FOREVER ALONE DATASET

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

KAGGLE CHALLENGE : Forever Alone Data

https://www.kaggle.com/kingburrito666/the-demographic-rforeveralone-dataset/data (https://www.kaggle.com/kingburrito666/the-demographic-rforeveralone-dataset/data)

WHY?

Last year a redditor created a survey to collect demographic data on the subreddit /r/ForeverAlone. Since then they have deleted their account but they left behind their dataset.

Candidates

1. Survey Results – Stackflow Developers Data
2. Forever Alone
3. Texas Execution

Selection Process

Here is my selection criterias for picking the data-set. 1. Size and Volume 2. Interest Factor (Scale 1 to 5) 3. Social Relevance & Impact (Scale 1 to 5)

Dataset	Size & Volume	Interest Factor	Social Relevance
Survey Results	51392x154, 90966 KB	3	2
Forever Alone	469x19, 108 KB	5	5
exas Execution	545x23, 270 KB	4	4

```
Winner           Forever Alone
```

Exploration

1. Examine Data-set and its size

2. Evaluate the variables data types
3. Identify the ones requiring data type conversion
    a. Time to timestamp
4. Boolean Variables
    a. Prosistitution_legal
    b. virgin
    c. social_fear
    d. depressed
    e. attempt_suicide
5. Variable Factor Conversion & Factor Value Ordering
    a. Pay_for_sex
    b. Income
    c. Race
    d. BodyWeight
    e. Gender
    f. Sexuality
    g. employment
    h. job_title
    i. edu_level
6. Feature Engineering
    a. Age Binning
    b. Friends Binning
7. Continuous Variables Analysis
    a. Age
    b. Friends
8. Text Variables
    a. what_help_from_others
    b. improve_yourself_how

Initial Configuration

```
library(ggplot2)
library(tidyverse)
```

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

```
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(okcupiddata)
library(faraway)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(stringr)
library(NHANES)
library(mdsr)
```

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

```
##
## Attaching package: 'lattice'
```

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

```
## Loading required package: ggformula
```

```
## Loading required package: ggstance
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affec
ted by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
##
## Have you tried the ggformula package for your plots?
```

```
##
## In accordance with CRAN policy, the 'mdsr' package
##            no longer attaches
## the 'tidyverse' package automatically.
## You may need to 'library(tidyverse)' in order to
##            use certain functions.
```

```
library(rpart)
```

```
##
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:faraway':
##
##     solder
```

```
library(partykit)
```

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

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

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

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

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

```
library(class)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```
library(rms)
```

```
## Loading required package: Hmisc
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following objects are masked from 'package:faraway':
##
##     rats, solder
```

```
## Loading required package: Formula
```

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

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

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

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

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

```
##
## Attaching package: 'rms'
```

```
## The following object is masked from 'package:faraway':
##
##     vif
```

```
library(naniar)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'lmtest'
```

```
## The following object is masked from 'package:rms':
##
##     lrtest
```

```
library(pdftools)
```

```
## Using poppler version 0.73.0
```

```
library(Dict)
library(haven)
library(writexl)
library(expss)
```

```
## Registered S3 methods overwritten by 'expss':
##   method                 from
##   [.labelled             Hmisc
##   as.data.frame.labelled base
##   print.labelled         Hmisc
```

```
##
## Use 'expss_output_viewer()' to display tables in the RStudio Viewer.
##  To return to the console output, use 'expss_output_default()'.
```

```
##
## Attaching package: 'expss'
```

```
## The following objects are masked from 'package:haven':
##
##     is.labelled, read_spss
```

```
## The following object is masked from 'package:naniar':
##
##     .where
```

```
## The following object is masked from 'package:mosaic':
##
##     prop
```

```
## The following objects are masked from 'package:stringr':
##
##     fixed, regex
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, compute, contains, first, last, na_if, recode, vars
```

```
## The following objects are masked from 'package:purrr':
##
##     keep, modify, modify_if, transpose, when
```

```
## The following objects are masked from 'package:tidyr':
##
##     contains, nest
```

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

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(na.tools)
```

```
##
## Attaching package: 'na.tools'
```

```
## The following objects are masked from 'package:naniar':
##
##     all_na, any_na, is_na, which_na
```

```
#library(RWeka)
#library(openNLP)
```

Reading the data file

```
## Rows: 469
## Columns: 19
## $ time                  <chr> "5/17/2016 20:04:18", "5/17/2016 20:04:30", "...
## $ gender                <chr> "Male", "Male", "Male", "Male", "Male", "Male...
## $ sexuallity            <chr> "Straight", "Bisexual", "Straight", "Straight...
## $ age                   <int> 35, 21, 22, 19, 23, 24, 22, 24, 20, 33, 32, 2...
## $ income                <chr> "$30,000 to $39,999", "$1 to $10,000", "$0", ...
## $ race                  <chr> "White non-Hispanic", "White non-Hispanic", "...
## $ bodyweight            <chr> "Normal weight", "Underweight", "Overweight",...
## $ virgin                <chr> "Yes", "Yes", "Yes", "Yes", "No", "Yes", "Yes...
## $ prostitution_legal    <chr> "No", "No", "No", "Yes", "No", "No", "No", "N...
## $ pay_for_sex           <chr> "No", "No", "No", "No", "Yes and I have", "Ye...
## $ friends               <dbl> 0, 0, 10, 8, 10, 2, 2, 10, 0, 6, 5, 0, 20, 1,...
## $ social_fear           <chr> "Yes", "Yes", "Yes", "Yes", "No", "Yes", "Yes...
## $ depressed             <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye...
## $ what_help_from_others <chr> "wingman/wingwoman, Set me up with a date", "...
## $ attempt_suicide       <chr> "Yes", "No", "No", "No", "No", "Yes", "No", "...
## $ employment            <chr> "Employed for wages", "Out of work and lookin...
## $ job_title             <chr> "mechanical drafter", "-", "unemployed", "stu...
## $ edu_level             <chr> "Associate degree", "Some college, no degree"...
## $ improve_yourself_how  <chr> "None", "join clubs/socual clubs/meet ups", "...
```

Identifying and printing the variables with NA counts

```
for (col in names(FA_data)) {
   NULL_count <- sum(is.na(FA_data[col]))
   msg <- str_c('Variable :',  col, ' With Null Values = ', NULL_count)
   print(msg)
}
```

```
## [1] "Variable :time With Null Values = 0"
## [1] "Variable :gender With Null Values = 0"
## [1] "Variable :sexuallity With Null Values = 0"
## [1] "Variable :age With Null Values = 0"
## [1] "Variable :income With Null Values = 0"
## [1] "Variable :race With Null Values = 0"
## [1] "Variable :bodyweight With Null Values = 0"
## [1] "Variable :virgin With Null Values = 0"
## [1] "Variable :prostitution_legal With Null Values = 0"
## [1] "Variable :pay_for_sex With Null Values = 0"
## [1] "Variable :friends With Null Values = 0"
## [1] "Variable :social_fear With Null Values = 0"
## [1] "Variable :depressed With Null Values = 0"
## [1] "Variable :what_help_from_others With Null Values = 0"
## [1] "Variable :attempt_suicide With Null Values = 0"
## [1] "Variable :employment With Null Values = 0"
## [1] "Variable :job_title With Null Values = 0"
## [1] "Variable :edu_level With Null Values = 0"
## [1] "Variable :improve_yourself_how With Null Values = 0"
```

Data Type Conversion - Char To TimeStamp

```
FA_data$time <- as_date(FA_data$time, format = "%m/%d/%Y %H:%M:%S")
head(FA_data$time)
```

```
## [1] "2016-05-17" "2016-05-17" "2016-05-17" "2016-05-17" "2016-05-17"
## [6] "2016-05-17"
```

Feature Engineering

Creatomg 2 Variables for identifying and counting the subjects

Value Resetting

```
FA_data$subjectId <- 1:469
FA_data$subjectCount <- 1
```

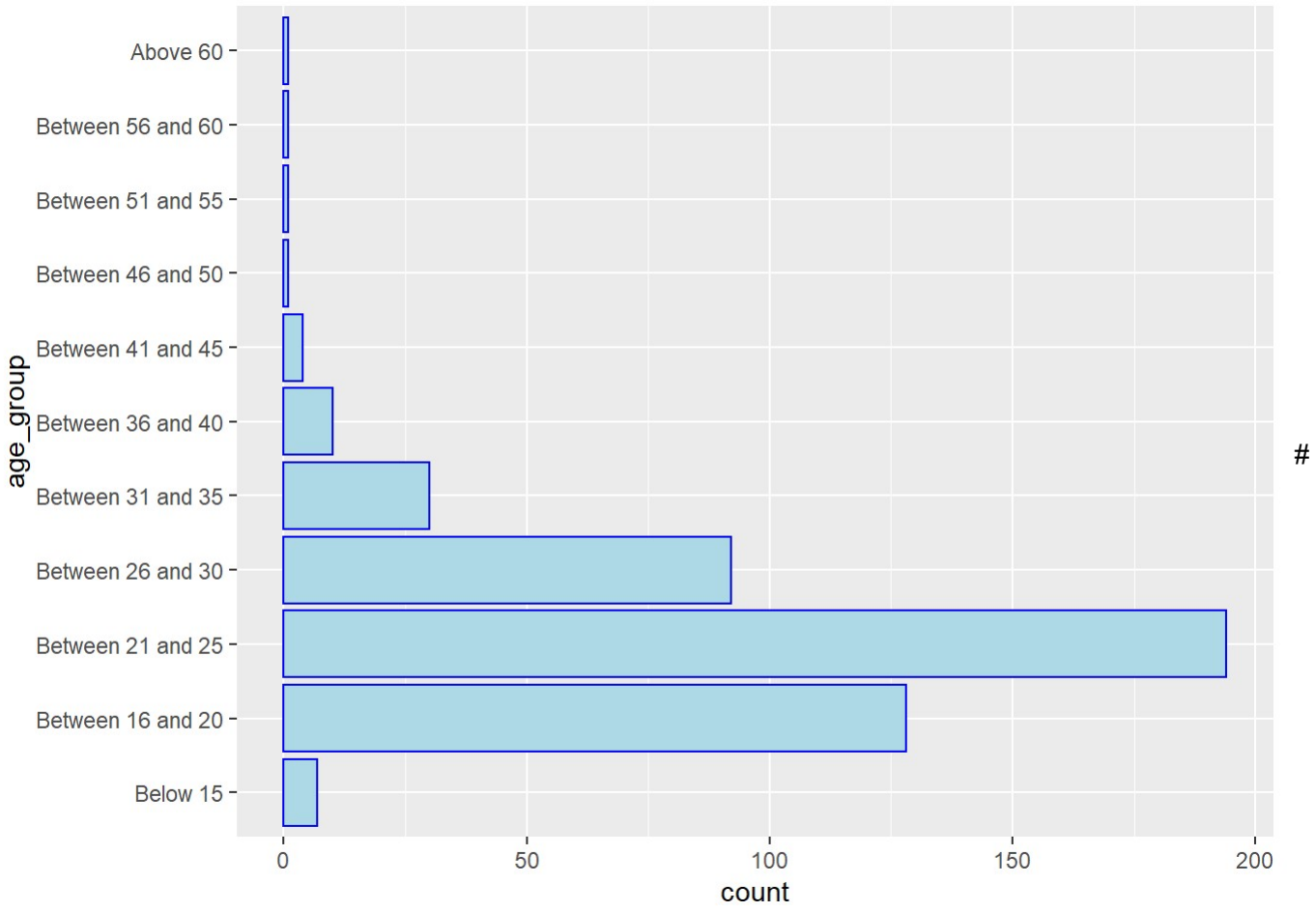
Age Groups : EDA

```
table(FA_data$age)
```

```
##
## 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
##  2  1  4  5 13 39 32 39 41 40 46 37 30 26 15 21 14 16 10  5  6  3  6  2  2  4
## 39 41 42 44 45 48 55 57 70
##  2  1  1  1  1  1  1  1  1
```

Creating age_group bins

```
FA_data$age_group       <- case_when( age <= 15             ~ 'Below 15',
                                      age >= 16 & age <= 20 ~ 'Between 16 and 20',
                                      age >= 21 & age <= 25 ~ 'Between 21 and 25',
                                      age >= 26 & age <= 30 ~ 'Between 26 and 30',
                                      age >= 31 & age <= 35 ~ 'Between 31 and 35',
                                      age >= 36 & age <= 40 ~ 'Between 36 and 40',
                                      age >= 41 & age <= 45 ~ 'Between 41 and 45',
                                      age >= 46 & age <= 50 ~ 'Between 46 and 50',
                                      age >= 51 & age <= 55 ~ 'Between 51 and 55',
                                      age >= 56 & age <= 60 ~ 'Between 56 and 60',
TRUE                    ~ 'Above 60' )

age_group_levels <- c('Below 15', 'Between 16 and 20', 'Between 21 and 25',
                      'Between 26 and 30', 'Between 31 and 35', 'Between 36 and 40',
                      'Between 41 and 45', 'Between 46 and 50', 'Between 51 and 55',
                      'Between 56 and 60', 'Above 60')

FA_data$age_group <- factor(FA_data$age_group, levels = age_group_levels)

ggplot(data = FA_data, aes(y = age_group))  + geom_bar(color = "blue", fill = "light
blue")
```

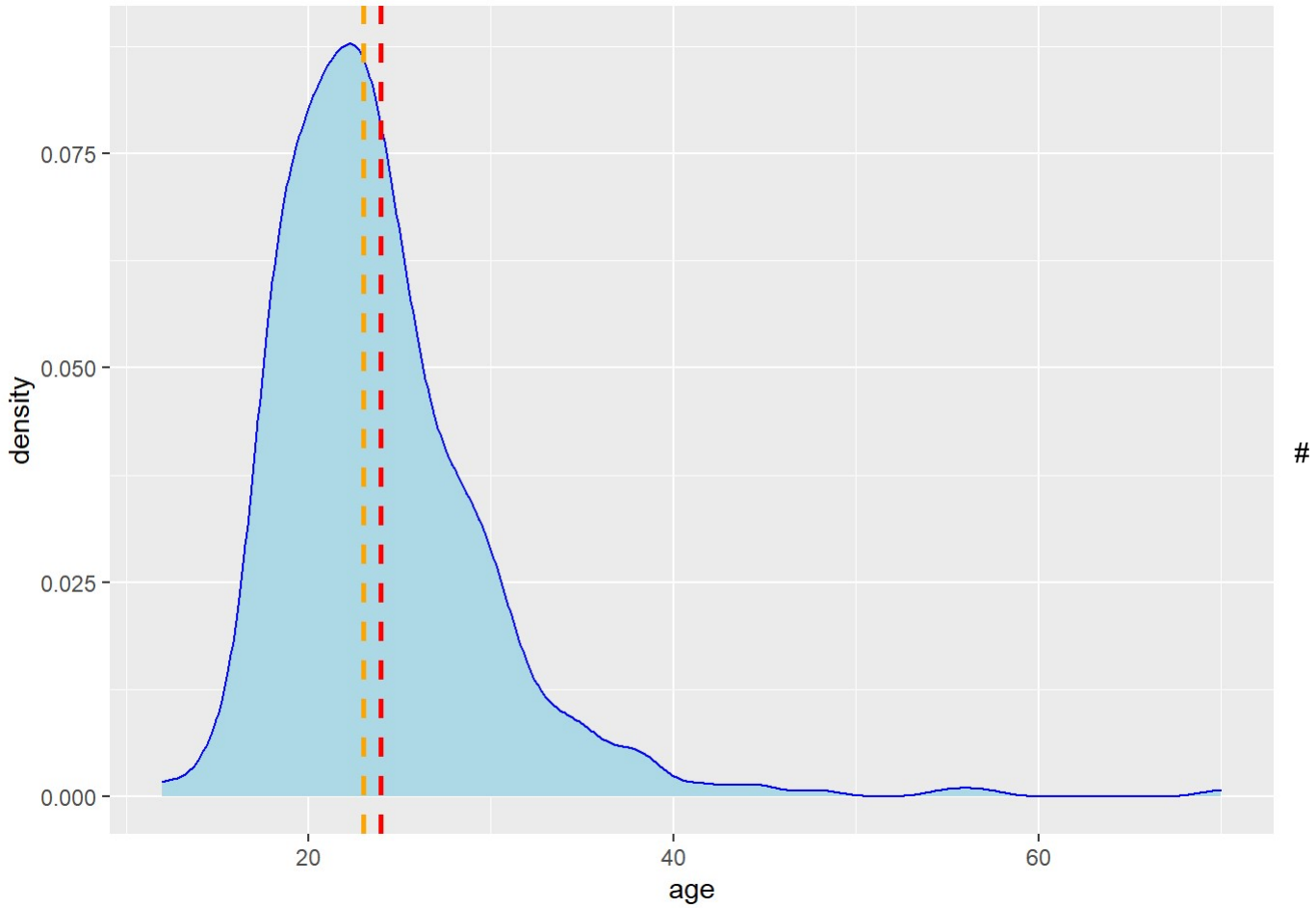


# #

——————————————————————————— # age: age distribution #

———————————————————————————

```
ggplot(data = FA_data, aes(x = age))  + geom_density(color = "blue", fill = "light bl
ue") +
      geom_vline(aes(xintercept=mean(age, na.rm=T)),   # Ignore NA values for mean
                 color="red", linetype="dashed", size=1) +
        geom_vline(aes(xintercept=median(age, na.rm=T)),   # Ignore NA values for mea
n
                 color="orange", linetype="dashed", size=1)
```



#

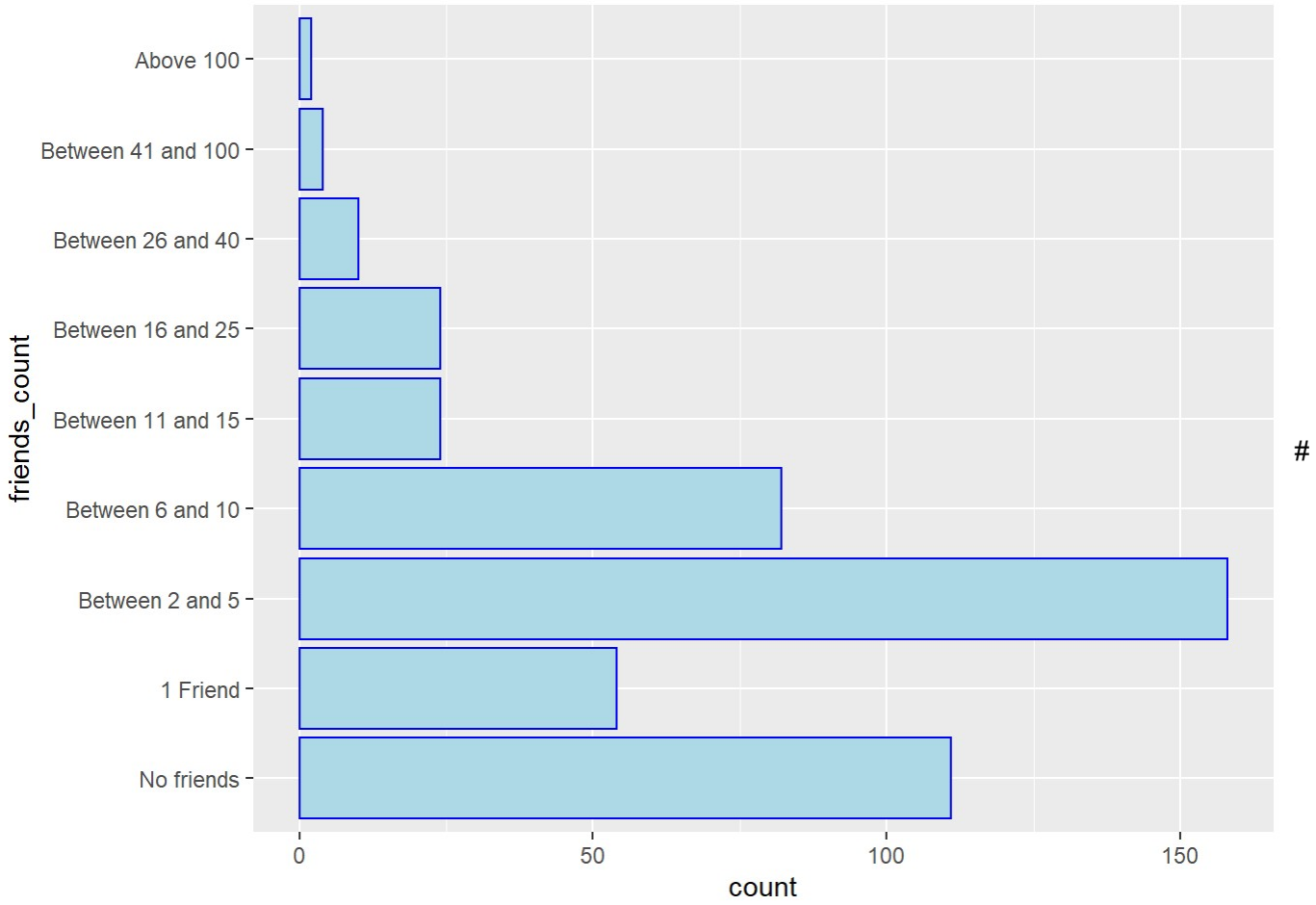
——————————————————————————— # Friends : EDA #
———————————————————————————

```
table(FA_data$friends)
```

```
##
##   0 0.2 0.5   1   2   3   4   5   6   7   8   9  10  11  12  13  15  16  17  18
## 109   1   1  54  42  44  27  45  26  10  10   2  34   1   8   3  12   3   2   1
##  20  25  28  30  40  60  80 100 400 600
##  16   2   1   6   3   1   1   2   1   1
```

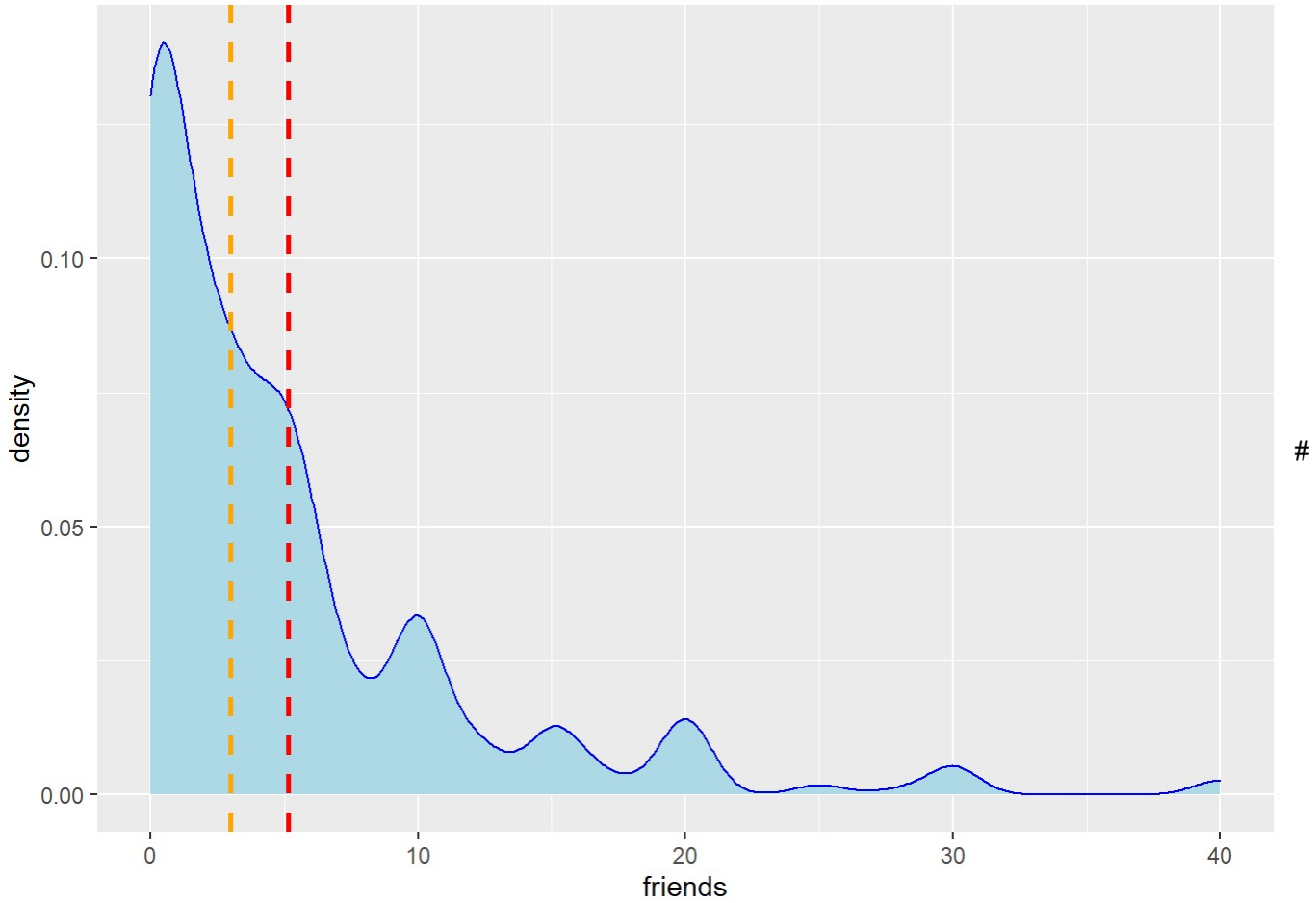
Creating friends_count bins

```
FA_data$friends_count   <- case_when(
                                    friends  < 01                  ~ 'No friends',
                                    friends == 01                  ~ '1 Friend',
                                    friends >= 02 & friends <= 05  ~ 'Between 2 and
5',
                                    friends >= 06 & friends <= 10  ~ 'Between 6 and
10',
                                    friends >= 11 & friends <= 15  ~ 'Between 11 an
d 15',
                                    friends >= 16 & friends <= 25  ~ 'Between 16 an
d 25',
                                    friends >= 26 & friends <= 40  ~ 'Between 26 an
d 40',
                                    friends >= 41 & friends <= 100 ~ 'Between 41 an
d 100',
                                    TRUE                           ~ 'Above 100' )

friends_count_levels <- c('No friends', '1 Friend', 'Between 2 and 5', 'Between 6 and
10',
                                    'Between 11 and 15', 'Between 16 and 25', 'Betw
een 26 and 40',
                                    'Between 41 and 100', 'Above 100')

FA_data$friends_count <- factor(FA_data$friends_count, levels = friends_count_levels)

ggplot(data = FA_data, aes(y = friends_count))  + geom_bar(color = "blue", fill = "li
ght blue")
```



friends: distribution

```
FA_data %>% filter(FA_data$friends <= 40) %>%
ggplot(aes(x = friends))  + geom_density(color = "blue", fill = "light blue") +
      geom_vline(aes(xintercept=mean(friends, na.rm=T)),   # Ignore NA values for mea
n
               color="red", linetype="dashed", size=1) +
        geom_vline(aes(xintercept=median(friends, na.rm=T)),   # Ignore NA values for
mean
               color="orange", linetype="dashed", size=1)
```



#

———————————————————————————— # Re-organizing the data-set #
————————————————————————————

```
col_list <- c('subjectId', 'time', 'gender', 'sexuallity',  'age', 'age_group', 'inco
me',
              'race', 'bodyweight', 'virgin', 'prostitution_legal', 'pay_for_sex', 'f
riends',
              'friends_count', 'social_fear', 'depressed', 'what_help_from_others', '
attempt_suicide',
              'employment', 'job_title', 'edu_level', 'improve_yourself_how', 'subjec
tCount')

FA_data <- FA_data[col_list]

rem_list <- c('time', 'age', 'friends')
#fa_ds <- FA_data[, !(names(FA_data) %in% rem_list)]

glimpse(FA_data)
```

```
## Rows: 469
## Columns: 23
## $ subjectId             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
## $ time                  <date> 2016-05-17, 2016-05-17, 2016-05-17, 2016-05-...
## $ gender                <chr> "Male", "Male", "Male", "Male", "Male", "Male...
## $ sexuallity            <chr> "Straight", "Bisexual", "Straight", "Straight...
## $ age                   <int> 35, 21, 22, 19, 23, 24, 22, 24, 20, 33, 32, 2...
## $ age_group             <fct> Between 31 and 35, Between 21 and 25, Between...
## $ income                <chr> "$30,000 to $39,999", "$1 to $10,000", "$0", ...
## $ race                  <chr> "White non-Hispanic", "White non-Hispanic", "...
## $ bodyweight            <chr> "Normal weight", "Underweight", "Overweight",...
## $ virgin                <chr> "Yes", "Yes", "Yes", "Yes", "No", "Yes", "Yes...
## $ prostitution_legal    <chr> "No", "No", "No", "Yes", "No", "No", "No", "N...
## $ pay_for_sex           <chr> "No", "No", "No", "No", "Yes and I have", "Ye...
## $ friends               <dbl> 0, 0, 10, 8, 10, 2, 2, 10, 0, 6, 5, 0, 20, 1,...
## $ friends_count         <fct> No friends, No friends, Between 6 and 10, Bet...
## $ social_fear           <chr> "Yes", "Yes", "Yes", "Yes", "No", "Yes", "Yes...
## $ depressed             <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye...
## $ what_help_from_others <chr> "wingman/wingwoman, Set me up with a date", "...
## $ attempt_suicide       <chr> "Yes", "No", "No", "No", "No", "Yes", "No", "...
## $ employment            <chr> "Employed for wages", "Out of work and lookin...
## $ job_title             <chr> "mechanical drafter", "-", "unemployed", "stu...
## $ edu_level             <chr> "Associate degree", "Some college, no degree"...
## $ improve_yourself_how  <chr> "None", "join clubs/socual clubs/meet ups", "...
## $ subjectCount          <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
```

EDA and Standardization Process for logical variables

```
f_boolean_EDA <- function(df_col) {
  df_col <- ifelse(toupper(str_trim(df_col, side = 'both')) == "YES", TRUE, FALSE)
  table(df_col)
}
```

---

prostitution_legal: Converting Yes/No to Boolean/Logical

```
f_boolean_EDA(FA_data$prostitution_legal)
```

```
## df_col
## FALSE  TRUE
##   361   108
```

Social_fear: Converting Yes/No to Boolean/Logical

```
f_boolean_EDA(FA_data$social_fear)
```

```
## df_col
## FALSE  TRUE
##   161   308
```

Depressed: Social_fear: Converting Yes/No to Boolean/Logical

```
f_boolean_EDA(FA_data$depressed)
```

```
## df_col
## FALSE  TRUE
##   157   312
```

Attempt_Suicide: Social_fear: Converting Yes/No to Boolean/Logical

```
f_boolean_EDA(FA_data$attempt_suicide)
```

```
## df_col
## FALSE  TRUE
##   384    85
```

Virgin : Converting Yes/No to Boolean/Logical

```
f_boolean_EDA(FA_data$virgin)
```

```
## df_col
## FALSE  TRUE
##   117   352
```

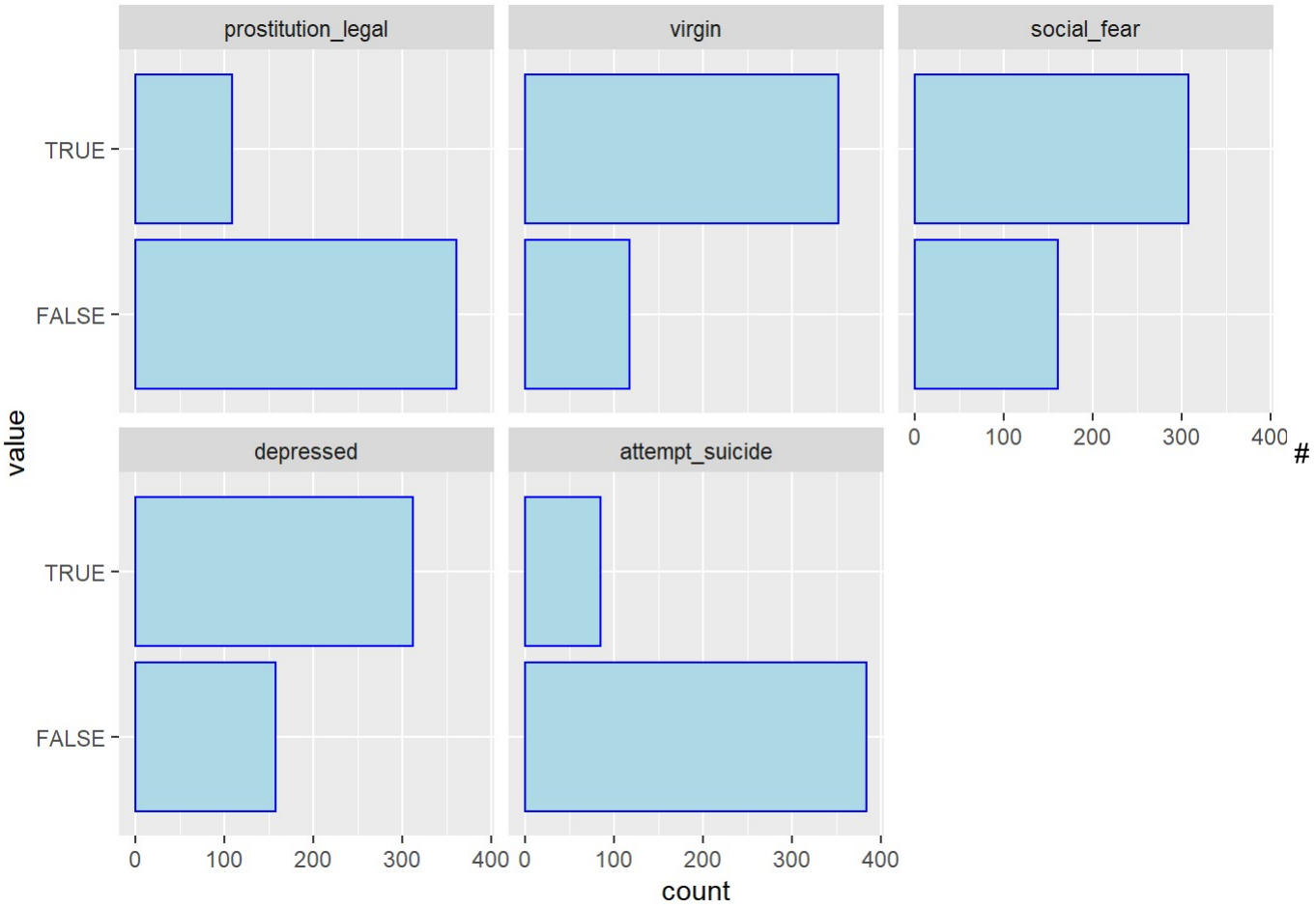
Converting Yes/No Variables to Boolean

```
bool_list <- c('prostitution_legal','virgin', 'social_fear', 'depressed', 'attempt_su
icide')

FA_data$prostitution_legal <- ifelse(FA_data$prostitution_legal == 'Yes', TRUE, FALS
E)
FA_data$virgin             <- ifelse(FA_data$virgin             == 'Yes', TRUE, FALS
E)
FA_data$social_fear        <- ifelse(FA_data$social_fear        == 'Yes', TRUE, FALS
E)
FA_data$depressed          <- ifelse(FA_data$depressed          == 'Yes', TRUE, FALS
E)
FA_data$attempt_suicide    <- ifelse(FA_data$attempt_suicide    == 'Yes', TRUE, FALS
E)

glimpse(FA_data)
```

```
## Rows: 469
## Columns: 23
## $ subjectId             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
## $ time                  <date> 2016-05-17, 2016-05-17, 2016-05-17, 2016-05-...
## $ gender                <chr> "Male", "Male", "Male", "Male", "Male", "Male...
## $ sexuallity            <chr> "Straight", "Bisexual", "Straight", "Straight...
## $ age                   <int> 35, 21, 22, 19, 23, 24, 22, 24, 20, 33, 32, 2...
## $ age_group             <fct> Between 31 and 35, Between 21 and 25, Between...
## $ income                <chr> "$30,000 to $39,999", "$1 to $10,000", "$0", ...
## $ race                  <chr> "White non-Hispanic", "White non-Hispanic", "...
## $ bodyweight            <chr> "Normal weight", "Underweight", "Overweight",...
## $ virgin                <lgl> TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TR...
## $ prostitution_legal    <lgl> FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALS...
## $ pay_for_sex           <chr> "No", "No", "No", "No", "Yes and I have", "Ye...
## $ friends               <dbl> 0, 0, 10, 8, 10, 2, 2, 10, 0, 6, 5, 0, 20, 1,...
## $ friends_count         <fct> No friends, No friends, Between 6 and 10, Bet...
## $ social_fear           <lgl> TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TR...
## $ depressed             <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRU...
## $ what_help_from_others <chr> "wingman/wingwoman, Set me up with a date", "...
## $ attempt_suicide       <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, TRUE, FALSE...
## $ employment            <chr> "Employed for wages", "Out of work and lookin...
## $ job_title             <chr> "mechanical drafter", "-", "unemployed", "stu...
## $ edu_level             <chr> "Associate degree", "Some college, no degree"...
## $ improve_yourself_how  <chr> "None", "join clubs/socual clubs/meet ups", "...
## $ subjectCount          <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
```

Visualization for 5 logical variables

```
bool_list <- c( 'subjectId', 'prostitution_legal','virgin', 'social_fear', 'depressed
', 'attempt_suicide')

fa_bool_ds <- FA_data[bool_list] %>%  melt(id=(c("subjectId"))) %>% dplyr::select(var
iable, value, subjectId)

ggplot(data = fa_bool_ds, aes(y = value))  + geom_bar(color = "blue", fill = 'light b
lue') + facet_wrap(~variable, nrow = 2)
```

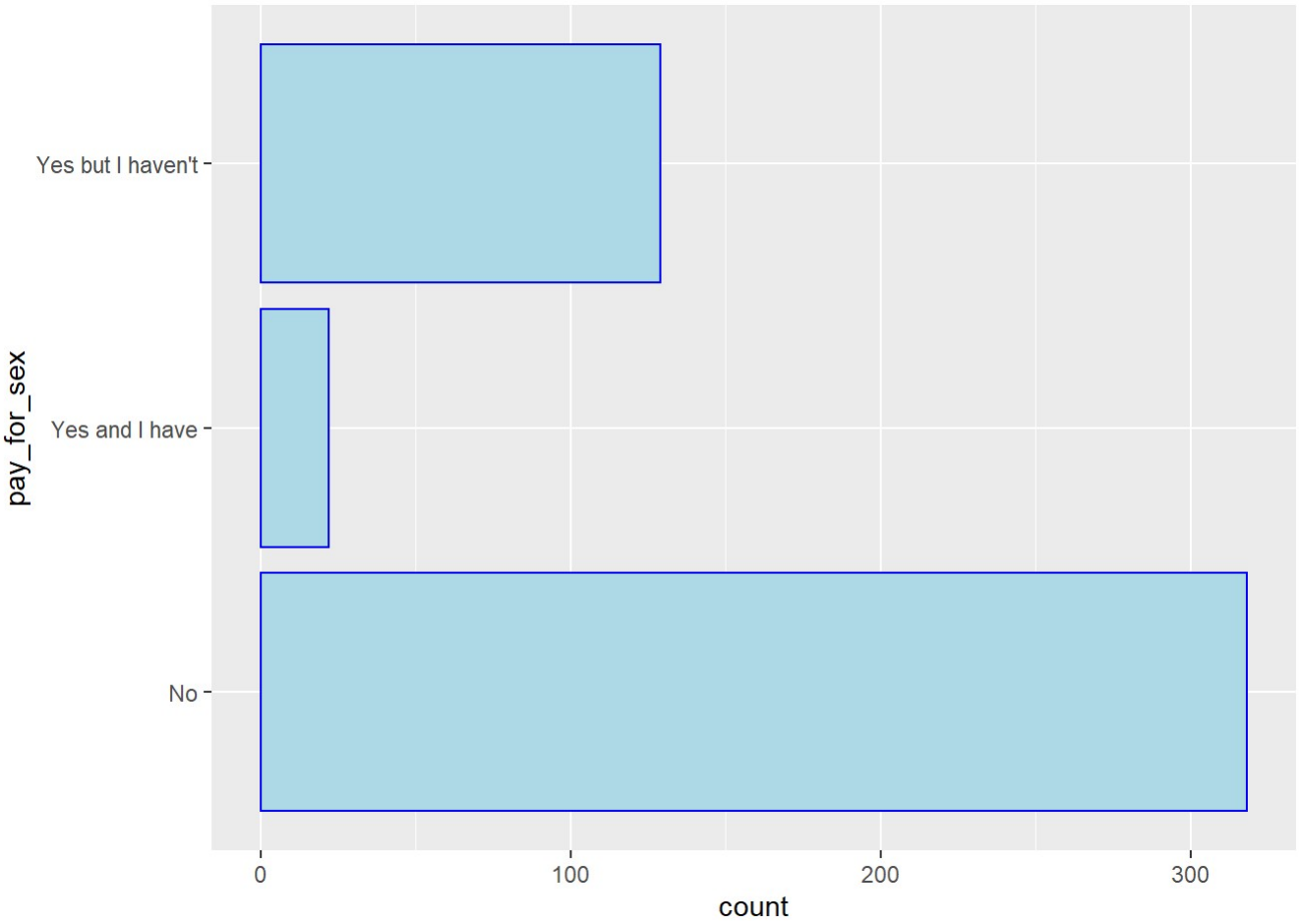


———————————————————————————— # Function for performing EDA on classification variables # ————————————————————————————

```
f_classifier_EDA <- function(df_col) {
  df_col <- tools::toTitleCase(str_trim(df_col, side = 'both'))
}
```

pay_for_sex : EDA & Factorization

```
f_classifier_EDA(FA_data$pay_for_sex)

pay_for_sex_levels <- c( "No", "Yes and I have", "Yes but I haven't")

FA_data$pay_for_sex <- factor(FA_data$pay_for_sex, levels = pay_for_sex_levels)

ggplot(data = FA_data, aes(y = pay_for_sex))  + geom_bar(color = "blue", fill = "ligh
t blue")
```

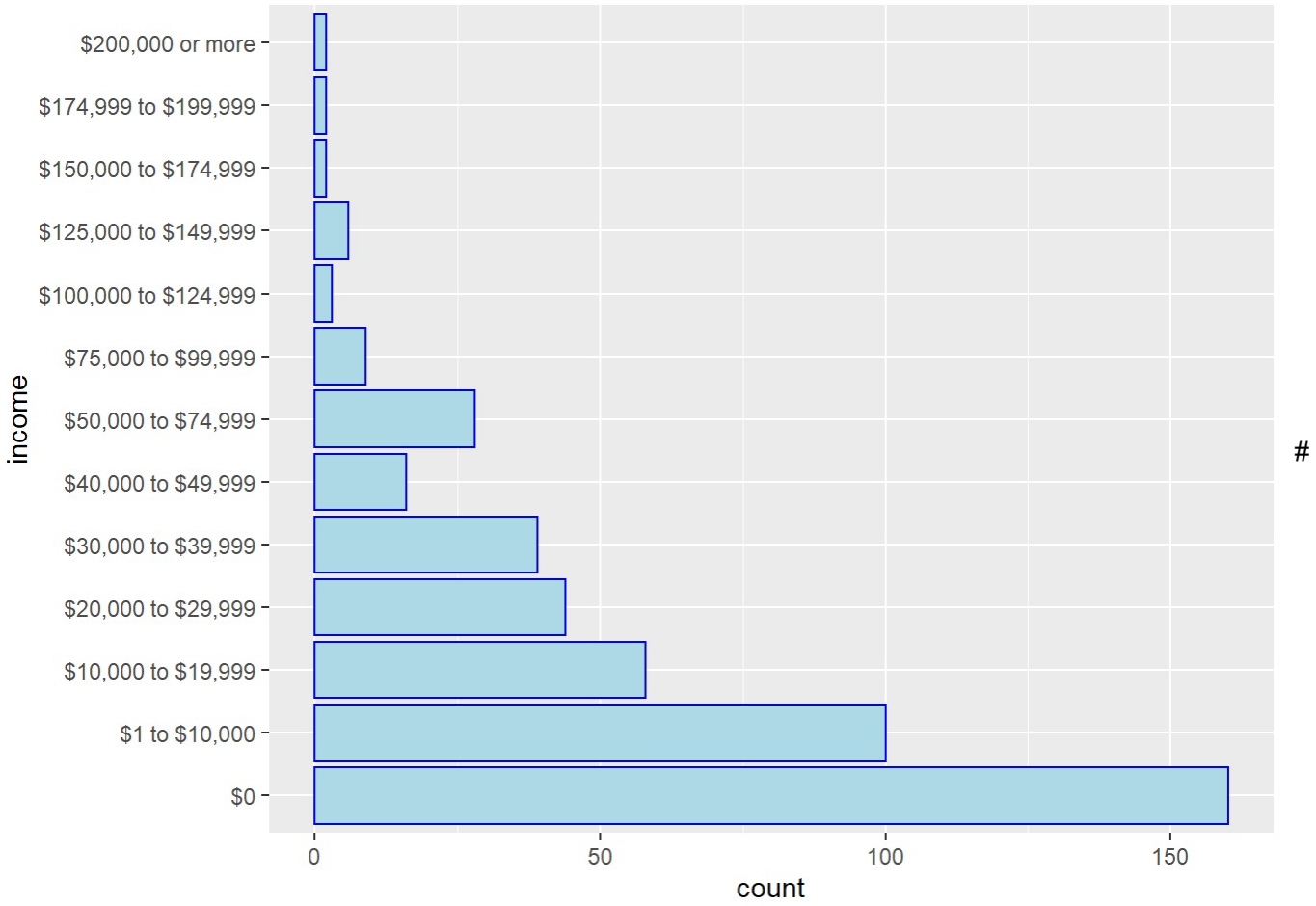



```
table(FA_data$income)
```

```
##
##                   $0         $1 to $10,000   $10,000 to $19,999
##                  160                   100                   58
## $100,000 to $124,999 $125,000 to $149,999 $150,000 to $174,999
##                    3                     6                     2
## $174,999 to $199,999   $20,000 to $29,999     $200,000 or more
##                    2                    44                     2
##   $30,000 to $39,999   $40,000 to $49,999   $50,000 to $74,999
##                   39                    16                    28
##   $75,000 to $99,999
##                    9
```

income: EDA & Factorizing

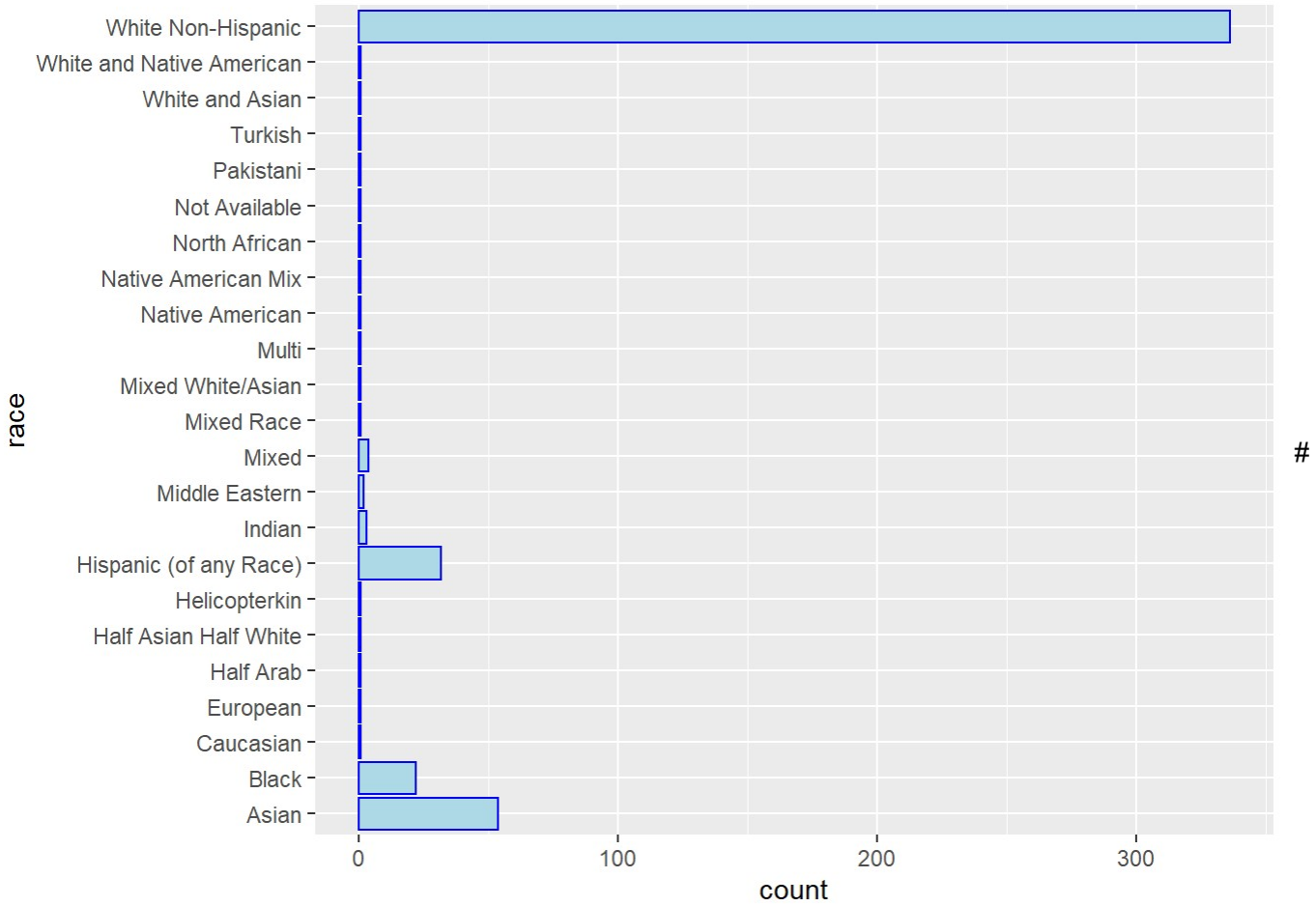
```
FA_data$income <- na.replace(FA_data$income, 'No Income')

f_classifier_EDA(FA_data$income)

income_levels <- c( 'No Income', '$0', '$1 to $10,000', '$10,000 to $19,999', '$20,00
0 to $29,999',
                    '$30,000 to $39,999', '$40,000 to $49,999', '$50,000 to $74,999',
                    '$75,000 to $99,999', '$100,000 to $124,999', '$125,000 to $149,9
99',
                    '$150,000 to $174,999', '$174,999 to $199,999', '$200,000 or more
')
FA_data$income <- factor(FA_data$income, levels = income_levels)

ggplot(data = FA_data, aes(y = income))  + geom_bar(color = "blue", fill = "light blu
e")
```



#

———————————————————————————————— # race : EDA & Factorization #
————————————————————————————————

```
f_classifier_EDA(FA_data$race)

FA_data$race <- ifelse(FA_data$race == 'First two answers. Gender is androgyne, not m
ale; sexuality is asexual, not bi.', 'Not Available', FA_data$race)

FA_data$race <- factor(tools::toTitleCase(FA_data$race))

ggplot(data = FA_data, aes(y = race))  + geom_bar(color = "blue", fill = "light blu
e")
```



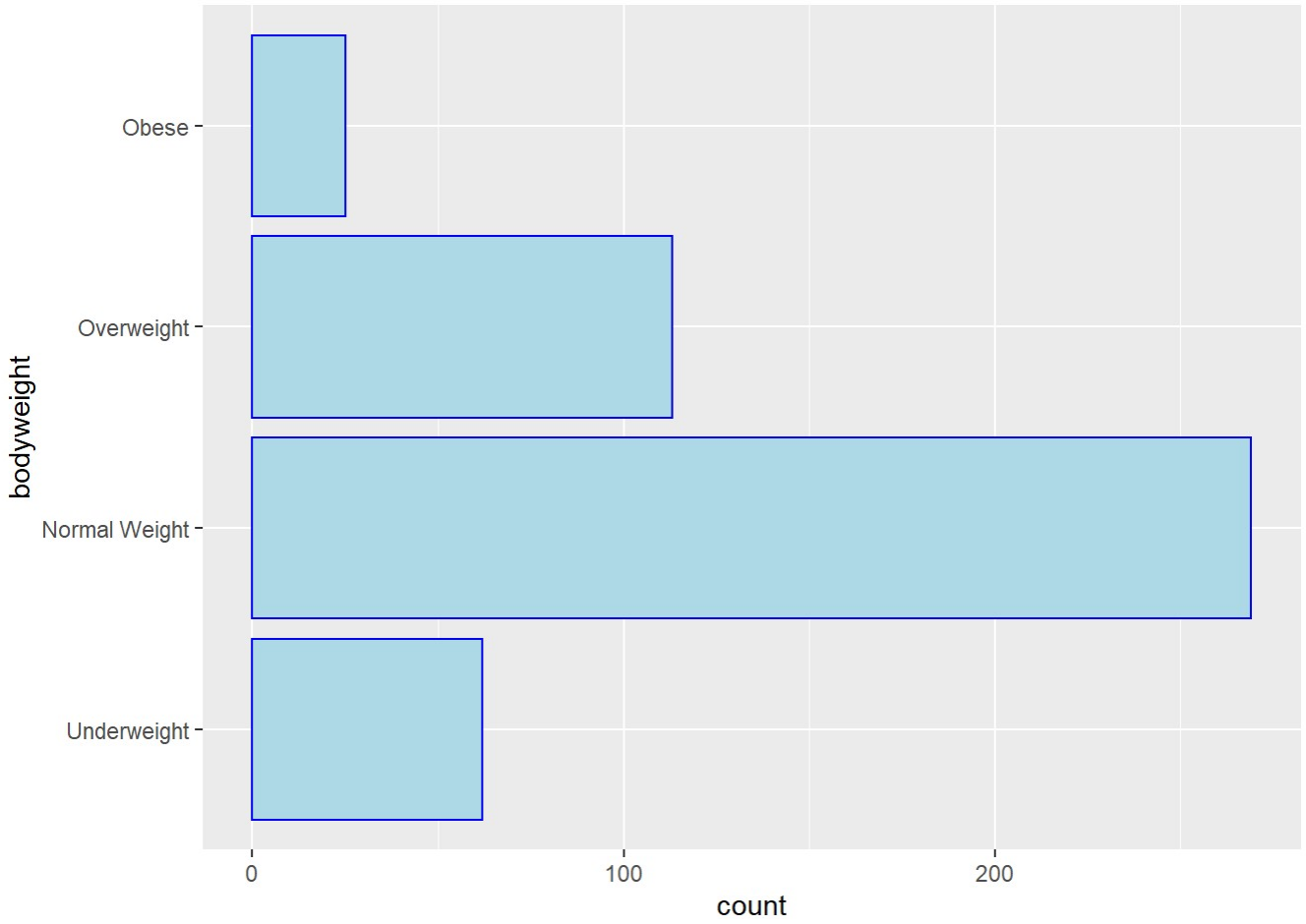
———————————————————————————— # bodyweight : EDA & Factorization # ————————————————————————————

```
f_classifier_EDA(FA_data$bodyweight)
```

---

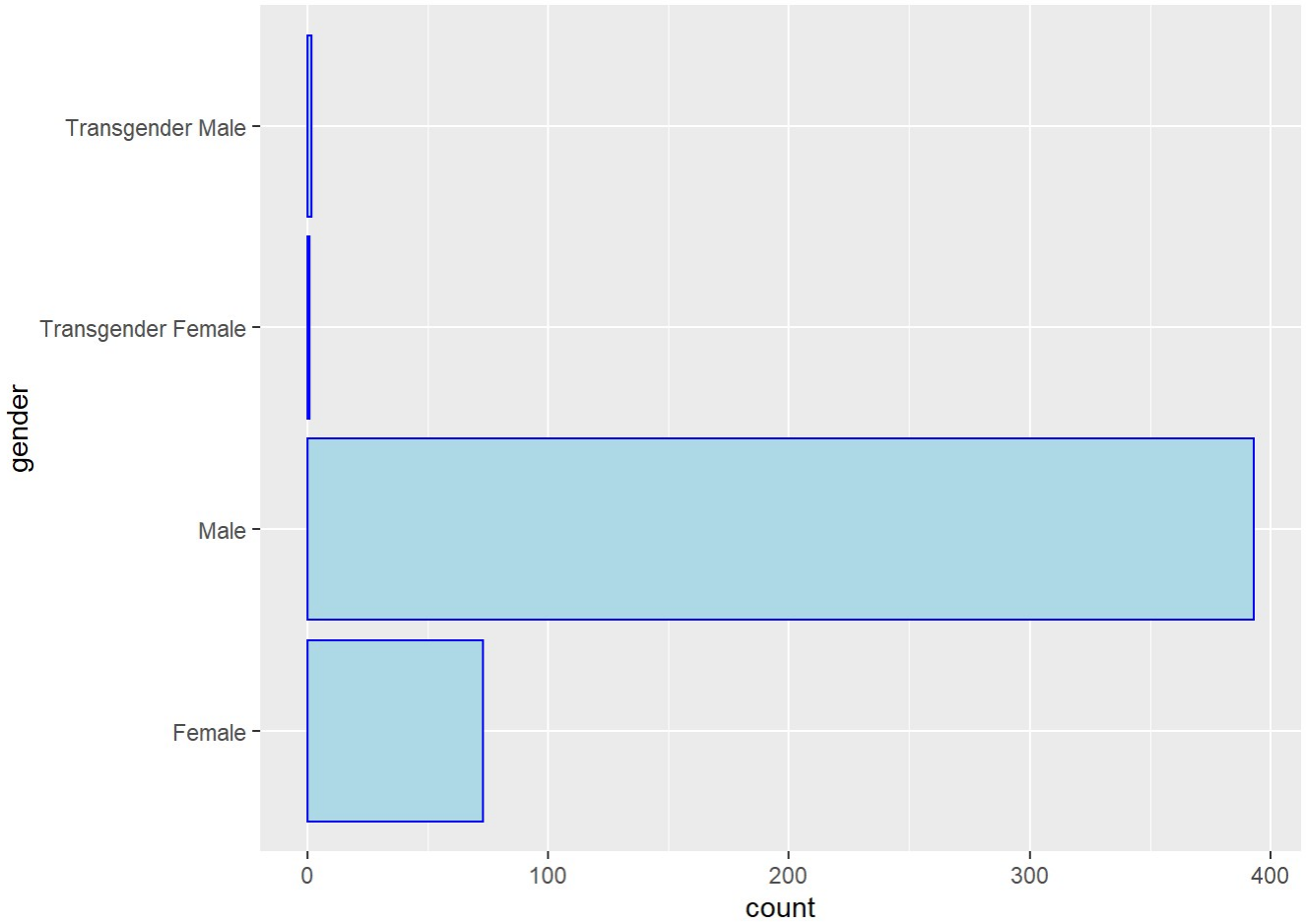
bodyweight : EDA & Factorization

```
bodyweight_levels <- c( "Underweight", "Normal Weight", "Overweight", "Obese")
FA_data$bodyweight <- factor(tools::toTitleCase(FA_data$bodyweight), levels = bodywei
ght_levels)

ggplot(data = FA_data, aes(y = bodyweight))  + geom_bar(color = "blue", fill = "light
blue")
```



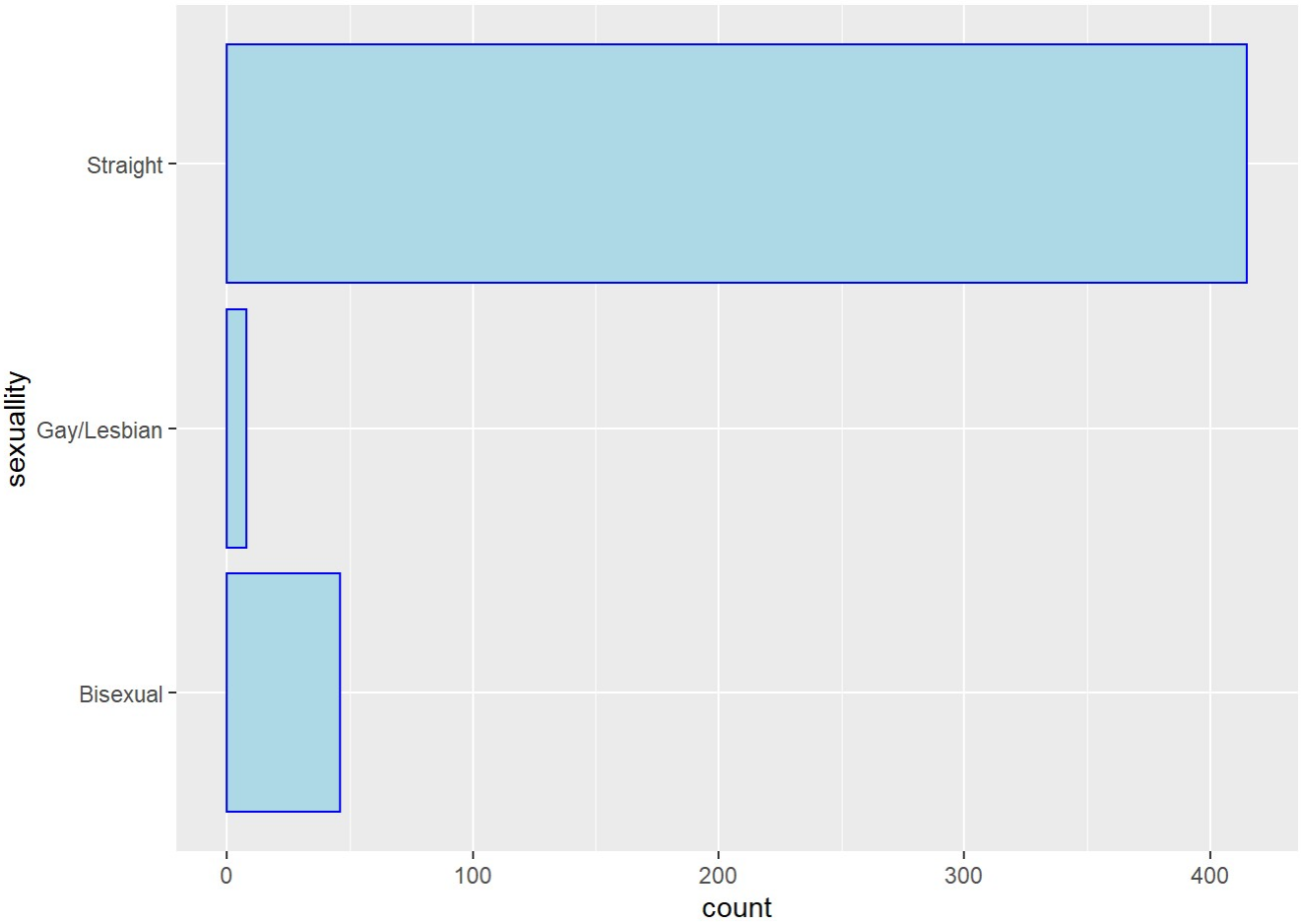
gender : EDA & Factorization

```
f_classifier_EDA(FA_data$gender)
FA_data$gender <- factor(tools::toTitleCase(FA_data$gender))

ggplot(data = FA_data, aes(y = gender))  + geom_bar(color = "blue", fill = "light blu
e")
```



sexuallity : EDA & Factorization

```
f_classifier_EDA(FA_data$sexuallity)
FA_data$sexuallity <- factor(tools::toTitleCase(FA_data$sexuallity))

ggplot(data = FA_data, aes(y = sexuallity))  + geom_bar(color = "blue", fill = "light
blue")
```



what_help_from_others : EDA and Factorization

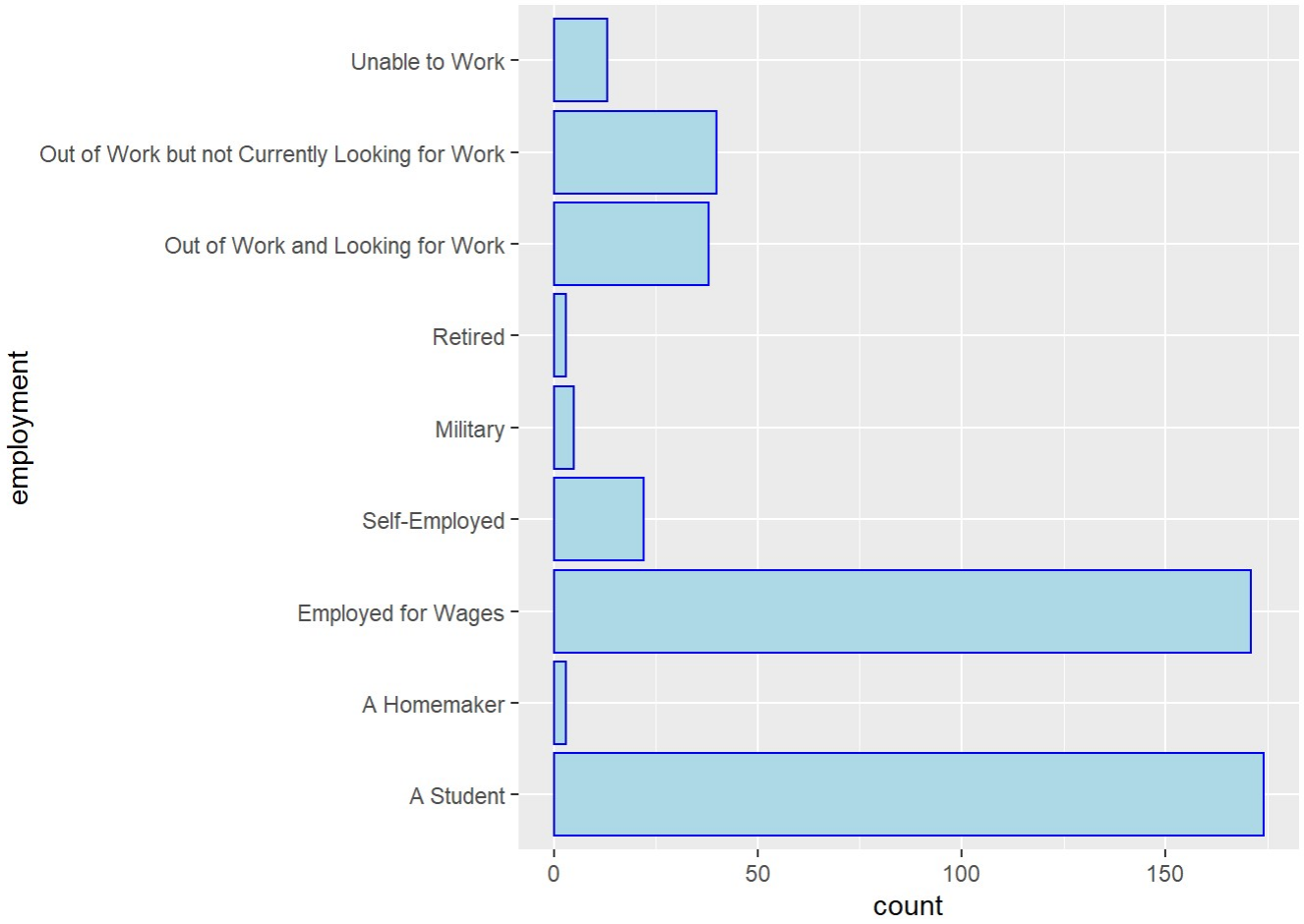
```
f_classifier_EDA(FA_data$what_help_from_others)
FA_data$what_help_from_others <- factor(tools::toTitleCase(FA_data$what_help_from_oth
ers))
#table(FA_data$what_help_from_others)
#ggplot(data = FA_data, aes(y = what_help_from_others))  + geom_bar(color = "blue", f
ill = "light blue")

#table(FA_data$what_help_from_others)
```

employment : EDA and Factorization

```
f_classifier_EDA(FA_data$employment)
FA_data$employment <- tools::toTitleCase(FA_data$employment)
```

employment : EDA and Factorization

```
employment_levels = c('A Student', 'A Homemaker', 'Employed for Wages', 'Self-Employe
d',
                      'Military', 'Retired', 'Out of Work and Looking for Work',
                      'Out of Work but not Currently Looking for Work', 'Unable to Wo
rk')

FA_data$employment <- factor(tools::toTitleCase(FA_data$employment), levels = employm
ent_levels)

ggplot(data = FA_data, aes(y = employment))  + geom_bar(color = "blue", fill = "light
blue")
```



job_title : EDA and Factorization

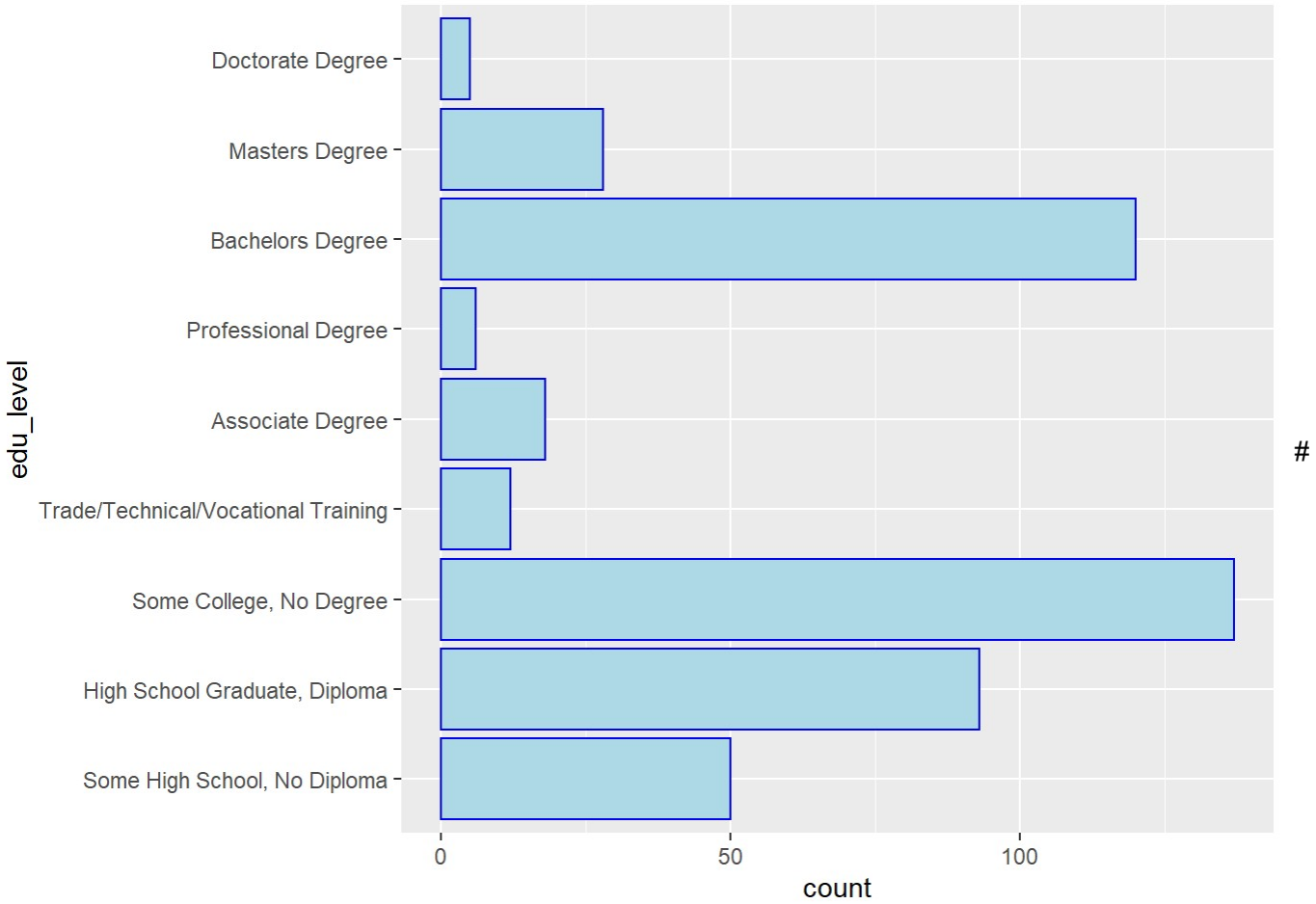
```
f_classifier_EDA(FA_data$job_title)

FA_data$job_title <- ifelse(FA_data$job_title == 'Coo', 'Cook', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'Admin', 'Administrator', FA_data$jo
b_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'Admin Assistant', 'Administrative A
ssistant', FA_data$job_title)

FA_data$job_title <- ifelse(FA_data$job_title == '-', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == '--', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == '---', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'n/a', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'Na', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'NEET', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'No', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'No Job', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'None', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'None (?)', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'Not Disclosing This', 'NA', FA_dat
a$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == 'Not Telling', 'NA', FA_data$job_tit
le)
FA_data$job_title <- ifelse(FA_data$job_title == 'Nothing', 'NA', FA_data$job_title)

FA_data$job_title <- ifelse(FA_data$job_title == '', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == '*', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == '.', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == '...', 'NA', FA_data$job_title)
FA_data$job_title <- ifelse(FA_data$job_title == '/', 'NA', FA_data$job_title)

FA_data$job_title <- factor(FA_data$job_title)

#ggplot(data = FA_data, aes(y = job_title))  + geom_bar(color = "blue", fill = "light
blue")
```

edu_level : EDA and Factorization

```
table(FA_data$edu_level)
```

```
##
##                                                  Associate degree
##                                                                18
##                                               Bachelorâ\200\231s degree
##                                                               120
##                                                  Doctorate degree
##                                                                 5
## High school graduate, diploma or the equivalent (for example: GED)
##                                                                93
##                                                Masterâ\200\231s degree
##                                                               28
##                                               Professional degree
##                                                                6
##                                           Some college, no degree
##                                                              137
##                                       Some high school, no diploma
##                                                               50
##                                  Trade/technical/vocational training
##                                                               12
```

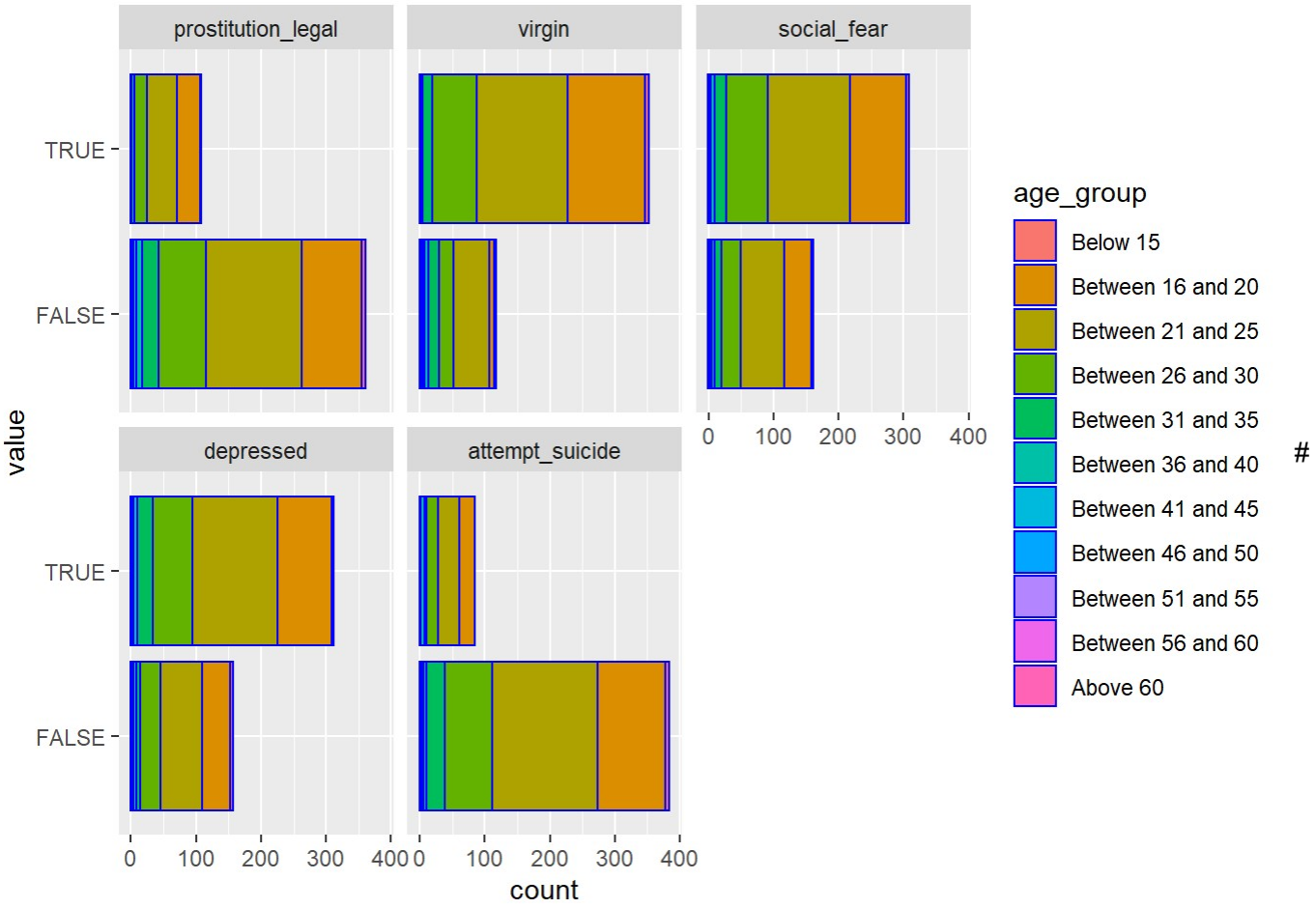
```
FA_data$edu_level <- na.replace(FA_data$edu_level, 'No Education')

f_classifier_EDA(FA_data$edu_level)

FA_data$edu_level <- ifelse(FA_data$edu_level == 'Masterâ€™s degree', 'Masters Degree
', FA_data$edu_level)
FA_data$edu_level <- ifelse(FA_data$edu_level == 'Bachelorâ€™s degree', 'Bachelors De
gree', FA_data$edu_level)
FA_data$edu_level <- ifelse(FA_data$edu_level == 'High school graduate, diploma or th
e equivalent (for example: GED)', 'High School Graduate, Diploma', FA_data$edu_level)

f_classifier_EDA(FA_data$edu_level)

edu_levels <- c('No Education',
                'Some High School, No Diploma',
                'High School Graduate, Diploma',
                'Some College, No Degree',
                'Trade/Technical/Vocational Training',
                'Associate Degree',
                'Professional Degree',
                'Bachelors Degree',
                'Masters Degree',
                'Doctorate Degree')

FA_data$edu_level <- factor(tools::toTitleCase(FA_data$edu_level), levels = edu_level
s)

ggplot(data = FA_data, aes(y = edu_level))  + geom_bar(color = "blue", fill = "light
blue")
```



———————————————————————————————— # Visualization for 5 logical variables by age_group

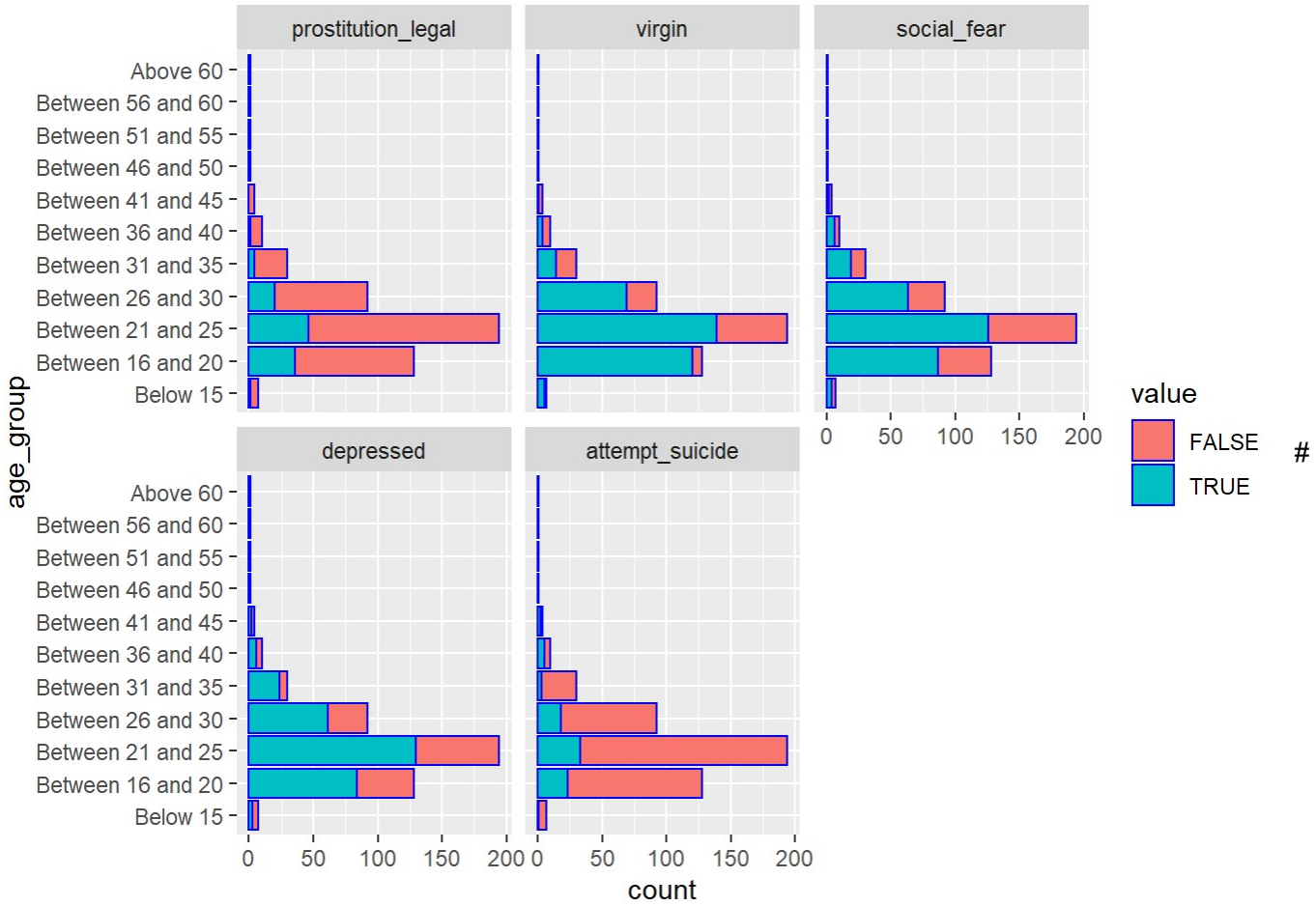
————————————————————————————————

```
bool_list <- c( 'subjectId', 'age_group', 'prostitution_legal','virgin', 'social_fear
', 'depressed', 'attempt_suicide')

FA_data[bool_list] %>%  melt(id=(c("subjectId", "age_group"))) %>% dplyr::select(vari
able, value, subjectId, age_group) %>% ggplot(aes(y = value, fill = age_group))  + ge
om_bar(color = "blue") + facet_wrap(~variable, nrow = 2)
```



Visualization for 5 logical variables by age_group

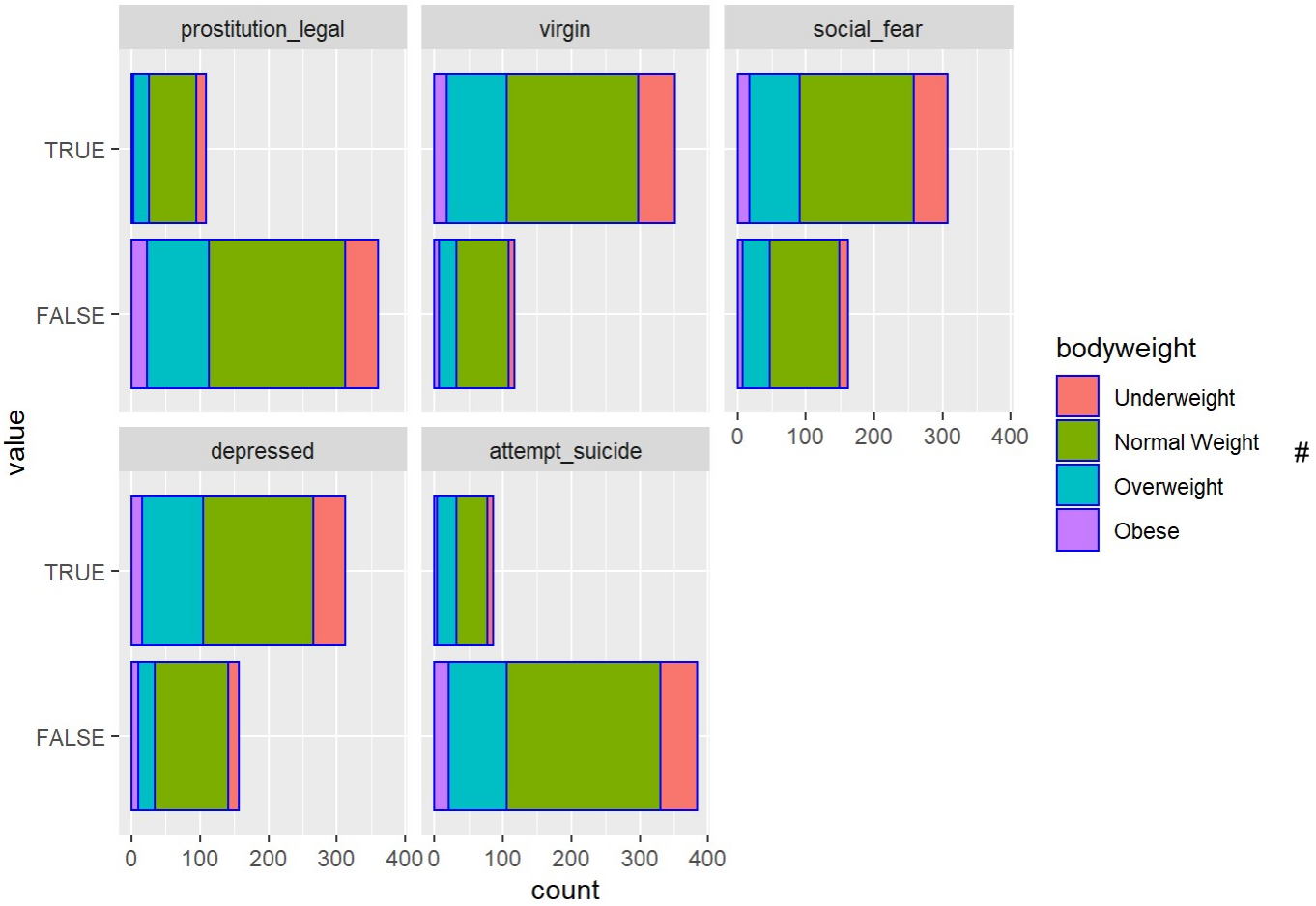
```
bool_list <- c( 'subjectId', 'age_group', 'prostitution_legal','virgin', 'social_fear
', 'depressed', 'attempt_suicide')

FA_data[bool_list] %>%  melt(id=(c("subjectId", "age_group"))) %>% dplyr::select(vari
able, value, subjectId, age_group) %>% ggplot(aes(y = age_group, fill = value))  + ge
om_bar(color = "blue") + facet_wrap(~variable, nrow = 2)
```



———————————————————————————— # Visualization for 5 logical variables by bodyweight

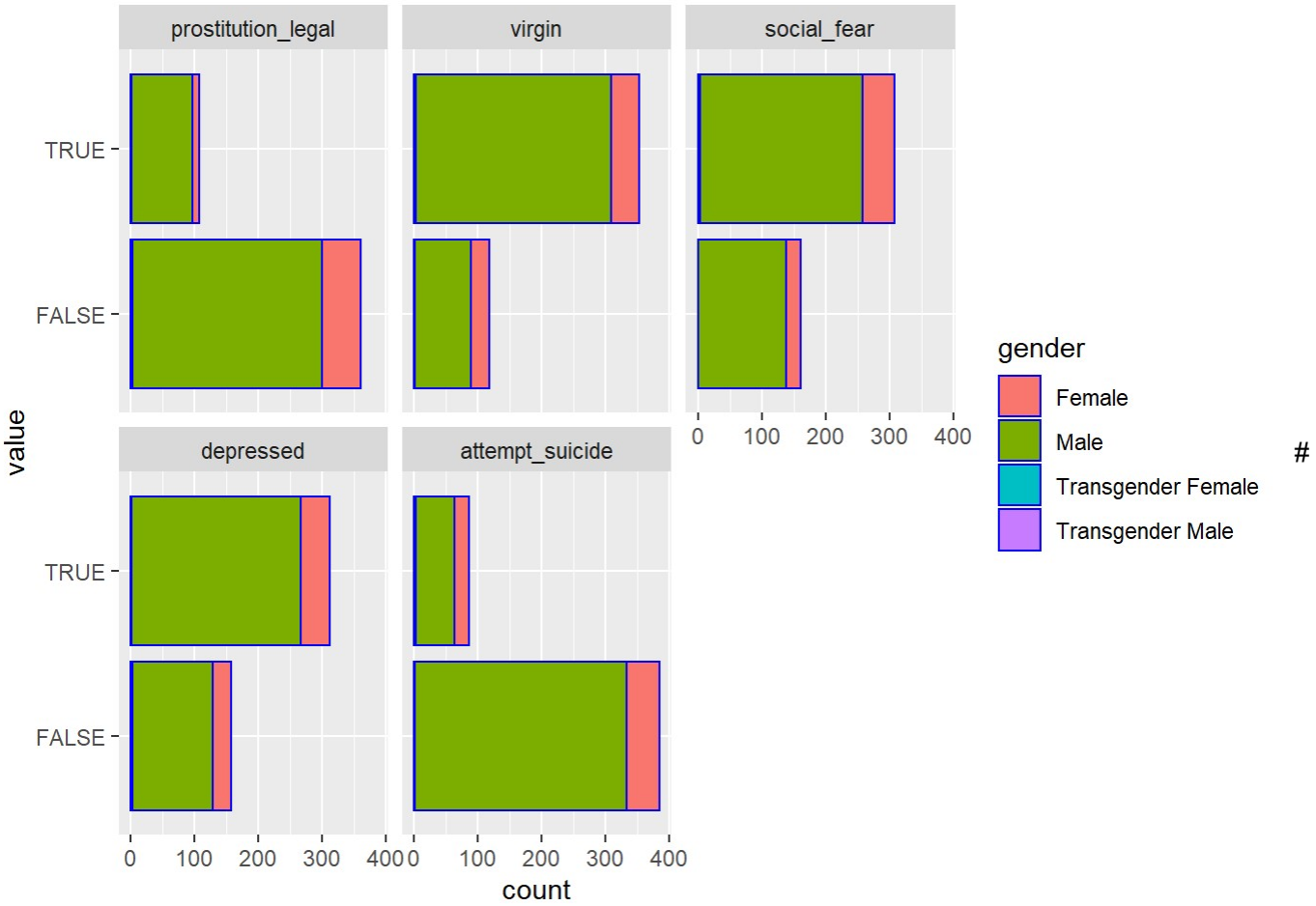
————————————————————————————————

```
bool_list <- c( 'subjectId', 'bodyweight', 'prostitution_legal','virgin', 'social_fea
r', 'depressed', 'attempt_suicide')

FA_data[bool_list] %>%  melt(id=(c("subjectId", "bodyweight"))) %>% dplyr::select(var
iable, value, subjectId, bodyweight) %>% ggplot(aes(y = value, fill = bodyweight))  +
geom_bar(color = "blue") + facet_wrap(~variable, nrow = 2)
```



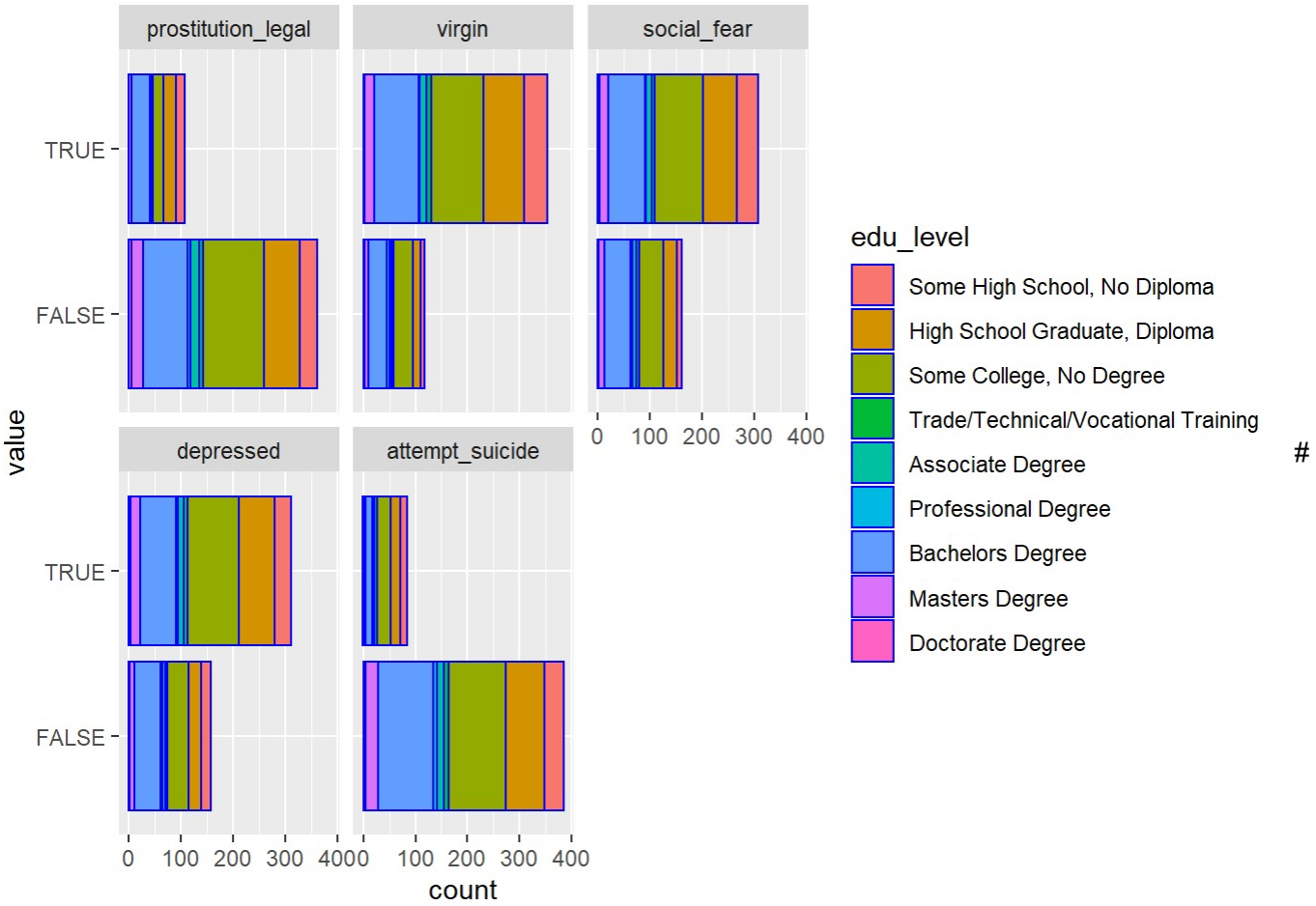
———————————————————————————— # Visualization for 5 logical variables by gender # ————————————————————————————

```
bool_list <- c( 'subjectId', 'gender', 'prostitution_legal','virgin', 'social_fear',
'depressed', 'attempt_suicide')

FA_data[bool_list] %>%  melt(id=(c("subjectId", "gender"))) %>% dplyr::select(variabl
e, value, subjectId, gender) %>% ggplot(aes(y = value, fill = gender))  + geom_bar(co
lor = "blue") + facet_wrap(~variable, nrow = 2)
```



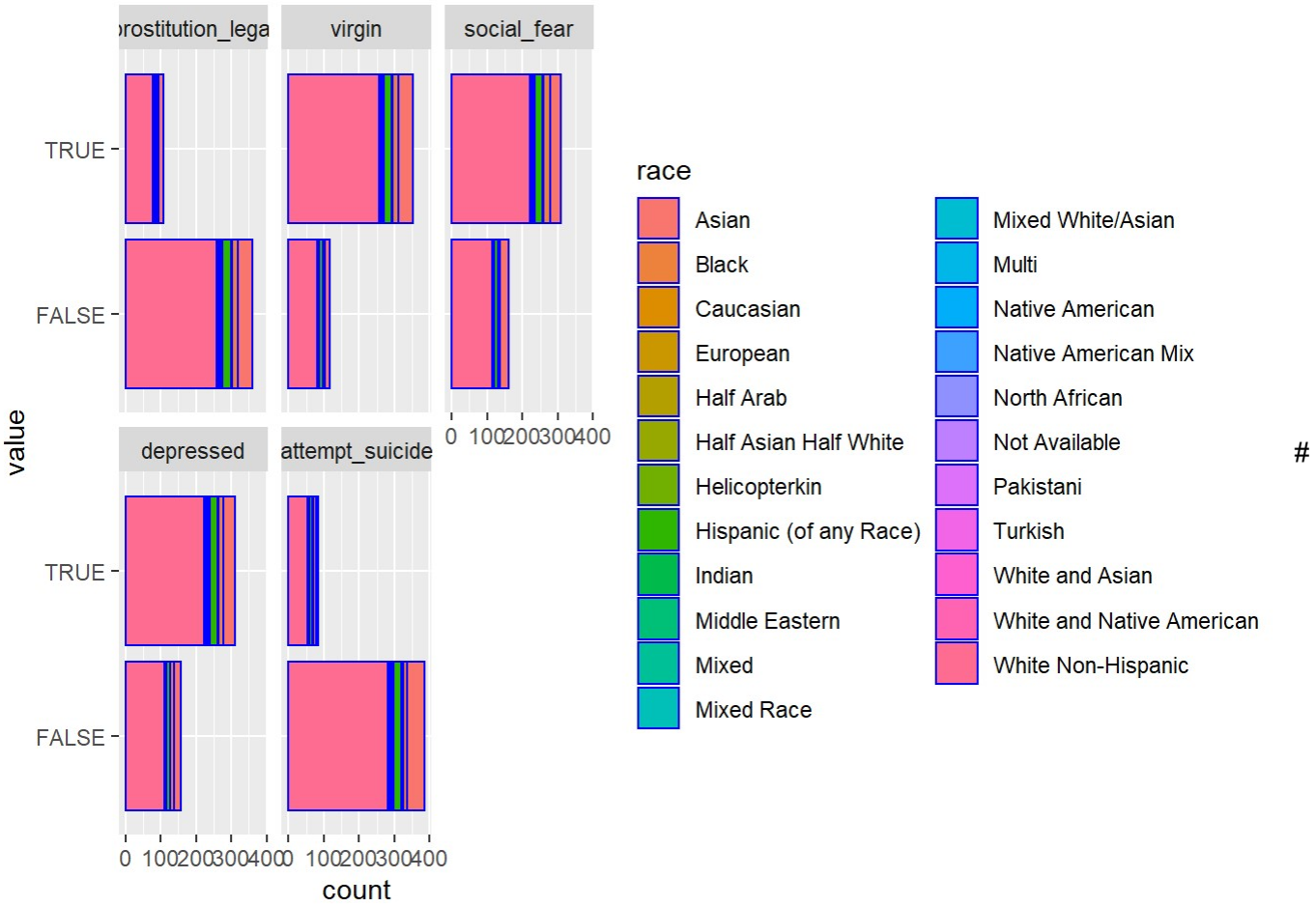
——————————————————————————— # Visualization for 5 logical variables by edu_level #
———————————————————————————

```
bool_list <- c( 'subjectId', 'edu_level', 'prostitution_legal','virgin', 'social_fear
', 'depressed', 'attempt_suicide')

FA_data[bool_list] %>%  melt(id=(c("subjectId", "edu_level"))) %>% dplyr::select(vari
able, value, subjectId, edu_level) %>% ggplot(aes(y = value, fill = edu_level))  + ge
om_bar(color = "blue") + facet_wrap(~variable, nrow = 2)
```



———————————————————————————— # Visualization for 5 logical variables by race # ————————————————————————————

```
bool_list <- c( 'subjectId', 'race', 'prostitution_legal','virgin', 'social_fear', 'd
epressed', 'attempt_suicide')

FA_data[bool_list] %>%  melt(id=(c("subjectId", "race"))) %>% dplyr::select(variable,
value, subjectId, race) %>% ggplot(aes(y = value, fill = race))  + geom_bar(color = "
blue") + facet_wrap(~variable, nrow = 2)
```



———————————————————————————— # improve_yourself_how : EDA and Factorization # ————————————————————————————

```
f_classifier_EDA(FA_data$improve_yourself_how)
FA_data$improve_yourself_how <- factor(FA_data$improve_yourself_how)

#ggplot(data = FA_data, aes(y = edu_level))  + geom_bar(color = "blue", fill = "light blue")
```

Insights and Conclusions

• It seems, there is a misconception that overweight people tend to be lonelier than one with normal body weight.

• People between the age of 18 and 30 are the most alone of all age groups.

• Males are obviously most alone, of all gender groups.

• People who have higher education, seems not to have

aloneness issue.

• People who have no college degree, seems to end up being alone.