In [2]:
```
# Put these at the top of every notebook, to get automatic reloading and inl
from IPython.core.display import display, HTML
import pandas as pd
import warnings
import ast
warnings.filterwarnings('ignore')

%reload_ext autoreload
%autoreload 1
%matplotlib inline

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

display(HTML("<style>.container { width:100% !important; }</style>"))
```

In [3]:
```
import os
import seaborn as sns
import pandas as pd
import math

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor, Gradi
from sklearn.model_selection import GridSearchCV

from sklearn.linear_model import Lasso, Ridge

from sklearn.metrics import mean_squared_error as MSE

from imblearn.over_sampling import SMOTE

from Utils.UtilsGeoViz import *
from Utils.UtilsViz import *
from Utils.DataUtils import *

RANDOM_SEED = 42
```

In [4]:
```
data_path = os.path.join("C:\\Users\\SSrih\\OneDrive\\UChicago\\DataMining\\
listings = pd.read_csv(data_path, index_col="id")
display(listings.shape)
```

```
(48855, 80)
```

Correlation

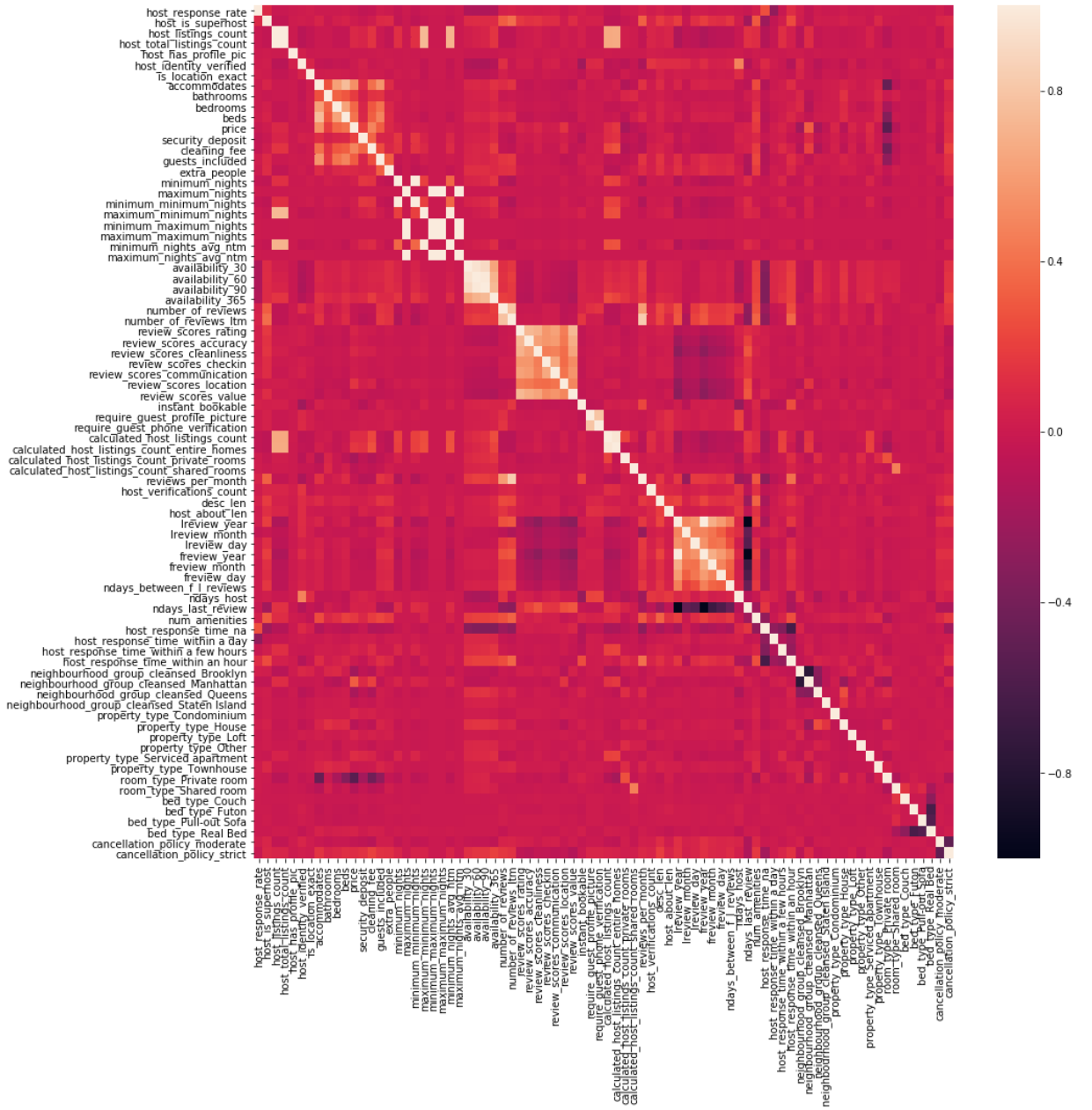
In [4]:
```
print(os.path.join(os.getcwd(),"corr.png"))
```

```
D:\Dev\Sources\Projects\GitProjects\listings\ssh\corr.png
```

In [5]:

```
corr_cols = ["host_response_rate", "host_listings_count", "host_total_listin
             "minimum_nights", "maximum_nights", "minimum_minimum_nights", "m
             "availability_30", "availability_60", "availability_90", "availa
             "review_scores_rating", "review_scores_accuracy", "review_scores
             "calculated_host_listings_count", "calculated_host_listings_coun
             "host_verifications_count", "desc_len", "host_about_len", "lrevi
              "ndays_between_f_l_reviews", "ndays_host", "ndays_last_review",
listings_corr = listings.corr()

f, ax = plt.subplots(1,1,figsize=(15,15))
hm = sns.heatmap(listings_corr, annot=False, ax=ax)
plt.yticks(np.arange(0.5, len(listings_corr.index), 1), listings_corr.index)
plt.xticks(np.arange(0.5, len(listings_corr.columns), 1), listings_corr.colu
# plt.show()
save_path = os.path.join(os.getcwd(),"corr.png")
fig = hm.get_figure()
fig.savefig(save_path)
```



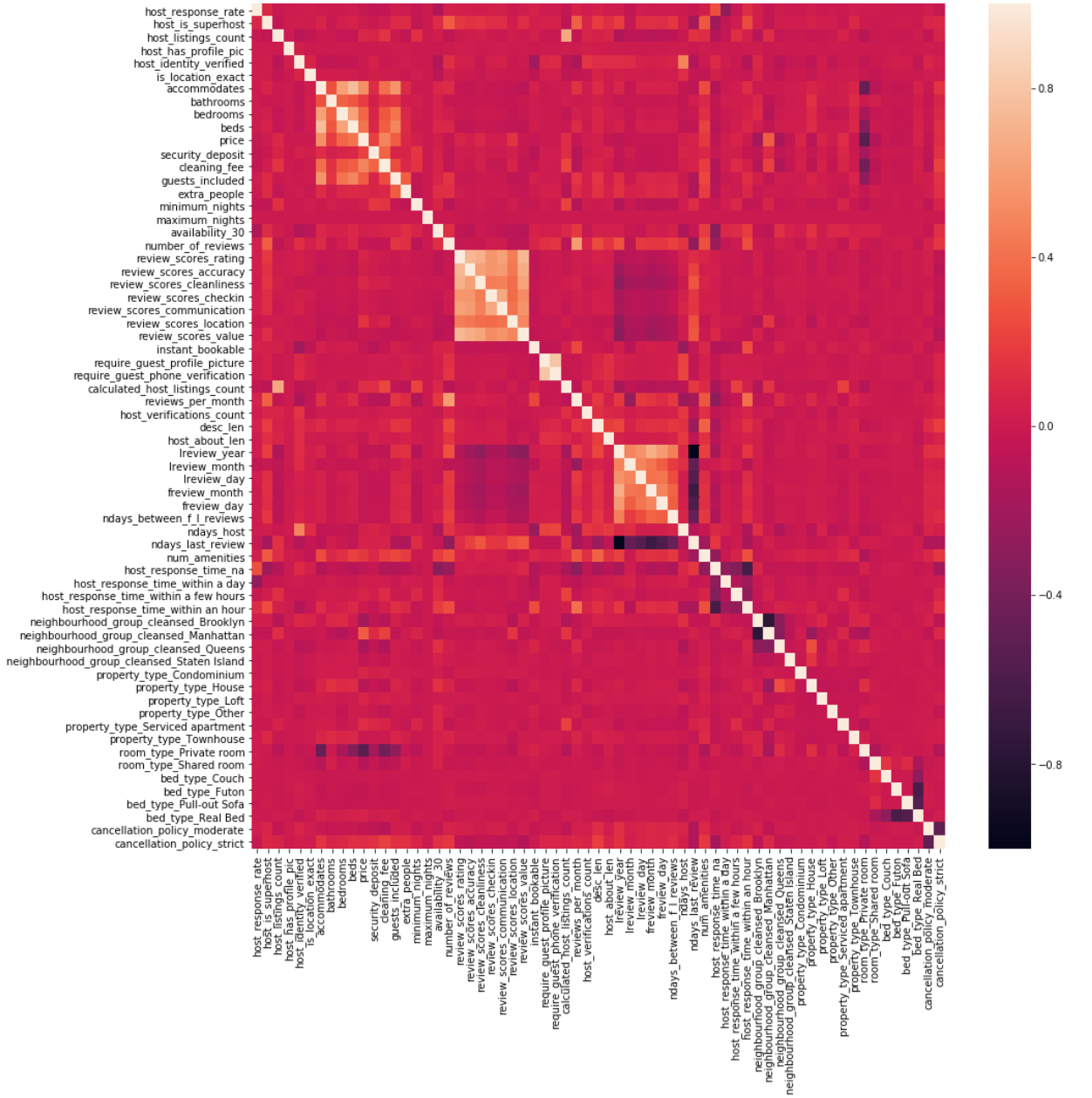
In [6]:

```
cols_to_drop = ["host_total_listings_count",
                    "minimum_minimum_nights", "minimum_maximum_nights", "max
                    "maximum_nights_avg_ntm", "availability_60", "availabili
                    "calculated_host_listings_count_entire_homes", "calculat
                    "calculated_host_listings_count_shared_rooms", "freview_
                    "maximum_minimum_nights", "minimum_nights_avg_ntm", "num

for col in cols_to_drop:
    if col in listings.columns:
        listings.drop(labels=col, inplace=True, axis=1)
```

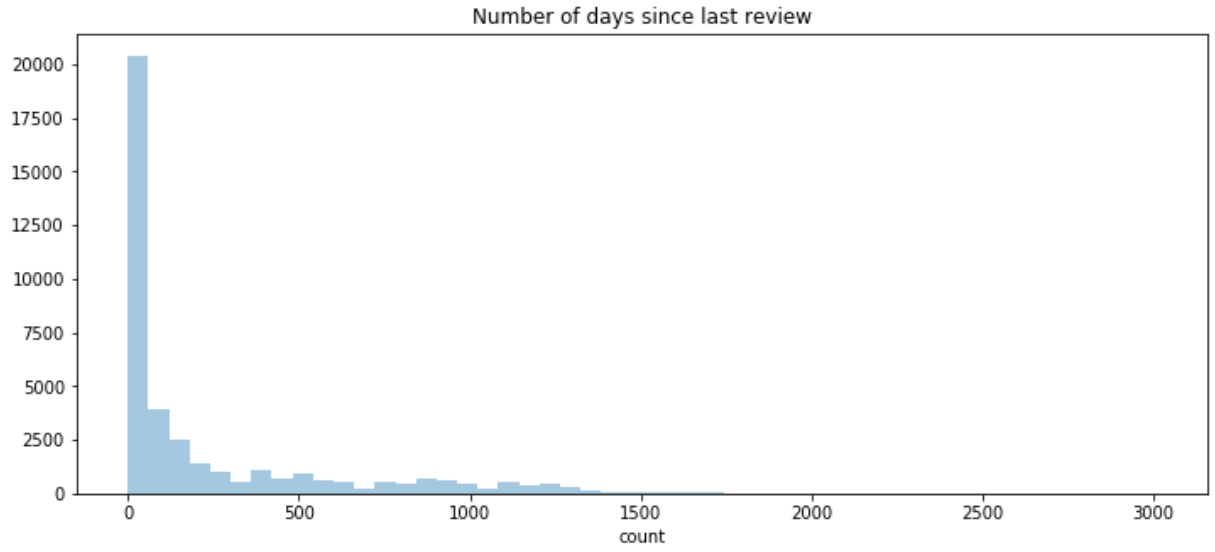
In [7]:
```
listings_corr = listings.corr()

f, ax = plt.subplots(1,1,figsize=(15,15))
hm = sns.heatmap(listings_corr, annot=False, ax=ax)
plt.yticks(np.arange(0.5, len(listings_corr.index), 1), listings_corr.index)
plt.xticks(np.arange(0.5, len(listings_corr.columns), 1), listings_corr.colu
# plt.show()
save_path = os.path.join(os.getcwd(),"corr_out.png")
fig = hm.get_figure()
# fig.savefig(save_path)
```



Feature engineered

In [7]:
```
f, ax = plt.subplots(1,1,figsize=(12,5))
g = sns.distplot(listings[listings["ndays_last_review"]<99999]["ndays_last_r
t = g.set(title="Number of days since last review", xlabel="count")
```



In [ ]:
```
1
```

In [9]:
```
f, ax = plt.subplots(1,1,figsize=(12,5))
g = sns.distplot(listings["ndays_host"], kde=False, ax=ax)
t = g.set(title="Number of days a host has been with Airbnb", xlabel="count"
```