In [5]:
```python
import pandas as pd

Listings = pd.read_csv(
    "Listings.csv",
    encoding="ISO-8859-1",
    low_memory=False,
    parse_dates=["host_since"]
)
```

In [6]:
```python
Listings.head()
```

Out[6]:

	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
0	281420	Beautiful Flat in le Village Montmartre, Paris	1466919	2011-12-03	Paris, Ile-de-France, France	NaN	NaN	NaN	f	1.0	...	2
1	3705183	39 mÃ□Â² Paris (Sacre CÃ□â□□ur)	10328771	2013-11-29	Paris, Ile-de-France, France	NaN	NaN	NaN	f	1.0	...	2
2	4082273	Lovely apartment with Terrace, 60m2	19252768	2014-07-31	Paris, Ile-de-France, France	NaN	NaN	NaN	f	1.0	...	2
3	4797344	Cosy studio (close to Eiffel tower)	10668311	2013-12-17	Paris, Ile-de-France, France	NaN	NaN	NaN	f	1.0	...	2
4	4823489	Close to Eiffel Tower - Beautiful flat : 2 rooms	24837558	2014-12-14	Paris, Ile-de-France, France	NaN	NaN	NaN	f	1.0	...	2

5 rows × 33 columns

In [7]:
```python
Listings.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                         279539 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           150930 non-null  object        
 6   host_response_rate           150930 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                279712 non-null  object        
 13  district                     37012 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         188307 non-null  float64       
 26  review_scores_accuracy       187999 non-null  float64       
 27  review_scores_cleanliness    188047 non-null  float64       
 28  review_scores_checkin        187941 non-null  float64       
 29  review_scores_communication  188025 non-null  float64       
 30  review_scores_location       187937 non-null  float64       
 31  review_scores_value          187927 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 [9]:

```
paris_Listings = (
    Listings
    .query("city == 'Paris'")
    .loc[:, ["host_since", "neighbourhood", "city", "accommodates", "price"]]
)

paris_Listings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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   city           64690 non-null  object
 3   accommodates   64690 non-null  int64
 4   price          64690 non-null  int64
dtypes: datetime64[ns](1), int64(2), object(2)
memory usage: 3.0+ MB
```

In [10]:

```
paris_Listings.describe()
```

Out[10]:

	accommodates	price
count	64690.000000	64690.000000
mean	3.037997	113.096445
std	1.588766	214.433668
min	0.000000	0.000000
25%	2.000000	59.000000
50%	2.000000	80.000000
75%	4.000000	120.000000
max	16.000000	12000.000000

In [13]:

```
paris_Listings.query("price == 0 and accommodates == 0").count()
```

Out[13]:

```
host_since       54
neighbourhood    54
city             54
accommodates     54
price            54
dtype: int64
```

In [17]:

```python
paris_Listings_neighbourhood = (paris_Listings.groupby("neighbourhood").agg({"price": "mean"}).sort_values("price"))

paris_Listings_neighbourhood.tail()
```

Out[17]:

	price
neighbourhood	
Luxembourg	155.638639
Palais-Bourbon	156.856578
Passy	161.144635
Louvre	175.379972
Elysee	210.536765

In [21]:

```python
paris_Listings_accommodates = (
     paris_Listings
     .query("neighbourhood == 'Elysee'")
    .groupby("accommodates")
    .agg({"price": "mean"})
    .sort_values("price")
)

paris_Listings_accommodates.head()
```

Out[21]:

	price
accommodates	
0	0.000000
1	79.522222
3	152.828767
2	155.103352
4	212.096070

In [22]:

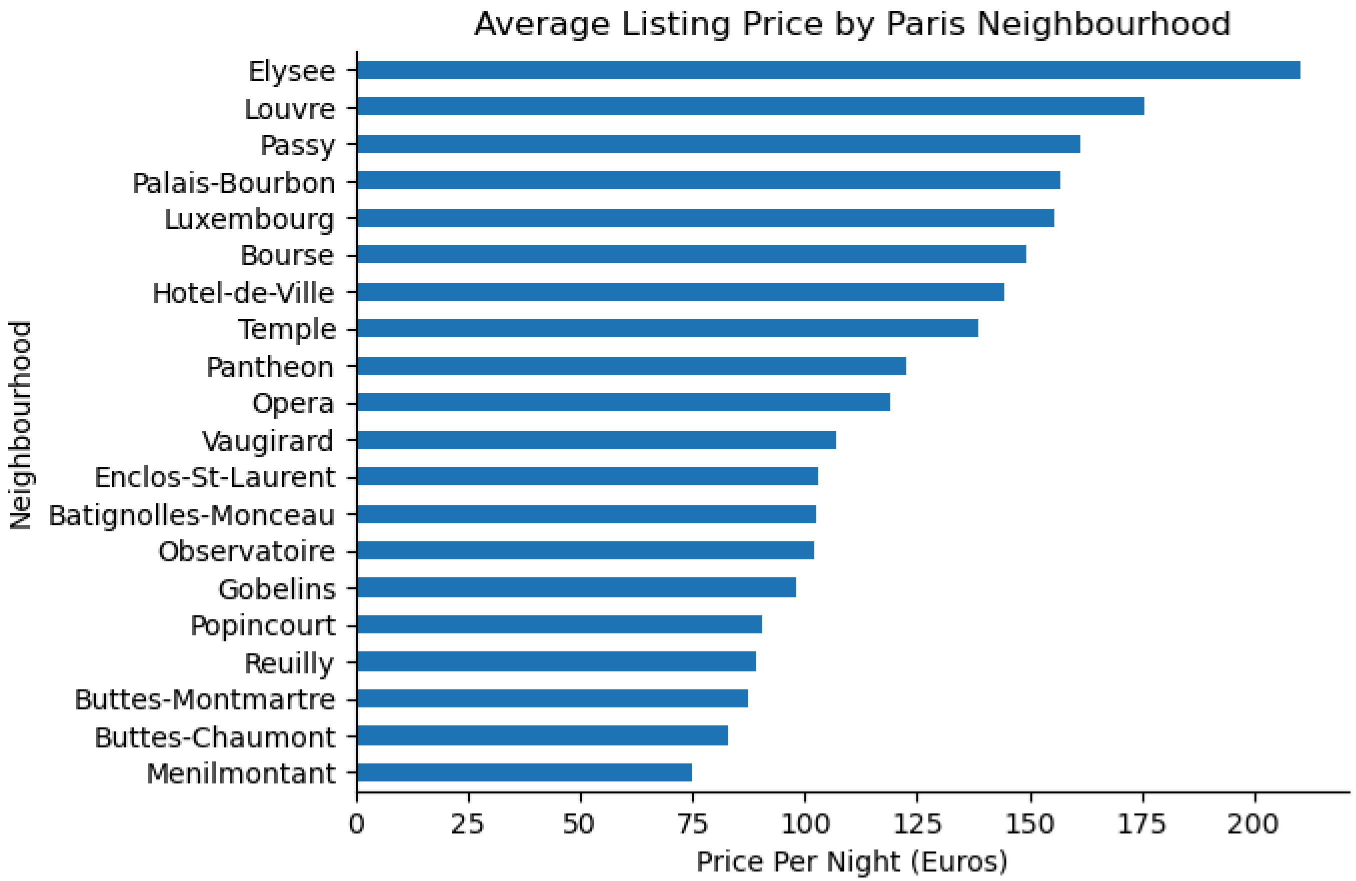
```python
paris_Listings_over_time = (
    paris_Listings
    .set_index("host_since")
    .resample("Y")
    .agg({
        "neighbourhood": "count",
        "price": "mean"
    })
)

paris_Listings_over_time.head()
```

Out[22]:

	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

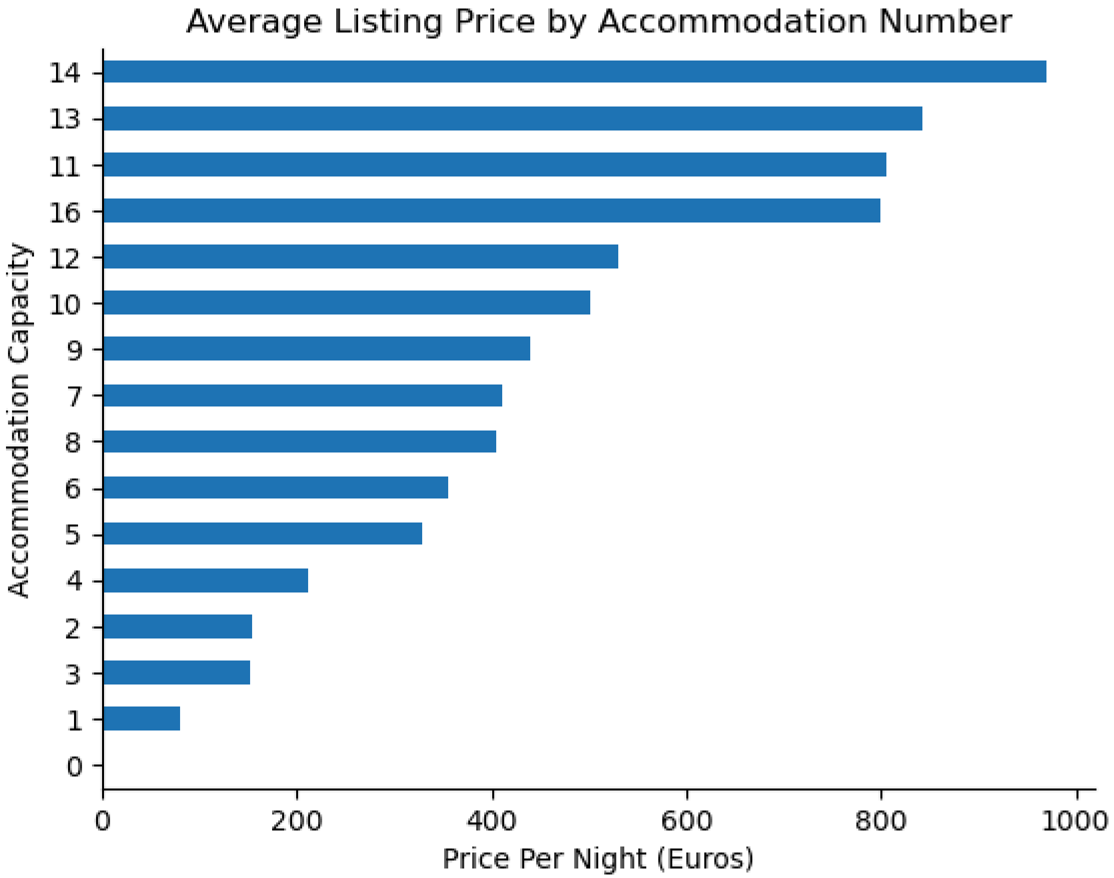
In [24]:

```
import seaborn as sns

(paris_Listings_neighbourhood
 .plot
 .barh(
     title="Average Listing Price by Paris Neighbourhood",
     xlabel="Price Per Night (Euros)",
     ylabel="Neighbourhood",
     legend=None
 )
)

sns.despine()
```



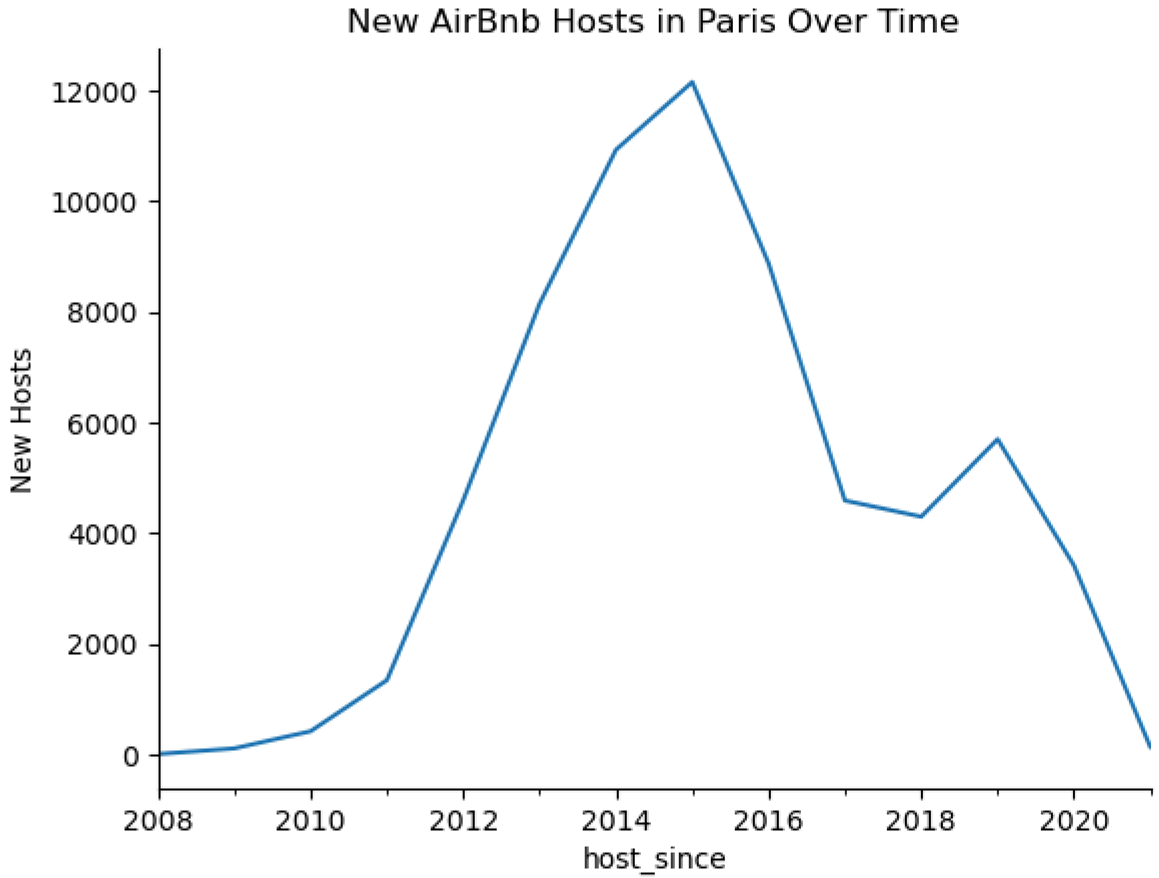
In [25]:

```
(paris_Listings_accommodates
 .plot
 .barh(
     title="Average Listing Price by Accommodation Number",
     xlabel="Price Per Night (Euros)",
     ylabel="Accommodation Capacity",
     legend=None
 )
)

sns.despine()
```



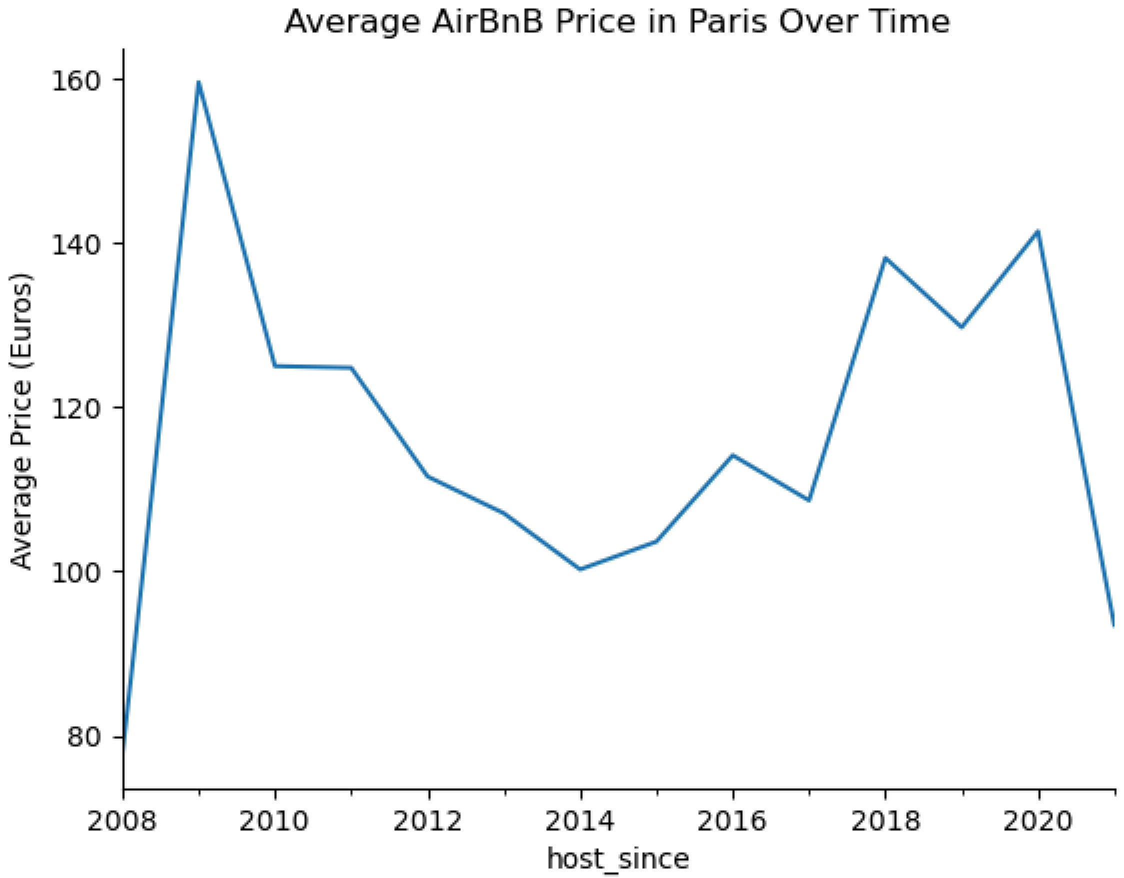
In [27]:

```
paris_Listings_over_time["neighbourhood"].plot(
    ylabel="New Hosts",
    title="New AirBnb Hosts in Paris Over Time"
)

sns.despine()
```



In [28]:

```
paris_Listings_over_time["price"].plot(
    ylabel="Average Price (Euros)",
    title="Average AirBnB Price in Paris Over Time"
)

sns.despine()
```

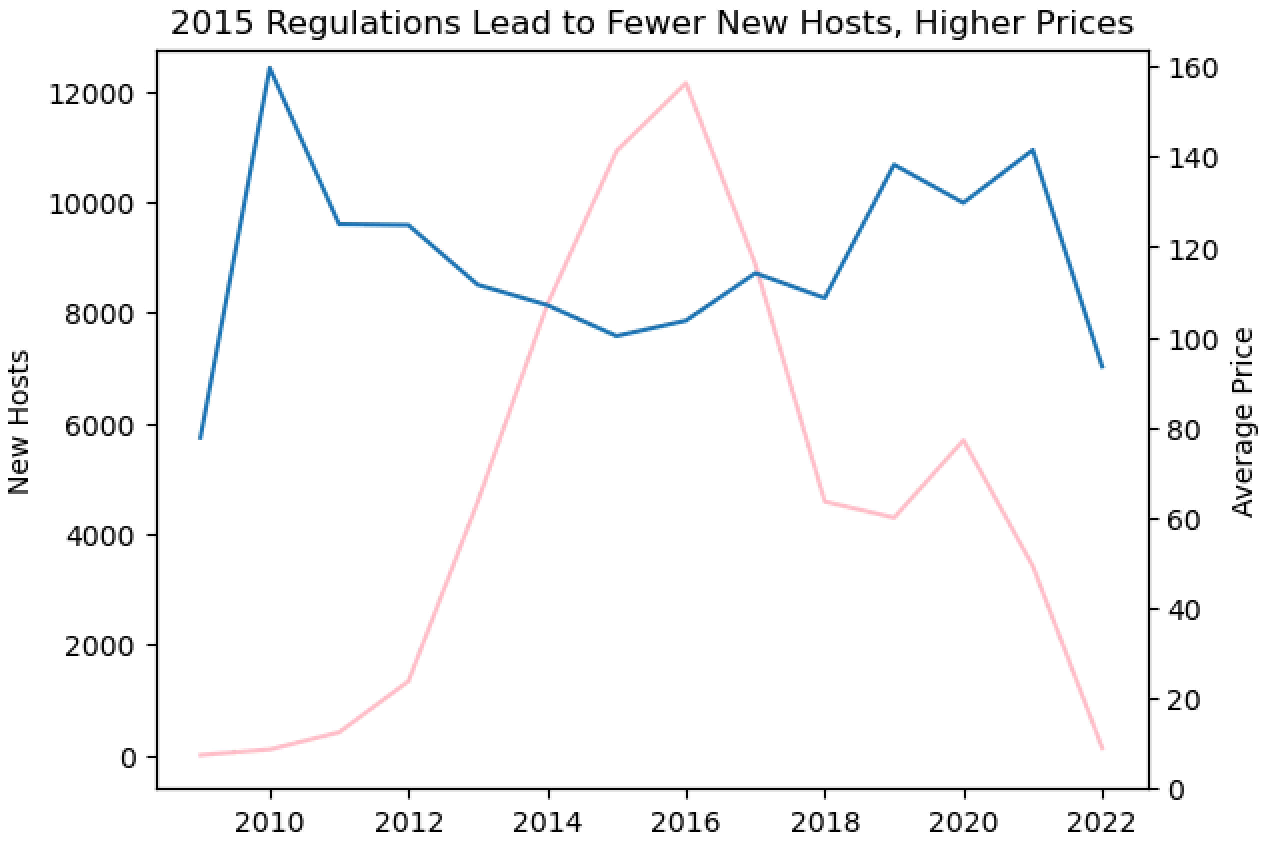


In [31]:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.plot(
    paris_Listings_over_time.index,
    paris_Listings_over_time["neighbourhood"],
    label="New Hosts",
    c="pink"
)

ax.set_ylabel("New Hosts")

ax2 = ax.twinx()

ax2.plot(
    paris_Listings_over_time.index,
    paris_Listings_over_time["price"],
    label="Average Price",
)
ax2.set_ylim(0)

ax2.set_ylabel("Average Price")

ax.set_title("2015 Regulations Lead to Fewer New Hosts, Higher Prices")
```

Out[31]:

```
Text(0.5, 1.0, '2015 Regulations Lead to Fewer New Hosts, Higher Prices')
```



In [ ]: