BankProject_08_01_2021_CR

August 1, 2021

[1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
import random
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from prettytable import PrettyTable
```

[2]:
```
bank = pd.read_csv(r"C:\Users\Chris\Desktop\504Project\bank\bank.csv",
sep=';')
```

[3]:
```
bank.shape
```

[3]:
```
(4521, 17)
```

[4]:
```
bank.head()
```

[4]:

	age	job	marital	education	default	balance	housing	loan	\
0	30	unemployed	married	primary	no	1787	no	no	
1	33	services	married	secondary	no	4789	yes	yes	
2	35	management	single	tertiary	no	1350	yes	no	
3	30	management	married	tertiary	no	1476	yes	yes	
4	59	blue-collar	married	secondary	no	0	yes	no	

	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	cellular	19	oct	79	1	-1	0	unknown	no
1	cellular	11	may	220	1	339	4	failure	no
2	cellular	16	apr	185	1	330	1	failure	no
3	unknown	3	jun	199	4	-1	0	unknown	no
4	unknown	5	may	226	1	-1	0	unknown	no

[5]:
```
bank.isna().sum()
```

[5]:
```
age          0
job          0
marital      0
education    0
default      0
```

```
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

[6]:
```
bank.dtypes
#There are numerical and categorical data.
```

[6]:
```
age           int64
job          object
marital      object
education    object
default      object
balance       int64
housing      object
loan         object
contact      object
day           int64
month        object
duration      int64
campaign      int64
pdays         int64
previous      int64
poutcome     object
y            object
dtype: object
```

[7]:
```
bank.columns
```

[7]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'y'],
      dtype='object')
```

[8]:
```
myTable = PrettyTable(["Variable", "Min", "Max", "Median", "mean"])

myTable.add_row(["age", bank["age"].min(), bank["age"].max(), bank["age"].
 ↪median(), bank["age"].mean()])
```

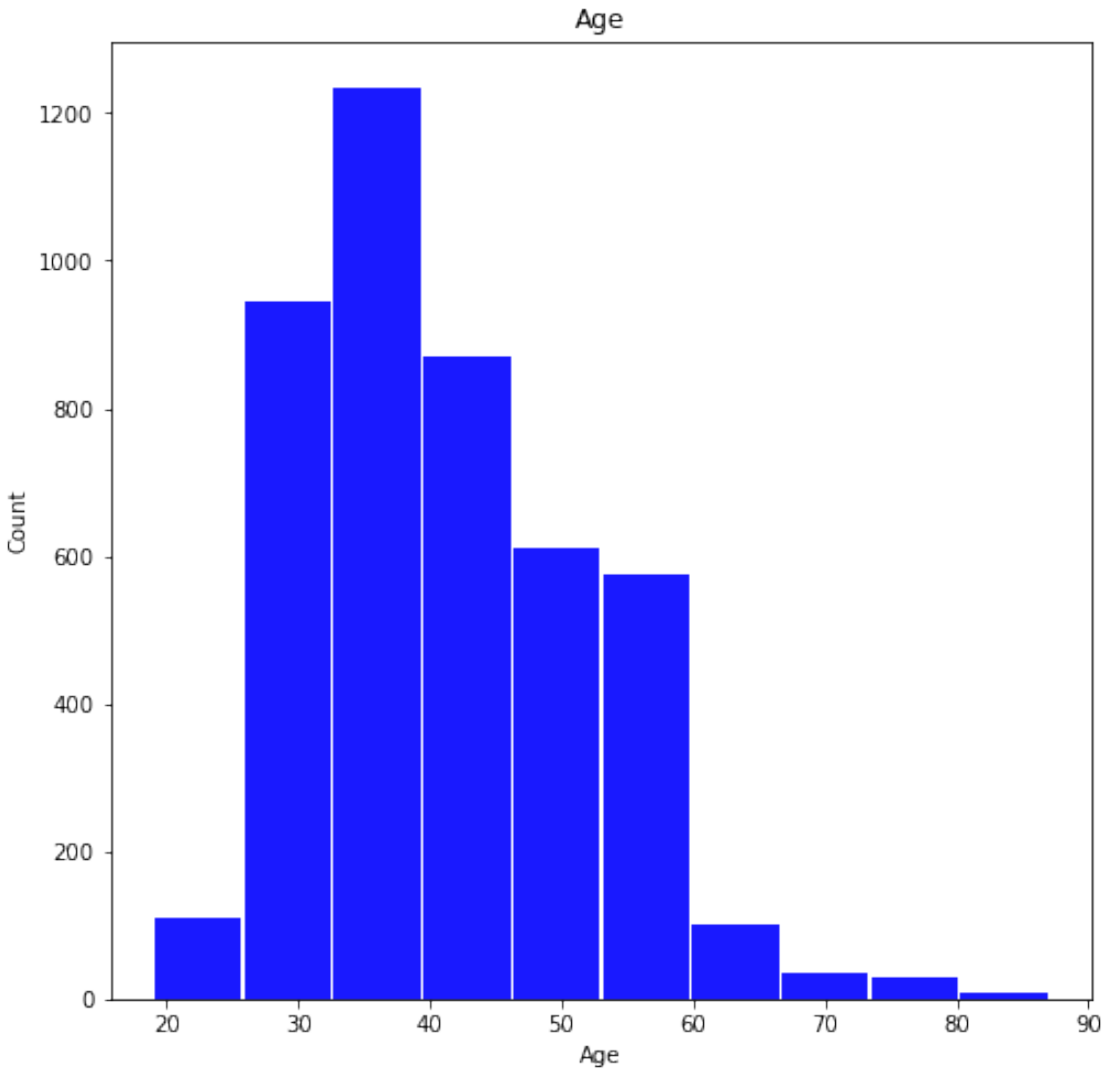
```
myTable.add_row(["balance", bank["balance"].min(), bank["balance"].max(),␣
 ↪bank["balance"].median(), bank["balance"].mean()])
myTable.add_row(["duration", bank["duration"].min(), bank["duration"].max(),␣
 ↪bank["duration"].median(), bank["duration"].mean()])
myTable.add_row(["campaign", bank["campaign"].min(), bank["campaign"].max(),␣
 ↪bank["campaign"].median(), bank["campaign"].mean()])
myTable.add_row(["pdays", bank["pdays"].min(), bank["pdays"].max(),␣
 ↪bank["pdays"].median(), bank["pdays"].mean()])
myTable.add_row(["previous", bank["previous"].min(), bank["previous"].max(),␣
 ↪bank["previous"].median(), bank["previous"].mean()])

print(myTable)
```

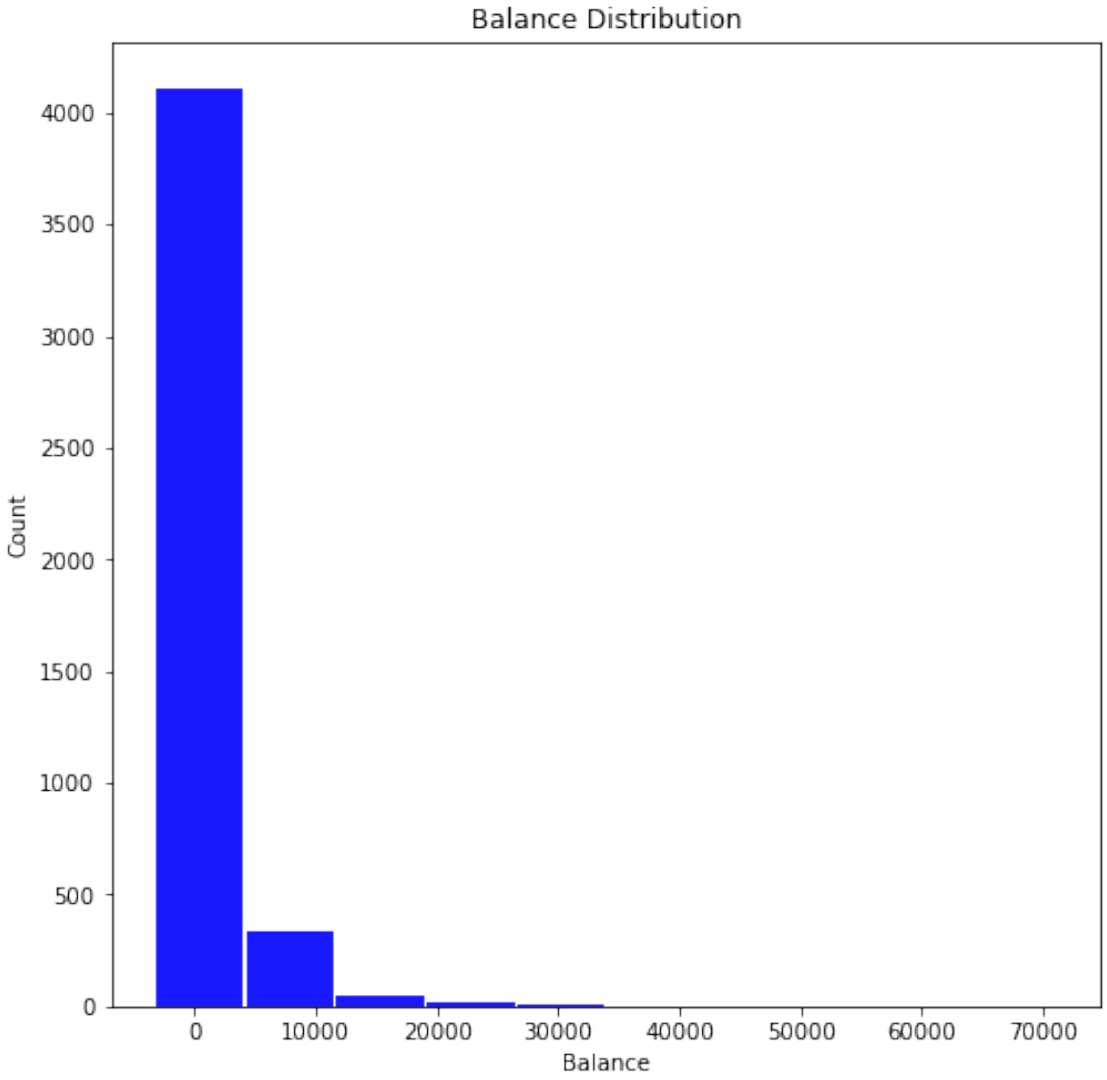
```
+----------+-------+-------+--------+--------------------+
| Variable |  Min  |  Max  | Median |        mean        |
+----------+-------+-------+--------+--------------------+
|   age    |   19  |   87  |  39.0  | 41.17009511170095  |
| balance  | -3313 | 71188 | 444.0  | 1422.6578190665782 |
| duration |   4   |  3025 | 185.0  | 263.96129174961294 |
| campaign |   1   |   50  |  2.0   | 2.793629727936297  |
|  pdays   |   -1  |  871  |  -1.0  | 39.766644547666445 |
| previous |   0   |   25  |  0.0   | 0.5425790754257908 |
+----------+-------+-------+--------+--------------------+
```

What simple models have you tried

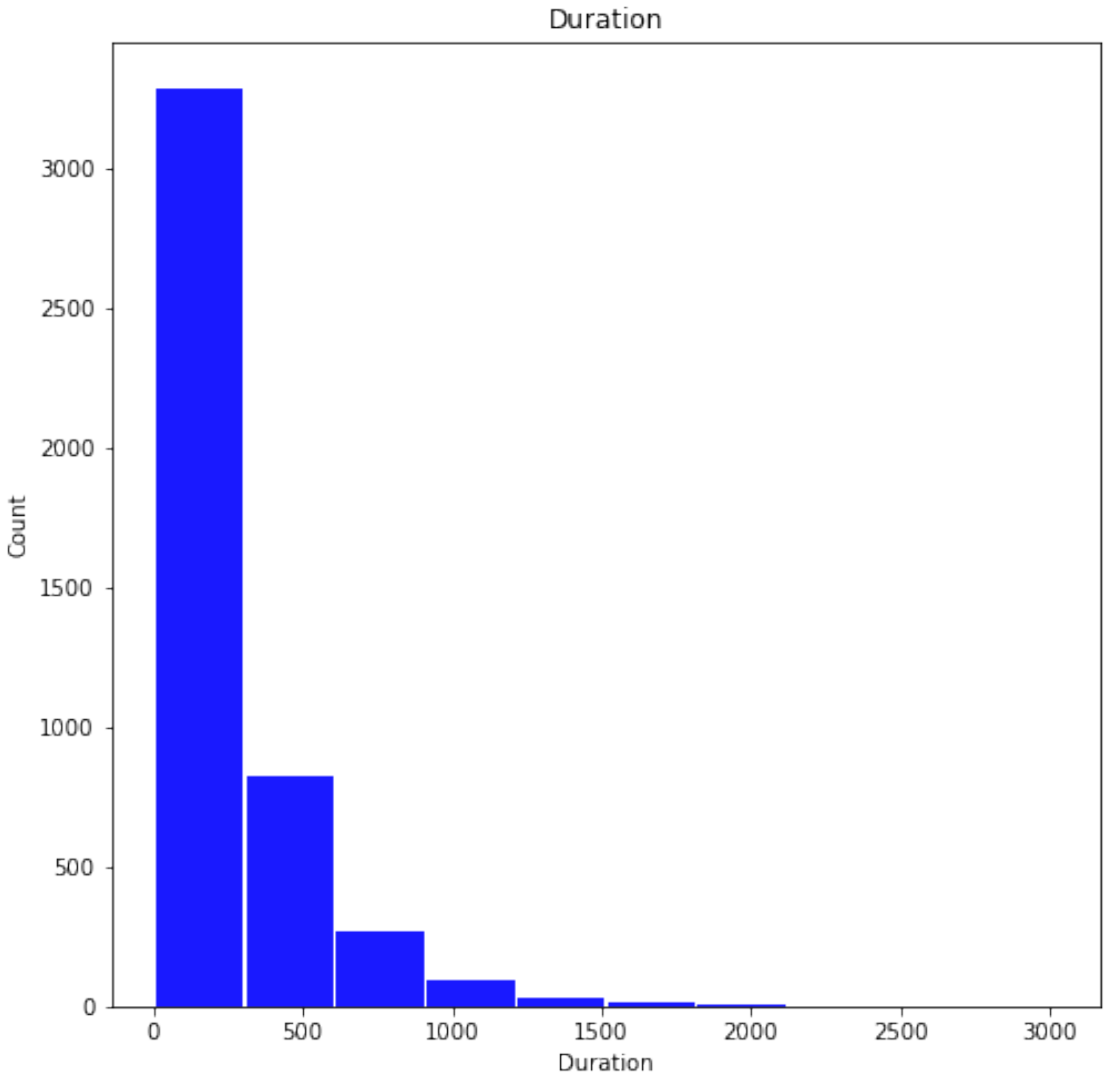
[9]:
```
fig = plt.figure(figsize = (8,8))
plt.hist(bank['age'], color='blue', alpha=0.9, rwidth=.97)
plt.title ('Age')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



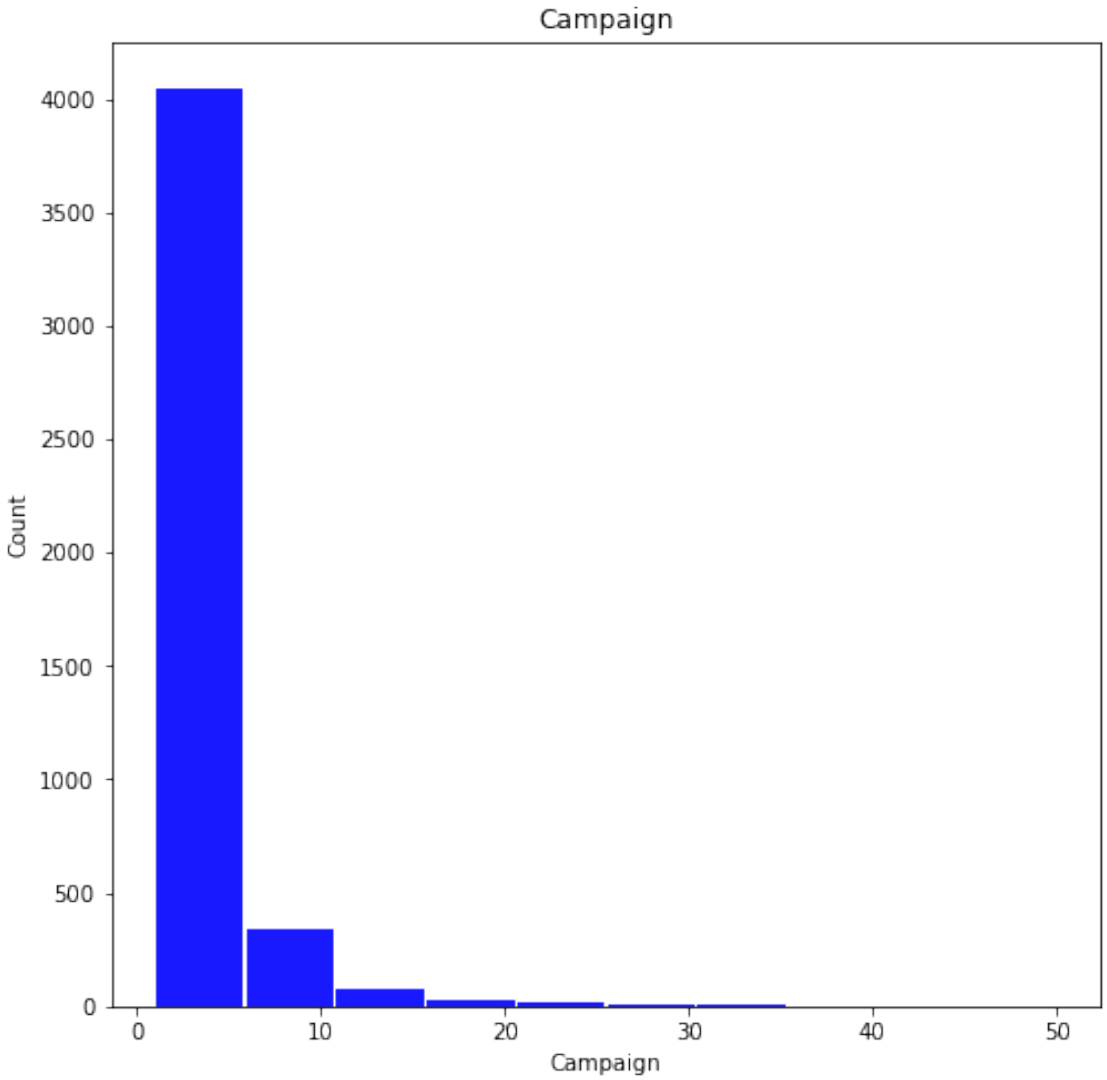
[10]:
```
fig = plt.figure(figsize = (8,8))
plt.hist(bank['balance'], color='blue', alpha=0.9, rwidth=.97)
plt.title ('Balance Distribution')
plt.xlabel('Balance')
plt.ylabel('Count')
plt.show()
```



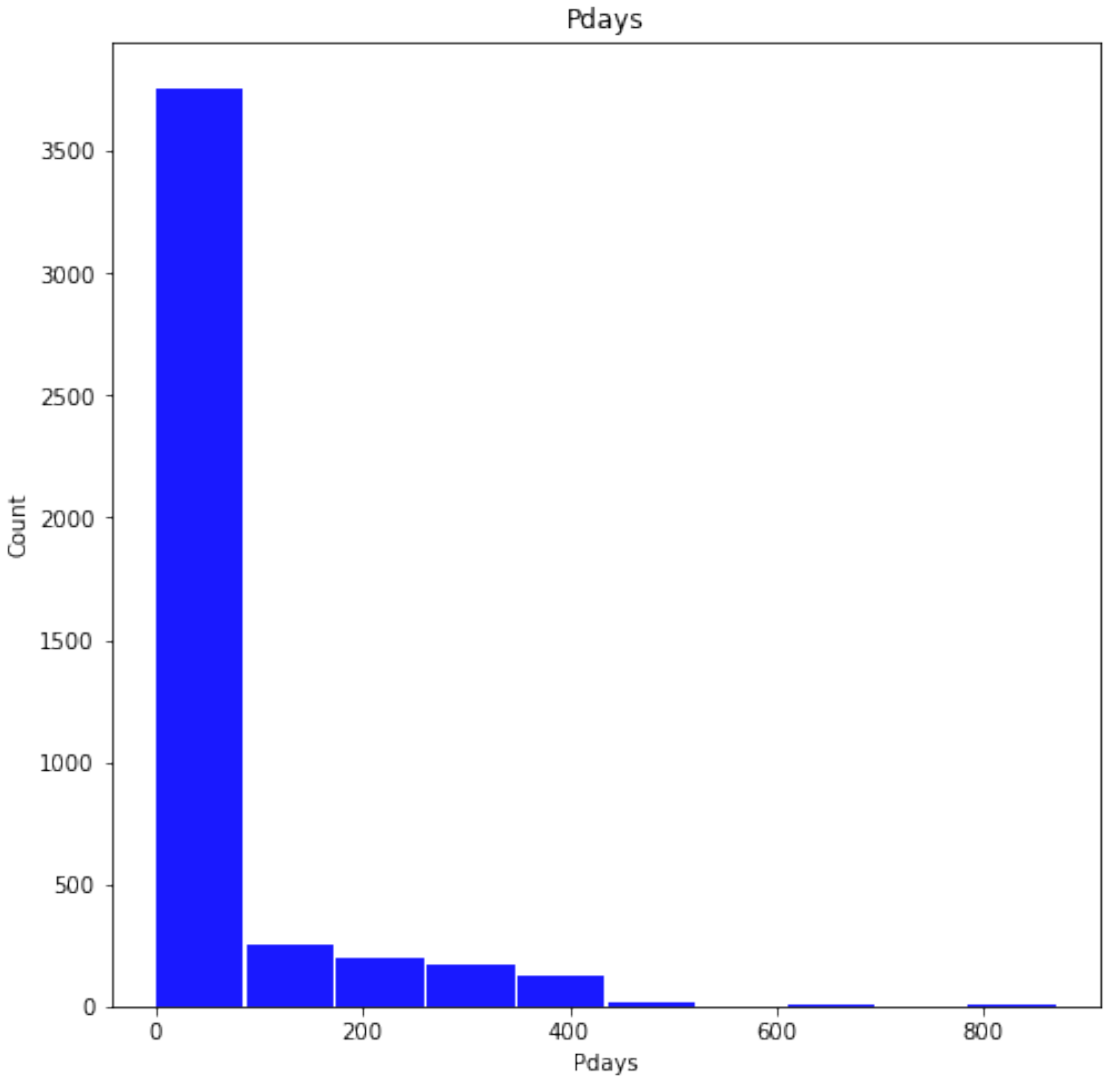
[11]:
```
fig = plt.figure(figsize = (8,8))
plt.hist(bank['duration'], color='blue', alpha=0.9, rwidth=.97)
plt.title ('Duration')
plt.xlabel('Duration')
plt.ylabel('Count')
plt.show()
```



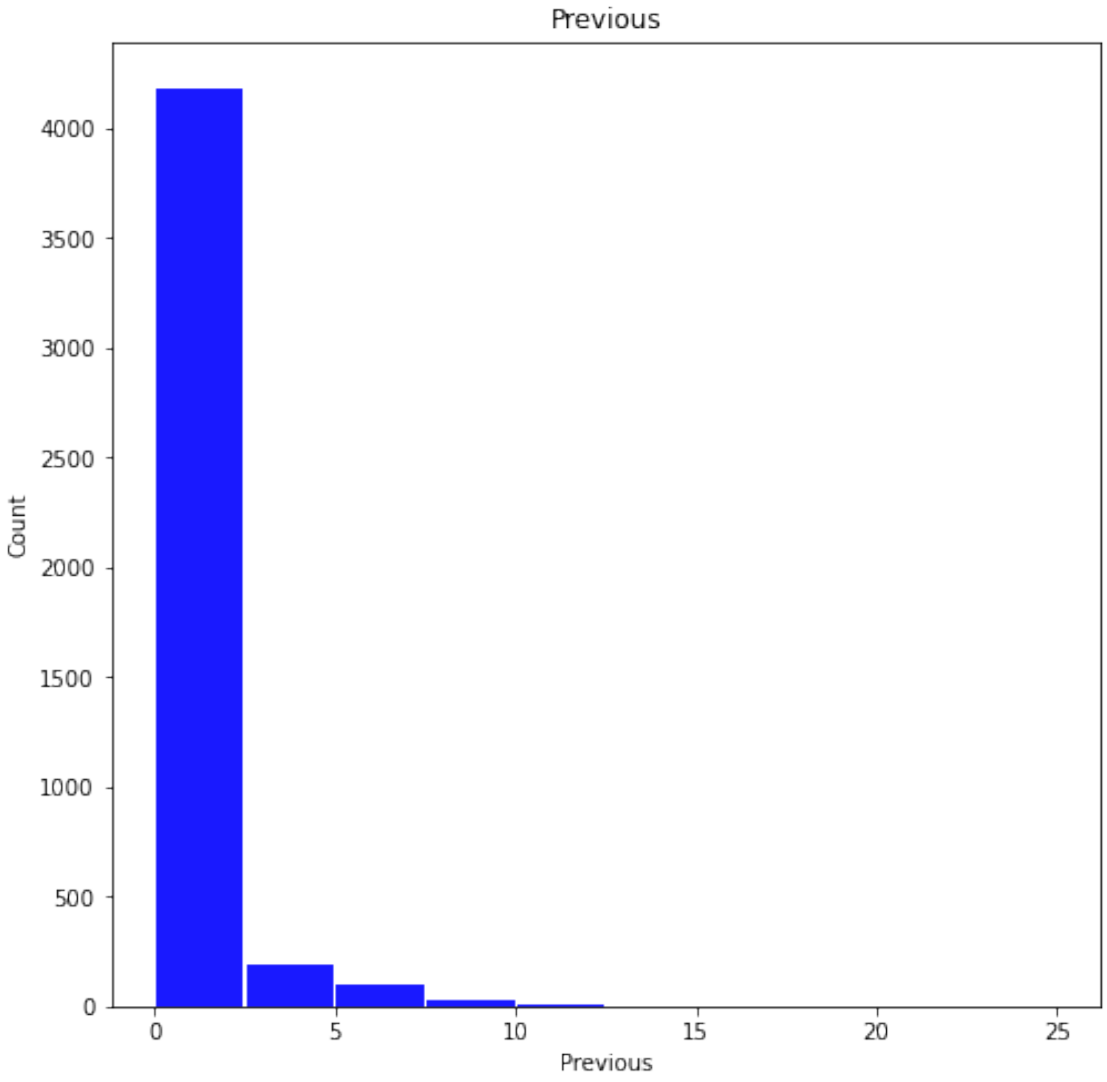
[12]:
```
fig = plt.figure(figsize = (8,8))
plt.hist(bank['campaign'], color='blue', alpha=0.9, rwidth=.97)
plt.title ('Campaign')
plt.xlabel('Campaign')
plt.ylabel('Count')
plt.show()
```



[13]:
```
fig = plt.figure(figsize = (8,8))
plt.hist(bank['pdays'], color='blue', alpha=0.9, rwidth=.97)
plt.title ('Pdays')
plt.xlabel('Pdays')
plt.ylabel('Count')
plt.show()
```



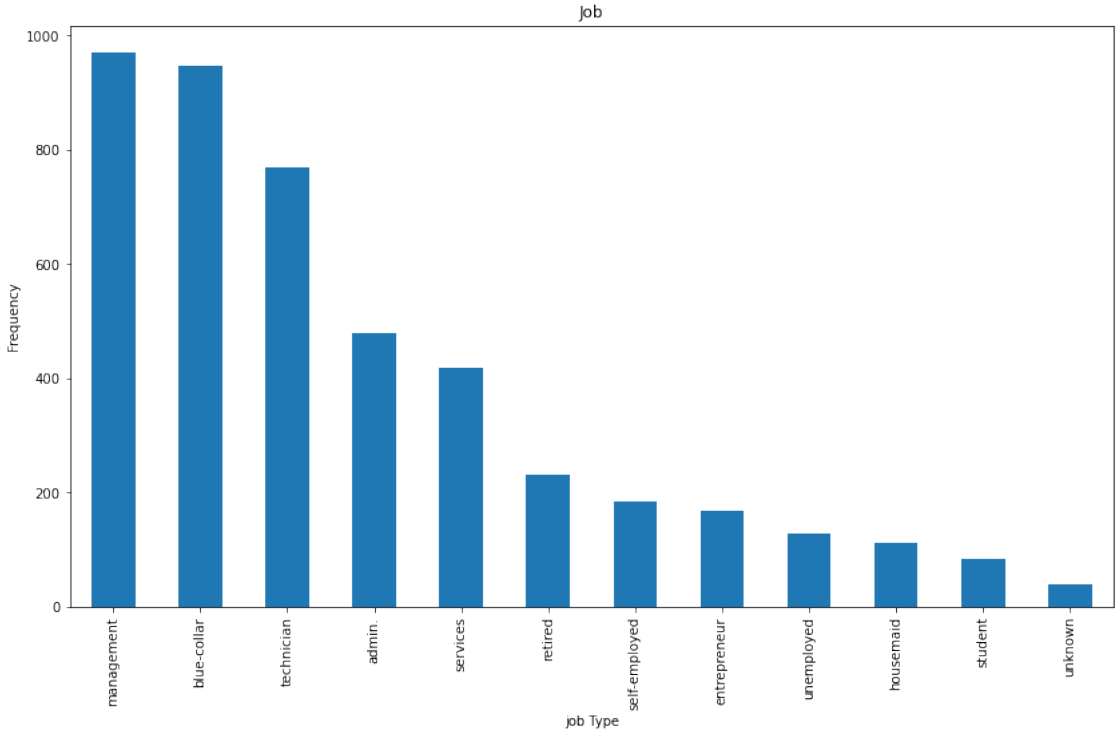
[14]:
```
fig = plt.figure(figsize = (8,8))
plt.hist(bank['previous'], color='blue', alpha=0.9, rwidth=.97)
plt.title ('Previous')
plt.xlabel('Previous')
plt.ylabel('Count')
plt.show()
```



[ ]:

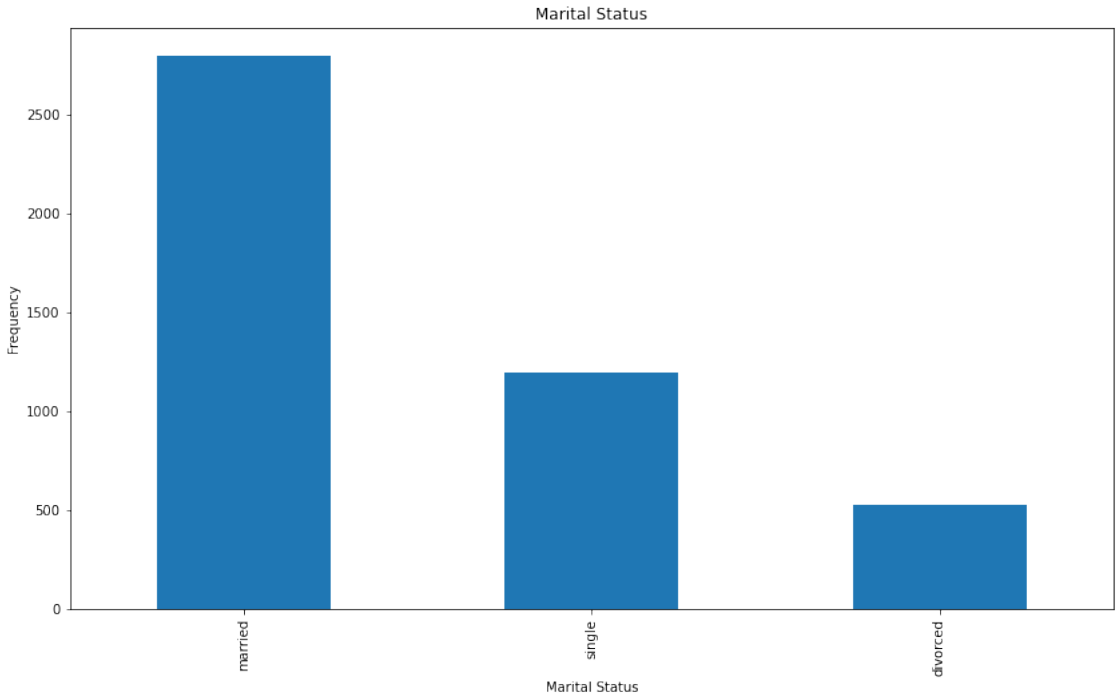
[15]:
```
ax = bank['job'].value_counts().plot(kind='bar',
                                     figsize=(14,8),
                                     title="Job")
ax.set_xlabel("job Type")
ax.set_ylabel("Frequency")
```

[15]: Text(0, 0.5, 'Frequency')



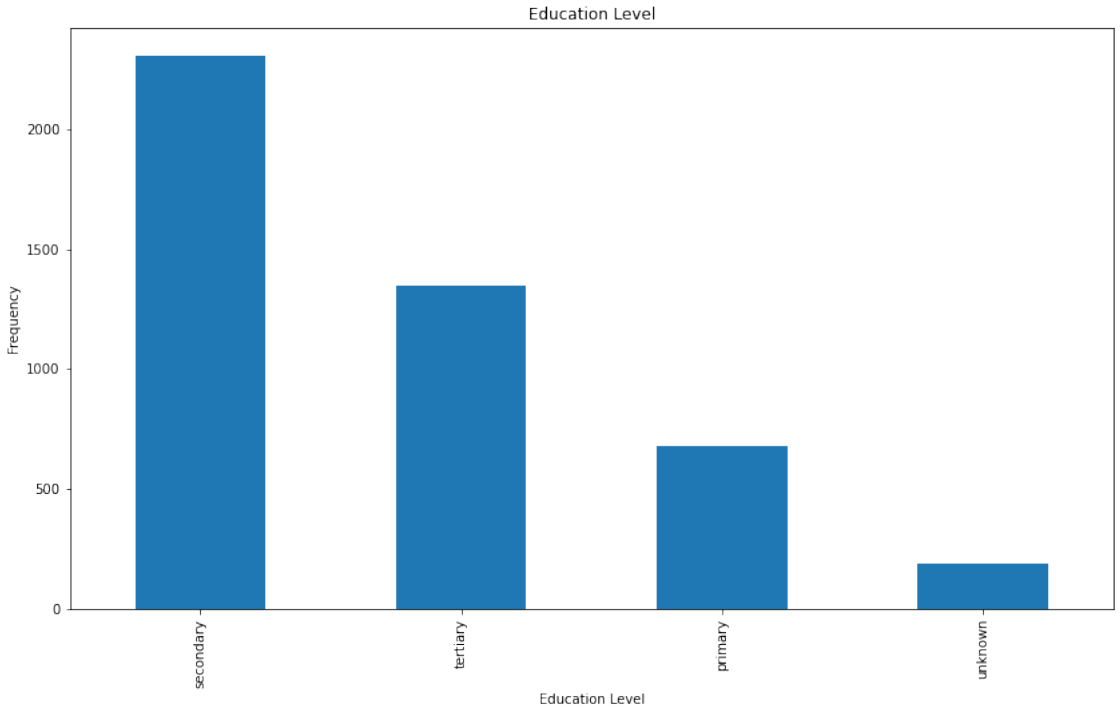
[16]:
```
ax = bank['marital'].value_counts().plot(kind='bar',
                                         figsize=(14,8),
                                         title="Marital Status")
ax.set_xlabel("Marital Status")
ax.set_ylabel("Frequency")
```

[16]: Text(0, 0.5, 'Frequency')



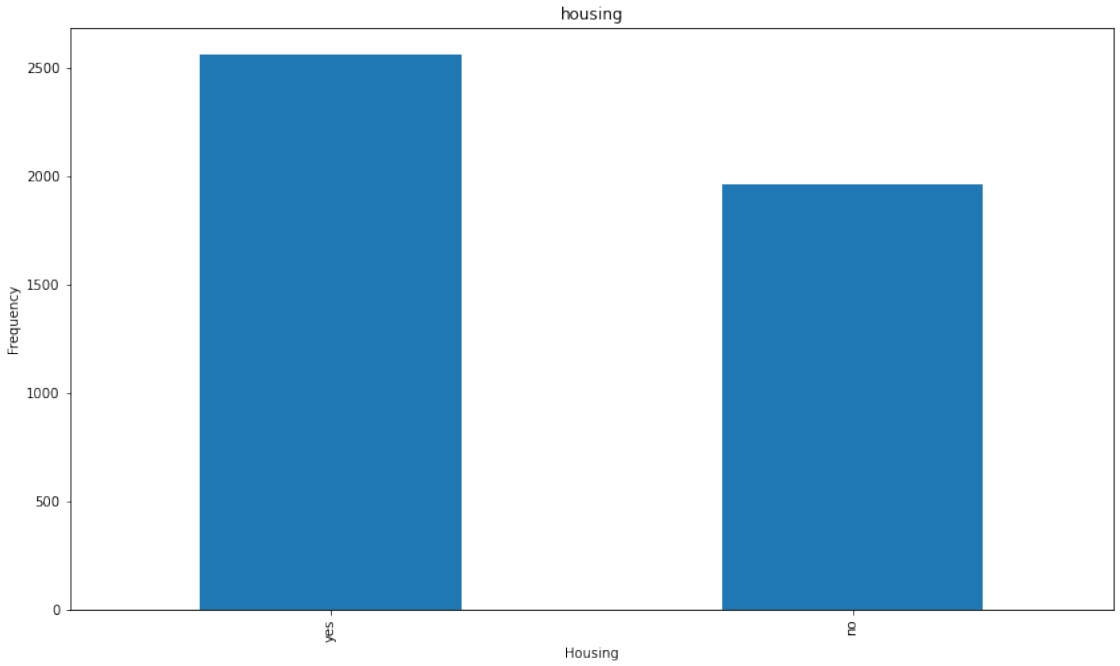
[17]:
```
ax = bank['education'].value_counts().plot(kind='bar',
                                           figsize=(14,8),
                                           title="Education Level")
ax.set_xlabel("Education Level")
ax.set_ylabel("Frequency")
```

[17]: Text(0, 0.5, 'Frequency')



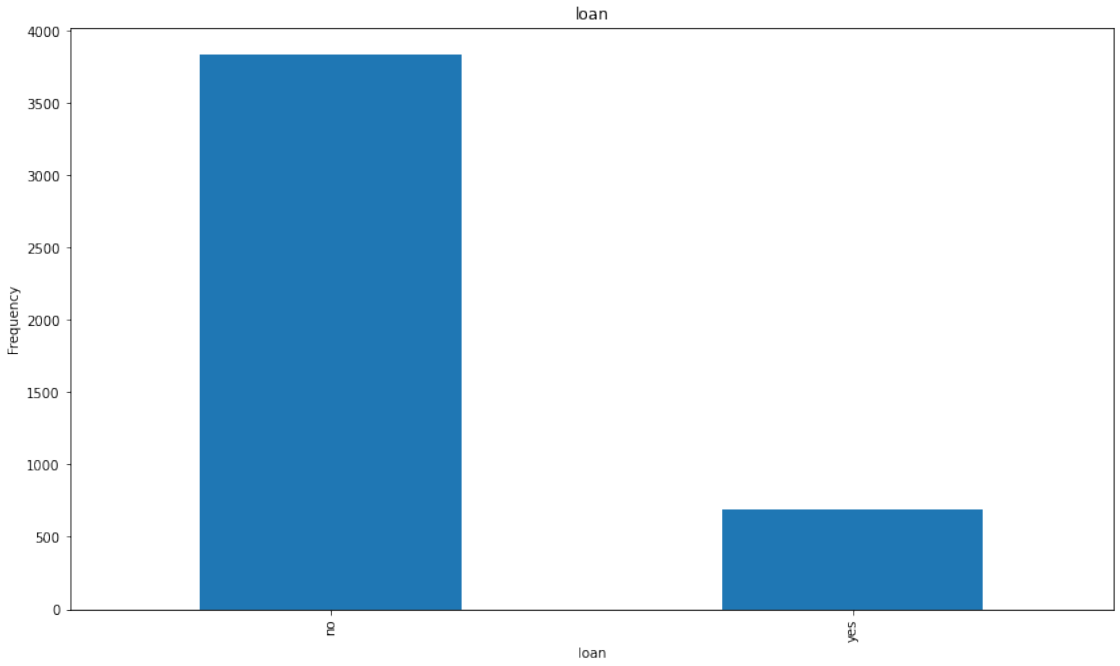
[18]:
```
ax = bank['housing'].value_counts().plot(kind='bar',
                                         figsize=(14,8),
                                         title="housing")
ax.set_xlabel("Housing")
ax.set_ylabel("Frequency")
```

[18]: Text(0, 0.5, 'Frequency')



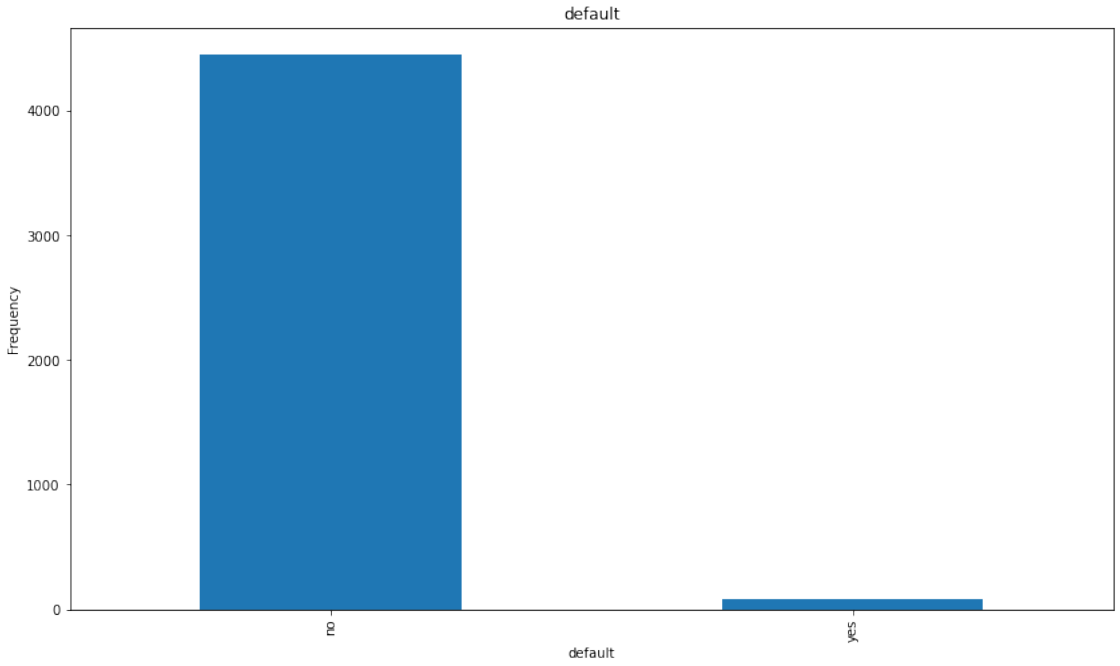
[19]:
```
ax = bank['loan'].value_counts().plot(kind='bar',
                                      figsize=(14,8),
                                      title="loan")
ax.set_xlabel("loan")
ax.set_ylabel("Frequency")
```

[19]: Text(0, 0.5, 'Frequency')



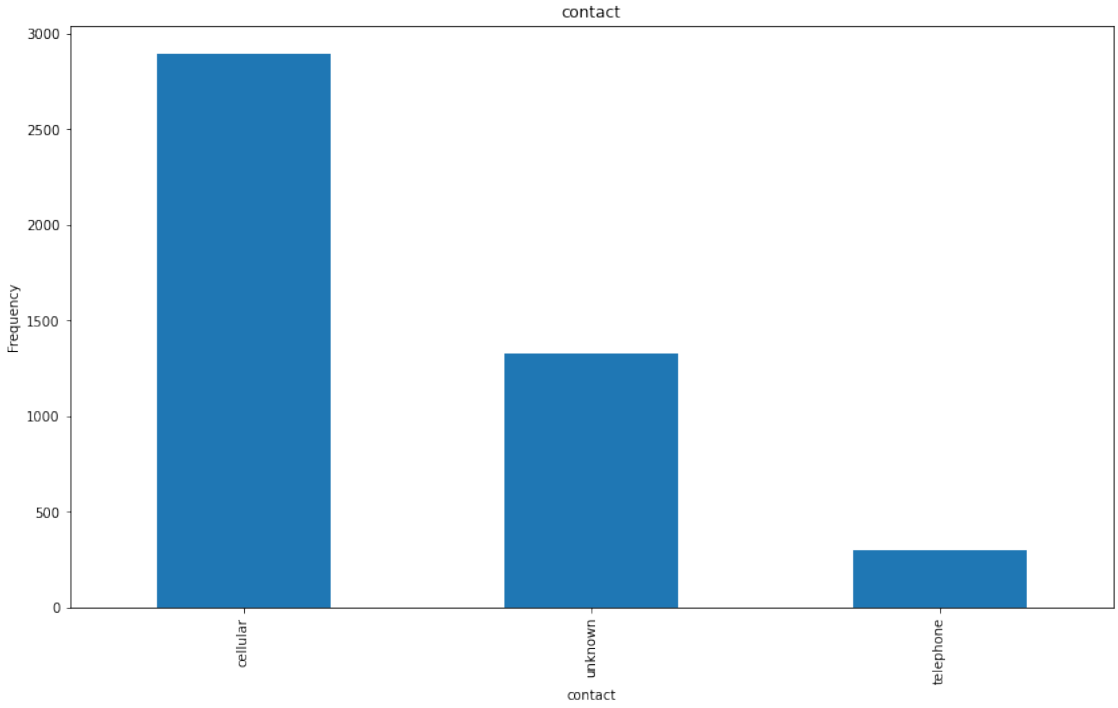
[20]:
```
ax = bank['default'].value_counts().plot(kind='bar',
                                         figsize=(14,8),
                                         title="default")
ax.set_xlabel("default")
ax.set_ylabel("Frequency")
```

[20]: Text(0, 0.5, 'Frequency')



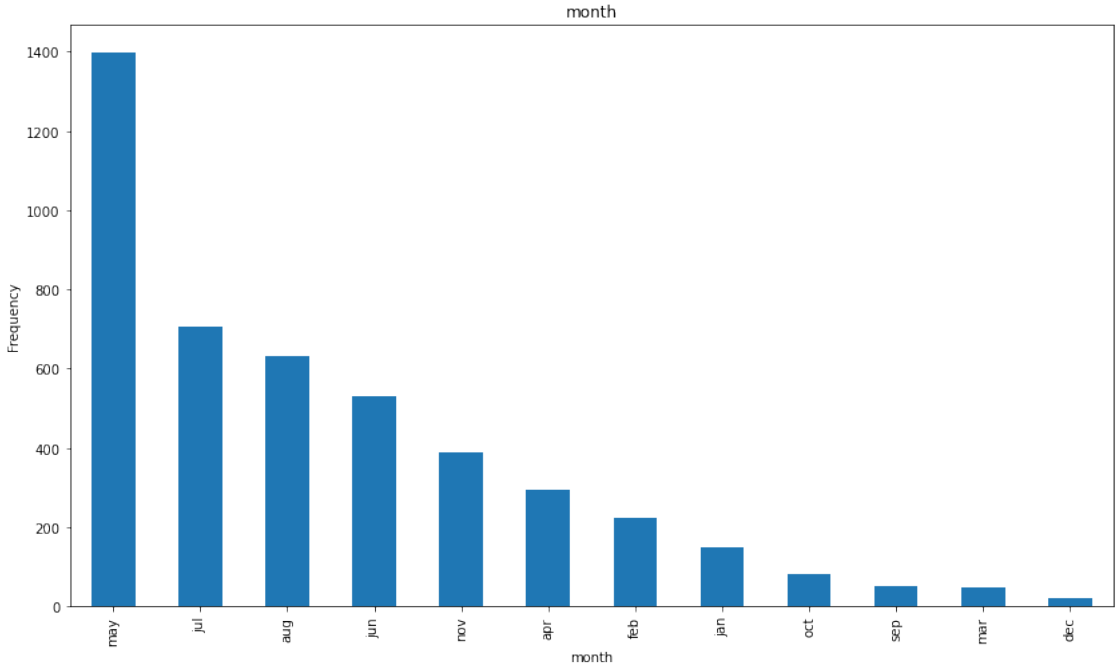
[21]:
```
ax = bank['contact'].value_counts().plot(kind='bar',
                                         figsize=(14,8),
                                         title="contact")
ax.set_xlabel("contact")
ax.set_ylabel("Frequency")
```

[21]:
```
Text(0, 0.5, 'Frequency')
```



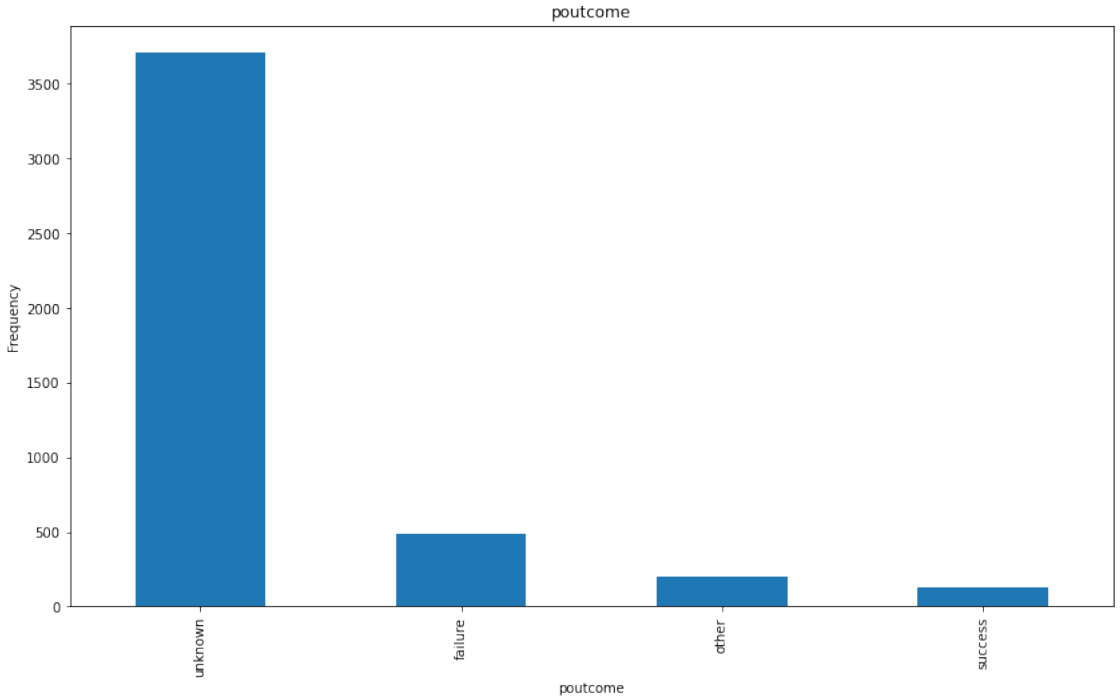
[22]:
```
ax = bank['month'].value_counts().plot(kind='bar',
                                       figsize=(14,8),
                                       title="month")
ax.set_xlabel("month")
ax.set_ylabel("Frequency")
```

[22]: Text(0, 0.5, 'Frequency')



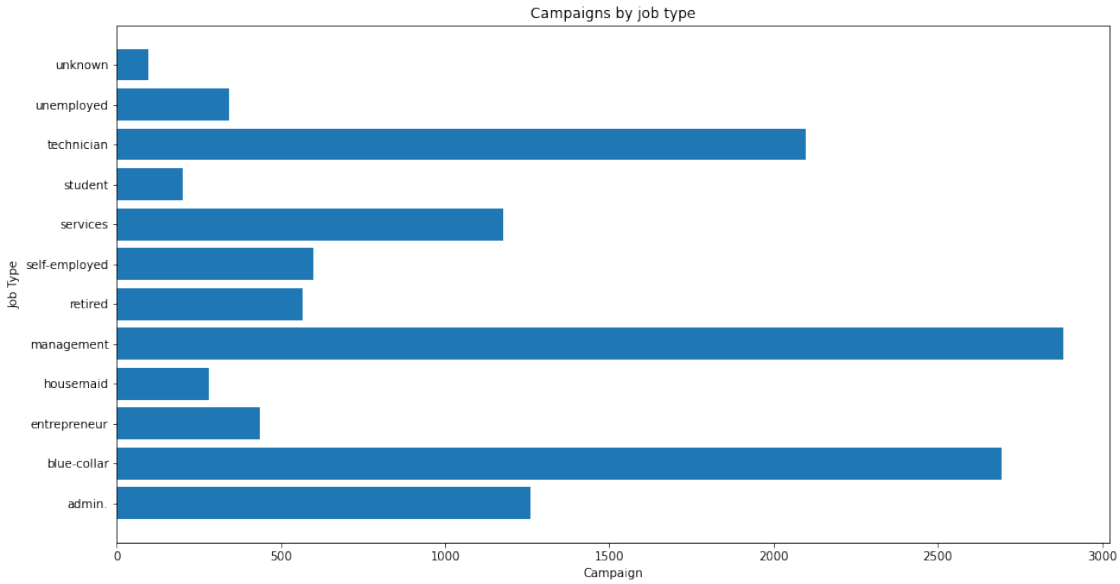
[23]:
```
ax = bank['poutcome'].value_counts().plot(kind='bar',
                                          figsize=(14,8),
                                          title="poutcome")
ax.set_xlabel("poutcome")
ax.set_ylabel("Frequency")
```

[23]: Text(0, 0.5, 'Frequency')



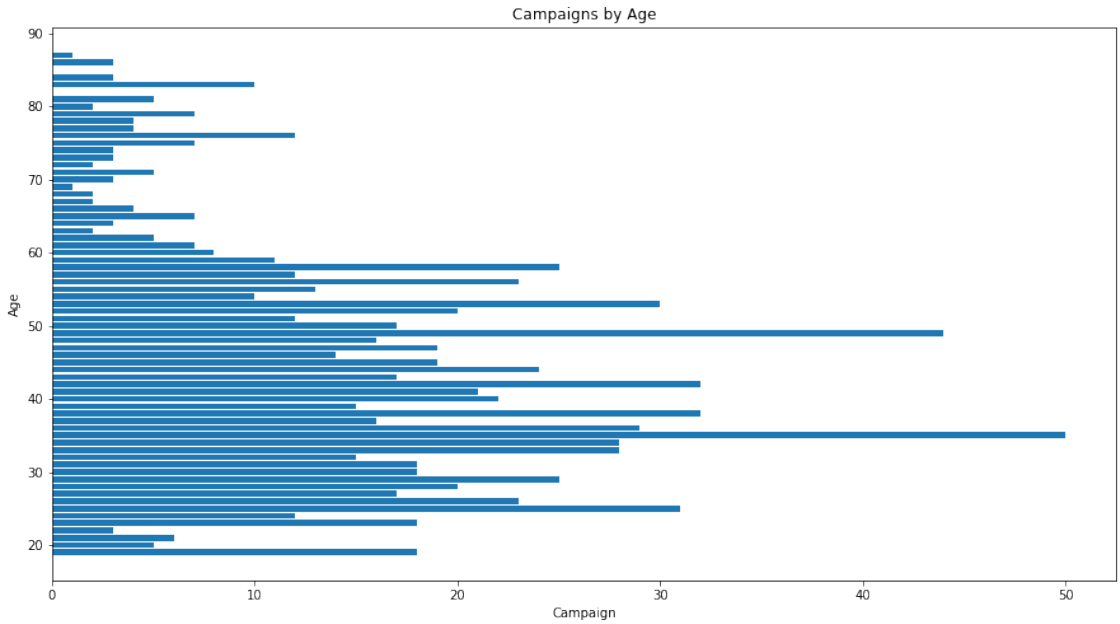
[24]:
```
campaignByjob = bank.groupby('job', as_index=False)['campaign'].sum()

fig = plt.figure(figsize = (15,8))
plt.barh(campaignByjob['job'], campaignByjob['campaign'])
plt.xlabel("Campaign")
plt.ylabel("Job Type")
plt.title("Campaigns by job type")
plt.show()
```

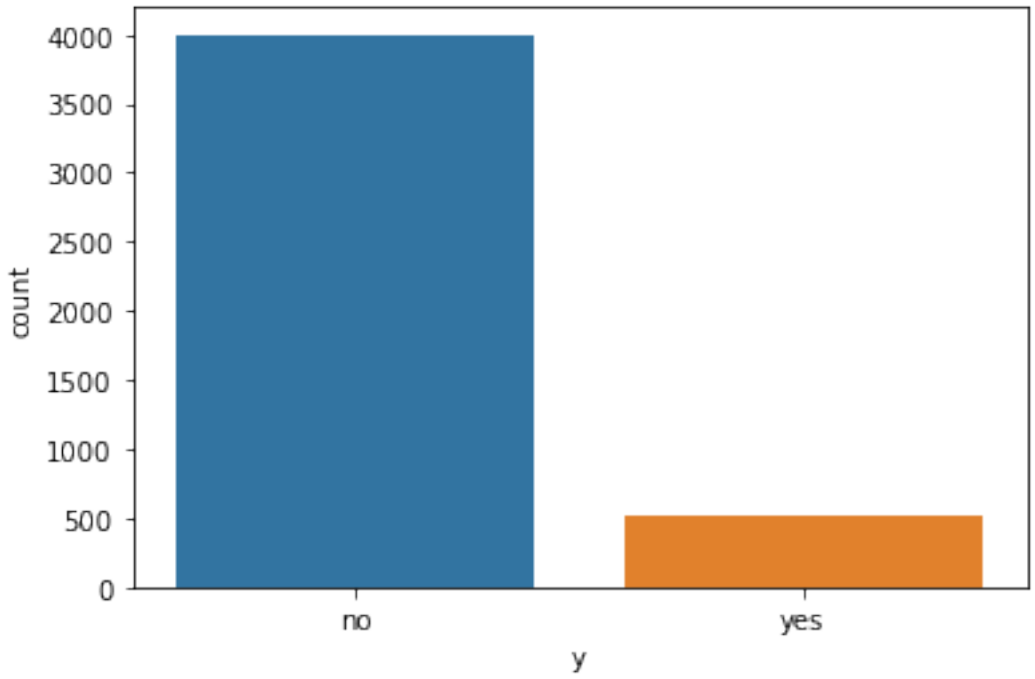


[25]:
```
#campaignByAge = bank.groupby('age', as_index=False)['campaign'].sum()

fig = plt.figure(figsize = (15,8))
plt.barh(bank['age'], bank['campaign'])
plt.xlabel("Campaign")
plt.ylabel("Age")
plt.title("Campaigns by Age")
plt.show()
```



[26]:
```
ax = sns.countplot(x = bank["y"])  #Imbalanced dataset
plt.show()
```

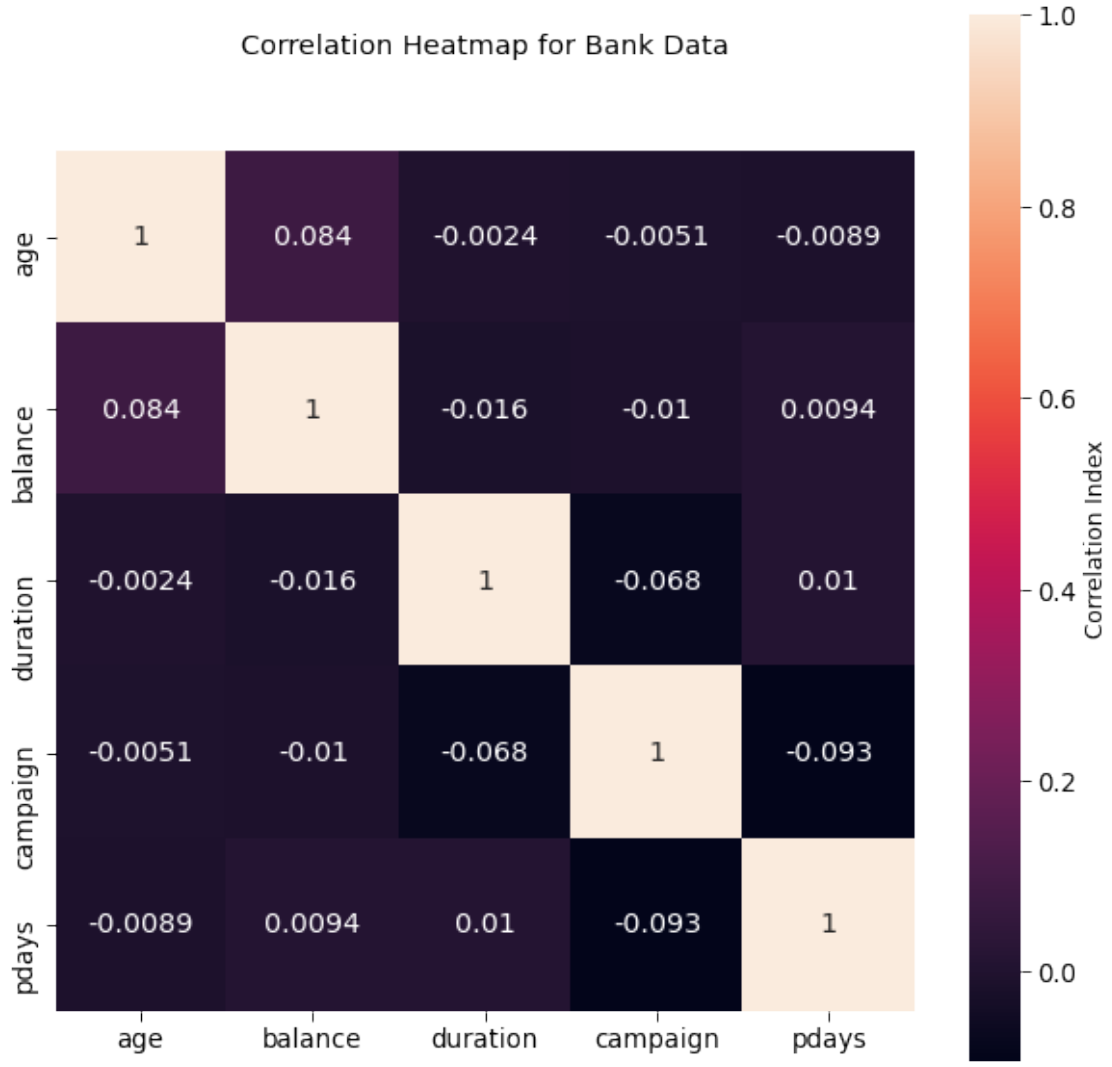


categorical variables = [“job”, “marital”, “education”, “default”, “housing”, “loan”, “contact”, “day”, “month”, “poutcome”]

numerical variables = [“age”, “balance”, “duration”, “campaign”, “pdays”, “previous”]

[27]:
```
heatmap = bank[['age','balance','duration','campaign','pdays']]
sns.set_context("paper", rc={"axes.labelsize":12}, font_scale = 1.5)
correlations = heatmap.corr()
plt.figure(figsize = (10,10))

ax = sns.heatmap(correlations[['age','balance','duration','campaign','pdays']],
annot = True, square = True, cbar_kws={'label': 'Correlation Index'})
ax.set_title('Correlation Heatmap for Bank Data')
ax.set_ylim(len(correlations), -0.5)
plt.show()
```


Correlation Heatmap for Bank Data
age
balance
duration
campaign
pdays
1
0.084
-0.0024
-0.0051
-0.0089
0.084
1
-0.016
-0.01
0.0094
-0.0024
-0.016
1
-0.068
0.01
-0.0051
-0.01
-0.068
1
-0.093
-0.0089
0.0094
0.01
-0.093
1
age
balance
duration
campaign
pdays
Correlation Index
1.0
0.8
0.6
0.4
0.2
0.0

[ ]: