Example 1 Extended

We first load the RMariaDB. The installation line is commented out, because we assume that you have already installed this package while watching the analogous video.

```
# install.packages("RMariaDB")
library(RMariaDB)
```

Now we make a connection to the elections database on Scholar.

```
connection<-dbConnect(RMariaDB::MariaDB(),
                      host="scholar-db.rcac.purdue.edu",
                      db="elections",
                      user="elections_user",
                      password="Dataelect!98")
```

We query the campaign contributions made by people who work at Purdue University.

```
myDF <- dbGetQuery(connection, "SELECT * FROM elections WHERE employer='PURDUE UNIVERSITY'")
```

Here are the first six of these donations.

```
head(myDF)
```

```
##     cmte_id amndt_ind rpt_tp transaction_pgi   image_num transaction_tp
## 1 C00006486         N     Q2                 80031592213             15
## 2 C00122176         A     MY                 81020041596             15
## 3 C00006486         A     Q1                 84033080815             15
## 4 C00193854         A     YE                 86033984599             15
## 5 C00003558         N     M7                 86034242154             15
## 6 C00122176         A     MY                 87020063442             15
##   entity_tp                   name           city state zip_code
## 1             SULLIVAN, GLENN H DR WEST LAFAYETTE    IN    47906
## 2             SULLIVAN, GLENN H DR    W LAFAYETTE    IN    47906
## 3                     BUTZ, EARL L WEST LAFAYETTE    IN    47906
## 4                  CHRISTIAN, JOHN WEST LAFAYETTE    IN    47906
## 5               BABB, EMERSON M MR        ALACHUA    FL    32615
## 6            DUNKELBERG, WILLIAM C WEST LAFAYETTE    IN    47906
##            employer occupation transaction_dt transaction_amt other_id tran_id
## 1 PURDUE UNIVERSITY                1980-06-19             300
## 2 PURDUE UNIVERSITY                1981-06-24             500
## 3 PURDUE UNIVERSITY                1984-02-22            1100
## 4 PURDUE UNIVERSITY                1985-10-17             500
## 5 PURDUE UNIVERSITY                1986-06-30             500
## 6 PURDUE UNIVERSITY                1987-04-15             500
##   file_num memo_cd memo_text      sub_id
## 1        0                   3.06192e+18
## 2        0                   3.06192e+18
## 3        0                   3.06192e+18
## 4        0                   3.06192e+18
## 5        0                   3.06192e+18
```

```
## 6         0                     3.06192e+18
```

Now we display the number of rows and columns in the result. The number of rows is the number of donations made by people who work at Purdue University. The number of columns is the number of variables that we have in this dataset.

```r
dim(myDF)
```

```
## [1] 5768   21
```

Finally, we extract the cities where people made the donations. We tabulate the results, and then sort this table. Among the donations made by employees of Purdue University, this shows the cities in which the donations were most frequently made.

```r
sort(table(myDF$city))
```

```
## 
##                           ALACHUA        BATESVILLE           BOULDER 
##                 1                 1                 1                 1 
##        BUCK CREEK         CHARLOTTE           CHILTON    ELIZABETHTOWN 
##                 1                 1                 1                 1 
##            EUGENE          EVANSTON      FLOYDS KNOBS      FORT COLLINS 
##                 1                 1                 1                 1 
##   FOUNTAIN VALLEY      FRANCESVILLE        GREENFIELD           HAMMOND 
##                 1                 1                 1                 1 
##         LALAYETTE          LAWRENCE        LOUISVILLE            MEQUON 
##                 1                 1                 1                 1 
##         NANTUCKET         NASHVILLE      NEEDHAM HGTS NORTH PLAINFIELD 
##                 1                 1                 1                 1 
##        PLAINFIELD      SCHERERVILLE     THE WOODLANDS       W. LFAYETTE 
##                 1                 1                 1                 1 
##            WARSAW               ADA         ANN ARBOR            DURHAM 
##                 1                 2                 2                 2 
##   FORT LAUDERDALE         MELBOURNE       MONROEVILLE           MUNSTER 
##                 2                 2                 2                 2 
##         ROCKVILLE        SPRINGDALE          STERLING              STOW 
##                 2                 2                 2                 2 
##         WATERTOWN    WESTLAFAYETTE          ANDERSON         CAMBRIDGE 
##                 2                 2                 3                 3 
##        CINCINNATI    CRAWFORDSVILLE         EARL PARK        FORT MYERS 
##                 3                 3                 3                 3 
##    GOODLETTSVILLE        GROVE CITY            KOKOMO     MISSION VIEJO 
##                 3                 3                 3                 3 
##          OAK PARK        PITTSBURGH          SANTA FE        VALPARAISO 
##                 3                 3                 3                 3 
##     WEST LAFAYETE       BEAVERCREEK              DYER             HAIKU 
##                 3                 4                 4                 4 
##                IN    MOUNT PLEASANT         PARKVILLE        WHITESTOWN 
##                 4                 4                 4                 4 
##       WILLOWBROOK           GLENCOE      GRAND RAPIDS     JAMAICA PLAIN 
##                 4                 5                 5                 5 
##            LOWELL      PHILADELPHIA         THORNTOWN         WESTPOINT 
##                 5                 5                 5                 5 
##        BROWNSBURG    West Lafayette      WILLIAMSPORT     BATTLE GROUND 
##                 6                 6                 6                 7 
##           FISHERS              KATY           LIVONIA          MONROVIA
```

```
##               7                7                7                7
##       SAN DIEGO      TRAIL CREEK       WASHINGTON      OGDEN DUNES
##               7                7                7                8
##    WEST LAFAYETT      BLOOMINGTON         HIGHLAND         LA PORTE
##               8                9                9                9
##      CHESTERTON        OTTERBEIN       SOUTH BEND   WOODLAND HILLS
##              10               10               10               10
##  EAST LONGMEADOW       LONG BEACH        SHEBOYGAN       EL SEGUNDO
##              12               12               12               13
##       WESTFIELD            NYACK      CROWN POINT          GRABILL
##              13               15               16               17
##    WEST LAFYETTE      NOBLESVILLE           CHAPIN          PORTAGE
##              17               18               23               24
##     W. LAFAYETTE       ZIONSVILLE           CARMEL          CHICAGO
##              24               30               36               36
##      VEEDERSBURG      LAKE FOREST       FORT WAYNE           ATTICA
##              55               67               97              133
##      W LAFAYETTE     INDIANAPOLIS        LAFAYETTE   WEST LAFAYETTE
##             159              329             1306             2980
```

Here are some more examples from the elections database:

Most of the Purdue employees (who are making donations) are from the State of Indiana.

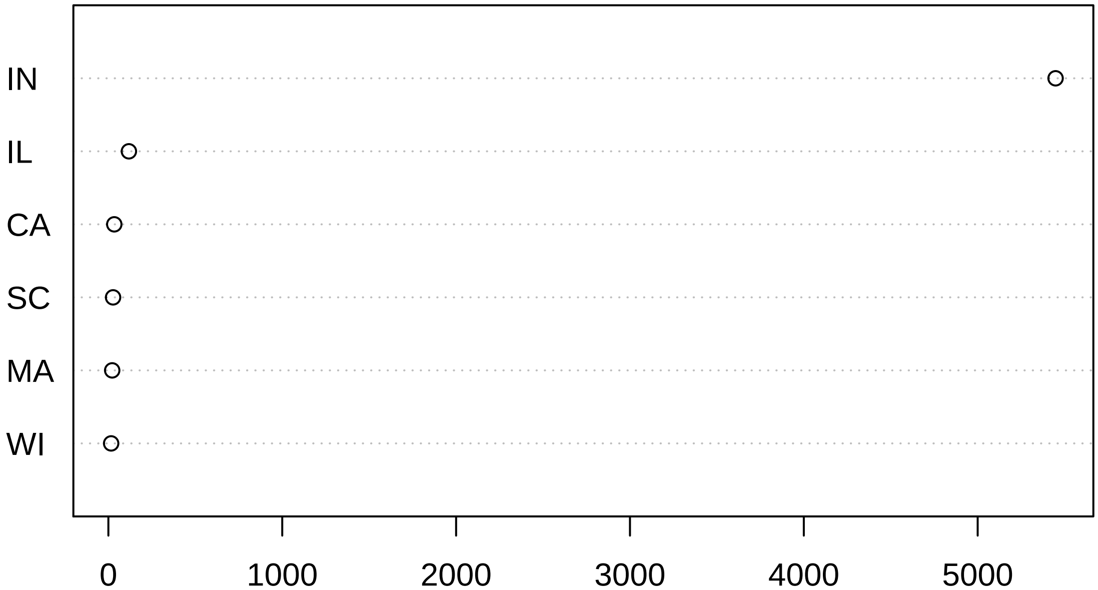
```
sort(table(myDF$state))
```

```
##
##   KS   NJ   OR   AR   CO   KY   NC   NM   HI   MO   TN   DC   FL   TX   PA   OH
##    1    1    1    2    2    2    3    3    4    4    4    7    8    8   10   13
##   MI   NY   WI   MA   SC   CA   IL   IN
##   15   15   16   22   27   34  118 5448
```

Here is a visualization of the number of donations made by Purdue University employees, grouped according to State. We are only showing the 6 most popular States, by this measure.

```
dotchart(tail(sort(table(myDF$state))))
```

```
## Warning in dotchart(tail(sort(table(myDF$state)))): 'x' is neither a vector nor
## a matrix: using as.numeric(x)
```



Note: A warning appears, letting us know that R will treat the data is numeric.

Making one simple change, namely, switching state to name, we can see who has made the largest number of donations. Note: This is not the largest monetary amount of donations! Instead, this is the greatest number of times that donations were made.

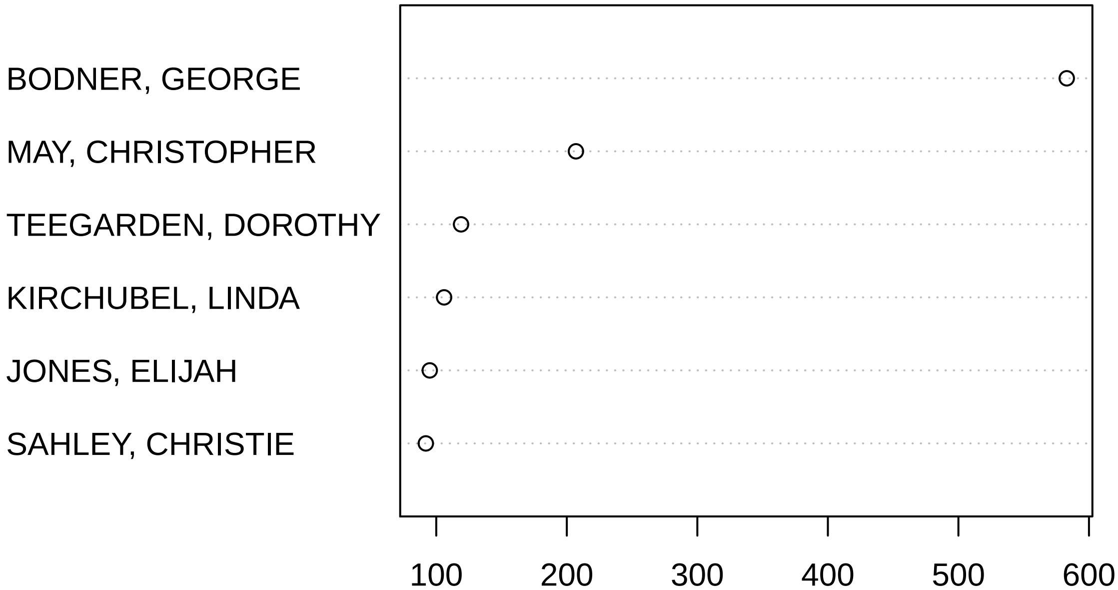
```r
tail(sort(table(myDF$name)))
```

```
##
##   SAHLEY, CHRISTIE      JONES, ELIJAH   KIRCHUBEL, LINDA TEEGARDEN, DOROTHY
##                 92                 95                106                119
##   MAY, CHRISTOPHER     BODNER, GEORGE
##                207                583
```

As before, we can plot this data.

```r
dotchart(tail(sort(table(myDF$name))))
```

```
## Warning in dotchart(tail(sort(table(myDF$name)))): 'x' is neither a vector nor a
## matrix: using as.numeric(x)
```



With this plot in mind, we see why we put the names on the y-axis and the number of donations on the x-axis. (If the names were on the x-axis, we would not have been able to squeeze all of the names in!)

In the questions above, note that we made 1 SQL query (namely, to get the data from people who work at Purdue), and we stored this data in a data frame called myDF. Then we did some analysis based on this data frame.

Now we make another query. This time, we lookup all of the donations across all of the years. We sum the amounts of the donations, grouping the amounts of the donations, according to the state where the donor lives.

```
myDF <- dbGetQuery(connection, "SELECT SUM(transaction_amt), state FROM elections GROUP BY state")
myDF
```

```
##    SUM(transaction_amt) state
## 1             652941123
## 2                   500    'H
## 3                   600    'U
## 4                   250    (A
## 5                   500    -0
## 6                  1614    00
## 7                  1814    01
## 8                     0    03
## 9                     0    06
## 10                    0    09
## 11                 1000     1
## 12                    0    10
## 13                   75    11
## 14                    0    15
## 15                  600    17
## 16                   -5    19
## 17                  750    20
## 18                    0    28
## 19                    0    29
## 20                    0    32
## 21                  210    36
## 22                    0    46
```

```
## 23          0   53
## 24          0   55
## 25       1000   60
## 26          0   64
## 27        500   67
## 28          0   72
## 29         80   75
## 30          0   77
## 31          0   78
## 32        500   79
## 33       2000   7E
## 34        167   81
## 35       5471   85
## 36          0   87
## 37        500   90
## 38          0   91
## 39          0   92
## 40          0   94
## 41          0   95
## 42       -141   98
## 43       1250    <
## 44       2650   ??
## 45       1320    A
## 46     678578   AA
## 47      19553   AB
## 48    1405541   AE
## 49        250   AF
## 50       1000   AI
## 51   69577306   AK
## 52  304041321   AL
## 53        500   AN
## 54        250   AO
## 55     681851   AP
## 56  262243128   AR
## 57     397614   AS
## 58        511   AT
## 59       4612   AU
## 60  532637404   AZ
## 61       1000    B
## 62      14896   BA
## 63      86064   BC
## 64        250   BD
## 65      14107   BE
## 66        151   BK
## 67        250   BL
## 68        505   BO
## 69       4600   BR
## 70       7000   BS
## 71        500   BU
## 72        400   BW
## 73       2000   BX
## 74       4938    C
## 75 5843659325   CA
## 76        500   CD
```

```
## 77                    725    CE
## 78                   6919    CH
## 79                   3250    CI
## 80                   2500    CM
## 81                  17661    CN
## 82              651796668    CO
## 83                   2000    CR
## 84              841002706    CT
## 85                    950    CZ
## 86                   1250     D
## 87                    500    D.
## 88             2614794626    DC
## 89              124615709    DE
## 90                   3280    DF
## 91                   1000    DI
## 92                   1000    DL
## 93                    500    DP
## 94                    500    DU
## 95                   1750    EA
## 96                    210    EE
## 97                    250    EG
## 98                   1000    EL
## 99                    200    EM
## 100                104140    EN
## 101                   426    ES
## 102                   110    EU
## 103                  1318     F
## 104                  4000    FA
## 105                  1600    FC
## 106                294424    FF
## 107                  1250    FI
## 108            2330169028    FL
## 109                135736    FM
## 110                  1000    FO
## 111                   250    FP
## 112                 30948    FR
## 113                  5659    FS
## 114                  1875     G
## 115             747525201    GA
## 116                  1243    GB
## 117                  7171    GE
## 118                  2722    GL
## 119                    40    GP
## 120                  2000    GR
## 121                   900    GT
## 122               4306645    GU
## 123                   300    GY
## 124                   364     H
## 125                  1290    HA
## 126                  6640    HE
## 127             102228962    HI
## 128                  5250    HK
## 129                  1000    HM
## 130                  6555    HO
```

```
## 131              1350   HU
## 132         198590085   IA
## 133              2000   IB
## 134          77527183   ID
## 135        1739757415   IL
## 136              9250   IM
## 137         414889271   IN
## 138              7200   IO
## 139           1048182   IR
## 140              1750   IS
## 141             18250   IT
## 142              2085   JA
## 143              1500   JN
## 144              2425   JP
## 145                39    K
## 146              2150   KA
## 147               300   KE
## 148               250   KI
## 149               500   KO
## 150         252134601   KS
## 151              2064   KU
## 152              1000   KX
## 153         317402936   KY
## 154               100    L
## 155         400073166   LA
## 156               885   LE
## 157                27   LL
## 158               250   LN
## 159             18855   LO
## 160               500   LU
## 161               450   LX
## 162              2058    M
## 163        1236991169   MA
## 164              6822   MB
## 165              3000   MC
## 166         874546900   MD
## 167         119437063   ME
## 168             16965   MH
## 169         795915409   MI
## 170              1900   MJ
## 171         489863090   MN
## 172         534339055   MO
## 173            916011   MP
## 174         148489537   MS
## 175          96255694   MT
## 176              2801   MX
## 177              1500   MY
## 178              5650    N
## 179             12525   N/
## 180            127168   NA
## 181              9486   NB
## 182         597318167   NC
## 183          56477858   ND
## 184         177995925   NE
```

```
## 185           165    NF
## 186     146933211    NH
## 187    1004221945    NJ
## 188          3851    NL
## 189     198249938    NM
## 190         20472    NO
## 191          1050    NR
## 192         13192    NS
## 193         16626    NT
## 194         34306    NU
## 195     677671421    NV
## 196    5327197118    NY
## 197          1676    NZ
## 198         22555     O
## 199     966562414    OH
## 200         24500    OJ
## 201     298419110    OK
## 202        172161    ON
## 203     273321583    OR
## 204          1000    OS
## 205          2750    OT
## 206           250    OV
## 207          1350     P
## 208    1259313542    PA
## 209          1652    PE
## 210          3044    PH
## 211          2200    PN
## 212          1000    PO
## 213      69402832    PR
## 214           500    PU
## 215          6610    PW
## 216           200     Q
## 217         27530    QC
## 218          2250    QU
## 219           250    RH
## 220      94373994    RI
## 221            80    RJ
## 222          6000    RM
## 223          2250    RO
## 224          6009    SA
## 225     245366225    SC
## 226      73622367    SD
## 227          4750    SE
## 228          1250    SH
## 229          3050    SI
## 230           533    SJ
## 231          1665    SK
## 232           250    SO
## 233         15202    SP
## 234           210    SR
## 235          1450    ST
## 236         13619    SU
## 237         25170    SW
## 238            40    SY
```

```
## 239                           500    SZ
## 240                           900     T
## 241                          1500    TA
## 242                          1500    TC
## 243                          3975    TE
## 244                           500    TH
## 245                           500    TI
## 246                          4200    TM
## 247                     537204492    TN
## 248                         10296    TO
## 249                           250    TU
## 250                            50    TW
## 251                    2945587994    TX
## 252                          3000    TZ
## 253                           195    U*
## 254                          1000    UH
## 255                         83523    UK
## 256                         45865    UN
## 257                          1000    UR
## 258                          2000    US
## 259                     170795379    UT
## 260                          2017     V
## 261                    1375724476    VA
## 262                           225    VD
## 263                          1500    VE
## 264                      10251026    VI
## 265                            50    VR
## 266                      66673746    VT
## 267                          1400     W
## 268                     719698705    WA
## 269                          1000    WD
## 270                          2580    WE
## 271                     389146459    WI
## 272                           500    WO
## 273                            10    WP
## 274                          2000    WS
## 275                           500    WU
## 276                     116375772    WV
## 277                     126752246    WY
## 278                          5750    XM
## 279                        203843    XX
## 280                          3263    YT
## 281                           165    ZA
## 282                          1818    ZH
## 283                           500    ZR
## 284                          1000    ZU
## 285                      25192260    ZZ
```

In our first SQL query, we retrieved all of the information about all of the variables that met our criteria. The fact that we wanted all variables in the first query is signified by the “ ″*. In general, when we see a ″*” in data science, it means that we want all such items or results.

In this second SQL query, however, we only extract two variables, namely, the sum of the transaction amouns, and the states. We also group the results from the SQL query according to the state where the donor lives.

Notice that the results have 285 rows and 2 columns.

```r
head(myDF)
```

```
##   SUM(transaction_amt) state
## 1            652941123
## 2                  500    'H
## 3                  600    'U
## 4                  250    (A
## 5                  500    -0
## 6                 1614    00
```

```r
dim(myDF)
```

```
## [1] 285   2
```

The first column is the sum of the transactions from the state, and the second column is the state. There are many erroneous states.

If we save the sum of transaction amounts as v and give each element of v the name of that state

```r
v <- myDF$`SUM(transaction_amt)`
names(v) <- myDF$state
```

then we are ready to sort the data, as we did before.

```r
sort(v)
```

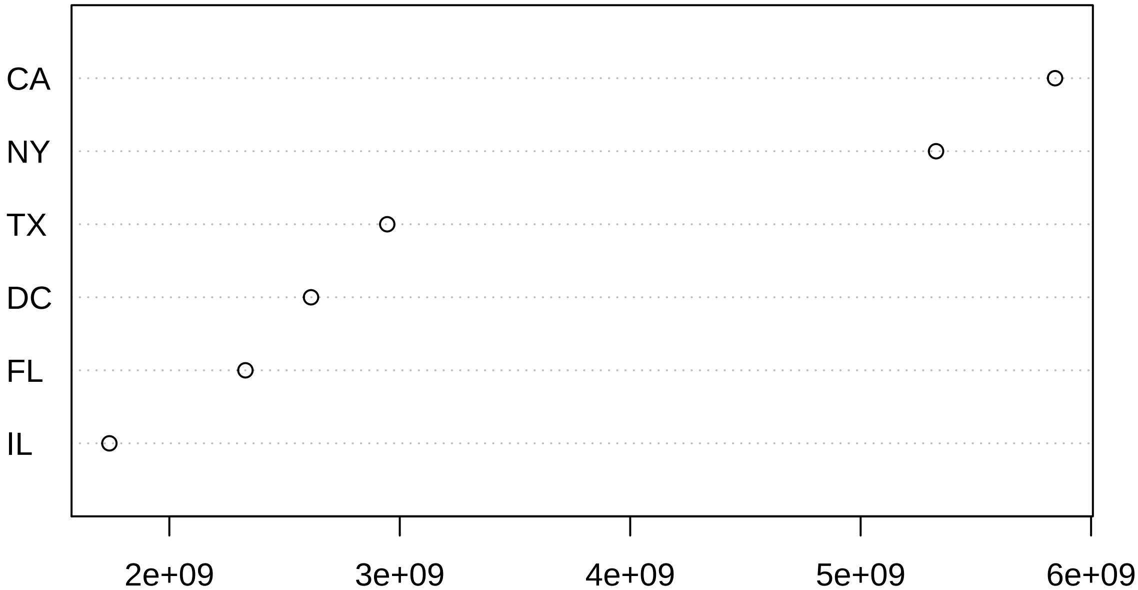
```
##       98      19      03      06      09      10      15
##     -141      -5       0       0       0       0       0
##       28      29      32      46      53      55      64
##        0       0       0       0       0       0       0
##       72      77      78      87      91      92      94
##        0       0       0       0       0       0       0
##       95      WP      LL       K      GP      SY      TW
##        0      10      27      39      40      40      50
##       VR      11      75      RJ       L      EU      BK
##       50      75      80      80     100     110     151
##       NF      ZA      81      U*      EM       Q      36
##      165     165     167     195     200     200     210
##       EE      SR      VD      (A      AF      AO      BD
##      210     210     225     250     250     250     250
##       BL      EG      FP      KI      LN      OV      RH
##      250     250     250     250     250     250     250
##       SO      TU      GY      KE       H      BW      ES
##      250     250     300     300     364     400     426
##       LX      'H      -0      67      79      90      AN
##      450     500     500     500     500     500     500
##       BU      CD      D.      DP      DU      KO      LU
##      500     500     500     500     500     500     500
##       PU      SZ      TH      TI      WO      WU      ZR
##      500     500     500     500     500     500     500
##       BO      AT      SJ      'U      17      CE      20
##      505     511     533     600     600     725     750
##       LE      GT       T      CZ       1      60      AI
##      885     900     900     950    1000    1000    1000
##        B      DI      DL      EL      FO      HM      KX
##     1000    1000    1000    1000    1000    1000    1000
```

```
##         OS         PO         UH         UR         WD         ZU         NR
##       1000       1000       1000       1000       1000       1000       1050
##         GB          <          D         FI         SH         HA          F
##       1243       1250       1250       1250       1250       1290       1318
##          A         HU          P          W         ST         JN         MY
##       1320       1350       1350       1400       1450       1500       1500
##         TA         TC         VE         FC         00         PE         SK
##       1500       1500       1500       1600       1614       1652       1665
##         NZ         EA         IS         01         ZH          G         MJ
##       1676       1750       1750       1814       1818       1875       1900
##         7E         BX         CR         GR         IB         US         WS
##       2000       2000       2000       2000       2000       2000       2000
##          V          M         KU         JA         KA         PN         QU
##       2017       2058       2064       2085       2150       2200       2250
##         RO         JP         CM         WE         ??         GL         OT
##       2250       2425       2500       2580       2650       2722       2750
##         MX         MC         TZ         PH         SI         CI         YT
##       2801       3000       3000       3044       3050       3250       3263
##         DF         NL         TE         FA         TM         BR         AU
##       3280       3851       3975       4000       4200       4600       4612
##         SE          C         HK         85          N         FS         XM
##       4750       4938       5250       5471       5650       5659       5750
##         RM         SA         HO         PW         HE         MB         CH
##       6000       6009       6555       6610       6640       6822       6919
##         BS         GE         IO         IM         NB         TO         N/
##       7000       7171       7200       9250       9486      10296      12525
##         NS         SU         BE         BA         SP         NT         MH
##      13192      13619      14107      14896      15202      16626      16965
##         CN         IT         LO         AB         NO          O         OJ
##      17661      18250      18855      19553      20472      22555      24500
##         SW         QC         FR         NU         UN         UK         BC
##      25170      27530      30948      34306      45865      83523      86064
##         EN         NA         FM         ON         XX         FF         AS
##     104140     127168     135736     172161     203843     294424     397614
##         AA         AP         MP         IR         AE         GU         VI
##     678578     681851     916011    1048182    1405541    4306645   10251026
##         ZZ         ND         VT         PR         AK         SD         ID
##   25192260   56477858   66673746   69402832   69577306   73622367   77527183
##         RI         MT         HI         WV         ME         DE         WY
##   94373994   96255694  102228962  116375772  119437063  124615709  126752246
##         NH         MS         UT         NE         NM         IA         SC
##  146933211  148489537  170795379  177995925  198249938  198590085  245366225
##         KS         AR         OR         OK         AL         KY         WI
##  252134601  262243128  273321583  298419110  304041321  317402936  389146459
##         LA         IN         MN         AZ         MO         TN         NC
##  400073166  414889271  489863090  532637404  534339055  537204492  597318167
##         CO                    NV         WA         GA         MI         CT
##  651796668  652941123  677671421  719698705  747525201  795915409  841002706
##         MD         OH         NJ         MA         PA         VA         IL
##  874546900  966562414 1004221945 1236991169 1259313542 1375724476 1739757415
##         FL         DC         TX         NY         CA
## 2330169028 2614794626 2945587994 5327197118 5843659325
```

and this looks reasonable. The greatest monetary donations, altogether, came from California, New York, Texas, etc.

We can plot this data. The y-axis shows the states, and the x-axis shows the total amount of donations, given in dollars.

```r
dotchart(tail(sort(v)))
```



If you want to just see (for instance) the first 10 rows of a database table and all of the variables, you can limit the results.

```r
myDF <- dbGetQuery(connection, "SELECT * FROM elections LIMIT 10")
myDF
```

```
##      cmte_id amndt_ind rpt_tp transaction_pgi   image_num transaction_tp
## 1  C00078279         A    M11               P 80031492155            22Y
## 2  C00078279         A    M11                 79031415137             15
## 3  C00078279         A    M11                 79031415137             15
## 4  C00078279         A    M11                 79031415137             15
## 5  C00078287         A     Q1                 79031231889             15
## 6  C00078287         A     Q1                 79031231889             15
## 7  C00078287         A     Q1                 79031231889             15
## 8  C00078287         A     Q1                 79031231889             15
## 9  C00078287         A     Q1                 79031231889             15
## 10 C00078287         A     Q1                 79031231889             15
##    entity_tp                   name    city state zip_code         employer
## 1                    MCKENNON, K R MIDLAND    MI    00000
## 2                      OREFFICE, P MIDLAND    MI    00000 DOW CHEMICAL CO
## 3                        DOWNEY, J MIDLAND    MI    00000 DOW CHEMICAL CO
## 4                         BLAIR, E MIDLAND    MI    00000 DOW CHEMICAL CO
## 5                BLANCHARD, JOHN A CHICAGO    IL    60685
## 6                   CRAMER, JOHN H CHICAGO    IL    60685
## 7                    MCHUGH, KEVIN CHICAGO    IL    60685
## 8                   NOHA, EDWARD J CHICAGO    IL    60685
## 9                RYCROFT, DONALD C CHICAGO    IL    60685
## 10          VANDERSLICE, WILLIAM D CHICAGO    IL    60685
##    occupation transaction_dt transaction_amt other_id tran_id file_num memo_cd
## 1                 1979-10-03             400                         0
```

```
## 2                 1979-10-26           1500                          0
## 3                 1979-10-26            300                          0
## 4                 1979-10-26           1000                          0
## 5                 1979-03-20            200                          0
## 6                 1979-02-28            200                          0
## 7                 1979-03-05            200                          0
## 8                 1979-03-12            300                          0
## 9                 1979-03-19            200                          0
## 10                1979-02-27            200                          0
##                         memo_text      sub_id
## 1  CONTRIBUTION REF TO INDIVIDUAL 3.06202e+18
## 2                                 3.06192e+18
## 3                                 3.06192e+18
## 4                                 3.06192e+18
## 5                                 3.06192e+18
## 6                                 3.06192e+18
## 7                                 3.06192e+18
## 8                                 3.06192e+18
## 9                                 3.06192e+18
## 10                                3.06192e+18
```

Warning: Please note that you probably do NOT want to try this: If we removed the LIMIT 10, then we would pull all of the data from the entire database table, and that would take a very long time! That is not recommended.

We can also check the employers, to compute the monetary amount were made by the employees from each company.

Warning: "Here be dragons!" The new query might take (say) 15 or 20 minutes to run.

```
myDF <- dbGetQuery(connection, "SELECT SUM(transaction_amt), employer FROM elections GROUP BY employer")
```

There are 4.4 million employwers listed in the database:

```
dim(myDF)
```

```
## [1] 4467789       2
```

Again, we save the sum of transaction amounts as v and give each element of v the name of that employer

```
v <- myDF$`SUM(transaction_amt)`
names(v) <- myDF$employer
```

then we are ready to sort the data, as we did before. There are too many results to see the whole list, so we look at the tail of the sorted results.

```
tail(sort(v))
```

```
## BLOOMBERG INC.             N/A   NOT EMPLOYED  SELF-EMPLOYED        RETIRED
##     1078605046     1151685215     1185574584     1739808973     2191230249
##
##     7319850748
```

and this looks reasonable. The greatest monetary donations, altogether, were either listed without an employer, or retired, or self-employed, or not employed, or N/A, or from Bloomberg Inc.

If we want to see a few more results, we can specify how many results we want to see in the tail command.

```
tail(sort(v), n=30)
```

```
##                      CORPORATION PALOMA PARTNERS ADVISORS, INC.
```

```
##                   35750424                    37496200
##               NOT-EMPLOYED                    INVESTOR
##                   41242677                    47676343
##   RENAISSANCE TECHNOLOGIES                         NEA
##                   47869641                    50639401
##        LAS VEGAS SANDS INC.                   PHYSICIAN
##                   52671200                    66286228
## LAS VEGAS SANDS CORPORATION         NEWSWEB CORPORATION
##                   67323975                    70225588
##              INFO REQUESTED            STATE OF FLORIDA
##                   70539797                    71432000
##                   CANDIDATE                       ULINE
##                   82183411                    98918238
##       INFORMATION REQUESTED         ADELSON DRUG CLINIC
##                  106636889                   108578400
##                BLOOMBERG LP                   HOUSEWIFE
##                  114417573                   131168498
##                    ATTORNEY                   FAHR, LLC
##                  143690457                   248244304
##                   HOMEMAKER                        NONE
##                  754539553                   768247155
##               SELF EMPLOYED                        SELF
##                  770420301                  1026722788
##              BLOOMBERG INC.                         N/A
##                 1078605046                  1151685215
##                NOT EMPLOYED               SELF-EMPLOYED
##                 1185574584                  1739808973
##                     RETIRED                            
##                 2191230249                  7319850748
```

Notice that there are lots of missing data values, and data for which the employer was not listed. This is common with real world data, and it is something to get used to, when working on data analysis.