Membership History - Bang Personal Training (Jan 2018 - Oct 2020)

Michael Hoang

01/01/2021

# **INTRODUCTION**

Since being hired in 2017, I’ve seen a numerous members walk-in through the doors of Bang Personal Training (formerly Bang Fitness). Being in a fitness studio setting, it isn’t too surprising to see clients come and go. However, considering that we are located in Downtown Toronto where there is a surplus of both boutique fitness studio and big box gyms readily available, it is important to create/know our niche in this market space.

While working at Bang, there is very little to no impact that we can provide in terms of the training service working in the membership service team. However, the same cannot be said as it pertains to non-training customer experience. Time and again, customer experience has been shown to be an important determinant factor in impacting business performance across multiple industries & across various demographics as noted [here](https://www.emerald.com/insight/content/doi/10.1108/IJQSS-01-2015-0008/full/html) and [here](https://www.pwc.com/us/en/advisory-services/publications/consumer-intelligence-series/pwc-consumer-intelligence-series-customer-experience.pdf#page=8). So, with our mission statement that “Big Box Gyms Suck”, which has routinely been marked with notions of [“less than satisfactory” customer experience](https://www.lbbonline.com/news/research-reveals-gyms-need-to-workout-their-customer-experience), this is one area where boutique fitness studios (such as ourselves) can gain a footing in the market space.

While boutique fitness studios tend to [excel over the big box gyms in this area](https://www.glofox.com/blog/defining-the-member-experience/) which is reflected in higher retention rates, it still doesn’t absolve us from membership churn as this is something that is bound to happen. Added to the fact that the 2020 Coronavirus pandemic impacting every industry, most notably the leisure-based industries [see here](https://www.spglobal.com/marketintelligence/en/news-insights/blog/industries-most-and-least-impacted-by-covid-19-from-a-probability-of-default-perspective-march-2020-update), there hasn’t been a more important time to get some introspection to see what needs to be done to come out of this situation as unscathed as possible. So, with the time spent over the initial Ontario lockdown in March 2020 learning Python + SQL + R (as there was only so much Netflix and YouTube content), I wanted to take a shot at examining the history of all members (past + present) to gleam some insight as it pertains to our demographics, membership type, attendance, types of customer-interactions, membership churn and their inter-relationship.

# **OBJECTIVE**

Perform an exploratory analysis examining the demographics and membership behavior of past and current Bang Personal Training members. The intention is to observe these factors in relation to membership retention as measured by length of membership before churn as well as measured by retention status at 3 major time points (3 months, 6 months and 12 months). Ultimately, through looking at membership churn over the last two years, I would get an idea of which factors play a significant role in membership retention so as to revamp the on-boarding and membership service operating procedure going forward.

# **METHOD**

## **Data**

For this analysis, I will be using a data set that I had created through the data that was *painfully* collected through the scheduling/payment software Wellness Living, entries from our CRM software Air Table, memory recall based on one-on-one interaction and the email history of all of the past/current members that existed between the period of January 1st, 2018 - October 5th, 2020. However, it is important to note that that this data set had looked at the entire history of our members since their initial start date which could be as early as January 2012. The following information that was collected for this dataset includes:

**Table1. Variables within the data set**

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| name | ~~Name of member~~ (redacted for privacy purposes) |
| id | Identifier of the member |
| age\_group | Age grouping for members |
| employment\_sector | Employment sector for member |
| start\_date | First ever day as a member of Bang Personal Training |
| end\_date | Last ever day as a member of Bang Personal Training ^\*^ |
| length | Total length of days as a member at Bang Personal Training |
| membership | Predominant membership type as a member at Bang Personal Training |
| reason\_to\_leave | Reason to leave Bang Personal Training |
| churn\_type | Type of membership churn for former Bang Personal Training members |
| lifetime\_revenue | Lifetime revenue of a Bang Personal Training member |
| num\_membership\_change | Number of membership changes as a member of Bang Personal Training |
| retention\_3m/6m/12m | Retention status at 3 months, 6 months and 12 months |
| current | Membership status (as of 10-05-2020) |
| active/former 1x/2x/3x/4x/unlimited/group/distance | Number of months at a given membership type as an active/former member |
| active/former 1x/2x/3x/4x/unlimited/group/distance rate | Weighted monthly rate at a given membership as an active/former member |
| num\_active\_breaks / num\_former\_breaks | Number of payment cycle breaks as an active/former member |
| num\_active\_reups / num\_former\_reups | Number of membership renewals as an active/former member (per 66 days) ^\*\*^ |
| num\_ticket\_billing | Number of email-interactions pertaining to billing issues |
| num\_ticket\_cx | Number of email-interactions not pertaining to billing/service/scheduling |
| num\_ticket\_scheduling | Number of email-interactions pertaining to scheduling requests/changes ^\*\*\*^ |
| num\_ticket\_service | Number of email-interactions pertaining to service-related requests \*\*\*\* |
| total\_sessions | Total possible number of sessions that could potentially be attended |
| attended | Number of attended sessions |
| cancelled | Number of canceled sessions |
| lost | Number of sessions lost |
| pending | Number of sessions with pending status (50:50) |
| new\_month | Was this the month that member started or returned |
| current\_month | Was this the month that member was a member at Bang Personal Training |
| leave\_month | Was this the month that member left Bang Personal Training |

* For current members, this “end” will be listed at Oct 5th, 2020

\*\* reupping the membership was based on 66 days as a baseline given its significance with behavior and habit adoption as noted by Lally P, van Jaarsveld CHM, Wardle J (2010). How are habits formed: modelling habit formation in the real world. Euro J Soc Psychol, 40: 998-1009; As long as member had at least 1 month of actual payments, will puy down “1” as an entry but will use analysis to clean this up.

\*\*\* This does not include clincal based services like chiropractic, acupuncture, RMT or physiotherapy

\*\*\*\* This includes things like receipts, sending items, putting membership holds and whatever else.

## **Tools Used**

The data set was compiled into a CSV file using Excel and all statistical analyses were conducted using R and R Studio.

RNGkind(sample.kind = "Rounding")

## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler  
## used

setwd("~/MyDatasets") # Used to set my working directory   
set.seed(123) # Needed for the sake of replicability   
  
library(readr) # To read a rectangular data set in a fast and friendly way.

## Warning: package 'readr' was built under R version 4.0.3

library(tidyverse) # A collection of packages that allow for ease in data analysis

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v dplyr 1.0.2  
## v tibble 3.0.4 v stringr 1.4.0  
## v tidyr 1.1.2 v forcats 0.5.0  
## v purrr 0.3.4

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## Warning: package 'dplyr' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr) # A means to provide grammar for processes used in data manipulation   
library(tidyr) # A set of functions to help with tidying up the data  
library(epiR) # A set of functions for demographic and empidemiologyical analysis

## Loading required package: survival

## Warning: package 'survival' was built under R version 4.0.3

## Package epiR 2.0.17 is loaded

## Type help(epi.about) for summary information

## Type browseVignettes(package = 'epiR') to learn how to use epiR for applied epidemiological analyses

##

library(ggplot2) # A system to allow for the creation of graphics based on the Grammar of Graphics   
library(stringr) # A set of functions to make it easier to work with strings  
library(purrr) # A means for enhancing R's functional programming toolkit to allow for work on functions and vectors (i.e. replace the needs for loops)  
library(tibble) # Essentially a new take on data frames that makes things easier to use when making data frames  
library(carat) # Provides functions and command-line user interface to generate allocation sequence by covariate-adaptive randomization for clincal trials as well as ability to evaluate + compare the performance of randomization procedures and test based on various criteria

## Warning: package 'carat' was built under R version 4.0.3

library(rmarkdown) # Turns my analysis into a document for reporting and presenting my work

## Warning: package 'rmarkdown' was built under R version 4.0.3

library(plyr) # Makes it simple to split data apart, do stuff to it and mash it back together.

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

library(magrittr) # A set of operators that improves code by operating as a pipe operator

## Warning: package 'magrittr' was built under R version 4.0.3

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

library(aod) # A set of functions to analyze overdispersed counts or proportions to serve as complements to more sophisticated methods of modelling such as generalized estimating equations or generalized linear mixed effect models

##   
## Attaching package: 'aod'

## The following object is masked from 'package:survival':  
##   
## rats

library(caret) # A set of functions to attempt to streamline the process for creating predictive models

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:survival':  
##   
## cluster

## The following object is masked from 'package:purrr':  
##   
## lift

library(lubridate) # A set of functions to work with date-times and time-spans

## Warning: package 'lubridate' was built under R version 4.0.3

##   
## Attaching package: 'lubridate'

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

library(ggpubr) # A means to provide some easy-to-use functions for creating and customizing ggplot2-based publication ready plots.

## Warning: package 'ggpubr' was built under R version 4.0.3

##   
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':  
##   
## mutate

library(rstatix) # A simple and intuitive pipe-friendly framework for performing basic statistical tests like t-test, Wilcoxon Test, ANOVA, Kruskal-Wallis and Correlation Analyses.

## Warning: package 'rstatix' was built under R version 4.0.3

##   
## Attaching package: 'rstatix'

## The following objects are masked from 'package:plyr':  
##   
## desc, mutate

## The following object is masked from 'package:stats':  
##   
## filter

library(mgcv) # A package used to create generalized additive mixed models

## Warning: package 'mgcv' was built under R version 4.0.3

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:carat':  
##   
## getData

## The following object is masked from 'package:dplyr':  
##   
## collapse

## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.

##   
## Attaching package: 'mgcv'

## The following object is masked from 'package:aod':  
##   
## negbin

library(FSA) # Need it to perform post-hoc analysis for non-parametric distributions (i.e. Dunn Test)

## Warning: package 'FSA' was built under R version 4.0.3

## ## FSA v0.8.31. See citation('FSA') if used in publication.  
## ## Run fishR() for related website and fishR('IFAR') for related book.

library(caret) # Contains functions to streamline the model training process for complex regression and classification problems  
library(survival) # Contains the core survival analysis routines, including definition of Surv objects, Kaplan-Meier and Aalen-Johansen (multi-state) curves, Cox models, and parametric accelerated failure time models.  
library(survminer) # Contains the function 'ggsurvplot()' for drawing easily beautiful and 'ready-to-publish' survival curves with the 'number at risk' table and 'censoring count plot'. Other functions are also available to plot adjusted curves for ‘Cox' model and to visually examine ’Cox' model assumptions.

## Warning: package 'survminer' was built under R version 4.0.3

library(mgcv) # Generalized additive (mixed) models, some of their extensions and other generalized ridge regression with multiple smoothing parameter estimation by (Restricted) Marginal Likelihood, Generalized Cross Validation and similar, or using iterated nested Laplace approximation for fully Bayesian inference  
library(car) # Need it to be able to use Variance Inflation Factor function

## Warning: package 'car' was built under R version 4.0.3

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:FSA':  
##   
## bootCase

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

library(rms) # Regression modeling, testing, estimation, validation, graphics, prediction, and typesetting by storing enhanced model design attributes in the fit

## Warning: package 'rms' was built under R version 4.0.3

## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 4.0.3

## Loading required package: Formula

## Warning: package 'Formula' was built under R version 4.0.3

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:plyr':  
##   
## is.discrete, summarize

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## Loading required package: SparseM

##   
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':  
##   
## backsolve

##   
## Attaching package: 'rms'

## The following objects are masked from 'package:car':  
##   
## Predict, vif

library(cowplot) # provides various features that help with creating publication-quality figures, such as a set of themes, functions to align plots and arrange them into complex compound figures, and functions that make it easy to annotate plots and or mix plots with images

## Warning: package 'cowplot' was built under R version 4.0.3

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggpubr':  
##   
## get\_legend

## The following object is masked from 'package:lubridate':  
##   
## stamp

library(randomForest) # To make Random Forrest Plots

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

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

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(randomForestSRC) # Fast OpenMP parallel computing of Breiman's random forests for survival, competing risks, regression and classification based on Ishwaran and Kogalur's popular random survival forests (RSF) package

## Warning: package 'randomForestSRC' was built under R version 4.0.3

##   
## randomForestSRC 2.9.3   
##   
## Type rfsrc.news() to see new features, changes, and bug fixes.   
##

##   
## Attaching package: 'randomForestSRC'

## The following object is masked from 'package:Hmisc':  
##   
## impute

## The following object is masked from 'package:purrr':  
##   
## partial

library(pec) # Validation of risk predictions obtained from survival models and competing risk models based on censored data using inverse weighting and cross-validation

## Warning: package 'pec' was built under R version 4.0.3

## Loading required package: prodlim

##   
## Attaching package: 'pec'

## The following object is masked from 'package:caret':  
##   
## R2

library(ggRandomForests) # Graphic elements for exploring Random Forests using the 'randomForest' or 'randomForestSRC' package for survival, regression and classification forests and 'ggplot2' package plotting

## Warning: package 'ggRandomForests' was built under R version 4.0.3

##   
## Attaching package: 'ggRandomForests'

## The following object is masked from 'package:randomForestSRC':  
##   
## partial.rfsrc

bang = read.csv("CSM\_Bang\_no\_names.csv", na = c(""))

## **Data cleaning**

Upon loading this data set into R, the first process was to properly format every variable to its correct data type. Each variable was formatted to the following type:

**Numeric**: id, length, lifetime revenue, number of membership change, active/former 1x/2x/3x/4x/unlimited/group/distance, active/former 1x/2x/3x/4x/unlimited/group/distance rate, number of breaks for active/former members, number of re-ups for active/former members, number of tickets for service/billing/scheduling/customer experience, total sessions, attended session, canceled sessions, lost sessions and pending sessions.

**Categorical**: age groups, employment sectors, reason to leave, churn type, retention at 3m/6m/12m, current, new month, current month and leave month

### AGE GROUPS

Age was determined based on birthday provided by member at time of signing up at Bang. It was then classified within 5 groups to somewhat match the generation division. These include (a) “Under 18”, (b) “18-29”, (c) “30-44”, (d) “45-64”, (e) “65+”. For those with unknown age, they are listed as “NaN”.

bang$age\_group = ifelse(bang$age\_group == 6, 5, bang$age\_group)  
bang$age\_group = as.factor(bang$age\_group)   
bang$age\_group = revalue(bang$age\_group, c('1' = 'Under 18', '2' = '18-29', '3' = '30-44', '4' = '45-64', '5' = '65+'))

### EMPLOYMENT SECTORS

Employment sectors was determined based on Google / LinkedIn search of the member (definitely not creepy at all). Based on area of work, member’s were classified into one of the following options that best described their employment: (a) finance/insurance, (b) scientific/academic/educational, (c) technology/information, (d) social services/non-profits, (e) government/legal, (f) advertisement/media/art/culture, (g) real estate/construction/waste, (h) natural resources/energy, (i) manufacturing/trade, (j) transportation, (k) health care/health services, (l) professional/technical services, (m) hospitality/retail/accommodation, (n) student, (o) entrepreneur/owns business and (p) other. For unknown entries, listed as “NaN”.

bang$employment\_sector = as.factor(bang$employment\_sector)  
bang$employment\_sector = revalue(  
 bang$employment\_sector,   
 c(  
 '1' = 'Finance/Insurance',  
 '2' = 'Scientific/Academic/Educational',  
 '3' = 'Technology/Information',  
 '4' = 'Social Services/Non-Profits',  
 '5' = 'Government/Legal',  
 '6' = 'Advertising/Media/Art/Culture',  
 '7' = 'Real Estate/Construction/Waste',  
 '8' = 'Natural Resource/Energy',  
 '9' = 'Manufacturing/Trade',  
 '10' = 'Transportation',  
 '11' = 'Health Care/Services',  
 '12' = 'Professional/Technical Services',  
 '13' = 'Hospitality/Retail/Accomodation',  
 '14' = 'Student',  
 '15' = 'Entrepreneural/Owns Business',  
 '16' = 'Other'  
 )  
)

### MEMBERSHIP

Seeing as some members had either increased or decreased their frequency, this can be a bit confusing. So in defining the membership of the member, it will be based on which membership the member had been frequently billed out as.

bang$membership = as.factor(bang$membership)  
bang$membership = revalue(bang$membership,   
 c(  
 "1" = "1x",  
 '2' = '2x',  
 '3' = '3x',  
 '4' = '4x',  
 '5' = 'unlimited',  
 '6' = 'group',  
 '7' = 'distance')   
 )

### ACTIVE / FORMER MONTH & RATES

In determining the bulk of the lifetime revenue of the member, their weighted monthly averages were determined from the total number of months at a particular membership type. This was further divided between those that were currently active and those that are not. This should be reflected as a numeric variable.

# Active and Former   
bang$active\_1x = as.numeric(bang$active\_1x)

## Warning: NAs introduced by coercion

bang$active\_rate\_1x = as.numeric(bang$active\_rate\_1x)

## Warning: NAs introduced by coercion

bang$active\_2x = as.numeric(bang$active\_2x)

## Warning: NAs introduced by coercion

bang$active\_rate\_2x = as.numeric(bang$active\_rate\_2x)

## Warning: NAs introduced by coercion

bang$active\_3x = as.numeric(bang$active\_3x)

## Warning: NAs introduced by coercion

bang$active\_rate\_3x = as.numeric(bang$active\_rate\_3x)

## Warning: NAs introduced by coercion

bang$active\_4x = as.numeric(bang$active\_4x)

## Warning: NAs introduced by coercion

bang$active\_rate\_4x = as.numeric(bang$active\_rate\_4x)

## Warning: NAs introduced by coercion

bang$active\_rate\_unlim = as.numeric(bang$active\_rate\_unlim)

## Warning: NAs introduced by coercion

bang$active\_unlim = as.numeric(bang$active\_unlim)

## Warning: NAs introduced by coercion

bang$active\_group = as.numeric(bang$active\_group)

## Warning: NAs introduced by coercion

bang$active\_rate\_group = as.numeric(bang$active\_rate\_group)

## Warning: NAs introduced by coercion

bang$active\_distance = as.numeric(bang$active\_distance)

## Warning: NAs introduced by coercion

bang$active\_rate\_distance = as.numeric(bang$active\_rate\_distance)

## Warning: NAs introduced by coercion

bang$former\_1x = as.numeric(bang$former\_1x)

## Warning: NAs introduced by coercion

bang$former\_rate\_1x = as.numeric(bang$former\_rate\_1x)

## Warning: NAs introduced by coercion

bang$former\_2x = as.numeric(bang$former\_2x)

## Warning: NAs introduced by coercion

bang$former\_rate\_2x = as.numeric(bang$former\_rate\_2x)

## Warning: NAs introduced by coercion

bang$former\_3x = as.numeric(bang$former\_3x)

## Warning: NAs introduced by coercion

bang$former\_rate\_3x = as.numeric(bang$former\_rate\_3x)

## Warning: NAs introduced by coercion

bang$former\_4x = as.numeric(bang$former\_4x)

## Warning: NAs introduced by coercion

bang$former\_rate\_4x = as.numeric(bang$former\_rate\_4x)

## Warning: NAs introduced by coercion

bang$former\_rate\_unlim = as.numeric(bang$former\_rate\_unlim)

## Warning: NAs introduced by coercion

bang$former\_unlim = as.numeric(bang$former\_unlim)

## Warning: NAs introduced by coercion

bang$former\_group = as.numeric(bang$former\_group)

## Warning: NAs introduced by coercion

bang$former\_rate\_group = as.numeric(bang$former\_rate\_group)

## Warning: NAs introduced by coercion

bang$former\_distance = as.numeric(bang$former\_distance)

## Warning: NAs introduced by coercion

bang$former\_rate\_distance = as.numeric(bang$former\_rate\_distance)

## Warning: NAs introduced by coercion

### REASON TO LEAVE

Based on email correspondence/exit surveys/CSM entries/memory recall, I’ve listed reasons for past members deciding to discontinue their membership at Bang based on the following categories: (a) loss of employment – unrelated to any global economic/pandemic reasonings, (b) finance/cost of membership, (c) medical/health-related reasons relating to themselves or immediate social circle, (d) moving away outside of neighbourhood area, (e) lack of accessibility or availability due to prior commitments in life, (f) pursuing other fitness interests, (g) just “ghosted” us, (h) was a time-based arrangement, (i) noted displeasure with Bang’s service or experience, (j) pandemic/global economic crisis or (k) anything other thing. For unknown reasons, it is listed as “NaN”.

bang$reason\_to\_leave = as.factor(bang$reason\_to\_leave)  
bang$reason\_to\_leave = revalue(bang$reason\_to\_leave,  
 c(  
 "1" = "loss of employment",  
 '2' = 'financial',  
 '3' = 'medical/health-related',  
 '4' = 'moving away',  
 '5' = 'lacking accessibility/availability',  
 '6' = 'other fitness interest',  
 '7' = 'ghosted us',  
 '8' = 'time-based arrangement',  
 '9' = 'noted displeasure with Bang',  
 '10' = 'pandemic/global crisis',   
 '11' = 'other'  
 )  
)

### CHURN TYPE

Using Lincoln Murphy’s description categorization of membership churn, which was originally used for software-as-a-service industry, former members were classified into one of four categories based on how they’ve left Bang through their email correspondence, recollection of exit, entries within CSM and other notes.

These categories include: (a) unexpected and unavoidable (i.e. came out of nowhere and really no way of really “saving this”), (b) unexpected and avoidable (i.e. came out of nowhere, but intervention could have been done at any earlier time to have avoided this), (c) expected and unavoidable (i.e. we knew that this was coming for some time but there was no way of preventing this), (d) expected and avoidable (i.e. we knew that this was coming, but could have been addressed earlier in some way to have prevented or at least acted upon it prior to notice)

bang$churn\_type = as.factor(bang$churn\_type)  
  
bang$churn\_type = revalue(bang$churn\_type, c('6' = '1'))  
  
bang$churn\_type = revalue(bang$churn\_type, c(  
 "1" = 'unexpected + unavoidable',  
 '2' = 'unexpected + avoidable',   
 '3' = 'expected + unavoidable',  
 '4' = 'expected + avoidable')  
)  
  
# Defining churn type   
  
bang = bang %>%   
 mutate(  
 expected\_churn = ifelse(c(churn\_type == "expected + unavoidable" | churn\_type == "expected + avoidable"), 'yes', 'no'),  
 unexpected\_churn = ifelse(c(churn\_type == 'unexpected + unavoidable' | churn\_type == "unexpected + avoidable"), 'yes', 'no'),  
 avoidable\_churn = ifelse(c(churn\_type == 'unexpected + avoidable' | churn\_type == "expected + avoidable"), 'yes', 'no'),  
 unavoidable\_churn = ifelse(c(churn\_type == 'unexpected + unavoidable' | churn\_type == "expected + unavoidable"), 'yes', 'no')  
 ) %>%  
 mutate(  
 expected\_churn = as.factor(expected\_churn),  
 unexpected\_churn = as.factor(unexpected\_churn),  
 avoidable\_churn = as.factor(avoidable\_churn),  
 unavoidable\_churn = as.factor(unavoidable\_churn)  
 ) %>%  
 mutate(  
 expected\_churn = relevel(expected\_churn, ref = 'no'),  
 unexpected\_churn = relevel(unexpected\_churn, ref = 'no'),  
 avoidable\_churn = relevel(avoidable\_churn, ref = 'no'),  
 unavoidable\_churn = relevel(unavoidable\_churn, ref = 'no')  
 )

### RETENTION 3M/6M/12M

Often used within the area of clinical addiction research as significant time points for the adoption of a behavior change, these same timelines were used here as a measure of membership retention. With a simple response of either “Yes” or “No”, it asks whether a member had continuously been a member at Bang for at least a 3-month, 6-month or 12-month stretch.

bang$retention\_3m = as.factor(bang$retention\_3m)  
bang$retention\_6m = as.factor(bang$retention\_6m)  
bang$retention\_12m = as.factor(bang$retention\_12m)  
  
bang = bang %>%   
 mutate(  
 retention\_3m = revalue(retention\_3m, c('0' = 'no', '1' = 'yes')),  
 retention\_6m = revalue(retention\_6m, c('0' = 'no', '1' = 'yes')),  
 retention\_12m = revalue(retention\_12m, c('0' = 'no', '1' = 'yes'))  
 )

### CURRENT

This is just a determinant to see if said member is currently a Hybrid Training member currently attending sessions as of October 5th, 2020.

bang$current = as.factor(bang$current)  
bang$current = revalue(bang$current,c("1" = "active", "0" = "former"))

### NEW / CURRENT / LEAVE STATUS

These variables were used in unison to determine the membership status from the period of January 1st 2018 to October 5th, 2020. For each month in this period a member was listed as being either: (a) first-ever month at Bang or first month returning as a member at Bang, (b) are they still a member at Bang during this month or (c) did they leave Bang at this month. Using the New/Current/Leave variables, which are either yes or no, there are 8 possible combinations. However, this variable will have the following classifications:

**Table2. Combinations of membership status across months/years**

|  |  |  |  |
| --- | --- | --- | --- |
| NEW | CURRENT | LEAVE | DESCRIPTION |
| No | No | No | Was not a member during this month |
| Yes | No | No | First month as a new/returning member |
| No | Yes | No | Was already a member during this month |
| No | No | Yes | Last month as a member |
| No | Yes | Yes | Last month as a member |
| Yes | No | Yes | Started and Left Bang within the same month |
| Yes | Yes | No | First month as a new/returning member |
| Yes | Yes | Yes | Started and left Bang within the same month |

From these options, the monthly status variable will classify each individual as being either (a) new/returning, (b) current member, (c) leaving or (d) started and left in the same month.

# membership\_months   
  
bang = bang %>%   
 mutate(  
 membership.01.18 =   
 ifelse(c(new\_month\_1 == 0 & current\_month\_1 == 0 & leave\_month\_1 == 0), "0",  
 ifelse(c(new\_month\_1 == 0 & current\_month\_1 == 1 & leave\_month\_1 == 0), "1",  
 ifelse(c(new\_month\_1 == 1 & current\_month\_1 == 0 & leave\_month\_1 == 0), "1",   
 ifelse(c(new\_month\_1 == 1 & current\_month\_1 == 1 & leave\_month\_1 == 0), '1', '2')))),   
   
 membership.02.18 =   
 ifelse(c(new\_month\_2 == 0 & current\_month\_2 == 0 & leave\_month\_2 == 0), "0",  
 ifelse(c(new\_month\_2 == 0 & current\_month\_2 == 1 & leave\_month\_2 == 0), "1",  
 ifelse(c(new\_month\_2 == 1 & current\_month\_2 == 0 & leave\_month\_2 == 0), "1",   
 ifelse(c(new\_month\_2 == 1 & current\_month\_2 == 1 & leave\_month\_2 == 0), '1', '2')))),   
   
 membership.03.18 =   
 ifelse(c(new\_month\_3 == 0 & current\_month\_3 == 0 & leave\_month\_3 == 0), "0",  
 ifelse(c(new\_month\_3 == 0 & current\_month\_3 == 1 & leave\_month\_3 == 0), "1",  
 ifelse(c(new\_month\_3 == 1 & current\_month\_3 == 0 & leave\_month\_3 == 0), "1",   
 ifelse(c(new\_month\_3 == 1 & current\_month\_3 == 1 & leave\_month\_3 == 0), '1', '2')))),   
   
 membership.04.18 =   
 ifelse(c(new\_month\_4 == 0 & current\_month\_4 == 0 & leave\_month\_4 == 0), "0",  
 ifelse(c(new\_month\_4 == 0 & current\_month\_4 == 1 & leave\_month\_4 == 0), "1",  
 ifelse(c(new\_month\_4 == 1 & current\_month\_4 == 0 & leave\_month\_4 == 0), "1",   
 ifelse(c(new\_month\_4 == 1 & current\_month\_4 == 1 & leave\_month\_4 == 0), '1', '2')))),  
   
 membership.05.18 =   
 ifelse(c(new\_month\_5 == 0 & current\_month\_5 == 0 & leave\_month\_5 == 0), "0",  
 ifelse(c(new\_month\_5 == 0 & current\_month\_5 == 1 & leave\_month\_5 == 0), "1",  
 ifelse(c(new\_month\_5 == 1 & current\_month\_5 == 0 & leave\_month\_5 == 0), "1",   
 ifelse(c(new\_month\_5 == 1 & current\_month\_5 == 1 & leave\_month\_5 == 0), '1', '2')))),   
   
 membership.06.18 =   
 ifelse(c(new\_month\_6 == 0 & current\_month\_6 == 0 & leave\_month\_6 == 0), "0",  
 ifelse(c(new\_month\_6 == 0 & current\_month\_6 == 1 & leave\_month\_6 == 0), "1",  
 ifelse(c(new\_month\_6 == 1 & current\_month\_6 == 0 & leave\_month\_6 == 0), "1",   
 ifelse(c(new\_month\_6 == 1 & current\_month\_6 == 1 & leave\_month\_6 == 0), '1', '2')))),   
   
 membership.07.18 =   
 ifelse(c(new\_month\_7 == 0 & current\_month\_7 == 0 & leave\_month\_7 == 0), "0",  
 ifelse(c(new\_month\_7 == 0 & current\_month\_7 == 1 & leave\_month\_7 == 0), "1",  
 ifelse(c(new\_month\_7 == 1 & current\_month\_7 == 0 & leave\_month\_7 == 0), "1",   
 ifelse(c(new\_month\_7 == 1 & current\_month\_7 == 1 & leave\_month\_7 == 0), '1', '2')))),   
   
 membership.08.18 =   
 ifelse(c(new\_month\_8 == 0 & current\_month\_8 == 0 & leave\_month\_8 == 0), "0",  
 ifelse(c(new\_month\_8 == 0 & current\_month\_8 == 1 & leave\_month\_8 == 0), "1",  
 ifelse(c(new\_month\_8 == 1 & current\_month\_8 == 0 & leave\_month\_8 == 0), "1",   
 ifelse(c(new\_month\_8 == 1 & current\_month\_8 == 1 & leave\_month\_8 == 0), '1', '2')))),  
   
 membership.09.18 =   
 ifelse(c(new\_month\_9 == 0 & current\_month\_9 == 0 & leave\_month\_9 == 0), "0",  
 ifelse(c(new\_month\_9 == 0 & current\_month\_9 == 1 & leave\_month\_9 == 0), "1",  
 ifelse(c(new\_month\_9 == 1 & current\_month\_9 == 0 & leave\_month\_9 == 0), "1",   
 ifelse(c(new\_month\_9 == 1 & current\_month\_9 == 1 & leave\_month\_9 == 0), '1', '2')))),   
   
 membership.10.18 =   
 ifelse(c(new\_month\_10 == 0 & current\_month\_10 == 0 & leave\_month\_10 == 0), "0",  
 ifelse(c(new\_month\_10 == 0 & current\_month\_10 == 1 & leave\_month\_10 == 0), "1",  
 ifelse(c(new\_month\_10 == 1 & current\_month\_10 == 0 & leave\_month\_10 == 0), "1",   
 ifelse(c(new\_month\_10 == 1 & current\_month\_10 == 1 & leave\_month\_10 == 0), '1', '2')))),   
   
 membership.11.18 =   
 ifelse(c(new\_month\_11 == 0 & current\_month\_11 == 0 & leave\_month\_11 == 0), "0",  
 ifelse(c(new\_month\_11 == 0 & current\_month\_11 == 1 & leave\_month\_11 == 0), "1",  
 ifelse(c(new\_month\_11 == 1 & current\_month\_11 == 0 & leave\_month\_11 == 0), "1",   
 ifelse(c(new\_month\_11 == 1 & current\_month\_11 == 1 & leave\_month\_11 == 0), '1', '2')))),   
   
 membership.12.18 =   
 ifelse(c(new\_month\_12 == 0 & current\_month\_12 == 0 & leave\_month\_12 == 0), "0",  
 ifelse(c(new\_month\_12 == 0 & current\_month\_12 == 1 & leave\_month\_12 == 0), "1",  
 ifelse(c(new\_month\_12 == 1 & current\_month\_12 == 0 & leave\_month\_12 == 0), "1",   
 ifelse(c(new\_month\_12 == 1 & current\_month\_12 == 1 & leave\_month\_12 == 0), '1', '2')))),  
   
 membership.01.19 =   
 ifelse(c(new\_month\_13 == 0 & current\_month\_13 == 0 & leave\_month\_13 == 0), "0",  
 ifelse(c(new\_month\_13 == 0 & current\_month\_13 == 1 & leave\_month\_13 == 0), "1",  
 ifelse(c(new\_month\_13 == 1 & current\_month\_13 == 0 & leave\_month\_13 == 0), "1",   
 ifelse(c(new\_month\_13 == 1 & current\_month\_13 == 1 & leave\_month\_13 == 0), '1', '2')))),   
   
 membership.02.19 =   
 ifelse(c(new\_month\_14 == 0 & current\_month\_14 == 0 & leave\_month\_14 == 0), "0",  
 ifelse(c(new\_month\_14 == 0 & current\_month\_14 == 1 & leave\_month\_14 == 0), "1",  
 ifelse(c(new\_month\_14 == 1 & current\_month\_14 == 0 & leave\_month\_14 == 0), "1",   
 ifelse(c(new\_month\_14 == 1 & current\_month\_14 == 1 & leave\_month\_14 == 0), '1', '2')))),   
   
 membership.03.19 =   
 ifelse(c(new\_month\_15 == 0 & current\_month\_15 == 0 & leave\_month\_15 == 0), "0",  
 ifelse(c(new\_month\_15 == 0 & current\_month\_15 == 1 & leave\_month\_15 == 0), "1",  
 ifelse(c(new\_month\_15 == 1 & current\_month\_15 == 0 & leave\_month\_15 == 0), "1",   
 ifelse(c(new\_month\_15 == 1 & current\_month\_15 == 1 & leave\_month\_15 == 0), '1', '2')))),   
   
 membership.04.19 =   
 ifelse(c(new\_month\_16 == 0 & current\_month\_16 == 0 & leave\_month\_16 == 0), "0",  
 ifelse(c(new\_month\_16 == 0 & current\_month\_16 == 1 & leave\_month\_16 == 0), "1",  
 ifelse(c(new\_month\_16 == 1 & current\_month\_16 == 0 & leave\_month\_16 == 0), "1",   
 ifelse(c(new\_month\_16 == 1 & current\_month\_16 == 1 & leave\_month\_16 == 0), '1', '2')))),   
   
 membership.05.19 =   
 ifelse(c(new\_month\_17 == 0 & current\_month\_17 == 0 & leave\_month\_17 == 0), "0",  
 ifelse(c(new\_month\_17 == 0 & current\_month\_17 == 1 & leave\_month\_17 == 0), "1",  
 ifelse(c(new\_month\_17 == 1 & current\_month\_17 == 0 & leave\_month\_17 == 0), "1",   
 ifelse(c(new\_month\_17 == 1 & current\_month\_17 == 1 & leave\_month\_17 == 0), '1', '2')))),   
   
 membership.06.19 =   
 ifelse(c(new\_month\_18 == 0 & current\_month\_18 == 0 & leave\_month\_18 == 0), "0",  
 ifelse(c(new\_month\_18 == 0 & current\_month\_18 == 1 & leave\_month\_18 == 0), "1",  
 ifelse(c(new\_month\_18 == 1 & current\_month\_18 == 0 & leave\_month\_18 == 0), "1",   
 ifelse(c(new\_month\_18 == 1 & current\_month\_18 == 1 & leave\_month\_18 == 0), '1', '2')))),  
   
 membership.07.19 =   
 ifelse(c(new\_month\_19 == 0 & current\_month\_19 == 0 & leave\_month\_19 == 0), "0",  
 ifelse(c(new\_month\_19 == 0 & current\_month\_19 == 1 & leave\_month\_19 == 0), "1",  
 ifelse(c(new\_month\_19 == 1 & current\_month\_19 == 0 & leave\_month\_19 == 0), "1",   
 ifelse(c(new\_month\_19 == 1 & current\_month\_19 == 1 & leave\_month\_19 == 0), '1', '2')))),   
   
 membership.08.19 =   
 ifelse(c(new\_month\_20 == 0 & current\_month\_20 == 0 & leave\_month\_20 == 0), "0",  
 ifelse(c(new\_month\_20 == 0 & current\_month\_20 == 1 & leave\_month\_20 == 0), "1",  
 ifelse(c(new\_month\_20 == 1 & current\_month\_20 == 0 & leave\_month\_20 == 0), "1",   
 ifelse(c(new\_month\_20 == 1 & current\_month\_20 == 1 & leave\_month\_20 == 0), '1', '2')))),   
   
 membership.09.19 =   
 ifelse(c(new\_month\_21 == 0 & current\_month\_21 == 0 & leave\_month\_21 == 0), "0",  
 ifelse(c(new\_month\_21 == 0 & current\_month\_21 == 1 & leave\_month\_21 == 0), "1",  
 ifelse(c(new\_month\_21 == 1 & current\_month\_21 == 0 & leave\_month\_21 == 0), "1",   
 ifelse(c(new\_month\_21 == 1 & current\_month\_21 == 1 & leave\_month\_21 == 0), '1', '2')))),   
   
 membership.10.19 =   
 ifelse(c(new\_month\_22 == 0 & current\_month\_22 == 0 & leave\_month\_22 == 0), "0",  
 ifelse(c(new\_month\_22 == 0 & current\_month\_22 == 1 & leave\_month\_22 == 0), "1",  
 ifelse(c(new\_month\_22 == 1 & current\_month\_22 == 0 & leave\_month\_22 == 0), "1",   
 ifelse(c(new\_month\_22 == 1 & current\_month\_22 == 1 & leave\_month\_22 == 0), '1', '2')))),  
   
 membership.11.19 =   
 ifelse(c(new\_month\_23 == 0 & current\_month\_23 == 0 & leave\_month\_23 == 0), "0",  
 ifelse(c(new\_month\_23 == 0 & current\_month\_23 == 1 & leave\_month\_23 == 0), "1",  
 ifelse(c(new\_month\_23 == 1 & current\_month\_23 == 0 & leave\_month\_23 == 0), "1",   
 ifelse(c(new\_month\_23 == 1 & current\_month\_23 == 1 & leave\_month\_23 == 0), '1', '2')))),   
   
 membership.12.19 =   
 ifelse(c(new\_month\_24 == 0 & current\_month\_24 == 0 & leave\_month\_24 == 0), "0",  
 ifelse(c(new\_month\_24 == 0 & current\_month\_24 == 1 & leave\_month\_24 == 0), "1",  
 ifelse(c(new\_month\_24 == 1 & current\_month\_24 == 0 & leave\_month\_24 == 0), "1",   
 ifelse(c(new\_month\_24 == 1 & current\_month\_24 == 1 & leave\_month\_24 == 0), '1', '2')))),   
   
 membership.01.20 =   
 ifelse(c(new\_month\_25 == 0 & current\_month\_25 == 0 & leave\_month\_25 == 0), "0",  
 ifelse(c(new\_month\_25 == 0 & current\_month\_25 == 1 & leave\_month\_25 == 0), "1",  
 ifelse(c(new\_month\_25 == 1 & current\_month\_25 == 0 & leave\_month\_25 == 0), "1",   
 ifelse(c(new\_month\_25 == 1 & current\_month\_25 == 1 & leave\_month\_25 == 0), '1', '2')))),   
   
 membership.02.20 =   
 ifelse(c(new\_month\_26 == 0 & current\_month\_26 == 0 & leave\_month\_26 == 0), "0",  
 ifelse(c(new\_month\_26 == 0 & current\_month\_26 == 1 & leave\_month\_26 == 0), "1",  
 ifelse(c(new\_month\_26 == 1 & current\_month\_26 == 0 & leave\_month\_26 == 0), "1",   
 ifelse(c(new\_month\_26 == 1 & current\_month\_26 == 1 & leave\_month\_26 == 0), '1', '2')))),  
   
 membership.03.20 =   
 ifelse(c(new\_month\_27 == 0 & current\_month\_27 == 0 & leave\_month\_27 == 0), "0",  
 ifelse(c(new\_month\_27 == 0 & current\_month\_27 == 1 & leave\_month\_27 == 0), "1",  
 ifelse(c(new\_month\_27 == 1 & current\_month\_27 == 0 & leave\_month\_27 == 0), "1",   
 ifelse(c(new\_month\_27 == 1 & current\_month\_27 == 1 & leave\_month\_27 == 0), '1', '2')))),   
   
 membership.04.20 =   
 ifelse(c(new\_month\_28 == 0 & current\_month\_28 == 0 & leave\_month\_28 == 0), "0",  
 ifelse(c(new\_month\_28 == 0 & current\_month\_28 == 1 & leave\_month\_28 == 0), "1",  
 ifelse(c(new\_month\_28 == 1 & current\_month\_28 == 0 & leave\_month\_28 == 0), "1",   
 ifelse(c(new\_month\_28 == 1 & current\_month\_28 == 1 & leave\_month\_28 == 0), '1', '2')))),  
   
 membership.05.20 =   
 ifelse(c(new\_month\_29 == 0 & current\_month\_29 == 0 & leave\_month\_29 == 0), "0",  
 ifelse(c(new\_month\_29 == 0 & current\_month\_29 == 1 & leave\_month\_29 == 0), "1",  
 ifelse(c(new\_month\_29 == 1 & current\_month\_29 == 0 & leave\_month\_29 == 0), "1",   
 ifelse(c(new\_month\_29 == 1 & current\_month\_29 == 1 & leave\_month\_29 == 0), '1', '2')))),   
   
 membership.06.20 =   
 ifelse(c(new\_month\_30 == 0 & current\_month\_30 == 0 & leave\_month\_30 == 0), "0",  
 ifelse(c(new\_month\_30 == 0 & current\_month\_30 == 1 & leave\_month\_30 == 0), "1",  
 ifelse(c(new\_month\_30 == 1 & current\_month\_30 == 0 & leave\_month\_30 == 0), "1",   
 ifelse(c(new\_month\_30 == 1 & current\_month\_30 == 1 & leave\_month\_30 == 0), '1', '2')))),   
   
 membership.07.20 =   
 ifelse(c(new\_month\_31 == 0 & current\_month\_31 == 0 & leave\_month\_31 == 0), "0",  
 ifelse(c(new\_month\_31 == 0 & current\_month\_31 == 1 & leave\_month\_31 == 0), "1",  
 ifelse(c(new\_month\_31 == 1 & current\_month\_31 == 0 & leave\_month\_31 == 0), "1",   
 ifelse(c(new\_month\_31 == 1 & current\_month\_31 == 1 & leave\_month\_31 == 0), '1', '2')))),  
   
 membership.08.20 =   
 ifelse(c(new\_month\_32 == 0 & current\_month\_32 == 0 & leave\_month\_32 == 0), "0",  
 ifelse(c(new\_month\_32 == 0 & current\_month\_32 == 1 & leave\_month\_32 == 0), "1",  
 ifelse(c(new\_month\_32 == 1 & current\_month\_32 == 0 & leave\_month\_32 == 0), "1",   
 ifelse(c(new\_month\_32 == 1 & current\_month\_32 == 1 & leave\_month\_32 == 0), '1', '2')))),   
   
 membership.09.20 =   
 ifelse(c(new\_month\_33 == 0 & current\_month\_33 == 0 & leave\_month\_33 == 0), "0",  
 ifelse(c(new\_month\_33 == 0 & current\_month\_33 == 1 & leave\_month\_33 == 0), "1",  
 ifelse(c(new\_month\_33 == 1 & current\_month\_33 == 0 & leave\_month\_33 == 0), "1",   
 ifelse(c(new\_month\_33 == 1 & current\_month\_33 == 1 & leave\_month\_33 == 0), '1', '2')))),   
   
 membership.10.20 =   
 ifelse(c(new\_month\_34 == 0 & current\_month\_34 == 0 & leave\_month\_34 == 0), "0",  
 ifelse(c(new\_month\_34 == 0 & current\_month\_34 == 1 & leave\_month\_34 == 0), "1",  
 ifelse(c(new\_month\_34 == 1 & current\_month\_34 == 0 & leave\_month\_34 == 0), "1",   
 ifelse(c(new\_month\_34 == 1 & current\_month\_34 == 1 & leave\_month\_34 == 0), '1', '2'))))  
 )  
  
bang = bang %>%   
 mutate(  
 membership.01.18 = as.factor(membership.01.18),  
 membership.03.18 = as.factor(membership.03.18),  
 membership.04.18 = as.factor(membership.04.18),  
 membership.05.18 = as.factor(membership.05.18),  
 membership.06.18 = as.factor(membership.06.18),  
 membership.07.18 = as.factor(membership.07.18),  
 membership.08.18 = as.factor(membership.08.18),  
 membership.09.18 = as.factor(membership.09.18),  
 membership.02.18 = as.factor(membership.02.18),  
 membership.10.18 = as.factor(membership.10.18),  
 membership.11.18 = as.factor(membership.11.18),  
 membership.12.18 = as.factor(membership.12.18),  
 membership.01.19 = as.factor(membership.01.19),  
 membership.03.19 = as.factor(membership.03.19),  
 membership.04.19 = as.factor(membership.04.19),  
 membership.05.19 = as.factor(membership.05.19),  
 membership.06.19 = as.factor(membership.06.19),  
 membership.07.19 = as.factor(membership.07.19),  
 membership.08.19 = as.factor(membership.08.19),  
 membership.09.19 = as.factor(membership.09.19),  
 membership.02.19 = as.factor(membership.02.19),  
 membership.10.19 = as.factor(membership.10.19),  
 membership.11.19 = as.factor(membership.11.19),  
 membership.12.19 = as.factor(membership.12.19),  
 membership.01.20 = as.factor(membership.01.20),  
 membership.03.20 = as.factor(membership.03.20),  
 membership.04.20 = as.factor(membership.04.20),  
 membership.05.20 = as.factor(membership.05.20),  
 membership.06.20 = as.factor(membership.06.20),  
 membership.07.20 = as.factor(membership.07.20),  
 membership.08.20 = as.factor(membership.08.20),  
 membership.09.20 = as.factor(membership.09.20),  
 membership.02.20 = as.factor(membership.02.20),  
 membership.10.20 = as.factor(membership.10.20)  
 )  
  
bang = bang %>%   
 mutate(  
 membership.01.18 = revalue(membership.01.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.02.18 = revalue(membership.02.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.03.18 = revalue(membership.03.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.04.18 = revalue(membership.04.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.05.18 = revalue(membership.05.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.06.18 = revalue(membership.06.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.07.18 = revalue(membership.07.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.08.18 = revalue(membership.08.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.09.18 = revalue(membership.09.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.10.18 = revalue(membership.10.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.11.18 = revalue(membership.11.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.12.18 = revalue(membership.12.18, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.01.19 = revalue(membership.01.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.02.19 = revalue(membership.02.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.03.19 = revalue(membership.03.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.04.19 = revalue(membership.04.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.05.19 = revalue(membership.05.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.06.19 = revalue(membership.06.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.07.19 = revalue(membership.07.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.08.19 = revalue(membership.08.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.09.19 = revalue(membership.09.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.10.19 = revalue(membership.10.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.11.19 = revalue(membership.11.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.12.19 = revalue(membership.12.19, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.01.20 = revalue(membership.01.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.02.20 = revalue(membership.02.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.03.20 = revalue(membership.03.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.04.20 = revalue(membership.04.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.05.20 = revalue(membership.05.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.06.20 = revalue(membership.06.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.07.20 = revalue(membership.07.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.08.20 = revalue(membership.08.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.09.20 = revalue(membership.09.20, c('0' = "no", "1" = 'yes', '2' = 'was but left')),  
 membership.10.20 = revalue(membership.10.20, c('0' = "no", "1" = 'yes', '2' = 'was but left'))  
 )

## The following `from` values were not present in `x`: 2

bang <- bang %>%   
 mutate(  
 status.01.18 = ifelse(c(new\_month\_1 == 0 & current\_month\_1 == 0 & leave\_month\_1 == 0), 0,  
 ifelse(c(new\_month\_1 == 1 & current\_month\_1 == 0 & leave\_month\_1 == 0), 1,   
 ifelse(c(new\_month\_1 == 0 & current\_month\_1 == 1 & leave\_month\_1 == 0), 2,   
 ifelse(c(new\_month\_1 == 0 & current\_month\_1 == 0 & leave\_month\_1 == 1), 4,  
 ifelse(c(new\_month\_1 == 1 & current\_month\_1 == 1 & leave\_month\_1 == 0), 1,  
 ifelse(c(new\_month\_1 == 1 & current\_month\_1 == 0 & leave\_month\_1 == 1), 3,  
 ifelse(c(new\_month\_1 == 0 & current\_month\_1 == 1 & leave\_month\_1 == 1), 4, 3))))))),   
   
 status.02.18 = ifelse(c(new\_month\_2 == 0 & current\_month\_2 == 0 & leave\_month\_2 == 0), 0,  
 ifelse(c(new\_month\_2 == 1 & current\_month\_2 == 0 & leave\_month\_2 == 0), 1,   
 ifelse(c(new\_month\_2 == 0 & current\_month\_2 == 1 & leave\_month\_2 == 0), 2,   
 ifelse(c(new\_month\_2 == 0 & current\_month\_2 == 0 & leave\_month\_2 == 1), 4,  
 ifelse(c(new\_month\_2 == 1 & current\_month\_2 == 1 & leave\_month\_2 == 0), 1,  
 ifelse(c(new\_month\_2 == 1 & current\_month\_2 == 0 & leave\_month\_2 == 1), 3,  
 ifelse(c(new\_month\_2 == 0 & current\_month\_2 == 1 & leave\_month\_2 == 1), 4, 3))))))),  
   
 status.03.18 = ifelse(c(new\_month\_3 == 0 & current\_month\_3 == 0 & leave\_month\_3 == 0), 0,  
 ifelse(c(new\_month\_3 == 1 & current\_month\_3 == 0 & leave\_month\_3 == 0), 1,   
 ifelse(c(new\_month\_3 == 0 & current\_month\_3 == 1 & leave\_month\_3 == 0), 2,   
 ifelse(c(new\_month\_3 == 0 & current\_month\_3 == 0 & leave\_month\_3 == 1), 4,  
 ifelse(c(new\_month\_3 == 1 & current\_month\_3 == 1 & leave\_month\_3 == 0), 1,  
 ifelse(c(new\_month\_3 == 1 & current\_month\_3 == 0 & leave\_month\_3 == 1), 3,  
 ifelse(c(new\_month\_3 == 0 & current\_month\_3 == 1 & leave\_month\_3 == 1), 4, 3))))))),  
   
 status.04.18 = ifelse(c(new\_month\_4 == 0 & current\_month\_4 == 0 & leave\_month\_4 == 0), 0,  
 ifelse(c(new\_month\_4 == 1 & current\_month\_4 == 0 & leave\_month\_4 == 0), 1,   
 ifelse(c(new\_month\_4 == 0 & current\_month\_4 == 1 & leave\_month\_4 == 0), 2,   
 ifelse(c(new\_month\_4 == 0 & current\_month\_4 == 0 & leave\_month\_4 == 1), 4,  
 ifelse(c(new\_month\_4 == 1 & current\_month\_4 == 1 & leave\_month\_4 == 0), 1,  
 ifelse(c(new\_month\_4 == 1 & current\_month\_4 == 0 & leave\_month\_4 == 1), 3,  
 ifelse(c(new\_month\_4 == 0 & current\_month\_4 == 1 & leave\_month\_4 == 1), 4, 3))))))),  
   
 status.05.18 = ifelse(c(new\_month\_5 == 0 & current\_month\_5 == 0 & leave\_month\_5 == 0), 0,  
 ifelse(c(new\_month\_5 == 1 & current\_month\_5 == 0 & leave\_month\_5 == 0), 1,   
 ifelse(c(new\_month\_5 == 0 & current\_month\_5 == 1 & leave\_month\_5 == 0), 2,   
 ifelse(c(new\_month\_5 == 0 & current\_month\_5 == 0 & leave\_month\_5 == 1), 4,  
 ifelse(c(new\_month\_5 == 1 & current\_month\_5 == 1 & leave\_month\_5 == 0), 1,  
 ifelse(c(new\_month\_5 == 1 & current\_month\_5 == 0 & leave\_month\_5 == 1), 3,  
 ifelse(c(new\_month\_5 == 0 & current\_month\_5 == 1 & leave\_month\_5 == 1), 4, 3))))))),  
   
 status.06.18 = ifelse(c(new\_month\_6 == 0 & current\_month\_6 == 0 & leave\_month\_6 == 0), 0,  
 ifelse(c(new\_month\_6 == 1 & current\_month\_6 == 0 & leave\_month\_6 == 0), 1,   
 ifelse(c(new\_month\_6 == 0 & current\_month\_6 == 1 & leave\_month\_6 == 0), 2,   
 ifelse(c(new\_month\_6 == 0 & current\_month\_6 == 0 & leave\_month\_6 == 1), 4,  
 ifelse(c(new\_month\_6 == 1 & current\_month\_6 == 1 & leave\_month\_6 == 0), 1,  
 ifelse(c(new\_month\_6 == 1 & current\_month\_6 == 0 & leave\_month\_6 == 1), 3,  
 ifelse(c(new\_month\_6 == 0 & current\_month\_6 == 1 & leave\_month\_6 == 1), 4, 3))))))),  
   
 status.07.18 = ifelse(c(new\_month\_7 == 0 & current\_month\_7 == 0 & leave\_month\_7 == 0), 0,  
 ifelse(c(new\_month\_7 == 1 & current\_month\_7 == 0 & leave\_month\_7 == 0), 1,   
 ifelse(c(new\_month\_7 == 0 & current\_month\_7 == 1 & leave\_month\_7 == 0), 2,   
 ifelse(c(new\_month\_7 == 0 & current\_month\_7 == 0 & leave\_month\_7 == 1), 4,  
 ifelse(c(new\_month\_7 == 1 & current\_month\_7 == 1 & leave\_month\_7 == 0), 1,  
 ifelse(c(new\_month\_7 == 1 & current\_month\_7 == 0 & leave\_month\_7 == 1), 3,  
 ifelse(c(new\_month\_7 == 0 & current\_month\_7 == 1 & leave\_month\_7 == 1), 4, 3))))))),  
   
 status.08.18 = ifelse(c(new\_month\_8 == 0 & current\_month\_8 == 0 & leave\_month\_8 == 0), 0,  
 ifelse(c(new\_month\_8 == 1 & current\_month\_8 == 0 & leave\_month\_8 == 0), 1,   
 ifelse(c(new\_month\_8 == 0 & current\_month\_8 == 1 & leave\_month\_8 == 0), 2,   
 ifelse(c(new\_month\_8 == 0 & current\_month\_8 == 0 & leave\_month\_8 == 1), 4,  
 ifelse(c(new\_month\_8 == 1 & current\_month\_8 == 1 & leave\_month\_8 == 0), 1,  
 ifelse(c(new\_month\_8 == 1 & current\_month\_8 == 0 & leave\_month\_8 == 1), 3,  
 ifelse(c(new\_month\_8 == 0 & current\_month\_8 == 1 & leave\_month\_8 == 1), 4, 3))))))),  
   
 status.09.18 = ifelse(c(new\_month\_9 == 0 & current\_month\_9 == 0 & leave\_month\_9 == 0), 0,  
 ifelse(c(new\_month\_9 == 1 & current\_month\_9 == 0 & leave\_month\_9 == 0), 1,   
 ifelse(c(new\_month\_9 == 0 & current\_month\_9 == 1 & leave\_month\_9 == 0), 2,   
 ifelse(c(new\_month\_9 == 0 & current\_month\_9 == 0 & leave\_month\_9 == 1), 4,  
 ifelse(c(new\_month\_9 == 1 & current\_month\_9 == 1 & leave\_month\_9 == 0), 1,  
 ifelse(c(new\_month\_9 == 1 & current\_month\_9 == 0 & leave\_month\_9 == 1), 3,  
 ifelse(c(new\_month\_9 == 0 & current\_month\_9 == 1 & leave\_month\_9 == 1), 4, 3))))))),  
   
 status.10.18 = ifelse(c(new\_month\_10 == 0 & current\_month\_10 == 0 & leave\_month\_10 == 0), 0,  
 ifelse(c(new\_month\_10 == 1 & current\_month\_10 == 0 & leave\_month\_10 == 0), 1,   
 ifelse(c(new\_month\_10 == 0 & current\_month\_10 == 1 & leave\_month\_10 == 0), 2,   
 ifelse(c(new\_month\_10 == 0 & current\_month\_10 == 0 & leave\_month\_10 == 1), 4,  
 ifelse(c(new\_month\_10 == 1 & current\_month\_10 == 1 & leave\_month\_10 == 0), 1,  
 ifelse(c(new\_month\_10 == 1 & current\_month\_10 == 0 & leave\_month\_10 == 1), 3,  
 ifelse(c(new\_month\_10 == 0 & current\_month\_10 == 1 & leave\_month\_10 == 1), 4, 3))))))),  
   
 status.11.18 = ifelse(c(new\_month\_11 == 0 & current\_month\_11 == 0 & leave\_month\_11 == 0), 0,  
 ifelse(c(new\_month\_11 == 1 & current\_month\_11 == 0 & leave\_month\_11 == 0), 1,   
 ifelse(c(new\_month\_11 == 0 & current\_month\_11 == 1 & leave\_month\_11 == 0), 2,   
 ifelse(c(new\_month\_11 == 0 & current\_month\_11 == 0 & leave\_month\_11 == 1), 4,  
 ifelse(c(new\_month\_11 == 1 & current\_month\_11 == 1 & leave\_month\_11 == 0), 1,  
 ifelse(c(new\_month\_11 == 1 & current\_month\_11 == 0 & leave\_month\_11 == 1), 3,  
 ifelse(c(new\_month\_11 == 0 & current\_month\_11 == 1 & leave\_month\_11 == 1), 4, 3))))))),  
   
 status.12.18 = ifelse(c(new\_month\_12 == 0 & current\_month\_12 == 0 & leave\_month\_12 == 0), 0,  
 ifelse(c(new\_month\_12 == 1 & current\_month\_12 == 0 & leave\_month\_12 == 0), 1,   
 ifelse(c(new\_month\_12 == 0 & current\_month\_12 == 1 & leave\_month\_12 == 0), 2,   
 ifelse(c(new\_month\_12 == 0 & current\_month\_12 == 0 & leave\_month\_12 == 1), 4,  
 ifelse(c(new\_month\_12 == 1 & current\_month\_12 == 1 & leave\_month\_12 == 0), 1,  
 ifelse(c(new\_month\_12 == 1 & current\_month\_12 == 0 & leave\_month\_12 == 1), 3,  
 ifelse(c(new\_month\_12 == 0 & current\_month\_12 == 1 & leave\_month\_12 == 1), 4, 3))))))),  
   
   
   
 status.01.19 = ifelse(c(new\_month\_13 == 0 & current\_month\_13 == 0 & leave\_month\_13 == 0), 0,  
 ifelse(c(new\_month\_13 == 1 & current\_month\_13 == 0 & leave\_month\_13 == 0), 1,   
 ifelse(c(new\_month\_13 == 0 & current\_month\_13 == 1 & leave\_month\_13 == 0), 2,   
 ifelse(c(new\_month\_13 == 0 & current\_month\_13 == 0 & leave\_month\_13 == 1), 4,  
 ifelse(c(new\_month\_13 == 1 & current\_month\_13 == 1 & leave\_month\_13 == 0), 1,  
 ifelse(c(new\_month\_13 == 1 & current\_month\_13 == 0 & leave\_month\_13 == 1), 3,  
 ifelse(c(new\_month\_13 == 0 & current\_month\_13 == 1 & leave\_month\_13 == 1), 4, 3))))))),  
   
 status.02.19 = ifelse(c(new\_month\_14 == 0 & current\_month\_14 == 0 & leave\_month\_14 == 0), 0,  
 ifelse(c(new\_month\_14 == 1 & current\_month\_14 == 0 & leave\_month\_14 == 0), 1,   
 ifelse(c(new\_month\_14 == 0 & current\_month\_14 == 1 & leave\_month\_14 == 0), 2,   
 ifelse(c(new\_month\_14 == 0 & current\_month\_14 == 0 & leave\_month\_14 == 1), 4,  
 ifelse(c(new\_month\_14 == 1 & current\_month\_14 == 1 & leave\_month\_14 == 0), 1,  
 ifelse(c(new\_month\_14 == 1 & current\_month\_14 == 0 & leave\_month\_14 == 1), 3,  
 ifelse(c(new\_month\_14 == 0 & current\_month\_14 == 1 & leave\_month\_14 == 1), 4, 3))))))),  
   
 status.03.19 = ifelse(c(new\_month\_15 == 0 & current\_month\_15 == 0 & leave\_month\_15 == 0), 0,  
 ifelse(c(new\_month\_15 == 1 & current\_month\_15 == 0 & leave\_month\_15 == 0), 1,   
 ifelse(c(new\_month\_15 == 0 & current\_month\_15 == 1 & leave\_month\_15 == 0), 2,   
 ifelse(c(new\_month\_15 == 0 & current\_month\_15 == 0 & leave\_month\_15 == 1), 4,  
 ifelse(c(new\_month\_15 == 1 & current\_month\_15 == 1 & leave\_month\_15 == 0), 1,  
 ifelse(c(new\_month\_15 == 1 & current\_month\_15 == 0 & leave\_month\_15 == 1), 3,  
 ifelse(c(new\_month\_15 == 0 & current\_month\_15 == 1 & leave\_month\_15 == 1), 4, 3))))))),  
   
 status.04.19 = ifelse(c(new\_month\_16 == 0 & current\_month\_16 == 0 & leave\_month\_16 == 0), 0,  
 ifelse(c(new\_month\_16 == 1 & current\_month\_16 == 0 & leave\_month\_16 == 0), 1,   
 ifelse(c(new\_month\_16 == 0 & current\_month\_16 == 1 & leave\_month\_16 == 0), 2,   
 ifelse(c(new\_month\_16 == 0 & current\_month\_16 == 0 & leave\_month\_16 == 1), 4,  
 ifelse(c(new\_month\_16 == 1 & current\_month\_16 == 1 & leave\_month\_16 == 0), 1,  
 ifelse(c(new\_month\_16 == 1 & current\_month\_16 == 0 & leave\_month\_16 == 1), 3,  
 ifelse(c(new\_month\_16 == 0 & current\_month\_16 == 1 & leave\_month\_16 == 1), 4, 3))))))),  
   
 status.05.19 = ifelse(c(new\_month\_17 == 0 & current\_month\_17 == 0 & leave\_month\_17 == 0), 0,  
 ifelse(c(new\_month\_17 == 1 & current\_month\_17 == 0 & leave\_month\_17 == 0), 1,   
 ifelse(c(new\_month\_17 == 0 & current\_month\_17 == 1 & leave\_month\_17 == 0), 2,   
 ifelse(c(new\_month\_17 == 0 & current\_month\_17 == 0 & leave\_month\_17 == 1), 4,  
 ifelse(c(new\_month\_17 == 1 & current\_month\_17 == 1 & leave\_month\_17 == 0), 1,  
 ifelse(c(new\_month\_17 == 1 & current\_month\_17 == 0 & leave\_month\_17 == 1), 3,  
 ifelse(c(new\_month\_17 == 0 & current\_month\_17 == 1 & leave\_month\_17 == 1), 4, 3))))))),  
   
 status.06.19 = ifelse(c(new\_month\_18 == 0 & current\_month\_18 == 0 & leave\_month\_18 == 0), 0,  
 ifelse(c(new\_month\_18 == 1 & current\_month\_18 == 0 & leave\_month\_18 == 0), 1,   
 ifelse(c(new\_month\_18 == 0 & current\_month\_18 == 1 & leave\_month\_18 == 0), 2,   
 ifelse(c(new\_month\_18 == 0 & current\_month\_18 == 0 & leave\_month\_18 == 1), 4,  
 ifelse(c(new\_month\_18 == 1 & current\_month\_18 == 1 & leave\_month\_18 == 0), 1,  
 ifelse(c(new\_month\_18 == 1 & current\_month\_18 == 0 & leave\_month\_18 == 1), 3,  
 ifelse(c(new\_month\_18 == 0 & current\_month\_18 == 1 & leave\_month\_18 == 1), 4, 3))))))),  
   
 status.07.19 = ifelse(c(new\_month\_19 == 0 & current\_month\_19 == 0 & leave\_month\_19 == 0), 0,  
 ifelse(c(new\_month\_19 == 1 & current\_month\_19 == 0 & leave\_month\_19 == 0), 1,   
 ifelse(c(new\_month\_19 == 0 & current\_month\_19 == 1 & leave\_month\_19 == 0), 2,   
 ifelse(c(new\_month\_19 == 0 & current\_month\_19 == 0 & leave\_month\_19 == 1), 4,  
 ifelse(c(new\_month\_19 == 1 & current\_month\_19 == 1 & leave\_month\_19 == 0), 1,  
 ifelse(c(new\_month\_19 == 1 & current\_month\_19 == 0 & leave\_month\_19 == 1), 3,  
 ifelse(c(new\_month\_19 == 0 & current\_month\_19 == 1 & leave\_month\_19 == 1), 4, 3))))))),  
   
 status.08.19 = ifelse(c(new\_month\_20 == 0 & current\_month\_20 == 0 & leave\_month\_20 == 0), 0,  
 ifelse(c(new\_month\_20 == 1 & current\_month\_20 == 0 & leave\_month\_20 == 0), 1,   
 ifelse(c(new\_month\_20 == 0 & current\_month\_20 == 1 & leave\_month\_20 == 0), 2,   
 ifelse(c(new\_month\_20 == 0 & current\_month\_20 == 0 & leave\_month\_20 == 1), 4,  
 ifelse(c(new\_month\_20 == 1 & current\_month\_20 == 1 & leave\_month\_20 == 0), 1,  
 ifelse(c(new\_month\_20 == 1 & current\_month\_20 == 0 & leave\_month\_20 == 1), 3,  
 ifelse(c(new\_month\_20 == 0 & current\_month\_20 == 1 & leave\_month\_20 == 1), 4, 3))))))),  
   
 status.09.19 = ifelse(c(new\_month\_21 == 0 & current\_month\_21 == 0 & leave\_month\_21 == 0), 0,  
 ifelse(c(new\_month\_21 == 1 & current\_month\_21 == 0 & leave\_month\_21 == 0), 1,   
 ifelse(c(new\_month\_21 == 0 & current\_month\_21 == 1 & leave\_month\_21 == 0), 2,   
 ifelse(c(new\_month\_21 == 0 & current\_month\_21 == 0 & leave\_month\_21 == 1), 4,  
 ifelse(c(new\_month\_21 == 1 & current\_month\_21 == 1 & leave\_month\_21 == 0), 1,  
 ifelse(c(new\_month\_21 == 1 & current\_month\_21 == 0 & leave\_month\_21 == 1), 3,  
 ifelse(c(new\_month\_21 == 0 & current\_month\_21 == 1 & leave\_month\_21 == 1), 4, 3))))))),  
   
 status.10.19 = ifelse(c(new\_month\_22 == 0 & current\_month\_22 == 0 & leave\_month\_22 == 0), 0,  
 ifelse(c(new\_month\_22 == 1 & current\_month\_22 == 0 & leave\_month\_22 == 0), 1,   
 ifelse(c(new\_month\_22 == 0 & current\_month\_22 == 1 & leave\_month\_22 == 0), 2,   
 ifelse(c(new\_month\_22 == 0 & current\_month\_22 == 0 & leave\_month\_22 == 1), 4,  
 ifelse(c(new\_month\_22 == 1 & current\_month\_22 == 1 & leave\_month\_22 == 0), 1,  
 ifelse(c(new\_month\_22 == 1 & current\_month\_22 == 0 & leave\_month\_22 == 1), 3,  
 ifelse(c(new\_month\_22 == 0 & current\_month\_22 == 1 & leave\_month\_22 == 1), 4, 3))))))),  
   
 status.11.19 = ifelse(c(new\_month\_23 == 0 & current\_month\_23 == 0 & leave\_month\_23 == 0), 0,  
 ifelse(c(new\_month\_23 == 1 & current\_month\_23 == 0 & leave\_month\_23 == 0), 1,   
 ifelse(c(new\_month\_23 == 0 & current\_month\_23 == 1 & leave\_month\_23 == 0), 2,   
 ifelse(c(new\_month\_23 == 0 & current\_month\_23 == 0 & leave\_month\_23 == 1), 4,  
 ifelse(c(new\_month\_23 == 1 & current\_month\_23 == 1 & leave\_month\_23 == 0), 1,  
 ifelse(c(new\_month\_23 == 1 & current\_month\_23 == 0 & leave\_month\_23 == 1), 3,  
 ifelse(c(new\_month\_23 == 0 & current\_month\_23 == 1 & leave\_month\_23 == 1), 4, 3))))))),  
   
 status.12.19 = ifelse(c(new\_month\_24 == 0 & current\_month\_24 == 0 & leave\_month\_24 == 0), 0,  
 ifelse(c(new\_month\_24 == 1 & current\_month\_24 == 0 & leave\_month\_24 == 0), 1,   
 ifelse(c(new\_month\_24 == 0 & current\_month\_24 == 1 & leave\_month\_24 == 0), 2,   
 ifelse(c(new\_month\_24 == 0 & current\_month\_24 == 0 & leave\_month\_24 == 1), 4,  
 ifelse(c(new\_month\_24 == 1 & current\_month\_24 == 1 & leave\_month\_24 == 0), 1,  
 ifelse(c(new\_month\_24 == 1 & current\_month\_24 == 0 & leave\_month\_24 == 1), 3,  
 ifelse(c(new\_month\_24 == 0 & current\_month\_24 == 1 & leave\_month\_24 == 1), 4, 3))))))),  
   
   
 status.01.20 = ifelse(c(new\_month\_25 == 0 & current\_month\_25 == 0 & leave\_month\_25 == 0), 0,  
 ifelse(c(new\_month\_25 == 1 & current\_month\_25 == 0 & leave\_month\_25 == 0), 1,   
 ifelse(c(new\_month\_25 == 0 & current\_month\_25 == 1 & leave\_month\_25 == 0), 2,   
 ifelse(c(new\_month\_25 == 0 & current\_month\_25 == 0 & leave\_month\_25 == 1), 4,  
 ifelse(c(new\_month\_25 == 1 & current\_month\_25 == 1 & leave\_month\_25 == 0), 1,  
 ifelse(c(new\_month\_25 == 1 & current\_month\_25 == 0 & leave\_month\_25 == 1), 3,  
 ifelse(c(new\_month\_25 == 0 & current\_month\_25 == 1 & leave\_month\_25 == 1), 4, 3))))))),  
   
 status.02.20 = ifelse(c(new\_month\_26 == 0 & current\_month\_26 == 0 & leave\_month\_26 == 0), 0,  
 ifelse(c(new\_month\_26 == 1 & current\_month\_26 == 0 & leave\_month\_26 == 0), 1,   
 ifelse(c(new\_month\_26 == 0 & current\_month\_26 == 1 & leave\_month\_26 == 0), 2,   
 ifelse(c(new\_month\_26 == 0 & current\_month\_26 == 0 & leave\_month\_26 == 1), 4,  
 ifelse(c(new\_month\_26 == 1 & current\_month\_26 == 1 & leave\_month\_26 == 0), 1,  
 ifelse(c(new\_month\_26 == 1 & current\_month\_26 == 0 & leave\_month\_26 == 1), 3,  
 ifelse(c(new\_month\_26 == 0 & current\_month\_26 == 1 & leave\_month\_26 == 1), 4, 3))))))),  
   
 status.03.20 = ifelse(c(new\_month\_27 == 0 & current\_month\_27 == 0 & leave\_month\_27 == 0), 0,  
 ifelse(c(new\_month\_27 == 1 & current\_month\_27 == 0 & leave\_month\_27 == 0), 1,   
 ifelse(c(new\_month\_27 == 0 & current\_month\_27 == 1 & leave\_month\_27 == 0), 2,   
 ifelse(c(new\_month\_27 == 0 & current\_month\_27 == 0 & leave\_month\_27 == 1), 4,  
 ifelse(c(new\_month\_27 == 1 & current\_month\_27 == 1 & leave\_month\_27 == 0), 1,  
 ifelse(c(new\_month\_27 == 1 & current\_month\_27 == 0 & leave\_month\_27 == 1), 3,  
 ifelse(c(new\_month\_27 == 0 & current\_month\_27 == 1 & leave\_month\_27 == 1), 4, 3))))))),  
   
 status.04.20 = ifelse(c(new\_month\_28 == 0 & current\_month\_28 == 0 & leave\_month\_28 == 0), 0,  
 ifelse(c(new\_month\_28 == 1 & current\_month\_28 == 0 & leave\_month\_28 == 0), 1,   
 ifelse(c(new\_month\_28 == 0 & current\_month\_28 == 1 & leave\_month\_28 == 0), 2,   
 ifelse(c(new\_month\_28 == 0 & current\_month\_28 == 0 & leave\_month\_28 == 1), 4,  
 ifelse(c(new\_month\_28 == 1 & current\_month\_28 == 1 & leave\_month\_28 == 0), 1,  
 ifelse(c(new\_month\_28 == 1 & current\_month\_28 == 0 & leave\_month\_28 == 1), 3,  
 ifelse(c(new\_month\_28 == 0 & current\_month\_28 == 1 & leave\_month\_28 == 1), 4, 3))))))),  
   
 status.05.20 = ifelse(c(new\_month\_29 == 0 & current\_month\_29 == 0 & leave\_month\_29 == 0), 0,  
 ifelse(c(new\_month\_29 == 1 & current\_month\_29 == 0 & leave\_month\_29 == 0), 1,   
 ifelse(c(new\_month\_29 == 0 & current\_month\_29 == 1 & leave\_month\_29 == 0), 2,   
 ifelse(c(new\_month\_29 == 0 & current\_month\_29 == 0 & leave\_month\_29 == 1), 4,  
 ifelse(c(new\_month\_29 == 1 & current\_month\_29 == 1 & leave\_month\_29 == 0), 1,  
 ifelse(c(new\_month\_29 == 1 & current\_month\_29 == 0 & leave\_month\_29 == 1), 3,  
 ifelse(c(new\_month\_29 == 0 & current\_month\_29 == 1 & leave\_month\_29 == 1), 4, 3))))))),  
   
 status.06.20 = ifelse(c(new\_month\_30 == 0 & current\_month\_30 == 0 & leave\_month\_30 == 0), 0,  
 ifelse(c(new\_month\_30 == 1 & current\_month\_30 == 0 & leave\_month\_30 == 0), 1,   
 ifelse(c(new\_month\_30 == 0 & current\_month\_30 == 1 & leave\_month\_30 == 0), 2,   
 ifelse(c(new\_month\_30 == 0 & current\_month\_30 == 0 & leave\_month\_30 == 1), 4,  
 ifelse(c(new\_month\_30 == 1 & current\_month\_30 == 1 & leave\_month\_30 == 0), 1,  
 ifelse(c(new\_month\_30 == 1 & current\_month\_30 == 0 & leave\_month\_30 == 1), 3,  
 ifelse(c(new\_month\_30 == 0 & current\_month\_30 == 1 & leave\_month\_30 == 1), 4, 3))))))),  
   
 status.07.20 = ifelse(c(new\_month\_31 == 0 & current\_month\_31 == 0 & leave\_month\_31 == 0), 0,  
 ifelse(c(new\_month\_31 == 1 & current\_month\_31 == 0 & leave\_month\_31 == 0), 1,   
 ifelse(c(new\_month\_31 == 0 & current\_month\_31 == 1 & leave\_month\_31 == 0), 2,   
 ifelse(c(new\_month\_31 == 0 & current\_month\_31 == 0 & leave\_month\_31 == 1), 4,  
 ifelse(c(new\_month\_31 == 1 & current\_month\_31 == 1 & leave\_month\_31 == 0), 1,  
 ifelse(c(new\_month\_31 == 1 & current\_month\_31 == 0 & leave\_month\_31 == 1), 3,  
 ifelse(c(new\_month\_31 == 0 & current\_month\_31 == 1 & leave\_month\_31 == 1), 4, 3))))))),  
   
 status.08.20 = ifelse(c(new\_month\_32 == 0 & current\_month\_32 == 0 & leave\_month\_32 == 0), 0,  
 ifelse(c(new\_month\_32 == 1 & current\_month\_32 == 0 & leave\_month\_32 == 0), 1,   
 ifelse(c(new\_month\_32 == 0 & current\_month\_32 == 1 & leave\_month\_32 == 0), 2,   
 ifelse(c(new\_month\_32 == 0 & current\_month\_32 == 0 & leave\_month\_32 == 1), 4,  
 ifelse(c(new\_month\_32 == 1 & current\_month\_32 == 1 & leave\_month\_32 == 0), 1,  
 ifelse(c(new\_month\_32 == 1 & current\_month\_32 == 0 & leave\_month\_32 == 1), 3,  
 ifelse(c(new\_month\_32 == 0 & current\_month\_32 == 1 & leave\_month\_32 == 1), 4, 3))))))),  
   
 status.09.20 = ifelse(c(new\_month\_33 == 0 & current\_month\_33 == 0 & leave\_month\_33 == 0), 0,  
 ifelse(c(new\_month\_33 == 1 & current\_month\_33 == 0 & leave\_month\_33 == 0), 1,   
 ifelse(c(new\_month\_33 == 0 & current\_month\_33 == 1 & leave\_month\_33 == 0), 2,   
 ifelse(c(new\_month\_33 == 0 & current\_month\_33 == 0 & leave\_month\_33 == 1), 4,  
 ifelse(c(new\_month\_33 == 1 & current\_month\_33 == 1 & leave\_month\_33 == 0), 1,  
 ifelse(c(new\_month\_33 == 1 & current\_month\_33 == 0 & leave\_month\_33 == 1), 3,  
 ifelse(c(new\_month\_33 == 0 & current\_month\_33 == 1 & leave\_month\_33 == 1), 4, 3))))))),  
   
 status.10.20 = ifelse(c(new\_month\_34 == 0 & current\_month\_34 == 0 & leave\_month\_34 == 0), 0,  
 ifelse(c(new\_month\_34 == 1 & current\_month\_34 == 0 & leave\_month\_34 == 0), 1,   
 ifelse(c(new\_month\_34 == 0 & current\_month\_34 == 1 & leave\_month\_34 == 0), 2,   
 ifelse(c(new\_month\_34 == 0 & current\_month\_34 == 0 & leave\_month\_34 == 1), 4,  
 ifelse(c(new\_month\_34 == 1 & current\_month\_34 == 1 & leave\_month\_34 == 0), 1,  
 ifelse(c(new\_month\_34 == 1 & current\_month\_34 == 0 & leave\_month\_34 == 1), 3,  
 ifelse(c(new\_month\_34 == 0 & current\_month\_34 == 1 & leave\_month\_34 == 1), 4, 3)))))))  
 )  
  
bang = bang %>%   
 mutate(  
 status.01.18 = as.factor(status.01.18),  
 status.02.18 = as.factor(status.02.18),  
 status.03.18 = as.factor(status.03.18),  
 status.04.18 = as.factor(status.04.18),  
 status.05.18 = as.factor(status.05.18),  
 status.06.18 = as.factor(status.06.18),  
 status.07.18 = as.factor(status.07.18),  
 status.08.18 = as.factor(status.08.18),  
 status.09.18 = as.factor(status.09.18),  
 status.10.18 = as.factor(status.10.18),  
 status.11.18 = as.factor(status.11.18),  
 status.12.18 = as.factor(status.12.18),  
 status.01.19 = as.factor(status.01.19),  
 status.02.19 = as.factor(status.02.19),  
 status.03.19 = as.factor(status.03.19),  
 status.04.19 = as.factor(status.04.19),  
 status.05.19 = as.factor(status.05.19),  
 status.06.19 = as.factor(status.06.19),  
 status.07.19 = as.factor(status.07.19),  
 status.08.19 = as.factor(status.08.19),  
 status.09.19 = as.factor(status.09.19),  
 status.10.19 = as.factor(status.10.19),  
 status.11.19 = as.factor(status.11.19),  
 status.12.19 = as.factor(status.12.19),  
 status.01.20 = as.factor(status.01.20),  
 status.02.20 = as.factor(status.02.20),  
 status.03.20 = as.factor(status.03.20),  
 status.04.20 = as.factor(status.04.20),  
 status.05.20 = as.factor(status.05.20),  
 status.06.20 = as.factor(status.06.20),  
 status.07.20 = as.factor(status.07.20),  
 status.08.20 = as.factor(status.08.20),  
 status.09.20 = as.factor(status.09.20),  
 status.10.20 = as.factor(status.10.20)  
 )  
  
  
bang = bang %>%   
 mutate(  
 status.01.18 = revalue(status.01.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.02.18 = revalue(status.02.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.03.18 = revalue(status.03.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.04.18 = revalue(status.04.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.05.18 = revalue(status.05.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.06.18 = revalue(status.06.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.07.18 = revalue(status.07.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.08.18 = revalue(status.08.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.09.18 = revalue(status.09.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.10.18 = revalue(status.10.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.11.18 = revalue(status.11.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.12.18 = revalue(status.12.18, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.01.19 = revalue(status.01.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.02.19 = revalue(status.02.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.03.19 = revalue(status.03.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.04.19 = revalue(status.04.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.05.19 = revalue(status.05.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.06.19 = revalue(status.06.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.07.19 = revalue(status.07.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.08.19 = revalue(status.08.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.09.19 = revalue(status.09.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.10.19 = revalue(status.10.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.11.19 = revalue(status.11.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.12.19 = revalue(status.12.19, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.01.20 = revalue(status.01.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.02.20 = revalue(status.02.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.03.20 = revalue(status.03.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.04.20 = revalue(status.04.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.05.20 = revalue(status.05.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.06.20 = revalue(status.06.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.07.20 = revalue(status.07.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.08.20 = revalue(status.08.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.09.20 = revalue(status.09.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving')),  
 status.10.20 = revalue(status.10.20, c('0' = 'no', '1' = 'new/returning', '2' = 'current', '3' = 'start and left', '4' = 'leaving'))  
 )

## The following `from` values were not present in `x`: 3

## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3  
## The following `from` values were not present in `x`: 3

## The following `from` values were not present in `x`: 3, 4

### MEMBERSHIP RENEWAL

In determining the number of renewals, it was operated on the idea that it takes a minimum of 66 days to adopt a health behavior change, based on a study by [Lally et al. 2010](https://psycnet.apa.org/record/2010-22273-010). Despite our memberships are month-to-month, which would make sense to account as every month of continuation = 1 renewal, I have set it up so that renewals were all based on 2-month intervals throughout instead. Now the troublesome part comes with those that undergone less than 66 days (i.e. only stayed for one month). For those members, the renewals will now be set as 0. So, going through fixing this variable, I made the cut-off that those that engaged in less than 66 days will be listed as 0.

bang = bang %>%   
 mutate(  
 former\_renewal = ifelse(c(length < 66 & !is.na(num\_former\_reups)), 0, num\_former\_reups),   
 active\_renewal = ifelse(c(length < 66 & !is.na(num\_active\_reups)), 0, num\_active\_reups)  
 )  
  
bang = bang %>%   
 mutate(  
 num\_renewals = ifelse(current == 'active', active\_renewal,   
 ifelse(current == 'former', former\_renewal, NA))  
 ) %>%   
 mutate(  
 num\_renewals = as.numeric(num\_renewals)  
 )

### NUMBER OF PAYMENT BREAKS

As it was inevitable that there were cases where some members will have a break in their payment cycle due to various reasons. So we want to also take note of the number of times that this has happened. This will be recorded as a numeric.

bang = bang %>%   
 mutate(  
 num\_breaks = ifelse(c(current == "active" & !is.na(num\_active\_breaks)), num\_active\_breaks,  
 ifelse(c(current == 'former' & !is.na(num\_former\_breaks)), num\_former\_breaks, NA))   
 ) %>%   
 mutate(  
 num\_breaks = as.numeric(num\_breaks)  
 )

### MISSINGNESS

In making the data set, there were cases where I was not able to attain information on certain demographic variables (on a side note: this was pretty creepy and also scary to think that anyone can find stuff on you). So will create a variable to identify those with or without these pieces of information. This will be used in determining how we will be able to handle missing variables for our analysis.

# Missingness Variable  
  
bang = bang %>%   
 mutate(  
 missing\_value = ifelse(c(age\_group == 'na' | employment\_sector == 'na'), 'missing', 'not')  
 )  
  
bang = bang %>% mutate(  
 missing\_value = as.factor(missing\_value)  
 )

### START DATE & END DATE

I will be converting the information on start + end date of membership for each member. However, it should be noted that for the purpose of this analysis, those that are current members will have their end date listed as Oct 5th, 2020.

bang$start\_date = ymd(bang$start\_date)  
bang$end\_date = ymd(bang$end\_date)

### NUMBER OF SESSIONS

In order to calculate the attendance of the member, I’ve collected information pertaining to their attendance history such as number of attended sessions, number of cancelled classes (as confirmed on the scheduling software Wellness Living), number of lost session (based upon the projected theoretical total of sessions that could’ve been attended) and number of unconfirmed attendance. These variables should be reflected as numeric variables.

bang = bang %>%   
 mutate(  
 total\_sessions = as.numeric(total\_sessions),  
 attended = as.numeric(attended),  
 cancelled = as.numeric(cancelled),  
 lost = as.numeric(lost),  
 pending = as.numeric(pending)  
 )

### ATTENDANCE & CANCELLATION RATE

This variable is a numeric variable that sums up the attendance history of the member based on the following formulas: (a) Attendance rate = (Attended + (1/2 pending)) / Total Sessions (b) Cancellation rate = (canceled + lost + (1/2 pending)) / Total Sessions

The use of the “pending” items is based on the theoretical likelihood that there is a 50% chance that the member attended the appointment or not have attended the appointment.

# Attendance\_Rate + Cancellation\_Rate  
  
bang = bang %>%   
 mutate(  
 attendance\_rate = round(((attended + (0.5\*pending)) / total\_sessions) \* 100, 2),  
 cancellation\_rate = round(((cancelled + lost + (0.5\*pending))/total\_sessions) \* 100, 2)  
 )

(FUTURE MIKE) When doing preliminary analysis, I realized that I will be handling non-normally distributed data. Considering how this would cause violations in certain regression analyses, I will need to transform this data through multiple means to allow these assumptions to hold.

bang = bang %>%   
 mutate(  
 attendance\_rate\_group = ifelse(c(attendance\_rate >= 0 & attendance\_rate < 10), "0-9.99",  
 ifelse(c(attendance\_rate >= 10 & attendance\_rate < 20), "10-19.99",  
 ifelse(c(attendance\_rate >= 20 & attendance\_rate < 30), "20-29.99",  
 ifelse(c(attendance\_rate >= 30 & attendance\_rate < 40), "30-39.99",  
 ifelse(c(attendance\_rate >= 40 & attendance\_rate < 50), "40-49.99",  
 ifelse(c(attendance\_rate >= 50 & attendance\_rate < 60), "50-59.99",  
 ifelse(c(attendance\_rate >= 60 & attendance\_rate < 70), "60-69.99",  
 ifelse(c(attendance\_rate >= 70 & attendance\_rate < 80), "70-79.99",  
 ifelse(c(attendance\_rate >= 80 & attendance\_rate < 90), "80-89.99",  
 ifelse(c(attendance\_rate >= 90 & attendance\_rate <= 100), "90-100", NA)))))))))),   
   
 attendance\_grouping\_ver.1 = ifelse(c(attendance\_rate >= 0 & attendance\_rate < 50), '0-49.99',  
 ifelse(c(attendance\_rate >= 50 & attendance\_rate < 60), "50-59.99",  
 ifelse(c(attendance\_rate >= 60 & attendance\_rate < 70), "60-69.99",   
 ifelse(c(attendance\_rate >= 70 & attendance\_rate < 80), "70-79.99",   
 ifelse(c(attendance\_rate >= 80 & attendance\_rate < 90), "80-89.99",   
 ifelse(attendance\_rate >= 90, '90-100', NA)))))),   
   
 attendance\_cutoff = ifelse(attendance\_rate < 60, 'no', 'yes'),   
   
 attendance\_rate\_cutoff = ifelse(c(attendance\_rate >= 68.97 & membership == '1x'), "yes",  
 ifelse(c(attendance\_rate < 68.97 & membership == '1x'), 'no',   
 ifelse(c(attendance\_rate >= 75.00 & membership == '2x'), 'yes',   
 ifelse(c(attendance\_rate < 75.00 & membership == '2x'), 'no',  
 ifelse(c(attendance\_rate >= 61.81 & membership == '3x'), 'yes',  
 ifelse(c(attendance\_rate < 61.81 & membership == '3x'), 'no',  
 ifelse(c(attendance\_rate >= 47.13 & membership == '4x'), 'yes',  
 ifelse(c(attendance\_rate < 47.13 & membership == '4x'), 'no',  
 ifelse(c(attendance\_rate >= 36.41 & membership == 'unlimited'), 'yes',  
 ifelse(c(attendance\_rate < 36.41 & membership == 'unlimited'), 'no',  
 ifelse(c(attendance\_rate >= 50 & membership == 'group'), 'yes',  
 ifelse(c(attendance\_rate < 50 & membership == 'group'), 'no',  
 ifelse(c(attendance\_rate >= 42.71 & membership == 'distance'), 'yes', 'no')))))))))))))  
 ) %>%   
 mutate(  
 attendance\_rate\_cutoff = as.factor(attendance\_rate\_cutoff),  
 attendance\_cutoff = as.factor(attendance\_cutoff),  
 attendance\_grouping\_ver.1 = as.factor(attendance\_grouping\_ver.1),  
 attendance\_rate\_group = as.factor(attendance\_rate\_group)  
 )

### EMAIL INTERACTIONS

In assessing the sort of interactions that membership service teams have with members, I’ve exclusively used Emails as that was the only thing that had a paper trail. By paper-trail, I basically mean that counting the number of a particular email interactions from the inbox. I’ve categorized this into 4 categories:

**Table3. Types of Email Interactions**

|  |  |
| --- | --- |
| \*\* Email Type \*\* | \*\* Description \*\* |
| Billing | Refers to any email interaction relating to any inquiries or events of unknown charge and/or billing errors |
| CX | Refers to Customer Experience interactions that relate to check-in or anything not related to providing service or rescheduling |
| Scheduling | Refers to any email interactions that pertain to request for rescheduling or scheduling appointments (Hybrid Training Only) |
| Service | Refers to any requests or inquiries pertaining to administrative tasks, memberships, etc. that isn’t relating to scheduling |

Realizing that some folks may not have any other emails types than one particular kind, I will look at this in terms of percentage of total emails.

bang = bang %>%  
 mutate(  
 num\_ticket\_billing = as.numeric(num\_ticket\_billing),  
 num\_ticket\_scheduling = as.numeric(num\_ticket\_scheduling),  
 num\_ticket\_service = as.numeric(num\_ticket\_service),  
 num\_ticket\_cx = as.numeric(num\_ticket\_cx)  
 )  
  
bang = bang %>%   
 mutate(  
 num\_ticket\_total = c(num\_ticket\_billing + num\_ticket\_cx + num\_ticket\_scheduling + num\_ticket\_service)  
 ) %>%   
 mutate(  
 per\_ticket\_billing = round( (num\_ticket\_billing/num\_ticket\_total)\*100 , 2),  
 per\_ticket\_cx = round( (num\_ticket\_cx/num\_ticket\_total)\*100 , 2),  
 per\_ticket\_scheduling = round( (num\_ticket\_scheduling/num\_ticket\_total)\*100 , 2),  
 per\_ticket\_service = round( (num\_ticket\_service/num\_ticket\_total)\*100 , 2)  
 ) %>%   
 mutate(  
  
 per\_ticket\_billing = ifelse(!is.na(per\_ticket\_billing), per\_ticket\_billing, 0),  
 per\_ticket\_cx = ifelse(!is.na(per\_ticket\_cx), per\_ticket\_cx, 0),  
 per\_ticket\_scheduling = ifelse(!is.na(per\_ticket\_scheduling), per\_ticket\_scheduling, 0),  
 per\_ticket\_service = ifelse(!is.na(per\_ticket\_service), per\_ticket\_service, 0)  
 )

(FUTURE MIKE) I realized that billing-related emails isn’t really great in terms of treating it as a percentage since it’s fairly rare occurrence and will throw off the analyses downstream. So I will instead create two new variables to break this aspect down and separate it from the inclusion of email types. I’ve also rearranged percentage to only assess CX, Scheduling and Service-related emails. This will lead to having a new sum total of emails which include all types EXCEPT for Billing-related emails. Lastly, I realize that the total number of these emails may not be as useful considering that it’s an absolute number. Thus, I’ve create a measure of mean number of emails per month.

bang = bang %>%   
 mutate(  
 ever\_billing\_issue = ifelse(num\_ticket\_billing > 0, 'yes', 'no')  
 ) %>%   
 mutate(  
 ever\_billing\_issue = as.factor(ever\_billing\_issue)  
 )  
  
bang = bang %>%   
 mutate(  
 num\_billing\_issue = ifelse(num\_ticket\_billing == 0, "no billing issue",  
 ifelse(c(num\_ticket\_billing >= 0 & num\_ticket\_billing < 2), "one billing issue", "two or more billing issue"))   
 ) %>%   
 mutate(  
 num\_billing\_issue = as.factor(num\_billing\_issue)  
 )  
  
bang = bang %>%   
 mutate(  
 new\_num\_total = c(num\_ticket\_cx + num\_ticket\_scheduling + num\_ticket\_service)  
 ) %>%   
 mutate(  
 new\_per\_ticket\_cx = round((num\_ticket\_cx / new\_num\_total)\*100, 2),  
 new\_per\_ticket\_scheduling = round((num\_ticket\_scheduling / new\_num\_total)\*100, 2),  
 new\_per\_ticket\_service = round((num\_ticket\_service / new\_num\_total)\*100, 2)  
 ) %>%  
 mutate(  
 new\_per\_ticket\_cx = ifelse(!is.na(new\_per\_ticket\_cx), new\_per\_ticket\_cx, 0),   
 new\_per\_ticket\_scheduling = ifelse(!is.na(new\_per\_ticket\_scheduling), new\_per\_ticket\_scheduling, 0),  
 new\_per\_ticket\_service = ifelse(!is.na(new\_per\_ticket\_service), new\_per\_ticket\_service, 0)  
 )  
  
bang = bang %>%   
 mutate(  
 num\_emails\_month = round((new\_num\_total / length) \* 30.5, 3) # avg number of days in a month   
 ) %>%  
 mutate(  
 num\_emails\_month = as.numeric(num\_emails\_month)  
 )

### MONTHLY MEMBERSHIP RATES

In order to determine the monthly rates of a member, it will be based on the weighted average of monthly rates across each membership type that the member had engaged in.

bang = bang %>%   
 mutate(  
 active\_1x\_total = ifelse(c(active\_1x > 0 & active\_rate\_1x > 0), c(active\_1x \* active\_rate\_1x),   
 ifelse(c(active\_1x > 0 & active\_rate\_1x == 0), c(active\_1x \* active\_rate\_1x),   
 ifelse(c(active\_1x == 0 & active\_rate\_1x > 0), c(active\_1x \* active\_rate\_1x),   
 ifelse(c(active\_1x == 0 & active\_rate\_1x == 0), 0, NA))))  
 ) %>%  
 mutate(  
 active\_2x\_total = ifelse(c(active\_2x > 0 & active\_rate\_2x > 0), c(active\_2x \* active\_rate\_2x),   
 ifelse(c(active\_2x > 0 & active\_rate\_2x == 0), c(active\_2x \* active\_rate\_2x),   
 ifelse(c(active\_2x == 0 & active\_rate\_2x > 0), c(active\_2x \* active\_rate\_2x),   
 ifelse(c(active\_2x == 0 & active\_rate\_2x == 0), 0, NA))))  
 ) %>%   
 mutate(  
 active\_3x\_total = ifelse(c(active\_3x > 0 & active\_rate\_3x > 0), c(active\_3x \* active\_rate\_3x),   
 ifelse(c(active\_3x > 0 & active\_rate\_3x == 0), c(active\_3x \* active\_rate\_3x),   
 ifelse(c(active\_3x == 0 & active\_rate\_3x > 0), c(active\_3x \* active\_rate\_3x),   
 ifelse(c(active\_3x == 0 & active\_rate\_3x == 0), 0, NA))))  
 ) %>%   
 mutate(  
 active\_4x\_total = ifelse(c(active\_4x > 0 & active\_rate\_4x > 0), c(active\_4x \* active\_rate\_4x),   
 ifelse(c(active\_4x > 0 & active\_rate\_4x == 0), c(active\_4x \* active\_rate\_4x),   
 ifelse(c(active\_4x == 0 & active\_rate\_4x > 0), c(active\_4x \* active\_rate\_4x),   
 ifelse(c(active\_4x == 0 & active\_rate\_4x == 0), 0, NA))))  
 ) %>%   
 mutate(  
 active\_unlim\_total = ifelse(c(active\_unlim > 0 & active\_rate\_unlim > 0), c(active\_unlim \* active\_rate\_unlim),   
 ifelse(c(active\_unlim > 0 & active\_rate\_unlim == 0), c(active\_unlim \* active\_rate\_unlim),   
 ifelse(c(active\_unlim == 0 & active\_rate\_unlim > 0), c(active\_unlim \* active\_rate\_unlim),   
 ifelse(c(active\_unlim == 0 & active\_rate\_unlim == 0), 0, NA))))  
 ) %>%  
 mutate(  
 active\_group\_total = ifelse(c(active\_group > 0 & active\_rate\_group > 0), c(active\_group \* active\_rate\_group),   
 ifelse(c(active\_group > 0 & active\_rate\_group == 0), c(active\_group \* active\_rate\_group),   
 ifelse(c(active\_group == 0 & active\_rate\_group > 0), c(active\_group \* active\_rate\_group),   
 ifelse(c(active\_group == 0 & active\_rate\_group == 0), 0, NA))))  
 ) %>%   
 mutate(  
 active\_distance\_total = ifelse(c(active\_distance > 0 & active\_rate\_distance > 0), c(active\_distance \* active\_rate\_distance),   
 ifelse(c(active\_distance > 0 & active\_rate\_distance == 0), c(active\_distance \* active\_rate\_distance),   
 ifelse(c(active\_distance == 0 & active\_rate\_distance > 0), c(active\_distance \* active\_rate\_distance),   
 ifelse(c(active\_distance == 0 & active\_rate\_distance == 0), 0, NA))))  
 ) %>%   
 mutate(  
 former\_1x\_total = ifelse(c(former\_1x > 0 & former\_rate\_1x > 0), c(former\_1x \* former\_rate\_1x),   
 ifelse(c(former\_1x > 0 & former\_rate\_1x == 0), c(former\_1x \* former\_rate\_1x),   
 ifelse(c(former\_1x == 0 & former\_rate\_1x > 0), c(former\_1x \* former\_rate\_1x),   
 ifelse(c(former\_1x == 0 & former\_rate\_1x == 0), 0, NA))))  
 ) %>%  
 mutate(  
 former\_2x\_total = ifelse(c(former\_2x > 0 & former\_rate\_2x > 0), c(former\_2x \* former\_rate\_2x),   
 ifelse(c(former\_2x > 0 & former\_rate\_2x == 0), c(former\_2x \* former\_rate\_2x),   
 ifelse(c(former\_2x == 0 & former\_rate\_2x > 0), c(former\_2x \* former\_rate\_2x),   
 ifelse(c(former\_2x == 0 & former\_rate\_2x == 0), 0, NA))))  
 ) %>%   
 mutate(  
 former\_3x\_total = ifelse(c(former\_3x > 0 & former\_rate\_3x > 0), c(former\_3x \* former\_rate\_3x),   
 ifelse(c(former\_3x > 0 & former\_rate\_3x == 0), c(former\_3x \* former\_rate\_3x),   
 ifelse(c(former\_3x == 0 & former\_rate\_3x > 0), c(former\_3x \* former\_rate\_3x),   
 ifelse(c(former\_3x == 0 & former\_rate\_3x == 0), 0, NA))))  
 ) %>%   
 mutate(  
 former\_4x\_total = ifelse(c(former\_4x > 0 & former\_rate\_4x > 0), c(former\_4x \* former\_rate\_4x),   
 ifelse(c(former\_4x > 0 & former\_rate\_4x == 0), c(former\_4x \* former\_rate\_4x),   
 ifelse(c(former\_4x == 0 & former\_rate\_4x > 0), c(former\_4x \* former\_rate\_4x),   
 ifelse(c(former\_4x == 0 & former\_rate\_4x == 0), 0, NA))))  
 ) %>%   
 mutate(  
 former\_unlim\_total = ifelse(c(former\_unlim > 0 & former\_rate\_unlim > 0), c(former\_unlim \* former\_rate\_unlim),   
 ifelse(c(former\_unlim > 0 & former\_rate\_unlim == 0), c(former\_unlim \* former\_rate\_unlim),   
 ifelse(c(former\_unlim == 0 & former\_rate\_unlim > 0), c(former\_unlim \* former\_rate\_unlim),   
 ifelse(c(former\_unlim == 0 & former\_rate\_unlim == 0), 0, NA))))  
 ) %>%  
 mutate(  
 former\_group\_total = ifelse(c(former\_group > 0 & former\_rate\_group > 0), c(former\_group \* former\_rate\_group),   
 ifelse(c(former\_group > 0 & former\_rate\_group == 0), c(former\_group \* former\_rate\_group),   
 ifelse(c(former\_group == 0 & former\_rate\_group > 0), c(former\_group \* former\_rate\_group),   
 ifelse(c(former\_group == 0 & former\_rate\_group == 0), 0, NA))))  
 ) %>%   
 mutate(  
 former\_distance\_total = ifelse(c(former\_distance > 0 & former\_rate\_distance > 0), c(former\_distance \* former\_rate\_distance),   
 ifelse(c(former\_distance > 0 & former\_rate\_distance == 0), c(former\_distance \* former\_rate\_distance),   
 ifelse(c(former\_distance == 0 & former\_rate\_distance > 0), c(former\_distance \* former\_rate\_distance),   
 ifelse(c(former\_distance == 0 & former\_rate\_distance == 0), 0, NA))))  
 )  
  
  
bang = bang %>%   
 mutate(  
 avg\_monthly\_rate = ifelse(current == 'active', round((active\_1x\_total + active\_2x\_total + active\_3x\_total + active\_4x\_total + active\_unlim\_total + active\_group\_total + active\_distance\_total) / (active\_1x + active\_2x + active\_3x + active\_4x + active\_unlim + active\_group + active\_distance), 2),  
 ifelse(current == 'former', round((former\_1x\_total + former\_2x\_total + former\_3x\_total + former\_4x\_total + former\_unlim\_total + former\_group\_total + former\_distance\_total) / (former\_1x + former\_2x + former\_3x + former\_4x + former\_unlim + former\_group + former\_distance), 2), NA))  
 )

(Future Mike here) Realized that I will run into issues with making regression models due to issues of assumptions not being met (like linearity, proportionality, etc.) So to correct for these issues, I will try to categorize monthly rate in the same way that I had done earlier with attendance rate.

bang = bang %>%   
 mutate(  
 monthly\_rate\_group = ifelse(c(avg\_monthly\_rate >= 0 & avg\_monthly\_rate < 50), "0-99.99",  
 ifelse(c(avg\_monthly\_rate >= 50 & avg\_monthly\_rate < 100), "0-99.99",  
 ifelse(c(avg\_monthly\_rate >= 100 & avg\_monthly\_rate < 150), "100-149.99",  
 ifelse(c(avg\_monthly\_rate >= 150 & avg\_monthly\_rate < 200), "150-199.99",  
 ifelse(c(avg\_monthly\_rate >= 200 & avg\_monthly\_rate < 250), "200-249.99",  
 ifelse(c(avg\_monthly\_rate >= 250 & avg\_monthly\_rate < 300), "250-299.99",  
 ifelse(c(avg\_monthly\_rate >= 300 & avg\_monthly\_rate < 350), "300-349.99",  
 ifelse(c(avg\_monthly\_rate >= 350 & avg\_monthly\_rate < 400), "350-399.99",  
 ifelse(c(avg\_monthly\_rate >= 400 & avg\_monthly\_rate < 450), "400-449.99",  
 ifelse(c(avg\_monthly\_rate >= 450 & avg\_monthly\_rate < 500), "450-499.99",  
 ifelse(c(avg\_monthly\_rate >= 500 & avg\_monthly\_rate < 550), "500-549.99",  
 ifelse(c(avg\_monthly\_rate >= 550 & avg\_monthly\_rate < 600), "550-599.99",  
 ifelse(c(avg\_monthly\_rate >= 600), "600+", NA ))))))))))))),  
   
 monthly\_grouping\_ver.1 = ifelse(c(avg\_monthly\_rate >= 0 & avg\_monthly\_rate < 150), "0-149.99",  
 ifelse(c(avg\_monthly\_rate >= 150 & avg\_monthly\_rate < 200), "150-199.99",   
 ifelse(c(avg\_monthly\_rate >= 200 & avg\_monthly\_rate < 300), "200-299.99",   
 ifelse(c(avg\_monthly\_rate >= 300 & avg\_monthly\_rate < 350), "300-349.99",  
 ifelse(c(avg\_monthly\_rate >= 350 & avg\_monthly\_rate < 400), "350-399.99",  
 ifelse(c(avg\_monthly\_rate >= 400 & avg\_monthly\_rate < 450), "400-449.99",  
 ifelse(c(avg\_monthly\_rate >= 450 & avg\_monthly\_rate < 500), "450-499.99",   
 ifelse(avg\_monthly\_rate >= 500, "500+", NA)))))))),   
   
 monthly\_rate\_cutoff = ifelse(c(avg\_monthly\_rate >= 246.76 & membership == '1x'), 'yes',  
 ifelse(c(avg\_monthly\_rate < 246.76 & membership == '1x'), 'no',  
 ifelse(c(avg\_monthly\_rate >= 340.10 & membership == '2x'), 'yes',  
 ifelse(c(avg\_monthly\_rate < 340.10 & membership == '2x'), 'no',  
 ifelse(c(avg\_monthly\_rate >= 377.60 & membership == '3x'), 'yes',  
 ifelse(c(avg\_monthly\_rate < 377.60 & membership == '3x'), 'no',  
 ifelse(c(avg\_monthly\_rate >= 412.90 & membership == '4x'), 'yes',  
 ifelse(c(avg\_monthly\_rate < 412.90 & membership == '4x'), 'no',  
 ifelse(c(avg\_monthly\_rate >= 450.90 & membership == 'unlimited'), 'yes',  
 ifelse(c(avg\_monthly\_rate < 450.90 & membership == 'unlimited'), 'no',  
 ifelse(c(avg\_monthly\_rate >= 226 & membership == 'group'), 'yes',  
 ifelse(c(avg\_monthly\_rate < 226 & membership == 'group'), 'no',  
 ifelse(c(avg\_monthly\_rate >= 231.50 & membership == 'distance'), 'yes', 'no')))))))))))))  
 ) %>%   
 mutate(  
 monthly\_rate\_group = as.factor(monthly\_rate\_group),  
 monthly\_grouping\_ver.1 = as.factor(monthly\_grouping\_ver.1),  
 monthly\_rate\_cutoff = as.factor(monthly\_rate\_cutoff)  
 )

(Future Mike here) At some point in analysing the distribution of certain data, namely the non-billing related emails. There happens to be some evidence of zero-inflation, which is to be expected since some folks are just not the email type of folks. Seeing as this will inevitably give me some suspect findings down the road, I’ll use the advice found [here](https://stats.stackexchange.com/questions/56306/time-spent-in-an-activity-as-an-independent-variable) and [here](https://stats.stackexchange.com/questions/62983/zero-inflated-predictors-in-regression) to have **both** a continuous variable of percent composition of a specific non-billing type of email interaction along with a variable that dictates whether there is a particular type of email interaction or not. Hopefully, this will give some clarity on the impact of that variable once we get into regression analyses.

bang = bang %>%   
 mutate(  
 ever\_email\_month = ifelse(num\_emails\_month < 0.5, '0', '1'),  
 ever\_cx = ifelse(new\_per\_ticket\_cx == 0, "0", "1"),  
 ever\_scheduling = ifelse(new\_per\_ticket\_scheduling == 0, "0", "1"),  
 ever\_service = ifelse(new\_per\_ticket\_service == 0, "0", "1")  
 ) %>%   
 mutate(  
 ever\_email\_month = as.factor(ever\_email\_month),  
 ever\_cx = as.factor(ever\_cx),  
 ever\_scheduling = as.factor(ever\_scheduling),  
 ever\_service = as.factor(ever\_service),  
 ) %>%   
 mutate(  
 ever\_email\_month = revalue(ever\_email\_month, c('0' = 'no', '1' = 'yes')),  
 ever\_cx = revalue(ever\_cx, c('0' = "no", '1' = "yes")),   
 ever\_scheduling = revalue(ever\_scheduling, c('0' = "no", '1' = "yes")),   
 ever\_service = revalue(ever\_service, c('0' = "no", '1' = "yes")),   
 )

## **Creating Multiple Data Sets**

In order to be observe and perform our analysis the way that is intended, I’ve divided the data across multiple data sets.The initial data set will only include the necessary data for our analysis. This will be considered our “final” data set that includes those with missing demographics variables. I’ve also made a series of data sets based on the month and year.

finalized\_bang = bang %>%   
 select(id, age\_group, employment\_sector, missing\_value, current,   
 start\_date, end\_date, length, retention\_3m, retention\_6m, retention\_12m, revenue\_lifetime,  
 reason\_to\_leave, expected\_churn, unexpected\_churn, avoidable\_churn, unavoidable\_churn,  
 churn\_type, membership, num\_membership\_change, num\_breaks, num\_active\_breaks,  
 num\_former\_breaks, num\_renewals, avg\_monthly\_rate, monthly\_rate\_group, monthly\_grouping\_ver.1, monthly\_rate\_cutoff,  
 active\_1x, active\_rate\_1x, active\_2x, active\_rate\_2x, active\_3x, active\_rate\_3x, active\_4x, active\_rate\_4x, active\_unlim, active\_rate\_unlim,  
 active\_group, active\_rate\_group, active\_distance, active\_rate\_distance, active\_renewal, former\_1x, former\_rate\_1x,  
 former\_2x, former\_rate\_2x, former\_3x, former\_rate\_3x, former\_4x, former\_rate\_4x, former\_unlim, former\_rate\_unlim,   
 former\_group, former\_rate\_group, former\_distance, former\_rate\_distance, former\_renewal, total\_sessions, attended,   
 cancelled, lost, pending, attendance\_rate, attendance\_rate\_group, attendance\_grouping\_ver.1, attendance\_rate\_cutoff, cancellation\_rate, num\_ticket\_billing,  
 num\_ticket\_cx, num\_ticket\_scheduling, num\_ticket\_service, num\_ticket\_total, per\_ticket\_billing, per\_ticket\_cx, per\_ticket\_scheduling, per\_ticket\_service,  
 ever\_billing\_issue, num\_billing\_issue, new\_num\_total, num\_emails\_month, ever\_email\_month, new\_per\_ticket\_cx, ever\_cx, new\_per\_ticket\_scheduling, ever\_scheduling,  
 new\_per\_ticket\_service, ever\_service, status.01.18,status.02.18, status.03.18, status.04.18, status.05.18, status.06.18,   
 status.07.18, status.08.18, status.09.18, status.10.18, status.11.18, status.12.18, status.01.19, status.02.19, status.03.19, status.04.19,   
 status.05.19, status.06.19, status.07.19, status.08.19, status.09.19, status.10.19, status.11.19, status.12.19, status.01.20, status.02.20,   
 status.03.20, status.04.20, status.05.20, status.06.20, status.07.20, status.08.20, status.09.20, status.10.20)  
  
January\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.01.18 != "no")  
  
  
February\_2018 = finalized\_bang %>%   
 select(-status.01.18, -status.03.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.02.18 != "no")  
  
  
March\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.01.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.03.18 != "no")  
  
April\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.01.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.04.18 != "no")  
  
  
May\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.05.18 != "no")  
  
June\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.01.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.06.18 != "no")  
  
July\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.01.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.07.18 != "no")  
  
August\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.01.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.08.18 != "no")  
  
September\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.01.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.09.18 != "no")  
  
October\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.01.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.10.18 != "no")  
  
November\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.01.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.11.18 != "no")  
  
December\_2018 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.01.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.12.18 != "no")  
  
  
  
January\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.18,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.01.19 != "no")  
  
  
February\_2019 = finalized\_bang %>%   
 select(-status.01.18, -status.03.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.01.18, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.02.19 != "no")  
  
  
March\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.01.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.01.18, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.03.19 != "no")  
  
April\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.01.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.01.18, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.04.19 != "no")  
  
  
May\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.01.18, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.05.19 != "no")  
  
June\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.01.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.01.18, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.06.19 != "no")  
  
July\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.01.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.01.18,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.07.19 != "no")  
  
August\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.01.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.01.18, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.08.19 != "no")  
  
September\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.01.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.01.18, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.09.19 != "no")  
  
October\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.01.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.01.18, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.10.19 != "no")  
  
November\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.01.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.01.18, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.11.19 != "no")  
  
December\_2019 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.01.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.01.18, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.12.19 != "no")  
  
  
  
January\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.18,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.18,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.01.20 != "no")  
  
  
February\_2020 = finalized\_bang %>%   
 select(-status.01.18, -status.03.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.01.18, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.01.18, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.02.20 != "no")  
  
  
March\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.01.18, -status.04.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.01.18, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.01.18, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.03.20 != "no")  
  
April\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.01.18, -status.05.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.01.18, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.01.18, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.04.20 != "no")  
  
  
May\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.01.18, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.01.18, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.05.20 != "no")  
  
June\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.01.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.01.18, -status.07.19,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.01.18, -status.07.20,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.06.20 != "no")  
  
July\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.01.18,   
 -status.08.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.01.18,  
 -status.08.19, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.01.18,  
 -status.08.20, -status.09.20, -status.10.20) %>%   
 filter(status.07.20 != "no")  
  
August\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.01.18, -status.09.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.01.18, -status.09.19, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.01.18, -status.09.20, -status.10.20) %>%   
 filter(status.08.20 != "no")  
  
September\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.01.18, -status.10.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.01.18, -status.10.19, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.01.18, -status.10.20) %>%   
 filter(status.09.20 != "no")  
  
October\_2020 = finalized\_bang %>%   
 select(-status.02.18, -status.03.18, -status.04.18, -status.01.18, -status.06.18, -status.07.18,   
 -status.08.18, -status.09.18, -status.01.18, -status.11.18, -status.12.18, -status.01.19,  
 -status.02.19, -status.03.19, -status.04.19, -status.05.19, -status.06.19, -status.07.19,  
 -status.08.19, -status.09.19, -status.01.18, -status.11.19, -status.12.19, -status.01.20,  
 -status.02.20, -status.03.20, -status.04.20, -status.05.20, -status.06.20, -status.07.20,  
 -status.08.20, -status.09.20, -status.01.18) %>%   
 filter(status.10.20 != "no")

### HANDLING INCONSISTENT FINDINGS & MISSINGNESS

Now, as this data set was compiled by me, there would definitely be typo errors, missing values and irregularities with some of the entries. Using both Wellness Living and Air Table as a final check for entries in the data set, adjustment were made in filling out incorrect entries. As for unknown variables, they only pertain to demographics (i.e. age and/or employment sector). Looking at the make up of this group, it seems to account for 7.45% of the entire data set. While the best bet is to just drop this subset, there could potentially be some bias introduced as a result. So, the plan is to create an entire subset of this data and compare its descriptive findings with the non-missing data to assert if there is any noted difference. However, for the sake of performing inferential statistics, we will only keep entries with no missing variables in the age and employment sector.

clean\_bang = finalized\_bang %>% filter(missing\_value == 'not')  
  
clean\_bang = clean\_bang %>%   
 mutate(  
 age\_group = ifelse(age\_group == "Under 18", "Under 18",  
 ifelse(age\_group == "18-29", "18-29",  
 ifelse(age\_group == "30-44", "30-44",  
 ifelse(age\_group == "45-64", "45-64",  
 ifelse(age\_group == "65+", "65+", NA)))))  
 )  
  
clean\_bang = clean\_bang %>%   
 mutate(   
 employment\_sector = ifelse(employment\_sector == "Finance/Insurance", "Finance/Insurance",  
 ifelse(employment\_sector == "Transportation", 'Transportation',   
 ifelse(employment\_sector == "Health Care/Services", "Health Care/Services",  
 ifelse(employment\_sector == "Professional/Technical Services", "Professional/Technical Services",  
 ifelse(employment\_sector == "Hospitality/Retail/Accomodation", "Hospitality/Retail/Accomodation",  
 ifelse(employment\_sector == "Student", "Student",  
 ifelse(employment\_sector == "Entrepreneural/Owns Business", "Entrepreneural/Owns Business",  
 ifelse(employment\_sector == "Other", "Other",  
 ifelse(employment\_sector == "Scientific/Academic/Educational", "Scientific/Academic/Educational",  
 ifelse(employment\_sector == "Technology/Information", "Technology/Information",  
 ifelse(employment\_sector == "Social Services/Non-Profits", "Social Services/Non-Profits",  
 ifelse(employment\_sector == "Government/Legal", "Government/Legal",  
 ifelse(employment\_sector == "Advertising/Media/Art/Culture", "Advertising/Media/Art/Culture",   
 ifelse(employment\_sector == "Real Estate/Construction/Waste", "Real Estate/Construction/Waste",  
 ifelse(employment\_sector == "Natural Resource/Energy", "Natural Resource/Energy",  
 ifelse(employment\_sector == "Manufacturing/Trade", "Manufacturing/Trade", NA))))))))))))))))  
 )  
   
clean\_bang = clean\_bang %>%   
 mutate(  
 reason\_to\_leave = ifelse(reason\_to\_leave == "loss of employment", "loss of employment",  
 ifelse(reason\_to\_leave == "pandemic/global crisis", "pandemic/global crisis",  
 ifelse(reason\_to\_leave == "other", "other",  
 ifelse(reason\_to\_leave == "financial", "financial",  
 ifelse(reason\_to\_leave == "medical/health-related", "medical/health-related",  
 ifelse(reason\_to\_leave == "moving away", "moving away",  
 ifelse(reason\_to\_leave == "lacking accessibility/availability", "lacking accessibility/availability",  
 ifelse(reason\_to\_leave == "other fitness interest", "other fitness interest",  
 ifelse(reason\_to\_leave == "ghosted us", "ghosted us",  
 ifelse(reason\_to\_leave == "time-based arrangement", "time-based arrangement",  
 ifelse(reason\_to\_leave == "noted displeasure with Bang", "noted displeasure with Bang", NA)))))))))))  
 )  
  
clean\_bang = clean\_bang %>%   
 mutate(   
 churn\_type = ifelse(churn\_type == "unexpected + unavoidable", "unexpected + unavoidable",  
 ifelse(churn\_type == "unexpected + avoidable", "unexpected + avoidable",  
 ifelse(churn\_type == "expected + unavoidable", "expected + unavoidable",  
 ifelse(churn\_type == "expected + avoidable", "expected + avoidable", NA))))  
 )  
  
  
clean\_bang = clean\_bang %>%   
 mutate(  
 age\_group = as.factor(age\_group),  
 employment\_sector = as.factor(employment\_sector),  
 reason\_to\_leave = as.factor(reason\_to\_leave),  
 churn\_type = as.factor(churn\_type)  
 )  
  
  
clean\_bang = clean\_bang %>%   
 mutate(  
 age\_group = relevel(age\_group, ref = "Under 18"),  
 employment\_sector = relevel(employment\_sector, ref = "Other"),  
 current = relevel(current, ref = "former"),  
 retention\_3m = relevel(retention\_3m, ref = 'no'),  
 retention\_6m = relevel(retention\_6m, ref = 'no'),  
 retention\_12m = relevel(retention\_12m, ref = 'no'),  
 reason\_to\_leave = relevel(reason\_to\_leave, ref = "other"),  
 churn\_type = relevel(churn\_type, ref = "unexpected + unavoidable"),  
 membership = relevel(membership, ref = "1x")  
 )  
  
clean\_bang\_final <- clean\_bang %>%   
 mutate(  
 reason\_to\_leave = ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "loss of employment"), "loss of employment",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "pandemic/global crisis"), "pandemic/global crisis",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "other"), "other",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "financial"), "financial",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "medical/health-related"), "medical/health-related",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "moving away"), "moving away",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "lacking accessibility/availability"), "lacking accessibility/availability",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "other fitness interest"), "other fitness interest",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "ghosted us"), "ghosted us",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "time-based arrangement"), "time-based arrangement",  
 ifelse(c(!is.na(reason\_to\_leave) & reason\_to\_leave == "noted displeasure with Bang"), "noted displeasure with Bang", "still a member")))))))))))  
 ) %>%   
 mutate(  
 churn\_type = ifelse(c(!is.na(churn\_type) & churn\_type == "unexpected + unavoidable"), "unexpected + unavoidable",  
 ifelse(c(!is.na(churn\_type) & churn\_type == "expected + unavoidable"), "expected + unavoidable",  
 ifelse(c(!is.na(churn\_type) & churn\_type == "unexpected + avoidable"), "unexpected + avoidable",  
 ifelse(c(!is.na(churn\_type) & churn\_type == "expected + avoidable"), "expected + avoidable", "still a member" ))))  
   
 ) %>%   
 mutate(  
 reason\_to\_leave = as.factor(reason\_to\_leave),  
 churn\_type = as.factor(churn\_type)  
 )  
  
clean\_bang\_final = clean\_bang\_final %>%   
 mutate(  
 reason\_to\_leave = relevel(reason\_to\_leave, ref = "still a member"),  
 churn\_type = relevel(churn\_type, ref = "still a member")  
 )   
  
clean\_bang\_final = clean\_bang\_final %>%  
 mutate(  
 expected\_churn = ifelse(c(churn\_type != 'still a member' & expected\_churn == "yes"), "yes",   
 ifelse(c(churn\_type != 'still a member' & expected\_churn == "no"), "no", NA)),  
 unexpected\_churn = ifelse(c(churn\_type != 'still a member' & unexpected\_churn == "yes"), "yes",   
 ifelse(c(churn\_type != 'still a member' & unexpected\_churn == "no"), "no", NA)),  
 avoidable\_churn = ifelse(c(churn\_type != 'still a member' & avoidable\_churn == "yes"), "yes",   
 ifelse(c(churn\_type != 'still a member' & avoidable\_churn == "no"), "no", NA)),  
 unavoidable\_churn = ifelse(c(churn\_type != 'still a member' & unavoidable\_churn == "yes"), "yes",   
 ifelse(c(churn\_type != 'still a member' & unavoidable\_churn == "no"), "no", NA))  
 )   
  
clean\_bang\_final = clean\_bang\_final %>%  
 mutate(  
 expected\_churn = as.factor(expected\_churn),  
 unexpected\_churn = as.factor(unexpected\_churn),  
 avoidable\_churn = as.factor(avoidable\_churn),  
 unavoidable\_churn = as.factor(unavoidable\_churn)  
 ) %>%   
 mutate(  
 expected\_churn = relevel(expected\_churn, ref = "no"),  
 unexpected\_churn = relevel(unexpected\_churn, ref = "no"),  
 unavoidable\_churn = relevel(unavoidable\_churn, ref = "no"),  
 avoidable\_churn = relevel(avoidable\_churn, ref = 'no')  
 )  
  
  
clean\_bang\_final = clean\_bang\_final %>%   
 mutate(  
 num\_ticket\_total = c(num\_ticket\_billing + num\_ticket\_cx + num\_ticket\_scheduling + num\_ticket\_service)  
 ) %>%   
 mutate(  
 per\_ticket\_billing = round( (num\_ticket\_billing/num\_ticket\_total)\*100 , 2),  
 per\_ticket\_cx = round( (num\_ticket\_cx/num\_ticket\_total)\*100 , 2),  
 per\_ticket\_scheduling = round( (num\_ticket\_scheduling/num\_ticket\_total)\*100 , 2),  
 per\_ticket\_service = round( (num\_ticket\_service/num\_ticket\_total)\*100 , 2)  
 ) %>%   
 mutate(  
  
 per\_ticket\_billing = ifelse(!is.na(per\_ticket\_billing), per\_ticket\_billing, 0),  
 per\_ticket\_cx = ifelse(!is.na(per\_ticket\_cx), per\_ticket\_cx, 0),  
 per\_ticket\_scheduling = ifelse(!is.na(per\_ticket\_scheduling), per\_ticket\_scheduling, 0),  
 per\_ticket\_service = ifelse(!is.na(per\_ticket\_service), per\_ticket\_service, 0)  
 )  
  
  
clean\_bang\_final = clean\_bang\_final %>%   
 mutate(  
 became\_former\_member = ifelse(current == "active", 0, 1)  
 )

## CREATING MULTIPLE DATA SETS

Additional data sets will be created that are divided based on:

1. only Hybrid vs Group vs Distance
2. former vs active members
3. age groups
4. employment sectors
5. membership types
6. retention status at 3/6/12 months
7. reasons to leave Bang Personal Training
8. churn type
9. membership status at a given month/year
10. monthly membership rates
11. attendance rates

I’ve also created pivoted data sets which stack related data together to get a more comprehensive look at the distribution of data. These include types of email interactions, monthly membership updates, etc.

clean\_bang\_hybrid\_only = clean\_bang\_final %>% filter((active\_group == 0 & active\_distance == 0) | (former\_group == 0 & former\_distance == 0))   
  
clean\_bang\_distance\_only = clean\_bang\_final %>%   
 filter(  
 (active\_group == 0 & (active\_1x == 0 & active\_2x == 0 & active\_3x == 0 & active\_4x == 0 & active\_unlim == 0)) |   
 (former\_group == 0 & (former\_1x == 0 & former\_2x == 0 & former\_3x == 0 & former\_4x == 0 & former\_unlim == 0))   
 )  
  
clean\_bang\_group\_only = clean\_bang\_final %>%   
 filter(  
 (active\_distance == 0 & (active\_1x == 0 & active\_2x == 0 & active\_3x == 0 & active\_4x == 0 & active\_unlim == 0)) |   
 (former\_distance == 0 & (former\_1x == 0 & former\_2x == 0 & former\_3x == 0 & former\_4x == 0 & former\_unlim == 0))   
 )   
  
unclean\_bang\_hybrid\_only = finalized\_bang %>% filter((active\_group == 0 & active\_distance == 0) | (former\_group == 0 & former\_distance == 0))  
  
unclean\_bang\_distance\_only = finalized\_bang %>%   
 filter(  
 (active\_group == 0 & (active\_1x == 0 & active\_2x == 0 & active\_3x == 0 & active\_4x == 0 & active\_unlim == 0)) |   
 (former\_group == 0 & (former\_1x == 0 & former\_2x == 0 & former\_3x == 0 & former\_4x == 0 & former\_unlim == 0))   
 )  
  
unclean\_bang\_group\_only = finalized\_bang %>%   
 filter(  
 (active\_distance == 0 & (active\_1x == 0 & active\_2x == 0 & active\_3x == 0 & active\_4x == 0 & active\_unlim == 0)) |   
 (former\_distance == 0 & (former\_1x == 0 & former\_2x == 0 & former\_3x == 0 & former\_4x == 0 & former\_unlim == 0))   
 )   
  
  
unclean\_former\_bang = finalized\_bang %>% filter(current == "former")  
unclean\_active\_bang = finalized\_bang %>% filter(current == 'active')  
former\_bang = clean\_bang\_final %>% filter(current == "former")  
active\_bang = clean\_bang\_final %>% filter(current == 'active')  
  
unclean\_bang\_under\_18 = finalized\_bang %>% filter(age\_group == "Under 18")  
unclean\_bang\_30\_to\_44 = finalized\_bang %>% filter(age\_group == "30-44")  
unclean\_bang\_45\_to\_64 = finalized\_bang %>% filter(age\_group == "45-64")  
unclean\_bang\_65 = finalized\_bang %>% filter(age\_group == "65+")  
unclean\_bang\_na = finalized\_bang %>% filter(age\_group == "na")  
bang\_under\_18 = clean\_bang\_final %>% filter(age\_group == "Under 18")  
bang\_30\_to\_44 = clean\_bang\_final %>% filter(age\_group == "30-44")  
bang\_45\_to\_64 = clean\_bang\_final %>% filter(age\_group == "45-64")  
bang\_65 = clean\_bang\_final %>% filter(age\_group == "65+")  
  
  
unclean\_bang\_finance = finalized\_bang %>% filter(employment\_sector == "Finance/Insurance")  
unclean\_bang\_transport = finalized\_bang %>% filter(employment\_sector == "Transportation")  
unclean\_bang\_health = finalized\_bang %>% filter(employment\_sector == "Health Care/Services")  
unclean\_bang\_pro = finalized\_bang%>% filter(employment\_sector == "Professional/Technical Services")  
unclean\_bang\_stu = finalized\_bang %>% filter(employment\_sector == "Student")  
unclean\_bang\_hra = finalized\_bang %>% filter(employment\_sector == "Hospitality/Retail/Accomodation")  
unclean\_bang\_entra = finalized\_bang %>% filter(employment\_sector == "Entrepreneural/Owns Business")  
unclean\_bang\_oth = finalized\_bang %>% filter(employment\_sector == "Other")  
unclean\_bang\_scixedu = finalized\_bang %>% filter(employment\_sector == "Scientific/Academic/Education")  
unclean\_bang\_tech = finalized\_bang %>% filter(employment\_sector == "Technology/Information")  
unclean\_bang\_soc = finalized\_bang %>% filter(employment\_sector == "Social Services/Non-Profits")  
unclean\_bang\_govxleg = finalized\_bang %>% filter(employment\_sector == "Government/Legal")  
unclean\_bang\_art = finalized\_bang %>% filter(employment\_sector == "Advertising/Media/Art/Culture")  
unclean\_bang\_rcw = finalized\_bang %>% filter(employment\_sector == "Real Estate/Construction/Waste")  
unclean\_bang\_nr = finalized\_bang %>% filter(employment\_sector == "Natural Resource/Energy")  
unclean\_bang\_manu = finalized\_bang %>% filter(employment\_sector == "Manufacturing/Trade")  
unclean\_bang\_na = finalized\_bang %>% filter(employment\_sector == "na")  
bang\_finance = clean\_bang\_final %>% filter(employment\_sector == "Finance/Insurance")  
bang\_transport = clean\_bang\_final %>% filter(employment\_sector == "Transportation")  
bang\_health = clean\_bang\_final %>% filter(employment\_sector == "Health Care/Services")  
bang\_pro = clean\_bang\_final %>% filter(employment\_sector == "Professional/Technical Services")  
bang\_stu = clean\_bang\_final %>% filter(employment\_sector == "Student")  
bang\_hra = clean\_bang\_final %>% filter(employment\_sector == "Hospitality/Retail/Accomodation")  
bang\_entra = clean\_bang\_final %>% filter(employment\_sector == "Entrepreneural/Owns Business")  
bang\_oth = clean\_bang\_final %>% filter(employment\_sector == "Other")  
bang\_scixedu = clean\_bang\_final %>% filter(employment\_sector == "Scientific/Academic/Education")  
bang\_tech = clean\_bang\_final %>% filter(employment\_sector == "Technology/Information")  
bang\_soc = clean\_bang\_final %>% filter(employment\_sector == "Social Services/Non-Profits")  
bang\_govxleg = clean\_bang\_final %>% filter(employment\_sector == "Government/Legal")  
bang\_art = clean\_bang\_final %>% filter(employment\_sector == "Advertising/Media/Art/Culture")  
bang\_rcw = clean\_bang\_final %>% filter(employment\_sector == "Real Estate/Construction/Waste")  
bang\_nr = clean\_bang\_final %>% filter(employment\_sector == "Natural Resource/Energy")  
bang\_manu = clean\_bang\_final %>% filter(employment\_sector == "Manufacturing/Trade")  
  
  
unclean\_bang\_1x = finalized\_bang %>% filter(membership == '1x')  
unclean\_bang\_2x = finalized\_bang %>% filter(membership == '2x')  
unclean\_bang\_3x = finalized\_bang %>% filter(membership == '3x')  
unclean\_bang\_4x = finalized\_bang %>% filter(membership == '4x')  
unclean\_bang\_unlim = finalized\_bang %>% filter(membership == 'unlimited')  
unclean\_bang\_group = finalized\_bang %>% filter(membership == 'group')  
unclean\_bang\_distance = finalized\_bang %>% filter(membership == 'distance')  
bang\_1x = clean\_bang\_final %>% filter(membership == '1x')  
bang\_2x = clean\_bang\_final %>% filter(membership == '2x')  
bang\_3x = clean\_bang\_final %>% filter(membership == '3x')  
bang\_4x = clean\_bang\_final %>% filter(membership == '4x')  
bang\_unlim = clean\_bang\_final %>% filter(membership == 'unlimited')  
bang\_group = clean\_bang\_final %>% filter(membership == 'group')  
bang\_distance = clean\_bang\_final %>% filter(membership == 'distance')  
  
  
unclean\_bang\_retain\_3m = finalized\_bang %>% filter(retention\_3m == 'yes')  
unclean\_bang\_not\_retain\_3m = finalized\_bang %>% filter(retention\_3m == 'no')  
unclean\_bang\_retain\_6m = finalized\_bang %>% filter(retention\_6m == 'yes')  
unclean\_bang\_not\_retain\_6m = finalized\_bang %>% filter(retention\_6m == 'no')  
unclean\_bang\_retain\_12m = finalized\_bang %>% filter(retention\_12m == 'yes')  
unclean\_bang\_not\_retain\_12m = finalized\_bang %>% filter(retention\_12m == 'no')  
clean\_bang\_retain\_3m = clean\_bang\_final %>% filter(retention\_3m == 'yes')  
clean\_bang\_not\_retain\_3m = clean\_bang\_final %>% filter(retention\_3m == 'no')  
clean\_bang\_retain\_6m = clean\_bang\_final %>% filter(retention\_6m == 'yes')  
clean\_bang\_not\_retain\_6m = clean\_bang\_final %>% filter(retention\_6m == 'no')  
clean\_bang\_retain\_12m = clean\_bang\_final %>% filter(retention\_12m == 'yes')  
clean\_bang\_not\_retain\_12m = clean\_bang\_final %>% filter(retention\_12m == 'no')  
  
  
unclean\_bang\_job = finalized\_bang %>% filter(reason\_to\_leave == "loss of employment")  
unclean\_bang\_covid = finalized\_bang %>% filter(reason\_to\_leave == "pandemic/global crisis")  
unclean\_bang\_oth = finalized\_bang %>% filter(reason\_to\_leave == "other")  
unclean\_bang\_cost = finalized\_bang %>% filter(reason\_to\_leave == "financial")  
unclean\_bang\_health = finalized\_bang %>% filter(reason\_to\_leave == "medical/health-related")  
unclean\_bang\_move = finalized\_bang %>% filter(reason\_to\_leave == "moving away")  
unclean\_bang\_interest = finalized\_bang %>% filter(reason\_to\_leave == "other fitness interest")  
unclean\_bang\_ghost = finalized\_bang %>% filter(reason\_to\_leave == "ghosted us")  
unclean\_bang\_time = finalized\_bang %>% filter(reason\_to\_leave == "time-based arrangement")  
unclean\_bang\_dislike = finalized\_bang %>% filter(reason\_to\_leave == "noted displeasure with Bang")  
unclean\_bang\_access = finalized\_bang %>% filter(reason\_to\_leave == "lacking accessibility/availability")  
bang\_job = clean\_bang\_final %>% filter(reason\_to\_leave == "loss of employment")  
bang\_covid = clean\_bang\_final %>% filter(reason\_to\_leave == "pandemic/global crisis")  
bang\_oth = clean\_bang\_final %>% filter(reason\_to\_leave == "other")  
bang\_cost = clean\_bang\_final %>% filter(reason\_to\_leave == "financial")  
bang\_health = clean\_bang\_final %>% filter(reason\_to\_leave == "medical/health-related")  
bang\_move = clean\_bang\_final %>% filter(reason\_to\_leave == "moving away")  
bang\_interest = clean\_bang\_final %>% filter(reason\_to\_leave == "other fitness interest")  
bang\_ghost = clean\_bang\_final %>% filter(reason\_to\_leave == "ghosted us")  
bang\_time = clean\_bang\_final %>% filter(reason\_to\_leave == "time-based arrangement")  
bang\_dislike = clean\_bang\_final %>% filter(reason\_to\_leave == "noted displeasure with Bang")  
bang\_access = clean\_bang\_final %>% filter(reason\_to\_leave == "lacking accessibility/availability")  
  
bang\_unexpxunavoid = clean\_bang\_final %>% filter(churn\_type == "unexpected + unavoidable")  
bang\_expxunavoid = clean\_bang\_final %>% filter(churn\_type == "expected + unavoidable")  
bang\_unexpxavoid = clean\_bang\_final %>% filter(churn\_type == "unexpected + avoidable")  
bang\_expxavoid = clean\_bang\_final %>% filter(churn\_type == "expected + avoidable")  
bang\_avoid = clean\_bang\_final %>% filter(churn\_type == "unexpected + avoidable" | churn\_type == "expected + avoidable")  
bang\_unavoid = clean\_bang\_final %>% filter(churn\_type == "expected + unavoidable" | churn\_type == "unexpected + unavoidable")  
bang\_exp = clean\_bang\_final %>% filter(churn\_type == "expected + unavoidable" | churn\_type == "expected + avoidable")  
bang\_unexp = clean\_bang\_final %>% filter(churn\_type == "unexpected + avoidable" | churn\_type == "unexpected + unavoidable")  
  
bang\_new\_01.18 = clean\_bang\_final %>% filter(status.01.18 == "new/returning")  
bang\_current\_01.18 = clean\_bang\_final %>% filter(status.01.18 == "current")  
bang\_leave\_01.18 = clean\_bang\_final %>% filter(status.01.18 == "leaving")  
bang\_member\_01.18 = clean\_bang\_final %>% filter(status.01.18 != 'no')  
bang\_new\_02.18 = clean\_bang\_final %>% filter(status.02.18 == "new/returning")  
bang\_current\_02.18 = clean\_bang\_final %>% filter(status.02.18 == "current")  
bang\_leave\_02.18 = clean\_bang\_final %>% filter(status.02.18 == "leaving")  
bang\_member\_02.18 = clean\_bang\_final %>% filter(status.02.18 != 'no')  
bang\_new\_03.18 = clean\_bang\_final %>% filter(status.03.18 == "new/returning")  
bang\_current\_03.18 = clean\_bang\_final %>% filter(status.03.18 == "current")  
bang\_leave\_03.18 = clean\_bang\_final %>% filter(status.03.18 == "leaving")  
bang\_member\_03.18 = clean\_bang\_final %>% filter(status.03.18 != 'no')  
bang\_new\_04.18 = clean\_bang\_final %>% filter(status.04.18 == "new/returning")  
bang\_current\_04.18 = clean\_bang\_final %>% filter(status.04.18 == "current")  
bang\_leave\_04.18 = clean\_bang\_final %>% filter(status.04.18 == "leaving")  
bang\_member\_04.18 = clean\_bang\_final %>% filter(status.04.18 != 'no')  
bang\_new\_05.18 = clean\_bang\_final %>% filter(status.05.18 == "new/returning")  
bang\_current\_05.18 = clean\_bang\_final %>% filter(status.05.18 == "current")  
bang\_leave\_05.18 = clean\_bang\_final %>% filter(status.05.18 == "leaving")  
bang\_member\_05.18 = clean\_bang\_final %>% filter(status.05.18 != 'no')  
bang\_new\_06.18 = clean\_bang\_final %>% filter(status.06.18 == "new/returning")  
bang\_current\_06.18 = clean\_bang\_final %>% filter(status.06.18 == "current")  
bang\_leave\_06.18 = clean\_bang\_final %>% filter(status.06.18 == "leaving")  
bang\_member\_06.18 = clean\_bang\_final %>% filter(status.06.18 != 'no')  
bang\_new\_07.18 = clean\_bang\_final %>% filter(status.07.18 == "new/returning")  
bang\_current\_07.18 = clean\_bang\_final %>% filter(status.07.18 == "current")  
bang\_leave\_07.18 = clean\_bang\_final %>% filter(status.07.18 == "leaving")  
bang\_member\_07.18 = clean\_bang\_final %>% filter(status.07.18 != 'no')  
bang\_new\_08.18 = clean\_bang\_final %>% filter(status.08.18 == "new/returning")  
bang\_current\_08.18 = clean\_bang\_final %>% filter(status.08.18 == "current")  
bang\_leave\_08.18 = clean\_bang\_final %>% filter(status.08.18 == "leaving")  
bang\_member\_08.18 = clean\_bang\_final %>% filter(status.08.18 != 'no')  
bang\_new\_09.18 = clean\_bang\_final %>% filter(status.09.18 == "new/returning")  
bang\_current\_09.18 = clean\_bang\_final %>% filter(status.09.18 == "current")  
bang\_leave\_09.18 = clean\_bang\_final %>% filter(status.09.18 == "leaving")  
bang\_member\_09.18 = clean\_bang\_final %>% filter(status.09.18 != 'no')  
bang\_new\_10.18 = clean\_bang\_final %>% filter(status.10.18 == "new/returning")  
bang\_current\_10.18 = clean\_bang\_final %>% filter(status.10.18 == "current")  
bang\_leave\_10.18 = clean\_bang\_final %>% filter(status.10.18 == "leaving")  
bang\_member\_10.18 = clean\_bang\_final %>% filter(status.10.18 != 'no')  
bang\_new\_11.18 = clean\_bang\_final %>% filter(status.11.18 == "new/returning")  
bang\_current\_11.18 = clean\_bang\_final %>% filter(status.11.18 == "current")  
bang\_leave\_11.18 = clean\_bang\_final %>% filter(status.11.18 == "leaving")  
bang\_member\_11.18 = clean\_bang\_final %>% filter(status.11.18 != 'no')  
bang\_new\_12.18 = clean\_bang\_final %>% filter(status.12.18 == "new/returning")  
bang\_current\_12.18 = clean\_bang\_final %>% filter(status.12.18 == "current")  
bang\_leave\_12.18 = clean\_bang\_final %>% filter(status.12.18 == "leaving")  
bang\_member\_12.18 = clean\_bang\_final %>% filter(status.12.18 != 'no')  
bang\_new\_01.19 = clean\_bang\_final %>% filter(status.01.19 == "new/returning")  
bang\_current\_01.19 = clean\_bang\_final %>% filter(status.01.19 == "current")  
bang\_leave\_01.19 = clean\_bang\_final %>% filter(status.01.19 == "leaving")  
bang\_member\_01.19 = clean\_bang\_final %>% filter(status.01.19 != 'no')  
bang\_new\_02.19 = clean\_bang\_final %>% filter(status.02.19 == "new/returning")  
bang\_current\_02.19 = clean\_bang\_final %>% filter(status.02.19 == "current")  
bang\_leave\_02.19 = clean\_bang\_final %>% filter(status.02.19 == "leaving")  
bang\_member\_02.19 = clean\_bang\_final %>% filter(status.02.19 != 'no')  
bang\_new\_03.19 = clean\_bang\_final %>% filter(status.03.19 == "new/returning")  
bang\_current\_03.19 = clean\_bang\_final %>% filter(status.03.19 == "current")  
bang\_leave\_03.19 = clean\_bang\_final %>% filter(status.03.19 == "leaving")  
bang\_member\_03.19 = clean\_bang\_final %>% filter(status.03.19 != 'no')  
bang\_new\_04.19 = clean\_bang\_final %>% filter(status.04.19 == "new/returning")  
bang\_current\_04.19 = clean\_bang\_final %>% filter(status.04.19 == "current")  
bang\_leave\_04.19 = clean\_bang\_final %>% filter(status.04.19 == "leaving")  
bang\_member\_04.19 = clean\_bang\_final %>% filter(status.04.19 != 'no')  
bang\_new\_05.19 = clean\_bang\_final %>% filter(status.05.19 == "new/returning")  
bang\_current\_05.19 = clean\_bang\_final %>% filter(status.05.19 == "current")  
bang\_leave\_05.19 = clean\_bang\_final %>% filter(status.05.19 == "leaving")  
bang\_member\_05.19 = clean\_bang\_final %>% filter(status.05.19 != 'no')  
bang\_new\_06.19 = clean\_bang\_final %>% filter(status.06.19 == "new/returning")  
bang\_current\_06.19 = clean\_bang\_final %>% filter(status.06.19 == "current")  
bang\_leave\_06.19 = clean\_bang\_final %>% filter(status.06.19 == "leaving")  
bang\_member\_06.19 = clean\_bang\_final %>% filter(status.06.19 != 'no')  
bang\_new\_07.19 = clean\_bang\_final %>% filter(status.07.19 == "new/returning")  
bang\_current\_07.19 = clean\_bang\_final %>% filter(status.07.19 == "current")  
bang\_leave\_07.19 = clean\_bang\_final %>% filter(status.07.19 == "leaving")  
bang\_member\_07.19 = clean\_bang\_final %>% filter(status.07.19 != 'no')  
bang\_new\_08.19 = clean\_bang\_final %>% filter(status.08.19 == "new/returning")  
bang\_current\_08.19 = clean\_bang\_final %>% filter(status.08.19 == "current")  
bang\_leave\_08.19 = clean\_bang\_final %>% filter(status.08.19 == "leaving")  
bang\_member\_08.19 = clean\_bang\_final %>% filter(status.08.19 != 'no')  
bang\_new\_09.19 = clean\_bang\_final %>% filter(status.09.19 == "new/returning")  
bang\_current\_09.19 = clean\_bang\_final %>% filter(status.09.19 == "current")  
bang\_leave\_09.19 = clean\_bang\_final %>% filter(status.09.19 == "leaving")  
bang\_member\_09.19 = clean\_bang\_final %>% filter(status.09.19 != 'no')  
bang\_new\_10.19 = clean\_bang\_final %>% filter(status.10.19 == "new/returning")  
bang\_current\_10.19 = clean\_bang\_final %>% filter(status.10.19 == "current")  
bang\_leave\_10.19 = clean\_bang\_final %>% filter(status.10.19 == "leaving")  
bang\_member\_10.19 = clean\_bang\_final %>% filter(status.10.19 != 'no')  
bang\_new\_11.19 = clean\_bang\_final %>% filter(status.11.19 == "new/returning")  
bang\_current\_11.19 = clean\_bang\_final %>% filter(status.11.19 == "current")  
bang\_leave\_11.19 = clean\_bang\_final %>% filter(status.11.19 == "leaving")  
bang\_member\_11.19 = clean\_bang\_final %>% filter(status.11.19 != 'no')  
bang\_new\_12.19 = clean\_bang\_final %>% filter(status.12.19 == "new/returning")  
bang\_current\_12.19 = clean\_bang\_final %>% filter(status.12.19 == "current")  
bang\_leave\_12.19 = clean\_bang\_final %>% filter(status.12.19 == "leaving")  
bang\_member\_12.19 = clean\_bang\_final %>% filter(status.12.19 != 'no')  
bang\_new\_01.20 = clean\_bang\_final %>% filter(status.01.20 == "new/returning")  
bang\_current\_01.20 = clean\_bang\_final %>% filter(status.01.20 == "current")  
bang\_leave\_01.20 = clean\_bang\_final %>% filter(status.01.20 == "leaving")  
bang\_member\_01.20 = clean\_bang\_final %>% filter(status.01.20 != 'no')  
bang\_new\_02.20 = clean\_bang\_final %>% filter(status.02.20 == "new/returning")  
bang\_current\_02.20 = clean\_bang\_final %>% filter(status.02.20 == "current")  
bang\_leave\_02.20 = clean\_bang\_final %>% filter(status.02.20 == "leaving")  
bang\_member\_02.20 = clean\_bang\_final %>% filter(status.02.20 != 'no')  
bang\_new\_03.20 = clean\_bang\_final %>% filter(status.03.20 == "new/returning")  
bang\_current\_03.20 = clean\_bang\_final %>% filter(status.03.20 == "current")  
bang\_leave\_03.20 = clean\_bang\_final %>% filter(status.03.20 == "leaving")  
bang\_member\_03.20 = clean\_bang\_final %>% filter(status.03.20 != 'no')  
bang\_new\_04.20 = clean\_bang\_final %>% filter(status.04.20 == "new/returning")  
bang\_current\_04.20 = clean\_bang\_final %>% filter(status.04.20 == "current")  
bang\_leave\_04.20 = clean\_bang\_final %>% filter(status.04.20 == "leaving")  
bang\_member\_04.20 = clean\_bang\_final %>% filter(status.04.20 != 'no')  
bang\_new\_05.20 = clean\_bang\_final %>% filter(status.05.20 == "new/returning")  
bang\_current\_05.20 = clean\_bang\_final %>% filter(status.05.20 == "current")  
bang\_leave\_05.20 = clean\_bang\_final %>% filter(status.05.20 == "leaving")  
bang\_member\_05.20 = clean\_bang\_final %>% filter(status.05.20 != 'no')  
bang\_new\_06.20 = clean\_bang\_final %>% filter(status.06.20 == "new/returning")  
bang\_current\_06.20 = clean\_bang\_final %>% filter(status.06.20 == "current")  
bang\_leave\_06.20 = clean\_bang\_final %>% filter(status.06.20 == "leaving")  
bang\_member\_06.20 = clean\_bang\_final %>% filter(status.06.20 != 'no')  
bang\_new\_07.20 = clean\_bang\_final %>% filter(status.07.20 == "new/returning")  
bang\_current\_07.20 = clean\_bang\_final %>% filter(status.07.20 == "current")  
bang\_leave\_07.20 = clean\_bang\_final %>% filter(status.07.20 == "leaving")  
bang\_member\_07.20 = clean\_bang\_final %>% filter(status.07.20 != 'no')  
bang\_new\_08.20 = clean\_bang\_final %>% filter(status.08.20 == "new/returning")  
bang\_current\_08.20 = clean\_bang\_final %>% filter(status.08.20 == "current")  
bang\_leave\_08.20 = clean\_bang\_final %>% filter(status.08.20 == "leaving")  
bang\_member\_08.20 = clean\_bang\_final %>% filter(status.08.20 != 'no')  
bang\_new\_09.20 = clean\_bang\_final %>% filter(status.09.20 == "new/returning")  
bang\_current\_09.20 = clean\_bang\_final %>% filter(status.09.20 == "current")  
bang\_leave\_09.20 = clean\_bang\_final %>% filter(status.09.20 == "leaving")  
bang\_member\_09.20 = clean\_bang\_final %>% filter(status.09.20 != 'no')  
bang\_new\_10.20 = clean\_bang\_final %>% filter(status.10.20 == "new/returning")  
bang\_current\_10.20 = clean\_bang\_final %>% filter(status.10.20 == "current")  
bang\_leave\_10.20 = clean\_bang\_final %>% filter(status.10.20 == "leaving")  
bang\_member\_10.20 = clean\_bang\_final %>% filter(status.10.20 != 'no')  
  
  
unclean\_bang\_new\_01.18 = finalized\_bang %>% filter(status.01.18 == "new/returning")  
unclean\_bang\_current\_01.18 = finalized\_bang %>% filter(status.01.18 == "current")  
unclean\_bang\_leave\_01.18 = finalized\_bang %>% filter(status.01.18 == "leaving")  
unclean\_bang\_member\_01.18 = finalized\_bang %>% filter(status.01.18 != 'no')  
unclean\_bang\_new\_02.18 = finalized\_bang %>% filter(status.02.18 == "new/returning")  
unclean\_bang\_current\_02.18 = finalized\_bang %>% filter(status.02.18 == "current")  
unclean\_bang\_leave\_02.18 = finalized\_bang %>% filter(status.02.18 == "leaving")  
unclean\_bang\_member\_02.18 = finalized\_bang %>% filter(status.02.18 != 'no')  
unclean\_bang\_new\_03.18 = finalized\_bang %>% filter(status.03.18 == "new/returning")  
unclean\_bang\_current\_03.18 = finalized\_bang %>% filter(status.03.18 == "current")  
unclean\_bang\_leave\_03.18 = finalized\_bang %>% filter(status.03.18 == "leaving")  
unclean\_bang\_member\_03.18 = finalized\_bang %>% filter(status.03.18 != 'no')  
unclean\_bang\_new\_04.18 = finalized\_bang %>% filter(status.04.18 == "new/returning")  
unclean\_bang\_current\_04.18 = finalized\_bang %>% filter(status.04.18 == "current")  
unclean\_bang\_leave\_04.18 = finalized\_bang %>% filter(status.04.18 == "leaving")  
unclean\_bang\_member\_04.18 = finalized\_bang %>% filter(status.04.18 != 'no')  
unclean\_bang\_new\_05.18 = finalized\_bang %>% filter(status.05.18 == "new/returning")  
unclean\_bang\_current\_05.18 = finalized\_bang %>% filter(status.05.18 == "current")  
unclean\_bang\_leave\_05.18 = finalized\_bang %>% filter(status.05.18 == "leaving")  
unclean\_bang\_member\_05.18 = finalized\_bang %>% filter(status.05.18 != 'no')  
unclean\_bang\_new\_06.18 = finalized\_bang %>% filter(status.06.18 == "new/returning")  
unclean\_bang\_current\_06.18 = finalized\_bang %>% filter(status.06.18 == "current")  
unclean\_bang\_leave\_06.18 = finalized\_bang %>% filter(status.06.18 == "leaving")  
unclean\_bang\_member\_06.18 = finalized\_bang %>% filter(status.06.18 != 'no')  
unclean\_bang\_new\_07.18 = finalized\_bang %>% filter(status.07.18 == "new/returning")  
unclean\_bang\_current\_07.18 = finalized\_bang %>% filter(status.07.18 == "current")  
unclean\_bang\_leave\_07.18 = finalized\_bang %>% filter(status.07.18 == "leaving")  
unclean\_bang\_member\_07.18 = finalized\_bang %>% filter(status.07.18 != 'no')  
unclean\_bang\_new\_08.18 = finalized\_bang %>% filter(status.08.18 == "new/returning")  
unclean\_bang\_current\_08.18 = finalized\_bang %>% filter(status.08.18 == "current")  
unclean\_bang\_leave\_08.18 = finalized\_bang %>% filter(status.08.18 == "leaving")  
unclean\_bang\_member\_08.18 = finalized\_bang %>% filter(status.08.18 != 'no')  
unclean\_bang\_new\_09.18 = finalized\_bang %>% filter(status.09.18 == "new/returning")  
unclean\_bang\_current\_09.18 = finalized\_bang %>% filter(status.09.18 == "current")  
unclean\_bang\_leave\_09.18 = finalized\_bang %>% filter(status.09.18 == "leaving")  
unclean\_bang\_member\_09.18 = finalized\_bang %>% filter(status.09.18 != 'no')  
unclean\_bang\_new\_10.18 = finalized\_bang %>% filter(status.10.18 == "new/returning")  
unclean\_bang\_current\_10.18 = finalized\_bang %>% filter(status.10.18 == "current")  
unclean\_bang\_leave\_10.18 = finalized\_bang %>% filter(status.10.18 == "leaving")  
unclean\_bang\_member\_10.18 = finalized\_bang %>% filter(status.10.18 != 'no')  
unclean\_bang\_new\_11.18 = finalized\_bang %>% filter(status.11.18 == "new/returning")  
unclean\_bang\_current\_11.18 = finalized\_bang %>% filter(status.11.18 == "current")  
unclean\_bang\_leave\_11.18 = finalized\_bang %>% filter(status.11.18 == "leaving")  
unclean\_bang\_member\_11.18 = finalized\_bang %>% filter(status.11.18 != 'no')  
unclean\_bang\_new\_12.18 = finalized\_bang %>% filter(status.12.18 == "new/returning")  
unclean\_bang\_current\_12.18 = finalized\_bang %>% filter(status.12.18 == "current")  
unclean\_bang\_leave\_12.18 = finalized\_bang %>% filter(status.12.18 == "leaving")  
unclean\_bang\_member\_12.18 = finalized\_bang %>% filter(status.12.18 != 'no')  
unclean\_bang\_new\_01.19 = finalized\_bang %>% filter(status.01.19 == "new/returning")  
unclean\_bang\_current\_01.19 = finalized\_bang %>% filter(status.01.19 == "current")  
unclean\_bang\_leave\_01.19 = finalized\_bang %>% filter(status.01.19 == "leaving")  
unclean\_bang\_member\_01.19 = finalized\_bang %>% filter(status.01.19 != 'no')  
unclean\_bang\_new\_02.19 = finalized\_bang %>% filter(status.02.19 == "new/returning")  
unclean\_bang\_current\_02.19 = finalized\_bang %>% filter(status.02.19 == "current")  
unclean\_bang\_leave\_02.19 = finalized\_bang %>% filter(status.02.19 == "leaving")  
unclean\_bang\_member\_02.19 = finalized\_bang %>% filter(status.02.19 != 'no')  
unclean\_bang\_new\_03.19 = finalized\_bang %>% filter(status.03.19 == "new/returning")  
unclean\_bang\_current\_03.19 = finalized\_bang %>% filter(status.03.19 == "current")  
unclean\_bang\_leave\_03.19 = finalized\_bang %>% filter(status.03.19 == "leaving")  
unclean\_bang\_member\_03.19 = finalized\_bang %>% filter(status.03.19 != 'no')  
unclean\_bang\_new\_04.19 = finalized\_bang %>% filter(status.04.19 == "new/returning")  
unclean\_bang\_current\_04.19 = finalized\_bang %>% filter(status.04.19 == "current")  
unclean\_bang\_leave\_04.19 = finalized\_bang %>% filter(status.04.19 == "leaving")  
unclean\_bang\_member\_04.19 = finalized\_bang %>% filter(status.04.19 != 'no')  
unclean\_bang\_new\_05.19 = finalized\_bang %>% filter(status.05.19 == "new/returning")  
unclean\_bang\_current\_05.19 = finalized\_bang %>% filter(status.05.19 == "current")  
unclean\_bang\_leave\_05.19 = finalized\_bang %>% filter(status.05.19 == "leaving")  
unclean\_bang\_member\_05.19 = finalized\_bang %>% filter(status.05.19 != 'no')  
unclean\_bang\_new\_06.19 = finalized\_bang %>% filter(status.06.19 == "new/returning")  
unclean\_bang\_current\_06.19 = finalized\_bang %>% filter(status.06.19 == "current")  
unclean\_bang\_leave\_06.19 = finalized\_bang %>% filter(status.06.19 == "leaving")  
unclean\_bang\_member\_06.19 = finalized\_bang %>% filter(status.06.19 != 'no')  
unclean\_bang\_new\_07.19 = finalized\_bang %>% filter(status.07.19 == "new/returning")  
unclean\_bang\_current\_07.19 = finalized\_bang %>% filter(status.07.19 == "current")  
unclean\_bang\_leave\_07.19 = finalized\_bang %>% filter(status.07.19 == "leaving")  
unclean\_bang\_member\_07.19 = finalized\_bang %>% filter(status.07.19 != 'no')  
unclean\_bang\_new\_08.19 = finalized\_bang %>% filter(status.08.19 == "new/returning")  
unclean\_bang\_current\_08.19 = finalized\_bang %>% filter(status.08.19 == "current")  
unclean\_bang\_leave\_08.19 = finalized\_bang %>% filter(status.08.19 == "leaving")  
unclean\_bang\_member\_08.19 = finalized\_bang %>% filter(status.08.19 != 'no')  
unclean\_bang\_new\_09.19 = finalized\_bang %>% filter(status.09.19 == "new/returning")  
unclean\_bang\_current\_09.19 = finalized\_bang %>% filter(status.09.19 == "current")  
unclean\_bang\_leave\_09.19 = finalized\_bang %>% filter(status.09.19 == "leaving")  
unclean\_bang\_member\_09.19 = finalized\_bang %>% filter(status.09.19 != 'no')  
unclean\_bang\_new\_10.19 = finalized\_bang %>% filter(status.10.19 == "new/returning")  
unclean\_bang\_current\_10.19 = finalized\_bang %>% filter(status.10.19 == "current")  
unclean\_bang\_leave\_10.19 = finalized\_bang %>% filter(status.10.19 == "leaving")  
unclean\_bang\_member\_10.19 = finalized\_bang %>% filter(status.10.19 != 'no')  
unclean\_bang\_new\_11.19 = finalized\_bang %>% filter(status.11.19 == "new/returning")  
unclean\_bang\_current\_11.19 = finalized\_bang %>% filter(status.11.19 == "current")  
unclean\_bang\_leave\_11.19 = finalized\_bang %>% filter(status.11.19 == "leaving")  
unclean\_bang\_member\_11.19 = finalized\_bang %>% filter(status.11.19 != 'no')  
unclean\_bang\_new\_12.19 = finalized\_bang %>% filter(status.12.19 == "new/returning")  
unclean\_bang\_current\_12.19 = finalized\_bang %>% filter(status.12.19 == "current")  
unclean\_bang\_leave\_12.19 = finalized\_bang %>% filter(status.12.19 == "leaving")  
unclean\_bang\_member\_12.19 = finalized\_bang %>% filter(status.12.19 != 'no')  
unclean\_bang\_new\_01.20 = finalized\_bang %>% filter(status.01.20 == "new/returning")  
unclean\_bang\_current\_01.20 = finalized\_bang %>% filter(status.01.20 == "current")  
unclean\_bang\_leave\_01.20 = finalized\_bang %>% filter(status.01.20 == "leaving")  
unclean\_bang\_member\_01.20 = finalized\_bang %>% filter(status.01.20 != 'no')  
unclean\_bang\_new\_02.20 = finalized\_bang %>% filter(status.02.20 == "new/returning")  
unclean\_bang\_current\_02.20 = finalized\_bang %>% filter(status.02.20 == "current")  
unclean\_bang\_leave\_02.20 = finalized\_bang %>% filter(status.02.20 == "leaving")  
unclean\_bang\_member\_02.20 = finalized\_bang %>% filter(status.02.20 != 'no')  
unclean\_bang\_new\_03.20 = finalized\_bang %>% filter(status.03.20 == "new/returning")  
unclean\_bang\_current\_03.20 = finalized\_bang %>% filter(status.03.20 == "current")  
unclean\_bang\_leave\_03.20 = finalized\_bang %>% filter(status.03.20 == "leaving")  
unclean\_bang\_member\_03.20 = finalized\_bang %>% filter(status.03.20 != 'no')  
unclean\_bang\_new\_04.20 = finalized\_bang %>% filter(status.04.20 == "new/returning")  
unclean\_bang\_current\_04.20 = finalized\_bang %>% filter(status.04.20 == "current")  
unclean\_bang\_leave\_04.20 = finalized\_bang %>% filter(status.04.20 == "leaving")  
unclean\_bang\_member\_04.20 = finalized\_bang %>% filter(status.04.20 != 'no')  
unclean\_bang\_new\_05.20 = finalized\_bang %>% filter(status.05.20 == "new/returning")  
unclean\_bang\_current\_05.20 = finalized\_bang %>% filter(status.05.20 == "current")  
unclean\_bang\_leave\_05.20 = finalized\_bang %>% filter(status.05.20 == "leaving")  
unclean\_bang\_member\_05.20 = finalized\_bang %>% filter(status.05.20 != 'no')  
unclean\_bang\_new\_06.20 = finalized\_bang %>% filter(status.06.20 == "new/returning")  
unclean\_bang\_current\_06.20 = finalized\_bang %>% filter(status.06.20 == "current")  
unclean\_bang\_leave\_06.20 = finalized\_bang %>% filter(status.06.20 == "leaving")  
unclean\_bang\_member\_06.20 = finalized\_bang %>% filter(status.06.20 != 'no')  
unclean\_bang\_new\_07.20 = finalized\_bang %>% filter(status.07.20 == "new/returning")  
unclean\_bang\_current\_07.20 = finalized\_bang %>% filter(status.07.20 == "current")  
unclean\_bang\_leave\_07.20 = finalized\_bang %>% filter(status.07.20 == "leaving")  
unclean\_bang\_member\_07.20 = finalized\_bang %>% filter(status.07.20 != 'no')  
unclean\_bang\_new\_08.20 = finalized\_bang %>% filter(status.08.20 == "new/returning")  
unclean\_bang\_current\_08.20 = finalized\_bang %>% filter(status.08.20 == "current")  
unclean\_bang\_leave\_08.20 = finalized\_bang %>% filter(status.08.20 == "leaving")  
unclean\_bang\_member\_08.20 = finalized\_bang %>% filter(status.08.20 != 'no')  
unclean\_bang\_new\_09.20 = finalized\_bang %>% filter(status.09.20 == "new/returning")  
unclean\_bang\_current\_09.20 = finalized\_bang %>% filter(status.09.20 == "current")  
unclean\_bang\_leave\_09.20 = finalized\_bang %>% filter(status.09.20 == "leaving")  
unclean\_bang\_member\_09.20 = finalized\_bang %>% filter(status.09.20 != 'no')  
unclean\_bang\_new\_10.20 = finalized\_bang %>% filter(status.10.20 == "new/returning")  
unclean\_bang\_current\_10.20 = finalized\_bang %>% filter(status.10.20 == "current")  
unclean\_bang\_leave\_10.20 = finalized\_bang %>% filter(status.10.20 == "leaving")  
unclean\_bang\_member\_10.20 = finalized\_bang %>% filter(status.10.20 != 'no')  
  
unclean\_bang\_less.than.100 = finalized\_bang %>% filter(monthly\_rate\_group == '0-99.99')  
unclean\_bang\_100\_to\_150 = finalized\_bang %>% filter(monthly\_rate\_group == '100-149.99')  
unclean\_bang\_150\_to\_200 = finalized\_bang %>% filter(monthly\_rate\_group == '150-199.99')  
unclean\_bang\_200\_to\_250 = finalized\_bang %>% filter(monthly\_rate\_group == '200-249.99')  
unclean\_bang\_250\_to\_300 = finalized\_bang %>% filter(monthly\_rate\_group == '250-299.99')  
unclean\_bang\_300\_to\_350 = finalized\_bang %>% filter(monthly\_rate\_group == '300-349.99')  
unclean\_bang\_350\_to\_400 = finalized\_bang %>% filter(monthly\_rate\_group == '350-399.99')  
unclean\_bang\_400\_to\_450 = finalized\_bang %>% filter(monthly\_rate\_group == '400-449.99')  
unclean\_bang\_450\_to\_500 = finalized\_bang %>% filter(monthly\_rate\_group == '450-499.99')  
unclean\_bang\_500\_to\_550 = finalized\_bang %>% filter(monthly\_rate\_group == '500-549.99')  
unclean\_bang\_550\_to\_600 = finalized\_bang %>% filter(monthly\_rate\_group == '550-599.99')  
unclean\_bang\_more.than.600 = finalized\_bang %>% filter(monthly\_rate\_group == '600+')  
  
clean\_bang\_less.than.100 = clean\_bang\_final %>% filter(monthly\_rate\_group == '0-99.99')  
clean\_bang\_100\_to\_150 = clean\_bang\_final %>% filter(monthly\_rate\_group == '100-149.99')  
clean\_bang\_150\_to\_200 = clean\_bang\_final %>% filter(monthly\_rate\_group == '150-199.99')  
clean\_bang\_200\_to\_250 = clean\_bang\_final %>% filter(monthly\_rate\_group == '200-249.99')  
clean\_bang\_250\_to\_300 = clean\_bang\_final %>% filter(monthly\_rate\_group == '250-299.99')  
clean\_bang\_300\_to\_350 = clean\_bang\_final %>% filter(monthly\_rate\_group == '300-349.99')  
clean\_bang\_350\_to\_400 = clean\_bang\_final %>% filter(monthly\_rate\_group == '350-399.99')  
clean\_bang\_400\_to\_450 = clean\_bang\_final %>% filter(monthly\_rate\_group == '400-449.99')  
clean\_bang\_450\_to\_500 = clean\_bang\_final %>% filter(monthly\_rate\_group == '450-499.99')  
clean\_bang\_500\_to\_550 = clean\_bang\_final %>% filter(monthly\_rate\_group == '500-549.99')  
clean\_bang\_550\_to\_600 = clean\_bang\_final %>% filter(monthly\_rate\_group == '550-599.99')  
clean\_bang\_more.than.600 = clean\_bang\_final %>% filter(monthly\_rate\_group == '600+')  
  
unclean\_bang\_monthly\_0.to.150 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '0-149.99')  
unclean\_bang\_monthly\_150.to.200 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '150-199.99')  
unclean\_bang\_monthly\_200.to.300 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '200-299.99')  
unclean\_bang\_monthly\_300.to.350 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '300-349.99')  
unclean\_bang\_monthly\_350.to.400 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '350-399.99')  
unclean\_bang\_monthly\_400.to.450 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '400-449.99')  
unclean\_bang\_monthly\_450.to.500 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '450-499.99')  
unclean\_bang\_monthly\_more.than.500 = finalized\_bang %>% filter(monthly\_grouping\_ver.1 == '500+')  
  
clean\_bang\_monthly\_0.to.150 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '0-149.99')  
clean\_bang\_monthly\_150.to.200 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '150-199.99')  
clean\_bang\_monthly\_200.to.300 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '200-299.99')  
clean\_bang\_monthly\_300.to.350 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '300-349.99')  
clean\_bang\_monthly\_350.to.400 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '350-399.99')  
clean\_bang\_monthly\_400.to.450 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '400-449.99')  
clean\_bang\_monthly\_450.to.500 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '450-499.99')  
clean\_bang\_monthly\_more.than.500 = clean\_bang\_final %>% filter(monthly\_grouping\_ver.1 == '500+')  
  
unclean\_bang\_monthly\_cutoff.yes = finalized\_bang %>% filter(monthly\_rate\_cutoff == "yes")  
unclean\_bang\_monthly\_cutoff.no = finalized\_bang %>% filter(monthly\_rate\_cutoff == "no")  
clean\_bang\_monthly\_cutoff.yes = clean\_bang\_final %>% filter(monthly\_rate\_cutoff == "yes")  
clean\_bang\_monthly\_cutoff.no = clean\_bang\_final %>% filter(monthly\_rate\_cutoff == "no")  
  
  
  
unclean\_bang\_less.than.50 = finalized\_bang %>% filter(attendance\_rate\_group == '0-9.99' | attendance\_rate\_group == '10-19.99' | attendance\_rate\_group == '20-29.99' | attendance\_rate\_group == '30-39.99' | attendance\_rate\_group == '40-49.99')  
unclean\_bang\_50\_to\_60 = finalized\_bang %>% filter(attendance\_rate\_group == '50-59.99')  
unclean\_bang\_60\_to\_70 = finalized\_bang %>% filter(attendance\_rate\_group == '60-69.99')  
unclean\_bang\_70\_to\_80 = finalized\_bang %>% filter(attendance\_rate\_group == '70-79.99')  
unclean\_bang\_more.than.80 = finalized\_bang %>% filter(attendance\_rate\_group == '80-89.99' | attendance\_rate\_group == "90-100")  
  
clean\_bang\_less.than.50 = clean\_bang\_final %>% filter(attendance\_rate\_group == '0-9.99' | attendance\_rate\_group == '10-19.99' | attendance\_rate\_group == '20-29.99' | attendance\_rate\_group == '30-39.99' | attendance\_rate\_group == '40-49.99')  
clean\_bang\_50\_to\_60 = clean\_bang\_final %>% filter(attendance\_rate\_group == '50-59.99')  
clean\_bang\_60\_to\_70 = clean\_bang\_final %>% filter(attendance\_rate\_group == '60-69.99')  
clean\_bang\_70\_to\_80 = clean\_bang\_final %>% filter(attendance\_rate\_group == '70-79.99')  
clean\_bang\_more.than.80 = clean\_bang\_final %>% filter(attendance\_rate\_group == '80-89.99' | attendance\_rate\_group == "90-100")  
  
unclean\_bang\_monthly\_cutoff.yes = finalized\_bang %>% filter(attendance\_rate\_cutoff == "yes")  
unclean\_bang\_monthly\_cutoff.no = finalized\_bang %>% filter(attendance\_rate\_cutoff == "no")  
clean\_bang\_monthly\_cutoff.yes = clean\_bang\_final %>% filter(attendance\_rate\_cutoff == "yes")  
clean\_bang\_monthly\_cutoff.no = clean\_bang\_final %>% filter(attendance\_rate\_cutoff == "no")  
  
  
unclean\_bang\_longer\_retention = finalized\_bang %>%   
 pivot\_longer(  
 cols = c(retention\_3m, retention\_6m, retention\_12m),   
 names\_to = "retention\_type",  
 values\_to = "retention\_status"  
 )  
clean\_bang\_longer\_retention = clean\_bang\_final %>%   
 pivot\_longer(  
 cols = c(retention\_3m, retention\_6m, retention\_12m),   
 names\_to = "retention\_type",  
 values\_to = "retention\_status"  
 )  
unclean\_bang\_longer\_retention = unclean\_bang\_longer\_retention %>%   
 mutate(  
 retention\_type = as.factor(retention\_type)  
 ) %>%   
 mutate(  
 retention\_type = factor(retention\_type, levels = c('retention\_3m', 'retention\_6m', 'retention\_12m'))  
 )  
clean\_bang\_longer\_retention = clean\_bang\_longer\_retention %>%   
 mutate(  
 retention\_type = as.factor(retention\_type)  
 ) %>%   
 mutate(  
 retention\_type = factor(retention\_type, levels = c('retention\_3m', 'retention\_6m', 'retention\_12m'))  
 )  
  
  
unclean\_bang\_longer\_attendance = finalized\_bang %>%   
 pivot\_longer(  
 cols = c(attendance\_rate, cancellation\_rate),  
 names\_to = "type",  
 values\_to = "rates"  
 )  
clean\_bang\_longer\_attendance = clean\_bang\_final %>%   
 pivot\_longer(  
 cols = c(attendance\_rate, cancellation\_rate),  
 names\_to = "type",  
 values\_to = "rates"  
 )  
  
  
unclean\_bang\_longer\_num\_emails = finalized\_bang %>%   
 pivot\_longer(  
 cols = c(num\_ticket\_billing, num\_ticket\_cx, num\_ticket\_scheduling, num\_ticket\_service),  
 names\_to = "email\_type",  
 values\_to = "email\_amount"  
 )  
clean\_bang\_longer\_num\_emails = clean\_bang\_final %>%   
 pivot\_longer(  
 cols = c(num\_ticket\_billing, num\_ticket\_cx, num\_ticket\_scheduling, num\_ticket\_service),  
 names\_to = "email\_type",  
 values\_to = "email\_amount"  
 )  
  
  
unclean\_bang\_longer\_per\_emails = finalized\_bang %>%   
 pivot\_longer(  
 cols = c(per\_ticket\_billing, per\_ticket\_cx, per\_ticket\_scheduling, per\_ticket\_service),  
 names\_to = "email\_type",  
 values\_to = "email\_percent"  
 )  
clean\_bang\_longer\_per\_emails = clean\_bang\_final %>%   
 pivot\_longer(  
 cols = c(per\_ticket\_billing, per\_ticket\_cx, per\_ticket\_scheduling, per\_ticket\_service),  
 names\_to = "email\_type",  
 values\_to = "email\_percent"  
 )  
  
  
unclean\_bang\_longer\_new\_per\_emails = finalized\_bang %>%   
 pivot\_longer(  
 cols = c(new\_per\_ticket\_cx, new\_per\_ticket\_scheduling, new\_per\_ticket\_service),  
 names\_to = "email\_type",  
 values\_to = 'email\_percent'  
 )  
clean\_bang\_longer\_new\_per\_emails = clean\_bang\_final %>%   
 pivot\_longer(  
 cols = c(new\_per\_ticket\_cx, new\_per\_ticket\_scheduling, new\_per\_ticket\_service),  
 names\_to = "email\_type",  
 values\_to = 'email\_percent'  
 )  
  
  
  
unclean\_bang\_longer\_status = finalized\_bang %>%   
 pivot\_longer(  
 cols = c(  
 status.01.18, status.02.18, status.03.18, status.04.18, status.05.18,   
 status.06.18, status.07.18, status.08.18, status.09.18, status.10.18,   
 status.11.18, status.12.18, status.01.19, status.02.19, status.03.19,   
 status.04.19, status.05.19, status.06.19, status.07.19, status.08.19,   
 status.09.19, status.10.19, status.11.19, status.12.19, status.01.20,   
 status.02.20, status.03.20, status.04.20, status.05.20, status.06.20,   
 status.07.20, status.08.20, status.09.20, status.10.20  
 ),  
 names\_to = "membership\_period",  
 values\_to = "membership\_status"  
 )  
clean\_bang\_longer\_status = clean\_bang\_final %>%   
 pivot\_longer(  
 cols = c(  
 status.01.18, status.02.18, status.03.18, status.04.18, status.05.18,   
 status.06.18, status.07.18, status.08.18, status.09.18, status.10.18,   
 status.11.18, status.12.18, status.01.19, status.02.19, status.03.19,   
 status.04.19, status.05.19, status.06.19, status.07.19, status.08.19,   
 status.09.19, status.10.19, status.11.19, status.12.19, status.01.20,   
 status.02.20, status.03.20, status.04.20, status.05.20, status.06.20,   
 status.07.20, status.08.20, status.09.20, status.10.20  
 ),  
 names\_to = "membership\_period",  
 values\_to = "membership\_status"  
 )  
  
unclean\_bang\_longer\_status = unclean\_bang\_longer\_status %>%   
 mutate(  
 membership\_period = factor(membership\_period, levels = c("status.01.18", "status.02.18","status.03.18", "status.04.18","status.05.18", "status.06.18",  
 "status.07.18", "status.08.18","status.09.18", "status.10.18","status.11.18", "status.12.18",  
 "status.01.19", "status.02.19","status.03.19", "status.04.19","status.05.19", "status.06.19",  
 "status.07.19", "status.08.19","status.09.19", "status.10.19","status.11.19", "status.12.19",  
 "status.01.20", "status.02.20","status.03.20", "status.04.20","status.05.20", "status.06.20",  
 "status.07.20", "status.08.20","status.09.20", "status.10.20"))  
 )  
  
clean\_bang\_longer\_status = clean\_bang\_longer\_status %>%   
 mutate(  
 membership\_period = factor(membership\_period, levels = c("status.01.18", "status.02.18","status.03.18", "status.04.18","status.05.18", "status.06.18",  
 "status.07.18", "status.08.18","status.09.18", "status.10.18","status.11.18", "status.12.18",  
 "status.01.19", "status.02.19","status.03.19", "status.04.19","status.05.19", "status.06.19",  
 "status.07.19", "status.08.19","status.09.19", "status.10.19","status.11.19", "status.12.19",  
 "status.01.20", "status.02.20","status.03.20", "status.04.20","status.05.20", "status.06.20",  
 "status.07.20", "status.08.20","status.09.20", "status.10.20"))  
 )

(Future Mike Here) I will also be creating a separate data set that will contain only the necessary variable that would be used in formulating models based on our data set. However, knowing that most of these variable will be not normally distributed and this will cause problems down the line, I will also be doing some data transformation in order to be able to build models that meet their own assumptions.

NOTE: Since I will be log transforming these data, there will be issues with entries of 0 which creates an undefined entry. A workaround that I’ve chosen to do is to add a constant that with a magnitude that is small enough as to not influence the overall impact of said variable. In this case, we will be doing the following:

* weighted average monthly membership rate will add 0.1, which is the equivalent of $0.10
* attendance rate will add 0.1, which is the equivalent of 0.1%
* total non-billing email interactions will add 1, which is the equivalent of 1 email
* percentage of total emails pertaining to CX-/scheduling-/service-related email interactions will add 0.1, which is the equivalent of 0.1%
* number of email-interactions per month will add 0.1, which is the equivalent of 0.1

clean\_bang\_select = clean\_bang\_final %>%   
 select(id,  
 age\_group,  
 employment\_sector,  
 current,  
 became\_former\_member,  
 churn\_type,  
 length,  
 retention\_3m,  
 retention\_6m,  
 retention\_12m,   
 revenue\_lifetime,  
 membership,   
 avg\_monthly\_rate,  
 monthly\_rate\_group,  
 monthly\_grouping\_ver.1,  
 monthly\_rate\_cutoff,  
 num\_breaks,  
 num\_renewals,  
 attendance\_rate,  
 attendance\_rate\_group,  
 attendance\_grouping\_ver.1,  
 attendance\_rate\_cutoff,  
 ever\_billing\_issue,  
 num\_billing\_issue,  
 new\_num\_total,  
 num\_emails\_month,  
 ever\_email\_month,  
 ever\_cx,  
 new\_per\_ticket\_cx,  
 ever\_scheduling,  
 new\_per\_ticket\_scheduling,  
 ever\_service,  
 new\_per\_ticket\_service   
 )  
  
clean\_bang\_select = clean\_bang\_select %>%   
 mutate(  
 log\_monthly\_rate = log(avg\_monthly\_rate + 0.1),  
 log\_attendance\_rate = log(attendance\_rate + 0.1),  
 log\_new\_num\_total = log(new\_num\_total + 1),  
 log\_new\_per\_ticket\_cx = log(new\_per\_ticket\_cx + 0.1),  
 log\_new\_per\_ticket\_scheduling = log(new\_per\_ticket\_scheduling + 0.1),  
 log\_new\_per\_ticket\_service = log(new\_per\_ticket\_service + 0.1),  
 num\_emails\_group = ifelse(new\_num\_total < 1, 0,  
 ifelse(c(new\_num\_total >= 1 & new\_num\_total < 10), 1,  
 ifelse(c(new\_num\_total >= 10 & new\_num\_total < 20), 2,  
 ifelse(c(new\_num\_total >= 20 & new\_num\_total < 50), 3,  
 ifelse(new\_num\_total >= 50, 4, NA))))),  
 log\_num\_emails\_month = log(num\_emails\_month + 0.1)  
 ) %>% mutate (  
 num\_emails\_group = as.factor(num\_emails\_group)   
 )   
  
clean\_bang\_select = clean\_bang\_select %>%   
 mutate(  
 num\_emails\_group = revalue(num\_emails\_group,   
 c("0" = "less than 1",  
 '1' = "1-9",  
 '2' = '10-19',  
 '3' = '20-49',  
 '4' = '50+'))  
 )   
  
clean\_bang\_select = clean\_bang\_select %>%   
 mutate(  
 num\_emails\_month\_group = ifelse(num\_emails\_month < 0.5, '0',  
 ifelse(c(num\_emails\_month >= 0.5 & num\_emails\_month < 1.5), "1",  
 ifelse(c(num\_emails\_month >= 1.5 & num\_emails\_month < 2.5), "2",  
 ifelse(num\_emails\_month >= 2.5, "3", NA))))   
 ) %>% mutate(  
 num\_emails\_month\_group = as.factor(num\_emails\_month\_group)  
 ) %>% mutate(  
 num\_emails\_month\_group = revalue(num\_emails\_month\_group,   
 c("0" = "none",   
 "1" = "1/month",   
 "2" = "2/month",   
 "3" = "3+/month"))  
 )   
  
clean\_bang\_select = clean\_bang\_select %>%   
 select(  
 id,   
 age\_group,   
 employment\_sector,  
 became\_former\_member,  
 churn\_type,  
 length,   
 retention\_3m,  
 retention\_6m,  
 retention\_12m,  
 revenue\_lifetime,  
 membership,  
 avg\_monthly\_rate,  
 log\_monthly\_rate,  
 monthly\_rate\_group,  
 monthly\_grouping\_ver.1,  
 monthly\_rate\_cutoff,  
 num\_breaks,  
 num\_renewals,  
 attendance\_rate,  
 log\_attendance\_rate,  
 attendance\_rate\_group,  
 attendance\_grouping\_ver.1,  
 attendance\_rate\_cutoff,  
 ever\_billing\_issue,  
 num\_billing\_issue,  
 new\_num\_total,   
 log\_new\_num\_total,  
 num\_emails\_group,  
 ever\_email\_month,  
 num\_emails\_month,  
 num\_emails\_month\_group,  
 log\_num\_emails\_month,  
 new\_num\_total,  
 ever\_cx,  
 new\_per\_ticket\_cx,  
 log\_new\_per\_ticket\_cx,  
 ever\_scheduling,  
 new\_per\_ticket\_scheduling,  
 log\_new\_per\_ticket\_scheduling,  
 ever\_service,  
 new\_per\_ticket\_service,  
 log\_new\_per\_ticket\_service  
 )  
  
View(clean\_bang\_select)

## **Analysis Plan**

### DESCRIPTIVE STATISTICS

Based on the data type, we will use either mean +/- SD or median to summarize the distribution of a given variable. However in the case of non-normal distributed data, we will instead be using median. This will be displayed on the appropriate medium. Primarily, we are interested in observing the distribution of demographics (i.e. age + employment sector), attendance rates, number of breaks email correspondence, etc. In comparing differences b/t groups, Student’s T-test will be used for continuous variable whilst the Pearson’s Chi-Square test will be run for categorical variables with significance cut-off set at p = 0.05. ANOVA will be used for cases where there are more than 3 groups for analyzing differences in continuous variable between groups with the Holm correction for pairwise adjustments. For instances of non-normal distribution, the Mann-Whitney Tests, Kruskal Wallis test (w. Holm correction) will be used instead.

### INFERENTIAL STATISTICS

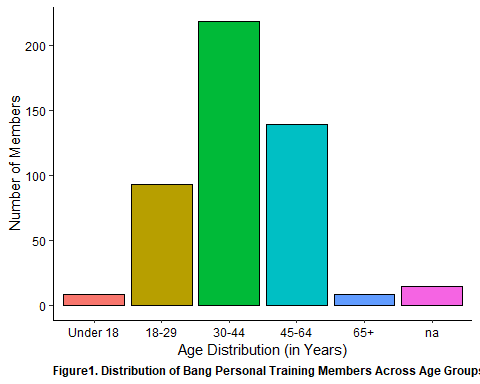
In order to assess the influence of various data gathered on membership churn, I will be looking to use a Cox Regression model in order to gain insight on how these factors play on length of membership prior to leaving. This will be determined through two methods: (a) random survival forest as well as (b) Cox regressional analysis. Additionally, the infuence of various predictors on retention status at 3-, 6- and 12-months were also examined through both (a) logisic regression analysis and (b) random forest modelling. The selection of predictors for the linear modelling approach (i.e. Cox-regression + logistic regression) were determined via stepwise regression using AIC as the measure for determining variable retention.

# **RESULTS**

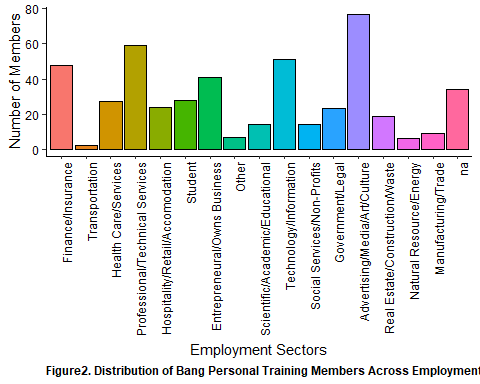
## Overall

Looking at the overall number of members over the period of Jan 2018 to Oct 2020, there were 483 members (98 current vs. 385 former). The majority of our members existed b/t the ages of 30-44 & the 45-64 age group. Notably, most of the members came from the advertising/media/culture/art, technology/information, professional/technical services and financial/insurance sectors. The most popular membership types were the 2x/week and 3x/week Hybrid Training memberships. As it pertained to the length of membership, the median duration is 121 days (i.e. approx. 4 months) with the average monthly membership rate of ~ $350. The average attendance rate for our members being approximately 60~ish% which isn’t surprising as the majority of our former members cited the lack of availability or accessibility as a reason for leaving Bang Personal Training. However, it is important to recognize that the pandemic played a noticeable role in loss of membership as noted by the drop in membership in March 2020.

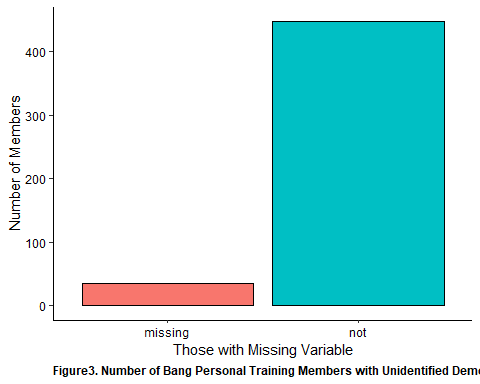
# PART A: Visualize - Unclean Data  
  
finalized\_bang %>%   
 ggplot(aes(x = age\_group, fill = age\_group)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(hjust = 0, face = 'bold')  
 ) +   
 xlab("Age Distribution (in Years)") +   
 ylab("Number of Members") +   
 labs(caption = "Figure1. Distribution of Bang Personal Training Members Across Age Groups") +   
 guides(fill = F)



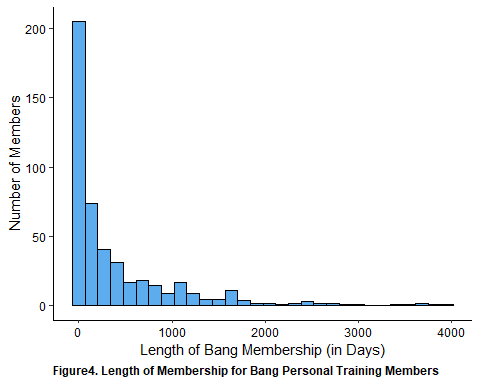
finalized\_bang %>%   
 ggplot(aes(x = employment\_sector, fill = employment\_sector)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(hjust = 0, face = 'bold')  
 ) +   
 xlab("Employment Sectors") +   
 ylab("Number of Members") +   
 labs(caption = "Figure2. Distribution of Bang Personal Training Members Across Employment Sectors")+  
 guides(fill = F)



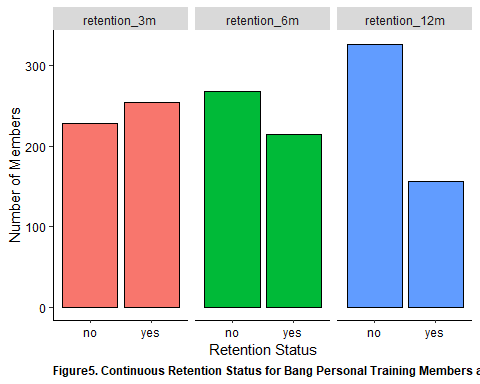
finalized\_bang %>%   
 ggplot(aes(x = missing\_value, fill = missing\_value)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(hjust = 0, face = 'bold')  
 ) +   
 xlab("Those with Missing Variable") +   
 ylab("Number of Members") +   
 guides(fill = F) +  
 labs(caption = "Figure3. Number of Bang Personal Training Members with Unidentified Demographic Variables")



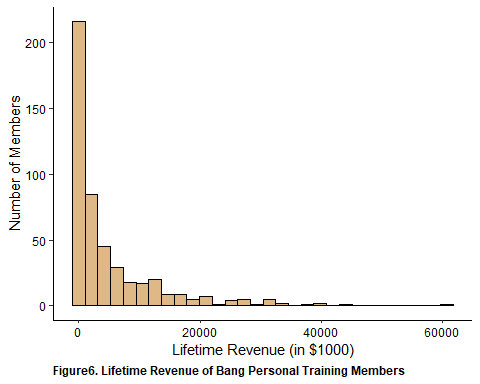
finalized\_bang %>%   
 ggplot(aes(x = length)) +  
 geom\_histogram(color = 'black', bins = 30, fill = 'steelblue2') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Length of Bang Membership (in Days)") +   
 ylab("Number of Members") +   
 guides(fill = F) +   
 labs(caption = 'Figure4. Length of Membership for Bang Personal Training Members')



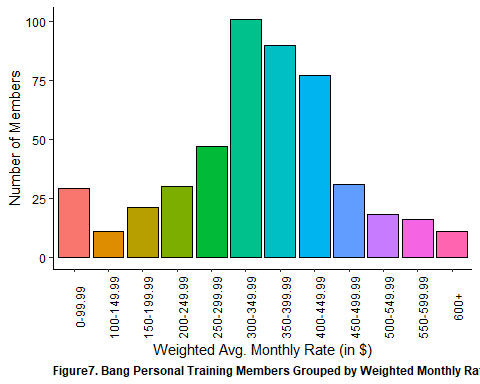
unclean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, fill = retention\_type)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 xlab("Retention Status") +   
 ylab('Number of Members') +   
 guides(fill = F) +   
 labs(caption = 'Figure5. Continuous Retention Status for Bang Personal Training Members at 3-Months, 6-Months and 12-Months')



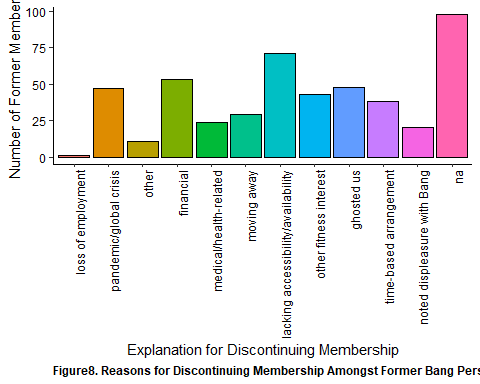
finalized\_bang %>%   
 ggplot(aes(x = c(revenue\_lifetime))) +   
 geom\_histogram(color = 'black', bins = 30, fill = 'burlywood') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +  
 xlab("Lifetime Revenue (in $1000)") +  
 ylab('Number of Members') +   
 guides(fill = F) +   
 labs(caption = 'Figure6. Lifetime Revenue of Bang Personal Training Members')



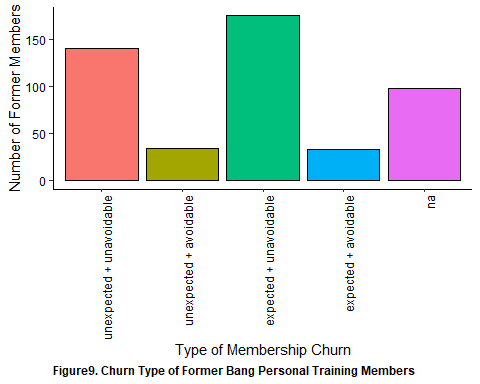
finalized\_bang %>%   
 filter(!is.na(monthly\_rate\_group)) %>%   
 ggplot(aes(x = monthly\_rate\_group, fill = monthly\_rate\_group)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 xlab("Weighted Avg. Monthly Rate (in $)") +  
 ylab("Number of Members") +  
 guides(fill = F) +  
 labs(caption = "Figure7. Bang Personal Training Members Grouped by Weighted Monthly Rates")



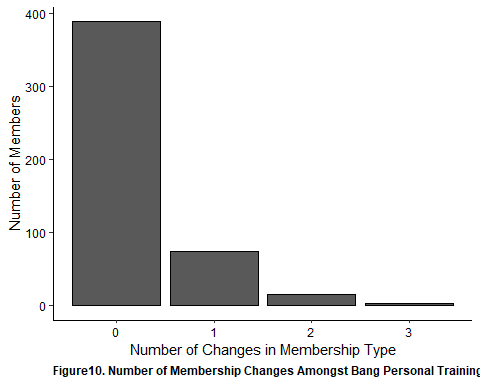
finalized\_bang %>%   
 ggplot(aes(x = reason\_to\_leave, fill = reason\_to\_leave)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Explanation for Discontinuing Membership") +   
 ylab("Number of Former Members") +   
 guides(fill = F) +   
 labs(caption = 'Figure8. Reasons for Discontinuing Membership Amongst Former Bang Personal Training Members')



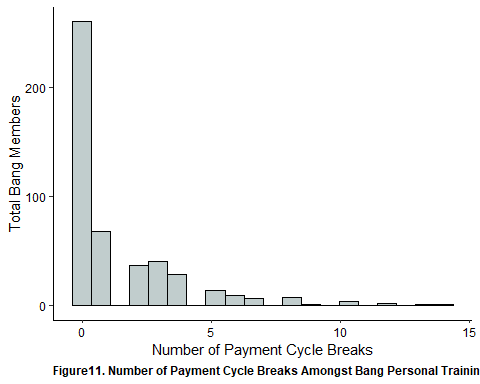
finalized\_bang %>%   
 ggplot(aes(x = churn\_type, fill = churn\_type)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Type of Membership Churn") +   
 ylab("Number of Former Members") +   
 guides(fill = F)+   
 labs(caption = 'Figure9. Churn Type of Former Bang Personal Training Members')



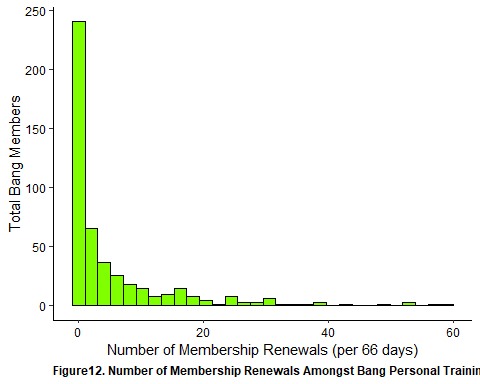
finalized\_bang %>%   
 ggplot(aes(x = num\_membership\_change)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Changes in Membership Type") +   
 ylab("Number of Members") +   
 guides(fill = F) +   
 labs(caption = 'Figure10. Number of Membership Changes Amongst Bang Personal Training Members')



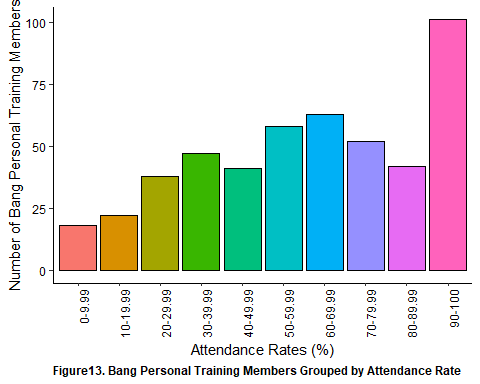
finalized\_bang %>%   
 ggplot(aes(x = num\_breaks)) +   
 geom\_histogram(color = 'black', fill = 'azure3', bins = 20) +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Payment Cycle Breaks") +  
 ylab("Total Bang Members") +   
 guides(fill = F) +   
 labs(caption = 'Figure11. Number of Payment Cycle Breaks Amongst Bang Personal Training Members')



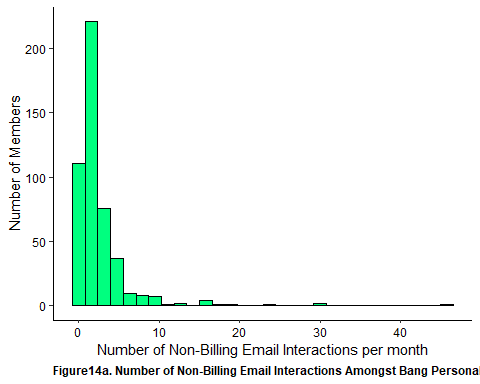
finalized\_bang %>%   
 ggplot(aes(x = num\_renewals)) +   
 geom\_histogram(color = 'black', fill = 'chartreuse', bins = 30) +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Membership Renewals (per 66 days)") +  
 ylab("Total Bang Members") +   
 guides(fill = F) +   
 labs(caption = 'Figure12. Number of Membership Renewals Amongst Bang Personal Training Members')



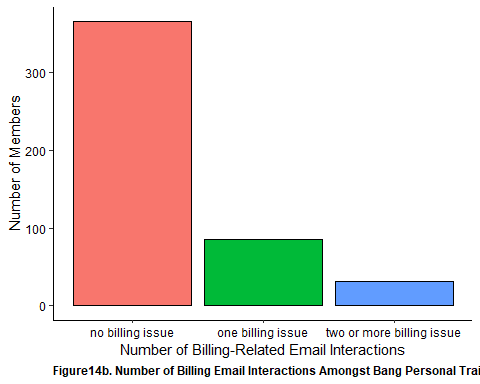
finalized\_bang %>%   
 filter(!is.na(attendance\_rate\_group)) %>%   
 ggplot(aes(x = attendance\_rate\_group, fill = attendance\_rate\_group)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Attendance Rates (%)") +   
 ylab("Number of Bang Personal Training Members") +   
 labs(caption = "Figure13. Bang Personal Training Members Grouped by Attendance Rate") +   
 guides(fill = F)



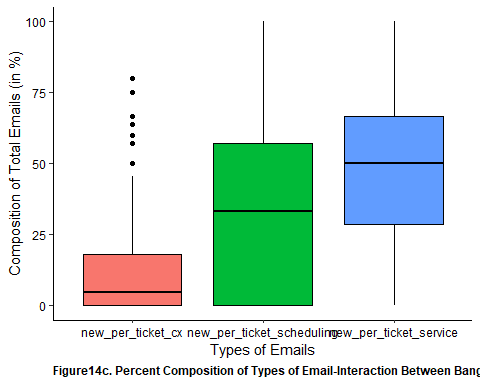
finalized\_bang %>%   
 ggplot(aes(x = num\_emails\_month)) +   
 geom\_histogram(color = 'black', fill = 'springgreen1', bins = 30) +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Non-Billing Email Interactions per month") +   
 ylab("Number of Members") +   
 guides(fill = F) +   
 labs(caption = "Figure14a. Number of Non-Billing Email Interactions Amongst Bang Personal Training Members")



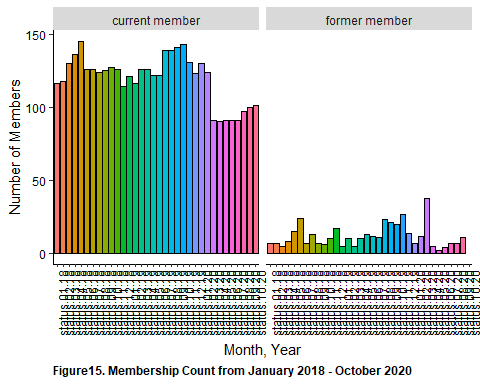
finalized\_bang %>%   
 ggplot(aes(x = num\_billing\_issue, fill = num\_billing\_issue)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Billing-Related Email Interactions") +   
 ylab("Number of Members") +   
 guides(fill = F) +   
 labs(caption = "Figure14b. Number of Billing Email Interactions Amongst Bang Personal Training Members")



unclean\_bang\_longer\_new\_per\_emails %>%   
 ggplot(aes(x = email\_type, y = email\_percent, fill = email\_type)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Types of Emails") +   
 ylab("Composition of Total Emails (in %)") +   
 guides(fill = F) +   
 labs(caption = 'Figure14c. Percent Composition of Types of Email-Interaction Between Bang Personal Training Members and Membership Service Staff')



unclean\_bang\_longer\_status %>%   
 mutate(  
 summarize\_membership\_status = ifelse(c(membership\_status == "new/returning" | membership\_status == "current"), "current member",  
 ifelse(c(membership\_status == "leaving" | membership\_status == "new/returning"), "former member",  
 ifelse(membership\_status == "no", "never a member", NA)))  
 ) %>%   
 mutate(  
 summarize\_membership\_status = as.factor(summarize\_membership\_status)  
 ) %>%   
 filter(summarize\_membership\_status != 'never a member') %>%   
 ggplot(aes(x = membership\_period, fill = membership\_period)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 axis.line = element\_line(colour = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text = element\_text(color = 'black')  
 ) +  
 guides(fill = F) +   
 ylab("Number of Members") +   
 xlab("Month, Year") +   
 facet\_wrap(vars(summarize\_membership\_status)) +  
 labs(caption = "Figure15. Membership Count from January 2018 - October 2020")



## Impact of Age, Employment and Membership Type

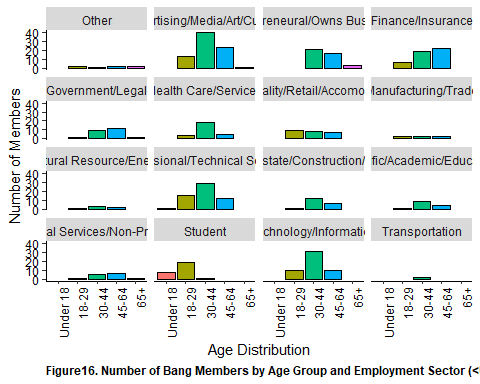
(NOTE: Going forward, we will be using the data set **WITHOUT** entries with missing demographic variables)

Looking further into the distribution of members, there was significant differences observed with respect to Age x Employment Sector and Age x Membership Type. While there were many differences found, it was notable to find that the majority of those within the 30-44 crowd were from the Technology/Information as well as Professional/Technical Services sector whilst those in the 45-64 crowd were the predominant age group within the finance/insurance sector. Additionally, although the 30-44 crowd were the predominant age group across most membership types, those in the 45-64 crowd were actually the predominant age group for distance coaching.

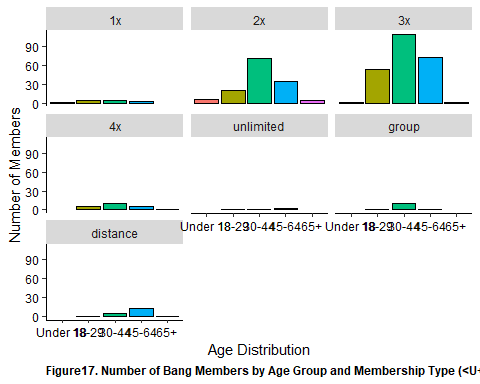
Looking at the impact of age, membership and employment sector on attendance rate, we see that attendance rate varied significantly with respect to age and employment sector, employment sector and membership type as well as membership type and age. Notably we see that:

\* Greater attendance rates among those with 2x/week membership across various age groups relative to other membership types  
\* Group membership were predominantly those within the 30-44 age category  
\* Those within the Technology/Information sector had the highest attendance rates relative to all other employment sector; the lowest were those within the hospitality/retail/accommodation sector   
\* Across age groups, we see those within the entrepreneurial space having the lowest median attendance rate.   
\* Lowest length were found in the Social Services/Non-Profit + Hospitality/Retail/Accomodation sector across age and membership types   
\* Highest overall across age and membership types were noted amongst those in the Entrepreneural and Tech sector (particularly at age 30-44); interestingly those within the Health Care sector had a very high length of membership for those aged 30-44   
\* As it relates to membership types, 2x/week had the highest length of membership across age + employment sector.

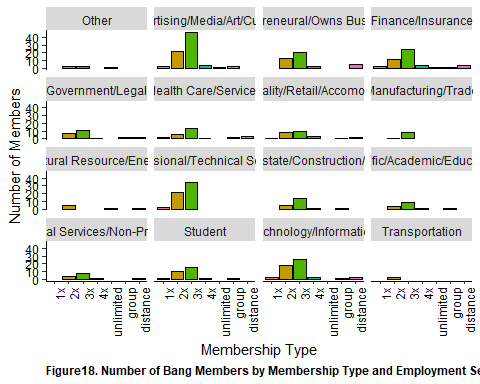
clean\_bang\_final %>%   
 ggplot(aes(x = age\_group, fill = age\_group)) +  
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Age Distribution") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure16. Number of Bang Members by Age Group and Employment Sector (χ2 = 251.52, p < 0.001)')



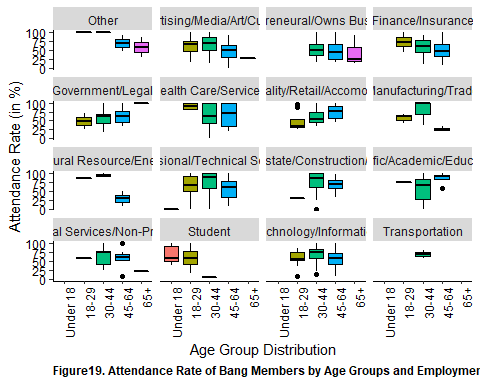
clean\_bang\_final %>%   
 ggplot(aes(x = age\_group, fill = age\_group)) +  
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Age Distribution") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(membership)) +  
 guides(fill = F) +   
 labs(caption = 'Figure17. Number of Bang Members by Age Group and Membership Type (χ2 = 36.85, p = 0.045)')



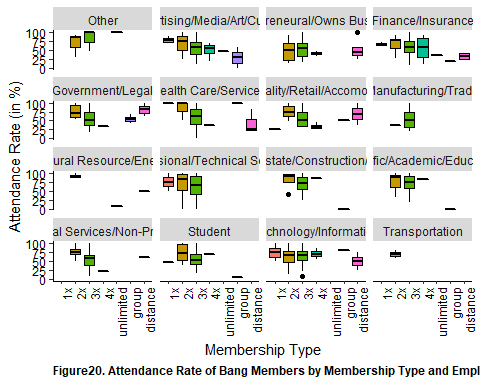
clean\_bang\_final %>%   
 ggplot(aes(x = membership, fill = membership)) +  
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Membership Type") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure18. Number of Bang Members by Membership Type and Employment Sector (χ2 = 101.73, p = 0.187)')



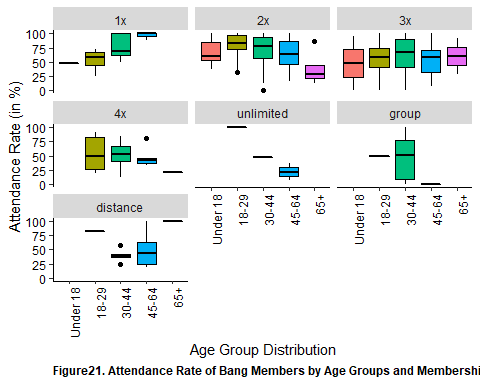
clean\_bang\_final %>%  
 ggplot(aes(x = age\_group, y = attendance\_rate, fill = age\_group)) +  
 geom\_boxplot(color = 'black')+  
 theme(   
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)  
 )+  
 xlab('Age Group Distribution')+  
 ylab("Attendance Rate (in %)") +  
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure19. Attendance Rate of Bang Members by Age Groups and Employment Sector')



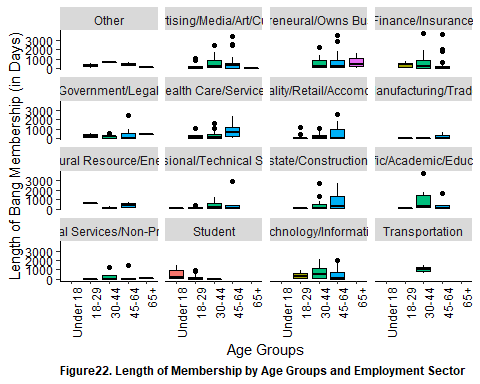
clean\_bang\_final %>%  
 ggplot(aes(x = membership, y = attendance\_rate, fill = membership)) +  
 geom\_boxplot(color = 'black')+  
 theme(   
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)  
 )+  
 xlab('Membership Type')+  
 ylab("Attendance Rate (in %)") +  
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure20. Attendance Rate of Bang Members by Membership Type and Employment Sector')



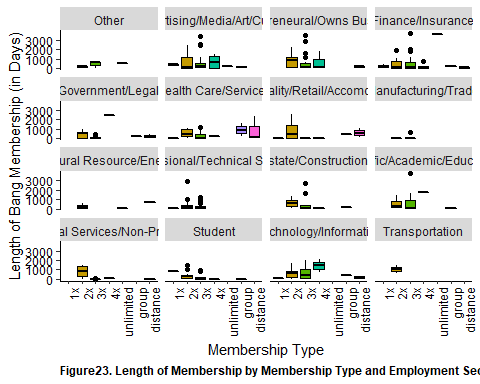
clean\_bang\_final %>%  
 ggplot(aes(x = age\_group, y = attendance\_rate, fill = age\_group)) +  
 geom\_boxplot(color = 'black')+  
 theme(   
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)  
 )+  
 xlab('Age Group Distribution')+  
 ylab("Attendance Rate (in %)") +  
 facet\_wrap(vars(membership)) +  
 guides(fill = F) +   
 labs(caption = 'Figure21. Attendance Rate of Bang Members by Age Groups and Membership Type')



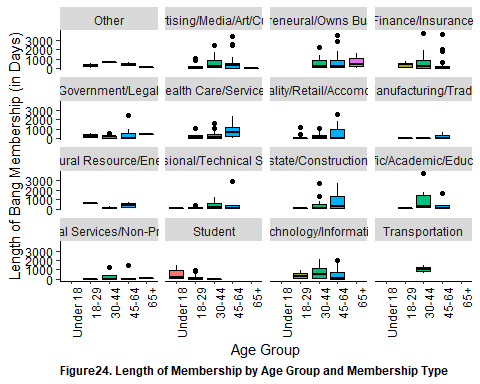
clean\_bang\_final %>%  
 ggplot(aes(x = age\_group, y = length, fill = age\_group)) +  
 geom\_boxplot(color = 'black')+  
 theme(   
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)  
 )+  
 xlab('Age Groups')+  
 ylab("Length of Bang Membership (in Days)") +  
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure22. Length of Membership by Age Groups and Employment Sector')



clean\_bang\_final %>%  
 ggplot(aes(x = membership, y = length, fill = membership)) +  
 geom\_boxplot(color = 'black')+  
 theme(   
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)  
 )+  
 xlab('Membership Type')+  
 ylab("Length of Bang Membership (in Days)") +  
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure23. Length of Membership by Membership Type and Employment Sector')



clean\_bang\_final %>%  
 ggplot(aes(x = age\_group, y = length, fill = age\_group)) +  
 geom\_boxplot(color = 'black')+  
 theme(   
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)  
 )+  
 xlab('Age Group')+  
 ylab("Length of Bang Membership (in Days)") +  
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure24. Length of Membership by Age Group and Membership Type')



## Current vs Former Members

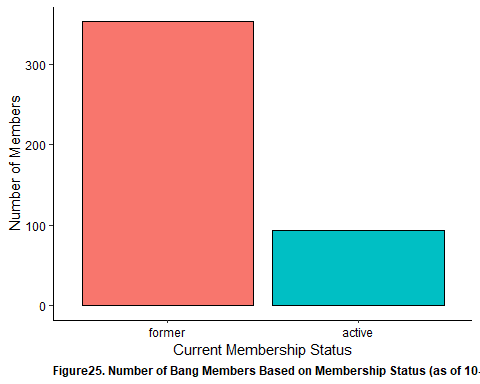
It was found that there was roughly and equivalent split of current members across 3 age categories (18-29, 30-44 and 45-64) for the most popular membership type (3x/week) between current and former members. However, overall, 30-44 was the most predominant age-group. Consistent with the overall data, the most common sector have been those within the advertisement/art/media/culture sector.

There were difference in the number of breaks in payment cycle as it was found there was significantly greater number of payment cycle breaks amongst current members than with former members. Additionally, the number of renewals were found to be significantly higher among active members than former members. However, there were no differences with respect to attendance rates or with number of membership changes b/t current and former members.

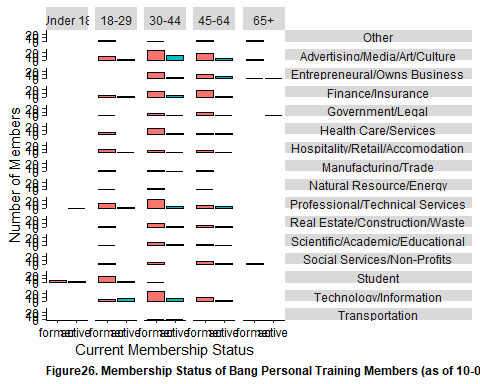
As it pertains to weighted average monthly membership rates, there were no difference b/t current and former members. However, once average monthly rates were categorized, it was found that those that a significantly greater proportion of active members have a higher monthly membership rate than former members.

Looking at email interactions, it was shown that those that are active members were more significantly more likely to have reported ever having a billing-related issue as compared to former members. Similarly, those that were current members were also found to have significantly greater percentage of their email interaction to be related to scheduling requests as compared to former members. Interestingly enough, while not statistically significant, those that were former members had a greater number of service-related email interaction as compared to former members. This relationship was also noted with respect to the number of non-billing related email interactions per month as those that were current members reported significantly less email interactions as compared to those that were former members.

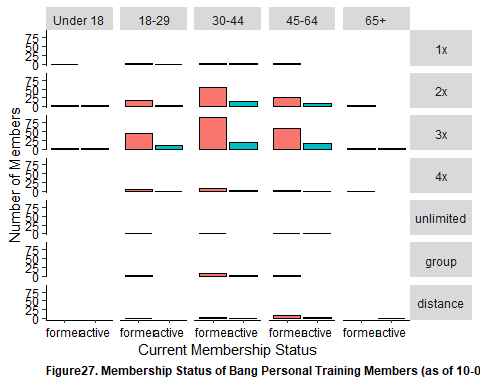
clean\_bang\_final %>%   
 ggplot(aes(x = current, fill = current)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Current Membership Status") +   
 ylab("Number of Members") +   
 labs(caption = "Figure25. Number of Bang Members Based on Membership Status (as of 10-05-2020)") +  
 guides(fill = F)



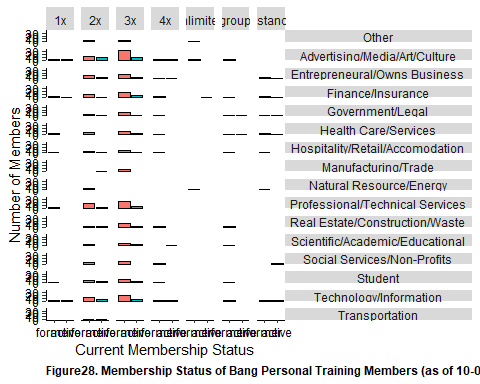
clean\_bang\_final %>%   
 ggplot(aes(x = current, fill = current)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Current Membership Status") +   
 ylab("Number of Members") +   
 labs(caption = "Figure26. Membership Status of Bang Personal Training Members (as of 10-05-2020) by Age and Employment Sector") +  
 facet\_grid(  
 cols = vars(age\_group),  
 rows = vars(employment\_sector)  
 ) +  
 guides(fill = F)



clean\_bang\_final %>%   
 ggplot(aes(x = current, fill = current)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Current Membership Status") +   
 ylab("Number of Members") +   
 labs(caption = "Figure27. Membership Status of Bang Personal Training Members (as of 10-05-2020) by Age and Membership Type") +  
 facet\_grid(  
 cols = vars(age\_group),  
 rows = vars(membership)  
 ) +  
 guides(fill = F)



clean\_bang\_final %>%   
 ggplot(aes(x = current, fill = current)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Current Membership Status") +   
 ylab("Number of Members") +   
 labs(caption = "Figure28. Membership Status of Bang Personal Training Members (as of 10-05-2020) by Employment Sector and Membership Type") +  
 facet\_grid(  
 cols = vars(membership),  
 rows = vars(employment\_sector)  
 ) +   
 guides(fill = F)



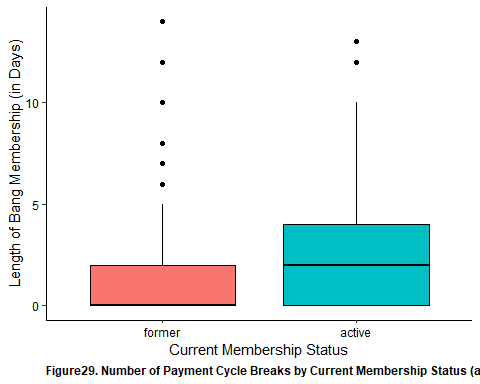
shapiro.test(clean\_bang\_final$num\_breaks) # Not a normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$num\_breaks  
## W = 0.7006, p-value < 2.2e-16

wilcox.test(num\_breaks ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: num\_breaks by current  
## W = 10640, p-value = 6.27e-09  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = current, y = num\_breaks, fill = current)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Current Membership Status") +   
 ylab("Length of Bang Membership (in Days)") +   
 labs(caption = "Figure29. Number of Payment Cycle Breaks by Current Membership Status (as of 10-05-2020) (W = 10640, p < 0.001)") +   
 guides(fill = F)



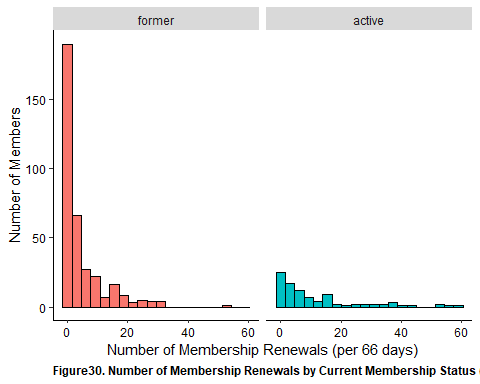
shapiro.test(clean\_bang\_final$num\_renewals) # Not a normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$num\_renewals  
## W = 0.65629, p-value < 2.2e-16

wilcox.test(num\_renewals ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: num\_renewals by current  
## W = 10478, p-value = 1.262e-08  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = num\_renewals, fill = current)) +  
 geom\_histogram(color = 'black', bins = 20) +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 facet\_wrap(vars(current)) +   
 guides(fill = F) +  
 xlab('Number of Membership Renewals (per 66 days)') +   
 ylab('Number of Members') +   
 labs(caption = "Figure30. Number of Membership Renewals by Current Membership Status (as of 10-05-2020) (W = 10478, p < 0.001)")



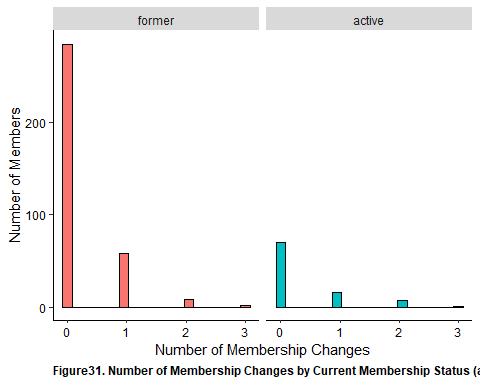
shapiro.test(clean\_bang\_final$num\_membership\_change) # Not a normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$num\_membership\_change  
## W = 0.51418, p-value < 2.2e-16

wilcox.test(num\_membership\_change ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: num\_membership\_change by current  
## W = 15402, p-value = 0.1289  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = num\_membership\_change, fill = current)) +   
 geom\_histogram(color = 'black', bins = 20) +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 facet\_wrap(vars(current)) +   
 guides(fill = F) +  
 xlab("Number of Membership Changes") +   
 ylab('Number of Members') +   
 labs(caption = 'Figure31. Number of Membership Changes by Current Membership Status (as of 10-05-2020) (W = 15402, p = 0.129)') +   
 guides(fill = F)



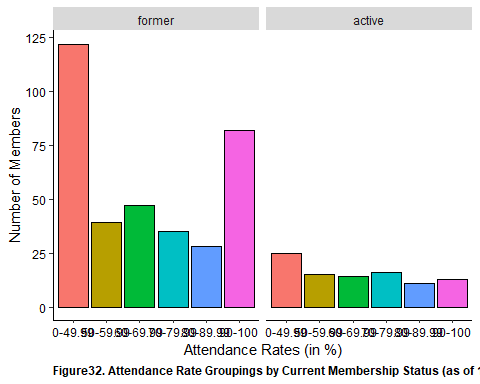
shapiro.test(clean\_bang\_final$attendance\_rate)

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$attendance\_rate  
## W = 0.95157, p-value = 6.257e-11

chisq.test(clean\_bang\_final$current, clean\_bang\_final$attendance\_grouping\_ver.1)

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$current and clean\_bang\_final$attendance\_grouping\_ver.1  
## X-squared = 10.63, df = 5, p-value = 0.05924

clean\_bang\_final %>%   
 ggplot(aes(x = attendance\_grouping\_ver.1, fill = attendance\_grouping\_ver.1)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Attendance Rates (in %)") +  
 ylab("Number of Members") +  
 facet\_wrap(vars(current)) +  
 labs(caption = "Figure32. Attendance Rate Groupings by Current Membership Status (as of 10-05-2020) (χ2 = 10.63, p = 0.059)") +  
 guides(fill = F)



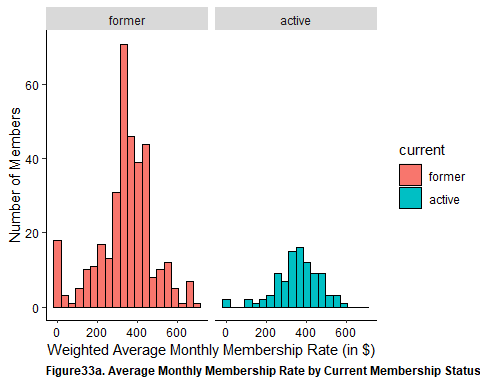
shapiro.test(clean\_bang\_final$avg\_monthly\_rate) # Not a normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$avg\_monthly\_rate  
## W = 0.95243, p-value = 8.273e-11

wilcox.test(avg\_monthly\_rate ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: avg\_monthly\_rate by current  
## W = 15328, p-value = 0.2563  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = avg\_monthly\_rate, fill = current)) +  
 geom\_histogram(color = 'black', bins = 20) +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 facet\_wrap(vars(current)) +   
 xlab('Weighted Average Monthly Membership Rate (in $)') +   
 ylab('Number of Members') +   
 labs(caption = "Figure33a. Average Monthly Membership Rate by Current Membership Status (as of 10-05-2020) (W = 15328, p = 0.256)")

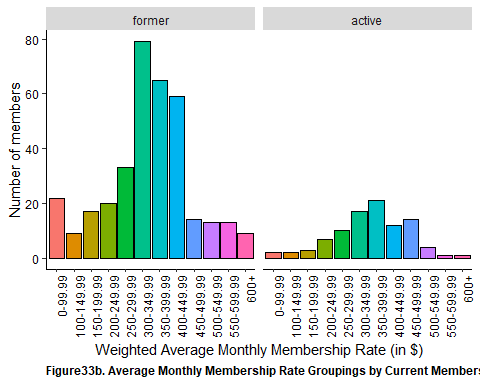


chisq.test(clean\_bang\_final$current, clean\_bang\_final$monthly\_rate\_group)

## Warning in chisq.test(clean\_bang\_final$current,  
## clean\_bang\_final$monthly\_rate\_group): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$current and clean\_bang\_final$monthly\_rate\_group  
## X-squared = 21.873, df = 11, p-value = 0.02537

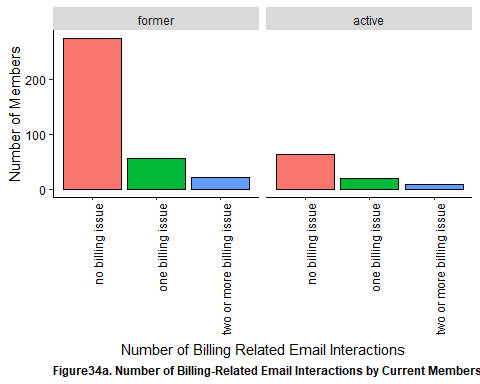
clean\_bang\_final %>%   
 ggplot(aes(x = monthly\_rate\_group, fill = monthly\_rate\_group)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 facet\_wrap(vars(current)) +   
 xlab("Weighted Average Monthly Membership Rate (in $)") +  
 ylab("Number of members") +   
 guides(fill = F) +  
 labs(caption = "Figure33b. Average Monthly Membership Rate Groupings by Current Membership Status (χ2 = 21.87, p = 0.025)")



chisq.test(clean\_bang\_final$num\_billing\_issue, clean\_bang\_final$current)

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$num\_billing\_issue and clean\_bang\_final$current  
## X-squared = 4.7508, df = 2, p-value = 0.09298

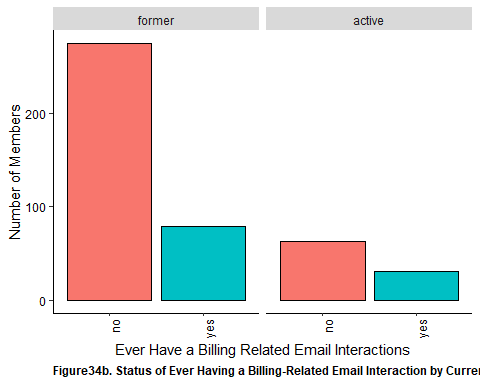
clean\_bang\_final %>%   
 ggplot(aes(x = num\_billing\_issue, fill = num\_billing\_issue)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 facet\_wrap(vars(current)) +  
 guides(fill = F) +  
 xlab("Number of Billing Related Email Interactions") +   
 ylab("Number of Members") +   
 labs(caption = "Figure34a. Number of Billing-Related Email Interactions by Current Membership Status (χ2 = 4.75, p = 0.093)")



chisq.test(clean\_bang\_final$ever\_billing\_issue, clean\_bang\_final$current)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: clean\_bang\_final$ever\_billing\_issue and clean\_bang\_final$current  
## X-squared = 3.9418, df = 1, p-value = 0.0471

clean\_bang\_final %>%   
 ggplot(aes(x = ever\_billing\_issue, fill = ever\_billing\_issue)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 facet\_wrap(vars(current)) +  
 guides(fill = F) +  
 xlab("Ever Have a Billing Related Email Interactions") +   
 ylab("Number of Members") +   
 labs(caption = "Figure34b. Status of Ever Having a Billing-Related Email Interaction by Current Membership Status (χ2 = 3.94, p = 0.047)")



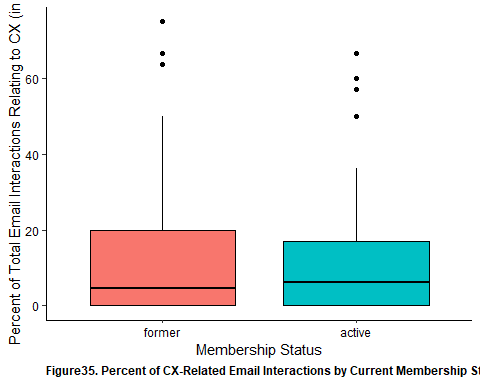
shapiro.test(clean\_bang\_final$new\_per\_ticket\_cx) # Not normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$new\_per\_ticket\_cx  
## W = 0.75859, p-value < 2.2e-16

wilcox.test(new\_per\_ticket\_cx ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: new\_per\_ticket\_cx by current  
## W = 15993, p-value = 0.5739  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = current, y = new\_per\_ticket\_cx, fill = current)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 guides(fill = F) +  
 xlab('Membership Status') +   
 ylab('Percent of Total Email Interactions Relating to CX (in %)') +   
 labs(caption = "Figure35. Percent of CX-Related Email Interactions by Current Membership Status (as of 10-05-2020) (W = 15993, p = 0.574)")



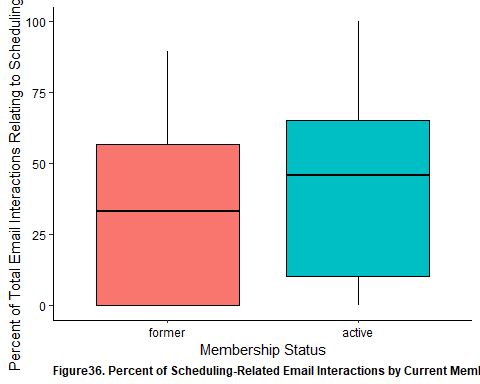
shapiro.test(clean\_bang\_final$new\_per\_ticket\_scheduling) # Not normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$new\_per\_ticket\_scheduling  
## W = 0.90146, p-value < 2.2e-16

wilcox.test(new\_per\_ticket\_scheduling ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: new\_per\_ticket\_scheduling by current  
## W = 14302, p-value = 0.03696  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = current, y = new\_per\_ticket\_scheduling, fill = current)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 guides(fill = F) +  
 xlab('Membership Status') +   
 ylab('Percent of Total Email Interactions Relating to Scheduling (in %)') +   
 labs(caption = "Figure36. Percent of Scheduling-Related Email Interactions by Current Membership Status (as of 10-05-2020) (W = 14302, p = 0.037)")



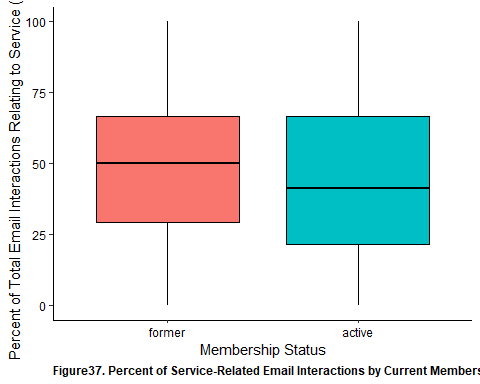
shapiro.test(clean\_bang\_final$new\_per\_ticket\_service) # Not normal distribution

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$new\_per\_ticket\_service  
## W = 0.93109, p-value = 1.71e-13

wilcox.test(new\_per\_ticket\_service ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: new\_per\_ticket\_service by current  
## W = 18718, p-value = 0.05499  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = current, y = new\_per\_ticket\_service, fill = current)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab('Membership Status') +   
 guides(fill = F) +  
 ylab('Percent of Total Email Interactions Relating to Service (in %)') +   
 labs(caption = "Figure37. Percent of Service-Related Email Interactions by Current Membership Status (as of 10-05-2020) (W = 18718, p = 0.054)")



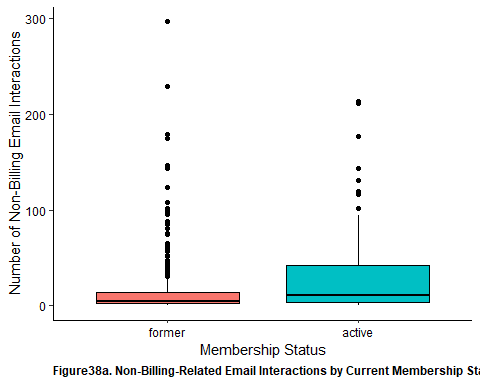
shapiro.test(clean\_bang\_final$new\_num\_total) # not normally distributed

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$new\_num\_total  
## W = 0.54485, p-value < 2.2e-16

wilcox.test(new\_num\_total ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: new\_num\_total by current  
## W = 12510, p-value = 0.0002348  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = current, y = new\_num\_total, fill = current)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab('Membership Status') +   
 guides(fill = F) +  
 ylab('Number of Non-Billing Email Interactions') +   
 labs(caption = "Figure38a. Non-Billing-Related Email Interactions by Current Membership Status (as of 10-05-2020) (W = 12510, p < 0.001)")



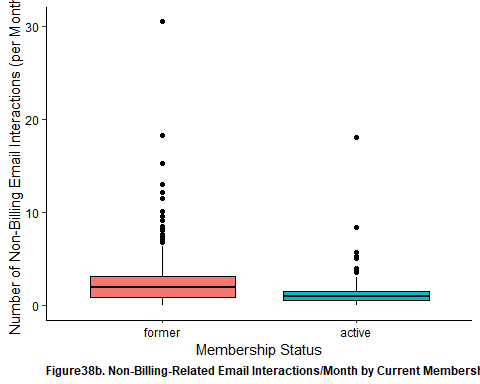
shapiro.test(clean\_bang\_final$num\_emails\_month) # not normally distributed

##   
## Shapiro-Wilk normality test  
##   
## data: clean\_bang\_final$num\_emails\_month  
## W = 0.6185, p-value < 2.2e-16

wilcox.test(num\_emails\_month ~ current, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: num\_emails\_month by current  
## W = 22726, p-value = 3.558e-08  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = current, y = num\_emails\_month, fill = current)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab('Membership Status') +   
 guides(fill = F) +  
 ylab('Number of Non-Billing Email Interactions (per Month)') +   
 labs(caption = "Figure38b. Non-Billing-Related Email Interactions/Month by Current Membership Status (as of 10-05-2020) (W = 22726, p < 0.001)")



## Length of Membership

Examining the length of membership across age groups, significantly longer membership length was observed in those aged 30-44 as compared to 18-29. This difference was also noted in terms of membership types with those with a 2x/week membership had significantly longer membership length as compared to 3x/week. As it relates to attendance rates, those that attended 70%-79% of the time had significantly longer membership length as compared to those engaging in less than 50% of the time.

In terms of email interactions, those that reported any billing-related issues were found to have longer membership rates than those without. Looking at the other types of email interactions, there was a significant correlation with increased percentage of email interaction with length of membership. In fact this was supported when examining the relationship between total non-billing related email interactions and length of membership.

Lastly in terms of average monthly rate, those with 350 to 449 per month were found to have significantly longer membership length as compared to all other monthly rates.

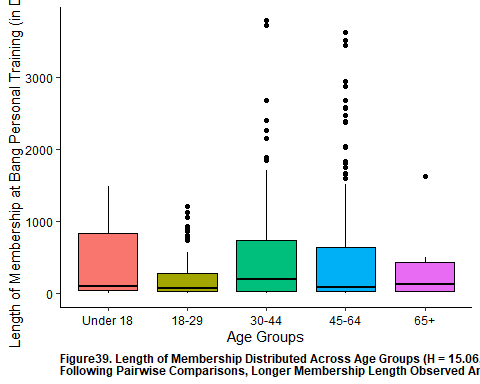
kruskal.test(length ~ age\_group, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by age\_group  
## Kruskal-Wallis chi-squared = 15.059, df = 4, p-value = 0.004581

clean\_bang\_final %>% dunn\_test(length ~ age\_group, p.adjust.method = 'holm')

## # A tibble: 10 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length Under 18 18-29 9 88 -1.36 0.175 1 ns   
## 2 length Under 18 30-44 9 211 0.0454 0.964 1 ns   
## 3 length Under 18 45-64 9 131 -0.384 0.701 1 ns   
## 4 length Under 18 65+ 9 8 -0.346 0.730 1 ns   
## 5 length 18-29 30-44 88 211 3.86 0.000114 0.00114 \*\*   
## 6 length 18-29 45-64 88 131 2.48 0.0131 0.117 ns   
## 7 length 18-29 65+ 88 8 0.829 0.407 1 ns   
## 8 length 30-44 45-64 211 131 -1.33 0.184 1 ns   
## 9 length 30-44 65+ 211 8 -0.509 0.611 1 ns   
## 10 length 45-64 65+ 131 8 -0.0984 0.922 1 ns

clean\_bang\_final %>%   
 ggplot(aes(x = age\_group, y = length, fill = age\_group)) +   
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Age Groups") +  
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure39. Length of Membership Distributed Across Age Groups (H = 15.06, p = 0.005).   
 Following Pairwise Comparisons, Longer Membership Length Observed Amongst 30-44 as Compared to 45-64 (Z = 3.86, p = 0.001)") +   
 guides(fill = F)



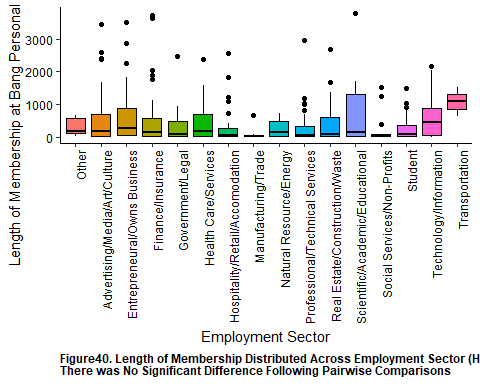
kruskal.test(length ~ employment\_sector, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by employment\_sector  
## Kruskal-Wallis chi-squared = 28.526, df = 15, p-value = 0.0185

clean\_bang\_final %>% dunn\_test(length ~ employment\_sector, p.adjust.method = 'holm')

## # A tibble: 120 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length Other Advertising/Me~ 7 77 -0.153 0.878 1 ns   
## 2 length Other Entrepreneural~ 7 41 0.183 0.855 1 ns   
## 3 length Other Finance/Insura~ 7 48 -0.591 0.554 1 ns   
## 4 length Other Government/Leg~ 7 23 -0.880 0.379 1 ns   
## 5 length Other Health Care/Se~ 7 27 -0.382 0.702 1 ns   
## 6 length Other Hospitality/Re~ 7 24 -0.969 0.332 1 ns   
## 7 length Other Manufacturing/~ 7 9 -1.84 0.0658 1 ns   
## 8 length Other Natural Resour~ 7 6 -0.463 0.643 1 ns   
## 9 length Other Professional/T~ 7 57 -1.05 0.292 1 ns   
## 10 length Other Real Estate/Co~ 7 19 -0.202 0.840 1 ns   
## # ... with 110 more rows

clean\_bang\_final %>%   
 ggplot(aes(x = employment\_sector, y = length, fill = employment\_sector)) +   
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Employment Sector") +  
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure40. Length of Membership Distributed Across Employment Sector (H = 28.53, p = 0.019).   
 There was No Significant Difference Following Pairwise Comparisons") +   
 guides(fill = F)



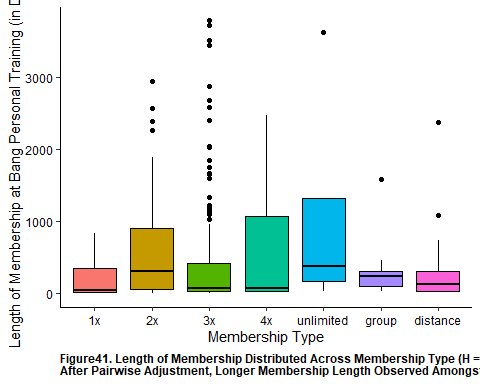
kruskal.test(length ~ membership, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by membership  
## Kruskal-Wallis chi-squared = 20.321, df = 6, p-value = 0.002427

clean\_bang\_final %>% dunn\_test(length ~ membership, p.adjust.method = 'holm')

## # A tibble: 21 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length 1x 2x 13 137 2.52 0.0118 0.235 ns   
## 2 length 1x 3x 13 239 1.04 0.300 1 ns   
## 3 length 1x 4x 13 22 1.42 0.155 1 ns   
## 4 length 1x unlimited 13 4 1.56 0.119 1 ns   
## 5 length 1x group 13 12 1.37 0.171 1 ns   
## 6 length 1x distance 13 20 0.965 0.335 1 ns   
## 7 length 2x 3x 137 239 -4.07 0.0000480 0.00101 \*\*   
## 8 length 2x 4x 137 22 -1.02 0.309 1 ns   
## 9 length 2x unlimited 137 4 0.315 0.753 1 ns   
## 10 length 2x group 137 12 -0.607 0.544 1 ns   
## # ... with 11 more rows

clean\_bang\_final %>%   
 ggplot(aes(x = membership, y = length, fill = membership)) +   
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Membership Type") +  
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure41. Length of Membership Distributed Across Membership Type (H = 19.94, p = 0.003.   
 After Pairwise Adjustment, Longer Membership Length Observed Amongst 2x/Week vs. 3x/Week (Z = 4.07, p = 0.001)") +   
 guides(fill = F)



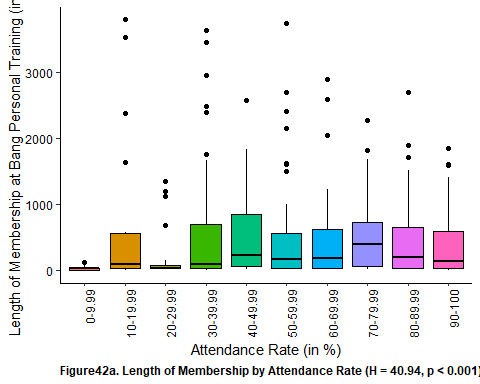
kruskal.test(length ~ attendance\_rate\_group, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by attendance\_rate\_group  
## Kruskal-Wallis chi-squared = 40.936, df = 9, p-value = 5.138e-06

dunn\_test(length ~ attendance\_rate\_group, data = clean\_bang\_final, p.adjust.method = 'holm')

## # A tibble: 45 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length 0-9.99 10-19.~ 16 19 2.83 4.72e-3 1.51e-1 ns   
## 2 length 0-9.99 20-29.~ 16 32 1.50 1.34e-1 1.00e+0 ns   
## 3 length 0-9.99 30-39.~ 16 42 3.63 2.83e-4 1.07e-2 \*   
## 4 length 0-9.99 40-49.~ 16 38 4.45 8.46e-6 3.72e-4 \*\*\*   
## 5 length 0-9.99 50-59.~ 16 54 4.04 5.45e-5 2.18e-3 \*\*   
## 6 length 0-9.99 60-69.~ 16 61 4.02 5.85e-5 2.28e-3 \*\*   
## 7 length 0-9.99 70-79.~ 16 51 4.97 6.62e-7 2.98e-5 \*\*\*\*   
## 8 length 0-9.99 80-89.~ 16 39 4.09 4.24e-5 1.74e-3 \*\*   
## 9 length 0-9.99 90-100 16 95 4.16 3.19e-5 1.34e-3 \*\*   
## 10 length 10-19.99 20-29.~ 19 32 -1.72 8.47e-2 1.00e+0 ns   
## # ... with 35 more rows

clean\_bang\_final %>%   
 ggplot(aes(x = attendance\_rate\_group, y = length, fill = attendance\_rate\_group)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 guides(fill = F) +  
 xlab("Attendance Rate (in %)") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure42a. Length of Membership by Attendance Rate (H = 40.94, p < 0.001)")



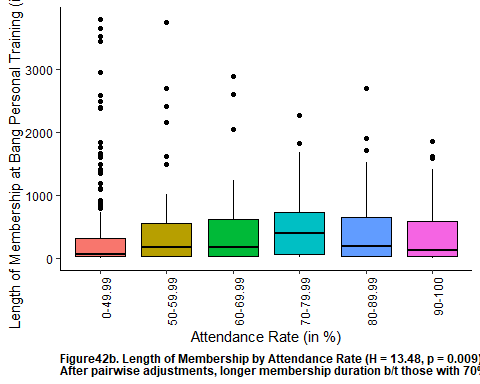
kruskal.test(length ~ attendance\_grouping\_ver.1, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by attendance\_grouping\_ver.1  
## Kruskal-Wallis chi-squared = 13.715, df = 5, p-value = 0.01752

dunn\_test(length ~ attendance\_grouping\_ver.1, data = clean\_bang\_final, p.adjust.method = 'holm')

## # A tibble: 15 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length 0-49.99 50-59.99 147 54 1.74 0.0819 0.983 ns   
## 2 length 0-49.99 60-69.99 147 61 1.69 0.0914 0.983 ns   
## 3 length 0-49.99 70-79.99 147 51 3.40 0.000666 0.0100 \*\*   
## 4 length 0-49.99 80-89.99 147 39 1.91 0.0563 0.775 ns   
## 5 length 0-49.99 90-100 147 95 1.92 0.0553 0.775 ns   
## 6 length 50-59.99 60-69.99 54 61 -0.106 0.916 1 ns   
## 7 length 50-59.99 70-79.99 54 51 1.41 0.157 1 ns   
## 8 length 50-59.99 80-89.99 54 39 0.318 0.750 1 ns   
## 9 length 50-59.99 90-100 54 95 -0.144 0.885 1 ns   
## 10 length 60-69.99 70-79.99 61 51 1.56 0.119 1 ns   
## 11 length 60-69.99 80-89.99 61 39 0.423 0.672 1 ns   
## 12 length 60-69.99 90-100 61 95 -0.0294 0.977 1 ns   
## 13 length 70-79.99 80-89.99 51 39 -0.984 0.325 1 ns   
## 14 length 70-79.99 90-100 51 95 -1.73 0.0831 0.983 ns   
## 15 length 80-89.99 90-100 39 95 -0.481 0.630 1 ns

clean\_bang\_final %>%   
 ggplot(aes(x = attendance\_grouping\_ver.1, y = length, fill = attendance\_grouping\_ver.1)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 guides(fill = F) +  
 xlab("Attendance Rate (in %)") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure42b. Length of Membership by Attendance Rate (H = 13.48, p = 0.009).   
 After pairwise adjustments, longer membership duration b/t those with 70%-79.99% attendance rate as compared to 0-49.99% (Z = 3.40, p = 0.007)")



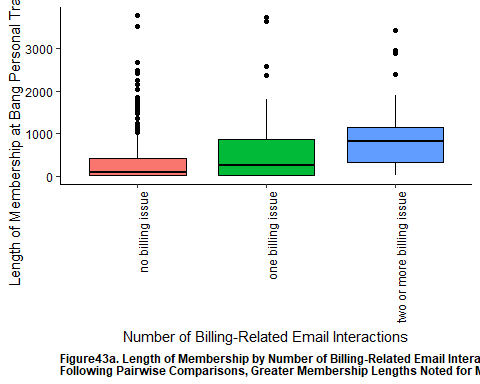
kruskal.test(length ~ num\_billing\_issue, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by num\_billing\_issue  
## Kruskal-Wallis chi-squared = 31.239, df = 2, p-value = 1.646e-07

dunn\_test(length ~ num\_billing\_issue, data = clean\_bang\_final, p.adjust.method = 'holm')

## # A tibble: 3 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length no billi~ one bill~ 337 78 1.91 5.60e-2 5.60e-2 ns   
## 2 length no billi~ two or m~ 337 32 5.45 4.94e-8 1.48e-7 \*\*\*\*   
## 3 length one bill~ two or m~ 78 32 3.66 2.51e-4 5.02e-4 \*\*\*

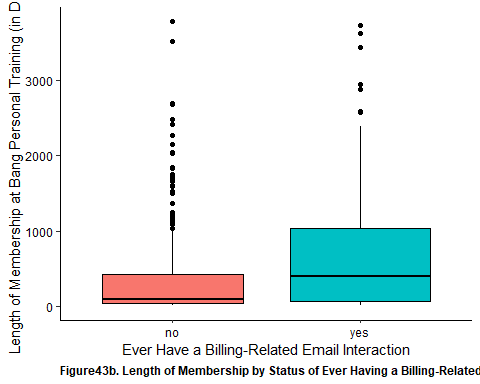
clean\_bang\_final %>%   
 ggplot(aes(x = num\_billing\_issue, y = length, fill = num\_billing\_issue)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 guides(fill = F) +  
 xlab("Number of Billing-Related Email Interactions") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure43a. Length of Membership by Number of Billing-Related Email Interactions (H = 31.24, p < 0.001).  
 Following Pairwise Comparisons, Greater Membership Lengths Noted for Members with 2 or More such Interactions.")



wilcox.test(length ~ ever\_billing\_issue, data = clean\_bang\_final)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: length by ever\_billing\_issue  
## W = 13570, p-value = 2.414e-05  
## alternative hypothesis: true location shift is not equal to 0

clean\_bang\_final %>%   
 ggplot(aes(x = ever\_billing\_issue, y = length, fill = ever\_billing\_issue)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 ) +   
 guides(fill = F) +  
 xlab("Ever Have a Billing-Related Email Interaction") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure43b. Length of Membership by Status of Ever Having a Billing-Related Issue (W = 13570, p < 0.001)")



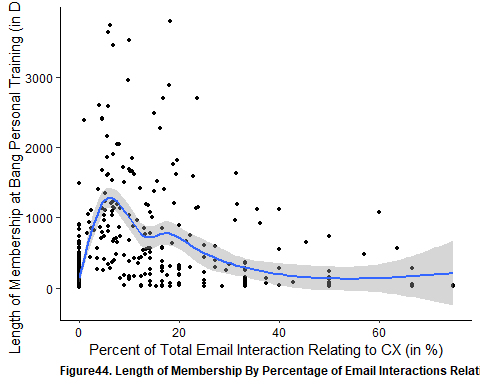
cor.test(x = clean\_bang\_final$new\_per\_ticket\_cx, y = clean\_bang\_final$length, method = 'spearman')

## Warning in cor.test.default(x = clean\_bang\_final$new\_per\_ticket\_cx, y =  
## clean\_bang\_final$length, : Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: clean\_bang\_final$new\_per\_ticket\_cx and clean\_bang\_final$length  
## S = 11297350, p-value = 2.488e-07  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.24106

clean\_bang\_final %>%   
 ggplot(aes(x = new\_per\_ticket\_cx, y = length)) +   
 geom\_point(color = 'black', size = 1) +   
 geom\_smooth(method = 'gam') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Percent of Total Email Interaction Relating to CX (in %)") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +  
 labs(caption = "Figure44. Length of Membership By Percentage of Email Interactions Relating to CX (ρ = 0.241, p < 0.001)")

## `geom\_smooth()` using formula 'y ~ s(x, bs = "cs")'



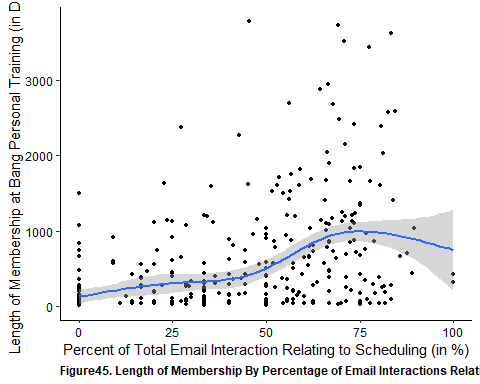
cor.test(x = clean\_bang\_final$new\_per\_ticket\_scheduling, y = clean\_bang\_final$length, method = 'spearman')

## Warning in cor.test.default(x = clean\_bang\_final$new\_per\_ticket\_scheduling, :  
## Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: clean\_bang\_final$new\_per\_ticket\_scheduling and clean\_bang\_final$length  
## S = 7629040, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.4874919

clean\_bang\_final %>%   
 ggplot(aes(x = new\_per\_ticket\_scheduling, y = length)) +   
 geom\_point(color = 'black', size = 1) +   
 geom\_smooth(method = 'gam') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +  
 labs(caption = "Figure45. Length of Membership By Percentage of Email Interactions Relating to Scheduling (ρ = 0.487, p < 0.001)")

## `geom\_smooth()` using formula 'y ~ s(x, bs = "cs")'



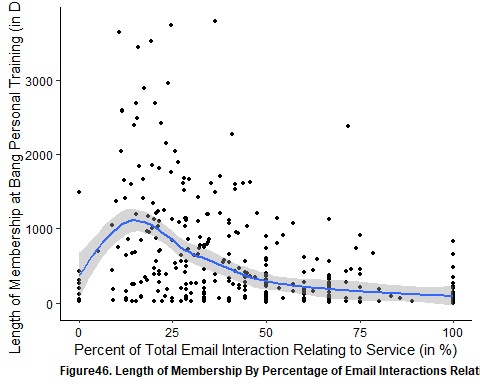
cor.test(x = clean\_bang\_final$new\_per\_ticket\_service, y = clean\_bang\_final$length, method = 'spearman')

## Warning in cor.test.default(x = clean\_bang\_final$new\_per\_ticket\_service, :  
## Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: clean\_bang\_final$new\_per\_ticket\_service and clean\_bang\_final$length  
## S = 21968204, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## -0.4757929

clean\_bang\_final %>%   
 ggplot(aes(x = new\_per\_ticket\_service, y = length)) +   
 geom\_point(color = 'black', size = 1) +   
 geom\_smooth(method = 'gam') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Percent of Total Email Interaction Relating to Service (in %)") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +  
 labs(caption = "Figure46. Length of Membership By Percentage of Email Interactions Relating to Service (ρ = -0.476, p < 0.001)")

## `geom\_smooth()` using formula 'y ~ s(x, bs = "cs")'



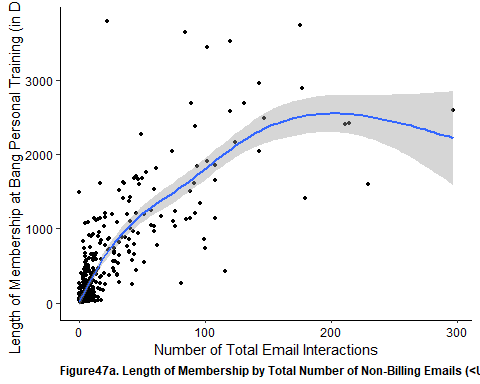
cor.test(x = clean\_bang\_final$new\_num\_total, y = clean\_bang\_final$length, method = 'spearman')

## Warning in cor.test.default(x = clean\_bang\_final$new\_num\_total, y =  
## clean\_bang\_final$length, : Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: clean\_bang\_final$new\_num\_total and clean\_bang\_final$length  
## S = 3263782, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.7807438

clean\_bang\_final %>%   
 ggplot(aes(x = new\_num\_total, y = length)) +   
 geom\_point(color = 'black', size = 1) +  
 geom\_smooth(method = 'gam') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Total Email Interactions") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = 'Figure47a. Length of Membership by Total Number of Non-Billing Emails (ρ = 0.781, p < 0.001)')

## `geom\_smooth()` using formula 'y ~ s(x, bs = "cs")'



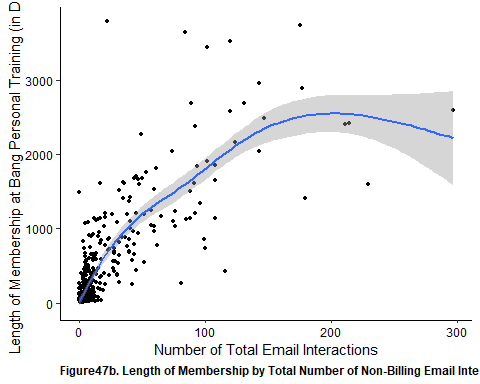
cor.test(x = clean\_bang\_final$num\_emails\_month, y = clean\_bang\_final$length, method = 'spearman')

## Warning in cor.test.default(x = clean\_bang\_final$num\_emails\_month, y =  
## clean\_bang\_final$length, : Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: clean\_bang\_final$num\_emails\_month and clean\_bang\_final$length  
## S = 23277953, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## -0.5637799

clean\_bang\_final %>%   
 ggplot(aes(x = new\_num\_total, y = length)) +   
 geom\_point(color = 'black', size = 1) +  
 geom\_smooth(method = 'gam') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Number of Total Email Interactions") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = 'Figure47b. Length of Membership by Total Number of Non-Billing Email Interactions per Month (ρ = -0.564, p < 0.001)')

## `geom\_smooth()` using formula 'y ~ s(x, bs = "cs")'



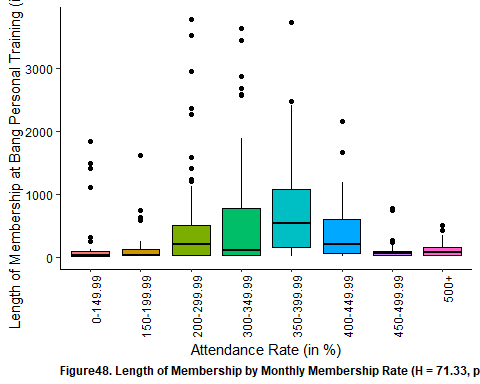
kruskal.test(length ~ monthly\_grouping\_ver.1, data = clean\_bang\_final)

##   
## Kruskal-Wallis rank sum test  
##   
## data: length by monthly\_grouping\_ver.1  
## Kruskal-Wallis chi-squared = 71.341, df = 7, p-value = 7.912e-13

dunn\_test(length ~ monthly\_grouping\_ver.1, data = clean\_bang\_final, p.adjust.method = 'holm')

## # A tibble: 28 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 length 0-149.99 150-19~ 35 20 1.15 2.52e- 1 1.00e+0 ns   
## 2 length 0-149.99 200-29~ 35 70 3.90 9.74e- 5 2.24e-3 \*\*   
## 3 length 0-149.99 300-34~ 35 96 3.78 1.55e- 4 3.24e-3 \*\*   
## 4 length 0-149.99 350-39~ 35 86 6.56 5.23e-11 1.46e-9 \*\*\*\*   
## 5 length 0-149.99 400-44~ 35 71 4.50 6.80e- 6 1.70e-4 \*\*\*   
## 6 length 0-149.99 450-49~ 35 28 0.995 3.20e- 1 1.00e+0 ns   
## 7 length 0-149.99 500+ 35 41 1.01 3.14e- 1 1.00e+0 ns   
## 8 length 150-199~ 200-29~ 20 70 1.92 5.54e- 2 6.10e-1 ns   
## 9 length 150-199~ 300-34~ 20 96 1.73 8.31e- 2 8.31e-1 ns   
## 10 length 150-199~ 350-39~ 20 86 4.01 6.12e- 5 1.47e-3 \*\*   
## # ... with 18 more rows

clean\_bang\_final %>%   
 ggplot(aes(x = monthly\_grouping\_ver.1, y = length, fill = monthly\_grouping\_ver.1)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 guides(fill = F) +  
 xlab("Attendance Rate (in %)") +   
 ylab("Length of Membership at Bang Personal Training (in Days)") +   
 labs(caption = "Figure48. Length of Membership by Monthly Membership Rate (H = 71.33, p < 0.001)")

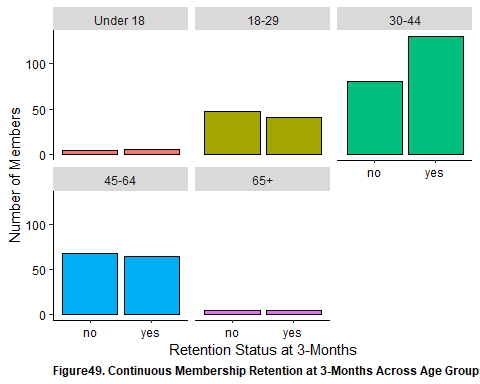


## Retention Status at 3-Months, 6-Months and 12-Months

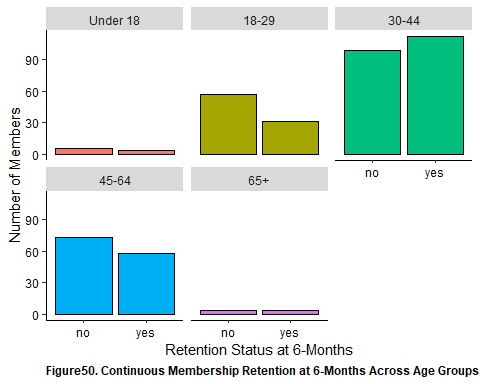
Looking at membership retention across the three time points, it was found that the rates of retention are 54.6% at 3 months, 46.8% at 6 months and 33.8% at 12 months. Notably it was found that retention status significantly differed between employment sectors at 3 and 6 months. This difference was also noted with respect to membership types as well. It was also found that those that had a higher attendance rate also were more likely to have remained a member at Bang. This relationship appeared to not have changed across all three time point. Similarly, there were also significant differences with respect to average monthly membership rates with greater retention rates across all time points with those with 350/month to 399/month as a membership rate.

Looking at the impact of email interactions, those that had continued their membership reported more billing-related email interactions than those that did not. Further look into the other types of email interactions, there were greater percentage of CX and scheduling-related email interactions amongst those that had maintained their membership at Bang Personal Training at each time point. However, there was a significantly lower percentage of service-related email interactions amongst those that had maintained their membership as compared to those that did not. Overall, those that retained their Bang Personal Training membership were found to have greater email interactions than those that did not across all time points. However, there is a lower number of email interaction per month amongst those that retained their membership, which was evident across all time points.

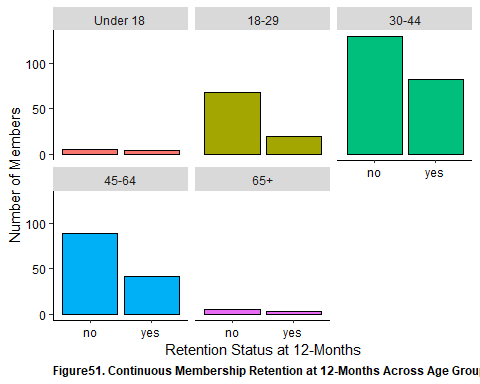
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, fill = age\_group)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 3-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(age\_group)) +  
 guides(fill = F) +   
 labs(caption = 'Figure49. Continuous Membership Retention at 3-Months Across Age Groups (χ2 = 8.28, p = 0.082)')



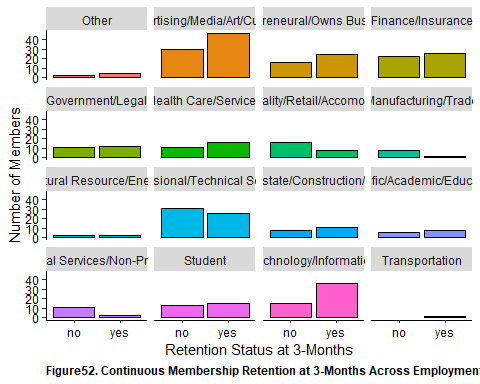
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, fill = age\_group)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 6-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(age\_group)) +  
 guides(fill = F) +   
 labs(caption = 'Figure50. Continuous Membership Retention at 6-Months Across Age Groups (χ2 = 8.47, p = 0.076)')



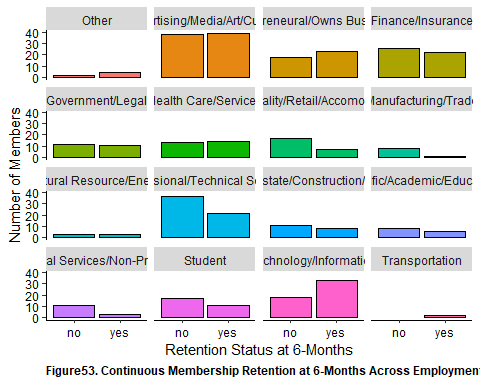
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, fill = age\_group)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 12-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(age\_group)) +  
 guides(fill = F) +   
 labs(caption = 'Figure51. Continuous Membership Retention at 12-Months Across Age Groups (χ2 = 7.92, p = 0.094)')



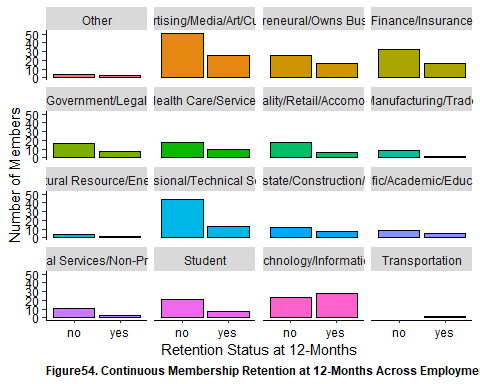
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, fill = employment\_sector)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 3-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure52. Continuous Membership Retention at 3-Months Across Employment Sectors (χ2 = 29.48, p = 0.014)')



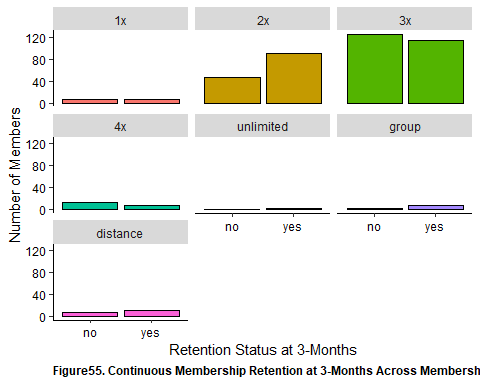
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, fill = employment\_sector)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 6-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure53. Continuous Membership Retention at 6-Months Across Employment Sectors (χ2 = 27.14, p = 0.028)')



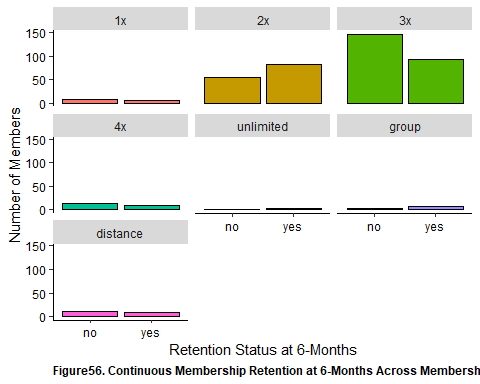
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, fill = employment\_sector)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 12-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(employment\_sector)) +  
 guides(fill = F) +   
 labs(caption = 'Figure54. Continuous Membership Retention at 12-Months Across Employment Sectors (χ2 = 22.96, p = 0.085)')



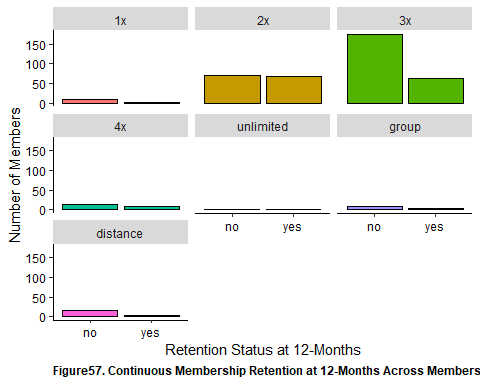
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, fill = membership)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 3-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(membership)) +  
 guides(fill = F) +   
 labs(caption = 'Figure55. Continuous Membership Retention at 3-Months Across Membership Types (χ2 = 16.99, p = 0.009)')



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, fill = membership)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 6-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(membership)) +  
 guides(fill = F) +   
 labs(caption = 'Figure56. Continuous Membership Retention at 6-Months Across Membership Types (χ2 = 18.40, p = 0.005)')



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, fill = membership)) +  
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line(color = 'black'),  
 axis.text = element\_text(color = 'black'),   
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Retention Status at 12-Months") +  
 ylab("Number of Members") +   
 facet\_wrap(vars(membership)) +  
 guides(fill = F) +   
 labs(caption = 'Figure57. Continuous Membership Retention at 12-Months Across Membership Types (χ2 = 22.02, p = 0.001)')



clean\_bang\_final %>% wilcox\_test(attendance\_rate ~ retention\_3m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 attendance\_rate no yes 203 244 20526. 0.00179 \*\*

clean\_bang\_final %>% wilcox\_test(attendance\_rate ~ retention\_6m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 attendance\_rate no yes 238 209 20674. 0.00203 \*\*

clean\_bang\_final %>% wilcox\_test(attendance\_rate ~ retention\_12m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 attendance\_rate no yes 296 151 18576. 0.00345 \*\*

kruskal.test(attendance\_rate[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention)

##   
## Kruskal-Wallis rank sum test  
##   
## data: attendance\_rate[retention\_status == "yes"] by retention\_type[retention\_status == "yes"]  
## Kruskal-Wallis chi-squared = 0.44316, df = 2, p-value = 0.8013

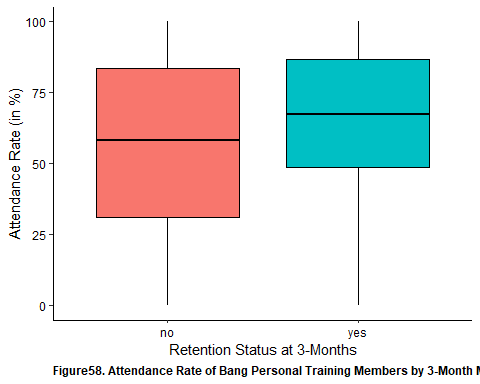
dunnTest(attendance\_rate[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention, method = 'holm')

## Dunn (1964) Kruskal-Wallis multiple comparison

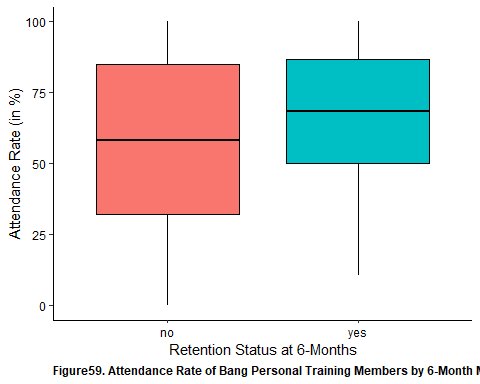
## p-values adjusted with the Holm method.

## Comparison Z P.unadj P.adj  
## 1 retention\_12m - retention\_3m 0.6618624 0.5080594 1.0000000  
## 2 retention\_12m - retention\_6m 0.4535289 0.6501679 1.0000000  
## 3 retention\_3m - retention\_6m -0.2131715 0.8311932 0.8311932

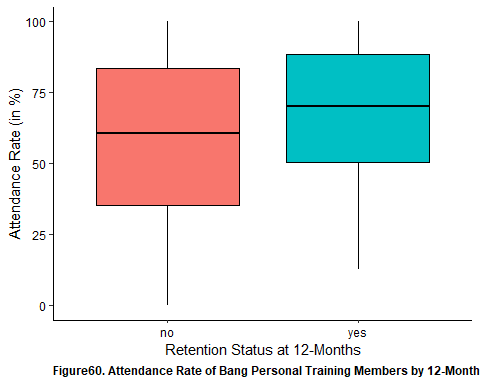
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m ,y = attendance\_rate, fill = retention\_3m)) +   
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 xlab("Retention Status at 3-Months") +  
 ylab("Attendance Rate (in %)") +   
 labs(caption = 'Figure58. Attendance Rate of Bang Personal Training Members by 3-Month Membership Retention Status (W = 20526, p = 0.002)') +  
 guides(fill = F)



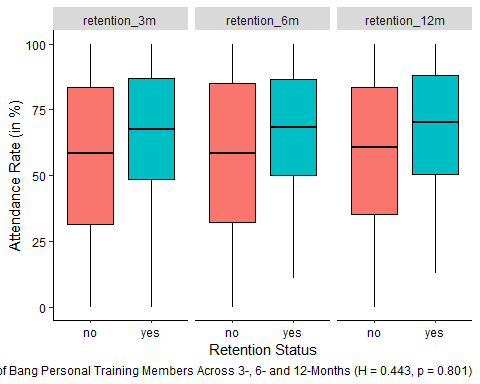
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m ,y = attendance\_rate, fill = retention\_6m)) +   
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 xlab("Retention Status at 6-Months") +  
 ylab("Attendance Rate (in %)") +   
 labs(caption = 'Figure59. Attendance Rate of Bang Personal Training Members by 6-Month Membership Retention Status (W = 20674, p = 0.002)') +  
 guides(fill = F)



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m ,y = attendance\_rate, fill = retention\_12m)) +   
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 xlab("Retention Status at 12-Months") +  
 ylab("Attendance Rate (in %)") +   
 labs(caption = 'Figure60. Attendance Rate of Bang Personal Training Members by 12-Month Membership Retention Status (W = 18576, p = 0.003)') +  
 guides(fill = F)



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, y = attendance\_rate, fill = retention\_status)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black')  
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Attendance Rate (in %)") +   
 labs(caption = "Figure61. Attendance Rate by Continuous Retention Status of Bang Personal Training Members Across 3-, 6- and 12-Months (H = 0.443, p = 0.801)")

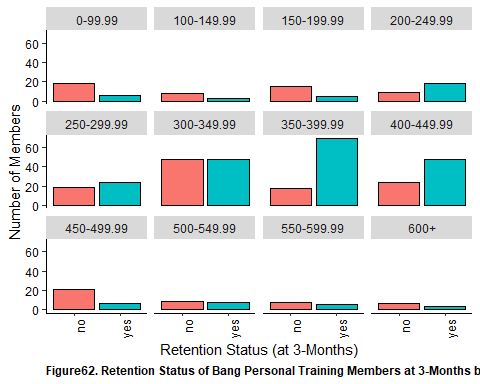


chisq.test(clean\_bang\_final$monthly\_rate\_group, clean\_bang\_final$retention\_3m)

## Warning in chisq.test(clean\_bang\_final$monthly\_rate\_group,  
## clean\_bang\_final$retention\_3m): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$monthly\_rate\_group and clean\_bang\_final$retention\_3m  
## X-squared = 61.448, df = 11, p-value = 4.986e-09

clean\_bang\_final %>%  
 ggplot(aes(x = retention\_3m, fill = retention\_3m)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0),   
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 facet\_wrap(vars(monthly\_rate\_group)) +  
 guides(fill = F) +  
 xlab("Retention Status (at 3-Months)") +   
 ylab("Number of Members") +   
 labs(caption = "Figure62. Retention Status of Bang Personal Training Members at 3-Months by Attendance Rate (χ2 = 61.45, p <0.001)")

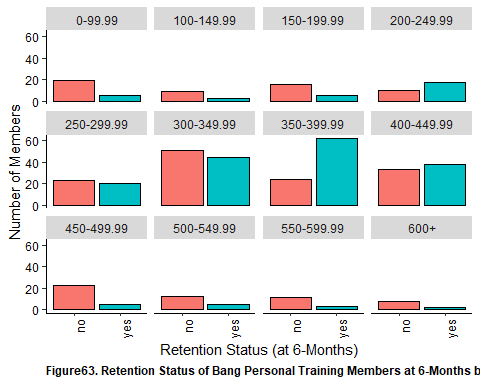


chisq.test(clean\_bang\_final$monthly\_rate\_group, clean\_bang\_final$retention\_6m)

## Warning in chisq.test(clean\_bang\_final$monthly\_rate\_group,  
## clean\_bang\_final$retention\_6m): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$monthly\_rate\_group and clean\_bang\_final$retention\_6m  
## X-squared = 58.152, df = 11, p-value = 2.04e-08

clean\_bang\_final %>%  
 ggplot(aes(x = retention\_6m, fill = retention\_6m)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0),   
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 facet\_wrap(vars(monthly\_rate\_group)) +  
 guides(fill = F) +  
 xlab("Retention Status (at 6-Months)") +   
 ylab("Number of Members") +   
 labs(caption = "Figure63. Retention Status of Bang Personal Training Members at 6-Months by Attendance Rate (χ2 = 58.15, p < 0.001)")

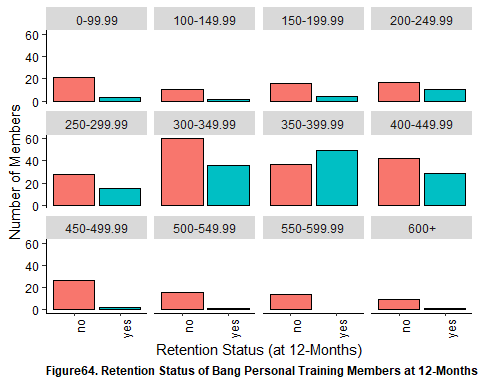


chisq.test(clean\_bang\_final$monthly\_rate\_group, clean\_bang\_final$retention\_12m)

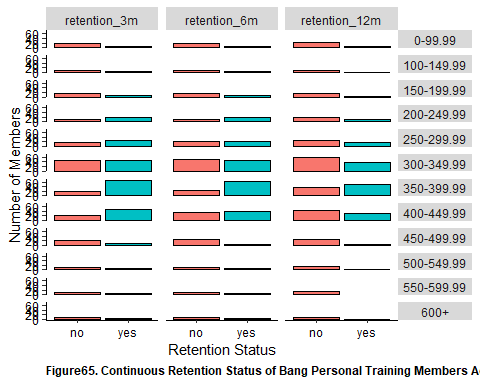
## Warning in chisq.test(clean\_bang\_final$monthly\_rate\_group,  
## clean\_bang\_final$retention\_12m): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$monthly\_rate\_group and clean\_bang\_final$retention\_12m  
## X-squared = 57.036, df = 11, p-value = 3.277e-08

clean\_bang\_final %>%  
 ggplot(aes(x = retention\_12m, fill = retention\_12m)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0),   
 axis.text.x = element\_text(angle = 90, hjust = 1)  
 ) +   
 facet\_wrap(vars(monthly\_rate\_group)) +  
 guides(fill = F) +  
 xlab("Retention Status (at 12-Months)") +   
 ylab("Number of Members") +   
 labs(caption = "Figure64. Retention Status of Bang Personal Training Members at 12-Months by Attendance Rate (χ2 = 57.04, p < 0.001)")



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, fill = retention\_status)) +  
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0),  
 strip.text.y = element\_text(angle = 0)   
 ) +   
 facet\_grid(  
 cols = vars(retention\_type),  
 rows = vars(monthly\_rate\_group)  
 ) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Number of Members") +   
 labs(caption = "Figure65. Continuous Retention Status of Bang Personal Training Members Across 3-, 6- and 12-Months by Monthly Membership Rates.")



chisq.test(clean\_bang\_final$retention\_3m, clean\_bang\_final$num\_billing\_issue)

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$retention\_3m and clean\_bang\_final$num\_billing\_issue  
## X-squared = 22.235, df = 2, p-value = 1.485e-05

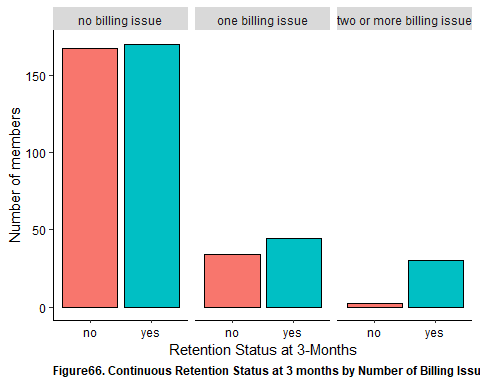
chisq.test(clean\_bang\_final$retention\_6m, clean\_bang\_final$num\_billing\_issue)

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$retention\_6m and clean\_bang\_final$num\_billing\_issue  
## X-squared = 26.074, df = 2, p-value = 2.178e-06

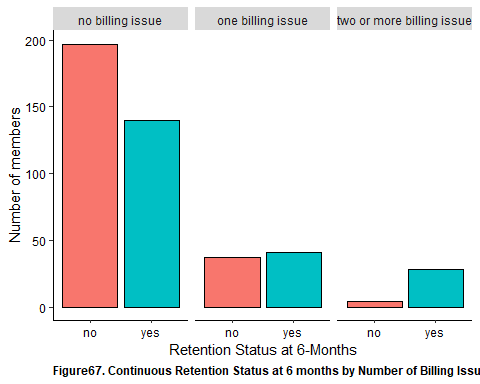
chisq.test(clean\_bang\_final$retention\_12m, clean\_bang\_final$num\_billing\_issue)

##   
## Pearson's Chi-squared test  
##   
## data: clean\_bang\_final$retention\_12m and clean\_bang\_final$num\_billing\_issue  
## X-squared = 29.337, df = 2, p-value = 4.262e-07

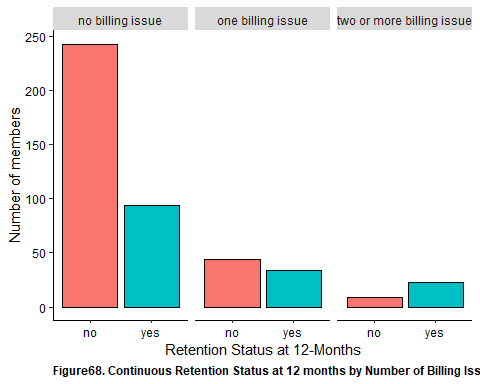
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, fill = retention\_3m)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +   
 facet\_wrap(vars(num\_billing\_issue)) +  
 xlab("Retention Status at 3-Months") +   
 ylab("Number of members") +   
 labs(caption = "Figure66. Continuous Retention Status at 3 months by Number of Billing Issue (χ2 = 22.24, p < 0.001)")



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, fill = retention\_6m)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +   
 facet\_wrap(vars(num\_billing\_issue)) +  
 xlab("Retention Status at 6-Months") +   
 ylab("Number of members") +   
 labs(caption = "Figure67. Continuous Retention Status at 6 months by Number of Billing Issue (χ2 = 26.07, p < 0.001)")



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, fill = retention\_12m)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +   
 facet\_wrap(vars(num\_billing\_issue)) +  
 xlab("Retention Status at 12-Months") +   
 ylab("Number of members") +   
 labs(caption = "Figure68. Continuous Retention Status at 12 months by Number of Billing Issue (χ2 = 29.34, p < 0.001)")



clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_cx ~ retention\_3m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_cx no yes 203 244 19284. 0.0000242 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_cx ~ retention\_6m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_cx no yes 238 209 19388 0.000025 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_cx ~ retention\_12m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_cx no yes 296 151 16706. 0.00000478 \*\*\*\*

kruskal.test(new\_per\_ticket\_cx[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention)

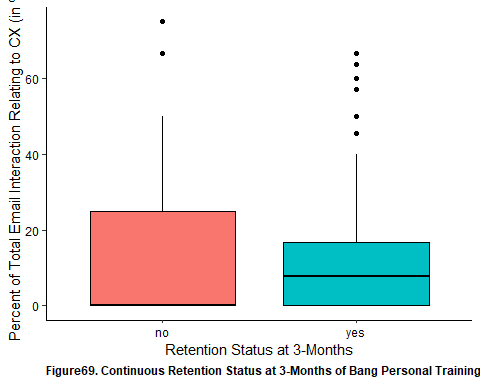
##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_per\_ticket\_cx[retention\_status == "yes"] by retention\_type[retention\_status == "yes"]  
## Kruskal-Wallis chi-squared = 0.95568, df = 2, p-value = 0.6201

dunnTest(new\_per\_ticket\_cx[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention, method = 'holm')

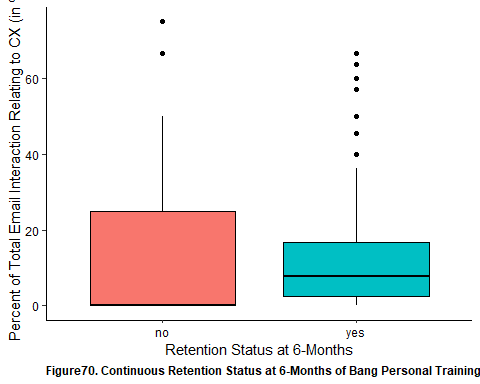
## Dunn (1964) Kruskal-Wallis multiple comparison  
## p-values adjusted with the Holm method.

## Comparison Z P.unadj P.adj  
## 1 retention\_12m - retention\_3m 0.9611324 0.3364856 1.0000000  
## 2 retention\_12m - retention\_6m 0.7186186 0.4723760 0.9447519  
## 3 retention\_3m - retention\_6m -0.2415444 0.8091332 0.8091332

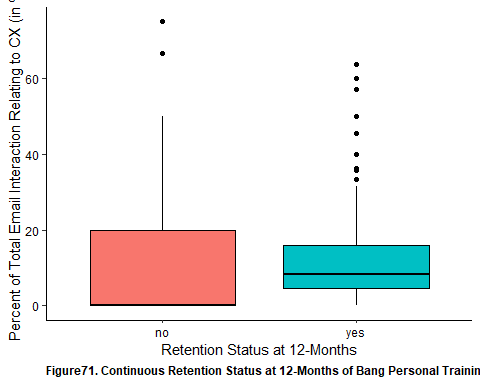
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, y = new\_per\_ticket\_cx, fill = retention\_3m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 3-Months") +  
 ylab("Percent of Total Email Interaction Relating to CX (in %)") +   
 labs(caption = 'Figure69. Continuous Retention Status at 3-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to CX (W = 19284.5, p < 0.001)')



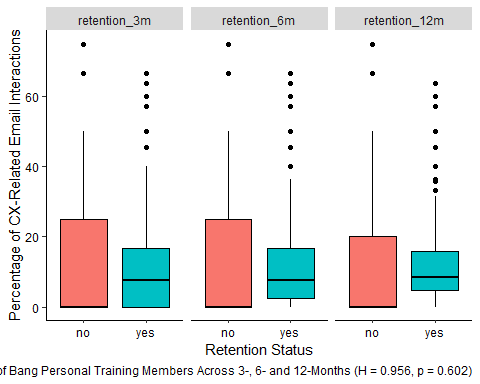
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, y = new\_per\_ticket\_cx, fill = retention\_6m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 6-Months") +  
 ylab("Percent of Total Email Interaction Relating to CX (in %)") +   
 labs(caption = 'Figure70. Continuous Retention Status at 6-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to CX (W = 19388, p < 0.001)')



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, y = new\_per\_ticket\_cx, fill = retention\_12m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 12-Months") +  
 ylab("Percent of Total Email Interaction Relating to CX (in %)") +   
 labs(caption = 'Figure71. Continuous Retention Status at 12-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to CX (W = 16706.5, p < 0.001)')



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, y = new\_per\_ticket\_cx, fill = retention\_status)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black')  
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Percentage of CX-Related Email Interactions") +   
 labs(caption = "Figure72. Percentage of Email Interactions Relating to CX by Continuous Retention Status of Bang Personal Training Members Across 3-, 6- and 12-Months (H = 0.956, p = 0.602)")



clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_scheduling ~ retention\_3m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_scheduli~ no yes 203 244 13837 3.52e-16 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_scheduling ~ retention\_6m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_scheduli~ no yes 238 209 12250. 5.66e-21 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_scheduling ~ retention\_12m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_scheduli~ no yes 296 151 9877 1.18e-22 \*\*\*\*

kruskal.test(new\_per\_ticket\_scheduling[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention)

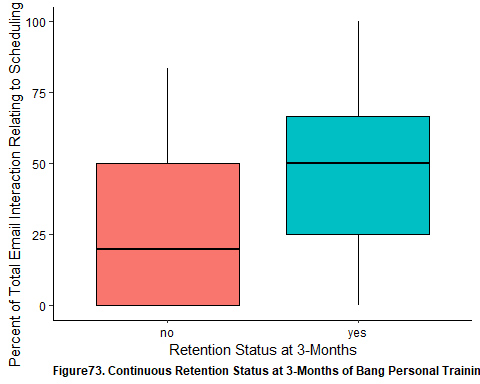
##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_per\_ticket\_scheduling[retention\_status == "yes"] by retention\_type[retention\_status == "yes"]  
## Kruskal-Wallis chi-squared = 8.4598, df = 2, p-value = 0.01455

dunnTest(new\_per\_ticket\_scheduling[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention, method = 'holm')

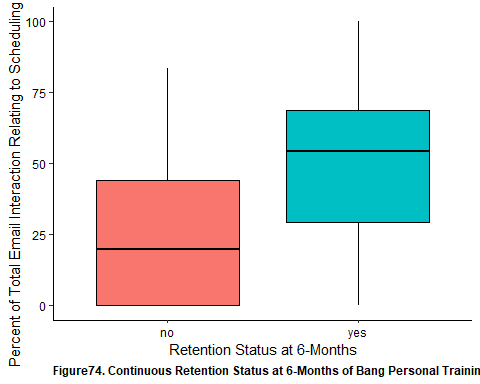
## Dunn (1964) Kruskal-Wallis multiple comparison  
## p-values adjusted with the Holm method.

## Comparison Z P.unadj P.adj  
## 1 retention\_12m - retention\_3m 2.907167 0.003647185 0.01094155  
## 2 retention\_12m - retention\_6m 1.668665 0.095183755 0.19036751  
## 3 retention\_3m - retention\_6m -1.302833 0.192631726 0.19263173

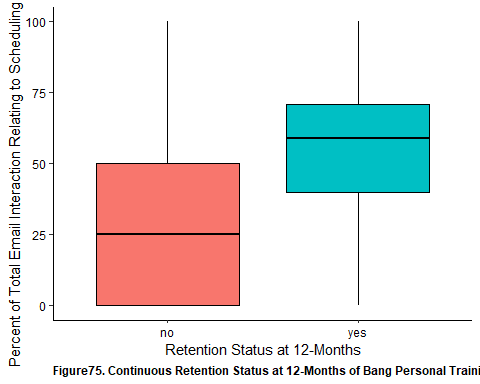
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, y = new\_per\_ticket\_scheduling, fill = retention\_3m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 3-Months") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = 'Figure73. Continuous Retention Status at 3-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to Scheduling (W = 13837, p < 0.001)')



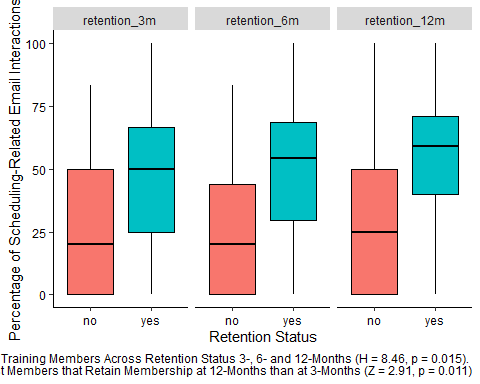
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, y = new\_per\_ticket\_scheduling, fill = retention\_6m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 6-Months") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = 'Figure74. Continuous Retention Status at 6-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to Scheduling (W = 12520.5, p < 0.001)')



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, y = new\_per\_ticket\_scheduling, fill = retention\_12m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 12-Months") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = 'Figure75. Continuous Retention Status at 12-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to Scheduling (W = 9877, p < 0.001)')



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, y = new\_per\_ticket\_scheduling, fill = retention\_status)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black')  
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Percentage of Scheduling-Related Email Interactions") +   
 labs(caption = "Figure76. Percentage of Email Interactions Relating to Scheduling by Continuous Retention Status of Bang Personal Training Members Across Retention Status 3-, 6- and 12-Months (H = 8.46, p = 0.015).   
 Following Pairwise Comparisons, Greater Proportion of Scheduling-Related Email Interactions Observed Amongst Members that Retain Membership at 12-Months than at 3-Months (Z = 2.91, p = 0.011)")



clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_service ~ retention\_3m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_service no yes 203 244 35410. 3.79e-15 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_service ~ retention\_6m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_service no yes 238 209 36710. 2.64e-18 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_per\_ticket\_service ~ retention\_12m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_per\_ticket\_service no yes 296 151 34180. 3.61e-20 \*\*\*\*

kruskal.test(new\_per\_ticket\_service[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention)

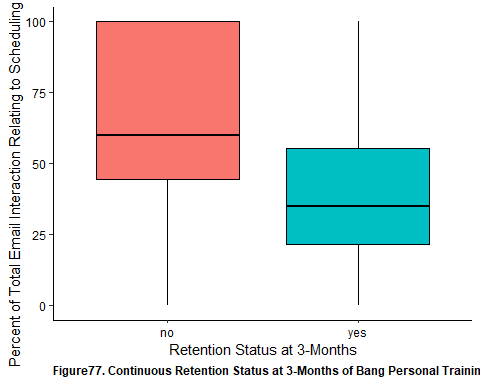
##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_per\_ticket\_service[retention\_status == "yes"] by retention\_type[retention\_status == "yes"]  
## Kruskal-Wallis chi-squared = 7.3388, df = 2, p-value = 0.02549

dunnTest(new\_per\_ticket\_service[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention, method = 'holm')

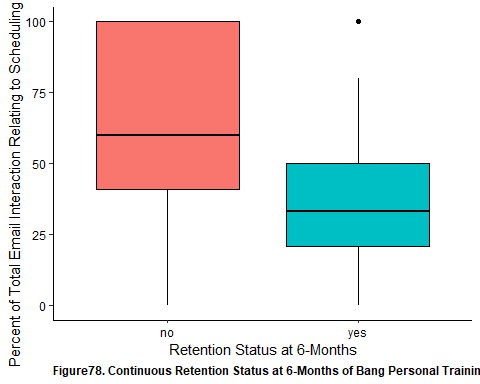
## Dunn (1964) Kruskal-Wallis multiple comparison  
## p-values adjusted with the Holm method.

## Comparison Z P.unadj P.adj  
## 1 retention\_12m - retention\_3m -2.707895 0.006771134 0.0203134  
## 2 retention\_12m - retention\_6m -1.684061 0.092169894 0.1843398  
## 3 retention\_3m - retention\_6m 1.066470 0.286211367 0.2862114

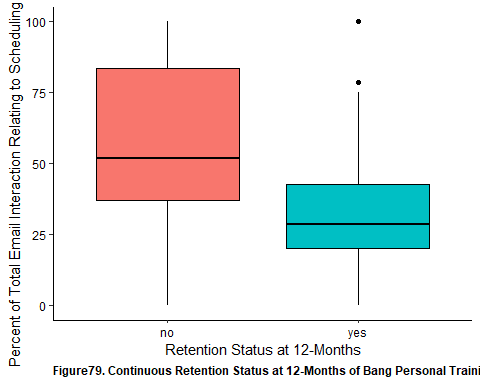
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, y = new\_per\_ticket\_service, fill = retention\_3m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 3-Months") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = 'Figure77. Continuous Retention Status at 3-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to Scheduling (W = 35410.5, p < 0.001)')



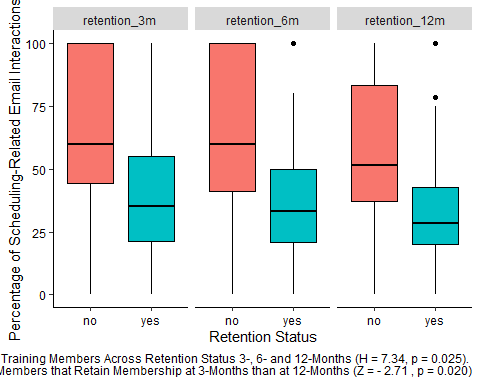
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, y = new\_per\_ticket\_service, fill = retention\_6m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 6-Months") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = 'Figure78. Continuous Retention Status at 6-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to Scheduling (W = 36710.5, p < 0.001)')



clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, y = new\_per\_ticket\_service, fill = retention\_12m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 12-Months") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = 'Figure79. Continuous Retention Status at 12-Months of Bang Personal Training Members By Percentage of Email Interactions Relating to Scheduling (W = 34179.5, p < 0.001)')



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, y = new\_per\_ticket\_service, fill = retention\_status)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black')  
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Percentage of Scheduling-Related Email Interactions") +   
 labs(caption = "Figure80. Percentage of Email Interactions Relating to Service by Continuous Retention Status of Bang Personal Training Members Across Retention Status 3-, 6- and 12-Months (H = 7.34, p = 0.025).   
 Following Pairwise Comparisons, Greater Proportion of Service-Related Email Interactions Observed Amongst Members that Retain Membership at 3-Months than at 12-Months (Z = - 2.71 , p = 0.020)")



clean\_bang\_final %>% wilcox\_test(new\_num\_total ~ retention\_3m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_num\_total no yes 203 244 6059 2.49e-43 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_num\_total ~ retention\_6m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_num\_total no yes 238 209 4900. 6.18e-49 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(new\_num\_total ~ retention\_12m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 new\_num\_total no yes 296 151 3364 3.34e-49 \*\*\*\*

kruskal.test(new\_num\_total[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention)

##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_num\_total[retention\_status == "yes"] by retention\_type[retention\_status == "yes"]  
## Kruskal-Wallis chi-squared = 21.333, df = 2, p-value = 2.331e-05

dunnTest(new\_num\_total[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention, method = 'holm')

## Dunn (1964) Kruskal-Wallis multiple comparison  
## p-values adjusted with the Holm method.

## Comparison Z P.unadj P.adj  
## 1 retention\_12m - retention\_3m 4.612334 3.981725e-06 1.194518e-05  
## 2 retention\_12m - retention\_6m 2.957483 3.101621e-03 6.203241e-03  
## 3 retention\_3m - retention\_6m -1.715611 8.623325e-02 8.623325e-02

clean\_bang\_final %>% wilcox\_test(num\_emails\_month ~ retention\_3m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 num\_emails\_month no yes 203 244 41870. 2.77e-36 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(num\_emails\_month ~ retention\_6m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 num\_emails\_month no yes 238 209 40174 2.91e-29 \*\*\*\*

clean\_bang\_final %>% wilcox\_test(num\_emails\_month ~ retention\_12m) %>% add\_significance()

## # A tibble: 1 x 8  
## .y. group1 group2 n1 n2 statistic p p.signif  
## <chr> <chr> <chr> <int> <int> <dbl> <dbl> <chr>   
## 1 num\_emails\_month no yes 296 151 32838 4.63e-16 \*\*\*\*

kruskal.test(num\_emails\_month[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention)

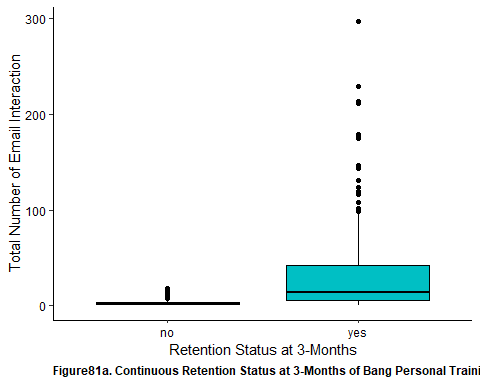
##   
## Kruskal-Wallis rank sum test  
##   
## data: num\_emails\_month[retention\_status == "yes"] by retention\_type[retention\_status == "yes"]  
## Kruskal-Wallis chi-squared = 0.15253, df = 2, p-value = 0.9266

dunnTest(num\_emails\_month[retention\_status == "yes"] ~ retention\_type[retention\_status == "yes"], data = clean\_bang\_longer\_retention, method = 'holm')

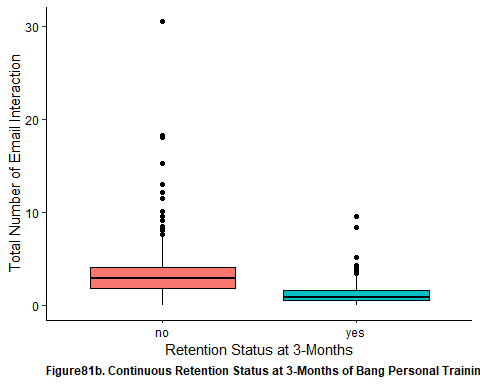
## Dunn (1964) Kruskal-Wallis multiple comparison  
## p-values adjusted with the Holm method.

## Comparison Z P.unadj P.adj  
## 1 retention\_12m - retention\_3m 0.04694502 0.9625570 0.962557  
## 2 retention\_12m - retention\_6m 0.33862690 0.7348908 1.000000  
## 3 retention\_3m - retention\_6m 0.33216064 0.7397680 1.000000

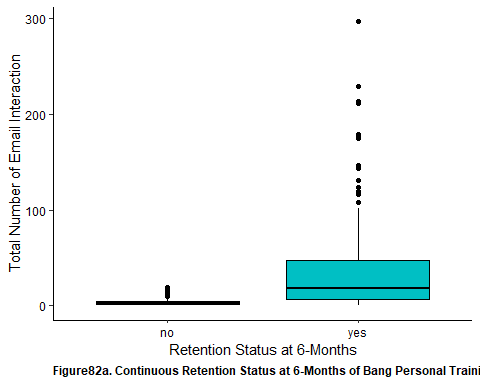
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, y = new\_num\_total, fill = retention\_3m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 3-Months") +  
 ylab("Total Number of Email Interaction") +   
 labs(caption = 'Figure81a. Continuous Retention Status at 3-Months of Bang Personal Training Members By Total Number of Non-Billing Email Interactions (W = 6059, p < 0.001)')



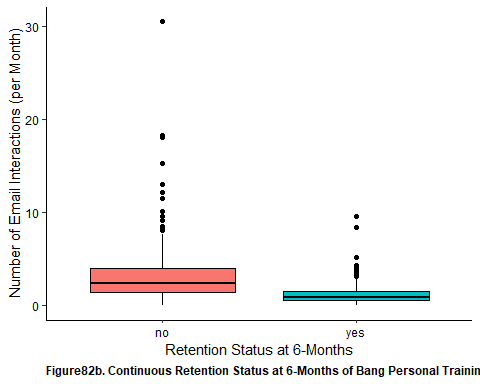
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_3m, y = num\_emails\_month, fill = retention\_3m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 3-Months") +  
 ylab("Total Number of Email Interaction") +   
 labs(caption = 'Figure81b. Continuous Retention Status at 3-Months of Bang Personal Training Members By Number of Non-Billing Email Interactions per Month (W = 40874, p < 0.001)')



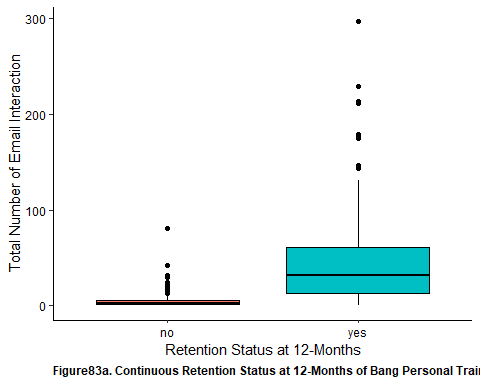
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, y = new\_num\_total, fill = retention\_6m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 6-Months") +  
 ylab("Total Number of Email Interaction") +   
 labs(caption = 'Figure82a. Continuous Retention Status at 6-Months of Bang Personal Training Members By Total Number of Non-Billing Email Interactions (W = 4900.5, p < 0.001)')



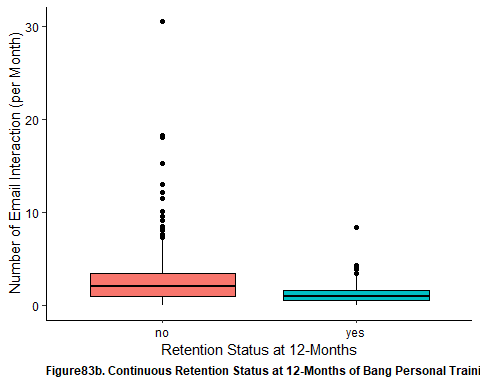
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_6m, y = num\_emails\_month, fill = retention\_6m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 6-Months") +  
 ylab("Number of Email Interactions (per Month)") +   
 labs(caption = 'Figure82b. Continuous Retention Status at 6-Months of Bang Personal Training Members By Number of Non-Billing Email Interactions per Month (W = 40174, p < 0.001)')



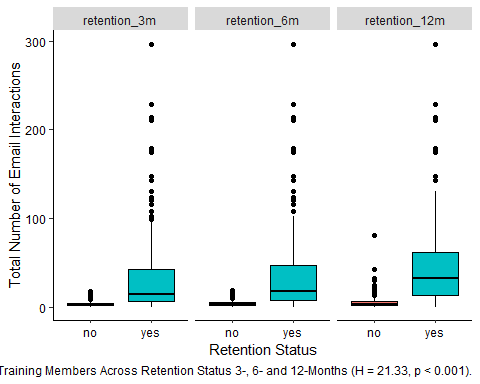
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, y = new\_num\_total, fill = retention\_12m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 12-Months") +  
 ylab("Total Number of Email Interaction") +   
 labs(caption = 'Figure83a. Continuous Retention Status at 12-Months of Bang Personal Training Members By Total Number of Non-Billing Email Interactions (W = 3364, p < 0.001)')



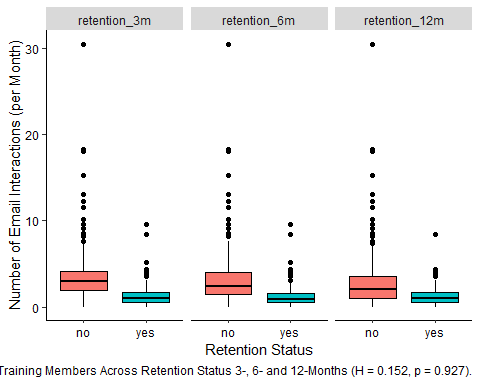
clean\_bang\_final %>%   
 ggplot(aes(x = retention\_12m, y = num\_emails\_month, fill = retention\_12m)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.line = element\_line(color = 'black'),   
 axis.text = element\_text(color = 'black'),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 guides(fill = F) +  
 xlab("Retention Status at 12-Months") +  
 ylab("Number of Email Interaction (per Month)") +   
 labs(caption = 'Figure83b. Continuous Retention Status at 12-Months of Bang Personal Training Members By Number of Non-Billing Email Interactions (w = 32838, p < 0.001)')



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, new\_num\_total, fill = retention\_status)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black')  
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Total Number of Email Interactions") +   
 labs(caption = "Figure84a. Number ofNon-Billing Email Interactions by Continuous Retention Status of Bang Personal Training Members Across Retention Status 3-, 6- and 12-Months (H = 21.33, p < 0.001).")



clean\_bang\_longer\_retention %>%   
 ggplot(aes(x = retention\_status, num\_emails\_month, fill = retention\_status)) +  
 geom\_boxplot(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),   
 axis.text = element\_text(color = 'black'),   
 axis.line = element\_line(color = 'black')  
 ) +   
 facet\_wrap(vars(retention\_type)) +  
 guides(fill = F) +   
 xlab("Retention Status") +   
 ylab("Number of Email Interactions (per Month)") +   
 labs(caption = "Figure84b. Number of Non-Billing Email Interactions per Month by Continuous Retention Status of Bang Personal Training Members Across Retention Status 3-, 6- and 12-Months (H = 0.152, p = 0.927).")

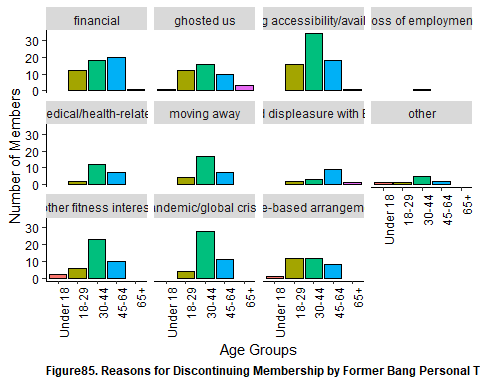


## Reason to Leave

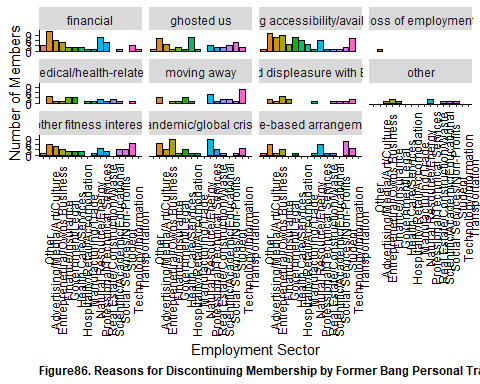
Examining past members, it was found that there were significant differences in rationale for leaving Bang Personal Training across age groups and membership types. Whilst lack of accessibility or availability in schedule was the most commonly cited reason amongst those aged 30-44, financial cost of the membership was found to join this rationale as the most commonly cited reason for those aged 45-64. However for those aged 18-29, time-based arrangement was commonly cited as the most prevalent reason to leave Bang Personal Training. In terms of membership types, those that were in the popular 3x/week membership were found to have left due to lack of availability or accessibility to use the membership. Although this was the most commonly cited reason, those that were in the 2x/week membership also cited moving away or desire to pursue other fitness interest for discontinuing membership. Interestingly enough, those with group memberships predominantly left due to the 2020 Pandemic.

Significant differences in citing reasons to discontinue membership were also noted across various attendance rates and average monthly rates. Notably, those that often cite financial cost or a time-based arrangement as a reason to discontinue membership tend to have the highest attendance rate. On the otherhand, those that had ghosted us or had cited lack of accessibility or availability tend to attend their appointments less than 50% of the time. As it relates to monthly membership rates, those citing lacking accessibility/availability tend to have higher membership rates compared to other cited reasons. Lastly, examining email interactions, it was found that there were significant differences with respect to both various types of email interactions.

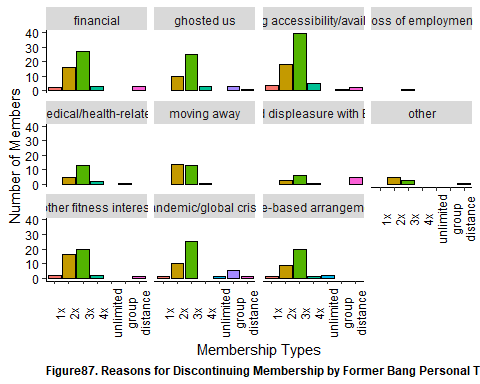
former\_bang %>%   
 ggplot(aes(x = age\_group, fill = age\_group)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text(angle = 90, hjust = 1),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Age Groups") +  
 ylab("Number of Members") +  
 facet\_wrap(vars(reason\_to\_leave)) +   
 labs(caption = "Figure85. Reasons for Discontinuing Membership by Former Bang Personal Training Members Distributed Across Age Groups (χ2 = 58.25, p = 0.031)") +   
 guides(fill = F)



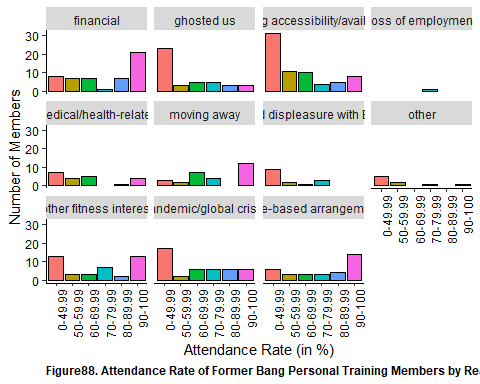
former\_bang %>%   
 ggplot(aes(x = employment\_sector, fill = employment\_sector)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Employment Sector") +  
 ylab("Number of Members") +  
 facet\_wrap(vars(reason\_to\_leave)) +   
 labs(caption = "Figure86. Reasons for Discontinuing Membership by Former Bang Personal Training Members Distributed Across Employment Sector (χ2 = 170.04, p = 0.126)") +  
 guides(fill = F)



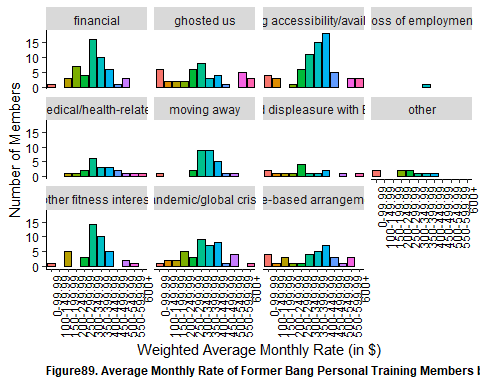
former\_bang %>%   
 ggplot(aes(x = membership, fill = membership)) +   
 geom\_bar(color = 'black') +  
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 strip.text.y = element\_text(angle = 0),  
 axis.text.x = element\_text (angle = 90),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Membership Types") +  
 ylab("Number of Members") +  
 facet\_wrap(vars(reason\_to\_leave)) +   
 labs(caption = "Figure87. Reasons for Discontinuing Membership by Former Bang Personal Training Members Distributed Across Membership Types (χ2 = 87.41, p = 0.012)") +   
 guides(fill = F)



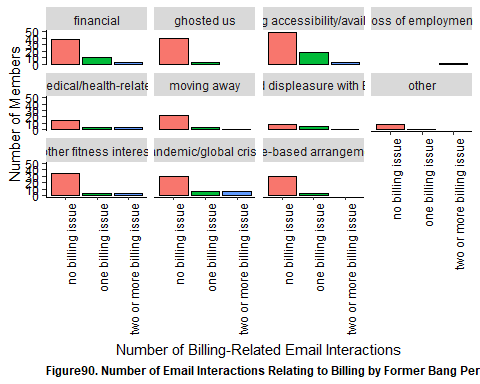
former\_bang %>%   
 ggplot(aes(x = attendance\_grouping\_ver.1, fill = attendance\_grouping\_ver.1)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90, hjust = 1),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 facet\_wrap(vars(reason\_to\_leave)) +  
 xlab("Attendance Rate (in %)") +  
 ylab("Number of Members") +   
 guides(fill = F) +  
 labs(caption = "Figure88. Attendance Rate of Former Bang Personal Training Members by Reasons to Discontinue Membership (χ2 = 90.12, p < 0.001)")



former\_bang %>%   
 ggplot(aes(x = monthly\_rate\_group, fill = monthly\_rate\_group)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90, hjust = 1),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 facet\_wrap(vars(reason\_to\_leave)) +  
 xlab("Weighted Average Monthly Rate (in $)") +  
 ylab("Number of Members") +   
 guides(fill = F) +  
 labs(caption = "Figure89. Average Monthly Rate of Former Bang Personal Training Members by Reasons to Discontinue Membership (χ2 = 140.11, p < 0.028)")



former\_bang %>%   
 ggplot(aes(x = num\_billing\_issue, fill = num\_billing\_issue)) +   
 geom\_bar(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90, hjust = 1),  
 strip.text.y = element\_text(angle = 0),  
 plot.caption = element\_text(face = 'bold', hjust = 0)   
 ) +   
 facet\_wrap(vars(reason\_to\_leave)) +  
 xlab("Number of Billing-Related Email Interactions") +  
 ylab("Number of Members") +   
 guides(fill = F) +  
 labs(caption = "Figure90. Number of Email Interactions Relating to Billing by Former Bang Personal Training Members by Reasons to Discontinue Membership (χ2 = 49.80, p < 0.001)")



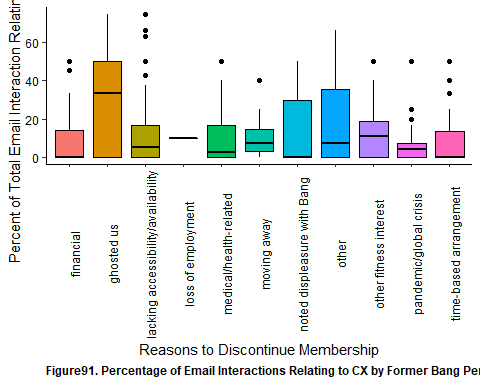
kruskal.test(new\_per\_ticket\_cx ~ reason\_to\_leave, data = former\_bang)

##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_per\_ticket\_cx by reason\_to\_leave  
## Kruskal-Wallis chi-squared = 19.63, df = 10, p-value = 0.03295

dunn\_test(new\_per\_ticket\_cx ~ reason\_to\_leave, data = former\_bang, p.adjust.method = 'holm')

## # A tibble: 55 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 new\_per~ finan~ ghosted us 51 42 3.36 7.90e-4 0.0434 \*   
## 2 new\_per~ finan~ lacking ac~ 51 69 1.34 1.81e-1 1 ns   
## 3 new\_per~ finan~ loss of em~ 51 1 0.650 5.16e-1 1 ns   
## 4 new\_per~ finan~ medical/he~ 51 21 0.833 4.05e-1 1 ns   
## 5 new\_per~ finan~ moving away 51 28 1.57 1.17e-1 1 ns   
## 6 new\_per~ finan~ noted disp~ 51 15 0.390 6.96e-1 1 ns   
## 7 new\_per~ finan~ other 51 9 1.64 1.02e-1 1 ns   
## 8 new\_per~ finan~ other fitn~ 51 41 1.85 6.36e-2 1 ns   
## 9 new\_per~ finan~ pandemic/g~ 51 43 -0.0701 9.44e-1 1 ns   
## 10 new\_per~ finan~ time-based~ 51 33 -0.0628 9.50e-1 1 ns   
## # ... with 45 more rows

former\_bang %>%   
 ggplot(aes(x = reason\_to\_leave, y = new\_per\_ticket\_cx, fill = reason\_to\_leave)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Reasons to Discontinue Membership") +  
 ylab("Percent of Total Email Interaction Relating to CX (in %)") +   
 labs(caption = "Figure91. Percentage of Email Interactions Relating to CX by Former Bang Personal Training Members by Reasons to Discontinue Membership (W = 19.63, p = 0.032). Pairwise comparisons saw signicant lower percentage of CX-related Email Interactions between those that left Bang due to financial cost and those that left as a result of moving away (Z = 3.36, p = 0.043) ") +  
 guides(fill = F)



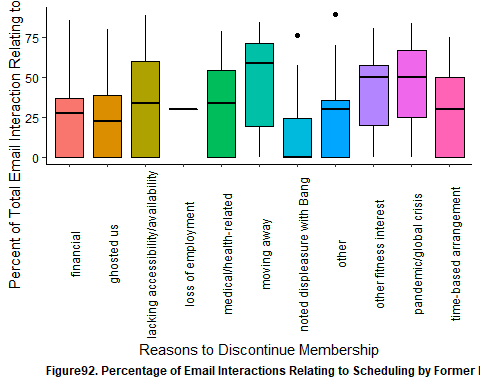
kruskal.test(new\_per\_ticket\_scheduling ~ reason\_to\_leave, data = former\_bang)

##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_per\_ticket\_scheduling by reason\_to\_leave  
## Kruskal-Wallis chi-squared = 32.769, df = 10, p-value = 0.0002978

dunn\_test(new\_per\_ticket\_scheduling ~ reason\_to\_leave, data = former\_bang, p.adjust.method = 'holm')

## # A tibble: 55 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 new\_per\_~ financ~ ghosted us 51 42 -0.558 0.577 1 ns   
## 2 new\_per\_~ financ~ lacking a~ 51 69 2.05 0.0405 1 ns   
## 3 new\_per\_~ financ~ loss of e~ 51 1 0.0311 0.975 1 ns   
## 4 new\_per\_~ financ~ medical/h~ 51 21 0.807 0.420 1 ns   
## 5 new\_per\_~ financ~ moving aw~ 51 28 3.04 0.00238 0.119 ns   
## 6 new\_per\_~ financ~ noted dis~ 51 15 -1.35 0.178 1 ns   
## 7 new\_per\_~ financ~ other 51 9 0.163 0.870 1 ns   
## 8 new\_per\_~ financ~ other fit~ 51 41 2.50 0.0124 0.545 ns   
## 9 new\_per\_~ financ~ pandemic/~ 51 43 2.99 0.00284 0.139 ns   
## 10 new\_per\_~ financ~ time-base~ 51 33 0.263 0.793 1 ns   
## # ... with 45 more rows

former\_bang %>%   
 ggplot(aes(x = reason\_to\_leave, y = new\_per\_ticket\_scheduling, fill = reason\_to\_leave)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Reasons to Discontinue Membership") +  
 ylab("Percent of Total Email Interaction Relating to Scheduling (in %)") +   
 labs(caption = "Figure92. Percentage of Email Interactions Relating to Scheduling by Former Bang Personal Training Members by Reasons to Discontinue Membership (W = 32.77, p < 0.001)") +  
 guides(fill = F)



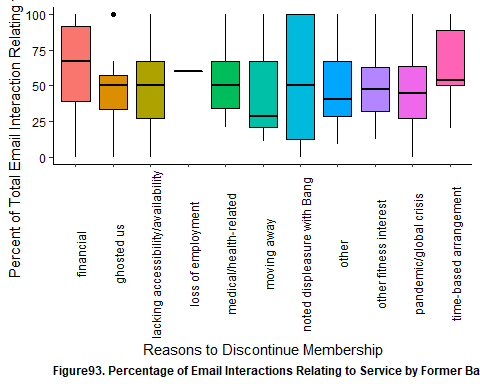
kruskal.test(new\_per\_ticket\_service ~ reason\_to\_leave, data = former\_bang)

##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_per\_ticket\_service by reason\_to\_leave  
## Kruskal-Wallis chi-squared = 20.452, df = 10, p-value = 0.02526

dunn\_test(new\_per\_ticket\_service ~ reason\_to\_leave, data = former\_bang, p.adjust.method = 'holm')

## # A tibble: 55 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 new\_per~ financ~ ghosted us 51 42 -2.15 0.0314 1 ns   
## 2 new\_per~ financ~ lacking a~ 51 69 -2.79 0.00531 0.281 ns   
## 3 new\_per~ financ~ loss of e~ 51 1 0.0778 0.938 1 ns   
## 4 new\_per~ financ~ medical/h~ 51 21 -0.825 0.410 1 ns   
## 5 new\_per~ financ~ moving aw~ 51 28 -3.18 0.00146 0.0804 ns   
## 6 new\_per~ financ~ noted dis~ 51 15 -1.46 0.143 1 ns   
## 7 new\_per~ financ~ other 51 9 -1.38 0.166 1 ns   
## 8 new\_per~ financ~ other fit~ 51 41 -2.49 0.0128 0.655 ns   
## 9 new\_per~ financ~ pandemic/~ 51 43 -2.66 0.00771 0.401 ns   
## 10 new\_per~ financ~ time-base~ 51 33 0.00472 0.996 1 ns   
## # ... with 45 more rows

former\_bang %>%   
 ggplot(aes(x = reason\_to\_leave, y = new\_per\_ticket\_service, fill = reason\_to\_leave)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Reasons to Discontinue Membership") +  
 ylab("Percent of Total Email Interaction Relating to Service (in %)") +   
 labs(caption = "Figure93. Percentage of Email Interactions Relating to Service by Former Bang Personal Training Members by Reasons to Discontinue Membership (W = 20.45, p = 0.025)") +  
 guides(fill = F)



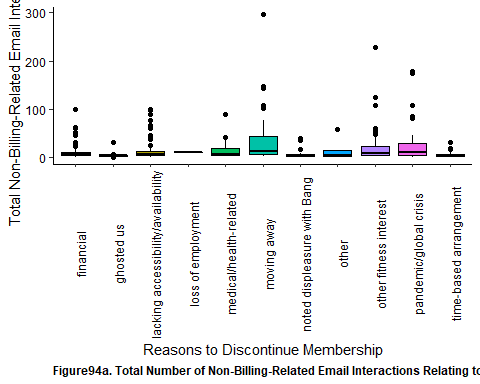
kruskal.test(new\_num\_total ~ reason\_to\_leave, data = former\_bang)

##   
## Kruskal-Wallis rank sum test  
##   
## data: new\_num\_total by reason\_to\_leave  
## Kruskal-Wallis chi-squared = 58.121, df = 10, p-value = 8.203e-09

dunn\_test(new\_num\_total ~ reason\_to\_leave, data = former\_bang, p.adjust.method = 'holm')

## # A tibble: 55 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 new\_nu~ financ~ ghosted us 51 42 -3.39 7.04e-4 0.0338 \*   
## 2 new\_nu~ financ~ lacking ac~ 51 69 0.783 4.34e-1 1 ns   
## 3 new\_nu~ financ~ loss of em~ 51 1 0.760 4.47e-1 1 ns   
## 4 new\_nu~ financ~ medical/he~ 51 21 0.901 3.68e-1 1 ns   
## 5 new\_nu~ financ~ moving away 51 28 2.88 3.97e-3 0.171 ns   
## 6 new\_nu~ financ~ noted disp~ 51 15 -1.77 7.64e-2 1 ns   
## 7 new\_nu~ financ~ other 51 9 -0.0636 9.49e-1 1 ns   
## 8 new\_nu~ financ~ other fitn~ 51 41 2.01 4.40e-2 1 ns   
## 9 new\_nu~ financ~ pandemic/g~ 51 43 2.26 2.36e-2 0.944 ns   
## 10 new\_nu~ financ~ time-based~ 51 33 -1.16 2.45e-1 1 ns   
## # ... with 45 more rows

former\_bang %>%   
 ggplot(aes(x = reason\_to\_leave, y = new\_num\_total, fill = reason\_to\_leave)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Reasons to Discontinue Membership") +  
 ylab("Total Non-Billing-Related Email Interactions") +   
 labs(caption = "Figure94a. Total Number of Non-Billing-Related Email Interactions Relating to Service by Former Bang Personal Training Members by Reasons to Discontinue Membership (W = 58.12, p < 0.001)") +  
 guides(fill = F)



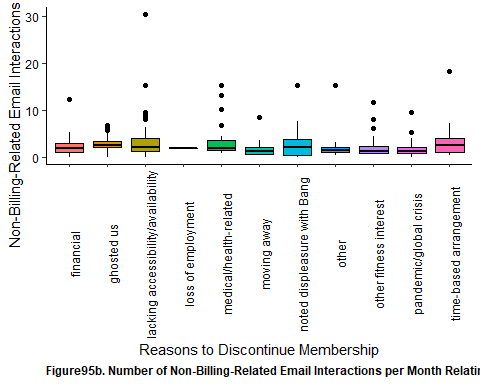
kruskal.test(num\_emails\_month ~ reason\_to\_leave, data = former\_bang)

##   
## Kruskal-Wallis rank sum test  
##   
## data: num\_emails\_month by reason\_to\_leave  
## Kruskal-Wallis chi-squared = 31.534, df = 10, p-value = 0.0004787

dunn\_test(num\_emails\_month ~ reason\_to\_leave, data = former\_bang, p.adjust.method = 'holm')

## # A tibble: 55 x 9  
## .y. group1 group2 n1 n2 statistic p p.adj p.adj.signif  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl> <dbl> <chr>   
## 1 num\_ema~ financ~ ghosted us 51 42 2.34 0.0195 0.898 ns   
## 2 num\_ema~ financ~ lacking acc~ 51 69 1.63 0.102 1 ns   
## 3 num\_ema~ financ~ loss of emp~ 51 1 0.0525 0.958 1 ns   
## 4 num\_ema~ financ~ medical/hea~ 51 21 1.21 0.227 1 ns   
## 5 num\_ema~ financ~ moving away 51 28 -1.87 0.0612 1 ns   
## 6 num\_ema~ financ~ noted displ~ 51 15 -0.0886 0.929 1 ns   
## 7 num\_ema~ financ~ other 51 9 -0.517 0.605 1 ns   
## 8 num\_ema~ financ~ other fitne~ 51 41 -0.918 0.359 1 ns   
## 9 num\_ema~ financ~ pandemic/gl~ 51 43 -1.44 0.150 1 ns   
## 10 num\_ema~ financ~ time-based ~ 51 33 1.51 0.130 1 ns   
## # ... with 45 more rows

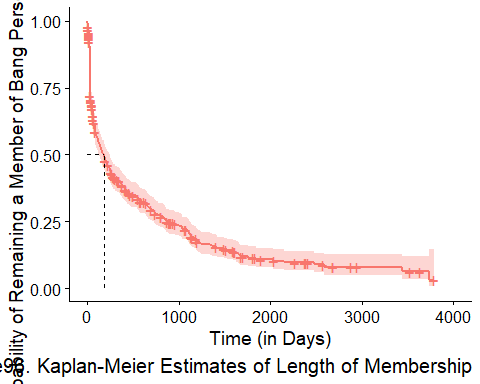
former\_bang %>%   
 ggplot(aes(x = reason\_to\_leave, y = num\_emails\_month, fill = reason\_to\_leave)) +   
 geom\_boxplot(color = 'black') +   
 theme(  
 panel.background = element\_rect(fill = 'white'),  
 axis.line = element\_line (color = 'black'),  
 axis.text = element\_text (color = 'black'),  
 axis.text.x = element\_text (angle = 90),  
 plot.caption = element\_text(face = 'bold', hjust = 0)  
 ) +   
 xlab("Reasons to Discontinue Membership") +  
 ylab("Non-Billing-Related Email Interactions (per Month)") +   
 labs(caption = "Figure95b. Number of Non-Billing-Related Email Interactions per Month Relating to Service by Former Bang Personal Training Members by Reasons to Discontinue Membership (W = 31.53, p < 0.001). Following pairwise comparisons, this was particularly noted with respect to those that lacked availability reporing more email interactions than those that moved away (Z = - 3.31, p = 0.049), along with those that had 'ghosted us' with more email interactions than those that moved away (Z = -3.80, p = 0.008) and those that had left due to the Pandemic (Z = -3.61, p = 0.016)") +  
 guides(fill = F)



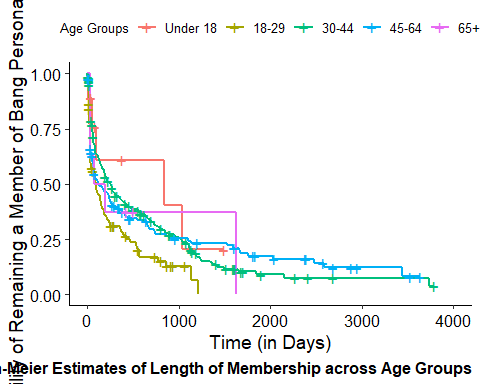
## Modelling Length of Bang Personal Training Membership

Examining the length of membership of members, it was found to range from as low as 2 days to as much as 3790 days with the median duration length being around 4.5 months. Based on the kaplan meier curves, it was suspected that age, employment sector, membership, attendance rate, average monthly rate and number of billing issues.

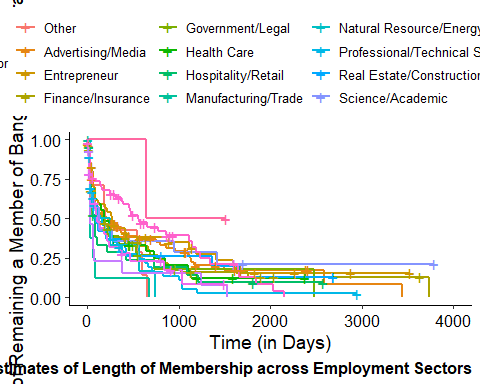
ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ 1, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 surv.median.line = 'hv',  
 caption = "Figure96. Kaplan-Meier Estimates of Length of Membership"  
 )



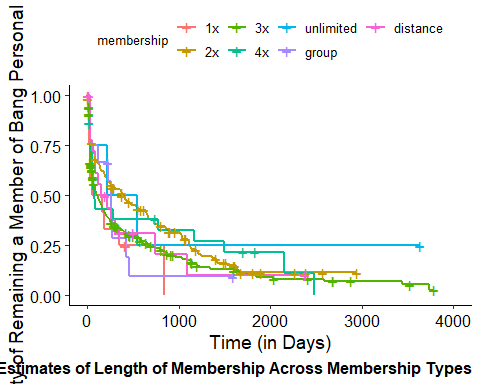
ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ age\_group, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 legend.title = 'Age Groups',  
 legend.labs = c("Under 18", "18-29", "30-44", "45-64", "65+"),  
 caption = "Figure97. Kaplan-Meier Estimates of Length of Membership across Age Groups"  
 ) +   
 theme\_survminer(  
 font.caption = c(12, 'bold', 'black')  
 )



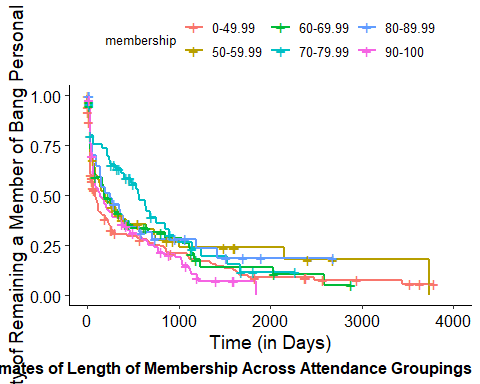
ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ employment\_sector, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 legend.title = 'Employment Sector',  
 legend.labs = c("Other", "Advertising/Media", "Entrepreneur", "Finance/Insurance",  
 "Government/Legal", "Health Care", "Hospitality/Retail", "Manufacturing/Trade",  
 "Natural Resource/Energy", "Professional/Technical Services", "Real Estate/Construction",  
 "Science/Academic", "Social Services", "Student", "Technology", "Transportation"),  
 caption = "Figure98. Kaplan-Meier Estimates of Length of Membership across Employment Sectors"  
 ) +   
 theme\_survminer(  
 font.caption = c(12, 'bold', 'black')  
 )



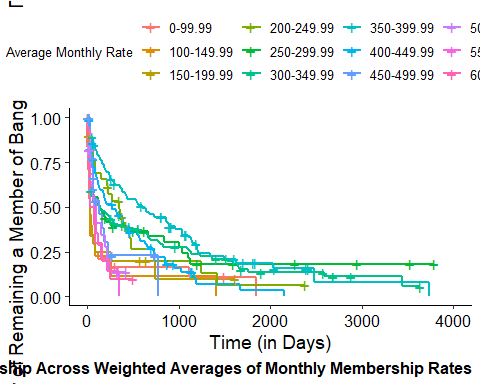
ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ membership, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 legend.title = 'membership',  
 legend.labs = c("1x", "2x", "3x", "4x", "unlimited", "group", "distance"),  
 caption = "Figure99. Kaplan-Meier Estimates of Length of Membership Across Membership Types"  
 ) +   
 theme\_survminer(  
 font.caption = c(12, 'bold', 'black')  
 )



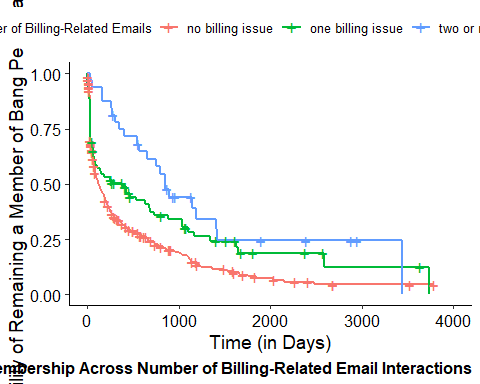
ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ attendance\_grouping\_ver.1, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 legend.title = 'membership',  
 legend.labs = c("0-49.99", "50-59.99", "60-69.99", "70-79.99", "80-89.99", "90-100"),  
 caption = "Figure100. Kaplan-Meier Estimates of Length of Membership Across Attendance Groupings"  
 ) +   
 theme\_survminer(  
 font.caption = c(12, 'bold', 'black')  
 )



ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ monthly\_rate\_group, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 legend.title = 'Average Monthly Rate',  
 legend.labs = c('0-99.99', '100-149.99', '150-199.99', '200-249.99', '250-299.99', '300-349.99', '350-399.99', '400-449.99', '450-499.99', '500-549.99', '550-599.99', '600+'),  
 caption = "Figure101. Kaplan-Meier Estimates of Length of Membership Across Weighted Averages of Monthly Membership Rates"  
 ) +   
 theme\_survminer(  
 font.caption = c(12, 'bold', 'black')  
 )



ggsurvplot(  
 fit = survfit(Surv(length, became\_former\_member) ~ num\_billing\_issue, data = clean\_bang\_select),  
 xlab = "Time (in Days)",  
 ylab = "Probability of Remaining a Member of Bang Personal Training",  
 legend = 'none',  
 legend.title = 'Number of Billing-Related Emails',  
 legend.labs = c('no billing issue', 'one billing issue', 'two or more billing issue'),  
 caption = "Figure102. Kaplan-Meier Estimates of Length of Membership Across Number of Billing-Related Email Interactions"  
 ) +   
 theme\_survminer(  
 font.caption = c(12, 'bold', 'black')  
 )



Seeing as I would like to gain some insight on how certain variable in our data set would play a role in predicting length of membership, this would in effect be performing a survival analysis. As such, I would be approaching this through two means: one is through Random Survival Forest (data science~y way) and the other is through the Cox Regression Proportional Hazard model. For the second case, as there will be assumptions that needs to be satisfied in order for the model to be “valid”, several tests will need to be conducted to ensure this (as noted in this [YouTube Video](https://www.youtube.com/watch?v=QAgtZKpKj9M&ab_channel=MarinStatsLectures-RProgramming%26Statistics)). As such, there will likely be the use of log-transformation or categorization of certain numeric variables that will be included into this model.

NOTE: For reference: random survival forest was modelled similarly as shown [here](https://www.youtube.com/watch?v=LsCUXzAxDXw&ab_channel=VamsidharAmbatipudi)

### CHURN ANALYSIS - RANDOM SURVIVAL FOREST

In developing the Random Survival Forest, I’ll need the dataset to only include variables that I would like to be tested to explain outcomes from happening. Thus a separate dataset will need to be created that contains only the variables that are of interest to us to determine churn outcomes. This will include: \* age\_group \* employment\_sector \* became\_former\_member (necessary for censoring) \* length (necessary as dependent variable) \* membership \* monthly\_rate\_group \* attendance\_grouping\_ver.1 \* num\_emails\_month \* ever\_emails\_month \* ever\_billing\_issue \* new\_per\_ticket\_cx \* ever\_cx \* new\_per\_ticket\_scheduling \* ever\_scheduling \* new\_per\_ticket\_service \* ever\_service

Using the random survival forest specific dataset, I’ve split the data set 80:20 with respect to training:test. In forming the training model, which has an error rate of **16.95%**, it was found that the error rate in predicting membership length to churn with the test data was **17.59%** (NOTE: this would be equivalent to the C-index via 1 - error rate, so 0.8241), so overall an OK model. Looking at the various ways to modify the parameters, it was found that the error rate more-or-less stabilized after 1000 trees as evident by the marginal differences in error rates at the higher number of trees. Similarly, findings were found with respect to mtry with the largest being between 7 and the default of 3.

Examining the importance of all of the testable predictors in impacting the outcome of predicting churn, it was found that the **number of non-billing email interactions per month** played the largest role, followed by ever having a CX-related email interaction, percent composition of scheduling-related email interaction, ever having a scheduling-related email interaction, percent composition of CX-related email interaction and percent composition of service-related email interaction. The rest plays a minimal importance. It is important to note that neither has a negative impact on membership churn. Notably, the degree of importance appears to hold regardless of the method of computing variable importance. Interestingly, looking at all of the possible combination of interactions of variables, doesn’t appear to be one.

# STEP 1: Create a RSF-specific dataset   
  
clean\_bang\_rsf = clean\_bang\_select %>%   
 select(  
 age\_group,  
 employment\_sector,  
 became\_former\_member,  
 length,   
 membership,   
 avg\_monthly\_rate,   
 attendance\_grouping\_ver.1,  
 ever\_email\_month,  
 num\_emails\_month,  
 ever\_billing\_issue,   
 ever\_cx,  
 new\_per\_ticket\_cx,  
 ever\_scheduling,  
 new\_per\_ticket\_scheduling,  
 ever\_service,  
 new\_per\_ticket\_service  
 )  
  
View(clean\_bang\_rsf)  
  
# STEP 2: Partition the data set into training data & testing data  
  
training.index.rsf = createDataPartition(clean\_bang\_rsf$length, p = 0.8, list = FALSE)  
clean\_bang\_rsf.train = clean\_bang\_rsf[training.index.rsf,]  
clean\_bang\_rsf.test = clean\_bang\_rsf[-training.index.rsf,]  
  
# STEP 3: Formulate the training model + modify parameters   
  
# modify the number of trees  
train.model.base = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 500, splitrule = "logrank", importance = TRUE)  
train.model = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 1000, splitrule = "logrank", importance = TRUE)  
train.model.1 = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE)  
train.model.2 = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 3000, splitrule = "logrank", importance = TRUE)  
train.model.3 = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 4000, splitrule = "logrank", importance = TRUE)  
  
train.model.base # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.34%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 500  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.95  
## No. of variables tried at each split: 4  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.34%

train.model # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.06%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 1000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.731  
## No. of variables tried at each split: 4  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.06%

train.model.1 # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.20%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.88  
## No. of variables tried at each split: 4  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.2%

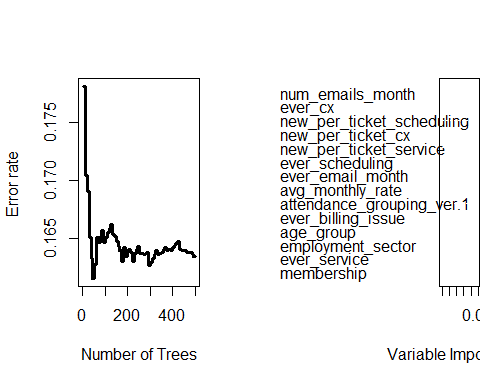
train.model.2 # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.26%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 3000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.897  
## No. of variables tried at each split: 4  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.26%

train.model.3 # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.16%

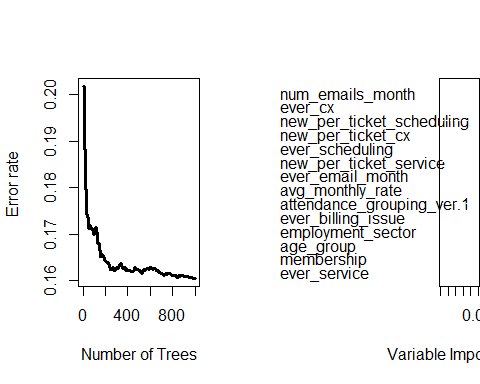
## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 4000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.82275  
## No. of variables tried at each split: 4  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.16%

plot(train.model.base)



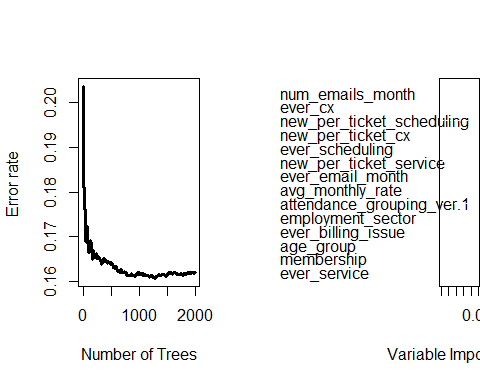
##   
## Importance Relative Imp  
## num\_emails\_month 0.1030 1.0000  
## ever\_cx 0.0343 0.3327  
## new\_per\_ticket\_scheduling 0.0242 0.2345  
## new\_per\_ticket\_cx 0.0172 0.1667  
## new\_per\_ticket\_service 0.0166 0.1614  
## ever\_scheduling 0.0158 0.1532  
## ever\_email\_month 0.0100 0.0969  
## avg\_monthly\_rate 0.0046 0.0449  
## attendance\_grouping\_ver.1 0.0011 0.0111  
## ever\_billing\_issue 0.0006 0.0060  
## age\_group 0.0005 0.0050  
## employment\_sector 0.0005 0.0044  
## ever\_service -0.0001 -0.0005  
## membership -0.0002 -0.0020

plot(train.model)



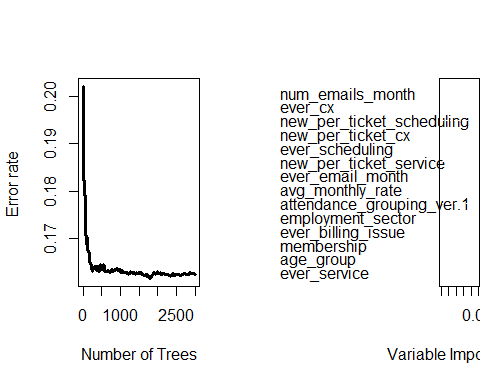
##   
## Importance Relative Imp  
## num\_emails\_month 0.0995 1.0000  
## ever\_cx 0.0336 0.3375  
## new\_per\_ticket\_scheduling 0.0258 0.2589  
## new\_per\_ticket\_cx 0.0201 0.2018  
## ever\_scheduling 0.0183 0.1840  
## new\_per\_ticket\_service 0.0133 0.1338  
## ever\_email\_month 0.0108 0.1082  
## avg\_monthly\_rate 0.0050 0.0503  
## attendance\_grouping\_ver.1 0.0010 0.0096  
## ever\_billing\_issue 0.0006 0.0064  
## employment\_sector 0.0005 0.0053  
## age\_group 0.0003 0.0029  
## membership 0.0002 0.0023  
## ever\_service 0.0000 0.0001

plot(train.model.1)



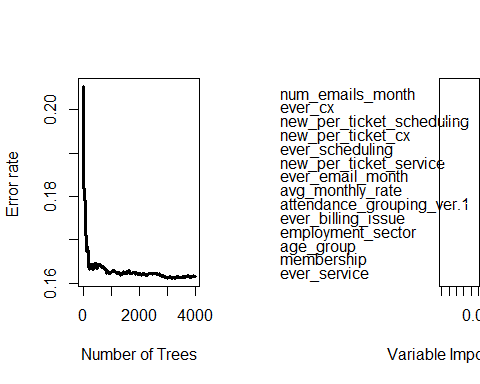
##   
## Importance Relative Imp  
## num\_emails\_month 0.1007 1.0000  
## ever\_cx 0.0341 0.3391  
## new\_per\_ticket\_scheduling 0.0231 0.2294  
## new\_per\_ticket\_cx 0.0173 0.1716  
## ever\_scheduling 0.0170 0.1691  
## new\_per\_ticket\_service 0.0158 0.1566  
## ever\_email\_month 0.0097 0.0968  
## avg\_monthly\_rate 0.0045 0.0444  
## attendance\_grouping\_ver.1 0.0014 0.0134  
## employment\_sector 0.0009 0.0088  
## ever\_billing\_issue 0.0007 0.0066  
## age\_group 0.0005 0.0046  
## membership 0.0002 0.0017  
## ever\_service 0.0000 -0.0004

plot(train.model.2)



##   
## Importance Relative Imp  
## num\_emails\_month 0.1006 1.0000  
## ever\_cx 0.0322 0.3199  
## new\_per\_ticket\_scheduling 0.0232 0.2303  
## new\_per\_ticket\_cx 0.0180 0.1790  
## ever\_scheduling 0.0174 0.1726  
## new\_per\_ticket\_service 0.0150 0.1496  
## ever\_email\_month 0.0102 0.1016  
## avg\_monthly\_rate 0.0045 0.0447  
## attendance\_grouping\_ver.1 0.0008 0.0079  
## employment\_sector 0.0008 0.0077  
## ever\_billing\_issue 0.0008 0.0077  
## membership 0.0003 0.0025  
## age\_group 0.0001 0.0015  
## ever\_service -0.0001 -0.0006

plot(train.model.3)



##   
## Importance Relative Imp  
## num\_emails\_month 0.1017 1.0000  
## ever\_cx 0.0331 0.3252  
## new\_per\_ticket\_scheduling 0.0241 0.2366  
## new\_per\_ticket\_cx 0.0184 0.1809  
## ever\_scheduling 0.0177 0.1741  
## new\_per\_ticket\_service 0.0145 0.1426  
## ever\_email\_month 0.0100 0.0979  
## avg\_monthly\_rate 0.0044 0.0428  
## attendance\_grouping\_ver.1 0.0009 0.0091  
## ever\_billing\_issue 0.0007 0.0071  
## employment\_sector 0.0005 0.0051  
## age\_group 0.0004 0.0037  
## membership 0.0002 0.0023  
## ever\_service 0.0000 -0.0001

# modify mtry   
train.model.a = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 1)  
train.model.b = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 2)  
train.model.c = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 3)  
train.model.d = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 4)  
train.model.e = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 5)  
train.model.f = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 6)  
train.model.g = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 7)  
train.model.h = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 8)  
train.model.i = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 9)  
train.model.j = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 10)  
  
train.model.a # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 19.56%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 11.0855  
## No. of variables tried at each split: 1  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 19.56%

train.model.b # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 17.51%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.5025  
## No. of variables tried at each split: 2  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 17.51%

train.model.c # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.52%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.848  
## No. of variables tried at each split: 3  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.52%

train.model.d # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.24%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.7715  
## No. of variables tried at each split: 4  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.24%

train.model.e # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.03%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.893  
## No. of variables tried at each split: 5  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.03%

train.model.f # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.08%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.7905  
## No. of variables tried at each split: 6  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.08%

train.model.g # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.11%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.809  
## No. of variables tried at each split: 7  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.11%

train.model.h # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 15.95%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.8365  
## No. of variables tried at each split: 8  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 15.95%

train.model.i # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.06%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.996  
## No. of variables tried at each split: 9  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.06%

train.model.j # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.08%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.9255  
## No. of variables tried at each split: 10  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.08%

train.model.proposed = rfsrc(Surv(length, became\_former\_member) ~ ., data = clean\_bang\_rsf.train, ntree = 2000, splitrule = "logrank", importance = TRUE, mtry = 7)  
train.model.proposed # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.95%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.898  
## No. of variables tried at each split: 7  
## Total no. of variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 16.01%

# STEP 4: Determining the important variables within the forest model   
  
vimp(train.model, importance = "permute")$importance

## age\_group employment\_sector membership   
## 1.180646e-04 6.693940e-04 8.724652e-05   
## avg\_monthly\_rate attendance\_grouping\_ver.1 ever\_email\_month   
## 4.596463e-03 7.516281e-04 1.079695e-02   
## num\_emails\_month ever\_billing\_issue ever\_cx   
## 9.865368e-02 7.988894e-04 3.200971e-02   
## new\_per\_ticket\_cx ever\_scheduling new\_per\_ticket\_scheduling   
## 1.963392e-02 1.723374e-02 2.487068e-02   
## ever\_service new\_per\_ticket\_service   
## 3.245604e-06 1.372441e-02

vimp(train.model, importance = "random")$importance

## age\_group employment\_sector membership   
## 0.0023852900 0.0016243457 0.0021772994   
## avg\_monthly\_rate attendance\_grouping\_ver.1 ever\_email\_month   
## 0.0141494624 0.0017012156 0.0180112071   
## num\_emails\_month ever\_billing\_issue ever\_cx   
## 0.1121255872 0.0014601847 0.0339431283   
## new\_per\_ticket\_cx ever\_scheduling new\_per\_ticket\_scheduling   
## 0.0251023681 0.0246676294 0.0313281251   
## ever\_service new\_per\_ticket\_service   
## 0.0006796067 0.0178245294

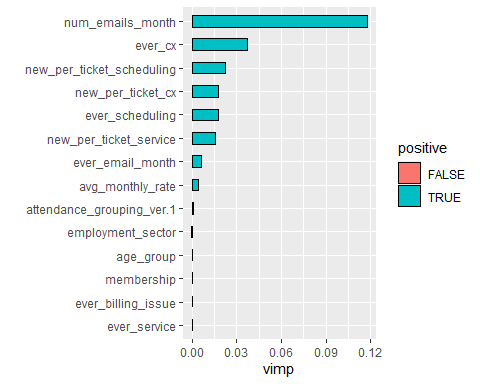
vimp(train.model.proposed, importance = 'permute')$importance

## age\_group employment\_sector membership   
## 1.024009e-04 5.491658e-04 1.851202e-04   
## avg\_monthly\_rate attendance\_grouping\_ver.1 ever\_email\_month   
## 4.467203e-03 6.638167e-04 5.972264e-03   
## num\_emails\_month ever\_billing\_issue ever\_cx   
## 1.187566e-01 2.148423e-04 3.924261e-02   
## new\_per\_ticket\_cx ever\_scheduling new\_per\_ticket\_scheduling   
## 1.835277e-02 1.789232e-02 2.224094e-02   
## ever\_service new\_per\_ticket\_service   
## 3.133369e-06 1.565345e-02

vimp(train.model.proposed, importance = 'random')$importance

## age\_group employment\_sector membership   
## 0.0022821057 0.0008977696 0.0020200669   
## avg\_monthly\_rate attendance\_grouping\_ver.1 ever\_email\_month   
## 0.0124594751 0.0009813869 0.0106837805   
## num\_emails\_month ever\_billing\_issue ever\_cx   
## 0.1347783847 0.0005491734 0.0380180142   
## new\_per\_ticket\_cx ever\_scheduling new\_per\_ticket\_scheduling   
## 0.0221306152 0.0246362171 0.0268971429   
## ever\_service new\_per\_ticket\_service   
## 0.0001299225 0.0189836193

plot(gg\_vimp(train.model.proposed)) # Top Predictors = num\_emails\_month, ever\_cx, new\_per\_ticket\_scheduling, ever\_scheduling, new\_per\_ticket\_cx, new\_per\_ticket\_service



var.select(train.model.proposed, method = 'md') # Top predictors: num\_emails\_month > new\_per\_ticket\_scheduling > new\_per\_ticket\_service > ever\_cx > new\_per\_ticket\_cx

## minimal depth variable selection ...  
##   
##   
## -----------------------------------------------------------  
## family : surv   
## var. selection : Minimal Depth   
## conservativeness : medium   
## x-weighting used? : TRUE   
## dimension : 14   
## sample size : 358   
## ntree : 2000   
## nsplit : 10   
## mtry : 7   
## nodesize : 15   
## refitted forest : FALSE   
## model size : 6   
## depth threshold : 3.5947   
## PE (true OOB) : 16.0104   
##   
##   
## Top variables:  
## depth vimp  
## num\_emails\_month 0.890 0.118  
## ever\_cx 2.499 0.037  
## new\_per\_ticket\_scheduling 3.102 0.022  
## new\_per\_ticket\_service 3.276 0.016  
## avg\_monthly\_rate 3.298 0.004  
## new\_per\_ticket\_cx 3.537 0.018  
## -----------------------------------------------------------

max.model.3 <- max.subtree(train.model.proposed)  
max.model.3$topvars # Top predictors: num\_emails\_month, ever\_cx, new\_per\_ticket\_scheduling, avg\_monthly\_rate, new\_per\_ticket\_service, new\_per\_ticket\_cx

## [1] "avg\_monthly\_rate" "num\_emails\_month"   
## [3] "ever\_cx" "new\_per\_ticket\_cx"   
## [5] "new\_per\_ticket\_scheduling" "new\_per\_ticket\_service"

train.model.proposed.ver1 = rfsrc(Surv(length, became\_former\_member) ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 new\_per\_ticket\_service +   
 ever\_cx +   
 new\_per\_ticket\_cx,  
 data = clean\_bang\_rsf.train, ntree = 2000, mtry = 7, splitrule = 'logrank', importance = TRUE)  
  
train.model.proposed.ver1 # 287 membership churn out of a possible 358 occured in this dataset (~ 80.2%); err.rate = 16.88%

## Sample size: 358  
## Number of deaths: 287  
## Number of trees: 2000  
## Forest terminal node size: 15  
## Average no. of terminal nodes: 14.676  
## No. of variables tried at each split: 5  
## Total no. of variables: 5  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 226  
## Analysis: RSF  
## Family: surv  
## Splitting rule: logrank \*random\*  
## Number of random split points: 10  
## Error rate: 17.12%

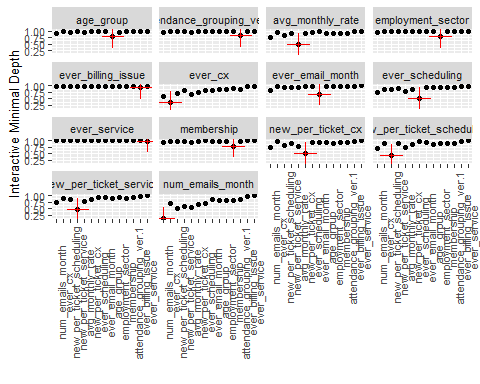
# STEP 5: Explore any potential interaction effects that may exist within the model   
  
find.interaction(train.model.proposed, nvar = 10, method = 'vimp')

## Pairing num\_emails\_month with ever\_cx   
## Pairing num\_emails\_month with new\_per\_ticket\_scheduling   
## Pairing num\_emails\_month with new\_per\_ticket\_cx   
## Pairing num\_emails\_month with ever\_scheduling   
## Pairing num\_emails\_month with new\_per\_ticket\_service   
## Pairing num\_emails\_month with ever\_email\_month   
## Pairing num\_emails\_month with avg\_monthly\_rate   
## Pairing num\_emails\_month with attendance\_grouping\_ver.1   
## Pairing num\_emails\_month with employment\_sector   
## Pairing ever\_cx with new\_per\_ticket\_scheduling   
## Pairing ever\_cx with new\_per\_ticket\_cx   
## Pairing ever\_cx with ever\_scheduling   
## Pairing ever\_cx with new\_per\_ticket\_service   
## Pairing ever\_cx with ever\_email\_month   
## Pairing ever\_cx with avg\_monthly\_rate   
## Pairing ever\_cx with attendance\_grouping\_ver.1   
## Pairing ever\_cx with employment\_sector   
## Pairing new\_per\_ticket\_scheduling with new\_per\_ticket\_cx   
## Pairing new\_per\_ticket\_scheduling with ever\_scheduling   
## Pairing new\_per\_ticket\_scheduling with new\_per\_ticket\_service   
## Pairing new\_per\_ticket\_scheduling with ever\_email\_month   
## Pairing new\_per\_ticket\_scheduling with avg\_monthly\_rate   
## Pairing new\_per\_ticket\_scheduling with attendance\_grouping\_ver.1   
## Pairing new\_per\_ticket\_scheduling with employment\_sector   
## Pairing new\_per\_ticket\_cx with ever\_scheduling   
## Pairing new\_per\_ticket\_cx with new\_per\_ticket\_service   
## Pairing new\_per\_ticket\_cx with ever\_email\_month   
## Pairing new\_per\_ticket\_cx with avg\_monthly\_rate   
## Pairing new\_per\_ticket\_cx with attendance\_grouping\_ver.1   
## Pairing new\_per\_ticket\_cx with employment\_sector   
## Pairing ever\_scheduling with new\_per\_ticket\_service   
## Pairing ever\_scheduling with ever\_email\_month   
## Pairing ever\_scheduling with avg\_monthly\_rate   
## Pairing ever\_scheduling with attendance\_grouping\_ver.1   
## Pairing ever\_scheduling with employment\_sector   
## Pairing new\_per\_ticket\_service with ever\_email\_month   
## Pairing new\_per\_ticket\_service with avg\_monthly\_rate   
## Pairing new\_per\_ticket\_service with attendance\_grouping\_ver.1   
## Pairing new\_per\_ticket\_service with employment\_sector   
## Pairing ever\_email\_month with avg\_monthly\_rate   
## Pairing ever\_email\_month with attendance\_grouping\_ver.1   
## Pairing ever\_email\_month with employment\_sector   
## Pairing avg\_monthly\_rate with attendance\_grouping\_ver.1   
## Pairing avg\_monthly\_rate with employment\_sector   
## Pairing attendance\_grouping\_ver.1 with employment\_sector   
##   
## Method: vimp  
## No. of variables: 10  
## Variables sorted by VIMP?: TRUE  
## No. of variables used for pairing: 10  
## Total no. of paired interactions: 45  
## Monte Carlo replications: 1  
## Type of noising up used for VIMP: permute  
##   
## Var 1 Var 2 Paired  
## num\_emails\_month:ever\_cx 0.1191 0.0385 0.1701  
## num\_emails\_month:new\_per\_ticket\_scheduling 0.1191 0.0223 0.1413  
## num\_emails\_month:new\_per\_ticket\_cx 0.1191 0.0174 0.1365  
## num\_emails\_month:ever\_scheduling 0.1191 0.0174 0.1392  
## num\_emails\_month:new\_per\_ticket\_service 0.1191 0.0156 0.1365  
## num\_emails\_month:ever\_email\_month 0.1191 0.0063 0.1339  
## num\_emails\_month:avg\_monthly\_rate 0.1191 0.0044 0.1231  
## num\_emails\_month:attendance\_grouping\_ver.1 0.1191 0.0006 0.1197  
## num\_emails\_month:employment\_sector 0.1191 0.0004 0.1194  
## ever\_cx:new\_per\_ticket\_scheduling 0.0382 0.0213 0.0636  
## ever\_cx:new\_per\_ticket\_cx 0.0382 0.0178 0.0563  
## ever\_cx:ever\_scheduling 0.0382 0.0171 0.0595  
## ever\_cx:new\_per\_ticket\_service 0.0382 0.0156 0.0629  
## ever\_cx:ever\_email\_month 0.0382 0.0061 0.0441  
## ever\_cx:avg\_monthly\_rate 0.0382 0.0042 0.0411  
## ever\_cx:attendance\_grouping\_ver.1 0.0382 0.0005 0.0390  
## ever\_cx:employment\_sector 0.0382 0.0004 0.0384  
## new\_per\_ticket\_scheduling:new\_per\_ticket\_cx 0.0226 0.0175 0.0439  
## new\_per\_ticket\_scheduling:ever\_scheduling 0.0226 0.0175 0.0454  
## new\_per\_ticket\_scheduling:new\_per\_ticket\_service 0.0226 0.0158 0.0451  
## new\_per\_ticket\_scheduling:ever\_email\_month 0.0226 0.0063 0.0279  
## new\_per\_ticket\_scheduling:avg\_monthly\_rate 0.0226 0.0041 0.0275  
## new\_per\_ticket\_scheduling:attendance\_grouping\_ver.1 0.0226 0.0007 0.0233  
## new\_per\_ticket\_scheduling:employment\_sector 0.0226 0.0004 0.0226  
## new\_per\_ticket\_cx:ever\_scheduling 0.0183 0.0169 0.0391  
## new\_per\_ticket\_cx:new\_per\_ticket\_service 0.0183 0.0160 0.0361  
## new\_per\_ticket\_cx:ever\_email\_month 0.0183 0.0060 0.0244  
## new\_per\_ticket\_cx:avg\_monthly\_rate 0.0183 0.0043 0.0218  
## new\_per\_ticket\_cx:attendance\_grouping\_ver.1 0.0183 0.0006 0.0187  
## new\_per\_ticket\_cx:employment\_sector 0.0183 0.0006 0.0185  
## ever\_scheduling:new\_per\_ticket\_service 0.0176 0.0163 0.0384  
## ever\_scheduling:ever\_email\_month 0.0176 0.0062 0.0227  
## ever\_scheduling:avg\_monthly\_rate 0.0176 0.0045 0.0210  
## ever\_scheduling:attendance\_grouping\_ver.1 0.0176 0.0004 0.0184  
## ever\_scheduling:employment\_sector 0.0176 0.0006 0.0180  
## new\_per\_ticket\_service:ever\_email\_month 0.0149 0.0061 0.0195  
## new\_per\_ticket\_service:avg\_monthly\_rate 0.0149 0.0043 0.0185  
## new\_per\_ticket\_service:attendance\_grouping\_ver.1 0.0149 0.0006 0.0151  
## new\_per\_ticket\_service:employment\_sector 0.0149 0.0004 0.0153  
## ever\_email\_month:avg\_monthly\_rate 0.0061 0.0043 0.0101  
## ever\_email\_month:attendance\_grouping\_ver.1 0.0061 0.0006 0.0063  
## ever\_email\_month:employment\_sector 0.0061 0.0004 0.0064  
## avg\_monthly\_rate:attendance\_grouping\_ver.1 0.0044 0.0006 0.0050  
## avg\_monthly\_rate:employment\_sector 0.0044 0.0005 0.0054  
## attendance\_grouping\_ver.1:employment\_sector 0.0005 0.0007 0.0011  
## Additive Difference  
## num\_emails\_month:ever\_cx 0.1576 0.0125  
## num\_emails\_month:new\_per\_ticket\_scheduling 0.1415 -0.0002  
## num\_emails\_month:new\_per\_ticket\_cx 0.1366 0.0000  
## num\_emails\_month:ever\_scheduling 0.1365 0.0027  
## num\_emails\_month:new\_per\_ticket\_service 0.1347 0.0018  
## num\_emails\_month:ever\_email\_month 0.1254 0.0085  
## num\_emails\_month:avg\_monthly\_rate 0.1235 -0.0004  
## num\_emails\_month:attendance\_grouping\_ver.1 0.1198 -0.0001  
## num\_emails\_month:employment\_sector 0.1196 -0.0001  
## ever\_cx:new\_per\_ticket\_scheduling 0.0595 0.0041  
## ever\_cx:new\_per\_ticket\_cx 0.0561 0.0002  
## ever\_cx:ever\_scheduling 0.0553 0.0042  
## ever\_cx:new\_per\_ticket\_service 0.0538 0.0090  
## ever\_cx:ever\_email\_month 0.0443 -0.0002  
## ever\_cx:avg\_monthly\_rate 0.0425 -0.0014  
## ever\_cx:attendance\_grouping\_ver.1 0.0387 0.0003  
## ever\_cx:employment\_sector 0.0387 -0.0003  
## new\_per\_ticket\_scheduling:new\_per\_ticket\_cx 0.0400 0.0039  
## new\_per\_ticket\_scheduling:ever\_scheduling 0.0401 0.0054  
## new\_per\_ticket\_scheduling:new\_per\_ticket\_service 0.0384 0.0067  
## new\_per\_ticket\_scheduling:ever\_email\_month 0.0288 -0.0009  
## new\_per\_ticket\_scheduling:avg\_monthly\_rate 0.0267 0.0009  
## new\_per\_ticket\_scheduling:attendance\_grouping\_ver.1 0.0232 0.0001  
## new\_per\_ticket\_scheduling:employment\_sector 0.0230 -0.0003  
## new\_per\_ticket\_cx:ever\_scheduling 0.0351 0.0040  
## new\_per\_ticket\_cx:new\_per\_ticket\_service 0.0343 0.0018  
## new\_per\_ticket\_cx:ever\_email\_month 0.0242 0.0002  
## new\_per\_ticket\_cx:avg\_monthly\_rate 0.0226 -0.0008  
## new\_per\_ticket\_cx:attendance\_grouping\_ver.1 0.0189 -0.0002  
## new\_per\_ticket\_cx:employment\_sector 0.0188 -0.0004  
## ever\_scheduling:new\_per\_ticket\_service 0.0339 0.0045  
## ever\_scheduling:ever\_email\_month 0.0239 -0.0012  
## ever\_scheduling:avg\_monthly\_rate 0.0221 -0.0011  
## ever\_scheduling:attendance\_grouping\_ver.1 0.0181 0.0003  
## ever\_scheduling:employment\_sector 0.0183 -0.0002  
## new\_per\_ticket\_service:ever\_email\_month 0.0210 -0.0015  
## new\_per\_ticket\_service:avg\_monthly\_rate 0.0192 -0.0007  
## new\_per\_ticket\_service:attendance\_grouping\_ver.1 0.0155 -0.0003  
## new\_per\_ticket\_service:employment\_sector 0.0152 0.0001  
## ever\_email\_month:avg\_monthly\_rate 0.0105 -0.0004  
## ever\_email\_month:attendance\_grouping\_ver.1 0.0067 -0.0004  
## ever\_email\_month:employment\_sector 0.0066 -0.0001  
## avg\_monthly\_rate:attendance\_grouping\_ver.1 0.0049 0.0000  
## avg\_monthly\_rate:employment\_sector 0.0049 0.0005  
## attendance\_grouping\_ver.1:employment\_sector 0.0012 -0.0001

plot(gg\_interaction(train.model.proposed))

## Warning in gg\_interaction.rfsrc(train.model.proposed): Forest object means we assume max.subtree method for finding interactions.   
## This may take some time.

##   
## Method: maxsubtree  
## No. of variables: 14  
## Variables sorted by minimal depth?: TRUE  
##   
## num\_emails\_month ever\_cx new\_per\_ticket\_scheduling  
## num\_emails\_month 0.13 0.71 0.52  
## ever\_cx 0.58 0.36 0.70  
## new\_per\_ticket\_scheduling 0.71 0.91 0.45  
## new\_per\_ticket\_service 0.72 0.88 0.86  
## avg\_monthly\_rate 0.74 0.96 0.85  
## new\_per\_ticket\_cx 0.73 0.97 0.83  
## ever\_scheduling 0.74 0.88 0.85  
## ever\_email\_month 0.82 0.99 0.95  
## age\_group 0.93 0.99 0.96  
## employment\_sector 0.94 0.99 0.98  
## membership 0.92 0.99 0.97  
## attendance\_grouping\_ver.1 0.94 1.00 0.98  
## ever\_billing\_issue 0.98 1.00 0.99  
## ever\_service 1.00 1.00 1.00  
## new\_per\_ticket\_service avg\_monthly\_rate  
## num\_emails\_month 0.59 0.55  
## ever\_cx 0.81 0.65  
## new\_per\_ticket\_scheduling 0.91 0.76  
## new\_per\_ticket\_service 0.48 0.80  
## avg\_monthly\_rate 0.89 0.49  
## new\_per\_ticket\_cx 0.89 0.78  
## ever\_scheduling 0.92 0.78  
## ever\_email\_month 0.94 0.93  
## age\_group 0.97 0.94  
## employment\_sector 0.98 0.97  
## membership 0.97 0.93  
## attendance\_grouping\_ver.1 0.98 0.95  
## ever\_billing\_issue 0.99 0.99  
## ever\_service 1.00 1.00  
## new\_per\_ticket\_cx ever\_scheduling ever\_email\_month  
## num\_emails\_month 0.67 0.70 0.87  
## ever\_cx 0.76 0.82 0.84  
## new\_per\_ticket\_scheduling 0.87 0.98 0.94  
## new\_per\_ticket\_service 0.86 0.94 0.93  
## avg\_monthly\_rate 0.90 0.96 0.97  
## new\_per\_ticket\_cx 0.51 0.93 0.94  
## ever\_scheduling 0.85 0.53 0.93  
## ever\_email\_month 0.99 0.97 0.69  
## age\_group 0.97 0.99 0.99  
## employment\_sector 0.98 1.00 0.99  
## membership 0.97 1.00 0.99  
## attendance\_grouping\_ver.1 0.98 1.00 0.99  
## ever\_billing\_issue 0.99 1.00 1.00  
## ever\_service 1.00 1.00 1.00  
## age\_group employment\_sector membership  
## num\_emails\_month 0.80 0.81 0.82  
## ever\_cx 0.88 0.87 0.89  
## new\_per\_ticket\_scheduling 0.88 0.89 0.91  
## new\_per\_ticket\_service 0.90 0.91 0.90  
## avg\_monthly\_rate 0.90 0.89 0.89  
## new\_per\_ticket\_cx 0.92 0.90 0.92  
## ever\_scheduling 0.92 0.93 0.93  
## ever\_email\_month 0.98 0.99 0.98  
## age\_group 0.78 0.96 0.97  
## employment\_sector 0.99 0.79 0.98  
## membership 0.97 0.96 0.80  
## attendance\_grouping\_ver.1 0.98 0.98 0.98  
## ever\_billing\_issue 1.00 0.99 1.00  
## ever\_service 1.00 1.00 1.00  
## attendance\_grouping\_ver.1 ever\_billing\_issue  
## num\_emails\_month 0.85 0.95  
## ever\_cx 0.88 0.97  
## new\_per\_ticket\_scheduling 0.90 0.98  
## new\_per\_ticket\_service 0.93 0.98  
## avg\_monthly\_rate 0.91 0.98  
## new\_per\_ticket\_cx 0.92 0.99  
## ever\_scheduling 0.93 0.98  
## ever\_email\_month 0.98 1.00  
## age\_group 0.98 0.99  
## employment\_sector 0.99 1.00  
## membership 0.97 1.00  
## attendance\_grouping\_ver.1 0.82 1.00  
## ever\_billing\_issue 1.00 0.94  
## ever\_service 1.00 1.00  
## ever\_service  
## num\_emails\_month 0.99  
## ever\_cx 1.00  
## new\_per\_ticket\_scheduling 1.00  
## new\_per\_ticket\_service 0.99  
## avg\_monthly\_rate 1.00  
## new\_per\_ticket\_cx 1.00  
## ever\_scheduling 1.00  
## ever\_email\_month 1.00  
## age\_group 1.00  
## employment\_sector 1.00  
## membership 1.00  
## attendance\_grouping\_ver.1 1.00  
## ever\_billing\_issue 1.00  
## ever\_service 0.99



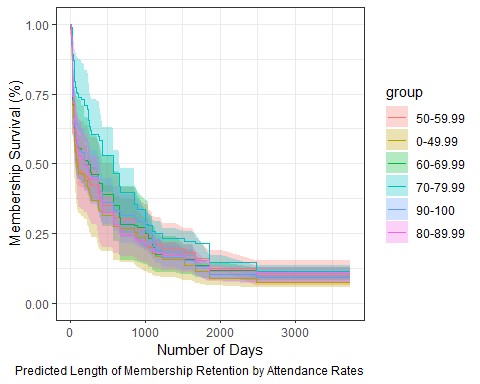
# Higher values indicate lower interactivity with target variable marked in red. Overall, there doesn't seem to be an interactive effect found   
  
  
# STEP 6: Evaluate the performance of the training model with test data set and compare  
  
pred\_churn = predict(train.model.proposed, clean\_bang\_rsf.test, outcome = 'test')  
pred\_churn # Out of 89 individuals, 66 were found to have reported to have churn. However in terms of the model predicting outcomes, there was an error rate of 17.59%

## Sample size of test (predict) data: 89  
## Number of deaths in test data: 66  
## Number of grow trees: 2000  
## Average no. of grow terminal nodes: 14.898  
## Total no. of grow variables: 14  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 56  
## Analysis: RSF  
## Family: surv  
## Test set error rate: 16.76%

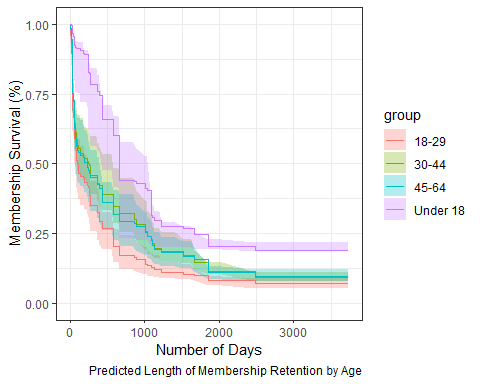
pred\_churn.a = predict(train.model.proposed.ver1, clean\_bang\_rsf.test, outcome = 'test')  
pred\_churn.a # out of 89 individuals, 66 found to have reported to churn membership. However in terms of model predicting outcomes, there was an error rate of 17.28%

## Sample size of test (predict) data: 89  
## Number of deaths in test data: 66  
## Number of grow trees: 2000  
## Average no. of grow terminal nodes: 14.676  
## Total no. of grow variables: 5  
## Resampling used to grow trees: swor  
## Resample size used to grow trees: 56  
## Analysis: RSF  
## Family: surv  
## Test set error rate: 17.21%

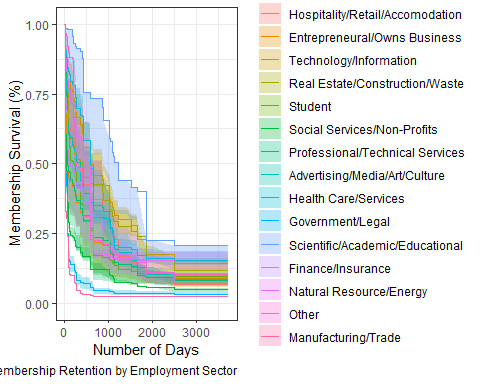
plot(gg\_rfsrc(pred\_churn, by = "attendance\_grouping\_ver.1")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Attendance Rates")



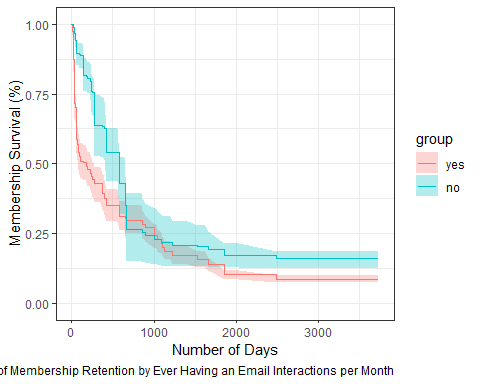
plot(gg\_rfsrc(pred\_churn, by = "age\_group")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Age")



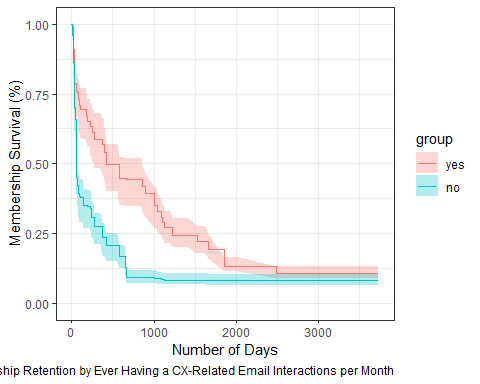
plot(gg\_rfsrc(pred\_churn, by = "employment\_sector")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Employment Sector")



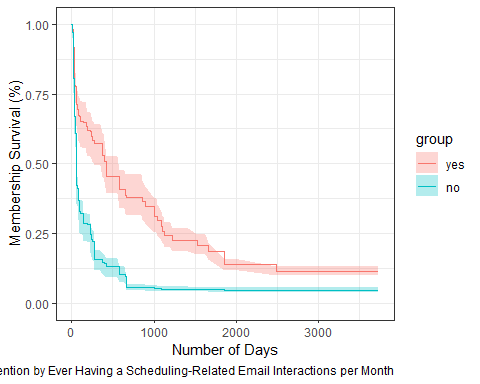
plot(gg\_rfsrc(pred\_churn, by = "ever\_email\_month")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Ever Having an Email Interactions per Month")



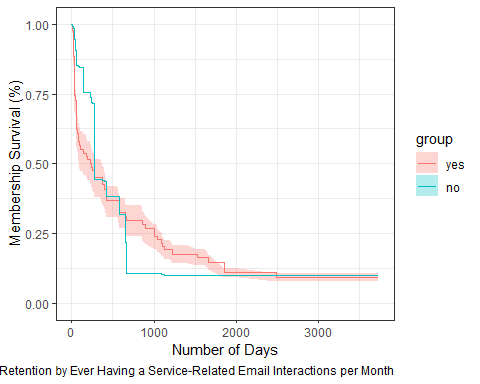
plot(gg\_rfsrc(pred\_churn, by = "ever\_cx")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Ever Having a CX-Related Email Interactions per Month")



plot(gg\_rfsrc(pred\_churn, by = "ever\_scheduling")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Ever Having a Scheduling-Related Email Interactions per Month")



plot(gg\_rfsrc(pred\_churn, by = "ever\_service")) +  
 labs(y = "Membership Survival (%)", x = "Number of Days") +   
 coord\_cartesian(ylim = c(-0.01, 1.01)) +   
 theme\_bw() +   
 labs(caption = "Predicted Length of Membership Retention by Ever Having a Service-Related Email Interactions per Month")



### CHURN ANALYSIS : COX-REGRESSION PROPORTIONAL HAZARD MODEL

Through a bi-directional Stepwise Regression to determine retained predictors of membership length before membership loss through a training dataset partitioning (80%), it was found that the variables that were retained were:

1. number of non-billing related email interactions per month
2. ever having a non-billing related email interaction
3. percent composition of email interactions relating to scheduling
4. percent composition of email interactions relating to CX
5. ever having a CX-related email interaction
6. attendance rate
7. age
8. weighted average monthly membership rate
9. ever having a billing-related email interaction
10. ever having a scheduling-related email interaction

However, this model failed to meet either the assumptions of proportionlity or non-linearity. The model met assumption following the removal of (i) ever having a billing-related email interaction, (ii) weighted average monthly membership rate, (iii) number of non-billing email interactions per month, along with the stratification by age groups. This new proposed model was found to have an error rate of 22.55% (C-statistic = 0.7745). Using this model to predict outcomes with the test data, it was found that the error rate was 22.44% (C-statistic = 0.7766). Overall, the model appears to be decent.

In observing the impact of each of these predictors with the entire data set, it was found that:

* Compared to those that attended less than 50% of their possible allowance, those that attended 50%-59% of the time had a 0.551 times the change in odds of leaving whilst those that attended 70%-89% of the time had a 0.570 times change in the odds of leaving.
* Those that had ever had an email interaction in a given month were found to have a **6.28** times the change in odds of leaving as compared to those that had not had an email interaction in a given month.
* While there was no significant impact with respect to those that did have an scheduling-related email interaction as compared to those that did not, there appears to be a 0.985 times the change in odds of leaving for those that had a 1 factor increase in the percent composition of scheduling-related email interactions.
* Lastly with respect to those that had a 1 factor increase in the percent composition of CX-related email interactions, there was also a 1.03 times the change in odds of leaving. However, there was a large reduction in odds of leaving amongst those that ever had a CX-related email interaction as compared to those that did not.

# STEP 1: Partition data set to a training + testing data set   
  
training.index.cox = createDataPartition(clean\_bang\_select$length, p = 0.8, list = FALSE)   
clean\_bang\_cox.train = clean\_bang\_select[training.index.cox,]   
clean\_bang\_cox.test = clean\_bang\_select[-training.index.cox,]  
  
survival.object = with(clean\_bang\_select, Surv(length, became\_former\_member))  
  
# STEP 2a: Model selection using backward selection  
  
selectCox(Surv(length, became\_former\_member) ~ age\_group +   
 employment\_sector +   
 membership +  
 attendance\_grouping\_ver.1 +   
 monthly\_rate\_group +   
 ever\_billing\_issue +   
 num\_emails\_month +   
 ever\_email\_month +   
 new\_per\_ticket\_scheduling +  
 ever\_scheduling +   
 new\_per\_ticket\_service +   
 ever\_service +  
 new\_per\_ticket\_cx +   
 ever\_cx,  
 data = clean\_bang\_cox.train,  
 rule = "aic")$In

## [1] "num\_emails\_month" "ever\_email\_month"   
## [3] "new\_per\_ticket\_scheduling" "new\_per\_ticket\_service"   
## [5] "ever\_service" "ever\_cx"

# Top variables retained were: "num\_emails\_month", "ever\_email\_month", "new\_per\_ticket\_scheduling", "new\_per\_ticket\_cx" and "ever\_cx"  
  
# STEP 2a: Model selection using bi-direction stepwise regression selection  
  
  
start.cox = coxph(Surv(length, became\_former\_member) ~ 1, data = clean\_bang\_cox.train)  
all.cox = coxph(Surv(length, became\_former\_member) ~ age\_group +   
 employment\_sector +   
 membership +  
 attendance\_grouping\_ver.1 +   
 monthly\_rate\_group +   
 ever\_billing\_issue +   
 num\_emails\_month +   
 ever\_email\_month +   
 new\_per\_ticket\_scheduling +  
 ever\_scheduling +   
 new\_per\_ticket\_service +   
 ever\_service +  
 new\_per\_ticket\_cx +   
 ever\_cx, data = clean\_bang\_cox.train)  
  
step(start.cox, direction = 'both', scope = formula(all.cox))

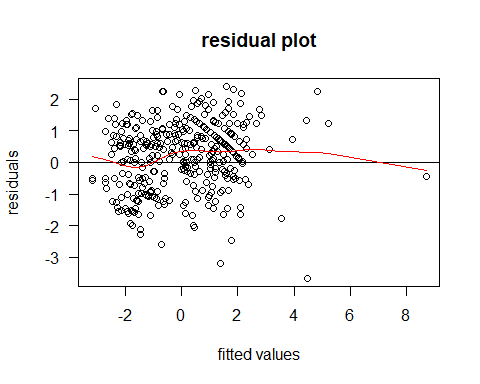
## Start: AIC=2858.32  
## Surv(length, became\_former\_member) ~ 1  
##   
## Df AIC  
## + num\_emails\_month 1 2778.1  
## + new\_per\_ticket\_service 1 2781.7  
## + new\_per\_ticket\_scheduling 1 2785.4  
## + ever\_cx 1 2786.9  
## + ever\_scheduling 1 2808.3  
## + ever\_billing\_issue 1 2845.7  
## + monthly\_rate\_group 11 2848.2  
## + ever\_email\_month 1 2852.3  
## + attendance\_grouping\_ver.1 5 2856.8  
## + age\_group 4 2857.5  
## <none> 2858.3  
## + membership 6 2859.2  
## + new\_per\_ticket\_cx 1 2860.3  
## + ever\_service 1 2860.3  
## + employment\_sector 15 2861.9  
##   
## Step: AIC=2778.12  
## Surv(length, became\_former\_member) ~ num\_emails\_month  
##   
## Df AIC  
## + new\_per\_ticket\_scheduling 1 2667.0  
## + new\_per\_ticket\_service 1 2682.7  
## + ever\_scheduling 1 2691.0  
## + ever\_cx 1 2702.6  
## + ever\_billing\_issue 1 2759.4  
## + age\_group 4 2773.2  
## + attendance\_grouping\_ver.1 5 2776.1  
## + membership 6 2776.5  
## <none> 2778.1  
## + monthly\_rate\_group 11 2778.6  
## + new\_per\_ticket\_cx 1 2778.8  
## + ever\_service 1 2779.4  
## + ever\_email\_month 1 2779.5  
## + employment\_sector 15 2783.7  
## - num\_emails\_month 1 2858.3  
##   
## Step: AIC=2667.04  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling  
##   
## Df AIC  
## + ever\_cx 1 2633.3  
## + ever\_email\_month 1 2636.9  
## + ever\_billing\_issue 1 2661.2  
## + attendance\_grouping\_ver.1 5 2661.3  
## + new\_per\_ticket\_service 1 2661.7  
## + membership 6 2661.9  
## + new\_per\_ticket\_cx 1 2663.7  
## + ever\_scheduling 1 2664.1  
## + monthly\_rate\_group 11 2665.3  
## <none> 2667.0  
## + ever\_service 1 2667.9  
## + age\_group 4 2669.5  
## + employment\_sector 15 2683.5  
## - new\_per\_ticket\_scheduling 1 2778.1  
## - num\_emails\_month 1 2785.4  
##   
## Step: AIC=2633.34  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx  
##   
## Df AIC  
## + ever\_email\_month 1 2573.4  
## + new\_per\_ticket\_cx 1 2619.9  
## + membership 6 2620.4  
## + attendance\_grouping\_ver.1 5 2622.3  
## + monthly\_rate\_group 11 2626.3  
## + ever\_service 1 2629.0  
## + age\_group 4 2629.8  
## + ever\_billing\_issue 1 2632.5  
## <none> 2633.3  
## + ever\_scheduling 1 2633.6  
## + new\_per\_ticket\_service 1 2635.2  
## + employment\_sector 15 2647.9  
## - ever\_cx 1 2667.0  
## - new\_per\_ticket\_scheduling 1 2702.6  
## - num\_emails\_month 1 2745.5  
##   
## Step: AIC=2573.44  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month  
##   
## Df AIC  
## + new\_per\_ticket\_cx 1 2556.0  
## + attendance\_grouping\_ver.1 5 2564.1  
## + monthly\_rate\_group 11 2569.1  
## + new\_per\_ticket\_service 1 2570.1  
## + ever\_billing\_issue 1 2570.3  
## + membership 6 2571.8  
## + age\_group 4 2571.9  
## + ever\_scheduling 1 2572.7  
## <none> 2573.4  
## + ever\_service 1 2575.3  
## + employment\_sector 15 2587.9  
## - ever\_email\_month 1 2633.3  
## - ever\_cx 1 2636.9  
## - num\_emails\_month 1 2657.8  
## - new\_per\_ticket\_scheduling 1 2689.1  
##   
## Step: AIC=2555.98  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx  
##   
## Df AIC  
## + attendance\_grouping\_ver.1 5 2549.8  
## + ever\_scheduling 1 2552.9  
## + age\_group 4 2553.3  
## + monthly\_rate\_group 11 2554.9  
## + membership 6 2554.9  
## + ever\_billing\_issue 1 2555.0  
## <none> 2556.0  
## + ever\_service 1 2557.9  
## + new\_per\_ticket\_service 1 2558.0  
## + employment\_sector 15 2569.1  
## - new\_per\_ticket\_cx 1 2573.4  
## - new\_per\_ticket\_scheduling 1 2609.5  
## - ever\_email\_month 1 2619.9  
## - ever\_cx 1 2627.1  
## - num\_emails\_month 1 2642.3  
##   
## Step: AIC=2549.77  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + attendance\_grouping\_ver.1  
##   
## Df AIC  
## + age\_group 4 2544.1  
## + monthly\_rate\_group 11 2546.7  
## + membership 6 2547.8  
## + ever\_scheduling 1 2548.6  
## <none> 2549.8  
## + ever\_billing\_issue 1 2549.9  
## + new\_per\_ticket\_service 1 2551.5  
## + ever\_service 1 2551.8  
## - attendance\_grouping\_ver.1 5 2556.0  
## + employment\_sector 15 2559.0  
## - new\_per\_ticket\_cx 1 2564.1  
## - new\_per\_ticket\_scheduling 1 2601.3  
## - ever\_email\_month 1 2610.3  
## - ever\_cx 1 2620.3  
## - num\_emails\_month 1 2637.3  
##   
## Step: AIC=2544.14  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + attendance\_grouping\_ver.1 +   
## age\_group  
##   
## Df AIC  
## + ever\_scheduling 1 2542.1  
## + monthly\_rate\_group 11 2542.5  
## <none> 2544.1  
## + ever\_billing\_issue 1 2544.2  
## + membership 6 2544.3  
## + new\_per\_ticket\_service 1 2544.5  
## + ever\_service 1 2545.2  
## - age\_group 4 2549.8  
## + employment\_sector 15 2552.3  
## - attendance\_grouping\_ver.1 5 2553.3  
## - new\_per\_ticket\_cx 1 2558.2  
## - new\_per\_ticket\_scheduling 1 2590.5  
## - ever\_email\_month 1 2606.3  
## - ever\_cx 1 2619.2  
## - num\_emails\_month 1 2634.2  
##   
## Step: AIC=2542.05  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + attendance\_grouping\_ver.1 +   
## age\_group + ever\_scheduling  
##   
## Df AIC  
## + ever\_billing\_issue 1 2540.9  
## + monthly\_rate\_group 11 2541.8  
## <none> 2542.1  
## + new\_per\_ticket\_service 1 2542.5  
## + ever\_service 1 2543.4  
## + membership 6 2543.8  
## - ever\_scheduling 1 2544.1  
## - age\_group 4 2548.6  
## - attendance\_grouping\_ver.1 5 2548.7  
## + employment\_sector 15 2550.8  
## - new\_per\_ticket\_scheduling 1 2552.5  
## - new\_per\_ticket\_cx 1 2558.8  
## - ever\_email\_month 1 2605.4  
## - ever\_cx 1 2618.4  
## - num\_emails\_month 1 2636.1  
##   
## Step: AIC=2540.9  
## Surv(length, became\_former\_member) ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + attendance\_grouping\_ver.1 +   
## age\_group + ever\_scheduling + ever\_billing\_issue  
##   
## Df AIC  
## <none> 2540.9  
## + new\_per\_ticket\_service 1 2541.6  
## + monthly\_rate\_group 11 2541.8  
## - ever\_billing\_issue 1 2542.1  
## + ever\_service 1 2542.5  
## + membership 6 2542.9  
## - ever\_scheduling 1 2544.2  
## - attendance\_grouping\_ver.1 5 2545.2  
## - age\_group 4 2548.1  
## - new\_per\_ticket\_scheduling 1 2549.8  
## + employment\_sector 15 2551.2  
## - new\_per\_ticket\_cx 1 2555.3  
## - ever\_email\_month 1 2605.2  
## - ever\_cx 1 2610.1  
## - num\_emails\_month 1 2636.3

## Call:  
## coxph(formula = Surv(length, became\_former\_member) ~ num\_emails\_month +   
## new\_per\_ticket\_scheduling + ever\_cx + ever\_email\_month +   
## new\_per\_ticket\_cx + attendance\_grouping\_ver.1 + age\_group +   
## ever\_scheduling + ever\_billing\_issue, data = clean\_bang\_cox.train)  
##   
## coef exp(coef) se(coef) z p  
## num\_emails\_month 0.236043 1.266229 0.020893 11.298 < 2e-16  
## new\_per\_ticket\_scheduling -0.014229 0.985872 0.004364 -3.260 0.00111  
## ever\_cxyes -2.061703 0.127237 0.243559 -8.465 < 2e-16  
## ever\_email\_monthyes 1.647599 5.194495 0.224810 7.329 2.32e-13  
## new\_per\_ticket\_cx 0.026816 1.027179 0.006372 4.209 2.57e-05  
## attendance\_grouping\_ver.150-59.99 -0.639316 0.527653 0.213196 -2.999 0.00271  
## attendance\_grouping\_ver.160-69.99 -0.462470 0.629726 0.223210 -2.072 0.03827  
## attendance\_grouping\_ver.170-79.99 -0.485612 0.615320 0.221934 -2.188 0.02866  
## attendance\_grouping\_ver.180-89.99 -0.455561 0.634092 0.232720 -1.958 0.05028  
## attendance\_grouping\_ver.190-100 -0.117967 0.888725 0.180385 -0.654 0.51313  
## age\_group18-29 1.108708 3.030441 0.480538 2.307 0.02104  
## age\_group30-44 0.802486 2.231080 0.465659 1.723 0.08483  
## age\_group45-64 0.500639 1.649775 0.478934 1.045 0.29587  
## age\_group65+ 0.217688 1.243200 0.653308 0.333 0.73898  
## ever\_schedulingyes -0.541749 0.581730 0.237778 -2.278 0.02270  
## ever\_billing\_issueyes -0.270696 0.762848 0.155513 -1.741 0.08174  
##   
## Likelihood ratio test=349.4 on 16 df, p=< 2.2e-16  
## n= 358, number of events= 283

# new\_per\_ticket\_scheduling, num\_emails\_month, ever\_cx, ever\_email\_month, new\_per\_ticket\_cx, attendance\_grouping\_ver.1, age\_group, monthly\_rate\_group, ever\_billing\_issue and ever\_scheduling  
  
  
  
testing = cph(Surv(length, became\_former\_member) ~   
 num\_emails\_month +   
 ever\_email\_month +   
 new\_per\_ticket\_scheduling +   
 new\_per\_ticket\_cx +   
 ever\_cx +   
 attendance\_grouping\_ver.1 +  
 age\_group +   
 monthly\_rate\_group +   
 ever\_billing\_issue +   
 ever\_scheduling, data = clean\_bang\_cox.train, x = T, y = T, surv = T)  
  
# Step 3a: Testing Assumption of the Model   
  
cox.zph(testing) # Issue with num\_emails\_month + new\_per\_ticket\_cx

## chisq df p  
## num\_emails\_month 6.32e+00 1 0.012  
## ever\_email\_month 3.42e+00 1 0.064  
## new\_per\_ticket\_scheduling 2.21e-06 1 0.999  
## new\_per\_ticket\_cx 5.53e+00 1 0.019  
## ever\_cx 3.96e+00 1 0.047  
## attendance\_grouping\_ver.1 5.94e+00 5 0.312  
## age\_group 2.10e+00 4 0.718  
## monthly\_rate\_group 5.51e+01 11 7.4e-08  
## ever\_billing\_issue 2.94e-01 1 0.588  
## ever\_scheduling 6.66e-01 1 0.415  
## GLOBAL 8.34e+01 27 1.1e-07

plot( predict(testing),   
 residuals(testing, type = "deviance"),   
 xlab = "fitted values",   
 ylab = "residuals",   
 main = "residual plot", las = 1  
)  
  
abline(h = 0)  
  
  
lines(smooth.spline(predict(testing), residuals(testing, type = 'deviance')), col = 'red')



# Not even close to satisfying the assumptions of non-linearity; need to do some re-working   
  
  
  
# Step 3b: Rework predictor selection + Re-testing Assumption of the Model   
  
proposed.cox.model.train = cph(Surv(length, became\_former\_member) ~   
 ever\_email\_month +   
 new\_per\_ticket\_scheduling +   
 new\_per\_ticket\_cx +   
 ever\_cx +   
 attendance\_grouping\_ver.1 +  
 strat(age\_group) +   
 ever\_scheduling, data = clean\_bang\_cox.train, x = T, y = T, surv = T)  
  
  
cox.zph(proposed.cox.model.train) # Holds the assumption of proportionality

## chisq df p  
## ever\_email\_month 4.6284 1 0.031  
## new\_per\_ticket\_scheduling 0.0133 1 0.908  
## new\_per\_ticket\_cx 3.2393 1 0.072  
## ever\_cx 1.2691 1 0.260  
## attendance\_grouping\_ver.1 2.3750 5 0.795  
## ever\_scheduling 1.9318 1 0.165  
## GLOBAL 12.1511 10 0.275

plot(  
 predict(proposed.cox.model.train),   
 residuals(proposed.cox.model.train, type = "deviance"),   
 xlab = "fitted values",   
 ylab = "residuals",   
 main = "residual plot", las = 1  
)  
  
abline(h = 0)  
  
lines(smooth.spline(predict(proposed.cox.model.train), residuals(proposed.cox.model.train, type = 'deviance')), col = 'red') # meh  
  
train\_surv = with(clean\_bang\_cox.train, Surv(length, became\_former\_member))  
train.estimates = survest(proposed.cox.model.train, newdata = clean\_bang\_cox.train, times = 69)$surv  
rcorr.cens(train.estimates, train\_surv) # c = 0.7745 (aka. err.rate = 22.55%)

## C Index Dxy S.D. n missing   
## 7.781534e-01 5.563068e-01 2.341529e-02 3.580000e+02 0.000000e+00   
## uncensored Relevant Pairs Concordant Uncertain   
## 2.830000e+02 1.073760e+05 8.355500e+04 1.874000e+04

# Step 4: validating my proposed mode with test data set   
  
test\_surv = with(clean\_bang\_cox.test, Surv(length, became\_former\_member)) # this is the survival object in which to test against   
estimates = survest(proposed.cox.model.train, newdata = clean\_bang\_cox.test, times = 69)$surv # time is just arbitrary here; survival estimates based on the training model using the test data set   
rcorr.cens(estimates, test\_surv) # C = 0.7756 or err.rate = 22.44%

## C Index Dxy S.D. n missing   
## 7.606494e-01 5.212988e-01 5.083363e-02 8.900000e+01 0.000000e+00   
## uncensored Relevant Pairs Concordant Uncertain   
## 7.000000e+01 6.714000e+03 5.107000e+03 9.080000e+02

# Step 5: Summary of the whole dataset using the proposed model  
  
cox.model.churn = coxph(Surv(length, became\_former\_member) ~   
 ever\_email\_month +   
 new\_per\_ticket\_scheduling +   
 new\_per\_ticket\_cx +   
 ever\_cx +   
 attendance\_grouping\_ver.1 +  
 strata(age\_group) +   
 ever\_scheduling, data = clean\_bang\_select)  
  
summary(cox.model.churn) # C-statistic = 0.781

## Call:  
## coxph(formula = Surv(length, became\_former\_member) ~ ever\_email\_month +   
## new\_per\_ticket\_scheduling + new\_per\_ticket\_cx + ever\_cx +   
## attendance\_grouping\_ver.1 + strata(age\_group) + ever\_scheduling,   
## data = clean\_bang\_select)  
##   
## n= 447, number of events= 353   
##   
## coef exp(coef) se(coef) z  
## ever\_email\_monthyes 1.838344 6.286117 0.206464 8.904  
## new\_per\_ticket\_scheduling -0.015299 0.984817 0.003925 -3.898  
## new\_per\_ticket\_cx 0.027677 1.028064 0.005403 5.123  
## ever\_cxyes -2.112674 0.120914 0.205221 -10.295  
## attendance\_grouping\_ver.150-59.99 -0.596172 0.550916 0.194012 -3.073  
## attendance\_grouping\_ver.160-69.99 -0.112368 0.893715 0.180600 -0.622  
## attendance\_grouping\_ver.170-79.99 -0.556584 0.573164 0.198920 -2.798  
## attendance\_grouping\_ver.180-89.99 -0.561434 0.570391 0.217047 -2.587  
## attendance\_grouping\_ver.190-100 -0.182059 0.833552 0.157828 -1.154  
## ever\_schedulingyes -0.177018 0.837765 0.208249 -0.850  
## Pr(>|z|)   
## ever\_email\_monthyes < 2e-16 \*\*\*  
## new\_per\_ticket\_scheduling 9.70e-05 \*\*\*  
## new\_per\_ticket\_cx 3.01e-07 \*\*\*  
## ever\_cxyes < 2e-16 \*\*\*  
## attendance\_grouping\_ver.150-59.99 0.00212 \*\*   
## attendance\_grouping\_ver.160-69.99 0.53381   
## attendance\_grouping\_ver.170-79.99 0.00514 \*\*   
## attendance\_grouping\_ver.180-89.99 0.00969 \*\*   
## attendance\_grouping\_ver.190-100 0.24869   
## ever\_schedulingyes 0.39531   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## exp(coef) exp(-coef) lower .95 upper .95  
## ever\_email\_monthyes 6.2861 0.1591 4.19411 9.4216  
## new\_per\_ticket\_scheduling 0.9848 1.0154 0.97727 0.9924  
## new\_per\_ticket\_cx 1.0281 0.9727 1.01723 1.0390  
## ever\_cxyes 0.1209 8.2703 0.08087 0.1808  
## attendance\_grouping\_ver.150-59.99 0.5509 1.8152 0.37665 0.8058  
## attendance\_grouping\_ver.160-69.99 0.8937 1.1189 0.62730 1.2733  
## attendance\_grouping\_ver.170-79.99 0.5732 1.7447 0.38811 0.8464  
## attendance\_grouping\_ver.180-89.99 0.5704 1.7532 0.37275 0.8728  
## attendance\_grouping\_ver.190-100 0.8336 1.1997 0.61177 1.1357  
## ever\_schedulingyes 0.8378 1.1937 0.55701 1.2600  
##   
## Concordance= 0.781 (se = 0.011 )  
## Likelihood ratio test= 294.5 on 10 df, p=<2e-16  
## Wald test = 259.2 on 10 df, p=<2e-16  
## Score (logrank) test = 300.5 on 10 df, p=<2e-16

AIC(cox.model.churn) # 2649.277

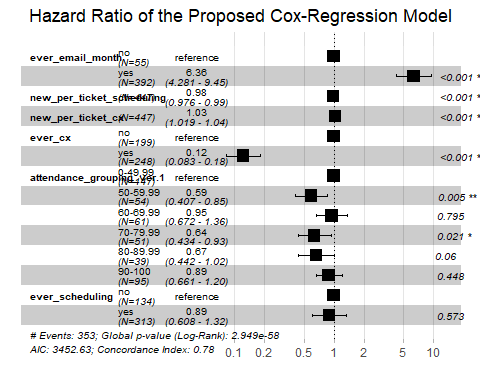
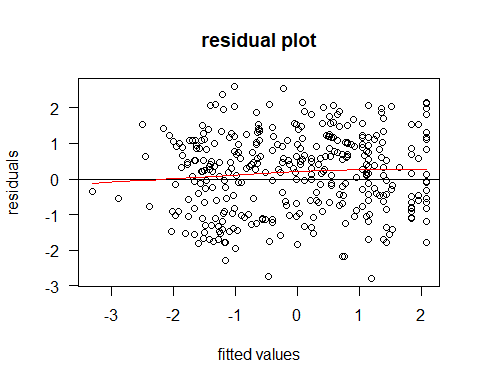
## [1] 2649.227

vif(cox.model.churn) # No issue of collinearity here

## ever\_email\_monthyes new\_per\_ticket\_scheduling   
## 1.172078 3.095399   
## new\_per\_ticket\_cx ever\_cxyes   
## 3.019967 2.660034   
## attendance\_grouping\_ver.150-59.99 attendance\_grouping\_ver.160-69.99   
## 1.226746 1.281800   
## attendance\_grouping\_ver.170-79.99 attendance\_grouping\_ver.180-89.99   
## 1.204111 1.168101   
## attendance\_grouping\_ver.190-100 ever\_schedulingyes   
## 1.442219 2.552844

cox.model.churn.a = coxph(Surv(length, became\_former\_member) ~   
 ever\_email\_month +   
 new\_per\_ticket\_scheduling +   
 new\_per\_ticket\_cx +   
 ever\_cx +   
 attendance\_grouping\_ver.1 +  
 ever\_scheduling, data = clean\_bang\_select)  
  
ggforest(cox.model.churn.a, main = "Hazard Ratio of the Proposed Cox-Regression Model")

## Warning in .get\_data(model, data = data): The `data` argument is not provided.  
## Data will be extracted from model fit.



# Note: no idea on how to include the strata function into the ggforest

## Modeling retention status at 3-/6- and 12- months.

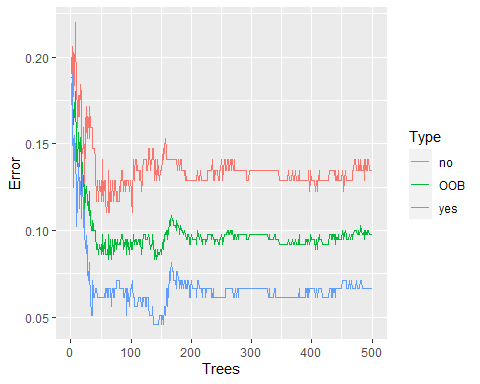
### RETENTION ANALYSIS: Membership status at 3-Months via Random Forest

Using the random survival forest specific dataset, I’ve split the data set 80:20 with respect to training:test. In forming the training model, which has an error rate of **6.69%**, it was found that the error rate in predicting membership length to churn with the test data was **10.23%**. So really a large differential. Looking at the various ways to modify the parameters, it was found that the error rate more-or-less stabilized after 1000 trees as evident by the marginal differences in error rates at the higher number of trees. However, in terms of tuning this model, I’ve adjusted the model to include ntree = 2000 and mtry at 4. Examining the importance of each variable used in this model, it was found that number of non-billing email interaction played the largest role, followed by the percent composition of non=billing related email interactions (scheduling, service and CX).

# Step 1: Create a specific data set to be used for retention status analysis   
  
clean\_bang\_retention\_3m = clean\_bang\_select %>%   
 select(  
 age\_group,   
 employment\_sector,   
 retention\_3m,  
 avg\_monthly\_rate,  
 attendance\_grouping\_ver.1,  
 ever\_email\_month,  
 num\_emails\_month,  
 ever\_billing\_issue,   
 ever\_cx,  
 new\_per\_ticket\_cx,  
 ever\_scheduling,  
 new\_per\_ticket\_scheduling,  
 ever\_service,  
 new\_per\_ticket\_service  
 )  
  
# Step 2: create a partition of this data set by splitting it based on retention status at 3 Months  
  
trainIndex\_3m = createDataPartition(clean\_bang\_retention\_3m$retention\_3m, p = 0.8, list = FALSE)  
clean\_bang\_retention\_3m.train = clean\_bang\_retention\_3m[trainIndex\_3m,]   
clean\_bang\_retention\_3m.test = clean\_bang\_retention\_3m[-trainIndex\_3m,]   
  
# Step 3: Create a random forest model using training data   
  
training.model.3m = randomForest(retention\_3m ~., data = clean\_bang\_retention\_3m.train, proximity = T)  
training.model.3m # OOB error rate is 8.71%

##   
## Call:  
## randomForest(formula = retention\_3m ~ ., data = clean\_bang\_retention\_3m.train, proximity = T)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 9.75%  
## Confusion matrix:  
## no yes class.error  
## no 141 22 0.13496933  
## yes 13 183 0.06632653

# Step 4: Create a data frame to see how the error rate changes as a function of increasing number of trees (currently capped at 500)  
  
oob.error.data.3m = data.frame(  
 Trees = rep(1:nrow(training.model.3m$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.3m$err.rate)),  
 Error = c(training.model.3m$err.rate[, "OOB"],   
 training.model.3m$err.rate[,"no"],   
 training.model.3m$err.rate[, 'yes']))  
  
View(oob.error.data.3m)  
  
ggplot(data = oob.error.data.3m, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



# Step 4a: Add more trees and see what happens:   
  
training.model.3m\_ver1 = randomForest(retention\_3m ~., data = clean\_bang\_retention\_3m.train, proximity = T, ntree = 1000)  
training.model.3m\_ver2 = randomForest(retention\_3m ~., data = clean\_bang\_retention\_3m.train, proximity = T, ntree = 2000)  
training.model.3m\_ver3 = randomForest(retention\_3m ~., data = clean\_bang\_retention\_3m.train, proximity = T, ntree = 3000)  
  
training.model.3m # REFERENCE

##   
## Call:  
## randomForest(formula = retention\_3m ~ ., data = clean\_bang\_retention\_3m.train, proximity = T)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 9.75%  
## Confusion matrix:  
## no yes class.error  
## no 141 22 0.13496933  
## yes 13 183 0.06632653

training.model.3m\_ver1 # 9.47%

##   
## Call:  
## randomForest(formula = retention\_3m ~ ., data = clean\_bang\_retention\_3m.train, proximity = T, ntree = 1000)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 9.47%  
## Confusion matrix:  
## no yes class.error  
## no 142 21 0.12883436  
## yes 13 183 0.06632653

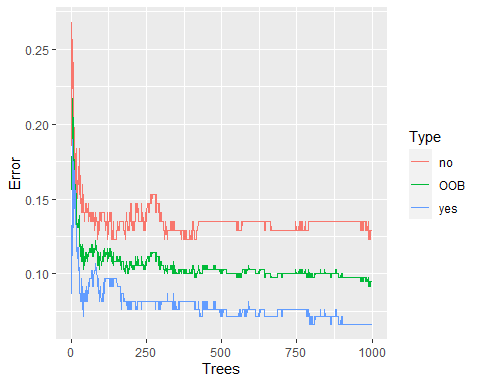
training.model.3m\_ver2 # 9.75%

##   
## Call:  
## randomForest(formula = retention\_3m ~ ., data = clean\_bang\_retention\_3m.train, proximity = T, ntree = 2000)   
## Type of random forest: classification  
## Number of trees: 2000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 10.03%  
## Confusion matrix:  
## no yes class.error  
## no 141 22 0.13496933  
## yes 14 182 0.07142857

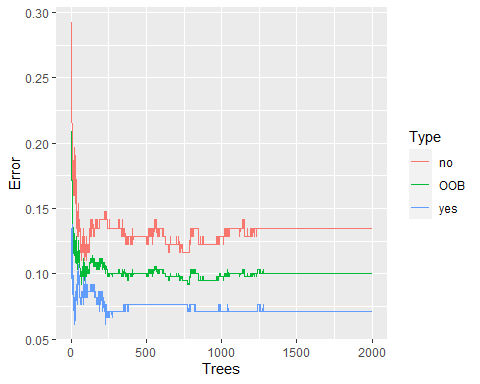
training.model.3m\_ver3 # 9.75%

##   
## Call:  
## randomForest(formula = retention\_3m ~ ., data = clean\_bang\_retention\_3m.train, proximity = T, ntree = 3000)   
## Type of random forest: classification  
## Number of trees: 3000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 9.75%  
## Confusion matrix:  
## no yes class.error  
## no 142 21 0.12883436  
## yes 14 182 0.07142857

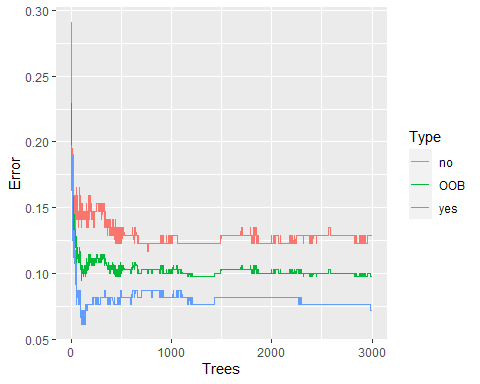
oob.error.data.3m\_ver1 = data.frame(  
 Trees = rep(1:nrow(training.model.3m\_ver1$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.3m\_ver1$err.rate)),  
 Error = c(training.model.3m\_ver1$err.rate[, "OOB"],   
 training.model.3m\_ver1$err.rate[,"no"],   
 training.model.3m\_ver1$err.rate[, 'yes']))  
  
oob.error.data.3m\_ver2 = data.frame(  
 Trees = rep(1:nrow(training.model.3m\_ver2$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.3m\_ver2$err.rate)),  
 Error = c(training.model.3m\_ver2$err.rate[, "OOB"],   
 training.model.3m\_ver2$err.rate[,"no"],   
 training.model.3m\_ver2$err.rate[, 'yes']))  
  
oob.error.data.3m\_ver3 = data.frame(  
 Trees = rep(1:nrow(training.model.3m\_ver3$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.3m\_ver3$err.rate)),  
 Error = c(training.model.3m\_ver3$err.rate[, "OOB"],   
 training.model.3m\_ver3$err.rate[,"no"],   
 training.model.3m\_ver3$err.rate[, 'yes']))  
  
ggplot(data = oob.error.data.3m\_ver1, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



ggplot(data = oob.error.data.3m\_ver2, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



ggplot(data = oob.error.data.3m\_ver3, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



# Looks like we did a worse job with increasing number of trees, but this leveled off after 2000.   
  
# STEP 3B: Fine tuning mtry   
  
oob.values <- vector(length = 10)  
for(i in 1:10) {  
 temp.model <- randomForest(retention\_3m ~., data = clean\_bang\_retention\_3m.train, mtry = i, ntree = 2000)  
 oob.values[i] <- temp.model$err.rate[nrow(temp.model$err.rate), 1]  
}  
  
oob.values

## [1] 0.14206128 0.09749304 0.08635097 0.09749304 0.11142061 0.10027855  
## [7] 0.10306407 0.10306407 0.10306407 0.10027855

# Looks like optimal value is 4  
  
proposed.training.model.3m = randomForest(retention\_3m ~., data = clean\_bang\_retention\_3m.train, proximity = T, mtry = 4, ntree = 2000) # err.rate = 6.69%  
  
attributes(proposed.training.model.3m)

## $names  
## [1] "call" "type" "predicted" "err.rate"   
## [5] "confusion" "votes" "oob.times" "classes"   
## [9] "importance" "importanceSD" "localImportance" "proximity"   
## [13] "ntree" "mtry" "forest" "y"   
## [17] "test" "inbag" "terms"   
##   
## $class  
## [1] "randomForest.formula" "randomForest"

proposed.training.model.3m$confusion

## no yes class.error  
## no 142 21 0.12883436  
## yes 15 181 0.07653061

# Step 4: Test this proposed model against testing data   
  
pred\_3m\_rf <- predict(proposed.training.model.3m, newdata = clean\_bang\_retention\_3m.test)  
head(pred\_3m\_rf)

## 5 6 7 8 13 14   
## yes yes yes yes yes yes   
## Levels: no yes

head(clean\_bang\_retention\_3m.test$retention\_3m)

## [1] yes yes yes yes yes yes  
## Levels: no yes

cbind(pred\_3m\_rf, clean\_bang\_retention\_3m.test$retention\_3m)

## pred\_3m\_rf   
## 5 2 2  
## 6 2 2  
## 7 2 2  
## 8 2 2  
## 13 2 2  
## 14 2 2  
## 16 1 1  
## 24 2 2  
## 28 1 1  
## 30 2 2  
## 31 2 2  
## 40 2 2  
## 44 1 1  
## 49 2 1  
## 53 2 2  
## 74 1 1  
## 78 2 2  
## 79 1 1  
## 84 1 1  
## 85 2 2  
## 92 2 2  
## 97 2 2  
## 102 1 1  
## 106 1 1  
## 116 1 1  
## 121 1 1  
## 128 2 2  
## 131 1 1  
## 141 2 2  
## 144 2 2  
## 146 2 2  
## 147 1 2  
## 154 2 2  
## 157 1 1  
## 164 2 2  
## 166 1 1  
## 170 2 2  
## 171 2 2  
## 173 2 2  
## 177 1 1  
## 184 1 1  
## 186 1 1  
## 191 1 1  
## 197 2 2  
## 199 2 2  
## 208 2 2  
## 213 1 1  
## 217 2 2  
## 219 2 2  
## 222 1 1  
## 226 1 1  
## 229 1 1  
## 230 2 2  
## 234 1 1  
## 236 1 1  
## 248 1 1  
## 249 2 2  
## 250 2 2  
## 263 1 1  
## 270 1 1  
## 272 2 2  
## 281 2 2  
## 283 1 1  
## 287 1 1  
## 289 1 1  
## 296 1 1  
## 299 1 1  
## 300 2 2  
## 309 2 2  
## 311 1 1  
## 316 1 1  
## 321 2 2  
## 350 2 2  
## 358 2 2  
## 365 2 1  
## 369 2 2  
## 372 2 2  
## 374 1 1  
## 384 2 2  
## 385 2 1  
## 393 1 1  
## 404 2 2  
## 405 1 1  
## 411 2 2  
## 412 2 2  
## 431 1 1  
## 433 2 2  
## 441 2 2

results\_3m = data.frame(  
 individuals = rep(1:nrow(clean\_bang\_retention\_3m.test)),  
 prediction = pred\_3m\_rf,  
 truth = clean\_bang\_retention\_3m.test$retention\_3m  
)  
results\_3m

## individuals prediction truth  
## 5 1 yes yes  
## 6 2 yes yes  
## 7 3 yes yes  
## 8 4 yes yes  
## 13 5 yes yes  
## 14 6 yes yes  
## 16 7 no no  
## 24 8 yes yes  
## 28 9 no no  
## 30 10 yes yes  
## 31 11 yes yes  
## 40 12 yes yes  
## 44 13 no no  
## 49 14 yes no  
## 53 15 yes yes  
## 74 16 no no  
## 78 17 yes yes  
## 79 18 no no  
## 84 19 no no  
## 85 20 yes yes  
## 92 21 yes yes  
## 97 22 yes yes  
## 102 23 no no  
## 106 24 no no  
## 116 25 no no  
## 121 26 no no  
## 128 27 yes yes  
## 131 28 no no  
## 141 29 yes yes  
## 144 30 yes yes  
## 146 31 yes yes  
## 147 32 no yes  
## 154 33 yes yes  
## 157 34 no no  
## 164 35 yes yes  
## 166 36 no no  
## 170 37 yes yes  
## 171 38 yes yes  
## 173 39 yes yes  
## 177 40 no no  
## 184 41 no no  
## 186 42 no no  
## 191 43 no no  
## 197 44 yes yes  
## 199 45 yes yes  
## 208 46 yes yes  
## 213 47 no no  
## 217 48 yes yes  
## 219 49 yes yes  
## 222 50 no no  
## 226 51 no no  
## 229 52 no no  
## 230 53 yes yes  
## 234 54 no no  
## 236 55 no no  
## 248 56 no no  
## 249 57 yes yes  
## 250 58 yes yes  
## 263 59 no no  
## 270 60 no no  
## 272 61 yes yes  
## 281 62 yes yes  
## 283 63 no no  
## 287 64 no no  
## 289 65 no no  
## 296 66 no no  
## 299 67 no no  
## 300 68 yes yes  
## 309 69 yes yes  
## 311 70 no no  
## 316 71 no no  
## 321 72 yes yes  
## 350 73 yes yes  
## 358 74 yes yes  
## 365 75 yes no  
## 369 76 yes yes  
## 372 77 yes yes  
## 374 78 no no  
## 384 79 yes yes  
## 385 80 yes no  
## 393 81 no no  
## 404 82 yes yes  
## 405 83 no no  
## 411 84 yes yes  
## 412 85 yes yes  
## 431 86 no no  
## 433 87 yes yes  
## 441 88 yes yes

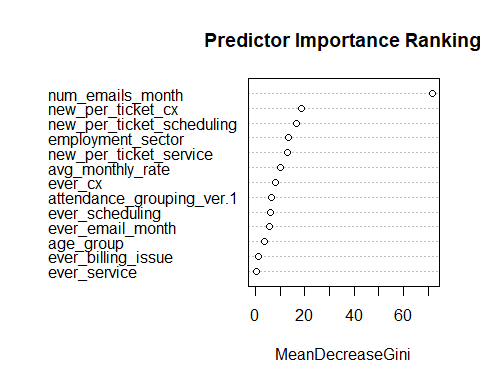
confusionMatrix(pred\_3m\_rf, clean\_bang\_retention\_3m.test$retention\_3m) # accuracy = 0.8977 or err.rate of 10.23%

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 37 1  
## yes 3 47  
##   
## Accuracy : 0.9545   
## 95% CI : (0.8877, 0.9875)  
## No Information Rate : 0.5455   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9079   
##   
## Mcnemar's Test P-Value : 0.6171   
##   
## Sensitivity : 0.9250   
## Specificity : 0.9792   
## Pos Pred Value : 0.9737   
## Neg Pred Value : 0.9400   
## Prevalence : 0.4545   
## Detection Rate : 0.4205   
## Detection Prevalence : 0.4318   
## Balanced Accuracy : 0.9521   
##   
## 'Positive' Class : no   
##

# STEP 5: Determining which variables are important predictors   
  
varImp(proposed.training.model.3m)

## Overall  
## age\_group 3.7686084  
## employment\_sector 13.5593857  
## avg\_monthly\_rate 10.2912174  
## attendance\_grouping\_ver.1 6.4924624  
## ever\_email\_month 5.6714667  
## num\_emails\_month 71.7819837  
## ever\_billing\_issue 1.3997070  
## ever\_cx 7.9958873  
## new\_per\_ticket\_cx 18.6814021  
## ever\_scheduling 5.9690840  
## new\_per\_ticket\_scheduling 16.7379079  
## ever\_service 0.5452137  
## new\_per\_ticket\_service 13.1416123

varImpPlot(proposed.training.model.3m, sort = T, main = "Predictor Importance Ranking")



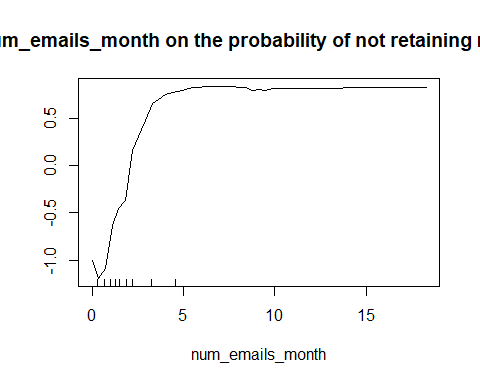
importance(proposed.training.model.3m)

## MeanDecreaseGini  
## age\_group 3.7686084  
## employment\_sector 13.5593857  
## avg\_monthly\_rate 10.2912174  
## attendance\_grouping\_ver.1 6.4924624  
## ever\_email\_month 5.6714667  
## num\_emails\_month 71.7819837  
## ever\_billing\_issue 1.3997070  
## ever\_cx 7.9958873  
## new\_per\_ticket\_cx 18.6814021  
## ever\_scheduling 5.9690840  
## new\_per\_ticket\_scheduling 16.7379079  
## ever\_service 0.5452137  
## new\_per\_ticket\_service 13.1416123

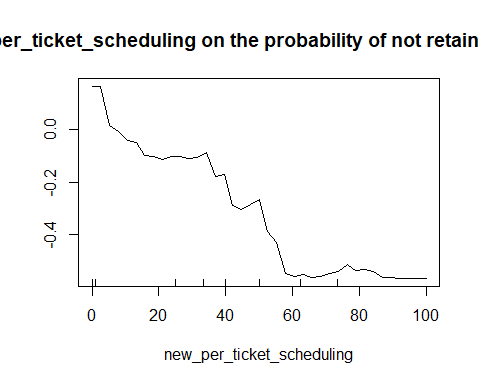
varUsed(proposed.training.model.3m)

## [1] 4093 9340 9273 6096 1752 14407 1638 1629 6455 1693 6616 557  
## [13] 6760

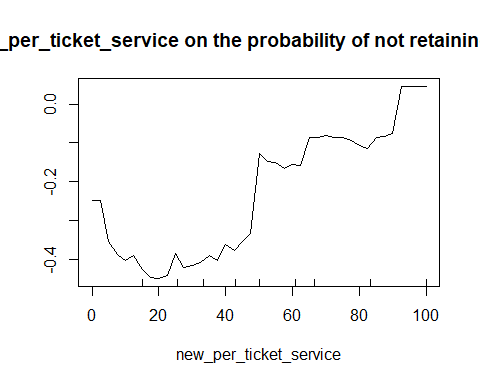
# Examining the model, it seems that num\_emails\_month played the most important role in predicting outcomes follwed by the percent compositions from each of the non-billing email interactions ( CX > scheduling > service).   
  
# Step 5a: Examining the effects of each variable on retention status (Top 4 predictors)  
  
partialPlot(proposed.training.model.3m, clean\_bang\_retention\_3m.test, num\_emails\_month, "no", main = "Marginal Effect of num\_emails\_month on the probability of not retaining membership at 3-Months")



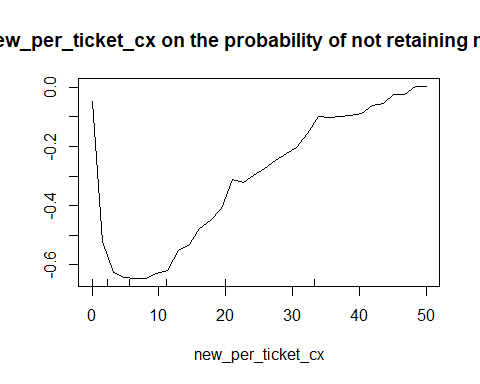
partialPlot(proposed.training.model.3m, clean\_bang\_retention\_3m.test, new\_per\_ticket\_scheduling, "no", main = "Marginal Effect of new\_per\_ticket\_scheduling on the probability of not retaining membership at 3-Months")



partialPlot(proposed.training.model.3m, clean\_bang\_retention\_3m.test, new\_per\_ticket\_service, "no", main = "Marginal Effect of new\_per\_ticket\_service on the probability of not retaining membership at 3-Months")



partialPlot(proposed.training.model.3m, clean\_bang\_retention\_3m.test, new\_per\_ticket\_cx, "no", main = "Marginal Effect of new\_per\_ticket\_cx on the probability of not retaining membership at 3-Months")



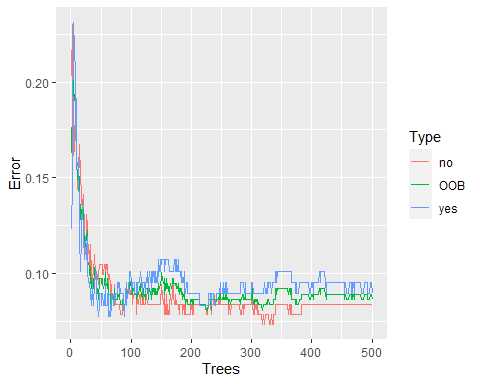
### RETENTION ANALYSIS: Membership status at 6-Months via Random Forest

Using the random survival forest specific dataset, I’ve split the data set 80:20 with respect to training:test. In forming the training model, which has an error rate of **7.24%**, it was found that the error rate in predicting membership length to churn with the test data was **7.52%**, which is not a very good sign. Looking at the various ways to modify the parameters, it was found that the error rate more-or-less stabilized after 1000 trees as evident by the marginal differences in error rates at the higher number of trees. However, in terms of tuning this model, I’ve adjusted the model to include ntree = 1000 and mtry at 3. Examining the importance of each variable used in this model, it was found that number of non-billing email interaction played the largest role, followed by the percent composition of non-billing related email interactions (scheduling, service and CX).

clean\_bang\_retention\_6m = clean\_bang\_select %>%   
 select(  
 age\_group,   
 employment\_sector,   
 retention\_6m,  
 avg\_monthly\_rate,  
 attendance\_grouping\_ver.1,  
 ever\_email\_month,  
 num\_emails\_month,  
 ever\_billing\_issue,   
 ever\_cx,  
 new\_per\_ticket\_cx,  
 ever\_scheduling,  
 new\_per\_ticket\_scheduling,  
 ever\_service,  
 new\_per\_ticket\_service  
 )  
  
# Step 2: create a partition of this data set by splitting it based on retention status at 6 Months  
  
trainIndex\_6m = createDataPartition(clean\_bang\_retention\_6m$retention\_6m, p = 0.8, list = FALSE)  
clean\_bang\_retention\_6m.train = clean\_bang\_retention\_6m[trainIndex\_6m,]   
clean\_bang\_retention\_6m.test = clean\_bang\_retention\_6m[-trainIndex\_6m,]   
  
# Step 3: Create a random forest model using training data   
  
training.model.6m = randomForest(retention\_6m ~., data = clean\_bang\_retention\_6m.train, proximity = T)  
training.model.6m # OOB error rate is 7.24%

##   
## Call:  
## randomForest(formula = retention\_6m ~ ., data = clean\_bang\_retention\_6m.train, proximity = T)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 8.64%  
## Confusion matrix:  
## no yes class.error  
## no 175 16 0.08376963  
## yes 15 153 0.08928571

# Step 4: Create a data frame to see how the error rate changes as a function of increasing number of trees (currently capped at 500)  
  
oob.error.data.6m = data.frame(  
 Trees = rep(1:nrow(training.model.6m$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.6m$err.rate)),  
 Error = c(training.model.6m$err.rate[, "OOB"],   
 training.model.6m$err.rate[,"no"],   
 training.model.6m$err.rate[, 'yes']))  
  
View(oob.error.data.6m)  
  
ggplot(data = oob.error.data.6m, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



# Step 4a: Add more trees and see what happens:   
  
training.model.6m\_ver1 = randomForest(retention\_6m ~., data = clean\_bang\_retention\_6m.train, proximity = T, ntree = 1000)  
training.model.6m\_ver2 = randomForest(retention\_6m ~., data = clean\_bang\_retention\_6m.train, proximity = T, ntree = 2000)  
training.model.6m\_ver3 = randomForest(retention\_6m ~., data = clean\_bang\_retention\_6m.train, proximity = T, ntree = 3000)  
  
training.model.6m # REFERENCE

##   
## Call:  
## randomForest(formula = retention\_6m ~ ., data = clean\_bang\_retention\_6m.train, proximity = T)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 8.64%  
## Confusion matrix:  
## no yes class.error  
## no 175 16 0.08376963  
## yes 15 153 0.08928571

training.model.6m\_ver1 # 9.19%

##   
## Call:  
## randomForest(formula = retention\_6m ~ ., data = clean\_bang\_retention\_6m.train, proximity = T, ntree = 1000)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 8.91%  
## Confusion matrix:  
## no yes class.error  
## no 174 17 0.08900524  
## yes 15 153 0.08928571

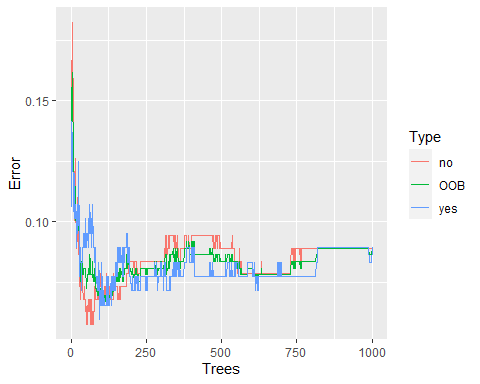
training.model.6m\_ver2 # 8.64%

##   
## Call:  
## randomForest(formula = retention\_6m ~ ., data = clean\_bang\_retention\_6m.train, proximity = T, ntree = 2000)   
## Type of random forest: classification  
## Number of trees: 2000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 8.91%  
## Confusion matrix:  
## no yes class.error  
## no 174 17 0.08900524  
## yes 15 153 0.08928571

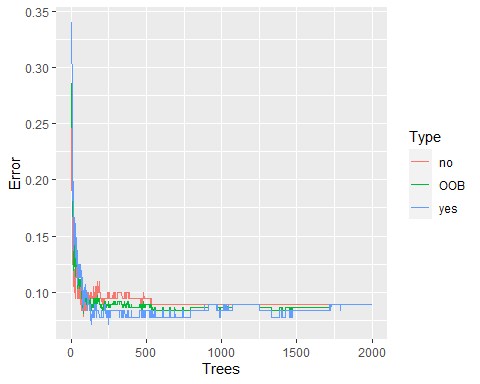
training.model.6m\_ver3 # 8.64%

##   
## Call:  
## randomForest(formula = retention\_6m ~ ., data = clean\_bang\_retention\_6m.train, proximity = T, ntree = 3000)   
## Type of random forest: classification  
## Number of trees: 3000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 8.64%  
## Confusion matrix:  
## no yes class.error  
## no 174 17 0.08900524  
## yes 14 154 0.08333333

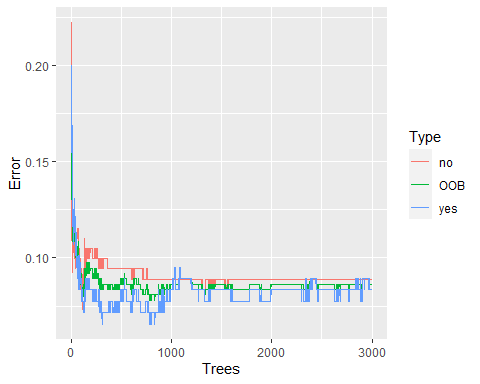
oob.error.data.6m\_ver1 = data.frame(  
 Trees = rep(1:nrow(training.model.6m\_ver1$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.6m\_ver1$err.rate)),  
 Error = c(training.model.6m\_ver1$err.rate[, "OOB"],   
 training.model.6m\_ver1$err.rate[,"no"],   
 training.model.6m\_ver1$err.rate[, 'yes']))  
  
oob.error.data.6m\_ver2 = data.frame(  
 Trees = rep(1:nrow(training.model.6m\_ver2$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.6m\_ver2$err.rate)),  
 Error = c(training.model.6m\_ver2$err.rate[, "OOB"],   
 training.model.6m\_ver2$err.rate[,"no"],   
 training.model.6m\_ver2$err.rate[, 'yes']))  
  
oob.error.data.6m\_ver3 = data.frame(  
 Trees = rep(1:nrow(training.model.6m\_ver3$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.6m\_ver3$err.rate)),  
 Error = c(training.model.6m\_ver3$err.rate[, "OOB"],   
 training.model.6m\_ver3$err.rate[,"no"],   
 training.model.6m\_ver3$err.rate[, 'yes']))  
  
ggplot(data = oob.error.data.6m\_ver1, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



ggplot(data = oob.error.data.6m\_ver2, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



ggplot(data = oob.error.data.6m\_ver3, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



# Looks like we did a worse job with increasing number of trees, but this leveled off after 2000.   
  
# STEP 3B: Fine tuning mtry   
  
oob.values <- vector(length = 10)  
for(i in 1:10) {  
 temp.model <- randomForest(retention\_6m ~., data = clean\_bang\_retention\_6m.train, mtry = i, ntree = 1000)  
 oob.values[i] <- temp.model$err.rate[nrow(temp.model$err.rate), 1]  
}  
  
oob.values

## [1] 0.13370474 0.09192201 0.08913649 0.08635097 0.09192201 0.08356546  
## [7] 0.08635097 0.10027855 0.10584958 0.11142061

# Looks like optimal value is 4  
  
proposed.training.model.6m = randomForest(retention\_6m ~., data = clean\_bang\_retention\_6m.train, proximity = T, mtry = 3, ntree = 1000) # err.rate = 7.52%  
  
proposed.training.model.6m$confusion

## no yes class.error  
## no 175 16 0.08376963  
## yes 13 155 0.07738095

# Step 4: Test this proposed model against testing data   
  
pred\_6m\_rf <- predict(proposed.training.model.6m, newdata = clean\_bang\_retention\_6m.test)  
head(pred\_6m\_rf)

## 9 10 14 24 26 27   
## yes yes yes yes yes yes   
## Levels: no yes

head(clean\_bang\_retention\_6m.test$retention\_6m)

## [1] yes yes yes no yes yes  
## Levels: no yes

cbind(pred\_6m\_rf, clean\_bang\_retention\_6m.test$retention\_6m)

## pred\_6m\_rf   
## 9 2 2  
## 10 2 2  
## 14 2 2  
## 24 2 1  
## 26 2 2  
## 27 2 2  
## 39 2 2  
## 53 1 2  
## 61 1 1  
## 63 1 1  
## 65 2 2  
## 69 2 2  
## 70 1 2  
## 82 1 1  
## 84 1 1  
## 85 2 2  
## 90 2 2  
## 92 2 2  
## 101 2 2  
## 104 2 2  
## 105 1 1  
## 108 1 1  
## 109 2 2  
## 110 2 2  
## 111 1 1  
## 114 2 2  
## 122 2 2  
## 124 2 2  
## 126 1 1  
## 128 2 2  
## 133 2 1  
## 141 2 2  
## 154 2 2  
## 156 1 1  
## 157 1 1  
## 163 2 2  
## 165 1 1  
## 168 1 1  
## 176 2 2  
## 177 1 1  
## 180 1 1  
## 181 2 2  
## 182 2 2  
## 186 1 1  
## 194 2 2  
## 198 2 2  
## 210 1 1  
## 213 1 1  
## 215 1 1  
## 220 2 2  
## 225 1 1  
## 237 2 2  
## 238 2 2  
## 247 1 1  
## 253 1 1  
## 258 2 2  
## 263 1 1  
## 264 1 1  
## 272 2 2  
## 280 2 2  
## 293 1 1  
## 297 1 1  
## 301 1 1  
## 303 1 1  
## 305 2 2  
## 327 1 1  
## 333 1 1  
## 335 1 1  
## 347 1 1  
## 353 2 2  
## 361 1 1  
## 362 1 1  
## 371 2 1  
## 373 2 1  
## 374 1 1  
## 377 2 1  
## 381 1 1  
## 398 2 1  
## 410 1 1  
## 418 1 1  
## 420 2 2  
## 427 2 2  
## 431 1 1  
## 433 2 2  
## 437 2 2  
## 438 2 1  
## 440 1 1  
## 443 1 1

results\_6m = data.frame(  
 individuals = rep(1:nrow(clean\_bang\_retention\_6m.test)),  
 prediction = pred\_6m\_rf,  
 truth = clean\_bang\_retention\_6m.test$retention\_6m  
)  
results\_6m

## individuals prediction truth  
## 9 1 yes yes  
## 10 2 yes yes  
## 14 3 yes yes  
## 24 4 yes no  
## 26 5 yes yes  
## 27 6 yes yes  
## 39 7 yes yes  
## 53 8 no yes  
## 61 9 no no  
## 63 10 no no  
## 65 11 yes yes  
## 69 12 yes yes  
## 70 13 no yes  
## 82 14 no no  
## 84 15 no no  
## 85 16 yes yes  
## 90 17 yes yes  
## 92 18 yes yes  
## 101 19 yes yes  
## 104 20 yes yes  
## 105 21 no no  
## 108 22 no no  
## 109 23 yes yes  
## 110 24 yes yes  
## 111 25 no no  
## 114 26 yes yes  
## 122 27 yes yes  
## 124 28 yes yes  
## 126 29 no no  
## 128 30 yes yes  
## 133 31 yes no  
## 141 32 yes yes  
## 154 33 yes yes  
## 156 34 no no  
## 157 35 no no  
## 163 36 yes yes  
## 165 37 no no  
## 168 38 no no  
## 176 39 yes yes  
## 177 40 no no  
## 180 41 no no  
## 181 42 yes yes  
## 182 43 yes yes  
## 186 44 no no  
## 194 45 yes yes  
## 198 46 yes yes  
## 210 47 no no  
## 213 48 no no  
## 215 49 no no  
## 220 50 yes yes  
## 225 51 no no  
## 237 52 yes yes  
## 238 53 yes yes  
## 247 54 no no  
## 253 55 no no  
## 258 56 yes yes  
## 263 57 no no  
## 264 58 no no  
## 272 59 yes yes  
## 280 60 yes yes  
## 293 61 no no  
## 297 62 no no  
## 301 63 no no  
## 303 64 no no  
## 305 65 yes yes  
## 327 66 no no  
## 333 67 no no  
## 335 68 no no  
## 347 69 no no  
## 353 70 yes yes  
## 361 71 no no  
## 362 72 no no  
## 371 73 yes no  
## 373 74 yes no  
## 374 75 no no  
## 377 76 yes no  
## 381 77 no no  
## 398 78 yes no  
## 410 79 no no  
## 418 80 no no  
## 420 81 yes yes  
## 427 82 yes yes  
## 431 83 no no  
## 433 84 yes yes  
## 437 85 yes yes  
## 438 86 yes no  
## 440 87 no no  
## 443 88 no no

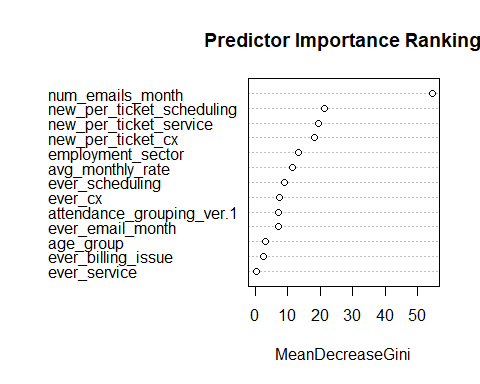
confusionMatrix(pred\_6m\_rf, clean\_bang\_retention\_6m.test$retention\_6m) # accuracy = 0.8295 or err.rate of 17.05%

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 40 2  
## yes 7 39  
##   
## Accuracy : 0.8977   
## 95% CI : (0.8147, 0.9522)  
## No Information Rate : 0.5341   
## P-Value [Acc > NIR] : 2.051e-13   
##   
## Kappa : 0.7961   
##   
## Mcnemar's Test P-Value : 0.1824   
##   
## Sensitivity : 0.8511   
## Specificity : 0.9512   
## Pos Pred Value : 0.9524   
## Neg Pred Value : 0.8478   
## Prevalence : 0.5341   
## Detection Rate : 0.4545   
## Detection Prevalence : 0.4773   
## Balanced Accuracy : 0.9011   
##   
## 'Positive' Class : no   
##

# STEP 5: Determining which variables are important predictors   
  
varImp(proposed.training.model.6m)

## Overall  
## age\_group 3.253068  
## employment\_sector 13.255506  
## avg\_monthly\_rate 11.464820  
## attendance\_grouping\_ver.1 7.061348  
## ever\_email\_month 7.027686  
## num\_emails\_month 54.578905  
## ever\_billing\_issue 2.362975  
## ever\_cx 7.443069  
## new\_per\_ticket\_cx 18.160248  
## ever\_scheduling 8.904537  
## new\_per\_ticket\_scheduling 21.260068  
## ever\_service 0.475466  
## new\_per\_ticket\_service 19.306840

varImpPlot(proposed.training.model.6m, sort = T, main = "Predictor Importance Ranking")



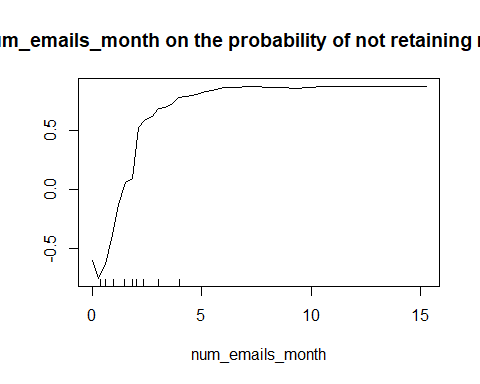
importance(proposed.training.model.6m)

## MeanDecreaseGini  
## age\_group 3.253068  
## employment\_sector 13.255506  
## avg\_monthly\_rate 11.464820  
## attendance\_grouping\_ver.1 7.061348  
## ever\_email\_month 7.027686  
## num\_emails\_month 54.578905  
## ever\_billing\_issue 2.362975  
## ever\_cx 7.443069  
## new\_per\_ticket\_cx 18.160248  
## ever\_scheduling 8.904537  
## new\_per\_ticket\_scheduling 21.260068  
## ever\_service 0.475466  
## new\_per\_ticket\_service 19.306840

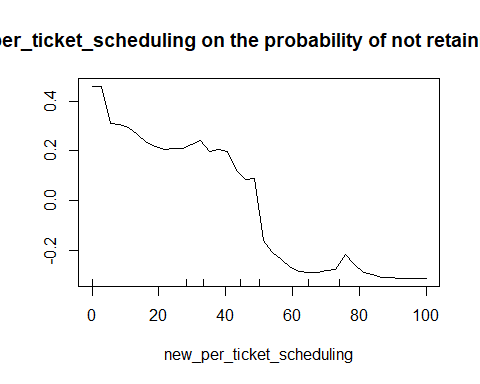
varUsed(proposed.training.model.6m)

## [1] 2268 4508 4719 3371 1213 6772 1281 1073 3566 1096 4028 236 4237

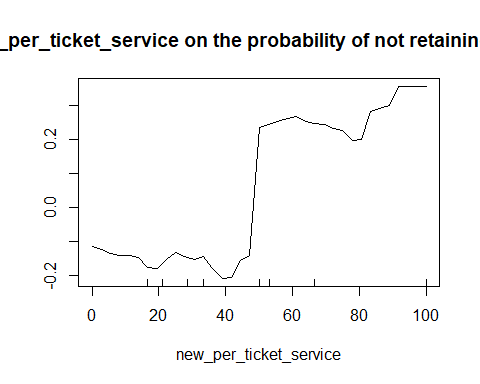
# Examining the model, it seems that num\_emails\_month played the most important role in predicting outcomes follwed by the percent compositions from each of the non-billing email interactions (scheduling > service > CX).   
  
# Step 5a: Examining the effects of each variable on retention status (Top 4 predictors)  
  
partialPlot(proposed.training.model.6m, clean\_bang\_retention\_6m.test, num\_emails\_month, "no", main = "Marginal Effect of num\_emails\_month on the probability of not retaining membership at 6-Months")



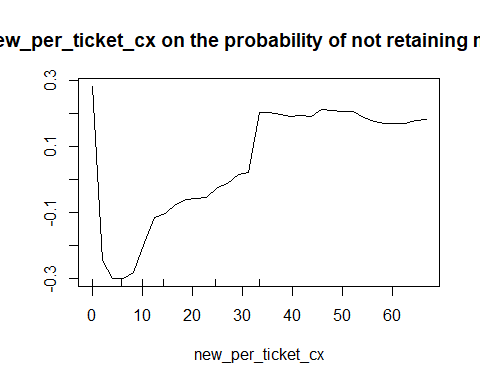
partialPlot(proposed.training.model.6m, clean\_bang\_retention\_6m.test, new\_per\_ticket\_scheduling, "no", main = "Marginal Effect of new\_per\_ticket\_scheduling on the probability of not retaining membership at 6-Months")



partialPlot(proposed.training.model.6m, clean\_bang\_retention\_6m.test, new\_per\_ticket\_service, "no", main = "Marginal Effect of new\_per\_ticket\_service on the probability of not retaining membership at 6-Months")



partialPlot(proposed.training.model.6m, clean\_bang\_retention\_6m.test, new\_per\_ticket\_cx, "no", main = "Marginal Effect of new\_per\_ticket\_cx on the probability of not retaining membership at 6-Months")



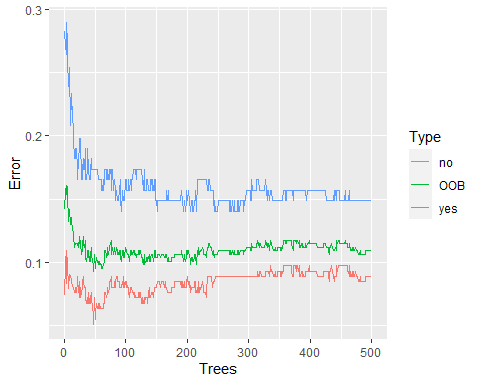
### RETENTION ANALYSIS: Membership status at 12-Months via Random Forest

Using the random survival forest specific data set, I’ve split the data set 80:20 with respect to training:test. In forming the training model, which has an error rate of **12.01**, it was found that the error rate in predicting membership length to churn with the test data was **12.36%**. Looking at the various ways to modify the parameters, it was found that the error rate more-or-less stabilized after 1000 trees as evident by the marginal differences in error rates at the higher number of trees. However, in terms of tuning this model, I’ve adjusted the model to include ntree = 2000 and mtry at 3. Examining the importance of each variable used in this model, it was found that number of non-billing email interaction played the largest role, followed by the percent composition of non-billing related email interactions (scheduling, service and CX).

clean\_bang\_retention\_12m = clean\_bang\_select %>%   
 select(  
 age\_group,   
 employment\_sector,   
 retention\_12m,  
 avg\_monthly\_rate,  
 attendance\_grouping\_ver.1,  
 ever\_email\_month,  
 num\_emails\_month,  
 ever\_billing\_issue,   
 ever\_cx,  
 new\_per\_ticket\_cx,  
 ever\_scheduling,  
 new\_per\_ticket\_scheduling,  
 ever\_service,  
 new\_per\_ticket\_service  
 )  
  
# Step 2: create a partition of this data set by splitting it based on retention status at 12 Months  
  
trainIndex\_12m = createDataPartition(clean\_bang\_retention\_12m$retention\_12m, p = 0.8, list = FALSE)  
clean\_bang\_retention\_12m.train = clean\_bang\_retention\_12m[trainIndex\_12m,]   
clean\_bang\_retention\_12m.test = clean\_bang\_retention\_12m[-trainIndex\_12m,]   
  
# Step 3: Create a random forest model using training data   
  
training.model.12m = randomForest(retention\_12m ~., data = clean\_bang\_retention\_12m.train, proximity = T)  
training.model.12m # OOB error rate is 12.85%

##   
## Call:  
## randomForest(formula = retention\_12m ~ ., data = clean\_bang\_retention\_12m.train, proximity = T)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 10.89%  
## Confusion matrix:  
## no yes class.error  
## no 216 21 0.08860759  
## yes 18 103 0.14876033

# Step 4: Create a data frame to see how the error rate changes as a function of increasing number of trees (currently capped at 500)  
  
oob.error.data.12m = data.frame(  
 Trees = rep(1:nrow(training.model.12m$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.12m$err.rate)),  
 Error = c(training.model.12m$err.rate[, "OOB"],   
 training.model.12m$err.rate[,"no"],   
 training.model.12m$err.rate[, 'yes']))  
  
View(oob.error.data.12m)  
  
ggplot(data = oob.error.data.12m, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



# Step 4a: Add more trees and see what happens:   
  
training.model.12m\_ver1 = randomForest(retention\_12m ~., data = clean\_bang\_retention\_12m.train, proximity = T, ntree = 1000)  
training.model.12m\_ver2 = randomForest(retention\_12m ~., data = clean\_bang\_retention\_12m.train, proximity = T, ntree = 2000)  
training.model.12m\_ver3 = randomForest(retention\_12m ~., data = clean\_bang\_retention\_12m.train, proximity = T, ntree = 3000)  
  
training.model.12m # REFERENCE

##   
## Call:  
## randomForest(formula = retention\_12m ~ ., data = clean\_bang\_retention\_12m.train, proximity = T)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 10.89%  
## Confusion matrix:  
## no yes class.error  
## no 216 21 0.08860759  
## yes 18 103 0.14876033

training.model.12m\_ver1 # 12.85%

##   
## Call:  
## randomForest(formula = retention\_12m ~ ., data = clean\_bang\_retention\_12m.train, proximity = T, ntree = 1000)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 11.17%  
## Confusion matrix:  
## no yes class.error  
## no 215 22 0.0928270  
## yes 18 103 0.1487603

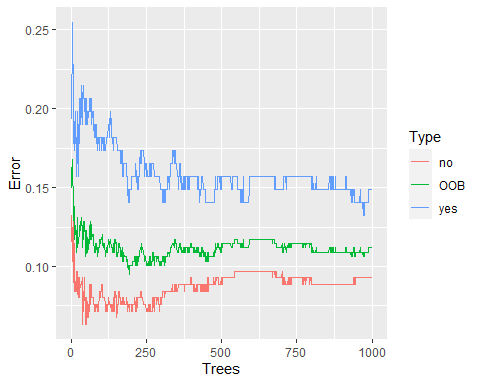
training.model.12m\_ver2 # 11.01%

##   
## Call:  
## randomForest(formula = retention\_12m ~ ., data = clean\_bang\_retention\_12m.train, proximity = T, ntree = 2000)   
## Type of random forest: classification  
## Number of trees: 2000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 10.61%  
## Confusion matrix:  
## no yes class.error  
## no 216 21 0.08860759  
## yes 17 104 0.14049587

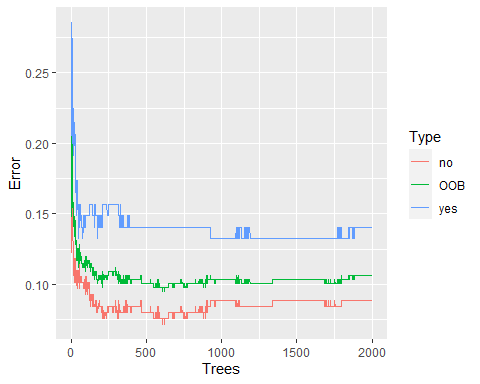
training.model.12m\_ver3 # 11.01%

##   
## Call:  
## randomForest(formula = retention\_12m ~ ., data = clean\_bang\_retention\_12m.train, proximity = T, ntree = 3000)   
## Type of random forest: classification  
## Number of trees: 3000  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 10.61%  
## Confusion matrix:  
## no yes class.error  
## no 216 21 0.08860759  
## yes 17 104 0.14049587

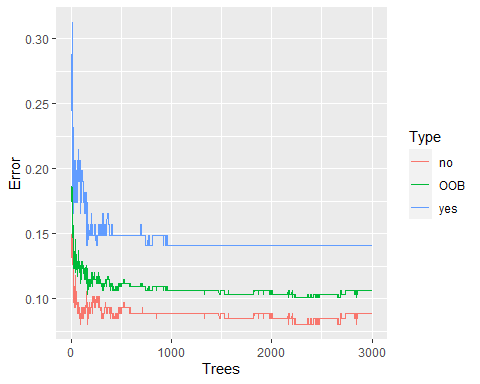
oob.error.data.12m\_ver1 = data.frame(  
 Trees = rep(1:nrow(training.model.12m\_ver1$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.12m\_ver1$err.rate)),  
 Error = c(training.model.12m\_ver1$err.rate[, "OOB"],   
 training.model.12m\_ver1$err.rate[,"no"],   
 training.model.12m\_ver1$err.rate[, 'yes']))  
  
oob.error.data.12m\_ver2 = data.frame(  
 Trees = rep(1:nrow(training.model.12m\_ver2$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.12m\_ver2$err.rate)),  
 Error = c(training.model.12m\_ver2$err.rate[, "OOB"],   
 training.model.12m\_ver2$err.rate[,"no"],   
 training.model.12m\_ver2$err.rate[, 'yes']))  
  
oob.error.data.12m\_ver3 = data.frame(  
 Trees = rep(1:nrow(training.model.12m\_ver3$err.rate), times = 3),  
 Type = rep(c("OOB", "no", 'yes'), each = nrow(training.model.12m\_ver3$err.rate)),  
 Error = c(training.model.12m\_ver3$err.rate[, "OOB"],   
 training.model.12m\_ver3$err.rate[,"no"],   
 training.model.12m\_ver3$err.rate[, 'yes']))  
  
ggplot(data = oob.error.data.12m\_ver1, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



ggplot(data = oob.error.data.12m\_ver2, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



ggplot(data = oob.error.data.12m\_ver3, aes (x = Trees, y = Error)) + geom\_line(aes(color = Type))



# Looks like we did a worse job with increasing number of trees, but this leveled off after 1000.   
  
# STEP 3B: Fine tuning mtry   
  
oob.values <- vector(length = 10)  
for(i in 1:10) {  
 temp.model <- randomForest(retention\_12m ~., data = clean\_bang\_retention\_12m.train, mtry = i, ntree = 1000)  
 oob.values[i] <- temp.model$err.rate[nrow(temp.model$err.rate), 1]  
}  
  
oob.values

## [1] 0.12849162 0.11452514 0.10893855 0.10614525 0.10893855 0.10893855  
## [7] 0.10335196 0.09776536 0.11173184 0.10055866

# Looks like optimal value is 8  
  
proposed.training.model.12m = randomForest(retention\_12m ~., data = clean\_bang\_retention\_12m.train, proximity = T, mtry = 8, ntree = 1000) # err.rate = 13.41%  
  
proposed.training.model.12m$confusion

## no yes class.error  
## no 217 20 0.08438819  
## yes 18 103 0.14876033

# Step 4: Test this proposed model against testing data   
  
pred\_12m\_rf <- predict(proposed.training.model.12m, newdata = clean\_bang\_retention\_12m.test)  
head(pred\_12m\_rf)

## 3 9 12 18 21 32   
## no no no no yes no   
## Levels: no yes

head(clean\_bang\_retention\_12m.test$retention\_12m)

## [1] no yes no no yes no   
## Levels: no yes

cbind(pred\_12m\_rf, clean\_bang\_retention\_12m.test$retention\_12m)

## pred\_12m\_rf   
## 3 1 1  
## 9 1 2  
## 12 1 1  
## 18 1 1  
## 21 2 2  
## 32 1 1  
## 35 2 2  
## 49 1 1  
## 55 1 1  
## 56 2 2  
## 62 2 2  
## 64 1 1  
## 71 1 1  
## 74 1 1  
## 78 1 2  
## 84 1 1  
## 88 1 1  
## 93 2 2  
## 100 1 1  
## 117 1 1  
## 130 1 1  
## 142 1 1  
## 145 1 1  
## 151 1 1  
## 162 2 2  
## 168 1 1  
## 169 2 2  
## 170 1 2  
## 173 1 2  
## 176 1 2  
## 180 1 1  
## 183 2 2  
## 188 1 1  
## 191 1 1  
## 195 1 1  
## 200 1 1  
## 201 1 1  
## 208 1 1  
## 212 1 1  
## 213 1 1  
## 214 2 2  
## 216 2 2  
## 220 1 2  
## 225 1 1  
## 228 1 1  
## 236 1 1  
## 238 2 2  
## 241 1 1  
## 245 1 1  
## 249 1 1  
## 257 1 1  
## 272 1 2  
## 280 2 2  
## 281 2 2  
## 291 1 1  
## 292 2 2  
## 294 2 2  
## 307 1 1  
## 311 1 1  
## 315 1 1  
## 320 1 1  
## 329 2 1  
## 330 2 2  
## 336 1 1  
## 347 1 1  
## 355 2 2  
## 368 1 2  
## 378 1 1  
## 382 1 1  
## 389 1 1  
## 390 1 1  
## 396 1 1  
## 402 1 1  
## 403 2 2  
## 405 1 1  
## 407 1 2  
## 408 1 1  
## 413 1 1  
## 421 1 1  
## 422 1 1  
## 423 2 1  
## 428 1 1  
## 431 1 1  
## 432 1 1  
## 433 2 2  
## 437 2 2  
## 442 1 1  
## 444 1 1  
## 445 2 2

results\_12m = data.frame(  
 individuals = rep(1:nrow(clean\_bang\_retention\_12m.test)),  
 prediction = pred\_12m\_rf,  
 truth = clean\_bang\_retention\_12m.test$retention\_12m  
)  
results\_12m

## individuals prediction truth  
## 3 1 no no  
## 9 2 no yes  
## 12 3 no no  
## 18 4 no no  
## 21 5 yes yes  
## 32 6 no no  
## 35 7 yes yes  
## 49 8 no no  
## 55 9 no no  
## 56 10 yes yes  
## 62 11 yes yes  
## 64 12 no no  
## 71 13 no no  
## 74 14 no no  
## 78 15 no yes  
## 84 16 no no  
## 88 17 no no  
## 93 18 yes yes  
## 100 19 no no  
## 117 20 no no  
## 130 21 no no  
## 142 22 no no  
## 145 23 no no  
## 151 24 no no  
## 162 25 yes yes  
## 168 26 no no  
## 169 27 yes yes  
## 170 28 no yes  
## 173 29 no yes  
## 176 30 no yes  
## 180 31 no no  
## 183 32 yes yes  
## 188 33 no no  
## 191 34 no no  
## 195 35 no no  
## 200 36 no no  
## 201 37 no no  
## 208 38 no no  
## 212 39 no no  
## 213 40 no no  
## 214 41 yes yes  
## 216 42 yes yes  
## 220 43 no yes  
## 225 44 no no  
## 228 45 no no  
## 236 46 no no  
## 238 47 yes yes  
## 241 48 no no  
## 245 49 no no  
## 249 50 no no  
## 257 51 no no  
## 272 52 no yes  
## 280 53 yes yes  
## 281 54 yes yes  
## 291 55 no no  
## 292 56 yes yes  
## 294 57 yes yes  
## 307 58 no no  
## 311 59 no no  
## 315 60 no no  
## 320 61 no no  
## 329 62 yes no  
## 330 63 yes yes  
## 336 64 no no  
## 347 65 no no  
## 355 66 yes yes  
## 368 67 no yes  
## 378 68 no no  
## 382 69 no no  
## 389 70 no no  
## 390 71 no no  
## 396 72 no no  
## 402 73 no no  
## 403 74 yes yes  
## 405 75 no no  
## 407 76 no yes  
## 408 77 no no  
## 413 78 no no  
## 421 79 no no  
## 422 80 no no  
## 423 81 yes no  
## 428 82 no no  
## 431 83 no no  
## 432 84 no no  
## 433 85 yes yes  
## 437 86 yes yes  
## 442 87 no no  
## 444 88 no no  
## 445 89 yes yes

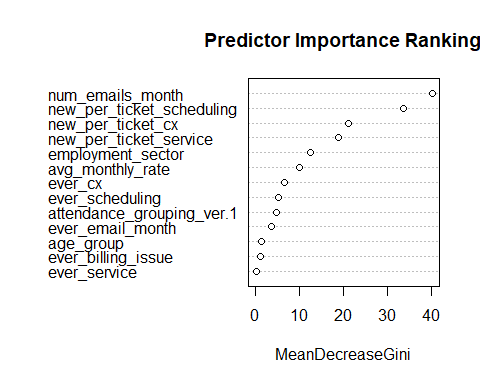
confusionMatrix(pred\_12m\_rf, clean\_bang\_retention\_12m.test$retention\_12m) # accuracy = 0.9326 or err.rate of 6.74%

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 57 9  
## yes 2 21  
##   
## Accuracy : 0.8764   
## 95% CI : (0.7896, 0.9367)  
## No Information Rate : 0.6629   
## P-Value [Acc > NIR] : 3.758e-06   
##   
## Kappa : 0.7066   
##   
## Mcnemar's Test P-Value : 0.07044   
##   
## Sensitivity : 0.9661   
## Specificity : 0.7000   
## Pos Pred Value : 0.8636   
## Neg Pred Value : 0.9130   
## Prevalence : 0.6629   
## Detection Rate : 0.6404   
## Detection Prevalence : 0.7416   
## Balanced Accuracy : 0.8331   
##   
## 'Positive' Class : no   
##

# STEP 5: Determining which variables are important predictors   
  
varImp(proposed.training.model.12m)

## Overall  
## age\_group 1.4262676  
## employment\_sector 12.6415438  
## avg\_monthly\_rate 10.0441763  
## attendance\_grouping\_ver.1 4.7512721  
## ever\_email\_month 3.7767475  
## num\_emails\_month 40.3538007  
## ever\_billing\_issue 1.0745443  
## ever\_cx 6.6241465  
## new\_per\_ticket\_cx 21.1135284  
## ever\_scheduling 5.1853804  
## new\_per\_ticket\_scheduling 33.7148842  
## ever\_service 0.2897049  
## new\_per\_ticket\_service 18.8905452

varImpPlot(proposed.training.model.12m, sort = T, main = "Predictor Importance Ranking")



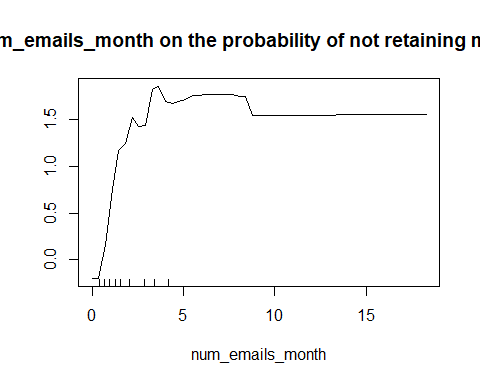
importance(proposed.training.model.12m)

## MeanDecreaseGini  
## age\_group 1.4262676  
## employment\_sector 12.6415438  
## avg\_monthly\_rate 10.0441763  
## attendance\_grouping\_ver.1 4.7512721  
## ever\_email\_month 3.7767475  
## num\_emails\_month 40.3538007  
## ever\_billing\_issue 1.0745443  
## ever\_cx 6.6241465  
## new\_per\_ticket\_cx 21.1135284  
## ever\_scheduling 5.1853804  
## new\_per\_ticket\_scheduling 33.7148842  
## ever\_service 0.2897049  
## new\_per\_ticket\_service 18.8905452

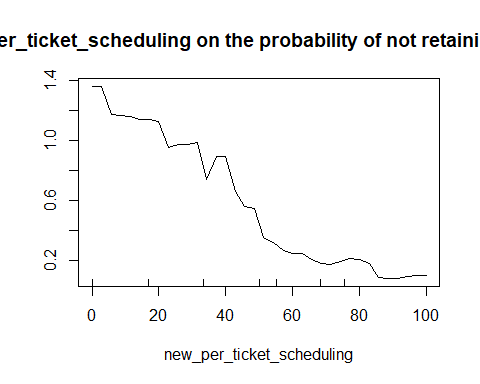
varUsed(proposed.training.model.12m)

## [1] 698 3986 3468 1974 461 5499 489 509 3372 435 3791 54 3600

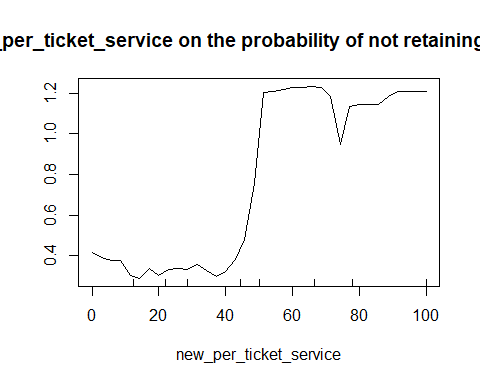
# Examining the model, it seems that num\_emails\_month played the most important role in predicting outcomes follwed by the percent compositions from each of the non-billing email interactions (scheduling > CX > service).   
  
# Step 5a: Examining the effects of each variable on retention status (Top 4 predictors)  
  
partialPlot(proposed.training.model.12m, clean\_bang\_retention\_12m.test, num\_emails\_month, "no", main = "Marginal Effect of num\_emails\_month on the probability of not retaining membership at 12-Months")



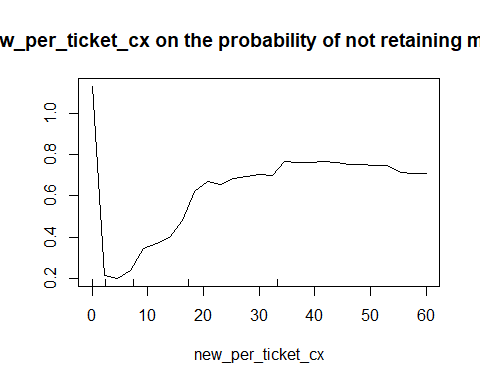
partialPlot(proposed.training.model.12m, clean\_bang\_retention\_12m.test, new\_per\_ticket\_scheduling, "no", main = "Marginal Effect of new\_per\_ticket\_scheduling on the probability of not retaining membership at 12-Months")



partialPlot(proposed.training.model.12m, clean\_bang\_retention\_12m.test, new\_per\_ticket\_service, "no", main = "Marginal Effect of new\_per\_ticket\_service on the probability of not retaining membership at 12-Months")



partialPlot(proposed.training.model.12m, clean\_bang\_retention\_12m.test, new\_per\_ticket\_cx, "no", main = "Marginal Effect of new\_per\_ticket\_cx on the probability of not retaining membership at 12-Months")



### RETENTION ANALYSIS: 3 Months via Logistic Regression

In generating a logistic regression model for membership status at 3-month, it was found that the variables that were retained through bi-directional stepwise regression through partitioned data (80-:20) were \* number of non-billing email interactions per month \* status of ever having a CX-related email interaction \* status of ever having a non-billing email interaction \* percent composition of total interactions being CX-related email interactions \* percent composition of total interactions being scheduling-related email interactions \* status of ever having a service-related email interaction

In cross-validating the proposed model through the validation set approach as well as repeated K-fold validation that the accuracy of the model ranged b/t **86.36% - 91.60%**. The major predictors turned out to be num\_emails\_month, new\_per\_ticket\_scheduling, ever\_cx and new\_per\_ticket\_cx.

# Step 1: Partition data  
  
trainIndex\_3m = createDataPartition(clean\_bang\_select$retention\_3m, p = 0.8, list = F)  
  
clean\_bang\_select.3m\_train = clean\_bang\_select[trainIndex\_3m,] # This is the Training Data (80% of the data)  
clean\_bang\_select.3m\_test = clean\_bang\_select[-trainIndex\_3m,] # This is the Testing Data (20% of the data)  
  
  
# Step 2: Bi-directional Stepwise regression   
  
model.start.train\_3m = glm(retention\_3m ~ 1, data = clean\_bang\_select.3m\_train , family = binomial(link = 'logit'))  
model.all.train\_3m = glm(retention\_3m ~ age\_group +   
 employment\_sector +   
 membership +   
 attendance\_grouping\_ver.1 +   
 monthly\_rate\_group +   
 ever\_email\_month +   
 num\_emails\_month +   
 ever\_cx+  
 new\_per\_ticket\_cx +   
 ever\_service+  
 new\_per\_ticket\_service +  
 ever\_scheduling +  
 new\_per\_ticket\_scheduling,   
 data = clean\_bang\_select.3m\_train , family = binomial(link = 'logit'))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

step(model.start.train\_3m, direction = 'both', scope = formula(model.all.train\_3m))

## Start: AIC=496.64  
## retention\_3m ~ 1  
##   
## Df Deviance AIC  
## + num\_emails\_month 1 362.20 366.20  
## + ever\_cx 1 436.58 440.58  
## + new\_per\_ticket\_scheduling 1 441.74 445.74  
## + ever\_scheduling 1 446.66 450.66  
## + new\_per\_ticket\_service 1 448.08 452.08  
## + monthly\_rate\_group 11 441.15 465.15  
## + ever\_email\_month 1 467.06 471.06  
## + attendance\_grouping\_ver.1 5 477.84 489.84  
## + membership 6 481.11 495.11  
## + age\_group 4 486.61 496.61  
## <none> 494.64 496.64  
## + new\_per\_ticket\_cx 1 494.27 498.27  
## + ever\_service 1 494.64 498.64  
## + employment\_sector 15 470.49 502.49  
##   
## Step: AIC=366.2  
## retention\_3m ~ num\_emails\_month

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + new\_per\_ticket\_scheduling 1 241.15 247.15  
## + ever\_scheduling 1 267.19 273.19  
## + new\_per\_ticket\_service 1 293.67 299.67  
## + ever\_cx 1 300.33 306.33  
## + ever\_service 1 355.36 361.36  
## + monthly\_rate\_group 11 336.13 362.13  
## <none> 362.20 366.20  
## + new\_per\_ticket\_cx 1 360.68 366.68  
## + ever\_email\_month 1 361.61 367.61  
## + attendance\_grouping\_ver.1 5 355.90 369.90  
## + membership 6 353.95 369.95  
## + age\_group 4 359.28 371.28  
## + employment\_sector 15 343.51 377.51  
## - num\_emails\_month 1 494.64 496.64

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=247.15  
## retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + ever\_cx 1 198.37 206.37  
## + new\_per\_ticket\_cx 1 235.27 243.27  
## + ever\_service 1 236.41 244.41  
## + ever\_email\_month 1 238.07 246.07  
## <none> 241.15 247.15  
## + ever\_scheduling 1 240.07 248.07  
## + new\_per\_ticket\_service 1 241.06 249.06  
## + attendance\_grouping\_ver.1 5 235.63 251.63  
## + monthly\_rate\_group 11 224.94 252.94  
## + age\_group 4 239.31 253.31  
## + membership 6 235.68 253.68  
## + employment\_sector 15 230.62 266.62  
## - new\_per\_ticket\_scheduling 1 362.20 366.20  
## - num\_emails\_month 1 441.74 445.74

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=206.37  
## retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + new\_per\_ticket\_cx 1 168.36 178.36  
## + ever\_email\_month 1 186.96 196.96  
## + new\_per\_ticket\_service 1 191.47 201.47  
## <none> 198.37 206.37  
## + ever\_service 1 197.25 207.25  
## + ever\_scheduling 1 198.36 208.36  
## + membership 6 189.32 209.32  
## + attendance\_grouping\_ver.1 5 192.19 210.19  
## + age\_group 4 195.76 211.76  
## + monthly\_rate\_group 11 185.35 215.35  
## + employment\_sector 15 189.03 227.03  
## - ever\_cx 1 241.15 247.15  
## - new\_per\_ticket\_scheduling 1 300.33 306.33  
## - num\_emails\_month 1 392.17 398.17

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=178.36  
## retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + ever\_email\_month 1 160.05 172.05  
## + ever\_service 1 166.25 178.25  
## <none> 168.36 178.36  
## + new\_per\_ticket\_service 1 166.89 178.89  
## + ever\_scheduling 1 168.20 180.20  
## + attendance\_grouping\_ver.1 5 162.79 182.79  
## + membership 6 161.21 183.21  
## + age\_group 4 166.49 184.49  
## + monthly\_rate\_group 11 153.85 185.85  
## + employment\_sector 15 158.94 198.94  
## - new\_per\_ticket\_cx 1 198.37 206.37  
## - new\_per\_ticket\_scheduling 1 205.47 213.47  
## - ever\_cx 1 235.27 243.27  
## - num\_emails\_month 1 357.62 365.62

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=172.05  
## retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + new\_per\_ticket\_service 1 145.29 159.29  
## + ever\_service 1 148.83 162.83  
## <none> 160.05 172.05  
## + ever\_scheduling 1 160.05 174.05  
## + attendance\_grouping\_ver.1 5 155.38 177.38  
## + membership 6 153.89 177.89  
## + age\_group 4 157.96 177.96  
## - ever\_email\_month 1 168.36 178.36  
## + monthly\_rate\_group 11 146.01 180.01  
## + employment\_sector 15 152.01 194.01  
## - new\_per\_ticket\_cx 1 186.96 196.96  
## - new\_per\_ticket\_scheduling 1 200.94 210.94  
## - ever\_cx 1 230.64 240.64  
## - num\_emails\_month 1 290.56 300.56

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=159.29  
## retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month + new\_per\_ticket\_service

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## <none> 145.29 159.29  
## - new\_per\_ticket\_cx 1 147.95 159.95  
## + ever\_service 1 144.23 160.23  
## + ever\_scheduling 1 145.29 161.29  
## + attendance\_grouping\_ver.1 5 140.38 164.38  
## + membership 6 138.87 164.87  
## + age\_group 4 143.57 165.57  
## + monthly\_rate\_group 11 132.99 168.99  
## - new\_per\_ticket\_service 1 160.05 172.05  
## + employment\_sector 15 134.46 178.46  
## - ever\_email\_month 1 166.88 178.88  
## - new\_per\_ticket\_scheduling 1 189.11 201.11  
## - ever\_cx 1 219.82 231.82  
## - num\_emails\_month 1 277.65 289.65

##   
## Call: glm(formula = retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month + new\_per\_ticket\_service,   
## family = binomial(link = "logit"), data = clean\_bang\_select.3m\_train)  
##   
## Coefficients:  
## (Intercept) num\_emails\_month   
## 1.186e-11 -1.647e+00   
## new\_per\_ticket\_scheduling ever\_cxyes   
## 1.431e+02 6.812e+00   
## new\_per\_ticket\_cx ever\_email\_monthyes   
## 1.429e+02 -1.430e+04   
## new\_per\_ticket\_service   
## 1.430e+02   
##   
## Degrees of Freedom: 358 Total (i.e. Null); 352 Residual  
## Null Deviance: 494.6   
## Residual Deviance: 145.3 AIC: 159.3

model.retained.train\_3m = glm(retention\_3m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 new\_per\_ticket\_cx +   
 ever\_email\_month +   
 ever\_service,   
 family = binomial(link = "logit"), data = clean\_bang\_select.3m\_train)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# Step 3: Assessing the proposed model  
  
summary(model.retained.train\_3m)

##   
## Call:  
## glm(formula = retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month + ever\_service,   
## family = binomial(link = "logit"), data = clean\_bang\_select.3m\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2514 -0.2642 0.0266 0.1255 3.3670   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.18571 0.68750 0.270 0.78706   
## num\_emails\_month -1.66858 0.25730 -6.485 8.88e-11 \*\*\*  
## new\_per\_ticket\_scheduling 0.06047 0.01117 5.412 6.22e-08 \*\*\*  
## ever\_cxyes 6.30430 1.08245 5.824 5.74e-09 \*\*\*  
## new\_per\_ticket\_cx -0.11841 0.02829 -4.185 2.85e-05 \*\*\*  
## ever\_email\_monthyes -4.09016 1.41914 -2.882 0.00395 \*\*   
## ever\_serviceyes 3.85723 1.39342 2.768 0.00564 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 494.64 on 358 degrees of freedom  
## Residual deviance: 148.83 on 352 degrees of freedom  
## AIC: 162.83  
##   
## Number of Fisher Scoring iterations: 8

AIC(model.retained.train\_3m) # 153.79

## [1] 162.8266

exp(cbind(OR = coef(model.retained.train\_3m), confint.default(model.retained.train\_3m)))

## OR 2.5 % 97.5 %  
## (Intercept) 1.20407462 0.312936270 4.6328784  
## num\_emails\_month 0.18851368 0.113848649 0.3121461  
## new\_per\_ticket\_scheduling 1.06233668 1.039325821 1.0858570  
## ever\_cxyes 546.91711064 65.544235859 4563.6099345  
## new\_per\_ticket\_cx 0.88833497 0.840418461 0.9389834  
## ever\_email\_monthyes 0.01673656 0.001036799 0.2701705  
## ever\_serviceyes 47.33411138 3.083872067 726.5275768

varImp(model.retained.train\_3m, sort = T) # Biggest predictors = num\_emails\_month, new\_per\_ticket\_scheduling, ever\_cx, new\_per\_ticket\_cx

## Overall  
## num\_emails\_month 6.484953  
## new\_per\_ticket\_scheduling 5.412243  
## ever\_cxyes 5.824076  
## new\_per\_ticket\_cx 4.185332  
## ever\_email\_monthyes 2.882146  
## ever\_serviceyes 2.768183

# Step 4a: validating the proposed model   
  
pred\_3m\_log <- predict(model.retained.train\_3m, newdata = clean\_bang\_select.3m\_test)  
pred\_3m\_log = ifelse(pred\_3m\_log > 0.5, 'yes', 'no')  
table(pred\_3m\_log, clean\_bang\_select.3m\_test$retention\_3m)

##   
## pred\_3m\_log no yes  
## no 36 4  
## yes 4 44

accuracy = table(pred\_3m\_log, clean\_bang\_select.3m\_test[, "retention\_3m"])  
accuracy

##   
## pred\_3m\_log no yes  
## no 36 4  
## yes 4 44

sum(diag(accuracy))/sum(accuracy)

## [1] 0.9090909

mean(pred\_3m\_log == clean\_bang\_select.3m\_test$retention\_3m) # 86.36% accuracy (or err.rate = 13.64%)

## [1] 0.9090909

# Step 4b: repeated k-fold validation   
  
repeat\_ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)  
proposed.model.retained.3m = train(retention\_3m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 new\_per\_ticket\_cx +   
 ever\_email\_month +   
 ever\_service,   
 data = clean\_bang\_select,   
 method = 'glm',  
 family = 'binomial',  
 trControl = repeat\_ctrl, tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

proposed.model.retained.3m # accuracy = 91.6%

## Generalized Linear Model   
##   
## 447 samples  
## 6 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 403, 402, 403, 402, 402, 401, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9151354 0.8289641

# Step 4c: k-fold validation   
  
ctrl = trainControl(method = 'cv', number = 10)  
proposed.model.retained.3m = train(retention\_3m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 new\_per\_ticket\_cx +   
 ever\_email\_month +   
 ever\_service,  
 data = clean\_bang\_select,   
 method = 'glm',  
 family = 'binomial',  
 trControl = ctrl, tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

proposed.model.retained.3m # accuracy 91.48%

## Generalized Linear Model   
##   
## 447 samples  
## 6 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 402, 402, 403, 402, 403, 403, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9172661 0.8333126

# Step 5: sumamry of proposed model on original data set  
  
log.model.3m = glm(retention\_3m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 new\_per\_ticket\_cx +   
 ever\_email\_month +   
 ever\_service,   
 family = binomial(link = 'logit'), data = clean\_bang\_select)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(log.model.3m)

##   
## Call:  
## glm(formula = retention\_3m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month + ever\_service,   
## family = binomial(link = "logit"), data = clean\_bang\_select)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3234 -0.2408 0.0277 0.1306 3.5831   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.320314 0.650792 0.492 0.622585   
## num\_emails\_month -1.767266 0.247842 -7.131 9.99e-13 \*\*\*  
## new\_per\_ticket\_scheduling 0.059788 0.009958 6.004 1.93e-09 \*\*\*  
## ever\_cxyes 6.359700 0.968783 6.565 5.22e-11 \*\*\*  
## new\_per\_ticket\_cx -0.116918 0.024910 -4.694 2.68e-06 \*\*\*  
## ever\_email\_monthyes -3.422673 0.951603 -3.597 0.000322 \*\*\*  
## ever\_serviceyes 3.102725 1.046623 2.965 0.003032 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 615.91 on 446 degrees of freedom  
## Residual deviance: 180.21 on 440 degrees of freedom  
## AIC: 194.21  
##   
## Number of Fisher Scoring iterations: 8

exp(cbind(OR = coef(log.model.3m), confint.default(log.model.3m)))

## OR 2.5 % 97.5 %  
## (Intercept) 1.3775596 0.384728921 4.9324870  
## num\_emails\_month 0.1707994 0.105080669 0.2776194  
## new\_per\_ticket\_scheduling 1.0616119 1.041092452 1.0825359  
## ever\_cxyes 578.0730454 86.567190966 3860.2205069  
## new\_per\_ticket\_cx 0.8896585 0.847266306 0.9341717  
## ever\_email\_monthyes 0.0326251 0.005052965 0.2106480  
## ever\_serviceyes 22.2585223 2.861604241 173.1342884

exp(coef(log.model.3m))

## (Intercept) num\_emails\_month new\_per\_ticket\_scheduling   
## 1.3775596 0.1707994 1.0616119   
## ever\_cxyes new\_per\_ticket\_cx ever\_email\_monthyes   
## 578.0730454 0.8896585 0.0326251   
## ever\_serviceyes   
## 22.2585223

vif(log.model.3m) # A potential concern of collinearity regarding ever\_cx + new\_per\_ticket\_cx

## num\_emails\_month new\_per\_ticket\_scheduling ever\_cxyes   
## 2.139657 1.686200 5.845623   
## new\_per\_ticket\_cx ever\_email\_monthyes ever\_serviceyes   
## 5.362813 2.637056 2.419064

AIC(log.model.3m)

## [1] 194.2052

### RETENTION ANALYSIS: 6-Months via Logistic Regression

In generating a logistic regression model for membership status at 6-month, it was found that the variables that were retained through bi-directional stepwise regression through partitioned data (80-:20) were \* number of non-billing email interactions per month \* percent composition of scheduling-related email interactions \* Status of ever having a non-billing-related email interaction \* status of ever haivng a CX-related email interaction \* percent composition of service-related email interactions \* percent composition of CX-related email interactions

In cross-validating the proposed model through the validation set approach as well as repeated K-fold validation that the accuracy of the model ranged b/t **85.22% - 90.44%**. The major predictors turned out to be num\_emails\_month, new\_per\_ticket\_scheduling, ever\_cx and ever\_email\_month.

# Step 1: Partition data  
  
trainIndex\_6m = createDataPartition(clean\_bang\_select$retention\_6m, p = 0.8, list = F)  
  
clean\_bang\_select.6m\_train = clean\_bang\_select[trainIndex\_6m,] # This is the Training Data (80% of the data)  
clean\_bang\_select.6m\_test = clean\_bang\_select[-trainIndex\_6m,] # This is the Testing Data (20% of the data)  
  
  
# Step 2: Bi-directional Stepwise regression   
  
model.start.train\_6m = glm(retention\_6m ~ 1, data = clean\_bang\_select.6m\_train, family = binomial(link = 'logit'))  
model.all.train\_6m = glm(retention\_6m ~ age\_group +   
 employment\_sector +   
 membership +   
 attendance\_grouping\_ver.1 +   
 monthly\_rate\_group +   
 ever\_email\_month +   
 num\_emails\_month +   
 ever\_cx+  
 new\_per\_ticket\_cx +   
 ever\_service+  
 new\_per\_ticket\_service +  
 ever\_scheduling +  
 new\_per\_ticket\_scheduling,   
 data = clean\_bang\_select.6m\_train, family = binomial(link = 'logit'))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

step(model.start.train\_6m, direction = 'both', scope = formula(model.all.train\_6m))

## Start: AIC=498.21  
## retention\_6m ~ 1  
##   
## Df Deviance AIC  
## + num\_emails\_month 1 379.45 383.45  
## + new\_per\_ticket\_scheduling 1 418.28 422.28  
## + ever\_cx 1 423.82 427.82  
## + new\_per\_ticket\_service 1 425.25 429.25  
## + ever\_scheduling 1 427.52 431.52  
## + monthly\_rate\_group 11 434.17 458.17  
## + ever\_email\_month 1 479.15 483.15  
## + attendance\_grouping\_ver.1 5 479.96 491.96  
## + membership 6 480.13 494.13  
## <none> 496.21 498.21  
## + ever\_service 1 496.01 500.01  
## + new\_per\_ticket\_cx 1 496.16 500.16  
## + age\_group 4 492.92 502.92  
## + employment\_sector 15 471.03 503.03  
##   
## Step: AIC=383.45  
## retention\_6m ~ num\_emails\_month

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + new\_per\_ticket\_scheduling 1 218.69 224.69  
## + ever\_scheduling 1 261.27 267.27  
## + new\_per\_ticket\_service 1 275.22 281.22  
## + ever\_cx 1 289.74 295.74  
## + monthly\_rate\_group 11 339.01 365.01  
## + ever\_service 1 371.25 377.25  
## + attendance\_grouping\_ver.1 5 369.17 383.17  
## <none> 379.45 383.45  
## + membership 6 368.26 384.26  
## + ever\_email\_month 1 378.91 384.91  
## + new\_per\_ticket\_cx 1 379.45 385.45  
## + employment\_sector 15 354.81 388.81  
## + age\_group 4 378.19 390.19  
## - num\_emails\_month 1 496.21 498.21

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=224.69  
## retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + ever\_cx 1 172.88 180.88  
## + new\_per\_ticket\_cx 1 206.21 214.21  
## + ever\_service 1 215.73 223.73  
## + attendance\_grouping\_ver.1 5 208.69 224.69  
## <none> 218.69 224.69  
## + ever\_email\_month 1 216.78 224.78  
## + new\_per\_ticket\_service 1 217.44 225.44  
## + ever\_scheduling 1 217.48 225.48  
## + monthly\_rate\_group 11 199.65 227.65  
## + age\_group 4 215.83 229.83  
## + membership 6 214.79 232.79  
## + employment\_sector 15 204.53 240.53  
## - new\_per\_ticket\_scheduling 1 379.45 383.45  
## - num\_emails\_month 1 418.28 422.28

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=180.88  
## retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + new\_per\_ticket\_cx 1 163.61 173.61  
## + ever\_email\_month 1 164.02 174.02  
## <none> 172.88 180.88  
## + new\_per\_ticket\_service 1 171.66 181.66  
## + attendance\_grouping\_ver.1 5 164.06 182.06  
## + ever\_service 1 172.73 182.73  
## + ever\_scheduling 1 172.84 182.84  
## + membership 6 167.51 187.51  
## + monthly\_rate\_group 11 158.18 188.18  
## + age\_group 4 172.31 188.31  
## + employment\_sector 15 158.58 196.58  
## - ever\_cx 1 218.69 224.69  
## - new\_per\_ticket\_scheduling 1 289.74 295.74  
## - num\_emails\_month 1 363.44 369.44

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=173.61  
## retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + ever\_email\_month 1 156.47 168.47  
## <none> 163.61 173.61  
## + ever\_service 1 163.15 175.15  
## + new\_per\_ticket\_service 1 163.48 175.48  
## + ever\_scheduling 1 163.61 175.61  
## + attendance\_grouping\_ver.1 5 155.90 175.90  
## + monthly\_rate\_group 11 148.01 180.01  
## + age\_group 4 162.63 180.63  
## + membership 6 158.77 180.77  
## - new\_per\_ticket\_cx 1 172.88 180.88  
## + employment\_sector 15 150.03 190.03  
## - ever\_cx 1 206.21 214.21  
## - new\_per\_ticket\_scheduling 1 216.57 224.57  
## - num\_emails\_month 1 343.90 351.90

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=168.47  
## retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## + ever\_service 1 153.50 167.50  
## + new\_per\_ticket\_service 1 154.25 168.25  
## + attendance\_grouping\_ver.1 5 146.29 168.29  
## <none> 156.47 168.47  
## + ever\_scheduling 1 156.09 170.09  
## - ever\_email\_month 1 163.61 173.61  
## - new\_per\_ticket\_cx 1 164.01 174.01  
## + age\_group 4 155.51 175.51  
## + monthly\_rate\_group 11 142.14 176.14  
## + membership 6 152.95 176.95  
## + employment\_sector 15 144.89 186.89  
## - ever\_cx 1 202.26 212.26  
## - new\_per\_ticket\_scheduling 1 214.01 224.01  
## - num\_emails\_month 1 277.17 287.17

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=167.5  
## retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month + ever\_service

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## <none> 153.50 167.50  
## + attendance\_grouping\_ver.1 5 144.20 168.20  
## - ever\_service 1 156.47 168.47  
## + ever\_scheduling 1 153.31 169.31  
## + new\_per\_ticket\_service 1 153.43 169.43  
## - new\_per\_ticket\_cx 1 161.02 173.02  
## + age\_group 4 152.31 174.31  
## - ever\_email\_month 1 163.15 175.15  
## + membership 6 149.96 175.96  
## + monthly\_rate\_group 11 140.00 176.00  
## + employment\_sector 15 141.54 185.54  
## - ever\_cx 1 198.01 210.01  
## - new\_per\_ticket\_scheduling 1 209.61 221.61  
## - num\_emails\_month 1 276.13 288.13

##   
## Call: glm(formula = retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + new\_per\_ticket\_cx + ever\_email\_month + ever\_service,   
## family = binomial(link = "logit"), data = clean\_bang\_select.6m\_train)  
##   
## Coefficients:  
## (Intercept) num\_emails\_month   
## -0.56632 -1.81654   
## new\_per\_ticket\_scheduling ever\_cxyes   
## 0.06898 4.32293   
## new\_per\_ticket\_cx ever\_email\_monthyes   
## -0.05707 -2.12075   
## ever\_serviceyes   
## 1.43947   
##   
## Degrees of Freedom: 358 Total (i.e. Null); 352 Residual  
## Null Deviance: 496.2   
## Residual Deviance: 153.5 AIC: 167.5

model.retained.train\_6m = glm(retention\_6m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 ever\_email\_month +   
 new\_per\_ticket\_service +  
 new\_per\_ticket\_cx,   
 family = binomial(link = "logit"), data = clean\_bang\_select.6m\_train)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# Step 3: Assessing the proposed model  
  
summary(model.retained.train\_6m)

##   
## Call:  
## glm(formula = retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_service + new\_per\_ticket\_cx,   
## family = binomial(link = "logit"), data = clean\_bang\_select.6m\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2986 -0.2409 -0.0018 0.2226 4.2011   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.51104 0.73032 -0.700 0.48408   
## num\_emails\_month -1.81042 0.29301 -6.179 6.46e-10 \*\*\*  
## new\_per\_ticket\_scheduling 0.08122 0.01424 5.703 1.17e-08 \*\*\*  
## ever\_cxyes 4.39892 0.74503 5.904 3.54e-09 \*\*\*  
## ever\_email\_monthyes -2.08283 0.72833 -2.860 0.00424 \*\*   
## new\_per\_ticket\_service 0.01385 0.00947 1.462 0.14364   
## new\_per\_ticket\_cx -0.04698 0.02184 -2.151 0.03151 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 496.21 on 358 degrees of freedom  
## Residual deviance: 154.25 on 352 degrees of freedom  
## AIC: 168.25  
##   
## Number of Fisher Scoring iterations: 8

AIC(model.retained.train\_6m)# 191.86

## [1] 168.2514

exp(cbind(OR = coef(model.retained.train\_6m), confint.default(model.retained.train\_6m)))

## OR 2.5 % 97.5 %  
## (Intercept) 0.5998701 0.14335428 2.5101735  
## num\_emails\_month 0.1635850 0.09211617 0.2905032  
## new\_per\_ticket\_scheduling 1.0846086 1.05475519 1.1153071  
## ever\_cxyes 81.3631710 18.89093742 350.4307616  
## ever\_email\_monthyes 0.1245772 0.02988686 0.5192745  
## new\_per\_ticket\_service 1.0139450 0.99529864 1.0329408  
## new\_per\_ticket\_cx 0.9541074 0.91412046 0.9958436

varImp(model.retained.train\_6m, sort = T) # Top predictors are: num\_emails\_month, new\_per\_ticket\_scheduling, ever\_cx, ever\_email\_month and new\_per\_ticket\_service

## Overall  
## num\_emails\_month 6.178782  
## new\_per\_ticket\_scheduling 5.703464  
## ever\_cxyes 5.904322  
## ever\_email\_monthyes 2.859721  
## new\_per\_ticket\_service 1.462353  
## new\_per\_ticket\_cx 2.150635

# Step 4a: validating the proposed model   
  
pred\_6m\_log <- predict(model.retained.train\_6m, newdata = clean\_bang\_select.6m\_test)  
pred\_6m\_log = ifelse(pred\_6m\_log > 0.5, 'yes', 'no')  
table(pred\_6m\_log, clean\_bang\_select.6m\_test$retention\_6m)

##   
## pred\_6m\_log no yes  
## no 43 10  
## yes 4 31

accuracy = table(pred\_6m\_log, clean\_bang\_select.6m\_test[, "retention\_6m"])  
accuracy

##   
## pred\_6m\_log no yes  
## no 43 10  
## yes 4 31

sum(diag(accuracy))/sum(accuracy)

## [1] 0.8409091

mean(pred\_6m\_log == clean\_bang\_select.6m\_test$retention\_6m) # 85.22% or err.rate of 14.78%

## [1] 0.8409091

# Step 4b: repeated k-fold validation   
  
repeat\_ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)  
proposed.model.retained.6m = train(retention\_6m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 ever\_email\_month +   
 new\_per\_ticket\_service +  
 new\_per\_ticket\_cx,   
 data = clean\_bang\_select,   
 method = 'glm',  
 family = 'binomial',  
 trControl = repeat\_ctrl, tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

proposed.model.retained.6m # accuracy = 90.44%

## Generalized Linear Model   
##   
## 447 samples  
## 6 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 403, 402, 403, 402, 402, 402, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9046449 0.8084935

# Step 4c: k-fold validation   
  
ctrl = trainControl(method = 'cv', number = 10)  
proposed.model.retained.6m = train(retention\_6m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 ever\_email\_month +   
 new\_per\_ticket\_service +  
 new\_per\_ticket\_cx,   
 data = clean\_bang\_select,   
 method = 'glm',  
 family = 'binomial',  
 trControl = ctrl, tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

proposed.model.retained.6m # accuracy 89.91%

## Generalized Linear Model   
##   
## 447 samples  
## 6 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 402, 402, 403, 402, 402, 403, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9059091 0.8110452

# Step 5: sumamry of proposed model on original data set  
  
log.model.6m = glm(retention\_6m ~ num\_emails\_month +   
 new\_per\_ticket\_scheduling +   
 ever\_cx +   
 ever\_email\_month +   
 new\_per\_ticket\_service +  
 new\_per\_ticket\_cx,   
 family = binomial(link = "logit"),   
 data = clean\_bang\_select)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(log.model.6m)

##   
## Call:  
## glm(formula = retention\_6m ~ num\_emails\_month + new\_per\_ticket\_scheduling +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_service + new\_per\_ticket\_cx,   
## family = binomial(link = "logit"), data = clean\_bang\_select)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9774 -0.2802 -0.0124 0.2499 3.5261   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.693312 0.707126 -0.980 0.3269   
## num\_emails\_month -1.392663 0.209425 -6.650 2.93e-11 \*\*\*  
## new\_per\_ticket\_scheduling 0.084125 0.012671 6.639 3.15e-11 \*\*\*  
## ever\_cxyes 4.206692 0.614087 6.850 7.37e-12 \*\*\*  
## ever\_email\_monthyes -3.044388 0.648682 -4.693 2.69e-06 \*\*\*  
## new\_per\_ticket\_service 0.021232 0.008894 2.387 0.0170 \*   
## new\_per\_ticket\_cx -0.044074 0.019326 -2.281 0.0226 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 617.79 on 446 degrees of freedom  
## Residual deviance: 214.87 on 440 degrees of freedom  
## AIC: 228.87  
##   
## Number of Fisher Scoring iterations: 7

AIC(log.model.6m)

## [1] 228.8748

exp(coef(log.model.6m))

## (Intercept) num\_emails\_month new\_per\_ticket\_scheduling   
## 0.49991747 0.24841293 1.08776434   
## ever\_cxyes ever\_email\_monthyes new\_per\_ticket\_service   
## 67.13410796 0.04762543 1.02145950   
## new\_per\_ticket\_cx   
## 0.95688338

vif(log.model.6m)

## num\_emails\_month new\_per\_ticket\_scheduling ever\_cxyes   
## 1.781654 3.387625 3.042386   
## ever\_email\_monthyes new\_per\_ticket\_service new\_per\_ticket\_cx   
## 1.999301 1.932293 3.356861

exp(cbind(OR = coef(log.model.6m), confint.default(log.model.6m)))

## OR 2.5 % 97.5 %  
## (Intercept) 0.49991747 0.1250234 1.9989653  
## num\_emails\_month 0.24841293 0.1647825 0.3744875  
## new\_per\_ticket\_scheduling 1.08776434 1.0610830 1.1151166  
## ever\_cxyes 67.13410796 20.1479688 223.6944326  
## ever\_email\_monthyes 0.04762543 0.0133561 0.1698236  
## new\_per\_ticket\_service 1.02145950 1.0038068 1.0394226  
## new\_per\_ticket\_cx 0.95688338 0.9213156 0.9938243

### RETENTION ANALYSIS: 12M Membership Retention Status

In generating a logistic regression model for membership status at 12-month, it was found that the variables that were retained through bi-directional stepwise regression through partitioned data (80-:20) were

\* status of ever haivng a CX-related email interaction  
 \* percent composition of scheduling-related email interactions  
 \* percent composition of service-related email interactions  
 \* Status of ever having a non-billing email interaction  
 \* monthly membership rates  
 \* number of non-billing related email interaction per month.

In cross-validating the proposed model through the validation set approach as well as repeated K-fold validation that the accuracy of the model ranged b/t **86.76% - 90.00%**. The major predictors turned out to be ever\_cx, new\_per\_ticket\_scheduling, num\_emails\_month and ever\_email\_month.

# Step 1: Partition data  
  
trainIndex\_12m = createDataPartition(clean\_bang\_select$retention\_12m, p = 0.8, list = F)  
  
clean\_bang\_select.12m\_train = clean\_bang\_select[trainIndex\_12m,] # This is the Training Data (80% of the data)  
clean\_bang\_select.12m\_test = clean\_bang\_select[-trainIndex\_12m,] # This is the Testing Data (20% of the data)  
  
  
# Step 2: Bi-directional Stepwise regression   
  
model.start.train\_12m = glm(retention\_12m ~ 1, data = clean\_bang\_select.12m\_train, family = binomial(link = 'logit'))  
model.all.train\_12m = glm(retention\_12m ~ age\_group +   
 employment\_sector +   
 membership +   
 attendance\_grouping\_ver.1 +   
 monthly\_rate\_group +   
 ever\_email\_month +   
 num\_emails\_month +   
 ever\_cx+  
 new\_per\_ticket\_cx +   
 ever\_service+  
 new\_per\_ticket\_service +  
 ever\_scheduling +  
 new\_per\_ticket\_scheduling,   
 data = clean\_bang\_select.12m\_train, family = binomial(link = 'logit'))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

step(model.start.train\_12m, direction = 'both', scope = formula(model.all.train\_12m))

## Start: AIC=460.02  
## retention\_12m ~ 1  
##   
## Df Deviance AIC  
## + new\_per\_ticket\_scheduling 1 373.27 377.27  
## + ever\_cx 1 383.59 387.59  
## + new\_per\_ticket\_service 1 387.79 391.79  
## + ever\_scheduling 1 400.75 404.75  
## + num\_emails\_month 1 402.90 406.90  
## + monthly\_rate\_group 11 407.35 431.35  
## + ever\_email\_month 1 443.14 447.14  
## + membership 6 437.40 451.40  
## + attendance\_grouping\_ver.1 5 447.54 459.54  
## <none> 458.02 460.02  
## + ever\_service 1 456.65 460.65  
## + new\_per\_ticket\_cx 1 457.90 461.90  
## + age\_group 4 454.29 464.29  
## + employment\_sector 15 445.90 477.90  
##   
## Step: AIC=377.27  
## retention\_12m ~ new\_per\_ticket\_scheduling  
##   
## Df Deviance AIC  
## + num\_emails\_month 1 275.66 281.66  
## + ever\_email\_month 1 318.22 324.22  
## + ever\_cx 1 318.93 324.93  
## + monthly\_rate\_group 11 331.86 357.86  
## + new\_per\_ticket\_cx 1 365.46 371.46  
## + membership 6 356.07 372.07  
## + new\_per\_ticket\_service 1 366.58 372.58  
## + attendance\_grouping\_ver.1 5 359.71 373.71  
## <none> 373.27 377.27  
## + ever\_scheduling 1 372.15 378.15  
## + ever\_service 1 372.61 378.61  
## + age\_group 4 372.77 384.77  
## + employment\_sector 15 363.43 397.43  
## - new\_per\_ticket\_scheduling 1 458.02 460.02  
##   
## Step: AIC=281.66  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month  
##   
## Df Deviance AIC  
## + ever\_cx 1 229.70 237.70  
## + ever\_email\_month 1 258.78 266.78  
## + new\_per\_ticket\_cx 1 266.82 274.82  
## + ever\_service 1 271.41 279.41  
## + membership 6 262.37 280.37  
## + monthly\_rate\_group 11 253.20 281.20  
## + new\_per\_ticket\_service 1 273.55 281.55  
## <none> 275.66 281.66  
## + ever\_scheduling 1 275.14 283.14  
## + attendance\_grouping\_ver.1 5 268.68 284.68  
## + age\_group 4 274.99 288.99  
## + employment\_sector 15 270.43 306.43  
## - num\_emails\_month 1 373.27 377.27  
## - new\_per\_ticket\_scheduling 1 402.90 406.90  
##   
## Step: AIC=237.7  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx  
##   
## Df Deviance AIC  
## + ever\_email\_month 1 192.72 202.72  
## + new\_per\_ticket\_cx 1 221.21 231.21  
## + new\_per\_ticket\_service 1 226.76 236.76  
## <none> 229.70 237.70  
## + ever\_service 1 228.57 238.57  
## + ever\_scheduling 1 229.69 239.69  
## + membership 6 220.98 240.98  
## + monthly\_rate\_group 11 211.98 241.98  
## + age\_group 4 226.48 242.48  
## + attendance\_grouping\_ver.1 5 224.57 242.57  
## + employment\_sector 15 225.15 263.15  
## - ever\_cx 1 275.66 281.66  
## - num\_emails\_month 1 318.93 324.93  
## - new\_per\_ticket\_scheduling 1 319.70 325.70  
##   
## Step: AIC=202.72  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month  
##   
## Df Deviance AIC  
## + new\_per\_ticket\_cx 1 184.21 196.21  
## + new\_per\_ticket\_service 1 187.06 199.06  
## + ever\_service 1 189.56 201.56  
## + monthly\_rate\_group 11 170.59 202.59  
## <none> 192.72 202.72  
## + ever\_scheduling 1 192.35 204.35  
## + age\_group 4 187.47 205.47  
## + membership 6 183.75 205.75  
## + attendance\_grouping\_ver.1 5 186.41 206.41  
## + employment\_sector 15 186.21 226.21  
## - ever\_email\_month 1 229.70 237.70  
## - num\_emails\_month 1 238.50 246.50  
## - ever\_cx 1 258.78 266.78  
## - new\_per\_ticket\_scheduling 1 310.05 318.05  
##   
## Step: AIC=196.21  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx  
##   
## Df Deviance AIC  
## + ever\_service 1 181.07 195.07  
## + monthly\_rate\_group 11 161.67 195.67  
## <none> 184.21 196.21  
## + membership 6 172.51 196.51  
## + new\_per\_ticket\_service 1 182.74 196.74  
## + age\_group 4 177.68 197.68  
## + ever\_scheduling 1 183.98 197.98  
## + attendance\_grouping\_ver.1 5 179.14 201.14  
## - new\_per\_ticket\_cx 1 192.72 202.72  
## + employment\_sector 15 176.32 218.32  
## - ever\_email\_month 1 221.21 231.21  
## - num\_emails\_month 1 227.96 237.96  
## - new\_per\_ticket\_scheduling 1 230.60 240.60  
## - ever\_cx 1 245.97 255.97  
##   
## Step: AIC=195.07  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + ever\_service  
##   
## Df Deviance AIC  
## + age\_group 4 170.78 192.78  
## + monthly\_rate\_group 11 158.91 194.91  
## <none> 181.07 195.07  
## - ever\_service 1 184.21 196.21  
## + new\_per\_ticket\_service 1 180.54 196.54  
## + membership 6 170.63 196.63  
## + ever\_scheduling 1 181.07 197.07  
## + attendance\_grouping\_ver.1 5 176.42 200.42  
## - new\_per\_ticket\_cx 1 189.56 201.56  
## + employment\_sector 15 172.43 216.43  
## - ever\_email\_month 1 220.54 232.54  
## - num\_emails\_month 1 225.89 237.89  
## - new\_per\_ticket\_scheduling 1 225.91 237.91  
## - ever\_cx 1 238.82 250.82  
##   
## Step: AIC=192.78  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + ever\_service +   
## age\_group  
##   
## Df Deviance AIC  
## + monthly\_rate\_group 11 147.33 191.33  
## <none> 170.78 192.78  
## + new\_per\_ticket\_service 1 170.03 194.03  
## + membership 6 160.54 194.54  
## + ever\_scheduling 1 170.77 194.77  
## - age\_group 4 181.07 195.07  
## + attendance\_grouping\_ver.1 5 165.57 197.57  
## - ever\_service 1 177.68 197.68  
## - new\_per\_ticket\_cx 1 181.05 201.05  
## + employment\_sector 15 158.48 210.48  
## - new\_per\_ticket\_scheduling 1 213.57 233.57  
## - num\_emails\_month 1 215.58 235.58  
## - ever\_email\_month 1 216.63 236.63  
## - ever\_cx 1 235.68 255.68  
##   
## Step: AIC=191.33  
## retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + ever\_service +   
## age\_group + monthly\_rate\_group

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## <none> 147.33 191.33  
## + new\_per\_ticket\_service 1 146.11 192.11  
## - monthly\_rate\_group 11 170.78 192.78  
## + ever\_scheduling 1 147.25 193.25  
## + membership 6 138.00 194.00  
## - ever\_service 1 152.28 194.28  
## - age\_group 4 158.91 194.91  
## + attendance\_grouping\_ver.1 5 143.91 197.91  
## - new\_per\_ticket\_cx 1 157.70 199.70  
## + employment\_sector 15 134.21 208.21  
## - num\_emails\_month 1 192.78 234.78  
## - new\_per\_ticket\_scheduling 1 193.30 235.30  
## - ever\_email\_month 1 199.38 241.38  
## - ever\_cx 1 215.46 257.46

##   
## Call: glm(formula = retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + new\_per\_ticket\_cx + ever\_service +   
## age\_group + monthly\_rate\_group, family = binomial(link = "logit"),   
## data = clean\_bang\_select.12m\_train)  
##   
## Coefficients:  
## (Intercept) new\_per\_ticket\_scheduling   
## -0.75350 0.07200   
## num\_emails\_month ever\_cxyes   
## -1.00049 6.09768   
## ever\_email\_monthyes new\_per\_ticket\_cx   
## -5.23329 -0.07431   
## ever\_serviceyes age\_group18-29   
## 2.74153 -5.43270   
## age\_group30-44 age\_group45-64   
## -4.55369 -4.90912   
## age\_group65+ monthly\_rate\_group100-149.99   
## -2.32706 -13.41350   
## monthly\_rate\_group150-199.99 monthly\_rate\_group200-249.99   
## 4.76924 3.51436   
## monthly\_rate\_group250-299.99 monthly\_rate\_group300-349.99   
## 2.26112 2.20111   
## monthly\_rate\_group350-399.99 monthly\_rate\_group400-449.99   
## 2.12920 1.33212   
## monthly\_rate\_group450-499.99 monthly\_rate\_group500-549.99   
## -0.09764 2.38252   
## monthly\_rate\_group550-599.99 monthly\_rate\_group600+   
## -17.13445 -0.77228   
##   
## Degrees of Freedom: 357 Total (i.e. Null); 336 Residual  
## Null Deviance: 458   
## Residual Deviance: 147.3 AIC: 191.3

model.retained.train\_12m = glm(retention\_12m ~ new\_per\_ticket\_scheduling +   
 num\_emails\_month +   
 ever\_cx +   
 ever\_email\_month +   
 monthly\_rate\_group +   
 new\_per\_ticket\_service,   
 family = binomial(link = "logit"),   
 data = clean\_bang\_select.12m\_train)  
  
  
# Step 3: Assessing the proposed model  
  
summary(model.retained.train\_12m)

##   
## Call:  
## glm(formula = retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + monthly\_rate\_group + new\_per\_ticket\_service,   
## family = binomial(link = "logit"), data = clean\_bang\_select.12m\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.1077 -0.2056 -0.0457 0.3382 3.2547   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.29163 2.20899 -2.396 0.0166 \*   
## new\_per\_ticket\_scheduling 0.10369 0.01651 6.281 3.37e-10 \*\*\*  
## num\_emails\_month -0.87199 0.18845 -4.627 3.71e-06 \*\*\*  
## ever\_cxyes 4.38159 0.70550 6.211 5.28e-10 \*\*\*  
## ever\_email\_monthyes -4.39315 0.79284 -5.541 3.01e-08 \*\*\*  
## monthly\_rate\_group100-149.99 -14.02040 1946.44247 -0.007 0.9943   
## monthly\_rate\_group150-199.99 3.25067 2.04037 1.593 0.1111   
## monthly\_rate\_group200-249.99 2.54751 1.76262 1.445 0.1484   
## monthly\_rate\_group250-299.99 1.39061 1.73531 0.801 0.4229   
## monthly\_rate\_group300-349.99 1.44718 1.65584 0.874 0.3821   
## monthly\_rate\_group350-399.99 1.29377 1.65376 0.782 0.4340   
## monthly\_rate\_group400-449.99 0.64073 1.65687 0.387 0.6990   
## monthly\_rate\_group450-499.99 -0.93255 1.99840 -0.467 0.6408   
## monthly\_rate\_group500-549.99 1.03832 2.18781 0.475 0.6351   
## monthly\_rate\_group550-599.99 -17.52737 1458.29768 -0.012 0.9904   
## monthly\_rate\_group600+ -1.27247 6.28458 -0.202 0.8395   
## new\_per\_ticket\_service 0.03175 0.01694 1.874 0.0609 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 458.02 on 357 degrees of freedom  
## Residual deviance: 165.74 on 341 degrees of freedom  
## AIC: 199.74  
##   
## Number of Fisher Scoring iterations: 17

AIC(model.retained.train\_12m) # 169.274

## [1] 199.7404

exp(cbind(OR = coef(model.retained.train\_12m), confint.default(model.retained.train\_12m)))

## OR 2.5 % 97.5 %  
## (Intercept) 5.033532e-03 6.631021e-05 3.820897e-01  
## new\_per\_ticket\_scheduling 1.109253e+00 1.073937e+00 1.145731e+00  
## num\_emails\_month 4.181200e-01 2.889954e-01 6.049381e-01  
## ever\_cxyes 7.996525e+01 2.006204e+01 3.187334e+02  
## ever\_email\_monthyes 1.236176e-02 2.613457e-03 5.847167e-02  
## monthly\_rate\_group100-149.99 8.147388e-07 0.000000e+00 Inf  
## monthly\_rate\_group150-199.99 2.580761e+01 4.731338e-01 1.407705e+03  
## monthly\_rate\_group200-249.99 1.277528e+01 4.036677e-01 4.043123e+02  
## monthly\_rate\_group250-299.99 4.017295e+00 1.339176e-01 1.205119e+02  
## monthly\_rate\_group300-349.99 4.251122e+00 1.655958e-01 1.091334e+02  
## monthly\_rate\_group350-399.99 3.646492e+00 1.426232e-01 9.323105e+01  
## monthly\_rate\_group400-449.99 1.897871e+00 7.377939e-02 4.882006e+01  
## monthly\_rate\_group450-499.99 3.935472e-01 7.833437e-03 1.977158e+01  
## monthly\_rate\_group500-549.99 2.824481e+00 3.878557e-02 2.056872e+02  
## monthly\_rate\_group550-599.99 2.443197e-08 0.000000e+00 Inf  
## monthly\_rate\_group600+ 2.801390e-01 1.252933e-06 6.263533e+04  
## new\_per\_ticket\_service 1.032262e+00 9.985456e-01 1.067116e+00

varImp(model.retained.train\_12m, sort = T) # Top predictors are:new\_per\_ticket\_scheduling, num\_emails\_month, ever\_email\_month and ever\_cx

## Overall  
## new\_per\_ticket\_scheduling 6.280943734  
## num\_emails\_month 4.627121100  
## ever\_cxyes 6.210582562  
## ever\_email\_monthyes 5.541038990  
## monthly\_rate\_group100-149.99 0.007203089  
## monthly\_rate\_group150-199.99 1.593178543  
## monthly\_rate\_group200-249.99 1.445296949  
## monthly\_rate\_group250-299.99 0.801361775  
## monthly\_rate\_group300-349.99 0.873986697  
## monthly\_rate\_group350-399.99 0.782316463  
## monthly\_rate\_group400-449.99 0.386712374  
## monthly\_rate\_group450-499.99 0.466649581  
## monthly\_rate\_group500-549.99 0.474595140  
## monthly\_rate\_group550-599.99 0.012019064  
## monthly\_rate\_group600+ 0.202474807  
## new\_per\_ticket\_service 1.874062773

# Step 4: validating the proposed model   
  
pred\_12m\_log <- predict(model.retained.train\_12m, newdata = clean\_bang\_select.12m\_test)  
pred\_12m\_log = ifelse(pred\_12m\_log > 0.5, 'yes', 'no')  
table(pred\_12m\_log, clean\_bang\_select.12m\_test$retention\_12m)

##   
## pred\_12m\_log no yes  
## no 58 4  
## yes 1 26

accuracy = table(pred\_12m\_log, clean\_bang\_select.12m\_test[, "retention\_12m"])  
accuracy

##   
## pred\_12m\_log no yes  
## no 58 4  
## yes 1 26

sum(diag(accuracy))/sum(accuracy)

## [1] 0.9438202

mean(pred\_12m\_log == clean\_bang\_select.12m\_test$retention\_12m) # 88.76% or err.rate of 11.24%

## [1] 0.9438202

#Step 4b: repeated k-fold validation   
  
repeat\_ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)  
proposed.model.retained.12m = train(retention\_12m ~ new\_per\_ticket\_scheduling +   
 num\_emails\_month +   
 ever\_cx +   
 ever\_email\_month +   
 monthly\_rate\_group +   
 new\_per\_ticket\_service,   
 data = clean\_bang\_select,   
 method = 'glm',  
 family = 'binomial',  
 trControl = repeat\_ctrl, tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

proposed.model.retained.12m # accuracy = 90.00%

## Generalized Linear Model   
##   
## 447 samples  
## 6 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 403, 403, 402, 402, 402, 402, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9045286 0.7862865

# Step 4c: k-fold validation   
  
ctrl = trainControl(method = 'cv', number = 10)  
proposed.model.retained.12m = train(retention\_12m ~ new\_per\_ticket\_scheduling +   
 num\_emails\_month +   
 ever\_cx +   
 ever\_email\_month +   
 monthly\_rate\_group +   
 new\_per\_ticket\_service,   
 data = clean\_bang\_select,   
 method = 'glm',  
 family = 'binomial',  
 trControl = ctrl, tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

proposed.model.retained.12m # accuracy 89.96%

## Generalized Linear Model   
##   
## 447 samples  
## 6 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 403, 402, 402, 402, 402, 403, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9037879 0.7839525

# Step 5: sumamry of proposed model on original data set  
  
log.model.12m = glm(retention\_12m ~ new\_per\_ticket\_scheduling +   
 num\_emails\_month +   
 ever\_cx +   
 ever\_email\_month +   
 monthly\_rate\_group +   
 new\_per\_ticket\_service,   
 family = binomial(link = "logit"),   
 data = clean\_bang\_select)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(log.model.12m)

##   
## Call:  
## glm(formula = retention\_12m ~ new\_per\_ticket\_scheduling + num\_emails\_month +   
## ever\_cx + ever\_email\_month + monthly\_rate\_group + new\_per\_ticket\_service,   
## family = binomial(link = "logit"), data = clean\_bang\_select)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0834 -0.2069 -0.0405 0.3074 3.2094   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.14442 2.13520 -2.409 0.0160 \*   
## new\_per\_ticket\_scheduling 0.10195 0.01464 6.963 3.34e-12 \*\*\*  
## num\_emails\_month -1.03940 0.20950 -4.961 7.00e-07 \*\*\*  
## ever\_cxyes 4.61203 0.65038 7.091 1.33e-12 \*\*\*  
## ever\_email\_monthyes -4.23238 0.74469 -5.683 1.32e-08 \*\*\*  
## monthly\_rate\_group100-149.99 1.04721 2.31017 0.453 0.6503   
## monthly\_rate\_group150-199.99 3.14763 1.98533 1.585 0.1129   
## monthly\_rate\_group200-249.99 2.45729 1.76467 1.392 0.1638   
## monthly\_rate\_group250-299.99 1.42599 1.77426 0.804 0.4216   
## monthly\_rate\_group300-349.99 1.50211 1.69796 0.885 0.3763   
## monthly\_rate\_group350-399.99 1.44543 1.68461 0.858 0.3909   
## monthly\_rate\_group400-449.99 0.67973 1.69283 0.402 0.6880   
## monthly\_rate\_group450-499.99 -1.04489 2.02148 -0.517 0.6052   
## monthly\_rate\_group500-549.99 0.66005 2.11065 0.313 0.7545   
## monthly\_rate\_group550-599.99 -17.37800 1263.34754 -0.014 0.9890   
## monthly\_rate\_group600+ -2.17309 4.52303 -0.480 0.6309   
## new\_per\_ticket\_service 0.02884 0.01515 1.904 0.0569 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 571.78 on 446 degrees of freedom  
## Residual deviance: 194.40 on 430 degrees of freedom  
## AIC: 228.4  
##   
## Number of Fisher Scoring iterations: 17

AIC(log.model.12m)

## [1] 228.3996

exp(coef(log.model.12m))

## (Intercept) new\_per\_ticket\_scheduling   
## 5.831828e-03 1.107330e+00   
## num\_emails\_month ever\_cxyes   
## 3.536678e-01 1.006879e+02   
## ever\_email\_monthyes monthly\_rate\_group100-149.99   
## 1.451780e-02 2.849694e+00   
## monthly\_rate\_group150-199.99 monthly\_rate\_group200-249.99   
## 2.328082e+01 1.167309e+01   
## monthly\_rate\_group250-299.99 monthly\_rate\_group300-349.99   
## 4.161993e+00 4.491163e+00   
## monthly\_rate\_group350-399.99 monthly\_rate\_group400-449.99   
## 4.243667e+00 1.973345e+00   
## monthly\_rate\_group450-499.99 monthly\_rate\_group500-549.99   
## 3.517288e-01 1.934892e+00   
## monthly\_rate\_group550-599.99 monthly\_rate\_group600+   
## 2.836799e-08 1.138254e-01   
## new\_per\_ticket\_service   
## 1.029261e+00

vif(log.model.12m)

## new\_per\_ticket\_scheduling num\_emails\_month   
## 4.073049 1.576013   
## ever\_cxyes ever\_email\_monthyes   
## 2.457218 2.308606   
## monthly\_rate\_group100-149.99 monthly\_rate\_group150-199.99   
## 2.061469 3.774348   
## monthly\_rate\_group200-249.99 monthly\_rate\_group250-299.99   
## 8.014555 6.860786   
## monthly\_rate\_group300-349.99 monthly\_rate\_group350-399.99   
## 13.992423 16.815340   
## monthly\_rate\_group400-449.99 monthly\_rate\_group450-499.99   
## 14.532071 2.944180   
## monthly\_rate\_group500-549.99 monthly\_rate\_group550-599.99   
## 2.605817 1.000002   
## monthly\_rate\_group600+ new\_per\_ticket\_service   
## 1.164648 3.903893

exp(cbind(OR = coef(log.model.12m), confint.default(log.model.12m)))

## OR 2.5 % 97.5 %  
## (Intercept) 5.831828e-03 8.878077e-05 3.830809e-01  
## new\_per\_ticket\_scheduling 1.107330e+00 1.076003e+00 1.139569e+00  
## num\_emails\_month 3.536678e-01 2.345684e-01 5.332386e-01  
## ever\_cxyes 1.006879e+02 2.814326e+01 3.602304e+02  
## ever\_email\_monthyes 1.451780e-02 3.372995e-03 6.248642e-02  
## monthly\_rate\_group100-149.99 2.849694e+00 3.078777e-02 2.637657e+02  
## monthly\_rate\_group150-199.99 2.328082e+01 4.754237e-01 1.140028e+03  
## monthly\_rate\_group200-249.99 1.167309e+01 3.673659e-01 3.709139e+02  
## monthly\_rate\_group250-299.99 4.161993e+00 1.285427e-01 1.347582e+02  
## monthly\_rate\_group300-349.99 4.491163e+00 1.610834e-01 1.252181e+02  
## monthly\_rate\_group350-399.99 4.243667e+00 1.562433e-01 1.152607e+02  
## monthly\_rate\_group400-449.99 1.973345e+00 7.149334e-02 5.446790e+01  
## monthly\_rate\_group450-499.99 3.517288e-01 6.691511e-03 1.848808e+01  
## monthly\_rate\_group500-549.99 1.934892e+00 3.090777e-02 1.211284e+02  
## monthly\_rate\_group550-599.99 2.836799e-08 0.000000e+00 Inf  
## monthly\_rate\_group600+ 1.138254e-01 1.607776e-05 8.058478e+02  
## new\_per\_ticket\_service 1.029261e+00 9.991531e-01 1.060277e+00

# **DISCUSSION**

During this analysis, several notable results were shown that can be taken in our approach to ensuring membership retention.

1. The most commonly noted reason for membership churn being related to finance + lack of accessibility/availability. There were several findings that seemed to support this such as: \* The majority of our clientele being in scheduling demanding fields: technology, advertising/media and finance. \* Majority having an attendance rate of less than 70% \* Approx. 1/3 of total email interaction with staff pertaining to scheduling or rescheduling requests.
2. While the median membership length was approximately 4.5 months, this differed across membership type and demographics

* \* Technology sector having one of the longest membership duration as compared to others VS. government/social services + retail/accommodation/hospitality having the lowest  
   \* 30-44 demographic had the longest membership length out of all age groups   
   \* 2x/week membership > 3x/week membership in terms of length. Interestingly, those that have unlimited number of Hybrid sessions also tend to have a longer membership retention  
   \* Monthly rates b/t 300-399 appear to be "sweet spot" in terms of length of membership with noticeable degradation in membership length with rates beyond $400

1. The impact of the COVID pandemic has significantly reduced the clientele by 31% from our initial shutdown. This was cited as the 3rd most common reason for membership loss over the last two years.
2. The importance of customer interactions was also noted.

Namely, it was found that an increase in the number of non-billing email interaction per month was associated with an increased odds of membership churn. However in terms of type of email interaction, having even a single CX-related email interaction outside of the onboarding process was found to significantly increased the odds of membership retention at 3-, 6- and 12-months. Although significant, the impact of an increasing the number of CX-related email interaction had a small decrease in the odds of membership churn.

In terms of improving retention status, efforts should be made with respect to minimizing the drop in attendance seeing as we are not going to be changing our business model anytime soon. Habit-focus development appears to be a key area to dive into to address this issue. Namely the adoption of a “fallback” option could be introduced and highlighted early on during the on-boarding transition phase highlighting strategies to prepare and implement for situations of inevitable scheduling disruptions. From the membership-service end, a work flow could be developed to track and assess attendance across several time points over a 90 day span to see what really is the best membership is for a new member. While the default option has been the push to 3x/week, this option has been shown to fail to sustain membership retention over 3-months as compared to the 2x/week option. This is possibly due to the sense of inadequate return in value considering the noticeable price point, along with some potential difficulty to attend enough sessions to justify said price point.

Possible solutions can be (1) revision into pricing –> considering that most current members were under older or modified pricing as compared to the updated membership rates (2) reframing return of value based on attendance or provide lower cost additions to justify membership price point (i.e. increased group class schedule, nutritional-habit coaching, etc.) (3) early membership service team intervention to handle finding out the right membership type based on previous attendance rates –> similar to those wine/snackbox subscription where we will find the best solution for the member for their value = improve CX –> this can be stratified based on certain demographics (4) Improving on schedule availability could be an option