final assignment 1

Yue Sun

2020/12/4

```
library(readxl)
df <- read_excel("C:/Users/Yue Sun/Desktop/msca/fall 2020/msca 31015 consulting/final project/file/Final
```

```
## New names:
## * `` -> ...67
## * `` -> ...68
```

```
head(df)
```

```
## # A tibble: 6 x 69
##      id logtarg     r     f     m   tof Ffiction1 Fclassics3 Fcartoons5
##   <dbl> <lgl>   <dbl> <dbl> <dbl> <dbl>     <dbl>      <dbl>      <dbl>
## 1   914 NA        194     7 319.   1703         1          0          1
## 2   957 NA          3    14 368.   2364         1          0          2
## 3  1406 NA       1489    15 423.   2371         0          0          0
## 4  1414 NA        155     4  71.2  1290         0          0          0
## 5  1546 NA        194     6 443.   2188         0          0          1
## 6  1651 NA       1797     2  47.9  1808         0          0          0
## # ... with 60 more variables: Flegends6 <dbl>, Fphilosophy7 <dbl>,
## #   Freligion8 <dbl>, Fpsychology9 <dbl>, Flinguistics10 <dbl>, Fart12 <dbl>,
## #   Fmusic14 <dbl>, Ffacsimile17 <dbl>, Fhistory19 <dbl>, Fconthist20 <dbl>,
## #   Feconomy21 <dbl>, Fpolitics22 <dbl>, Fscience23 <dbl>, Fcompsci26 <dbl>,
## #   Frailroads27 <dbl>, Fmaps30 <dbl>, Ftravelguides31 <dbl>, Fhealth35 <dbl>,
## #   Fcooking36 <dbl>, Flearning37 <dbl>, FGamesRiddles38 <dbl>,
## #   Fsports39 <dbl>, Fhobby40 <dbl>, Fnature41 <dbl>, Fencyclopaedia44 <dbl>,
## #   Fvideos50 <dbl>, Fnonbooks99 <dbl>, Mfiction1 <dbl>, Mclassics3 <dbl>,
## #   Mcartoons5 <dbl>, Mlegends6 <dbl>, Mphilosophy7 <dbl>, Mreligion8 <dbl>,
## #   Mpsychology9 <dbl>, Mlinguistics10 <dbl>, Mart12 <dbl>, Mmusic14 <dbl>,
## #   Mfacsimile17 <dbl>, Mhistory19 <dbl>, Mconthist20 <dbl>, Meconomy21 <dbl>,
## #   Mpolitics22 <dbl>, Mscience23 <dbl>, Mcompsci26 <dbl>, Mrailroads27 <dbl>,
## #   Mmaps30 <dbl>, Mtravelguides31 <dbl>, Mhealth35 <dbl>, Mcooking36 <dbl>,
## #   Mlearning37 <dbl>, MGamesRiddles38 <dbl>, Msports39 <dbl>, Mhobby40 <dbl>,
## #   Mnature41 <dbl>, Mencyclopaedia44 <dbl>, Mvideos50 <dbl>,
## #   Mnonbooks99 <dbl>, ...67 <lgl>, ...68 <lgl>, `Dear Greg,` <chr>
```

```
# drop columns
df = df[, -c(2,67:69)]
```

1. Perform descriptive statistics on all variables to help understand the data, the distributions and basic info you have to work with on this challenge. (Mean, Standard Deviation, Median, Min, Max and Histogram)

```
NROW(df$id)
```

```
## [1] 33713
```

```
# there are 33713 customers
summary(df)
```

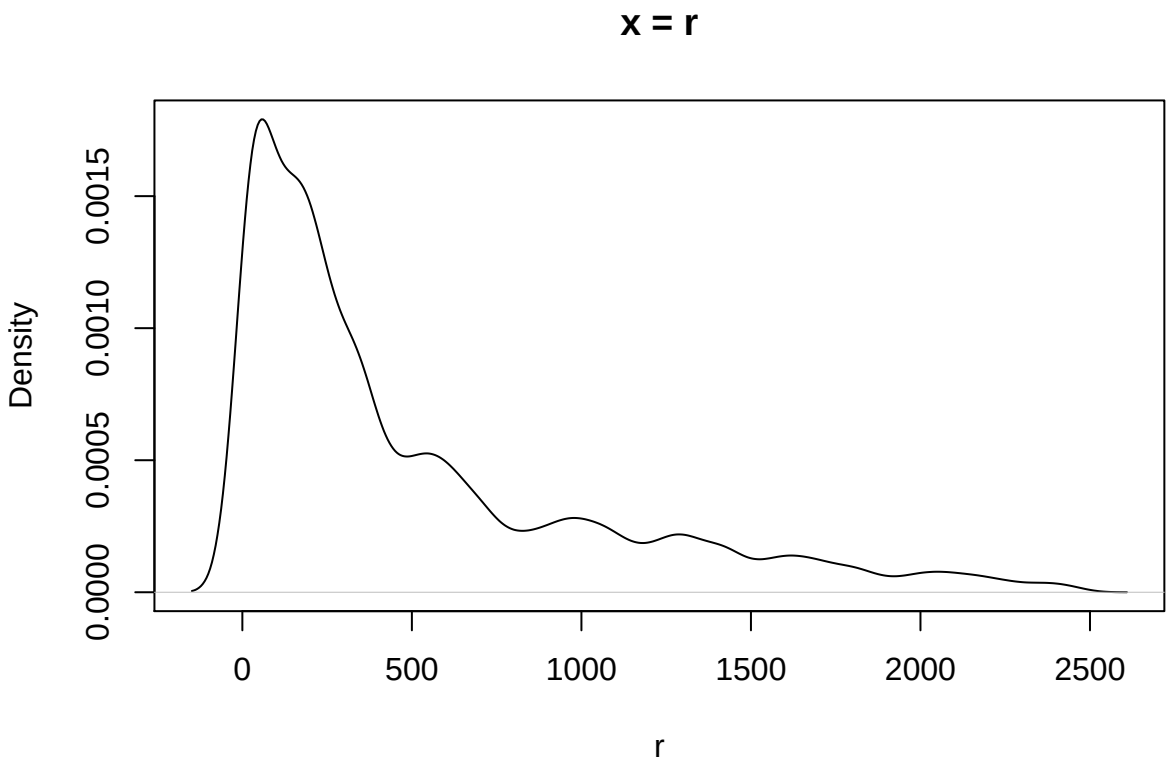
```
##        id                 r                f                m          
##  Min.   :     914   Min.   :   0.0   Min.   :  0.000   Min.   :     0.0  
##  1st Qu.: 4391616   1st Qu.: 117.0   1st Qu.:  1.000   1st Qu.:    39.9  
##  Median : 7967691   Median : 293.0   Median :  3.000   Median :   102.6  
##  Mean   : 8234209   Mean   : 508.4   Mean   :  5.821   Mean   :   251.9  
##  3rd Qu.:11998229   3rd Qu.: 712.0   3rd Qu.:  7.000   3rd Qu.:   247.3  
##  Max.   :16252640   Max.   :2460.0   Max.   :118.000   Max.   :532892.0  
##       tof          Ffiction1          Fclassics3        Fcartoons5       
##  Min.   :   0   Min.   :  0.0000   Min.   : 0.0000   Min.   : 0.0000  
##  1st Qu.: 536   1st Qu.:  0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000  
##  Median :1311   Median :  0.0000   Median : 0.0000   Median : 0.0000  
##  Mean   :1296   Mean   :  0.5909   Mean   : 0.2281   Mean   : 0.3565  
##  3rd Qu.:2132   3rd Qu.:  1.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000  
##  Max.   :2462   Max.   :109.0000   Max.   :29.0000   Max.   :49.0000  
##    Flegends6        Fphilosophy7       Freligion8       Fpsychology9     
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :  0.000   Min.   :0.00000  
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:  0.000   1st Qu.:0.00000  
##  Median : 0.0000   Median : 0.0000   Median :  0.000   Median :0.00000  
##  Mean   : 0.1048   Mean   : 0.2907   Mean   :  0.981   Mean   :0.04657  
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:  1.000   3rd Qu.:0.00000  
##  Max.   :22.0000   Max.   :35.0000   Max.   :134.000   Max.   :8.00000  
##  Flinguistics10        Fart12            Fmusic14        Ffacsimile17      
##  Min.   : 0.0000   Min.   :  0.0000   Min.   :  0.000   Min.   : 0.00000  
##  1st Qu.: 0.0000   1st Qu.:  0.0000   1st Qu.:  0.000   1st Qu.: 0.00000  
##  Median : 0.0000   Median :  0.0000   Median :  0.000   Median : 0.00000  
##  Mean   : 0.2032   Mean   :  0.6086   Mean   :  2.321   Mean   : 0.05096  
##  3rd Qu.: 0.0000   3rd Qu.:  1.0000   3rd Qu.:  2.000   3rd Qu.: 0.00000  
##  Max.   :21.0000   Max.   :113.0000   Max.   :264.000   Max.   :23.00000  
##    Fhistory19       Fconthist20        Feconomy21       Fpolitics22      
##  Min.   :  0.000   Min.   :  0.000   Min.   : 0.0000   Min.   :0.00000  
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.: 0.0000   1st Qu.:0.00000  
##  Median :  1.000   Median :  0.000   Median : 0.0000   Median :0.00000  
##  Mean   :  2.648   Mean   :  2.154   Mean   : 0.2084   Mean   :0.02996  
##  3rd Qu.:  3.000   3rd Qu.:  2.000   3rd Qu.: 0.0000   3rd Qu.:0.00000  
##  Max.   :192.000   Max.   :213.000   Max.   :18.0000   Max.   :6.00000  
##    Fscience23        Fcompsci26        Frailroads27       Fmaps30          
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.000   Min.   : 0.00000  
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.00000  
##  Median : 0.0000   Median : 0.0000   Median : 0.000   Median : 0.00000  
##  Mean   : 0.4108   Mean   : 0.1509   Mean   : 0.187   Mean   : 0.07499  
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.000   3rd Qu.: 0.00000
```

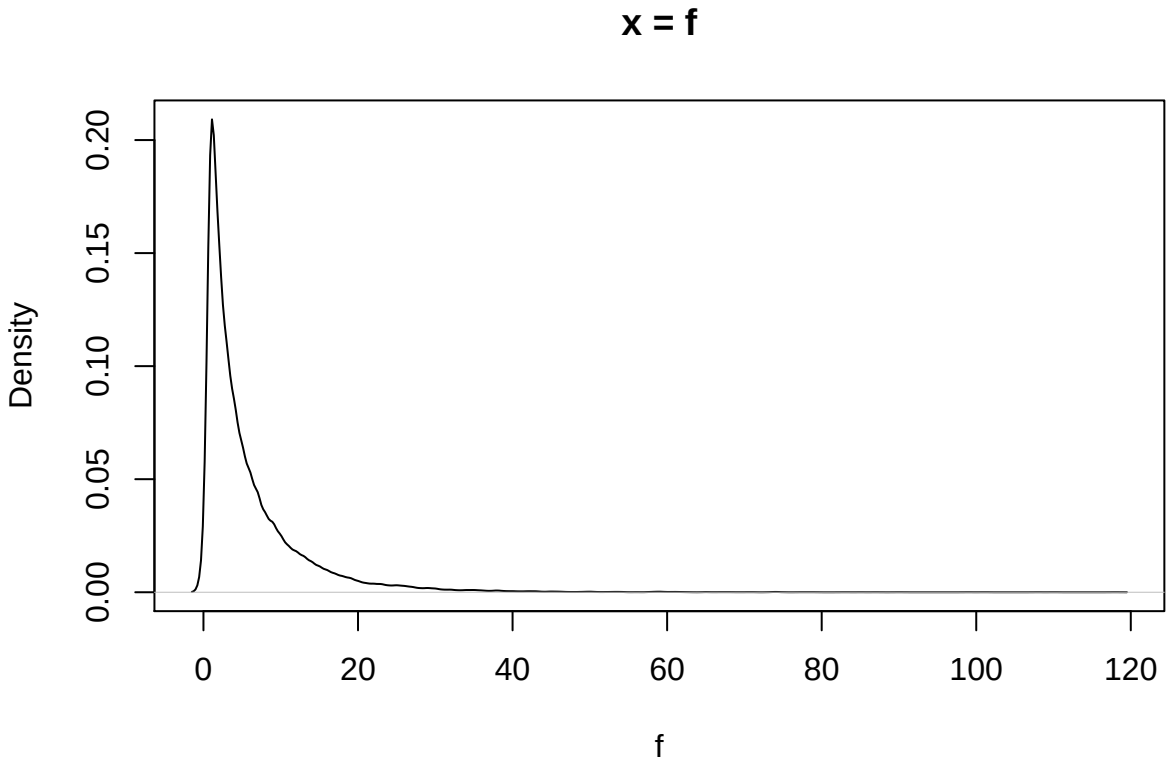
```
##  Max.   :55.0000   Max.   :17.0000   Max.   :61.000   Max.   :10.00000
##  Ftravelguides31     Fhealth35        Fcooking36       Flearning37
##  Min.   :  0.000   Min.   :  0.00   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.:  0.000   1st Qu.:  0.00   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median :  0.000   Median :  0.00   Median : 0.0000   Median : 0.0000
##  Mean   :  1.485   Mean   :  2.11   Mean   : 0.4254   Mean   : 0.6172
##  3rd Qu.:  1.000   3rd Qu.:  2.00   3rd Qu.: 0.0000   3rd Qu.: 1.0000
##  Max.   :164.000   Max.   :210.00   Max.   :78.0000   Max.   :59.0000
##  FGamesRiddles38    Fsports39           Fhobby40         Fnature41
##  Min.   :0.0000   Min.   :0.000000   Min.   : 0.0000   Min.   : 0.0000
##  1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median :0.0000   Median :0.000000   Median : 0.0000   Median : 0.0000
##  Mean   :0.0345   Mean   :0.001809   Mean   : 0.6377   Mean   : 0.3172
##  3rd Qu.:0.0000   3rd Qu.:0.000000   3rd Qu.: 1.0000   3rd Qu.: 0.0000
##  Max.   :6.0000   Max.   :2.000000   Max.   :56.0000   Max.   :53.0000
##  Fencyclopaedia44    Fvideos50        Fnonbooks99         Mfiction1
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :  0.0000   Min.   :   0.000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:  0.0000   1st Qu.:   0.000
##  Median : 0.0000   Median : 0.0000   Median :  0.0000   Median :   0.000
##  Mean   : 0.4809   Mean   : 0.4076   Mean   :  0.4632   Mean   :   5.192
##  3rd Qu.: 1.0000   3rd Qu.: 0.0000   3rd Qu.:  1.0000   3rd Qu.:   2.557
##  Max.   :50.0000   Max.   :55.0000   Max.   :110.0000   Max.   :1085.540
##    Mclassics3         Mcartoons5        Mlegends6       Mphilosophy7
##  Min.   :   0.000   Min.   :  0.000   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:   0.000   1st Qu.:  0.000   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :   0.000   Median :  0.000   Median :  0.00   Median :  0.000
##  Mean   :   3.909   Mean   :  2.494   Mean   :  1.25   Mean   :  3.807
##  3rd Qu.:   0.000   3rd Qu.:  0.000   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :1727.144   Max.   :344.656   Max.   :454.51   Max.   :654.949
##    Mreligion8         Mpsychology9      Mlinguistics10        Mart12
##  Min.   :     0.00   Min.   :  0.0000   Min.   :  0.000   Min.   :   0.000
##  1st Qu.:     0.00   1st Qu.:  0.0000   1st Qu.:  0.000   1st Qu.:   0.000
##  Median :     0.00   Median :  0.0000   Median :  0.000   Median :   0.000
##  Mean   :    16.37   Mean   :  0.7166   Mean   :  2.214   Mean   :  10.756
##  3rd Qu.:     7.95   3rd Qu.:  0.0000   3rd Qu.:  0.000   3rd Qu.:   5.087
##  Max.   :146154.75   Max.   :299.0000   Max.   :810.398   Max.   :2632.572
##     Mmusic14          Mfacsimile17        Mhistory19        Mconthist20
##  Min.   :     0.00   Min.   :   0.000   Min.   :   0.000   Min.   :   0.00
##  1st Qu.:     0.00   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:   0.00
##  Median :     0.00   Median :   0.000   Median :   8.641   Median :   0.00
##  Mean   :    29.26   Mean   :   3.695   Mean   :  39.064   Mean   :  24.95
##  3rd Qu.:    19.24   3rd Qu.:   0.000   3rd Qu.:  39.850   3rd Qu.:  20.40
##  Max.   :252683.38   Max.   :9440.883   Max.   :4158.086   Max.   :2489.91
##    Meconomy21        Mpolitics22        Mscience23        Mcompsci26
##  Min.   :  0.000   Min.   : 0.0000   Min.   :  0.000   Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.: 0.0000   1st Qu.:  0.000   1st Qu.:  0.000
##  Median :  0.000   Median : 0.0000   Median :  0.000   Median :  0.000
##  Mean   :  2.016   Mean   : 0.3151   Mean   :  4.862   Mean   :  1.328
##  3rd Qu.:  0.000   3rd Qu.: 0.0000   3rd Qu.:  0.000   3rd Qu.:  0.000
##  Max.   :242.270   Max.   :55.9080   Max.   :585.910   Max.   :173.527
##   Mrailroads27        Mmaps30         Mtravelguides31     Mhealth35
##  Min.   :  0.000   Min.   :  0.000   Min.   :   0.00   Min.   :     0.0
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:   0.00   1st Qu.:     0.0
##  Median :  0.000   Median :  0.000   Median :   0.00   Median :     0.0
```

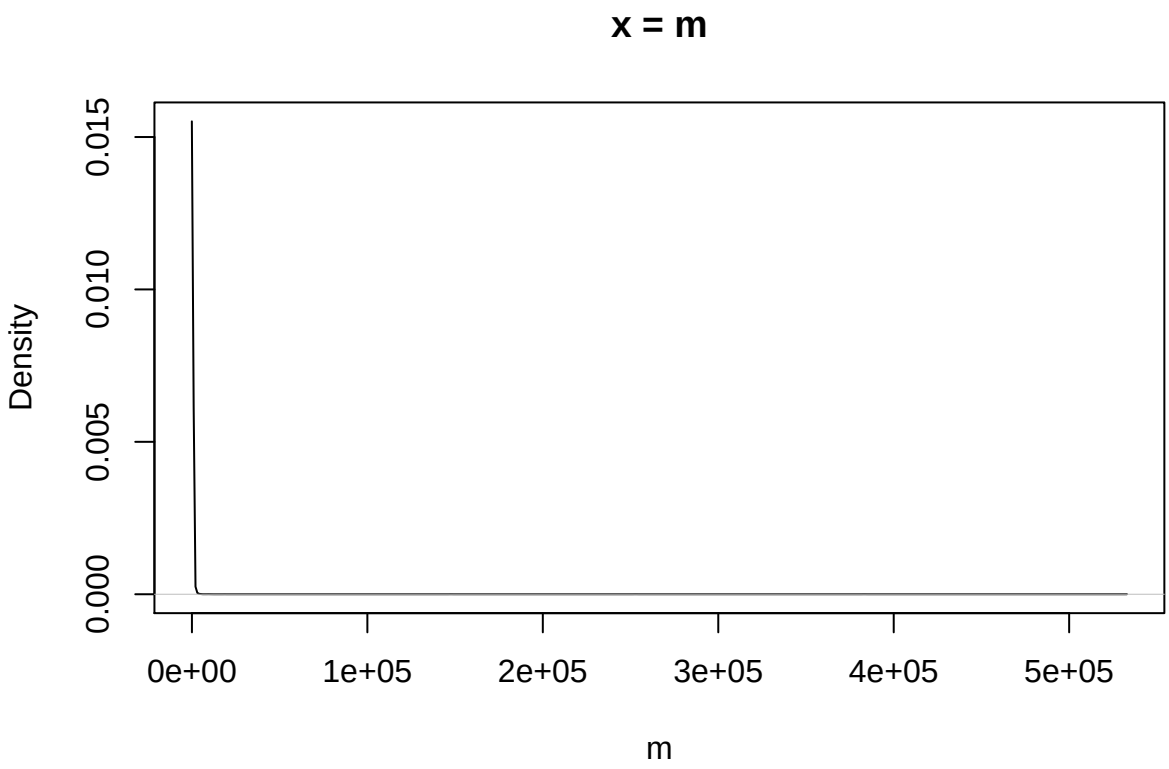
```
##  Mean   :  2.103   Mean   :  1.457   Mean   :  14.63   Mean   :    29.1
##  3rd Qu.:  0.000   3rd Qu.:  0.000   3rd Qu.:  12.68   3rd Qu.:    18.9
##  Max.   :570.722   Max.   :672.860   Max.   :1571.79   Max.   :383154.5
##    Mcooking36       Mlearning37       MGamesRiddles38     Msports39
##  Min.   :  0.000   Min.   :     0.0   Min.   : 0.0000   Min.   : 0.00000
##  1st Qu.:  0.000   1st Qu.:     0.0   1st Qu.: 0.0000   1st Qu.: 0.00000
##  Median :  0.000   Median :     0.0   Median : 0.0000   Median : 0.00000
##  Mean   :  4.223   Mean   :    21.1   Mean   : 0.3052   Mean   : 0.02068
##  3rd Qu.:  0.000   3rd Qu.:     2.5   3rd Qu.: 0.0000   3rd Qu.: 0.00000
##  Max.   :820.563   Max.   :532182.0   Max.   :78.7247   Max.   :21.44869
##     Mhobby40         Mnature41       Mencyclopaedia44     Mvideos50
##  Min.   :  0.000   Min.   :  0.000   Min.   :   0.000   Min.   :  0.00
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:   0.000   1st Qu.:  0.00
##  Median :  0.000   Median :  0.000   Median :   0.000   Median :  0.00
##  Mean   :  6.786   Mean   :  3.729   Mean   :   8.402   Mean   :  6.02
##  3rd Qu.:  5.113   3rd Qu.:  0.000   3rd Qu.:   0.000   3rd Qu.:  0.00
##  Max.   :771.611   Max.   :735.355   Max.   :1999.150   Max.   :794.14
##    Mnonbooks99
##  Min.   :   0.000
##  1st Qu.:   0.000
##  Median :   0.000
##  Mean   :   1.801
##  3rd Qu.:   0.000
##  Max.   :2412.961
```

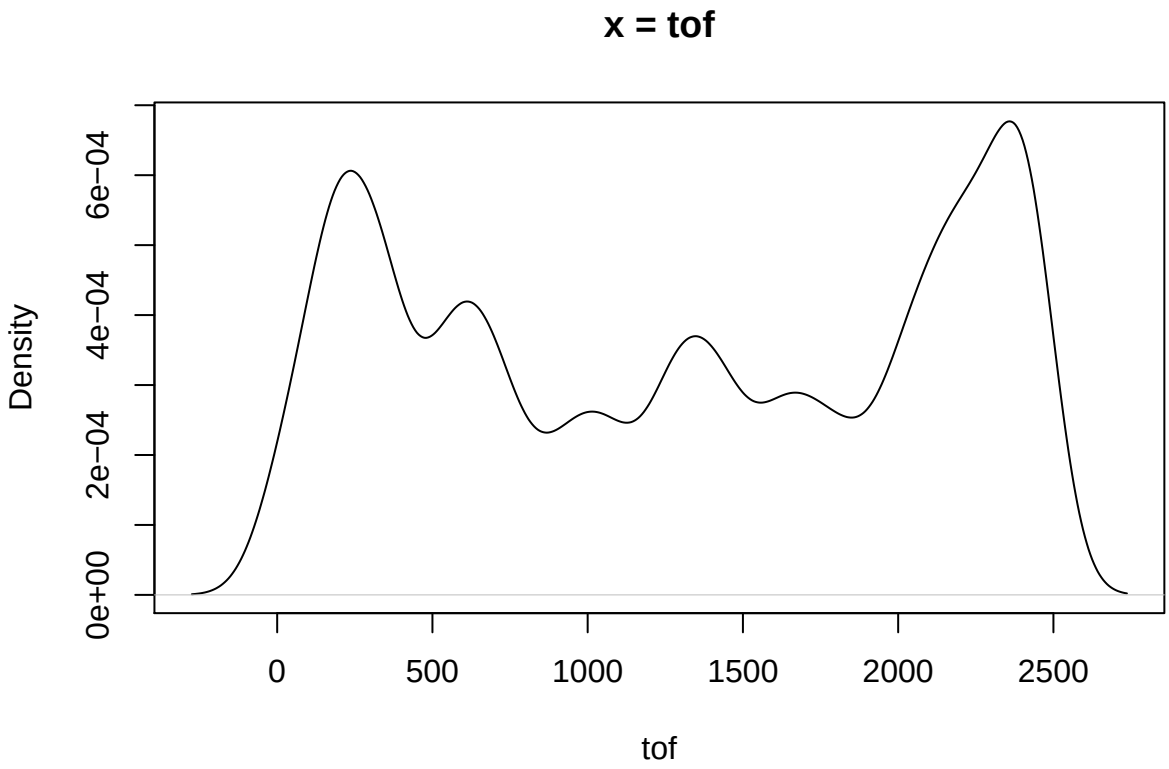
```
# find the outlier and remove the rows
View(df)
```

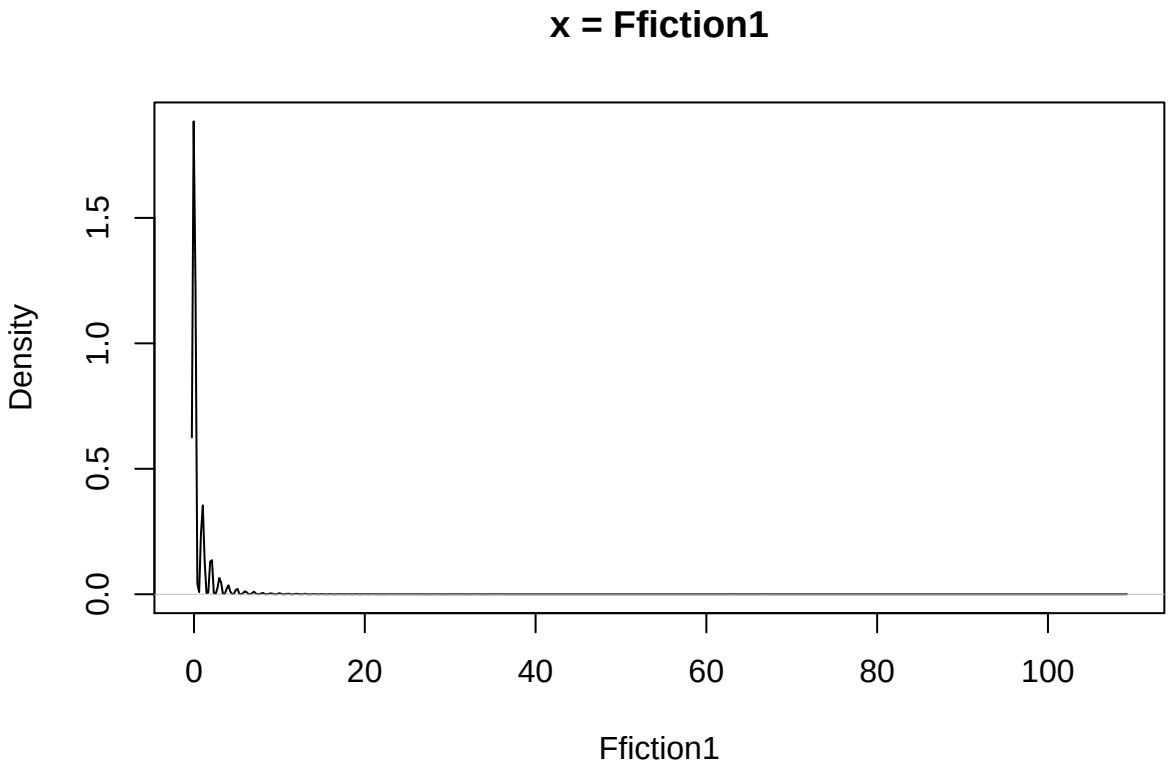
```
for (i in 2:65){
  plot(density(unlist(df[,i])), main = sprintf("x = %s", xlab = names(df[,i])), xlab = names(df[,i]))
}
```

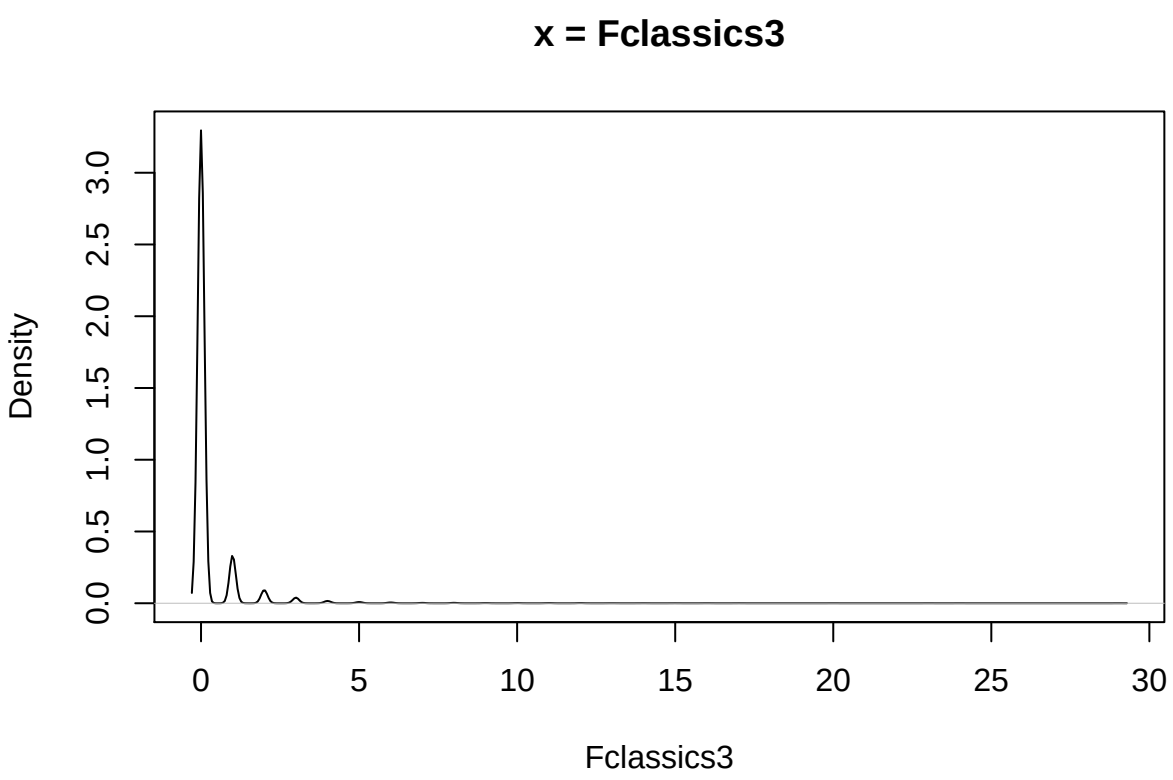

x = r
Density
0.0000
0.0005
0.0010
0.0015
0
500
1000
1500
2000
2500
r

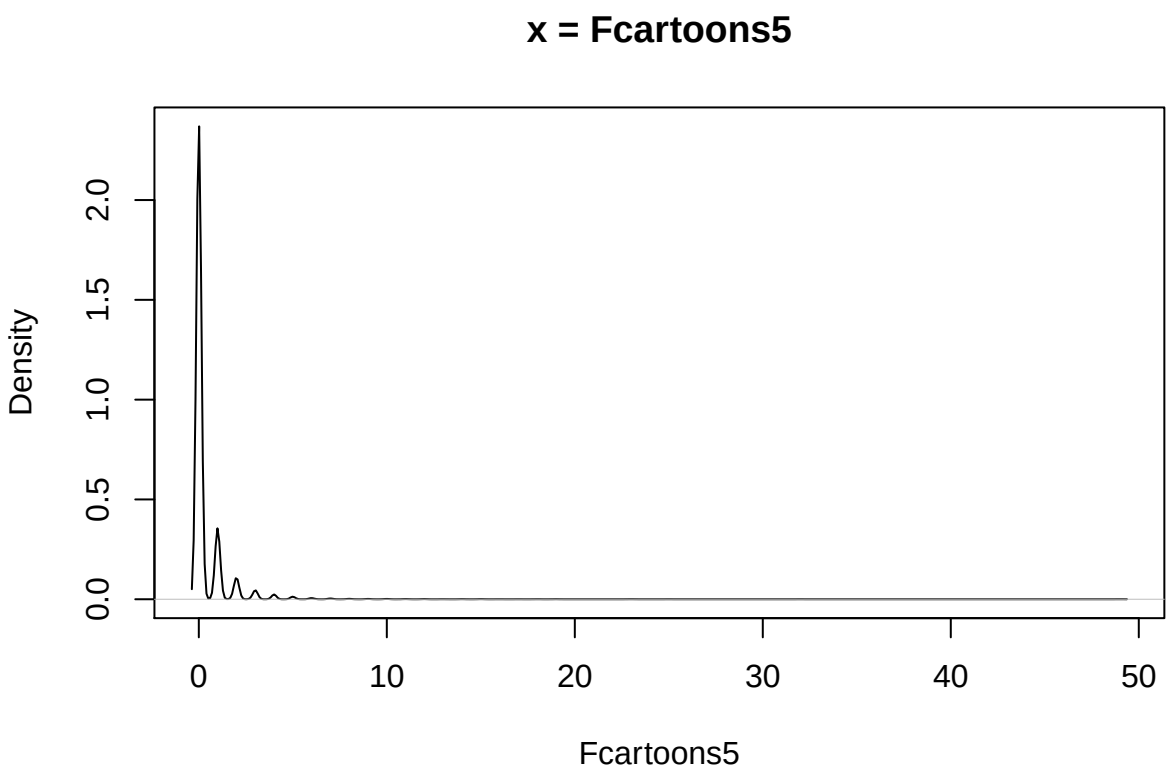

x = f
Density
0.00
0.05
0.10
0.15
0.20
0
20
40
60
80
100
120
f

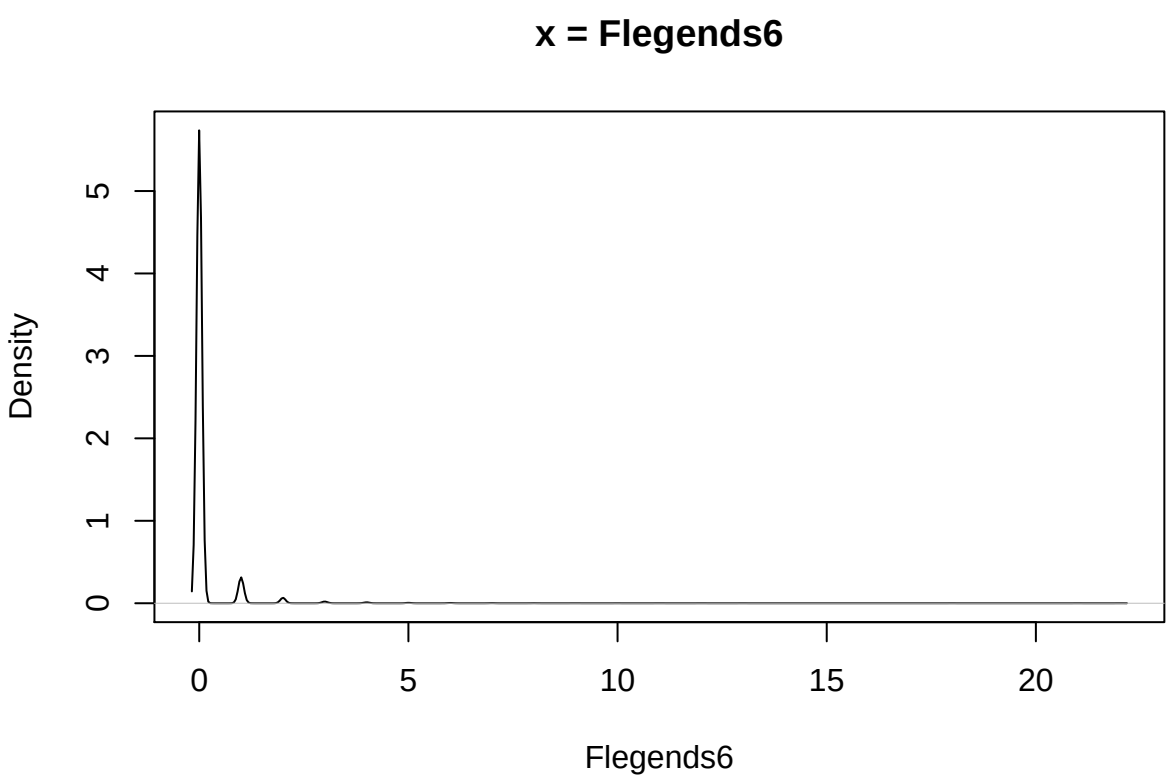

x = m
Density
0.000
0.005
0.010
0.015
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
m

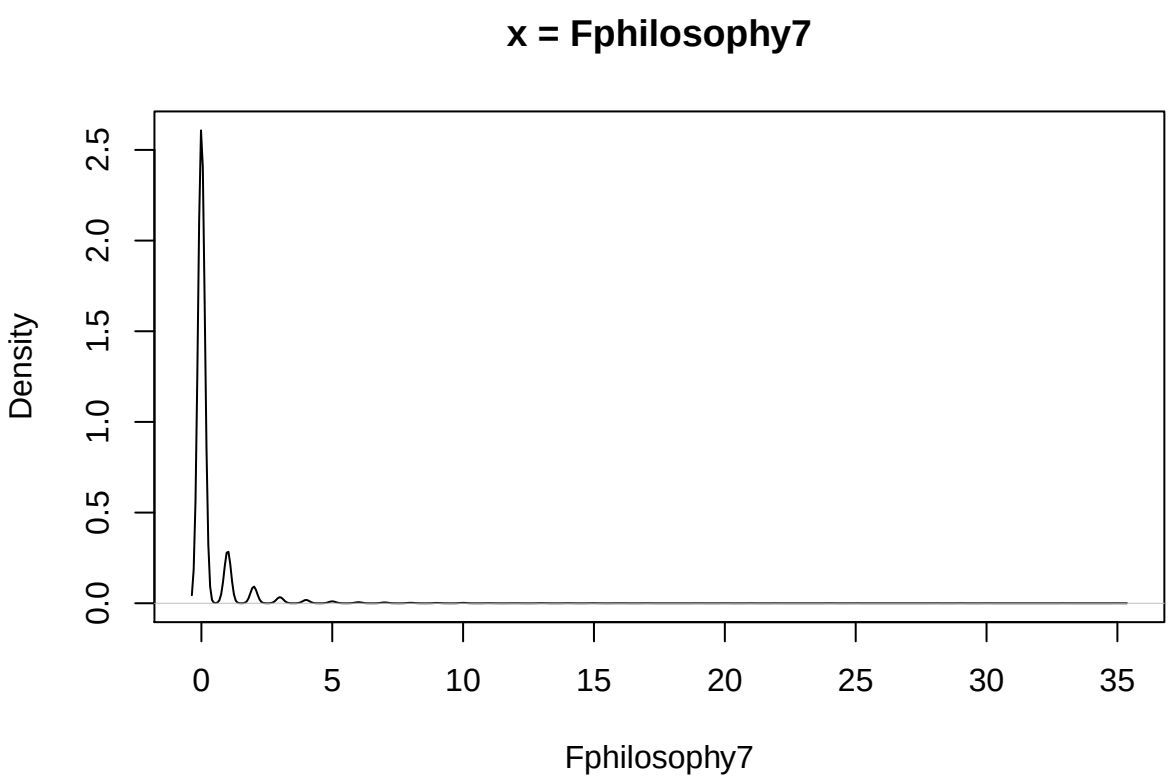

x = tof
Density
0e+00
2e−04
4e−04
6e−04
0
500
1000
1500
2000
2500
tof

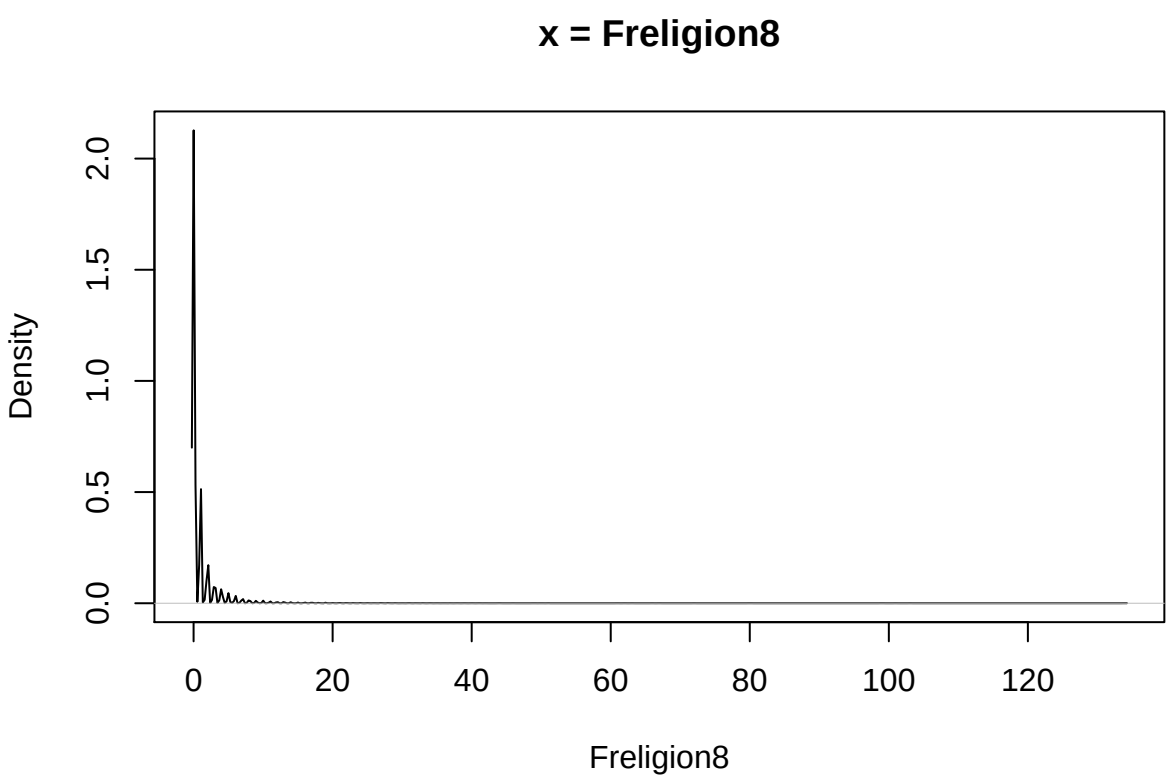

x = Ffiction1
Density
0.0
0.5
1.0
1.5
0
20
40
60
80
100
Ffiction1

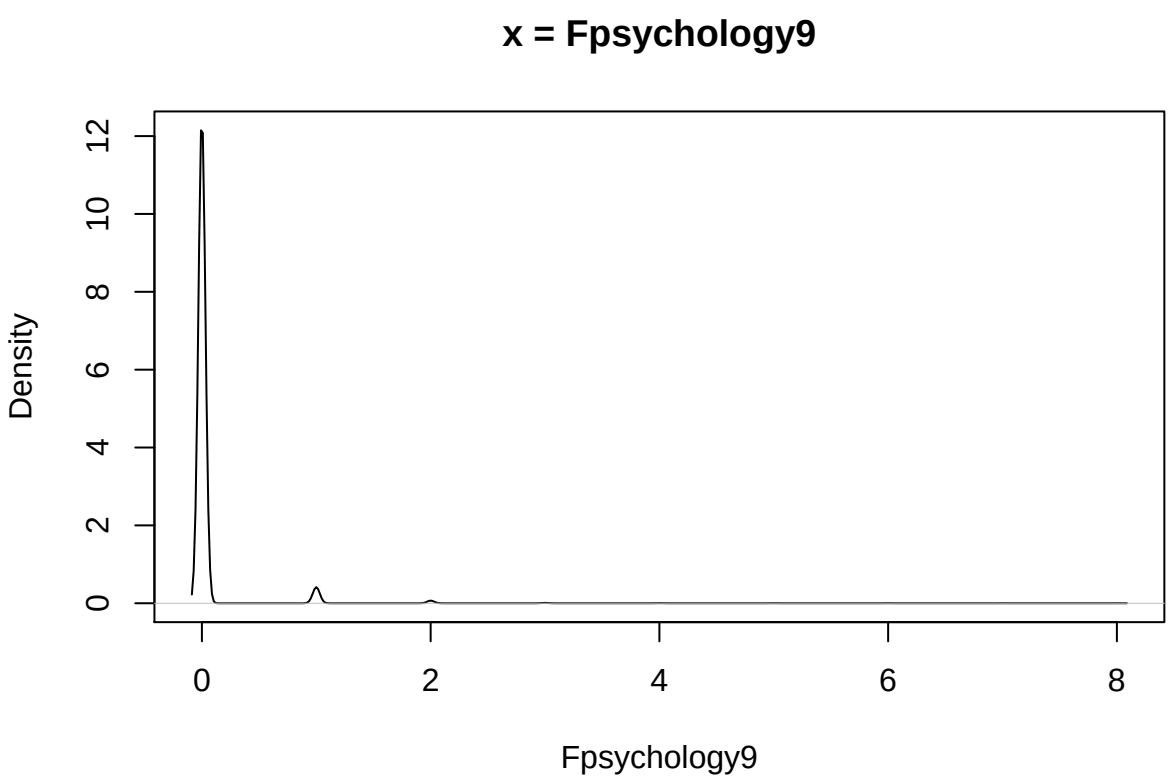

x = Fclassics3
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
5
10
15
20
25
30
Fclassics3

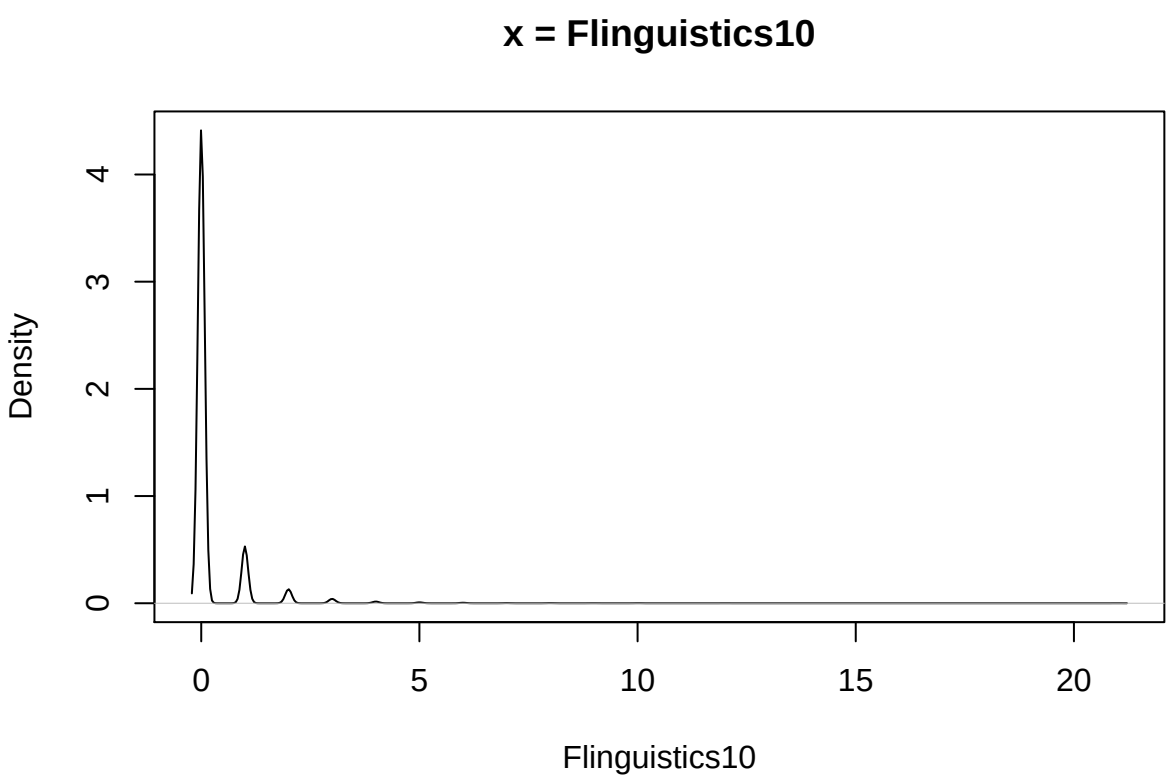

x = Fcartoons5
Density
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
50
Fcartoons5

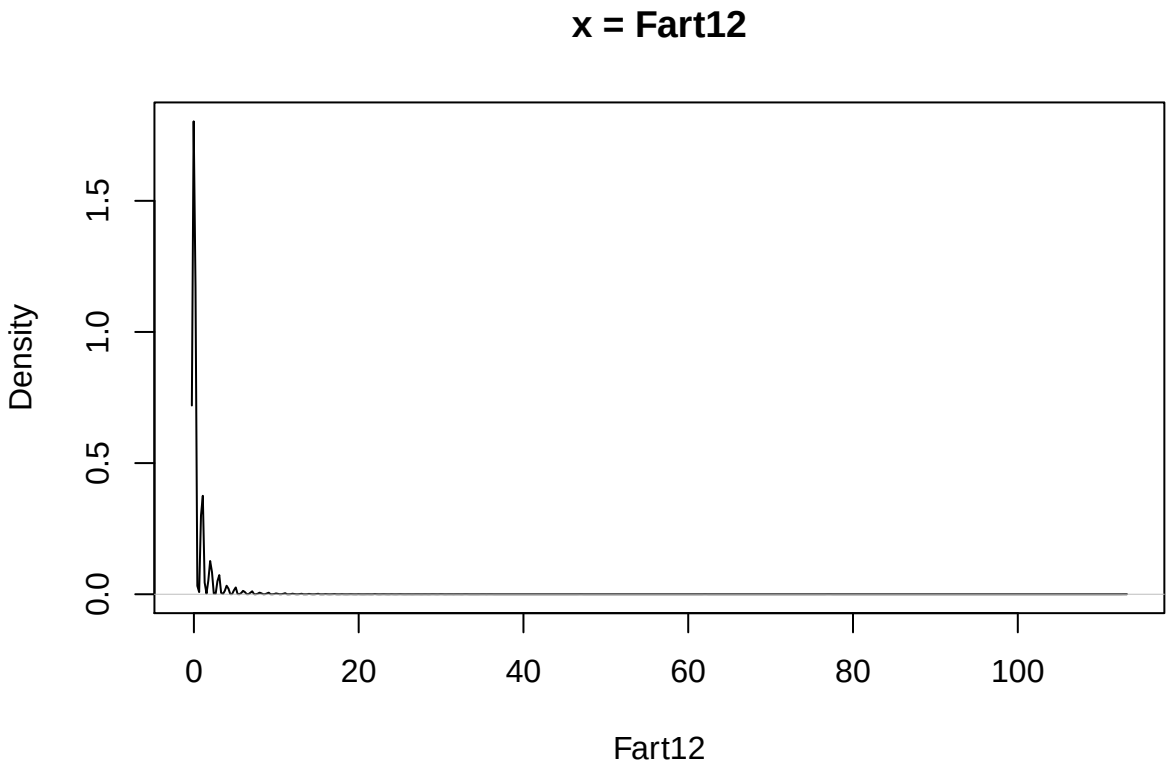

x = Flegends6
Density
0
1
2
3
4
5
0
5
10
15
20
Flegends6

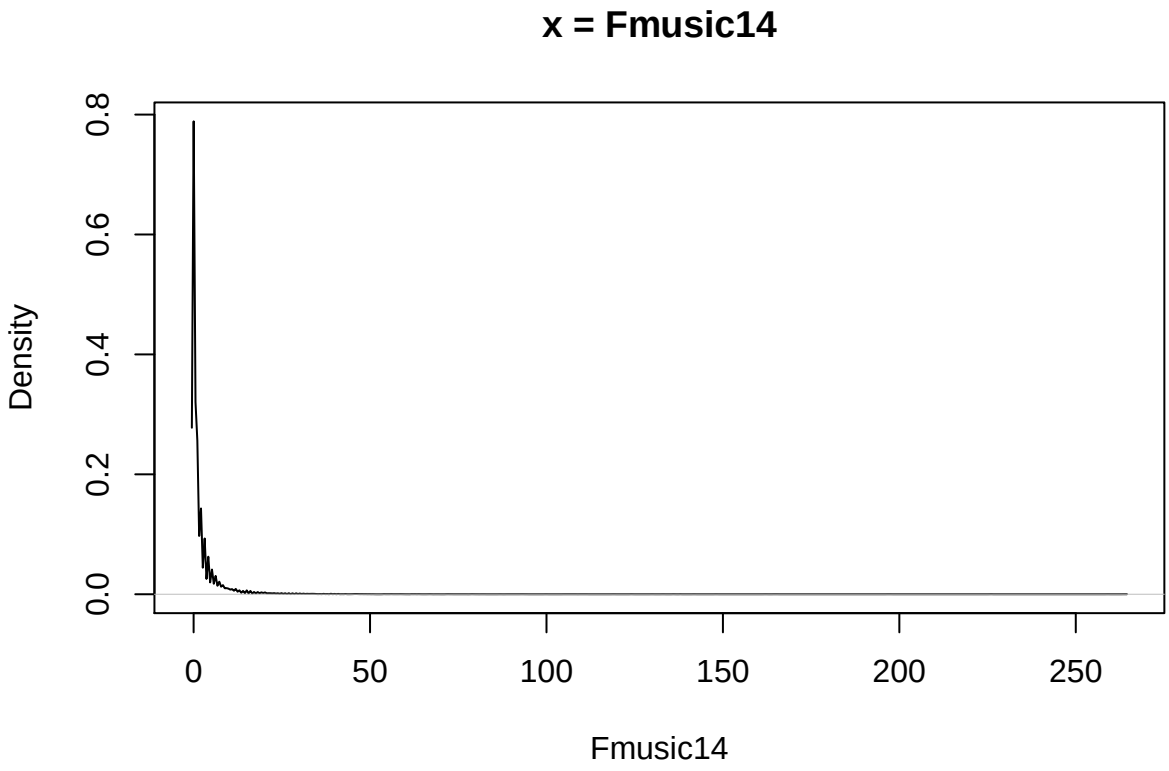

x = Fphilosophy7
Density
0.0
0.5
1.0
1.5
2.0
2.5
0
5
10
15
20
25
30
35
Fphilosophy7

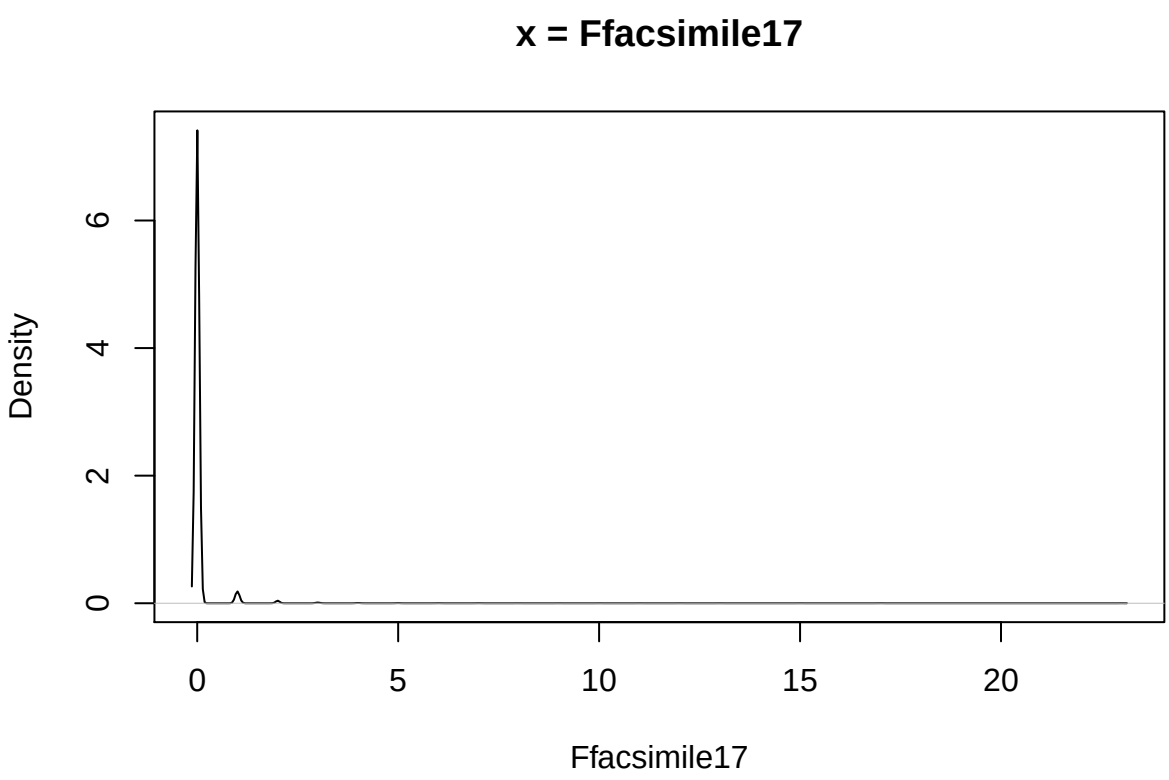

x = Freligion8
Density
0.0
0.5
1.0
1.5
2.0
0
20
40
60
80
100
120
Freligion8

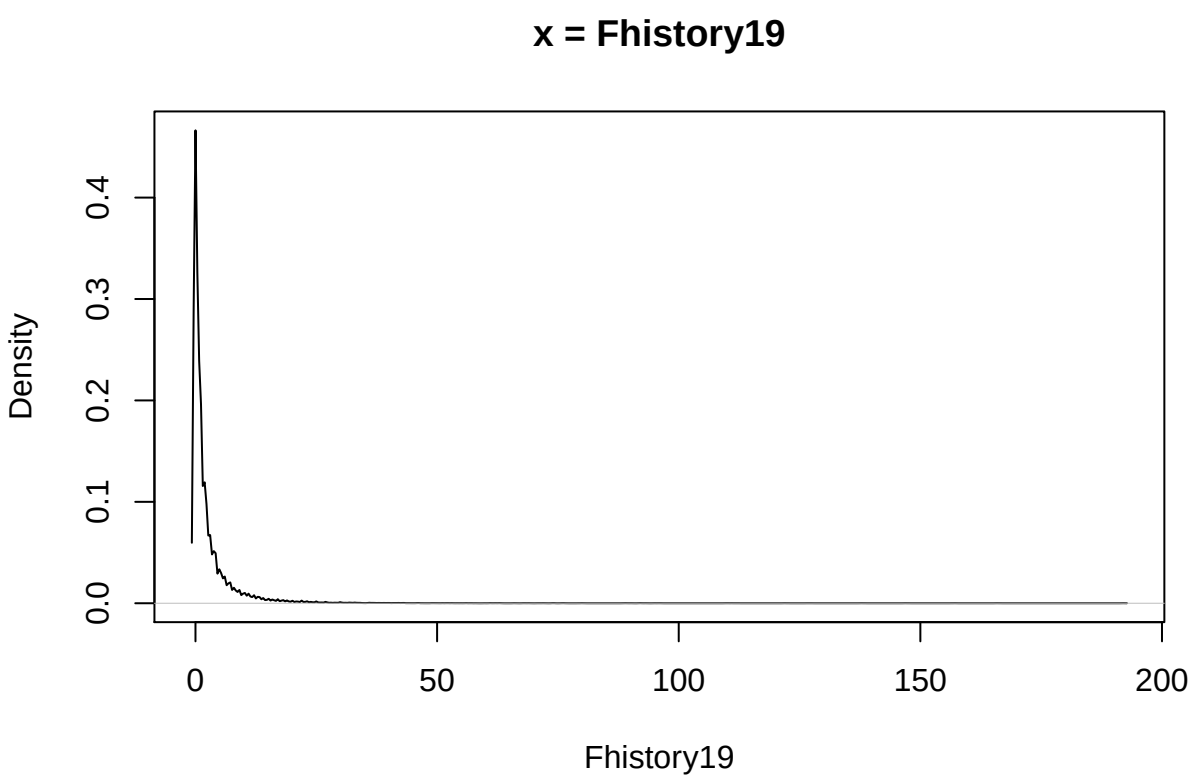

x = Fpsychology9
Density
0
2
4
6
8
10
12
0
2
4
6
8
Fpsychology9

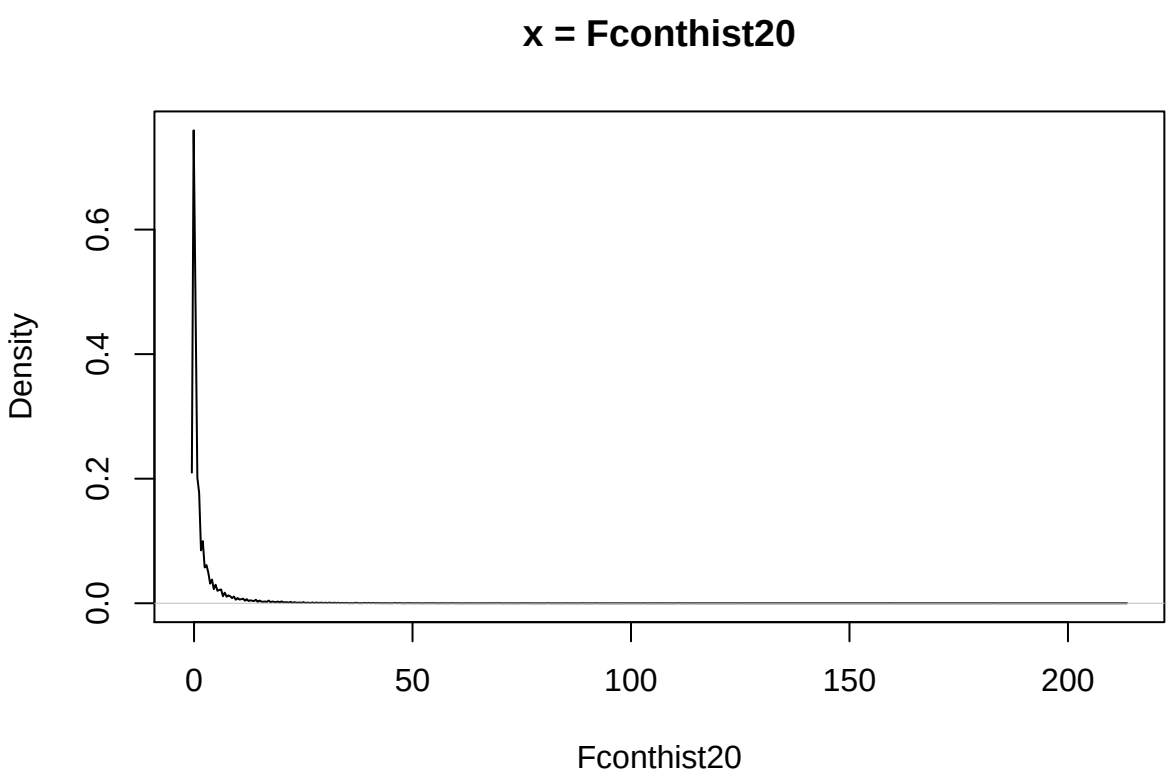

x = Flinguistics10
Density
0
1
2
3
4
0
5
10
15
20
Flinguistics10

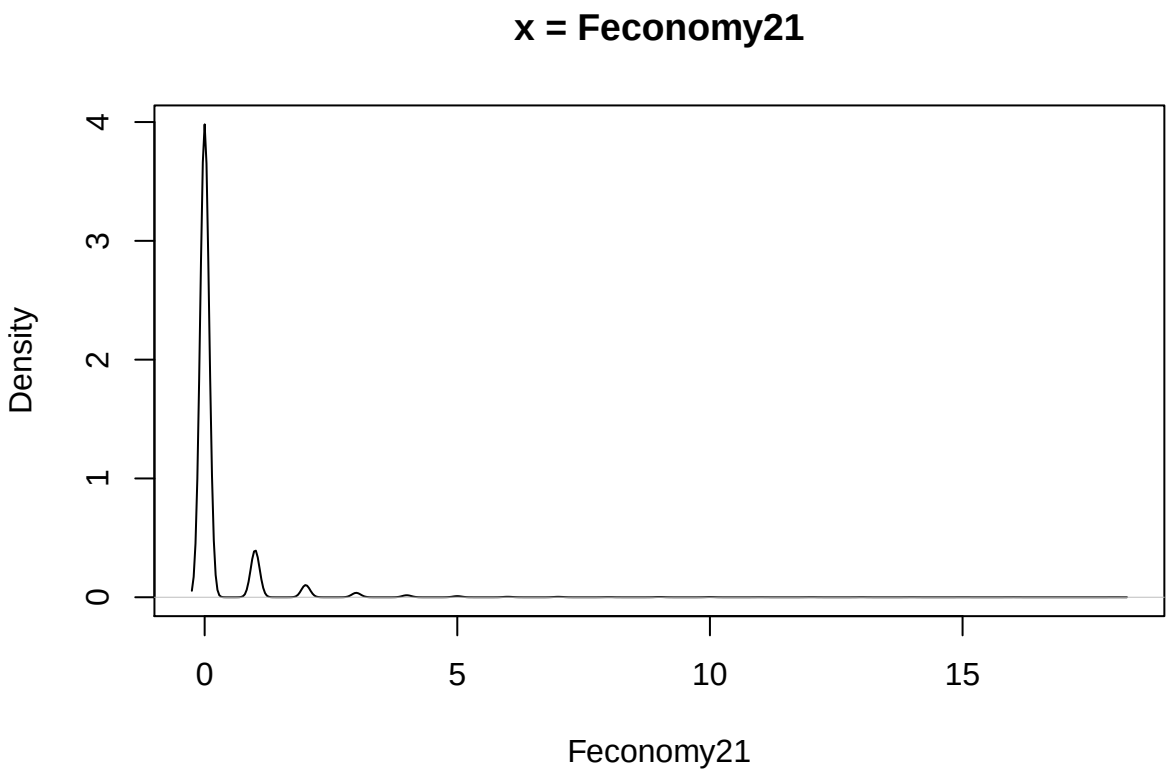

x = Fart12
Density
0.0
0.5
1.0
1.5
0
20
40
60
80
100
Fart12

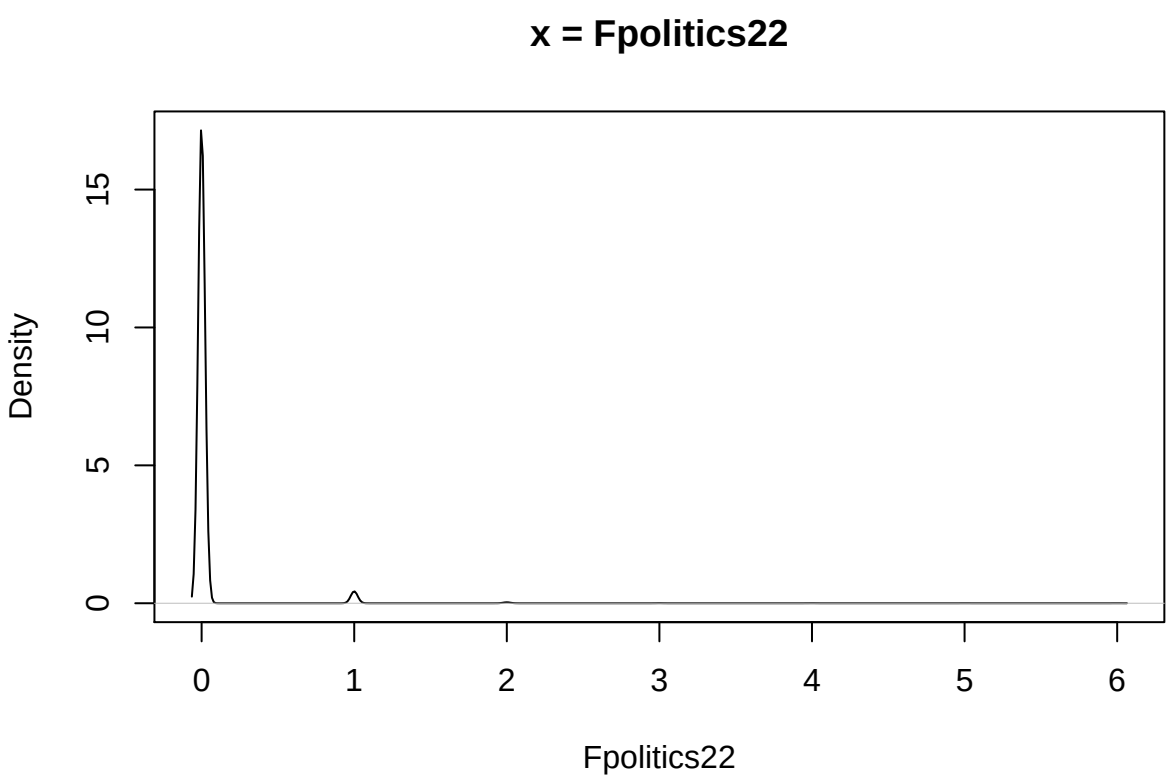

x = Fmusic14
Density
0.0
0.2
0.4
0.6
0.8
0
50
100
150
200
250
Fmusic14

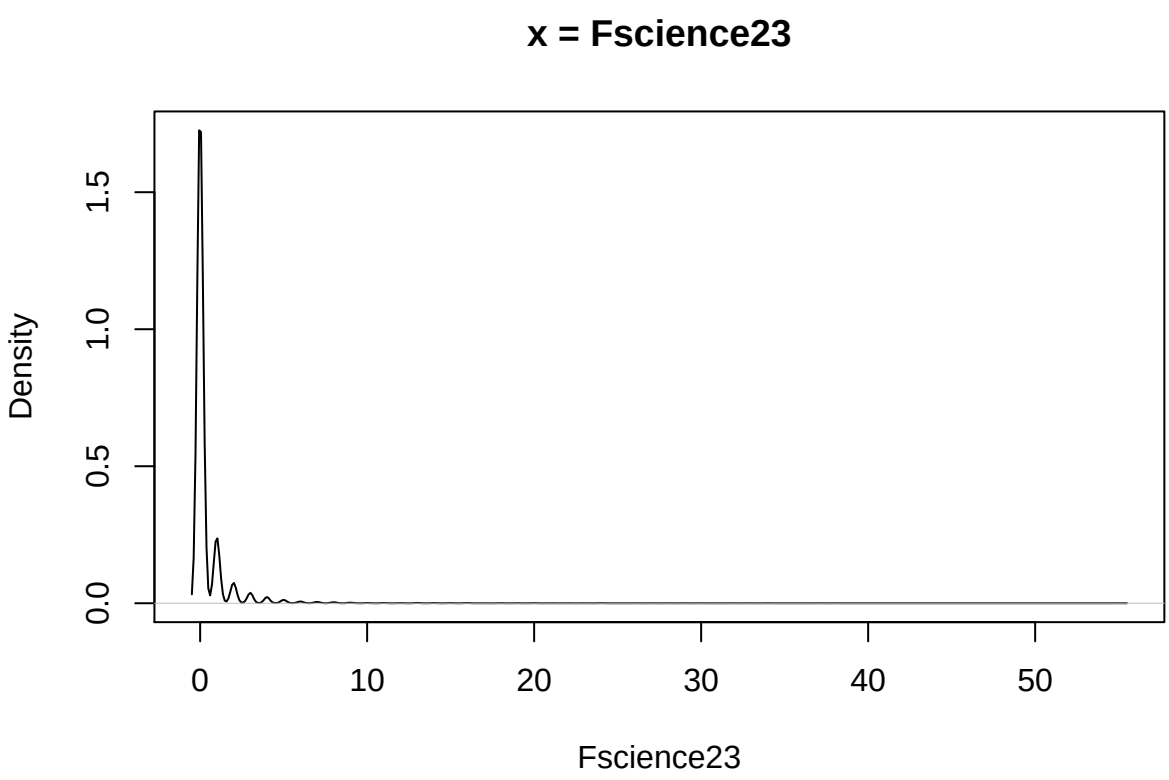

x = Ffacsimile17
Density
0
2
4
6
0
5
10
15
20
Ffacsimile17

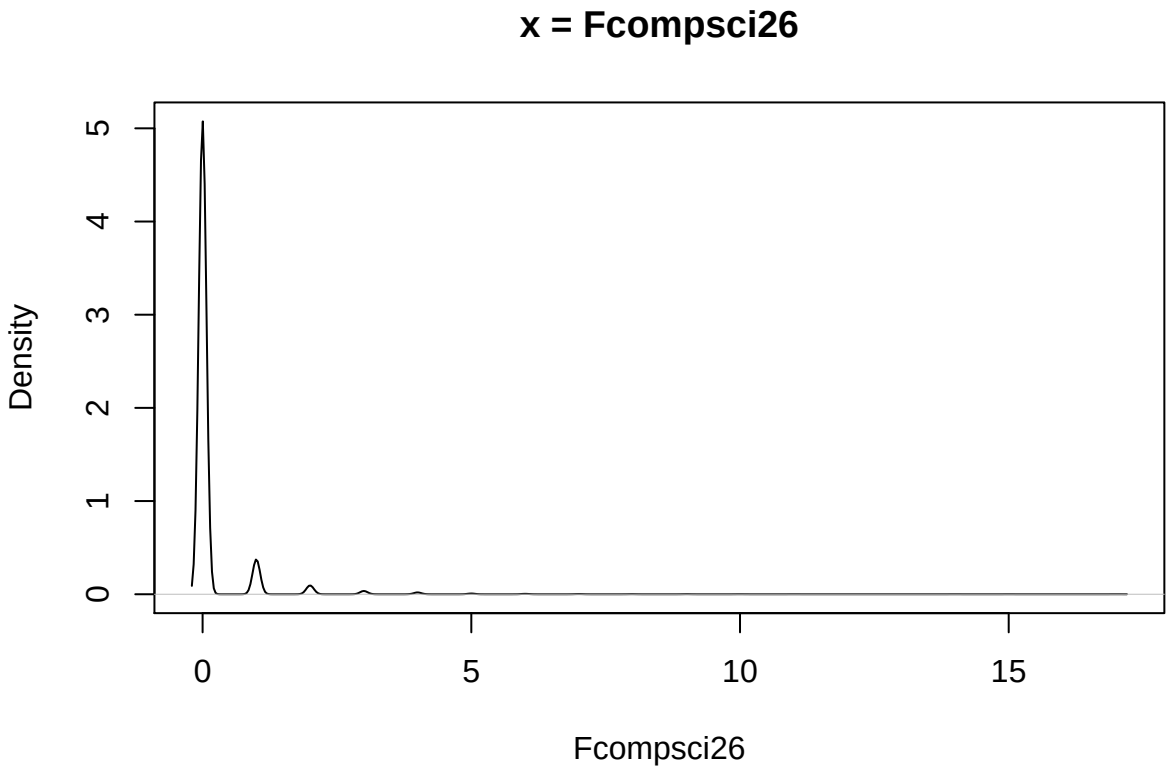

x = Fhistory19
Density
0.0
0.1
0.2
0.3
0.4
0
50
100
150
200
Fhistory19

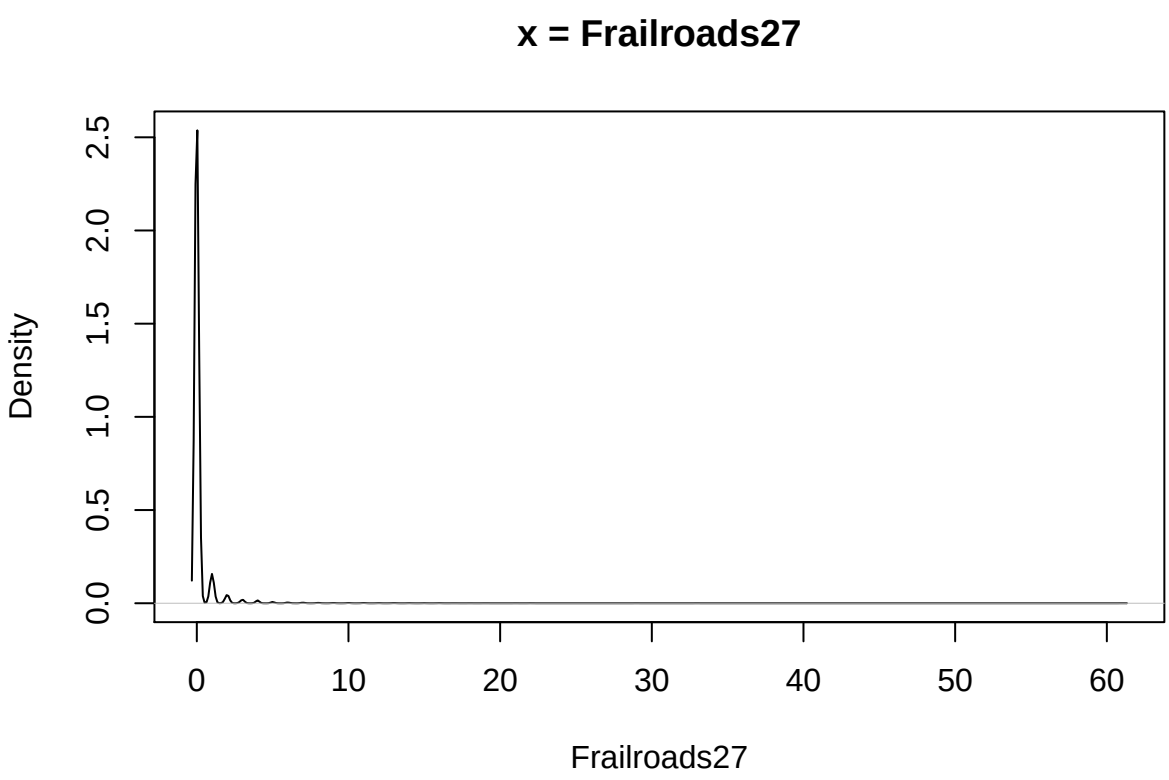

x = Fconthist20
Density
0.0
0.2
0.4
0.6
0
50
100
150
200
Fconthist20

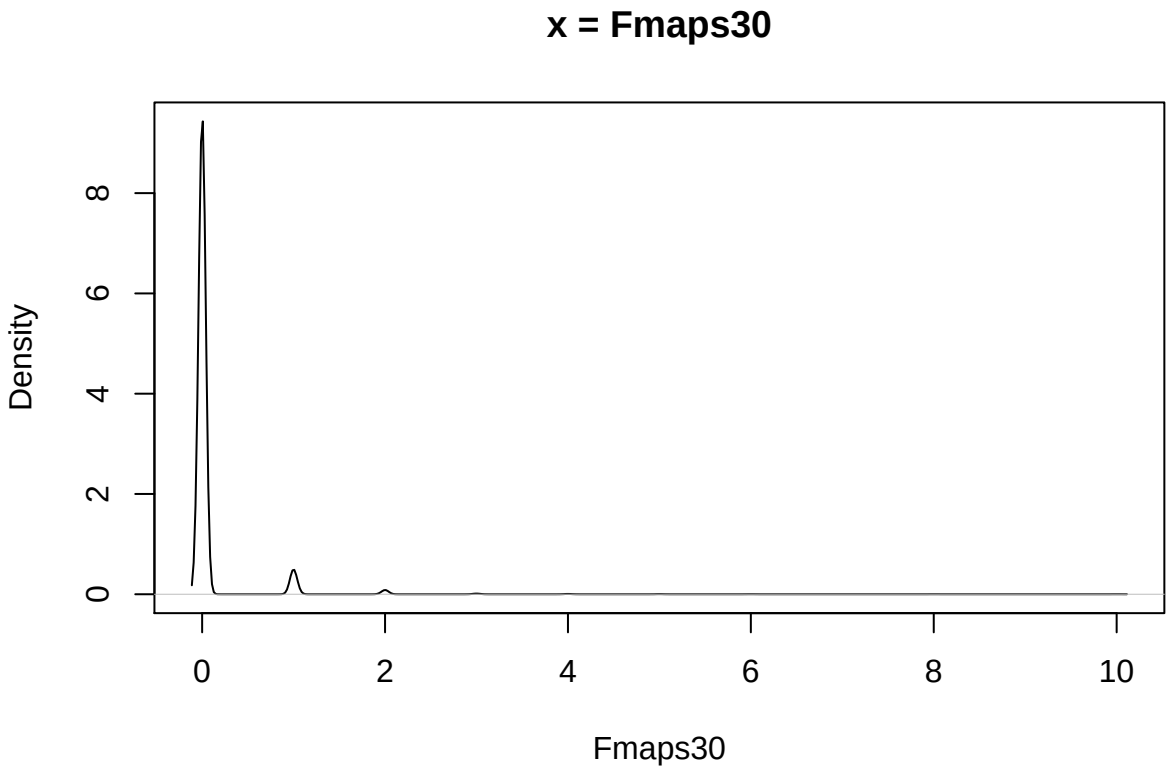

x = Feconomy21
Density
0
1
2
3
4
0
5
10
15
Feconomy21

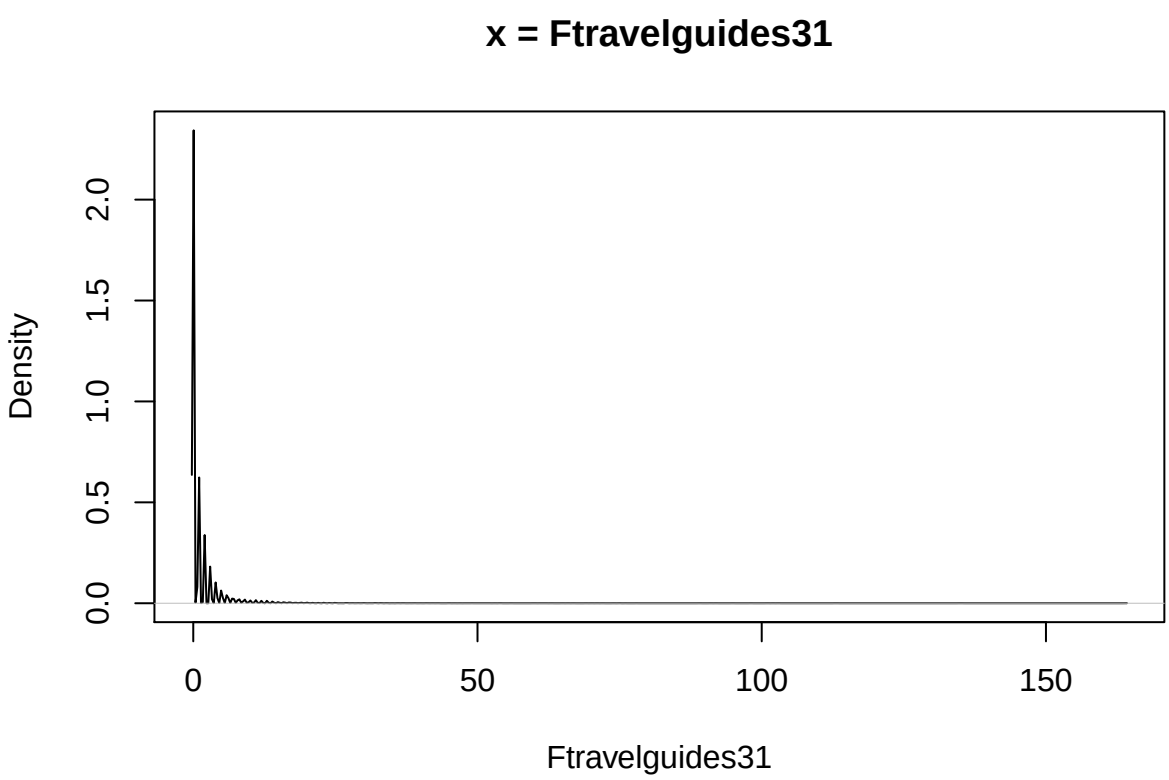

x = Fpolitics22
Density
0
5
10
15
0
1
2
3
4
5
6
Fpolitics22

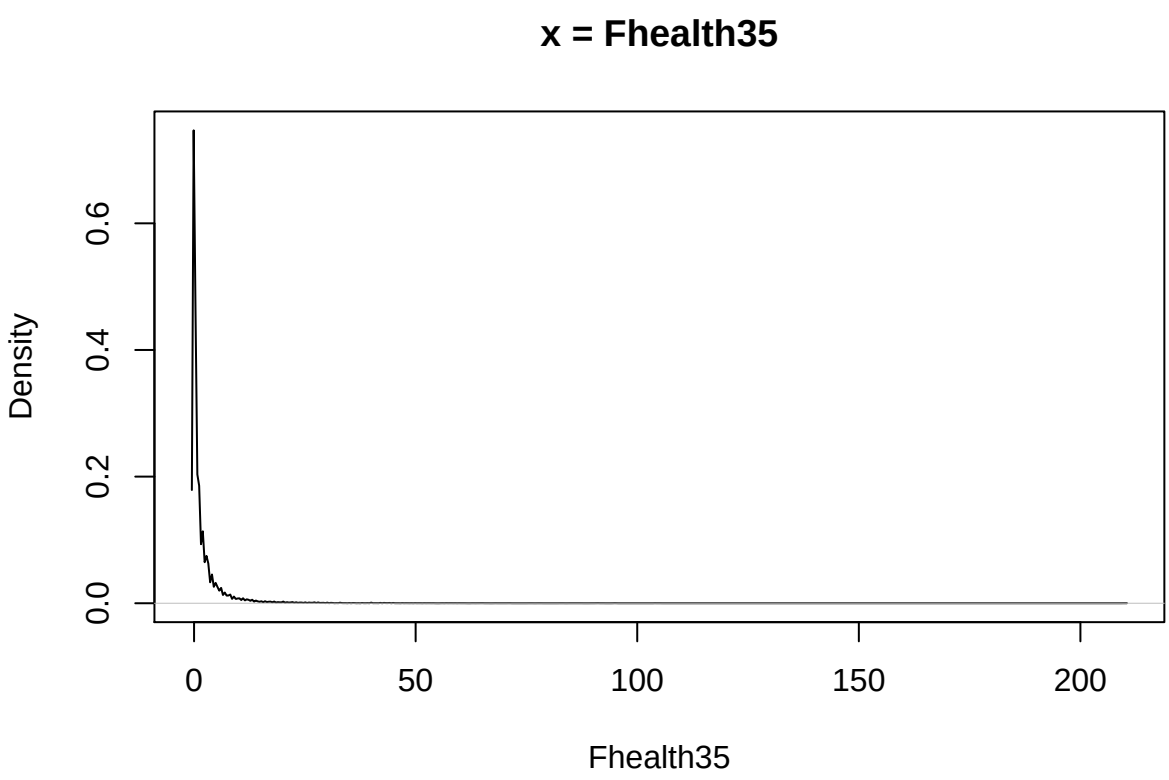

x = Fscience23
Density
0.0
0.5
1.0
1.5
0
10
20
30
40
50
Fscience23

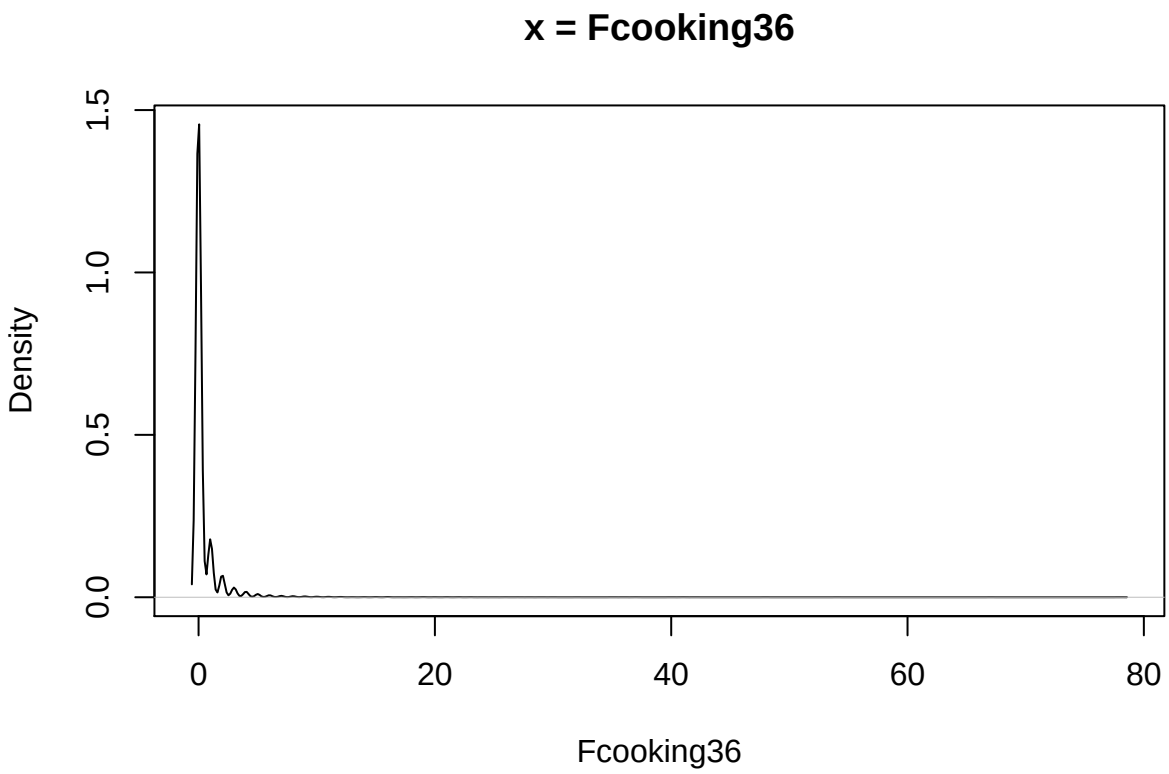

x = Fcompsci26
Density
0
1
2
3
4
5
0
5
10
15
Fcompsci26

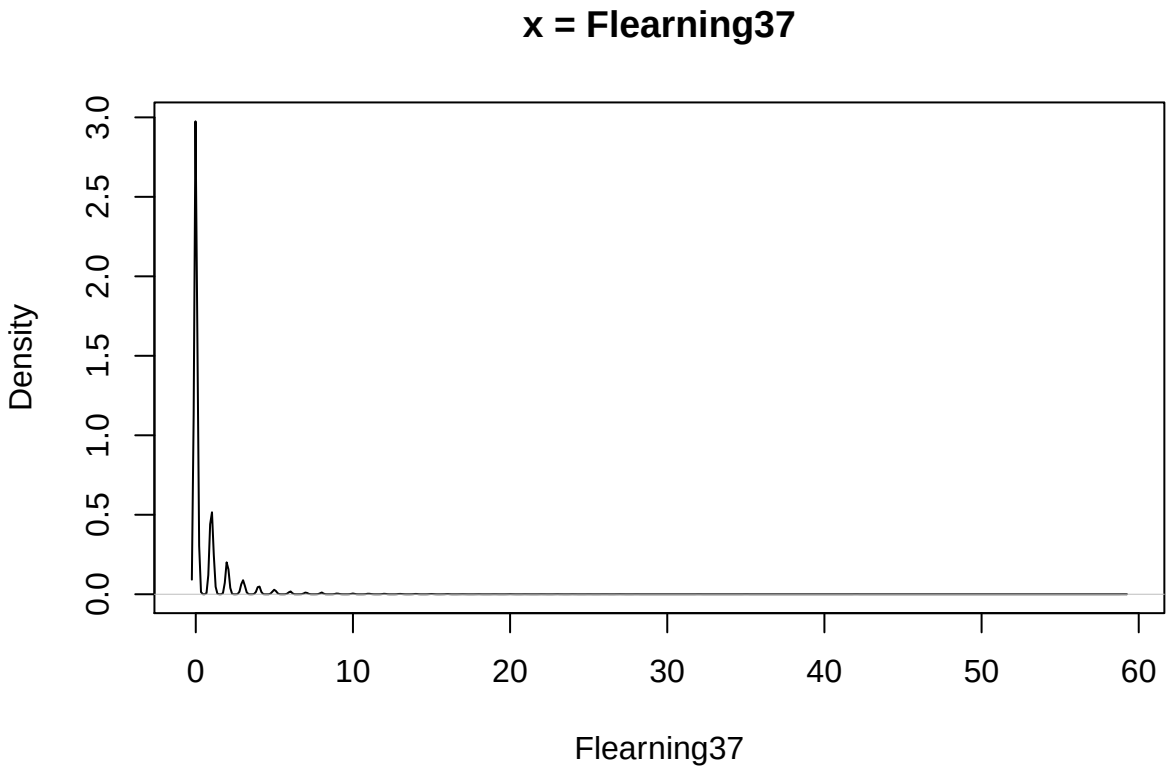

x = Frailroads27
Density
0.0
0.5
1.0
1.5
2.0
2.5
0
10
20
30
40
50
60
Frailroads27

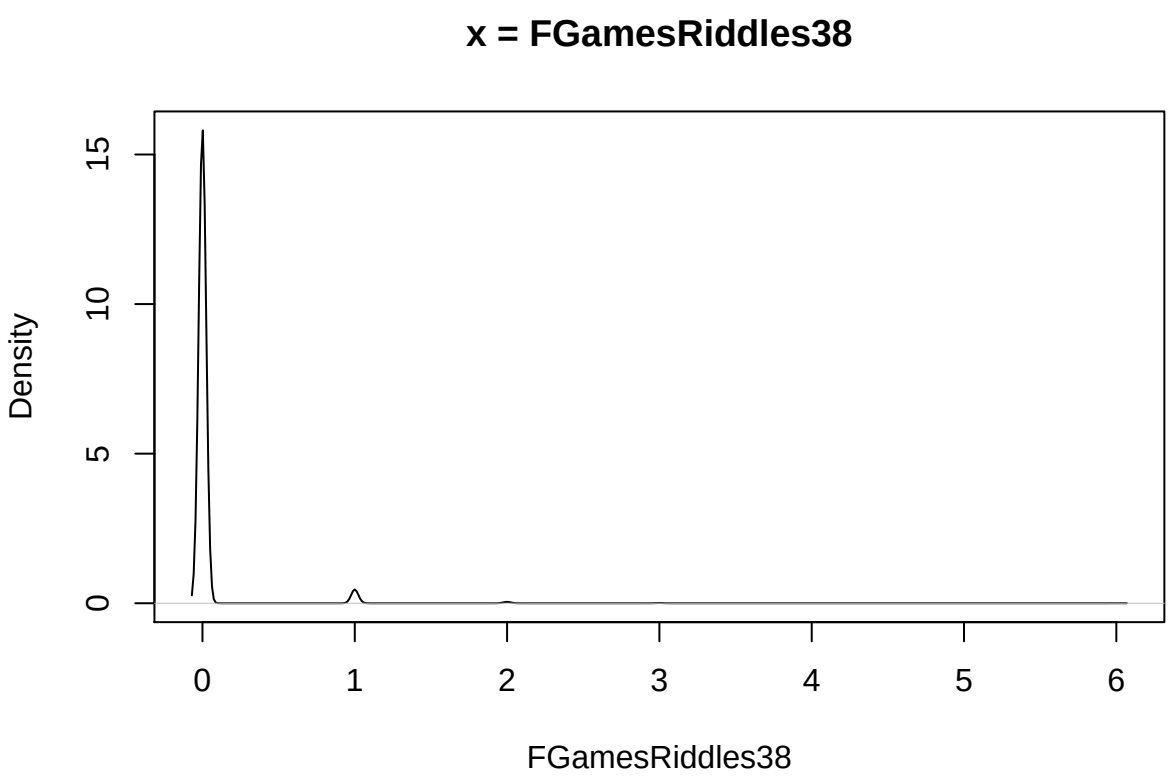

x = Fmaps30
Density
0
2
4
6
8
0
2
4
6
8
10
Fmaps30

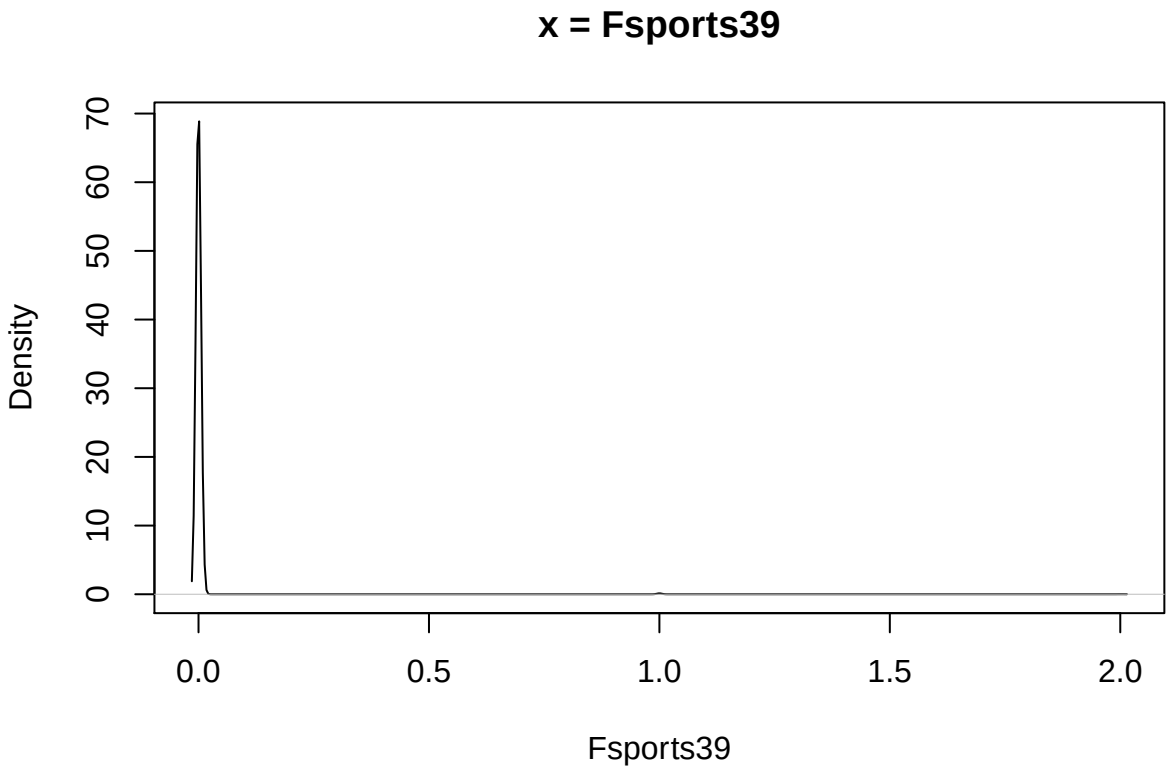

x = Ftravelguides31
Density
0.0
0.5
1.0
1.5
2.0
0
50
100
150
Ftravelguides31

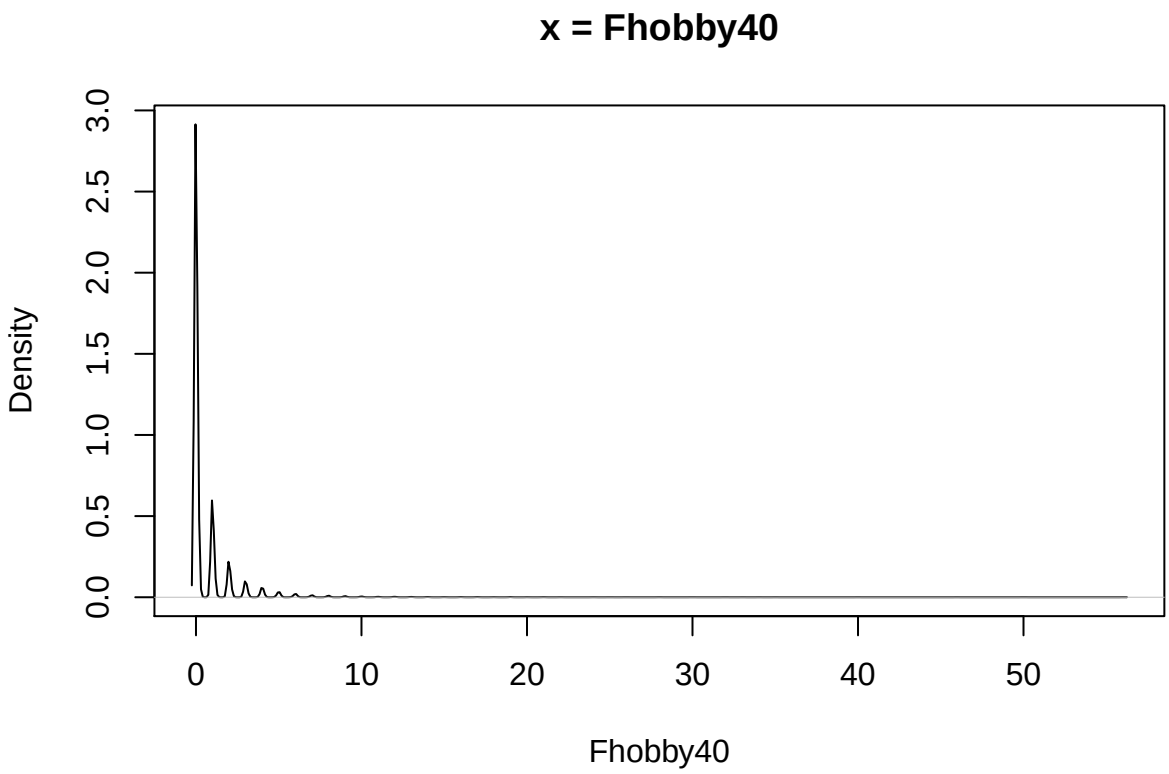

x = Fhealth35
Density
0.0
0.2
0.4
0.6
0
50
100
150
200
Fhealth35

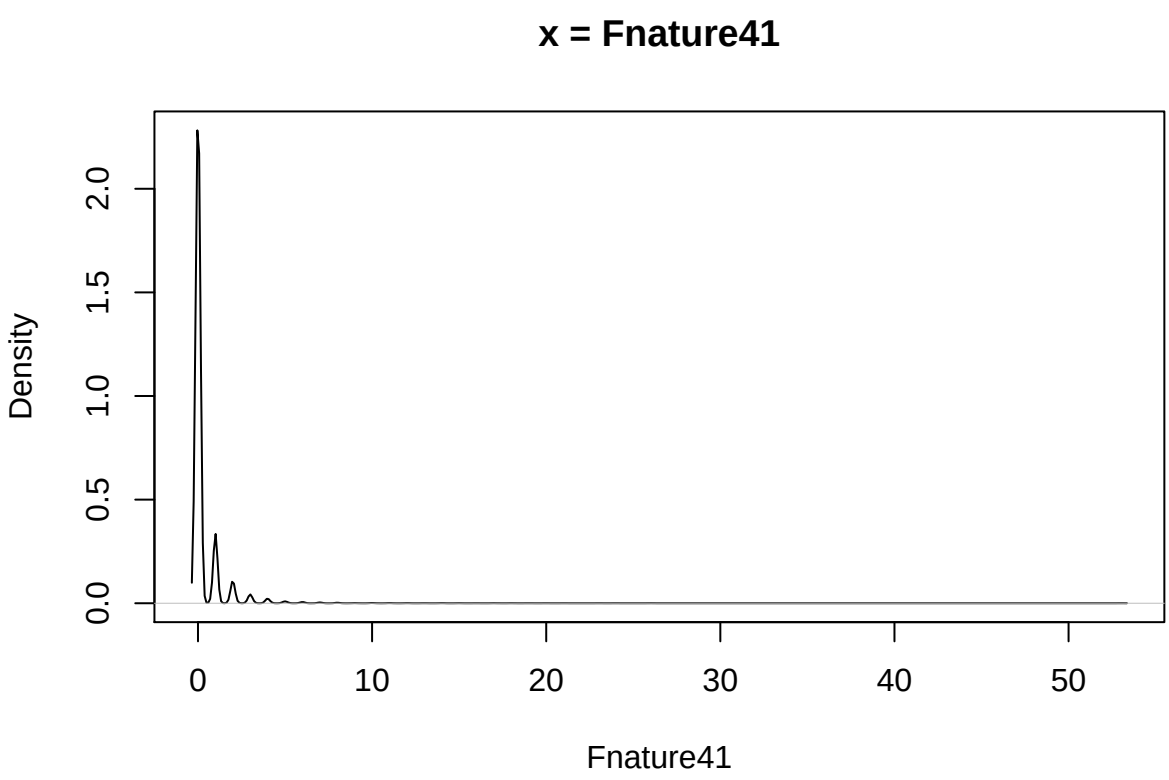

x = Fcooking36
Density
0.0
0.5
1.0
1.5
0
20
40
60
80
Fcooking36

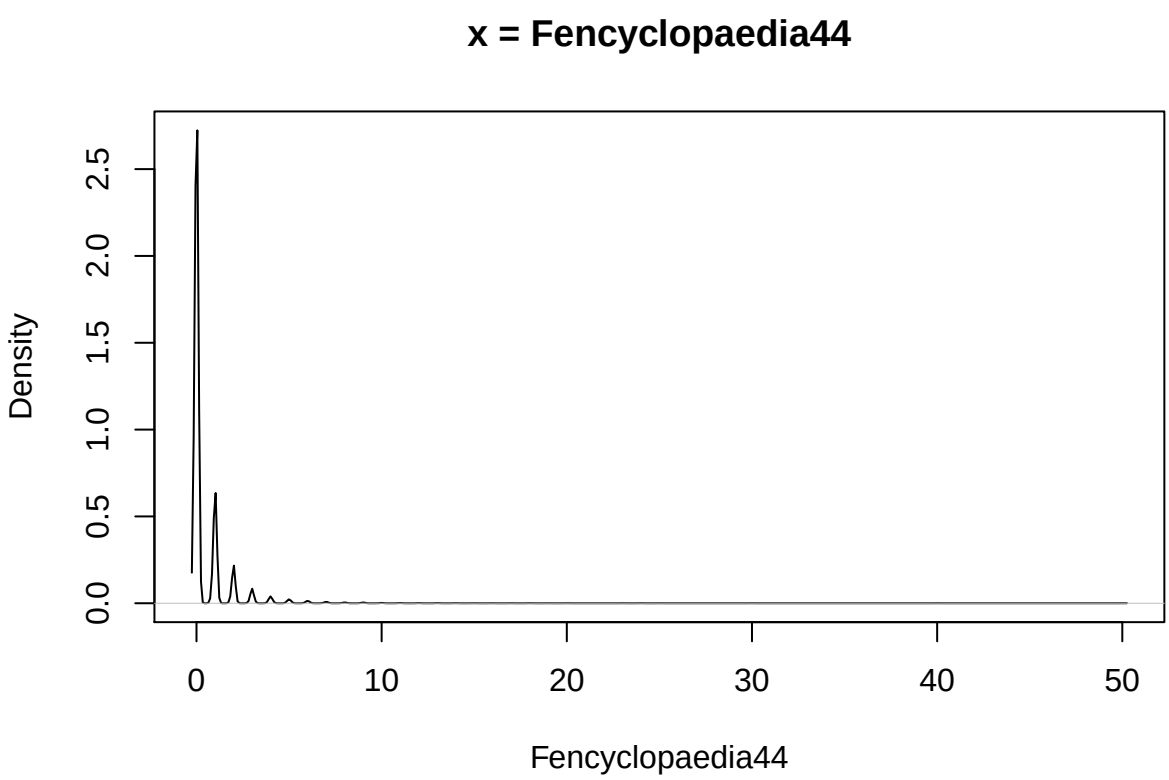

x = Flearning37
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
10
20
30
40
50
60
Flearning37

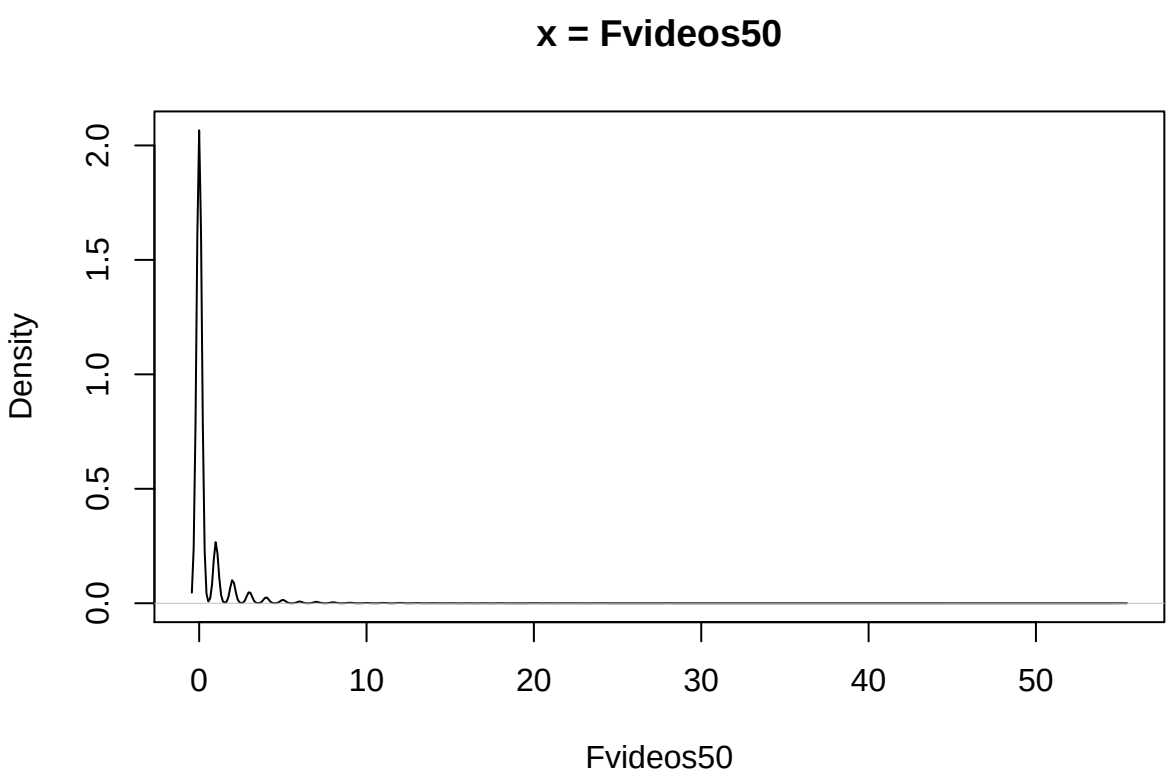

x = FGamesRiddles38
Density
0
5
10
15
0
1
2
3
4
5
6
FGamesRiddles38

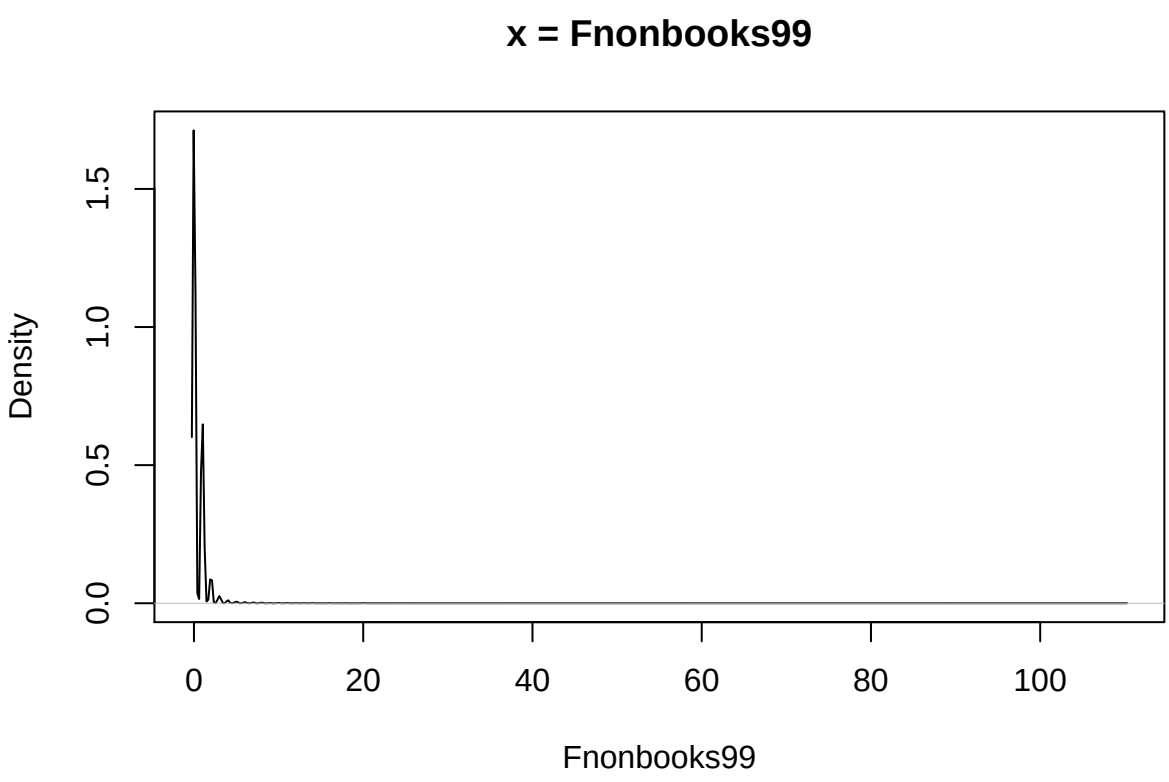

x = Fsports39
Density
0
10
20
30
40
50
60
70
0.0
0.5
1.0
1.5
2.0
Fsports39

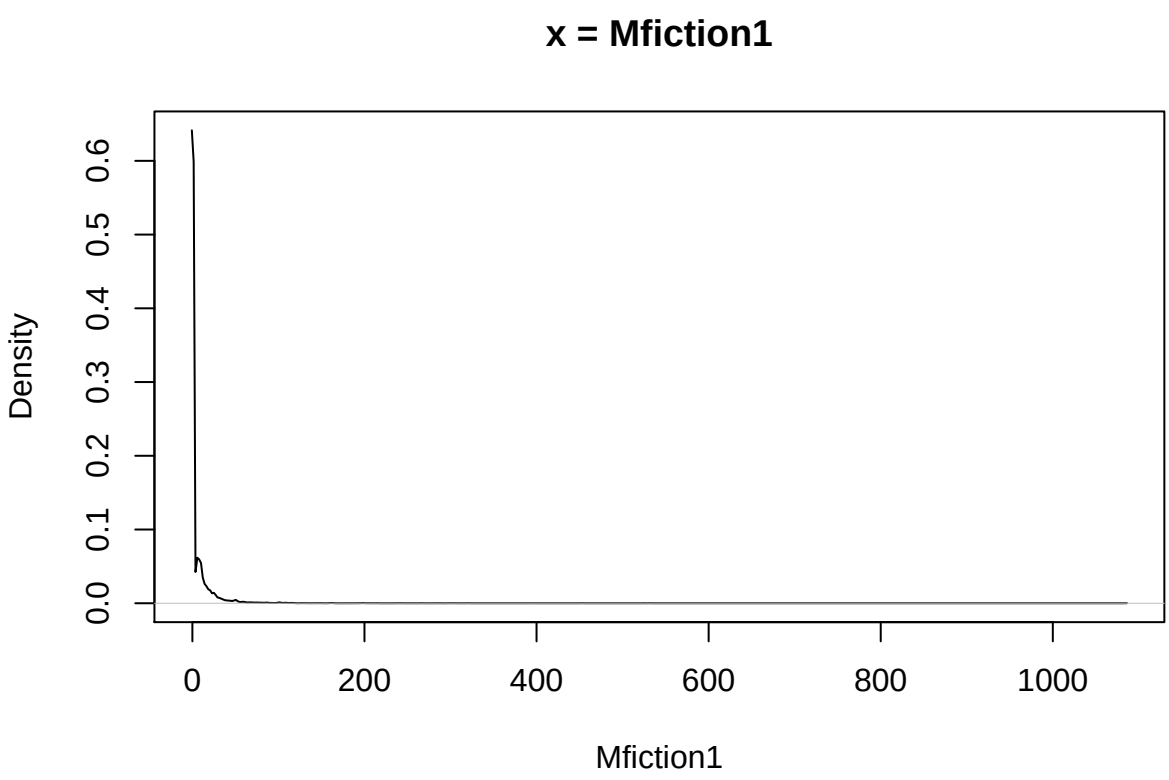

x = Fhobby40
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
10
20
30
40
50
Fhobby40

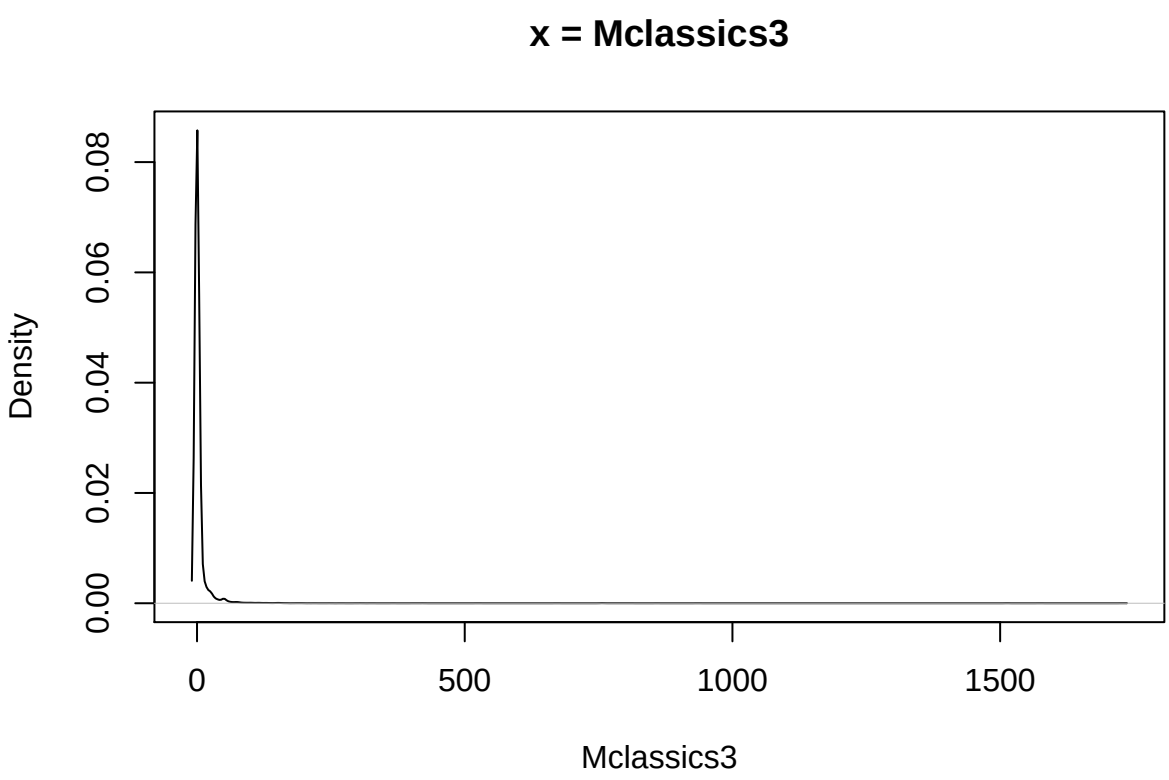

x = Fnature41
Density
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
50
Fnature41

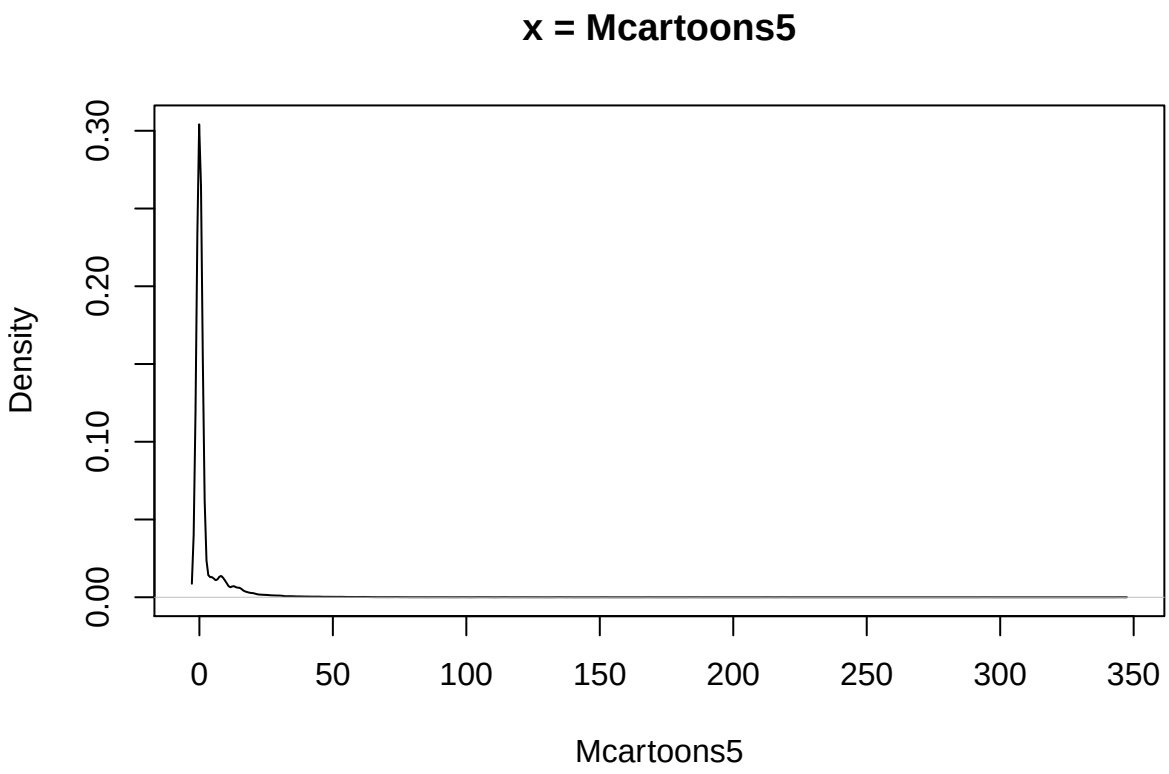

x = Fencyclopaedia44
Density
0.0
0.5
1.0
1.5
2.0
2.5
0
10
20
30
40
50
Fencyclopaedia44

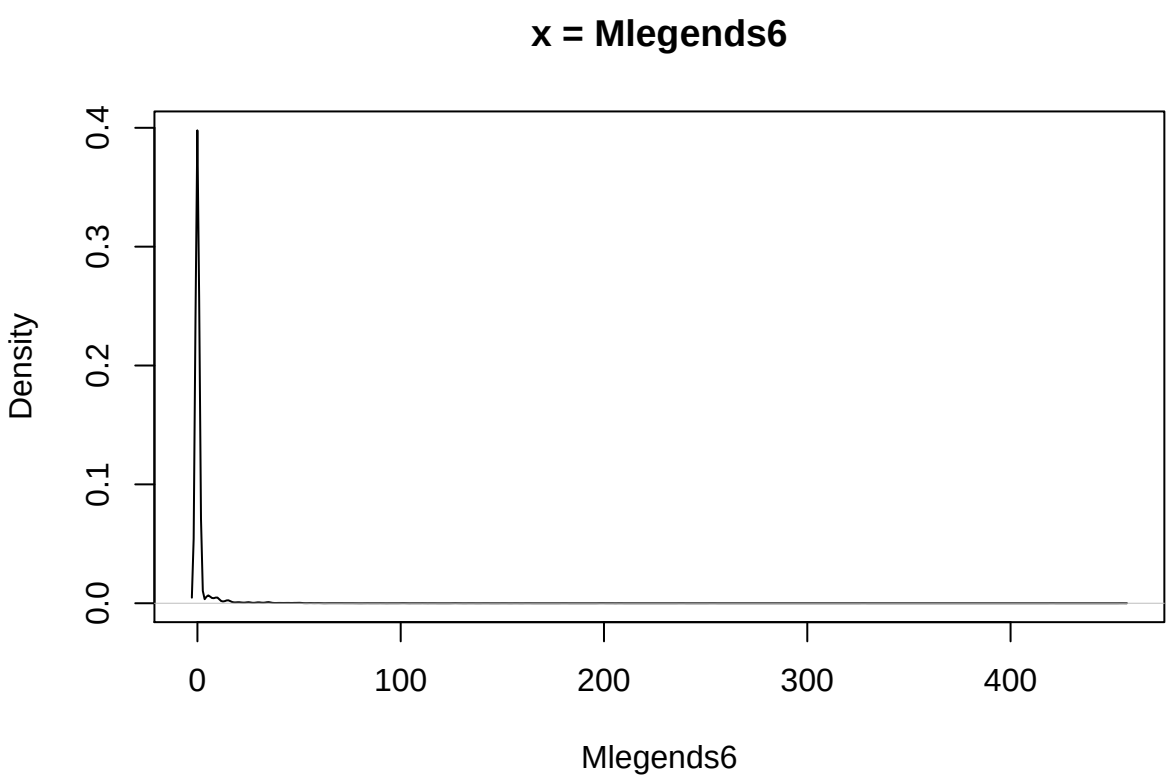

x = Fvideos50
Density
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
50
Fvideos50

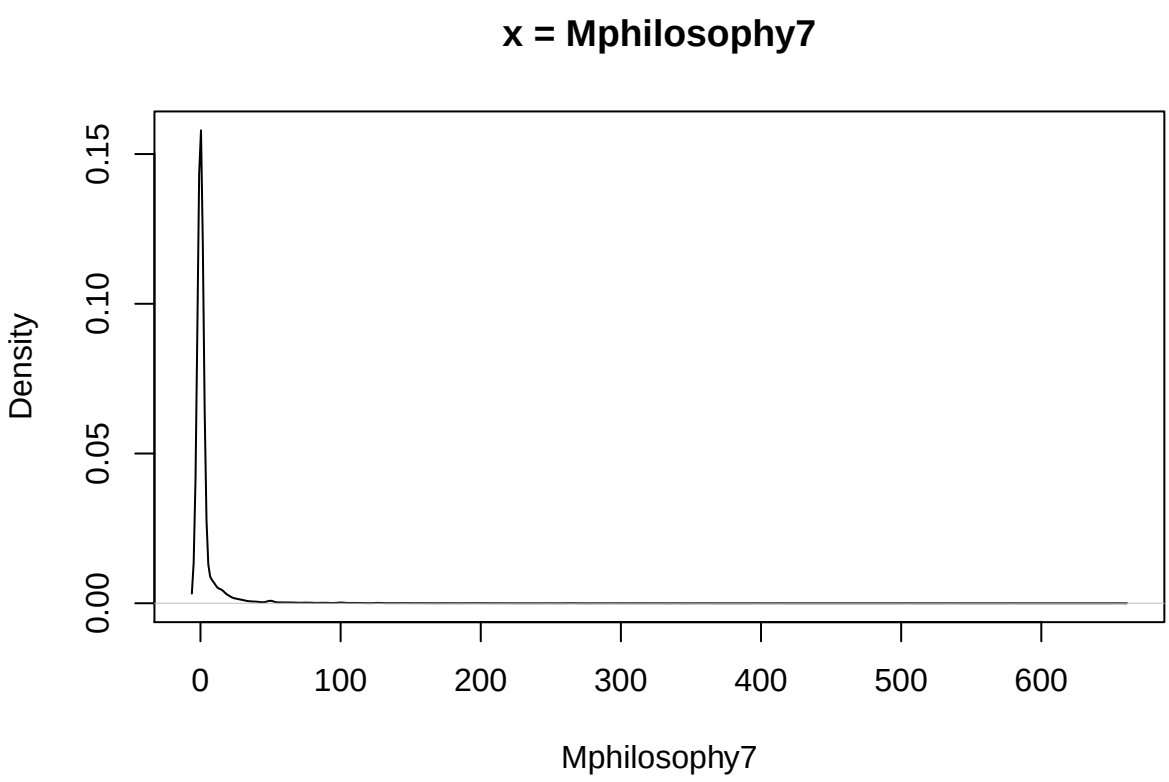

x = Fnonbooks99
Density
0.0
0.5
1.0
1.5
0
20
40
60
80
100
Fnonbooks99

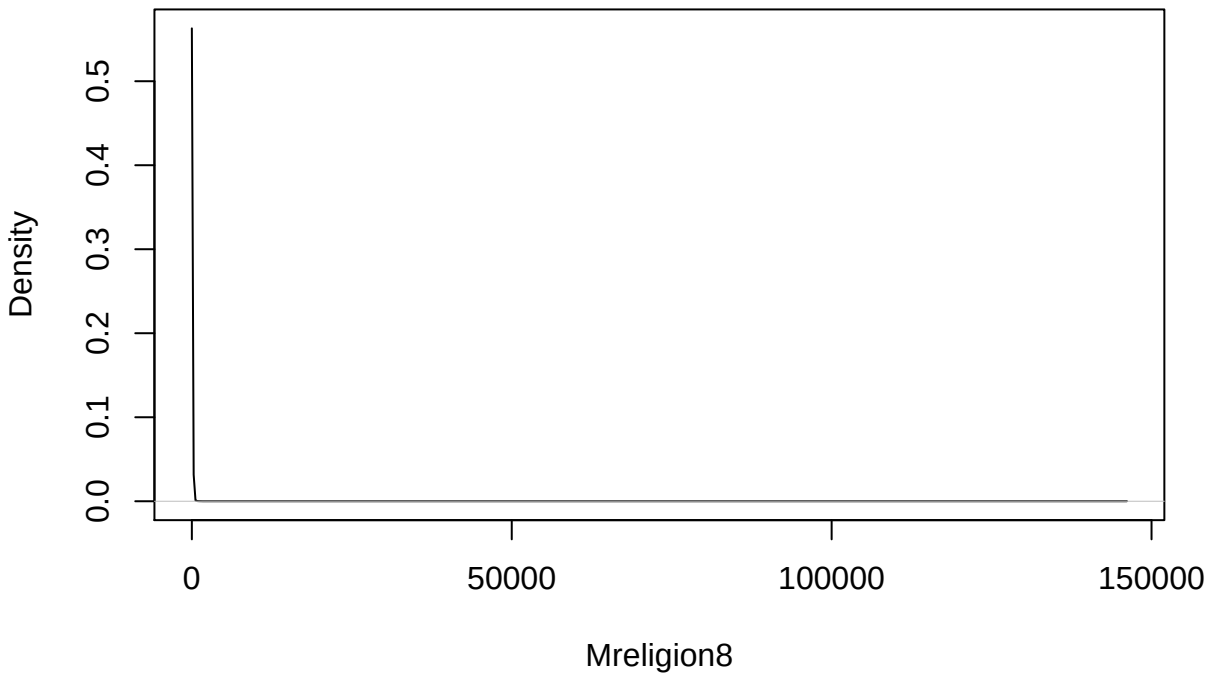

x = Mfiction1
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
200
400
600
800
1000
Mfiction1

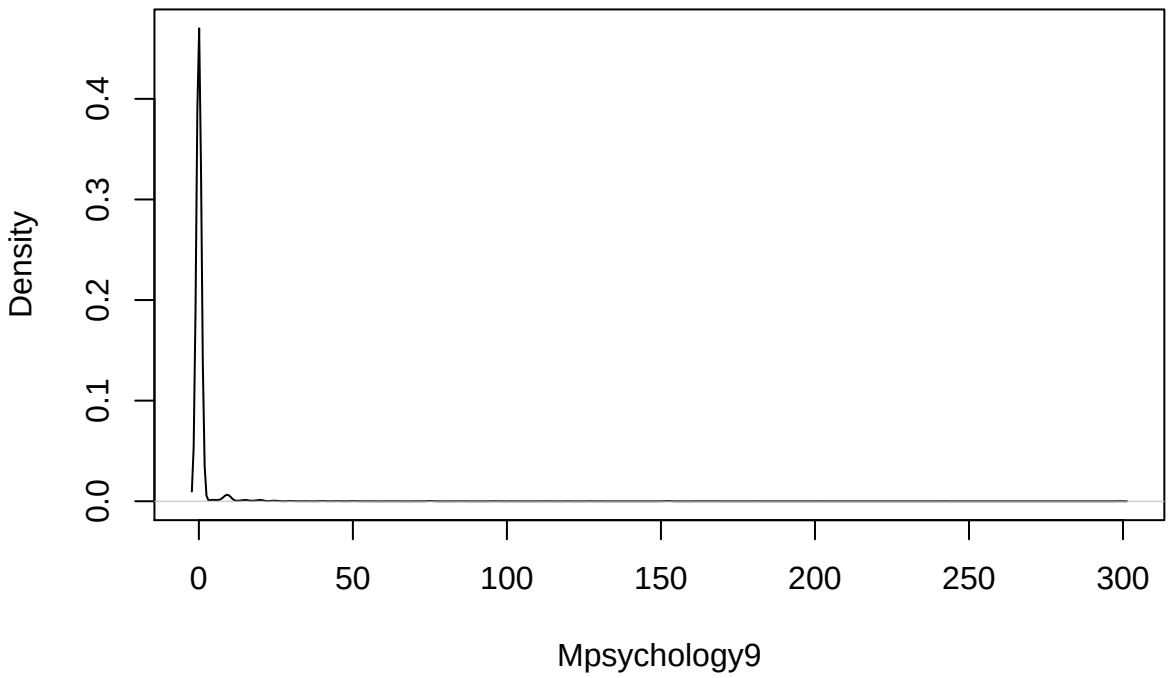

x = Mclassics3
Density
0.00
0.02
0.04
0.06
0.08
0
500
1000
1500
Mclassics3

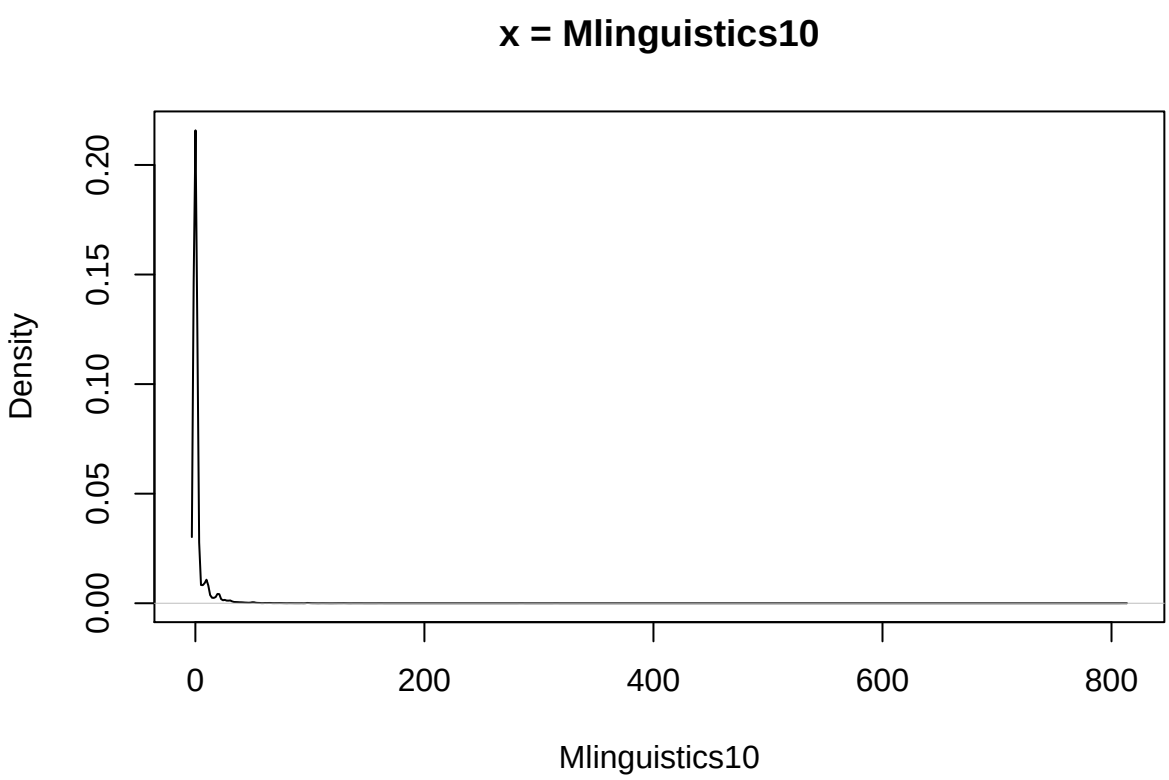

x = Mcartoons5
Density
0.00
0.10
0.20
0.30
0
50
100
150
200
250
300
350
Mcartoons5

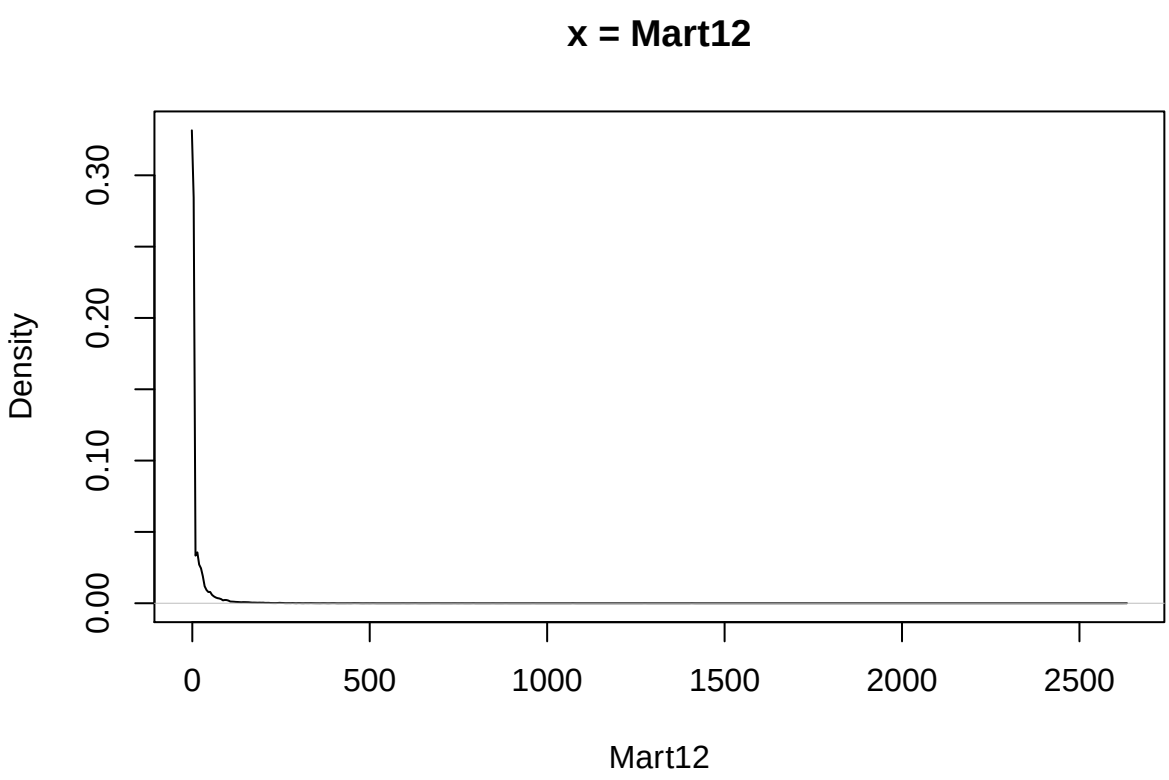

x = Mlegends6
Density
0.0
0.1
0.2
0.3
0.4
0
100
200
300
400
Mlegends6

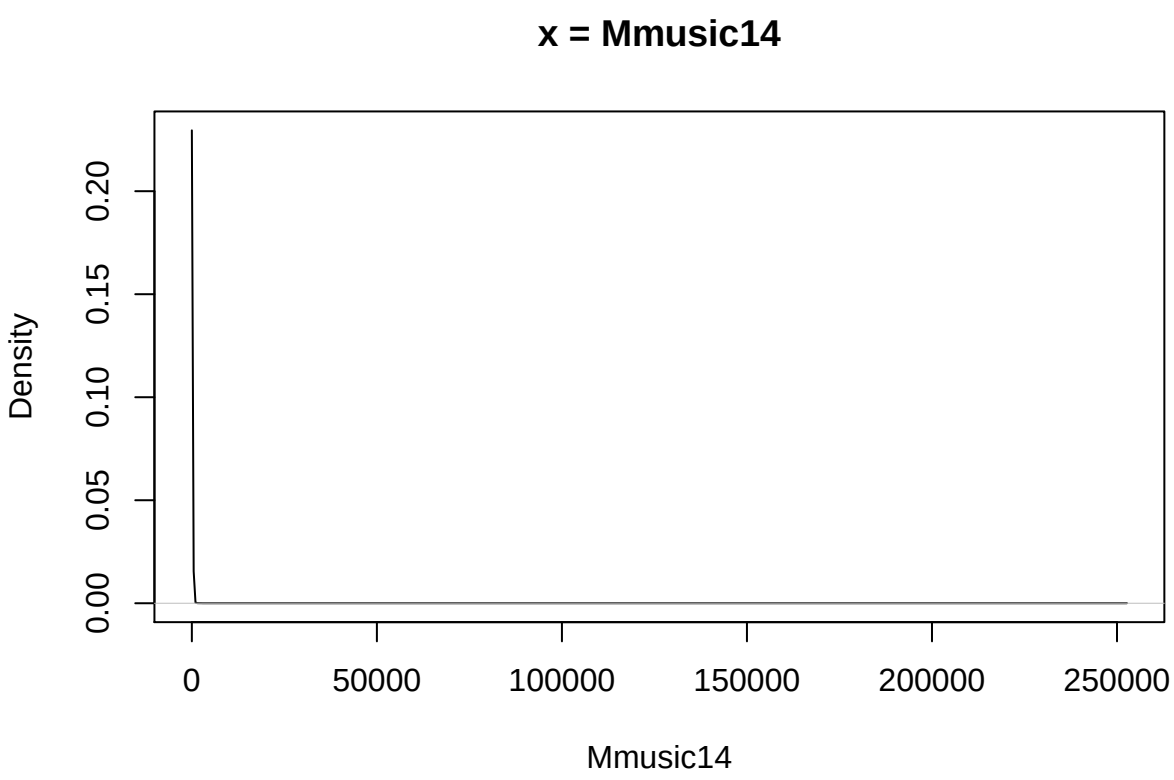

x = Mphilosophy7
Density
0.00
0.05
0.10
0.15
0
100
200
300
400
500
600
Mphilosophy7

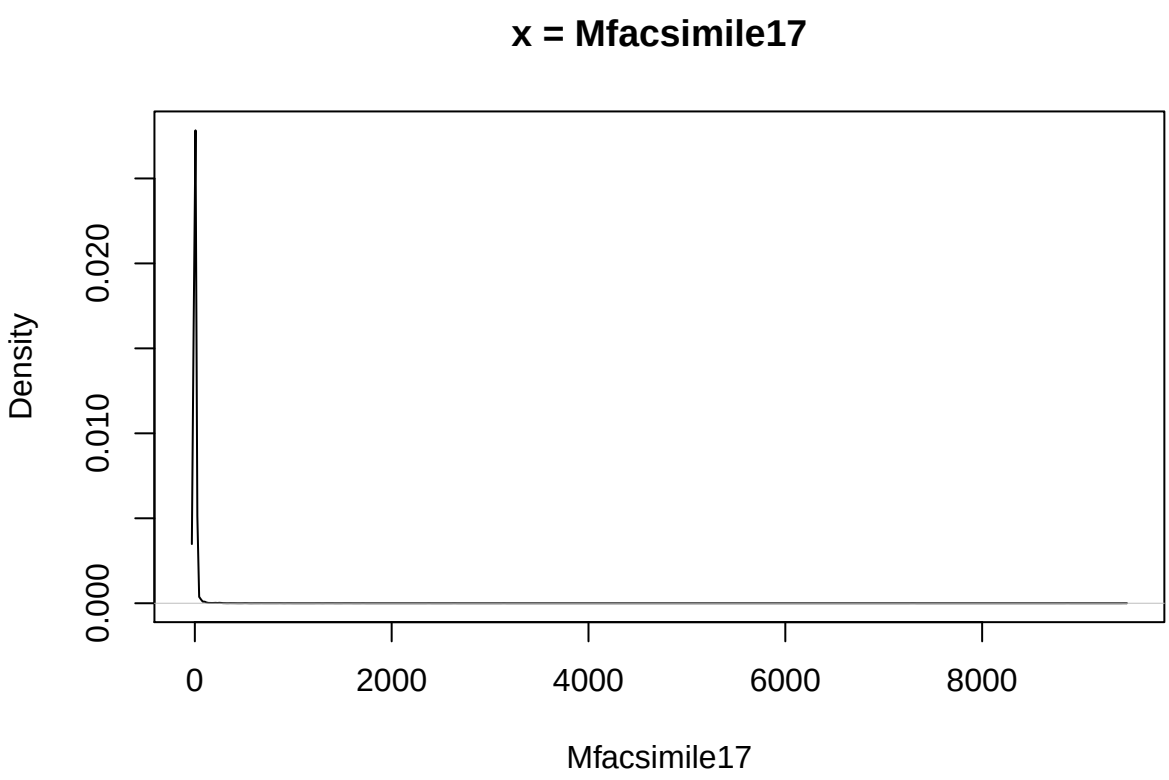

x = Mreligion8
Density
0.0
0.1
0.2
0.3
0.4
0.5
0
50000
100000
150000
Mreligion8

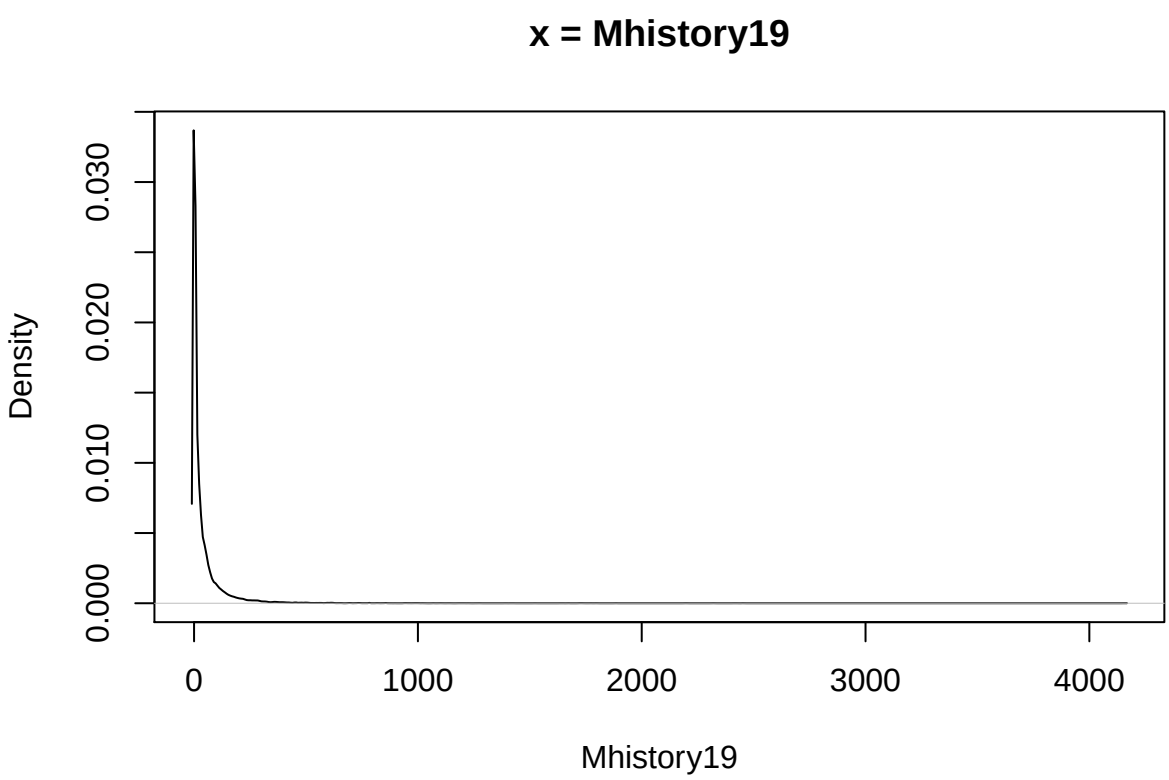

x = Mpsychology9
Density
0.0
0.1
0.2
0.3
0.4
0
50
100
150
200
250
300
Mpsychology9

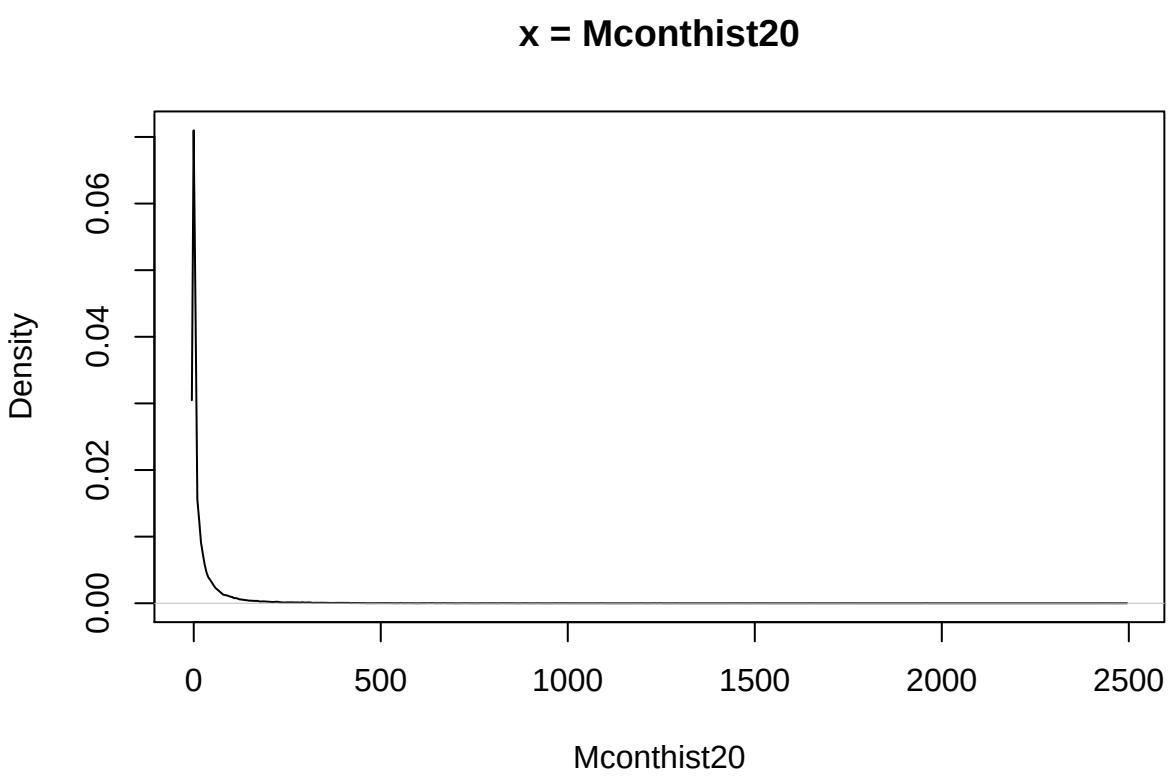

x = Mlinguistics10
Density
0.00
0.05
0.10
0.15
0.20
0
200
400
600
800
Mlinguistics10

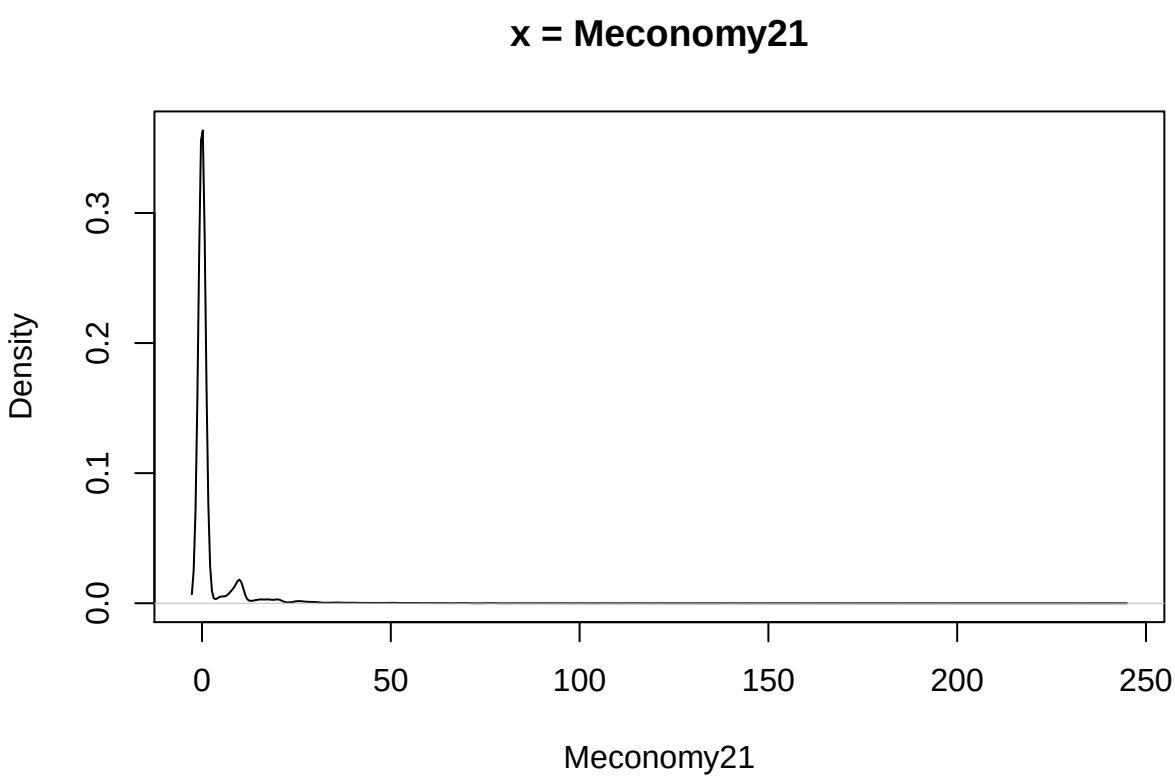

x = Mart12
Density
0.00
0.10
0.20
0.30
0
500
1000
1500
2000
2500
Mart12

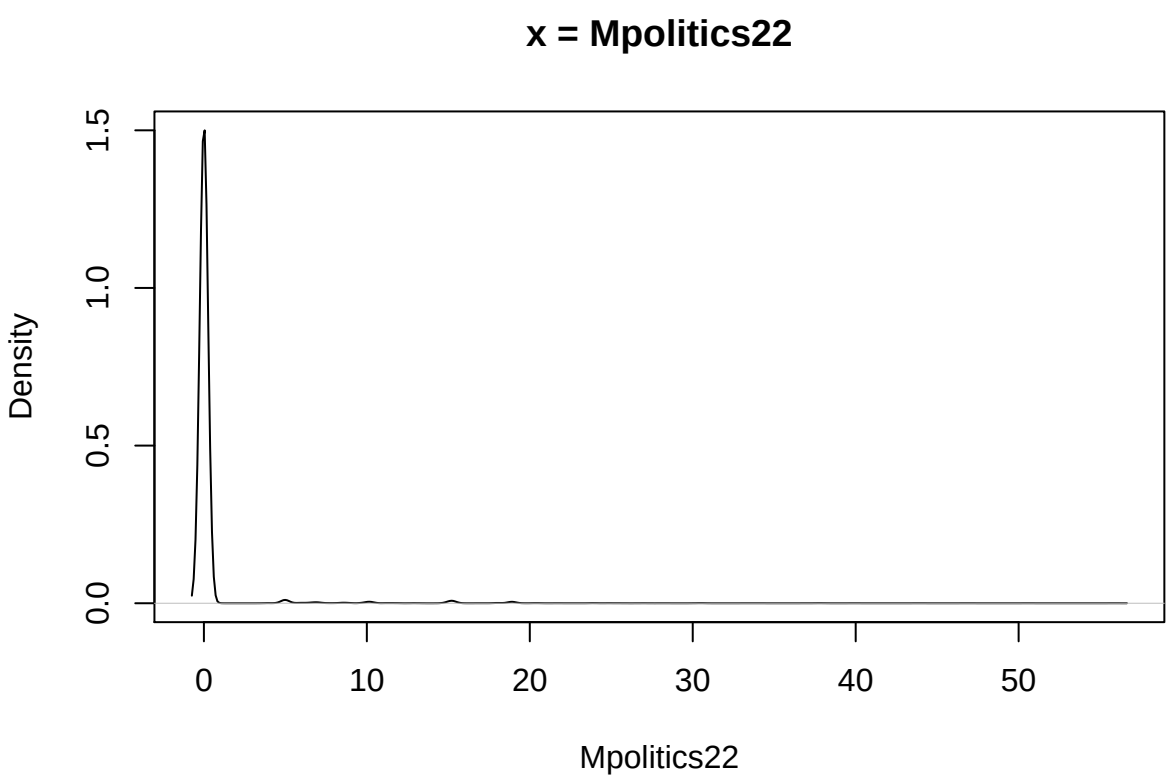

x = Mmusic14
Density
0.00
0.05
0.10
0.15
0.20
0
50000
100000
150000
200000
250000
Mmusic14

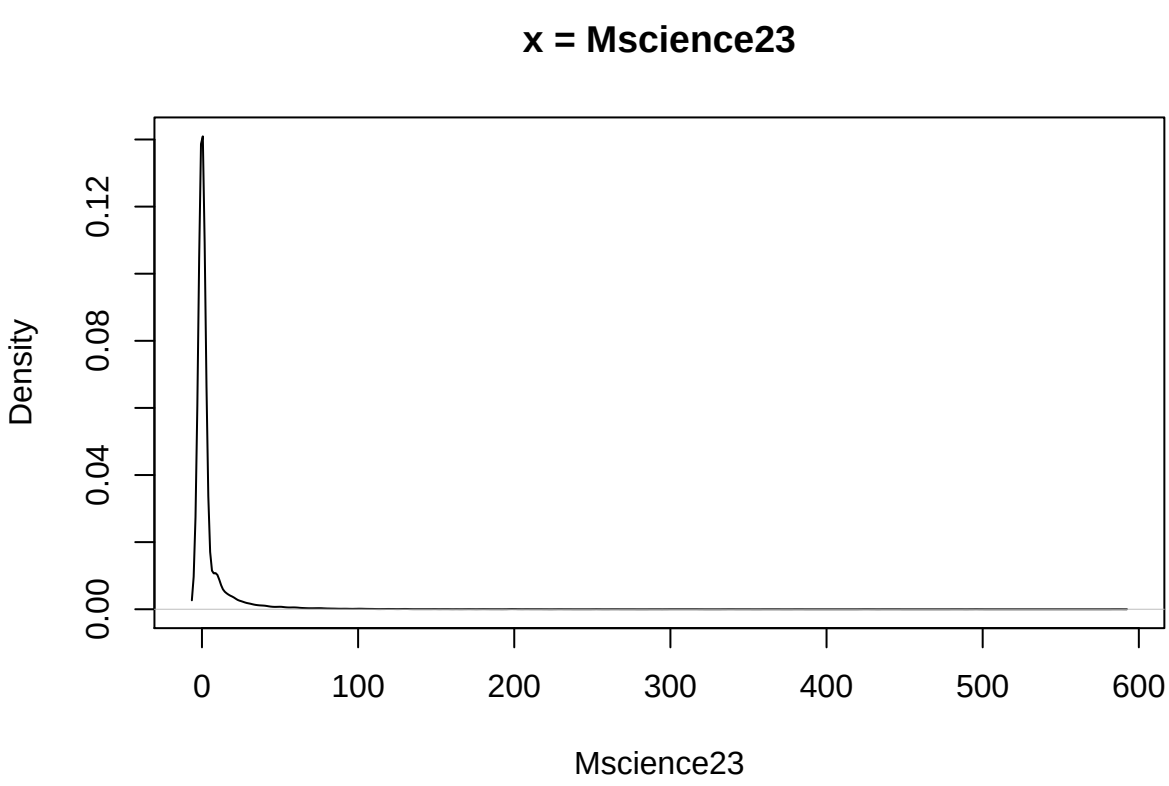

x = Mfacsimile17
Density
0.000
0.010
0.020
0
2000
4000
6000
8000
Mfacsimile17

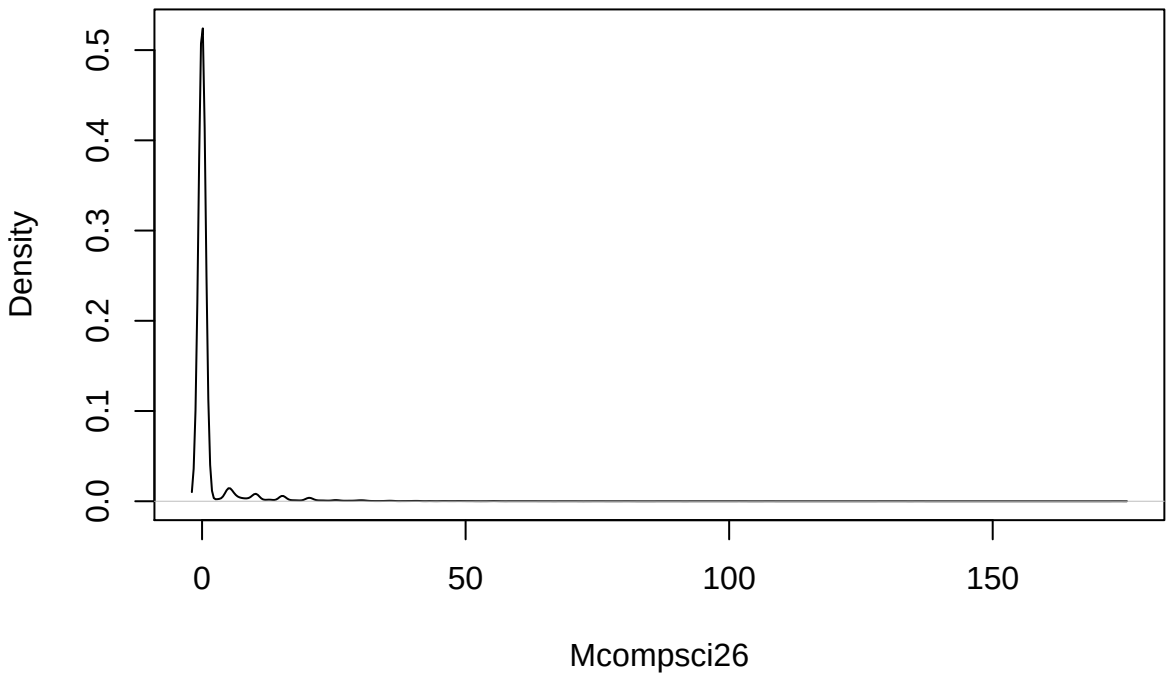

x = Mhistory19
Density
0.000
0.010
0.020
0.030
0
1000
2000
3000
4000
Mhistory19

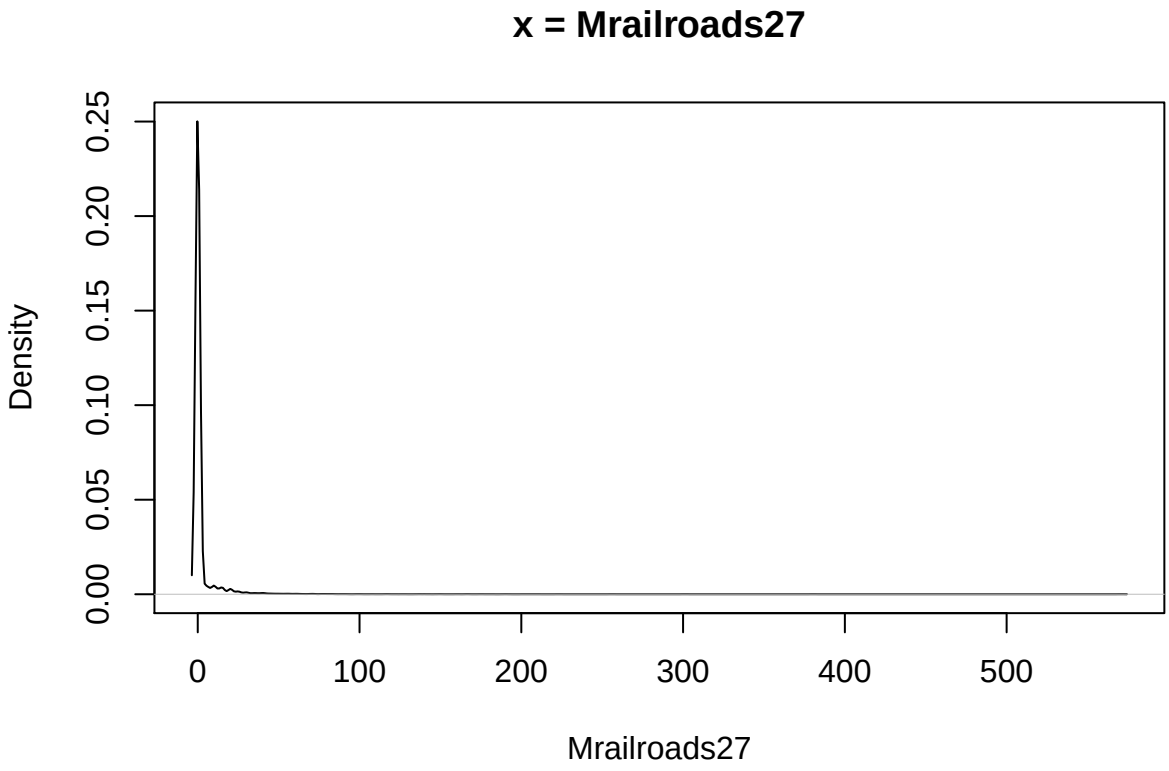

x = Mconthist20
Density
0.00
0.02
0.04
0.06
0
500
1000
1500
2000
2500
Mconthist20

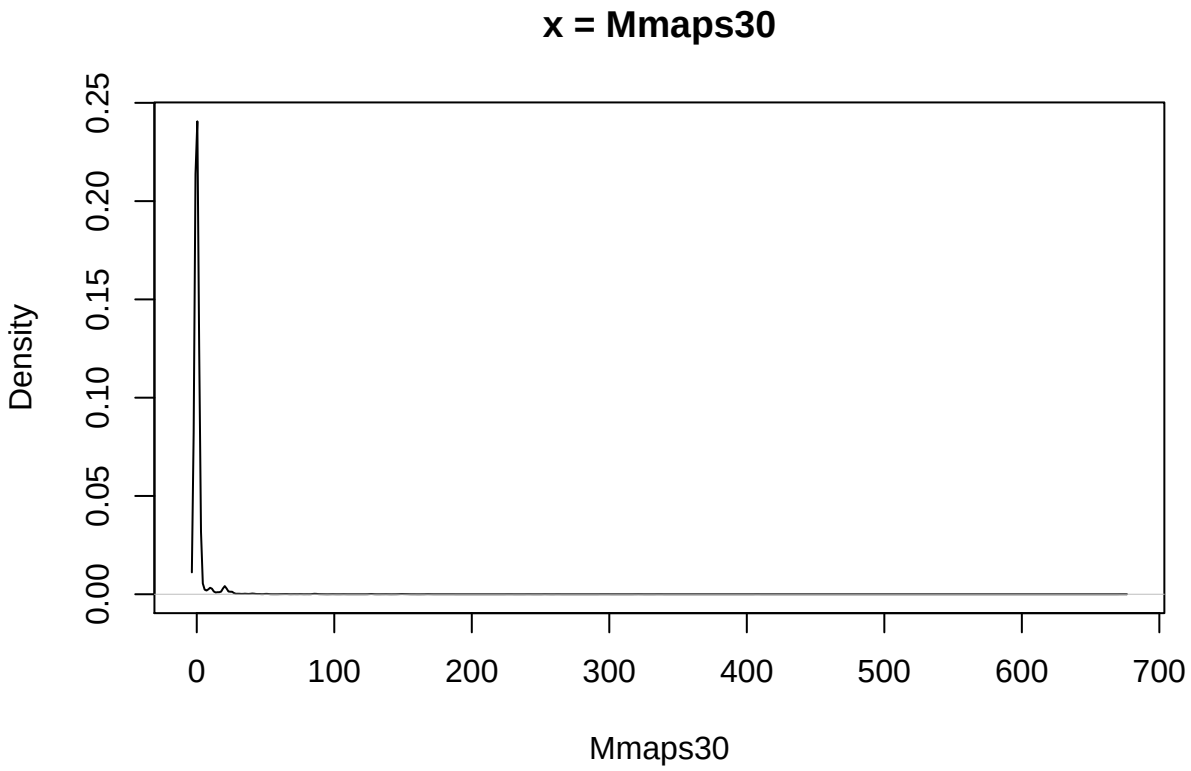

x = Meconomy21
Density
0.0
0.1
0.2
0.3
0
50
100
150
200
250
Meconomy21

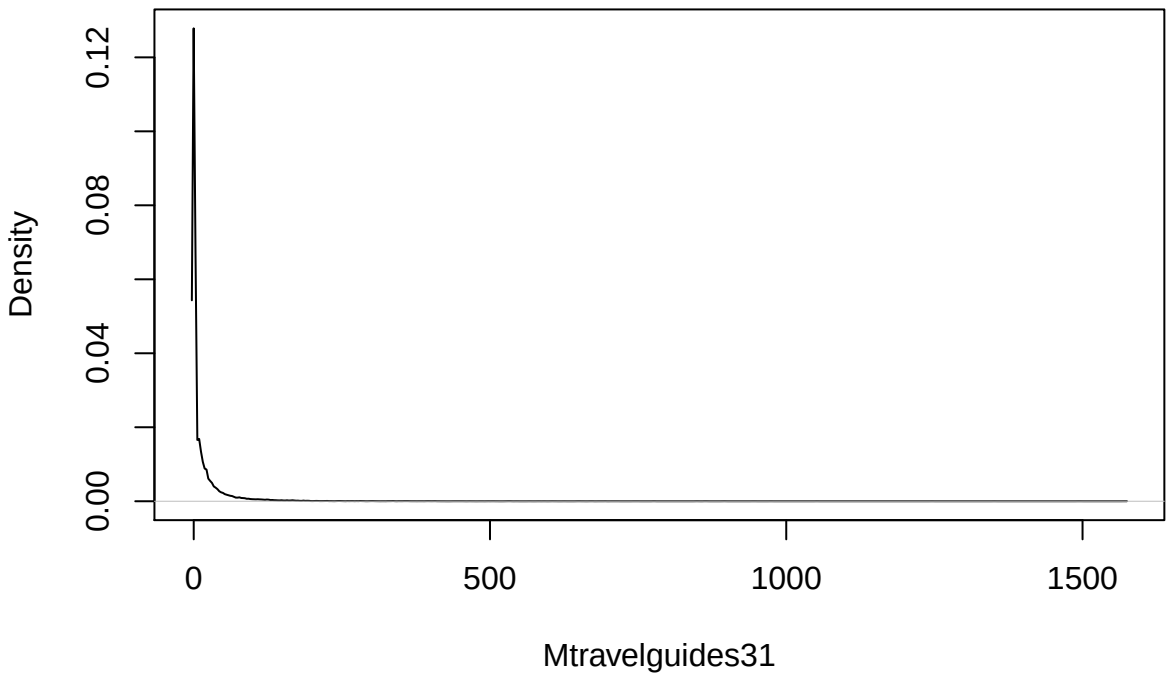

x = Mpolitics22
Density
0.0
0.5
1.0
1.5
0
10
20
30
40
50
Mpolitics22

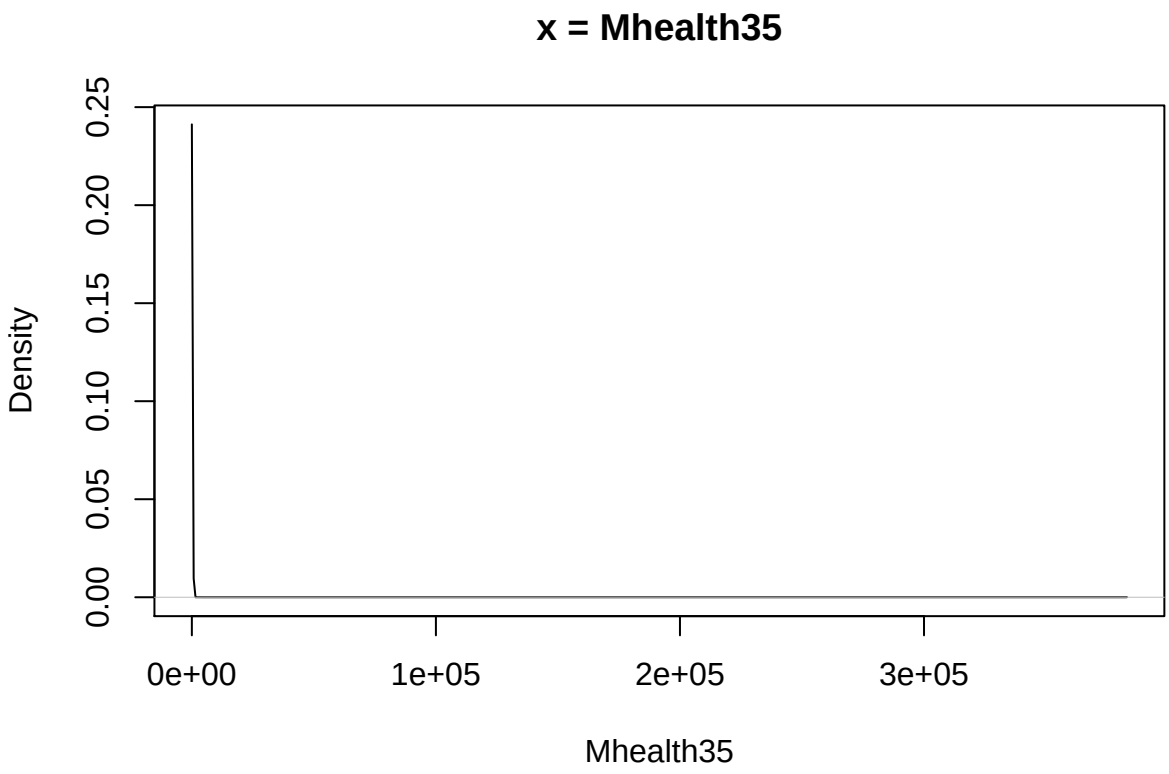

x = Mscience23
Density
0.00
0.04
0.08
0.12
0
100
200
300
400
500
600
Mscience23

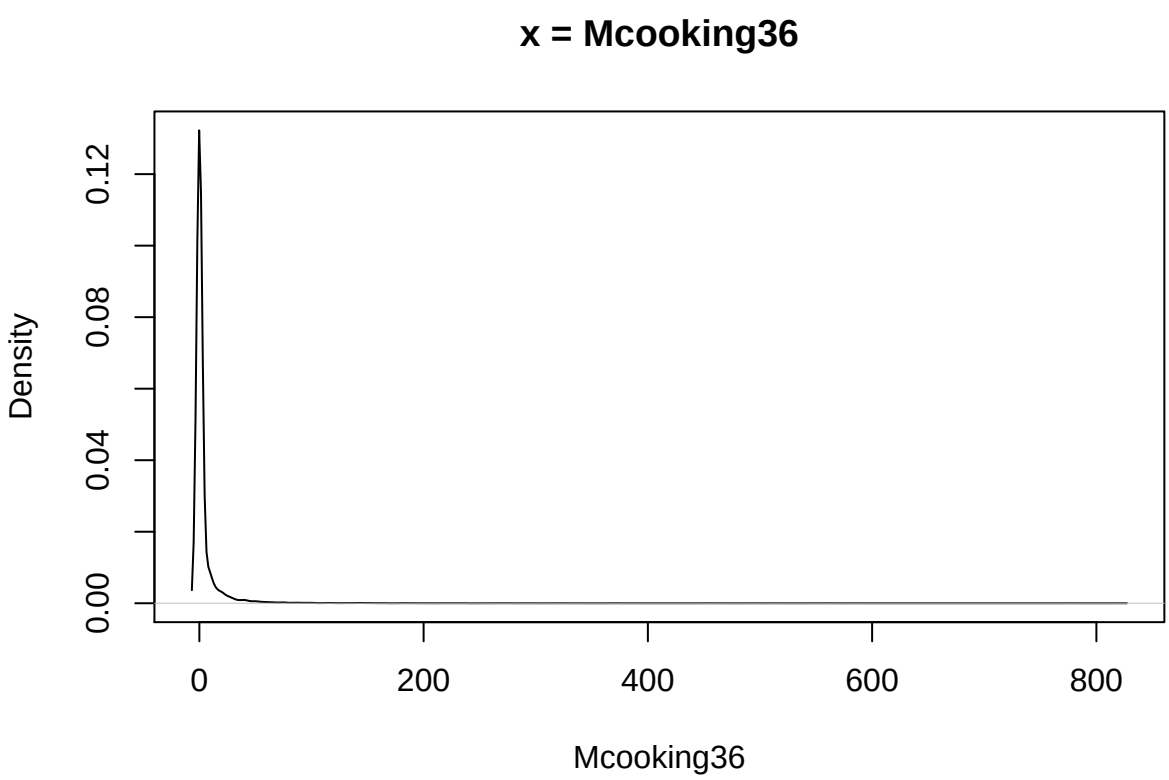

x = Mcompsci26
Density
0.0
0.1
0.2
0.3
0.4
0.5
0
50
100
150
Mcompsci26

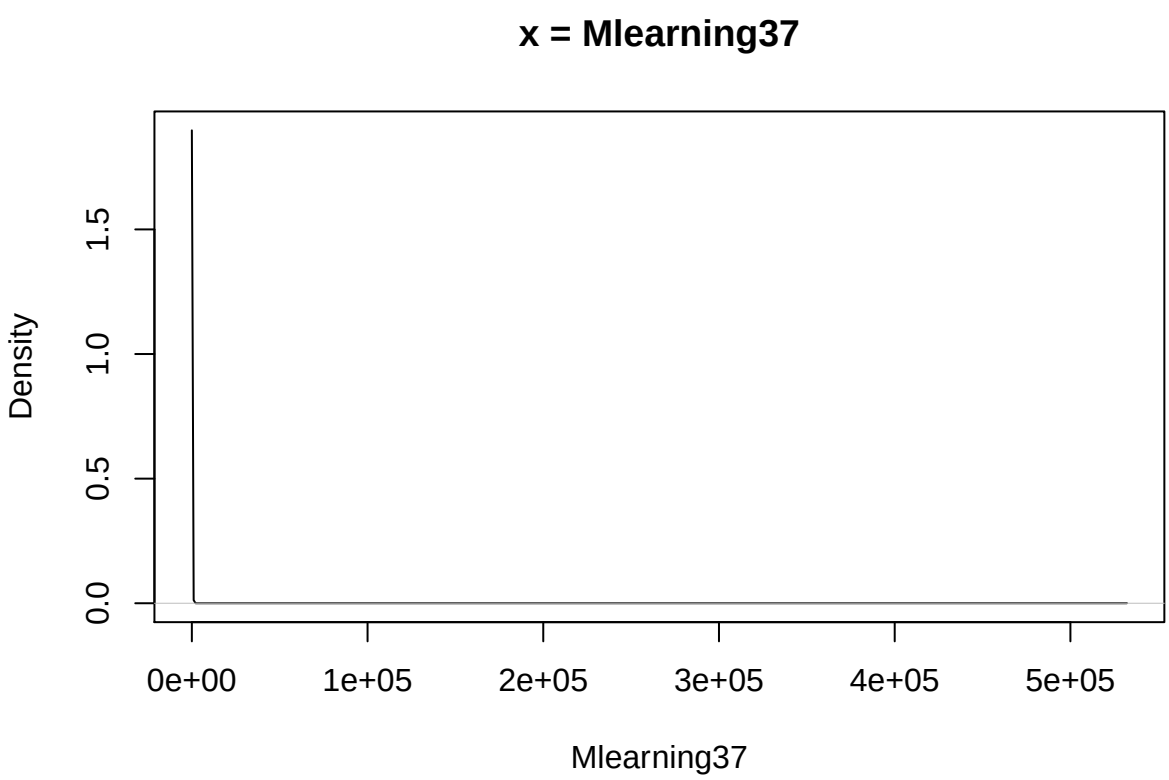

x = Mrailroads27
Density
0.00
0.05
0.10
0.15
0.20
0.25
0
100
200
300
400
500
Mrailroads27

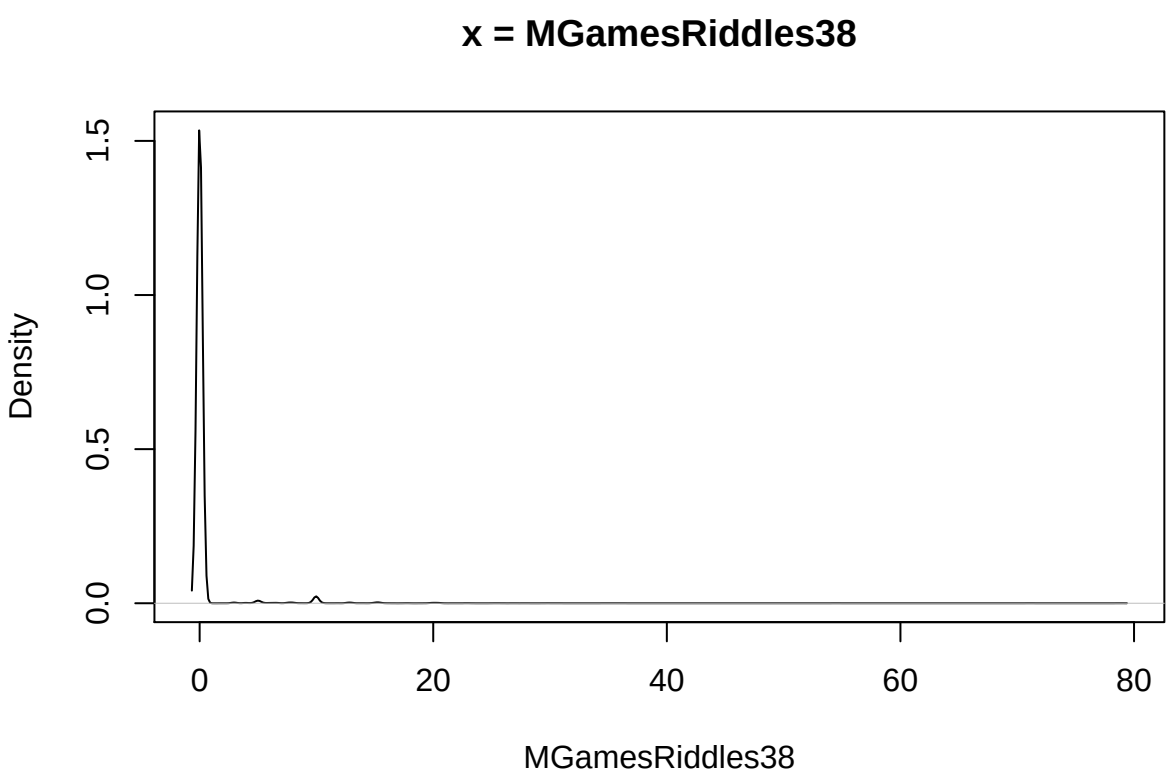

x = Mmaps30
Density
0.00
0.05
0.10
0.15
0.20
0.25
0
100
200
300
400
500
600
700
Mmaps30

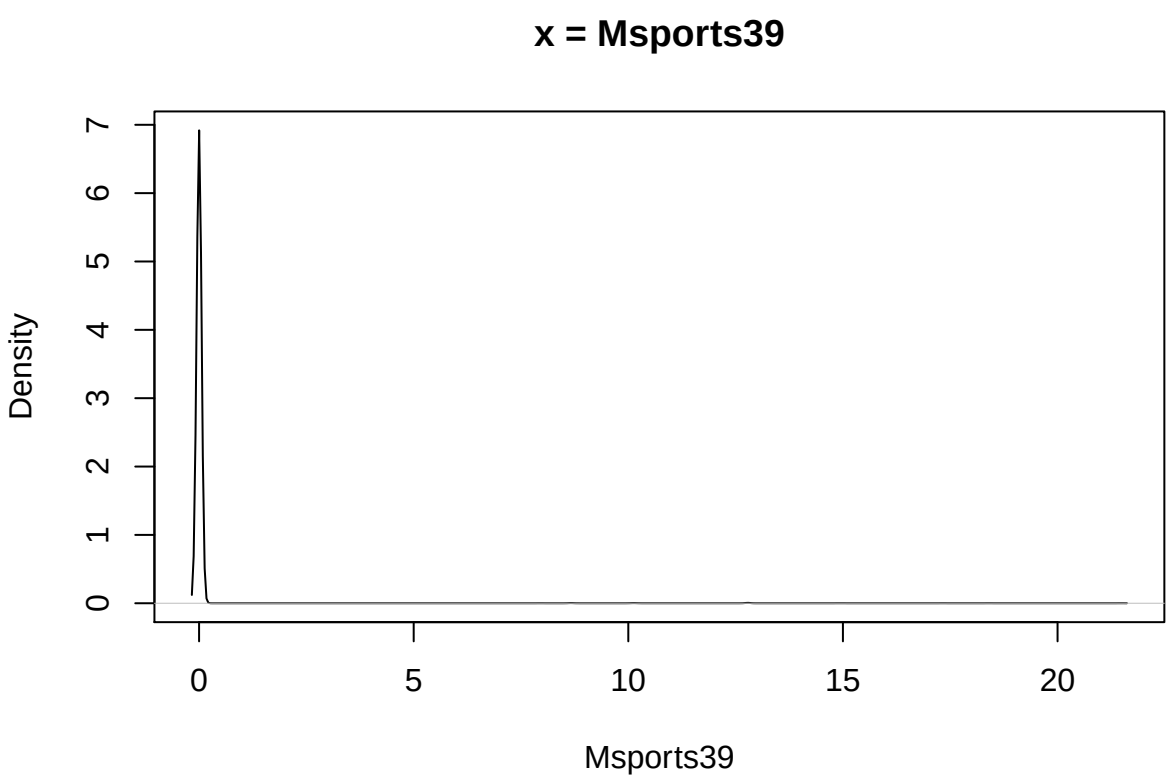

x = Mtravelguides31
Density
0.00
0.04
0.08
0.12
0
500
1000
1500
Mtravelguides31

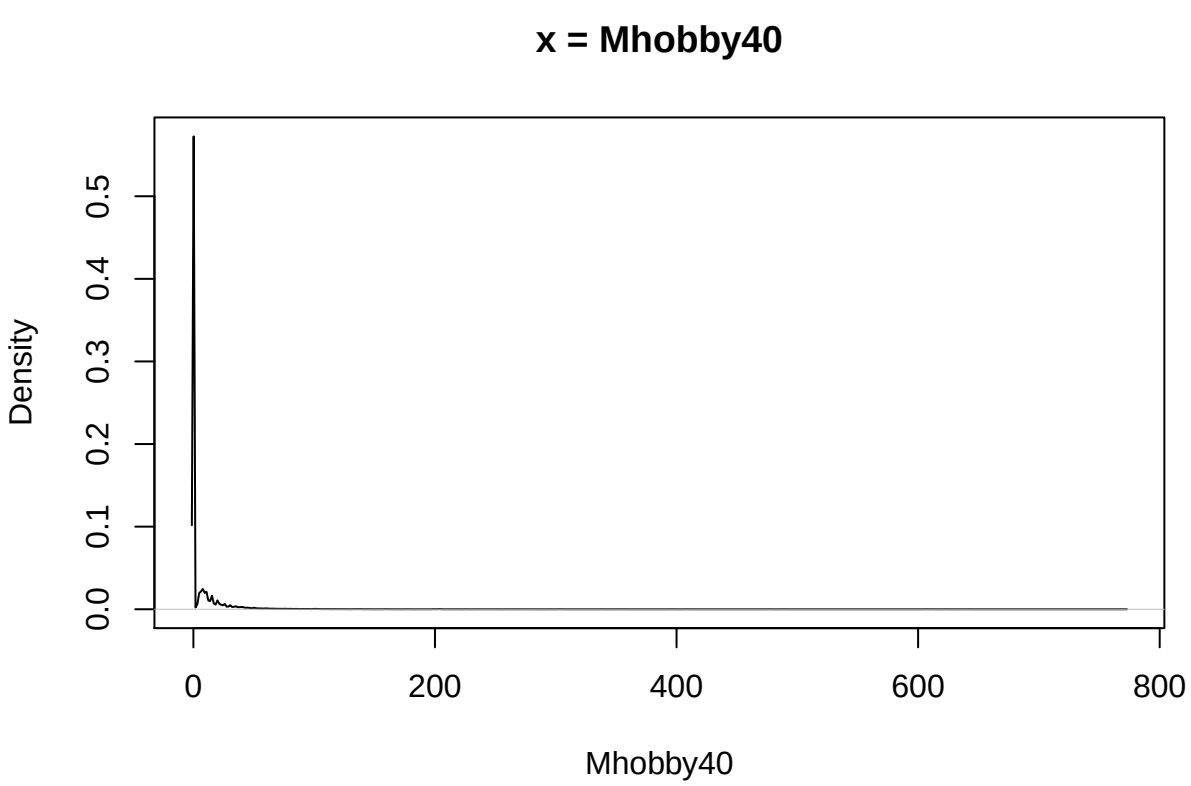

x = Mhealth35
Density
0.00
0.05
0.10
0.15
0.20
0.25
0e+00
1e+05
2e+05
3e+05
Mhealth35

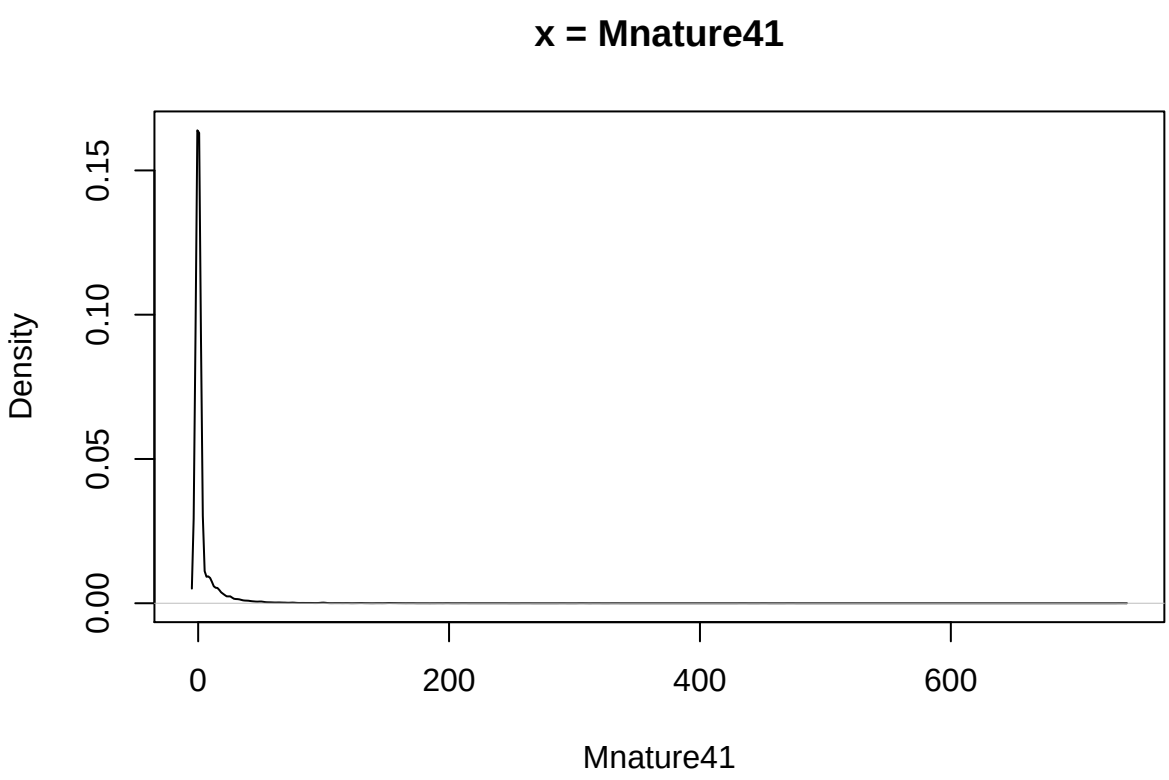

x = Mcooking36
Density
0.00
0.04
0.08
0.12
0
200
400
600
800
Mcooking36

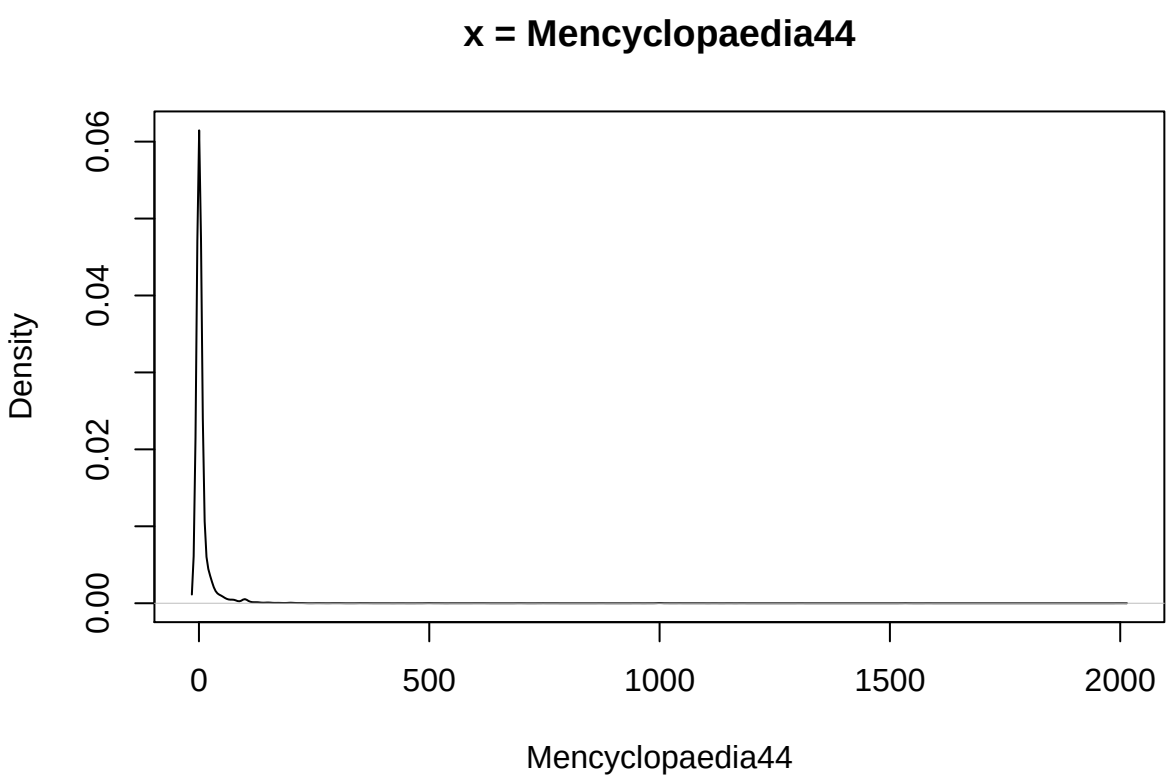

x = Mlearning37
Density
0.0
0.5
1.0
1.5
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Mlearning37

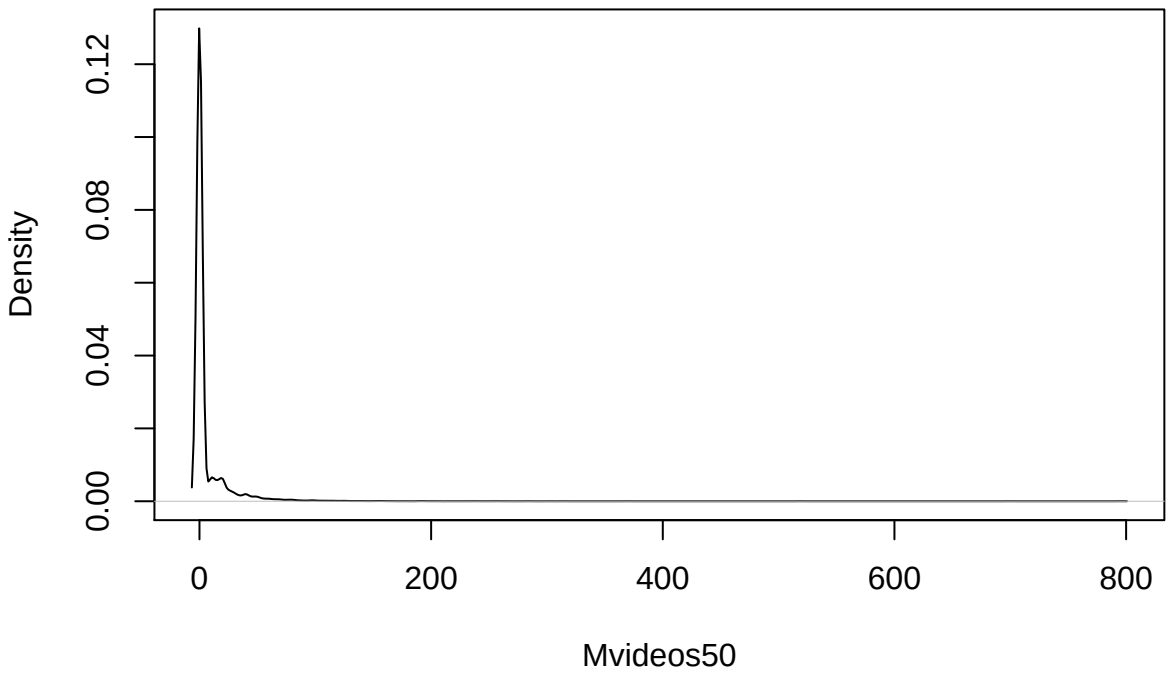

x = MGamesRiddles38
Density
0.0
0.5
1.0
1.5
0
20
40
60
80
MGamesRiddles38

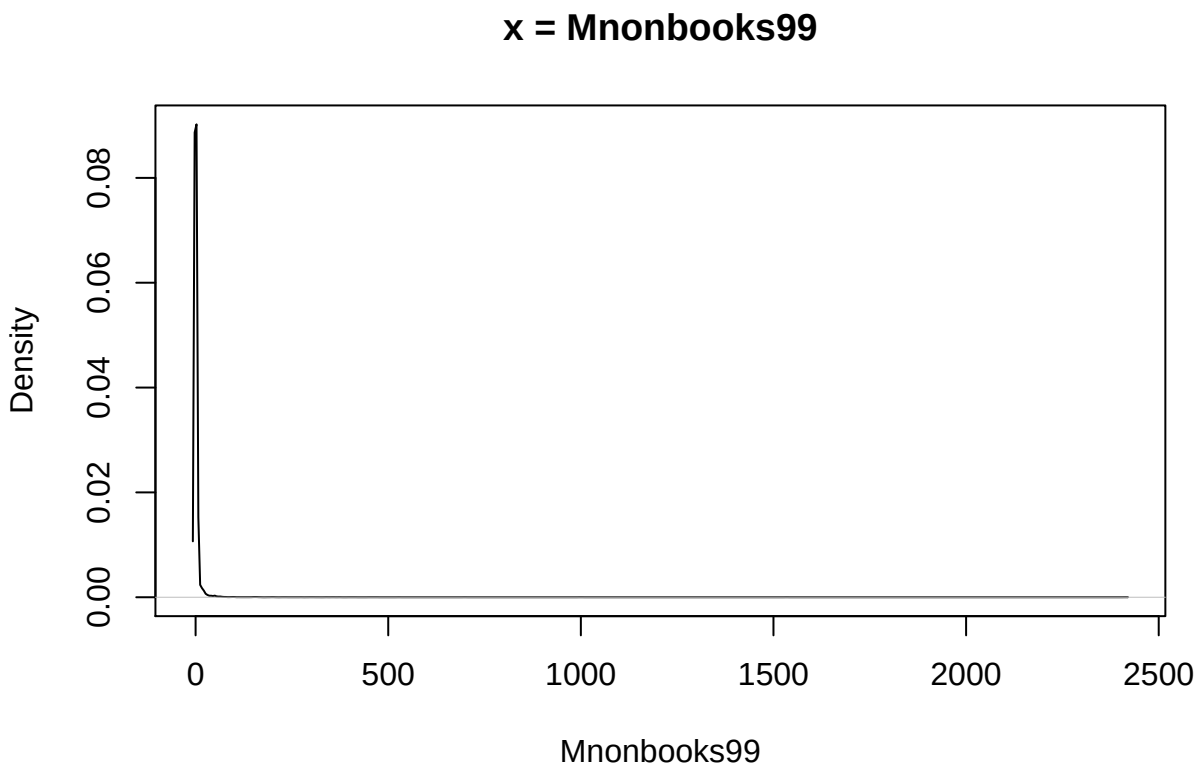

x = Msports39
Density
0
1
2
3
4
5
6
7
0
5
10
15
20
Msports39


x = Mhobby40
Density
0.0
0.1
0.2
0.3
0.4
0.5
0
200
400
600
800
Mhobby40

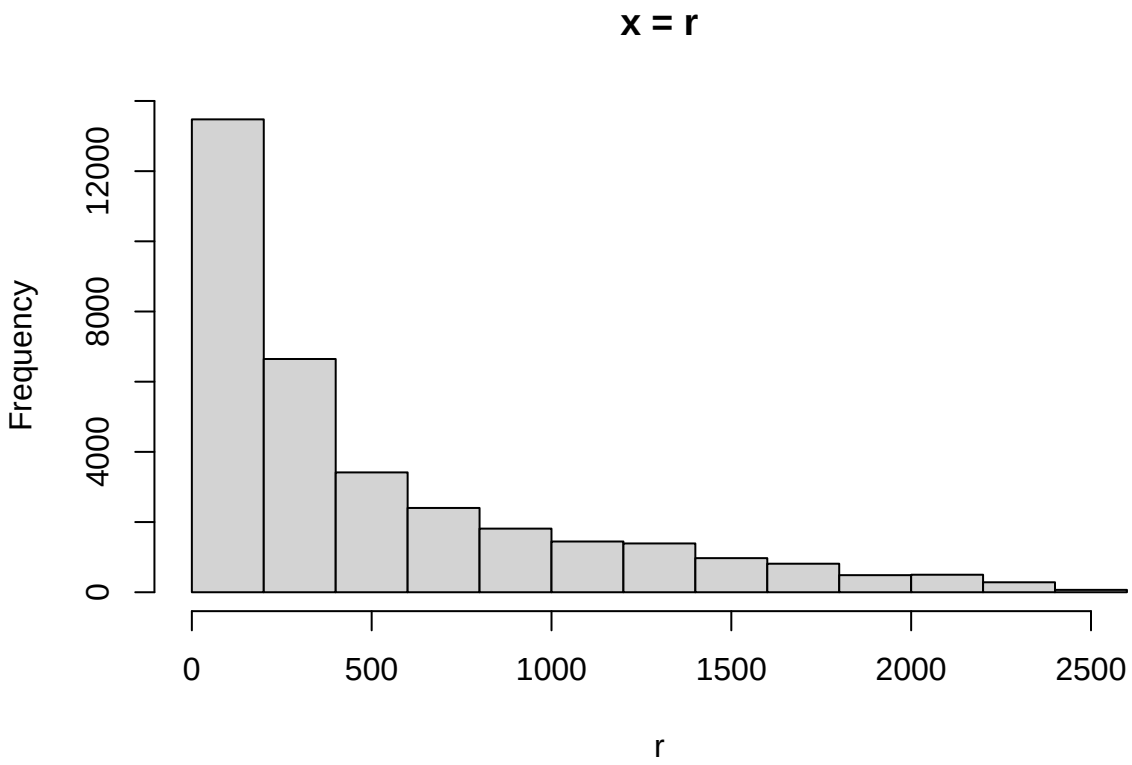

x = Mnature41
Density
0.00
0.05
0.10
0.15
0
200
400
600
Mnature41

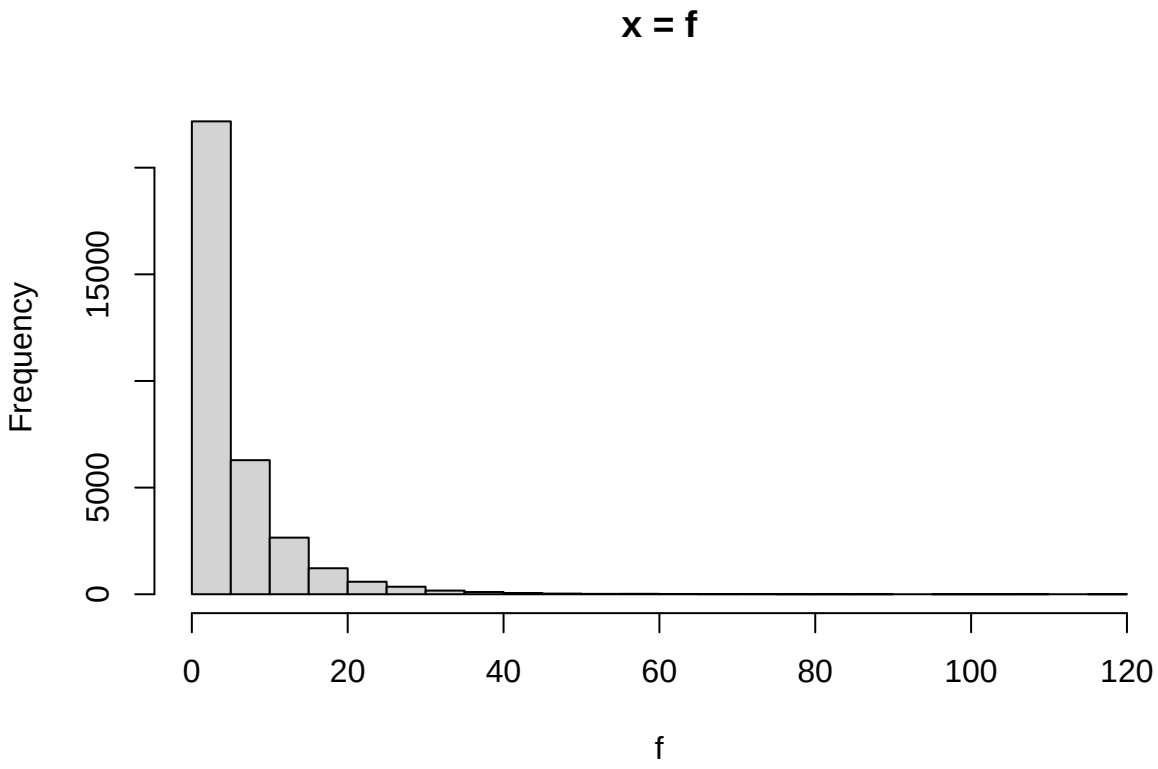

x = Mencyclopaedia44
Density
0.00
0.02
0.04
0.06
0
500
1000
1500
2000
Mencyclopaedia44

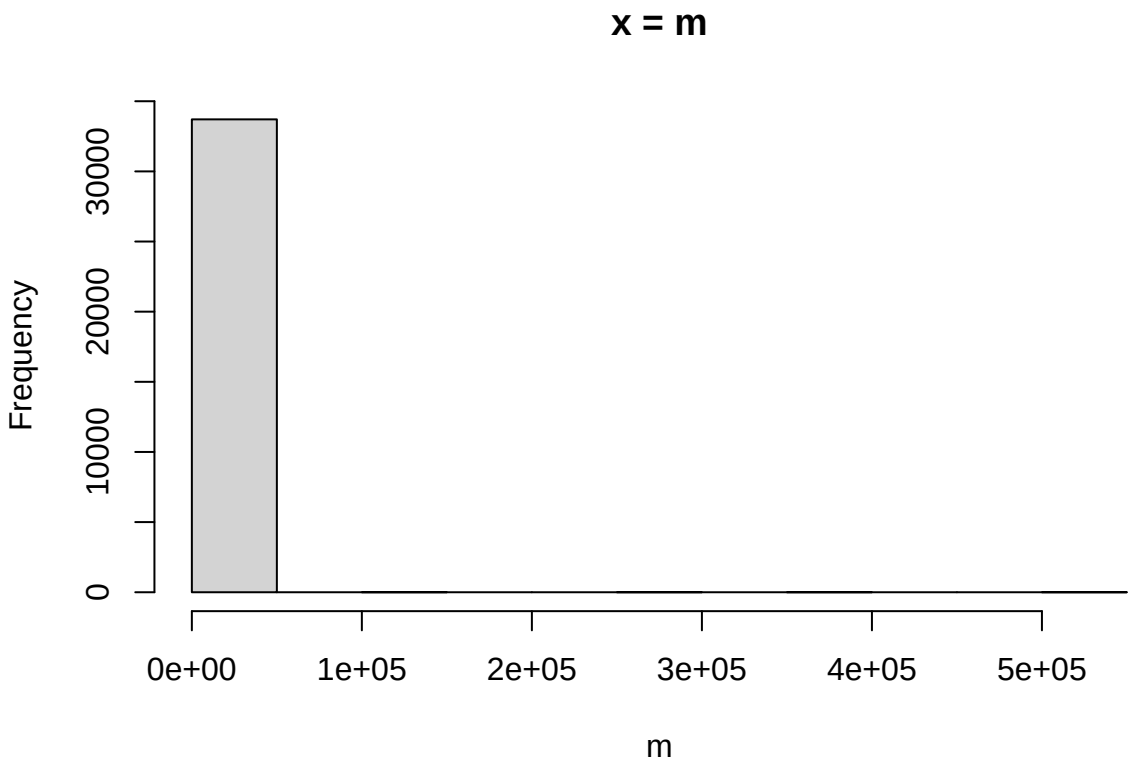

x = Mvideos50
Density
0.00
0.04
0.08
0.12
0
200
400
600
800
Mvideos50

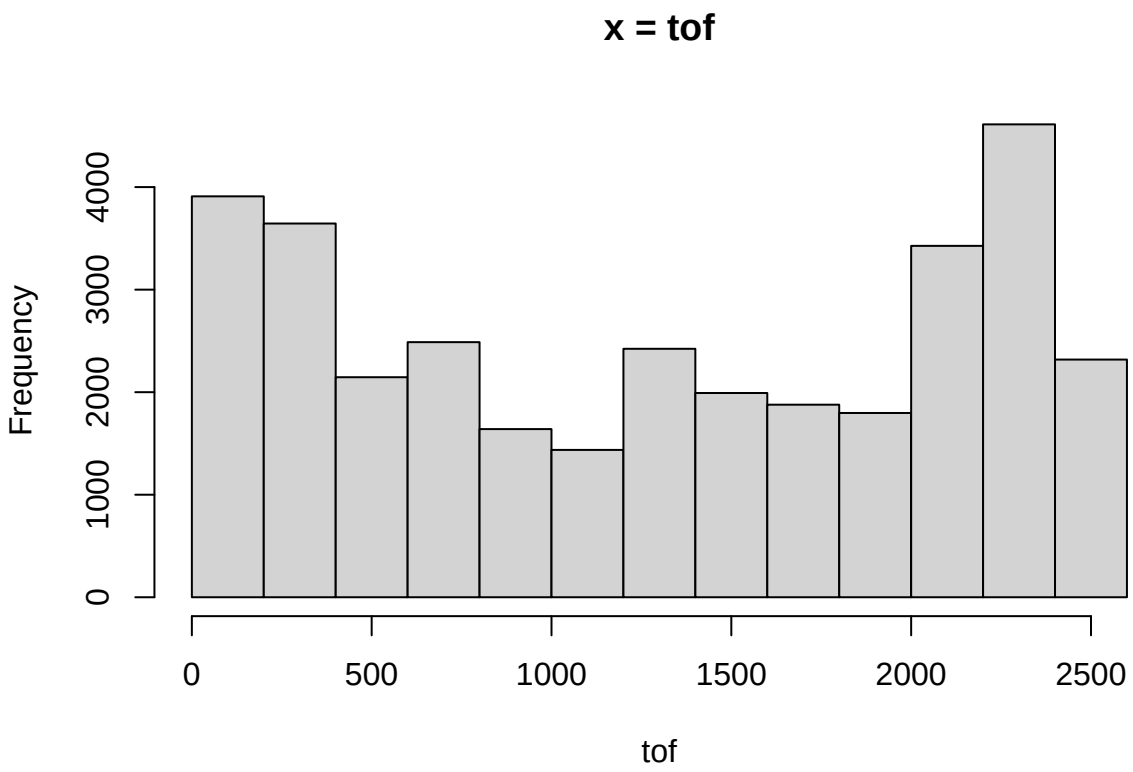


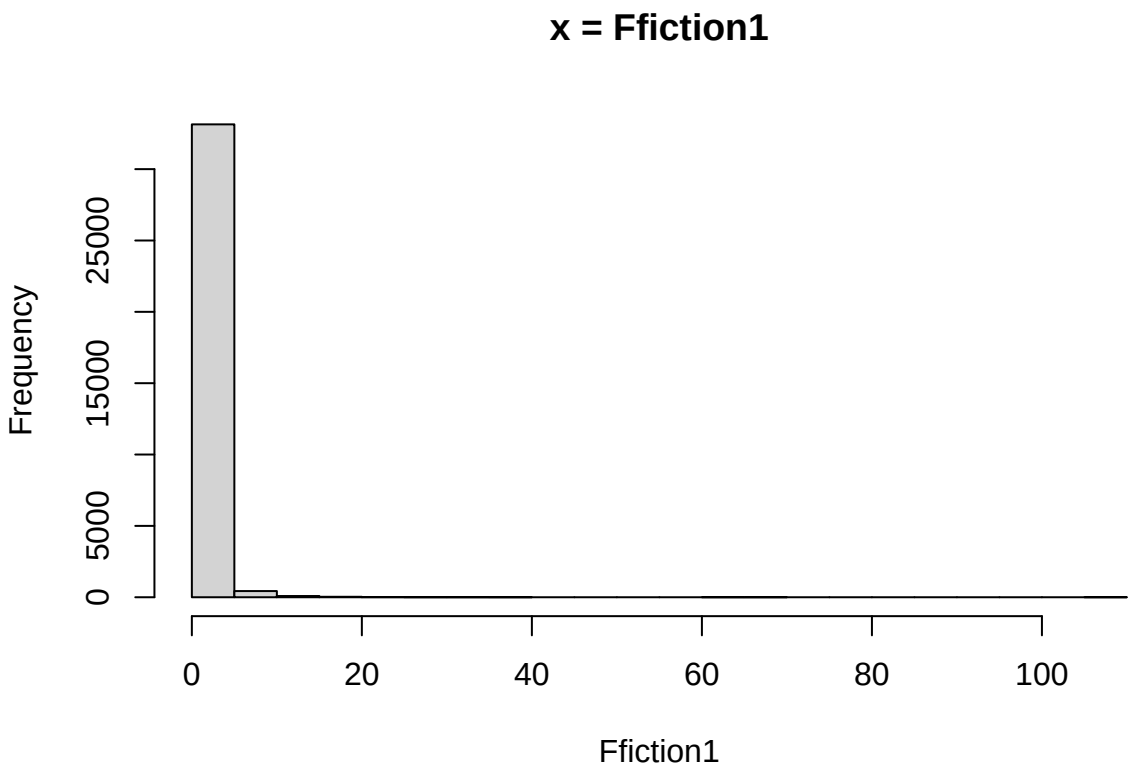

```
for (i in 2:65){
  hist(unlist(df[,i]), main = sprintf("x = %s", xlab = names(df[,i])), xlab = names(df[,i]))
}
```

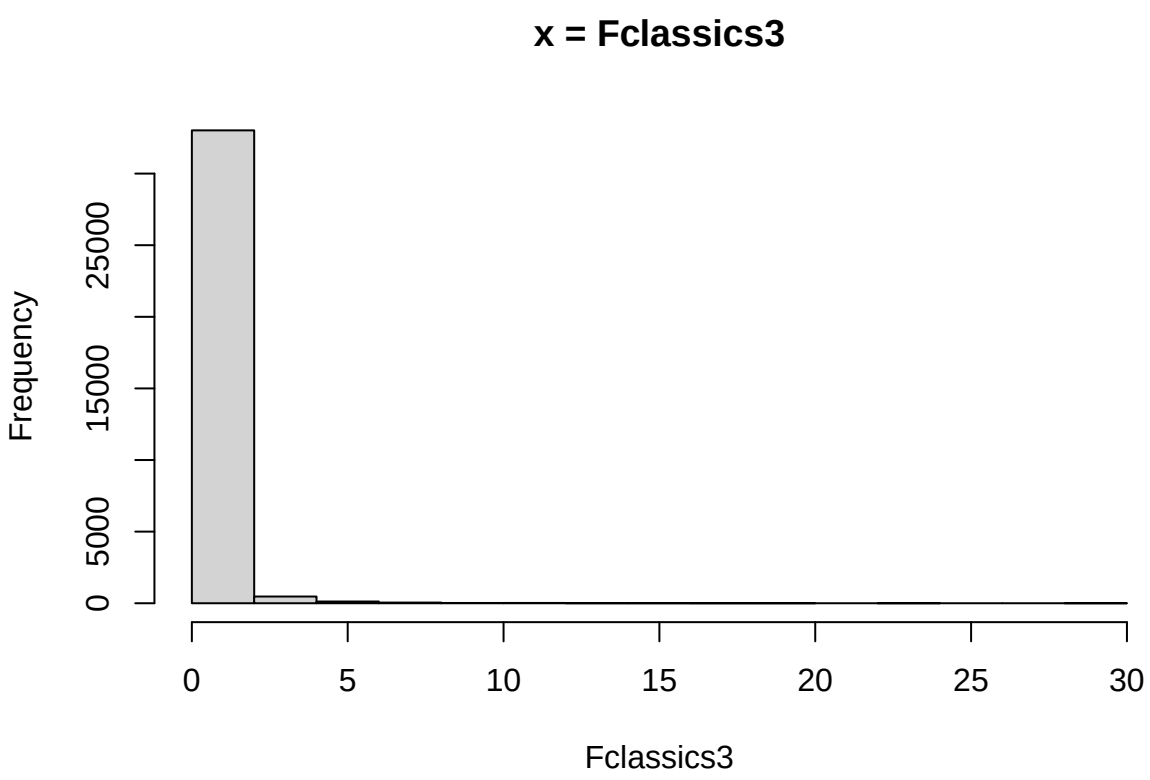

x = r
Frequency
0
4000
8000
12000
0
500
1000
1500
2000
2500
r

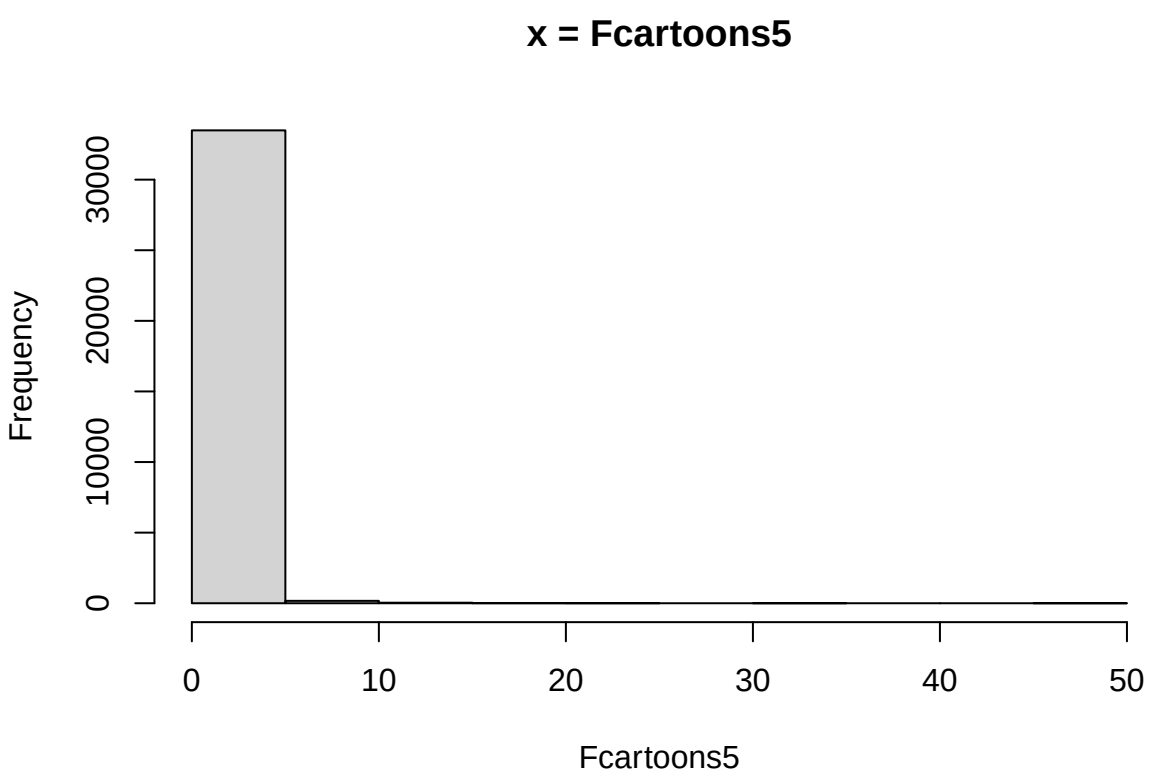

x = f
Frequency
0
5000
15000
0
20
40
60
80
100
120
f

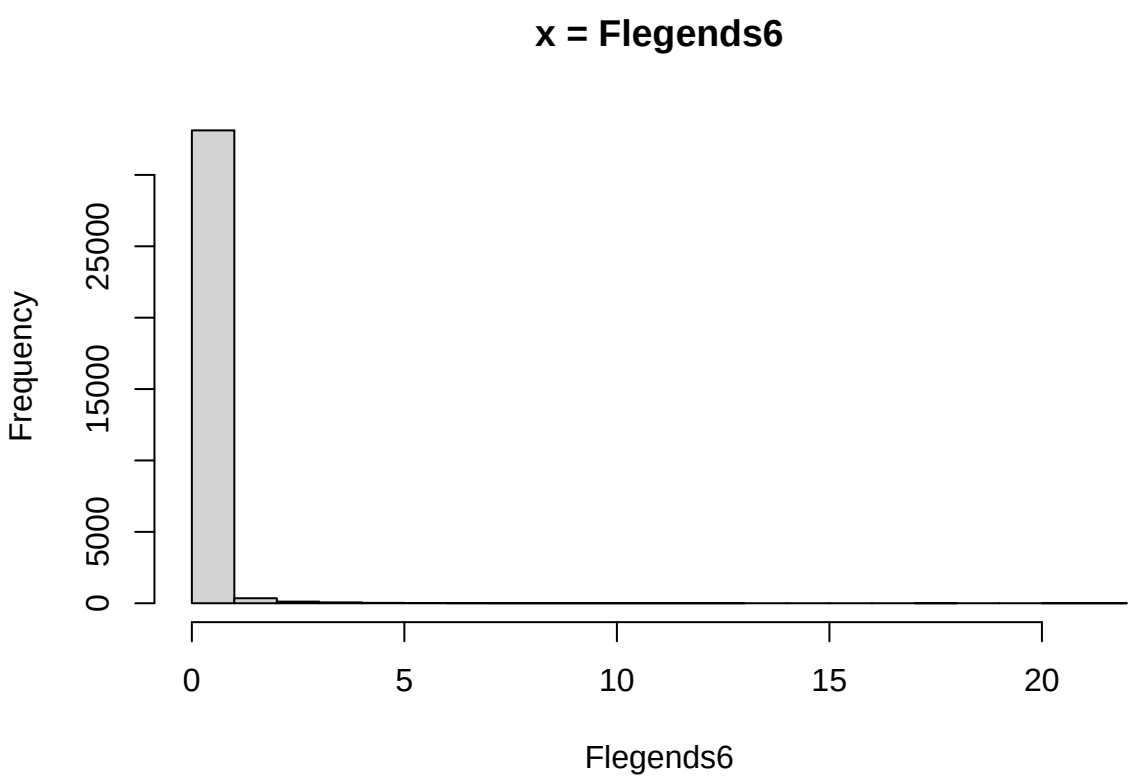

x = m
Frequency
0
10000
20000
30000
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
m

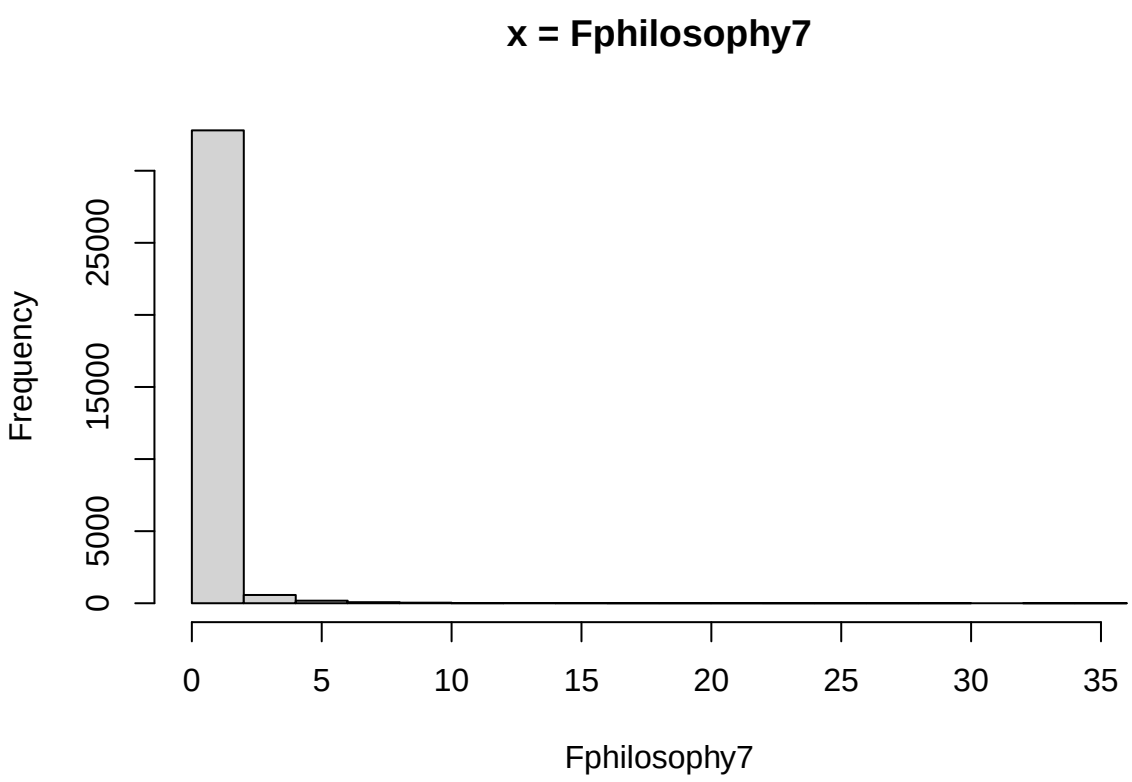

x = tof
Frequency
0
1000
2000
3000
4000
0
500
1000
1500
2000
2500
tof

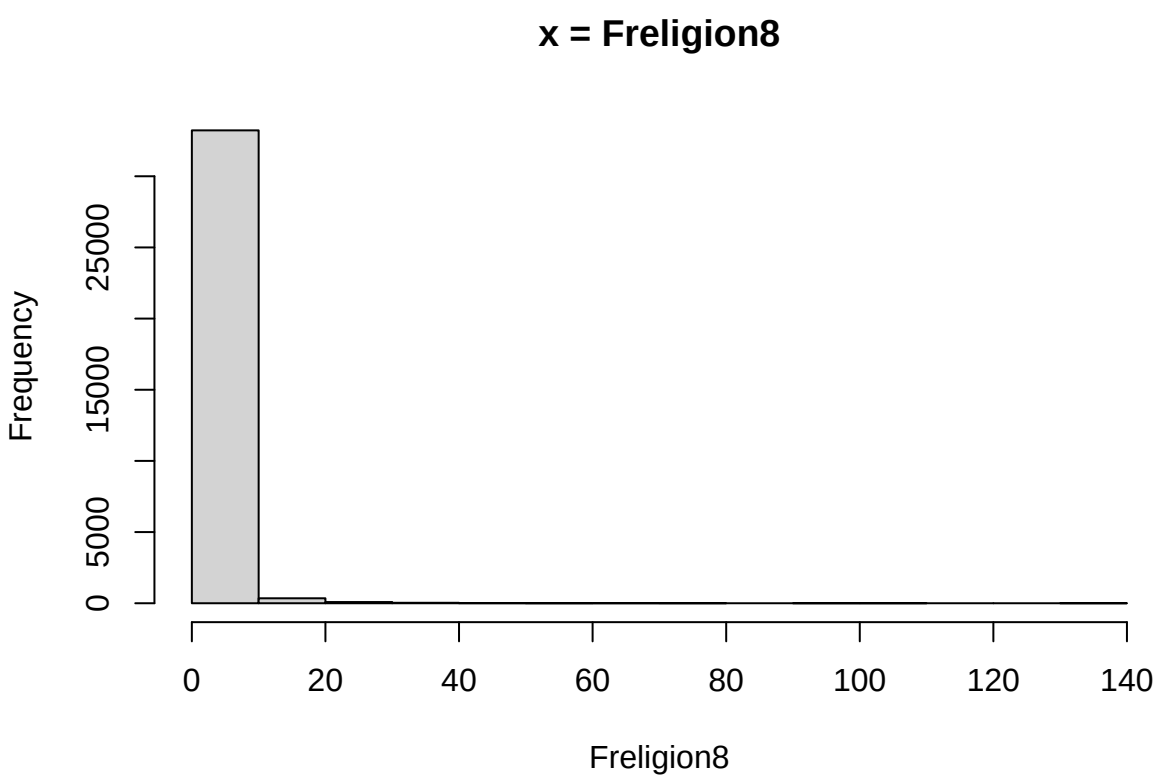

x = Ffiction1
Frequency
0
5000
15000
25000
0
20
40
60
80
100
Ffiction1

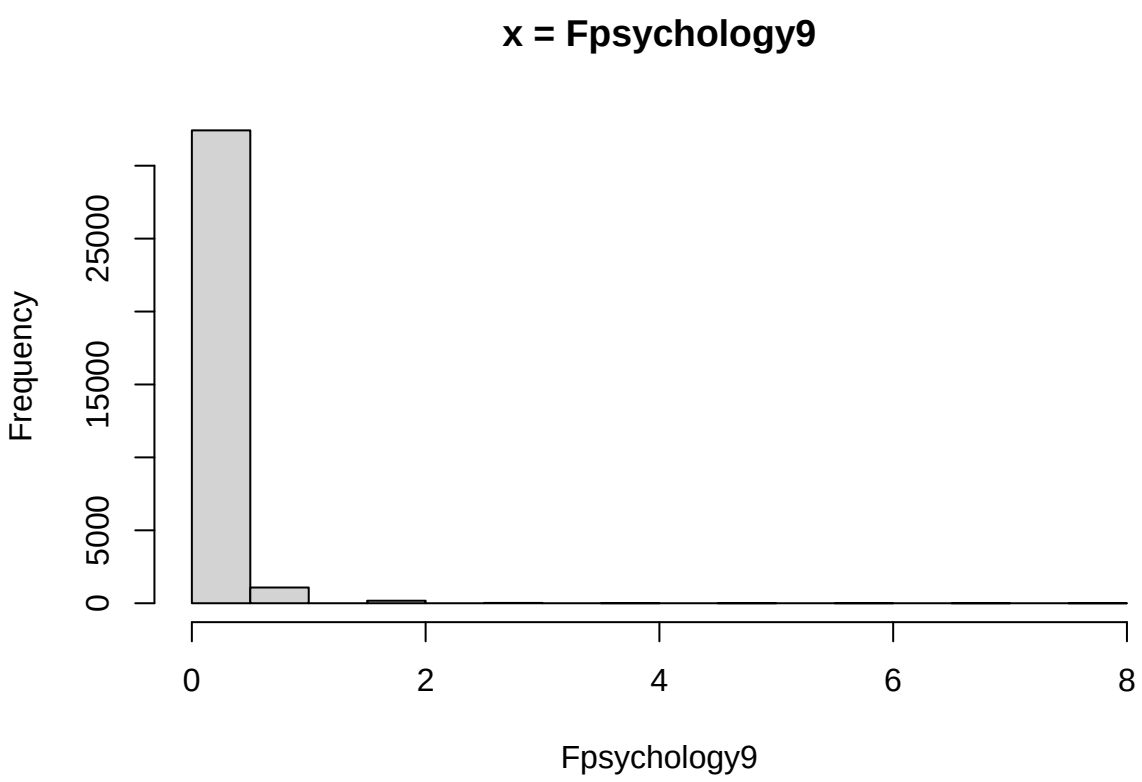

x = Fclassics3
Frequency
0
5000
15000
25000
0
5
10
15
20
25
30
Fclassics3

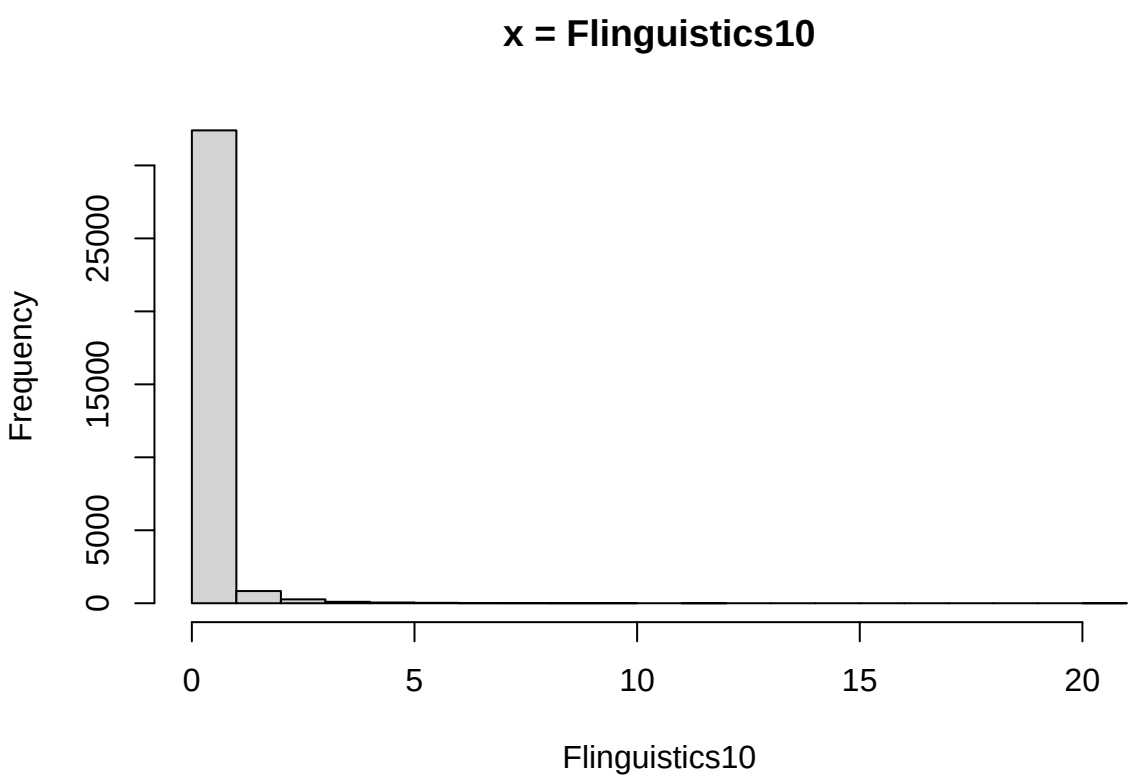

x = Fcartoons5
Frequency
0
10000
20000
30000
0
10
20
30
40
50
Fcartoons5

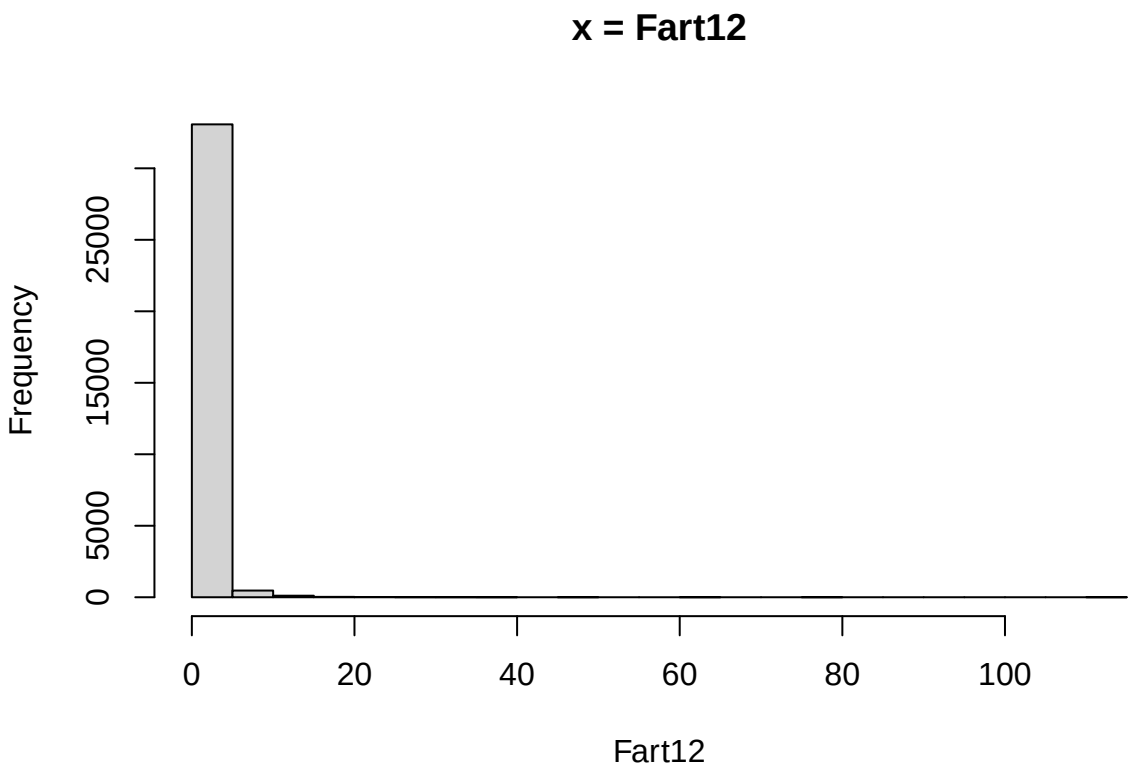

x = Flegends6
Frequency
0
5000
15000
25000
0
5
10
15
20
Flegends6

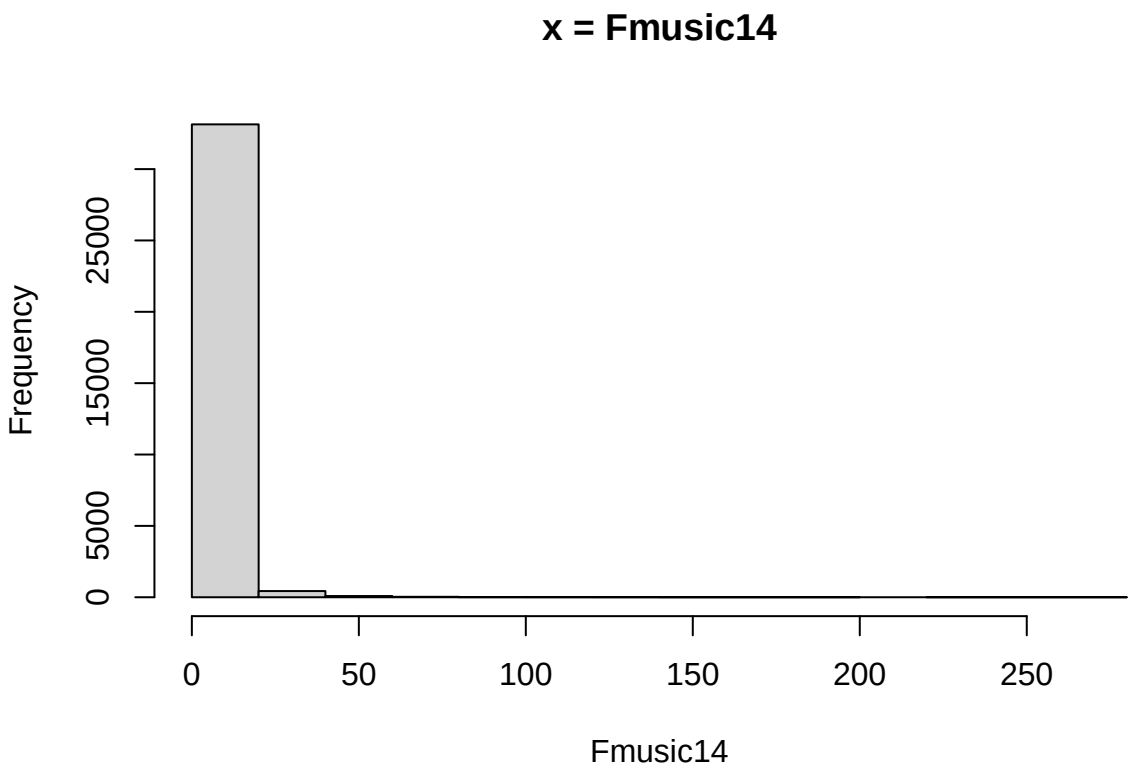

x = Fphilosophy7
Frequency
0
5000
15000
25000
0
5
10
15
20
25
30
35
Fphilosophy7

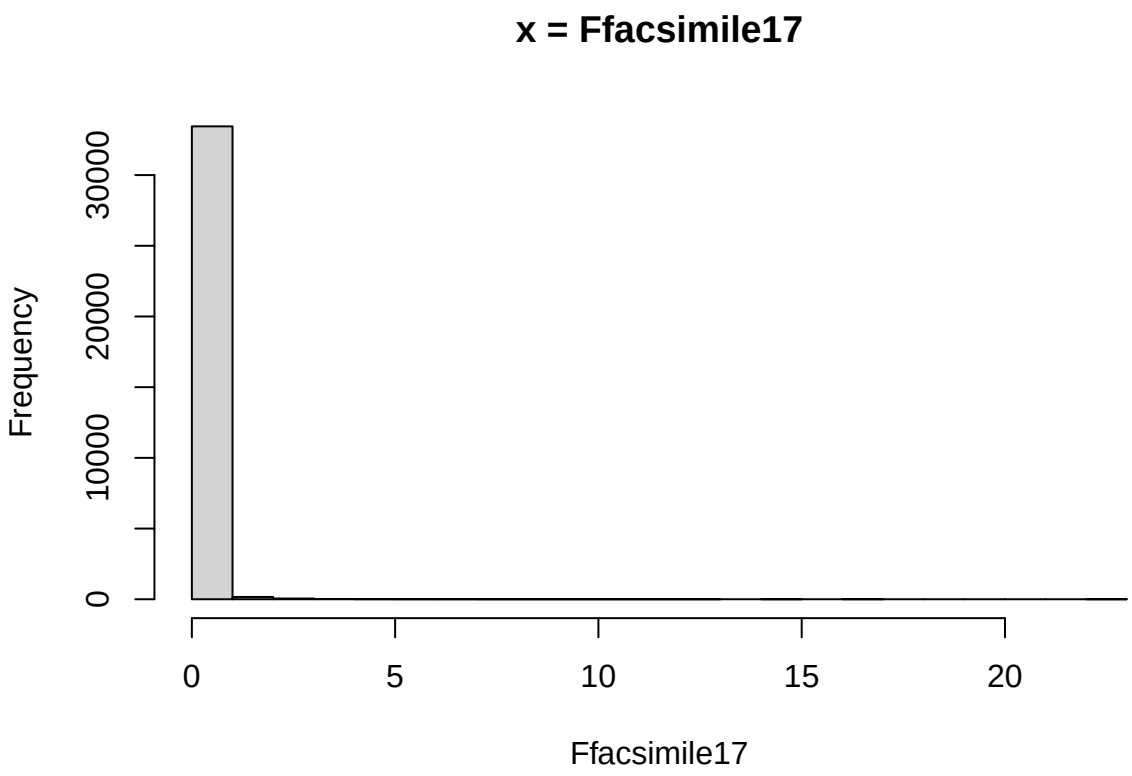

x = Freligion8
Frequency
0
5000
15000
25000
0
20
40
60
80
100
120
140
Freligion8

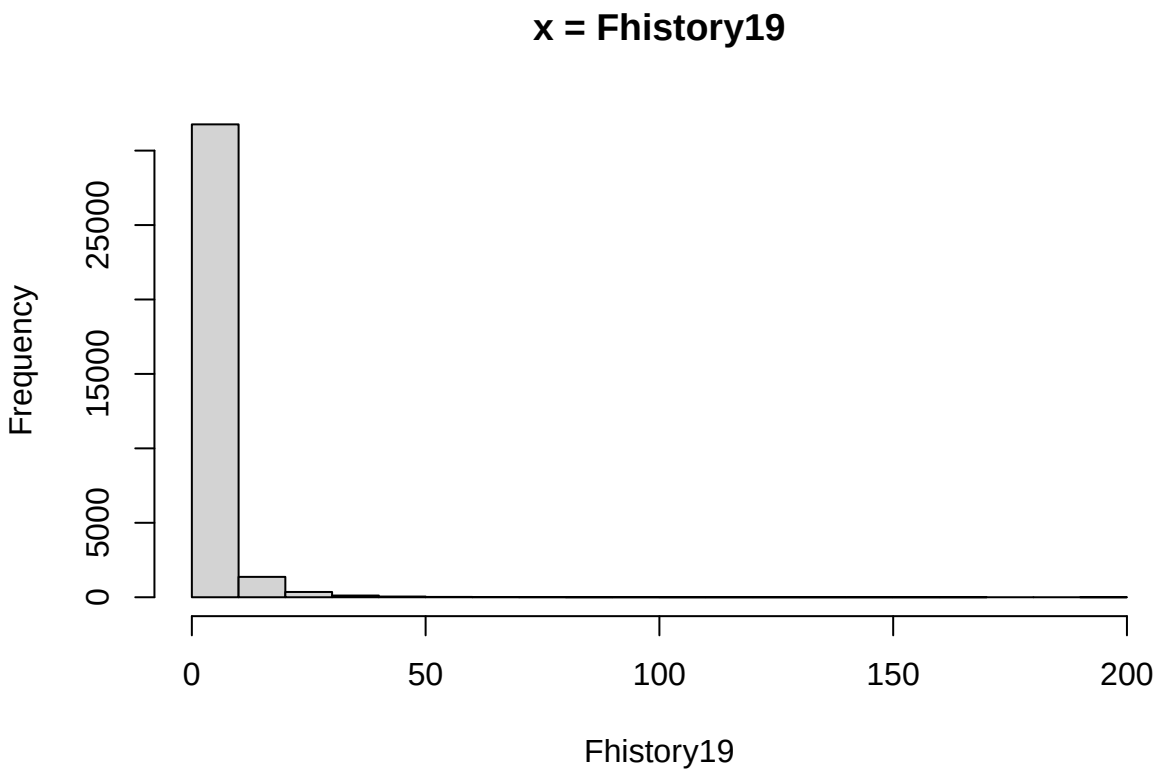

x = Fpsychology9
Frequency
0
5000
15000
25000
0
2
4
6
8
Fpsychology9

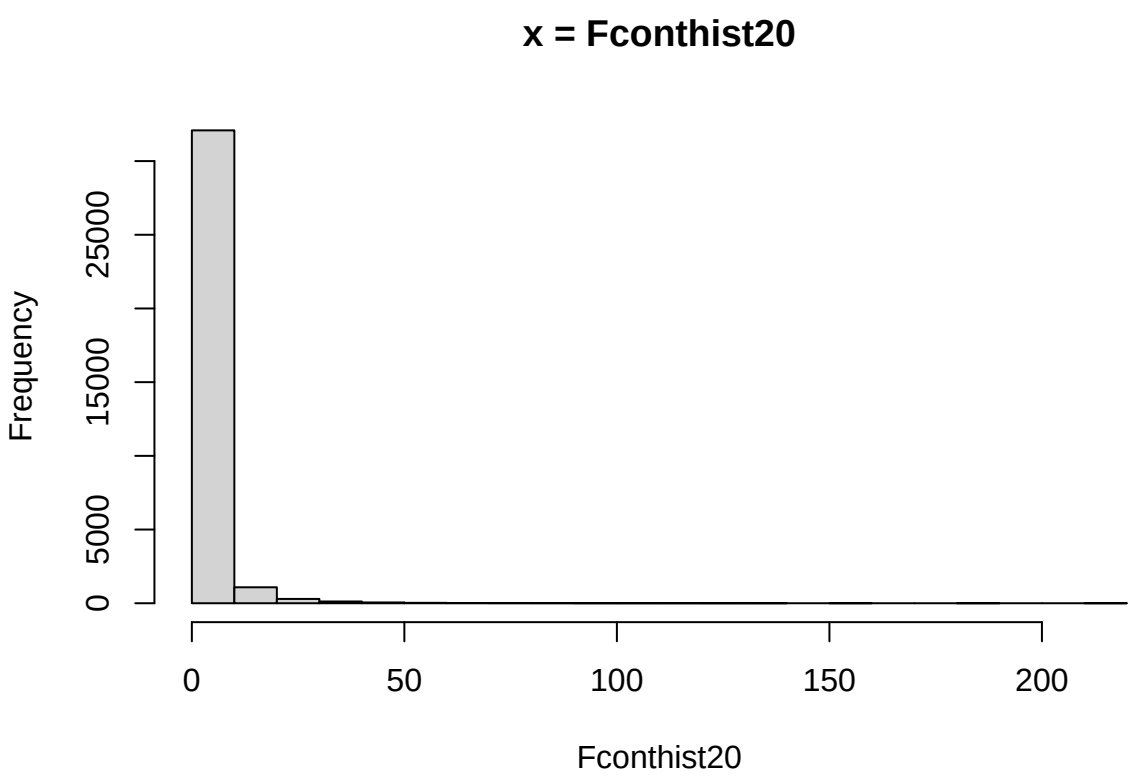

x = Flinguistics10
Frequency
0
5000
15000
25000
0
5
10
15
20
Flinguistics10

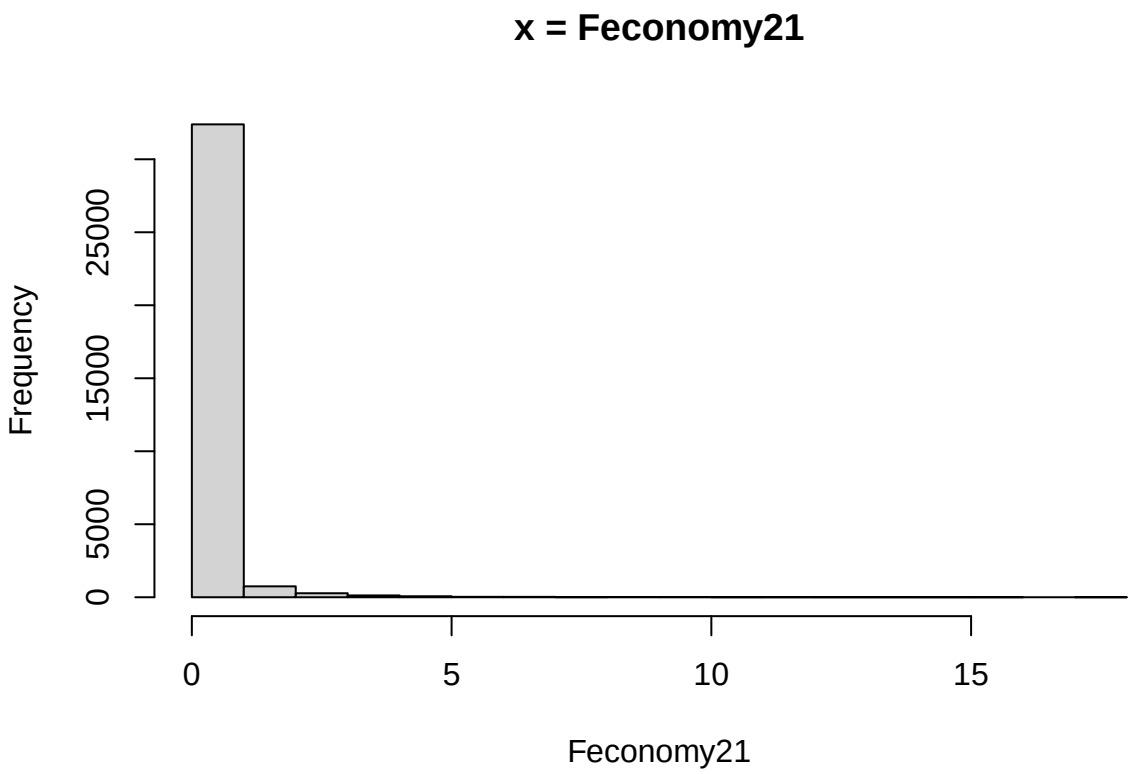

x = Fart12
Frequency
0
5000
15000
25000
0
20
40
60
80
100
Fart12

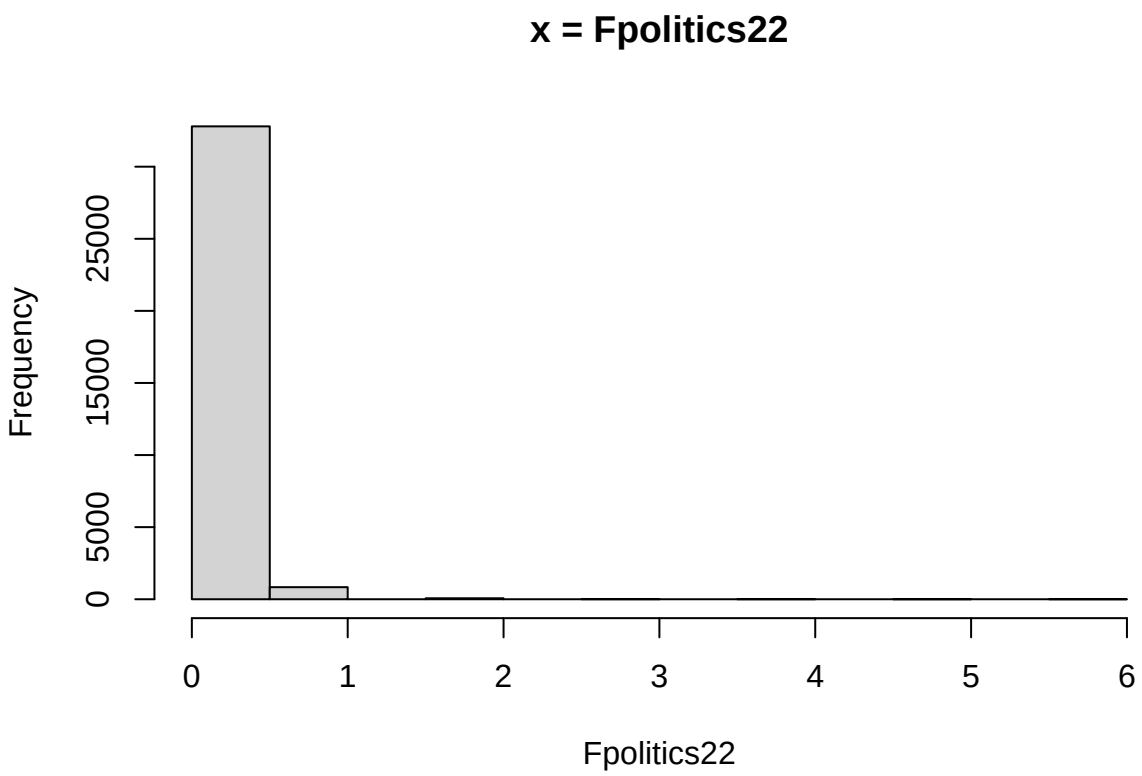

x = Fmusic14
Frequency
0
5000
15000
25000
0
50
100
150
200
250
Fmusic14

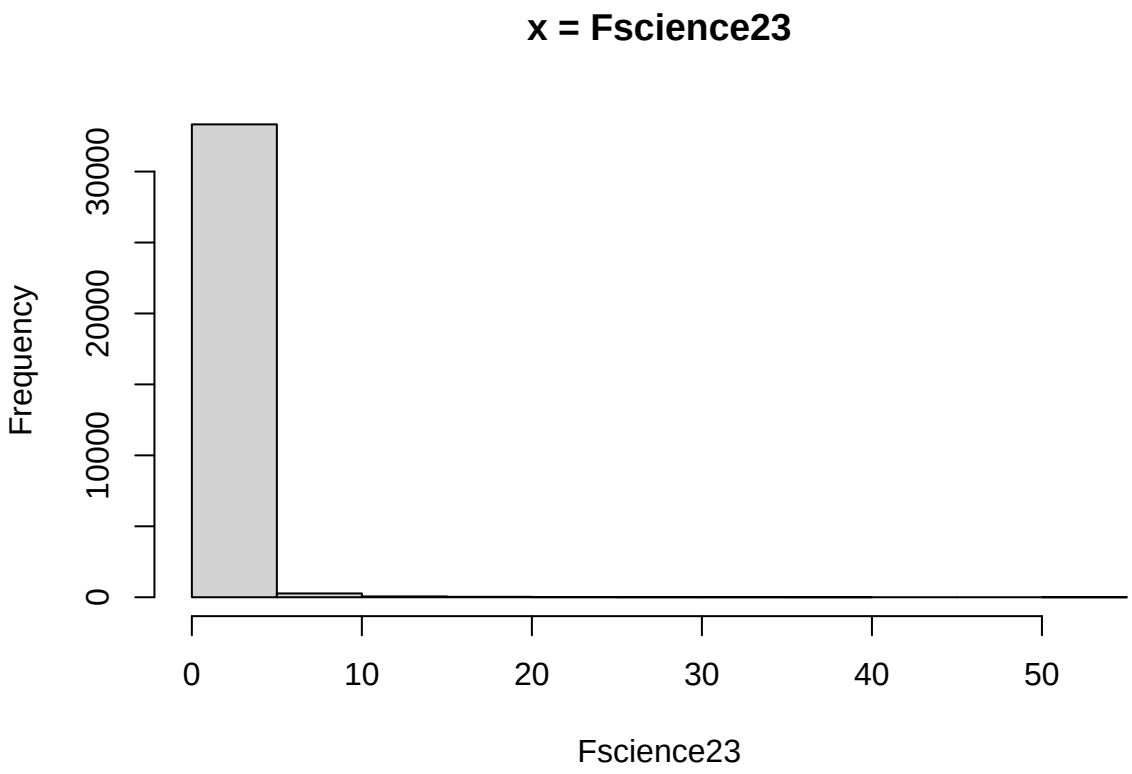

x = Ffacsimile17
Frequency
0
10000
20000
30000
0
5
10
15
20
Ffacsimile17

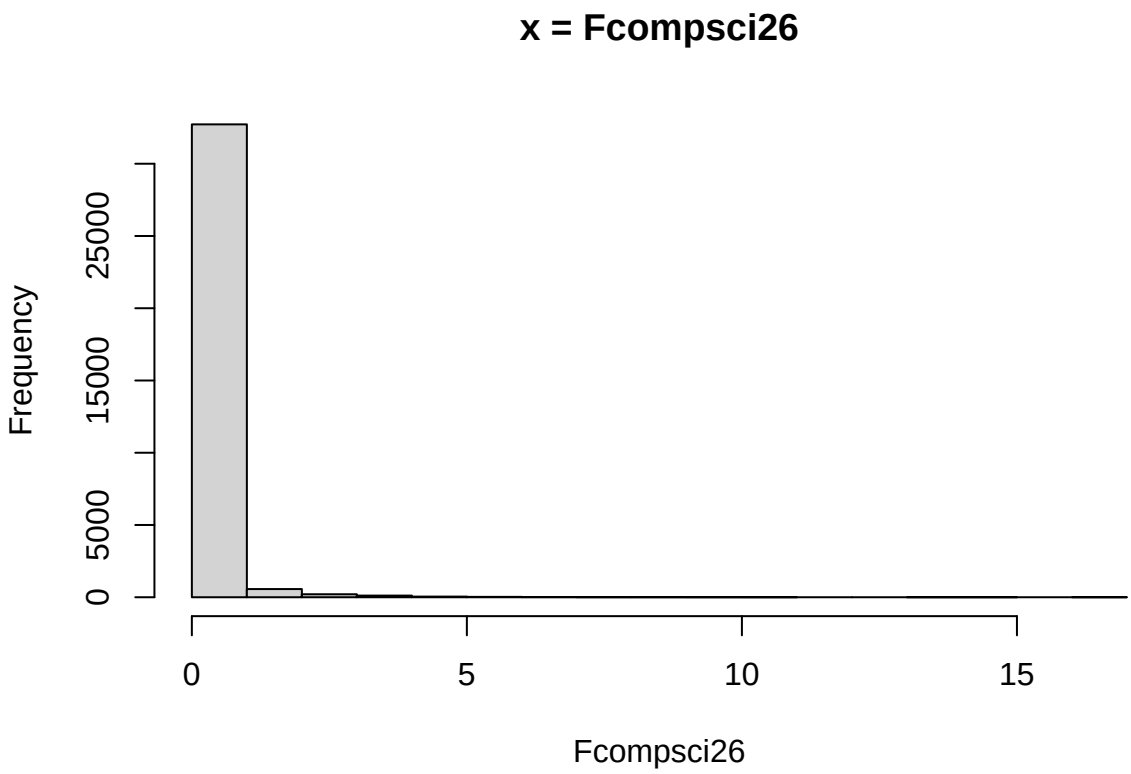

x = Fhistory19
Frequency
0
5000
15000
25000
0
50
100
150
200
Fhistory19

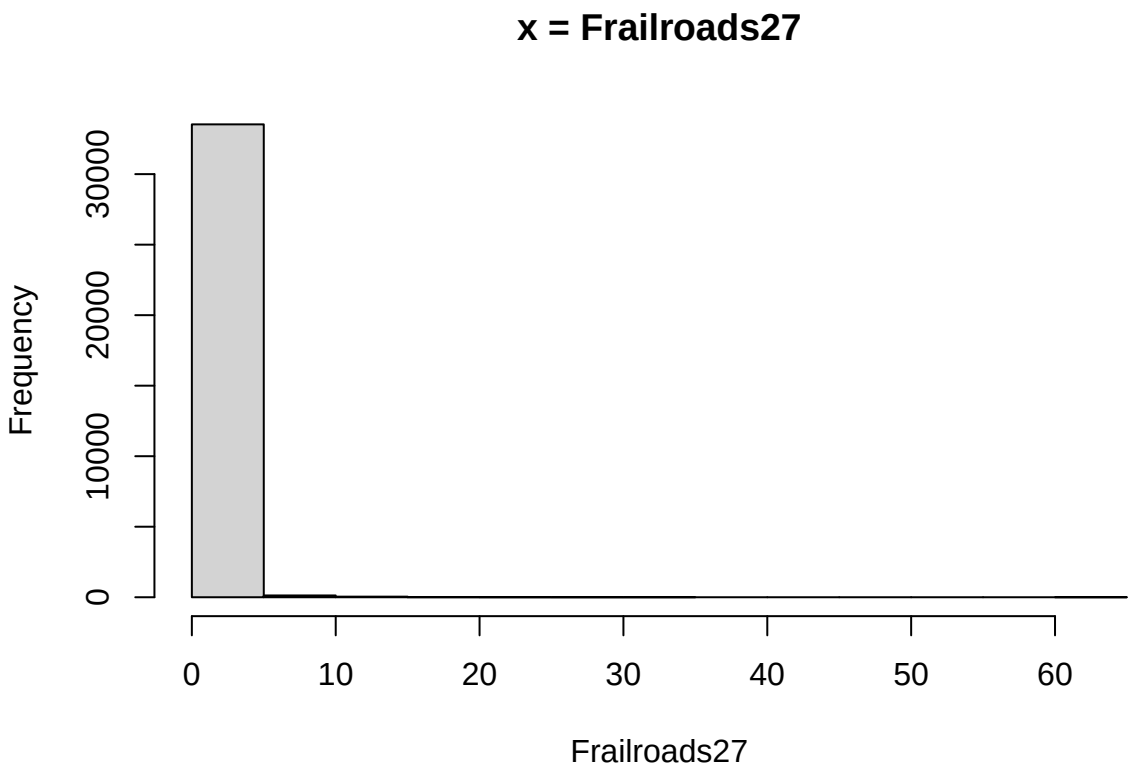

x = Fconthist20
Frequency
0
5000
15000
25000
0
50
100
150
200
Fconthist20

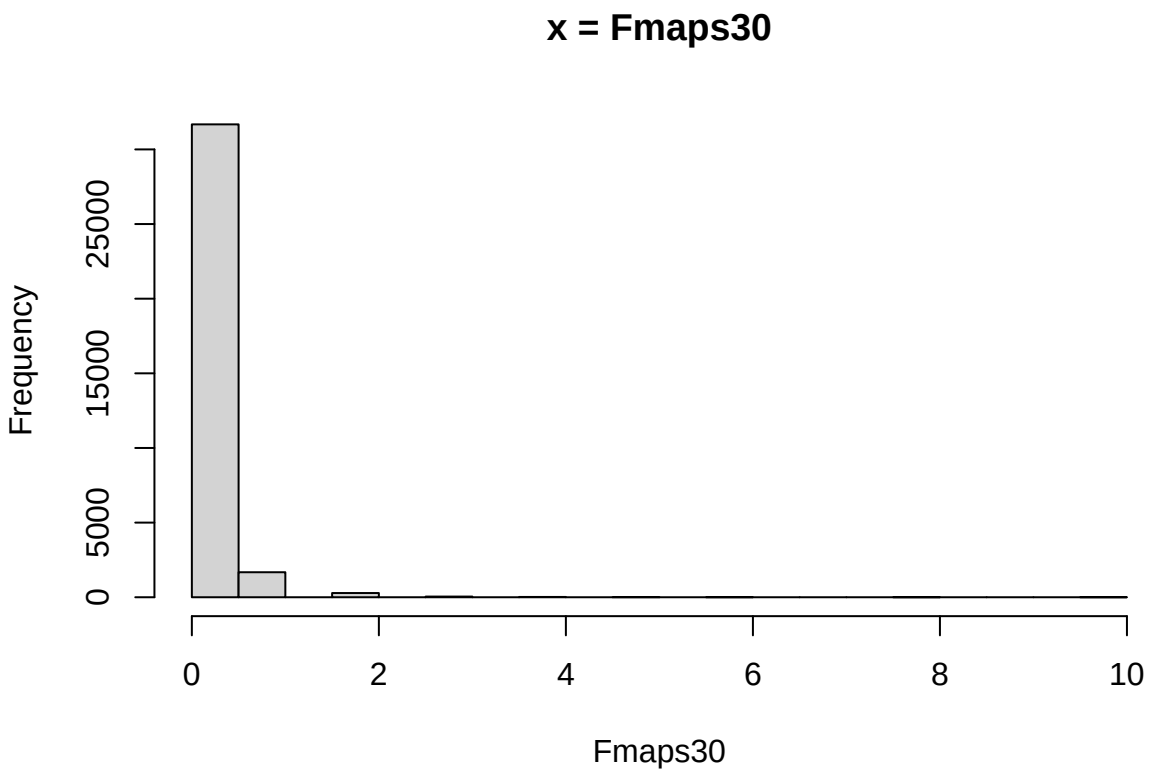

x = Feconomy21
Frequency
0
5000
15000
25000
0
5
10
15
Feconomy21

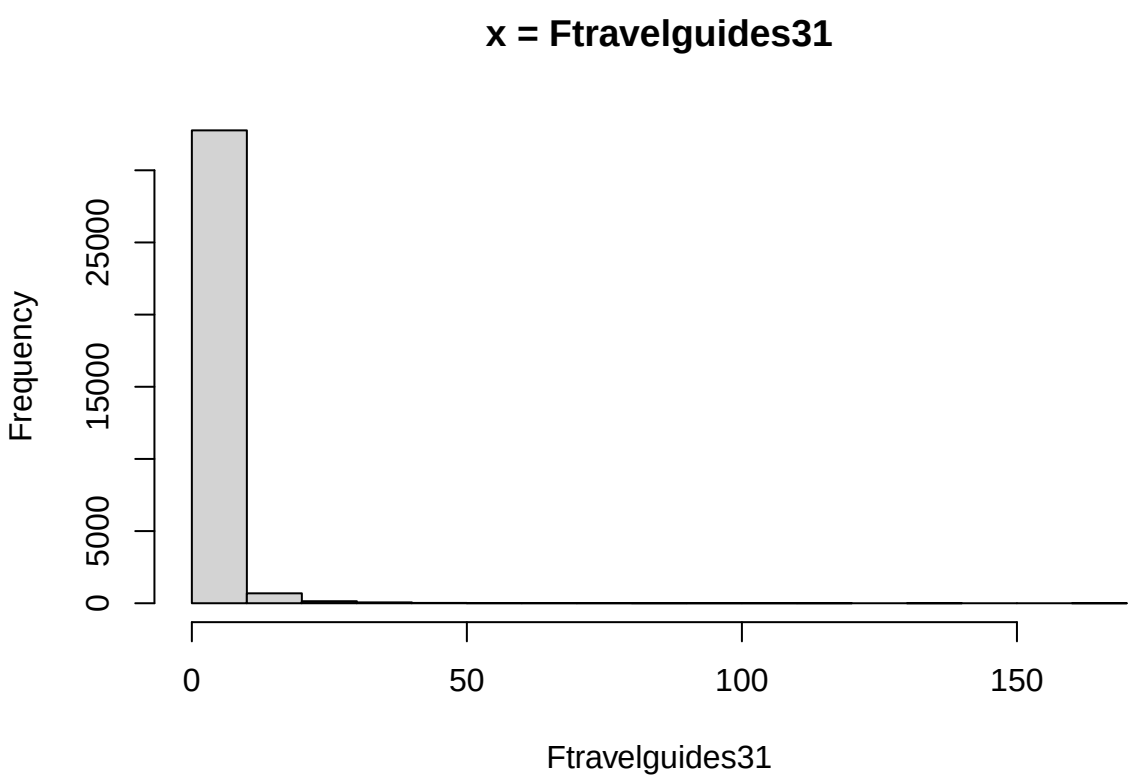

x = Fpolitics22
Frequency
0
5000
15000
25000
0
1
2
3
4
5
6
Fpolitics22

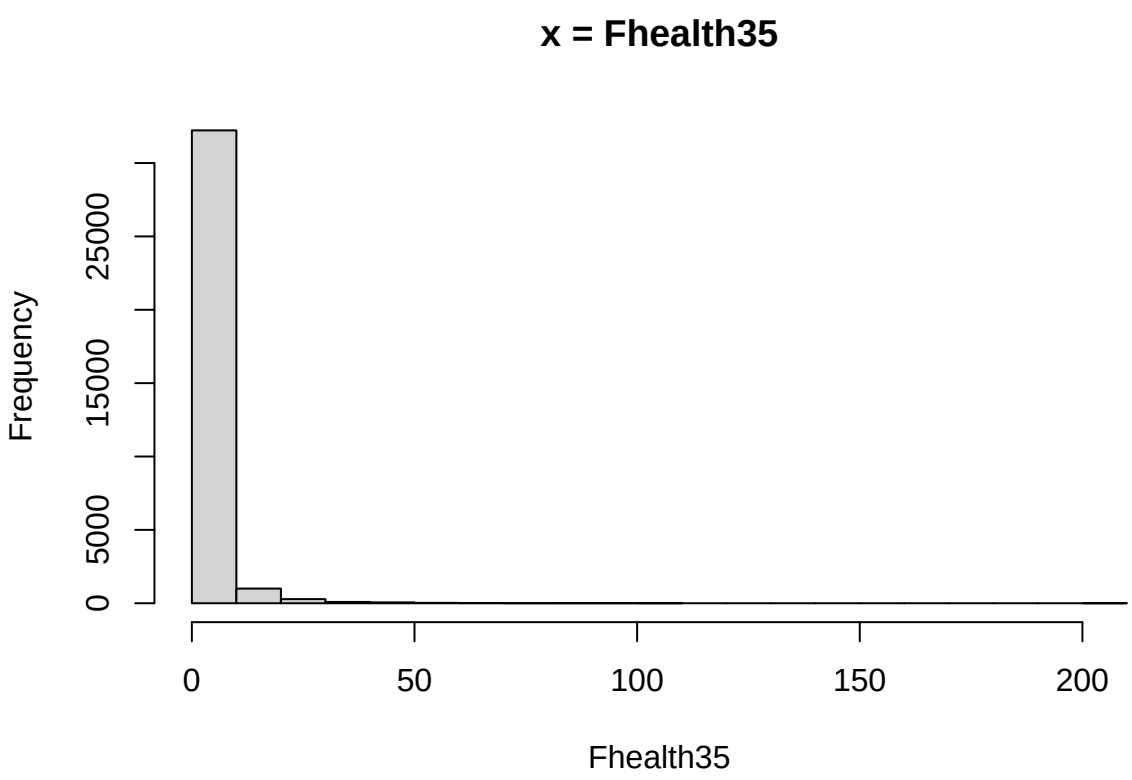

x = Fscience23
Frequency
0
10000
20000
30000
0
10
20
30
40
50
Fscience23

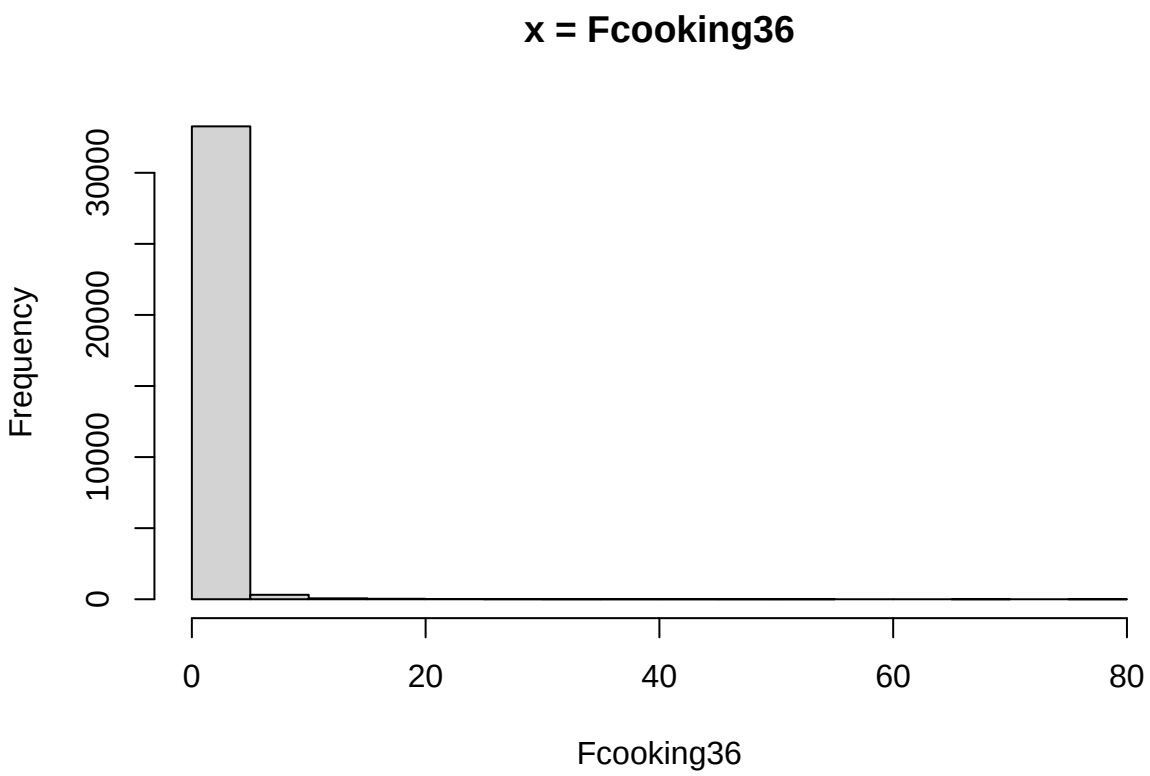

x = Fcompsci26
Frequency
0
5000
15000
25000
0
5
10
15
Fcompsci26

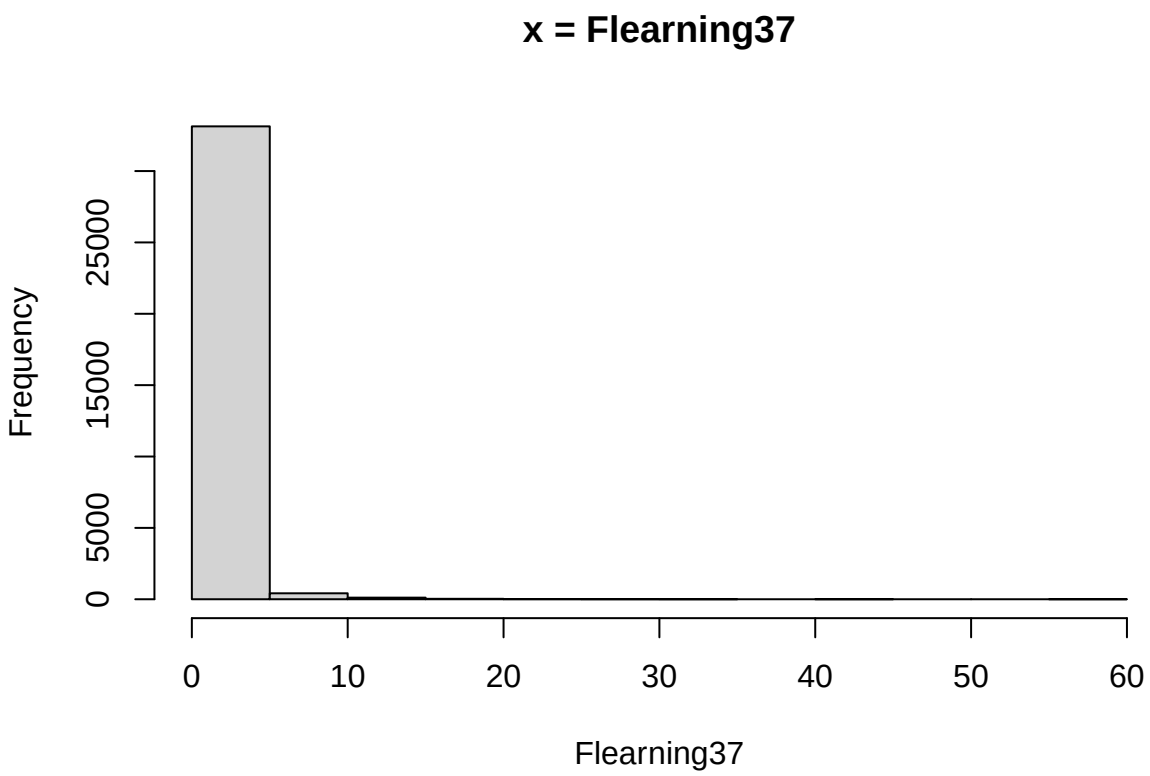

x = Frailroads27
Frequency
0
10000
20000
30000
0
10
20
30
40
50
60
Frailroads27

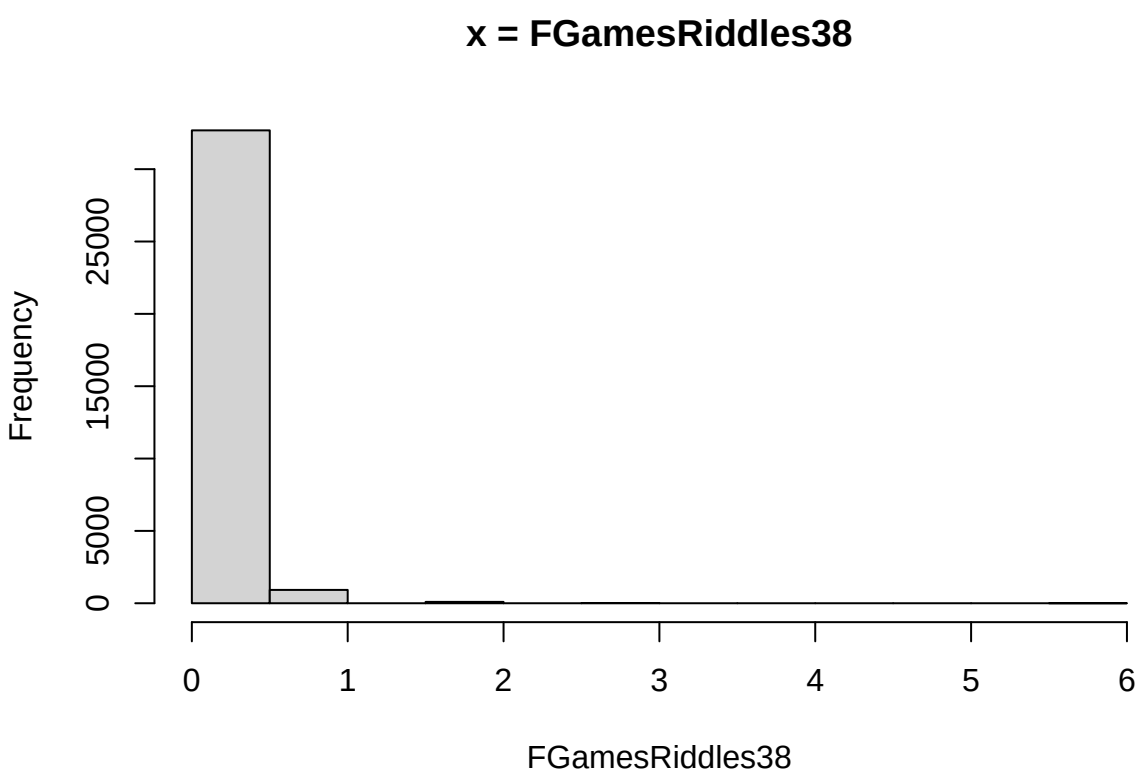

x = Fmaps30
Frequency
0
5000
15000
25000
0
2
4
6
8
10
Fmaps30

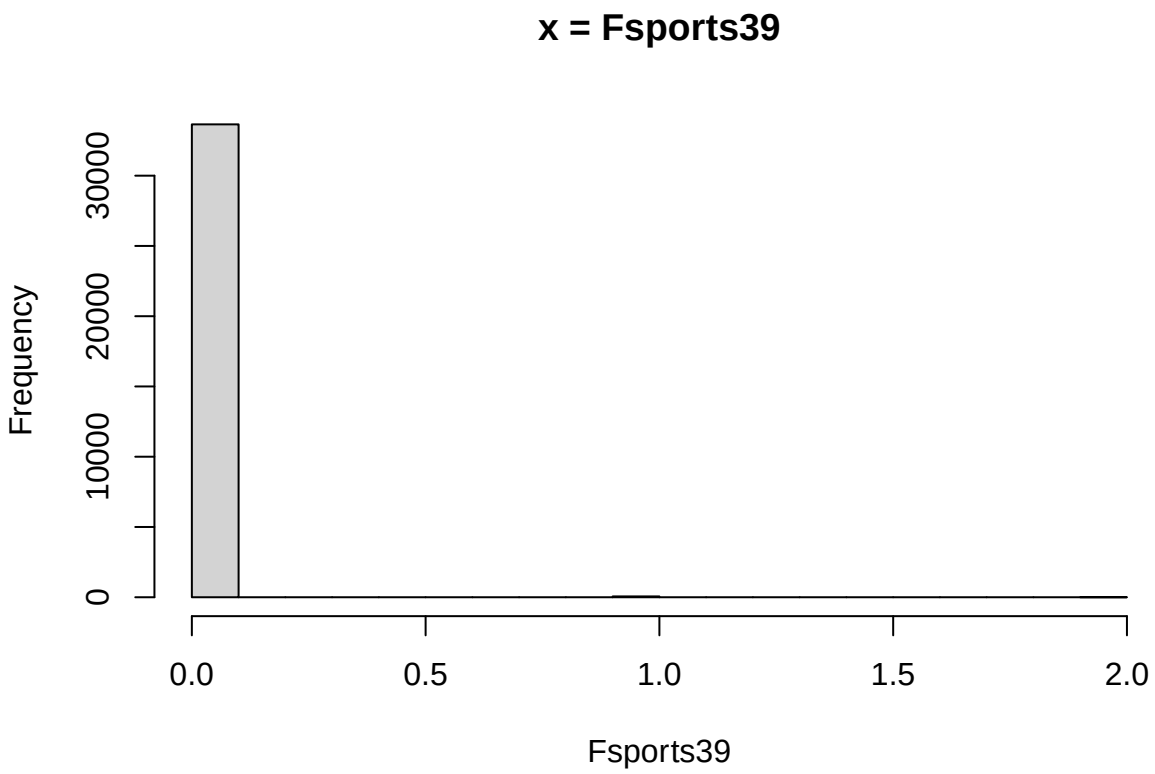

x = Ftravelguides31
Frequency
0
5000
15000
25000
0
50
100
150
Ftravelguides31

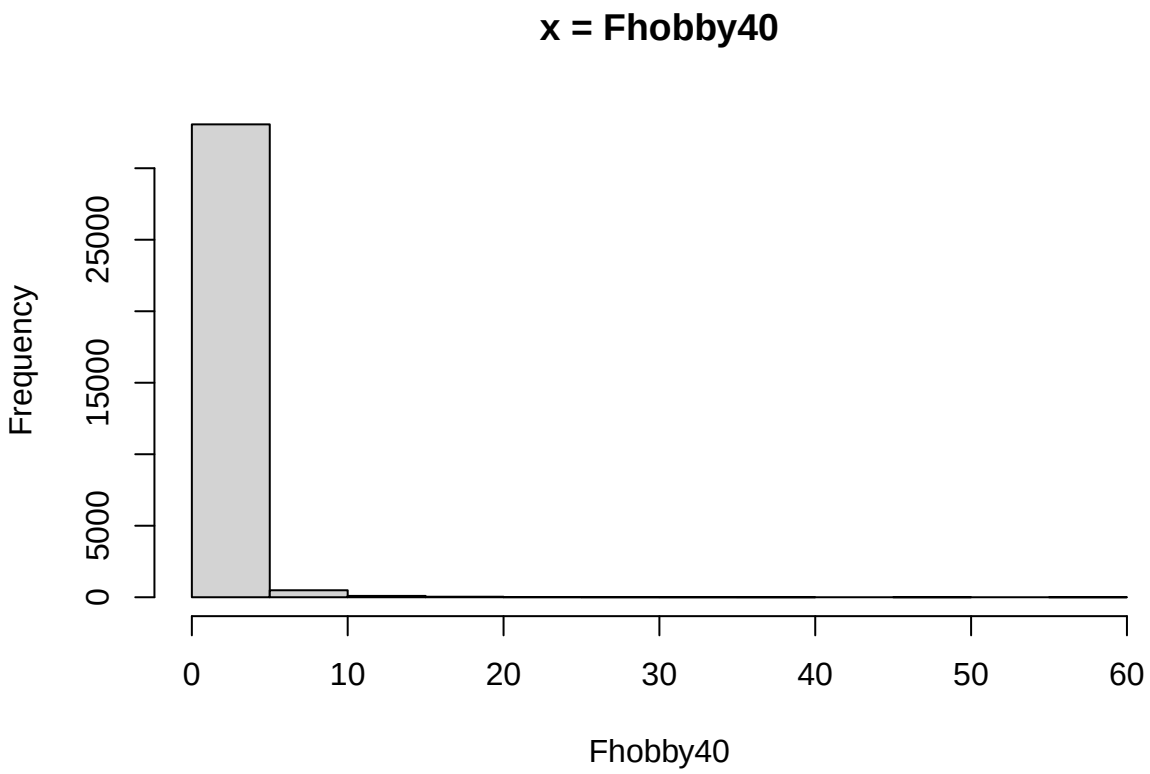

x = Fhealth35
Frequency
0
5000
15000
25000
0
50
100
150
200
Fhealth35

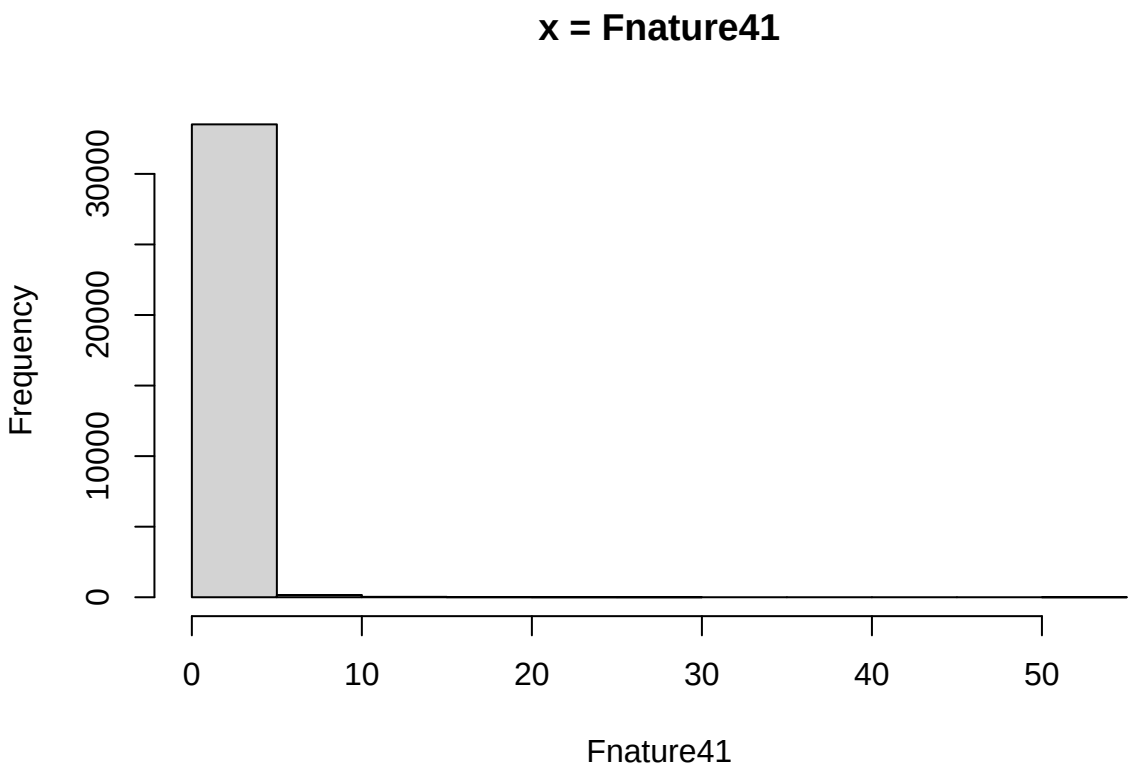

x = Fcooking36
Frequency
0
10000
20000
30000
0
20
40
60
80
Fcooking36

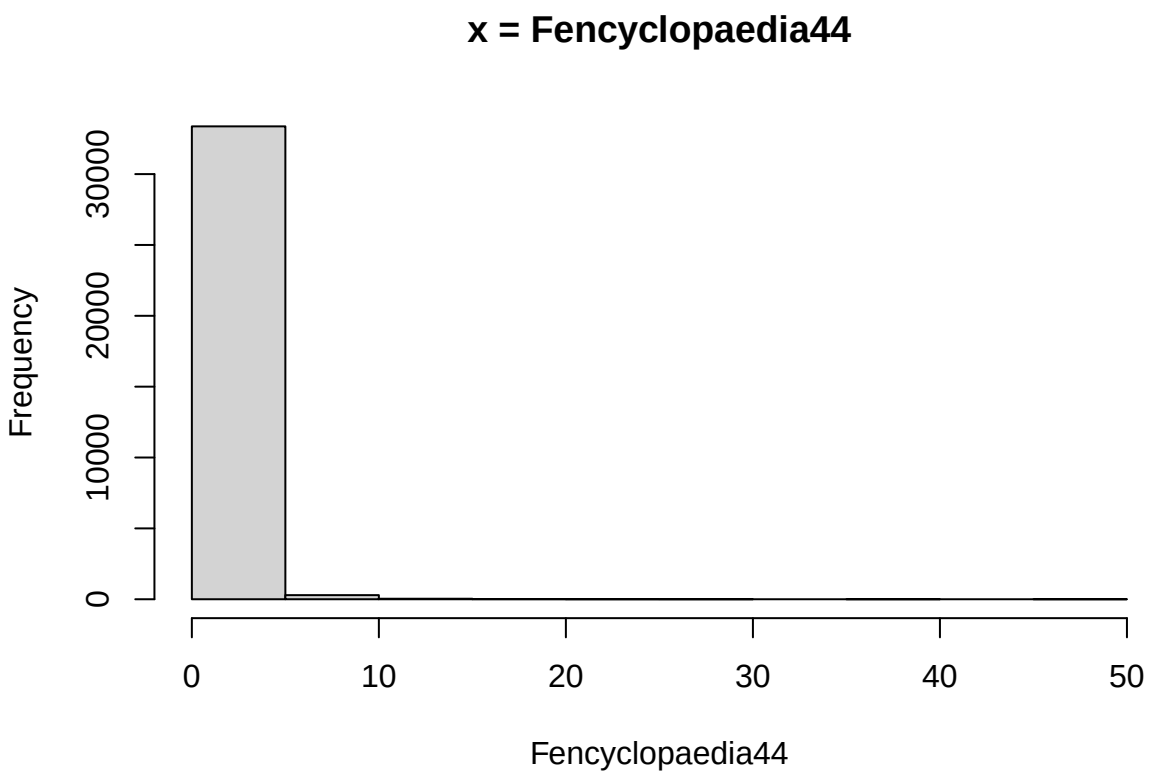

x = Flearning37
Frequency
0
5000
15000
25000
0
10
20
30
40
50
60
Flearning37

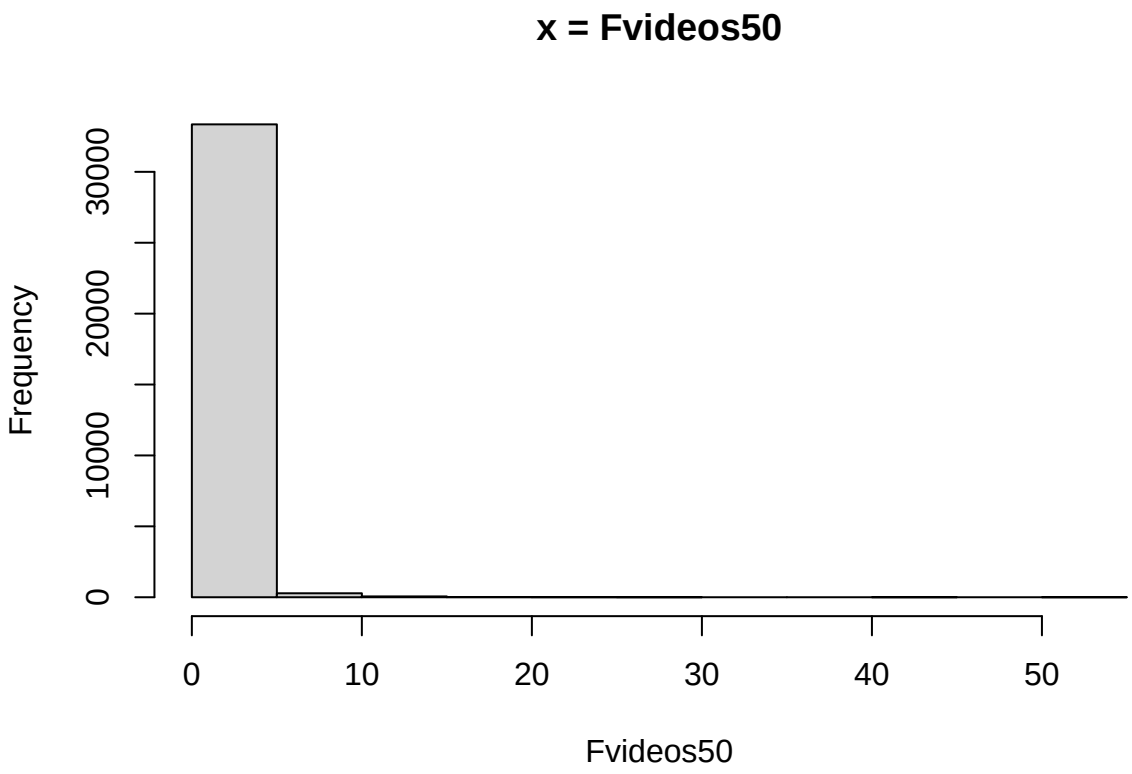

x = FGamesRiddles38
Frequency
0
5000
15000
25000
0
1
2
3
4
5
6
FGamesRiddles38

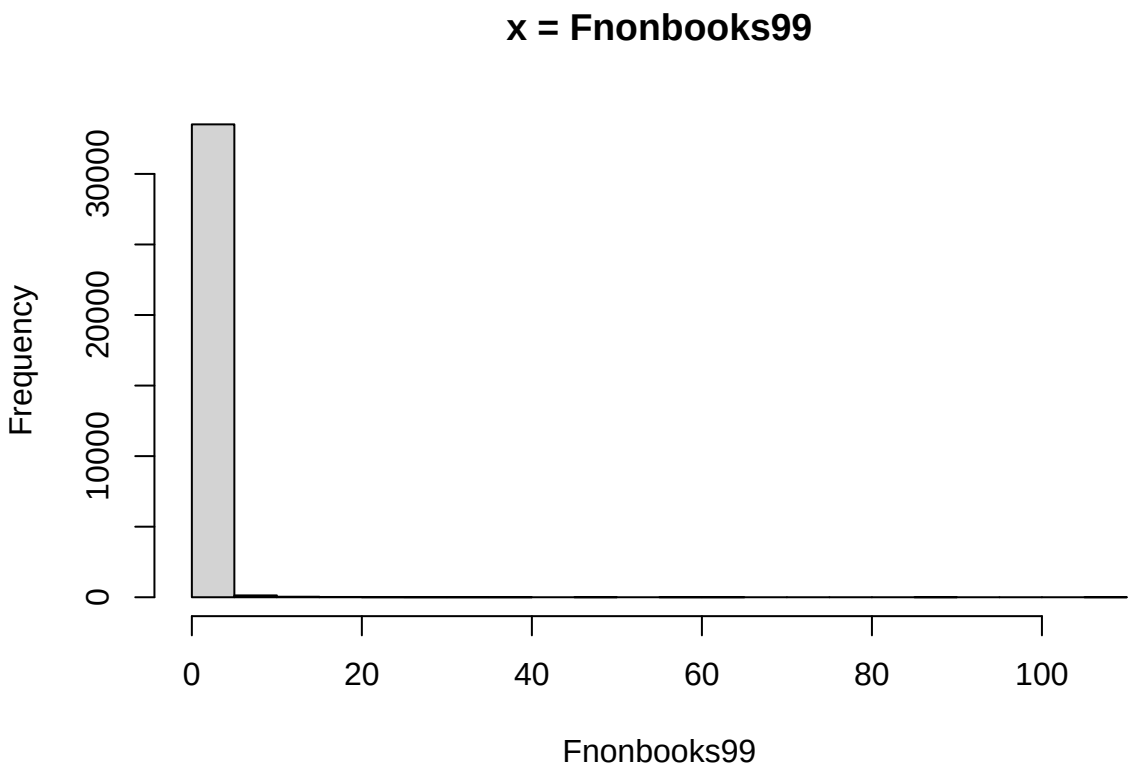

x = Fsports39
Frequency
0
10000
20000
30000
0.0
0.5
1.0
1.5
2.0
Fsports39

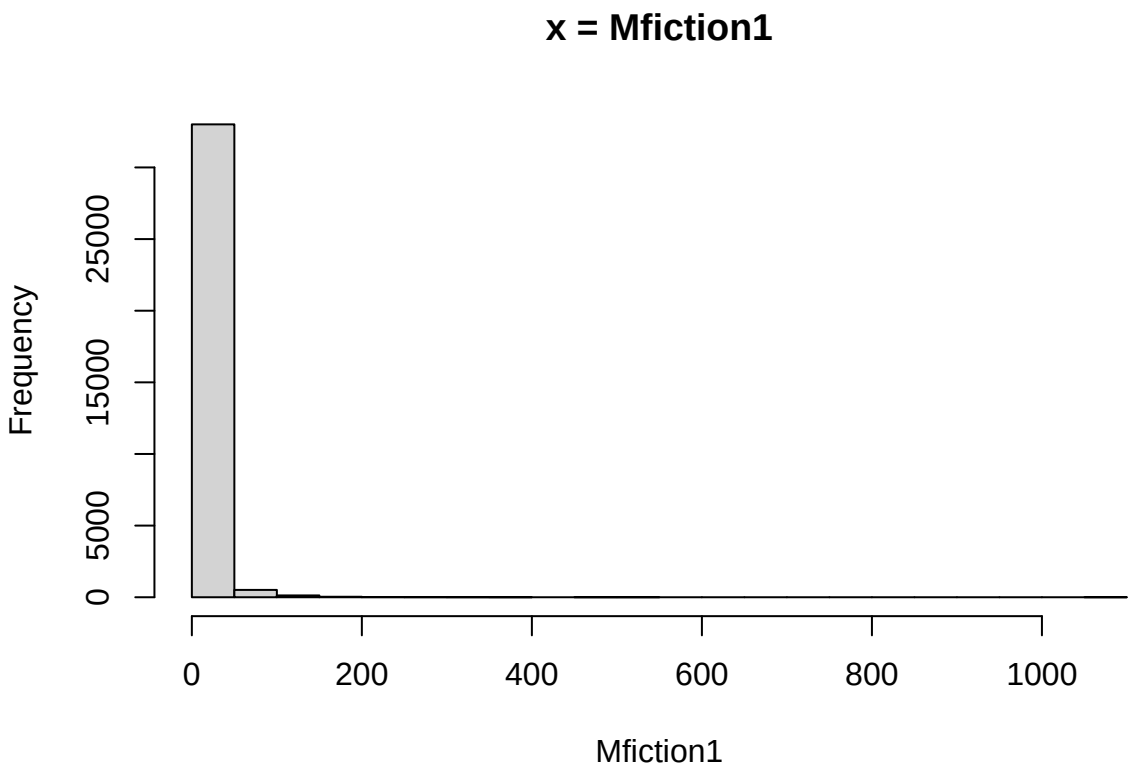

x = Fhobby40
Frequency
0
5000
15000
25000
0
10
20
30
40
50
60
Fhobby40

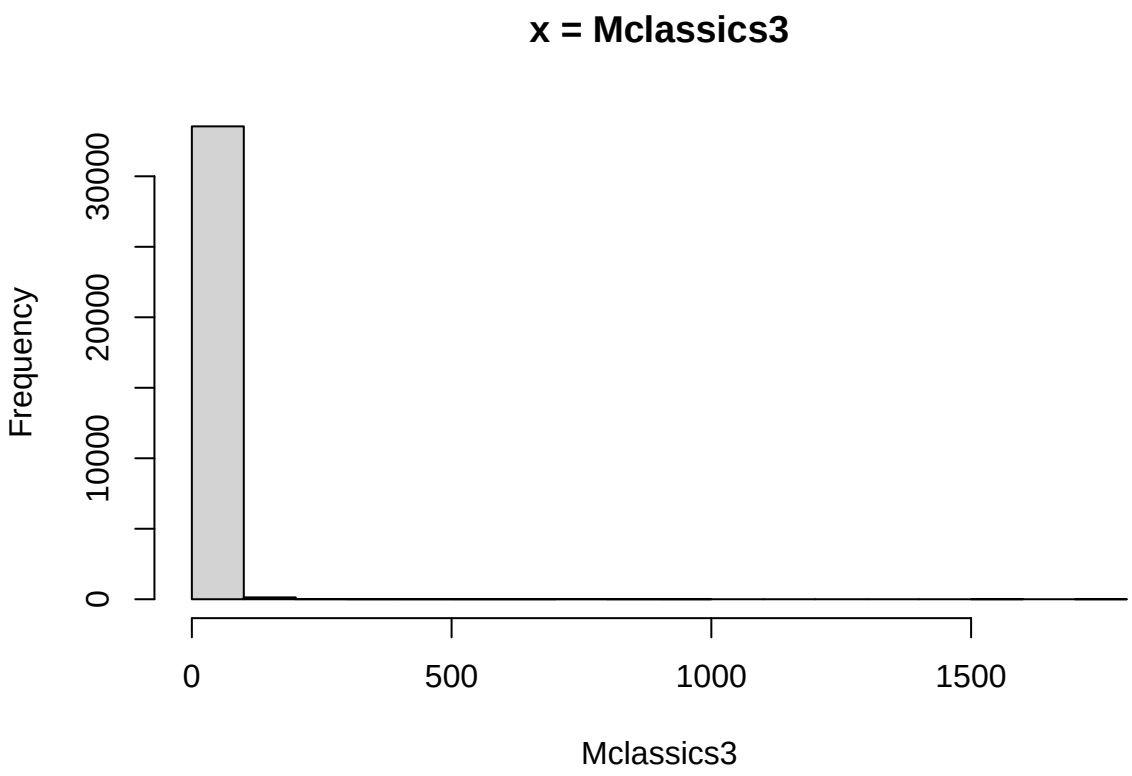

x = Fnature41
Frequency
0
10000
20000
30000
0
10
20
30
40
50
Fnature41

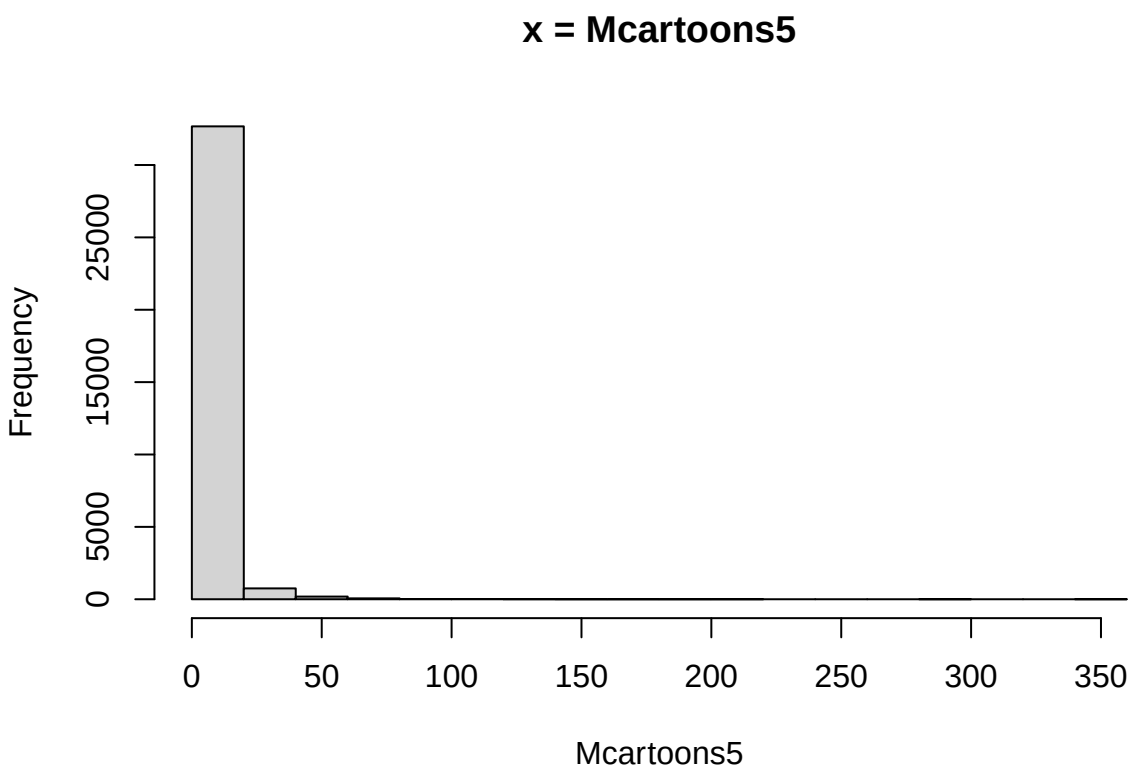

x = Fencyclopaedia44
Frequency
0
10000
20000
30000
0
10
20
30
40
50
Fencyclopaedia44

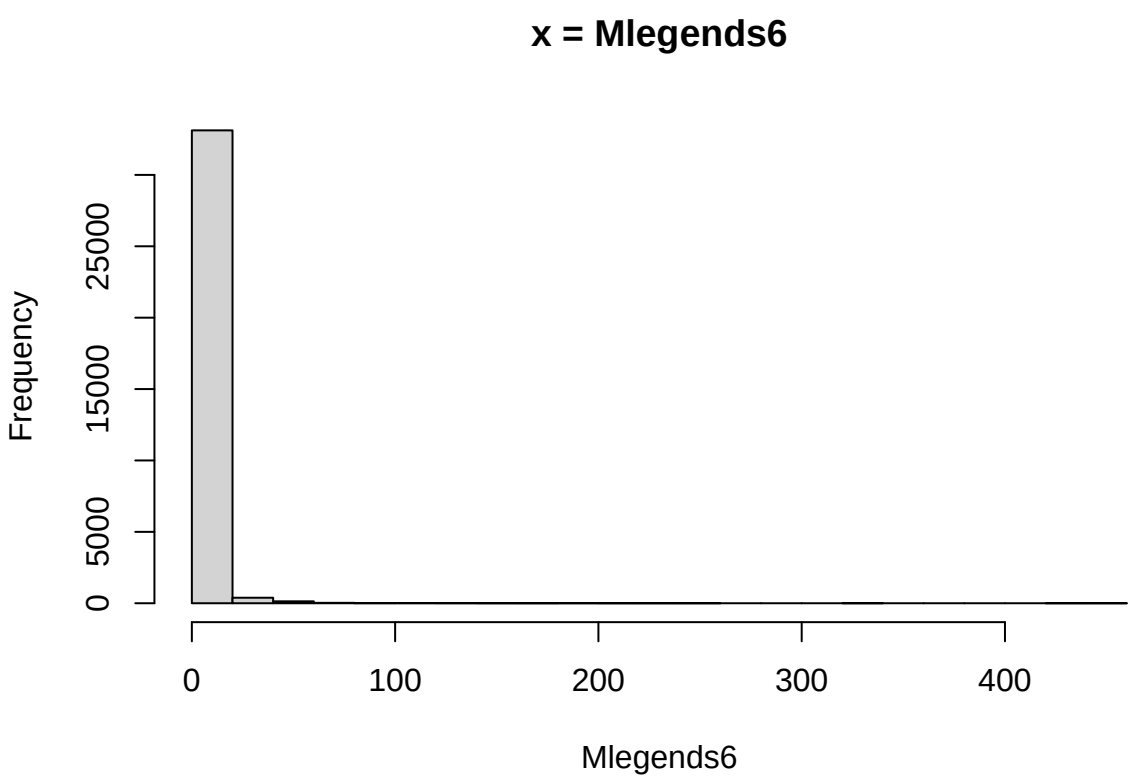

x = Fvideos50
Frequency
0
10000
20000
30000
0
10
20
30
40
50
Fvideos50

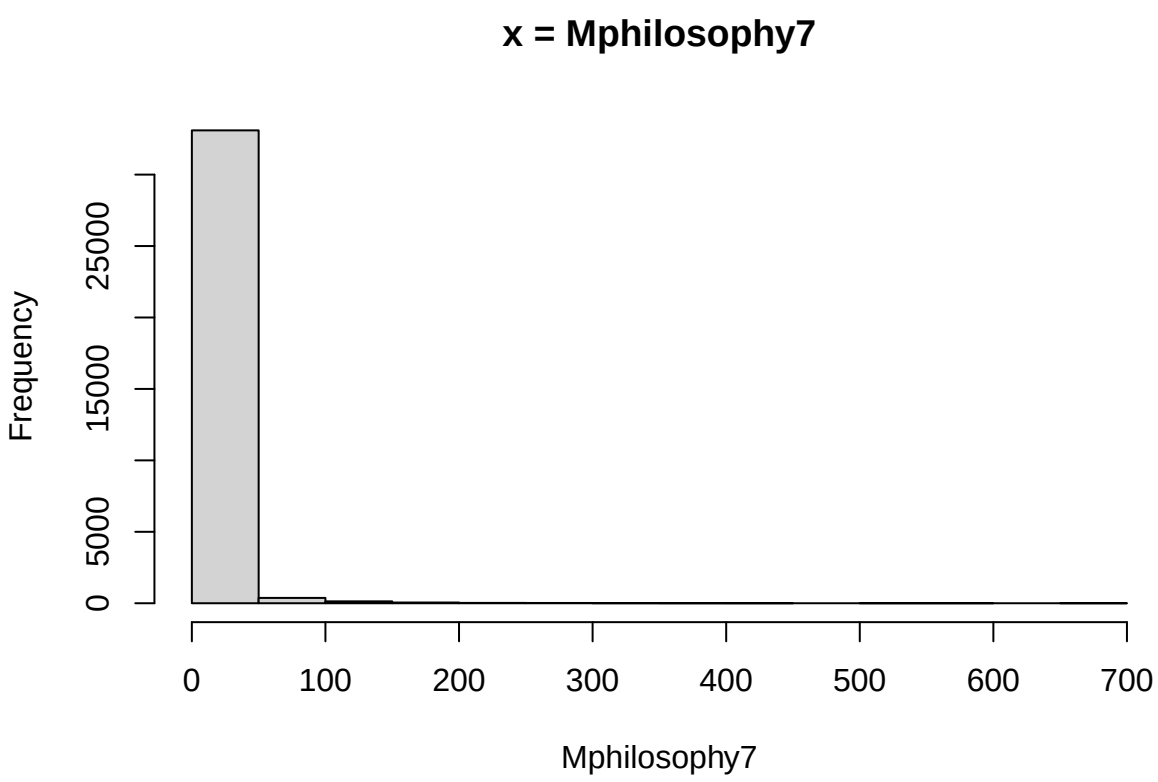

x = Fnonbooks99
Frequency
0
10000
20000
30000
0
20
40
60
80
100
Fnonbooks99

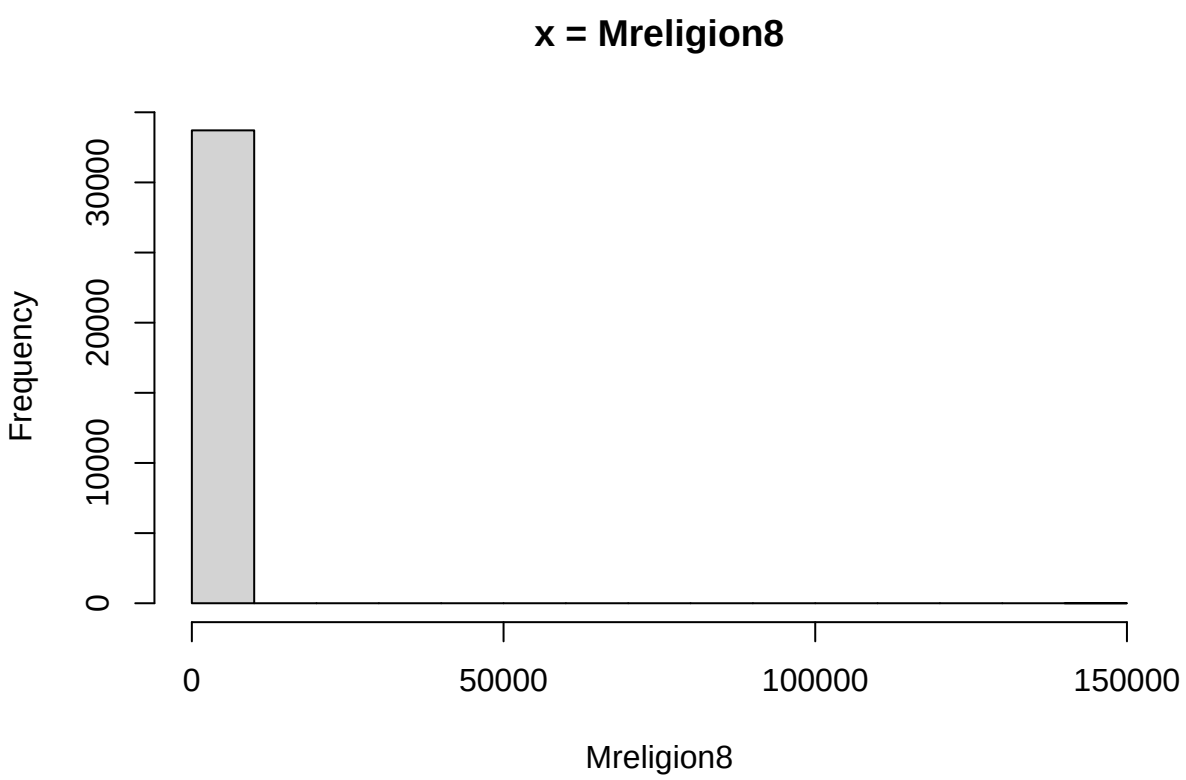

x = Mfiction1
Frequency
0
5000
15000
25000
0
200
400
600
800
1000
Mfiction1

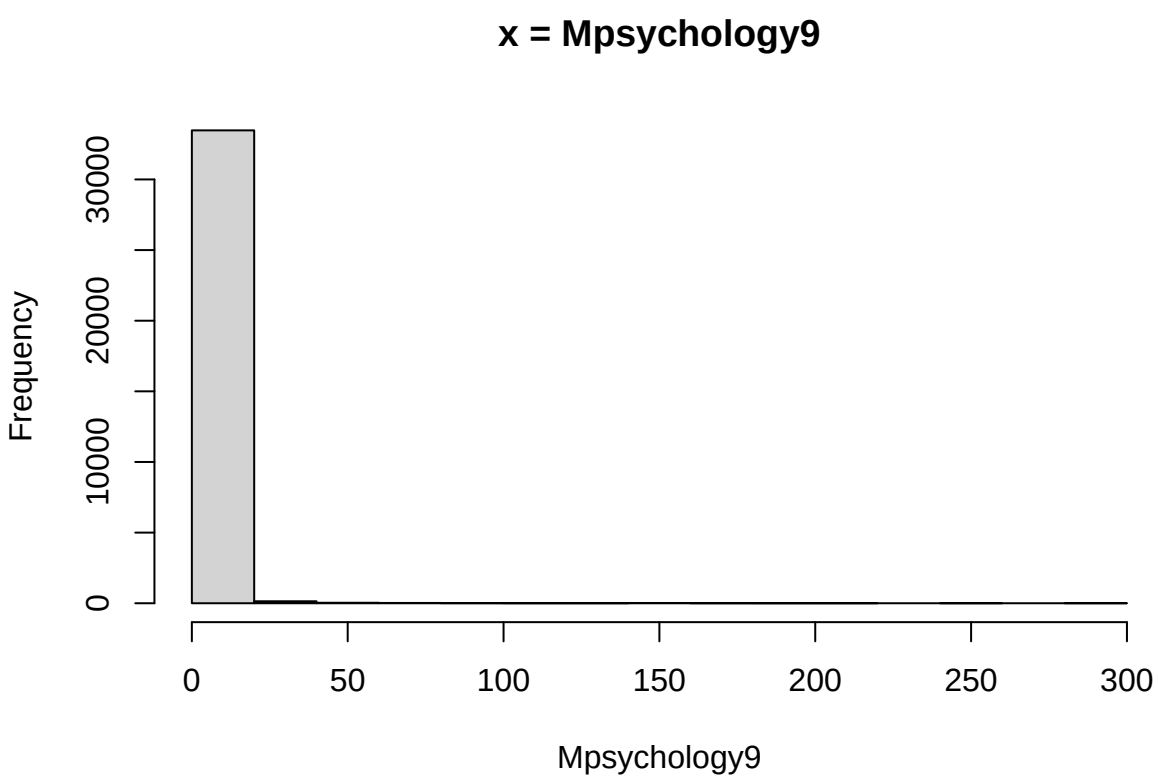

x = Mclassics3
Frequency
0
10000
20000
30000
0
500
1000
1500
Mclassics3

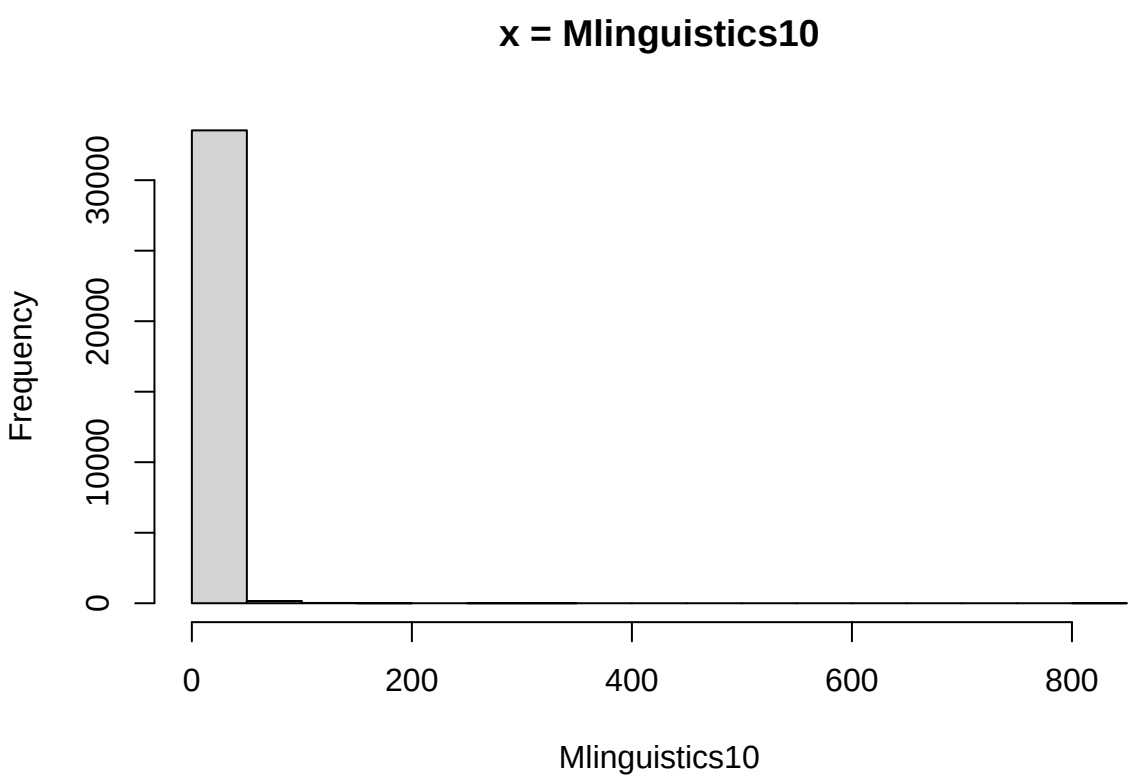

x = Mcartoons5
Frequency
0
5000
15000
25000
0
50
100
150
200
250
300
350
Mcartoons5

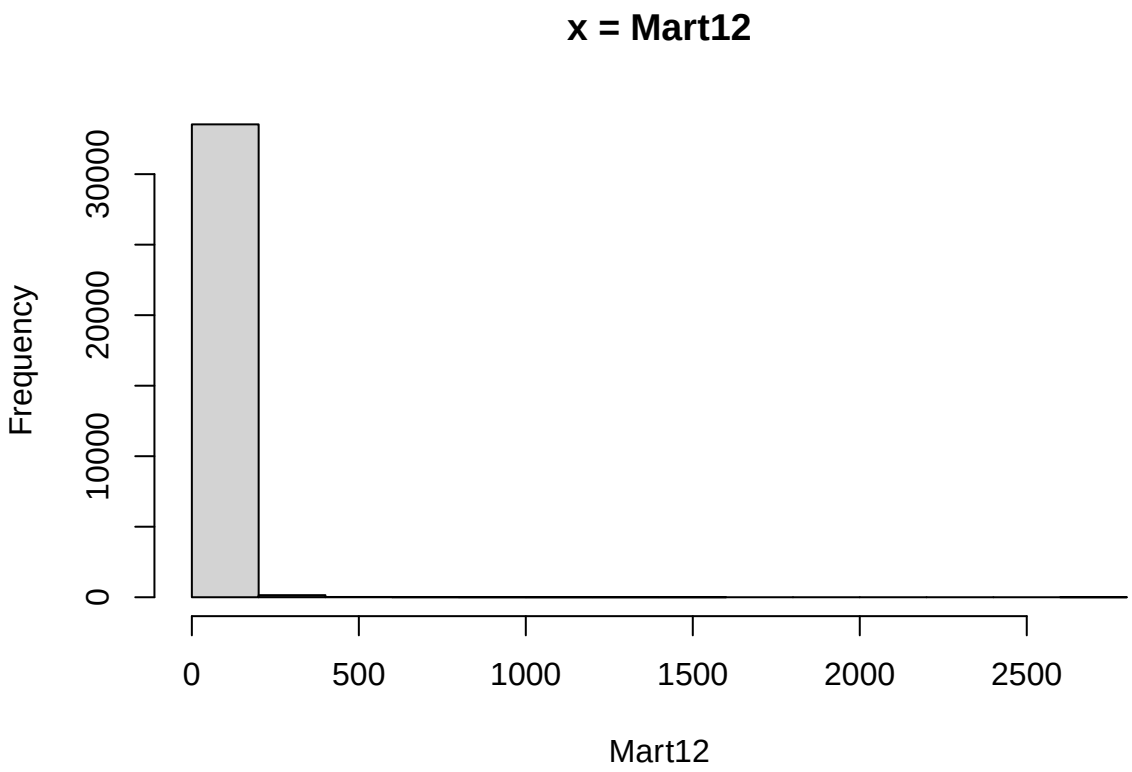

x = Mlegends6
Frequency
0
5000
15000
25000
0
100
200
300
400
Mlegends6

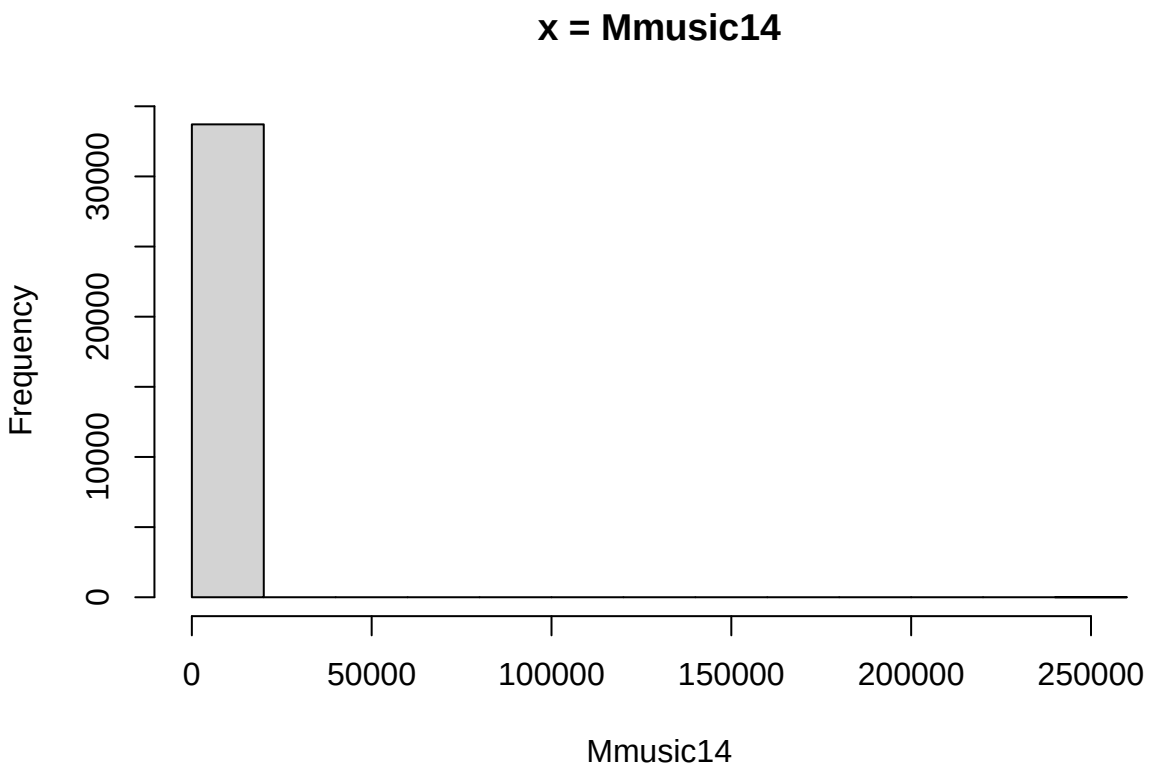

x = Mphilosophy7
Frequency
0
5000
15000
25000
0
100
200
300
400
500
600
700
Mphilosophy7

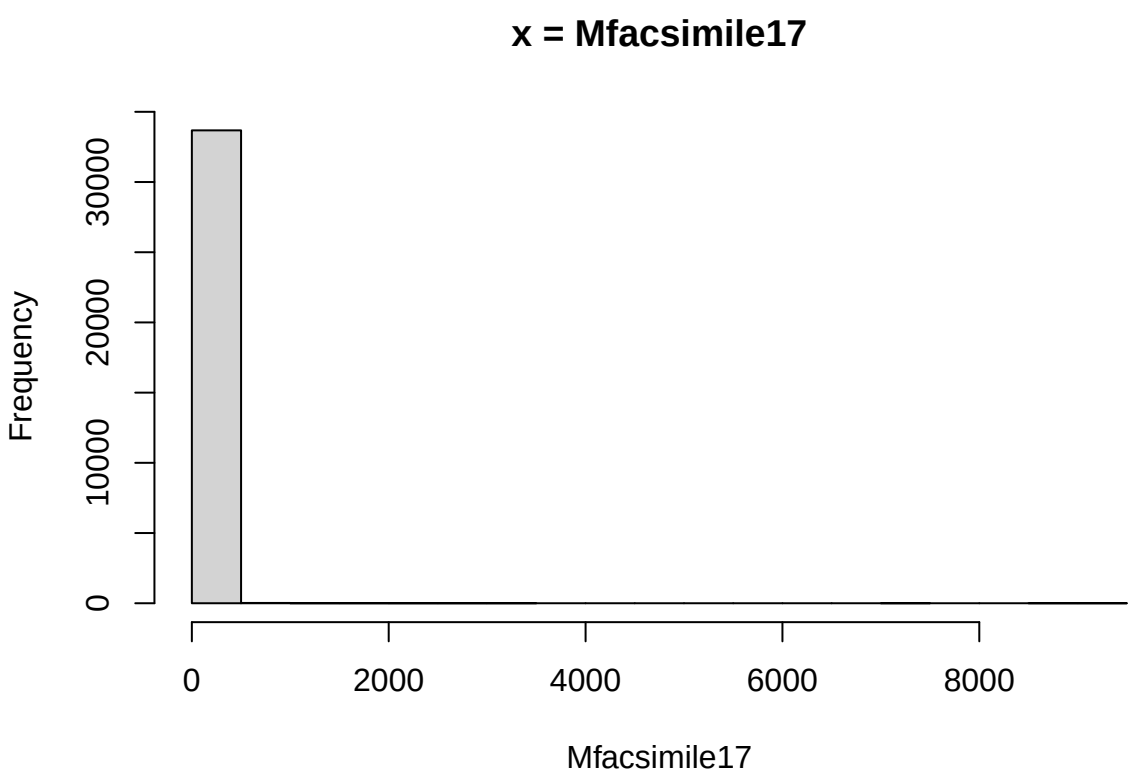

x = Mreligion8
Frequency
0
10000
20000
30000
0
50000
100000
150000
Mreligion8

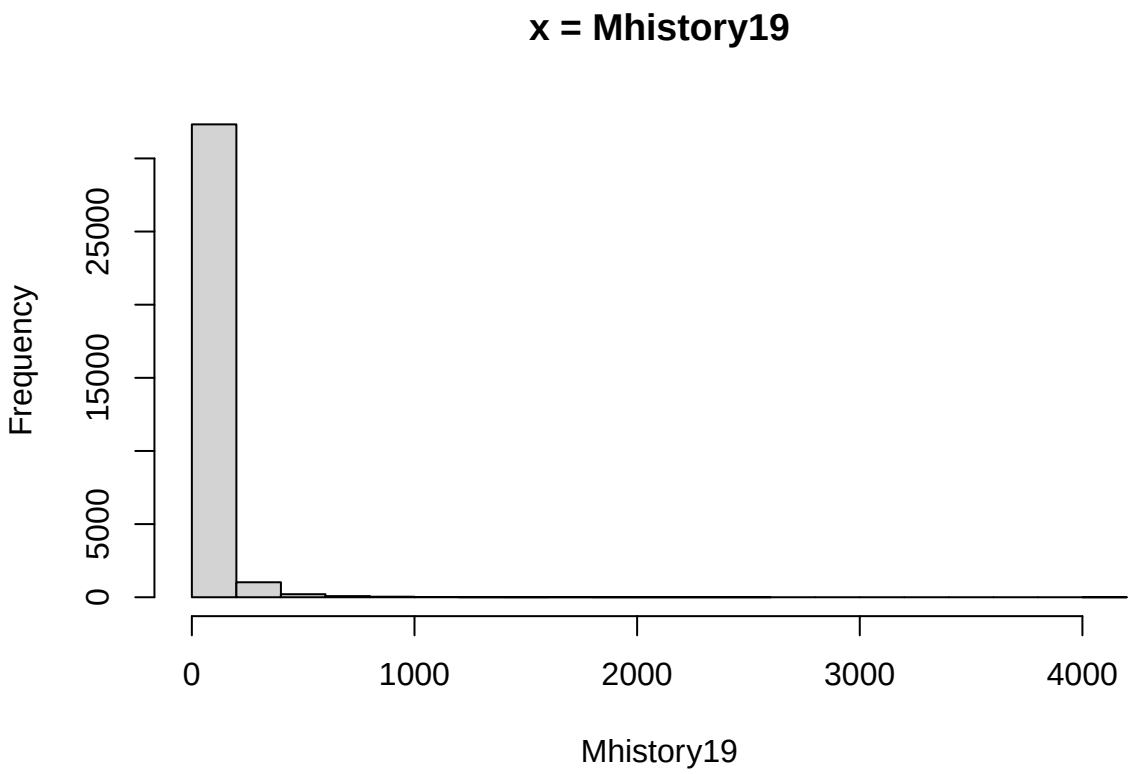

x = Mpsychology9
Frequency
0
10000
20000
30000
0
50
100
150
200
250
300
Mpsychology9

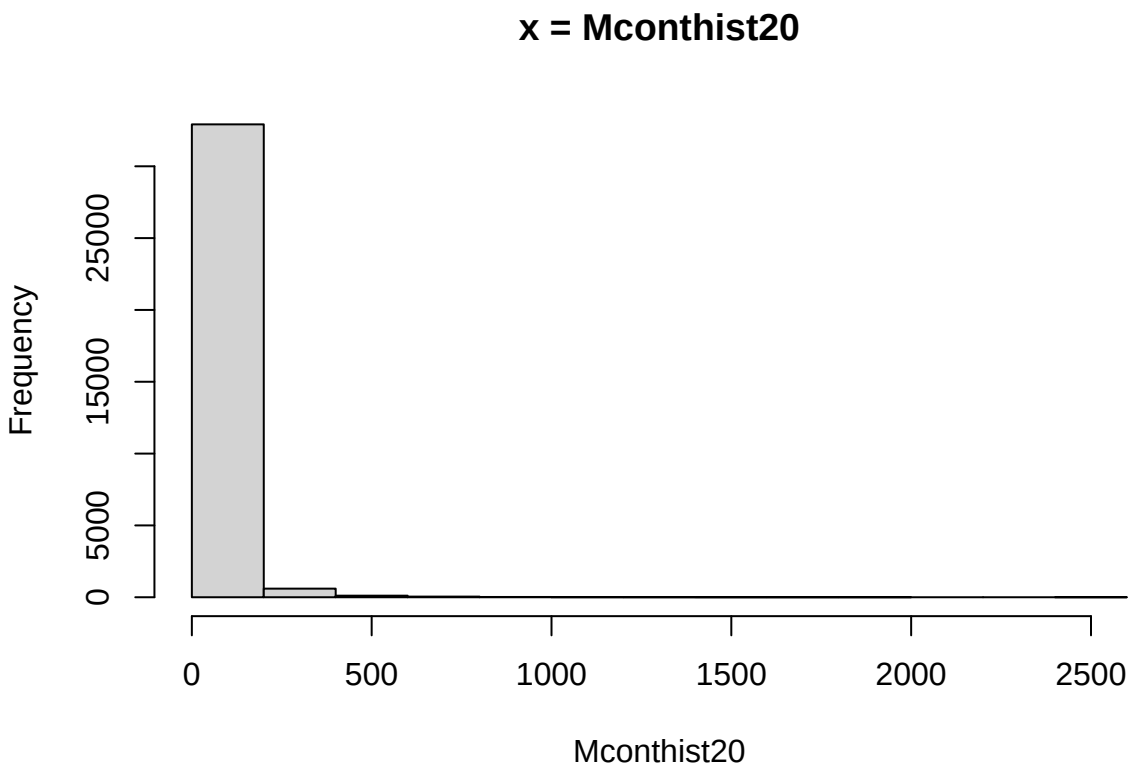

x = Mlinguistics10
Frequency
0
10000
20000
30000
0
200
400
600
800
Mlinguistics10

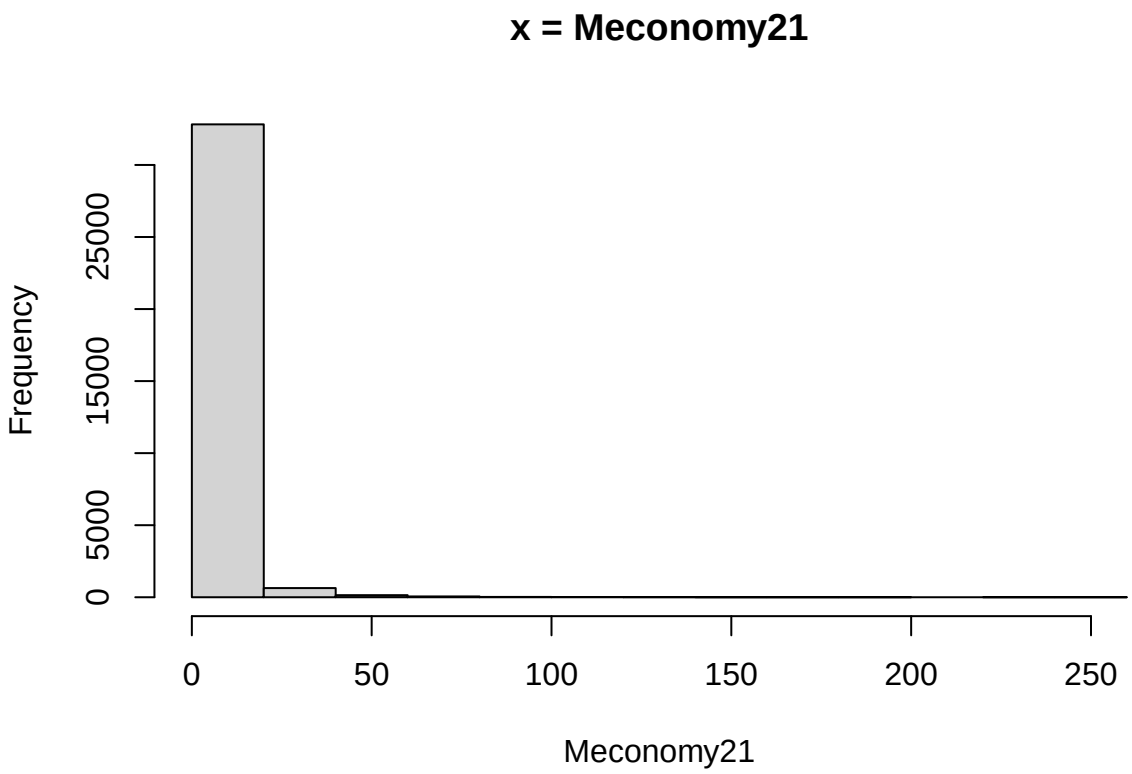

x = Mart12
Frequency
0
10000
20000
30000
0
500
1000
1500
2000
2500
Mart12

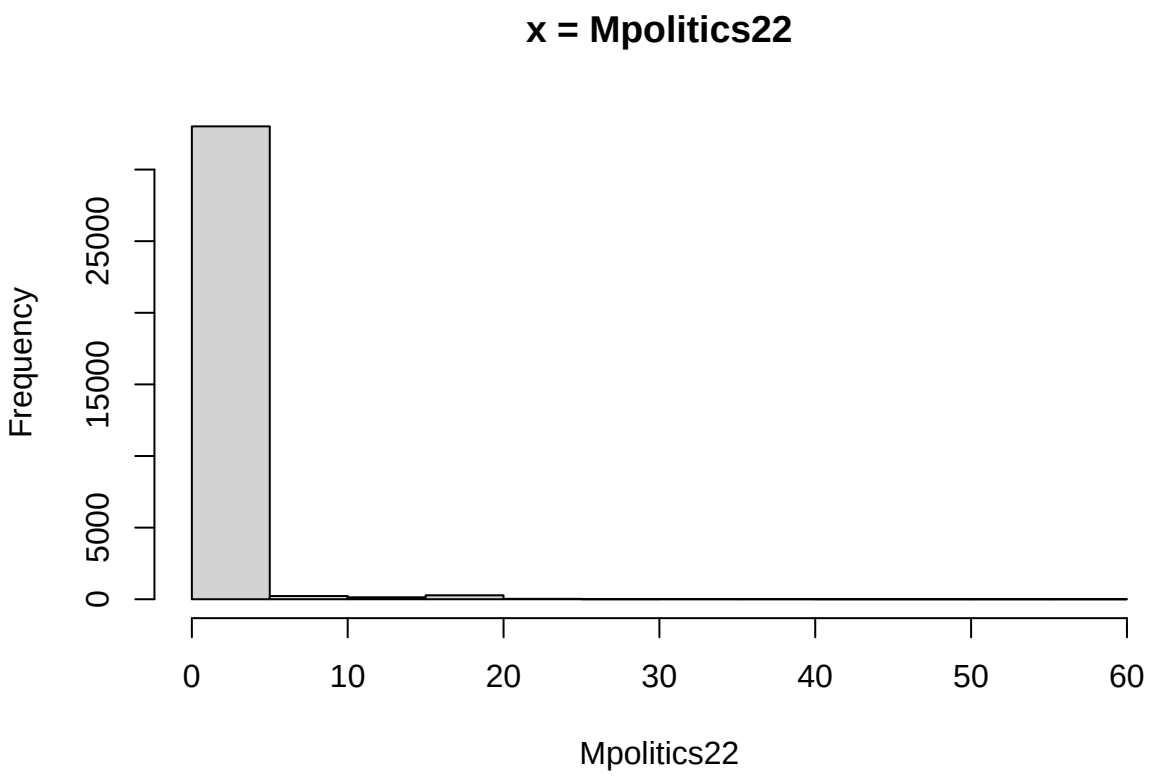

x = Mmusic14
Frequency
0
10000
20000
30000
0
50000
100000
150000
200000
250000
Mmusic14

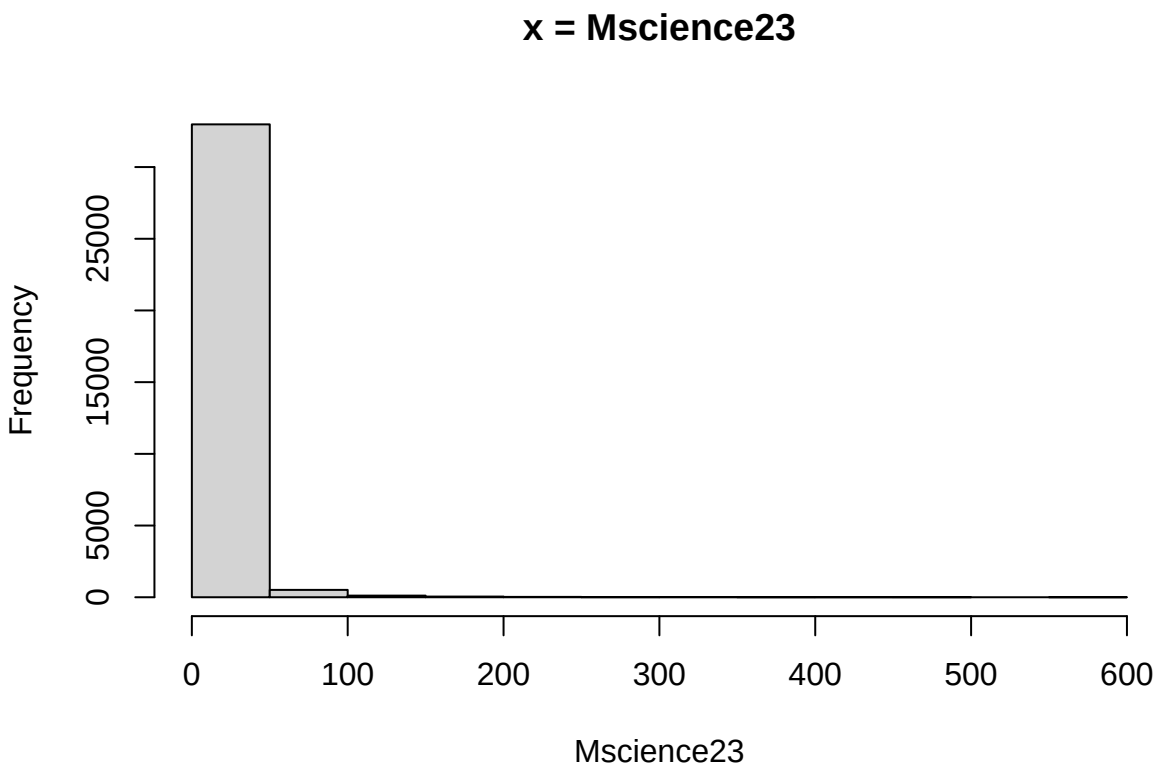

x = Mfacsimile17
Frequency
0
10000
20000
30000
0
2000
4000
6000
8000
Mfacsimile17

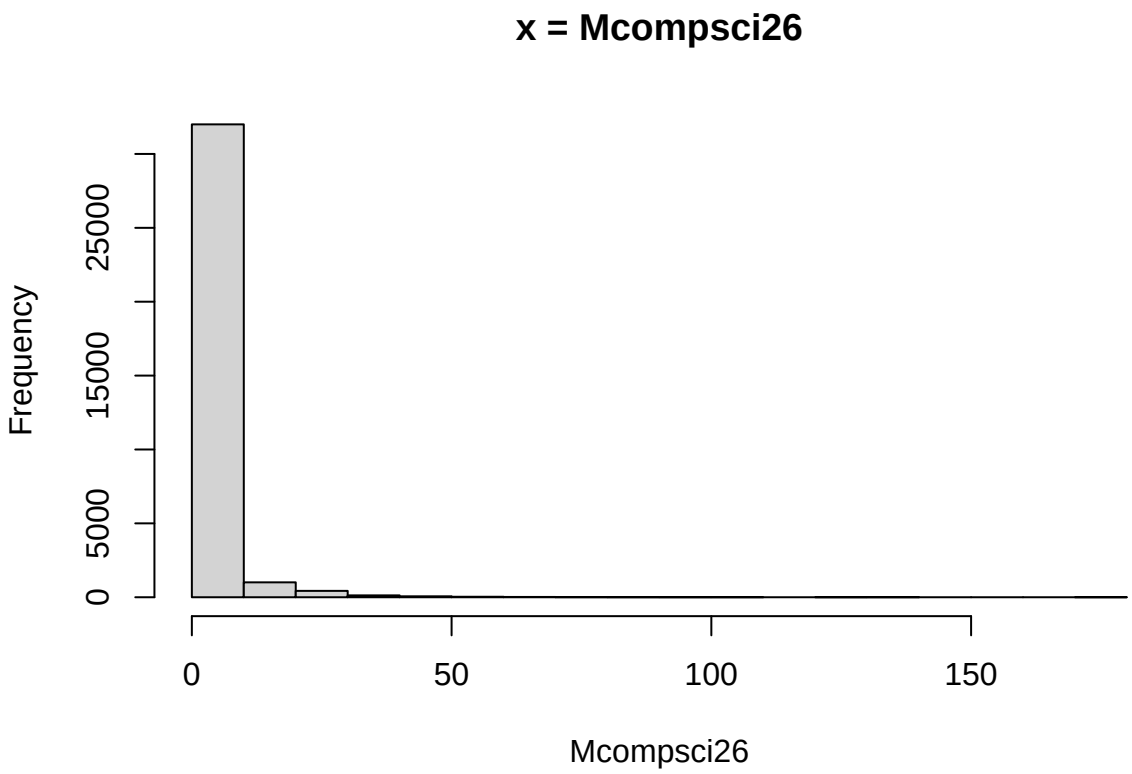

x = Mhistory19
Frequency
0
5000
15000
25000
0
1000
2000
3000
4000
Mhistory19

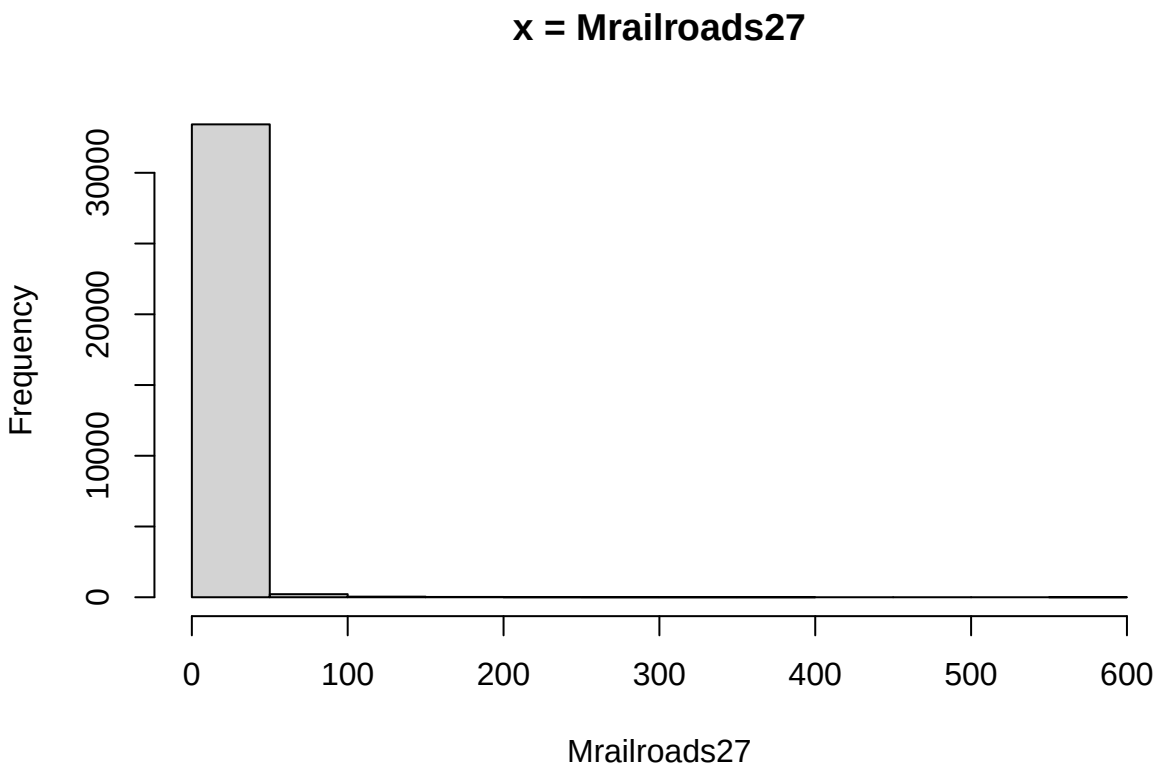

x = Mconthist20
Frequency
0
5000
15000
25000
0
500
1000
1500
2000
2500
Mconthist20

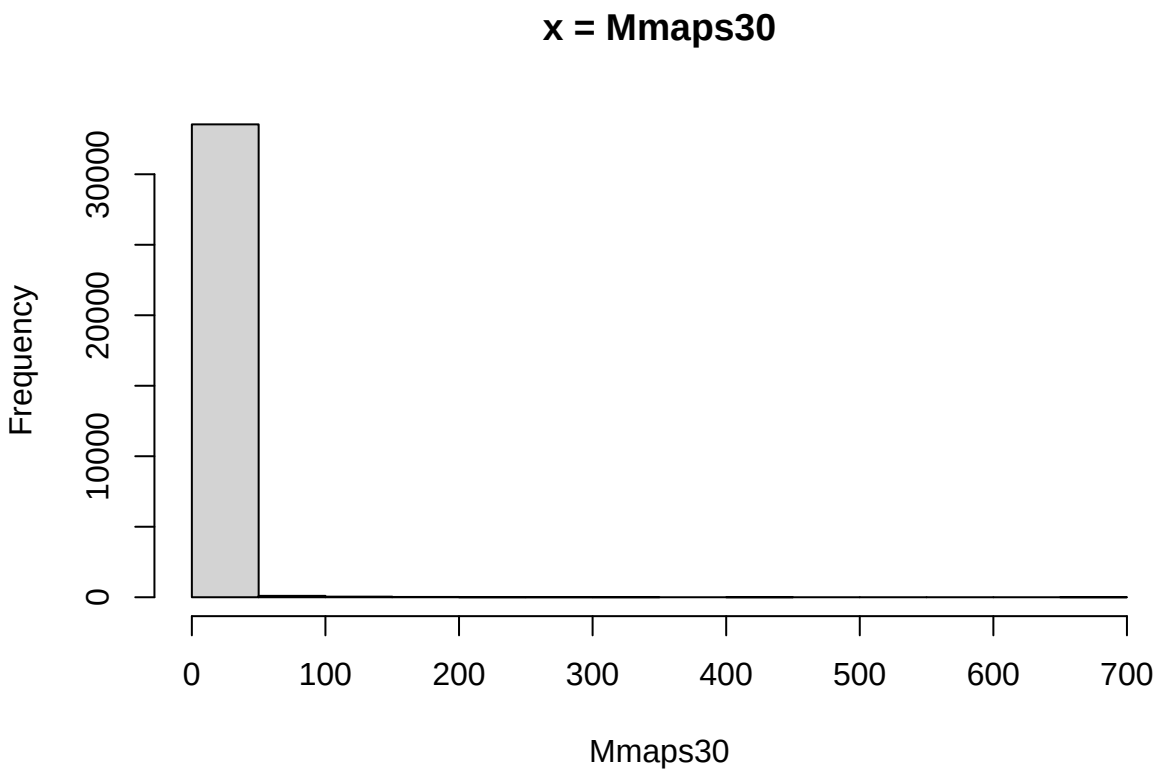

x = Meconomy21
Frequency
0
5000
15000
25000
0
50
100
150
200
250
Meconomy21

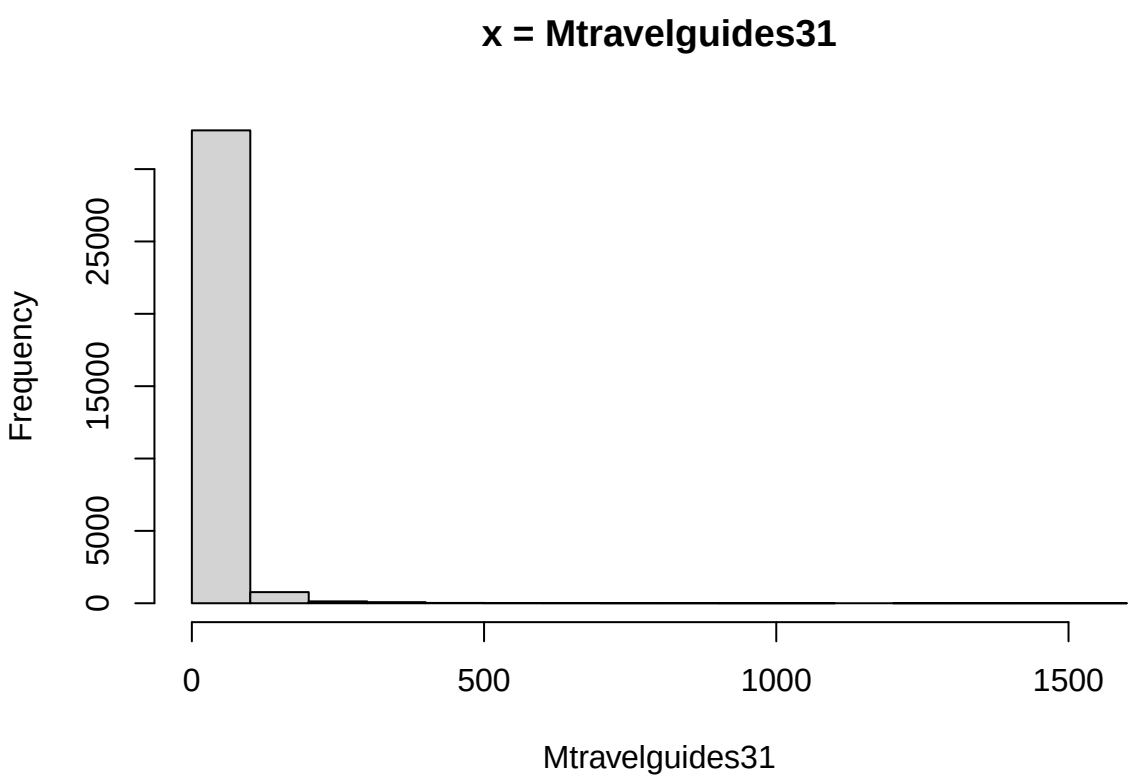

x = Mpolitics22
Frequency
0
5000
15000
25000
0
10
20
30
40
50
60
Mpolitics22

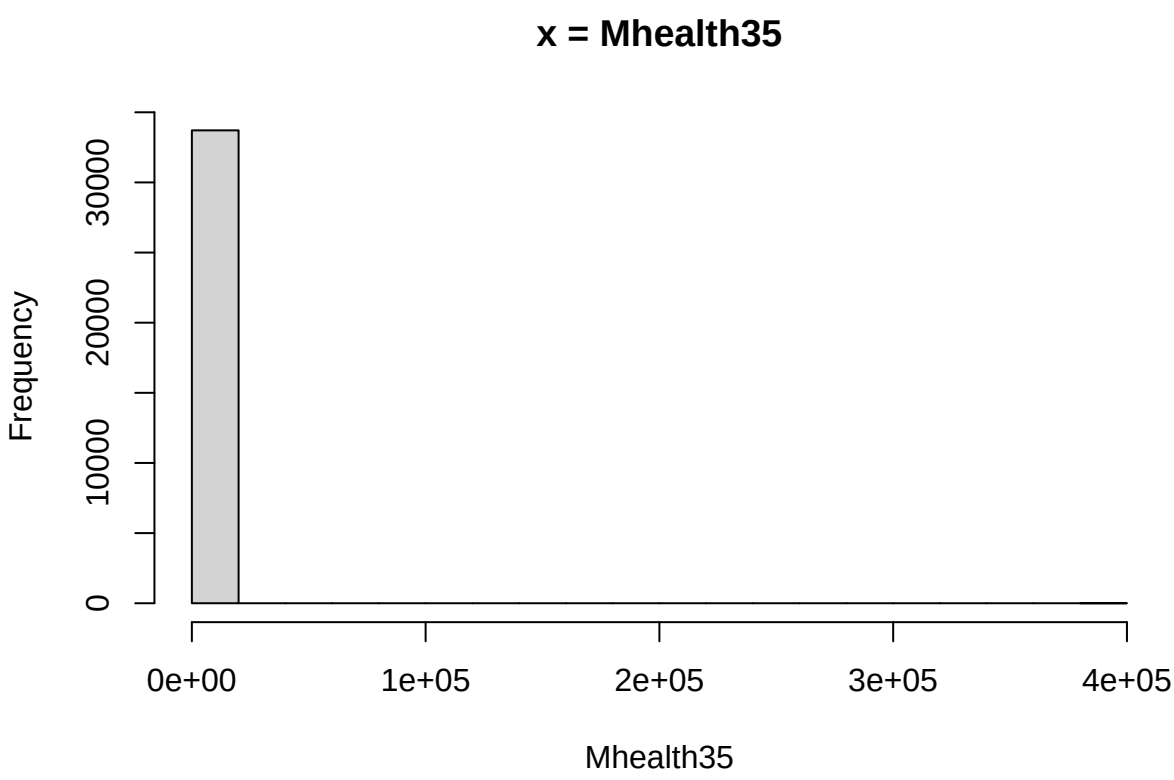

x = Mscience23
Frequency
0
5000
15000
25000
0
100
200
300
400
500
600
Mscience23

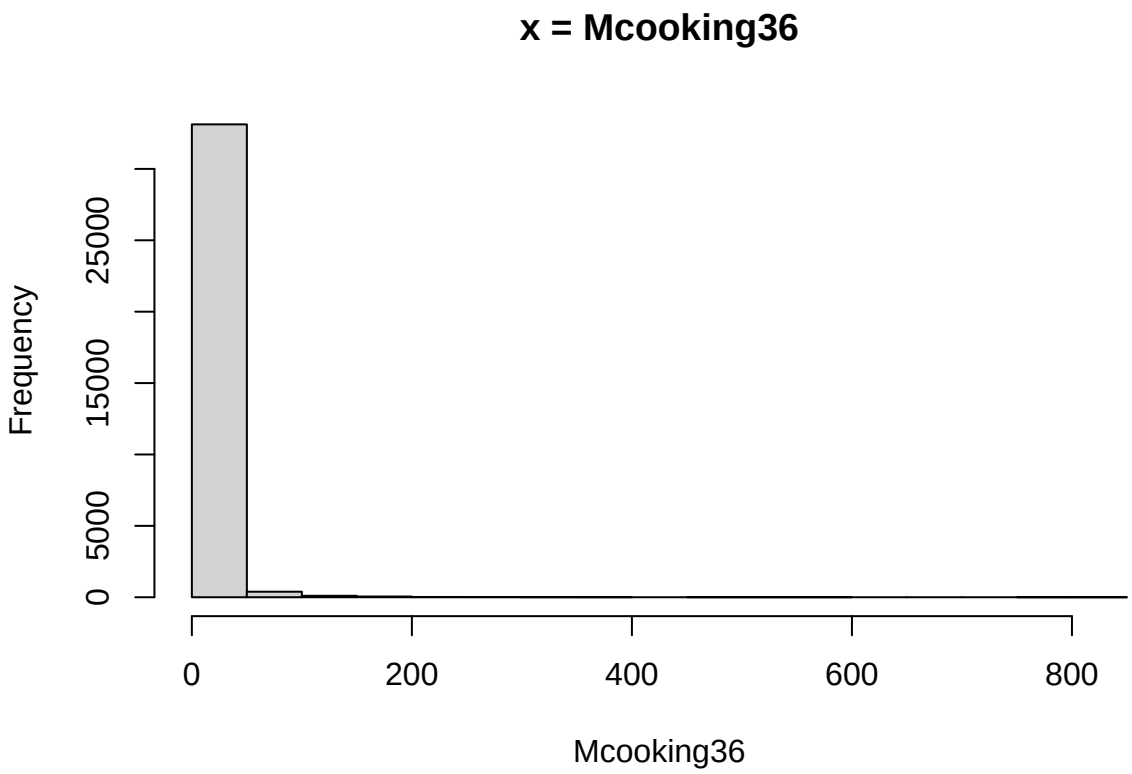

x = Mcompsci26
Frequency
0
5000
15000
25000
0
50
100
150
Mcompsci26

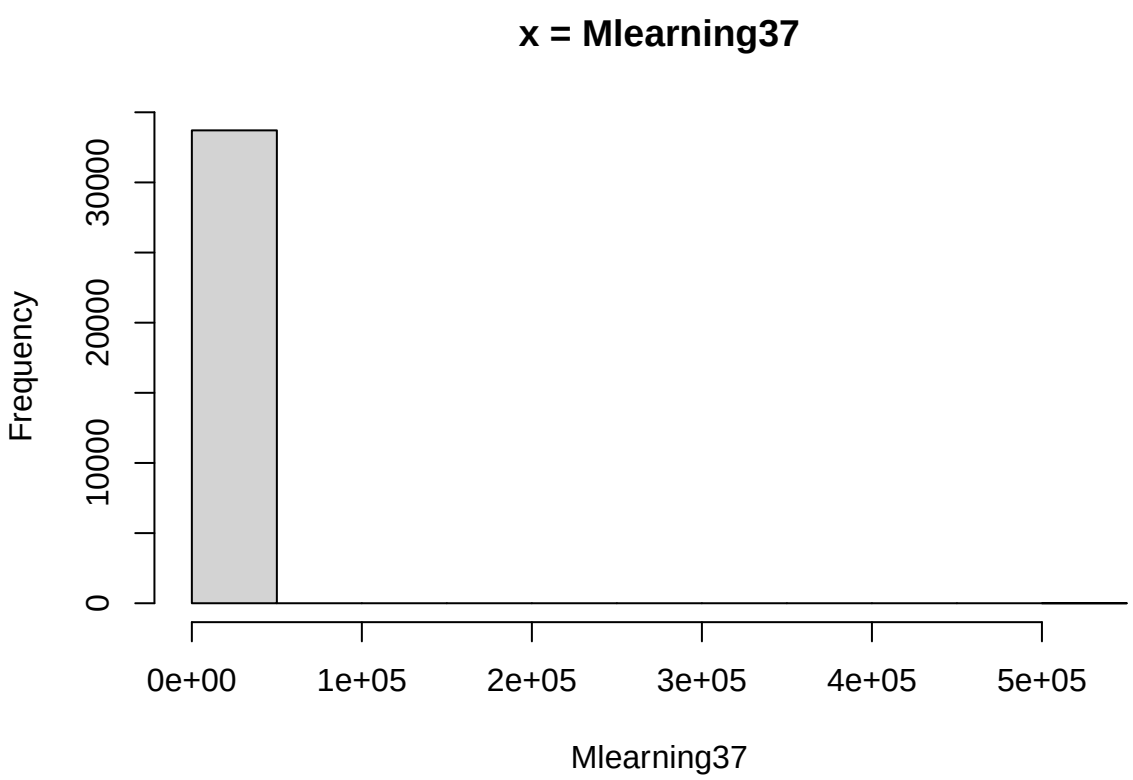

x = Mrailroads27
Frequency
0
10000
20000
30000
0
100
200
300
400
500
600
Mrailroads27

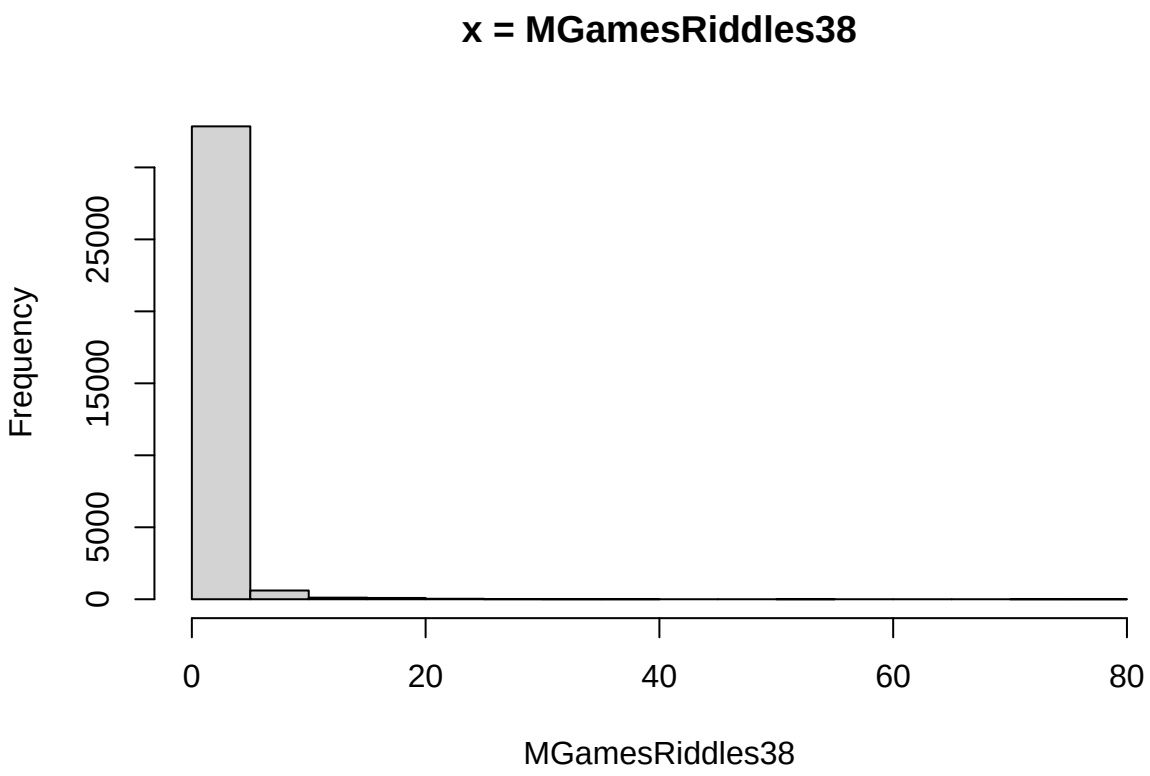

x = Mmaps30
Frequency
0
10000
20000
30000
0
100
200
300
400
500
600
700
Mmaps30

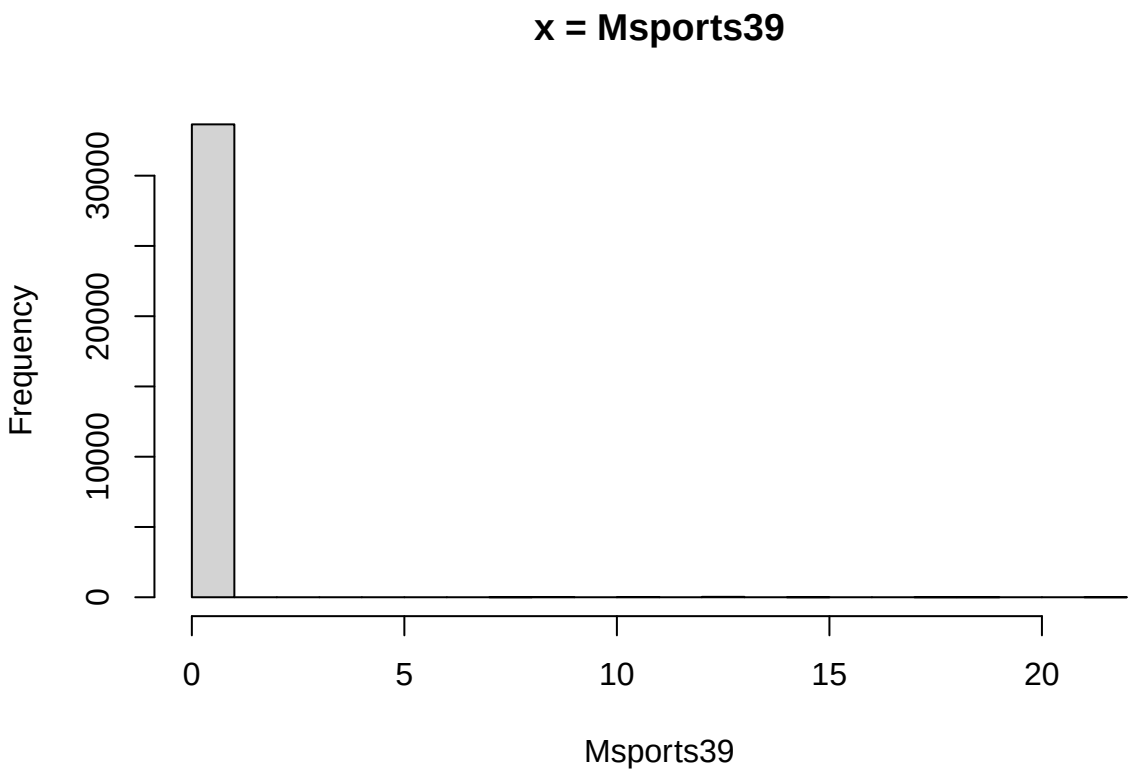

x = Mtravelguides31
Frequency
0
5000
15000
25000
0
500
1000
1500
Mtravelguides31

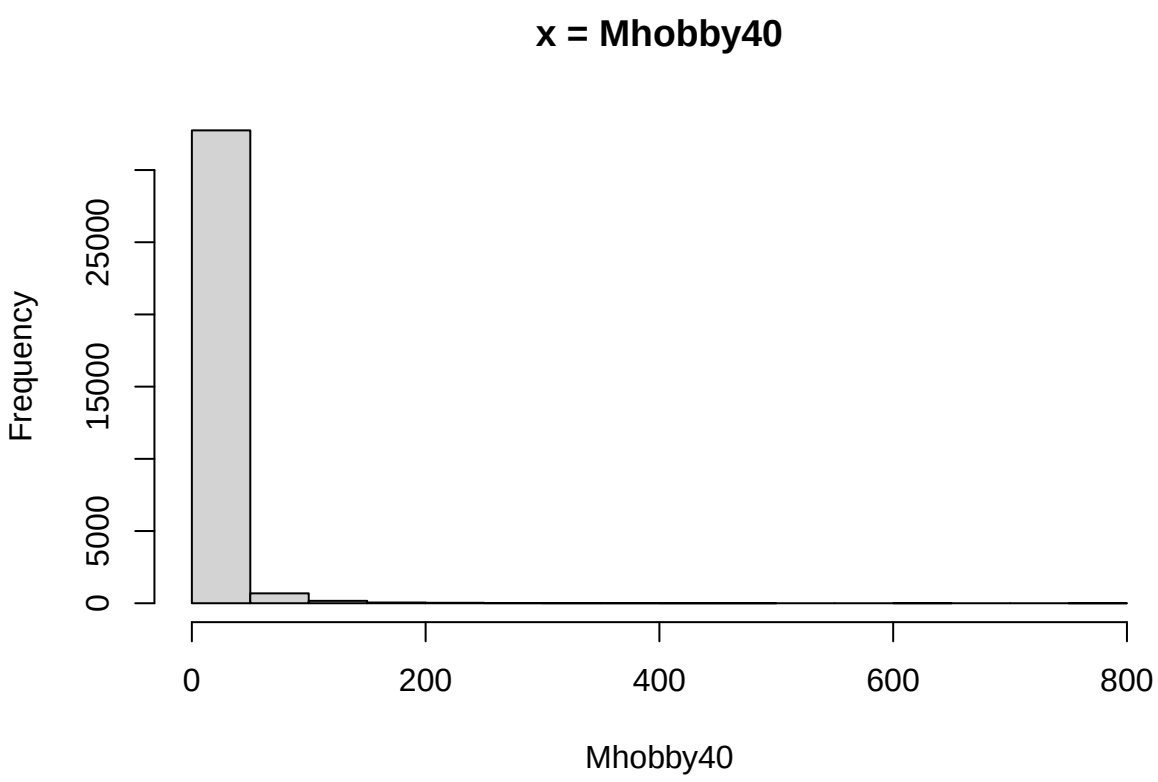

x = Mhealth35
Frequency
0
10000
20000
30000
0e+00
1e+05
2e+05
3e+05
4e+05
Mhealth35

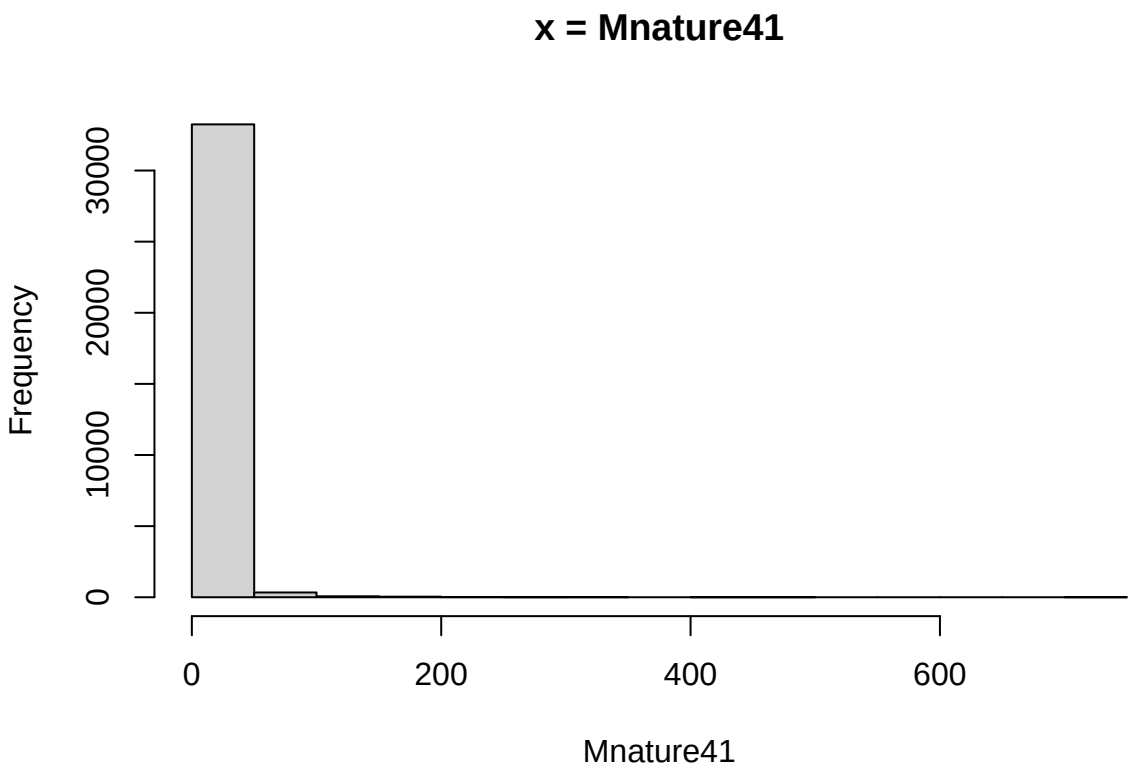

x = Mcooking36
Frequency
0
5000
15000
25000
0
200
400
600
800
Mcooking36

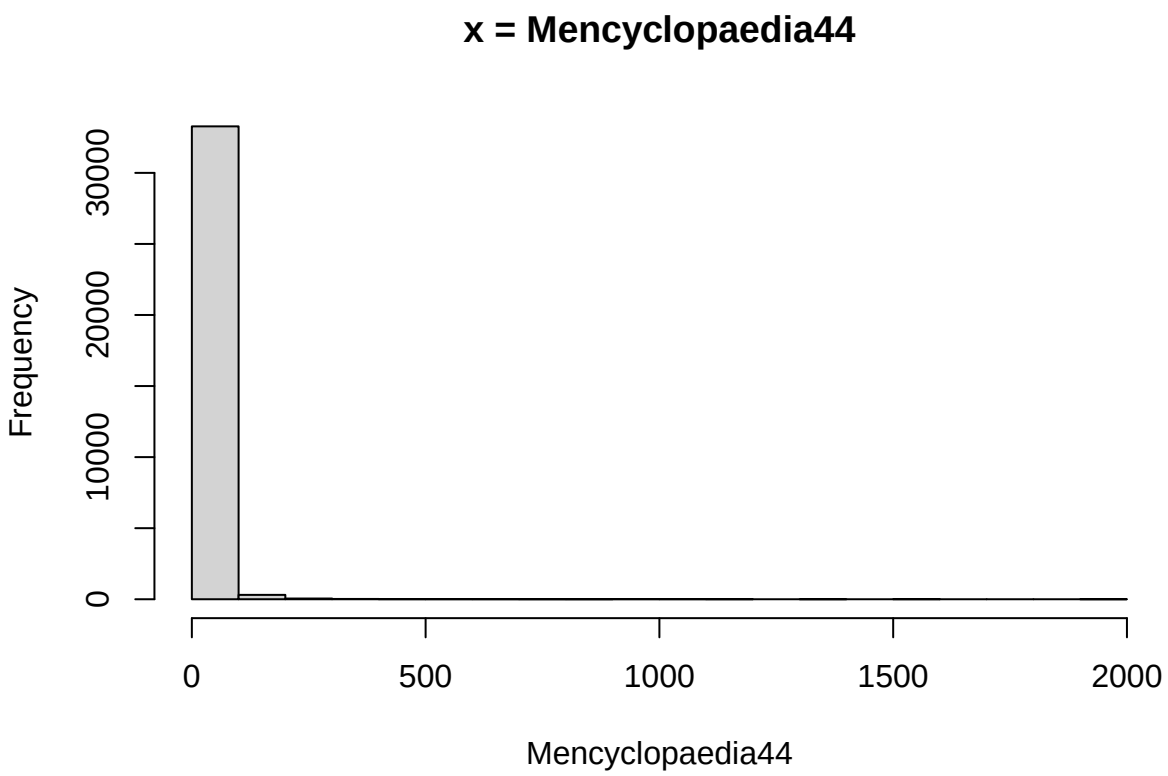

x = Mlearning37
Frequency
0
10000
20000
30000
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Mlearning37

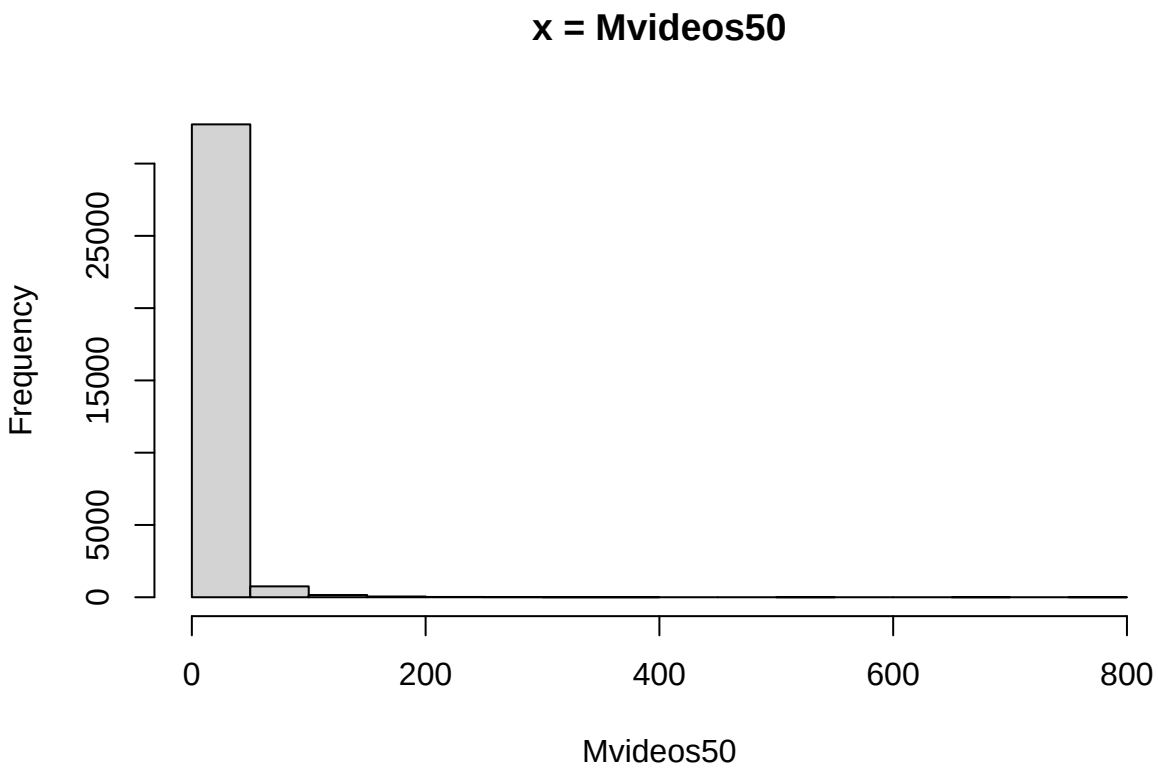

x = MGamesRiddles38
Frequency
0
5000
15000
25000
0
20
40
60
80
MGamesRiddles38

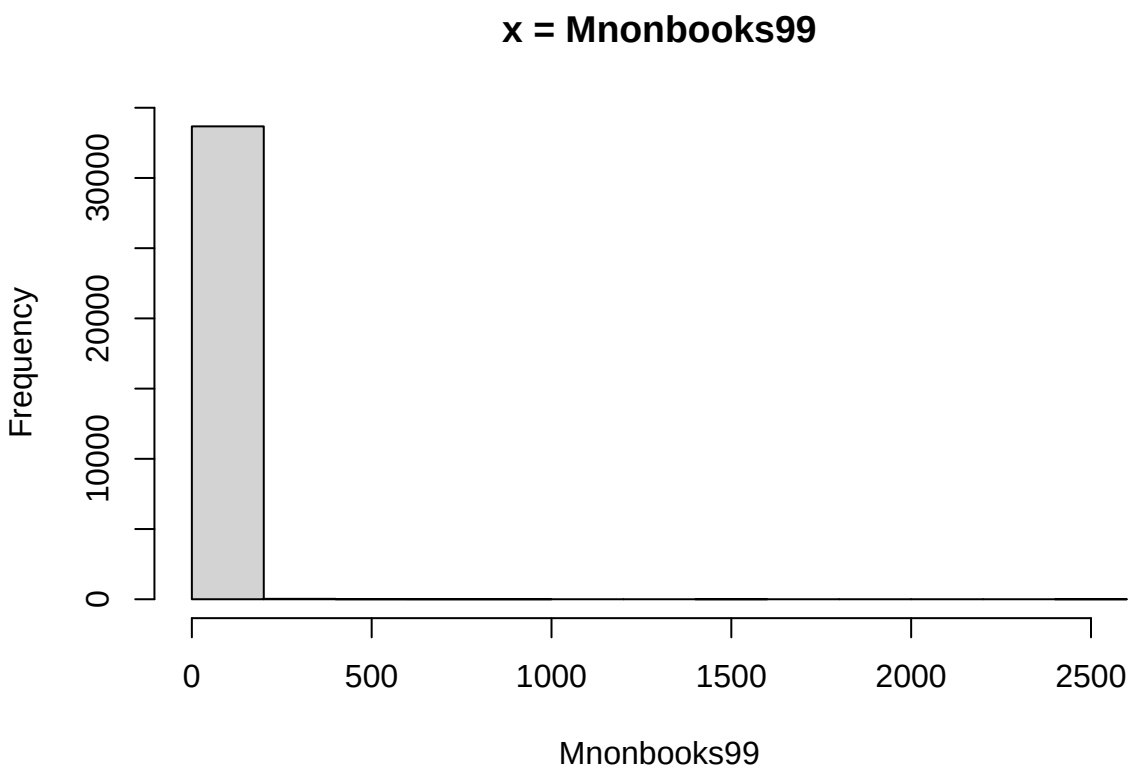

x = Msports39
Frequency
0
10000
20000
30000
0
5
10
15
20
Msports39


x = Mhobby40
Frequency
0
5000
15000
25000
0
200
400
600
800
Mhobby40


x = Mnature41
Frequency
0
10000
20000
30000
0
200
400
600
Mnature41


x = Mencyclopaedia44
Frequency
0
10000
20000
30000
0
500
1000
1500
2000
Mencyclopaedia44


x = Mvideos50
Frequency
0
5000
15000
25000
0
200
400
600
800
Mvideos50



2. Create a new data set with more descriptive labels and the data needed for analysis

```
View(df)
names(df)[names(df) == "r"] = "Recency"
names(df)[names(df) == "f"] = "Frequency"
names(df)[names(df) == "m"] = "Monetary Value"
names(df)[names(df) == "tof"] = "Time Of Flight"
which(is.na(df))
```

```
## integer(0)
```

```
df$average.payment = df$`Monetary Value`/df$Frequency
df[is.na(df)] = 0
summary(df$average.payment)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.00    18.07    27.72    40.13    42.78 84247.71
```

```
#for (i in 1:30){
#  df[,66 + i ] = df[,i+35] / df[, i +5]
#  names(df)[66 + i] = paste(names(df)[i + 35], "/", names(df[i + 5]))
#}
df[is.na(df)] = 0
new.df = df[, c(1:5, 66, 6:65)]
new.df = new.df[-which(new.df[,6] > 1600), ]
#head(new.df)
View(new.df)
```

```
write.csv(new.df, 'cleanned dataset average payment.csv')
```

```
View(new.df)
Fsort = apply(new.df[,7:36], 2, sum)
Msort = apply(new.df[,37:66], 2, sum)
#View(Fsort)
#View(Msort)
head(sort(Fsort, decreasing = TRUE))
```

```
##      Fhistory19        Fmusic14     Fconthist20       Fhealth35 Ftravelguides31
##           89262           78199           72612           71055           50038
##      Freligion8
##           33059
```

```
head(sort(Msort, decreasing = TRUE))
```

```
##      Mhistory19     Mconthist20        Mmusic14       Mhealth35 Mtravelguides31
##       1316880.7        841142.1        733043.3        594062.3        493086.8
##      Mreligion8
##        405650.1
```

```
FM = cbind(sort(Fsort, decreasing = TRUE) , sort(Msort, decreasing = TRUE) )
View(FM)
names(new.df)
```

```
##  [1] "id"               "Recency"          "Frequency"        "Monetary Value"
##  [5] "Time Of Flight"   "average.payment"  "Ffiction1"        "Fclassics3"
##  [9] "Fcartoons5"       "Flegends6"        "Fphilosophy7"     "Freligion8"
## [13] "Fpsychology9"     "Flinguistics10"   "Fart12"           "Fmusic14"
## [17] "Ffacsimile17"     "Fhistory19"       "Fconthist20"      "Feconomy21"
## [21] "Fpolitics22"      "Fscience23"       "Fcompsci26"       "Frailroads27"
## [25] "Fmaps30"          "Ftravelguides31"  "Fhealth35"        "Fcooking36"
## [29] "Flearning37"      "FGamesRiddles38"  "Fsports39"        "Fhobby40"
## [33] "Fnature41"        "Fencyclopaedia44" "Fvideos50"        "Fnonbooks99"
## [37] "Mfiction1"        "Mclassics3"       "Mcartoons5"       "Mlegends6"
## [41] "Mphilosophy7"     "Mreligion8"       "Mpsychology9"     "Mlinguistics10"
## [45] "Mart12"           "Mmusic14"         "Mfacsimile17"     "Mhistory19"
## [49] "Mconthist20"      "Meconomy21"       "Mpolitics22"      "Mscience23"
## [53] "Mcompsci26"       "Mrailroads27"     "Mmaps30"          "Mtravelguides31"
## [57] "Mhealth35"        "Mcooking36"       "Mlearning37"      "MGamesRiddles38"
## [61] "Msports39"        "Mhobby40"         "Mnature41"        "Mencyclopaedia44"
## [65] "Mvideos50"        "Mnonbooks99"
```

```
df = new.df
View(df)

# r median
df.test = df[(df$Recency <= 1632.7596) & (df$Recency >= 693.5915), 2]
median(df.test$Recency)
```

```
## [1] 1069
```

```
df.test = df[(df$Recency <= 693.5915) & (df$Recency >= 280.1592), 2]
median(df.test$Recency)
```

```
## [1] 432
```

```
df.test = df[(df$Recency <= 280.1592) & (df$Recency >= 192.3353), 2]
median(df.test$Recency)
```

```
## [1] 233
```

```
df.test = df[(df$Recency <= 192.3353) & (df$Recency >= 161.1333), 2]
median(df.test$Recency)
```

```
## [1] 179
```

```
df.test = df[(df$Recency <= 161.1333) & (df$Recency >= 160.2685), 2]
median(df.test$Recency)
```

```
## [1] 161
```

```
View(df.test)
```

```
                # r median
# 1163.17   1     1069
# 486.8753    2    432
# 236.2472     3   233
# 176.7343    4     179
# 160.7009    5     161
# < 160.7009   6    < 161
df$Recency.score[df$Recency >= 1069] = 1
```

```
## Warning: Unknown or uninitialised column: `Recency.score`.
```

```
df$Recency.score[df$Recency >= 432 & df$Recency < 1069] = 2
df$Recency.score[df$Recency >= 233 & df$Recency < 432] = 3
df$Recency.score[df$Recency >= 179 & df$Recency < 233] = 4
df$Recency.score[df$Recency >= 161 & df$Recency < 179] = 5
df$Recency.score[df$Recency < 161] = 6
# f median
df.test = df[(df$Frequency <= 42.388889) & (df$Frequency >= 23.652603), 3]
median(df.test$Frequency)
```

```
## [1] 28
```

```
df.test = df[(df$Frequency <= 23.652603) & (df$Frequency >= 10.406760), 3]
median(df.test$Frequency)
```

```
## [1] 14
```

```
df.test = df[(df$Frequency <= 10.406760) & (df$Frequency >= 3.208774), 3]
median(df.test$Frequency)
```

```
## [1] 6
```

```
df.test = df[(df$Frequency <= 3.208774) & (df$Frequency >= 2.523883), 3]
median(df.test$Frequency)
```

```
## [1] 3
```

```
df.test = df[(df$Frequency <= 2.523883), 3]
View(df.test)
median(df.test$Frequency)
```

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

```
# Frequency mean/ median :
# 2.430464       1
# 2.866329       3
# 6.807767      6
# 17.02968       14
# 33.02075       28
df$Frequncy.score[df$Frequency >= 33.02075] = 6
```

```
## Warning: Unknown or uninitialised column: `Frequncy.score`.
```

```
df$Frequncy.score[df$Frequency >= 17.02968 & df$Frequency < 33.02075] = 5
df$Frequncy.score[df$Frequency >= 6.807767 & df$Frequency < 17.02968] = 4
df$Frequncy.score[df$Frequency >= 2.866329 & df$Frequency < 6.807767] = 3
df$Frequncy.score[df$Frequency >= 2.430464 & df$Frequency < 2.866329] = 2
df$Frequncy.score[df$Frequency < 2.430464] = 1

# m median
df.test = df[(df$`Monetary Value` <= 2953.79475) & (df$`Monetary Value` >= 938.05044), 4]
median(df.test$`Monetary Value`)
```

```
## [1] 1249.796
```

```
df.test = df[(df$`Monetary Value` <= 938.05044) & (df$`Monetary Value` >= 362.47937), 4]
median(df.test$`Monetary Value`)
```

```
## [1] 517.0684
```

```
df.test = df[(df$`Monetary Value` <= 362.47937) & (df$`Monetary Value` >= 98.03783), 4]
median(df.test$`Monetary Value`)
```

```
## [1] 176.1854
```

```
df.test = df[(df$`Monetary Value` <= 98.03783) & (df$`Monetary Value` >= 96.26073), 4]
median(df.test$`Monetary Value`)
```

```
## [1] 97.24994
```

```
df.test = df[(df$`Monetary Value` <= 96.26073) & (df$`Monetary Value` >= 71.71050), 4]
median(df.test$`Monetary Value`)
```

```
## [1] 82.74472
```

```
             # monetary value median:
# 96.26073    82.74472
# 98.03783    97.24994
# 362.4794    176.1854
# 938.0504    517.0684
# 2953.795    1249.796
df$Monetary.value.score[df$`Monetary Value` >= 1249.796] = 6
```

```
## Warning: Unknown or uninitialised column: `Monetary.value.score`.
```

```
df$Monetary.value.score[df$`Monetary Value` >= 517.0684 & df$`Monetary Value` < 1249.796] = 5
df$Monetary.value.score[df$`Monetary Value` >= 176.1854 & df$`Monetary Value` < 517.0684] = 4
df$Monetary.value.score[df$`Monetary Value` >= 97.24994 & df$`Monetary Value` < 176.1854] = 3
df$Monetary.value.score[df$`Monetary Value` >= 82.74472 & df$`Monetary Value` < 97.24994] = 2
df$Monetary.value.score[df$`Monetary Value` < 82.74472] = 1
names(df)
```

```
##  [1] "id"                    "Recency"               "Frequency"
##  [4] "Monetary Value"        "Time Of Flight"        "average.payment"
##  [7] "Ffiction1"             "Fclassics3"            "Fcartoons5"
## [10] "Flegends6"             "Fphilosophy7"          "Freligion8"
## [13] "Fpsychology9"          "Flinguistics10"        "Fart12"
## [16] "Fmusic14"              "Ffacsimile17"          "Fhistory19"
## [19] "Fconthist20"           "Feconomy21"            "Fpolitics22"
## [22] "Fscience23"            "Fcompsci26"            "Frailroads27"
## [25] "Fmaps30"               "Ftravelguides31"       "Fhealth35"
## [28] "Fcooking36"            "Flearning37"           "FGamesRiddles38"
## [31] "Fsports39"             "Fhobby40"              "Fnature41"
## [34] "Fencyclopaedia44"      "Fvideos50"             "Fnonbooks99"
## [37] "Mfiction1"             "Mclassics3"            "Mcartoons5"
## [40] "Mlegends6"             "Mphilosophy7"          "Mreligion8"
## [43] "Mpsychology9"          "Mlinguistics10"        "Mart12"
## [46] "Mmusic14"              "Mfacsimile17"          "Mhistory19"
## [49] "Mconthist20"           "Meconomy21"            "Mpolitics22"
## [52] "Mscience23"            "Mcompsci26"            "Mrailroads27"
## [55] "Mmaps30"               "Mtravelguides31"       "Mhealth35"
## [58] "Mcooking36"            "Mlearning37"           "MGamesRiddles38"
## [61] "Msports39"             "Mhobby40"              "Mnature41"
## [64] "Mencyclopaedia44"      "Mvideos50"             "Mnonbooks99"
## [67] "Recency.score"         "Frequncy.score"        "Monetary.value.score"
```

```
df$RFM.score = 100 * df$Recency.score + 10 * df$Frequncy.score + df$Monetary.value.score

df = df[, c(1, 67:70, 2:66)]
View(df)

write.csv(df, 'cleanned dataset RFM score.csv')
```

```
#install.packages("rfm")
#library(rfm)
new.df = df
```

```
df = new.df

df$Segment[(df$Recency.score %in% c(4,5,6)) & ((df$Frequncy.score) %in% c(1,2,3,4)) & (df$Monetary.valu
```

```
## Warning: Unknown or uninitialised column: `Segment`.
```

```
df$Segment[(df$Recency.score %in% c(4,5,6)) & ((df$Frequncy.score) %in% c(4,5,6)) & (df$Monetary.value.s

df$Segment[(df$Recency.score %in% c(5,6)) & ((df$Frequncy.score) %in% c(5,6)) & (df$Monetary.value.score



# df$Segment[(df$Recency.score %in% c(5,3,4,6)) & ((df$Frequncy.score) %in% c(3,4,5,6)) # & (df$Monetary

df$Segment[(df$Recency.score %in% c(4,5)) & ((df$Frequncy.score) %in% c(1,2,4,3)) & (df$Monetary.value.s

df$Segment[(df$Recency.score %in% c(3,4,5)) & ((df$Frequncy.score) %in% c(3,4,5)) & (df$Monetary.value.s


df$Segment[(df$Recency.score %in% c(2,3)) & ((df$Frequncy.score) %in% c(1,3,2)) & (df$Monetary.value.sc
```

```
df$Segment[(df$Recency.score %in% c(2,1)) & ((df$Frequncy.score) %in% c(2,3,4,5,6)) & (df$Monetary.valu


# df$Segment[(df$Recency.score %in% c(1)) & ((df$Frequncy.score) %in% c(1,3,2,4)) &
# (df$Monetary.value.score %in% c(3,4))] = "Hibernating"


df$Segment[(df$Recency.score == 1) & (df$Frequncy.score %in% c(1,2,3,4))  & (df$Monetary.value.score %i

df$Segment[(df$Recency.score %in% c(1)) & (df$Frequncy.score %in% c(5,6)) & (df$Monetary.value.score %i

df$Segment[is.na(df$Segment)] = "Others"
View(df)
df = df[, c(1, 71, 2:70)]
sum(is.na(df$Segment))
```

```
## [1] 0
```

```
table(df$Segment)
```

```
##
##  About to Sleep         At Risk      Can't lose       Champions            Lost
##            7589            3952               7            1106            5264
## Loyal Customers  Need Attention   New Customers          Others       Promising
##            3809             780            7059            2285            1857
```

```
#library(dplyr)
# df %>%
#   group_by(Segment) %>%
#   summarise(n = n())%>%
#
#   mutate(proportion = n / sum(n)) %>%
#   arrange(desc(n))
# library(rfm)
# rfm_plot_median_recency(df$Segment)
#
# df[]

nrow(df)
```

```
## [1] 33708
```

```
sort(prop.table(table(df$Segment)), TRUE)
```

```
##
##  About to Sleep   New Customers            Lost         At Risk Loyal Customers
##    0.2251394328    0.2094161623    0.1561647087    0.1172421977    0.1129998813
##          Others       Promising       Champions  Need Attention      Can't lose
##    0.0677880622    0.0550907796    0.0328112021    0.0231399074    0.0002076658
```

```
#write.csv(df, 'cleanned dataset segment 4.csv')
View(df)
yue = lm(df$`Monetary Value` ~ df$Recency + df$Frequency)
summary(yue)
```

```
##
```

```
## Call:
## lm(formula = df$`Monetary Value` ~ df$Recency + df$Frequency)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3544.8   -58.3   -12.1    24.9 13990.5
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -24.897294   2.410005 -10.331  < 2e-16 ***
## df$Recency     0.019704   0.002615   7.535 5.01e-14 ***
## df$Frequency  39.123813   0.202454 193.248  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252.7 on 33705 degrees of freedom
## Multiple R-squared:  0.5362, Adjusted R-squared:  0.5362
## F-statistic: 1.949e+04 on 2 and 33705 DF,  p-value: < 2.2e-16
```