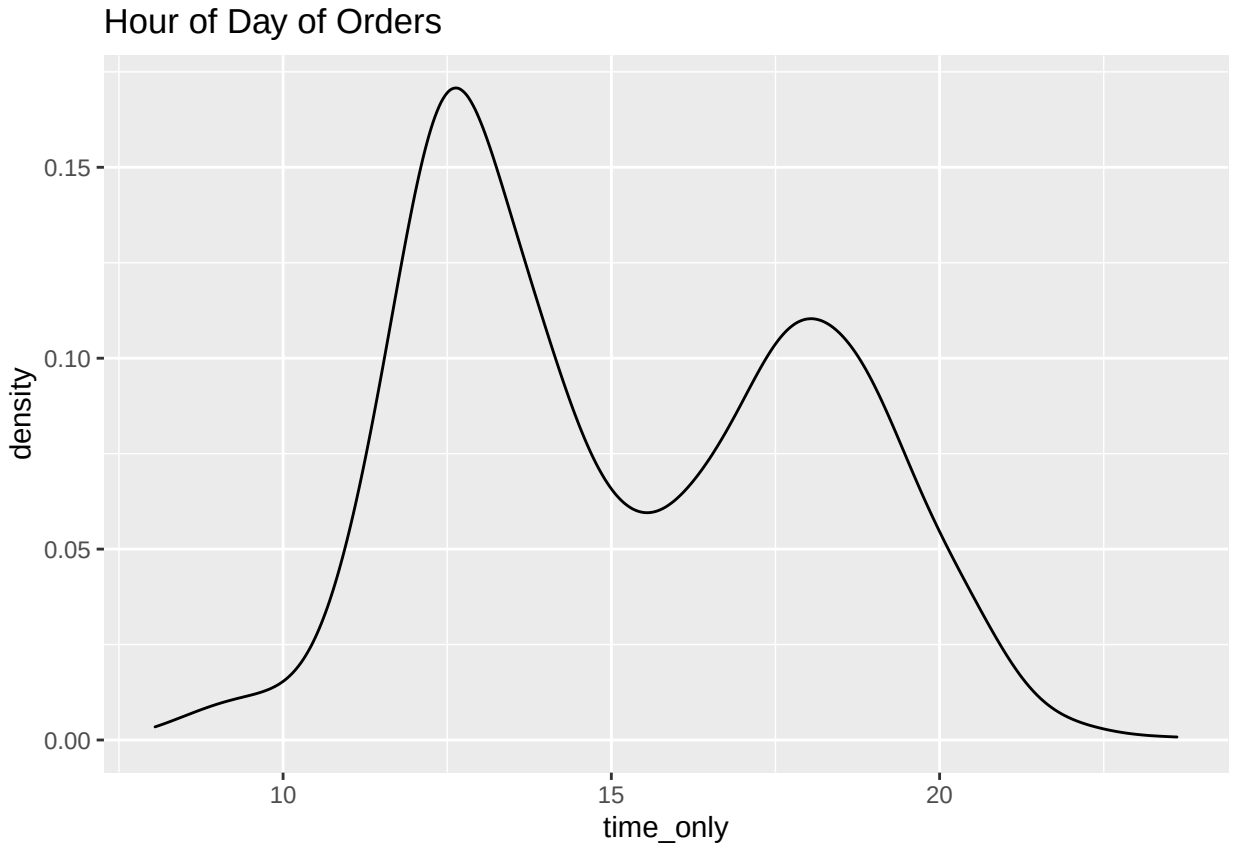

```
ggplot(data=df_f,aes(x=as.Date(time_opened)))+
         geom_line(stat='count')+
  ggtitle("# Sales by Day")
```




```
#table(as.Date(df_f$time_opened))

table(df_f$dow)
```

```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##       352       205       453       320       233       201       272
```

```
ggplot(data=df_f,aes(x=time_only))+
  geom_density()+
  ggtitle("Hour of Day of Orders")
```



Question 1

The prompt calls for an email style message. For time purposes I will go with bullet points and potentially circle back to an email format time permitting.

Early EDA Insights

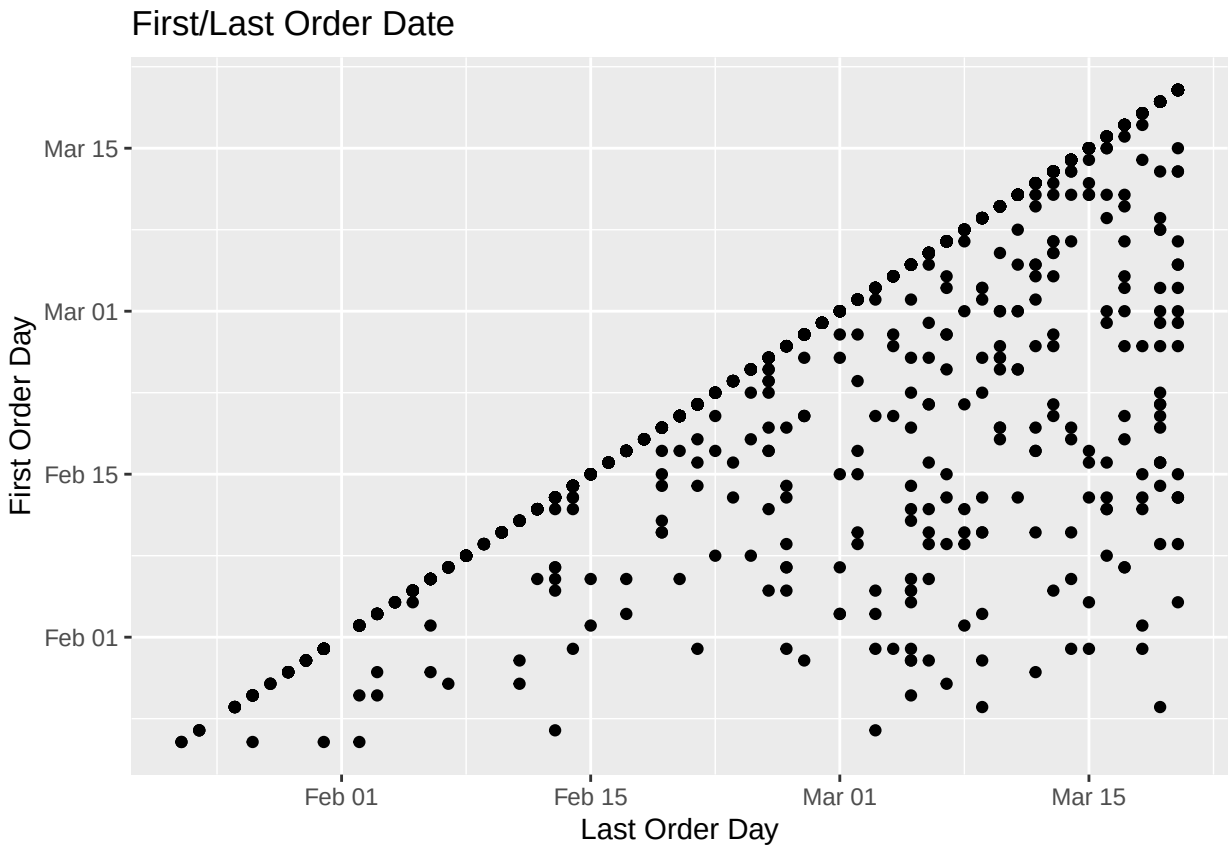
- Majority of sales are in about 15-25 dollar range. See a double peak would like to investigate further. Long tail as expected seems to be almost all sales under 100
- Sales data is from start of Feb to Mar 21. Strong rise of sales over time. Partial last day removed
- Sales Mon-Thur are pretty consistent. Large uptick Fri/Sat/Sun with Sat being pick. Fri/Sun are about 50% higher than average weekday. Sat is about double a normal weekday. (Note should check is we have different number of days of week in dataset, partial weeks, would be better to average by day. Skipping for time purposes.)
- The time of orders comes in two major peaks. About a half hour after noon is the first peak for lunch. About 6:30 is the dinner peak. Would be interesting to see how this changes by day of week.

```
#Feature Engineering by user

#Would probably be more efficient with vectorizing.
#Im just going to for loop it so I can make faster. Data is small enough

user_list<-unique(df_f$customer_id)
```

```r
df_user<-data.frame(user_list)
colnames(df_user)<-"user"

df_user$num_orders<-NA
df_user$average_price<-NA
df_user$max_price<-NA
df_user$min_price<-NA
df_user$first_order<-NA
df_user$last_order<-NA
df_user$first_order_day<-NA
df_user$last_order_day<-NA
df_user$prop_order_weekend<-NA

for (i in 1:nrow(df_user)){
  spc_user<-df_user[i,"user"]
  spc_df<-df_f[df_f$customer_id==spc_user,]

  df_user$num_orders[i]<-nrow(spc_df)
  df_user$average_price[i]<-mean(spc_df$sale_dol)
  df_user$max_price[i]<-max(spc_df$sale_dol)
  df_user$min_price[i]<-min(spc_df$sale_dol)
  df_user$first_order[i]<-min(spc_df$time_opened)
  df_user$last_order[i]<-max(spc_df$time_opened)
  df_user$first_order_day[i]<-as.Date(min(spc_df$time_opened))
  df_user$last_order_day[i]<-as.Date(max(spc_df$time_opened))
  df_user$prop_order_weekend<-sum(spc_df$dow %in% c("Saturday","Sunday"))/length(spc_df$dow)
}
```

```r
ggplot(data=df_user,aes(x=average_price,y=num_orders))+
  geom_point()+
  ggtitle("Number of Orders by Average Order price")
```


```
ggplot(data=df_user,aes(as.Date(x=last_order_day,origin='1970-01-01'),y=as.Date(first_order_day,origin=
  geom_point()+
  ggtitle("First/Last Order Date")+
  xlab("Last Order Day")+
  ylab("First Order Day")
```



2. Analysis by user

Most users are one time users. I would really look at two bins one-time and repeat customers. I want to look at this data over time and see how people fall off over time.

Splitting into cohorts. I want to see behavior over time and forecast how new users will tracks. I have data for 57 days. Im going to split into 8 cohorts each represented by 1 week of customers whose first order occured in a specific week (7 day period). Anyone who placed an order in the last day will be discarded.

```
df_user$cohort<-(df_user$first_order_day-18650)%/%7
df_user<-df_user[df_user$cohort!=8,]
```

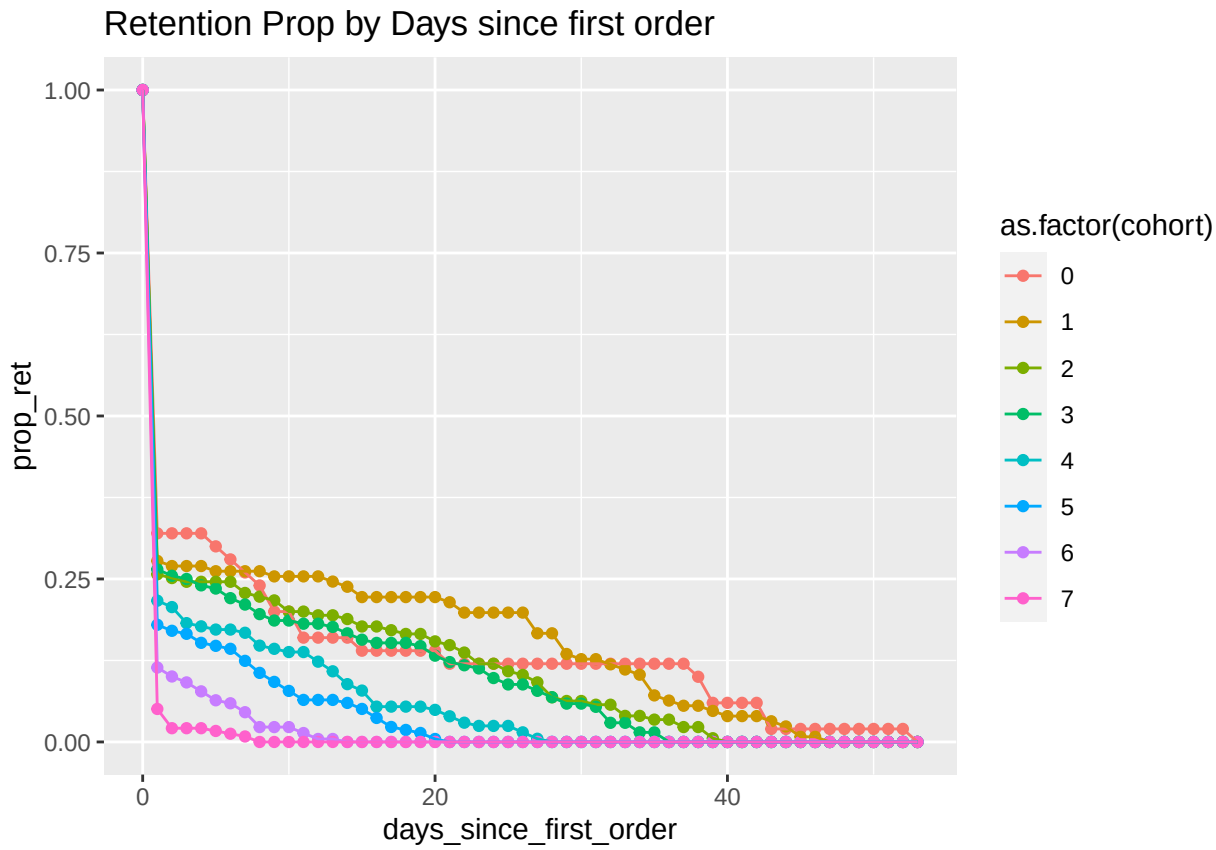
```
df_user$retention_days<-1+df_user$last_order_day-df_user$first_order_day
df_user$days_data<-max(df_user$last_order_day)-df_user$last_order_day+1
```

```
coh_ind_df<-unique(df_user[,c("user","cohort","first_order_day")])
merge_df<-merge(df_f,coh_ind_df,by.x="customer_id",by.y="user",all.x=TRUE)
merge_df_2<-merge_df[!is.na(merge_df$cohort),]
merge_df_2$order_day<-as.numeric(as.Date(merge_df_2$time_opened))
merge_df_2$days_since_first_order<-merge_df_2$order_day-merge_df_2$first_order_day
```

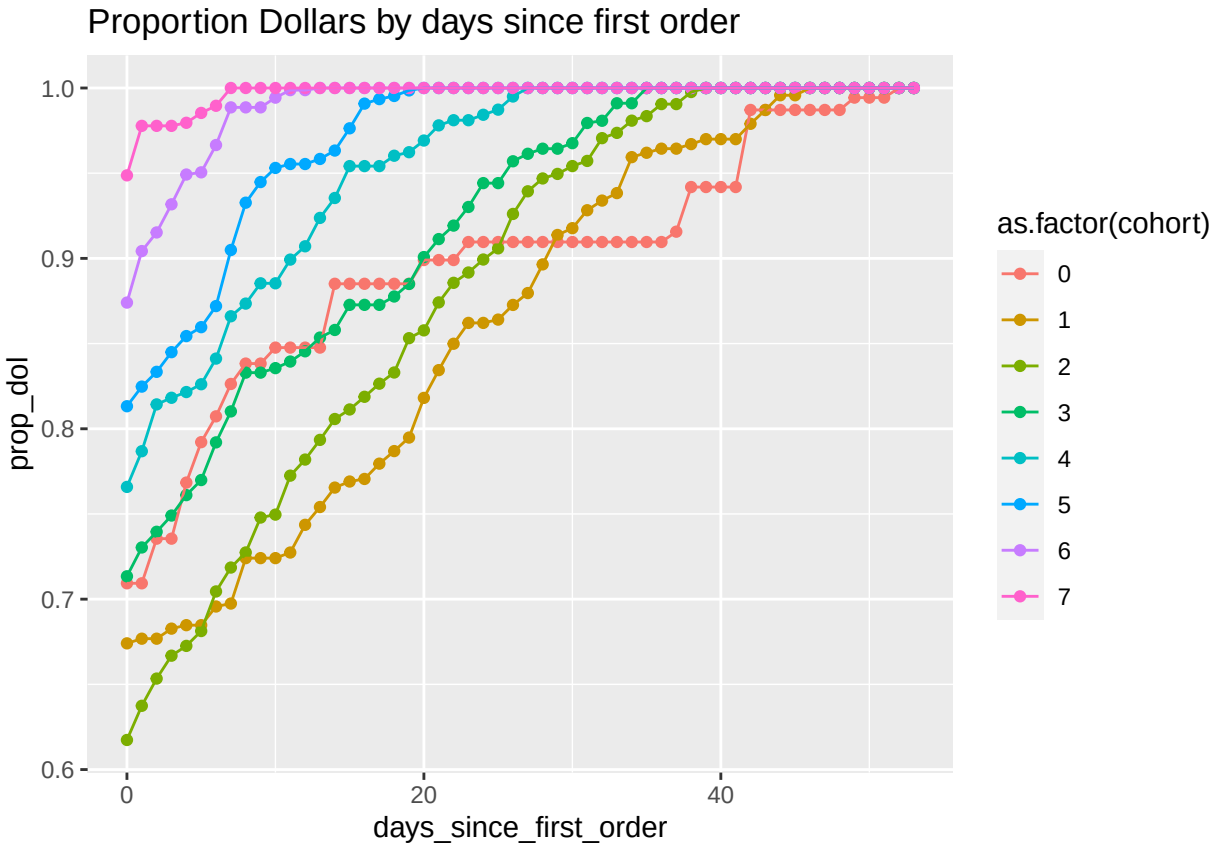
```
cohorts<-unique(df_user$cohort)
days<-0:max(df_user$retention_days)
```

```r
coh_df<-expand.grid(cohorts,days)
colnames(coh_df)<-c("cohort","days_since_first_order")

coh_df$ret_count<-NA

coh_df$cum_dol<-NA

#Using for loop because of time limitations. Vectorizing would be much more efficient
for (i in 1:nrow(coh_df)){
  spc_coh<-coh_df[i,"cohort"]
  spc_days_since<-coh_df[i,"days_since_first_order"]

  spc_df<-merge_df_2[merge_df_2$cohort==spc_coh & merge_df_2$days_since_first_order>=spc_days_since,]
  coh_df[i,"ret_count"]<-length(unique(spc_df$customer_id))

  spc_df_2<-merge_df_2[merge_df_2$cohort==spc_coh & merge_df_2$days_since_first_order<=spc_days_since,]
  coh_df[i,"cum_dol"]<-sum(spc_df_2$sale_dol)

}

coh_dol<-merge_df_2 %>%
  group_by(as.factor(cohort)) %>%
  summarise(tot_dol=sum(sale_dol))
colnames(coh_dol)[1]<-"cohort"

coh_tot_df<-table(df_user$cohort)

coh_df_2<-merge(coh_df,coh_tot_df,by.x="cohort",by.y="Var1")
colnames(coh_df_2)[5]<-"total_coh_count"
coh_df_2$prop_ret<-coh_df_2$ret_count/coh_df_2$total_coh_count


coh_df_3<-merge(coh_df_2,coh_dol,by.x="cohort",by.y='cohort')
coh_df_3$prop_dol<-coh_df_3$cum_dol/coh_df_3$tot_dol
```

```r
ggplot(data=coh_df_3,aes(x=days_since_first_order,y=prop_ret,color=as.factor(cohort)))+
  geom_point()+
  geom_line()+
  ggtitle("Retention Prop by Days since first order")
```




```
ggplot(data=coh_df_3,aes(x=days_since_first_order,y=prop_dol,color=as.factor(cohort)))+
  geom_point()+
  geom_line()+
  ggtitle("Proportion Dollars by days since first order")
```



2. Cohort Analysis by time

First a massive pinch of salt. This data is missing in many ways. We only have part of the life of our customers in many ways. Customers who started in the later part of the dataset are effectively being censored as we are missing data on their later orders. Because of this many of the later cohorts 7,6,5 should will have very different data. Over time I expect the cohorts to start to look more and more similar as later data rolls in.

We saw a huge percent of customers are 1 time customers. From the early cohorts we see ~30% customers ever place a second order. Dollar wise about 60-70% of money is made just on initial sale with the rest trickling in later.

Because of this I see the second order as a huge opportunity. I saw in an earlier graph that the drop of from second to third order is not bad. So I would focus on offering discounts or promos after a first order to try and hook a second purchase and hopefully form a habit.

I want to emphasize more data is needed for this kind of time based analysis. Too much data censoring is going on.

```
#Number of cohorts by group
coh_tot_df
```

```
##
##   0   1   2   3   4   5   6   7
##  50 126 175 204 203 217 219 237
```

```
coh_dol
```

```
## # A tibble: 8 x 2
##   cohort tot_dol
##   <fct>    <dbl>
## 1 0        1937.
## 2 1        5625.
## 3 2        8453.
## 4 3        8517.
## 5 4        6854.
## 6 5        7731.
## 7 6        7226.
## 8 7        6411.
```

```
#Using cohorts 0-3 most complete
coh_df_3[coh_df_3$cohort<4 & coh_df_3$days_since_first_order==0,]
```

```
##     cohort days_since_first_order ret_count cum_dol total_coh_count prop_ret
## 19       0                     0        50 1374.09              50        1
## 55       1                     0       126 3791.41             126        1
## 121      2                     0       175 5218.45             175        1
## 163      3                     0       204 6076.39             204        1
##     tot_dol  prop_dol
## 19  1937.22 0.7093102
## 55  5624.73 0.6740608
## 121 8452.85 0.6173598
## 163 8517.18 0.7134274
```

```
day_0_prop<-mean(coh_df_3[coh_df_3$cohort<4 & coh_df_3$days_since_first_order==0,]$prop_dol)
```

3. LTV

Using cohorts 0-3 (least censored). I see on average 67.8% of money is made on first sale. So I would value a cohort at Day 0 sales * (1/.678). This is an initial fast pass at valuation that could be improved on. This gives us a way to value our week of customers based on just their day 0 purchase value. This is probably an underestimate due to censored sales date that happened after this data set ends.

```
out_df<-coh_df_3[coh_df_3$days_since_first_order==0,]
out_df$coh_val<-out_df$cum_dol/day_0_prop

print(out_df)
```

```
##     cohort days_since_first_order ret_count cum_dol total_coh_count prop_ret
## 19       0                     0        50 1374.09              50        1
## 55       1                     0       126 3791.41             126        1
## 121      2                     0       175 5218.45             175        1
## 163      3                     0       204 6076.39             204        1
## 222      4                     0       203 5249.51             203        1
## 271      5                     0       217 6287.09             217        1
## 335      6                     0       219 6316.03             219        1
```

```
## 402      7                      0       237 6082.22             237        1
##     tot_dol  prop_dol  coh_val
## 19  1937.22 0.7093102 2025.070
## 55  5624.73 0.6740608 5587.603
## 121 8452.85 0.6173598 7690.708
## 163 8517.18 0.7134274 8955.100
## 222 6854.16 0.7658867 7736.483
## 271 7731.01 0.8132301 9265.620
## 335 7226.04 0.8740652 9308.271
## 402 6410.90 0.9487311 8963.692
```

```
print(out_df[,c("cohort","coh_val")])
```

```
##     cohort  coh_val
## 19       0 2025.070
## 55       1 5587.603
## 121      2 7690.708
## 163      3 8955.100
## 222      4 7736.483
## 271      5 9265.620
## 335      6 9308.271
## 402      7 8963.692
```