categorical variable EDA_03

load data

```
rm(list=ls())
loan <- read.csv('/Users/fanyang/Documents/lendingclub/2018_12_21/d2007_2015_loan.csv',
                 header = TRUE, stringsAsFactors = FALSE)
loanT <- loan
```

```
# identify categorical features
categorical_fea <- colnames(loan)[which(sapply(loan, function(x) {return(is.character(x))}))]
print(categorical_fea )
```

```
##  [1] "term"                       "grade"
##  [3] "sub_grade"                  "emp_title"
##  [5] "emp_length"                 "home_ownership"
##  [7] "verification_status"        "issue_d"
##  [9] "loan_status"                "pymnt_plan"
## [11] "desc"                       "purpose"
## [13] "title"                      "zip_code"
## [15] "addr_state"                 "earliest_cr_line"
## [17] "initial_list_status"        "last_pymnt_d"
## [19] "next_pymnt_d"               "last_credit_pull_d"
## [21] "application_type"           "verification_status_joint"
## [23] "issue_d_1"                  "last_pymnt_d_1"
## [25] "issue_d..78"                "last_pymnt_d..80"
## [27] "loan_status..90"            "last_pymnt_d..80_1"
```

```
# For convenience, convert label into categorical type
loan$next_pymnt_binary_l <- ifelse(loan$next_pymnt_binary == '1', 'no', 'yes') # next_pymnt
table(loan$next_pymnt_binary_l)
```

```
##
##     no    yes
##  91652 510127
```

‘term’

```
# test if next_pymnt_binary has the same distribution in short and long term loan
round(with(loan, table(term, next_pymnt_binary_l)) / as.numeric(table(loan$term)),2)
```

```
##            next_pymnt_binary_l
## term          no  yes
##    36 months 0.12 0.88
##    60 months 0.23 0.77
```

```
barplot(with(loan, table(term, next_pymnt_binary_l)) / as.numeric(table(loan$term)), col = c("darkblue", 're
d'), ylab = "ratio of next pymnt over term", xlab = "next payment", beside = TRUE, ylim = c(0.0, 1.0), legen
d = c('36 months', '60 months'), args.legend = list(x = "topleft", bty = "n", inset=c(0, 0)))
```


```
with(loan, chisq.test(next_pymnt_binary_l, term))
```

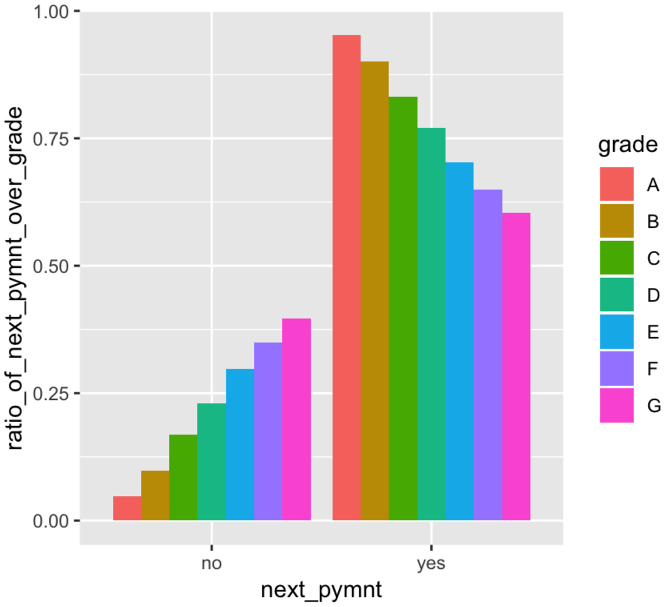
```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  next_pymnt_binary_l and term
## X-squared = 12322, df = 1, p-value < 2.2e-16
```

'grade'

```
# redundant with 'sub_grade'
round(with(loan, table(grade, next_pymnt_binary_l)) / as.numeric(table(loan$grade)),2)
```

```
##      next_pymnt_binary_l
## grade   no  yes
##     A 0.05 0.95
##     B 0.10 0.90
##     C 0.17 0.83
##     D 0.23 0.77
##     E 0.30 0.70
##     F 0.35 0.65
##     G 0.40 0.60
```

```
library("ggplot2")
d <- data.frame(with(loan, table(grade, next_pymnt_binary_l)) / as.numeric(table(loan$grade)))
colnames(d) <- c('grade', 'next_pymnt', 'ratio_of_next_pymnt_over_grade')
p <- ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_over_grade, fill=grade)) +
  geom_bar(stat="identity", position=position_dodge())
p
```




```
with(loan, chisq.test(next_pymnt_binary_l, grade))
```

```
##
##  Pearson's Chi-squared test
##
## data:  next_pymnt_binary_l and grade
## X-squared = 30402, df = 6, p-value < 2.2e-16
```

‘emp_length’

```
table(loan$emp_length)
```

```
##
##  < 1 year    1 year 10+ years   2 years   3 years   4 years   5 years
##     46622     37904    204834     52339     46908     34380     35676
##   6 years   7 years   8 years   9 years       n/a
##     26630     29048     30499     23846     33093
```

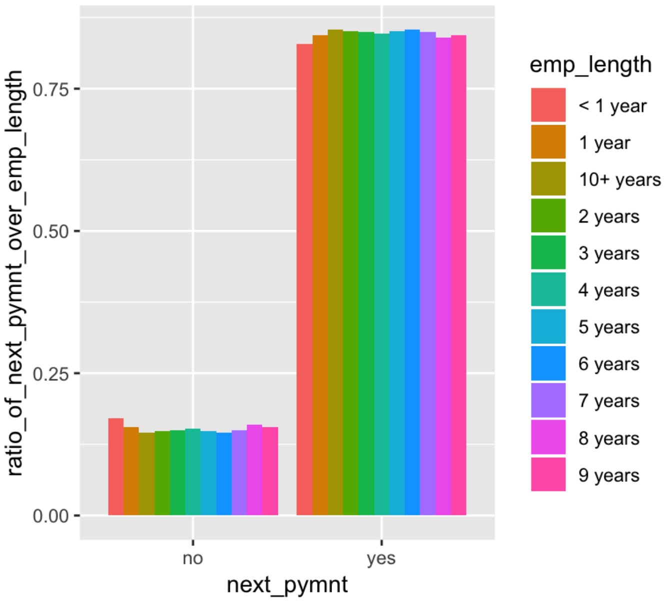
```
loan$emp_length = ifelse(loan$emp_length %in% c('n/a'), '< 1 year', loan$emp_length)
round(with(loan, table(emp_length, next_pymnt_binary_l)) / as.numeric(table(loan$emp_length)),2)
```

```
##              next_pymnt_binary_l
## emp_length      no  yes
##   < 1 year    0.17 0.83
##   1 year      0.16 0.84
##   10+ years 0.15 0.85
##   2 years     0.15 0.85
##   3 years     0.15 0.85
##   4 years     0.15 0.85
##   5 years     0.15 0.85
##   6 years     0.15 0.85
##   7 years     0.15 0.85
##   8 years     0.16 0.84
##   9 years     0.16 0.84
```

```
round(with(loan, table(emp_length, next_pymnt_binary_l)) / as.numeric(table(loan$emp_length)),2)
```

```
##            next_pymnt_binary_l
## emp_length    no  yes
##   < 1 year  0.17 0.83
##   1 year    0.16 0.84
##   10+ years 0.15 0.85
##   2 years   0.15 0.85
##   3 years   0.15 0.85
##   4 years   0.15 0.85
##   5 years   0.15 0.85
##   6 years   0.15 0.85
##   7 years   0.15 0.85
##   8 years   0.16 0.84
##   9 years   0.16 0.84
```

```
d <- data.frame(with(loan, table(emp_length, next_pymnt_binary_l)) / as.numeric(table(loan$emp_length)))
colnames(d) <- c('emp_length', 'next_pymnt', 'ratio_of_next_pymnt_over_emp_length')
p <- ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_over_emp_length, fill=emp_length))+
  geom_bar(stat="identity", position=position_dodge())
p
```




```
with(loan, chisq.test(next_pymnt_binary_l, emp_length))
```

```
##
##  Pearson's Chi-squared test
##
## data:  next_pymnt_binary_l and emp_length
## X-squared = 327.24, df = 10, p-value < 2.2e-16
```

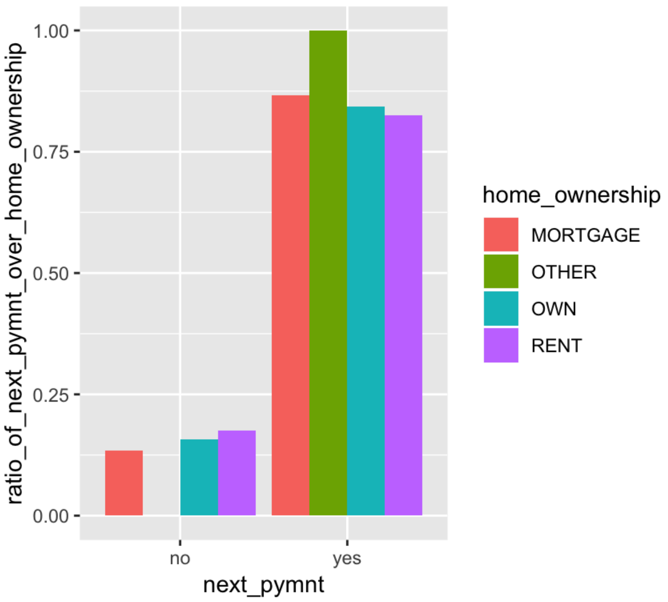
“home_ownership”

```
table(loan$home_ownership)
```

```
##
##      ANY MORTGAGE     NONE    OTHER      OWN     RENT
##        2   303764        2        3    62041   235967
```

```
loan$home_ownership = ifelse(loan$home_ownership %in% c('ANY', 'NONE', 'OTHER'), 'OTHER',
                             loan$home_ownership)
```

```
d <- data.frame(with(loan, table(home_ownership, next_pymnt_binary_l)) / as.numeric(table(loan$home_ownershi
p)))
colnames(d) <- c('home_ownership', 'next_pymnt', 'ratio_of_next_pymnt_over_home_ownership')
p <- ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_over_home_ownership, fill=home_ownership))+
  geom_bar(stat="identity", position=position_dodge())
p
```




```
with(loan, chisq.test(next_pymnt_binary_l, home_ownership))
```

```
## Warning in chisq.test(next_pymnt_binary_l, home_ownership): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  next_pymnt_binary_l and home_ownership
## X-squared = 1798.9, df = 3, p-value < 2.2e-16
```

"verification_status" and "verification_status_joint"

```
table(loan$verification_status)
```

```
##
##    Not Verified Source Verified        Verified
##          171400          243705          186674
```

```
table(loan$verification_status_joint)
```

```
##
##                    Not Verified Source Verified        Verified
##          601338             252              50             139
```

```
loan$verification_status = ifelse(loan$verification_status %in% c('Source Verified', 'Verified'),
                                  'Verified', loan$verification_status)

loan$verification_status_joint = ifelse(loan$verification_status_joint %in% c('Source Verified', 'Verified')
,
                                  'Verified', loan$verification_status_joint)
```

```
table(loan$verification_status)
```

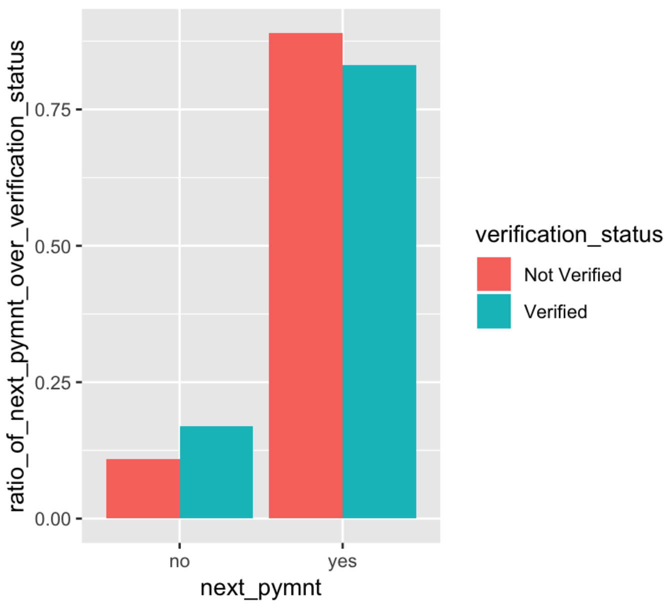
```
##
## Not Verified     Verified
##       171400       430379
```

```
table(loan$verification_status_joint)
```

```
##
##                Not Verified     Verified
##       601338            252          189
```

```
# it is unclear about 'verification_status_joint'
loan$verification_status_joint = NULL
```

```
d <- data.frame(with(loan, table(verification_status, next_pymnt_binary_l)) / as.numeric(table(loan$verifica
tion_status)))
colnames(d) <- c('verification_status', 'next_pymnt', 'ratio_of_next_pymnt_over_verification_status')
p <- ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_over_verification_status, fill=verification_st
atus))+
  geom_bar(stat = "identity", position = position_dodge())
p
```




```
with(loan, chisq.test(verification_status, next_pymnt_binary_l))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  verification_status and next_pymnt_binary_l
## X-squared = 3346.1, df = 1, p-value < 2.2e-16
```

“pymnt_plan”

```
table(loan$pymnt_plan)
```

```
##
##      n      y
## 601776      3
```

```
# delete extremely skewed feature
loan$pymnt_plan = NULL
```

“addr_state” (redundent “zip_code”)

```r
with(loan, table(addr_state, next_pymnt_binary_l)) / as.numeric(table(loan$addr_state))
```

```
##           next_pymnt_binary_l
## addr_state         no        yes
##         AK 0.15724983 0.84275017
##         AL 0.18505808 0.81494192
##         AR 0.19193444 0.80806556
##         AZ 0.15430507 0.84569493
##         CA 0.15021197 0.84978803
##         CO 0.12431401 0.87568599
##         CT 0.13343312 0.86656688
##         DC 0.11795204 0.88204796
##         DE 0.14971098 0.85028902
##         FL 0.16129662 0.83870338
##         GA 0.13769819 0.86230181
##         HI 0.15791292 0.84208708
##         IA 0.00000000 1.00000000
##         ID 0.00000000 1.00000000
##         IL 0.13275958 0.86724042
##         IN 0.15860955 0.84139045
##         KS 0.12167707 0.87832293
##         KY 0.16422872 0.83577128
##         LA 0.17641353 0.82358647
##         MA 0.13989903 0.86010097
##         MD 0.16164751 0.83835249
##         ME 0.12343096 0.87656904
##         MI 0.15191676 0.84808324
##         MN 0.15003655 0.84996345
##         MO 0.16310243 0.83689757
##         MS 0.20105328 0.79894672
##         MT 0.13689095 0.86310905
##         NC 0.15913276 0.84086724
##         ND 0.19369369 0.80630631
##         NE 0.22467772 0.77532228
##         NH 0.10282862 0.89717138
##         NJ 0.16184017 0.83815983
##         NM 0.16874259 0.83125741
##         NV 0.16961652 0.83038348
##         NY 0.16466680 0.83533320
##         OH 0.16202750 0.83797250
##         OK 0.18434164 0.81565836
##         OR 0.10526316 0.89473684
##         PA 0.15858243 0.84141757
##         RI 0.13566328 0.86433672
##         SC 0.12545187 0.87454813
##         SD 0.18083333 0.81916667
##         TN 0.16411907 0.83588093
##         TX 0.15318553 0.84681447
##         UT 0.14282084 0.85717916
##         VA 0.15281237 0.84718763
##         VT 0.09743202 0.90256798
##         WA 0.11623573 0.88376427
##         WI 0.13284816 0.86715184
##         WV 0.12564433 0.87435567
##         WY 0.13785558 0.86214442
```

```r
# display percentage of no next pymnt by each state on map
library("dplyr")
```

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

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

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

```r
library("rgdal")
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.3-6, (SVN revision 773)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01
##  Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/3.5/Resources/library/rgdal/gdal
##  GDAL binary built with GEOS: FALSE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: /Library/Frameworks/R.framework/Versions/3.5/Resources/library/rgdal/proj
##  Linking to sp version: 1.3-1
```

```r
library("choroplethrMaps")
library("sf")
```

```
## Linking to GEOS 3.7.1, GDAL 2.3.2, PROJ 5.2.0
```

```r
library("choroplethr")
```

```
## Loading required package: acs
```

```
## Loading required package: stringr
```

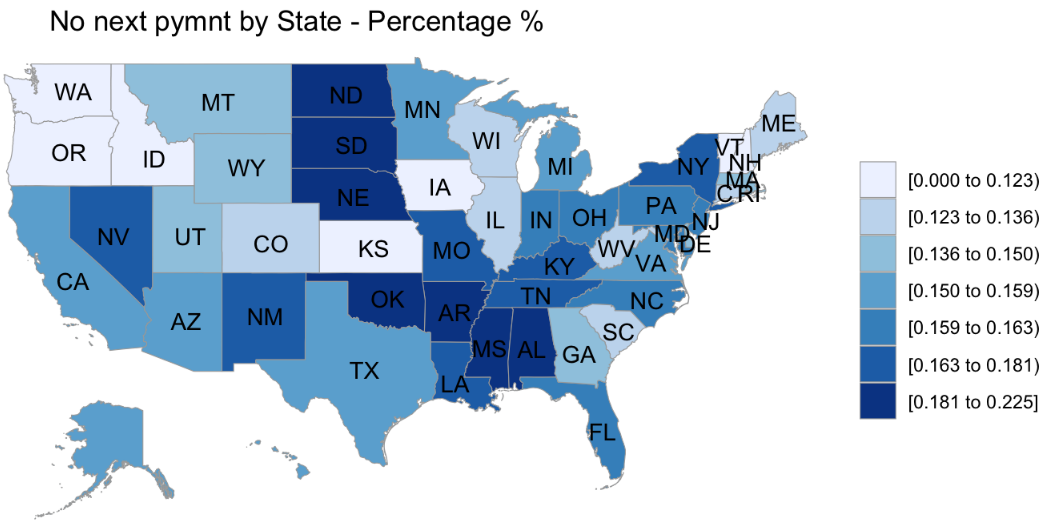
```
## Loading required package: XML
```

```
##
## Attaching package: 'acs'
```

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

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

```r
data("state.regions")

ratio_nonext_pymnt_by_state <- loan %>%
  group_by(addr_state) %>%
  summarize(`Percentage of no next pymnt (%)` = sum(next_pymnt_binary)/length(next_pymnt_binary)) %>%
  arrange(desc(`Percentage of no next pymnt (%)`))

colnames(ratio_nonext_pymnt_by_state) <- c("region","value")

ratio_nonext_pymnt_by_state$region <- sapply(ratio_nonext_pymnt_by_state$region, function(state_code) {

  inx <- grep(pattern = state_code, x = state.regions$abb)

  state.regions$region[inx]

})
state_choropleth(ratio_nonext_pymnt_by_state, title = "           No next pymnt by State - Percentage %")
```



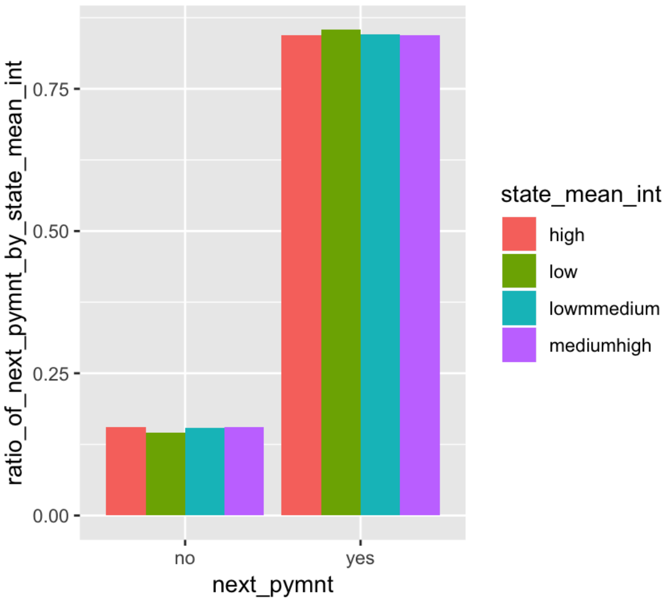

```
# divide add_state into bins based on mean of int_rate by state
int_state = by(loan, loan$addr_state, function(x) { return(mean(x$int_rate)) })
loan$state_mean_int =
  ifelse(loan$addr_state %in% names(int_state)[which(int_state <= quantile(int_state, 0.25))], 'low', ifelse
(loan$addr_state %in% names(int_state)[which(int_state <= quantile(int_state, 0.5))], 'lowmedium', ifelse(l
oan$addr_state %in% names(int_state)[which(int_state <= quantile(int_state, 0.75))], 'mediumhigh', 'high')))
table(loan$state_mean_int)
```

```
##
##       high        low lowmedium mediumhigh
##      61751     170553    232975     136500
```

```
with(loan, table(state_mean_int, next_pymnt_binary_l)) / as.numeric(table(loan$state_mean_int))
```

```
##                next_pymnt_binary_l
## state_mean_int        no       yes
##     high       0.1554145 0.8445855
##     low        0.1457436 0.8542564
##     lowmedium  0.1540466 0.8459534
##     mediumhigh 0.1561099 0.8438901
```

```
d <- data.frame(with(loan, table(state_mean_int, next_pymnt_binary_l)) / as.numeric(table(loan$state_mean_in
t)))
colnames(d) <- c('state_mean_int', 'next_pymnt', 'ratio_of_next_pymnt_by_state_mean_int')
p <- ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_by_state_mean_int, fill=state_mean_int))+
  geom_bar(stat = "identity", position = position_dodge())
p
```




```
with(loan, chisq.test(state_mean_int, next_pymnt_binary_l))
```

```
##
##  Pearson's Chi-squared test
##
## data:  state_mean_int and next_pymnt_binary_l
## X-squared = 82.278, df = 3, p-value < 2.2e-16
```

“purpose”

```
sort(table(loan$purpose))
```

```
##
##        educational   renewable_energy            wedding
##                  1                282                325
##              house           vacation             moving
##               1854               2946               3121
##                car     small_business            medical
##               4937               5020               5324
##     major_purchase              other   home_improvement
##              10308              26607              34980
##        credit_card debt_consolidation
##             149835             356239
```

```
with(loan, table(purpose,next_pymnt_binary_l)) / as.numeric(table(loan$purpose))
```

```
##                     next_pymnt_binary_l
## purpose                      no        yes
##   car                0.11667004 0.88332996
##   credit_card        0.12694631 0.87305369
##   debt_consolidation 0.16355312 0.83644688
##   educational        0.00000000 1.00000000
##   home_improvement   0.13899371 0.86100629
##   house              0.19417476 0.80582524
##   major_purchase     0.14386884 0.85613116
##   medical            0.15984222 0.84015778
##   moving             0.17622557 0.82377443
##   other              0.15819145 0.84180855
##   renewable_energy   0.19858156 0.80141844
##   small_business     0.19741036 0.80258964
##   vacation           0.14290563 0.85709437
##   wedding            0.02461538 0.97538462
```

```
with(loan, chisq.test(purpose,next_pymnt_binary_l))
```

```
## Warning in chisq.test(purpose, next_pymnt_binary_l): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  purpose and next_pymnt_binary_l
## X-squared = 1373.2, df = 13, p-value < 2.2e-16
```

"application_type"

```
sort(table(loan$application_type) )
```

```
##
##      JOINT INDIVIDUAL
##        441     601338
```

```
# delete "application_type"  because it is extremely skewed feature
loan$"application_type" = NULL
```

"desc"

```
length(loan$desc[which(loan$desc == '')]) / dim(loan)[1] # 0.9445527
```

```
## [1] 0.9445527
```

```
loan$desc = NULL
```

"initial_list_status"

```
sort(table(loan$initial_list_status)  )
```

```
##
##      f      w
## 259808 341971
```

```
with(loan, table(initial_list_status, next_pymnt_binary_l)) / as.numeric(table(loan$initial_list_status))
```

```
##                    next_pymnt_binary_l
## initial_list_status        no       yes
##                   f 0.1482441 0.8517559
##                   w 0.1553845 0.8446155
```

```
with(loan, chisq.test(initial_list_status, next_pymnt_binary_l))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  initial_list_status and next_pymnt_binary_l
## X-squared = 58.25, df = 1, p-value = 2.308e-14
```

"earliest_cr_line" and "issue_d"

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
head(loan$earliest_cr_line)
```

```
## [1] "Jan-1996" "Jul-2005" "Feb-1997" "Dec-2001" "May-2006" "Jul-2005"
```

```
head(loan$issue_d)
```

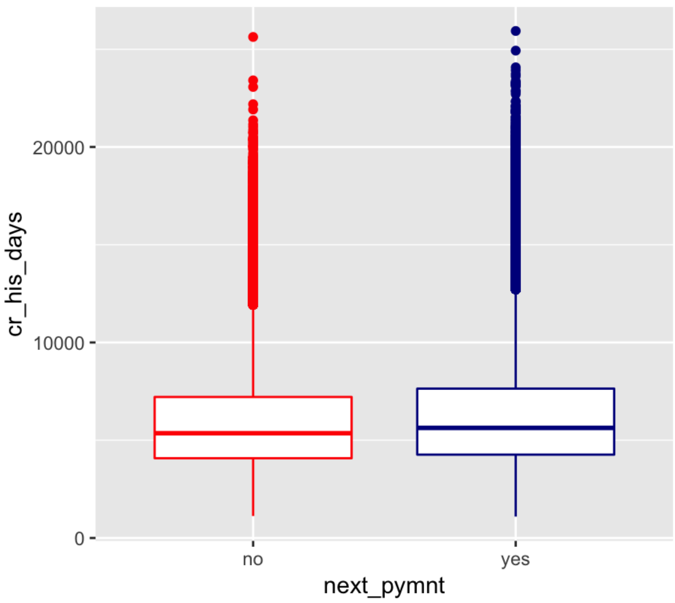
```
## [1] "Dec-2011" "Dec-2011" "Dec-2011" "Dec-2011" "Dec-2011" "Dec-2011"
```

```
# define: credit_history_by_days = loan_issued_date - earliest_credit_line_date
loan$earliest_cr_line_1 = as.Date(as.yearmon(loan$earliest_cr_line, "%b-%Y"))
loan$issue_d_1 = as.Date(as.yearmon(loan$issue_d, "%b-%Y"))
cr_his_days = difftime(loan$issue_d_1, loan$earliest_cr_line_1, units = "days")
loan$cr_his_days=as.numeric(cr_his_days)
summary(loan$cr_his_days)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1095    4232    5599    6150    7578   25933
```

```
d <- loan[,c('next_pymnt_binary_l', 'cr_his_days')]
colnames(d) <- c('next_pymnt', 'cred_his_days')
p <- ggplot(d, aes(x=next_pymnt, y = cr_his_days))+
  geom_boxplot(color=c('red', 'darkblue'))
p
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
# categorical features that are not selected:
# "emp_title", "title", "loan_status", "issue_d..76", "last_pymnt_d..78", "loan_status..88", "last_pymnt_d",
# # "next_pymnt_d"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.