econ725 project Q3

Demi Han, Ruyuan Mei

11/19/2020

######Sugandh######

```r
require(tidyverse)
require(ggplot2)
require(data.table)
require(plyr)
require(dplyr)
require(knitr)
require(foreign)
require(ggcorrplot)
require(corrplot)
require(caret)
require(gridExtra)
require(scales)
require(Rmisc)
require(ggrepel)
require(randomForest)
require(glmnet)
require(psych)
require(xgboost)
require(ggthemes)
```

```r
#loading the dataset
df <- read.dta("~/Desktop/ebaydatafinal.dta")
```

```r
#summary for the highest bid
summary(df$biddy1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       1    3576    8100   11840   15950 1780400   22522
```

There are 22522 null values in highest bid variable. This column will be revenue as its the amount the seller gets when he sells the item. one thing to notice here is that the maximum bid in dataset is 1780400.

```r
#keeping only items which have been sold
df <- df[df$sell == 1 ,]
```

I did this considering that if the item isn't sold , then there is no revenue for the seller.

Checking if we have null values now for highest bid

```r
summary(df$biddy1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    2025    5000    8238   10301 1780400
```

Data Cleaning

Formatting Dates:

The columns start date and end date

```
head(df$startdate)
```

```
## [1] "Jul-09-06 20:30:00 PDT" "Mar-07-06 18:21:55 PST" "May-07-06 14:09:21 PDT"
## [4] "Apr-17-06 18:26:37 PDT" "Jun-05-06 20:29:07 PDT" "May-09-06 17:30:00 PDT"
```

```
library("lubridate")
#converting strings into date format
df$startdate <- parse_date_time(df$startdate, orders="mdy HMS")
df$enddate <- parse_date_time(df$enddate, orders="mdy HMS")
```

```
#extracting months from dates
df$months <-  month(df$startdate)
df$days <- day(df$startdate)
df$monthe <-  month(df$enddate)
df$daye <- day(df$enddate)
```

```
#converting long dates to short dates and converting them to mm-dd-yy format
df$startdate <- date(df$startdate)
df$startdate <- format(df$startdate, "%m-%d-%y")
df$enddate <- date(df$enddate)
df$enddate <- format(df$enddate, "%m-%d-%y")
```

The most importent numeric variables

```
numericVars <- which(sapply(df, is.numeric)) #index vector numeric variables
numericVarNames <- names(numericVars) #saving names vector for use later on
cat('There are', length(numericVars), 'numeric variables')
```

```
## There are 510 numeric variables
```

```
df_numVar <- df[, numericVars]
#correlation of all numeric variables
cor_numVar <- cor(df_numVar, use="pairwise.complete.obs")
```

```
## Warning in cor(df_numVar, use = "pairwise.complete.obs"): the standard deviation
## is zero
```

```
#sort on decreasing correlations with highest bid
cor_sorted <- as.matrix(sort(cor_numVar[,'biddy1'], decreasing = TRUE))
```

Lets see which variables are positively correlated with highest bid

```
head(cor_sorted ,50)
```

```
##                         [,1]
## biddy1            1.00000000
## biddy2            0.98872751
## biddy22           0.66472324
## biddy3            0.65837862
## biddy4            0.63900127
## biddy5            0.61199261
## biddy21           0.59125905
## biddy6            0.57843989
```

```
## logbid1         0.57486302
## logbid2         0.54747315
## biddy7          0.54388224
## bookvalue       0.53562316
## logbid3         0.51156412
## biddy8          0.49994164
## logbook         0.46407885
## biddy9          0.46148836
## startbid        0.43734020
## biddy10         0.41987426
## biddy14         0.41781182
## biddy11         0.37896913
## biddy15         0.35227185
## biddy12         0.30442907
## biddy16         0.30106197
## warranty        0.29079211
## biddy13         0.26025689
## biddy17         0.24397853
## biddy18         0.20626768
## logstart        0.19740867
## biddy19         0.16290887
## options         0.12065889
## bidhour22       0.10960263
## biddate20       0.10730503
## biddy20         0.10445221
## phone           0.10109433
## logphotos       0.09621134
## logsize         0.09551801
## loghtml         0.09135437
## logtext         0.09113425
## numbids         0.08340665
## featured        0.08252577
## biddate19       0.07957167
## bidminute20     0.07502050
## descriptionsize 0.07455099
## text            0.07288074
## dealer          0.07265675
## html            0.07158798
## length          0.07073885
## inspection      0.07070693
## photos          0.06610737
## logage          0.06399863
```

From above we can see that biddy2 , bookvalue , startbid , warranty , options , phone, logsize , loghtml , logtext , numbids , featured , descriptionsize , dealer , length , inspection , photos , logage are highly correlated with highest bid .

Now , lets see which variables are negatively correlated with highest bid

```r
tail(cor_sorted ,50)
```

```
##                        [,1]
## dent_little  -0.01850790
## rust_nothing -0.01895390
## broken_no    -0.01913426
## ding_minor   -0.01940057
```

```
## scratch_some -0.01944354
## rust_photo   -0.01981950
## rust_couple  -0.01991869
## ding_small   -0.02054227
## problem_no   -0.02152195
## crack_no     -0.02172253
## ding         -0.02303903
## bidsecond13  -0.02330442
## rust_very    -0.02355268
## rust_one     -0.02396824
## dent_pic     -0.02501008
## rust_major   -0.02549010
## rust_only    -0.02557982
## dent_few     -0.02767722
## dent_minor   -0.02790008
## dent_small   -0.02918563
## bidhour12    -0.03095300
## ding_some    -0.03267418
## bidminute16  -0.03496511
## questions    -0.03591268
## rust_minor   -0.03619099
## rust_few     -0.03732942
## ding_few     -0.03743247
## bidminute22  -0.03764510
## problem      -0.03865022
## bidminute17  -0.03866674
## rust_small   -0.03871339
## dent_some    -0.04153336
## biddate22    -0.04164507
## rust_no      -0.04411493
## bidminute19  -0.04529091
## rust_pic     -0.04590219
## bidsecond21  -0.04620075
## sellerborn   -0.04742733
## broken       -0.04756502
## bidhour17    -0.04853052
## rust_little  -0.04907599
## age          -0.05409009
## bidsecond20  -0.06227907
## bidsecond19  -0.06336647
## crack        -0.06451006
## dent         -0.06838640
## rust_some    -0.07374476
## rust         -0.11572045
## miles        -0.21022800
## logmiles     -0.32073642
```

From above we can see that logmiles , rust , dent , crack , age , broken , problem are negatively correlated with highest bid .

Missing data , label encoding and Factorizing variables

```r
#which columns have missing values
NAcol <- which(colSums(is.na(df)) > 0)
```

```
NAcol
```

```
##         bookvalue highbidderfdback    sellfdbackpct           photos
##                19               25               28               29
##              year        pctfdback           biddy2           biddy3
##                30               33               39               41
##            biddy4           biddy5           biddy6           biddy7
##                43               45               47               49
##            biddy8           biddy9          biddy10          biddy11
##                51               53               55               57
##           biddy12          biddy13          biddy14          biddy15
##                59               61               63               65
##           biddy16          biddy17          biddy18          biddy19
##                67               69               71               73
##           biddy20          biddy21          biddy22         biddate1
##                75               77               79              416
##          bidhour1       bidminute1       bidsecond1         biddate2
##               417              418              419              420
##          bidhour2       bidminute2       bidsecond2         biddate3
##               421              422              423              424
##          bidhour3       bidminute3       bidsecond3         biddate4
##               425              426              427              428
##          bidhour4       bidminute4       bidsecond4         biddate5
##               429              430              431              432
##          bidhour5       bidminute5       bidsecond5         biddate6
##               433              434              435              436
##          bidhour6       bidminute6       bidsecond6         biddate7
##               437              438              439              440
##          bidhour7       bidminute7       bidsecond7         biddate8
##               441              442              443              444
##          bidhour8       bidminute8       bidsecond8         biddate9
##               445              446              447              448
##          bidhour9       bidminute9       bidsecond9        biddate10
##               449              450              451              452
##         bidhour10      bidminute10      bidsecond10        biddate11
##               453              454              455              456
##         bidhour11      bidminute11      bidsecond11        biddate12
##               457              458              459              460
##         bidhour12      bidminute12      bidsecond12        biddate13
##               461              462              463              464
##         bidhour13      bidminute13      bidsecond13        biddate14
##               465              466              467              468
##         bidhour14      bidminute14      bidsecond14        biddate15
##               469              470              471              472
##         bidhour15      bidminute15      bidsecond15        biddate16
##               473              474              475              476
##         bidhour16      bidminute16      bidsecond16        biddate17
##               477              478              479              480
##         bidhour17      bidminute17      bidsecond17        biddate18
##               481              482              483              484
##         bidhour18      bidminute18      bidsecond18        biddate19
##               485              486              487              488
##         bidhour19      bidminute19      bidsecond19        biddate20
##               489              490              491              492
```

```
##        bidhour20      bidminute20      bidsecond20        biddate21
##              493              494              495              496
##        bidhour21      bidminute21      bidsecond21        biddate22
##              497              498              499              500
##        bidhour22      bidminute22      bidsecond22              age
##              501              502              503              519
##             age2         logmiles        logfdback        logphotos
##              520              521              525              526
##           logage          logbook          logbid2          logbid3
##              531              533              538              539
##        compindex             temp
##              541              545
```

```
cat('There are', length(NAcol), 'columns with missing values')
```

```
## There are 126 columns with missing values
```

bookvalue has 19 missing values and photos has 29 missing values and biddy5 has 45 missing values , for now I am just dropping these missing values and we will think about imputingf them in future .

```
#deleting missing values
df=df[!is.na(df$bookvalue),]
df=df[!is.na(df$photos),]
df=df[!is.na(df$biddy5),]
```

Now lets try imputing age and logmiles variables. I am imputing these variables with the median

```
library(Hmisc)
df$age<-impute(df$age, median)
df$logmiles<-impute(df$logmiles, median)
```

Label Encoding / factorizing the charachter variables

```
Charcol <- names(df[,sapply(df, is.character)])
Charcol
```

```
##  [1] "membersince"    "maker"          "interior"       "name"
##  [5] "vin"            "highbiddername" "sellername"     "enddate"
##  [9] "startdate"      "exterior"       "location"       "biddername1"
## [13] "biddername2"    "biddername3"    "biddername4"    "biddername5"
## [17] "biddername6"    "biddername7"    "biddername8"    "biddername9"
## [21] "biddername10"   "biddername11"   "biddername12"   "biddername13"
## [25] "biddername14"   "biddername15"   "biddername16"   "biddername17"
## [29] "biddername18"   "biddername19"   "biddername20"   "biddername21"
## [33] "biddername22"   "software"       "caradphotos"
```

```
cat('There are', length(Charcol), 'remaining columns with character values')
```

```
## There are 35 remaining columns with character values
```

First lets consider variables maker , interior and exterior . They all are factor variables .

```
df$maker <- as.factor(df$maker)
table(df$maker)
```

```
##
## Chevrolet      Ford     Honda    Nissan    Toyota
##      2593      5121      3564       368      1728
```

```r
df$interior <- as.factor(df$interior)
table(df$interior)
```

```
##
##       --    Black     Blue    Brown Burgundy     Gold     Gray    Green
##      104     2204      560      183      205       29     6020       39
##    Other      Red      Tan     Teal    White
##      256      373     3240       12      149
```

```r
df$exterior <- as.factor(df$exterior)
table(df$exterior)
```

```
##
##    Black     Blue    Brown Burgundy     Gold     Gray    Green   Orange
##     2163     1293      107      596      439      677     1316       69
##    Other   Purple      Red   Silver      Tan     Teal    White   Yellow
##      302      101     1885     1208      398      218     2416      186
```

dealing with date variables

```r
df$membersince <- parse_date_time(df$membersince, orders="mdy")
df$monthm <-month(df$membersince)
df$daym <- day(df$membersince)
df$membersince <- date(df$membersince)
df$membersince <- format(df$membersince, "%m-%d-%y")
```

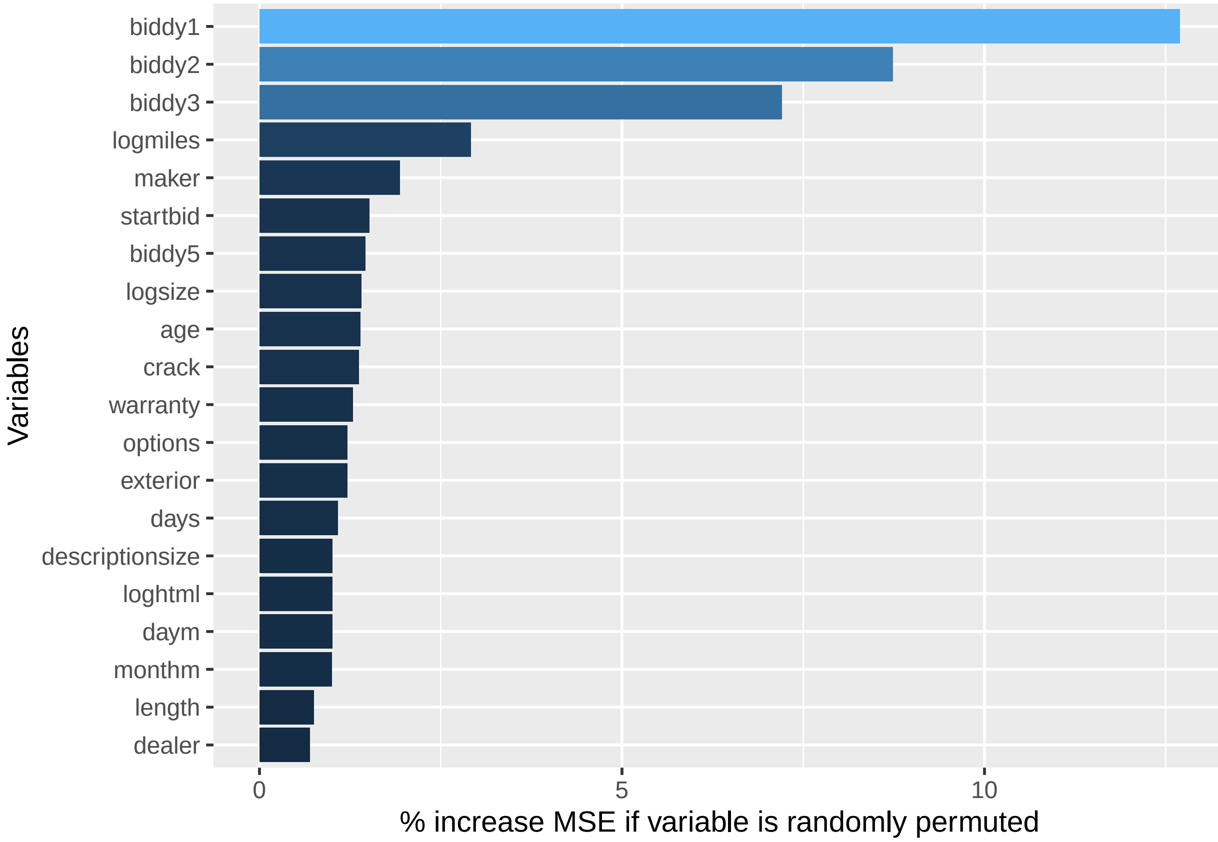
```r
df$months <- as.factor(df$months)
df$days <- as.factor(df$days)
df$monthe <- as.factor(df$monthe)
df$daye <- as.factor(df$daye)
df$monthm <- as.factor(df$monthm)
df$daym <- as.factor(df$daym)
```

Correlations

```r
#keeping only required columns
df<-df[, c("biddy1" , "biddy2" , "biddy3" ,"biddy4", "biddy5" ,"bookvalue", "photos",  "startbid" , "wa
```

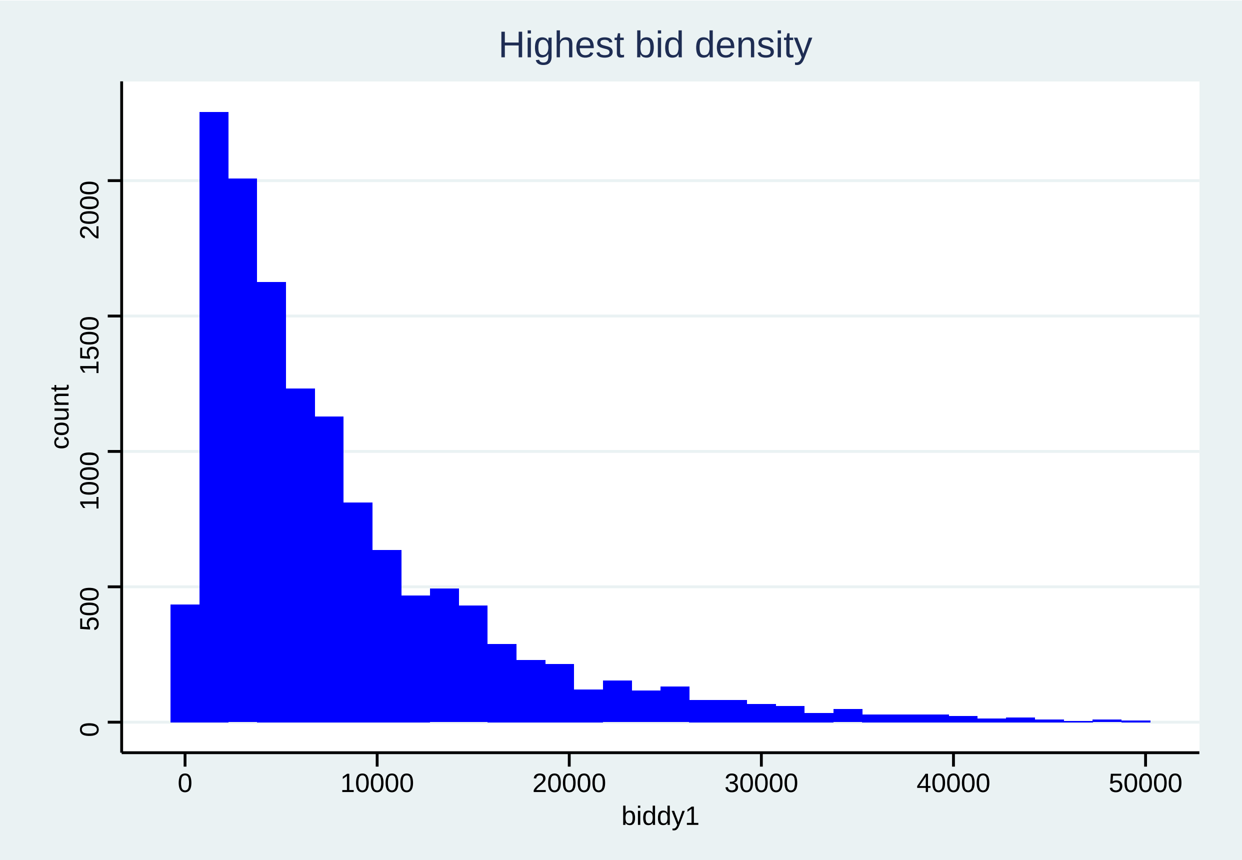
Finding variable importance with Random forest

```r
set.seed(2020)
quick_RF <- randomForest(x=df[1:13374,-36], y= df$biddy1[1:13374], ntree=100,importance=TRUE)
imp_RF <- importance(quick_RF)
imp_DF <- data.frame(Variables = row.names(imp_RF), MSE = imp_RF[,1])
imp_DF <- imp_DF[order(imp_DF$MSE, decreasing = TRUE),]
ggplot(imp_DF[1:20,], aes(x=reorder(Variables, MSE), y=MSE, fill=MSE)) + geom_bar(stat = 'identity') + l
```



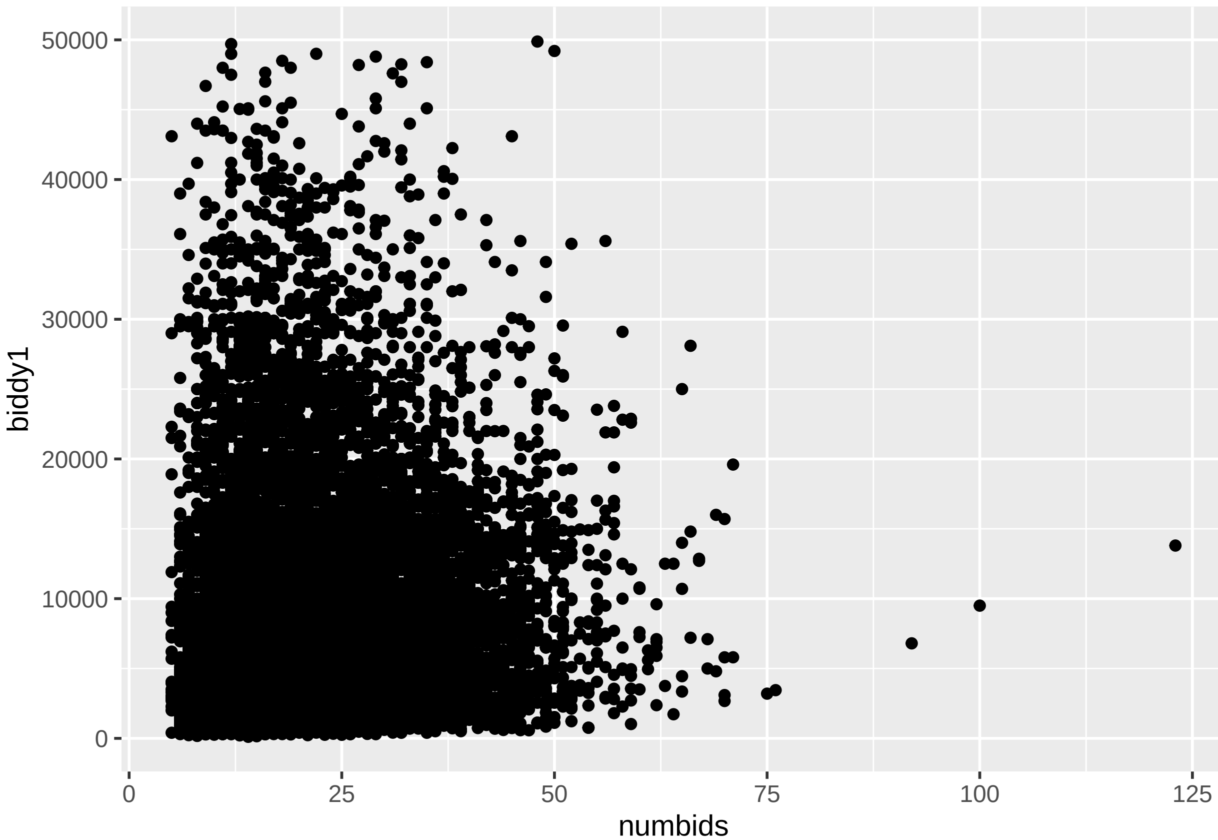
Lets draw some graphs associated with the highest bid/ revenue. first lets see the density of the biddy1

```
p2 <-ggplot(data=df[df$biddy1 < 50000,], aes(x= biddy1))+
  geom_histogram(fill="blue", binwidth = 1500)+
  ggtitle('Highest bid density ') + theme_stata()
p2
```



Lets look at the relationship of this biddy1/revenue with number of bidders

```
p2 <-ggplot(data= df[df$biddy1 < 50000,], aes(x = numbids,y= biddy1))+
  geom_point()
p2
```



######Demi Han, Ruyuan Mei######

```
``{r failed}
#split biddy1 by month
df1<-df[, c("biddy1" ,"months", "monthe", "days" , "daye" , "enddate" , "startdate")]
sp<-split(df1,df1[,c("monthe")],drop=TRUE)
sp1<-data.table(sp)
result1<-lapply(sp1,FUN=function(x) sum(x$AMOUNT))
result2<-lapply(sp1,FUN=function(x) mean(x$AMOUNT))
result<-cbind(result1,result2)
```

```
``{r failed}
#split biddy1 by month
df1<-df[, c("biddy1" ,"months" , "monthe", "days" , "daye" , "enddate" , "startdate")]


g <- split(df1,df1$monthe,)
g
df2<-data.table(g)

g1 <- lapply(df2,mean)
result1<-lapply(df2,FUN=function(x) sum(df2$biddy1))
df3 <- data.table(result1)


sp<-split(df1,df1[,c("monthe","biddy1")],drop=TRUE)
result1<-lapply(sp,FUN=function(x) sum(x$AMOUNT))
result2<-lapply(sp,FUN=function(x) mean(x$AMOUNT))
```

```
result<-cbind(result1,result2)
```

```r
#split biddy1 by month
df1<-df[, c("biddy1" ,"months" , "monthe", "days" , "daye" , "enddate" , "startdate")]
dfmonth2 <- df1[df1$monthe == 2,]
meanmonth2 <- mean(dfmonth2$biddy1)
summonth2 <- sum(dfmonth2$biddy1)
meanmonth2
```

```
## [1] 7851.901
```

```r
summonth2
```

```
## [1] 3596171
```

```r
dfmonth3 <- df1[df1$monthe == 3,]
meanmonth3 <- mean(dfmonth3$biddy1)
summonth3 <- sum(dfmonth3$biddy1)
meanmonth3
```

```
## [1] 8783.061
```

```r
summonth3
```

```
## [1] 19243688
```

```r
dfmonth4 <- df1[df1$monthe == 4,]
meanmonth4 <- mean(dfmonth4$biddy1)
summonth4 <- sum(dfmonth4$biddy1)
meanmonth4
```

```
## [1] 8699.378
```

```r
summonth4
```

```
## [1] 14623654
```

```r
dfmonth5 <- df1[df1$monthe == 5,]
meanmonth5 <- mean(dfmonth5$biddy1)
summonth5 <- sum(dfmonth5$biddy1)
meanmonth5
```

```
## [1] 9081.472
```

```r
summonth5
```

```
## [1] 10961337
```

```r
dfmonth6 <- df1[df1$monthe == 6,]
meanmonth6 <- mean(dfmonth6$biddy1)
summonth6 <- sum(dfmonth6$biddy1)
meanmonth6
```

```
## [1] 9262.462
```

```r
summonth6
```

```
## [1] 8706714
```

```r
dfmonth7 <- df1[df1$monthe == 7,]
meanmonth7 <- mean(dfmonth7$biddy1)
summonth7 <- sum(dfmonth7$biddy1)
meanmonth7
```

```
## [1] 8102.405
```

```r
summonth7
```

```
## [1] 15273034
```

```r
dfmonth8 <- df1[df1$monthe == 8,]
meanmonth8 <- mean(dfmonth8$biddy1)
summonth8 <- sum(dfmonth8$biddy1)
meanmonth8
```

```
## [1] 8195.45
```

```r
summonth8
```

```
## [1] 15702481
```

```r
dfmonth9 <- df1[df1$monthe == 9,]
meanmonth9 <- mean(dfmonth9$biddy1)
summonth9 <- sum(dfmonth9$biddy1)
meanmonth9
```

```
## [1] 8595.723
```

```r
summonth9
```

```
## [1] 17285998
```

```r
dfmonth10 <- df1[df1$monthe == 10,]
meanmonth10 <- mean(dfmonth10$biddy1)
summonth10 <- sum(dfmonth10$biddy1)
meanmonth10
```

```
## [1] 8070.22
```

```r
summonth10
```

```
## [1] 8756189
```

```r
#summary monthly mean
monthe <- c(2,3,4,5,6,7,8,9,10)
mean <- c(meanmonth2,meanmonth3,meanmonth4,meanmonth5,meanmonth6,meanmonth7,meanmonth8,meanmonth9,meanm
#summary monthly sum
sum <- c(summonth2,summonth3,summonth4,summonth5,summonth6,summonth7,summonth8,summonth9,summonth10)

monthly <- data.frame(monthe,mean,sum)
monthly
```

```
##   monthe     mean      sum
## 1      2 7851.901  3596171
## 2      3 8783.061 19243688
## 3      4 8699.378 14623654
## 4      5 9081.472 10961337
## 5      6 9262.462  8706714
## 6      7 8102.405 15273034
## 7      8 8195.450 15702481
## 8      9 8595.723 17285998
## 9     10 8070.220  8756189
```

```r
#graphically biddy1's mean by month
monthly <- tibble(
  month = c("2","3","4","5","6","7","8","9","10"),
```

```r
  mean = c(monthly$mean)
)
knitr::kable(monthly)
```

month	mean
2	7851.901
3	8783.061
4	8699.378
5	9081.472
6	9262.462
7	8102.405
8	8195.450
9	8595.723
10	8070.220

```r
p3 <- ggplot(data = monthly, mapping = aes(
  x = fct_reorder(month, desc(mean)),
  y = mean ))

p3 + geom_col(fill = "lightblue") +
  geom_text(mapping = aes(
    y = mean / 2, label = paste(mean))) +
  scale_y_continuous(breaks = NULL) +
  coord_flip() +
  labs(x = "month",
      y = "mean")
```


```
#summary by days
df1$daye<- as.numeric(df1$daye)
dfmonth_b <- df1[df1$daye <= 10,]
meanmonth_b <- mean(dfmonth_b$biddy1)
summonth_b <- sum(dfmonth_b$biddy1)
meanmonth_b
```

```
## [1] 8649.191
```

```
summonth_b
```

```
## [1] 36343900
```

```
dfmonth_m <- df1[df1$daye >=11 & df1$daye <= 20,]
meanmonth_m <- mean(dfmonth_m$biddy1)
summonth_m <- sum(dfmonth_m$biddy1)
meanmonth_m
```

```
## [1] 8593.81
```

```
summonth_m
```

```
## [1] 42513576
```

```
dfmonth_e <- df1[df1$daye >= 21 & df1$daye <= 31,]
meanmonth_e <- mean(dfmonth_e$biddy1)
summonth_e <- sum(dfmonth_e$biddy1)
meanmonth_e
```

```
## [1] 8353.086
```

```r
summonth_e
```

```
## [1] 35291790
```

```r
#summary mean
period <- c("Beginning of month","Middle of month","Ending of month")
meandays <- c(meanmonth_b,meanmonth_m,meanmonth_e)
#summary sum
sumdays <- c(summonth_b,summonth_m,summonth_e)

daily <- data.frame(period,meandays,sumdays)
daily
```

```
##               period meandays  sumdays
## 1 Beginning of month 8649.191 36343900
## 2    Middle of month 8593.810 42513576
## 3    Ending of month 8353.086 35291790
```

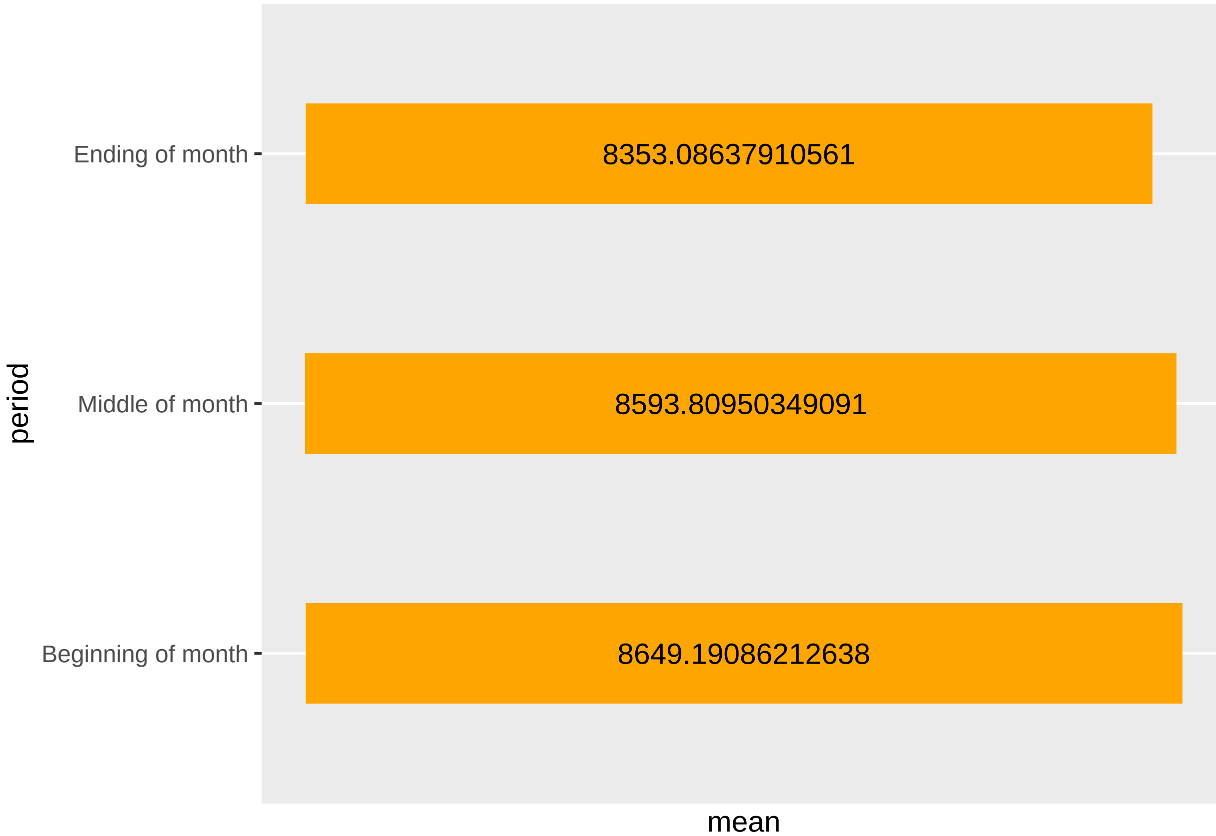
```r
daily <- data.table(daily)
```

```r
#graphically biddy1's mean by days
daily <- tibble(
  period = c(daily$period),
  meandays = c(daily$meandays)
)
knitr::kable(daily)
```

period	meandays
Beginning of month	8649.191
Middle of month	8593.810
Ending of month	8353.086

```r
p4 <- ggplot(data = daily, mapping = aes(
  x = fct_reorder(period, desc(meandays)),
  y = meandays ))

p4 + geom_col(fill = "orange", width = 0.4) +
  geom_text(mapping = aes(
    y = meandays / 2, label = paste(meandays))) +
  scale_y_continuous(breaks = NULL) +
  coord_flip() +
  labs(x = "period",
       y = "mean")
```



#In conclusion, as buyers, we could get a better price in the endding of the month, and avoid buying a car in March to June.

```
#weekdays
df <- read.dta("~/Desktop/ebaydatafinal.dta")
df <- df[df$sell == 1 ,]
library("lubridate")
#converting strings into date format
df$startdate <- parse_date_time(df$startdate, orders="mdy HMS")
df$enddate <- parse_date_time(df$enddate, orders="mdy HMS")
df$wdays <-  wday(df$enddate)

df4 <-df[, c("biddy1" ,  "enddate" , "wdays")]
```

```
#summary by weekdays
df4$wdays <- as.numeric(df4$wdays)

dfweekday <- df4[df4$wdays <= 5,]
meanweekday <- mean(dfweekday$biddy1)
meanweekday
```

```
## [1] 8223.766
```

```
dfweekend <- df4[df4$wdays > 5,]
meanweekend <- mean(dfweekend$biddy1)
meanweekend
```

```
## [1] 8287.095
```

```r
#graphically weekdays vs. weekend
wdays <- c("weekday","weekend")
meanwdays <- c(meanweekday,meanweekend)
weekly <- data.frame(wdays,meanwdays)
weekly
```

```
##     wdays meanwdays
## 1 weekday  8223.766
## 2 weekend  8287.095
```

```r
weekly <- data.table(weekly)
```

```r
#graphically biddy1's mean (weekdays vs. weekend)
weekly <- tibble(
  wdays = c(weekly$wdays),
  meanwdays = c(weekly$meanwdays)
)
knitr::kable(weekly)
```

wdays	meanwdays
weekday	8223.766
weekend	8287.095

```r
p5 <- ggplot(data = weekly, mapping = aes(
  x = wdays,
  y = meanwdays ))
p5 + geom_col(fill = "orange",width = 0.4) +
  geom_text(mapping = aes(
    y = meanwdays / 2, label = paste(meanwdays))) +
  scale_y_continuous(breaks = NULL) +
  coord_flip() +
  labs(x = "wdays",
       y = "mean")
```


8287.09471312211
8223.76605431436
weekend
weekday
wdays
mean