Introduction

The data is related with direct marketing campaigns of a Portuguese banking institution. The classification goal is to predict if the client will or will not subscribe to a term deposit (variable y).

Reading Dataset ¶

In [1]:

```
path="C:\Users\user\Desktop\Namita Chhibba\Semester III\Practical Data Science\Assignment 1"
```

In [2]:

```
import pandas as pd
import numpy as np
```

In [3]:

```
bank = pd.read_csv(path+"/part1.csv",delimiter=";", index_col=0) # reading the new csv file
```

In [4]:

```
bank['day_of_week'] = map(lambda x: x.lower(), bank['day_of_week'])
bank['day_of_week'].head(5)
```

Out[4]:

```
0    fri
1    fri
2    wed
3    fri
4    mon
Name: day_of_week, dtype: object
```

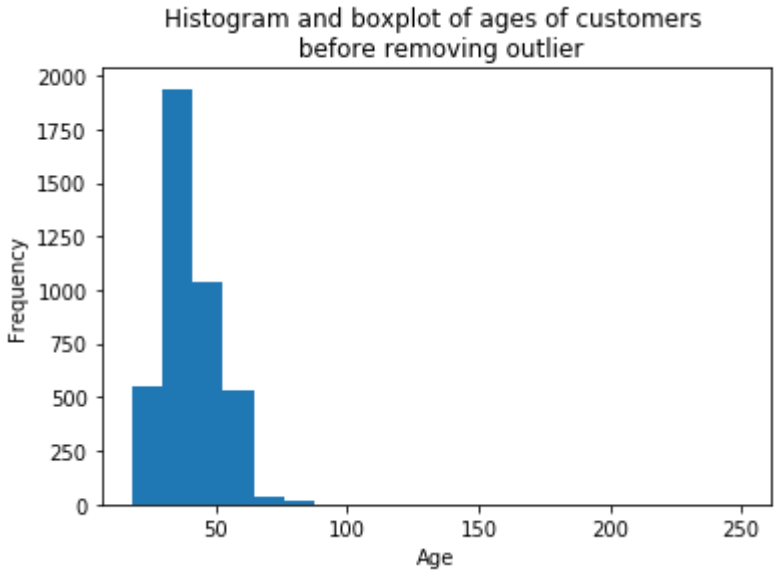
Data Cleaning and Exploration

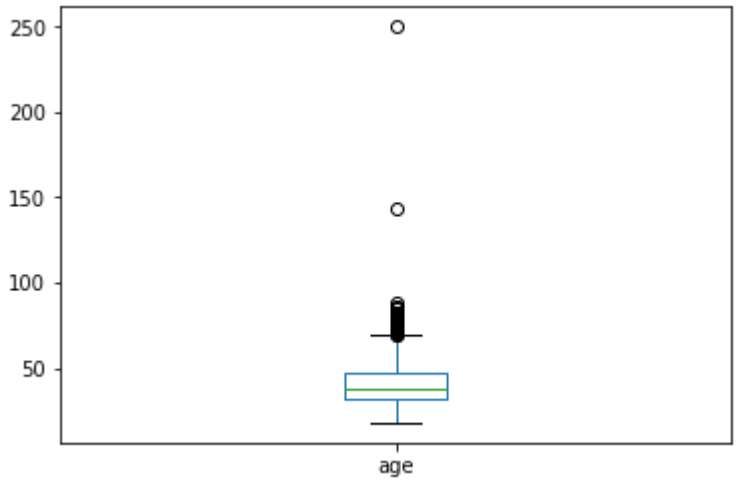
In [58]:

```
import matplotlib.pyplot as plt
```

In [7]:

```
bank['age'].plot(kind='hist',bins=20)
plt.title('Histogram and boxplot of ages of customers \n before removing outlier')
plt.xlabel('Age')
plt.show()

bank['age'].plot(kind='box')
#plt.title('Boxplot of ages of customers')
plt.show()
```





In [8]:

```
b= bank.sort_values(['age'], ascending=[False])
b['age'].head(10)
```

Out[8]:

```
394     250.0
385     143.0
1215     88.0
1796     86.0
696      86.0
1123     85.0
3549     82.0
150      82.0
4067     81.0
4066     81.0
Name: age, dtype: float64
```

In [9]:

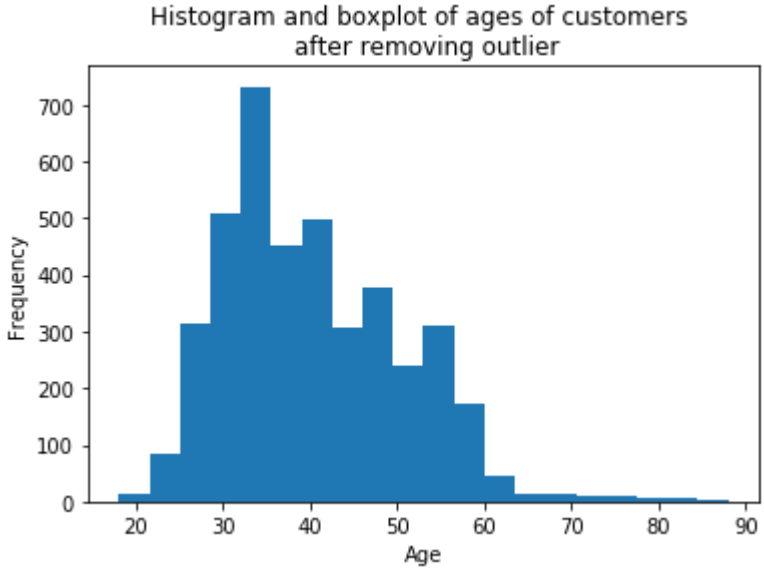
```
# Removing impossible age values and plotting again.
mask_age = bank['age'] <=100
bank['Age']=bank.loc[mask_age,'age']
bank['Age'].describe()
```

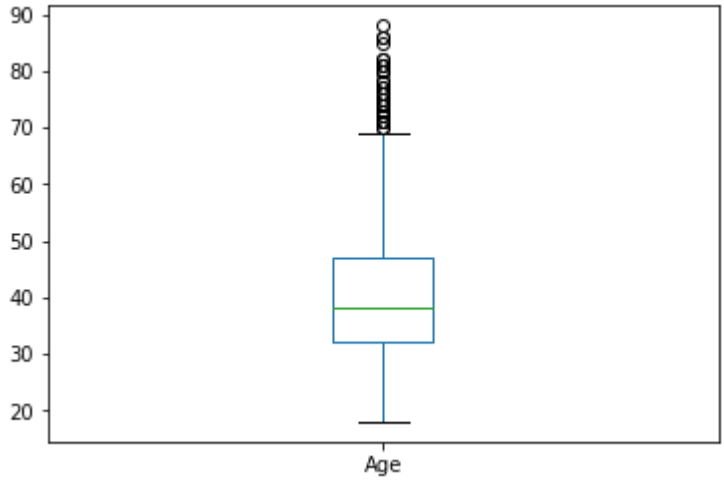
Out[9]:

```
count    4117.000000
mean       40.112598
std        10.314219
min        18.000000
25%        32.000000
50%        38.000000
75%        47.000000
max        88.000000
Name: Age, dtype: float64
```

In [10]:

```
bank['Age'].plot(kind='hist',bins=20)
plt.title('Histogram and boxplot of ages of customers \n after removing outlier')
plt.xlabel('Age')
plt.show()

bank['Age'].plot(kind='box')
# plt.title('Boxplot of ages of customers \n after removing impossible values')
plt.show()

bank['Age'].describe()
```



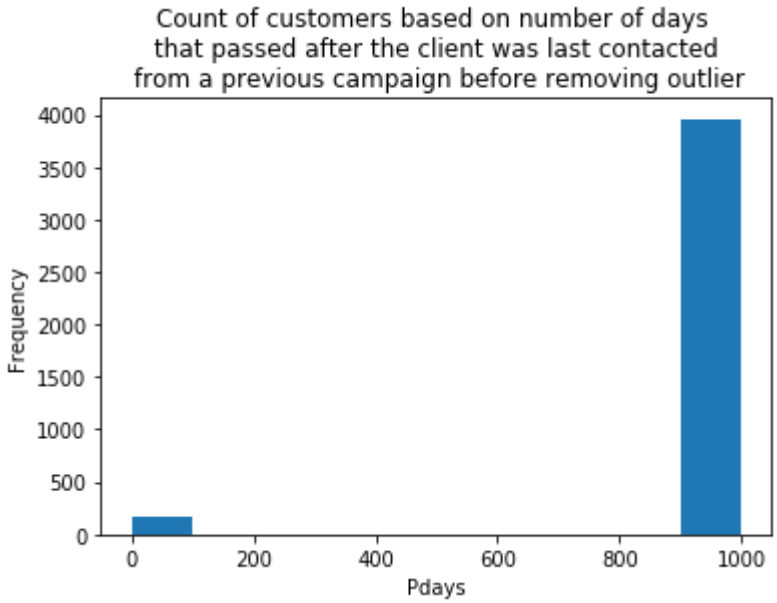


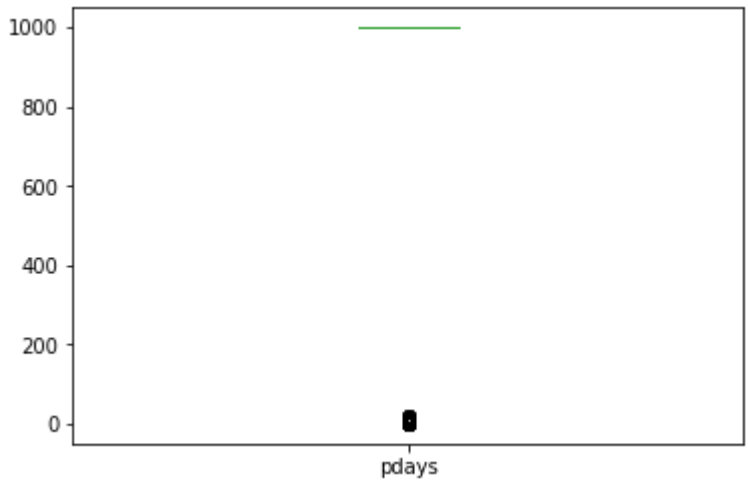
Out[10]:

```
count    4117.000000
mean       40.112598
std        10.314219
min        18.000000
25%        32.000000
50%        38.000000
75%        47.000000
max        88.000000
Name: Age, dtype: float64
```

In [11]:

```
bank['pdays'].plot(kind='hist')
plt.title('Count of customers based on number of days \n that passed after the client w
as last contacted \n from a previous campaign before removing outlier')
plt.xlabel('Pdays')
plt.show()
bank['pdays'].plot(kind='box')
#plt.title('Number of days that passed by after the client was \n last contacted from a
previous campaign for different customers')
plt.show()

bank['pdays'].describe()
```





Out[11]:

```
count    4119.000000
mean      960.422190
std       191.922786
min         0.000000
25%       999.000000
50%       999.000000
75%       999.000000
max       999.000000
Name: pdays, dtype: float64
```

In [12]:

```
b= bank.sort_values(['pdays'], ascending=[False])
b['pdays'].head(10)
```

Out[12]:

```
0       999
2722    999
2708    999
2710    999
2711    999
2712    999
2713    999
2714    999
2715    999
2716    999
Name: pdays, dtype: int64
```

In [13]:

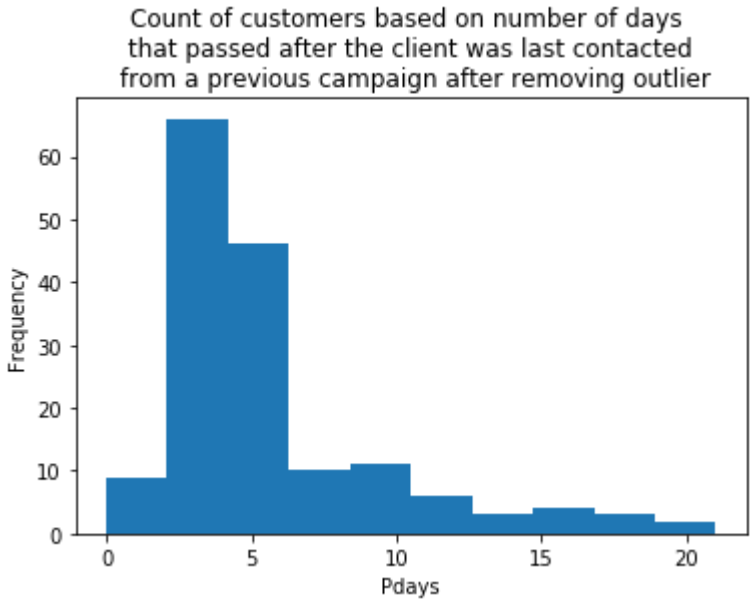
```
# As 999 is the data entry corresponding to the client not previously contacted. So we
 remove it and check the plot for remaining
# values.
mask_pdays = bank['pdays'] <999
bank['Pdays']=bank.loc[mask_pdays,'pdays']
bank['Pdays'].describe()
```

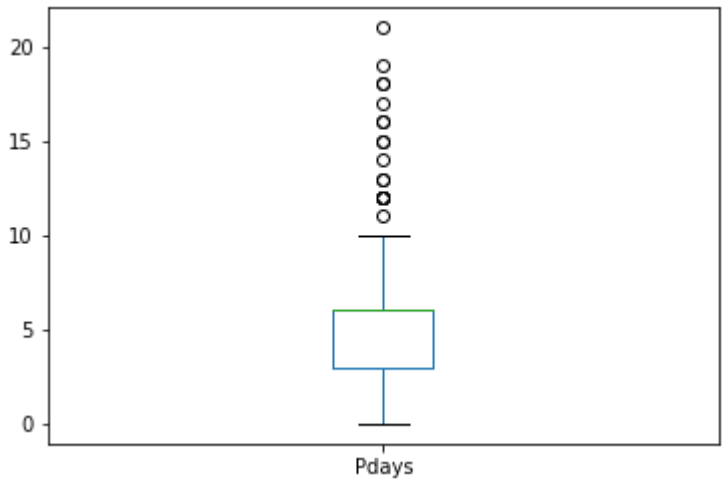
Out[13]:

```
count    160.000000
mean       5.862500
std        3.911743
min        0.000000
25%        3.000000
50%        6.000000
75%        6.000000
max       21.000000
Name: Pdays, dtype: float64
```

In [14]:

```
bank['Pdays'].plot(kind='hist')
plt.title('Count of customers based on number of days \n that passed after the client w
as last contacted \n from a previous campaign after removing outlier')
plt.xlabel('Pdays')
plt.show()
bank['Pdays'].plot(kind='box')
#plt.title('Number of days that passed by after the client was \n last contacted from a
previous campaign for different customers')
plt.show()

bank['Pdays'].describe()
```



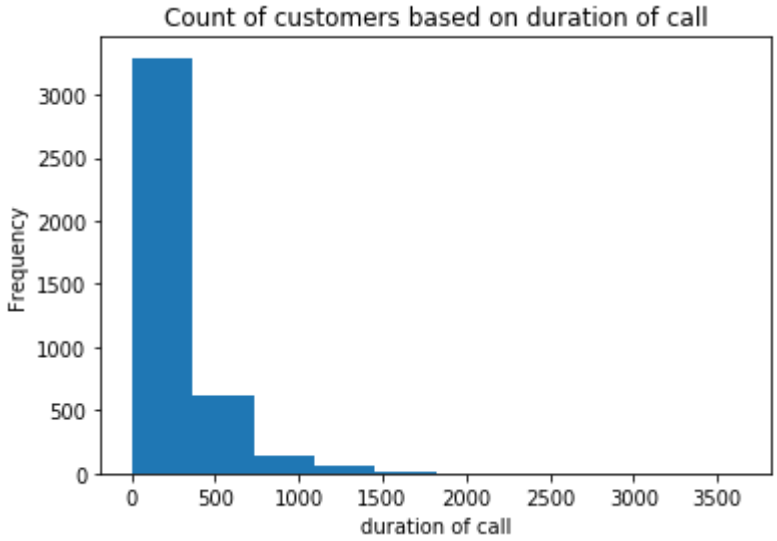


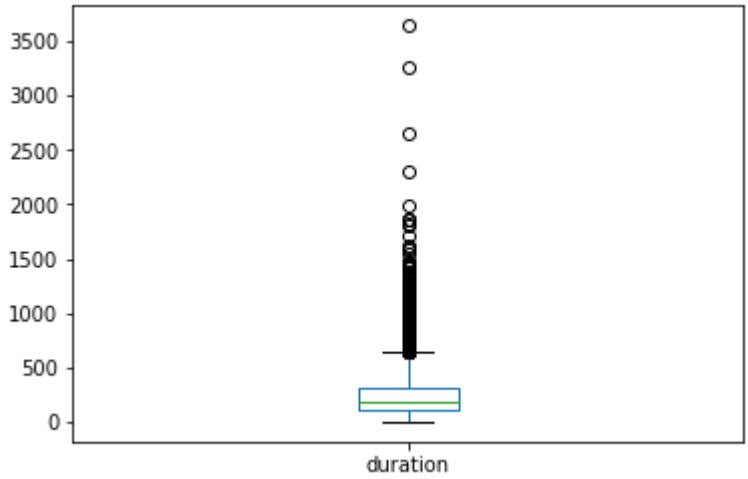
Out[14]:

```
count    160.000000
mean       5.862500
std        3.911743
min        0.000000
25%        3.000000
50%        6.000000
75%        6.000000
max       21.000000
Name: Pdays, dtype: float64
```

In [15]:

```
bank['duration'].plot(kind='hist')
plt.title('Count of customers based on duration of call')
plt.xlabel('duration of call')
plt.show()
bank['duration'].plot(kind='box')
#plt.title('Number of days that passed by after the client was \n last contacted from a previous campaign for different customers')
plt.show()

bank['duration'].describe()
```





Out[15]:

```
count    4119.000000
mean      256.754978
std       254.694889
min         0.000000
25%       103.000000
50%       181.000000
75%       317.000000
max      3643.000000
Name: duration, dtype: float64
```

In [16]:

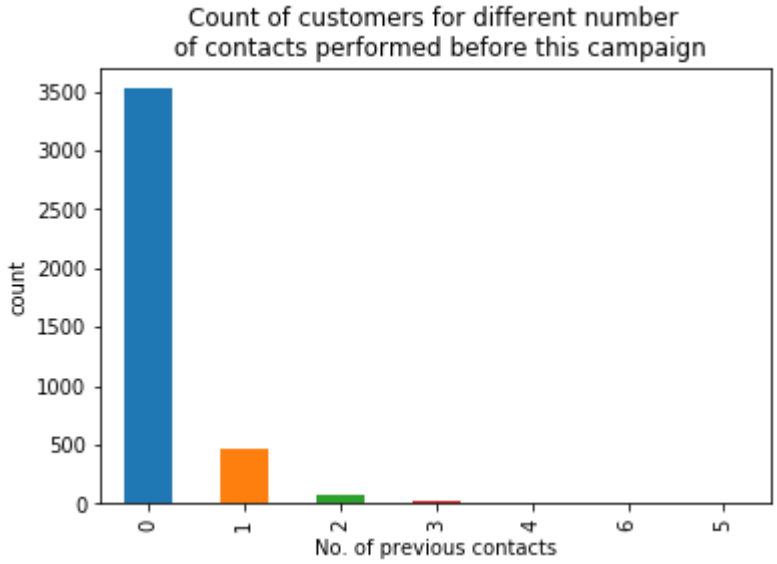
```
b= bank.sort_values(['duration'], ascending=[False])
b['duration'].head(10)
```

Out[16]:

```
2231    3643.0
1546    3253.0
1392    2653.0
1685    2301.0
3266    1980.0
262     1868.0
2530    1855.0
2900    1855.0
3256    1820.0
2957    1806.0
Name: duration, dtype: float64
```

In [17]:

```
bank['previous'].value_counts().plot(kind='bar')
plt.title('Count of customers for different number \n of contacts performed before this campaign')
plt.xlabel('No. of previous contacts')
plt.ylabel('count')
plt.show()

bank['previous'].describe()
```

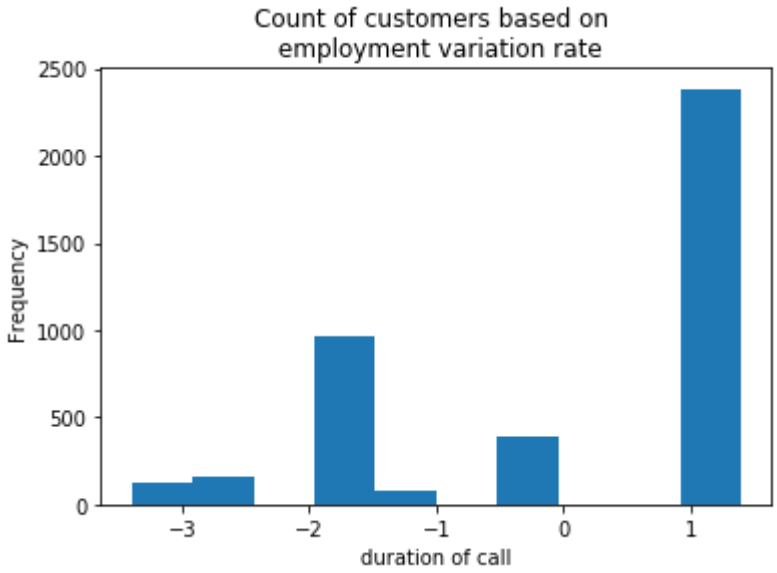


Out[17]:

```
count    4119.000000
mean        0.190337
std         0.541788
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         6.000000
Name: previous, dtype: float64
```

In [18]:

```
bank['emp.var.rate'].plot(kind='hist')
plt.title('Count of customers based on \n employment variation rate')
plt.xlabel('duration of call')
plt.show()

bank['emp.var.rate'].describe()
```

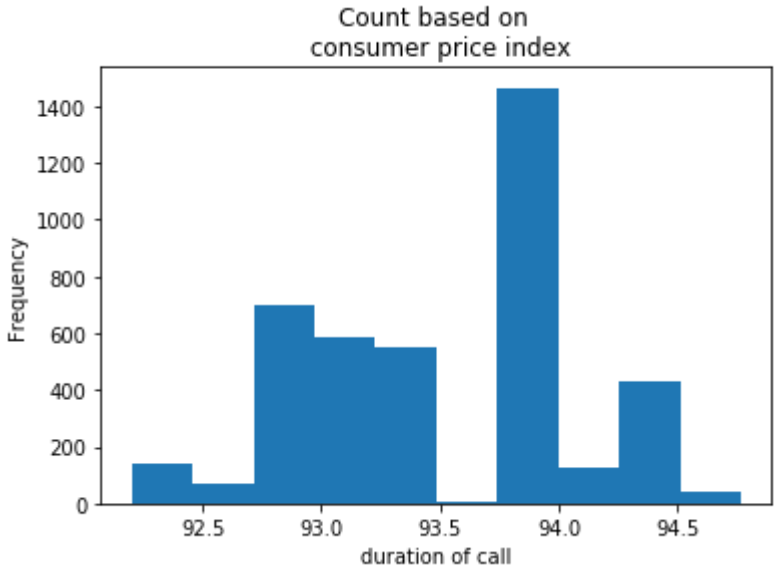


Out[18]:

```
count    4119.000000
mean        0.084972
std         1.563114
min        -3.400000
25%        -1.800000
50%         1.100000
75%         1.400000
max         1.400000
Name: emp.var.rate, dtype: float64
```

In [19]:

```
bank['cons.price.idx'].plot(kind='hist')
plt.title('Count based on \n consumer price index')
plt.xlabel('duration of call')
plt.show()

bank['cons.price.idx'].describe()
```

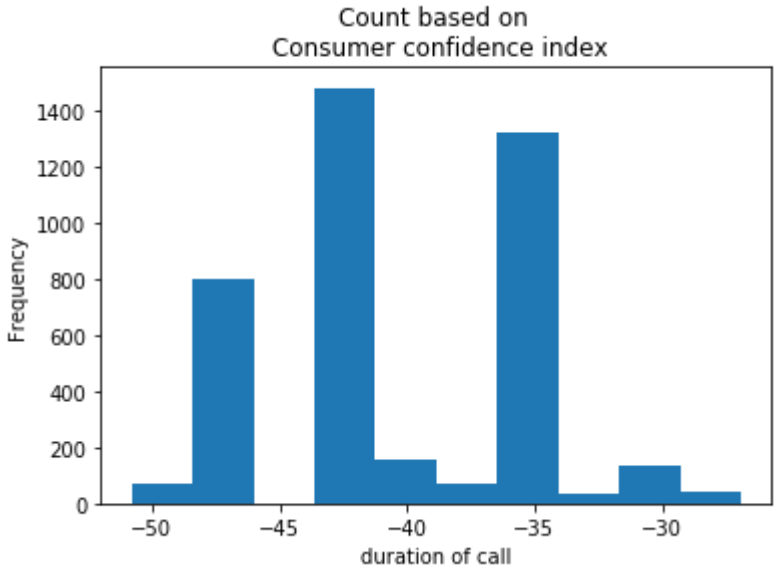


Out[19]:

```
count    4119.000000
mean       93.579421
std         0.579253
min        92.201000
25%        93.075000
50%        93.749000
75%        93.994000
max        94.767000
Name: cons.price.idx, dtype: float64
```

In [20]:

```
bank['cons.conf.idx'].plot(kind='hist')
plt.title('Count based on \n Consumer confidence index')
plt.xlabel('duration of call')
plt.show()

bank['cons.conf.idx'].describe()
```

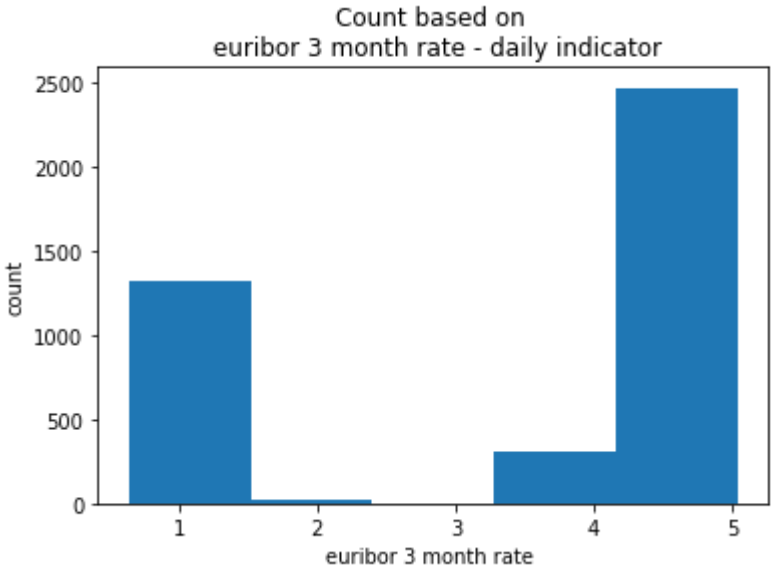


Out[20]:

```
count    4119.000000
mean      -40.499102
std         4.594578
min       -50.800000
25%       -42.700000
50%       -41.800000
75%       -36.400000
max       -26.900000
Name: cons.conf.idx, dtype: float64
```

In [21]:

```
bank['euribor3m'].plot(kind='hist', bins=5)
plt.title('Count based on \n euribor 3 month rate - daily indicator')
plt.xlabel('euribor 3 month rate ')
plt.ylabel('count')
plt.show()

bank['euribor3m'].describe()
```

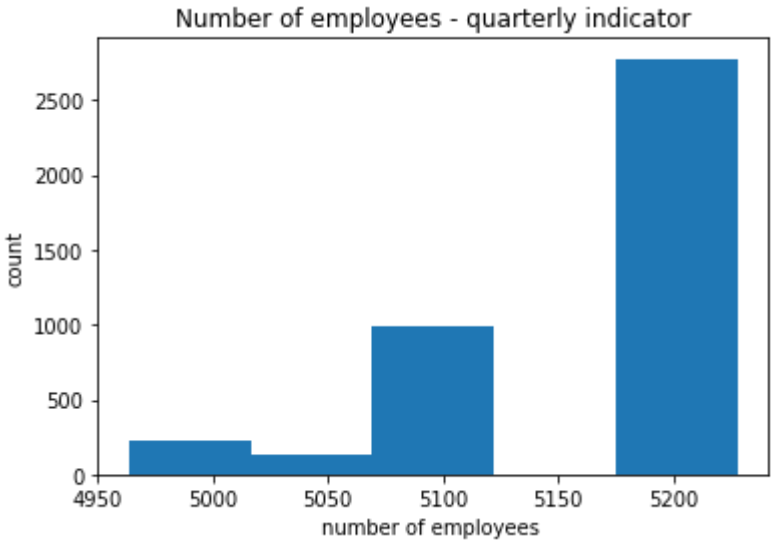


Out[21]:

```
count    4119.000000
mean        3.621787
std         1.733198
min         0.635000
25%         1.334000
50%         4.857000
75%         4.961000
max         5.045000
Name: euribor3m, dtype: float64
```

In [22]:

```
bank['nr.employed'].plot(kind='hist', bins=5)
plt.title('Number of employees - quarterly indicator')
plt.xlabel('number of employees ')
#plt.xlim(0,6)
#plt.axis([0, 6, min(y), max(y)])
plt.ylabel('count')
plt.show()

bank['nr.employed'].describe()
```



Out[22]:

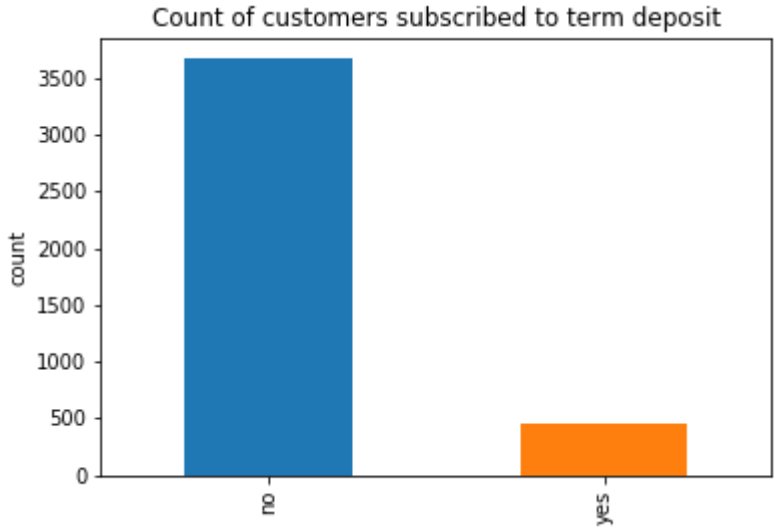
```
count    4119.000000
mean     5166.536386
std        73.619352
min      4963.600000
25%      5099.100000
50%      5191.000000
75%      5228.100000
max      5228.100000
Name: nr.employed, dtype: float64
```

Count of customers based on various variables

Based on Term Deposit

In [23]:

```
bank['y'].value_counts().plot(kind='bar')
plt.title('Count of customers subscribed to term deposit')
plt.ylabel('count')
plt.show()

bank['y'].describe()
```



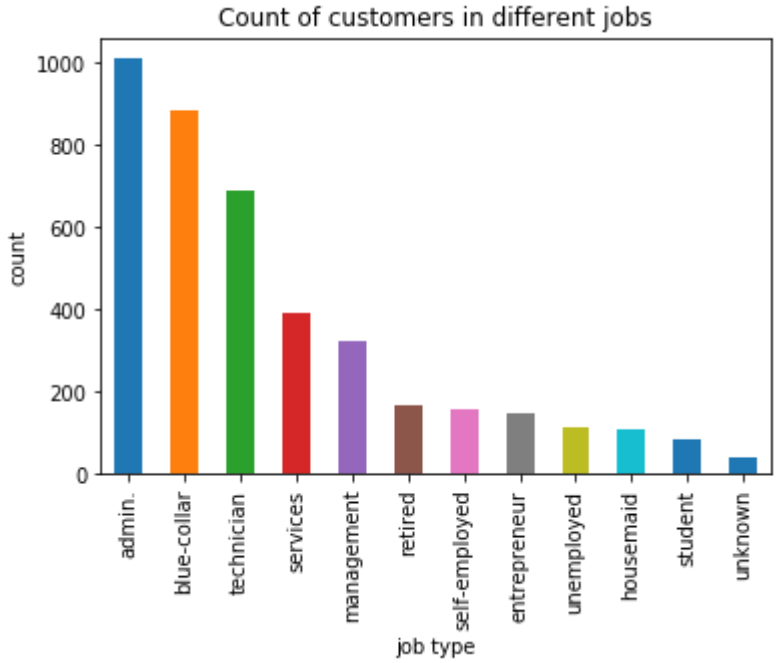
Out[23]:

```
count     4119
unique       2
top         no
freq      3668
Name: y, dtype: object
```

Based on different jobs

In [24]:

```
bank['job'].value_counts().plot(kind='bar')
plt.title('Count of customers in different jobs')
plt.xlabel('job type')
plt.ylabel('count')
plt.show()

bank['job'].describe()
```



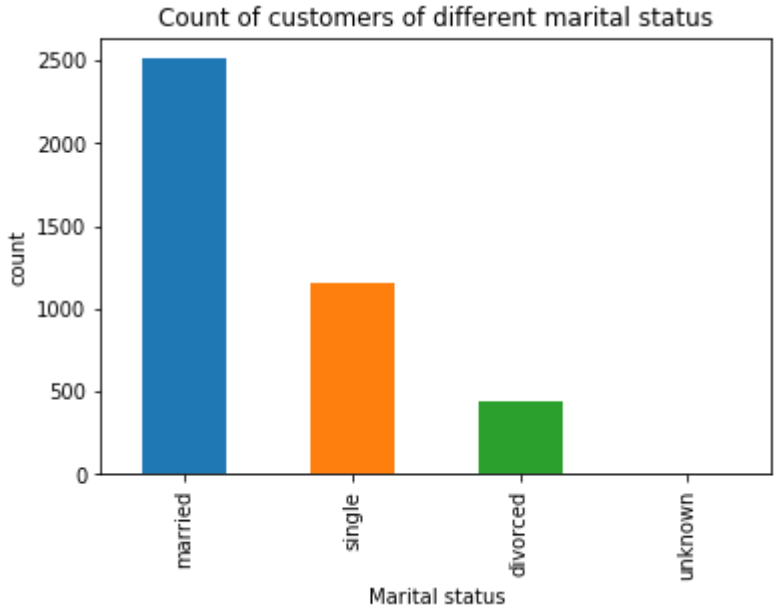
Out[24]:

```
count       4119
unique        12
top       admin.
freq        1012
Name: job, dtype: object
```

Based on Marital Status

In [25]:

```
bank['marital'].value_counts().plot(kind='bar')
plt.title('Count of customers of different marital status')
plt.xlabel('Marital status')
plt.ylabel('count')
plt.show()

bank['marital'].describe()
```



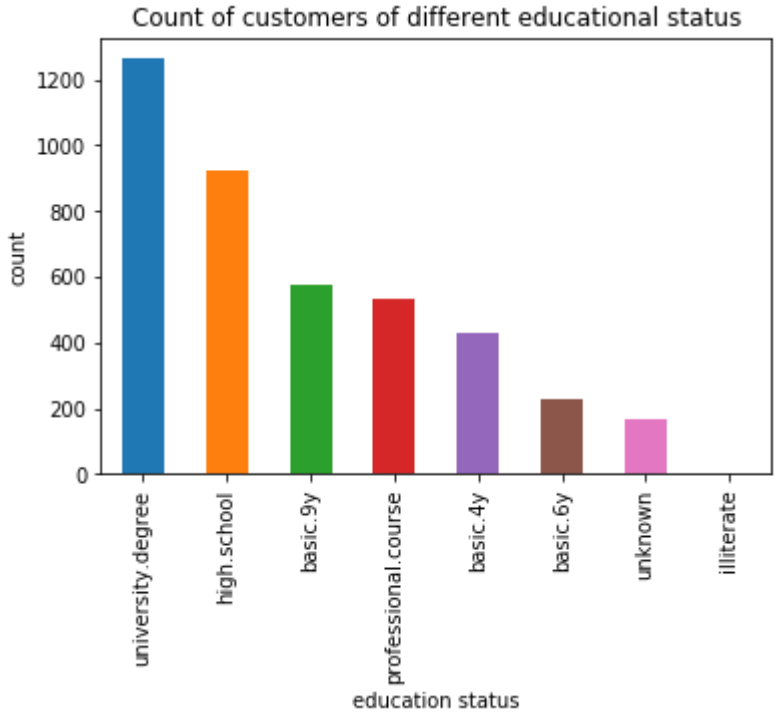
Out[25]:

```
count        4119
unique          4
top       married
freq         2509
Name: marital, dtype: object
```

Based on Educational Status

In [26]:

```
bank['education'].value_counts().plot(kind='bar')
plt.title('Count of customers of different educational status')
plt.xlabel('education status')
plt.ylabel('count')
plt.show()

bank['education'].describe()
```



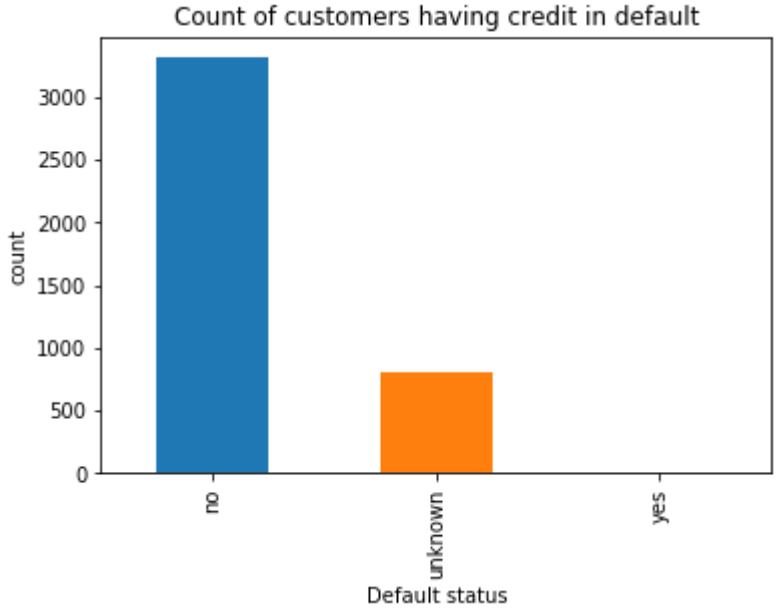
Out[26]:

```
count                 4119
unique                   8
top       university.degree
freq                  1264
Name: education, dtype: object
```

Based on Credit

In [27]:

```
bank['default'].value_counts().plot(kind='bar')
plt.title('Count of customers having credit in default')
plt.xlabel('Default status')
plt.ylabel('count')
plt.show()

bank['default'].describe()
```



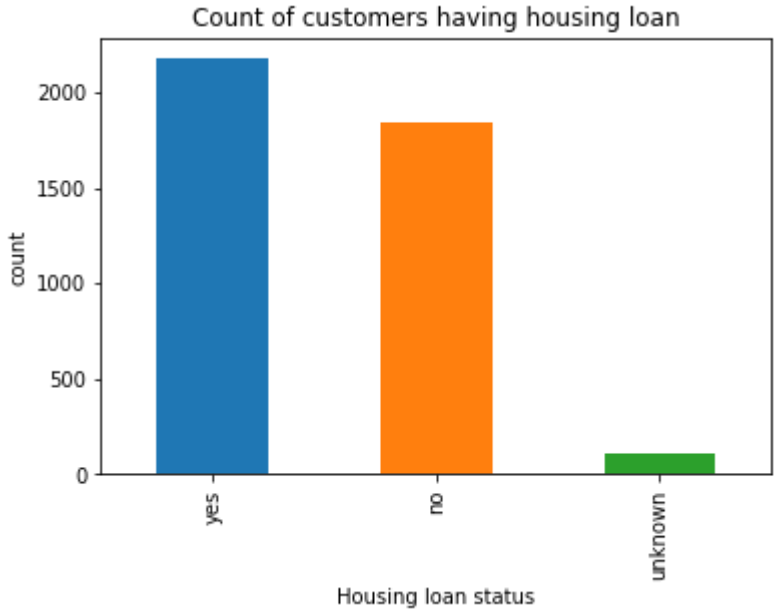
Out[27]:

```
count     4119
unique       3
top         no
freq      3315
Name: default, dtype: object
```

Based on Housing Loan

In [28]:

```
bank['housing'].value_counts().plot(kind='bar')
plt.title('Count of customers having housing loan')
plt.xlabel('Housing loan status')
plt.ylabel('count')
plt.show()

bank['housing'].describe()
```



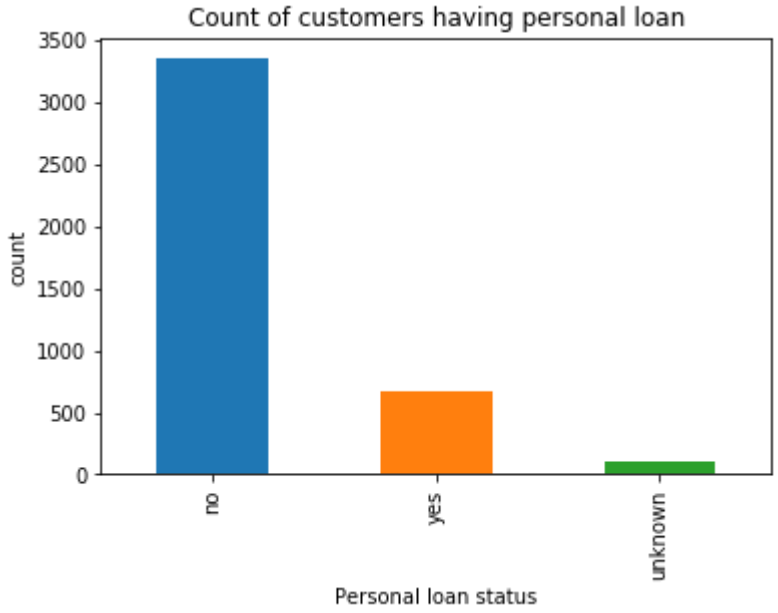
Out[28]:

```
count     4119
unique       3
top        yes
freq      2175
Name: housing, dtype: object
```

Based on Personal Loan

In [29]:

```
bank['loan'].value_counts().plot(kind='bar')
plt.title('Count of customers having personal loan')
plt.xlabel('Personal loan status')
plt.ylabel('count')
plt.show()

bank['loan'].describe()
```



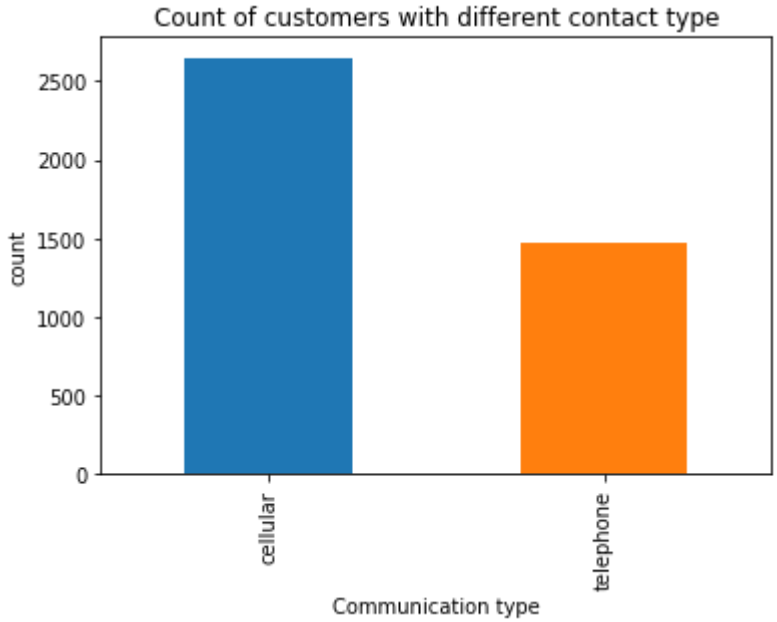
Out[29]:

```
count     4119
unique       3
top         no
freq      3349
Name: loan, dtype: object
```

Based on Contact Type

In [30]:

```
bank['contact'].value_counts().plot(kind='bar')
plt.title('Count of customers with different contact type')
plt.xlabel('Communication type')
plt.ylabel('count')
plt.show()

bank['contact'].describe()
```



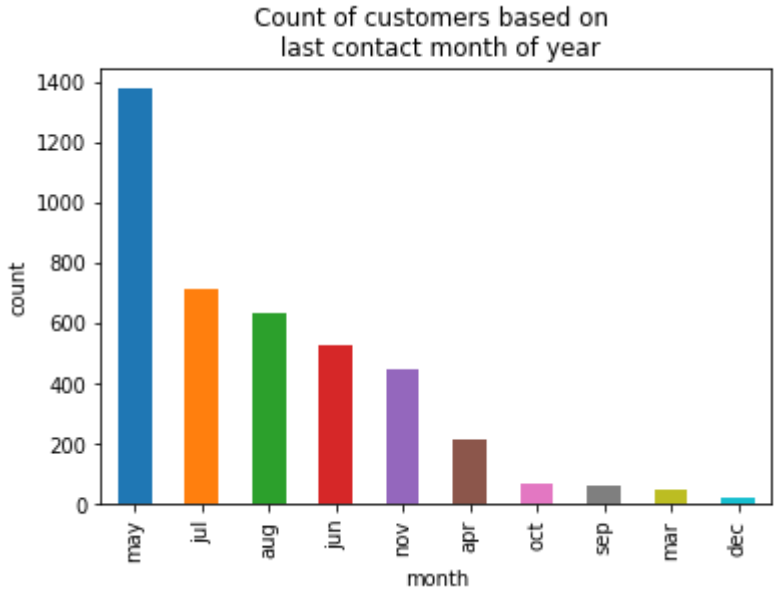
Out[30]:

```
count         4119
unique           2
top       cellular
freq          2652
Name: contact, dtype: object
```

Based on Last Contact Month of Year

In [31]:

```
bank['month'].value_counts().plot(kind='bar')
plt.title('Count of customers based on \n last contact month of year')
plt.xlabel('month')
plt.ylabel('count')
plt.show()

bank['month'].describe()
```



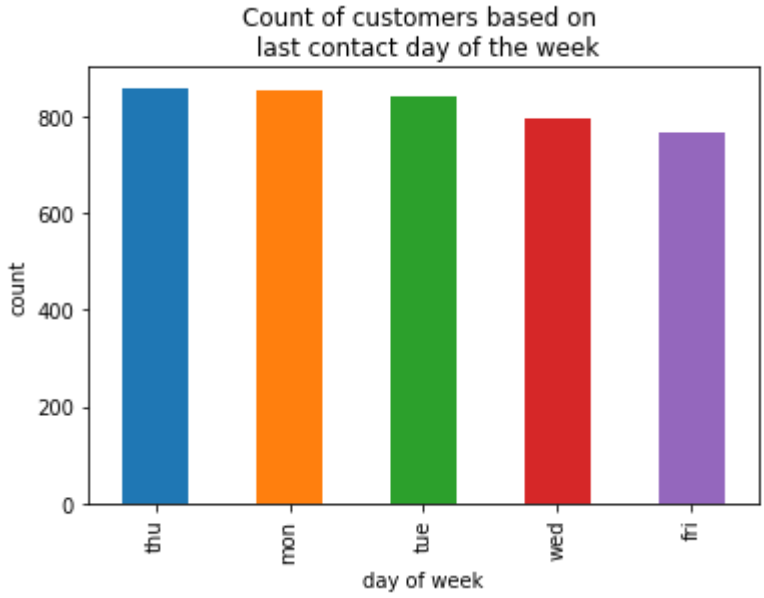
Out[31]:

```
count     4119
unique      10
top        may
freq      1378
Name: month, dtype: object
```

Based on Last Contact Day of Week

In [32]:

```
bank['day_of_week'].value_counts().plot(kind='bar')
plt.title('Count of customers based on \n last contact day of the week')
plt.xlabel('day of week')
plt.ylabel('count')
plt.show()

bank['day_of_week'].describe()
```



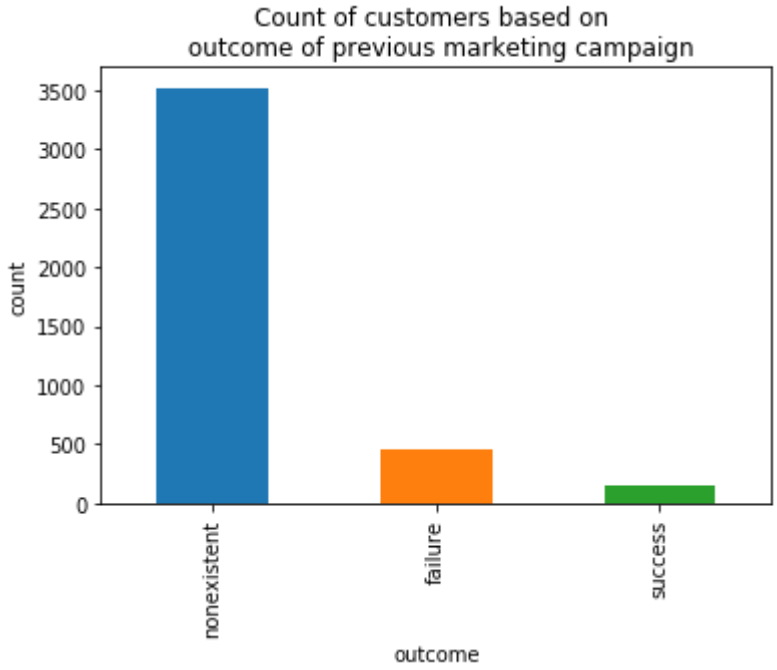
Out[32]:

```
count     4119
unique       5
top        thu
freq       860
Name: day_of_week, dtype: object
```

Based on Outcome of previous marketing campaign

In [33]:

```
bank['poutcome'].value_counts().plot(kind='bar')
plt.title('Count of customers based on \n outcome of previous marketing campaign')
plt.xlabel('outcome')
plt.ylabel('count')
plt.show()

bank['poutcome'].describe()
```



Out[33]:

```
count            4119
unique              3
top       nonexistent
freq             3523
Name: poutcome, dtype: object
```

In [34]:

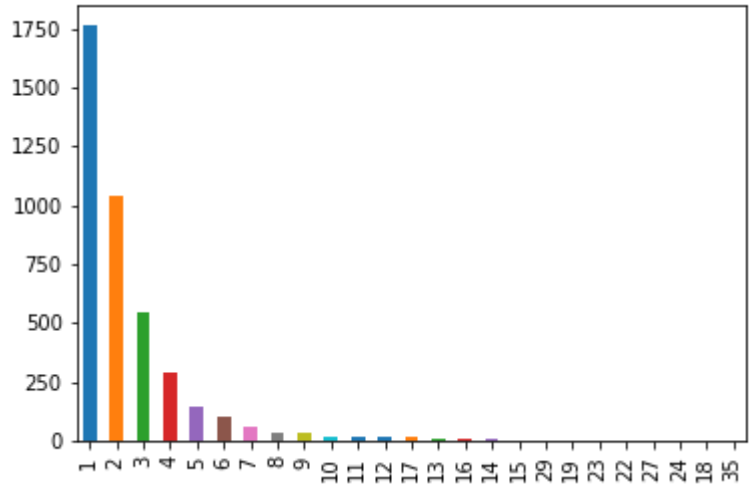
```
#bank.loc[471]['duration']
```

In [35]:

```
bank['campaign'].value_counts().plot(kind='bar')
bank['campaign'].describe()
```

Out[35]:

```
count    4119.000000
mean        2.537266
std         2.568159
min         1.000000
25%         1.000000
50%         2.000000
75%         3.000000
max        35.000000
Name: campaign, dtype: float64
```



Sanity Checks

1. age

In [38]:

```
bank[(bank['age']>100) | (bank['age']<0)]
```

Out[38]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_
385	143.0	unemployed	married	basic.4y	no	no	no	cellular	may	
394	250.0	admin.	married	basic.9y	no	no	no	cellular	aug	

2 rows × 23 columns

2. duration

In [40]:

```
bank[(bank['duration']== 0) & (bank['y'].str.contains('no'))]
# sanity check as per the requirement mentioned in column information
```

Out[40]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_wee
1950	39.0	admin.	married	high.school	no	yes	no	telephone	may	tu

1 rows × 23 columns

3. pdays

In [42]:

```
(bank['pdays'] ==999).sum(axis=0)
# bank.pdays.value_counts()[999]
```

Out[42]:

```
3959
```

In [43]:

```
print bank[bank.pdays == 999]
```

	age	job	marital	education	default	housing
0	30.0	blue-collar	married	basic.9y	no	yes
1	39.0	services	single	high.school	no	no
2	25.0	services	married	high.school	no	yes
3	38.0	services	married	basic.9y	no	unknown
4	47.0	admin.	married	university.degree	no	yes
5	32.0	services	single	university.degree	no	no
6	32.0	admin.	single	university.degree	no	yes
7	41.0	entrepreneur	married	university.degree	unknown	yes
8	31.0	services	divorced	professional.course	no	no
9	35.0	blue-collar	married	basic.9y	unknown	no
10	25.0	services	single	basic.6y	unknown	yes
11	36.0	self-employed	single	basic.4y	no	no
12	36.0	admin.	married	high.school	no	no
13	47.0	blue-collar	married	basic.4y	no	yes
14	29.0	admin.	single	high.school	no	no
15	27.0	services	single	university.degree	no	no
16	44.0	admin.	divorced	university.degree	no	no
17	46.0	admin.	divorced	university.degree	no	yes
18	45.0	entrepreneur	married	university.degree	unknown	yes
19	50.0	blue-collar	married	basic.4y	no	no
20	55.0	services	married	basic.6y	unknown	yes
22	29.0	technician	single	university.degree	no	yes
23	40.0	management	married	high.school	no	no
24	44.0	technician	married	professional.course	unknown	yes
25	38.0	technician	married	professional.course	no	yes
26	36.0	technician	divorced	professional.course	no	no
27	28.0	blue-collar	married	basic.6y	unknown	no
28	47.0	admin.	single	unknown	unknown	no
29	34.0	admin.	married	university.degree	no	no
30	38.0	technician	married	university.degree	no	yes
...	...	...	...	...	...	...
4087	29.0	admin.	married	university.degree	no	yes
4089	25.0	admin.	single	university.degree	no	yes
4090	43.0	blue-collar	married	basic.4y	unknown	yes
4091	38.0	management	married	high.school	unknown	no
4092	30.0	blue-collar	single	high.school	no	no
4093	56.0	retired	married	basic.4y	unknown	no
4095	36.0	admin.	single	university.degree	no	no
4096	33.0	services	married	high.school	no	no
4097	41.0	blue-collar	divorced	basic.9y	no	no
4098	34.0	housemaid	single	university.degree	no	yes
4099	58.0	admin.	divorced	high.school	no	no
4100	41.0	admin.	divorced	high.school	no	no
4101	35.0	entrepreneur	single	university.degree	no	yes
4102	31.0	blue-collar	single	basic.9y	unknown	no
4103	43.0	services	married	high.school	no	no
4104	42.0	technician	divorced	professional.course	no	yes
4105	47.0	housemaid	married	basic.4y	unknown	yes
4106	45.0	entrepreneur	divorced	basic.9y	no	yes
4107	36.0	admin.	married	university.degree	unknown	yes
4108	32.0	admin.	married	university.degree	no	yes
4109	63.0	retired	married	high.school	no	no
4110	53.0	housemaid	divorced	basic.6y	unknown	unknown
4111	30.0	technician	married	university.degree	no	no
4112	31.0	technician	single	professional.course	no	yes
4113	31.0	admin.	single	university.degree	no	yes
4114	30.0	admin.	married	basic.6y	no	yes
4115	39.0	admin.	married	high.school	no	yes
4116	27.0	student	single	high.school	no	no

```
4117  58.0         admin.   married          high.school       no       no
4118  34.0     management    single          high.school       no      yes

         loan    contact month day_of_week  ...    previous     poutcome
\
0          no   cellular   may         fri  ...           0  nonexistent
1          no  telephone   may         fri  ...           0  nonexistent
2          no  telephone   jun         wed  ...           0  nonexistent
3     unknown  telephone   jun         fri  ...           0  nonexistent
4          no   cellular   nov         mon  ...           0  nonexistent
5          no   cellular   sep         thu  ...           2      failure
6          no   cellular   sep         mon  ...           0  nonexistent
7          no   cellular   nov         mon  ...           0  nonexistent
8          no   cellular   nov         tue  ...           1      failure
9          no  telephone   may         thu  ...           0  nonexistent
10         no   cellular   jul         thu  ...           0  nonexistent
11         no   cellular   jul         thu  ...           0  nonexistent
12         no  telephone   may         wed  ...           0  nonexistent
13         no  telephone   jun         thu  ...           0  nonexistent
14         no   cellular   may         fri  ...           0  nonexistent
15         no   cellular   jul         wed  ...           0  nonexistent
16         no   cellular   jul         wed  ...           0  nonexistent
17         no  telephone   jul         mon  ...           0  nonexistent
18        yes   cellular   aug         mon  ...           0  nonexistent
19        yes   cellular   jul         tue  ...           0  nonexistent
20         no   cellular   jul         tue  ...           0  nonexistent
22        yes   cellular   aug         wed  ...           0  nonexistent
23        yes   cellular   aug         wed  ...           0  nonexistent
24         no  telephone   may         fri  ...           0  nonexistent
25         no   cellular   aug         mon  ...           0  nonexistent
26         no  telephone   may         wed  ...           0  nonexistent
27         no   cellular   may         mon  ...           1      failure
28         no  telephone   may         thu  ...           0  nonexistent
29         no   cellular   aug         tue  ...           0  nonexistent
30        yes   cellular   mar         tue  ...           1      failure
...       ...        ...   ...         ...  ...         ...          ...
4087       no  telephone   may         thu  ...           0  nonexistent
4089      yes   cellular   oct         fri  ...           1      failure
4090      yes  telephone   may         tue  ...           0  nonexistent
4091       no  telephone   may         thu  ...           0  nonexistent
4092       no  telephone   jul         wed  ...           0  nonexistent
4093       no   cellular   jul         tue  ...           0  nonexistent
4095      yes   cellular   aug         fri  ...           0  nonexistent
4096       no  telephone   may         mon  ...           0  nonexistent
4097       no   cellular   aug         tue  ...           0  nonexistent
4098       no   cellular   aug         thu  ...           0  nonexistent
4099       no   cellular   aug         tue  ...           0  nonexistent
4100       no   cellular   apr         fri  ...           0  nonexistent
4101       no   cellular   jul         mon  ...           0  nonexistent
4102      yes  telephone   jun         fri  ...           0  nonexistent
4103       no  telephone   may         mon  ...           0  nonexistent
4104       no   cellular   aug         mon  ...           0  nonexistent
4105       no  telephone   jul         tue  ...           0  nonexistent
4106       no   cellular   may         tue  ...           0  nonexistent
4107       no   cellular   aug         wed  ...           0  nonexistent
4108       no  telephone   may         thu  ...           0  nonexistent
4109       no   cellular   oct         wed  ...           0  nonexistent
4110  unknown  telephone   may         fri  ...           0  nonexistent
4111      yes   cellular   jun         fri  ...           1      failure
4112       no   cellular   nov         thu  ...           0  nonexistent
4113       no   cellular   nov         thu  ...           0  nonexistent
```

```
4114      yes   cellular   jul         thu  ...           0  nonexistent
4115       no  telephone   jul         fri  ...           0  nonexistent
4116       no   cellular   may         mon  ...           1      failure
4117       no   cellular   aug         fri  ...           0  nonexistent
4118       no   cellular   nov         wed  ...           0  nonexistent

      emp.var.rate  cons.price.idx cons.conf.idx  euribor3m  nr.employed
y  \
0             -1.8          92.893         -46.2      1.313  5099.100000
no
1              1.1          93.994         -36.4      4.855  5191.000000
no
2              1.4          94.465         -41.8      4.962  5228.100000
no
3              1.4          94.465         -41.8      4.959  5228.100000
no
4             -0.1          93.200         -42.0      4.191  5195.800000
no
5             -1.1          94.199         -37.5      0.884  4963.600000
no
6             -1.1          94.199         -37.5      0.879  4963.600000
no
7             -0.1          93.200         -42.0      4.191  5195.800000
no
8             -0.1          93.200         -42.0      4.153  5195.800000
no
9              1.1          93.994         -36.4      4.855  5191.000000
no
10             1.4          93.918         -42.7      4.958  5228.100000
no
11             1.4          93.918         -42.7      4.968  5228.100000
no
12             1.1          93.994         -36.4      4.859  5191.000000
no
13             1.4          94.465         -41.8      4.958  5228.100000
no
14            -1.8          92.893         -46.2      1.313  5099.100000
no
15             1.4          93.918         -42.7      4.963  5228.100000
no
16             1.4          93.918         -42.7      4.957  5228.100000
no
17             1.4          93.918         -42.7      4.962  5228.100000
no
18             1.4          93.444         -36.1      4.965  5228.100000
no
19             1.4          93.918         -42.7      4.961  5228.100000
yes
20             1.4          93.918         -42.7      4.962  5228.100000
no
22             1.4          93.444         -36.1      4.967  5228.100000
no
23             1.4          93.444         -36.1      4.965  5228.100000
no
24             1.1          93.994         -36.4      4.864  5191.000000
no
25             1.4          93.444         -36.1      4.965  5228.100000
yes
26             1.1          93.994         -36.4      4.856  5191.000000
no
27            -1.8          92.893         -46.2      1.299  5099.100000
```

```
no
28           1.1          93.994          -36.4      4.860  5191.000000
no
29           1.4          93.444          -36.1      4.963  5228.100000
no
30          -1.8          92.843          -50.0      1.687  5166.536386
no
...          ...             ...            ...        ...          ...
...
4087         1.1          93.994          -36.4      4.860  5191.000000
no
4089        -3.4          92.431          -26.9      0.739  5017.500000
yes
4090         1.1          93.994          -36.4      4.857  5191.000000
no
4091         1.1          93.994          -36.4      4.860  5191.000000
no
4092         1.4          93.918          -42.7      4.956  5228.100000
no
4093         1.4          93.918          -42.7      4.961  5228.100000
no
4095         1.4          93.444          -36.1      4.963  5228.100000
no
4096         1.1          93.994          -36.4      4.857  5191.000000
no
4097         1.4          93.444          -36.1      4.963  5228.100000
no
4098         1.4          93.444          -36.1      4.963  5228.100000
no
4099         1.4          93.444          -36.1      4.963  5228.100000
no
4100        -1.8          93.075          -47.1      1.405  5099.100000
no
4101         1.4          93.918          -42.7      4.960  5228.100000
no
4102         1.4          94.465          -41.8      4.959  5228.100000
no
4103         1.1          93.994          -36.4      4.857  5191.000000
no
4104         1.4          93.444          -36.1      4.970  5228.100000
no
4105         1.4          93.918          -42.7      4.961  5228.100000
no
4106        -1.8          92.893          -46.2      1.344  5099.100000
no
4107         1.4          93.444          -36.1      4.964  5228.100000
no
4108        -1.8          92.893          -46.2      1.266  5099.100000
no
4109        -3.4          92.431          -26.9      0.740  5017.500000
no
4110         1.1          93.994          -36.4      4.855  5191.000000
no
4111        -1.7          94.055          -39.8      0.748  4991.600000
no
4112        -0.1          93.200          -42.0      4.076  5195.800000
no
4113        -0.1          93.200          -42.0      4.076  5195.800000
no
4114         1.4          93.918          -42.7      4.958  5228.100000
no
```

```
4115           1.4          93.918          -42.7      4.959  5228.100000
no
4116          -1.8          92.893          -46.2      1.354  5099.100000
no
4117           1.4          93.444          -36.1      4.966  5228.100000
no
4118          -0.1          93.200          -42.0      4.120  5195.800000
no
```

	Age	Pdays
0	30.0	NaN
1	39.0	NaN
2	25.0	NaN
3	38.0	NaN
4	47.0	NaN
5	32.0	NaN
6	32.0	NaN
7	41.0	NaN
8	31.0	NaN
9	35.0	NaN
10	25.0	NaN
11	36.0	NaN
12	36.0	NaN
13	47.0	NaN
14	29.0	NaN
15	27.0	NaN
16	44.0	NaN
17	46.0	NaN
18	45.0	NaN
19	50.0	NaN
20	55.0	NaN
22	29.0	NaN
23	40.0	NaN
24	44.0	NaN
25	38.0	NaN
26	36.0	NaN
27	28.0	NaN
28	47.0	NaN
29	34.0	NaN
30	38.0	NaN
...	...	...
4087	29.0	NaN
4089	25.0	NaN
4090	43.0	NaN
4091	38.0	NaN
4092	30.0	NaN
4093	56.0	NaN
4095	36.0	NaN
4096	33.0	NaN
4097	41.0	NaN
4098	34.0	NaN
4099	58.0	NaN
4100	41.0	NaN
4101	35.0	NaN
4102	31.0	NaN
4103	43.0	NaN
4104	42.0	NaN
4105	47.0	NaN
4106	45.0	NaN
4107	36.0	NaN
4108	32.0	NaN

```
4109  63.0    NaN
4110  53.0    NaN
4111  30.0    NaN
4112  31.0    NaN
4113  31.0    NaN
4114  30.0    NaN
4115  39.0    NaN
4116  27.0    NaN
4117  58.0    NaN
4118  34.0    NaN

[3959 rows x 23 columns]
```

4. previous

In [45]:

```
bank[bank['previous']<0]
```

Out[45]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	pre

0 rows × 23 columns

In [46]:

```
bank[bank['campaign']<0]
```

Out[46]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	pre

0 rows × 23 columns

In [47]:

```
import numpy
stds=3   # 3 standard deviation
std=numpy.std(bank['emp.var.rate'])

z = bank[['emp.var.rate']].transform(
    lambda x: (x - x.mean()).div(x.std()))
outliers = z.abs() > stds
b= bank[outliers.any(axis=1)]
b['emp.var.rate']
```

Out[47]:

```
Series([], Name: emp.var.rate, dtype: float64)
```

In [48]:

```
#import numpy
stds=3   # 3 standard deviation
std=numpy.std(bank['cons.price.idx'])

z = bank[['cons.price.idx']].transform(
    lambda x: (x - x.mean()).div(x.std()))
outliers = z.abs() > stds
b= bank[outliers.any(axis=1)]
b['cons.price.idx']
```

Out[48]:

```
Series([], Name: cons.price.idx, dtype: float64)
```

In [49]:

```
#import numpy
stds=3   # 3 standard deviation
std=numpy.std(bank['cons.conf.idx'])

z = bank[['cons.conf.idx']].transform(
    lambda x: (x - x.mean()).div(x.std()))
outliers = z.abs() > stds
b= bank[outliers.any(axis=1)]
b['cons.conf.idx']
```

Out[49]:

```
Series([], Name: cons.conf.idx, dtype: float64)
```

In [50]:

```
#import numpy
stds=2   # 3 standard deviation
std=numpy.std(bank['euribor3m'])

z = bank[['euribor3m']].transform(
    lambda x: (x - x.mean()).div(x.std()))
outliers = z.abs() > stds
b= bank[outliers.any(axis=1)]
b['euribor3m']
```

Out[50]:

```
Series([], Name: euribor3m, dtype: float64)
```

In [51]:

```
#import numpy
stds=3   # 3 standard deviation
std=numpy.std(bank['nr.employed'])

z = bank[['nr.employed']].transform(
    lambda x: (x - x.mean()).div(x.std()))
outliers = z.abs() > stds
b= bank[outliers.any(axis=1)]
b['nr.employed']
```

Out[51]:

```
Series([], Name: nr.employed, dtype: float64)
```

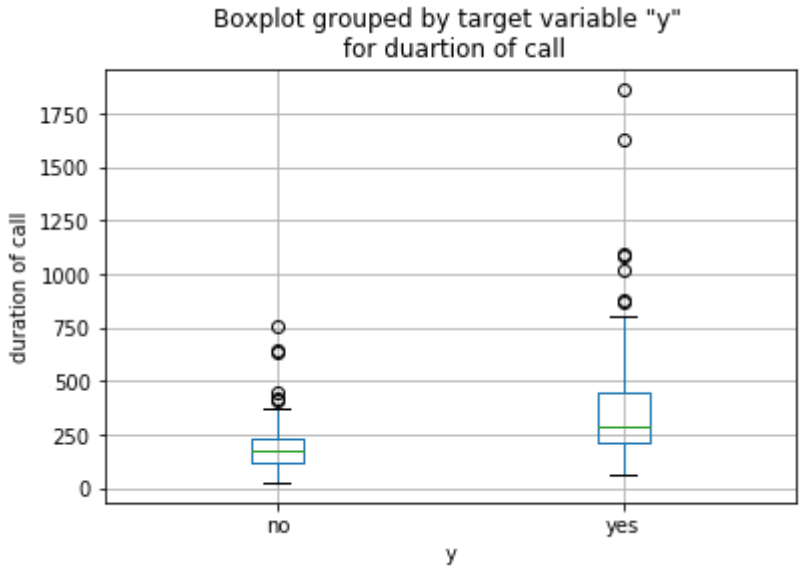
In [52]:

```
print("String type or not?")
for y in bank.columns:
    print "---- %s ---" % y
    if(bank[y].dtype == np.float64 or bank[y].dtype == np.int64):
          print("No")
    else:
          print("yes string type")
```

```
String type or not?
---- age ---
No
---- job ---
yes string type
---- marital ---
yes string type
---- education ---
yes string type
---- default ---
yes string type
---- housing ---
yes string type
---- loan ---
yes string type
---- contact ---
yes string type
---- month ---
yes string type
---- day_of_week ---
yes string type
---- duration ---
No
---- campaign ---
No
---- pdays ---
No
---- previous ---
No
---- poutcome ---
yes string type
---- emp.var.rate ---
No
---- cons.price.idx ---
No
---- cons.conf.idx ---
No
---- euribor3m ---
No
---- nr.employed ---
No
---- y ---
yes string type
---- Age ---
No
---- Pdays ---
No
```

In [53]:

```
bank.dropna().boxplot(column='duration',by='y')
plt.title('Boxplot grouped by target variable "y" \n for duartion of call')
plt.suptitle("") # to get rid of auto generated title for boxplot in python
plt.ylabel('duration of call')
plt.show()
```



In [54]:

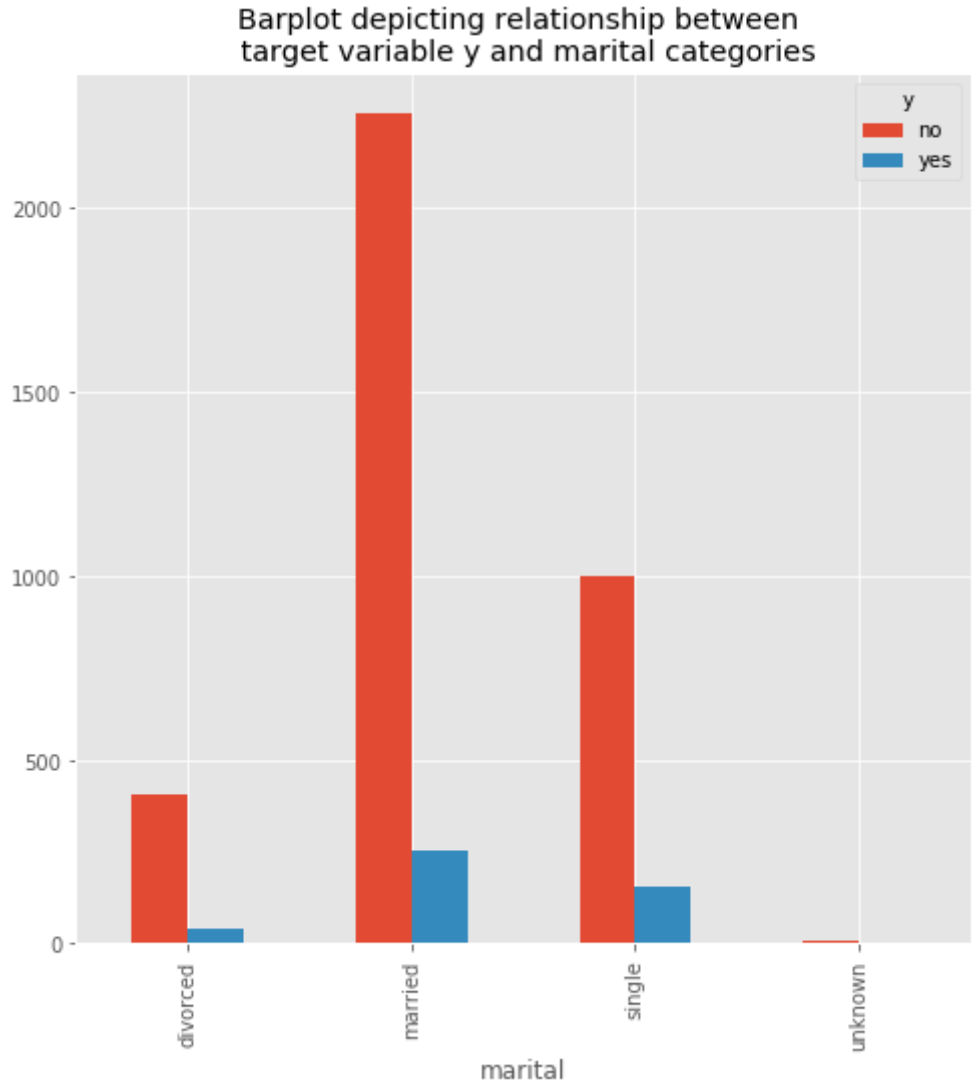
```
import matplotlib
matplotlib.style.use('ggplot')
table = pd.crosstab(index=bank["marital"],
                    columns=bank["y"])

table
```

Out[54]:

y	no	yes
marital		
divorced	403	43
married	2257	252
single	998	155
unknown	10	1

In [55]:

```
table.plot(kind="bar",
                 figsize=(8,8),
                 stacked=False, title="Barplot depicting relationship between \n target variable y and marital categories")
#plt.title('Barplot depicting relationship between \n target variable "y" and marital categories")
#plt.ylabel('duration of call')
plt.show()
```

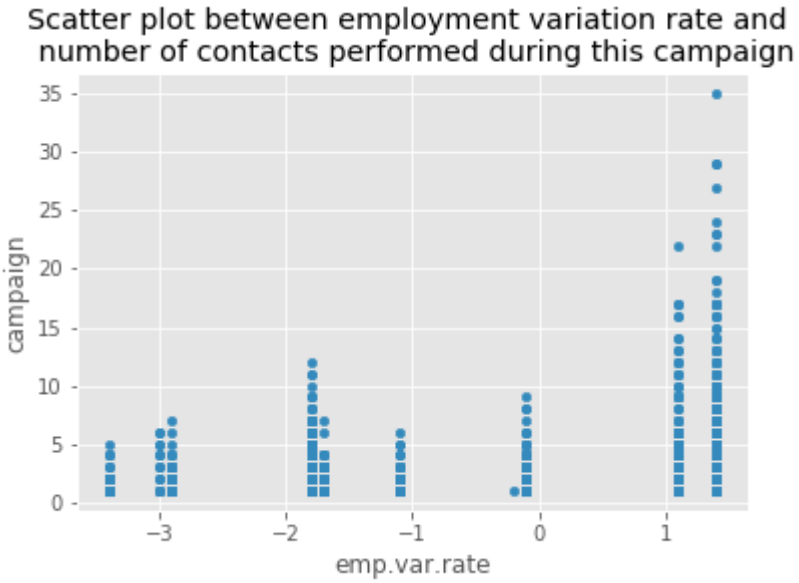


In [56]:

```
bank.plot(kind='scatter', x='emp.var.rate', y='campaign')
plt.title("Scatter plot between employment variation rate and \n number of contacts per
formed during this campaign")
```

Out[56]:

```
Text(0.5,1,u'Scatter plot between employment variation rate and \n number
of contacts performed during this campaign')
```



Scatter Matrix for numerical variables

In [57]:

```
from pandas.plotting import scatter_matrix
scatter_matrix(bank,alpha=0.2,figsize=(21,21),diagonal='hist')
plt.show()
```