Zadanie 2.2

Magdalena Opielińska

14 02 2020

```
#1
chooseCRANmirror(graphics=FALSE, ind=1)
knitr::opts_chunk$set(echo = TRUE)
dane <- read.csv("http://mlr.cs.umass.edu/ml/machine-learning-databases/cpu-performance/machine.data")
head(dane)
```

```
##   adviser   X32.60 X125  X256 X6000 X256.1 X16 X128 X198 X199
## 1  amdahl   470v/7   29  8000 32000     32   8   32  269  253
## 2  amdahl  470v/7a   29  8000 32000     32   8   32  220  253
## 3  amdahl  470v/7b   29  8000 32000     32   8   32  172  253
## 4  amdahl  470v/7c   29  8000 16000     32   8   16  132  132
## 5  amdahl   470v/b   26  8000 32000     64   8   32  318  290
## 6  amdahl 580-5840   23 16000 32000     64  16   32  367  381
```

```
colnames(dane) <- c("vendor name","Model Name","MYCT","MMIN","MMAX","CACH","CHMIN","CHMAX","PRP","ERP")
```

```
#2

anyNA(dane)
```

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

```
no.question.mark <- apply(dane, 1, function(r) !any(r %in% ' ?'))
no.question.mark
```

```
##   [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [76] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [91] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [106] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [121] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [136] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [151] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [166] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [181] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [196] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
#3

install.packages("ggplot2")
```

```
## Installing package into 'D:/OneDrive/Dokumenty/R/win-library/3.6'
```

```
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\forqa\AppData\Local\Temp\RtmpY1JrKH\downloaded_packages
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
library(dplyr)
```

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

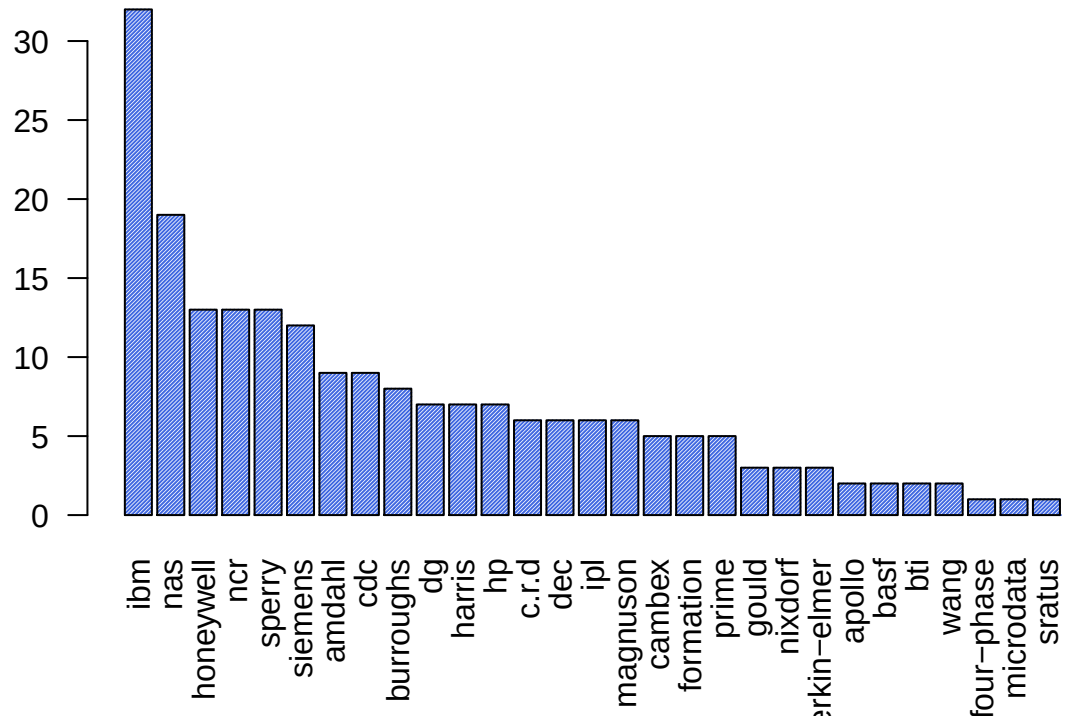
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
vendors <- as.data.frame(table(dane$`vendor name`))
colnames(vendors) <-c("vendor name", "freq")
head(vendors)
```

```
##   vendor name freq
## 1      amdahl    9
## 2      apollo    2
## 3        basf    2
## 4         bti    2
## 5   burroughs    8
## 6       c.r.d    6
```

```
vendors <- vendors[order(vendors$freq, decreasing = TRUE),]
barplot(vendors$freq, names.arg = vendors$`vendor name`, las = 2, col = 'royalblue', density = 69, main
```




```
#4


# MYCT histogram

factor(dane$MYCT)
```

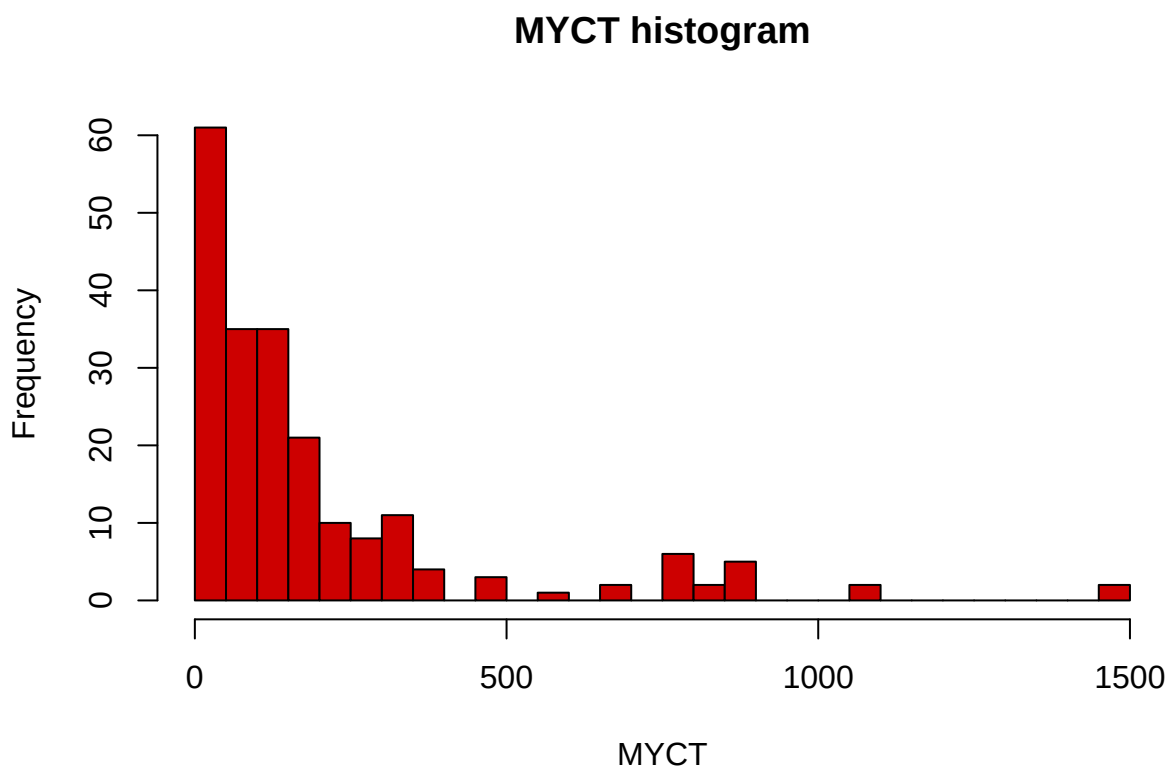
```
##   [1] 29   29   29   29   26   23   23   23   23   400  400  60   50   350  200
##  [16] 167  143  143  110  143  143  143  110  320  320  320  320  320  320  25
##  [31] 25   50   50   56   64   50   50   50   50   50   50   50   50   133  133
##  [46] 810  810  320  200  700  700  140  200  110  110  220  800  800  800  800
##  [61] 800  125  75   75   75   90   105  105  105  75   75   175  300  300  300
##  [76] 300  300  300  180  330  300  300  330  330  140  140  140  140  140  140
##  [91] 140  140  57   57   26   26   26   26   480  203  115  1100 1100 600  400
## [106] 400  900  900  900  900  900  225  225  180  185  180  225  25   25   17
## [121] 17   1500 1500 800  50   50   50   50   50   50   100  100  100  50   50
## [136] 50   150  115  115  92   92   92   75   60   60   60   50   72   72   40
## [151] 40   35   38   48   38   30   112  84   56   56   56   56   56   56   38
## [166] 38   38   38   38   200  200  200  250  250  250  160  160  160  160  160
## [181] 240  240  105  105  105  52   70   59   59   26   26   26   116  50   50
## [196] 50   50   30   30   180  180  180  180  124  98   125  480  480
## 60 Levels: 17 23 25 26 29 30 35 38 40 48 50 52 56 57 59 60 64 70 72 75 ... 1500
```

```
table(dane$MYCT)
```

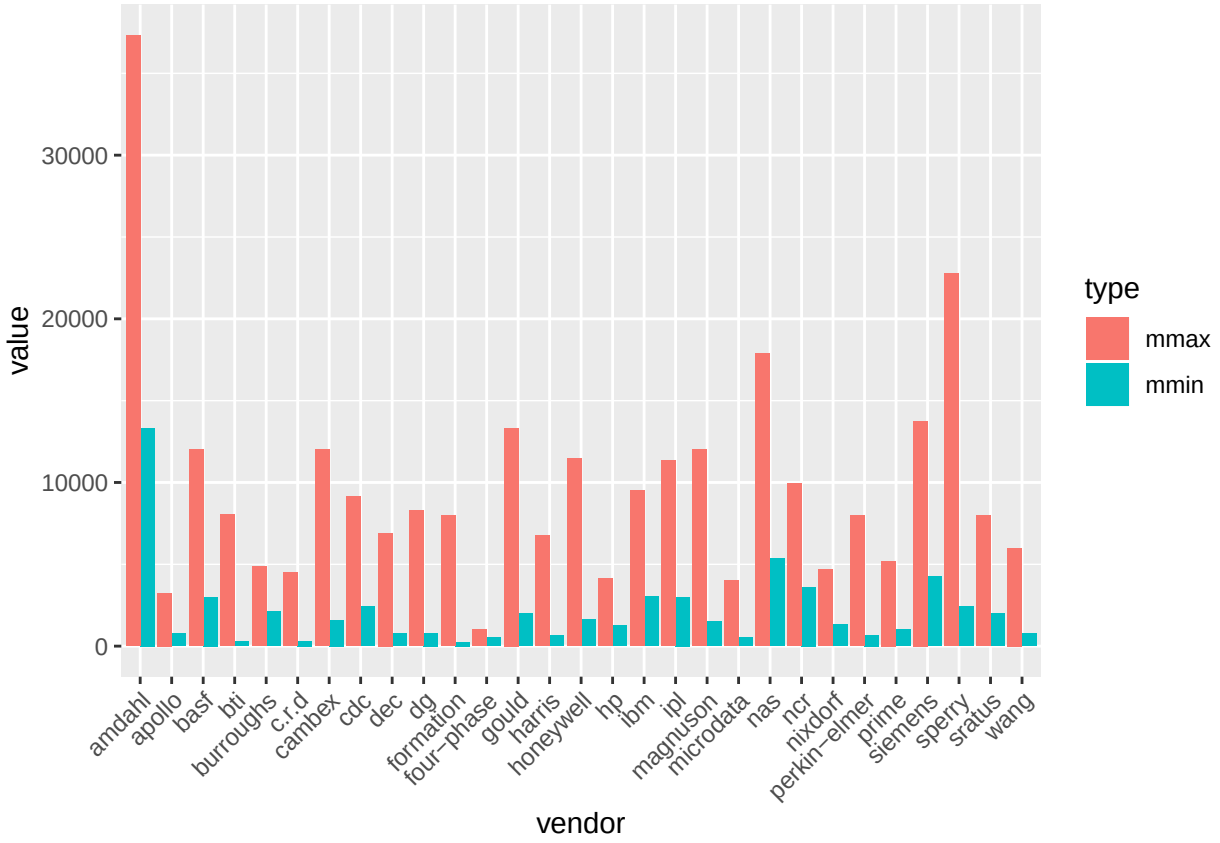
```
##
##   17   23   25   26   29   30   35   38   40   48   50   52   56   57   59   60
```

```
##     2     4     4     8     4     3     1     7     2     1    25     1     7     2     2     4
##    64    70    72    75    84    90    92    98   100   105   110   112   115   116   124   125
##     1     1     2     6     1     1     3     1     3     6     4     1     3     1     1     2
##   133   140   143   150   160   167   175   180   185   200   203   220   225   240   250   300
##     2     9     5     1     5     1     1     7     1     6     1     1     3     2     3     8
##   320   330   350   400   480   600   700   800   810   900  1100  1500
##     7     3     1     4     3     1     2     6     2     5     2     2
```

```
MYCT <- dane$MYCT

hist(MYCT, breaks = 50, col = 'red3', main = 'MYCT histogram')
```




```
# histogram prezentujący wielkość MMIN i MMAX na jednym wykresie

MMIN <- aggregate(dane$MMIN, list(dane$`vendor name`), mean)
MMAX <- aggregate(dane$MMAX, list(dane$`vendor name`), mean)
colnames(MMIN) <- c('vendor', 'value')
colnames(MMAX) <- c('vendor', 'value')
MMIN$type <- 'mmin'
MMAX$type <- 'mmax'
chart_data <- rbind(MMIN, MMAX)

ggplot(chart_data, aes(fill=type, y=value, x=vendor)) +  geom_bar(position="dodge", stat="identity",) +
```




```
#5

# procentowy udział procesorów z CHMAX większym niż 12 w grupach wyodrębnionych ze względu na producent

chmax_12 <- filter(dane, CHMAX > 12)
count <- aggregate(chmax_12$`vendor name`, list(chmax_12$`vendor name`), FUN = NROW)
procent <- mutate(count,
                  procent = round(count$x/nrow(chmax_12)*100))
colnames(procent) <- c('Producent', 'Ilość', 'Procent %')
procent
```

```
##     Producent Ilość Procent %
## 1      amdahl     9        11
## 2         bti     1         1
## 3   burroughs     8         9
## 4         cdc     6         7
## 5  four-phase     1         1
## 6       gould     3         4
## 7      harris     7         8
## 8   honeywell     8         9
## 9          hp     5         6
## 10        ibm     5         6
## 11   magnuson     3         4
## 12  microdata     1         1
## 13        nas     8         9
## 14        ncr     3         4
```

```
## 15      prime     3         4
## 16    siemens     6         7
## 17     sperry     7         8
## 18     sratus     1         1
```

```
#6

# Firmy produkujące procesory z CHMIN woększym niż 16

chmin_16 <- filter(dane, CHMIN > 16)
nrow(chmin_16)
```

```
## [1] 5
```

```
chmin_16
```

```
##   vendor name Model Name MYCT  MMIN  MMAX CACH CHMIN CHMAX  PRP  ERP
## 1      amdahl   580-5880   23 32000 64000  128    32    64 1144 1238
## 2     siemens    7.881-2   26  8000 32000  128    24    32  405  382
## 3      sperry    1100/82   50  2000 32000   48    26    52  208  227
## 4      sperry    1100/83   50  2000 32000  112    52   104  307  341
## 5      sperry    1100/84   50  4000 32000  112    52   104  397  360
```

```
# ERP dla 4 producentów, którzy produkują najwięcej typów CPU
library(tidyr)

top_4 <- data.frame(sort(table(dane$'vendor name'),decreasing=TRUE)[1:4])
top_4
```

```
##        Var1 Freq
## 1       ibm   32
## 2       nas   19
## 3 honeywell   13
## 4       ncr   13
```

```
producenci <- subset(dane, dane$'vendor name' %in% top_4$Var1)
producenci
```

```
##     vendor name  Model Name MYCT  MMIN  MMAX CACH CHMIN CHMAX PRP ERP
## 80    honeywell    dps:6/35  330  1000  3000    0     2     4  16  23
## 81    honeywell    dps:6/92  300  1000  4000    8     3    64  38  30
## 82    honeywell    dps:6/96  300  1000 16000    8     2   112  38  73
## 83    honeywell    dps:7/35  330  1000  2000    0     1     2  16  20
## 84    honeywell    dps:7/45  330  1000  4000    0     3     6  22  25
## 85    honeywell    dps:7/55  140  2000  4000    0     3     6  29  28
## 86    honeywell    dps:7/65  140  2000  4000    0     4     8  40  29
## 87    honeywell    dps:8/44  140  2000  4000    8     1    20  35  32
## 88    honeywell    dps:8/49  140  2000 32000   32     1    20 134 175
## 89    honeywell    dps:8/50  140  2000  8000   32     1    54  66  57
## 90    honeywell    dps:8/52  140  2000 32000   32     1    54 141 181
## 91    honeywell    dps:8/62  140  2000 32000   32     1    54 189 181
## 92    honeywell    dps:8/20  140  2000  4000    8     1    20  22  32
## 93          ibm      3033:s   57  4000 16000    1     6    12 132  82
## 94          ibm      3033:u   57  4000 24000   64    12    16 237 171
## 95          ibm        3081   26 16000 32000   64    16    24 465 361
## 96          ibm      3081:d   26 16000 32000   64     8    24 465 350
## 97          ibm      3083:b   26  8000 32000    0     8    24 277 220
```

```
## 98          ibm      3083:e   26  8000 16000    0     8    16 185 113
## 99          ibm   370/125-2  480    96   512    0     1     1   6  15
## 100         ibm     370/148  203  1000  2000    0     1     5  24  21
## 101         ibm   370/158-3  115   512  6000   16     1     6  45  35
## 102         ibm        38/3 1100   512  1500    0     1     1   7  18
## 103         ibm        38/4 1100   768  2000    0     1     1  13  20
## 104         ibm        38/5  600   768  2000    0     1     1  16  20
## 105         ibm        38/7  400  2000  4000    0     1     1  32  28
## 106         ibm        38/8  400  4000  8000    0     1     1  32  45
## 107         ibm        4321  900  1000  1000    0     1     2  11  18
## 108         ibm      4331-1  900   512  1000    0     1     2  11  17
## 109         ibm     4331-11  900  1000  4000    4     1     2  18  26
## 110         ibm      4331-2  900  1000  4000    8     1     2  22  28
## 111         ibm        4341  900  2000  4000    0     3     6  37  28
## 112         ibm      4341-1  225  2000  4000    8     3     6  40  31
## 113         ibm     4341-10  225  2000  4000    8     3     6  34  31
## 114         ibm     4341-11  180  2000  8000    8     1     6  50  42
## 115         ibm     4341-12  185  2000 16000   16     1     6  76  76
## 116         ibm      4341-2  180  2000 16000   16     1     6  66  76
## 117         ibm      4341-9  225  1000  4000    2     3     6  24  26
## 118         ibm      4361-4   25  2000 12000    8     1     4  49  59
## 119         ibm      4361-5   25  2000 12000   16     3     5  66  65
## 120         ibm      4381-1   17  4000 16000    8     6    12 100 101
## 121         ibm      4381-2   17  4000 16000   32     6    12 133 116
## 122         ibm      8130-a 1500   768  1000    0     0     0  12  18
## 123         ibm      8130-b 1500   768  2000    0     0     0  18  20
## 124         ibm        8140  800   768  2000    0     0     0  20  20
## 138         nas     as/3000  115  2000  8000   16     1     3  50  46
## 139         nas   as/3000-n  115  2000  4000    2     1     5  40  29
## 140         nas     as/5000   92  2000  8000   32     1     6  62  53
## 141         nas   as/5000-e   92  2000  8000   32     1     6  60  53
## 142         nas   as/5000-n   92  2000  8000    4     1     6  50  41
## 143         nas     as/6130   75  4000 16000   16     1     6  66  86
## 144         nas     as/6150   60  4000 16000   32     1     6  86  95
## 145         nas     as/6620   60  2000 16000   64     5     8  74 107
## 146         nas     as/6630   60  4000 16000   64     5     8  93 117
## 147         nas     as/6650   50  4000 16000   64     5    10 111 119
## 148         nas     as/7000   72  4000 16000   64     8    16 143 120
## 149         nas   as/7000-n   72  2000  8000   16     6     8 105  48
## 150         nas     as/8040   40  8000 16000   32     8    16 214 126
## 151         nas     as/8050   40  8000 32000   64     8    24 277 266
## 152         nas     as/8060   35  8000 32000   64     8    24 370 270
## 153         nas as/9000-dpc   38 16000 32000  128    16    32 510 426
## 154         nas   as/9000-n   48  4000 24000   32     8    24 214 151
## 155         nas     as/9040   38  8000 32000   64     8    24 326 267
## 156         nas     as/9060   30 16000 32000  256    16    24 510 603
## 157         ncr    v8535:ii  112  1000  1000    0     1     4   8  19
## 158         ncr    v8545:ii   84  1000  2000    0     1     6  12  21
## 159         ncr    v8555:ii   56  1000  4000    0     1     6  17  26
## 160         ncr    v8565:ii   56  2000  6000    0     1     8  21  35
## 161         ncr  v8565:ii-e   56  2000  8000    0     1     8  24  41
## 162         ncr    v8575:ii   56  4000  8000    0     1     8  34  47
## 163         ncr    v8585:ii   56  4000 12000    0     1     8  42  62
## 164         ncr    v8595:ii   56  4000 16000    0     1     8  46  78
```

```
## 165         ncr       v8635   38  4000  8000   32    16    32  51  80
## 166         ncr       v8650   38  4000  8000   32    16    32 116  80
## 167         ncr       v8655   38  8000 16000   64     4     8 100 142
## 168         ncr       v8665   38  8000 24000  160     4     8 140 281
## 169         ncr       v8670   38  4000 16000  128    16    32 212 190
```

```
producenci <- subset(dane, dane$'vendor name' %in% top_4$Var1)
total <- NROW(producenci)
producenci <- aggregate(producenci$'vendor name', list(producenci$'vendor name', producenci$ERP), FUN =
colnames(producenci) <- c('vendor', 'ERP', 'Count')
producenci$frequency <- producenci$Count * 100 / total
producenci
```

```
##       vendor ERP Count frequency
## 1        ibm  15     1  1.298701
## 2        ibm  17     1  1.298701
## 3        ibm  18     3  3.896104
## 4        ncr  19     1  1.298701
## 5  honeywell  20     1  1.298701
## 6        ibm  20     4  5.194805
## 7        ibm  21     1  1.298701
## 8        ncr  21     1  1.298701
## 9  honeywell  23     1  1.298701
## 10 honeywell  25     1  1.298701
## 11       ibm  26     2  2.597403
## 12       ncr  26     1  1.298701
## 13 honeywell  28     1  1.298701
## 14       ibm  28     3  3.896104
## 15 honeywell  29     1  1.298701
## 16       nas  29     1  1.298701
## 17 honeywell  30     1  1.298701
## 18       ibm  31     2  2.597403
## 19 honeywell  32     2  2.597403
## 20       ibm  35     1  1.298701
## 21       ncr  35     1  1.298701
## 22       nas  41     1  1.298701
## 23       ncr  41     1  1.298701
## 24       ibm  42     1  1.298701
## 25       ibm  45     1  1.298701
## 26       nas  46     1  1.298701
## 27       ncr  47     1  1.298701
## 28       nas  48     1  1.298701
## 29       nas  53     2  2.597403
## 30 honeywell  57     1  1.298701
## 31       ibm  59     1  1.298701
## 32       ncr  62     1  1.298701
## 33       ibm  65     1  1.298701
## 34 honeywell  73     1  1.298701
## 35       ibm  76     2  2.597403
## 36       ncr  78     1  1.298701
## 37       ncr  80     2  2.597403
## 38       ibm  82     1  1.298701
## 39       nas  86     1  1.298701
## 40       nas  95     1  1.298701
## 41       ibm 101     1  1.298701
```

```
## 42       nas 107     1  1.298701
## 43       ibm 113     1  1.298701
## 44       ibm 116     1  1.298701
## 45       nas 117     1  1.298701
## 46       nas 119     1  1.298701
## 47       nas 120     1  1.298701
## 48       nas 126     1  1.298701
## 49       ncr 142     1  1.298701
## 50       nas 151     1  1.298701
## 51       ibm 171     1  1.298701
## 52 honeywell 175     1  1.298701
## 53 honeywell 181     2  2.597403
## 54       ncr 190     1  1.298701
## 55       ibm 220     1  1.298701
## 56       nas 266     1  1.298701
## 57       nas 267     1  1.298701
## 58       nas 270     1  1.298701
## 59       ncr 281     1  1.298701
## 60       ibm 350     1  1.298701
## 61       ibm 361     1  1.298701
## 62       nas 426     1  1.298701
## 63       nas 603     1  1.298701
```

```
ggplot(data.frame(producenci$vendor), aes(x=producenci$vendor)) + geom_bar() + theme(axis.text.x = eleme
  xlab("Producenci")
```