Data exploration and preliminary feature selection

Fabio Morea

2022-02-01

Contents

1 Scope and objectives **3**
1.1 Background information: 3
1.2 Objectives of future research work - reserch questions 4
1.3 About this notebook 4
1.4 Data management plan 5

2 Exploring dataset "companiesFVG" **6**
2.1 companies 6
2.2 NACE activity codes 16
2.3 tidy dataset 18

3 Exploring dataset "bilanciFVG" **19**

4 Exploring dataset CO-FVG **24**

1 Scope and objectives

This notebook explores the datasets that will presumably underpin future research work for the PhD in Applied Data Science and Artificial Intelligence.

1.1 Background information:

Research, innovation and highly skilled people are considered to be important factors in economic and social development. Economic support policies often include funds to support research (for example with the creation of public research infrastructures), companies (for example with tenders to co-finance innovative projects) and the training of people with the necessary skills.

Area Science Park is a national research institution that manages a science and technology park located in Trieste (Italy). Its activities can be considered a public investment in support of research and innovation, for a value of approximately 20 million euros per year.

Currently Area is hosting 70 tenants (60 companies and 10 research centers) engaged in research activities in the fields of ICT, lifesceinces and materials. Their success (or lack of it) depends on a key - and often overlooked - asset: the community of over 1600 employees, researchers and entrepreneurs.

Area is interested in measuring the effectiveness and efficiency of its activities, focusing in particular on

- monitoring the economic performance of tenants,
- monitoring the community of skilled workers,
- comparing with similar groups, mainly at a regional or national scale, but also with respect to the science and technology parks in Austria and Slovenia.

To support research work, Area Science Park can provide some relevant datasets, cureated as a part of the project *innovation intelligence*. Innovation Intelligence aims to analyze information on companies in the Friuli Venezia Giulia region, using several data sources such as the chamber of commerce, the Regional Labor Market Observatory, a rating agency, as well as surveys on samples of companies.

1.2 Objectives of future research work - reserch questions

Research questions are currently defined on a general level:

- are tenant companies performing better than similar companies?
- how to measure similarity between two companies?
- how to exthed such measure to groups of companies?
- how to identify clusters or communities of companies?

The research questions above and the methodology outlined in this notebook are relevant also in other contexts, such ad sectoral cluster, public agencies supporting innovation and any kind of industrial area. The data set supports analysis focused on Friuli Venezia Giulia region, but can be extended to other regions (gathering relevant data from the Chamber of Commerce or from commercial data providers).

1.3 About this notebook

The notebook is divided in 6 sections: an introduction, a section for each dataset and a final section on potential future development.

1. Imprese_FVG
2. Bilanci_FVG
3. Rating_FVG
4. CO_FVG
5. Features: A basic example of sample feature selection, on a small subset, where each company is represented by 5 features
6. Further development: calculating the age of companies based on several dates, handling non metric features: defining a custimized similarity function to identify *similar* companies and estimate distances in a multi-dimensional space.

The notebook has been written using *R-Studio* and rendered with *boowdown* (https://bookdown.org/) package. Data data manipulation is based on *tidyverse* [https://www.tidyverse.org/], a data science library that includes *magrittr* (pipe operator %>%), *dplyr* (select, summarize...), *tibble* (a tidier version of the data.frame) and *ggplot2* (visualizations). A useful guide to tidverse is available online at the following address: [https://r4ds.had.co.nz/]

> TODO Some parts of the notebook are higlighted as "To Do", to highlight potential improvements in analysis, code efficiency or need for further clarifications.

1.4 Data management plan

Raw data: The original data has been pre-pocessed by Area Science Park to fulfill the folloqing rquirements:

- encoded in UTF-8 cleaned from non-printable characters
- table columns are attributes (features, independent variables), renamed to be human- and machine-readable
- table rows are observations If you have multiple tables, they should include a column in the table that allows them to be linked
- splitted into several tables, created unique identifiers to connect the tables
- saved each table to separate .csv file with a hunam-readable name At this stage no attributes were removed or summarized. Raw data is available in local folder *data/raw*

Tidy data: This notebook explores all the attributes available in the raw data, and by merging, subsetting and transforming, produces a smaller, cleaner data set ready for further analysis. Tidy data is saved in local folder *data/tidy*

This notebook describes the process to create tidy data, and the meaning of each variable: - meaning, summary and visualizations ofeach attribute in tidy data - information about attributes that are not contained in the tidy data (basic meaning and reason why thy have not been included)

Updates: raw and tidy data may be updated periodically, since Area Science Park updates the raw data set twice a year; anyway the current version in based on June 2021 version and does not provide automatic updating scripts.

2 Exploring dataset “companiesFVG”

This section is deditated to the exploration of data on companies that have their premises in Friuli Venezia Giulia. The origin of data is the Italian Business registry (https://www.registroimprese.it/il-registro-imprese-per-la-p.a.), and is managed by Infocamere (https://www.infocamere.it/).

```
data.files <- list.files(pathRawData, pattern = ".csv$", recursive = TRUE)
```

Pre-processed is availavle in folder **/data/raw**, organized in 10 files (namely bilanci-fvg.csv, d_ateco.csv, d_ng.csv, id_imp_loc.csv, pseudo_cf_id_impresa.csv, t_attivita.csv, t_codici.csv, t_imprese.csv, t_imprese_dp.csv, t_localizz.csv). Each will be loaded in a different *data.frame* and, after feature selection, saved in a new folder **/data/tidy**.

2.1 companies

The core data identifying companies can be found in *t_imprese.csv* .

```
companies <- read_delim( paste0(pathRawData,"/t_imprese.csv"))
```

```
## Rows: 108379 Columns: 34
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "|"
## chr  (19): fonte, mm_aaaa, denominazione, cf, piva, prov, reg_imp_n, sede_ul...
## dbl   (5): id_impresa, addetti_aaaa, addetti_indip, addetti_dip, capitale
## lgl   (2): data_canc, imp_sedi_ee
## date  (8): data_cost, data_isc_ri, data_isc_rd, data_isc_aa, data_ini_at, da...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message
```

```
spec(companies) # tydiverse for str(companies)
```

```
## cols(
##   fonte = col_character(),
##   mm_aaaa = col_character(),
##   id_impresa = col_double(),
##   denominazione = col_character(),
##   cf = col_character(),
##   piva = col_character(),
##   prov = col_character(),
##   reg_imp_n = col_character(),
##   sede_ul = col_character(),
##   `n-albo_art` = col_character(),
##   reg_imp_sez = col_character(),
##   ng2 = col_character(),
##   stato_impresa = col_character(),
##   data_cost = col_date(format = ""),
##   data_isc_ri = col_date(format = ""),
##   data_isc_rd = col_date(format = ""),
##   data_isc_aa = col_date(format = ""),
##   data_canc = col_logical(),
##   data_ini_at = col_date(format = ""),
##   data_cess_att = col_date(format = ""),
##   data_fall = col_date(format = ""),
##   data_liquid = col_date(format = ""),
##   addetti_aaaa = col_double(),
##   addetti_indip = col_double(),
##   addetti_dip = col_double(),
##   capitale = col_double(),
##   capitale_valuta = col_character(),
##   imp_sedi_ee = col_logical(),
##   imp_eefvg = col_character(),
##   imp_pmi = col_character(),
##   imp_startup = col_character(),
##   imp_femminile = col_character(),
##   imp_giovanile = col_character(),
##   imp_straniera = col_character()
## )
```

The attributes belong to different groups:

- *metadata*:ï..fonte, mm_aaaa:
- *identifier*: id_impresa,reg_imp_n, cf, piva, denominazione

- *address*:prov,sede_ul,n.albo_art,reg_imp_sez

- *type of company*: ng2
- *active status*: stato_impresa

- *dates*:data_ini_at, data_cess_att, data_fall, data_liquid, data_cost, data_isc_ri, data_isc_rd,data_isc_aa,data_canc
- *employees*: addetti_aaaa, addetti_indip, addetti_dip

- *share capital*: capitale, capitale_valuta
- *other attributes*: imp_startup, imp_femminile, imp_giovanile, imp_straniera, imp_pmi, imp_sedi_ee, imp_eefvg

2.1.1 Metadata

Metadata are generated by the pre-rpcessing algorithm and provide information about source and last update. Current version is 06_2021. The two attributes (ï..fonte, mm_aaaa) are not relevant for further analysis, and can be removed from the tidy dataset.

```
companies <- companies %>% select( !c(fonte, mm_aaaa))
```

2.1.2 Identifiers

Each company can be identified by its name, vat number, fiscal code, The following attributes will be used in: - denominazione: company name (not a unique identifier, can be spelled in different ways across datasets; moreover different companies may have the same name); - cf ("codice fiscale"): unique identifier, as factor. Values are unique for each company, but the structure depends on company type: generally a string of 11 digits, but individual companies refer to the owner's code, a string of 16 letters and digits; - idCompany ("id_impresa"): unique identifier, numeric, created in pre-processing phase. Other attributes (reg_imp_n,piva, n.albo_art,reg_imp_sez) are not relevant at this stage, and can be dropped. All identifiers will be converted to factors.

```
companies <- companies %>%
             select( !c(reg_imp_n, piva, `n-albo_art`,reg_imp_sez)) %>%
             rename(name = denominazione) %>%
             rename(idCompany = id_impresa) %>%
             mutate_if(is.character, as.factor)
```

Now we can ckeck if there are any missing values or duplicates

```
# check missing calues
sum(is.na(companies$name)) + sum(is.na(companies$cf)) == 0
```

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

```
# check duplicates in cf
length(unique(companies$cf)) == length(companies$cf)
```

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

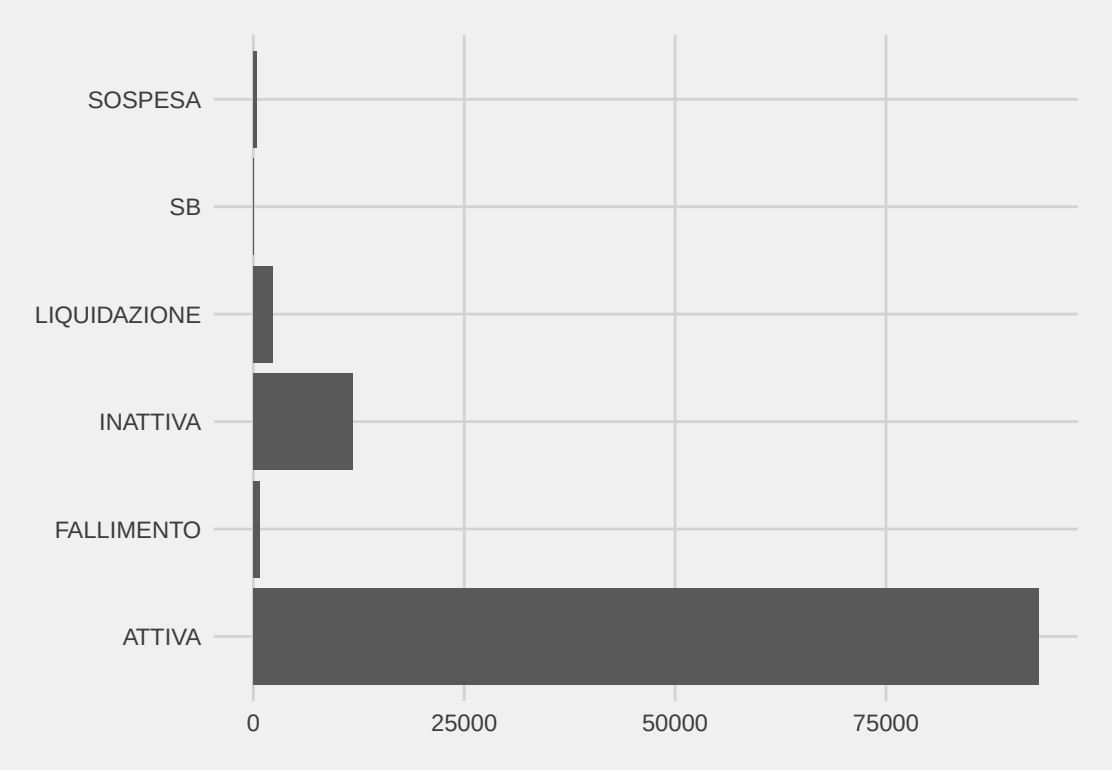
```
# check duplicates in name
uniqueNames <-length(unique(companies$name))
allNames<-length(companies$name)
print(paste("Company names are not a valid identifier for further analysis: the dataset contai
```

```
## [1] Company names are not a valid identifier for further analysis: the dataset contains 107
```

2.1.3 active status

Companies that are not active (e.g. due to bankruptcy, liquidation or suspended) are not relevant for the research objectives and can be removed from the dataset.

```
df<-companies %>% count(stato_impresa)
plot1 <- ggplot(data=df, aes(x=stato_impresa, y=n)) +   geom_bar(stat="identity") + coord_flip
plot1
```




```
companies <- subset(companies, stato_impresa =='ATTIVA')
print(paste("Number of active companies: ", nrow(companies)), quote=FALSE)
```

```
## [1] Number of active companies:  93106
```

2.1.4 head office and local units

Each company has a "registered office" (sede legale) and may have several local units (unità locale). Relevant data is stored in file "t_localizz.csv". > TODO: add description of variables. Select only companies that are located in Friuli Venezia Giulia, according to prov: province (GO, TS, UD, PN). Select companies that have head office abroad. sede_ul: "SEDE" or "UL-n" » factor SEDE = HeadOffice / UL = LocalUnit Extract the number of local units from attribute "sede_ul" Create new variable "head-office" true/false use "data apertura ul" to improve the estimate of company age (or remove the attribute) remove unnecessary variables

```
locs <- read_delim( paste0(pathRawData,"/t_localizz.csv")) %>%
             select( c(id_localiz, id_impresa, denominazione,tipo_localizzazione)) 
             rename(name = denominazione) %>%
```

```
              rename(idCompany = id_impresa) %>%
              mutate_if(is.character, as.factor)
```

```
## Rows: 205385 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "|"
## chr  (7): fonte, denominazione, tipo_localizzazione, prov_localiz, comune, i...
## dbl  (2): id_localiz, id_impresa
## dttm (1): data_apert_ul
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

2.1.5 Legal form of companies

The legal form of companies can be a relevant attribute for further research. A primary distinction should be made between between

- *limited liability companies* (società di capitali), mainly Private Limited Companies by Quotas (società a responsabilità limitata or S.r.l.), Simplified S.r.l. (S.r.l.s.), Public Limited Companies by Shares (società per azioni or S.p.A.). Limited liability companies disclose their financial statements, therefore more relevant information is available.
- *partnerships* (società di persone), mainly Società in nome collettivo (S.n.c.) and Società in accomandita semplice or (S.a.s.)

Legal form is encoded in the variable companies$ng, and codes are described in a separate file */d_ng.csv*. There are over 50 different codes, but the great majority of companies belong to a small number of types. For the purpose of data exploration, the following figure hignliths thee 10 most common legal forms.

```
# company type: keep only the relevan ones for the scope of our research.
types <- read.csv( paste0(pathRawData,"/d_ng.csv"), sep = "|")
companies$ng2           <- as.factor(companies$ng2)
names(types)<-c("ngGroup", "ng2", "ngDescription")

df <- companies  %>% count(ng2)
df <- df %>% inner_join(types)
```

```
## Joining, by = "ng2"
```

```
df <- df  %>% arrange(-n) %>% head(10)
df$ngDescription <- factor(df$ngDescription, levels = df$ngDescription)
df
```

```
## # A tibble: 10 x 4
##    ng2       n ngGroup              ngDescription
##    <chr> <int> <chr>                <fct>
##  1 DI    52751 IMPRESE INDIVIDUALI  IMPRESA INDIVIDUALE
##  2 SR    16836 SOCIETA' DI CAPITALE SOCIETA' A RESPONSABILITA' LIMITATA
##  3 SN     7209 SOCIETA' DI PERSONE  SOCIETA' IN NOME COLLETTIVO
##  4 AS     6457 SOCIETA' DI PERSONE  SOCIETA' IN ACCOMANDITA SEMPLICE
##  5 SU     2139 SOCIETA' DI CAPITALE SOCIETA' A RESPONSABILITA' LIMITATA CON UNI~
##  6 RS     2116 SOCIETA' DI CAPITALE SOCIETA' A RESPONSABILITA' LIMITATA SEMPLIF~
##  7 SE     2020 SOCIETA' DI PERSONE  SOCIETA' SEMPLICE
##  8 SP     1125 SOCIETA' DI CAPITALE SOCIETA' PER AZIONI
##  9 SC      726 ALTRE FORME          SOCIETA' COOPERATIVA
## 10 AC      543 ALTRE FORME          ASSOCIAZIONE
```

Some company types are not relevant for our research, for example individual companies (DI) and other specified below. Dropping the corresponding dataframe rows drastically reduces the size of the data set

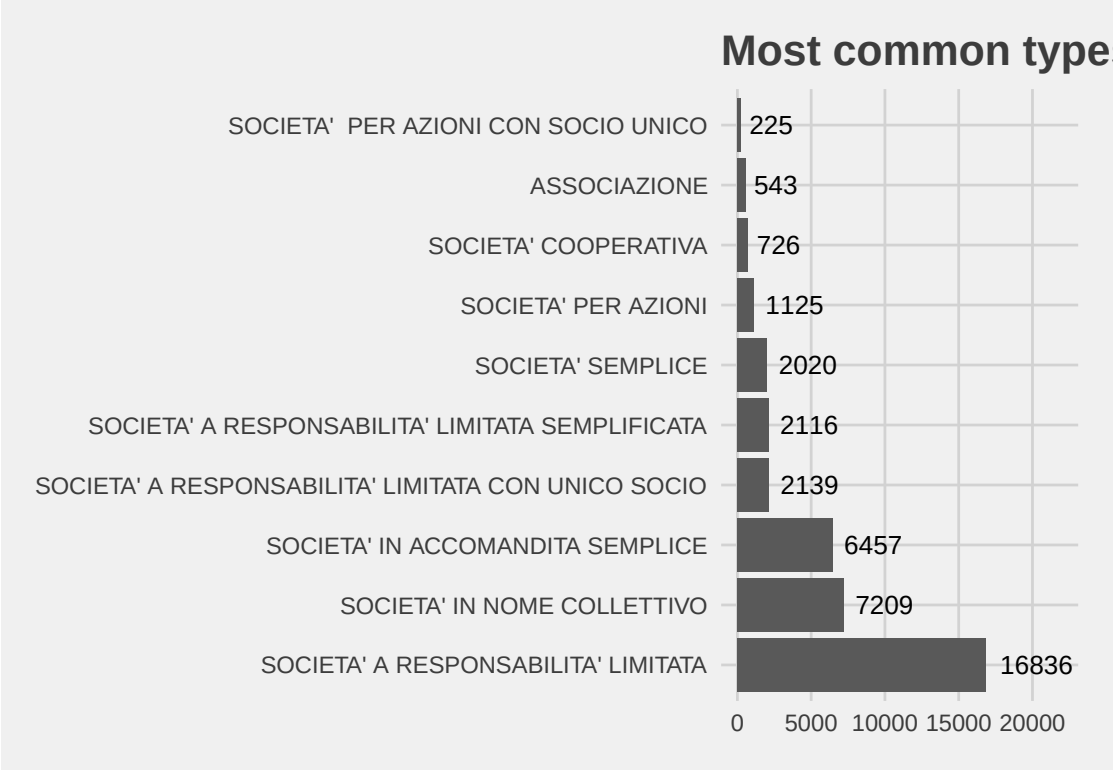
```
notRelevant = c("DI", "AZ", "IR", "ER", "EP", "EN", "EM", "EL", "EE", "SM", "MA", "
toBeRemoved<-which(companies$ng2 %in% notRelevant)
companies<-companies[-toBeRemoved,]

df <- companies  %>% count(ng2)
print(paste("The dataset contains ", nrow(df), "types of companies."), quote=FALSE)
```

```
## [1] The dataset contains  34 types of companies.
```

```
df <- df %>% inner_join(types)
```

```
## Joining, by = "ng2"
```

```
df <- df  %>% arrange(-n)
df$ngDescription <- factor(df$ngDescription, levels = df$ngDescription) #lock facto
p3<-ggplot(data=head(df, 10), aes(x=ngDescription, y=n))  + geom_bar(stat="identity
p3
```



2.1.6 dates, age of companies, years in business

The dataset provides relevant information in the form of dates.

```
companies %>% select(starts_with("data")) %>% summary(is.na())
```

```
##    data_cost          data_isc_ri         data_isc_rd
##  Min.   :1807-01-01   Min.   :1996-02-08   Min.   :1856-04-26
##  1st Qu.:1992-12-31   1st Qu.:1996-02-19   1st Qu.:1994-03-28
##  Median :2005-01-11   Median :2005-11-07   Median :2005-10-20
##  Mean   :2002-02-24   Mean   :2006-05-16   Mean   :2002-12-06
##  3rd Qu.:2014-03-25   3rd Qu.:2014-12-01   3rd Qu.:2014-10-31
##  Max.   :2021-05-28   Max.   :2021-06-01   Max.   :2021-05-31
##  NA's   :1229         NA's   :805          NA's   :5
##   data_isc_aa         data_canc       data_ini_at         data_cess_att
##  Min.   :1937-10-24   Mode:logical   Min.   :1856-04-26   Min.   :NA
##  1st Qu.:1989-07-06   NA's:40211     1st Qu.:1994-02-01   1st Qu.:NA
##  Median :2002-04-19                  Median :2006-01-04   Median :NA
##  Mean   :2000-01-31                  Mean   :2003-01-08   Mean   :NA
##  3rd Qu.:2012-02-01                  3rd Qu.:2014-09-20   3rd Qu.:NA
```

```
##  Max.   :2021-06-04                  Max.   :2105-02-28   Max.   :NA
##  NA's   :32392                       NA's   :1041         NA's   :40211
##    data_fall           data_liquid
##  Min.   :1982-10-01   Min.   :2009-06-15
##  1st Qu.:1997-07-03   1st Qu.:2010-12-25
##  Median :2004-02-02   Median :2012-10-08
##  Mean   :2003-10-16   Mean   :2013-11-12
##  3rd Qu.:2008-11-05   3rd Qu.:2014-09-30
##  Max.   :2021-05-18   Max.   :2019-12-20
##  NA's   :39671        NA's   :40194
```

The most relevant information for further research is the *age* of the company, that can be assessed as the number of years in business, i.e. the number of years from the earliest date of registration. There are several registration ates (data_iscr_*) and an official start date (data_ini_at), however each attribute has several missing values (NA).

```
#create new atribute "years in business"
companies <- companies %>%
  rowwise() %>%
  mutate(dateMin = min(data_isc_ri, data_isc_rd, data_isc_aa, data_ini_at,  na.rm=T
  mutate(yearsInBusiness = as.numeric(as.Date("2022-01-01") - dateMin) / 365  )
```

The only attribute to keep for further analysis is *yearsInBusiness*; all *date* columns can be removed from the tidy dataset.

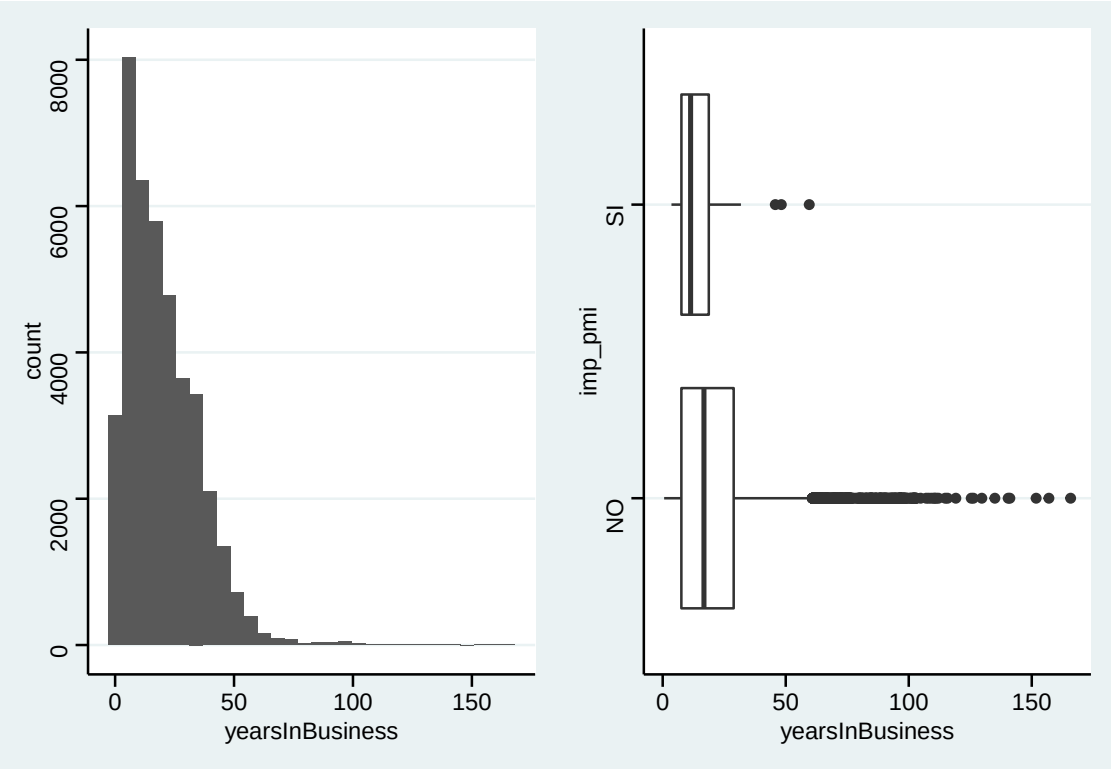
```
#remove all "date" sttributes
companies <- companies %>% select( !starts_with("data"))
```

The distribution of yearsInBusiness is shown below. A

```
plot1<-ggplot(data=companies, aes(x=yearsInBusiness), show.legend = FALSE) +
        geom_histogram()
plot2<-ggplot(data=companies, mapping=aes(x=imp_pmi, y=yearsInBusiness))+
        geom_boxplot()  + coord_flip()

labels = c("years in business", "focus on SMEs")
figure <- ggarrange(plot1, plot2)#,labels = labels, ncol = 1, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
figure
```



Information on bankarupcy should be redundant, since non-active companies have been removed; nevertheless, past check showed unconistend data in the original tables. If bancarupcy dates are present, the company is considered as non-actie and removed.

2.1.7 employees

addetti_aaaa, addetti_indip, addetti_dip

2.1.8 share capital

capitale, capitale_valuta

2.1.9 other attributes

TODO other attributes are relevant as signs of innovation imp_startup, imp_femminile, imp_giovanile, imp_straniera, imp_pmi, imp_sedi_ee,

imp_eefvg
show distribution for each

```
companies$imp_pmi       <- as.factor(companies$imp_pmi)
companies$imp_startup   <- as.factor(companies$imp_startup)
companies$imp_giovanile <- as.factor(companies$imp_giovanile)
companies$imp_femminile <- as.factor(companies$imp_femminile)
companies$imp_straniera <- as.factor(companies$imp_straniera)
companies$imp_sedi_ee   <- as.factor(companies$imp_sedi_ee)
```

2.2 NACE activity codes

Each company is associated with a list of activity codes, according to NACE classification. The corresponding information is available in *t_codici.csv* . As in previous files, metadata can be inored at this stage. More info: Complete list of all NACE Code NACE (Nomenclature of Economic Activities) is the European statistical classification of economic activities. NACE groups organizations according to their business activities. [https://nacev2.com/en]

```
NACECodes <- read_delim( paste0(pathRawData,"/t_codici.csv"))  %>%
  select(-c(fonte, mm_aaaa))
```

```
## Rows: 409713 Columns: 6
```

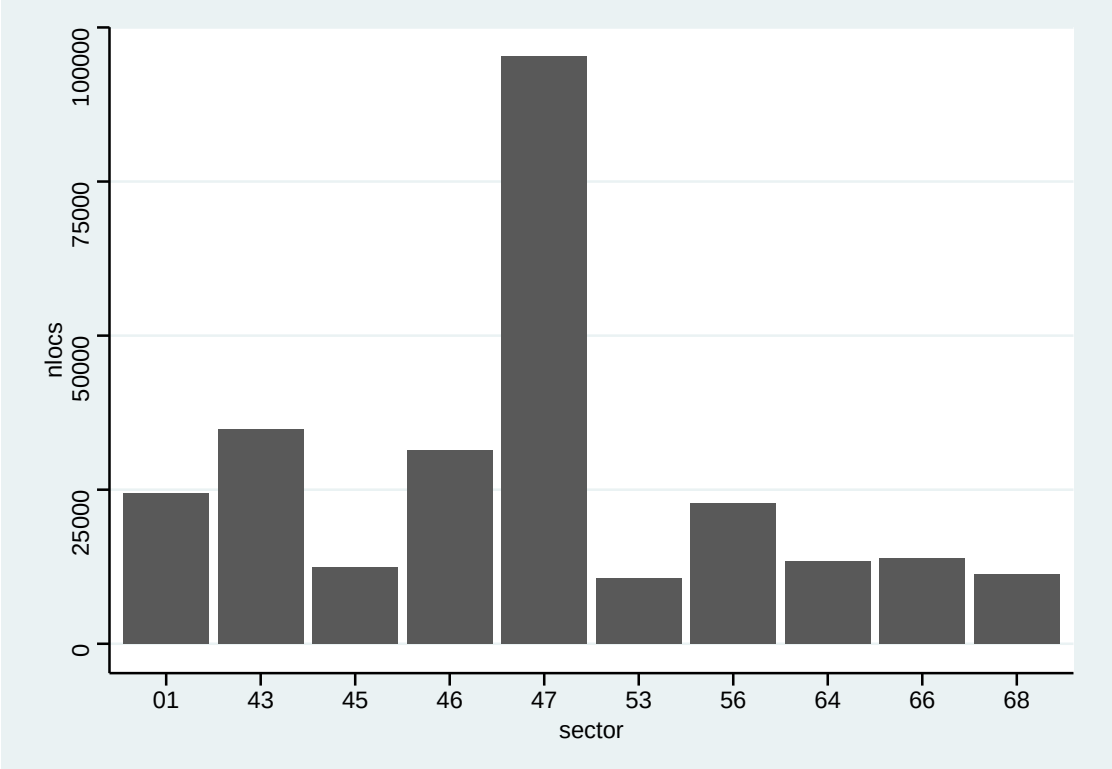
```
## -- Column specification --------------------------------------------------------
## Delimiter: "|"
## chr (4): fonte, mm_aaaa, ateco_tipo, ateco
## dbl (2): id_localiz, loc_n
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message
```

```
noCode <- NACECodes %>% filter(is.na(ateco))
percNA <- round(nrow(noCode)/nrow(NACECodes)*100, 2)
compNoCode <- noCode %>% distinct()
percCompNA <- round(nrow(compNoCode)/nrow(companies)*100,2)
```

The dataset contains 7339 companies, and 40211 NACE codes: each company may have one or more NACE codes, of different types: I (prevealente), P (Primario) and S (Secondario). Unicity is not guaranteed for any type (despite "I" codes are supposed to be unique, in fact many companies have several codes).

```
# TODO most common sectors (a sector is identified by first two digits of nacecode)
NACECodes["sector"]<-substr(NACECodes$ateco,start=1,stop=2)
df <- NACECodes %>% group_by(sector) %>% summarise( nlocs = n())  %>% arrange(desc(nlocs))   %
p<-ggplot(data=df, aes(x=sector, y=nlocs)) +  geom_bar(stat="identity")
p
```



A company can be **associated with a single NACE code** trough an algorithm, than is to some extent subjective.

Some companies have **no NACE codes associated**: this is an issue with the original data affecting a small portion of the dataset (in this example 7339 missing values out of 409713, or 1.79%). Since each company may have more codes, the number of companies without *any NACE code* is smaller: only 4 out of 40211 i.e. 18.09%. Rows with NAs are removed from the dataset.

Moreover, NACE codes are **associated with company localization**. In order to reduce the compexity, we may summarize NACE codes by company, but this can lead

to misinterpretation in some cases. Consider, for example, a company has its head office in Rome, with several codes in sector 47, and a local unit in Trieste engaged in different activities with a single code in sector 22. If we summarise codes by company our case study belongs to sector 47; instead, if we summarize by localization, the company belongs to sector 22.

For the purpose of data elxploration and preliminary feature selection, we filer companies and locs to retain only rows that are associated with NACE codes.

```
loc_in_NACE <- NACECodes %>% select(id_localiz) %>% distinct()   #unique id_localiz
locs <- locs %>% filter(id_localiz %in% loc_in_NACE$id_localiz)  #filter locs dataf
companies <- companies %>% filter(idCompany %in% locs$idCompany) #filter companies
```

"

2.3 tidy dataset

Now we can save the filtered and cleaned data to a csv. The number of features is reduced to... The dataset is composed of .. .main files (t_cmp.csv, t_nace.csv) and ... files with extended descriptions (d_ng, d_nace).

```
companies %>% write_csv(paste0(pathTidyData,"t_cmp.csv"),)
```

3 Exploring dataset “bilanciFVG”

This section is dedicated to load and preprocess financial statement data for the dataset *imprese-fvg*. The relevant file is “_DATA/imprese-fvg/bilanci-fvg.csv”.

The relevant file is *bilanci-fvg.csv*. Each observation is a summary of balance sheet data (bsd) of a company (identified by *cf*) for a given year. Column labels need some improvement to remove whitespaces and possibly short english names.

```
bsd <- read_delim( paste0(pathRawData,"imprese/bilanci-fvg.csv") )
```

```
## Rows: 125617 Columns: 18
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (16): cf, cia, Totale attivo, Totale Immobilizzazioni immateriali, Credi...
## dbl  (2): rea, anno
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
spec(bsd) # tydiverse for str(companies)
```

```
## cols(
##   cf = col_character(),
##   cia = col_character(),
##   rea = col_double(),
##   anno = col_double(),
##   ‘Totale attivo‘ = col_character(),
##   ‘Totale Immobilizzazioni immateriali‘ = col_character(),
##   ‘Crediti esigibili entro l’esercizio successivo‘ = col_character(),
##   ‘Totale patrimonio netto‘ = col_character(),
##   ‘Debiti esigibili entro l’esercizio successivo‘ = col_character(),
##   ‘Totale valore della produzione‘ = col_character(),
##   ‘Ricavi delle vendite‘ = col_character(),
```

```
##   `Totale Costi del Personale` = col_character(),
##   `Differenza tra valore e costi della produzione` = col_character(),
##   `Ammortamento Immobilizzazione Immateriali` = col_character(),
##   `Utile/perdita esercizio ultimi` = col_character(),
##   `valore aggiunto` = col_character(),
##   tot.aam.acc.svalutazioni = col_character(),
##   `(ron) reddito operativo netto` = col_character()
## )
```

```
bsd <- bsd %>%
        rename(year = anno) %>%
        rename(totEquity = `Totale patrimonio netto`) %>%
        rename(totAssets = `Totale attivo`) %>%
        rename(totIntang = `Totale Immobilizzazioni immateriali`) %>%
        rename(staffCost = `Totale Costi del Personale`) %>%
        rename(turnover  = `Ricavi delle vendite`) %>%
        select(cf,year,turnover, totAssets, totIntang, staffCost  )
```

```
bsd <- bsd %>%
  mutate(across(everything(), gsub, pattern = "[.]", replacement = "")) %>%
  mutate(across(everything(), gsub, pattern = ",", replacement = ".")) %>%
  mutate(across(.cols = 2:6, .fns = as.numeric))
```

```
bsd %>% write_csv(paste0(pathTidyData,"bsd.csv"))
```

There are 18 columns but in this project we will use only 4, namely "cf", "year", revenues" and "staff cost". Data should be loaded as string and then converted taking into account some issues with format of numerical variables.

To convert bsd*revenuesandbsd*staffcost to numbers, we need to remove the "." used as thousand separators, and replace "," with "." as a decimal separator.

We will focus the analysis on a list of companies that are tenats at Area Science Park. The list is available in the file "data/imprese-fvg/area-tenants.csv" so we can load it in al list ("filter") and use it to subset *bsd*.

```
tenants <- read_delim( paste0(pathRawData,"area-science-park/tenants.txt") ) %>%
  select(cf)
```

```
## New names:
## * `` -> ...7
```

```
## Rows: 68 Columns: 7
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (6): insediati, Ente/Azienda, cf, DENOMINAZIONEiifvg, Campus, Addetti (b...
## lgl (1): ...7
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

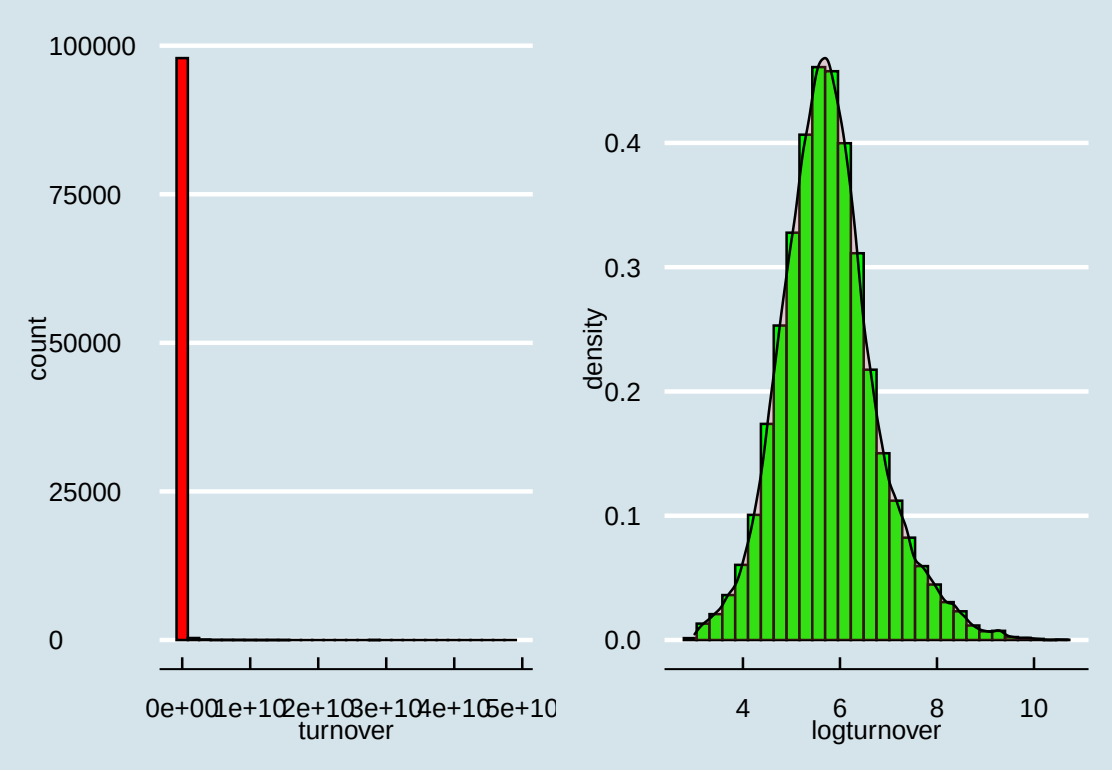
```
tens = c(tenants$cf)
bsd_tenants <- bsd %>% subset(cf %in% tenants$cf) %>%
                mutate(cf = as.factor(cf)) %>% drop_na()
```

The variable bsd$revenues spans from 0 to 1e9, so it is more convenient to work with log10

```
library(ggplot2)
library(ggpubr)
bsd3 <- bsd %>% subset(turnover > 1000) %>% subset(year = 2019)
bsd3$logturnover <- log10(bsd3$turnover)
# hist(bsd$turnover)
# hist(bsd$logturnover)
h1 <- ggplot(bsd3, aes(x=turnover)) + geom_histogram(color="black", fill="red")
h2 <- ggplot(bsd3, aes(x=logturnover)) + geom_histogram(color="black", fill="green", aes(y=..d
figure <- ggarrange(h1, h2)#labels = c("linear", "log"), ncol = 2, nrow = 1)
```

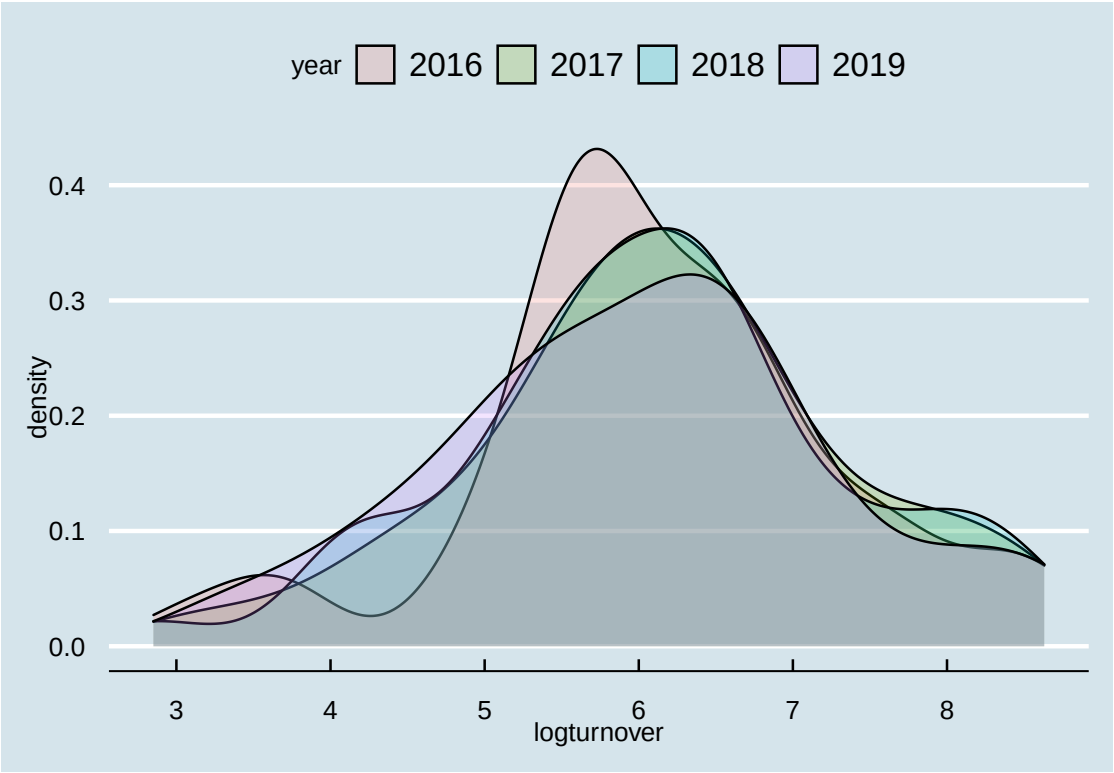
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
figure
```




```
bsd_tenants$logturnover <- log10(bsd_tenants$turnover)

tmp <- bsd_tenants  %>%
               subset(year >= 2016) %>%
               mutate(year = as.factor(year))

figure <- ggplot(tmp, aes(x=logturnover,fill=year)) + geom_density(alpha=.2)
figure
```

```
## Warning: Removed 10 rows containing non-finite values (stat_density).
```


year
2016
2017
2018
2019
0.4
0.3
0.2
0.1
0.0
density
3
4
5
6
7
8
logturnover

4 Exploring dataset CO-FVG

The original data is organized in 8 files: dati_2014.csv, dati_2015.csv, dati_2016.csv, dati_2017.csv, dati_2018.csv, dati_2019.csv, dati_2020.csv, dati_2021.csv. > TODO Currently, data exploration phase is focused on only one of the files above. Should extend it to all files using a for loop and appending results to a data.frame.

```
empl <- read_delim( paste0(pathRawData,"dati_2018.csv"))
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 395456 Columns: 43
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "|"
## chr  (25): CF, az_ragione_soc, genere, id_cittadino, professione, qualifica,...
## dbl   (8): ...1, anno, eta, mese, saldo, codice_istat, SLL_codice, qualifica...
## lgl   (5): somm, erroriEta, errori_qualifica, erroriCF, errori
## date  (5): data, data_fine, data_fine_prev, data_inizio, data_nascita
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message
```

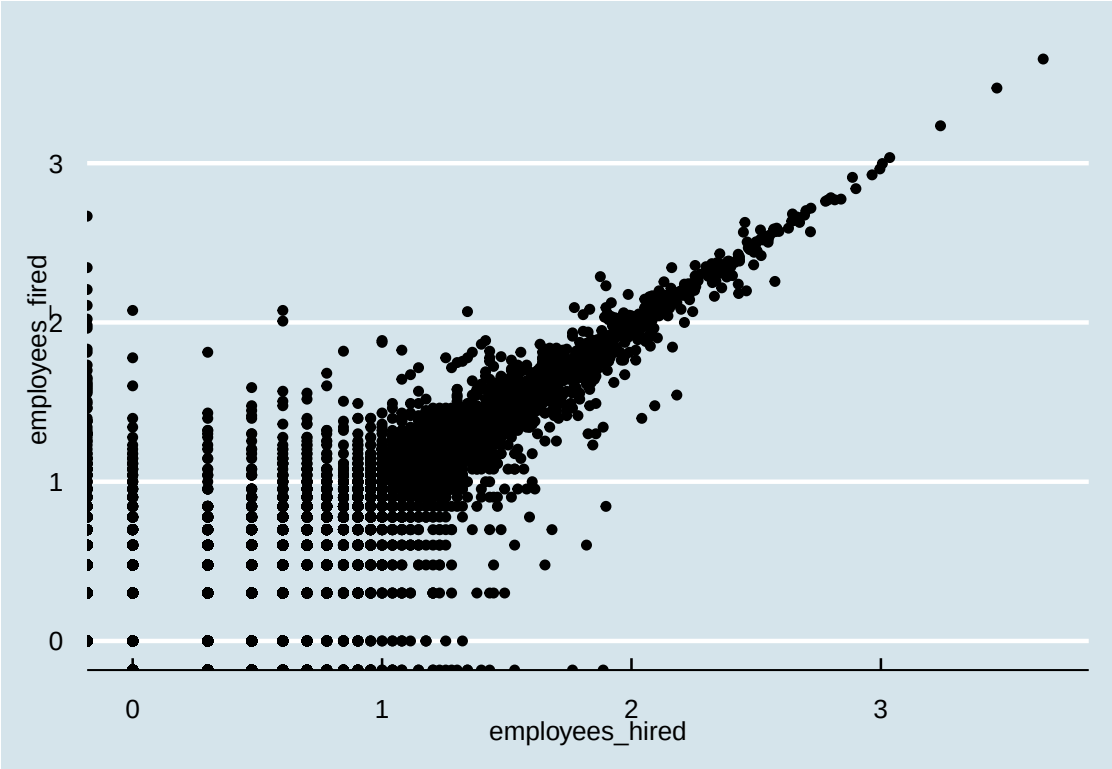
```
features <-names(empl)
some_features <- c("CF","anno","eta","genere","iso3","professione","qualifica","sal
```

There are 43 features available: ...1, CF, anno, az_ragione_soc, data, data_fine, data_fine_prev, data_inizio, data_nascita, eta, genere, id_cittadino, mese, professione, qualifica, qualifica_codice, rl_ateco, rl_ateco_macro, rl_ateco_settore, saldo, sede_op_ateco, sede_op_comune, sede_op_indirizzo, sede_op_provincia, somm, tipo_contratto, tipo_orario, cittadinanza, iso3, contientne, aggregazione, provincia, sigla_prov, comune_istat, codice_istat, SLL_codice, SLL_nome, contratto, erroriEta, errori_qualifica, qualifica_2_digit, erroriCF, errori. For the purpose of data exploration we will focus only on the following: CF, anno, eta, genere, iso3, professione, qualifica, saldo.

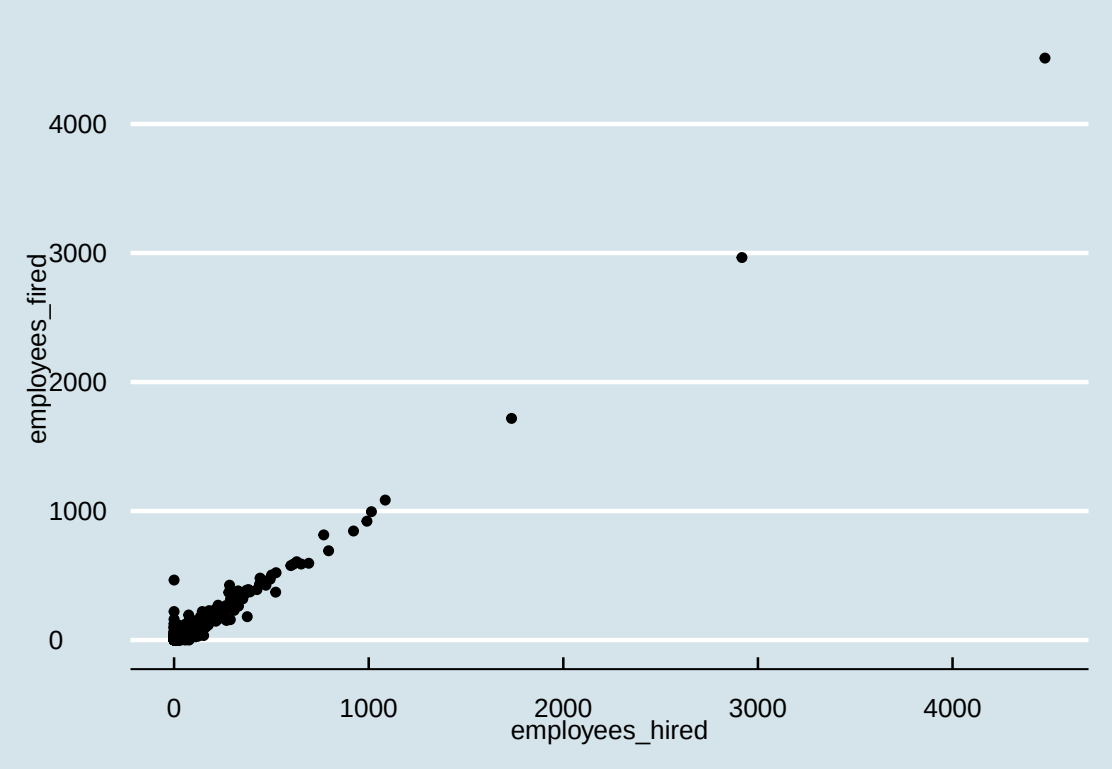
```
empl <- empl %>%
  select( one_of(some_features) ) %>%
  rename( year = anno)

empl_flows <- empl %>% select( c(CF, saldo, year)) %>%
  mutate(hf = factor(saldo))%>%
  mutate(hf=recode(hf,`-1`="fired",`1`="hired"))%>%
  group_by(CF,hf, year) %>%
  summarize(hiredfired=  sum(saldo) ) %>%
  pivot_wider( names_from = hf, values_from = hiredfired) %>%
  replace(is.na(.), 0) %>%
  mutate(turnover = hired-fired) %>%
  mutate(net = hired+fired)
```

```
## `summarise()` has grouped output by 'CF', 'hf'. You can override using the `.groups` argume
```

```
employees_hired = log10(empl_flows$hired)
employees_fired=  log10(-empl_flows$fired)
ggplot(empl_flows, aes(x=employees_hired, y=employees_fired))+
    geom_point()
```



```
employees_hired = (empl_flows$hired)
employees_fired=  (-empl_flows$fired)
ggplot(empl_flows, aes(x=employees_hired, y=employees_fired))+
    geom_point()
```



> TODO import, calculate net saldo and turnover, divide companies in quartiles

TODO improve formatting tables with library(kableExtra) %>% kable()

```
empl_flows %>% write_csv(paste0(pathTidyData,"empl_flows.csv"),)
```