R and the 3Rs
Using R to Explore Transitional Reinsurance
under the Affordable Care Act

Seth J. Chandler
Foundation Professor of Law
University of Houston Law Center
chandler@uh.ed

June 25, 2015

Introduction

The Affordable Care Act (ACA) is a landmark piece of legislation in the United States that significantly alters its health insurance market. The law, sometimes called Obama care, does not formally establish government funded healthcare but instead induces private insurers to sell insurance to individuals without the sort of traditional health underwriting that ordinarily occurs. To prevent the adverse selection death spiral that would otherwise likely overwhelm such a system, the ACA uses several tools. Among these are income-based subsidies to individuals so that more low risk individuals would purchase the policies even though they might, in an subsidized world, not be particularly good deals for them. Adverse selection does not occur to any great extent when insurance is a good deal for almost everyone. Other efforts to control adverse selection focused on subsidies to insurers. These provisions are generally called the "3 Rs" (transitional Reinsurance, Risk corridors and Risk Adjustment). This talk gives me time to speak about transitional reinsurance. I will leave risk corridors and risk adjustment to another day.

Transitional reinsurance is a program whereby the federal government provides free specific stop loss reinsurance to insurers selling policies in marketplaces called "exchanges" run by the states and federal government for the first three years of the program.[1] Promoters of this provision contended that the reinsur-

[1] Consistent with the United States traditional emphasis on state regulation of the insurance industry, as opposed to federal regulation, there are, in essence, one exchange for each of the 50 states. As a result of strident opposition to the ACA, some of the states have declined to establish the Exchanges and, consistent with a backstop provision in the law, the federal government has come in to run Exchanges in those non-cooperating states. The United States Supreme Court decided on June 25, 2015, in the case of *King v. Burwell* that it would interpret the ACA to require that all states would be treated the same for purposes of income-based subsidies regardless of whether they established an exchange themselves of left it to the federal government.

ance would induce insurers to enter a market in which the distribution of claims was not well known.[2]

To implement transitional reinsurance, the government sets an attachment point, maximum point and reimbursement rate for the reinsurance. Thus, if the attachment point were \$70,000 and the maximum point were \$250,000, and the reimbursement percentage was 80%, an insurer would be entitled to \$80,000 back if one of it's policy holders racked up a \$170,000 claim, meaning that the net cost of the claim would be \$90,000 rather than \$170,000. If the claim were \$400,000 the insurer's net cost would be \$256,000. If the claim was \$10,000 the net cost would be \$10,000. The ACA functionally requires that the value of the reinsurance policy diminish from 2014 to 2016 and to then be eliminated past 2016.[3]

There has been great concern in the United States in recent months about proposed rate hikes submitted by insurers who sell policies on the exchanges pursuant to the ACA. Some have identified the reduction in transitional reinsurance as the possible villain. To assess this claim requires that one compute the value of the transitional reinsurance. To undertake this computation, however, one needs data on the distribution or anticipated distribution of the dollar value of individual claims for policies purchased from the population seeking out insurance on the various Exchanges. Such data would enable one to compute the appropriate expectations.[4] Such data is notoriously hard to come by from private insurers. The federal government has, however, in order to determine whether the coverages provided by insurers selling policies on the Exchanges meet certain regulatory standards, published for 2014, 2015 and 2016, an "Actuarial Value Calculator." This "calculator" is actually a byzantine Excel spreadsheet with elaborate macros. [5] For my purposes, however, what is valuable about the Actuarial Value Calculator is that it contains fairly fine-grained distributions of claims expenses that are broken down by the type of claim (medical pharmaceutical or both) and by something known as the "metal level" of the policy. The metal level is determined by the ratio of the expected value of claims payments made by the insurer (after deductible, copays and coinsurance) to the expected value of claims. If that ratio is about 60% the policy is dubbed bronze whereas if the ratio is about 90%, the policy is dubbed "platinum." "Silver" and"Gold"

[2]Specific stop loss insurance, although well suited to subsidizing the insurance industry, is not optimally suited to reducing risk. Presumably what insurers care about is their aggregate claim experience, not the claims on any individual. It would thus have made more conceptual sense to offer aggregate stop loss insurance. Calibrating the parameters of aggregate stop loss reinsurance to each particular insurer would have been extremely difficult, however. The Risk Corridors program, one of the other 3Rs, takes on some this risk reduction role and has some resemblance to an aggregate reinsurance contract.

[3]See 42 U.S.C. §18061(b)(3)(B)(iii)

[4]I use "expectation" in its mathematical and statistical sense. Thus, for discrete distributions involving n possible events each of which has a probability of occurring of p_i and a value of x_i it is $\sum_{i=1}^{n} p_i x_i$. For continuous distributions it is $\int_{\mathcal{D}} p(x) x dx$ $\mathcal{D}$ is the domain of the probability distribution, p(x) is its density function, and x is the random variable.

[5] This marks, by the way, the first time I have seen regulation by Excel. Efforts through Freedom of Information Act requests to get an explanation of the code contained in the Excel macros have been unavailing.

lie between Bronze and Platinum. The calculator thus anticipates some degree of adverse selection as people with greater expected claims disproportionately opt for the Gold and Platinum policies (even though they are higher priced) whereas those with low expected claims opt disproportionately for Bronze and Silver).[6]

This paper shows how R can be used explore the extent to which anticipated changes in the value of the transitional reinsurance might contribute to both premium rate hikes and the value of these subsidies to insurers.[7] Much of the work involved using the *XLConnect* and *dplyr* packages to wrestle the data into a form convenient for my research. This research, in conjunction with work done in *Mathematica*, formed the basis for testimony before the United States Congress last week on Transitional Reinsurance and Risk Corridors. It is my hope that the importance of the topic on which the analysis was performed compensates for any lack of sophistication in what is essentially an project in database programming.

Using R to explore Transitional Reinsurance

Explaining the critical code

In this section I provide a fairly detailed description of how R was used to generate the relevant results. Those interested more in the results themselves may safely skip this section. Those fascinated by R, however, will want to scour each twist and turn of the code, searching for ways in which it could have been done more efficiently or expressively.

I begin by loading in the packages needed to extend R's base functionality and creating a larger-than-standard Java heap size for computations.

```
options( java.parameters = "-Xmx1g" )
library(XLConnect)

## Loading required package:  XLConnectJars
## XLConnect 0.2-11 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI, Apache Commons
##     Codec),
##   Stephen Colebourne [ctb, cph] (Joda-Time Java library)
```

[6]It is not clear to me whether the calculator also anticipates moral hazard and bakes that into its estimated distributions. People with low deductibles and copays may be more likely to consume medical care than people with high deductibles and copays even if the two groups have equal risk.

[7]It should be pointed out that these subsidies might instead have gone to insureds in the same income-sensitive manner as other subsidies. Such an approach might have reduced adverse selection and provided greater equity. On the other hand, the absence of specific stop loss reinsurance might have induced at least some insurers – probably the smaller ones – to attempt to purchase such reinsurance on the market and to thus place pressure on premiums.

```
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com

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

library(stringr)
library(tidyr)
library(xtable)
```

I now use the the *loadWorkbook* function of the *XLConnect* package to create three S4 objects that correspond with the Excel files containing the three Actuarial Value Calculators: one for 2014, 2015 and 2016 respectively.

```
#base_directory <- "/Users/sethchandler/Dropbox/Scholarship/Amsterdam15"
base_directory <- "/Users/sethchandler/Documents/Amsterdam temporary"
workbooks <- lapply(c("2014",
"2015", "2016"), function(year_string)
loadWorkbook(paste0(
base_directory,
"/avcalculator", year_string,
".xlsm"
)))
```

I will need to access a common set of multiple sheets within each of the Excel workbooks. The needed sheets are identifiable because they contain the string C̈ont. Tableïn their names. I use the Reduce functional construct within R to find the common set because the base *intersect* command in R does not accommodate more than two sets.The result is a vector of strings, relevant_sheet_names.

```
common_sheet_names <- Reduce(base::intersect,
 lapply(workbooks, getSheets))
relevant_sheet_names <- common_sheet_names[grepl(".+Cont. Table.+",
 common_sheet_names)]
```

I now process the sheets and produce a large data.frame. The code is a bit complex because I need to get 12 relevant sheets from each of three workbooks and because it is more efficient not to (re)load the workbook each time a sheet is accessed. We can look at the code from the inside out as follows. The innermost function just takes a workbook object and a relevant_sheet_name as the arguments to use the *readWorksheet* command from the *XLConnect* package to produce an R data.frame object that corresponds to the sheet.

```
inner_process<-function(wb,relevant_sheet_name) readWorksheet(wb,
    relevant_sheet_name,
    startRow = 4,
    endRow = 88,
    startCol = 1,
    endCol = 4)
```

Some of the claim distributions and, thus some of the sheets, will refer to all claims (combined), whereas others will refer only to what are known as Medical claims and still others will refer only to what are known as Pharmaceutical claims.[8] Some of the claim distributions, and, likewise, some of the sheets will refer only to claims associated with a specific metal level. I thus now form a function *inner_process_with_mutate* that performs the ïnner_processdescribed above but adds this additional information to each row in the resulting data.frame. For completeness and as an error check, I also include the sheet name from which the information was derived.

I thus now form a function *inner_process_with_mutate* that performs the ïnner_processdescribed above but adds this additional information to each row in the resulting data.frame.

```
mutation <- function(s,relevant_sheet_name)
  s %>%
  mutate(
  from_sheet = relevant_sheet_name,
  metal = gsub(" .*$",
  "", from_sheet),
  claim_type = str_replace(gsub("\\S+\\s+",
  "", from_sheet),
  "Only",
  "Rx")
  )

inner_process_with_mutate <- function(wb,relevant_sheet_name)
  inner_process(wb,relevant_sheet_name) %>% mutation(relevant_sheet_name)
```

[8]This paper will focus on the combined claims. To facilitate further research, however, and because the incremental coding costs of doing so are not too high, I also produce data on the other distributions contained in the Actuarial Value Calculators.

The exterior function takes a workbook and applies to each the inner_process_with mutate function for all of the relevant sheet names. I use this *ëxterior_process¨* function to create a list of dataframes each one corresponding to a year of data.

```
exterior_process <- function(wb,relevant_sheet_names)
  do.call(rbind,
  lapply(relevant_sheet_names,
  function(relevant_sheet_name)
  inner_process_with_mutate(wb,relevant_sheet_name)))

db_list <- lapply(workbooks,function(wb)
  exterior_process(wb,relevant_sheet_names))
```

Finally an additional rbind is called on the data.frames corresponding to each year (and each workbook) so that I end up with one big data.frame. The result is stored in a variable called {emphdb, a 3024x7 data.frame.

```
db <- do.call(rbind,db_list)
print(dim(db))
```

[1] 3024 7

There is now some clean up to do on the data.frame. The column names that *dplyr* infers from the Excel spreadsheet are unpleasant. Therefore, I adjust them. The b̈in_averagec̈olumn, which contains the average claim of someone in a particular claim bracket, unfortunately is imported as a string prefaced by a dollar sign. I convert these currency strings into numbers. To conserve memory, I factor the metal and claim_type columns. I also add a column showing the year associated with each row of data. As the Excel spreadsheets do not have year information conveniently accessible, I essentially add these by hand. The result is an 3024 x 8 data.frame

```
rm(db_list,workbooks)
colnames(db) <- c("bin", "count",
                  "max_d", "bin_average",
                  "from_sheet", "metal",
                  "claim_type")

db <- suppressWarnings(
    db %>%
        mutate(
            bin_average = as.numeric(gsub("[^0-9.]", "", bin_average)),
            metal = factor(metal,levels = c("Bronze",
                                             "Silver",
                                             "Gold",
                                             "Platinum")),
            claim_type = factor(claim_type,levels = c("Combined",
                                                          "Medical",
                                                          "Rx")),
             year = factor(c(
                 rep("2014",
                     1008), rep("2015", 1008),
                 rep("2016", 1008)
             ), levels = c("2014",
                           "2015", "2016"))
         )
)
db <-
    db %>%
    select(year,metal,claim_type,count,bin_average,bin,max_d,from_sheet)
samp <- sample_n(db,10) %>% select(-bin,-max_d,-from_sheet) %>%
    arrange(year,metal,claim_type)

db.table <- xtable(samp,floating = FALSE,digits = 2,
                   caption = 'Sample of rows in constructed dataframe')
print(db.table)
```

	year	metal	claim_type	count	bin_average
1	2014	Bronze	Combined	5180.00	1449.09
2	2014	Bronze	Rx	6609.00	648.27
3	2014	Silver	Rx	2414.00	2148.78
4	2014	Platinum	Rx	8272.00	2449.56
5	2015	Silver	Rx	4801.00	1248.91
6	2015	Gold	Combined	14489.00	37367.27
7	2015	Gold	Combined	30752.00	27328.88
8	2015	Platinum	Medical	6391.00	4349.48
9	2016	Silver	Combined	3.00	2759955.02
10	2016	Gold	Medical	10172.00	5350.00

Table 1: Sample of rows in constructed dataframe

At this point, enough work has been done that it makes sense to save the data in CSV and RData formats.

```
write.csv(db, paste0(base_directory,
 "/avcalculator2014-16.csv"))
saveRDS(db, file = paste0(base_directory,
 "/avcalculator2014-16.rdata"))
```

The Actuarial Value Calculator inconveniently represents the underlying claim distribution by using an absolute number of insureds for each bin rather than a fraction. To compute expectations, one needs fractions, not absolute numbers. I thus create a little database, *totals_db*, that calculates the total number of claimants for each of the 36 combinations of year, metal and claim type. This value will be used to normalize the data and determine the fraction of each year-metal-claim type grouping that falls into each bin.

```
totals_db <- db %>% group_by(year,
                             metal, claim_type) %>%
                             summarize(tot = sum(count))
print(dim(totals_db))

## [1] 36  4
```

One of the tasks of this research is to compare the expected net claims with the current amended transitional reinsurance program in place to what it would have been had there been no reinsurance. I also want to examine expected net claims under the original transitional reinsurance program in place to see what effect the recent amendment had.[9]

To compute the effect of reinsurance, I need a data.frame that specifies how the features of the transitional reinsurance program have changed over the three years 2014, 2015 and 2016. Because I will ultimately want to do some data.frame joins, I want to know the attachment point, maximum, and percentage for all combinations of claim types, years and metal levels.[10]

```
trp_parameters <- expand.grid(
  claim_type = c("Combined",
  "Medical", "Rx"), year = c("2014",
  "2015", "2016"), metal = c("Bronze",
  "Silver", "Gold", "Platinum")
  ) %>%
  mutate(
  attachment = sapply(year,
  function(year)
  switch(
  as.character(year),
  `2014` = 45000,
  `2015` = 70000,
  `2016` = 90000
  )),
```

[9]The Obama administration recently engaged in a retroactive modification of the reinsurance program for 2014, providing insurers with 100% rather than 80% reimbursement of claims between $45,000 and $250,000. This amendment was possible because enrollment in the Exchanges was considerably less than expected at the time the reinsurance parameters were formulated. Since a fixed budget was allocated to pay for transitional reinsurance, the smaller number of claimants permitted the federal government to give each insurer a higher reimbursement.

[10]I have a nagging sense that there is likely some more efficient way of performing this part of the computation. I welcome all suggestions.

```r
  max = sapply(year,
  function(year)
  switch(
  as.character(year),
  `2014` = 250000,
  `2015` = 250000,
  `2016` = 250000
  )),
  pct = sapply(year,
  function(year)
  switch(
  as.character(year),
  `2014` = 1,
  `2015` = 0.5,
  `2016` = 0.5
  ))
  ) %>%
  arrange(year)
```

I print out below a sample of the trp_parameters data.frame.

```r
samp_trp <- sample_n(trp_parameters,10) %>%
  arrange(year,metal,claim_type)
  print(
  xtable(
  samp_trp,floating = FALSE,digits = 2,
  caption = 'Sample of TRP Parameters (as amended)'
  )
  )
```

	claim_type	year	metal	attachment	max	pct
1	Medical	2014	Bronze	45000.00	250000.00	1.00
2	Combined	2014	Silver	45000.00	250000.00	1.00
3	Combined	2014	Gold	45000.00	250000.00	1.00
4	Rx	2015	Bronze	70000.00	250000.00	0.50
5	Combined	2015	Silver	70000.00	250000.00	0.50
6	Rx	2015	Silver	70000.00	250000.00	0.50
7	Rx	2015	Gold	70000.00	250000.00	0.50
8	Combined	2015	Platinum	70000.00	250000.00	0.50
9	Combined	2016	Silver	90000.00	250000.00	0.50
10	Medical	2016	Gold	90000.00	250000.00	0.50

Table 2: Sample of TRP Parameters (as amended)

I also want to compare the results of the current reinsurance program, which was amended just last week to (retroactively) be more generous to insurers, to the original version of the reinsurance program (*trp_original_parameters*), and to what the net claims of insurers would look like if there were no transitional reinsurance program (*trp_null_parameters*). These explorations are important to see the value of the g̈iftẗhe Obama administration bestowed on the insurance industry and to consider the effects on insurance pricing of the expiration of the transitional reinsurance program at the end of 2016.

```r
trp_original_parameters <- expand.grid(
  claim_type = c("Combined",
  "Medical", "Rx"), year = c("2014",
  "2015", "2016"), metal = c("Bronze",
  "Silver", "Gold", "Platinum")
  ) %>%
  mutate(
  attachment = sapply(year,
  function(year)
  switch(
  as.character(year),
  `2014` = 45000,
  `2015` = 70000,
  `2016` = 90000
  )),
  max = sapply(year,
  function(year)
  switch(
  as.character(year),
  `2014` = 250000,
  `2015` = 250000,
  `2016` = 250000
  )),
  pct = sapply(year,
  function(year)
  switch(
  as.character(year),
  `2014` = 0.8,
  `2015` = 0.5,
  `2016` = 0.5
  ))
  ) %>%
  arrange(year)

  trp_null_parameters <- expand.grid(
  claim_type = c("Combined",
  "Medical", "Rx"), year = c("2014",
  "2015", "2016"), metal = c("Bronze",
  "Silver", "Gold", "Platinum")
  ) %>%
  mutate(attachment = 0,
  max = 1e+08, pct = 0)
```

Again, I print out samples from each of these two dataframes.

```
samp_original <- sample_n(trp_original_parameters,10) %>%
arrange(year,metal,claim_type)
print(
  xtable(
  samp_original,floating = FALSE,digits = 2,
  caption = 'Sample of Original TRP Parameters '
  )
  )
```

	claim_type	year	metal	attachment	max	pct
1	Combined	2014	Bronze	45000.00	250000.00	0.80
2	Combined	2014	Silver	45000.00	250000.00	0.80
3	Rx	2014	Silver	45000.00	250000.00	0.80
4	Combined	2015	Bronze	70000.00	250000.00	0.50
5	Medical	2015	Bronze	70000.00	250000.00	0.50
6	Medical	2015	Silver	70000.00	250000.00	0.50
7	Rx	2015	Silver	70000.00	250000.00	0.50
8	Rx	2015	Platinum	70000.00	250000.00	0.50
9	Rx	2016	Silver	90000.00	250000.00	0.50
10	Rx	2016	Gold	90000.00	250000.00	0.50

Table 3: Sample of Original TRP Parameters

```
samp_null <- sample_n(trp_null_parameters,10) %>%
arrange(year,metal,claim_type)
print(
  xtable(
  samp_null,floating = FALSE,digits = 2,
  caption = 'Sample of TRP Parameters when there is no reinsurance '
  )
  )
```

An expectation is essentially a weighted sum. I thus need a function that determines the contribution to this sum for each bin of claims. I need to know the net contribution after reinsurance is taken into account. To do this, I create a function (*reins*) that looks at the fraction of insureds (*normalized_count*) with claims approximately equaling a bin average (*bin_average*) given three reinsurance parameters: an attachment point (*attachment*), a maximum amount (*max*)

	claim_type	year	metal	attachment	max	pct
1	Combined	2014	Silver	0.00	100000000.00	0.00
2	Combined	2014	Platinum	0.00	100000000.00	0.00
3	Rx	2015	Bronze	0.00	100000000.00	0.00
4	Combined	2015	Silver	0.00	100000000.00	0.00
5	Medical	2015	Silver	0.00	100000000.00	0.00
6	Combined	2015	Gold	0.00	100000000.00	0.00
7	Combined	2016	Bronze	0.00	100000000.00	0.00
8	Medical	2016	Bronze	0.00	100000000.00	0.00
9	Medical	2016	Gold	0.00	100000000.00	0.00
10	Rx	2016	Gold	0.00	100000000.00	0.00

Table 4: Sample of TRP Parameters when there is no reinsurance

and a reimbursement percentage (*pct*). The piecewise function underlying the computation is performed by finding the interval into which the bin average falls and then applying a different formula based on the applicable bin.[11]

```
reins <- function(normalized_count,
                  bin_average, attachment,
                  max, pct)
                  {
                  fi <- findInterval(bin_average,
                  c(-1, attachment,
                  max))
                  sw <- switch(
                  fi, bin_average,
                  attachment + (1 -
                  pct) * (bin_average -
                  attachment), attachment +
                  (1 - pct) * (max -
                  attachment) +
                  (bin_average -
                  max)
                  )

                  temp <- normalized_count *
                  sw
                  temp2 <- if (length(temp) ==
                  1)
                  temp
                  else
```

[11]I believe the R base language would benefit from having a construct, something like the Which statement in *Mathematica*, that makes multiple branching and description of piecewise functions a simpler proposition.

```
                    0
                    temp2

                    }
```

The central computation can now occur via a sequence of data.frame joins operations. I first left join the 3024 x 8 data.frame with the 36 x 4 data.frame containing the total number of claimants for each combination of year, metal level and claim type. The two data.frames have year, metal and claim_type in common, so they are used to perform the join. With this join performed, I can now calculate a normalized count, which gives us the frequency information we need for the *reins* function and that lets us compute the desired expectation. I now left join the resulting data.frame with the data.frame containing information on the reinsurance program. Again, since the two data.frames have year, metal and claim_type in common, these serve as the joining variables. I can now use *mapply* to calculate for each bin, for each year, for each metal level, and for each claim type, the contribution claims with the associated average claim (bin_average) contribute, after consideration of reinsurance, to the expected net claims of the insurer.

The final step is to compute the expected net claims of the insurer for each combination of year, metal, and claim type. This is done using the *group_by* and *summarize* functions of the *plyr* package. The whole computation can be wrapped in a function (*make_reins_db*) that takes as its arguments: (1) the data.frame containing information on bin averages and absolute counts, (2) the totals_db data.frame used to normalize the counts, and (3) the data.frame containing information on the applicable transitional reinsurance program.

```
make_reins_db <- function(db,
                          totals_db, trp)
                          left_join(
                          left_join(db,
                          totals_db, by = c("year",
                          "metal", "claim_type")) %>%
                          mutate(normalized_count = count / tot),
                          trp, by = c("year", "metal",
                          "claim_type")
                          ) %>%
                          mutate(
                          post_reinsurance_contribution = mapply(reins,
                          normalized_count,
                          bin_average, attachment,
                          max, pct)
                          ) %>% group_by(year,
                          metal, claim_type) %>%
                          summarize(claims_post_reinsurance =
                                    sum(post_reinsurance_contribution,
                               na.rm = TRUE))
```

Results concerning Transitional Reinsurance

It is now time to put this code to work.

Amended reinsurance vs. no reinsurance

I now use the *make_reins_db* function just created to compare the net expected costs of insurers with the current reinsurance parameters in place and without reinsurance. I do this for each combination of year and metal level[12] I then compute the ratio of the expected net costs with reinsurance to the expected net costs without reinsurance.[13]

```
  amended_v_none <- left_join(
    make_reins_db(db,
    totals_db, trp_parameters) %>%
    filter(claim_type == "Combined") %>%
    rename(Ec_post_reins = claims_post_reinsurance),
    make_reins_db(db, totals_db,
    trp_null_parameters) %>%
    filter(claim_type ==
    "Combined") %>%
    rename(Ec_without_reins = claims_post_reinsurance),
    by = c("year", "metal",
    "claim_type")
    ) %>%
    mutate(ratio = round(Ec_without_reins / Ec_post_reins,
    digits = 3)) %>%
    select(year,metal,Ec_post_reins,Ec_without_reins,ratio)
    amended_v_none_table <-
    xtable(
    amended_v_none,floating = FALSE,digits = 3,
    caption = 'Amended reinsurance versus no reinsurance'
    )
    print(amended_v_none_table)
```

[12]The comparison is done by joining two data.frames, the first created using the current reinsurance parameters and the second created using the reinsurance parameters that emulate the absence of any reinsurance. This latter data.frame has its column names slightly modified to make the comparison easier to read. The Ëcn̈otation (Ëxpected Claims)̈ is intended to evoke the idea that the associated data is a mathematical *expectation*.

[13]The *spread* command from the *tidyr* package is used to produce a more useful table.

	year	metal	Ec_post_reins	Ec_without_reins	ratio
1	2014	Bronze	3522.061	4064.374	1.154
2	2014	Silver	4095.713	4736.516	1.156
3	2014	Gold	4337.271	4995.660	1.152
4	2014	Platinum	5406.671	6153.633	1.138
5	2015	Bronze	3911.758	4064.374	1.039
6	2015	Silver	4556.695	4736.516	1.039
7	2015	Gold	4812.540	4995.660	1.038
8	2015	Platinum	5954.236	6153.633	1.033
9	2016	Bronze	4451.086	4602.961	1.034
10	2016	Silver	5204.569	5384.878	1.035
11	2016	Gold	5483.534	5666.202	1.033
12	2016	Platinum	6780.574	6979.604	1.029

Table 5: Amended reinsurance versus no reinsurance

```
spread_amended_v_none <-
        amended_v_none %>% spread(year,ratio)
        spread_amended_v_none_x <-
        xtable(
        spread_amended_v_none,floating = FALSE,
        digits = 3,
        caption = 'Ratio between amended reinsurance versus no reinsurance'
        )
        print(spread_amended_v_none_x)
```

	metal	Ec_post_reins	Ec_without_reins	2014	2015	2016
1	Bronze	3522.061	4064.374	1.154		
2	Bronze	3911.758	4064.374		1.039	
3	Bronze	4451.086	4602.961			1.034
4	Silver	4095.713	4736.516	1.156		
5	Silver	4556.695	4736.516		1.039	
6	Silver	5204.569	5384.878			1.035
7	Gold	4337.271	4995.660	1.152		
8	Gold	4812.540	4995.660		1.038	
9	Gold	5483.534	5666.202			1.033
10	Platinum	5406.671	6153.633	1.138		
11	Platinum	5954.236	6153.633		1.033	
12	Platinum	6780.574	6979.604			1.029

Table 6: Ratio between amended reinsurance versus no reinsurance

The results show that the subsidy was quite high in 2014, the first year of the program, for all metal levels, ranging from 13.8% for platinum to 15.4% for bronze. The subsidy dropped considerably in 2015, with value ranging from

3.3% to 3.9% for all metal levels. The subsidy does not drop much for 2016, ranging from 2.9% to 3.4%.[14] This is a crucial result. It is difficult to blame insurance premium hikes for 2016 on transitional reinsurance reductions when the reduction in subsidies is so small. The reason must be found elsewhere.

Amended reinsurance vs. original reinsurance

I can also compare expected net claims under the amended reinsurance parameters to the expected net claims under the original reinsurance parameters. I can do this for the various combinations of year and metal level. The results show that the amendment reduced insurer's net expected claims by about 3% for each of the four metal levels. In some sense, the federal government bestowed a 3% cash back rebate to insurers. This rebate is likely to be controversial given that some have complained that the subsidies provided to individuals to purchase policies, which are scaled to the purchaser's income, are too small. Also, it is unlikely that this rebate will be passed back to consumers in any direct way.[15]

```
amended_v_orig <- left_join(
  make_reins_db(db,
  totals_db, trp_parameters) %>%
  filter(claim_type == "Combined") %>%
  rename(Ec_post_reins = claims_post_reinsurance),
  make_reins_db(db, totals_db,
  trp_original_parameters) %>%
  filter(claim_type ==
  "Combined") %>%
  rename(Ec_orig_reins = claims_post_reinsurance),
  by = c("year", "metal",
  "claim_type")
  ) %>%
  mutate(ratio = round(Ec_orig_reins / Ec_post_reins,
  digits = 3)) %>%
  select(year,metal,Ec_post_reins,Ec_orig_reins,ratio)
```

[14]There is a bit of a puzzle in this result. The amount allocated for the transitional reinsurance program goes from \$12 billion in 2014 to \$8 billion in 2015 to \$5 billion in 2016. Thus, the drop in the subsidy between 2014 and 2015 appears to be too big and the drop in the subsidy between 2015 and 2016 appears to be to small. This is particularly so since the number of people enrolled in Exchange policies is expected to grow and since the bill for transitional reinsurance should scale roughly in a linear way with the size of the insurance pool. Thus, there appears to be an unresolved inconsistency between the continuance tables of the Actuarial Value Calculator and the reinsurance parameters.

[15]A complication with the ACA is the difficulty insurers have in making up in future years for losses suffered in previous years. A provision of the ACA states that in any given year, the ratio between, essentially, claims and premiums can not be less than 0.8. 42 U.S.C. §300-gg18(b)(1). If the insurer violates this provision, they have go give premiums back to customers until the ratio is satisfied. Thus one can not charge extra high premiums in a particular year to compensate for prior losses. I am indebted for this point to Al Redmer, Commissioner of the Maryland Insurance Administration.

```
      amended_v_orig_table <-
      xtable(
      amended_v_orig,floating = FALSE,digits = 3,
      caption = 'Amended reinsurance versus original reinsurance'
      )
      print(amended_v_orig_table)
```

	year	metal	Ec_post_reins	Ec_orig_reins	ratio
1	2014	Bronze	3522.061	3630.523	1.031
2	2014	Silver	4095.713	4223.873	1.031
3	2014	Gold	4337.271	4468.949	1.030
4	2014	Platinum	5406.671	5556.064	1.028
5	2015	Bronze	3911.758	3911.758	1.000
6	2015	Silver	4556.695	4556.695	1.000
7	2015	Gold	4812.540	4812.540	1.000
8	2015	Platinum	5954.236	5954.236	1.000
9	2016	Bronze	4451.086	4451.086	1.000
10	2016	Silver	5204.569	5204.569	1.000
11	2016	Gold	5483.534	5483.534	1.000
12	2016	Platinum	6780.574	6780.574	1.000

Table 7: Amended reinsurance versus original reinsurance

```
 spread_amended_v_orig <-
   amended_v_orig %>% spread(year,ratio)
   spread_amended_v_orig_x <-
   xtable(
   spread_amended_v_orig,floating = FALSE,digits = 3,
   caption = 'Ratio between amended reinsurance versus original reinsurance'
   )
   print(spread_amended_v_orig_x)
```

Changes in expected claims between 2015 and 2016

If one stares at the data for a bit, one can see the likely culprit in the spate of premium hikes many fear will hit Obamacare consumers in the year ahead. The table below computes the pre-reinsurance expected claims for 2015 and for 2016 for each of the metal levels.

```
   ec_by_year <- make_reins_db(db, totals_db,
    trp_null_parameters) %>%
    filter(claim_type ==
                            "Combined",year %in% c("2014","2015","2016")) %>%
```

	metal	Ec_post_reins	Ec_orig_reins	2014	2015	2016
1	Bronze	3522.061	3630.523	1.031		
2	Bronze	3911.758	3911.758		1.000	
3	Bronze	4451.086	4451.086			1.000
4	Silver	4095.713	4223.873	1.031		
5	Silver	4556.695	4556.695		1.000	
6	Silver	5204.569	5204.569			1.000
7	Gold	4337.271	4468.949	1.030		
8	Gold	4812.540	4812.540		1.000	
9	Gold	5483.534	5483.534			1.000
10	Platinum	5406.671	5556.064	1.028		
11	Platinum	5954.236	5954.236		1.000	
12	Platinum	6780.574	6780.574			1.000

Table 8: Ratio between amended reinsurance versus original reinsurance

```
                                rename(Ec_without_reins = claims_post_reinsurance) %>%
                                select(-claim_type) %>%
                                spread(year,Ec_without_reins)
ec_by_year_table <- xtable(ec_by_year,floating = FALSE,digits = 2,
caption = 'Expected claims without reinsurance by year')
print(ec_by_year_table)
```

	metal	2014	2015	2016
1	Bronze	4064.37	4064.37	4602.96
2	Silver	4736.52	4736.52	5384.88
3	Gold	4995.66	4995.66	5666.20
4	Platinum	6153.63	6153.63	6979.60

Table 9: Expected claims without reinsurance by year

The results are quite striking. The continuance tables in the actuarial value tables were apparently unchanged between 2014 and 2015, possibly because not enough data had been produced, but once the data was in, the expected value of claims rose sharply between 2015 and 2016. I can sharpen this analysis by looking at the ratio of expected claims for 2016 to expected claims for 2015. We can now see that . It is not clear whether these double-digit changes really took place between 2014 and 2016, with the 2015 numbers being wrong because the regulatory authority did not have the data or whether most if not all of the increase took place between 2015 and 2016. Either way, however, the table shows clearly that adverse claims experience is likely responsible for the majority of most large premium increases requested by insurers.

```
temp <- make_reins_db(db, totals_db,
                      trp_null_parameters) %>%
                      filter(claim_type ==
                      "Combined",year %in% c("2015","2016")) %>%
                      rename(Ec_without_reins = claims_post_reinsurance) %>%
                      select(-claim_type) %>%
                      spread(year,Ec_without_reins)
                      colnames(temp) <-
                      c("metal","Y2015","Y2016")
y2015v2016 <- temp %>% mutate(pct_change = Y2016 / Y2015 - 1)
y2015v2016_table <-
  xtable(y2015v2016,floating = FALSE,digits = 3,
         caption = 'Changes in expected claims between 2015 and 2016')
print(y2015v2016_table)
```

	metal	Y2015	Y2016	pct_change
1	Bronze	4064.374	4602.961	0.133
2	Silver	4736.516	5384.878	0.137
3	Gold	4995.660	5666.202	0.134
4	Platinum	6153.633	6979.604	0.134

Table 10: Changes in expected claims between 2015 and 2016

Conclusion

I have used database commands within R and the *dplyr* package to take a grouping of multi-sheet Excel spreadsheets created by the United States government to regulate insurers and produce an R dataframe useful for more detailed examination of government programs and the distribution of healthcare claims. The work done here indicates that while the phase out of transitional reinsurance might be responsible for smaller premium increases, to the extent that large premium increases are observed between 2015 and 2016, that phenomenon is caused primarily by other factors. If the Actuarial Value Calculator is an accurate projection – and, it should be given that it is central to the regulation of insurance under the ACA – then, as shown in this paper, there is an obvious basis for higher premiums: increased morbidity of the insured population. This latter phenomenon may be due in part to moral hazard, in part to a failure of effective utilization review,[16] and in part due to inadequate control of adverse selection. Regardless, given the extraordinarily prices already paid by Americans in a system in which public and private insurance mediates between them

[16] "Utilization review" in American healthcare jargon for procedures that attempt to curb use of medical services, such as making sure the services are medically necessary, that they are covered by the insurance contract, and that the practitioner is charging lawful prices for work actually done.

and healthcare providers, premium and healthcare cost increases on the order of those suggested by the data here are deeply troubling.