MAT 5196(4378)

Assignment 3

Soufiane Fadel — Pablo Señas Peón

January 5, 2020

Exercise 1

a)

Use these data to test the hypothesis that these attributes of classification are independent. Use Pearson's chi-square test for independence.

We build the test as follows: the attributes of classification shown are

$$H_0 : \text{are independent} \qquad H_1 : \text{not independent.}$$

As we know, H_0 is equivalent to saying that $\pi_{ij} = \pi_{i+}\pi_{+j}$, where π_{ij} is the probability that a unit is in cell (i, j) $(i \in \{1, 2, 3, 4\}, j \in \{1, 2, 3\})$. We will use R to apply Pearson's chi-square test for independence.

R Code:

```
c.table<-array(data =
    c(359,462,88,37,140,200,39,10,5,17,2,3), dim = c(4,3),
    dimnames = list(Number.of.hours = c("Less than HS", "HS or JC","BSD","
        GD"),
    Respiratory.Symptoms = c("Protestant", "Catholic","Jewish")))
c.table
ind.test<-chisq.test(c.table,correct=FALSE)
ind.test
```

Output:

```
                 Respiratory.Symptoms
Number.of.hours Protestant Catholic Jewish
   Less than HS 359 140 5
   HS or JC 462 200 17
   BSD 88 39 2
   GD 37 10 3

   Warning message:
In chisq.test(c.table, correct = FALSE) :
  Chi-squared approximation may be incorrect

    Pearson's Chi-squared test

data: c.table
```

```
X-squared = 9.9256, df = 6, p-value = 0.1278
```

Using this test for independence we get that the p-value (0.1278) is relatively big, and as a consequence we do not have evidence suggesting that there is an association between the religion of a person and his maximum educational level. However Cochran's Rule does not apply in this case, as the output indicates us, and we do not have any guarantee that the Chi-squared approximation is good.

b)

Use a parametric bootstrap to test the hypothesis that these attributes of classification are independent.

We will proceed as in class. That is, following the next steps:

1. Estimate the distribution under the null (a multinomial) using the initial data.
2. Generate 999 samples from the estimated distribution.
3. Use Pearson's chi-square test for independence in every sample. Let $\{X^2_{*_k}\}_{k=1}^{999}$ be the set of values of the statistic.
4. Compute an estimation of the p-value:

$$\widehat{p-value} = \frac{1+\sum_{k=1}^{999} I\{X^2_{*_k} \geq X^2_0\}}{1+999}$$

R Code:

```
#We set the number of samples that we are going to extract.
B<-999

#Total size of the sample.
n<-sum(c.table)
# Distribution of religion beliefs according to education level
p_i<-rowSums(c.table)/n
# Distribution of education level according to religion
p_j<-colSums(c.table)/n

#Matrix of probabilities of the distribution under H0
p_ij<- p_i %o% p_j

#Initialization of vector of statistics
Xsq.star<-numeric(B)

#Loop for computing the statistics of the new samples
for (i in 1:B){
  aux<-rmultinom(1,size=n,prob=p_ij)
  aux<-array(data=aux,dim=c(4,3))
  Xsq.star[i]<-chisq.test(aux,correct=FALSE)$statistic
}

#We compute the estimated of the pvalue

X0<-ind.test$statistic
pvalue<-(1+sum(Xsq.star>=X0))/(B+1)
pvalue
```

Output:

```
There were 50 or more warnings (use warnings() to see the first 50)

[1] 0.136
```

First of all, we note that the message with the warnings is related to the for loop. The estimated of the p-value that we get is 0.136, which as we know it is not an unbiased estimator but works well enough. It is even bigger that the p-value computed in a), and we still have no evidence of an association.

c)

Do you believe that we can rely on the chi-square approximation from part (a)?

The first thing that we might want to check is how much is the Cochran's Rule violated. As we know, it says that we cannot have too many cells with $\hat{E}_{ij} \leq 1$ and most of them with $\hat{E}_{ij} \geq 5$. We can look at the expected values to see how many of them violate this rule.

R Code:

```
ind.test$expected
```

Output:

```
                Respiratory.Symptoms
Number.of.hours Protestant Catholic Jewish
   Less than HS 350.06167 143.94714 9.9911894
   HS or JC 471.61087 193.92878 13.4603524
   BSD 89.59912 36.84361 2.5572687
   GD 34.72834 14.28047 0.9911894
```

As we can see, there are two cells where the expected value is smaller than what we would want (these cells are (BSD, Jewish) and (GD, Jewish)). In the rest of the cells there are no problems whatsoever. At first it does not seem that we are doing bad relying on this approximation. Besides, both p-values are pretty similar and the difference does not change anything. Finally, we can also compute the percentile of X_0^2 in the set of values $\{X_{*_k}^2\}_{k=1}^{999}$. If this percentile is too large, then we could have evidence suggesting that the Chi-squared approximation is not good.

R Code:

```
percentile<-ecdf(c(Xsq.star))
percentile(X0)
```

Output:

```
[1] 0.8648649
```

X_0^2 is in the 86th percentile. Again, this is not evidence against the Chi-squared approximation, and we can state that we can rely on the Chi-squared approximation.

Exercise 2: Exercise 21 in Section 2.4

(a)

Show how the lethal dose level is derived by solving for x in the logistic regression model.

In this case, the logistic regression model has the form

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x$$

Given a desired killed rate π, if we look at the regression model, the value of x that makes the identity hold is the amount of pesticide/herbicide needed for that desired killed rate. We can solve for x and get that

$$x = x_\pi \overset{\text{def}}{=} \frac{\text{logit}(\pi) - \beta_0}{\beta_1}$$

then the estimated value of x_π is given by :

$$\hat{x}_\pi \overset{\text{def}}{=} \frac{\text{logit}(\pi) - \hat{\beta}_0}{\hat{\beta}_1}$$

(b)

A (1-α)100 % confidence interval for x is the set of all possible values of x such that

$$\frac{|\hat{\beta}_0 + \hat{\beta}_1 x - \text{logit}(\hat{\pi})|}{\sqrt{\hat{\text{Var}}(\hat{\beta}_0) + x^2\hat{\text{Var}}(\hat{\beta}_1) + 2x\hat{\text{Cov}}(\hat{\beta}_0, \hat{\beta}_1)}} \leq Z_{1-\frac{\alpha}{2}}$$

is satisfied. Describe how this confidence interval for x_π is derived. Note that there is generally no closed-form solution for the confidence interval limits, which leads to the use of iterative numerical procedures.

We can say that $\text{logit}(\hat{\pi}) = \hat{\beta}_0 + \hat{\beta}_1 x$ is an estimator for $\beta_0 + \beta_1 x$ since $(\hat{\beta}_0, \hat{\beta}_1)$ are the estimators for (β_0, β_1). On the other hand we show that:

$$\begin{aligned}\text{Var}(\hat{\beta}_0 + \hat{\beta}_0 x) &= \text{Var}(\hat{\beta}_0) + \text{Var}(x\hat{\beta}_1) + 2\text{Cov}(\hat{\beta}_0, x\hat{\beta}_1) \\ &= \text{Var}(\hat{\beta}_0) + x^2\text{Var}(\hat{\beta}_1) + 2x\text{Cov}(\hat{\beta}_0, \hat{\beta}_1)\end{aligned}$$

$$\begin{aligned}Z &= \frac{|\hat{\beta}_0 + \hat{\beta}_1 x - \text{logit}(\hat{\pi})|}{\sqrt{\hat{\text{Var}}(\hat{\beta}_0) + x^2\hat{\text{Var}}(\hat{\beta}_1) + 2x\hat{\text{Cov}}(\hat{\beta}_0, \hat{\beta}_1)}} \\ &\approx \frac{|\hat{\beta}_0 + \hat{\beta}_1 x - \beta_0 - \beta_1 x|}{\sqrt{\text{Var}(\hat{\beta}_0 + \hat{\beta}_0 x)}} \sim \mathcal{N}(0,1) \qquad \text{(due to Central Limit Theorem)}\end{aligned}$$

As the $\hat{x}_\pi$ is derived from solving $\text{logit}(\hat{\pi}) = \hat{\beta}_0 + \hat{\beta}_1 x$, then the values of x that satisfy $|Z| < Z_{1-\alpha/2}$ can describe the confidence interval for x_π.

(c)

The *dose.p()* function of the MASS package and the *deltaMethod()* function of the car package can calculate $\hat{x}_\pi$ and $\widehat{Var}(\hat{x}_\pi)^{\frac{1}{2}}$, where the estimated variance is found using the delta method. Using this statistic and its corresponding variance, show how a $(1-\alpha)100\%$ Wald confidence interval for x_π can be formed.

We have that the general form of a $(1-\alpha)100\%$ Wald confidence interval for θ is

$$\hat{\theta} \pm Z_{1-\frac{\alpha}{2}} s\{\hat{\theta}\},$$

where $\hat{\theta}$ is the MLE for θ and $s\{\hat{\theta}\}$ is its standard deviation. As a consequence, we first compute x_π and $\widehat{Var}(\hat{x}_\pi)^{\frac{1}{2}}$ and our $(1-\alpha)100\%$ confidence interval will be

$$\hat{x}_\pi \pm Z_{1-\frac{\alpha}{2}} \widehat{Var}(\hat{x}_\pi)^{\frac{1}{2}}$$

We can compute these values in two different ways:

- Using *dose.p()* (MASS package): this function requires a glm object and the desired kill rate. It returns $\hat{x}_\pi$ and its standard deviation. It computes the latter using delta method. We can see the latter typing *dose.p* in order to get the code of the function and looking at the line

  $$\text{SE} < - \text{ sqrt}(((\text{pd } \%*\% \text{ vcov(obj)[cf, cf]}) * \text{ pd}) \%*\% \text{ c(1, 1)})$$

  which is the first order approximation used in delta method.
- Using *deltaMethod()* (car package): the syntax of this function is more complicated, as it can be used whenever we want to estimate a variance using delta method. However, it can be simplified for glm. It not only returns $\hat{x}_\pi$ and the standard deviation, but also the lower and upper bound for a 95% Wald confidence interval for x_π. Let mod be the object for our regression, kl the desired kill rate, vcov the variance matrix of the coefficients of the regression, and let these coefficients be "(Intercept)" and "beta0". Our syntax for using *deltaMethod()* will be

  R Code:

```
deltaMethod(mod,"(log((kl)/(1-kl))-(Intercept))/beta0",
vcov=vcov(mod, complete=FALSE),parameterNames=names(coef(mod)))
```

(d)

Derive the form of x_π using probit and complementary log-log models.

- **Probit**
  the probit regression is defined by :

  $$\text{probit}(\pi) = \Phi^{-1}(\pi) = \beta_0 + \beta_1 x_1 + \cdots \beta_p x_p$$

  where Φ^{-1} is inverse CDF of a standard normal distribution which is used as the link function. we proceed as before by solving the above equation for x and get that

  $$x \overset{\text{def}}{=} x_\pi = \frac{\text{probit}(\pi) - \beta_0}{\beta_1}$$

  then the estimated value of x_π is given by :

  $$\hat{x}_\pi = \frac{\text{probit}(\pi) - \hat{\beta}_0}{\hat{\beta}_1}$$

- **Complementary log-log**
  The complementary log-log regression is defined by :

  $$F^{-1}(\pi) = \beta_0 + \beta_1 x_1 + \cdots \beta_p x_p,$$

  where F^{-1} is the inverse CDF of the Gumbel distribution which is used as the link function and it is given by :

  $$F(x) = \exp\{-\exp[(x-\mu)/\sigma]\} \text{ for } -\infty < x < +\infty, \quad -\infty < \mu < +\infty \text{ and } \sigma > 0$$

then

$$\log(-\log(1-\pi)) = \beta_0 + \beta_1 x_1 + \cdots \beta_p x_p$$

In this particular case, $p = 1$ and as before we get

$$x \overset{\text{def}}{=} x_\pi = \frac{\log(-\log(1-\pi)) - \beta_0}{\beta_1}$$

then the estimated value of x_π is given by :

$$\hat{x}_\pi = \frac{\log(-\log(1-\pi)) - \hat{\beta}_0}{\hat{\beta}_1}$$

Exercise 3: Exercise 22 in Section 2.4

(a)

Estimate a logistic regression model using the picloram amount as the explanatory variable and the number of weeds killed as the response variable.

We will procede as usual. The logistic regression model, in this case, is

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x,$$

where π is the proportion of weeds killed and x the amount of picloram used.

R Code:

```
data<-read.csv("picloram.csv")
mod<-glm(kill/total ~ picloram, weights = total,
        family = binomial(link = logit), data = data)

summary(mod)
```

Output:

```
Call:
glm(formula = kill/total ~ picloram, family = binomial(link = logit),
    data = data, weights = total)

Deviance Residuals:
    Min 1Q Median 3Q Max
-2.4197 -1.1207 0.1021 0.3750 1.7191

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.4600 0.4536 -7.628 2.39e-14 ***
picloram 2.6567 0.3237 8.207 2.27e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 273.485 on 11 degrees of freedom
Residual deviance: 17.865 on 10 degrees of freedom
AIC: 40.623
```

```
Number of Fisher Scoring iterations: 6
```

The MLE for β_0 and β_1 are

$$\begin{cases} \hat{\beta}_0 \approx -3.4600 \\ \hat{\beta}_1 \approx 2.6567 \end{cases}$$

(b)

Plot the observed proportion of killed weeds and the estimated model. Describe how well the model fits the data.

R Code:

```
## Plot proportions of weeds killed versus Picloram used
with(data, plot(x = picloram, y = kill/total,
                      xlab = "Level of Picloram",
                      ylab = "Proportion Killed",
                      panel.first = grid(col = "gray", lty = "dotted")))
## Draw estimated logistic response on the plot
curve(expr = predict(object = mod, newdata =
                     data.frame(picloram = x),
                   type = "response"), col = "red", add = TRUE)

#################################################
# Bubble plot
#################################################
## Plot proportions versus x with circles proportional to the square root of the
    number of attempts
with(data, symbols(x = picloram, y = kill/total, circles =(total)^(0.5),inches =
    0.3,
                xlab = "Level of Picloram",
                ylab = "Proportion Killed",
                panel.first = grid(col = "gray", lty = "dotted")))

# Put estimated logistic response on the plot
curve(expr = predict(object = mod, newdata = data.frame(picloram = x), type = "
    response"), col = "red", add = TRUE)
```

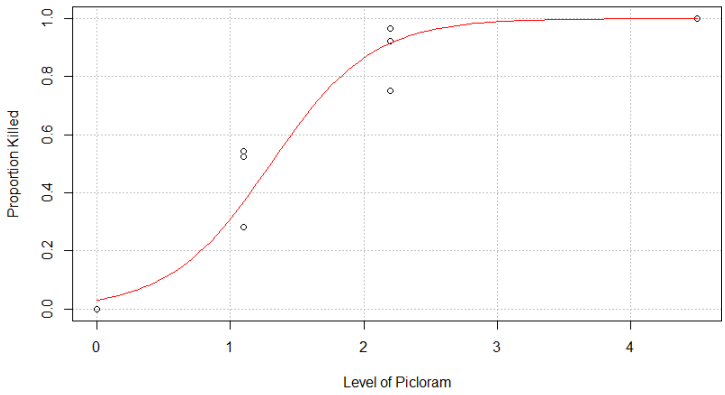


Figure 1: Scatter plot of the sample proportions of weeds killed against the level of Picloram used.

The plot below is a bubble plot that has the plotting point proportional to the total of the weeds at each level of picloram. We can see that the total number of weeds are somewhat similar for each observation (same radius) so there is no huge differences between the plots.

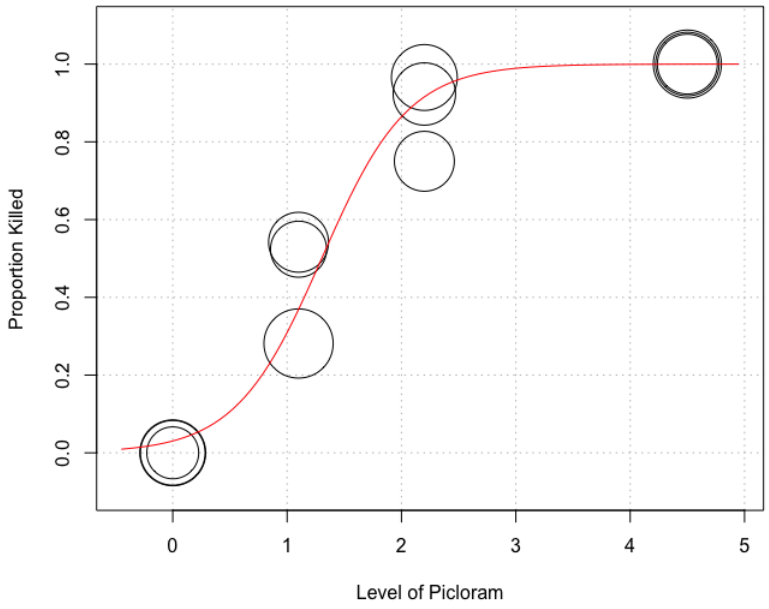


Figure 2: Scatter Bubble plot of the sample proportions of weeds killed against the level of Picloram used.

We can conclude that the fit is good, because most of the sample proportions (bubbles) are close to the estimated proportion from the model. We have to note that the model is a little far from some bubbles, but that is normal because there has been huge differences in the sample proportion using the same level of picloram, and it is not possible for a model like logit to being adaptative to such things. The overall impression, however, is good.

(c)

R Code:

```
# LD90 by numeric calculation
pi0<-0.9
LD90<-(log(pi0/(1-pi0)) - mod$coefficients[1])/mod$coefficients[2]

print(as.numeric(LD90))

segments(x0 = -1, y0 = pi0, x1 = LD90, y1 = pi0, lty = "dashed")
segments(x0 = LD90, y0 = -1, x1 = LD90, y1 = pi0, lty = "dashed")

# LD90 by graphic
segments(x0 = -1, y0 = pi0, x1 = LD90, y1 = pi0, lty = "dashed")
segments(x0 = LD90, y0 = -1, x1 = LD90, y1 = pi0, lty = "dashed")
```

Output:

```
> print(as.numeric(LD90))
[1] 2.12939
```

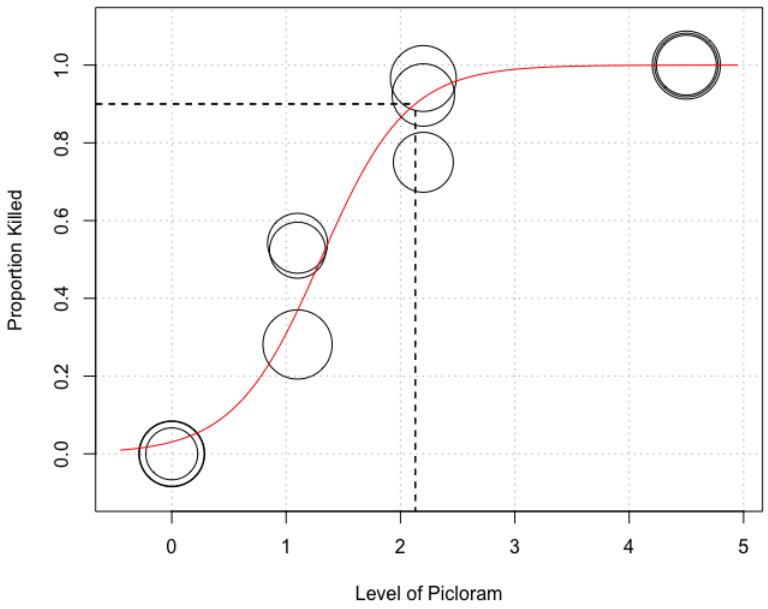


Figure 3: LD90 as added line to the plot (b)

(d) In order to calculate a confidence interval for the dosage that yields 0.9 kill rate we proceed by 3 steps:

- ♣ Creates a the function "root.func" that calculate the estimators of the coefficients of the logit model, the covariance matrix and returns the Z expression seen the exercise 2.
- ♣ Finds the lower bound of the interval by searching between the smallest value of picloram in the data set and the estimated LD90 and for that we can use the function uniroot() that returns a "root" of the equation $Z + 1.96 = 0$
- ♣ Finds the upper bound of the interval by searching between the estimated LD90 and the largest value of picloram in the data set and as before we will use the function the uniroot() that returns a "root" of the equation $Z - 1.96 = 0$.

R Code:

```
#Step 1
root.func<-function(x,mod.fit.obj,pi0,alpha){
  beta.hat<-mod.fit.obj$coefficients
  cov.mat<-vcov(mod.fit.obj)
  var.den<-cov.mat[1,1]+x^2*cov.mat[2,2]+2*x*cov.mat[1,2]
  abs(beta.hat[1]+beta.hat[2]*x-log(pi0/(1-pi0)))/(sqrt(var.den))-qnorm(1-alpha/
      2)
}
#Step 2
lower<-uniroot(f=root.func,
```

```
              interval=c(min(data$picloram),LD90),mod.fit.obj=mod.fit.obj,
              pi0=0.9, alpha=0.95)
#Step 3
upper<-uniroot(f=root.func,
              interval=c(LD90,max(data$picloram)),mod.fit.obj=mod.fit.obj,
              pi0=0.9, alpha=0.95)
```

Output:

```
lower$root
[1] 1.917257
upper$root
[1] 2.446096
```

As a conclusion the 95% confidence interval for LD90 is : $1.9173 < \widehat{LD90} < 2.4461$.

(e)

What amount of picloram should be used in order to have a 0.9 kill rate ?

- For the **point estimate** for LD90 we take 2.12 as seen on the question.
- By using, the upper bound for the LD90 for **the confidence interval** can be interpreted as the safest dosage level to use, also this upper bound can be preferable in an attempt to make sure at least 90% of the weeds are killed because higher the dosage, the higher the estimated probability of success(kills).

(f)

The data for this problem consist of only four different dosage levels of picloram. What assumptions are needed in order for the model to provide a good estimate of x_π ?

In order for the model to provide a good estimate of x_π some assumptions are needed to be satisfied:

1 The log of the odds is indeed linear on the explanatory variables. Our model can fit very well the sample that we have, but if we want to estimate the lethal dose at some point and there is no linear relationship, our estimation will not be good.

2 We also need independence in the trials in the sense that the kill rate just depends on the quantity of picloram used. That way, quantity of picloram and kill rate just depend on the other one.

Exercise 4: Exercise 23 in Section 2.4

Repeat the analysis in Exercise 22 using the probit and complementary log-log models. Do the results change in comparison to using logistic regression?

- Probit model: amongst the alternatives to logit link, this is the most similar to it. It relies on the inverse CDF of a standard normal distribution. It is a symmetric model and the interpretation (this is, for every unit increment in x_i, z score moves β_i units) is very clear. It is commonly used in social sciences. We remember the expression of the model:

$$\text{probit}(\pi) = \Phi^{-1}(\pi) = \beta_0 + \beta_1 x_1$$

We note that probit(π) can be computed with *pnorm()*.

a) **Estimation of the model and plot**

R Code:

```
data<-read.csv("picloram.csv")
mod<-glm(kill/total ~ picloram, weights = total,
        family = binomial(link = probit), data = data)
summary(mod)
```

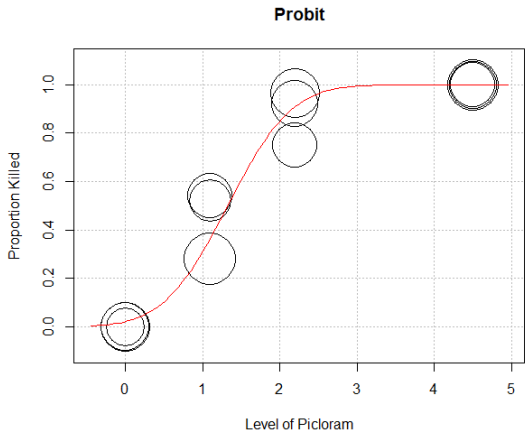
Output:

```
Call:
glm(formula = kill/total ~ picloram, family = binomial(link = probit),
    data = data, weights = total)

Deviance Residuals:
     Min 1Q Median 3Q Max
-2.32011 -1.03668 0.00525 0.45893 1.74565

Coefficients:
           Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.0302 0.2400 -8.460 <2e-16 ***
picloram 1.5347 0.1619 9.479 <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 273.485 on 11 degrees of freedom
Residual deviance: 16.244 on 10 degrees of freedom
AIC: 39.002

Number of Fisher Scoring iterations: 7
```

According to the model, we get

$$\begin{cases} \hat{\beta}_0 \approx -2.0302 \\ \hat{\beta}_1 \approx 1.5347 \end{cases}$$

We proceed to include the bubble plot with the observed proportion of killed weeds and the probit model. The code will not be included, as it is the same as in previous exercise.



b) **Computation of LD90, plot and confidence intervals.**

As we already know the process, we will do parts b, c, d and e together. Solving in the model for x, we get

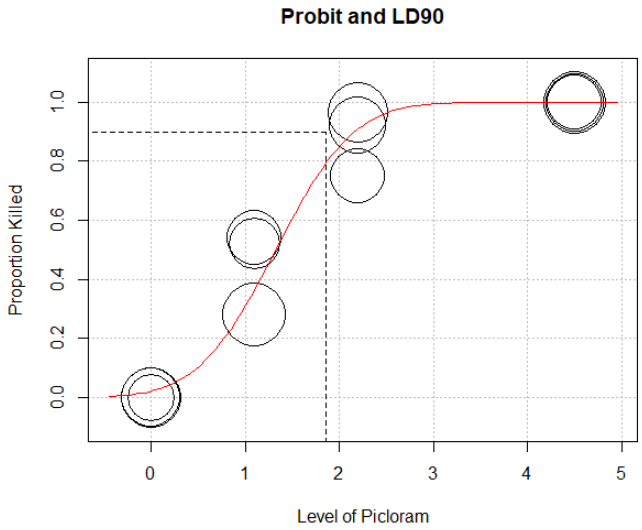
$$x_\pi = \frac{\Phi^{-1}(\pi) - \hat{\beta}_0}{\hat{\beta}_1}$$

R Code of LD90 and plot:

```
pi0<-0.9
LD90<-(pnorm(pi0) - mod$coefficients[1])/mod$coefficients[2]
print(as.numeric(LD90))
segments(x0 = -1, y0 = pi0, x1 = LD90, y1 = pi0, lty = "dashed")
segments(x0 = LD90, y0 = -1, x1 = LD90, y1 = pi0, lty = "dashed")
```

Output:

```
print(as.numeric(LD90))
[1] 1.854443
```



R Code of confidence interval

```
#Step 1
root.func<-function(x,mod.fit.obj,pi0,alpha){
  beta.hat<-mod.fit.obj$coefficients
  cov.mat<-vcov(mod.fit.obj)
  var.den<-cov.mat[1,1]+x^2*cov.mat[2,2]+2*x*cov.mat[1,2]
  abs(beta.hat[1]+beta.hat[2]*x-pnorm(pi0))/(sqrt(var.den))-qnorm(1-alpha/2)
}
#Step 2
lower<-uniroot(f=root.func,
               interval=c(min(data$picloram),LD90),mod.fit.obj=mod.fit.obj,
               pi0=0.9, alpha=0.05)
lower$root
#Step 3
upper<-uniroot(f=root.func,
```

```
              interval=c(LD90,max(data$picloram)),mod.fit.obj=mod.fit.obj,
              pi0=0.9, alpha=0.95)
upper$root
```

Output:

```
lower$root
[1] 1.692917
upper$root
[1] 2.064161
```

We have that $\widehat{\text{LD90}} = 1.8544$ and a 95% confidence interval for LD90 is $[1.6929, 2.0642]$. The recommended amount to use would be, as in the last exercise, the upper bound of the confidence interval, in order to have more conviction that (at least) 90% of the weeds will die.

- Complementary log-log (cloglog): the process is very similar to using logistic regression. The only thing that changes is the link that we use. Unfortunately, coefficients of complementary log-log model cannot be interpreted in an easy way, but it offers some advantages to logistic model in this case: it is an asymmetrical distribution and goes quickly to 1. The latter property can make the complementary log-log model useful in this case, because it is intuitive to see that the kill rate will increase much as we increase our dosage. We remember that the complementary log-log model is

$$\text{cloglog}(\pi) = \log(-\log(1-\pi)) = \beta_0 + \beta_1 x \qquad \pi = 1 - \exp(-\exp(\beta_0 + \beta_1 x))$$

a) **Estimation of the model and plot**

R Code:

```
data<-read.csv("picloram.csv")
mod<-glm(kill/total ~ picloram, weights = total,
        family = binomial(link = cloglog), data = data)
summary(mod)
```

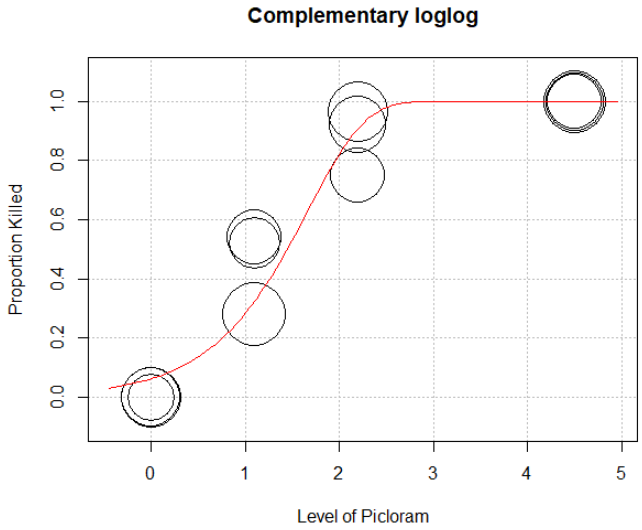
Output:

```
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(mod)

Call:
glm(formula = kill/total ~ picloram, family = binomial(link = cloglog),
    data = data, weights = total)

Deviance Residuals:
    Min 1Q Median 3Q Max
-2.2861 -1.6160 0.0000 0.4877 2.1838

Coefficients:
           Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.7435 0.3145 -8.722 <2e-16 ***
picloram 1.6449 0.1721 9.555 <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 273.49 on 11 degrees of freedom
Residual deviance: 24.95 on 10 degrees of freedom
AIC: 47.708

Number of Fisher Scoring iterations: 7
```

According to the model established above, we get

$$\begin{cases} \hat{\beta}_0 \approx -2.7435 \\ \hat{\beta}_1 \approx 1.6449 \end{cases}$$



b) **Computation of LD90, plot and confidence intervals.**

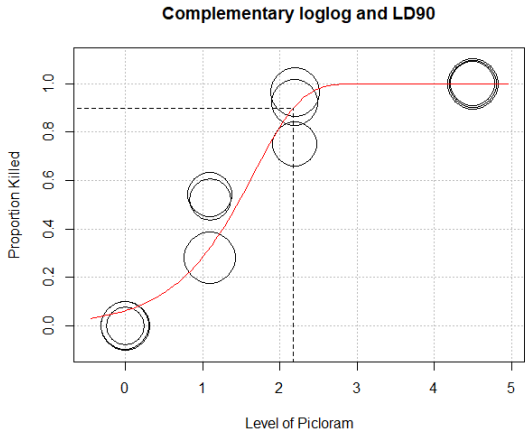
$$x_\pi = \frac{\log(-\log(1-\pi)) - \hat{\beta}_0}{\hat{\beta}_1}$$

R Code of LD90 and plot:

```
pi0<-0.9
LD90<-(log(-log(1-pi0)) - mod$coefficients[1])/mod$coefficients[2]
print(as.numeric(LD90))
segments(x0 = -1, y0 = pi0, x1 = LD90, y1 = pi0, lty = "dashed")
segments(x0 = LD90, y0 = -1, x1 = LD90, y1 = pi0, lty = "dashed")
# LD90 by graphic
segments(x0 = -1, y0 = pi0, x1 = LD90, y1 = pi0, lty = "dashed")
segments(x0 = LD90, y0 = -1, x1 = LD90, y1 = pi0, lty = "dashed")
```

Output:

```
print(as.numeric(LD90))
[1] 2.17497
```



R Code of confidence interval

```
#Step 1
root.func<-function(x,mod.fit.obj,pi0,alpha){
  beta.hat<-mod.fit.obj$coefficients
  cov.mat<-vcov(mod.fit.obj)
  var.den<-cov.mat[1,1]+x^2*cov.mat[2,2]+2*x*cov.mat[1,2]
  abs(beta.hat[1]+beta.hat[2]*x-log(-log(1-pi0)))/(sqrt(var.den))-qnorm(1-
      alpha/2)
}
#Step 2
lower<-uniroot(f=root.func,
               interval=c(min(data$picloram),LD90),mod.fit.obj=mod.fit.obj,
               pi0=0.9, alpha=0.05)
lower$root
#Step 3
upper<-uniroot(f=root.func,
               interval=c(LD90,max(data$picloram)),mod.fit.obj=mod.fit.obj,
               pi0=0.9, alpha=0.05)
upper$root
```

Output:

```
lower$root
[1] 2.022088
upper$root
[1] 2.369648
```

Consequently, $\widehat{\text{LD}90} = 2.1750$ and a 95% confidence interval for LD90 is $[2.0221, 2.3696]$.

We will now compare the results and see if there is any difference with logistic regression.

Comparison between logitic, probit and cloglog:

We can observe from the plots showing the observed proportion of killed weeds and the estimated model that the results are pretty similar in all three cases. This is not a surprise, as we had already seen in class, although the coefficients can differ much the final result will not be very different. However, although pretty similar in the overall, the results may very considerably when computing $x(\pi)$ or $\pi(x)$ (both things are equivalent in this case, where there is just one explanatory variable), as we have seen with $\widehat{\text{LD}90}$ and its confidence interval.

Table 1: Comparison of confidence intervals for LD90 using three links.

	Lower bound	$\widehat{\text{LD}}90$	Upper bound	Length of interval
Logit	1.9173	2.1293	2.4461	0.5288
Probit	1.6929	1.8544	2.0642	0.3712
Cloglog	2.0221	2.1750	2.3696	0.3476

The estimators extracted from logit and cloglog are similar, but the difference with probit estimator is important. However, we can see that although logit and cloglog do not differ in the estimator, they differ much in the confidence interval, as the length of cloglog's CI is considerably smaller. As a consequence, if we had to choose, it would be preferable the latter. However, we cannot do much comparison.

We can note the following: as we know, cloglog models go faster to 1. Taking into account this, we should expect the cloglog estimator of $\widehat{\text{LD}}90$ to be smaller than the logit and probit, which are symmetric models, but this has not been the case. One reason explaining this may be due to the warning returned when fitting the data (*glm.fit: fitted probabilities numerically 0 or 1 occurred*). However this warning does not surprise us, because there are cases when the kill rate is 100%.

Exercise 5: Exercise 12 in Section 3.6

(a)

The explanatory variables need to be re-formatted before proceeding further. First, divide each explanatory variable by its serving size to account for the different serving sizes among the cereals. Second, re-scale each variable to be within 0 and 1.

We will use the code given.

R Code:

```
cereal<-read.csv("cereal_dillons.csv")
head(cereal)

stand01 <- function( x ){ ( x - min( x ) ) /( max( x ) - min( x ) ) }
cereal2 <- data.frame( Shelf = cereal$Shelf , sugar =
                       stand01 ( x = cereal$sugar_g/cereal$size_g ) , fat =
                       stand01 ( x = cereal$fat_g/cereal$size_g ) , sodium =
                       stand01 ( x = cereal$sodium_mg/cereal$size_g ) )
head(cereal2)
```

Output:

```
 ID Shelf Cereal size_g sugar_g fat_g sodium_mg
1  1  1 Kellogs Razzle Dazzle Rice Crispies 28 10 0 170
2  2  1 Post Toasties Corn Flakes 28 2 0 270
3  3  1 Kelloggs Corn Flakes 28 2 0 300
4  4  1 Food Club Toasted Oats 32 2 2 280
5  5  1 Frosted Cheerios 30 13 1 210
6  6  1 Food Club Frosted Flakes 31 11 0 180

#After re-formatting our data:
head(cereal2)
  Shelf sugar fat sodium
1  1 0.6428571 0.000 0.5666667
2  1 0.1285714 0.000 0.9000000
3  1 0.1285714 0.000 1.0000000
```

```
4 1 0.1125000 0.675 0.8166667
5 1 0.7800000 0.360 0.6533333
6 1 0.6387097 0.000 0.5419355
```

(b)

Construct side-by-side box plots with dot plots overlaid for each of the explanatory variables. Also, construct a parallel coordinates plot for the explanatory variables and the shelf number. Discuss if possible content differences exist among the shelves.

We will first start doing the box plots for each one of the three explanatory variables. For this, we will use the code given.

R Code:

```
#Box plot with dot plots for sugar
par(mfrow=c(1,3))
boxplot( formula = sugar~Shelf , data = cereal2 , ylab =
           " Sugar " , xlab = " Shelf " , pars = list ( outpch = NA ) )
    stripchart(x = cereal2$sugar ~ cereal2$Shelf, lwd = 2, col = "red",
          method = "jitter", vertical = TRUE, pch = 1, add = TRUE)
boxplot( formula = fat~Shelf , data = cereal2 , ylab =
          " Fat " , xlab = " Shelf " , pars = list ( outpch = NA ) )
stripchart(x = cereal2$fat ~ cereal2$Shelf, lwd = 2, col = "red",
          method = "jitter", vertical = TRUE, pch = 1, add = TRUE)
boxplot( formula = sodium~Shelf , data = cereal2 , ylab =
          " Sodium " , xlab = " Shelf " , pars = list ( outpch = NA ) )
stripchart(x = cereal2$sodium ~ cereal2$Shelf, lwd = 2, col = "red",
          method = "jitter", vertical = TRUE, pch = 1, add = TRUE)
```

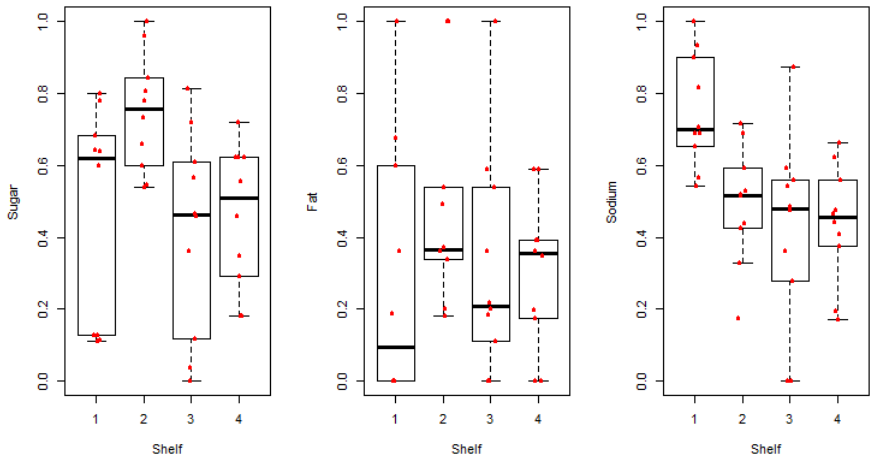


Figure 4: Box plots for each of the explanatory variables.

Now we proceed to draw the parallel coordinates plot for the explanatory variables and the shelf number. We are going to base our code on the code provided by the author of the textbook.

R Code:

```
library(package = MASS)
#We reorder the variables as we want to plot them
cereal22<-data.frame(cereals = 1:nrow(cereal2), cereal2[,c(2,3,4,1)])


# Colors by condition:
cereal22.colors<-ifelse(test = cereal22$Shelf=="1", yes = "black",
                  no = ifelse(test = cereal22$Shelf=="2", yes = "red",
                  no = ifelse(test = cereal22$Shelf=="3", yes = "green",
                  no = "blue4")))
# Line type by condition:
cereal22.lty<-ifelse(test = cereal22$Shelf=="1", yes = "solid",
           no = ifelse(test = cereal22$Shelf=="2", yes = "solid",
           no = ifelse(test = cereal22$Shelf=="3", yes = "solid",
           no = "solid")))

#We draw the parallel coordinates plot and its legend.
parcoord(x = cereal22, col = cereal22.colors, lty = cereal22.lty)
legend(x=1.33,y=1, legend = c("1", "2", "3","4"), lty = c("solid", "solid", "
    solid", "solid"),
      col=c("black", "red", "green","blue4"), cex=0.8, bty="n")
```

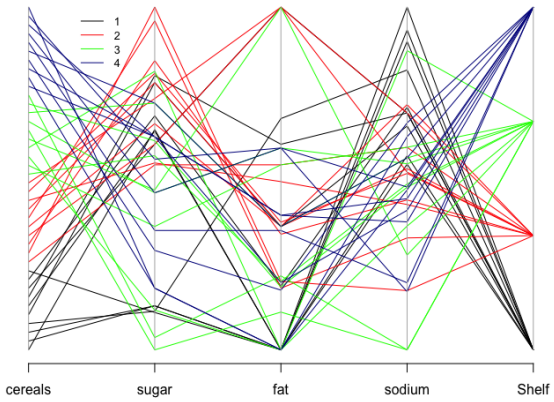


Figure 5: Parallel coordinates plot.

With the help of the box plots and the parallel coordinates plot we will discuss if there is any difference on the explanatory variables of each shelf.

Sugar: We can observe from the box plot that there are differences in sugar content among the shelves. Specifically, we see that the box plot of the second shelf is considerably higher than the rest of them, and it is easily possible to confirm this from the parallel coordinates plot. Apart from that comment, no differences can be remarked.

Fat: We cannot say much about differences in sodium content. The blox plot tells us that cereals from shelf 2 have more moderate contents than the rest, but it is not possible to ensure anything else from either the box plot or the parallel coordinates plot.

Sodium: Observing the box plots it is very clear that cereals from shelf 1 tend to have more sodium than the rest, and this is again what we can infer from the parallel coordinates plot. No other significative difference exists.

(c)

The response has values of $1, 2, 3$ and 4. Under what setting would it be desirable to take into account ordinality? Do you think this occurs here?

It would be desirable to take into account ordinality when there is, in fact, a natural ordering between the different categories of the response variable. For instance, in the example that we saw in class, the response variable was not having any respiratory symptoms, having mild respiratory symptoms or severe. There is an order in this example. However, here we cannot take into account ordinality because $1, 2, 3$ and 4 are just names for the shelves: the shelves cannot be arranged so that

$$\text{Shelf } 1 \leq \text{ Shelf } 2 \leq \text{Shelf } 3 \leq \text{ Shelf } 4.$$

(d)

Estimate a multinomial regression model with linear forms of the sugar, fat, and sodium variables. Perform LRTs to examine the importance of each explanatory variable.

For first part we will use function *multinom* from *nnet*. The baseline shelf will be Shelf 1, as default by R. We remember that the model is

$$\log\left(\frac{\pi_j}{\pi_1}\right) = \beta_{j0} + \beta_{j1}x_1 + \cdots, \beta_{jp}x_p \qquad j = 2, 3, 4$$

R Code:

```
library(nnet)
mod <- multinom(Shelf ~ sugar + fat + sodium, data = cereal2)
summary(mod)
```

Output:

```
Call:
multinom(formula = Shelf ~ sugar + fat + sodium, data = cereal2)

Coefficients:
  (Intercept) sugar fat sodium
2 6.900708 2.693071 4.0647092 -17.49373
3 21.680680 -12.216442 -0.5571273 -24.97850
4 21.288343 -11.393710 -0.8701180 -24.67385

Std. Errors:
  (Intercept) sugar fat sodium
2 6.487408 5.051689 2.307250 7.097098
3 7.450885 4.887954 2.414963 8.080261
4 7.435125 4.871338 2.405710 8.062295
```

As we can see from the output, we have that

$$\begin{cases} \hat{\beta}_{20} \approx 6.9001 \\ \hat{\beta}_{21} \approx 2.6931 \\ \hat{\beta}_{22} \approx 4.0647 \\ \hat{\beta}_{23} \approx -17.4937 \end{cases} \quad \begin{cases} \hat{\beta}_{30} \approx 21.6807 \\ \hat{\beta}_{31} \approx -12.2164 \\ \hat{\beta}_{32} \approx -0.5571 \\ \hat{\beta}_{33} \approx -24.9785 \end{cases} \quad \begin{cases} \hat{\beta}_{40} \approx 21.2883 \\ \hat{\beta}_{41} \approx -11.3937 \\ \hat{\beta}_{42} \approx -0.8701 \\ \hat{\beta}_{43} \approx -24.6739 \end{cases}$$

We will now proceed to do the LRT to test for $H_{0_j} : \text{logit}(\frac{\pi_j}{\pi_1}) = \beta_{j0}$. This is equivalent for testing an association between the explanatory and the response variables. For that we will use the *Anova* function.

R Code:

```
library(package="car")
Anova (mod)
```

Output:

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

Response: Shelf
       LR Chisq Df Pr(>Chisq)
sugar 22.7648 3 4.521e-05 ***
fat 5.2836 3 0.1522
sodium 26.6197 3 7.073e-06 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observing the $p-$values, we have enough evidence to say that there is an association between the sugar content and the shelves ($p-$value$= 4e-05$), as well as an association between the sodium and the shelves ($p-$value$= 7e-06$). However, we do not have evidence suggesting the same for fat content ($p-$value$= 0.15$). We note how all of these things support our previous work (plots and discussion) in b).

(e)

Show that there are no significant interactions among the explanatory variables (including an interaction among all three variables).

We will use again the function *multinom*, but in the formula we will now consider the interactions taking pairs of explanatory variables and the three of them together. This can be done with *sugar*fat*sodium*. As we will see, we are going to encounter some problems with convergence.

R Code:

```
mod.inter <- multinom(Shelf ~sugar*fat*sodium,data = cereal2)
```

Output:

```
# weights: 36 (24 variable)
initial value 55.451774
iter 10 value 36.170336
iter 20 value 31.166546
iter 30 value 29.963705
iter 40 value 28.414027
iter 50 value 27.891712
iter 60 value 27.763967
iter 70 value 27.622579
iter 80 value 27.438263
iter 90 value 27.015534
iter 100 value 26.772481
final value 26.772481
stopped after 100 iterations
```

One way to solve this is to increase the maximum number of iterations, which is fixed to 100 by default. However, it is possible that the program returns convergence notice even though this is not

true. To check this we have to variate the relative convergence tolerance (reltol) and see it the value of convergence of multinom is very different. We have done different trials and saw that setting a small tolerance the value of the convergence does not change much, so the convergence is good enough for our purposes. As we will see, the conclusions that we get are very strong so at the end there will be no doubt.

R Code:

```
mod.inter <- multinom(Shelf ~sugar*fat*sodium,data = cereal2,maxit=100000,reltol
    = 1.0e-15)

Anova (mod.inter)
```

Output:

```
#First information of convergence are not shown.
iter32040 value 25.504821
final value 25.504821
converged
> Anova (mod.inter)
Analysis of Deviance Table (Type II tests)

Response: Shelf
               LR Chisq Df Pr(>Chisq)
sugar 19.2525 3 0.0002424 ***
fat 6.1167 3 0.1060686
sodium 30.8407 3 9.183e-07 ***
sugar:fat 3.2309 3 0.3573733
sugar:sodium 3.0185 3 0.3887844
fat:sodium 3.1586 3 0.3678151
sugar:fat:sodium 5.1237 3 0.1629599
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The $p-$values for the interactions, as we can see, are 0.16 for the interaction between the three variables and about 0.35 for the rest. This is enough evidence to ensure that the interactions are not significant.

(f)

Kellogg's Apple Jacks is a cereal marketed for children. For a serving size of 28 grams, its sugar content is 12 grams, fat content is 0.5 grams, and sodium content is 130 milligrams. Estimate the shelf probabilities for Apple Jacks.

We will use the function *predict* with *type="class"*, as we have seen in class. However, we first have to normalize the data, as we did in section a.

R Code:

```
sugarnor<- function(x){
  ( x - min(cereal$sugar_g/cereal$size_g,x)) /
    ( max(cereal$sugar_g/cereal$size_g,x)- min( cereal$sugar_g/cereal$size_g,x) )
}
fatnor<- function(x){
  ( x - min(cereal$fat_g/cereal$size_g,x)) /
    ( max(cereal$fat_g/cereal$size_g,x)- min( cereal$fat_g/cereal$size_g,x) )
}
```

```
sodnor<- function(x){
  ( x - min(cereal$sodium_mg/cereal$size_g,x)) /
    ( max(cereal$sodium_mg/cereal$size_g,x)- min(cereal$sodium_mg/cereal$size_g,x
        ) )
}

predict.data <- data.frame(sugar = sugarnor ( x = 12/28 ) ,
                           fat = fatnor ( x = 0.5/28 ) ,
                           sodium = sodnor ( x = 130/28 ))
predict(mod,newdata=predict.data,type="probs")
```

Output:

```
        1 2 3 4
0.05326849 0.47194264 0.20042742 0.27436145
```

According to the predictions, the probability of Kelloggs Apple Jacks of being in shelves $1, 2, 3$ or 4 are, respectively, $0.05, 0.47, 0.20, 0.27$, and therefore the prediction is that Kellogg's Apple Jack will be at shelf 2.

(g)

Construct a plot similar to Figure 3.3 where the estimated probability for a shelf is on the y-axis and the sugar content is on the x-axis. Use the mean overall fat and sodium content as the corresponding variable values in the model. Interpret the plot with respect to sugar content.

Considering the multinomial regression model, we have that

$$\pi_j = \frac{\exp(\beta_{j0} + \beta_{j1}x_1 + \beta_{j2}x_2 + \beta_{j3}x_3)}{1 + \sum_{k=2}^{J} \exp(\beta_{k0} + \beta_{k1}x_1 + \beta_{k2}x_2 + \beta_{k3}x_3)} \qquad j = 2, 3, 4.$$

For the baseline, it is

$$\pi_1 = \frac{1}{1 + \sum_{k=2}^{J} \exp(\beta_{k0} + \beta_{k1}x_1 + \beta_{k2}x_2 + \beta_{k3}x_3)}$$

Following the notation from the previous sections, x_1 will denote the sugar content and x_2 and x_3 the fat and the sodium content, respectively. For the last two variables we are going to consider its mean, as indicated in the exercise.

R Code:

```
#We extract the mean of the fat and sodium content.
fatm<-mean(cereal2$fat)
sodiumm<-mean(cereal2$sodium)

#beta.hats of the model
beta.hat<-coefficients(mod)

lwd.mult<-2

# Shelf 1 : (the base line)
curve(expr = 1/
       (1 + exp(beta.hat[1,1] + beta.hat[1,2]*x+beta.hat[1,3]*fatm+beta.hat[1,4]
           *sodiumm) +
          exp(beta.hat[2,1] + beta.hat[2,2]*x+beta.hat[2,3]*fatm+beta.hat[2,4]*
              sodiumm) +
          exp(beta.hat[3,1] + beta.hat[3,2]*x+beta.hat[3,3]*fatm+beta.hat[3,4]*
```

```
              sodiumm)),
      col = "black", lty = "solid", lwd = lwd.mult, n = 1000,
      xlim = c(0, 1), ylim=c(0,1), xlab = "Content of sugar", ylab = "Estimated
          probability" )
#Function to plot the rest of the shelves.
#p argument for line type
plot<-function(i,p){
  curve(expr = exp(beta.hat[i,1] + beta.hat[i,2]*x+beta.hat[i,3]*fatm+beta.hat[i
      ,4]*sodiumm)/
          (1 + exp(beta.hat[1,1] + beta.hat[1,2]*x+beta.hat[1,3]*fatm+beta.hat
              [1,4]*sodiumm) +
             exp(beta.hat[2,1] + beta.hat[2,2]*x+beta.hat[2,3]*fatm+beta.hat[2,4]*
                 sodiumm) +
             exp(beta.hat[3,1] + beta.hat[3,2]*x+beta.hat[3,3]*fatm+beta.hat[3,4]*
                 sodiumm)),
        col = "black", lty = p, lwd = lwd.mult, n = 1000, add = TRUE,
        xlim = c(0, 1),ylim=c(0,1), xlab = "Content of sugar", ylab = "Estimated
            probability")
}

#We note that the matrix of the coefficients just has
#three rows, for shelves 2, 3 and 4.
plot(1,"dotdash")
plot(2,"longdash")
plot(3,"dotted")
legend("top", legend = c("Shelf 1", "Shelf 2", "Shelf 3","Shelf 4"), lty = c("
    solid", "dotdash", "longdash", "dotted"),
       col=c("black", "black", "black","black"), cex=0.8, bty="n")
```

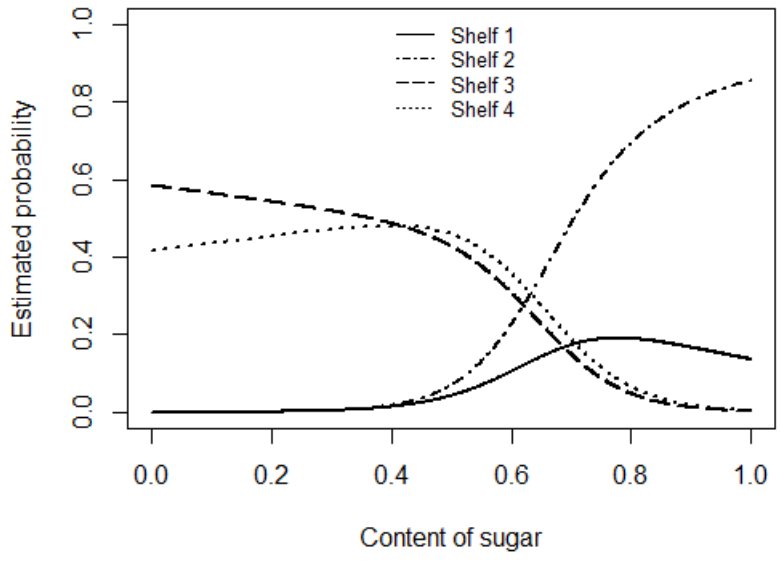


Figure 6: Estimated probability of shelves over content of sugar.

The plot that we obtained is very useful to get interpretations, as it allows us to do something similar

to predictive classification, in which we have values of the explanatory variables (in this case, sugar) and get the most likely response (number of shelf). We remember that fat and sodium content are fixed at its mean. We can get from the plot that

- For cereals with small values of sugar content, the most likely shelf is number 3, and it is close to number 4. On the other hand, it is highly unlikely that these cereals belong to shelf 1 or 2. As the sugar increases, the difference between shelf 3 and shelf 4 is smaller and, in fact, cereals with medium values of sugar content are likely to belong to shelf 4.
- The probability of belonging to Shelf 2 constantly increases, but at medium values it makes a big jump and it soon starts to dominate clearly. As a consequence cereals large values of sugar content are prone to be at Shelf 2. The interpretation for this plot supports what we had seen before with the box plots and the parallel coordinates plot.

(h)

Estimate odds ratios and calculate corresponding confidence intervals for each explanatory variables. Relate your interpretations back to the plots constructed for this exercise.

The following code is based on code written by the authors and included in the file *Wheat.R*.

R Code:

```
################################################
# Odds ratios and Wald CI
################################################
# Information about each variable to help with choosing c

sd.cereal<-apply(X = cereal2[,-1], MARGIN = 2, FUN = sd)
c.value<-c(sd.cereal) # class = 1 is first value
conf.beta<-confint(object = mod, level = 0.95)
orci<-function(i){
  beta.hat.i<-coefficients(mod)[i,2:4]
  ci.OR2<-exp(c.value*conf.beta[2:4,1:2,i])
  round(data.frame("low" = ci.OR2[,1], "or"=exp(c.value*beta.hat.i),
                   "up"= ci.OR2[,2],"ro"=exp(-c.value*beta.hat.i)), 4)
#ro allows us to compute the inverse odds ratio

#We now apply the function to the three response variables.
orci(1)
orci(2)
orci(3)
c.value
}
```

Output:

```
> orci(1) #Odds of shelf 2 compared to 1
         low or up ro
sugar 0.1436 2.0647 29.6795 0.4843
fat 0.8722 3.3719 13.0360 0.2966
sodium 0.0007 0.0179 0.4388 55.7393
> orci(2) #Odds of shelf 3 compared to 1
         low or up ro
sugar 0.0028 0.0373 0.4918 26.8096
fat 0.2056 0.8465 3.4861 1.1813
sodium 0.0001 0.0032 0.1223 311.3613
```

```
> orci(3) #Odds of shelf 4compared to 1
         low or up ro
sugar 0.0036 0.0465 0.6084 21.4833
fat 0.1882 0.7709 3.1574 1.2972
sodium 0.0001 0.0034 0.1301 290.3058
c.value
sugar fat sodium
0.2692078 0.2990292 0.2298359
```

We are comparing the odds of a certain category to the baseline, which is shelf 1. However the election is arbitrary and there is no reason why we should only compute these odds. We now proceed to compute the odds of one shelf compared to a shelf different than 1. In this case we will not compute confidence intervals.

R Code:

```
#ornb returns the odds of i compared to j and the odds of j compared to i
ornb<-function(i,j){
  aux<-orci(i)/orci(j)
  sugar<-aux[1,2]
  fat<-aux[2,2]
  sodium<-aux[3,2]
  data.frame(sugar=c(sugar,1/sugar),fat=c(fat,1/fat),sodium=c(sodium,1/sodium),
      row.names=c("i/j","j/i"))
}
ornb(2,1)
ornb(3,2)
ornb(3,1)
```

Output:

```
> ornb(2,1) #Odds of Shelf 3 compared to Shelf 2 and vice versa
         sugar fat sodium
i/j 0.01806558 0.2510454 0.1787709
j/i 55.35388740 3.9833432 5.5937500
> ornb(3,2) #Odds of Shelf 4 compared to Shelf 3 and vice versa
        sugar fat sodium
i/j 1.2466488 0.9106911 1.0625000
j/i 0.8021505 1.0980672 0.9411765
> ornb(3,1) #Odds of Shelf 4 compared to Shelf 2 and vice versa
         sugar fat sodium
i/j 0.02252143 0.2286248 0.1899441
j/i 44.40215054 4.3739785 5.2647059
```

Having computed all the estimators of the odds ratio that can be of interest for us, we remember its interpretation. First, we note that we have taking c as the standard error of each explanatory variable, following advice from the textbook. We have usually taken $c = 1$, but our data is normalized in a way that the maximum value is 1, so it is not the best option. Having said that, imagine that OR is the sugar odds ratio comparing shelf i to shelf j. The interpretation is the following:

The estimated odds of the shelf i versus the shelf j change by $\widehat{\text{OR}}$ times for a $c-$increase in the sugar content.

This is equivalent to saying that *We estimate that the odds of shelf i versus shelf j increase by* $(100\widehat{\text{OR}} - 100)\%$ *for each increase in c units of the sugar content.*

We will not do a full interpretation of all the coefficients, mainly because there are too many of them. We can try and see if the conclusions that we had from the plots hold here.

– One conclusion was that sugar content in shelf 2 was greater than in the rest of the shelves. Let us look at the odds ratio for the sugar content where shelf 2 is involved.

Sugar	Shelf 2
Shelf 1	2.0647
Shelf 3	55.3539
Shelf 4	44.4022

Table 2: Estimated odds ratio of shelf 2 compared to the rest in c-increment of sugar content.

According to the last table and the interpretation, it is clear that cereals from Shelf 2 are prone to have more sugar content than cereals from the rest of the shelves. We have to note that, although the confidence interval comparing Shelf 2 and Shelf 1 ($[0.1436, 29.6795]$) contains 1 (we do not know about confidence intervals comparing Shelf 2 to Shelf 3 and to Shelf 4, but it seems like 1 is not in that interval) the trend is undeniable.

– Another conclusion was that cereals from Shelf 1 had greater sodium content than cereals from the rest of the shelves.

Sodium	Shelf 1
Shelf 2	55.7393
Shelf 3	311.3613
Shelf 4	290.3058

Table 3: Estimated odds ratio of shelf 1 compared to the rest in c-increment of sodium content.

The results are very clear. Besides, if we look at the confidence intervals comparing the shelves to the first (we note that the numbers from the table are the inverse; the CIs are $[0.0007, 0.4388]$, $[0.0001, 0.1223]$ and $[0.001, 0.1301]$) we see that 1 is far from being inside.

– There is no much comparison that we can do without analyzing each estimation and confidence interval one by one. This does not seem to have a point if we do not have first some clues from the plots suggesting differences.