678 Midterm Project

Retail Data Analysis

Jinfei Xue

Dec 8, 2018

Abstract

Sales forecasting is a crucial part of the financial planning of a business. This report use the retail data from a company to understand the customer purchase behaviour. First, the report shows the structure of data and random sampling. After data cleaning, the sample dataset is divided into train and test datasets. Second, exploratory data analysis is made to show relationships between variables. Third, this report makes several types of models, including linear/polynomial/multinomial/multilevel models, check and compares them based on ANOVA test to select the relatively good model (multilevel model varying by intercepts). Finally, this report makes predictions based on the selected model, discusses the implication and limitations of the model, and indicates the future direction of retail forecasting methods.

Keywords: **retail forecasting**, **multilevel model**, **model check**

1 Introduction

1.1 Background

A retail company named "ABC Private Limited" wants to understand the customer purchase behaviour (specifically, purchase amount) against various products of different categories. They have shared purchase summary of various customers for selected high volume products from last month. The data set also contains customer demographics (age, gender, marital status, city_type, stay_in_current_city), product details (product_id and product category) and purchase amount of each client.

Now, they want to build a model to predict the purchase amount of customer against the other variables which will help them to create personalized offer for customers. Here I have to mention that because of the privacy, the occupation, City_Category, product categories are masked and the categories are represented by numbers or letters.

1.2 Data

The data can be downloaded from the website https://datahack.analyticsvidhya.com/contest/black-friday/?utm_source=auto-email. The original dataset has 550,068 observations and 12 variables. The main variables are listed as follows:

- User_ID (as group)
- Gender (M/F)
- Age (Age in bins)
- Occupation (0, 1, ..., 20)
- City_Category (A/B/C)
- Stay_In_Current_City_Years (the number of years stay in current city)
- Marital_Status (0/1)
- Product_Category_1 (the number of bought products in category 1)
- Product_Category_2 (the number of bought products in category 2)
- Product_Category_3 (the number of bought products in category 3)
- Purchase (Purchase amount in dollars)

1.2.1 Data Structure

The following gives an impression of the structure of the data.

The first six rows of the data are shown below:

```
##   User_ID Product_ID Gender   Age Occupation City_Category
## 1 1000001  P00069042      F  0-17         10             A
## 2 1000001  P00248942      F  0-17         10             A
## 3 1000001  P00087842      F  0-17         10             A
## 4 1000001  P00085442      F  0-17         10             A
## 5 1000002  P00285442      M   55+         16             C
## 6 1000003  P00193542      M 26-35         15             A
##   Stay_In_Current_City_Years Marital_Status Product_Category_1
## 1                          2              0                  3
## 2                          2              0                  1
## 3                          2              0                 12
## 4                          2              0                 12
## 5                         4+              0                  8
## 6                          3              0                  1
##   Product_Category_2 Product_Category_3 Purchase
## 1                 NA                 NA     8370
## 2                  6                 14    15200
## 3                 NA                 NA     1422
## 4                 14                 NA     1057
## 5                 NA                 NA     7969
## 6                  2                 NA    15227
```

The following shows the class of variables. Some variables need to be transformed to factor, like **Occupation** and **Marital_Status**. **User_ID** should be transformed to character or factor.

```
## Observations: 550,068
## Variables: 12
## $ User_ID                    <int> 1000001, 1000001, 1000001, 100000
1,...
## $ Product_ID                 <fct> P00069042, P00248942, P00087842,
P0...
## $ Gender                     <fct> F, F, F, F, M, M, M, M, M, M, M,
M,...
## $ Age                        <fct> 0-17, 0-17, 0-17, 0-17, 55+, 26-3
5,...
## $ Occupation                 <int> 10, 10, 10, 10, 16, 15, 7, 7, 7,
20...
## $ City_Category              <fct> A, A, A, A, C, A, B, B, B, A, A,
A,...
## $ Stay_In_Current_City_Years <fct> 2, 2, 2, 2, 4+, 3, 2, 2, 2, 1, 1,
 1...
## $ Marital_Status             <int> 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
1,...
## $ Product_Category_1         <int> 3, 1, 12, 12, 8, 1, 1, 1, 1, 8, 5,
 ...
## $ Product_Category_2         <int> NA, 6, NA, 14, NA, 2, 8, 15, 16,
NA...
```

```
## $ Product_Category_3         <int> NA, 14, NA, NA, NA, NA, 17, NA, N
A,...
## $ Purchase                   <int> 8370, 15200, 1422, 1057, 7969, 15
22...
```

The distributions of each variable are shown below:

```
##     User_ID            Product_ID     Gender        Age        
##  Min.   :1000001   P00265242:  1880   F:135809   0-17 : 15102  
##  1st Qu.:1001516   P00025442:  1615   M:414259   18-25: 99660  
##  Median :1003077   P00110742:  1612              26-35:219587  
##  Mean   :1003029   P00112142:  1562              36-45:110013  
##  3rd Qu.:1004478   P00057642:  1470              46-50: 45701  
##  Max.   :1006040   P00184942:  1440              51-55: 38501  
##                    (Other)  :540489              55+  : 21504  
##    Occupation     City_Category Stay_In_Current_City_Years
##  Min.   : 0.000   A:147720      0 : 74398                 
##  1st Qu.: 2.000   B:231173      1 :193821                 
##  Median : 7.000   C:171175      2 :101838                 
##  Mean   : 8.077                 3 : 95285                 
##  3rd Qu.:14.000                 4+: 84726                 
##  Max.   :20.000                                           
##                                                           
##  Marital_Status   Product_Category_1 Product_Category_2 Product_Cate
gory_3
##  Min.   :0.0000   Min.   : 1.000     Min.   : 2.00      Min.   : 3.0
   
##  1st Qu.:0.0000   1st Qu.: 1.000     1st Qu.: 5.00      1st Qu.: 9.0
   
##  Median :0.0000   Median : 5.000     Median : 9.00      Median :14.0
   
##  Mean   :0.4097   Mean   : 5.404     Mean   : 9.84      Mean   :12.7
   
##  3rd Qu.:1.0000   3rd Qu.: 8.000     3rd Qu.:15.00      3rd Qu.:16.0
   
##  Max.   :1.0000   Max.   :20.000     Max.   :18.00      Max.   :18.0
   
##                                      NA's   :173638     NA's   :3832
47
##     Purchase    
##  Min.   :   12  
##  1st Qu.: 5823  
##  Median : 8047  
##  Mean   : 9264  
##  3rd Qu.:12054  
##  Max.   :23961  
## 
```

1.2.2 Random Sampling

Because the number of observations in the original data is too large, we randomly choose 200 **User_ID**s as index to sample observations.

```
# Sample groups from original data
set.seed(123)
index <- data_frame(User_ID = sample(unique(data$User_ID), 200, replace
 = FALSE)) %>%
  arrange(User_ID)
sample <- inner_join(data, index) %>%
  arrange(User_ID)
```

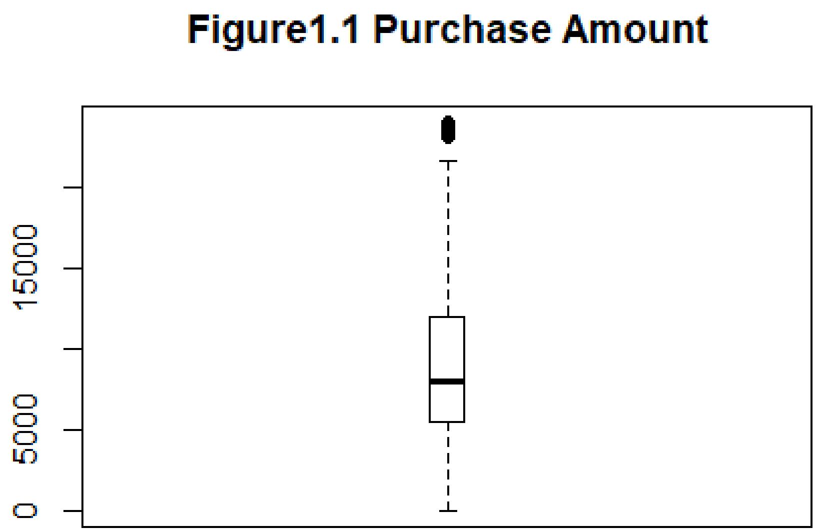
1.2.3 Data Cleaning

Because NAs only happen in product category variables and it represents the client didn't buy any products in that category, we replace NA with zero.

```
# Check NA
sapply(sample, function(x) sum(is.na(x)))

##                    User_ID                 Product_ID
##                          0                          0
##                     Gender                        Age
##                          0                          0
##                 Occupation              City_Category
##                          0                          0
## Stay_In_Current_City_Years             Marital_Status
##                          0                          0
##         Product_Category_1         Product_Category_2
##                          0                       5720
##         Product_Category_3                   Purchase
##                      11996                          0

# Replace NA with 0
sample[is.na(sample)] <- 0
sum(is.na(sample)) #check

## [1] 0
```

We can see now there is no NAs in the sample data by checking it.

```
# transfer variables to factor
sample$User_ID <- as.factor(sample$User_ID)
sample$Occupation <- as.factor(sample$Occupation)
sample$Marital_Status <- as.factor(sample$Marital_Status)

# Boxplot to show the outliers
boxplot(sample$Purchase, main="Figure1.1 Purchase Amount", boxwex=0.1)
```



From the bosxplot, we can see there are some outliers in **Purchase**. So then we will replace those outliers with median value of **Purchase**.

```
# replace outliers with median value of purchase amount
outlier_values <- boxplot.stats(sample$Purchase)$out  # outlier values
sample$Purchase[which(sample$Purchase %in% outlier_values)]=median(samp
le$Purchase)
```

1.2.4 Train and Test Datasets

Because after modeling we should use test data to predict response variable and check the accuracy of the model, we first divide the sample data into train and test datasets. The minimum number of observations in each **User_ID** is 11 so that we can use stratified sampling by **User_ID** to randomly choose 70% of the sample data as train dataset and the remaining as test dataset.

In order to ensure test dataset has the same User_IDs as train dataset, we finally check it and the output is "TRUE".

```
# Check the minimum number of obeservations in groups
group <- sample %>%
  group_by(User_ID) %>%
  summarise(number = n()) %>%
  arrange(User_ID)
min(group$number)

## [1] 11
```

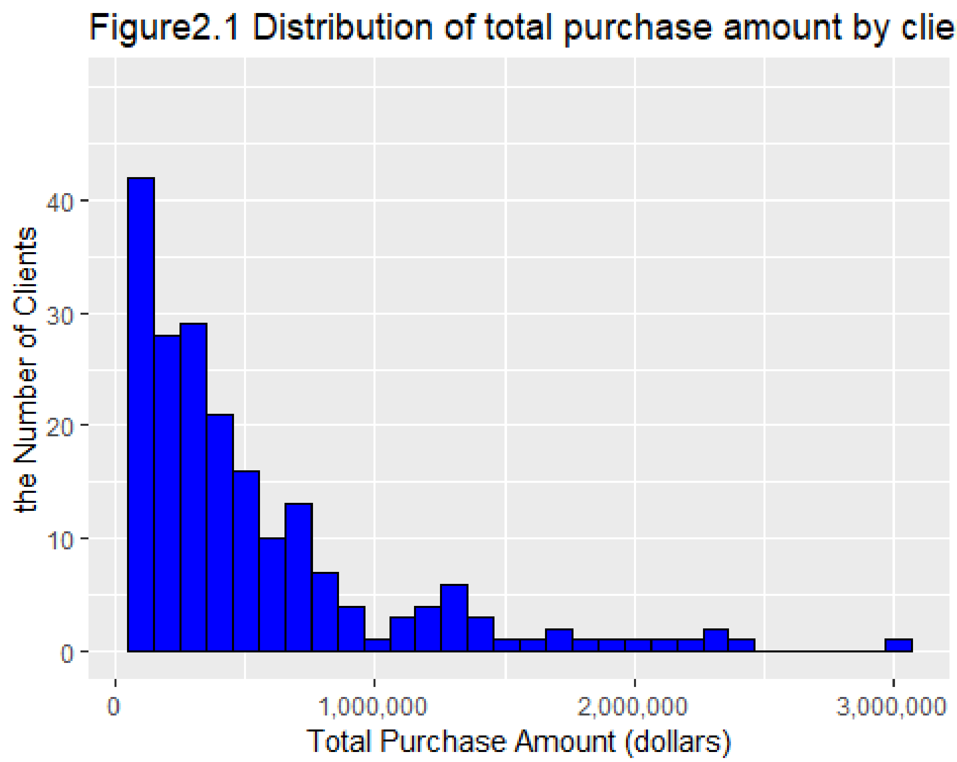
```
# So we can divide the sample into train and test datasets by groups (U
ser_ID)

# Train dataset
set.seed(221)
bf <- splitstackshape::stratified(sample, "User_ID", .7)

# Test dataset
bf_test <- anti_join(sample, bf)

# Test whether User_ID in test dataset is a subset of User_ID in train
dataset
sum(unique(bf_test$User_ID) %in% unique(bf$User_ID))==length(unique(bf_
test$User_ID))

## [1] TRUE
```

2 Exploratory Data Analysis

2.1 Total Purchase Amount Distribution

```
#total purchaser
bf %>%
  select(User_ID) %>%
  unique() %>%
  nrow() %>%
  paste("buyers sampled registered at Black Friday")

## [1] "200 buyers sampled registered at Black Friday"

library(ggplot2)
bf1 <- bf %>%
  group_by(User_ID, Gender, Age, Occupation, City_Category,
           Stay_In_Current_City_Years, Marital_Status) %>%
  summarise(total_Product_Category_1 = sum(Product_Category_1),
            total_Product_Category_2 = sum(Product_Category_2),
            total_Product_Category_3 = sum(Product_Category_3),
            total_purchase = sum(Purchase))
summary(bf1$total_purchase)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   53970  184000  362111  536592  686779 2970816

ggplot(data = bf1, aes(x = total_purchase)) +
  geom_histogram(col = 'black', fill = 'blue') +
  labs(x = 'Total Purchase Amount (dollars)', y = 'the Number of Client
s',
       title = "Figure2.1 Distribution of total purchase amount by clie
nts") +
  scale_y_continuous(limits = c(0,50), breaks = c(0,10,20,30,40)) +
```

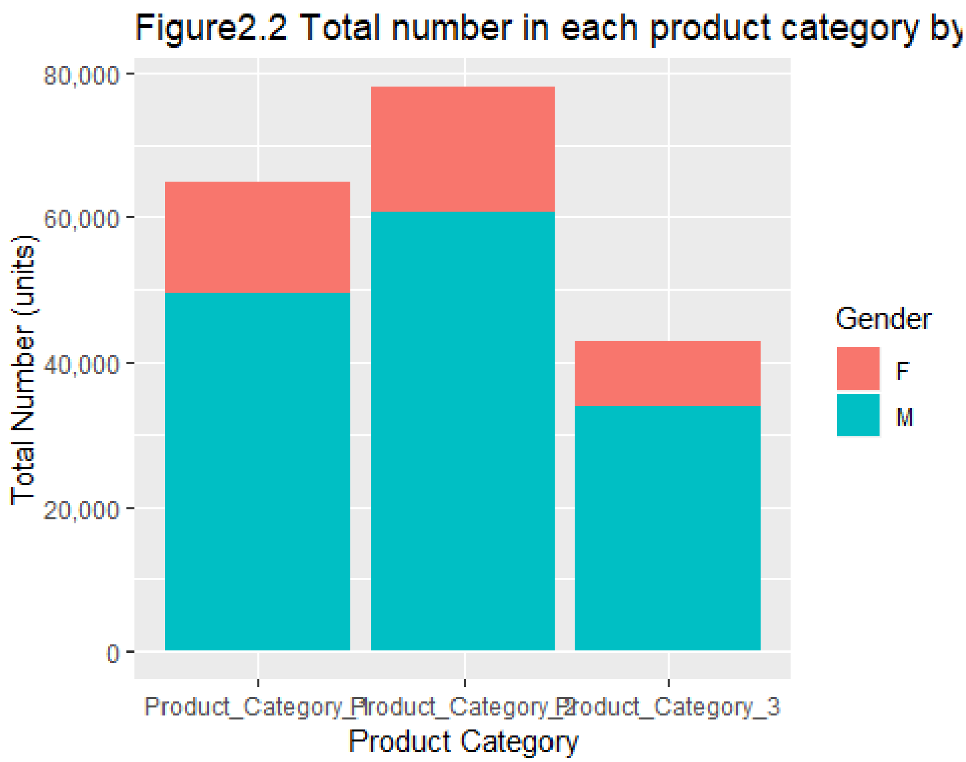
```
  scale_x_continuous(labels = scales::comma) #prevent scientific number
 in x-axis
```



From figure 2.1, we can see most ofvclients spent relatively small amount of money while there exists minority of clients spent very large amount of money last month.

2.2 Total Number in Each Product Category by Gender

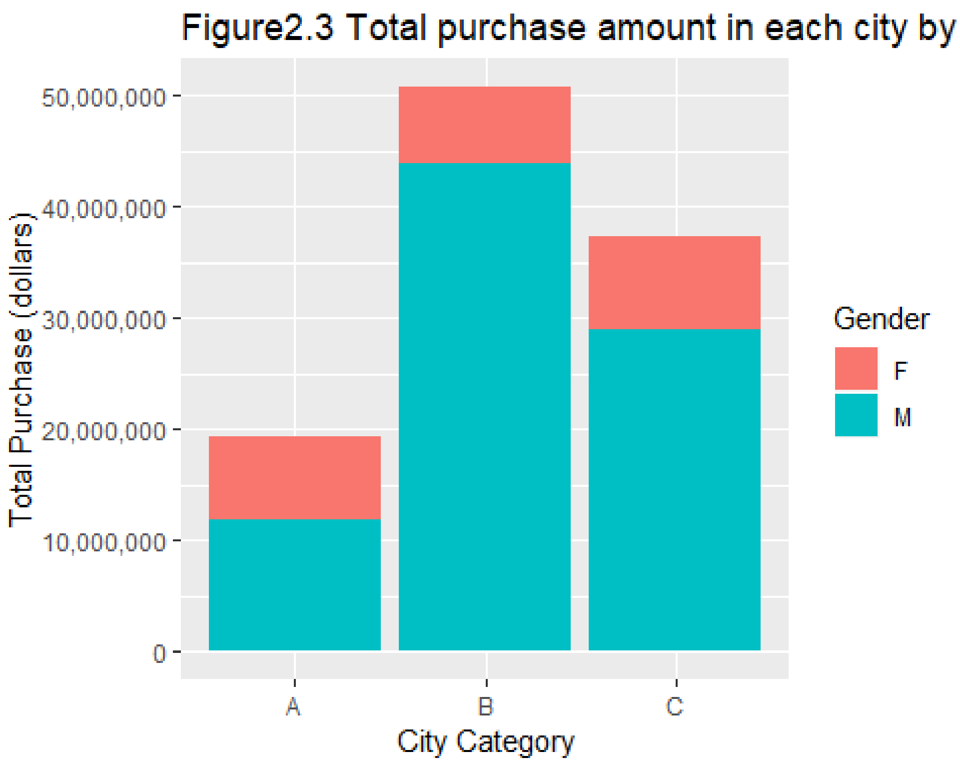
```
library(tidyr)
bf2 <- bf %>%
  group_by(Gender) %>%
  summarise(Product_Category_1 = sum(Product_Category_1),
            Product_Category_2 = sum(Product_Category_2),
            Product_Category_3 = sum(Product_Category_3)) %>%
  gather(key = Product_Category, value = total_number,
         Product_Category_1,Product_Category_2,Product_Category_3)

ggplot(data = bf2, aes(x=Product_Category, y = total_number,fill = Gend
er)) +
  geom_col() +
  labs(x = 'Product Category', y = 'Total Number (units)',
       title = "Figure2.2 Total number in each product category by gend
er") +
  guides(fill=guide_legend(title = "Gender")) +
  scale_y_continuous(labels = scales::comma) #prevent scientific number
 in x-axis
```



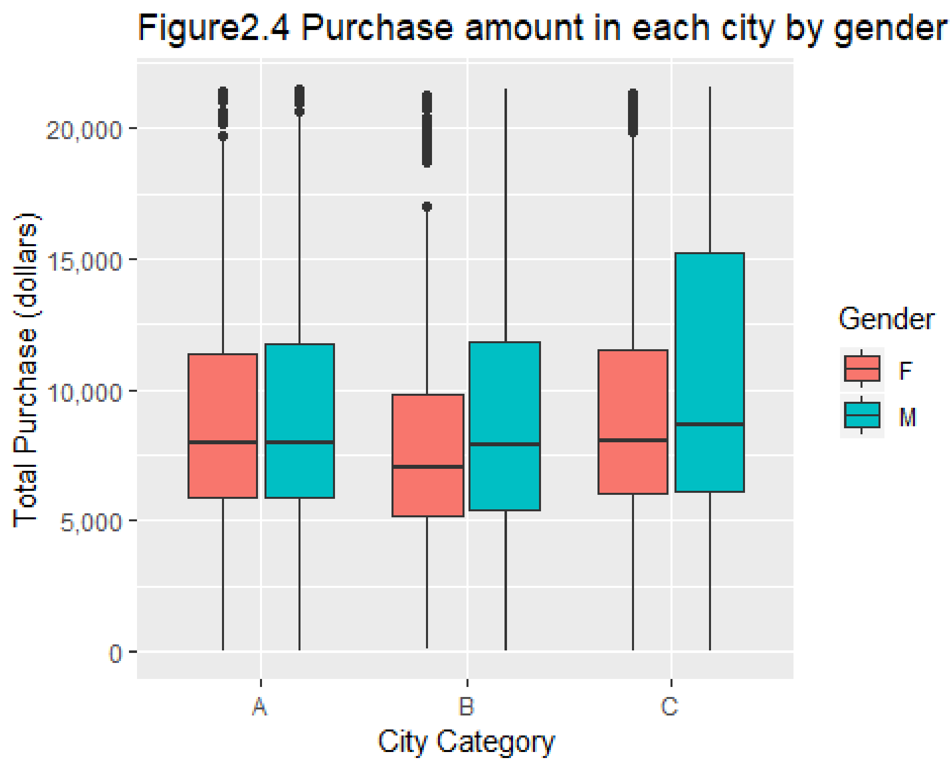
We can see the product category 2 is the most popular category in the retail store. Besides, males bought more products than females in all the three categories.

2.3 Total Purchase Amount in Each City by Gender

```
bf3 <- bf %>%
  group_by(City_Category, Gender) %>%
  summarise(total_purchase = sum(Purchase))

ggplot(data = bf3, aes(x=City_Category, y = total_purchase, fill = Gend
er)) +
  geom_col() +
  labs(x = 'City Category', y = 'Total Purchase (dollars)',
       title = "Figure2.3 Total purchase amount in each city by gender")
 +
  guides(fill=guide_legend(title = "Gender")) +
  scale_y_continuous(labels = scales::comma) #prevent scientific number
 in x-axis
```




```
ggplot(data = bf, aes(x=City_Category, y = Purchase, fill = Gender)) +
  geom_boxplot() +
  labs(x = 'City Category', y = 'Total Purchase (dollars)',
       title = "Figure2.4 Purchase amount in each city by gender") +
  guides(fill=guide_legend(title = "Gender")) +
  scale_y_continuous(labels = scales::comma) #prevent scientific number
 in x-axis
```

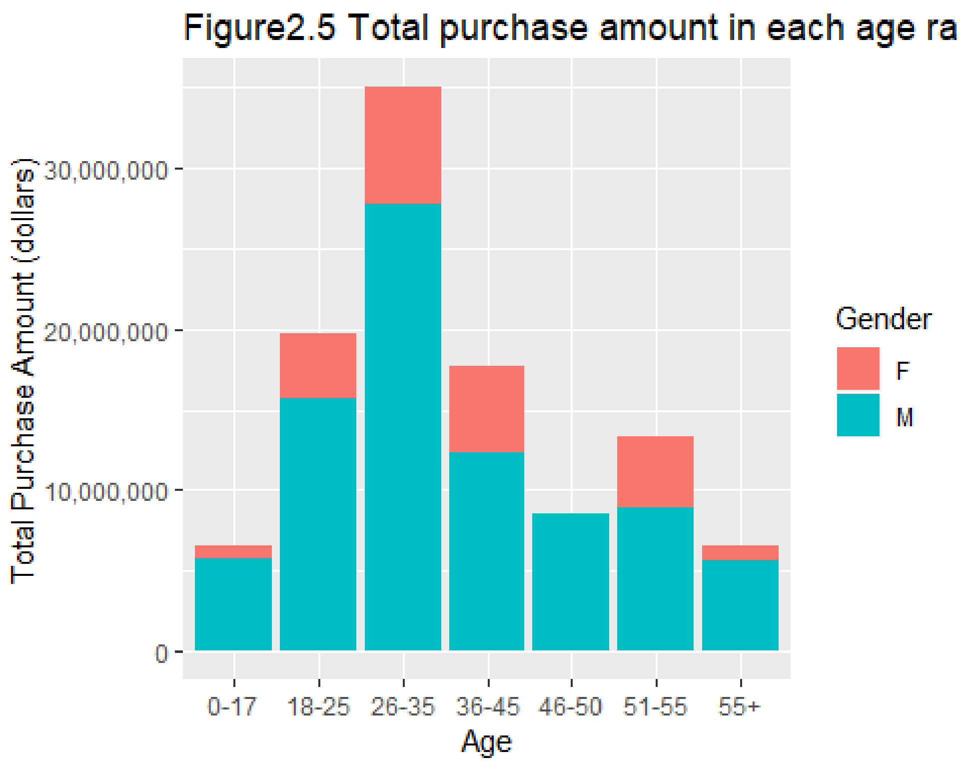


From figure 2.3, we can find that clients from city B spent the most money in the three cities and in each city, males spent more money than females. However, from figure 2.4 we can see that the reason why total amount in city B is the most is that there are more outliers whose values are very large.

2.4 Total Purchase Amount in Each Age Range by Gender

From the above data visualizations, we find that males spent more money than females, so there comes our next question: males of what age range will spend more money. Let's find out.

```
bf5 <- bf %>%
  group_by(Age, Gender) %>%
  summarise(total_purchase = sum(Purchase))

ggplot(data = bf5, aes(x=Age, y = total_purchase, fill = Gender)) +
  geom_col() +
  labs(x = 'Age', y = 'Total Purchase Amount (dollars)',
       title = "Figure2.5 Total purchase amount in each age range by ge
nder") +
  guides(fill=guide_legend(title = "Gender")) +
  scale_y_continuous(labels = scales::comma) #prevent scientific number
 in x-axis
```

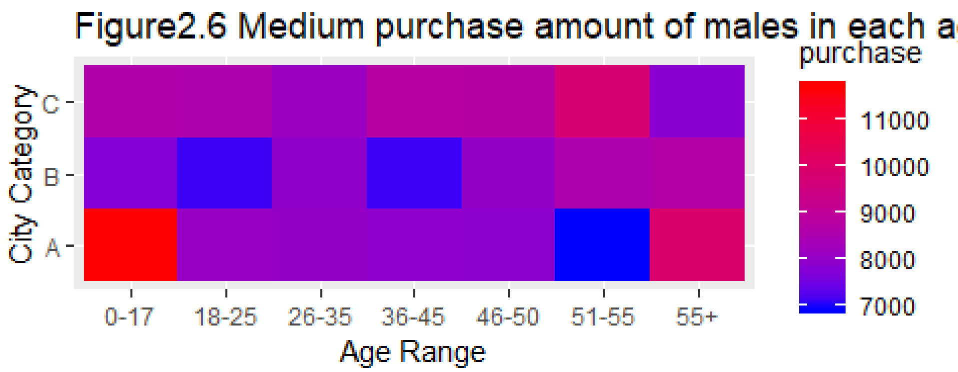


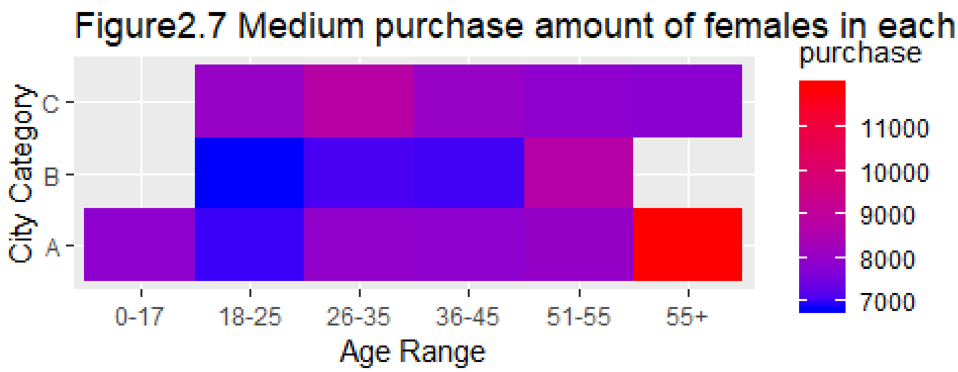
From figure 2.5, we can find that males who are 26-35 years old spent the most money, while males who are 0-17 or more than 55 years old spent the least money.

2.5 Medium Purchase Amount of Clients in Each Age Range by City Category

```
p3<-bf %>%
  filter(Gender=="M") %>%
  group_by(Age,City_Category) %>%
  summarise(purchase=median(Purchase)) %>%
  ggplot(aes(x=Age,y=City_Category,fill=purchase))+
  geom_tile()+
  scale_fill_continuous(low="blue",high="red")+
  labs(x = 'Age Range', y = 'City Category',
       title = "Figure2.6 Medium purchase amount of males in each age r
ange by city category")

p4<-bf %>%
  filter(Gender=="F") %>%
  group_by(Age,City_Category) %>%
  summarise(purchase=median(Purchase)) %>%
  ggplot(aes(x=Age,y=City_Category,fill=purchase))+
  geom_tile()+
  scale_fill_continuous(low="blue",high="red")+
  labs(x = 'Age Range', y = 'City Category',
       title = "Figure2.7 Medium purchase amount of females in each age
```

```
 range by city category")

gridExtra::grid.arrange(p3,p4)
```

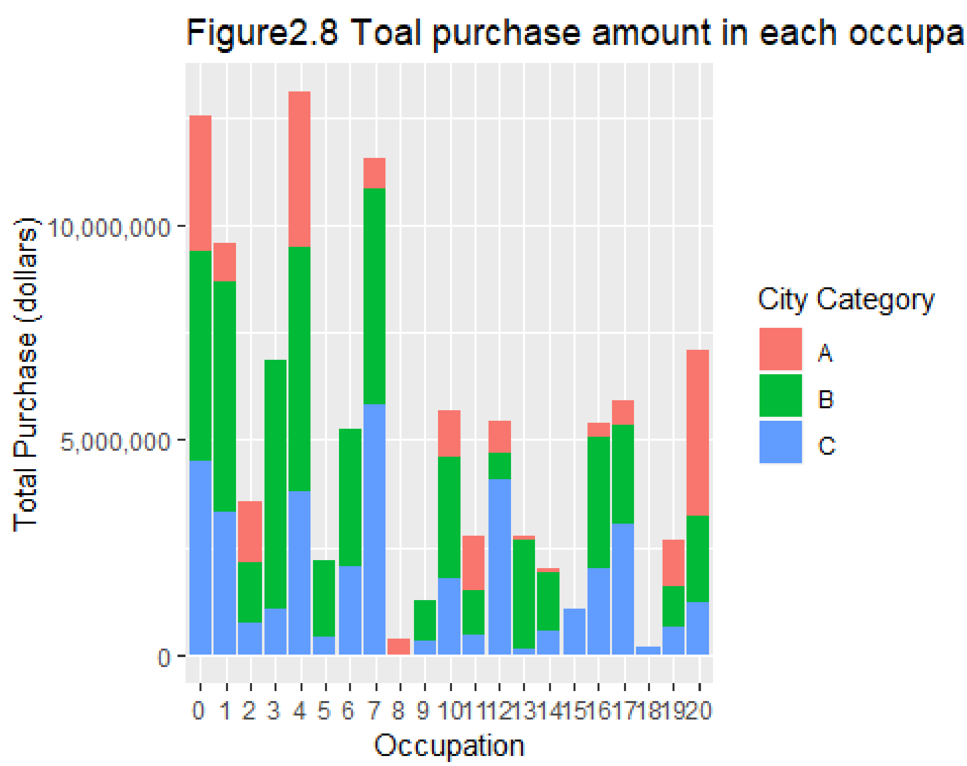




From figure 2.5 and 2.6, we can find that although the total amount of money males in 0-17 age range spent is much less than that in 26-35 age range, the medium amount of money males in 0-17 age range spent is more than that in 26-35 age range for clients from City A. From figure 2.7, we can see that for females, the medium amount of money clients from City A who are more than 55 years old spent is the largest.

2.6 Toal Purchase Amount in Each Occupation by City Category

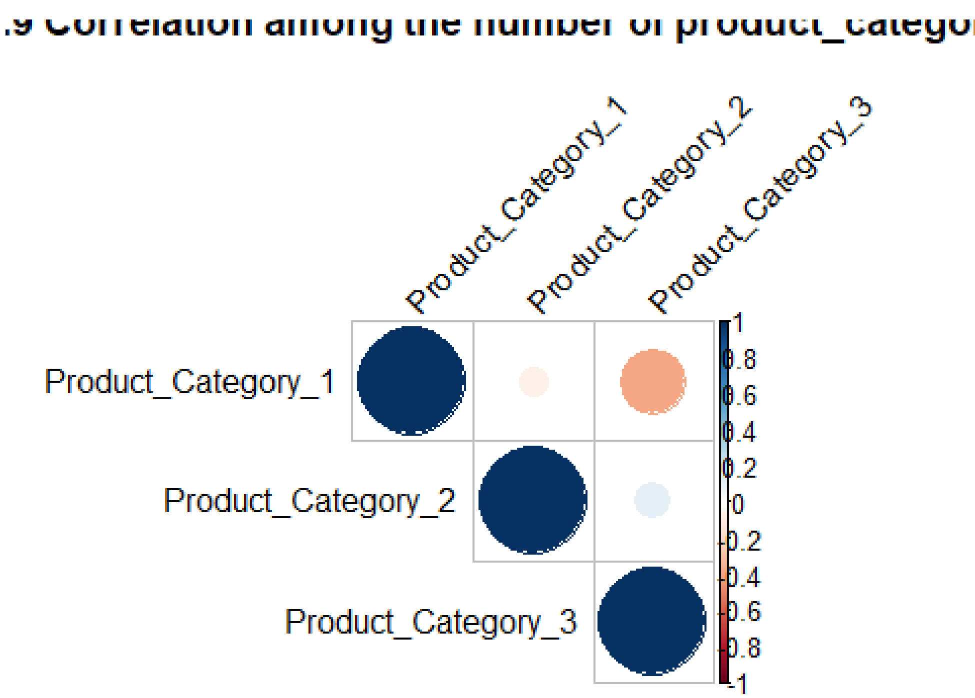
```
bf6 <- bf %>%
  group_by(Occupation, City_Category) %>%
  summarise(total_purchase = sum(Purchase))

ggplot(data = bf6, aes(x=Occupation, y = total_purchase, fill=City_Cate
gory)) +
  geom_col() +
  labs(x = 'Occupation', y = 'Total Purchase (dollars)',
       title = "Figure2.8 Toal purchase amount in each occupation by ci
ty category") +
  guides(fill=guide_legend(title = "City Category")) +
  scale_y_continuous(labels = scales::comma) #prevent scientific number
 in x-axis
```



From figure 2.8, we can find clients from the occupation number 8 spent the least money, while clients from occupation number 4 spent the most money.

2.7 Correlation among the Number of Product_Category_1/2/3

```
library(dplyr)
product <- bf %>%
  select(Product_Category_1, Product_Category_2, Product_Category_3)
res <- cor(product)
round(res, 2)

##                    Product_Category_1 Product_Category_2
## Product_Category_1               1.00              -0.08
## Product_Category_2              -0.08               1.00
## Product_Category_3              -0.38               0.12
##                    Product_Category_3
## Product_Category_1              -0.38
## Product_Category_2               0.12
## Product_Category_3               1.00

library(corrplot)
corrplot(res, type = "upper", order = "hclust", tl.col = "black", tl.sr
t = 45,
         title = "Figure2.9 Correlation among the number of product_cat
egory_1/2/3")
```



From both the table and figure 2.9, we can see there is negetive relationship between product category 1 and 3, and positive relationship between product category 2 and 3.

3 Modeling and Checking

Because our goal is to predict the purchase amount of clients, in this part we began to make models to fit the train dataset. We will first make simple linear model, and then add interaction in it, and try polynomial, multinomial and multilevel models. Finally, based on methods for checking models, we choose the best model for prediction.

The predictors include **Gender**, **Age**, **Occupation**, **City_Category**, **Stay_In_Current_City_Years**, **Marital_Status**, **Product_Category_1**, **Product_Category_2** and **Product_Category_3**. The response variable is **Purchase**.

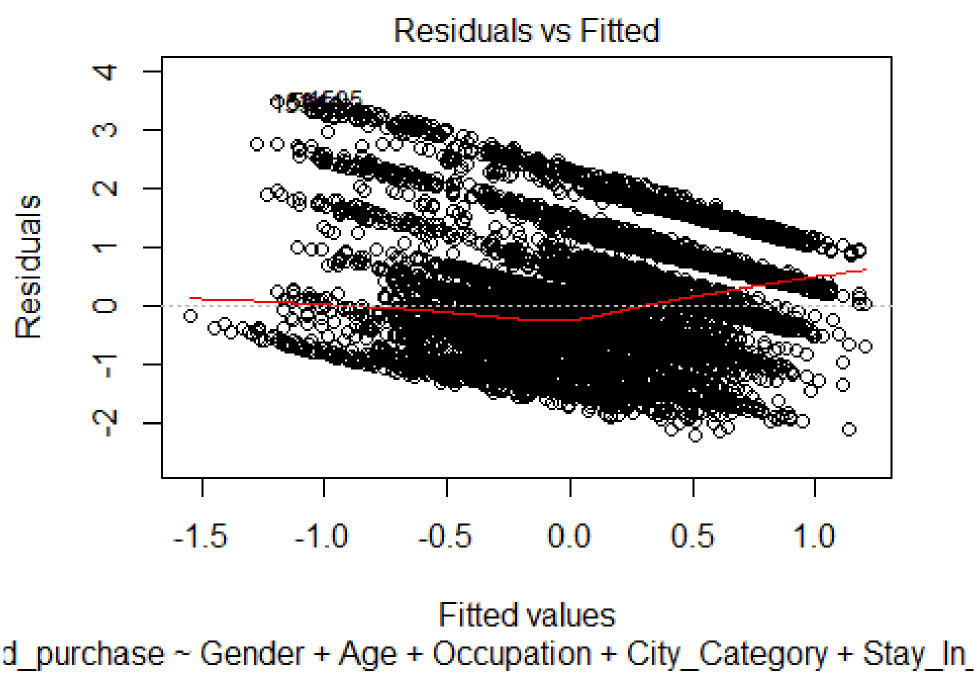
3.1 Simple Linear Regression Model

Let's first start with simple linear regression model. From the figure 2.1, we can see the purchase amount is skewed, so first we standardize purchase amount into **sd_purchase** as response variable. After trying many times, we can obtain the following model whose AIC is smallest with all the predictors.

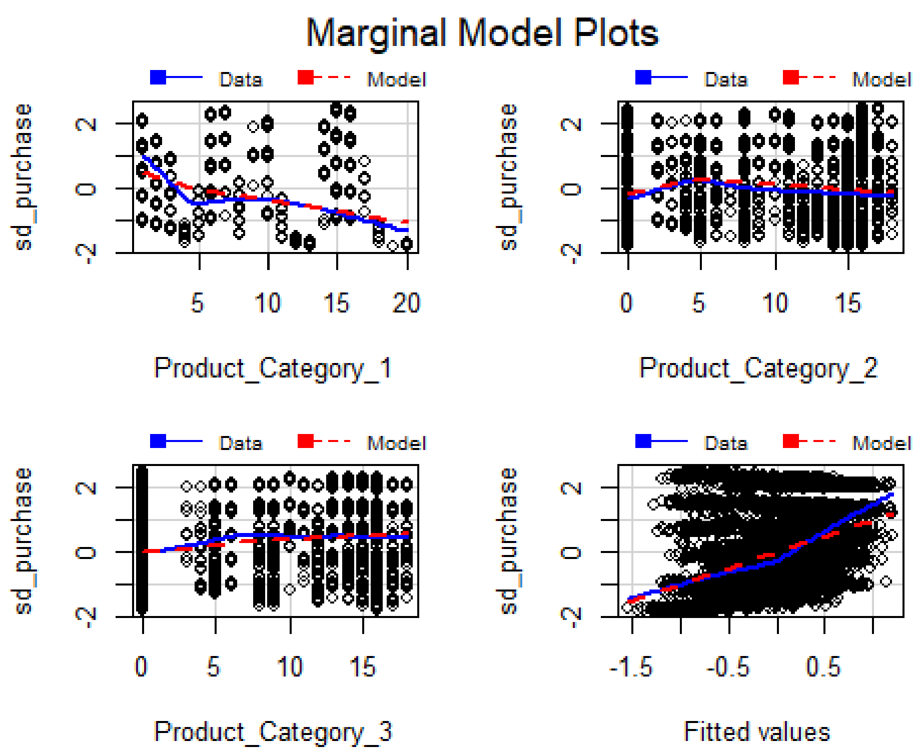
```r
# Standardize the response variable
bf$sd_purchase <- (bf$Purchase-mean(bf$Purchase))/sd(bf$Purchase)
# Fit the full model
r1 <- lm(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
           Product_Category_2 + Product_Category_3, data = bf)

summary(r1)

##
## Call:
## lm(formula = sd_purchase ~ Gender + Age + Occupation + City_Category
 +
##     Stay_In_Current_City_Years + Marital_Status + Product_Category_1
 +
##     Product_Category_2 + Product_Category_3, data = bf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1956 -0.6084 -0.1322  0.4390  3.5010
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)

## (Intercept)                 -0.1383251  0.1174563  -1.178 0.238951

## GenderM                      0.0921942  0.0235300   3.918 8.97e-05
***
## Age18-25                     0.2013755  0.1101775   1.828 0.067614 .

## Age26-35                     0.2927538  0.1089776   2.686 0.007234
**
## Age36-45                     0.3129005  0.1100265   2.844 0.004465
**
## Age46-50                     0.3816116  0.1121775   3.402 0.000672
***
## Age51-55                     0.2908368  0.1121964   2.592 0.009548
**
## Age55+                       0.2270808  0.1150297   1.974 0.048393
*
## Occupation1                 -0.0883069  0.0407128  -2.169 0.030101
*
## Occupation2                  0.1957090  0.0561720   3.484 0.000496
***
## Occupation3                  0.1702559  0.0450041   3.783 0.000156
***
## Occupation4                  0.0576382  0.0376590   1.531 0.125913

## Occupation5                 -0.0438047  0.0656672  -0.667 0.504740
```

```
## Occupation6                  -0.0921096  0.0467644  -1.970 0.048902
*
## Occupation7                   0.2273892  0.0407420   5.581 2.44e-08
***
## Occupation8                   0.5914485  0.1799854   3.286 0.001019
**
## Occupation9                  -0.3643749  0.0751249  -4.850 1.25e-06
***
## Occupation10                  0.3801086  0.1161307   3.273 0.001067
**
## Occupation11                  0.0922878  0.0608560   1.516 0.129421

## Occupation12                  0.1793782  0.0492085   3.645 0.000268
***
## Occupation13                  0.2552269  0.0726177   3.515 0.000442
***
## Occupation14                 -0.2511131  0.0621414  -4.041 5.36e-05
***
## Occupation15                  0.1159689  0.0911509   1.272 0.203301

## Occupation16                  0.1978790  0.0479363   4.128 3.69e-05
***
## Occupation17                 -0.0364159  0.0469153  -0.776 0.437643

## Occupation18                 -0.5098354  0.1858348  -2.743 0.006088
**
## Occupation19                 -0.1272301  0.0630849  -2.017 0.043738
*
## Occupation20                  0.0810667  0.0458964   1.766 0.077372 .

## City_CategoryB                0.0465924  0.0275599   1.691 0.090942 .

## City_CategoryC                0.1908364  0.0278671   6.848 7.86e-12
***
## Stay_In_Current_City_Years1  -0.1130571  0.0312327  -3.620 0.000296
***
## Stay_In_Current_City_Years2  -0.2060347  0.0357519  -5.763 8.48e-09
***
## Stay_In_Current_City_Years3   0.0356068  0.0395485   0.900 0.367962

## Stay_In_Current_City_Years4+ -0.1345186  0.0342189  -3.931 8.50e-05
***
## Marital_Status1               0.0862998  0.0224138   3.850 0.000119
***
## Product_Category_1           -0.0690583  0.0023691 -29.150  < 2e-16
***
## Product_Category_2            0.0007417  0.0013580   0.546 0.584968
```

```
## Product_Category_3            0.0277678  0.0014935  18.592  < 2e-16
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9104 on 11736 degrees of freedom
## Multiple R-squared:  0.1737, Adjusted R-squared:  0.1711
## F-statistic: 66.69 on 37 and 11736 DF,  p-value: < 2.2e-16

# Residual Plot
plot(r1, which = 1)
```



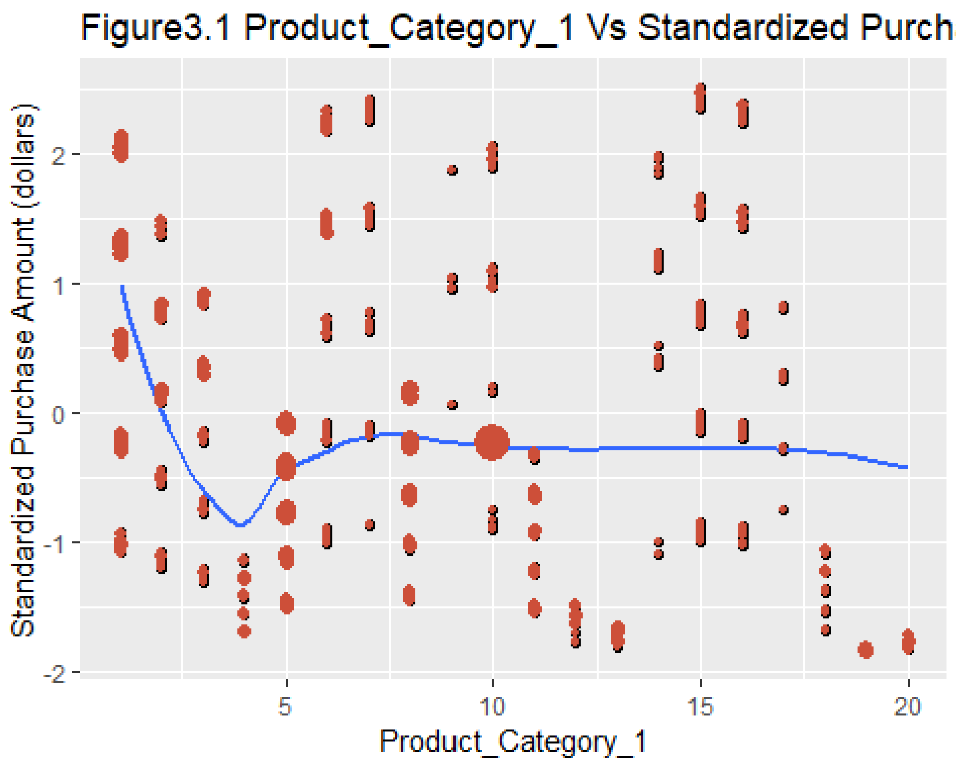

```
# Marginal model plots
library(car)
marginalModelPlots(r1)
```



From the summary result, the p value of F statistics is small and most coefficients are significant. However, from the residual plot, we can see that the points have a decreasing trend and are not randomly dispersed around the horizontal line at zero (the dashed black line). Also, we can see from the first marginal plot, there exists a big discrepency between the linear regression line and actual data line. And after looking at the last marginal plot, we can conclude the simple linear regression model does not fit the data well.

3.2 Polynomial regression model

```
ggplot(bf, aes(x=Product_Category_1, y=sd_purchase)) +
  geom_point() +
  geom_smooth(method="loess", se=F) +
  geom_count(col="tomato3", show.legend=F) +
  #xlim(c(0, 0.1)) +
  #ylim(c(0, 500000)) +
  labs(y="Standardized Purchase Amount (dollars)",
       x="Product_Category_1",
       title="Figure3.1 Product_Category_1 Vs Standardized Purchase Amo
unt")
```



From figure 3.1, we can see a nonlinear effect of **Product_Category_1** on **sd_purchase**. Therefore, so then we will try to fit a polynomial regression model.

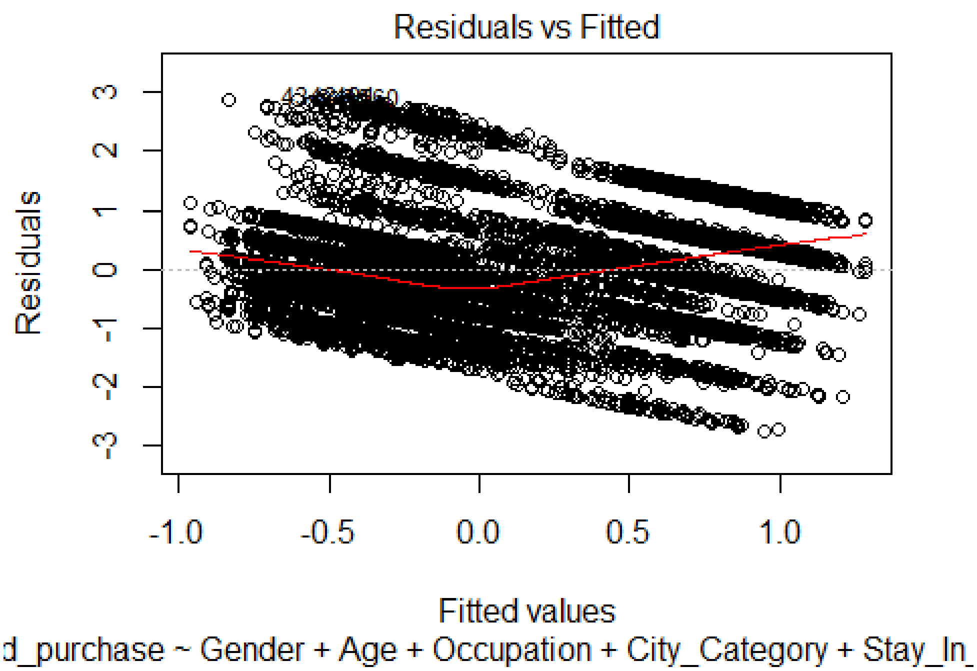
```
r2 <- lm(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status +
           poly(Product_Category_1, 2) + Product_Category_2 + Product_C
ategory_3, data = bf)
summary(r2)
```

```
##
## Call:
## lm(formula = sd_purchase ~ Gender + Age + Occupation + City_Category
 +
##     Stay_In_Current_City_Years + Marital_Status + poly(Product_Categ
ory_1,
##     2) + Product_Category_2 + Product_Category_3, data = bf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.74229 -0.51833 -0.05255  0.44591  2.93435
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)

## (Intercept)                      -0.453566   0.110859  -4.091 4.32e-05
***
## GenderM                           0.041606   0.022419   1.856 0.063503 .
```

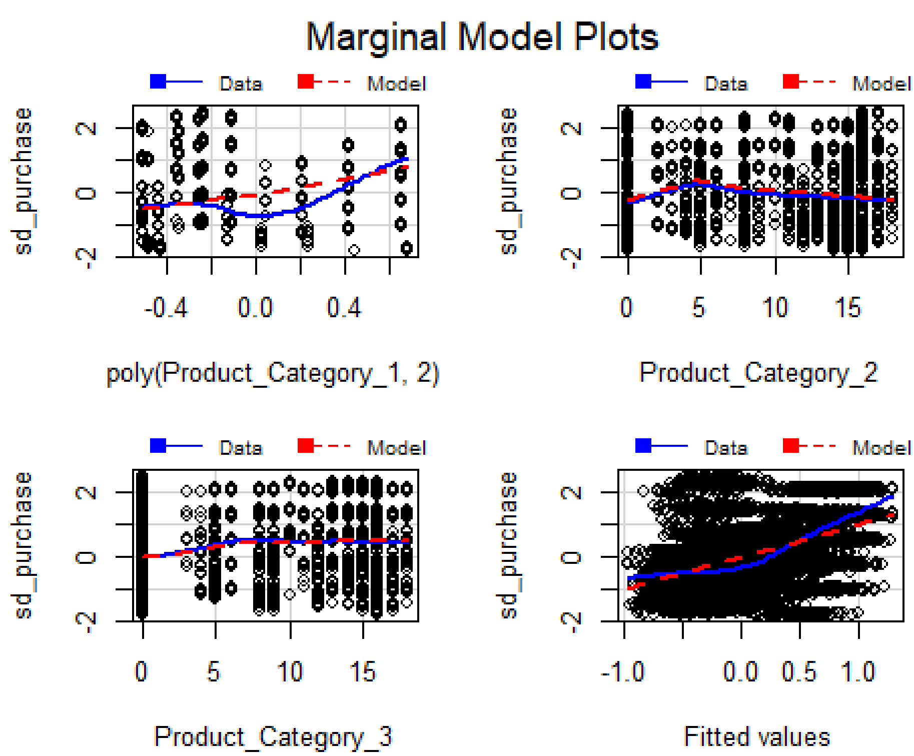
```
## Age18-25                          0.201618   0.104761   1.925 0.054309 .
## Age26-35                          0.307256   0.103621   2.965 0.003031
**
## Age36-45                          0.367681   0.104629   3.514 0.000443
***
## Age46-50                          0.395998   0.106663   3.713 0.000206
***
## Age51-55                          0.360201   0.106699   3.376 0.000738
***
## Age55+                            0.288440   0.109388   2.637 0.008379
**
## Occupation1                      -0.053911   0.038723  -1.392 0.163884

## Occupation2                       0.169551   0.053416   3.174 0.001506
**
## Occupation3                       0.151469   0.042795   3.539 0.000403
***
## Occupation4                       0.052545   0.035808   1.467 0.142293

## Occupation5                      -0.014191   0.062444  -0.227 0.820231

## Occupation6                      -0.115805   0.044470  -2.604 0.009224
**
## Occupation7                       0.200441   0.038747   5.173 2.34e-07
***
## Occupation8                       0.195598   0.171504   1.140 0.254105

## Occupation9                      -0.392450   0.071436  -5.494 4.02e-08
***
## Occupation10                      0.405663   0.110424   3.674 0.000240
***
## Occupation11                      0.144865   0.057883   2.503 0.012338
*
## Occupation12                      0.116785   0.046823   2.494 0.012638
*
## Occupation13                      0.188185   0.069074   2.724 0.006451
**
## Occupation14                     -0.156414   0.059147  -2.644 0.008193
**
## Occupation15                      0.246030   0.086748   2.836 0.004574
**
## Occupation16                      0.177436   0.045583   3.893 9.97e-05
***
## Occupation17                     -0.059197   0.044613  -1.327 0.184575

## Occupation18                     -0.512082   0.176699  -2.898 0.003762
**
```

```
## Occupation19                    -0.127706   0.059983  -2.129 0.033273
*
## Occupation20                     0.063909   0.043643   1.464 0.143117

## City_CategoryB                   0.054811   0.026206   2.092 0.036499
*
## City_CategoryC                   0.163833   0.026508   6.180 6.60e-10
***
## Stay_In_Current_City_Years1   -0.123720   0.029699  -4.166 3.12e-05
***
## Stay_In_Current_City_Years2   -0.247695   0.034015  -7.282 3.50e-13
***
## Stay_In_Current_City_Years3    0.012222   0.037610   0.325 0.745209

## Stay_In_Current_City_Years4+  -0.127412   0.032537  -3.916 9.06e-05
***
## Marital_Status1                  0.057078   0.021328   2.676 0.007456
**
## poly(Product_Category_1, 2)1 -32.517846   0.950386 -34.215  < 2e-16
***
## poly(Product_Category_1, 2)2  32.114361   0.909790  35.299  < 2e-16
***
## Product_Category_2               0.004900   0.001297   3.779 0.000158
***
## Product_Category_3               0.014899   0.001466  10.162  < 2e-16
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8657 on 11735 degrees of freedom
## Multiple R-squared:  0.253,  Adjusted R-squared:  0.2506
## F-statistic: 104.6 on 38 and 11735 DF,  p-value: < 2.2e-16
```

```
#round(r2$coefficients, digits = 2)
# Residual Plot
plot(r2, which = 1)
```




```r
# Marginal model plots
library(car)
marginalModelPlots(r2)
```



From the marginal plots, we can see there still exists a big discrepency between the linear regression line and actual data line.

3.3 Linear regression model with interaction

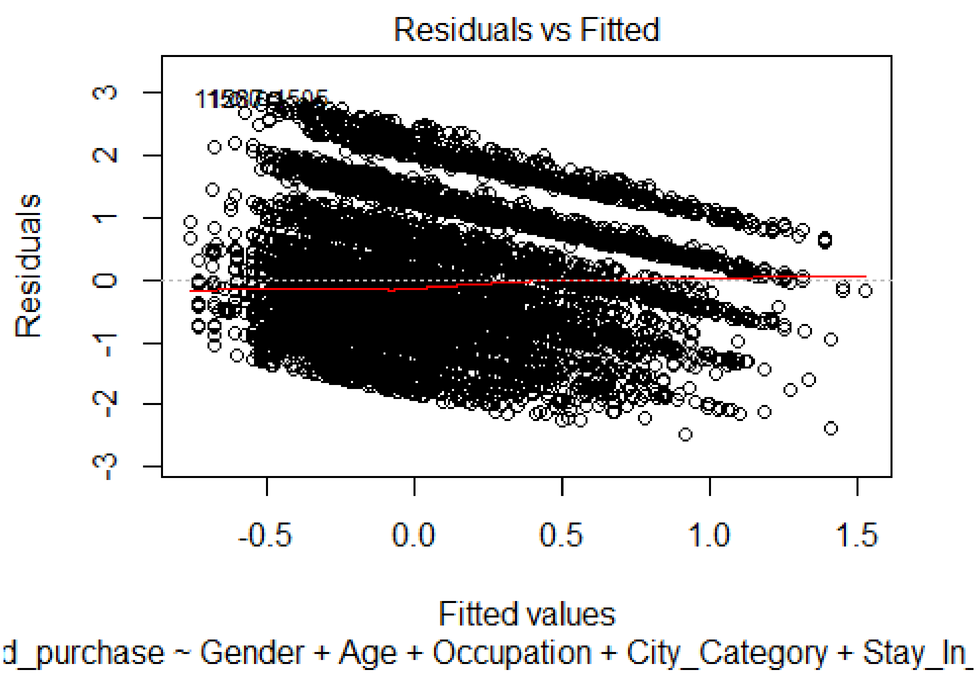
```
r3 <- lm(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status +
           Product_Category_2*Product_Category_3, data = bf)
summary(r3)
```

```
##
## Call:
## lm(formula = sd_purchase ~ Gender + Age + Occupation + City_Category
 +
##     Stay_In_Current_City_Years + Marital_Status + Product_Category_2
 *
##     Product_Category_3, data = bf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4599 -0.6001 -0.1252  0.4625  2.9105
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                            -0.5968124  0.1192921  -5.003
## GenderM                                 0.1095524  0.0240641   4.553
## Age18-25                                0.1010016  0.1127024   0.896
## Age26-35                                0.2081455  0.1114846   1.867
## Age36-45                                0.1985239  0.1125249   1.764
## Age46-50                                0.2779547  0.1147222   2.423
## Age51-55                                0.1934095  0.1147555   1.685
## Age55+                                  0.1134055  0.1176345   0.964
## Occupation1                            -0.0789066  0.0416676  -1.894
## Occupation2                             0.2033489  0.0574744   3.538
## Occupation3                             0.1934666  0.0460385   4.202
## Occupation4                             0.0643216  0.0385364   1.669
## Occupation5                            -0.0577872  0.0671882  -0.860
## Occupation6                            -0.0584319  0.0478387  -1.221
## Occupation7                             0.2409701  0.0416848   5.781
## Occupation8                             0.8017906  0.1839544   4.359
## Occupation9                            -0.2959337  0.0768136  -3.853
## Occupation10                            0.3004813  0.1187996   2.529
## Occupation11                            0.1007002  0.0622710   1.617
## Occupation12                            0.2188534  0.0503293   4.348
## Occupation13                            0.2252102  0.0742953   3.031
## Occupation14                           -0.2230410  0.0635928  -3.507
## Occupation15                            0.1314520  0.0932703   1.409
## Occupation16                            0.2098250  0.0490483   4.278
## Occupation17                           -0.0269722  0.0480041  -0.562
## Occupation18                           -0.3777800  0.1901224  -1.987
## Occupation19                           -0.1444966  0.0645479  -2.239
```

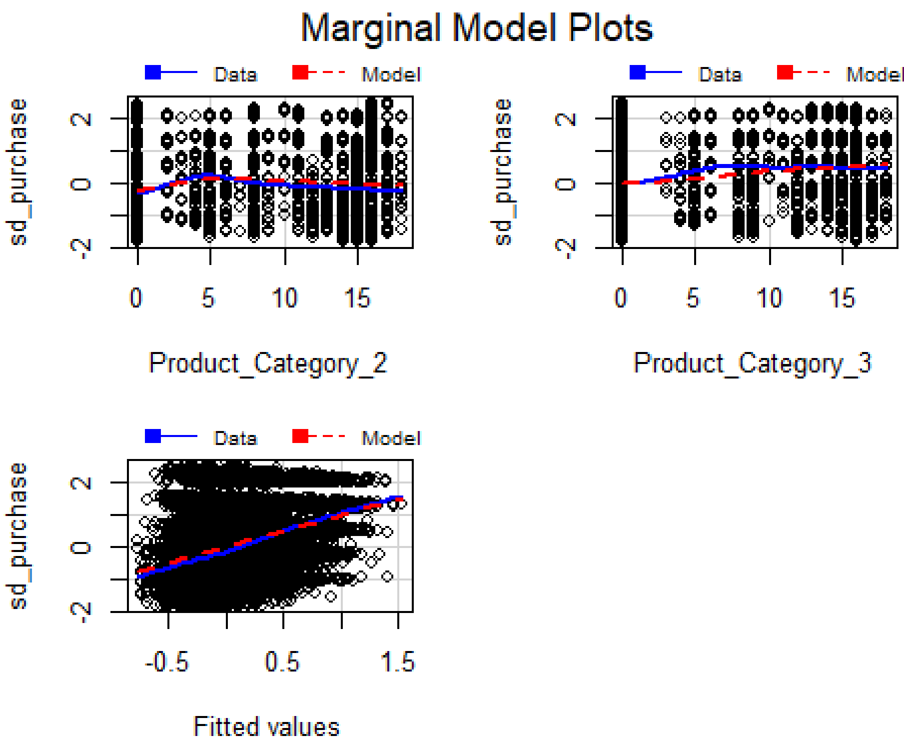
```
## Occupation20                                 0.0666358  0.0469624   1.419
## City_CategoryB                               0.0334383  0.0281957   1.186
## City_CategoryC                               0.1666766  0.0285048   5.847
## Stay_In_Current_City_Years1                 -0.0788085  0.0319246  -2.469
## Stay_In_Current_City_Years2                 -0.1719751  0.0365523  -4.705
## Stay_In_Current_City_Years3                  0.0685541  0.0404407   1.695
## Stay_In_Current_City_Years4+                -0.1034170  0.0349918  -2.955
## Marital_Status1                              0.0820674  0.0229345   3.578
## Product_Category_2                           0.0112335  0.0014890   7.545
## Product_Category_3                           0.0835368  0.0027370  30.521
## Product_Category_2:Product_Category_3 -0.0047240  0.0002797 -16.893
##                                        Pr(>|t|)
## (Intercept)                            5.73e-07 ***
## GenderM                                5.35e-06 ***
## Age18-25                               0.370175
## Age26-35                               0.061922 .
## Age36-45                               0.077713 .
## Age46-50                               0.015414 *
## Age51-55                               0.091937 .
## Age55+                                 0.335041
## Occupation1                            0.058287 .
## Occupation2                            0.000405 ***
## Occupation3                            2.66e-05 ***
## Occupation4                            0.095122 .
## Occupation5                            0.389763
## Occupation6                            0.221945
## Occupation7                            7.63e-09 ***
## Occupation8                            1.32e-05 ***
## Occupation9                            0.000117 ***
## Occupation10                           0.011442 *
## Occupation11                           0.105878
## Occupation12                           1.38e-05 ***
## Occupation13                           0.002440 **
## Occupation14                           0.000454 ***
## Occupation15                           0.158753
## Occupation16                           1.90e-05 ***
## Occupation17                           0.574213
## Occupation18                           0.046942 *
## Occupation19                           0.025201 *
## Occupation20                           0.155949
## City_CategoryB                         0.235671
## City_CategoryC                         5.13e-09 ***
## Stay_In_Current_City_Years1            0.013579 *
## Stay_In_Current_City_Years2            2.57e-06 ***
## Stay_In_Current_City_Years3            0.090069 .
## Stay_In_Current_City_Years4+           0.003128 **
## Marital_Status1                        0.000347 ***
## Product_Category_2                     4.87e-14 ***
## Product_Category_3                      < 2e-16 ***
## Product_Category_2:Product_Category_3  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9315 on 11736 degrees of freedom
## Multiple R-squared:  0.1349, Adjusted R-squared:  0.1322
## F-statistic: 49.48 on 37 and 11736 DF,  p-value: < 2.2e-16
```

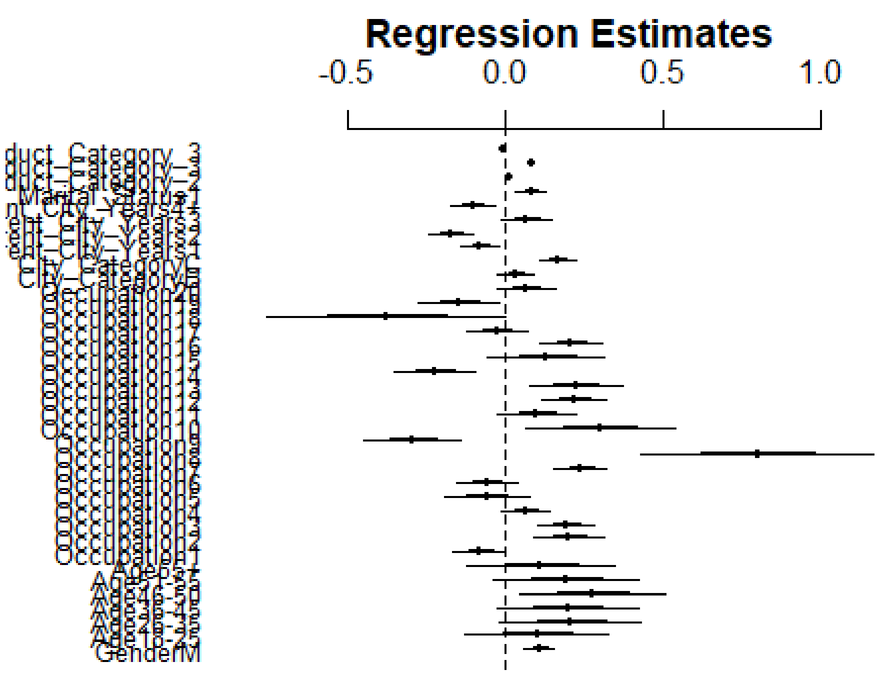
```r
#round(r2$coefficients, digits = 2)
# Residual Plot
plot(r3, which = 1)
```




```r
# Marginal model plots
library(car)
marginalModelPlots(r3)
```



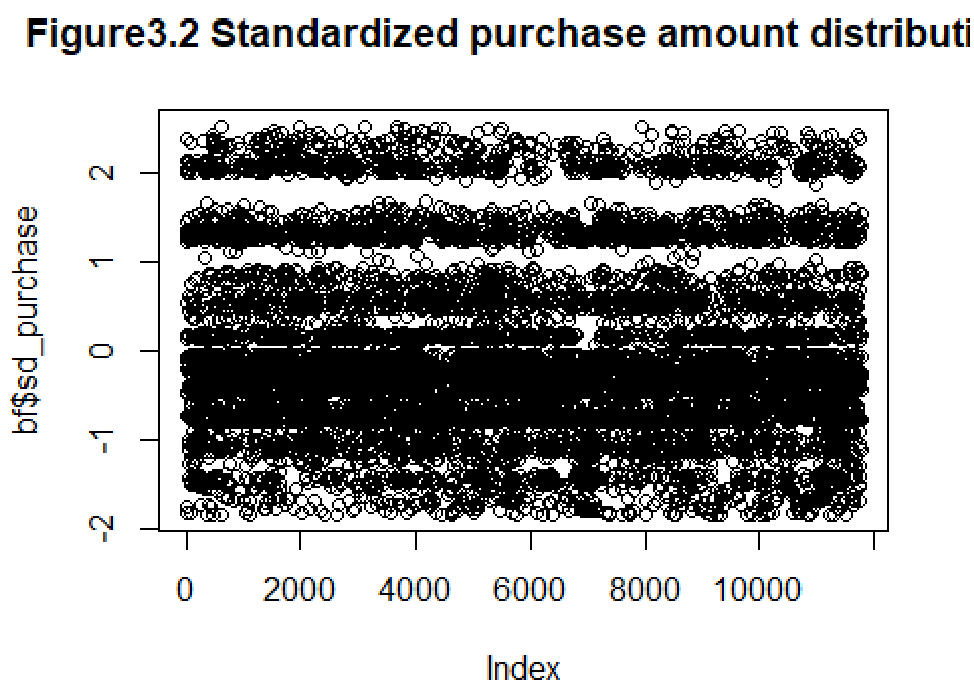

```r
# Coefficient plots
arm::coefplot(r3)
```



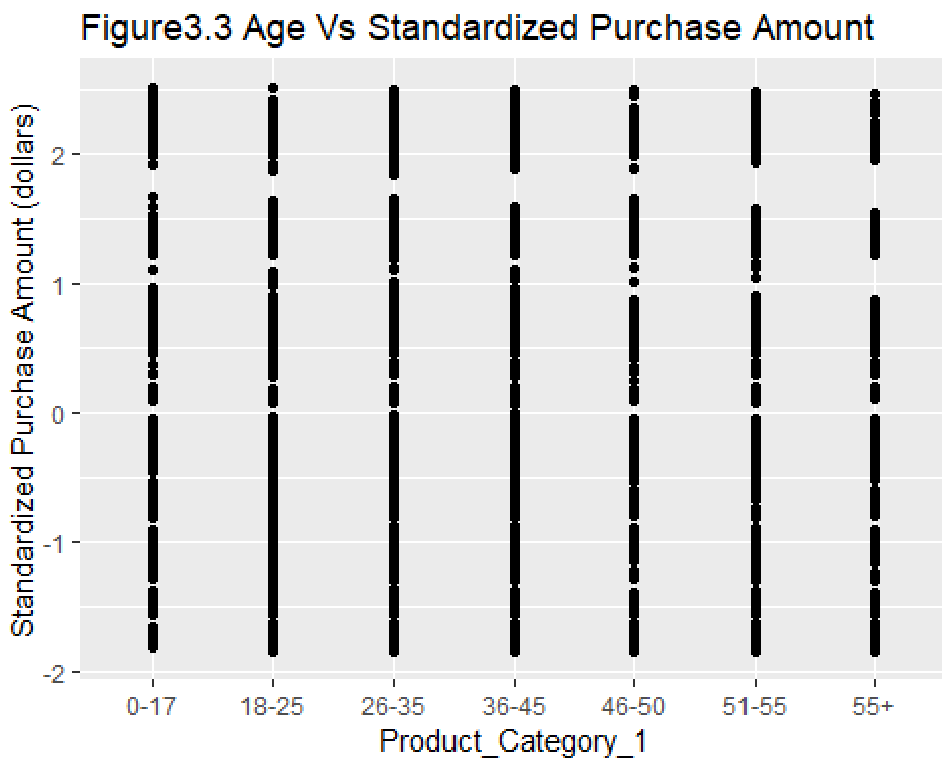
We can see after adding the interaction, the residual plot is a little better than before but there still exists a decreasing trend. Besides, almost half of coefficients are not significant. Therefore, this model cannot fit the data very well.

3.4 Cumulative logit model

```
plot(bf$sd_purchase, main = "Figure3.2 Standardized purchase amount dis
tribution")
```




```
ggplot(bf)+geom_point()+aes(x=Age,y=sd_purchase)+
  labs(y="Standardized Purchase Amount (dollars)",
       x="Product_Category_1",
       title="Figure3.3 Age Vs Standardized Purchase Amount")
```



From figure 3.2 and 3.3, we can see the standardized purchase amount have some gaps although it is a continuous variable. Therefore, we will divide the value of standardized purchase amount into several categories. Therefore, standardized purchase amount is transformed to ordinal variable based on its quantiles. We do this transformation both in train and test datasets.

```
# Transform sd_purchase into ordinal variable
# Train dataset
quan <- quantile(bf$sd_purchase)

bf <- bf%>%
  mutate(purchase_level=case_when(sd_purchase <= quan[2] ~"Low",
                                  sd_purchase > quan[2] & sd_purchase <
= quan[3] ~ "Somewhat Low",
                                  sd_purchase > quan[3] & sd_purchase <
 quan[4] ~ "Somewhat High",
                                  sd_purchase >= quan[4] ~"High"))

bf$purchase_level <- factor(bf$purchase_level,
                            levels=c("Low", "Somewhat Low", "Somewhat H
igh", "High"), ordered=TRUE)
# Test dataset
bf_test$sd_purchase <- (bf_test$Purchase-mean(bf_test$Purchase))/sd(bf_
test$Purchase)
bf_test <- bf_test %>%
  mutate(purchase_level=case_when(sd_purchase <= quan[2] ~"Low",
```

```
                                    sd_purchase > quan[2] & sd_purchase <
= quan[3] ~ "Somewhat Low",
                                    sd_purchase > quan[3] & sd_purchase <
 quan[4] ~ "Somewhat High",
                                    sd_purchase >= quan[4] ~"High"))

bf_test$purchase_level <- factor(bf_test$purchase_level,
                            levels=c("Low", "Somewhat Low", "Somewhat H
igh", "High"), ordered=TRUE)
```

The new response variable called **purchase_level** has 4 categories including "Low", "Somewhat Low", "Somewhat High" and "High". Then we began to make ordinal logit model.

```
library(arm)
r4 <- polr(purchase_level ~ Gender + Age + Occupation + City_Category +

           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
           Product_Category_2*Product_Category_3, data = bf)
summary(r4)

## Call:
## polr(formula = purchase_level ~ Gender + Age + Occupation + City_Cat
egory +
##     Stay_In_Current_City_Years + Marital_Status + Product_Category_1
 +
##     Product_Category_2 * Product_Category_3, data = bf)
##
## Coefficients:
##                                            Value Std. Error   t value
## GenderM                                 0.111920  0.0469592   2.38335
## Age18-25                                0.324707  0.2282511   1.42259
## Age26-35                                0.560270  0.2257973   2.48130
## Age36-45                                0.606368  0.2277330   2.66263
## Age46-50                                0.734403  0.2319731   3.16590
## Age51-55                                0.635470  0.2319138   2.74011
## Age55+                                  0.597281  0.2370038   2.52013
## Occupation1                            -0.087892  0.0815395  -1.07791
## Occupation2                             0.366988  0.1116187   3.28787
## Occupation3                             0.385501  0.0905634   4.25670
## Occupation4                             0.145758  0.0763458   1.90918
## Occupation5                            -0.065343  0.1321476  -0.49447
## Occupation6                            -0.213358  0.0959378  -2.22392
## Occupation7                             0.436931  0.0828305   5.27499
## Occupation8                             1.156594  0.4085247   2.83115
## Occupation9                            -0.869321  0.1531999  -5.67442
## Occupation10                            0.801912  0.2398644   3.34319
## Occupation11                            0.297509  0.1175505   2.53091
## Occupation12                            0.321667  0.1007869   3.19156
```

```
## Occupation13                               0.542395  0.1424050   3.80882
## Occupation14                              -0.347685  0.1276657  -2.72340
## Occupation15                               0.232124  0.1767304   1.31344
## Occupation16                               0.438382  0.0955651   4.58726
## Occupation17                               0.003665  0.0939162   0.03902
## Occupation18                              -0.959129  0.3981694  -2.40885
## Occupation19                              -0.173385  0.1274615  -1.36029
## Occupation20                               0.167378  0.0924104   1.81125
## City_CategoryB                             0.026563  0.0546965   0.48564
## City_CategoryC                             0.322640  0.0556907   5.79342
## Stay_In_Current_City_Years1               -0.323042  0.0623569  -5.18053
## Stay_In_Current_City_Years2               -0.510010  0.0718127  -7.10195
## Stay_In_Current_City_Years3                0.006671  0.0789741   0.08448
## Stay_In_Current_City_Years4+              -0.306791  0.0679372  -4.51580
## Marital_Status1                            0.200762  0.0451879   4.44283
## Product_Category_1                        -0.139601  0.0057704 -24.19240
## Product_Category_2                         0.011483  0.0028752   3.99378
## Product_Category_3                         0.098684  0.0063147  15.62768
## Product_Category_2:Product_Category_3 -0.005307  0.0005923  -8.95921
##
## Intercepts:
##                             Value    Std. Error t value
## Low|Somewhat Low            -1.0806   0.2429    -4.4491
## Somewhat Low|Somewhat High   0.1749   0.2426     0.7207
## Somewhat High|High           1.4464   0.2430     5.9522
##
## Residual Deviance: 30415.49
## AIC: 30497.49
```

```r
ctable <- coef(summary(r4))
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
ctable <- cbind(ctable, "p value" = p)
ctable
```

```
##                                              Value   Std. Error
## GenderM                                0.111919984 0.0469591520
## Age18-25                               0.324707498 0.2282510575
## Age26-35                               0.560269875 0.2257972931
## Age36-45                               0.606368362 0.2277329895
## Age46-50                               0.734402878 0.2319730867
## Age51-55                               0.635469531 0.2319137552
## Age55+                                 0.597281206 0.2370038119
## Occupation1                           -0.087892334 0.0815395153
## Occupation2                            0.366987830 0.1116187313
## Occupation3                            0.385500943 0.0905634087
## Occupation4                            0.145758107 0.0763458091
## Occupation5                           -0.065343097 0.1321475608
## Occupation6                           -0.213358094 0.0959378268
## Occupation7                            0.436930571 0.0828305219
## Occupation8                            1.156593826 0.4085246782
```

```
## Occupation9                              -0.869321354 0.1531999156
## Occupation10                              0.801912345 0.2398644196
## Occupation11                              0.297509475 0.1175505341
## Occupation12                              0.321667006 0.1007868736
## Occupation13                              0.542394871 0.1424049532
## Occupation14                             -0.347685051 0.1276657474
## Occupation15                              0.232124081 0.1767303706
## Occupation16                              0.438381571 0.0955650903
## Occupation17                              0.003664833 0.0939162272
## Occupation18                             -0.959128521 0.3981693954
## Occupation19                             -0.173384878 0.1274614841
## Occupation20                              0.167378326 0.0924104135
## City_CategoryB                            0.026562848 0.0546964550
## City_CategoryC                            0.322639782 0.0556907311
## Stay_In_Current_City_Years1              -0.323041737 0.0623568793
## Stay_In_Current_City_Years2              -0.510009900 0.0718126984
## Stay_In_Current_City_Years3               0.006671350 0.0789741185
## Stay_In_Current_City_Years4+             -0.306791135 0.0679372286
## Marital_Status1                           0.200762468 0.0451879304
## Product_Category_1                       -0.139600866 0.0057704439
## Product_Category_2                        0.011483108 0.0028752475
## Product_Category_3                        0.098683838 0.0063146832
## Product_Category_2:Product_Category_3 -0.005306624 0.0005923092
## Low|Somewhat Low                         -1.080582446 0.2428752767
## Somewhat Low|Somewhat High                0.174861595 0.2426409026
## Somewhat High|High                        1.446441364 0.2430107906
##                                              t value      p value
## GenderM                                   2.38334763  1.715599e-02
## Age18-25                                  1.42258924  1.548553e-01
## Age26-35                                  2.48129580  1.309057e-02
## Age36-45                                  2.66262856  7.753295e-03
## Age46-50                                  3.16589691  1.546056e-03
## Age51-55                                  2.74011143  6.141836e-03
## Age55+                                    2.52013333  1.173104e-02
## Occupation1                              -1.07791093  2.810735e-01
## Occupation2                               3.28786957  1.009486e-03
## Occupation3                               4.25669648  2.074697e-05
## Occupation4                               1.90918282  5.623851e-02
## Occupation5                              -0.49447070  6.209738e-01
## Occupation6                              -2.22392044  2.615380e-02
## Occupation7                               5.27499478  1.327599e-07
## Occupation8                               2.83114800  4.638125e-03
## Occupation9                              -5.67442450  1.391557e-08
## Occupation10                              3.34319007  8.282115e-04
## Occupation11                              2.53090705  1.137680e-02
## Occupation12                              3.19155654  1.415084e-03
## Occupation13                              3.80882026  1.396314e-04
## Occupation14                             -2.72340121  6.461354e-03
## Occupation15                              1.31343628  1.890360e-01
## Occupation16                              4.58725639  4.491090e-06
```

```
## Occupation17                                0.03902236  9.688726e-01
## Occupation18                               -2.40884541  1.600307e-02
## Occupation19                               -1.36029232  1.737374e-01
## Occupation20                                1.81124961  7.010222e-02
## City_CategoryB                              0.48564113  6.272216e-01
## City_CategoryC                              5.79341977  6.896744e-09
## Stay_In_Current_City_Years1                -5.18053084  2.212553e-07
## Stay_In_Current_City_Years2                -7.10194591  1.230124e-12
## Stay_In_Current_City_Years3                 0.08447515  9.326787e-01
## Stay_In_Current_City_Years4+               -4.51580291  6.307731e-06
## Marital_Status1                             4.44283387  8.878171e-06
## Product_Category_1                        -24.19239613 2.674950e-129
## Product_Category_2                          3.99378087  6.502796e-05
## Product_Category_3                         15.62767836  4.716779e-55
## Product_Category_2:Product_Category_3      -8.95921266  3.270049e-19
## Low|Somewhat Low                           -4.44912492  8.622086e-06
## Somewhat Low|Somewhat High                  0.72066001  4.711187e-01
## Somewhat High|High                          5.95216929  2.646115e-09
```

From the summary result and the table above, we can see the value of AIC is a little big although most of coefficients are significant. Then we will use test dataset to make predictions in order to check the accuracy of the model.

```
# Prediction in test dataset
predict.purchase <- predict(r4, bf_test)  # predict the classes directl
y
head(predict.purchase)

## [1] High High High High High High
## Levels: Low Somewhat Low Somewhat High High

predicted.prop <- predict(r4, bf_test, type="p")  # predict the probabi
lites
head(predicted.prop)

##          Low Somewhat Low Somewhat High      High
## 1 0.08172412    0.1562707     0.2889479 0.4730573
## 2 0.08172412    0.1562707     0.2889479 0.4730573
## 3 0.05053209    0.1068484     0.2424265 0.6001930
## 4 0.04991327    0.1057545     0.2410303 0.6033020
## 5 0.10448582    0.1860254     0.3030428 0.4064460
## 6 0.10173817    0.1826872     0.3019411 0.4136335

# Build a confusion matrix
table(predict.purchase, bf_test$purchase_level)

##
## predict.purchase Low Somewhat Low Somewhat High High
##    Low           592          509           293  194
##    Somewhat Low  274          369           298  100
```

```
##    Somewhat High 229          286           262  193
##    High          155          155           367  773
```

```r
# Compute the misclassification error rate of prediction
mean(as.character(predict.purchase) != as.character(bf_test$purchase_le
vel))
```

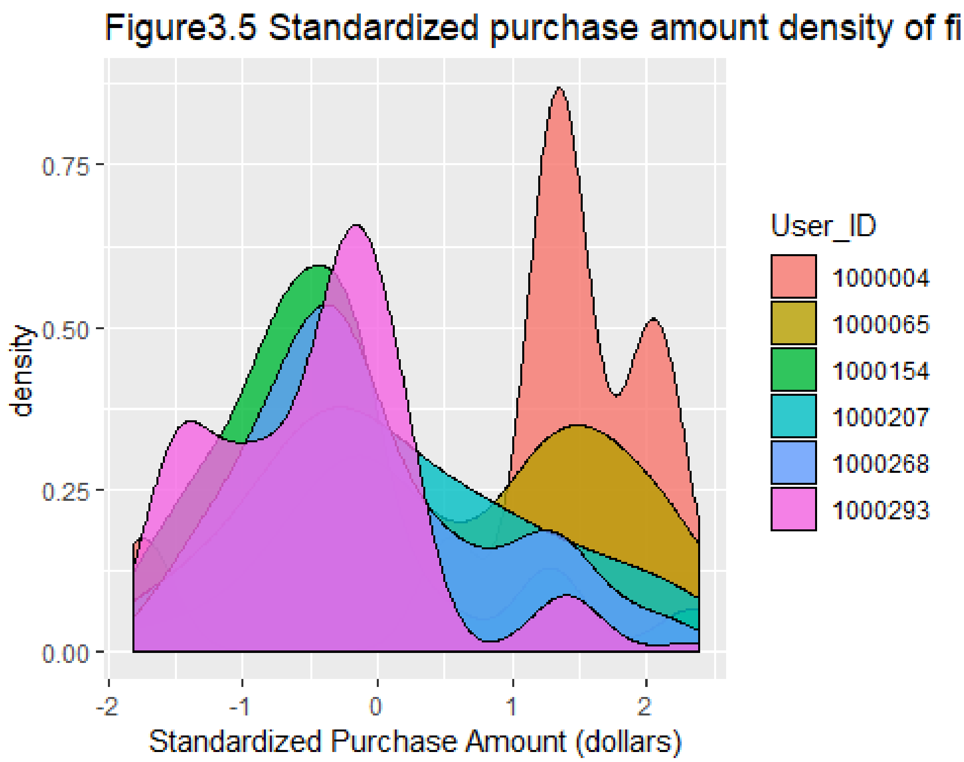
```
## [1] 0.6046742
```

A misclassification error of 60.47% is probably too high. Maybe it can be improved by trying Multilevel model to improve the accuracy.

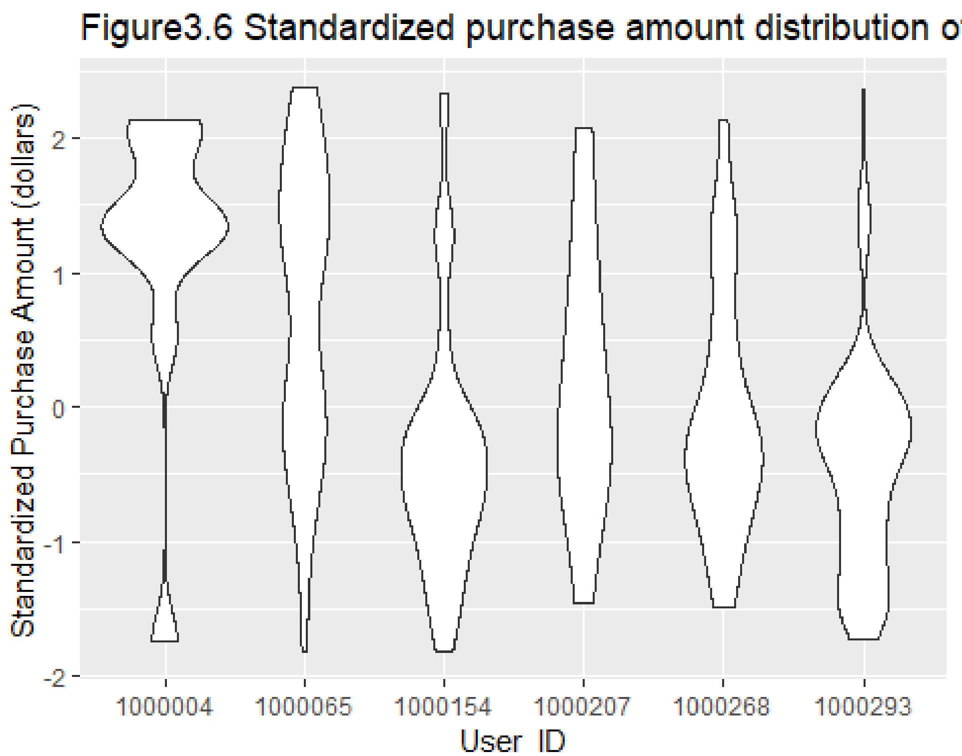
3.5 Mixed Effects Model

With this black friday retail dataset, since each User_ID has multiple purchase records, we can immediately see that this would violate the independence assumption that's important in linear modeling, which is to say multiple purchase records from the same User_ID cannot be regarded as independent from each other. Besides, in our scenario, every User_ID has a slightly different consumption habit, and this is going to be an idiosyncratic factor that affects the measurements from the different User_IDs.

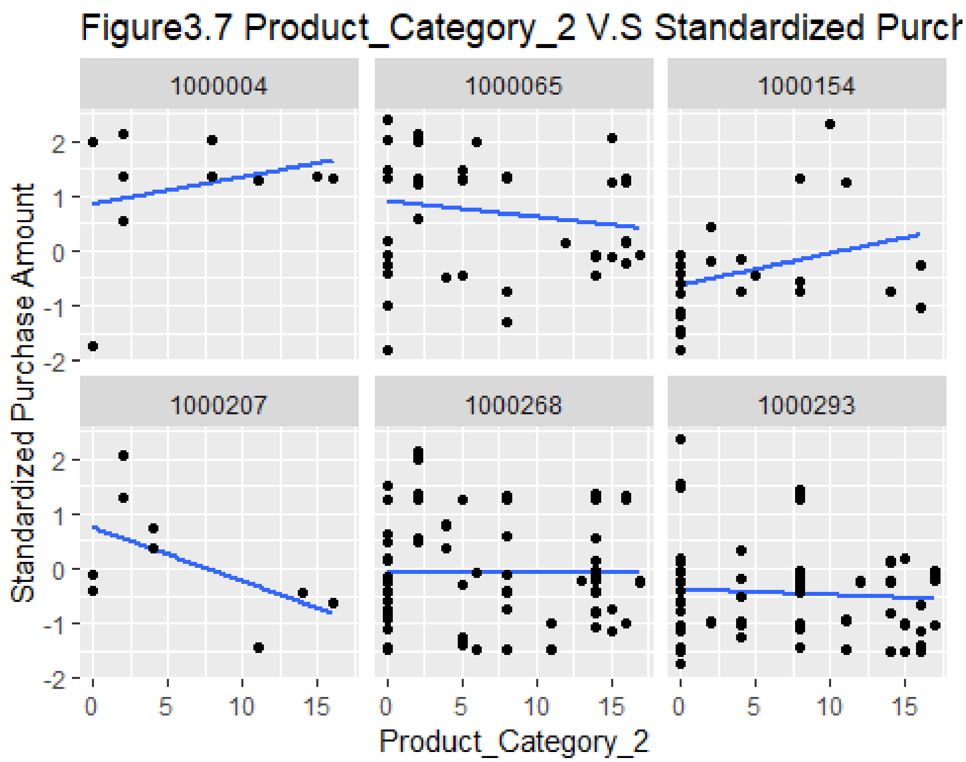
```r
ggplot(bf[1:313,], aes(sd_purchase)) +
  geom_density(aes(fill=factor(User_ID)), alpha=0.8) +
    labs(title="Figure3.5 Standardized purchase amount density of first
 six User_ID",
         x="Standardized Purchase Amount (dollars)",
         fill="User_ID")
```




```r
ggplot(bf[1:313,], aes(User_ID, sd_purchase)) +
  geom_violin() +
    labs(title="Figure3.6 Standardized purchase amount distribution of first six User_ID",
         x="User_ID",
         y="Standardized Purchase Amount (dollars)")
```




```
ggplot(bf[1:313,]) +
  aes(x = Product_Category_2, y = sd_purchase) +
  stat_smooth(method = "lm", se = FALSE) +
  geom_point() +
  facet_wrap("User_ID") +
  labs(title = "Figure3.7 Product_Category_2 V.S Standardized Purchase
Amount by User_ID",
       x = "Product_Category_2",
       y = "Standardized Purchase Amount")
```



From above three figures, we can see there are big differences in standardized purchase amount between groups. Therefore, in order to consider the differences among both individuals (each purchase) and groups (each User_ID), we should then fit the multilevel model.

Individual level variables include **Product_Category_1**, **Product_Category_2** and **Product_Category_3**. Group level variables include **Gender**, **Age**, **Occupation**, **City_Category**, **Stay_In_Current_City_Years**, **Marital_Status**.

3.5.1 Mixed Effects Model (vary by intercept)

First, we fit a mixed effects model with varying intercepts by groups (User_ID).

```
# Remove a variable
r5_0 <- lmer(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
             Product_Category_2 + Product_Category_3 + (1|User_ID), dat
a = bf)
r5_1 <- lmer(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
             Product_Category_2 + (1|User_ID), data = bf)
r5_2 <- lmer(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
             Product_Category_3 + (1|User_ID), data = bf)
```

```
r5_3 <- lmer(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status +
             Product_Category_2 + Product_Category_3 + (1|User_ID), dat
a = bf)
# Model choice
anova(r5_0, r5_1, r5_2, r5_3)

## Data: bf
## Models:
## r5_1: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_1:     Marital_Status + Product_Category_1 + Product_Category_2 +

## r5_1:     (1 | User_ID)
## r5_2: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_2:     Marital_Status + Product_Category_1 + Product_Category_3 +

## r5_2:     (1 | User_ID)
## r5_3: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_3:     Marital_Status + Product_Category_2 + Product_Category_3 +

## r5_3:     (1 | User_ID)
## r5_0: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_0:     Marital_Status + Product_Category_1 + Product_Category_2 +

## r5_0:     Product_Category_3 + (1 | User_ID)
##      Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## r5_1 39 31132 31420 -15527    31054
## r5_2 39 30825 31113 -15374    30747 306.91      0     <2e-16 ***
## r5_3 39 31664 31952 -15793    31587   0.00      0          1
## r5_0 40 30827 31122 -15374    30747 839.32      1     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(r5_2, r5_0)

## Data: bf
## Models:
## r5_2: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_2:     Marital_Status + Product_Category_1 + Product_Category_3 +

## r5_2:     (1 | User_ID)
## r5_0: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_0:     Marital_Status + Product_Category_1 + Product_Category_2 +
```

```
## r5_0:     Product_Category_3 + (1 | User_ID)
##      Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## r5_2 39 30825 31113 -15374    30747
## r5_0 40 30827 31122 -15374    30747 0.0677      1     0.7947

# Add interaction
r5 <- lmer(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
             Product_Category_2*Product_Category_3 + (1|User_ID), data
= bf)
# Model Choice
anova(r5, r5_0)

## Data: bf
## Models:
## r5_0: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay
_In_Current_City_Years +
## r5_0:     Marital_Status + Product_Category_1 + Product_Category_2 +

## r5_0:     Product_Category_3 + (1 | User_ID)
## r5: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay_I
n_Current_City_Years +
## r5:     Marital_Status + Product_Category_1 + Product_Category_2 *
## r5:     Product_Category_3 + (1 | User_ID)
##      Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## r5_0 40 30827 31122 -15374    30747
## r5   41 30738 31040 -15328    30656 91.296      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the results of anova method to compare models, we can see if we remove one continuous variable from r5_0 model, we should choose to remove **Product_Category_2** (in r5_2). However, when we campare r5_0 and r5_2 model, anova test shows p-value for chisq test is bigger than 0.05 so that we cannot reject the null hypothesis, which means the two models are equal in fitting the data. Then, we try to add the interaction between **Product_Category_2** and **Product_Category_3** in model r5, the anova test for r5 and r5_0 shows r5 model fits the data better because p-value is smaller than 0.05 and we reject the null hypothesis. Therefore, in the mixed effects model with varying intercepts, r5 model fits the data well.

```
summary(r5)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## sd_purchase ~ Gender + Age + Occupation + City_Category + Stay_In_Cu
rrent_City_Years +
##     Marital_Status + Product_Category_1 + Product_Category_2 *
##     Product_Category_3 + (1 | User_ID)
```

```
##    Data: bf
##
## REML criterion at convergence: 30800.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6362 -0.6514 -0.1439  0.4657  4.2884
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  User_ID  (Intercept) 0.08579  0.2929
##  Residual             0.77113  0.8781
## Number of obs: 11774, groups:  User_ID, 200
##
## Fixed effects:
##                                         Estimate Std. Error t value
## (Intercept)                           -0.1277536  0.2754686  -0.464
## GenderM                                0.0713082  0.0623061   1.144
## Age18-25                               0.1732779  0.2587341   0.670
## Age26-35                               0.2005021  0.2533916   0.791
## Age36-45                               0.1930804  0.2579900   0.748
## Age46-50                               0.3529137  0.2661166   1.326
## Age51-55                               0.2246492  0.2659012   0.845
## Age55+                                 0.1600292  0.2720784   0.588
## Occupation1                            0.0425328  0.1059412   0.401
## Occupation2                            0.2517649  0.1621506   1.553
## Occupation3                            0.0991652  0.1497995   0.662
## Occupation4                            0.1703617  0.1030540   1.653
## Occupation5                            0.0359152  0.1753035   0.205
## Occupation6                            0.1698595  0.1290777   1.316
## Occupation7                            0.3073702  0.0984003   3.124
## Occupation8                            0.5905539  0.3685750   1.602
## Occupation9                           -0.2494688  0.2082168  -1.198
## Occupation10                           0.3851687  0.2743397   1.404
## Occupation11                           0.2121830  0.1788698   1.186
## Occupation12                           0.2441199  0.1171056   2.085
## Occupation13                           0.2149167  0.2038736   1.054
## Occupation14                           0.0506720  0.1734661   0.292
## Occupation15                           0.1445001  0.2428778   0.595
## Occupation16                           0.2014946  0.1251018   1.611
## Occupation17                           0.0107928  0.1193190   0.090
## Occupation18                          -0.2155708  0.2906076  -0.742
## Occupation19                          -0.0068417  0.1672292  -0.041
## Occupation20                           0.1168973  0.1301609   0.898
## City_CategoryB                         0.1239789  0.0773280   1.603
## City_CategoryC                         0.2118447  0.0725018   2.922
## Stay_In_Current_City_Years1           -0.1678995  0.0803309  -2.090
## Stay_In_Current_City_Years2           -0.2097836  0.0943623  -2.223
## Stay_In_Current_City_Years3           -0.1681067  0.0988024  -1.701
## Stay_In_Current_City_Years4+          -0.2189166  0.0909213  -2.408
```

```
## Marital_Status1                          0.0567545  0.0583531   0.973
## Product_Category_1                       -0.0626410  0.0024142 -25.947
## Product_Category_2                        0.0048322  0.0014340   3.370
## Product_Category_3                        0.0493593  0.0028753  17.167
## Product_Category_2:Product_Category_3   -0.0026387  0.0002763  -9.550
```

3.5.2 Multilevel regression (varying slopes)

After trying many times, we can find that if slopes vary by City_Category, the model can converge. Therefore, the model is shown below.

```
r6 <- lmer(sd_purchase ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
             Product_Category_2 * Product_Category_3 + (1+City_Category
|User_ID), data = bf)
summary(r6)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## sd_purchase ~ Gender + Age + Occupation + City_Category + Stay_In_Cu
rrent_City_Years +
##     Marital_Status + Product_Category_1 + Product_Category_2 *
##     Product_Category_3 + (1 + City_Category | User_ID)
##    Data: bf
##
## REML criterion at convergence: 30800
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6360 -0.6529 -0.1438  0.4666  4.2923
##
## Random effects:
##  Groups   Name           Variance Std.Dev. Corr
##  User_ID  (Intercept)    0.08457  0.2908
##           City_CategoryB 0.08743  0.2957   -0.44
##           City_CategoryC 0.32598  0.5709   -1.00  0.38
##  Residual                0.77115  0.8782
## Number of obs: 11774, groups:  User_ID, 200
##
## Fixed effects:
##                                          Estimate Std. Error t value
## (Intercept)                             -0.1193725  0.2689593  -0.444
## GenderM                                  0.0664420  0.0621263   1.069
## Age18-25                                 0.1720903  0.2519040   0.683
## Age26-35                                 0.1950426  0.2464893   0.791
## Age36-45                                 0.1976701  0.2510279   0.787
## Age46-50                                 0.3539558  0.2596180   1.363
## Age51-55                                 0.2235689  0.2594797   0.862
## Age55+                                   0.1497742  0.2657466   0.564
```

```
## Occupation1                            0.0422204  0.1053743   0.401
## Occupation2                            0.2467850  0.1628264   1.516
## Occupation3                            0.0947221  0.1522543   0.622
## Occupation4                            0.1653018  0.1026344   1.611
## Occupation5                            0.0347985  0.1774367   0.196
## Occupation6                            0.1794325  0.1287735   1.393
## Occupation7                            0.3026962  0.0976689   3.099
## Occupation8                            0.5924061  0.3669616   1.614
## Occupation9                           -0.2408261  0.2121441  -1.135
## Occupation10                           0.3807634  0.2677598   1.422
## Occupation11                           0.2183752  0.1802703   1.211
## Occupation12                           0.2409959  0.1154049   2.088
## Occupation13                           0.2102231  0.2070237   1.015
## Occupation14                           0.0616765  0.1740060   0.354
## Occupation15                           0.1375440  0.2359916   0.583
## Occupation16                           0.1895165  0.1246229   1.521
## Occupation17                           0.0020896  0.1181165   0.018
## Occupation18                          -0.2237225  0.2845486  -0.786
## Occupation19                          -0.0171288  0.1675525  -0.102
## Occupation20                           0.1016737  0.1306304   0.778
## City_CategoryB                         0.1258347  0.0781556   1.610
## City_CategoryC                         0.2113777  0.0718111   2.944
## Stay_In_Current_City_Years1           -0.1686340  0.0802995  -2.100
## Stay_In_Current_City_Years2           -0.2076625  0.0944731  -2.198
## Stay_In_Current_City_Years3           -0.1709260  0.0980306  -1.744
## Stay_In_Current_City_Years4+          -0.2122288  0.0905475  -2.344
## Marital_Status1                        0.0563017  0.0586867   0.959
## Product_Category_1                    -0.0626430  0.0024142 -25.948
## Product_Category_2                     0.0048405  0.0014340   3.376
## Product_Category_3                     0.0493578  0.0028752  17.167
## Product_Category_2:Product_Category_3 -0.0026394  0.0002763  -9.553
```

Next, let's compare model r5 with r6 by anova test.

```
anova(r5, r6, refit=FALSE)

## Data: bf
## Models:
## r5: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay_I
n_Current_City_Years +
## r5:     Marital_Status + Product_Category_1 + Product_Category_2 *
## r5:     Product_Category_3 + (1 | User_ID)
## r6: sd_purchase ~ Gender + Age + Occupation + City_Category + Stay_I
n_Current_City_Years +
## r6:     Marital_Status + Product_Category_1 + Product_Category_2 *
## r6:     Product_Category_3 + (1 + City_Category | User_ID)
##    Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## r5 41 30882 31185 -15400    30800
## r6 46 30892 31231 -15400    30800 0.3101      5     0.9974

AIC(r5, r6)
```

```
##    df      AIC
## r5 41 30882.28
## r6 46 30891.97
```

We can see $\chi^2(5) = 0.3101, p = 0.9974$, which means we cannot reject the null hypothesis. That is to say, adding random slopes for each User_ID doesn't significantly improve model fit. Looking at the AIC values, AIC is higher for the more complex model (r6), so we want to go with the less complex (r5) model. In summary, it appears that we don't need to include random slopes for City_Category in the model.

4 Prediction and Discussion

4.1 Prediction

From part 3, we can finally decide to use model r5 to fit the train data. Then we can use it to make predictions in test datasets.

4.1.1 Prediction for Test Dataset

```
bf_test$Purchase.pre <- predict(r5, bf_test)*sd(bf$Purchase)+mean(bf$Pu
rchase)
head(bf_test$Purchase.pre)

## [1] 14073.80 14073.80 15386.94 15405.75 12703.16 12650.00
```

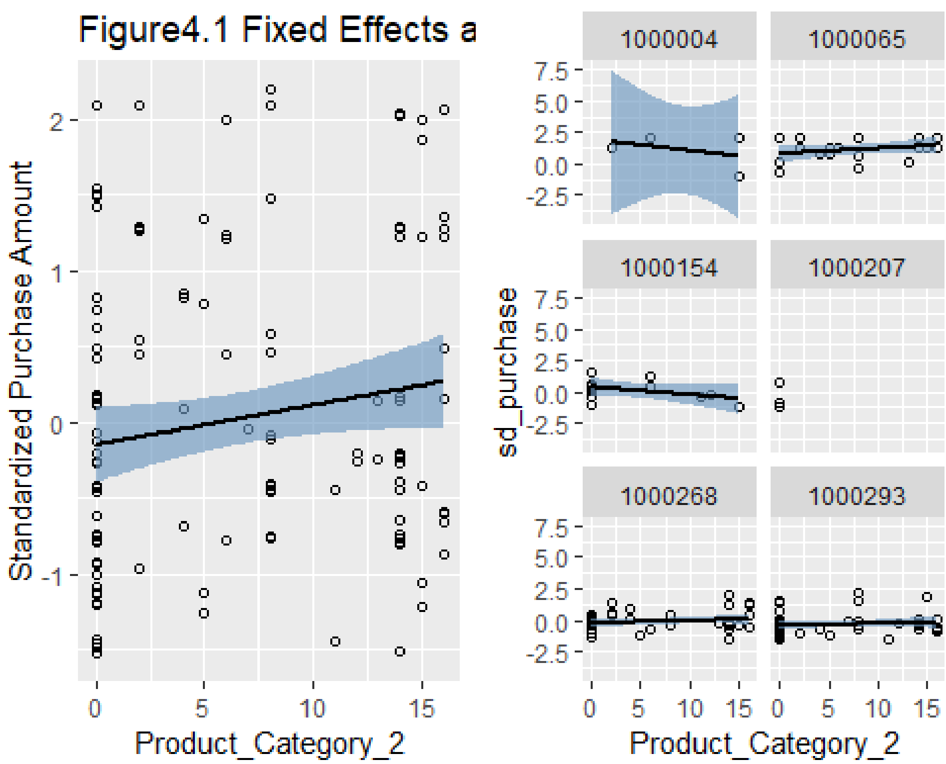
Using the model output, we can generate regression lines using the predict() function. Using this method, we can simply add a new column to the existing bf_test data frame, giving the fitted value for each row in the data. However, for visualization, it is very useful to generate the fitted values for specific combinations of predictor values, instead of generating a fitted value for every observation. To do this, I simply create dataframes with the relevant predictors, and feed these data frames as data to predict().

To get fitted values at the average level, we can just remove the User_ID. For the varying effects, we can create a data frame which include the User_ID. Both dataframes are selected from first 135 rows for bf_test dataset.

```
# Data frame to evaluate average effects predictions on
newavg <- bf_test[1:135,-1]
newavg$Reaction <- predict(r5, re.form = NA, newavg)
# Predictors for the varying effect's predictions
newvary <- bf_test[1:135,]
newvary$Reaction <- predict(r5, newvary)
```

On the left, a single fixed effects model versus the average regression line from the new multilevel model, and on the right the separate fixed effects models versus the varying regression lines from the multilevel model. Below, I use blue colors to indicate the fixed effects models' predictions, and black for the multilevel model's predictions.

```
p1 <- ggplot(bf_test[1:135,], aes(x = Product_Category_2, y = sd_purcha
se)) +
    geom_point(shape = 1) +
    geom_smooth(method = "lm", fill = "dodgerblue", level = .95)
p2 <- p1 + facet_wrap(~User_ID, nrow = 3)

pdp::grid.arrange(
    p1 + geom_smooth(data = newavg, method = "lm", color = "black", siz
e = 1)+
  labs(title = "Figure4.1 Fixed Effects and Predictions",
       x = "Product_Category_2",
       y = "Standardized Purchase Amount") ,
    p2 + geom_smooth(data = newvary, method = "lm", color = "black", si
ze = 1),
    ncol = 2)
```



As we can probably tell, the fixed effects regression line (blue), and the multilevel model's average regression line (black are nearly identical, because of the relatively balanced design.

4.1.2 Confidence interval-Average Level

The confidence interval reflects the uncertainty around the mean predictions. To display the 95% confidence intervals around the mean the predictions, specify the option interval = “confidence”:

The method I will illustrate relies on random samples of plausible parameter values, from which we can then generate regression lines or draw inferences about the parameters themselves. These regression lines can then be used as their own distribution with their own respective summaries, such as an X% interval.

The important parts of this code are:

1) Simulating plausible parameter values

2) Saving the simulated samples (a faux posterior distribution) in a data frame

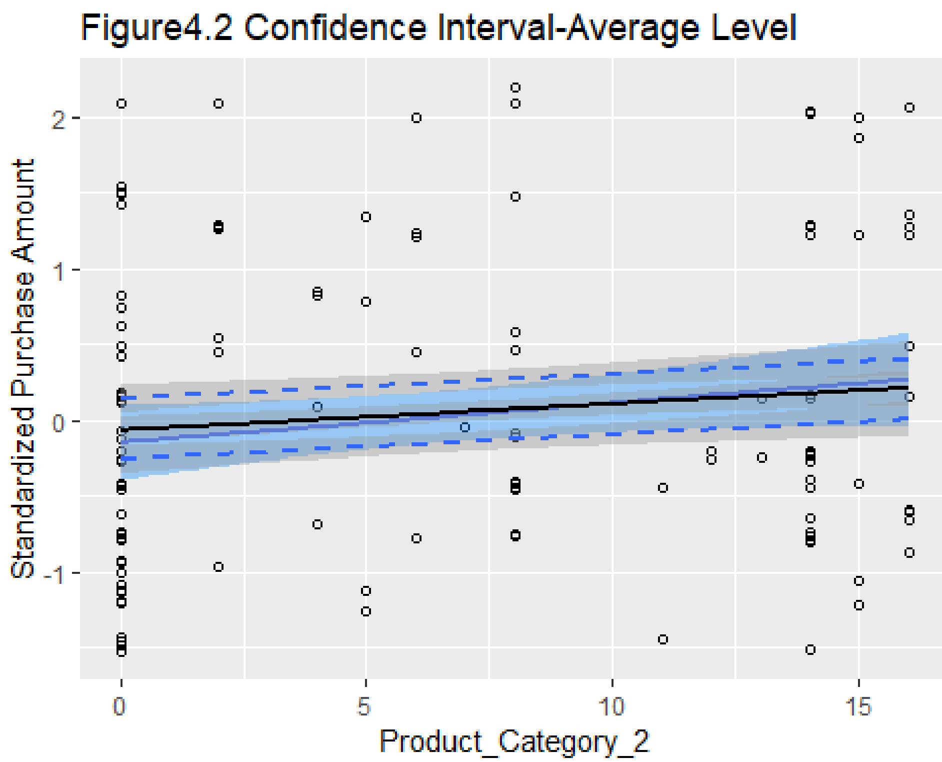
3) Creating a predictor matrix

4) Creating a matrix for the fitted values

5) Calculating fitted values for each combination of the predictor values, for each plausible combination of the parameter values

6) Calculating the desired quantiles of the fitted values

```
# Steps
sims <- sim(r5, n.sims = 135)  # 1
fs <- fixef(sims)  # 2
Xmat <- model.matrix( ~ Gender + Age + Occupation + City_Category +
           Stay_In_Current_City_Years + Marital_Status + Product_Catego
ry_1 +
             Product_Category_2*Product_Category_3, data = newavg)  # 3
fitmat <- matrix(ncol = nrow(fs), nrow = nrow(newavg))  # 4
for (i in 1:nrow(fs)) { fitmat[,i] <- Xmat %*% as.matrix(fs)[i,] }  # 5
newavg$lower <- apply(fitmat, 1, quantile, prob=0.05)  # 6
newavg$median <- apply(fitmat, 1, quantile, prob=0.5)  # 6
newavg$upper <- apply(fitmat, 1, quantile, prob=0.95)  # 6

# Plot
p1 + geom_smooth(data = newavg, aes(y = median), method = "lm", color =
 "black", size = 1) +
   geom_smooth(data = newavg, aes(y = lower), method = "lm", lty = 2)
+
   geom_smooth(data = newavg, aes(y = upper), method = "lm", lty = 2)
+
  labs(title = "Figure4.2 Confidence Interval-Average Level",
       x = "Product_Category_2",
       y = "Standardized Purchase Amount")
```



Again, the average regression line and the fixed effect model's regression line are nearly identical, but the former has a wider confidence interval (black dashed lines.)

4.2 Discussion

4.2.1 Implication

The goal of modeling here is to understand the customer purchase behaviour and forecast purchase amount of clients in the future so that the retail company can create personalized offer for customers.

Sales forecasting is a crucial part of the financial planning of a business. It's a self-assessment tool that uses past and current sales statistics to intelligently predict future performance. If a company predicts robust sales in the fourth quarter but only earns half that amount, it's a sign to stockholders that not only is the company performing poorly, but management is clueless. When attracting new investors to a private company, sales forecasts can be used to predict the potential return on investment. The overall effect of accurate sales forecasting is a business that runs more efficiently, saving money on excess inventory, increasing profit and serving its customers better.

Accurate forecasts that meet the forthcoming consumption demands of customers help retail business owners and management to maximize and extend profits over the long term. Forecasting permits price adjustments to correspond with the current level of consumer spending patterns. Maintaining and controlling a sufficient but moderate inventory that meets the need without being excessive also adds to long-term profits in the retail industry.

4.2.2 Limitation

Although we can use the relatively good model to help predict future phenomenon, no matter how good it is, the model will always have limitations.

1) **Missing Details**: Most models can't incorporate all the details of complex natural phenomena. For example, in the case discussed here, there maybe some other factors besides variables included in the model, like psychology and income of clients. Since models must be simple enough that you can use them to make predictions, they often leave out some of the details.

2) **Many Approximations**: The model we fit here include some approximations as a convenient way to describe something that happens in nature. These approximations are not exact, so predictions based on them tend to be a little bit different from what you actually observe – close, but not bang on. These approximations are good, but they are approximations nonetheless.

3) **Many Assumptions**: When we fit a model, we should make a lot of assumptions. For example, we need to assume the predictors are independent and the residuals are normally distributed and so on. But in reality, those assumptions cannot be completely realized.

4) **Experimental Errors**: Experimental errors include random errors and systematic errors. Random errors can be evaluated through statistical analysis and can be reduced by averaging over a large number of observations. However, in the dataset we discuss here, obviously the number of observations are not large enough, which may affect the accuracy of the prediction. Systematic errors are difficult to detect and cannot be analyzed statistically.

5) **Transparency**: The data used for modeling should be transparent. Otherwise, if the data is fabricated, the model would be not accurate enough to make predictions.

4.2.3 Future Direction

Retail forecasting methods anticipate the future purchasing actions of consumers by evaluating past revenue and consumer behavior over the previous months or year to discern patterns and develop forecasts for the upcoming months. Data is adjusted for seasonal trends, and then a plan for ordering and stocking products may follow the analysis. After fulfillment of current and forthcoming customer purchases and orders, an assessment of the results is compared with previous forecasts, and the entire procedure is repeated.

Acknowledgement

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this report. A special gratitude I give to our final project instructor, [Mr Yajima], whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in modeling selection. Sincere thanks go to my classmates, [Ms Wang, Yu, Rong and Mr Yan], who gave me useful materials.

Reference

Rune Haubo B Christensen. A Tutorial on fitting Cumulative Link Mixed Models with clmm2 from the ordinal Package. August 25, 2018

Andrew Gelman, Jennifer Hill. Data Analysis Using Regression and Multilevel_Hierarchical Models. 2006, Cambridge University Press

https://vuorre.netlify.com/post/2016/2016-03-06-multilevel-predictions/

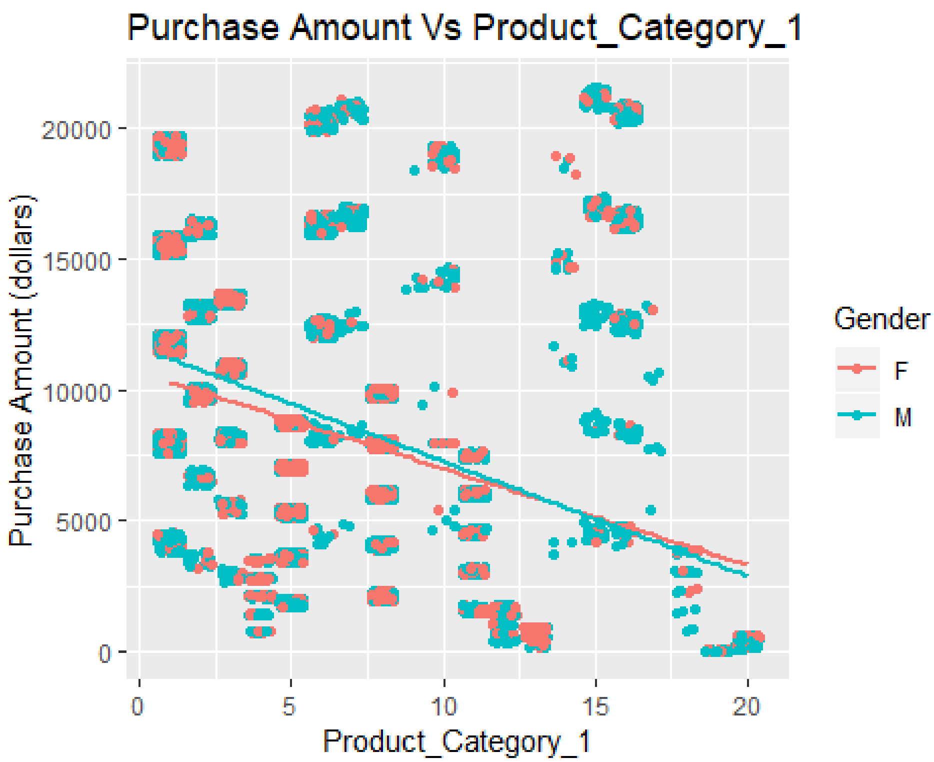
https://cran.r-project.org/web/packages/jtools/vignettes/summ.html

https://web.stanford.edu/class/psych252/section/Mixed_models_tutorial.html

http://r-statistics.co/Ordinal-Logistic-Regression-With-R.html

Appendix

```
ggplot(bf, aes(x=Product_Category_1, y=Purchase)) +
  geom_jitter(aes(col=Gender)) +
  geom_smooth(aes(col=Gender), method="lm", se=F) +
  labs(title="Purchase Amount Vs Product_Category_1",
       x = "Product_Category_1",
       y = "Purchase Amount (dollars)")
```


Purchase Amount Vs Product_Category_1
Purchase Amount (dollars)
20000
15000
10000
5000
0
0
5
10
15
20
Product_Category_1
Gender
F
M