MAT 5196(4378)

Assignment 4

Soufiane Fadel — Pablo Señas Peón

January 5, 2020

Exercise 1

i.d)

> *Estimate a proportional odds model using fat content to predict rating. Investigate whether or not a quadratic term is helpful.*

We remember that the proportional odds model has the form

$$\text{logit}(\mathbb{P}(Y \leq j)) = \log\left(\frac{\mathbb{P}(Y \leq j)}{\mathbb{P}(Y > j)}\right) = \beta_{j0} - \eta(x) \qquad \eta(x) = \beta_1 x_1 + \cdots + \beta_p x_p.$$

In this case, Y is the rating of the ice cream, that takes values in $\{1, \cdots, 9\}$, and the explanatory variable will be the fat content. The proportional odds is a special model that assumes that the effect of the fat content has the same effect in all cumulative probabilities of the rating ice cream. We can estimate the coefficients with the function *polr*, from the MASS package. We note that our response variable needs to be an ordered factor.

R Code:

```
library(MASS)
data<-read.csv("ice_cream.csv")
data$rating<-ordered(data$rating)
mod1<-polr(rating~fat,data=data,weights = count)
summary(mod1)
```

Output:

```
Re-fitting to get Hessian

Call:
polr(formula = rating ~ fat, data = data, weights = count)

Coefficients:
     Value Std. Error t value
fat 0.8855 0.8807 1.006

Intercepts:
    Value Std. Error t value
1|2 -3.6669 0.3266 -11.2281
2|3 -2.1356 0.1941 -11.0006
3|4 -1.5112 0.1712 -8.8263
4|5 -0.7415 0.1564 -4.7401
5|6 -0.4307 0.1536 -2.8035
```

```
6|7 0.2367 0.1521 1.5557
7|8 1.1899 0.1601 7.4301
8|9 3.0191 0.2367 12.7545

Residual Deviance: 1998.965
AIC: 2016.965
```

As a consequence, our estimation for the proportional odds model is

$$\text{logit}(\mathbb{P}(Y \leq j)) = \log\left(\frac{\mathbb{P}(Y \leq j)}{\mathbb{P}(Y > j)}\right) = \beta_{j0} - 0.8855x_1; \qquad \begin{bmatrix}\beta_{10}\\ \vdots \\ \beta_{90}\end{bmatrix} = \begin{bmatrix}-3.6669\\ -2.1356\\ -1.5112\\ -0.7415\\ -0.4307\\ 0.2367\\ 1.1899\\ 3.0191\end{bmatrix}$$

We are now asked to investigate if a quadratic term is helpful. A proportional odds model with quadratic term would have the form

$$\text{logit}(\mathbb{P}(Y \leq j)) = \beta_{j0} - \beta_1 x_1 - \beta_2 x_1^2$$

To do this, we just have to change the formula that we introduce in *polr*, adding the term $I(fat^2)$. To investigate if the quadratic term is helpful or not we will compare the deviances of the first model and the model with the quadratic term. Knowing that the difference of the deviance follows a χ^2 distribution, we also have to take into account the degrees of freedom. This could be done manually using mod$df and mod$deviance, but we will use the function anova to compute a p-value for the test

$$H_0 : \text{quadratic term is not helpful} \qquad H_1 : \text{quadratic term is helpful}$$

R Code:

```
mod2<-polr(rating~fat+I(fat^2),weights=count,data=data)
anova(mod1,mod2,test="Chisq") #Recall: mod1 does not have quadratic term
```

Output:

```
Likelihood ratio tests of ordinal regression models

Response: rating
          Model Resid. df Resid. Dev Test Df LR stat. Pr(Chi)
1 fat 487 1998.965
2 fat + I(fat^2) 486 1899.452 1 vs 2 1 99.51271 0
```

The numerical p-value obtained is 0, and as a consequence we reject H_0 and state: **the quadratic term is helpful**. We must remark that this value must not surprise us. The difference of the deviances is almost 100 with just 1 degree of freedom. A χ^2 distribution with 1 degree of freedom is the square of a normal distribution, and thence the p-value associated is so small that it is considered as 0.

i.e)

Perform a LRT test to assess the proportional odds assumption.

The proportional odds simplify our model a lot, but as we can note, the assumption done for the proportional odds model is very strong and cannot be used without having some certainty that it is reasonable to do so. We will test it against the full model, which has the form

$$\text{logit}(\mathbb{P}(Y \leq j)) = \beta_{j0} - \beta_{j1}x_1 \qquad j = 1, \cdots, 9 \qquad \text{(Non proportional odds model)}.$$

To build the non proportional odds model we must use the function *vglm()* (vector generalized linear models), which comes from the homonymous package. We can also build the proportional odds model with that function. We will finally compare both models, again, with *anova()*, although this can also be done manually. As we did in class, we will specify *type=1*. However, if we take a look at the data:

R Code:

```
head(data)
```

Output:

```
fat rating count
1 0 1 4
2 0 2 17
3 0 3 8
4 0 4 16
5 0 5 5
6 0 6 6
```

We see that it is not fully expanded. In order to use *vglm()* we have to expand it and repeat each measure count-times. We can do this with *expandRows()*, from splitstackshape package.

R Code:

```
library(VGAM)
library(splitstackshape)
data<-expandRows(data, "count")
## non proportional odds model
mod.ord.nonpar<-vglm(rating~fat,family=cumulative(parallel=FALSE),data=data)

## proportional odds model
mod.ord.par<-vglm(rating~fat,family=cumulative(parallel=TRUE),data=data)
mod.ord.par
anova(mod.ord.par,mod.ord.nonpar,type=1)
```

Output:

```
#This just indicates that some counts are 0
The following rows have been dropped from the input:

18, 19, 37, 46

#We will now show the coefficients of the proportional odds model built with vglm
     to check that it gives the same values as polr.
Call:
vglm(formula = rating ~ fat, family = cumulative(parallel = TRUE),
    data = data)


Coefficients:
```

```
(Intercept):1 (Intercept):2 (Intercept):3 (Intercept):4 (Intercept):5 (Intercept)
    :6
   -3.6668922 -2.1356553 -1.5112811 -0.7415236 -0.4307319 0.2366083
(Intercept):7 (Intercept):8 fat
    1.1898101 3.0190162 -0.8851452

Degrees of Freedom: 3968 Total; 3959 Residual
Residual deviance: 1998.965
Log-likelihood: -999.4825
> anova(mod.ord.par,mod.ord.nonpar,type=1)
Analysis of Deviance Table

Model 1: rating ~ fat
Model 2: rating ~ fat
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 3959 1999.0
2 3952 1989.8 7 9.2042 0.2383
```

The p-value (0.2383) is big enough, and as a consequence **we do not have enough evidence against the proportional odds model**.

ii)

Use a proportional odds model as a broken line regression to find the break point between low levels and high levels of fat content. Is this break point the optimal fat content? (Use odds ratios to help answer the latter question.)

Broken line regression is a good alternative to introducing a quadratic term because it allows us to maintain the interpretation for the coefficients, which we could not do otherwise. The proportional odds model as a broken line regression has the form

$$\text{logit}(\mathbb{P}(Y \leq j)) = \beta_{j0} - \beta_1 x_1 - \beta_2 \delta_b (x - b) \quad \delta_b = \begin{cases} 1 & x \geq b \\ 0 & x \leq b \end{cases} \quad \text{(model continuous at b)},$$

where b is some point in the range of the fat content values that we have yet to estimate. To do this we will maximize the log likelihood of the model, as a function of b. The function *optim()* minimizes, so our objective function will be the opposite of the log likelihood. This is equivalent to maximizing it.

R Code for estimating b:

```
#Broken line term
Id.Cent<-function(x){ (ifelse(test=(x>=b),yes=1,no=0))*(x-b) }
#Opposite of log likelihood
neg.profile.log.lik<-function(b)
{
  Id.Cent<-function(x){ (ifelse(test=(x>=b),yes=1,no=0))*(x-b) }
  mod<-polr(rating~ fat+I(Id.Cent(fat)),
          weights=count,data=data,method="logistic")
  return(-1*as.numeric(logLik(mod)))
 }
#We set up the range where the function has to look for
#the optimal value and the initial point.
lower<-min(data$fat)
upper<-max(data$fat)
initial<-mean(data$fat)
result<-optim(par = initial,lower=lower,upper=upper,
```

```
            fn=neg.profile.log.lik,method="Brent")


b<-result$par
b
```

Output:

```
> b
[1] 0.06354922
```

With out estimate $\hat{b} = 0.0635$ we will now build the proportional odds model as broken line regression.

R Code:

```
mod<-polr(rating~ fat+I(Id.Cent(fat)),
          weights=count,data=data,method="logistic")
summary(mod)
mod$coefficients
```

Output:

```
Re-fitting to get Hessian

Call:
polr(formula = rating ~ fat + I(Id.Cent(fat)), data = data, weights = count,
    method = "logistic")

Coefficients:
                Value Std. Error t value
fat 48.62 4.897 9.928
I(Id.Cent(fat)) -57.19 5.790 -9.877

Intercepts:
    Value Std. Error t value
1|2 -2.3963 0.3406 -7.0346
2|3 -0.7674 0.2278 -3.3691
3|4 -0.0633 0.2175 -0.2909
4|5 0.8442 0.2214 3.8135
5|6 1.2199 0.2256 5.4070
6|7 2.0050 0.2353 8.5223
7|8 3.0724 0.2503 12.2760
8|9 4.9803 0.3107 16.0271

Residual Deviance: 1895.082
AIC: 1915.082
> mod$coefficients
           fat I(Id.Cent(fat))
      48.61517 -57.18895
```

$$\text{logit}(\mathbb{P}(Y \leq j)) = \log\left(\frac{\mathbb{P}(Y \leq j)}{\mathbb{P}(Y > j)}\right) = \beta_{j0} - (48.6152x_1 - 57.1890\delta_b(x-b)); \quad \begin{bmatrix}\beta_{10}\\ \vdots \\ \beta_{90}\end{bmatrix} = \begin{bmatrix}-2.3963\\ -0.7674\\ -0.0633\\ 0.8422\\ 1.2199\\ 2.0050\\ 3.0724\\ 4.9803\end{bmatrix}$$

We are now going to compute odds ratio conditioned on a c increment in the fat content. Usually we would take $c = 1$, but taking into account that fat content is in the range $(0, 0.28)$ we will do the discussion with an arbitrary $c > 0$:

$$\frac{\mathbb{P}(Y > j|X = x+c)/\mathbb{P}(Y \leq j|X = x+c)}{\mathbb{P}(Y > j|X = x)/\mathbb{P}(Y \leq j|X = x)} = \frac{\exp(-\beta_{j0} + \beta_1(x+c) + \beta_2\delta_b(x+c-b))}{\exp(-\beta_{j0} + \beta_1(x) + \beta_2\delta_b(x-b))} = \exp(c(\beta_1 + \delta_b\beta_2))$$

We finally have that

$$\begin{aligned}\exp(c(\beta_1 + \delta_b\beta_2)) &= \begin{cases}\exp(c\beta_1) & x < b\\ \exp(c(\beta_1+\beta_2)) & x \geq b\end{cases}\\ &= \begin{cases}\exp(c \times 48.61517) & x < b\\ \exp(c(48.6151 - 57.18895)) & x \geq b\end{cases}\\ &= \begin{cases}\exp(c \times 48.61517) & x < b\\ \exp(-c \times 8.5738) & x \geq b\end{cases}\\ &= \begin{cases}\geq 1 & x < b\\ \leq 1 & x \geq b\end{cases}\end{aligned}$$

We see that when $x < b$, $c\beta_1 > 0$ and the odds when the fat content increases are going in the positive direction of the rating. On the other way, when $x > b$, $c(\beta_1 + \beta_2) < 0$ and we have the opposite interpretation. This is what we would think it happens near an optimal break point, and as a consequence we can conclude that our $\hat{b}$ is a good point estimator for the optimal break point.

Exercise 2

a)

What is the population of inference?

The population of inference is the set of the potential customers that visit Lincoln's Starbucks between 8:00 a.m and 8:30 a.m.

b)

Construct side-by-side dot plots of the data where the y-axis gives the number of customers and the x-axis is for the day of the week.

First we should change the level of the dataframe with the function factor().

R Code:

```
> data<-read.csv("starbucks.csv")
> levels(data$Day)
> data$Day<-factor(data$Day,levels=c("Monday","Tuesday" ,"Wednesday","Thursday" ,
    "Friday" ))
```

```
> levels(data$Day)
```

Output:

```
#Before
[1] "Friday" "Monday" "Thursday" "Tuesday" "Wednesday"
#After
[1] "Monday" "Tuesday" "Wednesday" "Thursday" "Friday"
```

The plot below is a **stripchart** plot that display the count of the customers according to the Days.

R Code:

```
> stripchart(Count~Day,data=data,method="jitter",vertical = TRUE, col = "red",
    main="Stripchart of the number of customers Counts")
> means <- aggregate(Count ~ Day, data, mean)
> points(1:5, means$Count, col = "blue")
```

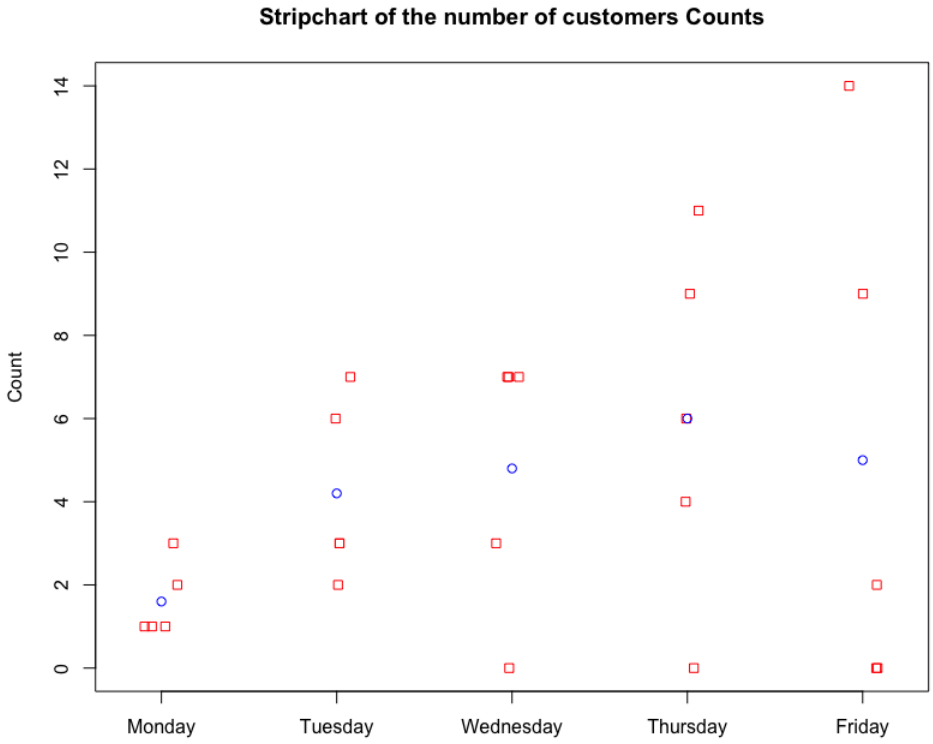

Figure 1: Stripchart of the number of customers Counts

Describe what information this plot provides regarding the mean number of customers per day. In particular, does it seem plausible that the true mean count is constant across the days?

In red we have the count of customers that entered to the Starbucks the corresponding day. Besides, the mean of each day has been plotted in blue. We can notice that the mean of the customers seems

to increase as the week goes by. Certainly, it does not seem that the true mean is constant across the days. For instance, on Monday the mean is less than "2", whereas on Thursday it is approximately "6". We cannot assure anything just looking at the plot, but it does not seem that the true mean count is constant.

c)

i)

Estimate the model using a Poisson regression model

We remember that the Poisson regression model has the form

$$\begin{aligned}\log(\mu) =&\beta_0 + \beta_1 \mathrm{I}(\text{Day=Tuesday}) + \beta_2 \mathrm{I}(\text{Day=Wednesday}) \\ &+\beta_3 \mathrm{I}(\text{Day=Thursday}) + \beta_4 \mathrm{I}(\text{Day=Friday}).\end{aligned}$$

We proceed to compute it using *glm()*.

R Code:

```
> mod<-glm(Count ~ Day , poisson(link=log), data=data)
> summary(mod)
```

Output:

```
Call:
glm(formula = Count ~ Day, family = poisson(link = log), data = data)

Deviance Residuals:
    Min 1Q Median 3Q Max
-3.4641 -0.8832 -0.5099 0.9392 3.2908

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.4700 0.3536 1.329 0.183726
DayTuesday 0.9651 0.4155 2.323 0.020188 *
DayWednesday 1.0986 0.4082 2.691 0.007123 **
DayThursday 1.3218 0.3979 3.322 0.000895 ***
DayFriday 1.1394 0.4062 2.805 0.005030 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 87.432 on 24 degrees of freedom
Residual deviance: 72.431 on 20 degrees of freedom
AIC: 150.9

Number of Fisher Scoring iterations: 5
```

$$\begin{aligned}\log(\mu) =&0.4700 + 0.9651 \mathrm{I}(\text{Day=Tuesday}) + 1.0986 \mathrm{I}(\text{Day=Wednesday}) \\ &+1.3218 \mathrm{I}(\text{Day=Thursday}) + 1.1394 \mathrm{I}(\text{Day=Friday}).\end{aligned}$$

ii)

Perform a LRT to determine if there is evidence that day of the week affects the number of customers waiting in line.

We can obtain the likelihood ratio test for the significance of the variable Day with the Anova function in the car package.

R Code:

```
> library(car)
> Anova(mod, test="LR")
```

Output:

```
Analysis of Deviance Table (Type II tests)

Response: Count
    LR Chisq Df Pr(>Chisq)
Day 15.002 4 0.004698 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is 0.0046, so we have evidence suggesting that the day of the week affects the number of customers waiting in line.

iii)

Estimate the ratio of means comparing each pair of the days of the week, and compute with control for the familywise confidence level the 95% confidence intervals for the ratio of means comparing each pair of the days of the week.

We will use pairwise comparisons to describe the effects of Day. to do so we use the function **glht** from the **multcomp** package to obtain the pairwise comparisons of the levels of the factor Day.

R Code:

```
> library(multcomp)
> mod<-glm(Count ~ Day , poisson(link=log), data=data)
> multcomp.obj1<-glht(mod,mcp(Day="Tukey"))
> exp(confint(multcomp.obj1,level = 0.95 )$confint)
```

Output:

```
                    Estimate lwr upr
Tuesday - Monday 2.6250000 0.8496641 8.109822
Wednesday - Monday 3.0000000 0.9902836 9.088306
Thursday - Monday 3.7500000 1.2730866 11.045989
Friday - Monday 3.1250000 1.0373083 9.414390
Wednesday - Tuesday 1.1428571 0.5077750 2.572247
Thursday - Tuesday 1.4285714 0.6598202 3.092989
Friday - Tuesday 1.1904762 0.5329825 2.659062
Thursday - Wednesday 1.2500000 0.5942960 2.629161
Friday - Wednesday 1.0416667 0.4794961 2.262937
Friday - Thursday 0.8333333 0.3995117 1.738233
attr(,"conf.level")
[1] 0.95
attr(,"calpha")
[1] 2.714956
```

Interpretation:

- ♣ We are estimating that the mean count at Tuesday is 163% larger than the mean count at Monday.
- ♣ We are estimating that the mean count at Wednesday is 200% larger than the mean count at Monday.
- ♣ We are estimating that the mean count at Thursday is 275% larger than the mean count at Monday.
- ♣ We are estimating that the mean count at Friday is 212% larger than the mean count at Monday.
- ♣ We are estimating that the mean count at Wednesday is 14% larger than the mean count at Tuesday.
- ♣ We are estimating that the mean count at Thursday is 42% larger than the mean count at Tuesday.
- ♣ We are estimating that the mean count at Friday is 19% larger than the mean count at Tuesday.
- ♣ We are estimating that the mean count at Thursday is 25% larger than the mean count at Wednesday.
- ♣ We are estimating that the mean count at Friday is 4% larger than the mean count at Wednesday.
- ♣ We are estimating that the mean count at Friday is 17% smaller than the mean count at Thursday.

iv)

Compute the estimated mean number of customers for each day of the week using the model. Compare these estimates to the observed means. Also, compute 95% confidence intervals for the mean number of customers for each day of the week.

To do this, we can use the function *predict()*, which not only provides us the estimated mean number of customers, but also standard errors for each estimation. This allows us to build a Wald confidence interval. We can also use the *mcprofile()* to build the LR confidence interval.

R Code of Wald CI:

```
> mu <- predict(object = mod, newdata = data.frame(Day = c("Monday","Tuesday
    " ,"Wednesday","Thursday" ,"Friday" )), interval="predict",type = "
    response",se=TRUE)
> lower<-mu$fit - qnorm(0.975)*mu$se
> upper<-mu$fit + qnorm(0.975)*mu$se
> data.frame(Day=data$Day[1:5],estim = mu$fit ,lower= lower,upper=upper)
```

Output:

```
       Day estim lower upper
1 Monday 1.6 0.4912769 2.708723
2 Tuesday 4.2 2.4036633 5.996337
3 Wednesday 4.8 2.8796354 6.720365
4 Thursday 6.0 3.8529672 8.147033
5 Friday 5.0 3.0401482 6.959852
```

R Code of LR CI:

```
K <- matrix ( data = c(1, 0, 0, 0, 0,
                       1, 1, 0, 0, 0,
                       1, 0, 1, 0, 0,
                       1, 0, 0, 1, 0,
                       1, 0, 0, 0, 1),
nrow = 5, ncol = 5,
byrow = TRUE )

library("mcprofile")
linear.combo <- mcprofile( object = mod , CM = K)
ci.log.mu <- confint( object = linear.combo , level = 0.95 , adjust = "none"
    )

mean.LR.ci1 <- data.frame(Day = c("Monday","Tuesday" ,"Wednesday","Thursday"
     ,"Friday" ) , Estimate = exp (ci.log.mu$estimate ), Lower = exp(ci.log.
    mu$confint[ ,1]) , Upper = exp(ci.log.mu$confint[ ,2]))
mean.LR.ci1
```

Output:

```
#LR confidence interval with mcprofile :
        Day Estimate Lower Upper
C1 Monday 1.6 0.7311759 2.978336
C2 Tuesday 4.2 2.6500947 6.260952
C3 Wednesday 4.8 3.1267253 6.984496
C4 Thursday 6.0 4.1010541 8.410386
C5 Friday 5.0 3.2873177 7.223949
```

Which gives us similar results. The observed means are given by:

R Code:

```
means <- aggregate(Count ~ Day, data, mean)
mean
```

Output:

```
       Day Count
1 Monday 1.6
2 Tuesday 4.2
3 Wednesday 4.8
4 Thursday 6.0
5 Friday 5.0
```

We notice that the estimated mean number of customers for each day of the week using the model are the same as the observed means. This was expected because *we are allowing the model to estimate a separate mean for each habitat, and the MLE in this case can be shown to be the respective sample means* (from the textbook. We note that in this case it is not habitats, but days).

d)

The hypotheses for the LRT in part (c) can be written as $H_0 : \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$ Vs H_a : At least one $\beta_r \neq 0$: These hypotheses can be equivalently expressed as $H_0 : \mu_{Monday} = \mu_{Tuesday} = \mu_{Wednesday} = \mu_{Thursday} = \mu_{Friday}$ Vs H_a : At least one pair of means is unequal, where μ_i represents the mean number of customers in line on day i. Discuss why these two ways of writing the hypotheses are equivalent. Write out the proper forms of the Poisson regression model to support your result.

The model of Poisson regression is given by:

$$\log(\mu) = \beta_0 + \beta_2 I(\text{Day} = Tuesday) + \beta_3 I(\text{Day=Wednesday}) + \beta_4 I(\text{Day} = \text{Thursday}) + \beta_5\ I(\text{Day} = \text{Friday})$$

The null hypothesis for testing the significance of the predictor is: $H_0 : \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$. Under this null hypothesis we have that:

$$\log(\mu) = \beta_0 \quad \Leftrightarrow \quad \mu = e^{\beta_0} = \text{constant}.$$

Therefore, the means are constant (i.e. $\mu_{Monday} = \mu_{Tuesday} = \mu_{Wednesday} = \mu_{Thursday} = \mu_{Friday}$)
On the other way, let us suppose that we have $\mu_{Monday} = \cdots = \mu_{Friday}$. We will compare $\mu_{Monday} = \mu_{Tuesday}$:

$$\beta_0 = \beta_0 + \beta_2 \Longleftrightarrow \beta_2 = 0.$$

Doing this comparison with every day, we get that $\beta_2 = \cdots = \beta_5 = 0$. We conclude that both hypotheses are equivalent.

Exercise 3

a)i)

Convert the explanatory variables Smoke and Social to variables Smokef and Socialf using as.factor()

First we should convert the levels of the dataframe on factors, and to do so we use the function factor().

R Code:

```
levels(data$Smoke)
levels(data$Social)
data$Smokef<-factor(data$Smoke,levels=c("1","2" ,"3" ))
data$Socialf<-factor(data$Social,levels=c("1","2" ,"3","4","5"))
levels(data$Smokef)
levels(data$Socialf)
head(data)
```

Output:

```
#Before
NULL
NULL
#After
[1] "1" "2" "3"
[1] "1" "2" "3" "4" "5"
#Head of data
  Social Smoke HT PU Count Smokef Socialf
1 1 1 y y 28 1 1
2 2 1 y y 50 1 2
3 3 1 y y 278 1 3
4 4 1 y y 63 1 4
```

```
5 5 1 y y 20 1 5
6 1 1 y n 82 1 1
```

c)i)

Fit the model and use the residual deviance to test whether the excluded four-variable interaction is needed in the model. Report results and draw conclusions.

Fitting the models : R Code:

```
mod4<-glm(Count ~ Socialf*Smokef*HT*PU, poisson(link=log), data=data)
mod3<-glm(Count ~ (Socialf+Smokef+HT+PU)^3, poisson(link=log), data=data)
mod2<-glm(Count ~ (Socialf+Smokef+HT+PU)^2, poisson(link=log), data=data)
mod0<-glm(Count ~ Socialf+Smokef+HT+PU, poisson(link=log), data=data)
```

If we want to use the residual deviance to test the necessity of including four-variable interaction we have to compare this model with another that lacks this interaction. We will consider the mutual independence model, the homogeneous model and another one, where we take into account interactions up to order 3. We will see that if we compare the 4-variable model with the mutual independence, this interaction seems to be needed. However, we will also see that there are simpler models with a similar performance, and thus it will not be needed.

R Code:

```
anova(mod2,mod4,test="Chisq")
anova(mod3,mod4,test="Chisq")
anova(mod0,mod4,test="Chisq")
```

Output:

```
Analysis of Deviance Table

Model 1: Count ~ (Socialf + Smokef + HT + PU)^2
Model 2: Count ~ Socialf * Smokef * HT * PU
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 30 40.472
2 0 0.000 30 40.472 0.09603 .
---
Analysis of Deviance Table

Model 1: Count ~ (Socialf + Smokef + HT + PU)^3
Model 2: Count ~ Socialf * Smokef * HT * PU
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 8 12.681
2 0 0.000 8 12.681 0.1233
Analysis of Deviance Table

Model 1: Count ~ Socialf + Smokef + HT + PU
Model 2: Count ~ Socialf * Smokef * HT * PU
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 51 1111.5
2 0 0.0 51 1111.5 < 2.2e-16 ***
---
```

model	Res.df	Res.Dev	Dev	LRT df	p-value
mod2	30	40.472	40.472	30	0.09603
mod3	8	12.681	12.681	8	0.1233
mod0	51	1111.5	1111.5	51	<2.2e-16

Conclusion:
We see that the four-variable interaction is not needed. In fact, homogeneous association model (that is, mod2, according to our notation) works well enough.

d)i)

Fit the simpler model that omits all 3-variable interactions and Test whether the simpler model provides an adequate fit relative to the model that includes all 3-variable interactions. State the hypotheses, test statistic, and p-value. Draw conclusions.

R Code:

```
> mod2<-glm(Count ~ (Socialf+Smokef+HT+PU)^2 , poisson(link=log), data=data)
> mod3<-glm(Count ~ (Socialf+Smokef+HT+PU)^3, poisson(link=log), data=data)
> anova(mod2,mod3,test="Chisq")
```

Output:

```
Analysis of Deviance Table

Model 1: Count ~ (Socialf + Smokef + HT + PU)^2
Model 2: Count ~ (Socialf + Smokef + HT + PU)^3
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 30 40.472
2 8 12.681 22 27.792 0.1827
---
```

The hypotheses:
We want to test the significance of the association between the predictors, we test to see if we can omit the third order interaction terms between these two variables.
the LRT statistic is 27.792, and the p-value for the LR test is 0.1827, so The conditional 3-interactions is not significant.

d)ii)

According to these results, are smoking or social class related to the association between proteinurea and hypertension? Explain.

From the analysis of the question d)i) we can assume that it might not be a reasonable to assume that smoking or social class related are associated to proteinurea and hypertension. We can do more by using the Anova function to test the assumption that there are no interactions between the predictors.

R Code:

```
> Anova(mod3,type = 3)
```

Output:

```
Analysis of Deviance Table (Type III tests)

Response: Count
                LR Chisq Df Pr(>Chisq)
Socialf 4707.8 4 < 2.2e-16 ***
Smokef 351.0 2 < 2.2e-16 ***
HT 122.8 1 < 2.2e-16 ***
PU 285.6 1 < 2.2e-16 ***
Socialf:Smokef 351.7 8 < 2.2e-16 ***
Socialf:HT 4.0 4 0.40735
Socialf:PU 12.1 4 0.01655 *
Smokef:HT 0.3 2 0.86890
Smokef:PU 0.4 2 0.82278
HT:PU 29.2 1 6.467e-08 ***
Socialf:Smokef:HT 15.0 8 0.05873 .
Socialf:Smokef:PU 3.7 8 0.88377
Socialf:HT:PU 3.0 4 0.56332
Smokef:HT:PU 6.2 2 0.04569 *
---
```

We can see that the HT:PU is much more significant than "**Smokef:HT:PU**" Or "**Socialf:HT:PU**", and as a consequence we can say that the smoking condition nor the social class are related to the association between HT and PU. We see that it is much more significant because the p-value (6.467e-08) is considerably small, whereas the others are 0.04569 and 0.5632.

d)iii)

Use odds ratios based on the SH term to examine the relationship between smoking and hypertension. Compare all pairs of smoking levels. List the estimated odds ratios and 95% confidence intervals and draw conclusions.

There are three levels in smoking and two different outcomes in hypertension. As a consequence, we will just have 3 different odds ratios based on the SH term.

As we saw in class, with the homogeneous model established, the expression for the odds ratios based on the SH term are:

$$\begin{aligned}\log(\mathrm{OR}_{S,H}) =& \log\left(\frac{\mu_{ijkl}\mu_{i'j'kl}}{\mu_{ij'kl}\mu_{i'jkl}}\right)\\ =& \log(\mu_{ijkl}) + \log(\mu_{i'j'kl}) - \log(\mu_{i'jkl}) - \log(\mu_{ij'kl})\\ =& \beta_{ij}^{S,H} + \beta_{i'j'}^{S,H} - \beta_{i'j}^{S,H} - \beta_{ij'}^{S,H},\end{aligned}$$

Where the term $\beta_{ij}^{S,H}$ is 0 if either i or j is the baseline. We will now see the odds ratios that we have to compute:

1. Smoke(Medium,Low) Hypertension(Yes,No): as the baselines are low smoking level and no hypertension, this odds ratio is just computed the coefficient $\beta_{Medium,Yes}^{SH}$.
2. Smoke(High,Medium) Hypertension(Yes,No): $\beta_{High,Yes}^{SH} - \beta_{Medium,Yes}^{SH}$.
3. Smoke(High,Low) Hypertension(Yes,No): $\beta_{High,Yes}^{S,H}$.

We will use *mcprofile()* and then *confint()* to compute the estimates for the odds ratio and a 95% confidence interval. First of all, we have to build a matrix with the linear combinations of the coefficients that we want to estimate, so we have to localize the terms that we are interested in.

R Code:

```r
coef(mod2)
```

Output:

```
(Intercept) Socialf2 Socialf3 Socialf4 Socialf5
    5.629344520 1.008813726 2.430502286 0.871519906 -0.141084787
        Smokef2 Smokef3 HTy PUy Socialf2:Smokef2
   -1.277179626 -3.108622131 -1.117143917 -2.602795002 0.357412990
Socialf3:Smokef2 Socialf4:Smokef2 Socialf5:Smokef2 Socialf2:Smokef3 Socialf3:Smokef3
    0.949516865 1.240352546 1.587642362 0.121247855 1.004878569
Socialf4:Smokef3 Socialf5:Smokef3 Socialf2:HTy Socialf3:HTy Socialf4:HTy
    1.708307340 1.797129705 0.133329852 0.053850382 -0.033330475
   Socialf5:HTy Socialf2:PUy Socialf3:PUy Socialf4:PUy Socialf5:PUy
   -0.023763469 -0.493053226 -0.221035659 -0.033065136 0.143663650
    Smokef2:HTy Smokef3:HTy Smokef2:PUy Smokef3:PUy HTy:PUy
   -0.432100586 -0.366252728 0.024626780 -0.005426109 1.377966580
```

We are just interested in the terms 26 and 27, which are Smokef2:HTy and Smokef3:HTy.

R Code:

```r
library(mcprofile)
MediumtoLow<-c(numeric(25),1,numeric(4)) #Medium-Low
HightoLow<-c(numeric(26),1,numeric(3)) #High-Low
HightoMedium<-c(numeric(25),-1,1,numeric(3)) #Medium-High
L=rbind(MediumtoLow,MediumtoHigh,HightoLow)
linear.combo<-mcprofile(mod2, CM = L)
exp(confint(object = linear.combo, level = 0.95, adjust = "none"))
```

Output:

```
 mcprofile - Confidence Intervals

level:   0.95
adjustment: single-step

            Estimate lower upper
MediumtoLow 0.649 0.585 0.720
HightoMedium 1.068 0.864 1.313
HightoLow 0.693 0.565 0.846
```

We estimate that

- The odds of being a medium smoker versus being a mild smoker decreases by 35% when having hypertension.
- The odds of of being a strong smoker versus a medium smoker increases by 6.8% when having hypertension.
- The odds of being a strong smoker versus a mild smoker decreases by 31% when having hypertension.

Exercise 4

a)

Modify the function PoissonReg from Example 30 from class to get a function that fits a logistic regression model with a binary response with Fisher Scoring.

In class we saw a user-defined Poisson Regression function, in which we could compute some of the things that we can also get from *glm()*: coefficients, variance-covariance matrix, deviances and degrees of freedom. This is done using Fisher-Scoring, and iterative method based on Newton's method. If $\vec{\beta}$ is a vector containing the coefficients of the model, the Fisher-Scoring method is

$$\vec{\beta}_{n+1} = \vec{\beta}_n + I^{-1}(\vec{\beta}_n)S(\vec{\beta}_n)$$

In this formula, I^{-1} is the inverse of the Fisher information, which can be approximated as the variance-covariance matrix, and S is the score. Changes in the deviance of the model along the different iterations will be used as a stopping criteria.

As we are creating our function for logistic regression adapting the Poisson function already created, we just have to change certain things:

- $\vec{\beta}_0$. We are still taking the initial value from the null model, but as this is logistic regression we will have $\vec{\beta}_{0,0} = \log\left(\frac{\overline{y}}{1-\overline{y}}\right)$ instead of the corresponding estimator for the Poisson case.
- $\hat{\pi}$ will now be computed as $\frac{\exp(X\vec{\beta})}{1+\exp(X\vec{\beta})}$, where X is the design matrix the exponential is component-wise.
- $\text{Cov}(Y) = \pi(1-\pi)$ and not π, which was the case with Poisson.
- The deviance is now computed differently:

$$-2\log(\Lambda) = -2\sum_{i=1}^{n}\left(y_i \log\left(\frac{\hat{\pi}_i}{y_i}\right) + (1-y_i)\log\left(\frac{1-\hat{\pi}_i}{1-y_i}\right)\right)$$

  To avoid computational problems whenever $y = 0$ we will compute it differently, as in class.

We will now present the pseudocode, and then the code written in R.

Algorithm 1: Scoring Fisher algorithm

Input :

$y \longleftarrow$ a numeric vector of the observed counts ;
$x \longleftarrow$ is the design matrix ;
$n \longleftarrow$ is the number of rows ;
$P \longleftarrow$ is the number of coefficients ;
$\epsilon \longleftarrow 10^{-8}$;
$M \longleftarrow$ maximum of iterations ;

Initialization
$\beta_0 \longleftarrow (\bar{y}, 0, \cdots, 0)$;
$\mu_0 \longleftarrow e^{x\beta_0}$;
$Cov(\beta_0) \longleftarrow (x'\text{diag}(\mu_0(1-\mu_0))x)^{-1}$ (using QR-decomposition of $Cov(Y)^{1/2} * X$) ;
$DEV_0 \longleftarrow -2(l - l_{saturated})$;
$S(\beta_0) \longleftarrow x'(y - \mu)$;
$i = 0$;
while $Error_i \geq \epsilon$ *or* $i < M$ **do**
  $i = i + 1$;
  $\beta_i \longleftarrow \beta_{i-1} + \text{Cov}(\beta_{i-1})S(\beta_{i-1})$;
  $\mu_i \longleftarrow e^{x\beta_i}$;
  $Cov(\beta_i) \longleftarrow$
   $(x'\text{diag}(\mu_i(1-\mu_i))x)^{-1}$ (using QR-decomposition of $\text{Cov}(Y)^{1/2} * X$) ;
  $DEV_i \longleftarrow -2(l_i - l_{saturated,i})$;
  $Error_i \longleftarrow |\frac{DEV_i - DEV_{i-1}|}{Dev_i + 0.1}$;
end
return: ;
coefficients $\longleftarrow \beta_i$;
vcov $\longleftarrow Cov(\beta_i)$;
deviance $\longleftarrow DEV_i$;
df$\longleftarrow n - P$;
null.dev$\longleftarrow DEV_0$;
df.null $\longleftarrow$ n-1

User-defined logistic regression for binomial variable

```
LogitReg<-function(y,x,epsilon= 1e-8,maxiteration=100){
  ## y is a numeric vector of the observed counts
  ## x is the design matrix
  ## n is the number of rows
  n<-length(y)
  ## P is the number of coefficients
  P<-ncol(x)

  ## initial beta (beta0=log(ybar), all other betas are 0)
  ## initial model is the null model
  coef<-numeric(P)
  coef[1]<-log(mean(y)/(1-mean(y))) # initialisation
  linPred<-x %*% coef
  pi<-as.vector(exp(linPred)/(1+exp(linPred)))

  ## QR-decomposition of Cov(Y)^{1/2}*X
  QR <- qr(diag(sqrt(pi*(1-pi)), nrow = length(pi)) %*% x)
```

```
  ## inverse of (R'R), which is inverse of Fisher Information
  Cov<- chol2inv(qr.R(QR)) ##
  score<-t(x) %*% matrix((y-pi),ncol=1)


  ## Residual Deviance
  ratio1=y/pi
  log.ratio1<-ifelse(ratio1==0, 0, log(ratio1))
  ratio2=((1-y)/(1-pi))
  log.ratio2<-ifelse(ratio2==0, 0, log(ratio2))
  dev.old<-deviance
  deviance=+2*sum(y*log.ratio1+(1-y)*log.ratio2)
  #deviance<-2*sum(-y*log(pi)+(1-y)*log(1-pi))
  null.dev<-deviance
  df.null<-n-1

  iteration<-0
  test.iteration<-(iteration<maxiteration)
  test.error.large<-TRUE

  while (test.iteration & test.error.large)
  {
    score<-t(x) %*% matrix((y-pi),ncol=1)
    iteration<-iteration+1
    coef<-coef+Cov %*% score

    linPred<-x %*% coef
    pi<-as.vector(exp(linPred)/(1+exp(linPred)))
    ## QR-decomposition of Cov(Y)^{1/2}*X
    QR <- qr(diag(sqrt(pi*(1-pi)), nrow = length(pi)) %*% x)

    ## inverse of (R'R), which is inverse of Fisher Information
    Cov<- chol2inv(qr.R(QR))
    ##
    score<-t(x) %*% matrix((y-pi),ncol=1)

    ratio1=y/pi
    log.ratio1<-ifelse(ratio1==0, 0, log(ratio1))
    ratio2=((1-y)/(1-pi))
    log.ratio2<-ifelse(ratio2==0, 0, log(ratio2))
    dev.old<-deviance
    deviance=+2*sum(y*log.ratio1+(1-y)*log.ratio2)
    #deviance<-2*sum(-y*log(pi)+(1-y)*log(1-pi))
    rel.error<-abs(deviance-dev.old)/(abs(deviance+0.1))
    test.error.large<-(rel.error>=epsilon)

  }


  print(paste("Fisher Scoring Iterations: ",iteration))
  #return(coefficients=as.vector(coef))
  return(list(coefficients=as.vector(coef),vcov=Cov,deviance=deviance,df=n-P,null
      .dev=null.dev,df.null=df.null))

}
```

b)

Use your function on the data from the file FluShot.csv from Example 17 to fit a model to describe the probability of vaccination as a function of age, awareness, and gender (as an additive model). Display the estimation of the coefficients and the estimated variance-covariance matrix.

To do this we first have to extract the response and the explanatory variables. Then we can proceed to run the function that we have built.

R Code:

```
data<-read.csv("FluShot.csv")
y<-data$y #Response variable
x<-data[,2:4] #Explanatory variables
x<-as.matrix(x)
x<-cbind(rep(1,nrow(x)),x)
colnames(x)[1]<-"intercept"
usermod<-LogitReg(y,x)
usermod$coefficients
usermod$vcov
```

Output:

```
> usermod$coefficients
[1] -1.17715922 0.07278802 -0.09898649 0.43397485
> usermod$vcov
           [,1] [,2] [,3] [,4]
[1,] 8.89484573 -0.0735554861 -0.0752476196 0.176322349
[2,] -0.07355549 0.0009229975 0.0002541096 -0.002890602
[3,] -0.07524762 0.0002541096 0.0011208142 -0.002720370
[4,] 0.17632235 -0.0028906018 -0.0027203697 0.272269082
```

The model estimated has the form

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = -1.1772 + 0.0728x_1 - 0.0990x_2 + 0.4340x_3.$$

The variance-covariance matrix of the coefficients is displayed on the output.

c)

Use the glm function to fit the model from b). Does it give the same results as your function?

To do the first part, we will apply the function *glm()* as usual. To check if the results are the same with our function and glm, we will display the difference between our models of the coefficients, the variance-covariance matrix and the rest of the information computed.

R Code:

```
mod<-glm(data=data,formula=y~x1+x2+x3,binomial(link=logit))
usermod$coefficients-mod$coefficients
usermod$vcov-mod$vcov
usermod$deviance-mod$deviance
usermod$df-mod$df.residual
usermod$null.dev-mod$null.deviance
usermod$df.null-mod$df.null
```

Output:

```
> mod<-glm(data=data,formula=y~x1+x2+x3,binomial(link=logit))
> usermod$coefficients-mod$coefficients
  (Intercept) x1 x2 x3
-2.589794e-09 -2.121497e-11 9.030589e-11 -6.895187e-10
> usermod$vcov-mod$vcov
numeric(0)
> usermod$deviance-mod$deviance
[1] 0
> usermod$df-mod$df.residual
[1] 0
> usermod$null.dev-mod$null.deviance
[1] 0
> usermod$df.null-mod$df.null
[1] 0
```

All the differences but the difference of the coefficients are 0. However, the greatest value in this difference has an order of magnitude of -9, which is **very** small (we note, for example, that the tolerance for the stopping criteria is 10^{-8}). We can state that we have the same result.