PROJECT 3
DATA CLEANING

Romain Courtois

Hye-Jin Cho-Drugeon

DATA DESCRIPTION

DATA DESCRIPTION

- train.csv: 45,211 rows and 18 columns ordered by date (from May 2008 to November 2010)
- test.csv: 4521 rows and 18 columns with 10% of the examples (4521), randomly selected from train.csv

AIM

direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls.

FINANCES

Term deposits are a major source of income for a bank. A term deposit is a cash investment held at a financial institution. Your money is invested for an agreed rate of interest over a fixed amount of time, or term. The bank has various outreach plans to sell term deposits to their customers such as email marketing, advertisements, telephonic marketing, and digital marketing.

BANK CLIENT DATA

1 - age (numeric)

2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services")

3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)

4 - education (categorical: "unknown","secondary","primary","tertiary")

5 - default: has credit in default? (binary: "yes","no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes","no")

8 - loan: has personal loan? (binary: "yes","no")

BANK CLIENT DATA

9 - contact: contact communication type (categorical: "unknown","telephone","cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")

12 - duration: last contact duration, in seconds (numeric)

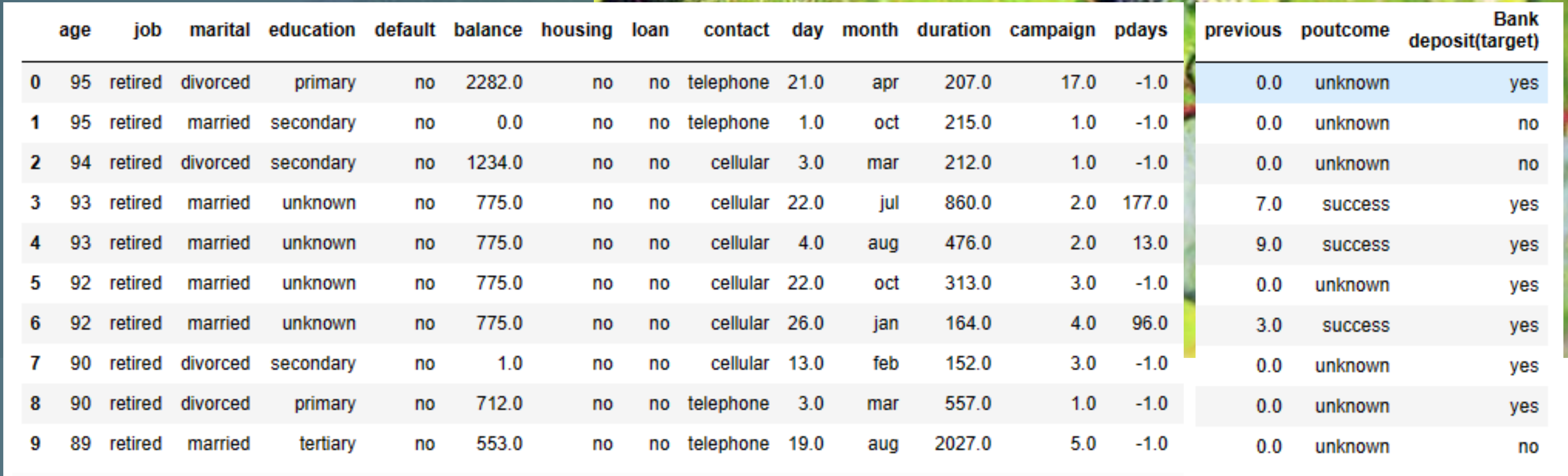
# other attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

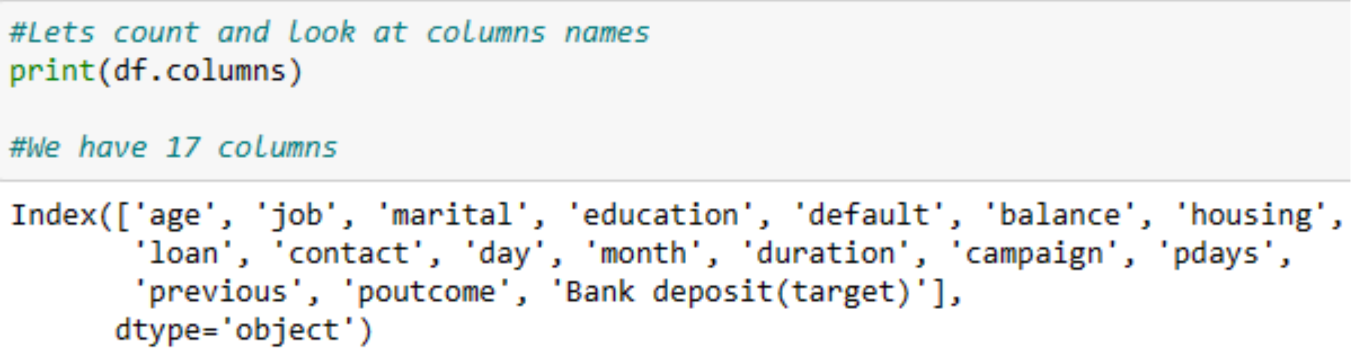
15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")



	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	Bank deposit(target)
0	95	retired	divorced	primary	no	2282.0	no	no	telephone	21.0	apr	207.0	17.0	-1.0	0.0	unknown	yes
1	95	retired	married	secondary	no	0.0	no	no	telephone	1.0	oct	215.0	1.0	-1.0	0.0	unknown	no
2	94	retired	divorced	secondary	no	1234.0	no	no	cellular	3.0	mar	212.0	1.0	-1.0	0.0	unknown	no
3	93	retired	married	unknown	no	775.0	no	no	cellular	22.0	jul	860.0	2.0	177.0	7.0	success	yes
4	93	retired	married	unknown	no	775.0	no	no	cellular	4.0	aug	476.0	2.0	13.0	9.0	success	yes
5	92	retired	married	unknown	no	775.0	no	no	cellular	22.0	oct	313.0	3.0	-1.0	0.0	unknown	yes
6	92	retired	married	unknown	no	775.0	no	no	cellular	26.0	jan	164.0	4.0	96.0	3.0	success	yes
7	90	retired	divorced	secondary	no	1.0	no	no	cellular	13.0	feb	152.0	3.0	-1.0	0.0	unknown	yes
8	90	retired	divorced	primary	no	712.0	no	no	telephone	3.0	mar	557.0	1.0	-1.0	0.0	unknown	yes
9	89	retired	married	tertiary	no	553.0	no	no	telephone	19.0	aug	2027.0	5.0	-1.0	0.0	unknown	no

STEP1
DATA FRAME HEAD10



```
#Lets count and look at columns names
print(df.columns)

#We have 17 columns
```

```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'Bank deposit(target)'],
      dtype='object')
```

#marital encoding column "marital" as numeric (married: 1 divorced: -1, single; 0)

#education, 27 missing value, 2 bad input (hjkl-unknown, Tertiary-moved to tertiary) / Unknown 717 + 1 (6 percents)

#default, binary, most frequent method (98 percent confidence)

#balance, string needed to be float,

#loan, 12 omitting variables, binary, most frequent method (98 percent confidence)

#contact unknown + 1 bad input (26 percent) except for cellular and telephone

# month 8 omitting

# campaign: max 43 times for contacting. normal distribution (maybe)

# pdays, previous: pdays(new customers as -1) and previous's comparison

# poutcome, no missing but 9214 unknown variables (encoding column "poutcome" as numeric (unknown: 0 failure: -1, success; 1 other 0))

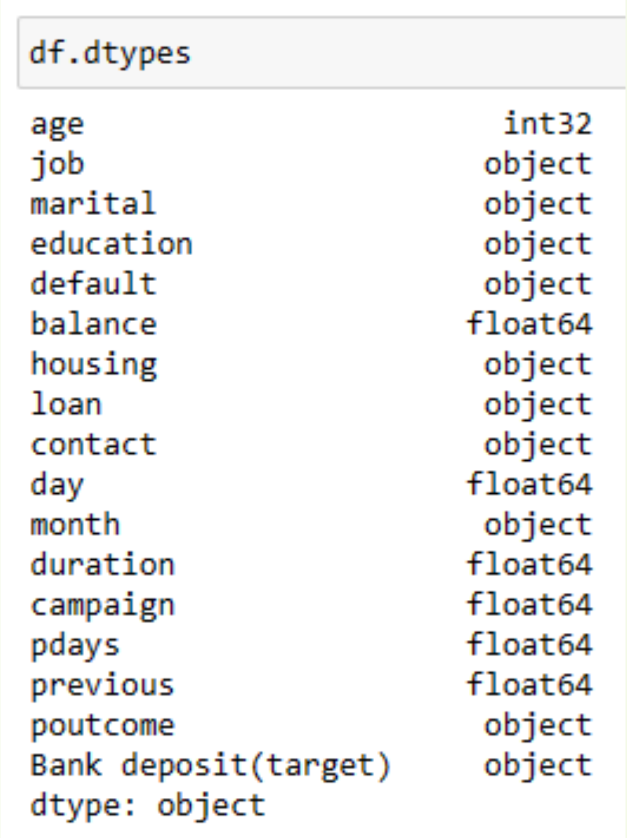
# Bank deposit

CHALLENGE

6 rows, empty

Meta data (4521 rows)

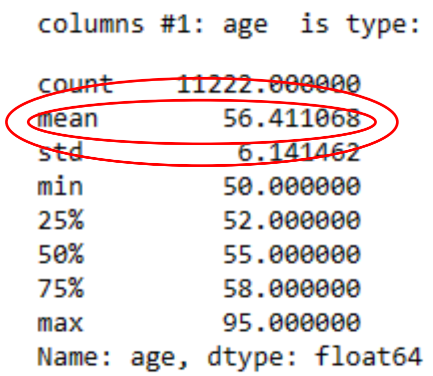
Each column has each problem



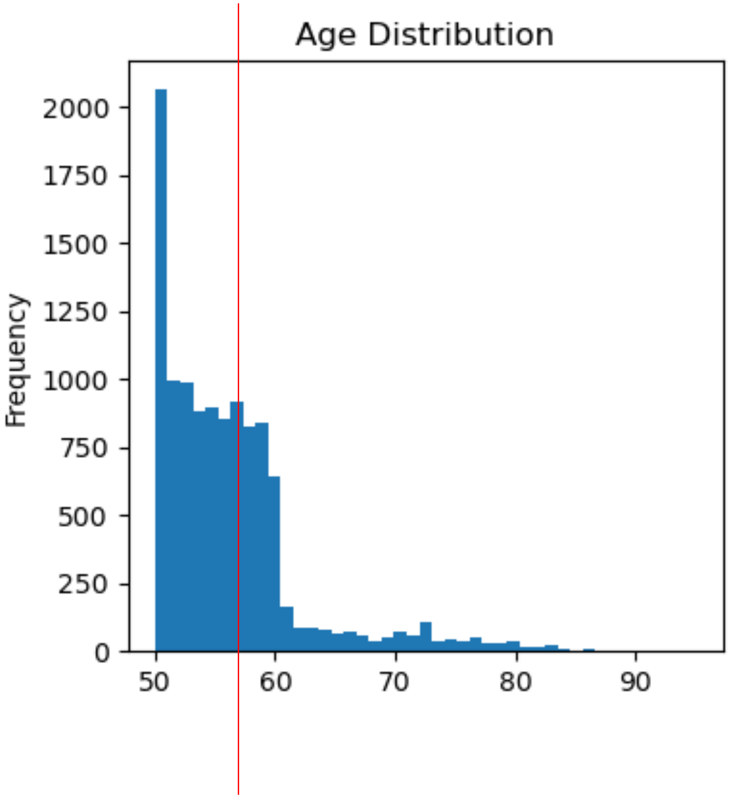
```
df.dtypes

age                         int32
job                        object
marital                    object
education                  object
default                    object
balance                   float64
housing                    object
loan                       object
contact                    object
day                       float64
month                      object
duration                  float64
campaign                  float64
pdays                     float64
previous                  float64
poutcome                   object
Bank deposit(target)       object
dtype: object
```

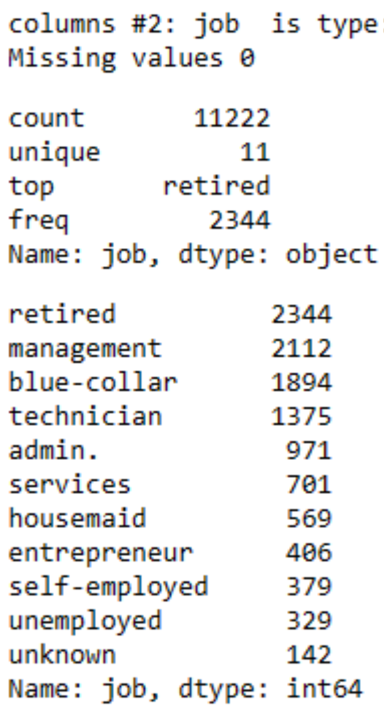
DATA ANALYSIS DESCRIPTIVE STATISTICS

Clean data

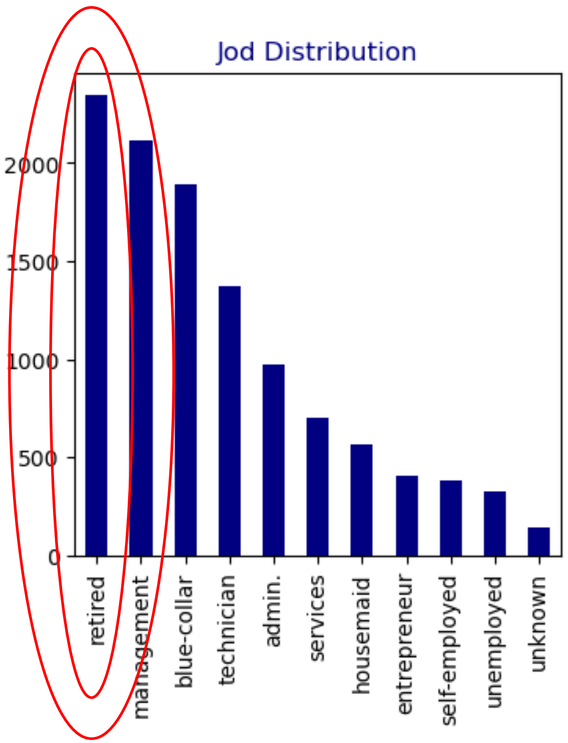


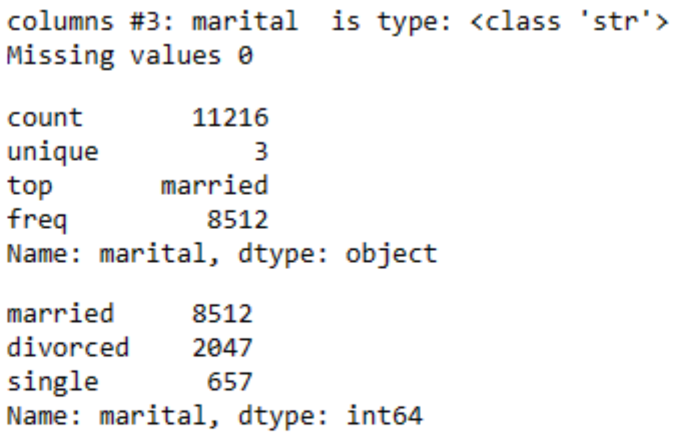
```
columns #1: age  is type: <class 'str'>

count    11222.000000
mean        56.411068
std          6.141462
min         50.000000
25%         52.000000
50%         55.000000
75%         58.000000
max         95.000000
Name: age, dtype: float64
```



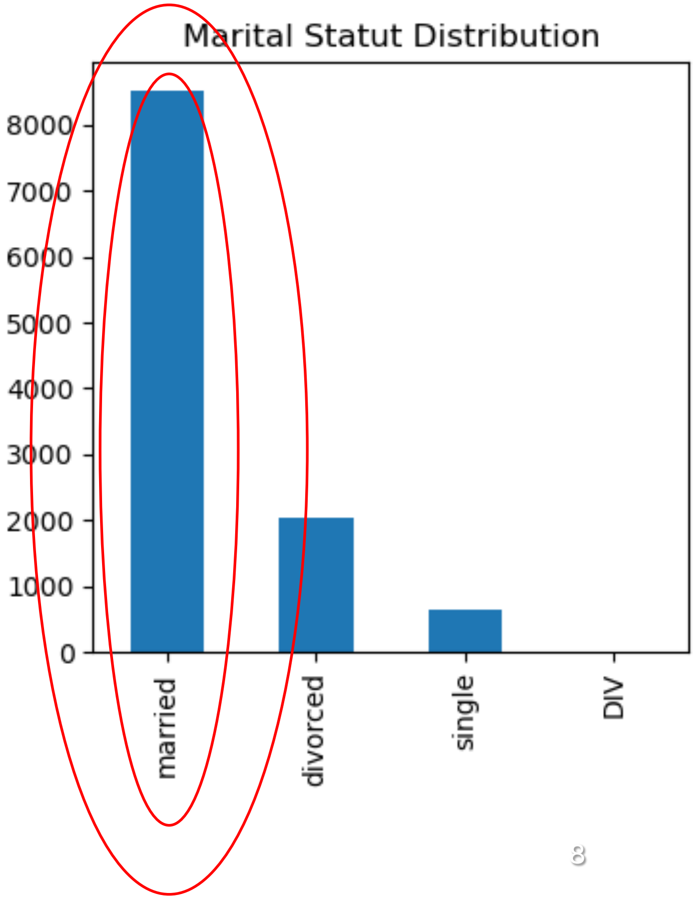


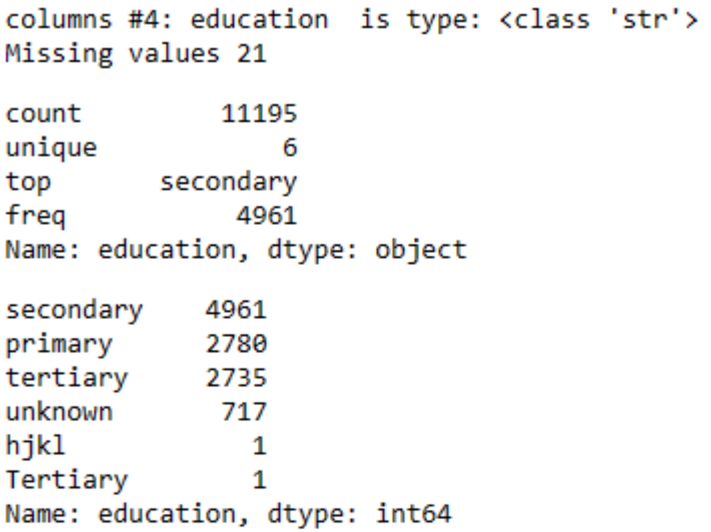
```
columns #2: job  is type: <class 'str'>
Missing values 0

count         11222
unique           11
top         retired
freq           2344
Name: job, dtype: object

retired          2344
management       2112
blue-collar      1894
technician       1375
admin.            971
services          701
housemaid         569
entrepreneur      406
self-employed     379
unemployed        329
unknown           142
Name: job, dtype: int64
```



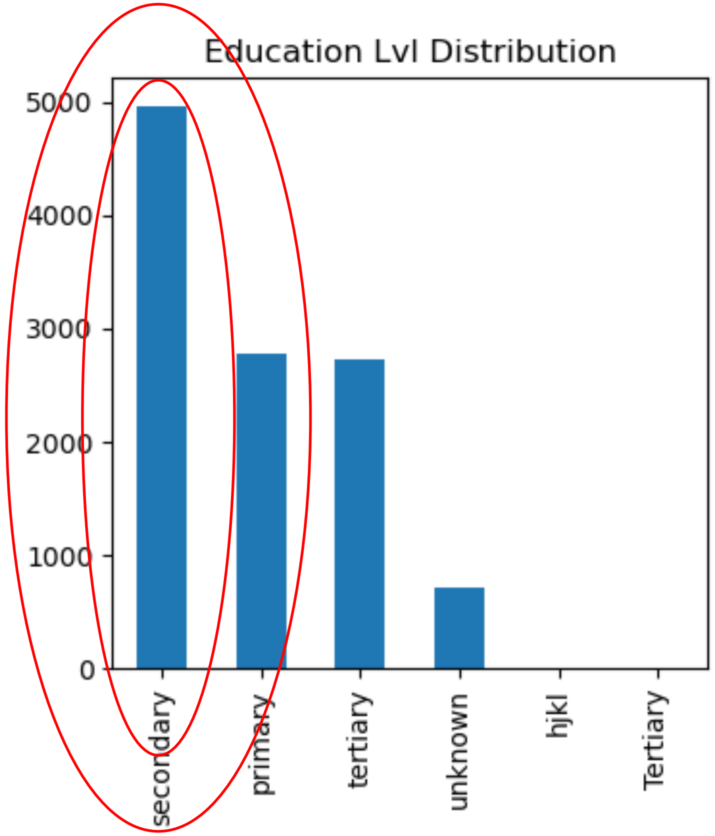


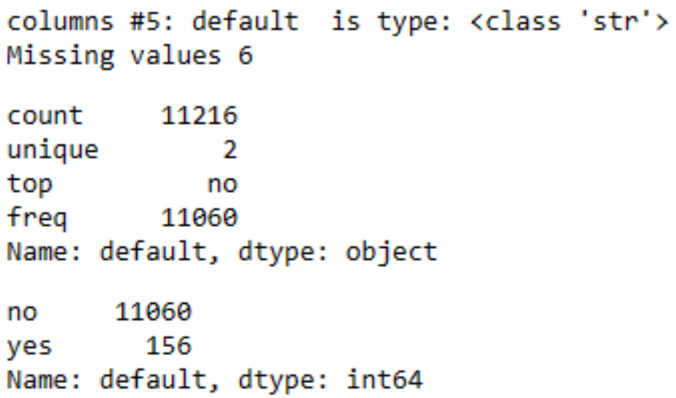
```
columns #3: marital  is type: <class 'str'>
Missing values 0

count       11216
unique          3
top       married
freq         8512
Name: marital, dtype: object

married     8512
divorced    2047
single       657
Name: marital, dtype: int64
```



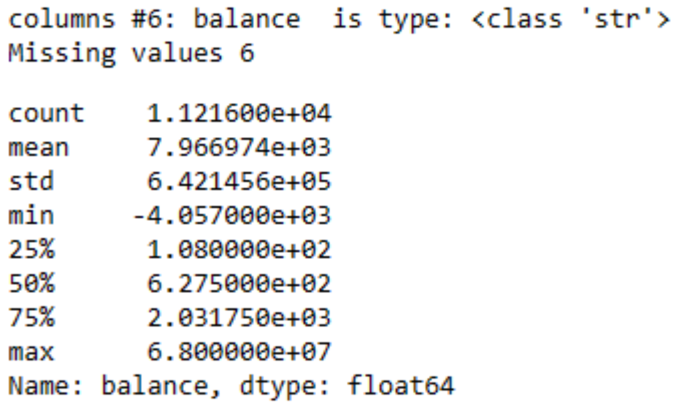


```
columns #4: education  is type: <class 'str'>
Missing values 21

count          11195
unique             6
top        secondary
freq            4961
Name: education, dtype: object

secondary    4961
primary      2780
tertiary     2735
unknown       717
hjkl            1
Tertiary        1
Name: education, dtype: int64
```

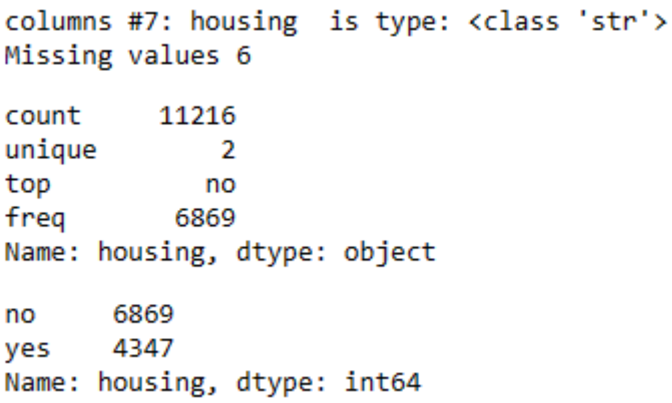




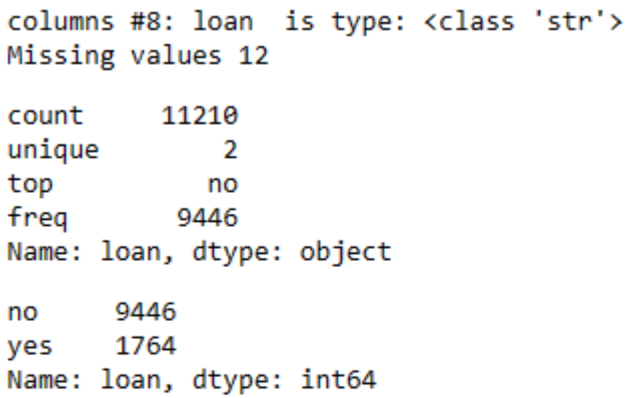
```
columns #5: default  is type: <class 'str'>
Missing values 6

count      11216
unique         2
top           no
freq       11060
Name: default, dtype: object

no     11060
yes      156
Name: default, dtype: int64
```



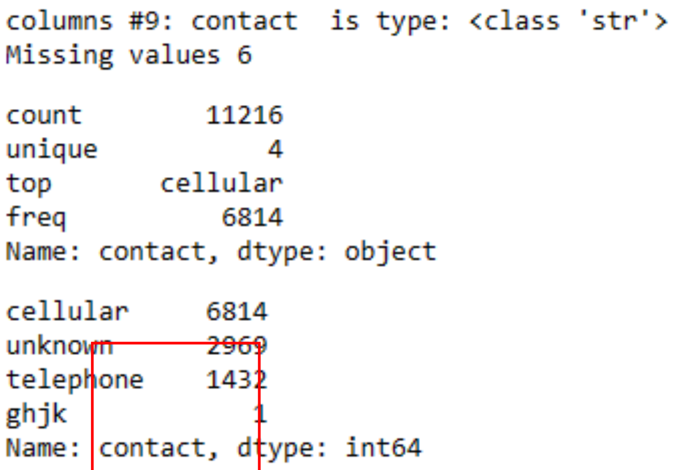
```
columns #6: balance  is type: <class 'str'>
Missing values 6

count     1.121600e+04
mean      7.966974e+03
std       6.421456e+05
min      -4.057000e+03
25%       1.080000e+02
50%       6.275000e+02
75%       2.031750e+03
max       6.800000e+07
Name: balance, dtype: float64
```



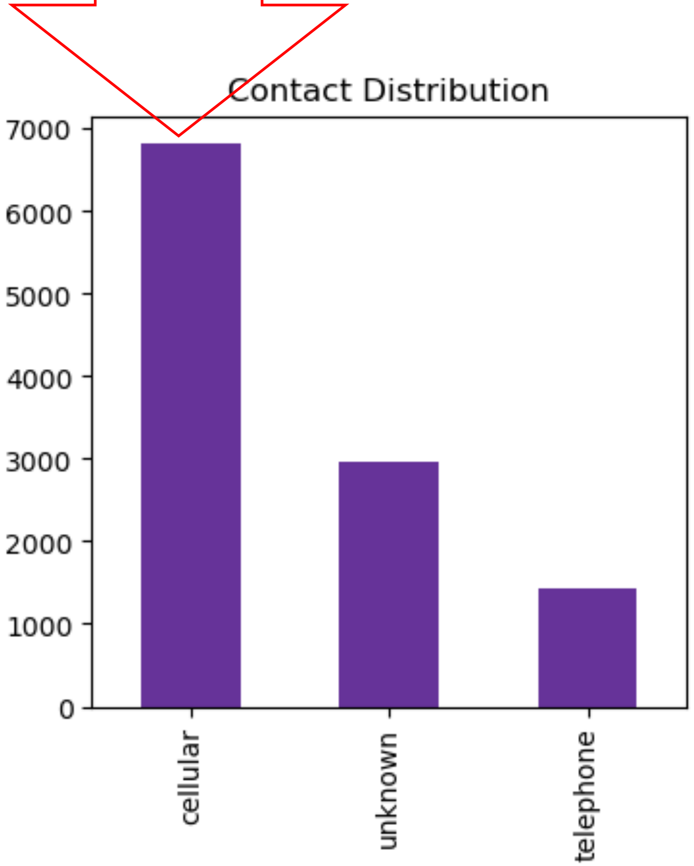
```
columns #7: housing  is type: <class 'str'>
Missing values 6

count      11216
unique         2
top           no
freq        6869
Name: housing, dtype: object

no     6869
yes    4347
Name: housing, dtype: int64
```



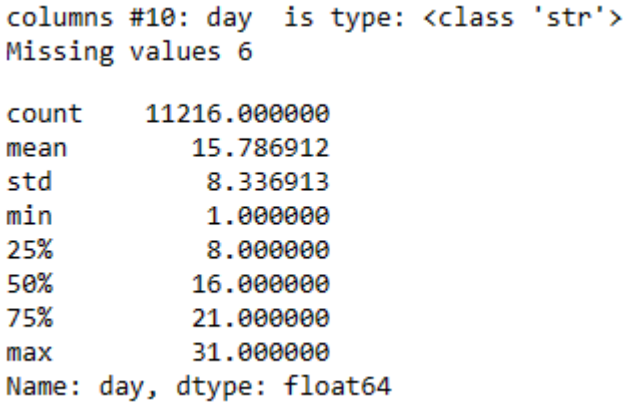
```
columns #8: loan  is type: <class 'str'>
Missing values 12

count      11210
unique         2
top           no
freq        9446
Name: loan, dtype: object

no     9446
yes    1764
Name: loan, dtype: int64
```



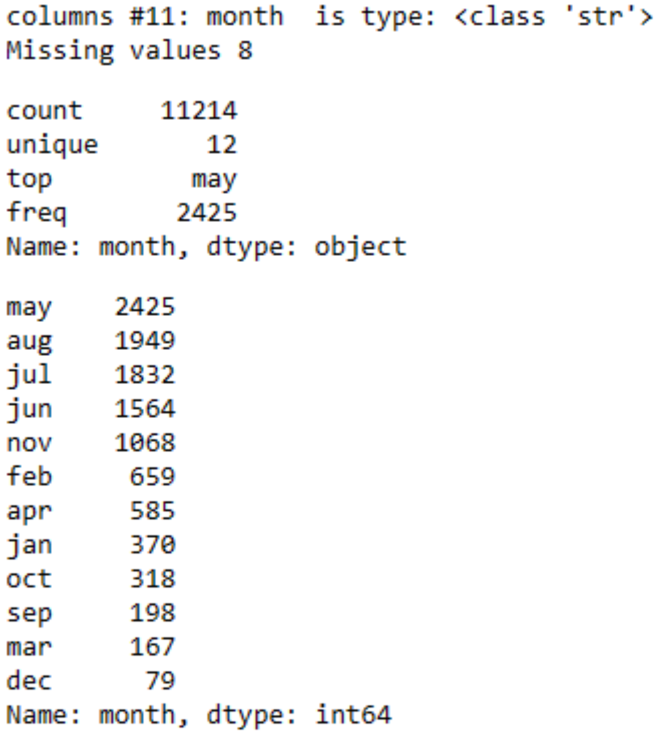
```
columns #9: contact  is type: <class 'str'>
Missing values 6

count         11216
unique            4
top        cellular
freq           6814
Name: contact, dtype: object

cellular     6814
unknown      2969
telephone    1432
ghjk            1
Name: contact, dtype: int64
```



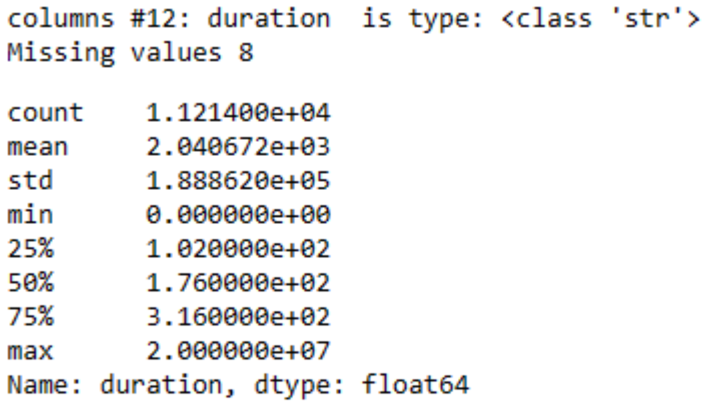
Contact Distribution



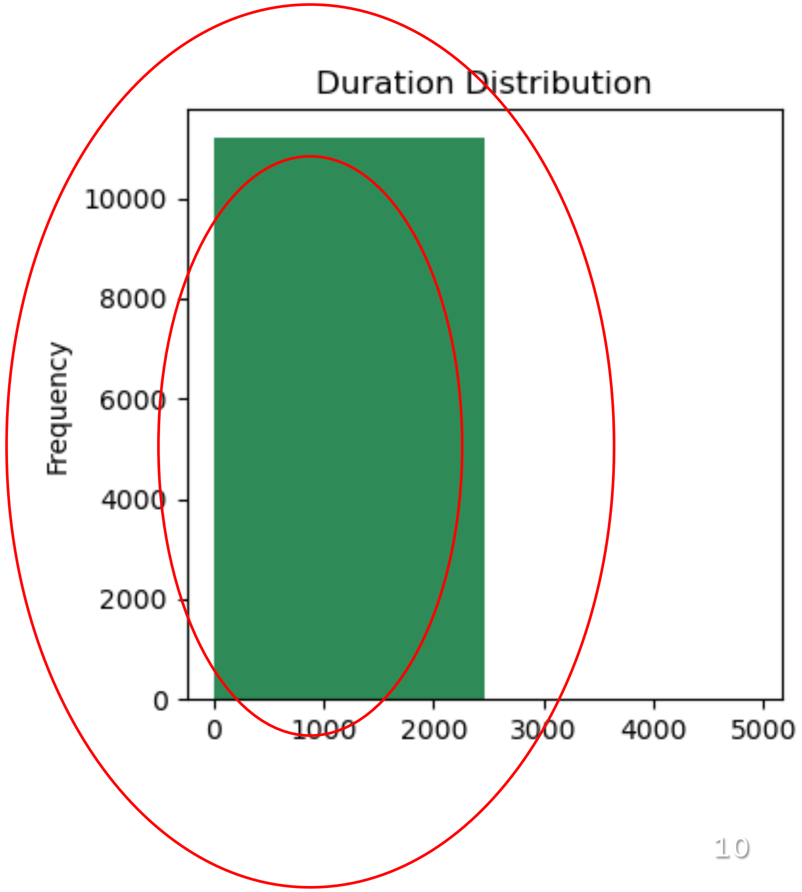
```
columns #10: day  is type: <class 'str'>
Missing values 6

count    11216.000000
mean        15.786912
std          8.336913
min          1.000000
25%          8.000000
50%         16.000000
75%         21.000000
max         31.000000
Name: day, dtype: float64
```



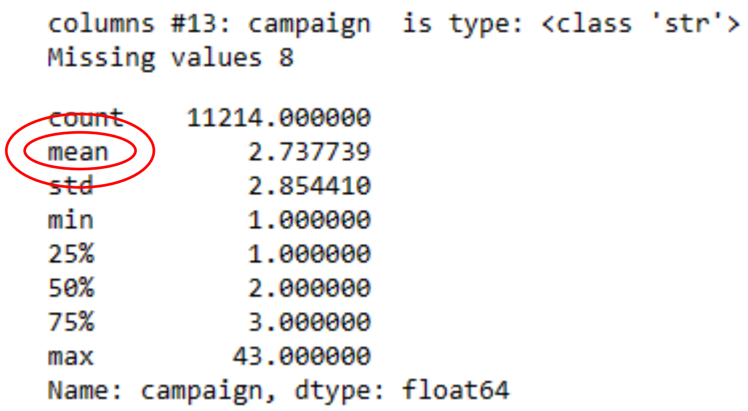
```
columns #11: month  is type: <class 'str'>
Missing values 8

count     11214
unique       12
top         may
freq       2425
Name: month, dtype: object

may    2425
aug    1949
jul    1832
jun    1564
nov    1068
feb     659
apr     585
jan     370
oct     318
sep     198
mar     167
dec      79
Name: month, dtype: int64
```



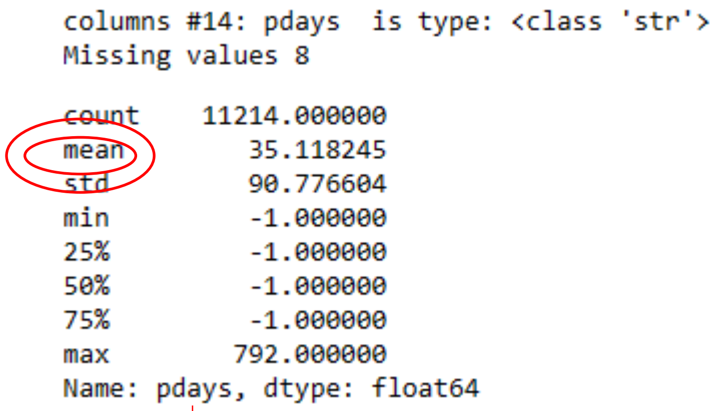
```
columns #12: duration  is type: <class 'str'>
Missing values 8

count    1.121400e+04
mean     2.040672e+03
std      1.888620e+05
min      0.000000e+00
25%      1.020000e+02
50%      1.760000e+02
75%      3.160000e+02
max      2.000000e+07
Name: duration, dtype: float64
```



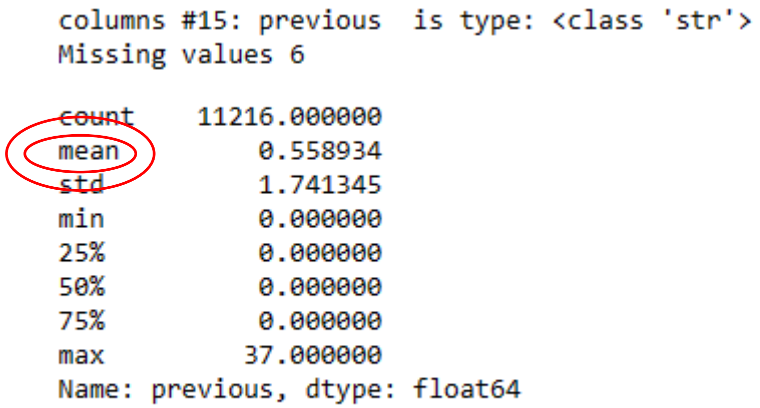
Duration Distribution



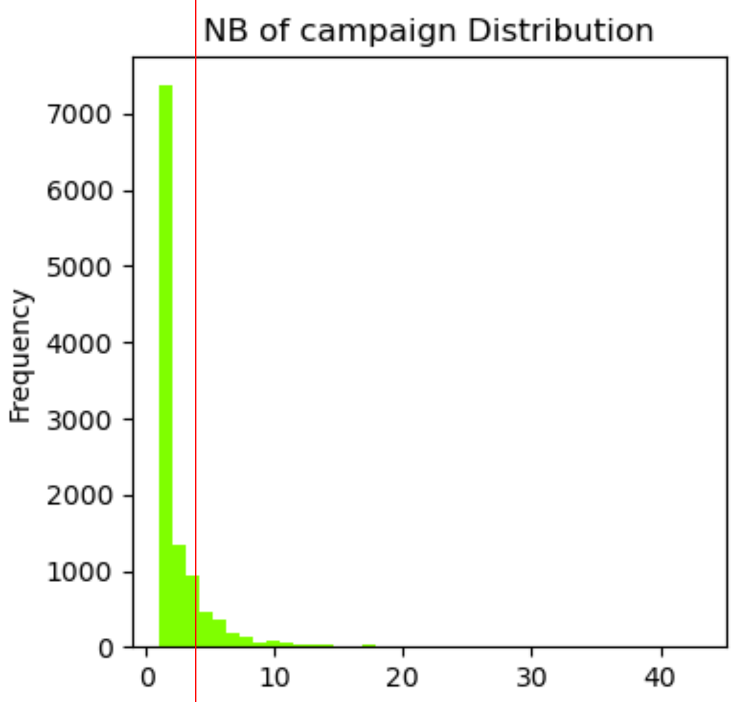
```
columns #13: campaign  is type: <class 'str'>
Missing values 8

count    11214.000000
mean         2.737739
std          2.854410
min          1.000000
25%          1.000000
50%          2.000000
75%          3.000000
max         43.000000
Name: campaign, dtype: float64
```

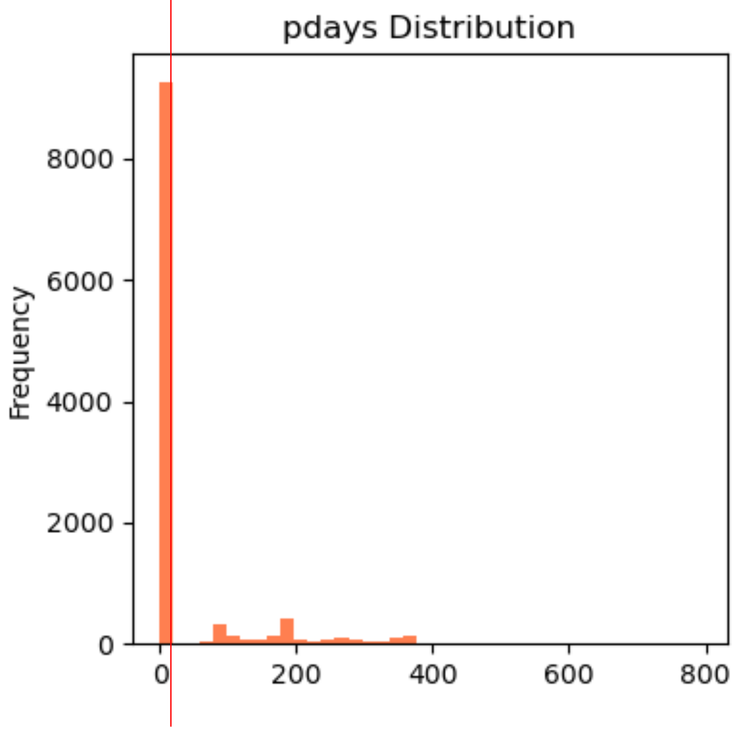


```
columns #14: pdays  is type: <class 'str'>
Missing values 8

count    11214.000000
mean        35.118245
std         90.776604
min         -1.000000
25%         -1.000000
50%         -1.000000
75%         -1.000000
max        792.000000
Name: pdays, dtype: float64
```

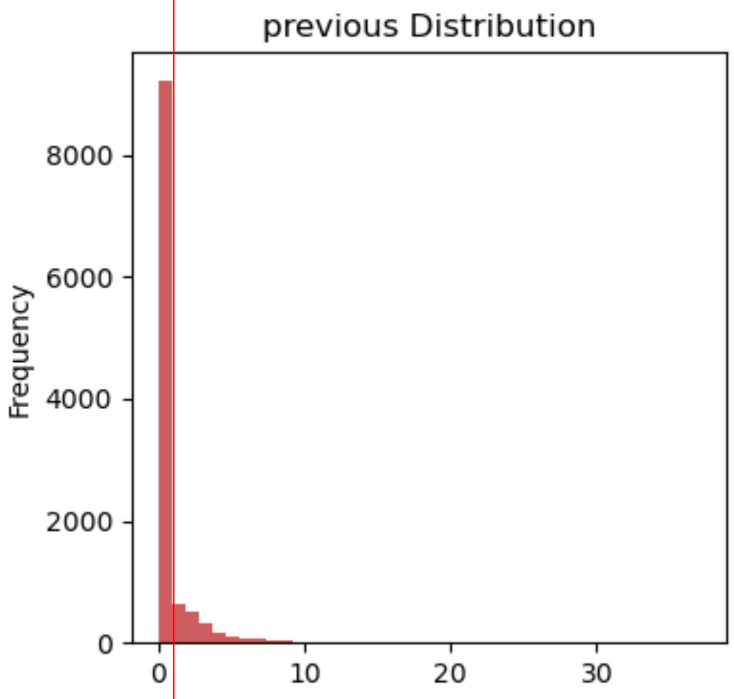


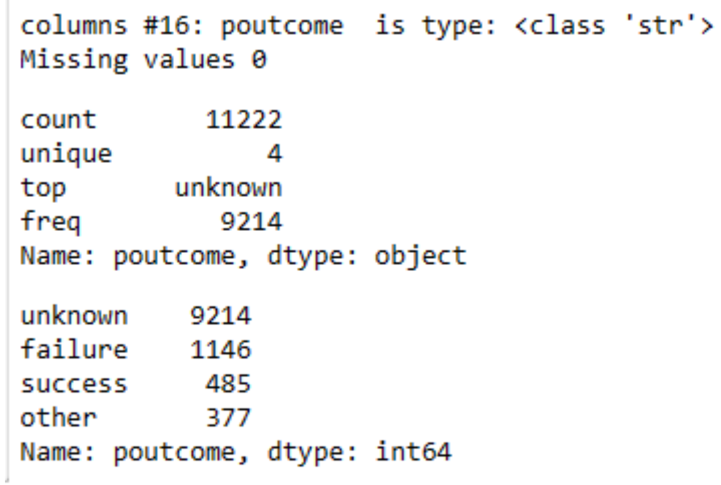
```
columns #15: previous  is type: <class 'str'>
Missing values 6

count    11216.000000
mean         0.558934
std          1.741345
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         37.000000
Name: previous, dtype: float64
```



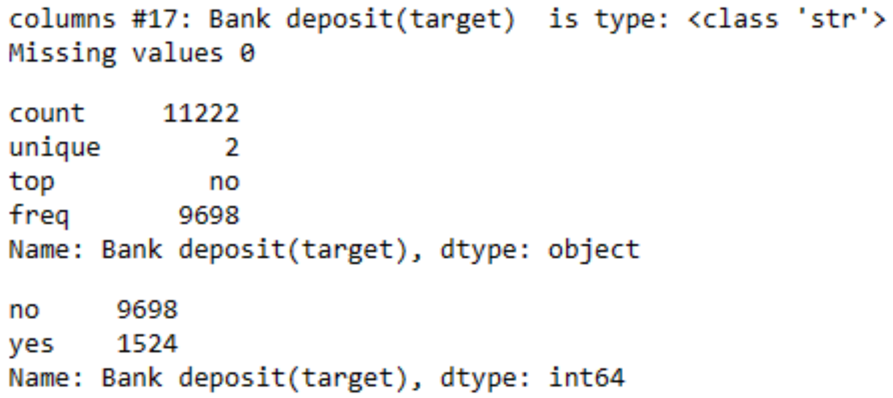


15 - previous (6 lignes ares full empty and will be dropped)



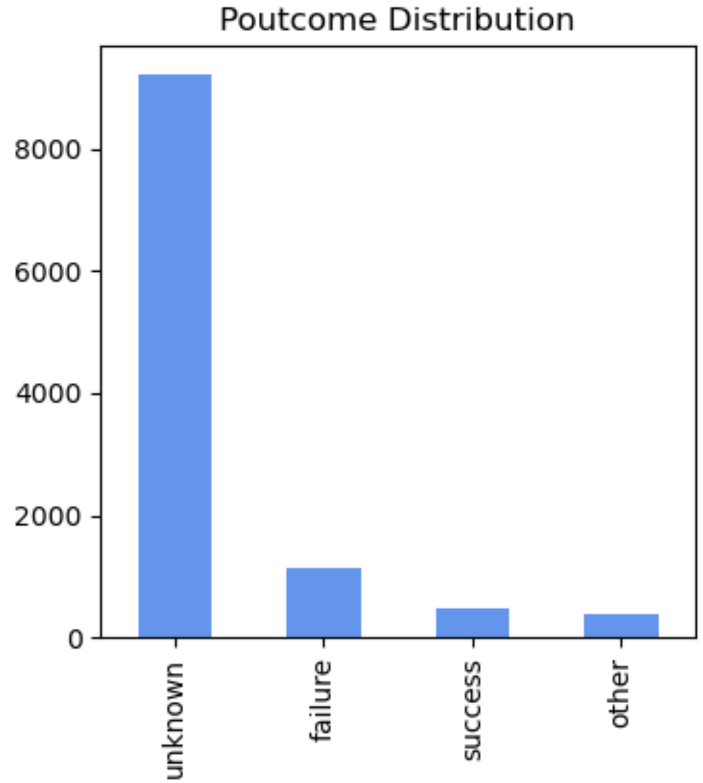


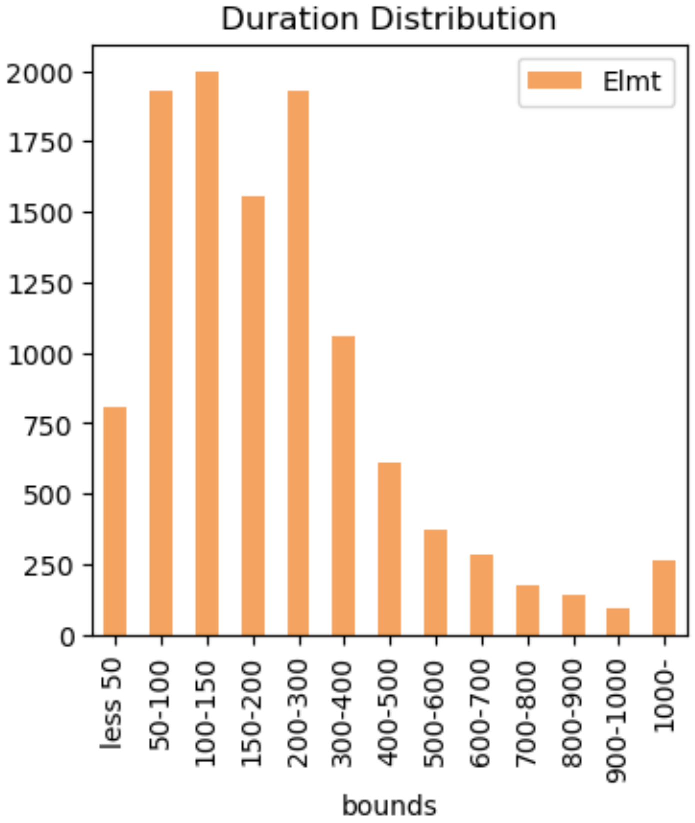
```
columns #16: poutcome  is type: <class 'str'>
Missing values 0

count       11222
unique          4
top       unknown
freq         9214
Name: poutcome, dtype: object

unknown    9214
failure    1146
success     485
other       377
Name: poutcome, dtype: int64
```

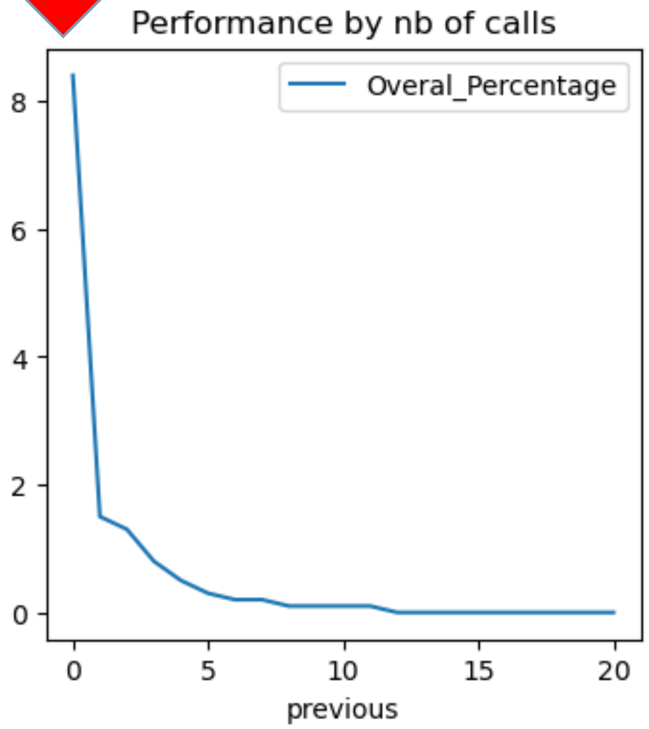


```
columns #17: Bank deposit(target)  is type: <class 'str'>
Missing values 0

count     11222
unique        2
top          no
freq       9698
Name: Bank deposit(target), dtype: object

no     9698
yes    1524
Name: Bank deposit(target), dtype: int64
```

17 Bank deposit(target) Categorical, convert it to True/False







PROCESS

Analysis of each column

```
#Step 0: 6 lignes ares full empty and will be dropped
df = df.drop(df[df['marital'].isna()].index)


#1 - age (numeric) is in string format and should be convert to integer (age)
df['age'] = df['age'].astype('Int64')


# 3 - marital : 1 bad imput 'DIV' must be relace by 'divorced', really low impact,
#We encode it in a new column as a numerical category {'married': 1, 'divorced' : -1, 'single' : 0}

df.loc[df['marital'] == 'DIV','marital'] = 'divorced'
df["n_marital"] = df["marital"].map({'married': 1, 'divorced' : -1, 'single' : 0})
```

```
#4 - education
#27 missing value, we'll try to impute it using KNN method

#717+1 unknown (6%), we will see wich impute strategy is best by testing them
#2 bad imput hjkl -> unknown, Tertiary -> tertiary
#df.loc[(df['education'].str.strip() == 'primary') & (df['education'].str.strip() == 'secondary') & (df['education'].str.strip() == 'tertiary') & (df['education'].str.strip() == 'unknown')]#,'education'] = 'tertiary'
df.loc[df['education'] == 'hjkl','education'] = 'unknown'


#df["education"].str.replace(r'(.*Terti.*)','tertiary', regex=True,)
```

```
# 4 - education (categorical:
"unknown","secondary","primary","tertiary")
print('columns #4:', df.columns[3] , ' is type:', type(
df.columns[3]) )
print('Missing values',df['education'].isna().sum())
display(df['education'].describe())
print(df['education'].value_counts() )
```

```
#5 - default
#Categorical, convert it to True/False
df['b_default'] = df['default'] == 'yes'
```

```
#6 - balance is string, need to be converted to float(2)
df['balance'] = df['balance'].astype(float)
```

```
# 7 - housing: has housing loan? (binary: "yes","no")
df['b_housing'] = df['housing'] == 'yes'
```

```
#8 - loan Categorical, convert it to True/False
#Missing values 12 ( - 6 dropped ligne) = 6 low impact.
We impute then using most frequent n
df.loc[df['loan'].isna(),'loan'] = 'no'
df['b_loan'] = df['loan'] == 'yes'
```

```
#9 - contact:
# 1 Bad input  ghjk -> unknown
df.loc[df['contact'] == 'ghjk','contact'] = 'unknown'
```

```
#10 - day  part of date, need to be concat
with month in a date
# 11 - month
# 2 missing val We impute the 2 ligne using
most frequent : may
df.loc[df['month'].isna(),'month'] = 'may'
```

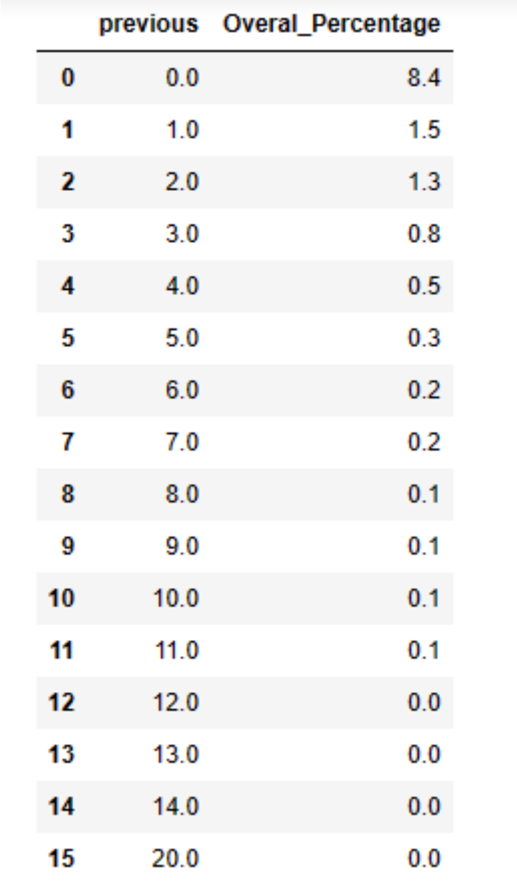
```
#12 - duration
# 8 - 6 Missing values (6 lignes ares full empty and will be dropped); We impute the 2 ligne using mean = 2040 (before removing 2E7 values) after 257.2:
# Max is 20 000 000s = > 231 days  let's investigate , 2nd max = 4918s ->  20 000 000 value will have the mean assigned

df = df.drop(df[df['duration'] == 20000000].index)
df.loc[df['duration'].isna(),'duration'] = 257


#13 - campaign
#We impute the 2 ligne using mean 3
df.loc[df['campaign'].isna(),'campaign'] = 3


#14 - pdays:
#We impute the 2 ligne using mean 3
df.loc[df['pdays'].isna(),'pdays'] = 35
```

```
#15 - previous
#16 - poutcome
#We can try to encode it as a numerical category {'unknown': 0, 'success' : 1, 'failure' : -1, 'other' : 0}
df["n_poutcome"] = df["poutcome"].map({'unknown': 0, 'success' : 1, 'failure' : -1, 'other' : 0})
```

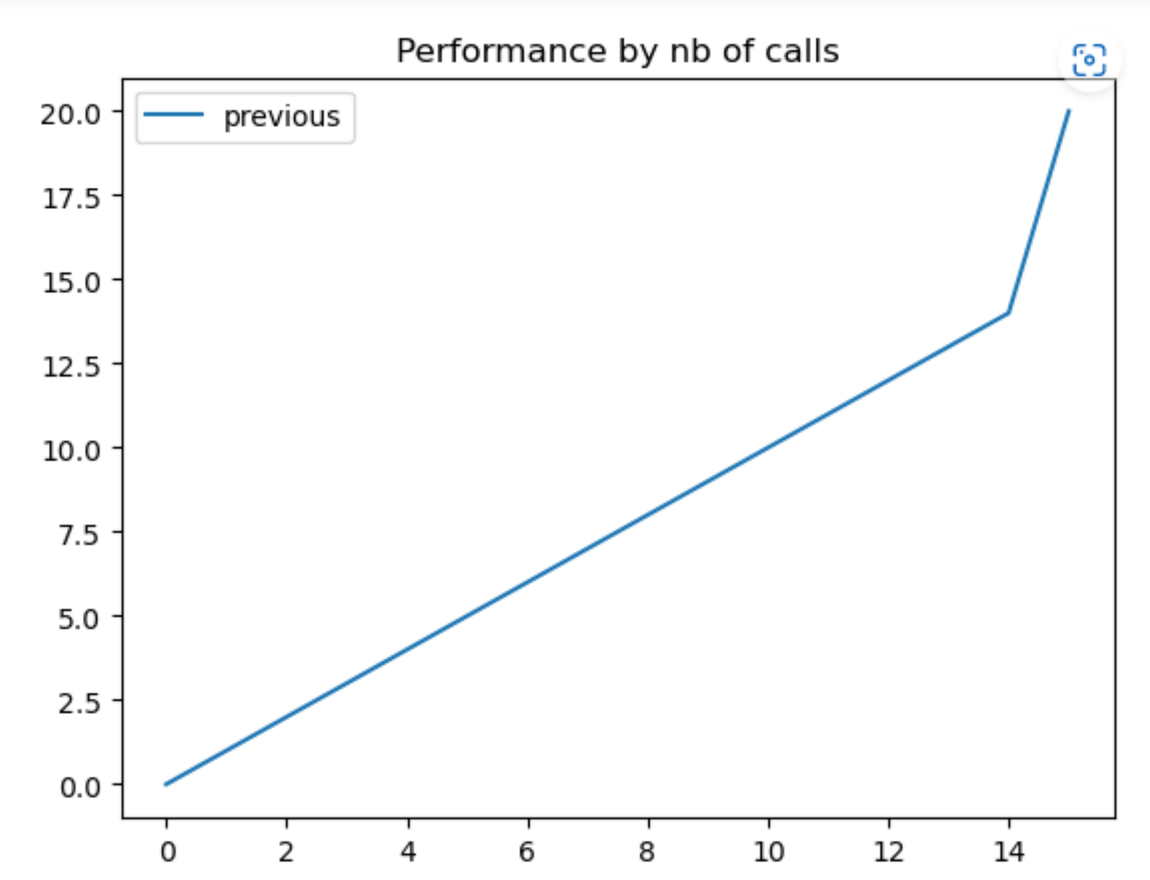
```
#17 Bank deposit(target)
#Categorical, convert it to True/False
df['b_deposit'] = df['Bank deposit(target)'] == 'yes'
```

USING SQLALCHEMY TO CONNECT TO DATABASE

#Read the view v_performance_by_nb_call



	previous	Overal_Percentage
0	0.0	8.4
1	1.0	1.5
2	2.0	1.3
3	3.0	0.8
4	4.0	0.5
5	5.0	0.3
6	6.0	0.2
7	7.0	0.2
8	8.0	0.1
9	9.0	0.1
10	10.0	0.1
11	11.0	0.1
12	12.0	0.0
13	13.0	0.0
14	14.0	0.0
15	20.0	0.0



THANK YOU