Lab 02

w271

Alex Yang, John Kenney, Ram Balasubramanian

Oct 12, 2017

SECTION - 1 INTRODUCTION & KEY RESULTS

Problem Introduction:

We have been hired by a Private University to identify who among their Alumni are most likely to contribute towards the University's foundation in future years. The university has provided us with data on past contributions from graduates - data includes some demogracramerc information (like gender, marital status), university specific information (graduation year, major of studies), and some information on how "connected"" an Alumnus is to the school (Alumni event attendance, historical contributions).

1.1 HIGH LEVEL DESCRIPTION OF MODELING APPROACH:

We have taken two approaches to the problem (named Beta-Hat and Y-Hat):
Approach "Beta-Hat": We will treat the problem as a "explanation" problem ($\hat{\beta}$). The goal here is to figure out if and how much certain aspects of a person and their association with the university determines how much they will contribute to the university's foundation. We will develop a regression model that considers the 2016 contributions as a variable that depends on one or more of the other data elements that have been provided. The regression coefficients can then be interpreted as a measure of how much each aspect of a person influences their contributions.

Approach "Y-Hat":
We will treat the problem as a "prediction" problem ($\hat{y}$ problem). Given all the data we have about a person and their past contributions, can we predict how much they will contribute in the future. We will develop a model that aims to predict the 2016 contribution amounts for each person. To evaluate the efficacy of our models, we will split the data into a "training" set and a "test" set. We will use the training data to estimate parameters for our prediction model and evaluate our model's prediction accuracy using the test set.

1.2 KEY RESULTS AND TECHNIQUES USED:

We will complete this section once we are done with the modeling work.

SECTION 2 - DATA EXAMINATION AND EDA:

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)

#Libraries required
library(car)
```

```r
library(dplyr)
library(Hmisc)#Used by author for 3D plotting
library(ggplot2)
library(gridExtra)
library(effsize) #Used to calculate Cohen's D for T-Test
library(aod)    #Used for effect size of the logit model
library(mcprofile) #Used for confidence intervals
library(package = MASS)  # Location of parcoord() function
library(vcd)
library(data.table)
library(stargazer)
library(caret) #Required for Confusion Matrix
library(ordinal)
```

```r
dt <- fread("lab2data.csv")
describe(dt)
```

```
## dt
##
##  12  Variables      1000  Observations
## ---------------------------------------------------------------------------
## V1
##        n  missing distinct
##     1000        0     1000
##
## lowest : 1    10   100  1002 1003, highest: 995  996  997  998  999
## ---------------------------------------------------------------------------
## Gender
##        n  missing distinct
##     1000        0        2
##
## Value          F     M
## Frequency    505   495
## Proportion 0.505 0.495
## ---------------------------------------------------------------------------
## Class.Year
##        n  missing distinct     Info     Mean      Gmd
##     1000        0        5    0.949     1996    15.07
##
## Value       1972  1982  1992  2002  2012
## Frequency    105   176   203   223   293
## Proportion 0.105 0.176 0.203 0.223 0.293
## ---------------------------------------------------------------------------
## Marital.Status
##        n  missing distinct
##     1000        0        4
##
## Value          D     M     S     W
## Frequency     61   584   344    11
## Proportion 0.061 0.584 0.344 0.011
```

```
## ----------------------------------------------------------------------------
## Major
##        n  missing distinct
##     1000        0       45
##
## lowest : American Studies     Anthropology         Art                  Biology              Ch
## highest: Spanish              Speech (Drama, etc.) Speech Correction    Theatre              Zo
## ----------------------------------------------------------------------------
## Next.Degree
##        n  missing distinct
##     1000        0       47
##
## lowest : AA   BA   BAE  BD   BFA , highest: UBDS UDDS UMD  UMDS UNKD
## ----------------------------------------------------------------------------
## AttendenceEvent
##        n  missing distinct     Info      Sum     Mean      Gmd
##     1000        0        2    0.717      605    0.605   0.4784
##
## ----------------------------------------------------------------------------
## FY12Giving
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0       66    0.826    186.9    345.5        0        0
##      .25      .50      .75      .90      .95
##        0        0       60      200      350
##
## lowest :     0.00     5.00     6.50     7.00     8.00
## highest: 10000.00 12000.00 16959.99 20000.00 21000.00
## ----------------------------------------------------------------------------
## FY13Giving
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0       78    0.864    311.5    590.4      0.0      0.0
##      .25      .50      .75      .90      .95
##      0.0      0.0     75.0    210.5    400.0
##
## Value           0    500   1000   1500   2000   2500   3000   5000   5500
## Frequency     920     48     13      4      2      3      2      2      1
## Proportion  0.920  0.048  0.013  0.004  0.002  0.003  0.002  0.002  0.001
##
## Value        8000  12000  13000  14500 161500
## Frequency       1      1      1      1      1
## Proportion  0.001  0.001  0.001  0.001  0.001
## ----------------------------------------------------------------------------
## FY14Giving
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0       80     0.83    142.6    255.5        0        0
##      .25      .50      .75      .90      .95
##        0        0       50      200      450
##
## lowest :     0.00     1.00     5.00     8.00    10.00
## highest:  5000.00  6000.00  8031.00 10000.00 11187.26
```

```
## ---------------------------------------------------------------------------
## FY15Giving
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0       62    0.817    252.2    470.7      0.0      0.0
##      .25      .50      .75      .90      .95
##      0.0      0.0     75.0    200.0    538.3
##
## lowest :     0.0     5.0    10.0    13.0    15.0
## highest: 10000.0 14776.0 15634.5 26500.0 58785.5
## ---------------------------------------------------------------------------
## FY16Giving
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0       71    0.798      170    308.2        0        0
##      .25      .50      .75      .90      .95
##        0        0       75      216      500
##
## lowest :     0.00     5.00    10.00    15.00    18.00
## highest:  5000.00  6500.00 11500.00 11505.84 14655.25
## ---------------------------------------------------------------------------
```

2.1 Brief Description of Data Available:

We have data for 1000 past graduates of the University. There are 12 variables provided for each Alumnus. They are: 1. V1: Identifier for each record (Alumnus)
2. Gender: M/F, roughly 50/50 in the sample provided. 3. Class.Year: Appears like the "decade" of the graduating year. Goes from 1972 - 2012. We will assume 1972 represents students graduating from 1963-1972; 1982 represents students graduating from 1973 to 1982 etc.
4. Marital.Status: Has 4 categories - coded D,M,S,W. We will assume it means Divorced, Married, Single, Widowed with over 90% in the "married" or "single" categories.
5. Major: There are 45 majors represented in the sample. History, English, Biology & Economics are the top 4 representing about 37% of the sample.
6. Next.Degree: We assume this means what the alumnus went on to do after graduating from the university. 38% shows "None" implying they did not pursue another degree. The remainder (62%) seems rather high for this metric.
7. AttendenceEvent: Indicates whether the alumnus attended an alumni event between 2012 and 2015. If we choose to use this variable to model "Giving" we should probably not use it to model 2012-2014 Giving
8. FYGiving: There are 5 of these variables named FY12 - FY16 representing full year 2012 through full year 2016 contribution from the alumnus. There are some "outliers" (e.g. $161,500 in 2013) in the data that we may need to be on the lookout for.

We do not have any missing values in the data; and there do not seem to be an obvious "data cleaning" that needs to be conducted. We will conduct an Exploratory Data Analysis next.

What are the important variables we want to include in our discussion here? What would we suppose would be meaningful? What can we omit?

```
# View the contents of Major and Next Degree - to identify if
# there are any obvious groupings
majortable = as.data.frame(round(prop.table(table(dt$Major)),
    2))
degtable = as.data.frame(round(prop.table(table(dt$Next.Degree)),
    2))
# describe(dt$MajorCat)
```

1.3 Create new variables:

Let's group the yearly contributions by the categories that the university is interested in; Classify the "next degree" variable into 0 (representing "none") and 1 (representing there was some next-degree). Create indicator variables for each year for giver(1) or not a giver(0). For each alumnus let's also count the number of years they have given between 2012 and 2015.

```
dt$FY16GivingCat <- cut(dt$FY16Giving, c(0, 1, 100, 250, 500,
    2e+05), right = FALSE)
describe(dt$FY16GivingCat)
```

```
## dt$FY16GivingCat
##        n  missing distinct
##     1000        0        5
##
## Value            [0,1)     [1,100)   [100,250)   [250,500) [500,2e+05)
## Frequency          586         173         143          39          59
## Proportion       0.586       0.173       0.143       0.039       0.059
```

```
dt$FY15GivingCat <- cut(dt$FY15Giving, c(0, 1, 100, 250, 500,
    2e+05), right = FALSE)
describe(dt$FY15GivingCat)
```

```
## dt$FY15GivingCat
##        n  missing distinct
##     1000        0        5
##
## Value            [0,1)     [1,100)   [100,250)   [250,500) [500,2e+05)
## Frequency          567         199         138          36          60
## Proportion       0.567       0.199       0.138       0.036       0.060
```

```
dt$FY14GivingCat <- cut(dt$FY14Giving, c(0, 1, 100, 250, 500,
    2e+05), right = FALSE)
describe(dt$FY14GivingCat)
```

```
## dt$FY14GivingCat
##        n  missing distinct
##     1000        0        5
```

```
##
## Value              [0,1)     [1,100)   [100,250)   [250,500) [500,2e+05)
## Frequency            553         226         136          36          49
## Proportion         0.553       0.226       0.136       0.036       0.049
```

```
dt$FY13GivingCat <- cut(dt$FY13Giving, c(0, 1, 100, 250, 500,
    2e+05), right = FALSE)
describe(dt$FY13GivingCat)
```

```
## dt$FY13GivingCat
##        n  missing distinct
##     1000        0        5
##
## Value              [0,1)     [1,100)   [100,250)   [250,500) [500,2e+05)
## Frequency            513         247         143          54          43
## Proportion         0.513       0.247       0.143       0.054       0.043
```

```
dt$FY12GivingCat <- cut(dt$FY12Giving, c(0, 1, 100, 250, 500,
    2e+05), right = FALSE)
describe(dt$FY12GivingCat)
```

```
## dt$FY12GivingCat
##        n  missing distinct
##     1000        0        5
##
## Value              [0,1)     [1,100)   [100,250)   [250,500) [500,2e+05)
## Frequency            558         213         149          37          43
## Proportion         0.558       0.213       0.149       0.037       0.043
```

```
# create an indicator for 'giver' and 'non giver' for each
# year.
dt$Giver16 = as.integer(dt$FY16Giving > 0)
dt$Giver15 = as.integer(dt$FY15Giving > 0)
dt$Giver14 = as.integer(dt$FY14Giving > 0)
dt$Giver13 = as.integer(dt$FY13Giving > 0)
dt$Giver12 = as.integer(dt$FY12Giving > 0)
dt$YearsGiven = dt$Giver12 + dt$Giver13 + dt$Giver14 + dt$Giver15

# Create identifier for next degree (1) or none (0)
dt$NextDegCat = 1 - as.integer((dt$Next.Degree == "NONE"))

# Group majors by broad categories

# I think we should consider grouping majors below a certain
# threshold as 'Rare' vs. 'Common'.  How were these
# categories decided upon?

# AY-I think we should consider spliting 'science' into
# 'STEM' and 'social Science' dt$MajorCat = ifelse(dt$Major
# %in% c('American
# Studies','Art','Chinese','Classics','Comparative
# #Literature','English','English-Journalism','French','German','History','Independent','Music','I
```

```r
# Education','Religious Studies','Russian','Spanish','Speech
# (Drama, etc.)','Theatre'), 'HUM_ART', ifelse(dt$Major %in%
# #c('Biology','Chemistry','Computer
# Science','Engineering','General Science','General
# Science-Biology','General #Science-Chemistry','General
# Science-Math','General
# Science-Physics','Mathematics','Mathematics-Physics','Physics','Zoology'
# ), 'SCIENCE','OTHER'))

# Alternative Grouping:
dt$MajorCat = ifelse(dt$Major %in% c("American Studies", "Art",
    "Chinese", "Classics", "Comparative Literature", "English",
    "English-Journalism", "French", "German", "History", "Independent",
    "Music", "Philosophy", "Philosophy-Religion", "Physical Education",
    "Religious Studies", "Russian", "Spanish", "Speech (Drama, etc.)",
    "Theatre"), "HUM_ART", ifelse(dt$Major %in% c("Biology",
    "Chemistry", "Computer Science", "Engineering", "General Science",
    "General Science-Biology", "General Science-Chemistry", "General Science-Math",
    "General Science-Physics", "Mathematics", "Mathematics-Physics",
    "Physics", "Zoology"), "STEM", ifelse(dt$Major %in% c("Economics",
    "Economics-Regional Stds.", "Sociology", "Pyschology", "Pol. Sci.-Regional Stds.",
    "Sociology-Anthropology", "Political Science", "Anthropology",
    "Economics-Business"), "SOCIAL_SCIENCE", "OTHER")))

# Maybe get rid of the 'Professional' category. Not sure
# there's that much m=similarity between journalism and
# business majors


dt$MaritalStatusCat = factor(dt$Marital.Status)
dt$ClassYearCat = factor(dt$Class.Year)
```

2.2 Exploratory Data Analysis:

2.2.1 Univariate Analysis:

Let's examine each variable first starting with the "variable of interest" - 2016 contributions.

FY16Giving: Given that the vast majority of people did not give in 2016 and the skewness of the data (with a few large contributions) - let's also look at the distribution after a log-transformation (this is something we may want to consider for our modeling purposes)

Let's do a log-transformation to see the distribution more clearly (note: 0 contributions are excluded in the log-transformed plot). The log-transformed distribution looks somewhat normal - with a few contributions in the tens of thousands of dollars.

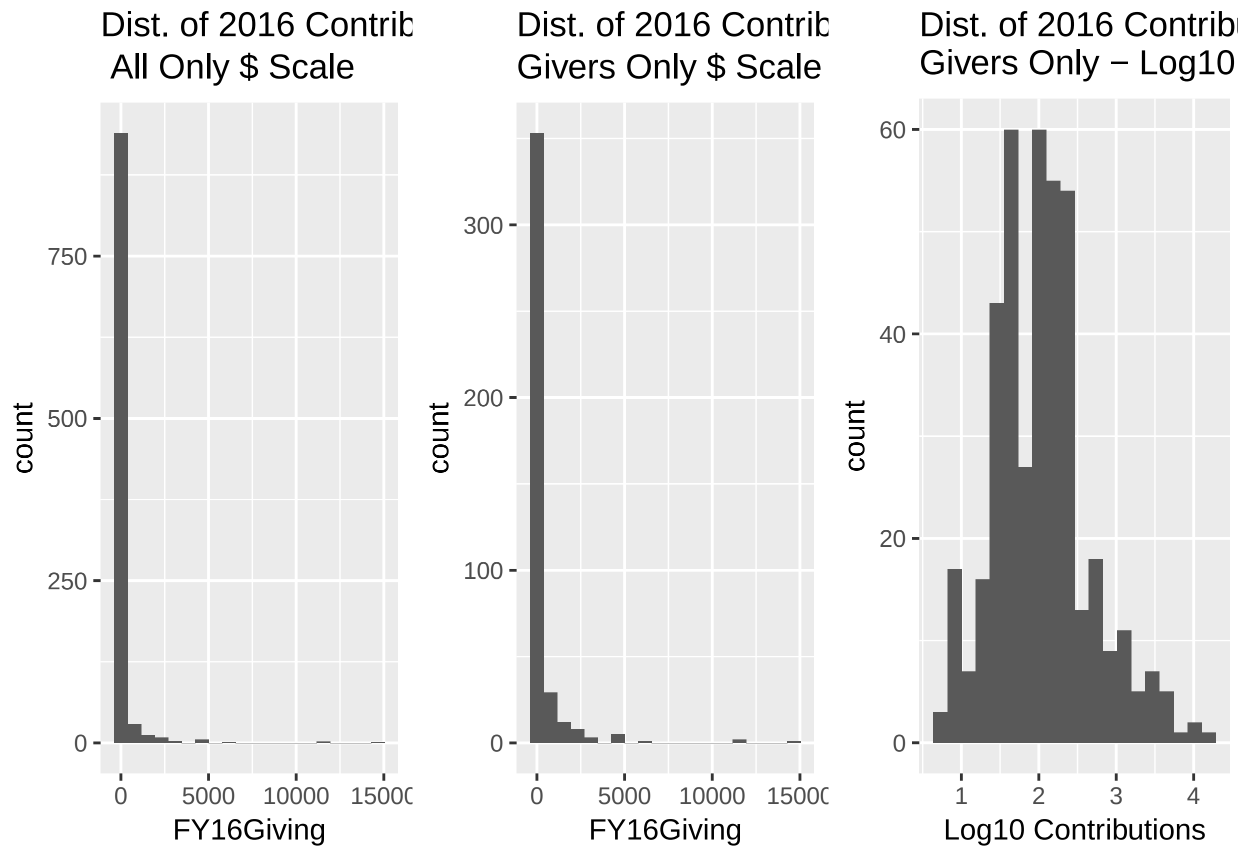
Let's look at a distribution of just the "givers" (i.e. take out the zero contributions) to get a better picture.

```r
h1 = ggplot(data = dt, aes(x = FY16Giving)) + geom_histogram(bins = 20) +
    ggtitle("Dist. of 2016 Contributions \n All Only $ Scale") +
    theme(plot.title = element_text(lineheight = 1))
```

```r
h2 = ggplot(data = dt[FY16Giving > 0], aes(x = FY16Giving)) +
    geom_histogram(bins = 20) + ggtitle("Dist. of 2016 Contributions \nGivers Only $ Scale") +
    theme(plot.title = element_text(lineheight = 1))
h3 = ggplot(data = dt[FY16Giving > 0], aes(x = log10(FY16Giving))) +
    geom_histogram(bins = 20) + ggtitle("Dist. of 2016 Contributions\nGivers Only - Log10 $ Scale'
    xlab("Log10 Contributions")
theme(plot.title = element_text(lineheight = 1, face = "bold"))
```

```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : chr "bold"
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : num 1
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

```r
grid.arrange(h1, h2, h3, ncol = 3)
```



Most of the alumni contributed $0; so we see a big spike at 0 and then it tapers off quickly. When we filter for alumni who donated more than 0, we again see a very similar pattern-most alumni who did contribute contributed very little. The distribution is still very skewed;a log-transform might help us see the distribution better.

The alumni donations for 2012 through 2016 all have similar 5th, 10th, 25th, 50th, 75th, 90th, and 95th percentiles and have distributions similar to that of 2016. Each year, roughly half the alumni population do not give anything. 20-25%% give less than $100; Around 14% Give $100-250; About 4% give $250-500 and less than 1% give more than $500. In 2013, we had the highest percentage of the Alumni contributing (about 49%) and 2015 marked the lowest % giving (43%) #How do you know? Where is the output of this? -Should we put a bunch of small histograms of the other years' giving, or can we just point to the previous out of the

Where's the rest of the univariate analysis? - Is there much else to do? All the rest are category or binary variables, and are described clearly bt the "describe" function

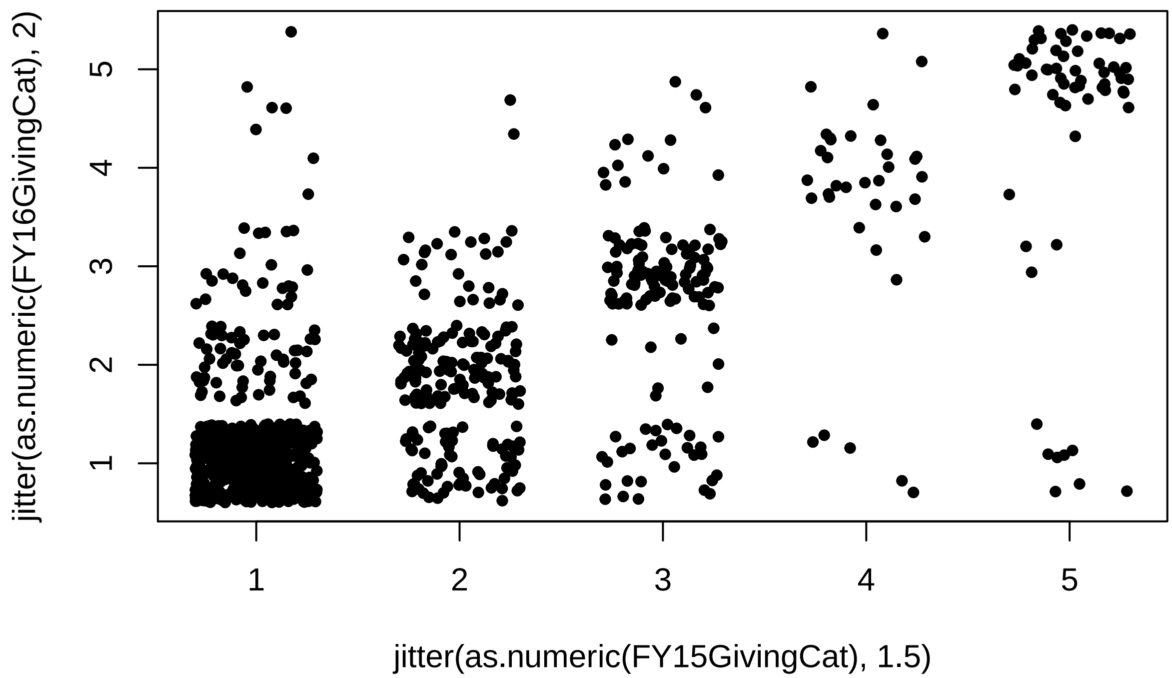
2.2.2 Bivariate Analysis:

Giving in 2016 vs. 2015:

Let's look at how 2016 giving relates to 2015. The plot shows that there is a reasonable correlation between the two (note the density of dots along the diagonal). It is also statistically confirmed by the Likelihood Ratio Test for independence where we reject the H_0: Giving Category in 2016 is Independent of Giving category in 2015.
We also conducted the test for each of the other years. Each of the tests show that there is a dependence

between 2016 giving and past years' giving category. Essentially, what we are observing is that someone's giving category in 2016 is most likely to be the same as their past; it is also interesting to note that in most instances the second highest category is the "[0,1)" category - so basically either they give like they have given in the past or not give at all!

```
plot(jitter(as.numeric(FY16GivingCat), 2) ~ jitter(as.numeric(FY15GivingCat),
    1.5), data = dt, pch = 20)
```




```
# Generic function to perform xtab on two variables and
# conduct LRT test of independence.

# Let's comment on assocstats

GenXtab = function(dframe, x1, x2, nlist) {
    x1vsx2 = xtabs(formula = ~x1 + x2, data = dframe)
    names(dimnames(x1vsx2)) = nlist
    print(x1vsx2)
    print("Percentage of Column Totals Shown Below")
    print(round(prop.table(x1vsx2, 2), 2))
    a.s = assocstats(x1vsx2)
    a.s
    if (is.null(a.s$phi) | is.na(a.s$phi)) {
        print("Phi: Not Applicable")
    } else if (abs(a.s$phi) > 0.5) {
        print("Phi: Large Effect")
    } else if (abs(a.s$phi) > 0.3) {
        print("Phi: Medium Effect")
    } else if (abs(a.s$phi) > 0.1) {
        print("Phi: Small Effect")
    } else {
        print("Phi: Negligible Effect")
    }
```

```r
    if (is.null(a.s$contingency)) {
        print("Contingency Coef: Not Applicable")
    } else if (abs(a.s$contingency) > 0.5) {
        print("Contingency Coef: Large Effect")
    } else if (abs(a.s$contingency) > 0.3) {
        print("Contingency Coef: Medium Effect")
    } else if (abs(a.s$contingency) > 0.1) {
        print("Contingency Coef: Small Effect")
    } else {
        print("Contingency Coef: Negligible Effect")
    }

    if (is.null(a.s$cramer)) {
        print("Cramer's V: Not Applicable")
    } else if (abs(a.s$cramer) > 0.5) {
        print("Cramer's V: Large Effect")
    } else if (abs(a.s$cramer) > 0.3) {
        print("Cramer's V: Medium Effect")
    } else if (abs(a.s$cramer) > 0.1) {
        print("Cramer's V: Small Effect")
    } else {
        print("Cramer's V: Negligible Effect")
    }

}

GenXtab(dt, dt$Giver16, dt$Giver15, c("FY16", "FY15"))
```

```
##      FY15
## FY16   0   1
##    0 480 106
##    1  87 327
## [1] "Percentage of Column Totals Shown Below"
##      FY15
## FY16    0    1
##    0 0.85 0.24
##    1 0.15 0.76
## [1] "Phi: Large Effect"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Large Effect"
```

```r
GenXtab(dt, dt$FY16GivingCat, dt$FY15GivingCat, c("FY16", "FY15"))
```

```
##                FY15
## FY16            [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)           480      64        29         5           8
##   [1,100)          57     108         8         0           0
##   [100,250)        23      25        88         4           3
##   [250,500)         3       1        10        23           2
##   [500,2e+05)       4       1         3         4          47
```

```
## [1] "Percentage of Column Totals Shown Below"
##              FY15
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)         0.85    0.32      0.21      0.14        0.13
##   [1,100)       0.10    0.54      0.06      0.00        0.00
##   [100,250)     0.04    0.13      0.64      0.11        0.05
##   [250,500)     0.01    0.01      0.07      0.64        0.03
##   [500,2e+05)   0.01    0.01      0.02      0.11        0.78
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Large Effect"
```

```r
GenXtab(dt, dt$Giver16, dt$Giver14, c("FY16", "FY14"))
```

```
##     FY14
## FY16   0   1
##    0 461 125
##    1  92 322
## [1] "Percentage of Column Totals Shown Below"
##     FY14
## FY16    0    1
##    0 0.83 0.28
##    1 0.17 0.72
## [1] "Phi: Large Effect"
## [1] "Contingency Coef: Medium Effect"
## [1] "Cramer's V: Large Effect"
```

```r
GenXtab(dt, dt$FY16GivingCat, dt$FY14GivingCat, c("FY16", "FY14"))
```

```
##              FY14
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)          461      82        34         4           5
##   [1,100)         56     108         8         1           0
##   [100,250)       23      33        74         8           5
##   [250,500)        5       2        15        17           0
##   [500,2e+05)      8       1         5         6          39
## [1] "Percentage of Column Totals Shown Below"
##              FY14
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)         0.83    0.36      0.25      0.11        0.10
##   [1,100)       0.10    0.48      0.06      0.03        0.00
##   [100,250)     0.04    0.15      0.54      0.22        0.10
##   [250,500)     0.01    0.01      0.11      0.47        0.00
##   [500,2e+05)   0.01    0.00      0.04      0.17        0.80
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Large Effect"
```

```r
GenXtab(dt, dt$Giver16, dt$Giver13, c("FY16", "FY13"))
```

```
##     FY13
## FY16   0   1
##    0 441 145
```

```
##    1  72 342
## [1] "Percentage of Column Totals Shown Below"
##     FY13
## FY16    0    1
##    0 0.86 0.30
##    1 0.14 0.70
## [1] "Phi: Large Effect"
## [1] "Contingency Coef: Medium Effect"
## [1] "Cramer's V: Large Effect"
```

```
GenXtab(dt, dt$FY16GivingCat, dt$FY13GivingCat, c("FY16", "FY13"))
```

```
##               FY13
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)          441      94        40         7           4
##   [1,100)         39     123        10         1           0
##   [100,250)       19      27        73        19           5
##   [250,500)        5       0        13        18           3
##   [500,2e+05)      9       3         7         9          31
## [1] "Percentage of Column Totals Shown Below"
##               FY13
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)         0.86    0.38      0.28      0.13        0.09
##   [1,100)       0.08    0.50      0.07      0.02        0.00
##   [100,250)     0.04    0.11      0.51      0.35        0.12
##   [250,500)     0.01    0.00      0.09      0.33        0.07
##   [500,2e+05)  0.02    0.01      0.05      0.17        0.72
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Large Effect"
```

```
GenXtab(dt, dt$Giver16, dt$Giver12, c("FY16", "FY12"))
```

```
##     FY12
## FY16   0   1
##    0 462 124
##    1  96 318
## [1] "Percentage of Column Totals Shown Below"
##     FY12
## FY16    0    1
##    0 0.83 0.28
##    1 0.17 0.72
## [1] "Phi: Large Effect"
## [1] "Contingency Coef: Medium Effect"
## [1] "Cramer's V: Large Effect"
```

```
GenXtab(dt, dt$FY16GivingCat, dt$FY12GivingCat, c("FY16", "FY12"))
```

```
##               FY12
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)          462      73        38         9           4
##   [1,100)         60      96        16         0           1
##   [100,250)       26      40        69         5           3
```

```
##   [250,500)       4       2        16         16             1
##   [500,2e+05)     6       2        10          7            34
## [1] "Percentage of Column Totals Shown Below"
##              FY12
## FY16           [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)         0.83    0.34      0.26      0.24        0.09
##   [1,100)       0.11    0.45      0.11      0.00        0.02
##   [100,250)     0.05    0.19      0.46      0.14        0.07
##   [250,500)     0.01    0.01      0.11      0.43        0.02
##   [500,2e+05)  0.01    0.01      0.07      0.19        0.79
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Large Effect"
```

```
GenXtab(dt, dt$Giver16, dt$YearsGiven, c("FY16", "YearsGiven"))
```

```
##     YearsGiven
## FY16   0   1   2   3   4
##    0 343  94  68  54  27
##    1  21  26  56  73 238
## [1] "Percentage of Column Totals Shown Below"
##     YearsGiven
## FY16    0    1    2    3    4
##    0 0.94 0.78 0.55 0.43 0.10
##    1 0.06 0.22 0.45 0.57 0.90
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Large Effect"
```

```
GenXtab(dt, dt$FY16GivingCat, dt$YearsGiven, c("FY16", "YearsGiven"))
```

```
##              YearsGiven
## FY16            0   1   2   3   4
##   [0,1)       343  94  68  54  27
##   [1,100)      15  15  36  35  72
##   [100,250)     4   7  14  26  92
##   [250,500)     1   1   2   6  29
##   [500,2e+05)   1   3   4   6  45
## [1] "Percentage of Column Totals Shown Below"
##              YearsGiven
## FY16             0    1    2    3    4
##   [0,1)       0.94 0.78 0.55 0.43 0.10
##   [1,100)     0.04 0.12 0.29 0.28 0.27
##   [100,250)   0.01 0.06 0.11 0.20 0.35
##   [250,500)   0.00 0.01 0.02 0.05 0.11
##   [500,2e+05) 0.00 0.02 0.03 0.05 0.17
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Large Effect"
## [1] "Cramer's V: Medium Effect"
```

```
GenXtab(dt, dt$Giver16, dt$MajorCat, c("FY16", "MajorCat"))
```

```
##     MajorCat
```

```
## FY16 HUM_ART OTHER SOCIAL_SCIENCE STEM
##    0     272    37            156  121
##    1     179    38            104   93
## [1] "Percentage of Column Totals Shown Below"
##     MajorCat
## FY16 HUM_ART OTHER SOCIAL_SCIENCE STEM
##    0    0.60  0.49           0.60 0.57
##    1    0.40  0.51           0.40 0.43
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Negligible Effect"
## [1] "Cramer's V: Negligible Effect"
```

```r
GenXtab(dt, dt$FY16GivingCat, dt$MajorCat, c("FY16", "MajorCat"))
```

```
##              MajorCat
## FY16           HUM_ART OTHER SOCIAL_SCIENCE STEM
##   [0,1)            272    37            156  121
##   [1,100)           72    14             46   41
##   [100,250)         65    14             32   32
##   [250,500)         19     2             10    8
##   [500,2e+05)       23     8             16   12
## [1] "Percentage of Column Totals Shown Below"
##              MajorCat
## FY16           HUM_ART OTHER SOCIAL_SCIENCE STEM
##   [0,1)           0.60  0.49           0.60 0.57
##   [1,100)         0.16  0.19           0.18 0.19
##   [100,250)       0.14  0.19           0.12 0.15
##   [250,500)       0.04  0.03           0.04 0.04
##   [500,2e+05)     0.05  0.11           0.06 0.06
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Negligible Effect"
## [1] "Cramer's V: Negligible Effect"
```

```r
GenXtab(dt, dt$Giver16, dt$NextDegCat, c("FY16", "NextDegCat"))
```

```
##     NextDegCat
## FY16   0   1
##    0 272 314
##    1 106 308
## [1] "Percentage of Column Totals Shown Below"
##     NextDegCat
## FY16    0    1
##    0 0.72 0.50
##    1 0.28 0.50
## [1] "Phi: Small Effect"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

```r
GenXtab(dt, dt$FY16GivingCat, dt$NextDegCat, c("FY16", "NextDegCat"))
```

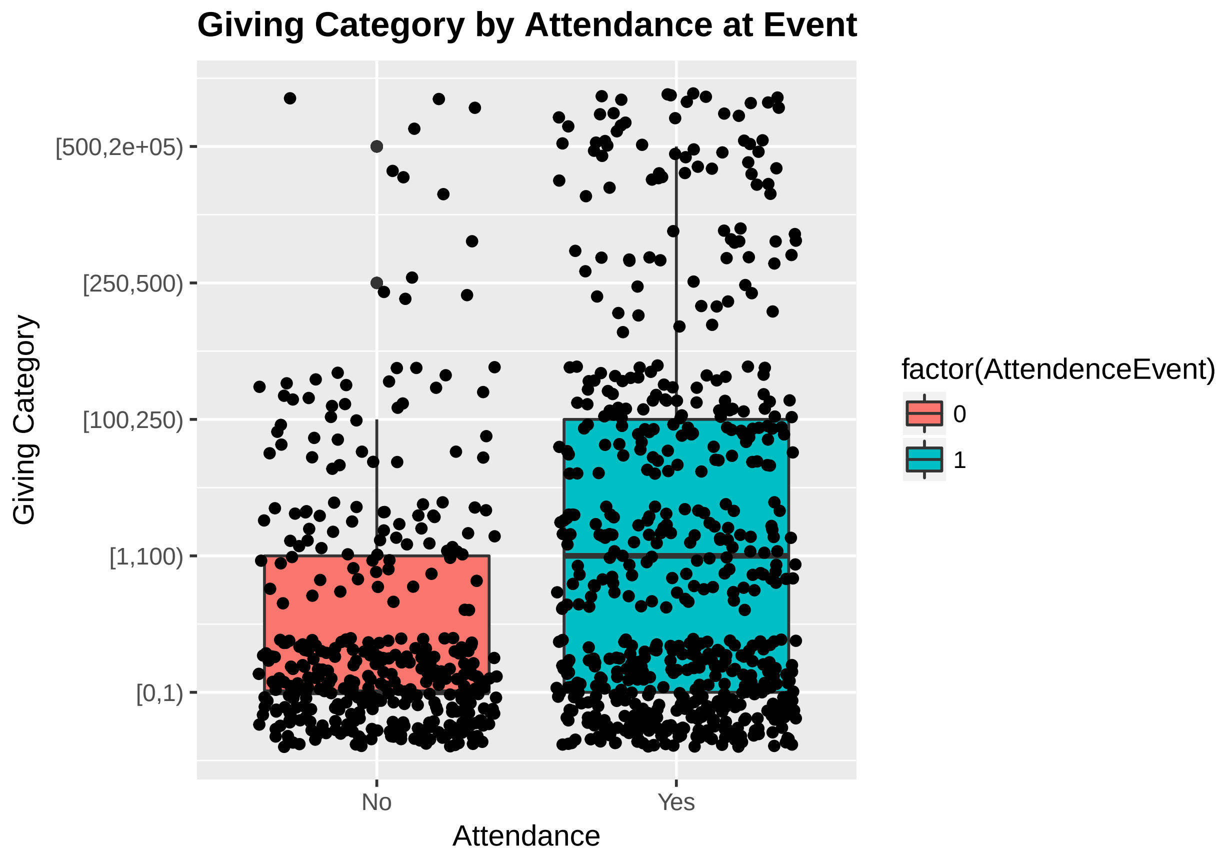
```
##              NextDegCat
## FY16            0   1
```

```
##   [0,1)       272 314
##   [1,100)      41 132
##   [100,250)    41 102
##   [250,500)    15  24
##   [500,2e+05)   9  50
## [1] "Percentage of Column Totals Shown Below"
##              NextDegCat
## FY16             0    1
##   [0,1)       0.72 0.50
##   [1,100)     0.11 0.21
##   [100,250)   0.11 0.16
##   [250,500)   0.04 0.04
##   [500,2e+05) 0.02 0.08
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

Do the xtabs take up too much space?

Giving in 2016 vs. Alumni Event Attendance:

```
ggplot(dt, aes(factor(AttendenceEvent), as.numeric(FY16GivingCat))) +
    geom_boxplot(aes(fill = factor(AttendenceEvent))) + ggtitle("Giving Category by Attendance at
    geom_jitter() + scale_x_discrete(name = "Attendance", labels = c("No",
    "Yes")) + scale_y_continuous(name = "Giving Category", breaks = 1:5,
    labels = c("[0,1)", "[1,100)", "[100,250)", "[250,500)",
        "[500,2e+05)")) + theme(plot.title = element_text(lineheight = 1,
    face = "bold"))
```




```
GenXtab(dt, dt$Giver16, dt$AttendenceEvent, c("FY16", "Attendance"))
```

```
##      Attendance
## FY16   0   1
##    0 286 300
##    1 109 305
## [1] "Percentage of Column Totals Shown Below"
##      Attendance
## FY16    0    1
##    0 0.72 0.50
##    1 0.28 0.50
## [1] "Phi: Small Effect"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

```
GenXtab(dt, dt$FY16GivingCat, dt$AttendenceEvent, c("FY16", "Attendance"))
```

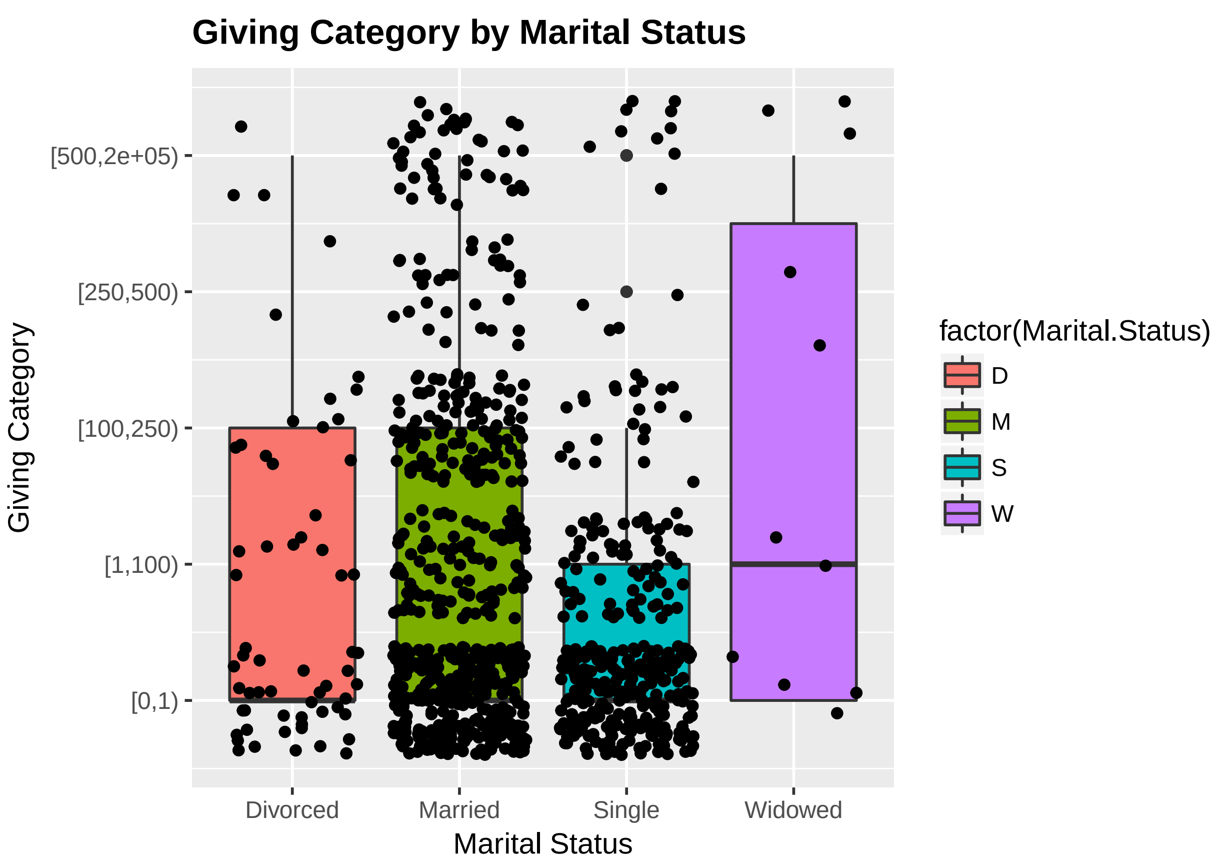
```
##               Attendance
## FY16             0   1
##   [0,1)        286 300
##   [1,100)       61 112
##   [100,250)     36 107
##   [250,500)      5  34
##   [500,2e+05)    7  52
## [1] "Percentage of Column Totals Shown Below"
##               Attendance
## FY16              0    1
```

```
##    [0,1)        0.72 0.50
##    [1,100)      0.15 0.19
##    [100,250)    0.09 0.18
##    [250,500)    0.01 0.06
##    [500,2e+05)  0.02 0.09
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

So there does seem to be a relationship between attendence at alumni events in 2012-2015 and 2016 giving. 50% of those attending gave in 2016 while only 28% of those not attending gave in 2016.

Giving in 2016 by Marital status:

```
ggplot(dt, aes(factor(Marital.Status), as.numeric(FY16GivingCat))) +
    geom_boxplot(aes(fill = factor(Marital.Status))) + ggtitle("Giving Category by Marital Status"
    geom_jitter() + scale_x_discrete(name = "Marital Status",
    labels = c("Divorced", "Married", "Single", "Widowed")) +
    scale_y_continuous(name = "Giving Category", breaks = 1:5,
        labels = c("[0,1)", "[1,100)", "[100,250)", "[250,500)",
            "[500,2e+05)")) + theme(plot.title = element_text(lineheight = 1,
    face = "bold"))
```




```
GenXtab(dt, dt$Giver16, dt$Marital.Status, c("FY16", "Marital Status"))
```

```
##     Marital Status
```

```
## FY16   D   M   S   W
##    0  36 305 241   4
##    1  25 279 103   7
## [1] "Percentage of Column Totals Shown Below"
##     Marital Status
## FY16    D    M    S    W
##    0 0.59 0.52 0.70 0.36
##    1 0.41 0.48 0.30 0.64
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

```r
GenXtab(dt, dt$FY16GivingCat, dt$Marital.Status, c("FY16", "Marital Status"))
```

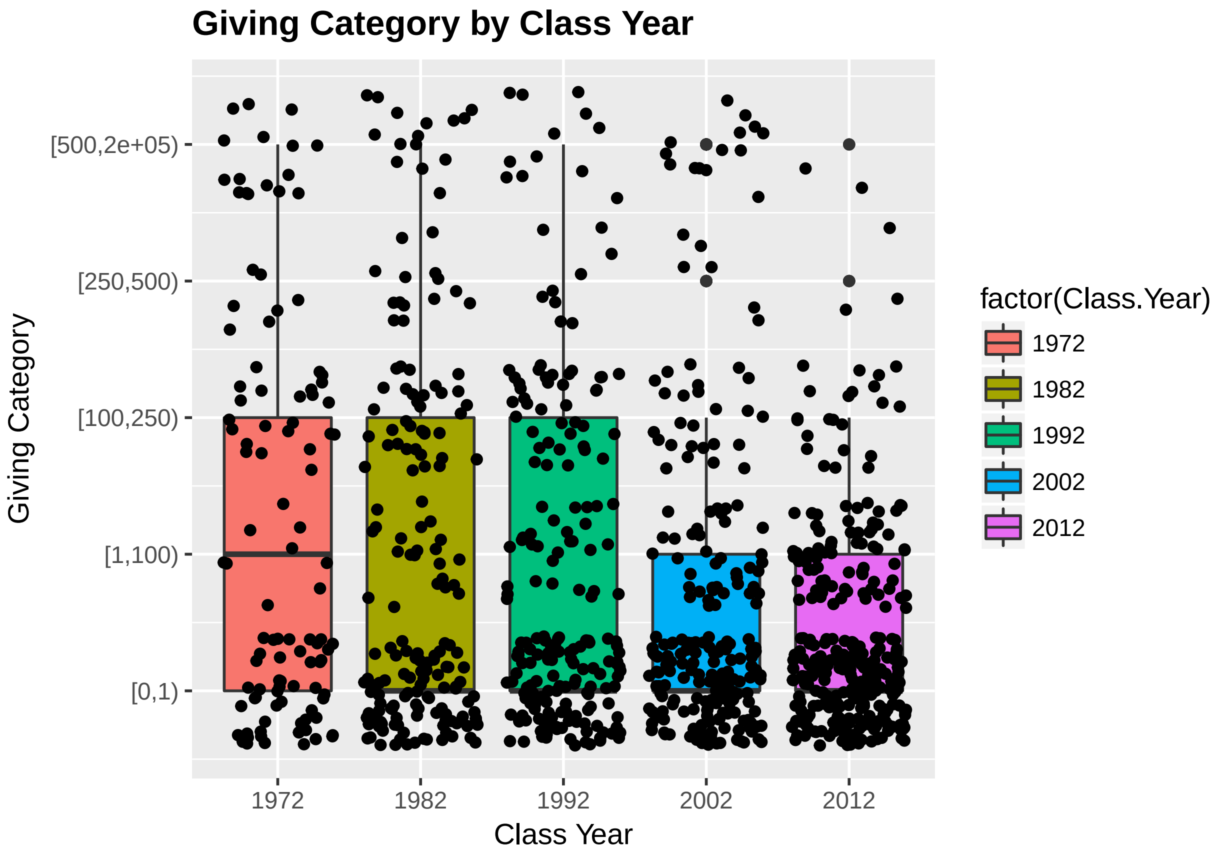
```
##              Marital Status
## FY16            D   M   S   W
##   [0,1)        36 305 241   4
##   [1,100)       9  96  66   2
##   [100,250)    11 109  23   0
##   [250,500)     2  31   4   2
##   [500,2e+05)   3  43  10   3
## [1] "Percentage of Column Totals Shown Below"
##              Marital Status
## FY16             D    M    S    W
##   [0,1)       0.59 0.52 0.70 0.36
##   [1,100)     0.15 0.16 0.19 0.18
##   [100,250)   0.18 0.19 0.07 0.00
##   [250,500)   0.03 0.05 0.01 0.18
##   [500,2e+05) 0.05 0.07 0.03 0.27
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

Most of the Alumni fall into either Married or Single category and Married Alumni are more likely to give than Single.
There are very few data points for Widowed and Divorced alumni - so we do not want to make broad conclusions, but it appears that there are both Divorced and Widowed Alumni that contribute high amounts (and there are those that contribute nothing too in these categories) One possible reason we are seeing Married giving more than Single might actually have to do with age. Older Alumni are more likely to be married and older alumni are also probably a bit more well established financially - so more likely to contribute to charitable causes. So Marital status vs. Giving might simply be capturing the relationship between Age and Giving. While age is not a variable that's available in the dataset, we have "Class Year" which is a good proxy for age.

Giving 2016 vs. Class Year:

```r
ggplot(dt, aes(factor(Class.Year), as.numeric(FY16GivingCat))) +
    geom_boxplot(aes(fill = factor(Class.Year))) + ggtitle("Giving Category by Class Year") +
    geom_jitter() + scale_x_discrete(name = "Class Year") + scale_y_continuous(name = "Giving Cate
    breaks = 1:5, labels = c("[0,1)", "[1,100)", "[100,250)",
```

```
        "[250,500)", "[500,2e+05)")) + theme(plot.title = element_text(lineheight = 1,
    face = "bold"))
```




```
GenXtab(dt, dt$Giver16, dt$Class.Year, c("FY16", "Class"))
```

```
##      Class
## FY16 1972 1982 1992 2002 2012
##    0   50   90  115  137  194
##    1   55   86   88   86   99
## [1] "Percentage of Column Totals Shown Below"
##      Class
## FY16 1972 1982 1992 2002 2012
##    0 0.48 0.51 0.57 0.61 0.66
##    1 0.52 0.49 0.43 0.39 0.34
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

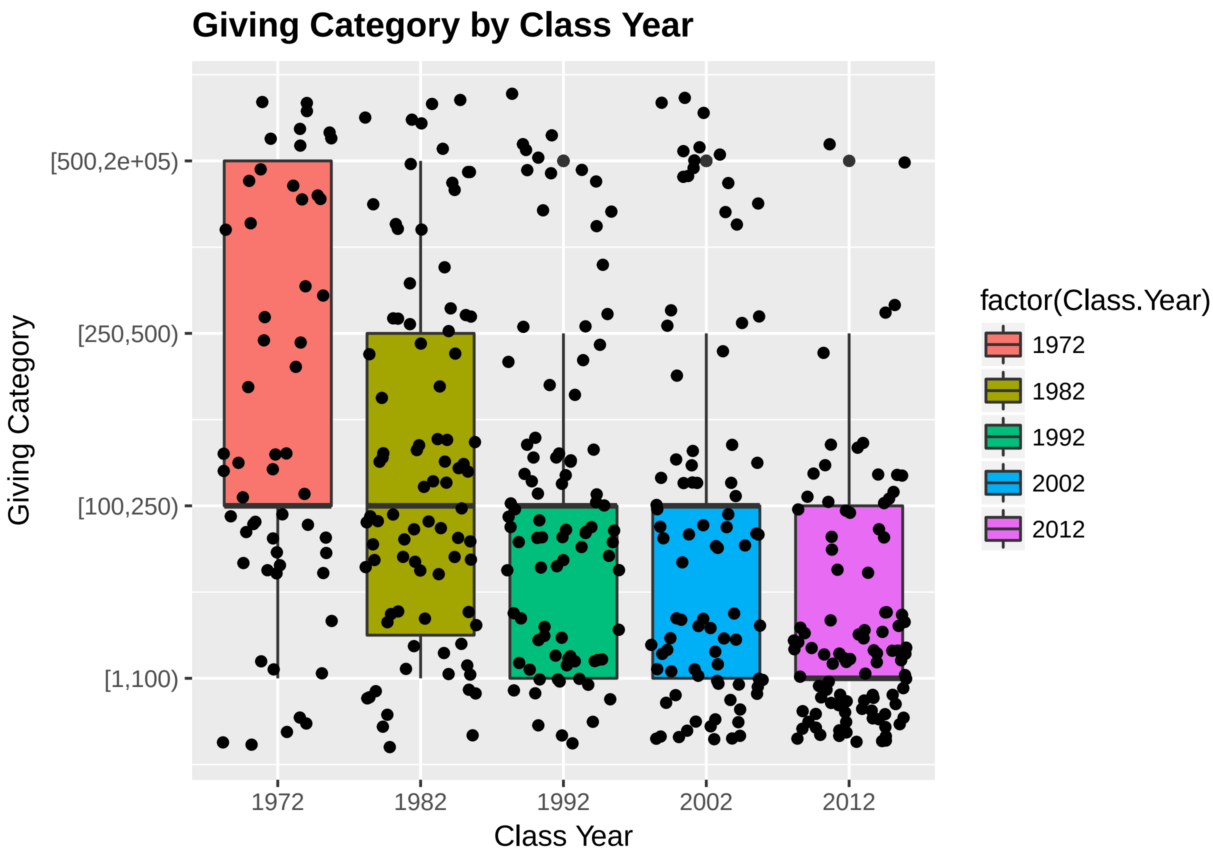
```
GenXtab(dt, dt$FY16GivingCat, dt$Class.Year, c("FY16", "Class"))
```

```
##               Class
## FY16           1972 1982 1992 2002 2012
##   [0,1)          50   90  115  137  194
##   [1,100)         9   22   29   41   72
##   [100,250)      23   35   38   25   22
##   [250,500)       7   14    9    6    3
##   [500,2e+05)    16   15   12   14    2
```

```
## [1] "Percentage of Column Totals Shown Below"
##                 Class
## FY16             1972 1982 1992 2002 2012
##   [0,1)          0.48 0.51 0.57 0.61 0.66
##   [1,100)        0.09 0.12 0.14 0.18 0.25
##   [100,250)      0.22 0.20 0.19 0.11 0.08
##   [250,500)      0.07 0.08 0.04 0.03 0.01
##   [500,2e+05) 0.15 0.09 0.06 0.06 0.01
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

Here, it indeed looks like the older the alumnus, the more likely he or she is to give money.

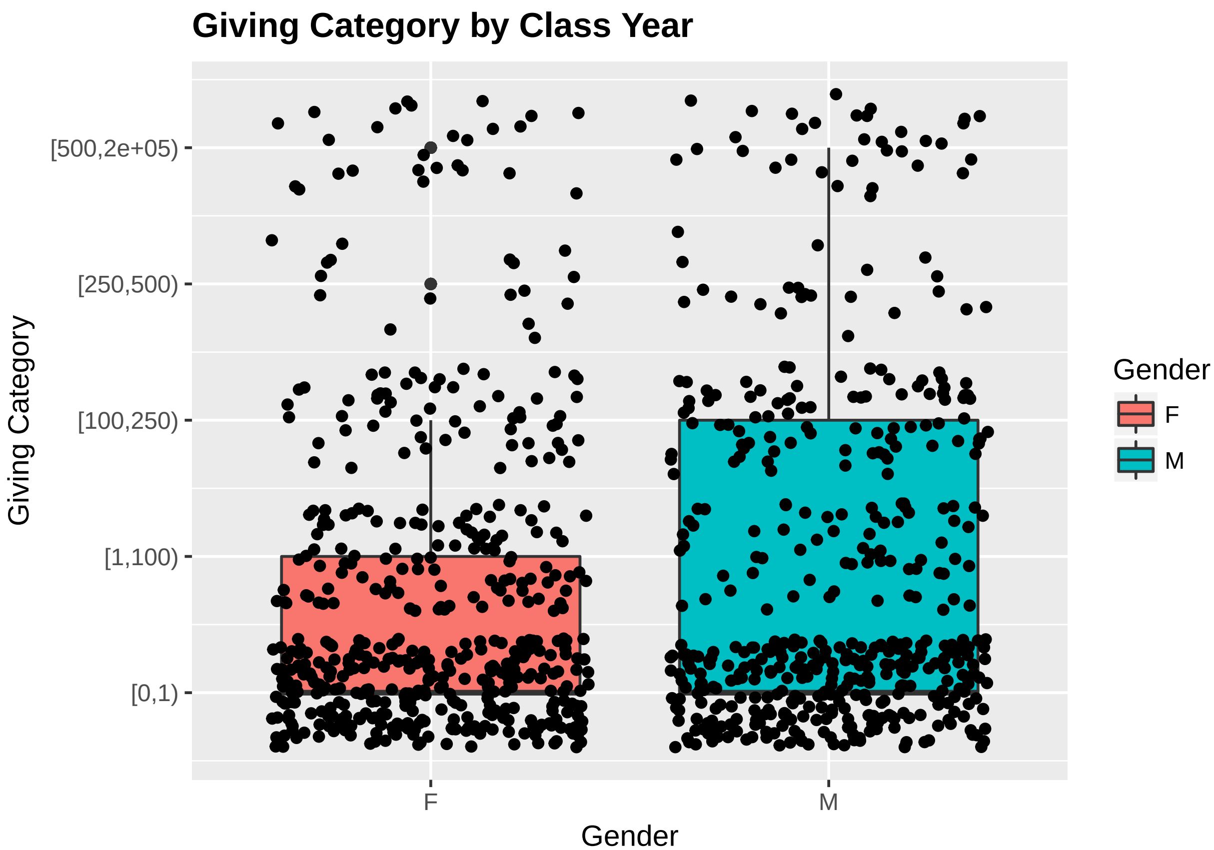
```
ggplot(dt[dt$FY16Giving > 0, ], aes(factor(Class.Year), as.numeric(FY16GivingCat))) +
    geom_boxplot(aes(fill = factor(Class.Year))) + ggtitle("Giving Category by Class Year") +
    geom_jitter() + scale_x_discrete(name = "Class Year") + scale_y_continuous(name = "Giving Cate
    breaks = 1:5, labels = c("[0,1)", "[1,100)", "[100,250)",
        "[250,500)", "[500,2e+05)")) + theme(plot.title = element_text(lineheight = 1,
    face = "bold"))
```



Giving in 2016 by Gender:

```
ggplot(dt, aes(Gender, as.numeric(FY16GivingCat))) + geom_boxplot(aes(fill = Gender)) +
    ggtitle("Giving Category by Class Year") + geom_jitter() +
    scale_x_discrete(name = "Gender") + scale_y_continuous(name = "Giving Category",
```

```r
    breaks = 1:5, labels = c("[0,1)", "[1,100)", "[100,250)",
        "[250,500)", "[500,2e+05)")) + theme(plot.title = element_text(lineheight = 1,
    face = "bold"))
```




```r
GenXtab(dt, dt$Giver16, dt$Gender, c("Giver16", "Gender"))
```

```
##          Gender
## Giver16    F   M
##        0 298 288
##        1 207 207
## [1] "Percentage of Column Totals Shown Below"
##          Gender
## Giver16     F    M
##        0 0.59 0.58
##        1 0.41 0.42
## [1] "Phi: Negligible Effect"
## [1] "Contingency Coef: Negligible Effect"
## [1] "Cramer's V: Negligible Effect"
```

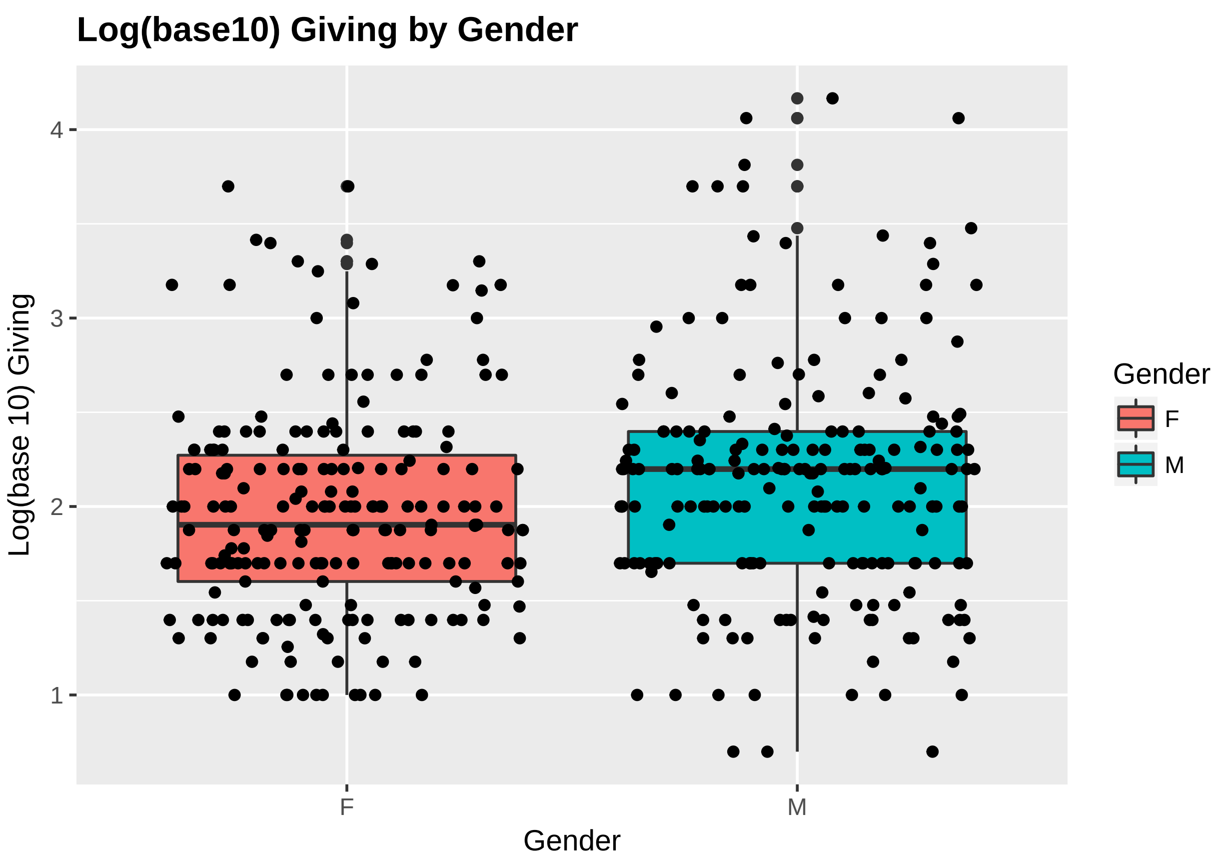
```r
GenXtab(dt, dt$FY16GivingCat, dt$Gender, c("FY16", "Gender"))
```

```
##              Gender
## FY16            F   M
##   [0,1)       298 288
##   [1,100)     106  67
##   [100,250)    58  85
##   [250,500)    17  22
```

```
##   [500,2e+05)  26  33
## [1] "Percentage of Column Totals Shown Below"
##              Gender
## FY16            F    M
##   [0,1)       0.59 0.58
##   [1,100)     0.21 0.14
##   [100,250)   0.11 0.17
##   [250,500)   0.03 0.04
##   [500,2e+05) 0.05 0.07
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

There is no statistical evidence that whether or not Alumni Give in 2016 varies by Gender (41% of Females Gave vs. 42% of Males). However when we look at the categories of contribution in 2016 by Gender, we see differences worth investigating later. There may be other factors at play here, for e.g. we know that older alumni tend to give more than younger alumni; it is possible that there are fewer "older" female alumni (fewer women attended college in 1972) than Male. > *AY Note: Not sure we should be rejecting hypothesis in the EDA stage*

```r
ggplot(dt[FY16Giving > 0], aes(Gender, log10(FY16Giving))) +
    geom_boxplot(aes(fill = Gender)) + ggtitle("Log(base10) Giving by Gender") +
    geom_jitter() + scale_x_discrete(name = "Gender") + scale_y_continuous(name = "Log(base 10) Gi
    theme(plot.title = element_text(lineheight = 1, face = "bold"))
```



Of those that donate, men donate more. But we can't be sure this is significant as the median man donated less than the 75th percentile woman.

Is there a relationship between Gender and Class Year i.e. do we see a higher proportion of "Men" in the sample as age goes up (proxy for age is Class Year)?

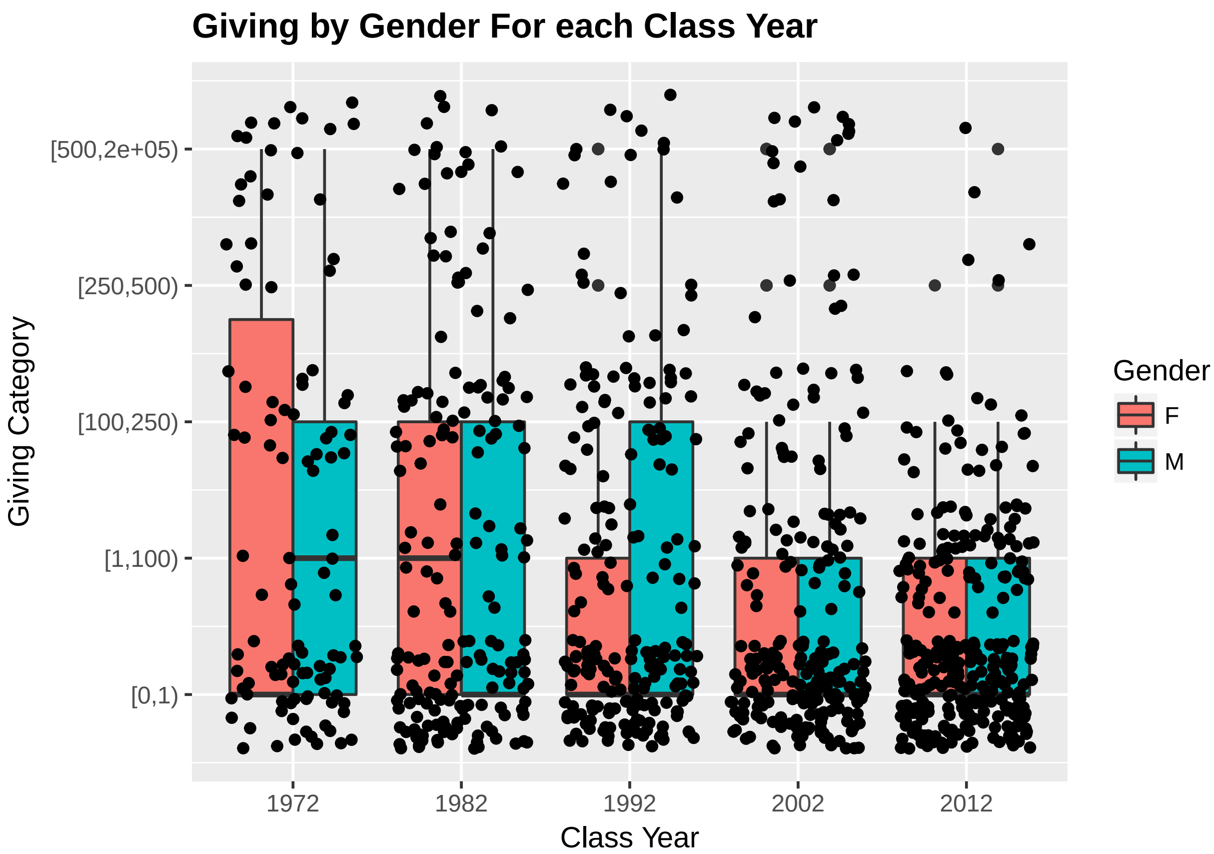
```
GenXtab(dt, dt$Gender, dt$Class.Year, c("Gender", "ClassYear"))
```

```
##       ClassYear
## Gender 1972 1982 1992 2002 2012
##      F   38   80  102  133  152
##      M   67   96  101   90  141
## [1] "Percentage of Column Totals Shown Below"
##       ClassYear
## Gender 1972 1982 1992 2002 2012
##      F 0.36 0.45 0.50 0.60 0.52
##      M 0.64 0.55 0.50 0.40 0.48
## [1] "Phi: Not Applicable"
## [1] "Contingency Coef: Small Effect"
## [1] "Cramer's V: Small Effect"
```

We do see that there are a higher proportion of Men in 1972 & 1982 compared to more recent years. Based the test for independence, we see that there is dependence between "Class Year" (or age) and Gender.

So let's look at Contribution by class year split by male and female:

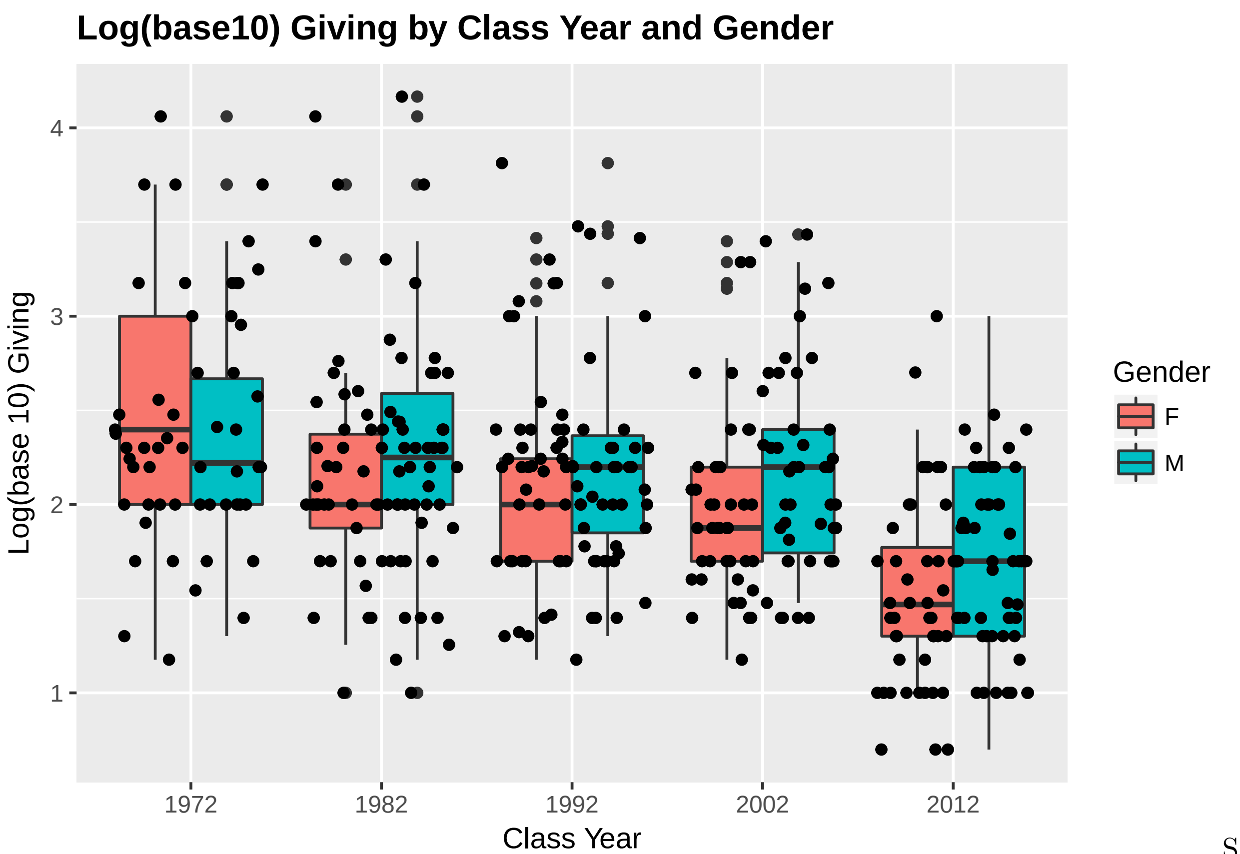
```
ggplot(dt, aes(factor(Class.Year), as.numeric(FY16GivingCat))) +
    geom_boxplot(aes(fill = Gender)) + ggtitle("Giving by Gender For each Class Year") +
    geom_jitter() + scale_x_discrete(name = "Class Year") + scale_y_continuous(name = "Giving Cate
    breaks = 1:5, labels = c("[0,1)", "[1,100)", "[100,250)",
        "[250,500)", "[500,2e+05)")) + theme(plot.title = element_text(lineheight = 1,
    face = "bold"))
```



So split

by class year, gender does not seem to be a significant factor, and in fact older women donate more than older men

```
ggplot(dt[FY16Giving > 0], aes(factor(Class.Year), log10(FY16Giving))) +
    geom_boxplot(aes(fill = Gender)) + ggtitle("Log(base10) Giving by Class Year and Gender") +
    geom_jitter() + scale_x_discrete(name = "Class Year") + scale_y_continuous(name = "Log(base 10
    theme(plot.title = element_text(lineheight = 1, face = "bold"))
```



SO limiting it to just alumni who donated in 2016, we do see that even split by age, men donated more than women, except for the oldest alumni

SECTION - 3 STATISTICAL MODELING:

Section 3: Statistical Modeling. Start the section summarizing the key results - what variables, if any, are the key predictors of the year 2016 contribution? What are the key techniques you have experimented? What method did you use in your final model? How did you choose the final model? What model performance criteria did you use to choose the final model? What statistical infernece did you perform? Explain them. Comment on statistical significance vs. economic significance.

```
# can delete this line below if dataset is already in
# workspace otherwise, run SECTION1.Rmd file (need this
# process of storing the data frame and loading it again if
# we knit the SECTION files separately)

# load('dt_dataframe.Rda')
```

```r
set.seed(1234)
train_index = createDataPartition(dt$FY16GivingCat, p = 0.8,
    list = FALSE, times = 1)
class(dt$FY16GivingCat)
```

```
## [1] "factor"
```

```r
dt_train = dt[train_index]
dt_test = dt[-train_index]


xtabs(~FY16GivingCat, data = dt_train)
```

```
## FY16GivingCat
##      [0,1)    [1,100)  [100,250)  [250,500) [500,2e+05)
##        469        139        115         32          48
```

```r
xtabs(~FY16GivingCat, data = dt_test)
```

```
## FY16GivingCat
##      [0,1)    [1,100)  [100,250)  [250,500) [500,2e+05)
##        117         34         28          7          11
```

```r
round(prop.table(xtabs(~FY16GivingCat, data = dt_train)), 2)
```

```
## FY16GivingCat
##      [0,1)    [1,100)  [100,250)  [250,500) [500,2e+05)
##       0.58       0.17       0.14       0.04        0.06
```

```r
round(prop.table(xtabs(~FY16GivingCat, data = dt_test)), 2)
```

```
## FY16GivingCat
##      [0,1)    [1,100)  [100,250)  [250,500) [500,2e+05)
##       0.59       0.17       0.14       0.04        0.06
```

```r
PredAcc = function(data_test, mod) {
    # Function to evaluate accuracy of model data_test: a data
    # frame or table thatcontains test data mod: model fit on
    # data

    # Predict the responses given model

    dep = names(mod$model[1])  #Name of Dependent Variable
    test.act = data_test[, ..dep]  #List of Actual Observed Outcomes
    if (class(mod)[1] == "glm") {
        test.prob = predict(mod, newdata = data_test, type = "response")  #Probability Binary is 1
        test.pred = ifelse(test.prob > 0.5, 1, 0)  #Predicted Result
    } else if (class(mod)[1] == "multinom") {
        test.prob = predict(mod, newdata = data_test, type = "probs")  #Probability of each catego
        test.pred = colnames(test.prob)[max.col(test.prob, ties.method = "random")]  #Category of
    }

    results = data.frame(pred = test.pred, act = test.act)
```

```r
    # If we decide not to use Confusion Matrix... accuracy =
    # round(mean(test.act == test.pred),3) #Part of
    # confusionMatrix GenXtab(dframe = results, x1 = results[,1],
    # x2 = results[,2], nlist = c('Predicted','Actual'))
    # paste('Overall Prediction Accuracy = ', accuracy)

    confusionMatrix(results[, 1], results[, 2], positive = "1")  #Analysis of Prediction Accuracy
}

Accuracy = function(CM_train, CM_test, model_name) {
    # Returns row with accuracy value for test and train sets for
    # the model
    round(data.frame(row.names = model_name, TrainAcc = CM_train$overall[1],
        TestAcc = CM_test$overall[1]), 2)

}

# models for determining whether someone will contribute or
# not
```

```r
# 2016 giving status as function of 2015
model.fit1a = glm(Giver16 ~ (Giver15), data = dt_train, family = binomial(link = "logit"))
summary(model.fit1a)
```

```
##
## Call:
## glm(formula = Giver16 ~ (Giver15), family = binomial(link = "logit"),
##     data = dt_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7125  -0.5888  -0.5888   0.7244   1.9173
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6646     0.1268  -13.13   <2e-16 ***
## Giver15       2.8686     0.1809   15.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1090.39  on 802  degrees of freedom
## Residual deviance:  772.74  on 801  degrees of freedom
## AIC: 776.74
##
## Number of Fisher Scoring iterations: 4
```

```r
Anova(model.fit1a, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
```

```
##
## Response: Giver16
##         LR Chisq Df Pr(>Chisq)
## Giver15   317.65  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
CMTr = PredAcc(dt_train, mod = model.fit1a)
CMTe = PredAcc(dt_test, mod = model.fit1a)
# initialize ac table for the first use later use rbind to
# add to the dataframe
ac_table_1 = Accuracy(CMTr, CMTe, "model.fit1a")

# 2016 giving status as function of prior year giving status
model.fit1b = glm(Giver16 ~ (Giver15 + Giver14 + Giver13 + Giver12),
    data = dt_train, family = binomial(link = "logit"))
summary(model.fit1b)
```

```
##
## Call:
## glm(formula = Giver16 ~ (Giver15 + Giver14 + Giver13 + Giver12),
##     family = binomial(link = "logit"), data = dt_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0766  -0.3722  -0.3722   0.4960   2.3256
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6350     0.1836 -14.355  < 2e-16 ***
## Giver15       1.9182     0.2082   9.214  < 2e-16 ***
## Giver14       0.7028     0.2386   2.945 0.003229 **
## Giver13       1.1332     0.2426   4.670 3.01e-06 ***
## Giver12       0.9140     0.2375   3.849 0.000119 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1090.39  on 802  degrees of freedom
## Residual deviance:  648.12  on 798  degrees of freedom
## AIC: 658.12
##
## Number of Fisher Scoring iterations: 5
```

```r
Anova(model.fit1b, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Giver16
##         LR Chisq Df Pr(>Chisq)
## Giver15   85.917  1  < 2.2e-16 ***
```

```
## Giver14    8.447  1  0.0036574 **
## Giver13   21.394  1   3.74e-06 ***
## Giver12   14.434  1  0.0001452 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = model.fit1b)
CMTe = PredAcc(dt_test, mod = model.fit1b)
ac_table_1 = rbind(ac_table_1, Accuracy(CMTr, CMTe, "model.fit1b"))

# 2016 giving status as function of class year, marital
# status, attendance (not using prior year)
model.fit1c = glm(Giver16 ~ factor(Class.Year) + factor(Marital.Status) +
    AttendenceEvent, data = dt_train, family = binomial(link = "logit"))
summary(model.fit1c)
```

```
##
## Call:
## glm(formula = Giver16 ~ factor(Class.Year) + factor(Marital.Status) +
##     AttendenceEvent, family = binomial(link = "logit"), data = dt_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7716  -0.9658  -0.6676   1.1519   1.8541
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.7659     0.3694  -2.074   0.0381 *
## factor(Class.Year)1982     0.1232     0.2851   0.432   0.6658
## factor(Class.Year)1992    -0.1249     0.2810  -0.444   0.6567
## factor(Class.Year)2002    -0.3114     0.2839  -1.097   0.2728
## factor(Class.Year)2012    -0.1778     0.2864  -0.621   0.5346
## factor(Marital.Status)M    0.2702     0.3170   0.852   0.3940
## factor(Marital.Status)S   -0.4441     0.3471  -1.279   0.2008
## factor(Marital.Status)W    1.1111     0.8846   1.256   0.2091
## AttendenceEvent            0.8675     0.1560   5.559 2.71e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1090.4  on 802  degrees of freedom
## Residual deviance: 1026.4  on 794  degrees of freedom
## AIC: 1044.4
##
## Number of Fisher Scoring iterations: 4
```

```
Anova(model.fit1c, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Giver16
```

```
##                         LR Chisq Df Pr(>Chisq)
## factor(Class.Year)         3.444  4  0.4864875
## factor(Marital.Status)    17.928  3  0.0004551 ***
## AttendenceEvent           32.056  1  1.498e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = model.fit1c)
CMTe = PredAcc(dt_test, mod = model.fit1c)
ac_table_1 = rbind(ac_table_1, Accuracy(CMTr, CMTe, "model.fit1c"))


model.fit1d = glm(Giver16 ~ Giver15 + factor(Class.Year) + factor(Marital.Status) +
    AttendenceEvent, data = dt_train, family = binomial(link = "logit"))
summary(model.fit1d)
```

```
##
## Call:
## glm(formula = Giver16 ~ Giver15 + factor(Class.Year) + factor(Marital.Status) +
##     AttendenceEvent, family = binomial(link = "logit"), data = dt_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2652  -0.6123  -0.5010   0.7224   2.1849
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -2.2744     0.4657  -4.883 1.04e-06 ***
## Giver15                    2.7894     0.1902  14.668  < 2e-16 ***
## factor(Class.Year)1982     0.5619     0.3472   1.619   0.1055
## factor(Class.Year)1992     0.5138     0.3425   1.500   0.1336
## factor(Class.Year)2002     0.3562     0.3481   1.023   0.3061
## factor(Class.Year)2012     0.6348     0.3529   1.799   0.0720 .
## factor(Marital.Status)M    0.1334     0.3860   0.345   0.7297
## factor(Marital.Status)S   -0.3722     0.4221  -0.882   0.3779
## factor(Marital.Status)W    1.0496     1.0426   1.007   0.3141
## AttendenceEvent            0.3592     0.1923   1.868   0.0618 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1090.39  on 802  degrees of freedom
## Residual deviance:  759.98  on 793  degrees of freedom
## AIC: 779.98
##
## Number of Fisher Scoring iterations: 4
```

```
Anova(model.fit1d, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
```

```
## Response: Giver16
##                          LR Chisq Df Pr(>Chisq)
## Giver15                   266.392  1    < 2e-16 ***
## factor(Class.Year)          4.059  4    0.39808
## factor(Marital.Status)      6.418  3    0.09294 .
## AttendenceEvent             3.474  1    0.06234 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = model.fit1d)
CMTe = PredAcc(dt_test, mod = model.fit1d)
ac_table_1 = rbind(ac_table_1, Accuracy(CMTr, CMTe, "model.fit1d"))


# 2016 giving status as function of ALL prior year giving
# statuses , class year, marital status, attendance
model.fit1e = glm(Giver16 ~ Giver15 + Giver14 + Giver13 + Giver12 +
    factor(Class.Year) + factor(Marital.Status) + AttendenceEvent,
    data = dt_train, family = binomial(link = "logit"))
summary(model.fit1e)
```

```
##
## Call:
## glm(formula = Giver16 ~ Giver15 + Giver14 + Giver13 + Giver12 +
##     factor(Class.Year) + factor(Marital.Status) + AttendenceEvent,
##     family = binomial(link = "logit"), data = dt_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1859  -0.4496  -0.3245   0.5211   2.4849
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -3.114085   0.559684  -5.564 2.64e-08 ***
## Giver15                   1.970464   0.216177   9.115  < 2e-16 ***
## Giver14                   0.670106   0.241782   2.772  0.00558 **
## Giver13                   1.147247   0.245217   4.678 2.89e-06 ***
## Giver12                   0.982828   0.244860   4.014 5.97e-05 ***
## factor(Class.Year)1982    0.330461   0.392590   0.842  0.39993
## factor(Class.Year)1992    0.417581   0.388718   1.074  0.28271
## factor(Class.Year)2002    0.196252   0.392533   0.500  0.61710
## factor(Class.Year)2012    0.797856   0.393666   2.027  0.04269 *
## factor(Marital.Status)M   0.073435   0.460772   0.159  0.87337
## factor(Marital.Status)S  -0.163348   0.499486  -0.327  0.74364
## factor(Marital.Status)W   1.274355   1.383990   0.921  0.35716
## AttendenceEvent           0.001879   0.216267   0.009  0.99307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1090.39  on 802  degrees of freedom
## Residual deviance:  641.36  on 790  degrees of freedom
## AIC: 667.36
##
## Number of Fisher Scoring iterations: 5
```

```
Anova(model.fit1e, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Giver16
##                          LR Chisq Df Pr(>Chisq)
## Giver15                    85.542  1  < 2.2e-16 ***
## Giver14                     7.502  1   0.006164 **
## Giver13                    21.523  1  3.496e-06 ***
## Giver12                    15.798  1  7.048e-05 ***
## factor(Class.Year)          5.992  4   0.199734
## factor(Marital.Status)      1.919  3   0.589471
## AttendenceEvent             0.000  1   0.993068
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = model.fit1e)
CMTe = PredAcc(dt_test, mod = model.fit1e)
ac_table_1 = rbind(ac_table_1, Accuracy(CMTr, CMTe, "model.fit1e"))
```

```
# models for determining category - using 2016giving category
# as function of 2015 giving category

library(nnet)
model.fit2a = multinom(FY16GivingCat ~ FY15GivingCat, data = dt_train,
    model = TRUE)
```

```
## # weights:  30 (20 variable)
## initial  value 1292.378644
## iter  10 value 628.527290
## iter  20 value 610.450429
## iter  30 value 610.110694
## iter  40 value 610.082208
## final  value 610.082162
## converged
```

```
summary(model.fit2a)
```

```
## Call:
## multinom(formula = FY16GivingCat ~ FY15GivingCat, data = dt_train,
##     model = TRUE)
##
## Coefficients:
##             (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)       -2.097511            2.6365178               1.047727
## [100,250)     -2.972974            2.2165905               4.166814
## [250,500)     -5.275642            1.3839617               4.477127
```

```
## [500,2e+05)   -4.582518           0.6907751                2.279924
##             FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                -10.452451               -11.683441
## [100,250)                3.377992                 2.125593
## [250,500)                7.472537                 4.022883
## [500,2e+05)              4.987631                 6.274179
##
## Std. Errors:
##             (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)       0.1529410            0.2360163              0.4650086
## [100,250)     0.2292540            0.3412421              0.3430843
## [250,500)     0.7089411            1.2340437              0.8146786
## [500,2e+05)   0.5025768            1.1282344              0.8958574
##             FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)               375.4966130              371.4869325
## [100,250)               0.9411553                0.7271642
## [250,500)               1.0285882                1.0702534
## [500,2e+05)             1.0420018                0.6494259
##
## Residual Deviance: 1220.164
## AIC: 1260.164
```

```
Anova(model.fit2a, test = "LR")
```

```
## # weights:  10 (4 variable)
## initial  value 1292.378644
## iter  10 value 957.836550
## final  value 957.835872
## converged

## Analysis of Deviance Table (Type II tests)
##
## Response: FY16GivingCat
##               LR Chisq Df Pr(>Chisq)
## FY15GivingCat   695.51 16  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = model.fit2a)
CMTe = PredAcc(dt_test, mod = model.fit2a)
ac_table_2 = Accuracy(CMTr, CMTe, "model.fit2a")


# 2015 giving category
library(nnet)

model.fit2b = multinom(FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
    FY13GivingCat + FY12GivingCat, data = dt_train, model = TRUE)
```

```
## # weights:  90 (68 variable)
## initial  value 1292.378644
## iter  10 value 581.833233
```

```
## iter  20 value 500.708030
## iter  30 value 493.829000
## iter  40 value 493.432783
## iter  50 value 493.340614
## iter  60 value 493.333505
## final  value 493.333350
## converged
```

```r
summary(model.fit2b)
```

```
## Call:
## multinom(formula = FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
##     FY13GivingCat + FY12GivingCat, data = dt_train, model = TRUE)
##
## Coefficients:
##              (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)        -2.971427             1.727297              0.7007431
## [100,250)      -4.152544             1.689409              3.0716558
## [250,500)      -8.432707             1.580145              3.7950245
## [500,2e+05)    -5.475330             0.936181              1.7579919
##              FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                  -14.066810               -10.748606
## [100,250)                  1.237906                 1.427913
## [250,500)                  6.229820                 4.522624
## [500,2e+05)                3.910959                 4.974061
##              FY14GivingCat[1,100) FY14GivingCat[100,250)
## [1,100)                 0.6909754            -0.08158044
## [100,250)               0.9494968             0.82347556
## [250,500)               1.9833417             0.94540719
## [500,2e+05)            -0.6527122            -1.26405416
##              FY14GivingCat[250,500) FY14GivingCat[500,2e+05)
## [1,100)                    1.832585               -3.3274244
## [100,250)                  3.605471               -0.4222616
## [250,500)                  4.720164              -19.9834637
## [500,2e+05)                2.521549               -0.7579473
##              FY13GivingCat[1,100) FY13GivingCat[100,250)
## [1,100)                 1.6256886              0.2293143
## [100,250)               0.1074959              1.1853643
## [250,500)             -11.3036455              0.2837161
## [500,2e+05)             0.8386664              1.1285736
##              FY13GivingCat[250,500) FY13GivingCat[500,2e+05)
## [1,100)                   0.2091837               -14.906599
## [100,250)                 1.6590300                 3.212279
## [250,500)                -0.3211780                 5.966285
## [500,2e+05)               0.5042796                 2.207961
##              FY12GivingCat[1,100) FY12GivingCat[100,250)
## [1,100)                 0.5108151               1.233003
## [100,250)               1.3625551               1.186103
## [250,500)               1.6635931               3.824028
## [500,2e+05)            -0.0399828               2.065528
##              FY12GivingCat[250,500) FY12GivingCat[500,2e+05)
```

```
## [1,100)              -15.34353279                  2.3438895
## [100,250)             -0.04013976                  0.1196071
## [250,500)              3.71920666                  0.5300824
## [500,2e+05)            2.57907659                  2.2706190
##
## Std. Errors:
##              (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)        0.2218769            0.2790877              0.5309738
## [100,250)      0.3453842            0.4149802              0.4116148
## [250,500)      1.4910565            1.5301791              1.0706033
## [500,2e+05)    0.6841367            1.2315796              0.9897120
##              FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                8.901798e-06              281.0859517
## [100,250)              1.121374e+00                0.9944612
## [250,500)              1.378034e+00                1.5689001
## [500,2e+05)            1.286613e+00                0.9372661
##              FY14GivingCat[1,100) FY14GivingCat[100,250)
## [1,100)                 0.3062676              0.6575810
## [100,250)               0.4397771              0.5080316
## [250,500)               1.3063443              1.0811126
## [500,2e+05)             1.3669321              1.1110755
##              FY14GivingCat[250,500) FY14GivingCat[500,2e+05)
## [1,100)                    1.569293             1.560618e+01
## [100,250)                  1.065142             1.257847e+00
## [250,500)                  1.400125             6.247240e-06
## [500,2e+05)                1.359429             1.179007e+00
##              FY13GivingCat[1,100) FY13GivingCat[100,250)
## [1,100)                 0.3081613              0.6112423
## [100,250)               0.4631776              0.5056919
## [250,500)             176.8673366              1.0383871
## [500,2e+05)             1.1490460              1.1283302
##              FY13GivingCat[250,500) FY13GivingCat[500,2e+05)
## [1,100)                   1.3171287              0.009095549
## [100,250)                 0.7896443              1.194635052
## [250,500)                 1.2026436              1.888111427
## [500,2e+05)               1.0910223              1.194922704
##              FY12GivingCat[1,100) FY12GivingCat[100,250)
## [1,100)                 0.3143358              0.5283894
## [100,250)               0.4389428              0.5010721
## [250,500)               1.4138353              1.1506962
## [500,2e+05)             1.4153830              0.9172723
##              FY12GivingCat[250,500) FY12GivingCat[500,2e+05)
## [1,100)                2.384639e-06                 1.709330
## [100,250)              8.735333e-01                 1.219065
## [250,500)              1.318004e+00                 2.145853
## [500,2e+05)            1.124498e+00                 1.248572
##
## Residual Deviance: 986.6667
## AIC: 1122.667
```

```r
Anova(model.fit2b, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: FY16GivingCat
##               LR Chisq Df Pr(>Chisq)
## FY15GivingCat  176.683 16  < 2.2e-16 ***
## FY14GivingCat   39.367 16  0.0009625 ***
## FY13GivingCat   68.252 16  2.016e-08 ***
## FY12GivingCat   43.045 16  0.0002751 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
CMTr = PredAcc(dt_train, mod = model.fit2b)
CMTe = PredAcc(dt_test, mod = model.fit2b)
ac_table_2 = rbind(ac_table_2, Accuracy(CMTr, CMTe, "model.fit2b"))
```

```r
# RB's models
rb.fit_kitchensink = multinom(FY16GivingCat ~ FY15GivingCat +
    FY14GivingCat + FY13GivingCat + FY12GivingCat + NextDegCat +
    AttendenceEvent + ClassYearCat + Gender + MajorCat, data = dt_train,
    model = TRUE)
```

```
## # weights:  140 (108 variable)
## initial  value 1292.378644
## iter  10 value 537.475642
## iter  20 value 472.484855
## iter  30 value 464.415371
## iter  40 value 462.096257
## iter  50 value 461.807659
## iter  60 value 461.700170
## iter  70 value 461.674643
## iter  80 value 461.657826
## iter  90 value 461.656184
## final  value 461.655964
## converged
```

```r
summary(rb.fit_kitchensink)
```

```
## Warning in sqrt(diag(vc)): NaNs produced

## Call:
## multinom(formula = FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
##     FY13GivingCat + FY12GivingCat + NextDegCat + AttendenceEvent +
##     ClassYearCat + Gender + MajorCat, data = dt_train, model = TRUE)
##
## Coefficients:
##             (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)       -3.942333            1.8034135              0.9215794
## [100,250)     -4.955095            1.6874921              3.1606376
## [250,500)    -10.377954            1.5188590              4.2465520
## [500,2e+05)   -9.612819            0.3667759              1.1208842
```

```
##             FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                 -28.222527               -10.824193
## [100,250)                 1.057528                 1.334485
## [250,500)                 6.917564                 5.065505
## [500,2e+05)               3.618132                 5.643227
##             FY14GivingCat[1,100) FY14GivingCat[100,250)
## [1,100)                0.6303379             -0.2881516
## [100,250)              0.9698625              0.8332672
## [250,500)              2.4312928              0.7196738
## [500,2e+05)           -0.4533744             -1.5438861
##             FY14GivingCat[250,500) FY14GivingCat[500,2e+05)
## [1,100)                   2.096020               -2.8799130
## [100,250)                 3.662775               -0.4734393
## [250,500)                 4.918631              -45.4157783
## [500,2e+05)               3.123875               -0.5334989
##             FY13GivingCat[1,100) FY13GivingCat[100,250)
## [1,100)               1.59761910              0.3393629
## [100,250)            -0.01742333              1.2243711
## [250,500)           -89.56833131              0.4949588
## [500,2e+05)           0.46765511              1.2403770
##             FY13GivingCat[250,500) FY13GivingCat[500,2e+05)
## [1,100)                  0.4123327              -28.5424515
## [100,250)                1.8308583                2.7813409
## [250,500)               -0.1389894                5.0137792
## [500,2e+05)              0.4361399                0.7652379
##             FY12GivingCat[1,100) FY12GivingCat[100,250)
## [1,100)                0.6030819               1.563368
## [100,250)              1.5754518               1.165908
## [250,500)              0.9315528               4.084716
## [500,2e+05)            0.3572462               3.031716
##             FY12GivingCat[250,500) FY12GivingCat[500,2e+05)  NextDegCat
## [1,100)               -30.77199484                 2.569124  0.67765233
## [100,250)              -0.09626078                 0.619910 -0.04336720
## [250,500)               4.14751331                 1.896554  0.02596288
## [500,2e+05)             3.53551167                 3.908415  1.77089316
##             AttendenceEvent ClassYearCat1982 ClassYearCat1992
## [1,100)          -0.4126530        0.3467598       0.46256126
## [100,250)         0.1894901        0.2831859       0.36654301
## [250,500)         0.7706499        0.3918237      -0.03586090
## [500,2e+05)       1.2270459       -0.6315098      -0.03861407
##             ClassYearCat2002 ClassYearCat2012    GenderM MajorCatOTHER
## [1,100)            0.5652301        1.2220019 -0.3893813    0.46955412
## [100,250)         -0.3209208        0.1178789  0.8379704    0.08124084
## [250,500)         -0.8718146        1.5521493  0.2900443   -0.83410193
## [500,2e+05)        0.8621056        1.1229878  1.3100549    2.00375263
##             MajorCatSOCIAL_SCIENCE MajorCatSTEM
## [1,100)                 0.04543533    0.2551761
## [100,250)              -0.13559512    0.3066910
## [250,500)               0.94668307    0.9607935
## [500,2e+05)            -0.01216357    0.1214892
```

```
##
## Std. Errors:
##             (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)       0.6088922            0.2909747              0.5557337
## [100,250)     0.6647895            0.4369791              0.4295670
## [250,500)     1.9720147            1.5667897              1.1626239
## [500,2e+05)   1.7159281            1.3087350              1.0861523
##             FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)               3.946840e-13             9.443554e-06
## [100,250)             1.131109e+00             1.075671e+00
## [250,500)             1.465049e+00             1.813491e+00
## [500,2e+05)           1.436638e+00             1.143366e+00
##             FY14GivingCat[1,100) FY14GivingCat[100,250)
## [1,100)                0.3174076              0.6903807
## [100,250)              0.4650396              0.5228836
## [250,500)              1.4255352              1.1188271
## [500,2e+05)            1.4110692              1.3025085
##             FY14GivingCat[250,500) FY14GivingCat[500,2e+05)
## [1,100)                   1.987539             5.610019e-06
## [100,250)                 1.074671             1.333990e+00
## [250,500)                 1.503379             2.299834e-15
## [500,2e+05)               1.594366             1.315912e+00
##             FY13GivingCat[1,100) FY13GivingCat[100,250)
## [1,100)                0.3152886              0.6390005
## [100,250)              0.4918003              0.5295733
## [250,500)                    NaN              1.1181451
## [500,2e+05)            1.3314140              1.3466236
##             FY13GivingCat[250,500) FY13GivingCat[500,2e+05)
## [1,100)                  1.3446154             3.272246e-12
## [100,250)                0.8097202             1.235600e+00
## [250,500)                1.2823580             2.033292e+00
## [500,2e+05)              1.2582051             1.272370e+00
##             FY12GivingCat[1,100) FY12GivingCat[100,250)
## [1,100)                0.3272795              0.5552139
## [100,250)              0.4629311              0.5261032
## [250,500)              1.6354136              1.2755839
## [500,2e+05)            1.4161995              1.1569377
##             FY12GivingCat[250,500) FY12GivingCat[500,2e+05) NextDegCat
## [1,100)               9.382352e-14                 2.057870  0.2794246
## [100,250)             8.953853e-01                 1.269874  0.3301071
## [250,500)             1.457227e+00                 2.340959  0.7237902
## [500,2e+05)           1.254080e+00                 1.476830  0.7614794
##             AttendenceEvent ClassYearCat1982 ClassYearCat1992
## [1,100)           0.2690828        0.6052970        0.5835419
## [100,250)         0.3343556        0.5472857        0.5384938
## [250,500)         0.8295951        0.9959716        1.0722727
## [500,2e+05)       0.8583585        1.0243404        0.9924595
##             ClassYearCat2002 ClassYearCat2012   GenderM MajorCatOTHER
## [1,100)            0.5719020        0.5473815 0.2617033     0.4582202
## [100,250)          0.5654286        0.5517447 0.3197624     0.5896076
```

```
## [250,500)          1.2250619        1.2905090 0.6843104     1.3207322
## [500,2e+05)        1.0384912        1.2966783 0.6717860     0.9344656
##             MajorCatSOCIAL_SCIENCE MajorCatSTEM
## [1,100)                  0.3229562    0.3249854
## [100,250)                0.3897479    0.3947299
## [250,500)                0.8577102    0.7988629
## [500,2e+05)              0.8258101    0.8752794
##
## Residual Deviance: 923.3119
## AIC: 1139.312
```

```
Anova(rb.fit_kitchensink, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: FY16GivingCat
##                 LR Chisq Df Pr(>Chisq)
## FY15GivingCat    177.124 16  < 2.2e-16 ***
## FY14GivingCat     39.798 16  0.0008332 ***
## FY13GivingCat     63.183 16  1.510e-07 ***
## FY12GivingCat     51.167 16  1.492e-05 ***
## NextDegCat        13.045  4  0.0110571 *
## AttendenceEvent    6.030  4  0.1968901
## ClassYearCat      19.997 16  0.2203728
## Gender            15.680  4  0.0034797 **
## MajorCat          12.153 12  0.4334835
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = rb.fit_kitchensink)
CMTe = PredAcc(dt_test, mod = rb.fit_kitchensink)
ac_table_2 = rbind(ac_table_2, Accuracy(CMTr, CMTe, "rb.fit_kitchensink"))


# Let's simplify this model - substituting yearsgiven instead
# of each year's category
rb.fit1b = multinom(FY16GivingCat ~ FY15GivingCat + YearsGiven +
    NextDegCat + AttendenceEvent + Gender, data = dt_train, model = TRUE)
```

```
## # weights:  50 (36 variable)
## initial  value 1292.378644
## iter  10 value 622.987012
## iter  20 value 536.805592
## iter  30 value 533.570208
## iter  40 value 533.423878
## iter  50 value 533.413497
## iter  60 value 533.410614
## final  value 533.410561
## converged
```

```
summary(rb.fit1b)
```

```
## Call:
```

```
## multinom(formula = FY16GivingCat ~ FY15GivingCat + YearsGiven +
##     NextDegCat + AttendenceEvent + Gender, data = dt_train, model = TRUE)
##
## Coefficients:
##             (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)       -2.899670            1.1503053             -0.6610199
## [100,250)     -4.723808            0.1617613              1.9916782
## [250,500)     -8.036691           -1.0979918              1.8841691
## [500,2e+05)   -8.049962           -1.2911883              0.2070200
##             FY15GivingCat[250,500) FY15GivingCat[500,2e+05) YearsGiven
## [1,100)                -64.8127451              -14.1313077  0.7839083
## [100,250)                0.7191287               -0.2878368  1.0614274
## [250,500)                4.4825591                1.2358686  1.2530806
## [500,2e+05)              2.7189126                4.2618505  0.8897569
##             NextDegCat AttendenceEvent    GenderM
## [1,100)     0.57779415      -0.2846865 -0.3934712
## [100,250)   0.04765656       0.1611762  0.7822443
## [250,500)   0.67824508       0.8611534  0.2409877
## [500,2e+05) 1.91351237       1.2010310  0.4522668
##
## Std. Errors:
##             (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)       0.2946682            0.3051480              0.5264938
## [100,250)     0.4488504            0.4351472              0.4209627
## [250,500)     1.2680526            1.3632415              0.9941446
## [500,2e+05)   1.1557559            1.2265404              1.0241310
##             FY15GivingCat[250,500) FY15GivingCat[500,2e+05) YearsGiven
## [1,100)               2.880355e-14             1.561555e-06  0.1063332
## [100,250)             1.028340e+00             8.212577e-01  0.1434758
## [250,500)             1.195076e+00             1.266668e+00  0.3716651
## [500,2e+05)           1.202572e+00             7.866910e-01  0.2654538
##             NextDegCat AttendenceEvent   GenderM
## [1,100)      0.2578242       0.2487498 0.2410849
## [100,250)    0.3006220       0.3052567 0.2890722
## [250,500)    0.6000244       0.6670510 0.5151204
## [500,2e+05)  0.6707902       0.6786679 0.5168277
##
## Residual Deviance: 1066.821
## AIC: 1138.821
```

```r
Anova(rb.fit1b, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: FY16GivingCat
##                 LR Chisq Df Pr(>Chisq)
## FY15GivingCat     350.25 16  < 2.2e-16 ***
## YearsGiven        110.13  4  < 2.2e-16 ***
## NextDegCat         14.83  4   0.005074 **
## AttendenceEvent     6.79  4   0.147335
## Gender             15.27  4   0.004176 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
CMTr = PredAcc(dt_train, mod = rb.fit1b)
CMTe = PredAcc(dt_test, mod = rb.fit1b)
ac_table_2 = rbind(ac_table_2, Accuracy(CMTr, CMTe, "rb.fit1b"))
```

```
# Below are the class of models without using prior year
# giving info
rb.fit3a = multinom(FY16GivingCat ~ NextDegCat + AttendenceEvent +
    Gender + ClassYearCat, data = dt_train, model = TRUE)
```

```
## # weights:  45 (32 variable)
## initial  value 1292.378644
## iter  10 value 884.888820
## iter  20 value 868.103976
## iter  30 value 867.312497
## final  value 867.306510
## converged
```

```
summary(rb.fit3a)
```

```
## Call:
## multinom(formula = FY16GivingCat ~ NextDegCat + AttendenceEvent +
##     Gender + ClassYearCat, data = dt_train, model = TRUE)
##
## Coefficients:
##              (Intercept) NextDegCat AttendenceEvent    GenderM
## [1,100)        -2.058358  0.9877683       0.2717782 -0.4540528
## [100,250)      -2.001637  0.4467707       0.8981246  0.5070502
## [250,500)      -3.504867  0.4056550       2.0847733  0.0665231
## [500,2e+05)    -3.730944  1.4937736       1.9900994  0.1961129
##              ClassYearCat1982 ClassYearCat1992 ClassYearCat2002
## [1,100)           -0.12315757       -0.1277318        0.2662751
## [100,250)          0.06030921       -0.2814611       -0.8015424
## [250,500)         -0.16517626       -0.7054081       -1.7222901
## [500,2e+05)       -0.60000056       -1.0949571       -1.2506622
##              ClassYearCat2012
## [1,100)             0.6166449
## [100,250)          -1.2230128
## [250,500)          -2.2285012
## [500,2e+05)        -3.1970347
##
## Std. Errors:
##              (Intercept) NextDegCat AttendenceEvent   GenderM
## [1,100)        0.4354032  0.2230367       0.2069830 0.2046068
## [100,250)      0.3777324  0.2338003       0.2319992 0.2216295
## [250,500)      0.6944899  0.4198344       0.5495730 0.3814650
## [500,2e+05)    0.6382202  0.4415967       0.4574431 0.3275147
##              ClassYearCat1982 ClassYearCat1992 ClassYearCat2002
## [1,100)             0.4828681        0.4641621        0.4428099
## [100,250)           0.3724192        0.3708442        0.3879664
```

```
## [250,500)          0.5602863        0.5793250        0.6720005
## [500,2e+05)        0.4883740        0.4992684        0.4987449
##             ClassYearCat2012
## [1,100)            0.4214478
## [100,250)          0.3883916
## [250,500)          0.7244754
## [500,2e+05)        0.8057314
##
## Residual Deviance: 1734.613
## AIC: 1798.613
```

```r
confint(rb.fit3a)
```

```
## , , [1,100)
##
##                       2.5 %      97.5 %
## (Intercept)      -2.9117329 -1.20498378
## NextDegCat        0.5506245  1.42491213
## AttendenceEvent  -0.1339010  0.67745739
## GenderM          -0.8550747 -0.05303082
## ClassYearCat1982 -1.0695617  0.82324660
## ClassYearCat1992 -1.0374728  0.78200912
## ClassYearCat2002 -0.6016164  1.13416657
## ClassYearCat2012 -0.2093777  1.44266740
##
## , , [100,250)
##
##                        2.5 %      97.5 %
## (Intercept)      -2.74197904 -1.26129520
## NextDegCat       -0.01146950  0.90501090
## AttendenceEvent   0.44341442  1.35283469
## GenderM           0.07266445  0.94143602
## ClassYearCat1982 -0.66961902  0.79023743
## ClassYearCat1992 -1.00830228  0.44538014
## ClassYearCat2002 -1.56194268 -0.04114221
## ClassYearCat2012 -1.98424633 -0.46177921
##
## , , [250,500)
##
##                       2.5 %     97.5 %
## (Intercept)      -4.8660417 -2.1436914
## NextDegCat       -0.4172053  1.2285153
## AttendenceEvent   1.0076300  3.1619165
## GenderM          -0.6811346  0.8141808
## ClassYearCat1982 -1.2633173  0.9329647
## ClassYearCat1992 -1.8408642  0.4300481
## ClassYearCat2002 -3.0393869 -0.4051934
## ClassYearCat2012 -3.6484469 -0.8085556
##
## , , [500,2e+05)
##
```

```
##                         2.5 %     97.5 %
## (Intercept)      -4.9818324 -2.4800553
## NextDegCat        0.6282599  2.3592872
## AttendenceEvent   1.0935274  2.8866714
## GenderM          -0.4458041  0.8380299
## ClassYearCat1982 -1.5571960  0.3571949
## ClassYearCat1992 -2.0735052 -0.1164090
## ClassYearCat2002 -2.2281842 -0.2731403
## ClassYearCat2012 -4.7762392 -1.6178302
```

```r
Anova(rb.fit1b, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: FY16GivingCat
##                  LR Chisq Df Pr(>Chisq)
## FY15GivingCat     350.25 16  < 2.2e-16 ***
## YearsGiven        110.13  4  < 2.2e-16 ***
## NextDegCat         14.83  4   0.005074 **
## AttendenceEvent     6.79  4   0.147335
## Gender             15.27  4   0.004176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
CMTr = PredAcc(dt_train, mod = rb.fit3a)
```

```
## Warning in confusionMatrix.default(results[, 1], results[, 2], positive =
## "1"): Levels are not in the same order for reference and data. Refactoring
## data to match.
```

```r
CMTe = PredAcc(dt_test, mod = rb.fit3a)
```

```
## Warning in confusionMatrix.default(results[, 1], results[, 2], positive =
## "1"): Levels are not in the same order for reference and data. Refactoring
## data to match.
```

```r
ac_table_3 = Accuracy(CMTr, CMTe, "rb.fit3a")
```

```r
# using full data set mostly replicating previous work
mod.multinom.1 <- multinom(FY16GivingCat ~ FY15GivingCat, data = dt,
    model = TRUE)
```

```
## # weights:  30 (20 variable)
## initial  value 1609.437912
## iter  10 value 835.934583
## iter  20 value 750.834554
## iter  30 value 747.736474
## iter  40 value 747.698758
## final  value 747.697882
## converged
```

```r
summary(mod.multinom.1)
```

```
## Call:
## multinom(formula = FY16GivingCat ~ FY15GivingCat, data = dt,
```

```
##     model = TRUE)
##
## Coefficients:
##              (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)        -2.130545            2.6537692              0.8412136
## [100,250)      -3.037929            2.0976080              4.1474213
## [250,500)      -5.074690            0.9150179              4.0087745
## [500,2e+05)    -4.787675            0.6280529              2.5186060
##              FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                   -9.727135                -8.237404
## [100,250)                  2.815656                 2.056782
## [250,500)                  6.601285                 3.687989
## [500,2e+05)                4.564640                 6.558276
##
## Std. Errors:
##              (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)        0.1400877            0.2109681              0.4233289
## [100,250)      0.2134184            0.3180905              0.3022841
## [250,500)      0.5790227            1.1625922              0.6854190
## [500,2e+05)    0.5021335            1.1262510              0.7873293
##              FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                 168.0652366               63.0682924
## [100,250)                 0.7039418                0.7098855
## [250,500)                 0.7608105                0.9800018
## [500,2e+05)               0.8380425                0.6311903
##
## Residual Deviance: 1495.396
## AIC: 1535.396
```

```r
# also replicating previous stuff
mod.multinom.2 <- multinom(FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
    FY13GivingCat + FY12GivingCat, data = dt, model = TRUE)
```

```
## # weights:  90 (68 variable)
## initial  value 1609.437912
## iter  10 value 694.983744
## iter  20 value 619.445431
## iter  30 value 612.557881
## iter  40 value 612.054816
## iter  50 value 611.847656
## iter  60 value 611.834283
## final  value 611.833835
## converged
```

```r
summary(mod.multinom.2)
```

```
## Call:
## multinom(formula = FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
##     FY13GivingCat + FY12GivingCat, data = dt, model = TRUE)
##
## Coefficients:
##              (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
```

```
## [1,100)       -3.018675           1.6530376              0.3604355
## [100,250)     -4.261052           1.4752362              2.7771538
## [250,500)     -6.427459           1.0367544              3.0309467
## [500,2e+05)   -5.550641           0.8098713              1.6173013
##              FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)                -14.3514647            -13.3417590
## [100,250)                0.6007561              0.7644009
## [250,500)                4.4580466              3.0335076
## [500,2e+05)              2.5473921              4.8113480
##              FY14GivingCat[1,100) FY14GivingCat[100,250)
## [1,100)               0.7147593             0.3248955
## [100,250)             0.9000026             1.2785078
## [250,500)             1.3153036             1.0912907
## [500,2e+05)          -0.9901099            -0.2677729
##              FY14GivingCat[250,500) FY14GivingCat[500,2e+05)
## [1,100)                   1.200101            -11.46675299
## [100,250)                 2.440186              0.43459787
## [250,500)                 2.734109            -23.92331785
## [500,2e+05)               1.949459              0.08900009
##              FY13GivingCat[1,100) FY13GivingCat[100,250)
## [1,100)               1.5985400             0.1376032
## [100,250)             0.5531868             1.4067940
## [250,500)           -24.9527721             0.4612445
## [500,2e+05)           0.4846545             1.2496012
##              FY13GivingCat[250,500) FY13GivingCat[500,2e+05)
## [1,100)                  0.4083779              -15.361336
## [100,250)                2.2191177                2.964382
## [250,500)                0.8049057                4.995048
## [500,2e+05)              0.8995235                1.591329
##              FY12GivingCat[1,100) FY12GivingCat[100,250)
## [1,100)               0.7133063             0.9615675
## [100,250)             1.2026939             0.7858424
## [250,500)             0.5493361             1.7386084
## [500,2e+05)           0.9925009             1.2967531
##              FY12GivingCat[250,500) FY12GivingCat[500,2e+05)
## [1,100)               -14.32494653              2.7145730
## [100,250)              -0.08290131              0.3485179
## [250,500)               2.45722996              0.3254234
## [500,2e+05)             2.11979353              2.5310318
##
## Std. Errors:
##              (Intercept) FY15GivingCat[1,100) FY15GivingCat[100,250)
## [1,100)        0.2020677           0.2533310              0.4806549
## [100,250)      0.3239547           0.3858052              0.3563347
## [250,500)      0.8713018           1.3264635              0.8329934
## [500,2e+05)    0.6418850           1.2041332              0.8707067
##              FY15GivingCat[250,500) FY15GivingCat[500,2e+05)
## [1,100)               5.500878e-07           1.312589e-06
## [100,250)             8.474527e-01           9.311261e-01
## [250,500)             9.534882e-01           1.266192e+00
```

```
## [500,2e+05)          1.006574e+00            8.139214e-01
##              FY14GivingCat[1,100) FY14GivingCat[100,250)
## [1,100)               0.2691675              0.5700144
## [100,250)             0.3958019              0.4469954
## [250,500)             1.0639646              0.8246949
## [500,2e+05)           1.3447815              0.9208284
##              FY14GivingCat[250,500) FY14GivingCat[500,2e+05)
## [1,100)                 1.4978587             3.120067e-06
## [100,250)               0.8996351             1.063986e+00
## [250,500)               1.0376402             1.757779e-09
## [500,2e+05)             1.0736973             1.028204e+00
##              FY13GivingCat[1,100) FY13GivingCat[100,250)
## [1,100)            2.690364e-01              0.5531156
## [100,250)          4.079561e-01              0.4501662
## [250,500)          4.842175e-11              0.8388319
## [500,2e+05)        1.085774e+00              0.9603641
##              FY13GivingCat[250,500) FY13GivingCat[500,2e+05)
## [1,100)                 1.2831122             9.173999e-07
## [100,250)               0.7335645             1.115568e+00
## [250,500)               0.9924310             1.620865e+00
## [500,2e+05)             1.0752577             1.123814e+00
##              FY12GivingCat[1,100) FY12GivingCat[100,250)
## [1,100)               0.2740850              0.4896488
## [100,250)             0.3873831              0.4410280
## [250,500)             1.1350712              0.7962112
## [500,2e+05)           1.0988395              0.8599725
##              FY12GivingCat[250,500) FY12GivingCat[500,2e+05)
## [1,100)              8.050819e-07                 1.674871
## [100,250)            8.315494e-01                 1.213899
## [250,500)            9.788799e-01                 1.889281
## [500,2e+05)          1.085112e+00                 1.128253
##
## Residual Deviance: 1223.668
## AIC: 1359.668
```

```r
anova(mod.multinom.1, mod.multinom.2)
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: FY16GivingCat
##                                                             Model Resid. df
## 1                                                   FY15GivingCat      3980
## 2 FY15GivingCat + FY14GivingCat + FY13GivingCat + FY12GivingCat      3932
##   Resid. Dev   Test    Df LR stat. Pr(Chi)
## 1   1495.396
## 2   1223.668 1 vs 2    48 271.7281       0
```

```r
AIC(mod.multinom.1, mod.multinom.2)
```

```
##                df      AIC
## mod.multinom.1 20 1535.396
## mod.multinom.2 68 1359.668
```

```r
BIC(mod.multinom.1, mod.multinom.2)
```

```
##                df      BIC
## mod.multinom.1 20 1633.551
## mod.multinom.2 68 1693.395
```

```r
mod.multinom.gar <- multinom(FY16GivingCat ~ FY15GivingCat +
    MajorCat, data = dt, model = TRUE)
```

```
## # weights:  45 (32 variable)
## initial  value 1609.437912
## iter  10 value 795.163986
## iter  20 value 753.622699
## iter  30 value 742.060934
## iter  40 value 741.855608
## iter  50 value 741.825161
## final  value 741.825045
## converged
```

```r
mod.multinom.3 <- multinom(FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
    FY13GivingCat + FY12GivingCat + Gender + factor(Class.Year) +
    AttendenceEvent + NextDegCat + ClassYearCat + MaritalStatusCat,
    data = dt, model = TRUE)
```

```
## # weights:  160 (124 variable)
## initial  value 1609.437912
## iter  10 value 661.340516
## iter  20 value 596.344554
## iter  30 value 583.640115
## iter  40 value 580.813437
## iter  50 value 580.195668
## iter  60 value 579.996605
## iter  70 value 579.924147
## iter  80 value 579.911864
## final  value 579.911026
## converged
```

```r
anova(mod.multinom.2, mod.multinom.3)
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: FY16GivingCat
##
## 1
## 2 FY15GivingCat + FY14GivingCat + FY13GivingCat + FY12GivingCat + Gender + factor(Class.Year) +
##   Resid. df Resid. Dev   Test    Df LR stat.     Pr(Chi)
## 1      3932   1223.668
## 2      3892   1159.822 1 vs 2    40 63.84562 0.009665263
```

```r
# so adding everything is significant-does it pass AIC test?
AIC(mod.multinom.2, mod.multinom.3)
```

```
##                df      AIC
```

```
## mod.multinom.2  68 1359.668
## mod.multinom.3 108 1375.822
```

```
BIC(mod.multinom.2, mod.multinom.3)
```

```
##                 df      BIC
## mod.multinom.2  68 1693.395
## mod.multinom.3 108 1905.860
```

```
# AIC and BIC are higher for the kitchen sink approach, so
# should choose a more contrainsted model,
```

```
# Changing the yearly categorical variables into ordered
# factors
l1 <- c("FY16GivingCat", "FY15GivingCat", "FY14GivingCat", "FY13GivingCat",
    "FY12GivingCat")



# this is the right way to do it
dt[, `:=`((l1), lapply(.SD, ordered)), .SDcols = l1]

levels(dt$FY15GivingCat)
```

```
## [1] "[0,1)"       "[1,100)"     "[100,250)"   "[250,500)"   "[500,2e+05)"
```

```
min(dt$FY15GivingCat)
```

```
## [1] [0,1)
## Levels: [0,1) < [1,100) < [100,250) < [250,500) < [500,2e+05)
```

```
mod.ord.1 <- clm(FY16GivingCat ~ FY15GivingCat, data = dt)
summary(mod.ord.1)
```

```
## formula: FY16GivingCat ~ FY15GivingCat
## data:    dt
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -844.85 1705.71 6(0)  1.24e-09 4.2e+01
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## FY15GivingCat.L   5.9864     0.3339  17.928   <2e-16 ***
## FY15GivingCat.Q   0.1264     0.2334   0.542    0.588
## FY15GivingCat.C   0.1779     0.2303   0.772    0.440
## FY15GivingCat^4  -0.1115     0.1932  -0.577    0.564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)          -2.0949     0.1541 -13.598
## [1,100)|[100,250)      -0.6854     0.1452  -4.722
## [100,250)|[250,500)     1.5336     0.1532  10.009
```

```
## [250,500)|[500,2e+05)   2.8272     0.2016  14.026
```

```r
Anova(mod.ord.1, type = "II")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: FY16GivingCat
##                Df  Chisq Pr(>Chisq)
## FY15GivingCat  4 404.78  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mod.ord.2 <- clm(FY16GivingCat ~ FY15GivingCat + AttendenceEvent,
    data = dt)
summary(mod.ord.2)
```

```
## formula: FY16GivingCat ~ FY15GivingCat + AttendenceEvent
## data:    dt
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -839.86 1697.73 6(0)  1.28e-09 4.7e+01
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## FY15GivingCat.L  5.88609    0.33537  17.551  < 2e-16 ***
## FY15GivingCat.Q  0.16919    0.23411   0.723  0.46987
## FY15GivingCat.C  0.18451    0.23080   0.799  0.42404
## FY15GivingCat^4 -0.08143    0.19368  -0.420  0.67418
## AttendenceEvent  0.48529    0.15451   3.141  0.00168 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)          -1.7482     0.1895  -9.226
## [1,100)|[100,250)      -0.3294     0.1849  -1.782
## [100,250)|[250,500)     1.9029     0.1951   9.755
## [250,500)|[500,2e+05)   3.2003     0.2357  13.575
```

```r
anova(mod.ord.1, mod.ord.2)
```

```
## Likelihood ratio tests of cumulative link models:
##
##           formula:                                      link: threshold:
## mod.ord.1 FY16GivingCat ~ FY15GivingCat                 logit flexible
## mod.ord.2 FY16GivingCat ~ FY15GivingCat + AttendenceEvent logit flexible
##
##           no.par    AIC  logLik LR.stat df Pr(>Chisq)
## mod.ord.1      8 1705.7 -844.85
## mod.ord.2      9 1697.7 -839.86  9.9832  1    0.00158 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod.ord.2)
```

```
## formula: FY16GivingCat ~ FY15GivingCat + AttendenceEvent
## data:    dt
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -839.86 1697.73 6(0)  1.28e-09 4.7e+01
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## FY15GivingCat.L  5.88609    0.33537  17.551  < 2e-16 ***
## FY15GivingCat.Q  0.16919    0.23411   0.723  0.46987
## FY15GivingCat.C  0.18451    0.23080   0.799  0.42404
## FY15GivingCat^4 -0.08143    0.19368  -0.420  0.67418
## AttendenceEvent  0.48529    0.15451   3.141  0.00168 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                           Estimate Std. Error z value
## [0,1)|[1,100)              -1.7482     0.1895  -9.226
## [1,100)|[100,250)          -0.3294     0.1849  -1.782
## [100,250)|[250,500)         1.9029     0.1951   9.755
## [250,500)|[500,2e+05)       3.2003     0.2357  13.575
```

```r
mod.ord.3 <- clm(FY16GivingCat ~ FY15GivingCat + AttendenceEvent +
    YearsGiven + MajorCat, data = dt)
summary(mod.ord.3)
```

```
## formula:
## FY16GivingCat ~ FY15GivingCat + AttendenceEvent + YearsGiven + MajorCat
## data:    dt
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -770.74 1567.48 6(0)  1.80e-09 7.1e+02
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## FY15GivingCat.L            4.63820    0.35802  12.955  < 2e-16 ***
## FY15GivingCat.Q            1.25962    0.25868   4.869 1.12e-06 ***
## FY15GivingCat.C           -0.26446    0.24184  -1.094   0.2742
## FY15GivingCat^4            0.12912    0.19885   0.649   0.5161
## AttendenceEvent            0.24771    0.16291   1.521   0.1284
## YearsGiven                 0.85633    0.07667  11.169  < 2e-16 ***
## MajorCatOTHER              0.45701    0.27257   1.677   0.0936 .
## MajorCatSOCIAL_SCIENCE -0.02288    0.18415  -0.124   0.9011
## MajorCatSTEM               0.09519    0.19125   0.498   0.6187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
```

```
##                        Estimate Std. Error z value
## [0,1)|[1,100)           0.5319     0.2960   1.797
## [1,100)|[100,250)       2.1430     0.3087   6.942
## [100,250)|[250,500)     4.4797     0.3240  13.824
## [250,500)|[500,2e+05)   5.7879     0.3511  16.486
```

```r
AIC(mod.ord.1, mod.ord.2, mod.ord.3)
```

```
##           df      AIC
## mod.ord.1  8 1705.709
## mod.ord.2  9 1697.725
## mod.ord.3 13 1567.482
```

```r
mod.ord.4 <- clm(FY16GivingCat ~ FY15GivingCat + FY14GivingCat +
    FY13GivingCat + FY12GivingCat + AttendenceEvent + YearsGiven +
    factor(Class.Year), data = dt)
summary(mod.ord.4)
```

```
## formula:
## FY16GivingCat ~ FY15GivingCat + FY14GivingCat + FY13GivingCat + FY12GivingCat + AttendenceEvent
## data:    dt
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -754.24 1558.47 6(0)  3.60e-09 1.7e+02
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## FY15GivingCat.L          3.759714   0.460232   8.169 3.11e-16 ***
## FY15GivingCat.Q          0.231584   0.335246   0.691 0.489700
## FY15GivingCat.C          0.295529   0.340503   0.868 0.385438
## FY15GivingCat^4          0.080724   0.254416   0.317 0.751022
## FY14GivingCat.L          1.350550   0.479213   2.818 0.004828 **
## FY14GivingCat.Q         -0.165754   0.388072  -0.427 0.669291
## FY14GivingCat.C         -0.216612   0.354439  -0.611 0.541107
## FY14GivingCat^4         -0.367712   0.264918  -1.388 0.165130
## FY13GivingCat.L          1.503276   0.447521   3.359 0.000782 ***
## FY13GivingCat.Q         -0.312301   0.395416  -0.790 0.429642
## FY13GivingCat.C          0.279136   0.314499   0.888 0.374779
## FY13GivingCat^4          0.007332   0.223222   0.033 0.973796
## FY12GivingCat.L          1.427224   0.465690   3.065 0.002179 **
## FY12GivingCat.Q          0.230905   0.381927   0.605 0.545458
## FY12GivingCat.C          0.462517   0.333309   1.388 0.165243
## FY12GivingCat^4          0.074133   0.252105   0.294 0.768714
## AttendenceEvent          0.309552   0.169510   1.826 0.067826 .
## YearsGiven                     NA         NA      NA       NA
## factor(Class.Year)1982  0.309830   0.295020   1.050 0.293627
## factor(Class.Year)1992  0.167738   0.288710   0.581 0.561245
## factor(Class.Year)2002  0.009807   0.288357   0.034 0.972869
## factor(Class.Year)2012  0.497635   0.289499   1.719 0.085623 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)          -2.3996     0.3026  -7.930
## [1,100)|[100,250)      -0.7565     0.2971  -2.547
## [100,250)|[250,500)     1.6757     0.3032   5.527
## [250,500)|[500,2e+05)   3.0544     0.3335   9.157
```

```r
mod.ord.5 <- clm(FY16GivingCat ~ Giver15 + Giver14 + Giver13 +
    Giver12 + AttendenceEvent, data = dt)
summary(mod.ord.5)
```

```
## formula:
## FY16GivingCat ~ Giver15 + Giver14 + Giver13 + Giver12 + AttendenceEvent
## data:    dt
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -889.30 1796.60 5(0)  8.69e-09 1.2e+02
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## Giver15           1.7309     0.1766   9.801  < 2e-16 ***
## Giver14           0.7820     0.1918   4.077 4.57e-05 ***
## Giver13           1.0210     0.2015   5.068 4.03e-07 ***
## Giver12           0.9123     0.1901   4.798 1.60e-06 ***
## AttendenceEvent   0.3655     0.1602   2.281   0.0225 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)           2.8205     0.1814   15.55
## [1,100)|[100,250)       4.2312     0.2145   19.72
## [100,250)|[250,500)     5.6679     0.2435   23.28
## [250,500)|[500,2e+05)   6.2957     0.2600   24.22
```

```r
mod.ord.none <- clm(FY16GivingCat ~ Gender + ClassYearCat + Marital.Status +
    AttendenceEvent + NextDegCat + MajorCat, data = dt)
summary(mod.ord.none)
```

```
## formula:
## FY16GivingCat ~ Gender + ClassYearCat + Marital.Status + AttendenceEvent + NextDegCat + MajorCa
## data:    dt
##
##  link  threshold nobs logLik   AIC     niter max.grad cond.H
##  logit flexible  1000 -1107.99 2249.98 6(0)  5.13e-11 4.9e+02
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## GenderM                 0.061582   0.132864   0.463 0.643007
## ClassYearCat1982       -0.237773   0.245848  -0.967 0.333467
## ClassYearCat1992       -0.598405   0.245867  -2.434 0.014939 *
## ClassYearCat2002       -0.918097   0.251528  -3.650 0.000262 ***
```

```
## ClassYearCat2012       -1.028840   0.252171  -4.080 4.50e-05 ***
## Marital.StatusM         0.513475   0.278115   1.846 0.064853 .
## Marital.StatusS        -0.002578   0.307048  -0.008 0.993301
## Marital.StatusW         1.403343   0.670309   2.094 0.036298 *
## AttendenceEvent         1.080591   0.142937   7.560 4.03e-14 ***
## NextDegCat              0.668261   0.145155   4.604 4.15e-06 ***
## MajorCatOTHER           0.252607   0.247434   1.021 0.307298
## MajorCatSOCIAL_SCIENCE  0.028703   0.161334   0.178 0.858795
## MajorCatSTEM            0.116787   0.169443   0.689 0.490673
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)           1.1734     0.3408   3.443
## [1,100)|[100,250)       2.0761     0.3450   6.018
## [100,250)|[250,500)     3.2656     0.3555   9.186
## [250,500)|[500,2e+05)   3.8545     0.3658  10.537
```