Banking deposit dataset

Term deposits serve as a significant revenue stream for banks, representing cash investments held within financial institutions. These investments involve committing funds for a predetermined period, during which they accrue interest at an agreed-upon rate. To promote term deposits, banks employ various outreach strategies including email marketing, advertisements, telephonic marketing, and digital marketing.

Despite the advent of digital channels, telephonic marketing campaigns persist as one of the most effective means of engaging customers. However, they necessitate substantial investment due to the requirement of large call centers to execute these campaigns. Therefore, it becomes essential to pre-identify potential customers likely to convert, enabling targeted outreach efforts via phone calls. The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (variable y).

Content The data is related to the direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed by the customer or not.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('banking_data.csv')
df
```

Out[2]:

	age	job	marital	marital_status	education	default	balance	housing	loan	contact	day	month	day_month
0	58	management	married	married	tertiary	no	2143	yes	no	unknown	5	may	5-May
1	44	technician	single	single	secondary	no	29	yes	no	unknown	5	may	5-May
2	33	entrepreneur	married	married	secondary	no	2	yes	yes	unknown	5	may	5-May
3	47	blue-collar	married	married	unknown	no	1506	yes	no	unknown	5	may	5-May
4	33	unknown	single	single	unknown	no	1	no	no	unknown	5	may	5-May
...	...	...	...	...	...	...	...	...	...	...	...	...	...
45211	29	management	single	single	tertiary	no	765	no	no	cellular	16	nov	16-Nov
45212	68	retired	married	married	secondary	no	1146	no	no	cellular	16	nov	16-Nov
45213	53	management	married	married	tertiary	no	583	no	no	cellular	17	nov	17-Nov
45214	73	retired	married	married	secondary	no	2850	no	no	cellular	17	nov	17-Nov
45215	71	retired	divorced	divorced	primary	no	1729	no	no	cellular	17	nov	17-Nov

45216 rows × 19 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45216 entries, 0 to 45215
Data columns (total 19 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             45216 non-null  int64 
 1   job             45216 non-null  object
 2   marital         45213 non-null  object
 3   marital_status  45213 non-null  object
 4   education       45213 non-null  object
 5   default         45216 non-null  object
 6   balance         45216 non-null  int64 
 7   housing         45216 non-null  object
 8   loan            45216 non-null  object
 9   contact         45216 non-null  object
 10  day             45216 non-null  int64 
 11  month           45216 non-null  object
 12  day_month       45216 non-null  object
 13  duration        45216 non-null  int64 
 14  campaign        45216 non-null  int64 
 15  pdays           45216 non-null  int64 
 16  previous        45216 non-null  int64 
 17  poutcome        45216 non-null  object
 18  y               45216 non-null  object
dtypes: int64(7), object(12)
memory usage: 6.6+ MB
```

In [4]:
```
df.isna().any()
```

Out[4]:
```
age               False
job               False
marital            True
marital_status     True
education          True
default           False
balance           False
housing           False
loan              False
contact           False
day               False
month             False
day_month         False
duration          False
campaign          False
pdays             False
previous          False
poutcome          False
y                 False
dtype: bool
```

In [5]:
```
df.isna().sum()
```

Out[5]:
```
age               0
job               0
marital           3
marital_status    3
education         3
default           0
balance           0
housing           0
loan              0
contact           0
day               0
month             0
day_month         0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
y                 0
dtype: int64
```

In [6]:
```
df.shape
```

Out[6]:
```
(45216, 19)
```

In [7]:
```
df.loc[df.marital.isna()].index
```

Out[7]:
```
Index([44996, 45077, 45209], dtype='int64')
```

In [8]:
```
df.loc[df.education.isna()]
```

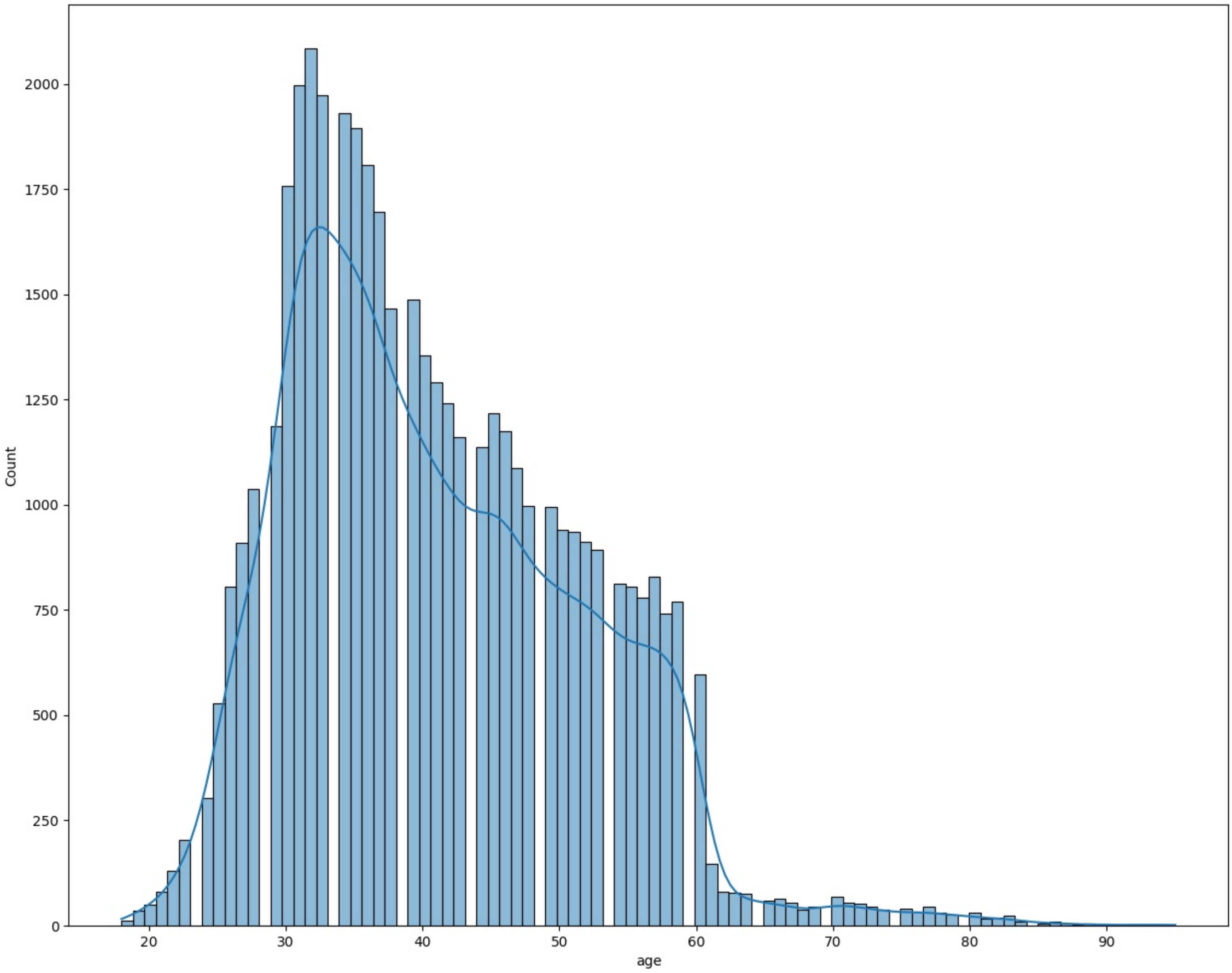
Out[8]:

	age	job	marital	marital_status	education	default	balance	housing	loan	contact	day	month	day_month
44957	32	management	single	single	NaN	no	3289	no	no	cellular	8	oct	8-Oct
45137	30	management	single	single	NaN	no	297	no	no	cellular	8	nov	8-Nov
45170	19	student	single	single	NaN	no	245	no	no	telephone	10	nov	10-Nov

What is the distribution of age among the clients?

In [9]:
```
plt.figure(figsize = (15,12))
sns.histplot(df.age,kde = True)

# df["age"].value_counts().plot(kind="hist")
```

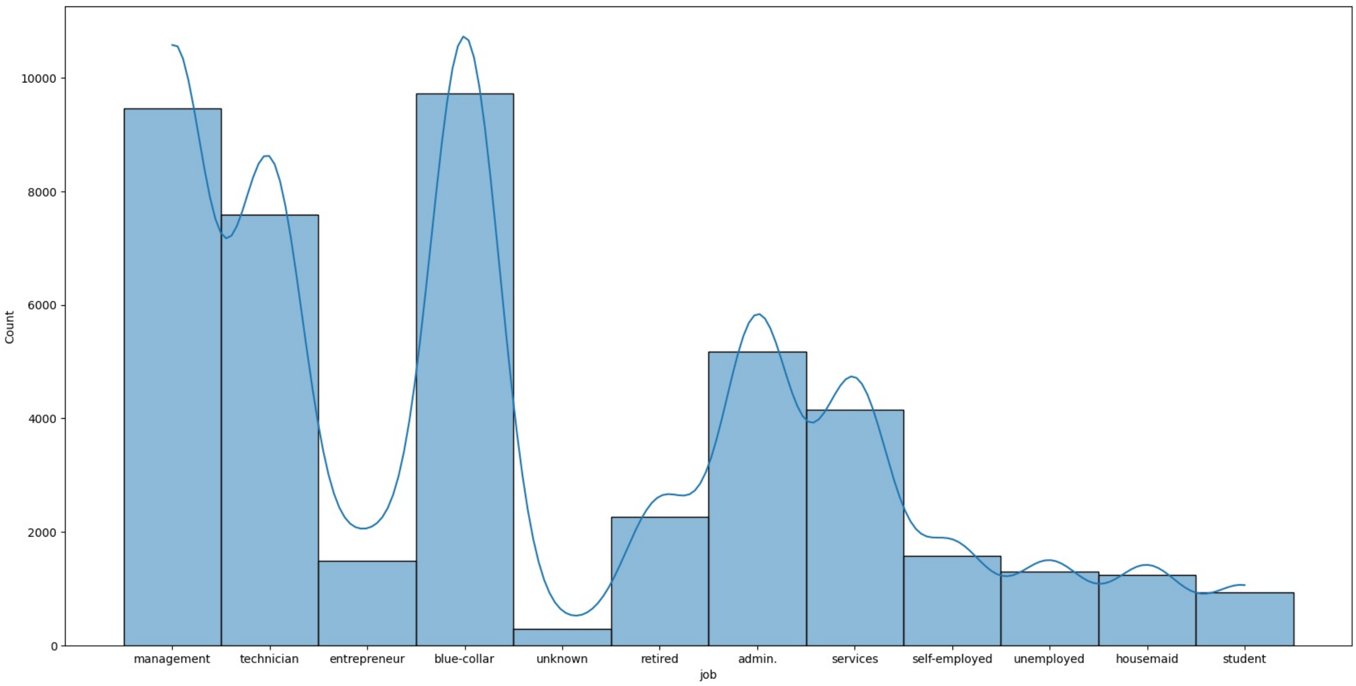
Out[9]:
```
<Axes: xlabel='age', ylabel='Count'>
```



How does the job type vary among the clients?

In [10]:
```
plt.figure(figsize = (20,10))
sns.histplot(df.job,kde = True)

# df["job"].value_counts().plot(kind="bar")
```

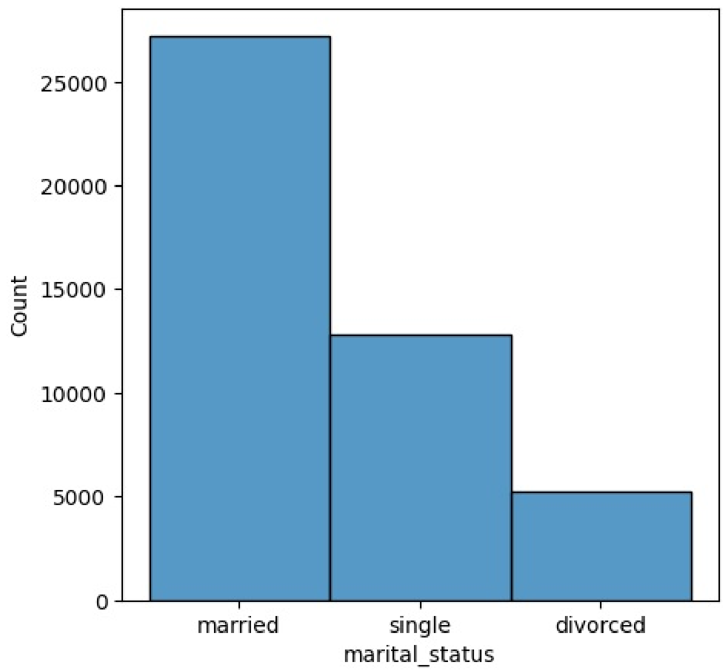
Out[10]: <Axes: xlabel='job', ylabel='Count'>



What is the marital status distribution of the clients?

In [11]:
```
plt.figure(figsize = (5,5))
sns.histplot(df.marital_status)
```

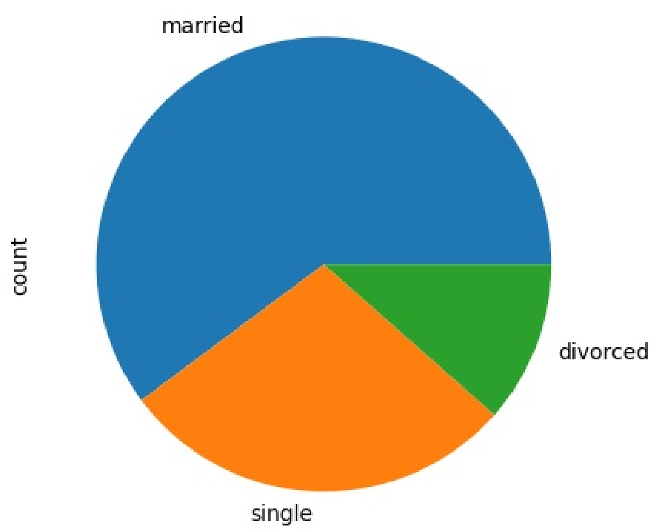
Out[11]: <Axes: xlabel='marital_status', ylabel='Count'>



What is the marital status distribution of the clients?

```
In [12]: df["marital"].value_counts().plot(kind="pie")
```

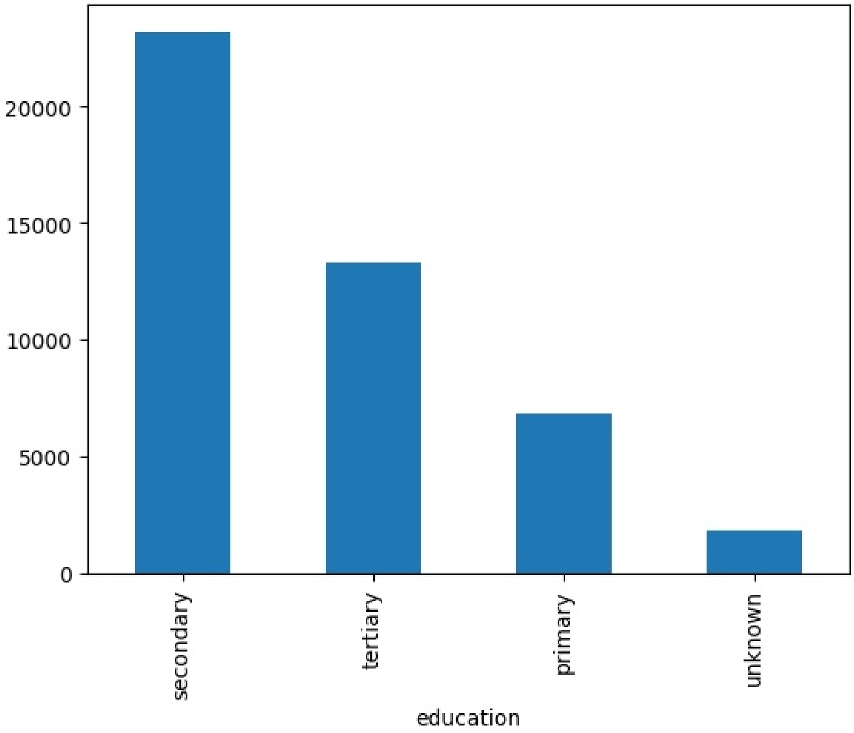
```
Out[12]: <Axes: ylabel='count'>
```



What is the level of education among the clients?

In [13]:
```
df["education"].value_counts().plot(kind="bar")
```

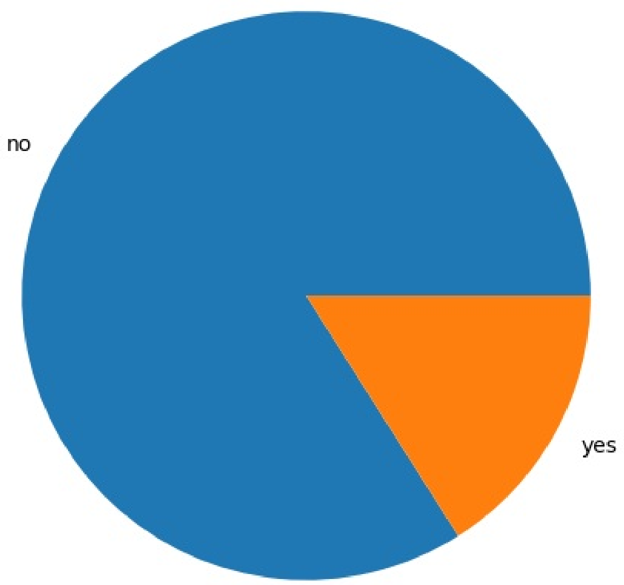
Out[13]: <Axes: xlabel='education'>



What proportion of clients have credit in default?

In [14]:
```
plt.figure(figsize = (10,6))
plt.pie(df.loan.value_counts() , labels = df.loan.unique())

default_count = df["default"].value_counts()["yes"]
total_clients = len(df)
default_proportion = default_count / total_clients
print(f"Proportion of clients with credit in default: {default_proportion:.2f}")
```

```
Proportion of clients with credit in default: 0.02
```

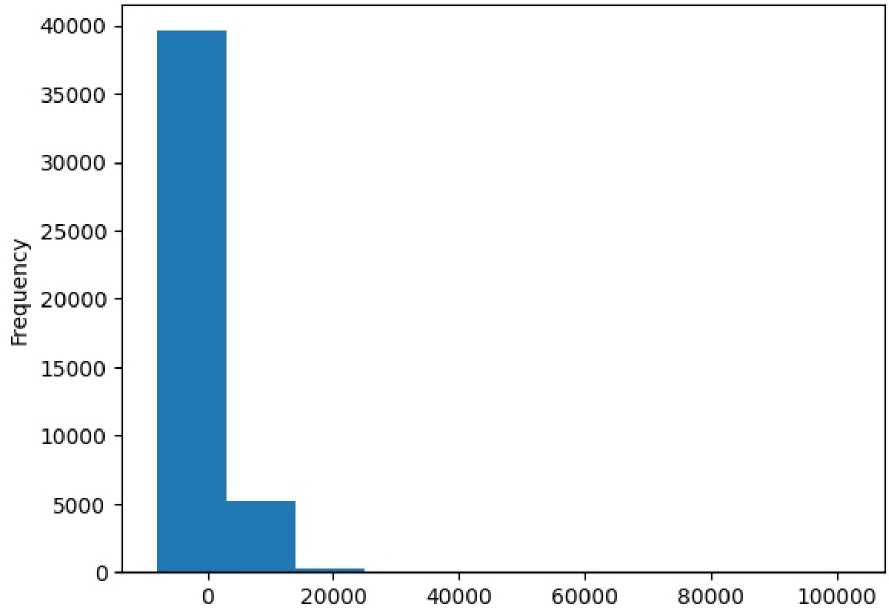


What is the distribution of average yearly balance among the clients?

In [15]:
```
# distribution of average yearly balance among the clients
# plt.figure(figsize = (10,10))
# plt.bar(df.age , df.balance)
```

```
df["balance"].plot(kind="hist")
```

Out[15]: `<Axes: ylabel='Frequency'>`

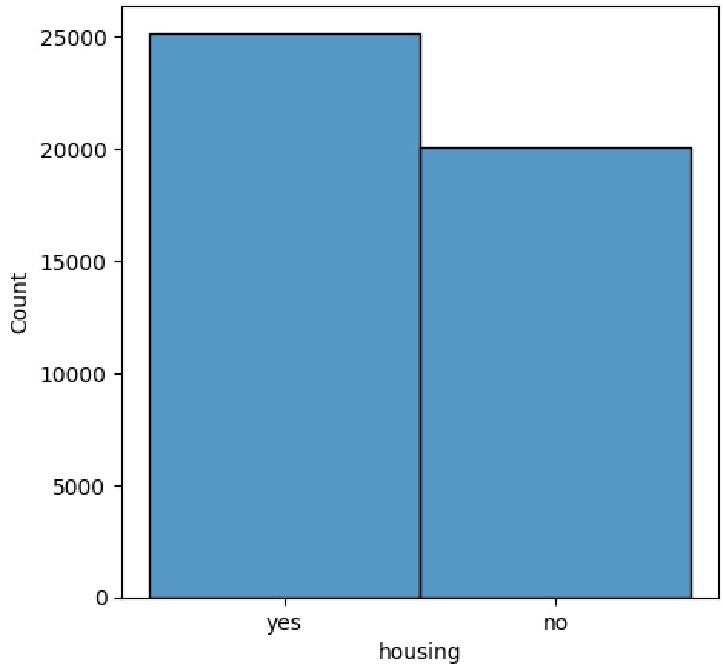


How many clients have housing loans?

In [16]:
```
housing_loans = df[df["housing"] == "yes"]
print(f"Number of clients with housing loans: {len(housing_loans)}")

plt.figure(figsize = (5,5))
sns.histplot(df.housing)
```

```
Number of clients with housing loans: 25130
```

Out[16]: `<Axes: xlabel='housing', ylabel='Count'>`

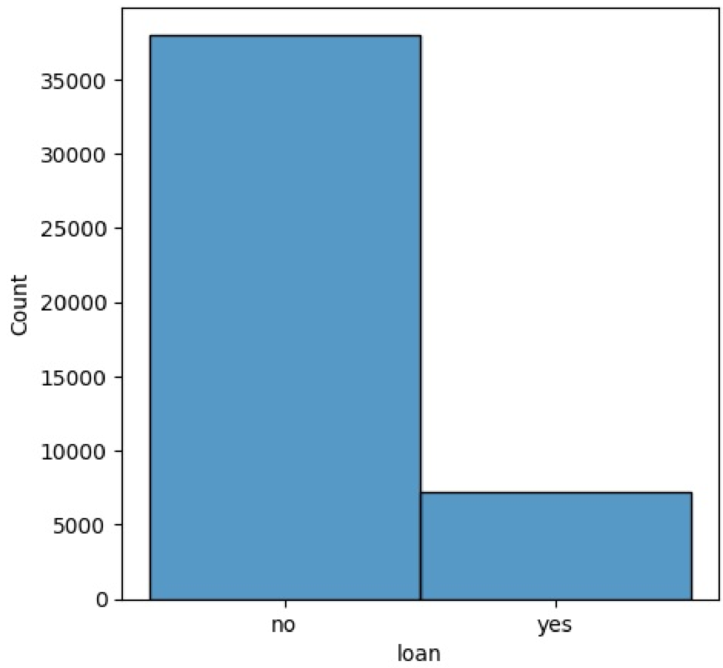


How many clients have personal loans?

In [17]:
```
personal_loans = df[df["loan"] == "yes"]
print(f"Number of clients with personal loans: {len(personal_loans)}")

plt.figure(figsize = (5,5))
sns.histplot(df.loan)
```

```
Number of clients with personal loans: 7244
```

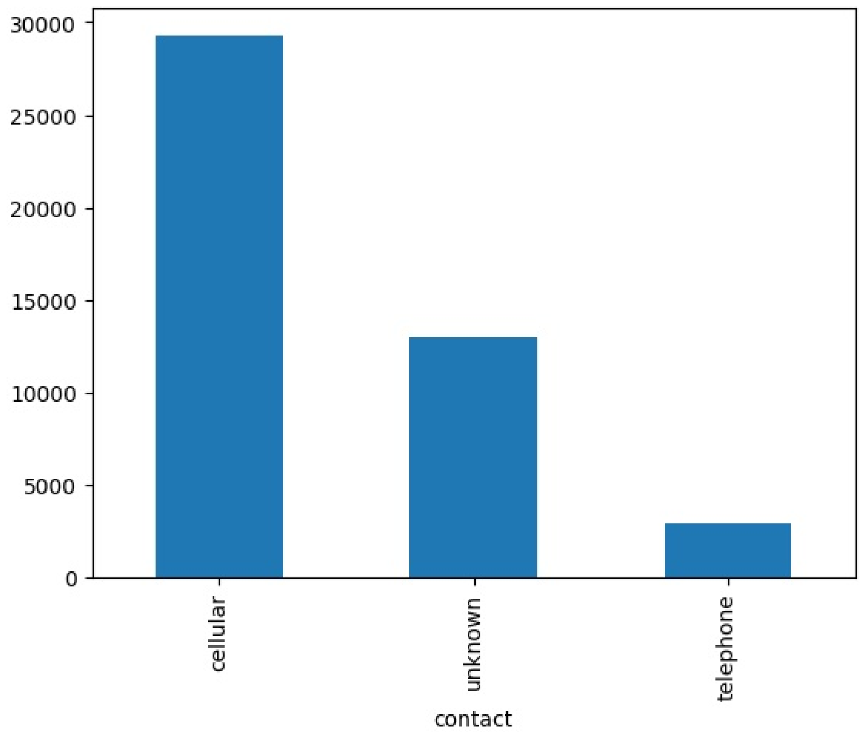
Out[17]: `<Axes: xlabel='loan', ylabel='Count'>`



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

In [18]:
```
df["contact"].value_counts().plot(kind="bar")
```

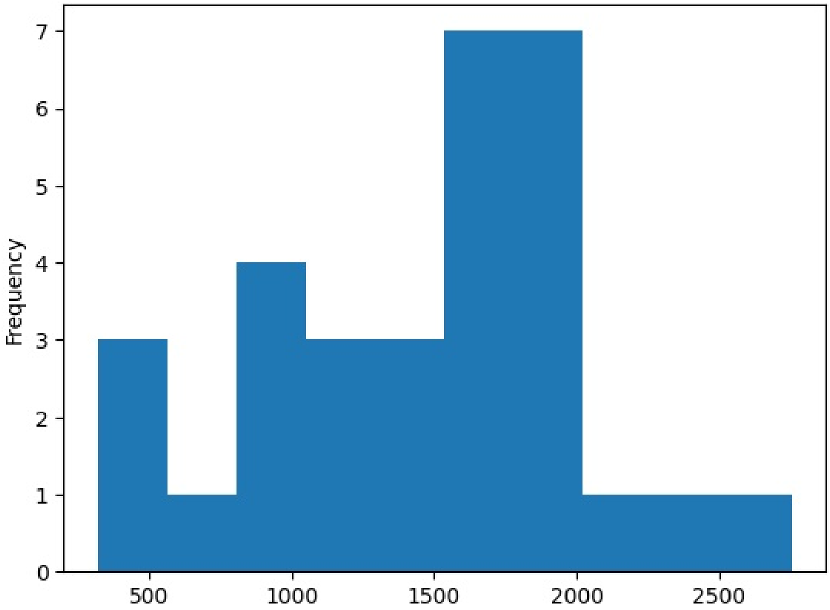
Out[18]: <Axes: xlabel='contact'>



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

In [19]:
```
df["day"].value_counts().plot(kind="hist")
```

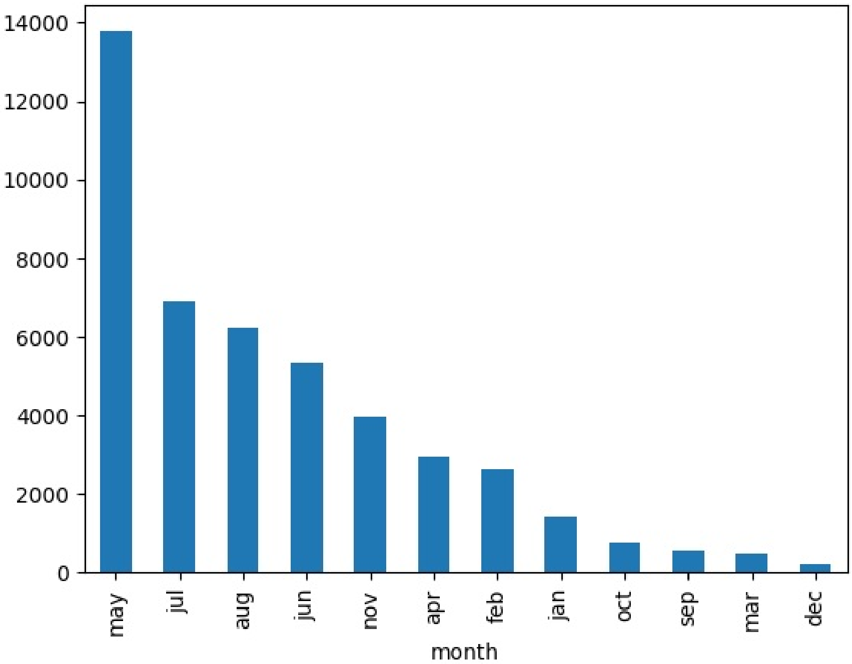
Out[19]: <Axes: ylabel='Frequency'>



How does the last contact month vary among the clients?

In [20]:
```
df["month"].value_counts().plot(kind="bar")
```

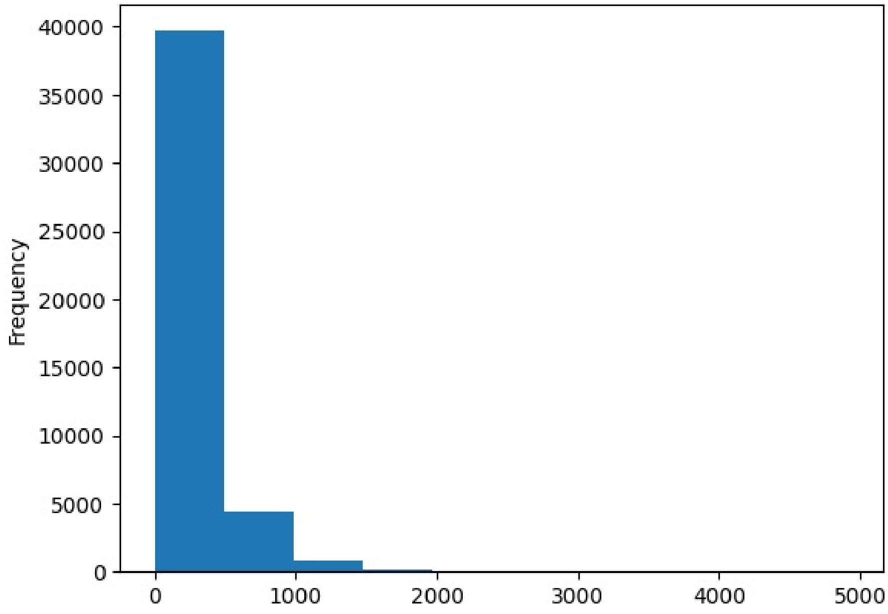
Out[20]: <Axes: xlabel='month'>



What is the distribution of the duration of the last contact?

In [21]:
```
df["duration"].plot(kind="hist")
```

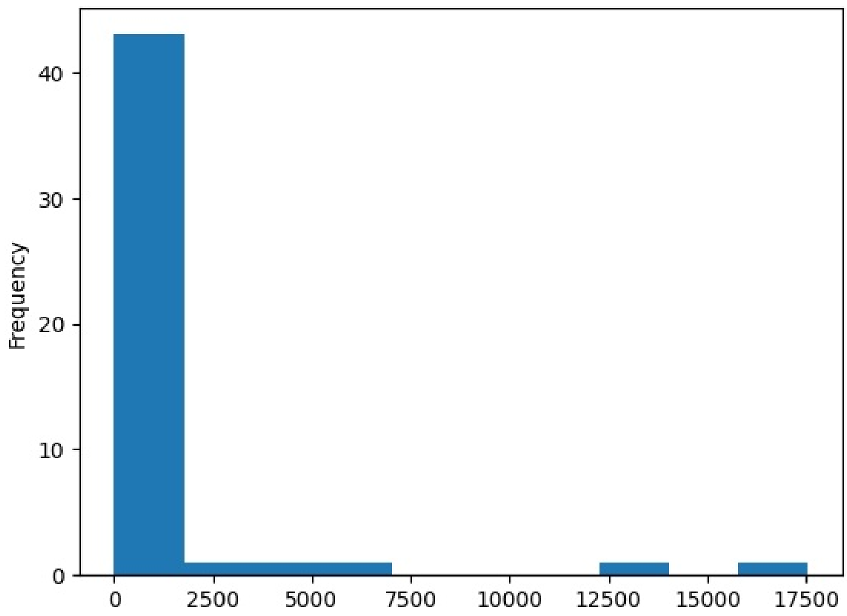
Out[21]: <Axes: ylabel='Frequency'>



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

In [22]:
```
df["campaign"].value_counts().plot(kind="hist")
```

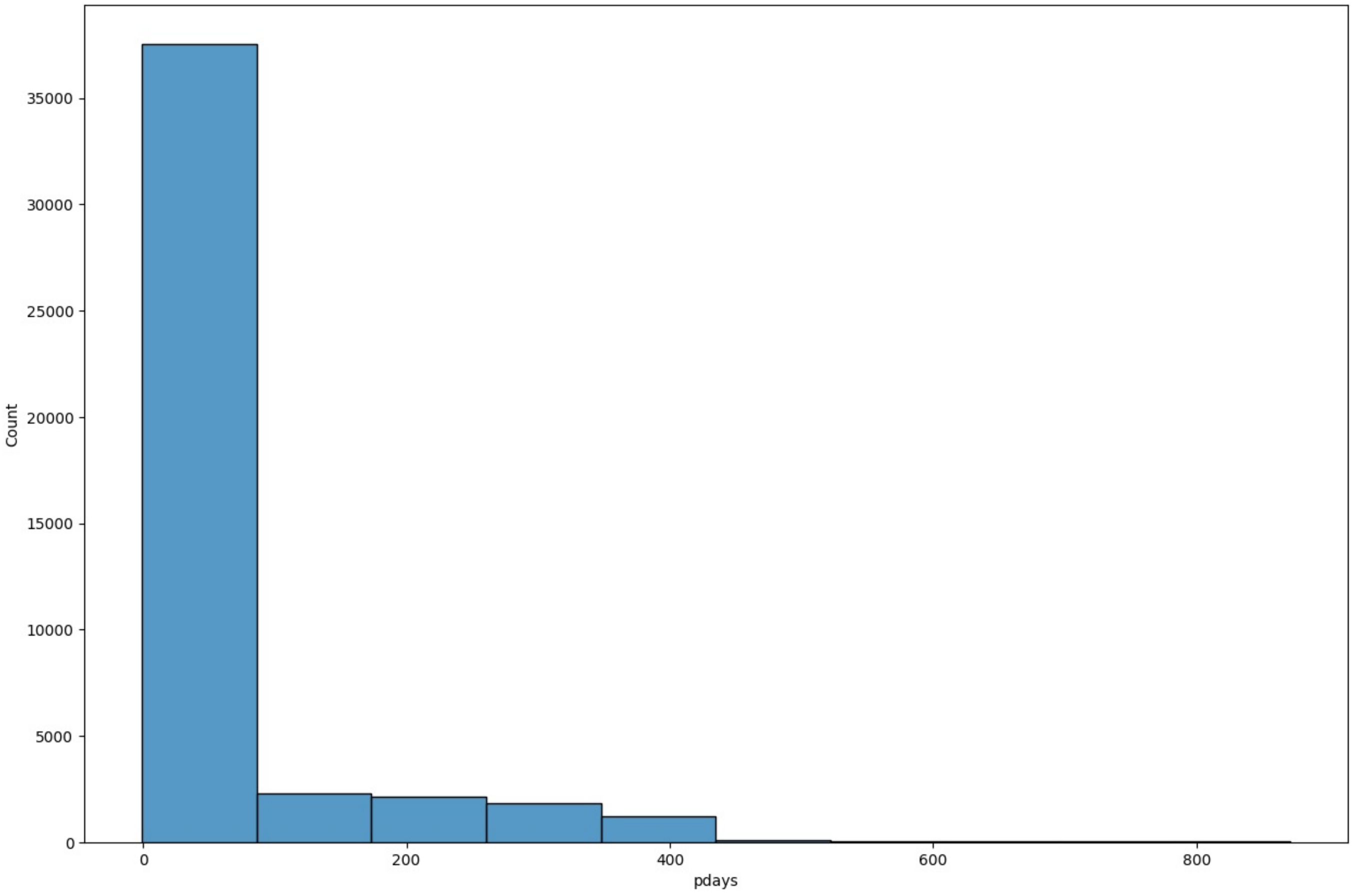
Out[22]:
```
<Axes: ylabel='Frequency'>
```



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

In [23]:
```
plt.figure(figsize = (15,10))
sns.histplot(df.pdays,bins = 10)
```

Out[23]:
```
<Axes: xlabel='pdays', ylabel='Count'>
```

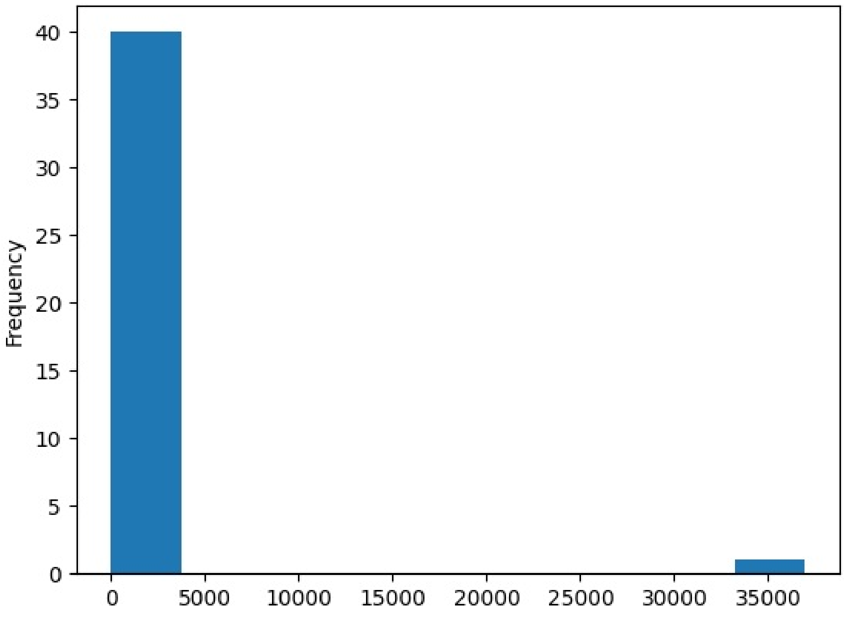


How many contacts were performed before the current campaign for each client?

In [24]:
```
print("Number of contacts were performed before the current campaign for each client :",df["previous"].value_cou
df["previous"].value_counts().plot(kind="hist")
```

```
Number of contacts were performed before the current campaign for each client : previous
0      36956
1       2772
2       2106
3       1142
4        715
5        459
6        278
7        205
8        130
9         92
10        67
11        65
12        44
13        38
15        20
14        19
17        15
16        13
19        11
20         8
23         8
18         6
22         6
24         5
27         5
21         4
29         4
25         4
30         3
38         2
37         2
26         2
28         2
51         1
275        1
58         1
32         1
40         1
55         1
35         1
41         1
Name: count, dtype: int64
```

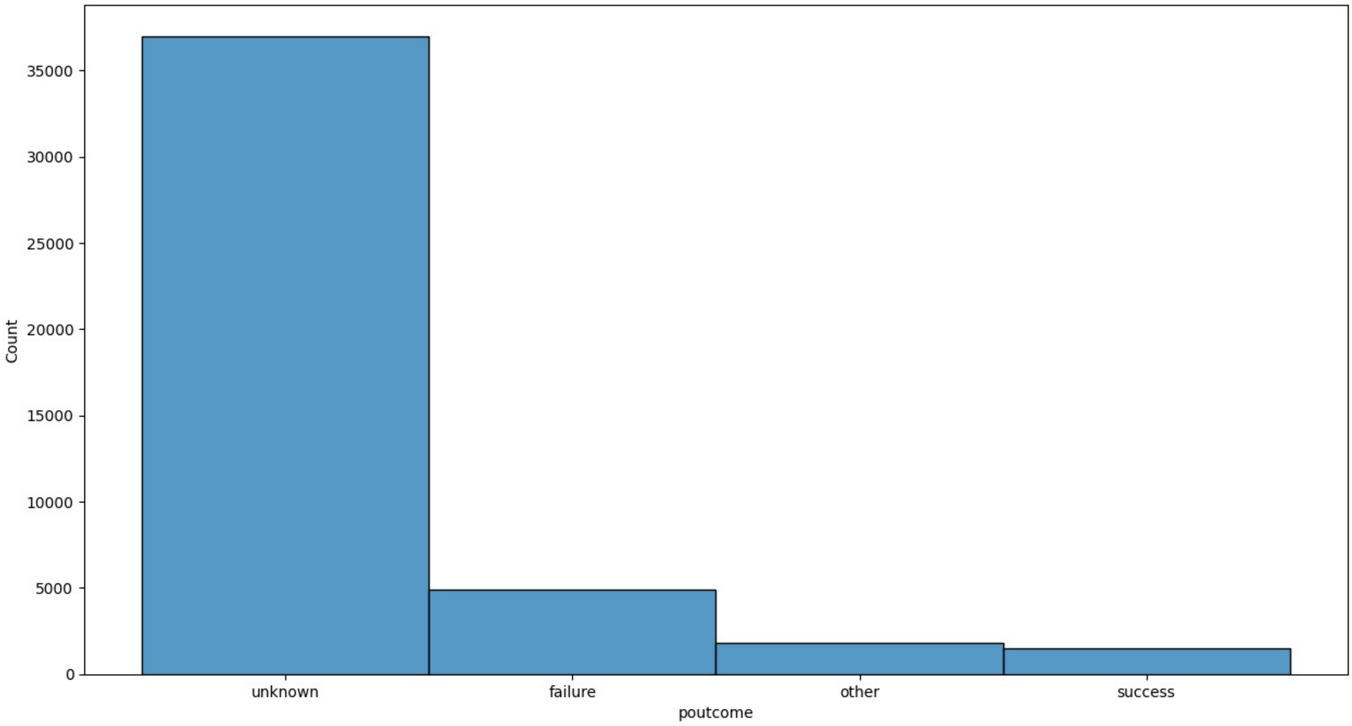
Out[24]: <Axes: ylabel='Frequency'>



What were the outcomes of the previous marketing campaigns?

In [25]:
```
plt.figure(figsize = (15,8))
sns.histplot(df.poutcome)
```

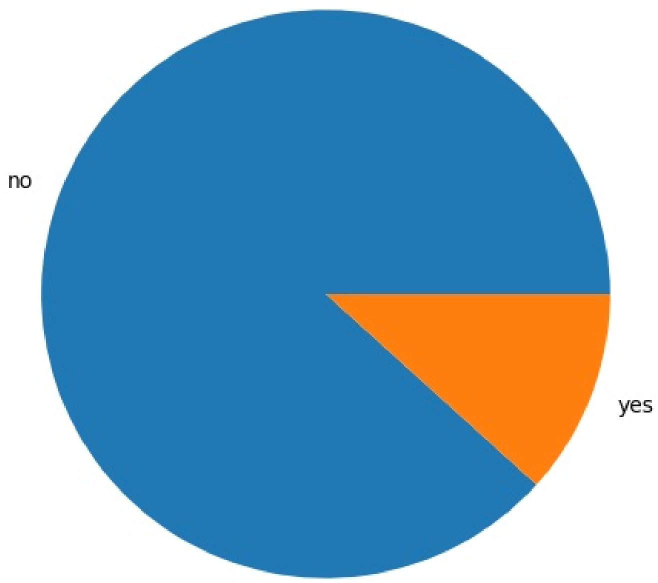
Out[25]: <Axes: xlabel='poutcome', ylabel='Count'>



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

In [26]:
```
plt.figure(figsize = (10,6))
plt.pie(df.y.value_counts() , labels = df.y.unique())
```

Out[26]:
```
([<matplotlib.patches.Wedge at 0x2453b8c0700>,
  <matplotlib.patches.Wedge at 0x2453b8c0640>],
 [Text(-1.0264226412734365, 0.39554590312789534, 'no'),
  Text(1.026422604239752, -0.3955459992285169, 'yes')])
```



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

In [28]:
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler

categorical_features = [
```

```python
    col for col in df.columns if df[col].dtype == object
]
print(categorical_features)
le = LabelEncoder()
for feature in categorical_features:
    df[feature] = le.fit_transform(df[feature])

numerical_features = [
    col for col in df.columns if df[col].dtype != object and col != "y"
]

scaler = StandardScaler()
df[numerical_features] = scaler.fit_transform(df[numerical_features])

correlation = df.corr()
print("Correlation matrix with encoded and standardized features:\n", correlation)
```

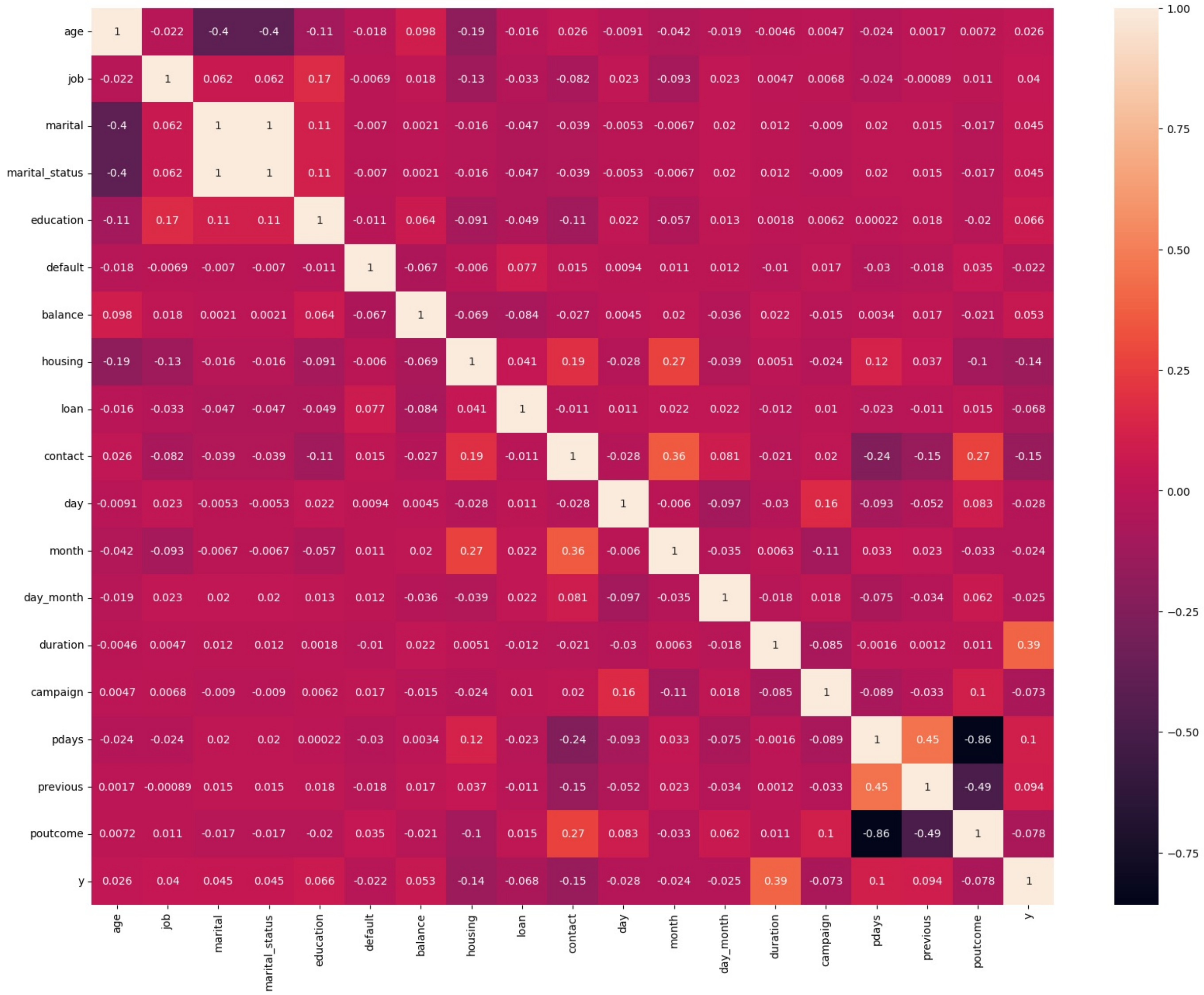
```
['job', 'marital', 'marital_status', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_month',
'poutcome', 'y']
Correlation matrix with encoded and standardized features:
                     age       job   marital  marital_status  education  \
age             1.000000 -0.021824 -0.402808       -0.402808  -0.107221
job            -0.021824  1.000000  0.062002        0.062002   0.166735
marital        -0.402808  0.062002  1.000000        1.000000   0.108850
marital_status -0.402808  0.062002  1.000000        1.000000   0.108850
education      -0.107221  0.166735  0.108850        0.108850   1.000000
default        -0.017899 -0.006854 -0.007047       -0.007047  -0.010744
balance         0.097789  0.018235  0.002068        0.002068   0.064440
housing        -0.185655 -0.125364 -0.016300       -0.016300  -0.091003
loan           -0.015732 -0.033007 -0.046960       -0.046960  -0.048652
contact         0.026080 -0.082067 -0.039079       -0.039079  -0.110908
day            -0.009095  0.022857 -0.005284       -0.005284   0.022465
month          -0.042127 -0.092852 -0.006723       -0.006723  -0.056985
day_month      -0.019074  0.022839  0.020248        0.020248   0.013064
duration       -0.004599  0.004747  0.011933        0.011933   0.001839
campaign        0.004673  0.006835 -0.009041       -0.009041   0.006165
pdays          -0.023647 -0.024451  0.019600        0.019600   0.000221
previous        0.001663 -0.000890  0.015029        0.015029   0.017653
poutcome        0.007162  0.010998 -0.017134       -0.017134  -0.019662
y               0.025648  0.040445  0.045392        0.045392   0.066270

                 default   balance   housing      loan   contact       day  \
age            -0.017899  0.097789 -0.185655 -0.015732  0.026080 -0.009095
job            -0.006854  0.018235 -0.125364 -0.033007 -0.082067  0.022857
marital        -0.007047  0.002068 -0.016300 -0.046960 -0.039079 -0.005284
marital_status -0.007047  0.002068 -0.016300 -0.046960 -0.039079 -0.005284
education      -0.010744  0.064440 -0.091003 -0.048652 -0.110908  0.022465
default         1.000000 -0.066745 -0.006008  0.077240  0.015414  0.009422
balance        -0.066745  1.000000 -0.068765 -0.084350 -0.027273  0.004504
housing        -0.006008 -0.068765  1.000000  0.041374  0.188193 -0.027991
loan            0.077240 -0.084350  0.041374  1.000000 -0.010838  0.011365
contact         0.015414 -0.027273  0.188193 -0.010838  1.000000 -0.027943
day             0.009422  0.004504 -0.027991  0.011365 -0.027943  1.000000
month           0.011468  0.019778  0.271299  0.022087  0.361017 -0.006015
day_month       0.011646 -0.036410 -0.039294  0.022089  0.081189 -0.096961
duration       -0.010023  0.021565  0.005061 -0.012417 -0.020847 -0.030204
campaign        0.016829 -0.014578 -0.023534  0.010004  0.019653  0.162483
pdays          -0.029984  0.003425  0.124112 -0.022772 -0.244830 -0.093036
previous       -0.018345  0.016693  0.036904 -0.011104 -0.147881 -0.051685
poutcome        0.034904 -0.020990 -0.099885  0.015484  0.272227  0.083444
y              -0.022451  0.052821 -0.139445 -0.068289 -0.148545 -0.028307

                   month  day_month  duration  campaign     pdays  previous  \
age            -0.042127  -0.019074 -0.004599  0.004673 -0.023647  0.001663
job            -0.092852   0.022839  0.004747  0.006835 -0.024451 -0.000890
marital        -0.006723   0.020248  0.011933 -0.009041  0.019600  0.015029
marital_status -0.006723   0.020248  0.011933 -0.009041  0.019600  0.015029
education      -0.056985   0.013064  0.001839  0.006165  0.000221  0.017653
default         0.011468   0.011646 -0.010023  0.016829 -0.029984 -0.018345
balance         0.019778  -0.036410  0.021565 -0.014578  0.003425  0.016693
housing         0.271299  -0.039294  0.005061 -0.023534  0.124112  0.036904
loan            0.022087   0.022089 -0.012417  0.010004 -0.022772 -0.011104
contact         0.361017   0.081189 -0.020847  0.019653 -0.244830 -0.147881
day            -0.006015  -0.096961 -0.030204  0.162483 -0.093036 -0.051685
month           1.000000  -0.035089  0.006327 -0.110086  0.033114  0.022888
day_month      -0.035089   1.000000 -0.018292  0.018185 -0.074761 -0.034303
duration        0.006327  -0.018292  1.000000 -0.084569 -0.001581  0.001197
campaign       -0.110086   0.018185 -0.084569  1.000000 -0.088651 -0.032932
pdays           0.033114  -0.074761 -0.001581 -0.088651  1.000000  0.454833
previous        0.022888  -0.034303  0.001197 -0.032932  0.454833  1.000000
poutcome       -0.033105   0.062041  0.010926  0.101616 -0.858289 -0.489849
y              -0.024108  -0.025387  0.394387 -0.073294  0.103699  0.093576

                poutcome         y
```

```
age            0.007162  0.025648
job            0.010998  0.040445
marital       -0.017134  0.045392
marital_status -0.017134  0.045392
education     -0.019662  0.066270
default        0.034904 -0.022451
balance       -0.020990  0.052821
housing       -0.099885 -0.139445
loan           0.015484 -0.068289
contact        0.272227 -0.148545
day            0.083444 -0.028307
month         -0.033105 -0.024108
day_month      0.062041 -0.025387
duration       0.010926  0.394387
campaign       0.101616 -0.073294
pdays         -0.858289  0.103699
previous      -0.489849  0.093576
poutcome       1.000000 -0.077973
y             -0.077973  1.000000
```

In [34]:

```python
plt.figure(figsize = (20,15))
sns.heatmap(correlation , annot = True)
```

Out[34]: <Axes: >



In [36]:

```python
from sklearn.feature_selection import f_classif

def analyze_correlations(data, target_variable="y", correlation_threshold=0.3):
  categorical_features = [
      col for col in data.columns if data[col].dtype == object and col != target_variable
  ]

  le = LabelEncoder()
  for feature in categorical_features:
      data[feature] = le.fit_transform(data[feature])

  numerical_features = [
      col for col in data.columns if data[col].dtype != object and col != target_variable
  ]
```

```python
    scaler = StandardScaler()
    data[numerical_features] = scaler.fit_transform(data[numerical_features])

    X = data.drop(target_variable, axis=1)
    y = data[target_variable]
    f_scores, p_values = f_classif(X, y)
    correlated_features = X.columns[f_scores > correlation_threshold]

    print("Correlated features (with correlation coefficient above", correlation_threshold, "):")
    for feature, score in zip(correlated_features, f_scores):
        print(f"{feature}: {score:.2f}")

    return data[correlated_features]

data = pd.read_csv("banking_data.csv")
correlated_data = analyze_correlations(data.copy())
```

```
Correlated features (with correlation coefficient above 0.3 ):
age: 29.76
job: 74.08
marital: 93.35
marital_status: 93.35
education: 199.44
default: 22.80
balance: 126.50
housing: 896.61
loan: 211.84
contact: 1020.19
day: 36.26
month: 26.29
day_month: 29.16
duration: 8327.97
campaign: 244.20
pdays: 491.50
previous: 399.41
poutcome: 276.57
```

In [ ]: