

Airbnb Listings & Reviews

In [1]: `import pandas as pd`

In [11]: `listings = pd.read_csv("Listings.csv" ,
encoding="ISO-8859-1" ,
low_memory= False ,
parse_dates = ["host_since"])`

In [4]: `listings.head()`

Out[4]:

	listing_id	name	host_id	host_since	host_location	host_response_time	host_response_rate
0	281420	Beautiful Flat in le Village Montmartre, Paris	1466919	2011-12-03	Paris, Ile-de-France, France	NaN	NaN
1	3705183	39 mÃ² Paris (Sacre CÃ©ur)	10328771	2013-11-29	Paris, Ile-de-France, France	NaN	NaN
2	4082273	Lovely apartment with Terrace, 60m2	19252768	2014-07-31	Paris, Ile-de-France, France	NaN	NaN
3	4797344	Cosy studio (close to Eiffel tower)	10668311	2013-12-17	Paris, Ile-de-France, France	NaN	NaN
4	4823489	Close to Eiffel Tower - Beautiful flat : 2 rooms	24837558	2014-12-14	Paris, Ile-de-France, France	NaN	NaN

5 rows × 33 columns

In [10]: `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                                  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                    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 [19]: paris_listings = listings.query("city == 'Paris']").loc[:, ["host_since", "neighbourho
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   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 [20]: paris_listings.isna().sum()
```

```
Out[20]: host_since      33
neighbourhood      0
city               0
accommodates       0
price              0
dtype: int64
```

```
In [21]: paris_listings.describe()
```

```
Out[21]:
```

	host_since	accommodates	price
count	64657	64690.000000	64690.000000
mean	2015-11-01 11:06:05.528867584	3.037997	113.096445
min	2008-08-30 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 [22]: paris_listings.query("price == 0 and accommodates==0 ").count()
```

```
Out[22]: host_since      54
neighbourhood      54
city               54
accommodates       54
price              54
dtype: int64
```

```
In [29]: price_listings_neighbourhood = (paris_listings.groupby("neighbourhood")
                                           .agg({"price" : "mean"})
                                           .sort_values("price"))
price_listings_neighbourhood.tail()
```

```
Out[29]:
```

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

```
In [57]: price_listings_accomodates = (
           paris_listings.query("neighbourhood == 'Elysee'")
           .groupby("accommodates")
           .agg({"price" : "mean"})
           .sort_values("price")
         )
price_listings_accomodates.tail()
```

Out[57]:

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

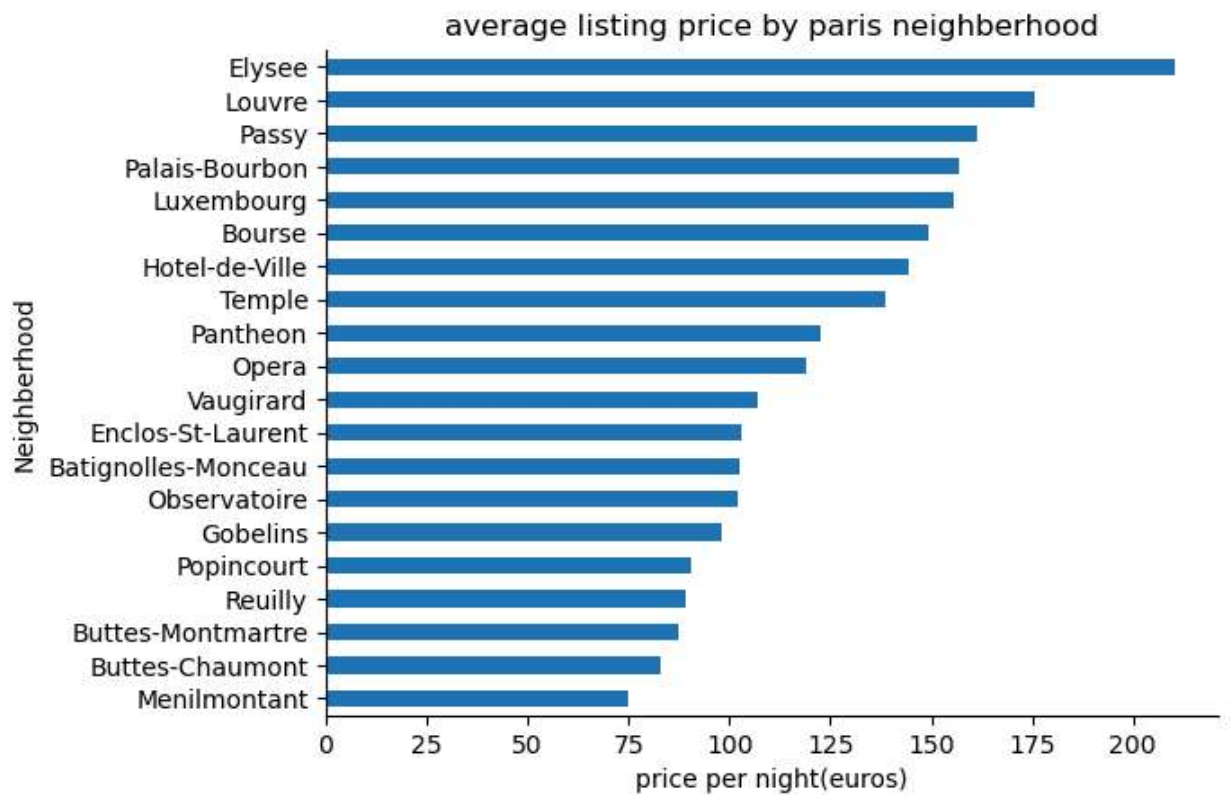
```
In [50]: paris_listings_over_time = (paris_listings
    .set_index("host_since")
    .resample("Y")
    .agg({"neighbourhood" : "count" , "price" : "mean"})
    )

paris_listings_over_time.head()
```

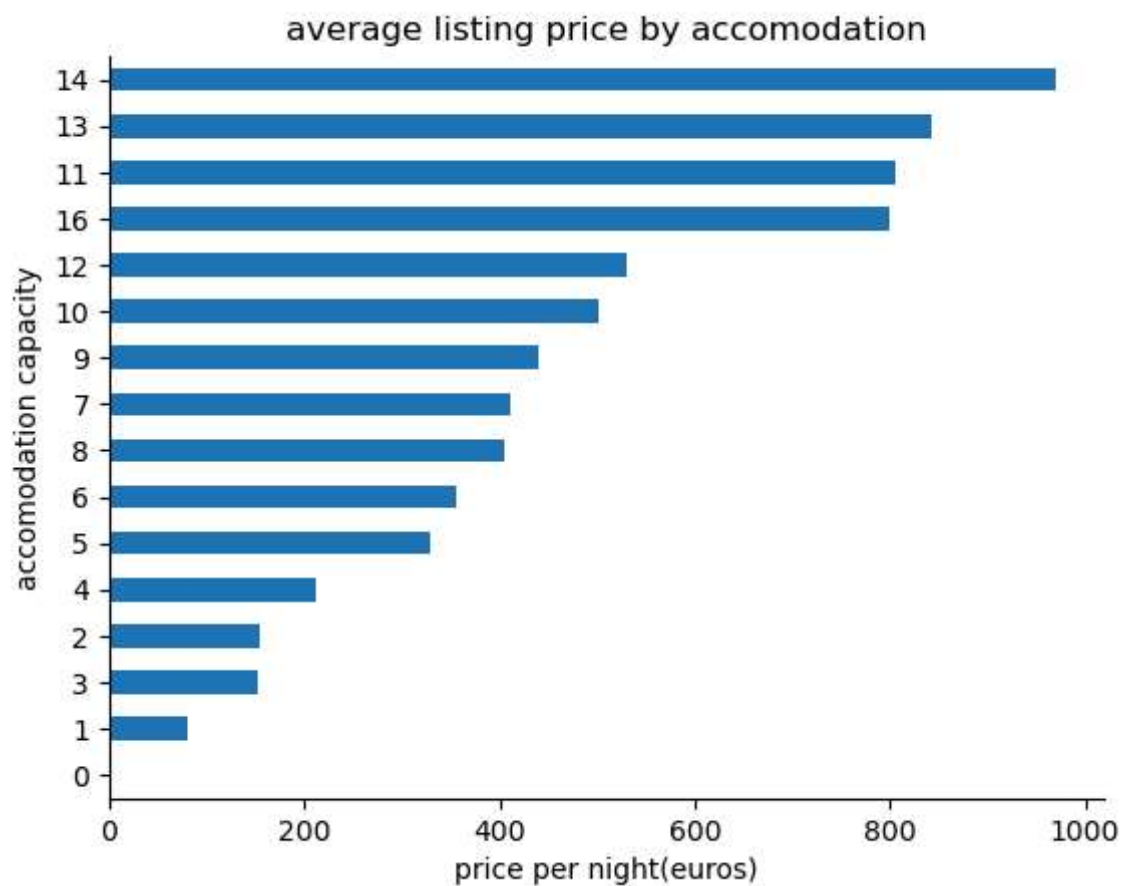
Out[50]:

	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

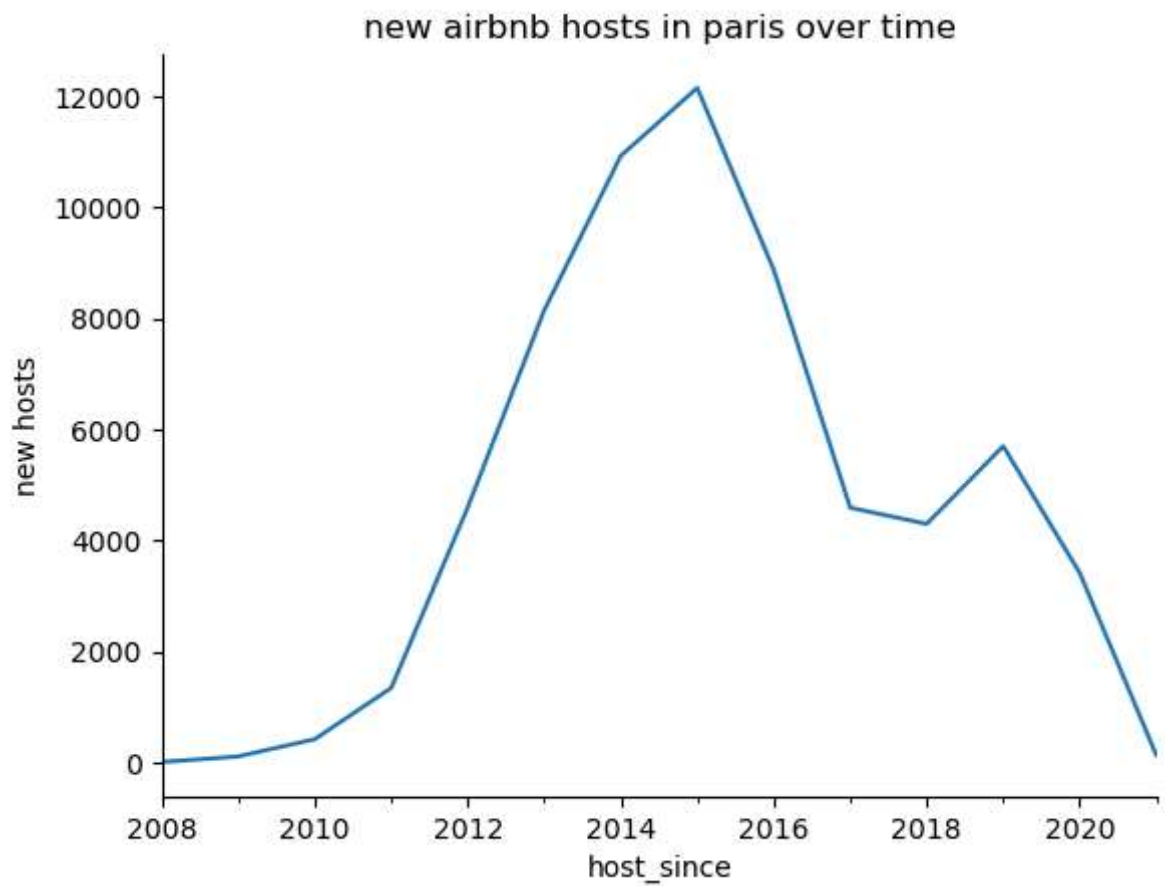
```
In [59]: import seaborn as sns
( price_listings_neighbourhood
    .plot
    .barh(
        title="average listing price by paris neighborhood" ,
        xlabel="price per night(euros)" ,
        ylabel="Neighborhood",
        legend= None )
    )
sns.despine()
```



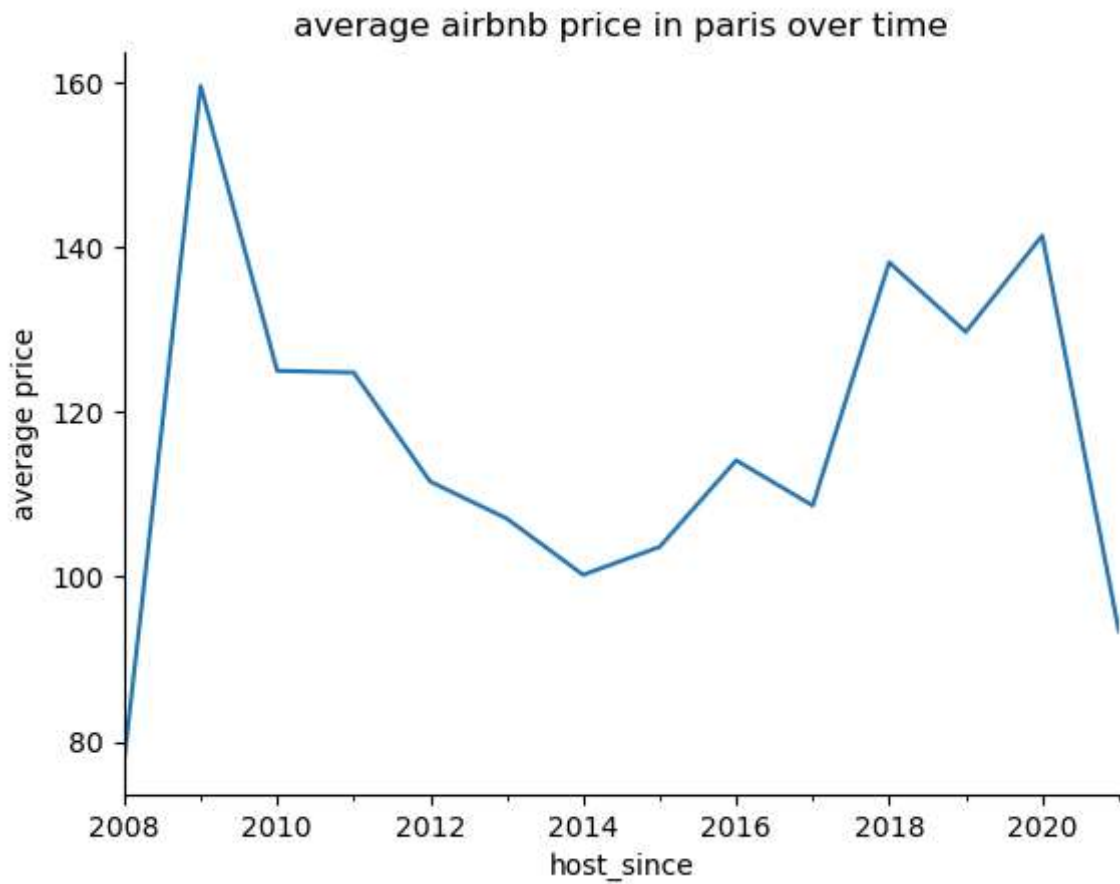
```
In [60]: ( price_listings_accomodates
          .plot
          .barh(
            title="average listing price by accomodation" ,
            xlabel="price per night(euros)" ,
            ylabel="accomodation capacity",
            legend= None )
          )
sns.despine()
```



```
In [61]: paris_listings_over_time["neighbourhood"].plot(  
                                                title=" new airbnb hosts in paris over time  
                                                ylabel="new hosts")  
sns.despine()
```



```
In [63]: paris_listings_over_time["price"].plot(  
        title=" average airbnb price in paris over t  
        ylabel="average price")  
sns.despine()
```

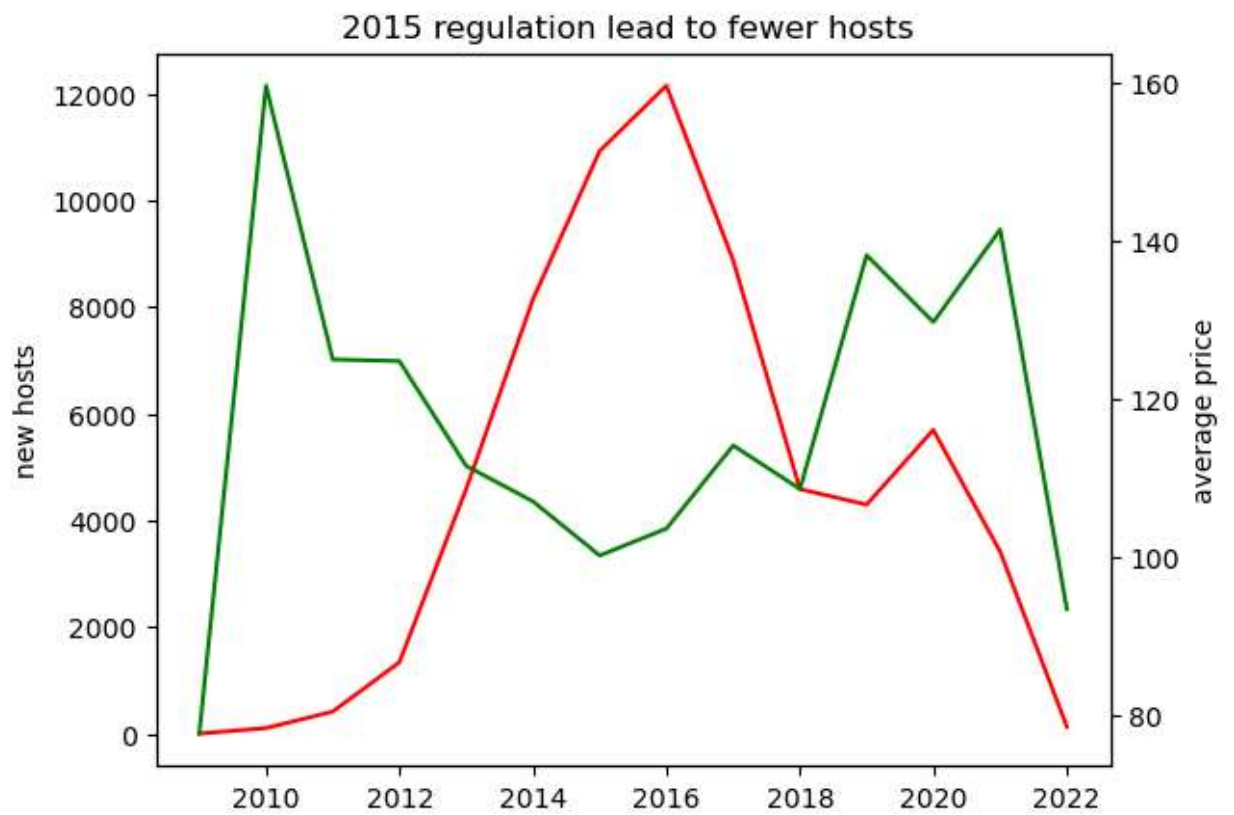


```
In [74]: import matplotlib.pyplot as plt

fig, ax = plt.subplots()
ax.plot(
    paris_listings_over_time.index ,
    paris_listings_over_time["neighbourhood"],
    label="new hosts",
    c="red"
)
ax.set_ylabel("new hosts")

ax2=ax.twinx()
ax2.plot(
    paris_listings_over_time.index ,
    paris_listings_over_time["price"],
    label="average price",
    c="green"
)
ax2.set_ylabel("average price")
ax.set_title("2015 regulation lead to fewer hosts ")
```

```
Out[74]: Text(0.5, 1.0, '2015 regulation lead to fewer hosts ')
```

In []: