# Lesson 12

Clark

Let's consider data on the career choice of some high school students

```
library(faraway)
library(tidyverse)
library(nnet)

data(hsb)
```

The purpose of this study was to determine which factors are related to the choice of the type of program: academic, vocational, or general, that students pursue in high school

Should this data be considered categorical or ordinal?

We can explore the data in a few different ways:

```
#qplot(hsb$prog, geom = "bar")
#pairs(hsb)
#table(hsb$gender, hsb$prog)
#table(hsb$ses, hsb$prog)
#pairs(prog ~ gender + write + math, data = hsb)
```

However, most of these are less than satisfying. We can build out a baseline-categories multinomial model. Perhaps we want `general` to be the baseline

```
hsb$prog <- relevel(hsb$prog, ref = "general")
mmod <- multinom(prog ~ gender+race+ses+schtyp+read+
                   write, hsb)
```

```
## # weights:  33 (20 variable)
## initial  value 219.722458
## iter  10 value 169.211065
## iter  20 value 168.041637
## final  value 168.040906
## converged
```

```
confint(mmod)
```

```
## , , academic
##
##                     2.5 %      97.5 %
```

```
## (Intercept)  -5.731594494  0.55231712
## gendermale   -0.972778935  0.67848598
## raceasian    -2.884106035  0.80156234
## racehispanic -1.187914325  2.19357841
## racewhite    -1.983447301  0.65909028
## seslow       -2.204991673 -0.01585211
## sesmiddle    -1.582809658  0.29486970
## schtyppublic -1.642506923  0.46932013
## read          0.004583181  0.10538707
## write        -0.014653818  0.09515066
##
## , , vocation
##
##                   2.5 %     97.5 %
## (Intercept)  -1.5423506 5.93079539
## gendermale   -1.4103115 0.50956566
## raceasian    -4.1946757 1.09913620
## racehispanic -1.4272363 2.04718788
## racewhite    -1.4770074 1.33298451
## seslow       -1.9116455 0.79557658
## sesmiddle    -0.6040696 1.76602037
## schtyppublic -0.3177766 3.09434365
## read         -0.0839191 0.03516436
## write        -0.1024226 0.01871853
```

What do we see here?

P values can be computed manually

```
z_stats <- summary(mmod)$coefficients / summary(mmod)$standard.errors
p_vals <- (1-pnorm(abs(z_stats), 0, 1))*2
p_vals
```

```
##          (Intercept) gendermale raceasian racehispanic racewhite     seslow
## academic   0.1062184  0.7268565 0.2680974    0.5599620 0.3259656 0.04677309
## vocation   0.2497541  0.3578062 0.2517610    0.7265473 0.9199826 0.41908644
##          sesmiddle schtyppublic       read     write
## academic 0.1788253    0.2762324 0.03250143 0.1507649
## vocation 0.3366105    0.1107359 0.42229754 0.1756525
```

Perhaps we want to compare to a simpler model that does not include `race` and `gender`

```
smaller_mod <- multinom(prog ~ ses+schtyp+read+
                  write, hsb)
```

```
## # weights:  21 (12 variable)
## initial  value 219.722458
```

```
## iter  10 value 172.749264
## final  value 171.323596
## converged
```

```
anova(smaller_mod,mmod,test="Chisq")
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: prog
##                                          Model Resid. df Resid. Dev   Test    Df
## 1                 ses + schtyp + read + write       388   342.6472
## 2 gender + race + ses + schtyp + read + write       380   336.0818 1 vs 2     8
##   LR stat.   Pr(Chi)
## 1
## 2  6.56538 0.5841652
```

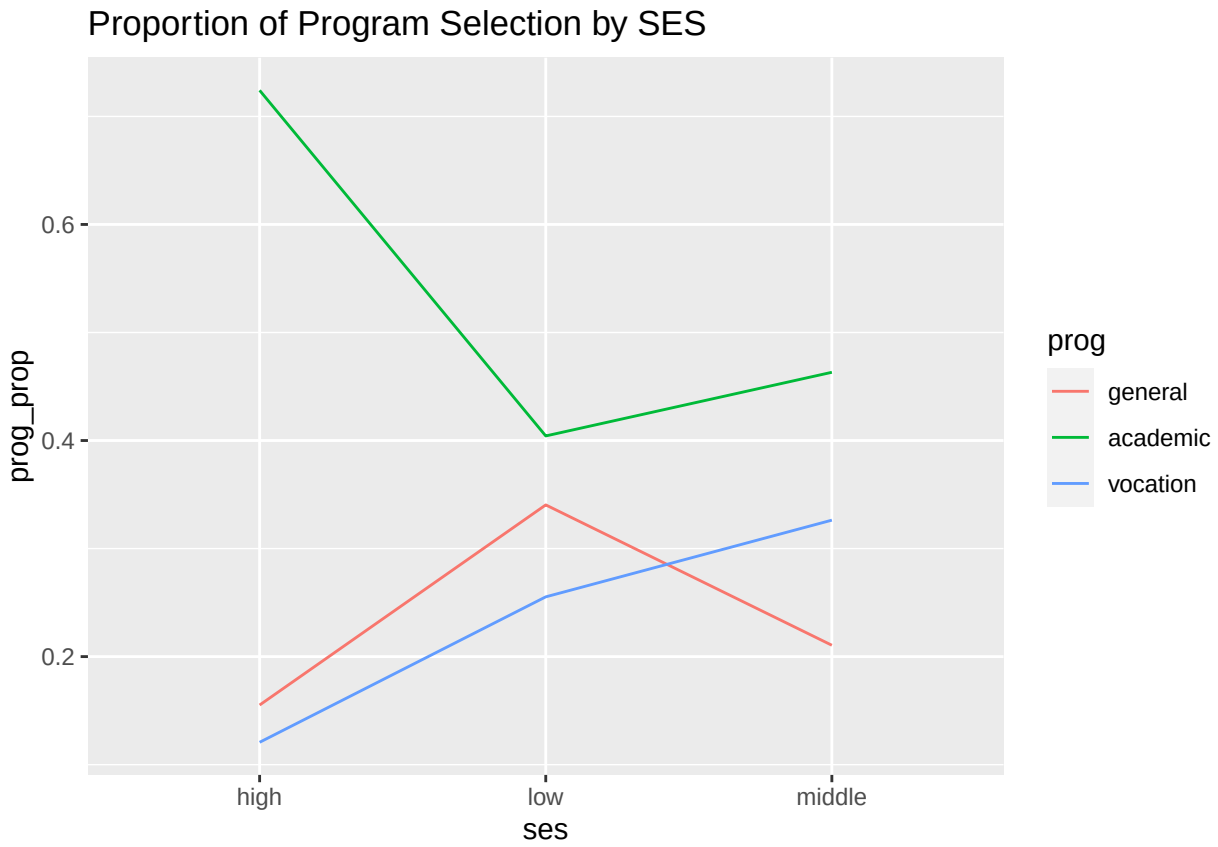
What does this suggest?

Let's dig into this a bit more. If we just look at socio-economic status we can turn our data into grouped data.

```
hsb_ses <- hsb %>%
  group_by(ses, prog) %>%
  summarise(count=n()) %>%
  mutate(ses_tot = sum(count),
         prog_prop = count/ses_tot)
```

```
## `summarise()` has grouped output by 'ses'. You can override using the `.groups`
## argument.
```

```
ggplot(hsb_ses, aes(x = ses, y = prog_prop,
                    group = prog, color = prog))+
  geom_line() +
  ggtitle("Proportion of Program Selection by SES")
```




With the data structured like this, if our primary aim is to analyze the impact of ses on program, we can also view the data as a contingency table

```
ct <- xtabs(count~ses+prog,data=hsb_ses)
ct
```

```
##         prog
## ses      general academic vocation
##   high         9       42        7
##   low         16       19       12
##   middle      20       44       31
```

Nowm one of the benefits of using a contingency table is it allows us to calculate one of the more famous statistical tests. Pearson's $\chi^2$ test. The construct of this test, which you may have seen before, is relatively straight forward.

```r
summary(ct)
```

```
## Call: xtabs(formula = count ~ ses + prog, data = hsb_ses)
## Number of cases in table: 200
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 16.604, df = 4, p-value = 0.002307
```

The degrees of freedom are the number of rows minus one times the number of columns minus one.

This analysis assumes that we set out to sample 200 students (which is probably a good assumption in this case). However, it's also possible that we just recorded every student who stopped into our office over the course of a month. In this case, we can't think about our data coming from a multinomial distribution (with a fixed number of trials). Rather we would assume that

From here we can build out a Poisson regression model (more to follow next lesson)

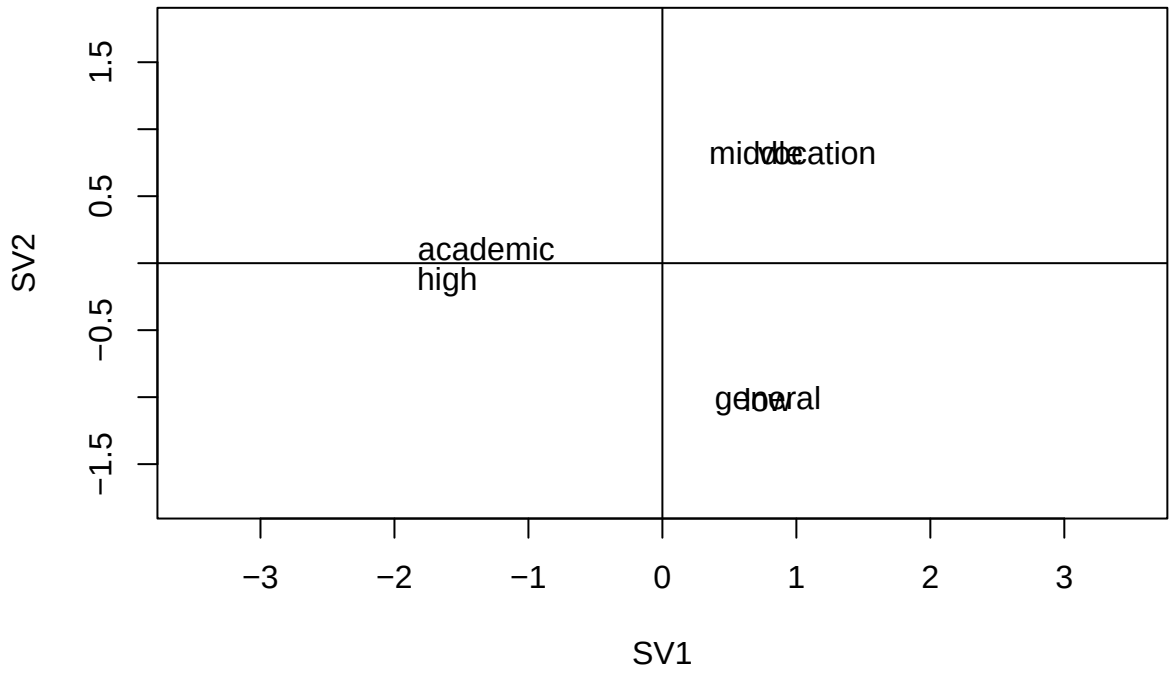
```r
test <- glm(count~ses+prog,data=hsb_ses,family=poisson)
summary(test)
```

```
##
## Call:
## glm(formula = count ~ ses + prog, family = poisson, data = hsb_ses)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.5688     0.1856  13.837  < 2e-16 ***
## seslow        -0.2103     0.1963  -1.072  0.28394
## sesmiddle      0.4934     0.1666   2.961  0.00306 **
## progacademic   0.8473     0.1782   4.755 1.98e-06 ***
## progvocation   0.1054     0.2055   0.513  0.60812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66.314  on 8  degrees of freedom
## Residual deviance: 16.783  on 4  degrees of freedom
## AIC: 69.717
##
## Number of Fisher Scoring iterations: 4
```

However, really all this tells us is that we have more middle income than high income and more `general` and `vocation` than academic. It does not tell us the relationship between ses and program.

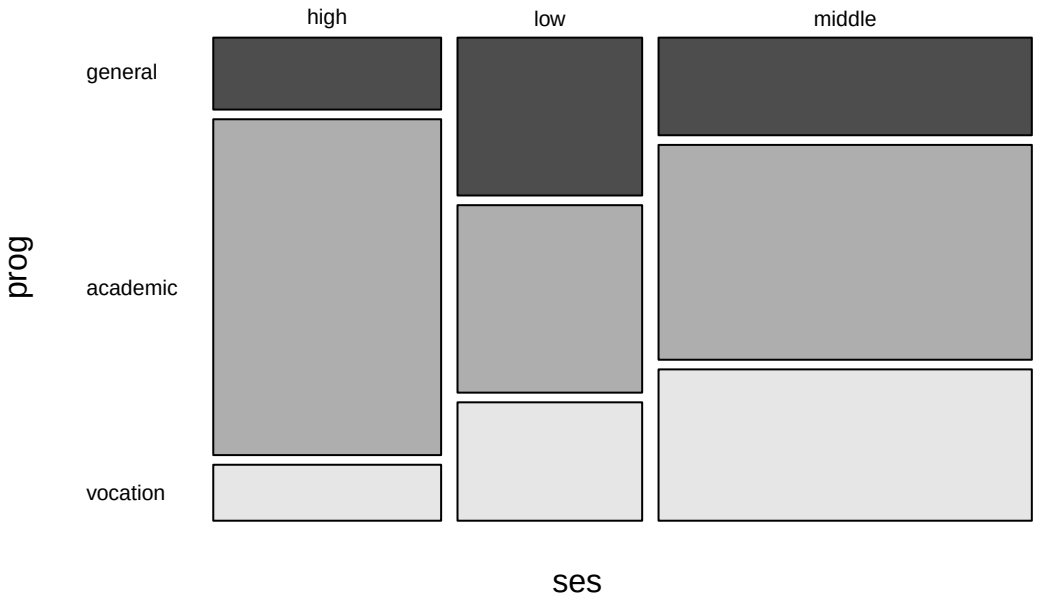
To do this, we can conduct correspondence analysis (CA). This is going to feel a lot like PCA, but it is for count data whereas PCA is for continuous data. Without getting too much into the weeds on this, correspondence analysis uses a $\chi^2$ distance between each of the observations. PCA works by looking at the Mahalonobis distance (sort of, I don't want to get too much into the weeds here). But, if you recall to compute PCA you compute the SVD of the X matrix. For CA you compute the SVD of the Pearson residuals from the Poisson regression model

```
z<-xtabs(residuals(test,type="pearson")~ses+prog,hsb_ses)

blah <-svd(z,3,3)

leftsv <- blah$u %*% diag(sqrt(blah$d[1:3]))

rightsv <- blah$v %*% diag(sqrt(blah$d[1:3]))

l1 <- 1.1*max(abs(rightsv),abs(leftsv))

plot(rbind(leftsv,rightsv),asp=1,xlim=c(-l1,l1),
     ylim=c(-l1,l1),xlab="SV1",ylab="SV2",type="n")
abline(h=0,v=0)
text(leftsv,dimnames(z)[[1]])
text(rightsv,dimnames(z)[[2]])
```




Another way the relationships (as a form of descriptive analytics) to view this is through a mosaic plot

```
mosaicplot(ct,color=TRUE,main=NULL,las = 1)
```

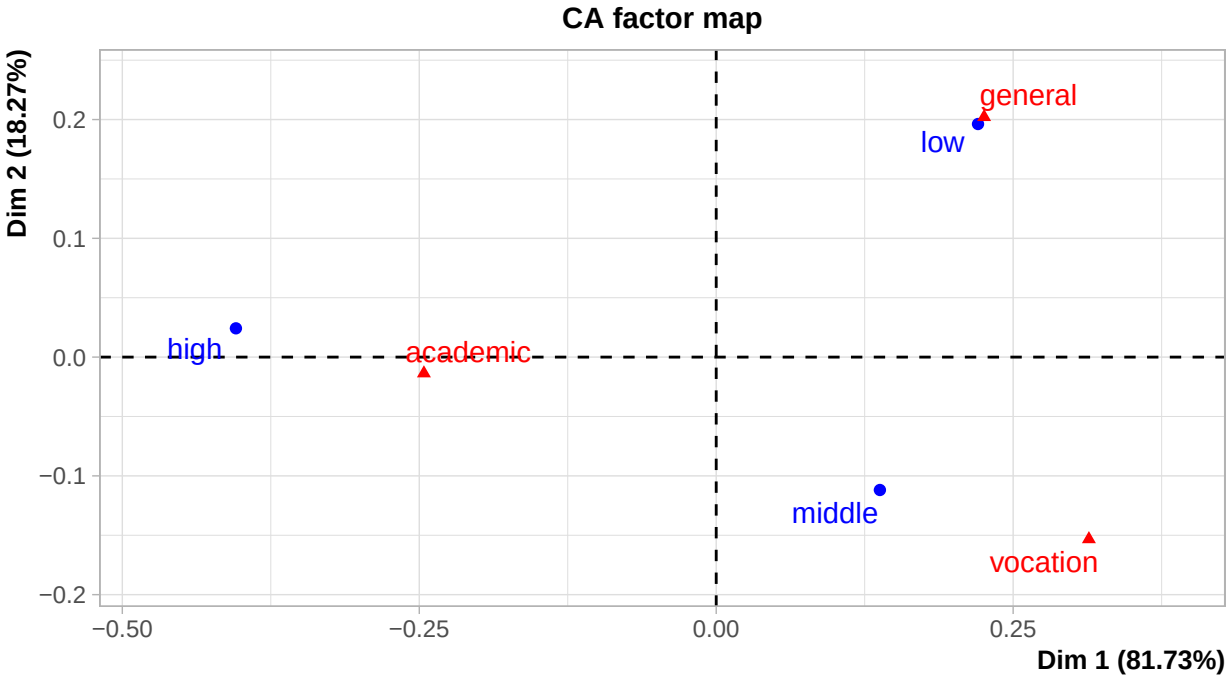



Here we can see what the CA is giving us. Apperantly there is an R package `FactoMineR` that has a function called `CA` that does the correspondence analysis. It's kind of nice

```
library(FactoMineR)
```

```
## Warning: package 'FactoMineR' was built under R version 4.3.2
```

```
CA(ct)
```




```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  3  categories; the column variable has 3 categories
## The chi square of independence between the two variables is equal to 16.60444 (p-value =  0.00230663
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```