Regression Project Proposal

Statistical Learning for Datamining

Oje Williams

**(a) Describe the response variable and the predictors. How was the data collected?**

The data for this proposal was collected from:

<https://www.kaggle.com/miroslavsabo/young-people-survey/data>

It was based on a 2013 survey of statistics students of Slovakian nationality, aged between 15-30 at FSEV UK.

* The data file (responses.csv) consists of 1010 rows and 150 columns (139  
  integer and 11 categorical).
* For convenience, the original variable names were shortened in the  
  data file. See the columns.csv file if you want to match the data with the original names.
* The data contain missing values.
* The survey was presented to participants in both electronic and written form.

The variables can be split into the following groups:

* Music preferences (19 items)
* Movie preferences (12 items)
* Hobbies & interests (32 items)
* Phobias (10 items)
* Health habits (3 items)
* Personality traits, views on life, & opinions (57 items)
* Spending habits (7 items)
* Demographics (10 items)

I am interested in seeing if we can predict height based on movie/music preferences, fears.

**(b) Impute missing data-points with their mean. What is n and p?**

The data contained a few missing values and I imputed the missing values with the mean by using the following code

1. for(i in 1:ncol(tempData1)){
2. tempData1[is.na(tempData1[,i]), i] <- round(mean(tempData1[,i], na.rm = TRUE))

Before this, I removed some of the categorical variables and recoded the remaining ones. Here is an example

1. yps<- read.csv("C://Users//ojewi//Desktop//R.workspace//Datamining//RRresponses.csv")
2. tempData <- yps
3. tempData1 <- select(tempData,-c(Internet.usage,Only.child,House...block.of.flats,Lying,Education))
4. map.smoke <- c( "never smoked" = 1, "tried smoking" = 2,"former smoker" = 3, "current smoker" = 4, na.rm = TRUE)
5. tempData1$Smoking <- map.smoke[tempData1$Smoking]
6. glimpse(tempData1)

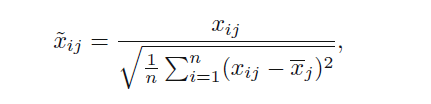
 n= 1010

p = 52

1. Observations: 1,010
2. Variables: 52
3. $ Height <dbl> 163, 163, 176, 172, 170, 186, 177, 184, 166, 174, 175...
4. $ Weight <dbl> 48, 58, 67, 59, 59, 77, 50, 90, 55, 60, 60, 60, 55, 5...
5. $ Music <dbl> 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 1, 5, 5,...
6. $ Slow.songs.or.fast.songs <dbl> 3, 4, 5, 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,...
7. $ Dance <dbl> 2, 2, 2, 2, 4, 2, 5, 3, 3, 2, 3, 1, 1, 5, 2, 2, 3, 3,...
8. $ Folk <dbl> 1, 1, 2, 1, 3, 3, 3, 2, 1, 5, 2, 1, 2, 3, 1, 2, 1, 3,...
9. $ Country <dbl> 2, 1, 3, 1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 3, 1, 3,...
10. $ Classical.music <dbl> 2, 1, 4, 1, 4, 3, 2, 2, 2, 2, 2, 4, 4, 1, 2, 4, 1, 2,...
11. $ Musical <dbl> 1, 2, 5, 1, 3, 3, 2, 2, 4, 5, 3, 1, 3, 5, 3, 3, 2, 2,...
12. $ Pop <dbl> 5, 3, 3, 2, 5, 2, 5, 4, 3, 3, 4, 2, 3, 5, 4, 3, 4, 4,...

**(c) Standardize the numerical predictors using equation (6.6) in the ISLR book.**

Based on the equation provided in the ISLR Textbook to standardize predictors, the following code would have been used, but since the data is mainly categorical there is little need to do so.



1. stdize <- function(x, na.rm = FALSE) (x / sd(x, na.rm)) # Stardardize function
2. std\_data <- tempData1 %>% select(4:145) %>%mutate\_all(stdize) %>%
3. mutate(Gender=tempData1$Gender,Handedness=tempData1$Left...right.handed,Residence=tempData1$Village...town)
4. std\_data <- std\_data[c(143,144,145,1:142)] # reorder columns putting target variables first
5. glimpse(std\_data)
6. Observations: 1,010
7. Variables: 145
8. $ Gender <dbl> 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2, ...
9. $ Handedness <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, ...
10. $ Residence <dbl> 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, ...
11. $ Music <dbl> 7.538957, 6.031166, 7.538957, 7.538957, 7.538957, 7.538957, 7.538957,...
12. $ Slow.songs.or.fast.songs <dbl> 3.600439, 4.800585, 6.000731, 3.600439, 3.600439, 3.600439, 6.000731,...
13. $ Dance <dbl> 1.7119366, 1.7119366, 1.7119366, 1.7119366, 3.4238733, 1.7119366, 4.2...
14. $ Folk <dbl> 0.8800714, 0.8800714, 1.7601429, 0.8800714, 2.6402143, 2.6402143, 2.6...
15. $ Country <dbl> 1.8630628, 0.9315314, 2.7945943, 0.9315314, 1.8630628, 1.8630628, 0.9...
16. $ Classical.music <dbl> 1.6022778, 0.8011389, 3.2045556, 0.8011389, 3.2045556, 2.4034167, 1.6...
17. $ Musical <dbl> 0.7938783, 1.5877567, 3.9693916, 0.7938783, 2.3816350, 2.3816350, 1.5...
18. $ Pop <dbl> 4.310506, 2.586304, 2.586304, 1.724202, 4.310506, 1.724202, 4.310506,...
19. $ Rock <dbl> 4.2320005, 4.2320005, 4.2320005, 1.6928002, 2.5392003, 4.2320005, 2.5...
20. $ Metal.or.Hardrock <dbl> 0.729345, 2.917380, 2.188035, 0.729345, 0.729345, 3.646725, 0.729345,...
21. $ Punk <dbl> 0.7712668, 3.0850673, 3.0850673, 3.0850673, 1.5425336, 2.3138004, 0.7...
22. $ Hiphop..Rap <dbl> 0.7283541, 0.7283541, 0.7283541, 1.4567083, 3.6417707, 2.9134165, 2.1...