Laboratory Project #1

Commercial Loan Rate Estimation

Ravi Shukla

January 25-29, 2017

Note: My notes are in blue font. The instructions provided by Professor Foote in the document are in red.

I executed all the R commands one line at a time in the console and examined the results in detail to convince myself that the commands worked and gave reasonable results. Once I was convincedthat the commands worked correctly, I copied and pasted them from the console to the R Markdown panel.

Purpose

This project will allow us to practice various R features using live data to support a decision regarding the provision of captive financing to customers at the beginning of this chapter. We will focus on translating regression statistics into R, plotting results, and interpreting ordinary least squares regression outcomes.

Problem

As we researched how to provide captive financing and insurance for our customers, we found that we needed to understand the relationships among lending rates and various terms and conditions of typical equipment financing contracts.

We will focus on one question:

> *What is the influence of terms and conditions on the lending rate of fully committed commercial loans with maturities greater than one year?*

Data

The data set commloan.csv contains data from the St. Louis Federal Reserve Bank's FRED website, which we will use to get some high level insights. The quarterly data extends from the first quarter of 2003 to the second quarter of 2016 and aggregates a survey administered by the St. Louis Fed. There are several time series included. Each loan record is collected by the time that pricing terms were set and by commitment, with maturities more than 365 days from a survey of all commercial banks. Here are the definitions.

Variable	Description	Units of Measure
rate	Weighted-Average Effective Loan Rate	percent
prepay	Percent of Value of Loans Subject to Prepayment Penalty	percent
maturity	Weighted-Average Maturity/Repricing Interval in Days	days
size	Average Loan Size	thousands USD
volume	Total Value of Loans	millions USD

Work Flow

1. Prepare the data.

- Visit the FRED website. Include any information on the site to enhance the interpretation of results.
- Use read.csv to read the data into R. Be sure to set the working directory where the data resides. Use na.omit() to clean the data.

I placed `commloans.csv` in a folder named `data` within my working directory (the directory where the .Rmd file is saved). This allows me to issue the command read.csv("data/commloans.csv") to read the file.

Note that we can set the working directory explicitly. If we don't do that, then the directory where the R Markdown file is saved becomes the working directory. I chose the latter alternative in this program.

Note the use of `n=5` in head and tail to view five records as required in the instructions. Bydefault head and tail show 6 records.

```r
# Read the data file
x.data <- read.csv("data/commloans.csv")
# omit missing data (data with na)
x.data <- na.omit(x.data)
# examine the first five and last five records from the data
head(x.data, n=5)
```

```
##        date prepaypenalty maturity rate size volume
## 1  4/1/2003          16.5      124 3.77  449  11406
## 2  7/1/2003          18.1       70 3.09  356  14586
## 3 10/1/2003          44.9       48 2.83  532  21022
## 4  1/1/2004          30.4       87 3.06  602  21472
## 5  4/1/2004          23.5       68 2.97  600  22359
```

```r
tail(x.data, n=5)
```

```
##         date prepaypenalty maturity rate size volume
## 50  7/1/2015          16.9       76 2.30 1405  30586
## 51 10/1/2015          11.7       77 2.31 1534  36840
## 52  1/1/2016          13.6       66 2.43 1317  36316
## 53  4/1/2016          20.6       93 2.63 1227  24803
## 54  7/1/2016          14.5       66 2.41 1460  40682
```

```r
# Summarize the data
summary(x.data)
```

```
##         date    prepaypenalty      maturity           rate
##  1/1/2004: 1   Min.   : 8.80   Min.   : 40.00   Min.   :2.240
##  1/1/2005: 1   1st Qu.:16.95   1st Qu.: 68.25   1st Qu.:2.482
##  1/1/2006: 1   Median :20.70   Median : 89.00   Median :2.825
##  1/1/2007: 1   Mean   :23.06   Mean   : 95.28   Mean   :3.652
##  1/1/2008: 1   3rd Qu.:29.93   3rd Qu.:112.25   3rd Qu.:4.197
##  1/1/2009: 1   Max.   :51.90   Max.   :396.00   Max.   :7.410
##  (Other) :48
##       size            volume
##  Min.   : 356.0   Min.   :11406
##  1st Qu.: 639.5   1st Qu.:15451
##  Median : 824.5   Median :18670
##  Mean   : 881.7   Mean   :20824
##  3rd Qu.:1017.8   3rd Qu.:24258
##  Max.   :1715.0   Max.   :40682
```

##

- Assign the data to a variable called x.data. Examine the first and last five entries (lookup head()). Run a summary of the data set.
- What anomalies appear based on these procedures?

We have quarterly data from April 1, 2003 to July 1, 2016. The maturity ranges from 40 days to 396 days. This seems inconsistent with what we expect since the database is supposed to be of loans with maturities more than 365 days. Perhaps, the maturity values are shorter because they refer to repricing rather than maturity. The max value (396) seems to be an outlier. Volume seems to be skewed to the right: There are some very large size loans.

2. Explore the data

- Let's plot the time series data using this code:

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(reshape2)
```

```
## Loading required package: reshape2
```

```
# Use melt() from reshape2 to build
# data frame with data as id and
# values of variables
x.melted <- melt(x.data[, c(1:4)], id = "date")
```

Here is an explanation of `melt(x.data[, c(1:4)], id = "date")`: Take all rows and columns 1 through 4 of `x.data` (date, prepaypenalty, maturity and rate) and create time-series using for prepeypenalty, maturity and rate using the `melt` function. For more information about the melt function which is a part of the reshape package, go to http://www.statmethods.net/management/reshape.html.

- Describe the data frame that melt() produces.

```
# Examine the first and last six records of x.melted
head(x.melted)
```

```
##        date      variable value
## 1  4/1/2003 prepaypenalty  16.5
## 2  7/1/2003 prepaypenalty  18.1
## 3 10/1/2003 prepaypenalty  44.9
## 4  1/1/2004 prepaypenalty  30.4
## 5  4/1/2004 prepaypenalty  23.5
## 6  7/1/2004 prepaypenalty  20.0
```

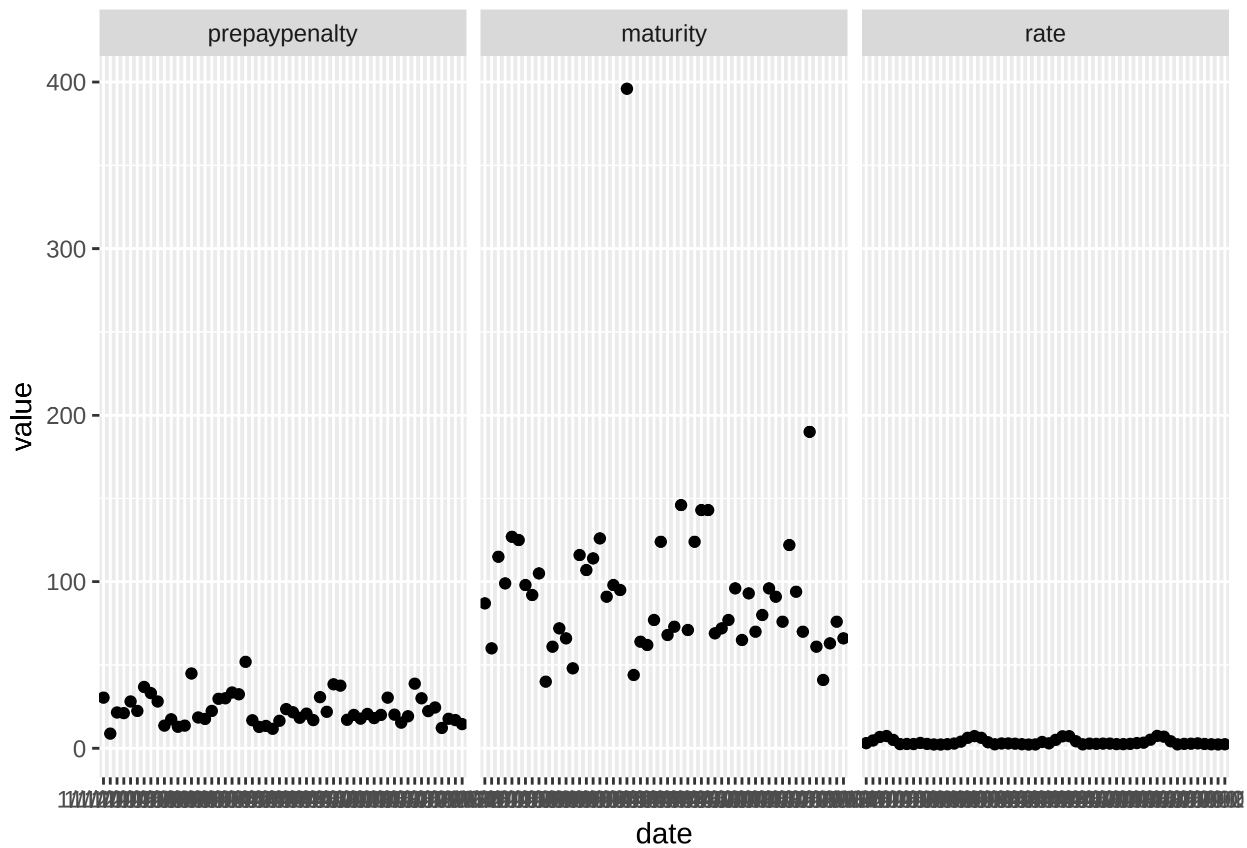
```
tail(x.melted)
```

```
##          date variable value
## 157  4/1/2015     rate  2.47
## 158  7/1/2015     rate  2.30
## 159 10/1/2015     rate  2.31
## 160  1/1/2016     rate  2.43
## 161  4/1/2016     rate  2.63
## 162  7/1/2016     rate  2.41
```

Here is the plot using the code given by Professor Foote.

```
# Plot the data
ggplot(data = x.melted, aes(x = date,
```

```r
y = value)) + geom_point() + facet_wrap(~variable,
scales = "free_x")
```



I also plot all six data items individually. I am reusing the variable name `x.melted`. This is perfectly legal.

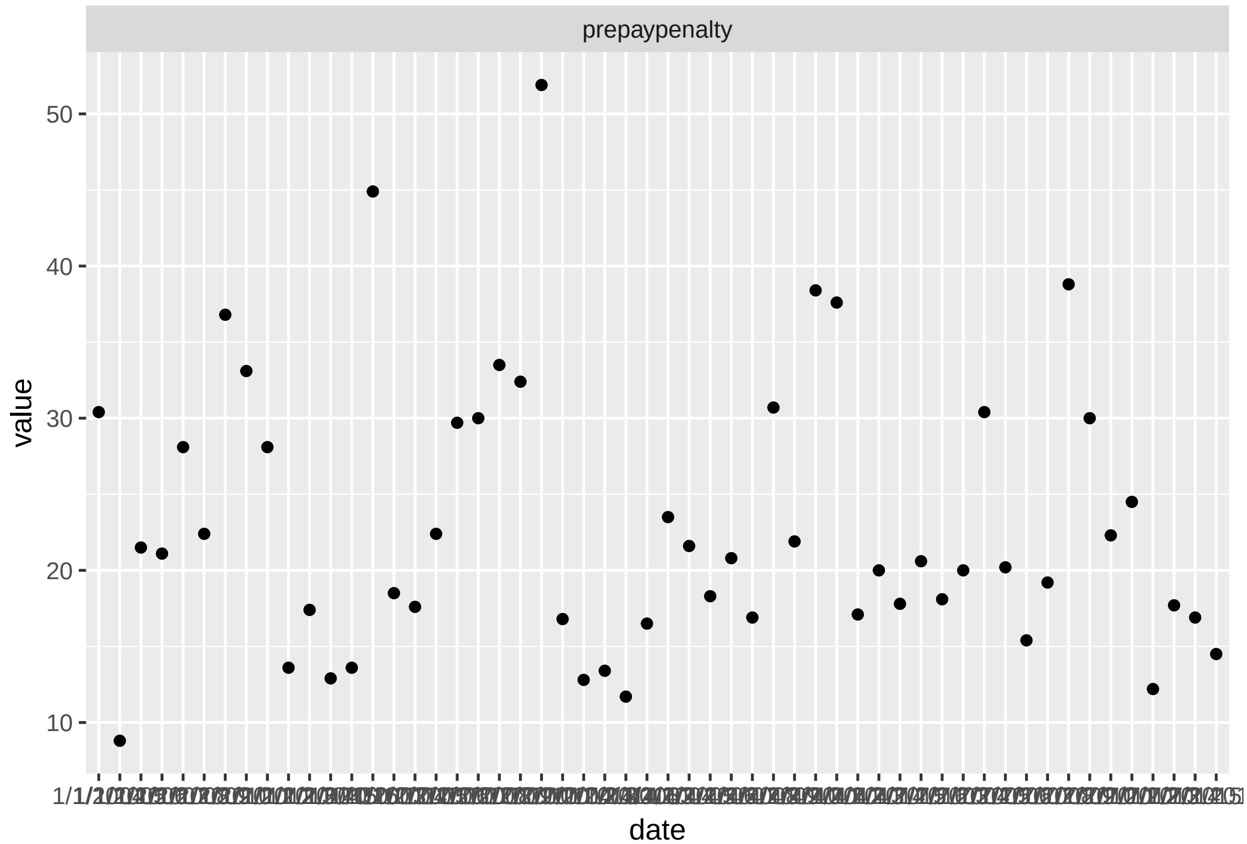
```r
# Variable 1: prepaypenalty
x.melted <- melt(x.data[, c(1,2)], id = "date")
# Examine the first and last six records
head(x.melted)
```

```
##        date      variable value
## 1  4/1/2003 prepaypenalty  16.5
## 2  7/1/2003 prepaypenalty  18.1
## 3 10/1/2003 prepaypenalty  44.9
## 4  1/1/2004 prepaypenalty  30.4
## 5  4/1/2004 prepaypenalty  23.5
## 6  7/1/2004 prepaypenalty  20.0
```

```r
tail(x.melted)
```

```
##         date      variable value
## 49  4/1/2015 prepaypenalty  17.8
## 50  7/1/2015 prepaypenalty  16.9
## 51 10/1/2015 prepaypenalty  11.7
## 52  1/1/2016 prepaypenalty  13.6
## 53  4/1/2016 prepaypenalty  20.6
## 54  7/1/2016 prepaypenalty  14.5
```

```r
# Plot the data
ggplot(data = x.melted, aes(x = date,
y = value)) + geom_point() + facet_wrap(~variable,
scales = "free_x")
```



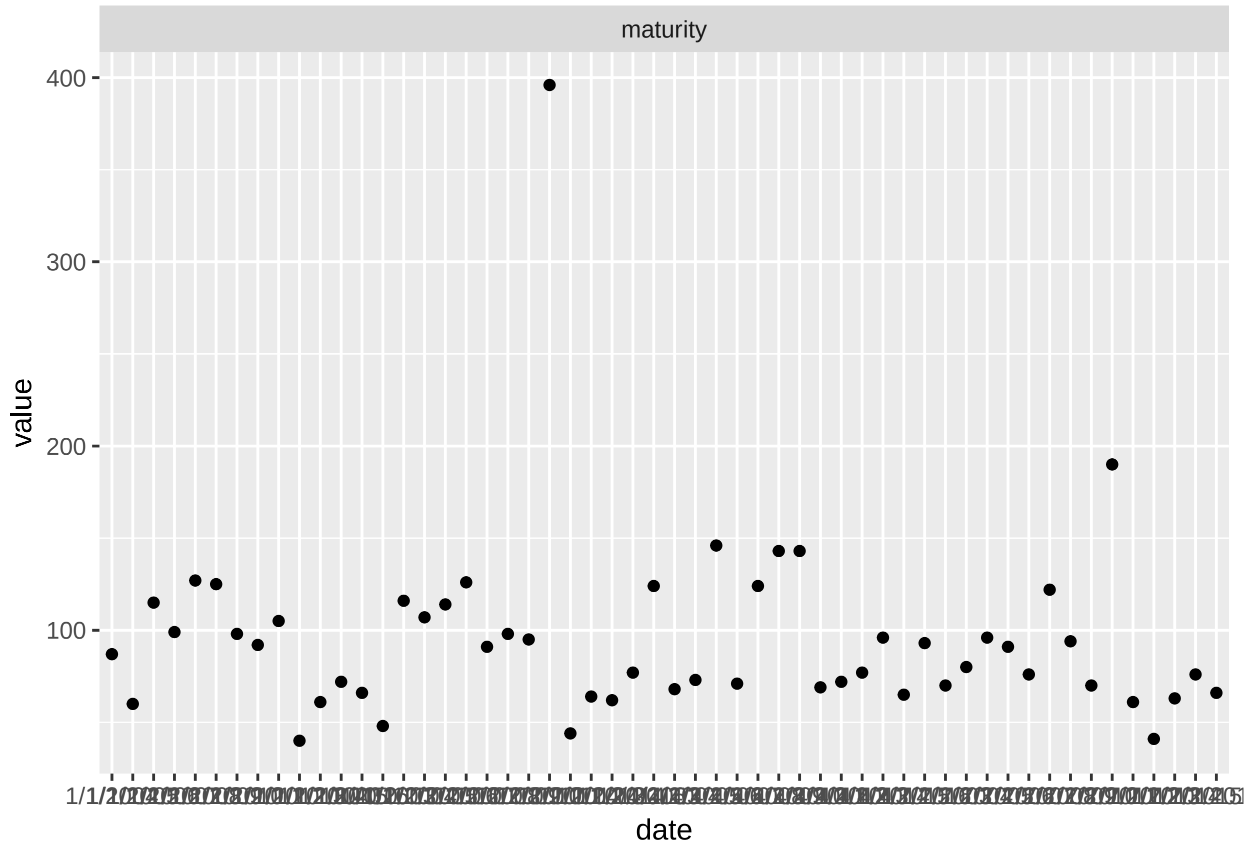

```r
# Variable 2: maturity
x.melted <- melt(x.data[, c(1,3)], id = "date")
# Examine the first and last six records of x.melted
head(x.melted)
```

```
##         date variable value
## 1   4/1/2003 maturity   124
## 2   7/1/2003 maturity    70
## 3  10/1/2003 maturity    48
## 4   1/1/2004 maturity    87
## 5   4/1/2004 maturity    68
## 6   7/1/2004 maturity    80
```

```r
tail(x.melted)
```

```
##          date variable value
## 49   4/1/2015 maturity    65
## 50   7/1/2015 maturity    76
## 51  10/1/2015 maturity    77
## 52   1/1/2016 maturity    66
## 53   4/1/2016 maturity    93
## 54   7/1/2016 maturity    66
```

```r
# Plot the data
ggplot(data = x.melted, aes(x = date,
y = value)) + geom_point() + facet_wrap(~variable,
scales = "free_x")
```



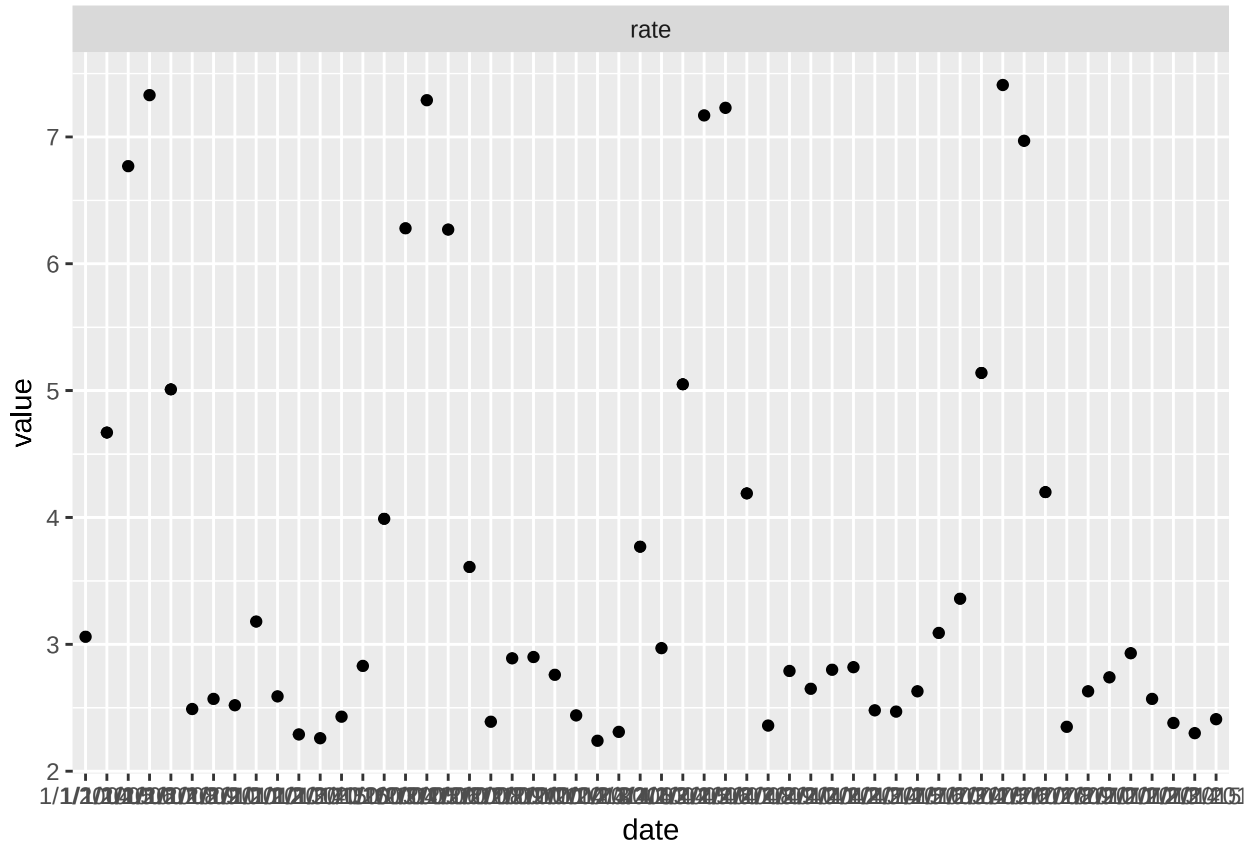

```r
# Variable 3: rate
x.melted <- melt(x.data[, c(1,4)], id = "date")
# Examine the first and last six records of x.melted
head(x.melted)
```

```
##         date variable value
## 1   4/1/2003     rate  3.77
## 2   7/1/2003     rate  3.09
## 3  10/1/2003     rate  2.83
## 4   1/1/2004     rate  3.06
## 5   4/1/2004     rate  2.97
## 6   7/1/2004     rate  3.36
```

```r
tail(x.melted)
```

```
##          date variable value
## 49   4/1/2015     rate  2.47
## 50   7/1/2015     rate  2.30
## 51  10/1/2015     rate  2.31
## 52   1/1/2016     rate  2.43
## 53   4/1/2016     rate  2.63
## 54   7/1/2016     rate  2.41
```

```r
# Plot the data
ggplot(data = x.melted, aes(x = date,
y = value)) + geom_point() + facet_wrap(~variable,
scales = "free_x")
```



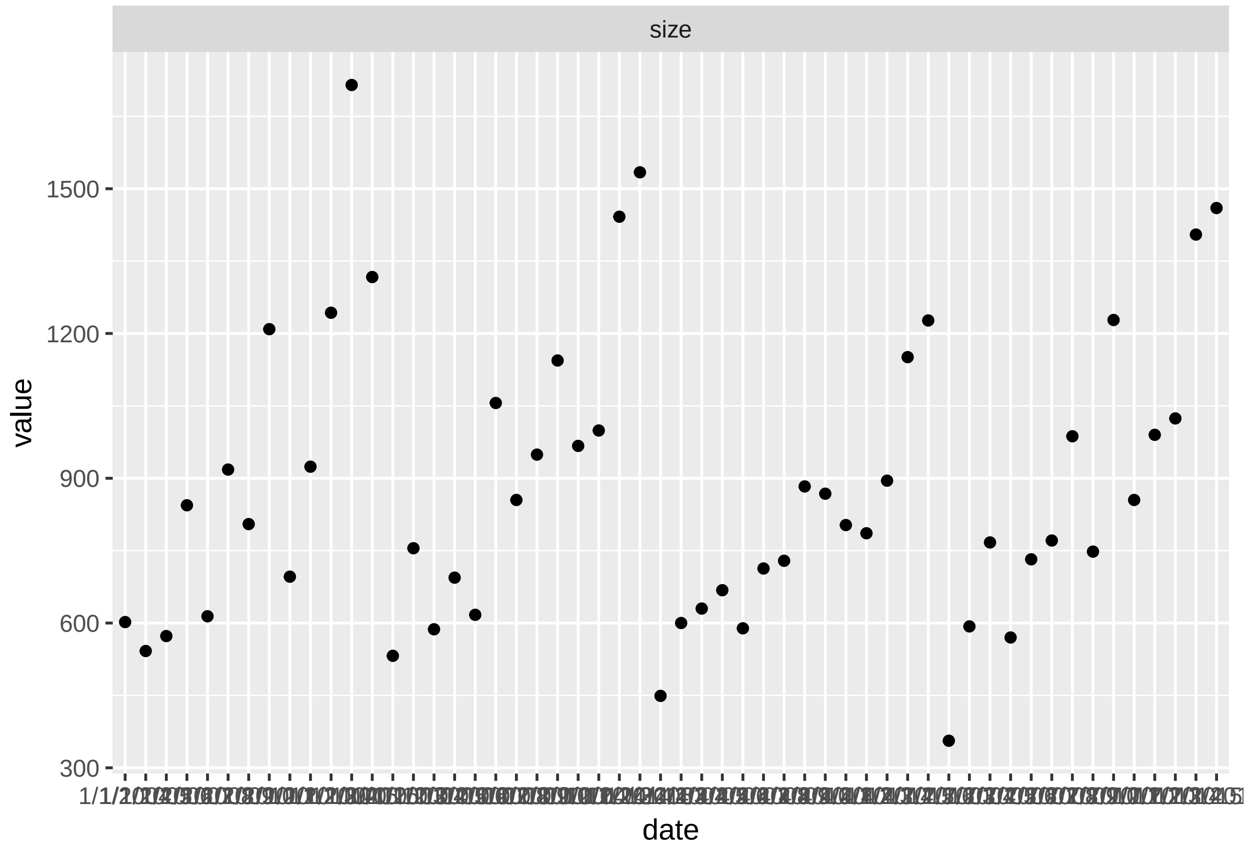

```r
# Variable 4: size
x.melted <- melt(x.data[, c(1,5)], id = "date")
# Examine the first and last six records of x.melted
head(x.melted)
```

```
##        date variable value
## 1  4/1/2003     size   449
## 2  7/1/2003     size   356
## 3 10/1/2003     size   532
## 4  1/1/2004     size   602
## 5  4/1/2004     size   600
## 6  7/1/2004     size   593
```

```r
tail(x.melted)
```

```
##         date variable value
## 49  4/1/2015     size  1151
## 50  7/1/2015     size  1405
## 51 10/1/2015     size  1534
## 52  1/1/2016     size  1317
## 53  4/1/2016     size  1227
## 54  7/1/2016     size  1460
```

```r
# Plot the data
ggplot(data = x.melted, aes(x = date,
y = value)) + geom_point() + facet_wrap(~variable,
scales = "free_x")
```



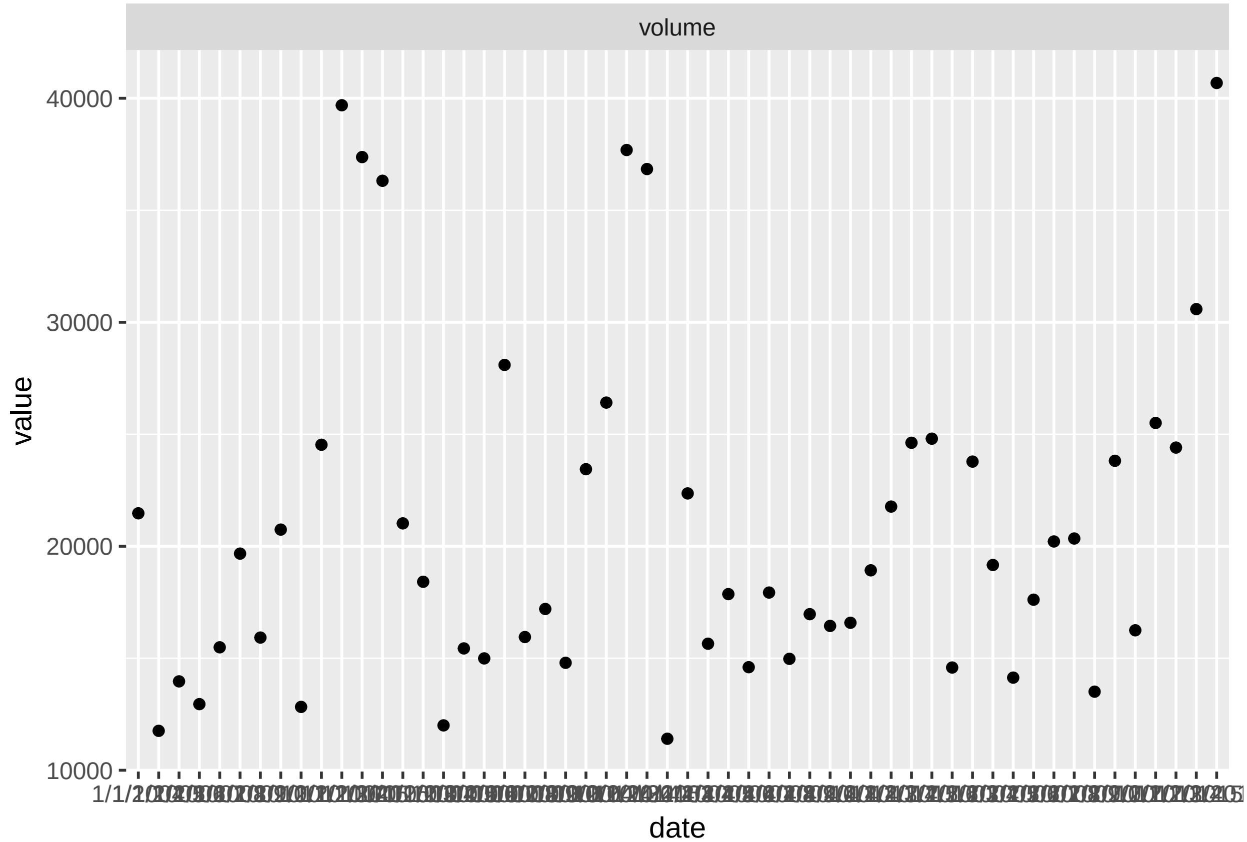

```r
# Variable 4: volume
x.melted <- melt(x.data[, c(1,6)], id = "date")
# Examine the first and last six records of x.melted
head(x.melted)
```

```
##         date variable value
## 1   4/1/2003   volume 11406
## 2   7/1/2003   volume 14586
## 3  10/1/2003   volume 21022
## 4   1/1/2004   volume 21472
## 5   4/1/2004   volume 22359
## 6   7/1/2004   volume 23780
```

```r
tail(x.melted)
```

```
##          date variable value
## 49   4/1/2015   volume 24620
## 50   7/1/2015   volume 30586
## 51  10/1/2015   volume 36840
## 52   1/1/2016   volume 36316
## 53   4/1/2016   volume 24803
## 54   7/1/2016   volume 40682
```

```r
# Plot the data
ggplot(data = x.melted, aes(x = date,
y = value)) + geom_point() + facet_wrap(~variable,
scales = "free_x")
```



- Let's load the psych library and produce a scatterplot matrix. Interpret this exploration.

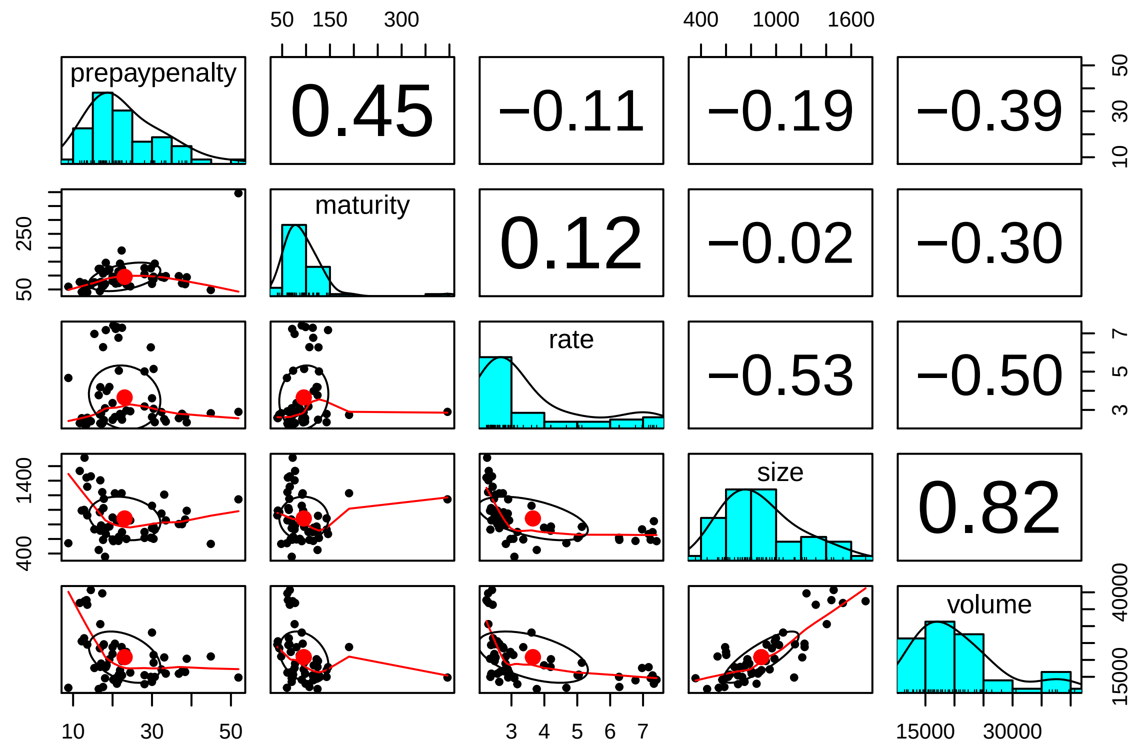
I create the scatterplot for only the five variables of interest which are in columns 2:6 of `x.data`.

```r
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
pairs.panels(x.data[,2:6])
```



Loan size and volume are highly positively correlated ($\rho = 0.82$). So, the higher the loan size, the higher the loan volume.Loan rate is mildly negatively correlated with size and volume. So, loans with low interest rates are smaller in size and have lower volume.

3. Analyze the data.

Let's regress `rate` on the rest of the variables in `x.data`. To do this we form a matrix of independent variables (predictor or explanatory variables) in the matrix $\mathbf{X}$ and a separate vector $\mathbf{y}$ for the dependent (response) variable rate. We recall that the $\mathbf{1}$ vector will produce a constant intercept in the regression model.

```
y <- as.vector(x.data[, "rate"])
X <- as.matrix(cbind(1, x.data[, c("prepaypenalty", "maturity", "size", "volume")]))
head(y)
```

```
## [1] 3.77 3.09 2.83 3.06 2.97 3.36
```

```
head(X)
```

```
##   1 prepaypenalty maturity size volume
## 1 1          16.5      124  449  11406
## 2 1          18.1       70  356  14586
## 3 1          44.9       48  532  21022
## 4 1          30.4       87  602  21472
## 5 1          23.5       68  600  22359
## 6 1          20.0       80  593  23780
```

- Explain the code used to form `y` and `X`.

`as.vector(x.data[, ¨rate]¨)` takes the `rate` column from `x.data` and creates a column vector `y`. `as.matrix(cbind(1, x.data[, c("prepaypenalty", "maturity", "size", "volume")]))` takes four columns identified in the formula and creates a matrix `X` of 4 columns.

- Calculate the $\hat{\beta}$ coefficients and interpret their meaning.

```
XtX.inverse <- solve(t(X) %*% X)
(beta.hat <- XtX.inverse %*% t(X) %*% y)
```

```
##                        [,1]
## 1               7.771438e+00
## prepaypenalty -6.968996e-02
## maturity       6.399952e-03
## size          -2.041351e-03
## volume        -6.347851e-05
```

I used the code provided by Professor Foote. The code is based on the standard multiple regression model:

$$\text{rate} = \beta_0 + \beta_1 \text{ prepaypenalty} + \beta_2 \text{ maturity} + \beta_3 \text{ size} + \beta_4 \text{ volume} + \epsilon$$

Writing the regression equation in matrix notation (See slide 11 of section 1.7 from asynchronous video for unit 1). $\mathbf{y}$ is the $n \times 1$ vector of rates, the dependent variable, $\mathbf{X}$ is the $n \times 4$ matrix of independent variables (prepaypenalty, maturity, size and volume), $\mathbf{B}$ is the 4×1 vector of coefficients (βs) and $\mathbf{E}$ is the $n \times 1$ vector of errors (ϵs). n is the number of observations. My fonts are somewhat different than those in Professor Foote's slides.

$$\mathbf{y} = \mathbf{XB} + \mathbf{E}$$

Now we can find the regression coefficients using standard regression

$$\hat{\mathbf{B}} = \left(\mathbf{X^T X}\right)^{-1} \mathbf{X^T y}$$

The `t` function in used to transpose the matrix and `solve` function is used to invert the matrix. The process is implemented in two steps here: First we calculate $\left(\mathbf{X^T X}\right)^{-1}$ as `XTX.inverse` and then multiply it by $\mathbf{X^T y}$ to get $\hat{\mathbf{B}}$ as `beta.hat`. Note that the parentheses around the expression makes R display the result (`beta.hat` here) without us having to ask for it explicitly.

The coefficients show that the dependent variable, `rate`, is positively related to maturity but negatively related to the other three variables `prepaypenalty`, `size` and `volume`.

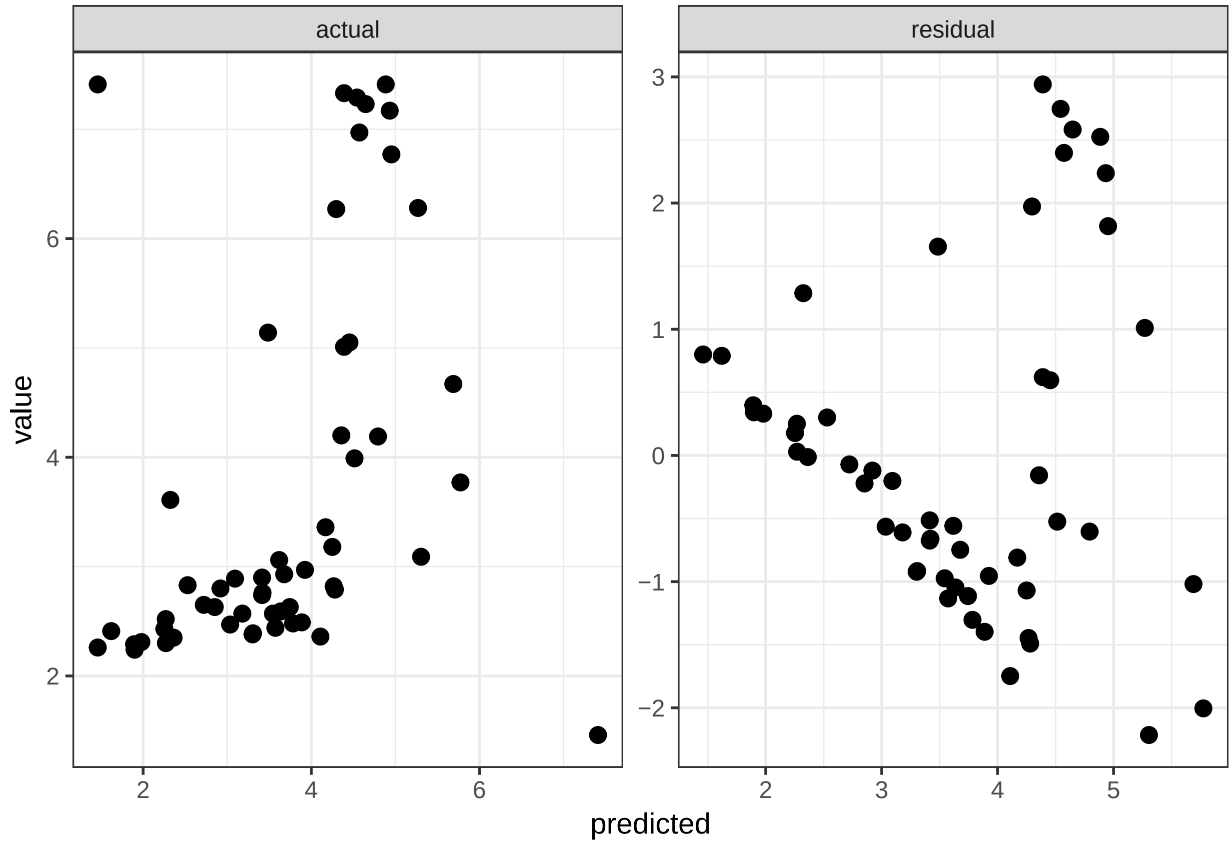
- Calculate actual and predicted rates and plot using this code.

```
# Insert comment here
#require(reshape2) # omitting this line since reshape2 has already been loaded
#require(ggplot2) # omitting this line since ggplot2 has already been loaded
actual <- y
predicted <- X %*% beta.hat
residual <- actual - predicted
results <- data.frame(actual = actual,
predicted = predicted, residual = residual)
head(predicted)
```

```
##       [,1]
## 1 5.774545
## 2 5.305428
## 3 2.529112
## 4 3.617755
## 5 3.924794
## 6 4.169595
```

```
# Insert comment here
min_xy <- min(min(results$actual), min(results$predicted))
```

```r
max_xy <- max(max(results$actual), max(results$predicted))
# Insert comment here
plot.melt <- melt(results, id.vars = "predicted")
# Insert comment here
plot.data <- rbind(plot.melt, data.frame(predicted = c(min_xy,
max_xy), variable = c("actual", "actual"),
value = c(max_xy, min_xy)))
# Insert comment here
p <- ggplot(plot.data, aes(x = predicted, y = value)) + geom_point(size = 2.5) + theme_bw()
p <- p + facet_wrap(~variable, scales = "free")
p
```




```r
# Calculate the errors, sum of squared errors and standard error of the regression
e <- y - X %*% beta.hat
(e.sse <- t(e) %*%e)
```

```
##          [,1]
## [1,] 88.62341
```

```r
(n <- dim(X)[1])
```

```
## [1] 54
```

```r
(k <- dim(beta.hat)[1])
```

```
## [1] 5
```

```r
(e.se <- (e.sse / (n - k))^0.5)
```

```
##          [,1]
## [1,] 1.344857
```

Another way to conduct the regression analysis (estimate the coefficients and calculate the SSE) is by using the `lm` function whch estimates the linear model. You can get help on the lm model at https://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html. I define `Z` as the matrix of independent variables.

```r
Z <- as.matrix(cbind(x.data[, c("prepaypenalty", "maturity", "size", "volume")]))
head(Z)
```

```
##   prepaypenalty maturity size volume
## 1          16.5      124  449  11406
## 2          18.1       70  356  14586
## 3          44.9       48  532  21022
## 4          30.4       87  602  21472
## 5          23.5       68  600  22359
## 6          20.0       80  593  23780
```

```r
# Estimate a linear model between y and Z. Remember that Z consists of four variables.
lmresult=lm(y~Z)
summary(lmresult)
```

```
##
## Call:
## lm(formula = y ~ Z)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2154 -0.9465 -0.2121  0.6145  2.9405
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.771e+00  9.423e-01   8.247 8.02e-11 ***
## Zprepaypenalty -6.969e-02  2.396e-02  -2.908  0.00545 **
## Zmaturity       6.400e-03  4.377e-03   1.462  0.15005
## Zsize          -2.041e-03  1.194e-03  -1.710  0.09364 .
## Zvolume        -6.348e-05  5.056e-05  -1.255  0.21528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.345 on 49 degrees of freedom
## Multiple R-squared:  0.4026, Adjusted R-squared:  0.3539
## F-statistic: 8.256 on 4 and 49 DF,  p-value: 3.581e-05
```

```r
# Get predicted/fitted values
predictedvalues=fitted.values(lmresult)
head(predictedvalues)
```

```
##        1        2        3        4        5        6
## 5.774545 5.305428 2.529112 3.617755 3.924794 4.169595
```

```r
# Calculate error based on the lm model for all data points
elm =  y - predicted
head(elm)
```

```
##          [,1]
## 1 -2.0045453
## 2 -2.2154279
## 3  0.3008875
## 4 -0.5577551
## 5 -0.9547941
## 6 -0.8095949
```

```
# Calculate the sum of squared errors and display it
(elm.sse=t(elm)%*%elm)
```

```
##          [,1]
## [1,] 88.62341
```

```
(n <- dim(Z)[1])
```

```
## [1] 54
```

```
(k <- dim(Z)[2])
```

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

```
(elm.se <- (elm.sse / (n - k - 1 ))^0.5)
```

```
##          [,1]
## [1,] 1.344857
```

Observations and Recommendations

The analysis shows that the interest rate on the loan is related in the statistically significant manner to the prepayment penalty (`prepaypenalty`). While `rate` is related positively with maturity and negatively with size and volume, those relationships are not statistically significant. The R^2 of 0.4026 or 40.26% indicates that we can have a reaosnable confidence in the model. To explore the reltionship more, we should regress rate on prepaypenalty alone to see how well this one variable alone explains the variability in rate.

Sources

- Various discussions on http://stackoverflow.com
- http://www.statmethods.net/management/reshape.html
- https://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html