Sarvesh Meenowa

Data Wrangling

19.10.2021

3.2 Missing data

```
#import libaries
#detach(package:plyr)
library(plyr)
library(dplyr)
library(lubridate)
library(ggplot2)
library(naniar)
library(plotly)
library(tidyverse)
library(tidyr)
library(xtable)
nest <- nest_legacy
unnest <- unnest_legacy
```

```
------------------------------------------------------------------------------
You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)
------------------------------------------------------------------------------

Attaching package: 'plyr'

The following object is masked from 'package:purrr':

    compact

The following objects are masked from 'package:plotly':

    arrange, mutate, rename, summarise

The following objects are masked from 'package:dplyr':

    arrange, count, desc, failwith, id, mutate, rename, summarise,
    summarize
```

```
#import python libraries
import pandas as pd
```

```
import gender_guesser.detector as gender
import plotly.express as px
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

```
#import data in python
df_py = pd.read_csv('H:/Downloads/Datatsets/theses_v2.csv')
```

```
#import data while converting implicit missing data to explicit
df <- read.csv("H:/Downloads/Datatsets/theses_v2.csv",sep=",",encoding="UTF-8", na.strings=c("
```

```
head(df)
```

Auteur	Identifiant.auteur	Titre	Directeur.de.these	Di
Saeed Al marri	NA	Le credit documentaire et l'onopposabilite des exceptions	Philippe Delebecque	De
Andrea Ramazzotti	174423705	Application de la PGD a la resolution de problemes transitoires couples en vue de l'allegement des structures composites.	Jean-Claude Grandidier,Marianne Beringhier	Gr Cl
OLIVIER BODENREIDER	NA	Conception d'un outil informatique d'etude des cinetiques observees en toxicologie clinique	Francois Kohler	Ko
Emmanuel Porte	NA	Socio-histoire des politiques publiques en matiere sociale concernant les etudiants.	Gilles Pollet	Po

Auteur	Identifiant.auteur	Titre	Directeur.de.these	Di
Arthur Devriendt	NA	LES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION ET LES NOUVELLES RURALITES.	Gabriel Dupuy	Du
Elmantsr Briak	NA	Integration forcee de l'afrique subsaharienne dans le processus de mondialisation "structuration des economies","destructuration des etats".	Edmond Jouve	Jou

```
#visualising missingness of data
vis_miss(df,warn_large_data=FALSE)
```

```
#remove scientific form of values exponential form
options(scipen=999)
#create table for missing values
missing_table <- miss_var_summary(df)
#round percentage to 2dp
missing_table$pct_miss <- round(missing_table$pct_miss,2)
#rename column names
colnames(missing_table) <- c('Variable','Number of missing data','Percentage of missing data')


missing_table
```

Variable	Number of missing data	Percentage of missing data
Date.de.premiere.inscription.en.doctorat	383668	85.71
Identifiant.auteur	129989	29.04
Langue.de.la.these	63765	14.24
Date.de.soutenance	56746	12.68

Variable	Number of missing data	Percentage of missing data
Year	56746	12.68
Identifiant.directeur	49172	10.98
Identifiant.etablissement	17085	3.82
Mise.a.jour.dans.theses.fr	177	0.04
Directeur.de.these	17	0.00
Directeur.de.these..nom.prenom.	17	0.00
Titre	9	0.00
Discipline	5	0.00
Etablissement.de.soutenance	4	0.00
Auteur	2	0.00
Statut	0	0.00
Identifiant.de.la.these	0	0.00
Accessible.en.ligne	0	0.00
Publication.dans.theses.fr	0	0.00

```
#get latex output
print(xtable(missing_table))
```

```
% latex table generated in R 3.6.3 by xtable 1.8-4 package
% Tue Oct 19 22:58:49 2021
\begin{table}[ht]
\centering
\begin{tabular}{rlrr}
  \hline
 & Variable & Number of missing data & Percentage of missing data \\
  \hline
1 & Date.de.premiere.inscription.en.doctorat & 383668 & 85.71 \\
  2 & Identifiant.auteur & 129989 & 29.04 \\
  3 & Langue.de.la.these & 63765 & 14.24 \\
  4 & Date.de.soutenance & 56746 & 12.68 \\
  5 & Year & 56746 & 12.68 \\
  6 & Identifiant.directeur & 49172 & 10.98 \\
  7 & Identifiant.etablissement & 17085 & 3.82 \\
  8 & Mise.a.jour.dans.theses.fr & 177 & 0.04 \\
  9 & Directeur.de.these &  17 & 0.00 \\
  10 & Directeur.de.these..nom.prenom. &  17 & 0.00 \\
```

```
  11 & Titre &   9 & 0.00 \\
  12 & Discipline &   5 & 0.00 \\
  13 & Etablissement.de.soutenance &   4 & 0.00 \\
  14 & Auteur &   2 & 0.00 \\
  15 & Statut &   0 & 0.00 \\
  16 & Identifiant.de.la.these &   0 & 0.00 \\
  17 & Accessible.en.ligne &   0 & 0.00 \\
  18 & Publication.dans.theses.fr &   0 & 0.00 \\
   \hline
\end{tabular}
\end{table}
```

```
#generate a new variable with normal distribution mean 200 and sd 50
#create a variable with a sequence 80% of the size of df
x <- seq(0,0.8*nrow(df))

#normalise seq x with mean and sd
y<-rnorm(x,mean=200,sd=50)
#convert to floor to int
y<-floor(y)
#create 20% size of df of NAs
na_col <- rep(NA,0.2*nrow(df))
#Create new column of n.pages with 80% values + 20% NAs
df$n.pages <- c(y,na_col)
#unique(df$n.pages)
#shuffle NAs and values
df$n.pages = sample(df$n.pages, replace=FALSE)
#view new missing plot
```

```
vis_miss(df,warn_large_data=FALSE)
```


```
#mean imputation
df$n.pages[is.na(df$n.pages)] <- mean(df$n.pages, na.rm = TRUE)
```

3.3 Common issues

- Number of defences in January **In python**

```
#drop missing dates
date_str = df_py['Date de soutenance'].dropna()
#convert to dates index
date_soutenance = pd.DatetimeIndex(date_str)
```

```
#count number of thesis on january first and all other dates, by seperating the days and month
from collections import Counter

dict_newyeareve = Counter(date_soutenance[np.logical_and(date_soutenance.day == 1, date_souten

dict_normal = Counter(date_soutenance.year.tolist())

df_normal = pd.DataFrame.from_dict(dict_normal, orient='index', columns=['nb_thesis']).reset_i

df_newyeareve = pd.DataFrame.from_dict(dict_newyeareve, orient='index', columns=['nb_thesis_01
```

```
#create proportion on jan 1st and total number of thesis
df_py = df_normal.merge(df_newyeareve, left_on='Year', right_on='Year')
df_py['Percentage'] = (df_py['nb_thesis_0101'] / df_py['nb_thesis']) * 100
```

```
df_py[['Year']].sort_values
```

```
<bound method DataFrame.sort_values of     Year
0   1993
1   2008
2   2005
3   2009
4   2013
5   2011
6   2010
7   2007
8   2012
9   2006
10  2004
11  2001
12  2015
13  2014
14  2016
15  1995
16  1997
17  1986
18  1992
19  1991
20  1987
21  1988
22  1998
23  1999
24  1985
25  1996
26  1994
27  2002
28  2000
29  1990
```

```
30  1989
31  2003
32  1982
33  1972
34  1971
35  1976
36  1973
37  2017
38  1984
39  2018
40  1980
41  1979>
```

```
fig = px.line(df_py, x="Year", y="Percentage",color="Year",text='Year')
fig.update_traces(textposition="bottom right")
fig.update_annotations(textangle=90)

fig.show()
```

3.4 Outliers

```
#create another column where author names and supervisor names are lower case
df <- df %>% mutate(auteur= tolower(Auteur))

df <- df %>% mutate(supervisors=tolower(Directeur.de.these))
```

```

```

```
#unnest rows which have more than 1 supervisor
df_unest <- df %>%
    mutate(supervisors=strsplit(supervisors, ",")) %>%
    unnest(supervisors)
```

```
#detach(package:plyr)
```

```
 #count number of supervisors based on Id and name
df_supervisor = filter(df_unest) %>% group_by(Identifiant.directeur,supervisors) %>% count() %
```

```
head(df_supervisor)
```

Identifiant.directeur	supervisors	n
59375140	jean-michel scherrmann	208
26730774	francois-paul blanc	205
26756625	pierre brunel	193
29561248	philippe delebecque	178
27084868	guy pujolle	177
98531891	michel bertucat	173

```
#drop first row
df_supervisor = df_supervisor[-1,]
```

```
#using percentiles method
```

```
#testing the lower bounds at 1 percentile
lower_bound <- quantile(df_supervisor$n, 0.003)
lower_bound
```

0.3%: 1

```
#testing the upper bound at 99th percentile
upper_bound <- quantile(df_supervisor$n, 0.997)
upper_bound
```

99.7%: 34

```

```

```
lower_bound <- quantile(df_supervisor$n, 0.01)
upper_bound <- quantile(df_supervisor$n, 0.997)

outlier_ind <- which(df_supervisor$n< lower_bound | df_supervisor$n> upper_bound)

df_supervisor[outlier_ind, ]
```

Identifiant.directeur	supervisors	n
59375140	jean-michel scherrmann	208

Identifiant.directeur	supervisors	n
26730774	francois-paul blanc	205
26756625	pierre brunel	193
29561248	philippe delebecque	178
27084868	guy pujolle	177
98531891	michel bertucat	173
27158578	bernard teyssie	146
26870177	bruno foucart	132
26997894	henry de lumley	132
58552499	jean-claude chaumeil	131
27001067	michel maffesoli	128
59209143	roger g. boulu	127
02705554X	daniel-henri pageaux	124
02703352X	georges molinie	116
26725916	jean bessiere	114
26702606	francis balle	109
35137576	gregoire loiseau	101
26787083	eliane chiron	96
27024938	michel meslin	96
55477046	pierre-philippe rey	96
27187217	jean-marie vincent	95
26913712	philippe hamon	92
59677309	jean-michel warnet	92
69632472	paul r. cohen	92
50540289	claude berthomieu	88
27079872	hugues portelli	87
26798484	catherine coquery-vidrovitch	86
70128545	alain greiner	86

Identifiant.directeur	supervisors	n
26940825	robert jouanny	84
27025403	jacques mestre	84
...	...	...
30044456	michel germain	35
30211379	remy cabrillac	35
30251699	raphael romi	35
30410789	pierre mayer	35
31104096	michel costagliola	35
31208584	christian vallar	35
31293751	didier alexandre	35
31853714	jean-marie chevalier	35
32278632	bernard salignon	35
32894031	yvon maday	35
34643095	michel gourgand	35
35266198	francois-xavier lucas	35
35715359	jean-marc chouvel	35
54373506	michel bibent	35
57586640	bernard gabriel	35
59376996	alain roger	35
59388153	jean-laurent mallet	35
59835060	hieu ha minh	35
60110988	dominique dormont	35
60167416	isabelle rico-lattes	35
60655992	didier decoster	35
60710500	laurent masegeta kashema bin muzigwa	35
66947073	rene soenen	35
69287317	jacques farisse	35

Identifiant.directeur	supervisors	n
70141738	rene soulayrol	35
70951098	pierre lamy	35
74531557	jean-francois jullien	35
75537990	yves berthier	35
75937689	bernard desablens	35
76008711	sylvain rault	35

```
nrow(df_supervisor[outlier_ind, ])
```

2015

```
length(unique(df_supervisor$supervisors))
```

116001

Verify whether they are mistakes or not

```
#funtion works to scrap the number of thesis by ID, works for outliers
NumberofTheses <- function(df,ID) {
  #Extracting author ids's which may be considered as outliers
    numberOftheses <- max(data.frame(filter(subset(df, Identifiant.auteur == ID))
                                     %>% group_by(Auteur,Identifiant.auteur) %>% count())$n)
    #filter subset of dataframe by ID , group by author name and author id and count
    df2<- data.frame(filter(subset(df, Identifiant.auteur == ID))
                            %>% group_by(Auteur,Identifiant.auteur) %>% count())
    #get length of Id
    lenID <-nchar(as.character(subset(df2,n == numberOftheses)$Identifiant.auteur))
    strID <- as.character(subset(df2,n == numberOftheses)$Identifiant.auteur)
    if (lenID == 8){
        ID <- paste(as.character('0'),as.character(strID),sep='')
    }
    #verify from theses.fr the link
    link <- paste('https://theses.fr/',ID,sep='')
    qi_webpage <- read_html(link)
    tagContent <- qi_webpage %>%
      html_nodes(xpath = "//*[@id='sommaire']/ul/li/a") %>%
      html_text()
    numberOfTheses <- str_extract_all(tagContent, "\\d+")[[1]]
}
```

3.5 Preliminary results

```
------------------------------------------------------------------------------
You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)
------------------------------------------------------------------------------

Attaching package: 'plyr'

The following object is masked from 'package:purrr':

    compact

The following objects are masked from 'package:plotly':

    arrange, mutate, rename, summarise

The following objects are masked from 'package:dplyr':

    arrange, count, desc, failwith, id, mutate, rename, summarise,
    summarize
```

```
detach(package:plyr)
```

```
Error in detach(package:plyr): invalid 'name' argument
Traceback:


1. detach(package:plyr)

2. stop("invalid 'name' argument")
```

```
#subset language and defense date
date_language= df %>% select(Date.de.soutenance, Langue.de.la.these)
library(tidyr)
#drop Nas
date_language <- date_language %>% drop_na()
#format date
date_language$Date.de.soutenance <- as.Date(date_language$Date.de.soutenance, "%d-%m-%y")
```

```
detach(package:plyr)
```

```
Error in detach(package:plyr): invalid 'name' argument
Traceback:


1. detach(package:plyr)

2. stop("invalid 'name' argument")
```

```
#convert all languages to lower case
date_language$Langue.de.la.these <- tolower(date_language$Langue.de.la.these)
#recode languages based on conditions
date_language <- date_language %>% mutate(Lang_Type = case_when(
    (Langue.de.la.these == "en") ~ "English",
    (Langue.de.la.these == "fr") ~ "French",
    (Langue.de.la.these == "enfr" | Langue.de.la.these =="Fren" ) ~ "Bilingual",
    ((length(Langue.de.la.these) == 1 & (Langue.de.la.these) != "en" & length(Langue.de.la.the
    ))


head(date_language)
#detach(package:plyr)
#get year for each language
date_language <- date_language %>% dplyr::mutate(year = lubridate::year(Date.de.soutenance)
                                  )
#count number of language based on type and year
date_language <- date_language %>% select(Lang_Type,year) %>% group_by(Lang_Type,year) %>% cou
#remove irrelevant data
date_language <- date_language %>% filter(year>=1988 & year <2019)
#calculate the percentage by language for each year
date_language <- date_language %>% group_by(year) %>% mutate(sum=sum(n)) %>% mutate(perc = n/s

head(date_language)
```

Date.de.soutenance	Langue.de.la.these	Lang_Type
1993-01-01	fr	French
2015-01-01	fr	French
2015-01-01	fr	French
2013-12-07	fr	French
2013-11-25	fr	French
2013-11-22	fr	French

Lang_Type	year	n	sum	perc
Bilingual	1988	150	11044	1.358204
Bilingual	1989	222	11101	1.999820
Bilingual	1990	203	11011	1.843611
Bilingual	1991	176	10831	1.624965
Bilingual	1992	231	12064	1.914788
Bilingual	1993	221	12308	1.795580

```
#stacked area plot
p <- date_language %>% ggplot(aes(x=year, y=perc, fill=Lang_Type, color=Lang_Type, text=Lang_T
    geom_area(alpha=0.6 , size=.4, color="grey")  +
    theme(plot.title = element_text(family = "serif", color = "black"),
    axis.text.x = element_text(family = "serif", color = "black"),
    axis.text.y = element_text(family = "serif", color = "black")) +
    labs(y = "Percentage of Theses ")
```

```
#ggplot stacked area plot
p
```


```
p2 <- date_language %>% ggplot(aes(x=year, y=n, fill=Lang_Type, color=Lang_Type, text=Lang_Typ
    geom_area(alpha=0.6 , size=.4, color="grey")  +
    theme(plot.title = element_text(family = "serif", color = "black"),
    axis.text.x = element_text(family = "serif", color = "black"),
    axis.text.y = element_text(family = "serif", color = "black")) +
    labs(y = "Number of Theses ")
```

```
p2
```


```
#plotly stacked area chart for language
ggplotly(p, tooltip="text")
```

```
head(date_language)
```

Lang_Type	year	n	sum	perc
Bilingual	1988	150	11044	1.358204
Bilingual	1989	222	11101	1.999820
Bilingual	1990	203	11011	1.843611
Bilingual	1991	176	10831	1.624965
Bilingual	1992	231	12064	1.914788
Bilingual	1993	221	12308	1.795580

```
#Stacked area chart for language using plotly
fig <- plot_ly(type='scatter', x = date_language$year, y = date_language$perc, color=date_lang
fig <- fig %>% layout(xaxis= list(title="Year",layout.font="Times New Roman"),
    yaxis = list(title="Percentage of theses",layout.font="Times New Roman"))

fig
```

```
No scatter mode specifed:
  Setting the mode to markers
  Read more about this attribute -> https://plot.ly/r/reference/#scatter-mode
No scatter mode specifed:
  Setting the mode to markers
  Read more about this attribute -> https://plot.ly/r/reference/#scatter-mode
```

```
library(plyr)
```

```
------------------------------------------------------------------------------
You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)
------------------------------------------------------------------------------

Attaching package: 'plyr'

The following object is masked from 'package:purrr':

    compact

The following objects are masked from 'package:plotly':

    arrange, mutate, rename, summarise

The following objects are masked from 'package:dplyr':

    arrange, count, desc, failwith, id, mutate, rename, summarise,
    summarize
```

```
#calculate mean and standard deviation of each januaries, februaries etc from 2009-2019
month_summary <- ddply(df_these, ~month, summarise, mean = mean(perc), sd = sd(perc))
month_summary

#convert month number to the name of the months
month_summary <- month_summary %>% mutate(Month = month.name[month])
month_summary$month <- factor(month_summary$month, levels = month_summary$month)
```

```
month_summary

month_summary <- arrange(month_summary,(month))
```

month	mean	sd
1	4.849091	0.50072856
2	3.647273	0.42965314
3	4.803636	0.41538591
4	3.634545	0.53556258
5	4.160909	0.33497625
6	8.452727	0.42778712
7	6.270909	0.52828883
8	0.330000	0.09726253
9	9.736364	1.87071790
10	10.174545	0.72918261
11	16.044545	1.47863696
12	27.891818	3.25557005

month	mean	sd	Month
1	4.849091	0.50072856	January
2	3.647273	0.42965314	February
3	4.803636	0.41538591	March
4	3.634545	0.53556258	April
5	4.160909	0.33497625	May
6	8.452727	0.42778712	June
7	6.270909	0.52828883	July
8	0.330000	0.09726253	August
9	9.736364	1.87071790	September

month	mean	sd	Month
10	10.174545	0.72918261	October
11	16.044545	1.47863696	November
12	27.891818	3.25557005	December

```
#divide sd by 2
month_summary <- month_summary %>% mutate(sd2= sd/2)
```

```
month_summary
```

month	mean	sd	Month	sd2
1	4.849091	0.50072856	January	0.25036428
2	3.647273	0.42965314	February	0.21482657
3	4.803636	0.41538591	March	0.20769296
4	3.634545	0.53556258	April	0.26778129
5	4.160909	0.33497625	May	0.16748813
6	8.452727	0.42778712	June	0.21389356
7	6.270909	0.52828883	July	0.26414442
8	0.330000	0.09726253	August	0.04863127
9	9.736364	1.87071790	September	0.93535895
10	10.174545	0.72918261	October	0.36459130
11	16.044545	1.47863696	November	0.73931848
12	27.891818	3.25557005	December	1.62778503

```
#set month as level
month_summary$Month <- factor(month_summary$Month, levels=c("January","February","March","Apri
```

```
ggplot(month_summary) +
geom_bar(aes(x=Month, y=mean), stat="identity", fill="blue", alpha=0.7) +
geom_errorbar(aes(x=Month, ymin=mean-sd2, ymax=mean+sd2), width=0.4, colour="black", alpha=0.9
theme( plot.title = element_text(family = "serif", color = "black", size = 12),
```

```
      axis.text.x = element_text(family = "serif", color = "black",vjust=0.6, size = 12, angle
     axis.text.y = element_text(family = "serif", color = "black", size = 12)) +
labs( x = "Month", y = "Mean Percentage of theses defended")
```

```
#extract author and date
df_gender = df_py[["Auteur","Date de soutenance"]]

df_gender.head()
#extract first name of author
df_gender['first_name']=df_gender.Auteur.str.split(expand=True)[[0]]

#instantiate class
d = gender.Detector()

#create function to get gender from first name
def get_gender_by_name(x,d):
    return d.get_gender(u"{}".format(x))

#convetr to title case
def title_case(x):
    if x is None:
        pass
    else:
        return x.title()
#apply title case to first name
df_gender["first_name"]=df_gender["first_name"].apply(lambda x: title_case(x))
#get gender from first name
df_gender["gender"] = df_gender['first_name'].apply(lambda x:get_gender_by_name(x,d))



#create year column from date of defense
df_gender['year'] = pd.DatetimeIndex(df_gender["Date de soutenance"]).year
#dropna
df_gender.dropna(subset=['year'],how='all',inplace=True)
#check Nas
df_gender.isnull().sum()
#check shape
df_gender.shape
#convert year column to integer
df_gender['year'] = df_gender['year'].astype(int)

#count number of authors by gender and year
df_gender_count = df_gender.groupby(['gender','year']).count().reset_index()
#remove border years
df_gender_count = df_gender_count.query('year >= 1988 & year < 2020')
#rename column to more meaningful name
df_gender_count.rename(columns={'Auteur':'Number of authors'},inplace=True)
#calculate sum of theses for each gender by year
sum_df= df_gender_count.groupby('year')["Date de soutenance"].sum()
```

```
#merge dataframe to get sum for each row
df_gender_count = pd.merge(df_gender_count,sum_df,on='year')

#calculate percentage by gender for each year
df_gender_count['percentage'] = df_gender_count['Date de soutenance_x']/df_gender_count['Date

df_gender_count


#stacked area for the percentage of theses by gender over years
fig1 = px.area(df_gender_count, x="year", y="percentage",color="gender",line_group="gender")

fig1.show()
#stacked area for the number of these by gender over years

fig2 = px.area(df_gender_count, x="year", y="Number of authors",color="gender",line_group="gen

fig.show()
```