Practice_2

P4

```r
summary(m7)
```

```
## Call:
## multinom(formula = PAIN ~ TH + AGE + GEN)
##
## Coefficients:
##    (Intercept)         TH         AGE         GEN
## 2 0.0003548329  1.3407833 -0.01822593 -1.18840437
## 3 0.3961681871 -0.2459782 -0.01131277 -0.65176745
## 4 1.5756946211 -1.2622678 -0.03802770  0.06707541
##
## Std. Errors:
##   (Intercept)        TH        AGE       GEN
## 2   0.9734467 0.5708325 0.02742365 0.6007834
## 3   0.9651109 0.5481900 0.02798168 0.6014755
## 4   0.9105105 0.5547571 0.02732269 0.5459158
##
## Residual Deviance: 311.1885
## AIC: 335.1885
```

```r
d0 <- deviance(m0)-deviance(m7)
d1 <- deviance(m1)-deviance(m7)
d2 <- deviance(m2)-deviance(m7)
d3 <- deviance(m3)-deviance(m7)
d4 <- deviance(m4)-deviance(m7)
d5 <- deviance(m5)-deviance(m7)
d6 <- deviance(m6)-deviance(m7)
df0 <- m7$edf-m0$edf
df1 <- m7$edf-m1$edf
df2 <- m7$edf-m2$edf
df3 <- m7$edf-m3$edf
df4 <- m7$edf-m4$edf
df5 <- m7$edf-m5$edf
df6 <- m7$edf-m6$edf
p0 <- pchisq(d0,df0,lower.tail=FALSE)
p1 <- pchisq(d1,df1,lower.tail=FALSE)
p2 <- pchisq(d2,df2,lower.tail=FALSE)
p3 <- pchisq(d3,df3,lower.tail=FALSE)
p4 <- pchisq(d4,df4,lower.tail=FALSE)
p5 <- pchisq(d5,df5,lower.tail=FALSE)
p6 <- pchisq(d6,df6,lower.tail=FALSE)
cbind(c(d0,d1,d2,d3,d4,d5,d6),c(df0,df1,df2,df3,df4,df5,df6), round(c(p0,p1,p2,p3,p4,p5,p6),3))
```

```
##            [,1] [,2]  [,3]
## [1,] 28.318980    9 0.001
## [2,]  7.921986    6 0.244
## [3,] 26.139835    6 0.000
## [4,] 23.885180    6 0.001
## [5,]  5.647379    3 0.130
## [6,]  2.088488    3 0.554
## [7,] 22.146053    3 0.000
```

```
#p5 and p1 are very large, let's compare m5 and m1
anova(m5,m1,test='Chi')
```

```
##      Model Resid. df Resid. Dev   Test    Df LR stat.   Pr(Chi)
## 1       TH       363   319.1105           NA       NA        NA
## 2 TH + GEN       360   313.2770 1 vs 2     3 5.833498 0.119998
```

```
summary(m1)
```

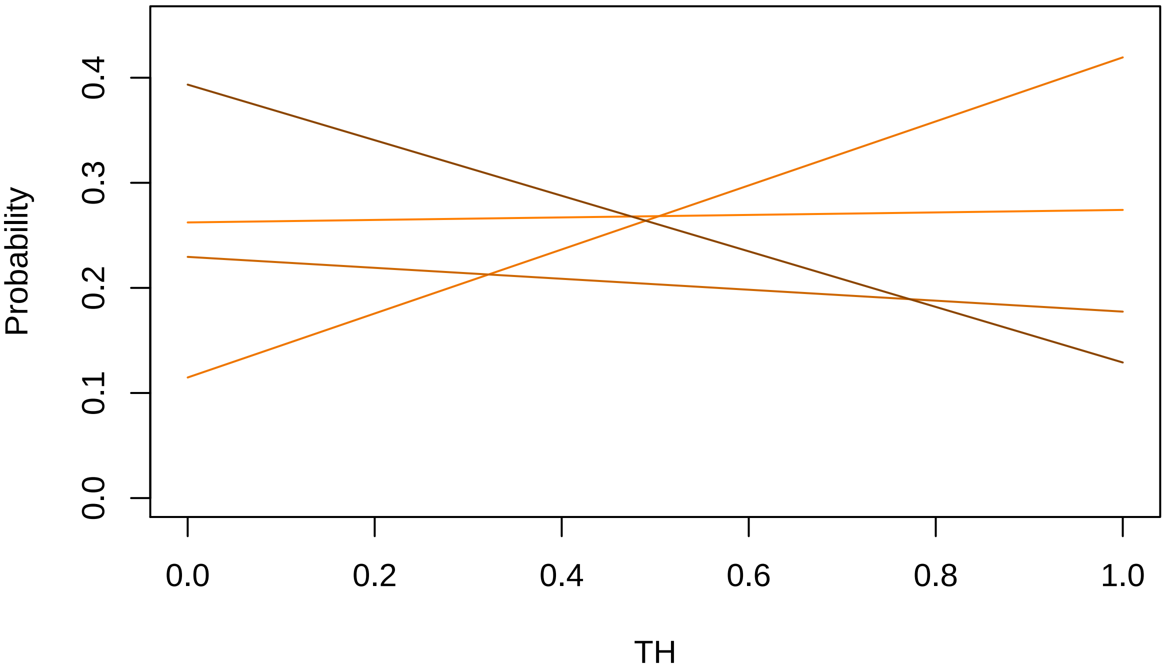
```
## Call:
## multinom(formula = PAIN ~ TH)
##
## Coefficients:
##   (Intercept)         TH
## 2  -0.8266765  1.2515469
## 3  -0.1335288 -0.3018058
## 4   0.4054703 -1.1592485
##
## Std. Errors:
##   (Intercept)        TH
## 2   0.4531638 0.5501292
## 3   0.3659628 0.5325988
## 4   0.3227487 0.5366464
##
## Residual Deviance: 319.1105
## AIC: 331.1105
```

Now for fitted probability

```
beta = coef(m1)
fit.0 = c( 1/(1+sum(exp(beta[,1]))) , exp(beta[,1])/(1+sum(exp(beta[,1]))))
fit.1 = c( 1/(1+sum(exp(beta[,1]+beta[,2]))) , exp(beta[,1]+beta[,2])/(1+sum(exp(beta[,1]+beta[,2]))))
fit = cbind(fit.0,fit.1)
colnames(fit) = c('TH=0','TH=1')
rownames(fit) = c('Pain =1','Pain =2','Pain =3','Pain =4')
fit_orginal = fit
fit
```

```
##              TH=0      TH=1
## Pain =1 0.2622943 0.2741960
## Pain =2 0.1147540 0.4193533
## Pain =3 0.2295081 0.1774180
## Pain =4 0.3934435 0.1290326
```

```
#
x = c(0,1)
plot(fit[1,]~x,type="l",col="darkorange1",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[2,],col="darkorange2",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[3,],col="darkorange3",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[4,],col="darkorange4",ylim=c(0,0.45),xlab="TH",ylab="Probability")
```



It shows how the probability of different pain levels changes as TH switches from 0 to 1. From light orange to dark orange, the color of lines corresponds to increasing pain level from 1 to 4. There are 4 pain levels, so we get 4 lines.

As we see, having treatment plays a different role in different pain level group. Compared to the placebo, if the treatment is given, the probability of pressure pain at level 4 is reduced from 0.3934435 to 0.1290326, at level 3 is reduced from 0.2295081 to 0.1774180, but at the level 2 is increased from 0.1147540 to 0.4193533, and at the level 1 is increased from 0.2622943 to 0.2741960.

And compared to the placebo group, the overall painlevel is shifted downward if the treatment is given. Though increments in percentage of pain level at level 1 and 2, the pain level percentage at 3 and 4 is decreased significantly.

P5

```r
library(VGAM)
```

```
## Loading required package: stats4

## Loading required package: splines
```

```r
v1 = vglm(ordered(PAIN)~1,data=pain,propodds(reverse = F))
v2 = vglm(ordered(PAIN)~TH,data=pain,propodds(reverse = F))
v3 = vglm(ordered(PAIN)~AGE,data=pain,propodds(reverse = F))
v4 = vglm(ordered(PAIN)~GEN,data=pain,propodds(reverse = F))
v5 = vglm(ordered(PAIN)~TH+AGE,data=pain,propodds(reverse = F))
v6 = vglm(ordered(PAIN)~TH+GEN,data=pain,propodds(reverse = F))
v7 = vglm(ordered(PAIN)~AGE+GEN,data=pain,propodds(reverse = F))
v8 = vglm(ordered(PAIN)~TH+GEN+AGE,data=pain,propodds(reverse = F))
d1 <- deviance(v1)-deviance(v8)
d2 <- deviance(v2)-deviance(v8)
d3 <- deviance(v3)-deviance(v8)
d4 <- deviance(v4)-deviance(v8)
d5 <- deviance(v5)-deviance(v8)
```

```
d6 <- deviance(v6)-deviance(v8)
d7 <- deviance(v7)-deviance(v8)
df1 <- 3
df2 <- 2
df3 <- 2
df4 <- 2
df5 <- 1
df6 <- 1
df7 <- 1
p1 <- pchisq(d1,df1,lower.tail=FALSE)
p2 <- pchisq(d2,df2,lower.tail=FALSE)
p3 <- pchisq(d3,df3,lower.tail=FALSE)
p4 <- pchisq(d4,df4,lower.tail=FALSE)
p5 <- pchisq(d5,df5,lower.tail=FALSE)
p6 <- pchisq(d6,df6,lower.tail=FALSE)
p7 <- pchisq(d7,df7,lower.tail=FALSE)
```

```
cbind(c(d1,d2,d3,d4,d5,d6,d7),c(df1,df2,df3,df4,df5,df6,df7), round(c(p1,p2,p3,p4,p5,p6,p7),3))
```

```
##              [,1] [,2]  [,3]
## [1,] 1.016902e+01    3 0.017
## [2,] 2.227252e+00    2 0.328
## [3,] 9.293938e+00    2 0.010
## [4,] 1.006251e+01    2 0.007
## [5,] 4.307456e-06    1 0.998
## [6,] 2.184210e+00    1 0.139
## [7,] 9.254034e+00    1 0.002
```

```
p_5_2 <- pchisq(deviance(v2)-deviance(v5),1,lower.tail=FALSE)
p_5_2
```

```
## [1] 0.1355952
```

```
p_2_1 <- pchisq(deviance(v1)-deviance(v2),1,lower.tail=FALSE)
p_2_1
```

```
## [1] 0.004830664
```

```
summary(v2)
```

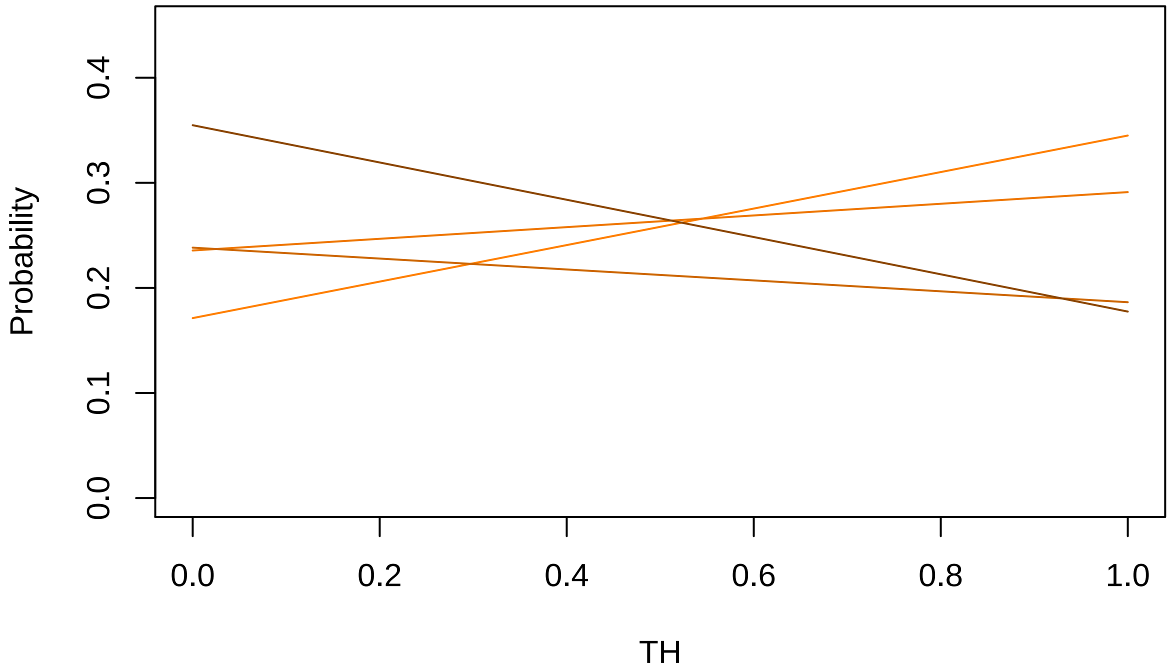
```
##
## Call:
## vglm(formula = ordered(PAIN) ~ TH, family = propodds(reverse = F),
##     data = pain)
##
## Pearson residuals:
##                        Min      1Q  Median     3Q   Max
## logitlink(P[Y<=1]) -1.108 -0.8968 -0.2653 1.3113 2.120
## logitlink(P[Y<=2]) -1.749 -0.5964  0.3668 1.0486 1.494
## logitlink(P[Y<=3]) -2.051 -1.2676  0.2696 0.4545 1.244
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -1.5767     0.2845  -5.543 2.98e-08 ***
## (Intercept):2  -0.3769     0.2472  -1.525  0.12734
## (Intercept):3   0.5979     0.2519   2.374  0.01761 *
## TH              0.9356     0.3321   2.817  0.00484 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2]),
## logitlink(P[Y<=3])
##
## Residual deviance: 331.5657 on 365 degrees of freedom
##
## Log-likelihood: -165.7828 on 365 degrees of freedom
##
## Number of Fisher scoring iterations: 5
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##       TH
## 2.548732
```

```
beta = coefficients(v2)
x = c(0,1)
cdf1 = exp(beta[1]+beta[4]*x)/(1+exp(beta[1]+beta[4]*x))
cdf2 = exp(beta[2]+beta[4]*x)/(1+exp(beta[2]+beta[4]*x))
cdf3 = exp(beta[3]+beta[4]*x)/(1+exp(beta[3]+beta[4]*x))
prob1 = cdf1
prob2 = cdf2-cdf1
prob3 = cdf3-cdf2
prob4 = 1-cdf3

fit = rbind(prob1,prob2,prob3,prob4)
colnames(fit) = c('TH=0','TH=1')
rownames(fit) = c('Pain =1','Pain =2','Pain =3','Pain =4')
fit
```

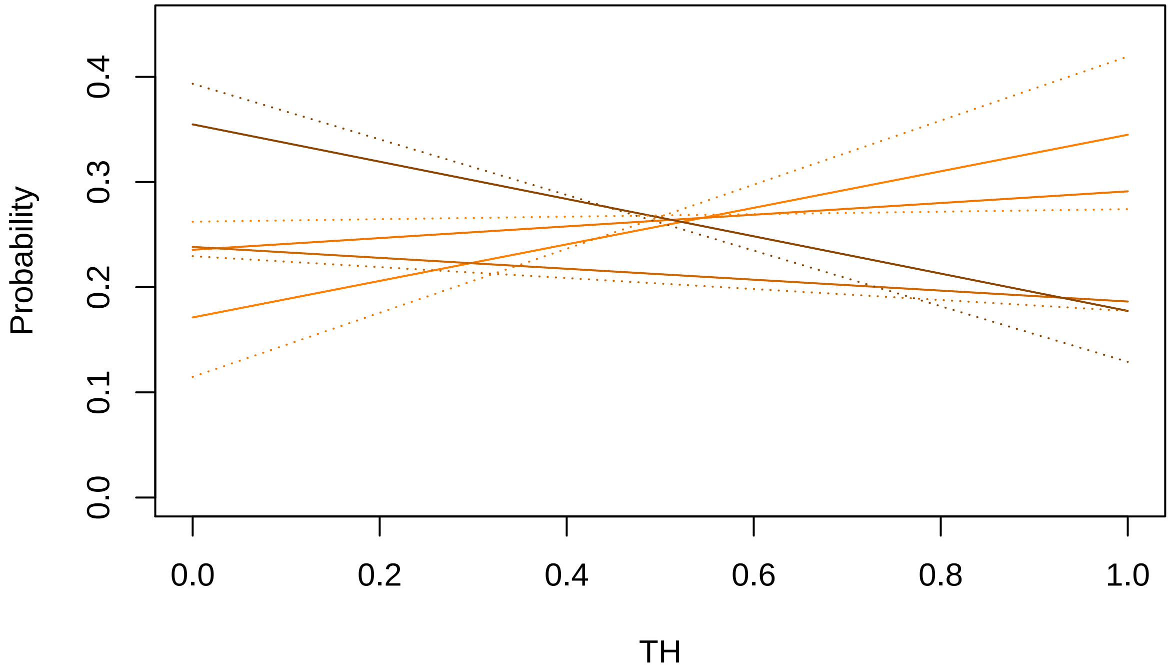
```
##              TH=0      TH=1
## Pain =1 0.1712587 0.3449896
## Pain =2 0.2356186 0.2911636
## Pain =3 0.2383062 0.1863690
## Pain =4 0.3548165 0.1774778
```

```
x = c(0,1)
plot(fit[1,]~x,type="l",col="darkorange1",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[2,],col="darkorange2",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[3,],col="darkorange3",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[4,],col="darkorange4",ylim=c(0,0.45),xlab="TH",ylab="Probability")
```



```
plot(fit[1,]~x,type="l",col="darkorange1",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[2,],col="darkorange2",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[3,],col="darkorange3",ylim=c(0,0.45),xlab="TH",ylab="Probability")
lines(x,fit[4,],col="darkorange4",ylim=c(0,0.45),xlab="TH",ylab="Probability")

fit = fit_orginal
lines(x,fit[1,],lty=3,col="darkorange1")
lines(x,fit[2,],lty=3,col="darkorange2")
lines(x,fit[3,],lty=3,col="darkorange3")
lines(x,fit[4,],lty=3,col="darkorange4")
```



The dashed line corresponds the result we obtain from the baseline category model. We observe the same phenomenon that having treatment plays a different role in different pain level group, and the result from this two model gives the similar conclusion about the effect of TH on the probaility in reducing the pain.