fathers initial romantic turbulence association with adolescent diagnosed depression

output: .pdf

NAME: Emma Sun

```
# Load Package
library(psych)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

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

```
library(gridExtra)
```

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

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

```
library(haven)
library(caret)
```

```
## Loading required package: lattice
```

```
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following objects are masked from 'package:caret':
##
```

```
##     MAE, RMSE
```

```
## The following object is masked from 'package:psych':
##
##     AUC
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
library(ggcorrplot)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

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

```
## The following object is masked from 'package:psych':
##
##     outlier
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

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

```
# Load dataset
filepath_2 <- "/Users/emmasun/Desktop/data/wave2/FF_wave2_2020v2.dta"
wave_2 <- read_dta(filepath_2)
# Check the shape
dim(wave_2)
```

```
## [1] 4898 1887
```

```
#str(wave_2)
filepath_6 <- "/Users/emmasun/Desktop/data/wave6/FF_wave6_2020v2.dta"
wave_6 <- read_dta(filepath_6)
dim(wave_6)
```

```
## [1] 4898 1617
```

```
#str(wave_6)
```

```
# Combine the datasets based on idnum
dataset <- full_join(wave_6, wave_2, by = "idnum")
# Select the desired variables
selected_vars <- c('idnum', 'cf2age', 'f2c20b1', 'f2c20b2', 'f2c20b3',
                   'f2c20b4', 'f2c20b5', 'f2f2b1','f2f2b2', 'f2f2b3',
                   'f2f2b4', 'f2f2b5', 'f2f2b6', 'f2f2b7', 'f2f2b8',
                   'f2f2b9', 'f2f2b10', 'f2f2c1', 'f2f2c2', 'f2f2c3',
                   'f2f2c4', 'f2f2c5', 'f2f2c6', 'f2f2c7', 'f2f2c8',
                   'f2f2c10', 'f2g1d', 'f2g5', 'f2g5a', 'f2g5b',
                   'f2k7', 'cf2span','f2a5', 'f2a5a', 'f2a6', 'f2a6a',
                   'f2a6a1', 'f2a6a2', 'f2a7a', 'f2a7a1a','f2a7a1b',
                   'f2a7b', 'f2a7b1a', 'f2a7b1b', 'f2a7c', 'f2a7c1a',
                   'f2a7c1b', 'f2a7d', 'f2a7d1', 'f2a7e', 'f2a7e1a',
                   'f2a7e1b', 'f2a8a', 'f2a8b', 'f2a8e', 'f2a8f',
                   'f2a8g','f2a8h', 'f2a9', 'f2a10', 'cf2marm', 'cf2cohm',
                   'f2b1a', 'f2c5', 'f2c8', 'f2c11a','f2c13c3', 'f2c14',
                   'f2c17', 'f2d1', 'f2d1x', 'f2d2c', 'f2d2d', 'f2d2e',
                   'f2d3', 'f2d3a', 'f2d4', 'f2d5a', 'f2d5b', 'f2d5c',
                   'f2d5d', 'f2d5e', 'f2d5f', 'f2d5g','f2d5h', 'f2d5i',
                   'f2d6', 'f2d7a', 'f2d7b', 'f2d7c', 'f2d7d', 'f2d7e',
                   'f2d7f','f2d7g', 'f2d7h', 'f2d7i', 'f2e1', 'f2e2a2',
                   'f2f0', 'f2h16d', 'f2l3', 'f2l6d','f2l7', 'f2l8',
                   'f2l8b', 'p6b5')
dataset <- dataset[, selected_vars]
# Check the shape
dim(dataset)
```

```
## [1] 4898  106
```

```
# Exclude "idnum" from the selected variables
selected_vars <- selected_vars[!selected_vars %in% "idnum"]
# Subset the dataset
dataset <- dataset[, selected_vars]
# Check the shape of the dataset
dim(dataset)
```

```
## [1] 4898  105
```

```
#str(dataset)
describe(dataset)
```

```
##         vars    n    mean     sd median trimmed   mad min  max range  skew
## cf2age     1 4898   17.30  18.64     24   17.50 13.34  -9   60    69 -0.46
## f2c20b1    2 4898   -3.80   7.41     -6   -5.42  4.45  -9   40    49  2.04
## f2c20b2    3 4898   -5.44   5.48     -6   -6.79  0.00  -9   40    49  3.36
## f2c20b3    4 4898   -6.33   3.76     -6   -6.79  0.00  -9   40    49  5.17
## f2c20b4    5 4898   -6.70   2.64     -6   -6.79  0.00  -9   40    49  7.02
## f2c20b5    6 4898   -6.81   2.17     -6   -6.79  0.00  -9   40    49  7.19
## f2f2b1     7 4898   -5.75   3.43     -6   -6.31  0.00  -9    2    11  1.29
## f2f2b2     8 4898   -6.04   3.08     -6   -6.62  0.00  -9    2    11  1.48
## f2f2b3     9 4898   -6.36   2.62     -6   -6.79  0.00  -9    2    11  1.70
## f2f2b4    10 4898   -6.65   2.11     -6   -6.79  0.00  -9    2    11  1.70
## f2f2b5    11 4898   -6.80   1.76     -6   -6.79  0.00  -9    2    11  1.22
```

```
## f2f2b6    12 4898   -6.87   1.57     -6   -6.79  0.00  -9    2    11  0.45
## f2f2b7    13 4898   -6.90   1.49     -6   -6.79  0.00  -9    2    11  0.03
## f2f2b8    14 4898   -6.92   1.41     -6   -6.79  0.00  -9    2    11 -0.62
## f2f2b9    15 4898   -6.93   1.39     -6   -6.79  0.00  -9   -6     3 -0.82
## f2f2b10   16 4898   -6.92   1.40     -6   -6.79  0.00  -9    1    10 -0.71
## f2f2c1    17 4898    0.24  18.61     -6   -4.93  0.00  -9   95   104  2.44
## f2f2c2    18 4898   -3.07  12.68     -6   -6.59  0.00  -9   91   100  3.45
## f2f2c3    19 4898   -5.30   6.90     -6   -6.79  0.00  -9   70    79  4.37
## f2f2c4    20 4898   -6.28   4.19     -6   -6.79  0.00  -9   59    68  5.89
## f2f2c5    21 4898   -6.67   2.71     -6   -6.79  0.00  -9   40    49  6.54
## f2f2c6    22 4898   -6.84   1.88     -6   -6.79  0.00  -9   26    35  4.50
## f2f2c7    23 4898   -6.88   1.63     -6   -6.79  0.00  -9   26    35  2.65
## f2f2c8    24 4898   -6.92   1.41     -6   -6.79  0.00  -9    6    15 -0.53
## f2f2c10   25 4898   -6.93   1.40     -6   -6.79  0.00 -11   -3     8 -0.79
## f2g1d     26 4898   -5.85   3.26     -6   -6.39  0.00  -9    2    11  1.31
## f2g5      27 4898   -6.17   7.49     -6   -6.79  0.00  -9  101   110 13.19
## f2g5a     28 4898   -6.62   2.11     -6   -6.79  0.00  -9    2    11  1.61
## f2g5b     29 4898   -6.45   6.57     -6   -6.79  0.00  -9  103   112 15.39
## f2k7      30 4898   -2.02   4.69      1   -1.63  0.00  -9    2    11 -0.81
## cf2span   31 4898   -2.72   4.22      0   -2.36  0.00  -9    1    10 -0.81
## f2a5      32 4898   -1.50   5.08      1   -1.16  1.48  -9    5    14 -0.75
## f2a5a     33 4898   -0.61  19.54      1   -2.15  0.00  -9  203   212  9.68
## f2a6      34 4898   -1.36   5.23      1   -1.12  1.48 -14    5    19 -0.69
## f2a6a     35 4898   -2.29  13.01      1   -2.90  1.48  -9  203   212 13.60
## f2a6a1    36 4898   -2.80   4.89      1   -2.63  1.48  -9    2    11 -0.27
## f2a6a2    37 4898   -3.93   5.81     -6   -4.67  4.45  -9    7    16  1.05
## f2a7a     38 4898   -1.67   4.93      1   -1.22  1.48  -9    2    11 -0.80
## f2a7a1a   39 4898   -3.73   6.35     -6   -4.78  4.45  -9   12    21  1.21
## f2a7a1b   40 4898  498.67 870.38     -6  374.80  4.45  -9 2001  2010  1.14
## f2a7b     41 4898   -2.01   4.70      1   -1.63  0.00  -9    2    11 -0.81
## f2a7b1a   42 4898   -0.38   7.39     -2   -0.73 10.38  -9   12    21  0.13
## f2a7b1b   43 4898 1140.62 991.41   1992 1176.87 10.38  -9 2001  2010 -0.30
## f2a7c     44 4898   -1.42   5.08      2   -0.90  0.00  -9    2    11 -0.82
## f2a7c1a   45 4898   -6.80   1.96     -6   -6.79  0.00  -9   12    21  3.39
## f2a7c1b   46 4898   12.73 197.68     -6   -6.79  0.00  -9 2001  2010  9.95
## f2a7d     47 4898   -1.78   4.91      1   -1.35  1.48  -9    2    11 -0.76
## f2a7d1    48 4898   -3.94   5.41     -6   -4.42  4.45  -9    5    14  0.76
## f2a7e     49 4898   -3.64   4.80     -6   -3.67  4.45 -14    2    16  0.16
## f2a7e1a   50 4898   -5.99   3.87     -6   -6.79  0.00  -9   12    21  2.81
## f2a7e1b   51 4898  151.98 542.05     -6   -6.79  0.00  -9 2001  2010  3.12
## f2a8a     52 4898   -6.25   2.85     -6   -6.79  0.00  -9    2    11  1.75
## f2a8b     53 4898   -6.25   2.86     -6   -6.79  0.00  -9    2    11  1.76
## f2a8e     54 4898   -6.31   2.66     -6   -6.79  0.00  -9    2    11  1.58
## f2a8f     55 4898   -6.25   2.86     -6   -6.79  0.00  -9    2    11  1.76
## f2a8g     56 4898   -6.24   2.86     -6   -6.79  0.00  -9    2    11  1.76
## f2a8h     57 4898   -5.46  10.18     -6   -6.79  0.00  -9  107   116 10.06
## f2a9      58 4898   -1.89   4.78      1   -1.49  1.48  -9    2    11 -0.80
## f2a10     59 4898   -5.26   3.97     -6   -5.83  4.45  -9    5    14  1.11
## cf2marm   60 4898   -2.54   4.35      0   -2.18  1.48  -9    1    10 -0.79
## cf2cohm   61 4898   -2.53   4.36      0   -2.17  1.48  -9    1    10 -0.79
## f2b1a     62 4898   -3.36   4.59      1   -3.24  1.48  -9    2    11 -0.15
## f2c5      63 4898   -1.62   4.96      1   -1.15  1.48  -9    2    11 -0.80
## f2c8      64 4898   -1.89   4.95      1   -1.49  1.48  -9    2    11 -0.68
## f2c11a    65 4898   -6.19   2.89     -6   -6.79  0.00  -9    2    11  1.59
```

```
## f2c13c3   66 4898   -6.19   2.88     -6   -6.79  0.00  -9    2    11  1.59
## f2c14     67 4898   -4.61   4.26     -6   -4.89  4.45  -9    2    11  0.55
## f2c17     68 4898   -4.25   4.46     -6   -4.44  4.45  -9    2    11  0.39
## f2d1      69 4898   -2.11   4.63      1   -1.65  0.00  -9    2    11 -0.81
## f2d1x     70 4898   -5.65   2.96     -5   -6.06  0.00 -10    3    13  1.04
## f2d2c     71 4898   -3.04   4.63     -2   -2.96  4.45  -9    4    13 -0.24
## f2d2d     72 4898   -3.05   4.61     -2   -2.97  4.45  -9    4    13 -0.25
## f2d2e     73 4898   -3.06   4.60     -2   -2.99  4.45  -9    4    13 -0.25
## f2d3      74 4898   -2.01   5.25      1   -1.91  4.45 -10    5    15 -0.37
## f2d3a     75 4898   -1.79   5.56      2   -1.66  4.45  -9    5    14 -0.28
## f2d4      76 4898   -2.05   4.74      1   -1.69  0.00  -9    2    11 -0.76
## f2d5a     77 4898   -2.80   4.91      1   -2.68  2.97  -9    3    12 -0.25
## f2d5b     78 4898   -3.01   4.71      1   -2.89  1.48  -9    3    12 -0.27
## f2d5c     79 4898   -2.33   5.33      2   -2.16  1.48  -9    3    12 -0.24
## f2d5d     80 4898   -2.98   4.74      1   -2.88  1.48  -9    3    12 -0.27
## f2d5e     81 4898   -2.20   5.44      2   -2.00  1.48  -9    3    12 -0.24
## f2d5f     82 4898   -2.08   5.54      3   -1.86  0.00  -9    3    12 -0.25
## f2d5g     83 4898   -2.14   5.49      2   -1.93  1.48  -9    3    12 -0.24
## f2d5h     84 4898   -3.46   4.58     -5   -3.45  5.93  -9    3    12 -0.06
## f2d5i     85 4898   -3.41   4.64     -5   -3.41  5.93  -9    3    12 -0.05
## f2d6      86 4898   -5.45   3.63     -6   -5.86  4.45  -9    2    11  0.98
## f2d7a     87 4898   -6.21   2.91     -6   -6.79  0.00  -9    3    12  1.74
## f2d7b     88 4898   -6.23   2.84     -6   -6.79  0.00  -9    3    12  1.69
## f2d7c     89 4898   -6.20   2.92     -6   -6.79  0.00  -9    3    12  1.75
## f2d7d     90 4898   -6.21   2.89     -6   -6.79  0.00  -9    3    12  1.73
## f2d7e     91 4898   -6.17   3.01     -6   -6.79  0.00  -9    3    12  1.81
## f2d7f     92 4898   -6.14   3.10     -6   -6.79  0.00  -9    3    12  1.86
## f2d7g     93 4898   -6.16   3.05     -6   -6.79  0.00  -9    3    12  1.84
## f2d7h     94 4898   -6.27   2.78     -6   -6.79  0.00  -9    3    12  1.73
## f2d7i     95 4898   -6.26   2.82     -6   -6.79  0.00  -9    3    12  1.77
## f2e1      96 4898   -2.26   4.54      1   -1.86  1.48  -9    2    11 -0.79
## f2e2a2    97 4898   -4.96   4.00     -6   -5.27  4.45  -9    2    11  0.69
## f2f0      98 4898   -1.89   4.78      1   -1.48  1.48  -9    2    11 -0.80
## f2h16d    99 4898   -1.89   4.78      1   -1.49  1.48  -9    2    11 -0.80
## f2l3     100 4898   -1.95   4.74      1   -1.56  0.00  -9    2    11 -0.80
## f2l6d    101 4898   -2.07   4.69      1   -1.69  0.00  -9    2    11 -0.78
## f2l7     102 4898   -1.97   5.32      1   -1.71  2.97  -9    3    12 -0.40
## f2l8     103 4898   -4.20   8.36     -6   -5.09  4.45  -9  203   212 14.21
## f2l8b    104 4898   -2.69   4.68      1   -2.43  0.00  -9    2    11 -0.47
## p6b5     105 4898   -1.04   4.84      2   -0.43  0.00  -9    2    11 -1.03
##          kurtosis    se
## cf2age      -1.27  0.27
## f2c20b1      3.96  0.11
## f2c20b2     13.37  0.08
## f2c20b3     37.30  0.05
## f2c20b4     84.65  0.04
## f2c20b5    107.35  0.03
## f2f2b1       0.65  0.05
## f2f2b2       1.61  0.04
## f2f2b3       3.40  0.04
## f2f2b4       5.64  0.03
## f2f2b5       6.16  0.03
## f2f2b6       3.99  0.02
## f2f2b7       2.49  0.02
```

```
## f2f2b8     -0.34  0.02
## f2f2b9     -1.33  0.02
## f2f2b10    -0.89  0.02
## f2f2c1      4.93  0.27
## f2f2c2     12.10  0.18
## f2f2c3     22.42  0.10
## f2f2c4     47.38  0.06
## f2f2c5     69.39  0.04
## f2f2c6     60.41  0.03
## f2f2c7     45.67  0.02
## f2f2c8      0.55  0.02
## f2f2c10    -1.26  0.02
## f2g1d       0.88  0.05
## f2g5      185.65  0.11
## f2g5a       5.33  0.03
## f2g5b     249.95  0.09
## f2k7       -1.33  0.07
## cf2span    -1.33  0.06
## f2a5       -1.32  0.07
## f2a5a      98.12  0.28
## f2a6       -1.32  0.07
## f2a6a     212.14  0.19
## f2a6a1     -1.80  0.07
## f2a6a2     -0.51  0.08
## f2a7a      -1.33  0.07
## f2a7a1a     0.01  0.09
## f2a7a1b    -0.70 12.44
## f2a7b      -1.33  0.07
## f2a7b1a    -1.49  0.11
## f2a7b1b    -1.91 14.17
## f2a7c      -1.33  0.07
## f2a7c1a    27.96  0.03
## f2a7c1b    97.00  2.82
## f2a7d      -1.38  0.07
## f2a7d1     -1.14  0.08
## f2a7e      -1.77  0.07
## f2a7e1a     8.51  0.06
## f2a7e1b     7.71  7.75
## f2a8a       3.11  0.04
## f2a8b       3.12  0.04
## f2a8e       2.67  0.04
## f2a8f       3.12  0.04
## f2a8g       3.12  0.04
## f2a8h     106.64  0.15
## f2a9       -1.33  0.07
## f2a10      -0.01  0.06
## cf2marm    -1.33  0.06
## cf2cohm    -1.33  0.06
## f2b1a      -1.83  0.07
## f2c5       -1.33  0.07
## f2c8       -1.48  0.07
## f2c11a      2.31  0.04
## f2c13c3     2.31  0.04
## f2c14      -1.34  0.06
```

```
## f2c17      -1.57  0.06
## f2d1       -1.34  0.07
## f2d1x       1.47  0.04
## f2d2c      -1.68  0.07
## f2d2d      -1.69  0.07
## f2d2e      -1.68  0.07
## f2d3       -1.62  0.08
## f2d3a      -1.70  0.08
## f2d4       -1.39  0.07
## f2d5a      -1.79  0.07
## f2d5b      -1.79  0.07
## f2d5c      -1.82  0.08
## f2d5d      -1.79  0.07
## f2d5e      -1.83  0.08
## f2d5f      -1.84  0.08
## f2d5g      -1.84  0.08
## f2d5h      -1.78  0.07
## f2d5i      -1.77  0.07
## f2d6       -0.36  0.05
## f2d7a       3.02  0.04
## f2d7b       2.89  0.04
## f2d7c       3.06  0.04
## f2d7d       3.02  0.04
## f2d7e       3.23  0.04
## f2d7f       3.29  0.04
## f2d7g       3.27  0.04
## f2d7h       3.24  0.04
## f2d7i       3.37  0.04
## f2e1       -1.33  0.06
## f2e2a2     -1.08  0.06
## f2f0       -1.33  0.07
## f2h16d     -1.33  0.07
## f2l3       -1.33  0.07
## f2l6d      -1.36  0.07
## f2l7       -1.72  0.08
## f2l8      322.39  0.12
## f2l8b      -1.69  0.07
## p6b5       -0.92  0.07
```

```
#summary(dataset)
# Check missing value
colSums(is.na(dataset))
```

```
##  cf2age f2c20b1 f2c20b2 f2c20b3 f2c20b4 f2c20b5  f2f2b1  f2f2b2  f2f2b3  f2f2b4
##       0       0       0       0       0       0       0       0       0       0
##  f2f2b5  f2f2b6  f2f2b7  f2f2b8  f2f2b9 f2f2b10  f2f2c1  f2f2c2  f2f2c3  f2f2c4
##       0       0       0       0       0       0       0       0       0       0
##  f2f2c5  f2f2c6  f2f2c7  f2f2c8 f2f2c10   f2g1d    f2g5   f2g5a   f2g5b    f2k7
##       0       0       0       0       0       0       0       0       0       0
## cf2span    f2a5   f2a5a    f2a6   f2a6a  f2a6a1  f2a6a2   f2a7a f2a7a1a f2a7a1b
##       0       0       0       0       0       0       0       0       0       0
##   f2a7b f2a7b1a f2a7b1b   f2a7c f2a7c1a f2a7c1b   f2a7d  f2a7d1   f2a7e f2a7e1a
##       0       0       0       0       0       0       0       0       0       0
## f2a7e1b   f2a8a   f2a8b   f2a8e   f2a8f   f2a8g   f2a8h    f2a9   f2a10 cf2marm
##       0       0       0       0       0       0       0       0       0       0
```

```
## cf2cohm   f2b1a    f2c5    f2c8  f2c11a f2c13c3   f2c14   f2c17    f2d1   f2d1x
##       0       0       0       0       0       0       0       0       0       0
##   f2d2c   f2d2d   f2d2e    f2d3   f2d3a    f2d4   f2d5a   f2d5b   f2d5c   f2d5d
##       0       0       0       0       0       0       0       0       0       0
##   f2d5e   f2d5f   f2d5g   f2d5h   f2d5i    f2d6   f2d7a   f2d7b   f2d7c   f2d7d
##       0       0       0       0       0       0       0       0       0       0
##   f2d7e   f2d7f   f2d7g   f2d7h   f2d7i    f2e1  f2e2a2    f2f0  f2h16d    f213
##       0       0       0       0       0       0       0       0       0       0
##   f216d    f217    f218   f218b    p6b5
##       0       0       0       0       0
```

```
# Check duplicate
table(duplicated(dataset))
```

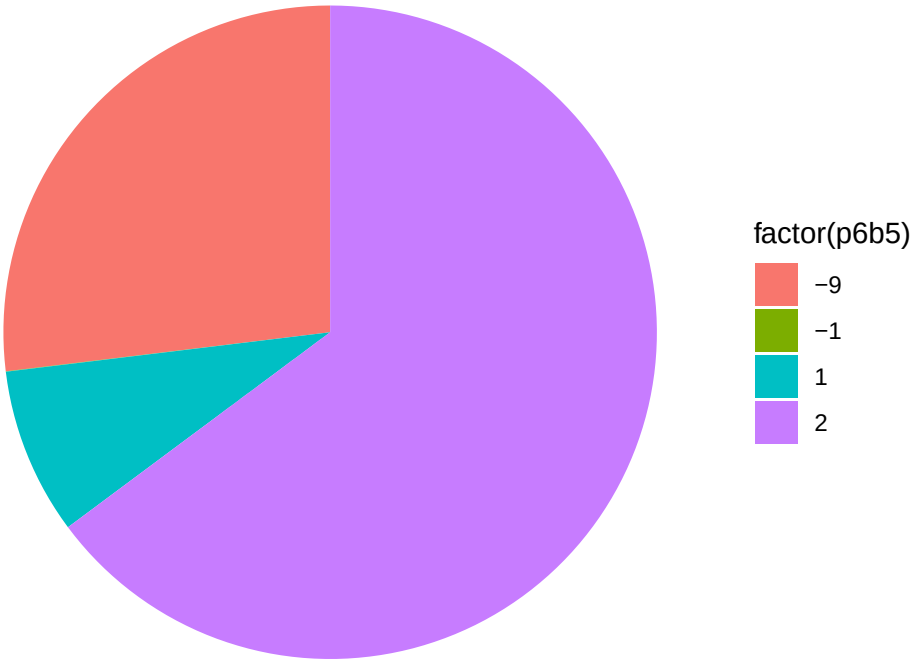
```
##
## FALSE  TRUE
##  3383  1515
```

```
#Check target variable
table(dataset$p6b5)
```

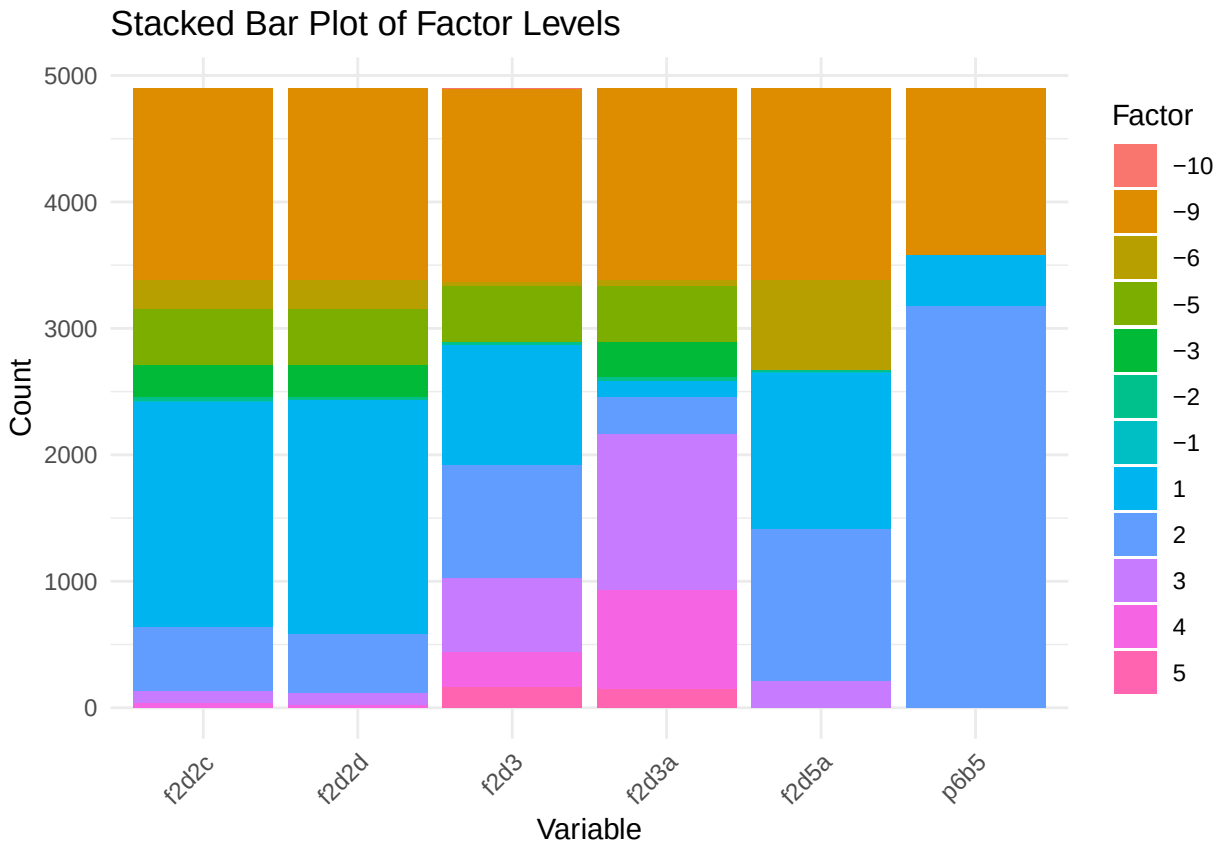
```
##
##   -9   -1    1    2
## 1318    1  404 3175
```

```
# Extract the variable
p6b5<- dataset$p6b5
# Compute frequencies of unique values
value_counts <- table(p6b5)
# Convert to data frame
value_counts_df <- as.data.frame(value_counts)
# Plot pie chart
ggplot(value_counts_df, aes(x = "", y = Freq, fill = factor(p6b5))) +
  geom_bar(stat = "identity") +
  coord_polar("y", start=0) +
  theme_void() +
  labs(title = "Distribution of Target Variable")
```

Distribution of Target Variable




```
# Make a copy of the dataset and rename it to "stake_plot_data"
stake_plot_data <- dataset
# Specify the column names to reorder
columns_to_reorder <- c("p6b5","f2d3a","f2d3","f2d2d","f2d2c","f2d5a")
# Loop through each column and reorder factor levels
for (col in columns_to_reorder) {
  stake_plot_data[[col]] <- factor(stake_plot_data[[col]],
                                   levels = c("-10","-9", "-8", "-7", "-6",
                                              "-5", "-4", "-3", "-2", "-1",
                                              "0", "1", "2", "3", "4", "5"))
}
# Reshape the data to long format for plotting
data_long <- stake_plot_data %>%
  select(all_of(columns_to_reorder)) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")
# Plot the stacked bar plot with title
ggplot(data_long, aes(x = Variable, fill = Value)) +
  geom_bar() +
  labs(x = "Variable", y = "Count", fill = "Factor") +
  ggtitle("Stacked Bar Plot of Factor Levels") +  # Add title
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Remove rows with negative values in 'p6b5'
dataset <- subset(dataset, p6b5 >= 0)
# Check the shape of the dataset after removal
dim(dataset)
```

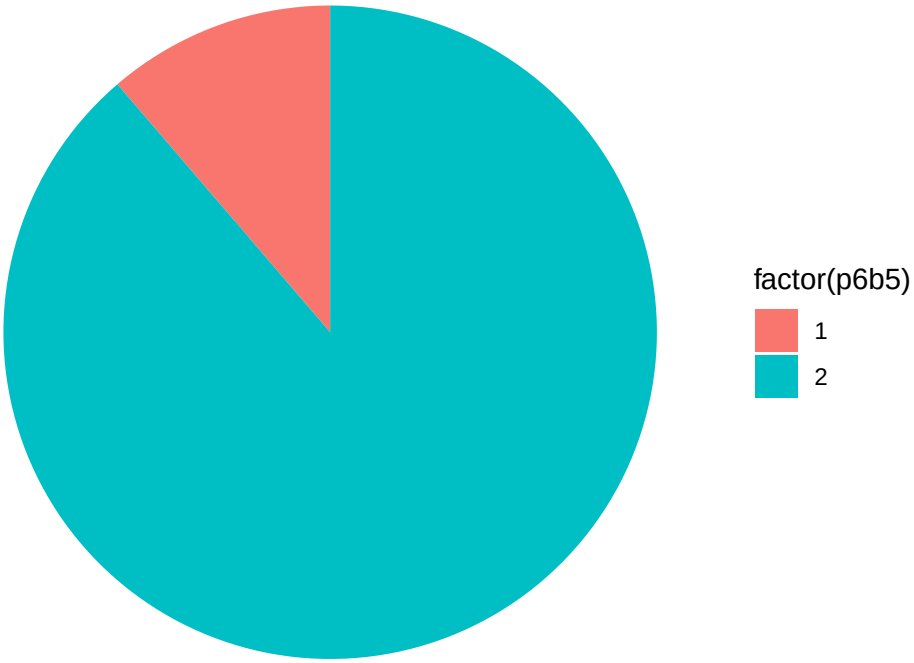
```
## [1] 3579  105
```

```
#Check target variable
table(dataset$p6b5)
```

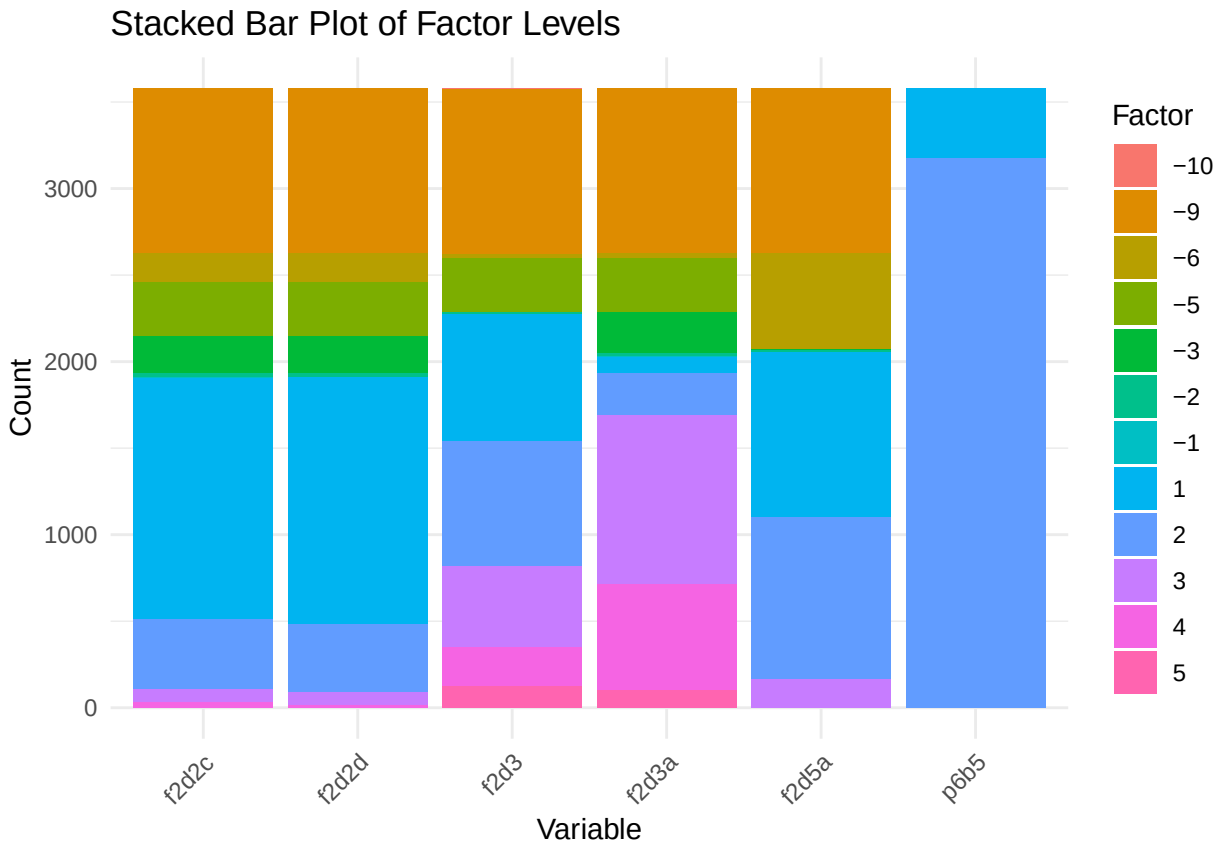
```
##
##    1    2
##  404 3175
```

```
# Extract the variable
p6b5<- dataset$p6b5
# Compute frequencies of unique values
value_counts <- table(p6b5)
# Convert to data frame
value_counts_df <- as.data.frame(value_counts)
# Plot pie chart
ggplot(value_counts_df, aes(x = "", y = Freq, fill = factor(p6b5))) +
  geom_bar(stat = "identity") +
  coord_polar("y", start=0) +
  theme_void() +
  labs(title = "Distribution of Target Variable")
```

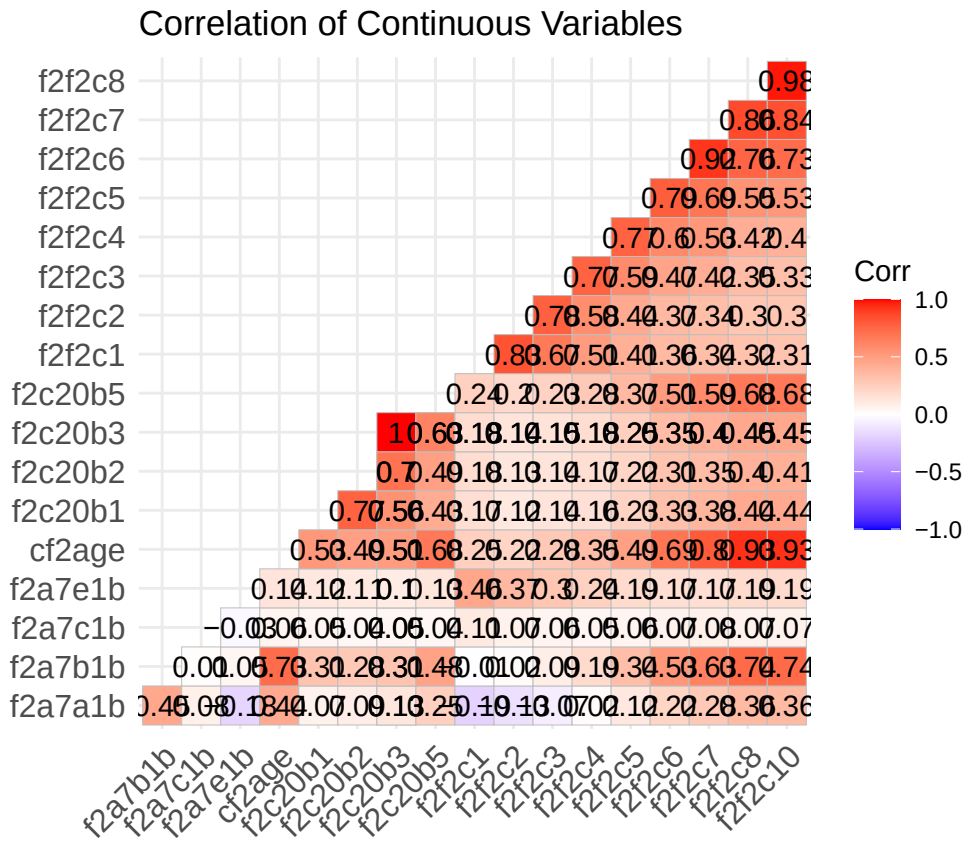
Distribution of Target Variable



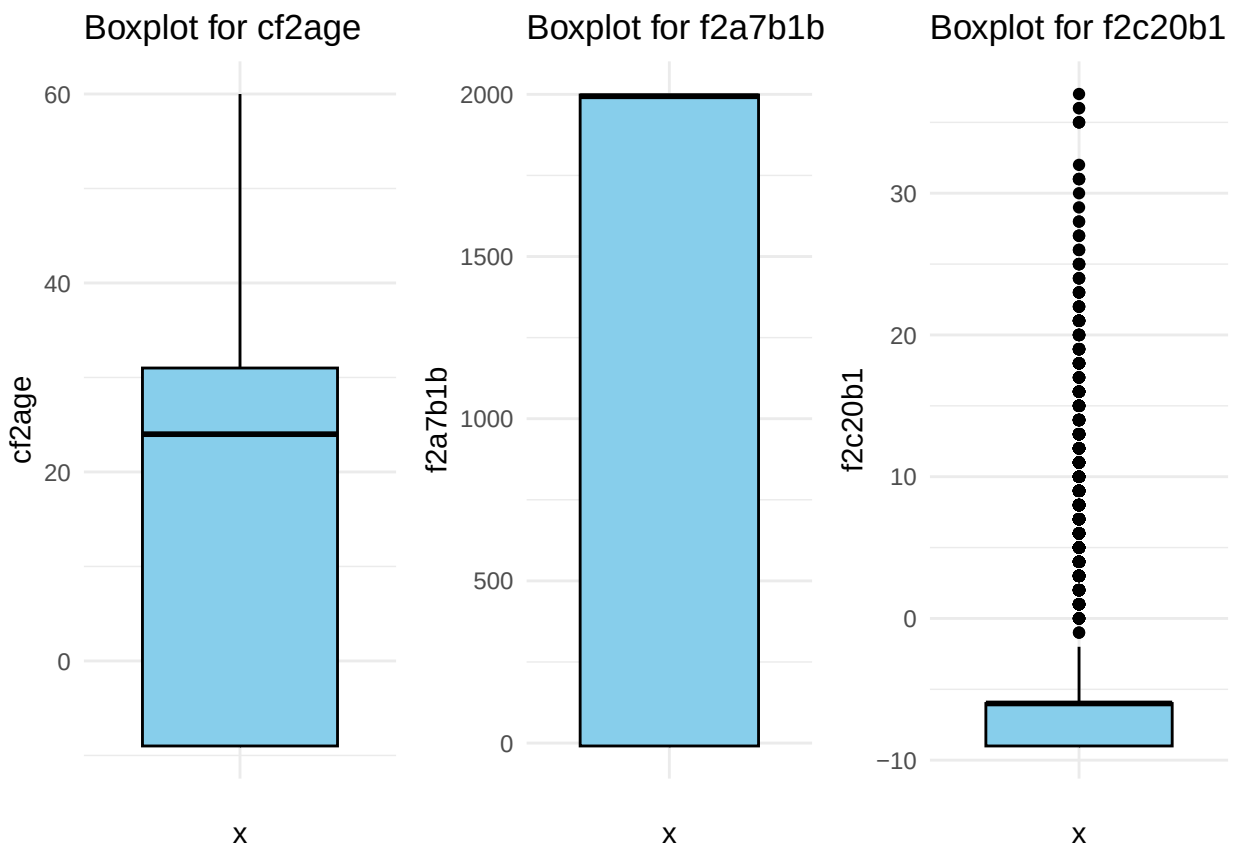

```
# Make a copy of the dataset and rename it to "stake_plot_data"
stake_plot_data <- dataset
# Specify the column names to reorder
columns_to_reorder <- c("p6b5","f2d3a","f2d3","f2d2d","f2d2c","f2d5a")
# Loop through each column and reorder factor levels
for (col in columns_to_reorder) {
  stake_plot_data[[col]] <- factor(stake_plot_data[[col]],
                                   levels = c("-10","-9", "-8", "-7", "-6",
                                              "-5", "-4", "-3", "-2", "-1", "0",
                                              "1", "2", "3", "4", "5"))
}
# Reshape the data to long format for plotting
data_long <- stake_plot_data %>%
  select(all_of(columns_to_reorder)) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")
# Plot the stacked bar plot with title
ggplot(data_long, aes(x = Variable, fill = Value)) +
  geom_bar() +
  labs(x = "Variable", y = "Count", fill = "Factor") +
  ggtitle("Stacked Bar Plot of Factor Levels") +  # Add title
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
num_cols <- c('f2a7a1b','f2a7b1b','f2a7c1b','f2a7e1b','cf2age','f2c20b1',
              'f2c20b2','f2c20b3','f2c20b3','f2c20b5','f2f2c1','f2f2c2',
              'f2f2c3','f2f2c4','f2f2c5','f2f2c6','f2f2c7','f2f2c8',
              'f2f2c10')
ggcorrplot(cor(dataset[,num_cols]) , type = "lower",
           lab=T, title = 'Correlation of Continuous Variables')
```




```
# Create boxplots for each variable
boxplot_cf2age <- ggplot(dataset, aes(x = "", y = cf2age)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot for cf2age", y = "cf2age") +
  theme_minimal()
boxplot_f2a7b1b <- ggplot(dataset, aes(x = "", y = f2a7b1b)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot for f2a7b1b", y = "f2a7b1b") +
  theme_minimal()
boxplot_f2c20b1 <- ggplot(dataset, aes(x = "", y = f2c20b1)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot for f2c20b1", y = "f2c20b1") +
  theme_minimal()
# Combine boxplots in a single plot
combined_boxplot <- grid.arrange(boxplot_cf2age, boxplot_f2a7b1b,
                                 boxplot_f2c20b1, ncol = 3)
```




```
# Display the combined boxplots
print(combined_boxplot)
```

```
## TableGrob (1 x 3) "arrange": 3 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (1-1,3-3) arrange gtable[layout]
```

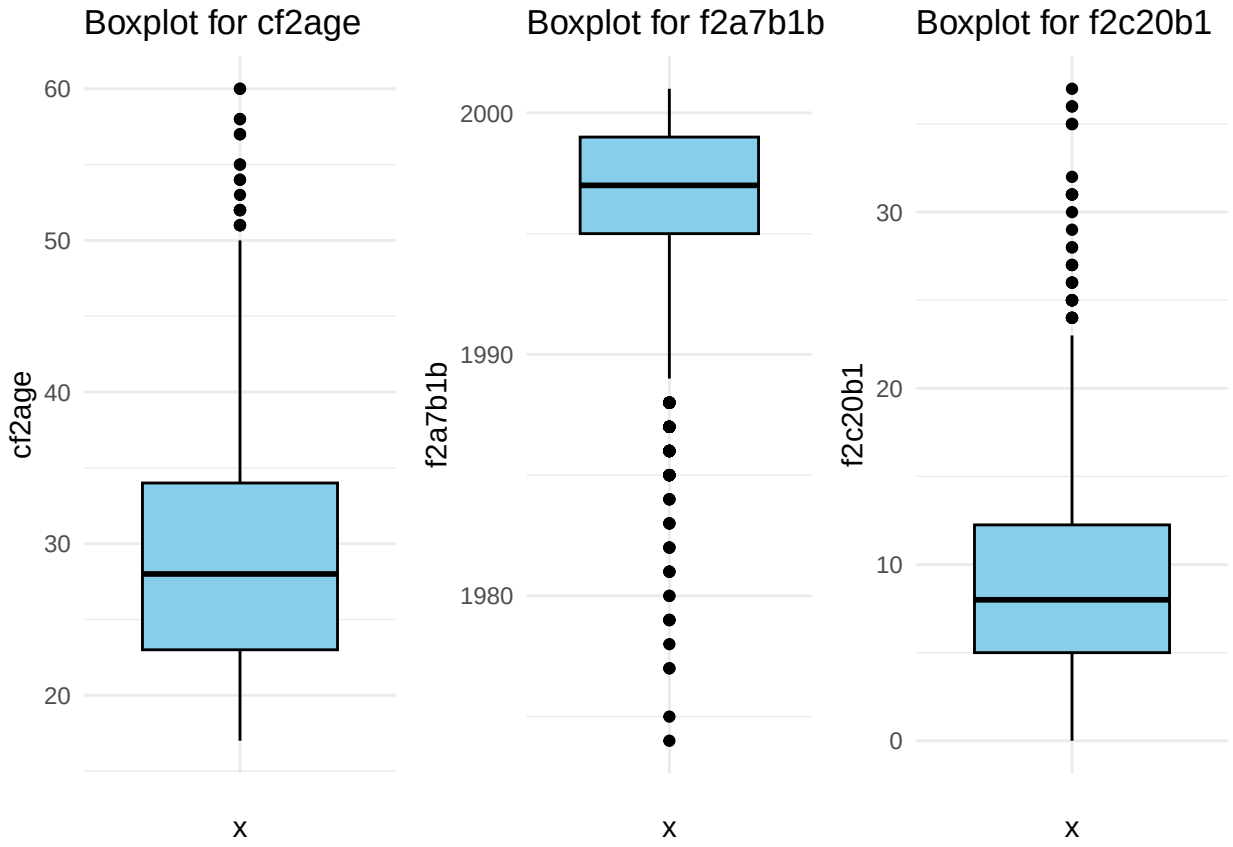
```
dataset3 <- dataset
# Delete negative values for each variable
dataset3$cf2age[dataset3$cf2age < 0] <- NA
dataset3$f2a7b1b[dataset3$f2a7b1b < 0] <- NA
dataset3$f2c20b1[dataset3$f2c20b1 < 0] <- NA
# Create boxplots for each variable
boxplot_cf2age <- ggplot(dataset3, aes(x = "", y = cf2age)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot for cf2age", y = "cf2age") +
  theme_minimal()
boxplot_f2a7b1b <- ggplot(dataset3, aes(x = "", y = f2a7b1b)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot for f2a7b1b", y = "f2a7b1b") +
  theme_minimal()
boxplot_f2c20b1 <- ggplot(dataset3, aes(x = "", y = f2c20b1)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(title = "Boxplot for f2c20b1", y = "f2c20b1") +
  theme_minimal()
# Combine boxplots in a single plot
```

```
combined_boxplot <- grid.arrange(boxplot_cf2age, boxplot_f2a7b1b,
                                 boxplot_f2c20b1, ncol = 3)
```

```
## Warning: Removed 952 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 1392 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 2799 rows containing non-finite values (`stat_boxplot()`).
```




```
# Display the combined boxplots
print(combined_boxplot)
```

```
## TableGrob (1 x 3) "arrange": 3 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (1-1,3-3) arrange gtable[layout]
```

```
# Factorizing each variable
dataset1 <- dataset
dataset1$p6b5 <- factor(dataset1$p6b5)
# Set the number of folds for stratified K-fold cross-validation
num_folds <- 5
# Create stratified folds
folds <- createFolds(dataset1$p6b5, k = num_folds,
                     list = TRUE, returnTrain = FALSE)
# Initialize lists to store evaluation metrics
accuracy1 <- vector("numeric", num_folds)
precision1 <- vector("numeric", num_folds)
```

```r
recall1 <- vector("numeric", num_folds)
f1_score1 <- vector("numeric", num_folds)
# Loop through each fold
for (i in 1:num_folds) {
  # Extract the indices for the current fold
  test_index <- unlist(folds[i])
  # Create training indices by excluding the test indices
  train_index <- setdiff(1:nrow(dataset1), test_index)
  # Extract train and test data using the indices
  train_data <- dataset1[train_index, ]
  test_data <- dataset1[test_index, ]
  # Fit GLM model
  glm_model1 <- glm(p6b5 ~ ., data = train_data, family = binomial)
  glm_model2 <- glm(p6b5 ~ f2f2b5 + f2f2c5 + f2d5f + f2d7c + f2d7i,
                    data = train_data, family = binomial)
  # Make predictions on the test data
  predictions <- predict(glm_model1, newdata = test_data, type = "response")
  # Convert predictions to binary outcomes (0 or 1) based on a threshold
  predicted_classes <- ifelse(predictions >= 0.5, 1, 0)
  # Evaluate the model performance
  confusion_matrix1 <- table(predicted_classes, test_data$p6b5)
  accuracy1[i] <- sum(diag(confusion_matrix1)) / sum(confusion_matrix1)
  precision1[i] <- confusion_matrix1[2, 2] / sum(confusion_matrix1[, 2])
  recall1[i] <- confusion_matrix1[2, 2] / sum(confusion_matrix1[2, ])
  f1_score1[i] <- 2 * precision1[i] * recall1[i] / (precision1[i] + recall1[i])
}
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# Print evaluation metrics
cat("Accuracy:", mean(accuracy1), "\n")
```

```
## Accuracy: 0.878737
```

```r
cat("Precision:", mean(precision1), "\n")
```

```
## Precision: 0.9892913
```

```r
cat("Recall:", mean(recall1), "\n")
```

```
## Recall: 0.8870317
```

```r
cat("F1 Score:", mean(f1_score1), "\n")
```

```
## F1 Score: 0.9353571
```

```
confusion_matrix1
```

```
##
## predicted_classes   1   2
##                 0   1  18
##                 1  80 617
```

```
# Print model summary
summary(glm_model1)
```

```
##
## Call:
## glm(formula = p6b5 ~ ., family = binomial, data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.047e+03  1.867e+05   0.027  0.97843
## cf2age      -8.640e-03  1.367e-02  -0.632  0.52745
## f2c20b1     -5.827e-03  1.329e-02  -0.438  0.66103
## f2c20b2      2.952e-02  2.252e-02   1.311  0.18992
## f2c20b3     -4.605e-02  2.834e-02  -1.625  0.10414
## f2c20b4      1.546e-01  1.145e-01   1.350  0.17706
## f2c20b5     -1.143e-01  1.287e-01  -0.888  0.37463
## f2f2b1       6.669e-02  8.863e-02   0.752  0.45177
## f2f2b2      -2.258e-02  8.962e-02  -0.252  0.80112
## f2f2b3      -1.760e-02  9.132e-02  -0.193  0.84719
## f2f2b4      -1.328e-01  1.198e-01  -1.108  0.26788
## f2f2b5       7.601e-01  3.051e-01   2.491  0.01273 *
## f2f2b6      -9.199e-01  6.920e-01  -1.329  0.18371
## f2f2b7       1.776e+01  1.748e+03   0.010  0.99189
## f2f2b8       3.655e+00  1.962e+04   0.000  0.99985
## f2f2b9       7.790e+02  2.962e+04   0.026  0.97902
## f2f2b10      1.218e+00  8.581e+02   0.001  0.99887
## f2f2c1      -5.040e-03  1.219e-02  -0.414  0.67919
## f2f2c2      -6.772e-03  1.514e-02  -0.447  0.65465
## f2f2c3       6.297e-03  2.553e-02   0.247  0.80520
## f2f2c4       5.560e-02  4.978e-02   1.117  0.26405
## f2f2c5      -3.006e-01  1.047e-01  -2.872  0.00408 **
## f2f2c6       3.695e-01  3.724e-01   0.992  0.32112
## f2f2c7      -1.022e+01  1.450e+03  -0.007  0.99438
## f2f2c8      -8.606e+00  2.203e+04   0.000  0.99969
## f2f2c10     -1.675e+00  1.032e+03  -0.002  0.99871
## f2g1d        4.038e-02  3.043e-02   1.327  0.18451
## f2g5         1.603e-04  1.259e-02   0.013  0.98984
## f2g5a        9.186e-02  6.070e-02   1.513  0.13021
## f2g5b       -1.714e-02  9.630e-03  -1.780  0.07506 .
## f2k7        -2.184e-01  2.476e-01  -0.882  0.37763
## cf2span      9.493e-02  3.154e-01   0.301  0.76343
## f2a5        -1.042e-01  2.337e-01  -0.446  0.65563
## f2a5a       -7.165e-03  3.406e-03  -2.103  0.03542 *
## f2a6        -3.065e-01  3.638e-01  -0.843  0.39951
## f2a6a        7.084e-03  8.620e-03   0.822  0.41121
## f2a6a1       2.403e+00  1.005e+02   0.024  0.98093
## f2a6a2      -3.321e-02  7.398e-02  -0.449  0.65351
## f2a7a       -1.719e+01  4.914e+03  -0.003  0.99721
```

```
## f2a7a1a     -4.663e-02  3.621e-02  -1.288  0.19783
## f2a7a1b      4.825e-04  4.058e-04   1.189  0.23437
## f2a7b        2.995e-01  5.338e-01   0.561  0.57476
## f2a7b1a      2.781e-02  1.951e-02   1.425  0.15413
## f2a7b1b     -4.926e-05  2.190e-04  -0.225  0.82205
## f2a7c       -1.738e+02  8.369e+03  -0.021  0.98343
## f2a7c1a      4.954e-02  2.024e-01   0.245  0.80662
## f2a7c1b     -1.644e-03  7.134e-03  -0.231  0.81769
## f2a7d        2.364e+01  8.468e+02   0.028  0.97773
## f2a7d1      -6.092e-02  1.125e-01  -0.541  0.58817
## f2a7e        3.362e-02  1.507e-01   0.223  0.82350
## f2a7e1a     -1.097e-03  4.895e-02  -0.022  0.98212
## f2a7e1b     -4.150e-04  4.743e-04  -0.875  0.38156
## f2a8a        1.732e+00  6.781e-01   2.555  0.01063 *
## f2a8b       -2.540e+00  2.538e+00  -1.001  0.31698
## f2a8e        2.003e-01  5.369e-01   0.373  0.70903
## f2a8f       -1.750e+01  3.627e+03  -0.005  0.99615
## f2a8g        1.830e+01  3.627e+03   0.005  0.99597
## f2a8h       -9.513e-03  6.489e-03  -1.466  0.14262
## f2a9        -1.767e+01  2.717e+03  -0.007  0.99481
## f2a10       -4.372e-02  1.176e-01  -0.372  0.71009
## cf2marm     -4.460e+01  4.971e+03  -0.009  0.99284
## cf2cohm      1.988e+00  3.207e+00   0.620  0.53532
## f2b1a       -6.570e-03  7.103e-02  -0.092  0.92631
## f2c5        -1.693e-01  3.840e-01  -0.441  0.65930
## f2c8         9.612e-01  7.036e-01   1.366  0.17190
## f2c11a      -1.852e+00  1.536e+03  -0.001  0.99904
## f2c13c3      8.142e-01  1.536e+03   0.001  0.99958
## f2c14       -1.027e+00  7.044e-01  -1.458  0.14484
## f2c17       -1.026e-02  7.408e-02  -0.139  0.88982
## f2d1        -1.203e+01  5.153e+02  -0.023  0.98138
## f2d1x        4.106e-02  1.359e-01   0.302  0.76260
## f2d2c       -2.746e-02  1.390e-01  -0.198  0.84334
## f2d2d        1.815e-01  1.667e-01   1.089  0.27637
## f2d2e       -1.252e-01  1.373e-01  -0.912  0.36176
## f2d3        -3.907e-02  8.074e-02  -0.484  0.62845
## f2d3a        6.129e-03  4.382e-02   0.140  0.88875
## f2d4        -1.591e+00  6.441e+01  -0.025  0.98030
## f2d5a        3.374e-02  1.322e-01   0.255  0.79862
## f2d5b       -1.030e-01  2.046e-01  -0.504  0.61455
## f2d5c       -1.695e-01  1.392e-01  -1.218  0.22316
## f2d5d       -9.111e-02  1.766e-01  -0.516  0.60587
## f2d5e       -4.503e-02  1.561e-01  -0.288  0.77304
## f2d5f        4.876e-01  2.106e-01   2.316  0.02058 *
## f2d5g        1.177e-01  1.447e-01   0.814  0.41589
## f2d5h        3.314e-01  2.238e-01   1.480  0.13879
## f2d5i       -1.926e-01  1.817e-01  -1.060  0.28920
## f2d6         6.435e-04  2.599e-01   0.002  0.99802
## f2d7a        5.774e-02  2.586e-01   0.223  0.82332
## f2d7b       -2.983e-01  2.889e-01  -1.033  0.30179
## f2d7c        7.571e-01  2.617e-01   2.892  0.00382 **
## f2d7d        3.373e-01  2.803e-01   1.204  0.22874
## f2d7e       -3.553e-02  2.584e-01  -0.137  0.89064
## f2d7f       -9.451e-01  3.640e-01  -2.597  0.00941 **
```

```
## f2d7g       -3.296e-02  2.786e-01  -0.118  0.90582
## f2d7h       -4.463e-01  3.433e-01  -1.300  0.19369
## f2d7i        6.583e-01  3.276e-01   2.010  0.04448 *
## f2e1        -8.541e-01  1.530e+00  -0.558  0.57672
## f2e2a2      -3.972e-02  9.969e-02  -0.398  0.69030
## f2f0         2.130e+01  2.718e+03   0.008  0.99375
## f2h16d      -1.207e+00  1.006e+02  -0.012  0.99043
## f2l3         6.245e-02  5.568e-01   0.112  0.91070
## f2l6d       -3.294e-02  1.632e-01  -0.202  0.84002
## f2l7         6.029e-02  1.329e-01   0.454  0.64998
## f2l8         3.114e-02  2.498e-02   1.247  0.21253
## f2l8b       -2.603e-02  1.675e-01  -0.155  0.87648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2017.7  on 2862  degrees of freedom
## Residual deviance: 1898.1  on 2758  degrees of freedom
## AIC: 2108.1
##
## Number of Fisher Scoring iterations: 18
```

```
# perform chi test
test <- anova(glm_model1, glm_model2, test = "Chisq")
print(test)
```

```
## Analysis of Deviance Table
##
## Model 1: p6b5 ~ cf2age + f2c20b1 + f2c20b2 + f2c20b3 + f2c20b4 + f2c20b5 +
##     f2f2b1 + f2f2b2 + f2f2b3 + f2f2b4 + f2f2b5 + f2f2b6 + f2f2b7 +
##     f2f2b8 + f2f2b9 + f2f2b10 + f2f2c1 + f2f2c2 + f2f2c3 + f2f2c4 +
##     f2f2c5 + f2f2c6 + f2f2c7 + f2f2c8 + f2f2c10 + f2g1d + f2g5 +
##     f2g5a + f2g5b + f2k7 + cf2span + f2a5 + f2a5a + f2a6 + f2a6a +
##     f2a6a1 + f2a6a2 + f2a7a + f2a7a1a + f2a7a1b + f2a7b + f2a7b1a +
##     f2a7b1b + f2a7c + f2a7c1a + f2a7c1b + f2a7d + f2a7d1 + f2a7e +
##     f2a7e1a + f2a7e1b + f2a8a + f2a8b + f2a8e + f2a8f + f2a8g +
##     f2a8h + f2a9 + f2a10 + cf2marm + cf2cohm + f2b1a + f2c5 +
##     f2c8 + f2c11a + f2c13c3 + f2c14 + f2c17 + f2d1 + f2d1x +
##     f2d2c + f2d2d + f2d2e + f2d3 + f2d3a + f2d4 + f2d5a + f2d5b +
##     f2d5c + f2d5d + f2d5e + f2d5f + f2d5g + f2d5h + f2d5i + f2d6 +
##     f2d7a + f2d7b + f2d7c + f2d7d + f2d7e + f2d7f + f2d7g + f2d7h +
##     f2d7i + f2e1 + f2e2a2 + f2f0 + f2h16d + f2l3 + f2l6d + f2l7 +
##     f2l8 + f2l8b
## Model 2: p6b5 ~ f2f2b5 + f2f2c5 + f2d5f + f2d7c + f2d7i
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      2758     1898.1
## 2      2857     1998.6 -99  -100.54    0.438
```

```
# Convert the target variable to a factor
dataset2 <- dataset
dataset2$p6b5 <- factor(dataset2$p6b5)
# Set the number of folds for stratified K-fold cross-validation
num_folds <- 5
# Create stratified folds
```

```r
folds <- createFolds(dataset2$p6b5, k = num_folds,
                     list = TRUE, returnTrain = FALSE)
# Initialize lists to store evaluation metrics
accuracy2 <- vector("numeric", num_folds)
precision2 <- vector("numeric", num_folds)
recall2 <- vector("numeric", num_folds)
f1_score2 <- vector("numeric", num_folds)
# Loop through each fold
for (i in 1:num_folds) {
  # Extract the indices for the current fold
  test_index <- unlist(folds[i])
  # Create training indices by excluding the test indices
  train_index <- setdiff(1:nrow(dataset2), test_index)
  # Extract train and test data using the indices
  train_data <- dataset2[train_index, ]
  test_data <- dataset2[test_index, ]
  # Fit model
  rf_model <- randomForest(p6b5 ~ ., data = train_data, ntree = 100)
  # Make predictions on the test data
  predictions <- predict(rf_model, newdata = test_data)
  # Evaluate the model performance
  confusion_matrix2 <- table(predictions, test_data$p6b5)
  # Calculate evaluation metrics
  accuracy2[i] <- sum(diag(confusion_matrix2)) / sum(confusion_matrix2)
  # Check if any class is missing in predictions
  if (length(levels(predictions)) < 2) {
    precision2[i] <- 0
    recall2[i] <- 0
    f1_score2[i] <- 0
  } else {
    precision2[i] <- confusion_matrix2[2, 2] / sum(confusion_matrix2[, 2])
    recall2[i] <- confusion_matrix2[2, 2] / sum(confusion_matrix2[2, ])
    f1_score2[i] <- 2 * precision2[i] * recall2[i] / (precision2[i] + recall2[i])
  }
}
# Print evaluation metrics
cat("Accuracy:", mean(accuracy2), "\n")
```

```
## Accuracy: 0.8865609
```

```r
cat("Precision:", mean(precision2), "\n")
```

```
## Precision: 0.9993701
```

```r
cat("Recall:", mean(recall2), "\n")
```

```
## Recall: 0.8870563
```

```r
cat("F1 Score:", mean(f1_score2), "\n")
```

```
## F1 Score: 0.9398696
```

```r
confusion_matrix2
```

```
##
## predictions   1   2
##           1   0   1
```

```
##           2  81 634
```

```
# Print model summary
summary(rf_model)
```

```
##                 Length Class  Mode
## call               4   -none- call
## type               1   -none- character
## predicted       2863   factor numeric
## err.rate         300   -none- numeric
## confusion          6   -none- numeric
## votes           5726   matrix numeric
## oob.times       2863   -none- numeric
## classes            2   -none- character
## importance       104   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               2863   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
# Calculate the baseline accuracy
baseline_accuracy <- max(table(dataset$p6b5)) / sum(table(dataset$p6b5))
# Print the baseline accuracy
print(paste("Baseline Accuracy:", baseline_accuracy))
```

```
## [1] "Baseline Accuracy: 0.887119307069014"
```

```
#employing F-beta score with beta of 0.5 as evaluation metric is beneficiary
#since it aligns perfectly with preference to prioritize maximizing true positives.
#It is also preferable to prioritize reducing false positives over reducing false negatives,
#because prioritizing precision over recall contributes to
#effective allocation of valuable mental health services.
#Based on the assumption that the baseline model predicts all points as class 1,
#this formula computes the baseline at:
#Baseline F-beta Score = 1.25 * Precision * Recall / (0.25 * Precision + Recall)
# Define the counts for each class
count_class_1 <- 404
count_class_0 <- 3175
# Calculate precision and recall for the baseline model
precision_baseline <- count_class_1 / (count_class_1 + count_class_0)
recall_baseline <- count_class_1 / (count_class_1 + 0)
# Calculate the baseline F-beta score
beta <- 0.25
Baseline_F_beta <- (1 + beta^2) * precision_baseline * recall_baseline /
  (beta^2 * precision_baseline + recall_baseline)
# Print the baseline F-beta score
print(paste("Baseline F-beta Score:", Baseline_F_beta))
```

```
## [1] "Baseline F-beta Score: 0.119095512242491"
```

```r
# Combine evaluation metrics into a single data frame
metrics_comparison <- data.frame(
  Model = c("GLM", "Random Forest"),
  Accuracy = c(mean(accuracy1), mean(accuracy2)),
  F1_Score = c(mean(f1_score1), mean(f1_score2)),
  Precision = c(mean(precision1), mean(precision2)),
  Recall = c(mean(recall1), mean(recall2)),
  Baseline_Accuracy = baseline_accuracy,
  Baseline_F_beta = Baseline_F_beta
)
# Print comparison table
print(metrics_comparison)
```

```
##           Model  Accuracy  F1_Score Precision    Recall Baseline_Accuracy
## 1           GLM 0.8787370 0.9353571 0.9892913 0.8870317         0.8871193
## 2 Random Forest 0.8865609 0.9398696 0.9993701 0.8870563         0.8871193
##   Baseline_F_beta
## 1       0.1190955
## 2       0.1190955
```

```r
# Get variable importance
rf_variable_importance <- importance(rf_model)
# Order the variable importance in decreasing order
rf_variable_importance <- rf_variable_importance[order(rf_variable_importance,
                                                       decreasing = TRUE),]
# Print variable importance
print("Variable Importance for Random Forest:")
```

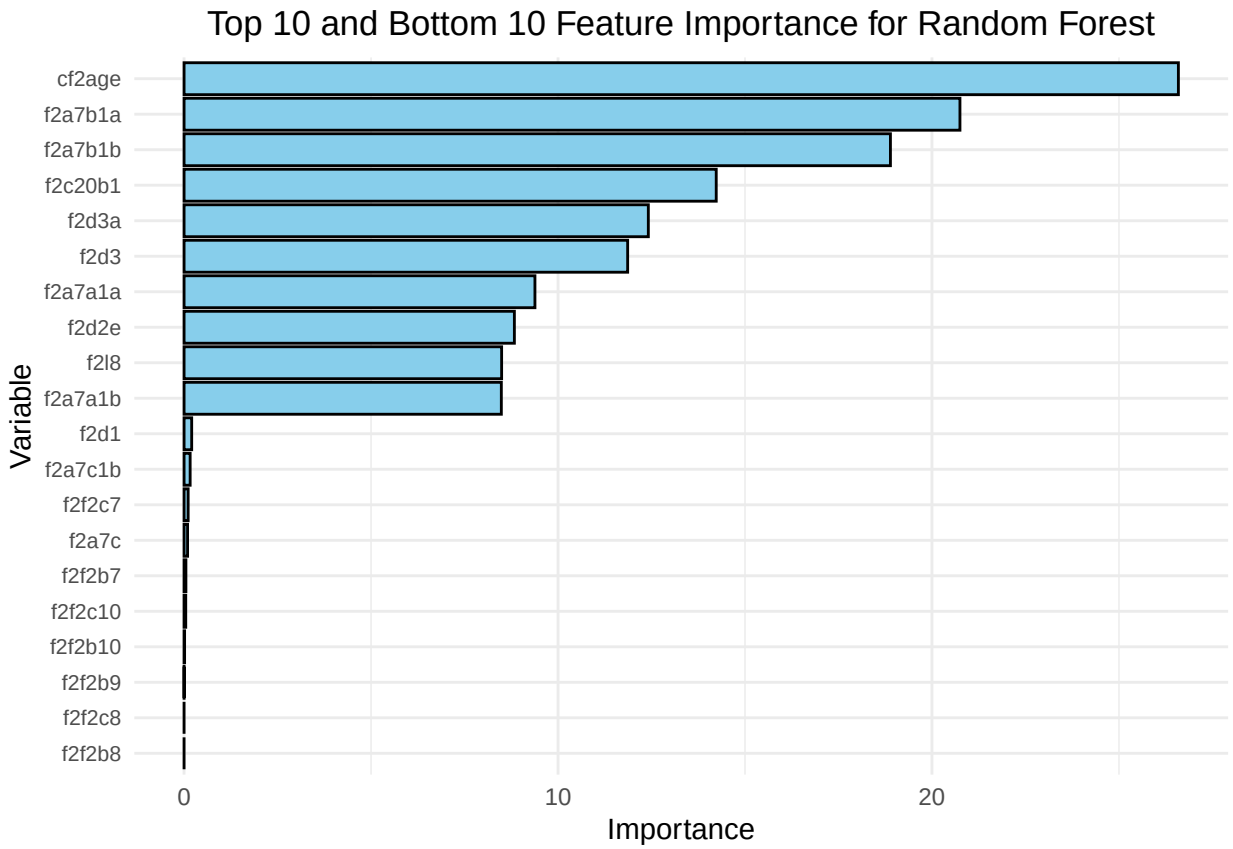
```
## [1] "Variable Importance for Random Forest:"
```

```r
print(rf_variable_importance)
```

```
##       cf2age      f2a7b1a      f2a7b1b      f2c20b1        f2d3a         f2d3
## 26.589675595 20.748234377 18.888593476 14.231009083 12.414732560 11.862474077
##      f2a7a1a        f2d2e         f2l8      f2a7a1b        f2d2c       f2f2c1
##  9.382949599  8.832732795  8.491727225  8.483943324  7.658771788  7.549178464
##      f2c20b2        f2d2d        f2d5a        f2d5c        f2a5a       f2f2c2
##  7.222144327  7.158623653  7.060674736  6.608317318  6.448806430  6.415250122
##        f2g1d         f2l7       f2a7d1        f2d5e       f2a6a2        f2d5d
##  6.413989869  6.240461489  5.964804626  5.660697371  5.149652709  4.997179513
##        f2d5i        f2d5b        f2d5g      f2a7e1a        f2a10       f2f2c3
##  4.888500171  4.798772123  4.705259501  4.245028818  3.938262321  3.918523150
##        f2d5h         f2a5         f2k7        f2d5f       f2f2b2      cf2span
##  3.622232287  3.216668465  3.173548239  3.093862623  2.822234403  2.756491316
##        f2b1a       f2f2b3       f2f2b1       f2f2c4         f2a6        f2c14
##  2.727654050  2.616747626  2.590305512  2.587830851  2.583259101  2.490170445
##      f2c13c3        f2g5b       f2c11a        f2d1x       f2e2a2      f2a7e1b
##  2.487228831  2.487190339  2.375328229  2.308586284  2.253460232  2.092636607
##         f2g5      f2c20b3        f2l8b        f2d7a        f2d7b        f2c17
##  2.014977729  1.973790085  1.925222251  1.900256380  1.884421640  1.762917019
##        f2d7i        f2g5a        f2d7c        f2a8h         f2c5       f2f2c5
##  1.706101088  1.641565024  1.592436825  1.581112480  1.545609416  1.515298300
##        f2d7h        f2d7e        f2d7g        f2d7d       f2f2b4        f2l6d
##  1.515081776  1.484964973  1.407338117  1.338586446  1.267091476  1.202516338
##        f2a6a         f2c8         f2d6        f2a8a        f2a7e         f2e1
```

```
##  1.182699567  1.152084723  1.075211986  1.071877519  1.050359524  0.997943967
##        f2d7f        f2l3         f2a8e        f2a7d        f2a7a       cf2cohm
##  0.992319857  0.942815584  0.913001914  0.867524416  0.841907519  0.827908447
##       f2a6a1       f2f2b5        f2a7b         f2a9      f2c20b4        f2a8b
##  0.787149618  0.749670038  0.731054827  0.617070982  0.556371265  0.502891330
##      f2a7c1a        f2f0         f2a8f        f2a8g       f2h16d       cf2marm
##  0.502754324  0.502025448  0.496764512  0.493998652  0.471966083  0.469738603
##         f2d4       f2f2c6       f2f2b6      f2c20b5         f2d1       f2a7c1b
##  0.421621643  0.391636872  0.230686680  0.230382208  0.204123067  0.162782338
##       f2f2c7        f2a7c        f2f2b7      f2f2c10      f2f2b10       f2f2b9
##  0.108892358  0.094394246  0.049329164  0.044882777  0.014476190  0.001050603
##       f2f2b8       f2f2c8
##  0.000000000  0.000000000
```

```r
# Get the top 5 and bottom 5 features
top_5_features <- names(head(rf_variable_importance, 10))
bottom_5_features <- names(tail(rf_variable_importance, 10))
# Combine top 5 and bottom 5 features
selected_features <- c(top_5_features, bottom_5_features)
# Filter the variable importance data frame to include only the selected features
rf_variable_importance_df <- data.frame(
  Variable = selected_features,
  Importance = rf_variable_importance[selected_features]
)
# Create ggplot
if (nrow(rf_variable_importance_df) > 0) {
  ggplot(rf_variable_importance_df, aes(x = reorder(Variable, Importance),
                                        y = Importance)) +
    geom_bar(stat = "identity", fill = "skyblue", color = "black") +
    coord_flip() +
    labs(title = "Top 10 and Bottom 10 Feature Importance for Random Forest",
         x = "Variable",
         y = "Importance") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5),
          axis.text.y = element_text(size = 8))
}
```




```r
# Number of bootstrap iterations
num_bootstrap <- 100
# Initialize lists to store evaluation metrics for each bootstrap iteration
bootstrap_accuracy <- vector("numeric", num_bootstrap)
bootstrap_precision <- vector("numeric", num_bootstrap)
bootstrap_recall <- vector("numeric", num_bootstrap)
bootstrap_f1_score <- vector("numeric", num_bootstrap)
# Loop through each bootstrap iteration
for (j in 1:num_bootstrap) {
  # Resample the dataset with replacement
  bootstrap_indices <- sample(1:nrow(dataset2), replace = TRUE)
  bootstrap_data <- dataset2[bootstrap_indices, ]
  # Create stratified folds on the bootstrapped dataset
  bootstrap_folds <- createFolds(bootstrap_data$p6b5, k = num_folds,
                                 list = TRUE, returnTrain = FALSE)
  # Initialize lists to store evaluation metrics for each fold
  accuracy <- vector("numeric", num_folds)
  precision <- vector("numeric", num_folds)
  recall <- vector("numeric", num_folds)
  f1_score <- vector("numeric", num_folds)
  # Loop through each fold
  for (i in 1:num_folds) {
    # Extract the indices for the current fold
    test_index <- unlist(bootstrap_folds[i])
    # Create training indices by excluding the test indices
    train_index <- setdiff(1:nrow(bootstrap_data), test_index)
    # Extract train and test data using the indices
```

```
    train_data <- bootstrap_data[train_index, ]
    test_data <- bootstrap_data[test_index, ]
    # Fit model
    rf_model <- randomForest(p6b5 ~ ., data = train_data, ntree = 100)
    # Make predictions on the test data
    predictions <- predict(rf_model, newdata = test_data)
    # Evaluate the model performance
    confusion_matrix <- table(predictions, test_data$p6b5)
    # Calculate evaluation metrics
    accuracy[i] <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
    # Check if any class is missing in predictions
    if (length(levels(predictions)) < 2) {
      precision[i] <- 0
      recall[i] <- 0
      f1_score[i] <- 0
    } else {
      precision[i] <- confusion_matrix[2, 2] / sum(confusion_matrix[, 2])
      recall[i] <- confusion_matrix[2, 2] / sum(confusion_matrix[2, ])
      f1_score[i] <- 2 * precision[i] * recall[i] / (precision[i] + recall[i])
    }
  }
  # Store mean metrics for the current bootstrap iteration
  bootstrap_accuracy[j] <- mean(accuracy)
  bootstrap_precision[j] <- mean(precision)
  bootstrap_recall[j] <- mean(recall)
  bootstrap_f1_score[j] <- mean(f1_score)
}
# Calculate confidence intervals for each metric
ci_accuracy <- quantile(bootstrap_accuracy, c(0.025, 0.975))
ci_precision <- quantile(bootstrap_precision, c(0.025, 0.975))
ci_recall <- quantile(bootstrap_recall, c(0.025, 0.975))
ci_f1_score <- quantile(bootstrap_f1_score, c(0.025, 0.975))
# Print evaluation metrics with confidence intervals
cat("Accuracy:", mean(bootstrap_accuracy),
    "(", ci_accuracy[1], "-", ci_accuracy[2], ")\n")
```

```
## Accuracy: 0.9225703 ( 0.9150605 - 0.9314125 )
```

```
cat("Precision:", mean(bootstrap_precision),
    "(", ci_precision[1], "-", ci_precision[2], ")\n")
```

```
## Precision: 0.9994862 ( 0.9981122 - 1 )
```

```
cat("Recall:", mean(bootstrap_recall),
    "(", ci_recall[1], "-", ci_recall[2], ")\n")
```

```
## Recall: 0.9201484 ( 0.9122858 - 0.92843 )
```

```
cat("F1 Score:", mean(bootstrap_f1_score),
    "(", ci_f1_score[1], "-", ci_f1_score[2], ")\n")
```

```
## F1 Score: 0.9581669 ( 0.9539143 - 0.962814 )
```

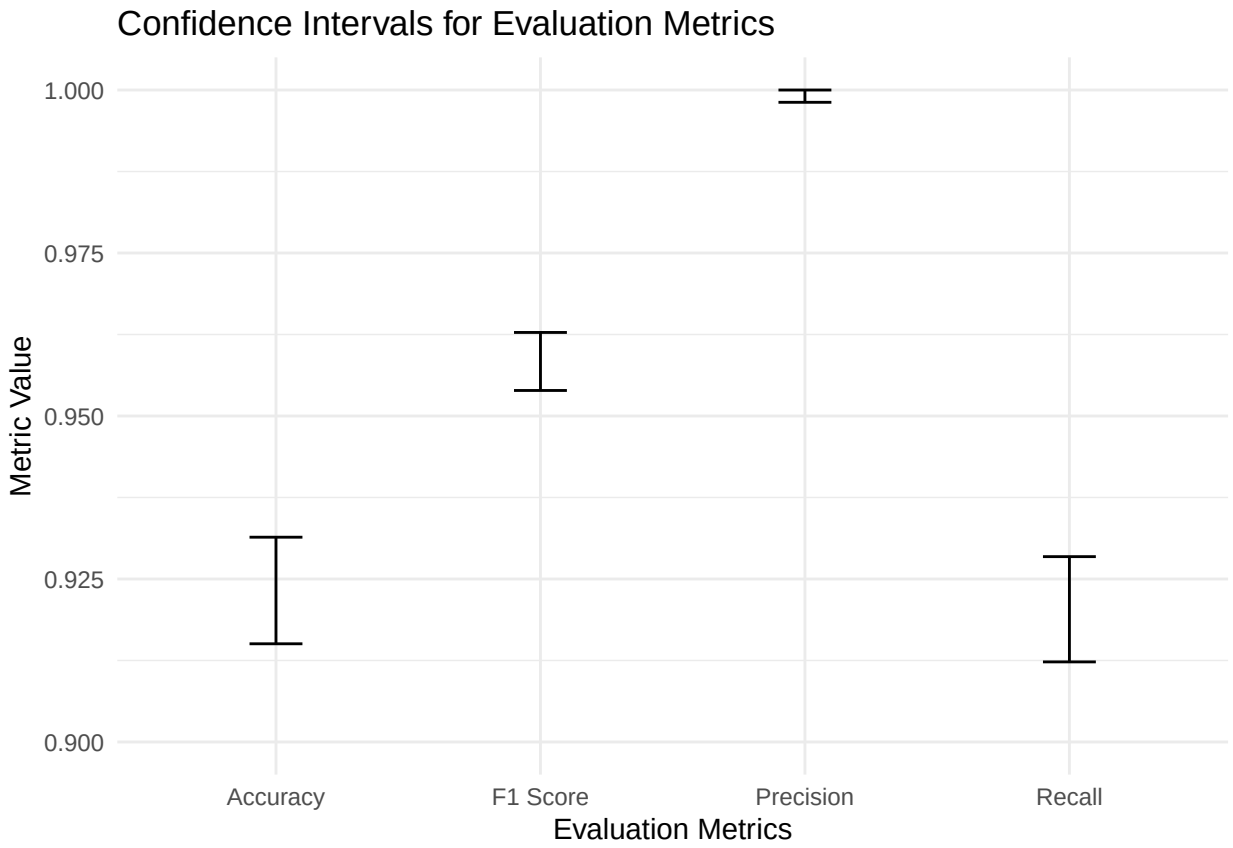
```
# Print model summary
summary(rf_model)
```

```
##                 Length Class  Mode
```

```
## call               4   -none- call
## type               1   -none- character
## predicted       2864   factor numeric
## err.rate         300   -none- numeric
## confusion          6   -none- numeric
## votes           5728   matrix numeric
## oob.times       2864   -none- numeric
## classes            2   -none- character
## importance       104   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               2864   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```r
# Define metrics and their confidence intervals
metrics <- c("Accuracy", "Precision", "Recall", "F1 Score")
ci_lower <- c(ci_accuracy[1], ci_precision[1], ci_recall[1], ci_f1_score[1])
ci_upper <- c(ci_accuracy[2], ci_precision[2], ci_recall[2], ci_f1_score[2])
# Create a data frame
ci_df <- data.frame(metrics, ci_lower, ci_upper)
# Plot the bar plot with error bars
ggplot(ci_df, aes(x = metrics, y = ci_upper)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.2,
                color = "black", position = position_dodge(0.9)) +
  labs(title = "Confidence Intervals for Evaluation Metrics",
       y = "Metric Value",
       x = "Evaluation Metrics") +
  theme_minimal() +
  ylim(0.9, 1.0)  # Set y-axis limits
```

```
## Warning: Removed 4 rows containing missing values (`geom_bar()`).
```




```
#data sources
#https://ffcws.princeton.edu/data-and-documentation/public-data-documentation
#publications and any previous work
#https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9407243/
```