Python Seaborn Package Introduction

Yingjie Qu (yq2350)

Liwen Zhu (lz2512)

We will introduce how to use Seaborn, a python package, to visualize data.

The first step is to import the package. We usually import seaborn as sns. Other packages we imported are also helpful for data visualization, but we will focus on seborn.

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

In [2]:
```
df = pd.read_csv('train.csv')
```

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

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl

5 rows × 81 columns

Histogram

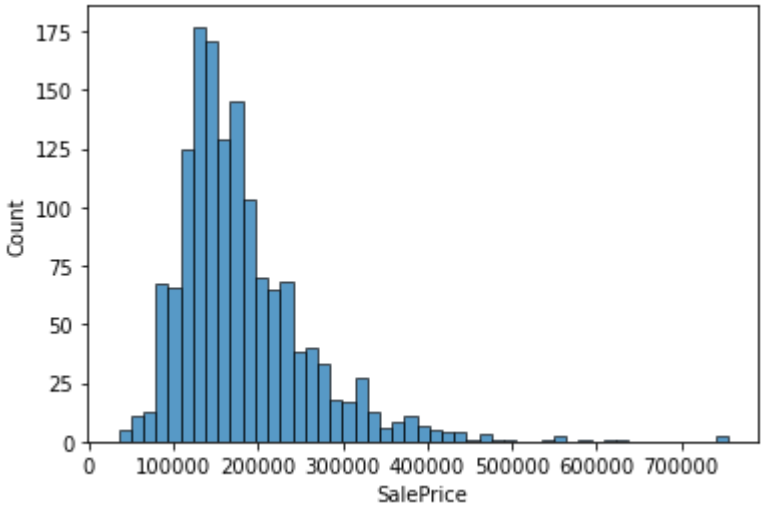
seaborn.histplot(data=None, *, x=None, y=None, hue=None, weights=None, stat='count', bins='auto', binwidth=None, binrange=None, discrete=None, cumulative=False, common_bins=True, common_norm=True, multiple='layer', element='bars', fill=True, shrink=1, kde=False, kde_kws=None, line_kws=None, thresh=0, pthresh=None, pmax=None, cbar=False, cbar_ax=None, cbar_kws=None, palette=None, hue_order=None, hue_norm=None, color=None, log_scale=None, legend=True, ax=None, **kwargs)

The **data** is the data we will visualize, and **x** is the variable we will present. We can also control the histogram by setting **bins** to limit the bin's number, **binwidth** to control the width, and **hue** to color each container.

Link for more details: https://seaborn.pydata.org/generated/seaborn.histplot.html

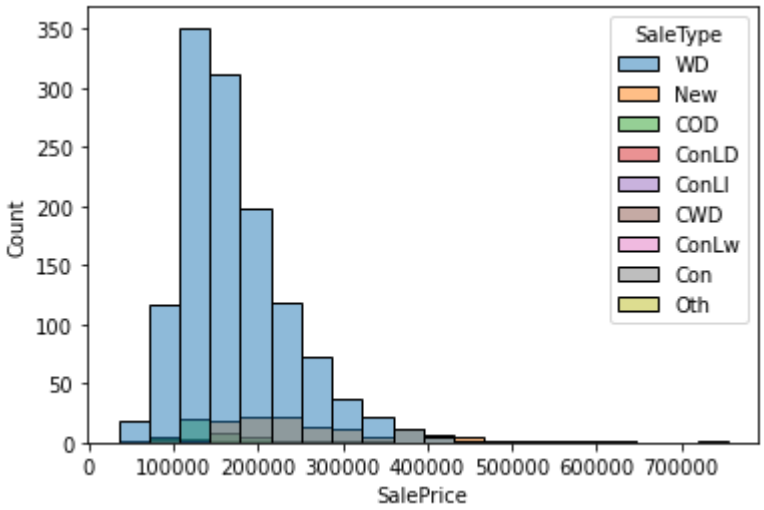
In [4]:
```
sns.histplot(data=df,x='SalePrice')
```

Out[4]: <AxesSubplot:xlabel='SalePrice', ylabel='Count'>



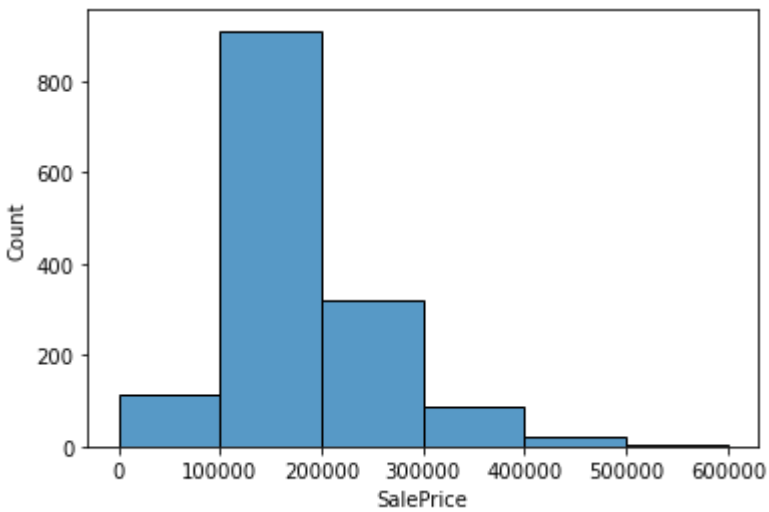
In [5]:
```
sns.histplot(data=df,x='SalePrice',hue='SaleType',bins=20)
```

Out[5]: <AxesSubplot:xlabel='SalePrice', ylabel='Count'>



In [6]:
```
sns.histplot(data=df,x='SalePrice',binwidth=100000,binrange=(0,600000))
```

Out[6]: <AxesSubplot:xlabel='SalePrice', ylabel='Count'>



Count plot

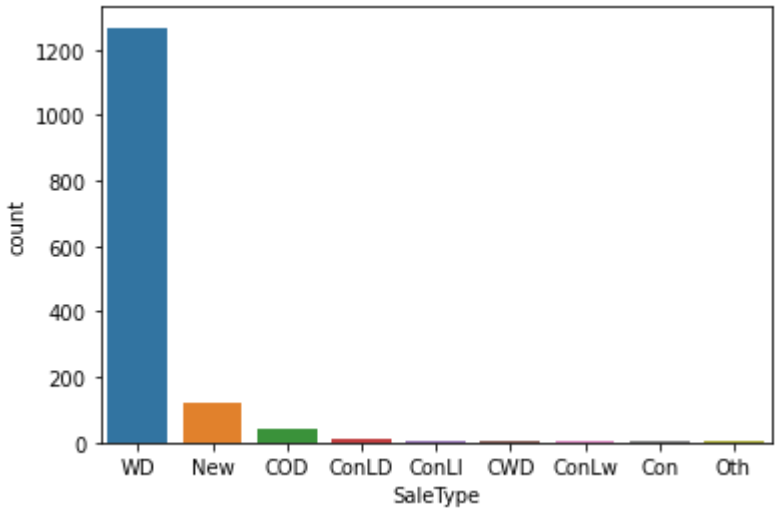
seaborn.countplot(data=None, *, x=None, y=None, hue=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, width=0.8, dodge=True, ax=None, **kwargs)

The count plot is useful for counting the frequencies of a categorical variable.

Link for more details: https://seaborn.pydata.org/generated/seaborn.countplot.html

In [7]:
```
sns.countplot(data=df,x='SaleType')
```

Out[7]:
```
<AxesSubplot:xlabel='SaleType', ylabel='count'>
```



Scatter Plot

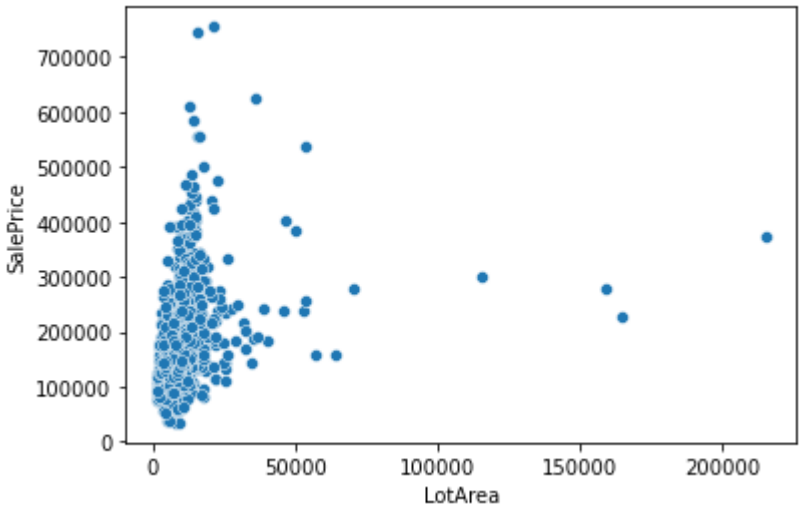
seaborn.scatterplot(data=None, *, x=None, y=None, hue=None, size=None, style=None, palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None, size_norm=None, markers=True, style_order=None, legend='auto', ax=None, **kwargs)

Besides setting x and y varibales, we can also set the **size** to control how big the dots are.

Link for more details: https://seaborn.pydata.org/generated/seaborn.scatterplot.html

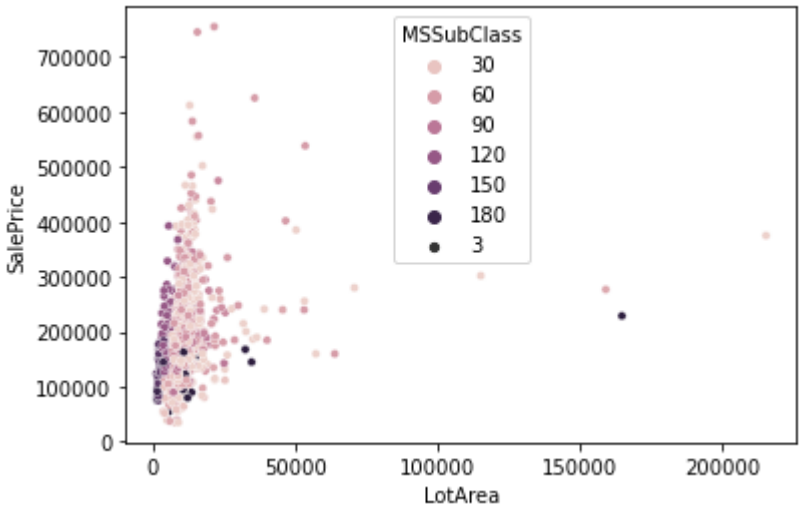
In [8]:
```
sns.scatterplot(data=df,x='LotArea',y='SalePrice')
```

Out[8]:
```
<AxesSubplot:xlabel='LotArea', ylabel='SalePrice'>
```



In [9]:
```
sns.scatterplot(data=df,x='LotArea',y='SalePrice',hue='MSSubClass',size=3)
```

Out[9]:
```
<AxesSubplot:xlabel='LotArea', ylabel='SalePrice'>
```



Box plot

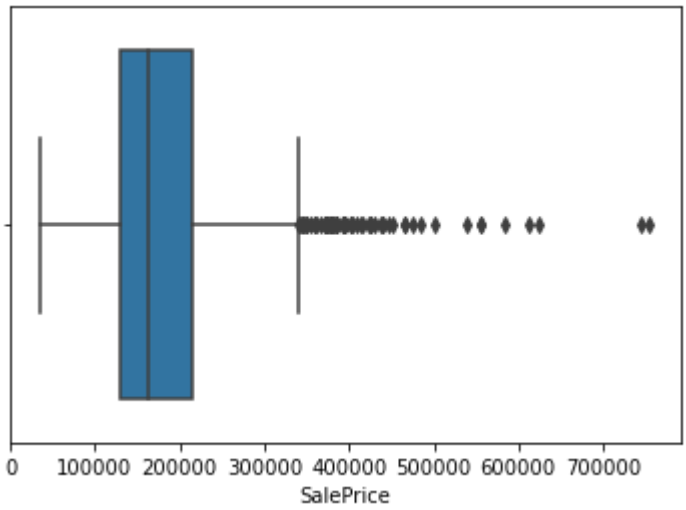
seaborn.boxplot(data=None, *, x=None, y=None, hue=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, width=0.8, dodge=True, fliersize=5, linewidth=None, whis=1.5, ax=None, **kwargs)

The color of box can be changed by setting **color**.

Link for more details: https://seaborn.pydata.org/generated/seaborn.boxplot.html

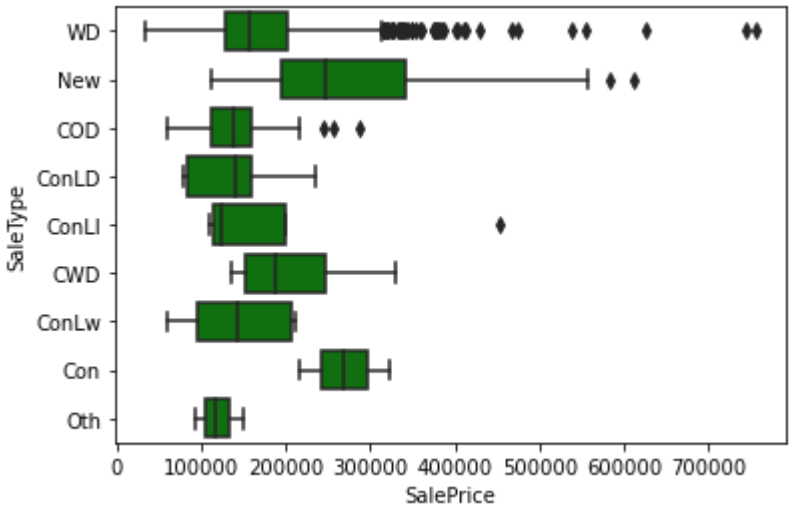
In [10]:
```
sns.boxplot(data=df,x='SalePrice')
```

Out[10]:
```
<AxesSubplot:xlabel='SalePrice'>
```



In [11]:
```
sns.boxplot(data=df,x='SalePrice',y='SaleType',color='g')
```

Out[11]:
```
<AxesSubplot:xlabel='SalePrice', ylabel='SaleType'>
```



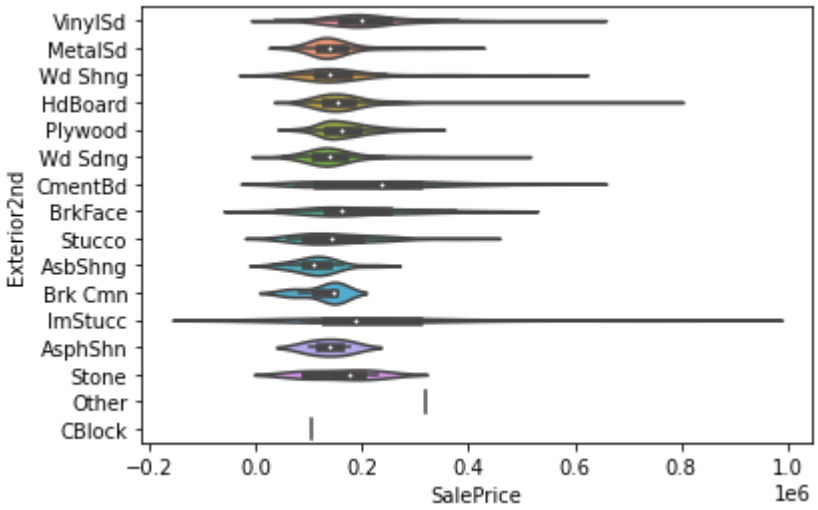
Violin Plot

seaborn.violinplot(data=None, *, x=None, y=None, hue=None, order=None, hue_order=None, bw='scott', cut=2, scale='area', scale_hue=True, gridsize=100, width=0.8, inner='box', split=False, dodge=True, orient=None, linewidth=None, color=None, palette=None, saturation=0.75, ax=None, **kwargs)

Link for more details: https://seaborn.pydata.org/generated/seaborn.violinplot.html

In [12]:
```
sns.violinplot(data=df, x="SalePrice", y="Exterior2nd")
```

Out[12]:
```
<AxesSubplot:xlabel='SalePrice', ylabel='Exterior2nd'>
```



Distribution plot

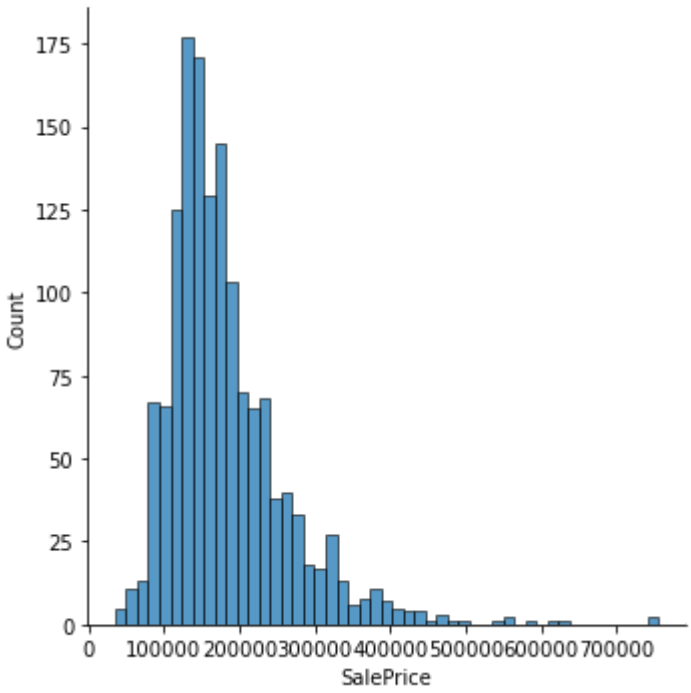
seaborn.displot(data=None, *, x=None, y=None, hue=None, row=None, col=None, weights=None, kind='hist', rug=False, rug_kws=None, log_scale=None, legend=True, palette=None, hue_order=None, hue_norm=None, color=None, col_wrap=None, row_order=None, col_order=None, height=5, aspect=1, facet_kws=None, **kwargs)

The distribution plot is similar to the histogram. The difference is that we can set **kind** to show the distribution curve.

Link for more details:
https://seaborn.pydata.org/generated/seaborn.displot.html#seaborn.displot

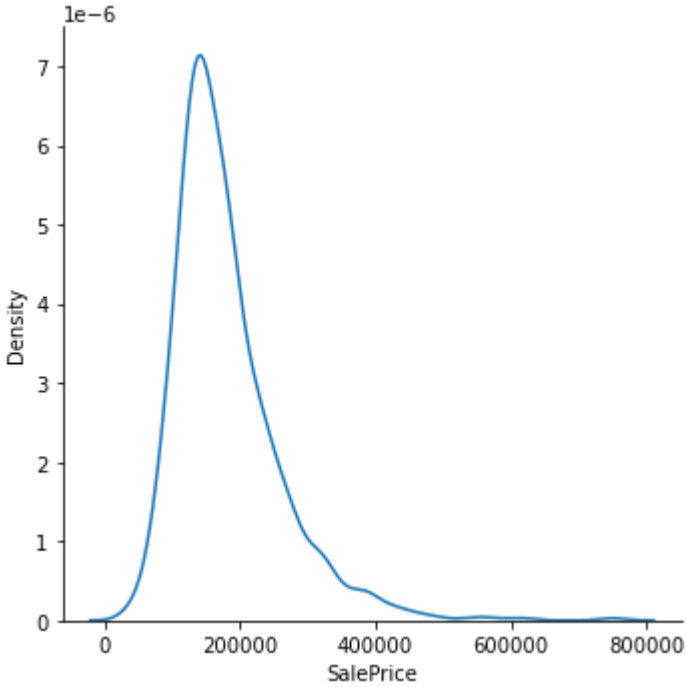
In [13]:
```
sns.displot(data=df,x='SalePrice')
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x7fabea1e56d0>
```



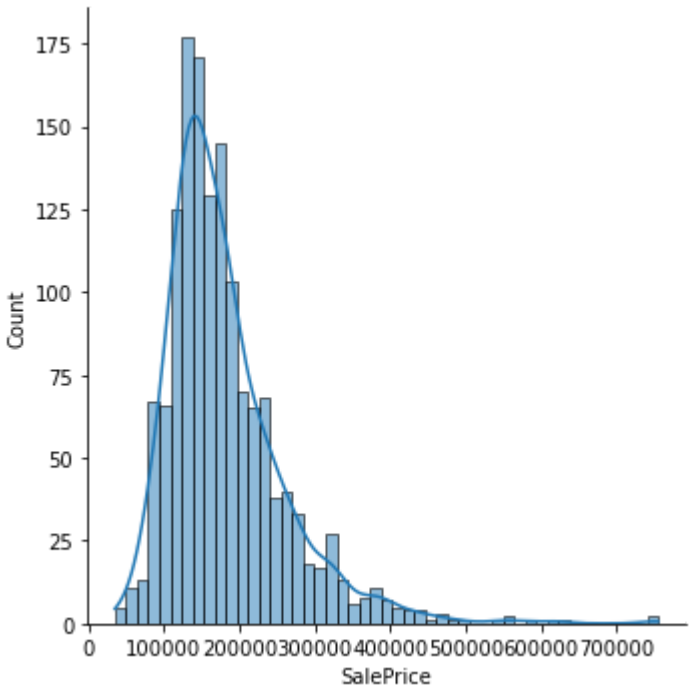
In [14]:
```
sns.displot(data=df,x='SalePrice',kind='kde')
```

Out[14]:
```
<seaborn.axisgrid.FacetGrid at 0x7fabea1a7c10>
```



In [15]:
```
sns.displot(data=df,x='SalePrice',kde=True)
```

Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x7fabea38aee0>
```



Heatmap

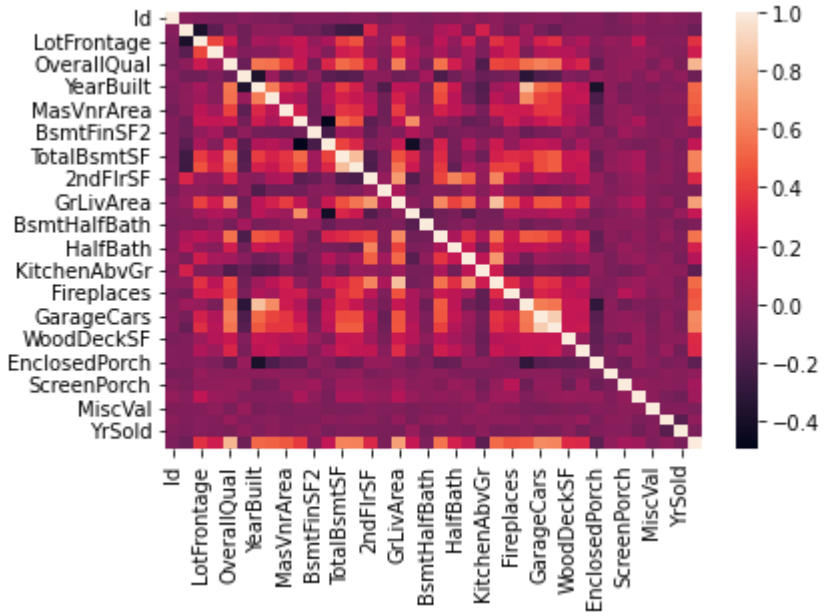
seaborn.heatmap(data, *, vmin=None, vmax=None, cmap=None, center=None, robust=False, annot=None, fmt='.2g', annot_kws=None, linewidths=0, linecolor='white', cbar=True, cbar_kws=None, cbar_ax=None, square=False, xticklabels='auto', yticklabels='auto', mask=None, ax=None, **kwargs)

The imput data of heatmap must be a 2 dimensional array.

Link for more details: https://seaborn.pydata.org/generated/seaborn.heatmap.html

In [16]:
```
corrmat = df.corr()
```

```
sns.heatmap(corrmat);
```



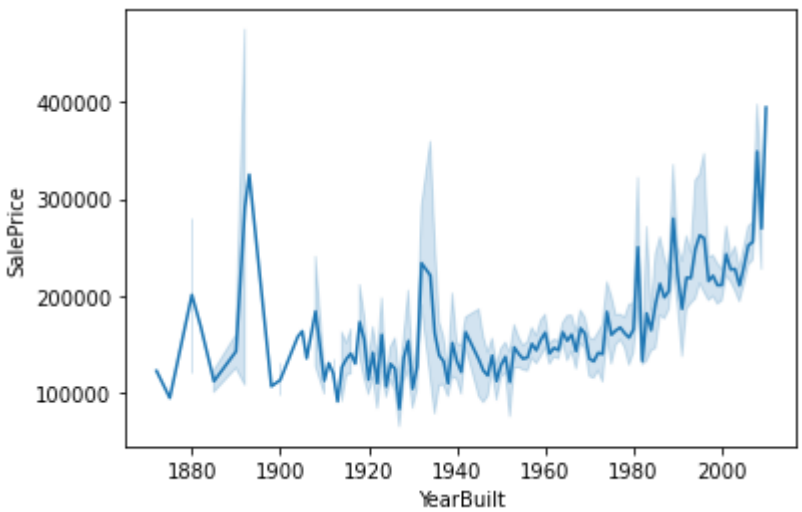
Line plot

seaborn.lineplot(data=None, *, x=None, y=None, hue=None, size=None, style=None, units=None, palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None, size_norm=None, dashes=True, markers=None, style_order=None, estimator='mean', errorbar=('ci', 95), n_boot=1000, seed=None, orient='x', sort=True, err_style='band', err_kws=None, legend='auto', ci='deprecated', ax=None, **kwargs)

Link for more details: https://seaborn.pydata.org/generated/seaborn.lineplot.html

In [17]:
```
sns.lineplot(data = df, x = "YearBuilt", y = "SalePrice")
```

Out[17]:
```
<AxesSubplot:xlabel='YearBuilt', ylabel='SalePrice'>
```



Regression Model

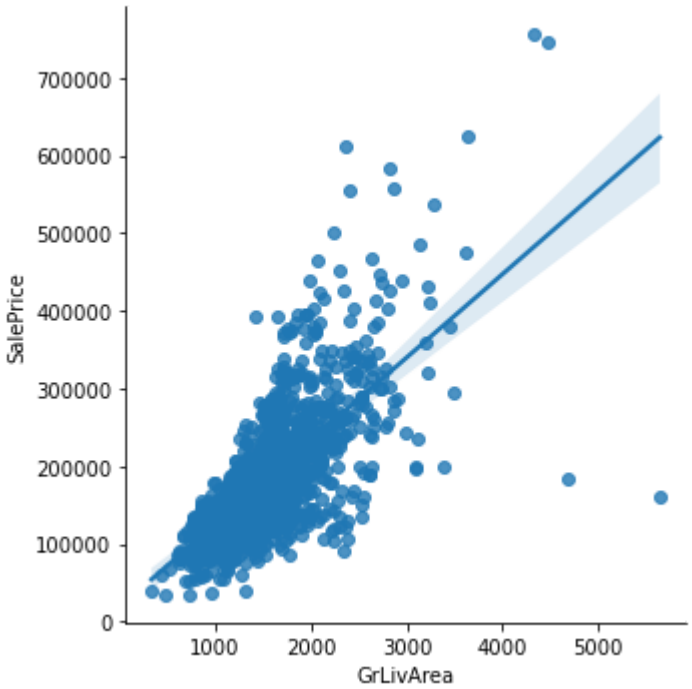
seaborn.lmplot(data=None, *, x=None, y=None, hue=None, col=None, row=None, palette=None, col_wrap=None, height=5, aspect=1, markers='o', sharex=None, sharey=None, hue_order=None, col_order=None, row_order=None, legend=True, legend_out=None, x_estimator=None, x_bins=None, x_ci='ci', scatter=True,

fit_reg=True, ci=95, n_boot=1000, units=None, seed=None, order=1, logistic=False, lowess=False, robust=False, logx=False, x_partial=None, y_partial=None, truncate=True, x_jitter=None, y_jitter=None, scatter_kws=None, line_kws=None, facet_kws=None)

Link for more details: https://seaborn.pydata.org/generated/seaborn.lmplot.html

In [18]:
```
sns.lmplot(data=df,x='GrLivArea',y='SalePrice')
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x7fabeb790040>
```



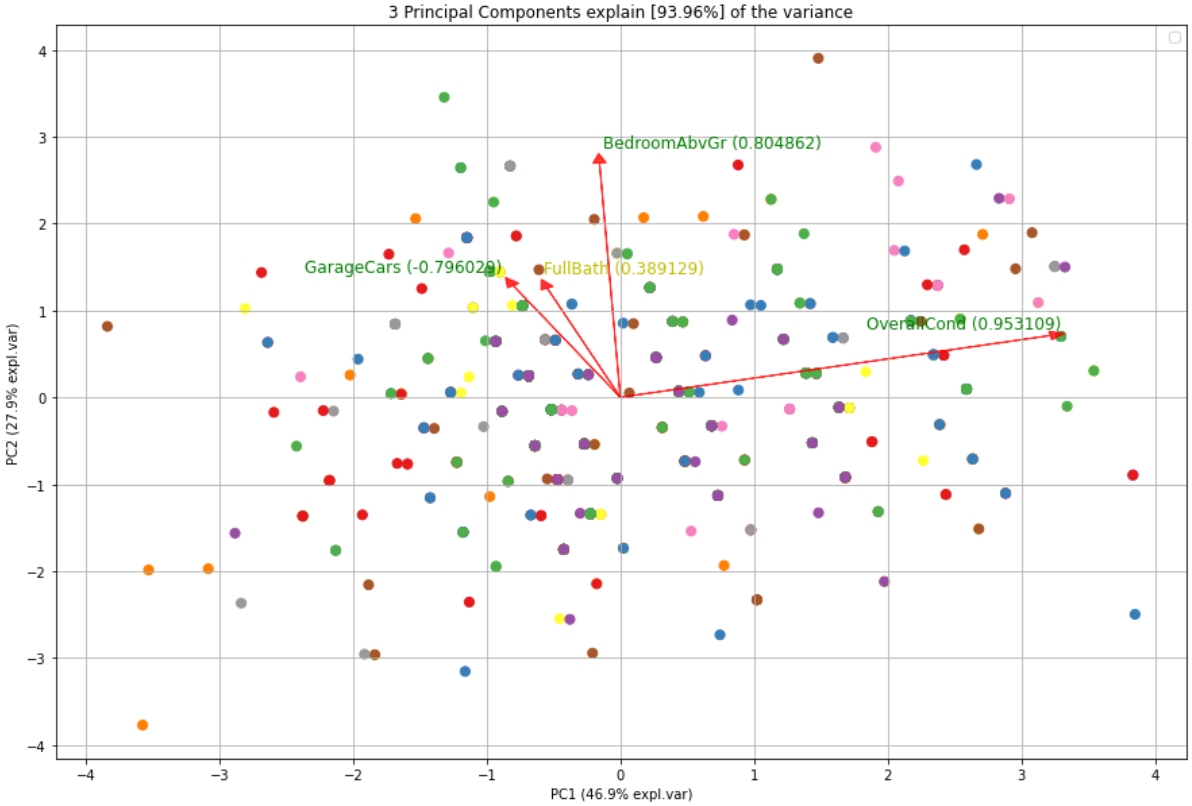
The following plots cannot be done by the seaborn itself. They need help from other packages. Since we have learned how to draw these graphs using R, we believe it will be helpful to understand how to visualize them using Python.

Biplot

Link for more details:https://stackoverflow.com/questions/39216897/plot-pca-loadings-and-loading-in-biplot-in-sklearn-like-rs-autoplot

In [19]:
```
X_numerical = df[["GarageCars", "BedroomAbvGr", "FullBath", "OverallCond"]]
X_numerical = X_numerical.dropna()
pca_model = pca(n_components = 3) # new model explains 90% variance
predicted_Y = pca_model.fit_transform(X_numerical)
pca_model.biplot(n_feat=4, label=None)
```

```
[pca] >Processing dataframe..
[pca] >The PCA reduction is performed on the [4] columns of the input datafr
ame.
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Compute explained variance.
[pca] >Outlier detection using Hotelling T2 test with alpha=[0.05] and n_com
ponents=[3]
[pca] >Outlier detection using SPE/DmodX with n_std=[2]
No artists with labels found to put in legend.  Note that artists whose labe
l start with an underscore are ignored when legend() is called with no argum
ent.
[pca] >Plot PC1 vs PC2 with loadings.
[colourmap]> Warning: Colormap [Set1] can not create [1460] unique colors! A
vailable unique colors: [9].
[colourmap]> Warning: Colormap [Set1] can not create [1460] unique colors! A
vailable unique colors: [9].
```



Out[19]:
```
(<Figure size 1080x720 with 1 Axes>,
 <AxesSubplot:title={'center':'3 Principal Components explain [93.96%] of th
e variance'}, xlabel='PC1 (46.9% expl.var)', ylabel='PC2 (27.9% expl.var)'>)
```

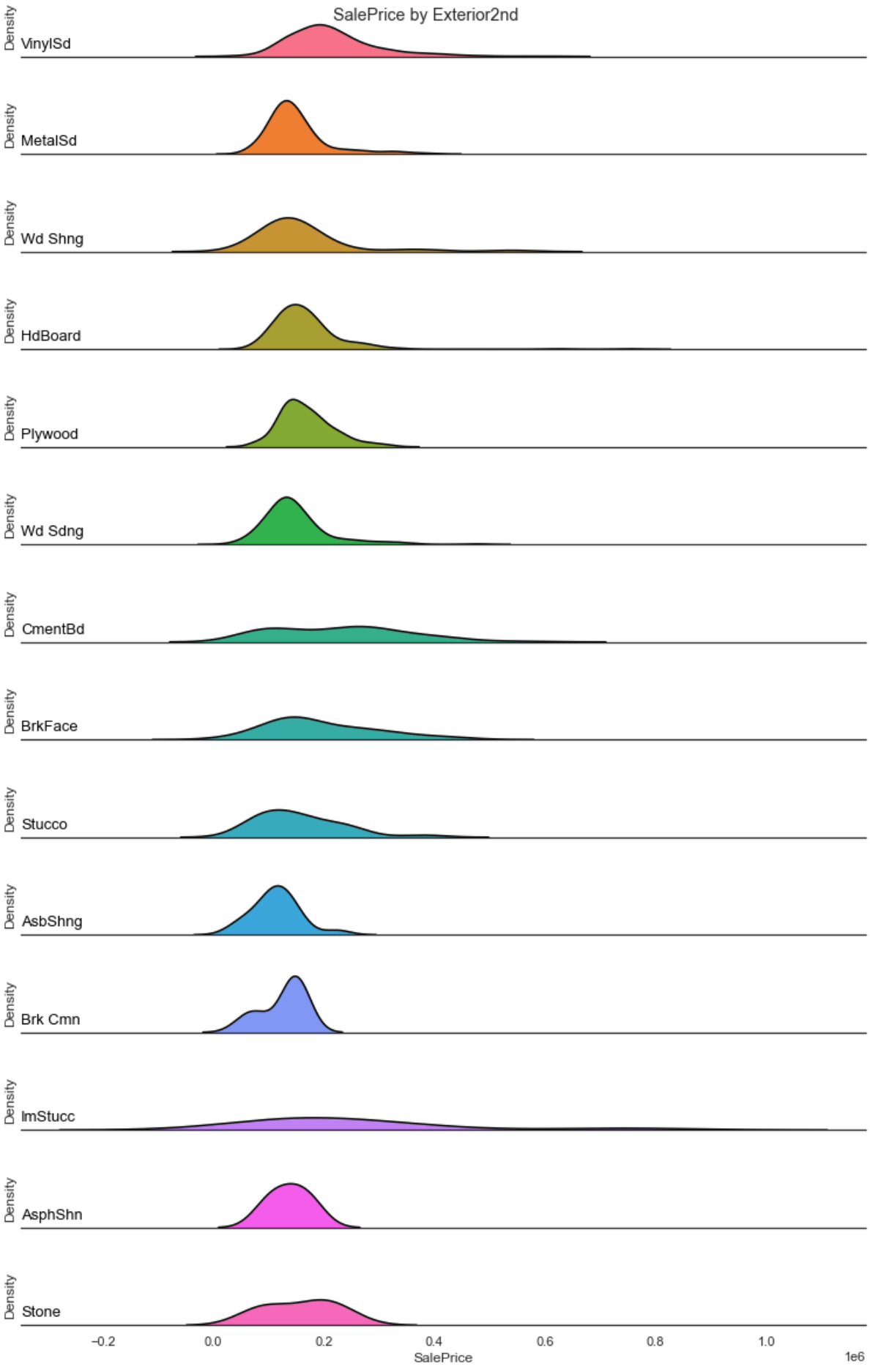
Ridge Plot

Link for more details: https://python.plainenglish.io/ridge-plots-with-pythons-seaborn-4de5725881af

In [20]:
```
sns.set_theme(style="white", rc={"axes.facecolor": (0, 0, 0, 0)})
# choose data from the dataset
sns.set_theme(style="white", rc={"axes.facecolor": (0, 0, 0, 0)})


# choose data from the dataset
df_ridge = df[["Exterior2nd", "SalePrice"]]
df_ridge = df_ridge[df_ridge["Exterior2nd"] != "CBlock"]
df_ridge = df_ridge[df_ridge["Exterior2nd"] != "Other"]
sns.set_theme(style="white")
g = sns.FacetGrid(df_ridge, row = "Exterior2nd", hue = "Exterior2nd", aspect
```

```
g.map_dataframe(sns.kdeplot, x = "SalePrice", fill=True, alpha = 1)
g.map_dataframe(sns.kdeplot, x = "SalePrice", color = 'black')

def label(x ,color, label):
     ax = plt.gca()
     ax.text(0, 0.2, label, color = 'black', fontsize = 13, ha = "left", va

g.map(label, "Exterior2nd")


g.set_titles("")
g.set(yticks=[], xlabel = "SalePrice")
g.despine(left=True)

plt.suptitle('SalePrice by Exterior2nd', y = 0.97)
```

Out[20]:
```
Text(0.5, 0.97, 'SalePrice by Exterior2nd')
```


SalePrice by Exterior2nd
Density
VinylSd
MetalSd
Wd Shng
HdBoard
Plywood
Wd Sdng
CmentBd
BrkFace
Stucco
AsbShng
Brk Cmn
ImStucc
AsphShn
Stone
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
SalePrice
1e6

In [ ]: