MA678 homework 05

Multinomial Regression

Your Name

September 2, 2017

Multinomial logit:

Using the individual-level survey data from the 2000 National Election Study (data in folder nes), predict party identification (which is on a 7-point scale) using ideology and demographics with an ordered multinomial logit model.

1. Summarize the parameter estimates numerically and also graphically.

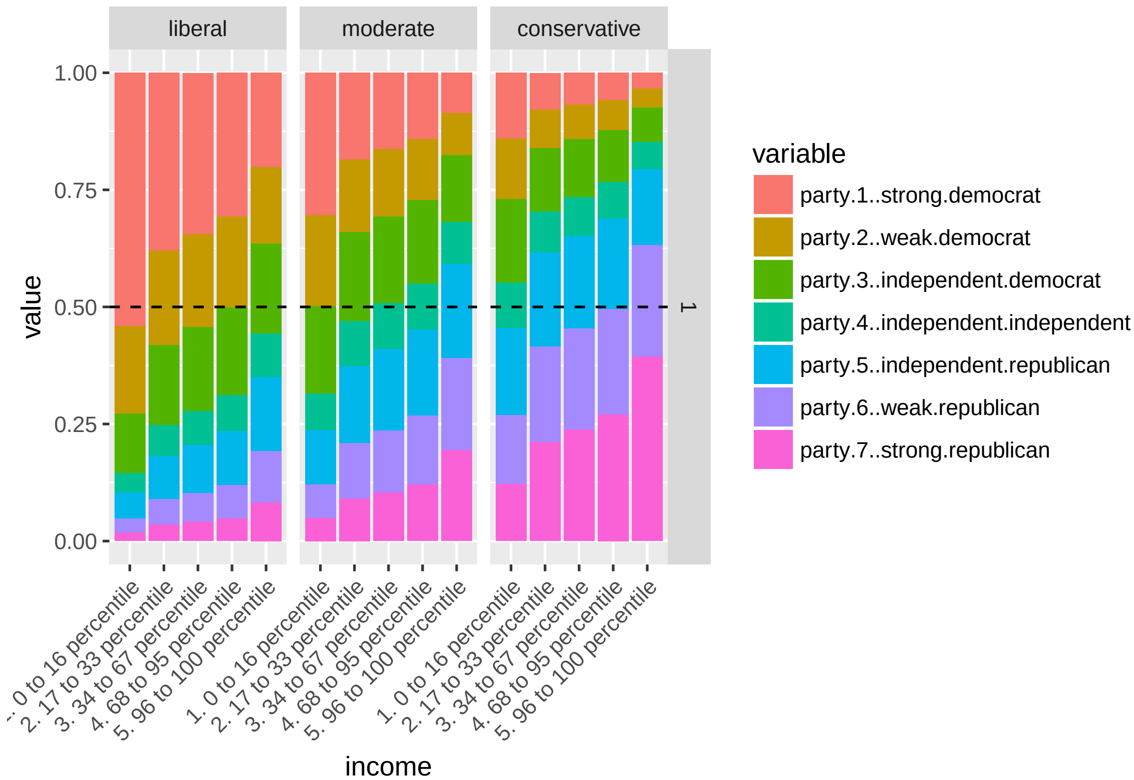
```
fit_polr<-polr(ordered(partyid7)~ideo+female+white+income, data=nes_data_comp)
# resd<-as.data.frame(cbind(nes_data_comp[,list(ideo)],fitted(fit_polr)))
# ggplot(melt(resd,id.var="ideo"))+
#   geom_bar(position = "fill",stat="identity")+
#   aes(x=ideo,y=value,fill=variable)
# resd<-as.data.frame(cbind(nes_data_comp[,list(income)],fitted(fit_polr)))
# ggplot(melt(resd,id.var="income"))+
#   geom_bar(position = "fill",stat="identity")+
#   aes(x=income,y=value,fill=variable)
display(fit_polr)
```

```
##
## Re-fitting to get Hessian
```

```
## polr(formula = ordered(partyid7) ~ ideo + female + white + income,
##     data = nes_data_comp)
##                                                        coef.est coef.se
## ideomoderate                                            0.99     0.33
## ideoconservative                                        1.98     0.18
## female                                                 -0.19     0.16
## white                                                   0.67     0.18
## income2. 17 to 33 percentile                            0.66     0.28
## income3. 34 to 67 percentile                            0.81     0.27
## income4. 68 to 95 percentile                            0.98     0.27
## income5. 96 to 100 percentile                           1.54     0.39
## 1. strong democrat|2. weak democrat                     0.83     0.31
## 2. weak democrat|3. independent-democrat                1.65     0.31
## 3. independent-democrat|4. independent-independent      2.43     0.32
## 4. independent-independent|5. independent-republican    2.83     0.33
## 5. independent-republican|6. weak republican            3.64     0.34
## 6. weak republican|7. strong republican                 4.62     0.36
## ---
## n = 557, k = 14 (including 6 intercepts)
## residual deviance = 1936.2, null deviance is not computed by polr
```

```
predx<-expand.grid(income=unique(nes_data_comp$income),
                   white=1,female=0,ideo=unique(nes_data_comp$ideo))

predy<-predict(fit_polr,newdata=predx,type="prob")
```

```r
resd<-data.frame(predx[,c("income","ideo","white")],party=predy)
ggplot(melt(resd,id.var=c("income","ideo","white")))+
  geom_bar(position = "fill",stat="identity")+
  aes(x=income,y=value,fill=variable)+
  facet_grid(white~ideo)+geom_hline(yintercept=0.5,lty=2)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



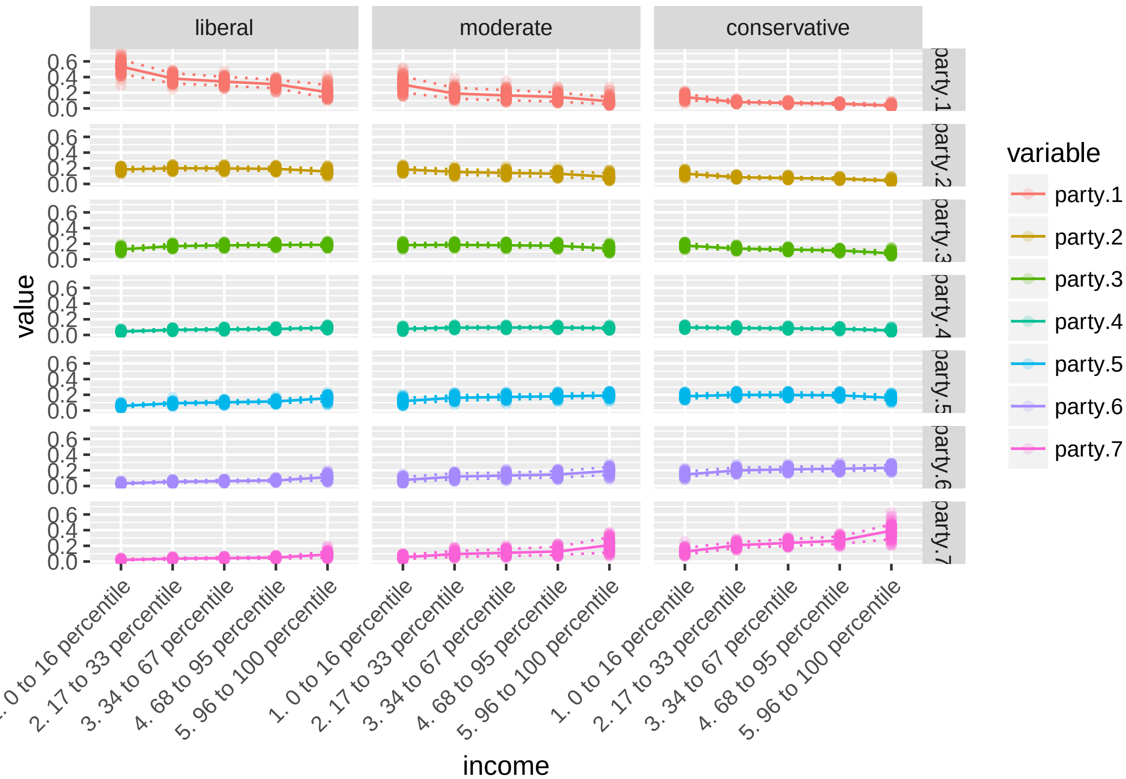
The above result is a point estimate and lacks information regarding the uncertainty of our estimates. We can add the uncertainty in the parameter estimate using the sim function.

```r
simfit<-sim(fit_polr)
```

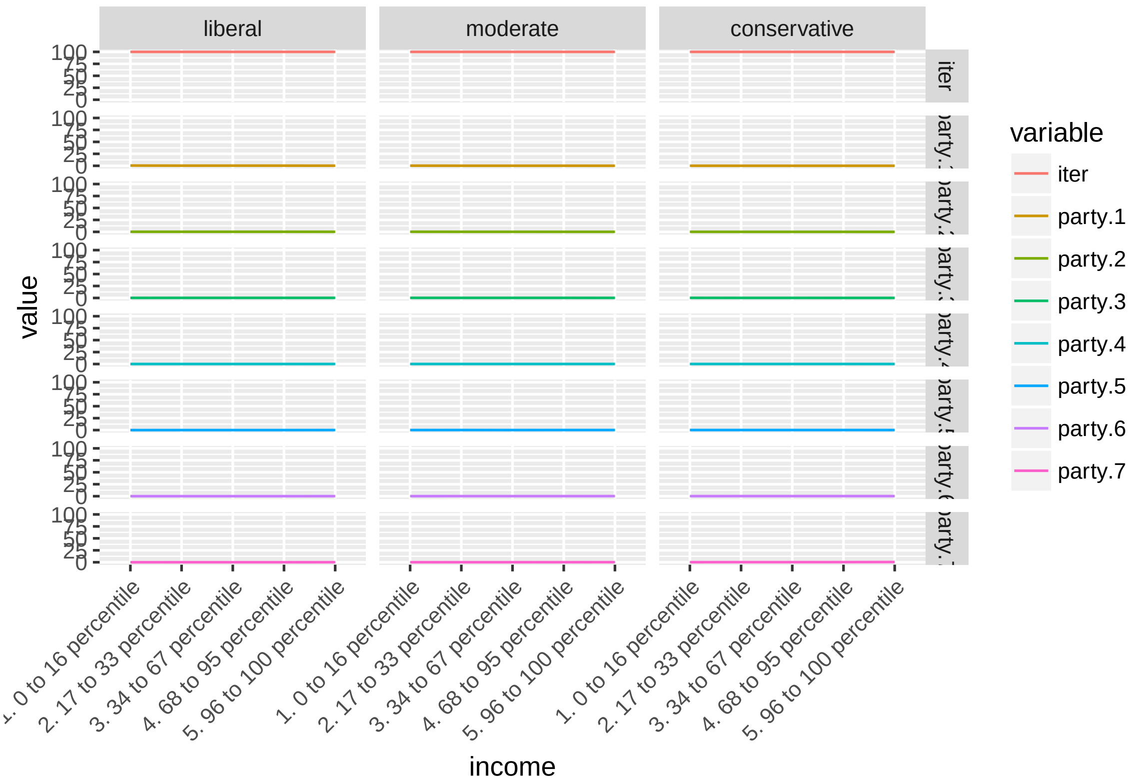
```
##
## Re-fitting to get Hessian
```

```r
xx<-model.matrix(~-1+ideo+female+white+income,data=predx)
xb<-xx[,colnames(simfit@coef)]%*%t(simfit@coef)

reslist<-vector("list",100)
for(iter in 1:100){
  pa<-invlogit(outer(-xb[,iter],simfit@zeta[iter,],"+"))
  pp<-cbind( pa[,1], pa[,2]-pa[,1],pa[,3]-pa[,2],pa[,4]-pa[,3], pa[,5]-pa[,4],pa[,6]-pa[,5],1-pa[,6])

  resd<-data.frame(predx[,c("income","ideo","white")],iter=iter,party=pp)
  reslist[[iter]]<-resd
}
ggplot(melt(rbindlist(reslist),
            id.var=c("income","ideo","white","iter")))+
geom_point(alpha=0.2)+
  aes(x=income,y=value,group=iter,color=variable)+
```

```
  facet_grid(variable~ideo)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+stat_summary(fun.y=mean,  geom="line", aes(g
  stat_summary(fun.y=function(x)quantile(x,0.1),  geom="line", lty=3, aes(group = 1))+
  stat_summary(fun.y=function(x)quantile(x,0.9),  geom="line", lty=3, aes(group = 1))
```



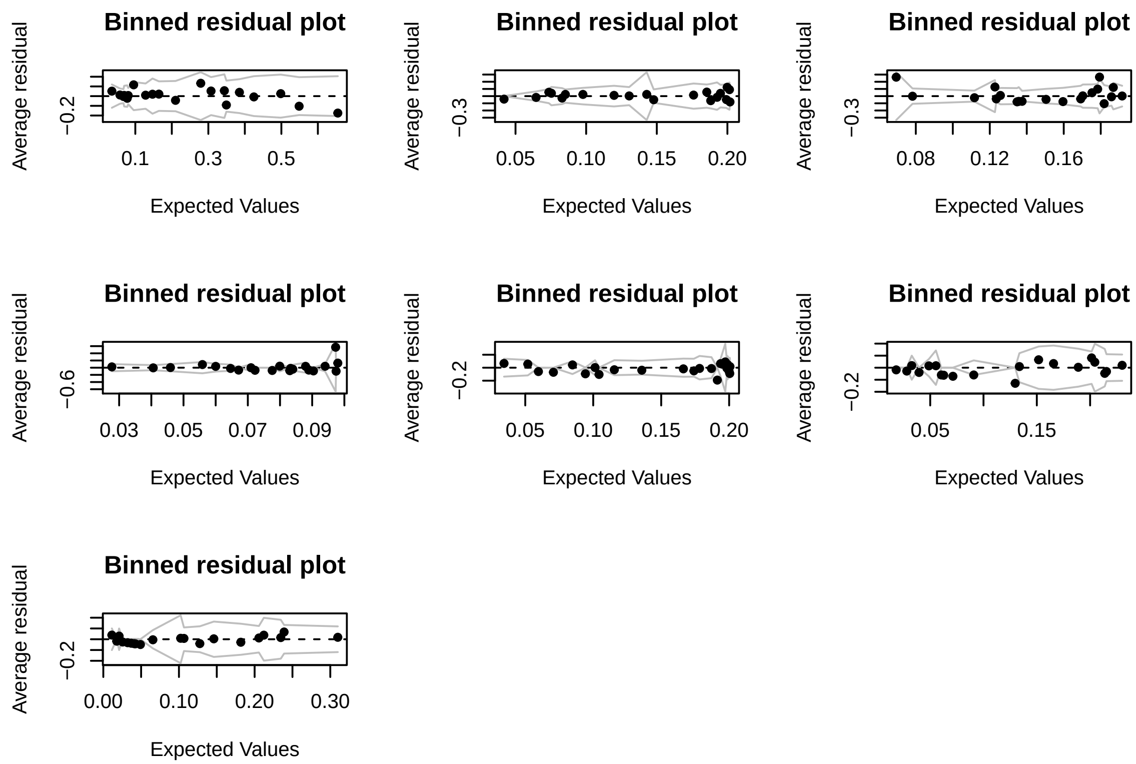

```
ggplot(melt(resd,id.var=c("income","ideo","white")))+
  geom_line()+
  aes(x=income,y=value,group=variable,color=variable)+
  facet_grid(variable~ideo)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



2. Explain the results from the fitted model.

3. Use a binned residual plot to assess the fit of the model.

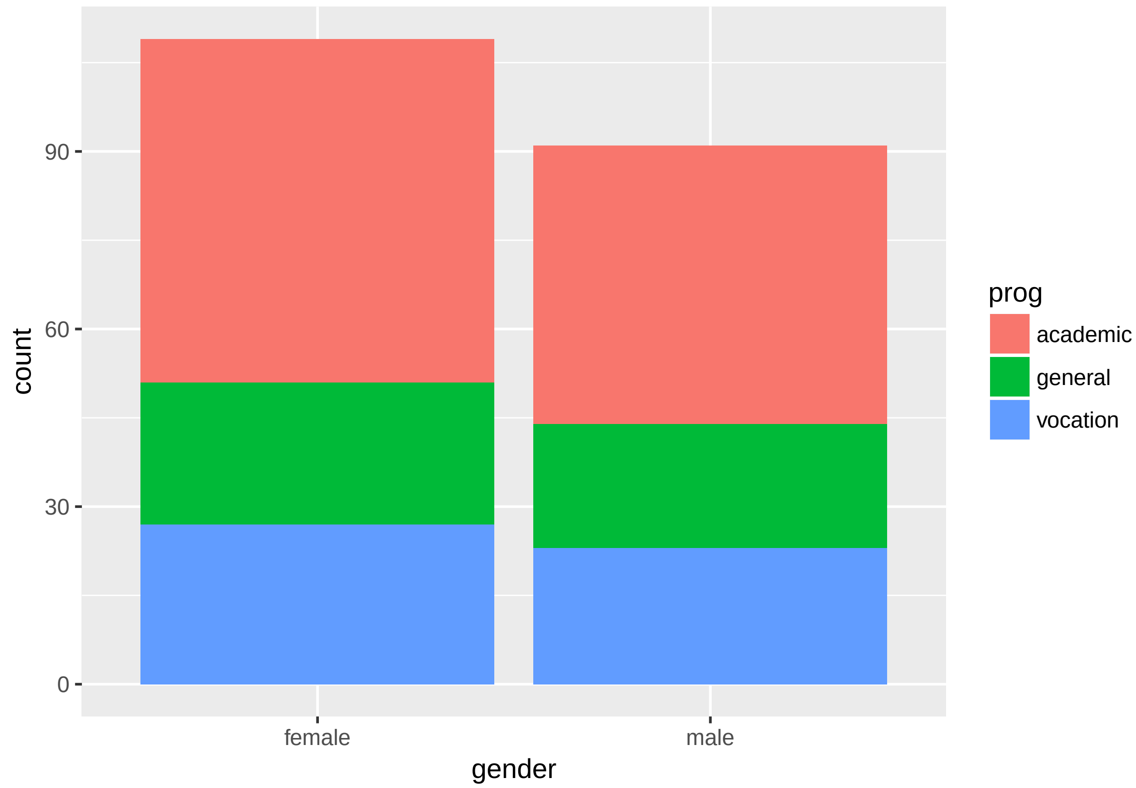
```
obsmat  <-model.matrix(~partyid7-1,data=nes_data_comp)
resdimat<-obsmat-fitted(fit_polr)
par(mfrow=c(3,3))
binnedplot(fitted(fit_polr)[,1],resdimat[,1])
binnedplot(fitted(fit_polr)[,2],resdimat[,2])
binnedplot(fitted(fit_polr)[,3],resdimat[,3])
binnedplot(fitted(fit_polr)[,4],resdimat[,4])
binnedplot(fitted(fit_polr)[,5],resdimat[,5])
binnedplot(fitted(fit_polr)[,6],resdimat[,6])
binnedplot(fitted(fit_polr)[,7],resdimat[,7])
```



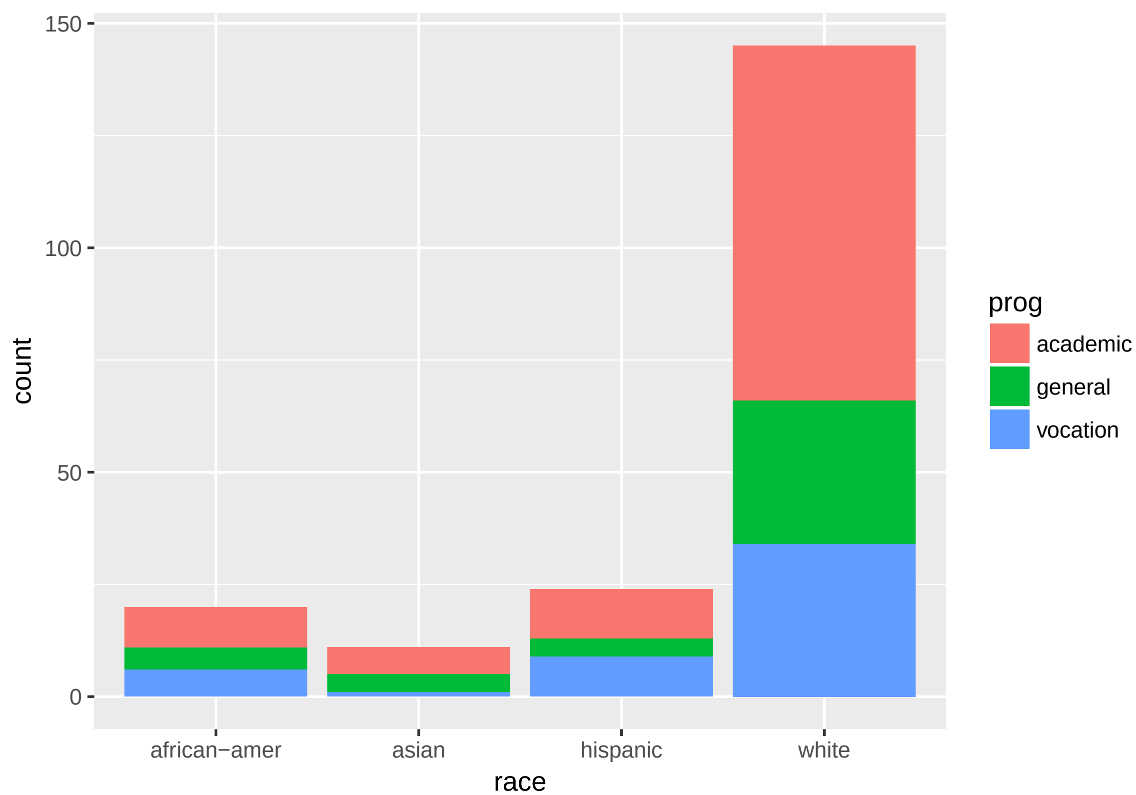
High School and Beyond

The hsb data was collected as a subset of the High School and Beyond study conducted by the National Education Longitudinal Studies program of the National Center for Education Statistics. The variables are gender; race; socioeconomic status; school type; chosen high school program type; scores on reading, writing, math, science, and social studies. We want to determine which factors are related to the choice of the type of program – academic, vocational, or general – that the students pursue in high school. The response is multinomial with three levels.

```
data(hsb);?hsb
ggplot(hsb)+geom_bar()+aes(x=gender,fill=prog)
```




```
ggplot(hsb)+geom_bar()+aes(x=race,fill=prog)
```




```
knitr::kable(hsb %>%
  group_by(gender,prog ) %>%
```

```r
  summarise_at(.funs=funs(mean(.)),.vars=vars("read", "write", "math" ,"science", "socst") ),digits=2 )
```

gender	prog	read	write	math	science	socst
female	academic	56.07	57.59	56.41	52.38	56.86
female	general	46.96	53.25	49.88	50.42	50.42
female	vocation	46.67	50.96	46.00	47.33	46.67
male	academic	56.28	54.62	57.13	55.55	56.49
male	general	52.95	49.14	50.19	54.76	50.81
male	vocation	45.65	41.83	46.91	47.09	43.09

```r
mytable <- xtabs(~gender+race+prog, data=hsb)
(ftable(mytable)) # print table
```

```
##                        prog academic general vocation
## gender race
## female african-amer                6       3        4
##        asian                       4       3        1
##        hispanic                    5       3        3
##        white                      43      15       19
## male   african-amer                3       2        2
##        asian                       2       1        0
##        hispanic                    6       1        6
##        white                      36      17       15
```

1. Fit a trinomial response model with the other relevant variables as predictors (untransformed).

```r
mmod0 <- multinom(prog ~ gender + race + ses + schtyp + read + write + math +
                          science + socst, hsb, trace = FALSE)
summary(mmod0)
```

```
## Call:
## multinom(formula = prog ~ gender + race + ses + schtyp + read +
##     write + math + science + socst, data = hsb, trace = FALSE)
##
## Coefficients:
##          (Intercept)  gendermale raceasian racehispanic racewhite
## general     3.631901 -0.09264717  1.352739   -0.6322019 0.2965156
## vocation    7.481381 -0.32104341 -0.700070   -0.1993556 0.3358881
##               seslow sesmiddle schtyppublic        read       write
## general   1.09864111 0.7029621    0.5845405 -0.04418353 -0.03627381
## vocation 0.04747323 1.1815808    2.0553336 -0.03481202 -0.03166001
##                math    science       socst
## general  -0.1092888 0.10193746 -0.01976995
## vocation -0.1139877 0.05229938 -0.08040129
##
## Std. Errors:
##          (Intercept) gendermale raceasian racehispanic racewhite    seslow
## general     1.823452  0.4548778  1.058754    0.8935504 0.7354829 0.6066763
## vocation    2.104698  0.5021132  1.470176    0.8393676 0.7480573 0.7045772
##          sesmiddle schtyppublic       read      write       math
## general  0.5045938    0.5642925 0.03103707 0.03381324 0.03522441
## vocation 0.5700833    0.8348229 0.03422409 0.03585729 0.03885131
##             science      socst
```

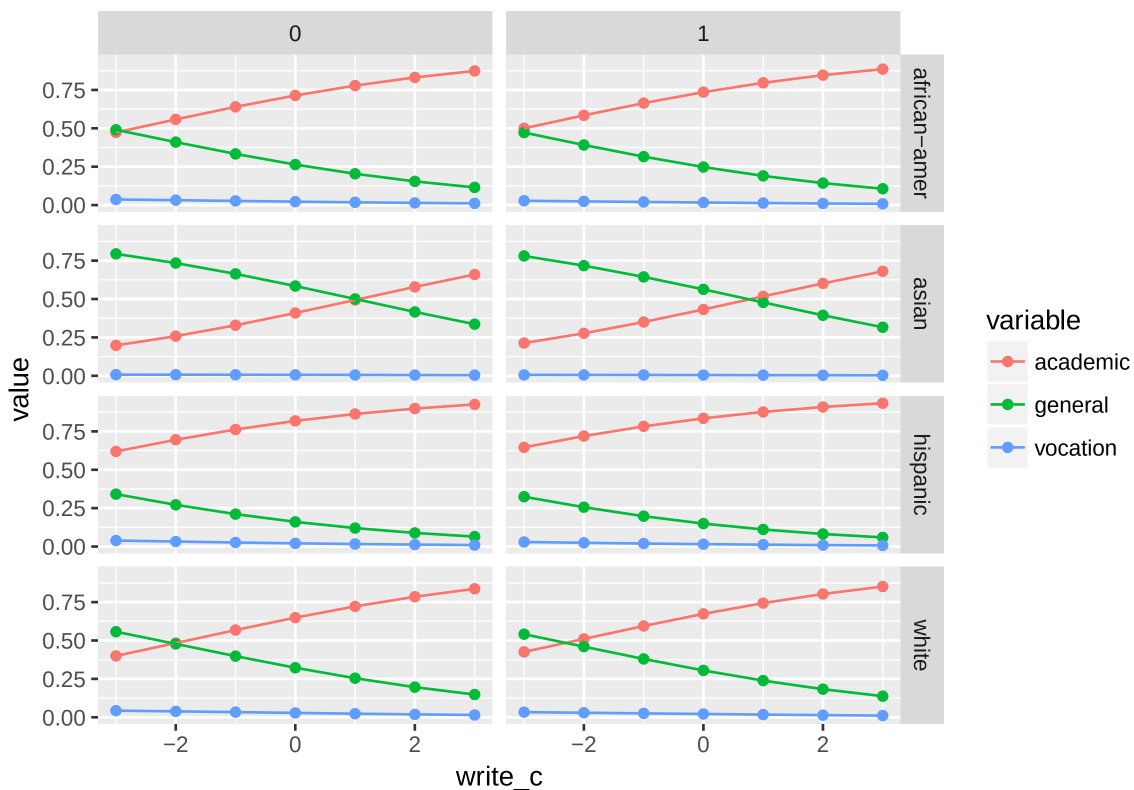
```
## general  0.03274038 0.02712589
## vocation 0.03424763 0.02938212
##
## Residual Deviance: 305.8705
## AIC: 357.8705
```

```r
hsb$male   <- 1*(hsb$gender=="male")
hsb$private<- 1*(hsb$schtyp=="private")
hsb$read_c <-scale(hsb$read,center=TRUE)
hsb$write_c<-scale(hsb$write,center=TRUE)
hsb$math_c <-scale(hsb$math,center=TRUE)
hsb$science_c<-scale(hsb$science,center=TRUE)
hsb$socst_c<-scale(hsb$socst,center=TRUE)

mmod <- vglm(prog ~ male + race + ses + private + read_c + write_c + math_c +
                             science_c + socst_c, hsb, family = multinomial)
summary(mmod)
```

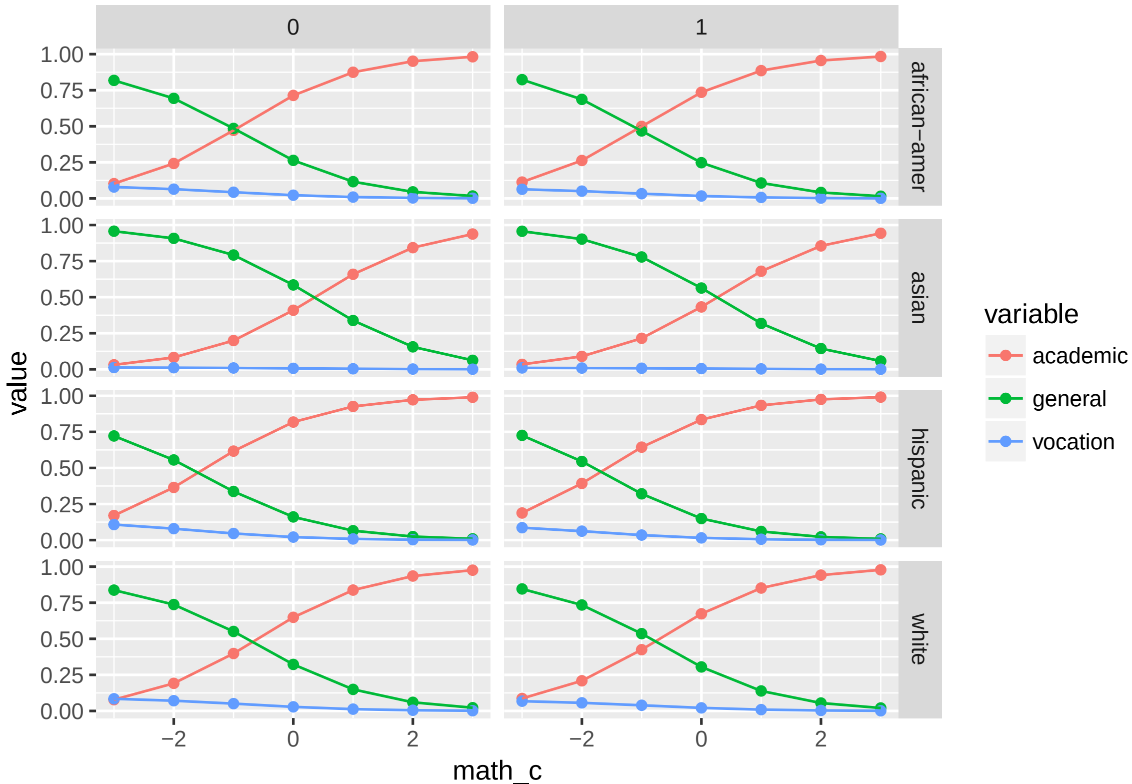
```
##
## Call:
## vglm(formula = prog ~ male + race + ses + private + read_c +
##     write_c + math_c + science_c + socst_c, family = multinomial,
##     data = hsb)
##
##
## Pearson residuals:
##                        Min      1Q  Median      3Q   Max
## log(mu[,1]/mu[,3]) -5.464 -0.5171  0.1712  0.5544 3.291
## log(mu[,2]/mu[,3]) -5.050 -0.4411 -0.2094 -0.0787 2.983
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept):1     1.45520    0.86770   1.677  0.09353 .
## (Intercept):2    -0.05465    0.92589  -0.059  0.95294
## male:1            0.32102    0.50210   0.639  0.52259
## male:2            0.22836    0.52744   0.433  0.66505
## raceasian:1       0.69946    1.47001   0.476  0.63420
## raceasian:2       2.05250    1.43700   1.428  0.15320
## racehispanic:1    0.19915    0.83937   0.237  0.81245
## racehispanic:2   -0.43286    0.91706  -0.472  0.63692
## racewhite:1      -0.33612    0.74806  -0.449  0.65320
## racewhite:2      -0.03943    0.76332  -0.052  0.95880
## seslow:1         -0.04748    0.70456  -0.067  0.94628
## seslow:2          1.05116    0.73124   1.438  0.15057
## sesmiddle:1      -1.18158    0.57007  -2.073  0.03820 *
## sesmiddle:2      -0.47862    0.62253  -0.769  0.44199
## private:1         2.05525    0.83473   2.462  0.01381 *
## private:2         1.47074    0.88945   1.654  0.09822 .
## read_c:1          0.35691    0.35089   1.017  0.30908
## read_c:2         -0.09608    0.35295  -0.272  0.78545
## write_c:1         0.30011    0.33987   0.883  0.37723
## write_c:2        -0.04374    0.35173  -0.124  0.90103
## math_c:1          1.06793    0.36397   2.934  0.00335 **
## math_c:2          0.04403    0.35729   0.123  0.90191
## science_c:1      -0.51778    0.33908  -1.527  0.12676
```

```
## science_c:2     0.49147    0.33333   1.474  0.14037
## socst_c:1       0.86317    0.31543   2.736  0.00621 **
## socst_c:2       0.65092    0.30940   2.104  0.03539 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  2
##
## Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
##
## Residual deviance: 305.8705 on 374 degrees of freedom
##
## Log-likelihood: -152.9353 on 374 degrees of freedom
##
## Number of iterations: 5
##
## Reference group is level  3  of the response
```

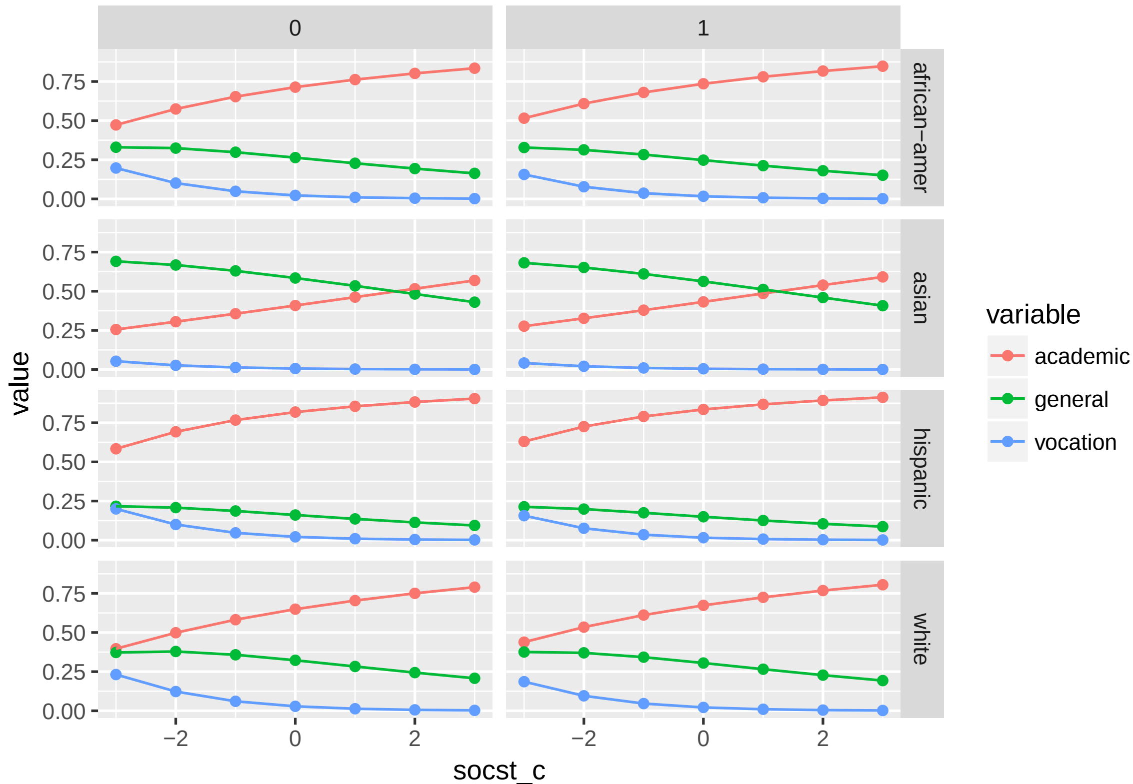
```
predmatx<-expand.grid( male =c(0,1),
                       race=c("african-amer", "asian", "hispanic", "white"),
                       ses="low" , private=1, read_c=0,
                       write_c=c(-3:3), math_c=0, science_c=0, socst_c=0)

predy<-predict(mmod,newdata=predmatx,type="response")
ggplot(melt(cbind(predmatx[,c("race","male","write_c")],predy),
            id.vars=c("race","male","write_c")))+
  geom_point()+aes(x=write_c,y=value,color=variable)+
  geom_line()+
  facet_grid(race~male)
```



```r
predmatx<-expand.grid( male =c(0,1),
                       race=c("african-amer", "asian", "hispanic", "white"),
                       ses="low" , private=1, read_c=0,
                       write_c=0, math_c=c(-3:3), science_c=0, socst_c=0)
predy<-predict(mmod,newdata=predmatx,type="response")

ggplot(melt(cbind(predmatx[,c("race","male","math_c")],predy),
            id.vars=c("race","male","math_c")))+
  geom_point()+aes(x=math_c,y=value,color=variable)+
  geom_line()+facet_grid(race~male)
```




```r
predmatx<-expand.grid( male =c(0,1),
                       race=c("african-amer", "asian", "hispanic", "white"),
                       ses="low" , private=1, read_c=0,
                       write_c=0, math_c=0, science_c=0, socst_c=c(-3:3))
predy<-predict(mmod,newdata=predmatx,type="response")

ggplot(melt(cbind(predmatx[,c("race","male","socst_c")],predy),
            id.vars=c("race","male","socst_c")))+
  geom_point()+aes(x=socst_c,y=value,color=variable)+
  geom_line()+facet_grid(race~male)
```



2. For the student with id 99, compute the predicted probabilities of the three possible choices.

```
hsb[hsb$id==99,]
```

```
##     id gender  race  ses schtyp    prog read write math science socst male
## 102 99 female white high public general   47    59   56      66    61    0
##     private      read_c   write_c   math_c science_c    socst_c
## 102       0 -0.5100977 0.6567435 0.358117  1.429164 0.8005929
```

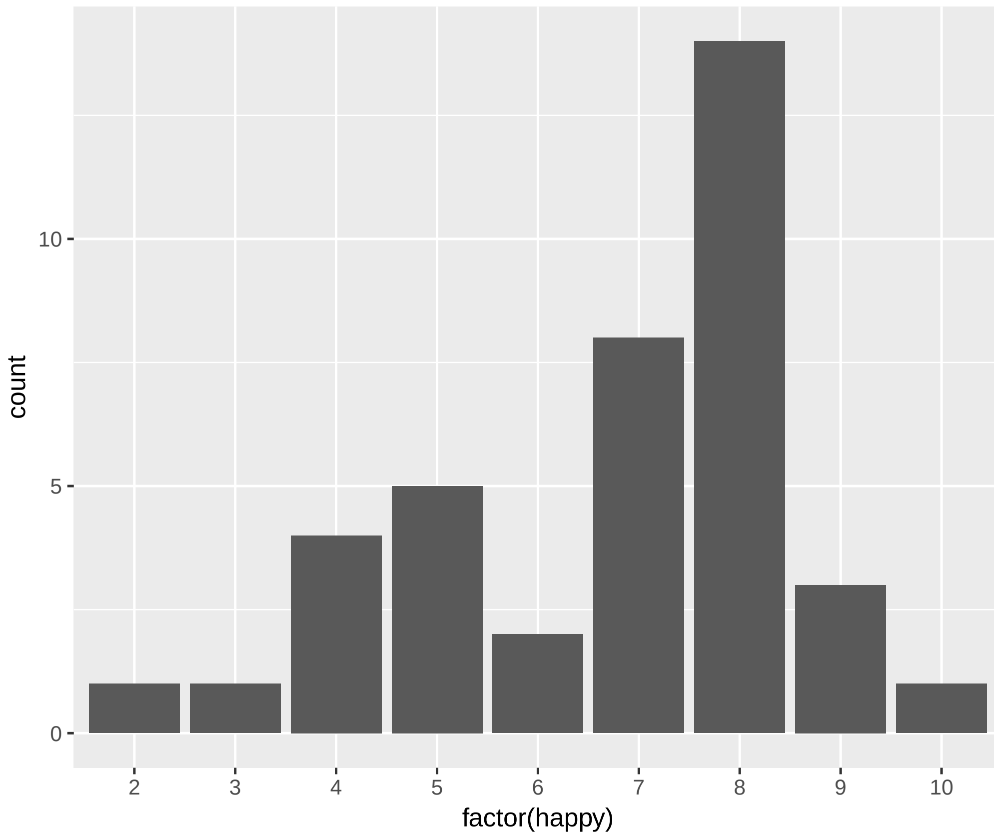
```
predict(mmod0, newdata = hsb[hsb$id==99,], type="probs")
```

```
##  academic   general  vocation
## 0.5076752 0.3753090 0.1170158
```

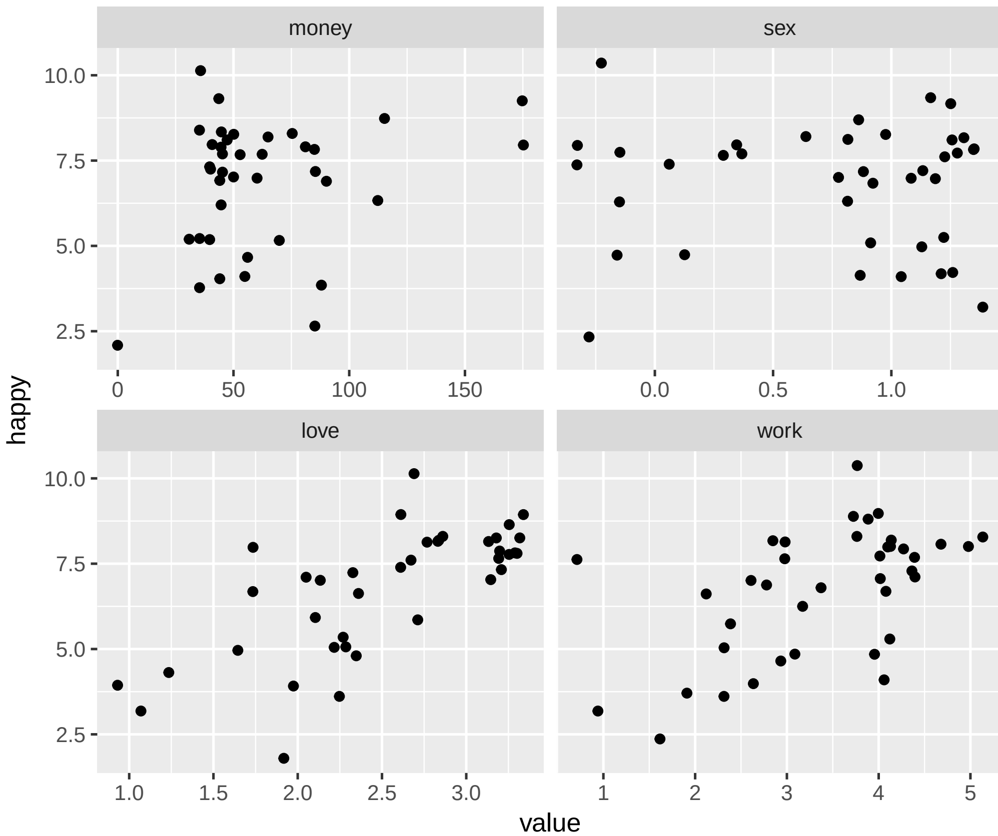
Happiness

Data were collected from 39 students in a University of Chicago MBA class and may be found in the dataset happy.

```
library(faraway)
data(happy)
?happy
ggplot(happy)+geom_bar()+aes(x=factor(happy))
```

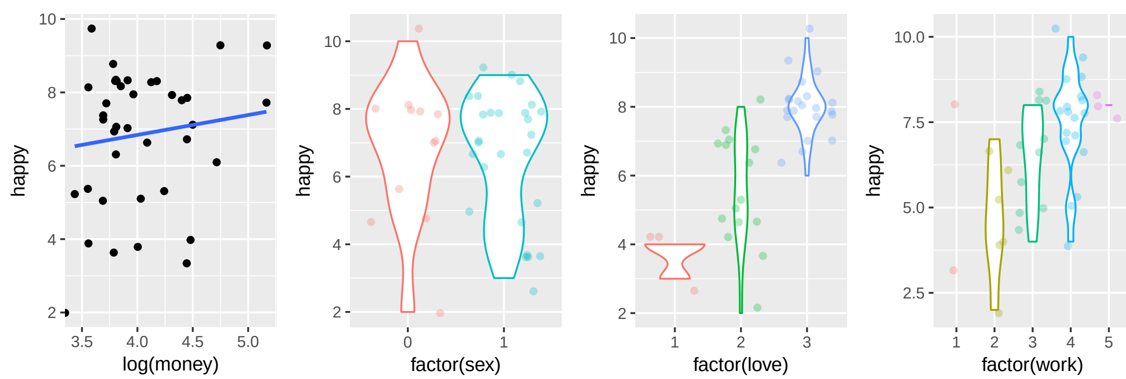



```
ggplot(melt(happy,id.vars = "happy"))+geom_jitter()+
  aes(x=value,y=happy)+facet_wrap(~variable, scale="free_x")
```




```
grid.arrange(
  ggplot(happy)+geom_jitter()+
  aes(x=log(money),y=happy)+geom_smooth(method="lm",se=FALSE),
  ggplot(happy)+geom_violin()+geom_jitter(alpha=0.3)+
  aes(x=factor(sex),y=happy,color=factor(sex)) + theme(legend.position="none"),
    ggplot(happy)+geom_violin()+geom_jitter(alpha=0.3)+
  aes(x=factor(love),y=happy,color=factor(love)) + theme(legend.position="none"),
      ggplot(happy)+geom_violin()+geom_jitter(alpha=0.3)+
  aes(x=factor(work),y=happy,color=factor(work)) + theme(legend.position="none"),
  ncol=4
)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```



1. Build a model for the level of happiness as a function of the other variables.

```r
happyfit<-polr(factor(happy)~money+love+sex+work,data=happy)
1-pchisq(deviance(happyfit),df.residual(happyfit))
```

```
## [1] 1.768587e-09
```

```r
# ggplot(melt(data.frame(money=predx$money,work=predx$work,love=predx$love,pred=predy),
#              id.vars=c("money","work","love")))+geom_line()+
#   aes(x=money,y=value,group=variable,color=variable)+
#   facet_grid(love~work)
```

We standardize money, center work at 3 aand love at 2.

```r
happy$money_c <- scale(happy$money,center=FALSE)
happy$work_c <- happy$work -3
happy$love_c <- happy$love -2
happyfitc<-polr(factor(happy)~money_c+love_c+sex+work_c,data=happy)
display(happyfitc)
```

```
##
## Re-fitting to get Hessian

## polr(formula = factor(happy) ~ money_c + love_c + sex + work_c,
##     data = happy)
##         coef.est coef.se
## money_c  1.62     0.77
## love_c   3.61     0.80
## sex     -0.47     0.79
## work_c   0.89     0.41
## 2|3     -4.41     1.57
## 3|4     -3.41     1.37
## 4|5     -0.72     0.95
## 5|6      1.09     0.84
## 6|7      1.63     0.85
## 7|8      3.67     1.02
## 8|9      7.41     1.49
## 9|10     9.13     1.81
## ---
## n = 39, k = 12 (including 8 intercepts)
## residual deviance = 94.9, null deviance is not computed by polr
```

We get sex that is negative and insignificant contrary to our expectation. It seems reasonable to remove sex variable from our model.

```r
happyfitc<-polr(factor(happy)~money_c+love_c+work_c,data=happy)
display(happyfitc)
```

```
##
## Re-fitting to get Hessian

## polr(formula = factor(happy) ~ money_c + love_c + work_c, data = happy)
##         coef.est coef.se
## money_c  1.49     0.73
## love_c   3.52     0.78
## work_c   0.97     0.39
## 2|3     -4.13     1.48
## 3|4     -3.11     1.27
## 4|5     -0.45     0.82
```

```
## 5|6      1.30     0.76
## 6|7      1.83     0.78
## 7|8      3.88     0.96
## 8|9      7.58     1.45
## 9|10     9.30     1.79
## ---
## n = 39, k = 11 (including 8 intercepts)
## residual deviance = 95.2, null deviance is not computed by polr
```

2. Interpret the parameters of your chosen model.

```r
display(happyfitc)
```

```
##
## Re-fitting to get Hessian
## polr(formula = factor(happy) ~ money_c + love_c + work_c, data = happy)
##         coef.est coef.se
## money_c  1.49     0.73
## love_c   3.52     0.78
## work_c   0.97     0.39
## 2|3     -4.13     1.48
## 3|4     -3.11     1.27
## 4|5     -0.45     0.82
## 5|6      1.30     0.76
## 6|7      1.83     0.78
## 7|8      3.88     0.96
## 8|9      7.58     1.45
## 9|10     9.30     1.79
## ---
## n = 39, k = 11 (including 8 intercepts)
## residual deviance = 95.2, null deviance is not computed by polr
```

```r
happyfitc2<-polr(factor(happy)~money+love_c+work_c,data=happy)
display(happyfitc)
```

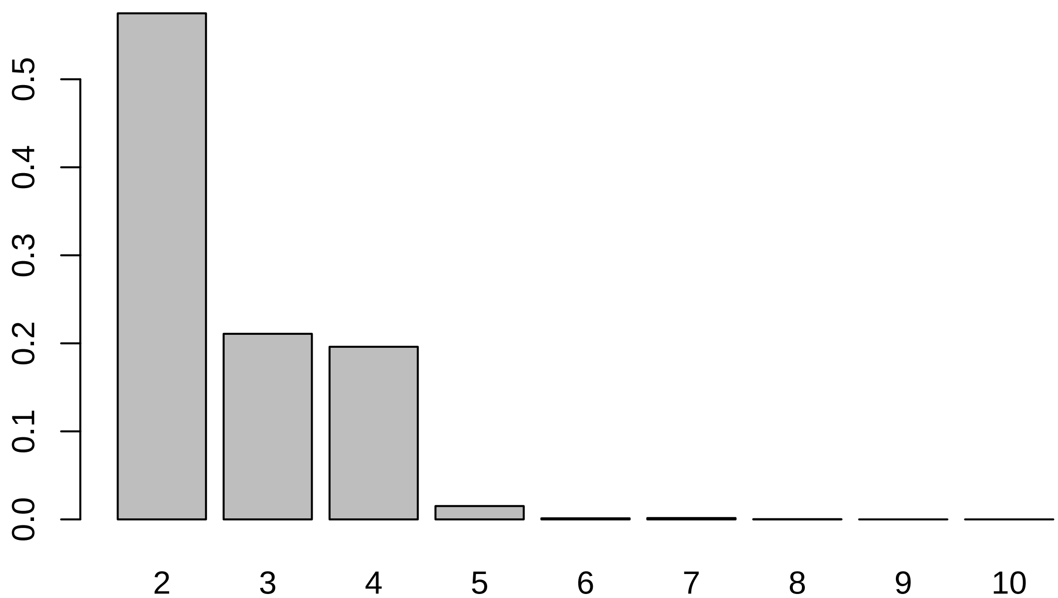
```
##
## Re-fitting to get Hessian
## polr(formula = factor(happy) ~ money_c + love_c + work_c, data = happy)
##         coef.est coef.se
## money_c  1.49     0.73
## love_c   3.52     0.78
## work_c   0.97     0.39
## 2|3     -4.13     1.48
## 3|4     -3.11     1.27
## 4|5     -0.45     0.82
## 5|6      1.30     0.76
## 6|7      1.83     0.78
## 7|8      3.88     0.96
## 8|9      7.58     1.45
## 9|10     9.30     1.79
## ---
## n = 39, k = 11 (including 8 intercepts)
## residual deviance = 95.2, null deviance is not computed by polr
```

```
# predx<-expand.grid(money=0:200,love_c=-1:1,work_c=-2:2)
# predy<-predict(happyfitc2,newdata=predx,type="prob")
# ggplot(melt(data.frame(predx,predy),id.vars = c("money",  "love_c", "work_c")))+geom_jitter()+
# aes(x=money,y=value,group=variable)+facet_grid(love_c~work_c,scale="free_x")
```

Overall, money, love and work seem to improve the happiness. However, one unit increase in love seem to have about equivalent effect as 2 standard deviation increase in money.

3. Predict the happiness distribution for subject whose parents earn $30,000 a year, who is lonely, not sexually active and has no job.

```
dist_est<-predict(happyfit,newdata=list(money=30,sex=0,work=1,love=1),type="prob")
barplot(dist_est)
```

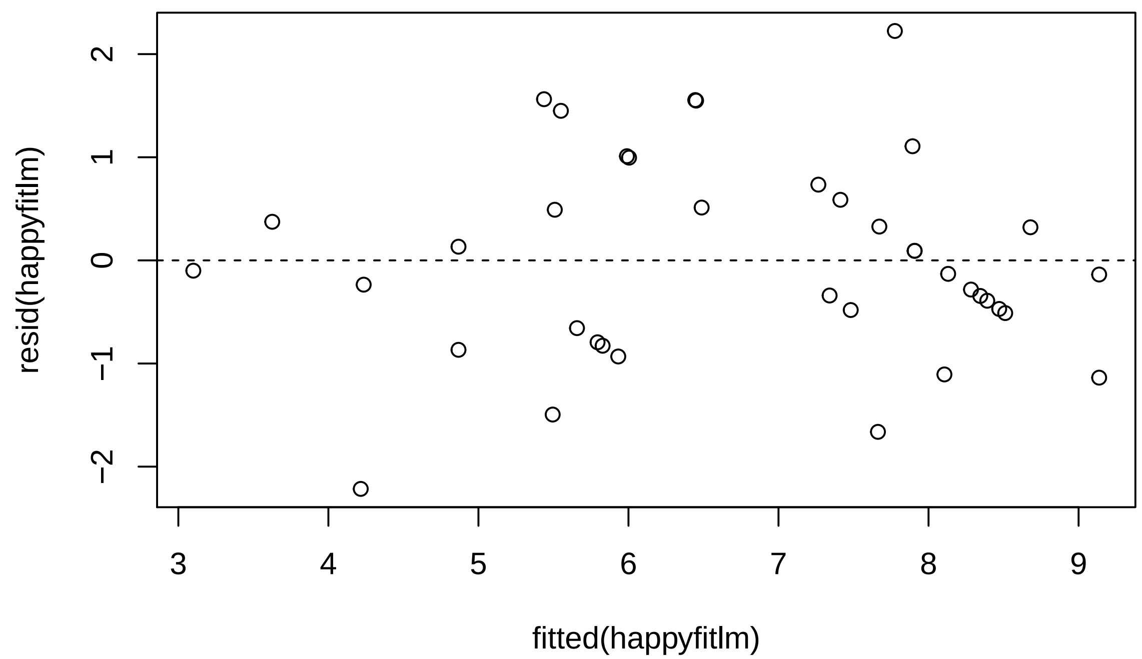


What happens if we use linear regression instead of cumulative logit?

```
happyfitlm<-lm(happy~log(money_c+1)+love_c+work_c,data=happy)
display(happyfitlm)
```

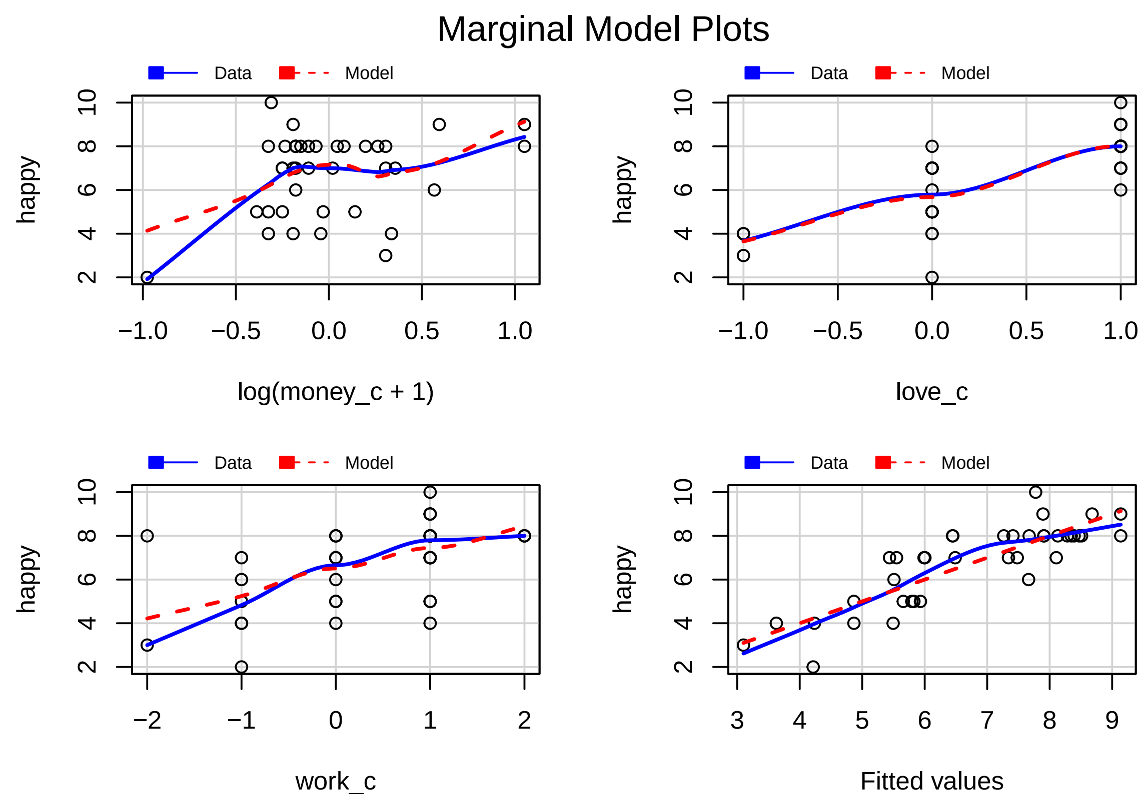
```
## lm(formula = happy ~ log(money_c + 1) + love_c + work_c, data = happy)
##                  coef.est coef.se
## (Intercept)      4.71     0.46
## log(money_c + 1) 1.65     0.71
## love_c           1.90     0.28
## work_c           0.49     0.18
## ---
## n = 39, k = 4
## residual sd = 1.02, R-Squared = 0.72
```

```
plot(fitted(happyfitlm),resid(happyfitlm));abline(h=0,lty=2)
```




```r
marginalModelPlots(happyfitlm)
```

```
## Warning in mmps(...): Splines and/or polynomials replaced by a fitted
## linear combination
```




```r
AIC(happyfitlm)
```

```
## [1] 117.9854
```

```r
AIC(happyfitc)
```

```
## [1] 117.2172
```

```r
table(pred=predict(happyfitc,type="class"),obs=happy$happy)
```

```
##     obs
## pred  2  3  4  5  6  7  8  9 10
##   2   0  1  0  0  0  0  0  0  0
##   3   0  0  0  0  0  0  0  0  0
##   4   1  0  3  1  0  0  0  0  0
##   5   0  0  1  2  1  2  0  0  0
##   6   0  0  0  0  0  0  0  0  0
##   7   0  0  0  2  0  3  2  0  0
##   8   0  0  0  0  1  3 11  2  1
##   9   0  0  0  0  0  0  1  1  0
##   10  0  0  0  0  0  0  0  0  0
```

```r
table(pred=round(predict(happyfitlm)),obs=happy$happy)
```

```
##     obs
## pred 2 3 4 5 6 7 8 9 10
##    3 0 1 0 0 0 0 0 0  0
##    4 1 0 2 0 0 0 0 0  0
##    5 0 0 2 1 0 1 0 0  0
##    6 0 0 0 4 1 4 2 0  0
##    7 0 0 0 0 0 2 2 0  0
##    8 0 0 0 0 1 1 8 1  1
##    9 0 0 0 0 0 0 2 2  0
```

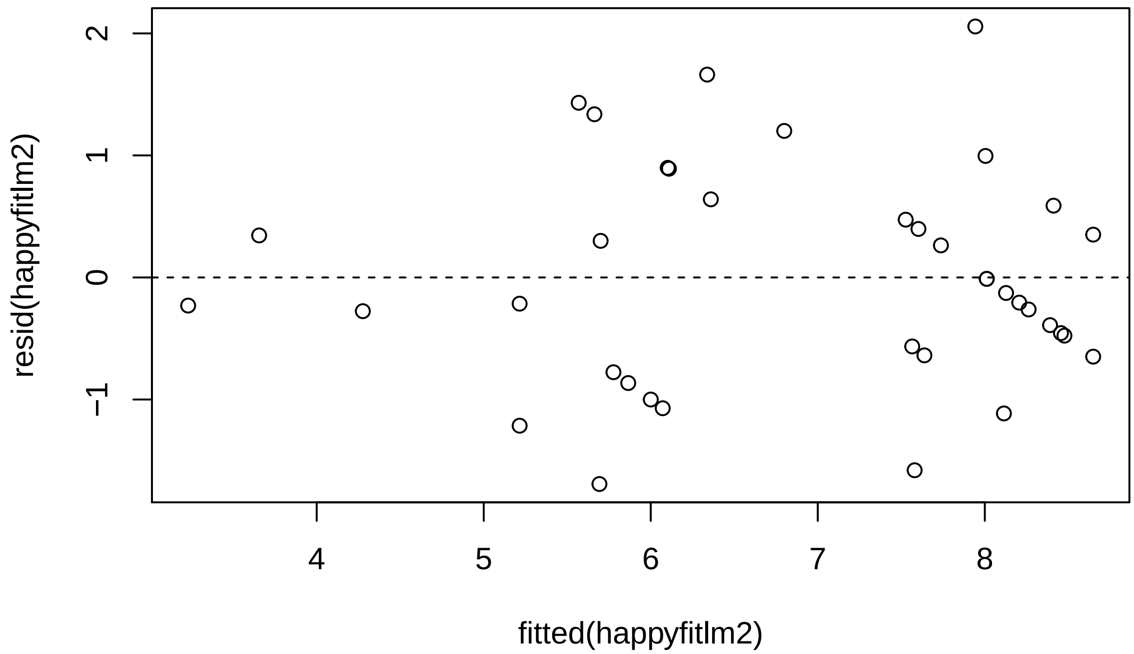
```r
table(predlm=round(predict(happyfitlm)),predml=predict(happyfitc,type="class"))
```

```
##       predml
## predlm  2  3  4  5  6  7  8  9 10
##      3  1  0  0  0  0  0  0  0  0
##      4  0  0  3  0  0  0  0  0  0
##      5  0  0  2  2  0  0  0  0  0
##      6  0  0  0  4  0  7  0  0  0
##      7  0  0  0  0  0  0  4  0  0
##      8  0  0  0  0  0  0 12  0  0
##      9  0  0  0  0  0  0  2  2  0
```

The model fits surprisingly well and the AIC is very close.

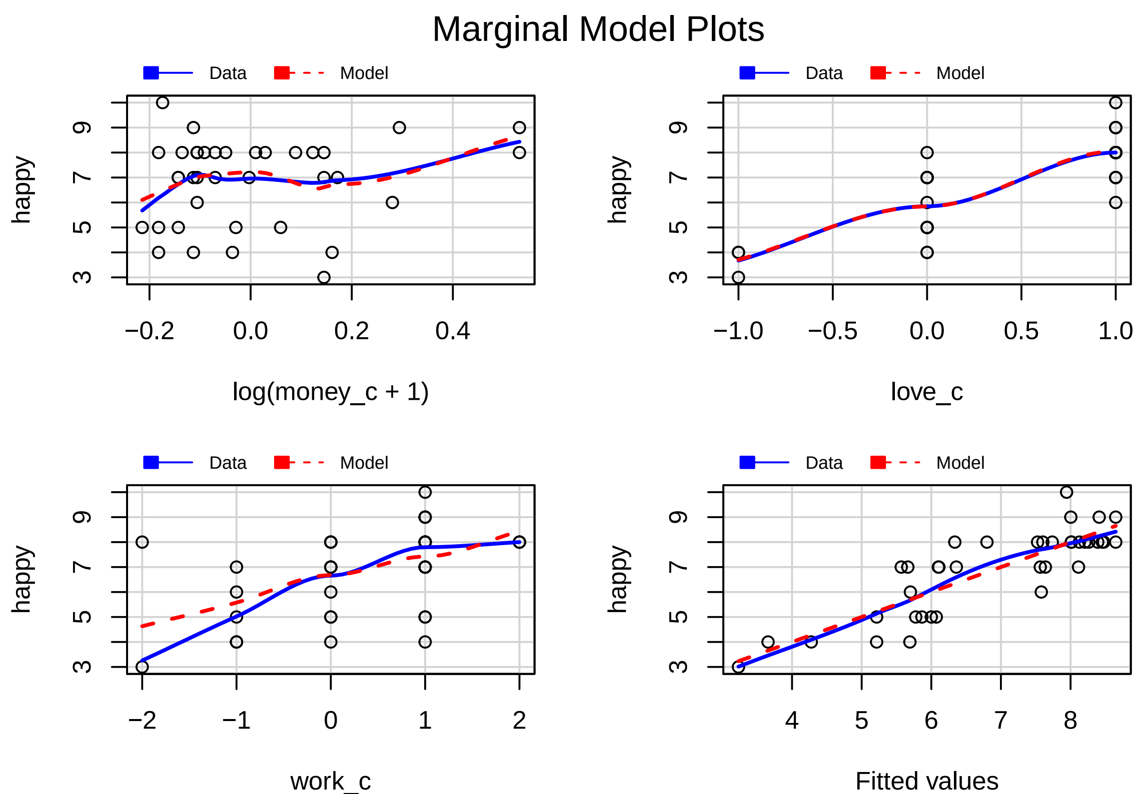
If we remove one indivisual that is a little off, we get a fairly good fit.

```r
happysub<-happy[happy$happy>2,]
happyfitlm2<-lm(happy~log(money_c+1)+love_c+work_c,data=happysub)
plot(fitted(happyfitlm2),resid(happyfitlm2));abline(h=0,lty=2)
```




```r
marginalModelPlots(happyfitlm2)
```

```
## Warning in mmps(...): Splines and/or polynomials replaced by a fitted
## linear combination
```




```r
table(pred=predict(happyfitc,newdata=happysub,type="class"),obs=happysub$happy)
```

```
##      obs
## pred  3  4  5  6  7  8  9 10
```

```
##   2    1  0  0  0  0  0  0  0
##   3    0  0  0  0  0  0  0  0
##   4    0  3  1  0  0  0  0  0
##   5    0  1  2  1  2  0  0  0
##   6    0  0  0  0  0  0  0  0
##   7    0  0  2  0  3  2  0  0
##   8    0  0  0  1  3 11  2  1
##   9    0  0  0  0  0  1  1  0
##   10   0  0  0  0  0  0  0  0
```

```
table(pred=round(predict(happyfitlm2)),obs=happysub$happy)
```

```
##     obs
## pred  3  4  5  6  7  8  9 10
##    3  1  0  0  0  0  0  0  0
##    4  0  2  0  0  0  0  0  0
##    5  0  1  1  0  0  0  0  0
##    6  0  1  4  1  5  1  0  0
##    7  0  0  0  0  0  1  0  0
##    8  0  0  0  1  3 11  2  1
##    9  0  0  0  0  0  1  1  0
```

```
table(predlm=round(predict(happyfitlm)),predml=predict(happyfitc,type="class"))
```

```
##       predml
## predlm  2  3  4  5  6  7  8  9 10
##      3  1  0  0  0  0  0  0  0  0
##      4  0  0  3  0  0  0  0  0  0
##      5  0  0  2  2  0  0  0  0  0
##      6  0  0  0  4  0  7  0  0  0
##      7  0  0  0  0  0  0  4  0  0
##      8  0  0  0  0  0  0 12  0  0
##      9  0  0  0  0  0  0  2  2  0
```

Which makes you think about the utility of using multinomial logit instead of simple linear regression.

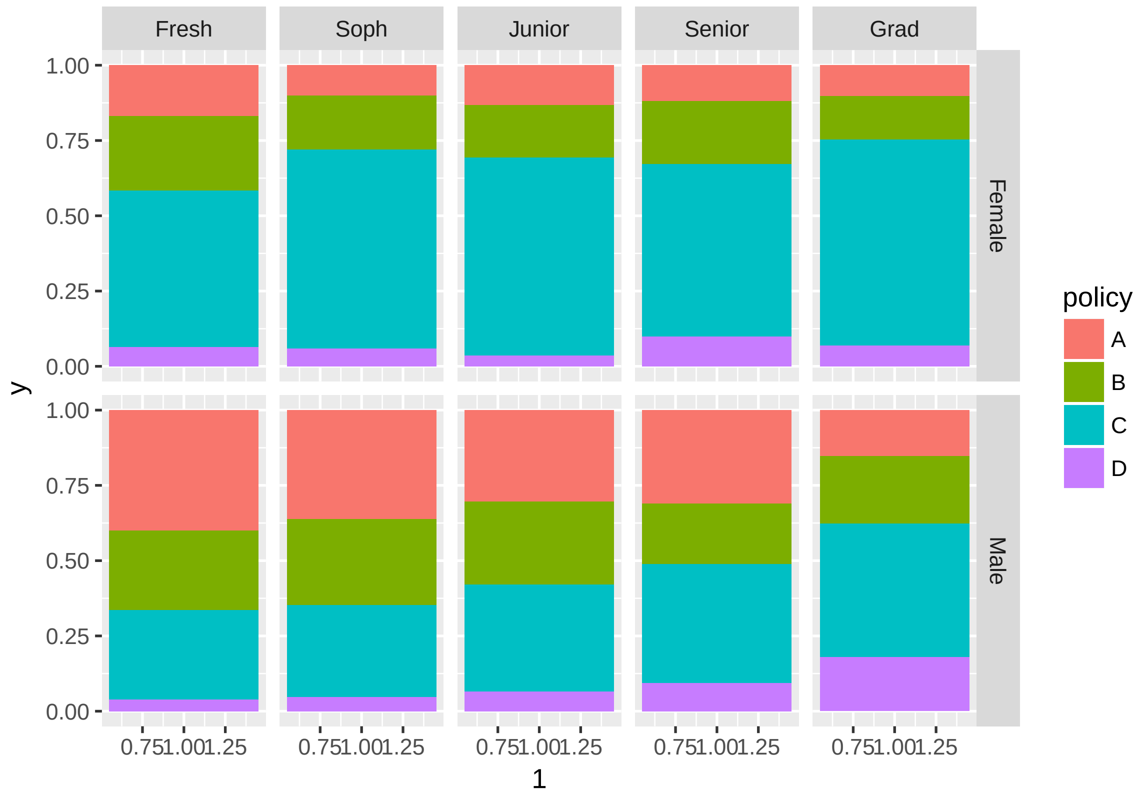
newspaper survey on Vietnam War

A student newspaper conducted a survey of student opinions about the Vietnam War in May 1967. Responses were classified by sex, year in the program and one of four opinions. The survey was voluntary. The data may be found in the dataset `uncviet`. Treat the opinion as the response and the sex and year as predictors. Build a proportional odds model, giving an interpretation to the estimates.

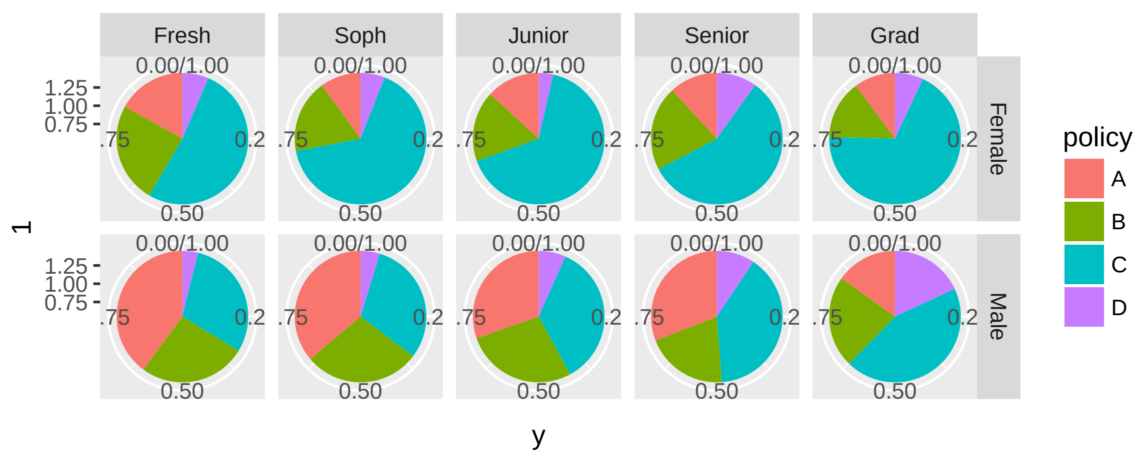
The options they had were:

- A (defeat power of North Vietnam by widesprefad bombing and land invasion)
- B (follow the present policy)
- C (withdraw troops to strong points and open negotiations on elections involving the Viet Cong)
- D (immediate withdrawal of all U.S. troops)

```
data(uncviet)
?uncviet
uncviet$year=factor(uncviet$year,levels=c("Fresh",   "Soph" ,  "Junior", "Senior","Grad" ) )
ggplot(uncviet)+geom_bar(position = "fill",stat="identity")+aes(x=1,y=y,fill=policy)+
  facet_grid(sex~year)
```




```
ggplot(uncviet)+geom_bar(position = "fill",stat="identity")+aes(x=1,y=y,fill=policy)+
  facet_grid(sex~year)+ coord_polar("y", start=0)
```

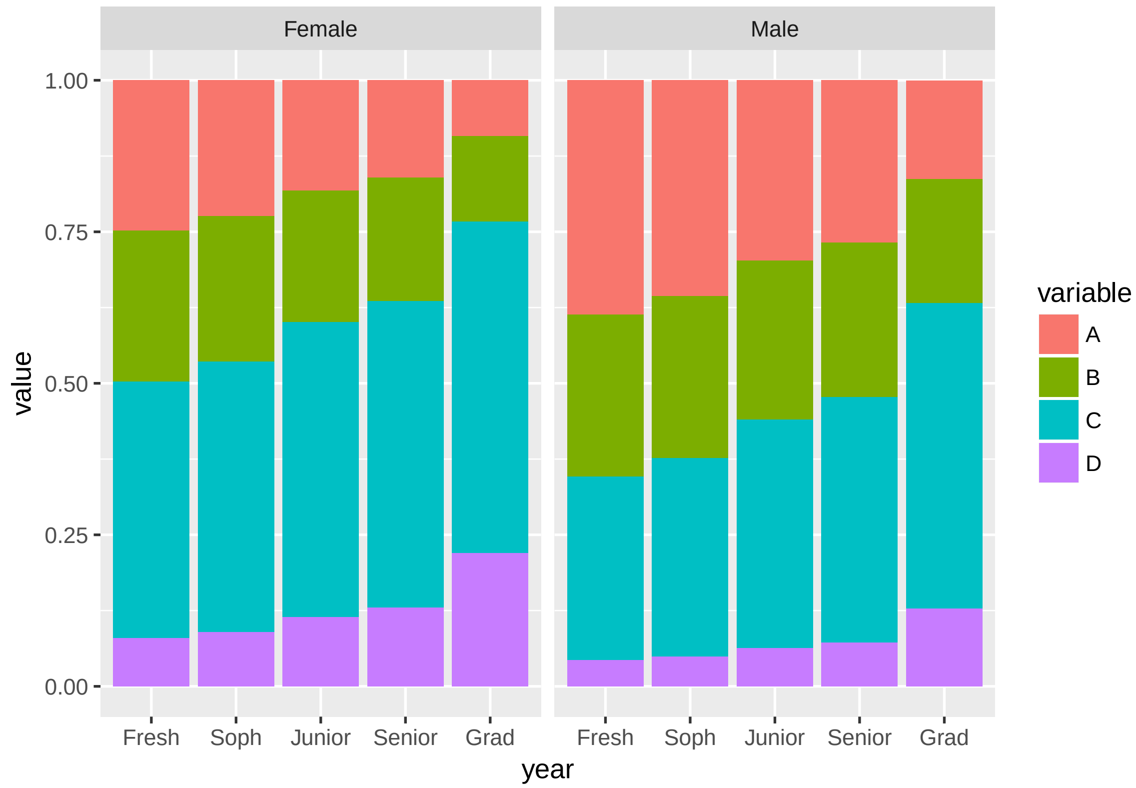



```
ddat<-dcast(uncviet, sex+year~policy,value.var = "y")
fit<-vglm(cbind(A,B,C,D)~sex+year,
        family=cumulative(parallel=TRUE),data=ddat)
summary(fit)
```

```
##
## Call:
## vglm(formula = cbind(A, B, C, D) ~ sex + year, family = cumulative(parallel = TRUE),
##     data = ddat)
##
##
## Pearson residuals:
```

```
##                    Min      1Q  Median     3Q   Max
## logit(P[Y<=1]) -1.599 -1.2074 -0.5335 0.2051 2.599
## logit(P[Y<=2]) -2.882 -1.1441 -0.5285 0.7850 1.761
## logit(P[Y<=3]) -4.508 -0.2575  0.5764 1.1012 5.072
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1 -1.10979    0.11220  -9.891  < 2e-16 ***
## (Intercept):2 -0.01305    0.11069  -0.118 0.906167
## (Intercept):3  2.44170    0.12118  20.149  < 2e-16 ***
## sexMale        0.64703    0.08720   7.420 1.17e-13 ***
## yearSoph      -0.13150    0.11532  -1.140 0.254141
## yearJunior    -0.39642    0.11054  -3.586 0.000335 ***
## yearSenior    -0.54439    0.11165  -4.876 1.08e-06 ***
## yearGrad      -1.17699    0.10238 -11.496  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 112.0238 on 22 degrees of freedom
##
## Log-likelihood: -131.8698 on 22 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##    sexMale   yearSoph yearJunior yearSenior   yearGrad
##  1.9098677  0.8767760  0.6727233  0.5801928  0.3082047
```

```r
predx<-expand.grid(sex=levels(uncviet$sex),year=levels(uncviet$year))
predy<-(predict(fit,newdata=predx,type="response"))
predx$year=factor(predx$year,levels=c("Fresh",   "Soph" ,  "Junior", "Senior","Grad" ) )
 ggplot(melt(data.frame(predx,predy),id.vars = c("sex",  "year")))+geom_bar(stat="identity")+
 aes(x=year,y=value,fill=variable)+facet_grid(~sex)
```

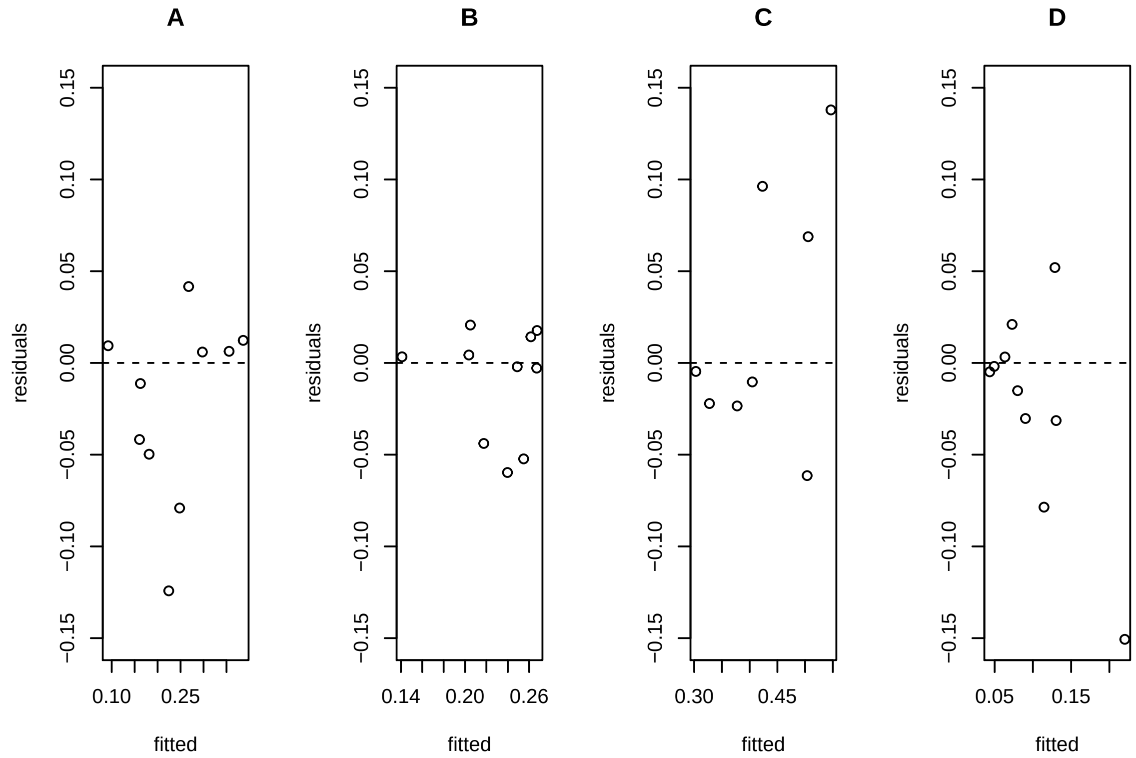



```
predy<-(predict(fit,newdata=ddat[,1:2],type="response"))
obsprob<-ddat[,3:6]/rowSums(ddat[,3:6])
res<-obsprob-predy
```

The result shows option C (withdraw troops to strong points and open negotiations on elections involving the Viet Cong) was the most popoular across the years and A (defeat power of North Vietnam by widespread bombing and land invasion) is popular among males particularily around their younger years.

When you look at the residual you see that the there are couple of observations that are off and you see trends in the residuals.

```
labs<-c("A","B","C","D")
par(mfrow=c(1,4))
for(i in 1:4)  {   plot(predy[,i],res[,i],ylim=c(-0.15,0.15),main=labs[i],xlab="fitted",ylab="residuals
```



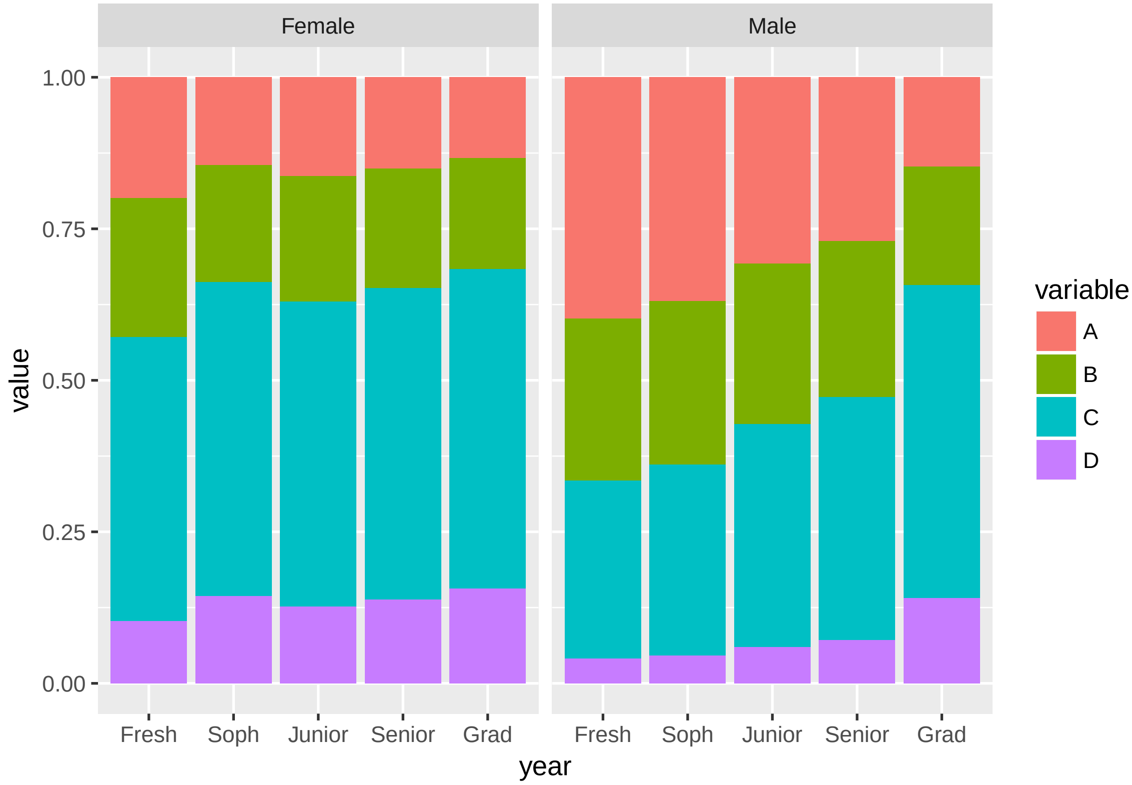
We can add interaction terms

```r
fit2<-vglm(cbind(A,B,C,D)~sex*year,
        family=cumulative(parallel=TRUE),data=ddat)
summary(fit2)
```

```
##
## Call:
## vglm(formula = cbind(A, B, C, D) ~ sex * year, family = cumulative(parallel = TRUE),
##     data = ddat)
##
##
## Pearson residuals:
##                     Min      1Q  Median      3Q   Max
## logit(P[Y<=1]) -0.9157 -0.8317 -0.6194 -0.1960 2.530
## logit(P[Y<=2]) -2.5022 -1.1209 -0.2865  0.3662 2.871
## logit(P[Y<=3]) -3.5256 -0.4885  0.7256  1.7279 3.890
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1        -1.3930     0.2130  -6.539 6.19e-11 ***
## (Intercept):2        -0.2898     0.2115  -1.370  0.17060
## (Intercept):3         2.1676     0.2173   9.974  < 2e-16 ***
## sexMale               0.9778     0.2288   4.274 1.92e-05 ***
## yearSoph             -0.3858     0.3396  -1.136  0.25599
## yearJunior           -0.2421     0.2558  -0.946  0.34396
## yearSenior           -0.3398     0.2819  -1.206  0.22798
## yearGrad             -0.4832     0.2523  -1.915  0.05551 .
## sexMale:yearSoph      0.2658     0.3613   0.736  0.46194
## sexMale:yearJunior   -0.1568     0.2842  -0.552  0.58099
## sexMale:yearSenior   -0.2400     0.3069  -0.782  0.43423
```

```
## sexMale:yearGrad    -0.8577     0.2755  -3.113  0.00185 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 91.0376 on 18 degrees of freedom
##
## Log-likelihood: -121.3767 on 18 degrees of freedom
##
## Number of iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##            sexMale           yearSoph         yearJunior
##          2.6586637          0.6798979          0.7849627
##         yearSenior           yearGrad   sexMale:yearSoph
##          0.7118924          0.6168357          1.3044855
## sexMale:yearJunior sexMale:yearSenior   sexMale:yearGrad
##          0.8548418          0.7866111          0.4241286
```

```
predx<-expand.grid(sex=levels(uncviet$sex),year=levels(uncviet$year))
predy<-(predict(fit2,newdata=predx,type="response"))
predx$year=factor(predx$year,levels=c("Fresh",   "Soph" ,  "Junior", "Senior","Grad" ) )
 ggplot(melt(data.frame(predx,predy),id.vars = c("sex",  "year")))+geom_bar(stat="identity")+
 aes(x=year,y=value,fill=variable)+facet_grid(~sex)
```

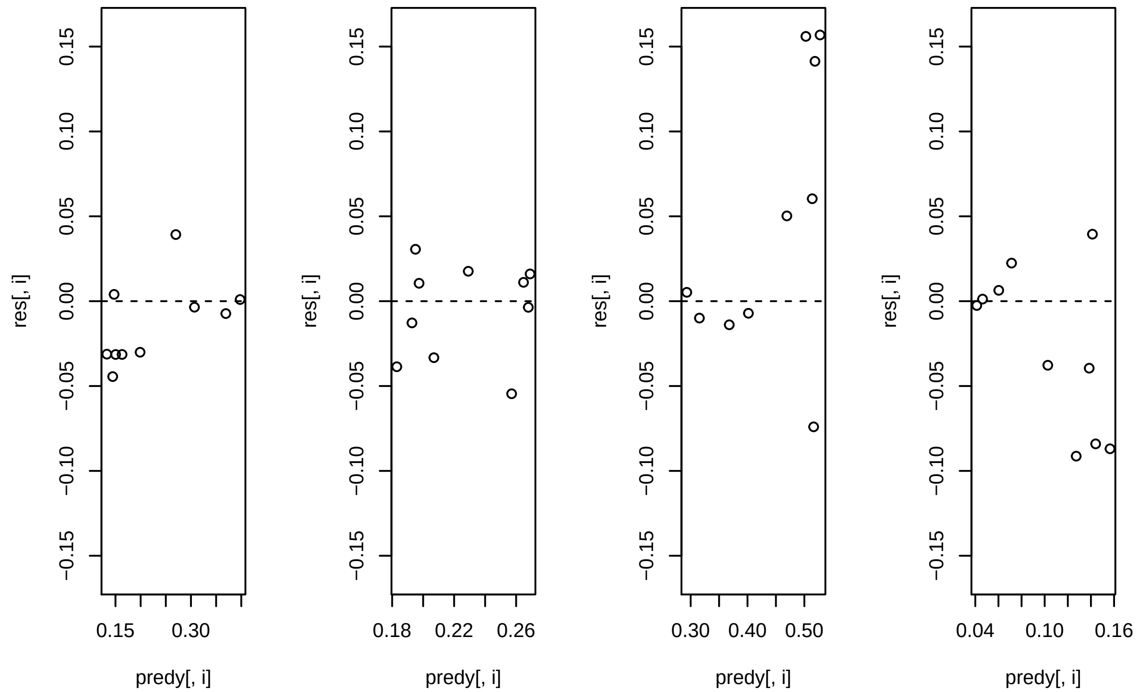



```
predy<-(predict(fit2,newdata=ddat[,1:2],type="response"))
obsprob<-ddat[,3:6]/rowSums(ddat[,3:6])
res<-obsprob-predy
```

which sees to show that females are more consistent in their reporting and the opinion of the males were changing with age.

However, we still see there is underestimation of opnion C, which might suggest the proportional odds model might not be the best choice.

```
par(mfrow=c(1,4))
for(i in 1:4)  {   plot(predy[,i],res[,i],ylim=c(-0.16,0.16));abline(h=0,lty=2)}
```



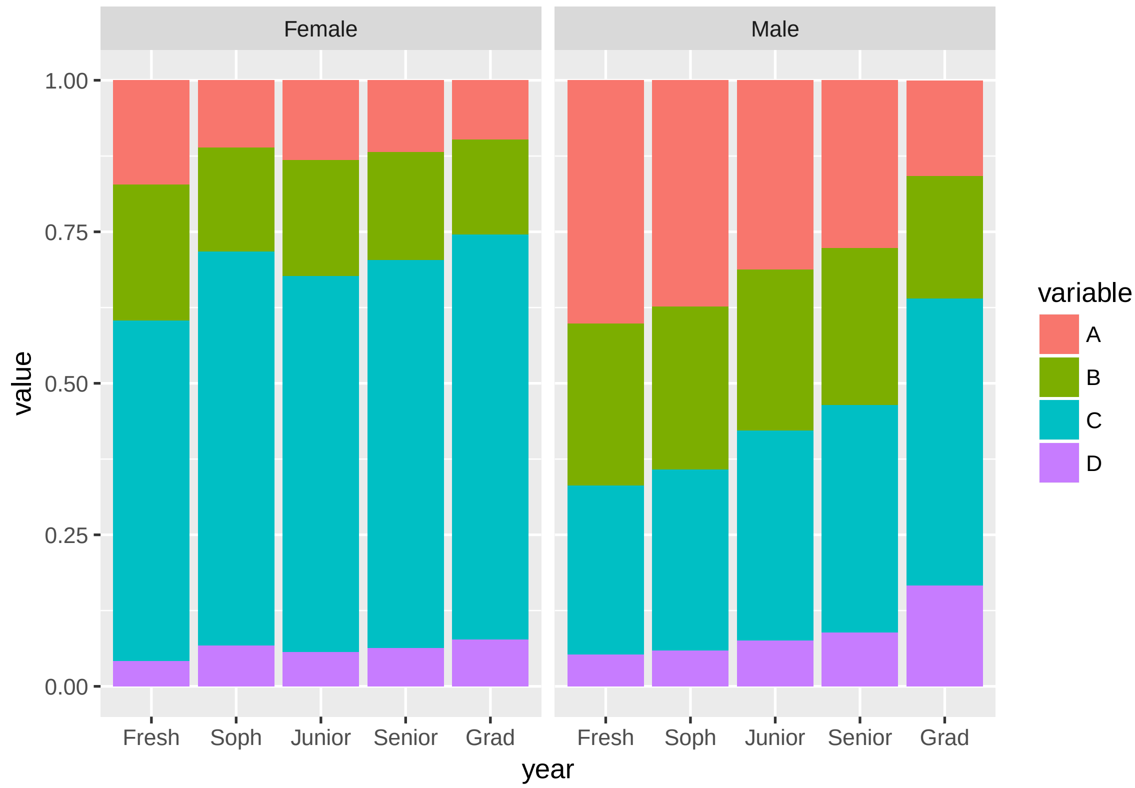
If we allow the gender coefficient to vary we get:

```
fit3<-vglm(cbind(A,B,C,D)~sex*year,
        family=cumulative(parallel=FALSE~sex),data=ddat)
summary(fit3)
```

```
##
## Call:
## vglm(formula = cbind(A, B, C, D) ~ sex * year, family = cumulative(parallel = FALSE ~
##     sex), data = ddat)
##
##
## Pearson residuals:
##                      Min      1Q   Median      3Q   Max
## logit(P[Y<=1]) -0.8177 -0.4638 -0.15348 0.07133 2.187
## logit(P[Y<=2]) -1.7635 -0.5283  0.05983 0.47883 1.617
## logit(P[Y<=3]) -1.5101 -0.8577  0.32277 0.96829 1.386
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept):1        -1.56931    0.23569  -6.658 2.77e-11 ***
## (Intercept):2        -0.42189    0.22169  -1.903   0.0570 .
## (Intercept):3         3.13171    0.27414  11.424  < 2e-16 ***
## sexMale:1             1.16848    0.25206   4.636 3.56e-06 ***
## sexMale:2             1.12256    0.23932   4.691 2.73e-06 ***
## sexMale:3            -0.24307    0.29498  -0.824   0.4099
## yearSoph             -0.50863    0.36353  -1.399   0.1618
## yearJunior           -0.31908    0.26903  -1.186   0.2356
## yearSenior           -0.44186    0.29881  -1.479   0.1392
## yearGrad             -0.65374    0.26855  -2.434   0.0149 *
## sexMale:yearSoph      0.39064    0.38383   1.018   0.3088
## sexMale:yearJunior -0.06967    0.29603  -0.235   0.8139
```

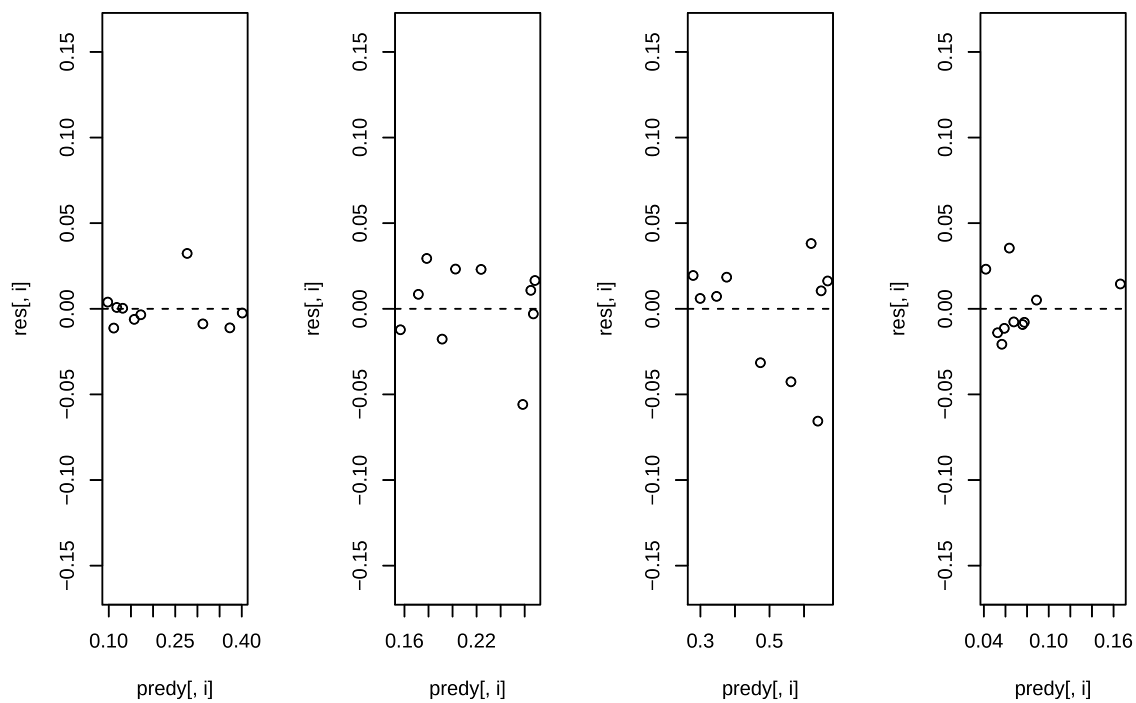
```
## sexMale:yearSenior -0.11748    0.32256  -0.364   0.7157
## sexMale:yearGrad   -0.62265    0.29086  -2.141   0.0323 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  3
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3])
##
## Residual deviance: 25.0046 on 16 degrees of freedom
##
## Log-likelihood: -88.3602 on 16 degrees of freedom
##
## Number of iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##          sexMale:1          sexMale:2          sexMale:3
##          3.2170979          3.0726972          0.7842154
##           yearSoph         yearJunior         yearSenior
##          0.6013208          0.7268190          0.6428406
##           yearGrad   sexMale:yearSoph sexMale:yearJunior
##          0.5200972          1.4779194          0.9326996
## sexMale:yearSenior   sexMale:yearGrad
##          0.8891547          0.5365204
```

```
predx<-expand.grid(sex=levels(uncviet$sex),year=levels(uncviet$year))
predy<-(predict(fit3,newdata=predx,type="response"))
predx$year=factor(predx$year,levels=c("Fresh",   "Soph" ,  "Junior", "Senior","Grad" ) )
 ggplot(melt(data.frame(predx,predy),id.vars = c("sex",  "year")))+geom_bar(stat="identity")+
 aes(x=year,y=value,fill=variable)+facet_grid(~sex)
```




```r
predy<-(predict(fit3,newdata=ddat[,1:2],type="response"))
obsprob<-ddat[,3:6]/rowSums(ddat[,3:6])
res<-obsprob-predy
```

```r
par(mfrow=c(1,4))
for(i in 1:4)  {   plot(predy[,i],res[,i],ylim=c(-0.16,0.16));abline(h=0,lty=2)}
```



which seems to fit the data fairly well.

You can also see this in the drop in AIC.

```
AIC(fit)
```

```
## [1] 279.7396
```

```
AIC(fit2)
```

```
## [1] 266.7534
```

```
AIC(fit3)
```

```
## [1] 204.7204
```

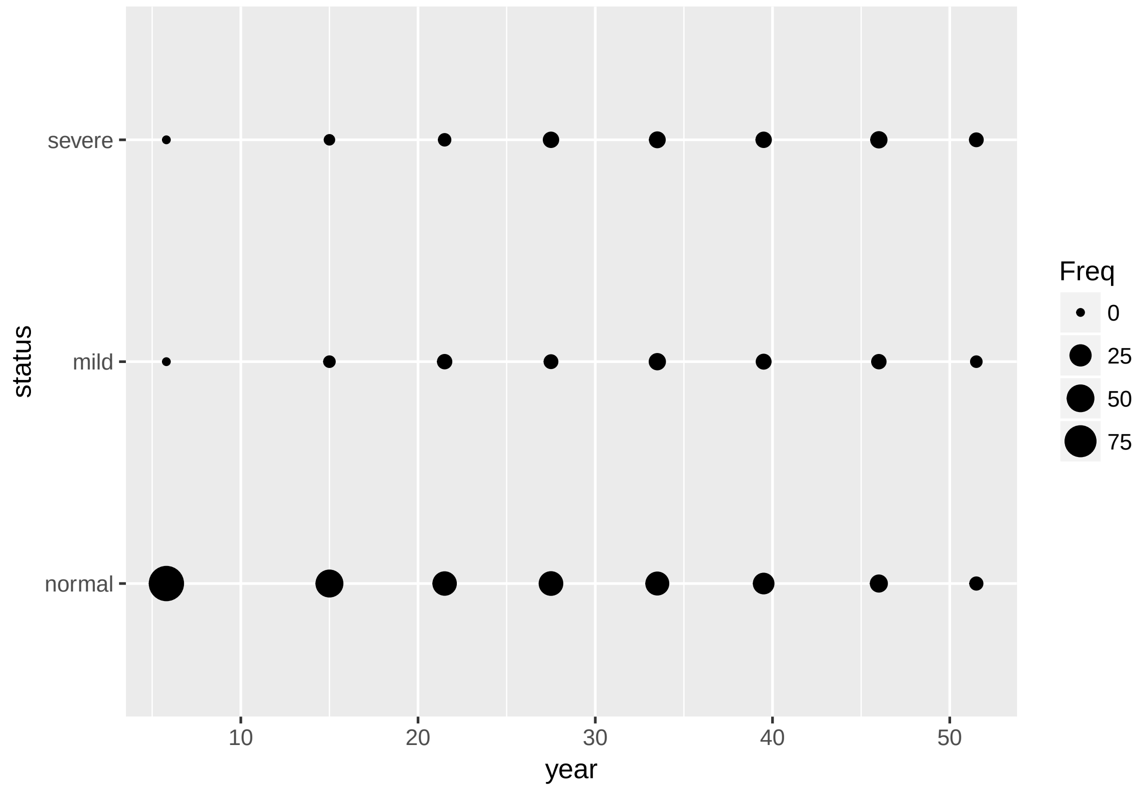
pneumonoconiosis of coal miners

The pneumo data gives the number of coal miners classified by radiological examination into one of three categories of pneumonoconiosis and by the number of years spent working at the coal face divided into eight categories.

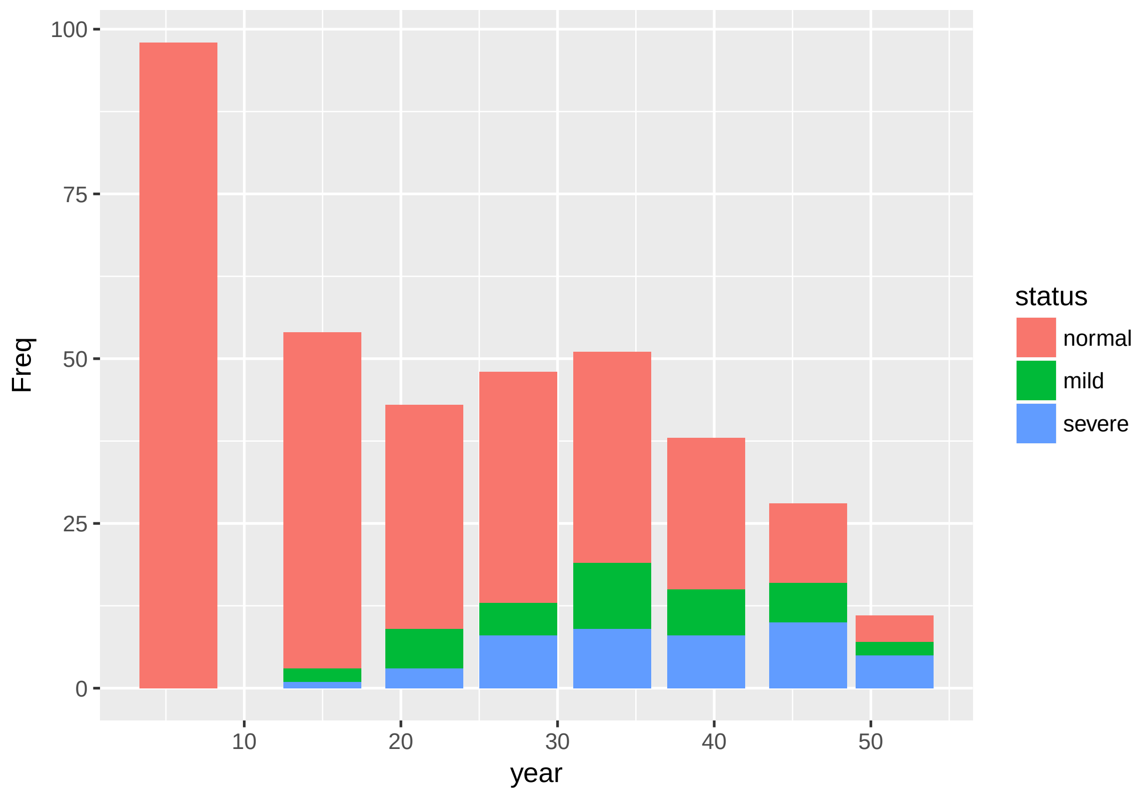
```
data(pneumo,package="faraway")
?pneumo
```

```
## Help on topic 'pneumo' was found in the following packages:
##
##   Package               Library
##   VGAM                  /Library/Frameworks/R.framework/Versions/3.4/Resources/library
##   faraway               /Library/Frameworks/R.framework/Versions/3.4/Resources/library
##
##
## Using the first match ...
```

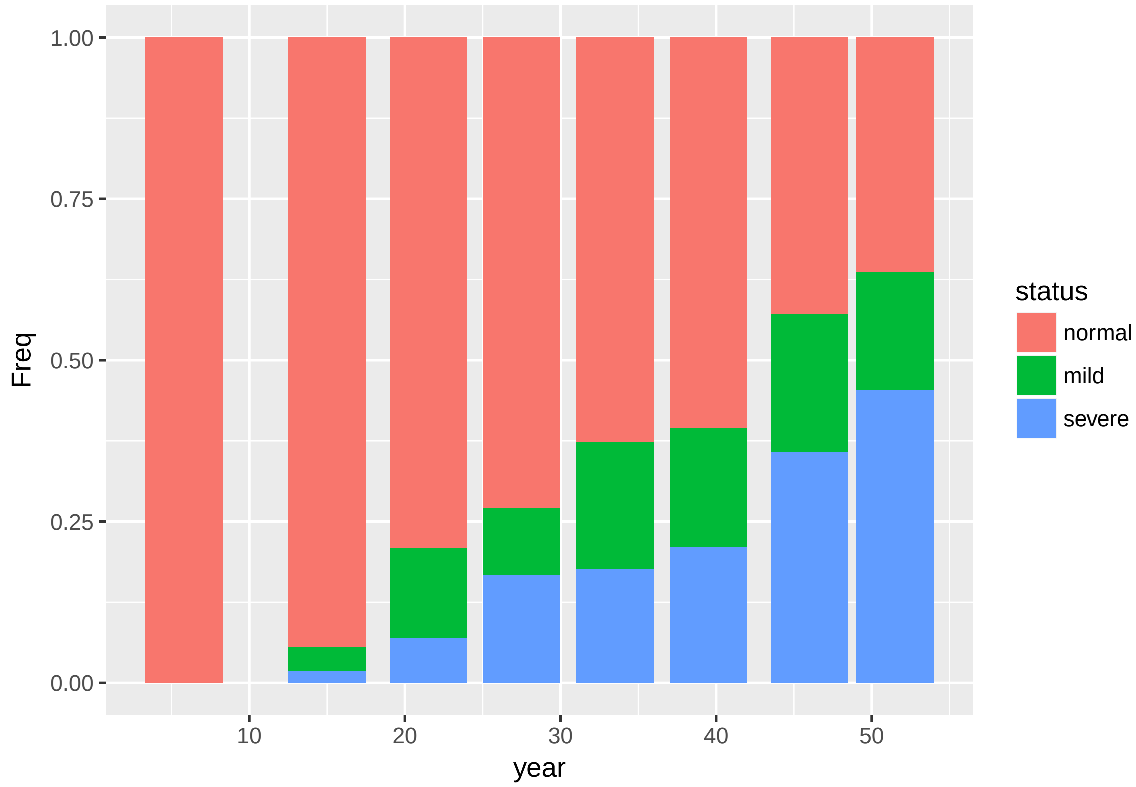
```
pneumo$status<-factor(pneumo$status,levels=c("normal","mild", "severe"))
ggplot(pneumo)+
  geom_point()+
  aes(x=year,y=status,size=Freq)
```




```r
ggplot(pneumo)+
  geom_bar(stat="identity")+
  aes(x=year,fill=status,y=Freq)
```



```r
ggplot(pneumo)+
  geom_bar(stat="identity",position="fill")+
  aes(x=year,fill=status,y=Freq)
```



1. Treating the pneumonoconiosis status as response variable as nominal, build a model for predicting the frequency of the three outcomes in terms of length of service and use it to predict the outcome for a miner with 25 years of service.

```r
counts <- xtabs(Freq ~ status + year, data=pneumo)
 round((props <- prop.table(counts, 2)),2)
```

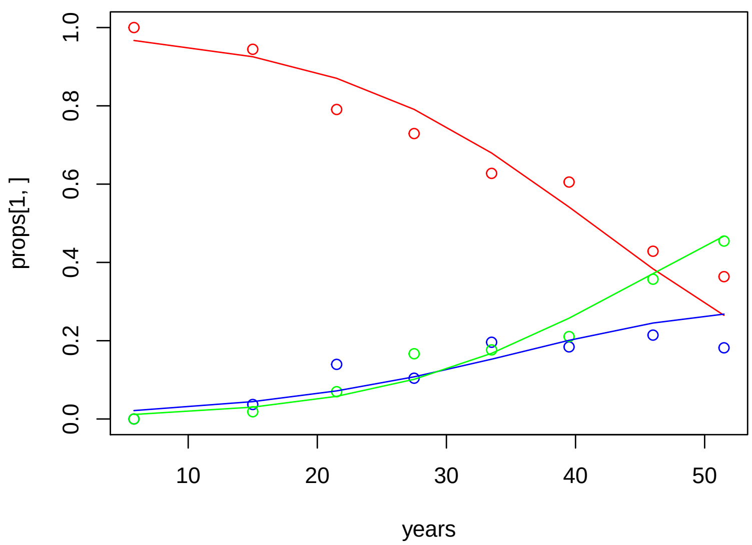
```
##         year
## status    5.8   15 21.5 27.5 33.5 39.5   46 51.5
##   normal 1.00 0.94 0.79 0.73 0.63 0.61 0.43 0.36
##   mild   0.00 0.04 0.14 0.10 0.20 0.18 0.21 0.18
##   severe 0.00 0.02 0.07 0.17 0.18 0.21 0.36 0.45
```

```r
years <- c(5.8, 15, 21.5, 27.5, 33.5, 39.5, 46, 51.5)
mmod <- multinom(t(counts) ~ years, trace=FALSE)
summary(mmod)
```

```
## Call:
## multinom(formula = t(counts) ~ years, trace = FALSE)
##
## Coefficients:
##        (Intercept)      years
## mild     -4.291680 0.08356529
## severe   -5.059849 0.10928549
##
## Std. Errors:
```

```
##          (Intercept)      years
## mild      0.5214120 0.01528046
## severe    0.5964319 0.01646978
##
## Residual Deviance: 417.4496
## AIC: 425.4496
```

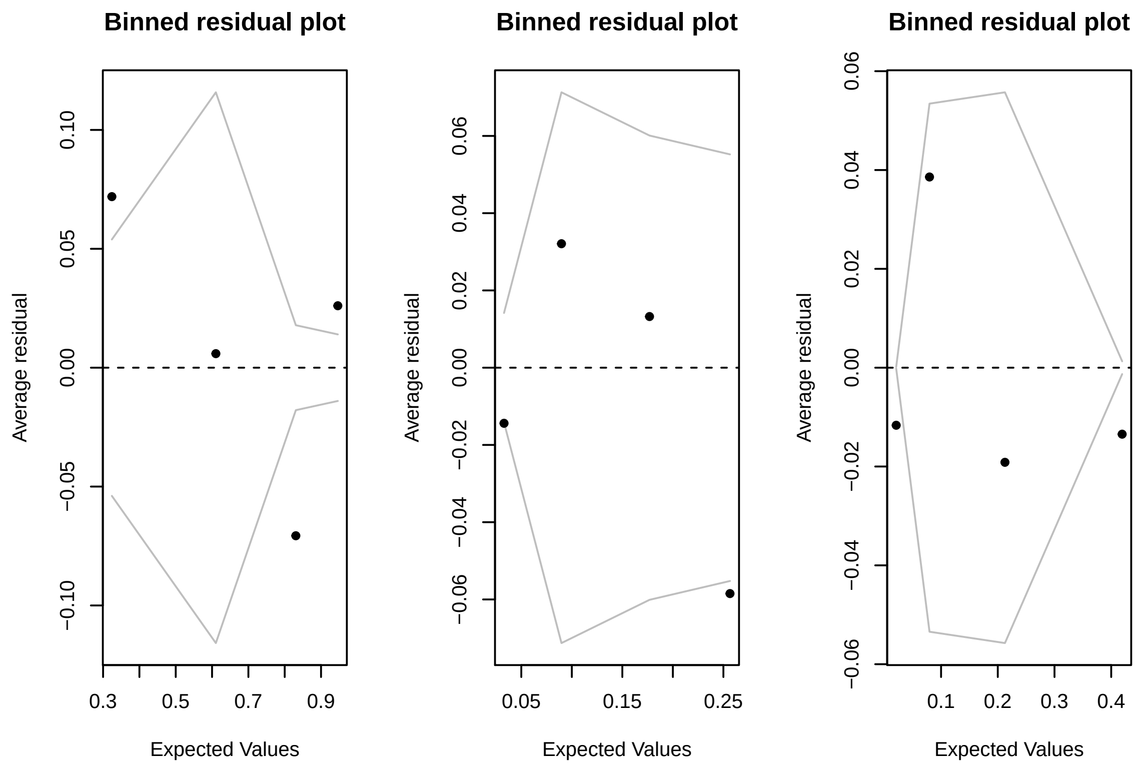
```r
par(mfrow=c(1,1))
plot(years, props[1,], col="red", ylim=c(0,1))
points(years, props[2,], col="blue")
points(years, props[3,], col="green")
fitted <- predict(mmod, newdata=list(year=years), type="probs")
lines(years, fitted[,1], col="red")
lines(years, fitted[,2], col="blue")
lines(years, fitted[,3], col="green")
```




```r
predict(mmod, newdata=list(years=25), type="probs")
```

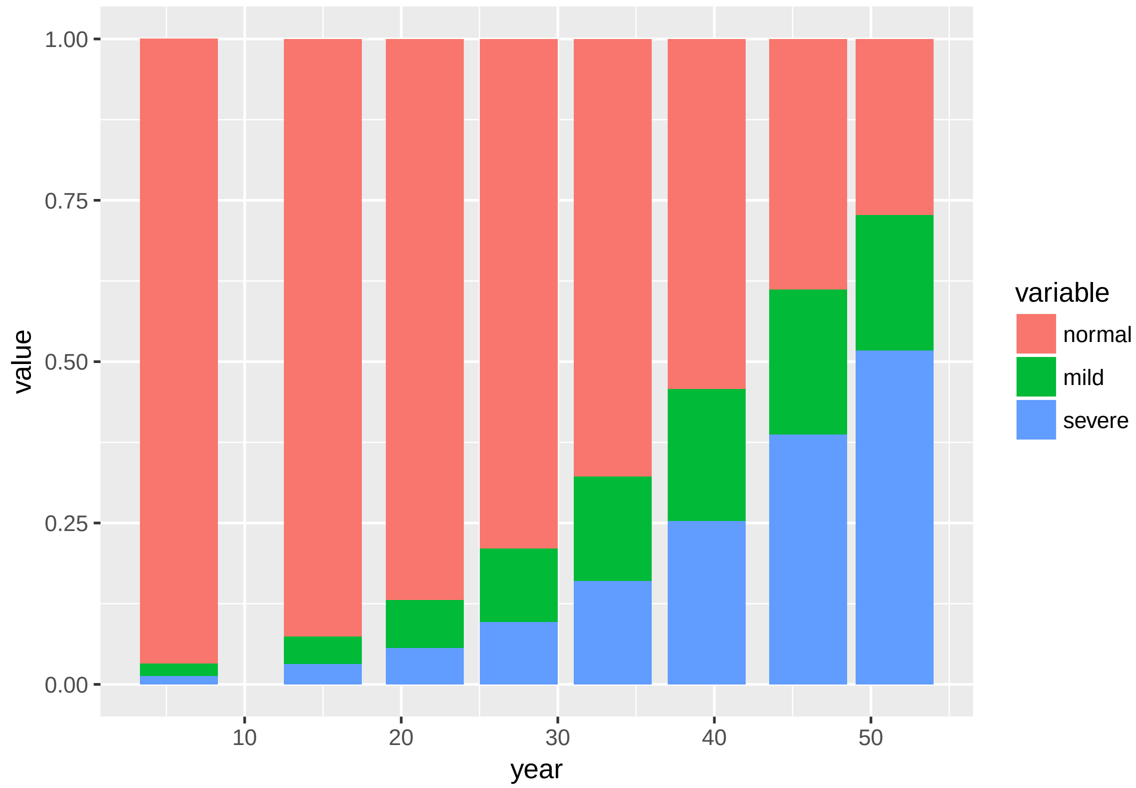
```
##     normal       mild     severe
## 0.82778727 0.09148803 0.08072470
```

```r
premmod<-predict(mmod,type="probs")
resmmod<-t(props)- premmod
par(mfrow=c(1,3))
binnedplot(premmod[,1],resmmod[,1])
binnedplot(premmod[,2],resmmod[,2])
binnedplot(premmod[,3],resmmod[,3])
```

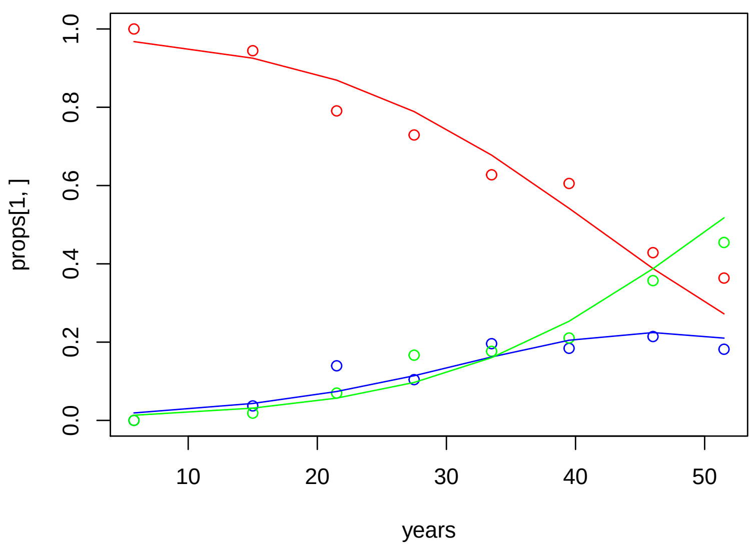


2. Repeat the analysis with the pneumonoconiosis status being treated as ordinal.

```
pneumo2 <- data.frame(status = rep(pneumo$status, pneumo$Freq),
                      year = rep(pneumo$year, pneumo$Freq))
pneumo2$status <- ordered(pneumo2$status, levels=c("normal", "mild", "severe"))
library(MASS)
omod <- polr(status ~ year, pneumo2)
xx<-data.frame(year=unique(pneumo$year),predict(omod,newdata=list(year=unique(pneumo$year)),type="prob"
ggplot(melt(xx,id.vars="year"))+geom_bar(stat="identity")+aes(x=year,y=value,fill=variable)
```




```
plot(years, props[1,], col="red", ylim=c(0,1))
points(years, props[2,], col="blue")
points(years, props[3,], col="green")
fitted <- predict(omod, newdata=list(year=years), type="probs")
lines(years, fitted[,1], col="red")
lines(years, fitted[,2], col="blue")
lines(years, fitted[,3], col="green")
```

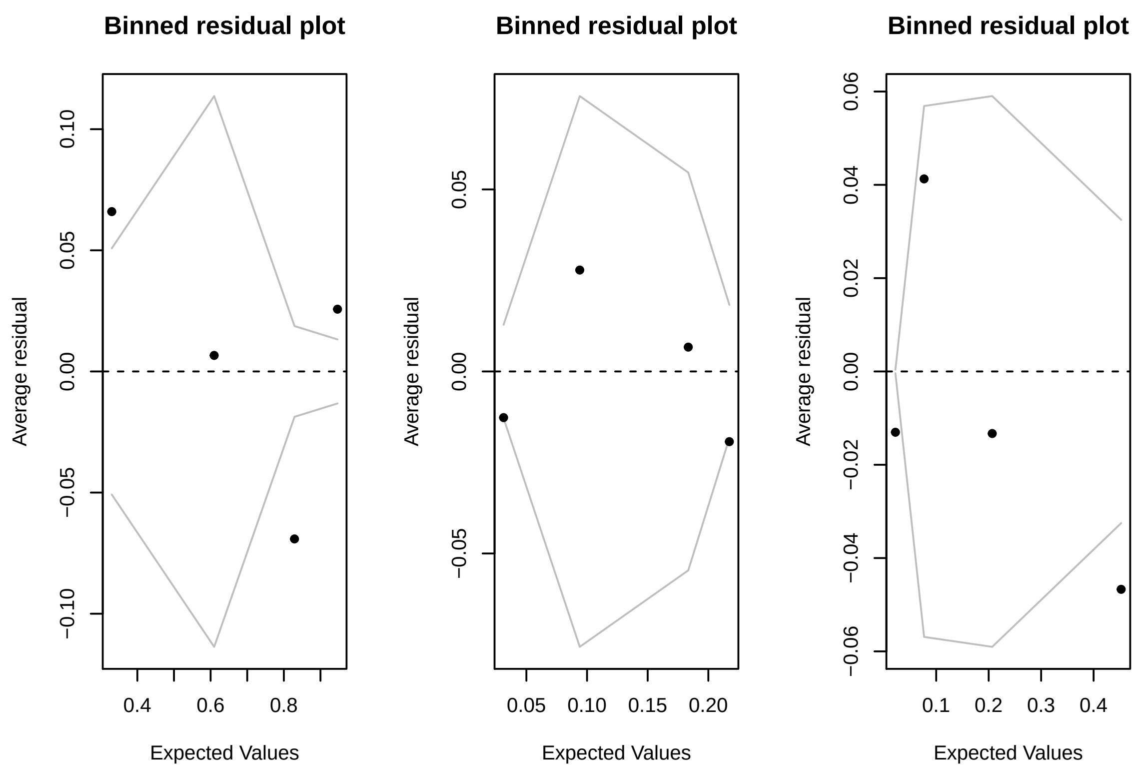



```
predict(omod, newdata=list(year=25), type="probs")
```

```
##     normal       mild     severe
## 0.82610096 0.09601474 0.07788430
```

residual

```r
resomod<-t(props)- fitted
par(mfrow=c(1,3))
binnedplot(fitted[,1],resomod[,1])
binnedplot(fitted[,2],resomod[,2])
binnedplot(fitted[,3],resomod[,3])
```



3.Now treat the response variable as hierarchical with top level indicating whether the miner has the disease and the second level indicating, given they have the disease, whether they have a moderate or severe case.

```r
pneumo3<- data.frame(normal=pneumo[pneumo$status == "normal","Freq"],
                     disease=pneumo[pneumo$status == "mild","Freq"]+
                             pneumo[pneumo$status == "severe","Freq"],
                     mild=pneumo[pneumo$status == "mild","Freq"],
                     severe=pneumo[pneumo$status == "severe","Freq"],
                     year=pneumo[pneumo$status == "mild","year"])

binmodw <- glm(cbind(disease,normal) ~ year, data=pneumo3,family=binomial)
binmodd <- glm(cbind(severe, mild) ~ year, data=pneumo3, family = binomial)
predict(binmodw,data=pneumo3,type="response")
```

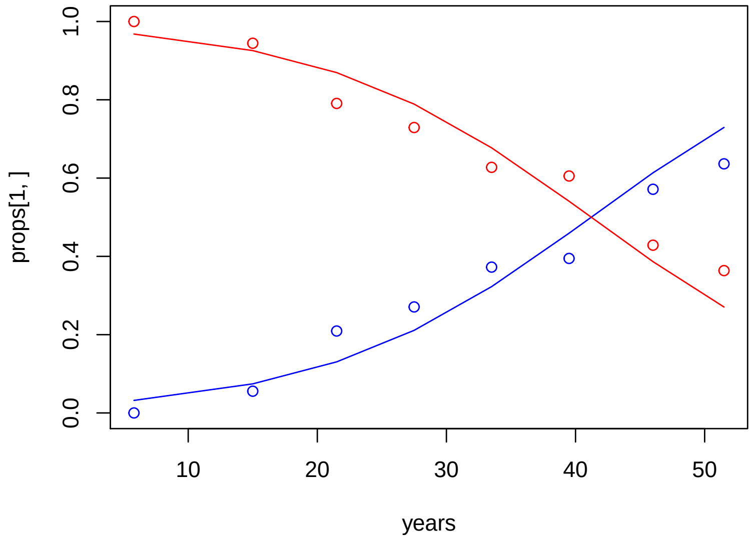
```
##          1          2          3          4          5          6
## 0.03204667 0.07430865 0.13049793 0.21099340 0.32271286 0.45916195
##          7          8
## 0.61349640 0.72938688
```

```r
predict(binmodd,data=pneumo3,type="response")
```

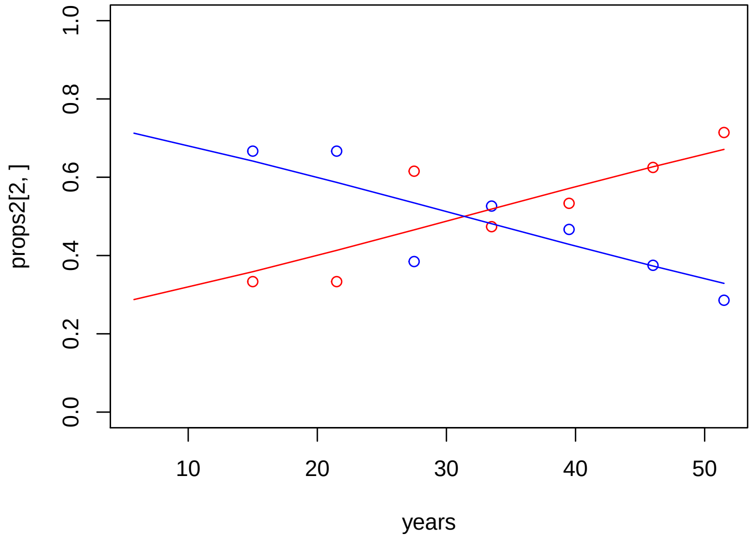
```
##         1         2         3         4         5         6         7
## 0.2874736 0.3586230 0.4131935 0.4655674 0.5187118 0.5714361 0.6267453
##         8
```

```
## 0.6711462
```

```r
plot(years, props[1,], col="red", ylim=c(0,1))
points(years, props[2,]+props[3,], col="blue")
fitted <- predict(binmodw,data=pneumo3,type="response")
lines(years, fitted, col="blue")
lines(years, 1-fitted, col="red")
```




```r
props2 <- prop.table(counts[2:3,], 2)
plot(years, props2[2,], col="red", ylim=c(0,1))
points(years, props2[1,], col="blue")
fitted2 <- predict(binmodd,data=pneumo3,type="response")
lines(years, fitted2, col="red")
lines(years, 1-fitted2, col="blue")
```




```r
predict(binmodw, newdata=list(year=25), type="response")
```

```
##        1
## 0.173701
```

```r
predict(binmodd, newdata=list(year=25), type="response")
```

```
##         1
## 0.4435842
```

```r
np<-1-predict(binmodw, newdata=list(year=25), type="response")
sp<-predict(binmodw, newdata=list(year=25), type="response")*predict(binmodd, newdata=list(year=25), typ
mp<-1-np-sp
c(np,mp,sp)
```

```
##          1          1          1
## 0.82629896 0.09664999 0.07705105
```

4. Compare the three analyses.

If you look at the year 25 we see that the predicted value is

```r
predict(mmod, newdata=list(years=25), type="probs")
```

```
##     normal       mild     severe
## 0.82778727 0.09148803 0.08072470
```

```r
predict(omod, newdata=list(year=25), type="probs")
```

```
##     normal       mild     severe
## 0.82610096 0.09601474 0.07788430
```

```r
c(np,mp,sp)
```

```
##          1          1          1
## 0.82629896 0.09664999 0.07705105
```

But all of the predicted value are fairly off from the observed values

```r
props
```

```
##         year
## status          5.8         15       21.5       27.5       33.5       39.5
##   normal 1.00000000 0.94444444 0.79069767 0.72916667 0.62745098 0.60526316
##   mild   0.00000000 0.03703704 0.13953488 0.10416667 0.19607843 0.18421053
##   severe 0.00000000 0.01851852 0.06976744 0.16666667 0.17647059 0.21052632
##         year
## status           46       51.5
##   normal 0.42857143 0.36363636
##   mild   0.21428571 0.18181818
##   severe 0.35714286 0.45454545
```

If we calculate the sum of the squared residuals we get For nominal

```r
sum((t(props)-predict(mmod, type="probs"))^2)
```

```
## [1] 0.05315203
```

For ordinal

```r
sum((t(props)-predict(omod,  newdata=list(year= unique(pneumo$year)), type="probs"))^2)
```

```
## [1] 0.04734851
```

For two stage

```
np<-1-predict(binmodw,  type="response")
sp<-predict(binmodw,  type="response")*predict(binmodd, type="response")
mp<-1-np-sp
sum((t(props)-cbind(np,mp,sp))^2)
```

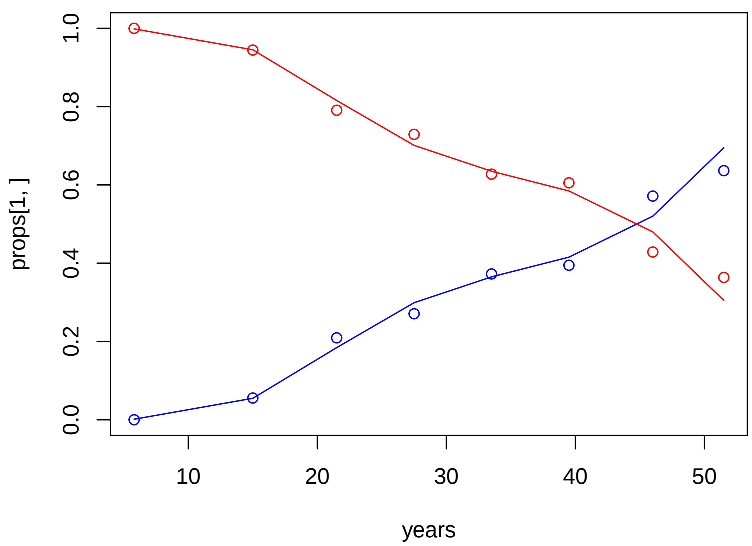
```
## [1] 0.04813249
```

Which seems to suggest that the ordinal model has the smallest residual.

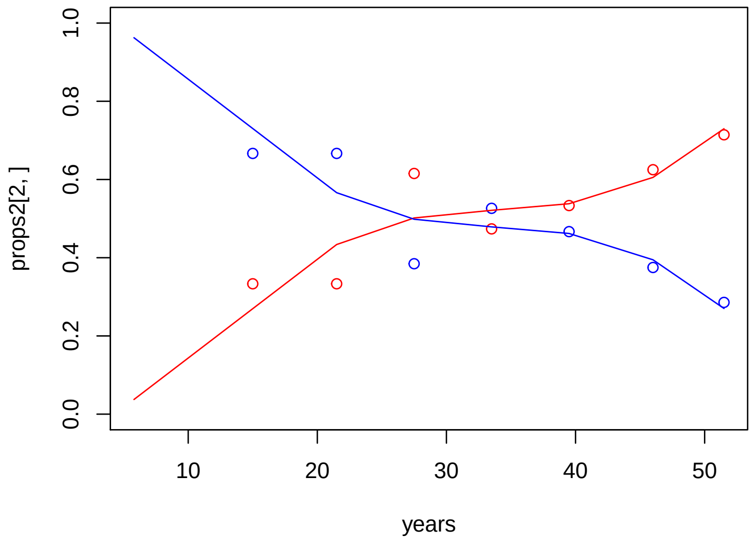
How about we add nonlinear effect for the years?

```
binmodw2 <- glm(cbind(disease,normal) ~ poly(year,3), data=pneumo3,family=binomial)
binmodd2 <- glm(cbind(severe, mild) ~  poly(year,3), data=pneumo3, family = binomial)
```

```
plot(years, props[1,], col="red", ylim=c(0,1))
points(years, props[2,]+props[3,], col="blue")
fitted <- predict(binmodw2,data=pneumo3,type="response")
lines(years, fitted, col="blue")
lines(years, 1-fitted, col="red")
```




```
props2 <- prop.table(counts[2:3,], 2)
plot(years, props2[2,], col="red", ylim=c(0,1))
points(years, props2[1,], col="blue")
fitted2 <- predict(binmodd2,data=pneumo3,type="response")
lines(years, fitted2, col="red")
lines(years, 1-fitted2, col="blue")
```




```
np<-1-predict(binmodw2,  newdata=pneumo3,type="response")
sp<-predict(binmodw2, newdata=pneumo3, type="response")*predict(binmodd2,newdata=pneumo3, type="respons
mp<-1-np-sp
sum((t(props)-cbind(np,mp,sp))^2)
```

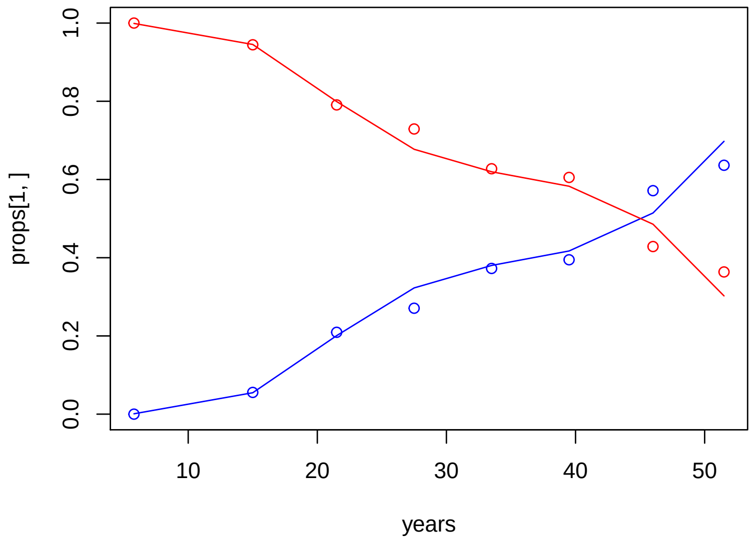
```
## [1] 0.01718263
```

What we see here is that the on the first level it seem to fit the trend much better but on the second level it does much worse. Looking back into the data you see that at year 27.5 the proportion of mild to sever reverses some how and that seems to make the estimation harder. It is questionable whether there is potential mislabeling of the data or if this is the level of variability in the data.

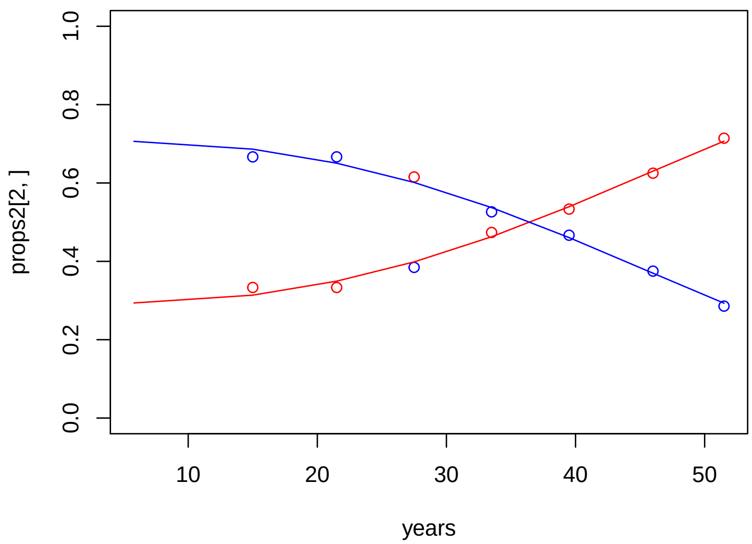
If we remove the year 27.5 and refit, we can achive much consitent trend.

```
binmodw3 <- glm(cbind(disease,normal) ~ poly(year,3), data=pneumo3[pneumo3$year!=27.5,],family=binomial
binmodd3 <- glm(cbind(severe, mild) ~  poly(year,3), data=pneumo3[pneumo3$year!=27.5,], family = binomi
```

```
plot(years, props[1,], col="red", ylim=c(0,1))
points(years, props[2,]+props[3,], col="blue")
fitted <- predict(binmodw3,newdata=pneumo3,type="response")
lines(years, fitted, col="blue")
lines(years, 1-fitted, col="red")
```




```
props2 <- prop.table(counts[2:3,], 2)
plot(years, props2[2,], col="red", ylim=c(0,1))
points(years, props2[1,], col="blue")
fitted2 <- predict(binmodd3,newdata=pneumo3,type="response")
lines(years, fitted2, col="red")
lines(years, 1-fitted2, col="blue")
```



However, due to the large discrepancy at 27.5 our residuals increase.

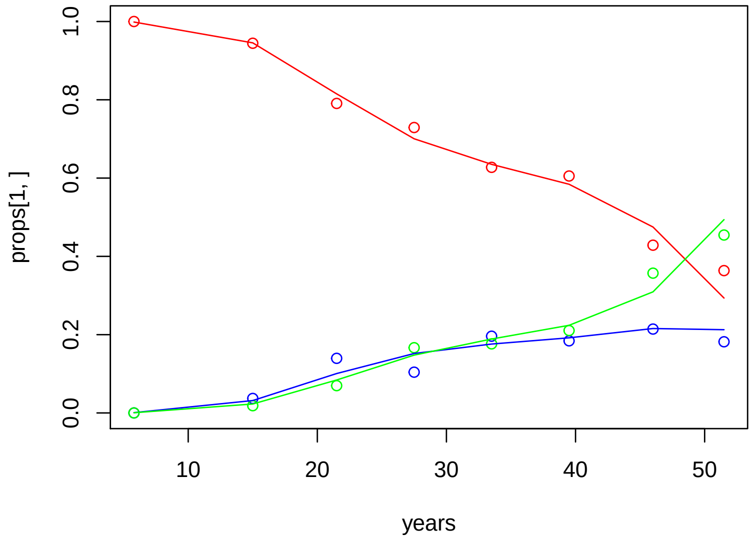
```
np<-1-predict(binmodw3,  newdata=pneumo3,type="response")
sp<-predict(binmodw3, newdata=pneumo3, type="response")*predict(binmodd,newdata=pneumo3, type="response
mp<-1-np-sp
sum((t(props)-cbind(np,mp,sp))^2)
```

```
## [1] 0.02150756
```

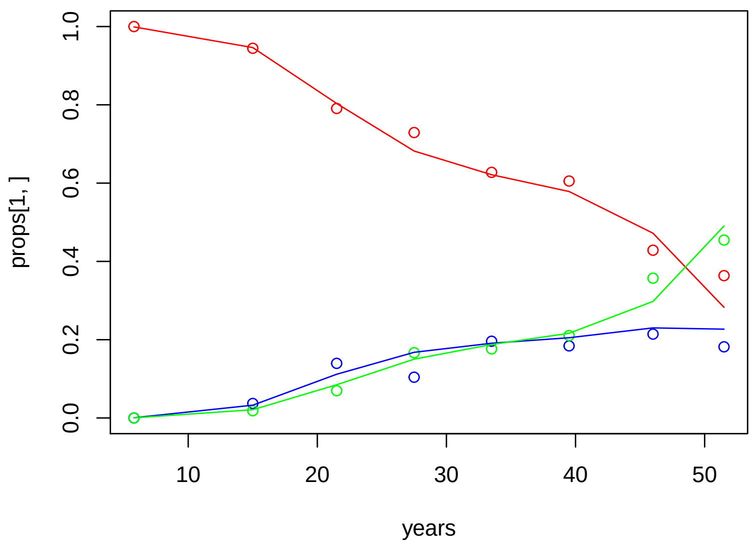
When we remove year 27.5 and refit the ordered model with nonliner trend we also see the similar result.

```
omod2 <- polr(status ~ poly(year,3), pneumo2)
omod3 <- polr(status ~ poly(year,3), pneumo2[pneumo2$year!=27.5,])
plot(years, props[1,], col="red", ylim=c(0,1))
```

```r
points(years, props[2,], col="blue")
points(years, props[3,], col="green")
fitted2 <- predict(omod2, newdata=list(year=years), type="probs")
lines(years, fitted2[,1], col="red")
lines(years, fitted2[,2], col="blue")
lines(years, fitted2[,3], col="green")
```




```r
plot(years, props[1,], col="red", ylim=c(0,1))
points(years, props[2,], col="blue")
points(years, props[3,], col="green")
fitted3 <- predict(omod3, newdata=list(year=years), type="probs")
lines(years, fitted3[,1], col="red")
lines(years, fitted3[,2], col="blue")
lines(years, fitted3[,3], col="green")
```




```r
sum((t(props)-predict(omod2,  newdata=list(year= unique(pneumo$year)), type="probs"))^2)
```

```
## [1] 0.01896773
```

```r
sum((t(props)-predict(omod3,  newdata=list(year= unique(pneumo$year)), type="probs"))^2)
```

```
## [1] 0.02464971
```

We can compare the AIC of the two models with that use the same data but we are not warranted to use AIC when the data is different.

```r
AIC(omod)
```

```
## [1] 422.9188
```

```r
AIC(omod2)
```

```
## [1] 416.3863
```

On the other hand BIC does increase with the more complex model, which suggests overfitting.

```r
BIC(omod)
```

```
## [1] 434.6674
```

```r
BIC(omod2)
```

```
## [1] 435.9673
```

(optional) Multinomial choice models:

Pardoe and Simonton (2006) fit a discrete choice model to predict winners of the Academy Awards. Their data are in the folder academy.awards.

name	description
No	unique nominee identifier
Year	movie release year (not ceremony year)
Comp	identifier for year/category
Name	short nominee name
PP	best picture indicator
DD	best director indicator
MM	lead actor indicator
FF	lead actress indicator
Ch	1 if win, 2 if lose
Movie	short movie name
Nom	total oscar nominations
Pic	picture nom
Dir	director nom
Aml	actor male lead nom
Afl	actor female lead nom
Ams	actor male supporting nom
Afs	actor female supporting nom
Scr	screenplay nom
Cin	cinematography nom
Art	art direction nom
Cos	costume nom
Sco	score nom
Son	song nom
Edi	editing nom
Sou	sound mixing nom

name	description
For	foreign nom
Anf	animated feature nom
Eff	sound editing/visual effects nom
Mak	makeup nom
Dan	dance nom
AD	assistant director nom
PrNl	previous lead actor nominations
PrWl	previous lead actor wins
PrNs	previous supporting actor nominations
PrWs	previous supporting actor wins
PrN	total previous actor/director nominations
PrW	total previous actor/director wins
Gdr	golden globe drama win
Gmc	golden globe musical/comedy win
Gd	golden globe director win
Gm1	golden globe male lead actor drama win
Gm2	golden globe male lead actor musical/comedy win
Gf1	golden globe female lead actor drama win
Gf2	golden globe female lead actor musical/comedy win
PGA	producer's guild of america win
DGA	director's guild of america win
SAM	screen actor's guild male win
SAF	screen actor's guild female win
PN	PP*Nom
PD	PP*Dir
DN	DD*Nom
DP	DD*Pic
DPrN	DD*PrN
DPrW	DD*PrW
MN	MM*Nom
MP	MM*Pic
MPrN	MM*PrNl
MPrW	MM*PrWl
FN	FF*Nom
FP	FF*Pic
FPrN	FF*PrNl
FPrW	FF*PrWl

1. Fit your own model to these data.
2. Display the fitted model on a plot that also shows the data.
3. Make a plot displaying the uncertainty in inferences from the fitted model.