affirm_analytics.R

ashimdatta

Sun Jul 24 05:08:13 2016

```
#### Solutions by Ashim Datta, email- datta.ashim2@gmail.com   #####



#### q1- Anomalies in data
## 1. 629 user information missin
## 2. No merchant information available for 1 merchant in funnel data
## 3. No merchant information available for 1 merchant in loan data
#  4. No loan information for one merchant in funnel data
#  5. Funnel information is missing for 3016 loans

#### q3- Which industries to go for
#  1. Focus on Jewellery- Jewellery has least confirmation rate and hence contributes to
 least revenue. But every thing else remaining same positively affects revenue
#  2. Try to increase satisfaction for repeat customers- Repeat customers negatively aff
ect revenue
#  3. Old is Gold- Older people and high fico scores contribute more to revenue



setwd("/Users/ashimdatta/Documents/Affirm Analytics Test")
library("sqldf")
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared
 object '/Library/Frameworks/R.framework/Resources/modules//R_X11.so':
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 6): Library not
 loaded: /opt/X11/lib/libSM.6.dylib
##   Referenced from: /Library/Frameworks/R.framework/Resources/modules//R_X11.so
##   Reason: image not found
```

```
## Could not load tcltk.  Will use slower R code instead.
```

```
## Loading required package: RSQLite
```

```
## Loading required package: DBI
```

```
library("randomForest")
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library("ggplot2")
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library("gridExtra")
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
library("lubridate")
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library("Hmisc")
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
funnel<-read.csv("funnel.csv")
loans<-read.csv("loans.csv")
merchants<-read.csv("merchants.csv")

summary(funnel)
```

```
##            merchant_id               user_id
##  YKHCNPR33GOHG3M6:192871   0              : 67202
##  H7GADDVO9CIZHMCL: 70152   4080-7608-SFDJ:   305
##  MNLK8D11U6PV4THN: 58218   0888-1688-PGKN:   275
##  ZXTF6RNQXU3VCMHV: 29234   3779-9245-UEGU:   253
##  P2T82BO89LRD4WYH: 26936   1453-7726-WNOY:   223
##  8L2VTJ7XV2QQ4PCU: 19830   2208-8737-VELT:   179
##  (Other)         : 36085   (Other)       :364889
##            checkout_id                     action
##  000IZ73IWYQQBNQH:     4   Checkout Completed : 40671
##  003K6QR0G29SC9DJ:     4   Checkout Loaded    :197950
##  005B5NYUZNC0FTRR:     4   Loan Terms Approved: 63328
##  006R5CXAY6MTVISH:     4   Loan Terms Run     :131377
##  00ALXB15Q2MOUFRK:     4
##  00C7R9W7DANWOE04:     4
##  (Other)         :433302
##        action_date
##  2/5/16 0:00 :  6153
##  2/11/16 0:00:  5992
##  2/12/16 0:00:  5902
##  2/29/16 0:00:  5858
##  3/14/16 0:00:  5853
##  2/10/16 0:00:  5802
##  (Other)     :397766
```

```
x<-is.na(funnel$checkout_id)
## Check if checkoutid was captured for every event
table(x)
```

```
## x
##  FALSE
## 433326
```

```
# All checkoutid were loaded

y<-table(funnel$checkout_id)
max(y)
```

```
## [1] 4
```

```
min(y)
```

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

```
## No checkoutid was present for more than 4 times

num_checkout_loaded<-length(which(funnel$action=='Checkout Loaded'))
num_loan_run<-length(which(funnel$action=='Loan Terms Run'))

num_users_missing<-length(which(funnel$user_id==0))

ideal_useers_missing<-num_checkout_loaded-num_loan_run

missing_data<-num_users_missing-ideal_useers_missing
# Number of user records missing
missing_data
```

```
## [1] 629
```

```
merchant1<-sqldf("select distinct merchant_id from funnel",drv="SQLite")
merchant2<-sqldf("select distinct merchant_id from merchants",drv="SQLite")
merchant3<-sqldf("select distinct merchant_id from loans",drv="SQLite")

merchant1_match_merchant2<-sqldf("select a.merchant_id as merchant_funnel,b.merchant_id
 from merchant1 a
left join
merchant2 b on a.merchant_id=b.merchant_id ",drv='SQLite')

merchant1_match_merchant3<-sqldf("select a.merchant_id as merchant_funnel,b.merchant_id
 from merchant1 a
left join
merchant3 b on a.merchant_id=b.merchant_id ",drv='SQLite')

merchant2_match_merchant3<-sqldf("select a.merchant_id as merchant_funnel,b.merchant_id
 from merchant3 a
left join
                                 merchant2 b on a.merchant_id=b.merchant_id ",drv='SQLit
e')

print(table(is.na(merchant1_match_merchant2$merchant_id)))
```

```
##
## FALSE  TRUE
##    16     1
```

```
## No merchant information available for 1 merchant in funnel data

print(table(is.na(merchant1_match_merchant3$merchant_id)))
```

```
##
## FALSE  TRUE
##    16     1
```

```
## No loan information for one merchant in funnel data

print(table(is.na(merchant2_match_merchant3$merchant_id)))
```

```
##
## FALSE  TRUE
##    15     1
```

```
## No merchant information available for 1 merchant in loan data

# Date for which we have data

# Minimum date for loan checkout from funnel data
min(as.Date(funnel[which(funnel$action=="Checkout Completed"),'action_date'],format = "%
m/%d/%y %H:%M"))
```

```
## [1] "2016-01-01"
```

```
# Maximum date for loan checkout from funnel data
max(as.Date(funnel[which(funnel$action=="Checkout Completed"),'action_date'],format = "%
m/%d/%y %H:%M"))
```

```
## [1] "2016-03-31"
```

```
# Minimum date for loan checkout from loans data
min(as.Date(loans$checkout_date,format = "%m/%d/%y %H:%M"))
```

```
## [1] "2016-01-01"
```

```
# Maximum date for loan checkout from loans data
max(as.Date(loans$checkout_date,format = "%m/%d/%y %H:%M"))
```

```
## [1] "2016-03-31"
```

```
## Number loans checked out from funnel data should ideally match with number loans in l
oans data

funnel_checkouts<-length(unique(funnel[which(funnel$action=="Checkout Completed"),'check
out_id']))
loans_checkouts<-length(unique(loans[,'checkout_id']))

print(loans_checkouts-funnel_checkouts)
```

```
## [1] 3016
```

```
# Funnel information is missing for 3016 loans


str(funnel)
```

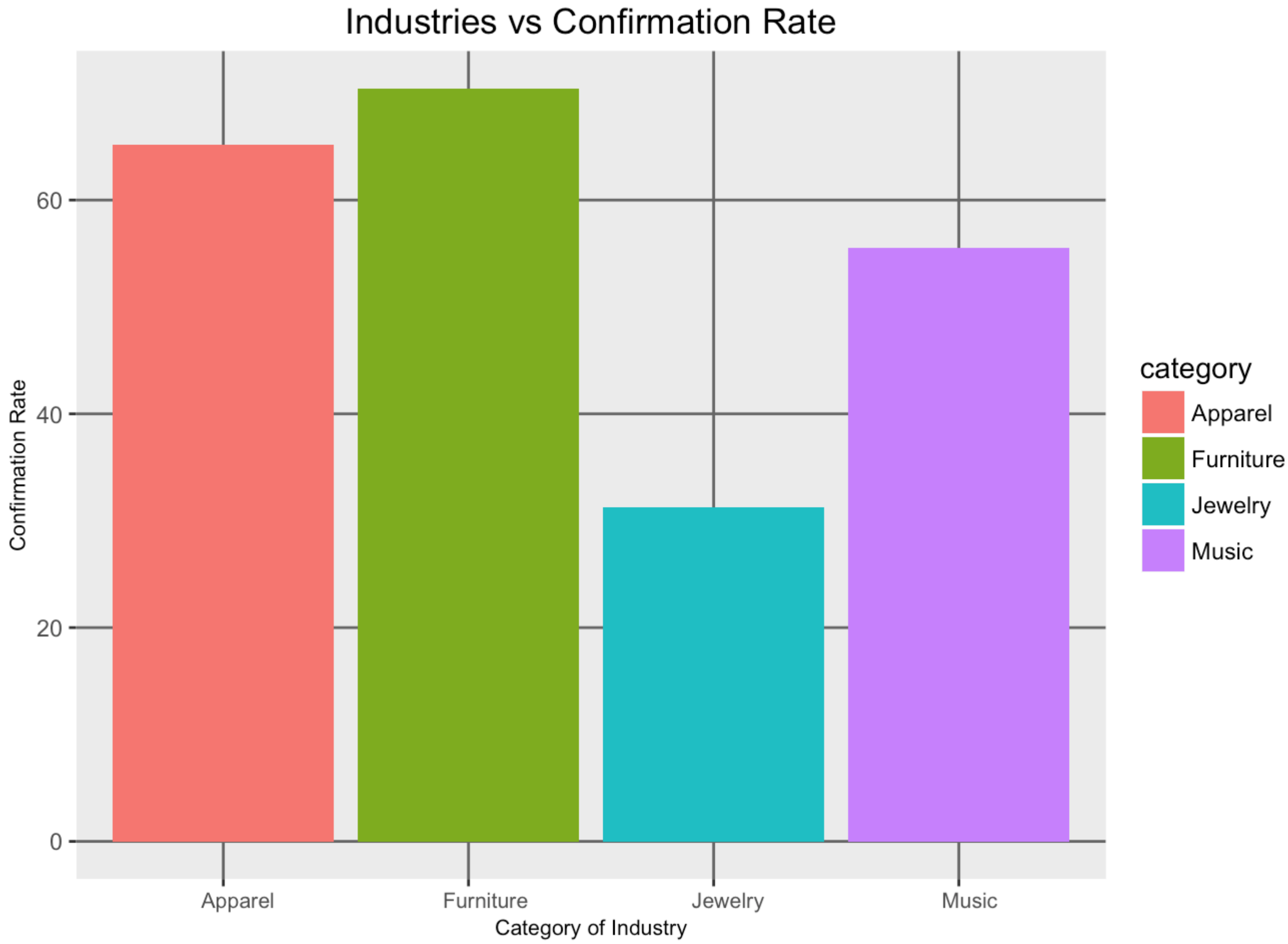
```
## 'data.frame':    433326 obs. of  5 variables:
##  $ merchant_id: Factor w/ 17 levels "2ZOAIY64Q3G5QU6Q",..: 8 2 9 13 9 9 16 16 16 10
 ...
##  $ user_id    : Factor w/ 82800 levels "0","0000-0356-AANL",..: 18879 40972 1 1 1 664
73 38917 29536 41682 1 ...
##  $ checkout_id: Factor w/ 198766 levels "000IZ73IWYQQBNQH",..: 130227 172473 17315 19
6835 75299 155259 64870 52588 21407 175869 ...
##  $ action     : Factor w/ 4 levels "Checkout Completed",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ action_date: Factor w/ 84 levels "1/1/16 0:00",..: 78 22 80 73 84 66 72 77 27 84
 ...
```

```
## Converting action_date to date

funnel$action_date2<-as.Date(funnel$action_date,format = "%m/%d/%y %H:%M")

funnel_agg<-sqldf("select action_date2,
sum(case when action='Checkout Loaded' then 1 else 0 end) as num_loaded,
sum(case when action='Loan Terms Run' then 1 else 0 end) as num_applied,
sum(case when action='Loan Terms Approved' then 1 else 0 end) as num_approved,
sum(case when action='Checkout Completed' then 1 else 0 end) as num_confirmed
from funnel group by 1",drv="SQLite")

funnel_agg$application_rate<-funnel_agg$num_applied/funnel_agg$num_loaded
funnel_agg$approval_rate<-funnel_agg$num_approved/funnel_agg$num_applied
funnel_agg$confirmation_rate<-funnel_agg$num_confirmed/funnel_agg$num_approved


##2. Calculate conversion through the funnel by day such that the data structure is:

head(funnel_agg)
```

```
##   action_date2 num_loaded num_applied num_approved num_confirmed
## 1   2016-01-01       1463        1070          663           397
## 2   2016-01-02       1802        1349          795           485
## 3   2016-01-03       1772        1339          810           488
## 4   2016-01-04       2012        1508          913           554
## 5   2016-01-05       2277        1585          899           577
## 6   2016-01-06       2080        1387          821           546
##   application_rate approval_rate confirmation_rate
## 1        0.7313739     0.6196262         0.5987934
## 2        0.7486127     0.5893254         0.6100629
## 3        0.7556433     0.6049291         0.6024691
## 4        0.7495030     0.6054377         0.6067908
## 5        0.6960913     0.5671924         0.6418242
## 6        0.6668269     0.5919250         0.6650426
```

```
###3.Which merchant industry and/or user demographic would you focus business developmen
t on based on current checkout funnel and loan performance? (Assume we have roughly the
 same market penetration in each so that saturation isn't a concern and assume revenue t
o Affirm = (mdr + loan_return_percentage) * loan_amount). Please put together a 3-page P
owerPoint presentation to the executive team with your recommendation (title and agenda
 slides don’t count in the total).


## To understand what affects funnel, we would not be able to look at user demographics
 as user demographics data exists only for loans confirmed
## We can look at the affect of industry on funnel

funnel_merchant<-sqldf("select a.*,b.merchant_name,b.category from
                      funnel a left join merchants b
                       on lower(a.merchant_id)=lower(b.merchant_id)", drv="SQLite")

funnel_merchant_agg<-sqldf("select category,
                  sum(case when action='Loan Terms Approved' then 1 else 0 end) as num_a
pproved,
                  sum(case when action='Checkout Completed' then 1 else 0 end) as num_co
nfirmed
                  from funnel_merchant group by 1",drv="SQLite")

funnel_merchant_agg$confirmation_rate<-funnel_merchant_agg$num_confirmed/funnel_merchant
_agg$num_approved

ggplot(funnel_merchant_agg, aes(x=category,y=round((confirmation_rate)*100,2),fill=categ
ory)) +
  geom_bar(stat = "identity") +
  theme(panel.grid.major = element_line(colour = "grey40"),
      panel.grid.minor = element_blank())+
  theme(axis.text.x = element_text(hjust = .3, size = 8),
        axis.title=element_text(size=8))+
  xlab("Category of Industry") + ylab("Confirmation Rate") +
  ggtitle("Industries vs Confirmation Rate")
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```


Industries vs Confirmation Rate
Confirmation Rate
0
20
40
60
Apparel
Furniture
Jewelry
Music
Category of Industry
category
Apparel
Furniture
Jewelry
Music

```
### Apparel and Furniture have the highest confirmation rate

## Let us try to see if industry can be a predictor for confirmation

funnel_merchant$confirmned<-ifelse(funnel_merchant$action=='Checkout Completed',1,0)
funnel_merchantapproved<-funnel_merchant[which(funnel_merchant$action=='Loan Terms Appro
ved'),]
funnel_merchantconfirm<-funnel_merchant[which(funnel_merchant$action=='Checkout Complete
d'),]

##confirmed loans for the ones which are approved
funnel_merchantapproved_conf<-sqldf("select a.category, b.confirmned
                                    from funnel_merchantapproved a
                                    left join
                                    funnel_merchantconfirm b
                                    on
                                    a.checkout_id=b.checkout_id", drv="SQLite")
funnel_merchantapproved_conf[is.na(funnel_merchantapproved_conf)]<-0

split <- sample(seq_len(nrow(funnel_merchantapproved_conf)), size = floor(0.75 * nrow(fu
nnel_merchantapproved_conf)))
trainData <- funnel_merchantapproved_conf[split, ]
testData <- funnel_merchantapproved_conf[-split, ]

## Using logistic regression as the dependent variable is categorical

model <- glm(confirmned ~.,family=binomial(link='logit'),data=trainData)
summary(model)
```

```
##
## Call:
## glm(formula = confirmned ~ ., family = binomial(link = "logit"),
##     data = trainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5509  -1.4575   0.8452   0.9211   1.5240
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.18232    0.30277   0.602  0.54705
## categoryApparel    0.45565    0.30302   1.504  0.13266
## categoryFurniture  0.66318    0.30345   2.185  0.02886 *
## categoryJewelry   -0.96806    0.30869  -3.136  0.00171 **
## categoryMusic      0.04337    0.30381   0.143  0.88649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 61921  on 47495  degrees of freedom
## Residual deviance: 60962  on 47491  degrees of freedom
## AIC: 60972
##
## Number of Fisher Scoring iterations: 4
```

```
## Amongst the industries, jewellery is possibly suffering the most and furniture has be
tter confirmation rates


anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: confirmned
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     47495      61921
## category  4    959.5     47491      60962 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fitted.results <- predict(model,newdata=testData,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)

misClasificError <- mean(fitted.results != testData$confirmned, na.rm = TRUE)
print(paste('Accuracy',1-misClasificError))
```

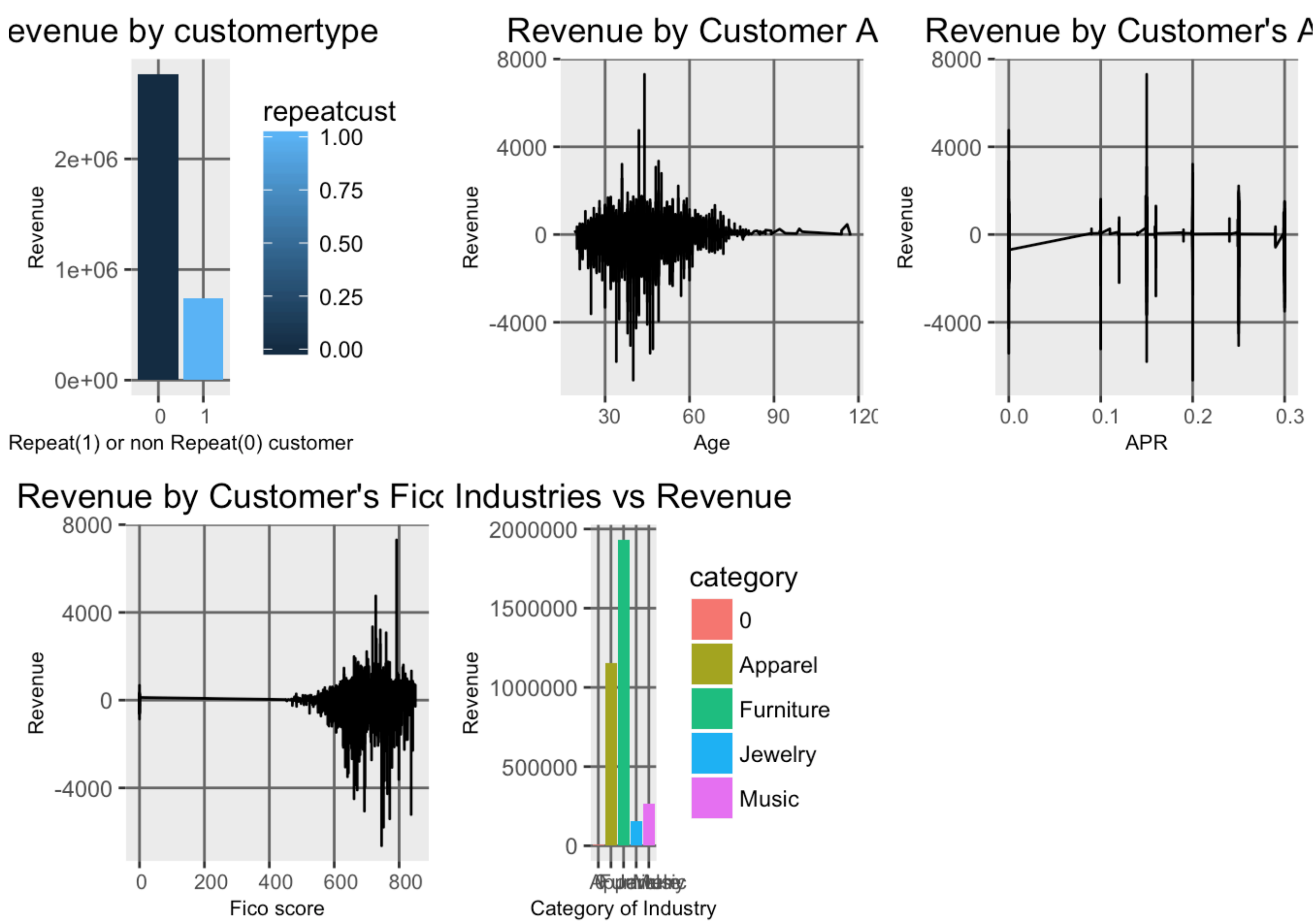
```
## [1] "Accuracy 0.651086407276402"
```

```
## This is possibly a good model as it can predict confirmation with ~65% accuracy

## Now, we will look into the loan performance for industry and users who have completed
 checkouts

loans_merchant<-sqldf("select a.*,b.merchant_name,b.category from
                      loans a left join merchants b
                      on lower(a.merchant_id)=lower(b.merchant_id)", drv="SQLite")
str(loans_merchant)
```

```
## 'data.frame':    43687 obs. of  15 variables:
##  $ merchant_id           : chr  "ZXTF6RNQXU3VCMHV" "YKHCNPR33GOHG3M6" "MNLK8D11U6PV4T
HN" "MNLK8D11U6PV4THN" ...
##  $ user_id               : Factor w/ 31587 levels "0000-0356-AANL",..: 20086 10079 83
63 553 19031 2897 8922 23964 14443 15068 ...
##  $ checkout_id           : Factor w/ 43687 levels "000IZ73IWYQQBNQH",..: 4012 42375 7
810 30660 40605 25309 24736 9276 41751 41671 ...
##  $ checkout_date         : Factor w/ 91 levels "1/1/16 0:00",..: 57 32 85 81 27 47 31
 63 67 22 ...
##  $ loan_amount           : num  1060 2300 850 950 859 ...
##  $ down_payment_amount   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ users_first_capture   : Factor w/ 586 levels "0","1/1/15","1/1/16",..: 73 1 1 1 1
 392 1 1 1 1 ...
##  $ user_dob_year         : int  1972 1981 1983 1981 1951 1977 1956 1952 1949 1971 ...
##  $ loan_length_months    : int  12 12 6 6 6 6 6 6 6 3 ...
##  $ mdr                   : num  0.025 0.019 0.059 0.059 0.059 0.019 0.029 0.019 0.059
 0.019 ...
##  $ apr                   : num  0.25 0.3 0 0 0 0.25 0.3 0.3 0 0.3 ...
##  $ fico_score            : int  685 628 808 612 783 700 644 655 757 646 ...
##  $ loan_return_percentage: num  0.0055 0.0353 0.0584 0.0759 0.1019 ...
##  $ merchant_name         : chr  "Goat, LLC" "Cheddar Inc." "Pepperjack Co." "Pepperja
ck Co." ...
##  $ category              : chr  "Apparel" "Apparel" "Furniture" "Furniture" ...
```

```
loans_merchant$revenue_affirm<-
(loans_merchant$mdr+loans_merchant$loan_return_percentage)*loans_merchant$loan_amount

loans_merchant$repeatcust<-ifelse(loans_merchant$users_first_capture==0,0,1)

loans_merchant$user_age<-year(Sys.Date()) - loans_merchant$user_dob_year

## Variables that can affect Affirm revenue are- repeatcust,user_age,apr,fico_score, mer
chant_name and cateogry

loans_merchant2<-
loans_merchant[,c('repeatcust','user_age','apr','fico_score','category','revenue_affirm')]


loans_merchant2[is.na(loans_merchant2)]<-0

loans_merchant2_agg<-sqldf("select repeatcust,sum(revenue_affirm) as revenue_affirm
                           from loans_merchant2 group by 1 ",
                           drv="SQLite")

x<-ggplot(loans_merchant2_agg, aes(x=as.factor(repeatcust),y=round((revenue_affirm),2),f
ill=repeatcust)) +
  geom_bar(stat = "identity") +
  theme(panel.grid.major = element_line(colour = "grey40"),
        panel.grid.minor = element_blank())+
  theme(axis.text.x = element_text(hjust = .3, size = 8),
        axis.title=element_text(size=8))+
  xlab("Repeat(1) or non Repeat(0) customer") + ylab("Revenue") +
  ggtitle("Revenue by customertype")


y<-ggplot(loans_merchant2, aes(x=user_age,y=round((revenue_affirm),2))) +
  geom_line() +
  theme(panel.grid.major = element_line(colour = "grey40"),
        panel.grid.minor = element_blank())+
  theme(axis.text.x = element_text(hjust = .3, size = 8),
        axis.title=element_text(size=8))+
  xlab("Age") + ylab("Revenue") +
  ggtitle("Revenue by Customer Age")

z<-ggplot(loans_merchant2, aes(x=apr,y=round((revenue_affirm),2))) +
  geom_line() +
  theme(panel.grid.major = element_line(colour = "grey40"),
        panel.grid.minor = element_blank())+
  theme(axis.text.x = element_text(hjust = .3, size = 8),
        axis.title=element_text(size=8))+
  xlab("APR") + ylab("Revenue") +
  ggtitle("Revenue by Customer's APR")


t<-ggplot(loans_merchant2, aes(x=fico_score,y=round((revenue_affirm),2))) +
  geom_line() +
  theme(panel.grid.major = element_line(colour = "grey40"),
```

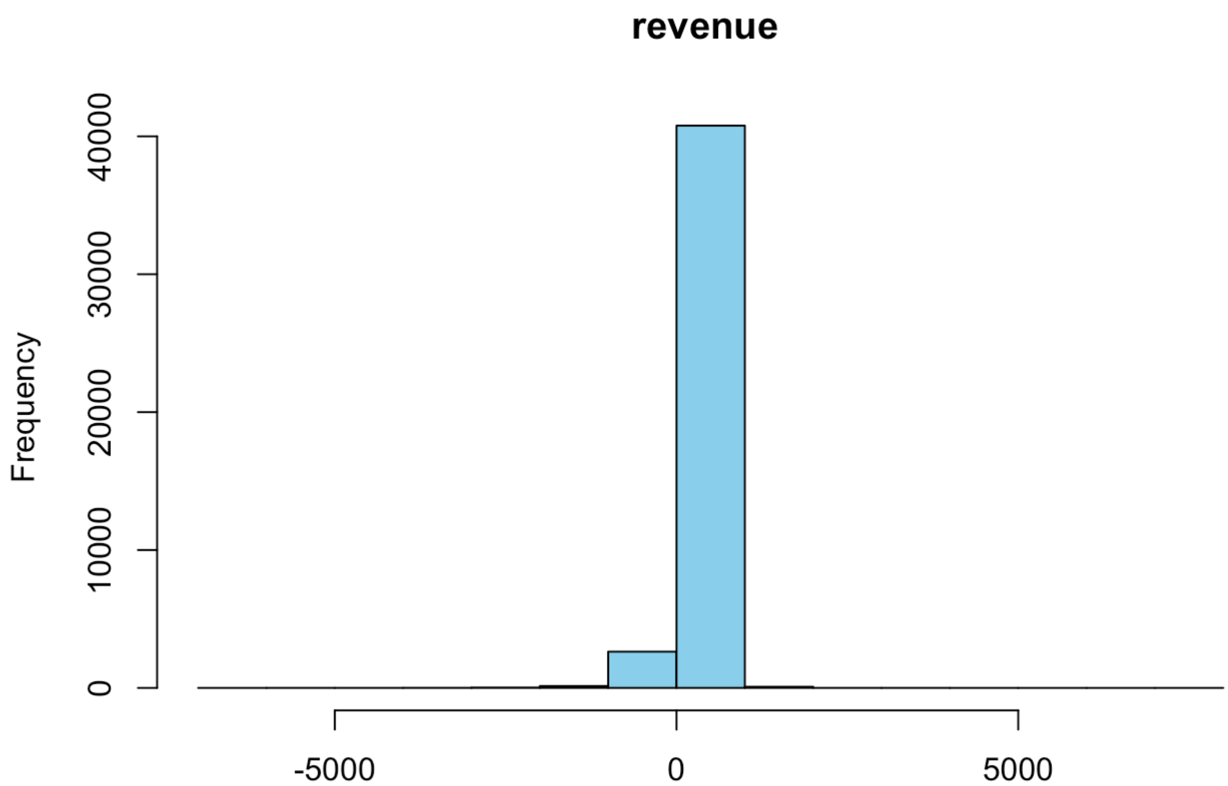
```
        panel.grid.minor = element_blank())+
  theme(axis.text.x = element_text(hjust = .3, size = 8),
        axis.title=element_text(size=8))+
  xlab("Fico score") + ylab("Revenue") +
  ggtitle("Revenue by Customer's Ficoscore")

loans_merchant2_agg2<-sqldf("select category,sum(revenue_affirm) as revenue_affirm
                           from loans_merchant2 group by 1 ",
                           drv="SQLite")

p<-ggplot(loans_merchant2_agg2,
aes(x=category,y=round((revenue_affirm),2),fill=category)) +
  geom_bar(stat = "identity") +
  theme(panel.grid.major = element_line(colour = "grey40"),
        panel.grid.minor = element_blank())+
  theme(axis.text.x = element_text(hjust = .3, size = 8),
        axis.title=element_text(size=8))+
  xlab("Category of Industry") + ylab("Revenue") +
  ggtitle("Industries vs Revenue")


grid.arrange(x, y, z, t, p, ncol=3, nrow =2)
```



```
### Points to Note
## 1.Repeat customers do not contribute to a lot of revenue
## Furniture category contributes to most revenue and Jewellery contributes to least



## Let us try to apply regression to the dataset and check if we can predict affirm_reve
nuebased on the above variables

head(loans_merchant)
```

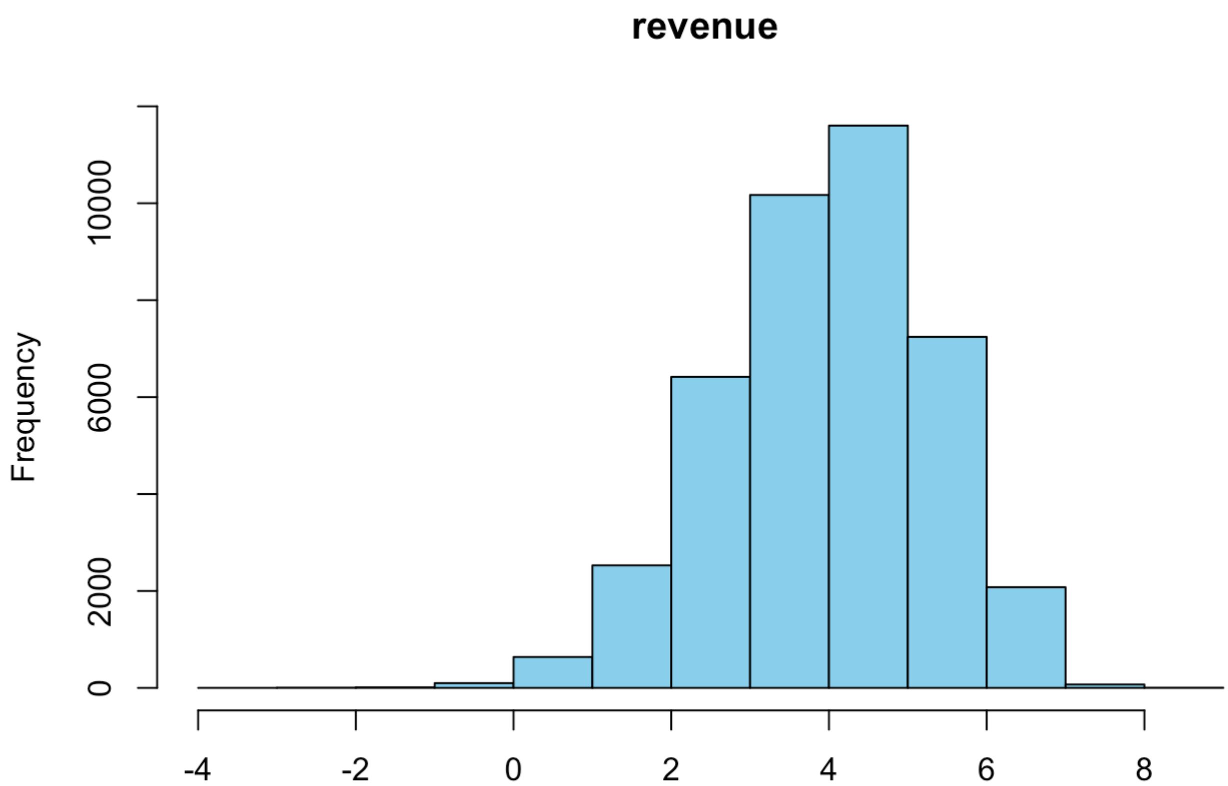
```
##        merchant_id        user_id      checkout_id checkout_date
## 1 ZXTF6RNQXU3VCMHV 6387-9021-JSOJ 3AOXIJUSJKQOE0UB   2/6/16 0:00
## 2 YKHCNPR33GOHG3M6 3200-9015-GCZG YWMTY1ZYAXB4G0LV   2/1/16 0:00
## 3 MNLK8D11U6PV4THN 2656-3540-OPOR 6ANRFMR2W3YTVBP6  3/31/16 0:00
## 4 MNLK8D11U6PV4THN 0175-9602-ERQN P5C004D6YERT9S8G  3/28/16 0:00
## 5 MNLK8D11U6PV4THN 6053-1602-FBDR XEIT9H4G4BNF8MNB   1/5/16 0:00
## 6 YKHCNPR33GOHG3M6 0907-5749-MLEQ KLA75J8V0LFZVNN3  2/23/16 0:00
##   loan_amount down_payment_amount users_first_capture user_dob_year
## 1        1060                   0            10/13/15          1972
## 2        2300                   0                   0          1981
## 3         850                   0                   0          1983
## 4         950                   0                   0          1981
## 5         859                   0                   0          1951
## 6         379                   0              4/6/15          1977
##   loan_length_months   mdr  apr fico_score loan_return_percentage
## 1                 12 0.025 0.25        685                 0.0055
## 2                 12 0.019 0.30        628                 0.0353
## 3                  6 0.059 0.00        808                 0.0584
## 4                  6 0.059 0.00        612                 0.0759
## 5                  6 0.059 0.00        783                 0.1019
## 6                  6 0.019 0.25        700                 0.1242
##    merchant_name  category revenue_affirm repeatcust user_age
## 1      Goat, LLC   Apparel        32.3300          1       44
## 2   Cheddar Inc.   Apparel       124.8900          0       35
## 3 Pepperjack Co. Furniture        99.7900          0       33
## 4 Pepperjack Co. Furniture       128.1550          0       35
## 5 Pepperjack Co. Furniture       138.2131          0       65
## 6   Cheddar Inc.   Apparel        54.2728          1       39
```

```
hist(loans_merchant2$revenue_affirm,xlab=" ",main="revenue ", col="skyblue")
```




```
hist(log(loans_merchant2$revenue_affirm),xlab=" ",main="revenue ", col="skyblue")
```

```
## Warning in log(loans_merchant2$revenue_affirm): NaNs produced
```

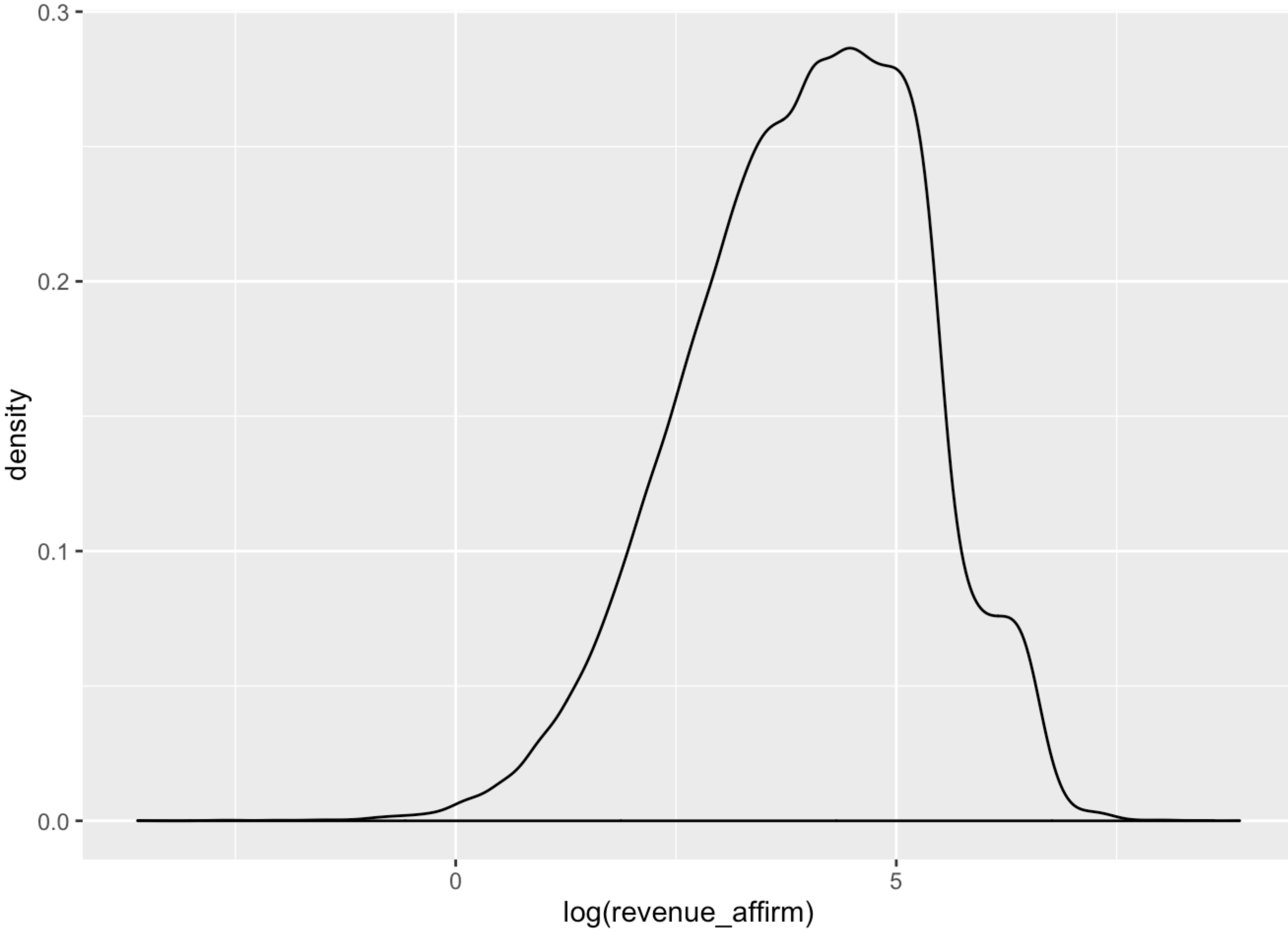



```
ggplot(loans_merchant2, aes(x=log(revenue_affirm))) + geom_density(alpha=.3)
```

```
## Warning in log(revenue_affirm): NaNs produced
```

```
## Warning in log(revenue_affirm): NaNs produced
```

```
## Warning: Removed 2814 rows containing non-finite values (stat_density).
```




```
## Log of dependent variable can be used to regress and hence we will apply log linear r
egression

set.seed(123)
split <- sample(seq_len(nrow(loans_merchant2)), size = floor(0.75 *
nrow(loans_merchant2)))
trainData <- loans_merchant2[split, ]
testData <- loans_merchant2[-split, ]

## base model

best.guess <- mean(trainData$revenue_affirm)

# Evaluate RMSE and MAE on the testing data
RMSE.baseline <- sqrt(mean((best.guess-testData$revenue_affirm)^2,na.rm=TRUE))
RMSE.baseline
```

```
## [1] 229.8227
```

```
MAE.baseline <- mean(abs(best.guess-testData$revenue_affirm),na.rm=TRUE)
MAE.baseline
```

```
## [1] 112.5977
```

```
predictionModel <- lm(log(revenue_affirm+1) ~ repeatcust + user_age + apr + fico_score
                        +category, data = trainData)
```

```
## Warning in log(revenue_affirm + 1): NaNs produced
```

```
summary(predictionModel)
```

```
##
## Call:
## lm(formula = log(revenue_affirm + 1) ~ repeatcust + user_age +
##     apr + fico_score + category, data = trainData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7585 -0.6475  0.0015  0.6685  4.8424
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.809e+00  2.350e-01  16.211  < 2e-16 ***
## repeatcust        -5.424e-01  1.346e-02 -40.293  < 2e-16 ***
## user_age           1.866e-03  5.010e-04   3.723 0.000197 ***
## apr                4.860e-02  1.067e-01   0.455 0.648890
## fico_score         1.396e-03  6.561e-05  21.284  < 2e-16 ***
## categoryApparel   -9.507e-01  2.271e-01  -4.186 2.85e-05 ***
## categoryFurniture  2.994e-01  2.285e-01   1.310 0.190128
## categoryJewelry    8.204e-01  2.324e-01   3.531 0.000415 ***
## categoryMusic     -6.815e-01  2.275e-01  -2.995 0.002744 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.014 on 30705 degrees of freedom
##   (2051 observations deleted due to missingness)
## Multiple R-squared:   0.36,  Adjusted R-squared:  0.3598
## F-statistic:  2159 on 8 and 30705 DF,  p-value: < 2.2e-16
```

```
test.pred.lin <- exp(predict(predictionModel,testData))-1

RMSE.lin.reg <- sqrt(mean((test.pred.lin-testData$revenue_affirm)^2,na.rm = TRUE))
RMSE.lin.reg
```

```
## [1] 220.8348
```

```
MAE.lin.reg <- mean(abs(test.pred.lin-testData$revenue_affirm),na.rm = TRUE)
MAE.lin.reg
```

```
## [1] 91.69327
```

```
## Root mean squared error and Mean absolute errors are reduced by this predictive model
## Points to note, every thing else remaining same
##     Repeat customers are not good for revenue
##     Older people contribute to more revenue
##     Better fico score users contribute to more revenue
##     Apparel and Jewellery is good for revenue
##     Music is bad for revenue
```