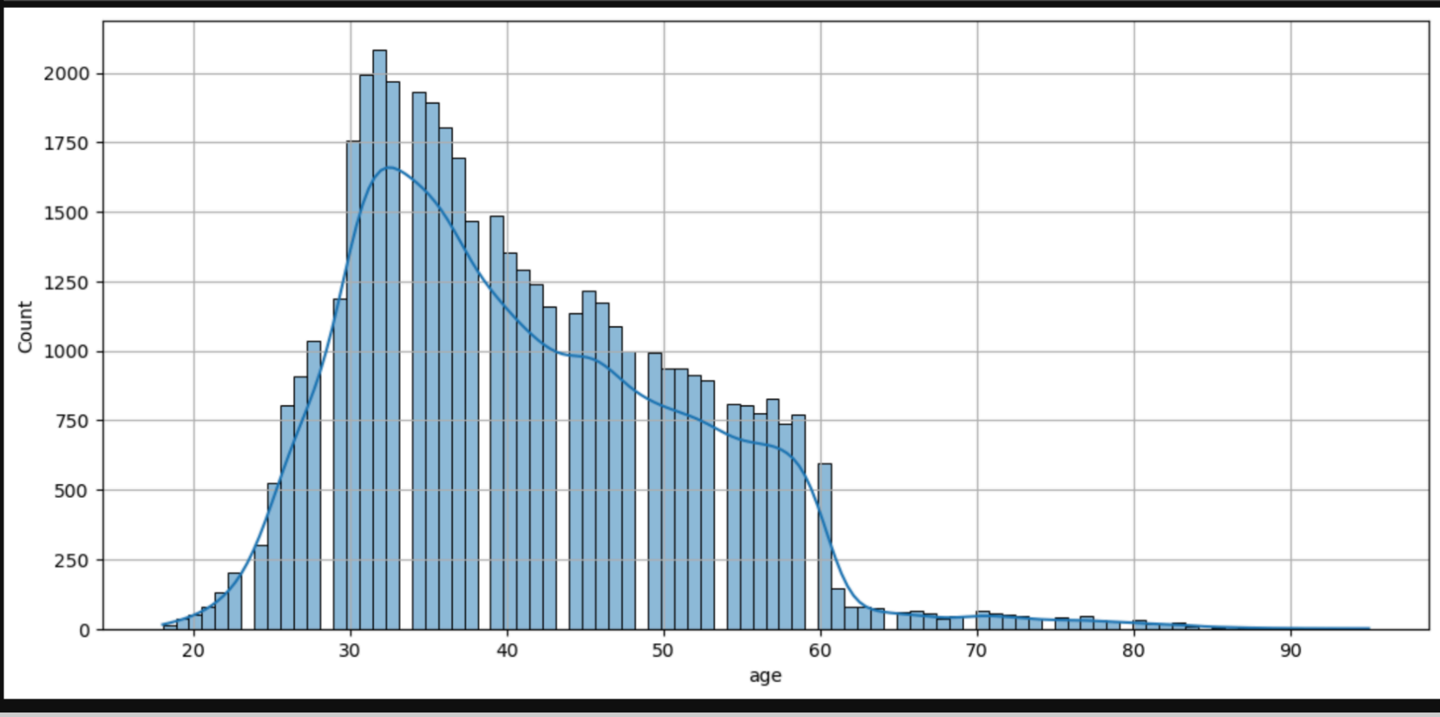
# Questions

1. What is the distribution of age among the clients ?
2. How does the job type vary among the clients?
3. What is the marital status distribution of the clients?
4. What is the level of education among the clients?
5. What proportion of clients have credit in default?
6. What is the distribution of average yearly balance among the clients?
7. How many clients have housing loans?
8. How many clients have personal loans?
9. What are the communication types used for contacting clients during the campaign?
10. What is the distribution of the last contact day of the month?
11. How does the last contact month vary among the clients?
12. What is the distribution of the duration of the last contact?
13. How many contacts were performed during the campaign for each client?
14. What is the distribution of the number of days passed since the client was last contacted from a previous campaign?
15. How many contacts were performed before the current campaign for each client?
16. What were the outcomes of the previous marketing campaigns?
17. What is the distribution of clients who subscribed to a term deposit vs. those who did not?
18. Are there any correlations between different attributes and the likelihood of subscribing to a term deposit?

# Answers

1. What is the distribution of age among the clients ?

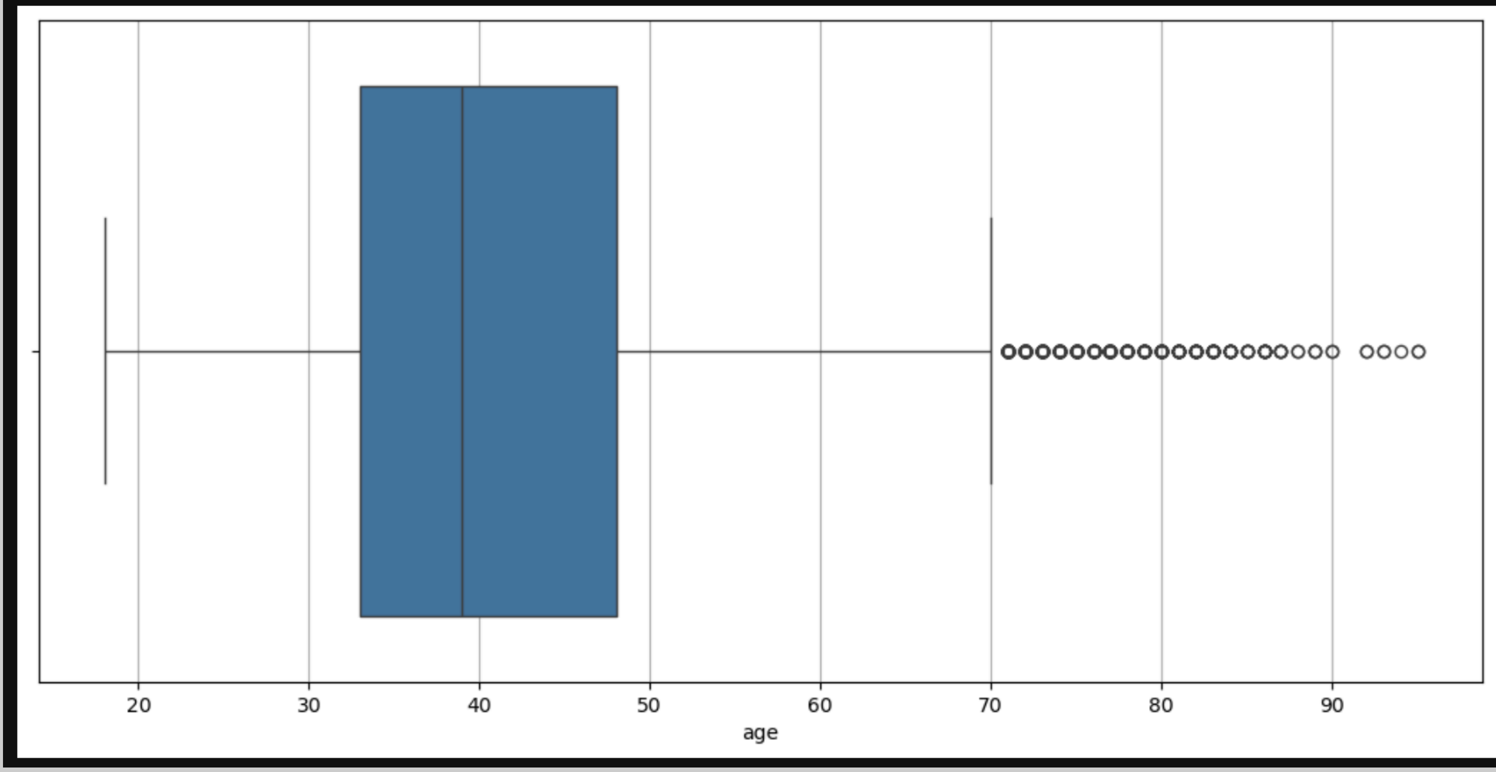
```
35]: plt.figure(figsize=[13,6])
     plt.grid()
     sns.histplot(data=dataframe, x='age', kde=True);
```




**The distribution of age :** The clients called by the bank have ages ranging from 18 to 95 years old .

But the majority of clients were between 33 to 48 ( based on the 25th and 75th percentiles )

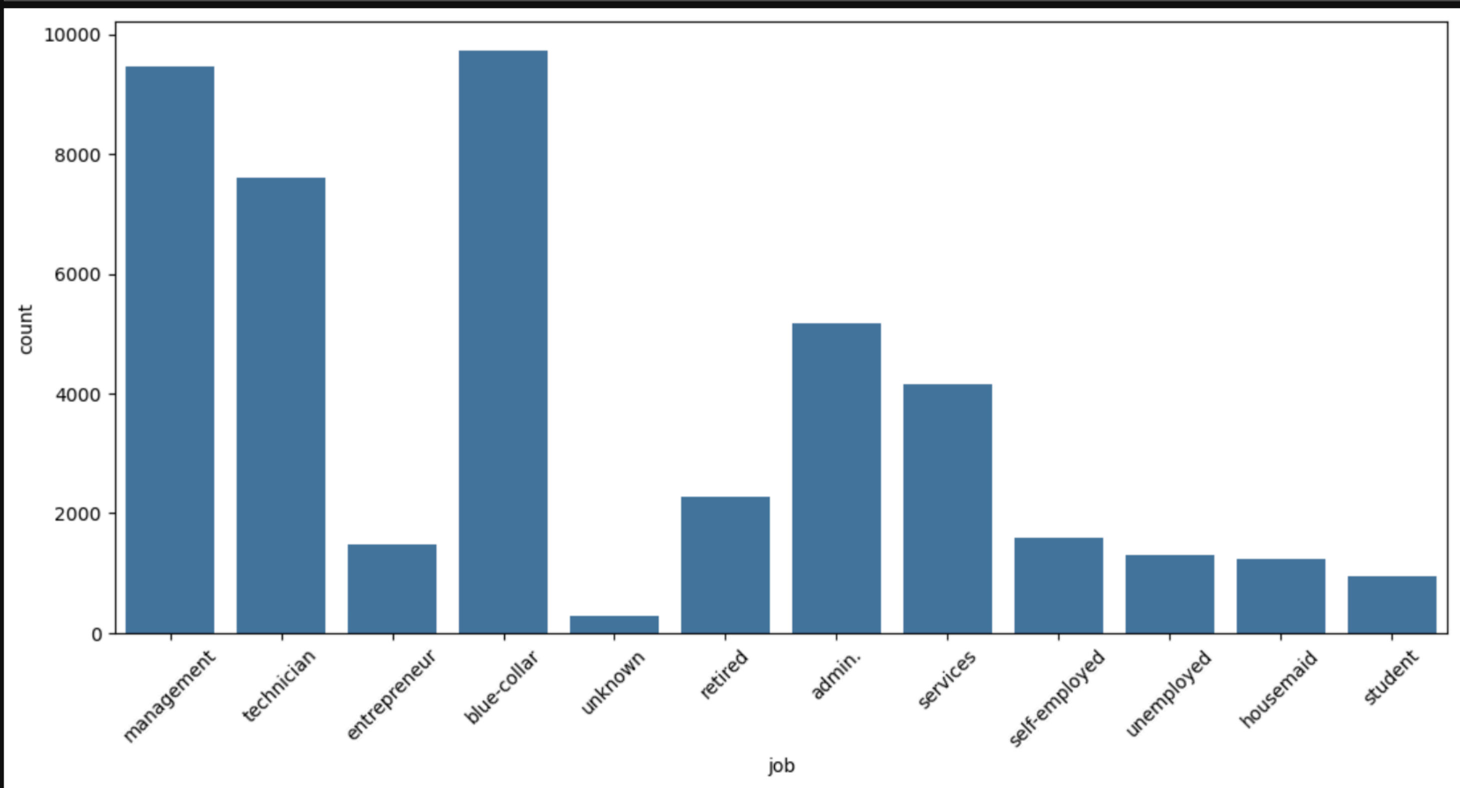
```
plt.figure(figsize=[13,6])
plt.grid()
sns.boxplot(data=dataframe, x='age');
```




The distribution of age seems fairly normal with a small standard deviation .

2. How does the job type vary among the clients ?

```
plt.figure(figsize=[13, 6])
sns.countplot(data=dataframe, x='job');
plt.xticks(rotation=45);
```



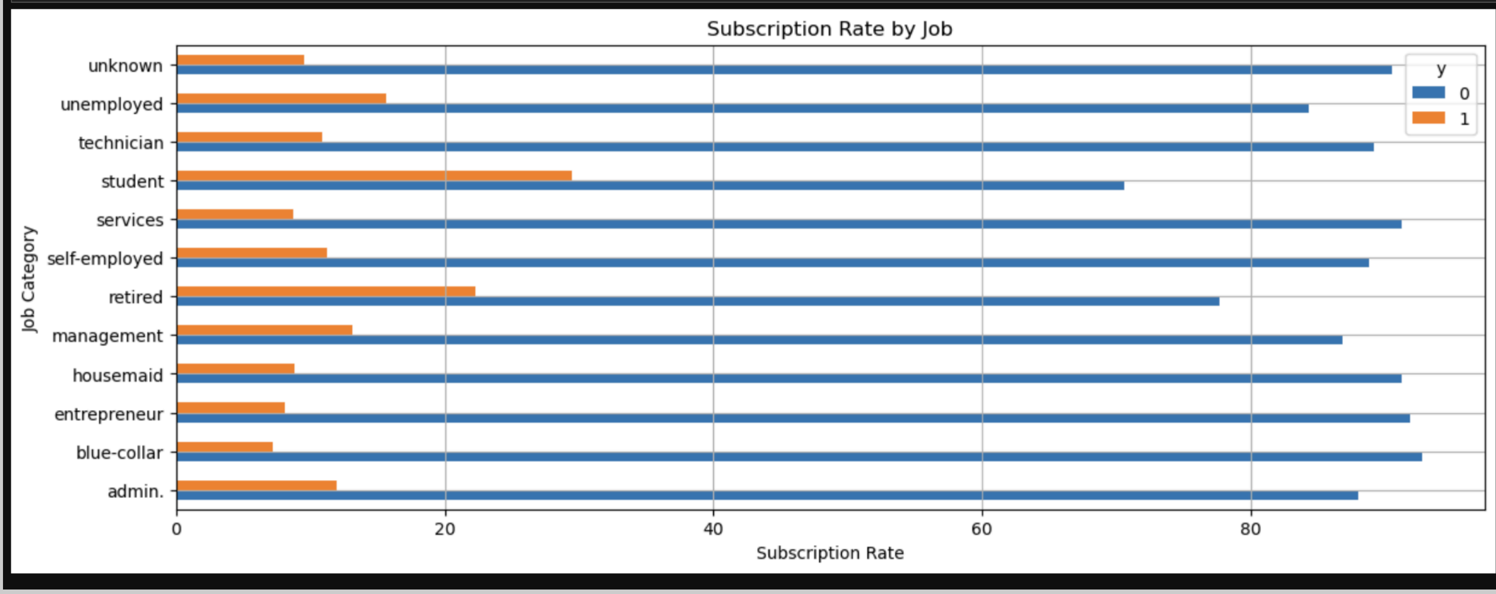
```
]: dataframe['job'].value_counts()

]: job
   blue-collar      9732
   management       9460
   technician       7597
   admin.           5171
   services         4154
   retired          2267
   self-employed    1579
   entrepreneur     1487
   unemployed       1303
   housemaid        1240
   student           938
   unknown           288
   Name: count, dtype: int64
```

There are 12 distinctions in the 'job' column .

- 'blue-collar' , 'management' and 'technician' job types share the majority of the distribution
- 'retired', 'entrepreneur' and 'self-employed' clients have a significant share
- Fair amount of unpaid jobs like ‘student’ and ‘unemployed’

Checking the subscription rate with respect to jobs

```
count_job_response_act = pd.crosstab(dataframe['y'], dataframe['job']).apply(lambda x : x/x.sum()*100)
count_job_response_act = count_job_response_act.transpose()
count_job_response_act.plot(kind='barh', figsize=[14, 5])
plt.grid()
plt.title('Subscription Rate by Job')
plt.xlabel('Subscription Rate')
plt.ylabel('Job Category');
```



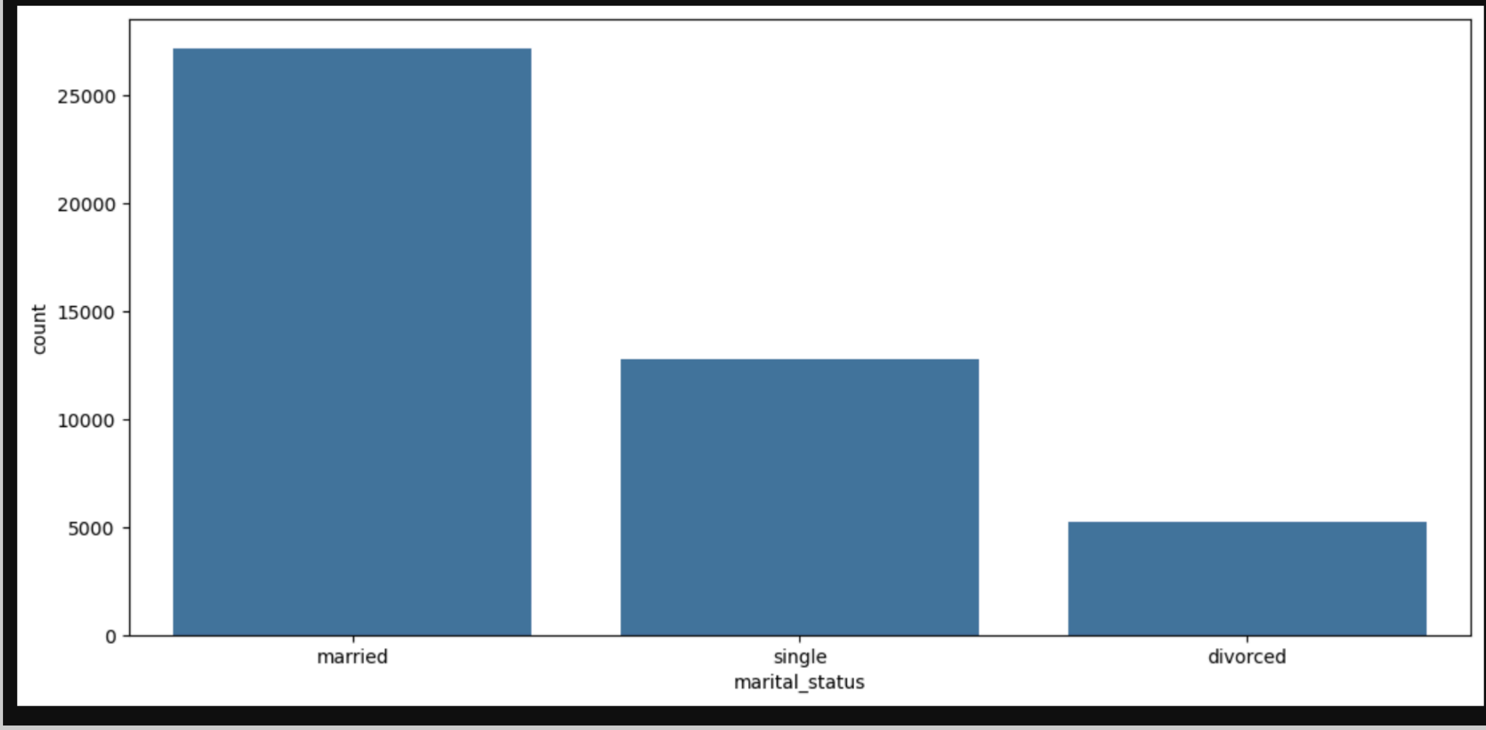

Subscription ‘rate’ amongst students and retired clients are the highest .

3. What is the marital status distribution of the clients?

```
dataframe['marital_status'].value_counts()
```

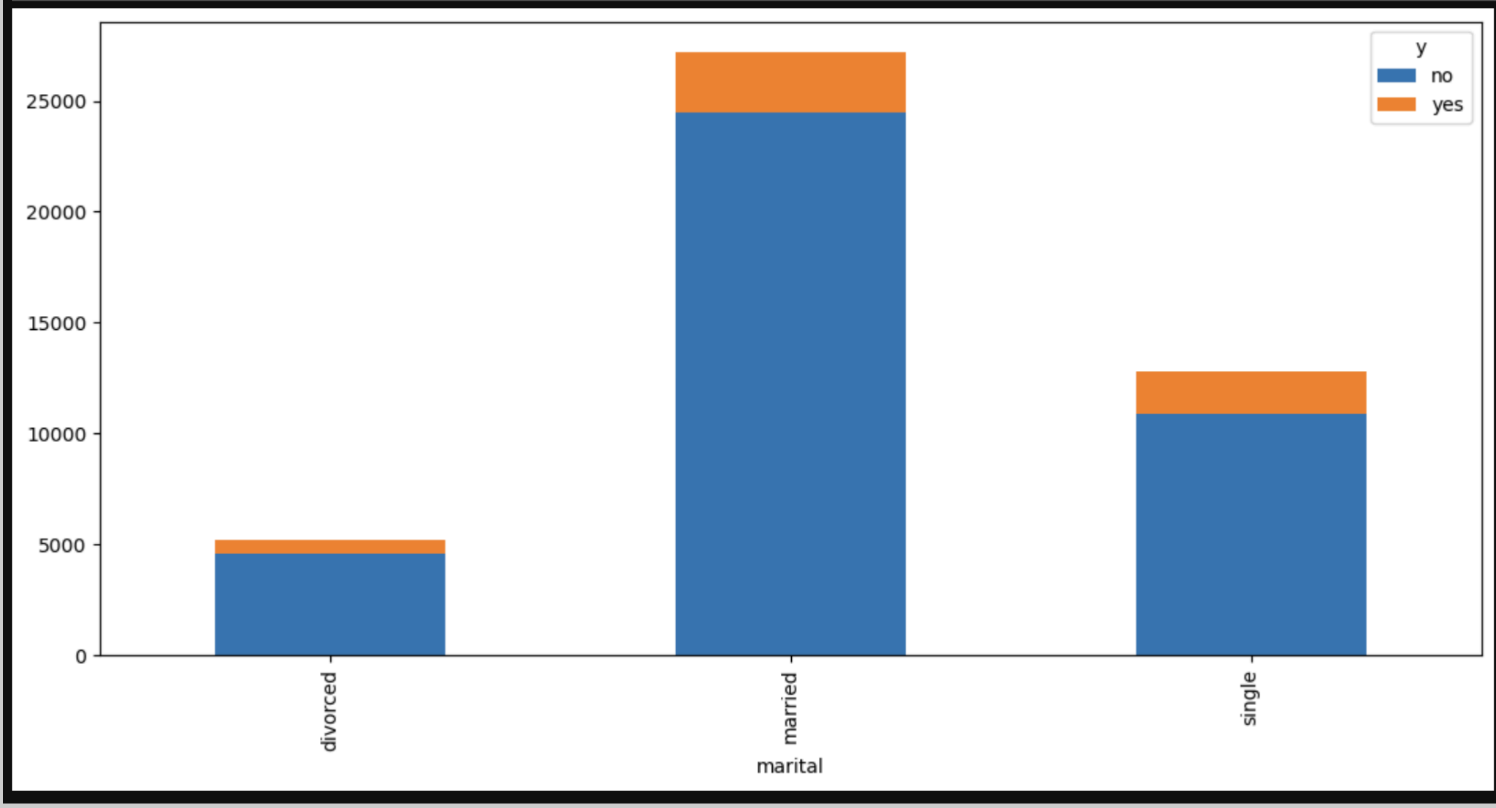
```
marital_status
married     27216
single      12790
divorced     5207
Name: count, dtype: int64
```

```
plt.figure(figsize=[13, 6])
sns.countplot(data=dataframe, x='marital_status');
```




Most of the clients contacted were married , followed by single and divorced clients .

```
grouped = dataframe.groupby('marital')['y'].value_counts().unstack()
grouped.plot(kind='bar', stacked=True, figsize=[13, 6]);
```

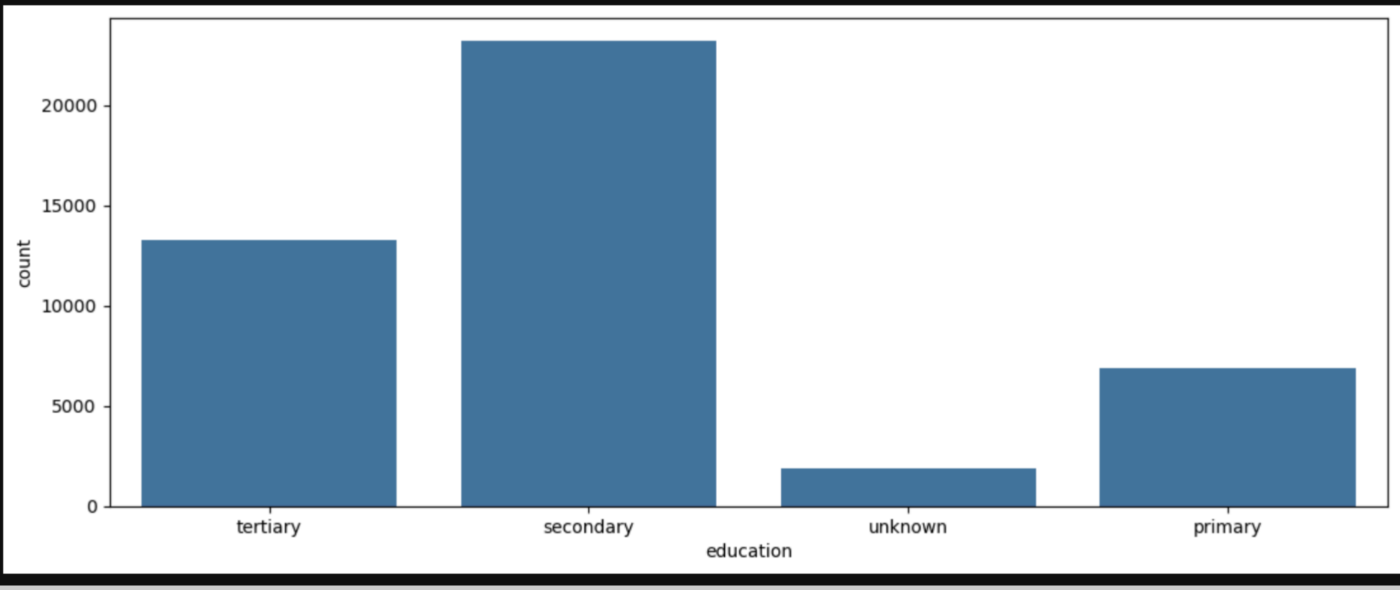



Rate of subscription amongst all 3 groups :

- ‘single’ clients have the highest subscription rate with 15 % .
- ‘divorced’ clients ( 12 % ) followed by ‘married’ clients ( 10 % )

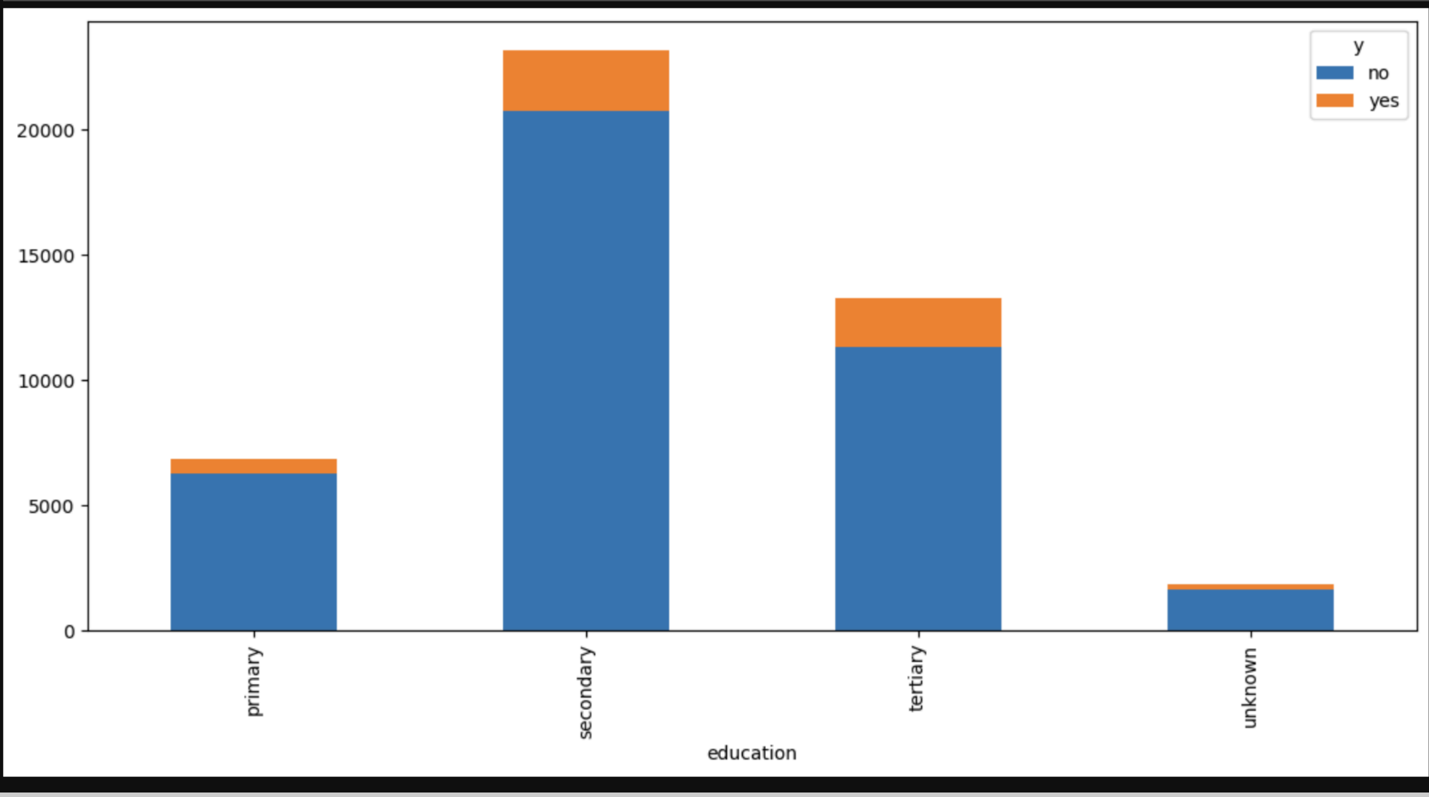
# 4. What is the level of education among the clients ?

```
[47]: print(f"Unique categories for the 'education' column : {dataframe['education'].nunique()}")

      Unique categories for the 'education' column : 4

[48]: dataframe['education'].value_counts()

[48]: education
      secondary    23204
      tertiary     13301
      primary       6851
      unknown       1857
      Name: count, dtype: int64

[50]: plt.figure(figsize=[13, 5])
      sns.countplot(data=dataframe, x='education');
```




Clients with a secondary level of education are the highest , followed by tertiary and primary . Clients whose educational qualifications are unknown are also present .

```
8]: grouped = dataframe.groupby('education')['y'].value_counts().unstack()
    grouped.plot(kind='bar', stacked=True, figsize=[13, 6]);
```

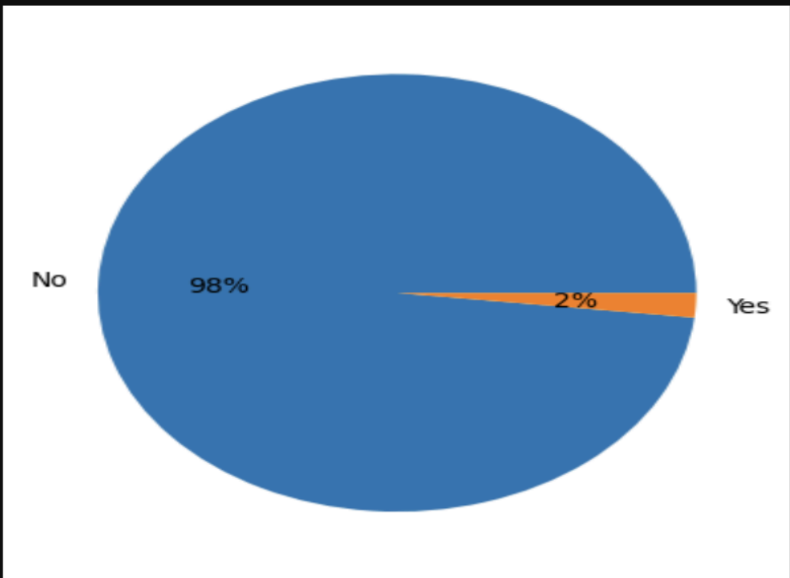



- Subscription percentage highest amongst clients with tertiary level of education ( 15% )

- Lowest subscription percentage amongst clients with primary level of education ( 8.5% )
- Clients whose educational qualifications are unknown have a subscription rate of 13.5 %.

5. What proportion of clients have credit in default?

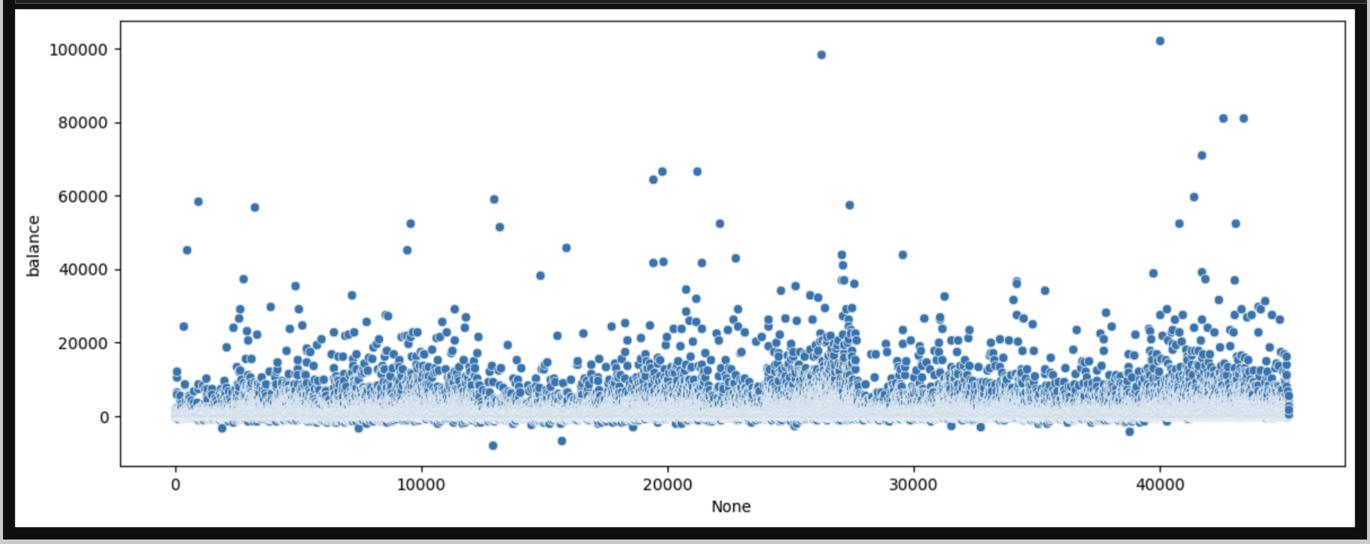
```
[61]: dataframe['default'].value_counts()

[61]: default
      no     44401
      yes      815
      Name: count, dtype: int64

[69]: plt.pie(x=dataframe['default'].value_counts(), labels=['No','Yes'], autopct="%0.0f%%");
```




Nearly 2% of clients have credit in default .

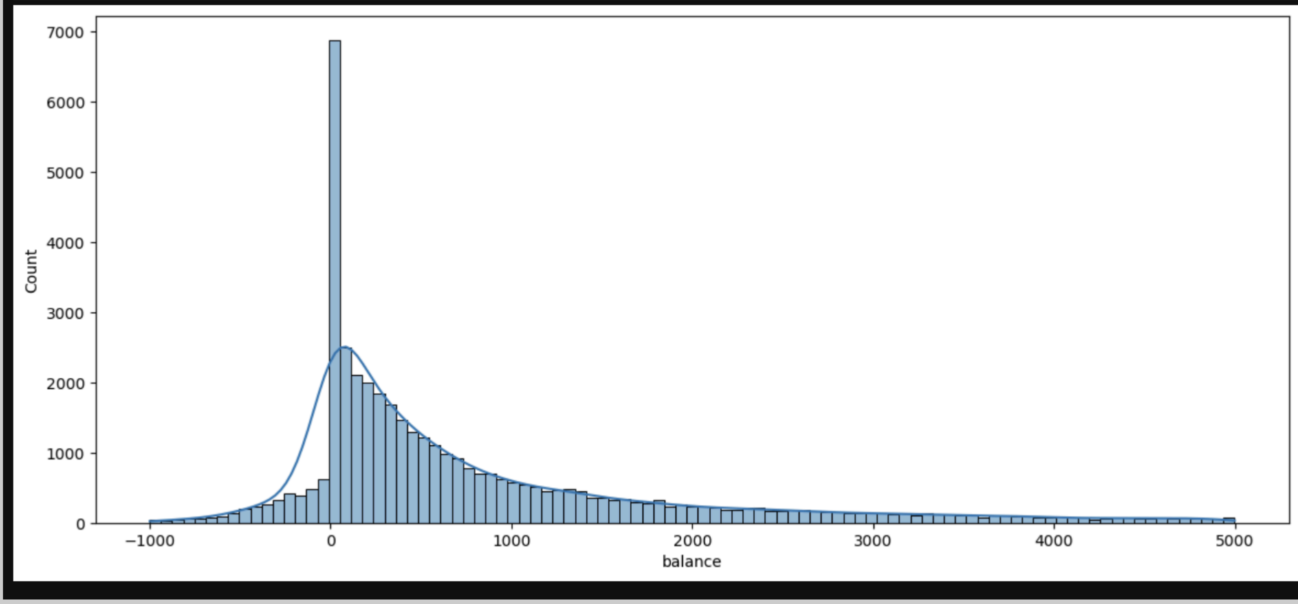
6. What is the distribution of average yearly balance among the clients?

```
plt.figure(figsize=[14,5])
sns.scatterplot(data=dataframe, x=dataframe.index, y='balance');
```




Removing outliers to get a better understanding of the distribution

```
# Magnifying
plt.figure(figsize=[14,6])
sns.histplot(data=dataframe[(dataframe['balance'] < 5000) & (dataframe['balance'] > -1000)], x='balance', kde=True);
```

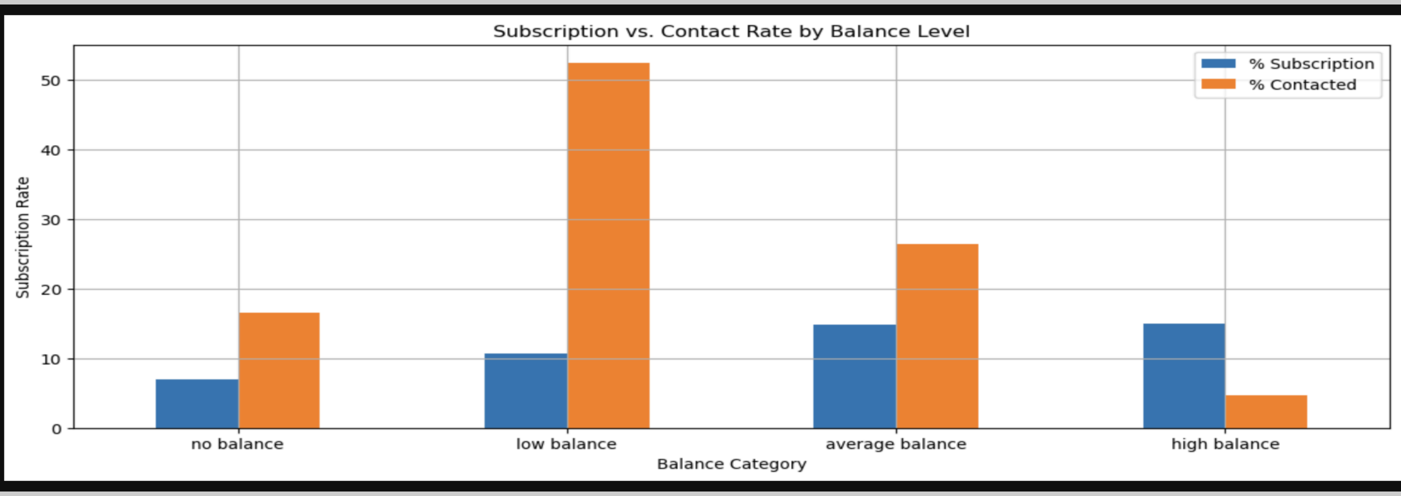



```
lst = [dataframe]
for column in lst:
    column.loc[column["balance"] <= 0,  'balance_group'] = 'no balance'
    column.loc[(column["balance"] > 0) & (column["balance"] <= 1000), 'balance_group'] = 'low balance'
    column.loc[(column["balance"] > 1000) & (column["balance"] <= 5000), 'balance_group'] = 'average balance'
    column.loc[(column["balance"] > 5000), 'balance_group'] = 'high balance'
```

```
count_age_response_act = pd.crosstab(dataframe['y'], dataframe['balance_group']).apply(lambda x : x/x.sum()*100)
count_age_response_act = count_age_response_act.transpose()
# count_age_response_act.rename({1:'Yes', 0:'No'}, inplace=True)
# count_age_response_pct.T['Yes']
count_age_response_act
bal = pd.DataFrame(dataframe['balance_group'].value_counts())
bal.rename(columns={'count':'balance_group'}, inplace=True)
bal['% Contacted'] = bal['balance_group']*100 / bal['balance_group'].sum()
bal['% Subscription'] = count_age_response_act[1]
bal.drop('balance_group', axis=1, inplace=True)
bal['bal'] = [1,2,0,3]
bal = bal.sort_values('bal', ascending=True)
display(bal)

plot_age = bal[['% Subscription','% Contacted']].plot(kind = 'bar',figsize=(14,5),)
plt.xlabel('Balance Category')
plt.ylabel('Subscription Rate')
plt.xticks(np.arange(4), ('no balance', 'low balance', 'average balance', 'high balance'),rotation = 'horizontal')
plt.title('Subscription vs. Contact Rate by Balance Level')
plt.grid()
plt.show()
```

| balance_group | % Contacted | % Subscription | bal |
|---|---|---|---|
| no balance | 16.592389 | 6.921488 | 0 |
| low balance | 52.397277 | 10.739321 | 1 |
| average balance | 26.325971 | 14.789322 | 2 |
| high balance | 4.684363 | 14.950340 | 3 |

Converted balance into 4 groups of 'no balance' (neg. balance) , 'low balance' (0-1000 euros) , 'average balance'(1000-5000 euros) and 'high balance' (> 5000 euros)



Inferences drawn are :

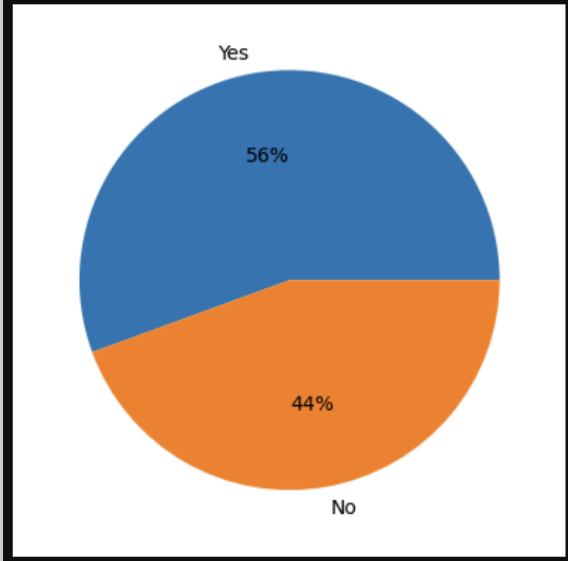
- clients with no balance unsurprisingly returned low subscription rates , whereas clients with high balance had the maximum subscription rate

## 7. How many clients have housing loans?

```
dataframe['housing'].value_counts()
```

```
housing
yes    25130
no     20086
Name: count, dtype: int64
```

```
plt.pie(x=dataframe['housing'].value_counts(), labels=['Yes','No'], autopct="%0.0f%%");
```




**Inference**

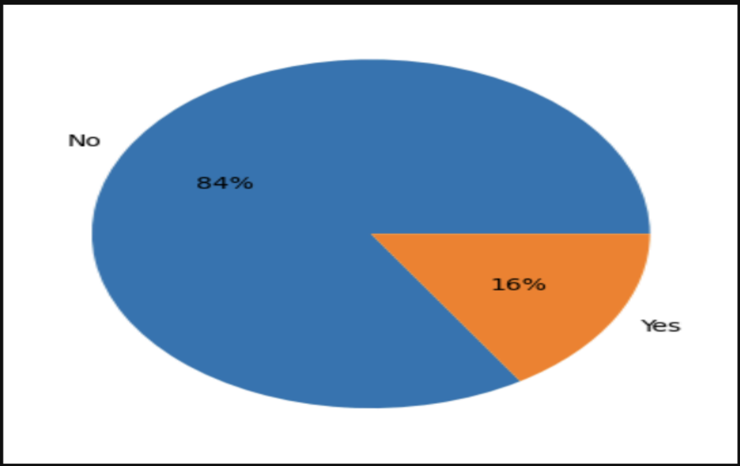
- About ~25000 people have housing loans
- About ~20000 people do not have housing loans

## 8. How many clients have personal loans?

```
dataframe['loan'].value_counts()
```

```
loan
no      37972
yes      7244
Name: count, dtype: int64
```

```
plt.pie(x=dataframe['loan'].value_counts(), labels=['No','Yes'], autopct="%0.0f%%");
```



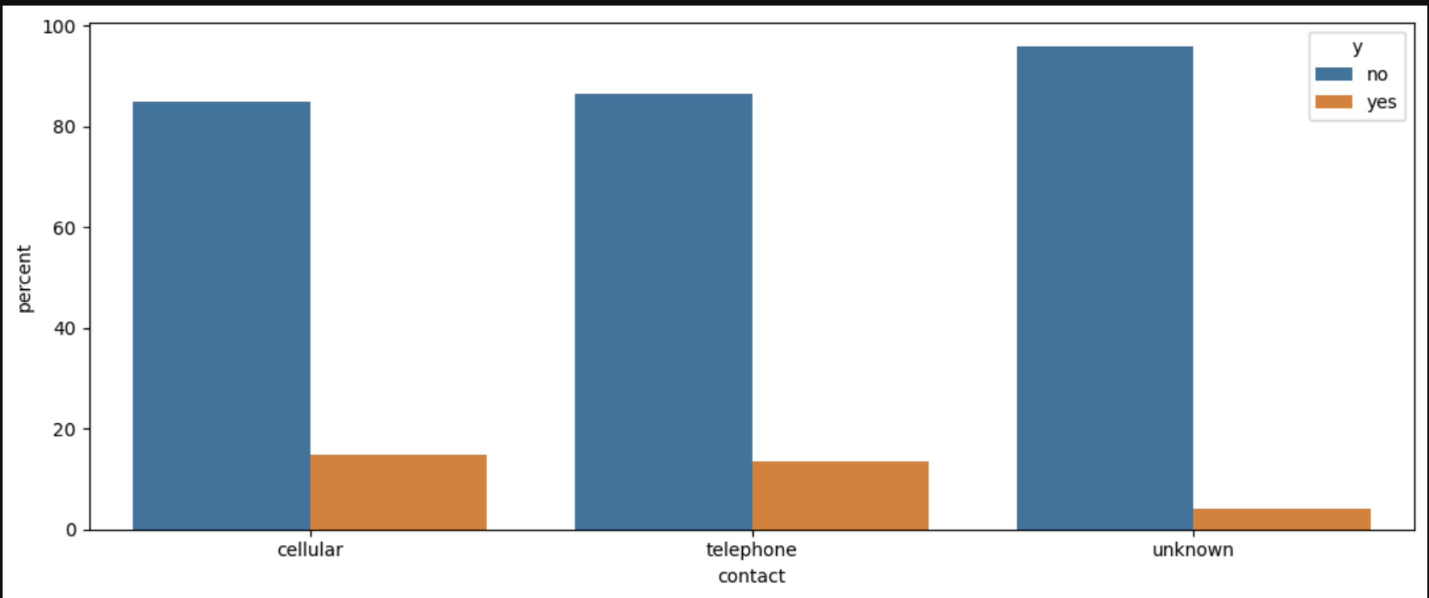

**Inference**

- Around ~38000 clients **do not** have a personal loan
- Around ~7000 clients have personal loan

9. What are the communication types used for contacting clients during the campaign?

```python
contact_percentages = dataframe.groupby('contact')['y'].value_counts(normalize=True).mul(100).rename('percent').reset_index()
display(contact_percentages)
plt.figure(figsize=[13, 5])
sns.barplot(data=contact_percentages, x='contact', y='percent', hue='y');
```

| | contact | y | percent |
|---|---|---|---|
| 0 | cellular | no | 85.066576 |
| 1 | cellular | yes | 14.933424 |
| 2 | telephone | no | 86.579491 |
| 3 | telephone | yes | 13.420509 |
| 4 | unknown | no | 95.929339 |
| 5 | unknown | yes | 4.070661 |



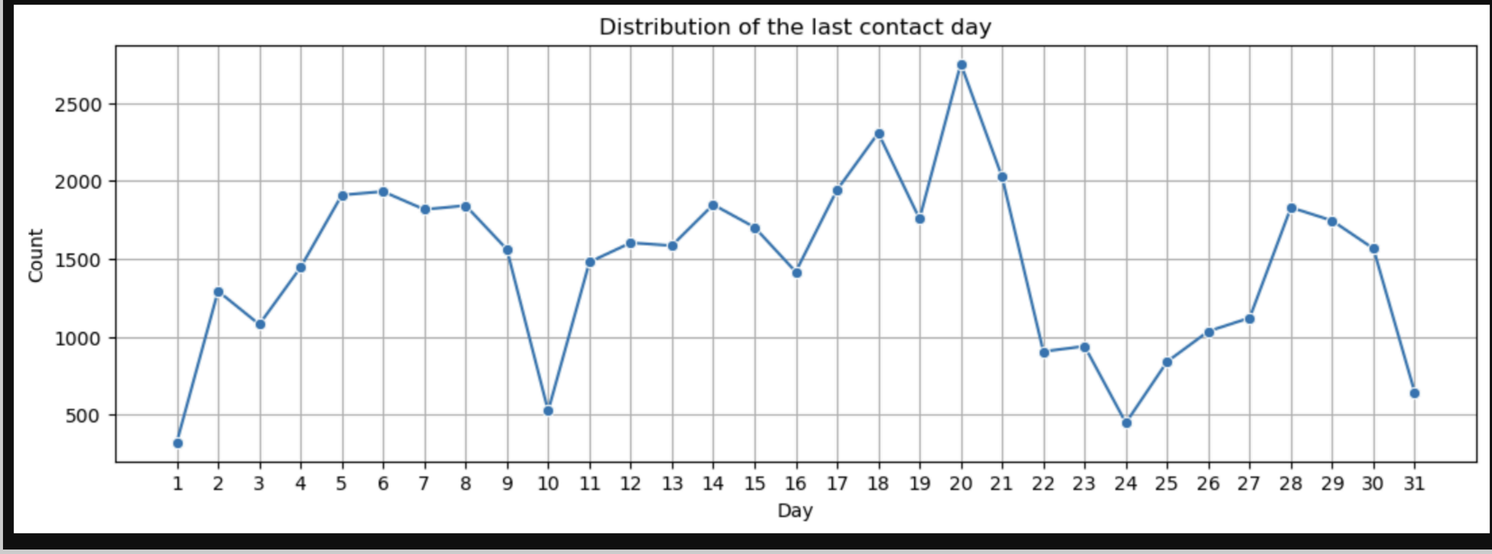

The communication types used for contacting clients during the campaign are :

- Cellular
- Telephone
- Another method of communication that has not been listed

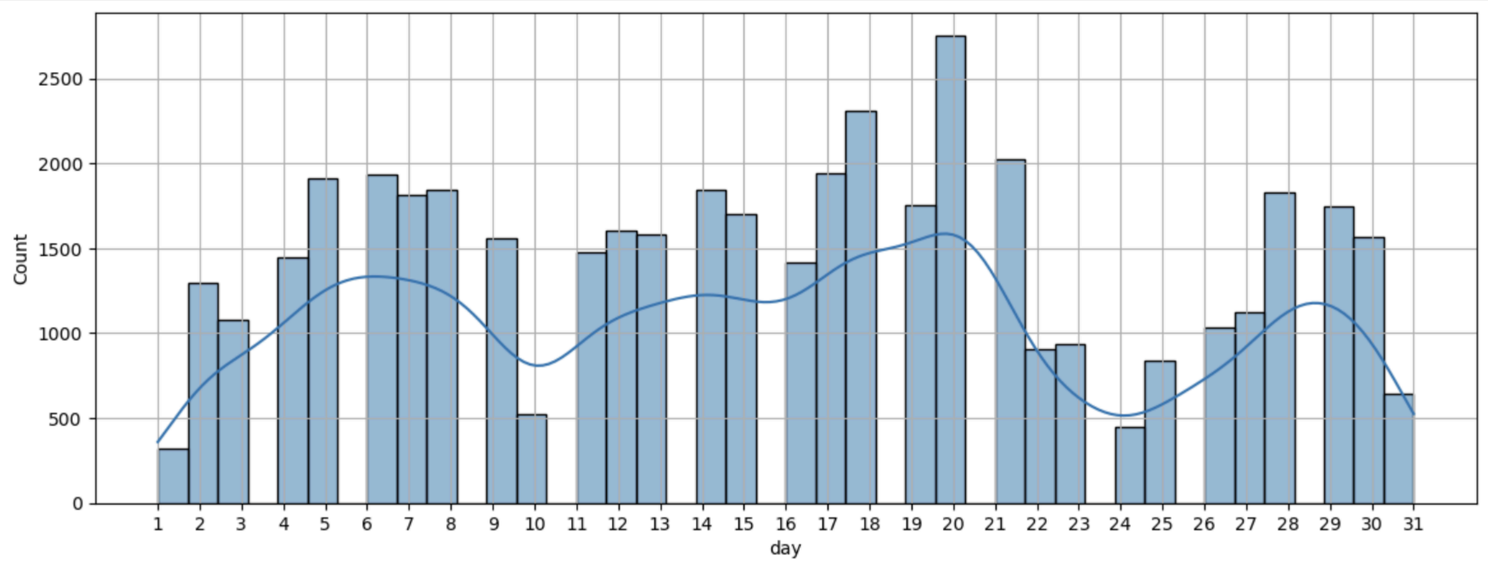
Both cellular and telephone contact method shares almost the same rate of subscription

10. What is the distribution of the last contact day of the month?

```python
plt.figure(figsize=[13, 4])
sns.lineplot(data=pd.DataFrame(dataframe['day'].value_counts()), x='day', y='count', marker='o');
plt.grid()
plt.title("Distribution of the last contact day")
plt.ylabel("Count")
plt.xlabel("Day")
plt.xticks(range(1,32));
```



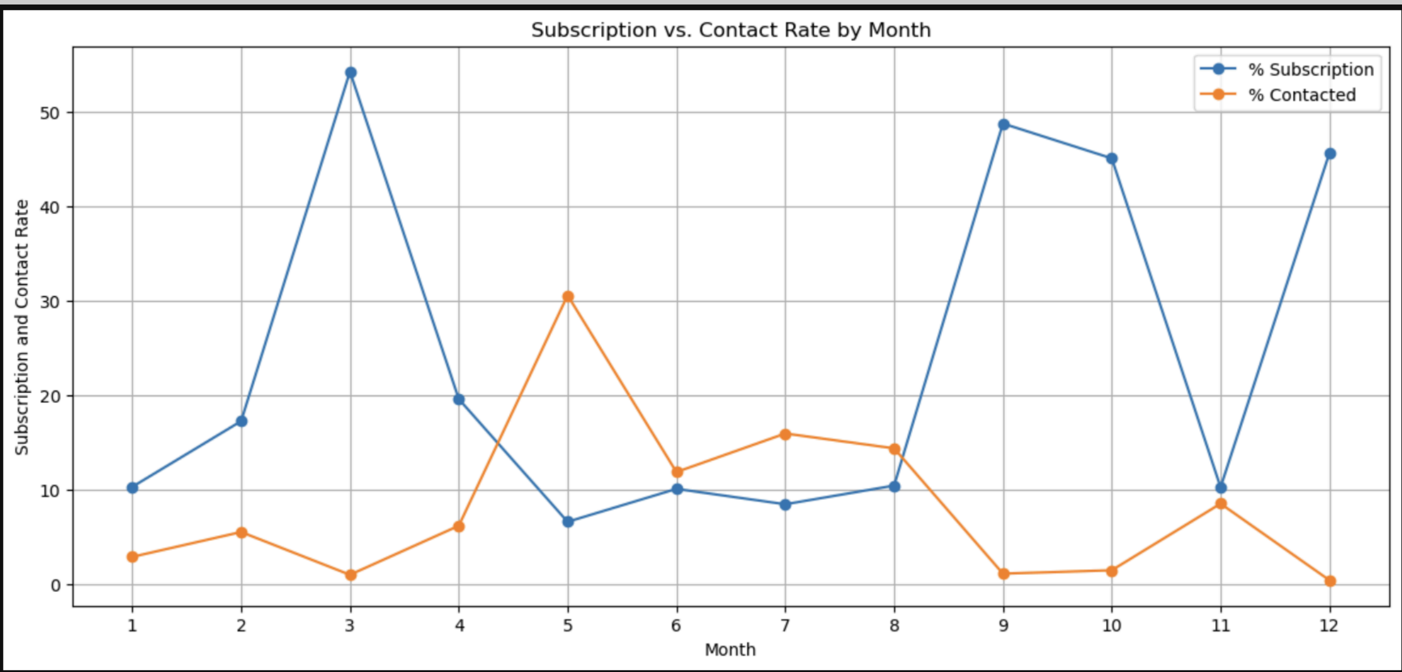

```python
plt.figure(figsize=[14,5])
plt.grid()
sns.histplot(data=dataframe, x='day', kde=True);
plt.xticks(range(1,32));
```




- Last contact day for the maximum number of clients was the 20th of every month .
- Least number of last day of contact was the first day of the month .

11. How does the last contact month vary among the clients?

```
count_month_response_pct = pd.crosstab(dataframe['y'],dataframe['month_int']).apply(lambda x: x/x.sum() * 100)
count_month_response_pct = count_month_response_pct.transpose()
month = pd.DataFrame(dataframe['month_int'].value_counts())
month.rename(columns={'count':'month_int'}, inplace=True)
month['% Contacted'] = month['month_int']*100/month['month_int'].sum()
month['% Subscription'] = count_month_response_pct[1]

month.drop('month_int',axis = 1,inplace = True)
month['Month'] = [5,7,8,6,11,4,2,1,10,9,3,12]

month = month.sort_values('Month',ascending = True)
# display(month)

month[['% Subscription', '% Contacted']].plot(kind='line', figsize=[14,6], marker='o');
plt.xticks(np.arange(1,13,1))
plt.grid()
plt.title('Subscription vs. Contact Rate by Month')
plt.ylabel('Subscription and Contact Rate')
plt.xlabel('Month');
```

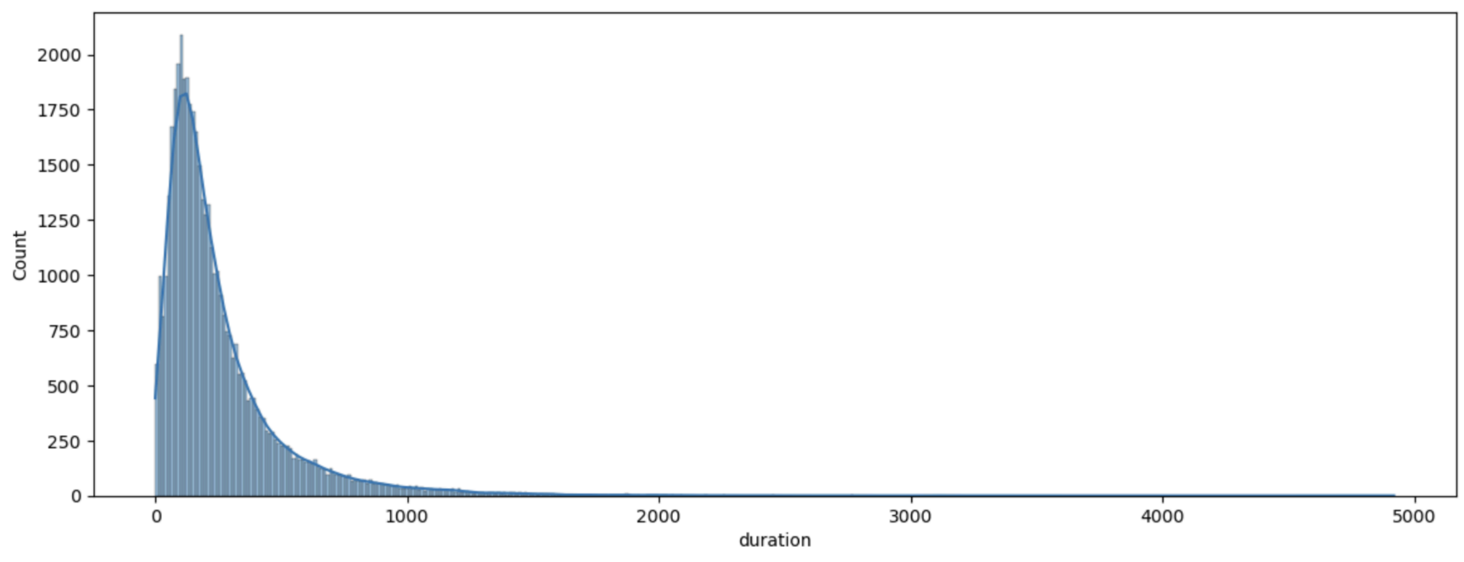



- The maximum number of contacts by the bank happened between May and August , with the highest contact rate happening at 30% in the month of May .
- Contacts rate closer to 0 from Jan - April and Sept-Dec .
- Subscription rate shows a different trend , with the highest subscription rate occurring in March at over 50% , followed by September , October and December .

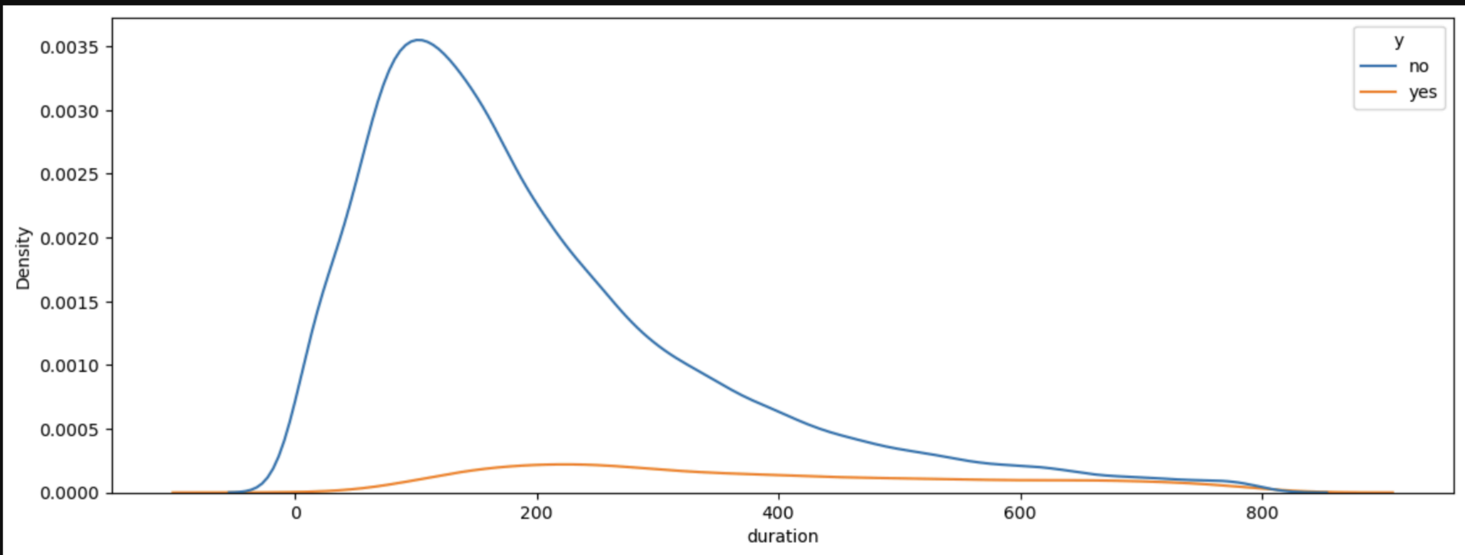
The movement of the lines show different trends , indicating the bank's lack of efficiency in recognising the most opportune timing to hold the bank’s marketing campaign .

## 12. What is the distribution of the duration of the last contact ?

```python
plt.figure(figsize=[14, 5])
sns.histplot(data=dataframe, x='duration', kde=True);
```




```python
# Magnifying
plt.figure(figsize=[14, 5])
sns.kdeplot(data=dataframe[dataframe['duration']<800], x='duration', hue='y');
```




Not a lot can be said about the duration of the call for people who subscribe to a term deposit plan as they are spread across different duration .

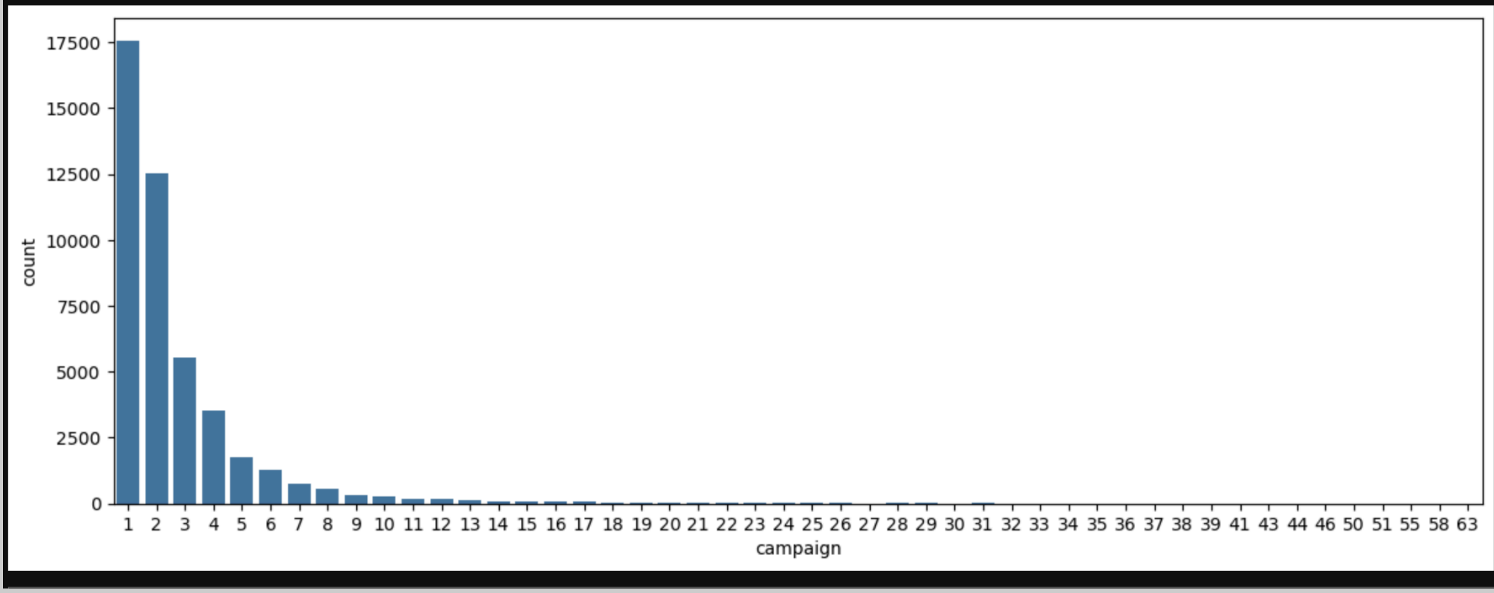
However , for people not subscribing , it usually occurs between 0 - 350/400 seconds .

13. How many contacts were performed during the campaign for each client ?

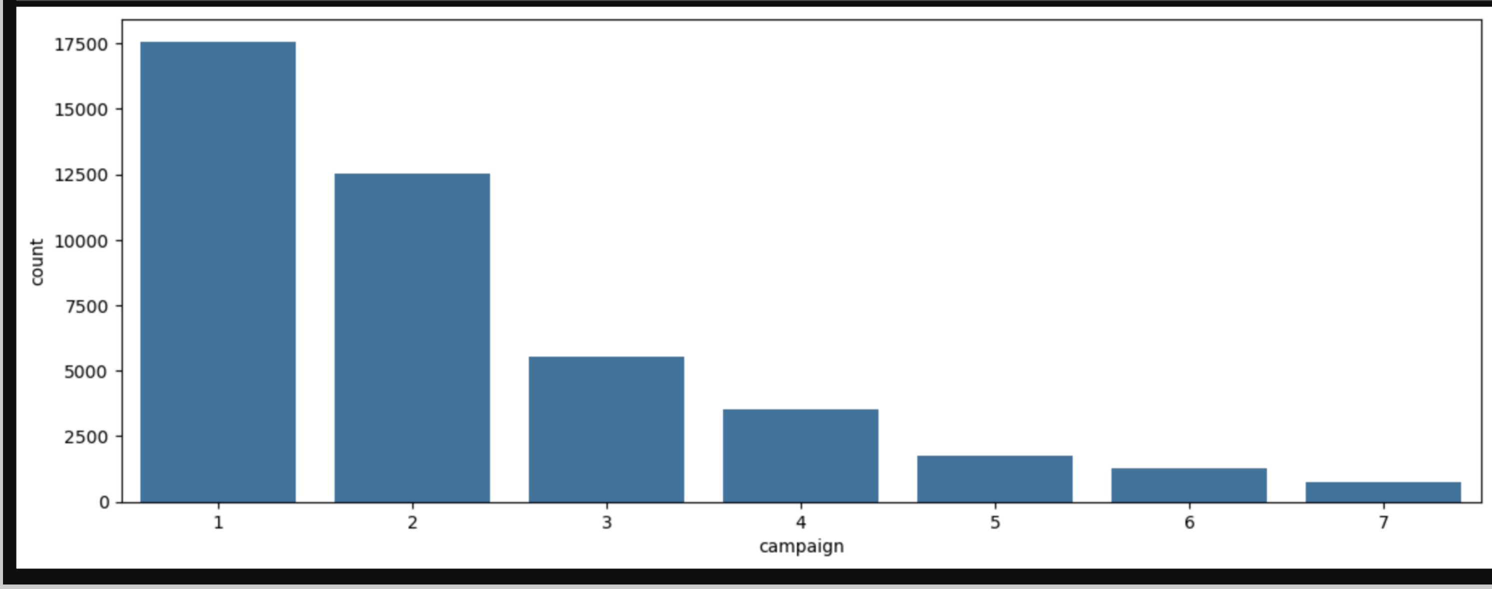
```
dataframe['campaign'].unique()
```

```
array([ 1,  2,  3,  5,  4,  6,  7,  8,  9, 10, 11, 12, 13, 19, 14, 24, 16,
       32, 18, 22, 15, 17, 25, 21, 43, 51, 63, 41, 26, 28, 55, 50, 38, 23,
       20, 29, 31, 37, 30, 46, 27, 58, 33, 35, 34, 36, 39, 44])
```

```
plt.figure(figsize=[14,5])
sns.countplot(data=dataframe, x='campaign');
```




```
plt.figure(figsize=[14,5])
sns.countplot(data=dataframe[dataframe['campaign'] < 8], x='campaign');
```




Client campaigns range from 1 all the way to 44 .

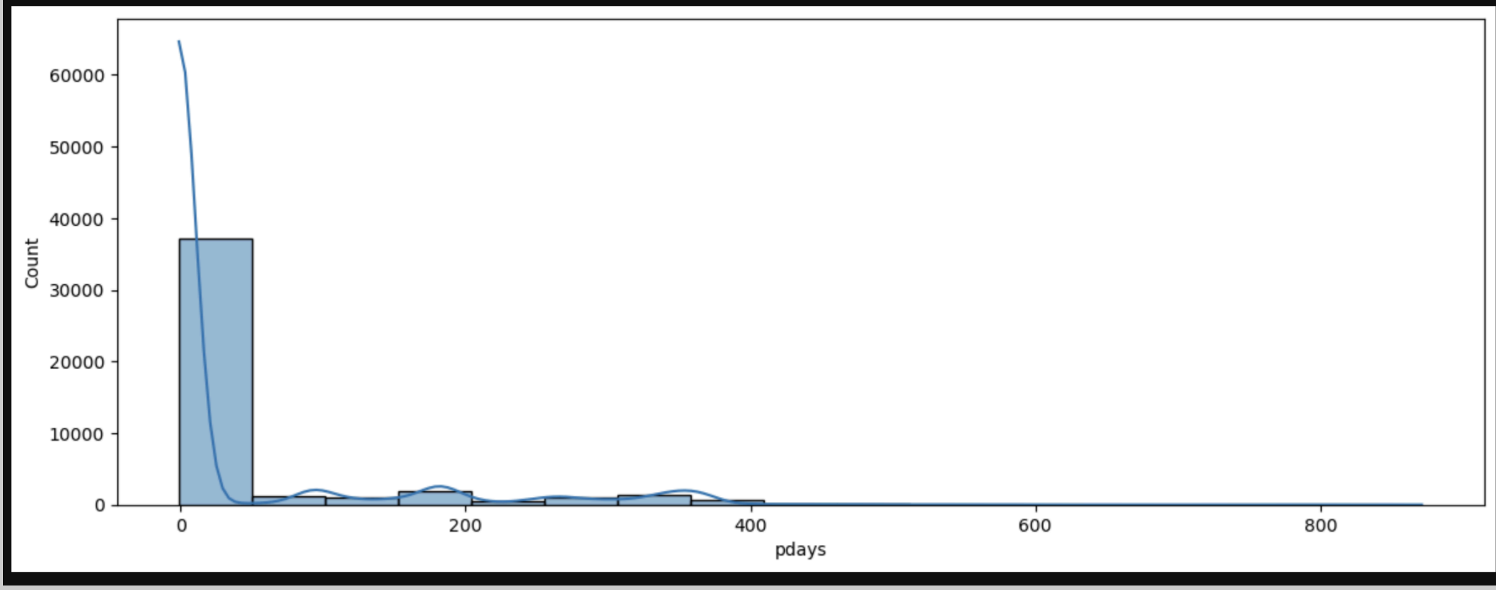
Maximum number of campaigns for clients stands at 1 and the client number keeps on decreasing as the campaign increases

14. What is the distribution of the number of days passed since the client was last contacted from a previous campaign?

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

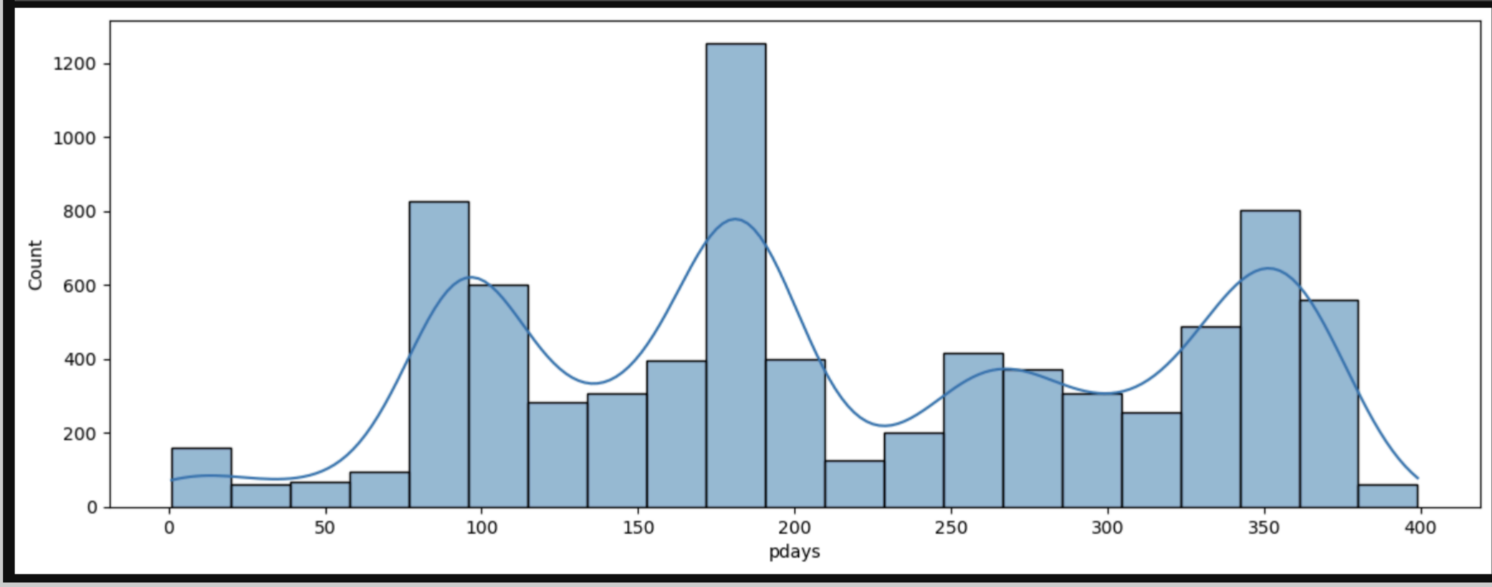
```
count    45216.000000
mean        40.202428
std        100.128248
min         -1.000000
25%         -1.000000
50%         -1.000000
75%         -1.000000
max        871.000000
Name: pdays, dtype: float64
```

```
plt.figure(figsize=[14, 5])
sns.histplot(data=dataframe, x='pdays', kde=True);
```




The majority of clients were not contacted more than once , hence -1 having the bulk share of the distribution , however looking for clients who have been contacted more than once shows …

```
plt.figure(figsize=[14, 5])
sns.histplot(data=dataframe[(dataframe['pdays'] > 0) & (dataframe['pdays'] < 400)], x='pdays', kde=True);
```



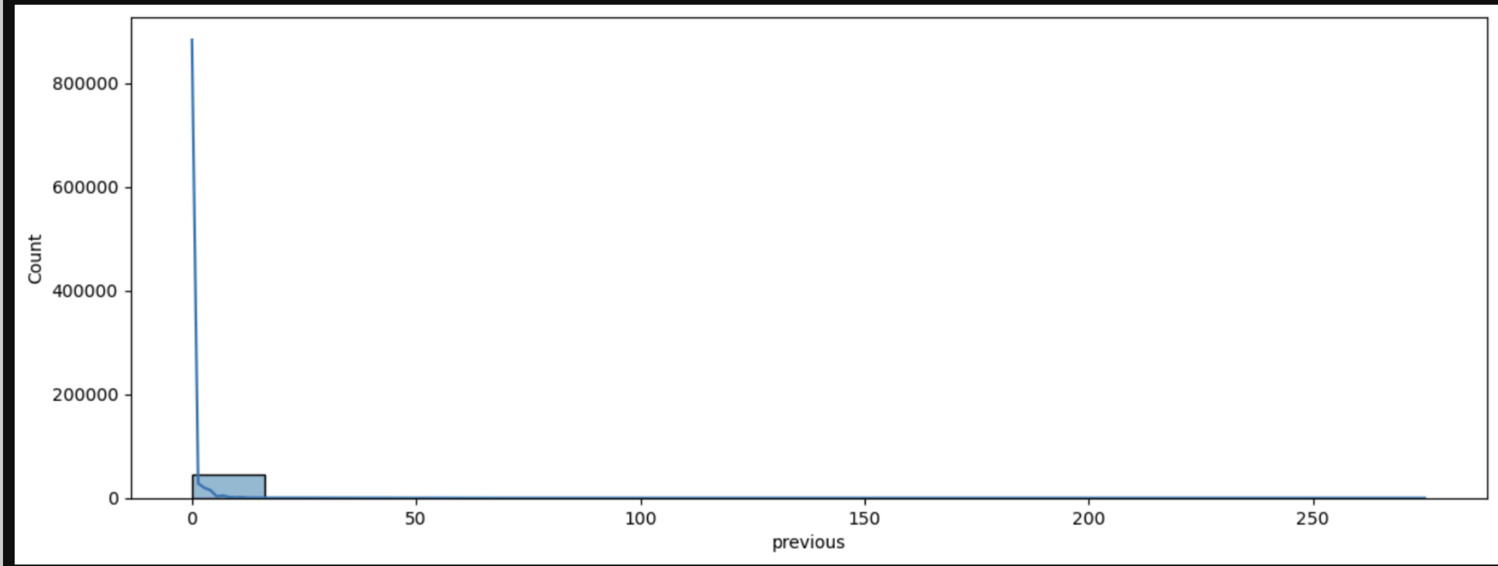

Number of days since they last contacted the clients hovers around 100 , 200 and 350 days .

## 15. How many contacts were performed before the current campaign for each client?

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

```
count    45216.000000
mean         0.580657
std          2.303778
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max        275.000000
Name: previous, dtype: float64
```

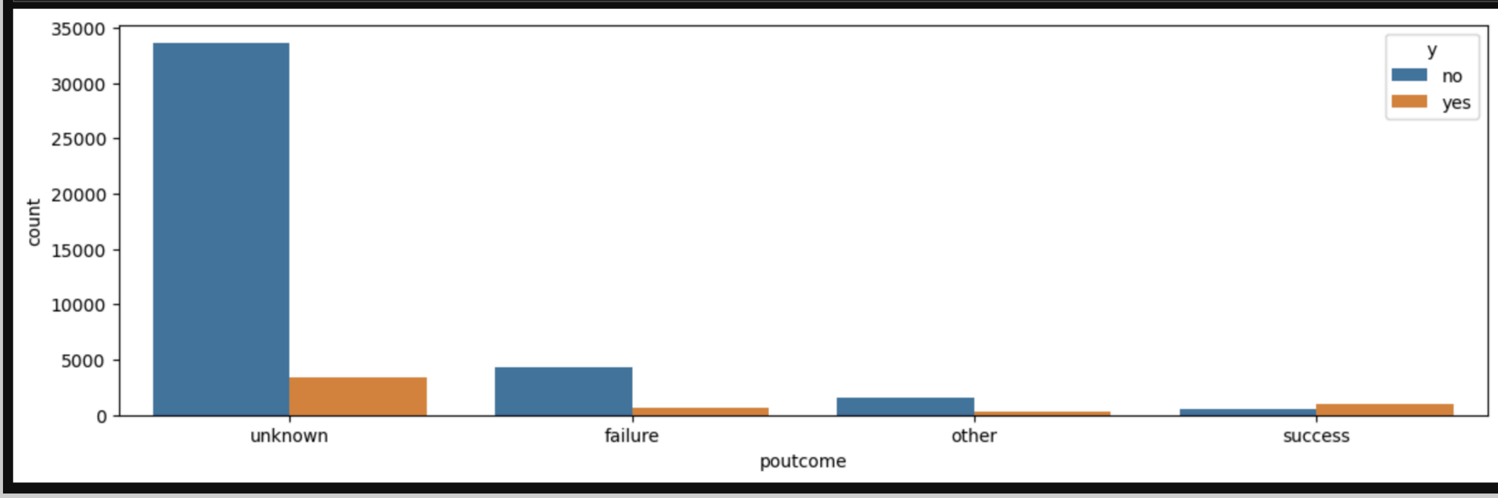
```
plt.figure(figsize=[14, 5])
sns.histplot(data=dataframe, x='previous', kde=True);
```




- Majority have no contacts performed before the start of the current campaign
- Number of clients decreasing as the count of the contacts performed before the current campaign starts increasing

## 16. What were the outcomes of the previous marketing campaigns?

```
plt.figure(figsize=[14, 4])
sns.countplot(data=dataframe, x='poutcome', hue='y');
```




People who failed to convert earlier , did not have much success in getting their rate of subscription higher

However , people who were successful in converting last campaign , also showed promise and had a higher conversion rate for client subscription .

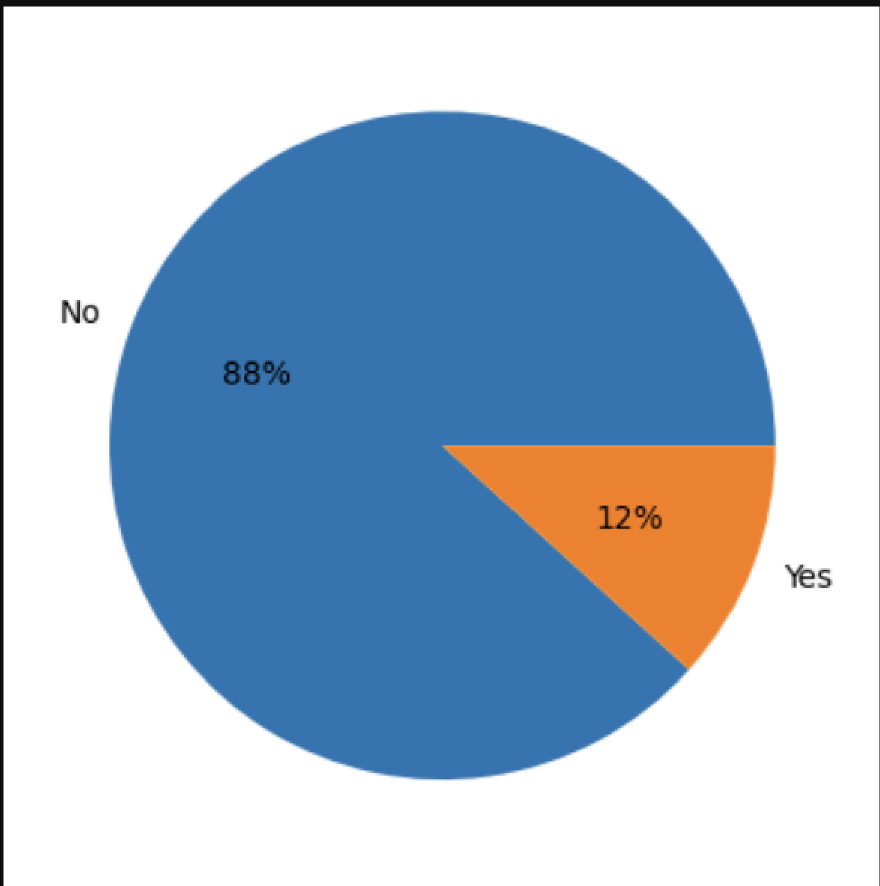
Majority of the data about the outcomes of the previous campaign are unknown .

17. What is the distribution of clients who subscribed to a term deposit vs. those who did not?

```
dataframe['y'].value_counts()
```

```
y
no     39922
yes     5294
Name: count, dtype: int64
```

```
plt.pie(x=dataframe['y'].value_counts(), labels=['No','Yes'], autopct="%0.0f%%");
```




18. Are there any correlations between different attributes and the likelihood of subscribing to a term deposit?

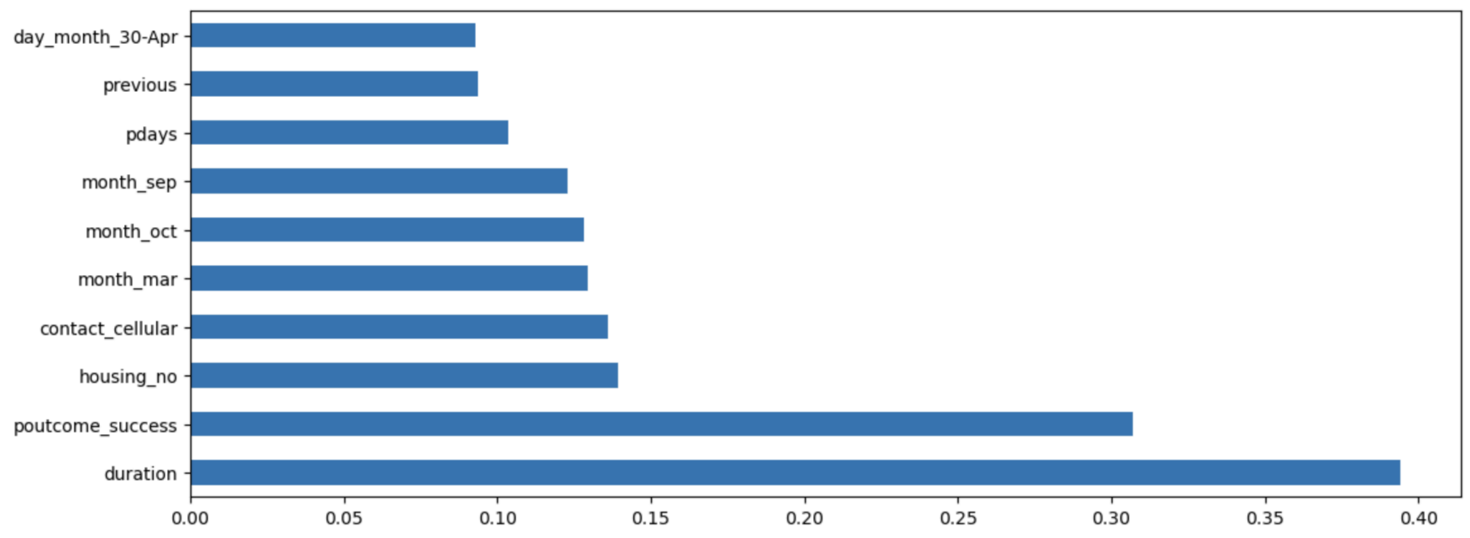
To view the correlation of the target variable (‘y’) with every other feature , I performed

- one-hot encoding for categorical features
- standard scaler operation for numerical features

Upon completion , the top 10 positive and negative correlations look something like this .

Positive correlation :

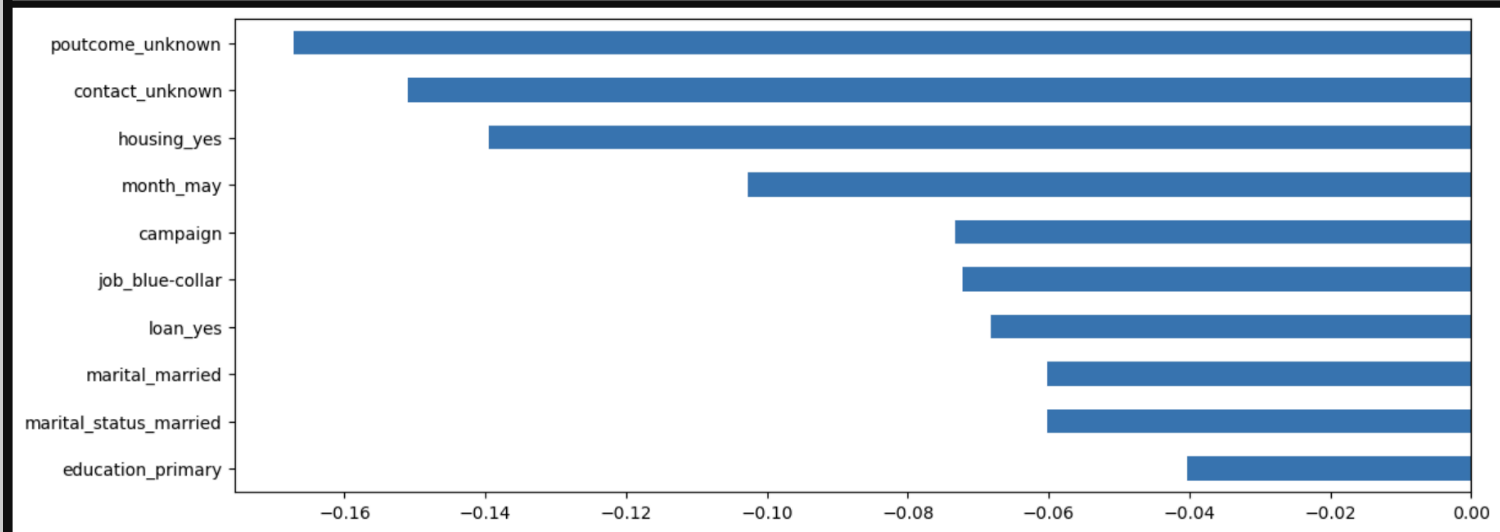
```
correlation_series[:10].plot(figsize=[13, 5], kind='barh');
```




```
correlation_series[:10]

duration            0.394387
poutcome_success    0.307083
housing_no          0.139445
contact_cellular    0.136036
month_mar           0.129371
month_oct           0.128439
month_sep           0.123099
pdays               0.103699
previous            0.093576
day_month_30-Apr    0.092786
Name: y, dtype: float64
```

Negative correlation

```
correlation_series[-10:].plot(figsize=[13, 5], kind='barh');
```



```
correlation_series[-10:]

education_primary         -0.040313
marital_status_married    -0.060216
marital_married           -0.060216
loan_yes                  -0.068289
job_blue-collar           -0.072211
campaign                  -0.073294
month_may                 -0.102656
housing_yes               -0.139445
contact_unknown           -0.151062
poutcome_unknown          -0.167284
Name: y, dtype: float64
```

We can infer from the information displayed above that :

- duration (duration of the last contact ) showed the highest association with client subscription , followed by the success of poutcome ( Outcome of the previous marketing campaign
- Other features were
  - housing ( whether housing loan was taken )
  - contact ( type of communication used )
  - month ( last contact month of the year )

```
[81]: # Correlation
corr_df = dataframe[['age','balance','duration','campaign','month_int','previous','y']]
plt.figure(figsize=[14,5])
sns.heatmap(corr_df.corr(), annot=True)
```

```
[81]: <Axes: >
```