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

In [2]:
```
df = pd.read_csv('Listings.csv', encoding='ISO-8859-1',low_memory=False)
```

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

Out[3]:

	listing_id	name	host_id	host_since	host_location	host_response_time	host_response_rate	host_acceptance_rate	host_is_superhost	host_total_listings_count	...	minimum_nights	maximum_nights	review_scores_rating	review_scores_accuracy
0	281420	Beautiful Flat in le Village Montmartre, Paris	1466919	2011-12-03	Paris, Île-de-France, France	NaN	NaN	NaN	f	1.0	...	2	1125	100.0	10.0
1	3705183	39 mÂ² Paris (Sacre CÃ¢ur)	10328771	2013-11-29	Paris, Île-de-France, France	NaN	NaN	NaN	f	1.0	...	2	1125	100.0	10.0
2	4082273	Lovely apartment with Terrace, 60m2	19252768	2014-01-31	Paris, Île-de-France, France	NaN	NaN	NaN	f	1.0	...	2	1125	100.0	10.0
3	4797344	Cosy studio (close to Eiffel tower)	10868311	2013-12-17	Paris, Île-de-France, France	NaN	NaN	NaN	f	1.0	...	2	1125	100.0	10.0
4	4823469	Close to Eiffel Tower - Beautiful flat : 2 rooms	24837558	2014-12-14	Paris, Île-de-France, France	NaN	NaN	NaN	f	1.0	...	2	1125	100.0	10.0

5 rows × 33 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 279712 entries, 0 to 279711
Data columns (total 33 columns):
 #   Column                        Non-Null Count   Dtype  
---  ------                        --------------   -----  
 0   listing_id                    279712 non-null  int64  
 1   name                          279537 non-null  object 
 2   host_id                       279712 non-null  int64  
 3   host_since                    279547 non-null  object 
 4   host_location                 278872 non-null  object 
 5   host_response_time            150530 non-null  object 
 6   host_response_rate            150530 non-null  float64
 7   host_acceptance_rate          166625 non-null  float64
 8   host_is_superhost             279547 non-null  object 
 9   host_total_listings_count     279547 non-null  float64
 10  host_has_profile_pic          279547 non-null  object 
 11  host_identity_verified        279547 non-null  object 
 12  neighbourhood                 187897 non-null  object 
 13  district                      97012 non-null   object 
 14  city                          279712 non-null  object 
 15  latitude                      279712 non-null  float64
 16  longitude                     279712 non-null  float64
 17  property_type                 279712 non-null  object 
 18  room_type                     279712 non-null  object 
 19  accommodates                  279712 non-null  int64  
 20  bedrooms                      250277 non-null  float64
 21  amenities                     279712 non-null  object 
 22  price                         279712 non-null  int64  
 23  minimum_nights                279712 non-null  int64  
 24  maximum_nights                279712 non-null  int64  
 25  review_scores_rating          188367 non-null  float64
 26  review_scores_accuracy        188367 non-null  float64
 27  review_scores_cleanliness     188367 non-null  float64
 28  review_scores_checkin         187961 non-null  float64
 29  review_scores_communication   188025 non-null  float64
 30  review_scores_location        187937 non-null  float64
 31  review_scores_value           187936 non-null  float64
 32  instant_bookable              279712 non-null  object 
dtypes: float64(13), int64(6), object(14)
memory usage: 70.4+ MB
```

In [5]:
```
df["host_since"] = pd.to_datetime(df["host_since"])
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 279712 entries, 0 to 279711
Data columns (total 33 columns):
 #   Column                        Non-Null Count   Dtype         
---  ------                        --------------   -----         
 0   listing_id                    279712 non-null  int64         
 1   name                          279537 non-null  object        
 2   host_id                       279712 non-null  int64         
 3   host_since                    279547 non-null  datetime64[ns]
 4   host_location                 278872 non-null  object        
 5   host_response_time            150530 non-null  object        
 6   host_response_rate            150530 non-null  float64       
 7   host_acceptance_rate          166625 non-null  float64       
 8   host_is_superhost             279547 non-null  object        
 9   host_total_listings_count     279547 non-null  float64       
 10  host_has_profile_pic          279547 non-null  object        
 11  host_identity_verified        279547 non-null  object        
 12  neighbourhood                 187897 non-null  object        
 13  district                      97012 non-null   object        
 14  city                          279712 non-null  object        
 15  latitude                      279712 non-null  float64       
 16  longitude                     279712 non-null  float64       
 17  property_type                 279712 non-null  object        
 18  room_type                     279712 non-null  object        
 19  accommodates                  279712 non-null  int64         
 20  bedrooms                      250277 non-null  float64       
 21  amenities                     279712 non-null  object        
 22  price                         279712 non-null  int64         
 23  minimum_nights                279712 non-null  int64         
 24  maximum_nights                279712 non-null  int64         
 25  review_scores_rating          188367 non-null  float64       
 26  review_scores_accuracy        188367 non-null  float64       
 27  review_scores_cleanliness     188367 non-null  float64       
 28  review_scores_checkin         187961 non-null  float64       
 29  review_scores_communication   188025 non-null  float64       
 30  review_scores_location        187937 non-null  float64       
 31  review_scores_value           187936 non-null  float64       
 32  instant_bookable              279712 non-null  object        
dtypes: datetime64[ns](1), float64(13), int64(6), object(13)
memory usage: 70.4+ MB
```

In [7]:
```
paris_listings = df.query("city == 'Paris'").loc[:,['host_since","neighbourhood","accommodates","city","price"]]
paris_listings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 64690 entries, 0 to 279711
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype         
---  ------          --------------  -----         
 0   host_since      64657 non-null  datetime64[ns]
 1   neighbourhood   64690 non-null  object        
 2   accomodates     64690 non-null  int64         
 3   city            64690 non-null  object        
 4   price           64690 non-null  int64         
dtypes: datetime64[ns](1), int64(2), object(2)
memory usage: 3.0+ MB
```

In [8]:
```
# checking missing values

paris_listings.isna().sum()
```

Out[8]:
```
host_since      33
neighbourhood    0
accomodates      0
city             0
price            0
dtype: int64
```

In [9]:
```
paris_listings.describe()
```

Out[9]:

	host_since	accommodates	price
count	64657	64690.000000	64690.000000
mean	2015-11-01 11:06:05.528687584	3.037997	113.096445
min	2008-06-30 00:00:00	0.000000	0.000000
25%	2014-03-09 00:00:00	2.000000	59.000000
50%	2015-07-07 00:00:00	2.000000	80.000000
75%	2017-05-29 00:00:00	4.000000	120.000000
max	2021-02-07 00:00:00	16.000000	12000.000000
std	NaN	1.588766	214.433668

In [10]:
```
paris_listings.columns
```

Out[10]:
```
Index(['host_since', 'neighbourhood', 'accommodates', 'city', 'price'], dtype='object')
```

In [11]:
```
paris_listings_neighbour =(
    paris_listings.groupby("neighbourhood")
    .agg({"price": "mean"})
    .sort_values("price")
)
paris_listings_neighbour.head()
```

Out[11]:

	price
neighbourhood	
Ménilmontant	74.942257
Buttes-Chaumont	82.690162
Buttes-Montmartre	87.209479
Reuilly	89.058402
Popincourt	90.559459

In [12]:
```
paris_listings_accomodates = (
    paris_listings.query("neighbourhood == 'Elysee'").groupby("accommodates").agg({"price": "mean"}).sort_values("price")
)
paris_listings_accomodates.tail()
```

Out[12]:

	price
accommodates	
12	529.625
16	800.000
11	805.000
13	842.500
14	971.000

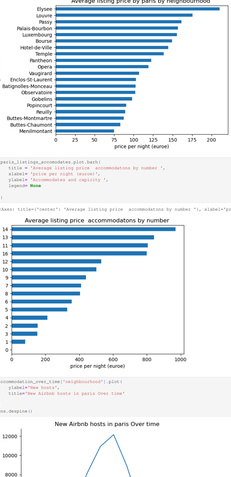
In [13]:
```
accomodation_over_time = (
    paris_listings.set_index("host_since")
    .resample("1Y")
    .agg(
        {"neighbourhood" : "count",
         "price" : "mean"
        })
)
accomodation_over_time.head()
```

Out[13]:

	neighbourhood	price
host_since		
2008-12-31	4	77.750000
2009-12-31	106	159.641509
2010-12-31	416	125.031250
2011-12-31	1339	124.828230
2012-12-31	4592	111.578615

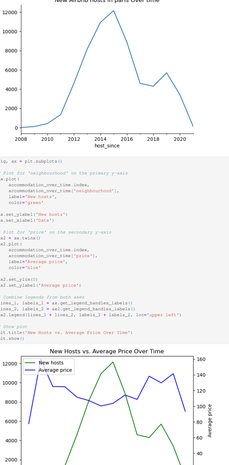
In [14]:
```
paris_listings_neighbour.plot.barh(
    title ='Average listing price by paris by neighbourhood ',
    xlabel= 'price per night (euros)',
    ylabel= 'Neighbourhood',
    legend= None
)
```

Out[14]:
```
<Axes: title={'center': 'Average listing price by paris by neighbourhood '}, xlabel='price per night (euros)', ylabel='Neighbourhood'>
```



In [15]:
```
paris_listings_accomodates.plot.barh(
    title ='Average listing price  accomodations by number ',
    xlabel= 'price per night (euros)',
    ylabel= 'Accomodates and capicity ',
    legend= None
)
```

Out[15]:
```
<Axes: title={'center': 'Average listing price  accomodations by number '}, xlabel='price per night (euros)', ylabel='Accomodates and capicity '>
```



In [16]:
```
accomodation_over_time['neighbourhood'].plot(
    ylabel="New hosts",
    title='New Airbnb hosts in paris Over Time'
)
sns.despine()
```

In [17]:
```
fig, ax = plt.subplots()

# Plot for "neighbourhood" on the primary y-axis
ax.plot(
    accomodation_over_time.index,
    accomodation_over_time['neighbourhood'],
    label='New hosts',
    color='green'
)
ax.set_ylabel('New hosts')
ax.set_xlabel('Date')

# Plot for 'price' on the secondary y-axis
ax2 = ax.twinx()
ax2.plot(
    accomodation_over_time.index,
    accomodation_over_time['price'],
    label='Average price',
    color='blue'
)
ax2.set_ylim(0)
ax2.set_ylabel('Average price')

# Combine legends from both axes
lines_1, labels_1 = ax.get_legend_handles_labels()
lines_2, labels_2 = ax2.get_legend_handles_labels()
ax2.legend(lines_1 + lines_2, labels_1 + labels_2, loc='upper left')

# Show plot
plt.title('New Hosts vs. Average Price Over Time')
plt.show()
```


In [18]:
```
# Group by "neighbourhood" and calculate the average price
neighbourhood_avg_price = paris_listings.groupby('neighbourhood').agg({'price': 'mean'})

# Find the neighborhood with the highest average price
highest_avg_price = neighbourhood_avg_price['price'].idxmax()
highest_price_value = neighbourhood_avg_price['price'].max()
```

```
print(f"The neighborhood in Paris with the highest average Airbnb listing price is '{highest_avg_price}' with an average price of {highest_price_value:.2f}.")
```

```
The neighborhood in Paris with the highest average Airbnb listing price is "Élysee" with an average price of 210.54.
```