

Capstone Day 1: Prepare Data

Adam Rich

July 25, 2018

The goal for day 1 is to create two datasets

- `pol_final` for analyzing frequency
- `claims_final` for analyzing severity

There are a few steps that have to be done

1. Load the four data files to memory
2. Spread `pol_rating`
3. Join the new wide `pol_rating` object to `pol_dates`
4. Join with the state lookup table
5. Put rating characteristics back in claims table
6. Aggregate claims data by policy
7. Join agg claims with policy data
8. Add some derived columns
9. Do some sense checking
10. Save files

I'll use `tidyverse` package because it automatically loads `dplyr` and `tidyr`. I'll need `tidyr` to “reshape” or “spread” the `pol_rating` data object. The capstone project ZIP also came with `resources.R` so let's `source` that, too.

```
require(tidyverse)

## Loading required package: tidyverse

## -- Attaching packages ----- tidyverse 1.2.1

## v ggplot2 3.0.0      v purrr   0.2.5
## v tibble  1.4.2      v dplyr  0.7.6
## v tidyr   0.8.1      v stringr 1.3.1
## v readr   1.1.1      v forcats 0.3.0

## -- Conflicts ----- tidyverse_conflicts_0.2.0
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

source('c:/home/git/other/ratemaking-capstone/R/resources.R')
```

Load the four data files to memory

Note: Putting parenthesis around a statement will force that statement's return value to be printed. Usually a function's return value *is* printed, but some, like `load` try to be “silent”. Since we are writing a report, I want to see the output.

The return value of `load` is a vector giving the names of the objects loaded.

```
(load("c:/home/git/other/ratemaking-capstone/share/claims.RData"))

## [1] "claims"
```

```
(load("c:/home/git/other/ratemaking-capstone/share/pol_dates.RData"))
```

```
## [1] "pol_dates"
```

```
(load("c:/home/git/other/ratemaking-capstone/share/pol_rating.RData"))
```

```
## [1] "pol_rating"
```

```
state_lookup <- read.csv('c:/home/git/other/ratemaking-capstone/share/states.csv', stringsAsFactors = F)
```

This is what the four data frames look like.

```
head(claims)
```

```
##   policy_number claim_ultimate claim_number
## 1  C1AE00783351      22447.10 CR0080343074
## 2  C1AE00075999      18380.63 CR0010605425
## 3  C1AE00141200     141429.06 CR0034027774
## 4  C1AE00264573      18057.69 CR0050581498
## 5  C1AE00212315      85790.85 CR0090796329
## 6  C1AE00212315     412740.42 CR0003635914
```

```
str(claims)
```

```
## 'data.frame':   15010 obs. of  3 variables:
##  $ policy_number : chr  "C1AE00783351" "C1AE00075999" "C1AE00141200" "C1AE00264573" ...
##  $ claim_ultimate: num  22447 18381 141429 18058 85791 ...
##  $ claim_number  : chr  "CR0080343074" "CR0010605425" "CR0034027774" "CR0050581498" ...
```

```
head(pol_dates)
```

```
##   policy_number inception expiration
## 1  C1AE00092766    201112    201212
## 2  C1AE00783351    200802    200902
## 3  C1AE00936879    201310    201410
## 4  C1AE00037943    200802    200902
## 5  C1AE00594232    201101    201201
## 6  C1AE00922402    201411    201511
```

```
str(pol_dates)
```

```
## 'data.frame':   100000 obs. of  3 variables:
##  $ policy_number: chr  "C1AE00092766" "C1AE00783351" "C1AE00936879" "C1AE00037943" ...
##  $ inception    : chr  "201112" "200802" "201310" "200802" ...
##  $ expiration    : chr  "201212" "200902" "201410" "200902" ...
```

```
head(pol_rating)
```

```
##   policy_number variable  value
## 1  C1AE00092766  revenue  379061
## 2  C1AE00783351  revenue 3771609
## 3  C1AE00936879  revenue   87795
## 4  C1AE00037943  revenue   59671
## 5  C1AE00594232  revenue  183667
## 6  C1AE00922402  revenue  950935
```

```
str(pol_rating)
```

```
## 'data.frame':   700000 obs. of  3 variables:
##  $ policy_number: chr  "C1AE00092766" "C1AE00783351" "C1AE00936879" "C1AE00037943" ...
```

```
## $ variable      : Factor w/ 7 levels "revenue","state",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ value         : chr  "379061" "3771609" "87795" "59671" ...
```

```
head(state_lookup)
```

```
##           State Frequency.Group Population
## 1    California             High  39250017
## 2      New York             Mid   19745289
## 3       Texas             Mid   27862596
## 4     Florida             High   20612439
## 5    Illinois             High   12801539
## 6 Pennsylvania            Mid   12784227
```

```
str(state_lookup)
```

```
## 'data.frame':   51 obs. of  3 variables:
## $ State         : chr  "California" "New York" "Texas" "Florida" ...
## $ Frequency.Group: chr  "High" "Mid" "Mid" "High" ...
## $ Population     : int  39250017 19745289 27862596 20612439 12801539 12784227 11646273 10310371 101...
```

Spread pol_rating

```
pol_rating_wide <- pol_rating %>%
  spread(key = variable, value = value)
```

```
head(pol_rating_wide)
```

```
##   policy_number revenue      state      discipline year_started
## 1  C1AE00000009 3817115 New Hampshire Mechanical Engineering      2011
## 2  C1AE00000014  56137      Ohio      Architect      1996
## 3  C1AE00000040 160166 California Landscape Architecture      2004
## 4  C1AE00000044 863241 Tennessee Landscape Architecture      2001
## 5  C1AE00000046  56963 New Jersey      Architect      2008
## 6  C1AE00000056  75829 New Jersey      Architect      2011
##   employee_count use_written_contracts five_year_claims
## 1              20                Y                3
## 2               1                N                0
## 3               3                Y                0
## 4              11                Y                0
## 5               1                N                0
## 6               1                Y                0
```

```
str(pol_rating_wide)
```

```
## 'data.frame':   100000 obs. of  8 variables:
## $ policy_number      : chr  "C1AE00000009" "C1AE00000014" "C1AE00000040" "C1AE00000044" ...
## $ revenue            : chr  "3817115" "56137" "160166" "863241" ...
## $ state              : chr  "New Hampshire" "Ohio" "California" "Tennessee" ...
## $ discipline         : chr  "Mechanical Engineering" "Architect" "Landscape Architecture" "Landscape Architecture" ...
## $ year_started       : chr  "2011" "1996" "2004" "2001" ...
## $ employee_count     : chr  "20" "1" "3" "11" ...
## $ use_written_contracts: chr  "Y" "N" "Y" "Y" ...
## $ five_year_claims   : chr  "3" "0" "0" "0" ...
```

Join the new wide pol_rating object to pol_dates

```
pol <- pol_rating_wide %>% inner_join(pol_dates)
```

```
## Joining, by = "policy_number"
```

```
head(pol)
```

```
##   policy_number revenue      state      discipline year_started
## 1 C1AE00000009 3817115 New Hampshire Mechanical Engineering      2011
## 2 C1AE00000014  56137      Ohio      Architect      1996
## 3 C1AE00000040 160166 California Landscape Architecture      2004
## 4 C1AE00000044 863241 Tennessee Landscape Architecture      2001
## 5 C1AE00000046  56963 New Jersey      Architect      2008
## 6 C1AE00000056  75829 New Jersey      Architect      2011
##   employee_count use_written_contracts five_year_claims inception
## 1             20                Y              3      201212
## 2              1                N              0      201310
## 3              3                Y              0      201603
## 4             11                Y              0      201304
## 5              1                N              0      200911
## 6              1                Y              0      201208
##   expiration
## 1      201312
## 2      201410
## 3      201703
## 4      201404
## 5      201011
## 6      201308
```

```
str(pol)
```

```
## 'data.frame': 100000 obs. of 10 variables:
## $ policy_number : chr "C1AE00000009" "C1AE00000014" "C1AE00000040" "C1AE00000044" ...
## $ revenue : chr "3817115" "56137" "160166" "863241" ...
## $ state : chr "New Hampshire" "Ohio" "California" "Tennessee" ...
## $ discipline : chr "Mechanical Engineering" "Architect" "Landscape Architecture" "Landscape Architecture" ...
## $ year_started : chr "2011" "1996" "2004" "2001" ...
## $ employee_count : chr "20" "1" "3" "11" ...
## $ use_written_contracts: chr "Y" "N" "Y" "Y" ...
## $ five_year_claims : chr "3" "0" "0" "0" ...
## $ inception : chr "201212" "201310" "201603" "201304" ...
## $ expiration : chr "201312" "201410" "201703" "201404" ...
```

Join with the state lookup table

The data frames `state_lookup` and `pol` do **not** spell “state” the same. One has an uppercase ‘S’ the other lowercase. Since R is case-sensitive, this is a problem. This statement would give an error

```
# Gives an error!
```

```
pol_state <- pol %>% inner_join(state_lookup)
```

One option is to use the `by` arg of `inner_join`.

```
pol_state <- pol %>%
  inner_join(state_lookup, by = c("state" = "State"))
```

Another option is to rename the columns of `state_lookup` first. Then do the join. I like this one because I want to rename the other columns anyway.

```
names(state_lookup) <- c('state', 'state_group', 'state_population')
pol_state <- pol %>%
  inner_join(state_lookup)
```

```
## Joining, by = "state"
```

OK, let's do this a third time, because I actually don't want `state_population` in the joined table.

```
state_group_lookup <- state_lookup[, c('state', 'state_group')]
pol_state <- pol %>%
  inner_join(state_group_lookup)
```

```
## Joining, by = "state"
```

```
head(pol_state)
```

```
##   policy_number revenue      state      discipline year_started
## 1  C1AE00000009 3817115 New Hampshire Mechanical Engineering      2011
## 2  C1AE00000014  56137      Ohio      Architect      1996
## 3  C1AE00000040 160166 California Landscape Architecture      2004
## 4  C1AE00000044 863241 Tennessee Landscape Architecture      2001
## 5  C1AE00000046  56963 New Jersey      Architect      2008
## 6  C1AE00000056  75829 New Jersey      Architect      2011
##   employee_count use_written_contracts five_year_claims inception
## 1              20                  Y              3      201212
## 2               1                  N              0      201310
## 3               3                  Y              0      201603
## 4              11                  Y              0      201304
## 5               1                  N              0      200911
## 6               1                  Y              0      201208
##   expiration state_group
## 1      201312      Low
## 2      201410      Low
## 3      201703      High
## 4      201404      Low
## 5      201011      Mid
## 6      201308      Mid
```

```
str(pol_state)
```

```
## 'data.frame':  100000 obs. of  11 variables:
##  $ policy_number      : chr  "C1AE00000009" "C1AE00000014" "C1AE00000040" "C1AE00000044" ...
##  $ revenue            : chr  "3817115" "56137" "160166" "863241" ...
##  $ state              : chr  "New Hampshire" "Ohio" "California" "Tennessee" ...
##  $ discipline         : chr  "Mechanical Engineering" "Architect" "Landscape Architecture" "Landscape Architecture" ...
##  $ year_started       : chr  "2011" "1996" "2004" "2001" ...
##  $ employee_count     : chr  "20" "1" "3" "11" ...
##  $ use_written_contracts: chr  "Y" "N" "Y" "Y" ...
##  $ five_year_claims   : chr  "3" "0" "0" "0" ...
##  $ inception          : chr  "201212" "201310" "201603" "201304" ...
##  $ expiration         : chr  "201312" "201410" "201703" "201404" ...
```

```
## $ state_group      : chr "Low" "Low" "High" "Low" ...
```

Put rating characteristics back in claims table

Let's do this step before aggregating the claims data and adding to the policy data. The reason is that I don't need "total claim count" by policy added back to the claims data.

```
claims_final <- claims %>% inner_join(pol_state)
```

```
## Joining, by = "policy_number"
```

```
head(claims_final)
```

```
##   policy_number claim_ultimate claim_number revenue      state
## 1 C1AE00783351      22447.10 CR0080343074 3771609    California
## 2 C1AE00075999      18380.63 CR0010605425 399223    Massachusetts
## 3 C1AE00141200      141429.06 CR0034027774 3769935         Utah
## 4 C1AE00264573      18057.69 CR0050581498 1937566    California
## 5 C1AE00212315       85790.85 CR0090796329 2011799      Missouri
## 6 C1AE00212315      412740.42 CR0003635914 2011799      Missouri
##           discipline year_started employee_count use_written_contracts
## 1 Structural Engineer          1989             29                  Y
## 2 Structural Engineer          1989              8                  Y
## 3           Architect          1993             37                  Y
## 4 Structural Engineer          2010             15                  Y
## 5 Structural Engineer          2000             34                  Y
## 6 Structural Engineer          2000             34                  Y
##   five_year_claims inception expiration state_group
## 1                5    200802    200902         High
## 2                0    200909    201009         Mid
## 3                3    200810    200910         Low
## 4               10    201408    201508         High
## 5                1    201211    201311         Low
## 6                1    201211    201311         Low
```

```
str(claims_final)
```

```
## 'data.frame':   15010 obs. of  13 variables:
## $ policy_number      : chr "C1AE00783351" "C1AE00075999" "C1AE00141200" "C1AE00264573" ...
## $ claim_ultimate     : num 22447 18381 141429 18058 85791 ...
## $ claim_number       : chr "CR0080343074" "CR0010605425" "CR0034027774" "CR0050581498" ...
## $ revenue            : chr "3771609" "399223" "3769935" "1937566" ...
## $ state              : chr "California" "Massachusetts" "Utah" "California" ...
## $ discipline         : chr "Structural Engineer" "Structural Engineer" "Architect" "Structural E
## $ year_started       : chr "1989" "1989" "1993" "2010" ...
## $ employee_count     : chr "29" "8" "37" "15" ...
## $ use_written_contracts: chr "Y" "Y" "Y" "Y" ...
## $ five_year_claims   : chr "5" "0" "3" "10" ...
## $ inception          : chr "200802" "200909" "200810" "201408" ...
## $ expiration         : chr "200902" "201009" "200910" "201508" ...
## $ state_group        : chr "High" "Mid" "Low" "High" ...
```

Aggregate claims data by policy

Like everything in R there is more than one way to do this. Who knows which is better...

```
# One option for aggregating claims
claims_agg <- claims %>%
  group_by(policy_number) %>%
  summarize(
    total_ultimate = sum(claim_ultimate),
    claim_count = n())
```

```
# Another option for aggregating claims
```

```
claims$count <- 1
claims_agg <- claims %>%
  group_by(policy_number) %>%
  summarize(
    total_ultimate = sum(claim_ultimate),
    claim_count = sum(count))
```

```
head(claims_agg)
```

```
## # A tibble: 6 x 3
##   policy_number total_ultimate claim_count
##   <chr>          <dbl>          <dbl>
## 1 C1AE00000328      43815.            1
## 2 C1AE00000550      26234.            1
## 3 C1AE00000570     114088.            3
## 4 C1AE00000595     273950.            4
## 5 C1AE00000846       5293.            1
## 6 C1AE00001193     401259.            4
```

```
str(claims_agg)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':   8526 obs. of  3 variables:
##  $ policy_number : chr  "C1AE00000328" "C1AE00000550" "C1AE00000570" "C1AE00000595" ...
##  $ total_ultimate: num  43815 26234 114088 273950 5293 ...
##  $ claim_count   : num  1 1 3 4 1 4 2 2 1 1 ...
```

Join agg claims with policy data

Doing the join is easy. But because this is a left join, Any policies without claims will have NAs in the `total_ultimate` and `claim_count` columns. But, this will cause problems in modeling later, so we will change NAs in these columns to 0.

```
pol_final <- pol_state %>%
  left_join(claims_agg)
```

```
## Joining, by = "policy_number"
```

There are different ways to replace NAs with zeros. I'll use one for `total_ultimate`.

```
pol_final$total_ultimate[is.na(pol_final$total_ultimate)] <- 0
```

And, another for `claim_count`.

```
pol_final$claim_count <-
  ifelse(is.na(pol_final$claim_count), 0, pol_final$claim_count)
```

```
head(pol_final)
```

```
##   policy_number revenue      state      discipline year_started
## 1  C1AE00000009 3817115 New Hampshire Mechanical Engineering      2011
## 2  C1AE00000014  56137      Ohio      Architect      1996
## 3  C1AE00000040 160166 California Landscape Architecture      2004
## 4  C1AE00000044 863241 Tennessee Landscape Architecture      2001
## 5  C1AE00000046  56963 New Jersey      Architect      2008
## 6  C1AE00000056  75829 New Jersey      Architect      2011
##   employee_count use_written_contracts five_year_claims inception
## 1              20                  Y              3      201212
## 2              1                  N              0      201310
## 3              3                  Y              0      201603
## 4             11                  Y              0      201304
## 5              1                  N              0      200911
## 6              1                  Y              0      201208
##   expiration state_group total_ultimate claim_count
## 1      201312         Low              0           0
## 2      201410         Low              0           0
## 3      201703         High              0           0
## 4      201404         Low              0           0
## 5      201011         Mid              0           0
## 6      201308         Mid              0           0
```

```
str(pol_final)
```

```
## 'data.frame': 100000 obs. of 13 variables:
## $ policy_number : chr "C1AE00000009" "C1AE00000014" "C1AE00000040" "C1AE00000044" ...
## $ revenue : chr "3817115" "56137" "160166" "863241" ...
## $ state : chr "New Hampshire" "Ohio" "California" "Tennessee" ...
## $ discipline : chr "Mechanical Engineering" "Architect" "Landscape Architecture" "Landscape Architecture" ...
## $ year_started : chr "2011" "1996" "2004" "2001" ...
## $ employee_count : chr "20" "1" "3" "11" ...
## $ use_written_contracts: chr "Y" "N" "Y" "Y" ...
## $ five_year_claims : chr "3" "0" "0" "0" ...
## $ inception : chr "201212" "201310" "201603" "201304" ...
## $ expiration : chr "201312" "201410" "201703" "201404" ...
## $ state_group : chr "Low" "Low" "High" "Low" ...
## $ total_ultimate : num 0 0 0 0 0 0 0 0 0 0 ...
## $ claim_count : num 0 0 0 0 0 0 0 0 0 0 ...
```

Add some derived columns

```
# I want this claim in both!
pol_final$years_in_business <-
  year_YYYYMM(pol_final$inception) - as.integer(pol_final$year_started) + 1

claims_final$years_in_business <-
  year_YYYYMM(claims_final$inception) - as.integer(claims_final$year_started) + 1

pol_final$average_severity <- ifelse(
  pol_final$claim_count == 0,
  0,
```



```
pol_final$total_ultimate / pol_final$claim_count
)
```

```
head(pol_final)
```

```
## policy_number revenue state discipline year_started
## 1 C1AE00000009 3817115 New Hampshire Mechanical Engineering 2011
## 2 C1AE00000014 56137 Ohio Architect 1996
## 3 C1AE00000040 160166 California Landscape Architecture 2004
## 4 C1AE00000044 863241 Tennessee Landscape Architecture 2001
## 5 C1AE00000046 56963 New Jersey Architect 2008
## 6 C1AE00000056 75829 New Jersey Architect 2011
## employee_count use_written_contracts five_year_claims inception
## 1 20 Y 3 201212
## 2 1 N 0 201310
## 3 3 Y 0 201603
## 4 11 Y 0 201304
## 5 1 N 0 200911
## 6 1 Y 0 201208
## expiration state_group total_ultimate claim_count years_in_business
## 1 201312 Low 0 0 2
## 2 201410 Low 0 0 18
## 3 201703 High 0 0 13
## 4 201404 Low 0 0 13
## 5 201011 Mid 0 0 2
## 6 201308 Mid 0 0 2
## average_severity
## 1 0
## 2 0
## 3 0
## 4 0
## 5 0
## 6 0
```

```
head(claims_final)
```

```
## policy_number claim_ultimate claim_number revenue state
## 1 C1AE00783351 22447.10 CR0080343074 3771609 California
## 2 C1AE00075999 18380.63 CR0010605425 399223 Massachusetts
## 3 C1AE00141200 141429.06 CR0034027774 3769935 Utah
## 4 C1AE00264573 18057.69 CR0050581498 1937566 California
## 5 C1AE00212315 85790.85 CR0090796329 2011799 Missouri
## 6 C1AE00212315 412740.42 CR0003635914 2011799 Missouri
## discipline year_started employee_count use_written_contracts
## 1 Structural Engineer 1989 29 Y
## 2 Structural Engineer 1989 8 Y
## 3 Architect 1993 37 Y
## 4 Structural Engineer 2010 15 Y
## 5 Structural Engineer 2000 34 Y
## 6 Structural Engineer 2000 34 Y
## five_year_claims inception expiration state_group years_in_business
## 1 5 200802 200902 High 20
## 2 0 200909 201009 Mid 21
## 3 3 200810 200910 Low 16
```

## 4	10	201408	201508	High	5
## 5	1	201211	201311	Low	13
## 6	1	201211	201311	Low	13

Do some sense checking

You should always check that you didn't lose or make up any data especially when doing joins.

Next two statements need to output the same number.

```
sum(pol_final$claim_count)
```

```
## [1] 15010
```

```
nrow(claims)
```

```
## [1] 15010
```

Next three statements need to output the same number.

```
sum(pol_final$total_ultimate)
```

```
## [1] 1456021336
```

```
sum(claims$claim_ultimate)
```

```
## [1] 1456021336
```

```
sum(pol_final$average_severity * pol_final$claim_count)
```

```
## [1] 1456021336
```

Next three statements need to output the same number.

```
nrow(pol_final)
```

```
## [1] 100000
```

```
length(unique(pol_final$policy_number))
```

```
## [1] 100000
```

```
length(unique(pol$policy_number))
```

```
## [1] 100000
```

Save files

I'm going to be fancy, because I want a timestamp in the file name.

```
fname <- paste0(
  'c:/home/git/other/ratemaking-capstone/demo/data-',
  format(Sys.time(), '%Y-%m-%d-%H%M'),
  '.RData')
```

```
print(fname)
```

```
## [1] "c:/home/git/other/ratemaking-capstone/demo/data-2018-07-27-1316.RData"
```

```
save(pol_final, claims_final, file = fname)
```