Homework 1

Jiaming Chen

12/12/2017

Data Cleaning

```r
getwd()
```

```
## [1] "/Users/jessie/Desktop/Bittiger/Month1"
```

```r
setwd('/Users/jessie/Desktop/Bittiger/Month1')
list.files()
```

```
## [1] "HW1.Rmd"  "Loan.Rmd"
```

```r
rm(list = ls())
loan <- read.csv('../Data/loan.csv', stringsAsFactors = FALSE)
str(loan)
```

```
## 'data.frame':    887379 obs. of  74 variables:
##  $ id                          : int  1077501 1077430 1077175 1076863 1075358 1075269 1069639 1072053
##  $ member_id                   : int  1296599 1314167 1313524 1277178 1311748 1311441 1304742 1288686
##  $ loan_amnt                   : num  5000 2500 2400 10000 3000 ...
##  $ funded_amnt                 : num  5000 2500 2400 10000 3000 ...
##  $ funded_amnt_inv             : num  4975 2500 2400 10000 3000 ...
##  $ term                        : chr  " 36 months" " 60 months" " 36 months" " 36 months" ...
##  $ int_rate                    : num  10.7 15.3 16 13.5 12.7 ...
##  $ installment                 : num  162.9 59.8 84.3 339.3 67.8 ...
##  $ grade                       : chr  "B" "C" "C" "C" ...
##  $ sub_grade                   : chr  "B2" "C4" "C5" "C1" ...
##  $ emp_title                   : chr  "" "Ryder" "" "AIR RESOURCES BOARD" ...
##  $ emp_length                  : chr  "10+ years" "< 1 year" "10+ years" "10+ years" ...
##  $ home_ownership              : chr  "RENT" "RENT" "RENT" "RENT" ...
##  $ annual_inc                  : num  24000 30000 12252 49200 80000 ...
##  $ verification_status         : chr  "Verified" "Source Verified" "Not Verified" "Source Verified" .
##  $ issue_d                     : chr  "Dec-2011" "Dec-2011" "Dec-2011" "Dec-2011" ...
##  $ loan_status                 : chr  "Fully Paid" "Charged Off" "Fully Paid" "Fully Paid" ...
##  $ pymnt_plan                  : chr  "n" "n" "n" "n" ...
##  $ url                         : chr  "https://www.lendingclub.com/browse/loanDetail.action?loan_id=1
##  $ desc                        : chr  "  Borrower added on 12/22/11 > I need to upgrade my business t
##  $ purpose                     : chr  "credit_card" "car" "small_business" "other" ...
##  $ title                       : chr  "Computer" "bike" "real estate business" "personel" ...
##  $ zip_code                    : chr  "860xx" "309xx" "606xx" "917xx" ...
##  $ addr_state                  : chr  "AZ" "GA" "IL" "CA" ...
##  $ dti                         : num  27.65 1 8.72 20 17.94 ...
##  $ delinq_2yrs                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ earliest_cr_line            : chr  "Jan-1985" "Apr-1999" "Nov-2001" "Feb-1996" ...
##  $ inq_last_6mths              : num  1 5 2 1 0 3 1 2 2 0 ...
##  $ mths_since_last_delinq      : num  NA NA NA 35 38 NA NA NA NA NA ...
##  $ mths_since_last_record      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_acc                    : num  3 3 2 10 15 9 7 4 11 2 ...
##  $ pub_rec                     : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ revol_bal                  : num  13648 1687 2956 5598 27783 ...
##  $ revol_util                 : num  83.7 9.4 98.5 21 53.9 28.3 85.6 87.5 32.6 36.5 ...
##  $ total_acc                  : num  9 4 10 37 38 12 11 4 13 3 ...
##  $ initial_list_status        : chr  "f" "f" "f" "f" ...
##  $ out_prncp                  : num  0 0 0 0 767 ...
##  $ out_prncp_inv              : num  0 0 0 0 767 ...
##  $ total_pymnt                : num  5861 1009 3004 12226 3242 ...
##  $ total_pymnt_inv            : num  5832 1009 3004 12226 3242 ...
##  $ total_rec_prncp            : num  5000 456 2400 10000 2233 ...
##  $ total_rec_int              : num  861 435 604 2209 1009 ...
##  $ total_rec_late_fee         : num  0 0 0 17 0 ...
##  $ recoveries                 : num  0 117 0 0 0 ...
##  $ collection_recovery_fee    : num  0 1.11 0 0 0 0 0 0 2.09 2.52 ...
##  $ last_pymnt_d               : chr  "Jan-2015" "Apr-2013" "Jun-2014" "Jan-2015" ...
##  $ last_pymnt_amnt            : num  171.6 119.7 649.9 357.5 67.8 ...
##  $ next_pymnt_d               : chr  "" "" "" "" ...
##  $ last_credit_pull_d         : chr  "Jan-2016" "Sep-2013" "Jan-2016" "Jan-2015" ...
##  $ collections_12_mths_ex_med : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ mths_since_last_major_derog: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ policy_code                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ application_type           : chr  "INDIVIDUAL" "INDIVIDUAL" "INDIVIDUAL" "INDIVIDUAL" ...
##  $ annual_inc_joint           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ dti_joint                  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ verification_status_joint  : chr  "" "" "" "" ...
##  $ acc_now_delinq             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ tot_coll_amt               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tot_cur_bal                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_acc_6m                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_il_6m                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_il_12m                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_il_24m                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ mths_since_rcnt_il         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ total_bal_il               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ il_util                    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_rv_12m                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_rv_24m                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_bal_bc                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ all_util                   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ total_rev_hi_lim           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ inq_fi                     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ total_cu_tl                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ inq_last_12m               : num  NA NA NA NA NA NA NA NA NA NA ...
```

removing the columns with over 80% of na values.

```r
num.NA <- sort(sapply(loan, function(x) {sum(is.na(x))}), decreasing=TRUE) # na values in each column
remain.col <- names(num.NA)[which(num.NA <= 0.8 * dim(loan)[1])]
loan <- loan[, remain.col] # remaining 57 columns
#loan$annual_inc[which(is.na(loan$annual_inc))] <- median(loan$annual_inc, na.rm = T)
```

Pick up 5 categorical and 5 numerical features

Categorical

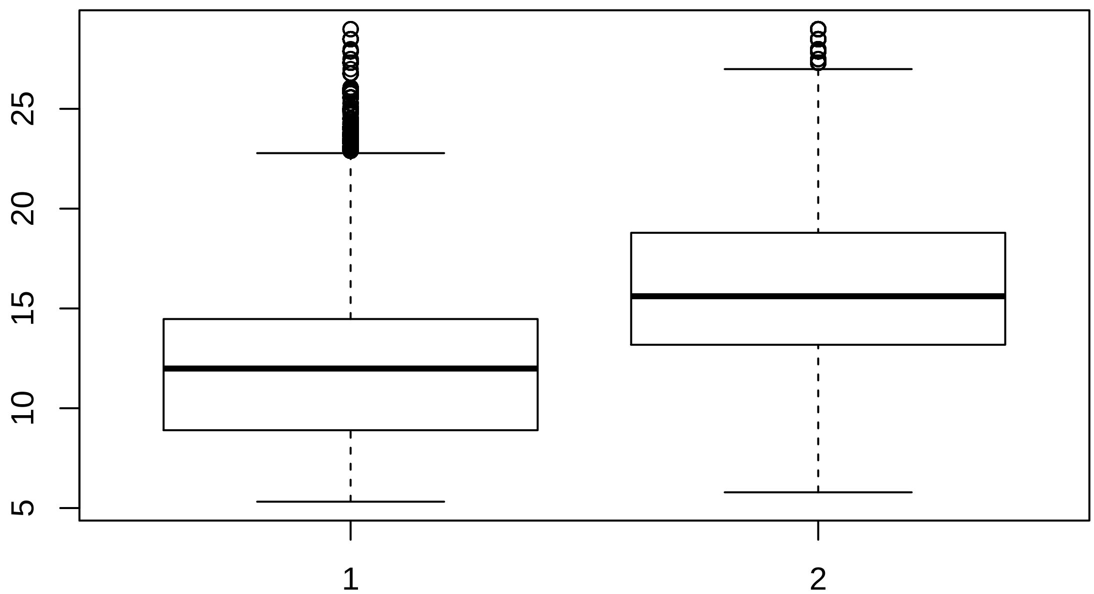
Check which features are numerical and which are categorical. Some features seem to be numerical but they're actually categorical without levels (e.g. id, member_id) or somehow numerical (e.g. last_credit_pull_d, issue_d, last_pymnt_d, earliest_cr_line)

```
is.num <- sapply(loan, is.numeric)
names(is.num[which(is.num!=TRUE)])
```

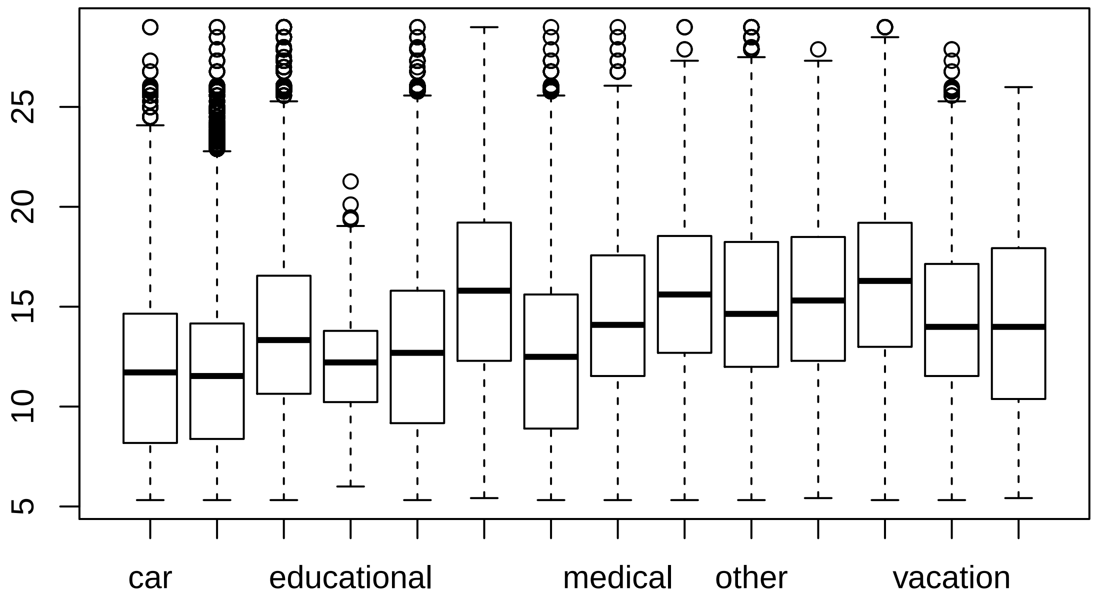
```
##  [1] "term"                      "grade"
##  [3] "sub_grade"                 "emp_title"
##  [5] "emp_length"                "home_ownership"
##  [7] "verification_status"       "issue_d"
##  [9] "loan_status"               "pymnt_plan"
## [11] "url"                       "desc"
## [13] "purpose"                   "title"
## [15] "zip_code"                  "addr_state"
## [17] "earliest_cr_line"          "initial_list_status"
## [19] "last_pymnt_d"              "next_pymnt_d"
## [21] "last_credit_pull_d"        "application_type"
## [23] "verification_status_joint"
```

For categorical features, using boxplot to see if there's a difference of response between each category is a way to explore the predictivity.

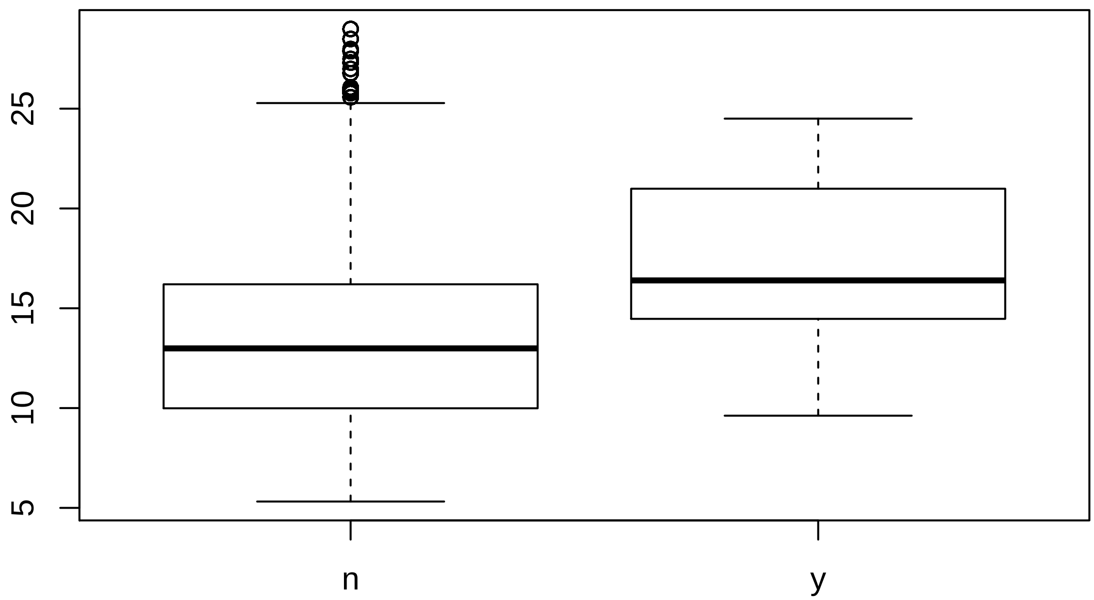
```
# Categorical variable with numerical response
boxplot(subset(loan, term == ' 36 months')$int_rate,
        subset(loan, term == ' 60 months')$int_rate)
```



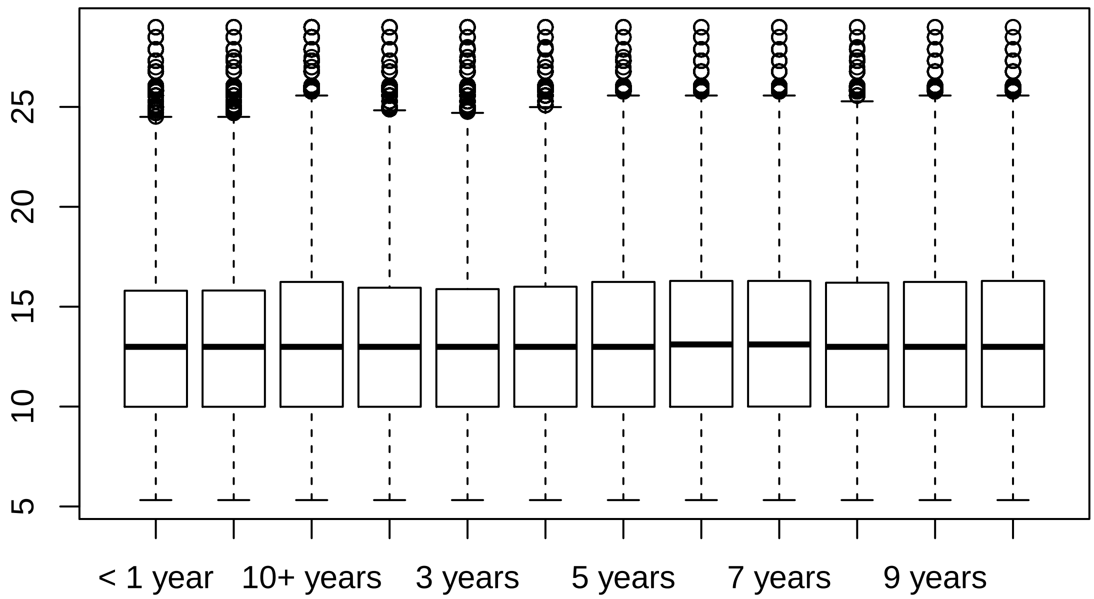

```
boxplot(int_rate ~ purpose, data = loan)
```



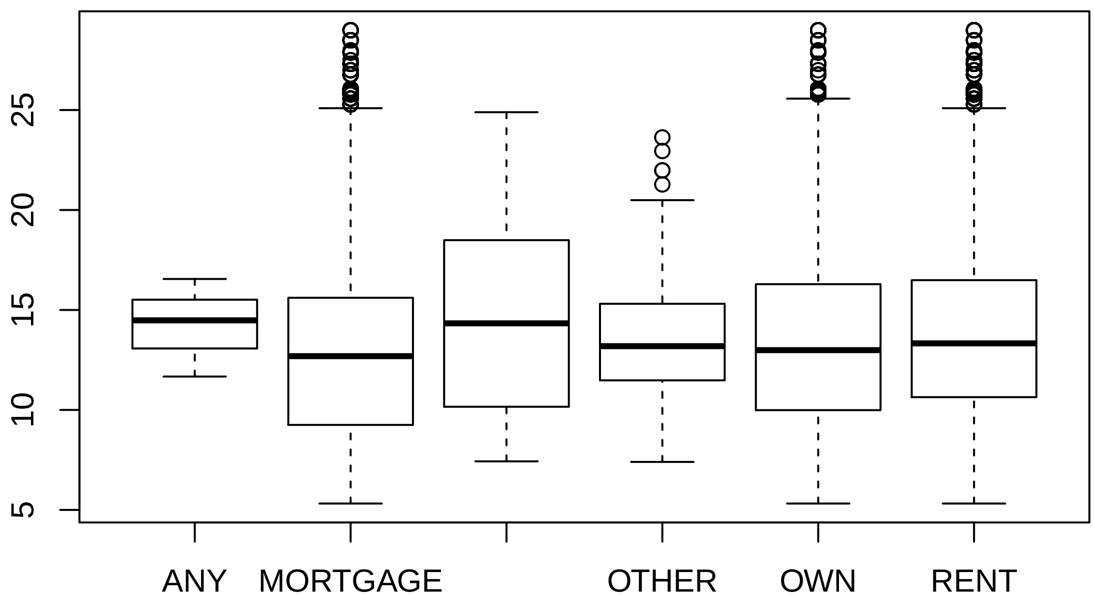

```r
boxplot(int_rate ~ pymnt_plan, data = loan)
```



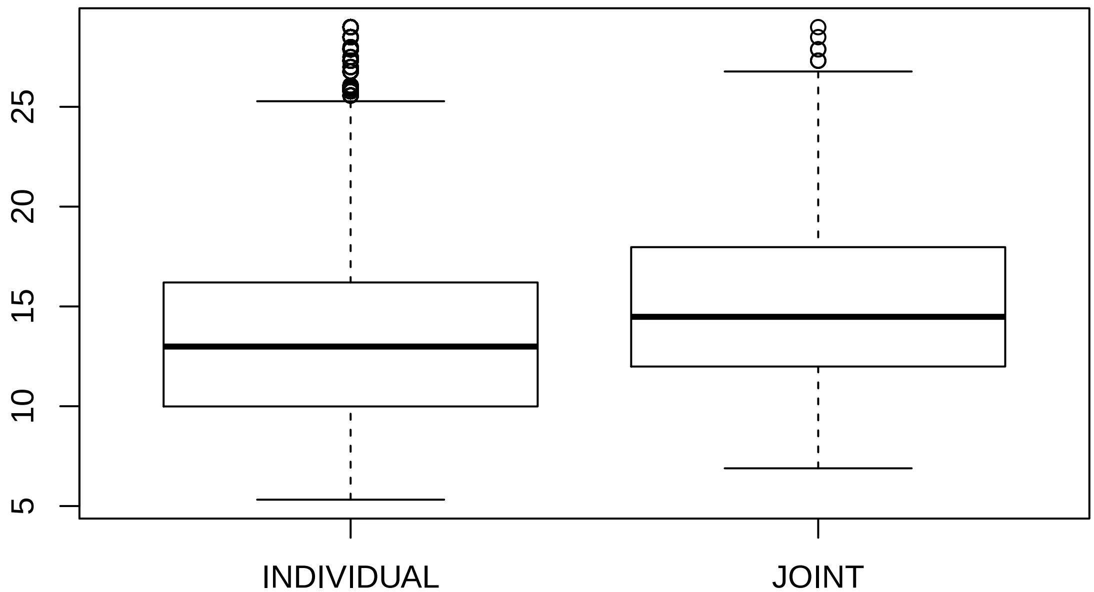

```r
boxplot(int_rate ~ emp_length, data = loan) # this variable does not seem very good for prediction
```



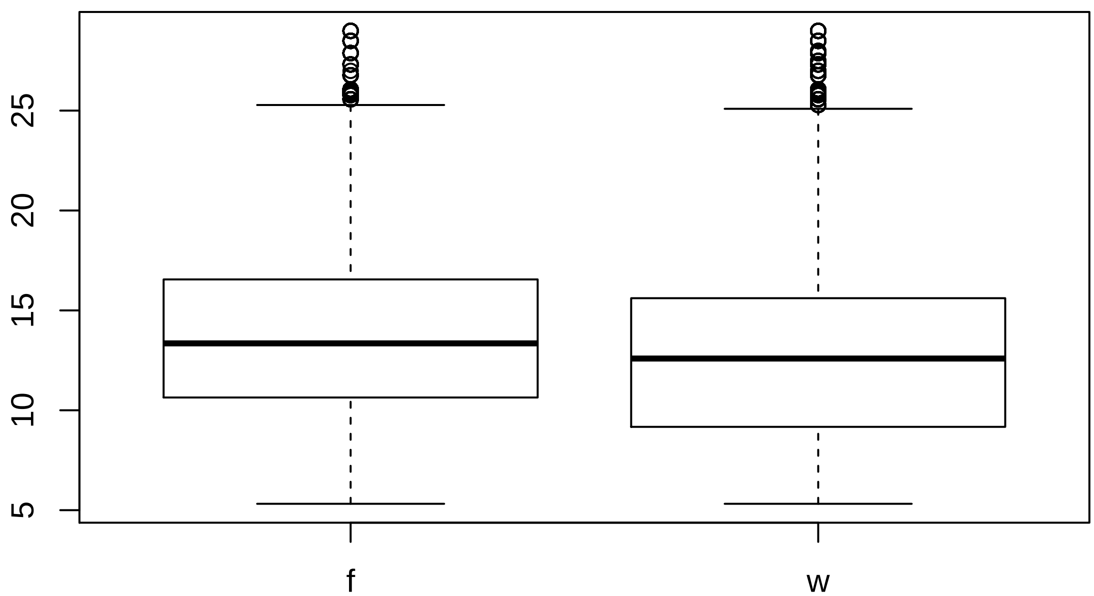

```r
boxplot(int_rate ~ home_ownership, data = loan)
```



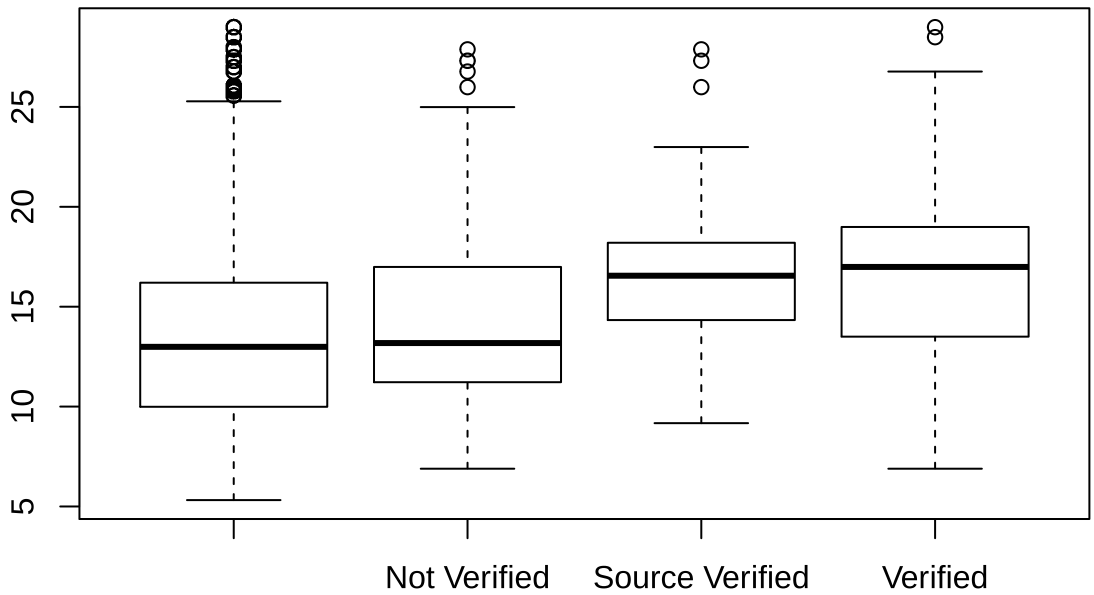

```r
boxplot(int_rate ~ application_type, data = loan)
```




```r
boxplot(int_rate ~ initial_list_status, data = loan)
```




```r
boxplot(int_rate ~ verification_status_joint, data = loan)
```



The five categorical variables I pick: > purpose > pymnt_plan > term > verification_status_joint > home_ownership

I pick these five because in the boxplot the numerical response looks different for different categories. This works the same way as a two sampled t-test. Also I try to avoid choosing two highly correlated variables at the same time, so I prefer to choose variables from different groups.

Numerical

```
names(is.num[which(is.num)])
```

```
##  [1] "mths_since_last_major_derog" "mths_since_last_delinq"
##  [3] "tot_coll_amt"                "tot_cur_bal"
##  [5] "total_rev_hi_lim"            "revol_util"
##  [7] "collections_12_mths_ex_med"  "delinq_2yrs"
##  [9] "inq_last_6mths"              "open_acc"
## [11] "pub_rec"                     "total_acc"
## [13] "acc_now_delinq"              "annual_inc"
## [15] "id"                          "member_id"
## [17] "loan_amnt"                   "funded_amnt"
## [19] "funded_amnt_inv"             "int_rate"
## [21] "installment"                 "dti"
## [23] "revol_bal"                   "out_prncp"
## [25] "out_prncp_inv"               "total_pymnt"
## [27] "total_pymnt_inv"             "total_rec_prncp"
## [29] "total_rec_int"               "total_rec_late_fee"
## [31] "recoveries"                  "collection_recovery_fee"
## [33] "last_pymnt_amnt"             "policy_code"
```

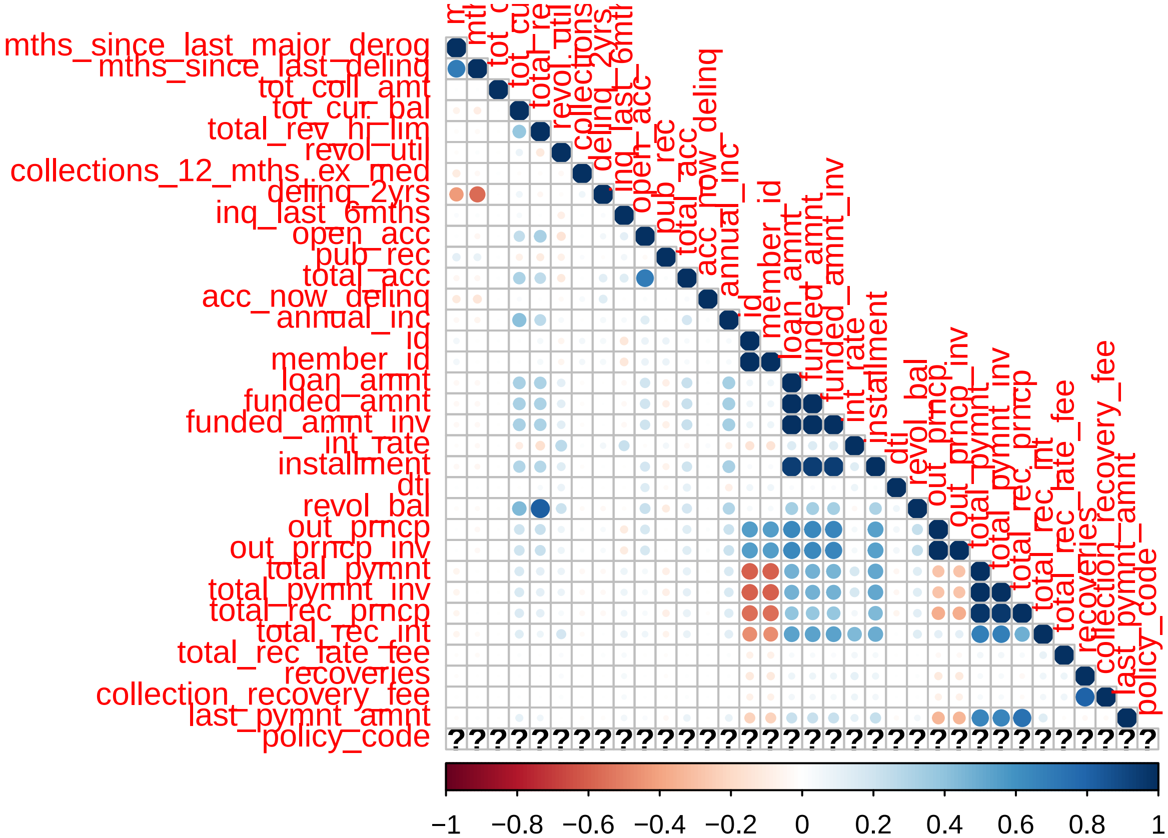
```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
correlations <- cor(loan[, names(is.num[which(is.num)])],
                    use = "pairwise.complete.obs")
```

```
## Warning in cor(loan[, names(is.num[which(is.num)])], use =
## "pairwise.complete.obs"): the standard deviation is zero
```

```r
corrplot(correlations, method = "circle", tl.cex = 1, type = 'lower')
```




```r
sort(abs(correlations[,'int_rate']), decreasing=TRUE) # check the absolute value of the correlation coe
```

```
##                 int_rate            total_rec_int
##              1.000000000              0.445678819
##               revol_util           inq_last_6mths
##              0.269138637              0.227650458
##          total_pymnt_inv              total_pymnt
##              0.171479330              0.170506295
##         total_rev_hi_lim          funded_amnt_inv
##              0.166119251              0.145205285
##              funded_amnt                loan_amnt
##              0.145160337              0.145023099
##                       id                member_id
##              0.142962880              0.142205296
##              installment               recoveries
##              0.133074919              0.106839959
##          last_pymnt_amnt              tot_cur_bal
##              0.101178600              0.091407796
##                      dti               annual_inc
##              0.079902551              0.072785627
##  collection_recovery_fee       total_rec_late_fee
##              0.070867058              0.057150121
##              delinq_2yrs          total_rec_prncp
```

```
##                 0.055177771                 0.054975269
##                     pub_rec                   out_prncp
##                 0.052156163                 0.042671370
##               out_prncp_inv                   total_acc
##                 0.042529006                 0.038618200
##                   revol_bal      mths_since_last_delinq
##                 0.035708090                 0.030032666
##                acc_now_delinq collections_12_mths_ex_med
##                 0.026478461                 0.013335911
## mths_since_last_major_derog                    open_acc
##                 0.011179705                 0.010380950
##                tot_coll_amt
##                 0.001129652
```

By using the correlation matrix, we can find the most correlated variables. The variables with the largest absolute correlation coefficients are the best predictors. > total_rec_int > revol_util > inq_last_6mths > total_pymnt_inv > total_rev_hi_lim

Note that total_pymnt_inv and total_pymnt are highly correlated, so I don't want to include them at the same time

How to generate potential useful features from the data ?

First, we need to clean the data so that we don't variables with too much NA values. And before start exploring features, we need to define our response. Then we can compare the variables by their data type (numerical or categorical). For categorical variables, we can use the concept of t-test and simply use the boxplots to filter out the uncorrelated variables. For numerical variables, we can calculate the correlation matrix and select the ones with higher scores.

However, I think the correlation matrix method is useful for linear data only. If the covariates and response have nonlinear relationship, this approach will not work (please correct me if I was wrong).

What other questions?

I didn't explore the variables with date strings in this homework. I sort of think we need to treat them as numerical variables, but a bit differently (they're time series). By plotting the boxplot of "last_pymnt_d", I actually find that there's some patterns hidden in the time as well. Maybe we can generate new variables/features by feature engineering and make use of them~!s