TSL Project

L. Insolia, J. Kim and G. Yeghikyan

04/03/2018

Contents

General information

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

We analyze a dataset published on Kaggle. It refers to french employment, salaries, population per town. The aim is to evaluate equality/inequalities in France, and geographical distribution of business according to their size.

Such data are collected by the INSEE. Information regarding the number of firms in every french town, categorized by size can be found here. Information about salaries around french town per job categories, age

and sex (expressed in average net amount per hour in euro) can be found here. Demographic information in France per town, age, sex and living mode can be found here.

Additional info about Population Data can be found here, it allows to add catÃ©gorie socioprofessionnelle.

Aim of the study

This project aims to explore existing patterns among French towns.

In particular, we are interested in:

- evaluating possible ineqialities: per towns/region, sex, age, etc.;
- predicting the ... using a regression model;
- reduce the dimensionality of ... performing a PCA;
- explore different algorithms to cluster male/females using ...

General pre-processing phase

Import data:

```
# options(encoding = "UTF-8")  # for Mac [PROBABLY, to check]
# options(encoding = "ISO-8859-1")  # for Windows [PROBABLY NOT WORKING]
setwd("./data")
firms       <- read.csv("base_etablissement_par_tranche_effectif.csv", encoding = "UTF-8")
geo         <- read.csv("name_geographic_information.csv", encoding = "UTF-8")
salary      <- read.csv("net_salary_per_town_categories.csv", encoding = "UTF-8")
population  <- read.csv("population.csv", encoding = "UTF-8")
```

Check variable names:

```
names(firms)
```

```
##  [1] "CODGEO"   "LIBGEO"   "REG"      "DEP"      "E14TST"   "E14TS0ND"
##  [7] "E14TS1"   "E14TS6"   "E14TS10"  "E14TS20"  "E14TS50"  "E14TS100"
## [13] "E14TS200" "E14TS500"
```

```
names(population)
```

```
## [1] "NIVGEO"    "CODGEO"    "LIBGEO"    "MOCO"      "AGEQ80_17" "SEXE"
## [7] "NB"
```

```
names(salary)
```

```
##  [1] "CODGEO"    "LIBGEO"    "SNHM14"    "SNHMC14"   "SNHMP14"
##  [6] "SNHME14"   "SNHMO14"   "SNHMF14"   "SNHMFC14"  "SNHMFP14"
## [11] "SNHMFE14"  "SNHMFO14"  "SNHMH14"   "SNHMHC14"  "SNHMHP14"
## [16] "SNHMHE14"  "SNHMHO14"  "SNHM1814"  "SNHM2614"  "SNHM5014"
## [21] "SNHMF1814" "SNHMF2614" "SNHMF5014" "SNHMH1814" "SNHMH2614"
## [26] "SNHMH5014"
```

```
names(geo)
```

```
##  [1] "EU_circo"               "code_région"
##  [3] "nom_région"             "chef.lieu_région"
##  [5] "numéro_département"     "nom_département"
##  [7] "préfecture"             "numéro_circonscription"
##  [9] "nom_commune"            "codes_postaux"
```

```
## [11] "code_insee"              "latitude"
## [13] "longitude"               "éloignement"
```

To better understand the data we assign meaningful names and drop some variables which are not needed (at the moment):

```
names(firms)[2:ncol(firms)] <-
  c("town",
    "regNum",
    "deptNum",
    "total",
    "null",
    "firmsEmpl_1_5",
    "firmsEmpl_6_9",
    "firmsEmpl_10_19",
    "firmsEmpl_20_49",
    "firmsEmpl_50_99",
    "firmsEmpl_100_199",
    "firmsEmpl_200_499",
    "firmsEmpl_500plus")

names(salary)[2:ncol(salary)] <-
  c("town",
    "sal_general",
    "sal_executive",
    "sal_midManager",
    "sal_employee",
    "sal_worker",
    "sal_Females",
    "sal_F_executive",
    "sal_F_midManager",
    "sal_F_employee",
    "sal_F_worker",
    "sal_Males",
    "sal_M_executive",
    "sal_M_midManager",
    "sal_M_employee",
    "sal_M_worker",
    "sal_18_25",
    "sal_26_50",
    "sal_51plus",
    "sal_F_18_25",
    "sal_F_26_50",
    "sal_F_51plus",
    "sal_M_18_25",
    "sal_M_26_50",
    "sal_M_51plus")

names(population)[5:7] <-
  c("ageCateg5",
    "sex",
    "peopleCategNum")

# Drop unnecessary columns (code/num and name represents same thing)
geo <- subset(geo, select = -c(EU_circo, code_région, numéro_département, préfecture, numéro_circonscri
```

```r
# change names
names(geo)[1:6] =
  c("region",
    "region_capital",
    "department",
    "town_name",
    "postal_code",
    "CODGEO")
```

Check variable names:

```r
names(firms)
```

```
##  [1] "CODGEO"            "town"              "regNum"
##  [4] "deptNum"           "total"             "null"
##  [7] "firmsEmpl_1_5"     "firmsEmpl_6_9"     "firmsEmpl_10_19"
## [10] "firmsEmpl_20_49"   "firmsEmpl_50_99"   "firmsEmpl_100_199"
## [13] "firmsEmpl_200_499" "firmsEmpl_500plus"
```

```r
names(population)
```

```
## [1] "NIVGEO"         "CODGEO"         "LIBGEO"         "MOCO"
## [5] "ageCateg5"      "sex"            "peopleCategNum"
```

```r
names(salary)
```

```
##  [1] "CODGEO"           "town"             "sal_general"
##  [4] "sal_executive"    "sal_midManager"   "sal_employee"
##  [7] "sal_worker"       "sal_Females"      "sal_F_executive"
## [10] "sal_F_midManager" "sal_F_employee"   "sal_F_worker"
## [13] "sal_Males"        "sal_M_executive"  "sal_M_midManager"
## [16] "sal_M_employee"   "sal_M_worker"     "sal_18_25"
## [19] "sal_26_50"        "sal_51plus"       "sal_F_18_25"
## [22] "sal_F_26_50"      "sal_F_51plus"     "sal_M_18_25"
## [25] "sal_M_26_50"      "sal_M_51plus"
```

```r
names(geo)
```

```
## [1] "region"         "region_capital" "department"     "town_name"
## [5] "postal_code"    "CODGEO"         "latitude"       "longitude"
```

[[MOVE LATER]] According to the information provided, the CODGEO variable (in firms, salary and population) and code_insee (in geo) have to be merged. However, for different reasons already identified by a kaggle user on his kernel they do not. To do so:

```r
firms$CODGEO <- sub("A", "0", firms$CODGEO)
firms$CODGEO <- sub("B", "0", firms$CODGEO)
salary$CODGEO <- sub("A", "0", salary$CODGEO)
salary$CODGEO <- sub("B", "0", salary$CODGEO)
population$CODGEO <- sub("A", "0", population$CODGEO)
population$CODGEO <- sub("B", "0", population$CODGEO)
```

Analyze firms data

Pre-processing

```
# preliminary checks
dim(firms)
```

```
## [1] 36681    14
```

```
names(firms)
```

```
##  [1] "CODGEO"            "town"              "regNum"
##  [4] "deptNum"           "total"             "null"
##  [7] "firmsEmpl_1_5"     "firmsEmpl_6_9"     "firmsEmpl_10_19"
## [10] "firmsEmpl_20_49"   "firmsEmpl_50_99"   "firmsEmpl_100_199"
## [13] "firmsEmpl_200_499" "firmsEmpl_500plus"
```

```
head(firms)
```

```
##   CODGEO                   town regNum deptNum total null firmsEmpl_1_5
## 1  01001 L'Abergement-Clémenciat     82      01    25   22             1
## 2  01002   L'Abergement-de-Varey     82      01    10    9             1
## 3  01004       Ambérieu-en-Bugey     82      01   996  577           272
## 4  01005     Ambérieux-en-Dombes     82      01    99   73            20
## 5  01006                 Ambléon     82      01     4    4             0
## 6  01007                Ambronay     82      01   124   87            20
##   firmsEmpl_6_9 firmsEmpl_10_19 firmsEmpl_20_49 firmsEmpl_50_99
## 1             2               0               0              0
## 2             0               0               0              0
## 3            63              46              24              9
## 4             3               1               2              0
## 5             0               0               0              0
## 6            10               5               2              0
##   firmsEmpl_100_199 firmsEmpl_200_499 firmsEmpl_500plus
## 1                 0                 0                0
## 2                 0                 0                0
## 3                 3                 2                0
## 4                 0                 0                0
## 5                 0                 0                0
## 6                 0                 0                0
```

```
str(firms)
```

```
## 'data.frame':    36681 obs. of  14 variables:
##  $ CODGEO          : chr  "01001" "01002" "01004" "01005" ...
##  $ town            : Factor w/ 34142 levels "Aast","Abainville",..: 13659 13661 442 444 460 480 484
##  $ regNum          : int  82 82 82 82 82 82 82 82 82 82 ...
##  $ deptNum         : Factor w/ 101 levels "01","02","03",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ total           : int  25 10 996 99 4 124 48 22 33 14 ...
##  $ null            : int  22 9 577 73 4 87 28 17 23 11 ...
##  $ firmsEmpl_1_5   : int  1 1 272 20 0 20 15 4 8 2 ...
##  $ firmsEmpl_6_9   : int  2 0 63 3 0 10 2 1 1 1 ...
##  $ firmsEmpl_10_19 : int  0 0 46 1 0 5 3 0 0 0 ...
##  $ firmsEmpl_20_49 : int  0 0 24 2 0 2 0 0 0 0 ...
##  $ firmsEmpl_50_99 : int  0 0 9 0 0 0 0 0 0 0 ...
##  $ firmsEmpl_100_199: int  0 0 3 0 0 0 0 0 1 0 ...
```

```
##  $ firmsEmpl_200_499: int  0 0 2 0 0 0 0 0 0 0 ...
##  $ firmsEmpl_500plus: int  0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(firms)
```

```
##     CODGEO                      town           regNum         deptNum
##  Length:36681       Sainte-Colombe:   14   Min.   : 1.00   62     :  895
##  Class :character   Saint-Sauveur :   12   1st Qu.:25.00   02     :  816
##  Mode  :character   Beaulieu      :   11   Median :43.00   80     :  782
##                     Sainte-Marie  :   11   Mean   :49.42   76     :  745
##                     Le Pin        :   10   3rd Qu.:73.00   57     :  730
##                     Saint-Aubin   :   10   Max.   :94.00   14     :  706
##                     (Other)       :36613                   (Other):32007
##      total                null          firmsEmpl_1_5
##  Min.   :     0.0   Min.   :     0.0   Min.   :    0.00
##  1st Qu.:     8.0   1st Qu.:     6.0   1st Qu.:    1.00
##  Median :    19.0   Median :    14.0   Median :    3.00
##  Mean   :   123.5   Mean   :    83.6   Mean   :   27.29
##  3rd Qu.:    54.0   3rd Qu.:    39.0   3rd Qu.:   11.00
##  Max.   :427385.0   Max.   :316603.0   Max.   :76368.00
##
##  firmsEmpl_6_9       firmsEmpl_10_19   firmsEmpl_20_49
##  Min.   :    0.000   Min.   :    0.0   Min.   :   0.000
##  1st Qu.:    0.000   1st Qu.:    0.0   1st Qu.:   0.000
##  Median :    0.000   Median :    0.0   Median :   0.000
##  Mean   :    5.221   Mean   :    3.8   Mean   :   2.296
##  3rd Qu.:    2.000   3rd Qu.:    1.0   3rd Qu.:   1.000
##  Max.   :14836.000   Max.   :10829.0   Max.   :5643.000
##
##  firmsEmpl_50_99     firmsEmpl_100_199  firmsEmpl_200_499
##  Min.   :   0.0000   Min.   :  0.0000   Min.   :  0.0000
##  1st Qu.:   0.0000   1st Qu.:  0.0000   1st Qu.:  0.0000
##  Median :   0.0000   Median :  0.0000   Median :  0.0000
##  Mean   :   0.7383   Mean   :  0.3324   Mean   :  0.1728
##  3rd Qu.:   0.0000   3rd Qu.:  0.0000   3rd Qu.:  0.0000
##  Max.   :1658.0000   Max.   :812.0000   Max.   :456.0000
##
##  firmsEmpl_500plus
##  Min.   :  0.00000
##  1st Qu.:  0.00000
##  Median :  0.00000
##  Mean   :  0.04842
##  3rd Qu.:  0.00000
##  Max.   :180.00000
##
```

```
# converting CODGEO format
firms$CODGEO <- as.numeric(firms$CODGEO)

# Check for duplicated data
sum(duplicated.data.frame(firms))
```

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

```
# Categorize firms' size according to EU standard, but slightly different for medium and large firms (m
# http://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Enterprise_size
```

```r
firms$micro   <- firms$firmsEmpl_1_5 + firms$firmsEmpl_6_9
firms$small   <- firms$firmsEmpl_10_19 + firms$firmsEmpl_20_49
firms$medium  <- firms$firmsEmpl_50_99 + firms$firmsEmpl_100_199
firms$large   <- firms$firmsEmpl_200_499 + firms$firmsEmpl_500plus

# Drop unnecessary (at the moment) columns
firms <- subset(firms, select = c(CODGEO, town, total, micro, small, medium, large, null))

# check
head(firms)
```

```
##   CODGEO                   town total micro small medium large null
## 1   1001 L'Abergement-Clémenciat    25     3     0      0     0   22
## 2   1002   L'Abergement-de-Varey    10     1     0      0     0    9
## 3   1004       Ambérieu-en-Bugey   996   335    70     12     2  577
## 4   1005     Ambérieux-en-Dombes    99    23     3      0     0   73
## 5   1006                 Ambléon     4     0     0      0     0    4
## 6   1007                Ambronay   124    30     7      0     0   87
```

```r
summary(firms)
```

```
##      CODGEO                  town           total
##  Min.   : 1001   Sainte-Colombe:   14   Min.   :     0.0
##  1st Qu.:24564   Saint-Sauveur :   12   1st Qu.:     8.0
##  Median :48171   Beaulieu      :   11   Median :    19.0
##  Mean   :46263   Sainte-Marie  :   11   Mean   :   123.5
##  3rd Qu.:67012   Le Pin        :   10   3rd Qu.:    54.0
##  Max.   :97617   Saint-Aubin   :   10   Max.   :427385.0
##                  (Other)       :36613
##      micro              small              medium
##  Min.   :    0.00   Min.   :    0.000   Min.   :   0.000
##  1st Qu.:    1.00   1st Qu.:    0.000   1st Qu.:   0.000
##  Median :    4.00   Median :    0.000   Median :   0.000
##  Mean   :   32.51   Mean   :    6.097   Mean   :   1.071
##  3rd Qu.:   13.00   3rd Qu.:    2.000   3rd Qu.:   0.000
##  Max.   :91204.00   Max.   :16472.000   Max.   :2470.000
##
##      large               null
##  Min.   :  0.0000   Min.   :     0.0
##  1st Qu.:  0.0000   1st Qu.:     6.0
##  Median :  0.0000   Median :    14.0
##  Mean   :  0.2212   Mean   :    83.6
##  3rd Qu.:  0.0000   3rd Qu.:    39.0
##  Max.   :636.0000   Max.   :316603.0
##
```

```r
str(firms)
```

```
## 'data.frame':    36681 obs. of  8 variables:
##  $ CODGEO: num  1001 1002 1004 1005 1006 ...
##  $ town  : Factor w/ 34142 levels "Aast","Abainville",..: 13659 13661 442 444 460 480 484 581 638 80
##  $ total : int  25 10 996 99 4 124 48 22 33 14 ...
##  $ micro : int  3 1 335 23 0 30 17 5 9 3 ...
##  $ small : int  0 0 70 3 0 7 3 0 0 0 ...
##  $ medium: int  0 0 12 0 0 0 0 0 1 0 ...
```

```
##  $ large : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ null  : int  22 9 577 73 4 87 28 17 23 11 ...
```

```
# deptNum has to be factor?
```

EDA

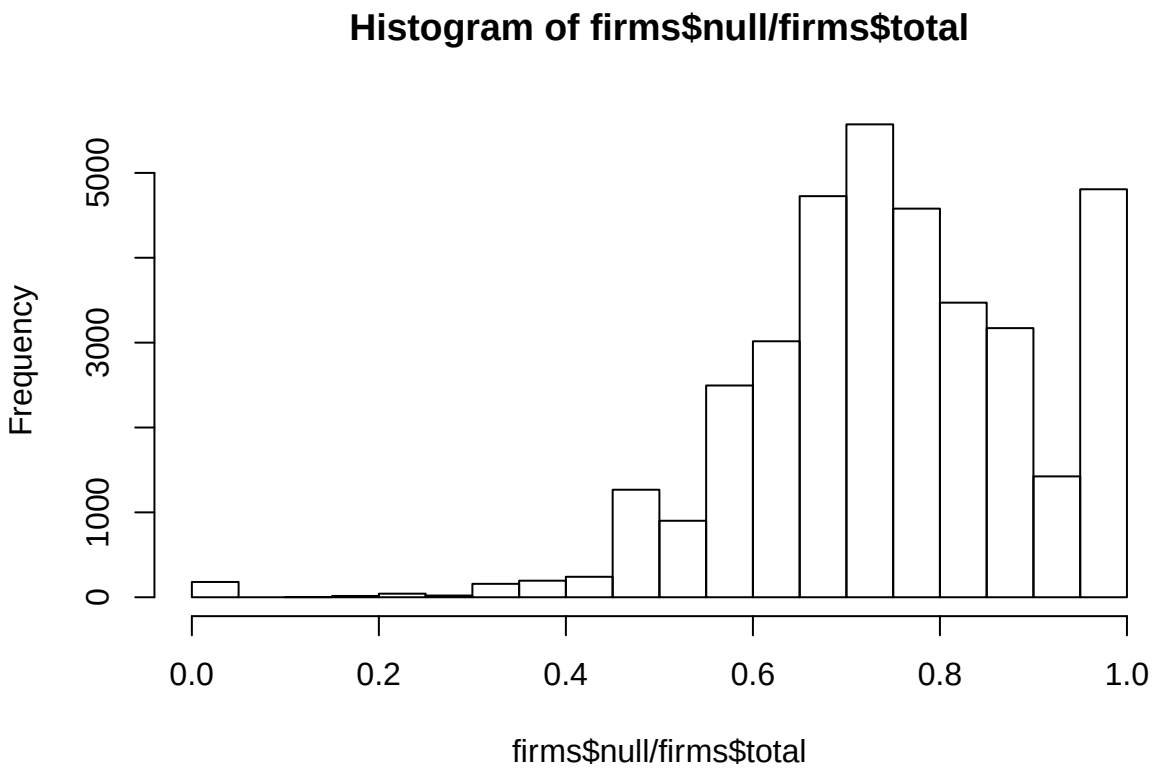
```
# there is an obs with more than 316K null data: we check if it is plausible
# get the highest 20 null values
str_firms <- sort(firms$null, decreasing = T)[1:20]
# get their indexes
str_firms_ind <- match(str_firms, firms$null)
# get the corresponding city
firms$town[str_firms_ind]
```

```
##  [1] Paris               Marseille           Lyon
##  [4] Nice                Toulouse            Bordeaux
##  [7] Montpellier         Nantes              Strasbourg
## [10] Lille               Aix-en-Provence     Boulogne-Billancourt
## [13] Fort-de-France      Rennes              Grenoble
## [16] Toulon              Cannes              Saint-Denis
## [19] Neuilly-sur-Seine   Nîmes
## 34142 Levels: Aast Abainville Abancourt Abaucourt ... Zuytpeene
```

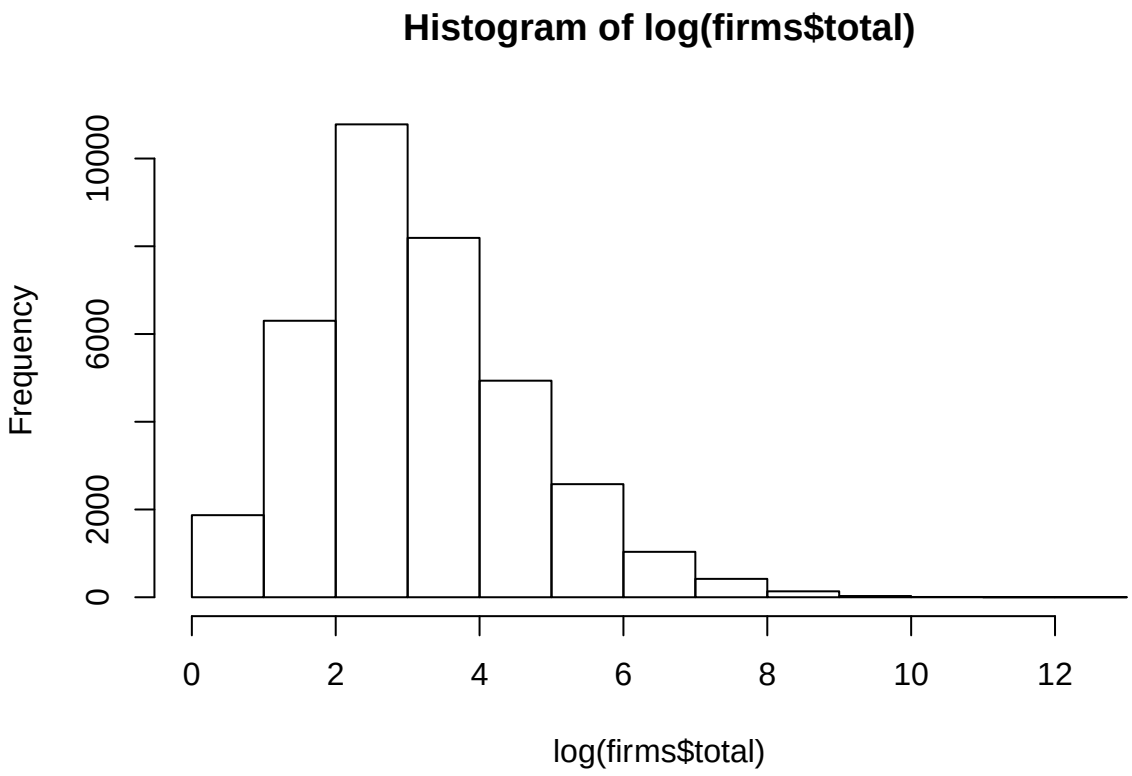
```
# hence, it seems reasonable..

# check the ratio of null for each town
summary(firms$null/firms$total)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.6603  0.7500  0.7526  0.8571  1.0000     399
```

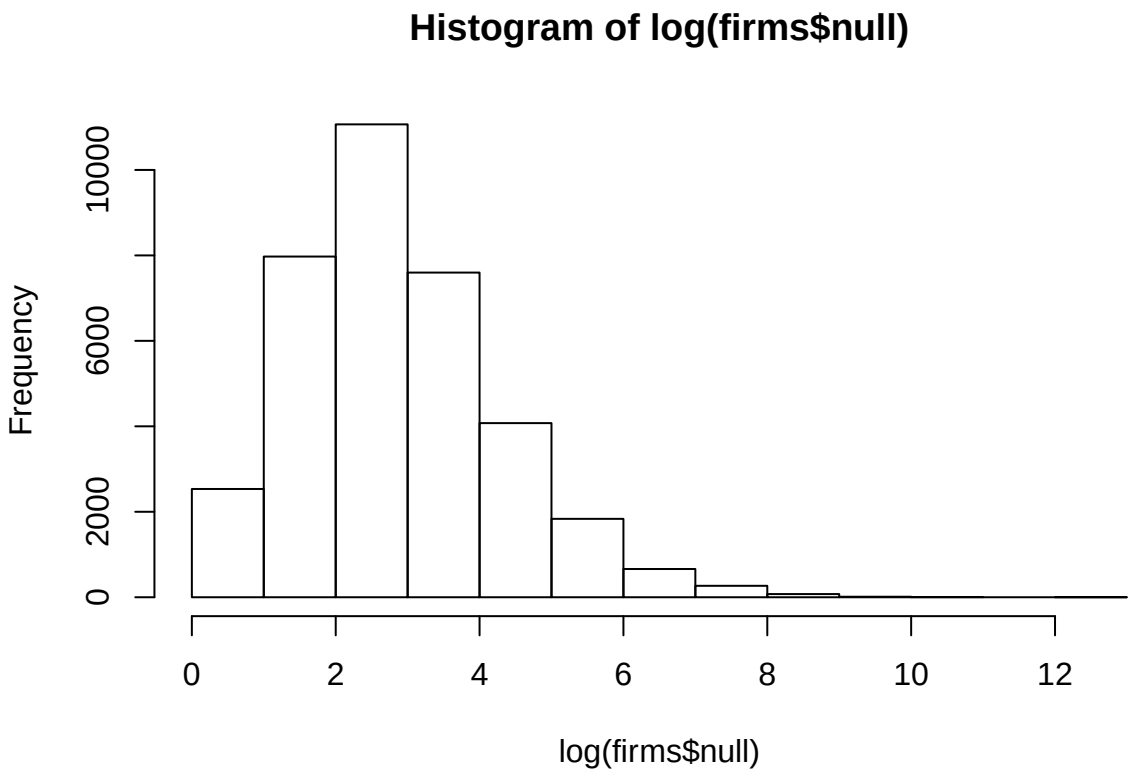
```
# a lot of information is missing
# should we remove these data?
hist(firms$null/firms$total)
```



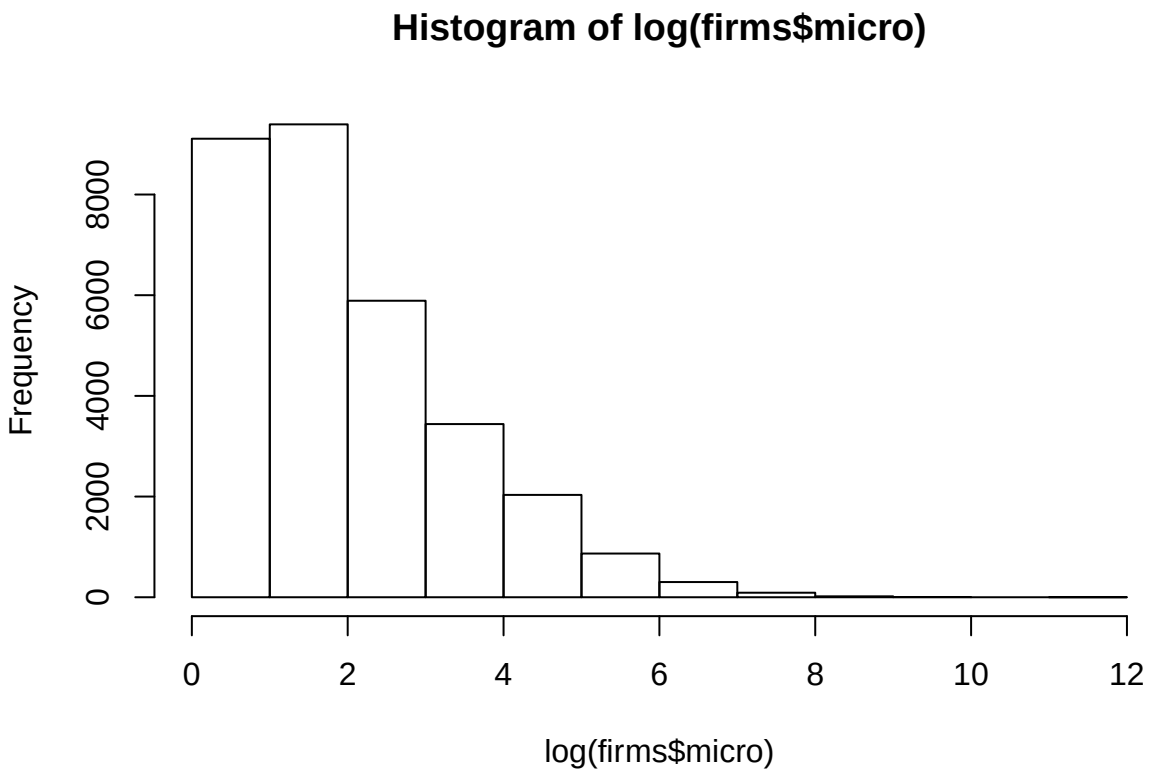

```r
# evaluate the distribution of all the sizes
# (log vs. ratio wrt total?)
hist(log(firms$total))
```



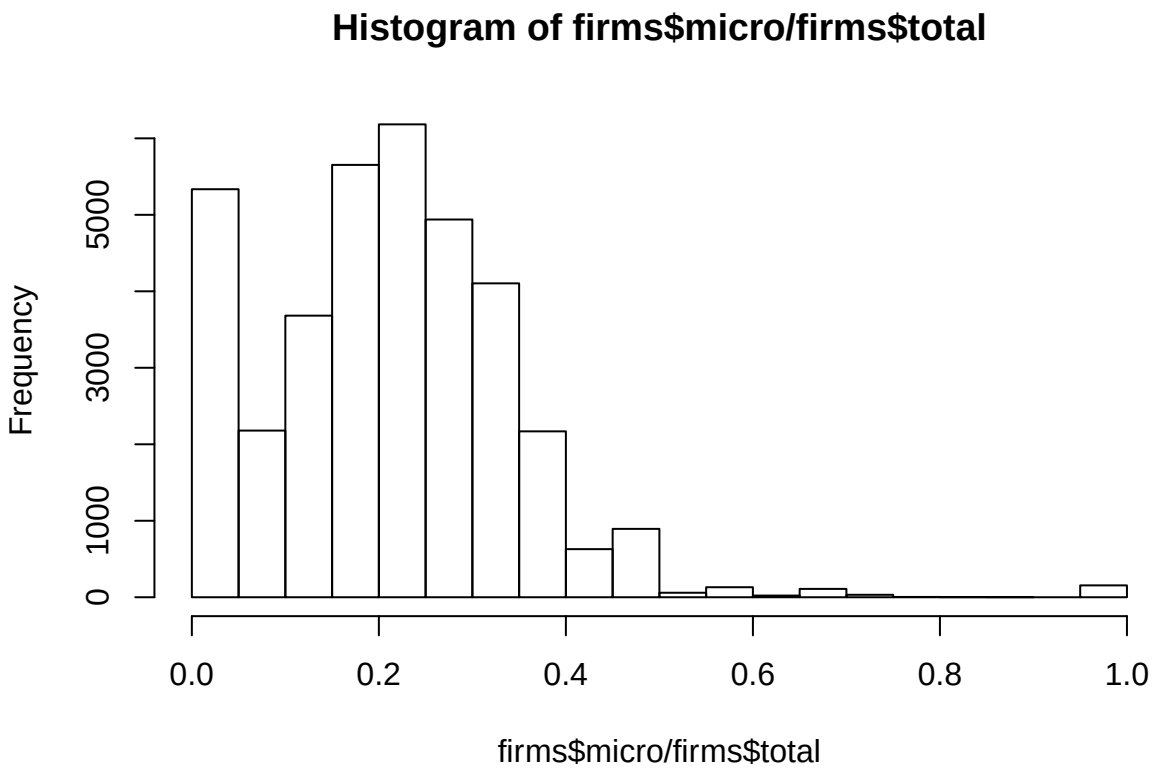

```
hist(log(firms$null))
```



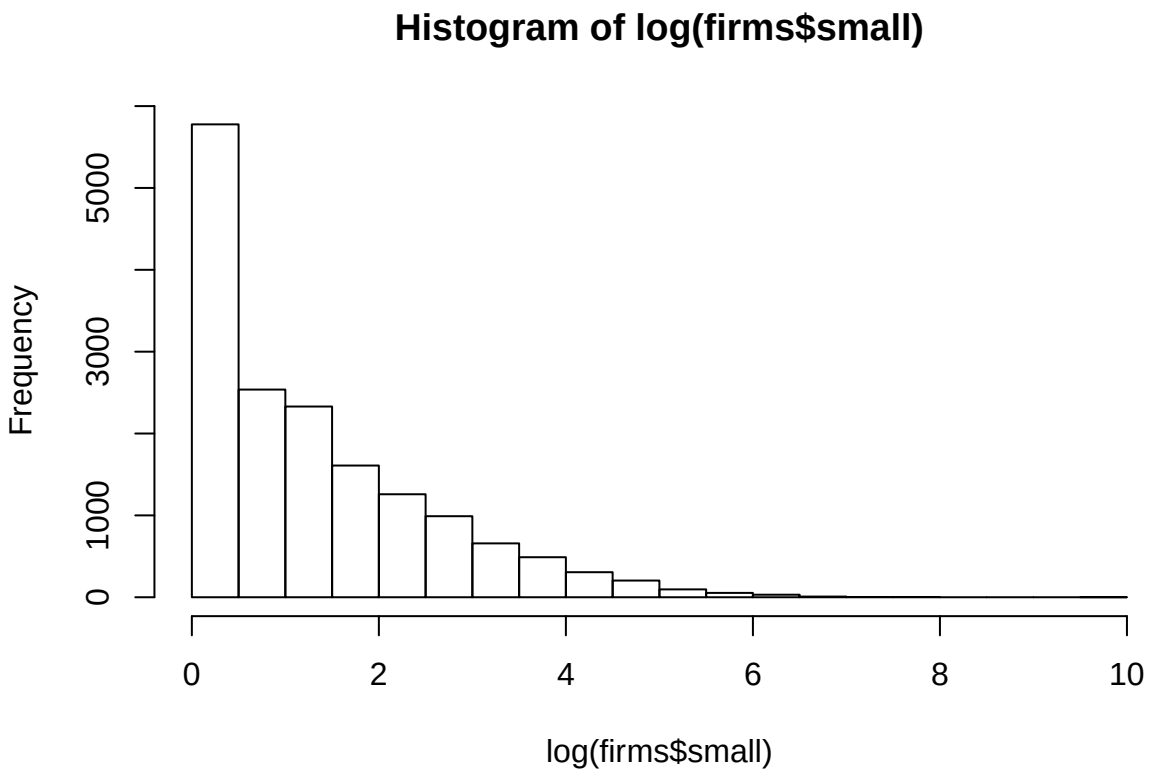

```
hist(log(firms$micro))
```



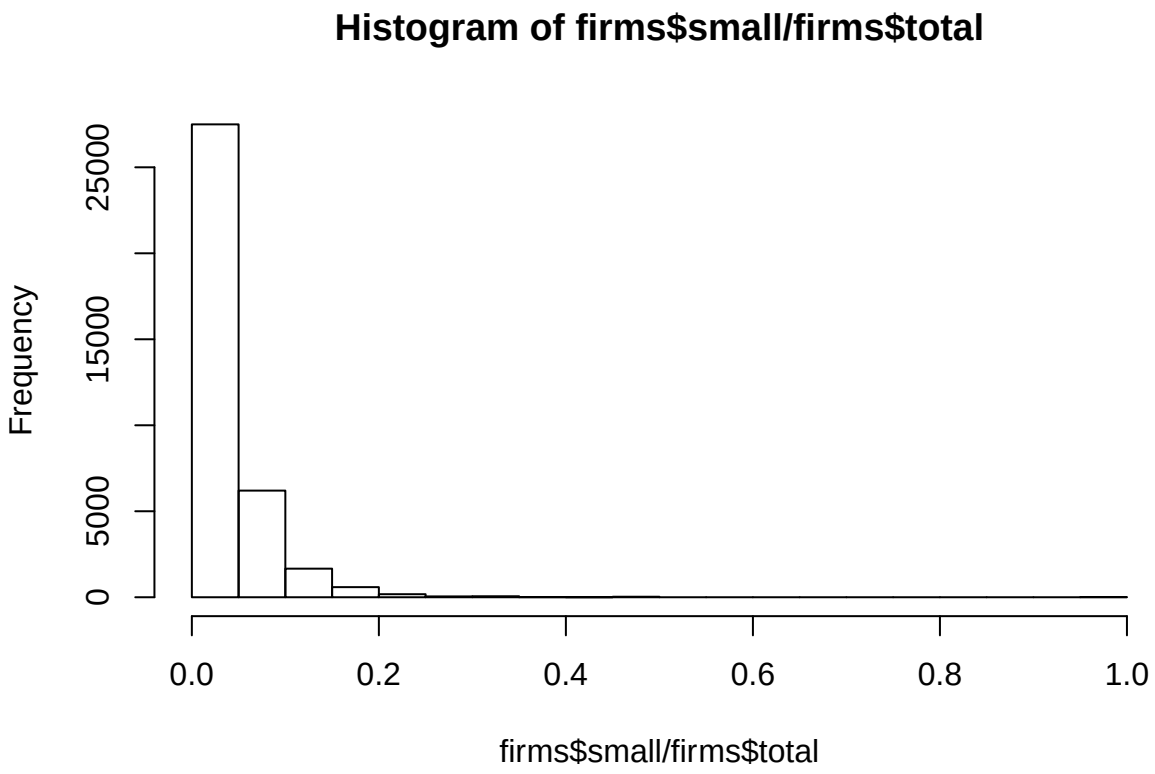

```
hist(firms$micro/firms$total)
```



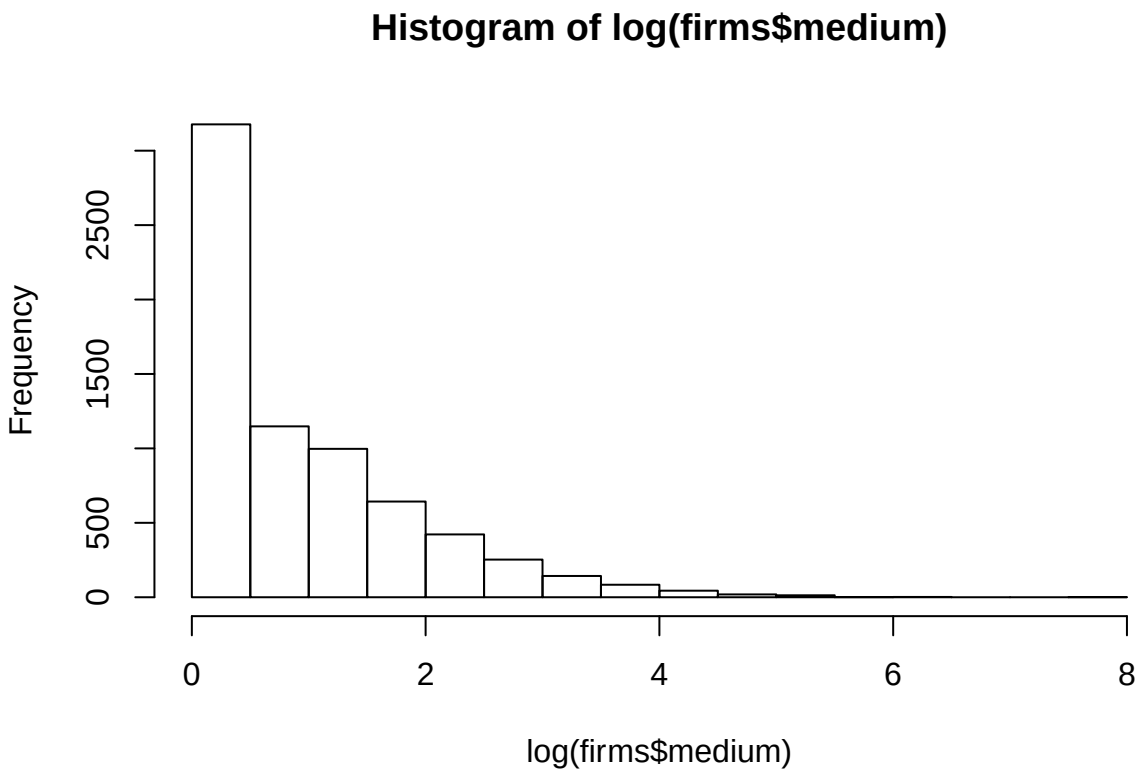

```
hist(log(firms$small))
```



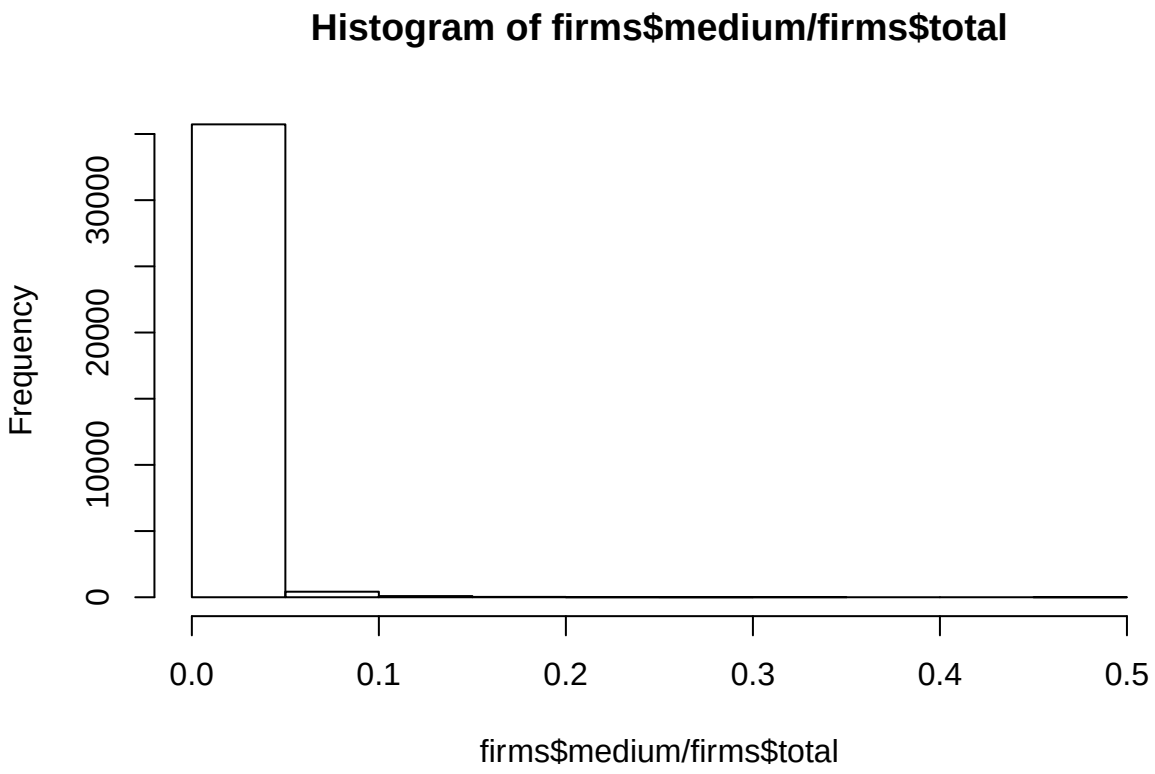

```
hist(firms$small/firms$total)
```



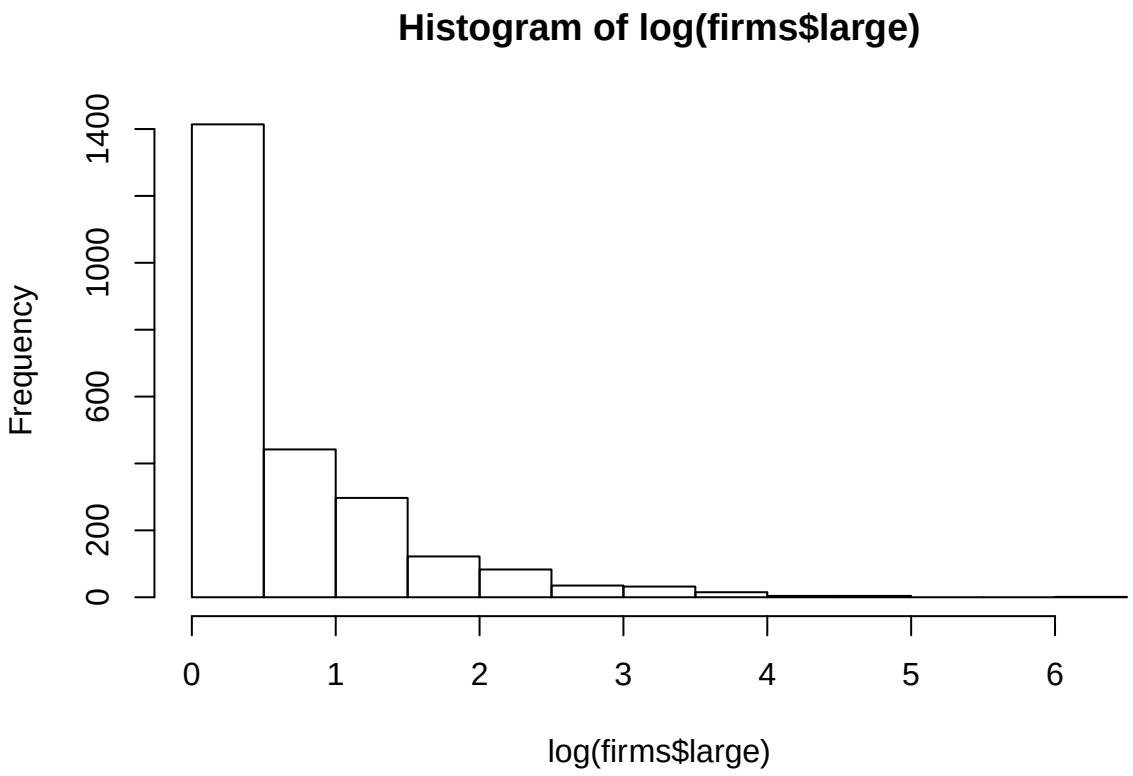

```
hist(log(firms$medium))
```



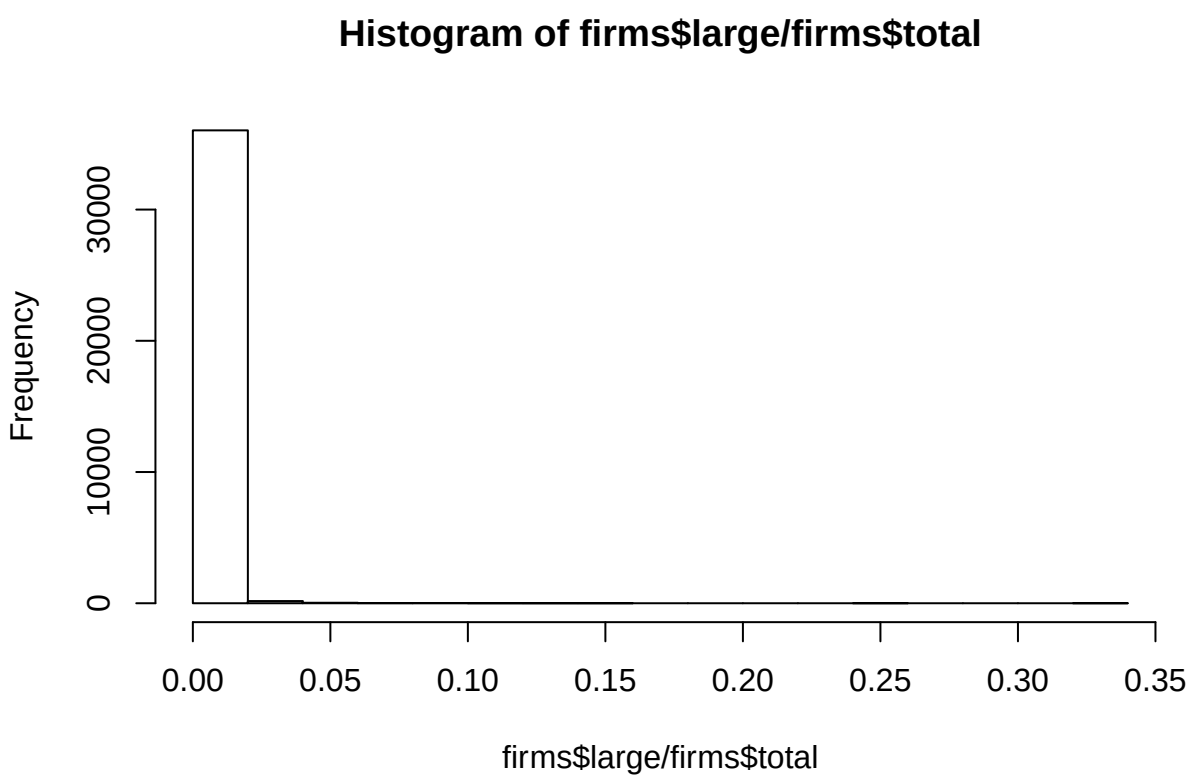

```
hist(firms$medium/firms$total)
```




```
hist(log(firms$large))
```




```
hist(firms$large/firms$total)
```




```
# keep only logs?
```

PCA on firms data:

```
firms_clean <- firms[firms$micro < 20000 & firms$large < 200,]
myPr <- prcomp(firms_clean[, 4:8], scale = TRUE)
#plot(scale(firms_clean$micro), scale(firms_clean$large))
#mean(firms_clean$micro)
#mean(firms_clean$large)
myPr
```

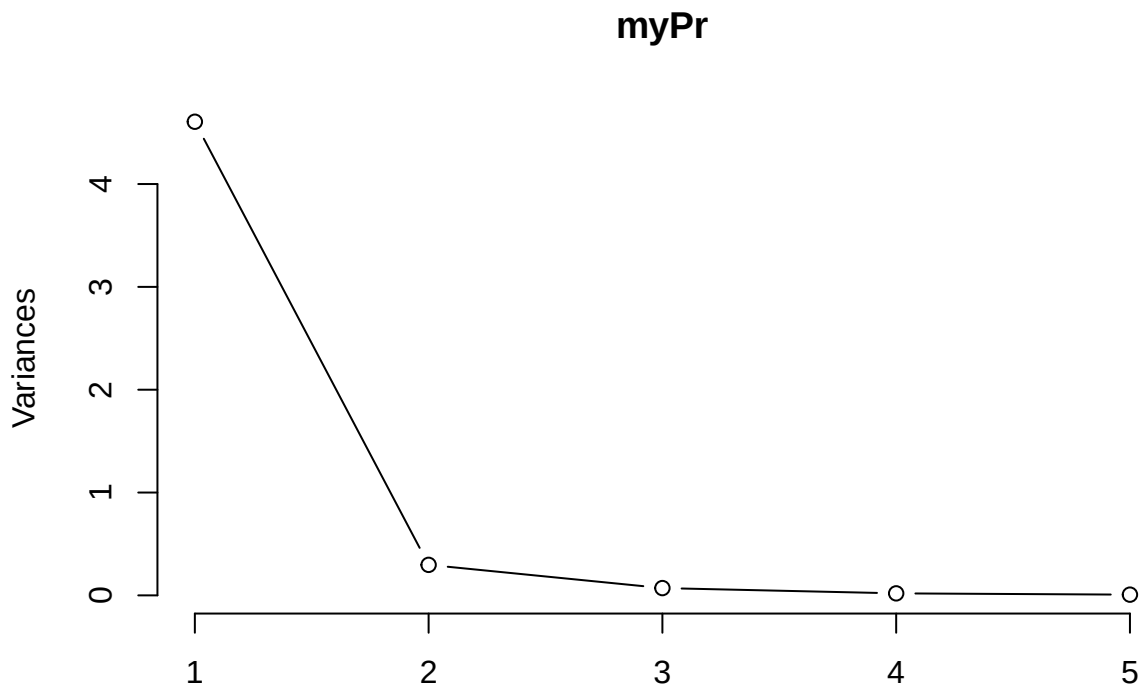
```
## Standard deviations (1, .., p=5):
## [1] 2.14610999 0.54465371 0.26631012 0.13983882 0.08419175
##
## Rotation (n x k) = (5 x 5):
##              PC1         PC2         PC3        PC4         PC5
## micro  0.4533436  0.40064594 -0.03896827  0.3114187  0.73175293
## small  0.4605093  0.09419396  0.42164740  0.5486804 -0.54792511
## medium 0.4547148 -0.24046742  0.56534498 -0.6277884  0.14722974
## large  0.4207158 -0.75389742 -0.46725510  0.1841806  0.04885769
## null   0.4456943  0.45213319 -0.53174492 -0.4170461 -0.37450240
```

```
summary(myPr)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5
## Standard deviation     2.1461 0.54465 0.26631 0.13984 0.08419
## Proportion of Variance 0.9212 0.05933 0.01418 0.00391 0.00142
```

```
## Cumulative Proportion  0.9212 0.98049 0.99467 0.99858 1.00000
```

```
plot(myPr, type = "l")
```




```
biplot(myPr, scale = 0)
#extract PC scores...
str(myPr)
```

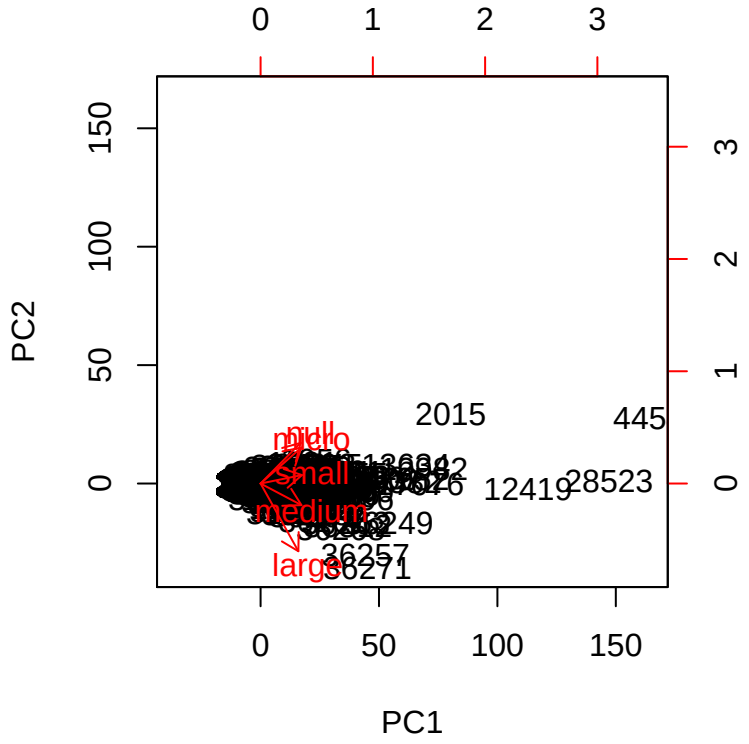
```
## List of 5
##  $ sdev    : num [1:5] 2.1461 0.5447 0.2663 0.1398 0.0842
##  $ rotation: num [1:5, 1:5] 0.453 0.461 0.455 0.421 0.446 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "micro" "small" "medium" "large" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:5] 30.026 5.648 1.003 0.204 74.926
##   ..- attr(*, "names")= chr [1:5] "micro" "small" "medium" "large" ...
##  $ scale   : Named num [1:5] 198.08 36.67 7.05 1.94 510.97
##   ..- attr(*, "names")= chr [1:5] "micro" "small" "medium" "large" ...
##  $ x       : num [1:36680, 1:5] -0.288 -0.304 3.043 -0.16 -0.31 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:36680] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
#myPr$x #checking principal component scores
firms2 <- cbind(firms_clean, myPr$x[, 1:2])
head(firms2)
```

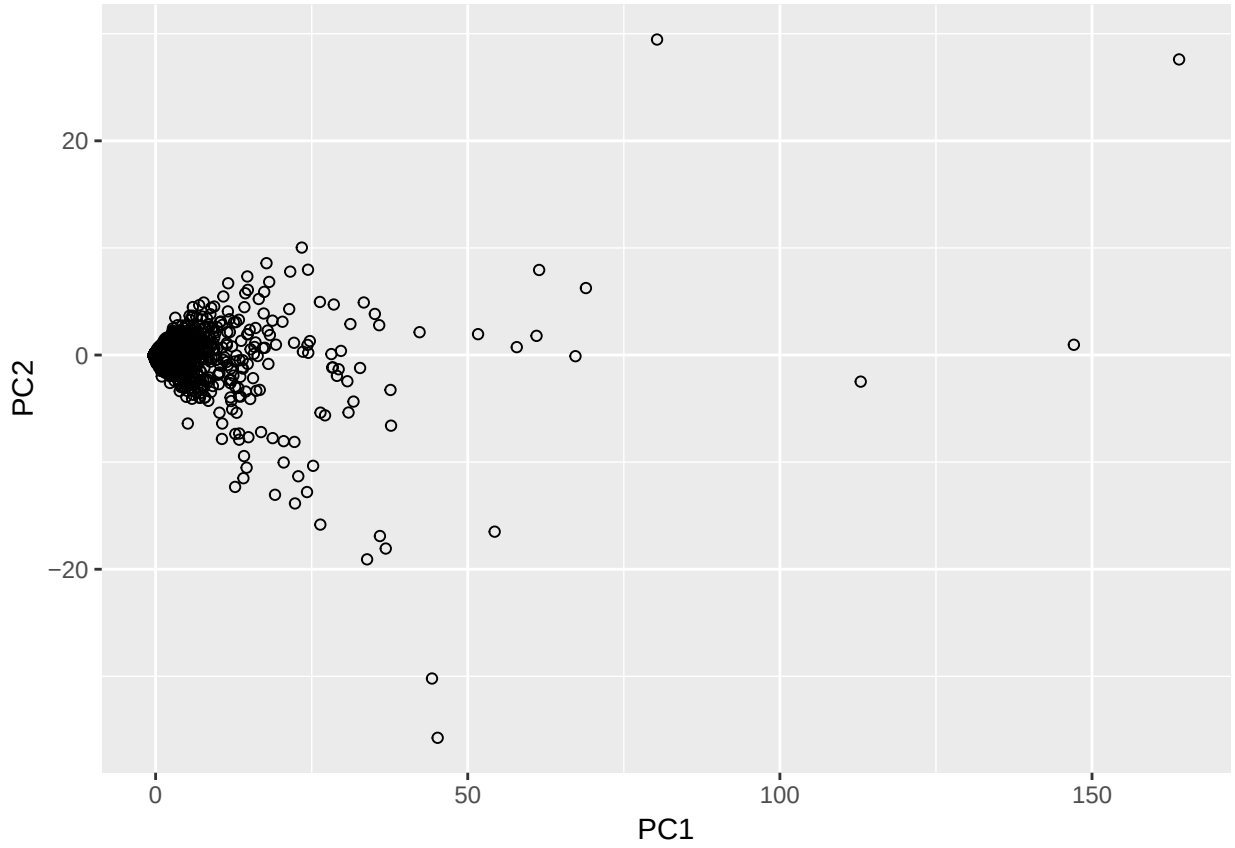
```
##   CODGEO                    town total micro small medium large null
## 1   1001 L'Abergement-Clémenciat    25     3     0      0     0   22
## 2   1002   L'Abergement-de-Varey    10     1     0      0     0    9
## 3   1004       Ambérieu-en-Bugey   996   335    70     12     2  577
## 4   1005     Ambérieux-en-Dombes    99    23     3      0     0   73
## 5   1006                 Ambléon     4     0     0      0     0    4
## 6   1007                Ambronay   124    30     7      0     0   87
##          PC1          PC2
## 1 -0.2879301 -0.002515545
## 2 -0.3038468 -0.018063930
## 3  3.0434231  0.152878901
## 4 -0.1599969  0.090770640
## 5 -0.3104967 -0.024510834
## 6 -0.0815311  0.127591863
```

```
#plot with ggplot...
require(ggplot2)
```

```
## Loading required package: ggplot2
```




```
ggplot(firms2, aes(PC1, PC2)) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```




```
# correlations between variables and PCs...
cor(firms_clean[, 4:8], firms2[,9:10])
```

```
##              PC1         PC2
## micro  0.9729251  0.21821330
## small  0.9883037  0.05130309
## medium 0.9758680 -0.13097147
## large  0.9029024 -0.41061303
## null   0.9565091  0.24625602
```

What we have learned

- More micro firms than small ones
- ...

How to use these data

We plan to use these for the following tasks:

- predict the salaries using such information as proxy for the competition in the job market;
- predict the total number of firms, using salary data;
- geo-spatial plot for firms' size
- ...

Analyze geographical data

Pre-processing

```
# preliminary checks
dim(geo)
```

```
## [1] 36840     8
```

```
names(geo)
```

```
## [1] "region"         "region_capital" "department"     "town_name"
## [5] "postal_code"    "CODGEO"         "latitude"       "longitude"
```

```
head(geo)
```

```
##        region region_capital department            town_name postal_code
## 1 Rhône-Alpes           Lyon        Ain             Attignat       01340
## 2 Rhône-Alpes           Lyon        Ain             Beaupont       01270
## 3 Rhône-Alpes           Lyon        Ain                 Bény       01370
## 4 Rhône-Alpes           Lyon        Ain            Béreyziat       01340
## 5 Rhône-Alpes           Lyon        Ain Bohas-Meyriat-Rignat       01250
## 6 Rhône-Alpes           Lyon        Ain      Bourg-en-Bresse       01000
##   CODGEO latitude longitude
## 1   1024 46.28333  5.166667
## 2   1029 46.40000  5.266667
## 3   1038 46.33333  5.283333
## 4   1040 46.36667      5.05
## 5   1245 46.13333       5.4
## 6   1053 46.20000  5.216667
```

```
str(geo)
```

```
## 'data.frame':    36840 obs. of  8 variables:
##  $ region        : Factor w/ 28 levels "Alsace","Aquitaine",..: 27 27 27 27 27 27 27 27 27 27 ...
##  $ region_capital: Factor w/ 28 levels "Ajaccio","Amiens",..: 14 14 14 14 14 14 14 14 14 14 ...
##  $ department    : Factor w/ 102 levels "Ain","Aisne",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ town_name     : Factor w/ 34142 levels "Aast","Abainville",..: 1264 2487 2798 2821 3515 4001 4708
##  $ postal_code   : Factor w/ 6106 levels "01000","01090",..: 26 19 29 26 17 1 23 16 17 17 ...
##  $ CODGEO        : int  1024 1029 1038 1040 1245 1053 1065 1069 1072 1095 ...
##  $ latitude      : num  46.3 46.4 46.3 46.4 46.1 ...
##  $ longitude     : Factor w/ 1151 levels "","-","-0,75",..: 874 880 881 864 891 877 872 880 885 892
```

```
summary(geo)
```

```
##            region       region_capital           department
##  Midi-Pyrénées: 3028   Toulouse: 3028   Pas-de-Calais :  898
##  Rhône-Alpes  : 2890   Lyon    : 2890   Aisne         :  816
##  Lorraine     : 2336   Metz    : 2336   Somme         :  783
##  Aquitaine    : 2300   Bordeaux: 2300   Seine-Maritime:  747
##  Picardie     : 2295   Amiens  : 2295   Moselle       :  732
##  Bourgogne    : 2050   Dijon   : 2050   Côte-d'Or     :  709
##  (Other)      :21941   (Other) :21941   (Other)       :32155
##            town_name      postal_code        CODGEO         latitude
##  Paris         :   21   51300  :   46   Min.   : 1001   Min.   :41.39
##  Sainte-Colombe:   14   51800  :   44   1st Qu.:24577   1st Qu.:45.22
```

```
##  Saint-Sauveur :   12   70000  :   42   Median :48191   Median :47.43
##  Beaulieu      :   11   88500  :   42   Mean   :46298   Mean   :47.00
##  Saint-Sulpice :   11   80140  :   40   3rd Qu.:67043   3rd Qu.:48.85
##  Sainte-Marie  :   11   02160  :   38   Max.   :97617   Max.   :51.08
##  (Other)       :36760   (Other):36588                   NA's   :2929
##     longitude
##          : 2841
##  2.433333:  105
##  2.333333:  100
##  1.833333:   99
##  2.116667:   98
##  2.25    :   94
##  (Other) :33503
```

```
# spot "," instead of "." in longitude
newLong       <- as.character(geo$longitude)    # copy the vector
sum(grep(",", newLong))                         # total commas
```

```
## [1] 994302
```

```
ind_long_err  <- grep(",", newLong)              # indexing them
newLong       <- gsub(",", ".", newLong)        # substituting them with dots
indNA_Long    <- is.na(as.numeric((newLong)))   # spot NA
```

```
## Warning: si è prodotto un NA per coercizione
```

```
geo$longitude[indNA_Long]                       # verify that they were actually missing
```

```
##    [1]
##   [35]
##   [69]
##  [103]
##  [137]
##  [171]
##  [205]
##  [239]
##  [273]
##  [307]
##  [341]
##  [375] -             -   -         -   -
##  [409]
##  [443]               -             -         -         -
##  [477]   -     - -             -       -
##  [511]
##  [545]
##  [579]
##  [613]
##  [647]
##  [681]
##  [715]
##  [749]
##  [783]
##  [817]
##  [851]
##  [885]
##  [919]
```

```
##  [953]
##  [987]                                                 -
## [1021]   -                   -           -
## [1055]                                   -                 - -           -
## [1089] - -         -       -                               -
## [1123]
## [1157]
## [1191]
## [1225]
## [1259]                                               -
## [1293]
## [1327]
## [1361]
## [1395]
## [1429]                                     - - -
## [1463]
## [1497]
## [1531]
## [1565]
## [1599]
## [1633]
## [1667]
## [1701]
## [1735]
## [1769]
## [1803]
## [1837]
## [1871]                                                       -   -
## [1905]     -                             -   -       -           -
## [1939]
## [1973]
## [2007]                                     -
## [2041]                                                         - - -
## [2075] -     - - - -   -           - -   - -   - - -
## [2109]
## [2143]
## [2177]
## [2211]
## [2245]
## [2279]                       - -     -
## [2313]
## [2347]
## [2381]
## [2415]
## [2449]                                                 -           -   -
## [2483]
## [2517]
## [2551]
## [2585]
## [2619]
## [2653]         -                       -                 -
## [2687]
## [2721]
## [2755]
```

```
## [2789]
## [2823]
## [2857]
## [2891]
## 1151 Levels:  - -0,75 -0.008333 -0.016667 -0.01679 -0.025 ... 9.516667
```

```
geo$longitude <- as.numeric(newLong)            # overwrite the longitude variable with the new one
```

```
## Warning: si è prodotto un NA per coercizione
```

```
# Check for duplicated data (e.g., cities with different postal codes, that we dropped):
  # es. to verify it:  try on the initial dataset
  # sum(geo$nom_commune == "Paris")
  # ind_duplic <- geo$nom_commune == "Paris"
  # geo[ind_duplic,]
sum(duplicated.data.frame(geo))
```

```
## [1] 117
```

```
# retaing unique postal cities
geo <- unique(geo, by = "CODGEO")

# check again
head(geo)
```

```
##        region region_capital department            town_name postal_code
## 1 Rhône-Alpes           Lyon        Ain             Attignat       01340
## 2 Rhône-Alpes           Lyon        Ain             Beaupont       01270
## 3 Rhône-Alpes           Lyon        Ain                 Bény       01370
## 4 Rhône-Alpes           Lyon        Ain            Béreyziat       01340
## 5 Rhône-Alpes           Lyon        Ain Bohas-Meyriat-Rignat       01250
## 6 Rhône-Alpes           Lyon        Ain      Bourg-en-Bresse       01000
##   CODGEO latitude longitude
## 1   1024 46.28333  5.166667
## 2   1029 46.40000  5.266667
## 3   1038 46.33333  5.283333
## 4   1040 46.36667  5.050000
## 5   1245 46.13333  5.400000
## 6   1053 46.20000  5.216667
```

```
summary(geo)
```

```
##            region       region_capital           department
##  Midi-Pyrénées: 3021   Toulouse: 3021   Pas-de-Calais :  894
##  Rhône-Alpes  : 2882   Lyon    : 2882   Aisne         :  816
##  Lorraine     : 2333   Metz    : 2333   Somme         :  782
##  Aquitaine    : 2296   Bordeaux: 2296   Seine-Maritime:  745
##  Picardie     : 2291   Amiens  : 2291   Moselle       :  730
##  Bourgogne    : 2046   Dijon   : 2046   Côte-d'Or     :  707
##  (Other)      :21854   (Other) :21854   (Other)       :32049
##           town_name      postal_code        CODGEO         latitude
##  Paris         :   19   51300  :   46   Min.   : 1001   Min.   :41.39
##  Sainte-Colombe:   14   51800  :   44   1st Qu.:24562   1st Qu.:45.22
##  Saint-Sauveur :   12   70000  :   42   Median :48180   Median :47.43
##  Beaulieu      :   11   88500  :   42   Mean   :46276   Mean   :47.00
##  Saint-Sulpice :   11   80140  :   40   3rd Qu.:67037   3rd Qu.:48.85
##  Sainte-Marie  :   11   02160  :   38   Max.   :97617   Max.   :51.08
```

```
##  (Other)       :36645   (Other):36471                   NA's   :2925
##    longitude
##  Min.   :-5.1000
##  1st Qu.: 0.6833
##  Median : 2.6167
##  Mean   : 2.7329
##  3rd Qu.: 4.8500
##  Max.   : 9.5167
##  NA's   :2903
```

Assign lat and long values for NAs units:

```
require(ggmap)
```

```
## Loading required package: ggmap
```

```
# [ delete? ]
# # compare numbers of NA in latitude and longitude
# sum(is.na(geo$latitude)) - sum(is.na(geo$longitude))
# # check if they match or not
# sum(!is.na(geo$latitude[indNA_Long]))
# # 64 obs are missing in longitude but not in latitude, hence 88 vice versa

# index of NAs and their total
indNA_coord = is.na(geo$latitude) | is.na(geo$longitude)
sum(indNA_coord)
```

```
## [1] 2989
```

```
# NO MORE NEEDED BECAUSE IS A CSV FILE
    # # initialize variables
    # city_search = 0
    # res = as.data.frame(matrix(c(0, 0, 0), 1, 3))
    # names(res) = c("lon", "lat", "address")
    #
    # # retrieve lat and long (Google API = 2500 request per day)
    # # my_iter = floor(sum(indNA_coord)/3)
    # for (i in 1:sum(indNA_coord)){
    #
    #   # city searched
    #   city_search[i] = paste(c(as.character(NA_coord$town_name[i]), as.character(NA_coord$postal_code
    #
    #   # solution
    #   res[i,] = geocode(city_search[i], output = "latlona", source = c("google", "dsk"), messaging = 
    #
    #   # retrieve still missing data, because of existing problems with API (up to 15 trials)
    #   j = 0
    #   while (any(is.na(res[i,])) & j < 25){
    #     res[i,] = geocode(city_search[i], output = "latlona", source = c("google", "dsk"), messaging 
    #     j = j + 1
    #   }
    # }

    # # check the solution
    # sol = cbind(searched = city_search, res)
```

```
    # # save it as a csv file to save time
    # write.csv(retrieved_geo_NA[,2:3], "geo_NA_Final.csv", quote = FALSE, row.names=FALSE, fileEncodin


# read the created csv
retrieved_geo_NA = read.csv("geo_NA_Final.csv", header = T, encoding = "UTF-8")
# get only long and lat and assign to original NA
geo$latitude[indNA_coord] = retrieved_geo_NA[,2]
geo$longitude[indNA_coord] = retrieved_geo_NA[,1]

# there are 37 still missing units, which are towns located in old colonies far from Europe
indNA_coord = is.na(geo$latitude) | is.na(geo$longitude)
sum(indNA_coord)
```

```
## [1] 37
```

```
# exclude those towns
geo = geo[!indNA_coord,]

summary(geo)
```

```
##            region       region_capital           department
##  Midi-Pyrénées: 3021   Toulouse: 3021   Pas-de-Calais :  894
##  Rhône-Alpes  : 2882   Lyon    : 2882   Aisne         :  816
##  Lorraine     : 2333   Metz    : 2333   Somme         :  782
##  Aquitaine    : 2296   Bordeaux: 2296   Seine-Maritime:  745
##  Picardie     : 2291   Amiens  : 2291   Moselle       :  730
##  Bourgogne    : 2046   Dijon   : 2046   Côte-d'Or     :  707
##  (Other)      :21817   (Other) :21817   (Other)       :32012
##           town_name      postal_code        CODGEO         latitude
##  Paris         :   19   51300  :   46   Min.   : 1001   Min.   :-21.38
##  Sainte-Colombe:   14   51800  :   44   1st Qu.:24551   1st Qu.: 45.15
##  Saint-Sauveur :   12   70000  :   42   Median :48162   Median : 47.38
##  Beaulieu      :   11   88500  :   42   Mean   :46224   Mean   : 46.87
##  Saint-Sulpice :   11   80140  :   40   3rd Qu.:67006   3rd Qu.: 48.83
##  Sainte-Marie  :   11   02160  :   38   Max.   :97611   Max.   : 51.08
##  (Other)       :36608   (Other):36434
##    longitude
##  Min.   :-63.0885
##  1st Qu.:  0.6667
##  Median :  2.6333
##  Mean   :  2.6617
##  3rd Qu.:  4.8667
##  Max.   : 55.8250
##
```

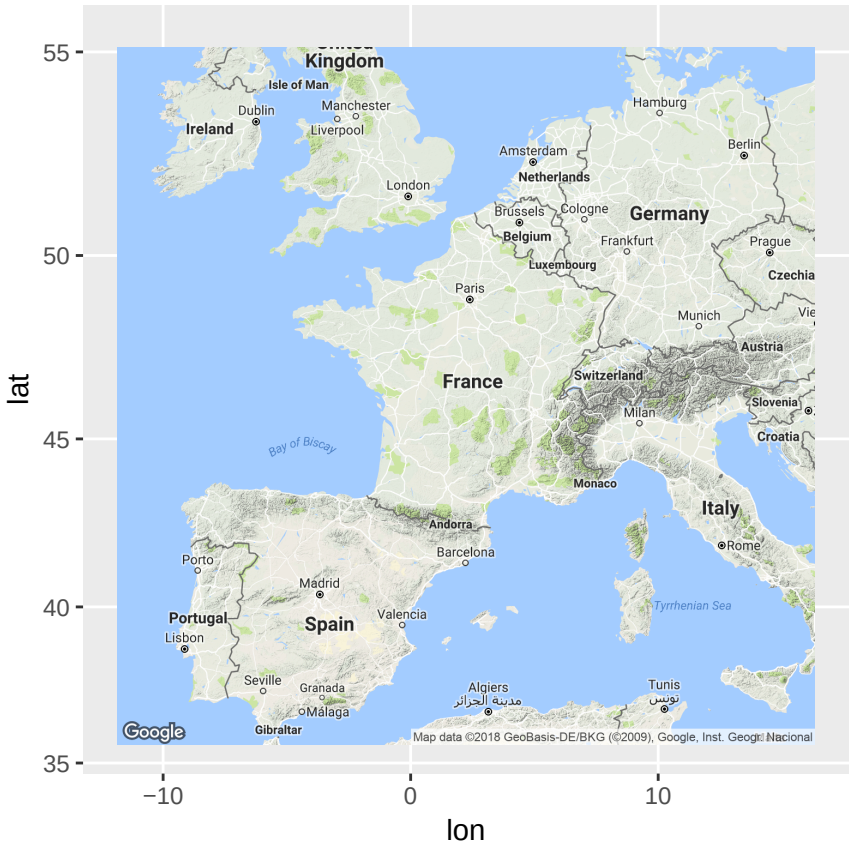
EDA

```
#install.packages("ggplot2")
#install.packages("ggmap")
require(ggplot2)
require(ggmap)
```

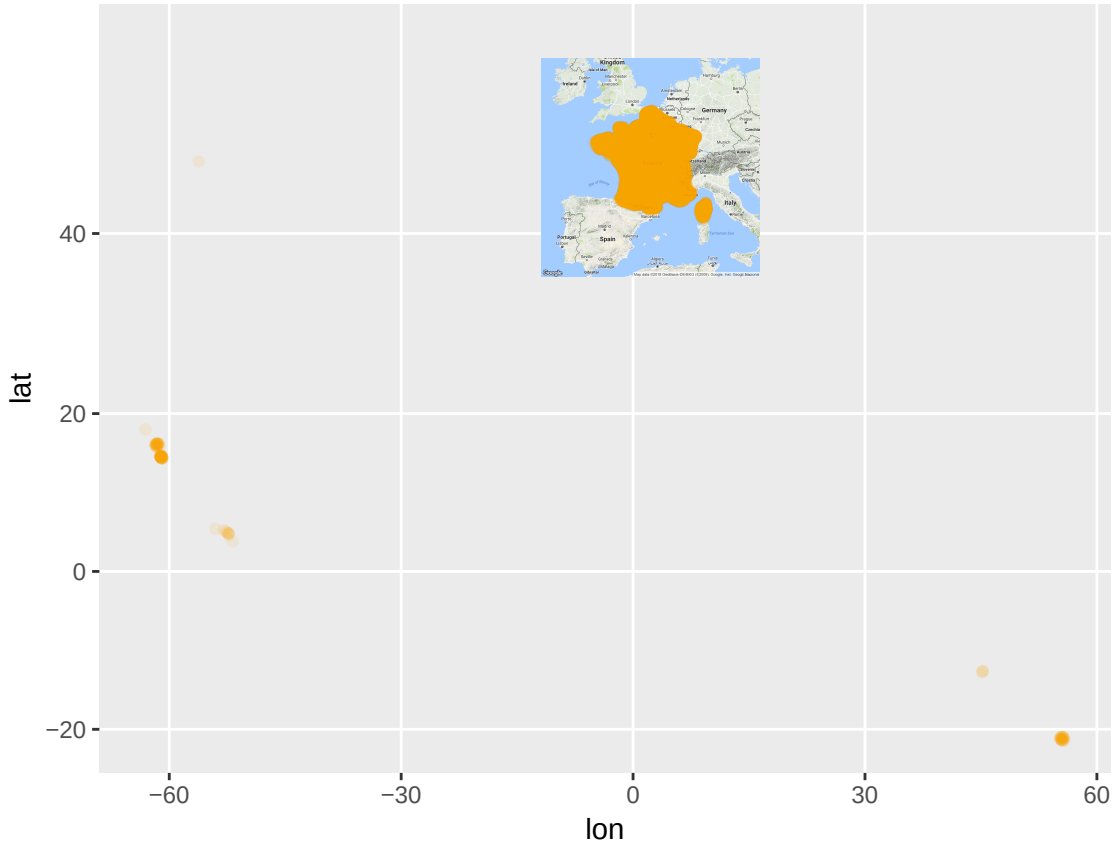
```r
# plot france (center: 2.213749 46.227638)
# fra_center = as.numeric(geocode("France"))
fra_center = c(2.213749, 46.227638)
FraMap = ggmap(get_googlemap(center=fra_center, scale=2, zoom=5), extent="normal")
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=46.227638,2.213749&zoom=5&size=6
```

```r
FraMap
```



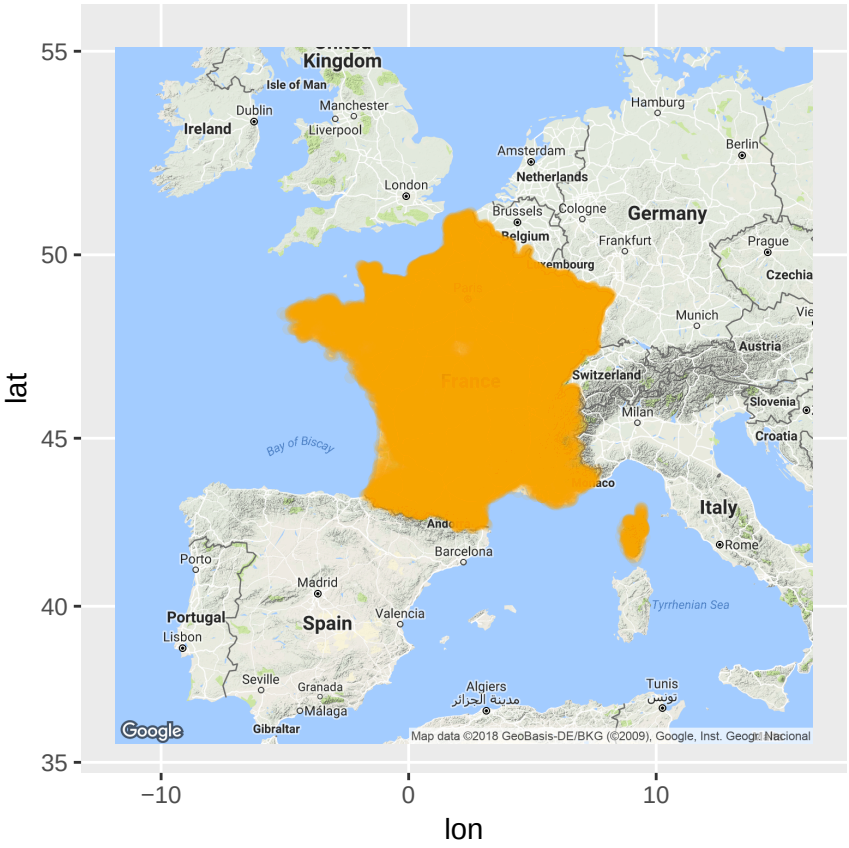

```r
# plot all towns available
geo_pos = as.data.frame(cbind(lon = geo$longitude, lat = geo$latitude))
geo_pos = geo_pos[complete.cases(geo_pos),]
FraMap +
  geom_point(aes(x=lon, y=lat), data=geo_pos, col="orange", alpha=0.1)
```




```
# delete non-European countries
ind_nonEur = geo$latitude < 30 | geo$latitude > 70 |geo$longitude < -20 | geo$longitude > 20
sum(ind_nonEur)
```

```
## [1] 92
```

```
geo = geo[!ind_nonEur,]

# plot all European towns available
geo_pos = as.data.frame(cbind(lon = geo$longitude, lat = geo$latitude))
geo_pos = geo_pos[complete.cases(geo_pos),]
FraMap +
  geom_point(aes(x=lon, y=lat), data=geo_pos, col="orange", alpha=0.1)
```



What we have learned

Solved:

- Why latitude is missing and not longitude?
- There are some duplications.

To do:

- What to do with non-European towns?

How to use these data

- Compare European towns vs. old colonies?
- Useful for all datasets/analyses

Analyze salary data

Pre-processing

```
# preliminary checks
dim(salary)
```

```
## [1] 5136   26
```

```
names(salary)
```

```
##  [1] "CODGEO"           "town"             "sal_general"
##  [4] "sal_executive"    "sal_midManager"   "sal_employee"
##  [7] "sal_worker"       "sal_Females"      "sal_F_executive"
## [10] "sal_F_midManager" "sal_F_employee"   "sal_F_worker"
## [13] "sal_Males"        "sal_M_executive"  "sal_M_midManager"
## [16] "sal_M_employee"   "sal_M_worker"     "sal_18_25"
## [19] "sal_26_50"        "sal_51plus"       "sal_F_18_25"
## [22] "sal_F_26_50"      "sal_F_51plus"     "sal_M_18_25"
## [25] "sal_M_26_50"      "sal_M_51plus"
```

```
head(salary)
```

```
##   CODGEO              town sal_general sal_executive sal_midManager
## 1  01004 Ambérieu-en-Bugey        13.7          24.2           15.5
## 2  01007          Ambronay        13.5          22.1           14.7
## 3  01014            Arbent        13.5          27.6           15.6
## 4  01024          Attignat        12.9          21.8           14.1
## 5  01025     Bâgé-la-Ville        13.0          22.8           14.1
## 6  01027             Balan        13.9          22.2           15.1
##   sal_employee sal_worker sal_Females sal_F_executive sal_F_midManager
## 1         10.3       11.2        11.6            19.1             13.2
## 2         10.7       11.4        11.9            19.0             13.3
## 3         11.1       11.1        10.9            19.5             11.7
## 4         11.0       11.3        11.4            19.0             13.0
## 5         10.5       11.1        11.6            19.4             13.6
## 6         11.0       11.4        12.5            20.3             14.0
##   sal_F_employee sal_F_worker sal_Males sal_M_executive sal_M_midManager
## 1           10.1          9.6      15.0            26.4             16.7
## 2           10.6         10.0      14.7            23.3             15.8
## 3           10.8          9.5      15.3            30.2             17.2
## 4           10.3          9.9      13.8            23.0             14.7
## 5           10.2          9.8      13.8            24.1             14.4
## 6           10.9         10.5      15.2            23.1             15.9
##   sal_M_employee sal_M_worker sal_18_25 sal_26_50 sal_51plus sal_F_18_25
## 1           11.0         11.6      10.5      13.7       16.1         9.7
## 2           11.3         11.7       9.8      13.8       14.6         9.2
## 3           12.4         11.8       9.3      13.3       16.0         8.9
## 4           13.2         11.6       9.6      12.9       14.2         9.3
## 5           11.7         11.4       9.4      12.8       15.2         9.0
## 6           12.1         11.7       9.7      14.1       15.4         9.5
##   sal_F_26_50 sal_F_51plus sal_M_18_25 sal_M_26_50 sal_M_51plus
## 1        11.8         12.5        11.0        14.9         18.6
## 2        12.2         12.5        10.2        14.9         16.4
## 3        10.6         12.5         9.6        15.1         18.6
## 4        11.4         12.2         9.7        13.8         15.9
## 5        11.8         12.3         9.7        13.4         16.9
## 6        12.8         13.0         9.9        15.3         17.2
```

```
str(salary)
```

```
## 'data.frame':    5136 obs. of  26 variables:
##  $ CODGEO          : chr  "01004" "01007" "01014" "01024" ...
```

```
##  $ town            : Factor w/ 5085 levels "Abbeville","Ablis",..: 73 79 133 192 275 298 407 395 400
##  $ sal_general     : num  13.7 13.5 13.5 12.9 13 13.9 12.4 14 11.5 12.4 ...
##  $ sal_executive   : num  24.2 22.1 27.6 21.8 22.8 22.2 24 23.1 21.2 23.4 ...
##  $ sal_midManager  : num  15.5 14.7 15.6 14.1 14.1 15.1 13.1 15.3 13.5 14.1 ...
##  $ sal_employee    : num  10.3 10.7 11.1 11 10.5 11 10.5 10.9 9.9 10.3 ...
##  $ sal_worker      : num  11.2 11.4 11.1 11.3 11.1 11.4 10.4 11.3 10.5 10.5 ...
##  $ sal_Females     : num  11.6 11.9 10.9 11.4 11.6 12.5 10.9 12.4 10.3 11 ...
##  $ sal_F_executive : num  19.1 19 19.5 19 19.4 20.3 20.7 20.5 20.8 21.5 ...
##  $ sal_F_midManager: num  13.2 13.3 11.7 13 13.6 14 11.8 13.9 12.3 13 ...
##  $ sal_F_employee  : num  10.1 10.6 10.8 10.3 10.2 10.9 10.4 10.7 9.8 9.9 ...
##  $ sal_F_worker    : num  9.6 10 9.5 9.9 9.8 10.5 9.3 10.3 9 9.5 ...
##  $ sal_Males       : num  15 14.7 15.3 13.8 13.8 15.2 13.4 15.4 12.3 13.2 ...
##  $ sal_M_executive : num  26.4 23.3 30.2 23 24.1 23.1 25.2 24.4 21.3 24 ...
##  $ sal_M_midManager: num  16.7 15.8 17.2 14.7 14.4 15.9 13.8 16.3 14.2 14.9 ...
##  $ sal_M_employee  : num  11 11.3 12.4 13.2 11.7 12.1 10.8 11.8 10.5 11.6 ...
##  $ sal_M_worker    : num  11.6 11.7 11.8 11.6 11.4 11.7 10.8 11.6 11 10.9 ...
##  $ sal_18_25       : num  10.5 9.8 9.3 9.6 9.4 9.7 9.3 9.7 9.6 9.7 ...
##  $ sal_26_50       : num  13.7 13.8 13.3 12.9 12.8 14.1 12.5 13.9 11.5 12.3 ...
##  $ sal_51plus      : num  16.1 14.6 16 14.2 15.2 15.4 13.3 16.7 12.7 13.7 ...
##  $ sal_F_18_25     : num  9.7 9.2 8.9 9.3 9 9.5 8.9 9.7 9.2 9.3 ...
##  $ sal_F_26_50     : num  11.8 12.2 10.6 11.4 11.8 12.8 11 12.4 10.3 11.2 ...
##  $ sal_F_51plus    : num  12.5 12.5 12.5 12.2 12.3 13 11.5 13.8 11.3 11.4 ...
##  $ sal_M_18_25     : num  11 10.2 9.6 9.7 9.7 9.9 9.6 9.6 10 9.9 ...
##  $ sal_M_26_50     : num  14.9 14.9 15.1 13.8 13.4 15.3 13.3 15 12.3 13 ...
##  $ sal_M_51plus    : num  18.6 16.4 18.6 15.9 16.9 17.2 14.9 19.3 13.9 15.4 ...
```

```
summary(salary)
```

```
##     CODGEO                    town       sal_general    sal_executive
##  Length:5136        Sainte-Marie:   4   Min.   :10.20   Min.   :16.0
##  Class :character   Saint-Ouen  :   3   1st Qu.:12.10   1st Qu.:21.9
##  Mode  :character   Allonnes    :   2   Median :13.00   Median :23.2
##                     Andilly     :   2   Mean   :13.71   Mean   :23.7
##                     Bassens     :   2   3rd Qu.:14.40   3rd Qu.:24.9
##                     Beaumont    :   2   Max.   :43.30   Max.   :51.5
##                     (Other)     :5121
##  sal_midManager   sal_employee     sal_worker     sal_Females
##  Min.   :11.60   Min.   : 8.70   Min.   : 8.30   Min.   : 9.30
##  1st Qu.:13.80   1st Qu.:10.00   1st Qu.:10.60   1st Qu.:10.90
##  Median :14.40   Median :10.40   Median :11.00   Median :11.50
##  Mean   :14.58   Mean   :10.56   Mean   :11.24   Mean   :12.04
##  3rd Qu.:15.10   3rd Qu.:10.90   3rd Qu.:11.60   3rd Qu.:12.70
##  Max.   :54.60   Max.   :17.50   Max.   :46.30   Max.   :26.70
##
##  sal_F_executive sal_F_midManager sal_F_employee   sal_F_worker
##  Min.   :12.00   Min.   :10.60    Min.   : 8.70   Min.   : 6.100
##  1st Qu.:18.80   1st Qu.:12.60    1st Qu.: 9.80   1st Qu.: 9.200
##  Median :20.00   Median :13.10    Median :10.10   Median : 9.700
##  Mean   :20.22   Mean   :13.27    Mean   :10.31   Mean   : 9.827
##  3rd Qu.:21.40   3rd Qu.:13.80    3rd Qu.:10.60   3rd Qu.:10.200
##  Max.   :35.50   Max.   :19.00    Max.   :16.10   Max.   :28.100
##
##    sal_Males     sal_M_executive sal_M_midManager sal_M_employee
##  Min.   :10.40   Min.   :13.8    Min.   :11.80    Min.   : 8.00
##  1st Qu.:12.90   1st Qu.:23.1    1st Qu.:14.50    1st Qu.:10.50
```

```
##  Median :14.10   Median :24.6    Median :15.20    Median :11.10
##  Mean   :14.85   Mean   :25.2    Mean   :15.49    Mean   :11.27
##  3rd Qu.:15.80   3rd Qu.:26.6    3rd Qu.:16.00    3rd Qu.:11.80
##  Max.   :52.40   Max.   :58.0    Max.   :93.40    Max.   :23.50
##
##   sal_M_worker    sal_18_25       sal_26_50      sal_51plus
##  Min.   : 8.9   Min.   : 7.90   Min.   : 9.7   Min.   :10.50
##  1st Qu.:10.8   1st Qu.: 9.20   1st Qu.:12.0   1st Qu.:13.70
##  Median :11.3   Median : 9.50   Median :12.9   Median :15.00
##  Mean   :11.5   Mean   : 9.55   Mean   :13.5   Mean   :15.88
##  3rd Qu.:11.9   3rd Qu.: 9.70   3rd Qu.:14.3   3rd Qu.:16.90
##  Max.   :53.2   Max.   :60.60   Max.   :38.1   Max.   :56.90
##
##   sal_F_18_25      sal_F_26_50     sal_F_51plus    sal_M_18_25
##  Min.   : 7.500   Min.   : 9.10   Min.   : 9.50   Min.   : 7.800
##  1st Qu.: 8.900   1st Qu.:10.90   1st Qu.:11.70   1st Qu.: 9.400
##  Median : 9.100   Median :11.60   Median :12.60   Median : 9.700
##  Mean   : 9.162   Mean   :12.06   Mean   :13.17   Mean   : 9.821
##  3rd Qu.: 9.400   3rd Qu.:12.70   3rd Qu.:14.00   3rd Qu.:10.000
##  Max.   :12.000   Max.   :26.60   Max.   :31.00   Max.   :93.300
##
##   sal_M_26_50     sal_M_51plus
##  Min.   : 9.60   Min.   :10.80
##  1st Qu.:12.70   1st Qu.:14.90
##  Median :13.80   Median :16.60
##  Mean   :14.49   Mean   :17.68
##  3rd Qu.:15.50   3rd Qu.:19.00
##  Max.   :45.40   Max.   :68.60
##
```

```
# Drop unnecessary columns (town name repeats in other table, is it surely possible to merge them?)
names(salary)
```

```
##  [1] "CODGEO"           "town"             "sal_general"
##  [4] "sal_executive"    "sal_midManager"   "sal_employee"
##  [7] "sal_worker"       "sal_Females"      "sal_F_executive"
## [10] "sal_F_midManager" "sal_F_employee"   "sal_F_worker"
## [13] "sal_Males"        "sal_M_executive"  "sal_M_midManager"
## [16] "sal_M_employee"   "sal_M_worker"     "sal_18_25"
## [19] "sal_26_50"        "sal_51plus"       "sal_F_18_25"
## [22] "sal_F_26_50"      "sal_F_51plus"     "sal_M_18_25"
## [25] "sal_M_26_50"      "sal_M_51plus"
```

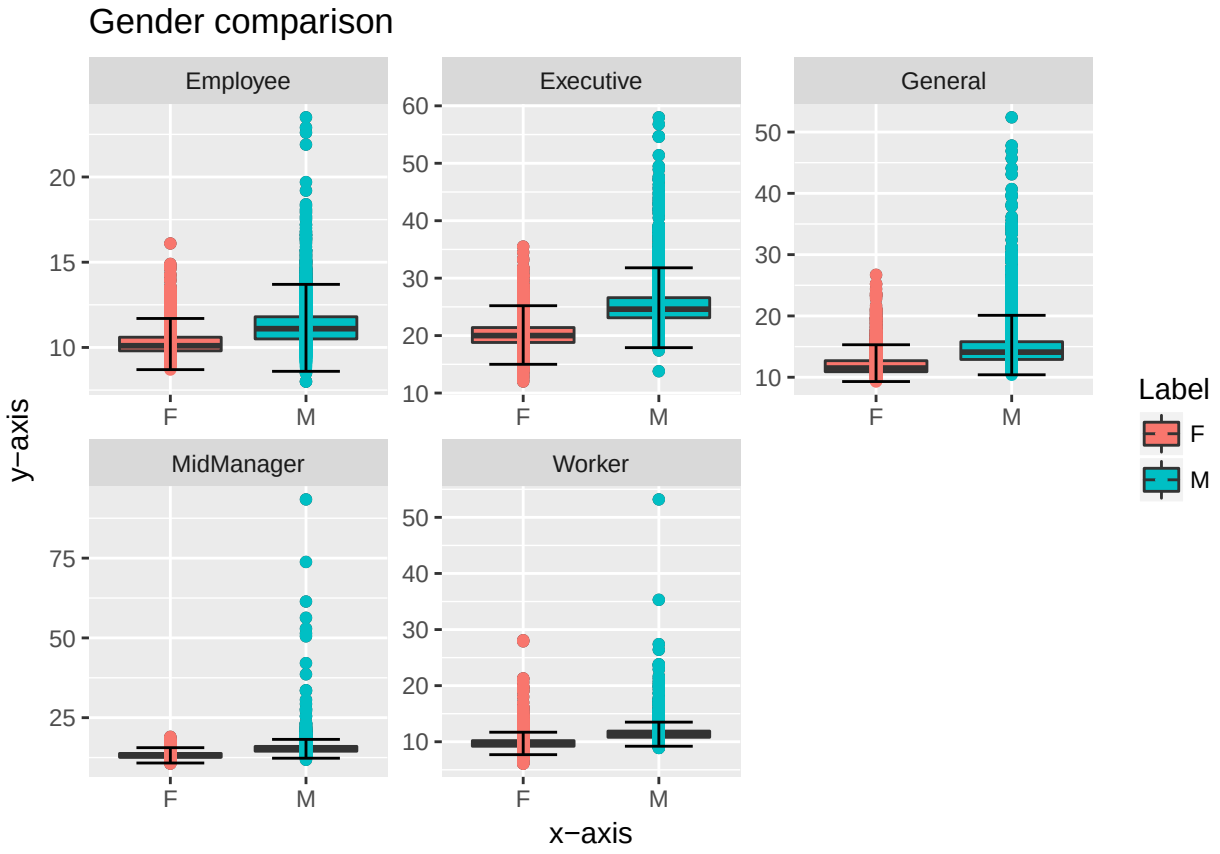
```
# salary <- subset(salary, select = -c(town))

# Convert CODGEO to numeric
salary$CODGEO <- as.numeric(as.character(salary$CODGEO))

# Check for duplicated data
sum(duplicated.data.frame(salary))
```

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

EDA

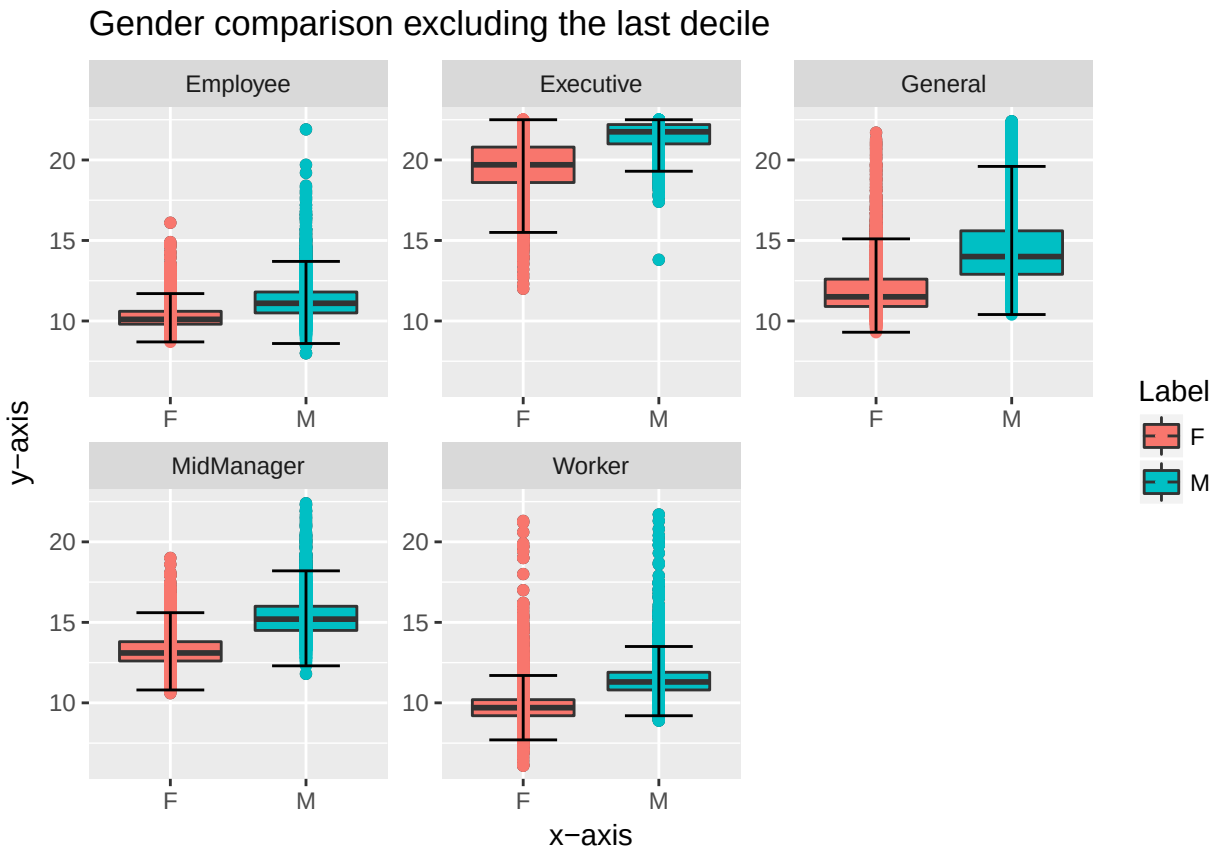
Univariate analysis comparing various job categories for both genders:

```
require(ggplot2)

#  number of units
n_sex <- length(salary$sal_Females)

# vector representing males and females
Label <- c(rep("M", n_sex*5), rep("F", n_sex*5))

# vector representing the variable considered
Variable <- c(rep("General", n_sex),
              rep("Executive", n_sex),
              rep("MidManager", n_sex),
              rep("Employee", n_sex),
              rep("Worker",n_sex),
              rep("General", n_sex),
              rep("Executive", n_sex),
              rep("MidManager", n_sex),
              rep("Employee", n_sex),
              rep("Worker",n_sex))

# merge these data
sal_sex = cbind.data.frame(Label = Label,
              value = c(salary$sal_Males, salary$sal_M_executive, salary$sal_M_midManager, salary$sal_M_
                        salary$sal_Females, salary$sal_F_executive, salary$sal_F_midManager, salary$sal_
              Variable = Variable)

# plotting phase
p <-  ggplot(data = sal_sex, aes(x=Label, y=value)) +
      geom_boxplot(aes(fill = Label)) +
      # not color points replacing colour = group instead of colour=Label
      geom_point(aes(y=value, colour=Label), position = position_dodge(width=0.75)) +
      facet_wrap( ~ Variable, scales="free") +
      xlab("x-axis") + ylab("y-axis") + ggtitle("Gender comparison") +
      stat_boxplot(geom = "errorbar", width = 0.5)
      # p <- p + guides(fill=guide_legend(title="Legend"))
p
```



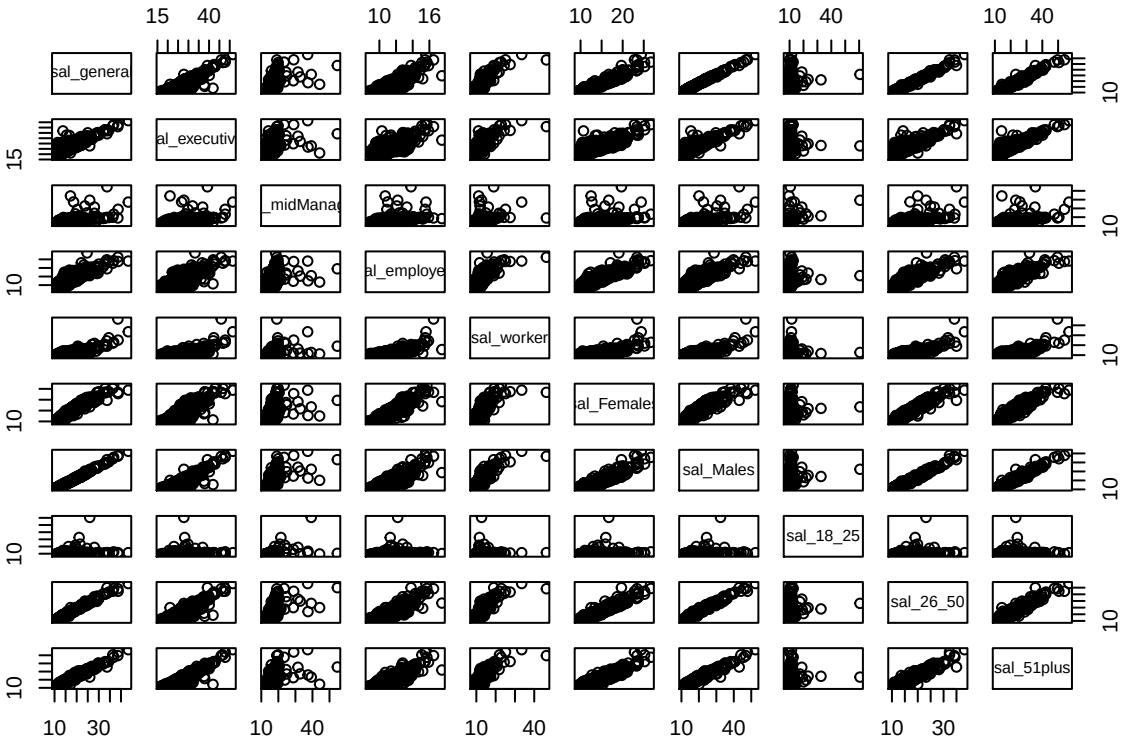

```
# excluding outliers
p2 <- ggplot(data = sal_sex, aes(x=Label, y=value)) +
      scale_y_continuous(limits = quantile(sal_sex$value, c(0, 0.9))) +
      geom_boxplot(aes(fill = Label)) +
      # not color points replacing colour = group instead of colour=Label
      geom_point(aes(y=value, colour=Label), position = position_dodge(width=0.75)) +
      facet_wrap( ~ Variable, scales="free") +
      xlab("x-axis") + ylab("y-axis") + ggtitle("Gender comparison excluding the last decile") +
      # p <- p + guides(fill=guide_legend(title="Legend"))
      stat_boxplot(geom = "errorbar", width = 0.5)
p2
```

```
## Warning: Removed 5067 rows containing non-finite values (stat_boxplot).

## Warning: Removed 5067 rows containing non-finite values (stat_boxplot).

## Warning: Removed 5067 rows containing missing values (geom_point).
```

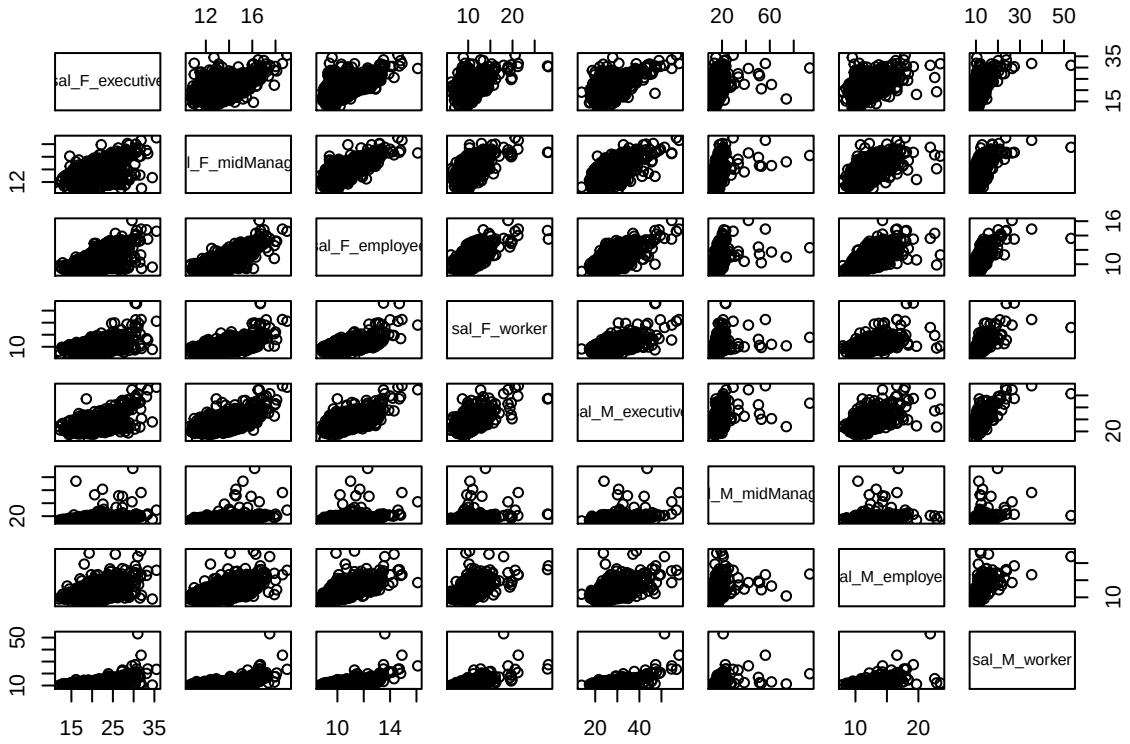


Highlight bivariate relations using scatter matrices:

```
# most general pairs
pairs(salary[c(3:8, 13, 18:20)])
```




```
# pairs highlighting genders' differences
pairs(salary[c(9:12, 14:17)])
```



Fit a regression model to predict the salaries of people in age 26-50 using as regressor 51+ years:

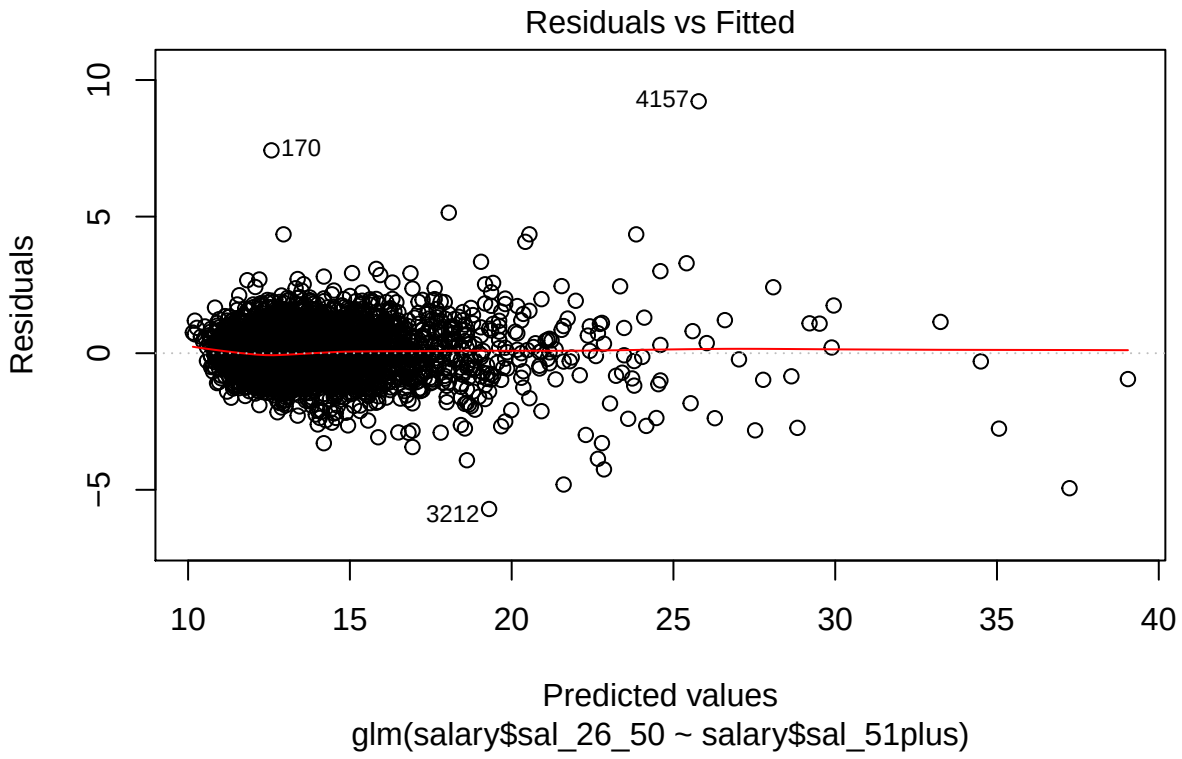
```
# fit and show OLS estimate
plot(salary$sal_26_50 ~ salary$sal_51plus)
```

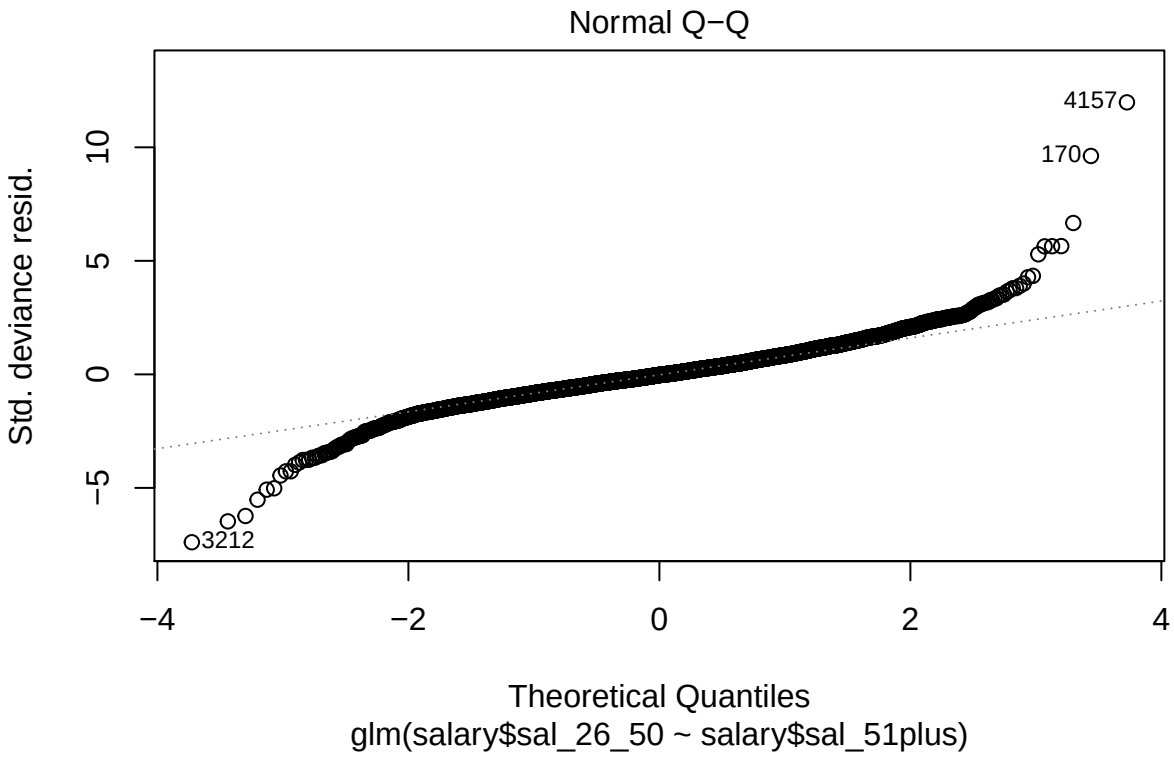

```
fit_LM_26_50 = glm(salary$sal_26_50 ~ salary$sal_51plus, data = salary)
abline(fit_LM_26_50, lwd=3, col="red")
```

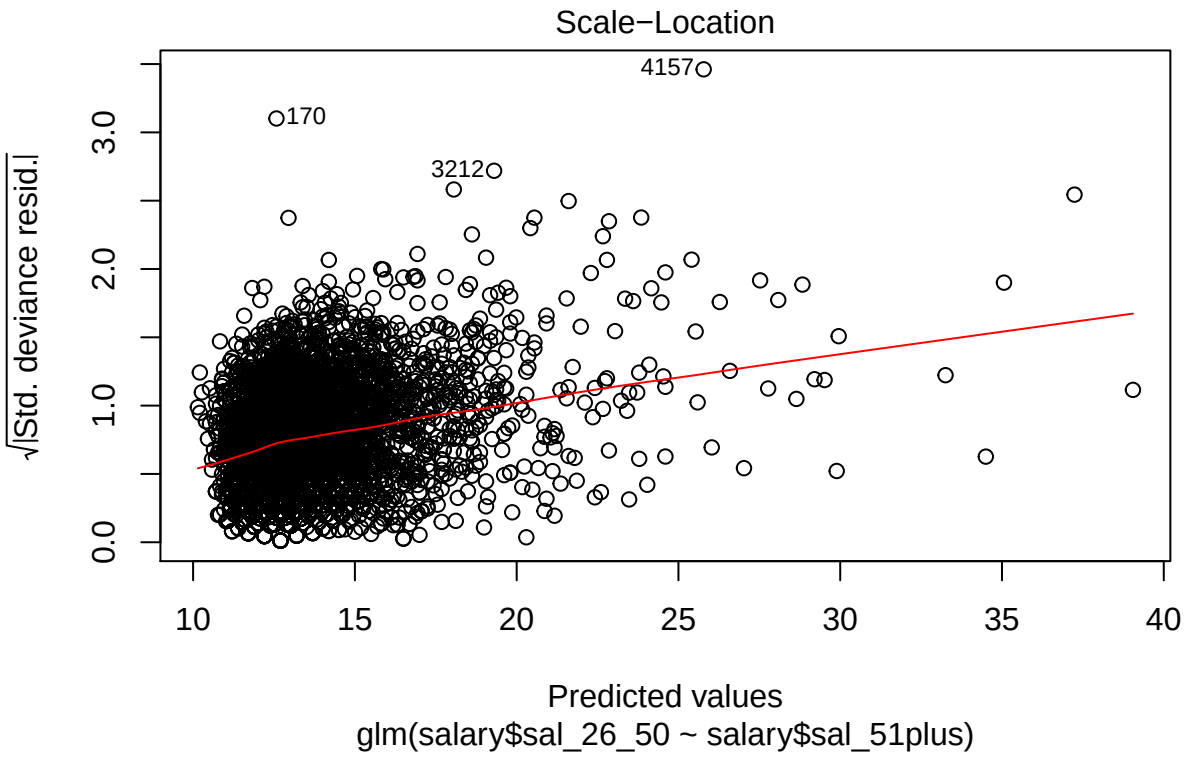

```
# diagnostics
summary(fit_LM_26_50)
```

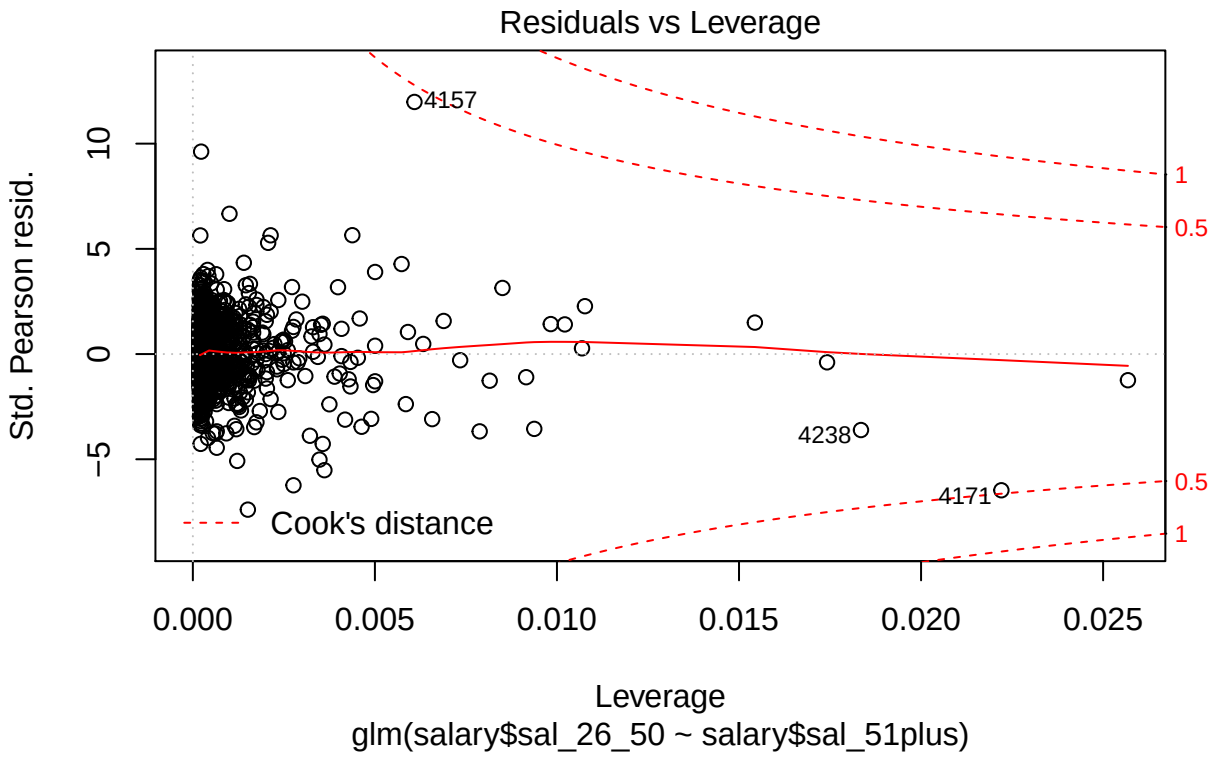
```
##
## Call:
## glm(formula = salary$sal_26_50 ~ salary$sal_51plus, data = salary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.7024  -0.4395  -0.0228   0.4069   9.2197
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.605893   0.048896   73.75   <2e-16 ***
## salary$sal_51plus 0.622877   0.003004  207.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.595825)
##
##     Null deviance: 28676  on 5135  degrees of freedom
## Residual deviance:  3059  on 5134  degrees of freedom
## AIC: 11920
##
## Number of Fisher Scoring iterations: 2
```

```
plot(fit_LM_26_50)
```




Normal Q−Q
Std. deviance resid.
10
5
0
−5
4157
170
3212
−4
−2
0
2
4
Theoretical Quantiles
glm(salary$sal_26_50 ~ salary$sal_51plus)


Scale−Location
4157
170
3212
√|Std. deviance resid.|
0.0
1.0
2.0
3.0
10
15
20
25
30
35
40
Predicted values
glm(salary$sal_26_50 ~ salary$sal_51plus)

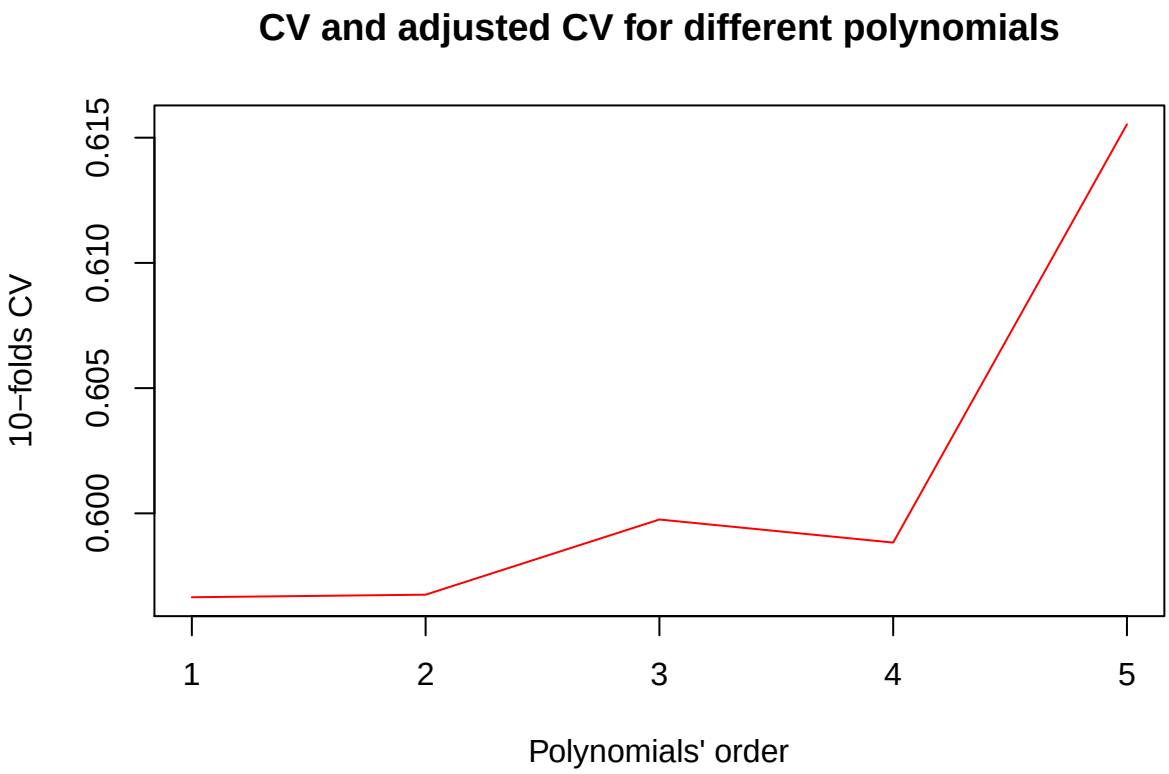


Same as before but adding polynomials which are evauated using 10-folds cross validation:

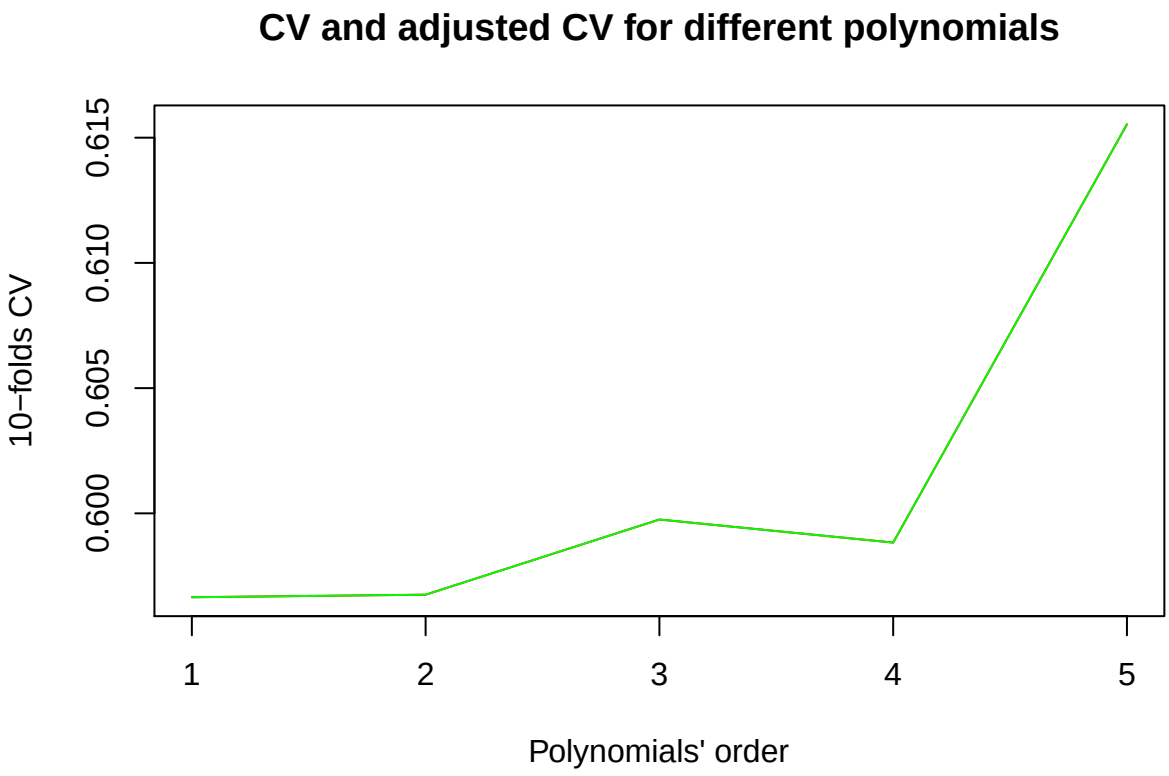
```
require(boot)
```

```
## Loading required package: boot
```

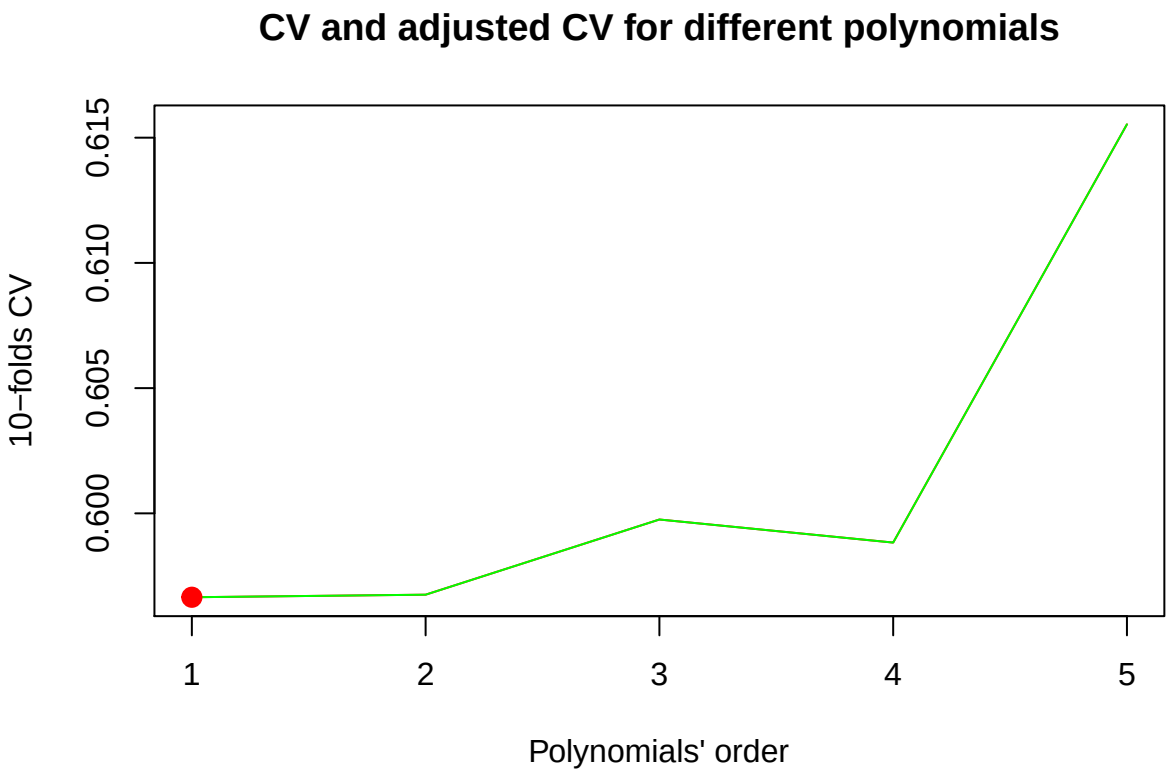
```
set.seed(1)

# k-Fold Cross-Validation
cv.err.K = rep(0, 5)
cv.err.K = rbind(cv.err.K, cv.err.K)
for (i in 1:5){
  fit_LM_26_50.K = glm(sal_26_50 ~ poly(sal_51plus, i), data = salary)
  cv.err.K[,i] = cv.glm(salary, fit_LM_26_50.K, K = 10)$delta[1]
}

# plotting results
plot(cv.err.K[1,], type = 'l', col = 'red', xlab = "Polynomials' order",
     ylab = "10-folds CV", main = "CV and adjusted CV for different polynomials")
```



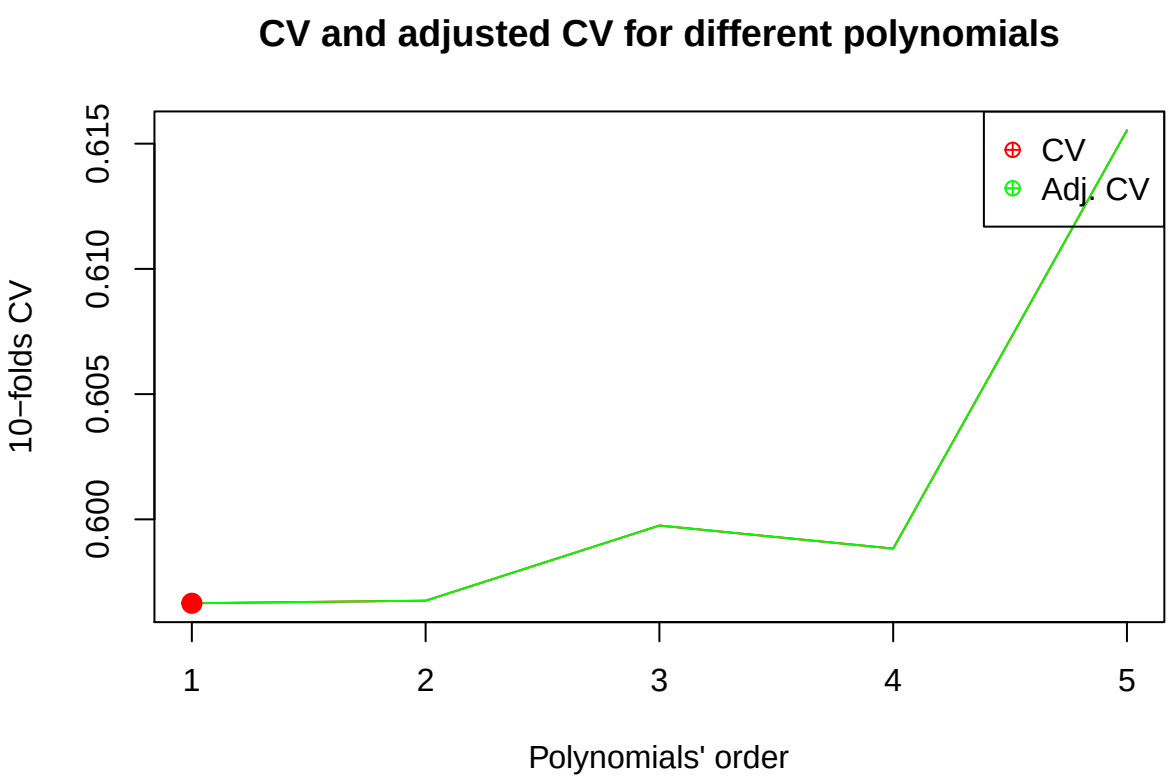

```
lines(cv.err.K[2,], col = 'green')
```




```
points(which.min(cv.err.K), cv.err.K[1, which.min(cv.err.K)], col = "red", cex=2, pch=20)
```




```
legend('topright', legend = c('CV', 'Adj. CV'), col = c('red', 'green'), pch = 10)
```



Predicting sal_executive with 5 regressers using Lasso with CV and 10-folds CV: [PROBLEM FOR COLLINEARITY? use ridge?] [add polynomials?]

```
require(glmnet)
```

```
## Loading required package: glmnet
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-13
```

```
set.seed(1)

# crate an X matrix excluding intercept
# x = cbind(salary$sal_midManager, salary$sal_employee, salary$sal_worker, salary$sal_18_25, salary$sal
# y = salary$sal_executive
# probably for ridge
names(salary)
```

```
##  [1] "CODGEO"           "town"             "sal_general"
##  [4] "sal_executive"    "sal_midManager"   "sal_employee"
##  [7] "sal_worker"       "sal_Females"      "sal_F_executive"
## [10] "sal_F_midManager" "sal_F_employee"   "sal_F_worker"
## [13] "sal_Males"        "sal_M_executive"  "sal_M_midManager"
## [16] "sal_M_employee"   "sal_M_worker"     "sal_18_25"
## [19] "sal_26_50"        "sal_51plus"       "sal_F_18_25"
## [22] "sal_F_26_50"      "sal_F_51plus"     "sal_M_18_25"
```

```
## [25] "sal_M_26_50"      "sal_M_51plus"
```

```
y = salary$sal_26_50
x = as.matrix(cbind(salary[, c(4:8, 13, 18, 20)]))
names(x)
```

```
## NULL
```

```
# grid for lambda values
grid = 10^seq(1, -5, length = 100)
lasso.mod = glmnet(x, y, alpha = 1, lambda = grid)
dim(coef(lasso.mod))
```

```
## [1]   9 100
```

```
# the norms are increasing in value because of the shrinkage
lasso.mod$lambda[1]                    # lambda value
```

```
## [1] 10
```

```
sqrt(sum(coef(lasso.mod)[-1,1]^2))    # L2 norm of its coeff
```

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

```
lasso.mod$lambda[51]
```

```
## [1] 0.009326033
```

```
sqrt(sum(coef(lasso.mod)[-1,51]^2))
```

```
## [1] 0.8159349
```

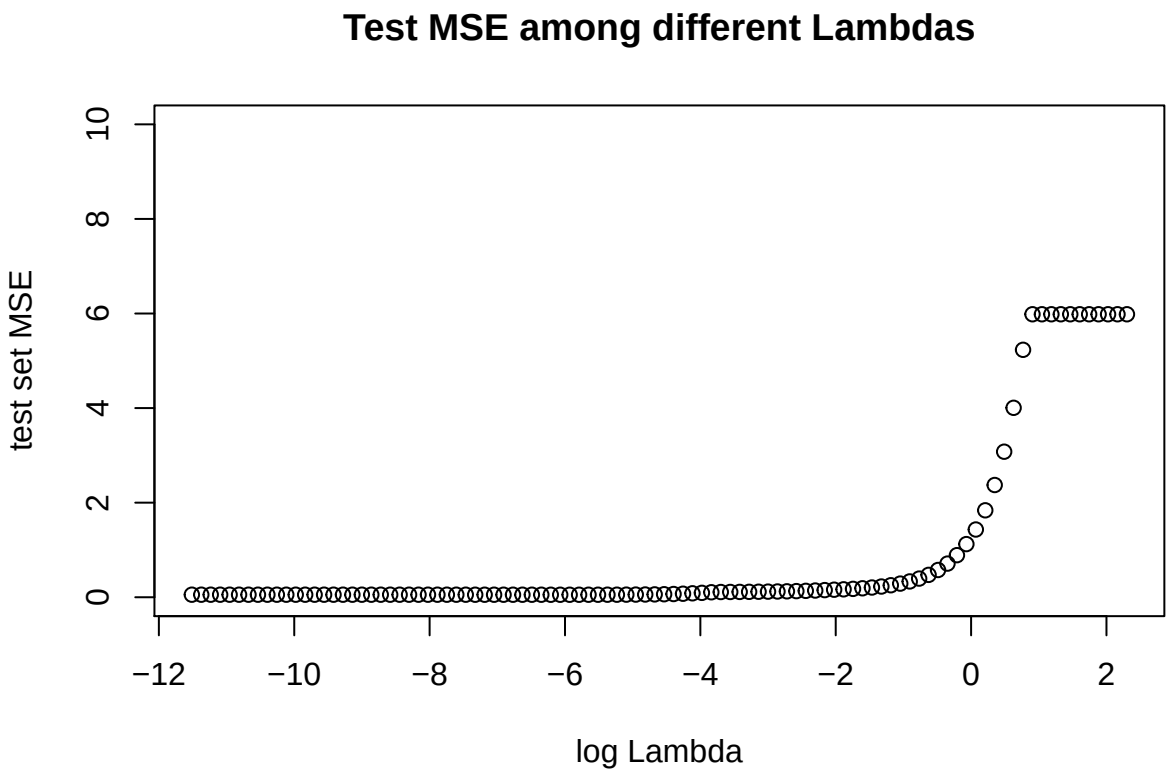
```
lasso.mod$lambda[90]
```

```
## [1] 4.037017e-05
```

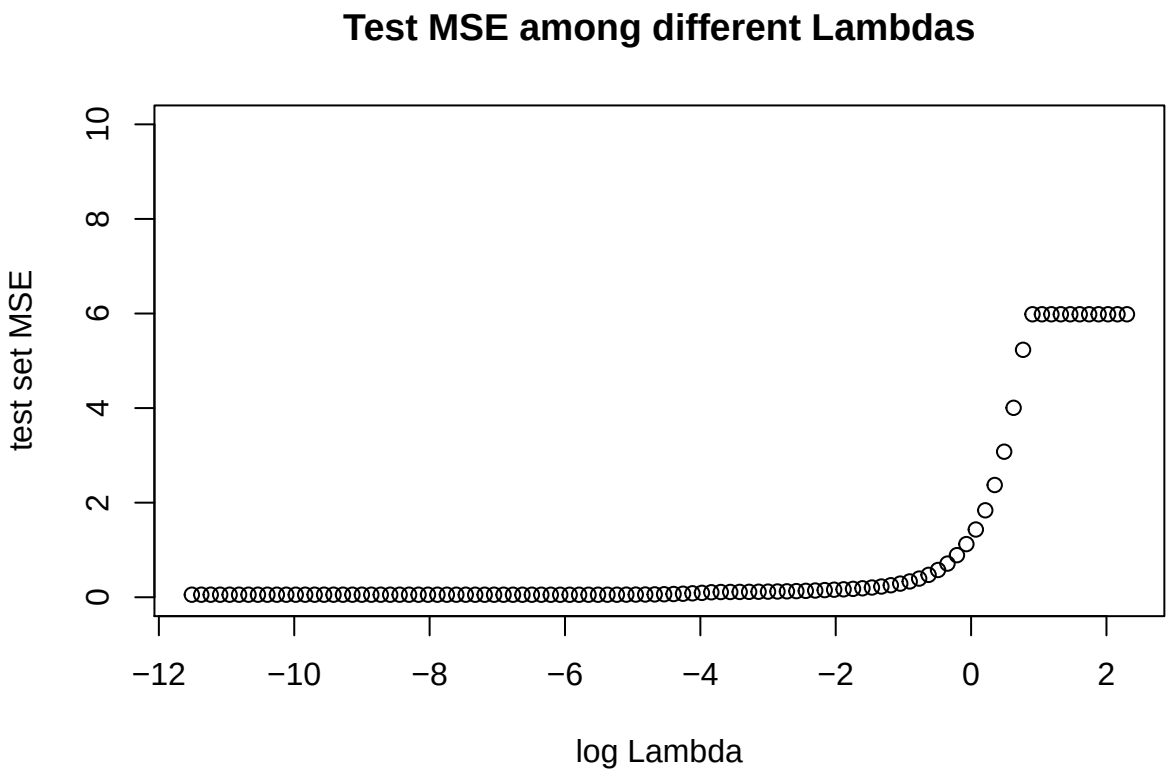
```
sqrt(sum(coef(lasso.mod)[-1,90]^2))
```

```
## [1] 0.9722552
```

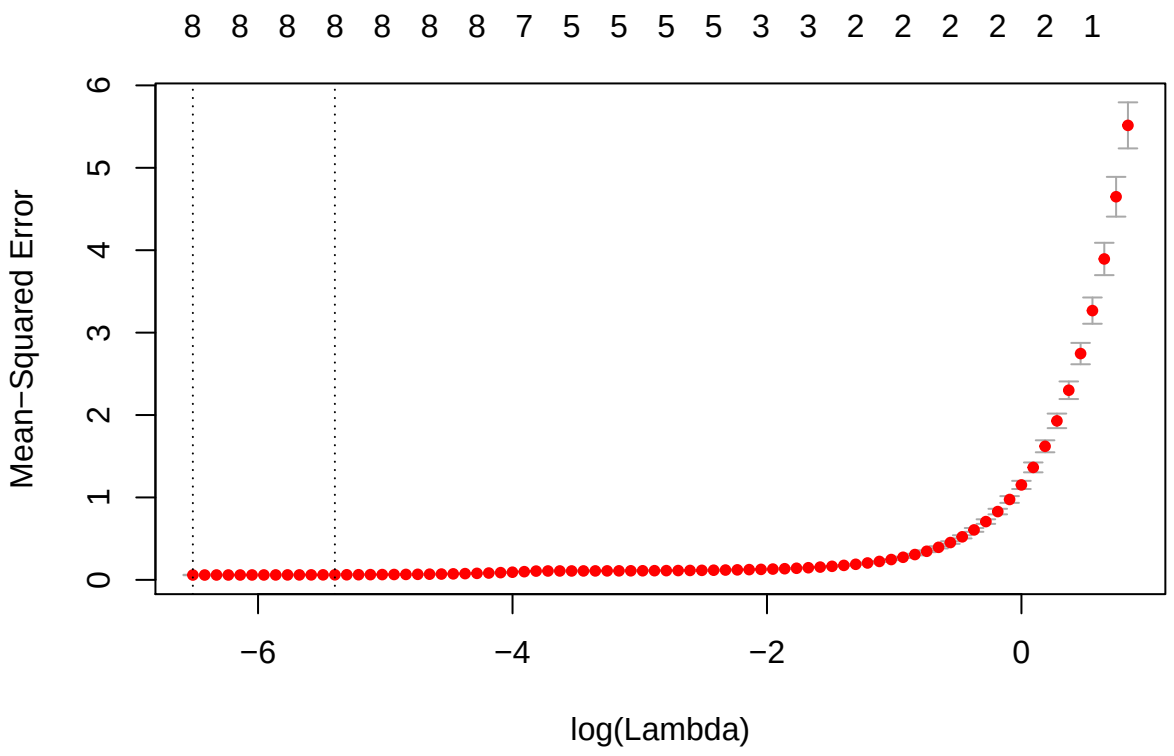
```
# predict values for a new lambda, e.g. OLS
OLS = predict(lasso.mod, s = 0, type = "coefficients")[1:nrow(coef(lasso.mod)),]

# split the date leaving the 10% for CV
train = sample(1:nrow(salary), floor(nrow(salary)*0.9))
test = -train
y.test = y[test]
lasso.mod = glmnet(x[train,], y[train], alpha = 1, lambda = grid, thresh = 1e-12)
err.i = rep("NA", length(grid))
for (i in 1:length(grid)){
  lasso.pred = predict(lasso.mod, s = grid[i], newx = x[test,])
  err.i[i] = mean((lasso.pred - y.test)^2)
}
plot(log(grid), err.i, xlab = 'log Lambda', ylab = 'test set MSE',
     main = 'Test MSE among different Lambdas', ylim = c(0, 10))
```




```
bestlam = grid[which.min(err.i)]
points(log(grid)[bestlam], err.i[bestlam], col ="red", cex=2, pch=20)
```




```
# high values of lambda are like fitting just the intercept

# using 10 folds CV
set.seed (1)
cv.out = cv.glmnet(x[train ,], y[train], alpha = 1)
plot(cv.out)
```




```
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 0.001484687
```

```
lasso.pred = predict(lasso.mod, s=bestlam, newx=x[test ,])
mean((lasso.pred - y.test)^2)
```

```
## [1] 0.05322848
```

```
# using best subset
require(leaps)
```

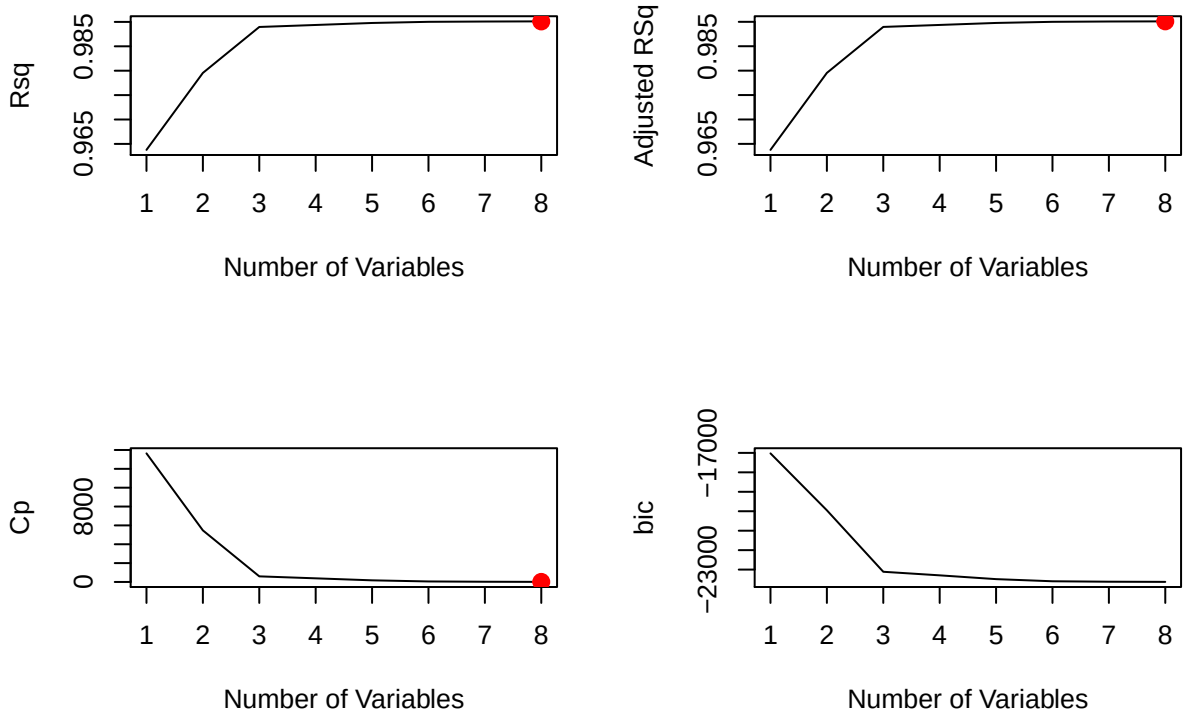
```
## Loading required package: leaps
```

```
dataBS = as.data.frame(cbind(y, x))
best.sub = regsubsets(y ~ x, data = dataBS, nvmax = nrow(coef(lasso.mod)))
best.sub.summary = summary(best.sub)
names(best.sub.summary)
```

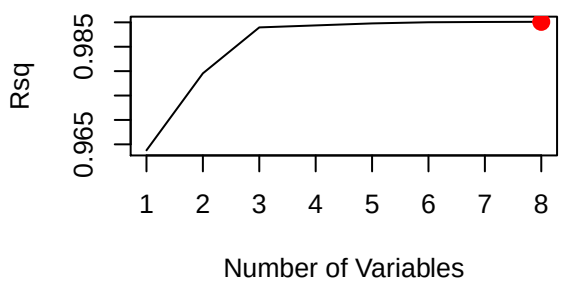
```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

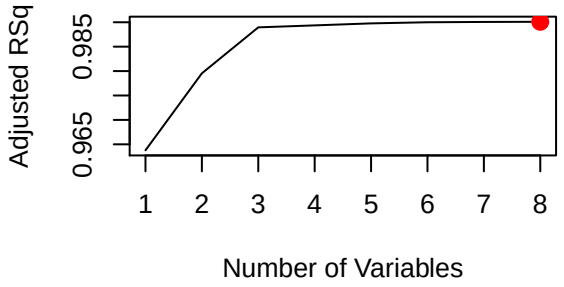
```
# manual plotting
par(mfrow =c(2,2))
# rsq
plot(best.sub.summary$rsq , xlab="Number of Variables", ylab="Rsq", type="l")
ind_Rsq = which.max(best.sub.summary$rsq)
points(ind_Rsq, best.sub.summary$adjr2[ind_Rsq], col ="red", cex=2, pch=20)
# adjRsq
```

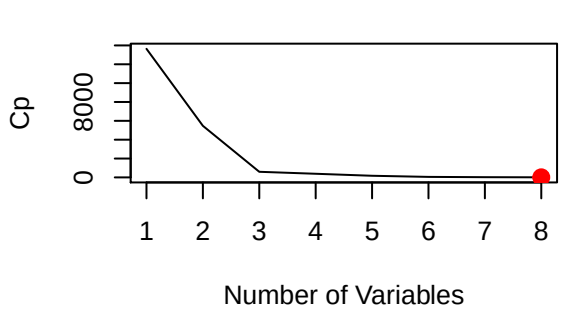
```r
plot(best.sub.summary$adjr2 ,xlab="Number of Variables", ylab="Adjusted RSq", type="l")
ind_adjRsq = which.max(best.sub.summary$adjr2)
points(ind_adjRsq, best.sub.summary$adjr2[ind_adjRsq], col ="red", cex=2, pch=20)
# Cp
plot(best.sub.summary$cp ,xlab="Number of Variables", ylab="Cp", type="l")
ind_Cp = which.min(best.sub.summary$cp)
points(ind_Cp, best.sub.summary$cp[ind_adjRsq], col ="red", cex=2, pch=20)
# bic
plot(best.sub.summary$bic ,xlab="Number of Variables", ylab="bic", type="l")
```

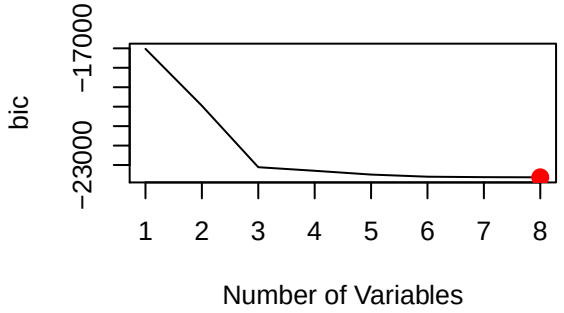



```r
ind_bic = which.min(best.sub.summary$bic)
points(ind_bic, best.sub.summary$bic[ind_bic], col ="red", cex=2, pch=20)
```







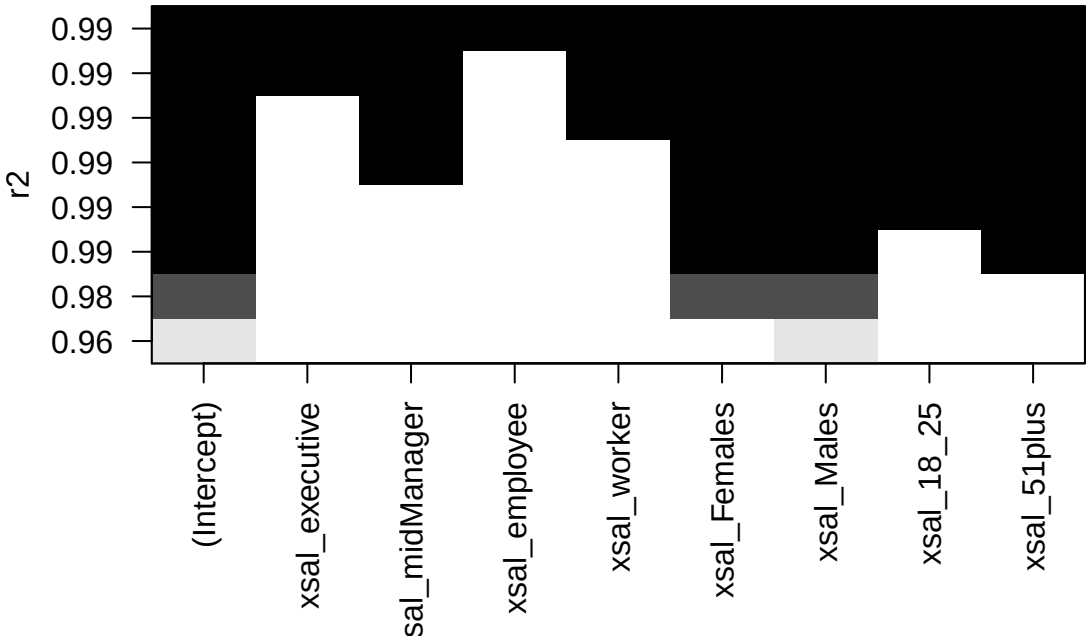



```
# built-in plots
?plot.regsubsets
```

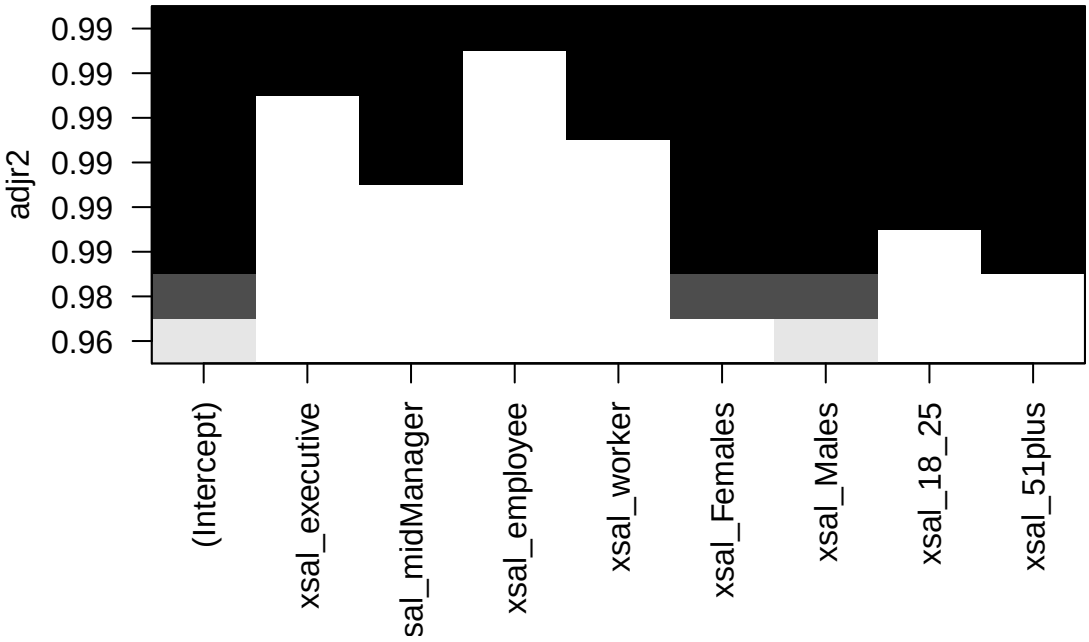
```
## starting httpd help server ...
```

```
##  done
```

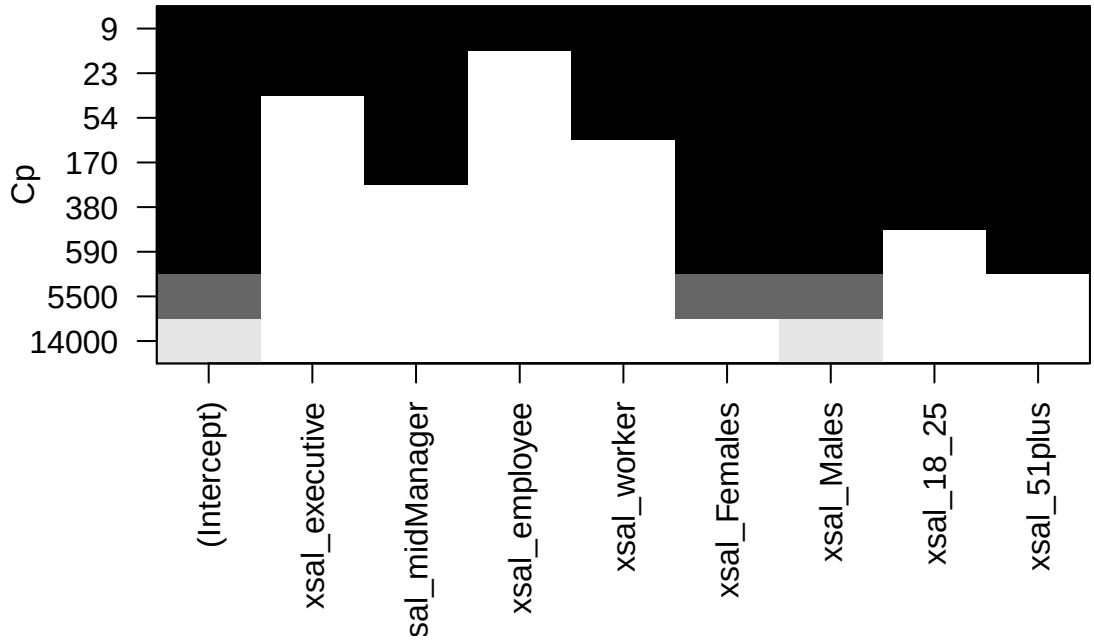
```
par(mfrow=c(1,1))
plot(best.sub, scale = "r2")
```



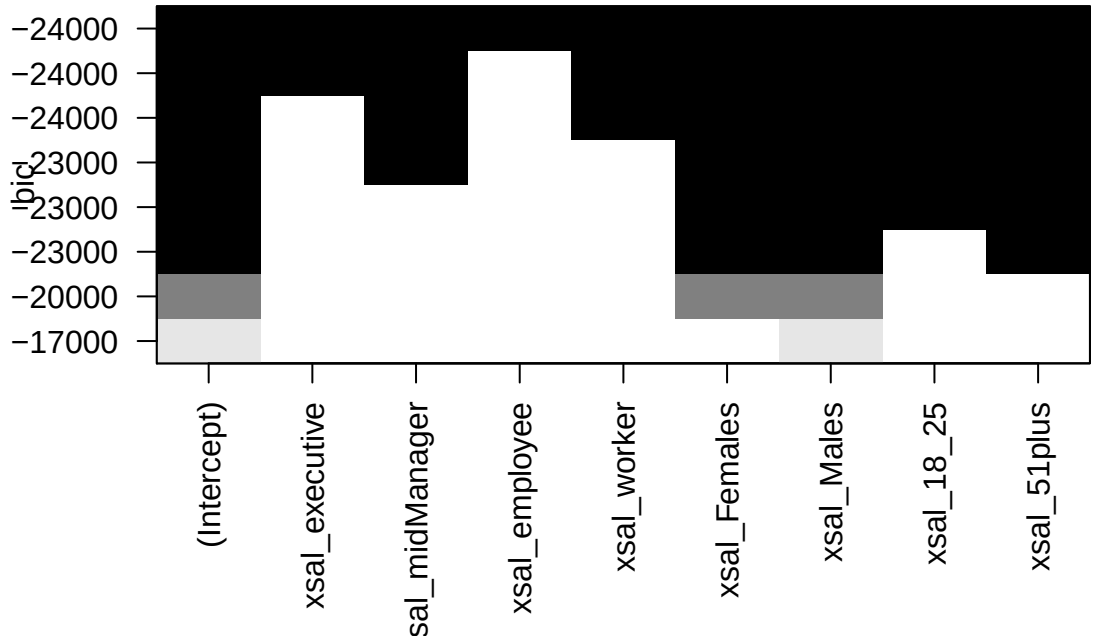

```
plot(best.sub, scale = "adjr2")
```




```
plot(best.sub, scale = "Cp")
```




```
plot(best.sub, scale = "bic")
```

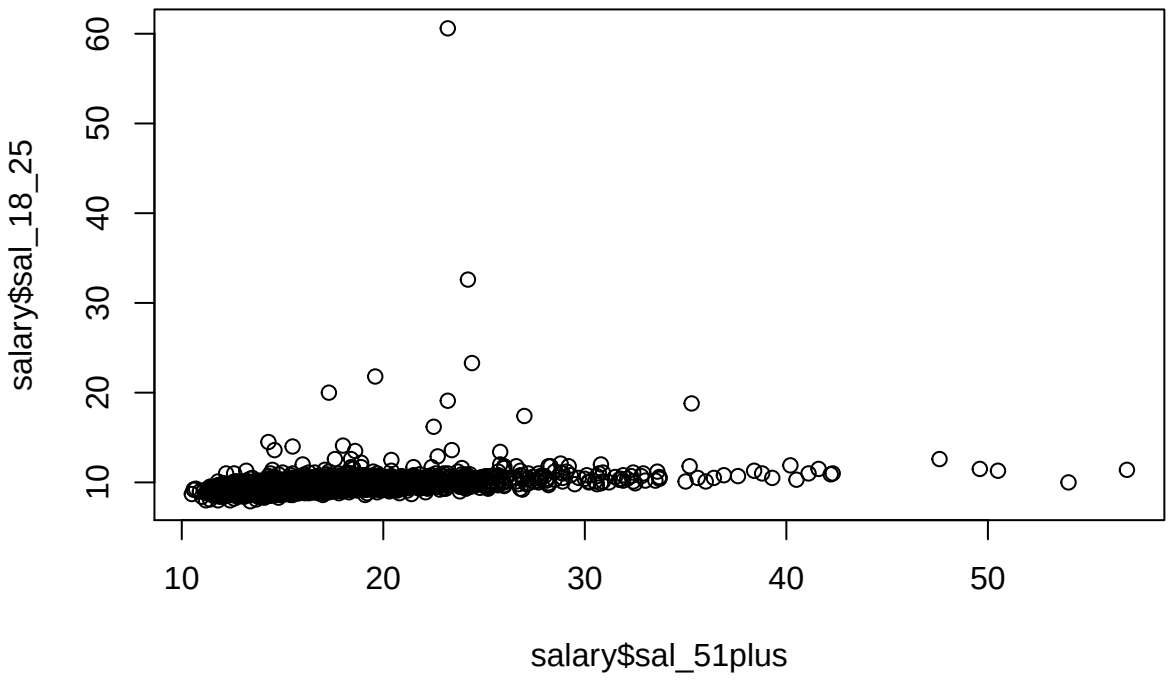



```
# retrieve the model with min BIC
coefficients(best.sub, which.min(best.sub.summary$bic))
```

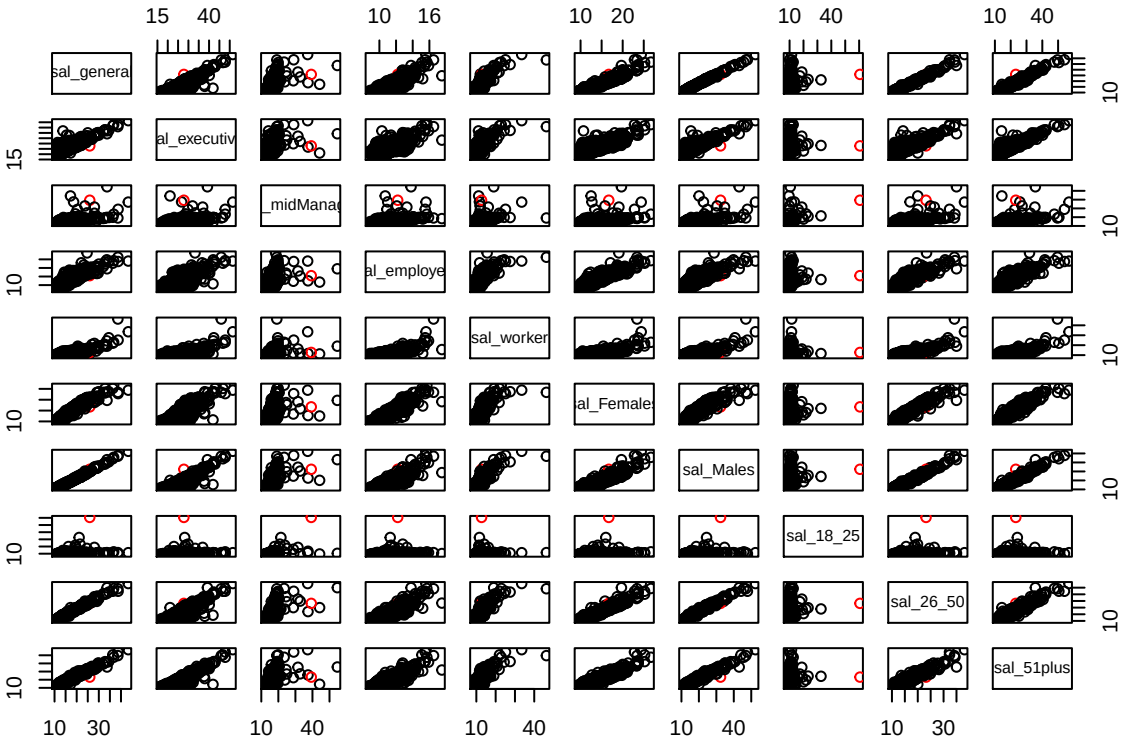
```
##     (Intercept)  xsal_executive xsal_midManager   xsal_employee
##     -0.25048976     -0.01269488      0.04760461      0.04132812
##     xsal_worker    xsal_Females      xsal_Males      xsal_18_25
##      0.04170107      0.56842466      0.73332696     -0.07044077
##     xsal_51plus
##     -0.29040179
```

Model salaries for people aged 18-25, which seems more difficult to predict:

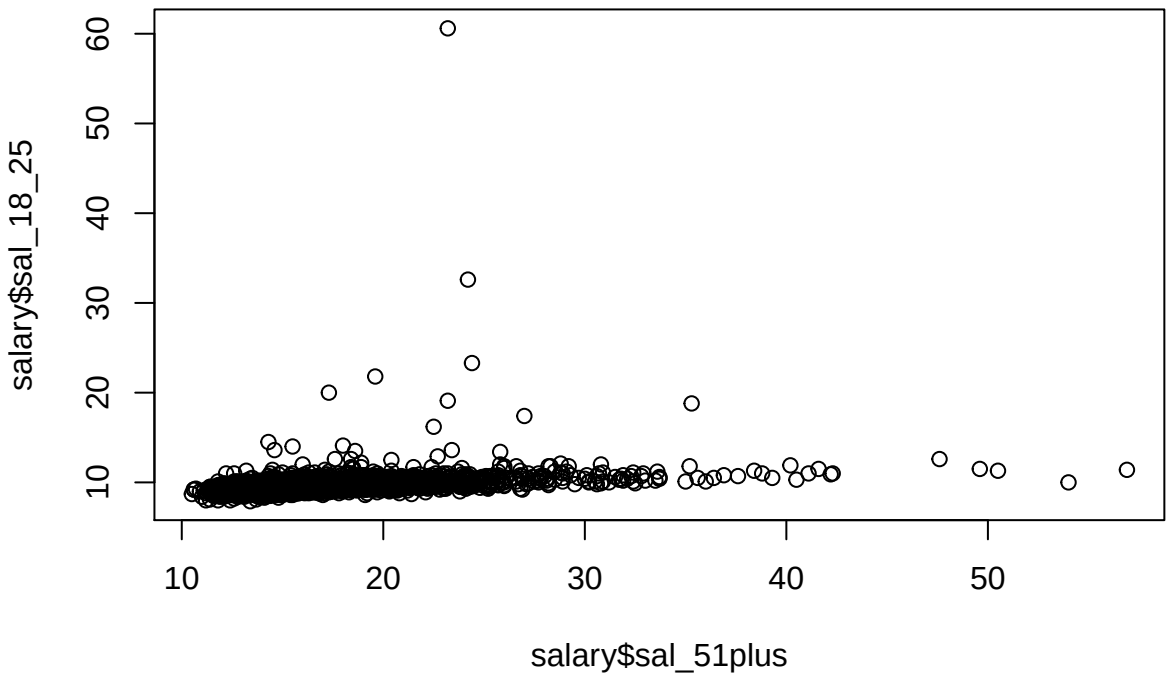
```
plot(y = salary$sal_18_25, x = salary$sal_51plus)
```



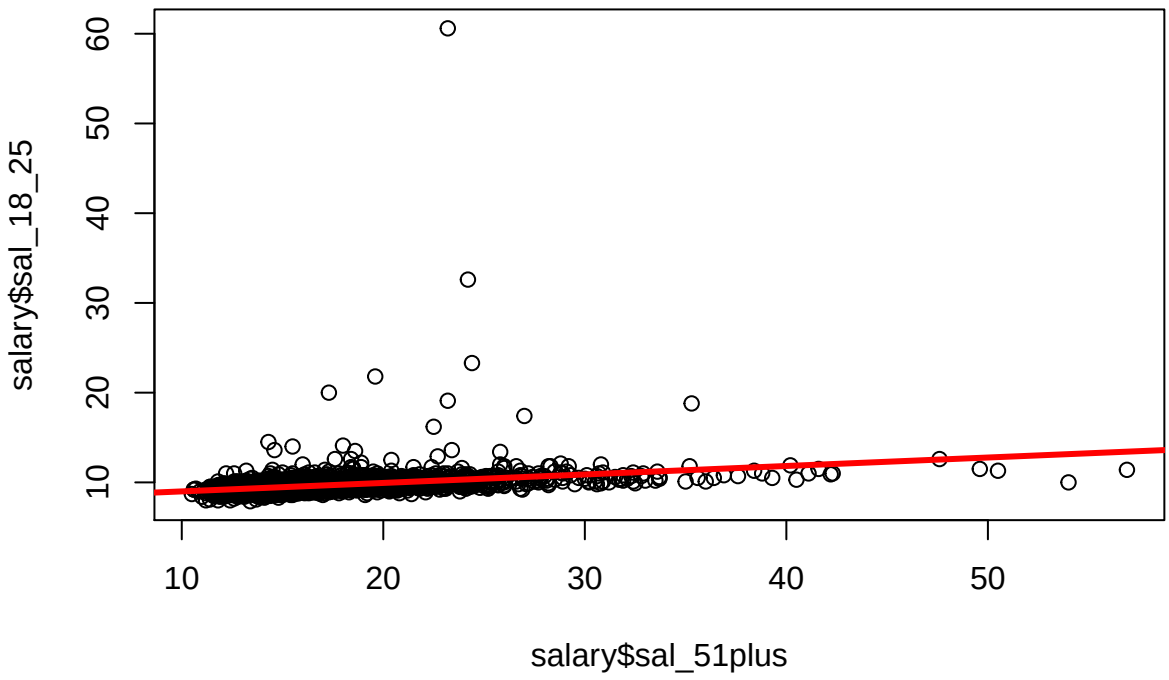

```
# there is a clear outlier
ind_out = which(salary$sal_18_25 == max(salary$sal_18_25))
# check if is so in all dimensions (apparently not)
col_col <- rep("black", nrow(salary))
col_col[ind_out] <- "red"
pairs(salary[, c(3:8, 13, 18:20)], col = col_col)
```




```
# evaluate an OLS fit
fit_LM_18_25 = lm(salary$sal_18_25 ~ salary$sal_51plus)
plot(y = salary$sal_18_25, x = salary$sal_51plus)
```




```
abline(fit_LM_18_25, lwd=3, col="red")
```




```
# # using more predictors
# fit_LM_2 = lm(salary$sal_M_18_25 ~ salary$sal_26_50 + salary$sal_51plus + salary$sal_general + salary
#                 salary$sal_midManager + salary$sal_employee + salary$sal_worker)
#
# summary(fit_LM_2)
```

PCA for salary:

```
myPr <- prcomp(salary[, 3:26], scale = TRUE)
myPr
```

```
## Standard deviations (1, .., p=24):
##  [1] 4.06846384 1.42984463 1.05721988 1.01203703 0.88322482 0.75520781
##  [7] 0.66774657 0.62302595 0.59239652 0.54263890 0.42693995 0.34427033
## [13] 0.24469681 0.18858814 0.09684001 0.08031760 0.07067011 0.06516261
## [19] 0.05970587 0.05220597 0.04552586 0.03013818 0.02484499 0.01231068
##
## Rotation (n x k) = (24 x 24):
##                          PC1          PC2          PC3          PC4
## sal_general      0.24256129  0.025437648  0.046824213 -0.001482692
## sal_executive    0.20442703  0.090840110  0.370504255  0.230155704
## sal_midManager   0.18190516 -0.298541632  0.240776890 -0.140328052
## sal_employee     0.22422720  0.071074352 -0.247268798 -0.004042860
## sal_worker       0.20363449  0.025689939 -0.009919171 -0.462312126
## sal_Females      0.23675441  0.077421630 -0.127192423  0.117888907
## sal_F_executive  0.17552119  0.100201572  0.134347070  0.360580387
## sal_F_midManager 0.20157041  0.055250814 -0.153976516  0.149031116
```

```
## sal_F_employee   0.21685115  0.070759163 -0.278486962  0.062336516
## sal_F_worker     0.17716298  0.057667740 -0.157481946 -0.339303819
## sal_Males        0.23974996  0.009058135  0.108615013 -0.027686738
## sal_M_executive  0.19790649  0.083491339  0.384046886  0.195161560
## sal_M_midManager 0.15142477 -0.358343778  0.337293684 -0.200802616
## sal_M_employee   0.18899143  0.053355476 -0.169087541 -0.135169015
## sal_M_worker     0.19759147  0.020542131  0.014779054 -0.466418793
## sal_18_25        0.10622764 -0.581187020 -0.184539817  0.163255107
## sal_26_50        0.24075123  0.021743461  0.009686478 -0.023148467
## sal_51plus       0.23674140  0.067331626  0.096987533  0.053894395
## sal_F_18_25      0.15872871 -0.016579718 -0.401651320  0.127716025
## sal_F_26_50      0.23464487  0.070413931 -0.134126680  0.098562960
## sal_F_51plus     0.22653207  0.089063623 -0.110555475  0.170095574
## sal_M_18_25      0.08512445 -0.608375999 -0.120979871  0.149766048
## sal_M_26_50      0.23769037  0.003296056  0.072159424 -0.061623773
## sal_M_51plus     0.23172375  0.063471847  0.155656542  0.046627399
##                         PC5         PC6          PC7          PC8
## sal_general       0.03076820 -0.06451689  0.055294659 -0.109456636
## sal_executive     0.21350371  0.06702947  0.104690622  0.145407429
## sal_midManager   -0.46401004  0.11032025 -0.039816322  0.040856787
## sal_employee     -0.12740320 -0.26329990  0.014366575  0.145665182
## sal_worker        0.22036076  0.13598966 -0.117636498  0.083375097
## sal_Females      -0.04566119 -0.00897671 -0.086720278 -0.154060756
## sal_F_executive   0.05685253  0.19723908 -0.624677246  0.515695940
## sal_F_midManager -0.30014668  0.08207409 -0.170354268 -0.244385975
## sal_F_employee   -0.16747839 -0.07036376 -0.061129067 -0.068005412
## sal_F_worker      0.23212123  0.26673263 -0.223203762 -0.157579547
## sal_Males         0.05507451 -0.09080437  0.109670620 -0.104435046
## sal_M_executive   0.22445421  0.03841633  0.283820931  0.005970518
## sal_M_midManager -0.45538809  0.09851366  0.008813757  0.102547206
## sal_M_employee   -0.04337624 -0.57642720  0.149112333  0.540346236
## sal_M_worker      0.22279564  0.11403974 -0.098836166  0.127142364
## sal_18_25         0.26393156 -0.01583997  0.002640546  0.001207696
## sal_26_50        -0.01895745 -0.07602419  0.036997646 -0.096505574
## sal_51plus        0.06972164 -0.03408813  0.097629771 -0.145711593
## sal_F_18_25      -0.07694791  0.60485366  0.529432039  0.330226773
## sal_F_26_50      -0.03913166 -0.02936739 -0.075060725 -0.157937046
## sal_F_51plus     -0.07331241  0.01993173 -0.147758727 -0.179672634
## sal_M_18_25       0.29054243 -0.13104749 -0.094878245 -0.061587053
## sal_M_26_50      -0.01062363 -0.10331821  0.086592887 -0.080401426
## sal_M_51plus      0.10720803 -0.05185343  0.173774628 -0.146634702
##                         PC9        PC10         PC11         PC12
## sal_general       0.027674801 -0.17031410  0.080977040 -0.072080349
## sal_executive    -0.095281235  0.27788374 -0.116184031 -0.091277662
## sal_midManager   -0.057858294  0.09569372  0.035330196  0.042527735
## sal_employee     -0.024675374  0.13257444 -0.327104176  0.043849057
## sal_worker        0.292638010  0.11437721 -0.009936159  0.020419566
## sal_Females       0.014970640 -0.13019350 -0.005154802 -0.052074804
## sal_F_executive   0.024647934 -0.19920315  0.053743470 -0.029629692
## sal_F_midManager  0.109517724  0.61073768  0.498058587 -0.074307675
## sal_F_employee    0.061094057  0.15347166 -0.656257011 -0.053673559
## sal_F_worker     -0.777224505  0.05063640  0.007552204  0.066475744
## sal_Males         0.018817088 -0.18588939  0.102679421 -0.081239041
## sal_M_executive  -0.115734021  0.37274958 -0.172453447 -0.060289875
```

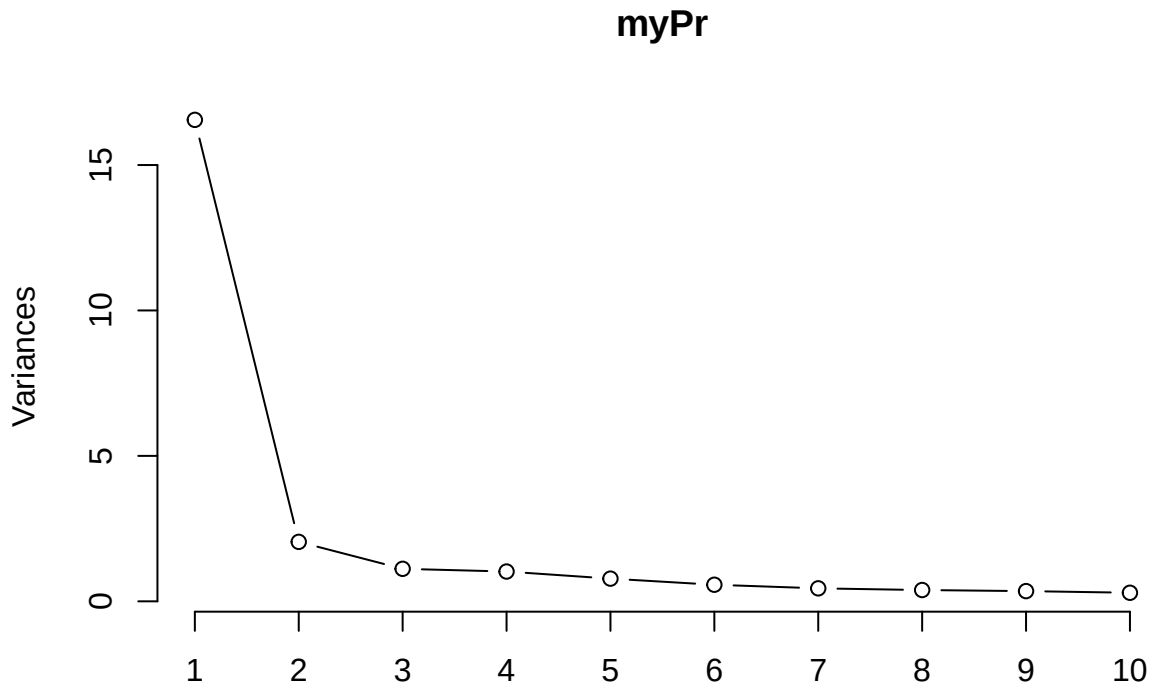
```
## sal_M_midManager -0.109776798 -0.12475279 -0.157598363  0.077146650
## sal_M_employee   -0.219903667  0.08926476  0.293843520  0.121124001
## sal_M_worker      0.436609055  0.12512635 -0.024954387 -0.004215415
## sal_18_25         0.023279317  0.04512387  0.017294431  0.002633665
## sal_26_50         0.002769069 -0.20498532  0.073845015 -0.364981228
## sal_51plus        0.074302719 -0.14219097  0.075553331  0.399980948
## sal_F_18_25       0.021573772 -0.10397906  0.086328029 -0.000224029
## sal_F_26_50       0.005991692 -0.15389749 -0.003958265 -0.313921974
## sal_F_51plus      0.072336569 -0.10300216 -0.029610665  0.522101524
## sal_M_18_25       0.020934887  0.05728994  0.001050080  0.009639714
## sal_M_26_50      -0.008463477 -0.22436907  0.096794152 -0.380087545
## sal_M_51plus      0.053171099 -0.15006742  0.092894490  0.355160148
##                          PC13         PC14        PC15         PC16
## sal_general       0.034011467 -0.087574401  0.11373308  0.537564433
## sal_executive    -0.050262712 -0.094749290  0.21491568  0.036365780
## sal_midManager    0.048701263  0.007430303  0.63673058 -0.057984682
## sal_employee      0.101729065 -0.054861091 -0.07035113  0.067292007
## sal_worker       -0.067190044  0.073124691  0.06597172  0.060818933
## sal_Females      -0.166052462  0.362943040 -0.03857624  0.369567679
## sal_F_executive   0.150284303 -0.070732588 -0.07123694 -0.015017006
## sal_F_midManager  0.125149260 -0.106259680 -0.21693732  0.007488834
## sal_F_employee    0.217741396 -0.203891674  0.02547217 -0.043943649
## sal_F_worker      0.056051310 -0.048241531 -0.01598948 -0.019478631
## sal_Males         0.098859708 -0.255477734 -0.16183855  0.397833221
## sal_M_executive  -0.220371258  0.201825172 -0.15473575 -0.031433765
## sal_M_midManager -0.065776558  0.139626301 -0.52596569  0.054175552
## sal_M_employee   -0.091199655  0.106345961  0.02179142 -0.032252775
## sal_M_worker     -0.041622685  0.045210768 -0.07283739 -0.075671203
## sal_18_25         0.013002070 -0.016293108  0.11267039  0.136111542
## sal_26_50        -0.100043127 -0.098288179  0.15596215 -0.240473153
## sal_51plus        0.234263596  0.011064433  0.20197212 -0.010017352
## sal_F_18_25      -0.008227617 -0.025813810 -0.02877802 -0.022084438
## sal_F_26_50       0.039401540  0.651433643  0.07061276 -0.232593633
## sal_F_51plus     -0.660491905 -0.132164963  0.01971171 -0.153288200
## sal_M_18_25      -0.004391171 -0.001068603 -0.11587697 -0.154164465
## sal_M_26_50      -0.163926845 -0.432693213 -0.12254212 -0.331702528
## sal_M_51plus      0.510074815  0.066303930 -0.16311384 -0.318822068
##                          PC17         PC18          PC19         PC20
## sal_general       0.02743203  0.017489988 -1.159451e-01 -0.004941698
## sal_executive     0.60389368  0.183292088  1.113154e-02  0.209707749
## sal_midManager   -0.28448202 -0.073557676 -1.579773e-01  0.048891526
## sal_employee      0.17510667 -0.754946928 -1.146551e-01  0.037348700
## sal_worker        0.11187552  0.010201055  1.064764e-01 -0.667777877
## sal_Females      -0.11208987  0.081286141  2.356819e-02  0.155294284
## sal_F_executive  -0.15296912 -0.053925877  2.295057e-05 -0.068963399
## sal_F_midManager  0.07002096  0.014105017  4.580890e-02 -0.026142503
## sal_F_employee   -0.14364839  0.483278945  7.405760e-02 -0.061767647
## sal_F_worker     -0.01489252 -0.012757413 -1.294547e-02  0.078849344
## sal_Males        -0.16104495  0.065851739 -1.876303e-01  0.002719009
## sal_M_executive  -0.48853916 -0.147678103 -1.172190e-02 -0.169556473
## sal_M_midManager  0.25541377  0.073609475  1.327127e-01 -0.021557735
## sal_M_employee   -0.05884846  0.283858888  4.685199e-02 -0.008460942
## sal_M_worker     -0.09758683  0.006946048 -8.677017e-02  0.610432699
## sal_18_25        -0.07112277 -0.109030858  6.634365e-01  0.107558876
```

```
## sal_26_50         0.12380994 -0.080862049  6.569177e-02 -0.023716404
## sal_51plus        0.19641903 -0.041589840  3.230681e-02 -0.164704064
## sal_F_18_25       0.01487853  0.017465862 -1.179785e-01 -0.021515015
## sal_F_26_50       0.12772260  0.028461809 -2.675494e-02 -0.014024159
## sal_F_51plus      0.01638906  0.022108181 -5.690660e-04  0.043471311
## sal_M_18_25       0.07360042  0.100079168 -6.191465e-01 -0.107431569
## sal_M_26_50      -0.09163891 -0.056093094  1.197454e-01 -0.039867540
## sal_M_51plus     -0.12772664  0.009630106  9.046612e-02  0.099182609
##                         PC21         PC22         PC23          PC24
## sal_general      -0.186444087 -0.171634618 -0.318250742 -0.6211947949
## sal_executive     0.237625932  0.094261014  0.051498899 -0.0009957993
## sal_midManager    0.145407708  0.074636362  0.046910335 -0.0028543575
## sal_employee      0.097556946  0.039721344  0.018583473 -0.0002310930
## sal_worker        0.259563686  0.033684544 -0.025598837 -0.0021445691
## sal_Females       0.414226899 -0.405780691  0.378673412  0.1835641555
## sal_F_executive  -0.063321530 -0.024249456 -0.013372003  0.0009117450
## sal_F_midManager -0.058101242 -0.025079710 -0.016977172  0.0013027139
## sal_F_employee   -0.087559551 -0.033287099 -0.022210938  0.0014008536
## sal_F_worker     -0.040868629 -0.004198171  0.001678796  0.0003570630
## sal_Males         0.045217001  0.551474149 -0.042726401  0.4589400496
## sal_M_executive  -0.197052690 -0.077600321 -0.040965609  0.0010058968
## sal_M_midManager -0.109754841 -0.056017369 -0.034265555  0.0021278696
## sal_M_employee   -0.035280604 -0.011036698 -0.007791783 -0.0002416376
## sal_M_worker     -0.203024554 -0.024335767  0.030257103  0.0025064712
## sal_18_25        -0.034850157  0.119248356 -0.037628737  0.0160530627
## sal_26_50        -0.117863633 -0.419501092 -0.475425046  0.4460448356
## sal_51plus       -0.540731132 -0.144167569  0.439057354  0.1690205439
## sal_F_18_25       0.001253538 -0.008376178  0.002582028 -0.0034582040
## sal_F_26_50      -0.150941457  0.456530696 -0.027601748 -0.1483871402
## sal_F_51plus      0.035742420  0.177587916 -0.187687161 -0.0459163395
## sal_M_18_25       0.032360802 -0.129213299  0.043266374 -0.0149062160
## sal_M_26_50       0.095574005  0.028289510  0.477463900 -0.3177600192
## sal_M_51plus      0.430473611 -0.062171646 -0.238321858 -0.1290375524
```

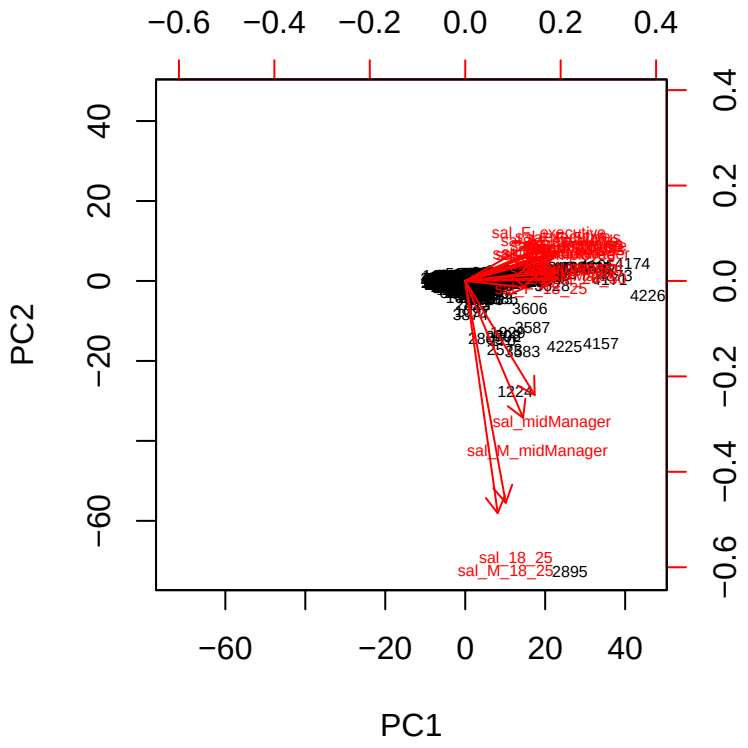
```
summary(myPr)
```

```
## Importance of components:
##                            PC1     PC2     PC3     PC4    PC5     PC6
## Standard deviation     4.0685 1.42984 1.05722 1.01204 0.8832 0.75521
## Proportion of Variance 0.6897 0.08519 0.04657 0.04268 0.0325 0.02376
## Cumulative Proportion  0.6897 0.77487 0.82144 0.86412 0.8966 0.92038
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.66775 0.62303 0.59240 0.54264 0.42694 0.34427
## Proportion of Variance 0.01858 0.01617 0.01462 0.01227 0.00759 0.00494
## Cumulative Proportion  0.93896 0.95514 0.96976 0.98203 0.98962 0.99456
##                           PC13    PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.24470 0.18859 0.09684 0.08032 0.07067 0.06516
## Proportion of Variance 0.00249 0.00148 0.00039 0.00027 0.00021 0.00018
## Cumulative Proportion  0.99706 0.99854 0.99893 0.99920 0.99940 0.99958
##                           PC19    PC20    PC21    PC22    PC23    PC24
## Standard deviation     0.05971 0.05221 0.04553 0.03014 0.02484 0.01231
## Proportion of Variance 0.00015 0.00011 0.00009 0.00004 0.00003 0.00001
## Cumulative Proportion  0.99973 0.99984 0.99993 0.99997 0.99999 1.00000
```

```r
plot(myPr, type = "l")
```




```r
biplot(myPr, scale = 0, cex = 0.5)
```




```
str(myPr)
```

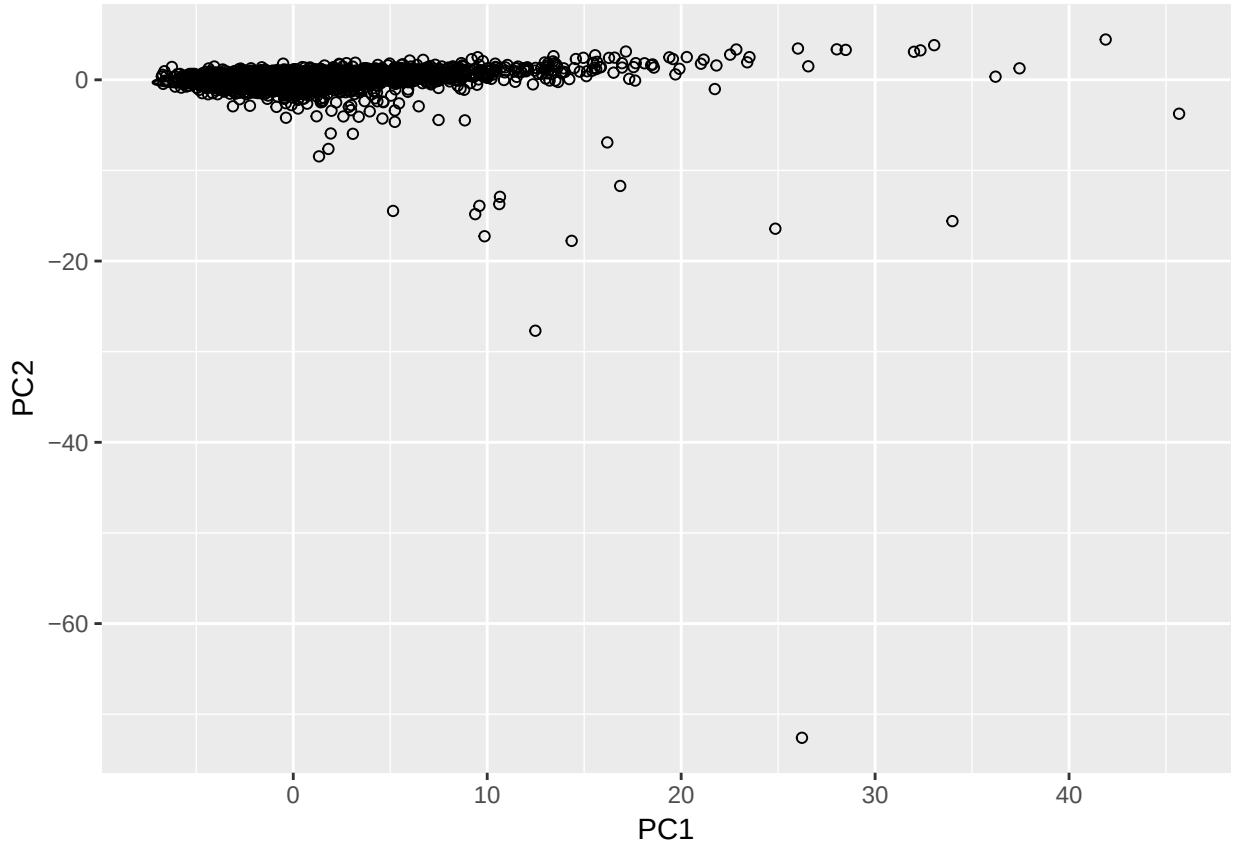
```
## List of 5
##  $ sdev    : num [1:24] 4.068 1.43 1.057 1.012 0.883 ...
##  $ rotation: num [1:24, 1:24] 0.243 0.204 0.182 0.224 0.204 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:24] "sal_general" "sal_executive" "sal_midManager" "sal_employee" ...
##   .. ..$ : chr [1:24] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:24] 13.7 23.7 14.6 10.6 11.2 ...
##   ..- attr(*, "names")= chr [1:24] "sal_general" "sal_executive" "sal_midManager" "sal_employee" ...
##  $ scale   : Named num [1:24] 2.559 2.836 1.49 0.812 1.222 ...
##   ..- attr(*, "names")= chr [1:24] "sal_general" "sal_executive" "sal_midManager" "sal_employee" ...
##  $ x       : num [1:5136, 1:24] 0.315 -0.185 0.443 -0.74 -0.922 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:24] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
#myPr$x #checking principal component scores
salary2 <- cbind(salary, myPr$x[, 1:2])
head(salary2)
```

```
##   CODGEO              town sal_general sal_executive sal_midManager
## 1   1004 Ambérieu-en-Bugey        13.7          24.2           15.5
## 2   1007          Ambronay        13.5          22.1           14.7
## 3   1014            Arbent        13.5          27.6           15.6
## 4   1024          Attignat        12.9          21.8           14.1
```

```
## 5   1025     Bâgé-la-Ville        13.0          22.8           14.1
## 6   1027             Balan        13.9          22.2           15.1
##   sal_employee sal_worker sal_Females sal_F_executive sal_F_midManager
## 1         10.3       11.2        11.6            19.1             13.2
## 2         10.7       11.4        11.9            19.0             13.3
## 3         11.1       11.1        10.9            19.5             11.7
## 4         11.0       11.3        11.4            19.0             13.0
## 5         10.5       11.1        11.6            19.4             13.6
## 6         11.0       11.4        12.5            20.3             14.0
##   sal_F_employee sal_F_worker sal_Males sal_M_executive sal_M_midManager
## 1           10.1          9.6      15.0            26.4             16.7
## 2           10.6         10.0      14.7            23.3             15.8
## 3           10.8          9.5      15.3            30.2             17.2
## 4           10.3          9.9      13.8            23.0             14.7
## 5           10.2          9.8      13.8            24.1             14.4
## 6           10.9         10.5      15.2            23.1             15.9
##   sal_M_employee sal_M_worker sal_18_25 sal_26_50 sal_51plus sal_F_18_25
## 1           11.0         11.6      10.5      13.7       16.1         9.7
## 2           11.3         11.7       9.8      13.8       14.6         9.2
## 3           12.4         11.8       9.3      13.3       16.0         8.9
## 4           13.2         11.6       9.6      12.9       14.2         9.3
## 5           11.7         11.4       9.4      12.8       15.2         9.0
## 6           12.1         11.7       9.7      14.1       15.4         9.5
##   sal_F_26_50 sal_F_51plus sal_M_18_25 sal_M_26_50 sal_M_51plus        PC1
## 1        11.8         12.5        11.0        14.9         18.6  0.3154969
## 2        12.2         12.5        10.2        14.9         16.4 -0.1853773
## 3        10.6         12.5         9.6        15.1         18.6  0.4431224
## 4        11.4         12.2         9.7        13.8         15.9 -0.7404781
## 5        11.8         12.3         9.7        13.4         16.9 -0.9218165
## 6        12.8         13.0         9.9        15.3         17.2  1.0063385
##           PC2
## 1 -1.51556989
## 2 -0.52833502
## 3 -0.07994227
## 4  0.01522957
## 5  0.22224766
## 6 -0.14256519
```

```r
#plot with ggplot...
#require(ggplot2)
ggplot(salary2, aes(PC1, PC2)) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```




```
# correlations between variables and PCs...
cor(salary[, 3:26], salary2[,27:28])
```

```
##                        PC1          PC2
## sal_general      0.9868518  0.036371885
## sal_executive    0.8317040  0.129887244
## sal_midManager   0.7400745 -0.426868150
## sal_employee     0.9122603  0.101625280
## sal_worker       0.8284796  0.036732621
## sal_Females      0.9632267  0.110700902
## sal_F_executive  0.7141016  0.143272680
## sal_F_midManager 0.8200819  0.079000080
## sal_F_employee   0.8822511  0.101174610
## sal_F_worker     0.7207812  0.082455908
## sal_Males        0.9754141  0.012951726
## sal_M_executive  0.8051754  0.119379642
## sal_M_midManager 0.6160662 -0.512375927
## sal_M_employee   0.7689048  0.076290041
## sal_M_worker     0.8038938  0.029372055
## sal_18_25        0.4321833 -0.831007140
## sal_26_50        0.9794877  0.031089772
## sal_51plus       0.9631738  0.096273763
## sal_F_18_25      0.6457820 -0.023706420
## sal_F_26_50      0.9546442  0.100680982
## sal_F_51plus     0.9216375  0.127347143
## sal_M_18_25      0.3463258 -0.869883156
```

```
## sal_M_26_50      0.9670347  0.004712847
## sal_M_51plus     0.9427597  0.090754879
```

What we have learned

- Check the outlier in sal_18_25: which city is it, etc.
- evaluate possible multicollinearity
- Try a regression with all predictors ans lasso

How to use these data

- ...

Analyze population data

Pre-processing

```
# preliminary checks
names(population)
```

```
## [1] "NIVGEO"         "CODGEO"         "LIBGEO"         "MOCO"
## [5] "ageCateg5"      "sex"            "peopleCategNum"
```

```
summary(population)
```

```
##  NIVGEO           CODGEO                     LIBGEO             MOCO
##  COM:8536584   Length:8536584     Sainte-Colombe:   3094   Min.   :11.00
##                Class :character   Saint-Sauveur :   2618   1st Qu.:12.00
##                Mode  :character   Sainte-Marie  :   2618   Median :22.00
##                                   Beaulieu      :   2380   Mean   :21.71
##                                   Le Pin        :   2380   3rd Qu.:31.00
##                                   Saint-Aubin   :   2380   Max.   :32.00
##                                   (Other)       :8521114
##    ageCateg5       sex      peopleCategNum
##  Min.   : 0   Min.   :1.0   Min.   :    0.00
##  1st Qu.:20   1st Qu.:1.0   1st Qu.:    0.00
##  Median :40   Median :1.5   Median :    0.00
##  Mean   :40   Mean   :1.5   Mean   :    7.45
##  3rd Qu.:60   3rd Qu.:2.0   3rd Qu.:    3.00
##  Max.   :80   Max.   :2.0   Max.   :48873.00
##
```

```
# Drop unnecessary columns (NIVGEO is the same for all)
population <- subset(population, select = -c(NIVGEO, LIBGEO))

# converting CODGEO to numeric
population$CODGEO <- as.numeric(population$CODGEO)

# Refactor sex and MOCO
population$MOCO <- factor(population$MOCO, levels = c(11,12,21,22,23,31,32),
                          labels = c("children_living_with_two_parents", "children living with one parer
```

```
                                    "adults_living_in_couple_without_child", "adults_living_in_couple_w
                                    "adults_living_alone_with_children","persons not from family living
                                    "persons_living_alone"))
population$sex <- factor(population$sex, levels = c(1,2), labels = c("Male", "Female"))
head(population)
```

```
##   CODGEO                              MOCO ageCateg5    sex peopleCategNum
## 1   1001 children_living_with_two_parents         0   Male             15
## 2   1001 children_living_with_two_parents         0 Female             15
## 3   1001 children_living_with_two_parents         5   Male             20
## 4   1001 children_living_with_two_parents         5 Female             20
## 5   1001 children_living_with_two_parents        10   Male             20
## 6   1001 children_living_with_two_parents        10 Female             45
```

```
# Take out rows with NB (number of people in this category) equal to 0
population <- population[population$peopleCategNum != 0,]

head(population)
```

```
##   CODGEO                              MOCO ageCateg5    sex peopleCategNum
## 1   1001 children_living_with_two_parents         0   Male             15
## 2   1001 children_living_with_two_parents         0 Female             15
## 3   1001 children_living_with_two_parents         5   Male             20
## 4   1001 children_living_with_two_parents         5 Female             20
## 5   1001 children_living_with_two_parents        10   Male             20
## 6   1001 children_living_with_two_parents        10 Female             45
```

```
summary(population)
```

```
##      CODGEO                                              MOCO
##  Min.   : 1001   children_living_with_two_parents          :337182
##  1st Qu.:27181   children living with one parent           :192130
##  Median :50394   adults_living_in_couple_without_child     :529268
##  Mean   :48360   adults_living_in_couple_with_children     :451501
##  3rd Qu.:69154   adults_living_alone_with_children         :160976
##  Max.   :97424   persons not from family living in the home:178635
##                  persons_living_alone                      :361261
##    ageCateg5         sex          peopleCategNum
##  Min.   : 0.00   Male  :1106500   Min.   :    1.00
##  1st Qu.:25.00   Female:1104453   1st Qu.:    4.00
##  Median :45.00                    Median :    8.00
##  Mean   :41.77                    Mean   :   28.75
##  3rd Qu.:60.00                    3rd Qu.:   20.00
##  Max.   :80.00                    Max.   :48873.00
##
```

EDA

```
#Compare age categories:
library(ggplot2)
#  number of units
n_cat <- length(population$CODGEO)
# extract unique categories
uniq_cat <- unique(population$ageCateg5!=5)
```

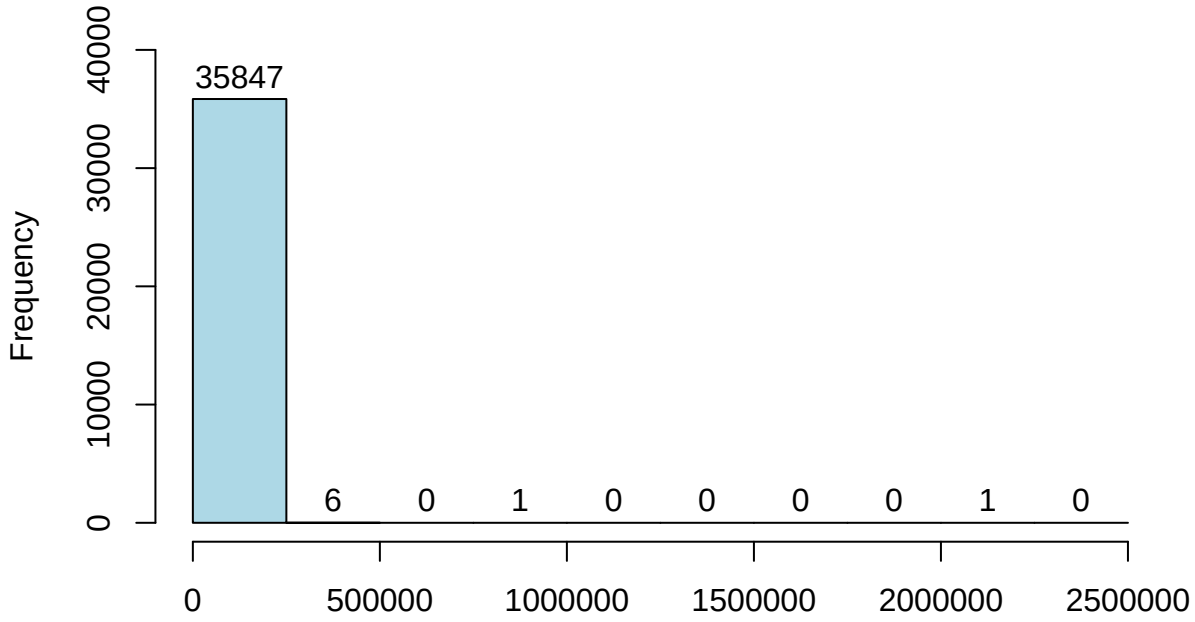
```r
# vector representing sex for each category
Label <- c(rep(c('Male', 'Female'), n_cat))
# vector representing the variable considered
Variable <- rep(uniq_cat, n_cat/length(uniq_cat))
Value=population$peopleCategNum
# merge these data
#pop_categ = cbind.data.frame(Label = Label,
#             value = Value,
#             Variable = Variable)
#p <- ggplot(data = pop_categ, aes(x=Label, y=value))
#p <- p + geom_boxplot(aes(fill = Label))
# if you want color for points replace group with colour=Label
#p <- p + geom_point(aes(y=value, colour=Label), position = position_dodge(width=0.75))
#p <- p + facet_wrap( ~ Variable, scales="free")
#p <- p + xlab("x-axis") + ylab("y-axis") + ggtitle("Category comparison")
# p <- p + guides(fill=guide_legend(title="Legend"))
#p
```

```r
# Restructure population data to produce the demographic profile per town
# install.packages("plyr")
library(plyr)
population_per_town_data <- ddply(population, .(CODGEO), function(population) {
  data.frame(total_population = sum(population$peopleCategNum),
             male = sum(population[population$sex == "Male",]$peopleCategNum),
             female = sum(population[population$sex == "Female",]$peopleCategNum),
             child = sum(population[population$ageCateg5 %in% seq(0, 10, by=5),]$peopleCategNum),
             elderly = sum(population[population$ageCateg5 %in% seq(65, 80, by=5),]$peopleCategNum),
             workforce = sum(population[population$ageCateg5 %in% seq(15, 60, by=5),]$peopleCategNum)
  )})

population_per_town_data$dependent <- population_per_town_data$child + population_per_town_data$elderly
population_per_town_data$sex_ratio <- ifelse(population_per_town_data$female==0, 0, population_per_town_
population_per_town_data$dependency_ratio <- ifelse(population_per_town_data$workforce==0, 0, populatio
population_per_town_data$aged_dependency_ratio <- ifelse(population_per_town_data$workforce==0, 0, popul
population_per_town_data$child_dependency_ratio <- ifelse(population_per_town_data$workforce==0, 0, popu
summary(population_per_town_data)
```

```
##      CODGEO      total_population       male              female
##  Min.   : 1001   Min.   :      2   Min.   :      0.0   Min.   :      0.0
##  1st Qu.:24541   1st Qu.:    184   1st Qu.:     92.0   1st Qu.:     90.0
##  Median :48098   Median :    420   Median :    212.0   Median :    209.0
##  Mean   :46276   Mean   :   1773   Mean   :    857.6   Mean   :    915.4
##  3rd Qu.:67076   3rd Qu.:   1060   3rd Qu.:    528.0   3rd Qu.:    535.0
##  Max.   :97424   Max.   :2173279   Max.   :1018597.0   Max.   :1154682.0
##      child             elderly           workforce
##  Min.   :     0.0   Min.   :     0.0   Min.   :      0.0
##  1st Qu.:    32.0   1st Qu.:    35.0   1st Qu.:    112.0
##  Median :    80.0   Median :    75.0   Median :    261.0
##  Mean   :   333.9   Mean   :   310.1   Mean   :   1128.9
##  3rd Qu.:   209.5   3rd Qu.:   190.0   3rd Qu.:    659.5
##  Max.   :315127.0   Max.   :338074.0   Max.   :1520078.0
##    dependent          sex_ratio       dependency_ratio
##  Min.   :     0.0   Min.   : 0.0000   Min.   :0.0000
##  1st Qu.:    70.0   1st Qu.: 0.9075   1st Qu.:0.5135
```

```
##  Median :   160.0   Median : 0.9971   Median :0.6088
##  Mean   :   644.1   Mean   : 1.0259   Mean   :0.6538
##  3rd Qu.:   401.0   3rd Qu.: 1.1026   3rd Qu.:0.7333
##  Max.   :653201.0   Max.   :10.0000   Max.   :9.0000
##  aged_dependency_ratio child_dependency_ratio
##  Min.   :0.0000        Min.   :0.0000
##  1st Qu.:0.2091        1st Qu.:0.2453
##  Median :0.2929        Median :0.3043
##  Mean   :0.3462        Mean   :0.3076
##  3rd Qu.:0.4166        3rd Qu.:0.3662
##  Max.   :8.0000        Max.   :3.0000
```

```r
# Scale population to log
hist(population_per_town_data$total_population, ylim=c(0,40000), breaks = seq(0, 2500000, by=250000), xl
```



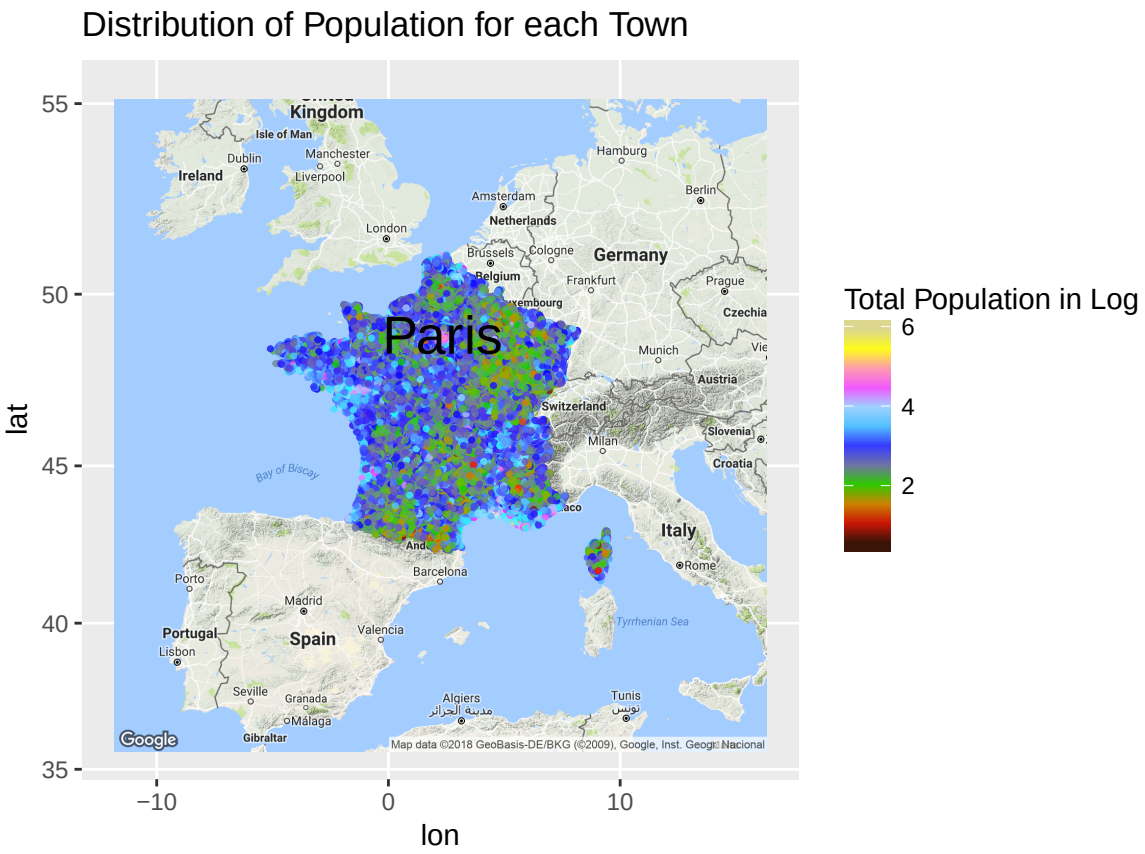

```r
population_per_town_data$total_population_log <- log10(population_per_town_data$total_population)

# Merge geo and pop
geo_pop_by_town <- merge(geo, population_per_town_data)
summary(geo_pop_by_town)
```

```
##      CODGEO                region       region_capital
##  Min.   : 1001   Midi-Pyrénées: 2980   Toulouse: 2980
##  1st Qu.:24510   Rhône-Alpes  : 2835   Lyon    : 2835
##  Median :48044   Lorraine     : 2323   Metz    : 2323
##  Mean   :46129   Aquitaine    : 2282   Bordeaux: 2282
##  3rd Qu.:67004   Picardie     : 2277   Amiens  : 2277
```

```
##  Max.   :97126   Bourgogne    : 2018   Dijon   : 2018
##                  (Other)      :21046   (Other) :21046
##           department             town_name      postal_code
##  Pas-de-Calais :  892   Paris         :   19   51300  :   46
##  Aisne         :  805   Sainte-Colombe:   14   51800  :   44
##  Somme         :  782   Saint-Sauveur :   12   70000  :   42
##  Moselle       :  729   Beaulieu      :   10   88500  :   42
##  Seine-Maritime:  718   Beaumont      :   10   80140  :   40
##  Côte-d'Or     :  705   Saint-Aubin   :   10   10200  :   36
##  (Other)       :31130   (Other)       :35686   (Other):35511
##     latitude       longitude       total_population       male
##  Min.   :41.39   Min.   :-5.0914   Min.   :      2   Min.   :      0
##  1st Qu.:45.13   1st Qu.: 0.7333   1st Qu.:    184   1st Qu.:     92
##  Median :47.37   Median : 2.6833   Median :    420   Median :    210
##  Mean   :46.96   Mean   : 2.7747   Mean   :   3024   Mean   :   1444
##  3rd Qu.:48.83   3rd Qu.: 4.8833   3rd Qu.:   1054   3rd Qu.:    524
##  Max.   :51.08   Max.   : 9.5167   Max.   :2173279   Max.   :1018597
##
##      female            child             elderly         workforce
##  Min.   :      0   Min.   :     0.0   Min.   :     0   Min.   :      0
##  1st Qu.:     90   1st Qu.:    32.0   1st Qu.:    35   1st Qu.:    112
##  Median :    208   Median :    80.0   Median :    75   Median :    260
##  Mean   :   1580   Mean   :   518.2   Mean   :   510   Mean   :   1996
##  3rd Qu.:    530   3rd Qu.:   208.0   3rd Qu.:   190   3rd Qu.:    655
##  Max.   :1154682   Max.   :315127.0   Max.   :338074   Max.   :1520078
##
##    dependent        sex_ratio       dependency_ratio aged_dependency_ratio
##  Min.   :     0   Min.   : 0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:    70   1st Qu.: 0.9074   1st Qu.:0.5132   1st Qu.:0.2094
##  Median :   160   Median : 0.9974   Median :0.6090   Median :0.2932
##  Mean   :  1028   Mean   : 1.0259   Mean   :0.6539   Mean   :0.3465
##  3rd Qu.:   399   3rd Qu.: 1.1029   3rd Qu.:0.7333   3rd Qu.:0.4167
##  Max.   :653201   Max.   :10.0000   Max.   :9.0000   Max.   :8.0000
##
##  child_dependency_ratio total_population_log
##  Min.   :0.0000         Min.   :0.301
##  1st Qu.:0.2450         1st Qu.:2.265
##  Median :0.3043         Median :2.623
##  Mean   :0.3073         Mean   :2.674
##  3rd Qu.:0.3661         3rd Qu.:3.023
##  Max.   :3.0000         Max.   :6.337
##
```

```
# Plot "Distribution of Population for each Town"
#myPalette(low = "white", high = c("green", "red"), mid=NULL, k =50)-Need "GLAD" package
sc <- scale_colour_gradientn(colours =palette(rainbow(8)), limits=c(min(geo_pop_by_town$total_populatio
poppulation_distribution <-
  FraMap +
  geom_point(aes(x=geo_pop_by_town$longitude, y=geo_pop_by_town$latitude, colour=geo_pop_by_town$total_
             data=geo_pop_by_town, alpha=0.8, size=0.6) +
  sc +
  geom_text(aes(label = town_name, x = longitude, y = latitude),
            data = subset(geo_pop_by_town, total_population_log %in% head(sort(total_population_log, de
            check_overlap = TRUE, size=7) +
```

```
  labs(color='Total Population in Log') +
  ggtitle("Distribution of Population for each Town")
poppulation_distribution
```




```
# Group population data by department because of small size of some towns and the given geojson file of
pop_by_department <- ddply(geo_pop_by_town, .(department), function(geo_pop_by_town) {
  data.frame(total_population = sum(geo_pop_by_town$total_population),
             male = sum(geo_pop_by_town$male),
             female = sum(geo_pop_by_town$female),
             child = sum(geo_pop_by_town$child),
             elderly = sum(geo_pop_by_town$elderly),
             dependent = sum(geo_pop_by_town$dependent),
             workforce = sum(geo_pop_by_town$workforce)
  )})

summary(pop_by_department)
```

```
##                   department total_population        male
##  Ain                    : 1   Min.   :    9048   Min.   :    4353
##  Aisne                  : 1   1st Qu.:  268814   1st Qu.:  131022
##  Allier                 : 1   Median :  512671   Median :  250784
##  Alpes-de-Haute-Provence: 1   Mean   : 1114958   Mean   :  532357
##  Alpes-Maritimes        : 1   3rd Qu.:  782721   3rd Qu.:  383158
##  Ardèche                : 1   Max.   :41292301   Max.   :19353343
##  (Other)                :91
##      female             child            elderly          dependent
```

```
##  Min.   :    4695   Min.   :   1700   Min.   :   2076   Min.   :    3776
##  1st Qu.:  137792   1st Qu.:  50051   1st Qu.:  55801   1st Qu.:  107317
##  Median :  261218   Median :  91961   Median :  92779   Median :  187742
##  Mean   :  582601   Mean   : 191059   Mean   : 188003   Mean   :  379062
##  3rd Qu.:  399563   3rd Qu.: 156655   3rd Qu.: 146183   3rd Qu.:  287695
##  Max.   :21938958   Max.   :5987413   Max.   :6423406   Max.   :12410819
##
##    workforce
##  Min.   :    5272
##  1st Qu.:  168468
##  Median :  324657
##  Mean   :  735896
##  3rd Qu.:  505345
##  Max.   :28881482
##
```

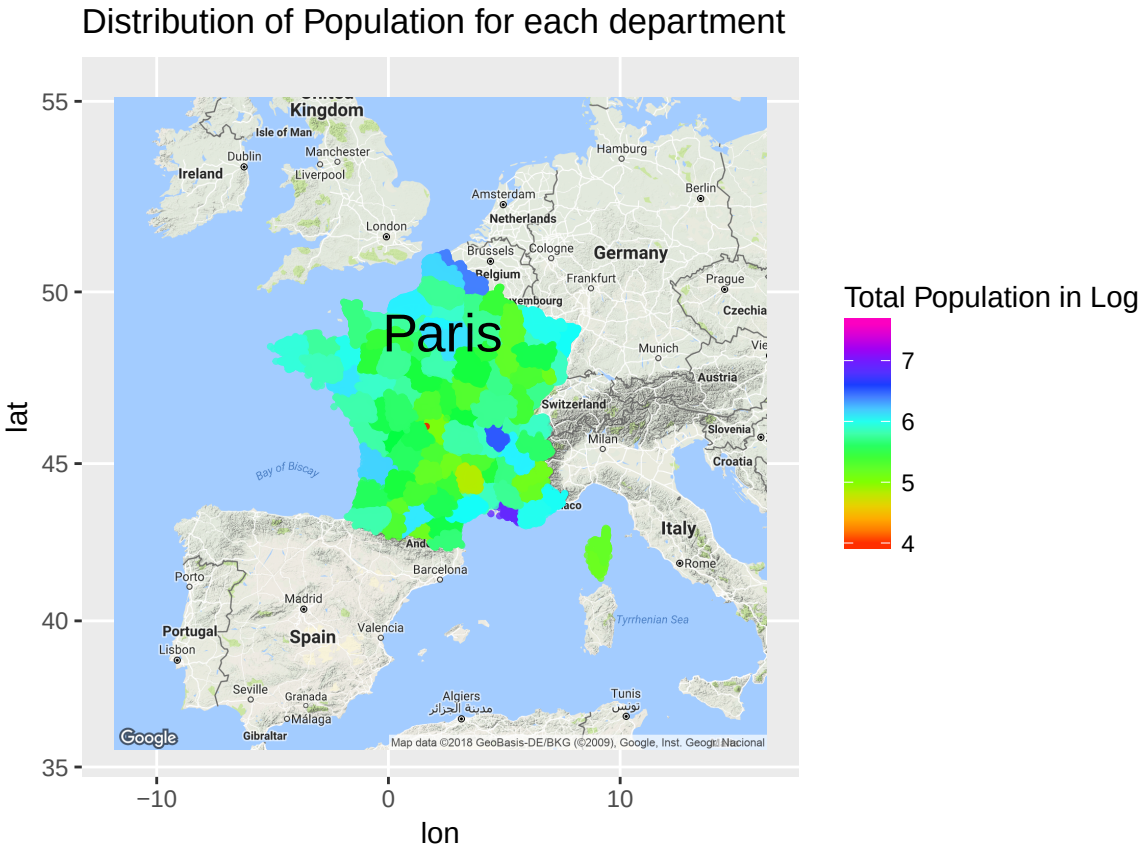
```
pop_by_department$dependency_ratio <- pop_by_department$dependent / pop_by_department$workforce
pop_by_department$aged_dependency_ratio <- pop_by_department$elderly / pop_by_department$workforce
pop_by_department$child_dependency_ratio <- pop_by_department$child / pop_by_department$workforce
```

```
# Scale population to log
pop_by_department$total_population_log <- log10(pop_by_department$total_population)

# Merge geo and pop
geo_pop_by_department <- merge(geo, pop_by_department)
summary(geo_pop_by_department)
```

```
##           department              region       region_capital
##  Pas-de-Calais :  894   Midi-Pyrénées: 3021   Toulouse: 3021
##  Aisne         :  816   Rhône-Alpes  : 2882   Lyon    : 2882
##  Somme         :  782   Lorraine     : 2333   Metz    : 2333
##  Seine-Maritime:  745   Aquitaine    : 2296   Bordeaux: 2296
##  Moselle       :  730   Picardie     : 2291   Amiens  : 2291
##  Côte-d'Or     :  707   Bourgogne    : 2046   Dijon   : 2046
##  (Other)       :31920   (Other)      :21725   (Other) :21725
##           town_name      postal_code        CODGEO         latitude
##  Paris         :   19   51300  :   46   Min.   : 1001   Min.   :41.39
##  Sainte-Colombe:   14   51800  :   44   1st Qu.:24527   1st Qu.:45.17
##  Saint-Sauveur :   12   70000  :   42   Median :48111   Median :47.40
##  Beaulieu      :   11   88500  :   42   Mean   :46096   Mean   :46.98
##  Saint-Sulpice :   11   80140  :   40   3rd Qu.:66169   3rd Qu.:48.83
##  Beaumont      :   10   02160  :   38   Max.   :97126   Max.   :51.08
##  (Other)       :36517   (Other):36342
##    longitude       total_population        male
##  Min.   :-5.0914   Min.   :    9048   Min.   :    4353
##  1st Qu.: 0.6667   1st Qu.:  317665   1st Qu.:  153234
##  Median : 2.6333   Median :  529618   Median :  257449
##  Mean   : 2.7380   Mean   :  694650   Mean   :  336127
##  3rd Qu.: 4.8667   3rd Qu.:  776291   3rd Qu.:  378636
##  Max.   : 9.5167   Max.   :41292301   Max.   :19353343
##
##      female             child            elderly          dependent
##  Min.   :    4695   Min.   :   1700   Min.   :   2076   Min.   :    3776
##  1st Qu.:  162818   1st Qu.:  54952   1st Qu.:  66927   1st Qu.:  125937
##  Median :  272169   Median : 101471   Median :  93119   Median :  193520
```

```
##  Mean   :  358523   Mean   : 129285   Mean   : 122310   Mean   :  251595
##  3rd Qu.:  397655   3rd Qu.: 156655   3rd Qu.: 143873   3rd Qu.:  278973
##  Max.   :21938958   Max.   :5987413   Max.   :6423406   Max.   :12410819
##
##    workforce        dependency_ratio aged_dependency_ratio
##  Min.   :    5272   Min.   :0.4297   Min.   :0.1636
##  1st Qu.:  189820   1st Qu.:0.5571   1st Qu.:0.2606
##  Median :  326192   Median :0.6023   Median :0.3071
##  Mean   :  443055   Mean   :0.6014   Mean   :0.3101
##  3rd Qu.:  496202   3rd Qu.:0.6458   3rd Qu.:0.3570
##  Max.   :28881482   Max.   :0.7162   Max.   :0.4536
##
##  child_dependency_ratio total_population_log
##  Min.   :0.2073         Min.   :3.957
##  1st Qu.:0.2708         1st Qu.:5.502
##  Median :0.2920         Median :5.724
##  Mean   :0.2913         Mean   :5.710
##  3rd Qu.:0.3054         3rd Qu.:5.890
##  Max.   :0.3468         Max.   :7.616
##
```

```
# Plot "Distribution of Population for each department"
#myPalette(low = "white", high = c("green", "red"), mid=NULL, k =50)-Need "GLAD" package
sc <- scale_colour_gradientn(colours =palette(rainbow(8)), limits=c(min(geo_pop_by_department$total_popu
pop_distribution_department <-
  FraMap +
  geom_point(aes(x=geo_pop_by_department$longitude, y=geo_pop_by_department$latitude, colour=geo_pop_by_
             data=geo_pop_by_department, alpha=0.8, size=0.6) +
  sc +
  geom_text(aes(label = town_name, x = longitude, y = latitude),
            data = subset(geo_pop_by_department, total_population_log %in% head(sort(total_population_lo
            check_overlap = TRUE, size=7) +
  labs(color='Total Population in Log') +
  ggtitle("Distribution of Population for each department")
pop_distribution_department
```



Produce consistent datasets

```
# use only integer values
geo$CODGEO = as.integer(geo$CODGEO)  # already integer
population$CODGEO = as.integer(population$CODGEO)
firms$CODGEO = as.integer(firms$CODGEO)
salary$CODGEO = as.integer(salary$CODGEO)

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

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
```

```
##     intersect, setdiff, setequal, union
```

```r
dataset = c("population", "salary", "firms", "geo")

# obtain sommon IDs for all datasets
for (i in dataset){
  # get i-th name and make a new variable adding NEW
  nam <- paste(i, "NEW", sep = "")
  # counter to identify the number of iteration in j
  iter = 1
  for (j in dataset){
    if (j != i){
      # datasets different from i-th
      if (iter == 1){
        # 1st iteration: use the original dataset (e.g., geo)
        assign(nam, semi_join(get(i), get(j), by = "CODGEO"))
      } else{
        # successive iteration: use the new dataset (e.g., geoNEW)
        assign(nam, semi_join(get(nam), get(j), by = "CODGEO"))
      }
      iter = iter + 1
    }
  }
}

# check how many observation have been deleted
for (i in dataset){
  del_rows = nrow(get(i)) - nrow(get(paste(i, "NEW", sep = "")))
  del_prop = del_rows / nrow(get(paste(i, "NEW", sep = "")))
  del_obs = paste("For", i, del_rows, "have been deleted.",
                  "They were the", round(del_prop, digits=2), "% of the total.", sep = " ")
  print(del_obs)
}
```

```
## [1] "For population 1521935 have been deleted. They were the 2.21 % of the total."
## [1] "For salary 113 have been deleted. They were the 0.02 % of the total."
## [1] "For firms 31658 have been deleted. They were the 6.3 % of the total."
## [1] "For geo 31543 have been deleted. They were the 6.24 % of the total."
```

Analysis

PCA

Regression

Clustering