lab 2 samir

Samir Datta

October 20, 2017

```
library(ggplot2)
library(ordinal)
```

```
## Warning: package 'ordinal' was built under R version 3.4.2
```

```
library(nnet)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.2
```

```
## Loading required package: lattice
```

```
labdata <- read.csv('C:/Users/Samir/Documents/MIDS/StatsF17/lab 2/lab2data.csv')
```

Feature enginerring

```
#split major into STEM and non-STEM
labdata$MajorType <- ifelse(labdata$Major=='Biology'|
                              labdata$Major=='Economics'|
                              labdata$Major=='Psychology'|
                              labdata$Major=='Physics'|
                              labdata$Major=='Chemistry'|
                              labdata$Major=='Mathematics'|
                              labdata$Major=='General Science-Chemistry'|
                              labdata$Major=='Economics-Business'|
                              labdata$Major=='General Science-Chemistry'|
                              labdata$Major=='Sociology-Anthropology'|
                              labdata$Major=='General Science-Psycho'|
                              labdata$Major=='General Science-Math'|
                              labdata$Major=='General Science-Biology'|
                              labdata$Major=='Computer Science'|
                              labdata$Major=='General Science'|
                              labdata$Major=='Mathematics-Physics'|
                              labdata$Major=='Economics-Regional Stds.'|
                              labdata$Major=='Zoology'|
                              labdata$Major=='Engineering'|
                              labdata$Major=='Sociology'|
                              labdata$Major=='Anthropology'|
                              labdata$Major=='General Science-Physics',
                            "STEM", "Non-STEM")

#create variable nextDegreeType to categorize the most common next degrees
labdata$NextDegreeType <- ifelse(labdata$Next.Degree=='JD', 'JD',
                           ifelse(labdata$Next.Degree=='MA', 'MA',
                           ifelse(labdata$Next.Degree=='PHD', 'PHD',
                          ifelse(labdata$Next.Degree=='NDA', 'NDA',
                                 ifelse(labdata$Next.Degree=='MS', 'MS',
                                      ifelse(labdata$Next.Degree=='MD', 'MD',
                                             ifelse(labdata$Next.Degree=='MBA', 'MBA',
         ifelse(labdata$Next.Degree=='NONE', 'NONE', 'Other'))))))))

#create simpler variable to represent if someone has an advanced degree or not
labdata$NextDegreeBinary <- ifelse(labdata$Next.Degree=='NONE', 0, 1)


#create buckets for all years
labdata$FY16cat <- cut(labdata$FY16Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY15cat <- cut(labdata$FY15Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY14cat <- cut(labdata$FY14Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY13cat <- cut(labdata$FY13Giving, c(0,1,100,250,500,200000), right=F)
labdata$FY12cat <- cut(labdata$FY12Giving, c(0,1,100,250,500,200000), right=F)

#turn class year into years since grad to make interpretaion easier
labdata$YearsSinceGrad <- 2017 - labdata$Class.Year


#loop through data to get each person's mean donation over the past 4 years
#and how many of the past years they've donated
labdata$meandonation <- NA
labdata$nPastYears <- NA
for (i in c(1:1000)){
  labdata[i,]$meandonation<-mean(c(labdata[i,]$FY12Giving,
                                    labdata[i,]$FY13Giving,
                                    labdata[i,]$FY14Giving,
                                    labdata[i,]$FY15Giving))

  labdata[i,]$nPastYears <- sum(c(labdata[i,]$FY12Giving>0,
                                  labdata[i,]$FY13Giving>0,
                                  labdata[i,]$FY14Giving>0,
                                  labdata[i,]$FY15Giving>0))
}

#binary variable - have they donated before or not?
labdata$past_binary <- ifelse(labdata$meandonation == 0,0,1)

#did they donate last year?
labdata$donatedLastYear <- ifelse(labdata$FY15Giving==0, 0, 1)

#how many of the past 4 years, consecutively, did they donate?
#note that this will give a 0 for those that donated 2012-2014 but NOT 2015
#since we're asking for consecutive years
labdata$nPastConsecutiveYears <- ifelse(labdata$FY15Giving==0, 0,
                                        ifelse(labdata$FY14Giving==0,1,
                                               ifelse(labdata$FY13Giving==0,2,
                                                      ifelse(labdata$FY12Giving==0,3,4))))

#did they give in 2015 or not?
labdata$gaveLastYear <- ifelse(labdata$FY15Giving==0,0,1)
```

```
labdata$donatedLastYear <- ifelse(labdata$FY15Giving==0, 0, 1)
```

```r
labdata$nPastConsecutiveYears <- ifelse(labdata$FY15Giving==0, 0,
                                        ifelse(labdata$FY14Giving==0,1,
                                               ifelse(labdata$FY13Giving==0,2,
                                                      ifelse(labdata$FY12Giving==0,3,4))))
labdata$gaveLastYear <- ifelse(labdata$FY15Giving==0,0,1)
```

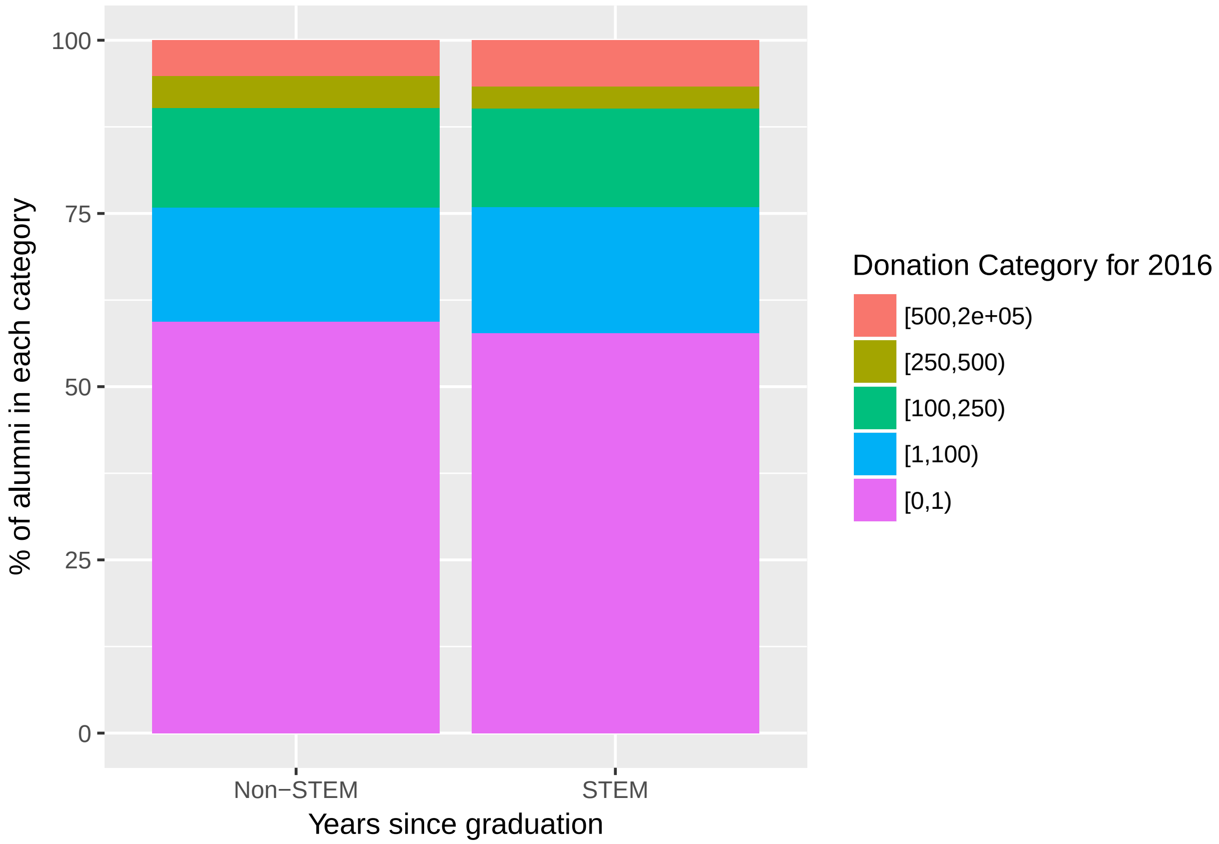
EDA

Major type - STEM vs. Non-STEM

```r
labdata_counts <- with(labdata,
                       aggregate(MajorType,
                                 list(MajorType=MajorType),
                                 length))
labdata_agg <- with(labdata,
                    aggregate(MajorType, list(MajorType=MajorType,
                                            FY16cat=FY16cat),
                    length))
labdata_agg <- merge(labdata_agg, labdata_counts, by="MajorType")
labdata_agg <- setNames(labdata_agg, c("MajorType", "FY16cat", "Count", "TotalCount"))
labdata_agg$percent <- 100*labdata_agg$Count/labdata_agg$TotalCount
labdata_agg$FY16cat <- factor(labdata_agg$FY16cat,
      levels=c("[500,2e+05)","[250,500)","[100,250)","[1,100)","[0,1)"))

ggp <- ggplot(labdata_agg, aes(x=MajorType,y=percent))

ggp + geom_bar(stat="identity", aes(fill=FY16cat))+
  ylab("% of alumni in each category")+xlab("Years since graduation")+
  scale_fill_discrete(name="Donation Category for 2016")
```



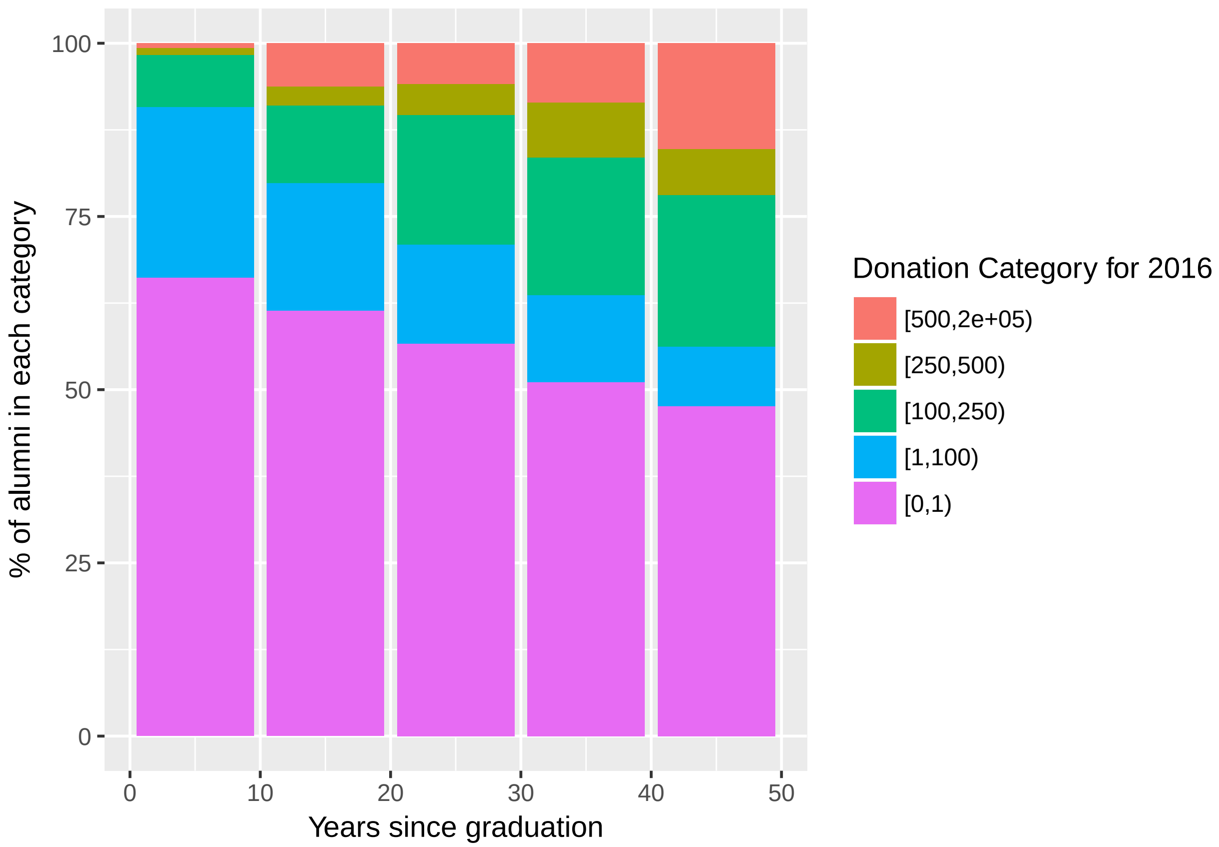
Major type does not seem to have an effect on the donation amount of an alumnnus, as the distribution of donation categories appears virtually identical regardless of whether they graduated with a STEM or non-STEM degree.

Next Degree (binary)

```
labdata_counts <- with(labdata,
                       aggregate(YearsSinceGrad,
                                 list(YearsSinceGrad=YearsSinceGrad),
                                 length))
labdata_agg <- with(labdata,
                    aggregate(YearsSinceGrad, list(YearsSinceGrad=YearsSinceGrad,
                                                   FY16cat=FY16cat),
                    length))
labdata_agg <- merge(labdata_agg, labdata_counts, by="YearsSinceGrad")
labdata_agg <- setNames(labdata_agg, c("YearsSinceGrad", "FY16cat", "Count", "TotalCount"))
labdata_agg$percent <- 100*labdata_agg$Count/labdata_agg$TotalCount
labdata_agg$FY16cat <- factor(labdata_agg$FY16cat,
      levels=c("[500,2e+05)","[250,500)","[100,250)","[1,100)","[0,1)"))

ggp <- ggplot(labdata_agg, aes(x=YearsSinceGrad,y=percent))

ggp + geom_bar(stat="identity", aes(fill=FY16cat))+
  ylab("% of alumni in each category")+xlab("Years since graduation")+
  scale_fill_discrete(name="Donation Category for 2016")
```



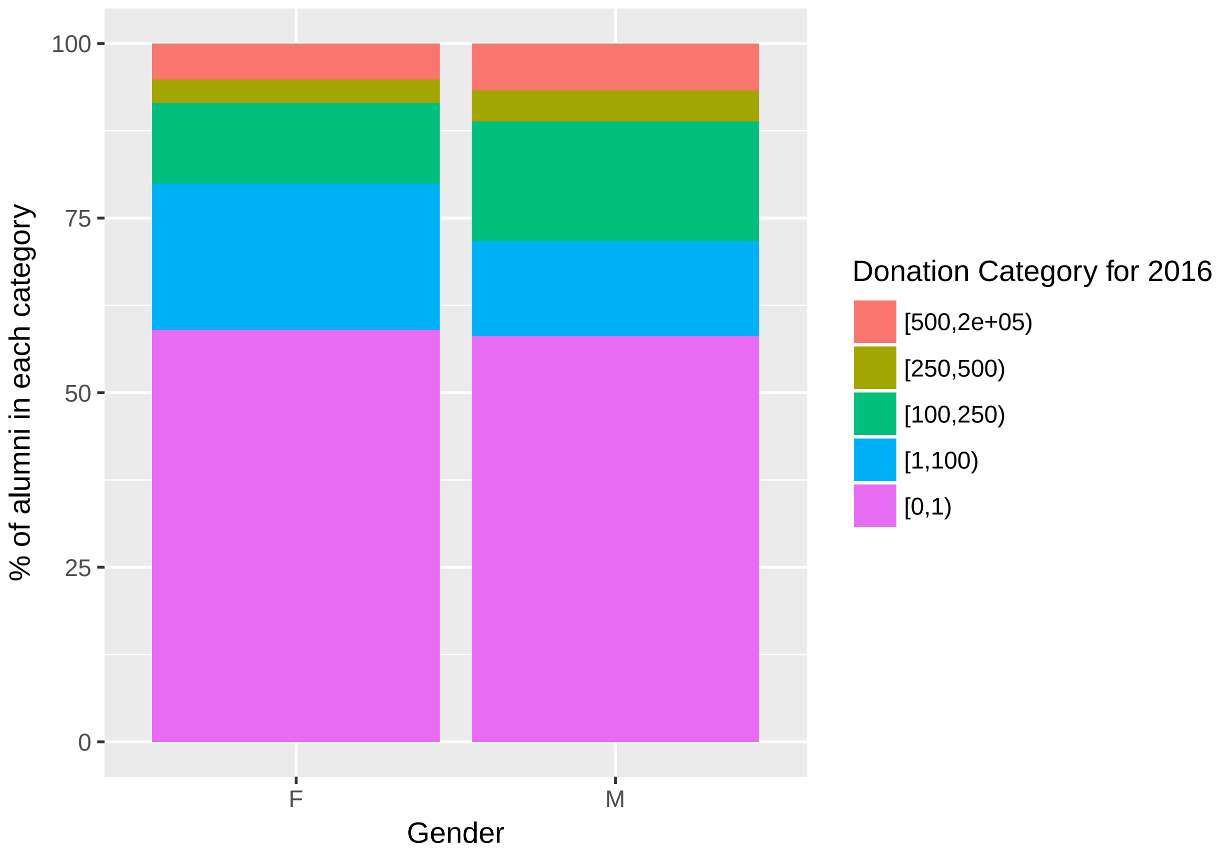
A clear ordinal relationship between years since grad and the amount donated in 2016 is shown in the violin plot, where those that graduated longer ago are more likely to not be in the [0,1] category and more likely to be in higher donation categories as well.

Gender

```
labdata_counts <- with(labdata,
                       aggregate(Gender,
                                 list(Gender=Gender),
                                 length))
labdata_agg <- with(labdata,
                    aggregate(Gender, list(Gender=Gender,
                                           FY16cat=FY16cat),
                    length))
labdata_agg <- merge(labdata_agg, labdata_counts, by="Gender")
labdata_agg <- setNames(labdata_agg, c("Gender", "FY16cat", "Count", "TotalCount"))
labdata_agg$percent <- 100*labdata_agg$Count/labdata_agg$TotalCount
labdata_agg$FY16cat <- factor(labdata_agg$FY16cat,
      levels=c("[500,2e+05)","[250,500)","[100,250)","[1,100)","[0,1)"))

ggp <- ggplot(labdata_agg, aes(x=Gender,y=percent))

ggp + geom_bar(stat="identity", aes(fill=FY16cat))+
  ylab("% of alumni in each category")+xlab("Gender")+
  scale_fill_discrete(name="Donation Category for 2016")
```



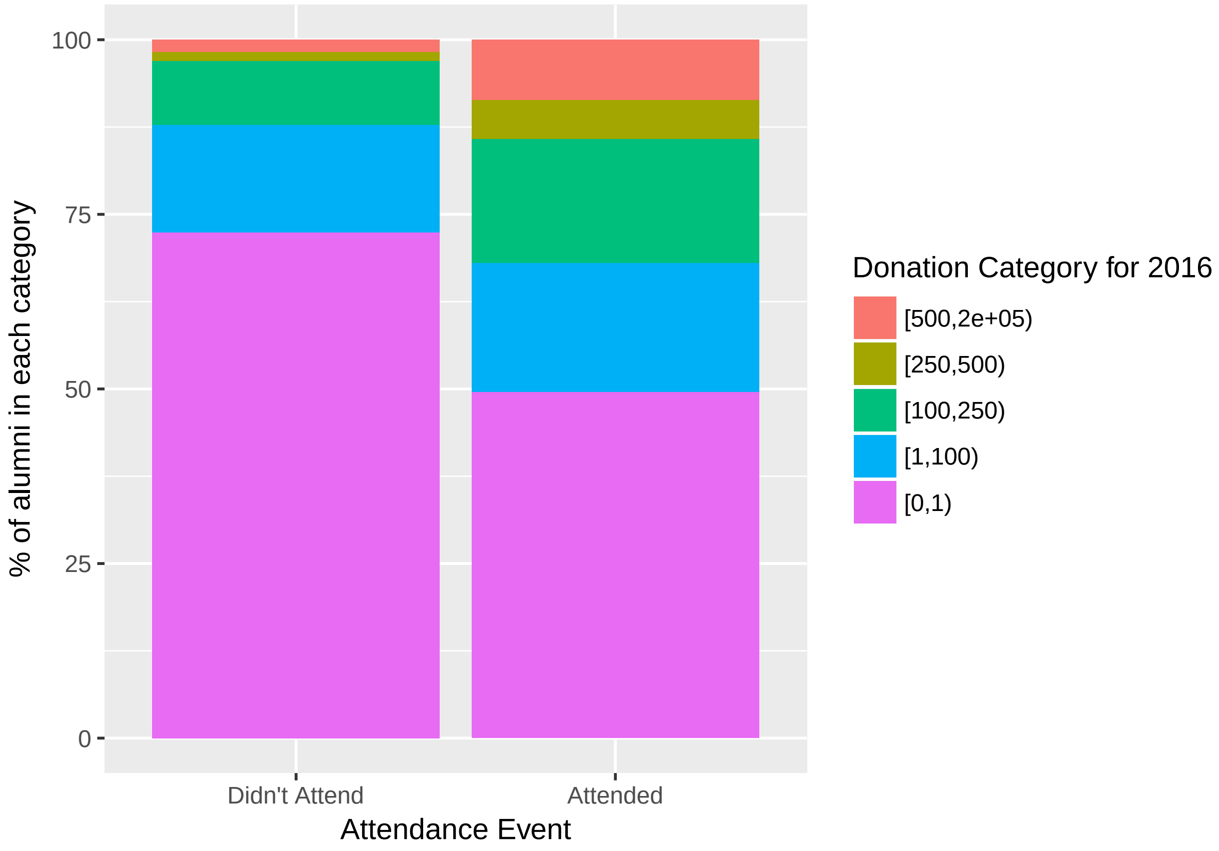
Men appear to be more likely to donate in the top 3 categories, while women appear to be more likely to donate in the [1,100) category. Interestingly, both men and women appear to be just as likely to donate nothing. This may suggest that gender would be more useful for a multinomial model instead of an ordinal model.

Attendance event

```
labdata_counts <- with(labdata,
                       aggregate(AttendenceEvent,
                                 list(AttendenceEvent=AttendenceEvent),
                                 length))
labdata_agg <- with(labdata,
                    aggregate(AttendenceEvent, list(AttendenceEvent=AttendenceEvent,
                                            FY16cat=FY16cat),
                    length))
labdata_agg <- merge(labdata_agg, labdata_counts, by="AttendenceEvent")
labdata_agg <- setNames(labdata_agg, c("AttendenceEvent", "FY16cat", "Count", "TotalCount"))
labdata_agg$percent <- 100*labdata_agg$Count/labdata_agg$TotalCount
labdata_agg$FY16cat <- factor(labdata_agg$FY16cat,
      levels=c("[500,2e+05)","[250,500)","[100,250)","[1,100)","[0,1)"))

ggp <- ggplot(labdata_agg, aes(x=as.factor(AttendenceEvent),y=percent))

ggp + geom_bar(stat="identity", aes(fill=FY16cat))+
  ylab("% of alumni in each category")+xlab("Donation category for 2015")+
  scale_fill_discrete(name="Donation Category for 2016")+
  xlab("Attendance Event")+scale_x_discrete(labels=c("Didn't Attend", "Attended"))
```



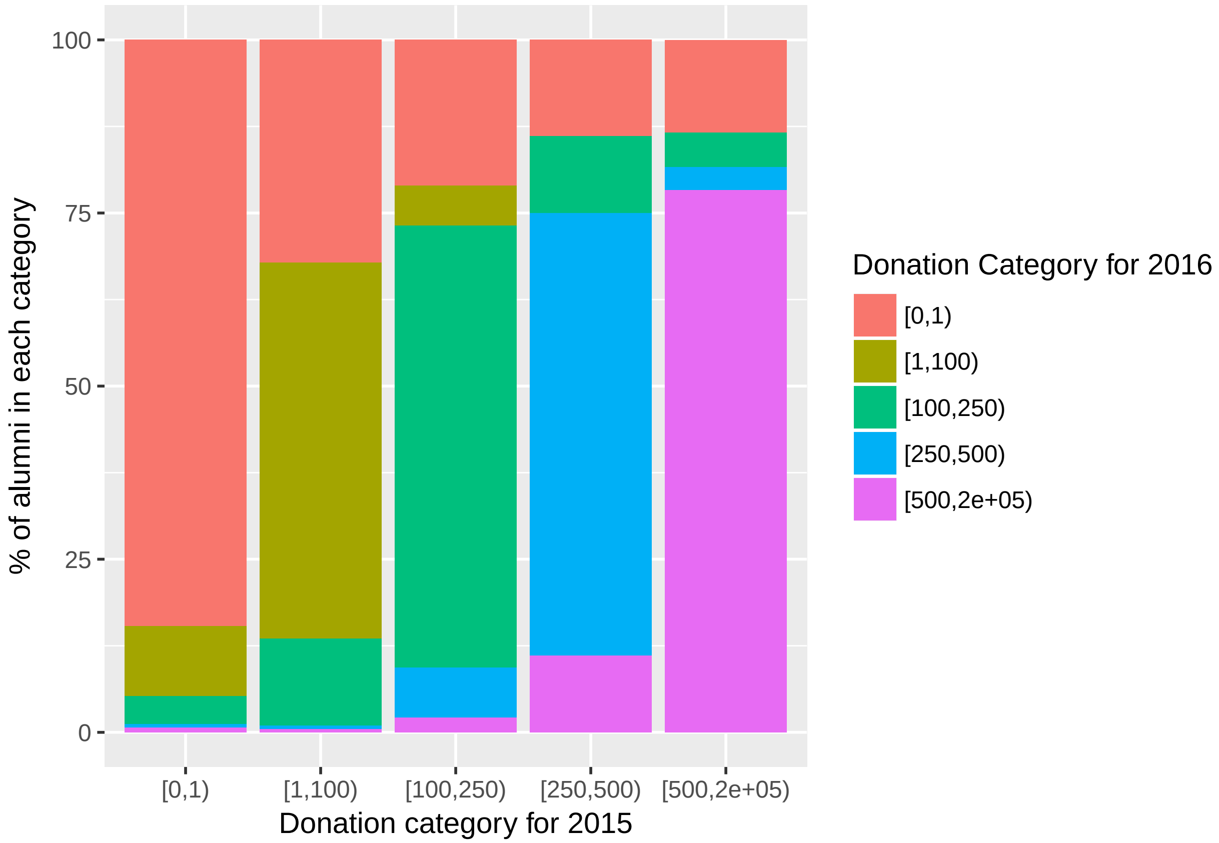
Those who went to the attendance event were much more likely to donate and especially more likely to donate in the higher categories.

Donation category for the previous year (2015)

```
labdata_counts <- with(labdata,
                       aggregate(FY15cat,
                                 list(FY15cat=FY15cat),
                                 length))
labdata_agg <- with(labdata,
                    aggregate(FY15cat, list(FY15cat=FY15cat,
                                            FY16cat=FY16cat),
                    length))
labdata_agg <- merge(labdata_agg, labdata_counts, by="FY15cat")
labdata_agg <- setNames(labdata_agg, c("FY15cat", "FY16cat", "Count", "TotalCount"))
labdata_agg$percent <- 100*labdata_agg$Count/labdata_agg$TotalCount

ggp <- ggplot(labdata_agg, aes(x=FY15cat,y=percent))

ggp + geom_bar(stat="identity", aes(fill=FY16cat))+
  ylab("% of alumni in each category")+xlab("Donation category for 2015")+
  scale_fill_discrete(name="Donation Category for 2016")
```



This stacked bar graph shows what proportion of alumni that fit into donation category X went into the same - or different - category in 2016. As expected, the largest bar in each group represents the same category. That is, the majority of those who donated $0 in 2015 also donated $0 in 2016, the majority of those who donated between $1-$100 in 2015 stayed in that category the next year, etc.

A key takeaway from this visualization is the relative instability of the [1,100) class - compared to other classes, this group was the least likely to donate in the same category. Still, past donations seem to be important in predicting future donations.

Model

Ordinal:

```
clm.out <- clm(FY16cat ~ AttendenceEvent + as.ordered(YearsSinceGrad)  +
                 Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
               data=labdata)
summary(clm.out)
```

```
## formula:
## FY16cat ~ AttendenceEvent + as.ordered(YearsSinceGrad) + Gender + NextDegreeBinary + log(meandonatio
## data:    labdata
##
##  link  threshold nobs logLik  AIC     niter max.grad cond.H
##  logit flexible  1000 -786.76 1601.53 6(0)  5.52e-12 2.0e+03
```

```
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## AttendenceEvent                     0.22961    0.16765   1.370 0.170830
## as.ordered(YearsSinceGrad).L       -0.36358    0.19797  -1.837 0.066278
## as.ordered(YearsSinceGrad).Q        0.41536    0.18099   2.295 0.021736
## as.ordered(YearsSinceGrad).C       -0.46970    0.17434  -2.694 0.007055
## as.ordered(YearsSinceGrad)^4       -0.01900    0.16548  -0.115 0.908571
## GenderM                             0.07905    0.14841   0.533 0.594285
## NextDegreeBinary                    0.35703    0.16287   2.192 0.028370
## log(meandonation + 1)               0.63212    0.07310   8.647  < 2e-16
## gaveLastYear                       -1.54742    0.42530  -3.638 0.000274
## log(meandonation + 1):gaveLastYear  0.77507    0.11776   6.582 4.65e-11
##
## AttendenceEvent
## as.ordered(YearsSinceGrad).L       .
## as.ordered(YearsSinceGrad).Q       *
## as.ordered(YearsSinceGrad).C       **
## as.ordered(YearsSinceGrad)^4
## GenderM
## NextDegreeBinary                   *
## log(meandonation + 1)              ***
## gaveLastYear                       ***
## log(meandonation + 1):gaveLastYear ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                       Estimate Std. Error z value
## [0,1)|[1,100)           3.2125     0.2541   12.64
## [1,100)|[100,250)       4.8274     0.2828   17.07
## [100,250)|[250,500)     6.9618     0.3302   21.08
## [250,500)|[500,2e+05)   7.9996     0.3638   21.99
```

Multinomial:

```
mn.out <- multinom(FY16cat ~ AttendenceEvent + as.ordered(YearsSinceGrad)  +
                 Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
               data=labdata)
```

```
## # weights:  60 (44 variable)
## initial  value 1609.437912
## iter  10 value 972.512190
## iter  20 value 798.678250
## iter  30 value 750.549831
## iter  40 value 747.422718
## iter  50 value 747.053742
## final  value 747.049617
## converged
```

```
summary(mn.out)
```

```
## Call:
## multinom(formula = FY16cat ~ AttendenceEvent + as.ordered(YearsSinceGrad) +
##     Gender + NextDegreeBinary + log(meandonation + 1) * gaveLastYear,
##     data = labdata)
```

```
##
## Coefficients:
##             (Intercept) AttendenceEvent as.ordered(YearsSinceGrad).L
## [1,100)       -3.166582     -0.04348484                  -0.79855530
## [100,250)     -5.356003      0.45061068                  -0.09209195
## [250,500)     -7.590248      1.35370518                   0.05971283
## [500,2e+05)   -9.218344      1.08724402                  -0.35886973
##             as.ordered(YearsSinceGrad).Q as.ordered(YearsSinceGrad).C
## [1,100)                       0.01769485                  -0.12499374
## [100,250)                     0.15079973                  -0.49176025
## [250,500)                    -0.01529638                  -0.86512284
## [500,2e+05)                   0.13728926                   0.03059583
##             as.ordered(YearsSinceGrad)^4    GenderM NextDegreeBinary
## [1,100)                      -0.08079175 -0.4394256        0.7748609
## [100,250)                     0.08061032  0.4057717        0.2472710
## [250,500)                    -0.19265818  0.3573167       -0.2262256
## [500,2e+05)                  -0.57005106  0.1253457        1.3491517
##             log(meandonation + 1) gaveLastYear
## [1,100)                 0.4031088    3.2336304
## [100,250)               0.7858133   -0.3226104
## [250,500)               0.7307393   -3.7569921
## [500,2e+05)             1.0349362   -9.0391079
##             log(meandonation + 1):gaveLastYear
## [1,100)                             -0.5311790
## [100,250)                            0.4248541
## [250,500)                            1.2736826
## [500,2e+05)                          2.0698748
##
## Std. Errors:
##             (Intercept) AttendenceEvent as.ordered(YearsSinceGrad).L
## [1,100)       0.2907128       0.2175366                    0.3062453
## [100,250)     0.5530622       0.2778891                    0.3186168
## [250,500)     1.3615012       0.5559241                    0.5743812
## [500,2e+05)   1.7189960       0.5814011                    0.6813522
##             as.ordered(YearsSinceGrad).Q as.ordered(YearsSinceGrad).C
## [1,100)                        0.2828600                    0.2577212
## [100,250)                      0.2952388                    0.2833009
## [250,500)                      0.5194160                    0.4677499
## [500,2e+05)                    0.6070903                    0.4953641
##             as.ordered(YearsSinceGrad)^4   GenderM NextDegreeBinary
## [1,100)                        0.2372343 0.2051271        0.2251815
## [100,250)                      0.2605105 0.2473229        0.2668604
## [250,500)                      0.4229064 0.4036913        0.4256573
## [500,2e+05)                    0.4579001 0.4389489        0.5616576
##             log(meandonation + 1) gaveLastYear
## [1,100)                0.08318712    0.5602868
## [100,250)              0.13466141    0.8861426
## [250,500)              0.33713904    1.8407730
## [500,2e+05)            0.37688786    2.2738091
##             log(meandonation + 1):gaveLastYear
## [1,100)                              0.1632546
## [100,250)                            0.2191812
## [250,500)                            0.4350921
## [500,2e+05)                          0.4870697
```

```
##
## Residual Deviance: 1494.099
## AIC: 1582.099
```

Predictions

```
#define randomized train/test set with 80/20 split
randomRows <- sample(1000, 800, replace=FALSE)
  train <- labdata[randomRows,]
  test <- labdata[-randomRows,]
```

Multinomial:

```
mn.out <- multinom(FY16cat ~ AttendenceEvent + YearsSinceGrad  +
                       Gender+NextDegreeBinary+log(meandonation+1)+gaveLastYear,
                     data=train)
```

```
## # weights:  40 (28 variable)
## initial  value 1287.550330
## iter  10 value 821.269162
## iter  20 value 636.372340
## iter  30 value 601.310056
## iter  40 value 594.352999
## final  value 594.338813
## converged
```

```
  mn.preds <- predict(mn.out, test, type="class")
confusionMatrix(mn.preds, test$FY16cat)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)          92      19         2         1           0
##   [1,100)         9      17         6         0           0
##   [100,250)       8       3        20         7           0
##   [250,500)       0       0         0         0           0
##   [500,2e+05)     4       0         2         0          10
##
## Overall Statistics
##
##                Accuracy : 0.695
##                  95% CI : (0.6261, 0.758)
##     No Information Rate : 0.565
##     P-Value [Acc > NIR] : 0.0001093
##
##                   Kappa : 0.5035
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: [0,1) Class: [1,100) Class: [100,250)
## Sensitivity                0.8142         0.4359           0.6667
## Specificity                0.7471         0.9068           0.8941
## Pos Pred Value             0.8070         0.5313           0.5263
## Neg Pred Value             0.7558         0.8690           0.9383
## Prevalence                 0.5650         0.1950           0.1500
## Detection Rate               0.4600         0.0850           0.1000
## Detection Prevalence         0.5700         0.1600           0.1900
## Balanced Accuracy            0.7806         0.6714           0.7804
##                      Class: [250,500) Class: [500,2e+05)
## Sensitivity                      0.00             1.0000
## Specificity                      1.00             0.9684
## Pos Pred Value                    NaN             0.6250
## Neg Pred Value                   0.96             1.0000
## Prevalence                       0.04             0.0500
## Detection Rate                   0.00             0.0500
## Detection Prevalence             0.00             0.0800
## Balanced Accuracy                0.50             0.9842
```

Ordinal:

```
clm.out <- clm(FY16cat ~ AttendenceEvent + as.ordered(YearsSinceGrad)  +
                 Gender+NextDegreeBinary+log(meandonation+1)*gaveLastYear,
                   data=train)

clm.preds <- predict(clm.out, test, type="class")
confusionMatrix(clm.preds$fit, test$FY16cat)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    [0,1) [1,100) [100,250) [250,500) [500,2e+05)
##   [0,1)          95      22         1         1           0
##   [1,100)         8      11         9         2           0
##   [100,250)       7       6        18         5           0
##   [250,500)       0       0         0         0           0
##   [500,2e+05)     3       0         2         0          10
##
## Overall Statistics
##
##                Accuracy : 0.67
##                  95% CI : (0.6002, 0.7347)
##     No Information Rate : 0.565
##     P-Value [Acc > NIR] : 0.001549
##
##                   Kappa : 0.4535
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: [0,1) Class: [1,100) Class: [100,250)
## Sensitivity                0.8407         0.2821           0.6000
## Specificity                0.7241         0.8820           0.8941
## Pos Pred Value             0.7983         0.3667           0.5000
## Neg Pred Value             0.7778         0.8353           0.9268
## Prevalence                 0.5650         0.1950           0.1500
## Detection Rate             0.4750         0.0550           0.0900
## Detection Prevalence       0.5950         0.1500           0.1800
## Balanced Accuracy          0.7824         0.5820           0.7471
##                      Class: [250,500) Class: [500,2e+05)
## Sensitivity                      0.00             1.0000
## Specificity                      1.00             0.9737
## Pos Pred Value                    NaN             0.6667
## Neg Pred Value                   0.96             1.0000
## Prevalence                       0.04             0.0500
## Detection Rate                   0.00             0.0500
## Detection Prevalence             0.00             0.0750
## Balanced Accuracy                0.50             0.9868
```