In [1]:
```
#Vraag 1

# Piet was grooter in vergelijking met de standaart afwijking
# als je de berkening doet is piet verder af van de standaart afwijking dan
# zijn zoon
```

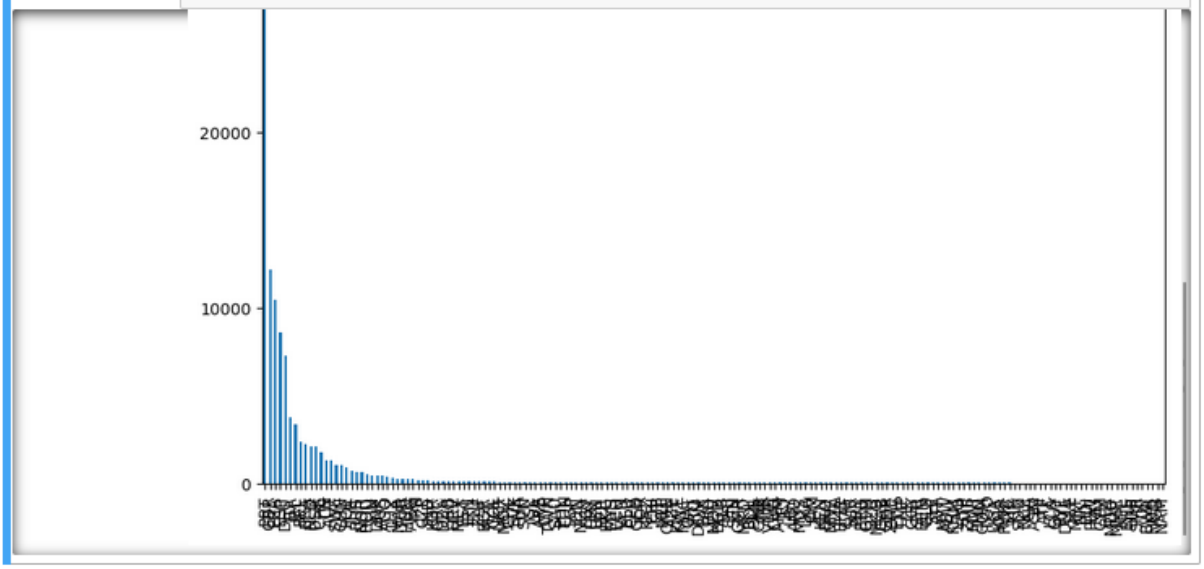
In [16]:
```
#Vraag 2
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
hotels = pd.read_csv('hotels.csv')
hotels.info()
hotels['country']
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
```

In [108]:
```
#Vraag 2
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
hotels = pd.read_csv('hotels.csv')
hotels['country']
```

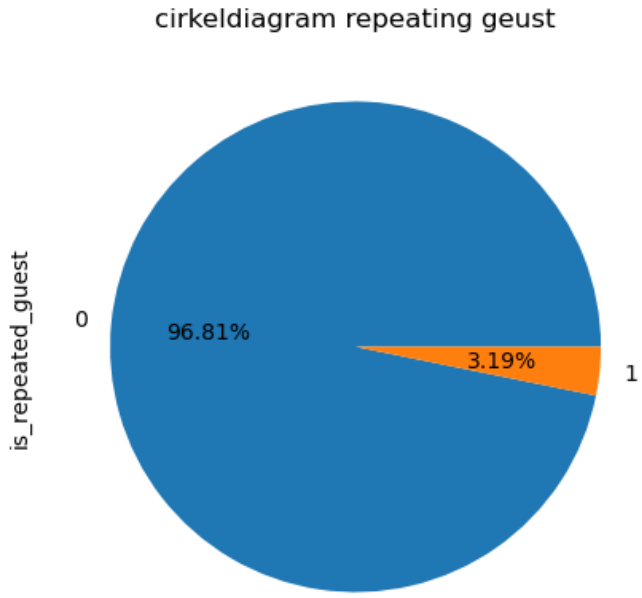
Out[108]:
```
0         PRT
1         PRT
2         GBR
3         GBR
4         GBR
          ...
119385    BEL
119386    FRA
119387    DEU
119388    GBR
119389    DEU
Name: country, Length: 119390, dtype: object
```

In [30]:
```
#Vraag 3
hotels_landen = hotels['country']
data_bar_hotels_plot = hotels_landen.value_counts()
bar_plot = data_bar_hotels_plot.plot.bar(figsize=(10,10), title = 'staafdiagr
```



In [107]:
```
#Vraag 4
reservatie = hotels['is_repeated_guest']
data_pie_plot = reservatie.value_counts()
print(data_pie_plot)
pie_plot = data_pie_plot.plot.pie(figsize=(5,5), autopct='%3.2f%%', title ='c
```

```
0    115580
1      3810
Name: is_repeated_guest, dtype: int64
```



In [64]:
```
#Vraag 5
pd.crosstab(hotels.lead_time, hotels.hotel, margins = True, normalize = True)
# het verband van de lead time tussen niet geanulleerd en geanulleerd is
# dat je kan zien welke moment het meest populaire zijn en wanneer mensen
# het vaakst gaan afzeggen
```

Out[64]:

hotel	City Hotel	Resort Hotel	All
lead_time			
0	0.026041	0.027104	0.053145
1	0.015621	0.013360	0.028981
2	0.009465	0.007865	0.017330
3	0.008560	0.006650	0.015211
4	0.008811	0.005553	0.014365
...	...	...	...
626	0.000251	0.000000	0.000251
629	0.000142	0.000000	0.000142
709	0.000000	0.000008	0.000008
737	0.000000	0.000008	0.000008
All	0.664461	0.335539	1.000000

480 rows × 3 columns

In [62]:
```
#vraag 6
hotels.describe(include = 'all')
pd.crosstab(hotels.lead_time, hotels.hotel, margins = True, normalize = True)
pd.crosstab(hotels.previous_cancellations, hotels.hotel, margins = True, norm
# er lijtk ocanscelattions te komen bij bijde en dan
# traag terug stijgen das er is een verband waarvoor de reservatie is gemaakt
```

Out[62]:

hotel	City Hotel	Resort Hotel	All
previous_cancellations			
0	0.619323	0.326367	0.945691
1	0.043178	0.007505	0.050683
2	0.000603	0.000369	0.000972
3	0.000427	0.000117	0.000544
4	0.000209	0.000050	0.000260
5	0.000134	0.000025	0.000159
6	0.000184	0.000000	0.000184
11	0.000293	0.000000	0.000293
13	0.000101	0.000000	0.000101
14	0.000000	0.000117	0.000117
19	0.000000	0.000159	0.000159
21	0.000008	0.000000	0.000008
24	0.000000	0.000402	0.000402
25	0.000000	0.000209	0.000209
26	0.000000	0.000218	0.000218
All	0.664461	0.335539	1.000000

In [69]:
```
# Vraag 7
import sklearn
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
hotels_df = pd.read_csv('hotels.csv')
hotels_df.head(10)
```

Out[69]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number
0	Resort Hotel	0	342	2015	July	27
1	Resort Hotel	0	737	2015	July	27
2	Resort Hotel	0	7	2015	July	27
3	Resort Hotel	0	13	2015	July	27
4	Resort Hotel	0	14	2015	July	27
5	Resort Hotel	0	14	2015	July	27
6	Resort Hotel	0	0	2015	July	27
7	Resort Hotel	0	9	2015	July	27
8	Resort Hotel	1	85	2015	July	27
9	Resort Hotel	1	75	2015	July	27

In [71]:
```
#Vraag 8
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
hotels_df['hotel'] = label_encoder.fit_transform(hotels_df['hotel'])
hotels_df.head(10)
```

Out[71]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number
0	1	0	342	2015	July	27
1	1	0	737	2015	July	27
2	1	0	7	2015	July	27
3	1	0	13	2015	July	27
4	1	0	14	2015	July	27
5	1	0	14	2015	July	27
6	1	0	0	2015	July	27
7	1	0	9	2015	July	27
8	1	1	85	2015	July	27
9	1	1	75	2015	July	27

In [82]:
```
#Vraag 9
#hotels_df.drop(['arrival_date_year', 'arrival_date_month', 'arrival_date_wee
#hotels_df.drop(['previous_bookings_not_canceled', 'reserved_room_type', 'ass
hotels_df.to_csv('hotels_train_df_klaar.csv', index = False)
hotels_ML = pd.read_csv('hotels_train_df_klaar.csv')
```

In [88]:
```
#Vraag 10
hotels_ML.head(5)
```

Out[88]:

	hotel	is_canceled	lead_time	adults	is_repeated_guest	previous_cancellations	required_car_pa
0	1	0	342	2	0	0	
1	1	0	737	2	0	0	
2	1	0	7	1	0	0	
3	1	0	13	1	0	0	
4	1	0	14	2	0	0	

In [92]:
```
#Vraag 11
from sklearn.model_selection import train_test_split
X = hotels_ML.drop('hotel', axis=1)
y = hotels_ML['hotel']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
print(X_train.head())
```

```
       is_canceled  lead_time  adults  is_repeated_guest  \
16267            0         68       2                  0
31480            0         69       2                  0
87274            0        122       2                  0
550              0         41       2                  0
34933            0         14       2                  0

       previous_cancellations  required_car_parking_spaces  \
16267                       0                            1
31480                       0                            1
87274                       0                            0
550                         0                            0
34933                       0                            0

       total_of_special_requests
16267                          0
31480                          3
87274                          0
550                            1
34933                          0
```

In [98]:
```
#Vraag 12
from sklearn.linear_model import LogisticRegression
logistic_model = LogisticRegression(penalty= 'l2', solver='liblinear').fit(X_
y_pred = logistic_model.predict(X_test)
pred_results = pd.DataFrame({'y_test': y_test, 'y_pred': y_pred})
pred_results.head(10)
```

Out[98]:

	y_test	y_pred
58845	0	0
53973	0	0
5386	1	0
113141	0	0
85602	0	0
114015	0	0
43445	0	0
35067	1	0
102724	0	0
111847	0	0

In [93]:
```
#Vraag 12
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
print('accuracy score', acc)
print('precision score', prec)
print('recall score', recall)
```

```
accuracy score 0.6430581613508443
precision score 0.3405720338983051
recall score 0.06396736967767608
```

In [101]:
```
#Vraag 13
y_test.head(20)
```

Out[101]:
```
16267     1
31480     1
87274     0
550       1
34933     1
41308     0
14335     1
66260     0
111082    0
40583     0
84576     0
18286     1
4684      1
12443     1
32788     1
62790     0
46980     0
31047     1
76934     0
103321    0
Name: hotel, dtype: int64
```

In [104]:
```
#Vraag 13
pred_results.head(20)
```

Out[104]:

	y_test	y_pred
58845	0	0
53973	0	0
5386	1	0
113141	0	0
85602	0	0
114015	0	0
43445	0	0
35067	1	0
102724	0	0
111847	0	0
117096	0	0
83544	0	0
61899	0	0
113603	0	0
50093	0	0
91909	0	0
36885	1	0
103871	0	0
70134	0	0
97904	0	0