W271 Assignment 2

Due 11:59pm Pacific Time, Sunday February 23, 2020

Hersh SOlanki

Instructions (Please Read Carefully):

- **Late submissions will not be accepted**
- No page limit, but be reasonable
- Do not modify fontsize, margin or line_spacing settings
- This assignment needs to be completed individually; this is not a group project
- Submission is by pushing to your student fork of the course repository
- Submit two files:
  1. A pdf file that details your answers (knit to pdf, do not knit to html then save as pdf). Include all R code used to produce the answers. Do not suppress the code in your pdf file
  2. The R markdown (Rmd) file used to produce the pdf file

  The assignment will not be graded unless **both** files are submitted
- Use the following file-naming convensation:
  - StudentFirstNameLastName_HWNumber.fileExtension
  - For example, if the student's name is Kyle Cartman for assignment 1, name your files follows:
    - KyleCartman_assignment2.Rmd
    - KyleCartman_assignment2.pdf
- Although it sounds obvious, please write your name on page 1 of your pdf and Rmd files
- Answers should clearly explain your reasoning; do not simply 'output dump' the results of code without explanation
- For statistical methods that we cover in this course, use the R libraries and functions that are covered in this course. If you use libraries and functions for statistical modeling that we have not covered, you must provide an explanation of why such libraries and functions are used and reference the library documentation. For data wrangling and data visualization, you are free to use other libraries, such as dplyr, ggplot2, etc.
- For mathematical formulae, type them in your R markdown file. Do not e.g. write them on a piece of paper, snap a photo, and use the image file.
- Incorrectly following submission instructions results in deduction of grades
- Students are expected to act with regard to UC Berkeley Academic Integrity

1. Strategic Placement of Products in Grocery Stores (5 points)

These questions are taken from Question 12 of chapter of the textbook.

In order to maximize sales, items within grocery stores are strategically placed to draw customer attention. This exercise examines one type of item—breakfast cereal. Typically, in large grocery stores, boxes of cereal are placed on sets of shelves located on one side of the aisle. By placing particular boxes of cereals on specific shelves, grocery stores may better attract customers to them. To investigate this further, a random sample of size 10 was taken from each of four shelves at a Dillons grocery store in Manhattan, KS. These data are given in the cereal_dillons.csv *file. The response variable is the shelf number, which is numbered from bottom (1) to top (4), and the explanatory variables are the sugar, fat, and sodium content of the cereals.*

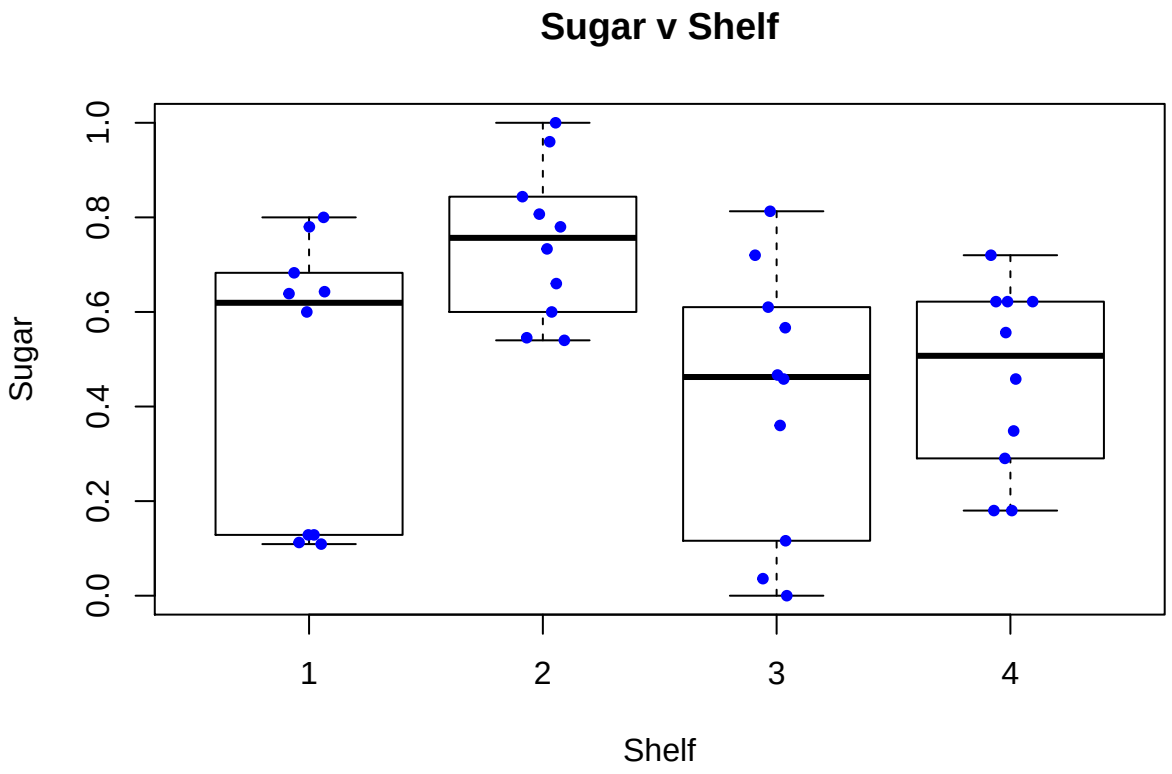
1.1 (1 point): The explanatory variables need to be reformatted before proceeding further (sample code is provided in the textbook). First, divide each explanatory variable by its serving size to account for the different serving sizes among the cereals. Second, rescale each variable to be within 0 and 1. 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 whether possible content differences exist among the shelves.

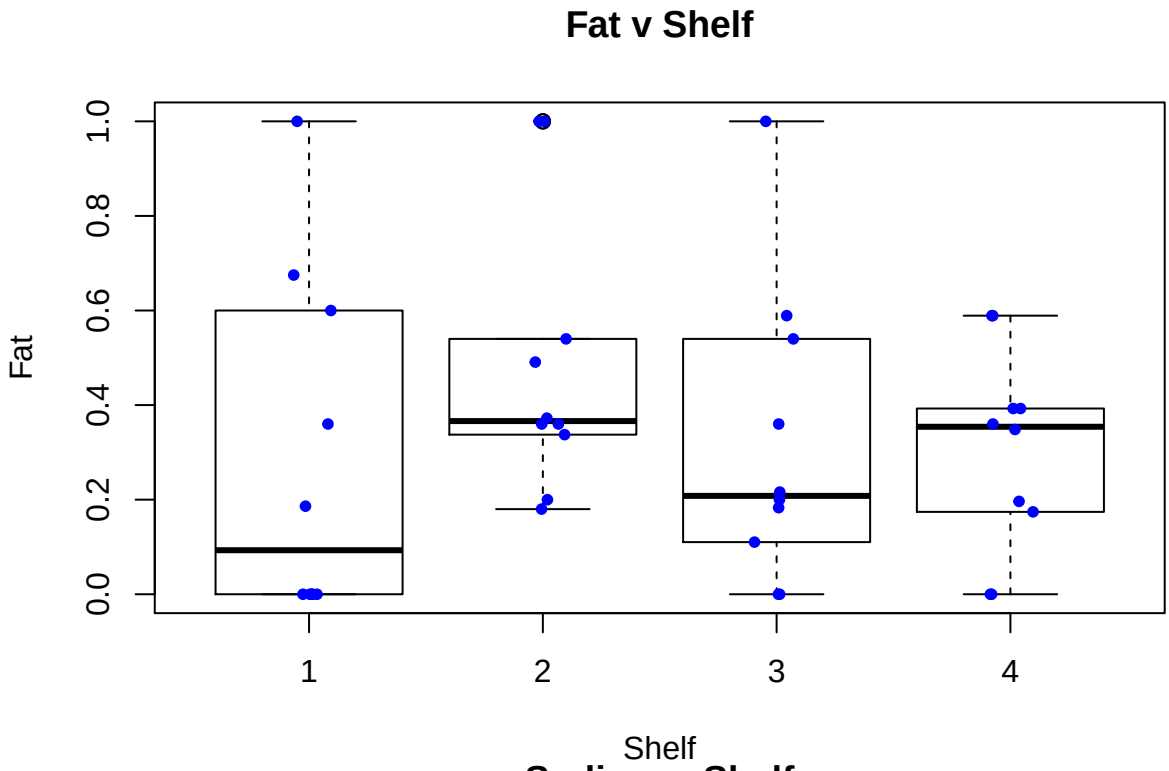
```
library(MASS)

d.1 <- read.csv("cereal_dillons.csv", header=TRUE, sep=",")
d.1$sugar_standard <- d.1$sugar_g / d.1$size_g
d.1$fat_standard <- d.1$fat_g / d.1$size_g
d.1$sodium_standard <- (d.1$sodium_mg) / d.1$size_g



d.1$sugar_standard <- (d.1$sugar_standard - min(d.1$sugar_standard)) / (max(d.1$sugar_standard)

d.1$fat_standard <- (d.1$fat_standard - min(d.1$fat_standard)) / (max(d.1$fat_standard) - min(d

d.1$sodium_standard <- (d.1$sodium_standard - min(d.1$sodium_standard)) / (max(d.1$sodium_stand

head(d.1)
```

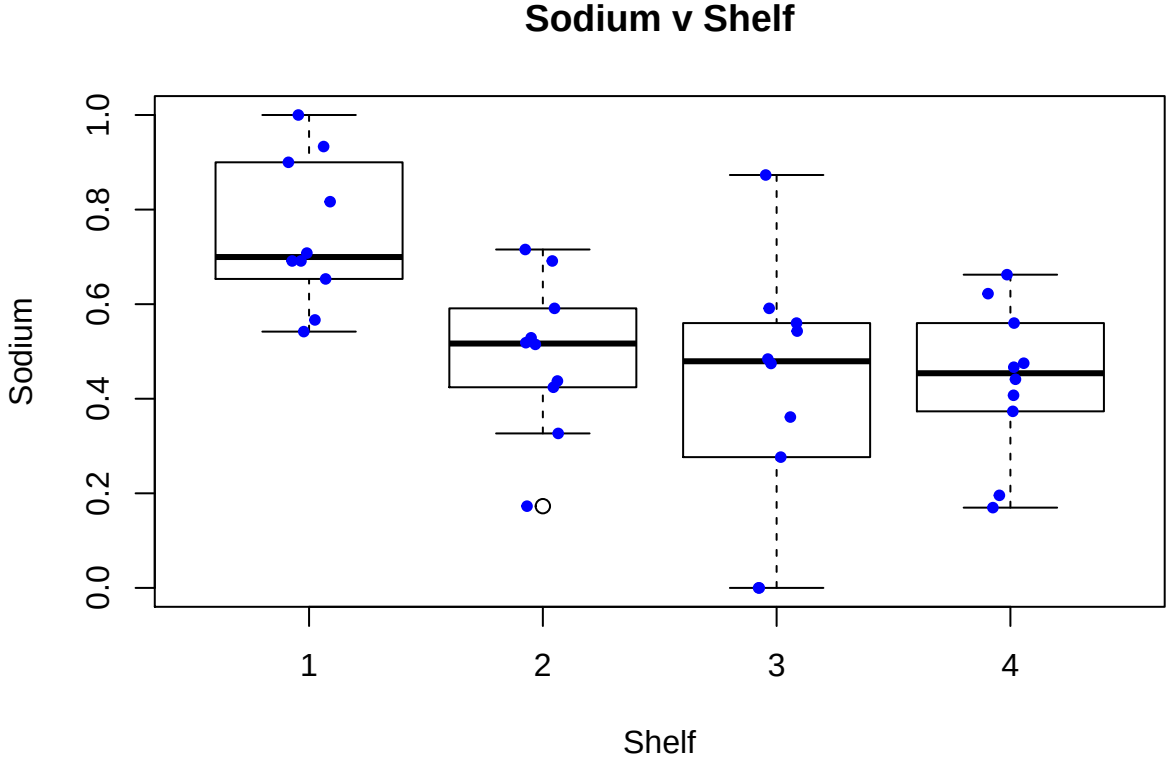
```
##   ID Shelf                                Cereal size_g sugar_g fat_g
## 1  1     1 Kellog's Razzle Dazzle Rice Crispies     28      10     0
## 2  2     1            Post Toasties Corn Flakes     28       2     0
## 3  3     1                Kellogg's Corn Flakes     28       2     0
## 4  4     1               Food Club Toasted Oats     32       2     2
## 5  5     1                     Frosted Cheerios     30      13     1
## 6  6     1             Food Club Frosted Flakes     31      11     0
##   sodium_mg sugar_standard fat_standard sodium_standard
## 1       170      0.6428571        0.000       0.5666667
## 2       270      0.1285714        0.000       0.9000000
## 3       300      0.1285714        0.000       1.0000000
## 4       280      0.1125000        0.675       0.8166667
## 5       210      0.7800000        0.360       0.6533333
```

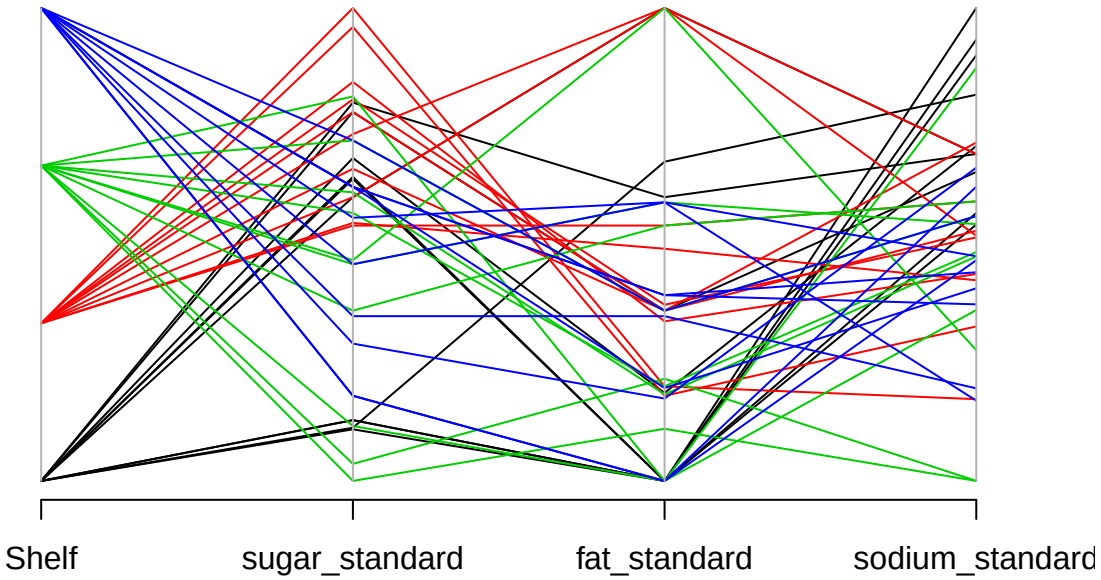
```
## 6       180       0.6387097         0.000       0.5419355
```

```r
# Using as reference - https://stackoverflow.com/questions/23675735/how-to-add-boxplots-to-sca
boxplot(sugar_standard ~ Shelf, data = d.1, ylab = "Sugar", xlab = "Shelf", main = "Sugar v She
stripchart(sugar_standard ~ Shelf, vertical = TRUE, data = d.1,
    method = "jitter", add = TRUE, pch = 20, col = 'blue')

boxplot(fat_standard ~ Shelf, data = d.1, ylab = "Fat", xlab = "Shelf", main = "Fat v Shelf")
stripchart(fat_standard ~ Shelf, vertical = TRUE, data = d.1,
    method = "jitter", add = TRUE, pch = 20, col = 'blue')

boxplot(sodium_standard ~ Shelf, data = d.1, ylab = "Sodium", xlab = "Shelf", main = "Sodium v
stripchart(sodium_standard ~ Shelf, vertical = TRUE, data = d.1,
    method = "jitter", add = TRUE, pch = 20, col = 'blue')

parcoord(x = d.1[, c("Shelf", "sugar_standard", "fat_standard" ,"sodium_standard")], col = d.1
```




Fat v Shelf
Fat
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
Shelf


Sodium v Shelf
Sodium
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
Shelf



Shelf 2 seems to have the most sugary prodcuts, while shelf 1 seems to have the ones with the most sodium In terms of fat, it is a bit more varied, but shelf 1/3 once again has the fattiest products.

1.2 (1 point): The response has values of $1, 2, 3$, and 4. Explain under what setting would it be desirable to take into account ordinality, and whether you think that this setting occurs here. Then estimate a suitable multinomial regression model with linear forms of the sugar, fat, and sodium variables. Perform LRTs to examine the importance of each explanatory variable. Show that there are no significant interactions among the explanatory variables (including an interaction among all three variables).

It could make sense to look at ordinality if certain shelves were significanlty more profitable than others. As a result, certain cerial makers would want to pay a lot more for a certain shelf vs another shelf. However, we don't have this information, so we ignore ordinality in this case.

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

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

```
d.1$Shelf <- factor(d.1$Shelf)
model.1.2 <- multinom(Shelf ~ sugar_standard + fat_standard + sodium_standard, data = d.1)
```

```
## # 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
```

```
summary(model.1.2)
```

```
## Call:
## multinom(formula = Shelf ~ sugar_standard + fat_standard + sodium_standard,
##     data = d.1)
##
## Coefficients:
##   (Intercept) sugar_standard fat_standard sodium_standard
## 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_standard fat_standard sodium_standard
## 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
```

The equation of the model for 1 v 2 is: 6.900708 + 2.693071(sugar standard) + 4.0647092(fat_standard) - 17.49373(sodium_standard).

The equation of the model for 1 v 3 is: 21.680680 - 12.216442(sugar standard) - 0.5571273(fat_standard) - 24.97850(sodium_standard).

The equation of the model for 1 v 4 is: 21.288343 - 11.393710(sugar standard) - 0.8701180(fat_standard) - 24.67385(sodium_standard).

```
Anova(model.1.2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
##                 LR Chisq Df Pr(>Chisq)
## sugar_standard   22.7648  3  4.521e-05 ***
## fat_standard      5.2836  3     0.1522
## sodium_standard  26.6197  3  7.073e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that sugar and sodium are statsitically significant, but fat is not.

```
model.combined<- multinom(Shelf ~ sugar_standard * fat_standard * sodium_standard , data = d.1)
```

```
## # 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(model.combined)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Shelf
##                                            LR Chisq Df Pr(>Chisq)
## sugar_standard                              19.2525  3  0.0002424 ***
## fat_standard                                 6.1167  3  0.1060686
## sodium_standard                             30.8407  3  9.183e-07 ***
## sugar_standard:fat_standard                  3.2309  3  0.3573733
## sugar_standard:sodium_standard               3.0185  3  0.3887844
## fat_standard:sodium_standard                 3.1586  3  0.3678151
## sugar_standard:fat_standard:sodium_standard  2.5884  3  0.4595299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that none of the interactions have any significance. The individual sugar and sodium effects are significant, but the interactions are not.

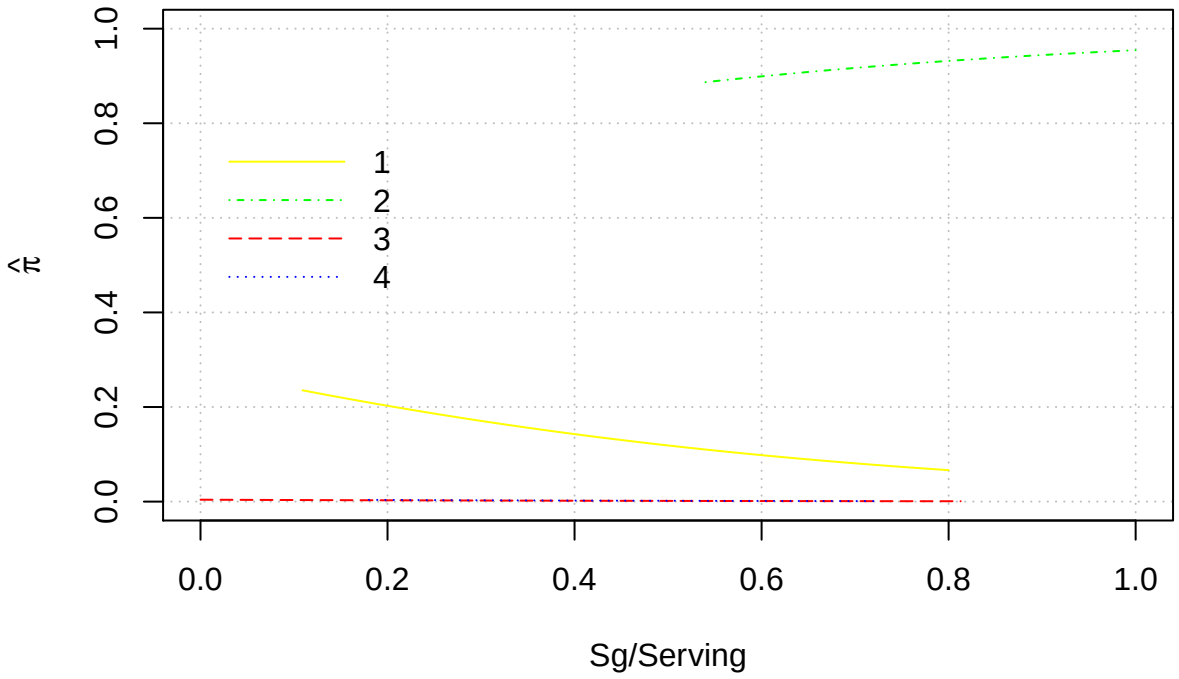
1.3 (1 point): Kellogg's Apple Jacks (http://www.applejacks.com) is a cereal marketed toward children. For a serving size of 28 grams, its sugar content is 12 grams, fat content is 0.5 grams, and sodium content is 130 milligrams. Estimate the shelf probabilities for Apple Jacks.

```
serving <- 28
sugar <- 12 / serving
fat <- 0.5 / serving
sodium <- 130 / serving

# Use this formula - https://stats.stackexchange.com/questions/70801/how-to-normalize-data-to-
sugar_sc <- (sugar - min(d.1$sugar_g / d.1$size_g)) / (max(d.1$sugar_g / d.1$size_g) - min(d.1
# print(sugar_sc)
fat_sc <- (fat - min(d.1$fat_g / d.1$size_g)) / (max(d.1$fat_g / d.1$size_g) - min(d.1$fat_g /
# print(fat_sc)
sodium_sc <- (sodium - min(d.1$sodium_mg / d.1$size_g)) / (max(d.1$sodium_mg / d.1$size_g) - m
# print(sodium_sc)

predictDF <- data.frame(sugar_standard = sugar_sc, fat_standard = fat_sc, sodium_standard = so

predict(object = model.1.2, predictDF, type = "probs")
```

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

1.4 (1 point): 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.

```
beta.hat <- coefficients(model.1.2)
fat_mean <- mean(d.1$fat_standard)
sodium_mean <- mean(d.1$sodium_standard)

curve(expr = 1/(1 + exp(beta.hat[1,2]*fat_mean + beta.hat[1,3]*x*sodium_mean) + exp(beta.hat[2
      xlim = c(0, 1), ylim = c(0,1), col = "black", lty = "solid", lwd = 2, n = 1000, type = "
      panel.first = grid(col = "gray", lty = "dotted"))

lwd.mult<-2

curve(expr = 1/(1 + exp(beta.hat[1,2]*fat_mean + beta.hat[1,3]*x*sodium_mean) + exp(beta.hat[2
      col = "yellow", lty = "solid", n = 1000, add = TRUE,
      xlim = c(min(d.1$sugar_standard[d.1$Shelf==1]), max(d.1$sugar_standard[d.1$Shelf==1])))


# For #2
curve(expr = exp(beta.hat[1,2]*fat_mean + beta.hat[1,3]*x*sodium_mean)/(1 + exp(beta.hat[1,2]*
      col = "green", lty = "dotdash", n = 1000, add = TRUE,
      xlim = c(min(d.1$sugar_standard[d.1$Shelf==2]), max(d.1$sugar_standard[d.1$Shelf==2])))

curve(expr = exp(beta.hat[2,2]*fat_mean + beta.hat[2,3]*x*sodium_mean)/(1 + exp(beta.hat[1,2]*
      col = "red", lty = "longdash", n = 1000, add = TRUE,
      xlim = c(min(d.1$sugar_standard[d.1$Shelf==3]), max(d.1$sugar_standard[d.1$Shelf==3])))

curve(expr = exp(beta.hat[3,2]*fat_mean + beta.hat[3,3]*x*sodium_mean)/(1 + exp(beta.hat[1,2]*
      col = "blue", lty = "dotted", n = 1000, add = TRUE,
      xlim = c(min(d.1$sugar_standard[d.1$Shelf==4]), max(d.1$sugar_standard[d.1$Shelf==4])))


legend(x = 0, y = 0.8, legend=c(1, 2, 3, 4), lty=c( "solid", "dotdash","longdash","dotted"),
      col=c("yellow", "green","red","blue"), bty="n", seg.len = 4)
```



1.5 (1 point): Estimate odds ratios and calculate corresponding confidence intervals for each explanatory variable. Relate your interpretations back to the plots constructed for this exercise.

```
# For 1 v 2
sd.cereal <- apply(X = d.1[8:10], MARGIN = 2, FUN = sd)

# See what the standard deviations are in order to interpret the odds
print(sd.cereal)
```

```
##  sugar_standard    fat_standard sodium_standard
##       0.2692078       0.2990292       0.2298359
```

```
c.val <- c(1, sd.cereal)
bhat <- coefficients(model.1.2)[1, 1:4]
round(exp(c.val*bhat), 2)[2:4]
```

```
##  sugar_standard    fat_standard sodium_standard
##            2.06            3.37            0.02
```

```
conf.beta <- confint(object = model.1.2, level = 0.95)
round(data.frame(low = exp(c.val*conf.beta[1:4, 1:2, 1])[,1], up = exp(c.val*conf.beta[1:4, 1:
```

```
##                   low    up
## sugar_standard  0.14 29.68
## fat_standard    0.87 13.04
## sodium_standard 0.00  0.44
```

For a 0.27 decrease in sugar, the odds of shelf 1 v 2 incease by 2.06x. For a 0.30 decrease in fat, the odds of shelf 1 v 2 incease by 3.37x. For a 0.23 decrease in sodium, the odds of shelf 1 v 2 incease by 0.02x. Without loss of generality, we can conduct the same analysis on shelf 1 v 3 and shelf 1 v 4.

```r
# For 1 v 3
sd.cereal <- apply(X = d.1[8:10], MARGIN = 2, FUN = sd)
c.val <- c(1, sd.cereal)
bhat <- coefficients(model.1.2)[2, 1:4]
round(exp(c.val*bhat), 2)[2:4]
```

```
##  sugar_standard    fat_standard sodium_standard
##            0.04            0.85            0.00
```

```r
conf.beta <- confint(object = model.1.2, level = 0.95)
round(data.frame(low = exp(c.val*conf.beta[1:4, 1:2, 2])[,1], up = exp(c.val*conf.beta[1:4, 1:
```

```
##                   low   up
## sugar_standard  0.00 0.49
## fat_standard    0.21 3.49
## sodium_standard 0.00 0.12
```

```r
# For 1 v 4
sd.cereal <- apply(X = d.1[8:10], MARGIN = 2, FUN = sd)
c.val <- c(1, sd.cereal)
bhat <- coefficients(model.1.2)[3, 1:4]
round(exp(c.val*bhat), 2)[2:4]
```

```
##  sugar_standard    fat_standard sodium_standard
##            0.05            0.77            0.00
```

```r
conf.beta <- confint(object = model.1.2, level = 0.95)
round(data.frame(low = exp(c.val*conf.beta[1:4, 1:2, 3])[,1], up = exp(c.val*conf.beta[1:4, 1:
```

```
##                   low   up
## sugar_standard  0.00 0.61
## fat_standard    0.19 3.16
## sodium_standard 0.00 0.13
```

2. Alcohol, self-esteem and negative relationship interactions (5 points)

Read the example **'Alcohol Consumption'** in chapter 4.2.2 of the textbook. This is based on a study in which moderate-to-heavy drinkers (defined as at least 12 alcoholic drinks/week for women, 15 for men) were recruited to keep a daily record of each drink that they consumed over a 30-day study period. Participants also completed a variety of rating scales covering daily events in their lives and items related to self-esteem. The data are given in the *DeHartSimplified.csv* data set. Questions 24-26 of chapter 3 of the textbook also relate to this data set and give more explanation of its variables.
The researchers stated the following hypothesis: *We hypothesized that negative interactions with romantic partners would be associated with alcohol consumption (and an increased desire to drink). We predicted that people with low trait self-esteem would drink more on days they experienced more negative relationship interactions compared with days during which they experienced fewer negative relationship interactions. The relation between drinking and negative relationship interactions should not be evident for individuals with high trait self-esteem.*

2.1 (2 points): Conduct a thorough EDA of the data set, giving special attention to the relationships relevant to the researchers' hypotheses. You will use this to guide the model specification in the following questions.

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode()
## x dplyr::select() masks MASS::select()
## x purrr::some()   masks car::some()
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
library(gridExtra)
```

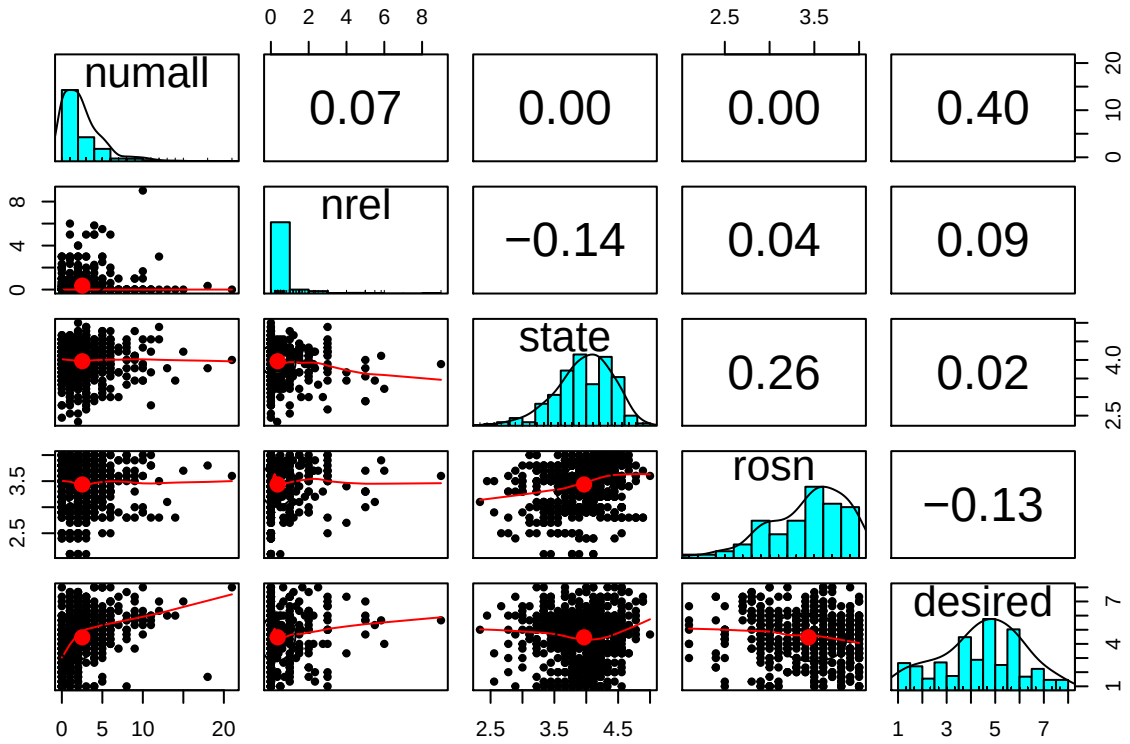
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
d.2 <- read.csv("DeHartSimplified.csv", header=TRUE, sep=",")
describe(d.2)
```

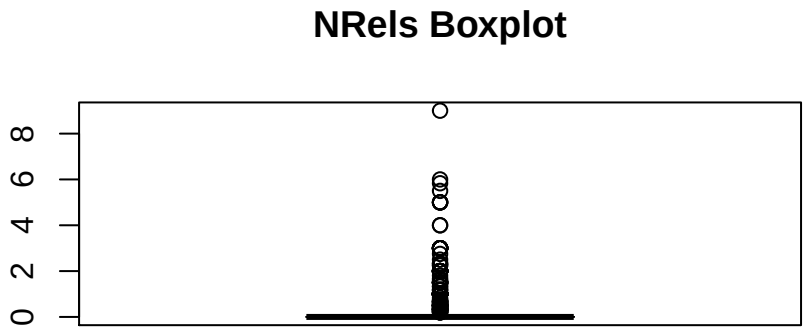
```
##          vars   n  mean    sd median trimmed   mad   min    max  range
## id          1 623 75.89 49.90  60.00   74.85 65.23  1.00 160.00 159.00
## studyday    2 623  4.00  2.00   4.00    4.00  2.97  1.00   7.00   6.00
## dayweek     3 623  4.00  2.00   4.00    4.00  2.97  1.00   7.00   6.00
## numall      4 622  2.52  2.66   2.00    2.09  1.48  0.00  21.00  21.00
## nrel        5 623  0.36  0.94   0.00    0.11  0.00  0.00   9.00   9.00
## prel        6 623  2.58  2.39   2.00    2.28  2.97  0.00   9.00   9.00
## negevent    7 623  0.44  0.39   0.35    0.39  0.35  0.00   2.38   2.38
## posevent    8 623  1.05  0.65   0.95    0.99  0.57  0.00   3.88   3.88
## gender      9 623  1.56  0.50   2.00    1.58  0.00  1.00   2.00   1.00
## rosn       10 623  3.44  0.42   3.50    3.47  0.44  2.10   4.00   1.90
## age        11 623 34.29  4.51  34.57   34.48  5.49 24.43  42.28  17.85
## desired    12 620  4.46  1.69   4.67    4.51  1.48  1.00   8.00   7.00
## state      13 620  3.97  0.44   4.00    3.99  0.49  2.33   5.00   2.67
##            skew kurtosis   se
## id         0.18    -1.47 2.00
## studyday   0.00    -1.26 0.08
## dayweek    0.00    -1.26 0.08
## numall     2.11     7.30 0.11
## nrel       4.03    21.14 0.04
## prel       0.91     0.22 0.10
## negevent   1.30     1.99 0.02
## posevent   0.96     1.18 0.03
## gender    -0.25    -1.94 0.02
## rosn      -0.76     0.14 0.02
## age       -0.26    -0.99 0.18
## desired   -0.24    -0.52 0.07
## state     -0.60     0.38 0.02
```

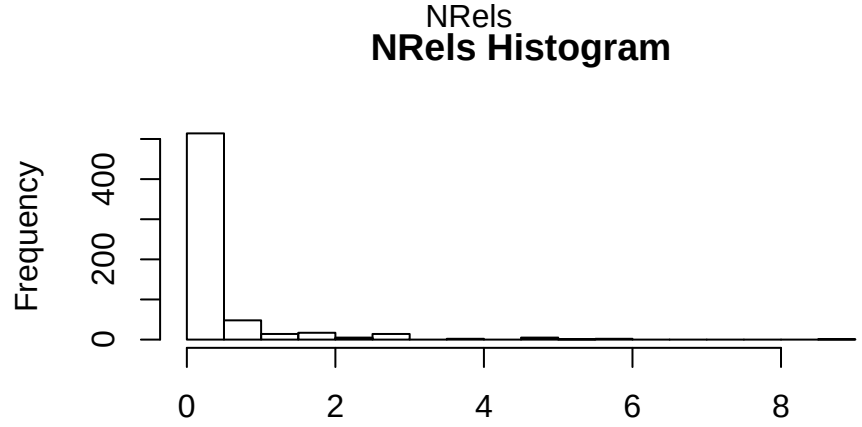
```
subset.d2 <- d.2 %>% select (numall, nrel, state, rosn, desired)
pairs.panels(subset.d2)
```

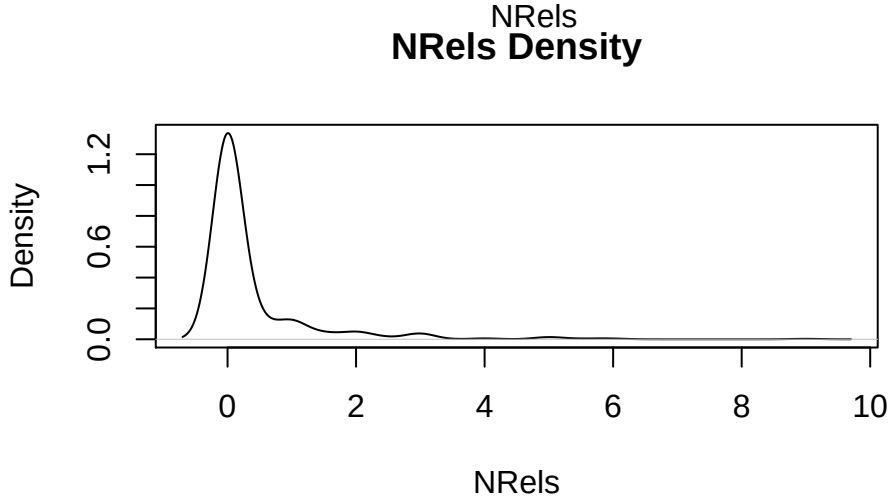


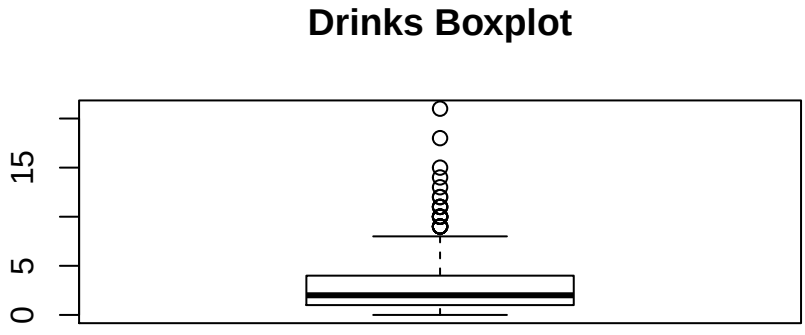
Below, we can see boxplots, histograms, and density plots for all the relevant variables.

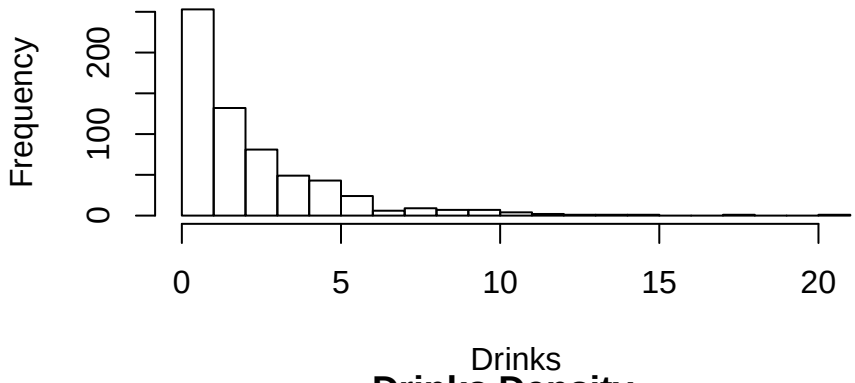
```
# NRel - negative romantic-relationship events
boxplot(d.2$nrel, main= "NRels Boxplot", xlab = "NRels" )
hist(d.2$nrel, breaks=20, main= "NRels Histogram", xlab = "NRels")
plot(density(d.2$nrel), main='NRels Density', xlab='NRels')

# Numall - number of drinks consumed
boxplot(d.2$numall, main= "Drinks Boxplot", xlab = "Drinks" )
hist(d.2$numall, breaks=20, main= "Drinks Histogram", xlab = "Drinks")
plot(density(na.omit(d.2$numall)), main='Drinks Density', xlab='Drinks')

# State - state (short-term) self- esteem
boxplot(d.2$state, main= "State Boxplot", xlab = "State" )
hist(d.2$state, breaks=20, main= "State Histogram", xlab = "State")
plot(density(na.omit(d.2$state)), main='State Density', xlab='State')

# Rosn -  trait (long-term) self-esteem
boxplot(d.2$rosn, main= "rosn Boxplot", xlab = "rosn" )
hist(d.2$rosn, breaks=20, main= "rosn Histogram", xlab = "rosn")
plot(density(d.2$rosn), main='rosn Density', xlab='rosn')

# Desired -  desired drinks
boxplot(d.2$desired, main= "rosn Boxplot", xlab = "rosn" )
hist(d.2$desired, breaks=20, main= "rosn Histogram", xlab = "rosn")
plot(density(na.omit(d.2$desired)), main='rosn Density', xlab='rosn')
```

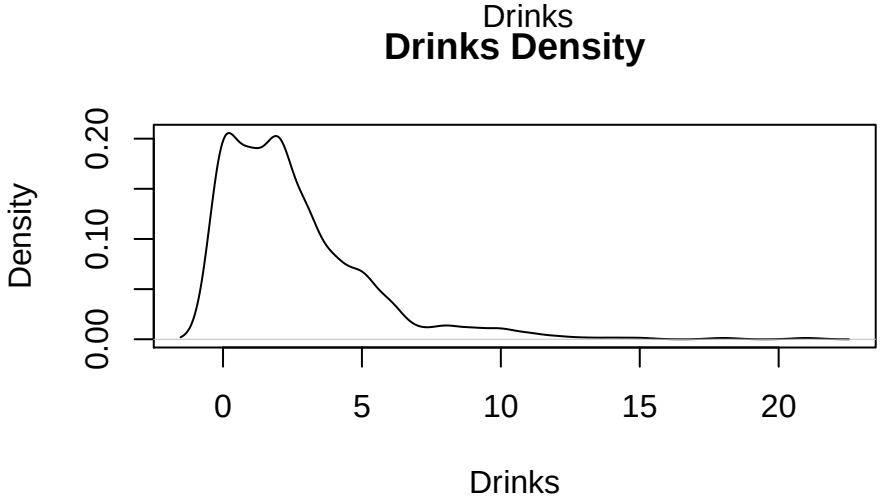

NRels Boxplot
0
2
4
6
8

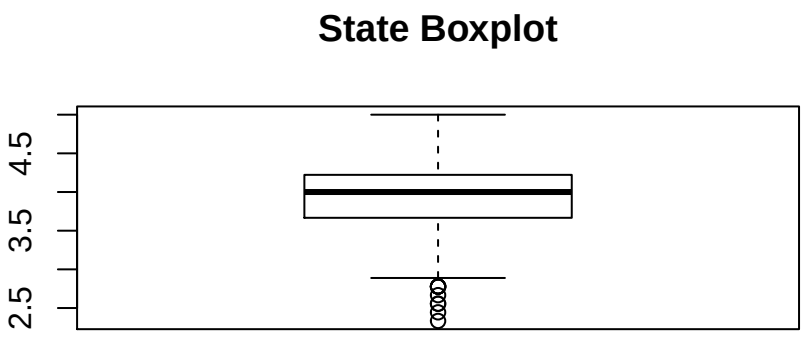

NRels
NRels Histogram
Frequency
0
200
400
0
2
4
6
8

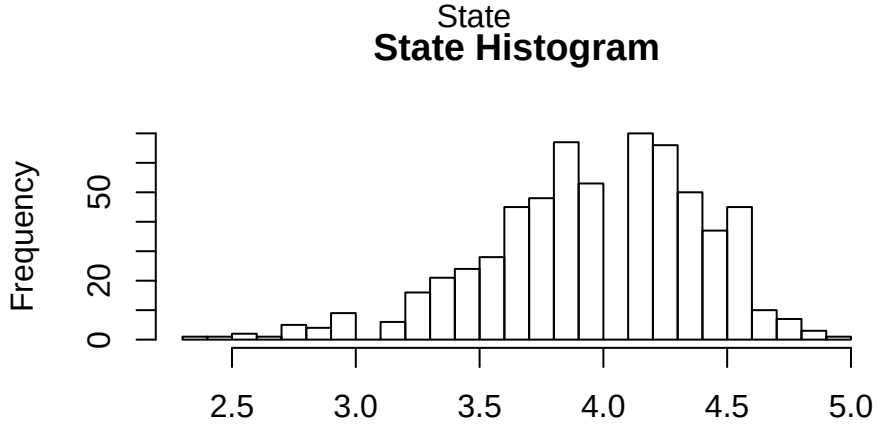

NRels
NRels Density
Density
0.0
0.6
1.2
0
2
4
6
8
10
NRels

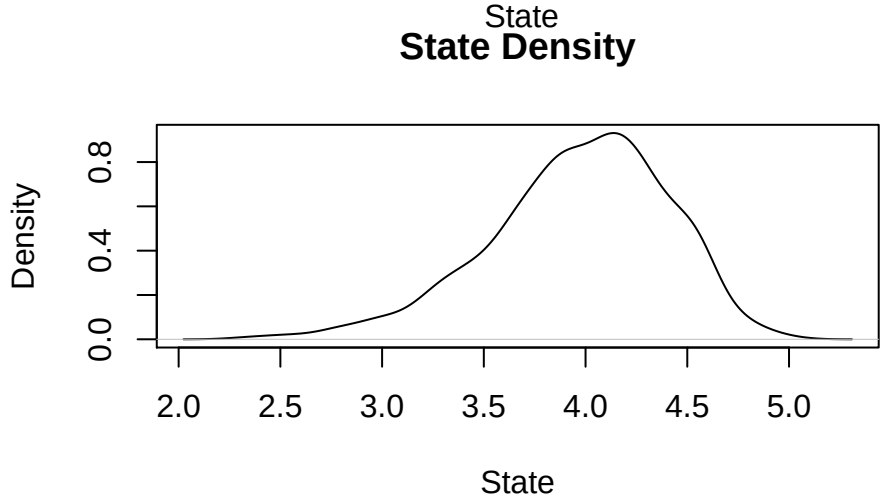

Drinks Boxplot
0
5
15
Drinks

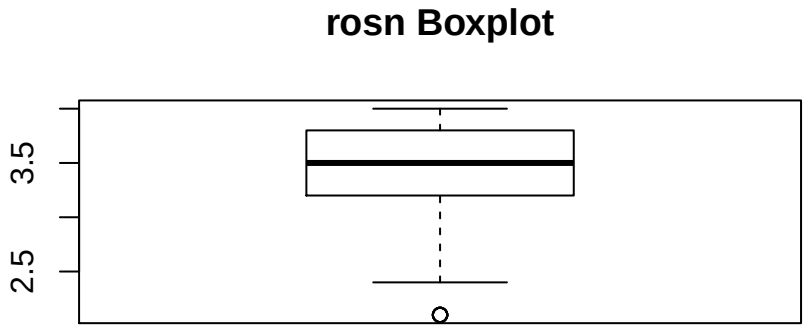

Drinks Histogram
Frequency
0
100
200
0
5
10
15
20
Drinks

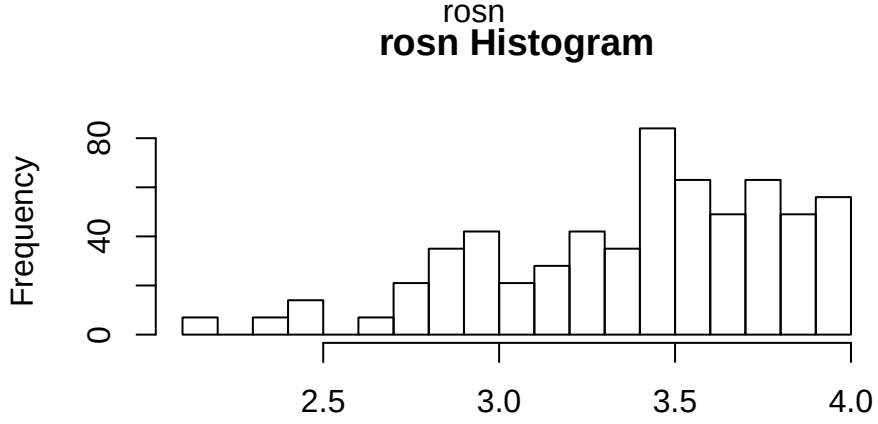

Drinks Density
Density
0.00
0.10
0.20
0
5
10
15
20
Drinks

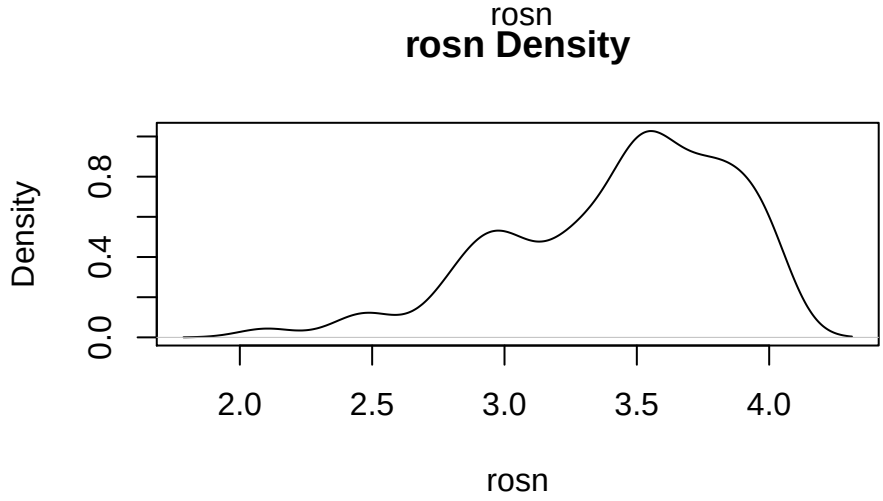

State Boxplot
2.5
3.5
4.5

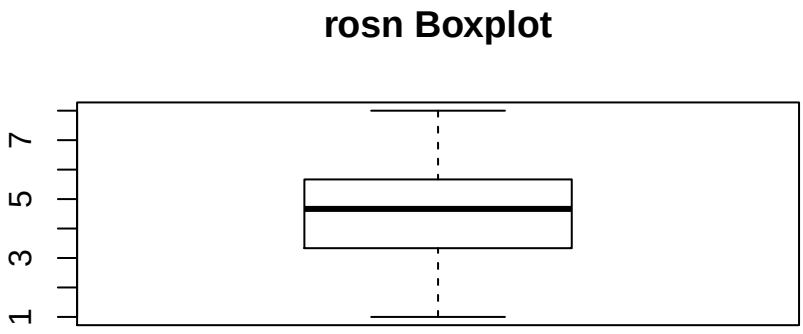

State
State Histogram
Frequency
0
20
50
2.5
3.0
3.5
4.0
4.5
5.0
State

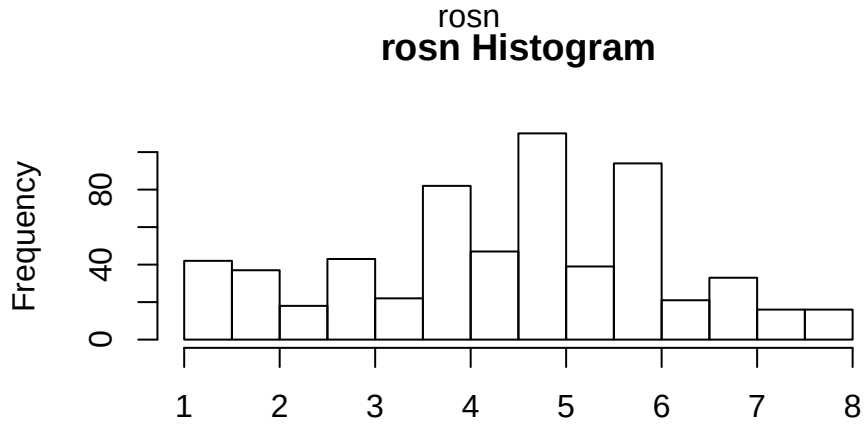

State Density
Density
0.0
0.4
0.8
2.0
2.5
3.0
3.5
4.0
4.5
5.0
State

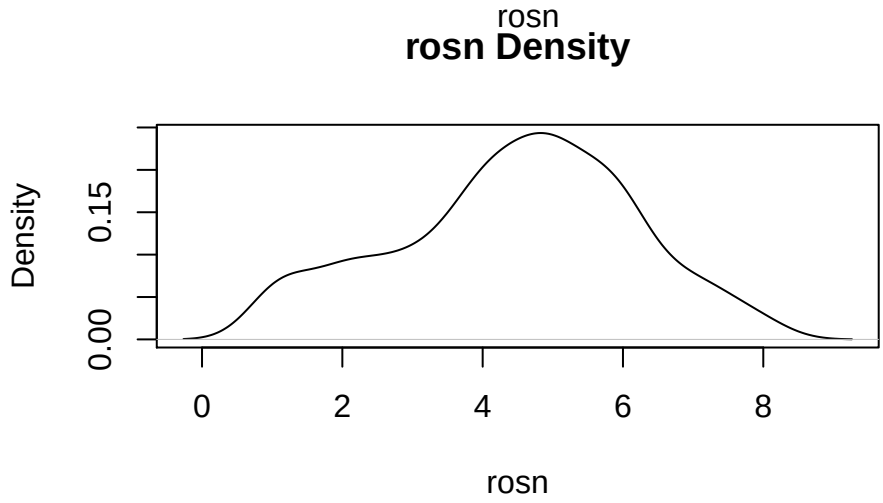

rosn Boxplot
2.5
3.5


rosn
rosn Histogram
Frequency
0
40
80
2.5
3.0
3.5
4.0


rosn
rosn Density
Density
0.0
0.4
0.8
2.0
2.5
3.0
3.5
4.0
rosn

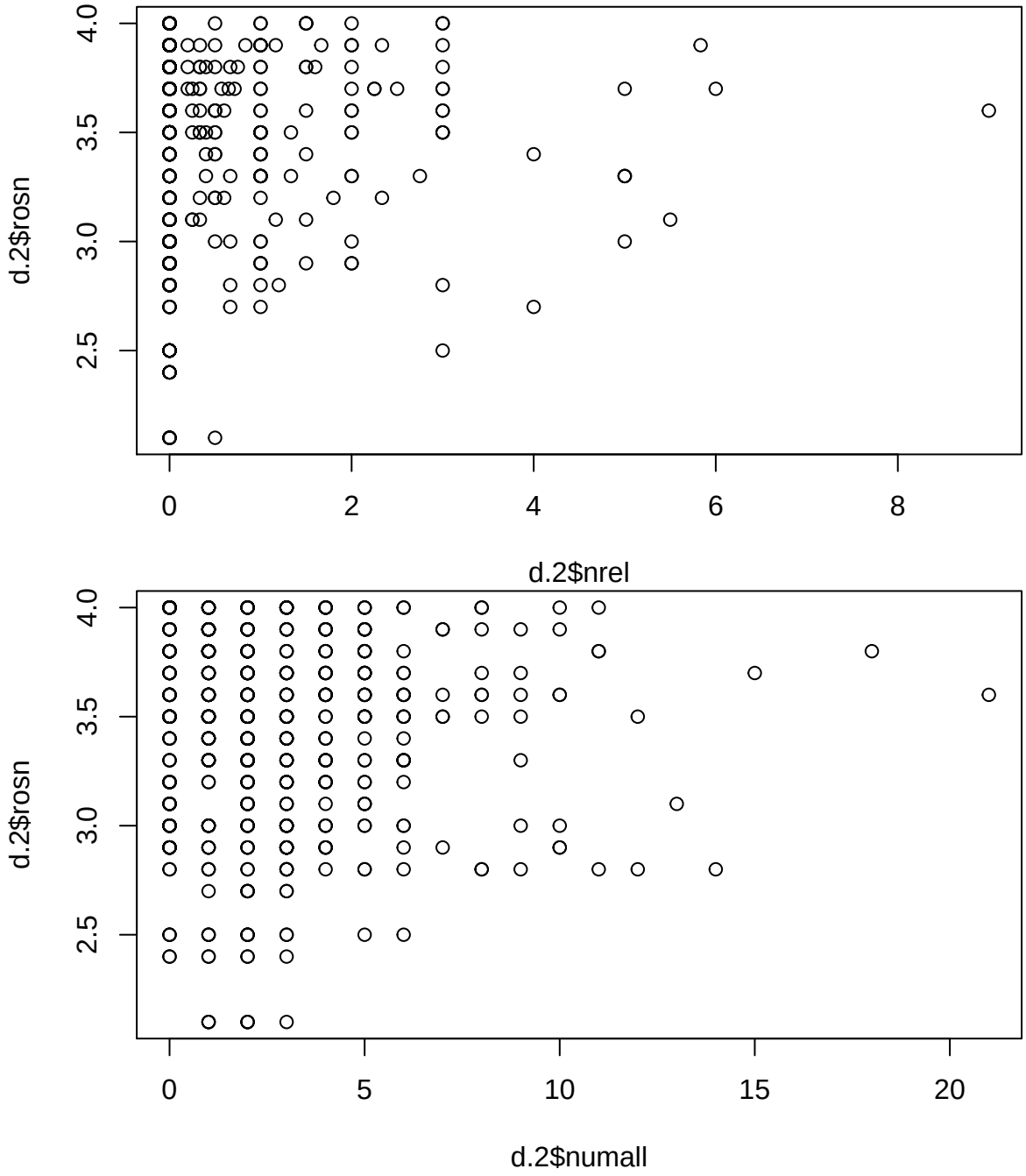


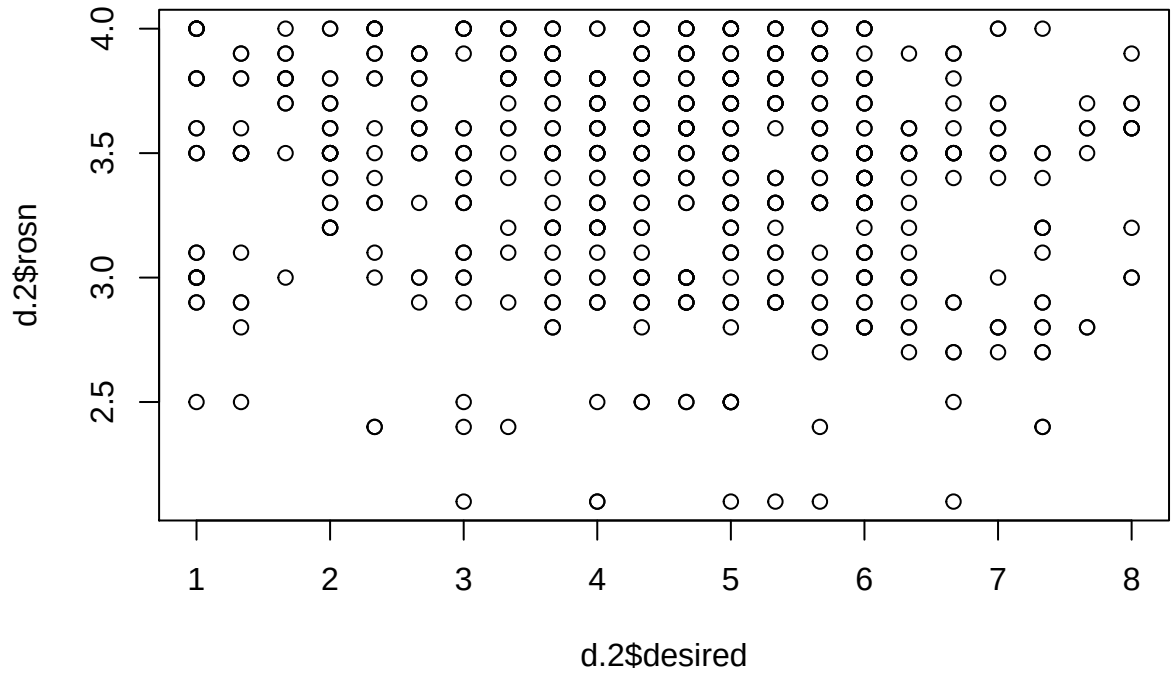




Here, we can do some quick bivariate analysis.

```
plot(d.2$nrel, d.2$rosn)
plot(d.2$numall, d.2$rosn)
plot(d.2$desired, d.2$rosn)
```


d.2$rosn
4.0
3.5
3.0
2.5
0
2
4
6
8
d.2$nrel
d.2$rosn
4.0
3.5
3.0
2.5
0
5
10
15
20
d.2$numall



We can see the difference between desired and actual drinks.

```
library(dplyr)

print(summarise_at(group_by(d.2,dayweek),vars(numall),funs(mean(.,na.rm=TRUE))))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
## # A tibble: 7 x 2
##   dayweek numall
##     <int>  <dbl>
## 1       1   2.01
## 2       2   1.72
## 3       3   1.90
## 4       4   2.46
## 5       5   2.97
## 6       6   4.10
## 7       7   2.51
```

```r
print(summarise_at(group_by(d.2,dayweek),vars(desired),funs(mean(.,na.rm=TRUE))))
```

```
## # A tibble: 7 x 2
##   dayweek desired
##     <int>   <dbl>
## 1       1    4.16
## 2       2    4.33
## 3       3    4.54
## 4       4    4.60
## 5       5    4.82
## 6       6    4.85
## 7       7    3.95
```

Over here, we compare the day of the week for the amount of drinks that were desired vs consumed. The amount desired stayed relatively constant throughout the days, the but the amount consumed spike aggresively on day 6, which is Saturday. Thus, we can run the models below using data without Saturday (which seems like a social day to drink, adding to the noise). We can then see what our results will look like.

2.2 (2 points): Using an appropriate model (or models), evaluate the evidence that negative relationship interactions are associated with higher alcohol consumption and/or an increased desire to drink.

```r
model.2.2 <- glm(numall ~ nrel, family=poisson(link="log"), data=d.2)
summary(model.2.2)
```

```
##
## Call:
## glm(formula = numall ~ nrel, family = poisson(link = "log"),
##     data = d.2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4440  -1.1490  -0.3041   0.3318   7.2776
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.90071    0.02716  33.163  < 2e-16 ***
## nrel         0.06447    0.02365   2.725  0.00642 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1590.3  on 621  degrees of freedom
## Residual deviance: 1583.4  on 620  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 2960.8
##
```

```
## Number of Fisher Scoring iterations: 5
```

```
model.2.2.1 <- glm(desired ~ nrel, data=d.2)
summary(model.2.2.1)
```

```
##
## Call:
## glm(formula = desired ~ nrel, data = d.2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9074  -1.0706   0.2627   1.2627   3.5960
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.40398    0.07259   60.67   <2e-16 ***
## nrel         0.16779    0.07202    2.33   0.0201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.848762)
##
##     Null deviance: 1776.0  on 619  degrees of freedom
## Residual deviance: 1760.5  on 618  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 2412.5
##
## Number of Fisher Scoring iterations: 2
```

The first model is: 0.90071 + 0.06447(nrel). The 2nd model is: 4.40398 + 0.16779(nrel).

At the 1% significant level, it is clear an increase in the negative romatic events leads to an increase in the amount of drinks consumed. At the 5% level, an increase in the negative romatic events leads to an increase in the amount of drinks desired. Both are important metrics to take a look at.

Here, we explore the effect of long term self esteem on the the number of drinks consumed, as well as desired amount of drinks.

```
model.2.2.2 <- glm(numall ~ nrel + rosn + nrel*rosn, family=poisson(link="log"), data=d.2)
summary(model.2.2.2)
```

```
##
## Call:
## glm(formula = numall ~ nrel + rosn + nrel * rosn, family = poisson(link = "log"),
##     data = d.2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5002  -1.1004  -0.3210   0.3894   7.2626
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.77930    0.22380   3.482 0.000497 ***
## nrel         0.46373    0.23216   1.997 0.045774 *
## rosn         0.03535    0.06468   0.547 0.584693
## nrel:rosn   -0.11546    0.06722  -1.718 0.085838 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1590.3  on 621  degrees of freedom
## Residual deviance: 1580.6  on 618  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 2961.9
##
## Number of Fisher Scoring iterations: 5
```

```
model.2.2.3 <- glm(desired ~ nrel + rosn + nrel*rosn, data=d.2)
summary(model.2.2.3)
```

```
##
## Call:
## glm(formula = desired ~ nrel + rosn + nrel * rosn, data = d.2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8840  -1.0335   0.1528   1.2034   3.6779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.0286     0.5863  10.283  < 2e-16 ***
## nrel          0.9269     0.7140   1.298  0.19472
## rosn         -0.4739     0.1697  -2.793  0.00539 **
## nrel:rosn    -0.2159     0.2045  -1.056  0.29151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.80268)
##
##     Null deviance: 1776.0  on 619  degrees of freedom
## Residual deviance: 1726.5  on 616  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 2404.4
##
## Number of Fisher Scoring iterations: 2
```

The first model is: 0.77930 + 0.46373(nrel) + 0.03535(rosn) - 0.11546(nrel:rosn). The 2nd model is: 6.0286 + 0.9269(nrel) - 0.4739(rosn) - .2159(nrel:rosn).

Here, we add some interaction affects. Particularily, we add the interaction between ngeative romantic relations and state. We see that there is signifiance at the 10% level for nrel*rosn. This is the interaction between long term self esteem and negative romantic relatiobships. As long term self esteem increases, the amount of drinks does seem to go down for a negative romantic relationship state.

When looking at desired amount of drinks, long term state (ROSN) is the only significant predictor.

```
Anova(model.2.2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: numall
##      LR Chisq Df Pr(>Chisq)
## nrel   6.8276  1   0.008976 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(model.2.2.1)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: desired
##      LR Chisq Df Pr(>Chisq)
## nrel   5.4276  1    0.01982 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(model.2.2.2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: numall
##           LR Chisq Df Pr(>Chisq)
## nrel        6.8258  1   0.008985 **
## rosn        0.0017  1   0.967556
## nrel:rosn   2.8849  1   0.089412 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(model.2.2.3)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: desired
##           LR Chisq Df Pr(>Chisq)
## nrel        6.1228  1  0.0133448 *
## rosn       11.0468  1  0.0008884 ***
## nrel:rosn   1.1145  1  0.2910995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can confirm this using our LRT test.

2.3 (1 points): Discuss whether the relationship between drinking and negative relationship interactions differs according to individuals' levels of trait self-esteem.

We could use the previous model, but I want to present another way to answer this question using bins of the data. I use the cut function to bin all the data, broken up by levels of self esteem.

```
# plot(d.2$state)
d.2$statebins <- cut(d.2$rosn, 3, labels=c("Low", "Med", "High"), include.lowest=TRUE)
d.2 <- na.omit(d.2, x = "statebins")
d.2 <- na.omit(d.2, x = "state")
d.2 <- na.omit(d.2, x = "nrels")

plot(d.2[d.2$statebins == "High", ]$numall, d.2[d.2$statebins == "High", ]$rosn, xlab = "Index
plot(d.2[d.2$statebins == "Med", ]$numall, d.2[d.2$statebins == "Med", ]$rosn, xlab = "Index",
plot(d.2[d.2$statebins == "Low", ]$numall, d.2[d.2$statebins == "Low", ]$rosn, xlab = "Index",

# d.2
model.2.4 <- glm(numall ~ nrel, family=poisson(link="log"), data=d.2[d.2$statebins == "High",
summary(model.2.4)
```

```
##
## Call:
## glm(formula = numall ~ nrel, family = poisson(link = "log"),
##     data = d.2[d.2$statebins == "High", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3661  -1.1350  -0.2918   0.6781   7.2982
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.89259    0.03435  25.982   <2e-16 ***
## nrel         0.04559    0.03082   1.479    0.139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1050.3  on 393  degrees of freedom
## Residual deviance: 1048.2  on 392  degrees of freedom
## AIC: 1906.2
##
## Number of Fisher Scoring iterations: 5
```

```
model.2.4.2 <- glm(numall ~ nrel, family=poisson(link="log"), data=d.2[d.2$statebins == "Med",
summary(model.2.4.2)
```

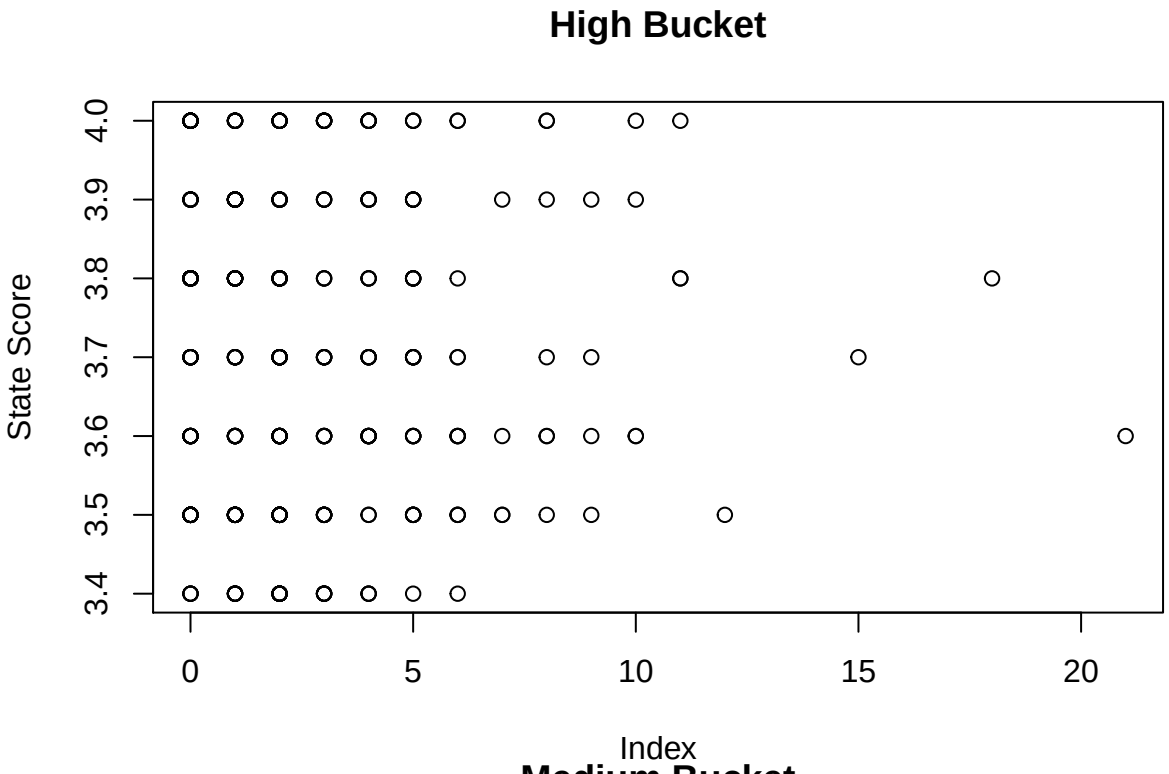
```
##
```

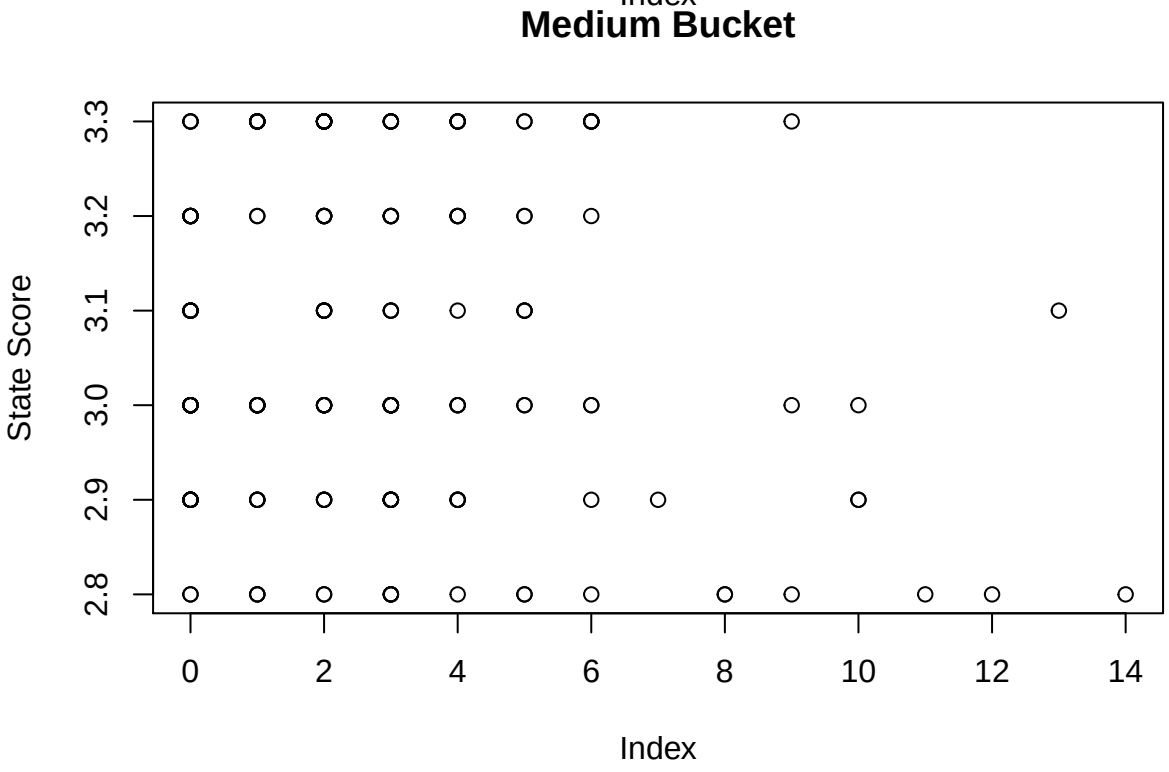
```
## Call:
## glm(formula = numall ~ nrel, family = poisson(link = "log"),
##     data = d.2[d.2$statebins == "Med", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5438  -1.2755  -0.3984   0.7284   4.9179
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.96225    0.04786   20.11   <2e-16 ***
## nrel         0.09083    0.03931    2.31   0.0209 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 486.86  on 188  degrees of freedom
## Residual deviance: 482.04  on 187  degrees of freedom
## AIC: 916.93
##
## Number of Fisher Scoring iterations: 5
```

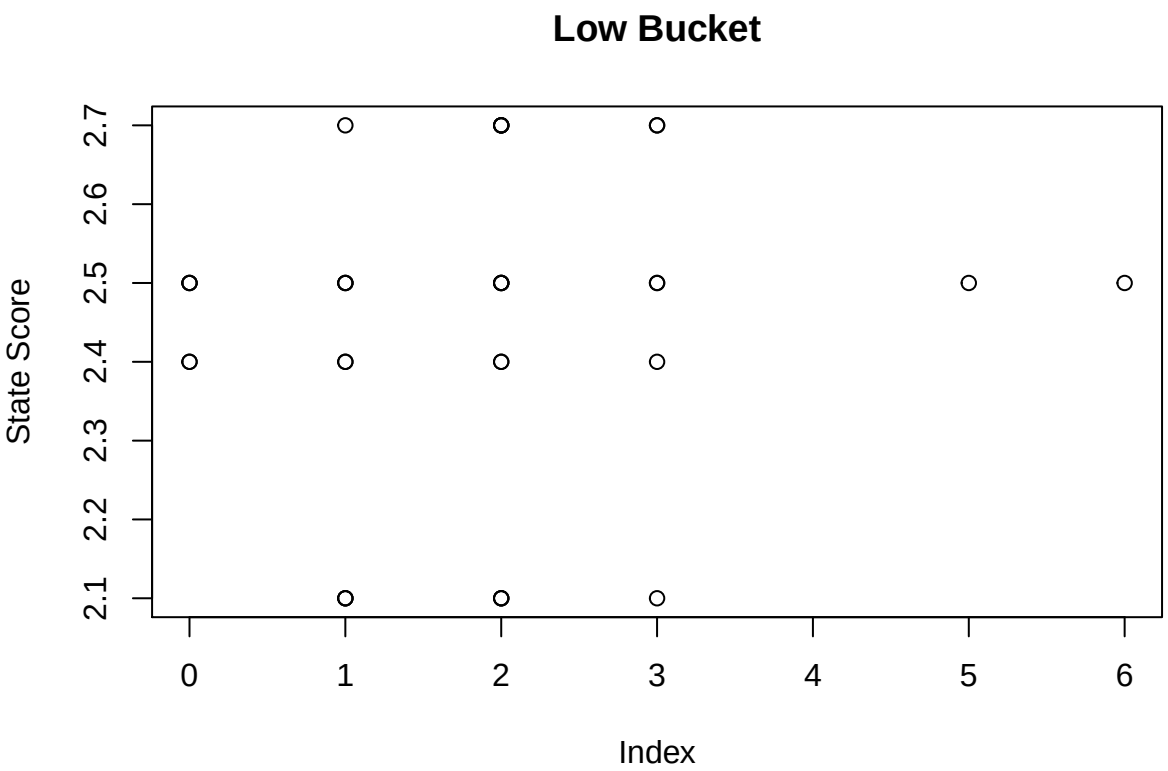
```
model.2.4.3 <- glm(numall ~ nrel, family=poisson(link="log"), data=d.2[d.2$statebins == "Low",
summary(model.2.4.3)
```

```
##
## Call:
## glm(formula = numall ~ nrel, family = poisson(link = "log"),
##     data = d.2[d.2$statebins == "Low", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8817  -0.6312   0.1690   0.1690   2.4875
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5712     0.1328   4.302 1.69e-05 ***
## nrel          0.1066     0.1282   0.832    0.406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 35.937  on 34  degrees of freedom
## Residual deviance: 35.311  on 33  degrees of freedom
## AIC: 116.38
##
```

```
## Number of Fisher Scoring iterations: 5
```







The first model is: 0.89259 + 0.04559(nrel). The 2nd model is: 0.96225 + 0.09083(nrel). The 3rd model is: 0.5712 + 0.1066(nrel).

I find that for low and high buckets for self esteem, a negative romantic relation has no significant effect on the amount one drinks. However, for the medium bucket, nrel is a signifcant predictor at the 5% level. This could be interpreted as those with high self esteems/low self esteems don't have romantic relationships affect how much they drink. However, the average indivdual does seem to be affected by a negative event in the relationship.