In [1]:	<pre>import numpy as np import pandas as pd</pre>
	<pre>import matplotlib.pyplot as plt import seaborn as sns %matplotlib inline  # Loading data and printing out a few lines. df = pd.read_csv('rentals_cleaned_csv', index_col=0)</pre>
In [2]:	Univariate Exploration  Let's start our exploration by looking at the main variable of interest: price to see its distribution type: skewed or symmetric?  # univariate plot of price
	<pre>bins = np.arange(0, df['price'].max()+5000, 5000) plt.hist(df.price, bins = bins) plt.xlabel('Price (USD)') plt.ylabel('Frequncy') plt.title('Price distribution');</pre> <pre></pre>
	1750 - 1500 - 1250 - 2 1000 -
	1000 - 500 - 250 -
In [3]:	# Setting Xlimit to get rid of outliers effect bins = np.arange(0, df['price'].max()+2000, 2000)
	<pre>plt.hist(df.price, bins = bins) plt.xlabel('Price (USD)') plt.ylabel('Frequncy') plt.title('Price distribution') plt.xlim((0,150000));</pre> <pre>Price distribution</pre>
	800 - 700 - 600 -
	500 - 900 - 200 - 100 -
	20000 40000 60000 80000 100000 120000 140000  Price (USD)  Price distribution is skewed, most pooular prices are under 20000\$
In [4]:	<pre>bins = np.arange(5, df['bathrooms'].max()+1, 1) plt.hist(data = df, x = 'bathrooms', bins = bins, rwidth = 0.7)</pre>
	<pre>plt.xticks(np.arange(0, 14+1, 1)) plt.xlabel('No. of bathrooms') plt.title('Nmber of bathrooms distribution') plt.ylabel('count');</pre> <pre>Nmber of bathrooms distribution</pre>
	5000 - 4000 - 15 3000 -
	8 2000 - 1000 -
In [5]:	
	<pre>bins = np.arange(5, df['bedrooms'].max()+1, 1) plt.hist(data = df, x = 'bedrooms', bins = bins, rwidth = 0.7) plt.xticks(np.arange(0, 14+1, 1)); plt.xlabel('No. of bedrooms') plt.title('Nmber of bedrooms distribution') plt.ylabel('count');</pre>
	Nmber of bedrooms distribution  4000 -
	3000 - 2000 - 1000 -
	Most popular rentals have only one to two bedrooms
In [6]:	<pre>df.minimum_nights.value_counts().head().plot(kind='bar') plt.xlabel('minimum_nights') plt.ylabel('count') plt.title('Minimum_nights distribution')</pre>
	Minimum_nights distribution  3000 - 2500 -
	2000 - 1500 - 1000 -
	500 - 30 2 1 3 4 minimum_nights
In [7]:	<pre>base_color = sns.color_palette()[0]</pre>
	<pre>type_order = df.room_type.value_counts().index n_rentals = df.room_type.value_counts().sum()  # Type_counts contains the frequency of unique values in the `property_type` column in decreasing order. type_counts = df.room_type.value_counts()  sns.countplot(data=df, x='room type', color=base color, order=type order);</pre>
	<pre>plt.title('Room type counts')  # Logic to print the proportion text on the bars for i in range (type_counts.shape[0]):     count = type_counts[i]     # Convert count into a percentage, and then into string</pre>
	<pre>pct_string = '{:0.1f}%'.format(100*count/n_rentals)     # Print the string value on the bar.     plt.text(i,count+50, pct_string, va='center')  # Specify the figure size in inches, for both X, and Y axes plt.rcParams['figure.figsize'] = 12,8</pre>
	Room type counts  5000  4000 - 35.7%
	3000 - 2000 - 1000 -
In [8]:	
. •	<pre>base_color = sns.color_palette()[0] type_order = df.property_type.value_counts().index n_rentals = df.property_type.value_counts().sum()  # Type_counts contains the frequency of unique values in the `property_type` column in decreasing order. type_counts = df.property_type.value_counts()</pre>
	<pre>sns.countplot(data=df, y='property_type', color=base_color, order=type_order); plt.title('Property type counts')  # Logic to print the proportion text on the bars for i in range (type_counts.shape[0]):     count = type_counts[i]     # Convert count into a percentage, and then into string</pre>
	<pre># Convert count into a percentage, and then into string pct_string = '{:0.1f}%'.format(100*count/n_rentals) # Print the string value on the bar. plt.text(count+1, i, pct_string, va='center')  # Specify the figure size in inches, for both X, and Y axes plt.rcParams['figure.figsize'] = 12,8</pre>
	Property type counts  Apartment House Condominium Guest suite 7.1%  Property type counts  28.6%
	Boutique hotel - 3.3%  Hotel - 1.9%  Townhouse - 1.5%  Serviced apartment - 1.5%  Loft - 1.1%  Hostel - 1.1%
	Bed and breakfast - 0.5%  Aparthotel - 0.5%  Guesthouse - 0.5%  Other - 0.3%  Bungalow - 0.2%  Resort - 0.2%
	Cottage   0.1%  Villa   0.1%  Castle   0.0%  Cabin   0.0%  Tiny house   0.0%  Earth house   0.0%  Hut   0.0%
	Dome house -0.0%  Camper/RV -0.0%  In-law -0.0%  0 500 1000 1500 2000 2500 3000  count
	Most common room and property type are the apartment type for both the room and the property.  Bivariate Exploration  Lets start with the numerical values effect on price
In [9]:	<pre># correlation plot numeric_vars = ['bedrooms', 'bathrooms', 'minimum_nights', 'price'] categoric_vars = ['room_type', 'property_type'] plt.figure(figsize = [8, 5]) sns.heatmap(df[numeric_vars].corr(), annot = True, fmt = '.3f',</pre>
	Correlation bewteen numerical variables  bedrooms - 1.000 0.325 -0.017 0.376
	bathrooms - 0.325 1.000 0.021 0.122 -0.6
	minimum_nights0.017 0.021 1.000 0.034 -0.4  price - 0.376 0.122 0.034 1.000 -0.2
In [10]:	# Visulizing the relation between bedrooms and bathrooms sns.scatterplot(data = df, x = 'bedrooms', y = 'bathrooms', ci=False) plt.xlabel('Number of bedrooms')
	plt.ylabel('Number of bathrooms') plt.title('relation between bedrooms and bathrooms');  relation between bedrooms and bathrooms  14
	10 -
	Number of bathrooms  8 -  8 -  8 -  10 -
	변화 6 - 4 -
	0 2 4 6 8 10 12 14
In [11]:	
In [11]:	<pre>We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot_line_heat() def plot_scatter_heat(df, price, column):     """ Shows the relation between price and the other numerical column by returning a heat map and a scatterp plt.figure(figsize = [18, 6])     plt.suptitle('relation between price and ' + column);  # PLOT ON LEFT     plt.subplot(1, 2, 1)</pre>
In [11]:	<pre>We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot_line_heat() def plot_scatter_heat(df, price, column):     """ Shows the relation between price and the other numerical column by returning a heat map and a scatterp plt.figure(figsize = [18, 6])     plt.suptitle('relation between price and ' + column);  # PLOT ON LEFT     plt.subplot(1, 2, 1)     sns.scatterplot(data = df, x = column, y = price, ci=False)     plt.xlabel('Number of '+ column)     plt.ylabel('Price')  # PLOT ON RIGHT     plt.subplot(1, 2, 2)</pre>
In [11]: In [12]:	<pre>We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot_line_heat() def plot_scatter_heat(df, price, column):     """ Shows the relation between price and the other numerical column by returning a heat map and a scatterp plt.figure(figsize = [18, 6])     plt.suptitle('relation between price and ' + column);  # PLOT ON LEFT     plt.subplot(1, 2, 1)     sns.scatterplot(data = df, x = column, y = price, ci=False)     plt.xlabel('Number of '+ column)     plt.ylabel('Price')  # PLOT ON RIGHT     plt.subplot(1, 2, 2)     plt.hist2d(data = df, x = column, y = price, cmin=0.5, cmap='viridis_r')     plt.colorbar()     plt.xlabel('Number of '+ column)     plt.ylabel('Price (USD)')</pre>
	<pre>We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot_line_heat() def plot_scatter_heat(df, price, column):     """ Shows the relation between price and the other numerical column by returning a heat map and a scatters plt.figure(figsize = [18, 6])     plt.suptitle('relation between price and ' + column);  # PLOT ON LEFT     plt.subplot(1, 2, 1)     sns.scatterplot(data = df, x = column, y = price, ci=False)     plt.xlabel('Number of '+ column)     plt.ylabel('Price')  # PLOT ON RIGHT     plt.subplot(1, 2, 2)     plt.hist2d(data = df, x = column, y = price, cmin=0.5, cmap='viridis_r')     plt.colorbar()     plt.xlabel('Number of '+ column)     plt.ylabel('Price (USD)')  # The relation between price and bathrooms plot_scatter_heat(df, 'price', 'bathrooms')      relation between price and bathrooms  plot_scatter_heat(df, 'price', 'bathrooms')</pre>
	<pre>We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot_line_heat() def plot_scatter_heat(df, price, column):     """ Shows the relation between price and the other numerical column by returning a heat map and a scatterp plt.figure(figsize = [18, 6])     plt.suptitle('relation between price and ' + column);  # PLOT ON LEFT plt.subplot(1, 2, 1)     sns.scatterplot(data = df, x = column, y = price, ci=False)     plt.xlabel('Number of '+ column)     plt.ylabel('Price')  # PLOT ON RIGHT plt.subplot(1, 2, 2) plt.hist2d(data = df, x = column, y = price, cmin=0.5, cmap='viridis_r') plt.colorbar() plt.xlabel('Number of '+ column) plt.ylabel('Price (USD)')  # The relation between price and bathrooms plot_scatter_heat(df, 'price', 'bathrooms')  relation between price and bathrooms  plot_scatter_heat(df, 'price', 'bathrooms')</pre>
	We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot line heat()  def plot_scatter_heat(df, price, column):  """ Shows the relation between price and the other numerical column by returning a heat map and a scatterp plot.sighure(figure) = [18, 6])  plt.suptitle('relation between price and ' + column);  # PDOT ON LEFT  plt.subplot(1, 2, 1)  sns.scatterplot(data = df, x = column, y = price, ci=False)  plt.xlabel('Number of '+ column)  plt.xlabel('Number of '+ column)  plt.subplot(1, 2, 2)  plt.hist2d(data = df, x = column, y = price, cmin=0.5, cmap='viridis_r')  plt.colorbar()  plt.xlabel('Price (USD)')  # The relation between price and bathrooms  plot_scatter_heat(df, 'price', 'bathrooms')  relation between price and bathrooms  plot_scatter_heat(df, 'price', 'bathrooms')  relation between price and bathrooms  10  -5000  -60000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -700000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -70000  -700000  -70
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot line hear()  def plot scatter hear(df, price, column):  """ Shows the relation between price and the other numerical column by returning a heat map and a scattery plt. figure (figure = (18, 6)) plt. auptitle('relation between price and ' + column);  # FEOTO M LEFT plt.subplot(1, 2, 1) span.scatterplot(data = df, x = column, y = price, ci=Felse) plt.ylabel('Price')  # FEOTO M RIGHT plt.ylabel('Price')  # FEOTO M RIGHT plt.subplot(1, 2, 2) plt.instructed(data = df, x = column, y = price, cmin=0.5, cmap='viridis_f') plt.xlabel('Price (USD)')  # The relation between price and bathrooms plot_scatter_heat(df, 'price', 'bothrooms')  relation between price and bathrooms  relation between price and bathrooms  There is no strong relation between price and number of bathrooms, this is what we concluded before from the statstical approach.
	We can see that as the number of bedrooms increases the number of bathroom increases  # Define a function plot line heat()  def plot scatter heat(df, price, column);  """ Shoes the relation between price and the obter numerical column by returning a heat map and a statters plut. Figure (figalize = [18, 6])  pll.rapit.let(relation between price and ' + column);  # FLOT ON LEFT  plt.subplot(1, 2, 1)  sns.scatterplot(date = df, x = column, y = price, ci=False)  plt.subplot(1, 2, 1)  sns.scatterplot(date = df, x = column, y = price, ci=False)  plt.scatter(return);  # SLOT ON RIGHT  plt.scatter(return);  plt.scatter(return);  plt.scatter(return);  plt.scatter(return);  plt.scatter(return);  plt.scatter(return);  plt.scatter(return);  relation between price and bathrooms  relation between price and bathrooms  **There is no strong relation between price and number of bathrooms, this is what we concluded before from the statical approach.  # The relation between price and number of bathrooms, this is what we concluded before from the statical approach.  # The relation between price and number of bathrooms, this is what we concluded before from the statical approach.  # The relation between price and number of bathrooms, this is what we concluded before from the statical approach.  # The relation between price and bathrooms  * Plot scatter_heat(df, 'price', 'bedrooms')
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases  ### Destine of Numbrian plan_liam_brake()  def plot scatter beacted; price, column):  """ Stows the relation between price and the other numerical column by returning a heat map and a scattery plit.digure([ipside = [10, 6]]) plit.apprit([ipside = [10, 6]]) plit.ap
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases  # Setime a function plot_ine_heat()  dee plot_mustater_lensi(df, price, column);  """ Shows the relation between price and ' + column);  # NOTE OF NORTH COLUMN (SET)  plot.*Sputie.*(setime at df, x = column, y = price, ci=False)  plot.*Sputie.*(setime at df, x = column, y = price, ci=False)  plot.*Spot('subtor of '+ column)  plot.*Spot('subtor of '+ column)  # PROFE OF NORTH  # PROFE OF NORT
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases  # Oberine a function plot line heac()  dad plot nonzero homo(of, price, calumn):  ## Blooms this relation between price and the other numerical cultum by returning a heat map and a most berg politique (liquide of 10, 10)  ## Plot of LEFT  ## PLOT ON LEFT  ## P
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases  # Total Control of the Co
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases  ### Application of the plant plant beauty  ### Application of the plant beauty  ### Application between processed and the other numerical column by transfer a financial plant beauty  ### Application between processed and in other numerical column by transfer a financial plant beauty  ### Application between processed and in other numerical column by transfer a financial plant beauty  ### Application between processed and in other numerical column by transfer and mapping a financial plant beauty  ### Application between processed and plant beauty  ### Application between processed plant beauty  ### Application between processed plant beauty  #### Application between processed plant beauty  #### Application between processed plant beauty  ##### Application between processed plant beauty  ###################################
In [12]:	We can see that as the number of bedrooms increases the number of bathroom increases.  2 Infects on discretion place (1964) bedrooms and the order managinal bedroom by increasing a page and a percentage of the Number of Bedrooms and the order managinal bedrooms by increasing the security of the place of
In [12]:	We can see that as the number of bedrooms increases the number of bedroom increases  # Contact or standards and price of (1) price (1) p
In [12]:	We can see that as the number of bethoms increases the number of bathround increases.  **Postage 3 Franciscop people place people of the content of bathround increases of the content of
In [12]:  In [13]:	We can see that as the number of budinous increases the number of budinous increases.  **Octave of street of the production person pers
In [12]:  In [13]:	We can see that as the muster of anthroom increases the number of anthroom increases.  **Best and **Increasing parts**, and **See **
In [12]:  In [13]:	We can see that as the number of bestcours increases the number of bestcours increases.  **Assume a Constitution of the Language of the Constitution of the Constituti
In [12]:  In [13]:	We can see that in the number of bedoods in access the number of bedonominous executions and access the second process of the process of the number of the n
In [12]:  In [13]:	The content of the content of bedooms invested the name of challman in the content of a structure of the content of the conten
In [12]:  In [13]:	When the control of t
In [12]: In [13]:	There is no construction to the control of the cont
In [12]: In [13]:	The contraction of the contract of the contrac
In [12]: In [13]:	The content of the co
In [12]: In [13]:	The carbon production of the dependence of the control of the cont
In [12]: In [13]:	Personal and the control of the cont
In [12]: In [13]:	The contract of the contract o
In [12]: In [13]:	Grant and the common of the co
In [13]:  In [14]:	Control of the third was also of the control of the
In [13]:  In [14]:	Secretary and the control of the con
In [13]:  In [14]:	When the control products are the control products and the control products are the control products and the control products are the control prod
In [13]:  In [14]:	Action and an action of a financial and action of acti
In [13]:  In [14]:	The control of the co
In [13]:  In [14]:	And the control of th
In [13]:  In [14]:	The content of the co
In [13]:  In [14]:	The control of the co
In [13]:  In [14]:	And the content of th
In [13]:  In [14]:	The control of the co
In [13]:  In [14]:  In [16]:	The second control of a control
In [13]:  In [14]:  In [16]:	The control of the co
In [13]:  In [14]:  In [16]:	Control of the contro
In [13]:  In [14]:  In [16]:	The content of the co
In [13]:  In [14]:  In [16]:  In [17]:	The second control of the control of
In [13]:  In [14]:  In [16]:  In [17]:	The content of the co
In [13]:  In [14]:  In [16]:  In [17]:	And the control of th