SBS 2000ID QUANTITATIVE RESEARCH PROJECT

LE' SEAN ROBERTS

FACTORS INFLUENCING EMPLOYMENT OUTCOMES FOR ADOLESCENCE: AN ANALYSIS USING ORDINAL AND CATEGORICAL VARIABLES - DATA WRANGLING AND DATA ANALYSIS FOR QUANTITATIVE RESEARCH IN THE SOCIAL AND BEHAVIORAL SCIENCES CONCERNING ADOLESCENCE

A research project that investigates employment with adolescence as the responsible variable with ordinal and categorical variables as predictors can provide valuable insights into the factors that influence employment outcomes. Project concerns comprehension of factors that affect employment status for the adolescent populous, and to build a predictive model for employment using a combination of ordinal and categorical predictor variables.

This project involves application the Inter-University Consortium for Political and Social Research (ICPSR), particularly the ICPSR 38503 Survey: Monitoring the Future: A Continuing Study of American Youth (12th-Grade Survey), 2021. An ICPSR codebook is used to identify variables of interest. The chosen topic for this project as the target or dependent variable is employment. A column or variable that identifies well with the notion of adolescent employment is "V2191 Work?", being an ordinal variable whose values range from 1 to 6 based on the average over a school year the amount of hours per week labored in a paid or unpaid job. Concerning the choice of features or independent variables, feature importance/selection methods such as correlation measure and the Boruta Algorithm are applied; a means to avoid the incorporation of cognitive bias. The number of features or independent variables chosen is 6 for computational power convenience. Alternatively, also applying entropy-based feature selection to compare such predictor selection with feature (predictor) importance from the Boruta Algorithm.
Much of the data from the ICPSR 38503 Survey is constructed into ordinal and categorical variables.
To acquire a wholesome view of the significance of all predictors based on the Boruta Algorithm and entropy method for predictor selection, the ordinal predictors are transformed into

categorical predictors based on the coding given by the ICPSR 38503 Survey codebook. The R environment economically serves well for all mentioned pursuits.

Commencing with data probing and wrangling.

```
library(readxl)
Youth_Data_Short_version_1_ <-
  read_excel("C:/Users/verlene/OneDrive/Desktop/CITY TECH FALL 2023/Courses/SBS 2000ID/HOM
head(Youth_Data_Short_version_1_)
```

```
# A tibble: 6 x 201
  RESPONDENT_ID    V1    V3 SURVEY_VERSION  V545  V548 RANDOM_GROUP RANDOM_TEST
          <dbl> <dbl> <dbl>          <dbl> <dbl> <dbl>        <dbl>       <dbl>
1            NA    NA    NA             NA    NA    NA           NA          NA
2         10006  2021     1              4     3     5            1           0
3         10019  2021     1              4     3     5            1           0
4         10023  2021     1              4     3     5            1           0
5         10030  2021     1              4     3     4            2           0
6         10035  2021     1              4     3     4            2           0
# i 193 more variables: ARCHIVE_WT <dbl>, V13 <dbl>, V16 <dbl>, V17 <dbl>,
#   RESPONDENT_AGE <dbl>, `2150 Sex` <dbl>, `2151 Race` <dbl>,
#   `V2152 Where` <dbl>, `V2153 Married` <dbl>, `V2155 Dad?` <dbl>,
#   `V2156 Mom?` <dbl>, `V2157 Siblings` <dbl>, `V49 # of Sibs` <dbl>,
#   `V2163 Dad Educ` <dbl>, `V2164 Mom Educ` <dbl>, `V2165 Mom Work` <dbl>,
#   `V2166 politics` <dbl>, `V2167 Beliefs` <dbl>, `V2169 religion` <dbl>,
#   V2170 <dbl>, V2171 <dbl>, V2172 <dbl>, V2173 <dbl>, ...
```

Data Probing and Summary Statistics

```
# Observing the dimension of the dataframe.
dim(Youth_Data_Short_version_1_)
```

```
[1] 2050  201
```

Observed variables count isn't consistent with direct Excel display. Hence to investigate or probe.

```
str(Youth_Data_Short_version_1_)
```

```
tibble [2,050 x 201] (S3: tbl_df/tbl/data.frame)
 $ RESPONDENT_ID      : num [1:2050] NA 10006 10019 10023 10030 ...
 $ V1                 : num [1:2050] NA 2021 2021 2021 2021 ...
 $ V3                 : num [1:2050] NA 1 1 1 1 1 1 1 1 1 ...
 $ SURVEY_VERSION     : num [1:2050] NA 4 4 4 4 4 4 3 3 3 ...
 $ V545               : num [1:2050] NA 3 3 3 3 3 3 3 3 3 ...
 $ V548               : num [1:2050] NA 5 5 5 4 4 4 4 4 4 ...
 $ RANDOM_GROUP       : num [1:2050] NA 1 1 1 2 2 2 1 1 1 ...
 $ RANDOM_TEST        : num [1:2050] NA 0 0 0 0 0 0 0 0 0 ...
 $ ARCHIVE_WT         : num [1:2050] NA 1.81 1.7 4.73 1.31 ...
 $ V13                : num [1:2050] NA 3 3 4 4 4 4 4 4 4 ...
 $ V16                : num [1:2050] NA 0 0 0 0 0 0 1 1 1 ...
 $ V17                : num [1:2050] NA 1 1 1 0 0 0 1 1 1 ...
 $ RESPONDENT_AGE     : num [1:2050] NA 2 1 2 2 2 2 2 2 1 ...
 $ 2150 Sex           : num [1:2050] NA 2 2 2 1 2 1 3 1 1 ...
 $ 2151 Race          : num [1:2050] NA 3 2 2 -9 3 3 -9 3 3 ...
 $ V2152 Where        : num [1:2050] NA 6 0 8 1 0 5 6 6 9 ...
 $ V2153 Married      : num [1:2050] NA 4 2 4 4 2 3 4 4 4 ...
 $ V2155 Dad?         : num [1:2050] NA 0 0 0 1 0 0 -9 0 1 ...
 $ V2156 Mom?         : num [1:2050] NA 0 0 0 1 0 0 -9 1 1 ...
 $ V2157 Siblings     : num [1:2050] NA 1 0 0 0 0 0 -9 0 1 ...
 $ V49 # of Sibs      : num [1:2050] NA 3 3 1 3 3 3 3 0 1 ...
 $ V2163 Dad Educ     : num [1:2050] NA 7 3 5 1 4 5 4 4 4 ...
 $ V2164 Mom Educ     : num [1:2050] NA 7 5 5 7 4 7 4 3 5 ...
 $ V2165 Mom Work     : num [1:2050] NA 1 3 4 4 2 4 3 4 4 ...
 $ V2166 politics     : num [1:2050] NA 7 5 4 1 5 4 6 5 7 ...
 $ V2167 Beliefs      : num [1:2050] NA 8 3 5 1 8 5 4 8 5 ...
 $ V2169 religion     : num [1:2050] NA 1 1 -9 -9 -9 -9 -9 -9 -9 ...
 $ V2170              : num [1:2050] NA 4 3 -9 -9 -9 -9 -9 -9 -9 ...
 $ V2171              : num [1:2050] NA 2 1 1 1 6 6 1 1 1 ...
 $ V2172              : num [1:2050] NA 4 1 4 4 4 4 2 2 2 ...
 $ V2173              : num [1:2050] NA 7 6 7 7 5 7 6 6 5 ...
 $ V2174 Smart?       : num [1:2050] NA 7 7 7 7 7 7 7 7 7 ...
 $ V2174CatSmartCat   : num [1:2050] NA 3 3 3 3 3 3 3 3 3 ...
 $ V2175 School sick  : num [1:2050] NA 1 3 1 1 6 1 1 1 1 ...
 $ V2176 school cut   : num [1:2050] NA 1 1 1 1 2 1 1 1 1 ...
 $ V2177              : num [1:2050] NA 1 1 1 1 1 1 1 2 1 ...
 $ V2178              : num [1:2050] NA 1 2 1 1 1 4 1 2 2 ...
 $ V1730 Sleep        : num [1:2050] -9 1 1 1 1 -9 1 1 1 1 ...
 $ V2179 GPA          : num [1:2050] NA 9 8 9 5 4 1 5 6 7 ...
 $ V2180              : num [1:2050] NA 2 3 1 4 1 1 2 2 1 ...
 $ V2181              : num [1:2050] NA 2 4 1 -9 2 -9 3 1 1 ...
 $ V2182              : num [1:2050] NA 2 4 1 -9 3 -9 3 1 1 ...
```

```
$ V2183  College?    : num [1:2050] NA 2 4 1 -9 2 -9 3 1 4 ...
$ V2184 Grad school? : num [1:2050] NA 2 -9 1 -9 1 -9 2 1 4 ...
$ V2185              : num [1:2050] NA 0 1 1 1 0 0 0 1 0 ...
$ V2186              : num [1:2050] NA 0 1 1 0 1 0 1 0 0 ...
$ V2187              : num [1:2050] NA 1 1 1 0 1 0 1 0 0 ...
$ V2188              : num [1:2050] NA 0 1 1 0 1 0 1 0 0 ...
$ V2189              : num [1:2050] NA 0 0 0 0 0 1 0 0 0 ...
$ V2190              : num [1:2050] NA 0 0 0 0 0 0 0 0 1 ...
$ V2191 Work?        : num [1:2050] NA 6 8 6 3 2 8 2 1 1 ...
$ V2192 Money?       : num [1:2050] NA 1 10 1 4 1 1 1 1 1 ...
$ V1633 Happy?       : num [1:2050] 2 2 2 2 2 2 2 2 1 2 ...
$ V2194   GO Out     : num [1:2050] NA 6 1 4 4 2 4 1 2 3 ...
$ V2195 On dates?    : num [1:2050] NA 4 2 6 3 2 3 1 3 4 ...
$ V2196              : num [1:2050] NA 3 6 5 5 2 4 3 3 3 ...
$ V2197              : num [1:2050] NA 0 2 3 0 0 3 0 1 0 ...
$ V2198              : num [1:2050] NA -9 0 0 -9 -9 0 -9 0 -9 ...
$ V2199              : num [1:2050] NA -9 0 1 -9 -9 0 -9 0 -9 ...
$ V2200              : num [1:2050] NA -9 0 2 -9 -9 0 -9 0 -9 ...
$ V2201              : num [1:2050] NA 0 4 4 0 0 4 0 0 0 ...
$ V2202              : num [1:2050] NA -9 0 0 -9 -9 0 -9 -9 -9 ...
$ V2203              : num [1:2050] NA -9 0 1 -9 -9 0 -9 -9 -9 ...
$ V2204              : num [1:2050] NA -9 0 2 -9 -9 0 -9 -9 -9 ...
$ V2205              : num [1:2050] NA -9 3 -9 -9 -9 -9 -9 -9 -9 ...
$ V2206              : num [1:2050] NA -9 2 -9 -9 -9 -9 -9 -9 -9 ...
$ V2207              : num [1:2050] NA -9 3 -9 -9 -9 -9 -9 -9 -9 ...
$ V7899              : num [1:2050] NA 2 2 1 1 1 1 2 2 1 ...
$ V7900              : num [1:2050] NA -9 -9 2 1 2 2 -9 -9 2 ...
$ V7901              : num [1:2050] NA -9 -9 -9 1 -9 -9 -9 -9 -9 ...
$ V7902              : num [1:2050] NA 3 3 1 2 3 2 1 3 1 ...
$ V7903              : num [1:2050] NA 0 0 0 0 1 0 0 0 0 ...
$ V7904              : num [1:2050] NA 0 0 0 0 0 0 0 1 1 ...
$ V7905              : num [1:2050] NA 0 1 0 0 1 0 0 0 1 ...
$ V7906              : num [1:2050] NA 1 0 1 1 0 1 1 0 0 ...
$ V7907              : num [1:2050] NA -9 2 -9 -9 2 -9 -9 1 1 ...
$ V7908              : num [1:2050] NA -9 2 1 3 1 2 1 1 1 ...
$ V7909              : num [1:2050] NA -9 2 1 4 3 1 2 3 2 ...
$ V7910              : num [1:2050] NA -9 4 3 3 3 3 1 1 1 ...
$ V7911              : num [1:2050] NA -9 2 5 1 2 1 2 2 1 ...
$ V7912              : num [1:2050] NA -9 1 5 1 2 1 5 1 3 ...
$ V7913              : num [1:2050] NA -9 2 5 1 2 1 1 1 1 ...
$ V7914              : num [1:2050] NA -9 5 5 3 5 1 3 3 5 ...
$ V7915              : num [1:2050] NA -9 5 5 3 5 1 3 3 3 ...
$ V7916              : num [1:2050] NA -9 5 5 3 5 1 3 5 5 ...
```

```
$ V7917              : num [1:2050] NA -9 5 5 3 5 1 5 5 5 ...
$ V7918              : num [1:2050] NA -9 5 5 3 5 1 3 3 4 ...
$ V7919              : num [1:2050] NA -9 5 5 3 5 1 3 4 4 ...
$ V7920              : num [1:2050] NA -9 5 5 3 5 1 3 3 4 ...
$ V7921              : num [1:2050] NA -9 5 5 3 5 1 4 3 4 ...
$ V7922              : num [1:2050] NA -9 5 5 3 5 1 3 3 1 ...
$ V7923              : num [1:2050] NA -9 5 5 3 3 1 3 3 3 ...
$ V7924              : num [1:2050] NA -9 5 5 3 5 1 3 4 3 ...
$ V2101 ever smoke   : num [1:2050] NA 2 4 1 5 1 3 2 1 1 ...
$ V2102 Smoke?       : num [1:2050] NA 5 3 1 7 1 2 1 1 1 ...
$ V2547              : num [1:2050] NA -9 -9 -9 -9 -9 -9 -9 -9 -9 ...
$ V2548              : num [1:2050] NA -9 -9 -9 -9 -9 -9 -9 -9 -9 ...
$ V2549              : num [1:2050] NA -9 -9 -9 -9 -9 -9 -9 -9 -9 ...
$ V2564              : num [1:2050] NA -9 -9 -9 -9 -9 -9 -9 -9 -9 ...
 [list output truncated]
```

Acquiring variables of interest. In the codebook the applied value of -9 in the variables is used to designate missing values; such values to be removed.

```r
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.3.2
```

```r
Research_Variables_of_interest <- Youth_Data_Short_version_1_ |>
  dplyr::select(RESPONDENT_AGE, `2150 Sex`, `2151 Race`, `V2152 Where`,
         `V2153 Married`, `V2155 Dad?`, `V2156 Mom?`, `V2157 Siblings`,
         `V2163 Dad Educ`, `V2164 Mom Educ`, `V2165 Mom Work`,
         `V2174 Smart?`, `V2179 GPA`, `V2183  College?`, `V2191 Work?`,
         `V2192 Money?`, `V2194   GO Out`, `V2195 On dates?`,
         `V2105 Drink alcohol`, `V2116 smoke grass?`) |>
  na.omit() |>
  dplyr::filter(RESPONDENT_AGE != -9, `2150 Sex` != -9, `2151 Race` != -9,
         `V2152 Where` != -9, `V2153 Married` != -9, `V2155 Dad?` != -9,
         `V2156 Mom?` != -9, `V2157 Siblings` != -9, `V2163 Dad Educ` != -9,
         `V2164 Mom Educ` != -9, `V2165 Mom Work` != -9,
         `V2174 Smart?` != -9, `V2179 GPA` != -9, `V2183  College?` != -9,
         `V2191 Work?` != -9, `V2192 Money?` != -9, `V2194   GO Out` != -9,
         `V2195 On dates?` != -9,
         `V2105 Drink alcohol` != -9, `V2116 smoke grass?` != -9)
```

```r
head(Research_Variables_of_interest)
```

```
# A tibble: 6 x 20
  RESPONDENT_AGE `2150 Sex` `2151 Race` `V2152 Where` `V2153 Married`
           <dbl>      <dbl>       <dbl>         <dbl>           <dbl>
1              1          2           2             0               2
2              2          2           2             8               4
3              2          2           3             0               2
4              2          1           3             6               4
5              1          1           3             9               4
6              2          1           2             4               4
# i 15 more variables: `V2155 Dad?` <dbl>, `V2156 Mom?` <dbl>,
#   `V2157 Siblings` <dbl>, `V2163 Dad Educ` <dbl>, `V2164 Mom Educ` <dbl>,
#   `V2165 Mom Work` <dbl>, `V2174 Smart?` <dbl>, `V2179 GPA` <dbl>,
#   `V2183  College?` <dbl>, `V2191 Work?` <dbl>, `V2192 Money?` <dbl>,
#   `V2194   GO Out` <dbl>, `V2195 On dates?` <dbl>,
#   `V2105 Drink alcohol` <dbl>, `V2116 smoke grass?` <dbl>
```

```r
str(Research_Variables_of_interest)
```

```
tibble [1,378 x 20] (S3: tbl_df/tbl/data.frame)
 $ RESPONDENT_AGE     : num [1:1378] 1 2 2 2 1 2 2 1 2 2 ...
 $ 2150 Sex           : num [1:1378] 2 2 2 1 1 1 2 2 2 2 ...
 $ 2151 Race          : num [1:1378] 2 2 3 3 3 2 2 3 2 2 ...
 $ V2152 Where        : num [1:1378] 0 8 0 6 9 4 8 8 5 3 ...
 $ V2153 Married      : num [1:1378] 2 4 2 4 4 4 4 4 4 4 ...
 $ V2155 Dad?         : num [1:1378] 0 0 0 0 1 1 1 0 0 1 ...
 $ V2156 Mom?         : num [1:1378] 0 0 0 1 1 1 1 1 1 1 ...
 $ V2157 Siblings     : num [1:1378] 0 0 0 0 1 1 1 1 1 1 ...
 $ V2163 Dad Educ     : num [1:1378] 3 5 4 4 4 6 6 6 3 5 ...
 $ V2164 Mom Educ     : num [1:1378] 5 5 4 3 5 6 5 4 5 5 ...
 $ V2165 Mom Work     : num [1:1378] 3 4 2 4 4 2 3 4 4 1 ...
 $ V2174 Smart?       : num [1:1378] 7 7 7 7 7 7 7 7 7 7 ...
 $ V2179 GPA          : num [1:1378] 8 9 4 6 7 8 9 9 8 9 ...
 $ V2183  College?    : num [1:1378] 4 1 2 1 4 4 4 4 3 4 ...
 $ V2191 Work?        : num [1:1378] 8 6 2 1 1 3 2 8 6 3 ...
 $ V2192 Money?       : num [1:1378] 10 1 1 1 1 5 1 10 10 8 ...
 $ V2194   GO Out     : num [1:1378] 1 4 2 2 3 3 5 3 5 3 ...
 $ V2195 On dates?    : num [1:1378] 2 6 2 3 4 1 4 2 5 2 ...
 $ V2105 Drink alcohol: num [1:1378] 4 4 2 1 2 1 6 3 4 1 ...
```

```
 $ V2116 smoke grass? : num [1:1378] 1 2 2 1 1 1 7 1 7 1 ...
 - attr(*, "na.action")= 'omit' Named int [1:3] 1 2049 2050
  ..- attr(*, "names")= chr [1:3] "1" "2049" "2050"
```

```r
dim(Research_Variables_of_interest)
```

```
[1] 1378   20
```

Synthesizing some basic summary/descriptive statistics for the data applied.

```r
summary(Research_Variables_of_interest)
```

```
 RESPONDENT_AGE     2150 Sex       2151 Race      V2152 Where    V2153 Married
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :0.000   Min.   :1.00
 1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:4.00
 Median :2.000   Median :2.000   Median :2.000   Median :4.000   Median :4.00
 Mean   :1.529   Mean   :1.569   Mean   :2.209   Mean   :4.043   Mean   :3.66
 3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:6.000   3rd Qu.:4.00
 Max.   :2.000   Max.   :3.000   Max.   :3.000   Max.   :9.000   Max.   :4.00
   V2155 Dad?        V2156 Mom?     V2157 Siblings   V2163 Dad Educ
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:3.000
 Median :1.0000   Median :1.0000   Median :1.0000   Median :4.000
 Mean   :0.7417   Mean   :0.9136   Mean   :0.7177   Mean   :4.112
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:5.000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :7.000
 V2164 Mom Educ  V2165 Mom Work   V2174 Smart?     V2179 GPA
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:3.000   1st Qu.:2.000   1st Qu.:4.000   1st Qu.:6.000
 Median :5.000   Median :4.000   Median :5.000   Median :7.000
 Mean   :4.335   Mean   :3.035   Mean   :4.848   Mean   :6.921
 3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:6.000   3rd Qu.:9.000
 Max.   :7.000   Max.   :4.000   Max.   :7.000   Max.   :9.000
 V2183  College?  V2191 Work?      V2192 Money?     V2194   GO Out
 Min.   :1.000   Min.   :1.000   Min.   : 1.000   Min.   :1.000
 1st Qu.:3.000   1st Qu.:1.000   1st Qu.: 1.000   1st Qu.:1.000
 Median :3.000   Median :2.000   Median : 3.000   Median :3.000
 Mean   :3.152   Mean   :3.044   Mean   : 4.493   Mean   :2.684
 3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:4.000
 Max.   :4.000   Max.   :8.000   Max.   :10.000   Max.   :6.000
```

```
V2195 On dates? V2105 Drink alcohol V2116 smoke grass?
Min.   :1.000   Min.   :1.00        Min.   :1.000
1st Qu.:1.000   1st Qu.:1.00        1st Qu.:1.000
Median :1.000   Median :1.00        Median :1.000
Mean   :2.012   Mean   :2.23        Mean   :1.816
3rd Qu.:3.000   3rd Qu.:3.00        3rd Qu.:2.000
Max.   :6.000   Max.   :7.00        Max.   :7.000
```

Dependent Variable (Response Variable or Target) of Interest

Employment is chosen to be the dependent variable (target or response variable) of interest since it is often identified with maturity, responsibility and self sufficiency among adolescence. A variable from the data set that strongly represents employment is "V2191 Work?" with the following structure:
On the average over the school year, how many hours per week do you work in a paid or unpaid job?
1 = "None"
2 = "5 or less hours"
3 = "6 to 10 hours"
4 = "11 to 15 hours"
5 = "16 to 20 hours"
6 = "21 to 25 hours"
7 = "26 to 30 hours"
8 = "More than 30 hours"
Observed is ordinal values since values can be ranked.

Feature Importance/Selection

From the prior probe observed are categorical variables, ordinal variables and numeric/continuous variables transformed into ordinal variables. Now, to pursue dimensional reduction to 6 predictors or independent variables.
When working with ordinal variables and categorical variables in R, you may want to perform feature selection to identify the most important predictors for your analysis. Feature selection helps you reduce the dimensionality of your data set and improve model performance. There are various methods and packages you can use for ordinal feature selection in R.

Correlation

Correlation is a primitive means of feature (independent variable) selection. Correlation is a statistic that measures the degree to which two variables move in relation to each other. Measurement of the size and direction of the relationship between two or more variables. The correlation between variables doesn't directly mean a causal relationship among the variables. Commencing with a correlation measure.

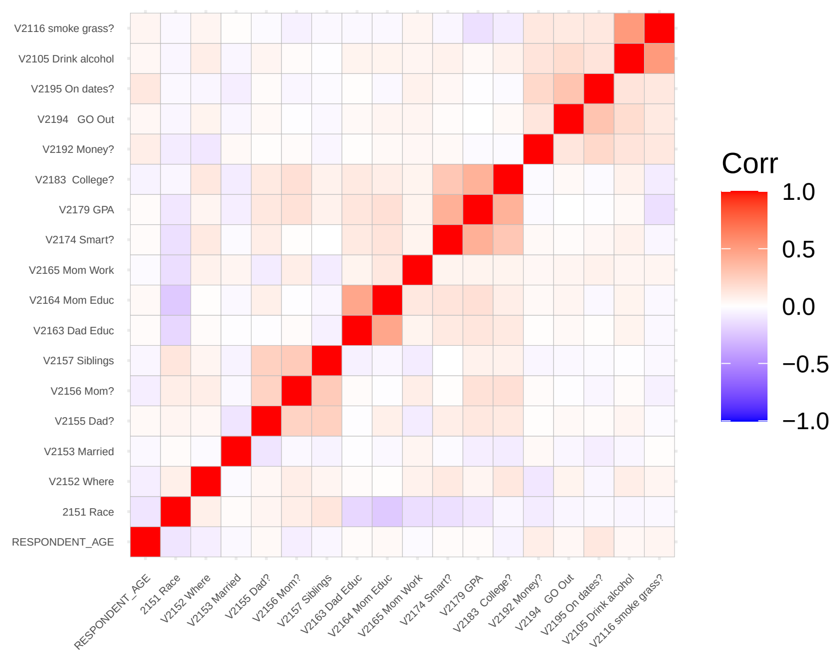
```
library(DataExplorer)
Features <- Research_Variables_of_interest |>
  dplyr::select(RESPONDENT_AGE, `2151 Race`, `V2152 Where`, `V2153 Married`,
         `V2155 Dad?`, `V2156 Mom?`, `V2157 Siblings`, `V2163 Dad Educ`,
         `V2164 Mom Educ`, `V2165 Mom Work`, `V2174 Smart?`, `V2179 GPA`,
         `V2183  College?`, `V2192 Money?`, `V2194   GO Out`,
         `V2195 On dates?`, `V2105 Drink alcohol`,
         `V2116 smoke grass?`
         )
library(corrr)
library(ggcorrplot)
```

```
Warning: package 'ggcorrplot' was built under R version 4.3.2
```

```
library(FactoMineR)
```

```
Warning: package 'FactoMineR' was built under R version 4.3.2
```

```
corr_matrix <- cor(Features)
ggcorrplot(corr_matrix, tl.cex = 4)
```



For fairness most of the variables were included to apply the primitive correlation measure. Results are observed prior, however such takes much time and effort to sort out due to the observed congestion in the display. Will pursue development of feature importance/selection to identify 6 of the most influential independent variables (or predictors).

Boruta Algorithm

The **Boruta Algorithm** is a wrapper built around the random forest classification algorithm. It tries to capture all the important, interesting features you might have in your dataset with respect to an outcome variable.

- First, it duplicates the dataset, and shuffle the values in each column. These values are called shadow features. Then, it trains a classifier, such as a Random Forest Classifier (ensemble learning with use of decision trees on various sub-samples of the data set and applies averaging to improve the predictive accuracy and control over-fitting). By doing this, you ensure that you can an idea of the importance -via the Mean Decrease Accuracy or Mean Decrease Impurity- for each of the features of your data set. The higher the score, the better or more important.
- Then, the algorithm checks for each of your real features if they have higher importance. That is, whether the feature has a higher Z-score than the maximum Z-score of its shadow features than the best of the shadow features. If they do, it records this in a

vector. These are called a hits. Next, it will continue with another iteration. After a predefined set of iterations, you will end up with a table of these hits.

- At every iteration, the algorithm compares the Z-scores of the shuffled copies of the features and the original features to see if the latter performed better than the former. If it does, the algorithm will mark the feature as important. In essence, the algorithm is trying to validate the importance of the feature by comparing with random shuffled copies, which increases the robustness. This is done by simply comparing the number of times a feature did better with the shadow features using a binomial distribution.

```
library(Boruta)
set.seed(111)
# Create a Boruta object
boruta_feature_results <- Boruta(`V2191 Work?` ~ .,
                                  data = Research_Variables_of_interest,
                                  doTrace = 2)
print(boruta_feature_results)
```

```
Boruta performed 99 iterations in 1.317763 mins.
 12 attributes confirmed important: 2151 Race, V2105 Drink alcohol,
V2116 smoke grass?, V2152 Where, V2163 Dad Educ and 7 more;
 6 attributes confirmed unimportant: 2150 Sex, RESPONDENT_AGE, V2153
Married, V2156 Mom?, V2157 Siblings and 1 more;
 1 tentative attributes left: V2155 Dad?;
```

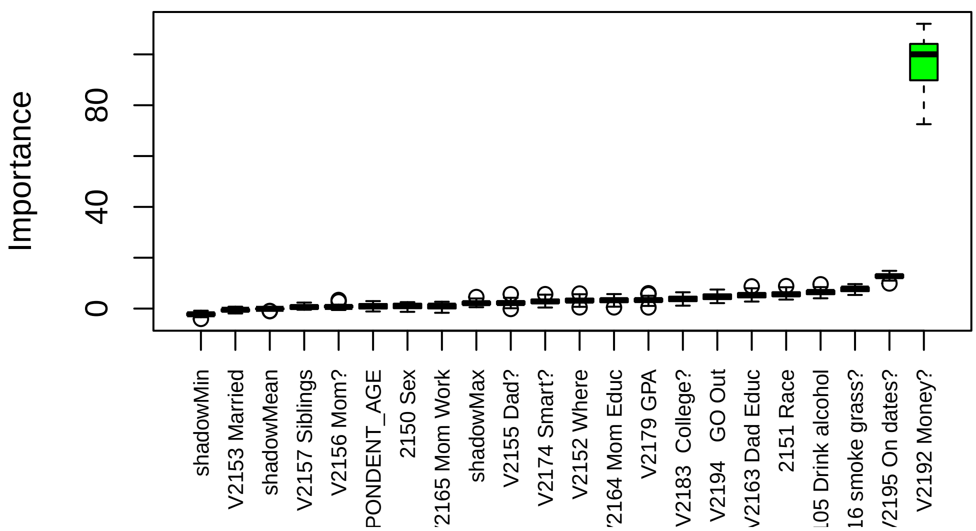
The `boruta` package also contains a `TentativeRoughFix()` function, which can be used to fill missing decisions by simple comparison of the median feature Z-score with the median Z-score of the most important shadow feature:

```
#take a call on tentative features
boruta_feature_results_fix <- TentativeRoughFix(boruta_feature_results)
print(boruta_feature_results_fix)
```

```
Boruta performed 99 iterations in 1.317763 mins.
Tentatives roughfixed over the last 99 iterations.
 13 attributes confirmed important: 2151 Race, V2105 Drink alcohol,
V2116 smoke grass?, V2152 Where, V2155 Dad? and 8 more;
 6 attributes confirmed unimportant: 2150 Sex, RESPONDENT_AGE, V2153
Married, V2156 Mom?, V2157 Siblings and 1 more;
```

To now plot the boruta variable importance chart by calling `plot(boruta.bank)`. However, the `x` axis labels will be horizontal. This won't be really neat. Hence, to add the feature labels to the `x` axis vertically:

```
plot(boruta_feature_results_fix, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(boruta_feature_results_fix$ImpHistory),function(i)
boruta_feature_results_fix$ImpHistory[is.finite(boruta_feature_results_fix$ImpHistory[,i])
names(lz) <- colnames(boruta_feature_results_fix$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
at = 1:ncol(boruta_feature_results_fix$ImpHistory), cex.axis = 0.7)
```



The blue boxplots correspond to minimal, average and maximum Z score of a shadow feature, while the red and green boxplots represent Z scores of rejected and confirmed features, respectively. As you can see the red boxplots have lower `Z` score than that of maximum `Z` score of shadow feature which is precisely the reason they were put in unimportant category.

Now, to confirm the importance of the features:

```
getSelectedAttributes(boruta_feature_results_fix, withTentative = F)
```

```
[1] "2151 Race"            "V2152 Where"          "V2155 Dad?"
```

```
 [4] "V2163 Dad Educ"      "V2164 Mom Educ"      "V2174 Smart?"
 [7] "V2179 GPA"           "V2183  College?"     "V2192 Money?"
[10] "V2194   GO Out"      "V2195 On dates?"     "V2105 Drink alcohol"
[13] "V2116 smoke grass?"
```

```
boruta_feature_results_fix_df <- attStats(boruta_feature_results_fix)
print(boruta_feature_results_fix_df)
```

```
                       meanImp   medianImp     minImp      maxImp   normHits
RESPONDENT_AGE       0.9802225   0.9857424 -1.1319098   2.9440125 0.05050505
2150 Sex             1.0306675   1.0639180 -1.2806903   2.5369272 0.04040404
2151 Race            5.6530080   5.6215558  3.5399436   8.8652162 1.00000000
V2152 Where          3.2390917   3.1201901  0.4636563   5.9230645 0.81818182
V2153 Married       -0.5667740  -0.4659278 -1.9235370   0.7388312 0.00000000
V2155 Dad?           2.2195332   2.1788089 -0.1608364   5.6718085 0.53535354
V2156 Mom?           0.8340082   0.6448146 -0.5563091   3.3852801 0.03030303
V2157 Siblings       0.6829942   0.6076659 -0.4604002   2.3608166 0.01010101
V2163 Dad Educ       5.2723574   5.3632175  2.7541002   8.7379128 0.98989899
V2164 Mom Educ       3.2094340   3.3228467  0.4691019   5.7078292 0.79797980
V2165 Mom Work       0.9283051   1.1478761 -1.6600635   2.6944000 0.05050505
V2174 Smart?         2.8672043   2.8194158  0.4038968   5.6670563 0.67676768
V2179 GPA            3.3040805   3.3422000  0.4520035   6.0392535 0.82828283
V2183  College?      3.8307004   3.8193080  1.1426720   6.4198249 0.88888889
V2192 Money?        97.2171315 100.0205023 72.5310440 112.0248818 1.00000000
V2194   GO Out       4.7169796   4.8020715  2.1504541   7.4885770 0.94949495
V2195 On dates?     12.7875085  12.7536721  9.8730118  14.8367839 1.00000000
V2105 Drink alcohol  6.3788575   6.5387818  4.0443437   9.5573033 1.00000000
V2116 smoke grass?   7.6769605   7.8713245  5.3569542   9.6284979 1.00000000
                     decision
RESPONDENT_AGE       Rejected
2150 Sex             Rejected
2151 Race           Confirmed
V2152 Where         Confirmed
V2153 Married        Rejected
V2155 Dad?          Confirmed
V2156 Mom?           Rejected
V2157 Siblings       Rejected
V2163 Dad Educ      Confirmed
V2164 Mom Educ      Confirmed
V2165 Mom Work       Rejected
V2174 Smart?        Confirmed
```

```
V2179 GPA           Confirmed
V2183  College?     Confirmed
V2192 Money?        Confirmed
V2194   GO Out      Confirmed
V2195 On dates?     Confirmed
V2105 Drink alcohol Confirmed
V2116 smoke grass?  Confirmed
```

The first line of code uses the **getSelectedAttributes()** function from the **Boruta** package to retrieve the selected attributes from the **boruta__feature__results__fix** object.

- The **withTentative** argument is set to **F** to exclude tentative attributes from the output.

- The selected attributes are printed as a character vector.

- The second line of code creates a new object called **bank__feature-results__fix__df** using the **attStats()** function from the **Boruta** package. This function calculates various statistics for each attribute in the **boruta__feature__results__fix** object, including mean importance, median importance, minimum importance, maximum importance, normalized hits, and decision (whether the attribute is confirmed, rejected, or tentative).

- The resulting object is a data frame.

- The third line of code prints the **bank__feature__results__fix__df** data frame to the console.

- The data frame shows the calculated statistics for each attribute in the **boruta__features__results__fix** object.

- The **meanImp** column represents the mean importance of each attribute, while the **decision** column indicates whether each attribute is confirmed, rejected, or tentative.

The selected features or independent variables (or predictors) are: “2151 Race”, “V2163 Dad Educ”, “V2192 Money?”, “V2195 On dates?”, “V2105 Drink alcohol”, and “V2116 smoke grass?”

Explanation of the Selected Predictors (Independent Variables/Features)

Variable characteristics from the ICPSR 38503 Survey codebook:

1. **First variable**, "V2192 Money?" is based on the following survey question: During an average week, how much money did you get from . . . a job or other work?
1="None"; 2="$1-5"; 3="$6-10"; 4="$11-20"; 5=$21-35"; 6="$36-50"; 7="$51-75"; 8="$76-125"; 9="$126-175"; 10="176+"
Observed is ordinal values since values can be ranked.

2. **Second variable**, "V2195 On dates?" is based on the following survey question: On the average, how often do you go out with a date (or your spouse/partner, if you are married)?
1="Never"; 2="Once a month or less"; 3="2 or 3 times a month"; 4="Once a week"; 5="2 or 3 times a week"; 6="Over 3 times a week".
Observed is ordinal values since values can be ranked.

3. **Third variable**, "V2116 smoke grass?" is based on the following questioning structure:
Form 1: On how many occasions (if any) have you used marijuana [sometimes called: Weed, Pot, Dope] or hashish [sometimes called: Hash, Hash oil]. . . during the last 12 months? [Separate questions for marijuana (Item 02080) and hashish (Item 02050) are combined in this variable for form 1.]
Form 3: On how many occasions (if any) have you used marijuana (weed, pot) or hashish (hash, hash oil) (Do NOT count any use of CBD products) . . . during the last 12 months?
Form 5: On how many occasions (if any) have you used marijuana (weed, pot) or hashish (hash, hash oil). . . .during the last 12 months?
Forms 2, 4, and 6: On how many occasions (if any) have you used marijuana in any form (e.g. smoking, vaping, edibles, hashish, hash oil). . . during the last 12 months?
1="0 Occasions"; 2="1-2 Occasions"; 3="3-5 Occasions"; 4="6-9 Occasions"; 5="10-19 Occasions"; 6="20-39 Occasions"; 7="40 or More".
Observed is ordinal values since values can be ranked.

4. **Fourth variable**, "V2105 Drink alcohol" is based on the following survey question: On how many occasions (if any) have you had alcoholic beverages to drink–more than just a few sips . . . during the last 12 months?
On how many occasions (if any) have you had alcoholic beverages to drink--more than just a few sips . . . during the last 12 months?
1="0 Occasions"; 2="1-2 Occasions"; 3="3-5 Occasions"; 4="6-9 Occasions"; 5="10-19 Occasions"; 6="20-39 Occasions"; 7="40 or More".
Observed is ordinal values since values can be ranked.

5. **Fifth variable**, "2151 Race" is based on the following survey question: How do you describe yourself?
Select one or more responses: Black or African American; Mexican American or Chicano;

Cuban American; Puerto Rican; Other Hispanic or Latino; Asian American; White (Caucasian); American Indian or Alaska Native; Native Hawaiian or Other Pacific Islander; Middle Eastern.
Recoded in this dataset so that "Black or African American" = 1, "White (Caucasian)" = 2; Hispanic = 3 ("Mexican…" or "Cuban…" or "Puerto Rican" or "Other Hispanic…").
Observed is categorical values since values in this case can't be ranked.

6. **Sixth variable**, "V2163 Dad Educ" is based on the following questioning structure:
The next three questions ask about your parents. If you were raised mostly by foster parents, stepparents, or others, answer for them. For example, if you have both a stepfather and a natural father, answer for the one that was the most important in raising you. What is the highest level of schooling your father completed?
1="Completed grade school or less"; 2="Some high school"; 3="Completed high school"; 4="Some college"; 5="Completed college"; 6="Graduate or professional school after college"; 7="Don't know, or does not apply".
Observed is ordinal values since values can be ranked.

Then, pursuing view of correlation for the selected independent variables based on the Boruta Algorithm:

```
Realized_features <- Research_Variables_of_interest |>
  dplyr::select(`2151 Race`, `V2163 Dad Educ`, `V2192 Money?`, `V2195 On dates?`,
         `V2105 Drink alcohol`, `V2116 smoke grass?`)
```

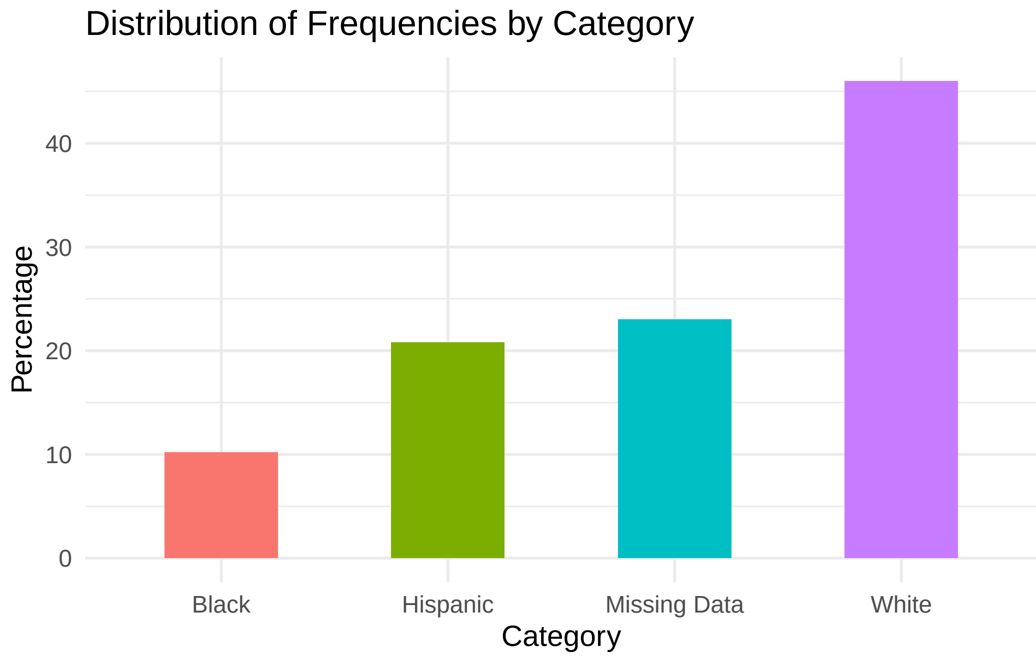
Will now observe the frequencies or distribution of each selected dependent variable (predictor).
From the ICPSR 38503 Survey codebook:
Variable "2121 Race". For unweighted frequencies recorded, 916 (Black), 4153 (White), 1878 (Hispanic), and 2075 (missing data) the respective percentages are 10.2, 46.0, 20.8 and 23.0. Develop distribution in R.

```
# Dataframe structuring example with high accuracy for percentages.
# Given unweighted frequencies
Black_freq <- 916
White_freq <- 4153
Hispanic_freq <- 1878
Missing_freq <- 2075

# Calculate the total frequency
total_freq <- Black_freq + White_freq + Hispanic_freq + Missing_freq

# Calculate percentages
```

```r
Black_percentage <- (Black_freq / total_freq) * 100
White_percentage <- (White_freq / total_freq) * 100
Hispanic_percentage <- (Hispanic_freq / total_freq) * 100
Missing_percentage <- (Missing_freq / total_freq) * 100

# Create a data frame for the distribution
Race_distribution <- data.frame(
  Category = c("Black", "White", "Hispanic", "Missing Data"),
  Frequency = c(Black_freq, White_freq, Hispanic_freq, Missing_freq),
  Percentage = c(Black_percentage, White_percentage, Hispanic_percentage,
                 Missing_percentage)
)

# Print the distribution
print(Race_distribution)
```

```
      Category Frequency Percentage
1        Black       916   10.15296
2        White      4153   46.03192
3     Hispanic      1878   20.81578
4 Missing Data      2075   22.99933
```

```r
# Given unweighted frequencies and percentages
categories <- c("Black", "White", "Hispanic", "Missing Data")
frequencies <- c(916, 4153, 1878, 2075)
percentages <- c(10.2, 46.0, 20.8, 23.0)

# Create a data frame
data <- data.frame(Category = categories, Frequency = frequencies,
                   Percentage = percentages)

# Create a bar plot
library(ggplot2)
ggplot(data, aes(x = Category, y = Percentage, fill = Category)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Distribution of Frequencies by Category",
       x = "Category",
       y = "Percentage") +
  theme_minimal() +
  theme(legend.position = "none")
```



Variable "V2163 Dad Educ". For unweighted frequencies recorded, 395 (Grade School), 1036 (Some High School), 2021(High school Graduate), 1063 (Some College), 1737 (College Graduate), 1061 (Graduate School), 901 (Don't Know) and 808 (Missing Data). Develop distribution in R.

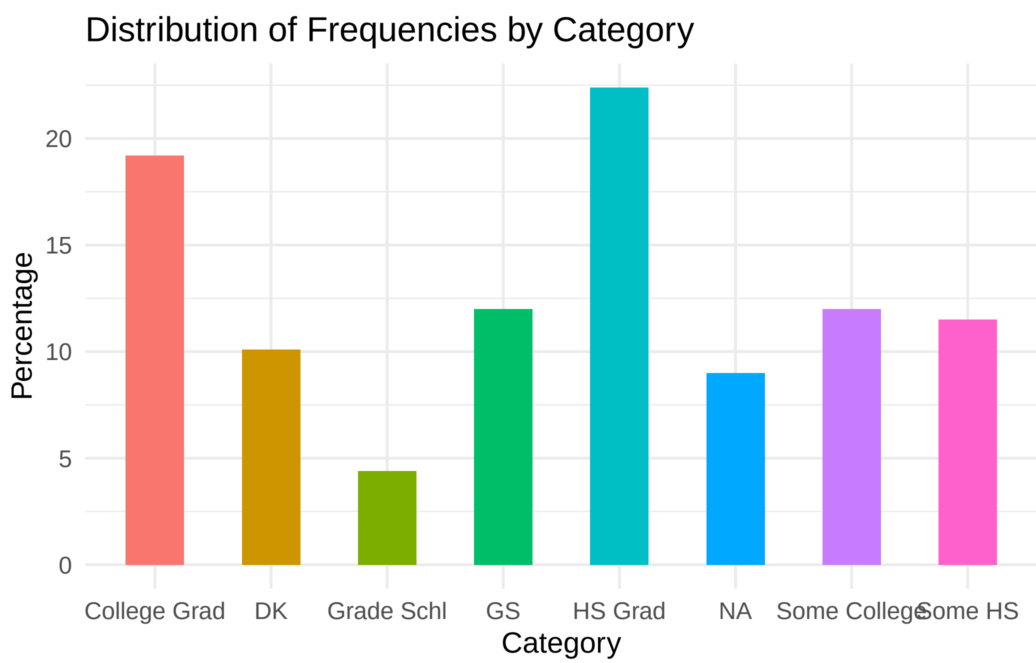
```
# Dataframe structuring example with high accuracy for percentages.
# Given unweighted frequencies
Grade_School_freq <- 395
Some_High_School_freq <- 1036
High_School_Graduate_freq <- 2021
Some_College_freq <- 1063
College_Graduate_freq <- 1737
Graduate_School_freq <- 1061
Dont_Know_freq <- 910
Missing_Data_freq <- 808

# Calculate the total frequency
total_freq <- Grade_School_freq + Some_High_School_freq +
  High_School_Graduate_freq + Some_College_freq + College_Graduate_freq +
  Graduate_School_freq + Dont_Know_freq + Missing_Data_freq

# Calculate percentages
```

```r
Grade_School_percentage <- (Grade_School_freq / total_freq) * 100
Some_High_School_percentage <- (Some_High_School_freq / total_freq) * 100
High_School_Graduate_percentage <- (High_School_Graduate_freq / total_freq) * 100
Some_College_percentage <- (Some_College_freq / total_freq) * 100
College_Graduate_percentage <- (College_Graduate_freq / total_freq) * 100
Graduate_School_percentage <- (Graduate_School_freq /total_freq) * 100
Dont_Know_percentage <- (Dont_Know_freq /total_freq) * 100
Missing_data_percentage <- (Missing_Data_freq/total_freq) * 100

# Create a data frame for the distribution
Dad_educ_distribution <- data.frame(
  Category = c("Grade School", "Some HS", "HS Grad", "Some College",
               "College Grad", "Grad School", "DK","Missing Data"),
  Frequency = c(Grade_School_freq, Some_High_School_freq,
                High_School_Graduate_freq, Some_College_freq,
                College_Graduate_freq, Graduate_School_freq,
                Dont_Know_freq, Missing_Data_freq),
  Percentage = c(Grade_School_percentage, Some_High_School_percentage,
                 High_School_Graduate_percentage, Some_College_percentage,
                 College_Graduate_percentage, Graduate_School_percentage,
                 Dont_Know_percentage, Missing_data_percentage))

# Print the distribution
print(Dad_educ_distribution)
```

```
      Category Frequency Percentage
1 Grade School       395   4.373823
2      Some HS      1036  11.471598
3      HS Grad      2021  22.378474
4 Some College      1063  11.770568
5 College Grad      1737  19.233750
6  Grad School      1061  11.748422
7           DK       910  10.076403
8 Missing Data       808   8.946960
```

```r
# Given unweighted frequencies and percentages
categories <- c("Grade Schl", "Some HS", "HS Grad", "Some College",
                "College Grad", "GS", "DK", "NA")
frequencies <- c(395, 1036, 2021, 1063, 1737, 1061, 910, 808)
percentages <- c(4.4, 11.5, 22.4, 12, 19.2, 12, 10.1, 9)
```

```
# Create a data frame
data <- data.frame(Category = categories,
                   Frequency = frequencies, Percentage = percentages)

# Create a bar plot
library(ggplot2)
ggplot(data, aes(x = Category, y = Percentage, fill = Category)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Distribution of Frequencies by Category",
       x = "Category",
       y = "Percentage") +
  theme_minimal() +
  theme(legend.position = "none")
```



Variable "V2192 Money?". For unweighted frequencies, 3487 (NONE), 27 ($1-5), 344 ($6-10), 481 ($11-20), 147 ($21-35), 202 ($36-50), 328 ($51-75), 868 ($76-125), 1393 ($176+)

```
# Dataframe structuring example with high accuracy for percentages.
# Given unweighted frequencies
NONE_freq <- 3487
one_to_five_freq <- 27
```

```
six_to_ten_freq <- 344
eleven_to_twenty_freq <- 481
twenty_one_to_thirty_five_freq <- 147
thrity_six_to_fifty_freq <- 202
fifty_one_to_seventy_five_freq <- 328
seventy_six_to_one_hundred_twenty_five_freq <- 868
one_hundred_twenty_six_to_one_hundred_seventy_five_freq <- 659
one_hundred_seventy_six_plus_freq <- 1393
missing_data_freq <- 12

# Calculate the total frequency
total_freq <- NONE_freq + one_to_five_freq + six_to_ten_freq
+ eleven_to_twenty_freq +
  twenty_one_to_thirty_five_freq +
  thrity_six_to_fifty_freq +
  fifty_one_to_seventy_five_freq +
  seventy_six_to_one_hundred_twenty_five_freq
```

```
[1] 2026
```

```
+ one_hundred_twenty_six_to_one_hundred_seventy_five_freq
```

```
[1] 659
```

```
+ one_hundred_seventy_six_plus_freq + missing_data_freq
```

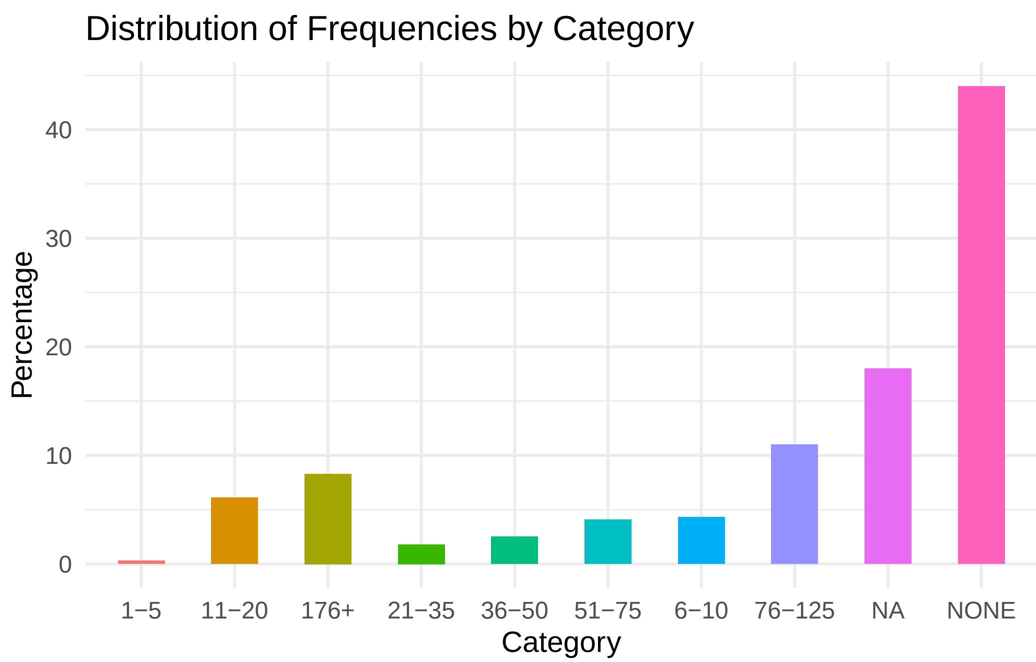
```
[1] 1405
```

```
# Calculate percentages
NONE_percentage <- (NONE_freq / total_freq) * 100

one_to_five_percentage <- (one_to_five_freq / total_freq) * 100

six_to_ten_percentage <- (six_to_ten_freq / total_freq) * 100

eleven_to_twenty_percentage <- (eleven_to_twenty_freq / total_freq) * 100

twenty_one_to_thirty_five_percentage <-
```

```r
  (twenty_one_to_thirty_five_freq / total_freq) * 100

thrity_six_to_fifty_percentage <- (thrity_six_to_fifty_freq / total_freq) * 100

fifty_one_to_seventy_five_percentage <-
  (fifty_one_to_seventy_five_freq / total_freq) * 100

seventy_six_to_one_hundred_twenty_five_percentage <-
  (seventy_six_to_one_hundred_twenty_five_freq / total_freq) * 100

one_hundred_twenty_six_to_one_hundred_seventy_five_percentage <-
  (one_hundred_twenty_six_to_one_hundred_seventy_five_freq / total_freq) * 100

one_hundred_seventy_six_plus_percentage <-
  (one_hundred_seventy_six_plus_freq / total_freq) * 100

missing_data_percentage <- (missing_data_freq / total_freq) * 100

# Create a data frame for the distribution
Money_distribution <- data.frame(
  Category = c("NONE", "1-5", "6-10", "11-20",
               "21-35", "36-50", "51-75","76-125", "126-175", "176+", "NA"),
  Frequency = c(NONE_freq, one_to_five_freq,
                six_to_ten_freq, eleven_to_twenty_freq,
                twenty_one_to_thirty_five_freq, thrity_six_to_fifty_freq,
                fifty_one_to_seventy_five_freq,
                seventy_six_to_one_hundred_twenty_five_freq,
                one_hundred_twenty_six_to_one_hundred_seventy_five_freq,
                one_hundred_seventy_six_plus_freq, missing_data_freq),
  Percentage = c(NONE_percentage, one_to_five_percentage,
                 six_to_ten_percentage,
                 eleven_to_twenty_percentage,
                 twenty_one_to_thirty_five_percentage,
                 thrity_six_to_fifty_percentage,
                 fifty_one_to_seventy_five_percentage,
                 seventy_six_to_one_hundred_twenty_five_percentage,
                 one_hundred_twenty_six_to_one_hundred_seventy_five_percentage,
                 one_hundred_seventy_six_plus_percentage, missing_data_percentage))

# Print the distribution
print(Money_distribution)
```

```
   Category Frequency Percentage
1      NONE      3487 90.3836185
2       1-5        27  0.6998445
3      6-10       344  8.9165371
4     11-20       481 12.4675998
5     21-35       147  3.8102644
6     36-50       202  5.2358735
7     51-75       328  8.5018144
8    76-125       868 22.4987040
9   126-175       659 17.0813893
10     176+      1393 36.1067911
11       NA        12  0.3110420
```

```r
# Given unweighted frequencies and percentages
categories <- c("NONE", "1-5", "6-10", "11-20",
                "21-35", "36-50", "51-75", "76-125", "176+", "NA")
frequencies <- c(3487, 27, 344, 481, 147, 202, 328, 868, 659, 1393)
percentages <- c(44, 0.3, 4.3, 6.1, 1.8, 2.5, 4.1, 11, 8.3, 18)

# Create a data frame
data <- data.frame(Category = categories,
                   Frequency = frequencies, Percentage = percentages)

# Create a bar plot
library(ggplot2)
ggplot(data, aes(x = Category, y = Percentage, fill = Category)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Distribution of Frequencies by Category",
       x = "Category",
       y = "Percentage") +
  theme_minimal() +
  theme(legend.position = "none")
```



Variable "V2195 On dates?". For unweighted frequencies, 4723(Never), 996(Once per month), 768(2-3X per month), 720(Once per week), 567(2-3X per week), 225(3+ per week)

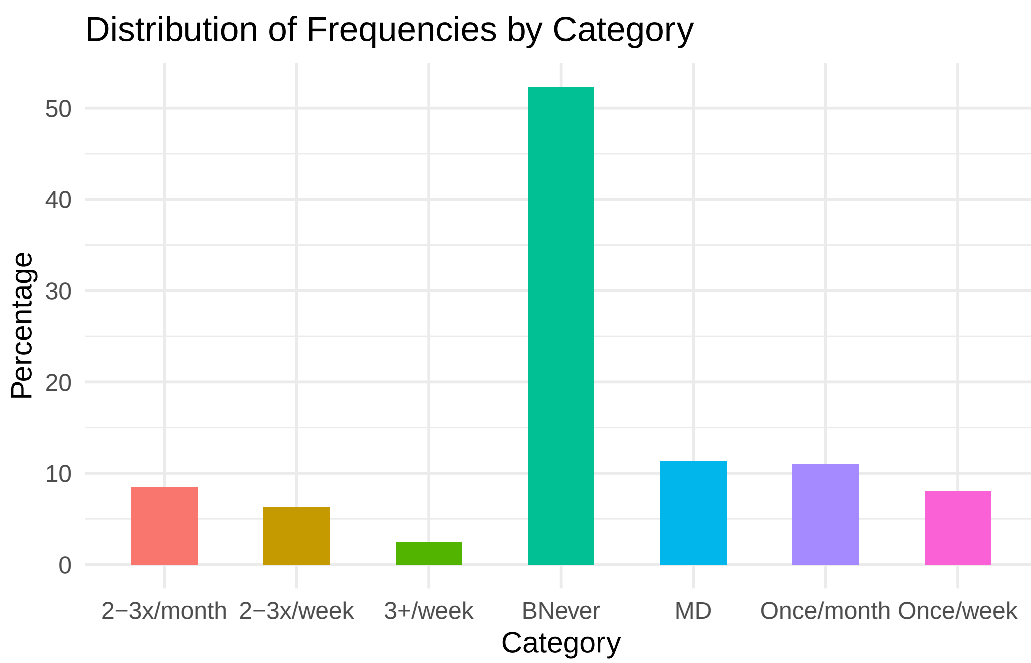
```
# Dataframe structuring example with high accuracy for percentages.
# Given unweighted frequencies
Never_freq <- 4723
Once_per_month_freq <- 996
Two_to_three_per_month_freq <- 768
Once_per_week_freq <- 720
Two_to_three_per_week_freq <- 567
three_or_more_per_week_freq <-225
Miss_data_freq <- 1023

# Calculate the total frequency
total_freq <- Never_freq + Once_per_month_freq +
  Two_to_three_per_month_freq + Once_per_week_freq +
  Two_to_three_per_week_freq + three_or_more_per_week_freq +
  Miss_data_freq

# Calculate percentages
Never_percentage <- (Never_freq / total_freq) * 100
```

```r
Once_per_month_percentage <- (Once_per_month_freq / total_freq) * 100
Two_to_three_per_month_percentage <- (Two_to_three_per_month_freq /
                                        total_freq) * 100
Once_per_week_percentage <- (Once_per_week_freq / total_freq) * 100
Two_to_three_per_week_percentage <- (Two_to_three_per_week_freq /
                                       total_freq) * 100
three_or_more_per_week_percentage <- (three_or_more_per_week_freq /
                                        total_freq) * 100
Miss_data_percentage <- (Miss_data_freq / total_freq) * 100

# Create a data frame for the distribution
Dates_distribution <- data.frame(
  Category = c("Never", "Once/month", "2-3x/month", "Once/week",
               "2-3x/week", "3+/week", "Missing Data"),
  Frequency = c(Never_freq, Once_per_month_freq, Two_to_three_per_month_freq,
                Once_per_week_freq, Two_to_three_per_week_freq,
                three_or_more_per_week_freq, Miss_data_freq),
  Percentage = c(Never_percentage, Once_per_month_percentage,
                 Two_to_three_per_month_percentage, Once_per_week_percentage,
                 Two_to_three_per_week_percentage,
                 three_or_more_per_week_percentage, Miss_data_percentage)
)

# Print the distribution
print(Dates_distribution)
```

```
      Category Frequency Percentage
1        Never      4723  52.349812
2   Once/month       996  11.039681
3   2-3x/month       768   8.512525
4    Once/week       720   7.980492
5    2-3x/week       567   6.284638
6      3+/week       225   2.493904
7 Missing Data      1023  11.338949
```

```r
# Given unweighted frequencies and percentages
categories <- c("BNever", "Once/month", "2-3x/month",
                "Once/week", "2-3x/week", "3+/week", "MD")
frequencies <- c(4723, 996, 768, 720, 567, 225, 1023)
percentages <- c(52.3, 11.0, 8.5, 8.0, 6.3, 2.5, 11.3)
```

```
# Create a data frame
data <- data.frame(Category = categories,
                   Frequency = frequencies, Percentage = percentages)

# Create a bar plot
library(ggplot2)
ggplot(data, aes(x = Category, y = Percentage, fill = Category)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Distribution of Frequencies by Category",
       x = "Category",
       y = "Percentage") +
  theme_minimal() +
  theme(legend.position = "none")
```

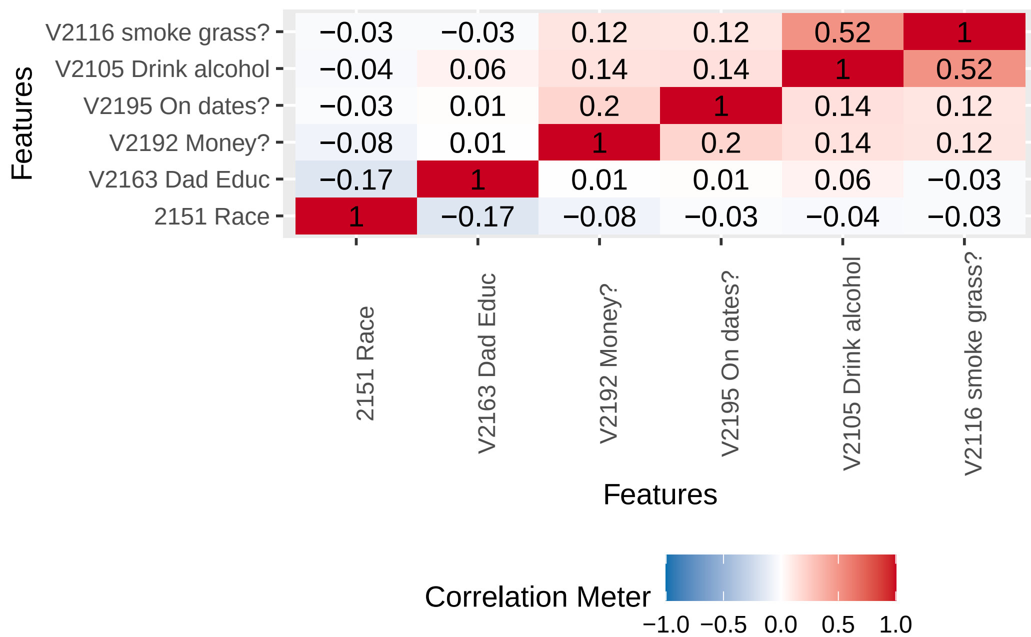


Distribution displays can also be developed for “V2105 Drink alcohol” and “V2116 smoke grass?” predictors in similar manner.

Correlation Measures and Descriptive Statistics for the Selected Predictors

Since feature importance/selection has been applied, to then review the correlations among the selected predictors, being easier to view compared to the large congested set of variables encountered in the beginning. Recalling, the correlation between variables doesn't directly mean a causal relationship among the variables. As well, the target or response variable of concern is employment, so lack of association among the chosen predictors is inconsequential. Commencing with the correlation measures.

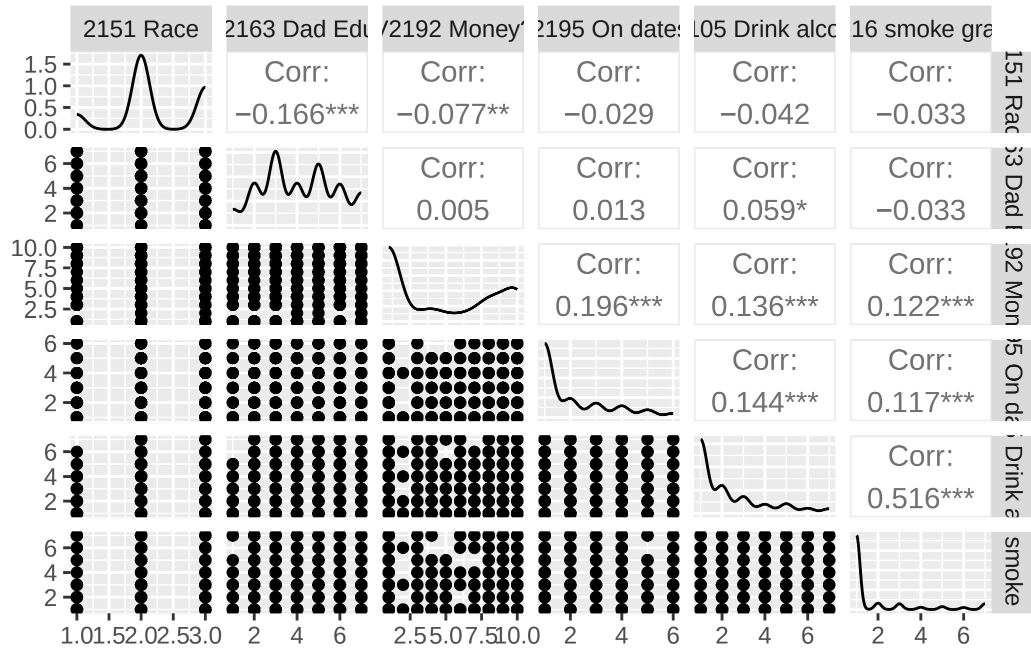
```
library(DataExplorer)
plot_correlation(Realized_features)
```




```
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```
ggpairs(Realized_features)
```



The observed correlation values among the selected independent variables convey lack of multicollinearity. Two variables are perfectly collinear if their correlation coefficient is +/- 1.0. Multicollinearity among independent variable will result in less reliable statistical inferences. Thus, the observed correlation values support the notion of highly "unique" features, namely, the features selected don't have "replicative" behavior among each other.
Since feature importance/selection has been applied, to then review the summary/descriptive statistics of the selected predictors (independent variables), being easier to view compared to the highly congested 19 prospect independent variables encountered in the beginning; recall that "V2191 Work?" was chosen to be the dependent variable (or response variable or target).

```
summary(Realized_features)
```

```
  2151 Race     V2163 Dad Educ   V2192 Money?    V2195 On dates?
 Min.   :1.000   Min.   :1.000   Min.   : 1.000   Min.   :1.000
 1st Qu.:2.000   1st Qu.:3.000   1st Qu.: 1.000   1st Qu.:1.000
 Median :2.000   Median :4.000   Median : 3.000   Median :1.000
 Mean   :2.209   Mean   :4.112   Mean   : 4.493   Mean   :2.012
 3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:3.000
 Max.   :3.000   Max.   :7.000   Max.   :10.000   Max.   :6.000
 V2105 Drink alcohol V2116 smoke grass?
 Min.   :1.00        Min.   :1.000
```

```
 1st Qu.:1.00        1st Qu.:1.000
 Median :1.00        Median :1.000
 Mean   :2.23        Mean   :1.816
 3rd Qu.:3.00        3rd Qu.:2.000
 Max.   :7.00        Max.   :7.000
```

Other possible statistics of interest include standard deviation, skewness and kurtosis.

```r
# Standard Deviation measures the dispersion of the data relative to:
     # its mean and is calculated as the square root of the variance.
# Standard Deviation values for the chosen predictors
sd(Realized_features$`2151 Race`)
```

```
[1] 0.6249522
```

```r
sd(Realized_features$`V2163 Dad Educ`)
```

```
[1] 1.70768
```

```r
sd(Realized_features$`V2192 Money?`)
```

```
[1] 3.767425
```

```r
sd(Realized_features$`V2195 On dates?`)
```

```
[1] 1.430244
```

```r
sd(Realized_features$`V2105 Drink alcohol`)
```

```
[1] 1.662539
```

```r
sd(Realized_features$`V2116 smoke grass?`)
```

```
[1] 1.708739
```

```r
library(moments)
# Skew measure (of symmetry) for the chosen predictors.
#Skewness is the degree of asymmetry observed in a probability distribution.
# Distributions can exhibit right (positive) skewness;
# Left (negative) skewness to varying degrees.
# A normal distribution (bell curve) exhibits zero skewness.
skewness(Realized_features$`2151 Race`)
```

```
[1] -0.183889
```

```r
skewness(Realized_features$`V2163 Dad Educ`)
```

```
[1] 0.07936094
```

```r
skewness(Realized_features$`V2192 Money?`)
```

```
[1] 0.364167
```

```r
skewness(Realized_features$`V2195 On dates?`)
```

```
[1] 1.26259
```

```r
skewness(Realized_features$`V2105 Drink alcohol`)
```

```
[1] 1.34813
```

```r
skewness(Realized_features$`V2116 smoke grass?`)
```

```
[1] 2.113706
```

```r
# Fisher-Pearson Kurtosis measure (of "tailedness") of the distribution;
# for the chosen predictors.There are three kurtosis categories, say,
# mesokurtic (normal), platykurtic (less than normal),
```

```r
# and leptokurtic (more than normal).
# Mesokurtic has kurtosis value of 3.0;
# Leptokurtic has kurtosis value > 3.0;
# Platykurtic has kurtosis value < 3.0.
kurtosis(Realized_features$`2151 Race`)
```

```
[1] 2.411048
```

```r
kurtosis(Realized_features$`V2163 Dad Educ`)
```

```
[1] 2.007182
```

```r
kurtosis(Realized_features$`V2192 Money?`)
```

```
[1] 1.370079
```

```r
kurtosis(Realized_features$`V2195 On dates?`)
```

```
[1] 3.450513
```

```r
kurtosis(Realized_features$`V2105 Drink alcohol`)
```

```
[1] 3.834658
```

```r
kurtosis(Realized_features$`V2116 smoke grass?`)
```

```
[1] 6.179103
```

Ordinal Regression Analysis based on Boruta Feature Importance

Ordinal Regression permits the modeling of the dependence of a polytomous (multi-score) ordinal response on a set of predictors, which can be factors or covariates. The design of Ordinal Regression is based on the methodology of McCullagh (1980, 1998). Standard regression concerns numerical or continuous variables where data is not ranked. Scoring or ranking with ordinal data can be arbitrary when it comes to scaling of the data in question. An example, temperature by consensus is identified as continuous data, where the difference in temperature between 150 degrees centigrade and 140 degrees is 10 degrees centigrade, which has the same meaning as the difference in temperature between 210 degrees centigrade and 200 degrees centigrade. These relationships do not necessarily hold for ordinal variables, in which the choice and number of response categories can be quite arbitrary. For the ordinal setting with boiling points, an example, various substances can be categorized where the temperature ranges among the categories can have great disparity; water, saltwater and tea in one category compared to a tin category, compared a nickle category, compared to copper and iron in another category. There's similar logic in karate where one tries to compare a green belt to a brown belt to a black belt, considering the amount of levels for each belt. Hence, seeking status quo distributions such as the normal distribution with ordinal variables may not be meaningful as one would like since the distribution orientation (skew, kurtosis and standard deviation) for the underlying numeric or continuous variable in question may be masked when its values are scaled to ranks by codes or rules that can greatly vary.

Commencing with Ordinal Regression Analysis. The predictor "V2192 Money?" recognized as the strongest predictor (feature) of importance by the Boruta algorithm, all ordinal logistic regression models to be compared will at least have such a predictor

```r
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```r
# fit ordered logit model and store results 'ordional_regressor'
ordinal_regressor_1 <- polr(as.factor(`V2191 Work?`) ~ `2151 Race` +
                              `V2163 Dad Educ` +
                              `V2192 Money?` + `V2195 On dates?` +
                              `V2116 smoke grass?` + `V2105 Drink alcohol`,
                            data = Research_Variables_of_interest)
```

```r
ordinal_regressor_1
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `2151 Race` + `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?` +
    `V2105 Drink alcohol`, data = Research_Variables_of_interest)

Coefficients:
          `2151 Race`      `V2163 Dad Educ`        `V2192 Money?`
          -0.18030392           -0.07779639            0.51037201
    `V2195 On dates?`  `V2116 smoke grass?` `V2105 Drink alcohol`
           0.10057719            0.03384829            0.02376181

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.335252 2.078458 2.965850 3.697442 4.491242 5.430123 6.389305

Residual Deviance: 3821.991
AIC: 3847.991
```

```r
summary(ordinal_regressor_1)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `2151 Race` + `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?` +
    `V2105 Drink alcohol`, data = Research_Variables_of_interest)

Coefficients:
                         Value Std. Error t value
`2151 Race`           -0.18030    0.09175 -1.9651
`V2163 Dad Educ`      -0.07780    0.03289 -2.3655
`V2192 Money?`         0.51037    0.01937 26.3440
`V2195 On dates?`      0.10058    0.03710  2.7107
`V2116 smoke grass?`   0.03385    0.03521  0.9613
`V2105 Drink alcohol`  0.02376    0.03701  0.6421

Intercepts:
    Value   Std. Error t value
1|2  1.3353  0.2909     4.5908
2|3  2.0785  0.2968     7.0038
```

```
3|4  2.9659  0.3046     9.7377
4|5  3.6974  0.3106    11.9032
5|6  4.4912  0.3167    14.1816
6|7  5.4301  0.3250    16.7077
7|8  6.3893  0.3395    18.8172

Residual Deviance: 3821.991
AIC: 3847.991
```

```r
library(car)
```

```
Loading required package: carData


Attaching package: 'car'


The following object is masked from 'package:dplyr':

    recode


The following object is masked from 'package:purrr':

    some
```

```r
poTest(ordinal_regressor_1)
```

```

Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `2151 Race` + `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?` +
    `V2105 Drink alcohol`, data = Research_Variables_of_interest)

                        b[polr]    b[>1]    b[>2]    b[>3]    b[>4]
Overall
`2151 Race`            -0.18030 -0.25888 -0.24306 -0.22275 -0.20880
`V2163 Dad Educ`       -0.07780 -0.03433 -0.12732 -0.12069 -0.09938
`V2192 Money?`          0.51037  0.65544  0.55552  0.44418  0.40758
`V2195 On dates?`       0.10058  0.16545  0.15018  0.19528  0.09687
`V2116 smoke grass?`    0.03385  0.04710  0.06682  0.02060  0.09391
```

```
`V2105 Drink alcohol`   0.02376   0.07524  -0.01895  -0.03730  -0.07795
                          b[>5]     b[>6]     b[>7] Chisquare df Pr(>Chisq)
Overall                                              117.43 36    1.5e-10 ***
`2151 Race`            -0.11583  -0.06023   0.24499      4.65  6      0.589
`V2163 Dad Educ`       -0.10059  -0.09618  -0.09332      5.56  6      0.475
`V2192 Money?`          0.34952   0.31131   0.27453     70.92  6    2.7e-13 ***
`V2195 On dates?`       0.13226   0.14051   0.09892      6.97  6      0.323
`V2116 smoke grass?`    0.11815  -0.01367  -0.09897     14.10  6      0.029 *
`V2105 Drink alcohol`  -0.07818  -0.00236   0.09497     10.41  6      0.108
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
library(DescTools)
```

```
Warning: package 'DescTools' was built under R version 4.3.2
```

```r
PseudoR2(ordinal_regressor_1, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3847.9908541    0.5263572    0.5423977
```

```r
# fit ordered logit model and store results 'ordional_regressor'
ordinal_regressor_2 <- polr(as.factor(`V2191 Work?`) ~`V2163 Dad Educ` +
                            `V2192 Money?` + `V2195 On dates?` +
                            `V2116 smoke grass?` + `V2105 Drink alcohol`,
                          data = Research_Variables_of_interest)
ordinal_regressor_2
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?` +
    `V2105 Drink alcohol`, data = Research_Variables_of_interest)

Coefficients:
     `V2163 Dad Educ`        `V2192 Money?`     `V2195 On dates?`
          -0.06740476            0.51179460            0.10011616
 `V2116 smoke grass?` `V2105 Drink alcohol`
           0.03517660            0.02429574
```

```
Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8 
1.784266 2.525211 3.411417 4.142102 4.935265 5.874617 6.834836 

Residual Deviance: 3825.844 
AIC: 3849.844 
```

```r
summary(ordinal_regressor_2)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2163 Dad Educ` + 
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?` + 
    `V2105 Drink alcohol`, data = Research_Variables_of_interest)

Coefficients:
                         Value Std. Error t value
`V2163 Dad Educ`      -0.06740    0.03234 -2.0840
`V2192 Money?`         0.51179    0.01936 26.4375
`V2195 On dates?`      0.10012    0.03710  2.6988
`V2116 smoke grass?`   0.03518    0.03519  0.9996
`V2105 Drink alcohol`  0.02430    0.03699  0.6569

Intercepts:
    Value   Std. Error t value
1|2  1.7843  0.1826     9.7721
2|3  2.5252  0.1933    13.0611
3|4  3.4114  0.2061    16.5547
4|5  4.1421  0.2156    19.2083
5|6  4.9353  0.2250    21.9379
6|7  5.8746  0.2368    24.8118
7|8  6.8348  0.2562    26.6812

Residual Deviance: 3825.844 
AIC: 3849.844 
```

The **residual deviance** above tells us how well the response variable can be predicted by a model with p predictor variables. The lower the value, the better the model is able to predict the value of the response variable; accompanied by a high AIC measure which conveys agreement….[…].

```r
poTest(ordinal_regressor_2)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?` +
    `V2105 Drink alcohol`, data = Research_Variables_of_interest)

                       b[polr]    b[>1]    b[>2]    b[>3]    b[>4]
Overall
`V2163 Dad Educ`      -0.06740 -0.01654 -0.11100 -0.10792 -0.08848
`V2192 Money?`         0.51179  0.65764  0.55672  0.44515  0.40830
`V2195 On dates?`      0.10012  0.16865  0.15423  0.19808  0.09740
`V2116 smoke grass?`   0.03518  0.04940  0.06840  0.02165  0.09474
`V2105 Drink alcohol`  0.02430  0.07610 -0.01886 -0.03806 -0.07853
                         b[>5]    b[>6]    b[>7] Chisquare df Pr(>Chisq)
Overall                                             112.78 30    1.6e-11 ***
`V2163 Dad Educ`      -0.09460 -0.09296 -0.10749      6.36  6      0.384
`V2192 Money?`         0.35027  0.31175  0.27226     71.62  6    1.9e-13 ***
`V2195 On dates?`      0.13157  0.13983  0.10050      7.03  6      0.318
`V2116 smoke grass?`   0.11887 -0.01310 -0.10030     14.24  6      0.027 *
`V2105 Drink alcohol` -0.07876 -0.00263  0.09352     10.47  6      0.106
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the prior summary statistics, having Pr(>Chisq) as p-values (where the decision rule is customarily 0.05), along with the t-values, observed is the "V2192 Money?" predictor or dependent variable having the most significance; much agreement with the Boruta algorithm feature importance ranking observed earlier. The other variables don't convey much significance.

```r
PseudoR2(ordinal_regressor_2, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3849.8435796    0.5250311    0.5410312
```

```r
# fit ordered logit model and store results 'ordional_regressor_2'
ordinal_regressor_3 <- polr(as.factor(`V2191 Work?`) ~`V2163 Dad Educ` +
                            `V2192 Money?` + `V2195 On dates?` +
                            `V2116 smoke grass?`,
                          data = Research_Variables_of_interest)
```

```r
ordinal_regressor_3
```

```

Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?`,
    data = Research_Variables_of_interest)

Coefficients:
    `V2163 Dad Educ`       `V2192 Money?`    `V2195 On dates?`
         -0.06471980           0.51211219           0.10218801
`V2116 smoke grass?`
          0.04733905

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.767936 2.508103 3.394319 4.125582 4.919510 5.859511 6.819514

Residual Deviance: 3826.274
AIC: 3848.274
```

```r
summary(ordinal_regressor_3)
```

```

Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?`,
    data = Research_Variables_of_interest)

Coefficients:
                        Value Std. Error t value
`V2163 Dad Educ`     -0.06472    0.03206  -2.018
`V2192 Money?`        0.51211    0.01935  26.462
`V2195 On dates?`     0.10219    0.03695   2.765
`V2116 smoke grass?`  0.04734    0.02992   1.582

Intercepts:
    Value   Std. Error t value
1|2  1.7679  0.1807     9.7817
2|3  2.5081  0.1914    13.1035
3|4  3.3943  0.2043    16.6177
4|5  4.1256  0.2140    19.2754
```

```
5|6  4.9195  0.2236    22.0051
6|7  5.8595  0.2355    24.8763
7|8  6.8195  0.2550    26.7430

Residual Deviance: 3826.274
AIC: 3848.274
```

```r
poTest(ordinal_regressor_3)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2163 Dad Educ` +
    `V2192 Money?` + `V2195 On dates?` + `V2116 smoke grass?`,
    data = Research_Variables_of_interest)

                       b[polr]    b[>1]    b[>2]    b[>3]    b[>4]
Overall
`V2163 Dad Educ`      -0.06472 -0.00901 -0.11294 -0.11115 -0.09402
`V2192 Money?`         0.51211  0.65990  0.55603  0.44399  0.40581
`V2195 On dates?`      0.10219  0.17609  0.15264  0.19529  0.09198
`V2116 smoke grass?`   0.04734  0.08858  0.05893  0.00317  0.05745
                         b[>5]    b[>6]    b[>7] Chisquare df Pr(>Chisq)
Overall                                             101.67 24    1.6e-11 ***
`V2163 Dad Educ`      -0.09997 -0.09315 -0.09923      7.62  6       0.27
`V2192 Money?`         0.34825  0.31170  0.27414     72.62  6    1.2e-13 ***
`V2195 On dates?`      0.12614  0.13965  0.10660      7.44  6       0.28
`V2116 smoke grass?`   0.08261 -0.01428 -0.05729     11.28  6       0.08 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(ordinal_regressor_3, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3848.2743996    0.5248825    0.5408781
```

```r
# fit ordered logit model and store results 'ordional_regressor_3'
ordinal_regressor_4 <- polr(as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?`
                          data = Research_Variables_of_interest)
ordinal_regressor_4
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?` +
    `V2116 smoke grass?`, data = Research_Variables_of_interest)

Coefficients:
      `V2192 Money?`     `V2195 On dates?` `V2116 smoke grass?`
          0.51081673           0.10077185           0.04898939

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
2.031973 2.771603 3.654453 4.383049 5.173318 6.109520 7.067956

Residual Deviance: 3830.361
AIC: 3850.361
```

```r
summary(ordinal_regressor_4)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?` +
    `V2116 smoke grass?`, data = Research_Variables_of_interest)

Coefficients:
                       Value Std. Error t value
`V2192 Money?`       0.51082    0.01932  26.433
`V2195 On dates?`    0.10077    0.03699   2.724
`V2116 smoke grass?` 0.04899    0.02987   1.640

Intercepts:
    Value   Std. Error t value
1|2  2.0320  0.1270    15.9989
2|3  2.7716  0.1421    19.5109
3|4  3.6545  0.1604    22.7884
4|5  4.3830  0.1737    25.2275
5|6  5.1733  0.1867    27.7066
6|7  6.1095  0.2023    30.2054
7|8  7.0680  0.2252    31.3842

Residual Deviance: 3830.361
AIC: 3850.361
```

```r
poTest(ordinal_regressor_4)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?` +
    `V2116 smoke grass?`, data = Research_Variables_of_interest)

                      b[polr]    b[>1]    b[>2]    b[>3]    b[>4]    b[>5]
Overall
`V2192 Money?`        0.51082  0.65974  0.55095  0.44046  0.40403  0.34724
`V2195 On dates?`     0.10077  0.17604  0.15157  0.19330  0.08958  0.12197
`V2116 smoke grass?`  0.04899  0.08905  0.06414  0.00779  0.06027  0.08489
                        b[>6]    b[>7] Chisquare df Pr(>Chisq)
Overall                                    93.59 18    3.3e-12 ***
`V2192 Money?`        0.31169  0.27465     72.36  6    1.3e-13 ***
`V2195 On dates?`     0.13505  0.10186      7.47  6      0.280
`V2116 smoke grass?` -0.01274 -0.05597     11.03  6      0.087 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
PseudoR2(ordinal_regressor_4, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3850.3612469    0.5234713    0.5394239
```

```
# fit ordered logit model and store results 'ordional_regressor_4'
ordinal_regressor_5 <- polr(as.factor(`V2191 Work?`) ~ `V2192 Money?` +
                              `V2195 On dates?`,
                            data = Research_Variables_of_interest)
ordinal_regressor_5
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?`,
    data = Research_Variables_of_interest)

Coefficients:
   `V2192 Money?` `V2195 On dates?`
        0.5128097         0.1044394

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.959249 2.697756 3.580166 4.308176 5.095812 6.029859 6.989251
```

```
Residual Deviance: 3833.031
AIC: 3851.031
```

```r
summary(ordinal_regressor_5)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?`,
    data = Research_Variables_of_interest)

Coefficients:
                   Value Std. Error t value
`V2192 Money?`    0.5128    0.01930  26.568
`V2195 On dates?` 0.1044    0.03694   2.827

Intercepts:
    Value   Std. Error t value
1|2  1.9592  0.1185    16.5375
2|3  2.6978  0.1342    20.1008
3|4  3.5802  0.1533    23.3526
4|5  4.3082  0.1671    25.7861
5|6  5.0958  0.1800    28.3124
6|7  6.0299  0.1956    30.8256
7|8  6.9893  0.2194    31.8619

Residual Deviance: 3833.031
AIC: 3851.031
```

```r
poTest(ordinal_regressor_5)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?`,
    data = Research_Variables_of_interest)

                  b[polr]   b[>1]   b[>2]   b[>3]   b[>4]   b[>5]   b[>6]
Overall
`V2192 Money?`     0.5128  0.6620  0.5531  0.4408  0.4060  0.3497  0.3112
`V2195 On dates?`  0.1044  0.1901  0.1595  0.1940  0.0940  0.1280  0.1341
                    b[>7] Chisquare df Pr(>Chisq)
```

```
Overall                          81.95 12    1.8e-12 ***
`V2192 Money?`      0.2726     73.53  6    7.7e-14 ***
`V2195 On dates?`   0.0976      7.17  6       0.31
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(ordinal_regressor_5, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3851.0307041    0.5225473    0.5384718
```

```r
# fit ordered logit model and store results 'ordional_regressor_5'
ordinal_regressor_6 <- polr(as.factor(`V2191 Work?`) ~ `V2192 Money?`,
                            data = Research_Variables_of_interest)
ordinal_regressor_6
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?`, data = Research_Variables_of_intere

Coefficients:
`V2192 Money?`
     0.5199097

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.783882 2.521509 3.401219 4.127180 4.914245 5.845639 6.802985

Residual Deviance: 3840.995
AIC: 3856.995
```

```r
summary(ordinal_regressor_6)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?`, data = Research_Variables_of_intere

Coefficients:
                Value Std. Error t value
`V2192 Money?` 0.5199    0.01918   27.11
```

```
Intercepts:
    Value   Std. Error t value
1|2  1.7839  0.0996    17.9169
2|3  2.5215  0.1177    21.4322
3|4  3.4012  0.1384    24.5720
4|5  4.1272  0.1530    26.9696
5|6  4.9142  0.1668    29.4651
6|7  5.8456  0.1829    31.9561
7|8  6.8030  0.2077    32.7527

Residual Deviance: 3840.995
AIC: 3856.995
```

```r
poTest(ordinal_regressor_6)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?`, data = Research_Variables_of_intere

               b[polr]  b[>1]  b[>2]  b[>3]  b[>4]  b[>5]  b[>6]  b[>7]
Overall
`V2192 Money?`   0.520  0.670  0.560  0.448  0.412  0.357  0.320  0.280
               Chisquare df Pr(>Chisq)
Overall             73.8  6    6.7e-14 ***
`V2192 Money?`      73.8  6    6.7e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(ordinal_regressor_6, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3856.9945868    0.5197800    0.5356201
```

The regression models applied prior are not the common ordinary least squared (OLS) regression with direct numerical (continuous) data; ordinal data is applied in our case. Hence the usual OLS summary statistics will not suffice. The above data frame (table) details some measures appropriate for ordinal (logistic) regression.
The **Akaike information Criterion** (**AIC**) is an estimator of predictor error and hence

identifies the relative quality of statistical models for a given set of data. For a collection of models for the data, the AIC estimates the quality of each model, relative to each of the other models for the data. Hence, the AIC is one method of model selection. The AIC is based on information theory. When a statistical model is used to represent the process that generated the data, the representation will almost never be exact; so some information will be lost by using the model to represent the process. AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. In estimating the amount of information lost by a model, AIC deals with the trade-off between the goodness of fit and the simplicity of the model. In other words, AIC deals with both the risk of overfitting and the risk of underfitting.
It's observe earlier that the summary data of the ordinal regressors didn't directly specify any other comparative models with the given AIC values. Hence, to sustain some realized validity of the given AIC measurement supplied, five ordinal regressors for statistics interpretation were directly developed. The higher the AIC, the poorer the model performance.
The **Residual Deviance**, tells how well the response variable can be predicted by a model with p predictor variables. The lower the value, the better the model is able to predict the value of the response variable.
Cox and Snell's R^2 is based on the log likelihood for the model compared to the log likelihood for a baseline model. However, with categorical outcomes, it has a theoretical maximum value of less than 1, even for a "perfect" model.
Nagelkerke's R^2 is an adjusted version of the Cox & Snell R-square that adjusts the scale of the statistic to cover the full range from 0 to 1
The **Overall P(>Chisq)** measure conveys the credibility of the model. Values below 0.05 conveys rejection of the null hypothesis[...].
Having **"V2192 Money?"** as the most significant predictor (or feature) based on the Boruta Algorthm; also confirmed by all ordinal regressors.

Using the FSelectorRccp Package for Feature Selection

Recalling the predictors chosen by the Boruta Algorithm: "V2192 Money?", "V2195 On dates?", "V2116 smoke grass?", "V2105 Drink alcohol", "2151 Race","V2163 Dad Educ". However, the third, fourth and sixth variables may neither be easily confirmed as honest nor directly influencing concerning adolescents daily lives and routines. Hence, will incorporate an alternative an alternative feature selection package called FSelectorRccp. FSelectorRccp is an Rccp (free of Java/Weka) implementation of FSelector ntropy-based feature selection algorithms with a sparse matrix support. It is also equipped with a parallel backend. Rccp is a "glue" that binds the power and versatility of R with the speed and efficiency of C++.

Of consequence, predictor candidates that can be easily confirmed as honest or identified as directly influencing the daily lives and routes of adolescents now to be preference, thus unfortunately incorporating some cognitive bias. Predictors of consideration:

"2151 Race". Described on numerous occasions prior.
"V2192 Money?". Described on numerous occasions prior.
"V2195 On dates?". Described on numerous occasions prior.
"V2174 Smart?". How intelligent do you think you are compared with others your age?
1="Far Below Average"; 2="Below Average"; 3="Slightly Below Average"; 4="Average"; 5="Slightly Above Average"; 6="Above Average"; 7="Far Above Average".
V2179 GPA. Which of the following best describes your average grade so far in high school?
9="A (93-100)"; 8="A- (90-92)"; 7="B+ (87-89)"; 6="B (83-86)"; 5="B- (80-82)"; 4="C+ (77-79)"; 3="C (73-76)"; 2="C- (70-72)"; 1="D (69 or below)"
"V2194 GO Out". During a typical week, on how many evenings do you go out for fun and recreation?
1="Less than one"; 2="One"; 3="Two"; 4="Three"; 5="Four or Five"; 6="Six or Seven".
"V2183 College?". How likely is it that you will do each of the following things after high school? Graduate from college (four-year program)
1="Definitely Won't" 2="Probably Won't" 3="Probably Will" 4="Definitely Will"
RESPONDENT_AGE. Item comprised of responses to:
Question C01: "In what year were you born?" (item 00010),
Question C02: "In what month were you born" (item 00020)
1="younger than 18" 2="18 years of age or over"

Implementing the FSelectorRccp package with the data.frame interface orientation:

```r
library(FSelectorRcpp)
```

```
Warning: package 'FSelectorRcpp' was built under R version 4.3.2
```

```r
Features_now <- Research_Variables_of_interest |>
  dplyr::select(RESPONDENT_AGE, `2151 Race`,
         `V2174 Smart?`, `V2179 GPA`, `V2183  College?`,
         `V2192 Money?`, `V2194   GO Out`, `V2195 On dates?`)
features_prospects <- information_gain(x = Features_now, y = Research_Variables_of_interes
features_prospects
```

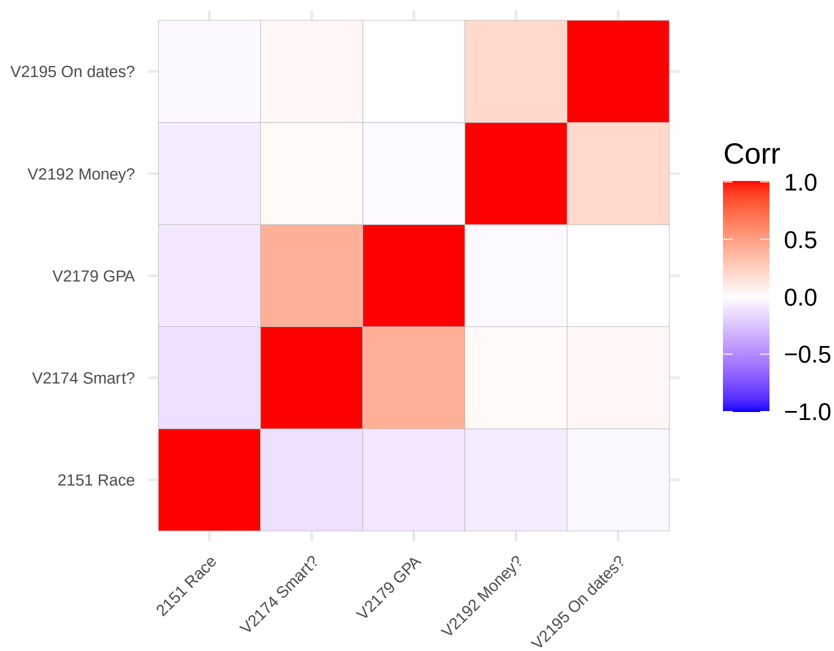
```
       attributes importance
1  RESPONDENT_AGE 0.00000000
2       2151 Race 0.01958635
3    V2174 Smart? 0.25240383
4       V2179 GPA 0.04394283
```

```
5 V2183  College? 0.00000000
6    V2192 Money? 0.39755340
7  V2194   GO Out 0.00000000
8 V2195 On dates? 0.02468373
```

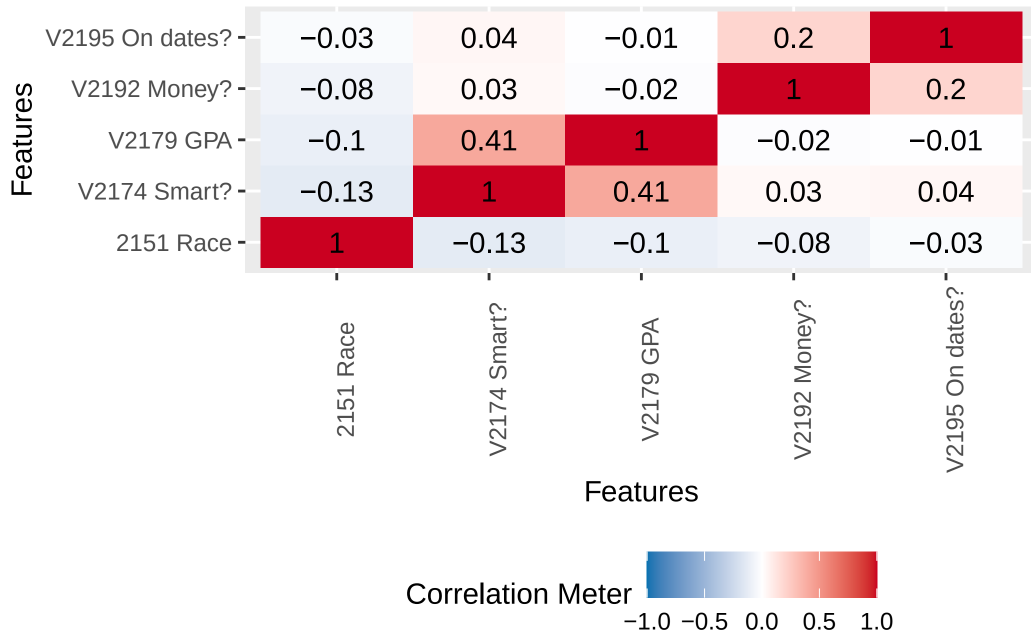
From the above results, listing results based on importance value:
1. V2192 Money?
2. V2174 Smart?
3. V2179 GPA
4. V2195 On dates?
5. 2151 Race
6. Three way tie among V2194 GO Out, V2183 College and RESPONDENT_AGE

Results forces use of 5 predictors over the preference of 6. Now pursuing correlation data:

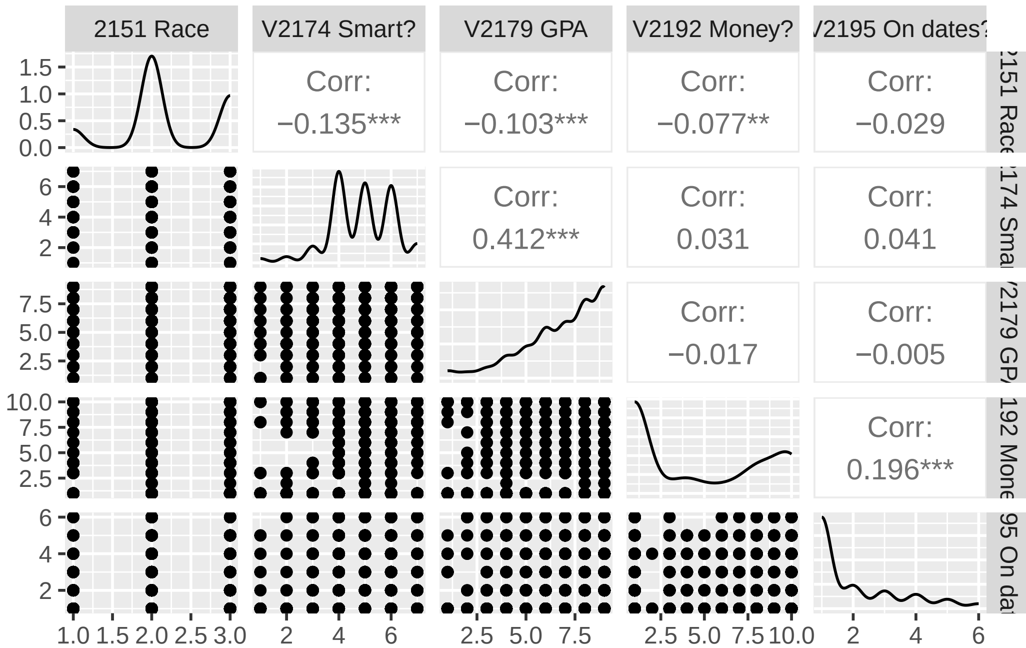
```r
FS_selected_features <- Features_now |>
  dplyr::select(`2151 Race`,
         `V2174 Smart?`, `V2179 GPA`,
         `V2192 Money?`, `V2195 On dates?`)
FS_corr_matrix <- cor(FS_selected_features)
ggcorrplot(FS_corr_matrix, tl.cex = 6)
```



```r
plot_correlation(FS_selected_features)
```




```r
ggpairs(FS_selected_features)
```



Developing summary statistics for FSelectorRccp based selected predictors:

```
summary(FS_selected_features)
```

```
   2151 Race       V2174 Smart?      V2179 GPA       V2192 Money?
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 1.000
 1st Qu.:2.000   1st Qu.:4.000   1st Qu.:6.000   1st Qu.: 1.000
 Median :2.000   Median :5.000   Median :7.000   Median : 3.000
 Mean   :2.209   Mean   :4.848   Mean   :6.921   Mean   : 4.493
 3rd Qu.:3.000   3rd Qu.:6.000   3rd Qu.:9.000   3rd Qu.: 8.000
 Max.   :3.000   Max.   :7.000   Max.   :9.000   Max.   :10.000
 V2195 On dates?
 Min.   :1.000
 1st Qu.:1.000
 Median :1.000
 Mean   :2.012
 3rd Qu.:3.000
 Max.   :6.000
```

Identifying the skewness values:

```r
skewness(FS_selected_features$`2151 Race`)
```

```
[1] -0.183889
```

```r
skewness(FS_selected_features$`V2174 Smart?`)
```

```
[1] -0.4591146
```

```r
skewness(FS_selected_features$`V2179 GPA`)
```

```
[1] -0.9788332
```

```r
skewness(FS_selected_features$`V2192 Money?`)
```

```
[1] 0.364167
```

```r
skewness(FS_selected_features$`V2195 On dates?`)
```

```
[1] 1.26259
```

Identifying the Kurtosis values:

```r
kurtosis(FS_selected_features$`2151 Race`)
```

```
[1] 2.411048
```

```r
kurtosis(FS_selected_features$`V2174 Smart?`)
```

```
[1] 3.410156
```

```r
kurtosis(FS_selected_features$`V2179 GPA`)
```

```
[1] 3.394998
```

```r
kurtosis(FS_selected_features$`V2192 Money?`)
```

```
[1] 1.370079
```

```r
kurtosis(FS_selected_features$`V2195 On dates?`)
```

```
[1] 3.450513
```

Ordinal Regression Analysis based on FSelectorRCCp Feature Selection

```r
FS_ordinal_regressor_1 <- polr(as.factor(`V2191 Work?`) ~ `2151 Race` +
                                `V2174 Smart?` + `V2179 GPA` +
                                `V2192 Money?` + `V2195 On dates?`,
                              data = Research_Variables_of_interest)
FS_ordinal_regressor_1
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `2151 Race` + `V2174 Smart?` +
    `V2179 GPA` + `V2192 Money?` + `V2195 On dates?`, data = Research_Variables_of_interest)

Coefficients:
      `2151 Race`    `V2174 Smart?`       `V2179 GPA`    `V2192 Money?`
      -0.16769512       -0.05095261       -0.02364925        0.51252574
`V2195 On dates?`
       0.10477092

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.172351 1.911391 2.796106 3.527361 4.320499 5.259381 6.219551

Residual Deviance: 3827.507
AIC: 3851.507
```

```r
summary(FS_ordinal_regressor_1)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `2151 Race` + `V2174 Smart?` +
    `V2179 GPA` + `V2192 Money?` + `V2195 On dates?`, data = Research_Variables_of_interest)

Coefficients:
                    Value Std. Error t value
`2151 Race`       -0.16770    0.09097 -1.8434
`V2174 Smart?`    -0.05095    0.05002 -1.0186
`V2179 GPA`       -0.02365    0.03022 -0.7825
`V2192 Money?`     0.51253    0.01936 26.4696
`V2195 On dates?`  0.10477    0.03699  2.8323

Intercepts:
    Value   Std. Error t value
1|2  1.1724  0.3552     3.3010
2|3  1.9114  0.3605     5.3014
3|4  2.7961  0.3669     7.6207
4|5  3.5274  0.3713     9.4988
5|6  4.3205  0.3748    11.5274
6|7  5.2594  0.3801    13.8357
7|8  6.2196  0.3922    15.8564

Residual Deviance: 3827.507
AIC: 3851.507
```

```r
poTest(FS_ordinal_regressor_1)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `2151 Race` + `V2174 Smart?` +
    `V2179 GPA` + `V2192 Money?` + `V2195 On dates?`, data = Research_Variables_of_interest)

                   b[polr]     b[>1]     b[>2]     b[>3]     b[>4]     b[>5]
Overall
`2151 Race`       -0.16770  -0.23301  -0.20156  -0.20617  -0.21020  -0.15782
`V2174 Smart?`    -0.05095   0.01739  -0.12377  -0.17555  -0.07595  -0.15385
`V2179 GPA`       -0.02365   0.04815   0.03006   0.00032  -0.07486  -0.13481
`V2192 Money?`     0.51253   0.65870   0.55446   0.44497   0.40694   0.35278
`V2195 On dates?`  0.10477   0.18483   0.15957   0.19517   0.09106   0.12752
                     b[>6]     b[>7] Chisquare df Pr(>Chisq)
Overall                               140.48 30    3.2e-16 ***
```

```
`2151 Race`         -0.06222   0.22972      4.24  6      0.644
`V2174 Smart?`      -0.03337  -0.07392     15.41  6      0.017 *
`V2179 GPA`         -0.09665  -0.07102     15.77  6      0.015 *
`V2192 Money?`       0.30778   0.27186     72.54  6    1.2e-13 ***
`V2195 On dates?`    0.13194   0.09613      7.46  6      0.281
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(FS_ordinal_regressor_1, c("AIC", "CoxSnell", "Nagel"))
```

```
          AIC      CoxSnell    Nagelkerke
3851.5074689    0.5244572     0.5404398
```

```r
FS_ordinal_regressor_2 <- polr(as.factor(`V2191 Work?`) ~
                                `V2174 Smart?` +
                                `V2179 GPA` + `V2192 Money?` +
                                `V2195 On dates?`,
                         data = Research_Variables_of_interest)
FS_ordinal_regressor_2
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2174 Smart?` + `V2179 GPA` +
    `V2192 Money?` + `V2195 On dates?`, data = Research_Variables_of_interest)

Coefficients:
   `V2174 Smart?`       `V2179 GPA`    `V2192 Money?` `V2195 On dates?`
      -0.04201254       -0.02197551        0.51395540        0.10491858

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.603670 2.340900 3.224704 3.955186 4.747691 5.686893 6.648070

Residual Deviance: 3830.898
AIC: 3852.898
```

```r
poTest(FS_ordinal_regressor_2)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2174 Smart?` + `V2179 GPA` +
    `V2192 Money?` + `V2195 On dates?`, data = Research_Variables_of_interest)

                   b[polr]    b[>1]    b[>2]    b[>3]    b[>4]    b[>5]
Overall
`V2174 Smart?`     -0.0420   0.0301  -0.1153  -0.1678  -0.0681  -0.1478
`V2179 GPA`        -0.0220   0.0519   0.0343   0.0046  -0.0704  -0.1320
`V2192 Money?`      0.5140   0.6609   0.5559   0.4461   0.4077   0.3536
`V2195 On dates?`   0.1049   0.1876   0.1630   0.1978   0.0923   0.1278
                     b[>6]    b[>7] Chisquare df Pr(>Chisq)
Overall                               136.69 24    < 2e-16 ***
`V2174 Smart?`     -0.0306  -0.0832    16.12  6      0.013 *
`V2179 GPA`        -0.0955  -0.0764    16.08  6      0.013 *
`V2192 Money?`      0.3082   0.2697    73.36  6    8.3e-14 ***
`V2195 On dates?`   0.1316   0.0966     7.53  6      0.274
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(FS_ordinal_regressor_2, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3852.8983345    0.5232856    0.5392325
```

```r
FS_ordinal_regressor_3 <- polr(as.factor(`V2191 Work?`) ~
                                  `V2179 GPA` + `V2192 Money?` +
                                  `V2195 On dates?`,
                               data = Research_Variables_of_interest)
FS_ordinal_regressor_3
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2179 GPA` + `V2192 Money?` +
    `V2195 On dates?`, data = Research_Variables_of_interest)

Coefficients:
      `V2179 GPA`    `V2192 Money?` `V2195 On dates?`
      -0.03267002        0.51326928        0.10425695

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
```

```
1.729829 2.466901 3.349707 4.079595 4.871373 5.809828 6.771209

Residual Deviance: 3831.61
AIC: 3851.61
```

```r
summary(FS_ordinal_regressor_3)
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2179 GPA` + `V2192 Money?` +
    `V2195 On dates?`, data = Research_Variables_of_interest)

Coefficients:
                     Value Std. Error t value
`V2179 GPA`       -0.03267    0.02733  -1.195
`V2192 Money?`     0.51327    0.01931  26.575
`V2195 On dates?`  0.10426    0.03696   2.821

Intercepts:
    Value   Std. Error t value
1|2  1.7298  0.2245     7.7065
2|3  2.4669  0.2341    10.5388
3|4  3.3497  0.2452    13.6610
4|5  4.0796  0.2528    16.1374
5|6  4.8714  0.2589    18.8123
6|7  5.8098  0.2674    21.7294
7|8  6.7712  0.2840    23.8388

Residual Deviance: 3831.61
AIC: 3851.61
```

```r
poTest(FS_ordinal_regressor_3)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2179 GPA` + `V2192 Money?` +
    `V2195 On dates?`, data = Research_Variables_of_interest)

                    b[polr]     b[>1]     b[>2]     b[>3]     b[>4]     b[>5]
Overall
`V2179 GPA`        -0.03267   0.05966   0.00484  -0.03714  -0.08682  -0.16569
```

```
`V2192 Money?`       0.51327   0.66080   0.55301   0.44096   0.40633   0.35012
`V2195 On dates?`    0.10426   0.18874   0.15953   0.19316   0.09077   0.12402
                      b[>6]     b[>7] Chisquare df Pr(>Chisq)
Overall                                  117.9 18    < 2e-16 ***
`V2179 GPA`         -0.10239  -0.09493      24.9  6    0.00036 ***
`V2192 Money?`       0.30774   0.26854      74.9  6    4.1e-14 ***
`V2195 On dates?`    0.13084   0.09416       7.3  6    0.29400
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(FS_ordinal_regressor_3, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3851.6096505    0.5230394    0.5389789
```

```r
FS_ordinal_regressor_4 <- polr(as.factor(`V2191 Work?`) ~
                                 `V2192 Money?` +
                                 `V2195 On dates?`,
                               data = Research_Variables_of_interest)
FS_ordinal_regressor_4
```

```
Call:
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?`,
    data = Research_Variables_of_interest)

Coefficients:
   `V2192 Money?` `V2195 On dates?`
        0.5128097         0.1044394

Intercepts:
     1|2      2|3      3|4      4|5      5|6      6|7      7|8
1.959249 2.697756 3.580166 4.308176 5.095812 6.029859 6.989251

Residual Deviance: 3833.031
AIC: 3851.031
```

```r
poTest(FS_ordinal_regressor_4)
```

```
Tests for Proportional Odds
polr(formula = as.factor(`V2191 Work?`) ~ `V2192 Money?` + `V2195 On dates?`,
    data = Research_Variables_of_interest)

                  b[polr]   b[>1]   b[>2]   b[>3]   b[>4]   b[>5]   b[>6]
Overall
`V2192 Money?`     0.5128  0.6620  0.5531  0.4408  0.4060  0.3497  0.3112
`V2195 On dates?`  0.1044  0.1901  0.1595  0.1940  0.0940  0.1280  0.1341
                    b[>7] Chisquare df Pr(>Chisq)
Overall                       81.95 12    1.8e-12 ***
`V2192 Money?`     0.2726     73.53  6    7.7e-14 ***
`V2195 On dates?`  0.0976      7.17  6       0.31
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
PseudoR2(FS_ordinal_regressor_4, c("AIC", "CoxSnell", "Nagel"))
```

```
         AIC     CoxSnell   Nagelkerke
3851.0307041    0.5225473    0.5384718
```

Tables to Comparatively Evaluate Ordinal Logistic Regression Models

```r
# Boruta based
AIC <- c(3847.991, 3849.844, 3848.274, 3850.361, 3851.031, 3856.995)
Resid_D <- c(3821.991, 3825.844, 3826.274, 3830.361, 3833.031,
                        3840.995)
CoxSnell <- c(0.5263572, 0.5250311, 0.5248825, 0.5234713, 0.5225473,
              0.5197800)
Nagelkerke <- c(0.5423977, 0.5410312, 0.5408781, 0.5394239, 0.5384718,
                0.5356201)
Overall_Pr_Chisq <- c(1.5*10**-10, 1.6*10**-11, 1.6*10**-11, 3.3*10**-12,
                      1.8*10**-12, 6.7*10**-14)
Money_Significant_predictor <- c(2.7*10**-13, 1.9*10**-13, 1.2*10**-13, 1.3*10**-13,
                           7.7*10**-14, 6.7*10**-14)

# Create a data frame
df <- data.frame(
  Regressor = c("OLR_1", "OLR_2", "OLR_3",
                "OLR_4", "OLR_5",
```

```r
                   "OLR_6"),
  AIC = AIC,
  Residual_Deviance = Resid_D,
  CoxSnell = CoxSnell,
  Nagelkerke = Nagelkerke,
  Overall_Pr_Chisq = Overall_Pr_Chisq,
  Money_Significant_predictor = Money_Significant_predictor
)

# Set the data frame name
attr(df, "name") <- "Results for Ordinal Logistic Regression Model Results"

# Print the data frame
print(df)
```

```
  Regressor      AIC Residual_Deviance  CoxSnell Nagelkerke Overall_Pr_Chisq
1     OLR_1 3847.991          3821.991 0.5263572  0.5423977          1.5e-10
2     OLR_2 3849.844          3825.844 0.5250311  0.5410312          1.6e-11
3     OLR_3 3848.274          3826.274 0.5248825  0.5408781          1.6e-11
4     OLR_4 3850.361          3830.361 0.5234713  0.5394239          3.3e-12
5     OLR_5 3851.031          3833.031 0.5225473  0.5384718          1.8e-12
6     OLR_6 3856.995          3840.995 0.5197800  0.5356201          6.7e-14
  Money_Significant_predictor
1                     2.7e-13
2                     1.9e-13
3                     1.2e-13
4                     1.3e-13
5                     7.7e-14
6                     6.7e-14
```

```r
# FSelectorRccp based
AIC <- c(3851.507, 3857.898, 3851.61, 3851.031)
Residual_D <- c(3827.507, 3830.898, 3831.61, 3833.031)
CoxSnell <- c(0.5244572, 0.5232856, 0.5230394, 0.5225473)
Nagelkerke <- c(0.5404398, 0.5392325, 0.5389789, 0.5384718)
Overall_Pr_Chisq <- c(3.2*10**-16, 2.0*10**-16,
                      2.0*10**-16, 1.8*10**-12)
Money_Significant_predictor <- c(1.2*10**-13, 8.3*10**-14,
                                 4.1*10**-14, 7.7*10**-14)

# Create a data frame
```

```r
df <- data.frame(
  Regressor = c("FS_OLR_1", "FS_OLR_2", "FS_OLR_3",
                "FS_OLR_4"),
  AIC = AIC,
  Residual_Deviance = Residual_D,
  CoxSnell = CoxSnell,
  Nagelkerke = Nagelkerke,
  Overall_Pr_Chisq = Overall_Pr_Chisq,
  Money_Significant_predictor = Money_Significant_predictor
)

# Set the data frame name
attr(df, "name") <- "Results for Ordinal Logistic Regression Model Results"

# Print the data frame
print(df)
```

```
  Regressor      AIC Residual_Deviance  CoxSnell Nagelkerke Overall_Pr_Chisq
1  FS_OLR_1 3851.507          3827.507 0.5244572  0.5404398          3.2e-16
2  FS_OLR_2 3857.898          3830.898 0.5232856  0.5392325          2.0e-16
3  FS_OLR_3 3851.610          3831.610 0.5230394  0.5389789          2.0e-16
4  FS_OLR_4 3851.031          3833.031 0.5225473  0.5384718          1.8e-12
  Money_Significant_predictor
1                     1.2e-13
2                     8.3e-14
3                     4.1e-14
4                     7.7e-14
```

Categorical Variables and the Chi-Square Test of Independence

For the Chi-square Test of Independence we have the following hypothesis test:

$$H_0 : There\ Is\ No\ Significant\ Association\ Between\ The\ Categorical\ Variables$$

$$H_1 :\ There\ Is\ No\ significant\ Association\ Between\ The\ Categorical\ Variables$$

$$Decision\ Rule :\ for\ \alpha = 0.05,\ reject\ H_0\ if\ p - value <\ \alpha$$

Recalling the dependent variable or response variable to be "V2191 Work?" with the following structure:
On the average over the school year, how many hours per week do you work in a paid or unpaid job?
1 = "None"
2 = "5 or less hours"
3 = "6 to 10 hours"
4 = "11 to 15 hours"
5 = "16 to 20 hours"
6 = "21 to 25 hours"
7 = "26 to 30 hours"
8 = "More than 30 hours"
Observed is a ordinal variable with ranked ordering with no middle level. Will now transform this response variable into a categorical form by applying the categorical responses instead of the response numbering.

```r
# Create a vector to specify the levels and labels
levels <- c("None", "5 or less hours", "6 to 10 hours",
            "11 to 15 hours", "16 to 20 hours",
            "21 to 25 hours", "26 to 30 hours", "More than 30 hours")

# Transform the variable to a factor
V2191_Work_Categorical <- factor(Research_Variables_of_interest$`V2191 Work?`,
                                 levels = 1:8, labels = levels,
                                 ordered = TRUE)

# Now, 'V2191_Work_Categorical' is a categorical variable with the specified labels
```

```r
head(V2191_Work_Categorical)
```

```
[1] More than 30 hours 21 to 25 hours     5 or less hours    None
[5] None               6 to 10 hours
8 Levels: None < 5 or less hours < 6 to 10 hours < ... < More than 30 hours
```

Recalling the selected independent variables or predictors identified as categorical since no rank can be observed –

"2151 Race" is based on the following survey question: How do you describe yourself?
Select one or more responses: Black or African American; Mexican American or Chicano; Cuban American; Puerto Rican; Other Hispanic or Latino; Asian American; White (Caucasian); American Indian or Alaska Native; Native Hawaiian or Other Pacific Islander; Middle Eastern.

Recoded in this dataset so that "Black or African American" = 1, "White (Caucasian)" = 2; Hispanic = 3 ("Mexican…" or "Cuban…" or "Puerto Rican" or "Other Hispanic…"). Observed is categorical values since values in this case can't be ranked.

For both predictor variables will also transform into categorical forms by applying the categorical responses instead of the response numbering.

```
# For "V2151 Race" creating a vector to specify the levels and labels
levels <- c("Black or African American", "White (Caucasian)", "Hispanic")

# Transform the variable to a factor
Race_Categorical <- factor(Research_Variables_of_interest$`2151 Race`,
                           levels = 1:3, labels = levels,
                           ordered = TRUE)

# Now, 'Race_Categorical' is a (EXPLICIT) categorical variable with the specified labels.
```

```
head(Race_Categorical)
```

```
[1] White (Caucasian) White (Caucasian) Hispanic          Hispanic
[5] Hispanic          White (Caucasian)
Levels: Black or African American < White (Caucasian) < Hispanic
```

Now, the Chi-Square Test of Independence is used to determine if there's an association between two categorical variables; assessing whether the observed frequencies in a contingency table are significantly unique from the expected frequencies under the assumption of independence.

```
# Creating a contingency table
Race_contingency_table <- table(V2191_Work_Categorical,
                                Race_Categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Race_contingency_table)
```

```
chi_square_test
```

```

	Pearson's Chi-squared test

data:  Race_contingency_table
X-squared = 70.423, df = 14, p-value = 1.618e-09
```

Alternatively, the Fisher Exact Test is a statistical significance test used in the analysis of contingency tables. It can be used to examine the significance of the association (contingency) between the two kinds of classification.

```
Race_test <- fisher.test(Race_contingency_table,
                         simulate.p.value=TRUE)
Race_test
```

```
	Fisher's Exact Test for Count Data with simulated p-value (based on
	2000 replicates)

data:  Race_contingency_table
p-value = 0.0004998
alternative hypothesis: two.sided
```

Similar to what was done with the dependent variable based on the codebook, the ordinal variables can be transformed based on the applied categorical instances for each variable. Of consequence, it's also possible to apply all selected independent variables (predictors) to the chi-square test of independence.

"V2192 Money?" is based on the following survey question: During an average week, how much money did you get from . . . a job or other work?
1="None"; 2="$1-5"; 3="$6-10"; 4="$11-20"; 5=$21-35"; 6="$36-50"; 7="$51-75"; 8="$76-125"; 9="$126-175"; 10="176+"
Observed is ordinal values since values can be ranked.

```
# For "V2192 Money?" creating a vector to specify the levels and labels
levels <- c("None", "$1-5", "$6-10", "$11-20", "$21-35",
            "$36-50", "$51-75", "$76-125", "$126-175", "$176+")

# Transform the variable to a factor
Money_Categorical <- factor(Research_Variables_of_interest$`V2192 Money?`,
                            levels = 1:10, labels = levels,
                            ordered = TRUE)

# Now, 'Money_Categorical' is a categorical variable with the specified labels.
```

```
head(Money_Categorical)
```

```
[1] $176+  None   None   None   None   $21-35
10 Levels: None < $1-5 < $6-10 < $11-20 < $21-35 < $36-50 < ... < $176+
```

```r
# Creating a contingency table
Money_contingency_table <- table(V2191_Work_Categorical,
                                 Money_Categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Money_contingency_table)
```

```
Warning in chisq.test(Money_contingency_table): Chi-squared approximation may
be incorrect
```

Above warning message is due to the small cell values in the contingency table. Resorting to Fisher's Exact Test. The hypotheses of the Fisher's Exact Test are the same than for the Chi-Square Test of Independence.

```r
Money_test <- fisher.test(Money_contingency_table,
                          simulate.p.value=TRUE)
Money_test
```

```

	Fisher's Exact Test for Count Data with simulated p-value (based on
	2000 replicates)

data:  Money_contingency_table
p-value = 0.0004998
alternative hypothesis: two.sided
```

"V2195 On dates?" is based on the following survey question: On the average, how often do you go out with a date (or your spouse/partner, if you are married)?
1="Never"; 2="Once a month or less"; 3="2 or 3 times a month"; 4="Once a week"; 5="2 or 3 times a week"; 6="Over 3 times a week".
Observed is ordinal values since values can be ranked.

```r
# For "V2195 On dates?" creating a vector to specify the levels and labels
levels <- c("Never", "Once a month or less", "2 or 3 times a month",
            "Once a week", "2 or 3 times a week",
            "Over 3 times a week")

# Transform the variable to a factor
Dates_Categorical <- factor(Research_Variables_of_interest$`V2195 On dates?`,
                            levels = 1:6, labels = levels,
                            ordered = TRUE)
```

```r
# Now, 'Dates_Categorical' is a categorical variable with the specified labels.
```

```r
head(Dates_Categorical)
```

```
[1] Once a month or less Over 3 times a week  Once a month or less
[4] 2 or 3 times a month Once a week          Never
6 Levels: Never < Once a month or less < ... < Over 3 times a week
```

```r
# Creating a contingency table
Dates_contingency_table <- table(V2191_Work_Categorical,
                                 Dates_Categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Dates_contingency_table)
```

```
Warning in chisq.test(Dates_contingency_table): Chi-squared approximation may
be incorrect
```

Resorting to Fisher's Exact Test.

```r
Dates_test <- fisher.test(Dates_contingency_table, simulate.p.value=TRUE)
Dates_test
```

```

	Fisher's Exact Test for Count Data with simulated p-value (based on
	2000 replicates)

data:  Dates_contingency_table
p-value = 0.0004998
alternative hypothesis: two.sided
```

"V2116 smoke grass?" is based on the following questioning structure:
Form 1: On how many occasions (if any) have you used marijuana [sometimes called: Weed, Pot, Dope] or hashish [sometimes called: Hash, Hash oil]. . . during the last 12 months? [Separate questions for marijuana (Item 02080) and hashish (Item 02050) are combined in this variable for form 1.]
Form 3: On how many occasions (if any) have you used marijuana (weed, pot) or hashish (hash, hash oil) (Do NOT count any use of CBD products) . . . during the last 12 months?
Form 5: On how many occasions (if any) have you used marijuana (weed, pot) or hashish

(hash, hash oil). . . .during the last 12 months?
Forms 2, 4, and 6: On how many occasions (if any) have you used marijuana in any form (e.g. smoking, vaping, edibles, hashish, hash oil). . . during the last 12 months?
1="0 Occasions"; 2="1-2 Occasions"; 3="3-5 Occasions"; 4="6-9 Occasions"; 5="10-19 Occasions"; 6="20-39 Occasions"; 7="40 or More".
Observed is ordinal values since values can be ranked.

```
# For "V2116 smoke grass?" creating a vector to specify the levels and labels
levels <- c("0 Occassions", "1-2 Occassions", "3-5 Occasions",
            "6-9 Occasions", "10-19 Occasions",
            "20-39 Occasions", "40 or More")

# Transform the variable to a factor
Smoke_grass_Categorical <- factor(Research_Variables_of_interest$`V2116 smoke grass?`,
                              levels = 1:7, labels = levels,
                              ordered = TRUE)

# Now, 'Smoke_grass_Categorical' is a categorical variable with the specified labels.
```

```
head(Smoke_grass_Categorical)
```

```
[1] 0 Occassions   1-2 Occassions 1-2 Occassions 0 Occassions   0 Occassions
[6] 0 Occassions
7 Levels: 0 Occassions < 1-2 Occassions < 3-5 Occasions < ... < 40 or More
```

```
# Creating a contingency table
Smoke_grass_contingency_table <- table(V2191_Work_Categorical, Smoke_grass_Categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Smoke_grass_contingency_table)
```

```
Warning in chisq.test(Smoke_grass_contingency_table): Chi-squared approximation
may be incorrect
```

Resorting to Fisher's Exact Test.

```
Smoke_grass_test <- fisher.test(Smoke_grass_contingency_table,
                                simulate.p.value=TRUE)
Smoke_grass_test
```

```
    Fisher's Exact Test for Count Data with simulated p-value (based on
    2000 replicates)

data:  Smoke_grass_contingency_table
p-value = 0.0004998
alternative hypothesis: two.sided
```

"V2105 Drink alcohol" is based on the following survey question: On how many occasions (if any) have you had alcoholic beverages to drink–more than just a few sips . . . during the last 12 months?
On how many occasions (if any) have you had alcoholic beverages to drink--more than just a few sips . . . during the last 12 months?
1="0 Occasions"; 2="1-2 Occasions"; 3="3-5 Occasions"; 4="6-9 Occasions"; 5="10-19 Occasions"; 6="20-39 Occasions"; 7="40 or More".
Observed is ordinal values since values can be ranked.

```r
# For "V2105 Drink alcohol" creating a vector to specify the levels and labels
levels <- c("0 Occassions", "1-2 Occassions", "3-5 Occasions",
            "6-9 Occasions", "10-19 Occasions",
            "20-39 Occasions", "40 or More")

# Transform the variable to a factor
Drink_alcohol_Categorical <- factor(Research_Variables_of_interest$`V2105 Drink alcohol`,
                              levels = 1:7, labels = levels,
                              ordered = TRUE)

# Now, 'Drink_alcohol_Categorical' is a categorical variable with the specified labels.
```

```r
head(Drink_alcohol_Categorical)
```

```
[1] 6-9 Occasions  6-9 Occasions  1-2 Occassions 0 Occassions   1-2 Occassions
[6] 0 Occassions
7 Levels: 0 Occassions < 1-2 Occassions < 3-5 Occasions < ... < 40 or More
```

```r
# Creating a contingency table
Drink_alcohol_contingency_table <- table(V2191_Work_Categorical,
                                         Drink_alcohol_Categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Drink_alcohol_contingency_table)
```

```
Warning in chisq.test(Drink_alcohol_contingency_table): Chi-squared
approximation may be incorrect
```

Resorting to Fisher's Exact Test.

```
Drink_alcohol_test <- fisher.test(Drink_alcohol_contingency_table,
                                  simulate.p.value=TRUE)
Drink_alcohol_test
```

```
	Fisher's Exact Test for Count Data with simulated p-value (based on
	2000 replicates)

data:  Drink_alcohol_contingency_table
p-value = 0.0004998
alternative hypothesis: two.sided
```

"V2163 Dad Educ" is based on the following questioning structure:
The next three questions ask about your parents. If you were raised mostly by foster parents, stepparents, or others, answer for them. For example, if you have both a stepfather and a natural father, answer for the one that was the most important in raising you. What is the highest level of schooling your father completed?
1="Completed grade school or less"; 2="Some high school"; 3="Completed high school"; 4="Some college"; 5="Completed college"; 6="Graduate or professional school after college"; 7="Don't know, or does not apply".
Observed is ordinal values since values can be ranked.

```
# For "V2163 Dad Educ" creating a vector to specify the levels and labels
levels <- c("Completed grade school or less",
            "Some high school", "Completed high school",
            "Some college", "Completed College",
            "Graduate or professional school after college",
            "Don't know or does not apply")

# Transform the variable to a factor
Dad_Educ_categorical <- factor(Research_Variables_of_interest$`V2163 Dad Educ`,
                            levels = 1:7, labels = levels,
                            ordered = TRUE)

# Now, 'Dad_Educ_Categorical' is a categorical variable with the specified labels.
```

```r
head(Dad_Educ_categorical)
```

```
[1] Completed high school
[2] Completed College
[3] Some college
[4] Some college
[5] Some college
[6] Graduate or professional school after college
7 Levels: Completed grade school or less < ... < Don't know or does not apply
```

```r
# Creating a contingency table
Dad_Educ_contingency_table <- table(V2191_Work_Categorical,
                                    Dad_Educ_categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Dad_Educ_contingency_table)
```

```
Warning in chisq.test(Dad_Educ_contingency_table): Chi-squared approximation
may be incorrect
```

Resorting to Fisher's Exact Test.

```r
Dad_Educ_test <- fisher.test(Dad_Educ_contingency_table,
                             simulate.p.value=TRUE)
Dad_Educ_test
```

```
	Fisher's Exact Test for Count Data with simulated p-value (based on
	2000 replicates)

data:  Dad_Educ_contingency_table
p-value = 0.004498
alternative hypothesis: two.sided
```

"V2174 Smart?". How intelligent do you think you are compared with others your age? 1="Far Below Average"; 2="Below Average"; 3="Slightly Below Average"; 4="Average"; 5="Slightly Above Average"; 6="Above Average"; 7="Far Above Average".

```r
# For "V2174 Smart?" creating a vector to specify the levels and labels
levels <- c("Far Below Average",
            "Below Average", "Slightly Below Average",
            "Average", "Slightly Abover Average",
            "Above Average",
            "Far Above Average")

# Transform the variable to a factor
Smart_categorical <- factor(Research_Variables_of_interest$`V2174 Smart?`,
                           levels = 1:7, labels = levels,
                           ordered = TRUE)

# Now, 'Smart_Categorical' is a categorical variable with the specified labels.
```

```r
head(Smart_categorical)
```

```
[1] Far Above Average Far Above Average Far Above Average Far Above Average
[5] Far Above Average Far Above Average
7 Levels: Far Below Average < Below Average < ... < Far Above Average
```

```r
# Creating a contingency table
Smart_contingency_table <- table(V2191_Work_Categorical,
                                 Smart_categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Smart_contingency_table)
```

```
Warning in chisq.test(Smart_contingency_table): Chi-squared approximation may
be incorrect
```

Resorting to Fisher's Exact Test

```r
Smart_test <- fisher.test(Smart_contingency_table,
                          simulate.p.value=TRUE)
Smart_test
```

```

    Fisher's Exact Test for Count Data with simulated p-value (based on
    2000 replicates)
```

```
data:  Smart_contingency_table
p-value = 0.02899
alternative hypothesis: two.sided
```

V2179 GPA. Which of the following best describes your average grade so far in high school? 9="A (93-100)"; 8="A- (90-92)"; 7="B+ (87-89)"; 6="B (83-86)"; 5="B- (80-82)"; 4="C+ (77-79)"; 3="C (73-76)"; 2="C- (70-72)"; 1="D (69 or below)"

```r
# For "V2179 GPA" creating a vector to specify the levels and labels
levels <- c("9 = A (93-100)",
            "8 = A- (90-92)", "7 = B+ (87-89)",
            "6 = B (83-86)", "5 = B- (80-82)",
            "4 = C+ (77-79)",
            "3 = C (73-76)", "2 = C- (70-72)", "1 = D (69 or below")

# Transform the variable to a factor
GPA_categorical <- factor(Research_Variables_of_interest$`V2179 GPA`,
                          levels = 1:9, labels = levels,
                          ordered = TRUE)

# Now, 'GPA_Categorical' is a categorical variable with the specified labels.
```

```r
head(GPA_categorical)
```

```
[1] 2 = C- (70-72)     1 = D (69 or below 6 = B (83-86)      4 = C+ (77-79)
[5] 3 = C (73-76)      2 = C- (70-72)
9 Levels: 9 = A (93-100) < 8 = A- (90-92) < ... < 1 = D (69 or below
```

```r
# Creating a contingency table
GPA_contingency_table <- table(V2191_Work_Categorical,
                               GPA_categorical)
# Performing the chi-square test of independence
chi_square_test <- chisq.test(Smart_contingency_table)
```

```
Warning in chisq.test(Smart_contingency_table): Chi-squared approximation may
be incorrect
```

```r
GPA_test <- fisher.test(GPA_contingency_table,
                             simulate.p.value=TRUE)
GPA_test
```

```
	Fisher's Exact Test for Count Data with simulated p-value (based on
	2000 replicates)

data:  GPA_contingency_table
p-value = 0.005497
alternative hypothesis: two.sided
```

Creating a data frame (or chart) for the Test of Independence.

```r
Model <- c("Chi2 = 70.423", "Fisher Exact",
           "Fisher Exact",
           "Fisher Exact",
           "Fisher Exact",
           "Fisher Exact",
           "Fisher Exact",
           "Fisher Exact",
           "Fisher Exact")
Parameter <- c("df =14", "NA",
               "NA",
           "NA",
           "NA",
           "NA",
           "NA",
           "NA",
           "NA")
P_value <- c(1.618*10**-09, 0.0004998, 0.0004998, 0.0004998, 0.0004998,
             0.0004998, 0.004498, 0.03548, 0.003998)

# Create a data frame
df <- data.frame(
  Predictor = c("Race", "Race (Fisher)", "Money",
                "Dates", "Smoking Grass",
                "Alcohol", "Dad Educ", "Smart", "GPA"),
  Model = Model,
  Parameter = Parameter,
  P_value = P_value)
```

```r
# Set the data frame name
attr(df, "name") <- "Test of Independence with Work Response Variable"

# Print the data frame
print(df)
```

```
      Predictor         Model Parameter   P_value
1          Race Chi2 = 70.423    df =14 1.618e-09
2 Race (Fisher)  Fisher Exact        NA 4.998e-04
3         Money  Fisher Exact        NA 4.998e-04
4         Dates  Fisher Exact        NA 4.998e-04
5 Smoking Grass  Fisher Exact        NA 4.998e-04
6       Alcohol  Fisher Exact        NA 4.998e-04
7      Dad Educ  Fisher Exact        NA 4.498e-03
8         Smart  Fisher Exact        NA 3.548e-02
9           GPA  Fisher Exact        NA 3.998e-03
```

REFERENCES

1. Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & B. F. Csaki (Eds.),*Second International Symposium on Information Theory*, (pp. 267–281). Academiai Kiado: Budapest.
2. Brant, R. (1990). Assessing Proportionality in the Proportional Odds Model for Ordinal Logistic Regression. *Biometrics*, *46*(4), 1171–1178.
3. Cox, D. R., and E. J. Snell. (1989). *The Analysis of Binary Data*, 2nd ed. London: Chapman and Hall.
4. Fisher, R. A. (1922). On the Interpretation of ^{2} from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society*, *85*(1), 87–94.
5. Fisher, R.A. (1992). Statistical Methods for Research Workers. In: Kotz, S., Johnson, N.L. (eds) Breakthroughs in Statistics. Springer Series in Statistics. Springer, New York, NY
6. McCullagh, P. (1980). Regression Models for Ordinal Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, *42*(2), 109–142.
7. Miech, R. A., Johnston, L. D., Bachman, J. G., O'Malley, P. M., Schulenberg, J. E., & Patrick, M. E. (2022, October 31). *Monitoring the future: A Continuing Study of American Youth (12th-Grade Survey), 2021.* Monitoring the Future: A Continuing Study of American Youth (12th-Grade Survey), 2021. https://www.icpsr.umich.edu/web/NAHDAP/studies/38503
8. Nagelkerke, N. J. D. 1991. A note on the General Definition of the Coefficient of Determination. *Biometrika,* 78:3, 691-692.