w271 Lab 2: Cereal Shelf Placement

Jessica Hays Fisher, Alice Lam, Marshall Ratliff, Paul Varjan

6/3/2018

Contents

(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. 3

(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. 4

(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? 7

(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. . . 7

(e) Show that there are no significant interactions among the explanatory variables (including an interaction among all three variables). 9

(f) Kellogg's Apple Jacks (http://www.applejacks.com) is a cereal marketed toward 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. 12

(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. 13

(h) Estimate odds ratios and calculate corresponding confidence intervals for each explanatory variable. Relate your interpretations back to the plots constructed for this exercise. 16

```r
#Load libraries and insert a function to tidy up the code when they are printed out
library(vcd)
```

```
## Loading required package: grid
```

```r
library(nnet)
library(car)
```

```
## Loading required package: carData
```

```r
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(skimr)
library(MASS)

rm(list = ls())
library(knitr)
```

```
##
## Attaching package: 'knitr'

## The following object is masked from 'package:skimr':
##
##     kable
```

```
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

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

```
## 'data.frame':    40 obs. of  7 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Shelf    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Cereal   : Factor w/ 38 levels "Basic 4","Capn Crunch",..: 17 34 19 13 16 9 2 3 30 8 ...
##  $ size_g   : int  28 28 28 32 30 31 27 27 29 33 ...
##  $ sugar_g  : int  10 2 2 2 13 11 12 9 11 2 ...
##  $ fat_g    : num  0 0 0 2 1 0 1.5 2.5 0.5 0 ...
##  $ sodium_mg: int  170 270 300 280 210 180 200 200 220 330 ...
```

```
# Examine the data to check data validity before proceeding
# with the questions.
skim(cereal)
```

```
## Skim summary statistics
##  n obs: 40
##  n variables: 7
##
## -- Variable type:factor ---------------------------------------------------------------------
##  variable missing complete  n n_unique                     top_counts
##    Cereal       0       40 40       38 Cap: 2, Foo: 2, Bas: 1, Cap: 1
##  ordered
##    FALSE
##
## -- Variable type:integer --------------------------------------------------------------------
##   variable missing complete  n  mean    sd p0    p25   p50    p75 p100
##         ID       0       40 40  20.5 11.69  1  10.75  20.5  30.25   40
##      Shelf       0       40 40    2.5  1.13  1   1.75    2.5   3.25    4
##     size_g       0       40 40   37.2 11.79 27   29.75   31     51      60
##  sodium_mg       0       40 40 195.5 81.67  0 157.5  200   262.5   330
##    sugar_g       0       40 40   10.4  5.67  0   6     11     14      20
##      hist
##
##
##
##
##
##
## -- Variable type:numeric ------------------------------------------------------------------
##  variable missing complete  n mean  sd p0 p25 p50  p75 p100     hist
##     fat_g       0       40 40  1.2 1.1  0 0.5   1 1.62    5
```

There are 7 variables with 40 observations evenly distributed across 4 shelves. There's no missing data. There are 38 types of cereal, with sugar content ranging 0 to 20 gram, fat content ranging from 0 to 5 gram, sodium content from 0 to 330 gram, serving size ranging 27 to 60 gram.

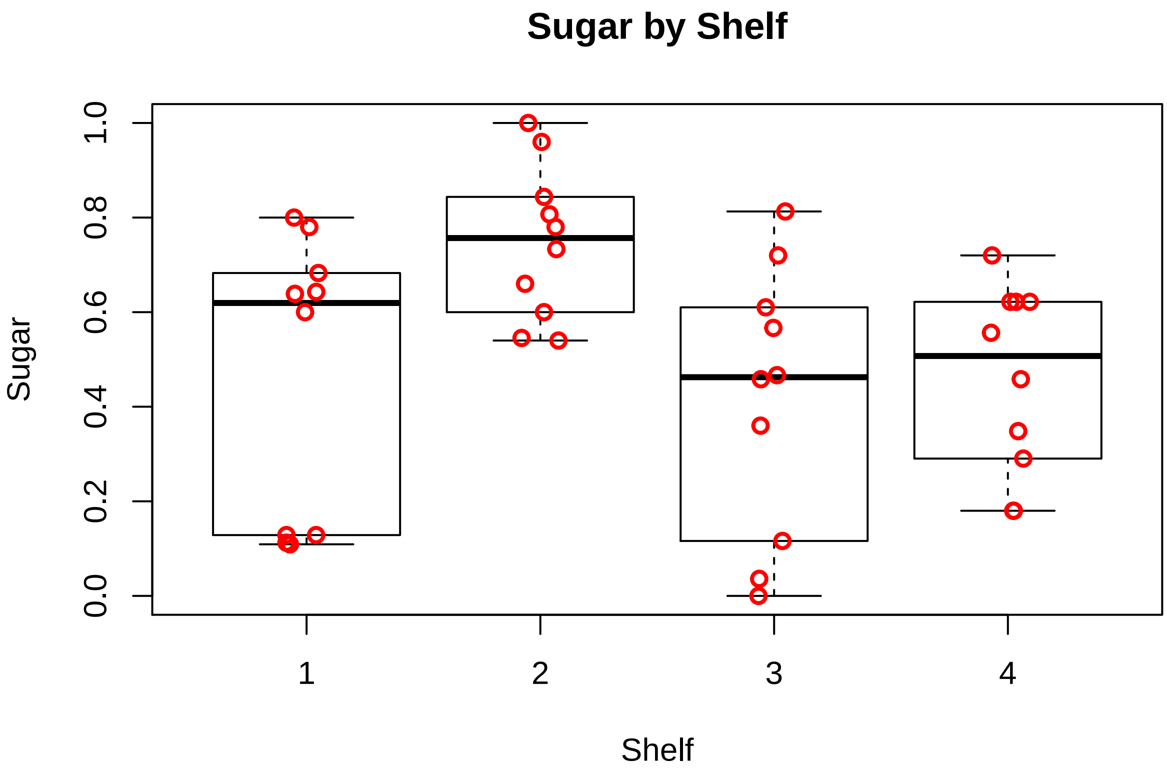
(a) The explanatory variables need to be re-formatted before proceeding further. First, divide each explanatory variable by its serving size to account for the different serving sizes among the cereals. Second, re-scale each variable to be within 0 and 1.

```r
# the formula is from textbook
standardize <- function(x) {
    (x - min(x))/(max(x) - min(x))
}
cereal2 <- data.frame(Shelf = cereal$Shelf, Cereal = cereal$Cereal,
    sugar = standardize(cereal$sugar_g/cereal$size_g), fat = standardize(cereal$fat_g/cereal$si
    sodium = standardize(cereal$sodium/cereal$size_g))

head(cereal2)
```

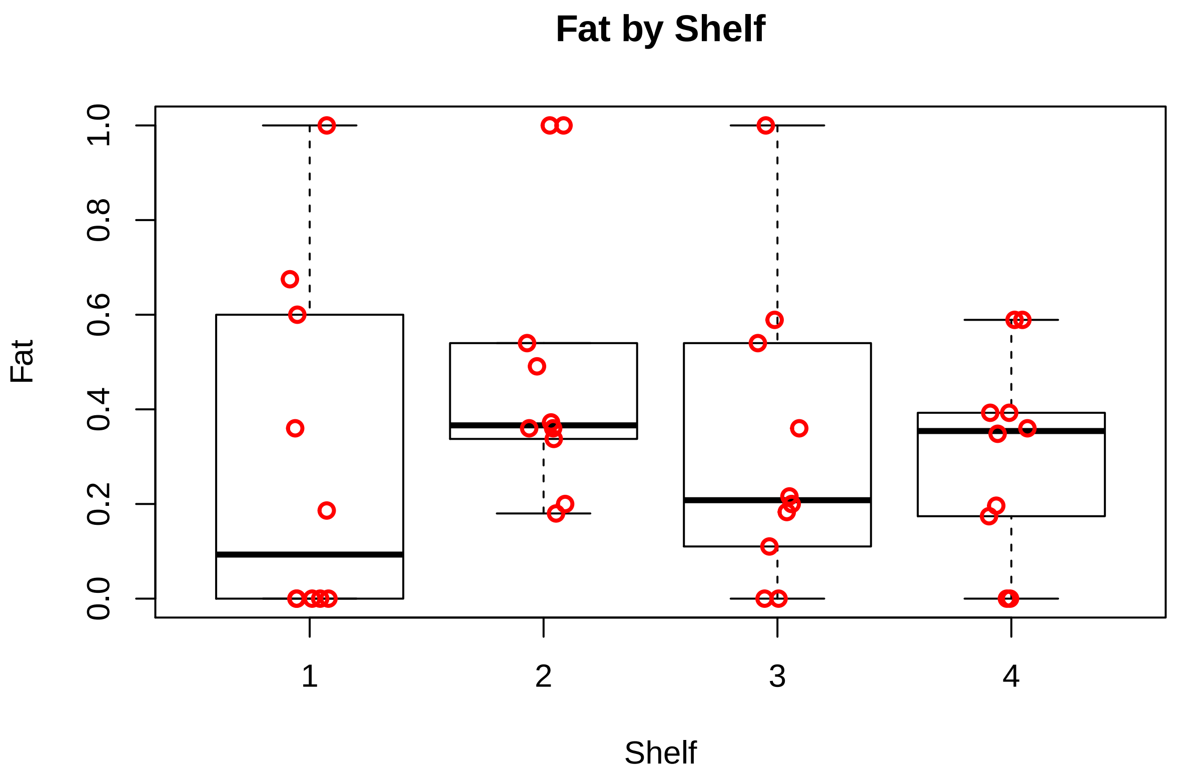
```
##   Shelf                               Cereal     sugar   fat    sodium
## 1     1 Kellog's Razzle Dazzle Rice Crispies 0.6428571 0.000 0.5666667
## 2     1            Post Toasties Corn Flakes 0.1285714 0.000 0.9000000
## 3     1                Kellogg's Corn Flakes 0.1285714 0.000 1.0000000
## 4     1               Food Club Toasted Oats 0.1125000 0.675 0.8166667
## 5     1                     Frosted Cheerios 0.7800000 0.360 0.6533333
## 6     1             Food Club Frosted Flakes 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.

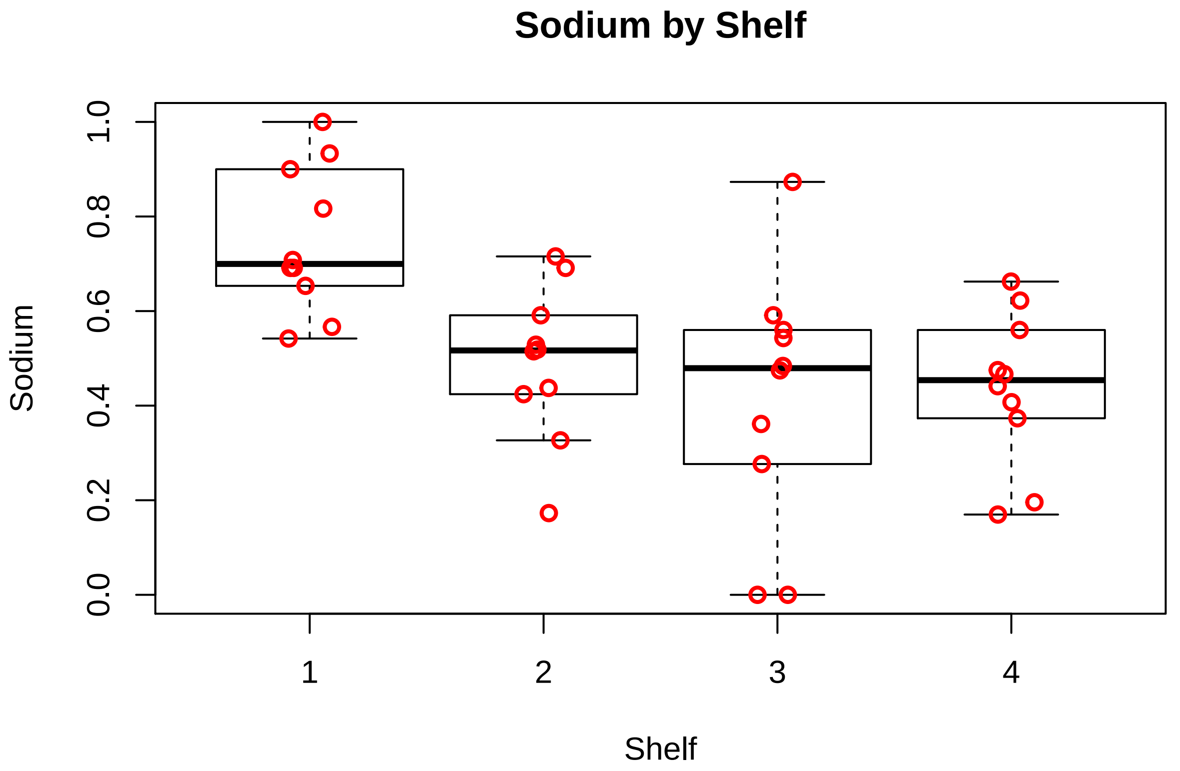
```r
# this code is from textbook
boxplot(formula = sugar ~ Shelf, data = cereal2, ylab = "Sugar",
    xlab = "Shelf", main = "Sugar by Shelf", pars = list(outpch = NA))
stripchart(x = cereal2$sugar ~ cereal2$Shelf, lwd = 2, col = "red",
    method = "jitter", vertical = TRUE, pch = 1, add = TRUE)
```



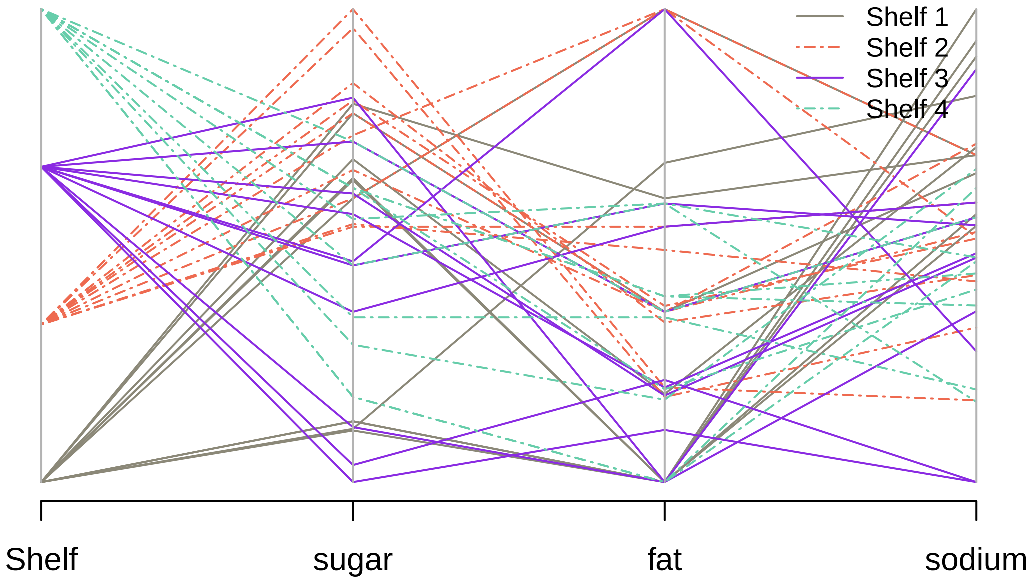
```r
boxplot(formula = fat ~ Shelf, data = cereal2, ylab = "Fat",
    xlab = "Shelf", main = "Fat by Shelf", pars = list(outpch = NA))
stripchart(x = cereal2$fat ~ cereal2$Shelf, lwd = 2, col = "red",
    method = "jitter", vertical = TRUE, pch = 1, add = TRUE)
```



```r
boxplot(formula = sodium ~ Shelf, data = cereal2, ylab = "Sodium",
    xlab = "Shelf", main = "Sodium by Shelf", pars = list(outpch = NA))
stripchart(x = cereal2$sodium ~ cereal2$Shelf, lwd = 2, col = "red",
    method = "jitter", vertical = TRUE, pch = 1, add = TRUE)
```



```
cereal3 <- data.frame(cereal2[1], cereal2[, 3:5])

# Colors by condition:
cereal.colors <- ifelse(test = cereal2$Shelf == 1, yes = "cornsilk4",
    no = ifelse(test = cereal2$Shelf == 2, yes = "coral2", no = ifelse(test = cereal2$Shelf ==
        3, yes = "blueviolet", no = "aquamarine3")))
# Line type by condition:
cereal.lty <- ifelse(test = cereal2$Shelf == 1, yes = "solid",
    no = ifelse(test = cereal2$Shelf == 2, yes = "twodash", no = ifelse(test = cereal2$Shelf =
        3, yes = "solid", no = "twodash")))

parcoord(x = cereal3, col = cereal.colors, lty = cereal.lty)  # Plot
legend(x = 3.35, y = 1.05, legend = c("Shelf 1", "Shelf 2", "Shelf 3",
    "Shelf 4"), lty = c("solid", "twodash", "solid", "twodash"),
    col = c("cornsilk4", "coral2", "blueviolet", "aquamarine3"),
    cex = 0.8, bty = "n")
```



High sugar content seems to be most prevalent among Shelf 2. The other shelves have a pretty wide spread of sugar content, with means roughly in the same places. Of note is Shelf 1's bimodal distribution of sugar, with one cluster of cereals with nearly no sugar and the other cluster having above average sugar content. Without that low sugar cluster, the rest of the shelf would have a mean sugar content much closer to Shelf 2.

Fat content seems to be pretty evenly distributed across shelves. In cereal this most likely corresponds to contents like nuts and oilseeds. There is a heavy occurrence of fat content at both extremes (1 and 0). Shelf 1 has so many 0 fat score cereals that its mean is lower than the others. Perhaps also notable is that shelf 2 is the only shelf with no cereals with a 0 score for fat, and that shelf 4 is the only shelf with no cereals with a 1 score, but visually, that information does not add muchin light of the rest of the fat content plots.

Sodium content is notably highest on Shelf 1, but otherwise the other shelves have a more or less similar mean, with Shelf 3 showing the most breadth of sodium levels within that shelf.

(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?

Answer: If we believed that there was a natural ordering to the shelves, or that they could be arranged in an order such that shelf 1 < shelf 2 < shelf 3, etc. - then we would be desirable to take into account ordinality (especially if we believed that the "distance" between each level were equal). However, we do not believe that is the case with this data, as it is not clear whether being on a low shelf is objectively better than on a high shelf, or vice versa. There are attractors/detractors from each shelf height and for different customers - for example, children are at the height of lower shelves than adults are - but that ordering is not universal and therefore not desireable to take into account in our modeling. If other data could be brought in that demonstrated the desirability or marketability of each shelf had some order (which probably does exist), that could also be used as a factor for ordinality.

(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.

```r
levels(as.factor(cereal2$Shelf))
```

```
## [1] "1" "2" "3" "4"
```

```r
mod1 <- multinom(as.factor(Shelf) ~ sugar + fat + sodium, data = cereal2)
```

```
## # weights:  20 (12 variable)
## initial  value 55.451774
## iter  10 value 37.329384
## iter  20 value 33.775257
## iter  30 value 33.608495
## iter  40 value 33.596631
## iter  50 value 33.595909
## iter  60 value 33.595564
## iter  70 value 33.595277
## iter  80 value 33.595147
## final  value 33.595139
## converged
```

```r
summary(mod1)
```

```
## Call:
## multinom(formula = as.factor(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
##
## Residual Deviance: 67.19028
## AIC: 91.19028
```

```
Anova(mod1)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: as.factor(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
```

- We cannot use mcprofile package for likelihood ratio, as nnet package author does not believe that one at a time intervals should be calculated. We use value of c equal to 1 standard deviation instead. *

```
sd.cereal <- apply(X = cereal2[, -c(2)], MARGIN = 2, FUN = sd)
c.value <- c(sd.cereal)[2:4]
round(c.value, 2)
```

```
##  sugar    fat sodium
##   0.27   0.30   0.23
```

```
# coefficients(mod1)
beta.hat2 <- coefficients(mod1)[1, 2:4]
beta.hat3 <- coefficients(mod1)[2, 2:4]
beta.hat4 <- coefficients(mod1)[3, 2:4]
# OR for j = 2 (Shelf 2 vs Shelf 1)
print("OR for j = 2")
```

```
## [1] "OR for j = 2"
```

```
round(exp(c.value * beta.hat2), 2)
```

```
##  sugar    fat sodium
##   2.06   3.37   0.02
```

```
round(1/exp(c.value * beta.hat2), 2)
```

```
##  sugar    fat sodium
##   0.48   0.30  55.74
```

```
# OR for j = 3 (Shelf 3 vs Shelf 1)
print("OR for j = 3")
```

```
## [1] "OR for j = 3"
```

```
round(exp(c.value * beta.hat3), 2)
```

```
##  sugar    fat sodium
##   0.04   0.85   0.00
```

```
round(1/exp(c.value * beta.hat3), 2)
```

```
##  sugar    fat sodium
##  26.81   1.18 311.36
```

```
# OR for j = 4 (Shelf 4 vs Shelf 1)
print("OR for j = 3")
```

```
## [1] "OR for j = 3"
```

```
round(exp(c.value * beta.hat4), 2)
```

```
##  sugar    fat sodium
##   0.05   0.77   0.00
```

```
round(1/exp(c.value * beta.hat4), 2)
```

```
##  sugar    fat sodium
##  21.48   1.30 290.31
```

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

```
## Not actually sure if this is what the question means?

modA <- multinom(as.factor(Shelf) ~ sugar + fat + sodium + sugar:fat,
    data = cereal2)
```

```
## # weights:  24 (15 variable)
## initial  value 55.451774
## iter  10 value 37.518284
## iter  20 value 33.008954
## iter  30 value 31.923790
## iter  40 value 31.590382
## iter  50 value 31.417819
## iter  60 value 31.212591
## iter  70 value 31.158365
## iter  80 value 31.068476
## iter  90 value 31.042861
## iter 100 value 30.998934
## final  value 30.998934
## stopped after 100 iterations
```

```
Anova(modA)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: as.factor(Shelf)
##           LR Chisq Df Pr(>Chisq)
## sugar      22.7648  3  4.521e-05 ***
## fat         5.2836  3     0.1522
## sodium     30.8407  3  9.183e-07 ***
## sugar:fat   5.1924  3     0.1582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
modB <- multinom(as.factor(Shelf) ~ sugar + fat + sodium + sugar:sodium,
    data = cereal2)
```

```
## # weights:  24 (15 variable)
## initial  value 55.451774
## iter  10 value 36.577939
## iter  20 value 33.026993
## iter  30 value 32.740384
## iter  40 value 32.604061
## iter  50 value 32.452790
## iter  60 value 32.427677
## iter  70 value 32.423013
## iter  80 value 32.420834
## iter  90 value 32.420382
## iter 100 value 32.420219
## final  value 32.420219
## stopped after 100 iterations
```

```
Anova(modB)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: as.factor(Shelf)
##              LR Chisq Df Pr(>Chisq)
## sugar         22.7648  3  4.521e-05 ***
## fat            6.1167  3     0.1061
## sodium        26.6197  3  7.073e-06 ***
## sugar:sodium   2.3498  3     0.5030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
modC <- multinom(as.factor(Shelf) ~ sugar + fat + sodium + fat:sodium,
    data = cereal2)
```

```
## # weights:  24 (15 variable)
## initial  value 55.451774
## iter  10 value 36.936754
## iter  20 value 32.546388
## iter  30 value 32.211409
## iter  40 value 32.085217
## iter  50 value 31.709427
## iter  60 value 31.095835
## iter  70 value 30.872944
## iter  80 value 30.734369
## iter  90 value 30.653914
## iter 100 value 30.639405
## final  value 30.639405
## stopped after 100 iterations
```

```r
Anova(modC)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: as.factor(Shelf)
##            LR Chisq Df Pr(>Chisq)
## sugar       19.2525  3  0.0002424 ***
## fat          5.2836  3  0.1521727
## sodium      26.6197  3  7.073e-06 ***
## fat:sodium   5.9115  3  0.1159978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
modD <- multinom(as.factor(Shelf) ~ sugar + fat + sodium + fat:sodium:sugar,
    data = cereal2)
```

```
## # weights:  24 (15 variable)
## initial  value 55.451774
## iter  10 value 37.410894
## iter  20 value 33.855213
## iter  30 value 33.056977
## iter  40 value 32.715781
## iter  50 value 32.677949
## iter  60 value 32.592524
## iter  70 value 32.578128
## iter  80 value 32.550159
## iter  90 value 32.541053
## iter 100 value 32.535394
## final  value 32.535394
## stopped after 100 iterations
```

```r
Anova(modD)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: as.factor(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 ***
## sugar:fat:sodium   2.1195  3     0.5480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
modE <- multinom(as.factor(Shelf) ~ sugar + fat + sodium + sugar:fat +
    sugar:sodium + fat:sodium + fat:sodium:sugar, data = cereal2)
```

```
## # 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
```

```
Anova(modE)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: as.factor(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   2.5884  3  0.4595299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(f) Kellogg's Apple Jacks (http://www.applejacks.com) is a cereal marketed toward 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.

Alice's solution:

```
# recall rescale function from textbook stand01 <-
# function(x) { (x - min(x))/(max(x) - min(x)) } cereal has
# the pre-scaled variable numbers. Don't use cereal2
stand02 <- function(x, variable) {
    (x - min(variable))/(max(variable) - min(variable))
}
sugar1 <- stand02(x = 12/28, variable = cereal$sugar)
fat1 <- stand02(x = 0.5/28, variable = cereal$fat)
sodium1 <- stand02(x = 1.3/28, variable = cereal$sodium)
newdata <- data.frame(sugar = sugar1, fat = fat1, sodium = sodium1)
lin.pred.hat <- predict(object = mod1, newdata = newdata, type = "probs",
    se = TRUE)
round(lin.pred.hat, 7)
```

```
##         1         2         3         4
## 0.0000000 0.0000003 0.5928551 0.4071446
```

Jessica's solution:

```
stand.new <- function(meas, serv_size, comparison) {
    (meas/serv_size - min(comparison))/(max(comparison) - min(comparison))
}

newdata <- data.frame(sugar = stand.new(12, 28, cereal$sugar_g/cereal$size_g),
    fat = stand.new(0.5, 28, cereal$fat_g/cereal$size_g), sodium = stand.new(130,
        28, cereal$sodium_mg/cereal$size_g))

round(predict(object = mod1, newdata = newdata, type = "probs",
    se.fit = TRUE), 7)
```

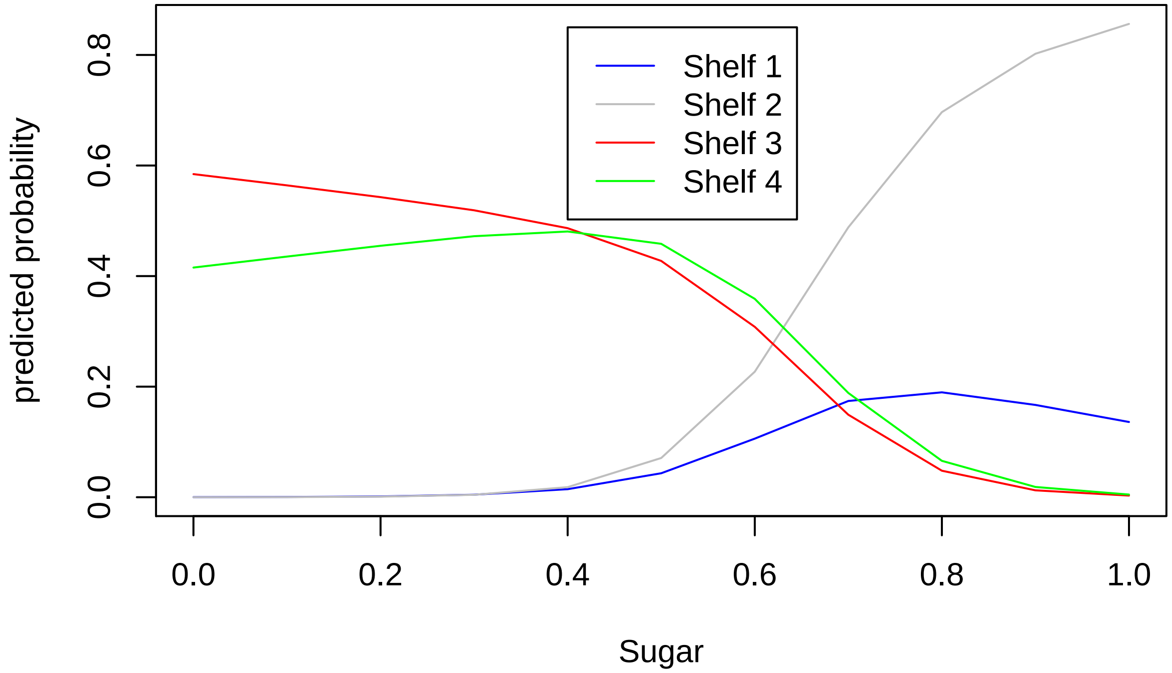
```
##         1         2         3         4
## 0.0532685 0.4719426 0.2004274 0.2743615
```

(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.

Alice's solution:

```
newfat <- c(rep(mean(cereal2$fat), 11))
newsodium <- c(rep(mean(cereal2$sodium), 11))
newsugar <- c(seq(0, 1, 0.1))
simulated.data <- data.frame(sugar = newsugar, fat = newfat,
    sodium = newsodium)
pi.hat.sim <- predict(mod1, newdata = simulated.data, type = "probs")
plot.new()
x <- seq(0, 1, 0.1)
```

```
plot(x, pi.hat.sim[, 1], type = "l", col = "blue", ylim = range(min(pi.hat.sim),
    max(pi.hat.sim)), xlab = "Sugar", ylab = "predicted probability")
lines(x, pi.hat.sim[, 2], col = "gray")
lines(x, pi.hat.sim[, 3], col = "red")
lines(x, pi.hat.sim[, 4], col = "green")

legend(x = 0.4, y = 0.85, legend = c("Shelf 1", "Shelf 2", "Shelf 3",
    "Shelf 4"), col = c("blue", "gray", "red", "green"), lty = "solid")
```



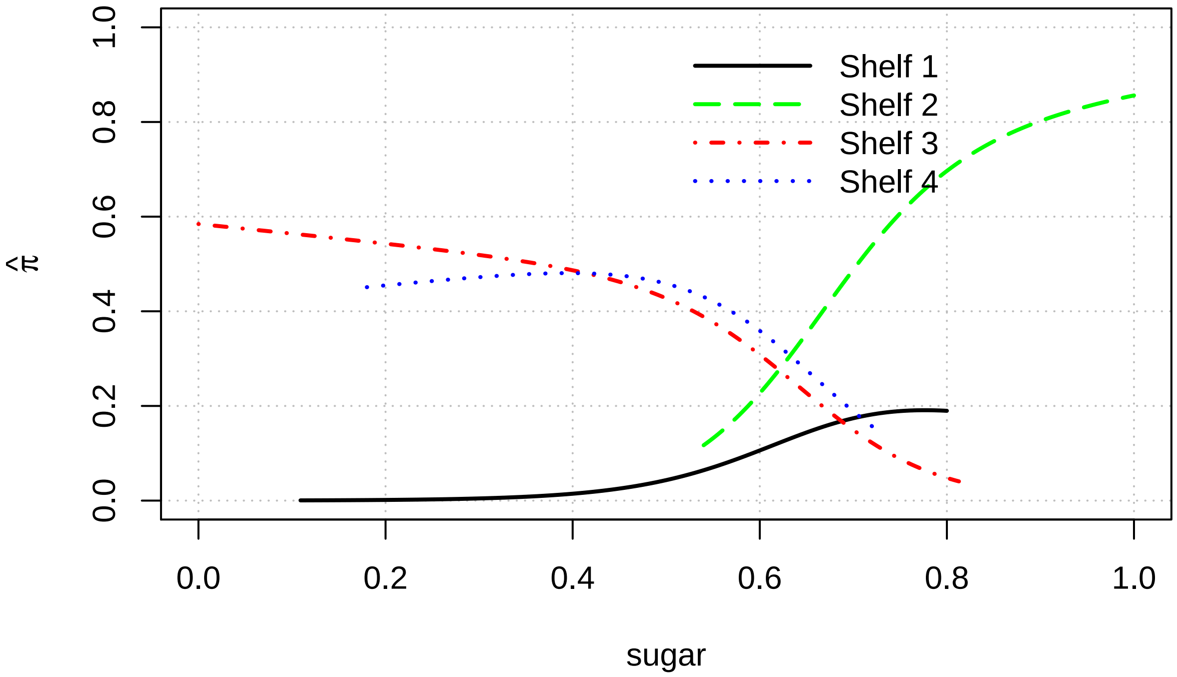
Jessica's solution:

```
## Open question: in figure 3.3, they have each line stop at
## the min/max for each type of kernel (equivalent to each
## shelf in our example). Is this what we want, or do we want
## it to go from 0 to 1 for each shelf? (code for both below,
## 0-1 is commented out.)


beta.hat <- coefficients(mod1)
beta.hat
```

```
##   (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
```

```
mean_fat <- mean(cereal2$fat)
mean_sodium <- mean(cereal2$sodium)
```

```
# Create plotting area first to make sure get the whole
# region with respect to x-axis
curve(expr = 1/(1 + exp(beta.hat[1, 1] + beta.hat[1, 2] * x) +
    exp(beta.hat[2, 1] + beta.hat[2, 2] * x)), ylab = expression(hat(pi)),
    xlab = "sugar", xlim = c(min(cereal2$sugar), max(cereal2$sugar)),
    ylim = c(0, 1), col = "black", lty = "solid", lwd = 2, n = 1000,
    type = "n", panel.first = grid(col = "gray", lty = "dotted"))

## Plot each pi_j Shelf1
curve(expr = 1/(1 + exp(beta.hat[1, 1] + beta.hat[1, 2] * x +
    beta.hat[1, 3] * mean_fat + beta.hat[1, 4] * mean_sodium) +
    exp(beta.hat[2, 1] + beta.hat[2, 2] * x + beta.hat[2, 3] *
        mean_fat + beta.hat[2, 4] * mean_sodium) + exp(beta.hat[3,
    1] + beta.hat[3, 2] * x + beta.hat[3, 3] * mean_fat + beta.hat[3,
    4] * mean_sodium)), col = "black", lty = "solid", lwd = 2,
    n = 1000, add = TRUE, xlim = c(min(cereal2$sugar[cereal2$Shelf ==
        1]), max(cereal2$sugar[cereal2$Shelf == 1])))
# xlim = c(0,1))

# Shelf2
curve(expr = exp(beta.hat[1, 1] + beta.hat[1, 2] * x + beta.hat[1,
    3] * mean_fat + beta.hat[1, 4] * mean_sodium)/(1 + exp(beta.hat[1,
    1] + beta.hat[1, 2] * x + beta.hat[1, 3] * mean_fat + beta.hat[1,
    4] * mean_sodium) + exp(beta.hat[2, 1] + beta.hat[2, 2] *
    x + beta.hat[2, 3] * mean_fat + beta.hat[2, 4] * mean_sodium) +
    exp(beta.hat[3, 1] + beta.hat[3, 2] * x + beta.hat[3, 3] *
        mean_fat + beta.hat[3, 4] * mean_sodium)), col = "green",
    lty = "longdash", lwd = 2, n = 1000, add = TRUE, xlim = c(min(cereal2$sugar[cereal2$Shelf
        2]), max(cereal2$sugar[cereal2$Shelf == 2])))
# xlim = c(0,1))

# Shelf3
curve(expr = exp(beta.hat[2, 1] + beta.hat[2, 2] * x + beta.hat[2,
    3] * mean_fat + beta.hat[2, 4] * mean_sodium)/(1 + exp(beta.hat[1,
    1] + beta.hat[1, 2] * x + beta.hat[1, 3] * mean_fat + beta.hat[1,
    4] * mean_sodium) + exp(beta.hat[2, 1] + beta.hat[2, 2] *
    x + beta.hat[2, 3] * mean_fat + beta.hat[2, 4] * mean_sodium) +
    exp(beta.hat[3, 1] + beta.hat[3, 2] * x + beta.hat[3, 3] *
        mean_fat + beta.hat[3, 4] * mean_sodium)), col = "red",
    lty = "dotdash", lwd = 2, n = 1000, add = TRUE, xlim = c(min(cereal2$sugar[cereal2$Shelf ==
        3]), max(cereal2$sugar[cereal2$Shelf == 3])))
# xlim = c(0,1))

# Shelf4
curve(expr = exp(beta.hat[3, 1] + beta.hat[3, 2] * x + beta.hat[3,
    3] * mean_fat + beta.hat[3, 4] * mean_sodium)/(1 + exp(beta.hat[1,
    1] + beta.hat[1, 2] * x + beta.hat[1, 3] * mean_fat + beta.hat[1,
    4] * mean_sodium) + exp(beta.hat[2, 1] + beta.hat[2, 2] *
    x + beta.hat[2, 3] * mean_fat + beta.hat[2, 4] * mean_sodium) +
    exp(beta.hat[3, 1] + beta.hat[3, 2] * x + beta.hat[3, 3] *
        mean_fat + beta.hat[3, 4] * mean_sodium)), col = "blue",
    lty = "dotted", lwd = 2, n = 1000, add = TRUE, xlim = c(min(cereal2$sugar[cereal2$Shelf ==
        4]), max(cereal2$sugar[cereal2$Shelf == 4])))
# xlim = c(0,1))




legend(x = 0.5, y = 1, legend = c("Shelf 1", "Shelf 2", "Shelf 3",
    "Shelf 4"), lty = c("solid", "longdash", "dotdash", "dotted"),
    col = c("black", "green", "red", "blue"), bty = "n", lwd = c(2,
        2, 2), seg.len = 4)
```



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