Economic Impacts Assessment

Scott Coffin, Ph.D.

1/13/2021

Table of Contents

[Introduction 1](#_Toc61590899)

[How to Read this Report? 2](#_Toc61590900)

[Survey Design 2](#_Toc61590901)

[Survey Completeness 2](#_Toc61590902)

[Multiple Imputation 10](#_Toc61590903)

[Missing Data Assumptions 11](#_Toc61590904)

[<10,000 Service Connection Imputation 12](#_Toc61590905)

[Larges Imputation 35](#_Toc61590906)

[Survey Weighting 54](#_Toc61590907)

[Base Weights 54](#_Toc61590908)

[Small Systems 54](#_Toc61590909)

[Large Systems 56](#_Toc61590910)

[Response Propensity Adjustment 57](#_Toc61590911)

[Small Systems 57](#_Toc61590912)

[Large Systems 89](#_Toc61590913)

[Weight Calibration 97](#_Toc61590914)

[Evaluation of Correlative factors 98](#_Toc61590915)

[Calibration 108](#_Toc61590916)

[Summary Statistics 133](#_Toc61590917)

[Small Systems (>10,000 Service Connections) 133](#_Toc61590918)

[Risk 135](#_Toc61590919)

[References 153](#_Toc61590920)

# Introduction

A sample of public water systems were surveyed in late 2020 to determine economic impacts due to the Governor’s COVID-19 water shutoff order. Different surveys (questions and survey design) were completed for systems with less than 10,000 service connections and greater than 10,000 service connections, referred to in this report as “small” and “large” systems, respectively. This report details the procedures to adjust the weights of the sampled population to be representative of the whole population. This report also includes informaiton for the imputation of missing data (both response and auxiliary variables), and some statewide projections based on weighted data.

This report was generated using *R* version 4.02, Rmarkdown, and the following packages: tidyverse, BAMMtools,cowplot,rstatix,ggpubr,grid,gridExtra,calecopal,readxl,survey,RCurl,mitools,mice,foreign,magrittr,sampling,DT,lattice,PracTools,RColorBrewer,pheatmap,aod,finalfit,NonProbEst, matchins ,psych, scatterplot, ,car,prettydoc,viridis. Bethany Robinson, Ph.D. and Marielle Pinheiro, Ph.D. asssisted with survey design, and data cleanup.

# How to Read this Report?

This is an interactive HTML document that allows for continuous scrolling.

# Survey Design

Two distinct surveys were carried out to assess economic impacts to water systems in California in response to the COVID-19 pandemic. Water systems serving less than 10,000 service connections were binned according to service connection using the Jenk’s Natural Breaks method to reduce variance. Within these four discrete strata, water systems were randomly sampled. Systems with greater than 10,000 service connections were surveyed without strata, and with distinct questions. The sampling design strata are annotated in the below histogram.

Distribution of <10,000 service connection water systems by service connections, annotated with natural breaks

Distribution of <10,000 service connection water systems by service connections, annotated with natural breaks

# Survey Completeness

Out of the list of 555 water systems randomly selected from the population of 2,395 total systems serving fewer than 10,000 service connections, a total of 416 systems responded. The proportions of responses, annotated by sampling strata (bin) and system fee code are displayed below.

Proportion of water systems (<10,000 service connections) that completed the survey annotated by sampling bin and fee code

Proportion of water systems (<10,000 service connections) that completed the survey annotated by sampling bin and fee code

## [1] 0.4137931

Not all questions were answered by respondents. If only complete cases are used, just 41.4% of survey samples would be retained. If we were to simply exclude incomplete cases, we would trim a substantive poriton of the dataset, potentially reducing the power of the study to unnacceptable levels.

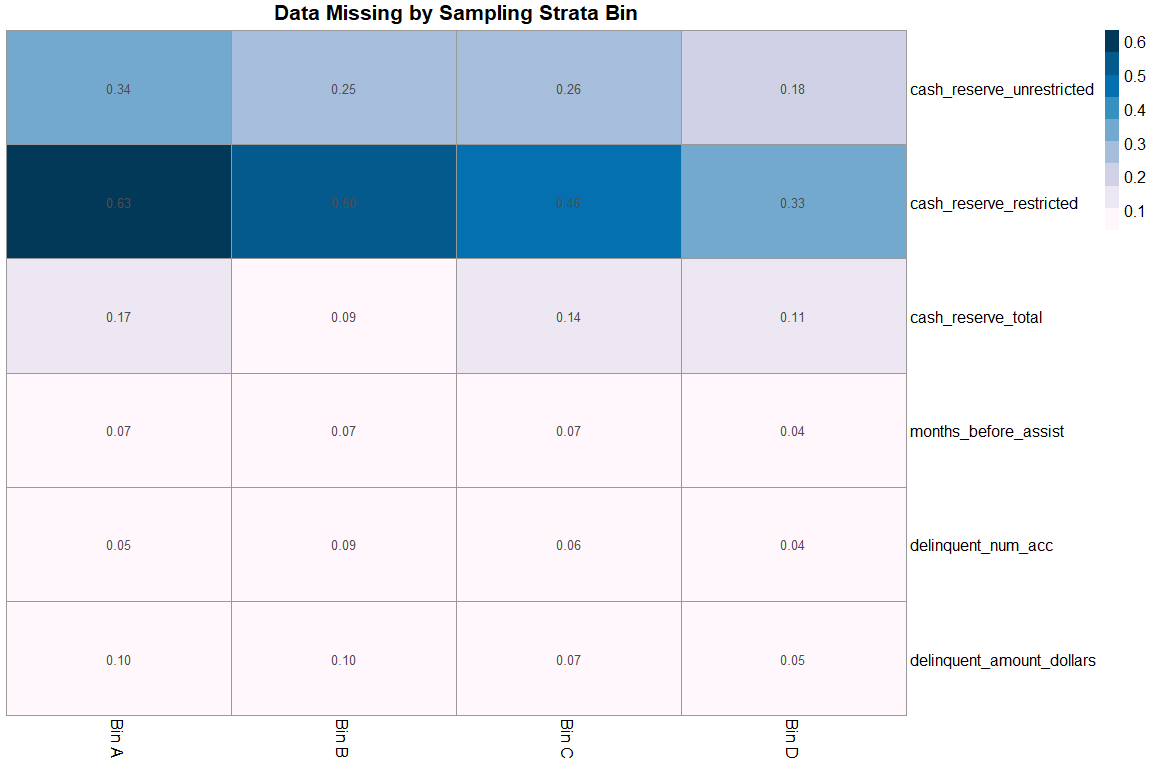
Since data are missing for some categories within responses, we have three choices: 1) listwise-deletion: remove rows that contain missing data. This will, of course, reduce the strength of the dataset. 2) mean/median substitution: another quick fix that takes the mean/median of the existing data points and substitutes the missing data points. This would obviously bias the analysis since it decreases variance. 3) Multiple imputation: With this approach, rather than replacing missing values with a single value, we use the distribution of the observed data/variables to estimate multiple possible values for the data points. This allows us to account for the uncertainty around the true value, and obtain approximately unbiased estimates (under certain conditions). Moreover, accounting for uncertainty allows us to calculate standard errors around estimations, which in turn leads to a better sense of uncertainty for the analysis.

The heatmap below shows if there were differences in response completeness by sampling bin.

## # A tibble: 4 x 7  
## tag cash\_reserve\_unrestricted cash\_reserve\_restricted cash\_reserve\_total  
## <fct> <dbl> <dbl> <dbl>  
## 1 Bin A 0.34 0.63 0.17  
## 2 Bin B 0.25 0.5 0.09  
## 3 Bin C 0.26 0.46 0.14  
## 4 Bin D 0.18 0.33 0.11  
## months\_before\_assist delinquent\_num\_acc delinquent\_amount\_dollars  
## <dbl> <dbl> <dbl>  
## 1 0.07 0.05 0.1   
## 2 0.07 0.09 0.1   
## 3 0.07 0.06 0.07  
## 4 0.04 0.04 0.05

Again, these data may be easier to view as a heatmap.

#convert to matrix and transpose  
transposedTag <- as.data.frame(t(as.matrix(tagCompletenessSummary[2:7]))) #1 is category, 2-6 are variables  
#reassign column names  
colnames(transposedTag) <- c("Bin A", "Bin B", "Bin C", "Bin D")  
#transposedTag  
#format as matrix  
MissingMatrixTag <- data.matrix(transposedTag)  
#build heatmap  
pheatmap(MissingMatrixTag,  
 main = "Data Missing by Sampling Strata Bin", #title  
 fontsize = 12,  
 cluster\_rows = FALSE, cluster\_cols = FALSE,#disable dendrograms  
 display\_numbers = TRUE,  
 treeheight\_row = 0, treeheight\_col = 0, #keeps clustering after dropping dendrograms  
 col = brewer.pal(n = 9, name = "PuBu")) #blue color scheme with 9 colors)

 There seems to be a clear step-wise trend for the willingness to report unresicted cash reserve by size (Bin A are smallest, Bin D are largest). This tracks with the above statement regarding District Staff reponses from surveys. Furthermore, unrestricted cash reserves are not applicable to small systems that do not have restrictions, so these missing values are not an issue. Luckily, no other obvious trends for missing response values appear to be present. Let’s see if there’s a relationship between reporting of this value and service connections.

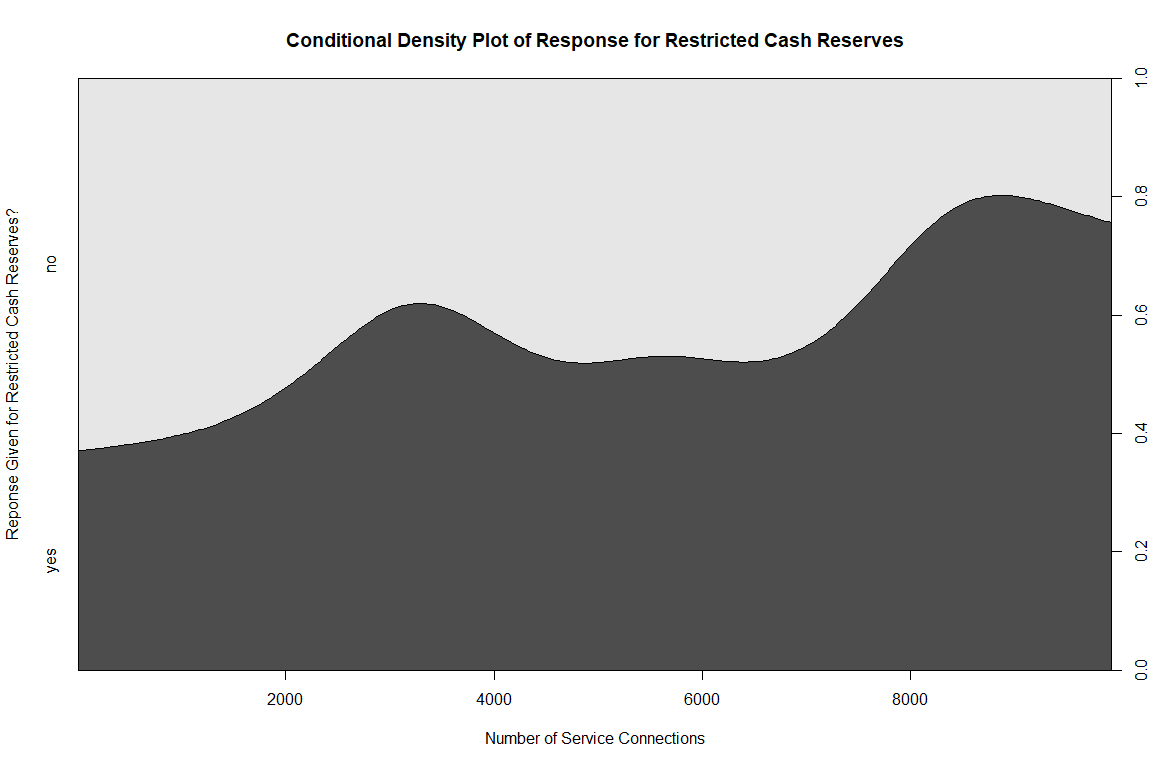
rm(MissingMatrixTag, tagCompletenessSummary, completenessSummary, transposed, transposedTag)  
#convert numeric to 1  
allSmalls.responded <- allSmalls.responded %>%   
 mutate(cash\_reserve\_restricted\_NA = case\_when(cash\_reserve\_restricted >= 0 ~ 1))  
  
#Convert NA to 0  
allSmalls.responded$cash\_reserve\_restricted\_NA <- allSmalls.responded$cash\_reserve\_restricted\_NA %>% replace\_na(0) %>% as.factor()  
#convert to factor  
allSmalls.responded <- allSmalls.responded %>%   
 mutate(cash\_reserve\_restricted\_NA\_yn = case\_when(cash\_reserve\_restricted\_NA == "0" ~ "no",  
 cash\_reserve\_restricted\_NA == "1" ~ "yes"))   
allSmalls.responded$cash\_reserve\_restricted\_NA\_yn <- as.factor(allSmalls.responded$cash\_reserve\_restricted\_NA\_yn)  
  
#examine relationship of response to restricted reseved cash question with number of service connections  
  
lgit<- glm(cash\_reserve\_restricted\_NA ~ Service\_Connections + Fee\_Code,   
 data = allSmalls.responded, family = "binomial")  
summary(lgit)

##   
## Call:  
## glm(formula = cash\_reserve\_restricted\_NA ~ Service\_Connections +   
## Fee\_Code, family = "binomial", data = allSmalls.responded)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8538 -0.9708 -0.8431 1.2038 1.6242   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.90728038 0.30915646 -2.935 0.003339 \*\*   
## Service\_Connections 0.00018978 0.00005792 3.277 0.001051 \*\*   
## Fee\_CodeDAVCL 1.28236170 0.33537707 3.824 0.000131 \*\*\*  
## Fee\_CodeDAVCS -0.12001029 0.60124208 -0.200 0.841790   
## Fee\_CodeSC 0.37875059 0.34164433 1.109 0.267598   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 575.53 on 415 degrees of freedom  
## Residual deviance: 537.17 on 411 degrees of freedom  
## AIC: 547.17  
##   
## Number of Fisher Scoring iterations: 4

Here we can see that a significant relationship exists between number of service connections and the response level for the question regarding restricted cash reserves. There also seems to be a significant relationship between the disadvantaged community larges, which is related to service connections.

This may also be visualized in a conditional density plot.

cdplot(cash\_reserve\_restricted\_NA\_yn ~ Service\_Connections, data = allSmalls.responded,  
 main = "Conditional Density Plot of Response for Restricted Cash Reserves", xlab = "Number of Service Connections", ylab = "Reponse Given for Restricted Cash Reserves?")



Let’s further test these grouping differences in response with a chi-squared test.

wald.test(b = coef(lgit), #coefficients from logit model   
 Sigma = vcov(lgit), #variance covariance matrix of the error terms  
 Terms = 3:5) #categorical terms (fee codes)

## Wald test:  
## ----------  
##   
## Chi-squared test:  
## X2 = 15.9, df = 3, P(> X2) = 0.0012

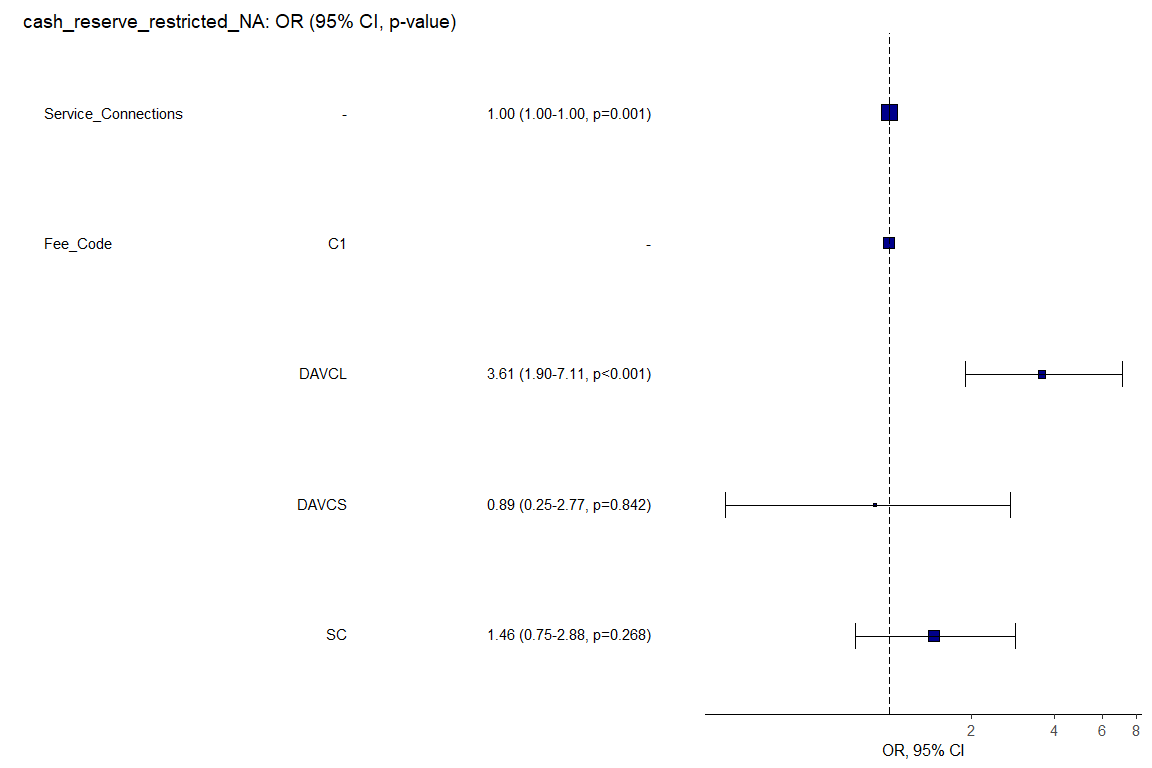
The p-value of 0.0035 again suggests that fee code is a significant predictor for willingness to respond to this question. An easier way to conceptualize this may be through odds-ratios.

#exponentiate the coefficients (i.e. create odds-ratios)  
round(exp(cbind(OddsRatio = coef(lgit), confint(lgit))),2)

## OddsRatio 2.5 % 97.5 %  
## (Intercept) 0.40 0.22 0.73  
## Service\_Connections 1.00 1.00 1.00  
## Fee\_CodeDAVCL 3.61 1.90 7.11  
## Fee\_CodeDAVCS 0.89 0.25 2.77  
## Fee\_CodeSC 1.46 0.75 2.88

Now we can clearly see that disadvantaged community Larges are 3.47 times more likely to report their unrestricted cash reserves (97.5% CI: 1.83 - 6.84). Odds ratios are plotted below:

#remove model   
rm(lgit)  
#define explanatory variables  
explanatory = c("Service\_Connections", "Fee\_Code")  
#plot ORs  
allSmalls.responded %>%   
 or\_plot("cash\_reserve\_restricted\_NA",  
 explanatory, table\_text\_size=4, title\_text\_size=14,   
 plot\_opts=list(xlab("OR, 95% CI"), theme(axis.title = element\_text(size=12))))

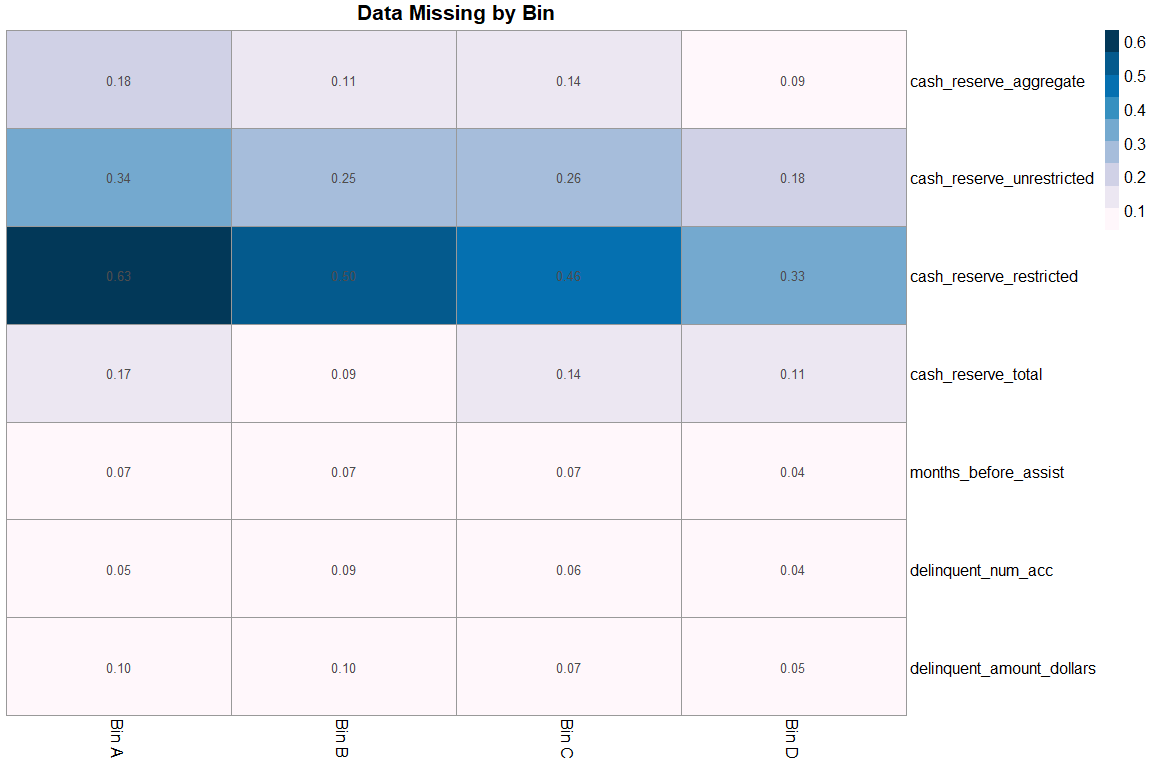


Since we’ve determined (both statistically, and qualitatively from District Staff responses) that the reporting of restricted/unrestricted cash reserves is dependent on accounting systems that are reflective of the system’s relative size, a reliable assumption may be made that total cash reserves can be expressed as unrestricted cash reserves for systems that only report totals and no restriction status. This simple conversion will allow for meaningful comparisons and reduce the amount of missing data while retaining high confidence by avoiding the need for sophisticated approaches (e.g. imputation) that would introduce additional uncertainty.

A simple logical test is used to create a new variable called “cash\_reserve\_aggregate”, which is equivalent to total cash reserves if no information is provided for restricted cash reserves, or unrestricted cash reserves if information is provided for restricted cash reserves.

The filling in of missing data from the above conversion may be visualized in the heatmap below.

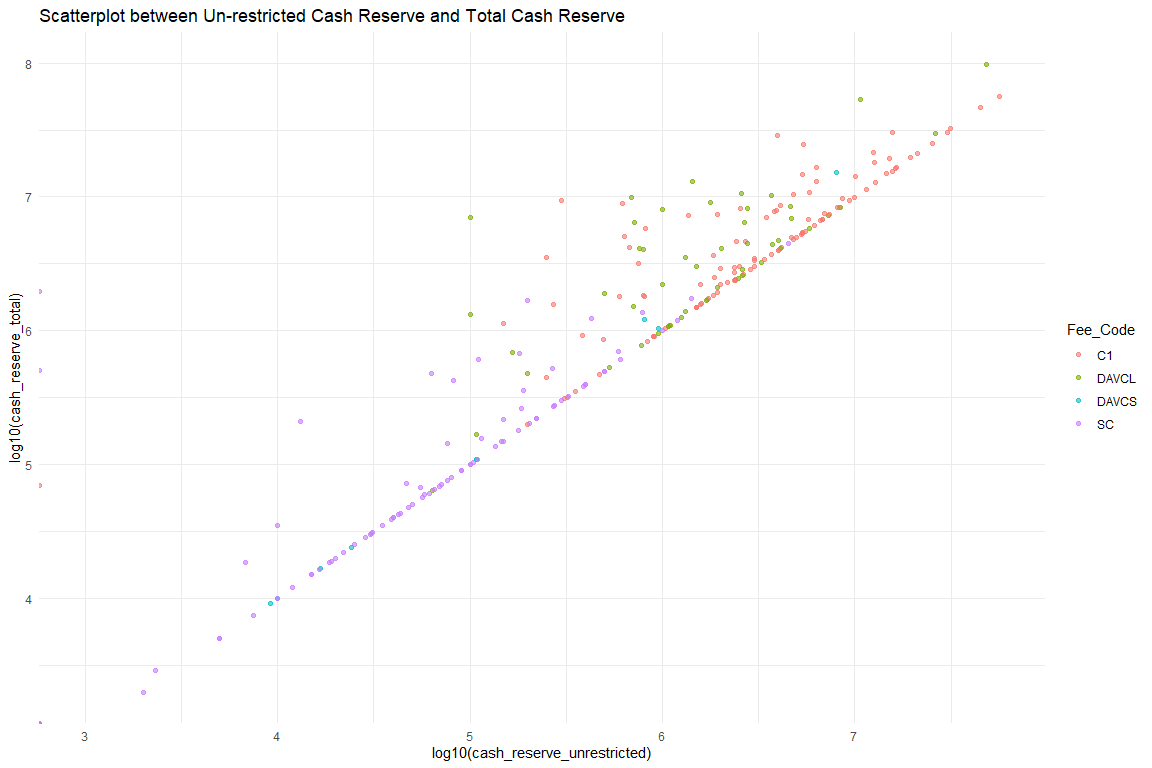
#create table  
tagCompletenessSummary <- allSmalls.responded %>%   
 select(tag, cash\_reserve\_aggregate, cash\_reserve\_unrestricted, cash\_reserve\_restricted, cash\_reserve\_restricted, cash\_reserve\_total, months\_before\_assist, delinquent\_num\_acc, delinquent\_amount\_dollars) %>%  
 group\_by(tag) %>%  
 summarise\_all((name = ~sum(is.na(.))/length(.))) %>%   
 mutate(across(is.numeric,~round(., 2)))   
  
#convert to matrix and transpose  
transposedTag <- as.data.frame(t(as.matrix(tagCompletenessSummary[2:8]))) #1 is category, 2-6 are variables  
#reassign column names  
colnames(transposedTag) <- c("Bin A", "Bin B", "Bin C", "Bin D")  
#transposedTag  
#format as matrix  
MissingMatrixTag <- data.matrix(transposedTag)  
#build heatmap  
pheatmap(MissingMatrixTag,  
 main = "Data Missing by Bin", #title  
 fontsize = 12,  
 cluster\_rows = FALSE, cluster\_cols = FALSE,#disable dendrograms  
 display\_numbers = TRUE,  
 treeheight\_row = 0, treeheight\_col = 0, #keeps clustering after dropping dendrograms  
 col = brewer.pal(n = 9, name = "PuBu")) #blue color scheme with 9 colors)



In the above heatmap, it can be seen that the the newly created variable “cash\_reserve\_aggregate” has virtually the same amount of missing data as the total cash reserve variable, thus confirming that the conversion process was successful.

While data is missing for unrestricted cash reserves, it may be possible to estimate this variable from other available response data. For example, it’s likely that there is a relationship between unrestricted cash reserve and total cash reserve This may be easily visualized in a dot plot.

allSmalls.requested.responded %>%   
 ggplot(aes(x = log10(cash\_reserve\_unrestricted), y = log10(cash\_reserve\_total), color = Fee\_Code)) +   
 geom\_point(alpha = 0.6) +  
 labs(title = "Scatterplot between Un-restricted Cash Reserve and Total Cash Reserve") +  
 theme\_minimal()



We can see discrete grouping of cash reserve (total and unrestricted) by fee code with community larges (C1) and disadvantaged community larges have higher cash reserves, while small community (SC) and disadvantaged small communities have lower cash reserves. There also seems to be a clear relationship between these variables, with some outliers. Independent linear regression models may be used to predict missing data, however more sophisticated methods are likely necessary to reliably predict outliers and impute missing data in a reliable manner.

Since there are many missing data, we cannot simply rely on single models for each variable. It is therefore necessary to perform multiple imputation to preserve our dataset. The mice *R* package is used to perform this imputation (which stands for Multivariate Imputation by Chained Equations), developed by Stef van Buuren. The basic steps of multiple imputation employed herein are described by Rubin (1976).

# Multiple Imputation

The basic steps for imputation of missing data are described below (Katitas, 2019).

1. impute the missing values by using an appropriate model which incorporates random variation.
2. repeat the first step 3-5 times.
3. perform the desired analysis on each data set by using standard, complete data methods.
4. average the values of the parameter estimates across the missing value samples in order to obtain a single point estimate.
5. calculate the standard errors by averaging the squared standard errors of the missing value estimates. After this, calculate the variance of the missing value parameter across the samples. Finally, combine the two quantities in multiple imputation for missing data to calculate the standard errors.

More simply, the steps of multiple imputation can be described as follows: a) choose values that keep the relationship in the dataset intact in place of missing values; b) create independently drawn imputed datasets; c) calculate new standard errors using variation across datasets to take into account the uncertainty created by these imputed datasets (Kropko et al. 2014).

## Missing Data Assumptions

Data missingness can be described in three distinct ways (Rubin 1976):

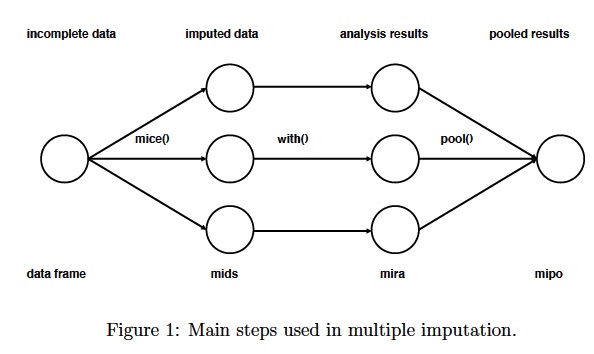
Missing Completely at Random (MCAR) – missingness of data points are random, meaning that the pattern of missing values is uncorrelated with the structure of the data.

Missing at Random (MAR) – missingness is not completely random, however the propensity of missingness depends on the observed data, not the missing data. In this case, the missing value can be predicted by looking at the answers for the personal information question.

Missing Not at Random (MNAR) – missingness is not random, and correlates with unobservable characteristics.

Like all multiple imputation techniques, this effort will start with the MAR assumption. While MCAR is desirable, in general it is unrealistic for the data. Thus, it may be necessary to assume that missing values can be replaced by predictions derived by the observable portion of the dataset. This fundamental assumption allows for the plausible prediction of missing values.

Conditional Multiple Imputation is utilized to impute missing data from this survey. This approach follows an iterative procedure, modeling the conditional distribution of a certain variable given the other variables. This technique allows for enhanced flexibility as a distribution is assumed for each variable rather than the whole dataset.



Three main steps to impute data. Source: University of Virgina Library

## <10,000 Service Connection Imputation

First, an Ordinary Least Squares (OLS) linear regression is used to predict cash in reserve (total) based on relevant predictors for all systems (e.g. service connections, median 12 month household income, delinquent number of accounts, delinquent amout in dollars, CalEnvrioScreen 3 scores).

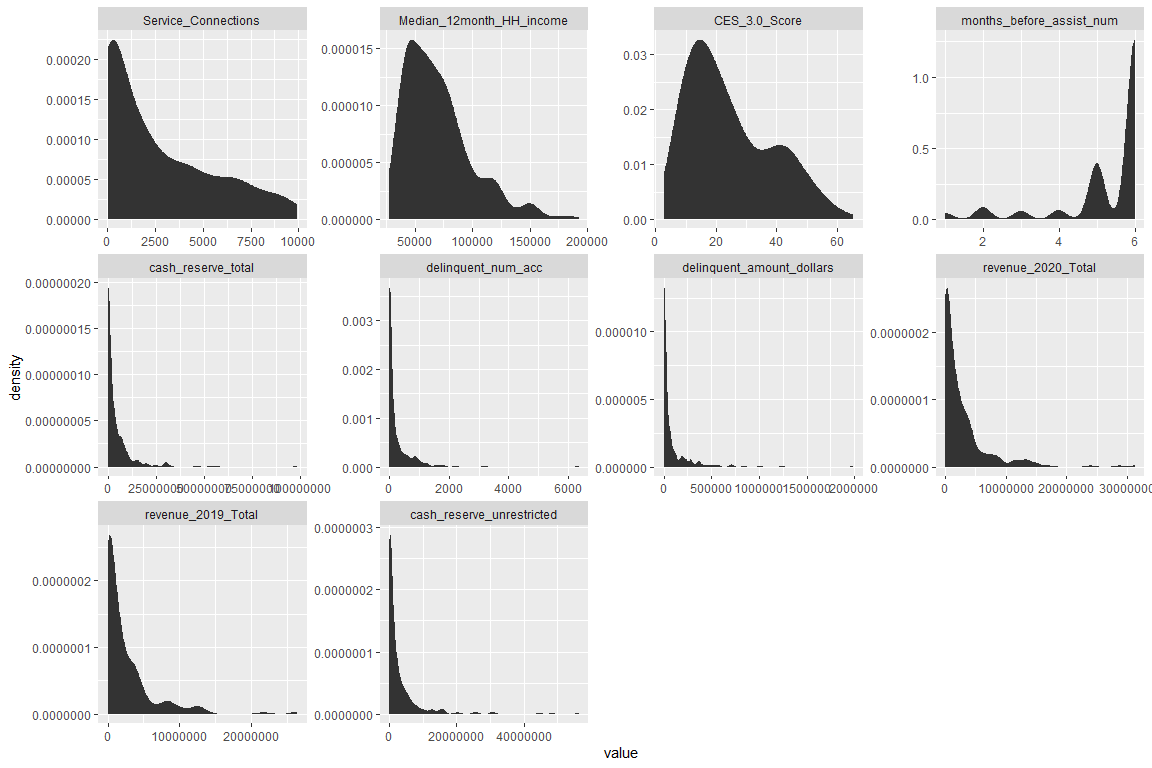
## Estimate an OLS Regression  
fitols <- lm(cash\_reserve\_total ~ Service\_Connections + Population + Median\_12month\_HH\_income +  
 Median\_rent\_pct\_income + CES\_3.0\_Score + delinquent\_num\_acc +  
 delinquent\_amount\_dollars + cash\_reserve\_unrestricted + expense\_2019\_Apr + expense\_2019\_May + expense\_2019\_Jun + expense\_2019\_Jul + expense\_2019\_Aug + expense\_2019\_Sep + expense\_2019\_Oct + expense\_2019\_Total + expense\_2020\_Apr + expense\_2020\_May + expense\_2020\_Jun + expense\_2020\_Jul + expense\_2020\_Aug + expense\_2020\_Sep + expense\_2020\_Oct + expense\_2020\_Total, na.action = na.omit,   
 data = allSmalls.requested.responded)  
summary(fitols)

##   
## Call:  
## lm(formula = cash\_reserve\_total ~ Service\_Connections + Population +   
## Median\_12month\_HH\_income + Median\_rent\_pct\_income + CES\_3.0\_Score +   
## delinquent\_num\_acc + delinquent\_amount\_dollars + cash\_reserve\_unrestricted +   
## expense\_2019\_Apr + expense\_2019\_May + expense\_2019\_Jun +   
## expense\_2019\_Jul + expense\_2019\_Aug + expense\_2019\_Sep +   
## expense\_2019\_Oct + expense\_2019\_Total + expense\_2020\_Apr +   
## expense\_2020\_May + expense\_2020\_Jun + expense\_2020\_Jul +   
## expense\_2020\_Aug + expense\_2020\_Sep + expense\_2020\_Oct +   
## expense\_2020\_Total, data = allSmalls.requested.responded,   
## na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16054936 -2006561 -64978 757039 26096948   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -2499896.73832 3664027.63834 -0.682  
## Service\_Connections 899.29028 447.43858 2.010  
## Population -75.34618 103.36274 -0.729  
## Median\_12month\_HH\_income 1.33301 18.76990 0.071  
## Median\_rent\_pct\_income 56510.28630 84468.57108 0.669  
## CES\_3.0\_Score 21061.11757 37231.40880 0.566  
## delinquent\_num\_acc -2073.36506 1612.66467 -1.286  
## delinquent\_amount\_dollars 7.70584 4.91284 1.569  
## cash\_reserve\_unrestricted 1.64825 0.08092 20.369  
## expense\_2019\_Apr -0.06721 0.23730 -0.283  
## expense\_2019\_May 1.78100 7.20815 0.247  
## expense\_2019\_Jun -2.62133 3.78072 -0.693  
## expense\_2019\_Jul -4.93574 5.11676 -0.965  
## expense\_2019\_Aug -7.07638 4.51667 -1.567  
## expense\_2019\_Sep 5.43994 3.88695 1.400  
## expense\_2019\_Oct -2.60561 5.38237 -0.484  
## expense\_2019\_Total 2.15139 3.68226 0.584  
## expense\_2020\_Apr 51.54168 124.98708 0.412  
## expense\_2020\_May 59.98199 125.23684 0.479  
## expense\_2020\_Jun 61.88590 125.18047 0.494  
## expense\_2020\_Jul 63.99888 125.16524 0.511  
## expense\_2020\_Aug 61.12710 125.11865 0.489  
## expense\_2020\_Sep 62.59513 125.20098 0.500  
## expense\_2020\_Oct 59.02376 125.02681 0.472  
## expense\_2020\_Total -61.70121 125.09058 -0.493  
## Pr(>|t|)   
## (Intercept) 0.4963   
## Service\_Connections 0.0465 \*   
## Population 0.4673   
## Median\_12month\_HH\_income 0.9435   
## Median\_rent\_pct\_income 0.5047   
## CES\_3.0\_Score 0.5726   
## delinquent\_num\_acc 0.2008   
## delinquent\_amount\_dollars 0.1192   
## cash\_reserve\_unrestricted <0.0000000000000002 \*\*\*  
## expense\_2019\_Apr 0.7774   
## expense\_2019\_May 0.8052   
## expense\_2019\_Jun 0.4893   
## expense\_2019\_Jul 0.3365   
## expense\_2019\_Aug 0.1196   
## expense\_2019\_Sep 0.1640   
## expense\_2019\_Oct 0.6291   
## expense\_2019\_Total 0.5601   
## expense\_2020\_Apr 0.6807   
## expense\_2020\_May 0.6328   
## expense\_2020\_Jun 0.6219   
## expense\_2020\_Jul 0.6100   
## expense\_2020\_Aug 0.6260   
## expense\_2020\_Sep 0.6180   
## expense\_2020\_Oct 0.6377   
## expense\_2020\_Total 0.6227   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4936000 on 130 degrees of freedom  
## (222 observations deleted due to missingness)  
## Multiple R-squared: 0.8464, Adjusted R-squared: 0.818   
## F-statistic: 29.85 on 24 and 130 DF, p-value: < 0.00000000000000022

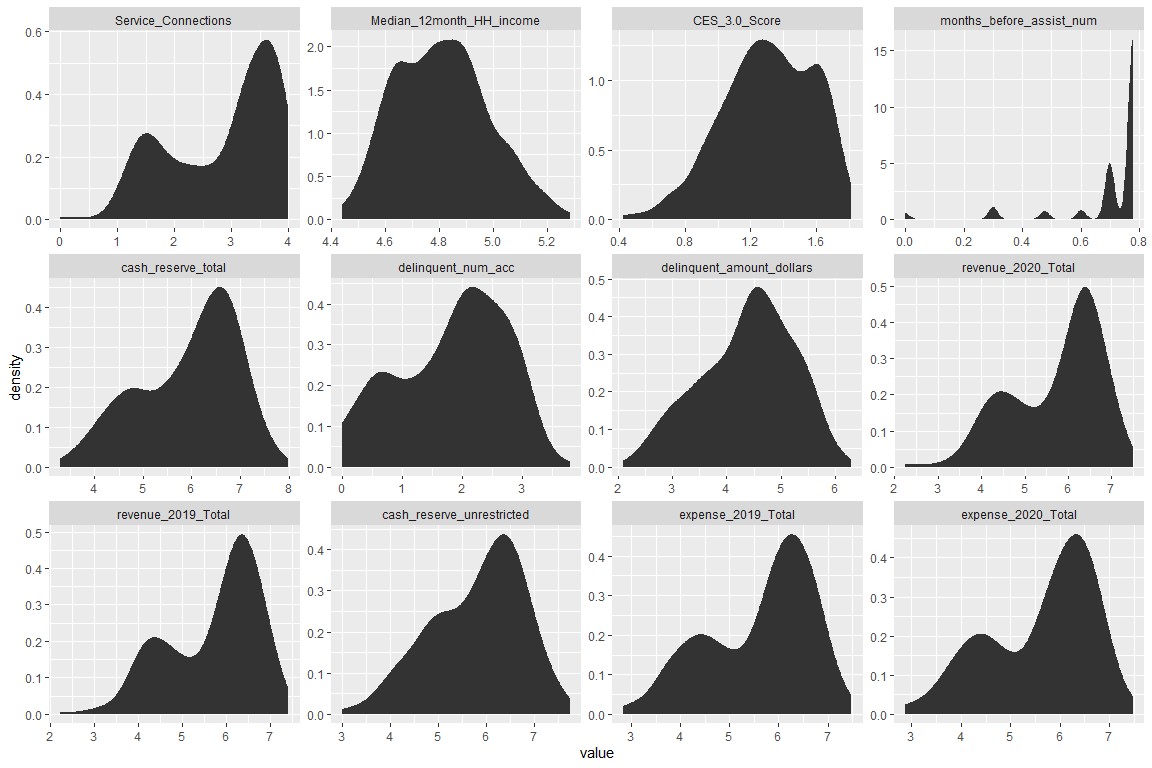
As we can see in the table above, a significant number of observations were deleted due to missingness (76 out of 371). Since this is greater than 14% of the whole dataset, our survey would be limited in power.

Imputation can now proceed. First we need to prepare the dataset for imputation. It’s best to keep it in it’s rawest form, so any categorical factors should be left as so, instead of using the ordinal transformed variable from above. Further, we want to remove variables that have haver than 25% missing values because they may mess up the imputation. It’s also important to remove variables that are highly correlated with others so as to stop the imputation working otherwise. Additionally, any extreme outliers should be removed, as they may dramatically impact results.

require(reshape2)  
allSmalls.requested.responded %>%   
 select(PWSID, Service\_Connections, Median\_12month\_HH\_income, CES\_3.0\_Score,  
 months\_before\_assist\_num, cash\_reserve\_total, delinquent\_num\_acc, delinquent\_amount\_dollars,  
 revenue\_2020\_Total, revenue\_2019\_Total, cash\_reserve\_unrestricted) %>%   
 # expense\_2019\_Apr , expense\_2019\_May , expense\_2019\_Jun , expense\_2019\_Jul , expense\_2019\_Aug , expense\_2019\_Sep , expense\_2019\_Oct , expense\_2019\_Total , expense\_2020\_Apr , expense\_2020\_May , expense\_2020\_Jun , expense\_2020\_Jul , expense\_2020\_Aug , expense\_2020\_Sep , expense\_2020\_Oct , expense\_2020\_Total) %>%   
 melt() %>% #convert wide to long  
 ggplot(aes(x = value)) +   
 stat\_density() +   
 facet\_wrap(~variable, scales = "free")

 Distribution would benefit from log10 transformation for visualizing and statistics.

require(reshape2)  
allSmalls.requested.responded %>%   
 select(PWSID, Service\_Connections, Median\_12month\_HH\_income, CES\_3.0\_Score,  
 months\_before\_assist\_num, cash\_reserve\_total, delinquent\_num\_acc, delinquent\_amount\_dollars,  
 revenue\_2020\_Total, revenue\_2019\_Total, cash\_reserve\_unrestricted, expense\_2019\_Total, expense\_2020\_Total) %>%   
 mutate\_if(~is.numeric(.) && (.) > 0, log10) %>%   
 melt() %>% #convert wide to long  
 ggplot(aes(x = value)) +   
 stat\_density() +   
 facet\_wrap(~variable, scales = "free")

 Now distributions look more or less “normal.”

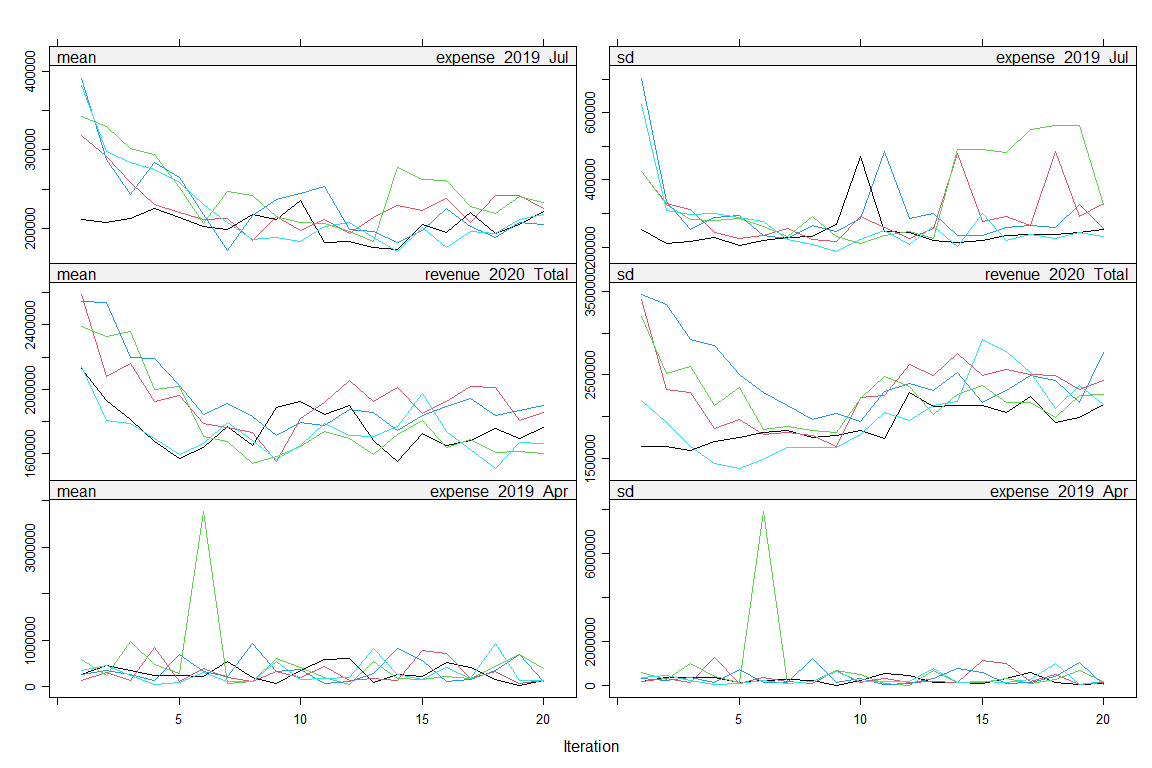
Variables missing at high rates, such as sub-metered status will not be imputed. Also, the monetary values are bins and are not “missing” so much as they are just formatted long-wise binary. We’ll remove those too.

The mice package assumes a distribution for each variable and imputes missing variables according to that distribution.

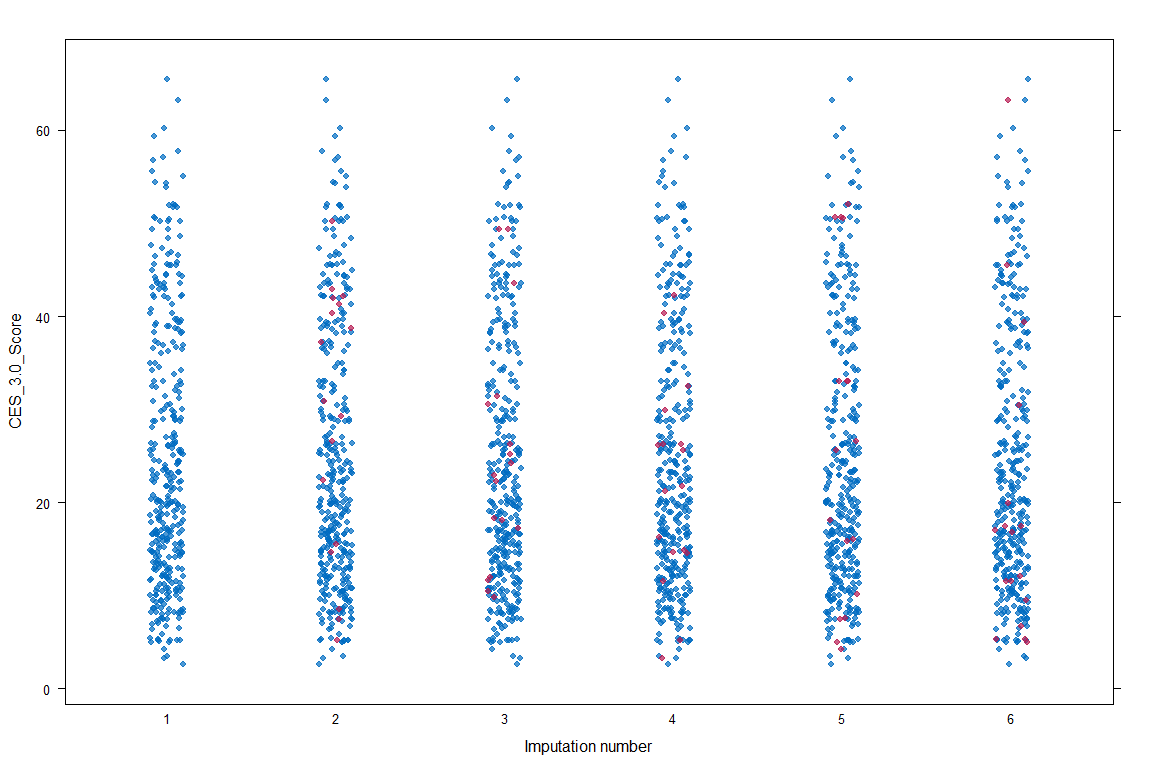
## PWSID Service\_Connections Median\_12month\_HH\_income   
## "" "" "pmm"   
## CES\_3.0\_Score months\_before\_assist cash\_reserve\_total   
## "pmm" "polr" "pmm"   
## delinquent\_num\_acc delinquent\_amount\_dollars revenue\_2020\_Total   
## "pmm" "pmm" "pmm"   
## revenue\_2019\_Total cash\_reserve\_unrestricted expense\_2019\_Apr   
## "pmm" "pmm" "pmm"   
## expense\_2019\_May expense\_2019\_Jun expense\_2019\_Jul   
## "pmm" "pmm" "pmm"   
## expense\_2019\_Aug expense\_2019\_Sep expense\_2019\_Oct   
## "pmm" "pmm" "pmm"   
## expense\_2019\_Total expense\_2020\_Apr expense\_2020\_May   
## "pmm" "pmm" "pmm"   
## expense\_2020\_Jun expense\_2020\_Jul expense\_2020\_Aug   
## "pmm" "pmm" "pmm"   
## expense\_2020\_Sep expense\_2020\_Oct expense\_2020\_Total   
## "pmm" "pmm" "pmm"

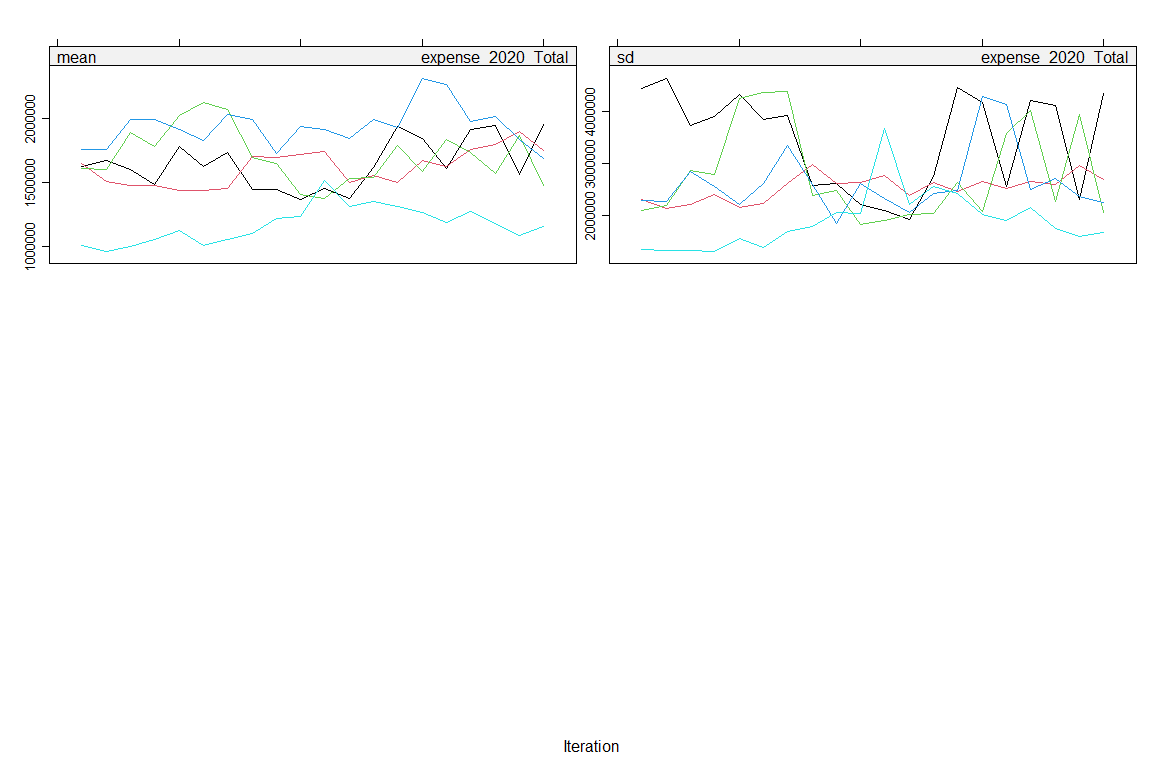
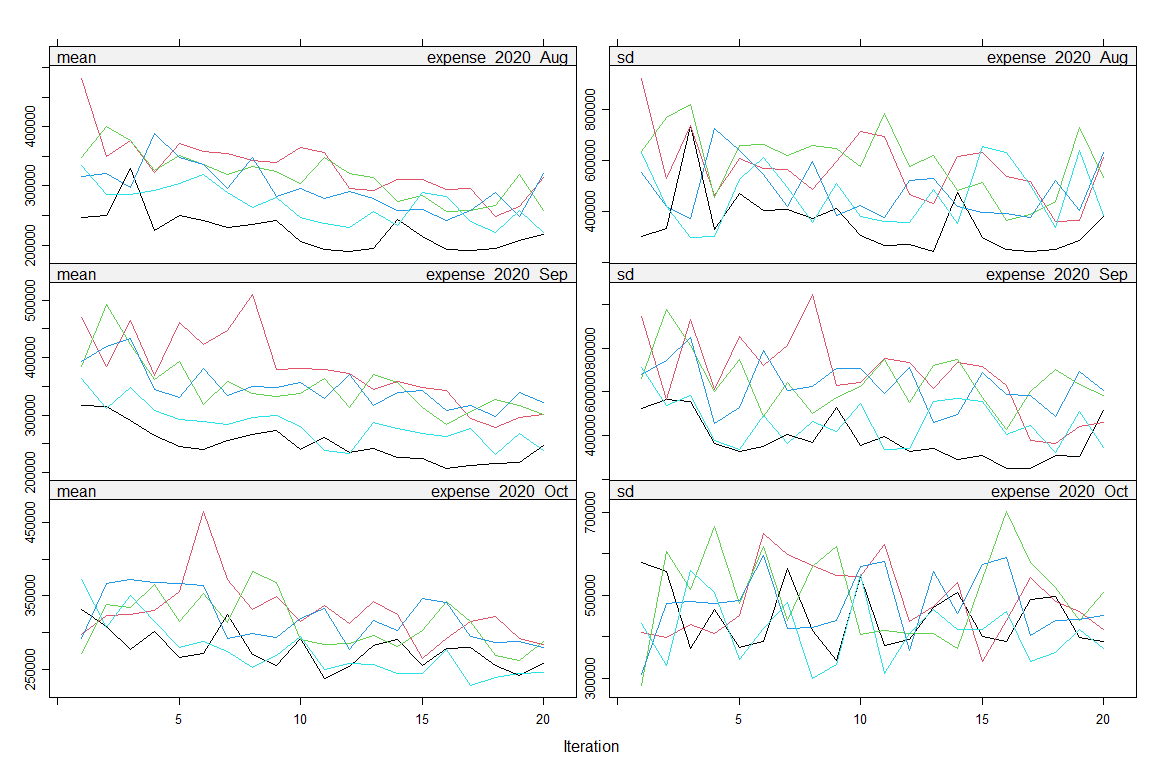
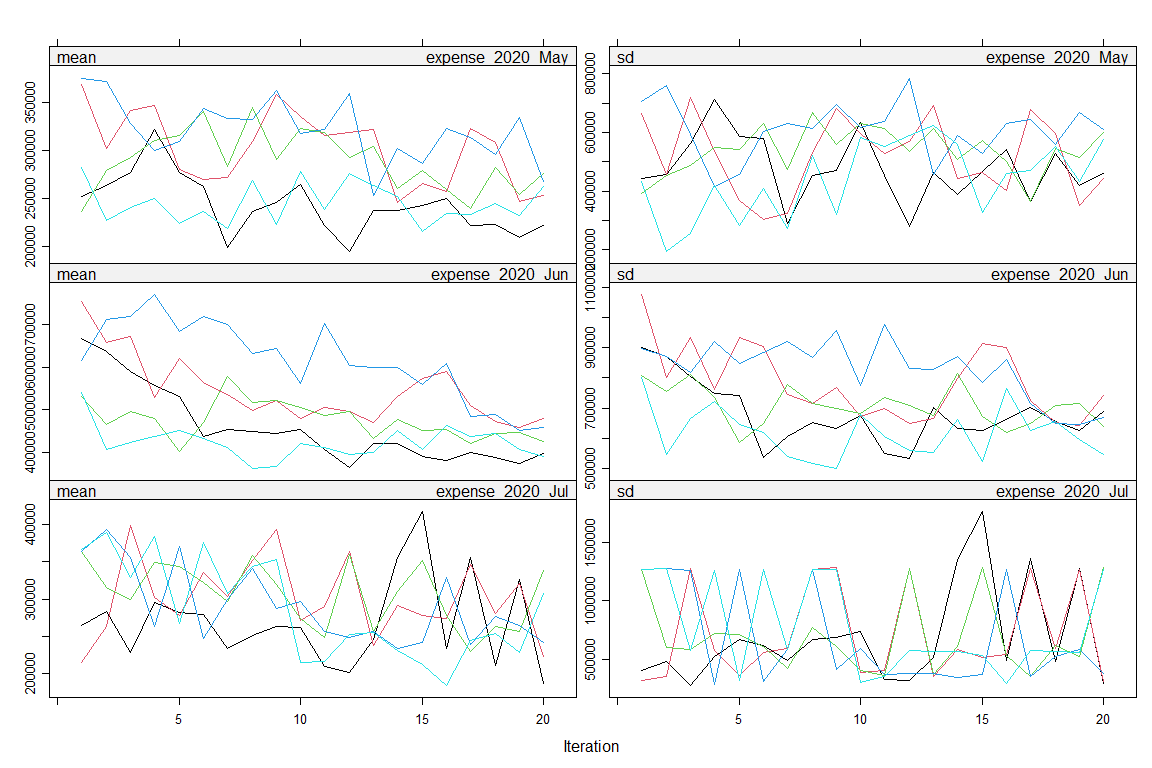
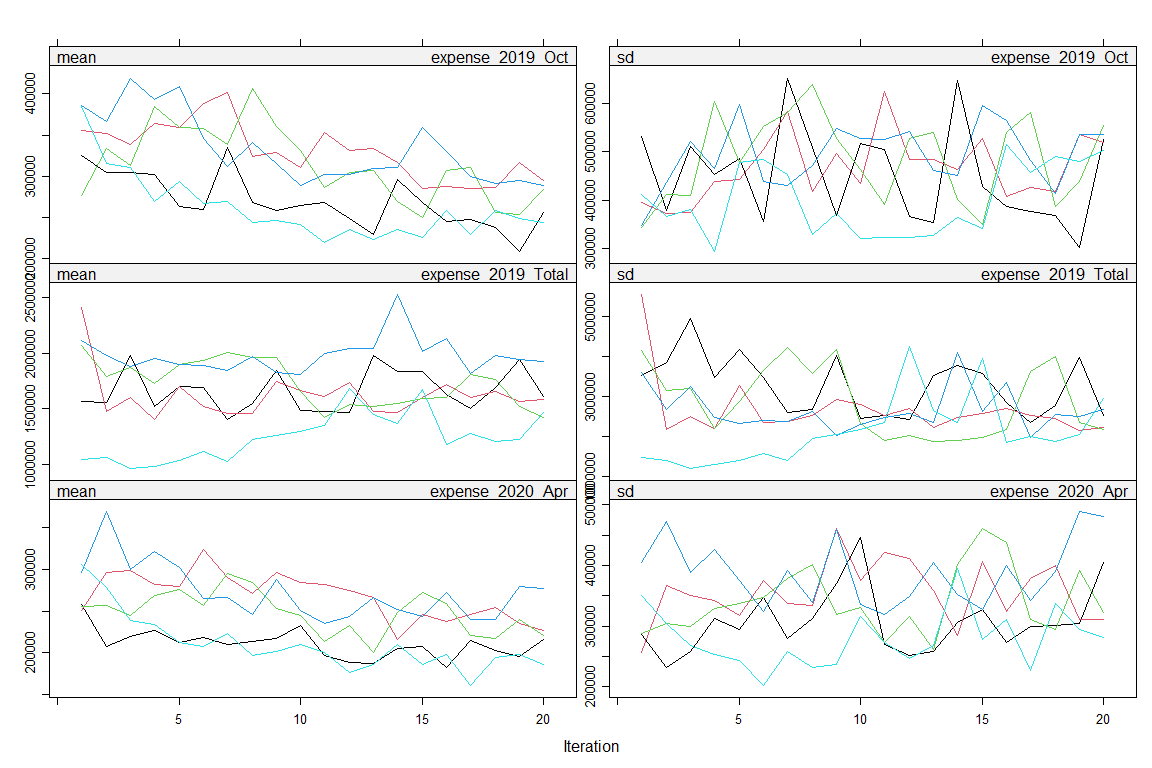
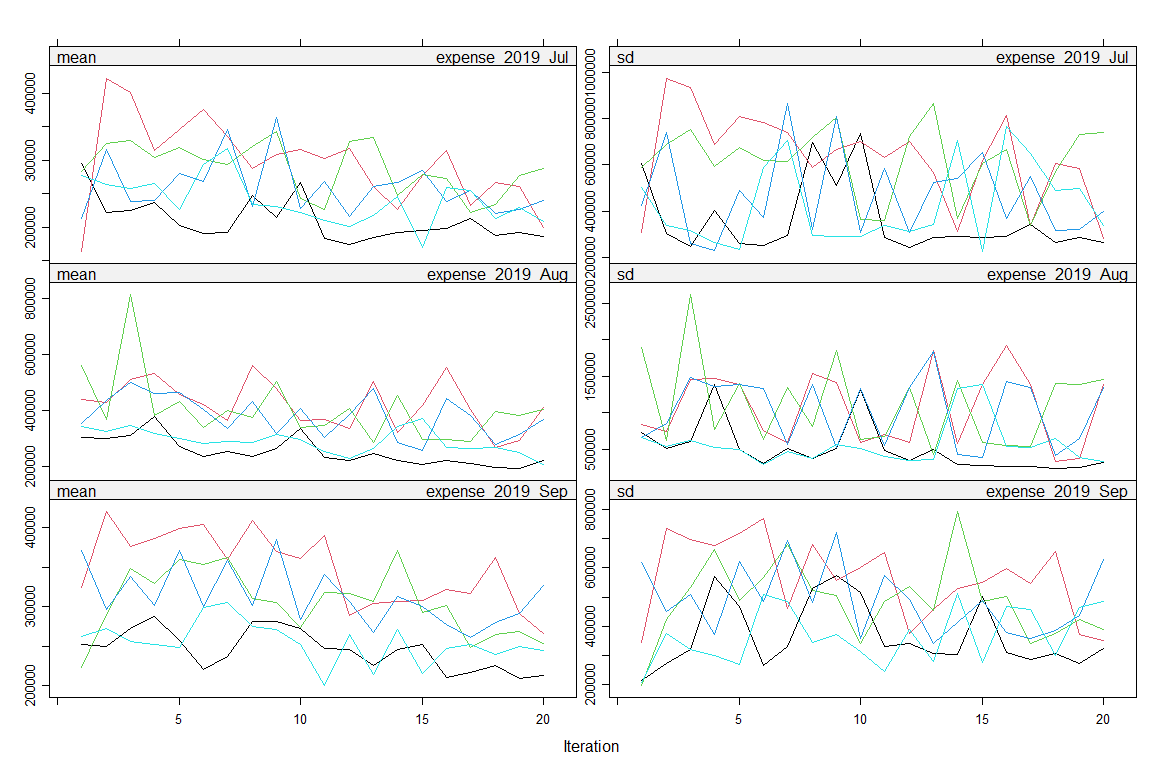
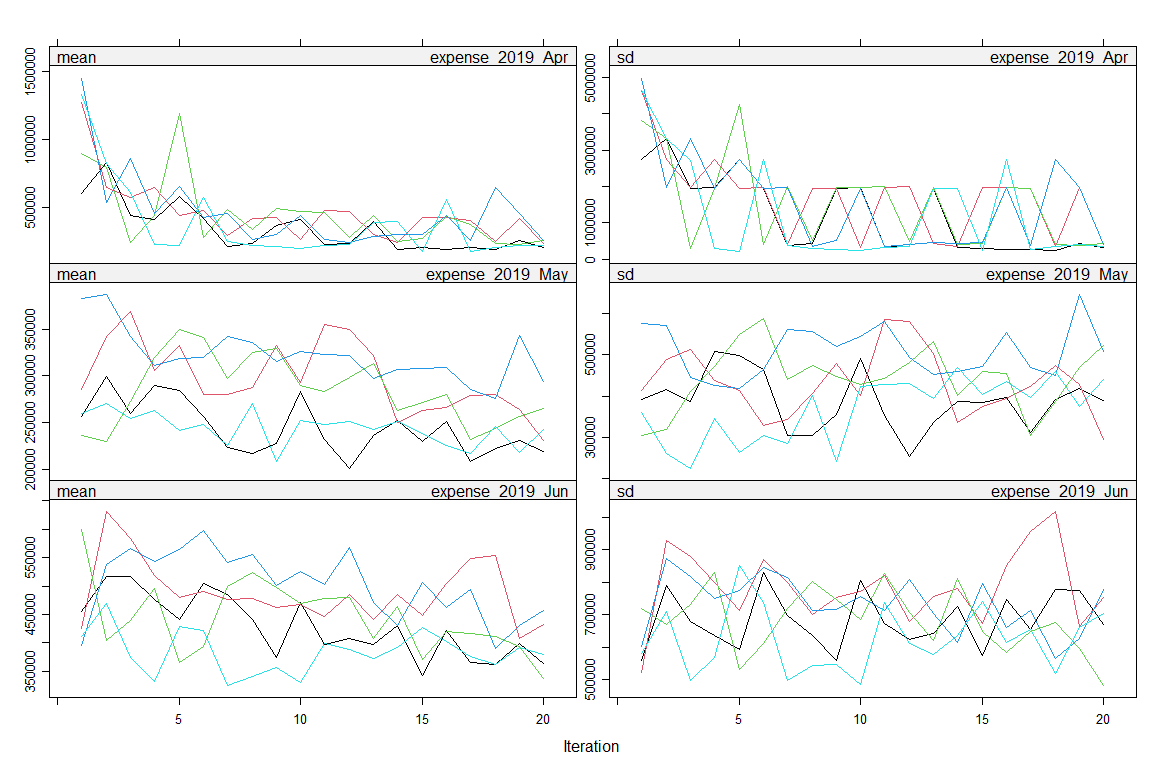
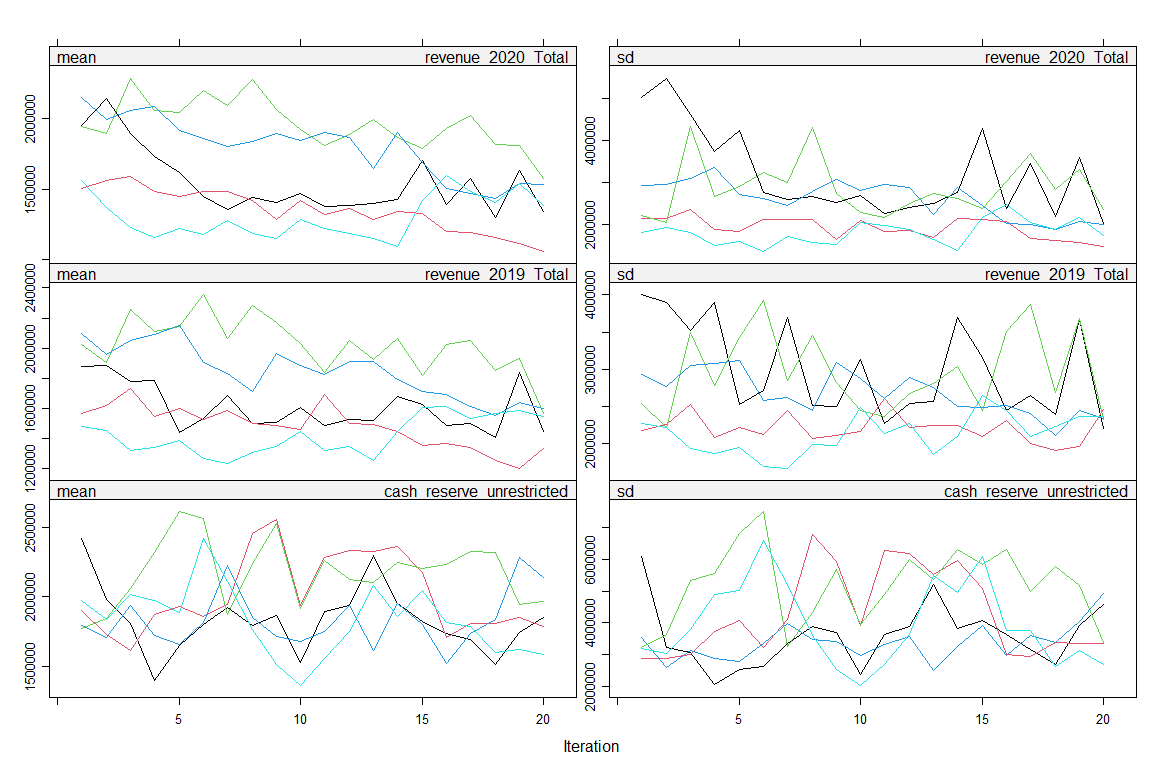
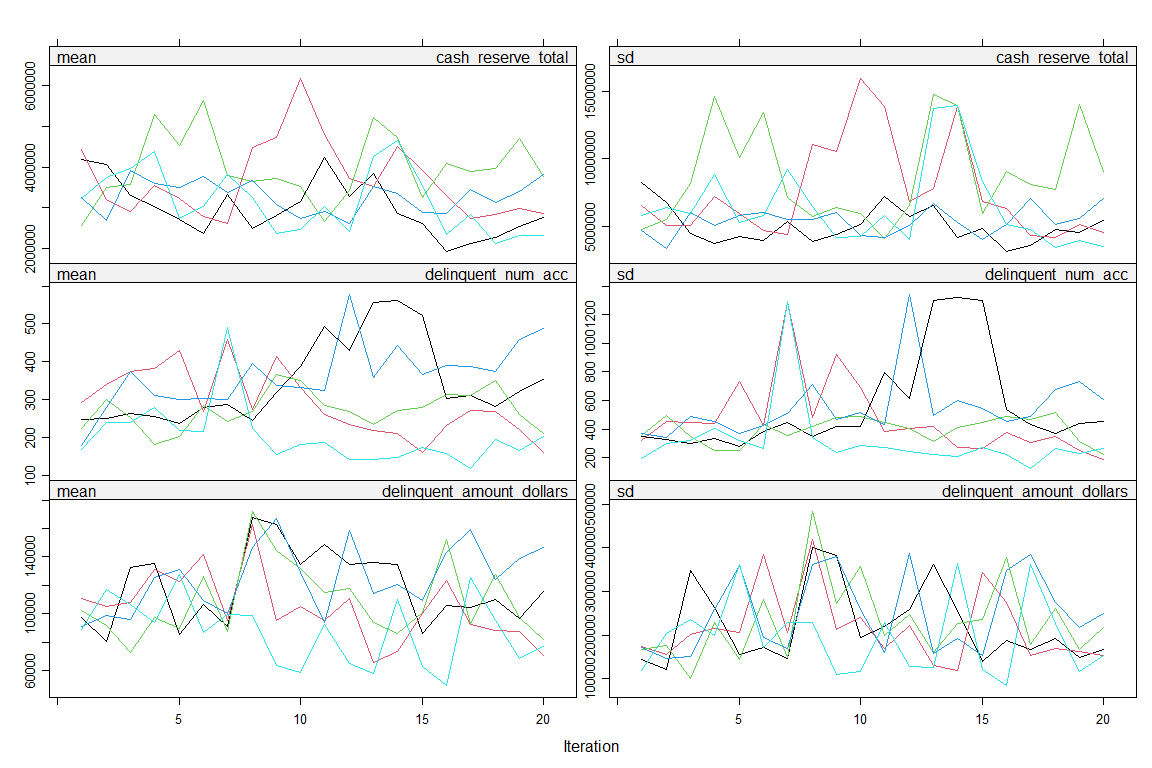
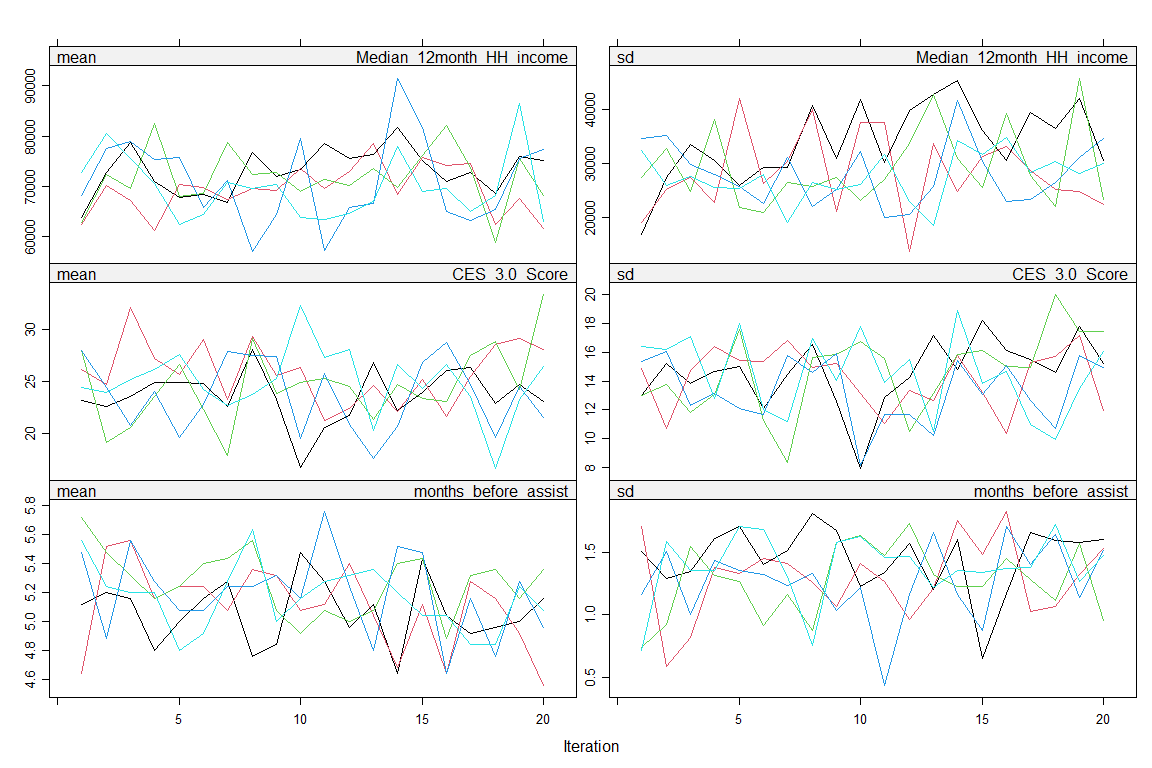
Our variables of interest are now configured to be imputed with the imputation method we specified. Empty cells in the method matrix means that those variables aren’t going to be imputed.We are now ready for multiple imputation. This step may take a few minutes.

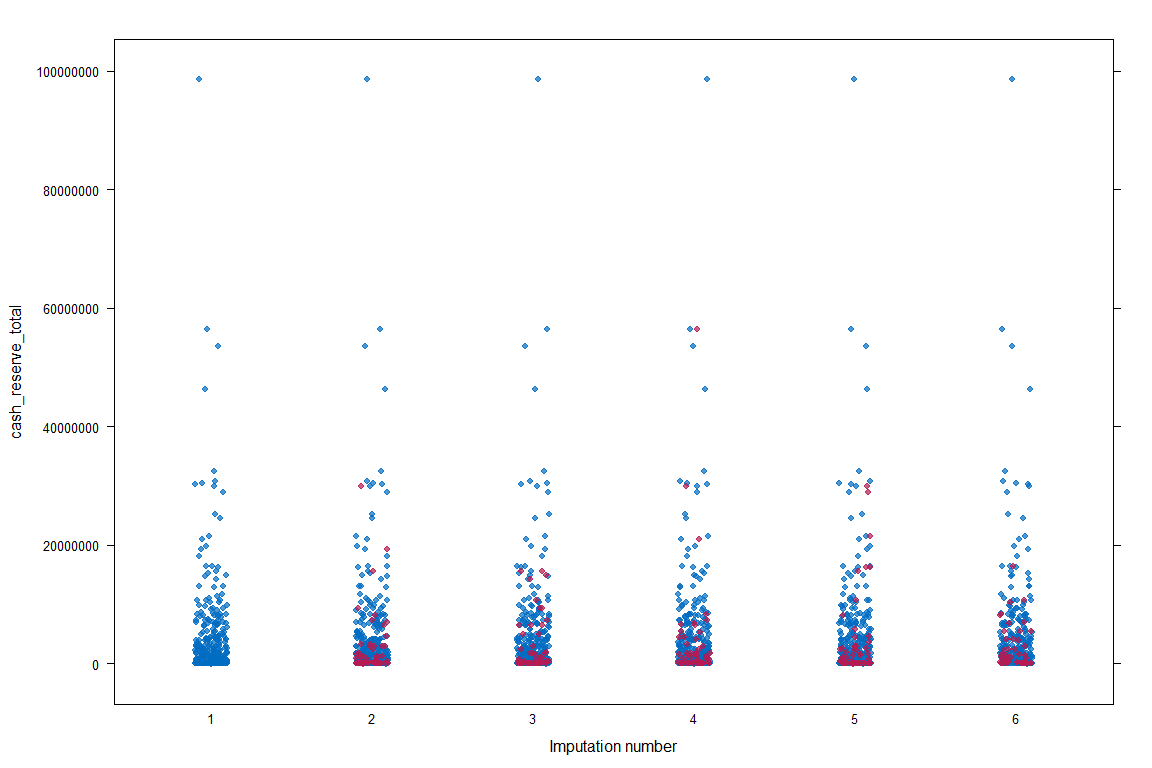
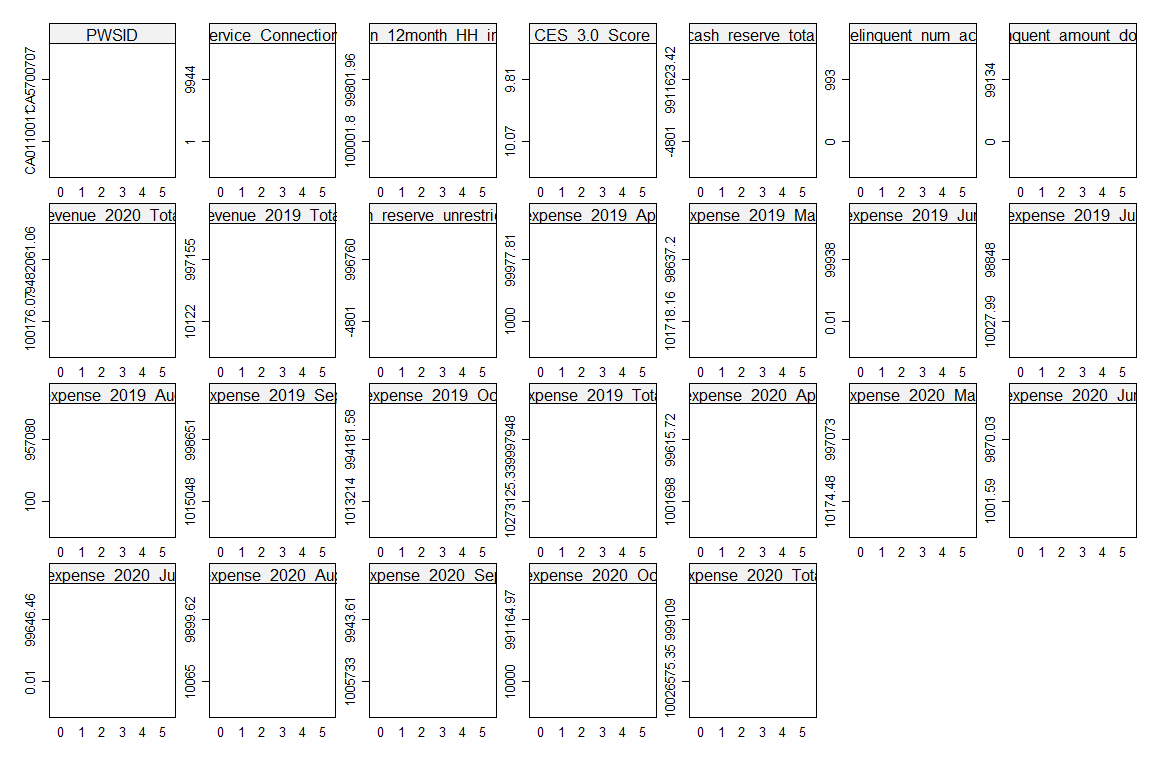
#There is a special function called quickpred() for a quick selection procedure of predictors, which can be handy for datasets containing many variables. selecting predictors according to data relations with a minimum correlation of ρ=.30 can be done by:  
ini <- mice(allSmalls.simple, pred=quickpred(allSmalls.simple, mincor=.3), print=F,  
 maxit = 20)  
#plot(ini)  
plot(ini, c("expense\_2019\_Jul", "revenue\_2020\_Total","expense\_2019\_Apr"))

 The mice package implements an iterative Markov Chain Monte Carlo type of algorithm. The plots above show the mean(left) and standard deviation(right) of the imputed values. In general, streams should intermingle and be free of any trends at the later iterations. Convergence is inspected for all variables (subset shown in the above figure).

# inspect quality of imputations  
stripplot(ini, CES\_3.0\_Score~.imp, pch = 19, xlab = "Imputation number")

 In the above figure, imputations are annotated below in red, with observed values in blue for CalEnviroScreen Score 3.0.



Further diagnostics are run on imputation (not shown here) to ensure that the best model is chosen for each imputed variable. In particular, data imputated values were inspected to ensure they are plausible values, i.e. values that could have been observed if they had not been missing. Plausibility is checked both graphically for each variable, and statistically by comparing summary statistics and relationship with against the observed values. Following these diagnostics, it is determined that the missing data are missing completely at random due to the imputations having the same distributions as the observed data.  We can see above that the imputed values are indeed realistic. 

Ordinal least squares regression is carried out on each of the imputed datasets, and theestimates are pooled together to get average regression coefficients and correct standard errors.

#First, turn the datasets into long format  
allSmalls.simple\_long <- mice::complete(imp2, action="long", include = TRUE)  
  
# #provide integer tag for voluntary status  
# allSmalls.simple\_long %<>%   
# mutate(volunteered = case\_when(voluntary == "y" ~ 1,voluntary == "n" ~ 0))  
#provide integer tag for loan status  
  
# Convert back to mids type - mice can work with this type  
allSmalls.simple\_long\_mids <- as.mids(allSmalls.simple\_long)  
# Regression   
  
fitimp <- with(allSmalls.simple\_long\_mids,  
 lm(months\_before\_assist\_num ~ Service\_Connections + Median\_12month\_HH\_income + Median\_rent\_pct\_income + CES\_3.0\_Score + delinquent\_num\_acc + volunteered + delinquent\_amount\_dollars, na.action = na.omit, data = allSmalls.responded))  
  
kable(summary(pool(fitimp)),  
 caption = "Ordinal Least Squares Regression for Imputed dataset",  
 digits = 4)

Ordinal Least Squares Regression for Imputed dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | df | p.value |
| (Intercept) | 4.5795 | 0.5259 | 8.7075 | 341.9828 | 0.0000 |
| Service\_Connections | 0.0000 | 0.0000 | 0.8622 | 341.9828 | 0.3892 |
| Median\_12month\_HH\_income | 0.0000 | 0.0000 | 2.3551 | 341.9828 | 0.0191 |
| Median\_rent\_pct\_income | 0.0192 | 0.0123 | 1.5634 | 341.9828 | 0.1189 |
| CES\_3.0\_Score | -0.0111 | 0.0053 | -2.0986 | 341.9828 | 0.0366 |
| delinquent\_num\_acc | -0.0005 | 0.0003 | -2.1005 | 341.9828 | 0.0364 |
| volunteered | -0.4054 | 0.2362 | -1.7160 | 341.9828 | 0.0871 |
| delinquent\_amount\_dollars | 0.0000 | 0.0000 | -0.3339 | 341.9828 | 0.7387 |

The pooled coefficients and p-values from the imputed datasets are compared to determine if any trends are altered (either become more pronounced or less). They appear to be similiar to the listwise-deletion technique.

## Class: mipo m = 5   
## term m estimate ubar b  
## 1 (Intercept) 5 4.5795100195280 0.2766019655008780109 0  
## 2 Service\_Connections 5 0.0000252272303 0.0000000008560314812 0  
## 3 Median\_12month\_HH\_income 5 0.0000066316540 0.0000000000079289679 0  
## 4 Median\_rent\_pct\_income 5 0.0192089051312 0.0001509654131979459 0  
## 5 CES\_3.0\_Score 5 -0.0110661839976 0.0000278067022851698 0  
## 6 delinquent\_num\_acc 5 -0.0005409444659 0.0000000663234425403 0  
## 7 volunteered 5 -0.4053639147646 0.0558005662408733727 0  
## 8 delinquent\_amount\_dollars 5 -0.0000002359419 0.0000000000004994012 0  
## t dfcom df riv lambda fmi  
## 1 0.2766019655008780109 344 341.9828 0 0 0.005797391  
## 2 0.0000000008560314812 344 341.9828 0 0 0.005797391  
## 3 0.0000000000079289679 344 341.9828 0 0 0.005797391  
## 4 0.0001509654131979459 344 341.9828 0 0 0.005797391  
## 5 0.0000278067022851698 344 341.9828 0 0 0.005797391  
## 6 0.0000000663234425403 344 341.9828 0 0 0.005797391  
## 7 0.0558005662408733727 344 341.9828 0 0 0.005797391  
## 8 0.0000000000004994012 344 341.9828 0 0 0.005797391

These relationships are compared to a default imputation model based on a “passive” imputation method.

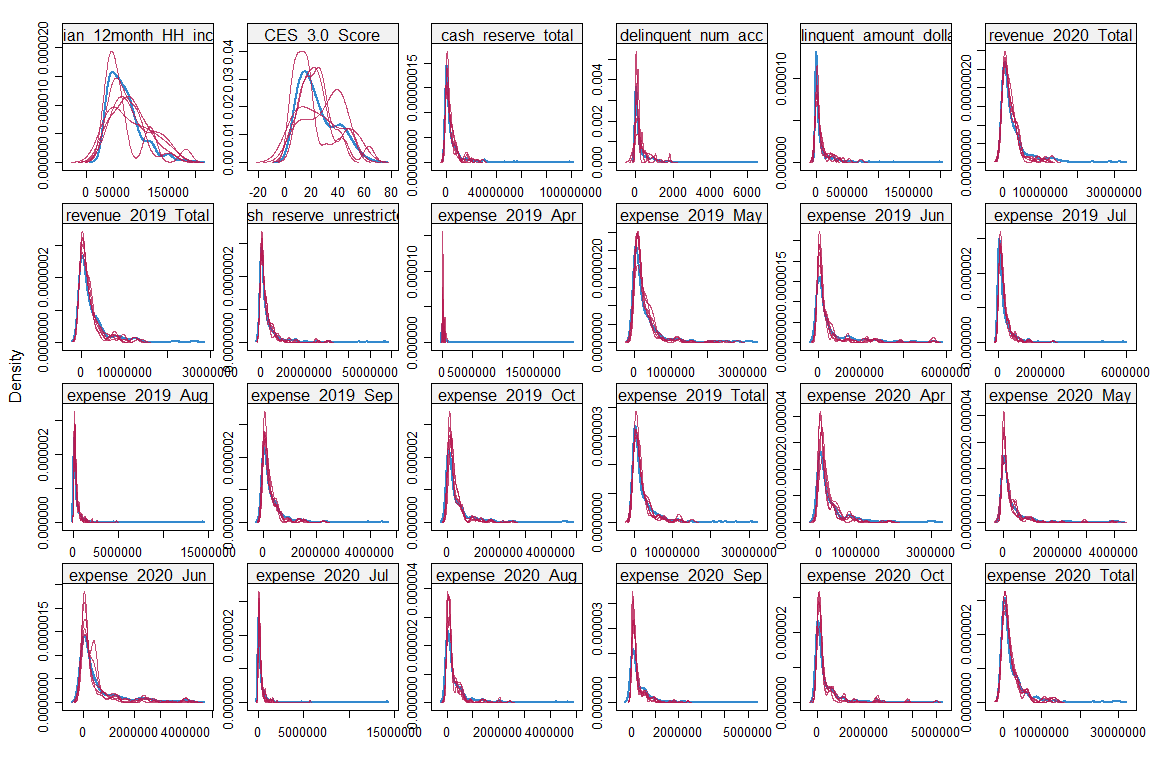
#First, turn the datasets into long format  
allSmalls.simple\_long\_1 <- mice::complete(ini, action="long", include = TRUE)  
  
# #provide integer tag for voluntary status  
# allSmalls.simple\_long %<>%   
# mutate(volunteered = case\_when(voluntary == "y" ~ 1,voluntary == "n" ~ 0))  
#provide integer tag for loan status  
  
# Convert back to mids type - mice can work with this type  
allSmalls.simple\_long\_mids\_1 <- as.mids(allSmalls.simple\_long\_1)  
# Regression   
  
fitimp\_1 <- with(allSmalls.simple\_long\_mids\_1,  
 lm(months\_before\_assist\_num ~ Service\_Connections + Median\_12month\_HH\_income + Median\_rent\_pct\_income + CES\_3.0\_Score + delinquent\_num\_acc + volunteered + delinquent\_amount\_dollars, na.action = na.omit, data = allSmalls.responded))  
  
kable(summary(pool(fitimp\_1)),  
 caption = "Ordinary Least Squares Regression for Passive Imputation",  
 digits = 4)

Ordinary Least Squares Regression for Passive Imputation

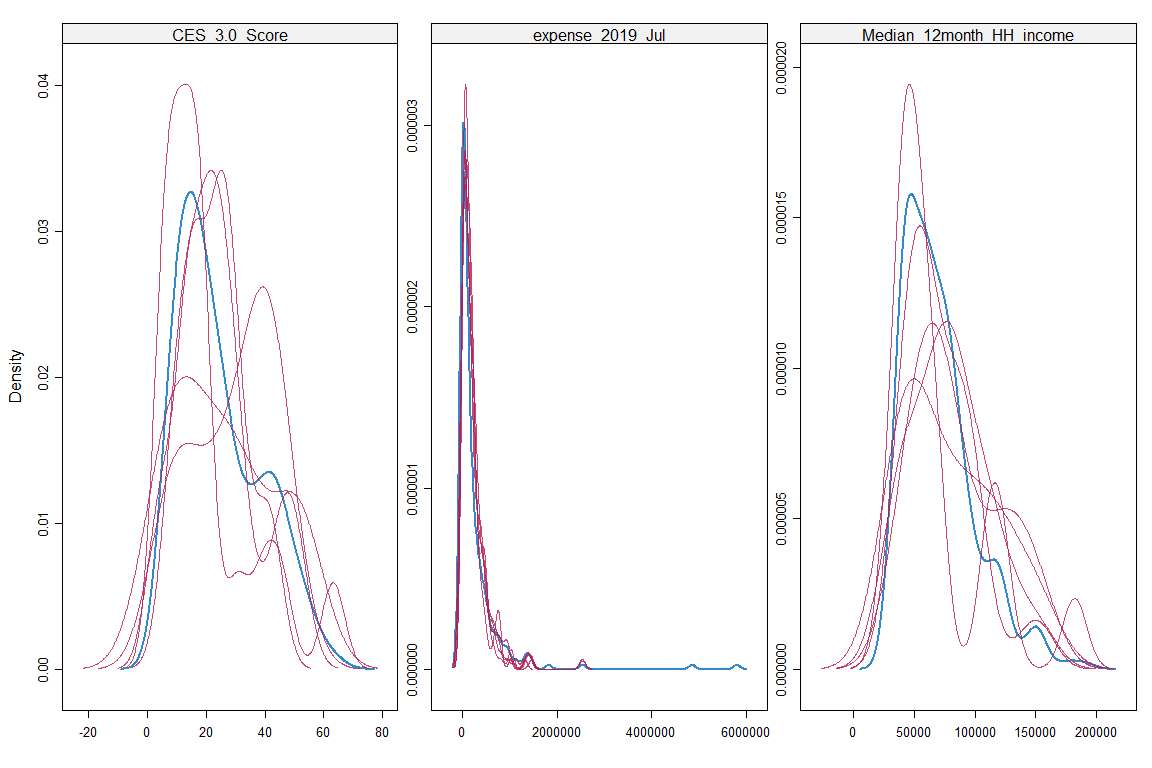
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | df | p.value |
| (Intercept) | 4.5795 | 0.5259 | 8.7075 | 341.9828 | 0.0000 |
| Service\_Connections | 0.0000 | 0.0000 | 0.8622 | 341.9828 | 0.3892 |
| Median\_12month\_HH\_income | 0.0000 | 0.0000 | 2.3551 | 341.9828 | 0.0191 |
| Median\_rent\_pct\_income | 0.0192 | 0.0123 | 1.5634 | 341.9828 | 0.1189 |
| CES\_3.0\_Score | -0.0111 | 0.0053 | -2.0986 | 341.9828 | 0.0366 |
| delinquent\_num\_acc | -0.0005 | 0.0003 | -2.1005 | 341.9828 | 0.0364 |
| volunteered | -0.4054 | 0.2362 | -1.7160 | 341.9828 | 0.0871 |
| delinquent\_amount\_dollars | 0.0000 | 0.0000 | -0.3339 | 341.9828 | 0.7387 |

## Class: mipo m = 5   
## term m estimate ubar b  
## 1 (Intercept) 5 4.5795100195280 0.2766019655008780109 0  
## 2 Service\_Connections 5 0.0000252272303 0.0000000008560314812 0  
## 3 Median\_12month\_HH\_income 5 0.0000066316540 0.0000000000079289679 0  
## 4 Median\_rent\_pct\_income 5 0.0192089051312 0.0001509654131979459 0  
## 5 CES\_3.0\_Score 5 -0.0110661839976 0.0000278067022851698 0  
## 6 delinquent\_num\_acc 5 -0.0005409444659 0.0000000663234425403 0  
## 7 volunteered 5 -0.4053639147646 0.0558005662408733727 0  
## 8 delinquent\_amount\_dollars 5 -0.0000002359419 0.0000000000004994012 0  
## t dfcom df riv lambda fmi  
## 1 0.2766019655008780109 344 341.9828 0 0 0.005797391  
## 2 0.0000000008560314812 344 341.9828 0 0 0.005797391  
## 3 0.0000000000079289679 344 341.9828 0 0 0.005797391  
## 4 0.0001509654131979459 344 341.9828 0 0 0.005797391  
## 5 0.0000278067022851698 344 341.9828 0 0 0.005797391  
## 6 0.0000000663234425403 344 341.9828 0 0 0.005797391  
## 7 0.0558005662408733727 344 341.9828 0 0 0.005797391  
## 8 0.0000000000004994012 344 341.9828 0 0 0.005797391

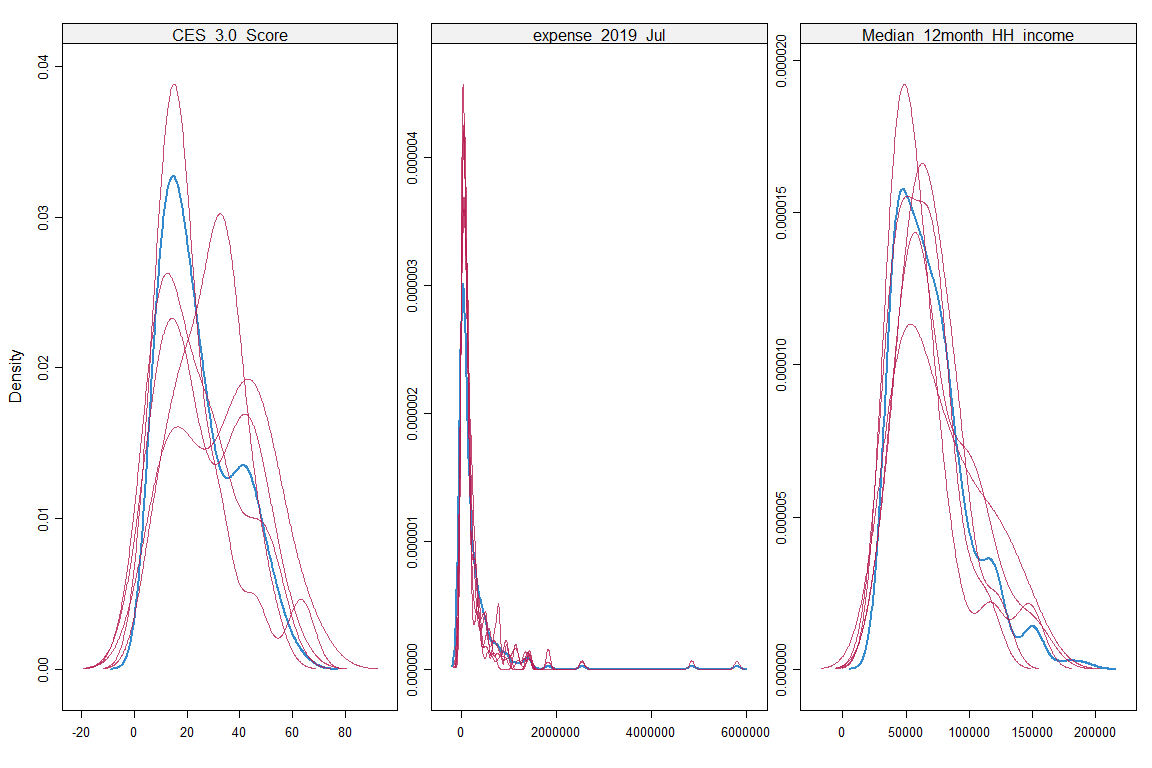
These two models are virtually the same.

Imputation models are further inspected to confirm the plausibilitiy of the imputed values by plotting the histograms (using density plots) for the real (blue) and imputed (red) values.  Variables seem to be well-predicted by the ‘quick-prediction’ model. The least well-imputed factors seem to be CES 3.0 score and 12-month household income. A sample of imputed distributions are plotted below.

densityplot(ini, ~CES\_3.0\_Score + expense\_2019\_Jul + Median\_12month\_HH\_income)

 We can see here that the quick prediction model biases towards lower CalEnviroScreen 3.0 scores. To see if the classification and regression tree (CART) model worked any better, these same plots are repeated for those imputations below.

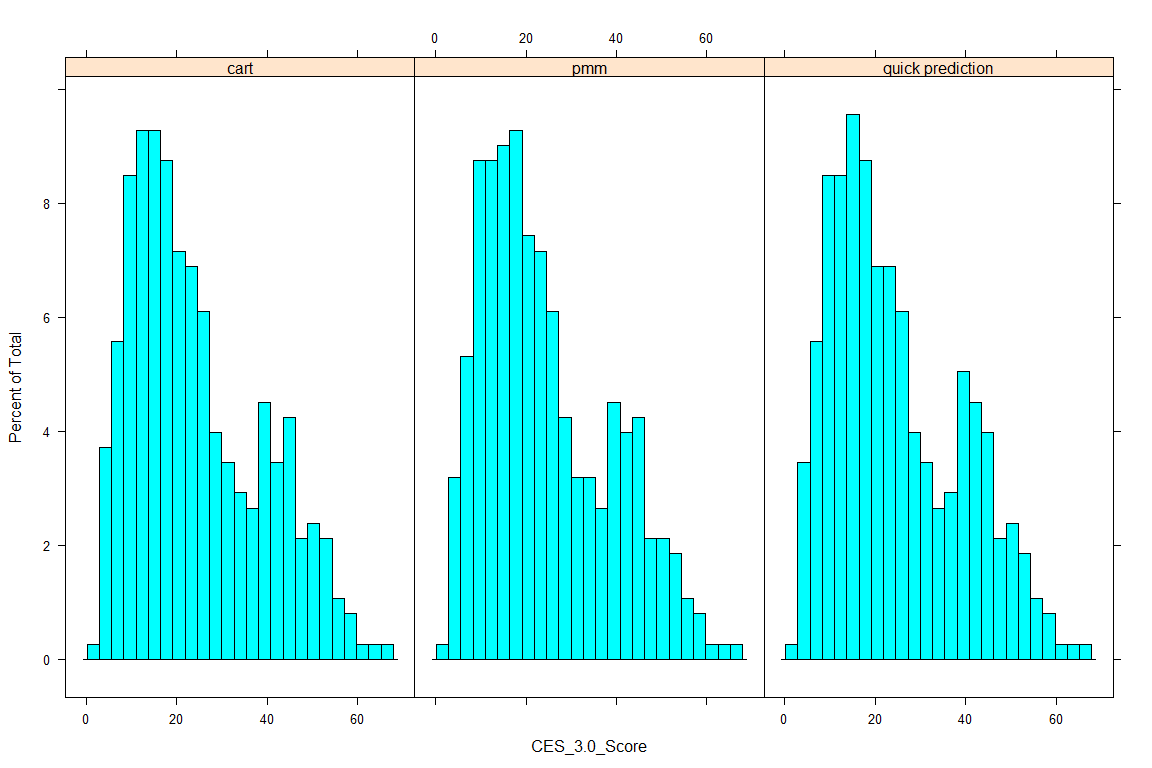
densityplot(imp2, ~CES\_3.0\_Score + expense\_2019\_Jul + Median\_12month\_HH\_income)



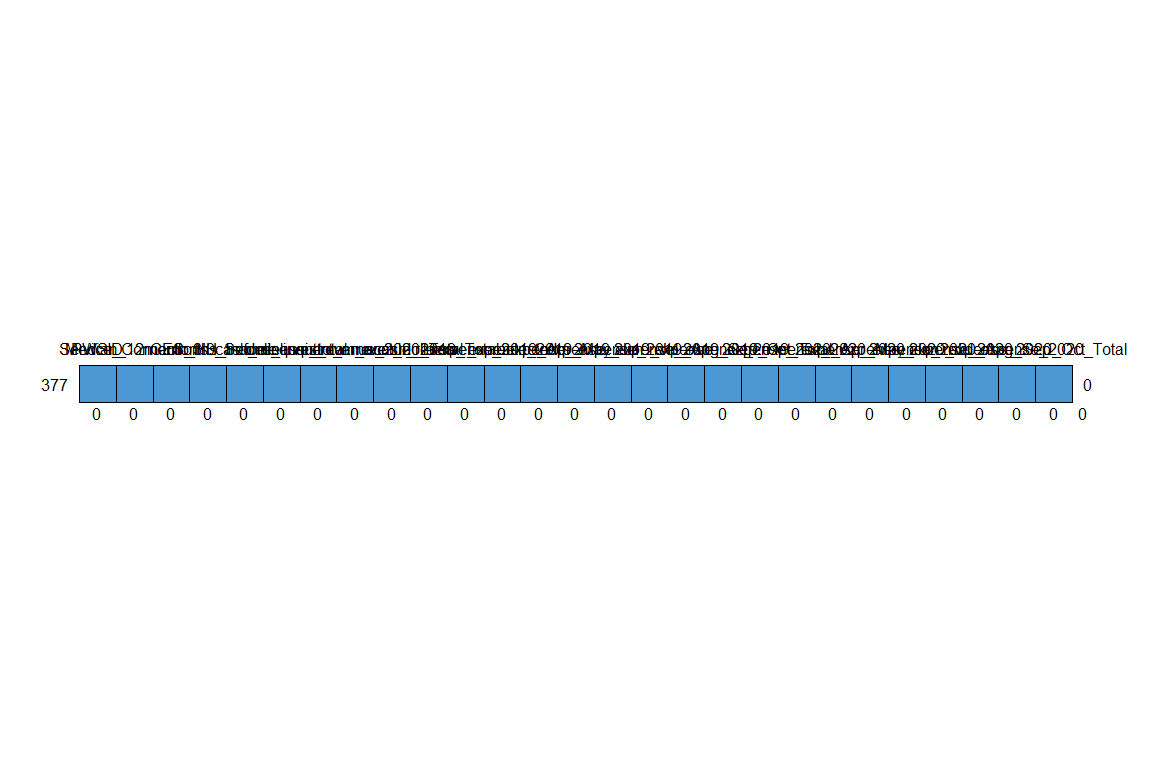
The CART method seems to have learned the distribution slightly more accurately for this variable. The most complex distribution is months before assistance needed.

The classification and regression trees method and quick-prediction imputation models are further compared with the default imputation method (passive multiple imputation) for CES 3.0 score.

CES\_3.0\_Score <- c(complete(imp)$CES\_3.0\_Score, complete(ini)$CES\_3.0\_Score, complete(imp2)$CES\_3.0\_Score)  
method <- rep(c("pmm", "quick prediction", "cart"), each = nrow(allSmalls.simple))  
CES\_3.0\_Score\_m <- data.frame(CES\_3.0\_Score = CES\_3.0\_Score, method = method)  
#plot histogram  
histogram( ~CES\_3.0\_Score | method, data = CES\_3.0\_Score\_m, nint = 25)

 Here we see that these three methods do not seemingly differ in prediction for this parameter. Based on additional inspections of imputed distributions, the CART method is chosen.

## /\ /\  
## { `---' }  
## { O O }  
## ==> V <== No need for mice. This data set is completely observed.  
## \ \|/ /  
## `-----'



## PWSID Service\_Connections Median\_12month\_HH\_income CES\_3.0\_Score  
## 377 1 1 1 1  
## 0 0 0 0  
## months\_before\_assist cash\_reserve\_total delinquent\_num\_acc  
## 377 1 1 1  
## 0 0 0  
## delinquent\_amount\_dollars revenue\_2020\_Total revenue\_2019\_Total  
## 377 1 1 1  
## 0 0 0  
## cash\_reserve\_unrestricted expense\_2019\_Apr expense\_2019\_May  
## 377 1 1 1  
## 0 0 0  
## expense\_2019\_Jun expense\_2019\_Jul expense\_2019\_Aug expense\_2019\_Sep  
## 377 1 1 1 1  
## 0 0 0 0  
## expense\_2019\_Oct expense\_2019\_Total expense\_2020\_Apr expense\_2020\_May  
## 377 1 1 1 1  
## 0 0 0 0  
## expense\_2020\_Jun expense\_2020\_Jul expense\_2020\_Aug expense\_2020\_Sep  
## 377 1 1 1 1  
## 0 0 0 0  
## expense\_2020\_Oct expense\_2020\_Total   
## 377 1 1 0  
## 0 0 0

Summary statistics for the imputed variables are compared with those in the original dataset.

simpleImputed %>%   
 mutate(Grp = "Imputed") %>%  
 bind\_rows(mutate(allSmalls.requested.responded, Grp = "Original")) %>%  
 select(Grp, delinquent\_num\_acc, delinquent\_amount\_dollars, revenue\_2020\_Total, revenue\_2019\_Total) %>%  
 group\_by(Grp) %>%   
 drop\_na() %>%   
 summarize\_all(list(mean = "mean", median = "median")) %>%   
 t() %>% #transpose  
 kable(caption = "Summary Statistics for Selected Variables fo Imputed and Original Data")

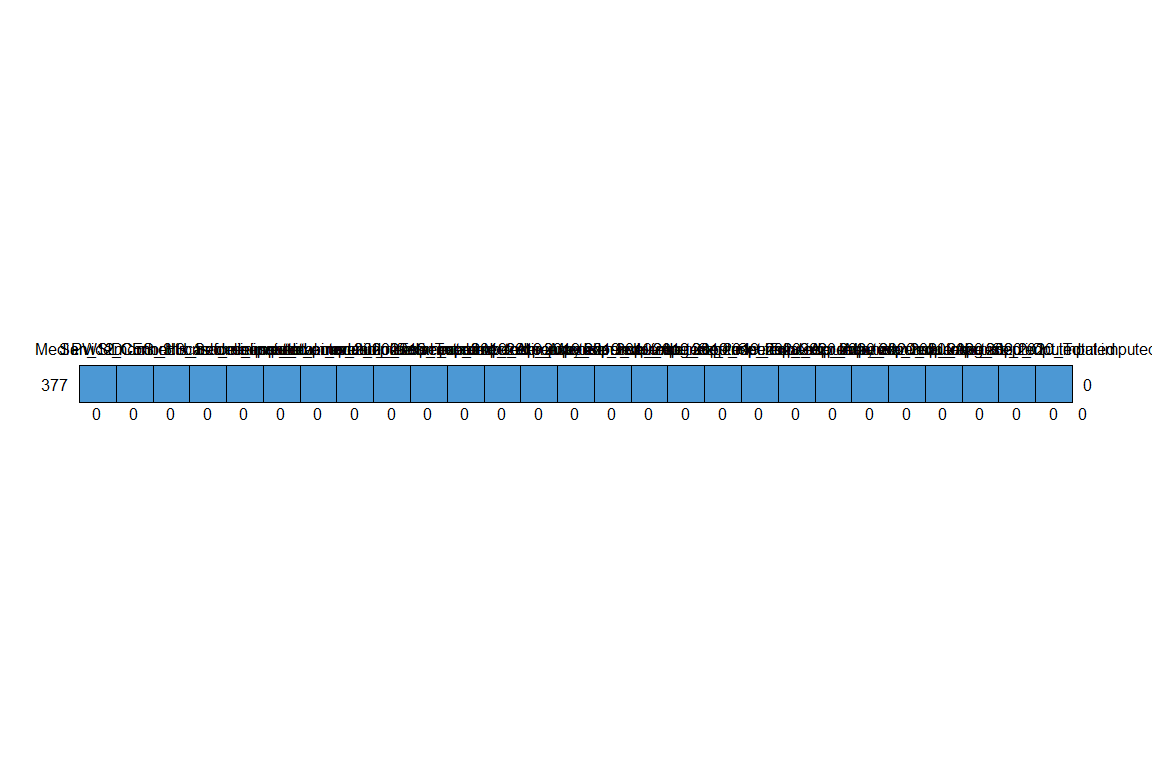
Summary Statistics for Selected Variables fo Imputed and Original Data

|  |  |  |
| --- | --- | --- |
| Grp | Imputed | Original |
| delinquent\_num\_acc\_mean | 238.5411 | 250.3636 |
| delinquent\_amount\_dollars\_mean | 82731.60 | 85696.35 |
| revenue\_2020\_Total\_mean | 2718987 | 2649865 |
| revenue\_2019\_Total\_mean | 2523240 | 2555228 |
| delinquent\_num\_acc\_median | 58 | 67 |
| delinquent\_amount\_dollars\_median | 22435.44 | 23064.53 |
| revenue\_2020\_Total\_median | 1183711 | 1288004 |
| revenue\_2019\_Total\_median | 1190912 | 1187129 |

Summary stats are plotted below.

#join data for plotting  
simpleImputed %>% mutate(Grp = "Imputed") %>%  
 bind\_rows(mutate(allSmalls.requested.responded, Grp = "Original")) %>%  
 select(Grp, delinquent\_num\_acc, delinquent\_amount\_dollars, revenue\_2020\_Total, revenue\_2019\_Total) %>%   
 melt() %>% #transforms values and variables to long format  
 ggplot(aes(x = variable, y = log10(value))) +  
 geom\_boxplot(aes(fill = Grp))+  
 geom\_jitter(aes(color = Grp), alpha = 0.3) +  
 labs(title = "Boxplot Comparing Imputed and Observed Values for Selected Variables")

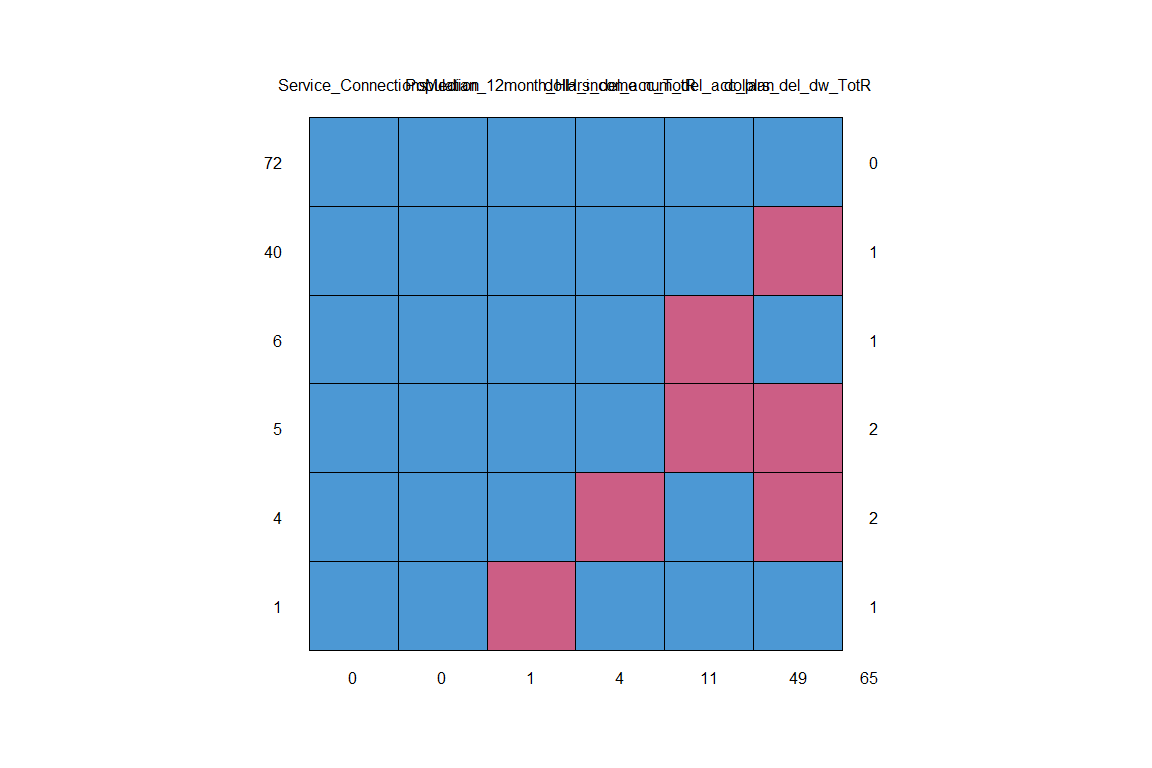
 The above figure demonstrates that imputation has not noticeably changed the summary statistics of these data.

 Note that some variables are still missing, however these were deliberately not imputed either due to no being applicable (such as binned data) or would were calculated from other variables.

## Larges Imputation

The above steps are generally repeated for large systems (>10,000 service connections) surveyed. Prior to determining if multiple imputation is necessary (as opposed to row-wise deletion or mean imputation), primary auxiliary and response variables will be examined for completeness in the matrix below.

larges <- allLarges %>% filter(responded == "y")  
#extract the meaningful parameters  
meaningfulVariables <- complete(larges) %>% select(dollars\_del\_acc\_TotR, dollars\_del\_dw\_TotR, Median\_12month\_HH\_income, Service\_Connections, Population, num\_del\_acc\_plan)  
#the imputed data can also be extracted in long format.  
#c.long <- complete(imp, "long")   
md.pattern(meaningfulVariables)



## Service\_Connections Population Median\_12month\_HH\_income dollars\_del\_acc\_TotR  
## 72 1 1 1 1  
## 40 1 1 1 1  
## 6 1 1 1 1  
## 5 1 1 1 1  
## 4 1 1 1 0  
## 1 1 1 0 1  
## 0 0 1 4  
## num\_del\_acc\_plan dollars\_del\_dw\_TotR   
## 72 1 1 0  
## 40 1 0 1  
## 6 0 1 1  
## 5 0 0 2  
## 4 1 0 2  
## 1 1 1 1  
## 11 49 65

The matrix above shows relatively few responses are missing for most variables. These can be viewed in tabular format as well.

#extract the meaningful parameters  
larges %>%   
 select(dollars\_del\_acc\_TotR, dollars\_del\_dw\_TotR, Median\_12month\_HH\_income, Service\_Connections, Population, num\_del\_acc\_plan, Fee\_Code) %>%   
 summarise\_all(funs(sum(is.na(.)))) %>% #summarize NA  
 t() %>% #transpose  
 kable(caption = "Number of Missing Values in Sample Set (n = 128)")

Number of Missing Values in Sample Set (n = 128)

|  |  |
| --- | --- |
| dollars\_del\_acc\_TotR | 4 |
| dollars\_del\_dw\_TotR | 49 |
| Median\_12month\_HH\_income | 1 |
| Service\_Connections | 0 |
| Population | 0 |
| num\_del\_acc\_plan | 11 |
| Fee\_Code | 0 |

Luckily, auxiliary variables (service connections, population, fee code) are all complete. One system lacks median-12-month income data. Vital response variables are less complete with total delinquent dollars (drinking water) the most incomplete (49 out of 128 cases missing). Multiple imputation will be employed to fill in missing values using predictive statistics.

An Ordinary Least Squares (OLS) linear regression is used to predict total delinquent dollars (i.e. debt) based on reasonable predictors that are available for all systems (e.g. service connections, population, median 12 month household income, median rent percent income, fee code, number of delinquent account plans).

## Estimate an OLS Regression  
fitols <- lm(dollars\_del\_dw\_TotR ~ Service\_Connections + Median\_12month\_HH\_income + Fee\_Code + Population +  
 Median\_rent\_pct\_income + num\_del\_acc\_plan + dollars\_del\_acc\_TotR,  
 na.action = na.omit,   
 data = larges)  
summary(fitols)

##   
## Call:  
## lm(formula = dollars\_del\_dw\_TotR ~ Service\_Connections + Median\_12month\_HH\_income +   
## Fee\_Code + Population + Median\_rent\_pct\_income + num\_del\_acc\_plan +   
## dollars\_del\_acc\_TotR, data = larges, na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4848986 -245399 108510 326119 9057736   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -1633812.36062 2693295.14129 -0.607  
## Service\_Connections -8.95018 18.21092 -0.491  
## Median\_12month\_HH\_income 0.13354 8.35601 0.016  
## Fee\_CodeDAVCL 89148.36909 819453.23399 0.109  
## Population 10.84986 3.93210 2.759  
## Median\_rent\_pct\_income 35188.17623 67117.27586 0.524  
## num\_del\_acc\_plan -148.94652 70.50575 -2.113  
## dollars\_del\_acc\_TotR 0.09934 0.01118 8.884  
## Pr(>|t|)   
## (Intercept) 0.54625   
## Service\_Connections 0.62477   
## Median\_12month\_HH\_income 0.98730   
## Fee\_CodeDAVCL 0.91371   
## Population 0.00755 \*\*   
## Median\_rent\_pct\_income 0.60190   
## num\_del\_acc\_plan 0.03855 \*   
## dollars\_del\_acc\_TotR 0.000000000000903 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1504000 on 64 degrees of freedom  
## (56 observations deleted due to missingness)  
## Multiple R-squared: 0.9784, Adjusted R-squared: 0.976   
## F-statistic: 414.3 on 7 and 64 DF, p-value: < 0.00000000000000022

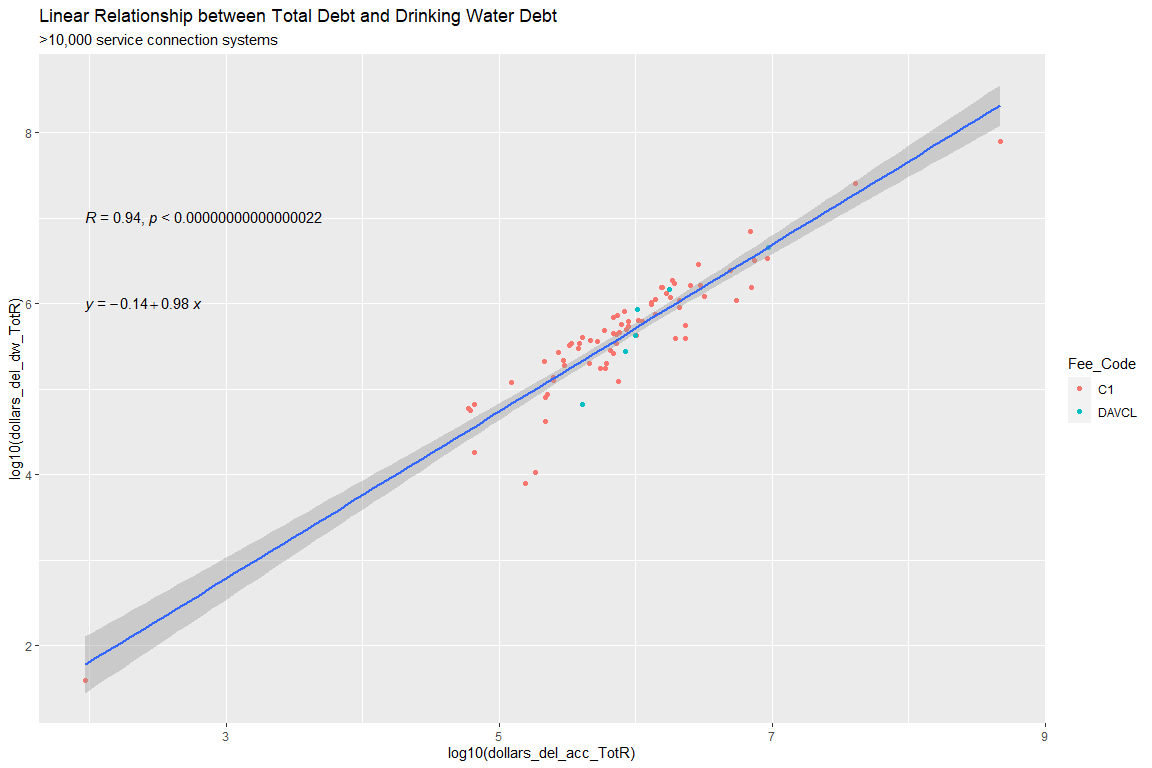
As we can see in the table above, a significant number of observations were deleted due to missingness (56 out of 128). Since this is greater than 43% of the whole dataset, our survey would be limited in power. However, as seen in the above matrix, auxiliary variables are complete, and the OLS demonstrates a strong predictive power between population and delinquent drinking water dollars (*p* < 0.01). This is intuitive due to economies of scale. Further, since delinquent dollars (total) are more available, this response variable can be used to predict debt for drinking water (*p*<0.00001). Imputation of these values will decrease variance, but potentially increase the uncertainty of this survey (albeit slightly). The relationship between these two variables is demonstrated in a linear regression.

## Estimate an OLS Regression  
summary(lm(dollars\_del\_dw\_TotR ~ dollars\_del\_acc\_TotR,  
 na.action = na.omit,   
 data = larges))

##   
## Call:  
## lm(formula = dollars\_del\_dw\_TotR ~ dollars\_del\_acc\_TotR, data = larges,   
## na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1653363 -684319 -490164 -67353 17576738   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 782198.824780 248966.417918 3.142 0.00238  
## dollars\_del\_acc\_TotR 0.171066 0.004695 36.432 < 0.0000000000000002  
##   
## (Intercept) \*\*   
## dollars\_del\_acc\_TotR \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2188000 on 77 degrees of freedom  
## (49 observations deleted due to missingness)  
## Multiple R-squared: 0.9452, Adjusted R-squared: 0.9445   
## F-statistic: 1327 on 1 and 77 DF, p-value: < 0.00000000000000022

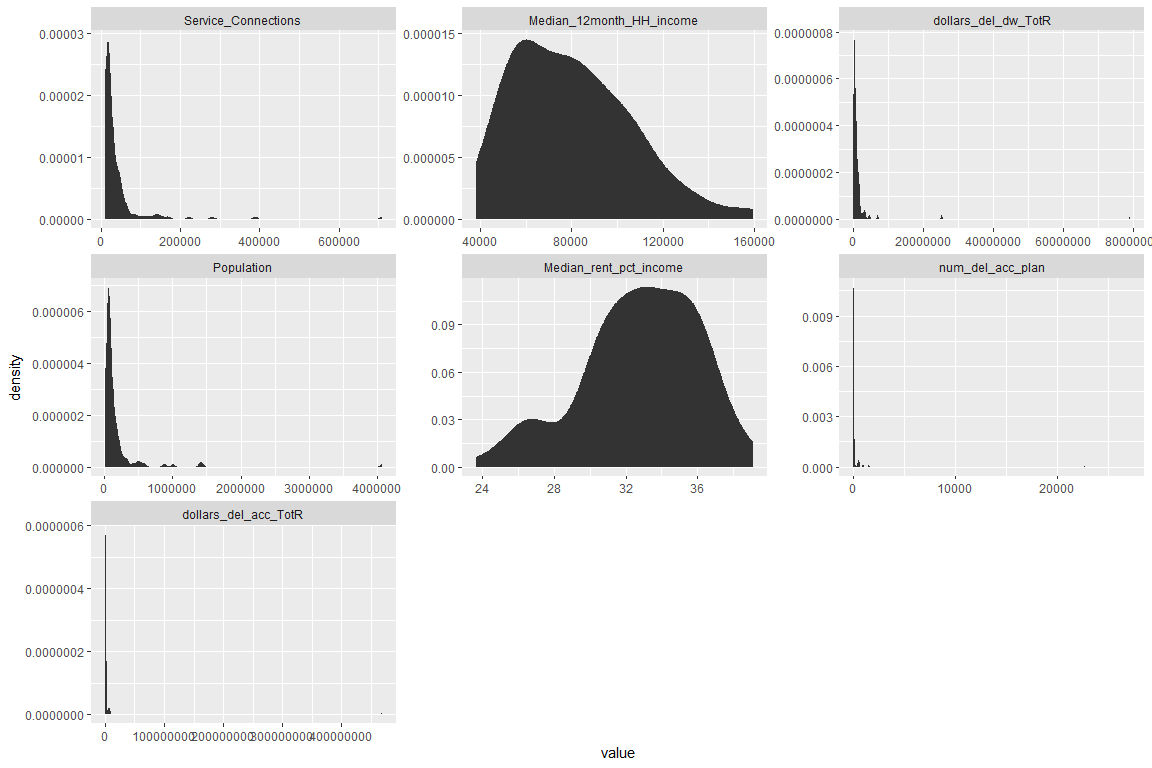
This relationship is plotted below.

#define linear relationship  
larges %>%   
 ggplot(aes(x = log10(dollars\_del\_acc\_TotR), y = log10(dollars\_del\_dw\_TotR))) +  
 geom\_point(aes(color = Fee\_Code)) +  
 geom\_smooth(method = "lm") +  
 stat\_cor(label.y = 7)+ #this means at 35th unit in the y axis, the r squared and p value will be shown  
 stat\_regline\_equation(label.y = 6) + #this means at 30th unit regresion line equation will be shown  
 labs(title = "Linear Relationship between Total Debt and Drinking Water Debt",  
 subtitle = ">10,000 service connection systems")

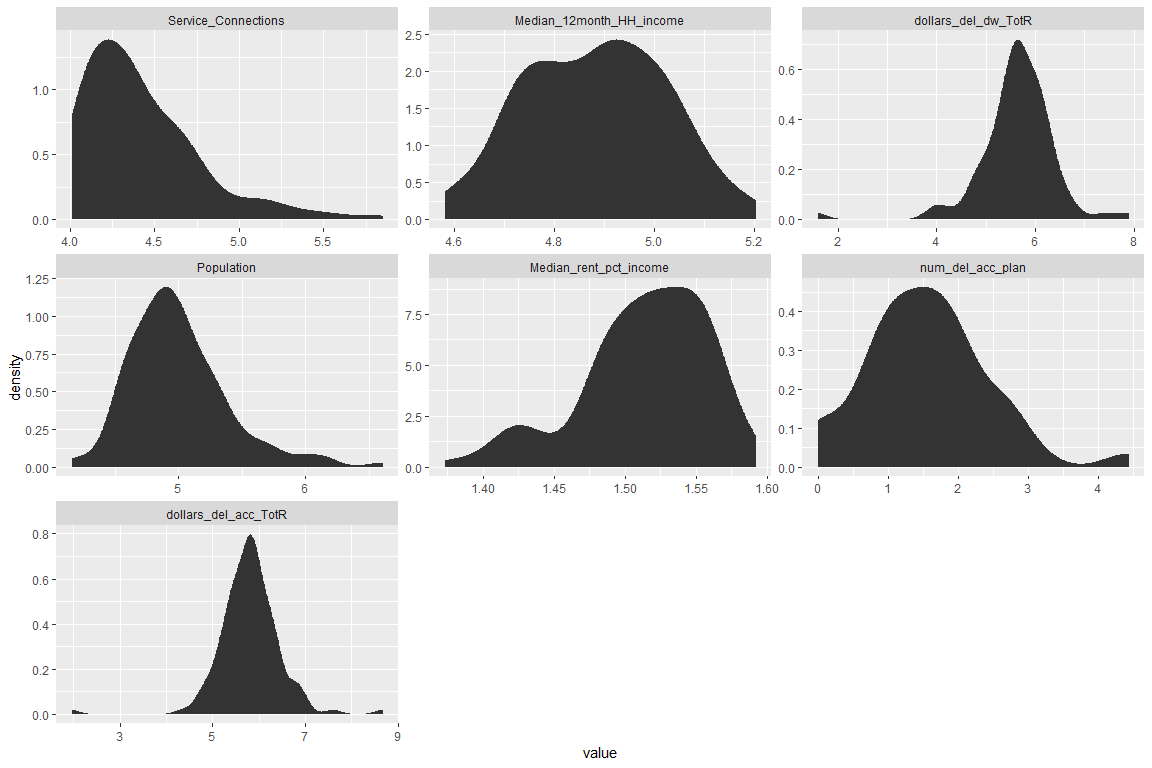


Prior to imputation, the dataset should be prepared such that variables remain in their rawest form, so any categorical factors should be left as so, instead of using the ordinal transformed variable from above. Further, we want to remove variables that have haver than 25% missing values because they may mess up the imputation (no such variables). It’s also important to remove variables that are highly correlated with others so as to stop the imputation working otherwise. Additionally, any extreme outliers should be removed, as they may dramatically impact results.

require(reshape2)  
larges %>%   
 select(PWSID, Service\_Connections, Median\_12month\_HH\_income, dollars\_del\_dw\_TotR, Fee\_Code, Population,  
 Median\_rent\_pct\_income,num\_del\_acc\_plan, dollars\_del\_acc\_TotR) %>%   
 melt() %>% #convert wide to long  
 ggplot(aes(x = value)) +   
 stat\_density() +   
 facet\_wrap(~variable, scales = "free")

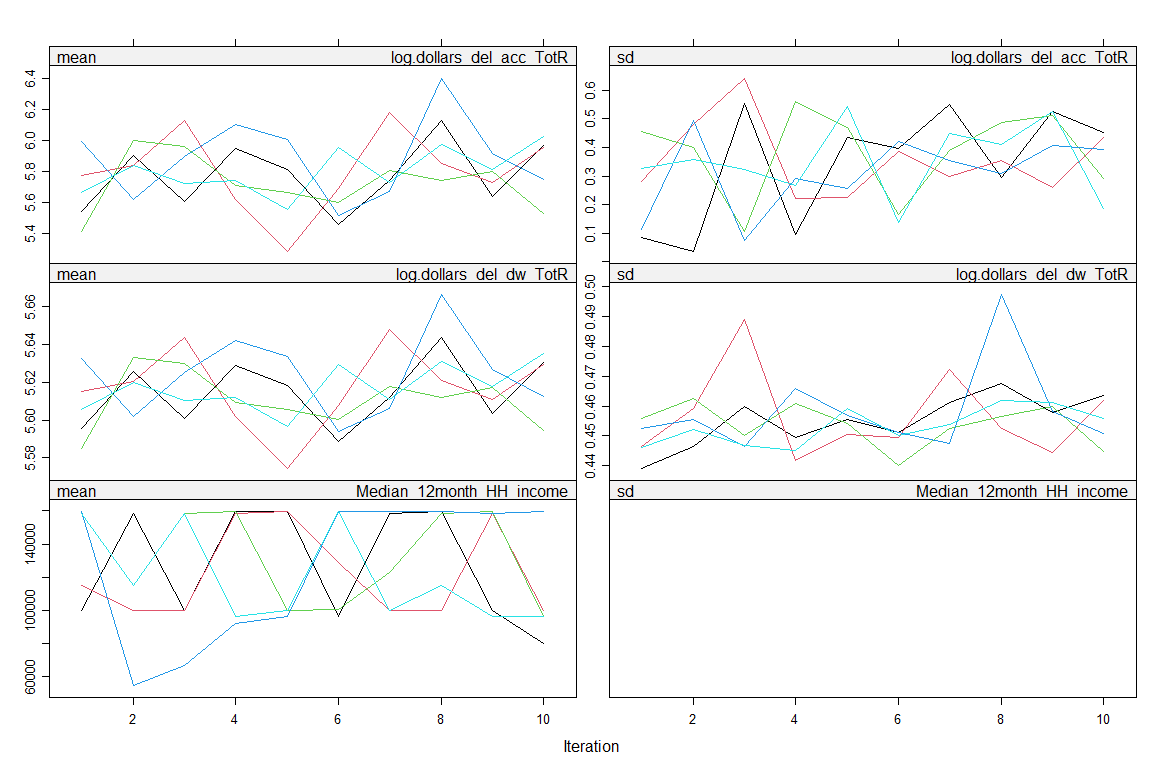
 Everything looks more or less realistic, however several variables would benefit from log10 transformation for visualizing and possibly for statistics.

require(reshape2)  
larges %>%   
 select(PWSID, Service\_Connections, Median\_12month\_HH\_income, dollars\_del\_dw\_TotR, Fee\_Code, Population,  
 Median\_rent\_pct\_income,num\_del\_acc\_plan, dollars\_del\_acc\_TotR) %>%   
 melt() %>% #convert wide to long  
 mutate\_if(~is.numeric(.) && (.) > 0, log10) %>%   
 ggplot(aes(x = value)) +   
 stat\_density() +   
 facet\_wrap(~variable, scales = "free")

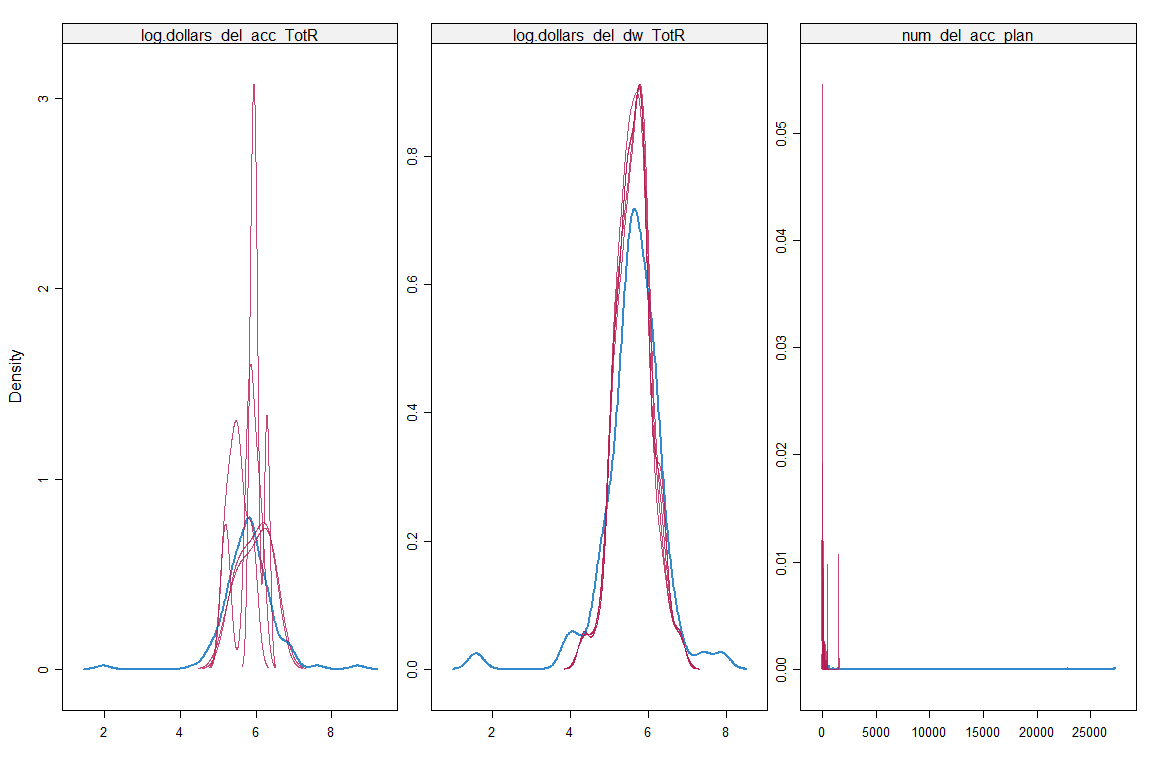
 Now distributions look more or less “normal.”

Based on the strong linear relationship between significant variables a simple predictive mean matching model will be used, with a manual correction for one-way prediction between total debt and drinking water debt, to avoid circularity.

ini <- mice(larges.simple, maxit=0, print=F)  
meth<- ini$meth  
pred <- ini$pred  
pred[c("log.dollars\_del\_acc\_TotR"), "log.dollars\_del\_dw\_TotR"] <- 0 #exclude drinking water debt as a predictor of total debt to avoid circularity  
meth["log.dollars\_del\_dw\_TotR"]<- "~ I(log.dollars\_del\_acc\_TotR)" #define predictor for drinking water debt  
passive.imp <- mice(larges.simple, meth=meth, pred=pred, maxit=10, seed=123, print=F)  
plot(passive.imp, c("log.dollars\_del\_acc\_TotR","log.dollars\_del\_dw\_TotR", "Median\_12month\_HH\_income"))

 The above figure demonstrates good convergence for imputed log debt, with some circularity observed for median 12 month household income. These imputed distributions are plotted below.

densityplot(passive.imp)

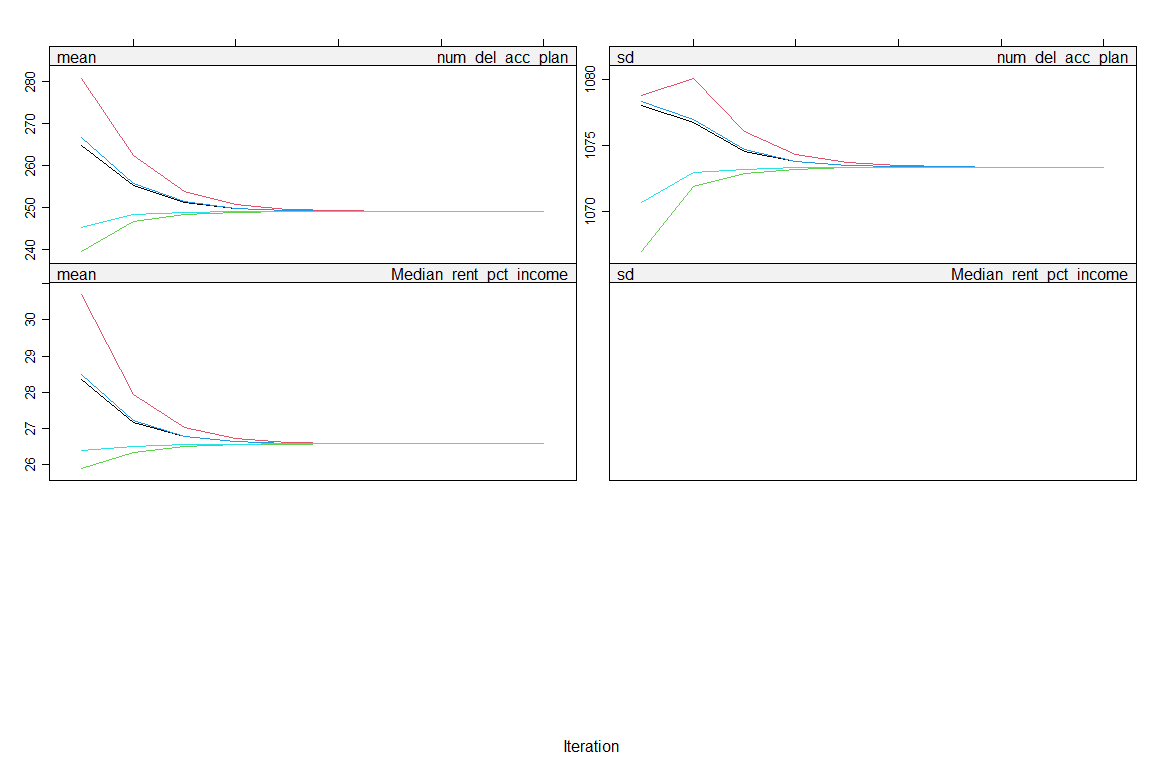
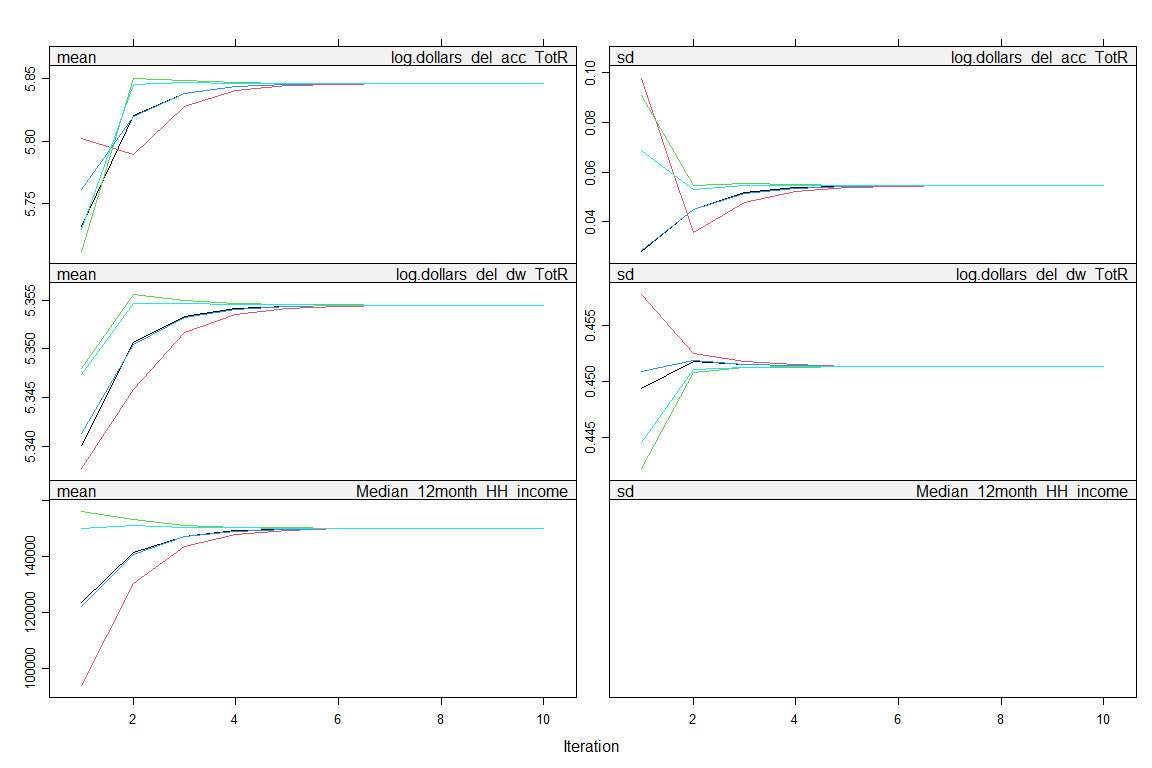


#, ~log.dollars\_del\_acc\_TotR + log.dollars\_del\_dw\_TotR)

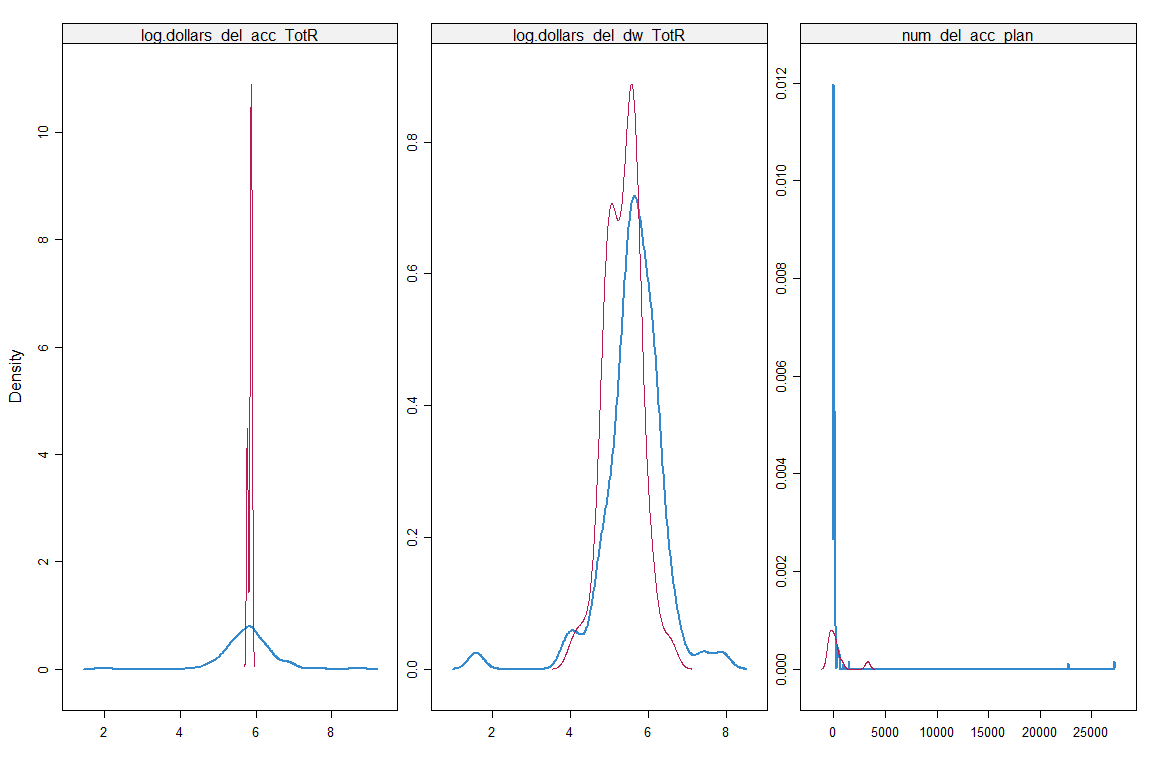
This method was accurate for imputed deqlinquent drinking water debt, but was very poor for other parameters. A mixed model may perform better.

Bayesian linear regression will be tested as an imputation method.

normpredict.imp <- mice(larges.simple, meth="norm.predict", pred=pred, maxit=10, seed=123, print=F)  
plot(normpredict.imp)

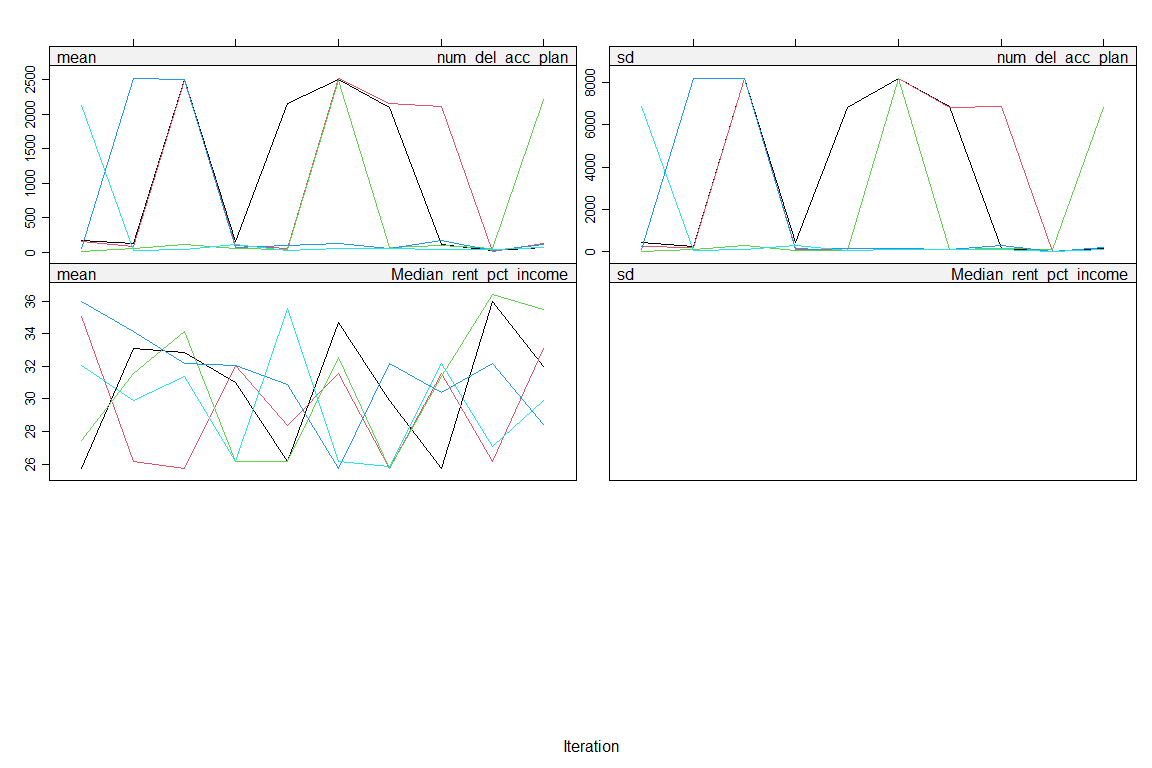
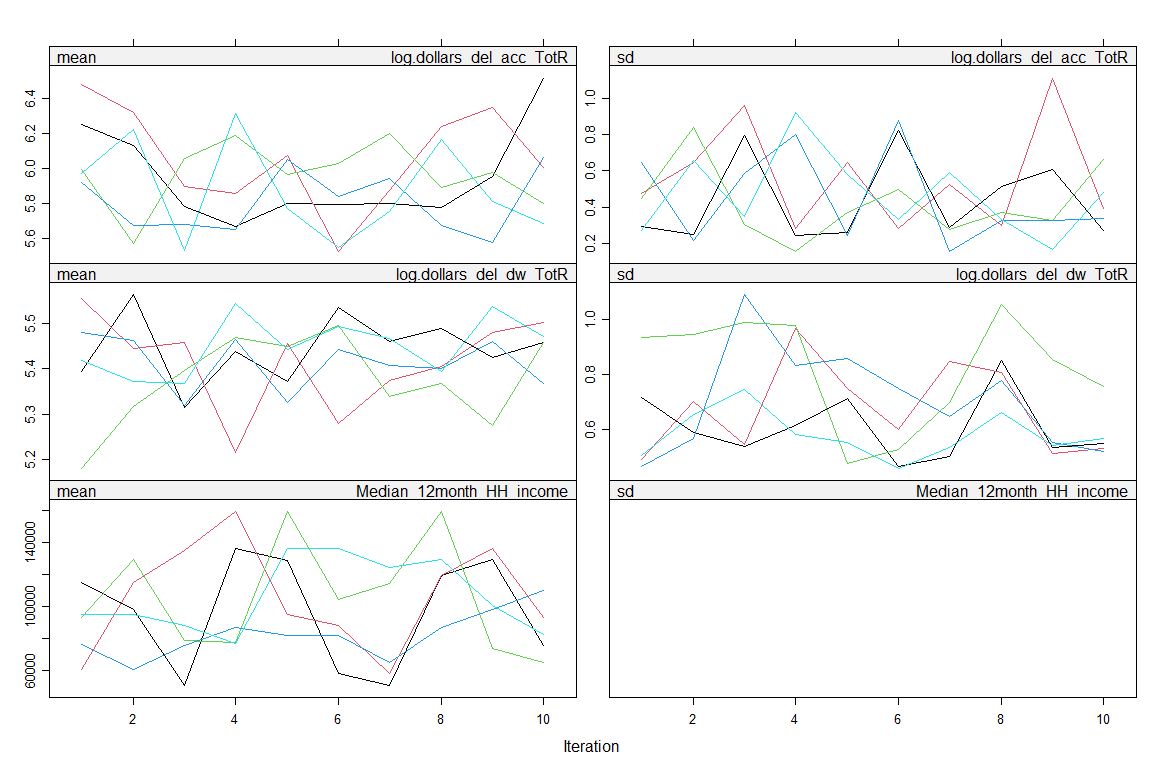
 Bayesian linear regression causes extreme convergence, as the model is quite simple.

densityplot(normpredict.imp)#, ~log.dollars\_del\_dw\_TotR)

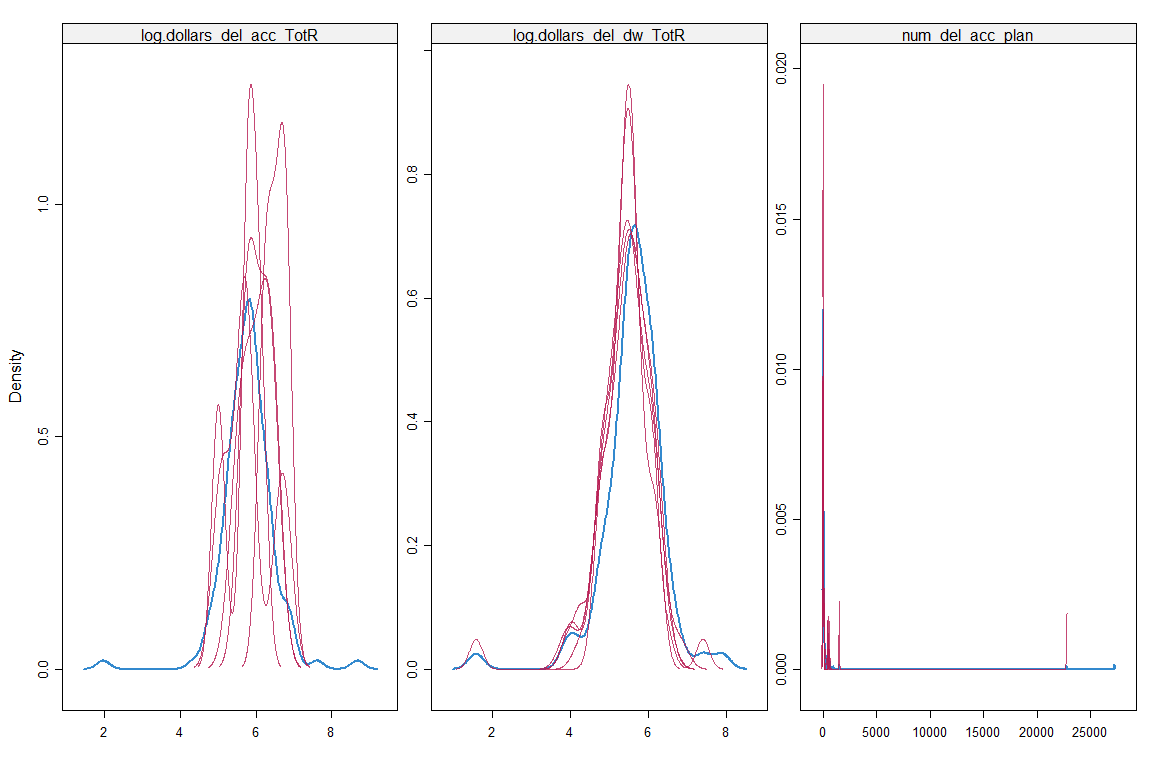


Bayesian linear regression performs quite poorly for predicting total debt. A random forest model may perform better.

rf.imp <- mice(larges.simple, meth="rf", pred=pred, maxit=10, seed=123, print=F)  
plot(rf.imp)

 Bayesian linear regression causes extreme convergence, as the model is quite simple.

densityplot(rf.imp)#, ~log.dollars\_del\_dw\_TotR)

 Random forest modeling seems to have improved the prediction of drinking water debt, and seems to have smoothed.

Observed data is plotted in blue, and imputed is in red. The figure graphs the data values of chl before and after imputation. The figure indicates that the distributions of the imputed and the observed values are similar. Finally, we need to run the regression on each of the datasets and pool the estimates together to get average regression coefficients and correct standard errors.

#First, turn the datasets into long format  
larges.simple\_long <- mice::complete(rf.imp, action="long", include = TRUE)  
  
# Convert back to mids type - mice can work with this type  
larges.simple\_long\_mids <- as.mids(larges.simple\_long)  
# Regression   
  
fitimp <- with(larges.simple\_long\_mids,  
 lm(log.dollars\_del\_dw\_TotR ~ Service\_Connections + Median\_12month\_HH\_income + Fee\_Code + Population +  
 Median\_rent\_pct\_income + num\_del\_acc\_plan + log.dollars\_del\_acc\_TotR,  
 na.action = na.omit, data = larges))  
  
kable(summary(pool(fitimp)),  
 caption = "Ordinal Least Squares Regression for Random Forest Model (Large Systems)",  
 digits = 5)

Ordinal Least Squares Regression for Random Forest Model (Large Systems)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | df | p.value |
| (Intercept) | -1.32112 | 0.70505 | -1.87378 | 62.08333 | 0.06567 |
| Service\_Connections | 0.00000 | 0.00000 | 1.29777 | 62.08333 | 0.19917 |
| Median\_12month\_HH\_income | 0.00000 | 0.00000 | 0.46621 | 62.08333 | 0.64270 |
| Fee\_CodeDAVCL | -0.14126 | 0.14768 | -0.95647 | 62.08333 | 0.34254 |
| Population | 0.00000 | 0.00000 | -1.34372 | 62.08333 | 0.18393 |
| Median\_rent\_pct\_income | 0.03297 | 0.01209 | 2.72715 | 62.08333 | 0.00829 |
| num\_del\_acc\_plan | -0.00001 | 0.00001 | -1.05396 | 62.08333 | 0.29599 |
| log.dollars\_del\_acc\_TotR | 0.98095 | 0.08554 | 11.46821 | 62.08333 | 0.00000 |

Pooled coefficients and p-values from the imputed datasets are compared with the original dataset to see if any trends are altered (either become more pronounced or less). They are similiar to the listwise-deletion technique.

## Class: mipo m = 5   
## term m estimate ubar b  
## 1 (Intercept) 5 -1.3211176945632 0.4971007279722560179 0  
## 2 Service\_Connections 5 0.0000034907419 0.0000000000072350692 0  
## 3 Median\_12month\_HH\_income 5 0.0000007264613 0.0000000000024280976 0  
## 4 Fee\_CodeDAVCL 5 -0.1412559248547 0.0218107779407125876 0  
## 5 Population 5 -0.0000006305195 0.0000000000002201793 0  
## 6 Median\_rent\_pct\_income 5 0.0329663779491 0.0001461250905609044 0  
## 7 num\_del\_acc\_plan 5 -0.0000125580545 0.0000000001419705942 0  
## 8 log.dollars\_del\_acc\_TotR 5 0.9809491774576 0.0073164723508763133 0  
## t dfcom df riv lambda fmi  
## 1 0.4971007279722560179 64 62.08333 0 0 0.03072983  
## 2 0.0000000000072350692 64 62.08333 0 0 0.03072983  
## 3 0.0000000000024280976 64 62.08333 0 0 0.03072983  
## 4 0.0218107779407125876 64 62.08333 0 0 0.03072983  
## 5 0.0000000000002201793 64 62.08333 0 0 0.03072983  
## 6 0.0001461250905609044 64 62.08333 0 0 0.03072983  
## 7 0.0000000001419705942 64 62.08333 0 0 0.03072983  
## 8 0.0073164723508763133 64 62.08333 0 0 0.03072983

Comparisons with the Bayesian linear regression model are below.

#First, turn the datasets into long format  
larges.simple\_long\_1 <- mice::complete(normpredict.imp, action="long", include = TRUE)  
  
# Convert back to mids type - mice can work with this type  
larges.simple\_long\_mids\_1 <- as.mids(larges.simple\_long\_1)  
# Regression   
  
fitimp\_1 <- with(larges.simple\_long\_mids\_1,  
 lm(log.dollars\_del\_dw\_TotR ~ Service\_Connections + Median\_12month\_HH\_income + Fee\_Code + Population +  
 Median\_rent\_pct\_income + num\_del\_acc\_plan + log.dollars\_del\_acc\_TotR,  
 na.action = na.omit, data = larges))  
  
kable(summary(pool(fitimp\_1)),  
 caption = "Ordinal Least Squares Regression for Bayesian linear regression Imputation Model (Large Systems)",  
 digits = 5)

Ordinal Least Squares Regression for Bayesian linear regression Imputation Model (Large Systems)

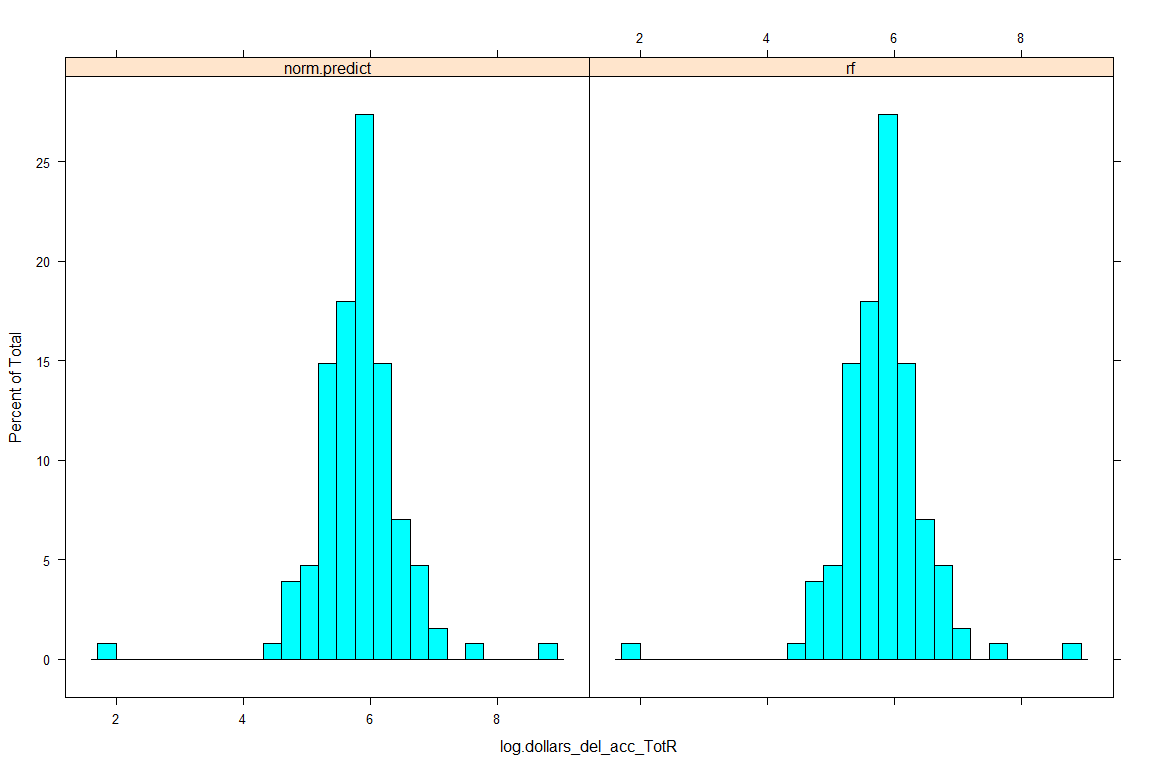
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | df | p.value |
| (Intercept) | -1.32112 | 0.70505 | -1.87378 | 62.08333 | 0.06567 |
| Service\_Connections | 0.00000 | 0.00000 | 1.29777 | 62.08333 | 0.19917 |
| Median\_12month\_HH\_income | 0.00000 | 0.00000 | 0.46621 | 62.08333 | 0.64270 |
| Fee\_CodeDAVCL | -0.14126 | 0.14768 | -0.95647 | 62.08333 | 0.34254 |
| Population | 0.00000 | 0.00000 | -1.34372 | 62.08333 | 0.18393 |
| Median\_rent\_pct\_income | 0.03297 | 0.01209 | 2.72715 | 62.08333 | 0.00829 |
| num\_del\_acc\_plan | -0.00001 | 0.00001 | -1.05396 | 62.08333 | 0.29599 |
| log.dollars\_del\_acc\_TotR | 0.98095 | 0.08554 | 11.46821 | 62.08333 | 0.00000 |

## Class: mipo m = 5   
## term m estimate ubar b  
## 1 (Intercept) 5 -1.3211176945632 0.4971007279722560179 0  
## 2 Service\_Connections 5 0.0000034907419 0.0000000000072350692 0  
## 3 Median\_12month\_HH\_income 5 0.0000007264613 0.0000000000024280976 0  
## 4 Fee\_CodeDAVCL 5 -0.1412559248547 0.0218107779407125876 0  
## 5 Population 5 -0.0000006305195 0.0000000000002201793 0  
## 6 Median\_rent\_pct\_income 5 0.0329663779491 0.0001461250905609044 0  
## 7 num\_del\_acc\_plan 5 -0.0000125580545 0.0000000001419705942 0  
## 8 log.dollars\_del\_acc\_TotR 5 0.9809491774576 0.0073164723508763133 0  
## t dfcom df riv lambda fmi  
## 1 0.4971007279722560179 64 62.08333 0 0 0.03072983  
## 2 0.0000000000072350692 64 62.08333 0 0 0.03072983  
## 3 0.0000000000024280976 64 62.08333 0 0 0.03072983  
## 4 0.0218107779407125876 64 62.08333 0 0 0.03072983  
## 5 0.0000000000002201793 64 62.08333 0 0 0.03072983  
## 6 0.0001461250905609044 64 62.08333 0 0 0.03072983  
## 7 0.0000000001419705942 64 62.08333 0 0 0.03072983  
## 8 0.0073164723508763133 64 62.08333 0 0 0.03072983

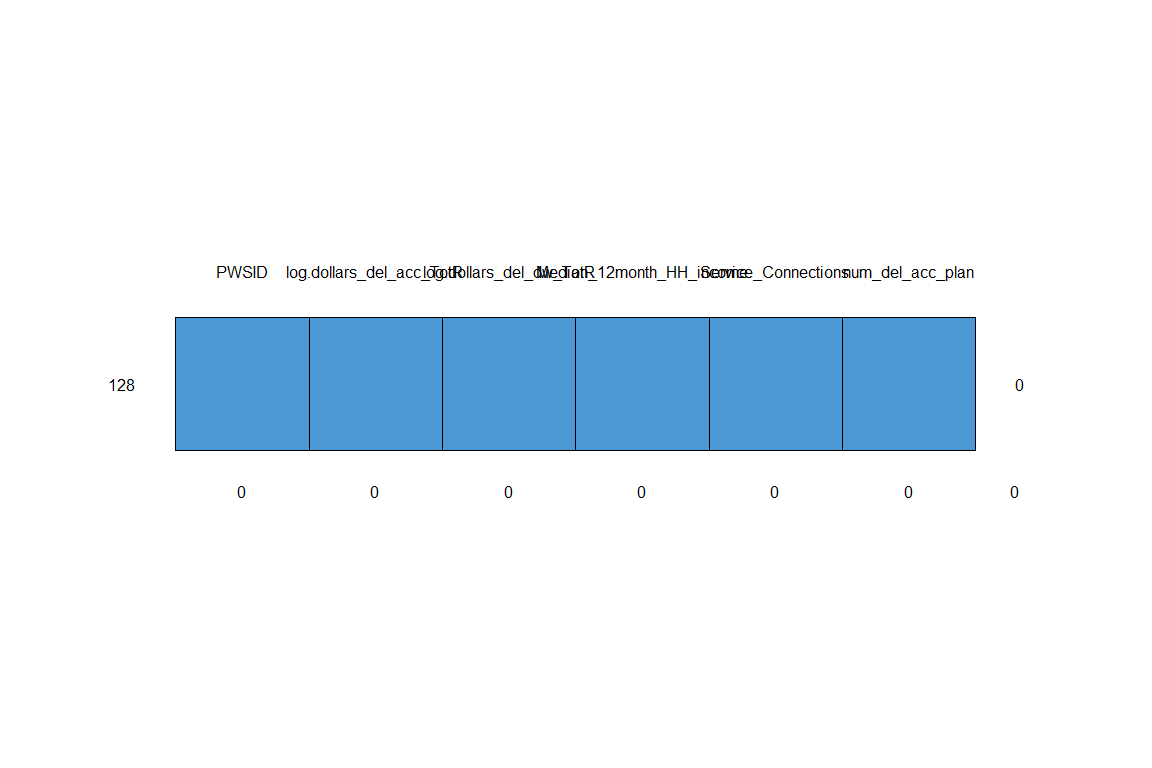
These two models differ dramatically, with the Bayesian linear regression having significantly lower variance due to nonresponse (lambdi) and fraction of information missing due to nonresponse (fmi), with a strong linear relationship retained between total delinquent debt and drinking water debt.

Let’s further compare the Bayesian linear regression and random forest methods for log.dollars\_del\_acc\_TotR.

log.dollars\_del\_acc\_TotR <- c(complete(passive.imp)$log.dollars\_del\_acc\_TotR, complete(passive.imp)$log.dollars\_del\_acc\_TotR)  
method <- rep(c("norm.predict", "rf"), each = nrow(larges.simple))  
log.dollars\_del\_acc\_TotR\_m <- data.frame(log.dollars\_del\_acc\_TotR = log.dollars\_del\_acc\_TotR, method = method)  
#plot histogram  
histogram( ~log.dollars\_del\_acc\_TotR | method, data = log.dollars\_del\_acc\_TotR\_m, nint = 25)

 Here we see that these two method do not seemingly differ in prediction. Since the random forest performed better for the total water debt, and performed approximately the same for drinking water debt, we will use the random forest.

## /\ /\  
## { `---' }  
## { O O }  
## ==> V <== No need for mice. This data set is completely observed.  
## \ \|/ /  
## `-----'



## PWSID log.dollars\_del\_acc\_TotR log.dollars\_del\_dw\_TotR  
## 128 1 1 1  
## 0 0 0  
## Median\_12month\_HH\_income Service\_Connections num\_del\_acc\_plan   
## 128 1 1 1 0  
## 0 0 0 0

The table below compares summary statistics for the imputed variables with those in the original dataset for large systems.

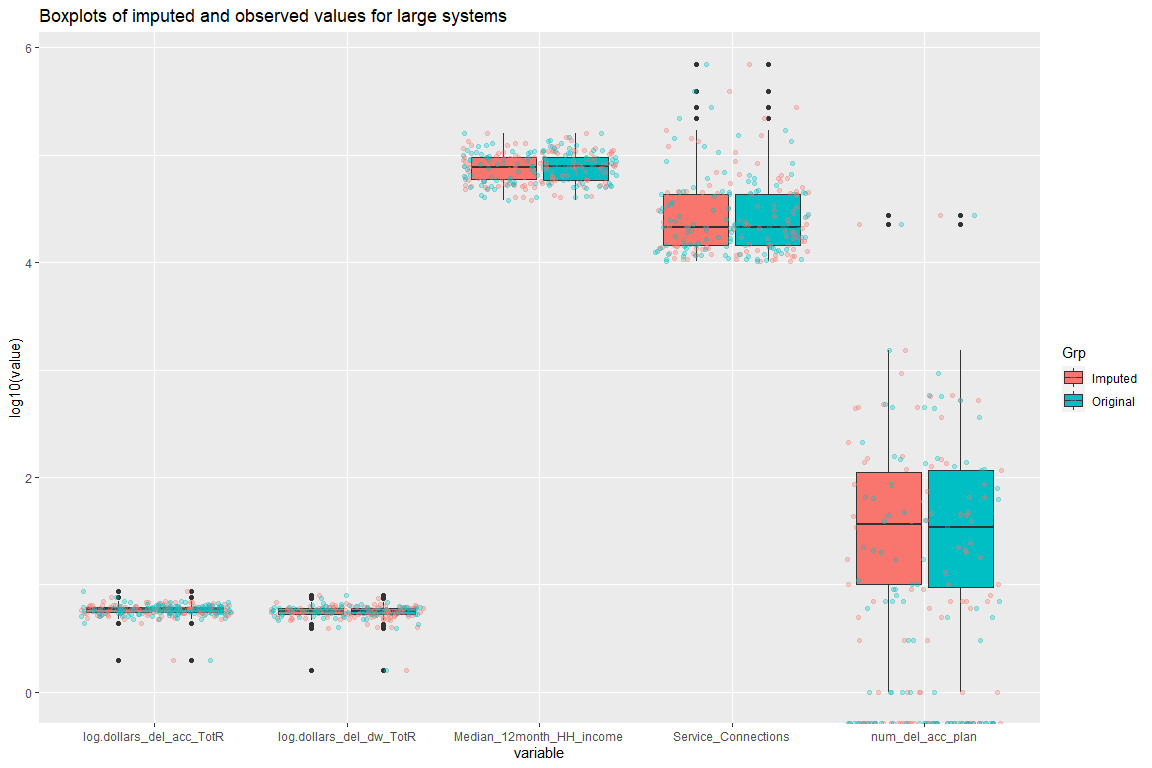
larges.simple.imputed %>%   
 mutate(Grp = "Imputed") %>%  
 bind\_rows(mutate(larges, Grp = "Original")) %>%  
 select(Grp, log.dollars\_del\_acc\_TotR, log.dollars\_del\_dw\_TotR,   
 Median\_12month\_HH\_income, Service\_Connections, num\_del\_acc\_plan) %>%  
 group\_by(Grp) %>%   
 drop\_na() %>%   
 summarize\_all(list(mean = "mean", median = "median", sd = "sd")) %>%   
 t() %>%   
 kable(caption = "Summary Statistics for Imputed and Observed Values for Large Systems",  
 digits = 5)

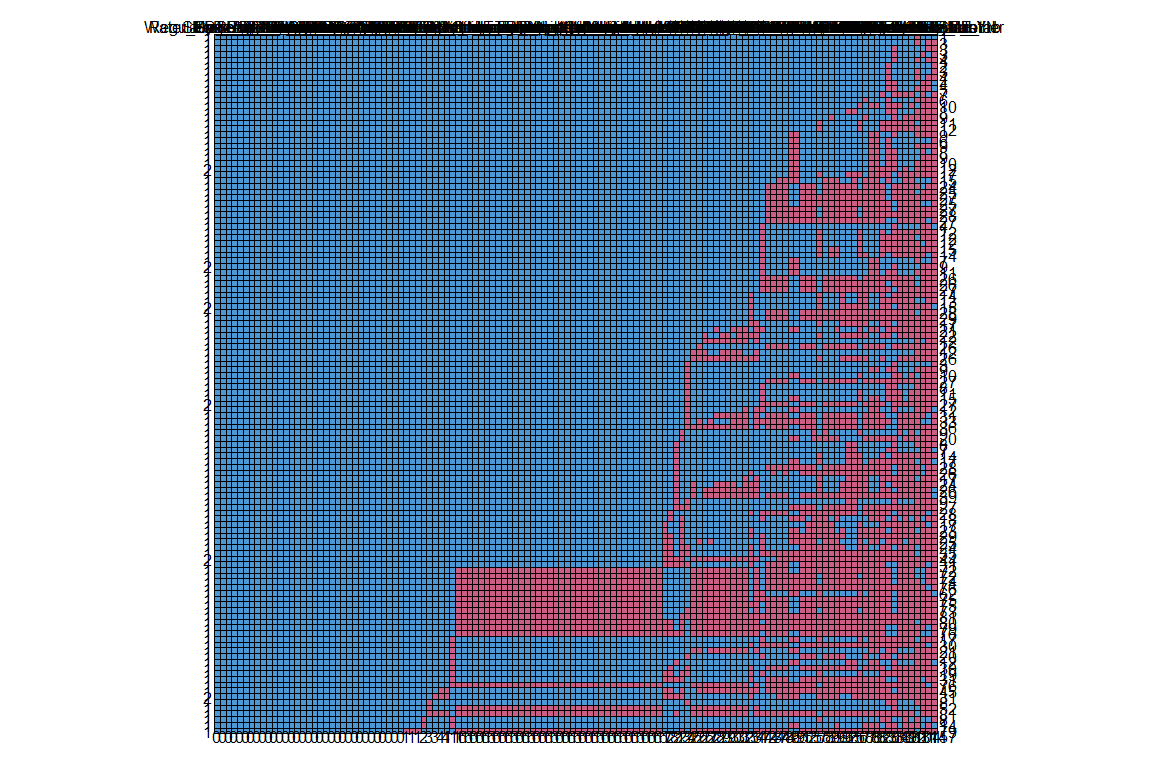
Summary Statistics for Imputed and Observed Values for Large Systems

|  |  |  |
| --- | --- | --- |
| Grp | Imputed | Original |
| log.dollars\_del\_acc\_TotR\_mean | 5.821113 | 5.933754 |
| log.dollars\_del\_dw\_TotR\_mean | 5.562068 | 5.660199 |
| Median\_12month\_HH\_income\_mean | 80258.45 | 82011.35 |
| Service\_Connections\_mean | 43117.38 | 51509.12 |
| num\_del\_acc\_plan\_mean | 467.8906 | 796.8194 |
| log.dollars\_del\_acc\_TotR\_median | 5.830793 | 5.875180 |
| log.dollars\_del\_dw\_TotR\_median | 5.599236 | 5.643958 |
| Median\_12month\_HH\_income\_median | 77504.89 | 77530.96 |
| Service\_Connections\_median | 21384 | 24093 |
| num\_del\_acc\_plan\_median | 10 | 12 |
| log.dollars\_del\_acc\_TotR\_sd | 0.6847560 | 0.6227642 |
| log.dollars\_del\_dw\_TotR\_sd | 0.7118188 | 0.6647100 |
| Median\_12month\_HH\_income\_sd | 26085.59 | 27505.81 |
| Service\_Connections\_sd | 78062.98 | 92506.14 |
| num\_del\_acc\_plan\_sd | 3121.619 | 4143.842 |

Summary stats are not significantly altered by imputation. The mean and median number of service connections and number of deliquent account plans are beyond 10% different. Since service connections is an auxiliary variable this will be dropped from the imputed dataset. Imputation was not optimized for count data (number of delinquent accounts), and will not be used. Summary stats are plotted below.

require(reshape2)  
#join data for plotting  
larges.simple.imputed %>%   
 mutate(Grp = "Imputed") %>%  
 bind\_rows(mutate(larges, Grp = "Original")) %>%  
 select(Grp, log.dollars\_del\_acc\_TotR, log.dollars\_del\_dw\_TotR,   
 Median\_12month\_HH\_income, Service\_Connections, num\_del\_acc\_plan) %>%  
 melt() %>% #transforms values and variables to long format  
 ggplot(aes(x = variable, y = log10(value))) +  
 geom\_boxplot(aes(fill = Grp))+  
 geom\_jitter(aes(color = Grp), alpha = 0.3) +  
 labs(title = "Boxplots of imputed and observed values for large systems")





# Survey Weighting

Survey weights are a key component to producing population estimates. As described in Valliant et al. (2018), an estimated total has the form where is a response provided by the *i*th sample member and is the corresponding analysis weight. Without the use of weights, estimates from survey data may simply reflecy nuances of a particular sample, containing significant levels of bias. The series of weighting that will be applied to this survey data include computation of base weights, nonresponse adjustments, and use of auxiliary data to reduce variances (i.e. calibration).

## Base Weights

The first basic step to make reliable extrapolations to the population is to generate base weights. Note that this is one of three weight adjustments that will be applied,including non-response and calibration adjustments to weights.

The first step in weighing is taking into account the different probabilities of being sampled that respondents may have simply based on proportions in the population. This is also known as generating *base weights*. Note that this survey is a random stratified sampling design without replacement. We will calculate inclusion probabilities for each strata for the small systems (tags A, B, C, D). Large systems (>10,000 service connections) will be treated as a single strata for the calculation of base weights, and will be handled separated.

### Small Systems

probSumm <- allSmalls %>%   
 filter(voluntary != "y") %>% #remove volunteers  
 mutate(sample = case\_when(requested == "y" ~ 1,requested == "n" ~ 0)) %>% #provide integer tag for sampled  
 group\_by(tag) %>%   
 summarize(N = n(),samples = sum(sample), respondedTotal = sum(response, na.rm = TRUE), prob = samples/N, base.weight = 1/prob,   
 sum.base.weight = base.weight \* samples, fpc = samples/N) %>%   
 drop\_na  
#join the probability to each sample  
allSmalls <- right\_join(allSmalls, probSumm, by = "tag")  
allSmalls.requested.responded <- right\_join(allSmalls.requested.responded, probSumm, by = "tag")  
#print  
kable(probSumm,  
 caption = "Base Weights for Small Water System Survey",  
 digits = 4)

Base Weights for Small Water System Survey

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| tag | N | samples | respondedTotal | prob | base.weight | sum.base.weight | fpc |
| Bin A | 1914 | 228 | 154 | 0.1191 | 8.3947 | 1914 | 0.1191 |
| Bin B | 259 | 135 | 99 | 0.5212 | 1.9185 | 259 | 0.5212 |
| Bin C | 107 | 84 | 69 | 0.7850 | 1.2738 | 107 | 0.7850 |
| Bin D | 76 | 69 | 55 | 0.9079 | 1.1014 | 76 | 0.9079 |

We can see here that the probability of being chosen within each strata is slightly different, as planned by the stratified sampling design.

## # A tibble: 4 x 4  
## tag min.probability mean.probability max.probability  
## <fct> <dbl> <dbl> <dbl>  
## 1 Bin A 0.119 0.119 0.119  
## 2 Bin B 0.521 0.521 0.521  
## 3 Bin C 0.785 0.785 0.785  
## 4 Bin D 0.908 0.908 0.908

There may be underlying factors that make some types of water systems more or less likely to be sampled. For instance, let’s look at the mean probabilities by fee code.

allSmalls %>%  
 filter(!is.na(prob)) %>%  
 group\_by(tag, Fee\_Code) %>%  
 summarise(n = n(),  
 mean.prob.percentage = mean(prob, na.rm = T)\*100) %>%  
 arrange(desc(mean.prob.percentage)) %>%   
 head() %>%   
 kable(caption = "Mean Probability for Inclusion in Survey by Fee Code")

Mean Probability for Inclusion in Survey by Fee Code

|  |  |  |  |
| --- | --- | --- | --- |
| tag | Fee\_Code | n | mean.prob.percentage |
| Bin D | C1 | 64 | 90.78947 |
| Bin D | DAVCL | 12 | 90.78947 |
| Bin C | C1 | 83 | 78.50467 |
| Bin C | DAVCL | 25 | 78.50467 |
| Bin B | C1 | 176 | 52.12355 |
| Bin B | DAVCL | 87 | 52.12355 |

Since some fee codes roughly fit within sampling bins (strata), it makes sense that we see discrete groupings.

## # A tibble: 6 x 4  
## # Groups: months\_before\_assist [6]  
## months\_before\_assist tag n mean.prob.percentage  
## <fct> <fct> <int> <dbl>  
## 1 A Bin D 1 90.8  
## 2 B Bin D 1 90.8  
## 3 C Bin D 2 90.8  
## 4 D Bin D 3 90.8  
## 5 E Bin D 16 90.8  
## 6 F Bin D 30 90.8

To correct for these differential probabilities, we must design weights (sometimes called base weights) so that our sample does not over- or under-represent relevant groups. The design weights are equal to the inverse of the probability of inclusion to the sample. Therefore, the design weight (d0) of a respondent (i) will be equal to: where is the probability of that unit being included in the sampling.

A simple interpretation of design weights is ‘the number of units in our population that each unit in our sample represents’. There is a simple but important test that we should perform after computing design weights. **The sum of all design weights should be equal to the total number of units in our population.**

Now to ensure design weights sum up to the entire population from which each population (bin) was drawn, a table is drawn below.

allSmalls %>%   
 mutate(sample = case\_when(requested == "y" ~ 1,requested == "n" ~ 0)) %>% #provide integer tag for sampled  
 group\_by(tag) %>%   
 summarize(n = n(),samples = sum(sample), prob = samples/n, base.weight = 1/prob,   
 sum.base.weight = base.weight\* samples) %>%   
 drop\_na %>%   
 kable()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| tag | n | samples | prob | base.weight | sum.base.weight |
| Bin A | 1944 | 228 | 0.1172840 | 8.526316 | 1944 |
| Bin B | 267 | 135 | 0.5056180 | 1.977778 | 267 |
| Bin C | 108 | 84 | 0.7777778 | 1.285714 | 108 |
| Bin D | 76 | 69 | 0.9078947 | 1.101449 | 76 |

### Large Systems

Below these steps are briefly repeated for large systems (greater than 10,000 service connections).

probSumm <- allLarges %>%   
 mutate(sample = case\_when(responded == "y" ~ 1,responded == "n" ~ 0)) %>% #provide integer tag for sampled  
 group\_by(tag) %>%   
 summarize(N = n(),samples = sum(sample), prob = samples/N, base.weight = 1/prob,  
 sum.base.weight = base.weight \* samples, fpc = samples/N)  
#join the probability to each sample  
allLarges <- right\_join(allLarges, probSumm, by = "tag")  
#print  
kable(probSumm,  
 caption = "Probability for inclusion in sample (large systems)")

Probability for inclusion in sample (large systems)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tag | N | samples | prob | base.weight | sum.base.weight | fpc |
| E | 223 | 128 | 0.573991 | 1.742188 | 223 | 0.573991 |

## Response Propensity Adjustment

Now that we have designed our basic weights, we now need to account for differences in the propensity to respond. It’s possible that certain profiles (e.g. lower income communities) had different propensities to respond than another profile (e.g. higher income communities). We could then imagine that the characteristics of both profiles were associated with reponse variables that we collected in this survey (e.g. water systems with lower income having more delinquent accounts than higher income systems). As we then would have a larger proportion of higher income/fewer delinquent accounts in our sample, our analyses would be biased.

### Small Systems

First let’s see the response rate for each of the bins for the small systems.

allSmalls %>% group\_by(tag) %>% filter(requested == "y") %>%   
 summarize(completeness = sum(response)/n() \* 100 , completed = sum(response), total = n()) %>%   
 kable(caption = "Response Proportion by Sampling Strata")

Response Proportion by Sampling Strata

|  |  |  |  |
| --- | --- | --- | --- |
| tag | completeness | completed | total |
| Bin A | 67.54386 | 154 | 228 |
| Bin B | 73.33333 | 99 | 135 |
| Bin C | 82.14286 | 69 | 84 |
| Bin D | 79.71014 | 55 | 69 |

Computing the probability of replying to this survey is challenging because we can not directly observe the probability of replying to the survey, therefore we need to estimate it. This may be done using information which we know for both respondent and non-respondent units. The most reliable (albeit most complicated) of calculating non-response probabilities is through predictive modelling.

[Valliant et al (2013)](https://link.springer.com/book/10.1007%2F978-1-4614-6449-5) recommend estimating the response propensities and then grouping them in classes. The grouping step should avoid extreme weights. One way of estimating the response propensities is using logistic regression. This logistic regression should be unweighted. We will use a general linear model function to predict non-responses with hypothesized response variables for which we have data, including service connections, population, fee code, and regulating agency.

#trim data to those in sample list with auxiliary data  
allSmalls.requested.response.model <- allSmalls %>%   
 filter(requested == "y") %>%   
 select(response, Service\_Connections, Population, CES\_3.0\_Score, Median\_12month\_HH\_income,  
 Median\_rent\_pct\_income) %>%   
 drop\_na()  
  
#prepare formula  
formula.resp <- as.formula("response ~ Service\_Connections + Population + CES\_3.0\_Score + Median\_12month\_HH\_income + Median\_rent\_pct\_income")  
options(na.action = 'na.pass')  
  
#format as matrix  
x.matrix <- model.matrix(formula.resp , data = allSmalls.requested.response.model)[, -1]  
  
#fit binomial model  
log.reg.m <- glm(formula.resp, data = allSmalls.requested.response.model,family = "binomial")  
  
coef.response.log <- coef(log.reg.m)  
predicted.log <- log.reg.m$fitted.values  
non.responsepredicted.log <- predicted.log  
  
predicted.log %>% head() %>% kable(caption = "First Six Bins of Estimated response propensities from Linear Regression for Small Systems")

First Six Bins of Estimated response propensities from Linear Regression for Small Systems

|  |
| --- |
| x |
| 0.8703461 |
| 0.7392626 |
| 0.8655408 |
| 0.9114733 |
| 0.8905100 |
| 0.8148212 |

The above output shows the first six estimated response propensities of our dataset. Since we don’t have pardata for all samples, we are now computing our predictor estimates from a subset of sampled units.

The package *PracTools* has tools to determine propensity classes. Auxiliary variables used to fit response propensity model include service connections, population, 12-month household income, median rent percent income, and fee code.

#redefine all smalls  
allSmalls.requested <- allSmalls %>% filter(requested == "y")  
#fit  
fitResponse <- lm(response ~ Service\_Connections + Population + Median\_12month\_HH\_income + Median\_rent\_pct\_income + CES\_3.0\_Score + Fee\_Code, na.action = na.omit, data = allSmalls.requested)  
summary(fitResponse)

##   
## Call:  
## lm(formula = response ~ Service\_Connections + Population + Median\_12month\_HH\_income +   
## Median\_rent\_pct\_income + CES\_3.0\_Score + Fee\_Code, data = allSmalls.requested,   
## na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.9158 -0.5981 0.2313 0.3057 0.5302   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.6826891707 0.1765457230 3.867 0.000125 \*\*\*  
## Service\_Connections 0.0000418843 0.0000202749 2.066 0.039376 \*   
## Population -0.0000074645 0.0000043509 -1.716 0.086871 .   
## Median\_12month\_HH\_income -0.0000003250 0.0000008822 -0.368 0.712762   
## Median\_rent\_pct\_income 0.0033984795 0.0037709554 0.901 0.367916   
## CES\_3.0\_Score -0.0027866305 0.0017033705 -1.636 0.102499   
## Fee\_CodeDAVCL 0.0842018936 0.0658133935 1.279 0.201367   
## Fee\_CodeDAVCS 0.0039998124 0.1135471346 0.035 0.971914   
## Fee\_CodeSC -0.0390688342 0.0632269529 -0.618 0.536922   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4403 on 485 degrees of freedom  
## (22 observations deleted due to missingness)  
## Multiple R-squared: 0.03724, Adjusted R-squared: 0.02136   
## F-statistic: 2.345 on 8 and 485 DF, p-value: 0.01764

Here we can see that the number of service connections and population are good predictors for response propensity, and possibly fee code (DAVCL). This is convenient, because we know these values for all systems. We will now define classes using this parameters using a logistic regression.

#be sure to use the dataset with non-response data  
out <- pclass(formula = response ~ Service\_Connections + Population + Fee\_Code,  
 data = allSmalls.requested,  
 type = "unwtd", #already have base weights  
 link = "logit",  
 numcl = 4) #classes to create  
kable(table(out$p.class, useNA = "always"),  
 caption = "Response Propensity Classes from Logistic Regression for Small Systems") # ensures no unit has a missing class value

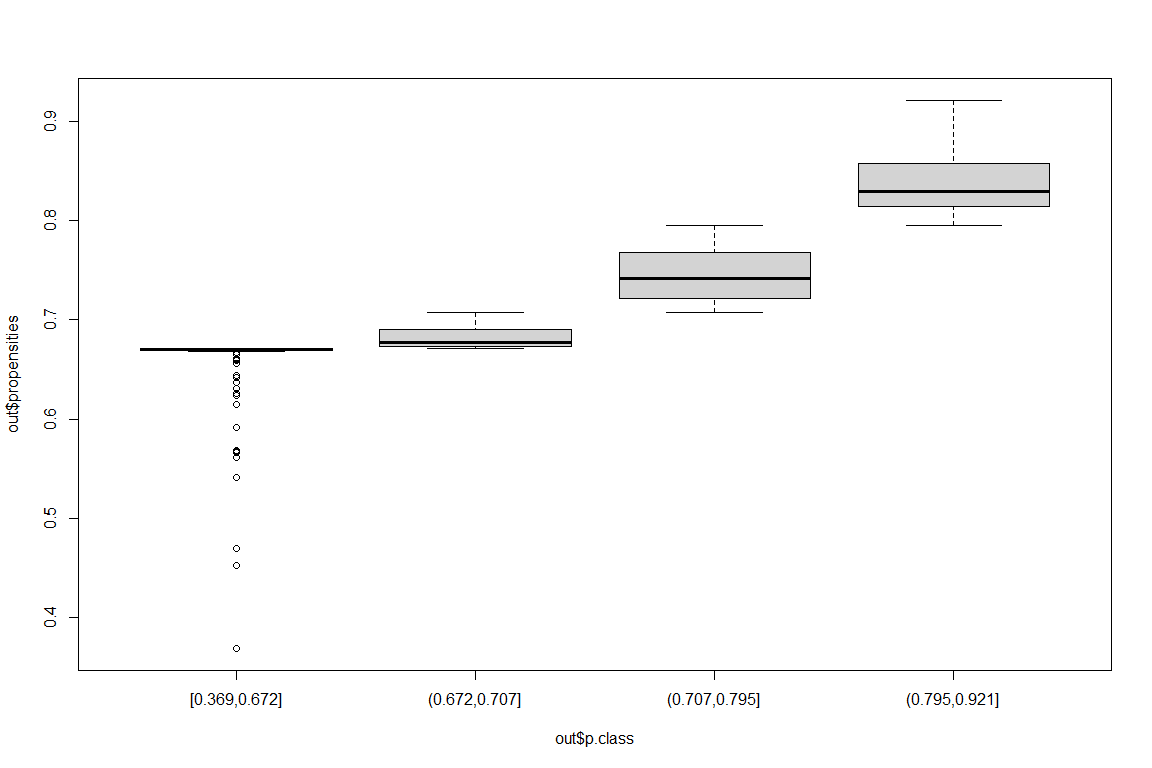
Response Propensity Classes from Logistic Regression for Small Systems

|  |  |
| --- | --- |
| Var1 | Freq |
| [0.369,0.672] | 129 |
| (0.672,0.707] | 129 |
| (0.707,0.795] | 129 |
| (0.795,0.921] | 129 |
| NA | 0 |

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3693 0.6716 0.7073 0.7306 0.7954 0.9209

These propensity classes can be viewed in boxplot form.

boxplot(out$propensities ~ out$p.class)

 We can see discrete propensity classes with rather tight spread for the four uppermost categories. We see a string of outliers in the lowest class. We have several options to adjust our weights for non-response. These include multiplying the input weights by the inverse of cell response propensities.

## mean median weighted  
## [0.369,0.672] 0.6575047 0.6706690 0.6670335  
## (0.672,0.707] 0.6827578 0.6769723 0.6801689  
## (0.707,0.795] 0.7446261 0.7414721 0.7373909  
## (0.795,0.921] 0.8375879 0.8295569 0.8355669

We can see that these are all quite similiar. So weighting by the base weights may not be necessary or distinct. Just to ensure, we perform a check on covariate balance by fitting an ANOVA model to service connections, which is continuous. We do not use the survey weights below since the interest is in whether balance has been achieved in the sample that was selected. Checks could be made using the weights, in which case the check would be on whether the census-fit model shows evidence of balance.

#extract classes  
p.class <- out$p.class  
#build glm  
chk1 <- glm(Service\_Connections ~ p.class + response + p.class\*response,  
data = allSmalls.requested)  
#print  
summary(chk1)

##   
## Call:  
## glm(formula = Service\_Connections ~ p.class + response + p.class \*   
## response, data = allSmalls.requested)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4434.7 -698.8 -451.4 634.6 8968.4   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 713.57 261.49 2.729 0.006576  
## p.class(0.672,0.707] 67.46 420.69 0.160 0.872674  
## p.class(0.707,0.795] 1687.28 447.50 3.770 0.000182  
## p.class(0.795,0.921] 4957.69 478.46 10.362 < 0.0000000000000002  
## response -243.19 342.95 -0.709 0.478581  
## p.class(0.672,0.707]:response 77.78 514.86 0.151 0.879982  
## p.class(0.707,0.795]:response 819.51 534.83 1.532 0.126078  
## p.class(0.795,0.921]:response 193.64 559.46 0.346 0.729389  
##   
## (Intercept) \*\*   
## p.class(0.672,0.707]   
## p.class(0.707,0.795] \*\*\*  
## p.class(0.795,0.921] \*\*\*  
## response   
## p.class(0.672,0.707]:response   
## p.class(0.707,0.795]:response   
## p.class(0.795,0.921]:response   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 3692485)  
##   
## Null deviance: 4079789347 on 515 degrees of freedom  
## Residual deviance: 1875782600 on 508 degrees of freedom  
## AIC: 9277.1  
##   
## Number of Fisher Scoring iterations: 2

In this case, the *p.class* factors all have coefficients that are significant while the p.class*resp interactions are not—the desired outcomes if mean service connections differs between classes but is the same for respondents and nonrespondents within a class. Another check is to fit a second model that includes only* p.class\* and to test whether the models are equivalent:

chk2 <- glm(Service\_Connections ~ p.class, data = allSmalls.requested)  
anova(chk2, chk1, test="F")

## Analysis of Deviance Table  
##   
## Model 1: Service\_Connections ~ p.class  
## Model 2: Service\_Connections ~ p.class + response + p.class \* response  
## Resid. Df Resid. Dev Df Deviance F Pr(>F)  
## 1 512 1885652237   
## 2 508 1875782600 4 9869637 0.6682 0.6143

The F-statistic is 0.647 with 506 and 502 degrees of freedom and has a p-value of 0.63. Thus, the model without a factor for responding is judged to be adequate.

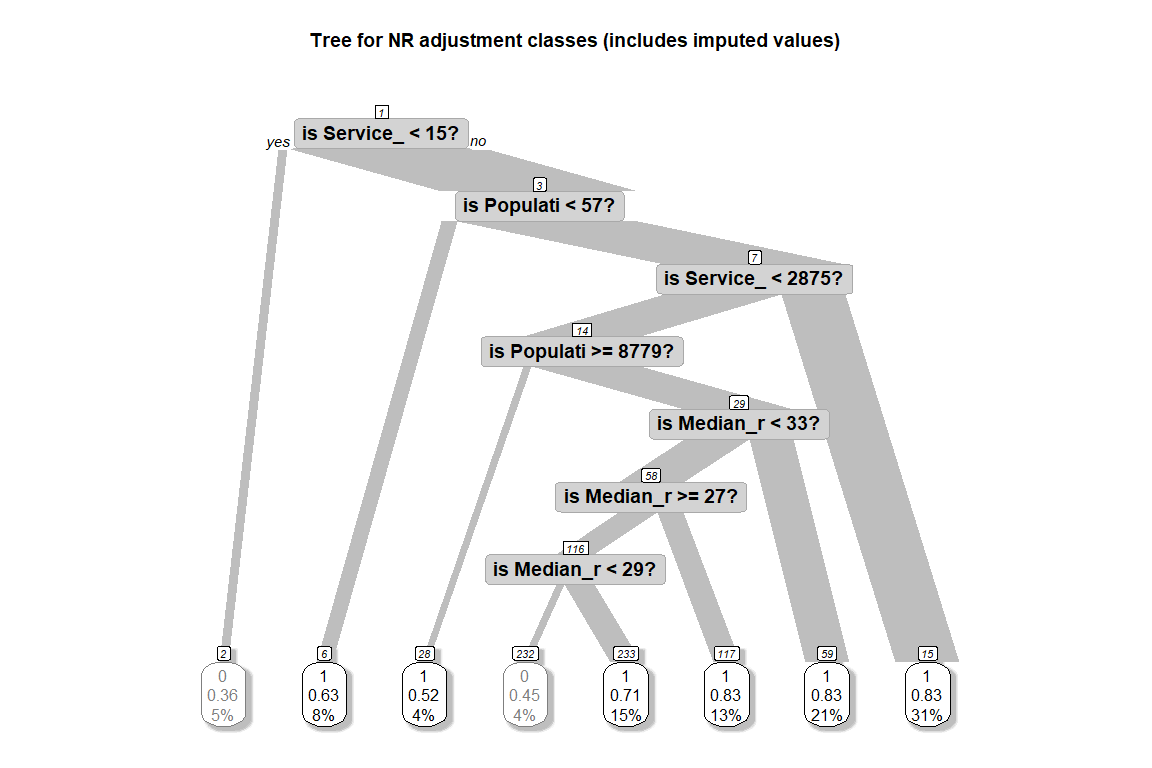
#### Classification and Regression Tree

In the nonrespondent application, the decision tree will classify cases using available covariates into classes that are related to their likelihood of being respondents. Advantages of CART compared to propensity modeling are that: 1. Interactions of covariates are handled automatically. 2. The way in which covariates enter the model does not have to be made explicit. 3. Selection of which covariates and associated interactions should be included is done automatically. 4. Variable values, whether categorical or continuous, are combined (grouped) automatically.

## n= 555   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 555 139 1 (0.2505 0.7495)   
## 2) Service\_Connections< 14.5 25 9 0 (0.6400 0.3600) \*  
## 3) Service\_Connections>=14.5 530 123 1 (0.2321 0.7679)   
## 6) Population< 57 43 16 1 (0.3721 0.6279) \*  
## 7) Population>=57 487 107 1 (0.2197 0.7803)   
## 14) Service\_Connections< 2874 315 77 1 (0.2444 0.7556)   
## 28) Population>=8779 21 10 1 (0.4762 0.5238) \*  
## 29) Population< 8779 294 67 1 (0.2279 0.7721)   
## 58) Median\_rent\_pct\_income< 32.91 175 47 1 (0.2686 0.7314)   
## 116) Median\_rent\_pct\_income>=27.4 103 35 1 (0.3398 0.6602)   
## 232) Median\_rent\_pct\_income< 28.66 20 9 0 (0.5500 0.4500) \*  
## 233) Median\_rent\_pct\_income>=28.66 83 24 1 (0.2892 0.7108) \*  
## 117) Median\_rent\_pct\_income< 27.4 72 12 1 (0.1667 0.8333) \*  
## 59) Median\_rent\_pct\_income>=32.91 119 20 1 (0.1681 0.8319) \*  
## 15) Service\_Connections>=2874 172 30 1 (0.1744 0.8256) \*

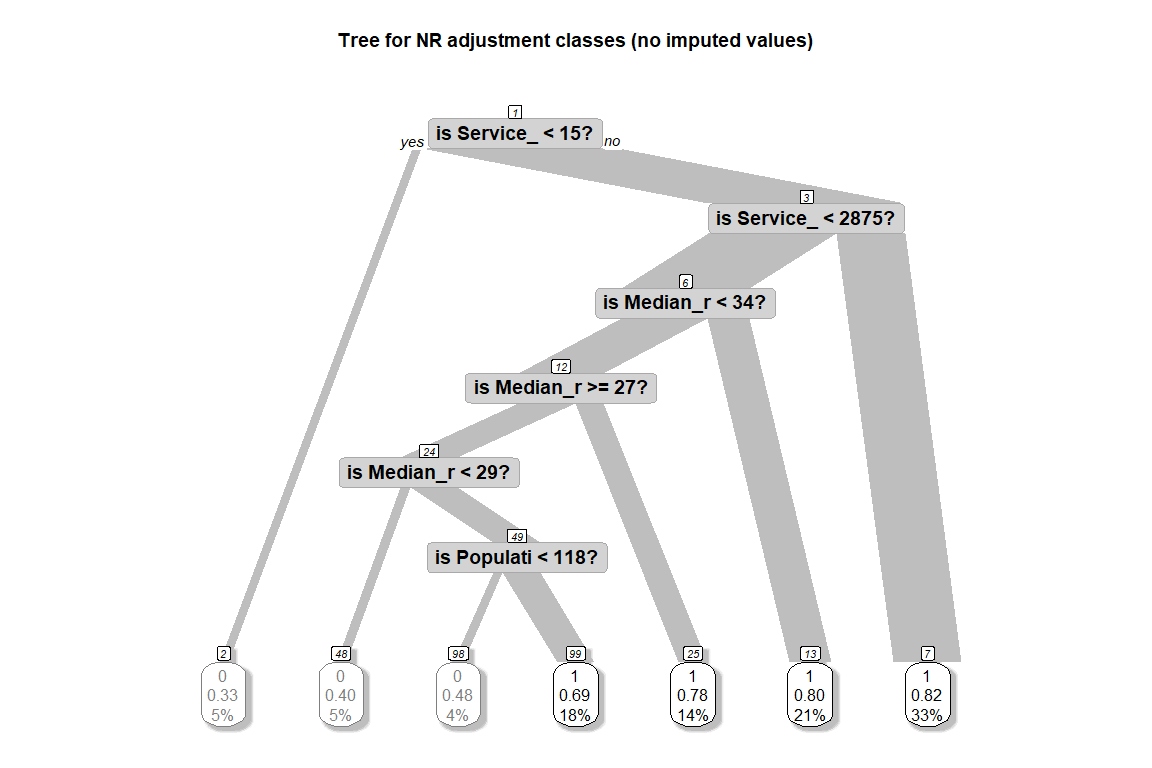
require(rpart.plot)  
cols <- ifelse(t1$frame$yval == 1, "gray50", "black")  
prp(t1, main="Tree for NR adjustment classes (includes imputed values)",  
 extra=106, # display prob of survival and percent of obs  
 nn=TRUE, # display node numbers  
 fallen.leaves=TRUE, # put leaves on the bottom of page  
 branch=.5, # change angle of branch lines  
 faclen=0, # do not abbreviate factor levels  
 trace=1, # print automatically calculated cex  
 shadow.col="gray", # shadows under the leaves  
 branch.lty=1, # draw branches using solid lines  
 branch.type=5, # branch lines width = weight(frame$wt), no. of cases here  
 split.cex=1.2, # make split text larger than node text  
 split.prefix="is ", # put "is " before split text  
 split.suffix="?", # put "?" after split text  
 col=cols, border.col=cols, # cols[2] if survived  
 split.box.col="lightgray", # lightgray split boxes (default is white)  
 split.border.col="darkgray", # darkgray border on split boxes  
 split.round=0.5) # round the split box corners a tad

## cex 1 xlim c(-0.2, 1.2) ylim c(0, 1)

 The classification and regression tree for non-response for small systems is depticted above.

## n= 516   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 516 139 1 (0.2694 0.7306)   
## 2) Service\_Connections< 14.5 24 8 0 (0.6667 0.3333) \*  
## 3) Service\_Connections>=14.5 492 123 1 (0.2500 0.7500)   
## 6) Service\_Connections< 2874 321 93 1 (0.2897 0.7103)   
## 12) Median\_rent\_pct\_income< 34.33 214 72 1 (0.3364 0.6636)   
## 24) Median\_rent\_pct\_income>=27.17 142 56 1 (0.3944 0.6056)   
## 48) Median\_rent\_pct\_income< 28.66 25 10 0 (0.6000 0.4000) \*  
## 49) Median\_rent\_pct\_income>=28.66 117 41 1 (0.3504 0.6496)   
## 98) Population< 118 23 11 0 (0.5217 0.4783) \*  
## 99) Population>=118 94 29 1 (0.3085 0.6915) \*  
## 25) Median\_rent\_pct\_income< 27.17 72 16 1 (0.2222 0.7778) \*  
## 13) Median\_rent\_pct\_income>=34.33 107 21 1 (0.1963 0.8037) \*  
## 7) Service\_Connections>=2874 171 30 1 (0.1754 0.8246) \*

## cex 1 xlim c(-0.2, 1.2) ylim c(0, 1)



#### Random Forest Model

A single regression tree tends to overfit data in the sense of creating a model that may not be accurate for a new dataset (like the units that were not sampled or another sample selected using the same methods that is also subject to nonresponse) (Valliant et al. 2018). The fitted model from a single tree may not be the best representation of the underlying response mechanism for a nonresponse adjustment. To account for this “shrinkage issue”, Breiman (2001) formulated random forests. According to Valliant et al. (2018), random forests fit many regression trees and average the results with the goal of producing more robust, lower variance predictions. Random forests can incorporate a large number of weighting variables and can find complicated relationships between adjustment variables that a researcher may not be aware of in advance (Zhao et al. 2016).

##   
## [> ] 0% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed

## [1] 0.2478495 0.7521505

|  |  |  |
| --- | --- | --- |
|  | rpart | cforest |
| 2 | 0.3600 | 0.4346 |
| 4 | 0.6279 | 0.6786 |
| 7 | 0.5238 | 0.6667 |
| 11 | 0.4500 | 0.6614 |
| 12 | 0.7108 | 0.7330 |
| 13 | 0.8333 | 0.7615 |
| 14 | 0.8319 | 0.7975 |
| 15 | 0.8256 | 0.8117 |

The estimated overall response rate from the random forest model is 0.748. This is very close to the actual overall response rate of 0.7468. Above we can see the different propensity classes predicted by the rforest and the cforest models.

require(party)  
crf.srvy.2 <- cforest(as.factor(response) ~ Service\_Connections + Fee\_Code + Population + Median\_rent\_pct\_income + Median\_12month\_HH\_income,   
 controls = cforest\_control(ntree = 500,   
 mincriterion = qnorm(0.8),   
 trace = TRUE), # adds project bar because it's very slow  
 data=allSmalls.requested)

##   
## [> ] 0% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[=> ] 2% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[==> ] 4% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[===> ] 6% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[====> ] 8% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[=====> ] 10% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[======> ] 12% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[=======> ] 14% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[========> ] 16% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[=========> ] 18% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[==========> ] 20% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[===========> ] 22% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[============> ] 24% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[=============> ] 26% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[==============> ] 28% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[===============> ] 30% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[================> ] 32% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[=================> ] 34% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[==================> ] 36% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[===================> ] 38% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[====================> ] 40% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[=====================> ] 42% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[======================> ] 44% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[=======================> ] 46% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[========================> ] 48% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[=========================> ] 50% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[==========================> ] 52% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[===========================> ] 54% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[============================> ] 56% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[=============================> ] 58% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[==============================> ] 60% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[===============================> ] 62% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[================================> ] 64% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[=================================> ] 66% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[==================================> ] 68% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[===================================> ] 70% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[====================================> ] 72% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[=====================================> ] 74% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[======================================> ] 76% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[=======================================> ] 78% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[========================================> ] 80% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[=========================================> ] 82% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[==========================================> ] 84% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[===========================================> ] 86% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[============================================> ] 88% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[=============================================> ] 90% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[==============================================> ] 92% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[===============================================> ] 94% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[================================================> ] 96% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[=================================================> ] 98% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed[==================================================>] 100% completed

crfsrvy.prob2 <- predict(crf.srvy.2,newdata = allSmalls.requested,type="prob")  
rpart.prob2 <- predict(t2, newdata = allSmalls.requested,type="prob")  
crf.prob2 <- matrix(unlist(crfsrvy.prob2), ncol=2, byrow=TRUE)  
apply(crf.prob2,2,mean)

## [1] 0.2679982 0.7320018

#[1] 0.2711782 0.7288218  
tab <- round(cbind(by(rpart.prob2[,2], INDICES=t2$where, mean), by(crf.prob2[,2], INDICES=t2$where, mean)), 4)  
colnames(tab) <- c("rpart (no imputation)", "cforest (no imputation)")  
kable(tab)

|  |  |  |
| --- | --- | --- |
|  | rpart (no imputation) | cforest (no imputation) |
| 2 | 0.3333 | 0.3982 |
| 7 | 0.4000 | 0.6191 |
| 9 | 0.4783 | 0.6250 |
| 10 | 0.6915 | 0.7029 |
| 11 | 0.7778 | 0.7204 |
| 12 | 0.8037 | 0.7670 |
| 13 | 0.8246 | 0.8087 |

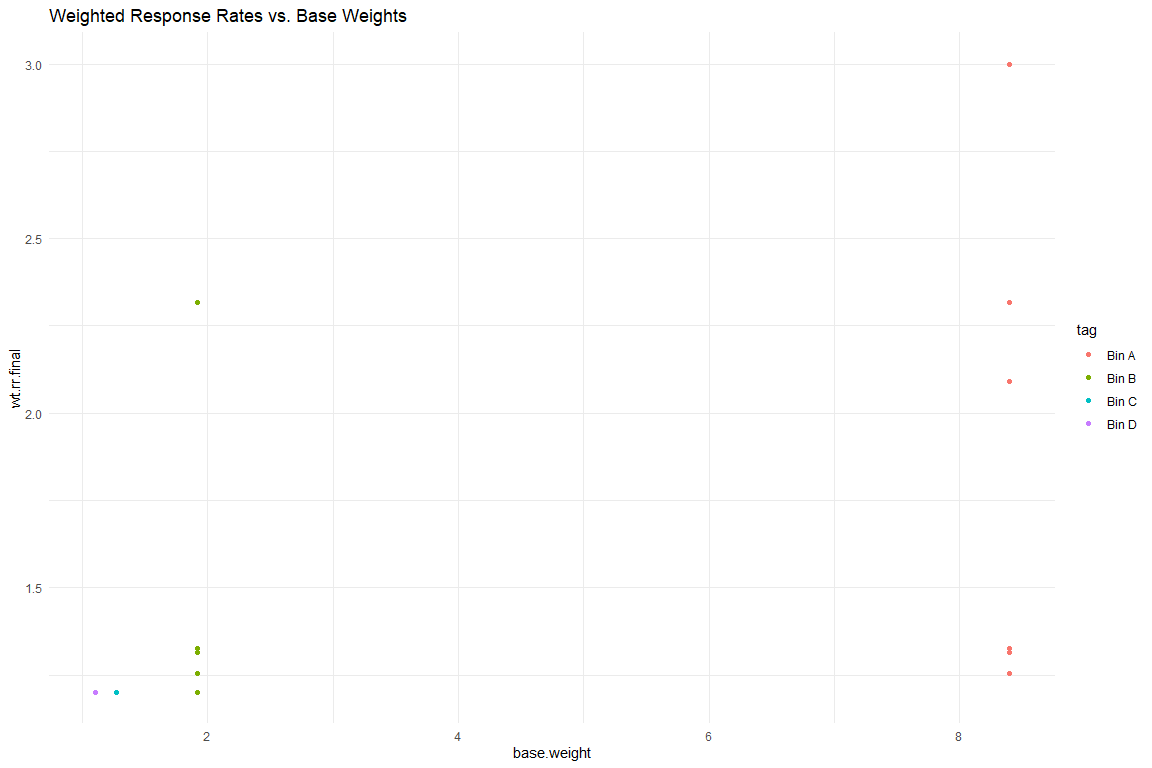
Adjusted weights can now be computed and merged into the data file.

## [1] 1.201262 1.254923 1.314483 1.324701 2.090909 2.317558 3.000000

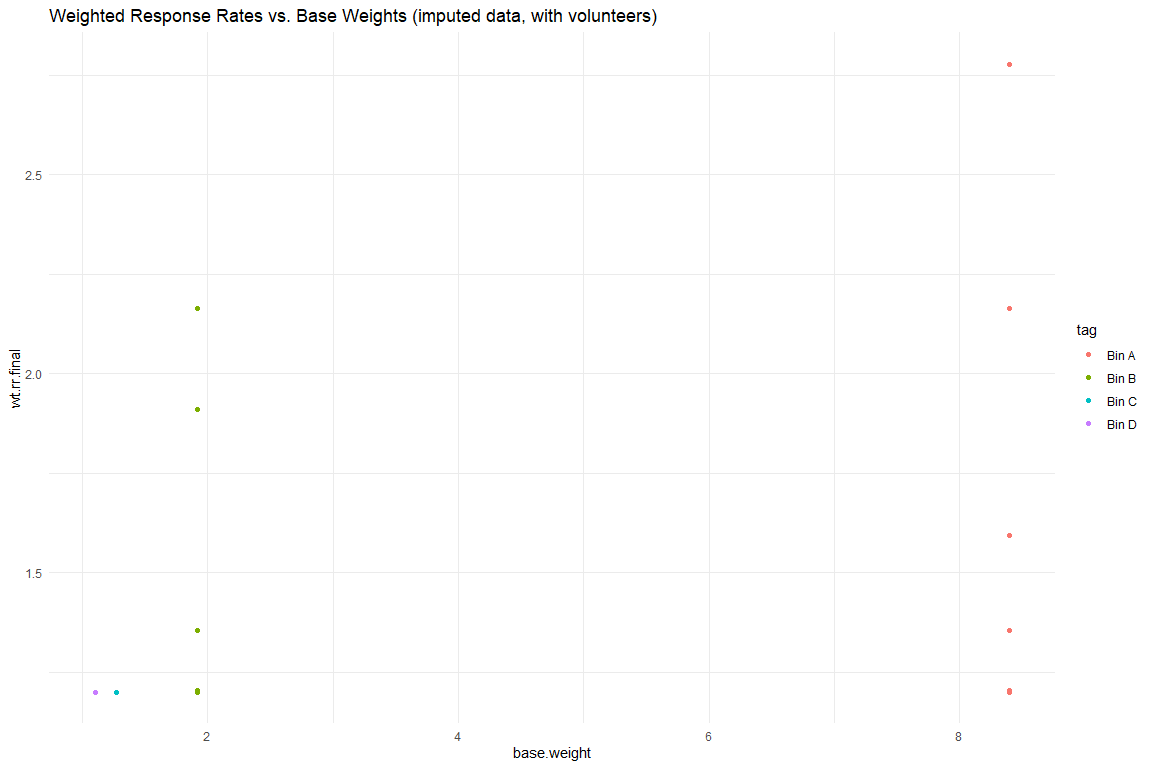
## [1] 1.212766 1.244186 1.285714 1.446154 2.090909 2.500000 3.000000

These nonresponse weights are plotted relative to respective baseweights below.

#plot versus base weights  
ggplot(data = allSmalls.requested.NR, aes(x = base.weight, y = wt.rr.final, color = tag)) + geom\_point() +  
 labs(title = "Weighted Response Rates vs. Base Weights",  
 xlab = "Base Weights",  
 ylab = "Weighted Reponse Rates") +  
 theme\_minimal()



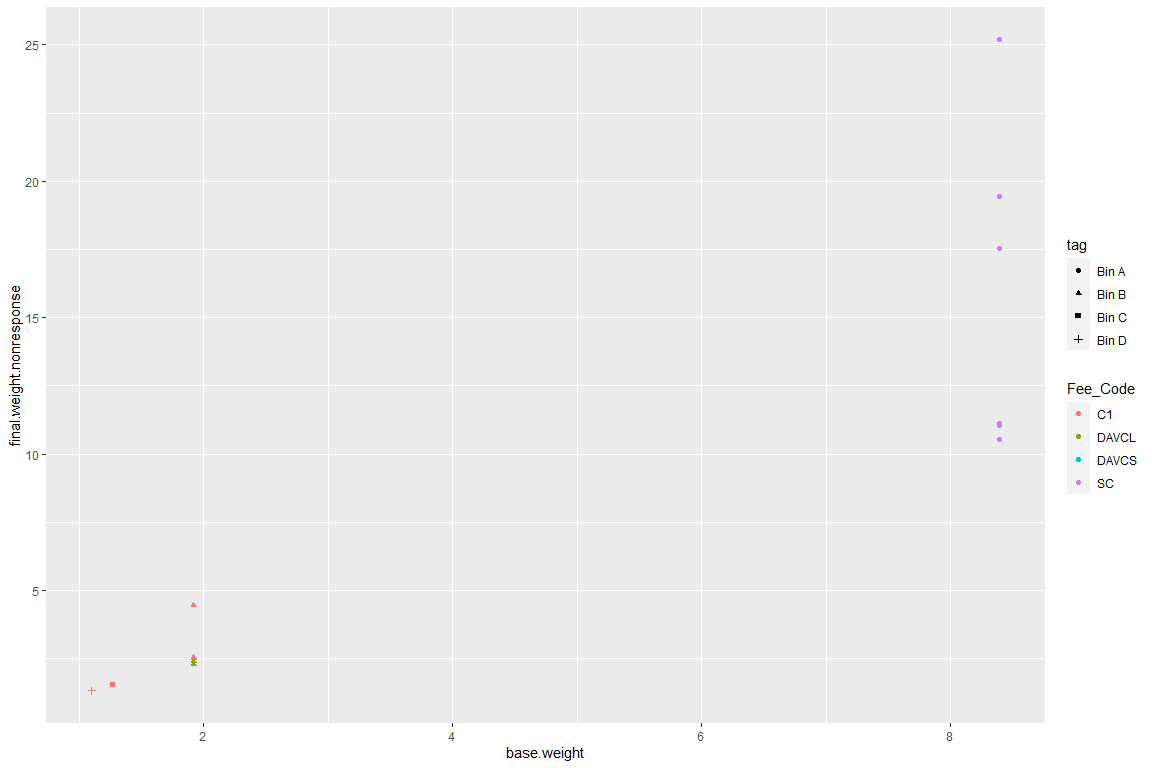
## [1] 1.199856 1.200000 1.203111 1.354916 1.592593 1.909091 2.164748 2.777778



If we had no information about population estimates, we would end the weighting procedure here. The ‘final weight’ would be the multiplication of both base scaled weight and the scaled non-response weight. Here we will call this new weights ‘final weights’ although we still have to perform adjustments to them and so will not really be ‘final’.

Before going to the next step we will include the computed non-response weights using adjustment classes to the main ‘data’ dataframe object. Then we will drop all non-respondents as we are not going to use them any more in the next steps of our analysis. After that, we will scale the non-response weights to the sample of respondents and multiply the design weights and the non-response weights.

## # A tibble: 4 x 5  
## tag sum.final.weight totalSamples population percent.diff  
## <fct> <dbl> <dbl> <int> <dbl>  
## 1 Bin A 1932. 228 1914 -0.933  
## 2 Bin B 246. 135 259 5.10   
## 3 Bin C 106. 84 107 1.32   
## 4 Bin D 72.8 69 76 4.25



## # A tibble: 4 x 6  
## tag sum.final.weight sum.final.weight.unwt.rr totalSamples population  
## <fct> <dbl> <dbl> <dbl> <int>  
## 1 Bin A 1914. 3299. 228 1914  
## 2 Bin B 259. 379. 135 259  
## 3 Bin C 107. 131. 84 107  
## 4 Bin D 76.0 92.1 69 76  
## percent.diff  
## <dbl>  
## 1 -0.00000485  
## 2 -0.00000267  
## 3 0.0000124   
## 4 0.0000124

### Large Systems

First let’s see the response rate for each of the bins for the large systems.

allLarges %>% filter(requested == "y") %>%  
 summarize(completeness = sum(response)/n() \* 100 , completed = sum(response), total = n()) %>%   
 kable(caption = "Completeness for Large Systems")

Completeness for Large Systems

|  |  |  |
| --- | --- | --- |
| completeness | completed | total |
| 83.33333 | 125 | 150 |

As mentioned in the previous section, computing the probability of replying to this survey is challenging because we can not directly observe the probability of replying to the survey, therefore we need to estimate it. This may be done using information which we know for both respondent and non-respondent units. The most reliable (albeit most complicated) of calculating non-response probabilities is through predictive modelling.

[Valliant et al (2013)](https://link.springer.com/book/10.1007%2F978-1-4614-6449-5) recommend estimating the response propensities and then grouping them in classes. The grouping step should avoid extreme weights. One way of estimating the response propensities is using logistic regression. This logistic regression should be unweighted. We will use a general linear model function to predict non-responses with hypothesized response variables for which we have data, including service connections, population, fee code, median 12-month household income, and median percent income.

#trim data to those in sample list with auxiliary data  
allLarges.requested.response.model <- allLarges %>%  
 filter(requested == "y") %>%  
 select(response, Service\_Connections, Population, CES\_3.0\_Score, Median\_12month\_HH\_income,  
 Median\_rent\_pct\_income) %>%  
 drop\_na()  
  
#prepare formula  
formula.resp <- as.formula("response ~ Service\_Connections + Population + CES\_3.0\_Score + Median\_12month\_HH\_income + Median\_rent\_pct\_income")  
options(na.action = 'na.pass')  
  
#format as matrix  
x.matrix <- model.matrix(formula.resp , data = allLarges.requested.response.model)[, -1]  
  
#fit binomial model  
log.reg.m <- glm(formula.resp, data = allLarges.requested.response.model,family = "binomial")  
  
coef.response.log <- coef(log.reg.m)  
predicted.log <- log.reg.m$fitted.values  
non.responsepredicted.log <- predicted.log  
  
predicted.log %>% head() %>% kable(caption = "Linear Regression-predicted Top Six Response Propensity Classes")

Linear Regression-predicted Top Six Response Propensity Classes

|  |
| --- |
| x |
| 1.0000000 |
| 0.9999996 |
| 0.9999848 |
| 0.9997569 |
| 0.9988361 |
| 0.9983223 |

The above output shows the first six estimated response propensities of our dataset. Since we don’t have pardata for all samples, we are now computing our predictor estimates from a subset of sampled units.

Let’s define our survey structure in the *survey* package.

The package *PracTools* has tools to determine propensity classes. We will use service connections, population, fee code, and regulating agency as predictor variables.

First let’s start by testing which variables are predictive of response.

#redefine all larges  
allLarges.requested <- allLarges %>% filter(requested == "y")  
#fit  
fitResponse <- lm(response ~ Service\_Connections + Population + Median\_12month\_HH\_income + Median\_rent\_pct\_income + CES\_3.0\_Score + Fee\_Code, na.action = na.omit, data = allLarges.requested)  
summary(fitResponse)

##   
## Call:  
## lm(formula = response ~ Service\_Connections + Population + Median\_12month\_HH\_income +   
## Median\_rent\_pct\_income + CES\_3.0\_Score + Fee\_Code, data = allLarges.requested,   
## na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.88331 0.04215 0.15029 0.20839 0.31883   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.2925492669 0.5068248900 2.550 0.0118 \*  
## Service\_Connections 0.0000025976 0.0000022025 1.179 0.2402   
## Population -0.0000003729 0.0000004223 -0.883 0.3787   
## Median\_12month\_HH\_income -0.0000030378 0.0000018087 -1.680 0.0952 .  
## Median\_rent\_pct\_income -0.0044052821 0.0122772700 -0.359 0.7203   
## CES\_3.0\_Score -0.0046871891 0.0037238508 -1.259 0.2102   
## Fee\_CodeDAVCL 0.1482603278 0.1235563826 1.200 0.2321   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3704 on 143 degrees of freedom  
## Multiple R-squared: 0.05852, Adjusted R-squared: 0.01902   
## F-statistic: 1.481 on 6 and 143 DF, p-value: 0.1886

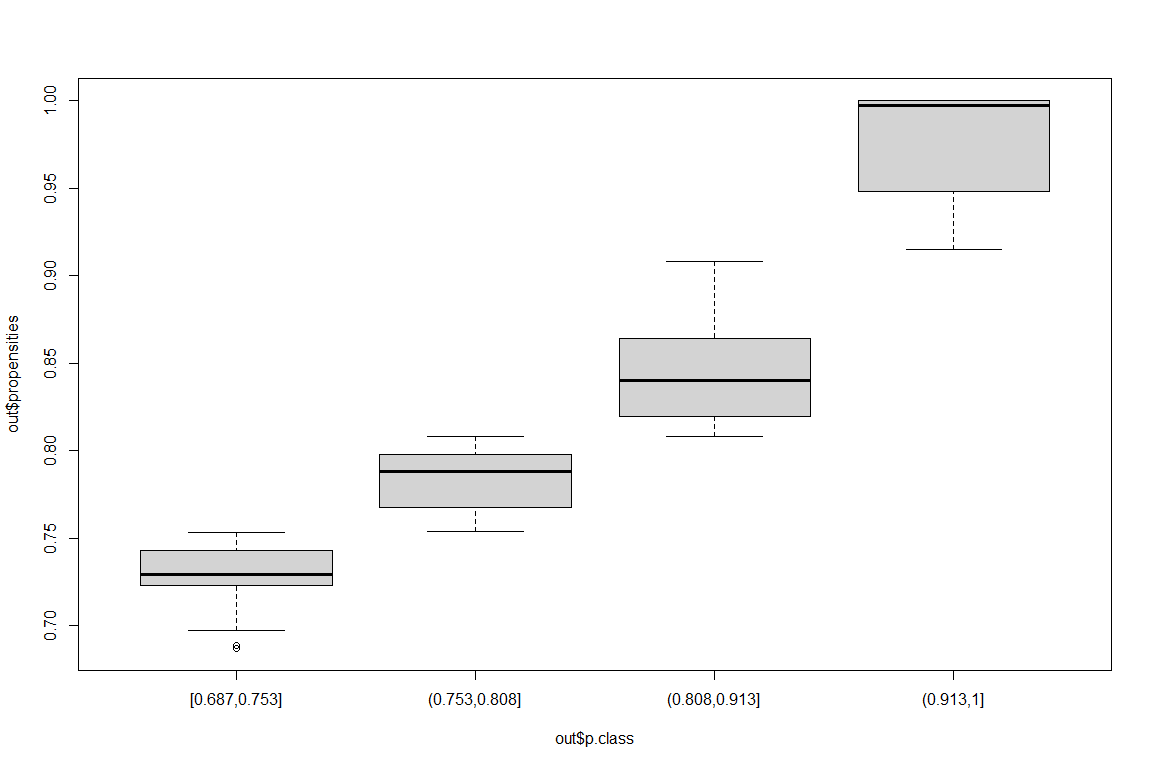
Unlike small systems, there are no good predictors for response propensity for large systems. This may be indicative of a non-systematic non-response.

##   
## [0.687,0.753] (0.753,0.808] (0.808,0.913] (0.913,1] <NA>   
## 38 37 37 38 0

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.6873 0.7534 0.8082 0.8333 0.9134 1.0000

These propensity classes can be viewed in boxplot form.

boxplot(out$propensities ~ out$p.class)

 We can see discrete propensity classes with rather tight spread for the four uppermost categories. Let’s perform a check on covariate balance by fitting an ANOVA model to service connections, which is continuous. We do not use the survey weights below since the interest is in whether balance has been achieved in the sample that was selected. Checks could be made using the weights, in which case the check would be on whether the census-fit model shows evidence of balance.

#extract classes  
p.class <- out$p.class  
#build glm  
chk1 <- glm(Service\_Connections ~ p.class + response + p.class\*response,  
data = allLarges.requested)  
#print  
summary(chk1)

##   
## Call:  
## glm(formula = Service\_Connections ~ p.class + response + p.class \*   
## response, data = allLarges.requested)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -84924 -5392 -1217 1291 613647   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12982.7 21975.8 0.591 0.556   
## p.class(0.753,0.808] 5197.3 32035.0 0.162 0.871   
## p.class(0.808,0.913] 18081.3 32035.0 0.564 0.573   
## p.class(0.913,1] 82859.5 16256.0 5.097 0.00000107 \*\*\*  
## response -353.8 25155.8 -0.014 0.989   
## p.class(0.753,0.808]:response 2576.1 36414.2 0.071 0.944   
## p.class(0.808,0.913]:response -159.8 36414.2 -0.004 0.997   
## p.class(0.913,1]:response NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 4346432665)  
##   
## Null deviance: 785009242279 on 149 degrees of freedom  
## Residual deviance: 621539871061 on 143 degrees of freedom  
## AIC: 3763.4  
##   
## Number of Fisher Scoring iterations: 2

In this case, the *p.class* factors do not have coefficients that are significant (except for the upper-most p-class). Further, the p.class*resp interactions are not significant—the desired outcomes if mean service connections differs between classes but is the same for respondents and nonrespondents within a class. Another check is to fit a second model that includes only* p.class\* and to test whether the models are equivalent:

chk2 <- glm(Service\_Connections ~ p.class, data = allLarges.requested)  
anova(chk2, chk1, test="F")

## Analysis of Deviance Table  
##   
## Model 1: Service\_Connections ~ p.class  
## Model 2: Service\_Connections ~ p.class + response + p.class \* response  
## Resid. Df Resid. Dev Df Deviance F Pr(>F)  
## 1 146 621573352295   
## 2 143 621539871061 3 33481234 0.0026 0.9998

The F-statistic is 0.0026 with 146 and 143 degrees of freedom and has a p-value of 0.9998. Thus, the model without a factor for responding is judged to be adequate.

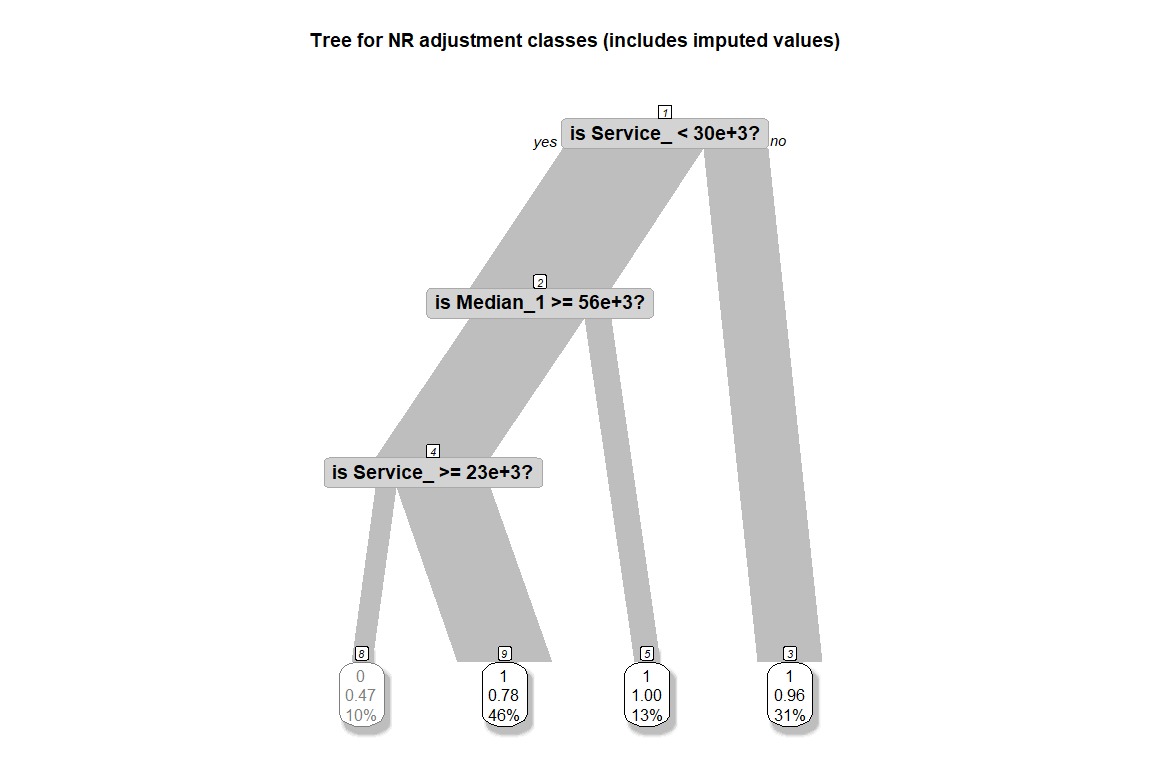
#### Classification and Regression Tree

In the nonrespondent application, the decision tree will classify cases using available covariates into classes that are related to their likelihood of being respondents. Advantages ofa Classification and Regression Tree compared to propensity modeling are that: 1. Interactions of covariates are handled automatically. 2. The way in which covariates enter the model does not have to be made explicit. 3. Selection of which covariates and associated interactions should be included is done automatically. 4. Variable values, whether categorical or continuous, are combined (grouped) automatically.

## n= 150   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 150 25 1 (0.16667 0.83333)   
## 2) Service\_Connections< 3.011e+04 103 23 1 (0.22330 0.77670)   
## 4) Median\_12month\_HH\_income>=5.639e+04 84 23 1 (0.27381 0.72619)   
## 8) Service\_Connections>=2.328e+04 15 7 0 (0.53333 0.46667) \*  
## 9) Service\_Connections< 2.328e+04 69 15 1 (0.21739 0.78261) \*  
## 5) Median\_12month\_HH\_income< 5.639e+04 19 0 1 (0.00000 1.00000) \*  
## 3) Service\_Connections>=3.011e+04 47 2 1 (0.04255 0.95745) \*

require(rpart.plot)  
cols <- ifelse(t1$frame$yval == 1, "gray50", "black")  
prp(t1, main="Tree for NR adjustment classes (includes imputed values)",  
 extra=106, # display prob of survival and percent of obs  
 nn=TRUE, # display node numbers  
 fallen.leaves=TRUE, # put leaves on the bottom of page  
 branch=.5, # change angle of branch lines  
 faclen=0, # do not abbreviate factor levels  
 trace=1, # print automatically calculated cex  
 shadow.col="gray", # shadows under the leaves  
 branch.lty=1, # draw branches using solid lines  
 branch.type=5, # branch lines width = weight(frame$wt), no. of cases here  
 split.cex=1.2, # make split text larger than node text  
 split.prefix="is ", # put "is " before split text  
 split.suffix="?", # put "?" after split text  
 col=cols, border.col=cols, # cols[2] if survived  
 split.box.col="lightgray", # lightgray split boxes (default is white)  
 split.border.col="darkgray", # darkgray border on split boxes  
 split.round=0.5) # round the split box corners a tad

## cex 1 xlim c(-0.65, 1.65) ylim c(0, 1)



#### Random Forest Modelling

As described above with small systems, a random forest model is compared it it’s prediction of response propensity for large systems.

require(party)  
crf.srvy.larges <- cforest(as.factor(response) ~ Service\_Connections + Fee\_Code + Population + Median\_rent\_pct\_income + Median\_12month\_HH\_income,  
 controls = cforest\_control(ntree = 500,  
 mincriterion = qnorm(0.8),  
 trace = TRUE), # adds project bar because it's very slow  
 data=allLarges.requested)  
  
crfsrvy.prob.larges <- predict(crf.srvy.larges,newdata=allLarges.requested,type="prob")  
rpart.prob.larges <- predict(t1, newdata=allLarges.requested,type="prob")  
crf.prob.larges <- matrix(unlist(crfsrvy.prob.larges), ncol=2, byrow=TRUE)  
apply(crf.prob.larges,2,mean)  
#[1] 0.1701657 0.8298343  
tab <- round(cbind(by(rpart.prob.larges[,2], INDICES=t1$where, mean), by(crf.prob.larges[,2], INDICES=t1$where, mean)), 4)  
colnames(tab) <- c("rpart", "cforest")  
kable(tab,  
 caption = "predicted response propensity classes for large systems")

The estimated overall response rate is 0.8298. This is very close to the actual overall response rate of 0.833. Above we can see the different propensity classes predicted by the rforest and the cforest. The cforest seemed to generate more realistic propensity classes that match the data (see boxplot above).

## [1] 1.018274 1.018946 1.019195 1.021061 1.022284 1.024795 1.025349 1.029813  
## [9] 1.032465 1.032820 1.034235 1.034735 1.035547 1.035569 1.036394 1.037952  
## [17] 1.038420 1.040481 1.041059 1.041555 1.043095 1.044565 1.048608 1.053234  
## [25] 1.057827 1.058501 1.059835 1.060522 1.068071 1.070041 1.071270 1.072454  
## [33] 1.072849 1.073904 1.077108 1.077889 1.080527 1.086333 1.087073 1.090726  
## [41] 1.091263 1.092958 1.093776 1.097694 1.099201 1.100280 1.100583 1.102730  
## [49] 1.104026 1.104289 1.107506 1.110761 1.112706 1.115361 1.117382 1.118045  
## [57] 1.121640 1.126347 1.126794 1.128269 1.132682 1.134564 1.135758 1.137673  
## [65] 1.139261 1.140321 1.142467 1.143065 1.143638 1.143824 1.144279 1.146196  
## [73] 1.148880 1.155428 1.155941 1.157872 1.160714 1.161081 1.170602 1.176663  
## [81] 1.182862 1.183461 1.184467 1.185414 1.202297 1.210265 1.212545 1.216678  
## [89] 1.218884 1.221027 1.221881 1.223461 1.226169 1.226461 1.233573 1.234879  
## [97] 1.237489 1.237781 1.238656 1.239810 1.257561 1.264989 1.279552 1.280178  
## [105] 1.281235 1.282022 1.289293 1.289968 1.303474 1.305638 1.312134 1.314026  
## [113] 1.314523 1.315763 1.322209 1.327147 1.327215 1.336327 1.350335 1.367290  
## [121] 1.388395 1.390945 1.392759 1.403273 1.415402 1.419795 1.425021 1.438948  
## [129] 1.444469 1.463698 1.489879 1.494060 1.511230 1.511289 1.524122 1.547445  
## [137] 1.554574 1.556485 1.564907 1.605095 1.609787 1.614493 1.615918 1.625159  
## [145] 1.641292 1.685041 1.729847 1.757939 1.794773 1.823946

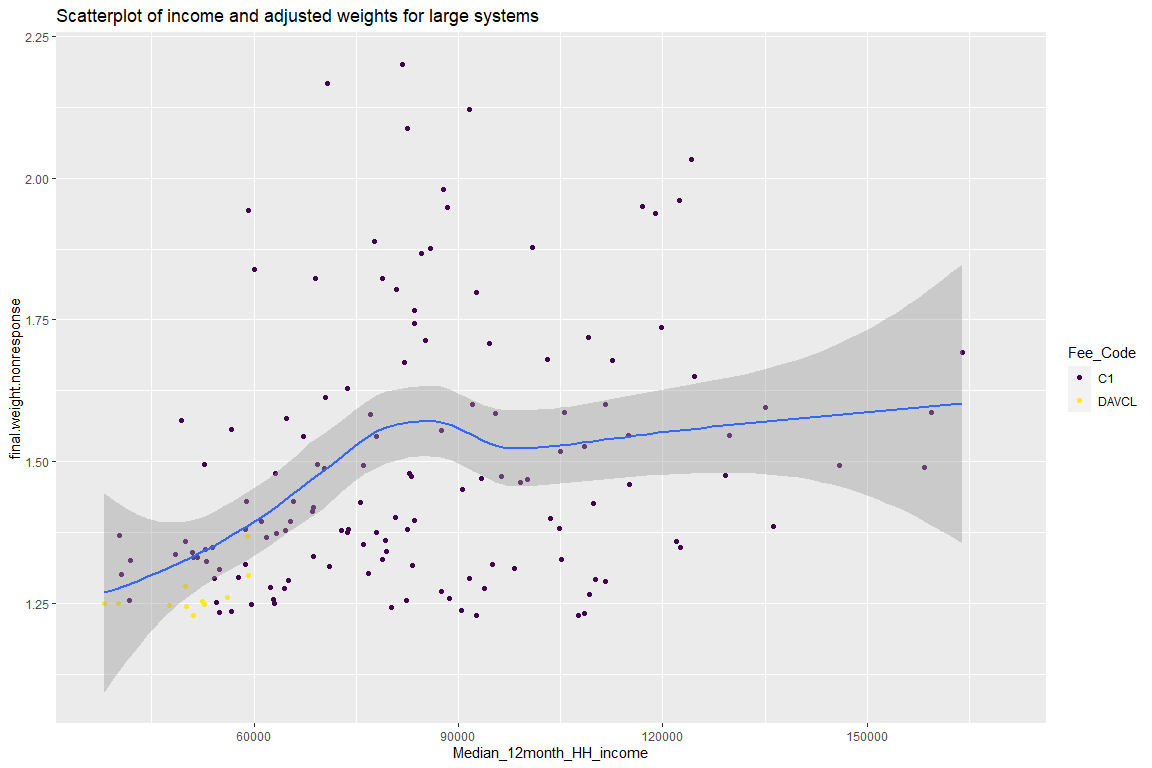
If we had no information about population estimates, we would end the weighting procedure here. The ‘final weight’ would be the multiplication of both base scaled weight and the scaled non-response weight. Here we will call this new weights ‘final weights’ although we still have to perform adjustments to them and so will not really be ‘final’.

Before going to the next step we will include the computed non-response weights using adjustment classes to the main ‘data’ dataframe object. Then we will drop all non-respondents as we are not going to use them any more in the next steps of our analysis. After that, we will scale the non-response weights to the sample of respondents and multiply the design weights and the non-response weights.

## sum.final.weight totalSamples population percent.diff  
## 1 222.0626 128 223 0.4203467

Here we can see that the sum of the weights adjusted for non-response are the same as the population. The adjusted weights are plotted below:

allLarges %>%  
 ggplot(aes(x = Median\_12month\_HH\_income, y =final.weight.nonresponse)) +  
 geom\_point(aes(color = Fee\_Code))+  
 scale\_color\_viridis\_d()+  
 geom\_smooth() +  
 labs(title = "Scatterplot of income and adjusted weights for large systems")

 Although the relationship between non-response and median 12 month household income is non-linear (as predicted above using a multi-variate, machine-learning classification-based approach), a relationship may be observed between response propensity and median income in the above figure.

## Weight Calibration

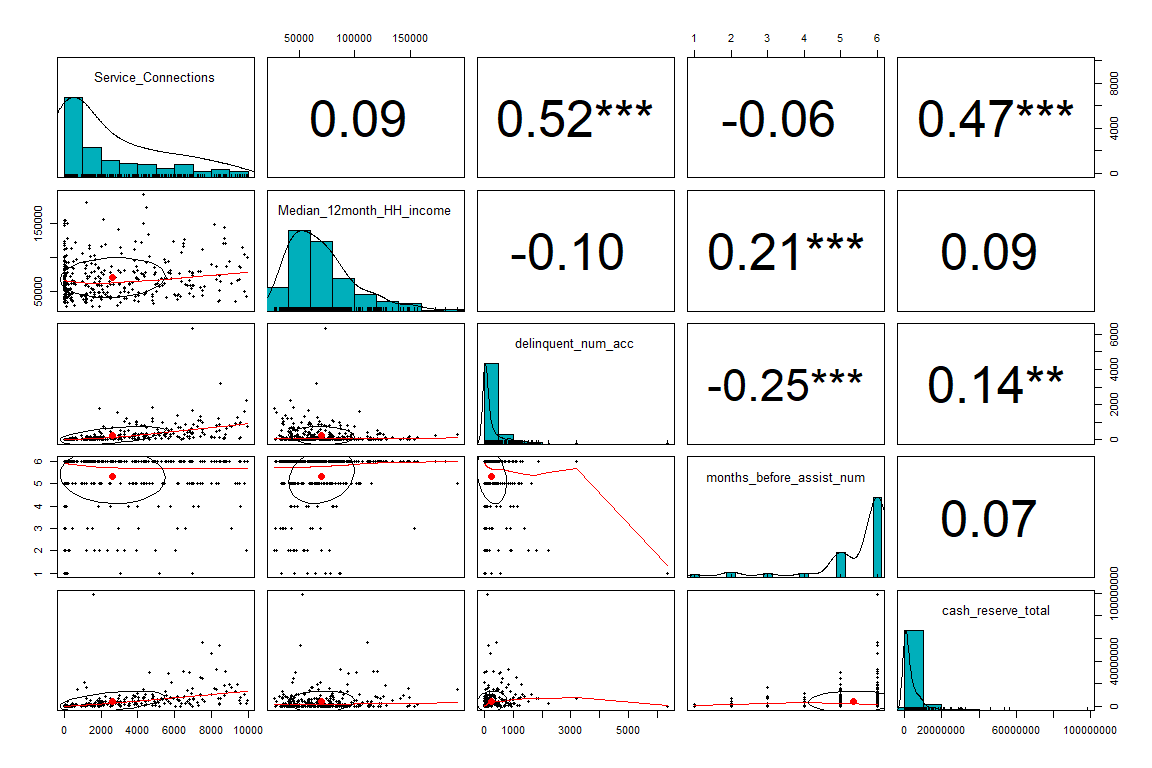
We’ve now generated *base weights* and *nonresponse adjustments* to those weights. The last step, which is extremely important in many surveys, is to use auxiliary data to correct coverage problems and to reduce standard errors.By auxiliary data, we mean information that is available for the entire frame or target population, either for each individual population unit or in aggregate form (i.e. househould income, service connections, population, fee code, regulating agency).Using auxiliary variable to adjust survey variables may reduce variances, but improve the representativeness of the survey.

Since our auxiliary variables are both quantitative(i.e. househould income, service connections, population) and qualitative (fee code, regulating agency), we will use a general regression estimator or GREG approach, which is a type of raking estimator. A GREG is approximately unbiased in repeated sampling if the frame provides full coverage of the target population (in this case we have census data for these auxiliary variables) ( [Vallian et al 2018](http://link.springer.com/10.1007/978-3-319-93632-1)).

### Evaluation of Correlative factors

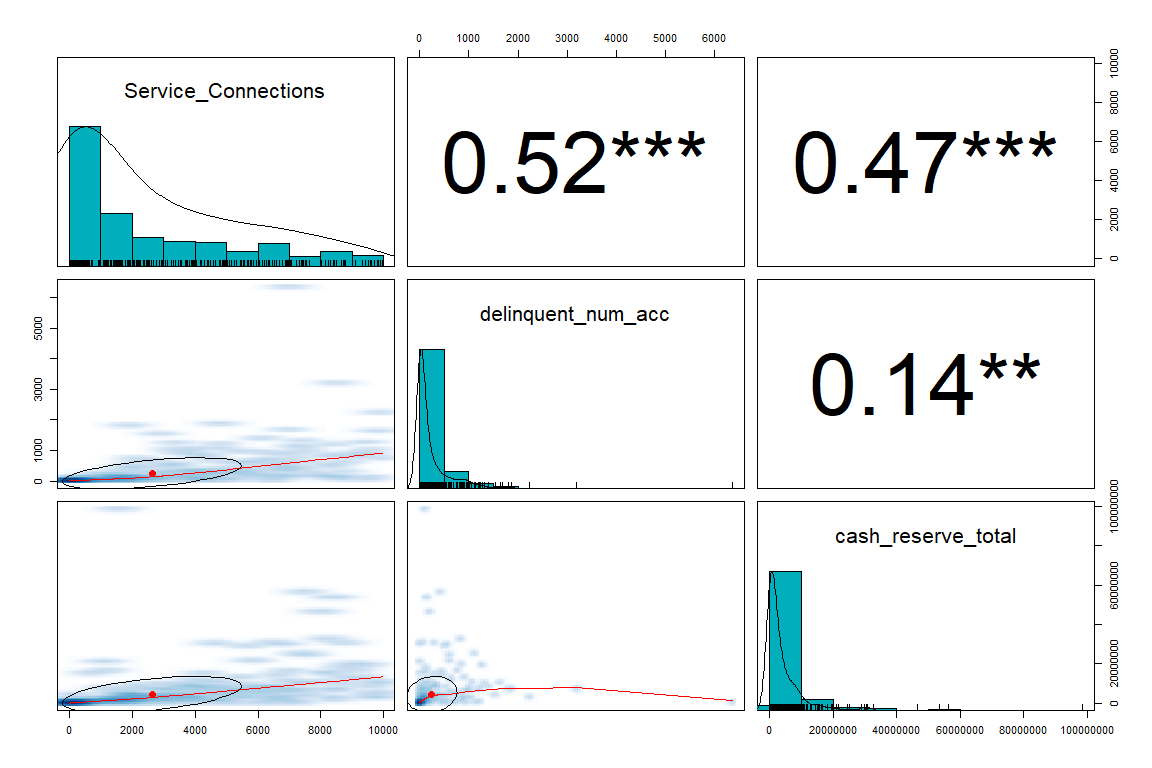
The first step in selecting appropriate auxiliary variables to use for raking is to deterimine correlations with response variables of interest. This can be achieved using a scatterplot matrix of the quantitative variables in the dataset.

require(psych)  
#trim data  
aux.response <- allSmalls.requested.responded %>% select(Service\_Connections,Median\_12month\_HH\_income, delinquent\_num\_acc, months\_before\_assist\_num, cash\_reserve\_total)  
# plot matrix  
pairs.panels(aux.response,  
 method = "pearson",  
 hist.col = "#00AFBB", # color  
 density = TRUE, #show histograms  
 ellipses = TRUE,#annotate correlation elliopses  
 stars = TRUE) #show significance)



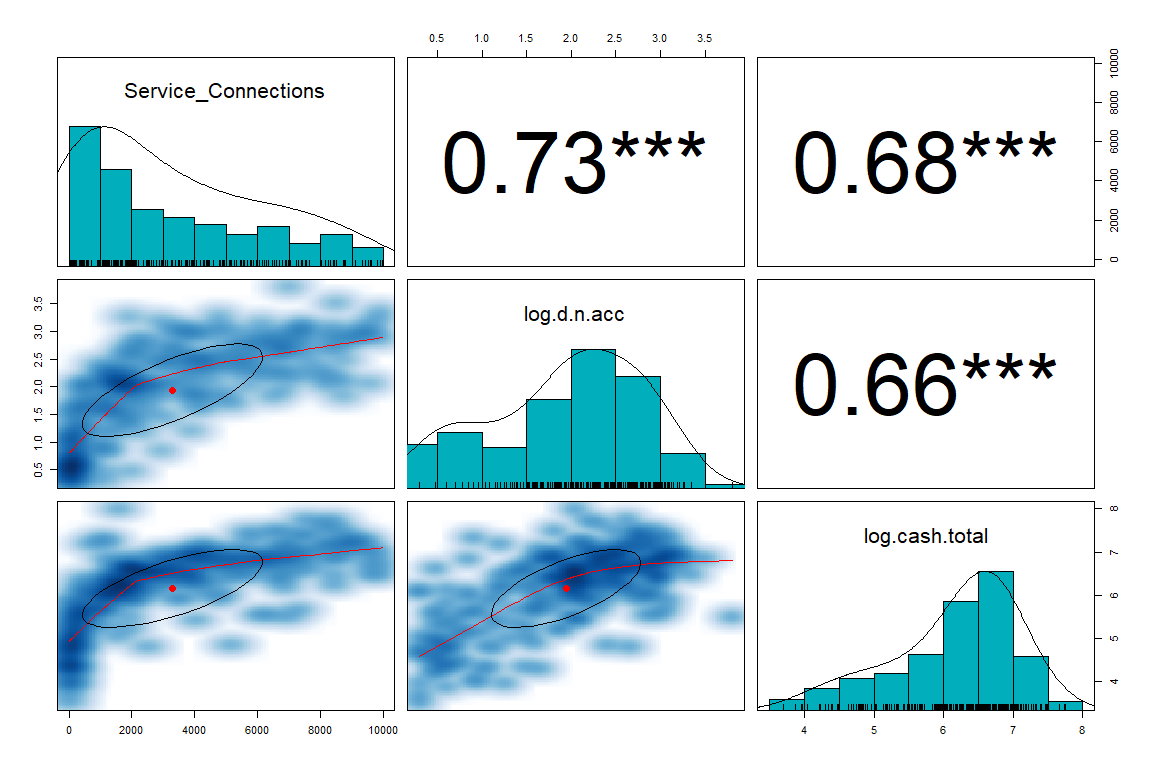
Here we can see that all of the quantitative auxiliary variables (Service\_Connections, Median\_12month\_HH\_income) are correlated with the selected response variables (delinquent number of accounts, months before assistance needed, cash reserve total). These are appropriate variables to use for raking. Let’s look a little closer at the relationships between service connections and delinquent number of accounts and total cash reserve.

require(psych)  
#trim data  
aux.response <- allSmalls.requested.responded %>%   
 select(Service\_Connections, delinquent\_num\_acc, cash\_reserve\_total)  
# plot matrix  
pairs.panels(aux.response,  
 method = "pearson",  
 hist.col = "#00AFBB", # color  
 density = TRUE, #show histograms  
 ellipses = TRUE,#annotate correlation elliopses  
 smoother = TRUE,  
 stars = TRUE) #show significance)

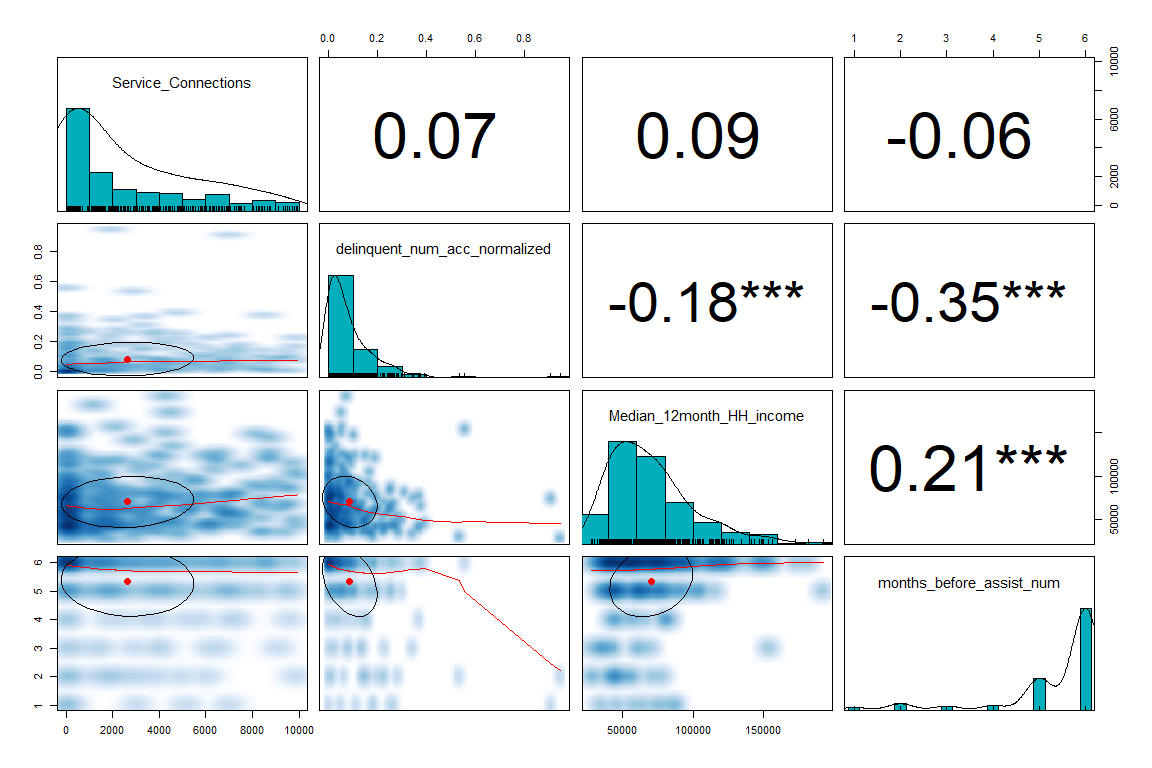
 We can see that service connections is a useful linear predictor for delinquent number of accounts (r = 0.52) and total cash reserve (r = 0.47). Interestingly, the delinquent number of accoutns and total cash reserves has a very non-linear relationship, resembling a volcano plot.

It may be better to visualize the number of delinquent accounts and cash totals on a log10 scale.

#trim data  
aux.response\_log <- allSmalls.requested.responded %>%   
 select(Service\_Connections, delinquent\_num\_acc, cash\_reserve\_total) %>%   
 mutate(log.d.n.acc = log10(delinquent\_num\_acc)) %>%   
 mutate(log.cash.total = log10(cash\_reserve\_total)) %>%   
 drop\_na() %>%   
 filter(log.d.n.acc > 0) %>%   
 filter(log.cash.total > 0) %>%   
 select(Service\_Connections, log.d.n.acc, log.cash.total)  
# plot matrix  
pairs.panels(aux.response\_log,  
 method = "pearson",  
 hist.col = "#00AFBB", # color  
 density = TRUE, #show histograms  
 ellipses = TRUE,#annotate correlation elliopses  
 smoother = TRUE,  
 stars = TRUE) #show significance)

 After log transformation we can see more linear relationships between the predictor (service connections) and response variables. Delinquent number of accounts normalized to service connections is a critical response variable that may be predicted from auxiliary variables.

#trim data  
aux.response <- allSmalls.requested.responded %>%   
 mutate(logServiceConnections = log10(Service\_Connections)) %>%   
 select(Service\_Connections, delinquent\_num\_acc\_normalized, Median\_12month\_HH\_income, months\_before\_assist\_num)  
# plot matrix  
pairs.panels(aux.response,  
 method = "pearson",  
 hist.col = "#00AFBB", # color  
 density = TRUE, #show histograms  
 ellipses = TRUE,#annotate correlation elliopses  
 smoother = TRUE,  
 stars = TRUE) #show significance)



Here we can see that service connections and median 12 month household income are not good predictors of the service-connection *normalized* number of delinquent accounts. However, a strong relationship exists between the *normalized* number of delinquent accounts and the qualitative variable for months before assistance needed. This is not useful for raking, as neither variables are auxiliary, however this is an interesting (and expected) correlation. Again, median 12 month household income and months before assistance needed are significantly correlated.

To continue calibration, we will now perform more formal analyses. Using all reliable, complete auxiliary data present, let’s see the linear relationships to the continuous variable months before assistance needed (from factor).

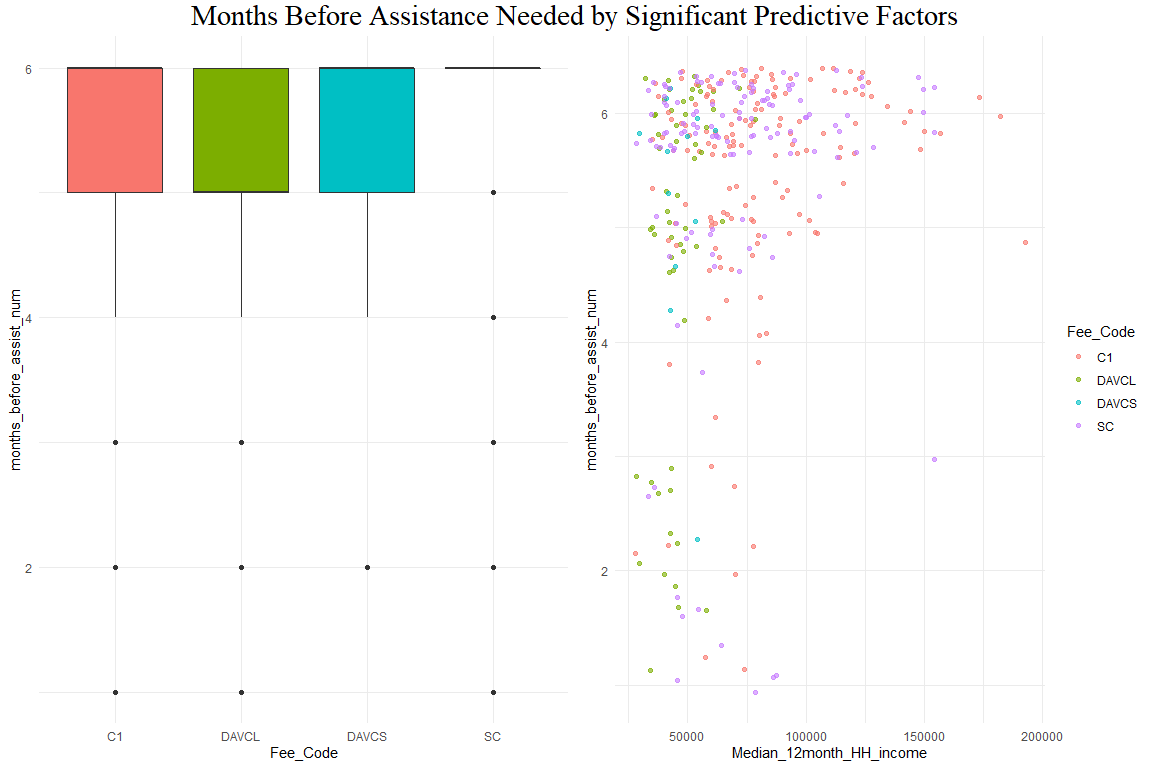
m2 <- glm(months\_before\_assist\_num ~ Service\_Connections + Median\_12month\_HH\_income + Median\_rent\_pct\_income + Fee\_Code, na.action = na.exclude, data = allSmalls.requested.responded)  
summary(m2)

##   
## Call:  
## glm(formula = months\_before\_assist\_num ~ Service\_Connections +   
## Median\_12month\_HH\_income + Median\_rent\_pct\_income + Fee\_Code,   
## data = allSmalls.requested.responded, na.action = na.exclude)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.7491 -0.2229 0.3951 0.6825 1.3910   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.137306312 0.527438955 7.844 0.0000000000000615  
## Service\_Connections -0.000025077 0.000033778 -0.742 0.458373  
## Median\_12month\_HH\_income 0.000008711 0.000002563 3.399 0.000761  
## Median\_rent\_pct\_income 0.021822315 0.012416149 1.758 0.079750  
## Fee\_CodeDAVCL -0.389227265 0.206986478 -1.880 0.060930  
## Fee\_CodeDAVCS 0.032420714 0.393655413 0.082 0.934412  
## Fee\_CodeSC 0.015550889 0.209543346 0.074 0.940886  
##   
## (Intercept) \*\*\*  
## Service\_Connections   
## Median\_12month\_HH\_income \*\*\*  
## Median\_rent\_pct\_income .   
## Fee\_CodeDAVCL .   
## Fee\_CodeDAVCS   
## Fee\_CodeSC   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 1.419319)  
##   
## Null deviance: 502.33 on 335 degrees of freedom  
## Residual deviance: 466.96 on 329 degrees of freedom  
## (41 observations deleted due to missingness)  
## AIC: 1080.1  
##   
## Number of Fisher Scoring iterations: 2

Again we can see that median 12 month household income is a strong linear predictor for months before assistance needed. A non-significant trend exists between disadvantaged smalls and months before assistance (as would be expected).

The relationship between fee code and months before assistance needed can be visualized as a boxplot.

p1 <- allSmalls.requested.responded %>%   
 ggplot(aes(x = Fee\_Code, y = months\_before\_assist\_num, fill = Fee\_Code)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 theme(legend.position = "none")  
p2 <- allSmalls.requested.responded %>%   
 ggplot(aes(x = Median\_12month\_HH\_income, y = months\_before\_assist\_num, color = Fee\_Code)) +  
 geom\_point(position = "jitter", alpha = 0.6) +  
 theme\_minimal()  
  
grid.arrange(p1, p2,ncol = 2,  
 top = textGrob("Months Before Assistance Needed by Significant Predictive Factors",  
 gp=gpar(fontsize = 22,font=6)))

 We can see quite clearly the relationship between fee code and months before assistance needed, with the disadvantaged community larges representing the most at-risk overall, and the community smalls representing lowest risk. Again remember that these values are unweighted and are only used for selecting auxiliary variables for raking.Nonetheless, the relationship between median 12 month household income and months before assistance needed seems to be quite robust. There are, however, some outliers that may negatively affect raking. We may choose to remove these outliers for raking purposes.

## rstudent unadjusted p-value Bonferroni p  
## 305 -3.933773 0.000083623 0.028097  
## 341 -3.925684 0.000086484 0.029059  
## 259 -3.875518 0.000106400 0.035750

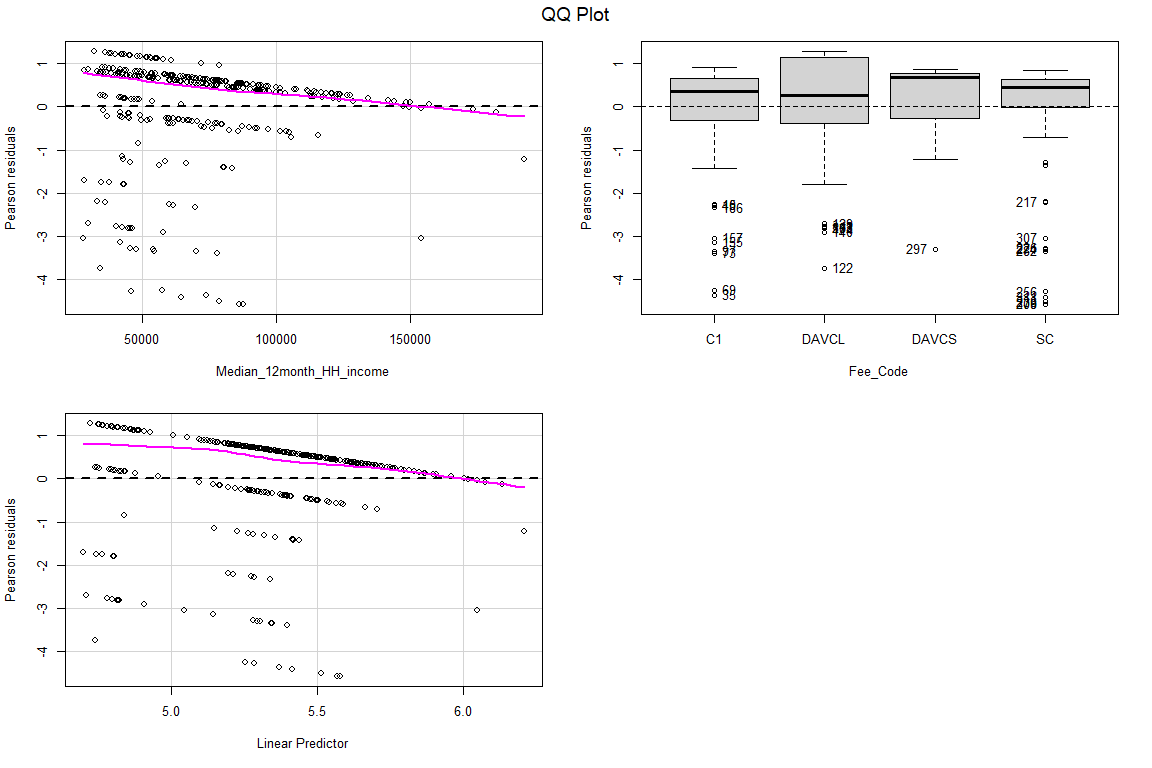
#code row number  
allSmalls.requested.responded %<>%   
 mutate(id = row\_number())  
#examine cases  
allSmalls.requested.responded %>%  
 filter(id == 305 | id == 341 | id == 259)

Let’s look at residuals for predictive variables.

summary(m3)

##   
## Call:  
## glm(formula = months\_before\_assist\_num ~ Median\_12month\_HH\_income +   
## Fee\_Code, data = allSmalls.requested.responded, na.action = na.exclude)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.5772 -0.2820 0.4354 0.6860 1.2787   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.846150113 0.217354419 22.296 < 0.0000000000000002  
## Median\_12month\_HH\_income 0.000007077 0.000002411 2.935 0.00357  
## Fee\_CodeDAVCL -0.350304495 0.205185521 -1.707 0.08871  
## Fee\_CodeDAVCS 0.075870097 0.367880088 0.206 0.83673  
## Fee\_CodeSC 0.112144482 0.147551287 0.760 0.44777  
##   
## (Intercept) \*\*\*  
## Median\_12month\_HH\_income \*\*   
## Fee\_CodeDAVCL .   
## Fee\_CodeDAVCS   
## Fee\_CodeSC   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 1.425995)  
##   
## Null deviance: 502.33 on 335 degrees of freedom  
## Residual deviance: 472.00 on 331 degrees of freedom  
## (41 observations deleted due to missingness)  
## AIC: 1079.7  
##   
## Number of Fisher Scoring iterations: 2

residualPlots(m3, main="QQ Plot") #qq plot for studentized resid



## Test stat Pr(>|Test stat|)  
## Median\_12month\_HH\_income 2.0059 0.1567  
## Fee\_Code

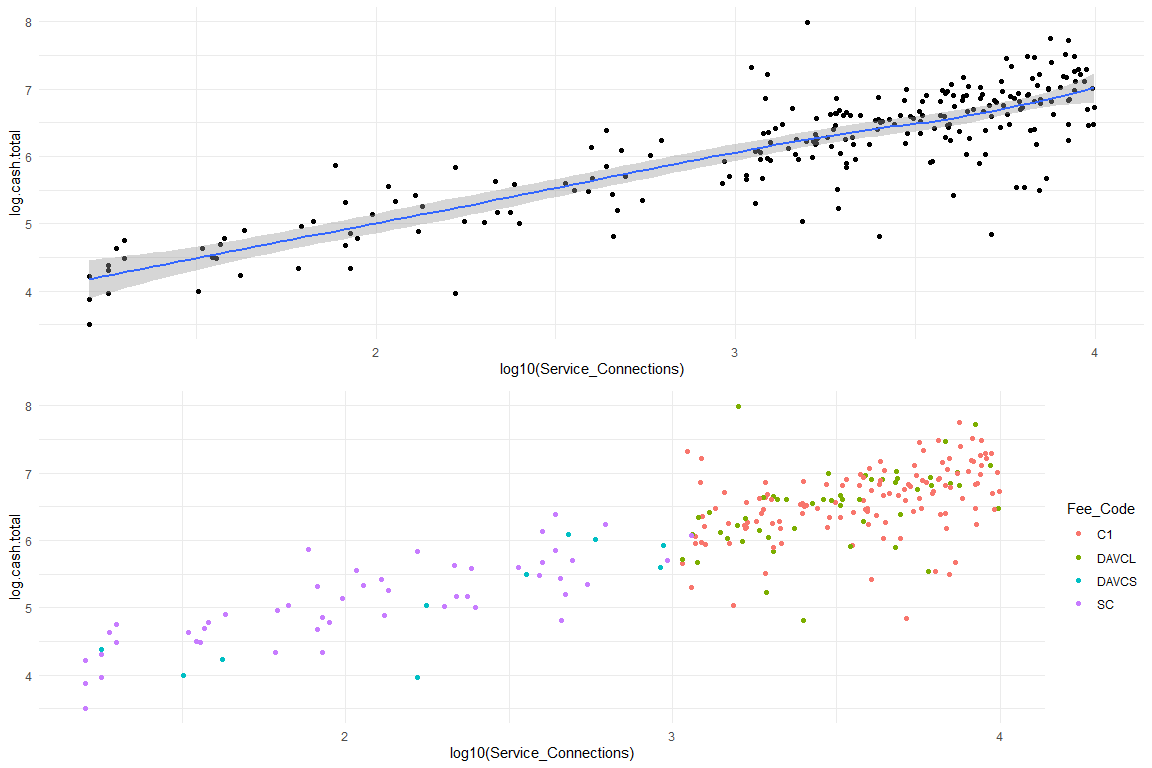
Since there are considerable deviant residuals for this model, let’s try a more simple model with an intuitive and likely simpler dependent variable (total cash reserves).

#create simple dataset for modeling that drops missing values and provides log transform   
simpleModel <- allSmalls.requested.responded %>%   
 select(PWSID, base.weight, final.weight.nonresponse, Service\_Connections, delinquent\_num\_acc, cash\_reserve\_total, Fee\_Code, Median\_12month\_HH\_income) %>%   
 mutate(log.d.n.acc = log10(delinquent\_num\_acc)) %>%   
 mutate(log.cash.total = log10(cash\_reserve\_total)) %>%   
 drop\_na() %>%   
 filter(log.d.n.acc > 0) %>%   
 filter(log.cash.total > 0) %>%   
 select(Service\_Connections, log.d.n.acc, log.cash.total, PWSID, base.weight, final.weight.nonresponse, Service\_Connections, delinquent\_num\_acc, cash\_reserve\_total, Fee\_Code, Median\_12month\_HH\_income) %>%   
 droplevels()  
#build glm  
m5 <- glm(log.cash.total ~ Service\_Connections + Median\_12month\_HH\_income + Fee\_Code, na.action = na.exclude, data = simpleModel)  
#report  
summary(m5)

##   
## Call:  
## glm(formula = log.cash.total ~ Service\_Connections + Median\_12month\_HH\_income +   
## Fee\_Code, data = simpleModel, na.action = na.exclude)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7542 -0.2662 0.0561 0.3109 1.6958   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 5.976528956 0.132823414 44.996 < 0.0000000000000002  
## Service\_Connections 0.000100326 0.000016102 6.231 0.00000000218  
## Median\_12month\_HH\_income 0.000001885 0.000001379 1.367 0.173  
## Fee\_CodeDAVCL 0.062504988 0.099081745 0.631 0.529  
## Fee\_CodeDAVCS -1.030640099 0.190510953 -5.410 0.00000015735  
## Fee\_CodeSC -1.042674500 0.113021073 -9.225 < 0.0000000000000002  
##   
## (Intercept) \*\*\*  
## Service\_Connections \*\*\*  
## Median\_12month\_HH\_income   
## Fee\_CodeDAVCL   
## Fee\_CodeDAVCS \*\*\*  
## Fee\_CodeSC \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.2821804)  
##   
## Null deviance: 174.486 on 236 degrees of freedom  
## Residual deviance: 65.184 on 231 degrees of freedom  
## AIC: 380.65  
##   
## Number of Fisher Scoring iterations: 2

Here we can see a much simpler model with a lower AIC (380.65) compared to the models that tried to predict months before assistance needed (AIC = 1079.7). We will use this simple relationship to calibrate the data. Let’s see this highly correlated relationship in graphical form.

p1 <- simpleModel %>%   
 ggplot(aes(x = log10(Service\_Connections), y = log.cash.total)) +  
 geom\_point() +  
 geom\_smooth()+  
 theme\_minimal()   
  
p2 <- simpleModel %>%   
 ggplot(aes(x = log10(Service\_Connections), y = log.cash.total, color = Fee\_Code)) +  
 geom\_point() +  
 theme\_minimal()   
grid.arrange(p1, p2)

 Here we can see the strong linear relationship between cash total and service connections, which will form the basis for building a model to rake the data.

### Calibration

Weights are now calibrated to auxiliary variables.

#### Small Systems

The code below uses the *sampling* package to select a sample with probability proportional to the median 12 month household income . The method of selection is to randomize the order of the population and then select a systematic sample (Hartley and Rao 1962).

The discrete breaks for median 12-month household income, calculated using the Jenk’s Natural Breaks method, are below.

Before calibration, non-response-adjusted weights are be examined for completeness.

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.323 1.530 2.522 6.249 11.035 25.184

Since there are no missing adjusted weights, calibration may proceed.

#if multiple post-strata are used, check that weights are calibrated  
kable(svytotal(~interaction(Fee\_Code, tag), ps.dsgn),  
 caption = "Interaction Between Post-Strata for Small Systems",  
 digits = 4)

Interaction Between Post-Strata for Small Systems

|  |  |  |
| --- | --- | --- |
|  | total | SE |
| interaction(Fee\_Code, tag)C1.Bin A | 0.0000 | 0.0000 |
| interaction(Fee\_Code, tag)DAVCL.Bin A | 0.0000 | 0.0000 |
| interaction(Fee\_Code, tag)DAVCS.Bin A | 238.0000 | 0.0000 |
| interaction(Fee\_Code, tag)SC.Bin A | 1697.5278 | 1.7103 |
| interaction(Fee\_Code, tag)C1.Bin B | 174.4956 | 4.7611 |
| interaction(Fee\_Code, tag)DAVCL.Bin B | 88.3369 | 3.1696 |
| interaction(Fee\_Code, tag)DAVCS.Bin B | 0.0000 | 0.0000 |
| interaction(Fee\_Code, tag)SC.Bin B | 2.4722 | 1.7103 |
| interaction(Fee\_Code, tag)C1.Bin C | 89.1467 | 3.2598 |
| interaction(Fee\_Code, tag)DAVCL.Bin C | 25.5965 | 2.7052 |
| interaction(Fee\_Code, tag)DAVCS.Bin C | 0.0000 | 0.0000 |
| interaction(Fee\_Code, tag)SC.Bin C | 0.0000 | 0.0000 |
| interaction(Fee\_Code, tag)C1.Bin D | 68.3577 | 2.2319 |
| interaction(Fee\_Code, tag)DAVCL.Bin D | 11.0665 | 1.2730 |
| interaction(Fee\_Code, tag)DAVCS.Bin D | 0.0000 | 0.0000 |
| interaction(Fee\_Code, tag)SC.Bin D | 0.0000 | 0.0000 |

Let’s compare the weights calculated from post-stratification with the non-reponse-adjusted weights and the volunteer imputed survey weights.

#examine new weights  
kable(rbind(summary(weights(srv.dsgn)), summary(weights(ps.dsgn)), summary(weights(ps.dsgn.imputed.volunteers))),  
 caption = "Weights Comparison Following Post-Stratification for Small Systems",  
 digits = 3)

Weights Comparison Following Post-Stratification for Small Systems

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| 1.323 | 1.530 | 2.522 | 6.249 | 11.035 | 25.184 |
| 1.383 | 1.682 | 2.657 | 6.353 | 10.734 | 24.795 |
| 1.262 | 1.733 | 2.611 | 5.757 | 10.227 | 13.222 |

We can see that post-stratification by fee code alone did not alter the weights significantly. This is not surprising due to the original survey design. However, significantly smaller weights are avilable for the post-stratified imputed survey set with volunteers (line 3).

The estimated proportion of fee codes and their their SEs are reported below for the post-stratified.

kable(svytotal(~Fee\_Code, ps.dsgn, na.rm = TRUE),  
 caption = "Post-Stratification estimates of Systems per Fee Code")

Post-Stratification estimates of Systems per Fee Code

|  |  |  |
| --- | --- | --- |
|  | total | SE |
| Fee\_CodeC1 | 332 | 0 |
| Fee\_CodeDAVCL | 125 | 0 |
| Fee\_CodeDAVCS | 238 | 0 |
| Fee\_CodeSC | 1700 | 0 |

## Fee\_CodeC1 Fee\_CodeDAVCL Fee\_CodeDAVCS   
## 0.00000000000000001404514 0.00000000000000003453847 0.00000000000000005681339   
## Fee\_CodeSC   
## 0.00000000000000004320784

If we compare these values to those in the original survey design (without post-stratification), we can see the differences in totals and standard errors.

kable(svytotal(~Fee\_Code, srv.dsgn, na.rm = TRUE),  
 digits = 4,  
 caption = "Pre-Stratification Estimates of Systems per Fee Code")

Pre-Stratification Estimates of Systems per Fee Code

|  |  |  |
| --- | --- | --- |
|  | total | SE |
| Fee\_CodeC1 | 302.0304 | 9.0372 |
| Fee\_CodeDAVCL | 119.5614 | 8.4429 |
| Fee\_CodeDAVCS | 186.7495 | 44.4154 |
| Fee\_CodeSC | 1747.6587 | 61.8026 |

## Fee\_CodeC1 Fee\_CodeDAVCL Fee\_CodeDAVCS Fee\_CodeSC   
## 0.02992148 0.07061537 0.23783391 0.03536308

## total SE  
## Fee\_CodeC1 332 0  
## Fee\_CodeDAVCL 125 0  
## Fee\_CodeDAVCS 238 0  
## Fee\_CodeSC 1700 0

Let’s see how the change in weights changes means for response variables.

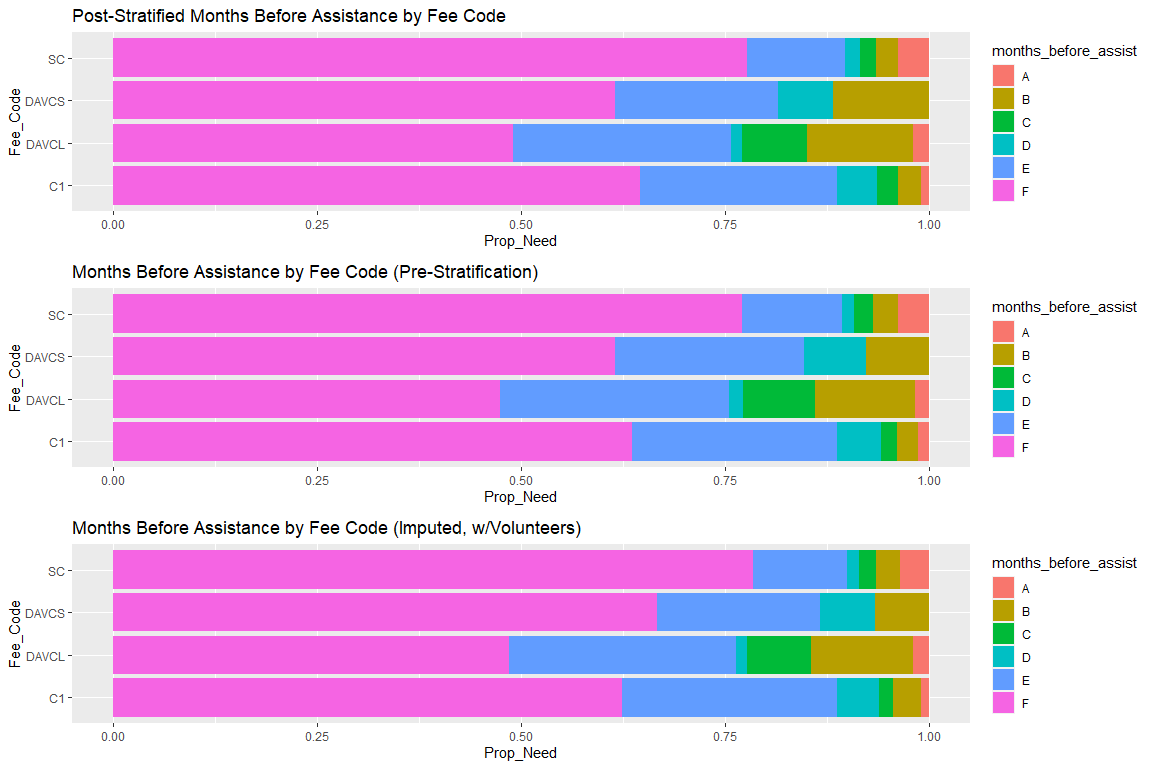
#calcualte unweighted means  
srv.mean.unweighted <- svymean(~cash\_reserve\_total + cash\_reserve\_restricted + cash\_reserve\_unrestricted +  
 months\_before\_assist\_num + delinquent\_num\_acc +   
 delinquent\_amount\_dollars,   
 srv.dsgn.unweighted, na.rm = TRUE)  
srv.mean.unweighted <- data.frame(srv.mean.unweighted)  
#reassign names  
srv.mean.unweighted <- setNames(cbind(rownames(srv.mean.unweighted), srv.mean.unweighted, row.names = NULL),  
 c("variable", "srv.Mean.unweighted", "srv.SE.unweighted"))   
  
#calcualte un-adjusted weights means  
srv.mean.unadjusted <- svymean(~cash\_reserve\_total + cash\_reserve\_restricted + cash\_reserve\_unrestricted +  
 months\_before\_assist\_num + delinquent\_num\_acc +   
 delinquent\_amount\_dollars,   
 srv.dsgn.unadjusted, na.rm = TRUE)  
srv.mean.unadjusted <- data.frame(srv.mean.unadjusted)  
#reassign names  
srv.mean.unadjusted <- setNames(cbind(rownames(srv.mean.unadjusted), srv.mean.unadjusted, row.names = NULL),  
 c("variable", "srv.Mean.unadjusted", "srv.SE.unadjusted"))   
  
#calcualte survey means  
srv.mean <- svymean(~cash\_reserve\_total + cash\_reserve\_restricted + cash\_reserve\_unrestricted +  
 months\_before\_assist\_num + delinquent\_num\_acc +   
 delinquent\_amount\_dollars,   
 srv.dsgn, na.rm = TRUE)  
srv.mean <- data.frame(srv.mean)  
#reassign names  
srv.mean <- setNames(cbind(rownames(srv.mean), srv.mean, row.names = NULL),  
 c("variable", "srv.Mean", "srv.SE"))   
#calcualte post-stratification means  
post.strat.mean <- svymean(~cash\_reserve\_total + cash\_reserve\_restricted +  
 cash\_reserve\_unrestricted +  
 months\_before\_assist\_num + delinquent\_num\_acc +   
 delinquent\_amount\_dollars,  
 ps.dsgn, na.rm = TRUE)  
  
post.strat.mean <- data.frame(post.strat.mean)  
post.strat.mean <- setNames(cbind(rownames(post.strat.mean), post.strat.mean, row.names = NULL),  
 c("variable", "post.strat.Mean", "post.strate.SE"))  
  
#calcualte post-stratified means for IMPUTED data  
srv.mean.imputed.volunteers <- svymean(~cash\_reserve\_total + cash\_reserve\_restricted + cash\_reserve\_unrestricted  
 +  
 months\_before\_assist\_num + delinquent\_num\_acc +   
 delinquent\_amount\_dollars,   
 srv.dsgn.imputed.volunteers, na.rm = TRUE)  
srv.mean.imputed.volunteers <- data.frame(srv.mean.imputed.volunteers)  
#reassign names  
srv.mean.imputed.volunteers <- setNames(cbind(rownames(srv.mean.imputed.volunteers), srv.mean.imputed.volunteers, row.names = NULL),  
 c("variable", "srv.mean.imputed.volunteers", "srv.SE.imputed.volunteers"))   
  
  
#join tables  
smallsSurveySummary <- left\_join(left\_join(left\_join(left\_join(srv.mean,post.strat.mean, by = "variable"), srv.mean.unadjusted),srv.mean.unweighted), srv.mean.imputed.volunteers)  
#save  
# write.csv(smallsSurveySummary,  
# file = "R/output/smallsSurveySummary.csv")  
#print  
kable(smallsSurveySummary)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| variable | srv.Mean | srv.SE | post.strat.Mean | post.strate.SE | srv.Mean.unadjusted | srv.SE.unadjusted | srv.Mean.unweighted | srv.SE.unweighted | srv.mean.imputed.volunteers | srv.SE.imputed.volunteers |
| cash\_reserve\_total | 2598181.106343 | 357593.5158799 | 2777232.253704 | 379886.4581813 | 2885224.336375 | 376400.0635541 | 6253564.708 | 508603.4104307 | 2612290.741904 | 320275.086109 |
| cash\_reserve\_restricted | 1174506.502467 | 176728.1797411 | 1247737.213428 | 187778.5011243 | 1294389.077524 | 188058.5944320 | 2717783.336 | 271316.1283184 | 1189253.254811 | 158377.845895 |
| cash\_reserve\_unrestricted | 1423099.105766 | 190613.0910411 | 1528903.078129 | 202046.8861951 | 1590226.291164 | 199373.4483943 | 3534690.974 | 274817.5182424 | 1590537.463924 | 221787.765429 |
| months\_before\_assist\_num | 5.469616 | 0.1090541 | 5.466149 | 0.1052986 | 5.447426 | 0.1128974 | 5.250 | 0.0698904 | 5.466202 | 0.096449 |
| delinquent\_num\_acc | 116.842693 | 12.3033905 | 123.803253 | 12.6260560 | 132.376153 | 12.7767566 | 326.859 | 21.2170535 | 117.989410 | 10.585028 |
| delinquent\_amount\_dollars | 39570.682922 | 3948.8522954 | 41953.490575 | 4077.5253871 | 44920.184821 | 4073.1190765 | 112497.612 | 6975.3936626 | 39812.597222 | 3372.701015 |

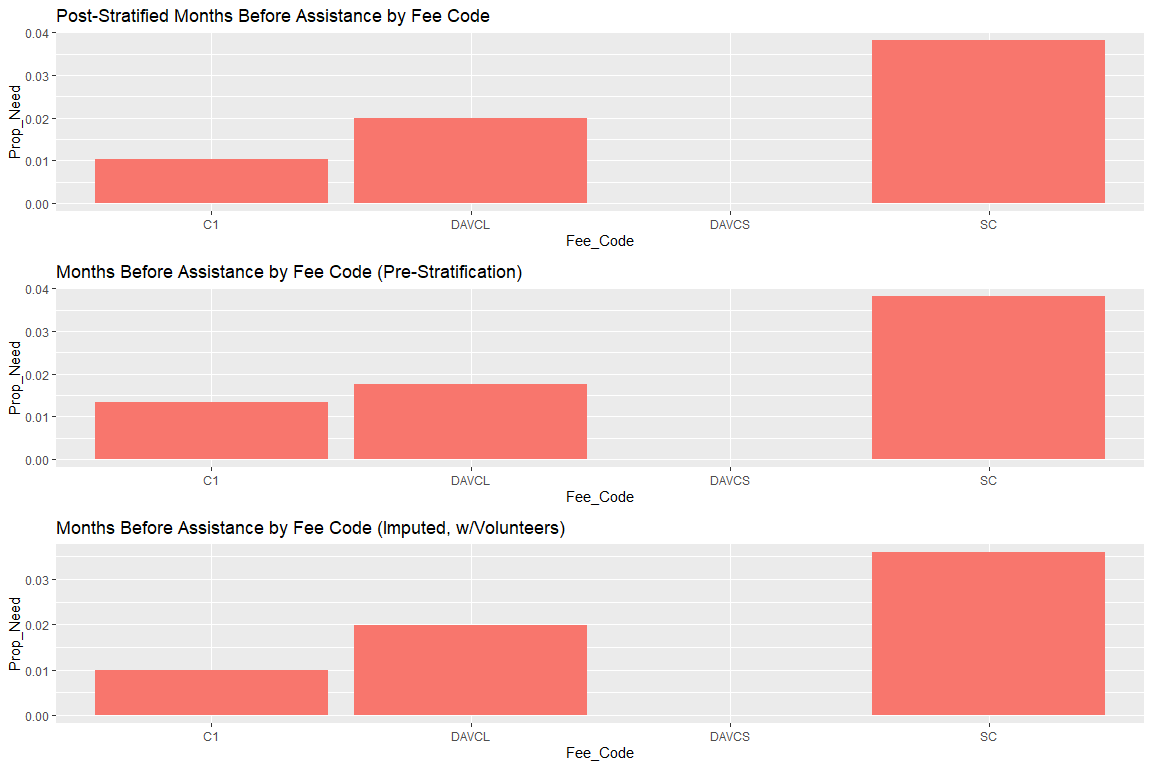
We can see that adjusted for non-response and post-stratifying by fee code has increased the standard error and the mean relative to no adjustment for most variables.

# construct and display a frequency table STRATIFIED  
tab\_fee\_need\_cond <- svytable(~Fee\_Code + months\_before\_assist,  
 design = ps.dsgn) %>%  
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_Fee\_Code = sum(Freq), Prop\_Need = Freq/n\_Fee\_Code) %>%  
 ungroup()  
# repeate for unweighted  
tab\_fee\_need\_cond\_unwt <- svytable(~Fee\_Code + months\_before\_assist,  
 design = srv.dsgn.unweighted) %>%  
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_Fee\_Code = sum(Freq), Prop\_Need = Freq/n\_Fee\_Code) %>%  
 ungroup()  
#repeat for volunteers and imputed data  
tab\_fee\_need\_cond\_imputed <- svytable(~Fee\_Code + months\_before\_assist,  
 design = srv.dsgn.imputed.volunteers) %>%  
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_Fee\_Code = sum(Freq), Prop\_Need = Freq/n\_Fee\_Code) %>%  
 ungroup()

# Create a segmented bar graph of the conditional proportions in table  
p1 <- ggplot(data = tab\_fee\_need\_cond,  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = months\_before\_assist)) +   
 geom\_col() +   
 coord\_flip() +  
 labs(title = "Post-Stratified Months Before Assistance by Fee Code",  
 xlab = "Proportion")  
#before strat  
p2 <- ggplot(data = tab\_fee\_need\_cond\_unwt,  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = months\_before\_assist)) +   
 geom\_col() +   
 coord\_flip() +  
 labs(title = "Months Before Assistance by Fee Code (Pre-Stratification)",  
 xlab = "Proportion")   
# Imputed with volunteers  
p3 <- ggplot(data = tab\_fee\_need\_cond\_imputed,  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = months\_before\_assist)) +   
 geom\_col() +   
 coord\_flip() +  
 labs(title = "Months Before Assistance by Fee Code (Imputed, w/Volunteers)",  
 xlab = "Proportion")   
  
grid.arrange(p1, p2, p3)

 We can see that the weighing does not apprecitably change the results. However there does seem to be a slight change in the number of disadvantaged community larges with immediate need in the volunteer group with imputation.

# Create a segmented bar graph of the conditional proportions in table  
p1 <- tab\_fee\_need\_cond %>%   
 filter(months\_before\_assist == "A") %>%   
 ggplot(mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = months\_before\_assist)) +   
 geom\_col() +  
  
 labs(title = "Post-Stratified Months Before Assistance by Fee Code",  
 xlab = "Proportion")+  
 theme(legend.position = "none")  
#before strat  
p2 <- tab\_fee\_need\_cond\_unwt %>%   
 filter(months\_before\_assist == "A") %>%   
 ggplot(  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = months\_before\_assist)) +   
 geom\_col() +   
 labs(title = "Months Before Assistance by Fee Code (Pre-Stratification)",  
 xlab = "Proportion") +  
 theme(legend.position = "none")  
# Imputed with volunteers  
p3 <- tab\_fee\_need\_cond\_imputed %>%   
 filter(months\_before\_assist == "A") %>%   
 ggplot(  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = months\_before\_assist)) +   
 geom\_col() +   
 labs(title = "Months Before Assistance by Fee Code (Imputed, w/Volunteers)",  
 xlab = "Proportion") +  
 theme(legend.position = "none")  
  
grid.arrange(p1, p2, p3)

 Our survey has now been weighted to adequately represent fee code distributions.

This is repeated below for totals.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| variable | srv.total | srv.SE | post.strat.total | post.strate.SE | srv.total.unadjusted | srv.SE.unadjusted | srv.total.unweighted | srv.SE.unweighted | srv.total.imputed.volunteers | srv.SE.imputed.volunteers |
| cash\_reserve\_total | 1595868290.755 | 175870376.7282 | 1709722768.526 | 182842729.841 | 1293805101.151 | 140200645.6173 | 882254962 | 76140289.3808 | 1392819163.460 | 151549021.8798 |
| cash\_reserve\_restricted | 729262550.041 | 94314592.1354 | 776789181.362 | 97660777.140 | 589864288.980 | 75176864.6470 | 398068424 | 41663422.4330 | 650558663.026 | 75669232.6847 |
| cash\_reserve\_unrestricted | 866177026.195 | 90026506.4567 | 932485371.323 | 94447506.030 | 703614665.297 | 71867154.0118 | 484016539 | 39379919.3431 | 847365938.145 | 122239314.2664 |
| months\_before\_assist\_num | 3006.182 | 316.3216 | 3072.969 | 323.288 | 2278.416 | 229.4597 | 709 | 34.8936 | 2573.024 | 246.9053 |
| delinquent\_num\_acc | 84238.064 | 5839.1655 | 90658.228 | 6255.301 | 68601.616 | 4751.9456 | 49122 | 3130.9001 | 72337.928 | 4798.1437 |
| Total number of delinquent residential accounts | 76489.644 | 5807.4330 | 82351.701 | 6243.688 | 62402.261 | 4729.7527 | 44303 | 3128.5803 | 65743.867 | 4775.0988 |
| delinquent\_amount\_dollars | 28234207.086 | 1726393.9606 | 30417721.417 | 1865480.817 | 23043351.000 | 1411395.0525 | 16775044 | 1012221.5295 | 24539704.789 | 1476837.6671 |

## variable srv.total srv.SE post.strat.total post.strate.SE  
## 1 delinquent\_amount\_dollars 53566965 2305146 57595916 2470236  
## srv.total.unadjusted srv.SE.unadjusted srv.total.unweighted srv.SE.unweighted  
## 1 42872920 1728195 29264025 1130439  
## srv.total.imputed.volunteers srv.SE.imputed.volunteers  
## 1 51910335 3983377

#calcualte unweighted total accounts  
srv.total.unweighted.accounts <- as.data.frame(svytotal(~`Total number of delinquent residential accounts`,   
 srv.dsgn.unweighted, na.rm = TRUE))  
#reassign names  
srv.total.unweighted.accounts <- setNames(cbind(rownames(srv.total.unweighted.accounts), srv.total.unweighted.accounts, row.names = NULL),  
 c("variable", "srv.total.unweighted", "srv.SE.unweighted"))   
  
#calcualte un-adjusted weights total accounts  
srv.total.unadjusted.accounts <- as.data.frame(svytotal(~`Total number of delinquent residential accounts`,   
 srv.dsgn.unadjusted, na.rm = TRUE))  
#reassign names  
srv.total.unadjusted.accounts <- setNames(cbind(rownames(srv.total.unadjusted.accounts), srv.total.unadjusted.accounts, row.names = NULL),  
 c("variable", "srv.total.unadjusted", "srv.SE.unadjusted"))   
  
#calcualte survey total accounts  
srv.total.accounts <- as.data.frame(svytotal(~`Total number of delinquent residential accounts`,   
 srv.dsgn, na.rm = TRUE))  
#reassign names  
srv.total.accounts <- setNames(cbind(rownames(srv.total.accounts), srv.total.accounts, row.names = NULL),  
 c("variable", "srv.total", "srv.SE"))   
#calcualte post-stratification totals  
post.strat.total.accounts <- as.data.frame(svytotal(~`Total number of delinquent residential accounts`,  
 ps.dsgn, na.rm = TRUE))  
  
post.strat.total.accounts <- setNames(cbind(rownames(post.strat.total.accounts), post.strat.total.accounts, row.names = NULL),  
 c("variable", "post.strat.total", "post.strate.SE"))  
  
#calcualte post-stratified totals for IMPUTED data  
srv.total.imputed.volunteers.accounts <- as.data.frame(svytotal(~`Total number of delinquent residential accounts`,   
 srv.dsgn.imputed.volunteers, na.rm = TRUE))  
  
#reassign names  
srv.total.imputed.volunteers.accounts <- setNames(cbind(rownames(srv.total.imputed.volunteers.accounts), srv.total.imputed.volunteers.accounts, row.names = NULL),  
 c("variable", "srv.total.imputed.volunteers", "srv.SE.imputed.volunteers"))   
  
  
#join tables  
smallsSurveySummaryTotalaccounts <- left\_join(left\_join(left\_join(left\_join(srv.total.accounts,post.strat.total.accounts, by = "variable"), srv.total.unadjusted.accounts),srv.total.unweighted.accounts), srv.total.imputed.volunteers.accounts)  
# #save  
# write.csv(smallsSurveySummaryTotalaccounts,  
# file = "R/output/smallsSurveySummaryTotalaccounts.csv")  
#print  
smallsSurveySummaryTotalaccounts %>%   
 t() %>%   
 kable(caption = "Total number of delinquent residential account Estimate by Stratification")

Total number of delinquent residential account Estimate by Stratification

|  |  |
| --- | --- |
| variable | Total number of delinquent residential accounts |
| srv.total | 131419 |
| srv.SE | 6819.591 |
| post.strat.total | 142034.2 |
| post.strate.SE | 7491.594 |
| srv.total.unadjusted | 106265.4 |
| srv.SE.unadjusted | 5476.787 |
| srv.total.unweighted | 71954 |
| srv.SE.unweighted | 3326.587 |
| srv.total.imputed.volunteers | 112566.2 |
| srv.SE.imputed.volunteers | 5683.73 |

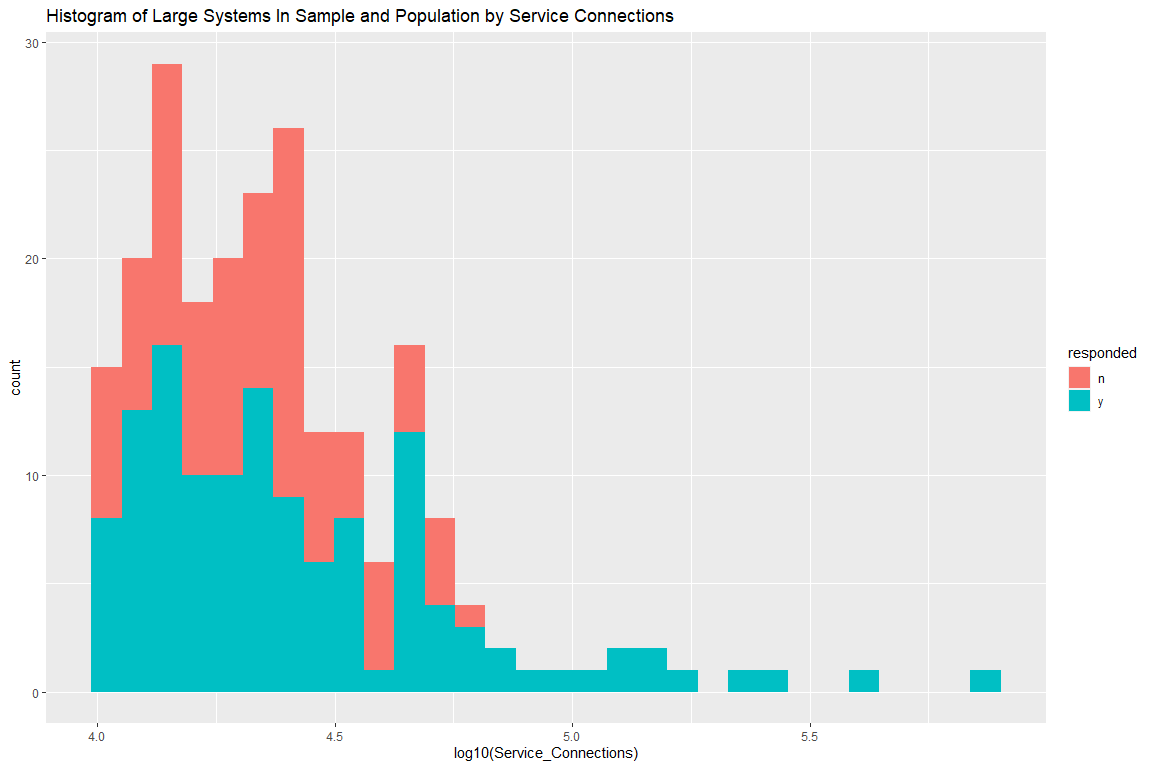
#### Large systems

The above steps are repeated for large systems, using service connections asa proxy for representativeness. Jenk’s Natural Breaks are used to bin systems naturally by service connections for calibration.

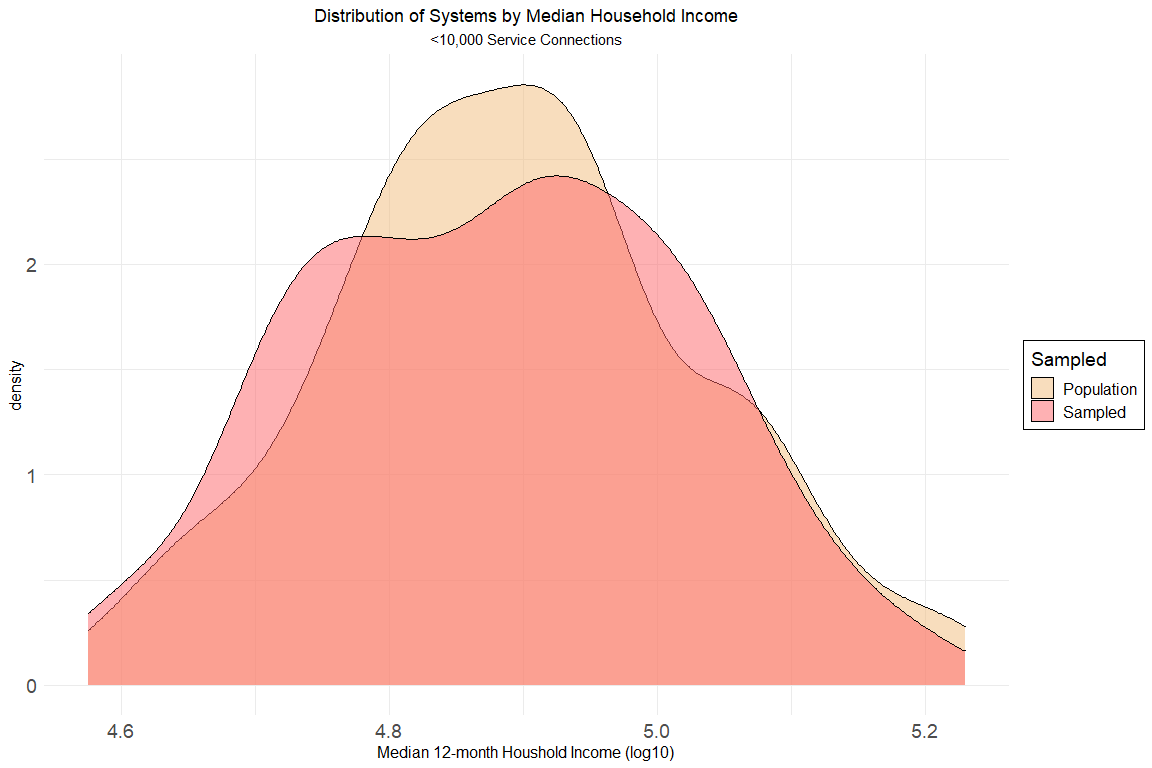
## mean var  
## 1 34878.89 3635566334

Below we can see the mean and variance of service connections for large systems visualized as histrograms.

allLarges %>%   
 ggplot(aes(x = log10(Service\_Connections), fill = responded)) +  
 geom\_histogram() +  
 labs(title = "Histogram of Large Systems In Sample and Population by Service Connections")



require(wesanderson)  
allLarges %>%   
 ggplot(aes(x = log10(Median\_12month\_HH\_income), fill = responded)) +  
 geom\_density(alpha = 0.5)+  
 scale\_fill\_manual(values = wes\_palette("GrandBudapest1"),  
 name = "Sampled",  
 labels = c("Population", "Sampled")) +  
 xlab("Median 12-month Houshold Income (log10)") +  
 labs(title = "Distribution of Systems by Median Household Income",  
 subtitle = "<10,000 Service Connections") +  
 theme\_minimal() +  
 theme(  
 legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 14),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))

 Natural breaks of service connections and median 12-month household income are determined using the Jenk’s method.

#### Natural Breaks ####  
#Determine natural breaks and assign  
breaks <- getJenksBreaks(allLarges$Service\_Connections, 4)  
  
#specify bin labels  
postStrata <- c("A", "B", "C")#, "D")#, "E", "F")#, "G", "H", "I")  
  
#bucket values into bins  
bins <- cut(allLarges$Service\_Connections,  
 breaks = breaks,  
 include.lowest = TRUE,  
 right = FALSE,  
 labels = postStrata)  
  
#inspect bins  
kable(summary(bins),  
 caption = "Number of Large Systems per Post-Stratified Bin (By Service Connection)")

Number of Large Systems per Post-Stratified Bin (By Service Connection)

|  |  |
| --- | --- |
|  | x |
| A | 213 |
| B | 8 |
| C | 2 |

A summary of the number of systems in each bin ( by service connections) is shown above.

#Store group as new column  
allLarges <-as\_tibble(allLarges) %>%   
 mutate(postStrata = case\_when(  
 Service\_Connections >= breaks[1] & Service\_Connections < breaks[2] ~postStrata[1],  
 Service\_Connections >= breaks[2] & Service\_Connections < breaks[3] ~postStrata[2],  
 Service\_Connections >= breaks[3] & Service\_Connections <= breaks[4] ~postStrata[3],  
 ))  
  
#tag is character vector, so convert to factor  
allLarges$postStrata <- factor(allLarges$postStrata,  
 ordered = FALSE)  
breaks

## [1] 10008 105731 389835 709135

The discrete breaks for service connections is above. These steps are repeated for median 12-month household income below.

#### Natural Breaks ####  
#Determine natural breaks and assign  
breaks <- getJenksBreaks(allLarges$Median\_12month\_HH\_income, 4)  
  
#specify bin labels  
postStrata\_income <- c("A", "B", "C")#, "D")#, "E", "F")#, "G", "H", "I")  
  
#bucket values into bins  
bins <- cut(allLarges$Median\_12month\_HH\_income,  
 breaks = breaks,  
 include.lowest = TRUE,  
 right = FALSE,  
 labels = postStrata\_income)  
  
#inspect bins  
kable(summary(bins),  
 caption = "Post-Strata bins for large systems by income")

Post-Strata bins for large systems by income

|  |  |
| --- | --- |
|  | x |
| A | 92 |
| B | 81 |
| C | 44 |
| NA’s | 6 |

A summary of the number of systems in each bin (by median 12-month household income) is shown above.

#Store group as new column  
allLarges <-as\_tibble(allLarges) %>%   
 mutate(postStrata\_income = case\_when(  
 Median\_12month\_HH\_income >= breaks[1] & Median\_12month\_HH\_income < breaks[2] ~postStrata\_income[1],  
 Median\_12month\_HH\_income >= breaks[2] & Median\_12month\_HH\_income < breaks[3] ~postStrata\_income[2],  
 Median\_12month\_HH\_income >= breaks[3] & Median\_12month\_HH\_income <= breaks[4] ~postStrata\_income[3],  
 ))  
  
#replace missing value with median (B)  
allLarges$postStrata\_income %<>%   
 replace\_na("B")  
  
#tag is character vector, so convert to factor  
allLarges$postStrata\_income <- factor(allLarges$postStrata\_income,  
 ordered = FALSE)  
breaks

## [1] 37597.92 71235.12 103552.52 169636.28

The discrete breaks for the three bins of income are displayed above (in US dollars).

Now that we have defined strata to adjust the data with, calibration can be carried out. These will also be used to calibrate the surveys.

Before calibration, non-response-adjusted weights are examined for completeness.

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 1.229 1.294 1.371 1.409 1.489 1.981 3

It seems that 3 out of 150 weights are missing, likely due to missing data for predictor variables (likely median household income). Since this is a very small percentage of the total, replacement with mean values is reasonable.

#several weights are missing. Replace with average values  
larges$final.weight.nonresponse %<>%   
 replace\_na(1.412023)  
  
#examine adjusted weights  
summary(larges$final.weight.nonresponse)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.229 1.295 1.374 1.409 1.482 1.981

Now that there are no missing adjusted weights, calibration may proceed,

#build survey design with no weights  
srv.dsgn.unweighted.larges <- svydesign(ids = ~0,  
 strata = ~tag,  
 data = larges,  
 fpc = ~fpc,  
 weights = ~NULL)  
#build survey design with non-adjusted weights  
srv.dsgn.unadjusted.larges <- svydesign(ids = ~ 0, # no clusters  
 strata = ~tag,  
 data = larges,  
 fpc = ~fpc,  
 weights = ~base.weight)  
  
# #build survey design with adjusted non-response weights  
srv.dsgn.adjusted.larges <- svydesign(ids = ~ 0, # no clusters  
 strata = ~tag,  
 data = larges,  
 fpc = ~fpc,  
 weights = ~final.weight.nonresponse)  
  
#PostStratify by fee code  
ps.dsgn.fee.code <- postStratify(design = srv.dsgn.adjusted.larges,  
 strata = ~Fee\_Code,  
 partial = TRUE,  
 population = N.PS)  
#Post stratify further by service connections  
ps.dsgn.fee.code.SC <- postStratify(design = ps.dsgn.fee.code,  
 strata = ~postStrata,  
 partial = TRUE,  
 population = N.SC)  
#post-stratify by fee code, service connections, and income  
ps.dsgn.larges <- postStratify(design = srv.dsgn.adjusted.larges,  
 strata = #~Fee\_Code +   
 ~postStrata + postStrata\_income,  
 partial = TRUE,  
 population = N.PS.SC)

Weights have now been adjusted for median household income and service connections. The interaction of these weights are seen below.

#if multiple post-strata are used, check that weights are calibrated  
kable(svytotal(~interaction(postStrata, postStrata\_income), ps.dsgn.larges),  
 caption = "Estimates of Systems per Strata (Interaction of post-strata)")

Estimates of Systems per Strata (Interaction of post-strata)

|  |  |  |
| --- | --- | --- |
|  | total | SE |
| interaction(postStrata, postStrata\_income)A.A | 87 | 0 |
| interaction(postStrata, postStrata\_income)B.A | 4 | 0 |
| interaction(postStrata, postStrata\_income)C.A | 1 | 0 |
| interaction(postStrata, postStrata\_income)A.B | 85 | 0 |
| interaction(postStrata, postStrata\_income)B.B | 1 | 0 |
| interaction(postStrata, postStrata\_income)C.B | 1 | 0 |
| interaction(postStrata, postStrata\_income)A.C | 41 | 0 |
| interaction(postStrata, postStrata\_income)B.C | 3 | 0 |
| interaction(postStrata, postStrata\_income)C.C | 0 | 0 |

Let’s compare the weights calculated from post-stratification with the non-reponse-adjusted weights and the imputed survey weights.

#examine new weights  
weights.summary <- rbind(summary(weights(srv.dsgn.unadjusted.larges)),summary(weights(srv.dsgn.adjusted.larges)),  
 summary(weights(ps.dsgn.fee.code)), summary(weights(ps.dsgn.fee.code.SC)), summary(weights(ps.dsgn.larges)))  
#name rows  
rownames(weights.summary) <- c("Unadjusted", "Non-Response Adjusted", "Post-Stratified (fee codes)", "Post-Stratified(Fee Code/Srv.Conn)", "Post-Stratified (Income/Srv.Conn.)")  
kable(weights.summary,  
 caption = "Summary Statistics for Weights for Larges")

Summary Statistics for Weights for Larges

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| Unadjusted | 1.7421875 | 1.742188 | 1.742188 | 1.742188 | 1.742188 | 1.742188 |
| Non-Response Adjusted | 1.2288037 | 1.294544 | 1.373848 | 1.408894 | 1.482179 | 1.980632 |
| Post-Stratified (fee codes) | 1.4926195 | 1.601936 | 1.694713 | 1.742188 | 1.877126 | 2.404272 |
| Post-Stratified(Fee Code/Srv.Conn) | 0.9531179 | 1.641447 | 1.738562 | 1.742188 | 1.927730 | 2.469088 |
| Post-Stratified (Income/Srv.Conn.) | 0.9746389 | 1.653418 | 1.755933 | 1.742188 | 1.862976 | 2.480353 |

We can see that post-stratification by fee code alone (second row) did not alter the weights significantly. Further post-stratification by service connection natural breaks and fee codes increased the range of weights, and notably lowered the minimum weights below nominal (demonstrating over-representation in the original sample). Post-stratification by service connections and median 12-month household income is virtually identical as post-stratification by fee code and service connections, albeit slightly right-shifted. Either will do.

The estimated proportion of fee codes and their their SEs are reported below for the post-stratified.

kable(svytotal(~Fee\_Code, ps.dsgn.larges, na.rm = TRUE),  
 caption = "Post-Stratified Estimated of Systems within Fee Codes for Larges",  
 digits = 3)

Post-Stratified Estimated of Systems within Fee Codes for Larges

|  |  |  |
| --- | --- | --- |
|  | total | SE |
| Fee\_CodeC1 | 205.252 | 3.196 |
| Fee\_CodeDAVCL | 17.748 | 3.196 |

## Fee\_CodeC1 Fee\_CodeDAVCL   
## 0.01557195 0.18008124

If we compare these values to those in the original survey design (without post-stratification), we can see the differences in totals and standard errors.

kable(svytotal(~Fee\_Code, srv.dsgn.unadjusted.larges, na.rm = TRUE),  
 caption = "Pre-Stratified Estimated of Systems within Fee Codes for Larges")

Pre-Stratified Estimated of Systems within Fee Codes for Larges

|  |  |  |
| --- | --- | --- |
|  | total | SE |
| Fee\_CodeC1 | 203.83594 | 3.619859 |
| Fee\_CodeDAVCL | 19.16406 | 3.619859 |

A slight decrease in variance is noted with post-stratificaiton.

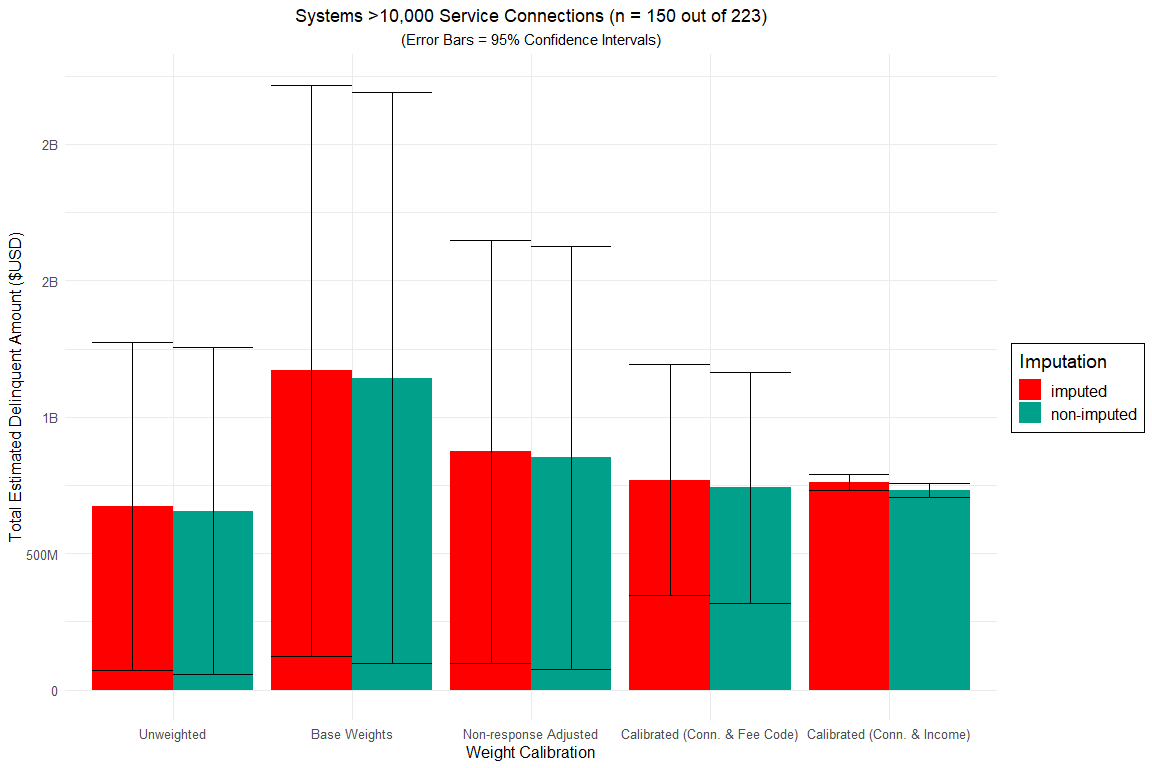
## Fee\_CodeC1 Fee\_CodeDAVCL   
## 0.01775869 0.18888788

## total SE  
## dollars\_del\_acc\_TotR 731881220 13748553

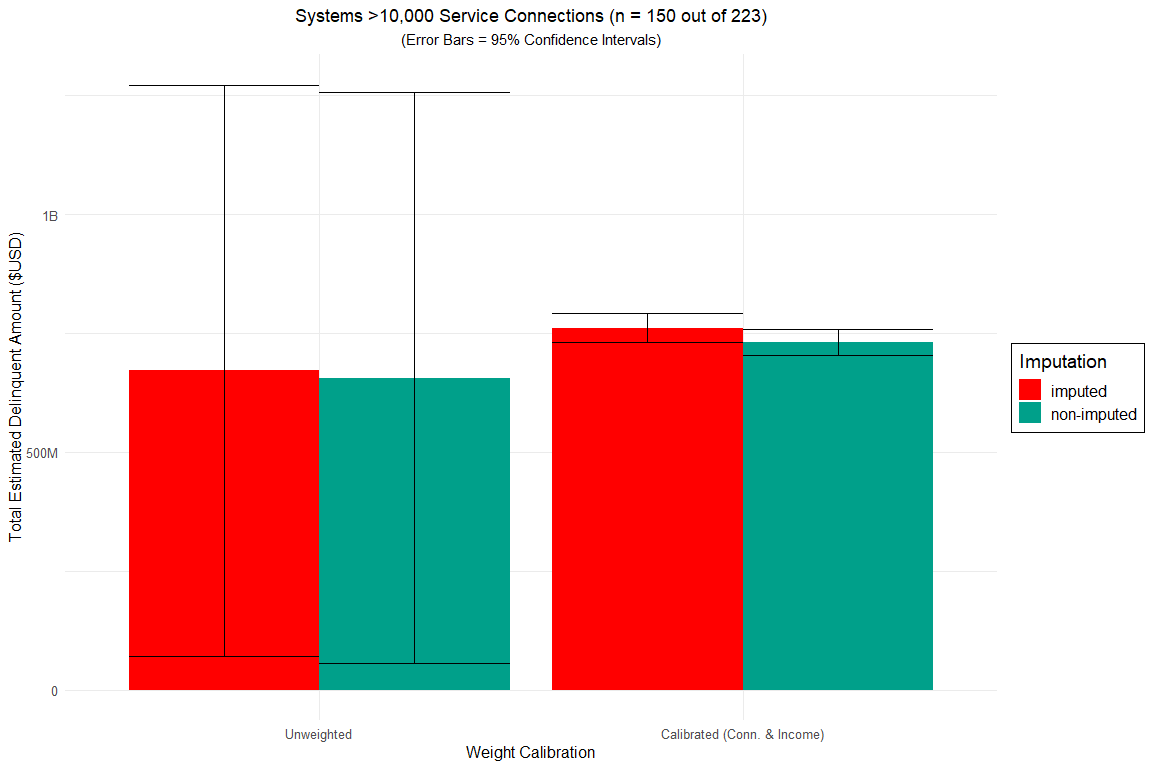
## total SE  
## dollars\_del\_acc\_TotR 1143909455 533991657

Post-stratification by service connections and median household income using Jenk’s Natural Breaks decreased the variance in estimates of statewide large water system delinquent accounts by a factor of 38.97 times! Further, weight adjustment using non-response propensity and calibration decreased the statewide estimate of debt by a factor of 1.56, or 156%. These values are visualized below.

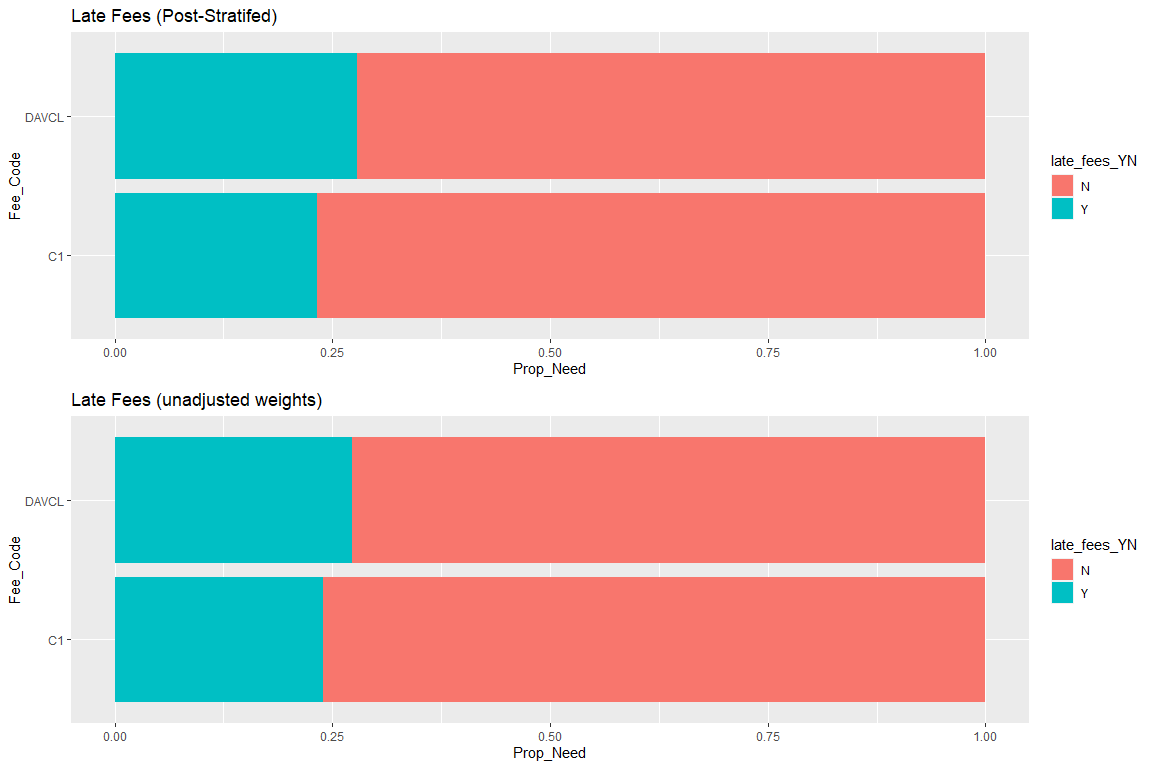
### Do for non-imputed ###  
  
#totals for no weights  
del\_dollars\_larges\_unweighted <- data.frame(svytotal(~dollars\_del\_acc\_TotR, srv.dsgn.unweighted.larges, na.rm = TRUE)) %>%   
 mutate(calibration = "Unweighted") %>%   
 mutate(imputed = "non-imputed")  
  
#totals for unadjusted weights (base weights)  
del\_dollars\_larges\_unadjusted <- data.frame(svytotal(~dollars\_del\_acc\_TotR, srv.dsgn.unadjusted.larges, na.rm = TRUE)) %>%   
 mutate(calibration = "Base Weights") %>%   
 mutate(imputed = "non-imputed")  
  
#totals for unadjusted weights (response propensity)  
del\_dollars\_larges\_adjusted <- data.frame(svytotal(~dollars\_del\_acc\_TotR, srv.dsgn.adjusted.larges, na.rm = TRUE)) %>%   
 mutate(calibration = "Non-response Adjusted") %>%   
 mutate(imputed = "non-imputed")  
  
#totals for adjusted weights (calibration)  
del\_dollars\_larges\_calibrated\_fee <- data.frame(svytotal(~dollars\_del\_acc\_TotR, ps.dsgn.fee.code.SC, na.rm = TRUE)) %>%   
 mutate(calibration = "Calibrated (Conn. & Fee Code)") %>%   
 mutate(imputed = "non-imputed")  
  
#totals for adjusted weights (calibration)  
del\_dollars\_larges\_calibrated <- data.frame(svytotal(~dollars\_del\_acc\_TotR, ps.dsgn.larges, na.rm = TRUE)) %>%  
 mutate(calibration = "Calibrated (Conn. & Income)") %>%   
 mutate(imputed = "non-imputed")  
  
#### Do for imputed ####  
#totals for no weights  
del\_dollars\_larges\_unweighted\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, srv.dsgn.unweighted.larges, na.rm = TRUE)) %>%   
 mutate(calibration = "Unweighted") %>%   
 mutate(imputed = "imputed")%>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
#totals for unadjusted weights (base weights)  
del\_dollars\_larges\_unadjusted\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, srv.dsgn.unadjusted.larges, na.rm = TRUE)) %>% mutate(calibration = "Base Weights") %>%   
 mutate(imputed = "imputed")%>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
#totals for unadjusted weights (response propensity)  
del\_dollars\_larges\_adjusted\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, srv.dsgn.adjusted.larges, na.rm = TRUE)) %>% mutate(calibration = "Non-response Adjusted") %>%   
 mutate(imputed = "imputed")%>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
#totals for adjusted weights (calibration)  
del\_dollars\_larges\_calibrated\_fee\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, ps.dsgn.fee.code.SC, na.rm = TRUE)) %>% mutate(calibration = "Calibrated (Conn. & Fee Code)") %>%   
 mutate(imputed = "imputed")%>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
#totals for adjusted weights (calibration)  
del\_dollars\_larges\_calibrated\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, ps.dsgn.larges, na.rm = TRUE)) %>% mutate(calibration = "Calibrated (Conn. & Income)") %>%   
 mutate(imputed = "imputed") %>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
#join dataframes  
del\_dollars\_Larges\_c1 <- rbind(del\_dollars\_larges\_unweighted,   
del\_dollars\_larges\_unadjusted, del\_dollars\_larges\_adjusted, del\_dollars\_larges\_calibrated\_fee, del\_dollars\_larges\_calibrated, del\_dollars\_larges\_unweighted\_imputed,   
del\_dollars\_larges\_unadjusted\_imputed, del\_dollars\_larges\_adjusted\_imputed, del\_dollars\_larges\_calibrated\_fee\_imputed, del\_dollars\_larges\_calibrated\_imputed)  
  
#add error bars  
del\_dollars\_Larges\_comparison <- mutate(del\_dollars\_Larges\_c1,  
 lower = total - 1.96\*dollars\_del\_acc\_TotR, #95% CI  
 upper = total + 1.96\*dollars\_del\_acc\_TotR) %>% #95% CI  
 mutate\_if(is.character, as.factor)  
  
require(wesanderson)  
#Construct a barplot  
del\_dollars\_Larges\_comparison %>%   
 mutate(calibration = fct\_relevel(calibration,  
 "Unweighted", "Base Weights", "Non-response Adjusted", "Calibrated (Conn. & Fee Code)", "Calibrated (Conn. & Income)")) %>% #manually reorder by calibration status  
ggplot(aes(x = calibration, y = total,   
 ymin = lower, ymax = upper, fill = imputed)) +  
 geom\_col(position = "dodge") +  
 scale\_x\_discrete(name = "Weight Calibration")+  
 scale\_fill\_manual(values = wes\_palette("Darjeeling1"),  
 name = "Imputation") +  
 scale\_y\_continuous(labels = scales::label\_number\_si()) +  
 labs(y = "Total Estimated Delinquent Amount ($USD)") +  
 geom\_errorbar(width = 0.9, position = "dodge") +  
 labs(title = "Systems >10,000 Service Connections (n = 150 out of 223)",  
 subtitle = "(Error Bars = 95% Confidence Intervals)") +  
 theme\_minimal() +  
 theme(  
 legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))

 The impact of weighting (and adjusting weights using calibration) on both estimate totals and error can be seen above.A simpler figure is below.

### Do for non-imputed ###  
  
#totals for no weights  
del\_dollars\_larges\_unweighted <- data.frame(svytotal(~dollars\_del\_acc\_TotR, srv.dsgn.unweighted.larges, na.rm = TRUE)) %>%   
 mutate(calibration = "Unweighted") %>%   
 mutate(imputed = "non-imputed")  
  
  
#totals for adjusted weights (calibration)  
del\_dollars\_larges\_calibrated <- data.frame(svytotal(~dollars\_del\_acc\_TotR, ps.dsgn.larges, na.rm = TRUE)) %>%  
 mutate(calibration = "Calibrated (Conn. & Income)") %>%   
 mutate(imputed = "non-imputed")  
  
#### Do for imputed ####  
#totals for no weights  
del\_dollars\_larges\_unweighted\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, srv.dsgn.unweighted.larges, na.rm = TRUE)) %>%   
 mutate(calibration = "Unweighted") %>%   
 mutate(imputed = "imputed")%>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
  
#totals for adjusted weights (calibration)  
del\_dollars\_larges\_calibrated\_imputed <- data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed, ps.dsgn.larges, na.rm = TRUE)) %>% mutate(calibration = "Calibrated (Conn. & Income)") %>%   
 mutate(imputed = "imputed") %>%   
 rename(c("dollars\_del\_acc\_TotR" = dollars\_del\_acc\_TotR.imputed))  
  
#join dataframes  
del\_dollars\_Larges\_c1 <- rbind(del\_dollars\_larges\_unweighted, del\_dollars\_larges\_calibrated, del\_dollars\_larges\_unweighted\_imputed,   
del\_dollars\_larges\_calibrated\_imputed)  
  
#add error bars  
del\_dollars\_Larges\_comparison <- mutate(del\_dollars\_Larges\_c1,  
 lower = total - 1.96\*dollars\_del\_acc\_TotR, #95% CI  
 upper = total + 1.96\*dollars\_del\_acc\_TotR) %>% #95% CI  
 mutate\_if(is.character, as.factor)  
  
require(wesanderson)  
#Construct a barplot  
del\_dollars\_Larges\_comparison %>%   
 mutate(calibration = fct\_relevel(calibration,  
 "Unweighted", "Calibrated (Conn. & Income)")) %>% #manually reorder by calibration status  
ggplot(aes(x = calibration, y = total,   
 ymin = lower, ymax = upper, fill = imputed)) +  
 geom\_col(position = "dodge") +  
 scale\_x\_discrete(name = "Weight Calibration")+  
 scale\_fill\_manual(values = wes\_palette("Darjeeling1"),  
 name = "Imputation") +  
 scale\_y\_continuous(labels = scales::label\_number\_si()) +  
 labs(y = "Total Estimated Delinquent Amount ($USD)") +  
 geom\_errorbar(width = 0.9, position = "dodge") +  
 labs(title = "Systems >10,000 Service Connections (n = 150 out of 223)",  
 subtitle = "(Error Bars = 95% Confidence Intervals)") +  
 theme\_minimal() +  
 theme(  
 legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))

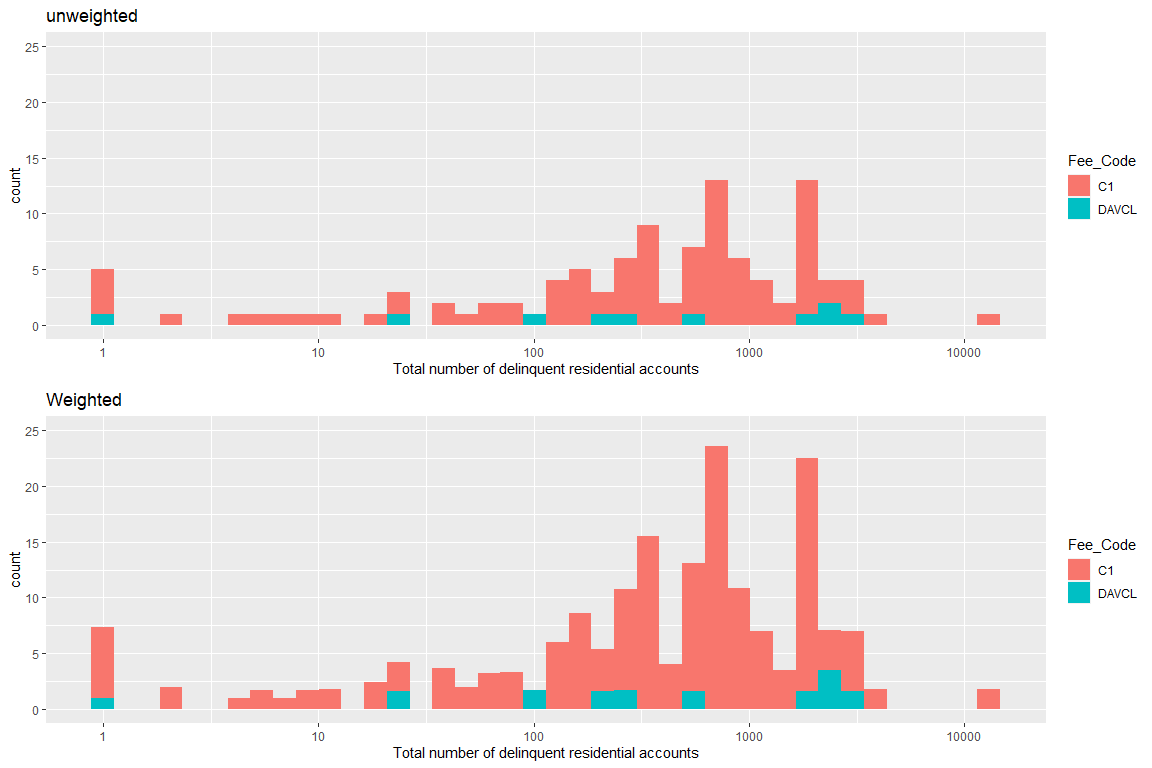


# construct and display a frequency table STRATIFIED  
tab\_fee\_need\_cond <- svytable(~Fee\_Code + late\_fees\_YN,  
 design = ps.dsgn.larges) %>%  
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_Fee\_Code = sum(Freq), Prop\_Need = Freq/n\_Fee\_Code) %>%  
 ungroup()  
# repeate for unadjusted weights  
tab\_fee\_need\_cond\_unwt <- svytable(~Fee\_Code + late\_fees\_YN,  
 design = srv.dsgn.unadjusted.larges) %>%  
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_Fee\_Code = sum(Freq), Prop\_Need = Freq/n\_Fee\_Code) %>%  
 ungroup()   
# Create a segmented bar graph of the conditional proportions in table  
p1 <- ggplot(data = tab\_fee\_need\_cond,  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = late\_fees\_YN)) +   
 geom\_col() +   
 coord\_flip() +  
 labs(title = "Late Fees (Post-Stratifed)",  
 xlab = "Proportion")  
#before strat  
p2 <- ggplot(data = tab\_fee\_need\_cond\_unwt,  
 mapping = aes(x = Fee\_Code, y = Prop\_Need, fill = late\_fees\_YN)) +   
 geom\_col() +   
 coord\_flip() +  
 labs(title = "Late Fees (unadjusted weights)",  
 xlab = "Proportion")  
  
grid.arrange(p1, p2)



Now to ensure adjustedweights sum up to the entire population from which each population was drawn.

weightedHist <- larges %>%   
 ggplot(aes(x = `Total number of delinquent residential accounts`, weight = final.weight, fill = Fee\_Code)) +  
 geom\_histogram(bins = 40)+  
 scale\_x\_log10() +  
 scale\_y\_continuous(limits = c(0,25)) +  
 labs(title = "Weighted")  
  
unweightedHist <- larges %>%   
 ggplot(aes(x = `Total number of delinquent residential accounts`, fill = Fee\_Code)) +  
 geom\_histogram(bins = 40)+  
 scale\_x\_log10()+  
 scale\_y\_continuous(limits = c(0,25)) +  
 labs(title = "unweighted")  
  
grid.arrange(unweightedHist,weightedHist)



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| variable | total.unweighted | total.SE.unweighted | total.unadjusted | SE.unadjusted | post.strat.total | post.strate.SE |
| dollars\_del\_acc\_TotR | 25664456 | 4730267.41269 | 44712294.350391 | 8241012.758046 | 39978966.413718 | 6596002.514981 |
| dollars\_del\_dw\_TotR | 15273472 | 2751818.70081 | 26609252.765691 | 4794184.142820 | 23879324.105850 | 4009448.330321 |
| Total number of delinquent residential accounts | 19615 | 4198.14480 | 34173.007812 | 7313.955402 | 34477.942435 | 7278.398545 |
| num\_del\_acc\_plan | 23261 | 14849.97251 | 40525.023438 | 25871.436478 | 24865.228955 | 13390.712378 |
| late\_fees\_YNN | 5 | 1.43630 | 8.710938 | 2.502303 | 9.398671 | 2.692535 |
| late\_fees\_YNY | 14 | 2.31379 | 24.390625 | 4.031056 | 23.102899 | 3.865042 |
| late\_fees\_dollars | 2528993 | 952205.03690 | 4405979.939922 | 1658919.712728 | 3436241.777560 | 1012560.933758 |

## variable srv.total.unadjusted srv.SE.unadjusted post.strat.total  
## 1 dollars\_del\_acc\_TotR 1143909455 533991657 731881220  
## post.strate.SE srv.total.unweighted srv.SE.unweighted  
## 1 13748553 656593768 306506422

## variable srv.total.unadjusted srv.SE.unadjusted post.strat.total  
## 1 dollars\_del\_dw\_TotR 294011783 94561239 199960210  
## post.strate.SE srv.total.unweighted srv.SE.unweighted  
## 1 7230338 168760127 54277303

## variable srv.total.unadjusted  
## 1 `Total number of delinquent residential accounts` 168238.3  
## srv.SE.unadjusted post.strat.total post.strate.SE srv.total.unweighted  
## 1 17621.41 170747 17404.82 96567.27  
## srv.SE.unweighted  
## 1 10114.53

Now that we have our adjusted weights, let’s extract them and bind to the dataframes.

# Summary Statistics

Summary statistics (totals, means, variances, ratios, and quantiles) are computed using adjusted weights below for small systems and larges.

## Small Systems (>10,000 Service Connections)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.9746 1.0000 1.5763 1.4023 1.7197 2.4804 89

Below is a table estimating the total statewide debt for small systems, grouped by fee code (total +- standard error). Lower and upper refer to 95% confidence intervals.

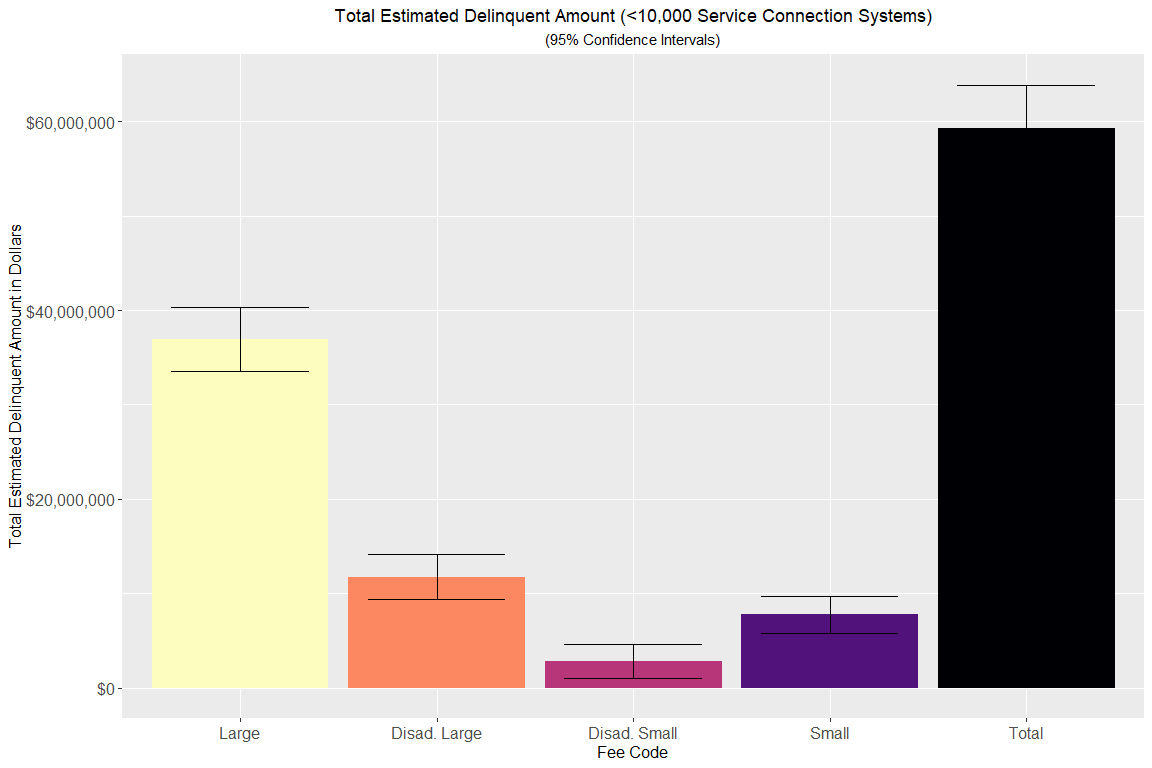
#smalls by fee code  
del\_dollars\_smalls <- as.data.frame(svyby(~delinquent\_amount\_dollars,   
 by = ~Fee\_Code,   
 design = smalls.dsgn,  
 FUN = svytotal,  
 na.rm = TRUE))  
#totals  
del\_dollars\_smalls\_total <- as.data.frame(svytotal(~delinquent\_amount\_dollars,   
 design = smalls.dsgn,  
 FUN = svytotal,  
 na.rm = TRUE)) %>%   
 mutate(Fee\_Code = "Total") %>%   
 rename(se = delinquent\_amount\_dollars) %>%   
 rename(delinquent\_amount\_dollars = total)   
   
#bind totals and fee codes  
del\_dollars\_smalls\_col <- rbind(del\_dollars\_smalls, del\_dollars\_smalls\_total) %>%   
 mutate(lower = delinquent\_amount\_dollars - 1.96\*se, #95% CI  
 upper = delinquent\_amount\_dollars + 1.96\*se)#95% CI  
#print table  
kable(del\_dollars\_smalls\_col,  
 caption = "Estimated total statewide debt for small systems, grouped by fee code")

Estimated total statewide debt for small systems, grouped by fee code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fee\_Code | delinquent\_amount\_dollars | se | lower | upper |
| C1 | 36961443 | 1737146.6 | 33556636 | 40366251 |
| DAVCL | 11742461 | 1211722.2 | 9367485 | 14117436 |
| DAVCS | 2837983 | 922044.2 | 1030777 | 4645190 |
| SC | 7774645 | 1011949.5 | 5791224 | 9758066 |
| Total | 59316532 | 2324207.5 | 54761085 | 63871979 |

This tabular data is plotted as a bar-graph below.

#Construct a barplot  
del\_dollars\_smalls\_col %>%   
ggplot(aes(x = Fee\_Code, y = delinquent\_amount\_dollars,   
 ymin = lower, ymax = upper, fill = Fee\_Code)) +  
 geom\_col() +  
 scale\_fill\_viridis\_d(option = "A", direction = -1) +  
 scale\_x\_discrete(name = "Fee Code",   
 labels = c("Large",   
 "Disad. Large",  
 "Disad. Small",   
 "Small",  
 "Total")) +  
 scale\_y\_continuous(labels = scales::dollar\_format()) +  
 labs(y = "Total Estimated Delinquent Amount in Dollars") +  
 geom\_errorbar(width = 0.7) +  
 labs(title = "Total Estimated Delinquent Amount (<10,000 Service Connection Systems)",  
 subtitle = "(95% Confidence Intervals)") +  
 theme(legend.position = "none",  
 legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))



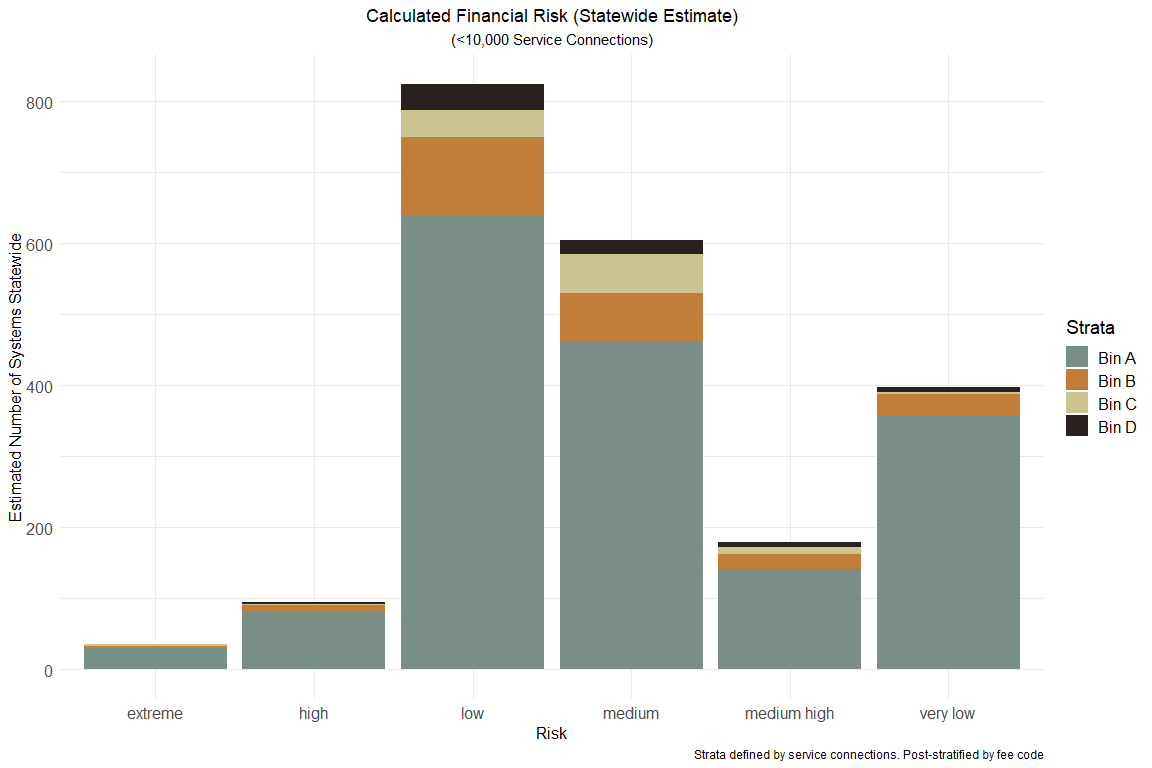
### Risk

Financial risk may be estimated in two distinct ways. A question in the survey regarding financial risk, asked as “how many months do you believe you’ll be able to go without assistance?” provides qualitative *perception* based estimates of risk. An alternative approach was also employed, in which quantitative reported information (i.e. restricted and unrestricted cash reserves) was used to estimate financial risk under the conservative assumption that current unrestricted funds are all that are available, with no further income. Both risk estimates are reported below.

#### Calculated Risk

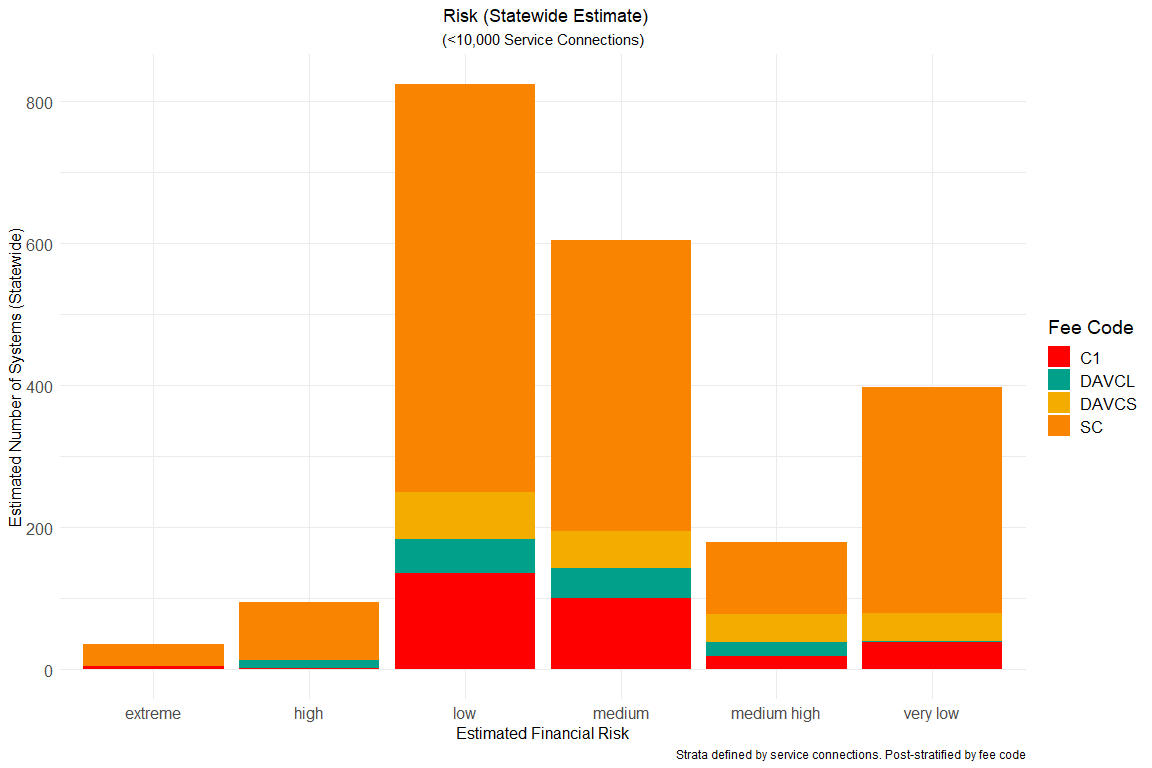
Calculated risk score (derived by **Marielle**), is calculated and plotted below for small systems.Statewide estimated totals by sampling strata are displayed in the below figure for the arbitrarily binned categories for calculated financial risk.

# construct and display a frequency table STRATIFIED  
tab\_tag\_risk\_cond <- svytable(~tag + Risk\_label,  
 design = smalls.dsgn) %>%   
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(tag) %>%  
 mutate(n\_tag = sum(Freq), Prop\_Risk = Freq/n\_tag) %>%  
 ungroup()  
  
# Create a segmented bar graph of the conditional proportions in table  
tab\_tag\_risk\_cond %>%   
ggplot(aes(x = Risk\_label, y = Freq, fill = tag)) +   
 geom\_col(position = position\_stack(reverse = TRUE)) +   
 labs(title = "Calculated Financial Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)",  
 caption = "Strata defined by service connections. Post-stratified by fee code") +  
 xlab("Risk") +  
 ylab("Estimated Number of Systems Statewide") +  
 scale\_fill\_manual(values = wes\_palette("Moonrise2"),  
 name = "Strata") +  
 theme\_minimal() +  
 theme(legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))



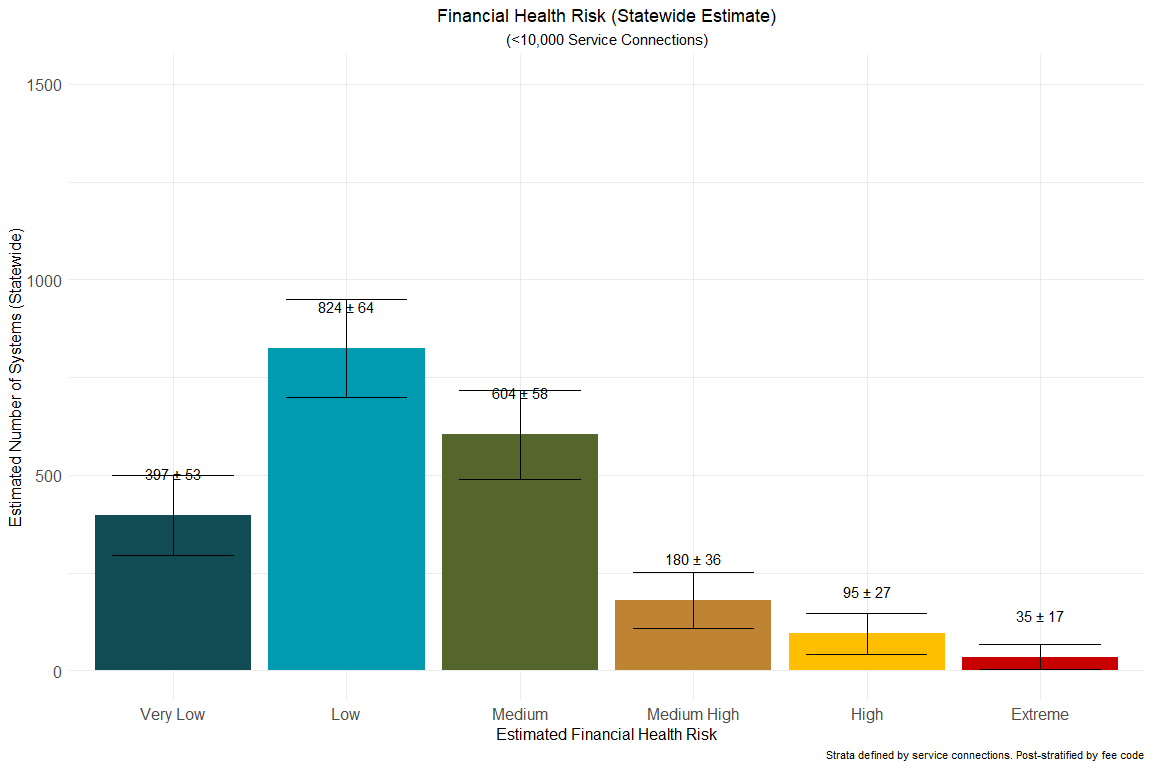
The same estimated values are plotted again with the fee code annotated.Note that uncertainties are not displayed for these plots due to over-plotting.

# construct and display a frequency table STRATIFIED  
tab\_feeCode\_risk\_cond <- svytable(~Fee\_Code + Risk\_label,  
 design = smalls.dsgn) %>%   
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_tag = sum(Freq), Prop\_Risk = Freq/n\_tag) %>%  
 ungroup()  
  
# Create a segmented bar graph of the conditional proportions in table  
tab\_feeCode\_risk\_cond %>%   
ggplot(aes(x = Risk\_label, y = Freq, fill = Fee\_Code)) +   
 geom\_col(position = position\_stack(reverse = TRUE)) +   
 labs(title = " Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)",  
 caption = "Strata defined by service connections. Post-stratified by fee code") +  
 xlab("Estimated Financial Risk") +  
 ylab("Estimated Number of Systems (Statewide)") +  
 scale\_fill\_manual(values = wes\_palette("Darjeeling1"),  
 name = "Fee Code") +  
 theme\_minimal() +  
 theme(legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))



These data are reported more simply as combined totals with 95% confidence intervals below.

tab\_risk <- svytotal(~Risk\_label,   
 design = smalls.dsgn,  
 na.rm = TRUE) %>%   
 data.frame()  
#reassign names  
tab\_risk <- set\_rownames(tab\_risk,  
 c("Extreme", "High", "Low", "Medium", "Medium High", "Very Low"))  
#convert rownames to column  
tab\_risk %<>%   
 rownames\_to\_column("Risk")  
  
#add lower and upper columns  
tab\_risk\_smalls <- mutate(tab\_risk,  
 lower = total - 1.96 \* SE, #95% CI  
 upper = total + 1.96 \* SE) %>%   
 mutate\_if(is.character, as.factor) %>%   
 mutate(Risk = fct\_relevel(Risk,  
 "Very Low", "Low", "Medium", "Medium High", "High", "Extreme"))  
  
#Construct a barplot  
require(calecopal)  
tab\_risk\_smalls %>%   
ggplot(aes(x = Risk, y = total,   
 ymin = lower, ymax = upper, fill = Risk)) +  
 geom\_col() +   
 geom\_errorbar(width = 0.7) +  
 geom\_text(aes(label = paste0(paste0(round(total)," \u00B1 "),round(SE))),  
 vjust = -3) +  
 labs(title = "Financial Health Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)",  
 caption = "Strata defined by service connections. Post-stratified by fee code") +  
 xlab("Estimated Financial Health Risk") +  
 ylab("Estimated Number of Systems (Statewide)") +  
 scale\_y\_continuous(limits = c(0,1500))+  
 scale\_fill\_manual(values = rev(cal\_palette("kelp1")), #reverse color order  
 name = "Strata") +  
 theme\_minimal() +  
 theme(legend.position = "none",  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5),  
 plot.caption = element\_text(size = 8))



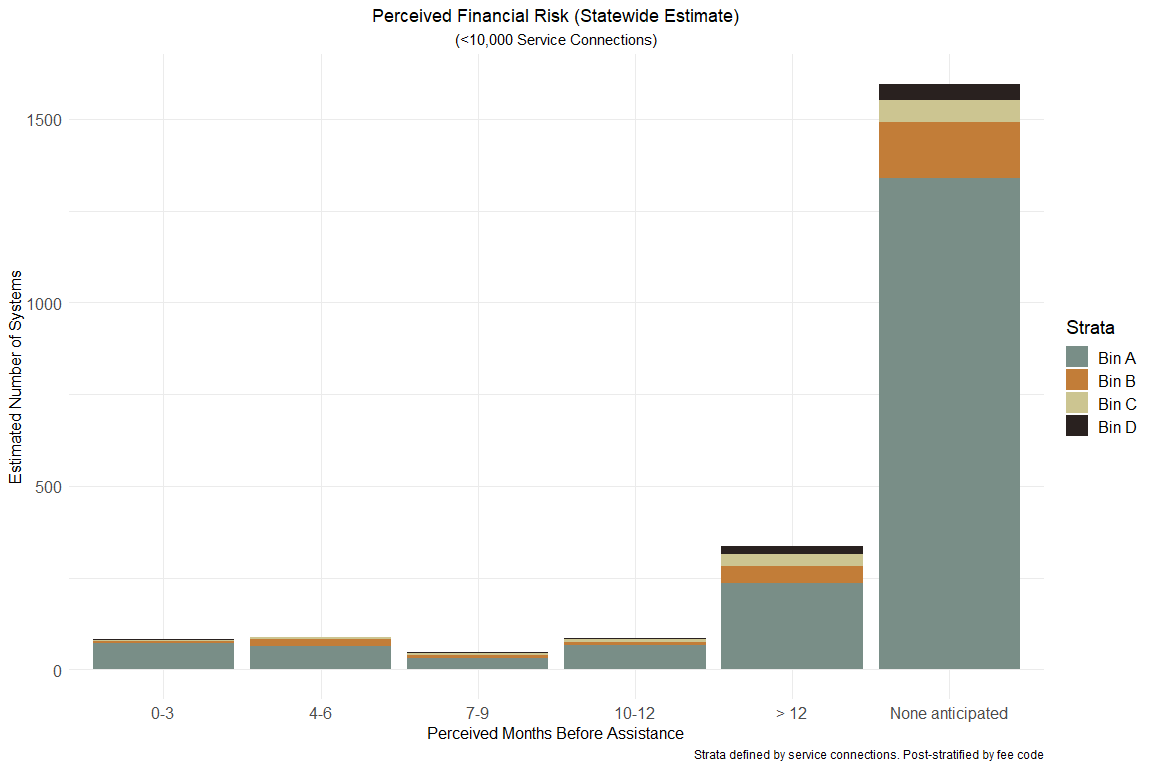
#### *Perceived* Risk

Perceived financial risk was reported by system interviewees in the form of the question “how many months do you estimate that you will be able to go for needing finanical assistance?” with discrete available responses:

A: 0-3 months B: 4-6 months C: 7-9 months D: 10-12 months E: Greater than 12 months F: No financial assistance anticipated

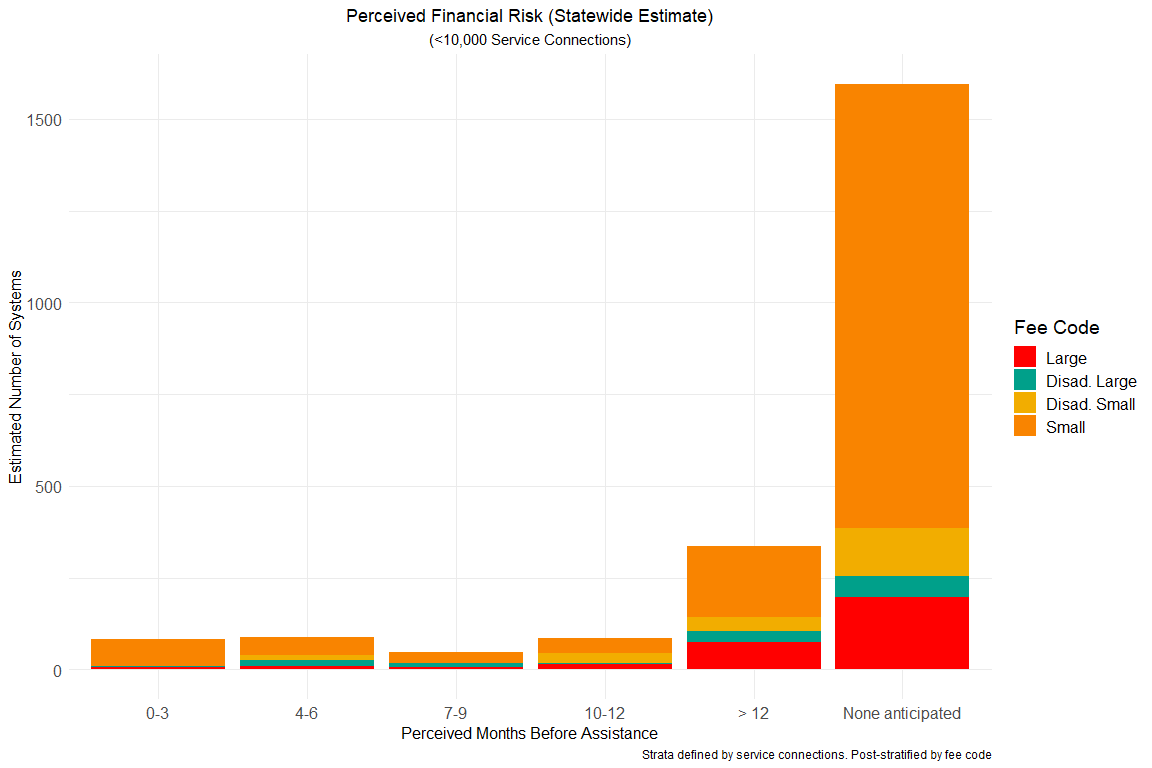
Statewide estimates for the number of water systems with less than 10,000 service connections experiencing *perceived* financial risk for the above discrete categories is plotted below by sampling strata.

# construct and display a frequency table STRATIFIED  
tab\_tag\_risk\_cond <- svytable(~tag + months\_before\_assist,  
 design = smalls.dsgn) %>%   
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(tag) %>%  
 mutate(n\_tag = sum(Freq), Prop\_Risk = Freq/n\_tag) %>%  
 ungroup()  
  
# Create a segmented bar graph of the conditional proportions in table  
tab\_tag\_risk\_cond %>%   
ggplot(aes(x = months\_before\_assist, y = Freq, fill = tag)) +   
 geom\_col(position = position\_stack(reverse = TRUE)) +   
 # geom\_text(aes(label = round(Freq)),  
 # vjust = -4) +  
 labs(title = "Perceived Financial Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)",  
 caption = "Strata defined by service connections. Post-stratified by fee code") +  
 xlab("Perceived Financial Risk") +  
 ylab("Estimated Number of Systems") +  
 scale\_fill\_manual(values = wes\_palette("Moonrise2"),  
 name = "Strata") +  
 scale\_x\_discrete(name = "Perceived Months Before Assistance",   
 labels = c("0-3",   
 "4-6",  
 "7-9",   
 "10-12",  
 "> 12",  
 "None anticipated")) +  
 theme\_minimal() +  
 theme(legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))



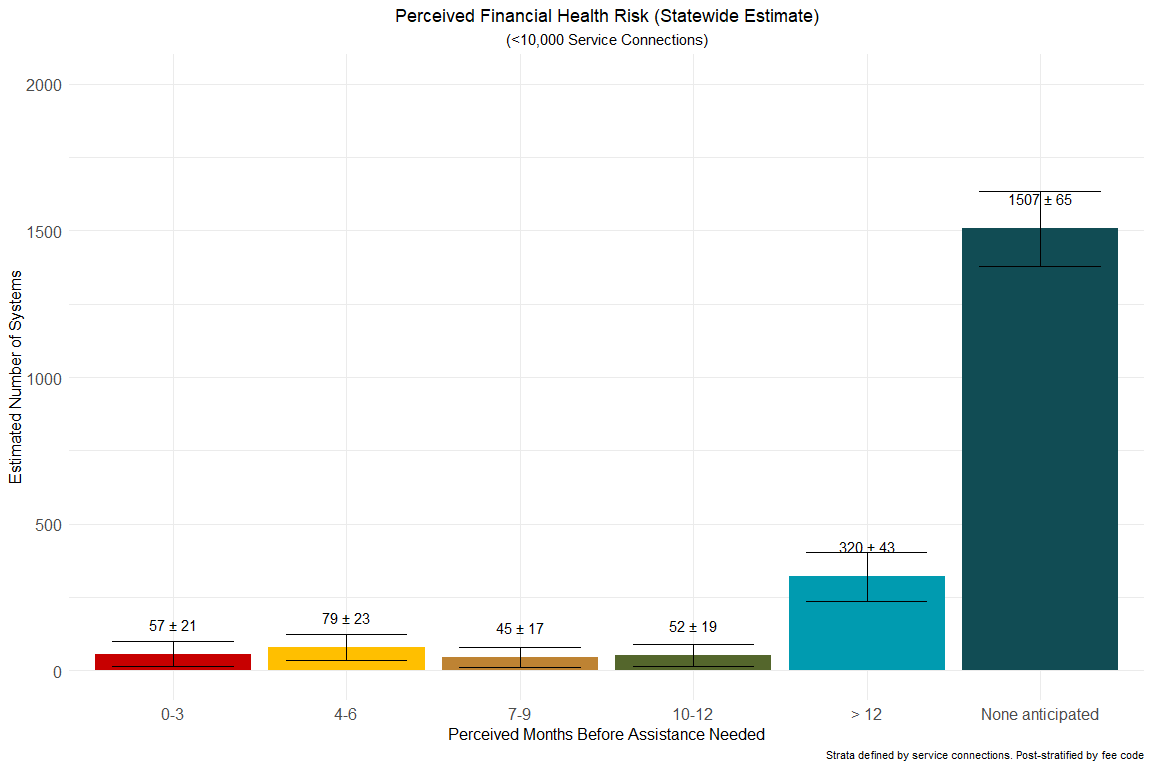
Statewide estimates for the number of water systems with less than 10,000 service connections experiencing *perceived* financial risk for the above discrete categories is plotted below by fee code.

# construct and display a frequency table STRATIFIED  
tab\_feeCode\_risk\_cond <- svytable(~Fee\_Code + months\_before\_assist,  
 design = smalls.dsgn) %>%   
 # Add conditional proportions to table  
 as.data.frame() %>%  
 group\_by(Fee\_Code) %>%  
 mutate(n\_tag = sum(Freq), Prop\_Risk = Freq/n\_tag) %>%  
 ungroup()  
  
# Create a segmented bar graph of the conditional proportions in table  
tab\_feeCode\_risk\_cond %>%   
ggplot(aes(x = months\_before\_assist, y = Freq, fill = Fee\_Code)) +   
 geom\_col(position = position\_stack(reverse = TRUE)) +   
 labs(title = "Perceived Financial Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)",  
 caption = "Strata defined by service connections. Post-stratified by fee code") +  
 xlab("Perceived Financial Risk") +  
 ylab("Estimated Number of Systems") +  
 scale\_fill\_manual(values = wes\_palette("Darjeeling1"),  
 name = "Fee Code",  
 labels = c("Large",  
 "Disad. Large",  
 "Disad. Small",  
 "Small",  
 "Total")) +  
 scale\_x\_discrete(name = "Perceived Months Before Assistance",   
 labels = c("0-3",   
 "4-6",  
 "7-9",   
 "10-12",  
 "> 12",  
 "None anticipated")) +  
 theme\_minimal() +  
 theme(legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))

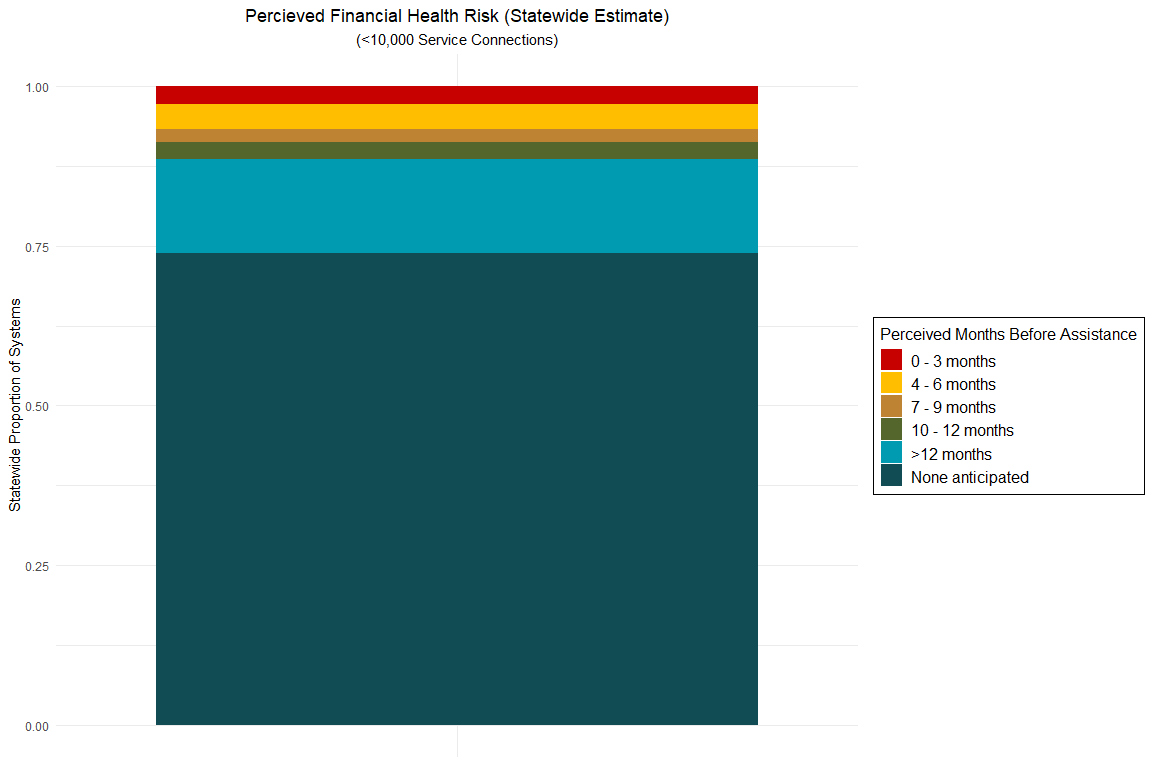


Statewide estimates for the number of water systems with less than 10,000 service connections experiencing *perceived* financial risk for the above discrete categories is plotted below with 95% confidence levels.

tab\_risk <- svytotal(~months\_before\_assist.imputed,   
 design = smalls.dsgn,  
 na.rm = TRUE) %>%   
 data.frame()  
#reassign names  
tab\_risk <- set\_rownames(tab\_risk,  
 c("0-3",   
 "4-6",  
 "7-9",   
 "10-12",  
 "> 12",  
 "None anticipated"))  
  
  
#convert rownames to column  
tab\_risk %<>%   
 rownames\_to\_column("Risk")  
  
#add lower and upper columns  
tab\_risk\_smalls <- mutate(tab\_risk,  
 lower = total - 1.96 \* SE, #95% CI  
 upper = total + 1.96 \* SE) %>%   
 mutate\_if(is.character, as.factor) %>%   
 mutate(Risk = fct\_relevel(Risk,  
 "0-3",  
 "4-6",  
 "7-9",  
 "10-12",  
 "> 12",  
 "None anticipated"))  
  
#Construct a barplot  
require(calecopal)  
tab\_risk\_smalls %>%   
ggplot(aes(x = Risk, y = total,   
 ymin = lower, ymax = upper, fill = Risk)) +  
 geom\_col() +   
 geom\_errorbar(width = 0.7) +  
 geom\_text(aes(label = paste0(paste0(round(total)," \u00B1 "),round(SE))),  
 vjust = -2) +  
 labs(title = "Perceived Financial Health Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)",  
 caption = "Strata defined by service connections. Post-stratified by fee code") +  
 xlab("Perceived Months Before Assistance Needed") +  
 ylab("Estimated Number of Systems") +  
 scale\_y\_continuous(limits = c(0,2000))+  
 scale\_fill\_manual(values = cal\_palette("kelp1"), #reverse color order  
 name = "Strata") +  
 theme\_minimal() +  
 theme(legend.position = "none",  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5),  
 plot.caption = element\_text(size = 8))

 These data are plotted again below as proportions of total systems statewide.

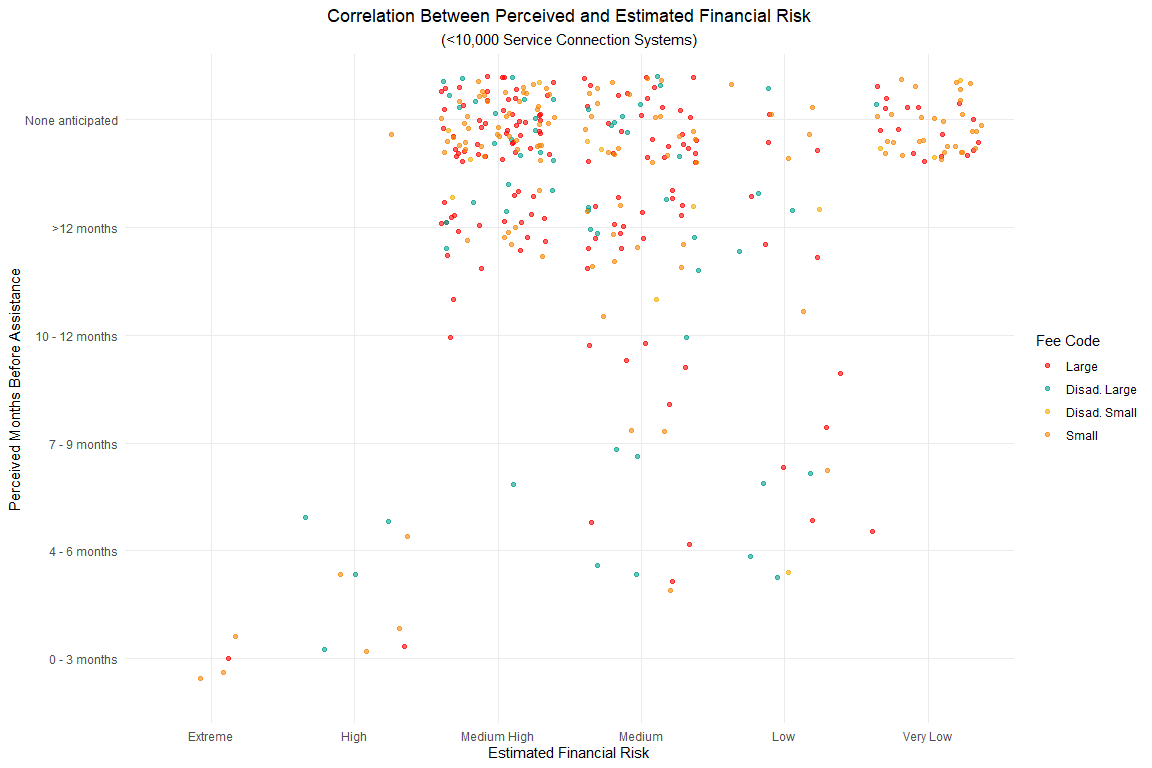
# construct and display a frequency table  
tab\_need\_cond <- svytable(~months\_before\_assist.imputed,  
 design = ps.dsgn) %>%  
 # Add conditional proportions to table  
 as.data.frame() %>%  
 mutate(n = sum(Freq), Prop\_Need = Freq/n) %>%   
 mutate(year = "2020")  
  
# Create a segmented bar graph of the conditional proportions in table  
tab\_need\_cond %>%   
 ggplot(aes(x = year, y = Prop\_Need, fill = months\_before\_assist.imputed)) +   
 geom\_col() +   
 scale\_fill\_manual(values = cal\_palette("kelp1"),  
 name = "Perceived Months Before Assistance",   
 labels = c("0 - 3 months",   
 "4 - 6 months",  
 "7 - 9 months",   
 "10 - 12 months",  
 ">12 months",  
 "None anticipated")) +  
 ylab("Statewide Proportion of Systems") +  
 labs(title = "Percieved Financial Health Risk (Statewide Estimate)",  
 subtitle = "(<10,000 Service Connections)") +  
 theme\_minimal() +  
 theme(axis.title.x = element\_blank(),  
 axis.text.x = element\_blank(),  
 legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 12),  
 legend.text = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))



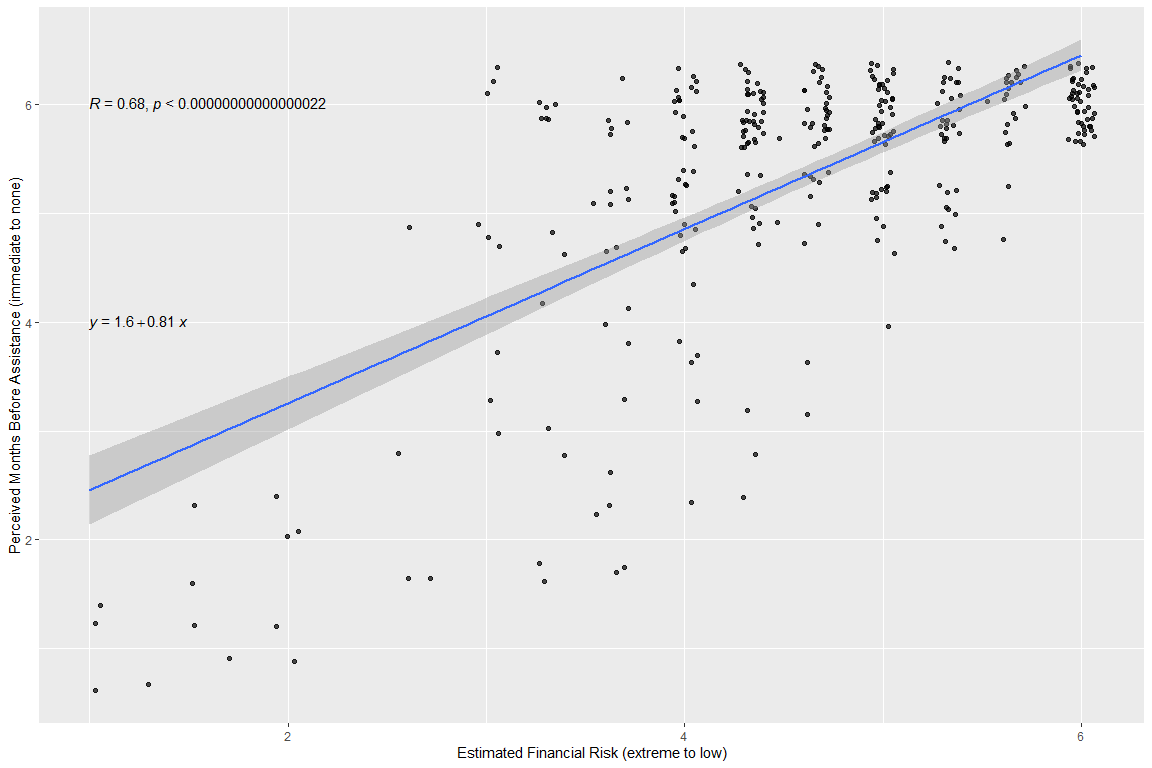
#### Correlation: *Perceived* vs. Estimated Risk

To understand the relationship between the *perceived* (i.e. directly reported, qualitative) and estimated (i.e. inferred from quantitative reported data) financial risk of surveyed water systems, a simple scatterplot between these two variables are plotted below.

allSmalls.complete %>%   
 drop\_na(Risk\_label) %>%   
 drop\_na(months\_before\_assist.imputed) %>%   
 ggplot(aes(x =Risk\_label, y = months\_before\_assist.imputed, weights= final.weight, color = Fee\_Code)) +  
 geom\_jitter(alpha = 0.6) +  
 scale\_color\_manual(values = wes\_palette("Darjeeling1"),  
 name = "Fee Code",  
 labels = c("Large",  
 "Disad. Large",  
 "Disad. Small",  
 "Small",  
 "Total"))+  
 scale\_y\_discrete(name = "Perceived Months Before Assistance",   
 labels = c("0 - 3 months",   
 "4 - 6 months",  
 "7 - 9 months",   
 "10 - 12 months",  
 ">12 months",  
 "None anticipated"))+  
 scale\_x\_discrete(name = "Estimated Financial Risk",   
 labels = c("Extreme", "High", "Medium High","Medium", "Low", "Very Low")) +  
 labs(title = "Correlation Between Perceived and Estimated Financial Risk",  
 subtitle = "(<10,000 Service Connection Systems)") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5))

 Note in the scatterplot above that points are ‘jittered’ to allow visualization. A simple linear regression between these two parameters may be calculated by re-coding the discrete ordinal variables as numeric (i.e. extreme = 6, very low = 1, etc.)

## Recode ordinal variables to numeric  
allSmalls.complete %<>%   
 mutate(months\_before\_assist\_num.imputed = case\_when(months\_before\_assist.imputed == "A" ~ 1,  
 months\_before\_assist == "B" ~ 2,  
 months\_before\_assist == "C" ~ 3,  
 months\_before\_assist == "D" ~ 4,  
 months\_before\_assist == "E" ~ 5,  
 months\_before\_assist == "F" ~ 6)) %>%   
 mutate(risk\_score\_avg\_rev = 6 - risk\_score\_avg)  
  
allSmalls.complete %>%   
 drop\_na(Risk\_label) %>%   
 drop\_na(months\_before\_assist.imputed) %>%   
 ggplot(aes(x = risk\_score\_avg\_rev, y = months\_before\_assist\_num.imputed, weights= final.weight)) +  
 geom\_jitter(alpha = 0.7) +  
 geom\_smooth(method = "lm", aes(weight = final.weight)) +  
 stat\_cor(label.y = 6)+ #this means at 35th unit in the y axis, the r squared and p value will be shown  
 stat\_regline\_equation(label.y = 4) + #this means at 30th unit regresion line equation will be shown  
 scale\_y\_continuous(name = "Perceived Months Before Assistance (immediate to none)")+  
 scale\_x\_continuous(name = "Estimated Financial Risk (extreme to low)")

 While there is a considerable amount of noise, an extremely significant (*p*<0.0001) correlation between reported finanical risk and inferred financial risk from other quantitatively-reported variables is apparent from the above figure. ## Large Systems (>10,000 Service Connections)

Statewide estimates for responses from surveyed large water systems (>10,000 service connections) are calcualted and plotted below.

Below is a table listing the total amount of debt ($USD; including other expenses such as power) reported by water systems. Note that data were imputed for this variable for 4 out of 128 respondents (3%).

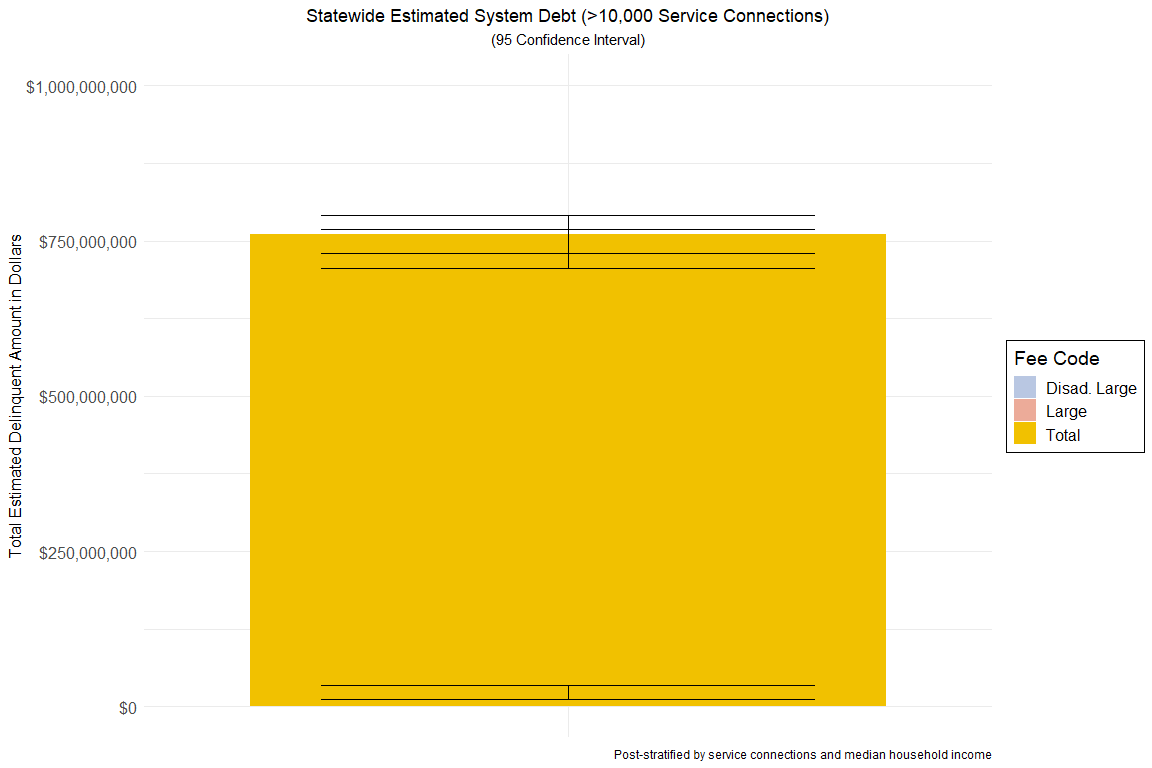
#larges by fee code  
dollars\_del\_acc\_TotR <- as.data.frame(svyby(~dollars\_del\_acc\_TotR.imputed,   
 by = ~Fee\_Code,   
 design = ps.dsgn.larges,  
 FUN = svytotal,  
 na.rm = TRUE))  
#totals  
dollars\_del\_acc\_TotR\_total <- as.data.frame(svytotal(~dollars\_del\_acc\_TotR.imputed,   
 design = ps.dsgn.larges,  
 FUN = svytotal,  
 na.rm = TRUE)) %>%   
 mutate(Fee\_Code = "Total") %>%   
 rename(se = dollars\_del\_acc\_TotR.imputed) %>%   
 rename(dollars\_del\_acc\_TotR.imputed = total)   
   
#bind totals and fee codes  
dollars\_del\_acc\_TotR\_col <- rbind(dollars\_del\_acc\_TotR, dollars\_del\_acc\_TotR\_total) %>%   
 mutate(lower = dollars\_del\_acc\_TotR.imputed - 1.96\*se, #95% CI  
 upper = dollars\_del\_acc\_TotR.imputed + 1.96\*se) %>% #95% CI  
 mutate(year = "2020") #dummy variable for plotting  
  
#print table  
kable(dollars\_del\_acc\_TotR\_col,  
 caption = "Estimated total amount of debt ($USD; including other expenses such as power)")

Estimated total amount of debt ($USD; including other expenses such as power)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fee\_Code | dollars\_del\_acc\_TotR.imputed | se | lower | upper | year |
| C1 | 737830395 | 16087056 | 706299766 | 769361024 | 2020 |
| DAVCL | 22985187 | 5981942 | 11260581 | 34709793 | 2020 |
| Total | 760815582 | 15552337 | 730333001 | 791298163 | 2020 |

Data in the above table are plotted graphically below as a stacked barplot with uncertainty annotated (95% confidence intervals).

#Construct a barplot  
dollars\_del\_acc\_TotR\_col %>%   
 mutate(Fee\_Code = fct\_reorder(Fee\_Code, dollars\_del\_acc\_TotR.imputed)) %>% #reoder so davcl is on bottom  
ggplot(aes(x = year, y = dollars\_del\_acc\_TotR.imputed,   
 ymin = lower, ymax = upper, fill = Fee\_Code)) +  
 geom\_col(position = "identity") +  
 scale\_fill\_manual(values = cal\_palette("superbloom1"),  
 name = "Fee Code",  
 labels = c("Disad. Large",   
 "Large",  
 "Total")) +  
 # scale\_x\_discrete(name = "Fee Code",   
 # labels = c("Large",   
 # "Disad. Large")) +  
 scale\_y\_continuous(labels = scales::dollar\_format(),  
 limits = c(0,1000000000)) +  
 labs(y = "Total Estimated Delinquent Amount in Dollars") +  
 geom\_errorbar(width = 0.7) +  
 labs(title = "Statewide Estimated System Debt (>10,000 Service Connections)",  
 subtitle = "(95 Confidence Interval)",  
 caption = "Post-stratified by service connections and median household income") +  
 theme\_minimal() +  
 theme(legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5),  
 axis.text.x = element\_blank(),  
 axis.title.x = element\_blank())



Below is a table listing the drinking water account-related ebt ($USD) reported aby water systems (including other expenses such as power).Note that data were imputed for this variable for 49 out of 128 respondents (38%).

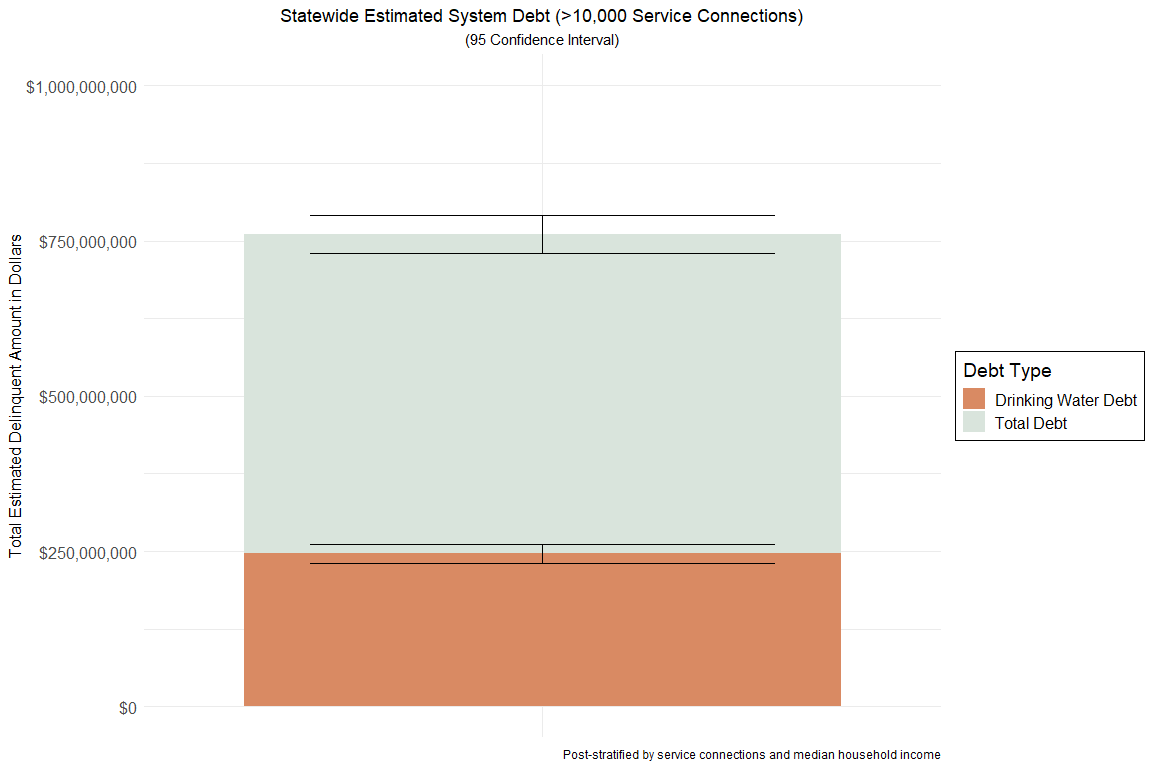
#larges by fee code  
dollars\_del\_dw\_TotR.imputed <- as.data.frame(svyby(~dollars\_del\_dw\_TotR.imputed,   
 by = ~Fee\_Code,   
 design = ps.dsgn.larges,  
 FUN = svytotal,  
 na.rm = TRUE))  
#totals  
dollars\_del\_dw\_TotR.imputed\_total <- as.data.frame(svytotal(~ dollars\_del\_dw\_TotR.imputed,   
 design = ps.dsgn.larges,  
 FUN = svytotal,  
 na.rm = TRUE)) %>%   
 mutate(Fee\_Code = "Total") %>%   
 rename(se = dollars\_del\_dw\_TotR.imputed) %>%   
 rename(dollars\_del\_dw\_TotR.imputed = total)   
   
#bind totals and fee codes  
 dollars\_del\_dw\_TotR\_col <- rbind(dollars\_del\_dw\_TotR.imputed, dollars\_del\_dw\_TotR.imputed\_total) %>%   
 mutate(lower = dollars\_del\_dw\_TotR.imputed - 1.96\*se, #95% CI  
 upper = dollars\_del\_dw\_TotR.imputed + 1.96\*se) %>% #95% CI  
 mutate(year = "2020") #dummy variable for plotting  
  
#print table  
kable(dollars\_del\_dw\_TotR\_col,  
 caption = "Estimated Total amount of debt ($USD; drinking water)")

Estimated Total amount of debt ($USD; drinking water)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fee\_Code | dollars\_del\_dw\_TotR.imputed | se | lower | upper | year |
| C1 | 232412728 | 8038079 | 216658094 | 248167363 | 2020 |
| DAVCL | 13958914 | 3441281 | 7214002 | 20703825 | 2020 |
| Total | 246371642 | 7890087 | 230907072 | 261836212 | 2020 |

An additional plot showing both drinking water expenses and total expenses is below.

### join drinking water with total debt tables ###  
#create dummy variables  
dollars\_del\_acc\_TotR\_col %<>%   
 mutate(debt = "total") %>%   
 rename(total = dollars\_del\_acc\_TotR.imputed)  
  
dollars\_del\_dw\_TotR\_col %<>%   
 mutate(debt = "drinking water") %>%   
 rename(total = dollars\_del\_dw\_TotR.imputed)  
  
#join tables  
dollars\_del\_TotR\_col <- rbind(dollars\_del\_acc\_TotR\_col,dollars\_del\_dw\_TotR\_col)  
  
#Construct a barplot  
dollars\_del\_TotR\_col %>%   
 filter(Fee\_Code == "Total") %>%   
ggplot(aes(x = year, y = total,   
 ymin = lower, ymax = upper, fill = debt)) +  
 geom\_col(position = "identity") +  
 scale\_fill\_manual(values = cal\_palette("chaparral2"),  
 name = "Debt Type",  
 labels = c("Drinking Water Debt",  
 "Total Debt")) +  
 scale\_y\_continuous(labels = scales::dollar\_format(),  
 limits = c(0,1000000000)) +  
 labs(y = "Total Estimated Delinquent Amount in Dollars") +  
 geom\_errorbar(width = 0.7) +  
 labs(title = "Statewide Estimated System Debt (>10,000 Service Connections)",  
 subtitle = "(95 Confidence Interval)",  
 caption = "Post-stratified by service connections and median household income") +  
 theme\_minimal() +  
 theme(legend.box.background = element\_rect(color = "black"),  
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(size = 12),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 12),  
 plot.title = element\_text(hjust = 0.5),  
 plot.subtitle = element\_text(hjust = 0.5),  
 axis.title.x = element\_blank(),  
 axis.text.x = element\_blank())



# References

Breiman L. (2001). Random forests. Machine Learning 45:5–32.

Hartley H. O., Rao J. N. K. (1962). Sampling with unequal probabilities and with replacement. Annals of Mathematical Statistics 33(2):350–374.

Katitas, Aycan. 2019.“Getting Started with Multiple Imputation in R.” University of Virginia Library. Obtained from <https://data.library.virginia.edu/getting-started-with-multiple-imputation-in-r/>)

Kropko, Jonathan, Ben Goodrich, Andrew Gelman, and Jennifer Hill. 2014. “Multiple imputation for continuous and categorical data: Comparing joint multivariate normal and conditional approaches.” Political Analysis 22, no. 4.

Rubin, Donald B. 1976. “Inference and missing data.” Biometrika 63, no. 3: 581-592.

Valliant, Richard, Jill A. Dever, and Frauke Kreuter. 2018. Practical Tools for Designing and Weighting Survey Samples. Statistics for Social and Behavioral Sciences. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-93632-1>.

Zhao, Peng et al. “Propensity score and proximity matching using random forest.” Contemporary clinical trials vol. 47 (2016): 85-92. <doi:10.1016/j.cct.2015.12.012>