

Are transfer student rates persistence at a University better after attending an Orientation Session?

Olabode Makinde

December 20, 2017

Problem

The Association of American colleges and Universities acknowledged in 2002 that "transfer students represent an increasingly growing but understudied student population". The transition of these students across multiple institutions can be very stressful and problematic. As a result, it has become paramount for any educational institution to dedicate resources to attract and recruit these unique students.

Furthermore, De los Santos and Wright wrote in their study published in 1989 that "Transfer students are an incredibly varied student population. Like all student cohorts, transfer students can be of traditional age or older, attending part-time or full-time, commuting or living on campus, and working full- or part-time. But they can also enter as first-year students, sophomores, juniors, or seniors via any paths: co-enrolling, reverse transferring, and swirling" (p.82).

Orientation programming is a critical and useful component of this recruiting process. According to Mayhew et al (2008), "The primary mission of orientation programs is to aid in the successful transition of new students to college, but orientation professionals are often illequipped to provide evidence related to the efficacy of their programs" (p.2).

The trend that I noticed as I consumed research papers dedicated to this topic is that an overwhelming majority of them are about first year students, general education courses, PostBacc e.t.c

Williford argued in 2001 that "students not only attended these orientation programs but credited them with strongly influencing their expectations of the collegiate experience and in improving adjustment, rates of performance, retention and graduation (p.327)"

This University is a regional, comprehensive public university located on a 380-acre campus in Oldenburg in the US state of Washington. For fiscal year 2017, The University accepted 2553 unique transfer students but only 83% of them enrolled for classes.

The University's director of Sophomore Year Experience department, One of the programs set up by the university to help improve retention and persistence of these transfer students is looking at ways to increase the efficiency of their recruiting tools. Orientation programming is one those important tools.

Research Design

The bulk of analysis for this study is performed using R. R has a robust number of libraries with unique functions that can help us efficiently analyze the relationship between transfer students and orientation sessions. This study will use the noncentral chi-squared distribution which is a generalization of Pearson's Chi-squared test of independence.

The idea here is very straightforward, the analysis will be run in R to calculate a chi-square statistic and then use that to show the relationship between the two categorical variables we are concerned about in this study (Orientation and Persistence). A chi-square test for independence compares two variables in a contingency table to see if they are related. In a more general sense, it tests to see whether distributions of categorical variables differ from each another.

Cohorts of transfer students that enrolled at the university for Fall 2014, Fall 2015 and Fall 2016 will be analysed for this study. This study will answer questions that relates to these cohorts only due to lack of rich historical data on orientation at the university. At the end of the study, comparisons will be made across the cohorts to establish any differences that exists between them.

The calculated chi-square statistic will be compared to a critical value from a chi-square table. If the chi-square value is more than the critical value, then there is a significant difference. The p value from pchisq will be used to indicate significance across all orientation session types involved in this study.

See below for my definition of chi square statistic as used in this study where

O = Observed frequencies

E = Expected frequencies

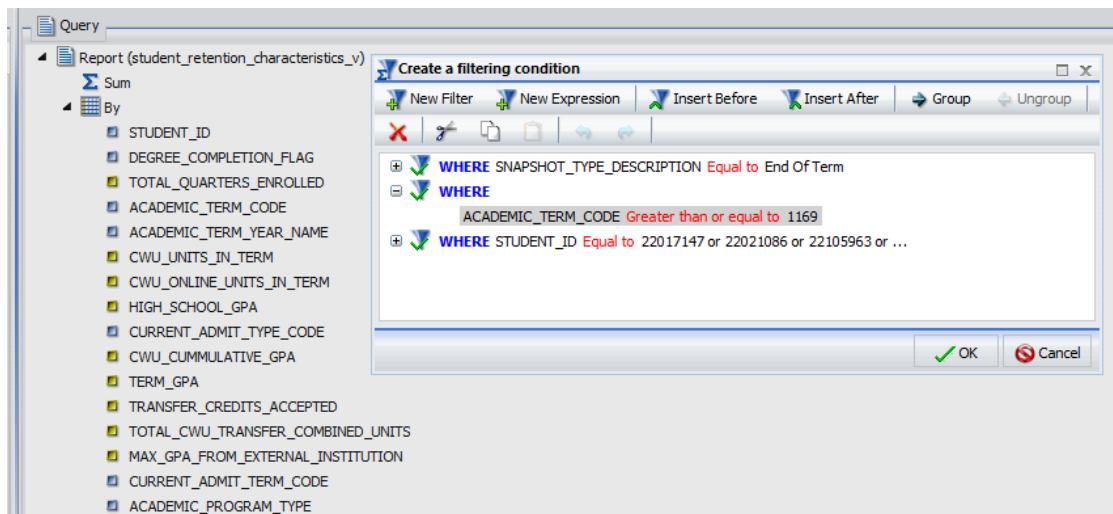
X^2 = test statistic

$$\chi^2 = \sum \frac{(o - e)^2}{e}$$

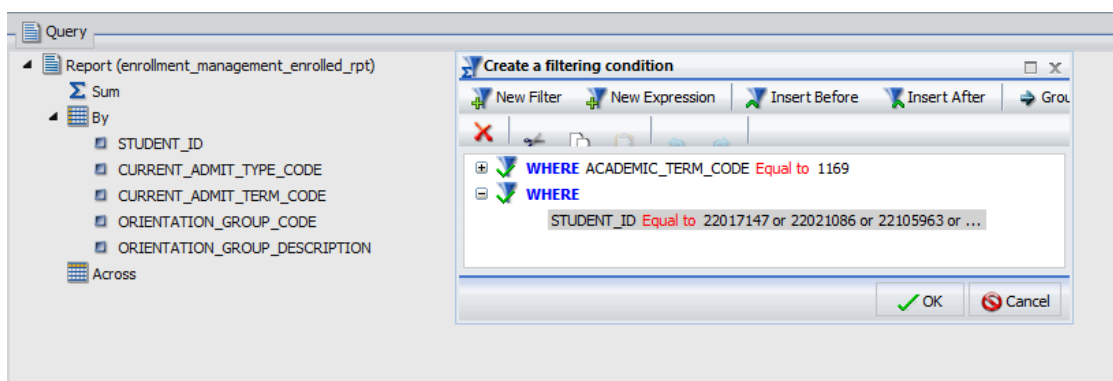
Statistic

Focus Executables:

Note that student ids are unique for each cohort



Student Retention Characteristics



Orientation Description

DataPulls:

- unique IDs of transfer students for Fall 2014 cohort, Fall 2015 cohort, and Fall 2016 cohort (these cohorts are based on IPEDs definitions) then aggregate information on these unique students about their:
- a file of retention information (retention_characteristics)
- a table of orientation codes pulled from enrollment_management_enrolled_rpt

Load Packages & define funtions

#Load in dataset in sequential order as arranged in root folder

#install packages and enforce deficiencies - remove these comments if your have never installed any of this packages in your version of Rstudio

#install.packages("dplyr", dependencies = TRUE)

```
#install.packages("sjPlot", dependencies = TRUE)
#install.packages("ggPlot2", dependencies = TRUE)
#install.packages("reshape2", dependencies = TRUE)
#install.packages("gplots", dependencies = TRUE)
#install.packages("e1071", dependencies = TRUE)
#install.packages("class", dependencies = TRUE)
#install.packages("data.table", dependencies = TRUE)
#install.packages("caret", dependencies = TRUE)
#install.packages("mosaic", dependencies = TRUE)
#install.packages("ca", dependencies = TRUE)

#Load package Libraries

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.3
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(sjPlot)

## Warning: package 'sjPlot' was built under R version 3.4.3
## Warning in checkMatrixPackageVersion(): Package version inconsistency
## detected.
## TMB was built with Matrix version 1.2.12
## Current Matrix version is 1.2.11
## Please re-install 'TMB' from source or restore original 'Matrix' package
## #refugeeswelcome

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.3

library(reshape2)

## Warning: package 'reshape2' was built under R version 3.4.3

library(gplots)

## Warning: package 'gplots' was built under R version 3.4.3
```

```
##
## Attaching package: 'gplots'

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

library(e1071)

## Warning: package 'e1071' was built under R version 3.4.3

library(class)

## Warning: package 'class' was built under R version 3.4.3

library(data.table)

## Warning: package 'data.table' was built under R version 3.4.3

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
##
##      dcast, melt

## The following objects are masked from 'package:dplyr':
##
##      between, first, last

library(caret)

## Warning: package 'caret' was built under R version 3.4.3

## Loading required package: lattice

library(mosaic)

## Warning: package 'mosaic' was built under R version 3.4.3

## Loading required package: ggformula

## Warning: package 'ggformula' was built under R version 3.4.3

##
## New to ggformula? Try the tutorials:
##   learnr::run_tutorial("introduction", package = "ggformula")
##   learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Warning: package 'mosaicData' was built under R version 3.4.3

## Loading required package: Matrix
```

```
##
## The 'mosaic' package masks several functions from core packages in order
to add
## additional features. The original behavior of these functions should not
be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading
mosaic.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##      mean

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

## The following objects are masked from 'package:dplyr':
##
##      count, do, tally

## The following objects are masked from 'package:stats':
##
##      binom.test, cor, cor.test, cov, fivenum, IQR, median,
##      prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##      max, mean, min, prod, range, sample, sum

#symbol %in% is an alias meaning is_in

##>% is a pipe Lefthand operator, makes code cleaner. here Left hand values
of code forward into expressions that appear on the right hand side, this is
an integral part of the dplyr package and will appear often in all my code

'!in%' <- function(x,y){('%in%')(x,y)}

#Load in 2014 data set
#here we add the retention charactersitics we are intrested in

transfer1149<- read.csv(file = file.choose()) %>%
filter(CURRENT_ADMIT_TYPE_CODE != ".") %>% filter(CURRENT_ADMIT_TERM_CODE
%in% c(1149,1151,1153,1156,1159,1161,1163,1166,1169,1171,1173,1176,1179))

#orientation 2014 data pull
orient1149 <- read.csv(file = file.choose())
```

```
#Load in 2015 data set
transfer1159<- read.csv(file = file.choose()) %>%
filter(CURRENT_ADMIT_TYPE_CODE != ".") %>% filter(CURRENT_ADMIT_TERM_CODE
%in% c(1156,1159,1161,1163,1166,1169,1171,1173,1176,1179))
```

```
#orientation 2015 data pull
orient1159 <- read.csv(file = file.choose())
```

```
#Load in 2016 data set
transfer1169<- read.csv(file = file.choose()) %>%
filter(CURRENT_ADMIT_TYPE_CODE != ".") %>% filter(CURRENT_ADMIT_TERM_CODE
%in% c(1166,1169,1171,1173,1176,1179))
```

```
#orientation 2016 data pull
orient1169 <- read.csv(file = file.choose())
```

Fall 2014 Orientation Sessions and Transfer students Persistence statistical significance

*#transform raw transfer retention charatersitics dataset into a data.table
format in order to make it easier to change input by reference*

```
setDT(transfer1149)[,paste0("ACADEMIC_TERM_YEAR_NAME", 1:2) :=
tstrsplit(ACADEMIC_TERM_YEAR_NAME, " ")] #split academic year name into just  
academic year and actual Year.
```

```
#create new variable in our data.table to accomodate split
transfer1149$ACADEMIC_TERM_YEAR_NAME2 <-
as.numeric(transfer1149$ACADEMIC_TERM_YEAR_NAME2) #academic year name only  
variable added
```

```
transfer1149$ActualYear <- ifelse(grepl("Spring|Winter|Summer",
transfer1149$ACADEMIC_TERM_YEAR_NAME),
transfer1149$ACADEMIC_TERM_YEAR_NAME2 - 1,
transfer1149$ACADEMIC_TERM_YEAR_NAME2) #year numeric only variable added
```

*#there are duplicate records for each student ID to indicate their progress
after they enrolled at the university
#now we will attempt to combine duplicate record to figure out who dropped or
retained*

#here you see how %>% makes the code easier to read

#Find Maximum Quarters Enrolled

```
maxterm <- transfer1149 %>%
  group_by(STUDENT_ID) %>%
  filter(TOTAL_QUARTERS_ENROLLED == max(TOTAL_QUARTERS_ENROLLED)) %>%
  mutate(MaxQuarters = TOTAL_QUARTERS_ENROLLED) %>%
  dplyr::select(STUDENT_ID, MaxQuarters)
```

#Find Terms Enrolled in First Year

```
minyr <- transfer1149 %>%
```

```

group_by(STUDENT_ID) %>%
slice(which.min(ActualYear)) %>%
mutate(MinYear = ActualYear) %>%
dplyr::select(STUDENT_ID, MinYear)

minyr2 <- merge(transfer1149, minyr, by="STUDENT_ID", all.x=TRUE)

termsinminyear <- minyr2 %>%
  filter(ACADEMIC_TERM_YEAR_NAME1 != "Summer") %>%
  group_by(STUDENT_ID) %>%
  filter(ActualYear == MinYear) %>%
  tally()
setnames(termsinminyear, "n", "FYTermsRetained")

#Did Student Come Back for Second Fall?
WhichFallsRetained <- transfer1149 %>%
  filter(ACADEMIC_TERM_YEAR_NAME1 == "Fall") %>%
  group_by(STUDENT_ID) %>%
  slice(which.min(ACADEMIC_TERM_CODE)) %>%
  mutate(FirstFall = ACADEMIC_TERM_CODE) %>%
  mutate(SecondFall = FirstFall +10)%>%
  dplyr::select(STUDENT_ID, SecondFall)

hadsecondfall <- merge(transfer1149, WhichFallsRetained, by="STUDENT_ID",
all.x=TRUE)
hadsecondfall <- hadsecondfall %>%
  group_by(STUDENT_ID) %>%
  mutate(SecondFallHappened = ifelse(SecondFall == ACADEMIC_TERM_CODE, "Yes",
"No")) %>%
  filter(SecondFallHappened == "Yes") %>%
  dplyr::select(STUDENT_ID, SecondFallHappened)

#Now Merge ALL These, Find Student Status
findingretained <- merge(maxterm, termsinminyear, by="STUDENT_ID",
all.x=TRUE)
findingretained <- merge(findingretained, hadsecondfall, by="STUDENT_ID",
all.x=TRUE)
findingretained <- findingretained %>%
  mutate(StudentStatus = ifelse(is.na(SecondFallHappened) & MaxQuarters <=3,
"Drop", ifelse(is.na(SecondFallHappened) & MaxQuarters >3, "Stop-Out",
ifelse(SecondFallHappened == "Yes" & FYTermsRetained <3, "Stop-Out",
"Retained")))) %>%
  mutate(StudentStatusGen = ifelse(is.na(SecondFallHappened) & MaxQuarters
<=3, "Drop/Trans/Stop", ifelse(is.na(SecondFallHappened) & MaxQuarters >3,
"Drop/Trans/Stop", ifelse(SecondFallHappened == "Yes" & FYTermsRetained <3,
"Retained", "Retained"))))

#here we join the retained variables to the original dataset

```



```
persitencetransfer1149 <- merge(transfer1149, findingretained,  
by="STUDENT_ID", all.x=TRUE)
```

#Now, take only the first term for the rest of this study to ignore duplicates

```
persitencetransfer1149 <- persitencetransfer1149 %>%  
  filter(TOTAL_QUARTERS_ENROLLED == 1) %>% filter(ACADEMIC_TERM_CODE ==  
"1149")
```

#data integrity check - open this document to see the retained info properly joined to original

```
write.csv(persitencetransfer1149, "retainedJoinRet1149.csv")
```

#here we add in orientation info

```
persitencetransfer1149orient <- merge(persitencetransfer1149, orient1149,  
by=c("STUDENT_ID", "CURRENT_ADMIT_TERM_CODE", "CURRENT_ADMIT_TYPE_CODE"),  
all.x=TRUE)
```

#data integrity check - open this document to see that new orientation info properly joined to retainedRet

```
write.csv(persitencetransfer1149orient, "orientJoinRet1149.csv")
```

##Fall 2014 Transfer Cohort chiSquare analysis

#data integrity check - open this document to see that table is correct

```
write.csv(persitencetransfer1149orient, "prepForChiSquareAnaysis1149.csv")
```

#prep knit table

```
persitencetransfer1149orient$ORIENTATION_GROUP_DESCRIPTION <-  
factor(persitencetransfer1149orient$ORIENTATION_GROUP_DESCRIPTION,  
levels=c("Orientation Session 1", "Orientation Session 2", "Orientation  
Session 3", "Orientation Session 4", "Orientation Session 5", "Orientation  
Session 6", "Late Orientation Session 1", "Late Orientation Session 2", "Late  
Orientation Session 3", "Late Orientation Session 4", "None Attended ",  
"Des Moines Orientation", "Lynnwood Orientation", "Online Orientation  
Session", "Orientation Session-Quarterly", "Pierce Orientation", "Transfer  
Orientation Session", "Transfer Orientation Session 2", "Wenatchee  
Orientation"))
```

#Fall 2014 Transfer Cohort chiSquare analysis

#here we calculate statistic

#here we make data readable

```

OverallPercent <- persitencetransfer1149orient %>% group_by(StudentStatusGen)
%>% tally() %>% mutate(sum = sum(n), percent=n/sum) %>%
dplyr::select(StudentStatusGen, percent)
Pret <- as.numeric(as.character(OverallPercent[2,2]))
Pdrop <- as.numeric(as.character(OverallPercent[1,2]))

otally <- persitencetransfer1149orient %>%
  filter(ActualYear %in% 2014) %>%
  group_by(ORIENTATION_GROUP_DESCRIPTION, StudentStatusGen) %>%
  tally() %>%
  dcast(ORIENTATION_GROUP_DESCRIPTION~StudentStatusGen, value.var="n") %>%
  filter(ORIENTATION_GROUP_DESCRIPTION %in% c("Orientation Session 1",
"Orientation Session 2", "Orientation Session 3", "Orientation Session 4",
"Orientation Session 5", "Orientation Session 6", "Late Orientation Session 1",
"Late Orientation Session 2", "Late Orientation Session 3", "Late Orientation
Session 4", "None Attended ",
"Des Moines Orientation", "Lynnwood Orientation", "Online Orientation
Session", "Orientation Session-Quarterly", "Pierce Orientation", "Transfer
Orientation Session", "Transfer Orientation Session 2", "Wenatchee
Orientation"))
otally[is.na(otally)] <- 0
otally <- otally %>%
  mutate(Total = Retained + `Drop/Trans/Stop`) %>%
  mutate(PercentageRetained = paste(as.integer(100*Retained/Total), "%")) %>%
  mutate(ExpectedDrop = Pdrop*Total) %>%
  mutate(ExpectedRetained = Pret*Total) %>%
  mutate(ChiVar = ((Retained-ExpectedRetained)^2/ExpectedRetained) +
((`Drop/Trans/Stop`-ExpectedDrop)^2/ExpectedDrop)) %>%
  mutate(pvalue = round(1-(pchisq(ChiVar,1)),3)) %>%
  mutate(SignificantDifference = ifelse(pvalue<.001, "***",
ifelse(pvalue<.01, "**", ifelse(pvalue<.05, "*", "")))) %>%
  dplyr::select(ORIENTATION_GROUP_DESCRIPTION, `Retained`, `Drop/Trans/Stop`,
`Total`, `PercentageRetained`, SignificantDifference)

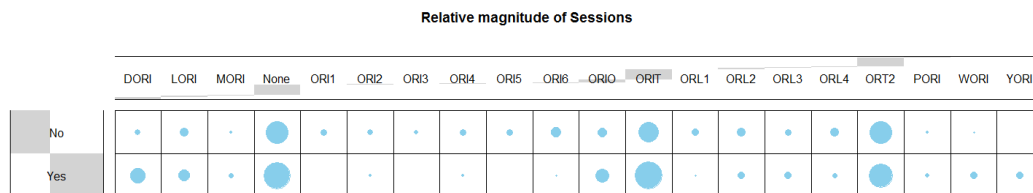
colnames(otally) <- c("Orientation Code", "Retained", "Not Retained",
"Total", "Percent Retained", "Statistical Significance")

knitr::kable(otally)

```

Orientation Code	Retained	Not Retained	Total	Percent Retained	Statistical Significance
Orientation Session 1	6	3	9	66 %	
Orientation Session 2	8	2	10	80 %	
Orientation Session 3	3	1	4	75 %	
Orientation Session 4	9	2	11	81 %	

Orientation Session 5	5	5	10	50 %	*
Orientation Session 6	13	10	23	56 %	**
Late Orientation Session 1	10	2	12	83 %	
Late Orientation Session 2	19	11	30	63 %	*
Late Orientation Session 3	16	5	21	76 %	
Late Orientation Session 4	18	4	22	81 %	
Des Moines Orientation	52	6	58	89 %	
Lynnwood Orientation	38	7	45	84 %	
Online Orientation Session	36	10	46	78 %	
Pierce Orientation	5	1	6	83 %	
Transfer Orientation Session	232	31	263	88 %	**
Transfer Orientation Session 2	195	37	232	84 %	
Wenatchee Orientation	10	2	12	83 %	



```

#library("graphics")
#mosaicplot(orient1149.tab, shade = TRUE, las=2,
            #main = "Insight into Orientation Sessions")

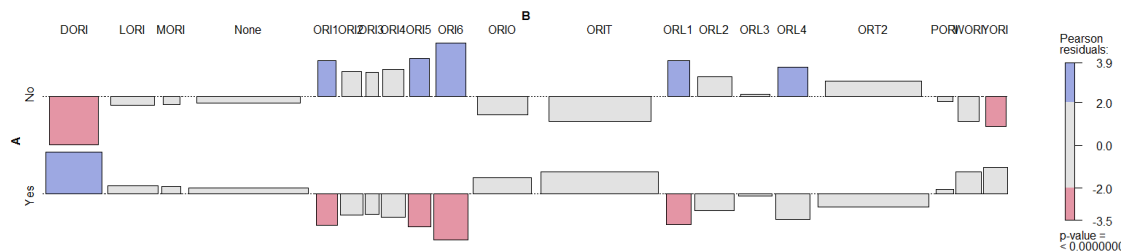
# install.packages("vcd")
library("vcd")

## Warning: package 'vcd' was built under R version 3.4.3
## Loading required package: grid
##
## Attaching package: 'vcd'
## The following object is masked from 'package:mosaic':
##
##      mplot
?assoc

## starting httpd help server ...
## done

# plot just a subset of the table
assoc(head(orient1149.tab), shade = TRUE, split_vertical = FALSE, las=2)

```



```

chisq.test(orient1149.tab)

## Warning in chisq.test(orient1149.tab): Chi-squared approximation may be
## incorrect
##
## Pearson's Chi-squared test
##
## data: orient1149.tab
## X-squared = 140.79, df = 19, p-value < 0.00000000000000022

```

Fall 2015 Orientation Sessions and Transfer students Persistence statistical significance

#transform raw transfer retention charatersitics dataset into a data.table format in order to make it easier to change input by reference

```

setDT(transfer1159)[, paste0("ACADEMIC_TERM_YEAR_NAME", 1:2) :=
tstrsplit(ACADEMIC_TERM_YEAR_NAME, " ")] #split academic year name into just
academic year and actual Year.

#create new variable in our data.table to accomodate split
transfer1159$ACADEMIC_TERM_YEAR_NAME2 <-
as.numeric(transfer1159$ACADEMIC_TERM_YEAR_NAME2) #academic year name only
variable added
transfer1159$ActualYear <- ifelse(grepl("Spring|Winter|Summer",
transfer1159$ACADEMIC_TERM_YEAR_NAME),
transfer1159$ACADEMIC_TERM_YEAR_NAME2 - 1,
transfer1159$ACADEMIC_TERM_YEAR_NAME2) #year numeric only variable added

#there are duplicate records for each student ID to indicate their progress
after they enrolled at the university
#now we will attempt to combine duplicate record to figure out who dropped or
retained

#here you see how %>% makes the code easier to read

#Find Maximum Quarters Enrolled
maxterm1159 <- transfer1159 %>%
  group_by(STUDENT_ID) %>%
  filter(TOTAL_QUARTERS_ENROLLED == max(TOTAL_QUARTERS_ENROLLED)) %>%
  mutate(MaxQuarters = TOTAL_QUARTERS_ENROLLED) %>%
  dplyr::select(STUDENT_ID, MaxQuarters)

#Find Terms Enrolled in First Year
minyr1159 <- transfer1159 %>%
  group_by(STUDENT_ID) %>%
  slice(which.min(ActualYear)) %>%
  mutate(MinYear = ActualYear) %>%
  dplyr::select(STUDENT_ID, MinYear)

minyr21159 <- merge(transfer1159, minyr1159, by="STUDENT_ID", all.x=TRUE)

termsinminyear1159 <- minyr21159 %>%
  filter(ACADEMIC_TERM_YEAR_NAME1 != "Summer") %>%
  group_by(STUDENT_ID) %>%
  filter(ActualYear == MinYear) %>%
  tally()
setnames(termsinminyear1159, "n", "FYTermsRetained")

#Did Student Come Back for Second Fall?

WhichFallsRetained1159 <- transfer1159 %>%
  filter(ACADEMIC_TERM_YEAR_NAME1 == "Fall") %>%
  group_by(STUDENT_ID) %>%

```

```

slice(which.min(ACADEMIC_TERM_CODE)) %>%
mutate(FirstFall = ACADEMIC_TERM_CODE) %>%
mutate(SecondFall = FirstFall +10)%>%
dplyr::select(STUDENT_ID, SecondFall)

```

```

hadsecondfall1159 <- merge(transfer1159, WhichFallsRetained1159,
by="STUDENT_ID", all.x=TRUE)
hadsecondfall1159 <- hadsecondfall1159 %>%
  group_by(STUDENT_ID) %>%
  mutate(SecondFallHappened = ifelse(SecondFall == ACADEMIC_TERM_CODE, "Yes",
"No")) %>%
  filter(SecondFallHappened == "Yes") %>%
  dplyr::select(STUDENT_ID, SecondFallHappened)

```

#Now Merge ALL These, Find Student Status

```

findingretained1159 <- merge(maxterm1159, termsinminyear1159,
by="STUDENT_ID", all.x=TRUE)
findingretained1159 <- merge(findingretained1159, hadsecondfall1159,
by="STUDENT_ID", all.x=TRUE)
findingretained1159 <- findingretained1159 %>%
  mutate(StudentStatus = ifelse(is.na(SecondFallHappened) & MaxQuarters <=3,
"Drop", ifelse(is.na(SecondFallHappened) & MaxQuarters >3, "Stop-Out",
ifelse(SecondFallHappened == "Yes" & FYTermsRetained <3, "Stop-Out",
"Retained")))) %>%
  mutate(StudentStatusGen = ifelse(is.na(SecondFallHappened) & MaxQuarters
<=3, "Drop/Trans/Stop", ifelse(is.na(SecondFallHappened) & MaxQuarters >3,
"Drop/Trans/Stop", ifelse(SecondFallHappened == "Yes" & FYTermsRetained <3,
"Retained", "Retained"))))

```

#here we join the retained variables to the original dataset

```

persitencetransfer1159 <- merge(transfer1159, findingretained1159,
by="STUDENT_ID", all.x=TRUE)

```

#Now, take only the first term for the rest of this study to ignore duplicates

```

persitencetransfer1159 <- persitencetransfer1159 %>%
  filter(TOTAL_QUARTERS_ENROLLED == 1) %>% filter(ACADEMIC_TERM_CODE ==
"1159")

```

#data integrity check - open this document to see the retained info properly joined to original

```

write.csv(persitencetransfer1159, "retainedJoinRet1159.csv")

```

#here we add in orientation info

```

persitencetransfer1159orient <- merge(persitencetransfer1159, orient1159,

```

```

by=c("STUDENT_ID", "CURRENT_ADMIT_TERM_CODE", "CURRENT_ADMIT_TYPE_CODE"),
all.x=TRUE)

#data integrity check - open this document to see that new orientation info
properly joined to retainedRet
write.csv(persitencetransfer1159orient, "orientJoinRet1159.csv")

#Fall 2015 Transfer Cohort chiSquare analysis

#data integrity check - open this document to see that table is correct

#write.csv(persitencetransfer1159orient, "prepForChiSquareAnalysis1159.csv")

OverallPercent1159 <- persitencetransfer1159orient %>%
group_by(StudentStatusGen) %>% tally() %>% mutate(sum = sum(n),
percent=n/sum) %>% dplyr::select(StudentStatusGen, percent)
Pret1159 <- as.numeric(as.character(OverallPercent[2,2]))
Pdrop1159<- as.numeric(as.character(OverallPercent[1,2]))

otally1159 <- persitencetransfer1159orient %>%
  filter(ActualYear %in% 2015) %>%
  group_by(ORIENTATION_GROUP_DESCRIPTION, StudentStatusGen) %>%
  tally() %>%
  dcast(ORIENTATION_GROUP_DESCRIPTION~StudentStatusGen, value.var="n")

otally1159[is.na(otally1159)] <-0
otally1159 <- otally1159 %>%
  mutate(Total = Retained + `Drop/Trans/Stop`) %>%
  mutate(PercentageRetained = paste(as.integer(100*Retained/Total), "%")) %>%
  mutate(ExpectedDrop = Pdrop*Total) %>%
  mutate(ExpectedRetained = Pret*Total) %>%
  mutate(ChiVar = ((Retained-ExpectedRetained)^2/ExpectedRetained) +
  ((`Drop/Trans/Stop`-ExpectedDrop)^2/ExpectedDrop)) %>%
  mutate(pvalue = round(1-(pchisq(ChiVar,1)),3)) %>%
  mutate(SignificantDifference = ifelse(pvalue<.001, "***",
ifelse(pvalue<.01, "**", ifelse(pvalue<.05, "*", "")))) %>%
  dplyr::select(ORIENTATION_GROUP_DESCRIPTION, `Retained`, `Drop/Trans/Stop`,
`Total`, `PercentageRetained`, SignificantDifference)

colnames(otally1159) <- c("Orientation Code", "Retained", "Not Retained",
"Total", "Percent Retained", "Statistical Significance")

knitr::kable(otally1159)

```

Orientation Code	Retained	Not Retained	Total	Percent Retained	Statistical Significance
Des Moines Orientation	52	3	55	94 %	*
Late Orientation Session 1	16	5	21	76 %	
Late Orientation Session 2	19	4	23	82 %	
Late Orientation Session 3	19	4	23	82 %	
Late Orientation Session 4	14	2	16	87 %	
Lynnwood Orientation	50	6	56	89 %	
Moses Lake Orientation	2	1	3	66 %	
None Attended	272	92	364	74 %	**
Online Orientation Session	24	3	27	88 %	
Orientation Session 1	3	0	3	100 %	
Orientation Session 2	5	4	9	55 %	*
Orientation Session 3	5	9	14	35 %	***
Orientation Session 4	8	1	9	88 %	
Orientation Session 5	6	3	9	66 %	
Orientation Session 6	35	14	49	71 %	
Pierce Orientation	26	3	29	89 %	
Transfer Orientation Session	242	40	282	85 %	
Transfer Orientation Session 2	190	23	213	89 %	**
Wenatchee Orientation	3	1	4	75 %	
Yakima Orientation	2	1	3	66 %	

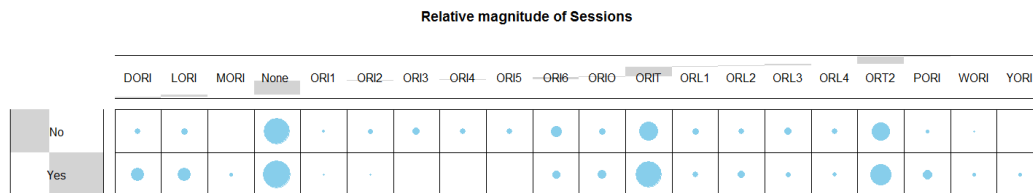
```

#creating crosstabs for categorical tables
#check table integrity
library("gplots")
# 1. convert the data as a table
#dt <- as.table(as.matrix(orient1149))
# 2. Graph

```



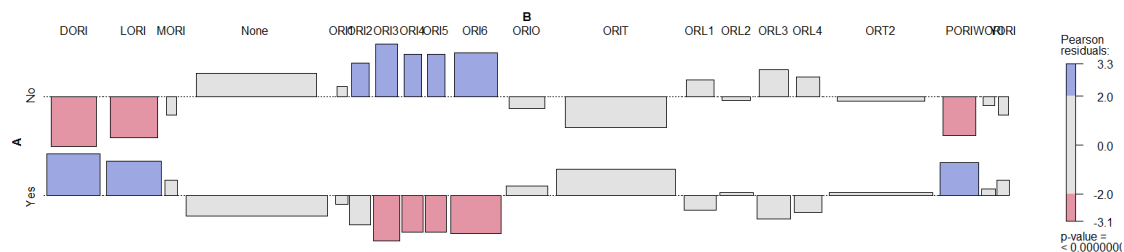
```
orient1159.tab <-
table(orient1159$DTA_FLAG,orient1159$ORIENTATION_GROUP_CODE)
balloonplot(t(orient1159.tab), main = "Relative magnitude of Sessions", xlab
="", ylab="",
          label = FALSE, show.margins = FALSE)
```



```
#library("graphics")
#mosaicplot(orient1159.tab, shade = TRUE, las=2,
            # main = "Insight into Orientation Sessions")
```

```
# install.packages("vcd")
library("vcd")
```

```
?assoc
# plot just a subset of the table
assoc(head(orient1159.tab), shade = TRUE, split_vertical = FALSE, las=2)
```



```
chisq.test(orient1159.tab)
```

```
## Warning in chisq.test(orient1159.tab): Chi-squared approximation may be
## incorrect
```

```
##
## Pearson's Chi-squared test
##
## data: orient1159.tab
## X-squared = 130.66, df = 19, p-value < 0.00000000000000022
```

Fall 2016 Orientation Sessions and Transfer students Persistence statistical significance

```
#transform raw transfer retention charatersitics dataset into a data.table
format in order to make it easier to change input by reference
```

```

setDT(transfer1169)[, paste0("ACADEMIC_TERM_YEAR_NAME", 1:2) :=
tstrsplit(ACADEMIC_TERM_YEAR_NAME, " ")] #split academic year name into just
academic year and actual Year.

#create new variable in our data.table to accomodate split

transfer1169$ACADEMIC_TERM_YEAR_NAME2 <-
as.numeric(transfer1169$ACADEMIC_TERM_YEAR_NAME2)

#academic year name only variable added

transfer1169$ActualYear <- ifelse(grepl("Spring|Winter|Summer",
transfer1169$ACADEMIC_TERM_YEAR_NAME),
transfer1169$ACADEMIC_TERM_YEAR_NAME2 - 1,
transfer1169$ACADEMIC_TERM_YEAR_NAME2) #year numeric only variable added

#there are duplicate records for each student ID to indicate their progress
after they enrolled at the university
#now we will attempt to combine duplicate record to figure out who dropped or
retained

#here you see how %>% makes the code easier to read

#Find Maximum Quarters Enrolled
maxterm1169 <- transfer1169 %>%
  group_by(STUDENT_ID) %>%
  filter(TOTAL_QUARTERS_ENROLLED == max(TOTAL_QUARTERS_ENROLLED)) %>%
  mutate(MaxQuarters = TOTAL_QUARTERS_ENROLLED) %>%
  dplyr::select(STUDENT_ID, MaxQuarters)

#Find Terms Enrolled in First Year
minyr1169 <- transfer1169 %>%
  group_by(STUDENT_ID) %>%
  slice(which.min(ActualYear)) %>%
  mutate(MinYear = ActualYear) %>%
  dplyr::select(STUDENT_ID, MinYear)

minyr21169 <- merge(transfer1169, minyr1169, by="STUDENT_ID", all.x=TRUE)

termsinminyear1169 <- minyr21169 %>%
  filter(ACADEMIC_TERM_YEAR_NAME1 != "Summer") %>%
  group_by(STUDENT_ID) %>%
  filter(ActualYear == MinYear) %>%
  tally()
setnames(termsinminyear1169, "n", "FYTermsRetained")

#Did Student Come Back for Second Fall?
WhichFallsRetained1169 <- transfer1169 %>%

```

```

filter(ACADEMIC_TERM_YEAR_NAME1 == "Fall") %>%
group_by(STUDENT_ID) %>%
slice(which.min(ACADEMIC_TERM_CODE)) %>%
mutate(FirstFall = ACADEMIC_TERM_CODE) %>%
mutate(SecondFall = FirstFall +10)%>%
dplyr::select(STUDENT_ID, SecondFall)

hadsecondfall1169 <- merge(transfer1169, WhichFallsRetained1169,
by="STUDENT_ID", all.x=TRUE)
hadsecondfall1169 <- hadsecondfall1169 %>%
  group_by(STUDENT_ID) %>%
  mutate(SecondFallHappened = ifelse(SecondFall == ACADEMIC_TERM_CODE, "Yes",
"No")) %>%
  filter(SecondFallHappened == "Yes") %>%
  dplyr::select(STUDENT_ID, SecondFallHappened)

#Now Merge ALL These, Find Student Status
findingretained1169 <- merge(maxterm1169, termsinminyear1169,
by="STUDENT_ID", all.x=TRUE)
findingretained1169 <- merge(findingretained1169, hadsecondfall1169,
by="STUDENT_ID", all.x=TRUE)
findingretained1169 <- findingretained1169 %>%
  mutate(StudentStatus = ifelse(is.na(SecondFallHappened) & MaxQuarters <=3,
"Drop", ifelse(is.na(SecondFallHappened) & MaxQuarters >3, "Stop-Out",
ifelse(SecondFallHappened == "Yes" & FYTermsRetained <3, "Stop-Out",
"Retained")))) %>%
  mutate(StudentStatusGen = ifelse(is.na(SecondFallHappened) & MaxQuarters
<=3, "Drop/Trans/Stop", ifelse(is.na(SecondFallHappened) & MaxQuarters >3,
"Drop/Trans/Stop", ifelse(SecondFallHappened == "Yes" & FYTermsRetained <3,
"Retained", "Retained"))))

#here we join the retained variables to the original dataset

persitencetransfer1169 <- merge(transfer1169, findingretained1169,
by="STUDENT_ID", all.x=TRUE)

#Now, take only the first term for the rest of this study to ignore
duplicates

persitencetransfer1169 <- persitencetransfer1169 %>%
  filter(TOTAL_QUARTERS_ENROLLED == 1) %>% filter(ACADEMIC_TERM_CODE ==
"1169")

#data integrity check - open this document to see the retained info properly
joined to original
write.csv(persitencetransfer1169, "retainedJoinRet1169.csv")

#here we add in orientation info

```

```
persitencetransfer1169orient <- merge(persitencetransfer1169, orient1169,
by=c("STUDENT_ID", "CURRENT_ADMIT_TERM_CODE", "CURRENT_ADMIT_TYPE_CODE"),
all.x=TRUE)
```

#data integrity check - open this document to see that new orientation info properly joined to retainedRet

```
write.csv(persitencetransfer1169orient, "orientJoinRet1169.csv")
```

#Fall 2015 Transfer Cohort chiSquare analysis

#here we calculate statistic

#here we make data readable

```
OverallPercent1169 <- persitencetransfer1169orient %>%
group_by(StudentStatusGen) %>% tally() %>% mutate(sum = sum(n),
percent=n/sum) %>% dplyr::select(StudentStatusGen, percent)
Pret1169 <- as.numeric(as.character(OverallPercent[2,2]))
Pdrop1169<- as.numeric(as.character(OverallPercent[1,2]))
```

#here we aggregate

```
otally1169 <- persitencetransfer1169orient %>%
  filter(ActualYear %in% 2016) %>%
  group_by(ORIENTATION_GROUP_DESCRIPTION, StudentStatusGen) %>%
  tally() %>%
  dcast(ORIENTATION_GROUP_DESCRIPTION~StudentStatusGen, value.var="n")
```

#integrity check -

```
otally1169[is.na(otally1169)] <-0
```

#then continue with aggregation

```
otally1169[is.na(otally1169)] <-0
otally1169 <- otally1169 %>%
  mutate(Total = Retained + `Drop/Trans/Stop`) %>%
  mutate(PercentageRetained = paste(as.integer(100*Retained/Total), "%")) %>%
  mutate(ExpectedDrop = Pdrop*Total) %>%
  mutate(ExpectedRetained = Pret*Total) %>%
  mutate(ChiVar = ((Retained-ExpectedRetained)^2/ExpectedRetained) +
  ((`Drop/Trans/Stop`-ExpectedDrop)^2/ExpectedDrop)) %>%
  mutate(pvalue = round(1-(pchisq(ChiVar,1)),3)) %>%
  mutate(SignificantDifference = ifelse(pvalue<.001, "***",
ifelse(pvalue<.01, "**", ifelse(pvalue<.05, "*", "")))) %>%
  dplyr::select(ORIENTATION_GROUP_DESCRIPTION, `Retained`, `Drop/Trans/Stop`,
`Total`, `PercentageRetained`, SignificantDifference)

colnames(otally1169) <- c("Orientation Code", "Retained", "Not Retained",
```

"Total", "Percent Retained", "Statistical Significance")

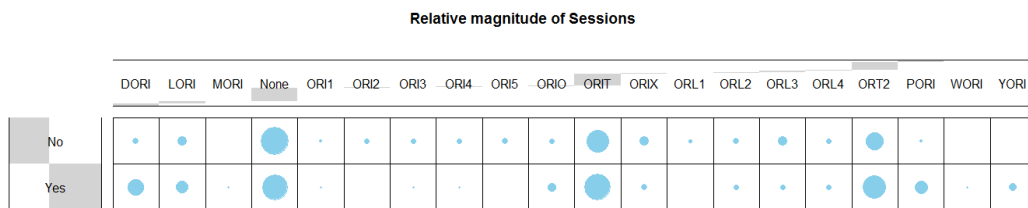
knitr::kable(otally1169)

Orientation Code	Retained	Not Retained	Total	Percent Retained	Statistical Significance
Des Moines Orientation	68	6	74	91 %	*
Late Orientation Session 1	4	0	4	100 %	
Late Orientation Session 2	15	4	19	78 %	
Late Orientation Session 3	24	8	32	75 %	
Late Orientation Session 4	11	3	14	78 %	
Lynnwood Orientation	47	7	54	87 %	
Moses Lake Orientation	1	0	1	100 %	
None Attended	250	70	320	78 %	
Online Orientation Session	13	0	13	100 %	
ORI Approved Unable to Attend	24	6	30	80 %	
Orientation Session 1	3	1	4	75 %	
Orientation Session 2	6	1	7	85 %	
Orientation Session 3	7	1	8	87 %	
Orientation Session 4	7	1	8	87 %	
Orientation Session 5	8	3	11	72 %	
Pierce Orientation	35	11	46	76 %	
Transfer Orientation Session	262	43	305	85 %	*
Transfer Orientation Session 2	192	31	223	86 %	
Wenatchee Orientation	0	1	1	0 %	*
Yakima Orientation	14	0	14	100 %	

```

#creating crosstabs for categorical tables
#check table integrity
library("gplots")
# 1. convert the data as a table
#dt <- as.table(as.matrix(orient1149))
# 2. Graph
orient1169.tab <-
table(orient1169$DTA_FLAG,orient1169$ORIENTATION_GROUP_CODE)
balloonplot(t(orient1169.tab), main = "Relative magnitude of Sessions", xlab
="", ylab="",
           label = FALSE, show.margins = FALSE)

```



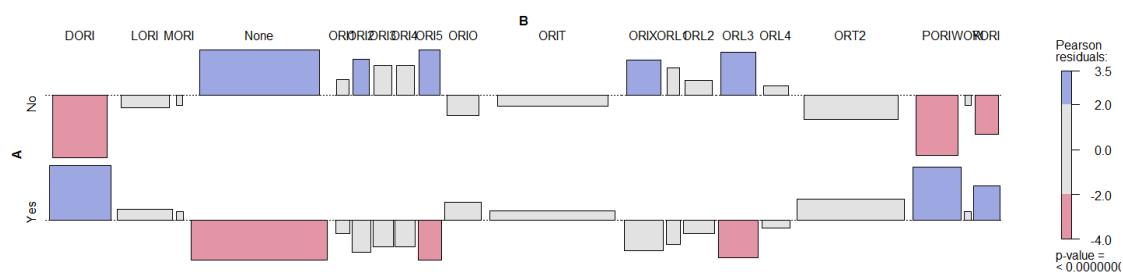
```

#library("graphics")
#mosaicplot(orient1169.tab, shade = TRUE, las=2,
#           main = "Insight into Orientation Sessions")

# install.packages("vcd")
library("vcd")

?assoc
# plot just a subset of the table
assoc(head(orient1169.tab), shade = TRUE, split_vertical = FALSE, las=2)

```



```

chisq.test(orient1169.tab)

## Warning in chisq.test(orient1169.tab): Chi-squared approximation may be
## incorrect

##
## Pearson's Chi-squared test
##

```

```
## data: orient1169.tab
## X-squared = 155.58, df = 19, p-value < 0.00000000000000022
```

Conclusion

This study equips orientation professionals here at the university with facts about the program for transfer students. In this study the 3 transfer student cohorts were analyzed (Fall 2014, Fall 2015 and Fall 2016).

In all the cohorts, students that did not attend any orientation session showed a strong significant positive relationship to returning to the university for their second year. At least 70 percent of these transfer students that choose not to attend any session were retained by the university.

"Transfer Orientation session" and "Transfer Orientation Session 2" students showed strong persistence as well. With at least 80 percent retention of the students after their first year.

I did not find any strong relationship between the other orientation sessions. This is in part attributed to the low number of attendants for those sessions.

Orientation is still valuable to transfer students but clearly more transfer students are skipping this program. These students are more experienced and may benefit from a robust orientation program that focuses on how they can quickly get into their classes or find cheaper housing. If the university dedicates more resources into orientation, for example establishing a department for orientation programming.

The strategy of the two high performing sessions should be copied and made the standard for all sessions offered to transfer students going forward. This would increase the attendance of orientation at the university for transfer students.

One way to improve this study is to analyze exit surveys from the less performing orientation sessions. It would be interesting to see if there is any relationship between a student's persistence at the university and their orientation experience.

Acknowledgment

The author thanks Dr Wetherell for her help in writing this paper.

References

De los Santos, A. G., & Wright, I. (1989). Community college and university student transfers. Educational Record, 79 (3/4), 82-84.

Mayhew, Matthew & Vanderlinden, Kim & Kyung Kim, Eun. (2010). A MultiLevel Assessment of the Impact of Orientation Programs on Student Learning. Research in Higher Education - RES HIGH EDUC. 51. 320-345.