zhaozixi wangyuean

2025-12-22

1.

2.

2048 79 “ ”

```
library(tidyverse)
library(ggplot2)
library(cluster)
library(lubridate)
library(naniar)
library(corrplot)
library(jiebaRD)
library(wordcloud)
library(scatterplot3d)
library(MASS)
library(scales)
library(dplyr)
library(stringr)
library(randomForest)
library(broom)
library(car)
library(moments)
library(nnet)
library(caret)
library(reshape2)
library(lmPerm)

data <- read_csv("     /data.csv")

# ==== 1    ====

###
names(data)
```

```
##  [1] "name"              "author"            "create_time"
##  [4] "introduction"      "play_count"        "collect_count"
##  [7] "share_count"       "comment_count"     "topics"
## [10] "fans"              "grade"             "playlists"
```

```
## [13] "identity"           "length_name"        "english_name"
## [16] "name_language"      "name_style"         "name_scene"
## [19] "name_instruments"   "name_feeling"       "name_praise"
## [22] "name_location"      "name_ "          "name_BGM"
## [25] "name_ "          "name_ "          "name_ "
## [28] "name_ "          "name_ "          "name_ "
## [31] "name_ "          "name_ "          "name_ "
## [34] "name_ "          "name_amp"           "name_ "
## [37] "name_ "          "length_intro"       "intro_ "
## [40] "intro_ "         "intro_ "         "intro_ "
## [43] "intro_ "         "intro_ "         "intro_ "
## [46] "intro_ "         "intro_quot"         "intro_amp"
## [49] "intro_ "         "intro_ "         "intro_ "
## [52] "intro_ "         "intro_ "         "intro_ "
## [55] " "               " "               " "
## [58] " "               "ACG"                " "
## [61] " "               "  "             " "
## [64] " "               " "               " "
## [67] " "               " "               " "
## [70] "  "           " "               " "
## [73] " "               " "               "number_songs"
## [76] "number_hot_singers" "talent"             "verification"
## [79] "musician"
```

```
str(data)
```

```
## spc_tbl_ [2,049 x 79] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ name             : chr [1:2049] "- Dance with me." "- Russian" "- - - -" " GOD'S DEATH 888 T% RAP "
##  $ author           : chr [1:2049] "MONSTER-CAT6" "LavidaLoca_z" "    " "YOUNG-RICH-WORLD-PEACE" ...
##  $ create_time      : num [1:2049] 1.39e+09 1.40e+09 1.46e+09 1.42e+09 1.48e+09 ...
##  $ introduction     : chr [1:2049] "<b>  </b>    <br>" "<b>  </b>    .<br>" "<b>  </b>     <br>  <br>
##  $ play_count       : num [1:2049] 248047 69769 311920 250759 53208 ...
##  $ collect_count    : num [1:2049] 2134 1464 10256 3815 350 ...
##  $ share_count      : num [1:2049] 39 13 216 65 11 193 20 48 13 25 ...
##  $ comment_count    : num [1:2049] 52 15 201 60 9 202 10 23 7 34 ...
##  $ topics           : chr [1:2049] "[\" \",\" \",\" \"]" "[\"  \",\" \",\" \"]" "[\" \",\" / \",\"
##  $ fans             : num [1:2049] 5974 17163 8237 277 1595 ...
##  $ grade            : num [1:2049] 9 8 9 9 8 9 9 9 9 7 ...
##  $ playlists        : num [1:2049] 22 128 51 8 25 22 118 27 34 15 ...
##  $ identity         : chr [1:2049] " " " " " " " " " " ...
##  $ length_name      : num [1:2049] 16 9 7 24 7 11 29 9 18 14 ...
##  $ english_name     : chr [1:2049] " " " " " " " " " " ...
##  $ name_language    : chr [1:2049] " " " " " " " " " " ...
##  $ name_style       : chr [1:2049] " " " " " " " " " " ...
##  $ name_scene       : chr [1:2049] " " " " " " " " " " ...
##  $ name_instruments : chr [1:2049] " " " " " " " " " " ...
##  $ name_feeling     : chr [1:2049] " " " " " " " " " " ...
##  $ name_praise      : chr [1:2049] " " " " " " " " " " ...
##  $ name_location    : chr [1:2049] " " " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " " " ...
##  $ name_BGM         : chr [1:2049] " " " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " " " ...
```

```
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_amp          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ name_          : chr [1:2049] " " " " " " " " ...
##  $ length_intro      : num [1:2049] 21 21 435 68 21 454 57 22 109 35 ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_quot        : chr [1:2049] " " " " " " " " ...
##  $ intro_amp         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $ intro_         : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $ ACG               : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $              : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $             : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $                : chr [1:2049] " " " " " " " " ...
##  $ number_songs      : num [1:2049] 496 50 252 773 123 51 24 237 90 33 ...
##  $ number_hot_singers: num [1:2049] 30 2 5 23 6 1 0 5 5 0 ...
##  $ talent            : chr [1:2049] " " " " " " " " ...
##  $ verification      : chr [1:2049] " " " " " " " " ...
##  $ musician          : chr [1:2049] " " " " " " " " ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   name = col_character(),
```

```
##   ..   author = col_character(),
##   ..   create_time = col_double(),
##   ..   introduction = col_character(),
##   ..   play_count = col_double(),
##   ..   collect_count = col_double(),
##   ..   share_count = col_double(),
##   ..   comment_count = col_double(),
##   ..   topics = col_character(),
##   ..   fans = col_double(),
##   ..   grade = col_double(),
##   ..   playlists = col_double(),
##   ..   identity = col_character(),
##   ..   length_name = col_double(),
##   ..   english_name = col_character(),
##   ..   name_language = col_character(),
##   ..   name_style = col_character(),
##   ..   name_scene = col_character(),
##   ..   name_instruments = col_character(),
##   ..   name_feeling = col_character(),
##   ..   name_praise = col_character(),
##   ..   name_location = col_character(),
##   ..   name_  = col_character(),
##   ..   name_BGM = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   name_amp = col_character(),
##   ..   name_  = col_character(),
##   ..   name_  = col_character(),
##   ..   length_intro = col_double(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_quot = col_character(),
##   ..   intro_amp = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..   intro_  = col_character(),
##   ..     = col_character(),
```

```
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..   ACG = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..      = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..      = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..     = col_character(),
##   ..   number_songs = col_double(),
##   ..   number_hot_singers = col_double(),
##   ..   talent = col_character(),
##   ..   verification = col_character(),
##   ..   musician = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
cat("   ", dim(data), "\n")
```

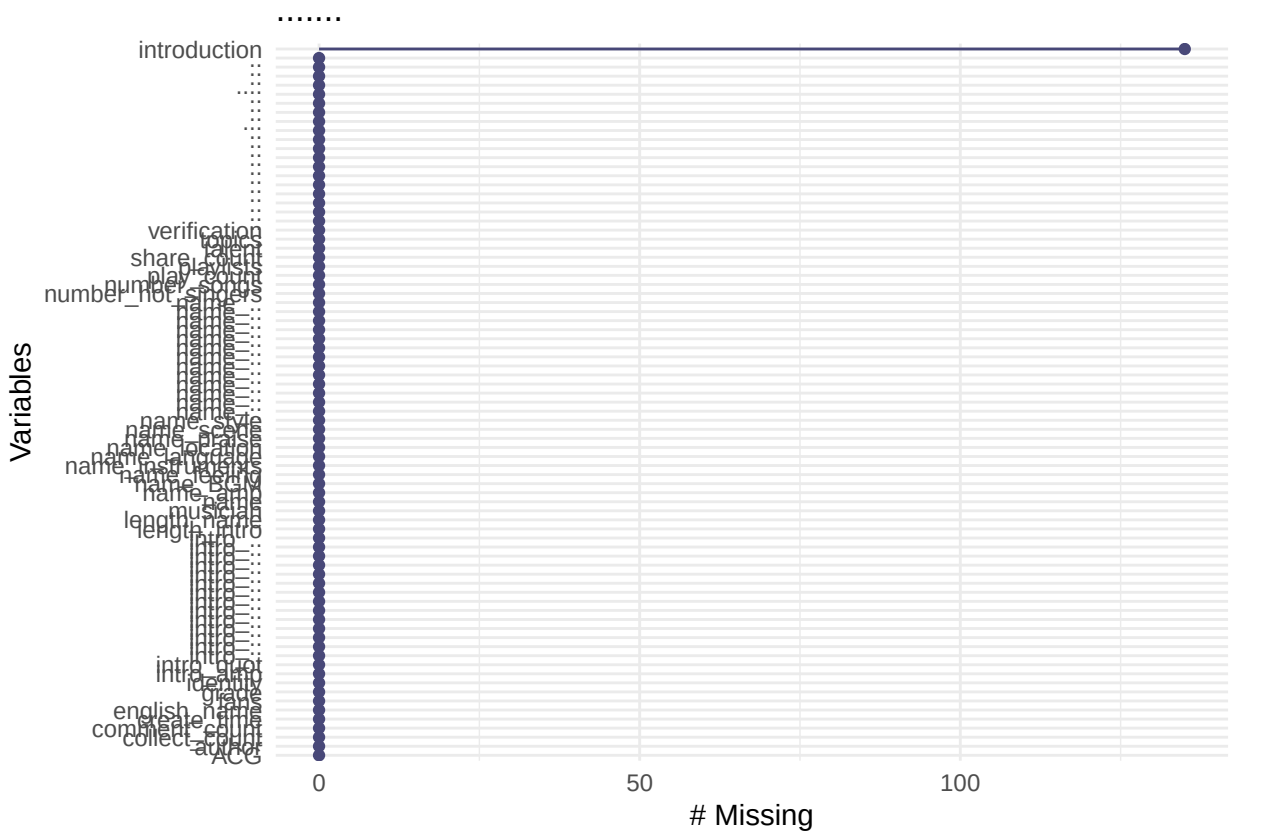
```
##     2049 79
```

```
colSums(is.na(data))
```

```
##              name            author       create_time      introduction
##                 0                 0                 0               135
##        play_count   collect_count       share_count     comment_count
##                 0                 0                 0                 0
##            topics              fans             grade         playlists
##                 0                 0                 0                 0
##          identity     length_name      english_name     name_language
##                 0                 0                 0                 0
##        name_style       name_scene  name_instruments      name_feeling
##                 0                 0                 0                 0
##       name_praise     name_location           name_            name_BGM
##                 0                 0                 0                 0
##         name_           name_           name_           name_
##                 0                 0                 0                 0
##         name_           name_           name_           name_
##                 0                 0                 0                 0
##         name_           name_         name_amp          name_
##                 0                 0                 0                 0
##         name_       length_intro        intro_          intro_
##                 0                 0                 0                 0
##        intro_          intro_          intro_          intro_
##                 0                 0                 0                 0
##        intro_          intro_       intro_quot        intro_amp
```

```
##                 0                 0                 0                 0
##         intro_            intro_            intro_             intro_
##                 0                 0                 0                 0
##         intro_            intro_
##                 0                 0                 0                 0
##                                                 ACG
##                 0                 0                 0                 0
##
##                 0                 0                 0                 0
##
##                 0                 0                 0                 0
##
##                 0                 0                 0                 0
##                                       number_songs number_hot_singers
##                 0                 0                 0                 0
##             talent      verification           musician
##                 0                 0                 0
```

```
gg_miss_var(data) + labs(title = "    ")
```




```
###     introduction     "[ ]"  null
data$introduction[is.na(data$introduction) | data$introduction == ""] <- "[ ]"

###        factor
convert_binary_proper <- function(data) {
  for(col_name in names(data)) {
    if(is.character(data[[col_name]]) || is.factor(data[[col_name]])) {
      unique_vals <- unique(data[[col_name]])
```

```
      #  " "/" "
      if(length(unique_vals) == 2 && all(sort(unique_vals) == c(" ", " "))) {
        #         0/1
        data[[paste0(col_name, "_binary")]] <- ifelse(
          data[[col_name]] == " ", 1, 0
        )
        cat("  ", col_name, "     =1 =0 \n")
      }
    }
  }
  return(data)
}
data <- convert_binary_proper(data) #
```

```
##    english_name      =1 =0
##    name_language      =1 =0
##    name_style      =1 =0
##    name_scene      =1 =0
##    name_instruments      =1 =0
##    name_feeling      =1 =0
##    name_praise      =1 =0
##    name_location      =1 =0
##    name_       =1 =0
##    name_BGM      =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    name_amp      =1 =0
##    name_       =1 =0
##    name_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_quot      =1 =0
##    intro_amp      =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
##    intro_       =1 =0
```

```
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##    ACG      =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##            =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##           =1 =0
##    talent      =1 =0
##    verification      =1 =0
##    musician      =1 =0
```

```
###
remove_columns_tidy <- function(data, col_names) {
  existing_cols <- col_names[col_names %in% names(data)]
  missing_cols <- col_names[!col_names %in% names(data)]

  if(length(missing_cols) > 0) {
    warning("      : ", paste(missing_cols, collapse = ", "))
  }

  if(length(existing_cols) > 0) {
    data <- data %>% dplyr::select(-all_of(existing_cols))
    cat("  :", paste(existing_cols, collapse = ", "), "\n")
  }
  return(data)
}

data_cleaned = data
```

3.

* 8.969987 log1p * 80 90

```
# ==== 2      ====

###  numeric factor
numeric_vars <- names(data_cleaned)[sapply(data_cleaned, is.numeric)]
factor_vars <- names(data_cleaned)[sapply(data_cleaned, is.factor)]
desc_stats <- (data_cleaned[, numeric_vars])
summary(desc_stats)
```

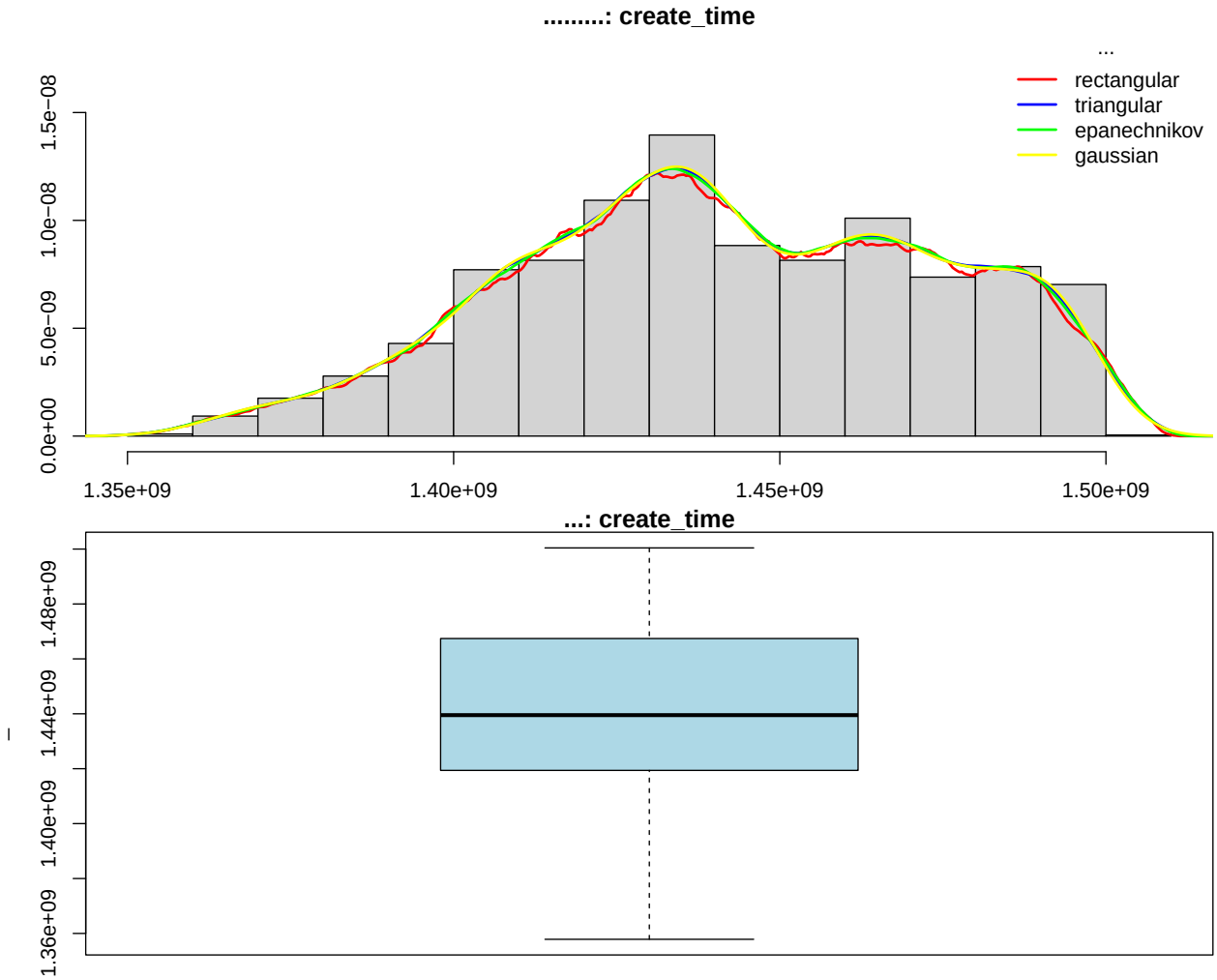
```
##    create_time          play_count       collect_count     share_count
```

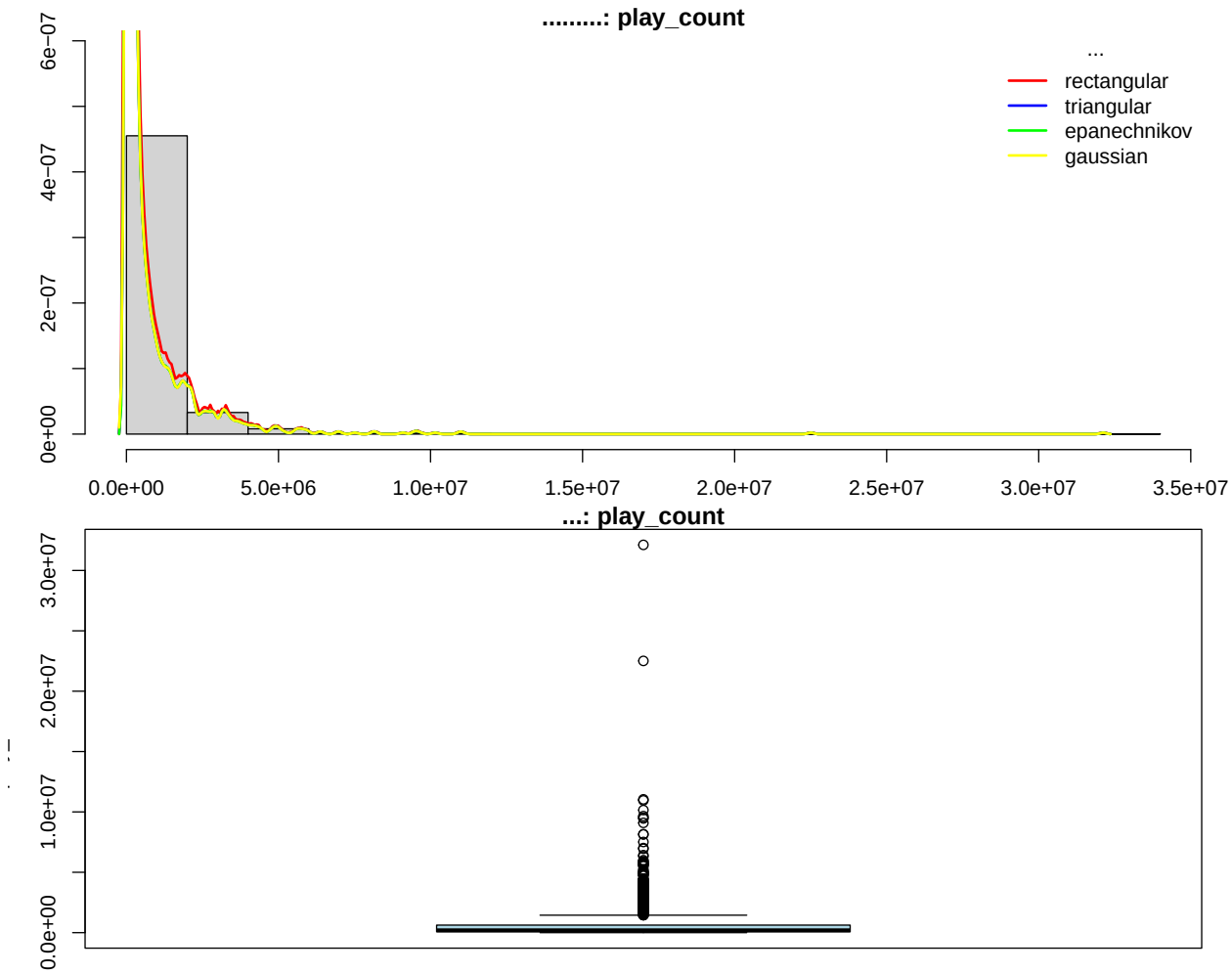
```
##  Min.   :1.358e+09   Min.   :      49   Min.   :     5   Min.   :    1.0
##  1st Qu.:1.419e+09   1st Qu.:   74111   1st Qu.:  1999   1st Qu.:   24.0
##  Median :1.440e+09   Median :  193324   Median :  5965   Median :   63.0
##  Mean   :1.442e+09   Mean   :  658842   Mean   : 15287   Mean   :  177.5
##  3rd Qu.:1.467e+09   3rd Qu.:  625232   3rd Qu.: 17617   3rd Qu.:  174.0
##  Max.   :1.500e+09   Max.   :32119004   Max.   :738775   Max.   :13105.0
##  comment_count         fans            grade          playlists
##  Min.   :   0.0   Min.   :     1   Min.   : 0.000   Min.   :   2.0
##  1st Qu.:  29.0   1st Qu.:   407   1st Qu.: 8.000   1st Qu.:  22.0
##  Median :  77.0   Median :  4765   Median : 9.000   Median :  45.0
##  Mean   : 162.3   Mean   : 11751   Mean   : 8.446   Mean   : 101.9
##  3rd Qu.: 187.0   3rd Qu.: 14076   3rd Qu.: 9.000   3rd Qu.:  98.0
##  Max.   :7718.0   Max.   :264183   Max.   :10.000   Max.   :1000.0
##   length_name     length_intro     number_songs    number_hot_singers
##  Min.   : 3.00   Min.   :   0.0   Min.   :   5.0   Min.   :  0.00
##  1st Qu.:11.00   1st Qu.:  36.0   1st Qu.:  31.0   1st Qu.:  0.00
##  Median :14.00   Median :  82.0   Median :  57.0   Median :  1.00
##  Mean   :14.65   Mean   : 167.5   Mean   : 112.7   Mean   : 10.75
##  3rd Qu.:18.00   3rd Qu.: 162.0   3rd Qu.: 122.0   3rd Qu.:  7.00
##  Max.   :37.00   Max.   :1354.0   Max.   :1000.0   Max.   :412.00
##  english_name_binary name_language_binary name_style_binary name_scene_binary
##  Min.   :0.0000      Min.   :0.00000      Min.   :0.0000    Min.   :0.00000
##  1st Qu.:0.0000      1st Qu.:0.00000      1st Qu.:0.0000    1st Qu.:0.00000
##  Median :0.0000      Median :0.00000      Median :0.0000    Median :0.00000
##  Mean   :0.3167      Mean   :0.04929      Mean   :0.2831    Mean   :0.02684
##  3rd Qu.:1.0000      3rd Qu.:0.00000      3rd Qu.:1.0000    3rd Qu.:0.00000
##  Max.   :1.0000      Max.   :1.00000      Max.   :1.0000    Max.   :1.00000
##  name_instruments_binary name_feeling_binary name_praise_binary
##  Min.   :0.00000         Min.   :0.0000      Min.   :0.00000
##  1st Qu.:0.00000         1st Qu.:0.0000      1st Qu.:0.00000
##  Median :0.00000         Median :0.0000      Median :0.00000
##  Mean   :0.06442         Mean   :0.1454      Mean   :0.08882
##  3rd Qu.:0.00000         3rd Qu.:0.0000      3rd Qu.:0.00000
##  Max.   :1.00000         Max.   :1.0000      Max.   :1.00000
##  name_location_binary name_ _binary  name_BGM_binary   name_ _binary
##  Min.   :0.00000      Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000      1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000      Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.09956      Mean   :0.04539   Mean   :0.02147   Mean   :0.02879
##  3rd Qu.:0.00000      3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000      Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  name_ _binary  name_ _binary  name_ _binary  name_ _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.02587   Mean   :0.02099   Mean   :0.03953   Mean   :0.02294
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  name_ _binary  name_ _binary  name_ _binary  name_ _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.02977   Mean   :0.02635   Mean   :0.02245   Mean   :0.02001
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
```

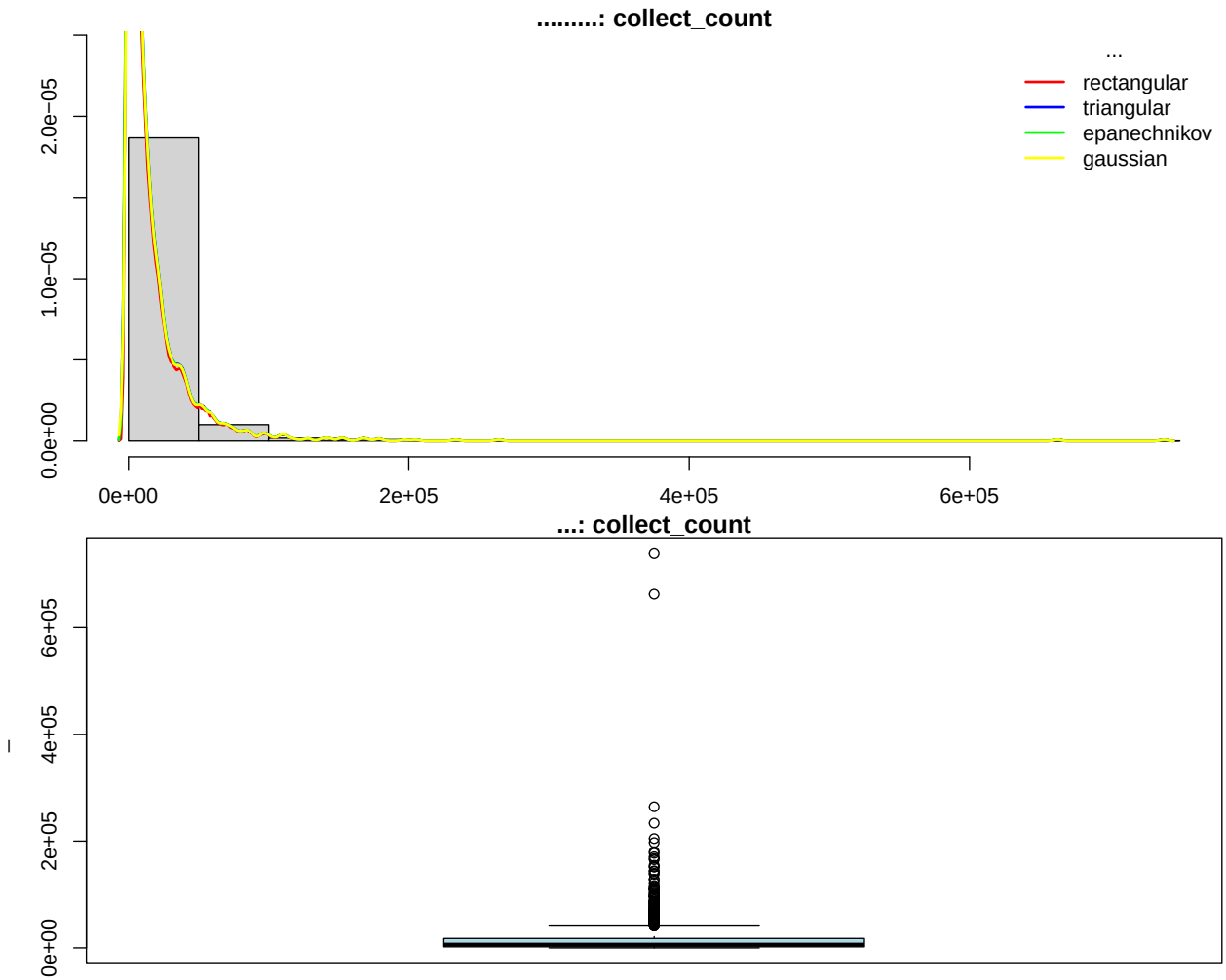
```
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  name_ _binary  name_amp_binary   name_ _binary  name_ _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.01708   Mean   :0.02879   Mean   :0.01708   Mean   :0.02587
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  intro_ _binary intro_ _binary intro_ _binary intro_ _binary
##  Min.   :0.000     Min.   :0.0000    Min.   :0.0000    Min.   :0.0000
##  1st Qu.:0.000     1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.0000
##  Median :0.000     Median :0.0000    Median :0.0000    Median :0.0000
##  Mean   :0.163     Mean   :0.2328    Mean   :0.1327    Mean   :0.1088
##  3rd Qu.:0.000     3rd Qu.:0.0000    3rd Qu.:0.0000    3rd Qu.:0.0000
##  Max.   :1.000     Max.   :1.0000    Max.   :1.0000    Max.   :1.0000
##  intro_ _binary intro_ _binary intro_ _binary intro_ _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.08443   Mean   :0.03856   Mean   :0.04392   Mean   :0.07418
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  intro_quot_binary intro_amp_binary  intro_ _binary intro_ _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.01269   Mean   :0.03367   Mean   :0.08053   Mean   :0.05368
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  intro_ _binary intro_ _binary intro_ _binary intro_ _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.04685   Mean   :0.05905   Mean   :0.04636   Mean   :0.02684
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##    _binary        _binary        _binary       _binary
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.000   Median :0.0000
##  Mean   :0.2621   Mean   :0.1957   Mean   :0.162   Mean   :0.1484
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000
##    ACG_binary       _binary        _binary       _binary
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.1215   Mean   :0.1157   Mean   :0.0898   Mean   :0.0815
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    _binary         _binary         _binary         _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
```

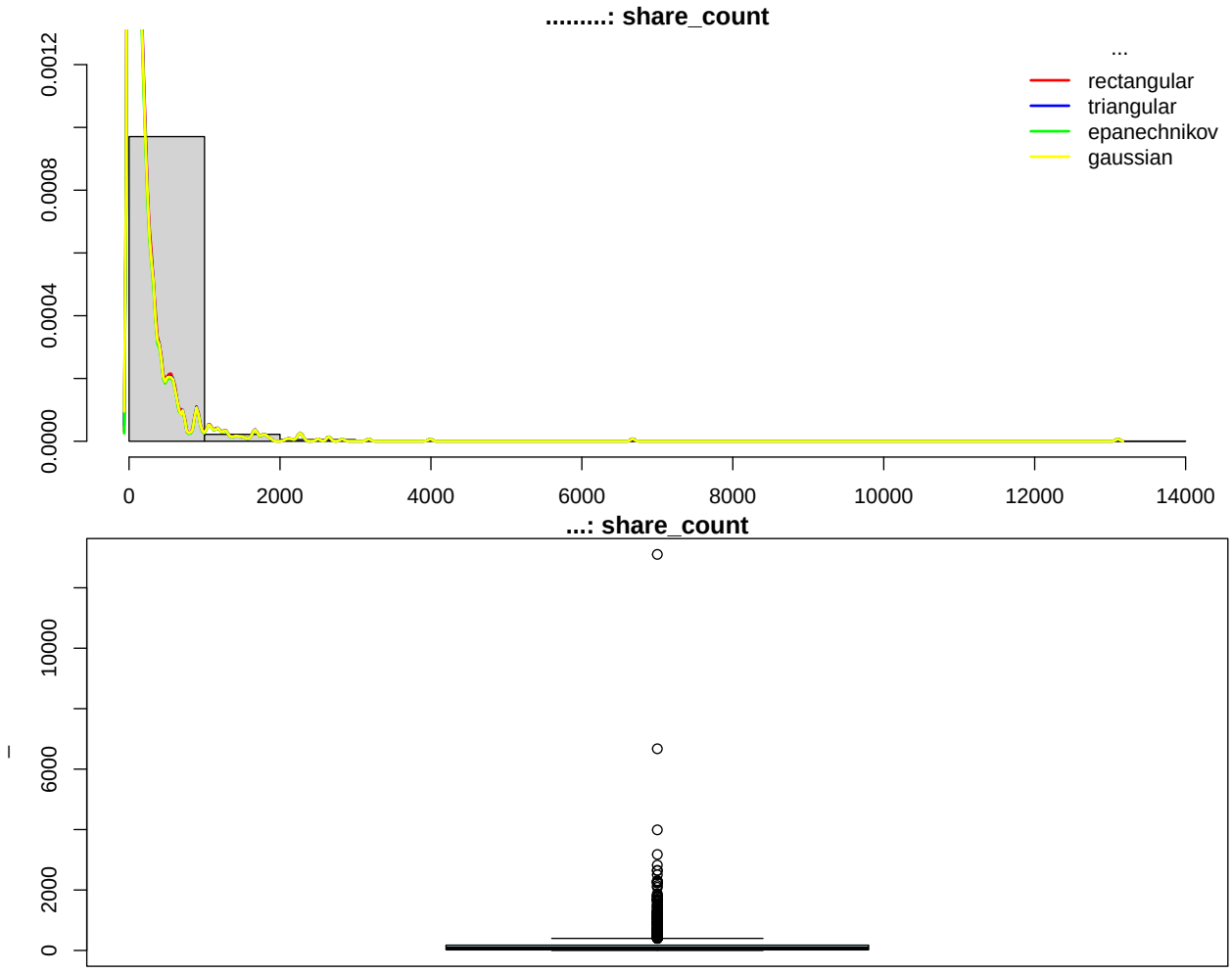
```
##  Mean   :0.07077   Mean   :0.06003   Mean   :0.05808   Mean   :0.05661
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##    _binary         _binary         _binary         _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.05661   Mean   :0.05271   Mean   :0.05271   Mean   :0.05222
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##    _binary         _binary         _binary         _binary
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.05173   Mean   :0.05124   Mean   :0.04832   Mean   :0.04783
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  talent_binary    verification_binary musician_binary
##  Min.   :0.0000   Min.   :0.0000      Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000      1st Qu.:0.00000
##  Median :1.0000   Median :0.0000      Median :0.00000
##  Mean   :0.5432   Mean   :0.0327      Mean   :0.03172
##  3rd Qu.:1.0000   3rd Qu.:0.0000      3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.0000      Max.   :1.00000
```

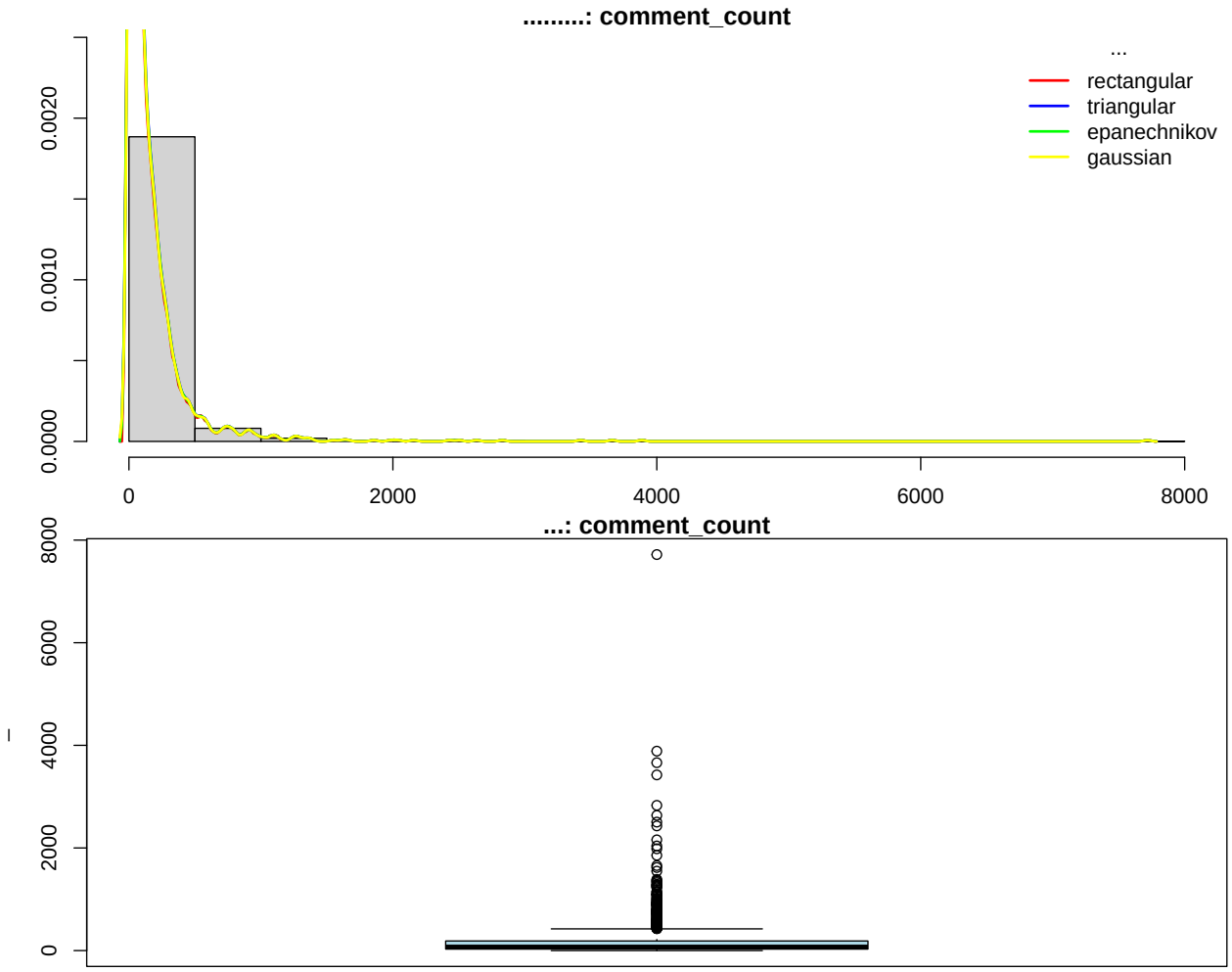
```r
###
par(mfrow = c(2, 1), mar = c(2, 3, 1, 2) + 0.1)

for (var in numeric_vars) {
  current_data = data_cleaned[[var]]
  kernels <- c("rectangular", "triangular", "epanechnikov", "gaussian")
  colors <- c("red", "blue", "green", "yellow")
  hist(current_data, freq = FALSE, main = paste("     :", var), xlab = var, ylab = " ", col = "lightgray

  bandwidths = 1
  for (i in seq_along(kernels)) {
    kernel_type <- kernels[i]
    color <- colors[i]
    kde <- density(current_data, kernel = kernel_type, bandwidths = bandwidths, adjust = 1)
    lines(kde, col = color, lty = 1, lwd = 2)
  }

  legend("topright", legend = kernels, col = colors, lty = 1, lwd = 2, title = "  ", bty = "n")

  boxplot(current_data, main = paste("  :", var), ylab = var, col = "lightblue")
}
```

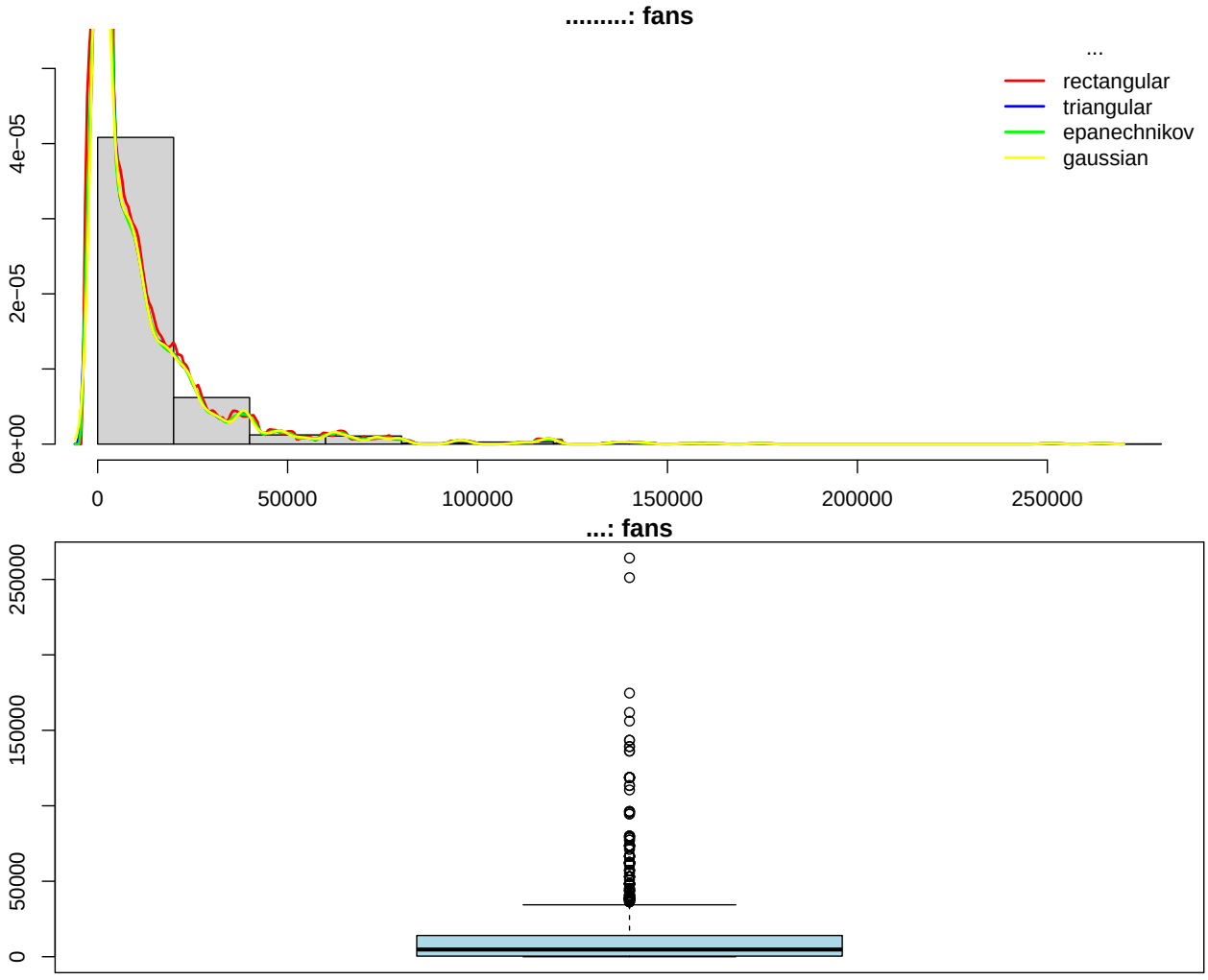

.....: create_time
...
rectangular
triangular
epanechnikov
gaussian
0.0e+00
5.0e−09
1.0e−08
1.5e−08
1.35e+09
1.40e+09
1.45e+09
1.50e+09
...: create_time
1.36e+09
1.40e+09
1.44e+09
1.48e+09

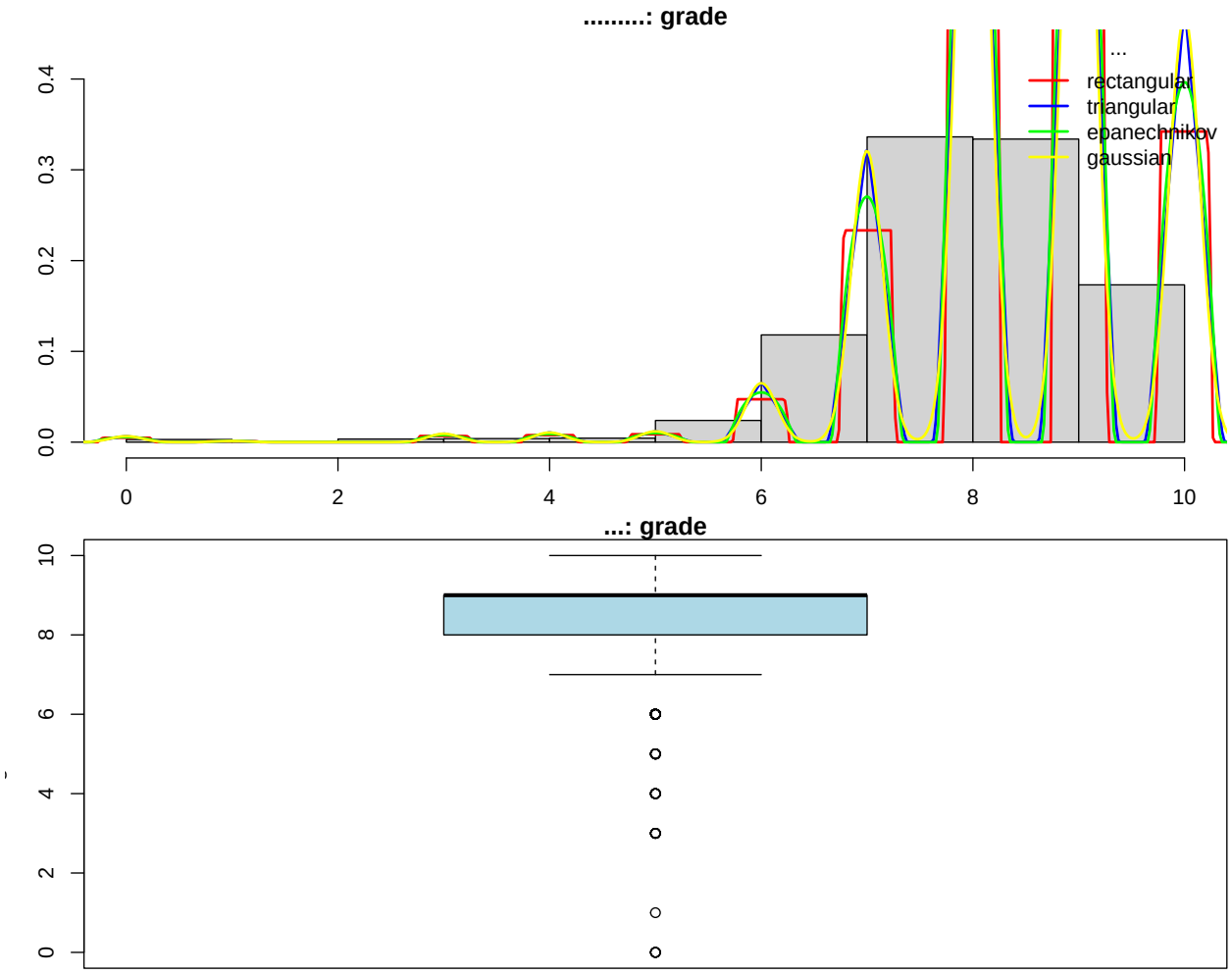

.....: play_count
...
rectangular
triangular
epanechnikov
gaussian
0e+00
2e−07
4e−07
6e−07
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
2.5e+07
3.0e+07
3.5e+07
...: play_count
0.0e+00
1.0e+07
2.0e+07
3.0e+07

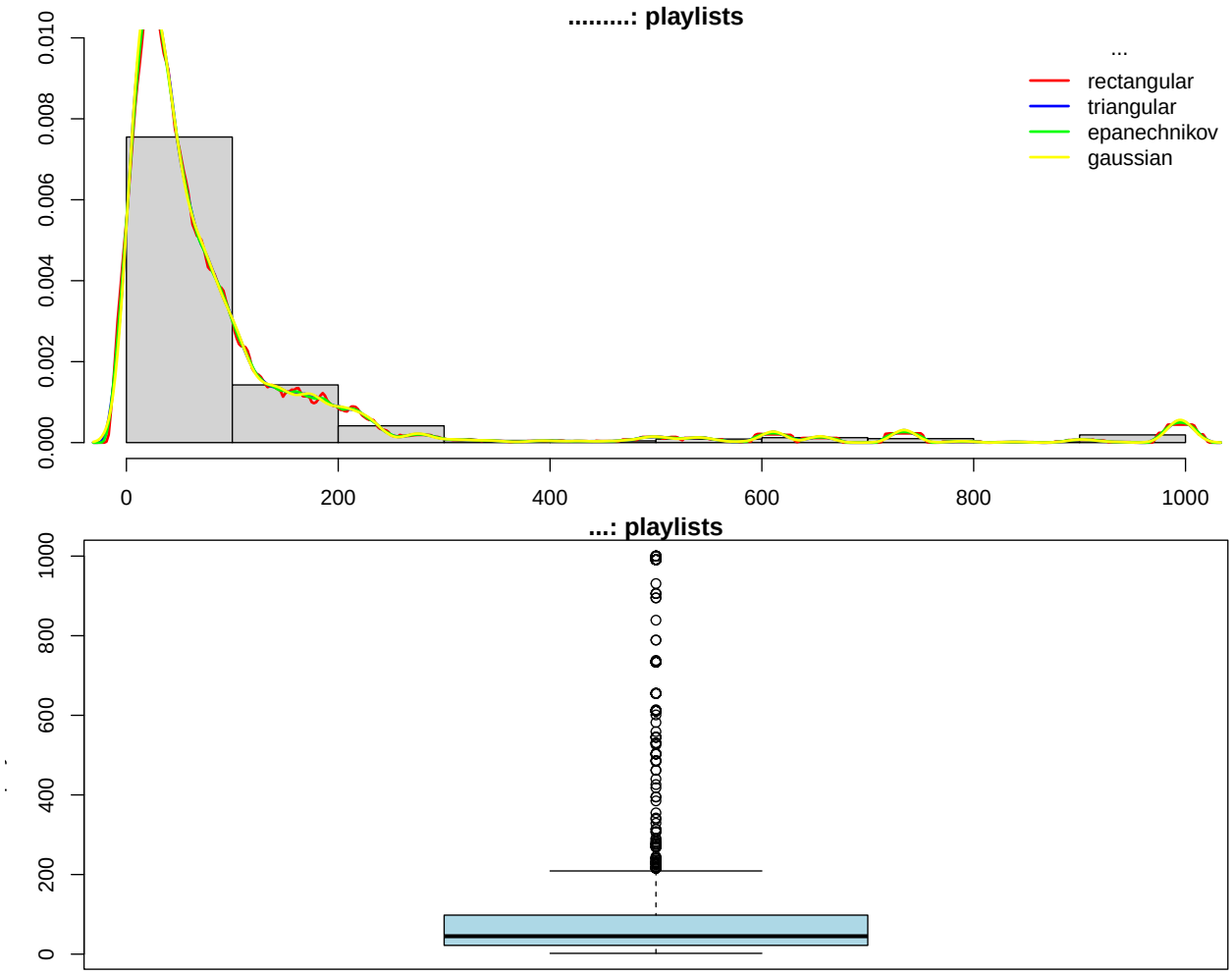

.....: collect_count
...
rectangular
triangular
epanechnikov
gaussian
0.0e+00
1.0e−05
2.0e−05
0e+00
2e+05
4e+05
6e+05
...: collect_count
0e+00
2e+05
4e+05
6e+05

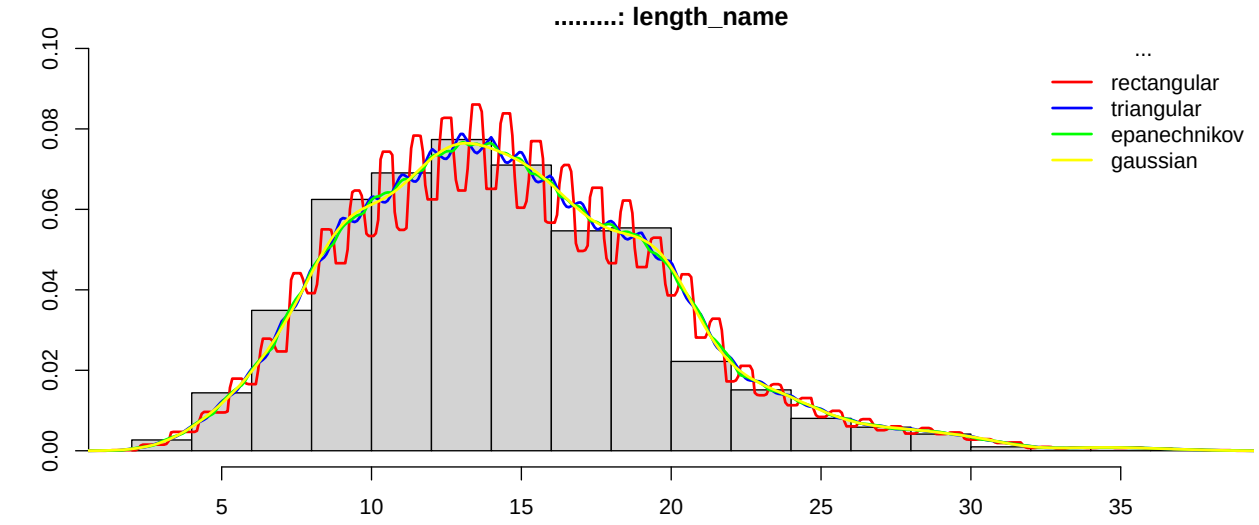

.....: share_count
...
rectangular
triangular
epanechnikov
gaussian
0.0000
0.0004
0.0008
0.0012
0
2000
4000
6000
8000
10000
12000
14000
...: share_count
0
2000
6000
10000

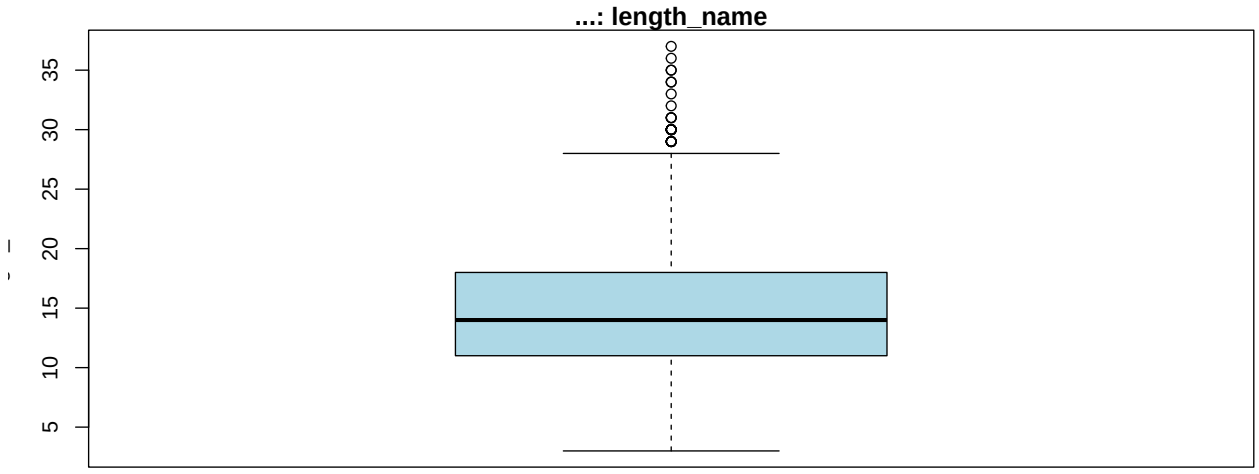

.....: comment_count
...
rectangular
triangular
epanechnikov
gaussian
0.0000
0.0010
0.0020
0
2000
4000
6000
8000
...: comment_count
0
2000
4000
6000
8000

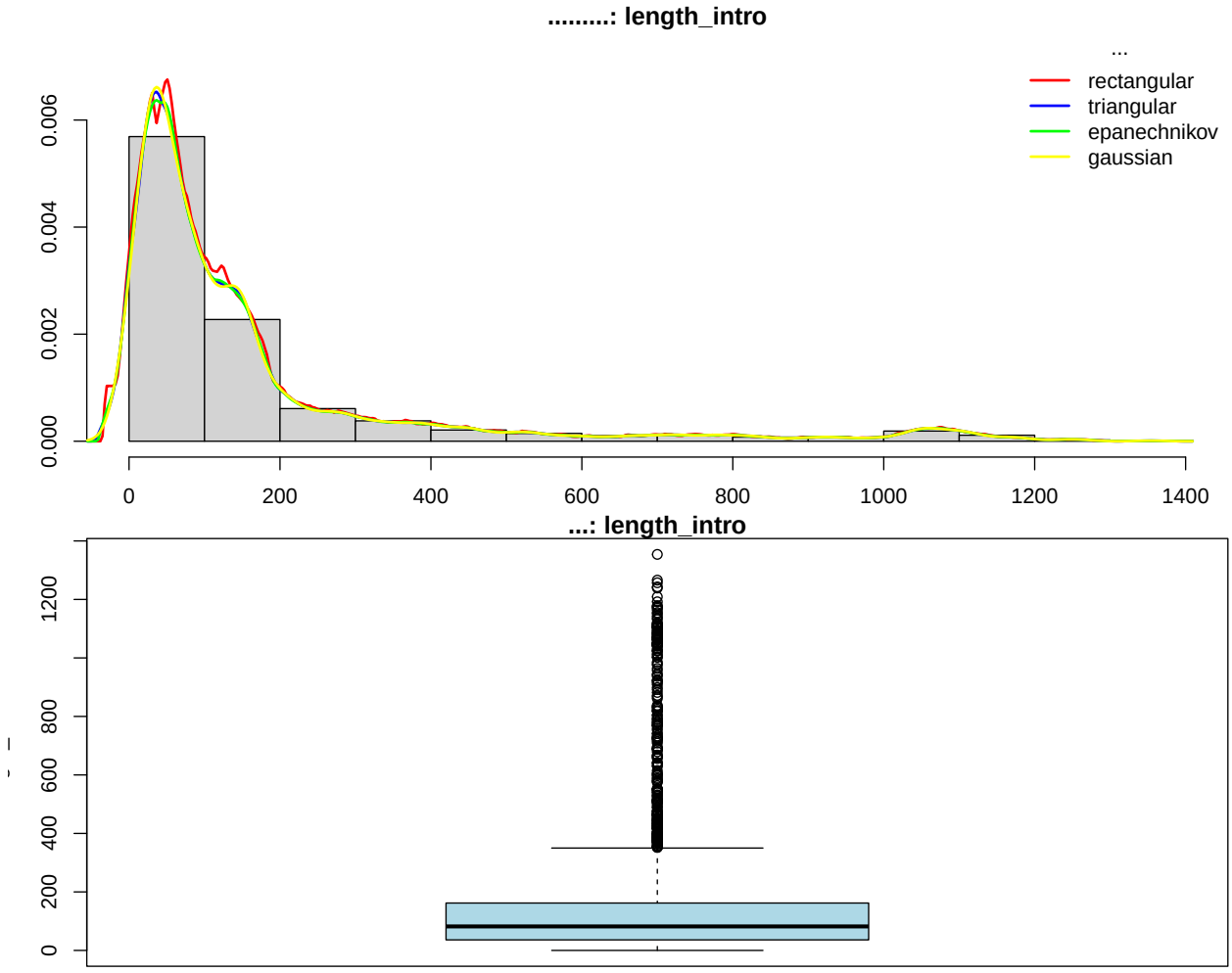

.....: fans
...
rectangular
triangular
epanechnikov
gaussian
0e+00
2e−05
4e−05
0
50000
100000
150000
200000
250000
...: fans
0
50000
150000
250000

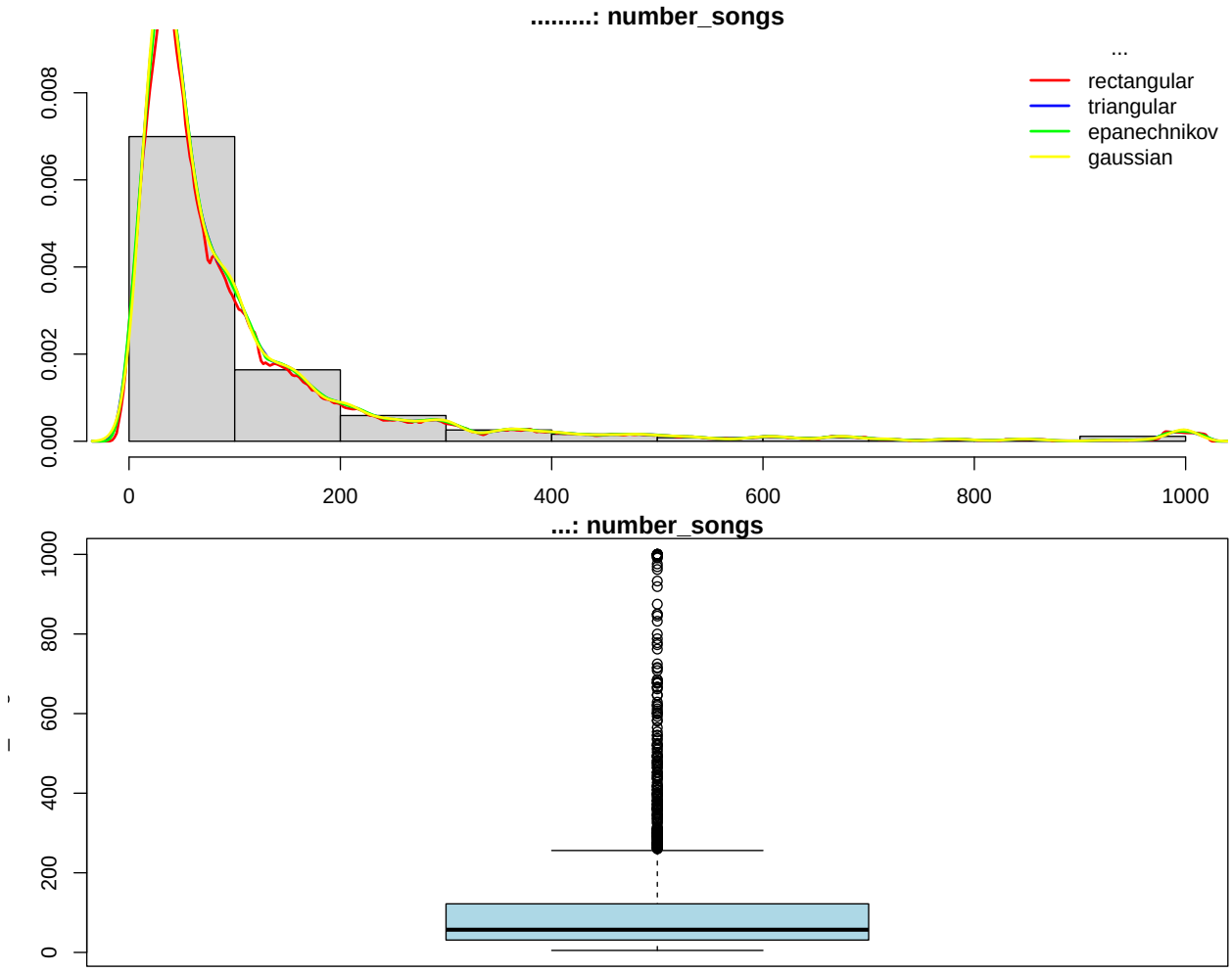

.....: grade
...
rectangular
triangular
epanechnikov
gaussian
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10
...: grade
0
2
4
6
8
10

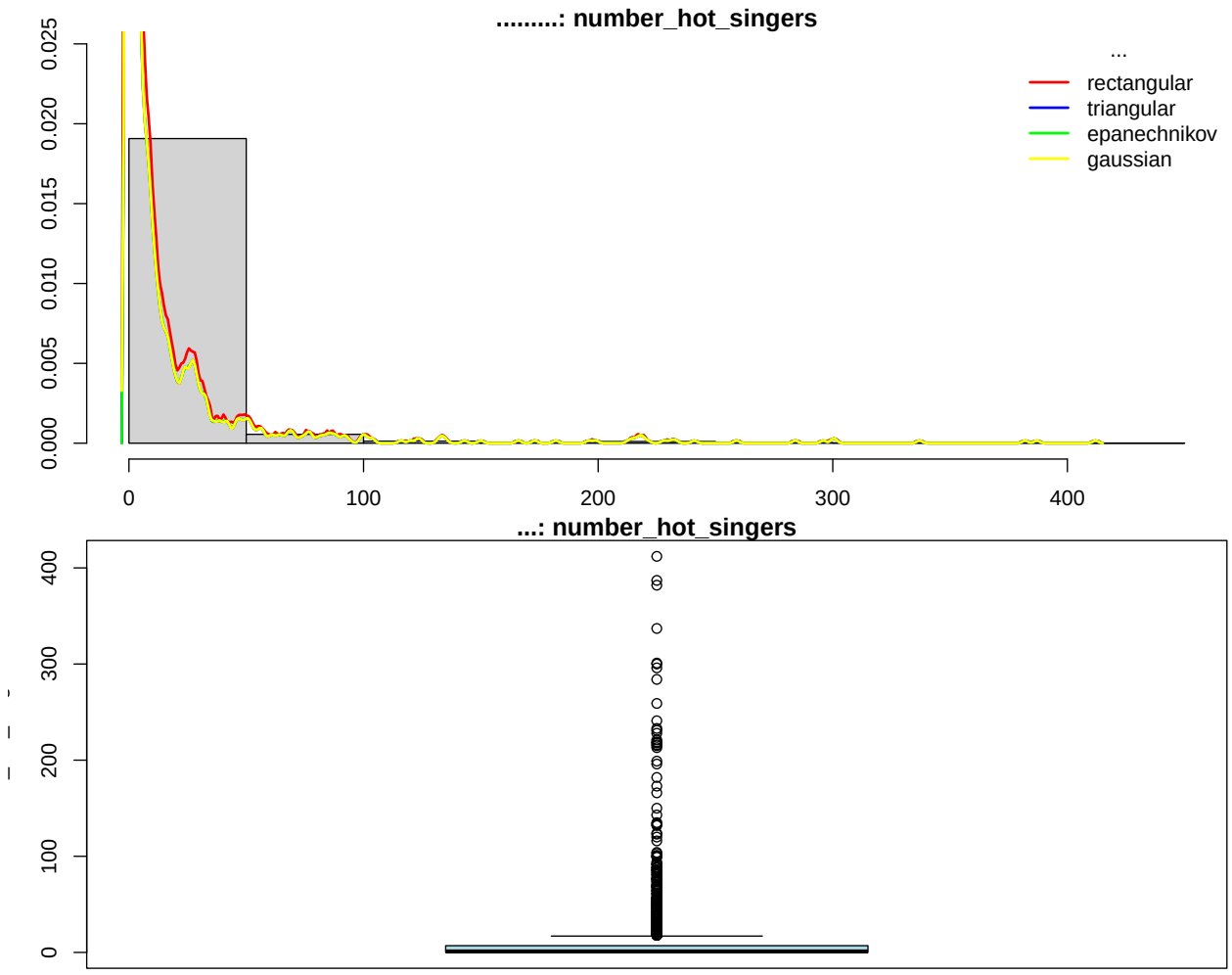

.....: playlists
...
rectangular
triangular
epanechnikov
gaussian
0.000
0.002
0.004
0.006
0.008
0.010
0
200
400
600
800
1000
...: playlists
0
200
400
600
800
1000

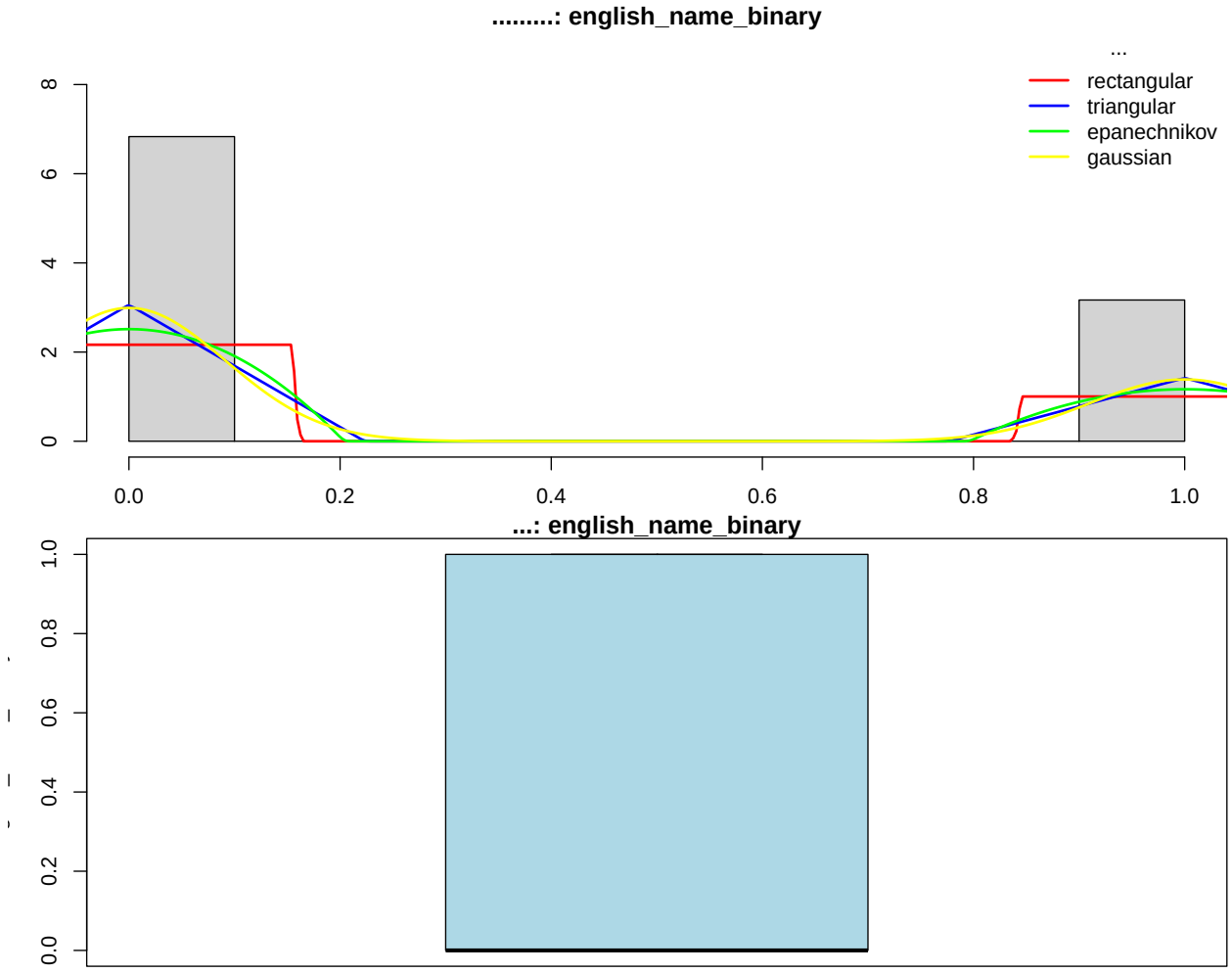

.....: length_name
...
rectangular
triangular
epanechnikov
gaussian
0.00
0.02
0.04
0.06
0.08
0.10
5
10
15
20
25
30
35

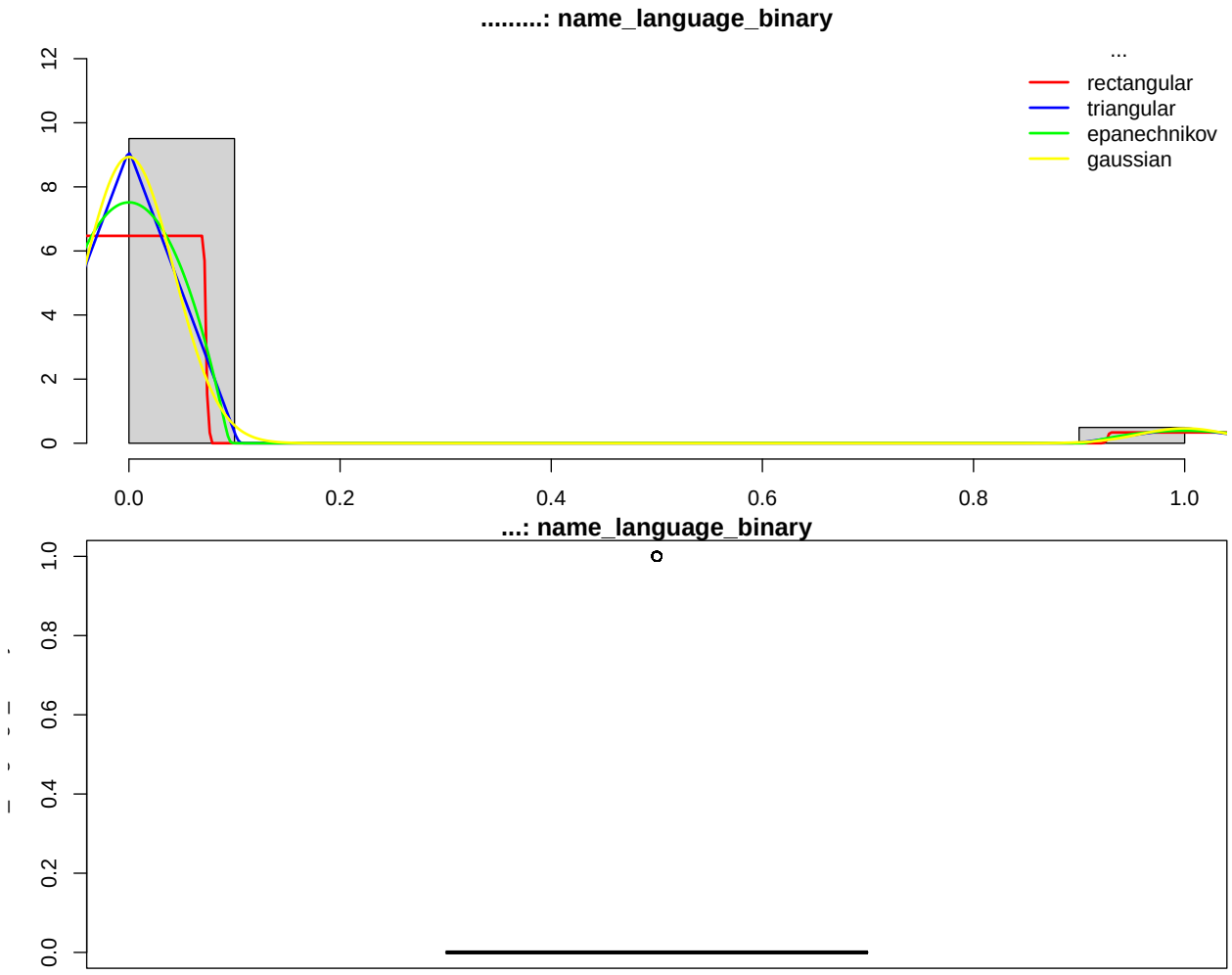

...: length_name
5
10
15
20
25
30
35

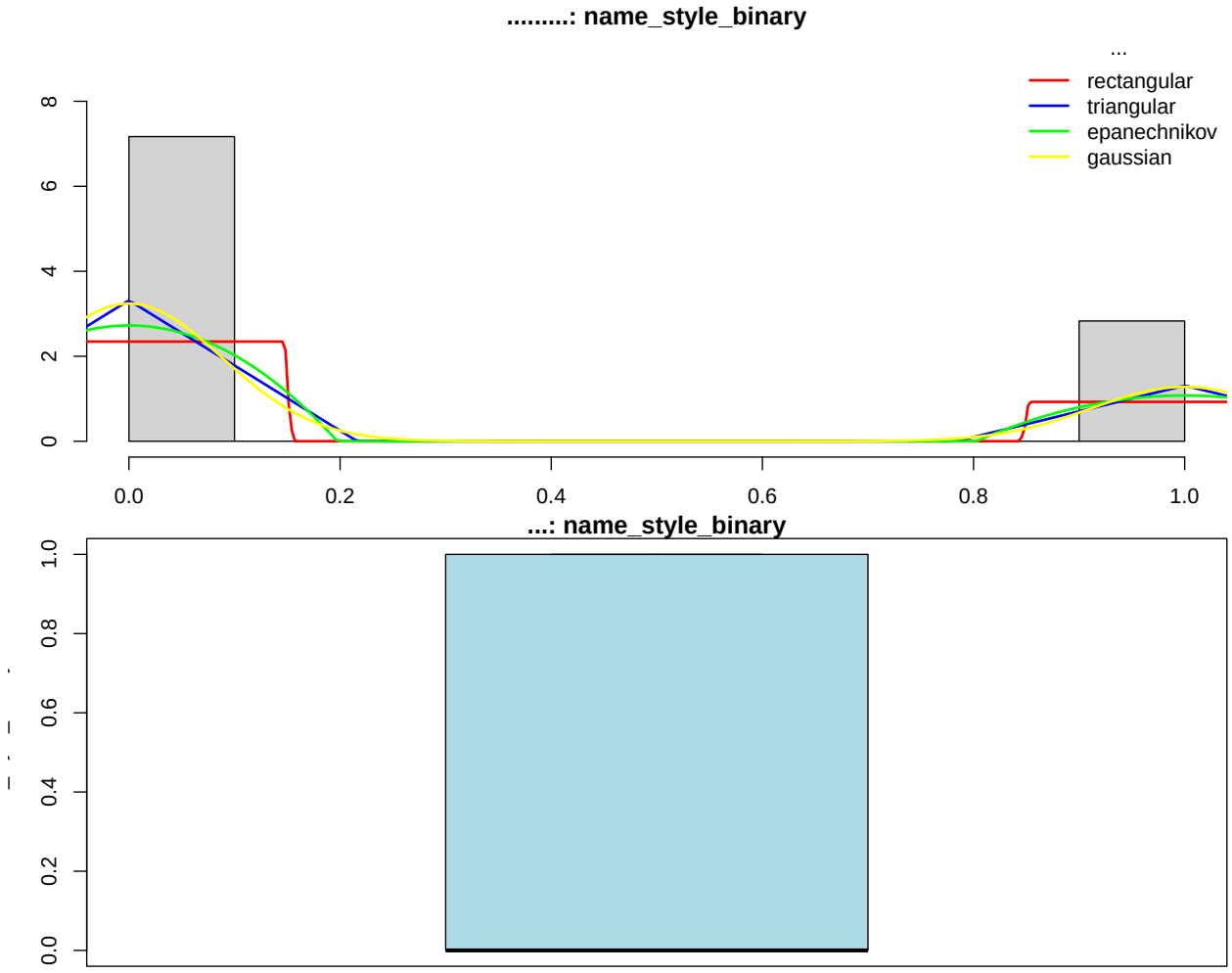

.....: length_intro
...
rectangular
triangular
epanechnikov
gaussian
0.000
0.002
0.004
0.006
0
200
400
600
800
1000
1200
1400
...: length_intro
0
200
400
600
800
1200

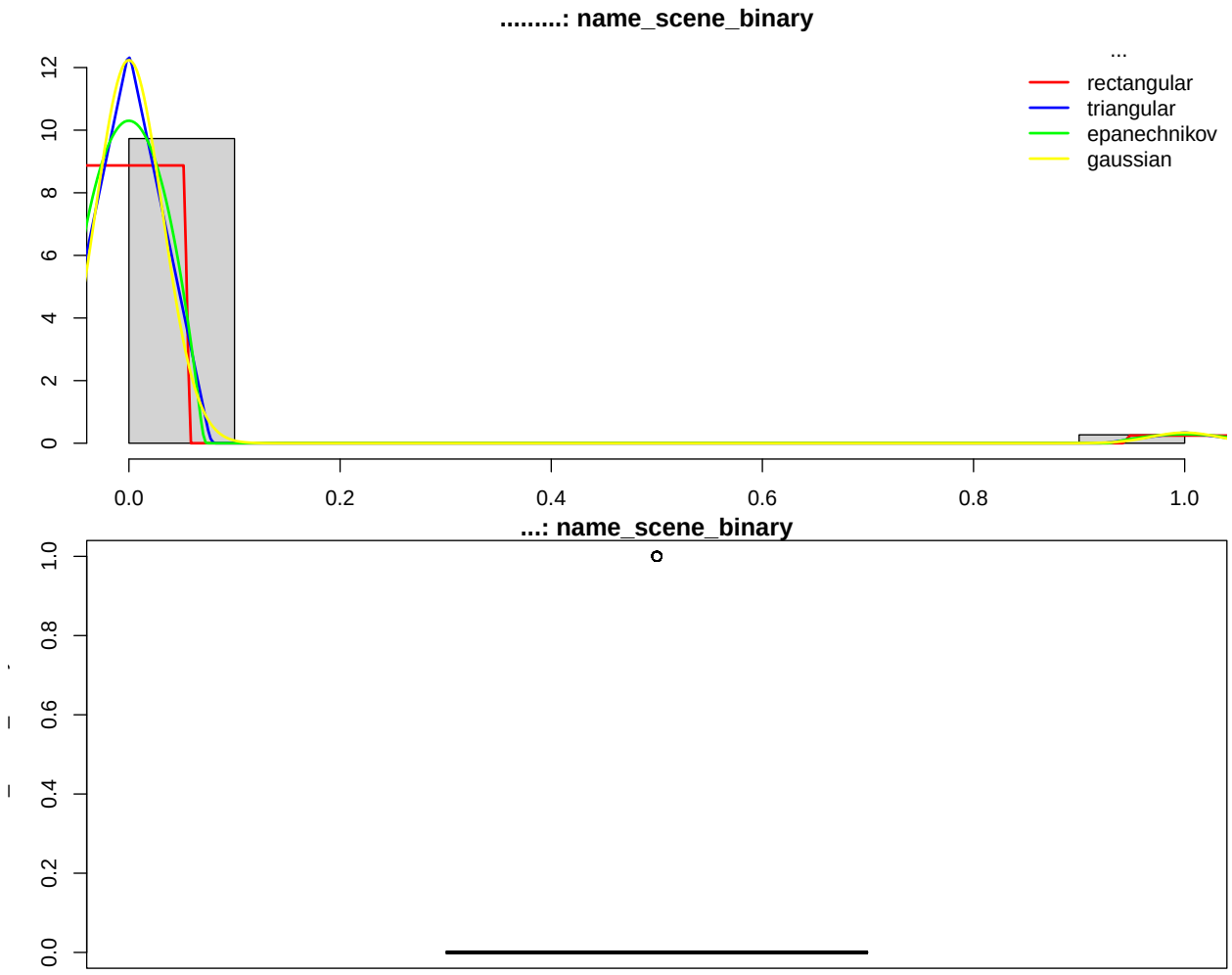

.....: number_songs
...
rectangular
triangular
epanechnikov
gaussian
0.000
0.002
0.004
0.006
0.008
0
200
400
600
800
1000
...: number_songs
0
200
400
600
800
1000

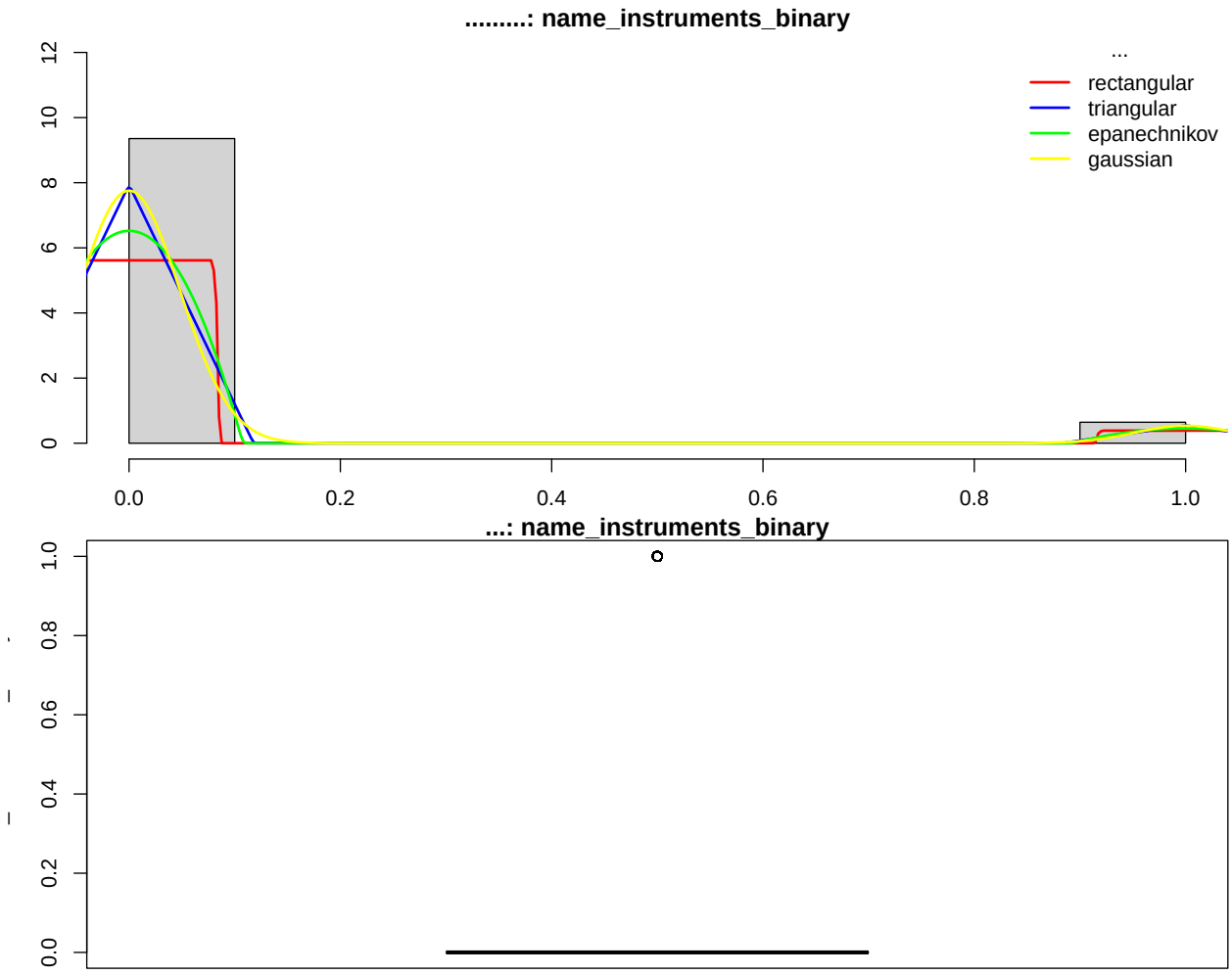

.....: number_hot_singers
...
rectangular
triangular
epanechnikov
gaussian
0.000
0.005
0.010
0.015
0.020
0.025
0
100
200
300
400
...: number_hot_singers
0
100
200
300
400

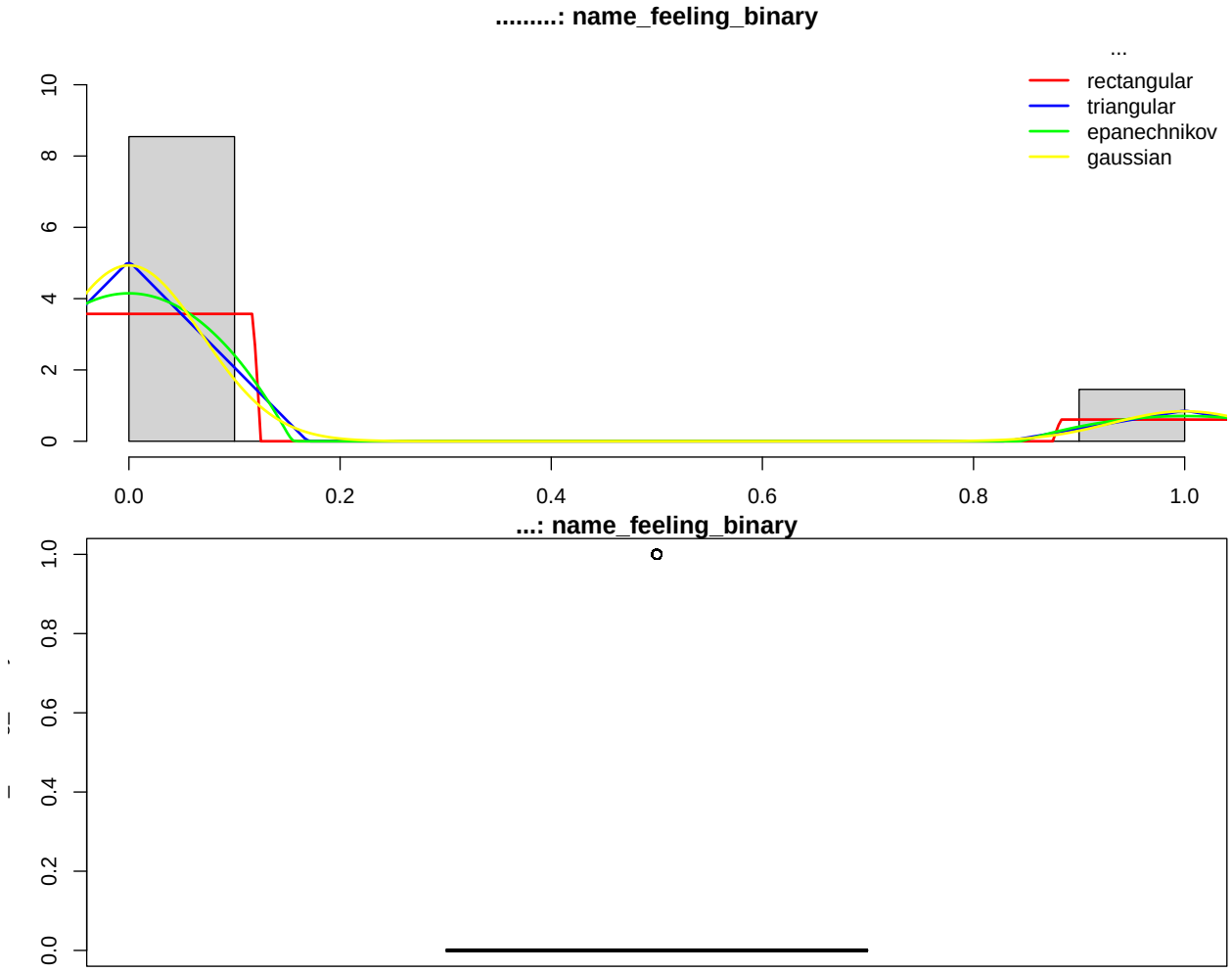

.....: english_name_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
...: english_name_binary
0.0
0.2
0.4
0.6
0.8
1.0

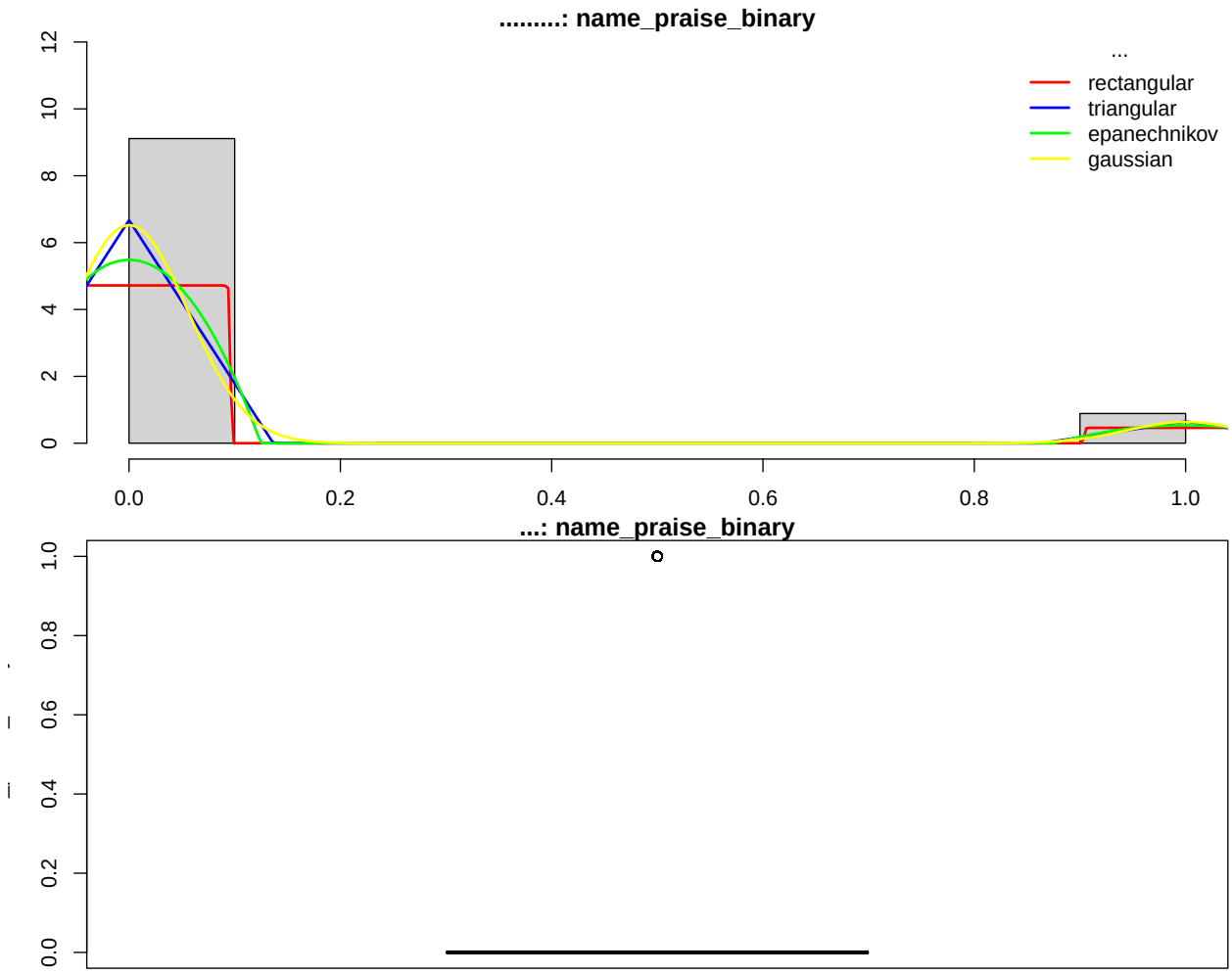

.....: name_language_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_language_binary
0.0
0.2
0.4
0.6
0.8
1.0

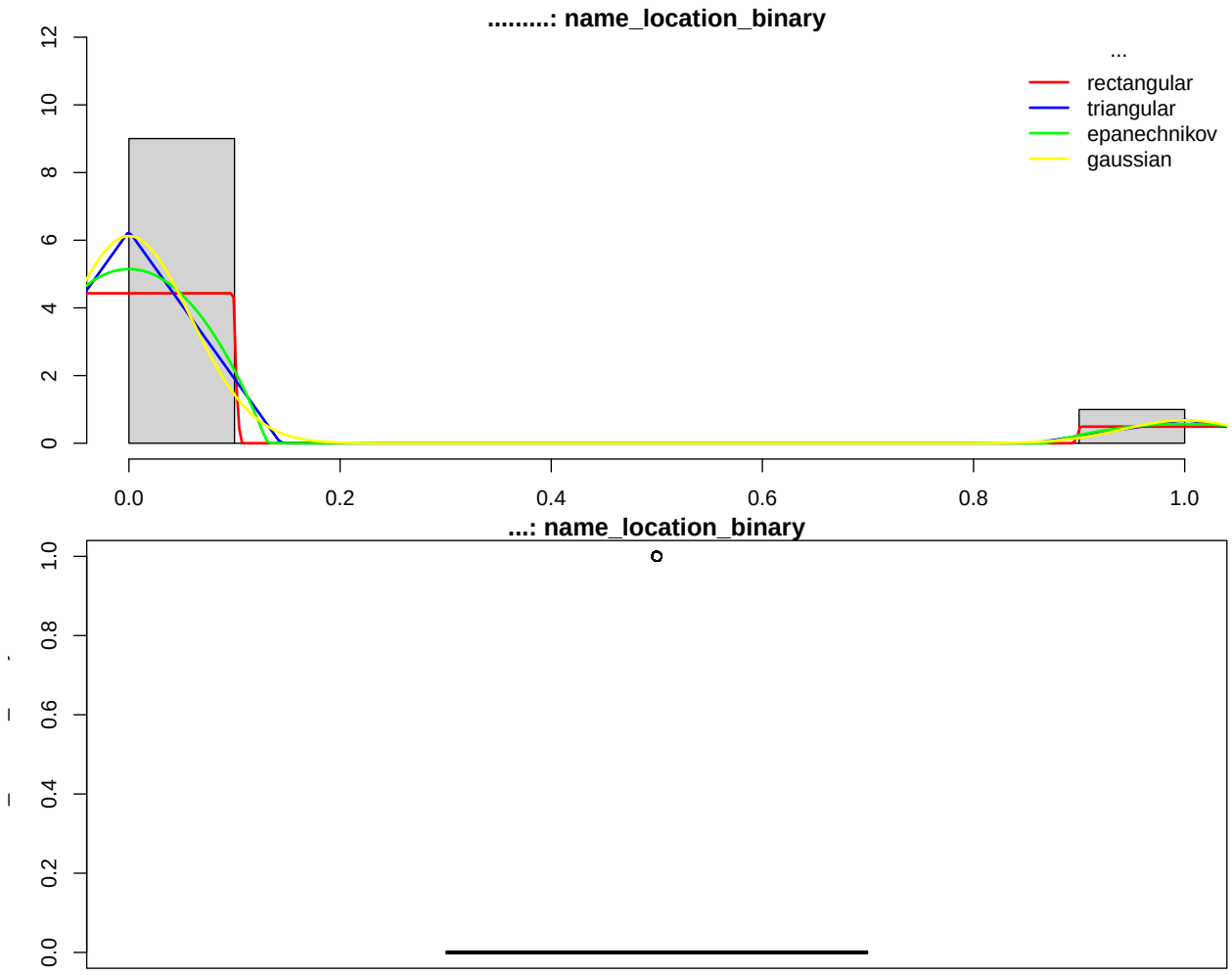

.....: name_style_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
...: name_style_binary
0.0
0.2
0.4
0.6
0.8
1.0

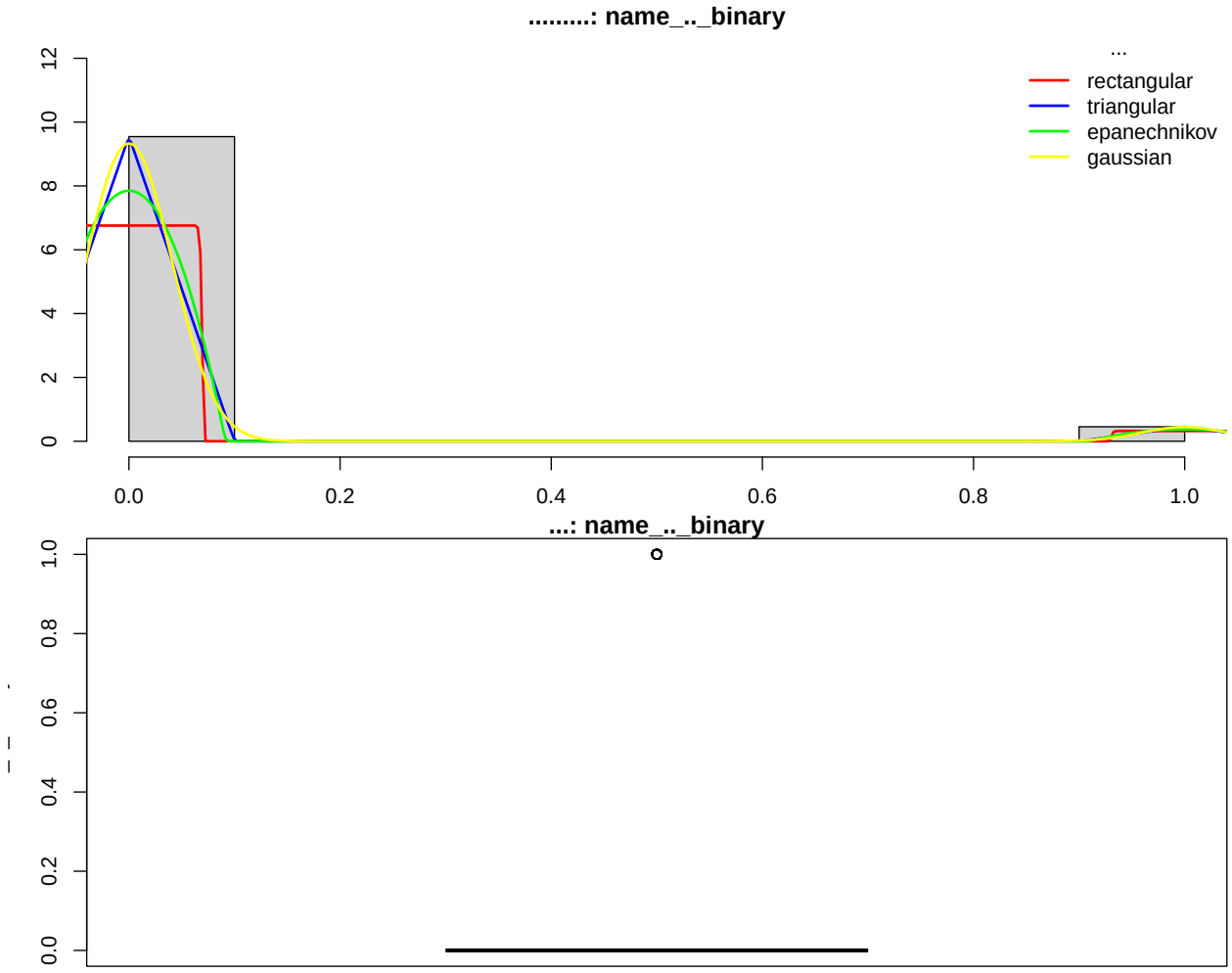

.....: name_scene_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_scene_binary
0.0
0.2
0.4
0.6
0.8
1.0

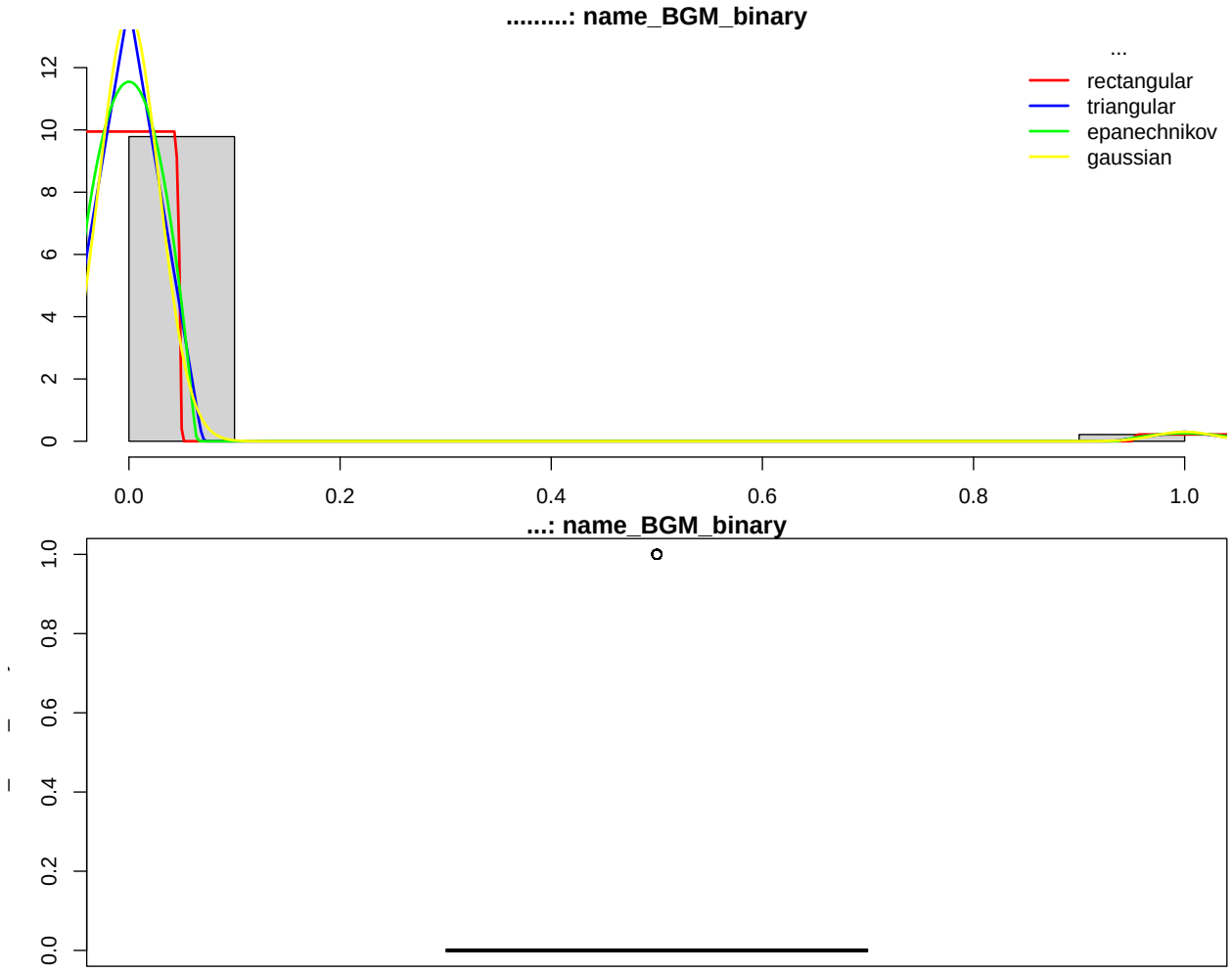

.....: name_instruments_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_instruments_binary
0.0
0.2
0.4
0.6
0.8
1.0

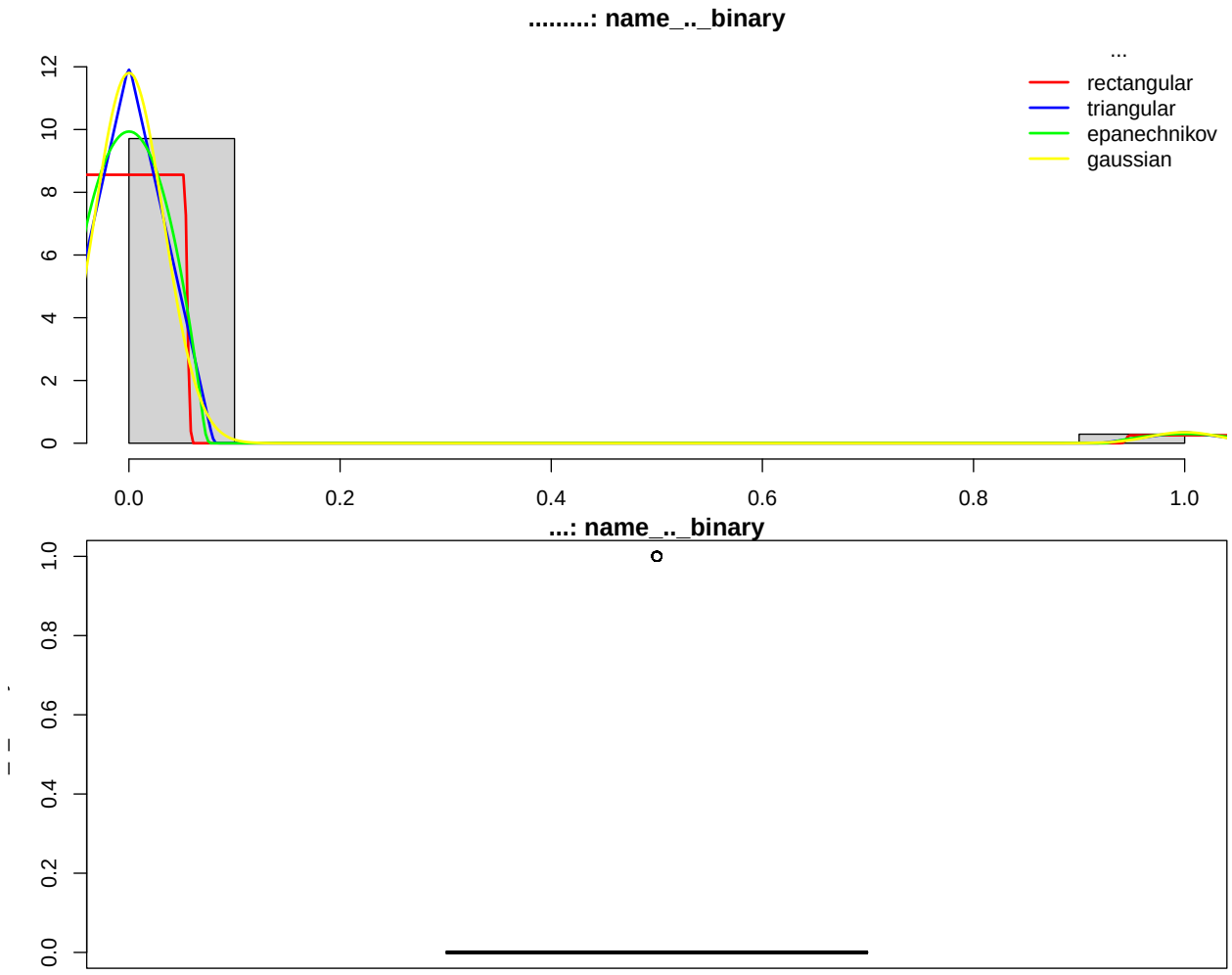

.....: name_feeling_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: name_feeling_binary
0.0
0.2
0.4
0.6
0.8
1.0

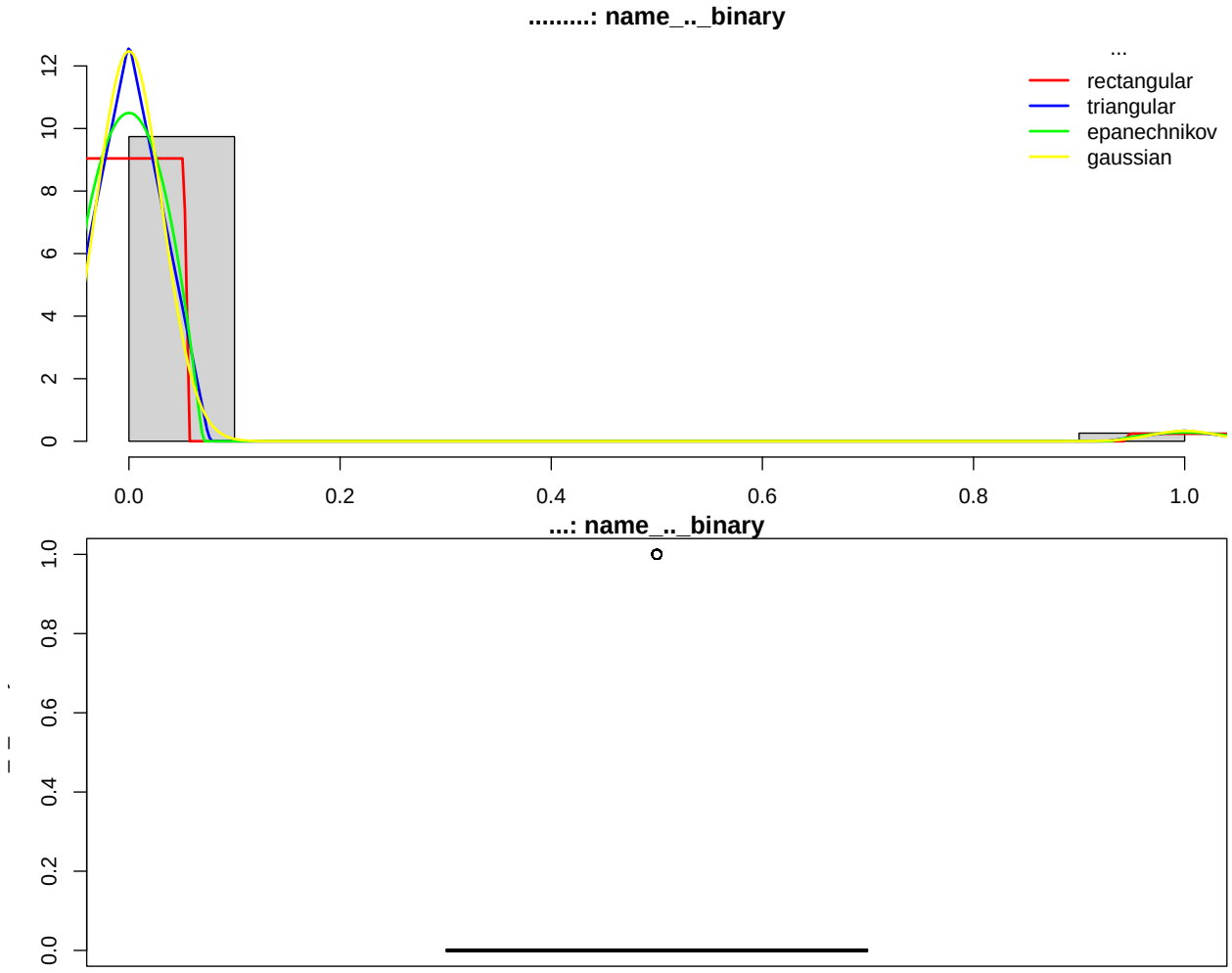

.....: name_praise_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_praise_binary
0.0
0.2
0.4
0.6
0.8
1.0

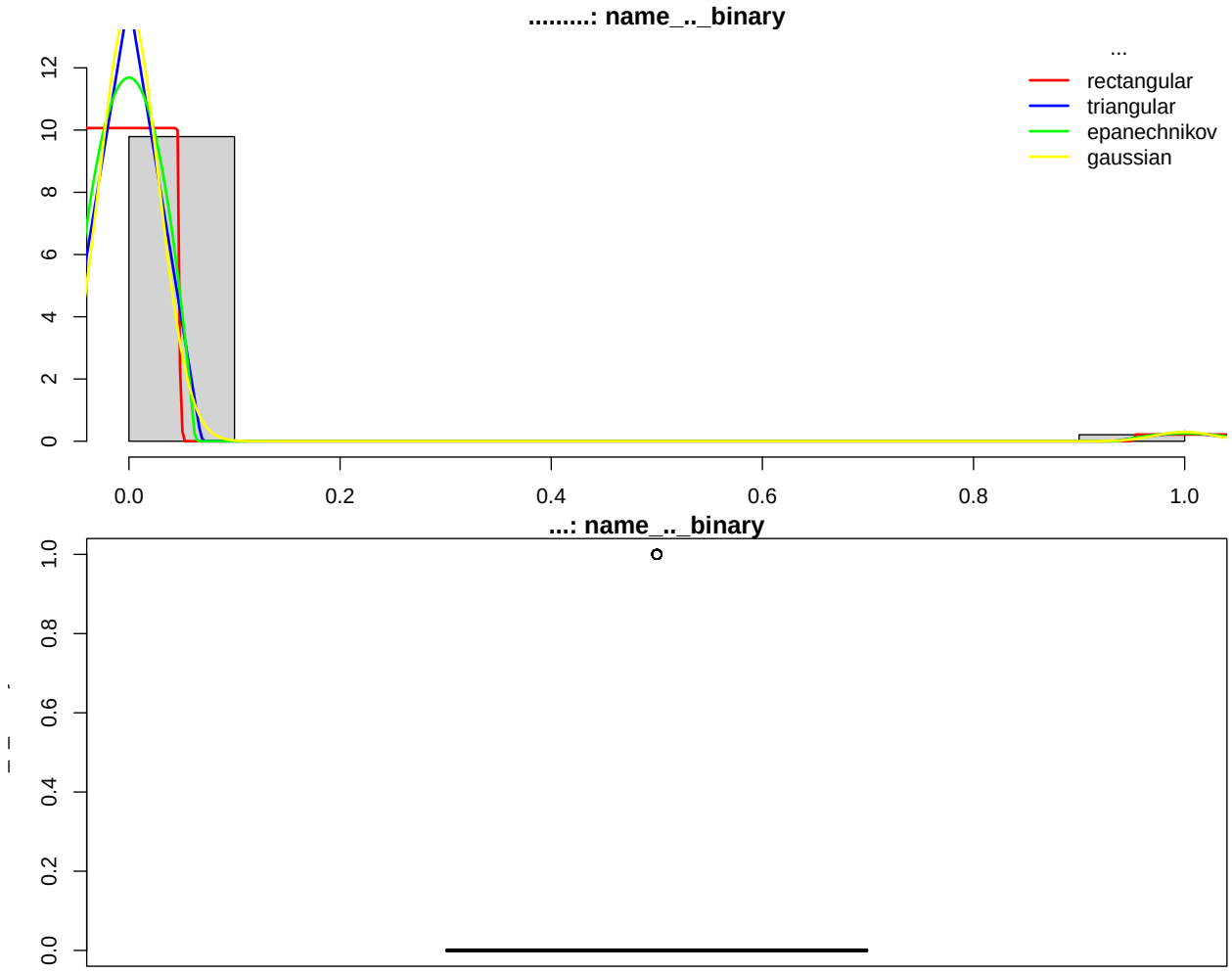

.....: name_location_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_location_binary
0.0
0.2
0.4
0.6
0.8
1.0

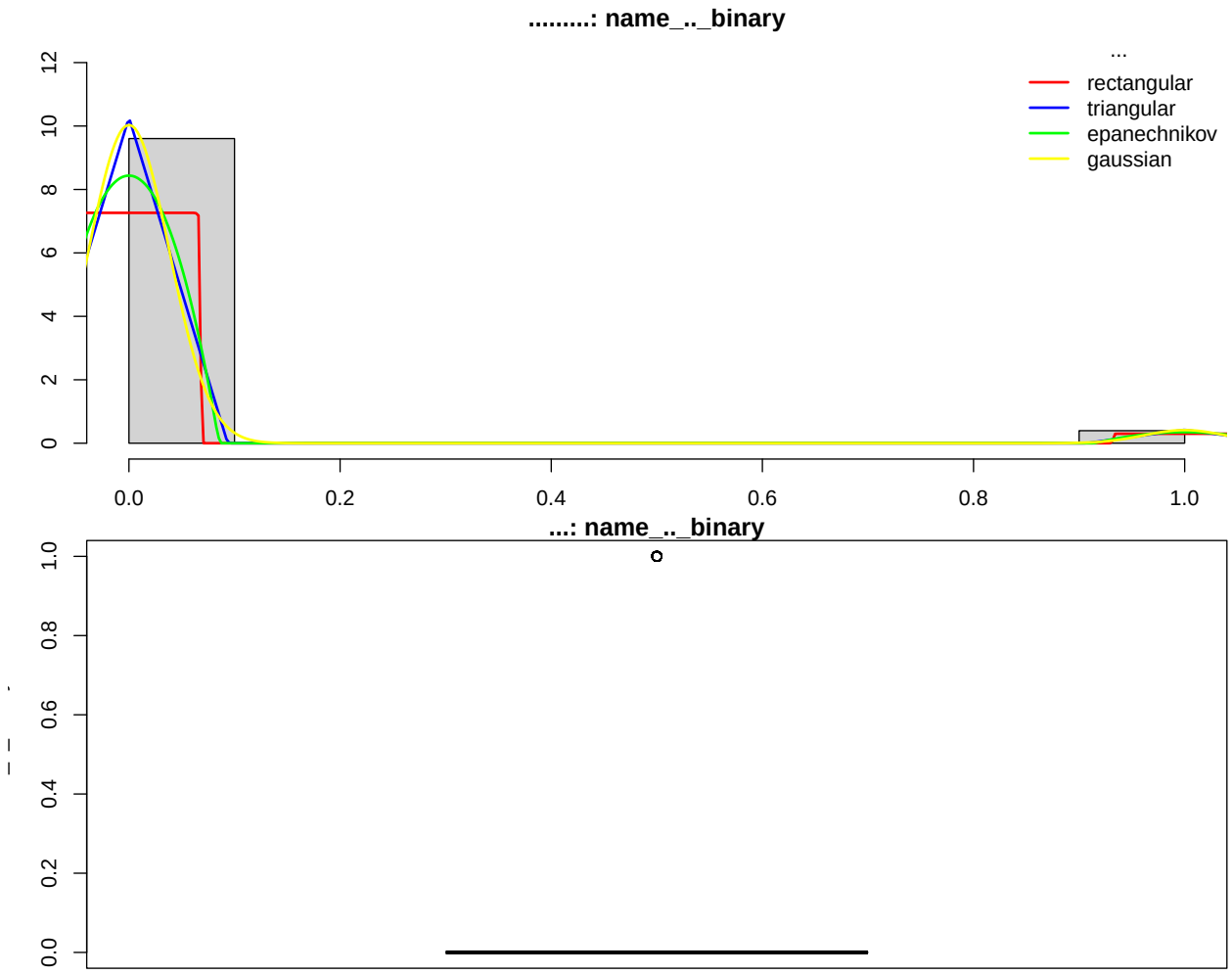

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

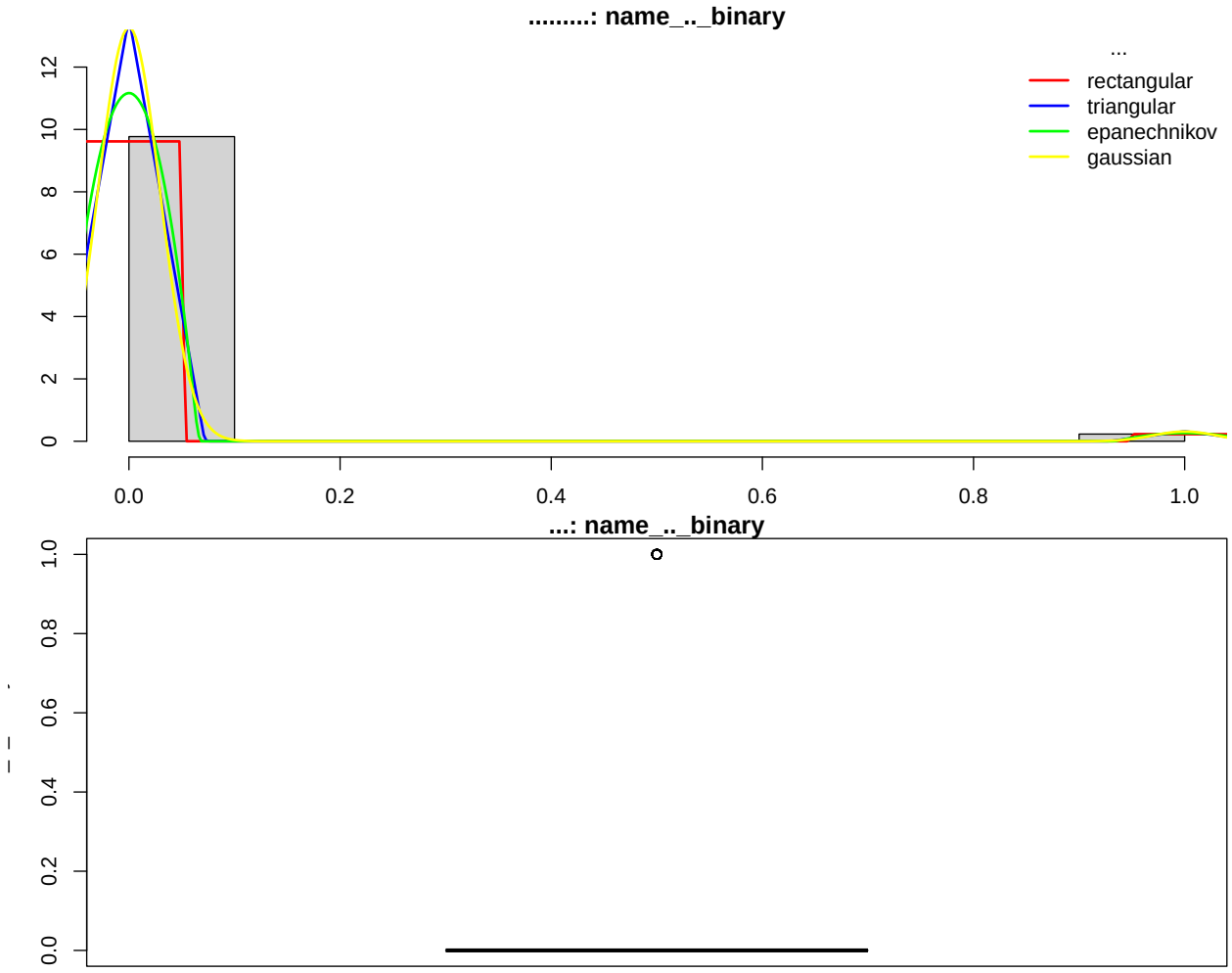

.....: name_BGM_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_BGM_binary
0.0
0.2
0.4
0.6
0.8
1.0

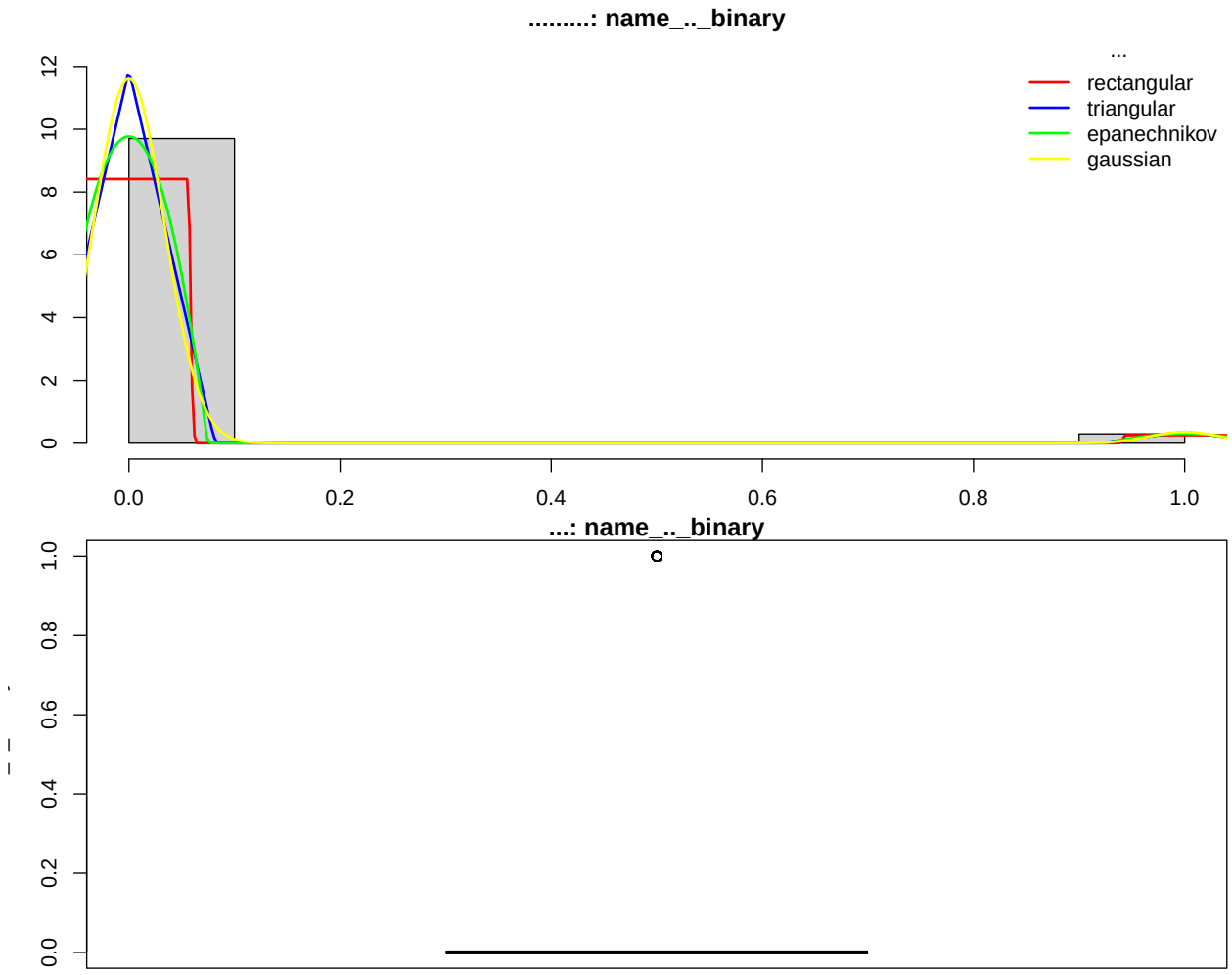

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

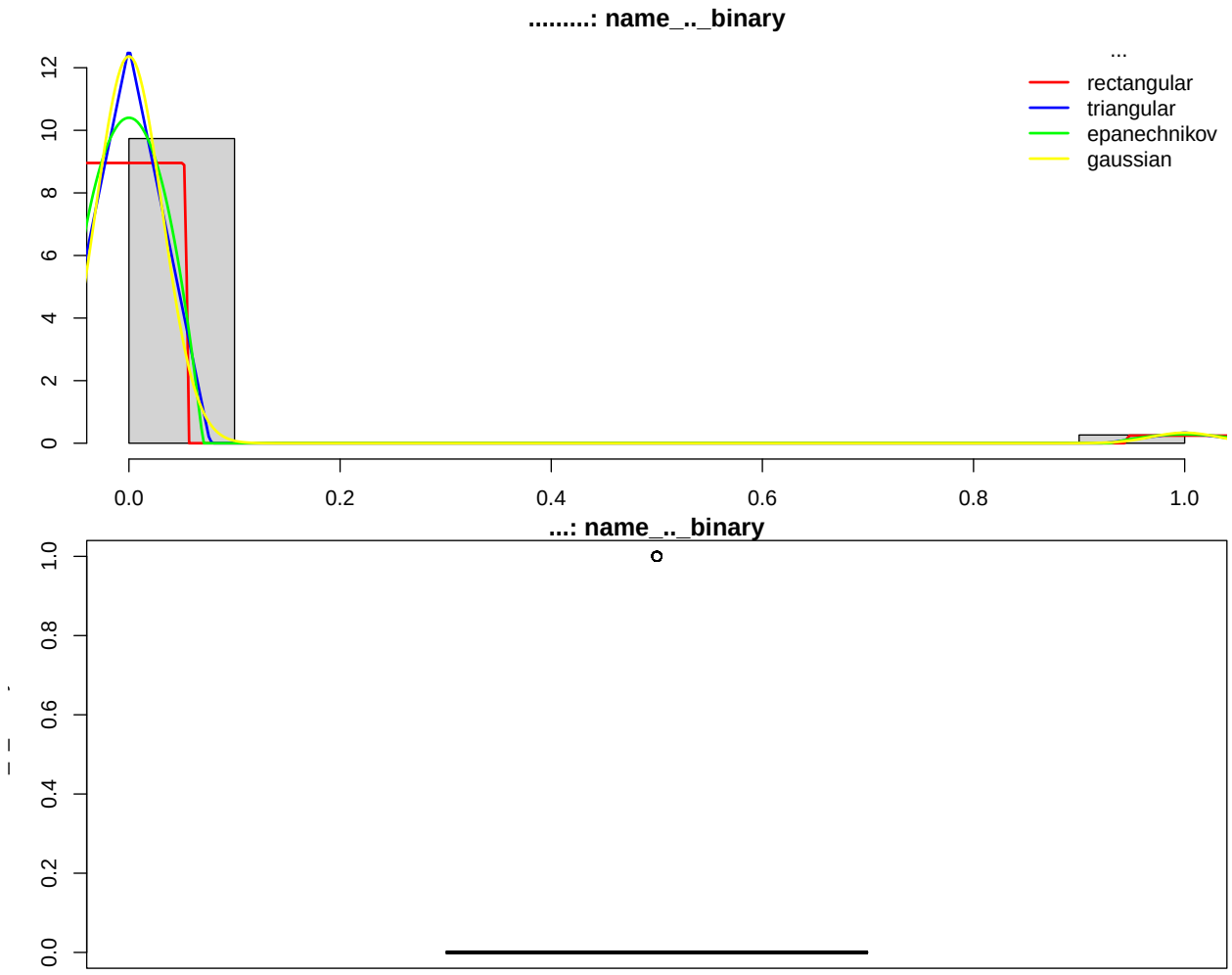

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

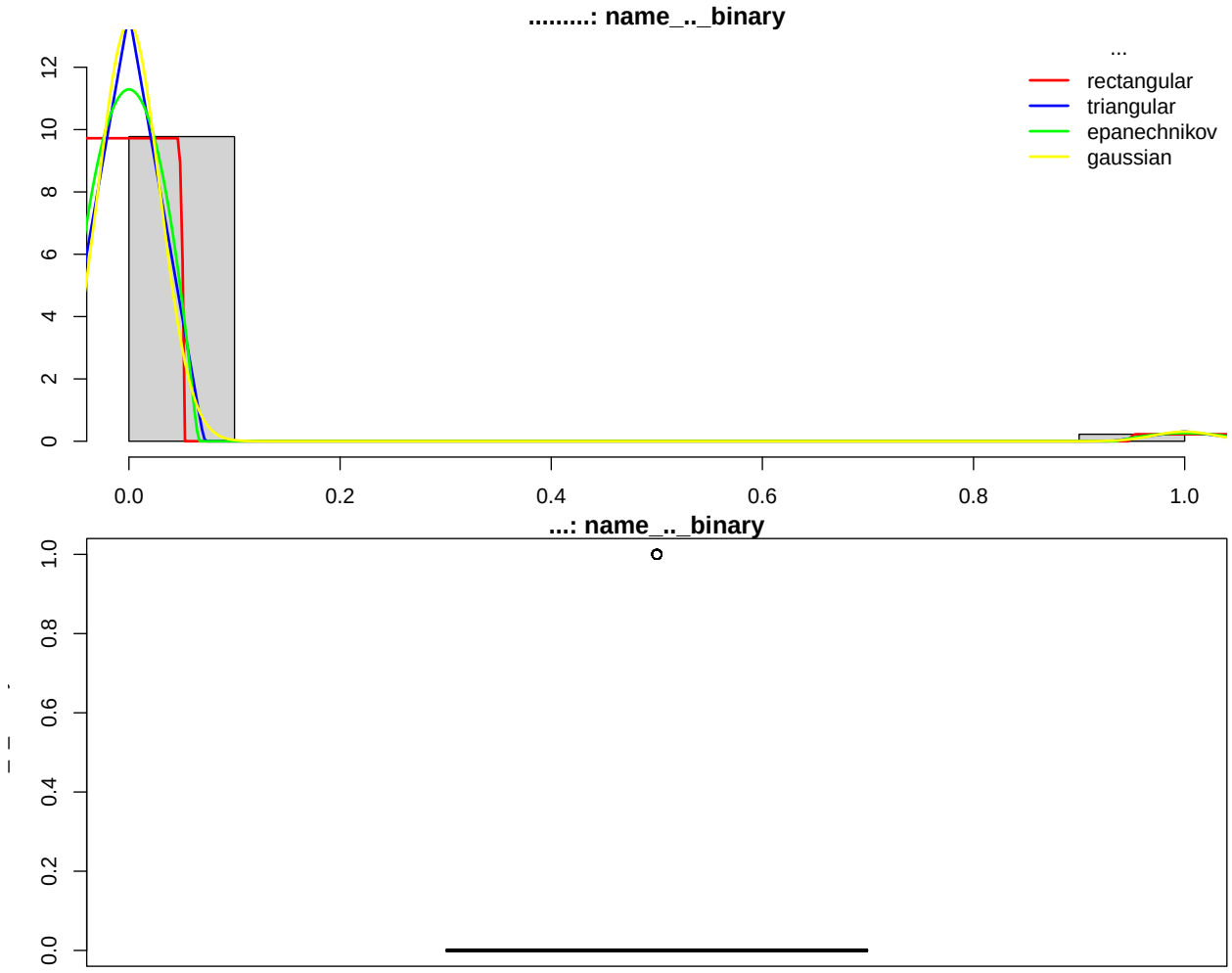

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

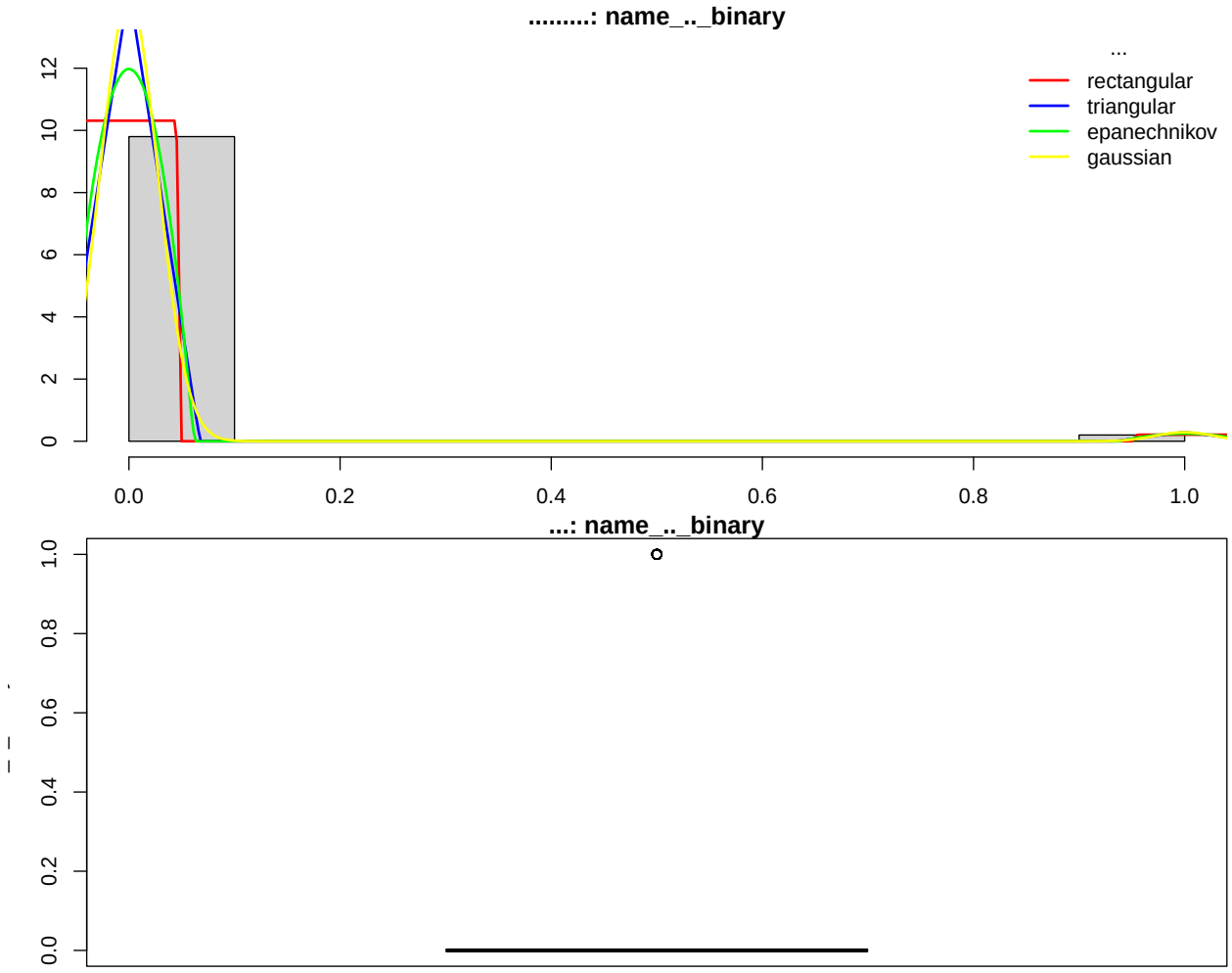

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

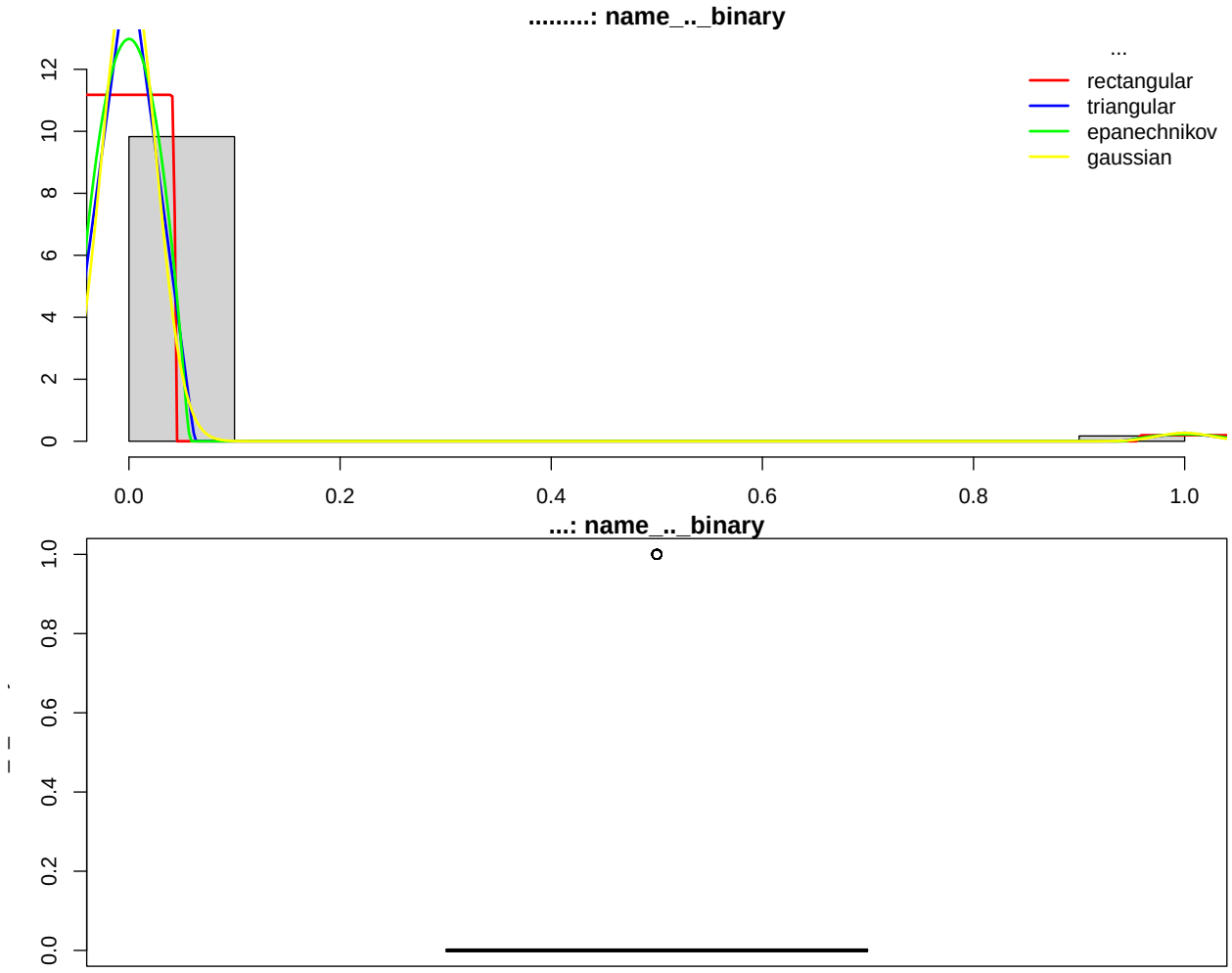

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

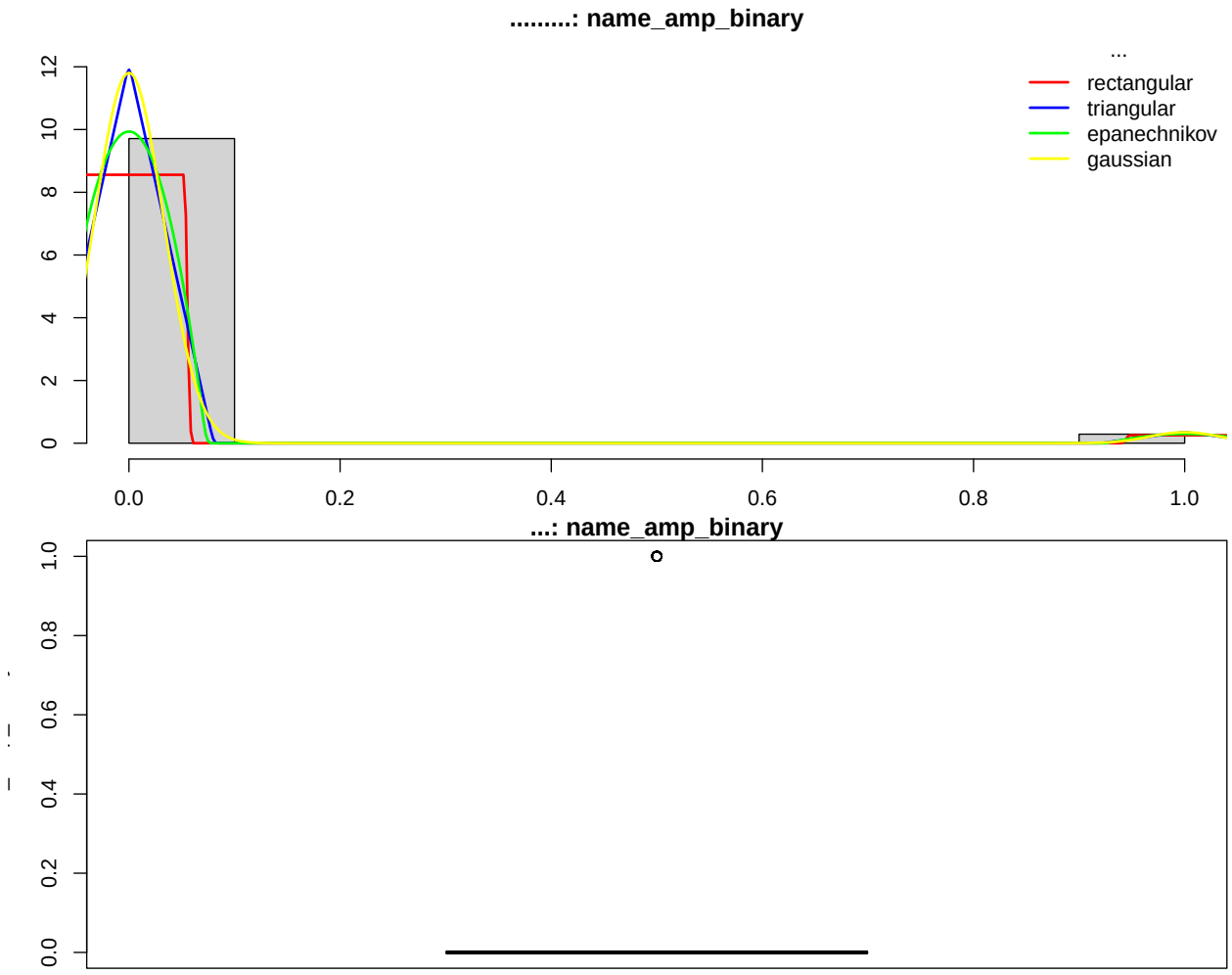

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

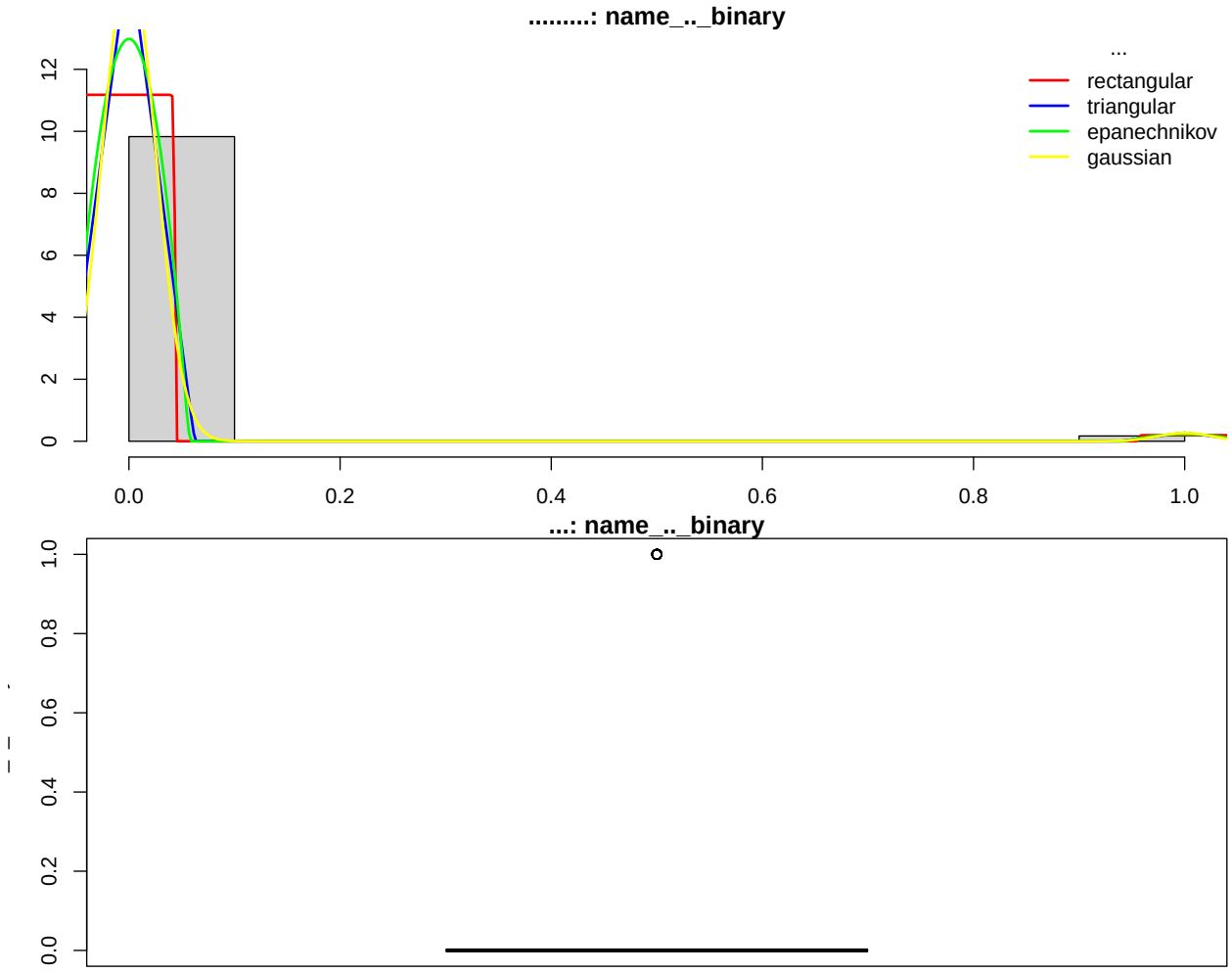

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

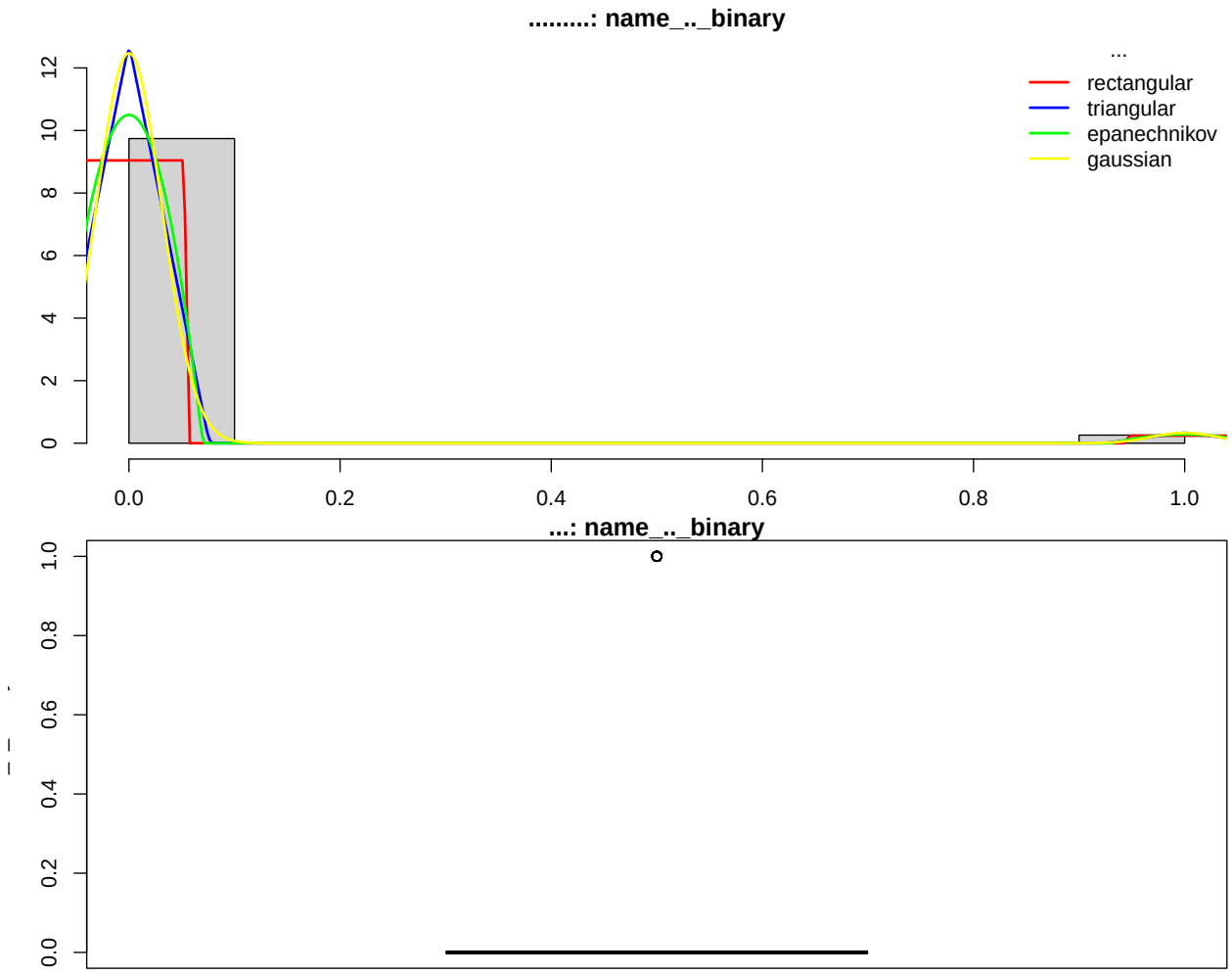

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

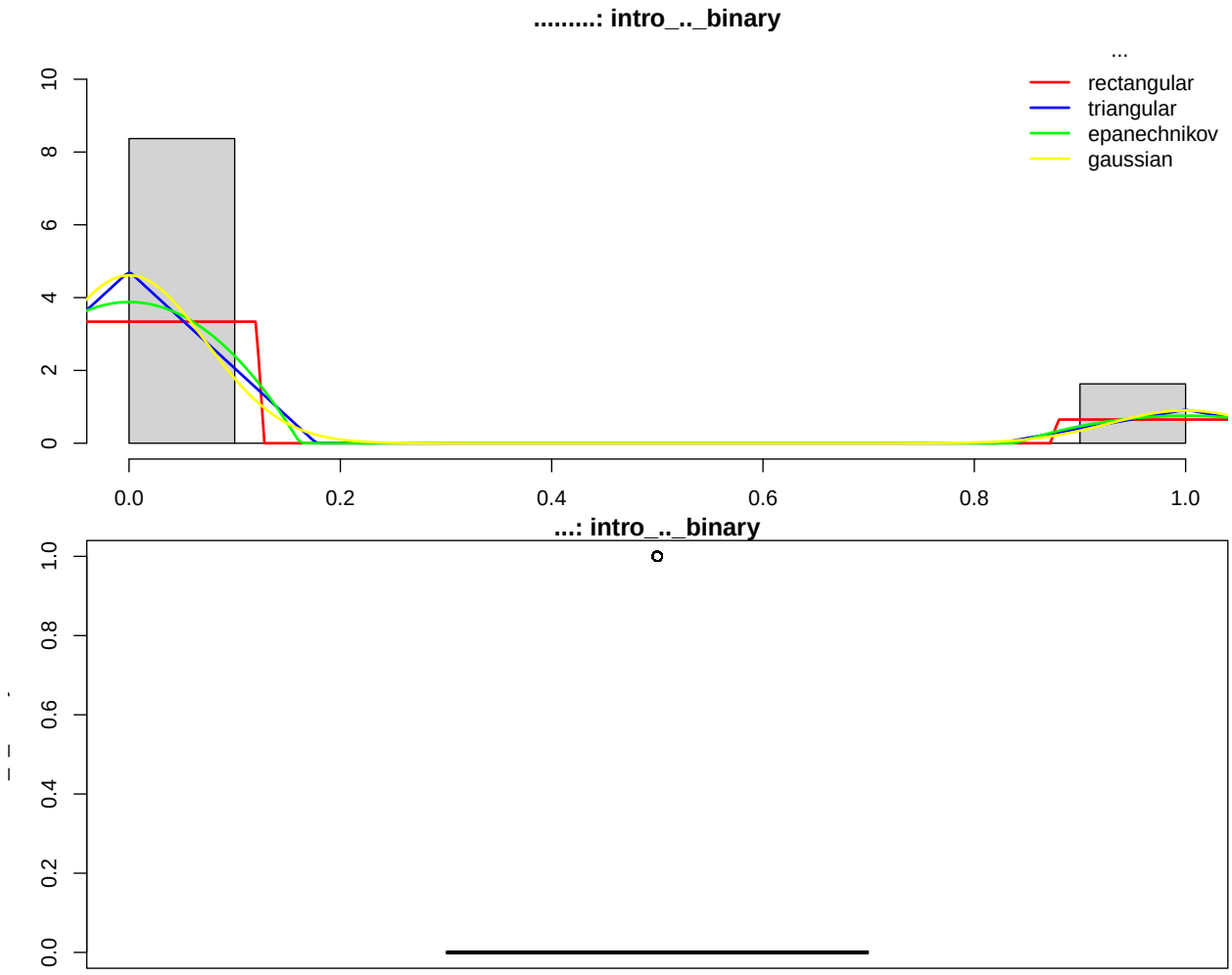

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

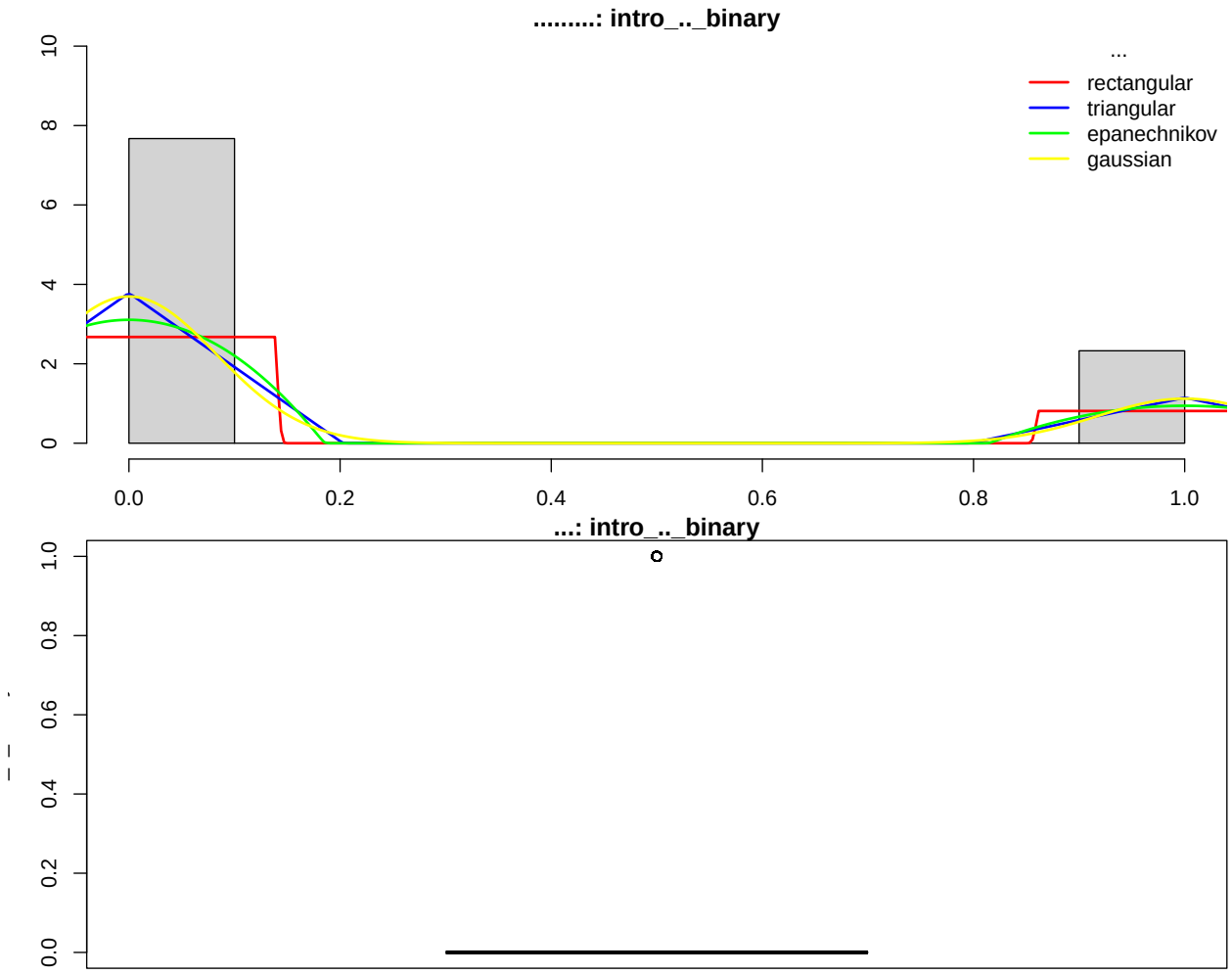

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

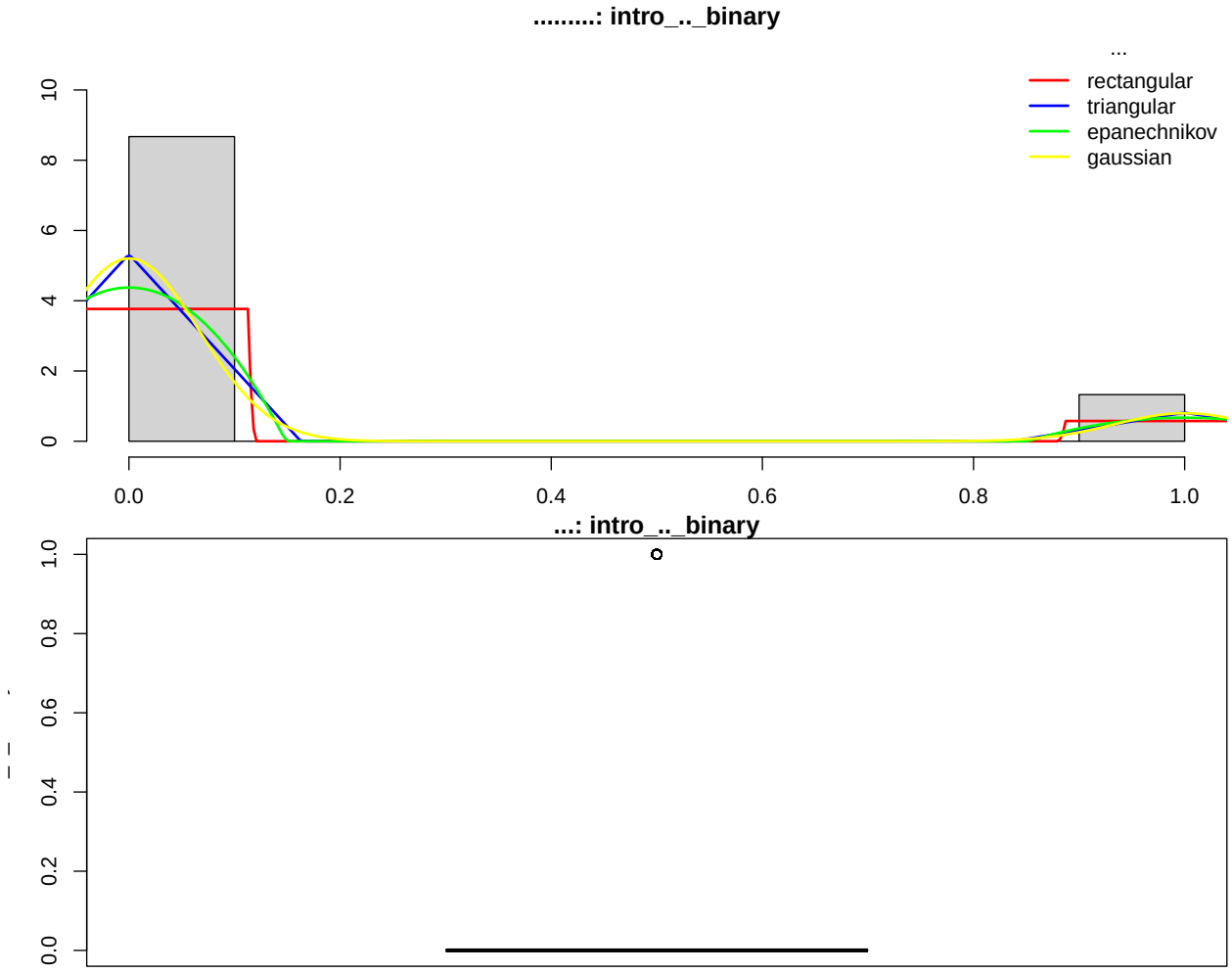

.....: name_amp_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_amp_binary
0.0
0.2
0.4
0.6
0.8
1.0

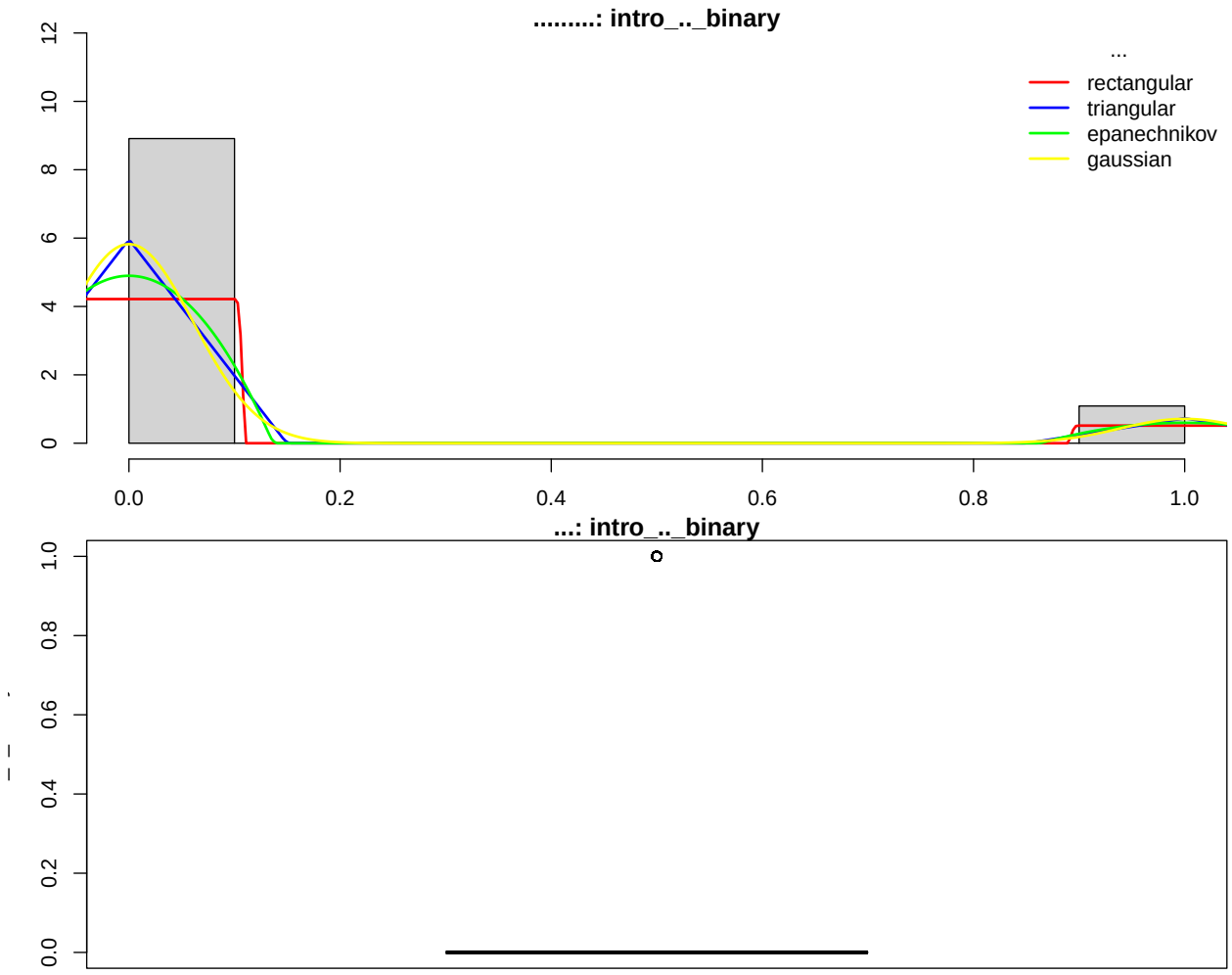

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

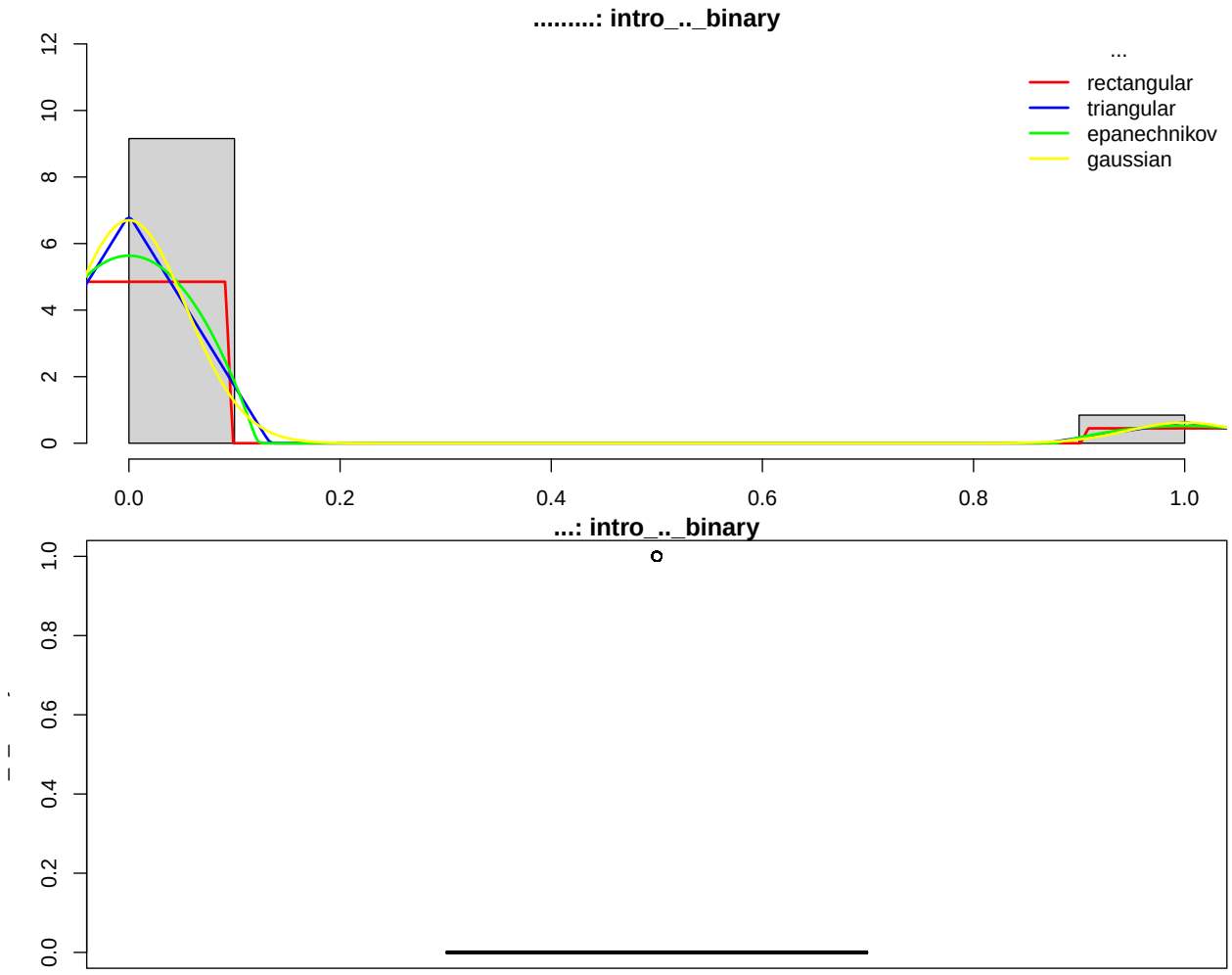

.....: name_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: name_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

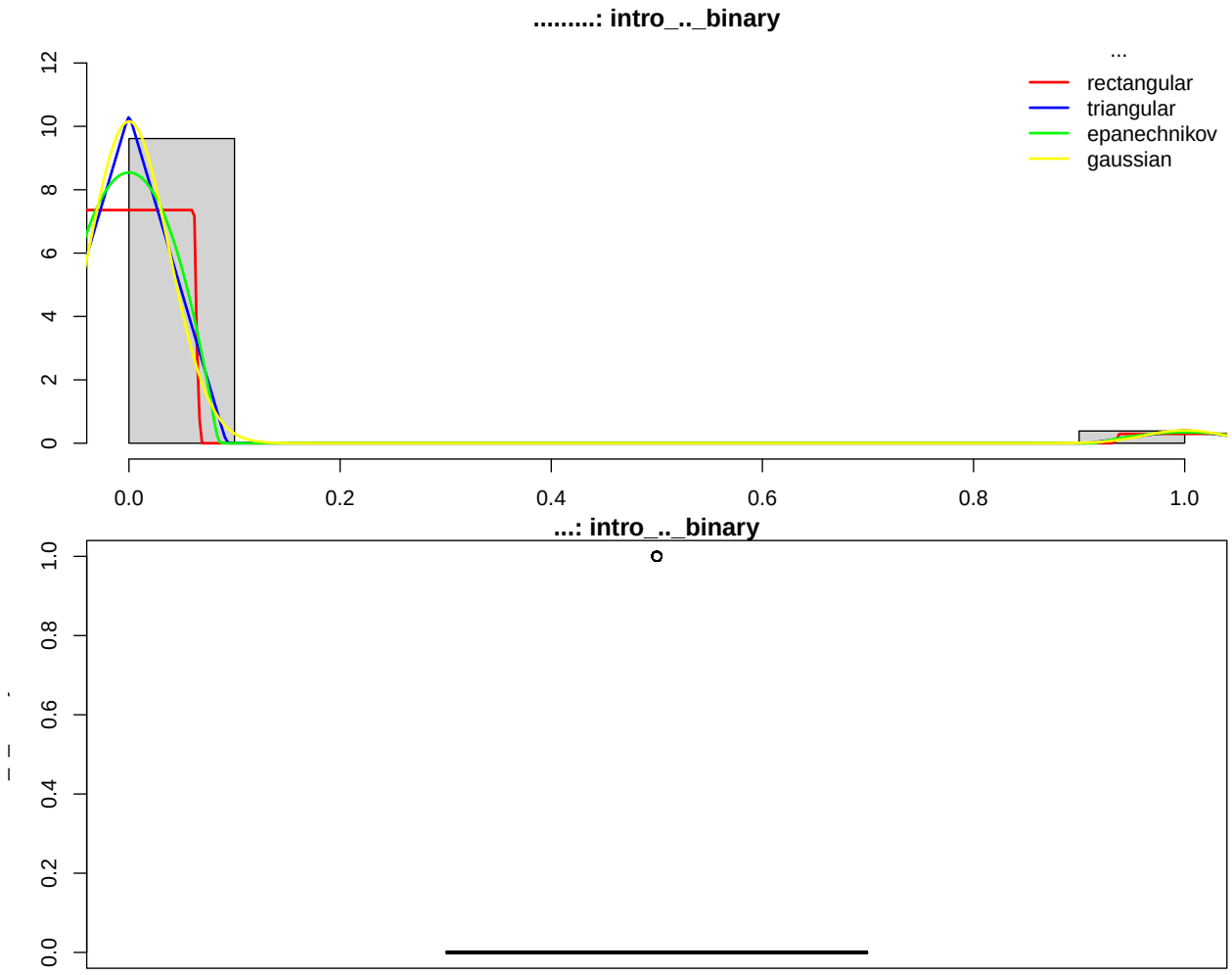

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

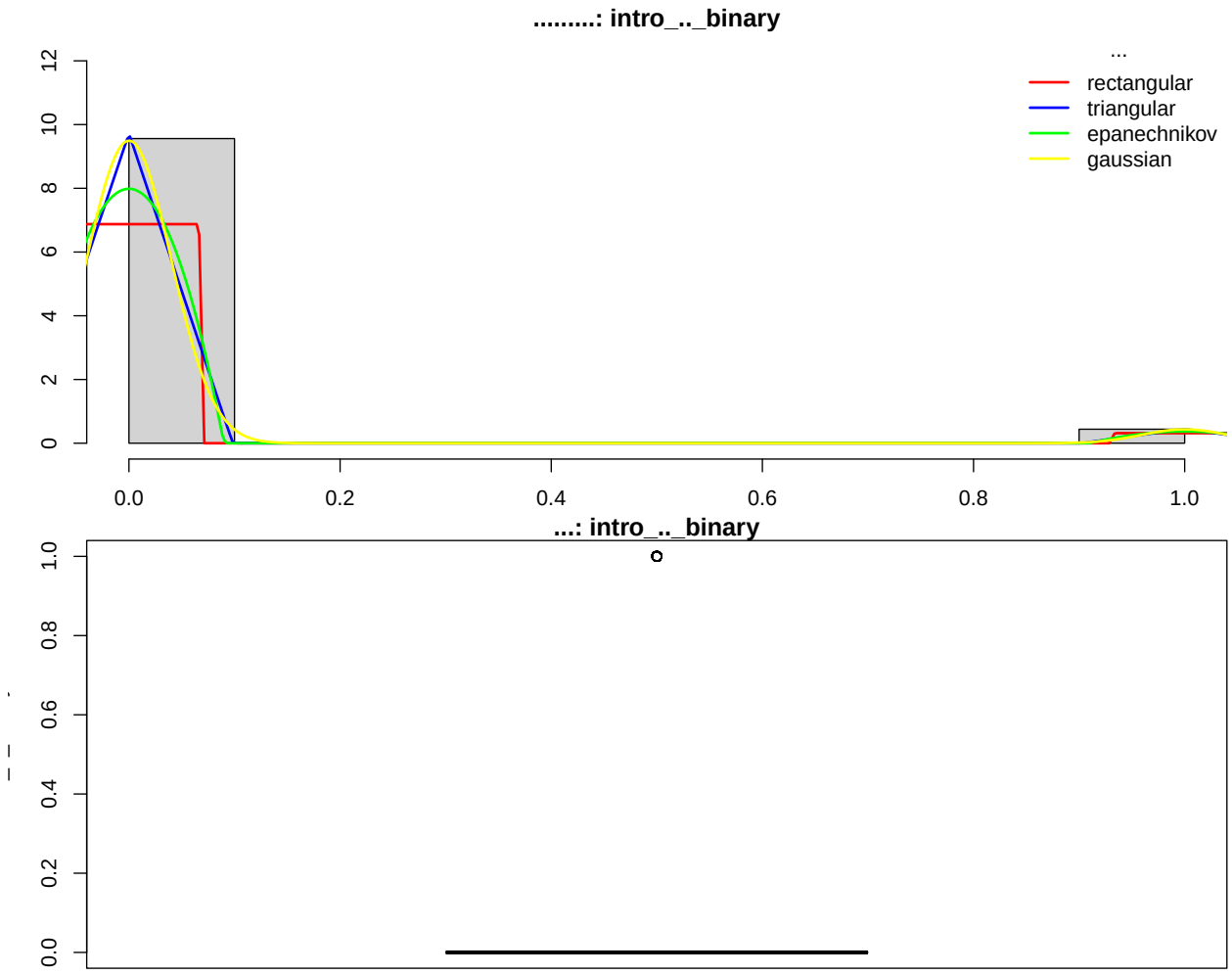

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

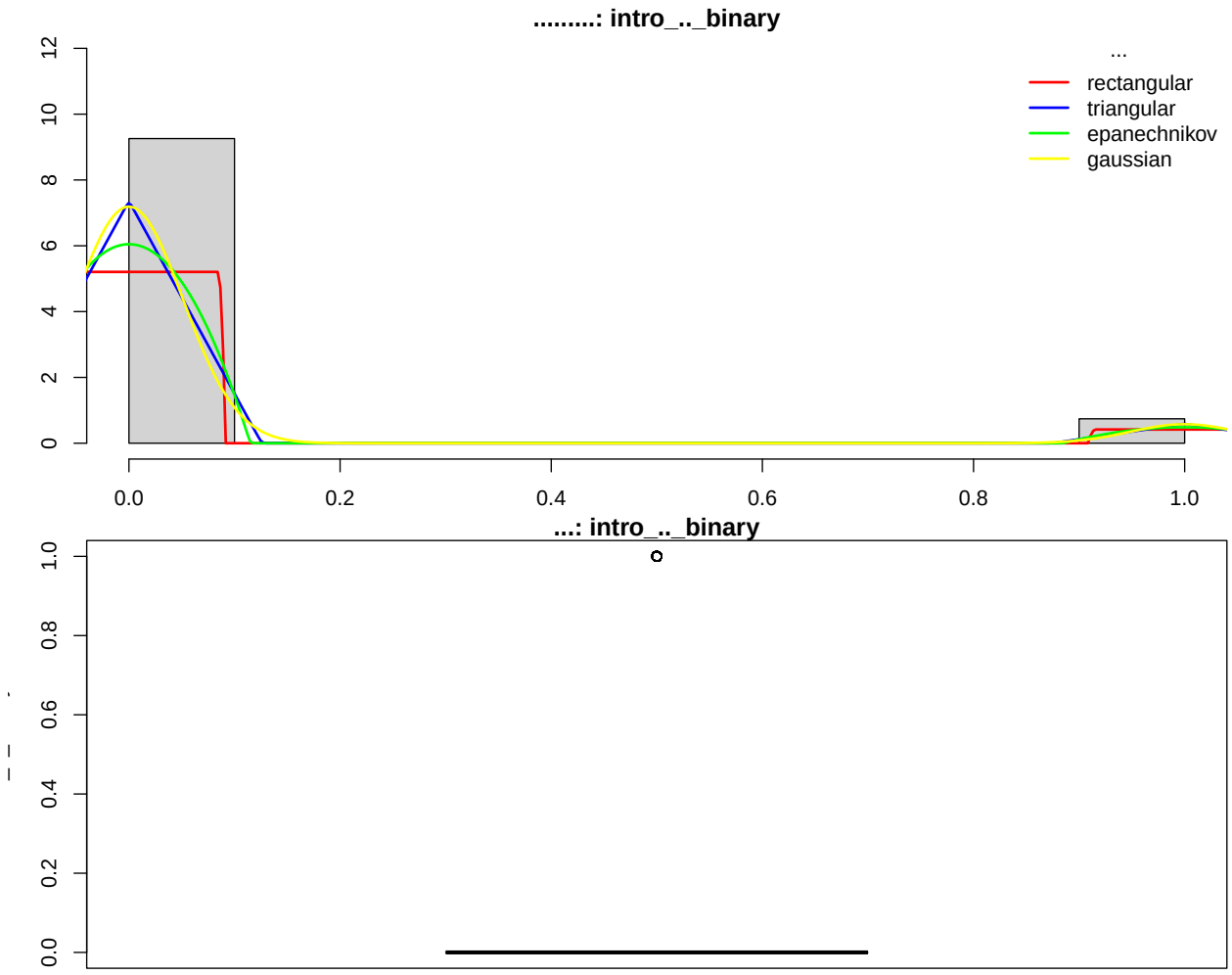

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

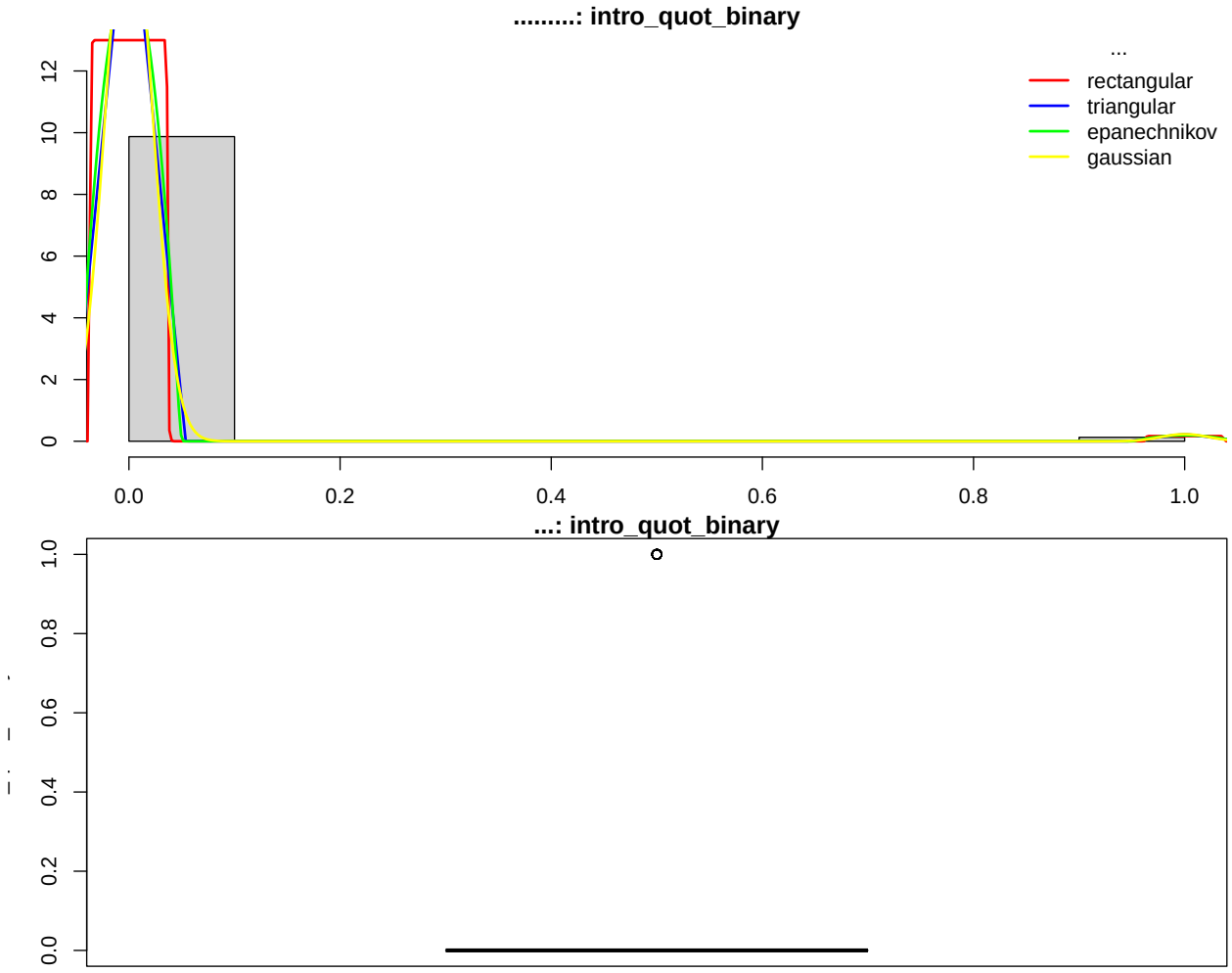

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

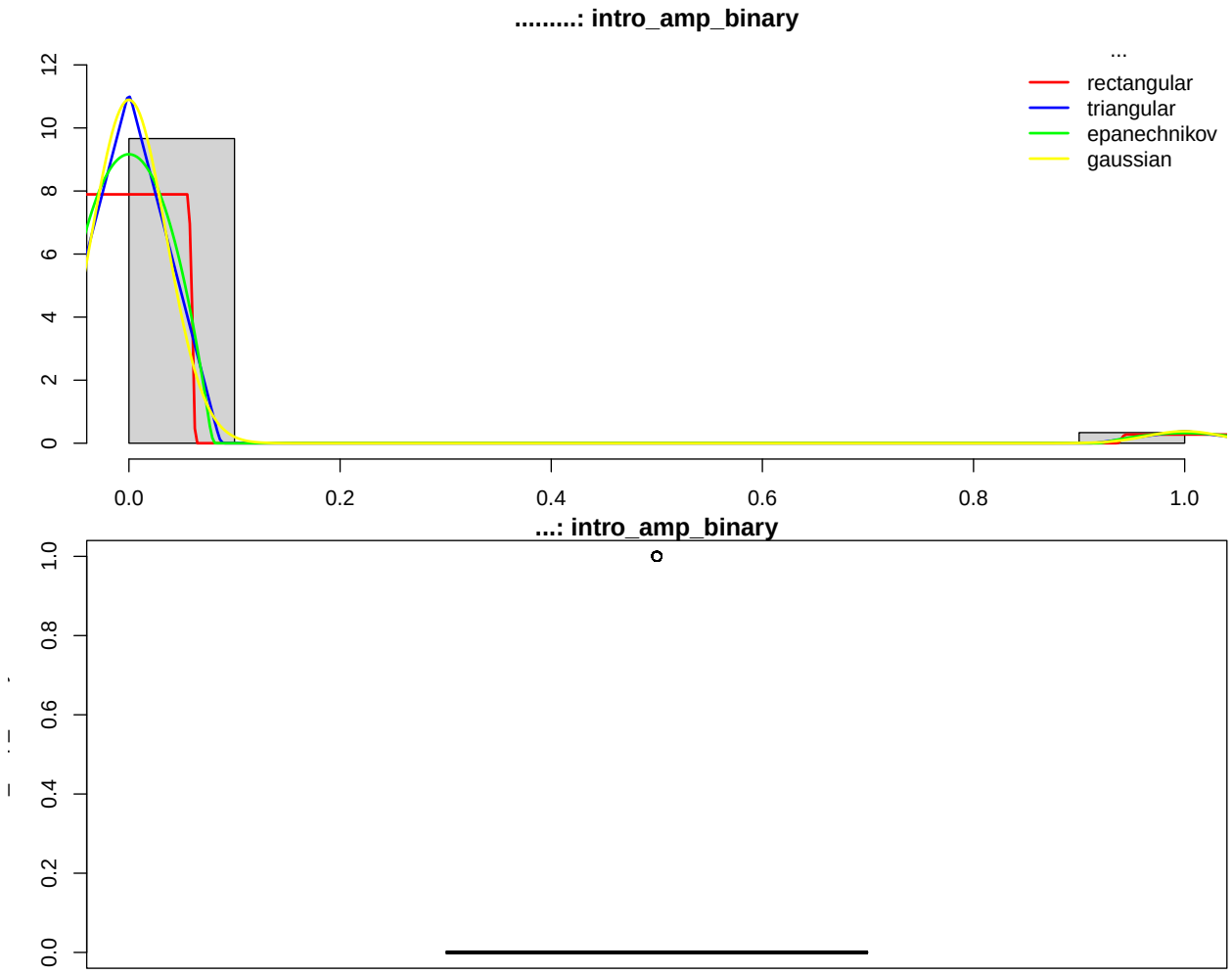

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

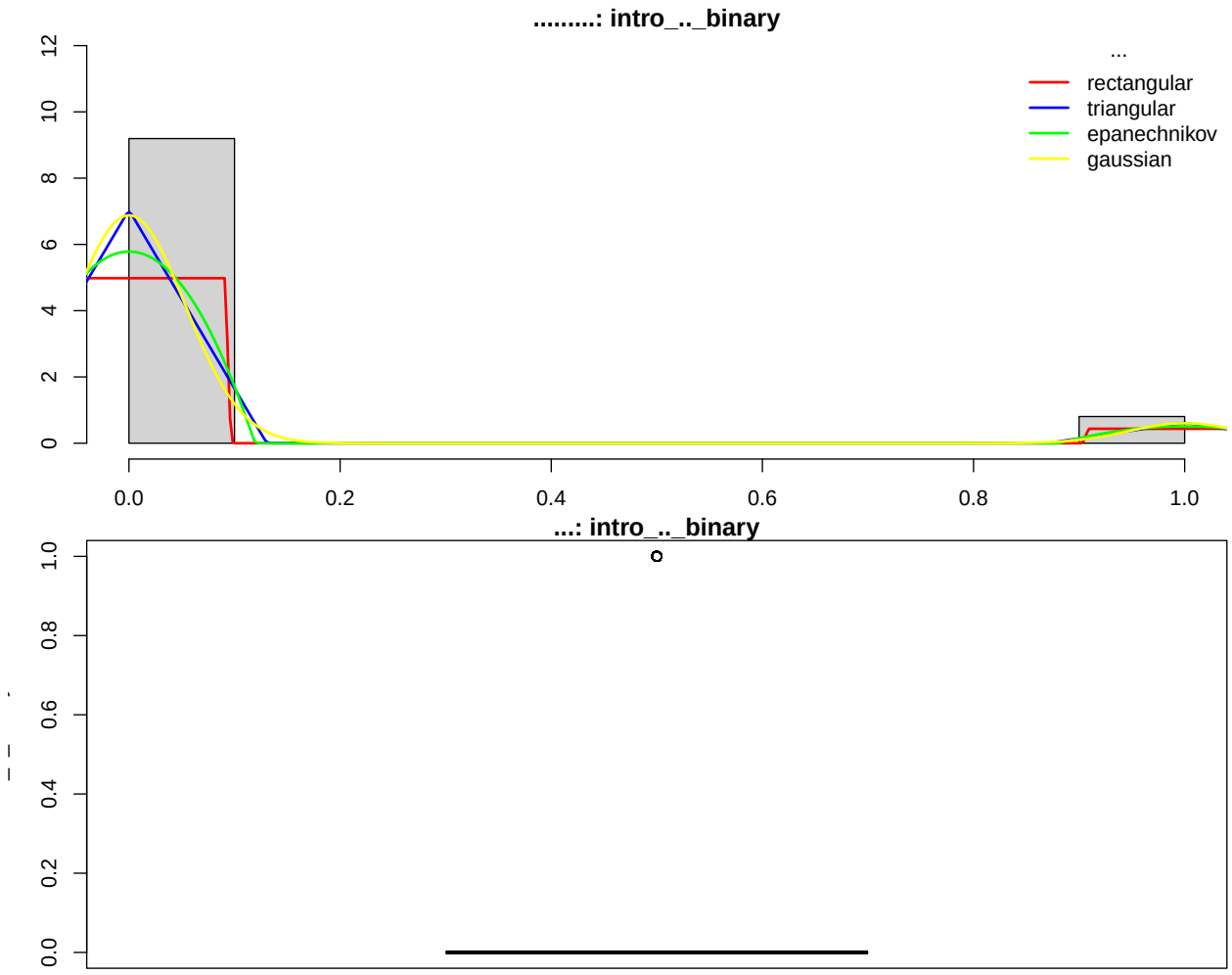

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

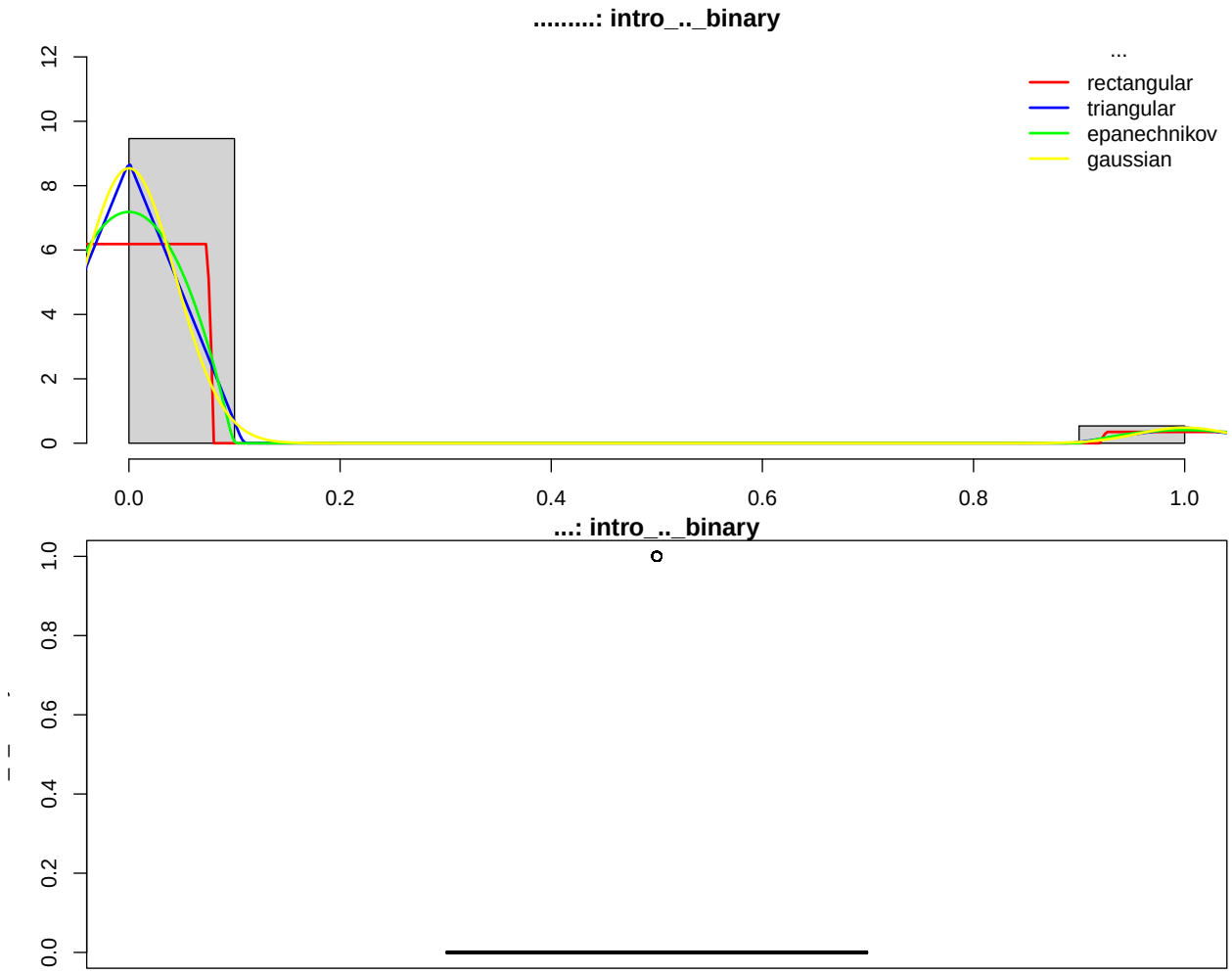

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

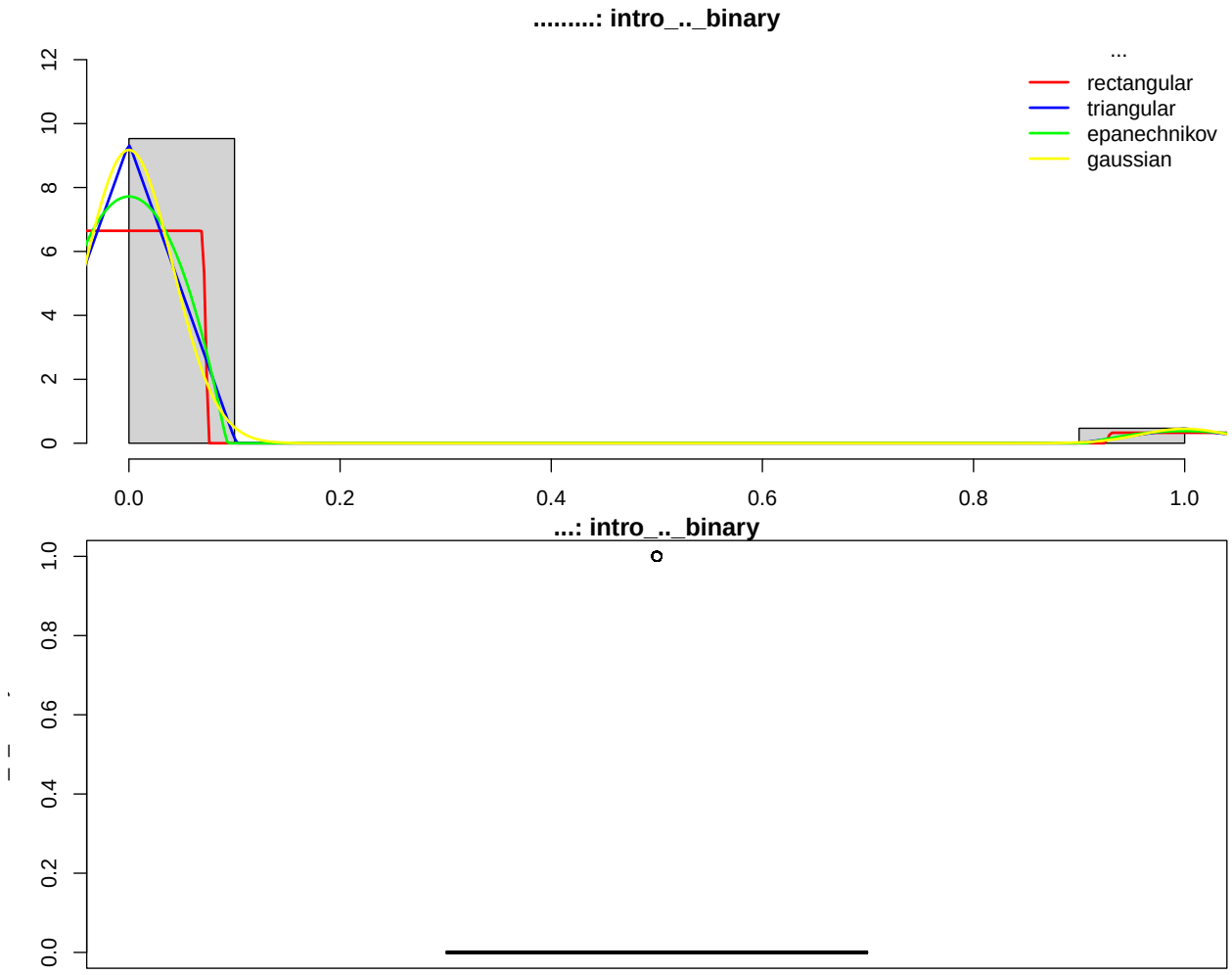

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

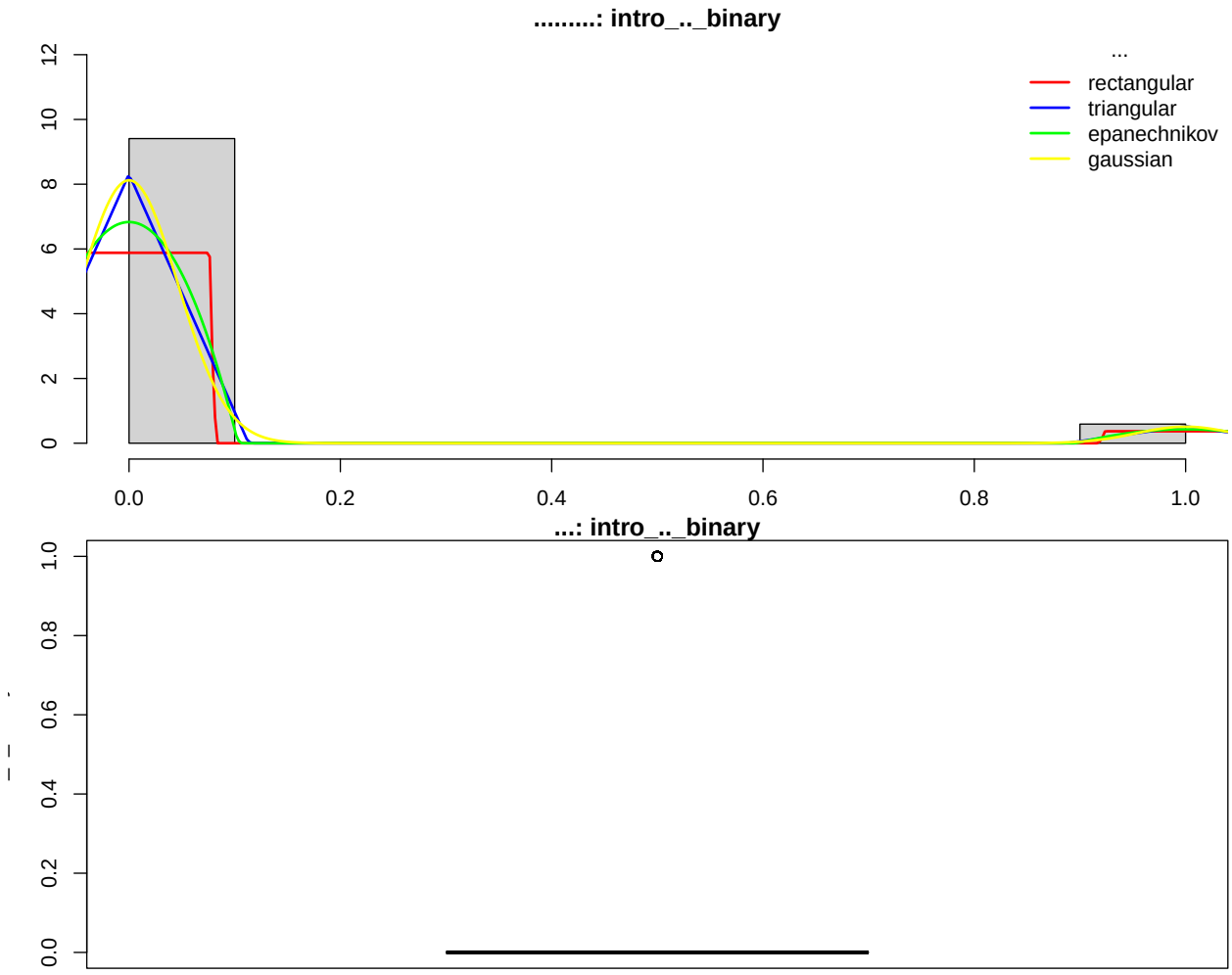

.....: intro_quot_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_quot_binary
0.0
0.2
0.4
0.6
0.8
1.0

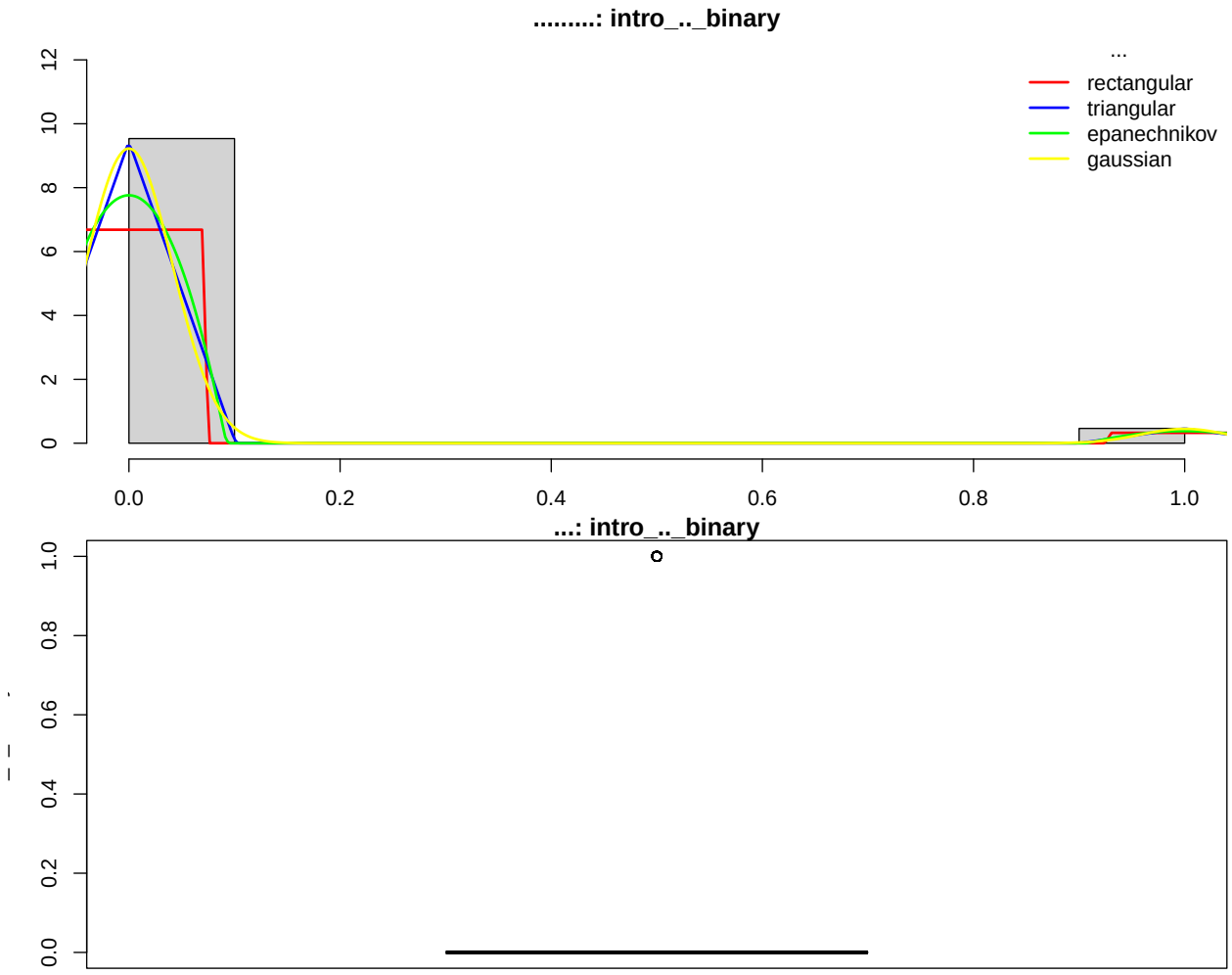

.....: intro_amp_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_amp_binary
0.0
0.2
0.4
0.6
0.8
1.0

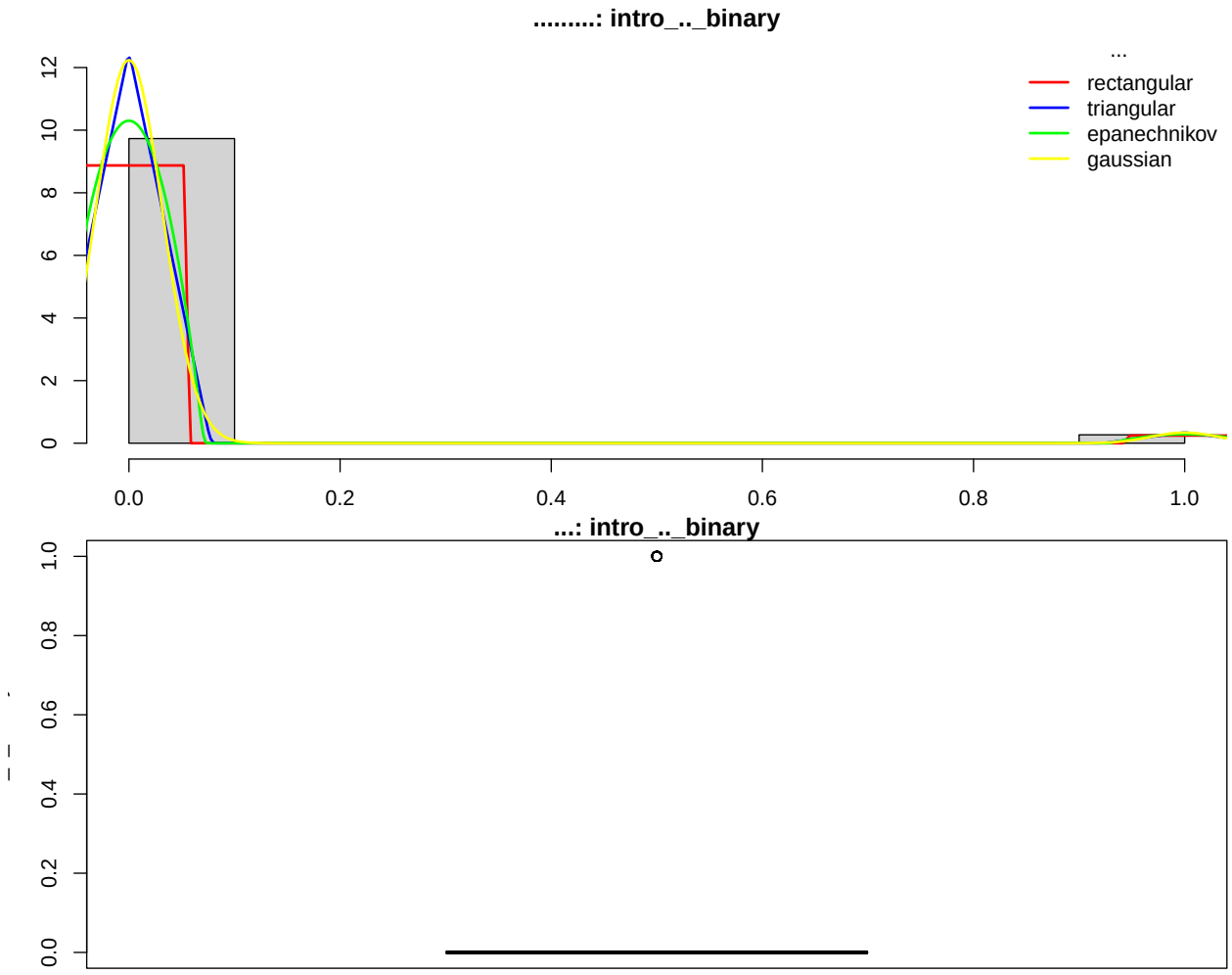

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

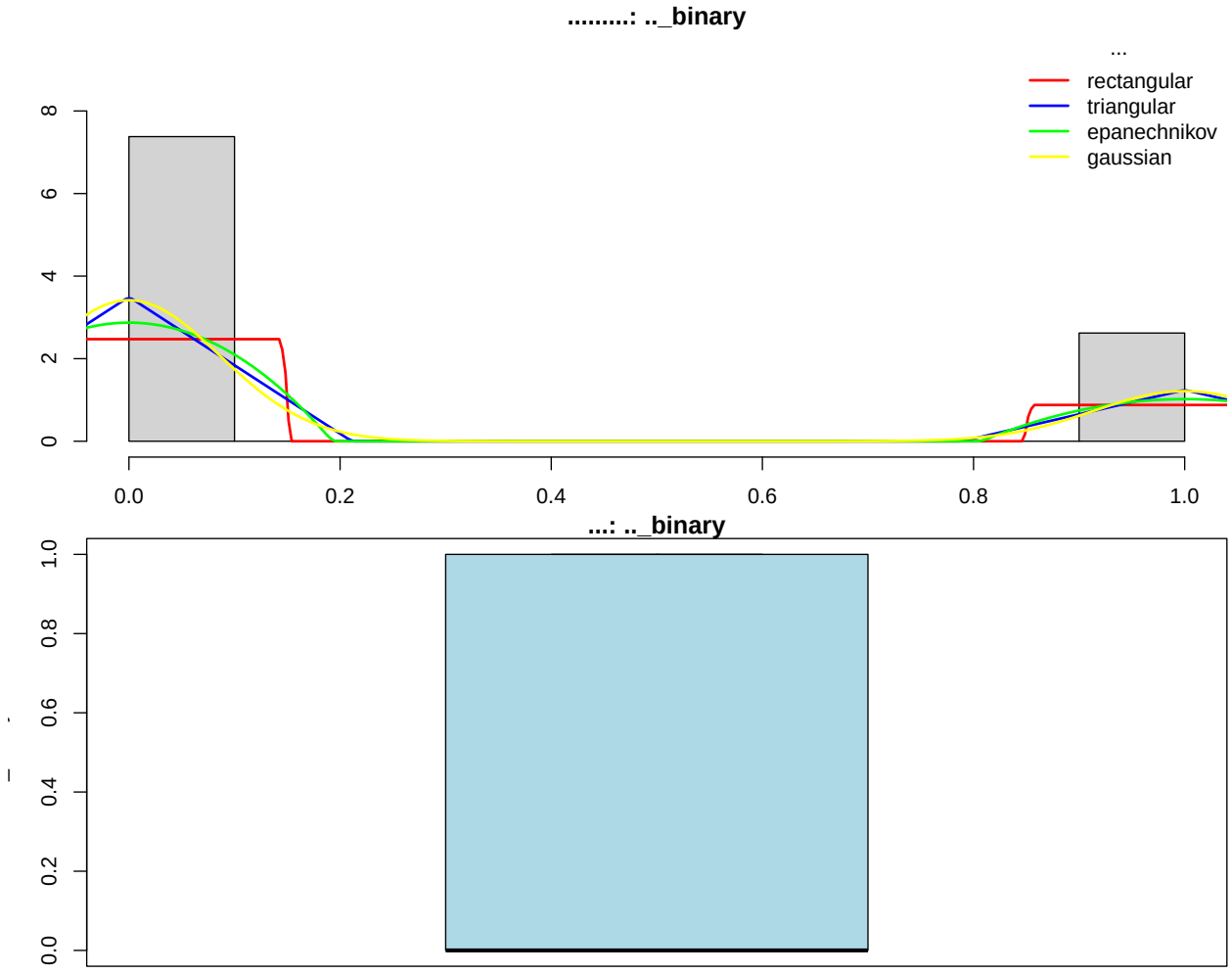

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

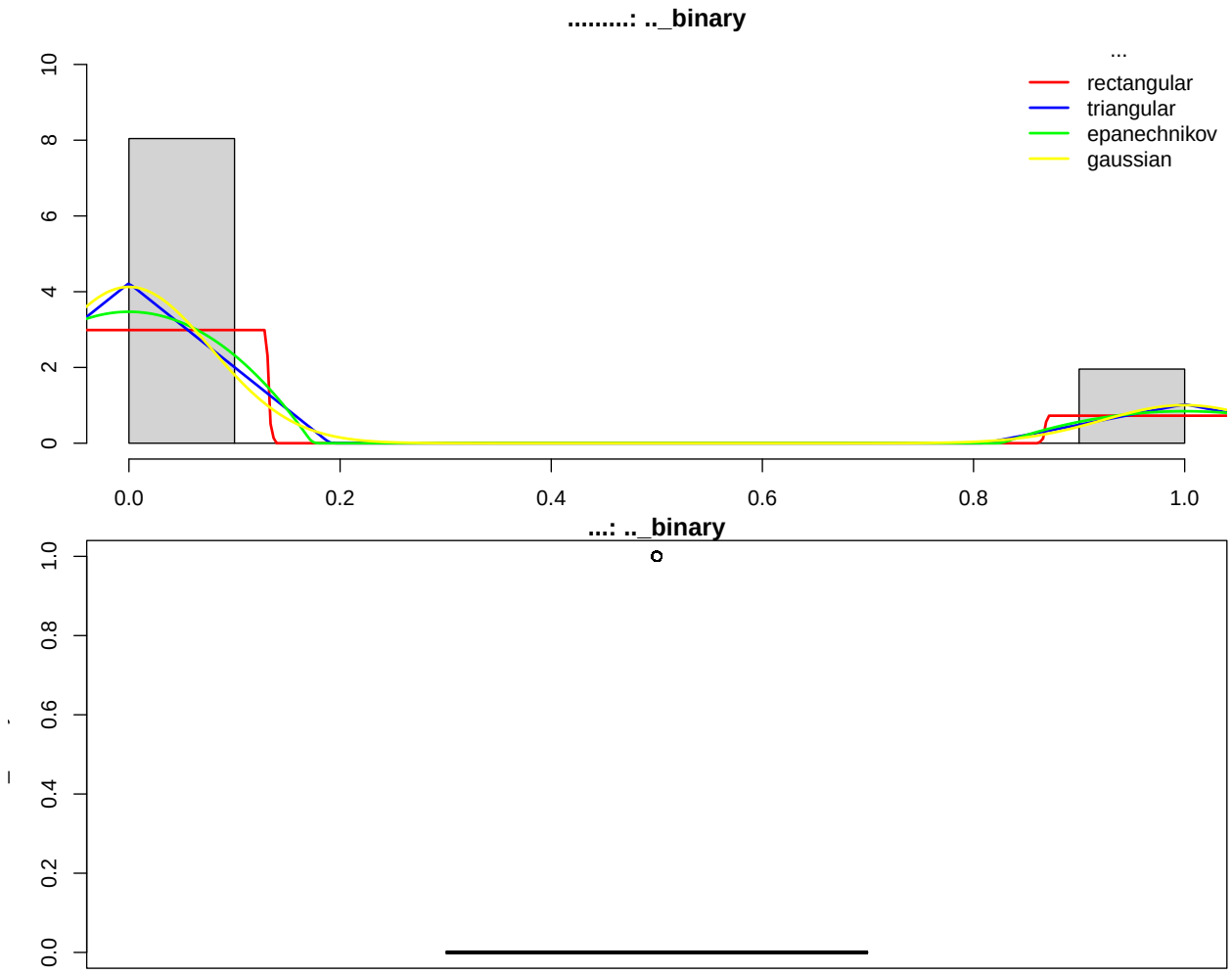

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

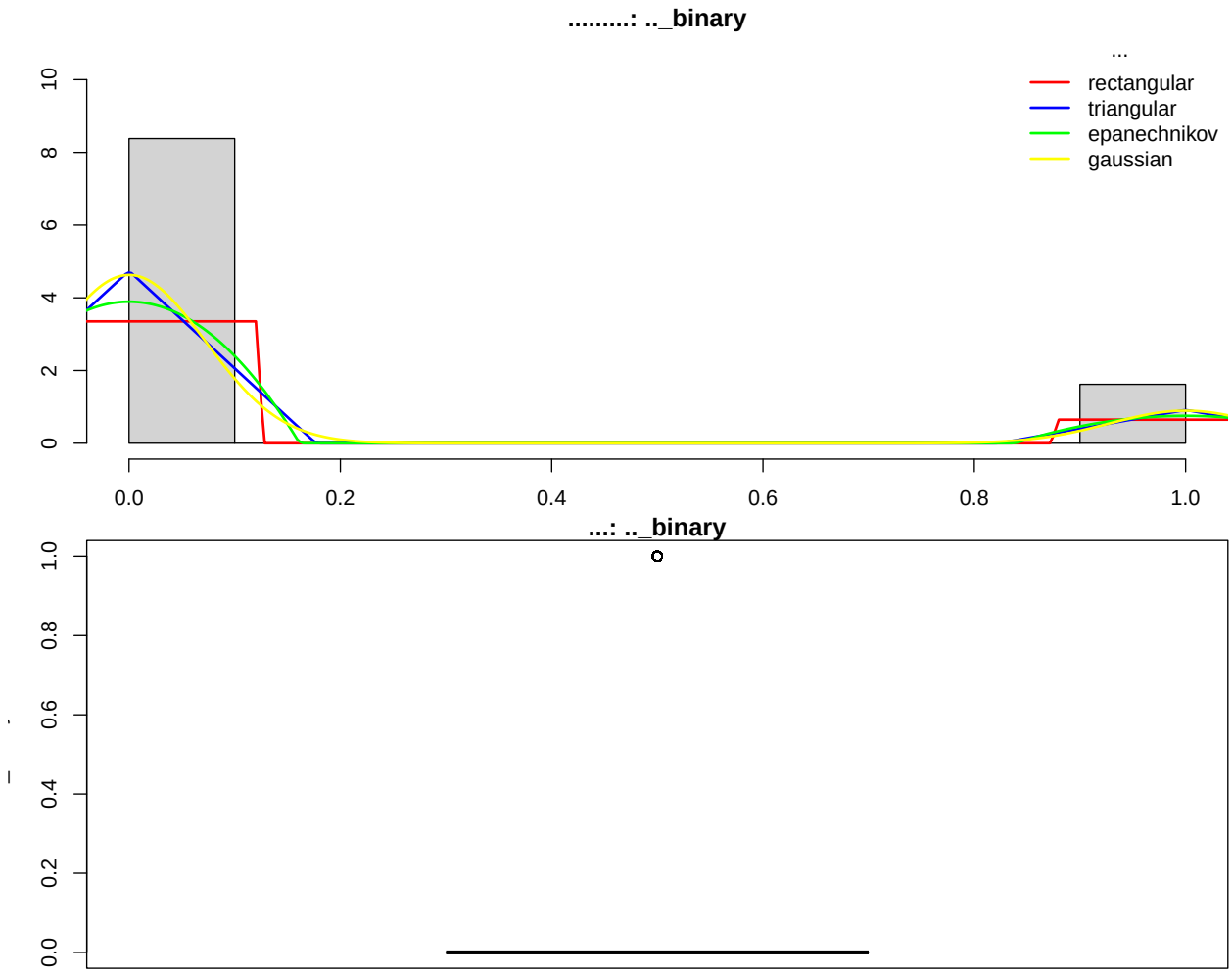

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

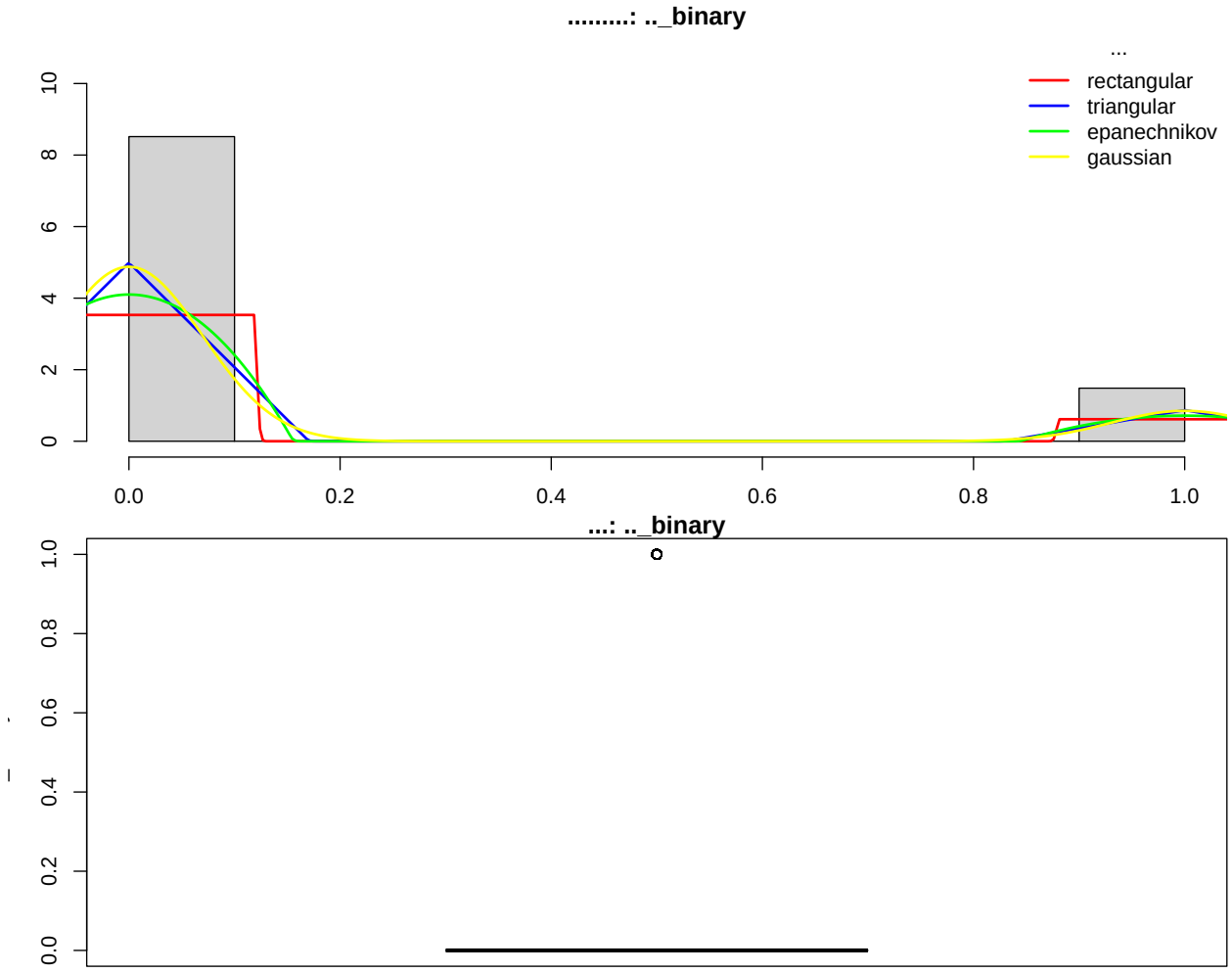

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

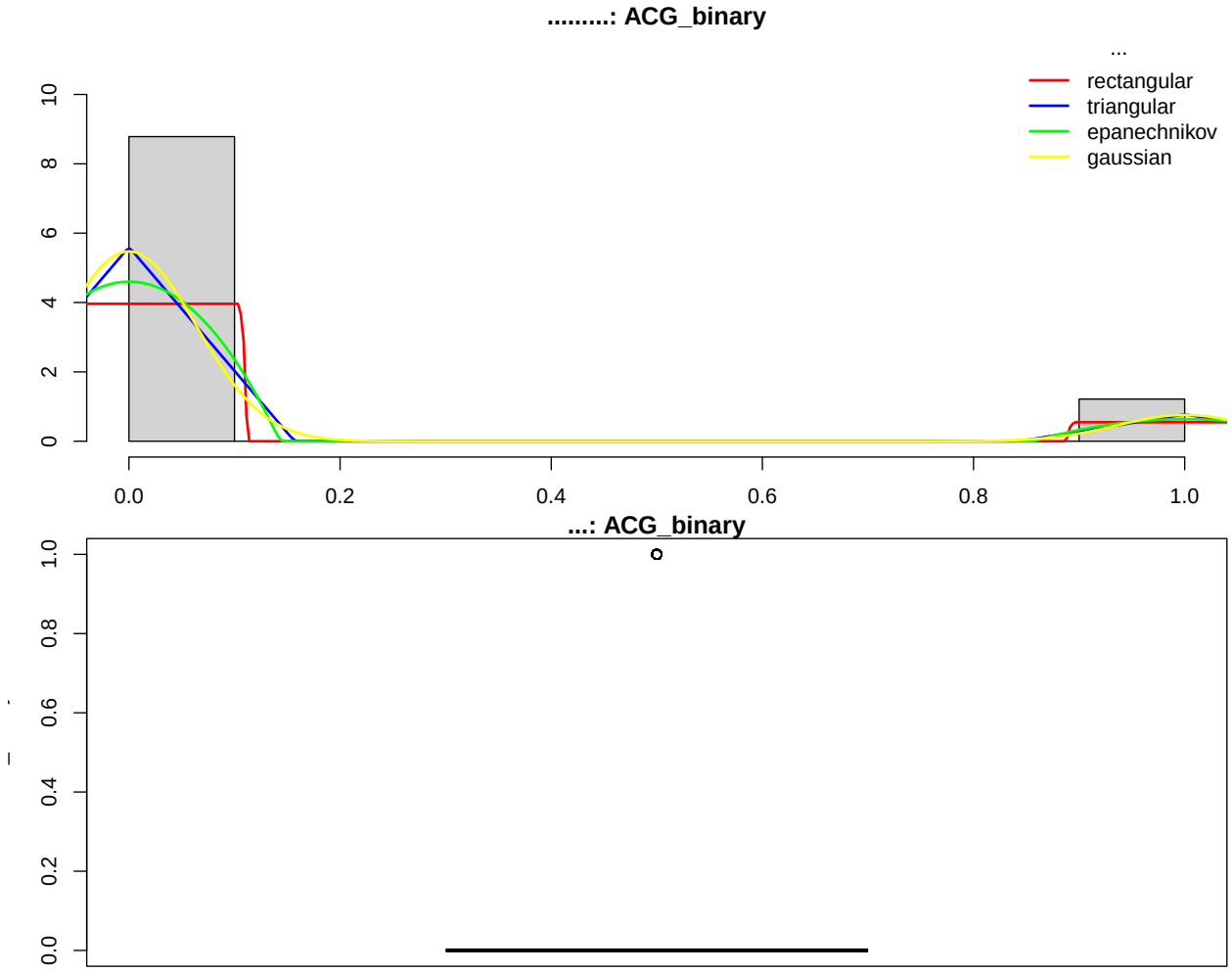

.....: intro_.._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: intro_.._binary
0.0
0.2
0.4
0.6
0.8
1.0

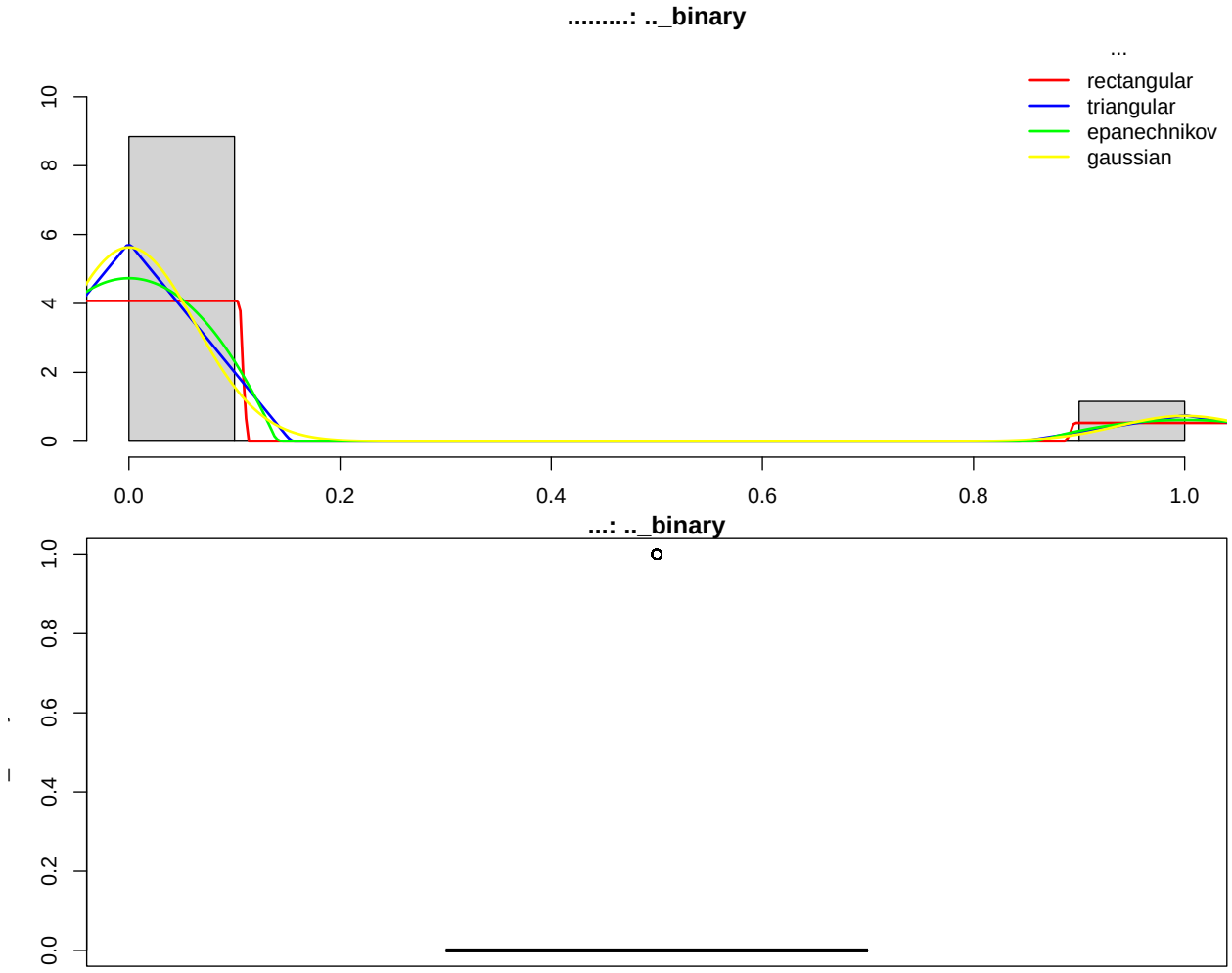

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

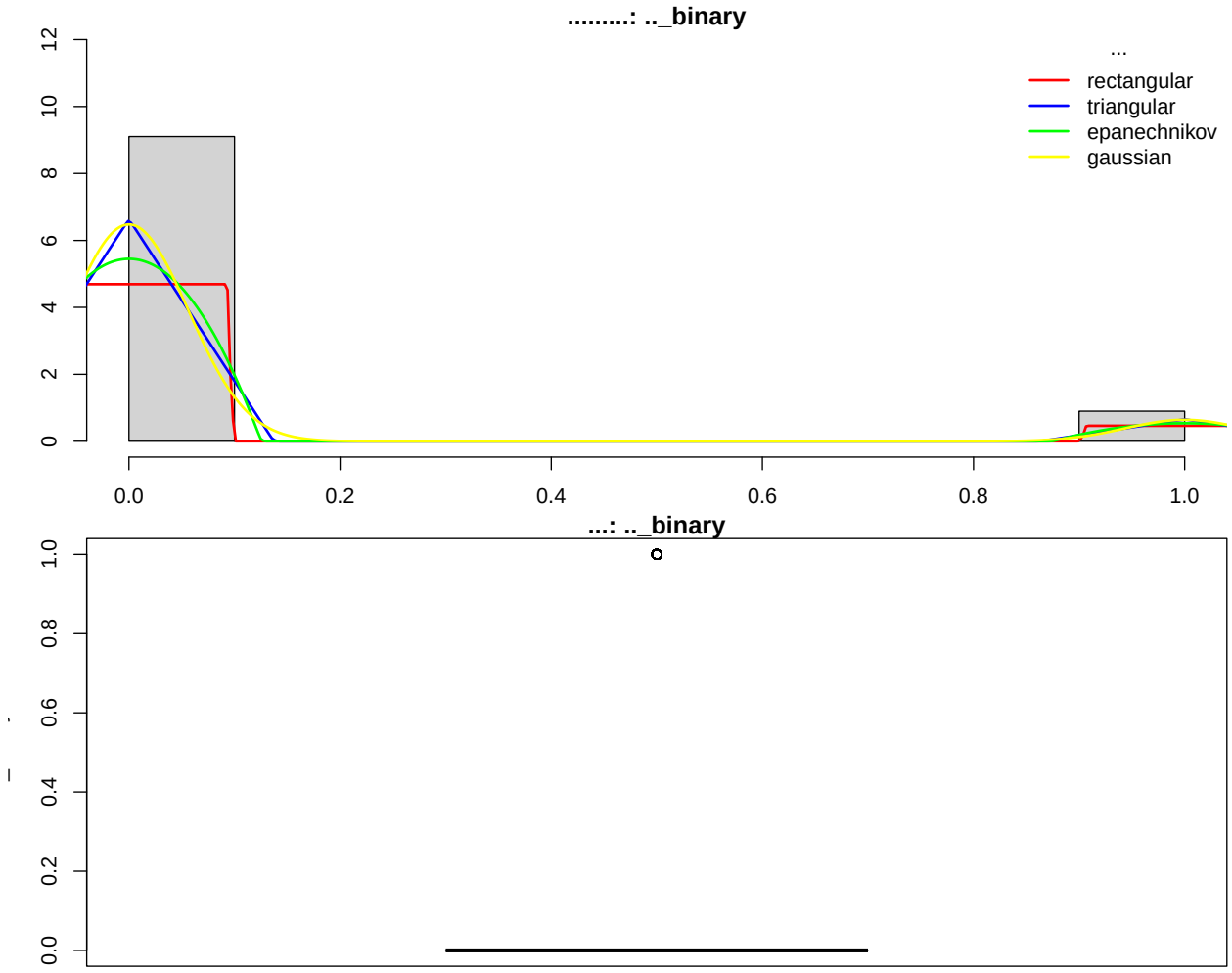

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

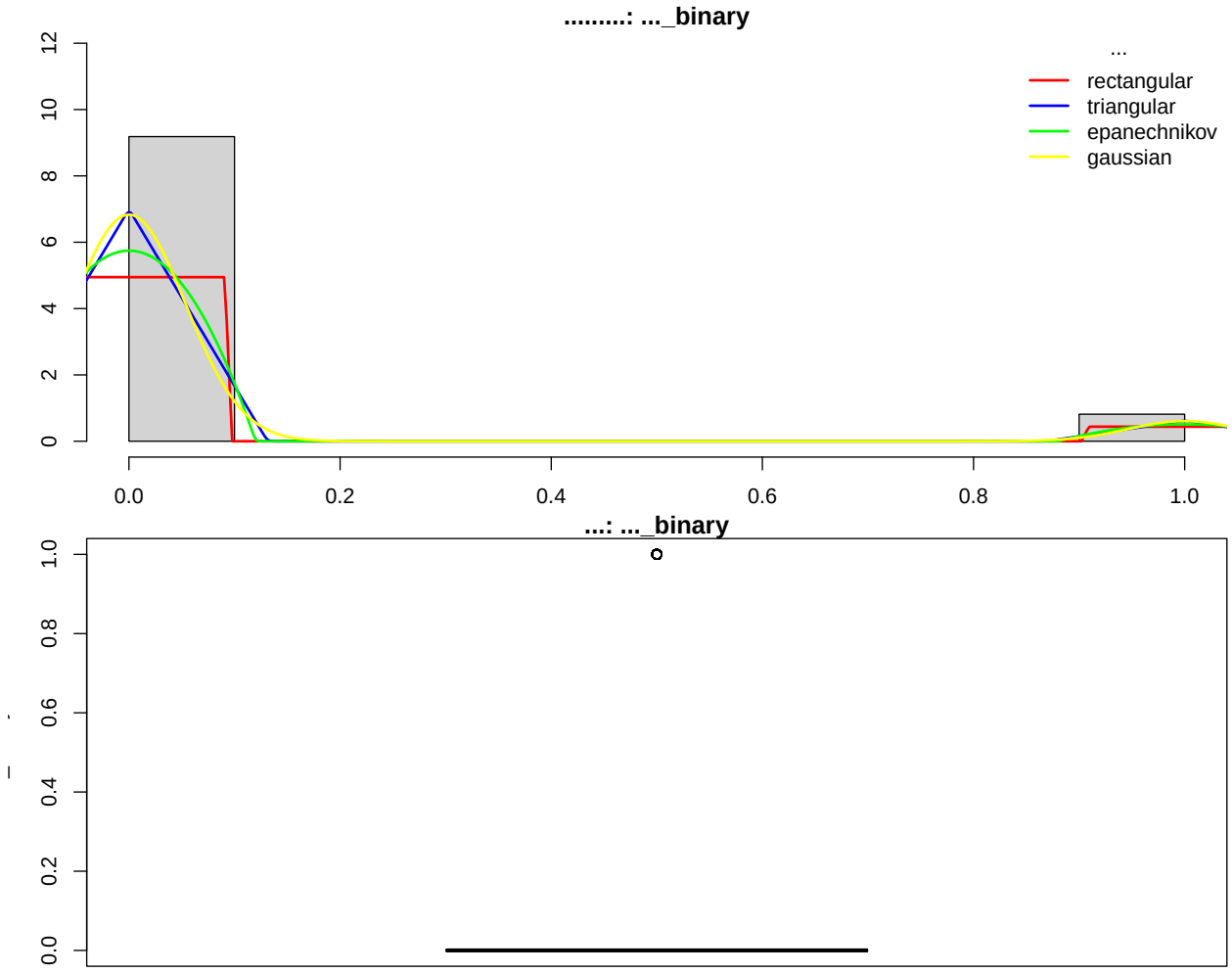

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

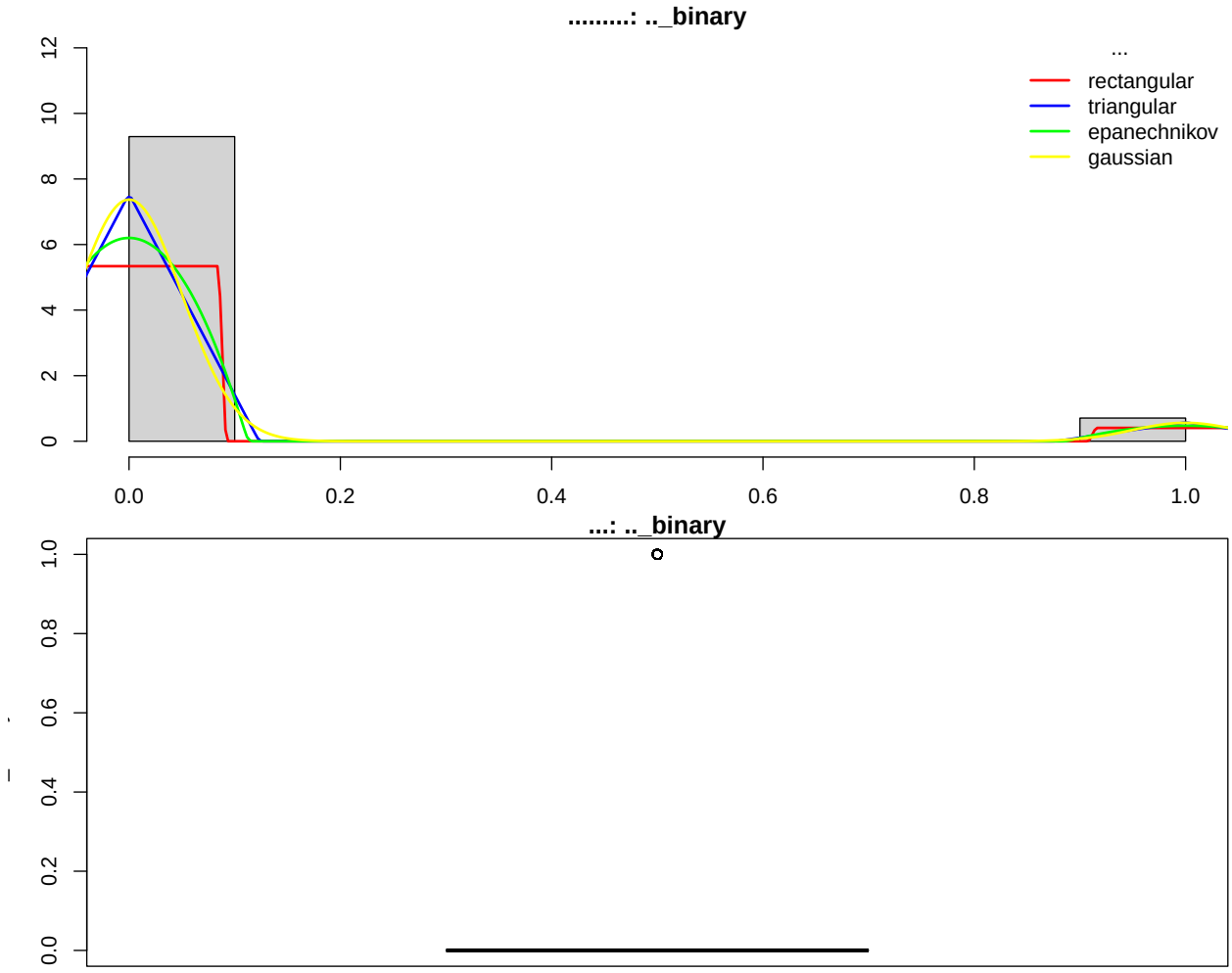

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

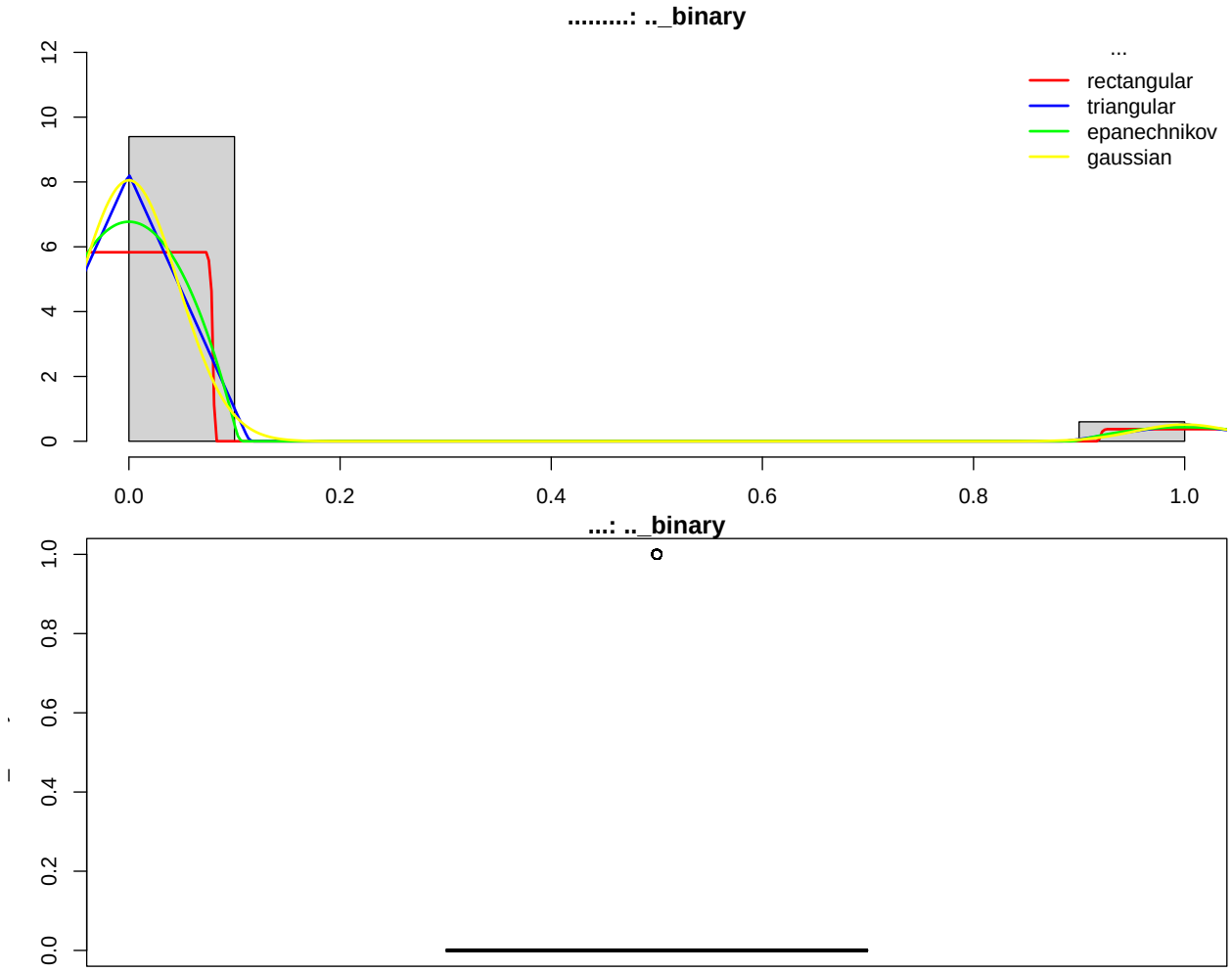

.....: ACG_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: ACG_binary
0.0
0.2
0.4
0.6
0.8
1.0

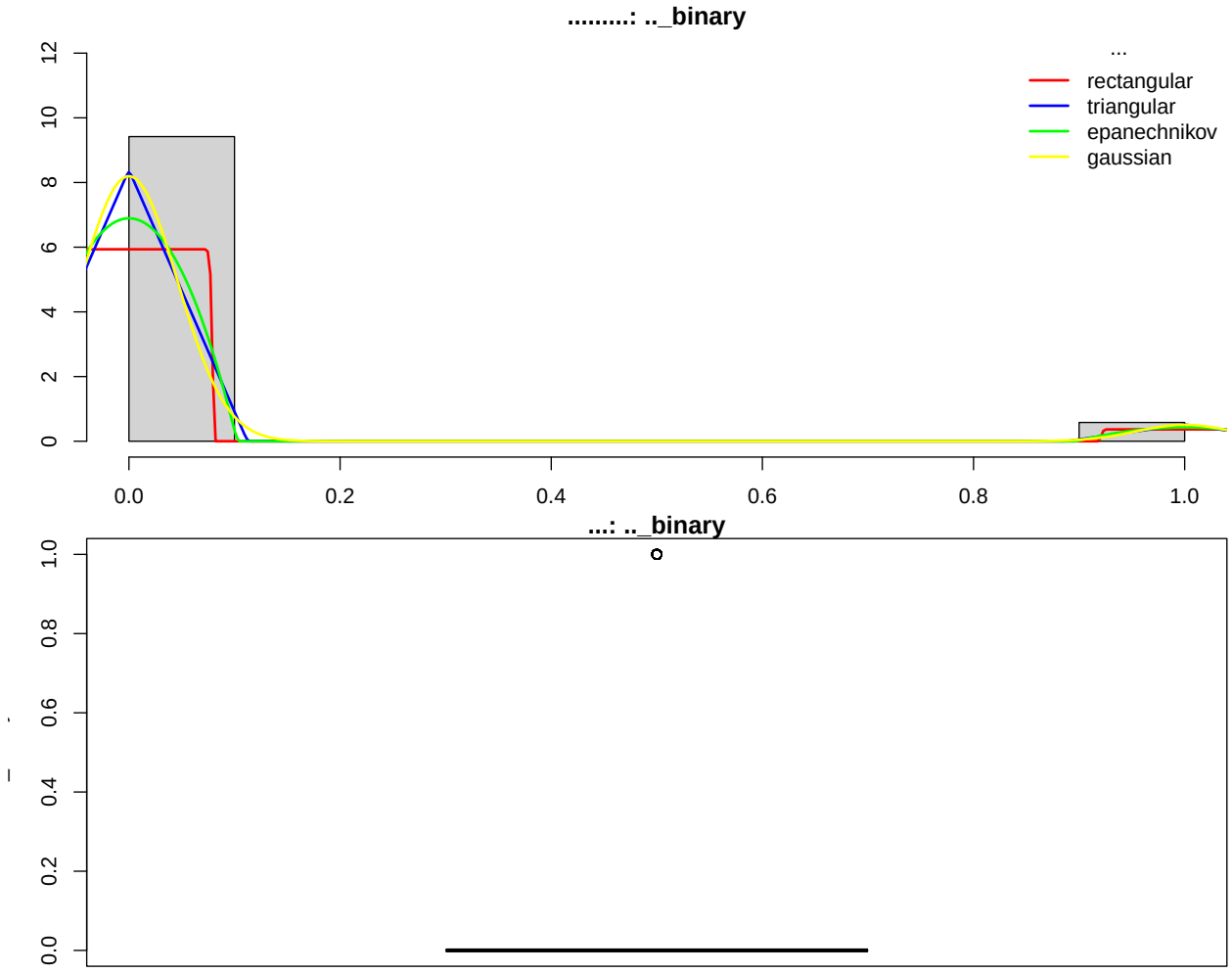

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

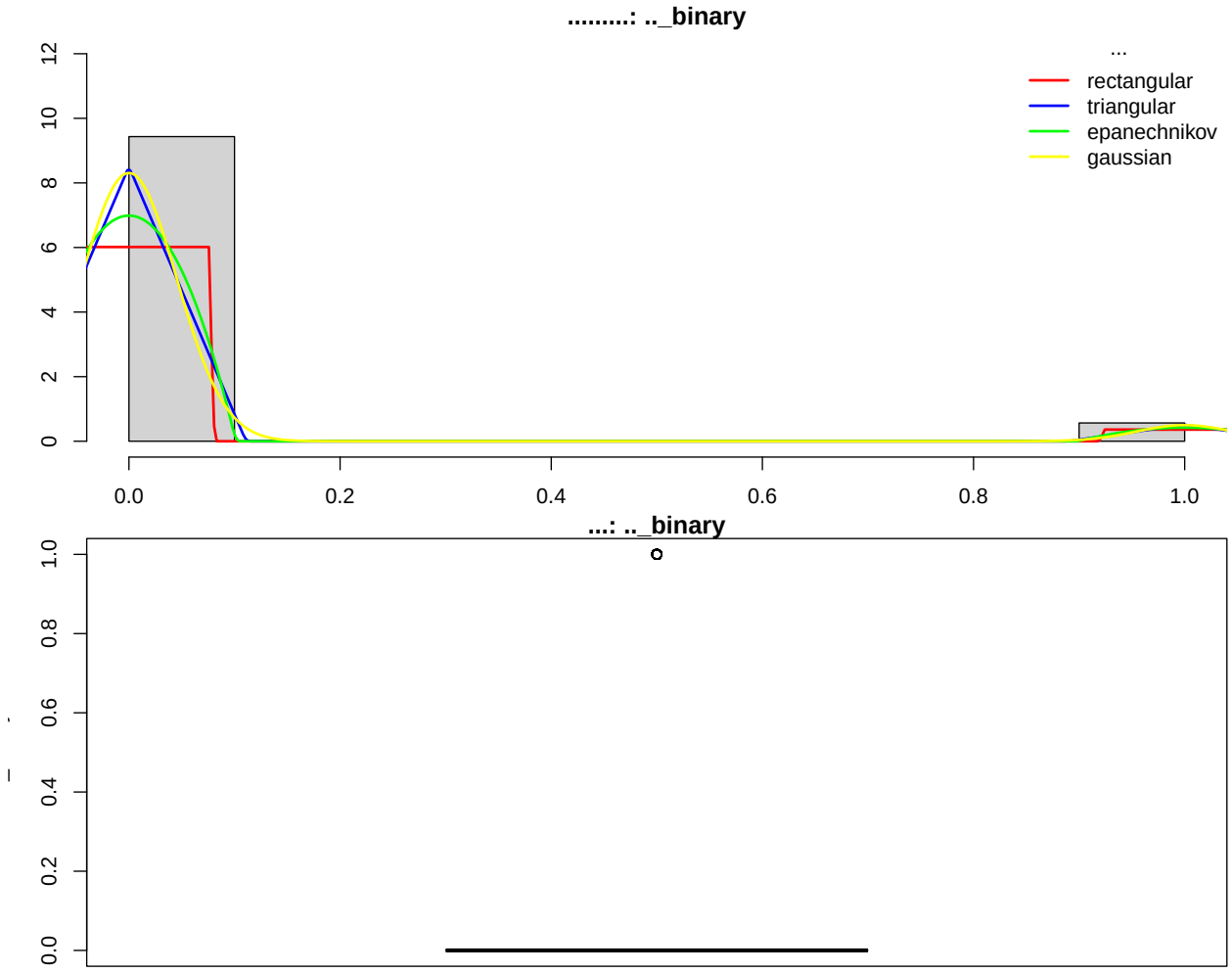

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

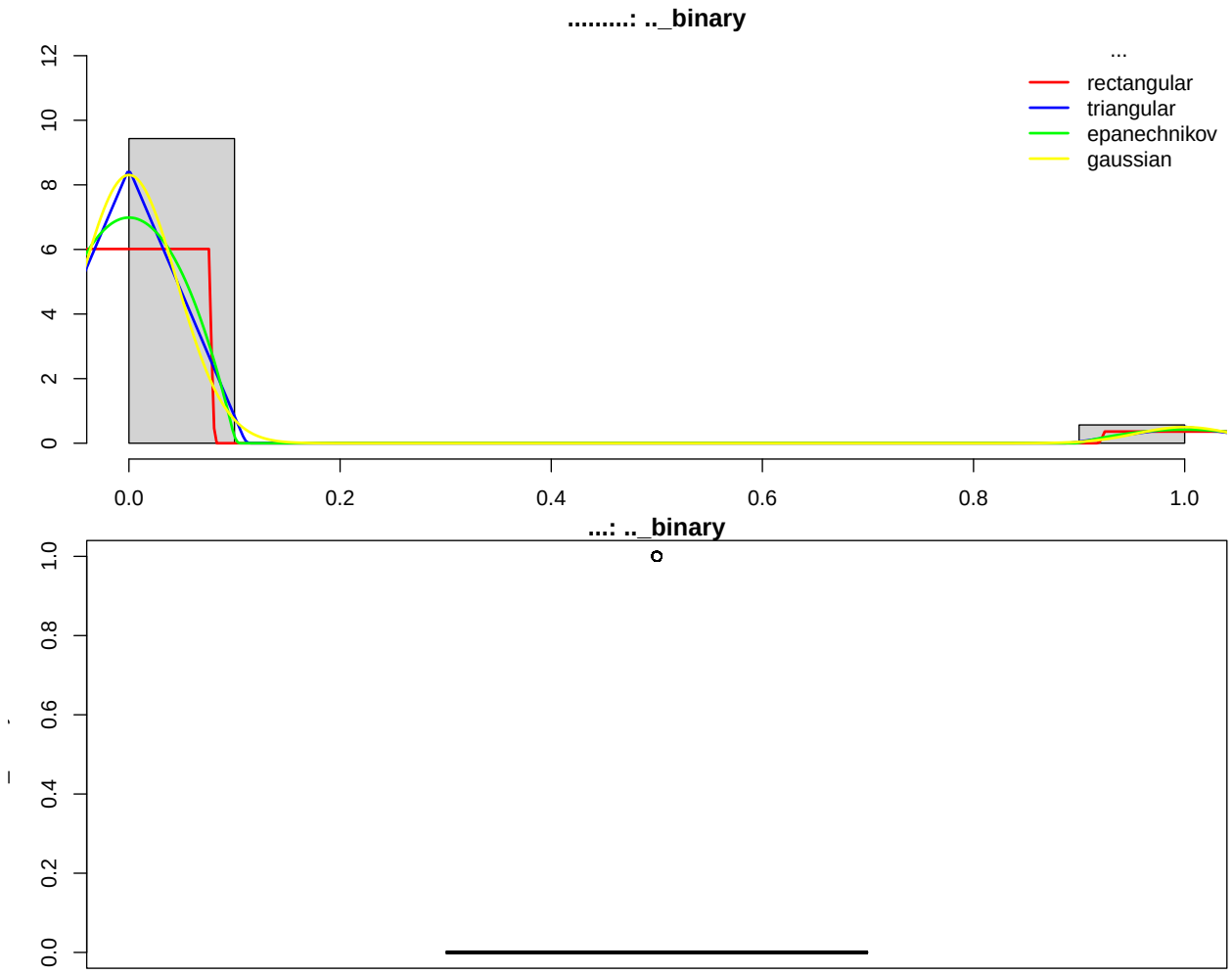

.....: ..._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: ..._binary
0.0
0.2
0.4
0.6
0.8
1.0

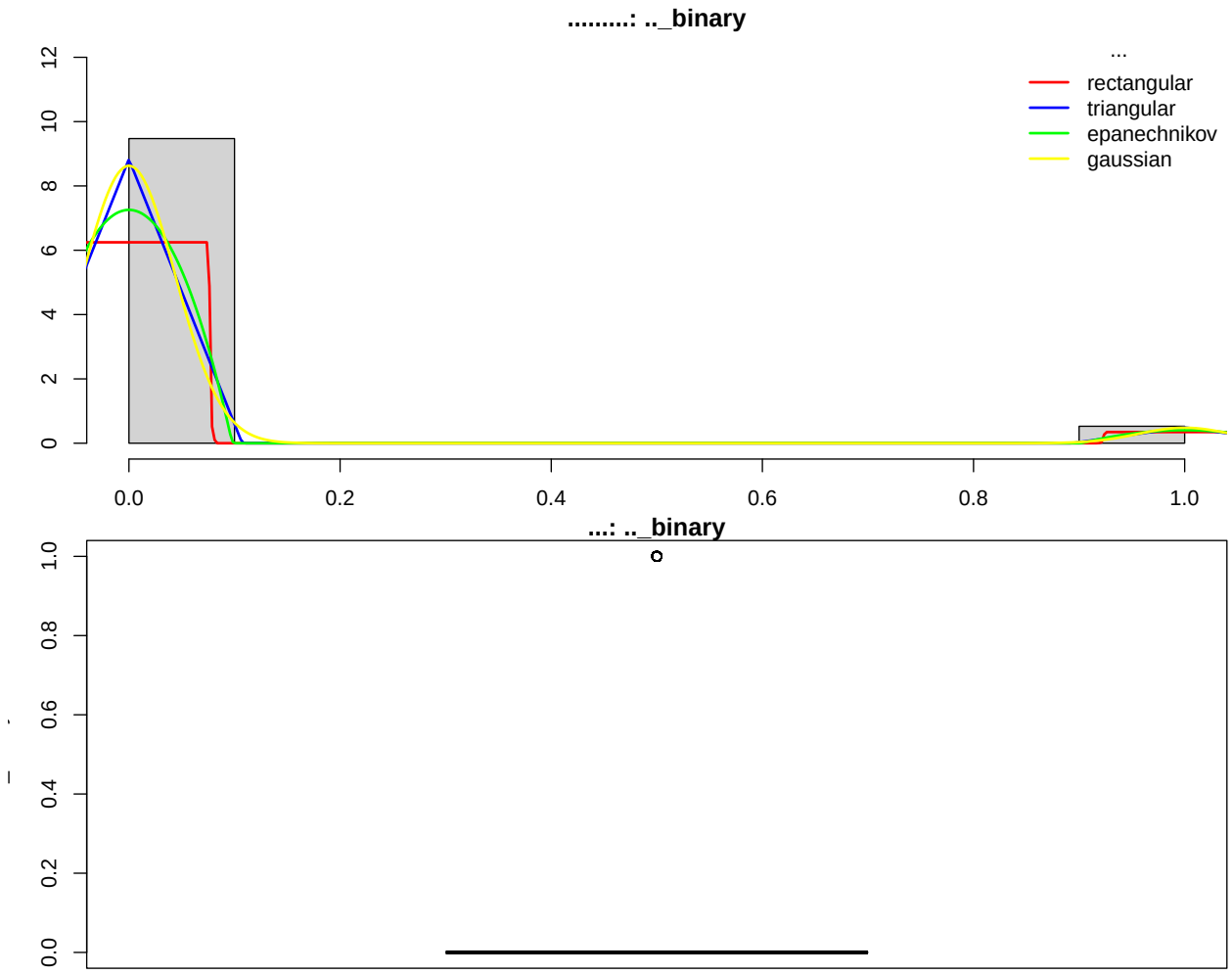

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

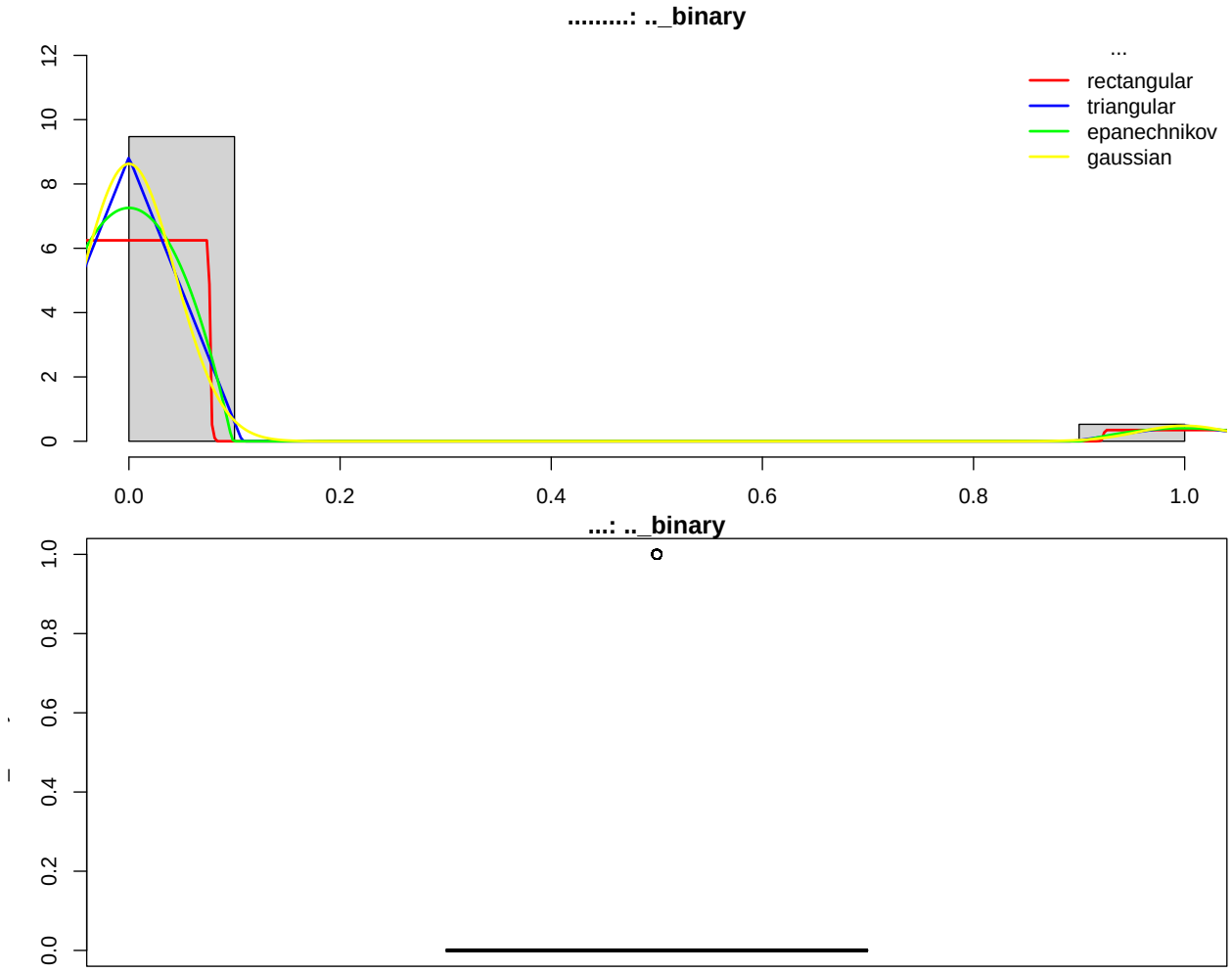

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

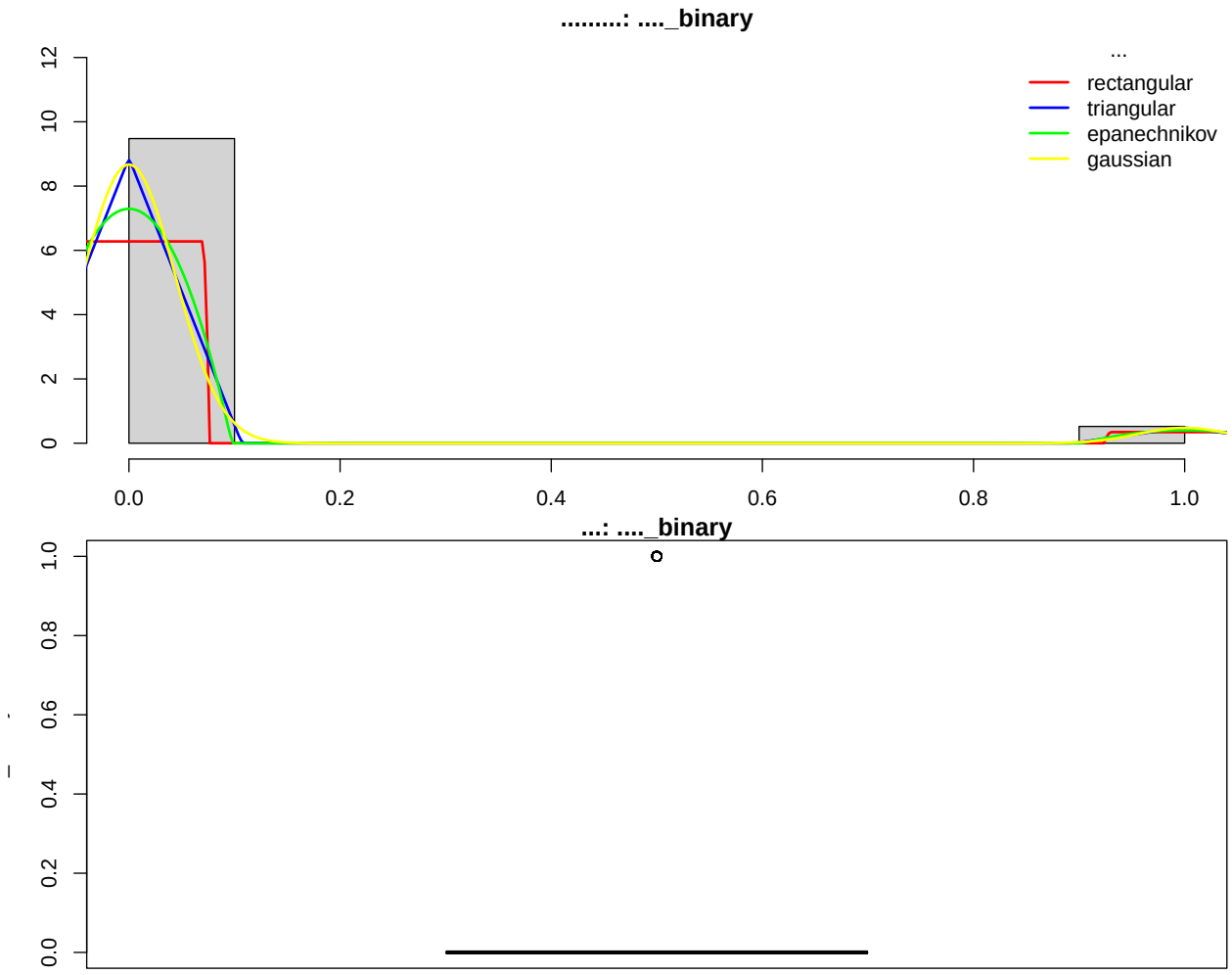

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

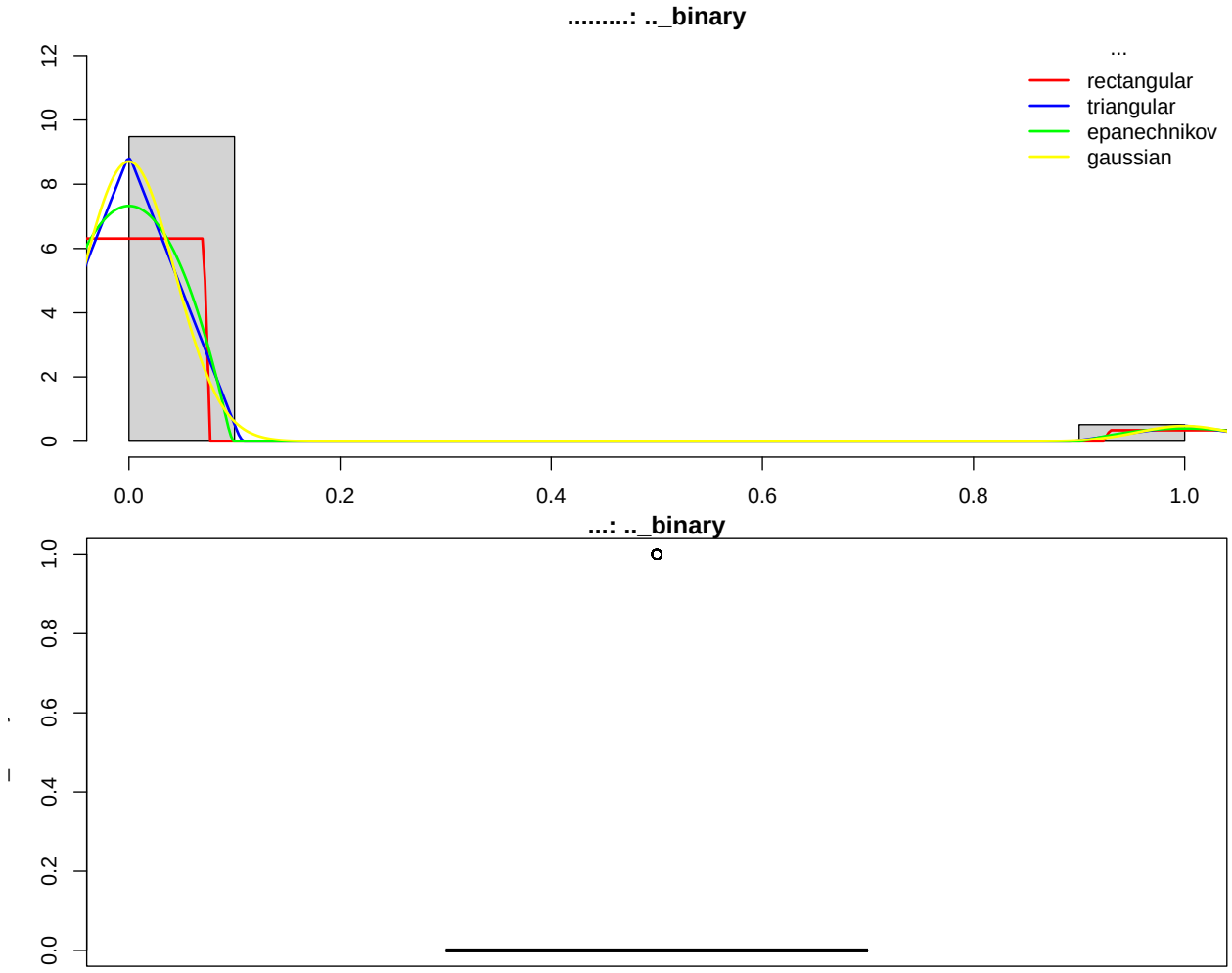

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

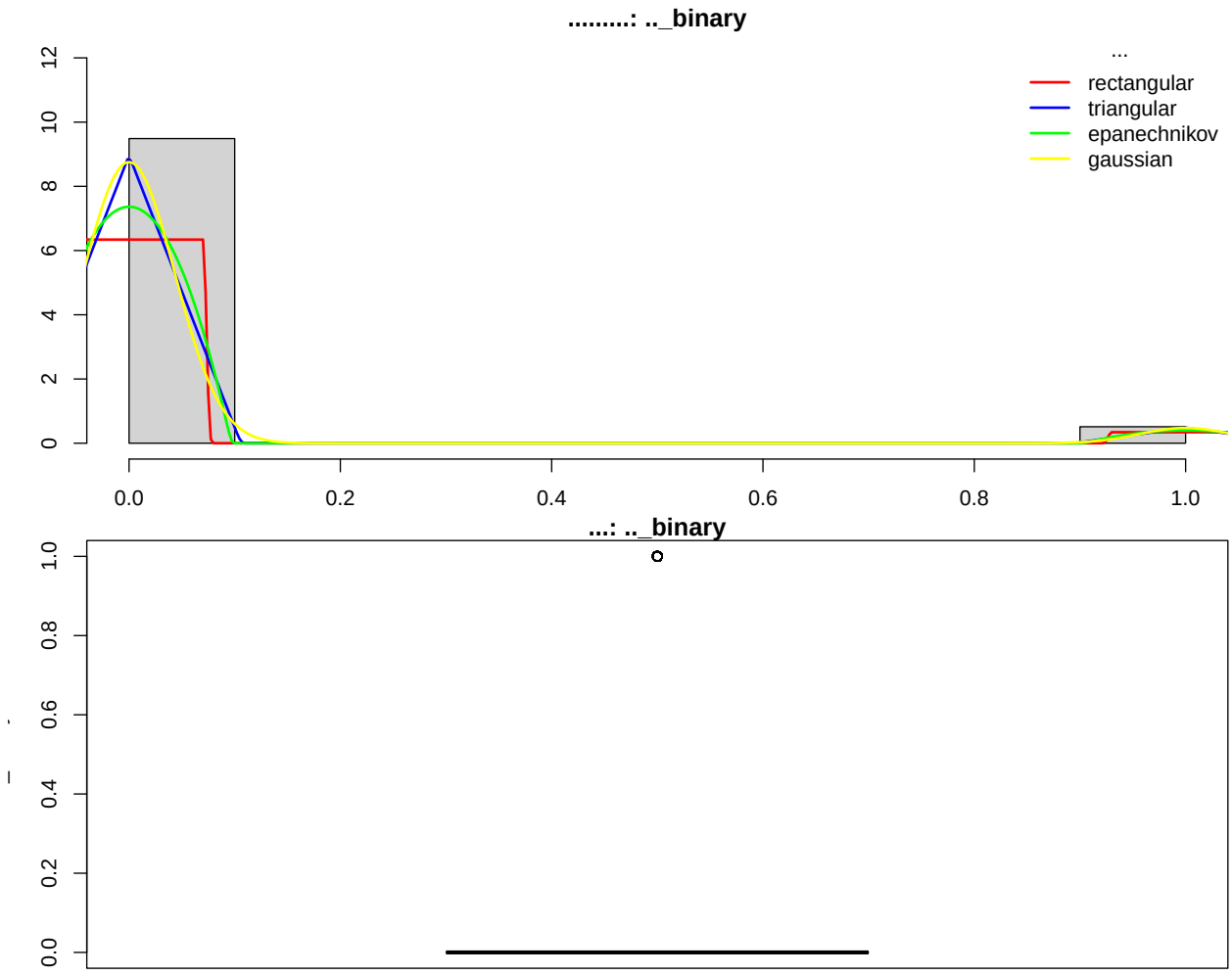

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

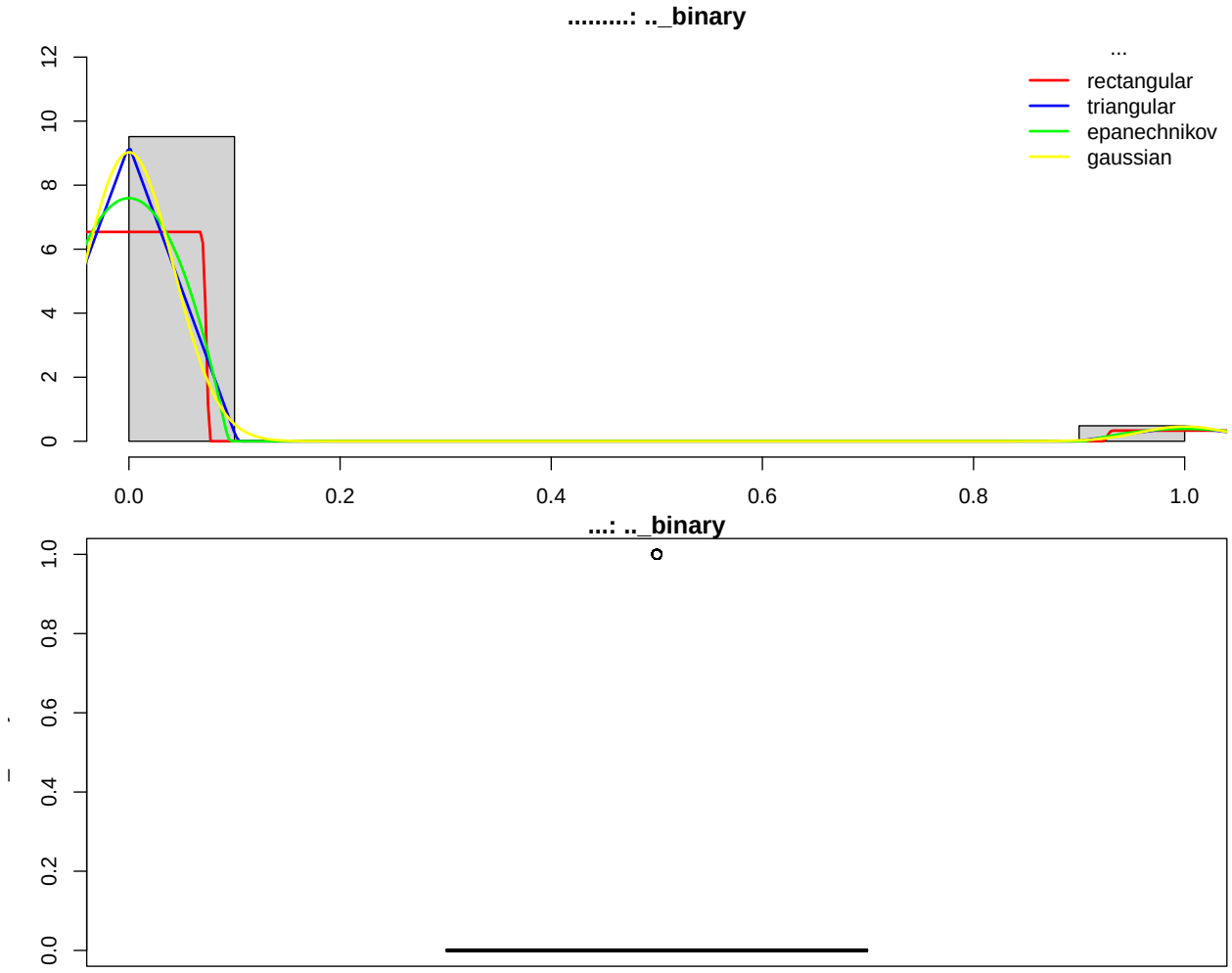

.........: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

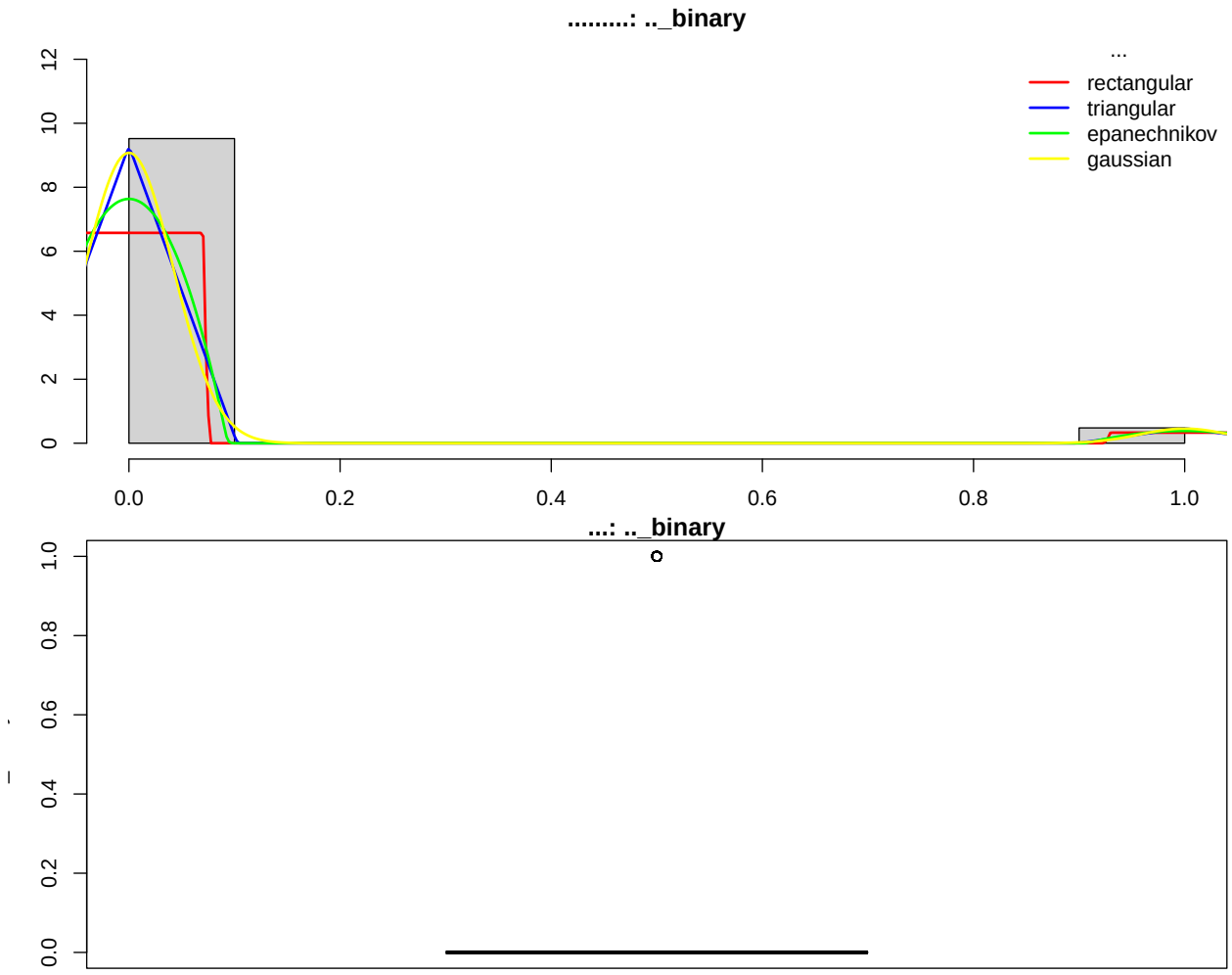

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

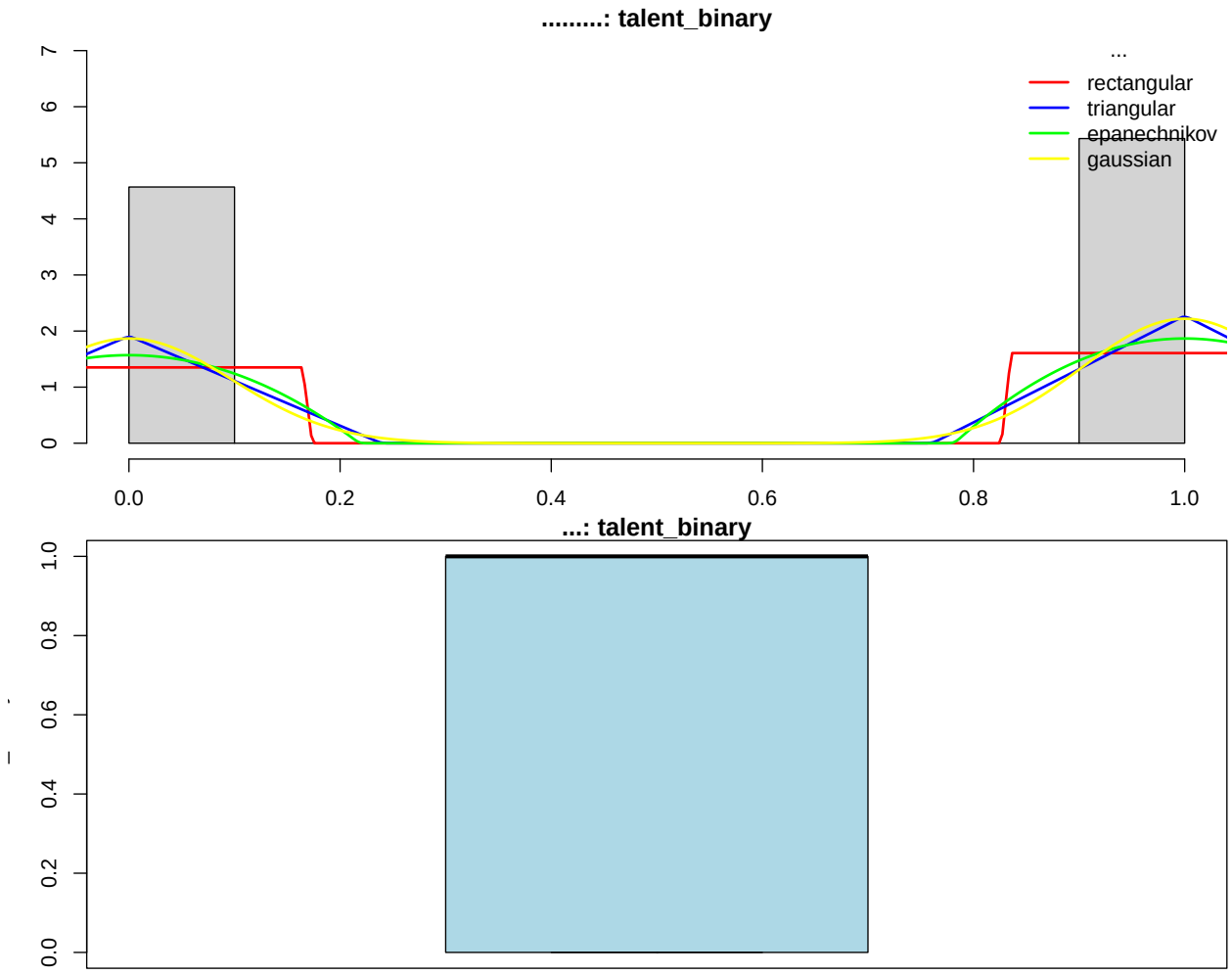

.....: ...._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: ...._binary
0.0
0.2
0.4
0.6
0.8
1.0

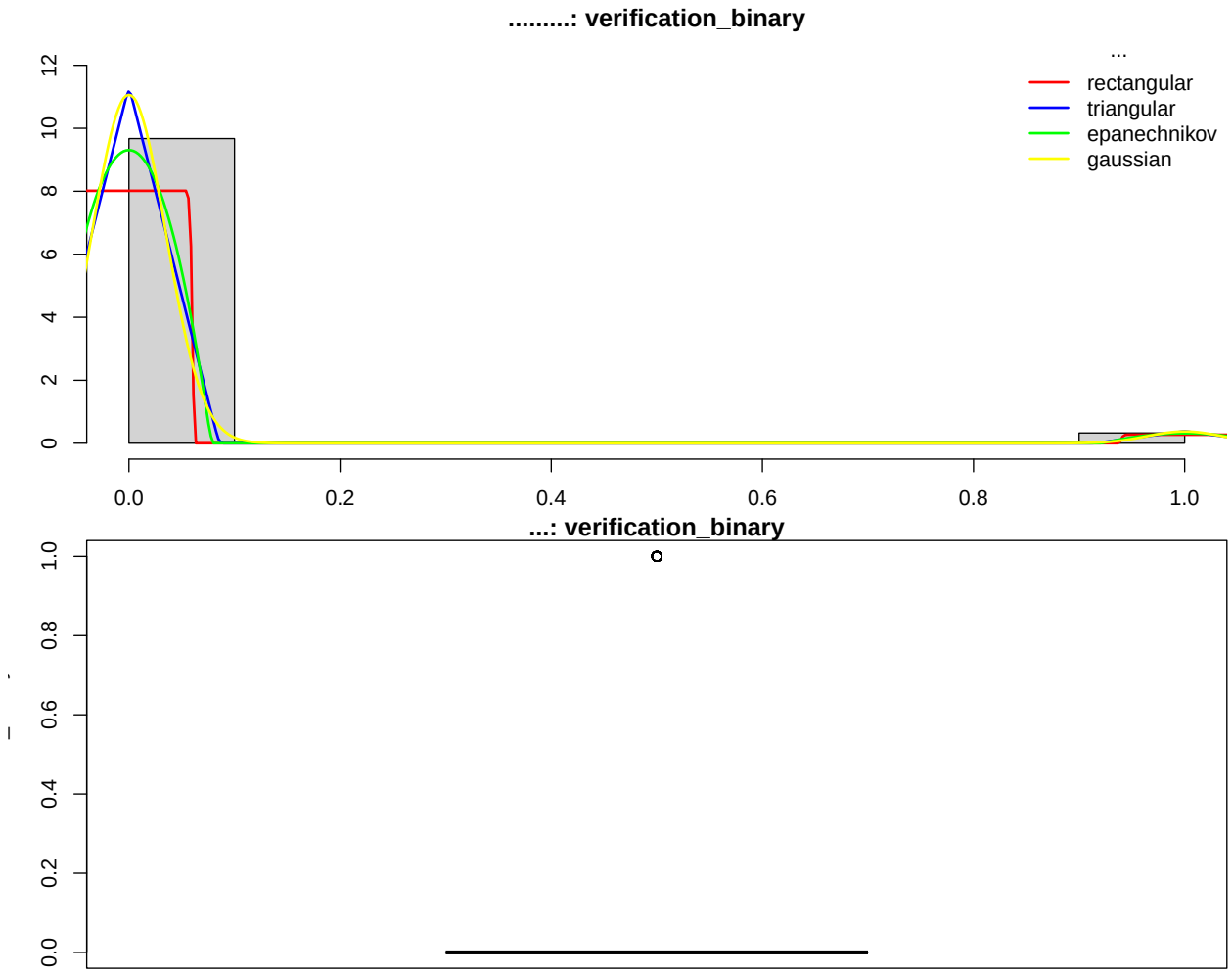

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0

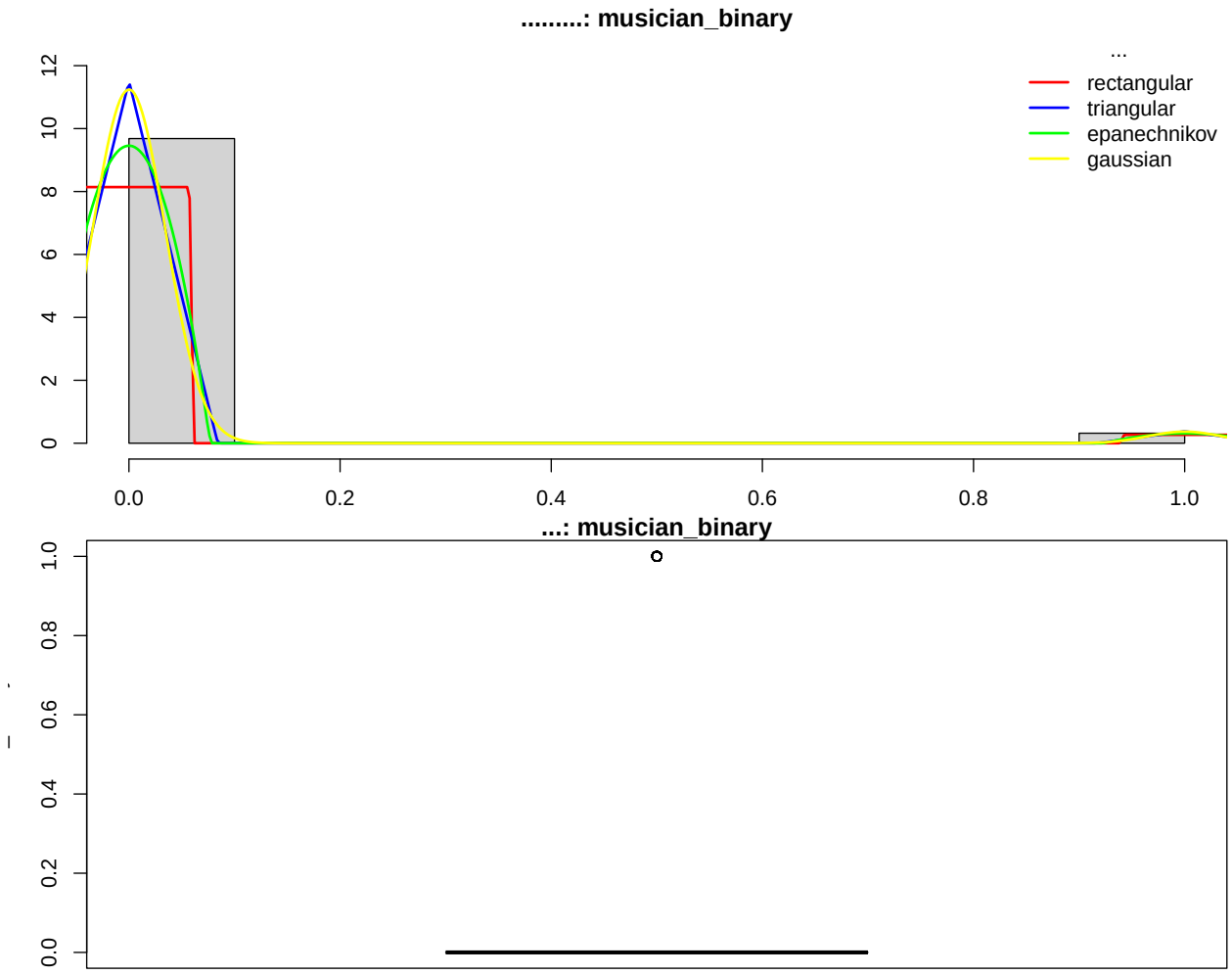

.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0


.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0


.....: .._binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: .._binary
0.0
0.2
0.4
0.6
0.8
1.0


.....: talent_binary
...
rectangular
triangular
epanechnikov
gaussian
0
1
2
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0
...: talent_binary
0.0
0.2
0.4
0.6
0.8
1.0


.....: verification_binary
...
rectangular
triangular
epanechnikov
gaussian
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0
...: verification_binary
0.0
0.2
0.4
0.6
0.8
1.0




```r
par(mfrow = c(1, 1))
```

```r
### factor
#   factor_vars
if(length(factor_vars) > 0){
  par(mfrow = c(1, 3), mar = c(4, 4, 3, 2))

  for (var in factor_vars) {
    freq_data <- table(data_cleaned[[var]])
    prop_data <- prop.table(freq_data)

    categories <- names(freq_data)
    colors <- c("#1f77b4", "#ff7f0e")  #

    barplot(freq_data,
            main = paste("  :", var),
            xlab = var,
            ylab = " ",
            col = colors[1:length(categories)],
            border = "black",
            ylim = c(0, max(freq_data) * 1.2))

    text(x = seq_along(freq_data),
```

```
        y = freq_data,
        label = freq_data,
        pos = 3,  #
        cex = 0.8,
        col = "black")

    barplot(prop_data * 100,
            main = paste("   :", var),
            xlab = var,
            ylab = "   (%)",
            col = colors[1:length(categories)],
            border = "black",
            ylim = c(0, 100))

    text(x = seq_along(prop_data),
         y = prop_data * 100,
         label = paste0(round(prop_data * 100, 1), "%"),
         pos = 3,
         cex = 0.8,
         col = "black")

    pie(freq_data,
        main = paste(" :", var),
        col = colors[1:length(categories)],
        labels = paste0(categories, "\n",
                        freq_data, " (",
                        round(prop_data * 100, 1), "%)"),
        cex = 0.9)

    legend("topright",
           legend = categories,
           fill = colors[1:length(categories)],
           title = paste(" :", var))
  }
  par(mfrow = c(1, 1))
}
```

```
###
windowsFonts(SimHei = windowsFont("SimHei"))

###
stopwords_custom <- c("br", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " ")

###
text_clean <- function (texts) {
  #   NA
  texts[is.na(texts)] <- ""
  all_text <- paste(texts, collapse = " ")
  all_text <- str_replace_all(all_text, "[ ]", " ")
  return(all_text)
}

###
text_analysis <- function(texts) {
```

```
  text_cleaned <- text_clean(texts)

  ###
  cutter <- worker()
  words <- cutter[text_cleaned]
  words_filtered <- words[
    nchar(words) > 1 &
      !words %in% stopwords_custom
  ]
  word_freq <- table(words_filtered) %>%
    as.data.frame() %>%
    arrange(desc(Freq)) %>%
    head(50)
  colnames(word_freq) <- c("Word", "Frequency")

  ###
  tryCatch({
    wordcloud(words = word_freq$Word,
              freq = word_freq$Frequency,
              min.freq = 5,
              max.words = 100,
              random.order = FALSE,
              colors = rainbow(10),
              family = "SimHei")
  }, error = function(e) {message("    :      ")})

  ###
  return(word_freq)
}

#
name_analysis <- text_analysis(data_cleaned$name)
```

```
## Error in worker(): could not find function "worker"
```

```
introduction_analysis <- text_analysis(data_cleaned$introduction)
```

```
## Error in worker(): could not find function "worker"
```

```
###  log
skewness(data_cleaned$play_count)
```

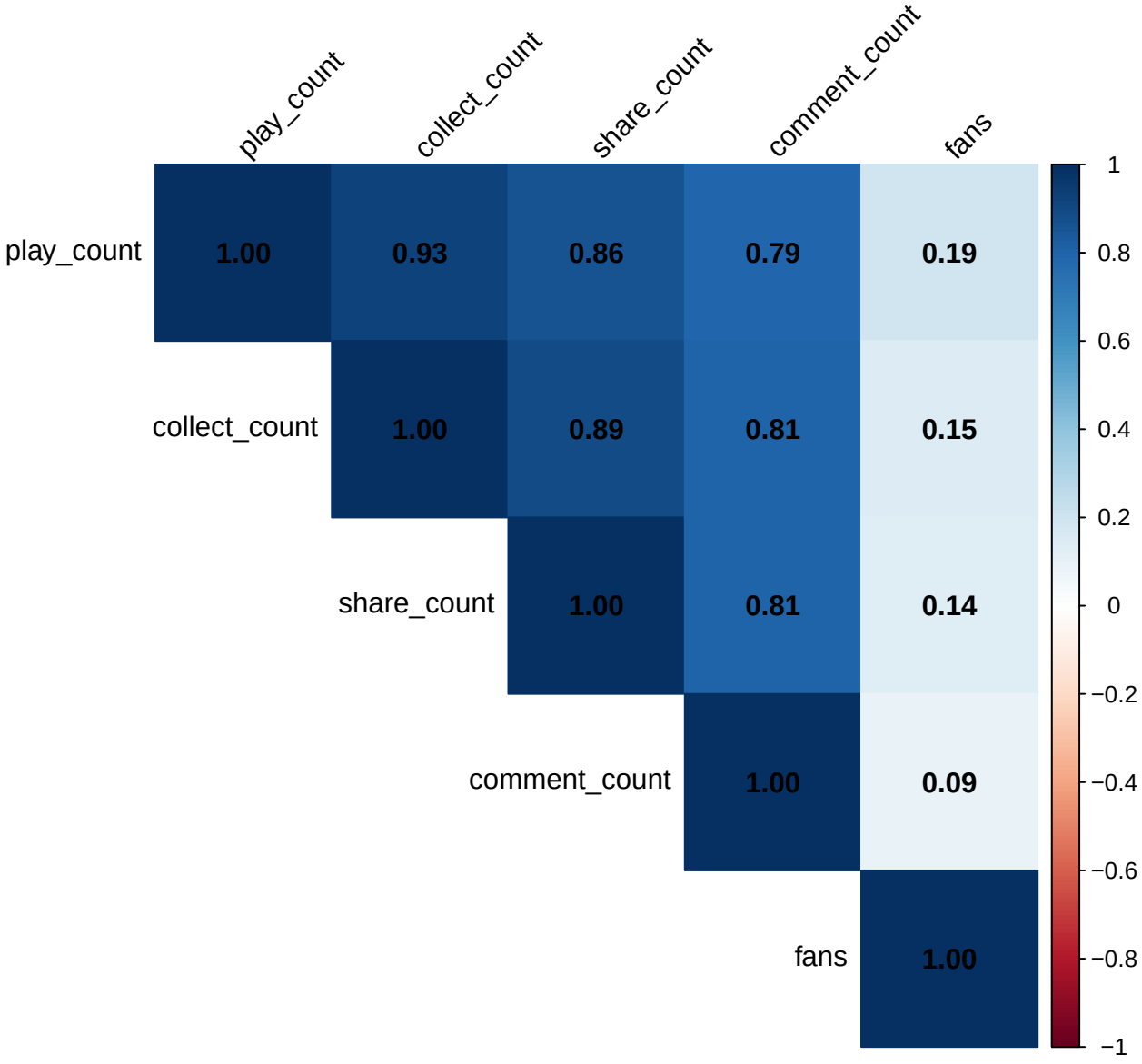
```
## [1] 8.969987
```

4. ( )

Pearson Correlation Matrix play_count

1. collect_count
2. share_count comment_count Multicollinearity
3. fans
4.

```r
# ==== 1                    ====
result_vars <- data_cleaned %>%
  dplyr::select(play_count, collect_count, share_count, comment_count, fans)
#
cor_matrix <- cor(result_vars, use = "complete.obs")
#
corrplot(cor_matrix,
         method = "color",
         type = "upper",
         addCoef.col = "black", #
         tl.col = "black",      #
         tl.srt = 45,           #
         title = "Correlation Matrix",
         mar = c(0,0,1,0))
```

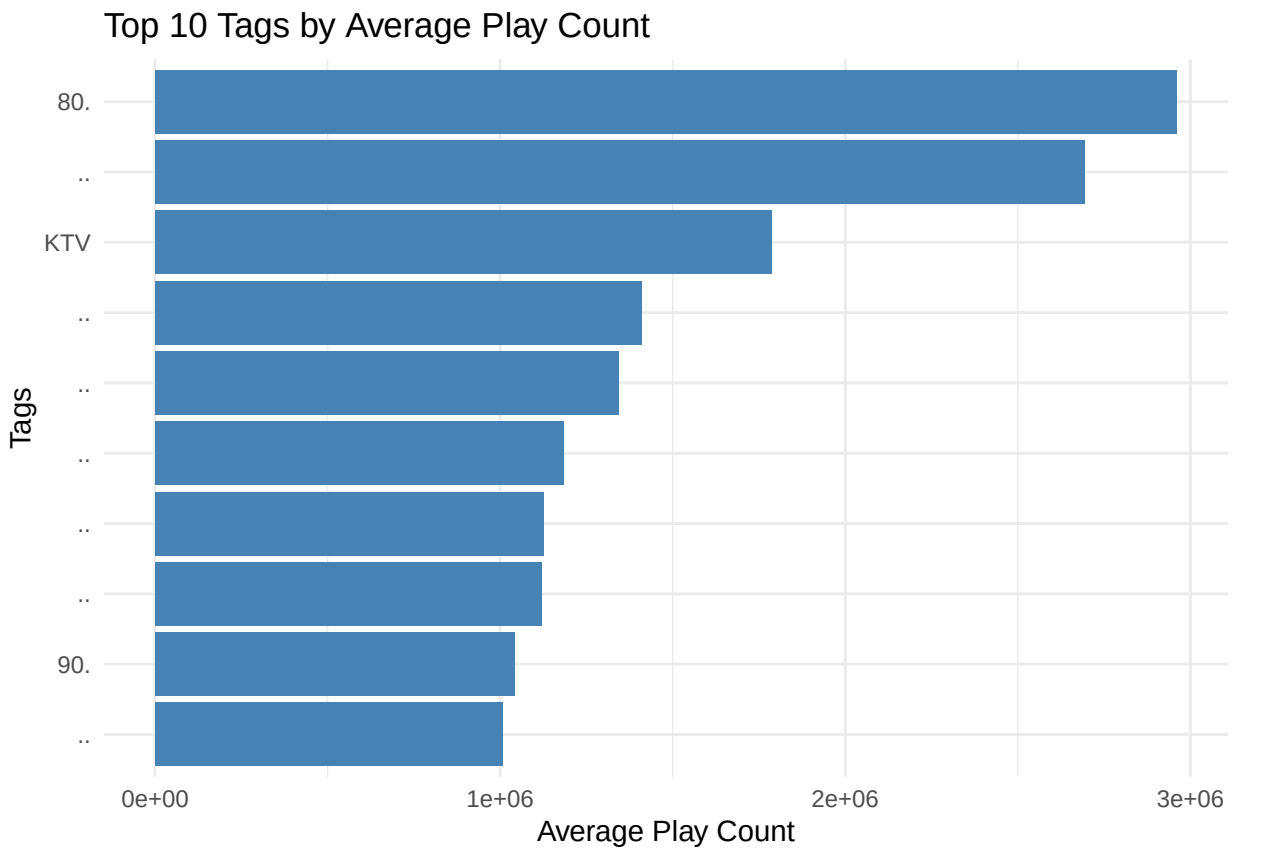

Correlation Matrix
play_count
collect_count
share_count
comment_count
fans
1.00
0.93
0.86
0.79
0.19
1.00
0.89
0.81
0.15
1.00
0.81
0.14
1.00
0.09
1.00
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

5. ( )

1. “ ” “ ” “90 ”
2.
3. “ ” “ ” “ ” “ ”

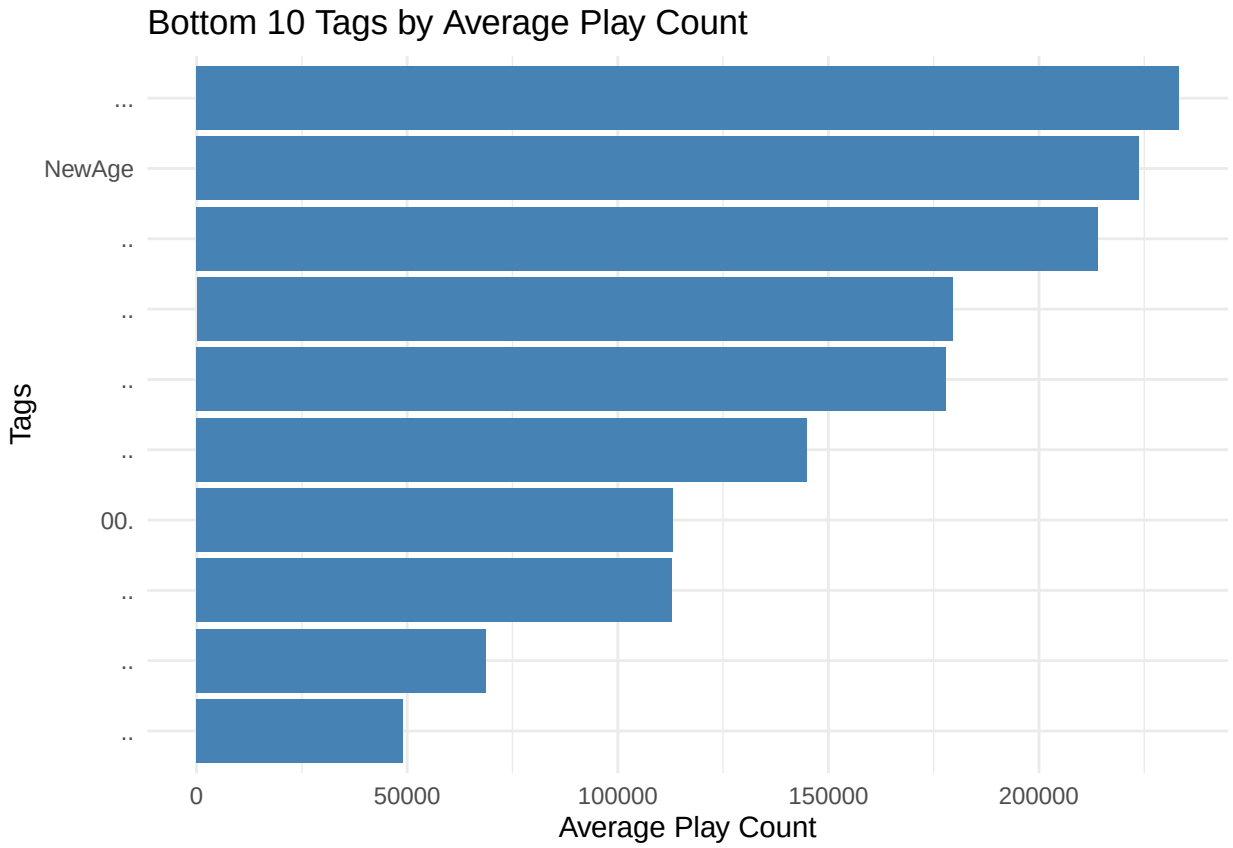
```r
#==== 2    ====
tag_analysis_top <- data_cleaned %>%
  dplyr::select(topics, play_count) %>%
  mutate(topics = str_remove_all(topics, "\\[|\\]|'|\"| ")) %>%
  separate_rows(topics, sep = ",") %>%
  #
  group_by(topics) %>%
  summarise(
    avg_play = mean(play_count),
    count = n()
  ) %>%
  #
  arrange(desc(avg_play)) %>%
  slice_head(n = 10)

tag_analysis_bottom <- data_cleaned %>%
  dplyr::select(topics, play_count) %>%
  mutate(topics = str_remove_all(topics, "\\[|\\]|'|\"| ")) %>%
  separate_rows(topics, sep = ",") %>%
  #
  group_by(topics) %>%
  summarise(
    avg_play = mean(play_count),
    count = n()
  ) %>%
  #
  arrange(avg_play) %>%
  slice_head(n = 10)

tag_analysis_avoid_extreme_top <- data_cleaned %>%
  dplyr::select(topics, play_count) %>%
  mutate(topics = str_remove_all(topics, "\\[|\\]|'|\"| ")) %>%
  separate_rows(topics, sep = ",") %>%
  #
  group_by(topics) %>%
  summarise(
    avg_play = mean(play_count),
    count = n()
  ) %>%
  filter(count > 10) %>%
  #
  arrange(desc(avg_play)) %>%
  slice_head(n = 10)

tag_analysis_avoid_extreme_bottom <- data_cleaned %>%
  dplyr::select(topics, play_count) %>%
  mutate(topics = str_remove_all(topics, "\\[|\\]|'|\"| ")) %>%
  separate_rows(topics, sep = ",") %>%
  #
  group_by(topics) %>%
  summarise(
    avg_play = mean(play_count),
    count = n()
```

```
  ) %>%
  filter(count > 10) %>%
  #
  arrange(avg_play) %>%
  slice_head(n = 10)

tag_freq <- data_cleaned %>%
  dplyr::select(topics, play_count) %>%
  mutate(topics = str_remove_all(topics, "\\[|\\]|'|\"| ")) %>%
  separate_rows(topics, sep = ",") %>%
  #
  group_by(topics) %>%
  summarise(
    count = n()
  ) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)

#
ggplot(tag_analysis_top, aes(x = reorder(topics, avg_play), y = avg_play)) +
  geom_col(fill = "steelblue") +
  coord_flip() + #
  labs(title = "Top 10 Tags by Average Play Count", x = "Tags", y = "Average Play Count") +
  theme_minimal()
```

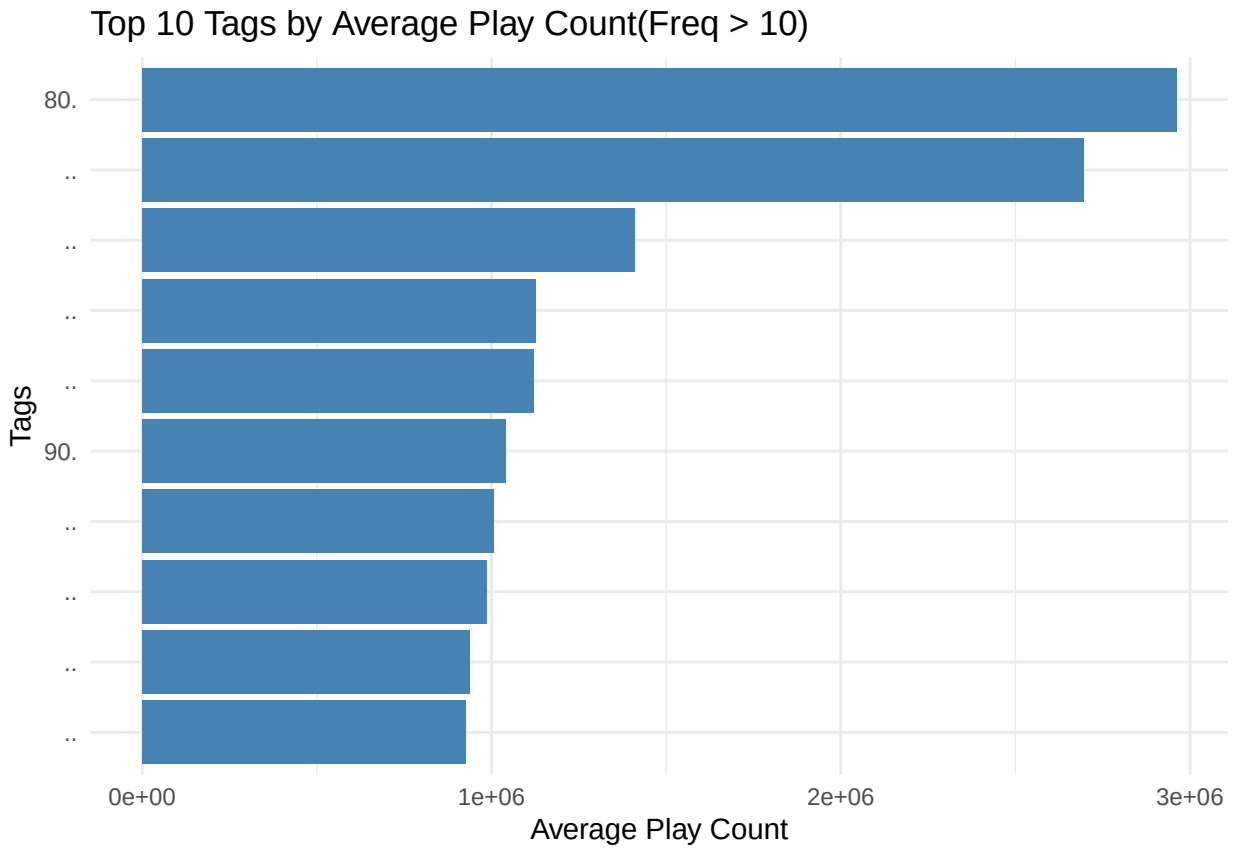



```
ggplot(tag_analysis_bottom, aes(x = reorder(topics, avg_play), y = avg_play)) +
  geom_col(fill = "steelblue") +
```

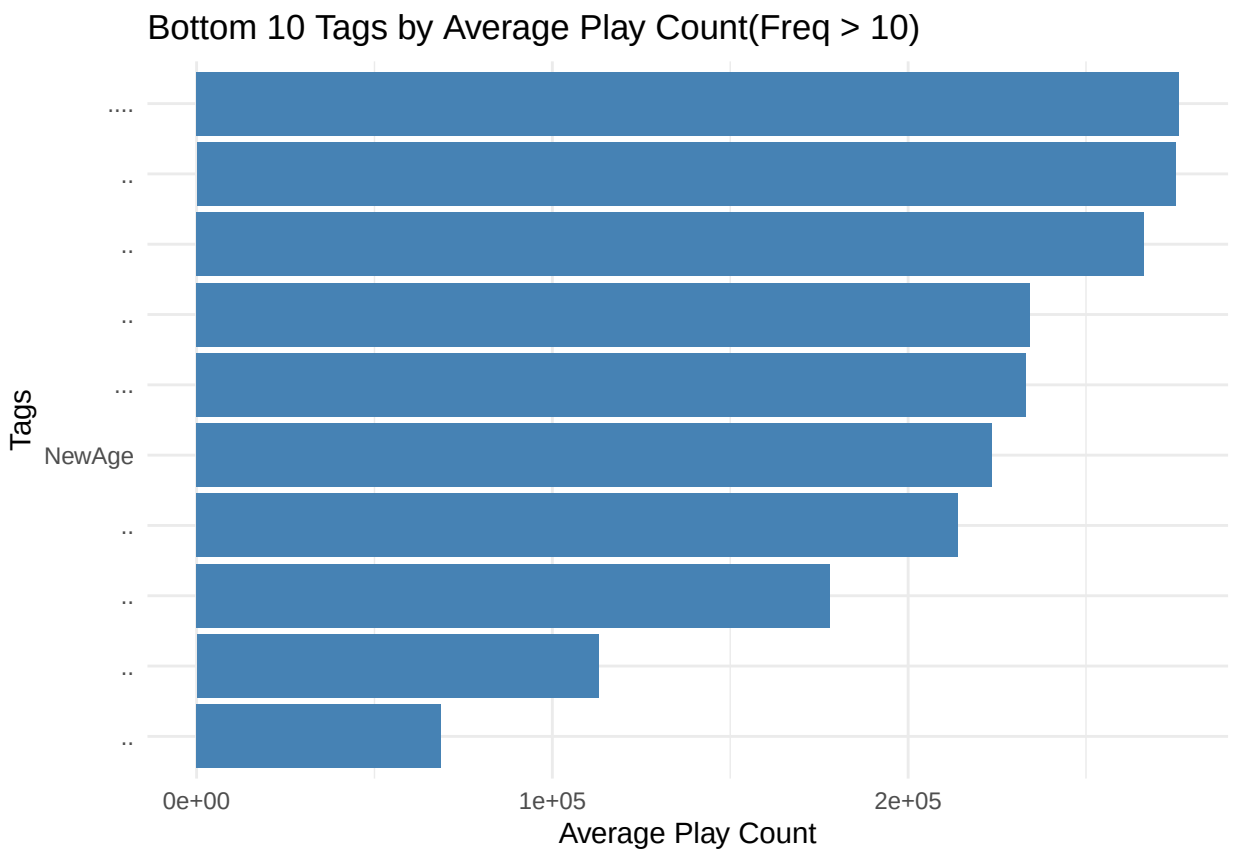
```
  coord_flip() +
  labs(title = "Bottom 10 Tags by Average Play Count", x = "Tags", y = "Average Play Count") +
  theme_minimal()
```



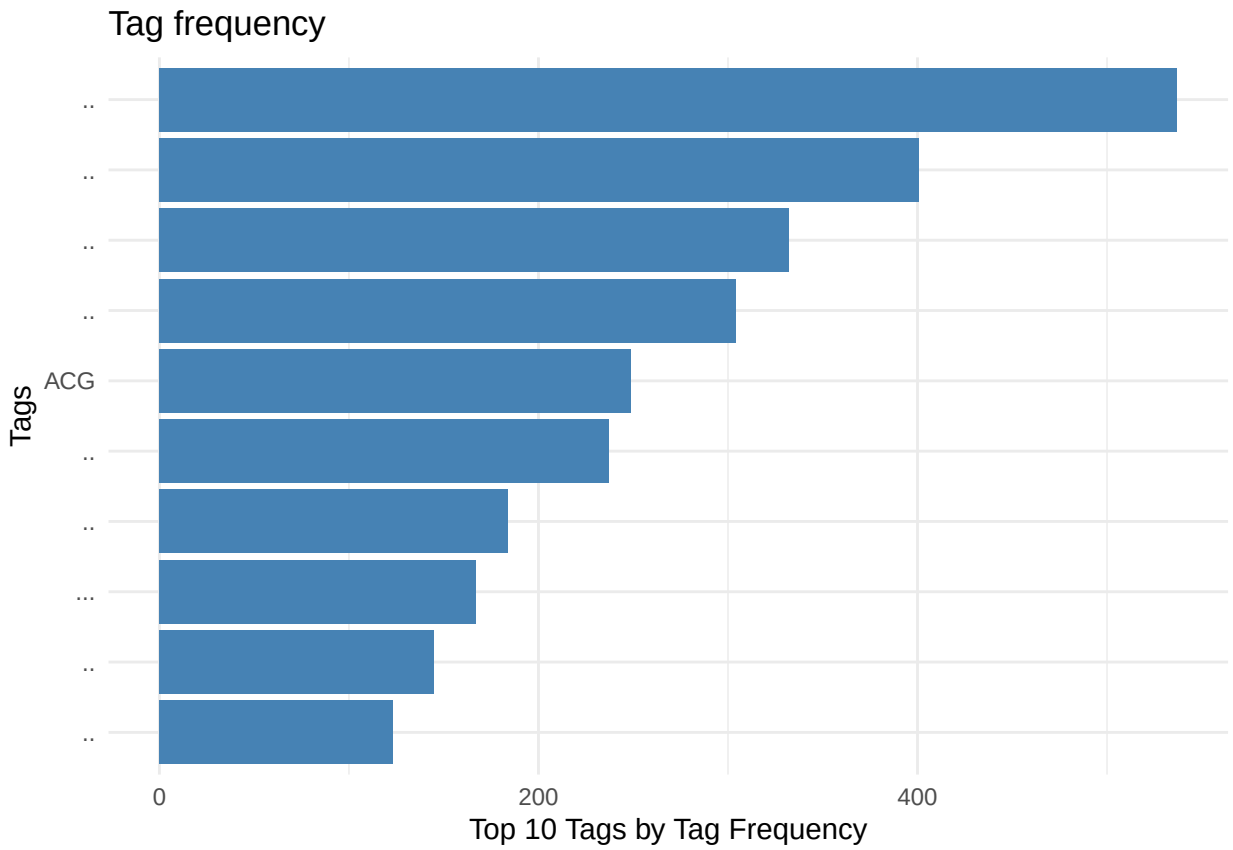

```
ggplot(tag_analysis_avoid_extreme_top, aes(x = reorder(topics, avg_play), y = avg_play)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 Tags by Average Play Count(Freq > 10)", x = "Tags", y = "Average Play Count") +
  theme_minimal()
```




```
ggplot(tag_analysis_avoid_extreme_bottom, aes(x = reorder(topics, avg_play), y = avg_play)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Bottom 10 Tags by Average Play Count(Freq > 10)", x = "Tags", y = "Average Play Count") +
  theme_minimal()
```




```
ggplot(tag_freq, aes(x = reorder(topics, count), y = count)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Tag frequency", x = "Tags", y = "Top 10 Tags by Tag Frequency") +
  theme_minimal()
```



6. ( )

$$\log(\text{play\_count}) \sim \text{tags}$$

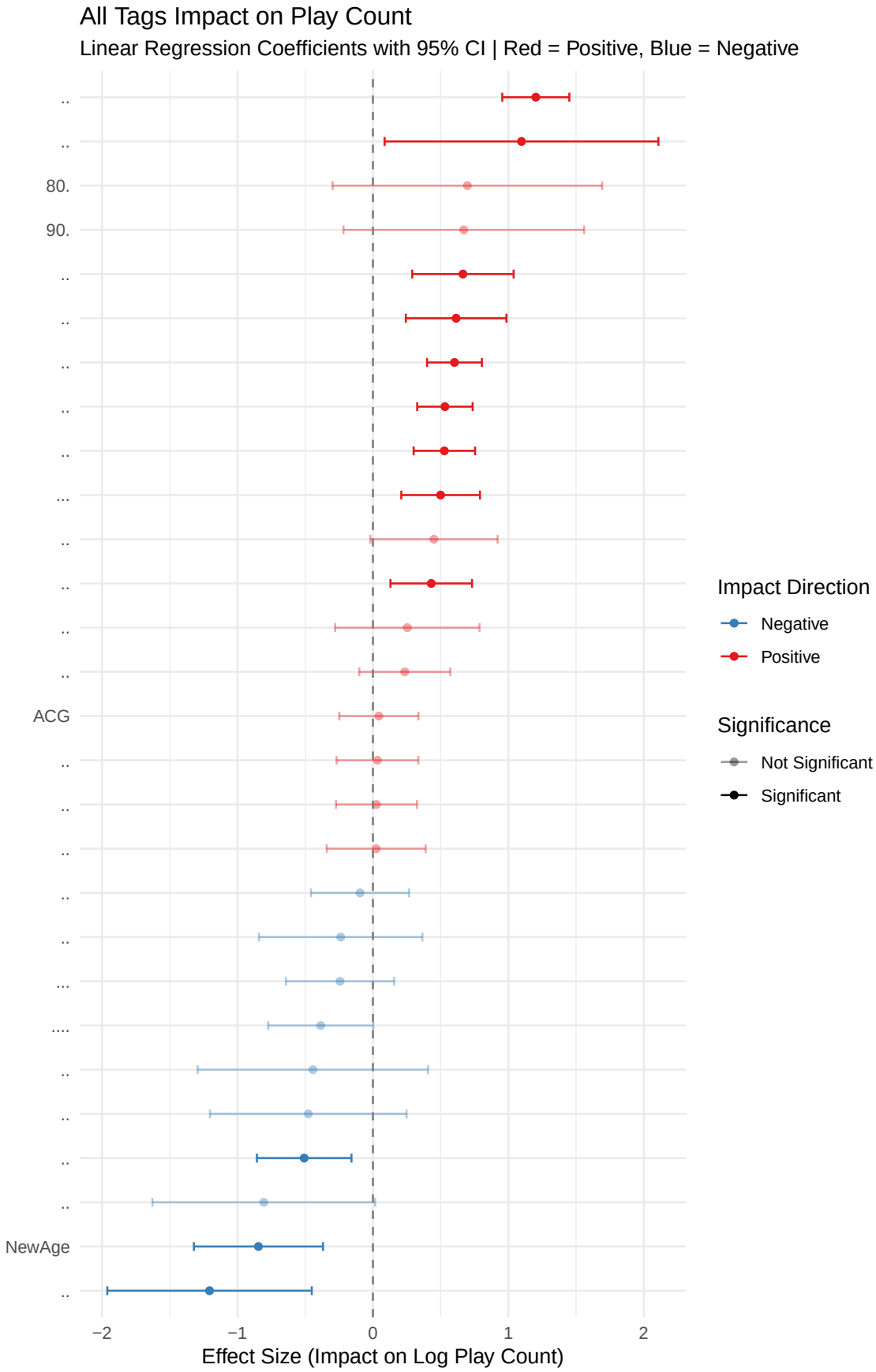
1. $X = 0$ “ ”
2. $X = 0$ NewAge “ ” “ ” Confidence Interval 0
3. 0 $\alpha = 0.05$ “ ” “ ”

```
#==== 3       tag impact====
###
all_tags <- unique(c(
  tag_analysis_avoid_extreme_top$topics,
  tag_analysis_avoid_extreme_bottom$topics,
  tag_freq$topics
))

cat("  ", length(all_tags), "    \n")
```

```
##    28
```

```
### topic
model_data_long <- data_cleaned %>%
  dplyr::select(play_count, topics) %>%
  mutate(
    topics_clean = str_remove_all(topics, "\\[|\\]|'|\"| "),
    log_play = log1p(play_count)
  ) %>%
```

```
  separate_rows(topics_clean, sep = ",") %>%
  filter(topics_clean != "")

###
unique_songs <- model_data_long %>% distinct(play_count, log_play)
tag_matrix <- matrix(0L,
                     nrow = nrow(unique_songs),
                     ncol = length(all_tags))
colnames(tag_matrix) <- paste0("tag_", make.names(all_tags))

#
for(i in seq_len(nrow(unique_songs))) {
  song_play_count <- unique_songs$play_count[i]
  song_tags <- model_data_long$topics_clean[model_data_long$play_count == song_play_count]

  for(tag in song_tags) {
    if(tag %in% all_tags) {
      col_idx <- match(make.names(tag), make.names(all_tags))
      tag_matrix[i, col_idx] <- 1L
    }
  }
}
model_data_wide <- cbind(unique_songs, as.data.frame(tag_matrix))

###
feature_cols <- colnames(tag_matrix)
formula_str <- paste("log_play ~", paste(feature_cols, collapse = " + "))
model_lm <- lm(as.formula(formula_str), data = model_data_wide)

###
model_all_tags <- tidy(model_lm, conf.int = TRUE) %>%
  filter(term %in% feature_cols) %>%
  mutate(
    tag_name = all_tags[match(str_remove(term, "tag_"), make.names(all_tags))],
    significance = ifelse(p.value < 0.05, "Significant", "Not Significant")
  )

#
ggplot(model_all_tags, aes(x = estimate, y = reorder(tag_name, estimate), color = ifelse(estimate > 0,
  geom_point() +
  geom_errorbar(aes(xmin = conf.low, xmax = conf.high), width = 0.2) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "gray50") +
  scale_color_manual(values = c("Positive" = "#E41A1C", "Negative" = "#377EB8")) +
  scale_alpha_manual(values = c("Significant" = 1.0, "Not Significant" = 0.4)) +
  labs(title = "All Tags Impact on Play Count",
       subtitle = "Linear Regression Coefficients with 95% CI | Red = Positive, Blue = Negative",
       x = "Effect Size (Impact on Log Play Count)",
       y = NULL,
       color = "Impact Direction",
       alpha = "Significance") +
  theme_minimal()
```


All Tags Impact on Play Count
Linear Regression Coefficients with 95% CI | Red = Positive, Blue = Negative
..
..
80.
90.
..
..
..
..
..
...
..
..
..
..
ACG
..
..
..
..
..
..
...
....
..
..
..
..
NewAge
..
−2
−1
0
1
2
Effect Size (Impact on Log Play Count)
Impact Direction
Negative
Positive
Significance
Not Significant
Significant

7. ( )

Diagnostic OLS Model log_play ~ collect + share + comment

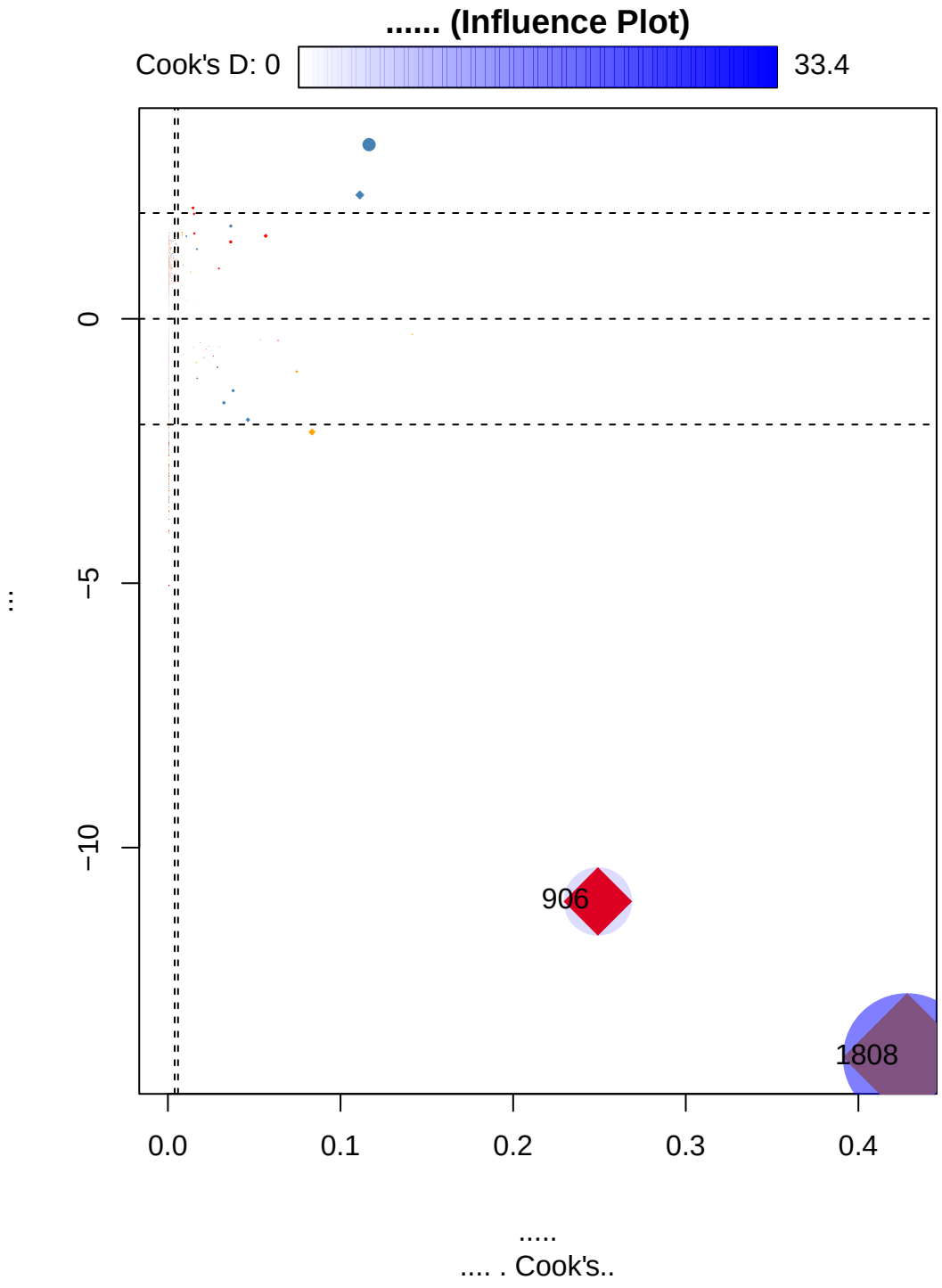
1. F 283.9 ($p < 2.2e^{-16}$) $R^2 = 0.294$ 29.4%
2. collect_count $3.085e^{-05}$ ($t = 12.48, p < 0.001$) " " share_count $(-8.037e^{-04})$
3. Outliers, n=47 47 Cook's Distance " " " " High Leverage Points, n=74 74
4. K-Means 47

```
#==== 4        ====

data_cleaned <- data_cleaned %>%
  #  log1p 0
  mutate(log_play = log1p(play_count)) %>%
  filter(play_count > 0, collect_count >= 0, share_count >= 0, comment_count >= 0) %>%
  na.omit()

### model
diagnostic_formula <- log_play ~ collect_count + share_count + comment_count
diag_model <- lm(diagnostic_formula, data = data_cleaned)
summary(diag_model)
```

```
##
## Call:
## lm(formula = diagnostic_formula, data = data_cleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5107  -0.5694   0.1921   0.9842   4.7456
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.161e+01  3.854e-02 301.162  < 2e-16 ***
## collect_count  3.085e-05  2.472e-06  12.477  < 2e-16 ***
## share_count   -8.037e-04  1.760e-04  -4.566 5.28e-06 ***
## comment_count  1.157e-03  1.902e-04   6.084 1.40e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.537 on 2045 degrees of freedom
## Multiple R-squared:  0.294,  Adjusted R-squared:  0.293
## F-statistic: 283.9 on 3 and 2045 DF,  p-value: < 2.2e-16
```

```
###
par(mar = c(6, 6, 5, 3) + 0.1, mgp = c(3.5, 1, 0))
cooks_dist <- cooks.distance(diag_model)
leverage <- hatvalues(diag_model)
std_resid <- rstandard(diag_model)
p_influence <- influencePlot(
  diag_model,
  main = "    (Influence Plot)",
```

```
  sub = "     Cook's " ,
  xlab = "   ",
  ylab = "  ",
  col = c("steelblue", "orange", "red"),
  pch = c(16, 18),
  lwd = 1.5
)
```



```r
threshold_cook <- 4 / nrow(data_cleaned)
threshold_leverage <- 3 * length(coef(diag_model)) / nrow(data_cleaned)

###
outlier_mask <- cooks_dist > threshold_cook
leverage_mask <- leverage > threshold_leverage

# out_vals <- outlier_mask[outlier_mask == TRUE] #
cat("    :", sum(outlier_mask), " \n")
```

```
##     : 47
```

```r
cat("    :", sum(leverage_mask), " \n")
```

```
##     : 74
```

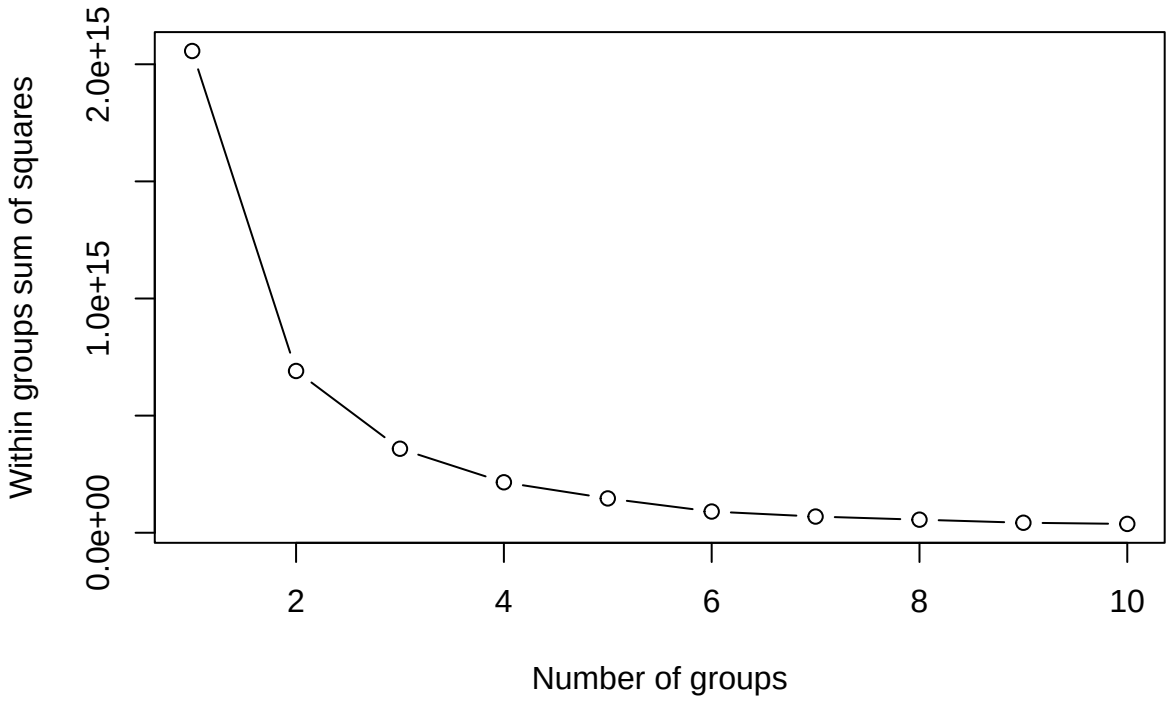
8. K-Means ( )

Outliers K-Means Elbow Method $k = 4$ Between-Cluster Sum of Squares, BCSS Total Sum of Squares, TSS

1. 89.53% $BCSS/TSS$ 90% 4
2. “ ” 99% “ - - ”
3. $k = 4$ Centroids Cluster 1 ( ) Cluster 2 ( ) Cluster 3 ( ) Cluster 4 ( )

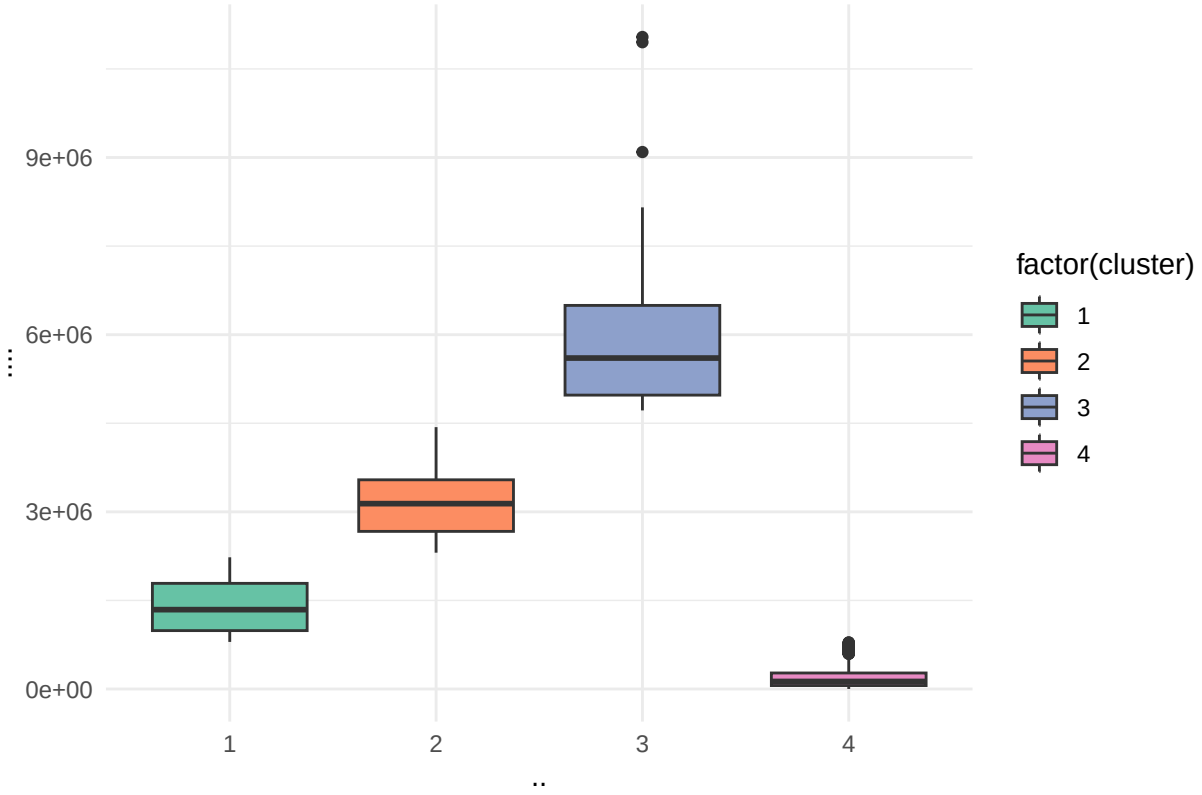
```r
#==== 5            ====
###
data_normal <- data_cleaned %>% filter(!outlier_mask) %>% dplyr::select(log_play, play_count)
data_outliers <- data_cleaned %>% filter(outlier_mask) %>% dplyr::select(log_play, play_count) %>% muta

###
set.seed(1234)
wss = numeric(10)
for (i in 1:10) {
  km = kmeans(data_normal$play_count, centers = i, nstart = 25)
  wss[i] = km$tot.withinss
}
plot(1:10, wss, type = "b", xlab = "Number of groups", ylab = "Within groups sum of squares", main = "
```

.....



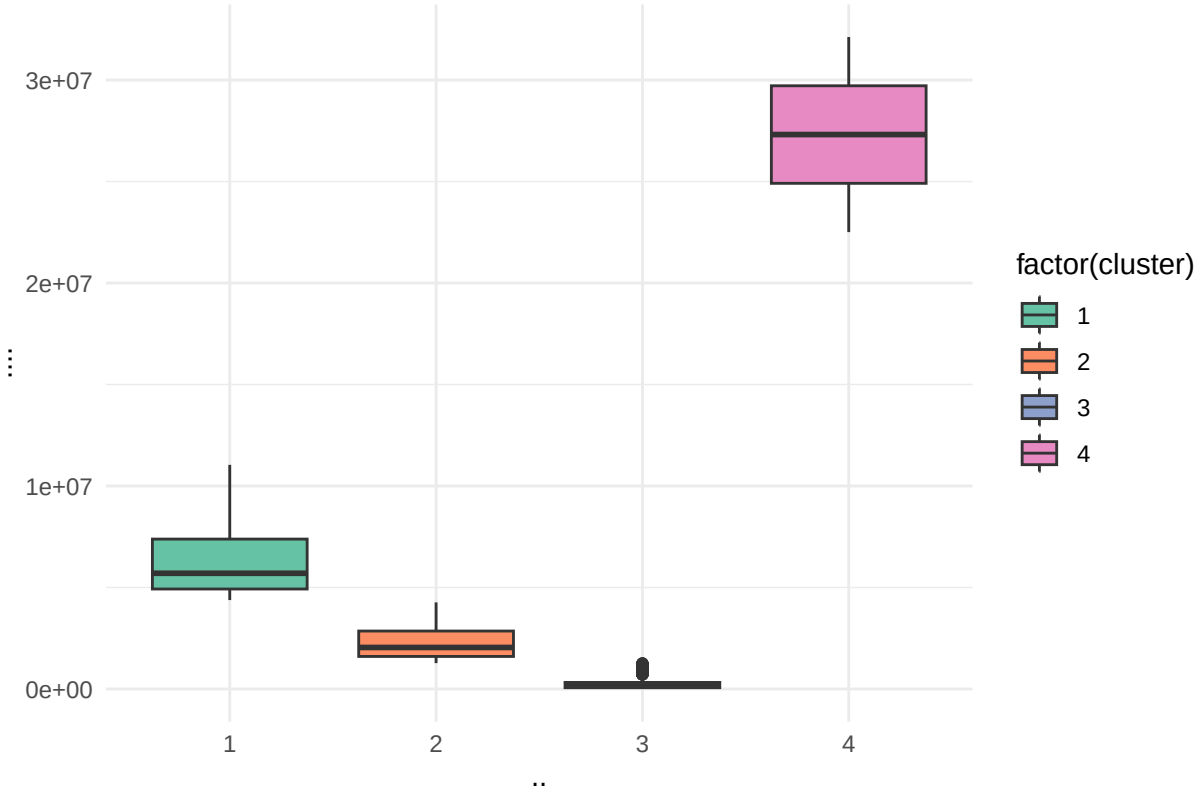

```r
kmeans_function <- function(data, col_names, k) {
  existing_cols <- col_names[col_names %in% names(data)]
  if(length(existing_cols) > 0) {
    data_subset <- data[, existing_cols, drop = FALSE]
    set.seed(123)
    kmeans_result <- kmeans(data_subset, centers = k, nstart = 25)
    cluster_centers <- kmeans_result$centers
    sorted_centers <- cluster_centers[order(cluster_centers[, 1]), , drop = FALSE]
    data$cluster <- kmeans_result$cluster
  } else {
    stop("         ")
  }
  return(list(kmeans_result = kmeans_result, data_with_clusters = data))
}

###
result1 <- kmeans_function(data_normal, "play_count", k = 4)
result2 <- kmeans_function(data_cleaned, "play_count", k = 4)

###
data_clustered1 <- result1$data_with_clusters
ggplot(data_clustered1, aes(x = factor(cluster), y = play_count, fill = factor(cluster))) +
  geom_boxplot() +
  labs(title = "         ", x = " ", y = "   ") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```

.....(.....)




```
data_clustered2 <- result2$data_with_clusters
ggplot(data_clustered2, aes(x = factor(cluster), y = play_count, fill = factor(cluster))) +
  geom_boxplot() +
  labs(title = "          ", x = " ", y = "   ") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```

.....(.....)



..

```
###
explained_variance1 <- result1$kmeans_result$betweenss / result1$kmeans_result$totss
cat("       : ", round(explained_variance1 * 100, 2), "%\n")
```

```
##        :  89.53 %
```

9. ( )

 “ ” “ Daily Play Velocity ” $TotalPlays/DaysAlive$

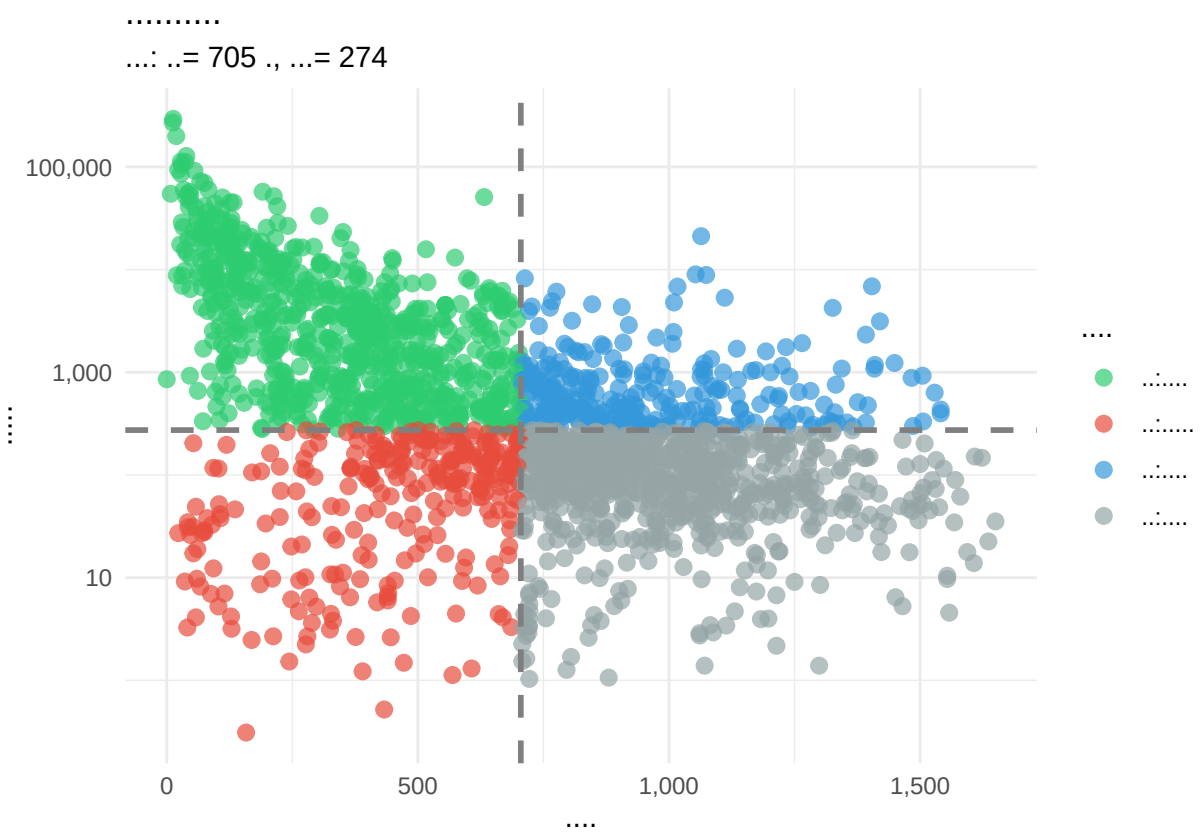
1. top_velocity ° 29.36 / “ ” 19.9 / “ Hook ” 13.8 191.8 “ ”
2. “ - ”

```
#==== 6   ====
reference_date <- max(data_cleaned$create_time)

data_velocity <- data_cleaned %>%
  mutate(
    #
    days_alive = as.numeric(difftime(reference_date, create_time, units = "days")),
    #
    plays_per_day = play_count / ifelse(days_alive==0, 1, days_alive)
  ) %>%
  arrange(desc(plays_per_day))
```

```r
top_velocity <- head(data_velocity %>% dplyr::select(name, plays_per_day, days_alive, fans), 10)
print(top_velocity)
```

```
## # A tibble: 10 x 4
##    name                                    plays_per_day days_alive   fans
##    <chr>                                           <dbl>      <dbl>  <dbl>
##  1     °                    293633.          13   8237
##  2         100            270744           12   1006
##  3                           199384.          19  18482
##  4 Ghost?Producer|              127737.          39  32965
##  5                           112489.          28  38745
##  6                   111933.          37   1008
##  7   |                   104926.          30  10058
##  8 ?                   93943.          23   3032
##  9   |                         91915.          55 139228
## 10                          85897.          28  18482
```

```r
median_days <- median(data_velocity$days_alive)
median_plays <- median(data_velocity$plays_per_day)
data_velocity$quadrant <- with(data_velocity, {
  case_when(
    days_alive <= median_days & plays_per_day > median_plays ~ "    ",
    days_alive <= median_days & plays_per_day <= median_plays ~ "     ",
    days_alive > median_days & plays_per_day > median_plays ~ "    ",
    days_alive > median_days & plays_per_day <= median_plays ~ "    "
  ) %>% factor(levels = c("    ", "     ", "    ", "    "))
})

ggplot(data_velocity, aes(x = days_alive, y = plays_per_day, color = quadrant)) +
  geom_point(alpha = 0.7, size = 2.5) +
  geom_vline(xintercept = median_days, linetype = "dashed", color = "gray50", size = 1) +
  geom_hline(yintercept = median_plays, linetype = "dashed", color = "gray50", size = 1) +
  scale_y_log10(labels = scales::comma) +
  scale_x_continuous(labels = scales::comma) +
  labs(
    title = "      ",
    subtitle = paste("  :  =", round(median_days), " ,   =", round(median_plays)),
    x = "  ", y = "   ",
    color = "  "
  ) +
  scale_color_manual(values = c(
    "    " = "#2ECC71",
    "     " = "#E74C3C",
    "    " = "#3498DB",
    "    " = "#95A5A6"
  )) +
  theme_minimal()
```



10. ( )

StepAIC

1. ANOVA “ ” “ ” “ ” “ ”
2. SEO “ ”

```
#==== 7    ====
data_morph <- data_cleaned %>%
  dplyr::select(play_count, length_name, length_intro, number_songs, number_hot_singers) %>%
  filter(play_count > 0) %>%
  na.omit()

### log   ~    +    +    +
model_morph <- lm(log(play_count) ~ length_name + length_intro + number_songs + number_hot_singers, data
stepAIC(model_morph, direction = "backward")
```

```
## Start:  AIC=2362.53
## log(play_count) ~ length_name + length_intro + number_songs +
##     number_hot_singers
##
##                      Df Sum of Sq    RSS    AIC
## - number_songs        1     1.059 6460.2 2360.9
## <none>                            6459.1 2362.5
## - length_name         1    25.778 6484.9 2368.7
## - length_intro        1   115.969 6575.1 2397.0
```

```
## - number_hot_singers  1   194.424 6653.5 2421.3
##
## Step:  AIC=2360.87
## log(play_count) ~ length_name + length_intro + number_hot_singers
##
##                      Df Sum of Sq    RSS    AIC
## <none>                             6460.2 2360.9
## - length_name         1    26.212 6486.4 2367.2
## - length_intro        1   115.360 6575.5 2395.1
## - number_hot_singers  1   241.949 6702.1 2434.2
```

```
##
## Call:
## lm(formula = log(play_count) ~ length_name + length_intro + number_hot_singers,
##     data = data_morph)
##
## Coefficients:
##        (Intercept)         length_name        length_intro  number_hot_singers
##          11.518501            0.022089            0.001007            0.010535
```

```
model_morph_changed <- lm(log(play_count) ~ length_name + length_intro + number_hot_singers, data = data
anova(model_morph_changed, model_morph)
```

```
## Analysis of Variance Table
##
## Model 1: log(play_count) ~ length_name + length_intro + number_hot_singers
## Model 2: log(play_count) ~ length_name + length_intro + number_songs +
##     number_hot_singers
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   2045 6460.2
## 2   2044 6459.1  1    1.0591 0.3352 0.5627
```

```
summary(model_morph_changed)
```

```
##
## Call:
## lm(formula = log(play_count) ~ length_name + length_intro + number_hot_singers,
##     data = data_morph)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1794 -0.8477  0.0643  1.1811  4.6403
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.152e+01  1.192e-01  96.651  < 2e-16 ***
## length_name        2.209e-02  7.668e-03   2.881  0.00401 **
## length_intro       1.007e-03  1.667e-04   6.043 1.79e-09 ***
## number_hot_singers 1.054e-02  1.204e-03   8.752  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.777 on 2045 degrees of freedom
## Multiple R-squared:  0.05576,    Adjusted R-squared:  0.05438
## F-statistic: 40.26 on 3 and 2045 DF,  p-value: < 2.2e-16
```

11. ( )

log(play_count) ~ fans * grade * playlists + talent * verification $R^2 = 0.1241$ F

1. “ ” playlists $(-1.213e^{-02}, p < 0.001)$ “ ”
2. talent $(0.317, p < 0.001)$ verification $(p = 0.816)$ “ ” “ ” “ ” “ ”
3. talent:verification × $(-1.242, p < 0.05)$ fans:grade × $(p < 0.05)$

```
#==== 8    ====
data_author <- data_cleaned %>%
  dplyr::select(play_count, fans, grade, playlists, talent, verification, musician) %>%
  filter(play_count > 0) %>%
  na.omit()

# BoxCox  (   RMD      )
# boxcox(lm(play_count ~ fans * grade * playlists + talent * verification * musician, data = data_autho
#        lambda = seq(-2, 2, length.out = 100))

model_full <- lm(
  log(play_count) ~ fans * grade * playlists + talent * verification * musician,
  data = data_author
)
summary(model_full)
```

```
##
## Call:
## lm(formula = log(play_count) ~ fans * grade * playlists + talent *
##     verification * musician, data = data_author)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1452 -0.8695  0.0776  1.1014  5.9566
##
## Coefficients: (2 not defined because of singularities)
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         1.267e+01  3.353e-01  37.773  < 2e-16 ***
## fans                                6.163e-06  1.049e-05   0.588 0.556752
## grade                              -8.408e-02  4.141e-02  -2.030 0.042453 *
## playlists                          -1.145e-02  2.654e-03  -4.313 1.69e-05 ***
## talent                              3.239e-01  9.045e-02   3.581 0.000351 ***
## verification                        9.943e-02  5.610e-01   0.177 0.859336
## musician                           -4.474e-02  3.104e-01  -0.144 0.885425
## fans:grade                          2.408e-06  1.195e-06   2.015 0.043991 *
## fans:playlists                     -1.704e-07  2.420e-07  -0.704 0.481306
## grade:playlists                     1.009e-03  2.874e-04   3.513 0.000453 ***
## talent :verification               -1.242e+00  6.083e-01  -2.042 0.041296 *
## talent :musician                   -2.290e-01  4.338e-01  -0.528 0.597607
## verification :musician                     NA         NA      NA       NA
## fans:grade:playlists                1.355e-08  2.446e-08   0.554 0.579662
## talent :verification :musician           NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.715 on 2036 degrees of freedom
## Multiple R-squared:  0.1246, Adjusted R-squared:  0.1194
## F-statistic: 24.15 on 12 and 2036 DF,  p-value: < 2.2e-16
```

```
model_changed <- stepAIC(model_full, direction = "backward")
```

```
## Start:  AIC=2223.76
## log(play_count) ~ fans * grade * playlists + talent * verification *
##     musician
##
##
## Step:  AIC=2223.76
## log(play_count) ~ fans + grade + playlists + talent + verification +
##     musician + fans:grade + fans:playlists + grade:playlists +
##     talent:verification + talent:musician + verification:musician +
##     fans:grade:playlists
##
##
## Step:  AIC=2223.76
## log(play_count) ~ fans + grade + playlists + talent + verification +
##     musician + fans:grade + fans:playlists + grade:playlists +
##     talent:verification + talent:musician + fans:grade:playlists
##
##                         Df Sum of Sq    RSS    AIC
## - talent:musician        1    0.8199 5990.0 2222.0
## - fans:grade:playlists   1    0.9027 5990.1 2222.1
## <none>                               5989.2 2223.8
## - talent:verification    1   12.2641 6001.4 2226.0
##
## Step:  AIC=2222.04
## log(play_count) ~ fans + grade + playlists + talent + verification +
##     musician + fans:grade + fans:playlists + grade:playlists +
##     talent:verification + fans:grade:playlists
##
##                         Df Sum of Sq    RSS    AIC
## - fans:grade:playlists   1    0.9031 5990.9 2220.3
## - musician               1    1.5991 5991.6 2220.6
## <none>                               5990.0 2222.0
## - talent:verification    1   12.0696 6002.1 2224.2
##
## Step:  AIC=2220.35
## log(play_count) ~ fans + grade + playlists + talent + verification +
##     musician + fans:grade + fans:playlists + grade:playlists +
##     talent:verification
##
##                        Df Sum of Sq    RSS    AIC
## - musician              1     1.460 5992.4 2218.8
## <none>                               5990.9 2220.3
## - fans:playlists        1     8.363 5999.3 2221.2
## - talent:verification   1    12.079 6003.0 2222.5
## - fans:grade            1    15.986 6006.9 2223.8
## - grade:playlists       1    54.343 6045.3 2236.9
##
## Step:  AIC=2218.85
## log(play_count) ~ fans + grade + playlists + talent + verification +
```

```
##     fans:grade + fans:playlists + grade:playlists + talent:verification
##
##                       Df Sum of Sq    RSS    AIC
## <none>                              5992.4 2218.8
## - fans:playlists       1     7.922 6000.3 2219.6
## - talent:verification  1    12.280 6004.6 2221.1
## - fans:grade           1    15.838 6008.2 2222.3
## - grade:playlists      1    54.104 6046.5 2235.3
```

```
summary(model_changed)
```

```
##
## Call:
## lm(formula = log(play_count) ~ fans + grade + playlists + talent +
##     verification + fans:grade + fans:playlists + grade:playlists +
##     talent:verification, data = data_author)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1449 -0.8706  0.0753  1.1012  6.0116
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.266e+01  3.350e-01  37.803  < 2e-16 ***
## fans                     3.332e-06  9.104e-06   0.366 0.714423
## grade                   -8.322e-02  4.133e-02  -2.014 0.044182 *
## playlists               -1.213e-02  2.301e-03  -5.270 1.51e-07 ***
## talent                  3.170e-01  8.885e-02   3.568 0.000368 ***
## verification            1.303e-01  5.589e-01   0.233 0.815675
## fans:grade               2.602e-06  1.121e-06   2.321 0.020361 *
## fans:playlists          -3.569e-08  2.174e-08  -1.642 0.100780
## grade:playlists          1.080e-03  2.517e-04   4.291 1.86e-05 ***
## talent :verification  -1.242e+00  6.075e-01  -2.044 0.041068 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.714 on 2039 degrees of freedom
## Multiple R-squared:  0.1241, Adjusted R-squared:  0.1203
## F-statistic: 32.11 on 9 and 2039 DF,  p-value: < 2.2e-16
```

```
anova(model_changed, model_full)
```

```
## Analysis of Variance Table
##
## Model 1: log(play_count) ~ fans + grade + playlists + talent + verification +
##     fans:grade + fans:playlists + grade:playlists + talent:verification
## Model 2: log(play_count) ~ fans * grade * playlists + talent * verification *
##     musician
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   2039 5992.4
## 2   2036 5989.2  3    3.1829 0.3607 0.7814
```

12. ( )

1. “ ” Talent vs None +24.5 P $(p < 0.001)$ “ ”
2. “ ” Verification vs None +53.3 P 0.025 P “ ”
3. “ ” Musician vs None P 0.120 0.05 “ ”

```r
#==== 9  (permutation test  )====
data_id <- data_cleaned %>%
  mutate(
    is_talent = ifelse(talent == " ", 1, 0),
    is_verified = ifelse(verification == " ", 1, 0),
    is_musician = ifelse(musician == " ", 1, 0)
  )

#  4
data_id <- data_id %>%
  mutate(
    identity_4level = case_when(
      is_verified == 1 ~ "Verification",
      is_talent == 1 ~ "Talent",
      is_musician == 1 ~ "Musician",
      TRUE ~ "None"
    ) %>% factor(levels = c("None", "Musician", "Talent", "Verification")))

comparison_pairs <- list(
  c("None", "Talent"),
  c("None", "Musician"),
  c("None", "Verification"),
  c("Musician", "Talent"),
  c("Talent", "Verification"),
  c("Musician", "Verification")
)

run_coin_test <- function(data, group_col, value_col, pair, n_perm = 1000) { # n_perm
  sub_data <- data %>%
    filter(!!sym(group_col) %in% pair) %>%
    dplyr::select(!!sym(group_col), !!sym(value_col)) %>%
    na.omit()

  sub_data[[group_col]] <- factor(sub_data[[group_col]], levels = pair)

  if(nrow(sub_data) < 4 || length(unique(sub_data[[group_col]])) < 2) {
    return(NULL)
  }

  group1_vec <- sub_data[[value_col]][sub_data[[group_col]] == pair[1]]
  group2_vec <- sub_data[[value_col]][sub_data[[group_col]] == pair[2]]
  observed_diff <- mean(group2_vec) - mean(group1_vec)

  test_result <- coin::oneway_test(
    as.formula(paste(value_col, "~", group_col)),
    data = sub_data,
    alternative = "less",
    distribution = coin::approximate(nresample = n_perm)
```

```
  )
  p_value <- coin::pvalue(test_result)

  list(
    comparison = paste(pair[2], "vs", pair[1]),
    mean_diff = observed_diff,
    p_value = as.numeric(p_value),
    statistic = coin::statistic(test_result)
  )
}

results <- lapply(comparison_pairs, function(pair) {
  run_coin_test(data_id, "identity_4level", "play_count", pair)
}) %>% bind_rows()

print(results)
```

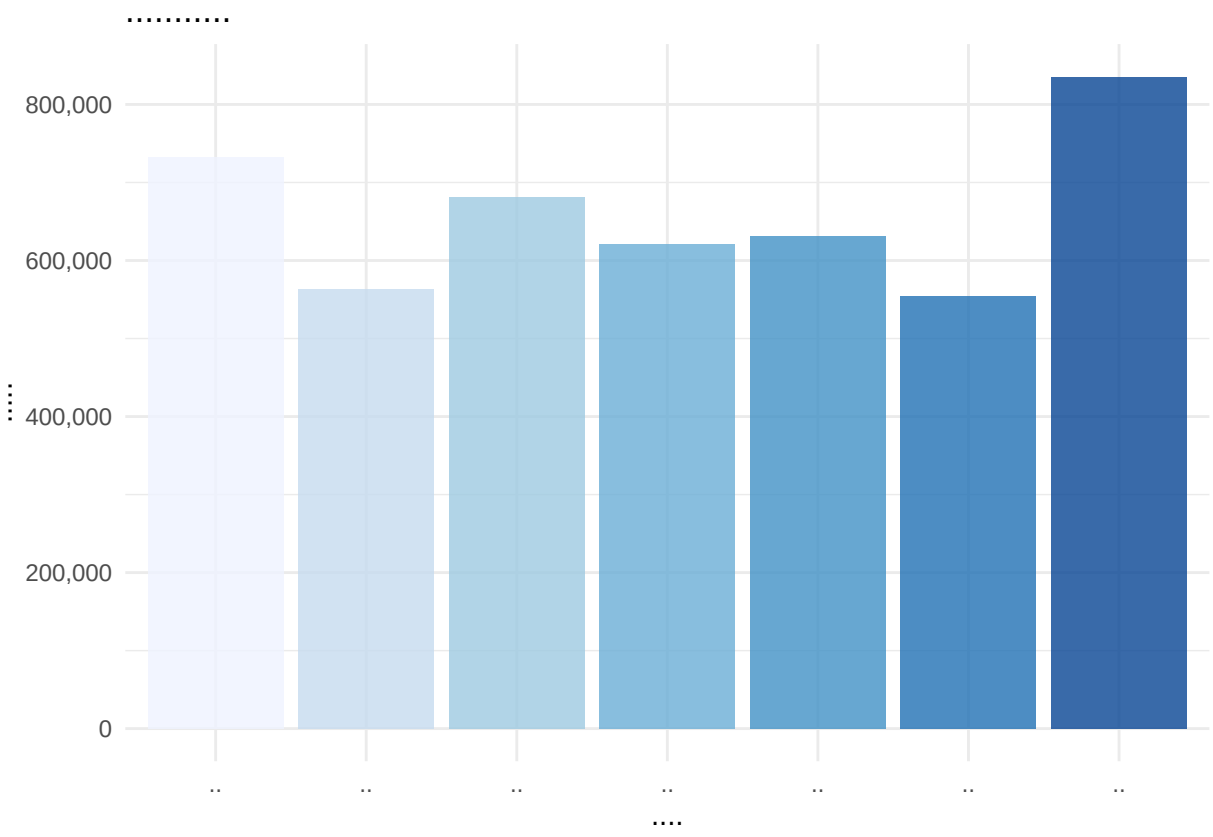
```
## # A tibble: 6 x 4
##   comparison               mean_diff p_value statistic
##   <chr>                        <dbl>   <dbl>     <dbl>
## 1 Talent vs None             245714.   0        -3.77
## 2 Musician vs None           199005.   0.138    -0.709
## 3 Verification vs None       533093.   0.025    -2.71
## 4 Talent vs Musician          46710.   0.455    -0.200
## 5 Verification vs Talent     287379.   0.037    -1.75
## 6 Verification vs Musician   334089.   0.075    -1.23
```

13. ( )

“ Timing ” Temporal Aggregation Day of Week Month of Year

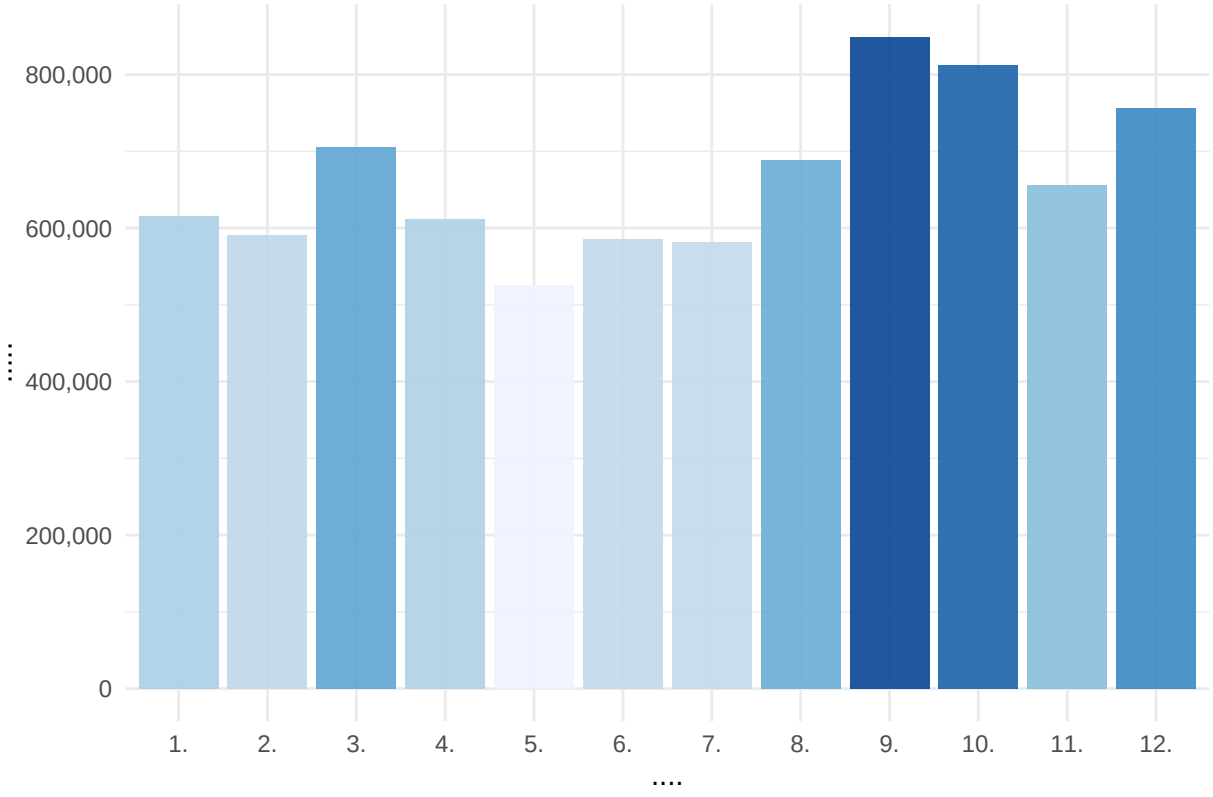
1. “ ” Sunday
2. 9 9 “ ” “ ”

```
#==== 10  ====
###
# 1.
data_time_week <- data_cleaned %>%
  mutate(
    create_dt = as_datetime(create_time),
    day_of_week = factor(wday(create_dt, label = TRUE, week_start = 1))
  ) %>%
  group_by(day_of_week) %>%
  summarise(avg_play = mean(play_count, na.rm = TRUE))

#
ggplot(data_time_week, aes(x = day_of_week, y = avg_play, fill = day_of_week)) +
  geom_col(alpha = 0.8) +
  scale_y_continuous(labels = comma) +
  scale_fill_brewer(palette = "Blues") +
  labs(title = "      ", x = "  ", y = "   ") +
  theme_minimal() +
  theme(legend.position = "none")
```




```
#
data_time_month <- data_cleaned %>%
  mutate(
    create_dt = as_datetime(create_time),
    month_of_year = month(create_dt, label = TRUE, abbr = TRUE)
  ) %>%
  filter(!is.na(month_of_year)) %>%
  group_by(month_of_year) %>%
  summarise(avg_play = mean(play_count, na.rm = TRUE)) %>%
  ungroup()

ggplot(data_time_month, aes(x = month_of_year, y = avg_play)) +
  geom_col(aes(fill = avg_play), alpha = 0.9) +
  scale_fill_distiller(palette = "Blues", direction = 1) +
  scale_y_continuous(labels = comma) +
  labs(title = "      ", x = "  ", y = "   ") +
  theme_minimal() +
  theme(legend.position = "none")
```

.....



.... #

14.

15. PCA- ( )

“ ” PCA Neural Network PCA Latent Features

1. PC1 PC4 “ ” “ ” “ ”
2. $R^2 = 0.741$ “ PCA ” “ ”
3. Prediction Error Matrix “ Tier Classification ” “Low” 409 “ ” Low Tier “Medium” “High”

```
# ==== 12     error ====

nn_data <- data_cleaned %>%
  dplyr::select(
    play_count,
    collect_count, share_count, comment_count, #   /
    fans, grade, playlists,                #
    length_name, length_intro, number_songs, number_hot_singers #
  ) %>%
  na.omit()

# 2. PCA
pca_features <- nn_data %>%
  dplyr::select(fans, grade, playlists, length_name, length_intro, number_songs, number_hot_singers)
```

```r
pca_res <- prcomp(pca_features, scale. = TRUE)
pca_scores <- as.data.frame(pca_res$x[, 1:4])
names(pca_scores) <- c("PC1", "PC2", "PC3", "PC4")

#
model_data <- cbind(
  dplyr::select(nn_data, play_count, collect_count, share_count, comment_count),
  pca_scores
)

#  /
preproc_values <- preProcess(model_data, method = c("center", "scale"))
model_data_scaled <- predict(preproc_values, model_data)

#
set.seed(123)
train_index <- createDataPartition(model_data_scaled$play_count, p = 0.7, list = FALSE)
train_set <- model_data_scaled[train_index, ]
test_set <- model_data_scaled[-train_index, ]

#
nn_formula <- play_count ~ collect_count + share_count + comment_count + PC1 + PC2 + PC3 + PC4
nn_model <- nnet(nn_formula, data = train_set, size = 10, decay = 0.01, linout = TRUE, maxit = 500, tra

#
predictions_scaled <- predict(nn_model, test_set)

#
play_count_mean <- mean(model_data$play_count)
play_count_sd <- sd(model_data$play_count)
predictions_raw <- (predictions_scaled * play_count_sd) + play_count_mean
test_set$play_count_raw <- (test_set$play_count * play_count_sd) + play_count_mean

rmse_val <- sqrt(mean((predictions_raw - test_set$play_count_raw)^2))
r2_val <- cor(predictions_raw, test_set$play_count_raw)^2

cat("\n===      ===\n")
```

```
##
## ===      ===
```

```r
cat("RMSE:", rmse_val, "\n")
```

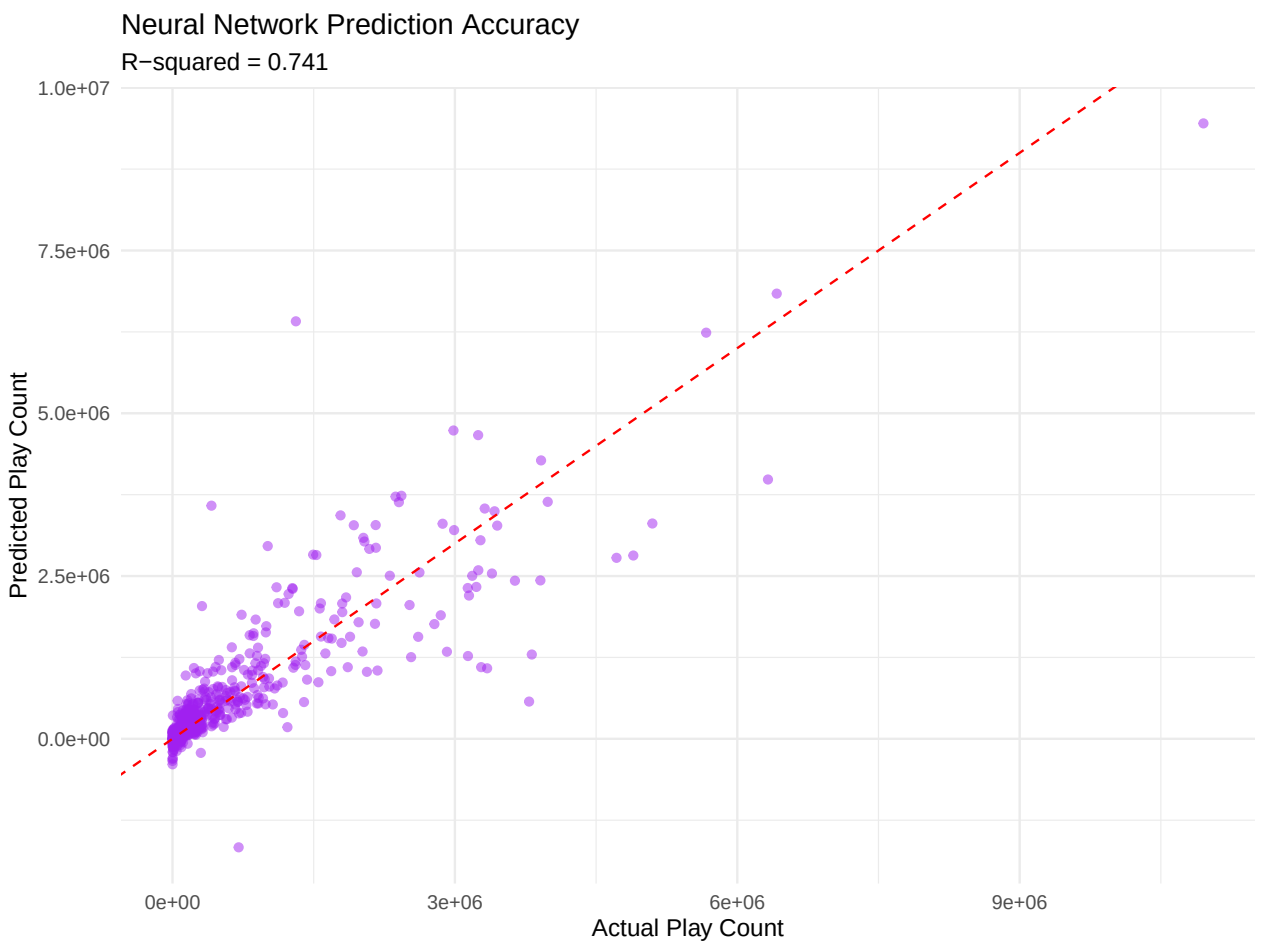
```
## RMSE: 535945.9
```

```r
cat("R-squared:", r2_val, "\n")
```

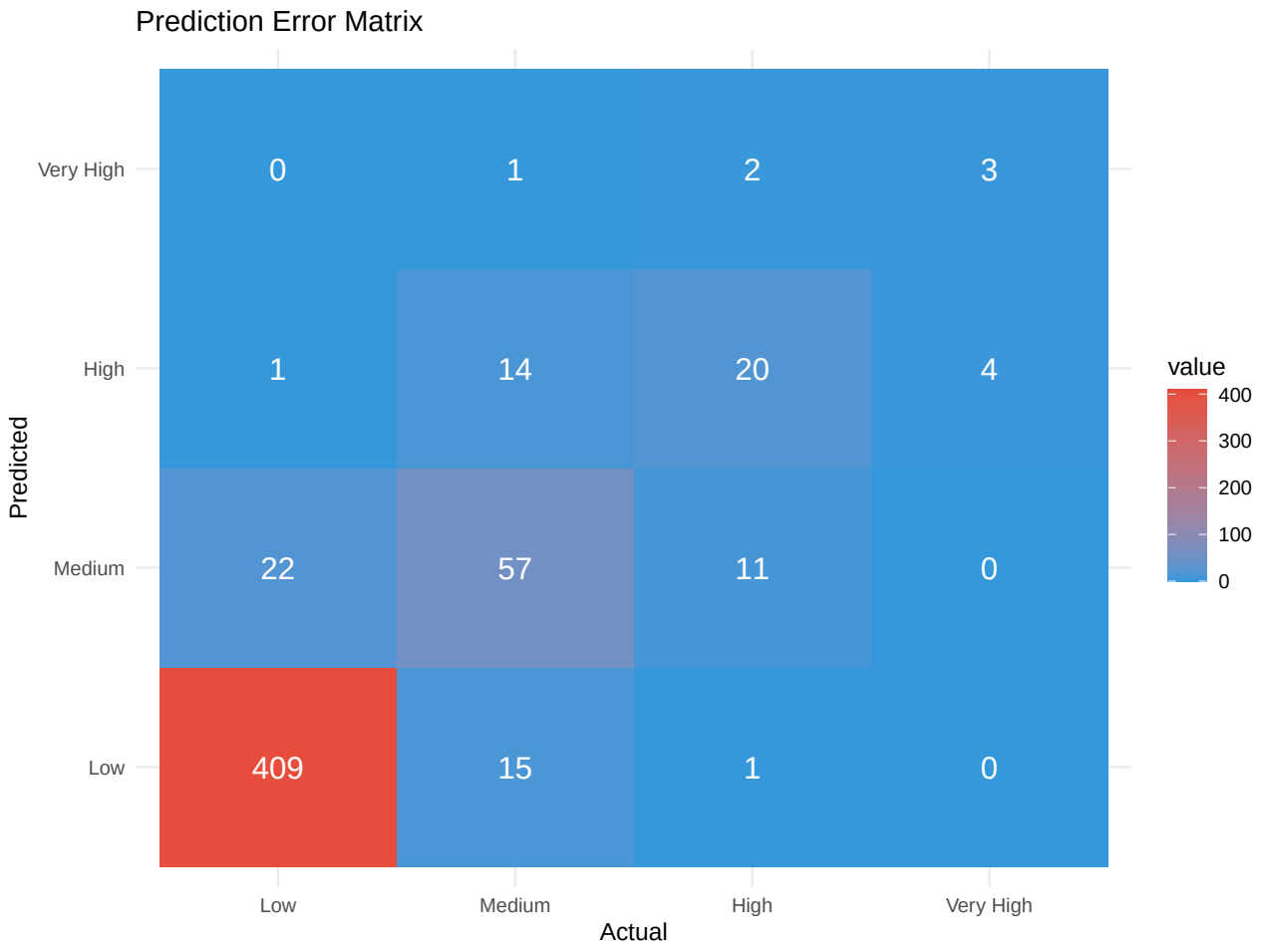
```
## R-squared: 0.741072
```

```r
#    vs
pred_df <- data.frame(Actual = test_set$play_count_raw, Predicted = predictions_raw)

ggplot(pred_df, aes(x = Actual, y = Predicted)) +
  geom_point(alpha = 0.5, color = "purple") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(
```

```
    title = "Neural Network Prediction Accuracy",
    subtitle = paste("R-squared =", round(r2_val, 3)),
    x = "Actual Play Count",
    y = "Predicted Play Count"
  ) +
  theme_minimal()
```




```
#
breaks <- c(0, 793134.5, 2269060.5, 4575695, 32119005)
actual_class <- cut(test_set$play_count_raw, breaks = breaks, labels = c("Low", "Medium", "High", "Very
pred_class <- cut(predictions_raw, breaks = breaks, labels = c("Low", "Medium", "High", "Very High"), in

error_matrix <- table(Predicted = pred_class, Actual = actual_class)
melted_cmat <- melt(error_matrix)

ggplot(data = melted_cmat, aes(x = Actual, y = Predicted, fill = value)) +
  geom_tile() +
  geom_text(aes(label = value), color = "white", size = 5) +
  scale_fill_gradient(low = "#3498DB", high = "#E74C3C") +
  labs(title = "Prediction Error Matrix") +
  theme_minimal()
```



16.

2048 “ ” PCA- $R^2 = 0.741$ “ ”

- “ ” PCA- “ ” “ ” Low Tier “Low” 40% “ ” “Medium” “High” “ ”
- SEO “ ” Text Density “2+1” 2 1 9 “ ”
- “ ” “ / ” “ ”