



# Fourth Industrial Summer School

Day 2: Data Analysis Foundations - Morning

## Exploratory Data Analysis

# Session Objectives

- ✓ Seaborn
  - heatmap
  - Correlation matrix
- ✓ Checking missing data
- ✓ Removing features with missing data
- ✓ Exploring
  - numeric data
  - categorical data



# Exploratory Data Analysis (EDA)



- To give insight into your data
- Understand the underlying structure
- Extract important features and relationships
- Generate Hypothesis

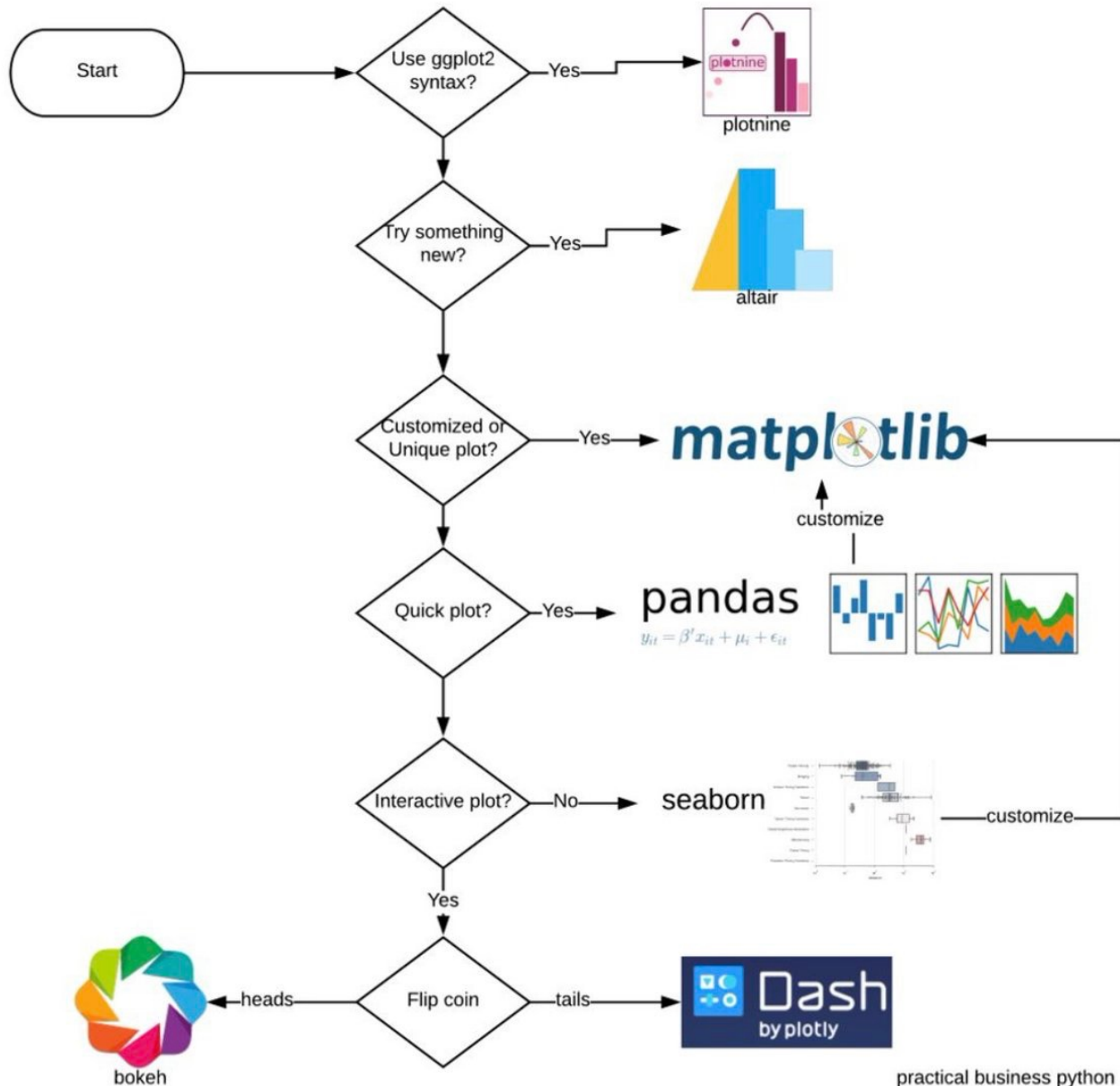
# Seaborn



- **Seaborn** is a Python data visualization library based on matplotlib.
- It provides a high-level interface for drawing attractive and informative statistical graphics.
- It introduces additional plot types.
- It also makes your traditional Matplotlib plots look a bit prettier.
- Similar (in style) to the popular ggplot2 library in R

Link: <https://seaborn.pydata.org/>

# Choosing a python visualization tool



# Data Exploration example

- Employee2.csv
  - Company employees information
  - Used before as Employee.csv, but with some missing data

emp_id	Gender	Age	Sales	BMI	Income
1	M	34	123	Normal	350
2	F	40	114	Overweight	450
3	F	37	135	Obesity	169
4	M	30	139	Underweight	
5	F	44	117	Underweight	183
6	M	36	121	Normal	80
7	M		133	Obesity	166
8	F	26	140	Normal	120
9		32	133	Normal	75
10	M	36	133	Underweight	40

# Import

- Import libraries

```
1 import pandas as pd
2 import seaborn as sns
3 sns.set()
4 import matplotlib.pyplot as plt
```

- Notice the missing data

```
1 df = pd.read_csv('Employee2.csv')
2 df.head()
```

	emp_id	Gender	Age	Sales	BMI	Income
0	1	M	34.0	123	Normal	350.0
1	2	F	40.0	114	Overweight	450.0
2	3	F	37.0	135	Obesity	169.0
3	4	M	30.0	139	Underweight	NaN
4	5	F	44.0	117	Underweight	183.0

# Data Shape

## ■ Data shape

- 10 rows
- 6 columns

```
1 df.shape
```

```
(10, 6)
```

## ■ Data information

- Check data types

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10 entries, 0 to 9  
Data columns (total 6 columns):  
emp_id      10 non-null int64  
Gender      9 non-null object  
Age         9 non-null float64  
Sales       10 non-null int64  
BMI         10 non-null object  
Income      9 non-null float64  
dtypes: float64(2), int64(2), object(2)  
memory usage: 560.0+ bytes
```



# Descriptive Statistics

- Specific variable statistics

```
1 df.Income.describe()
```

```
count      9.000000
mean     181.444444
std     135.246175
min      40.000000
25%      80.000000
50%     166.000000
75%     183.000000
max     450.000000
Name: Income, dtype: float64
```

# Missing Values

- Check missing values

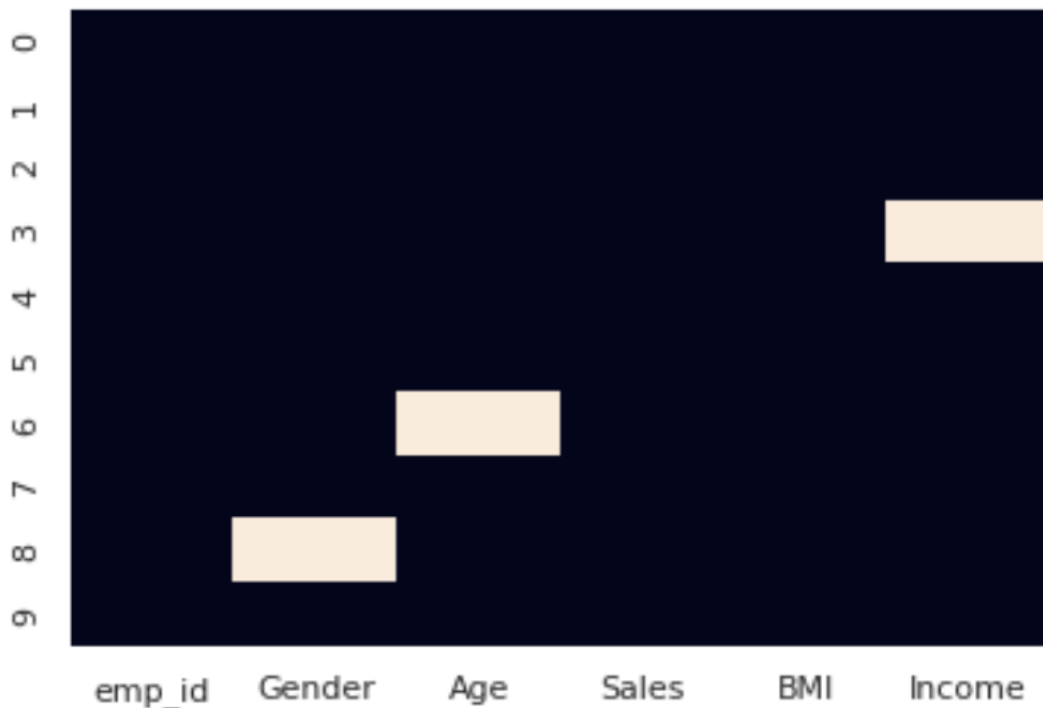
```
1 df.isnull().sum()
```

```
emp_id    0  
Gender    1  
Age       1  
Sales     0  
BMI       0  
Income    1  
dtype: int64
```

# Visualize Missing Values

- Use heatmap to visualize the missing values

```
1 sns.heatmap(df.isnull(), cbar=False)
```



# Handle Missing Data

- Fill missing **numerical** values with mean

```
1 df_num = df.select_dtypes(include=['number'])
2 df[df_num.columns] = df_num.fillna(df_num.mean())
```

- Fill missing **categorical** values with mode

```
1 df_cat = df.select_dtypes(exclude=['number'])
2 mode_values = df_cat.mode().iloc[0]
3 df[df_cat.columns] = df_cat.fillna(mode_values)
```

# Re-Visualize Missing Values

- Replot the heatmap

```
1 df = pd.concat([df_num, df_cat], axis=1)
2 sns.heatmap(df.isnull(), cbar=False)
```

<Axes: >



# Correlation Matrix

## ■ Correlation as a heatmap

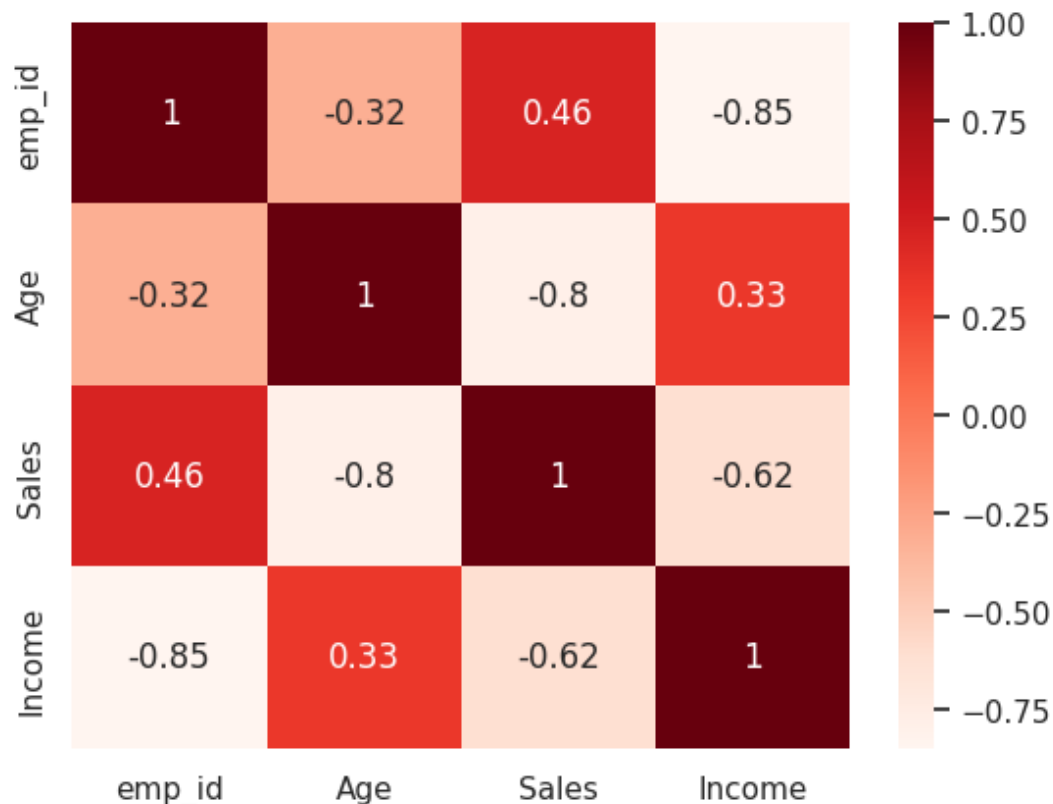
```
1 sns.heatmap(df_num.corr(), cmap='Reds')
```



# Correlation Matrix

- Correlation as a heatmap with correlation annotation

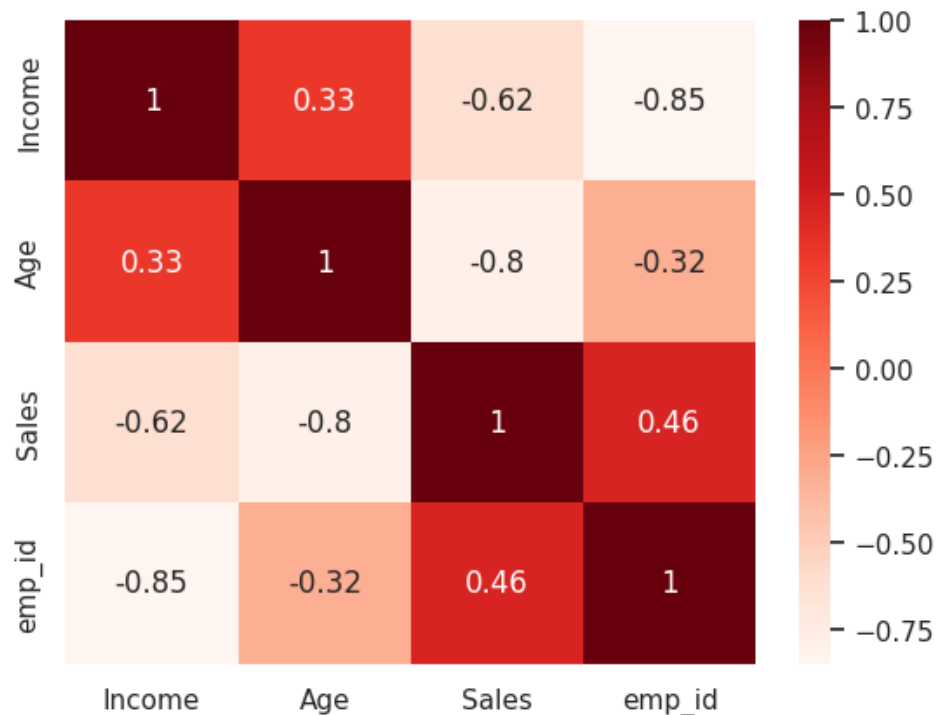
```
1 sns.heatmap(df_num.corr(), annot=True, cmap='Reds')
```



# Correlation Matrix

- Correlation matrix **sorted**
  - Calculate the top k correlations with 'Income'

```
1 k = len(df)
2 cols = df_num.corr().nlargest(k, 'Income')['Income'].index
3 cm = df_num[cols].corr()
4 sns.heatmap(cm, annot=True, cmap='Reds')
```





# Data Exploration session

## House Sales

# House Sales Data Exploration

## ■ HouseSales.csv

```
1 df = pd.read_csv('HousePrices.csv')
2 df.head()
```

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

5 rows x 81 columns

# House Sales Data Exploration

## ■ Data information

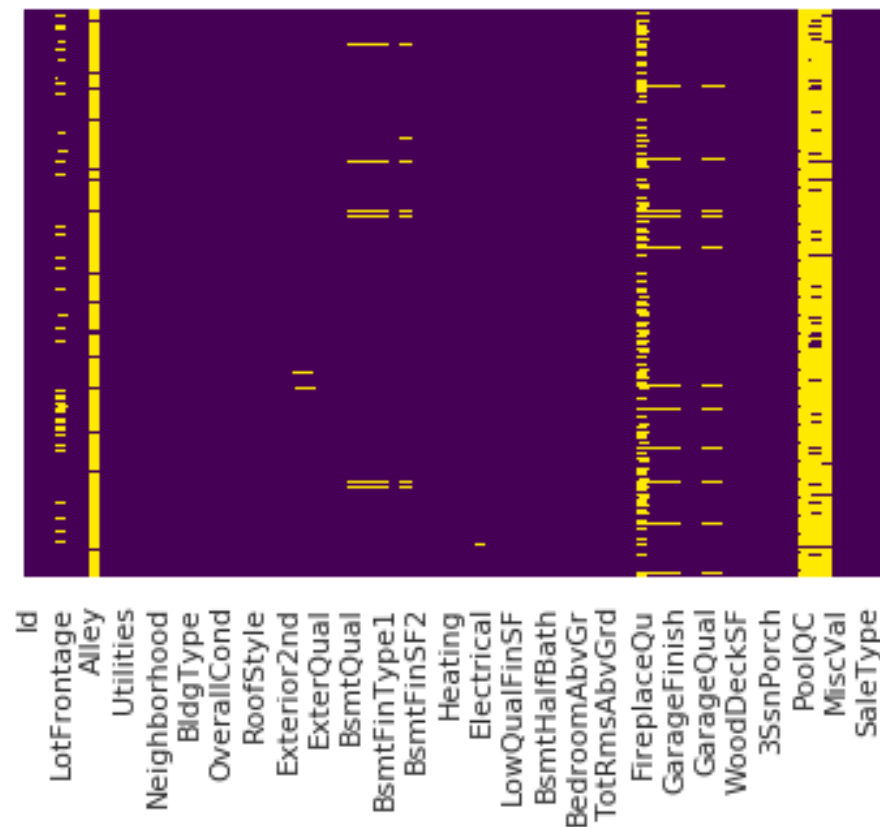
```
1 df.info()
```

```
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
Id                1460 non-null int64
MSSubClass        1460 non-null int64
MSZoning          1460 non-null object
LotFrontage       1201 non-null float64
LotArea           1460 non-null int64
Street            1460 non-null object
Alley             91 non-null object
LotShape          1460 non-null object
LandContour       1460 non-null object
Utilities         1460 non-null object
LotConfig         1460 non-null object
LandSlope         1460 non-null object
Neighborhood      1460 non-null object
Condition1        1460 non-null object
Condition2        1460 non-null object
BldgType          1460 non-null object
HouseStyle        1460 non-null object
```

# House Sales Data Exploration

- Investigate missing data
  - Yellow color

```
1 sns.heatmap(df.isnull(), cbar=False, yticklabels=False, cmap='viridis')
```



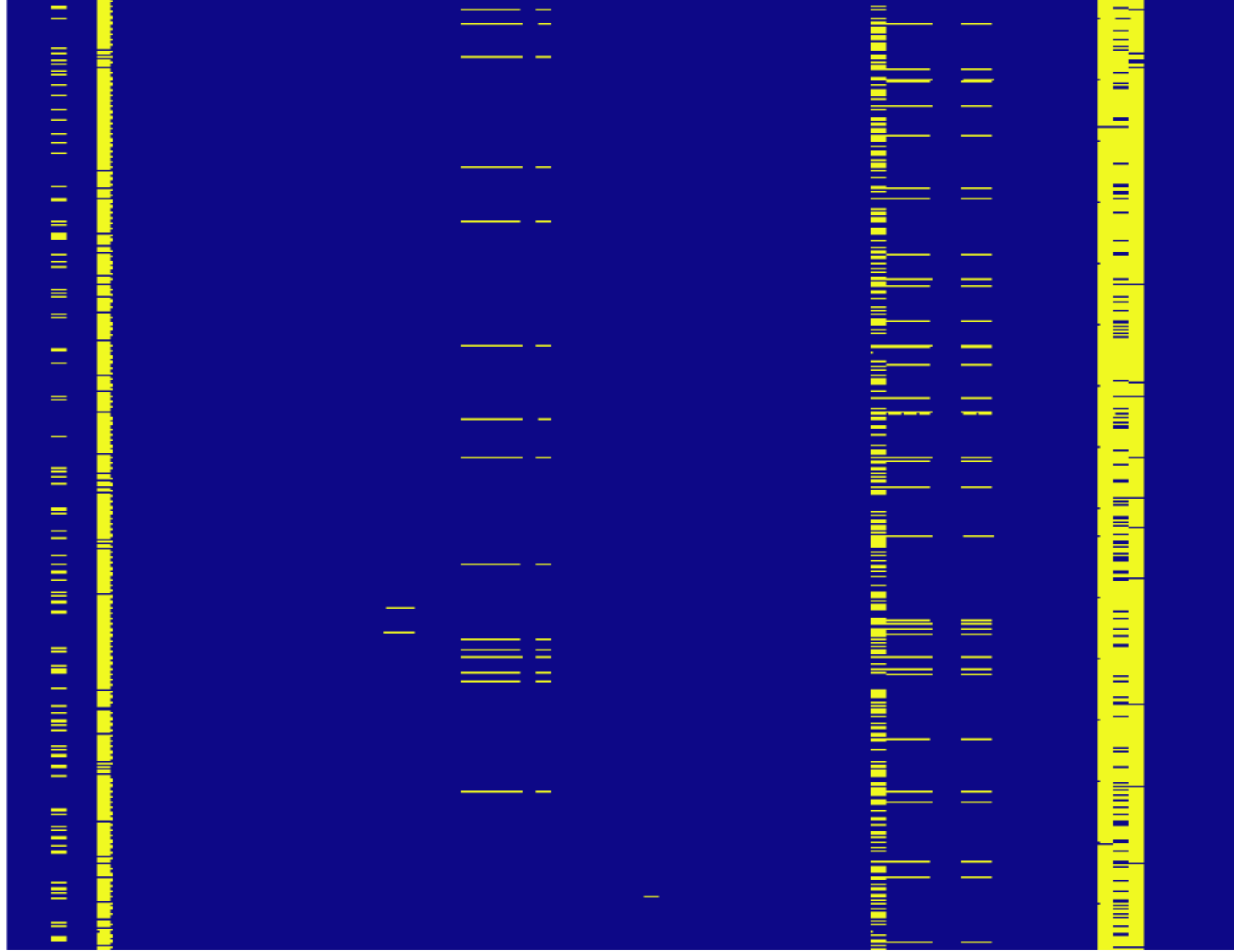
# House Sales Data Exploration

- Increase the figure size for better quality
- Try different colors

```
1 plt.subplots(figsize=(10,10))
2 sns.heatmap(df.isnull(), cbar=False, yticklabels=False, cmap='plasma')
```

[https://matplotlib.org/3.2.1/api/\\_as\\_gen/matplotlib.pyplot.subplots.html](https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.subplots.html)

Id  
 MSZoning  
 LotArea  
 Alley  
 LandContour  
 LotConfig  
 Neighborhood  
 Condition2  
 HouseStyle  
 OverallCond  
 YearRemodAdd  
 RoofMatl  
 Exterior2nd  
 MasVnrArea  
 ExterCond  
 BsmtQual  
 BsmtExposure  
 BsmtFinSF1  
 BsmtFinSF2  
 TotalBsmtSF  
 HeatingQC  
 Electrical  
 2ndFlrSF  
 GrLivArea  
 BsmtHalfBath  
 HalfBath  
 KitchenAbvGr  
 TotRmsAbvGrd  
 Fireplaces  
 GarageType  
 GarageFinish  
 GarageArea  
 GarageCond  
 WoodDeckSF  
 EnclosedPorch  
 ScreenPorch  
 PoolQC  
 MiscFeature  
 MoSold  
 SaleType  
 SalePrice



# House Sales Data Exploration



- What are your observations?
- Some features has many missing values
  - Should we keep them?
- Let us remove features with too many missing data

# House Sales Data Exploration

- Remove features with 50% or less missing data

```
1 df2 = df[[column for column in df if df[column].count() / len(df) >= 0.5]]
2 del df2['Id']
3 print("Features dropped:")
4 for c in df.columns:
5     if c not in df2.columns:
6         print(c)
7 df = df2
```

- Code to check which features were dropped

```
☞ Features dropped:
   Id
   Alley
   PoolQC
   Fence
   MiscFeature
```



# House Sales Data Exploration

- Replot heatmap to check missing values

```
1 plt.subplots(figsize=(10,10))  
2 sns.heatmap(df.isnull(), cbar=False, yticklabels=False, cmap='viridis')
```

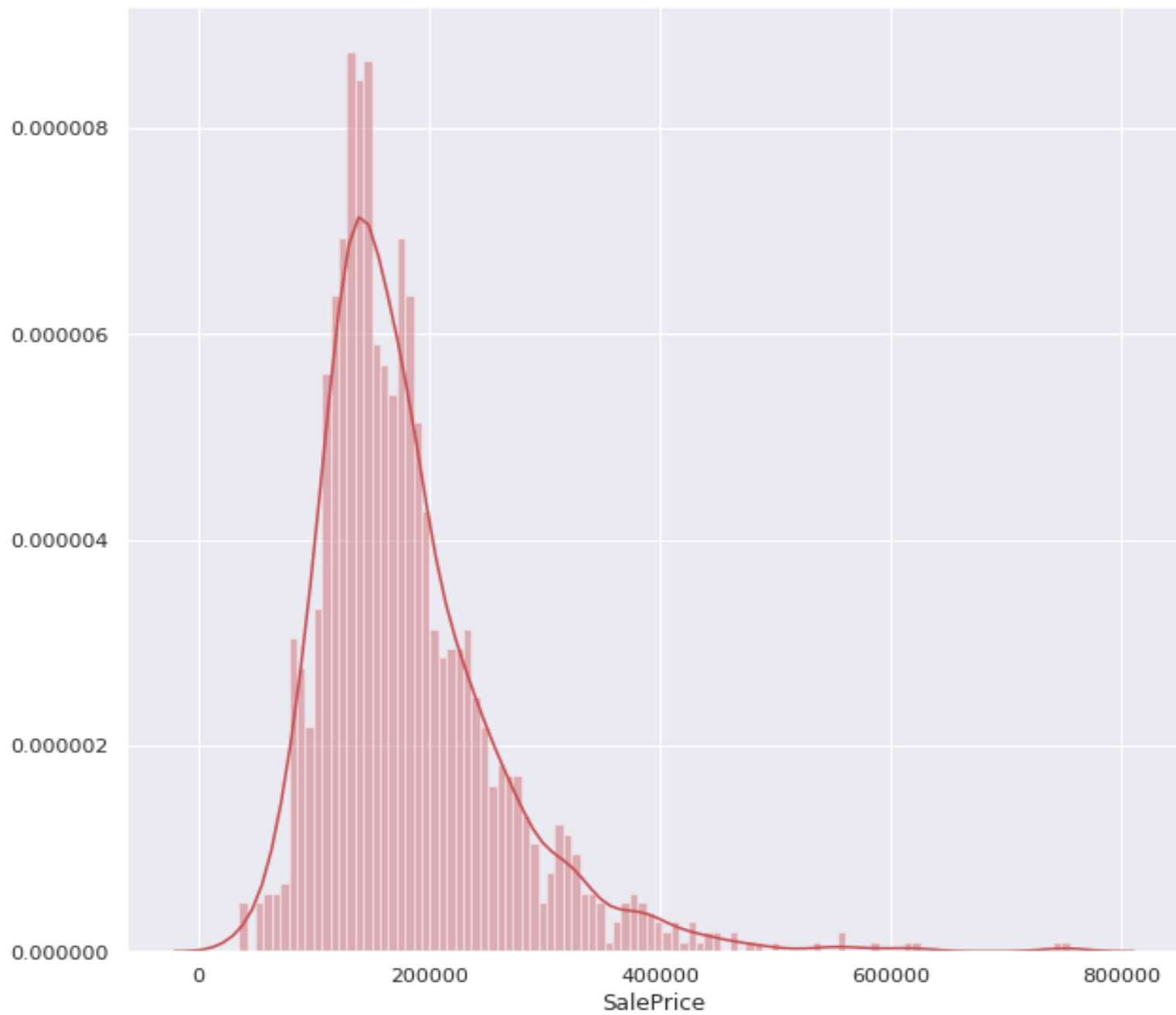


# House Sales Data Exploration

- Study the House sales distribution
  - Seaborn provides a nice histogram plot function
    - `displot`

```
1 print(df['SalePrice'].describe())
2 plt.figure(figsize=(10, 10))
3 sns.distplot(df['SalePrice'], color='r', bins=100)
```

```
↳ count      1460.000000
   mean      180921.195890
   std       79442.502883
   min       34900.000000
   25%      129975.000000
   50%      163000.000000
   75%      214000.000000
   max       755000.000000
   Name: SalePrice, dtype: float64
```



# House Sales Data Exploration



- Data contains both Numerical and Categorical data
- We will explore both types in the next slides

# House Sales Data Exploration

- Numerical data exploration
- First, we need to filter the data based on their type
  - int and float

```
1 df_num = df.select_dtypes(include = ['float64', 'int64'])
```

- New dataframe with only numeric type
- Check how many features were retrieved.

# House Sales Data Exploration

- Explore all the features using histograms

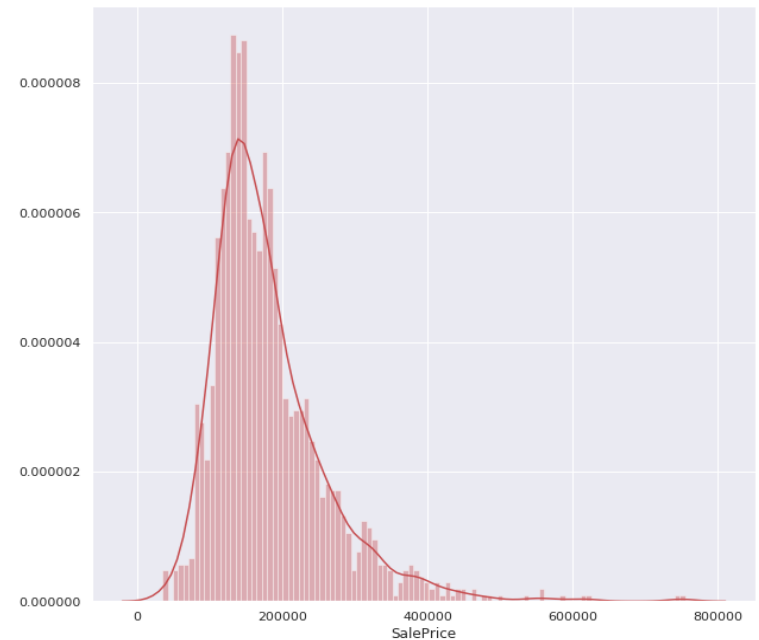
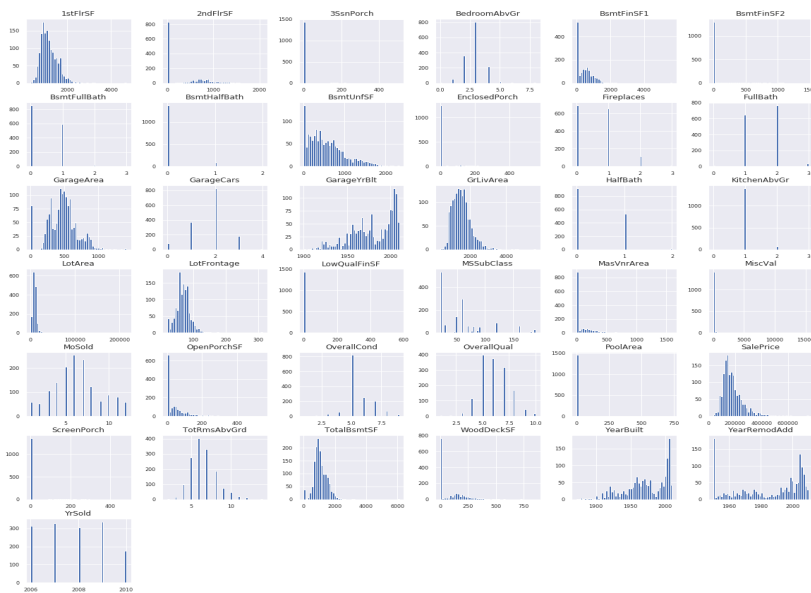
```
1 df_num.hist(figsize=(20, 20), bins=50, xlabelsize=8, ylabelsize=8)
```





# House Sales Data Exploration

- Which features histograms shares similar distribution as Sales ??
  - e.g. 1stFlrSF feature



# House Sales Data Exploration

- Next study correlation between features and Sales Price
  - Focus on high correlation with  $> 0.5$  score

```
1 df_num_corr = df_num.corr()['SalePrice']
2 selectedFeatures=df_num_corr[abs(df_num_corr)>0.5].sort_values(ascending=False)
3 print(selectedFeatures)
```

```
SalePrice      1.000000
OverallQual    0.790982
GrLivArea      0.708624
GarageCars     0.640409
GarageArea     0.623431
TotalBsmtSF    0.613581
1stFlrSF       0.605852
FullBath       0.560664
TotRmsAbvGrd   0.533723
YearBuilt      0.522897
YearRemodAdd   0.507101
Name: SalePrice, dtype: float64
```

- Do you think correlation is effected by outliers?
  - Make a scatter plot to observe the relationships

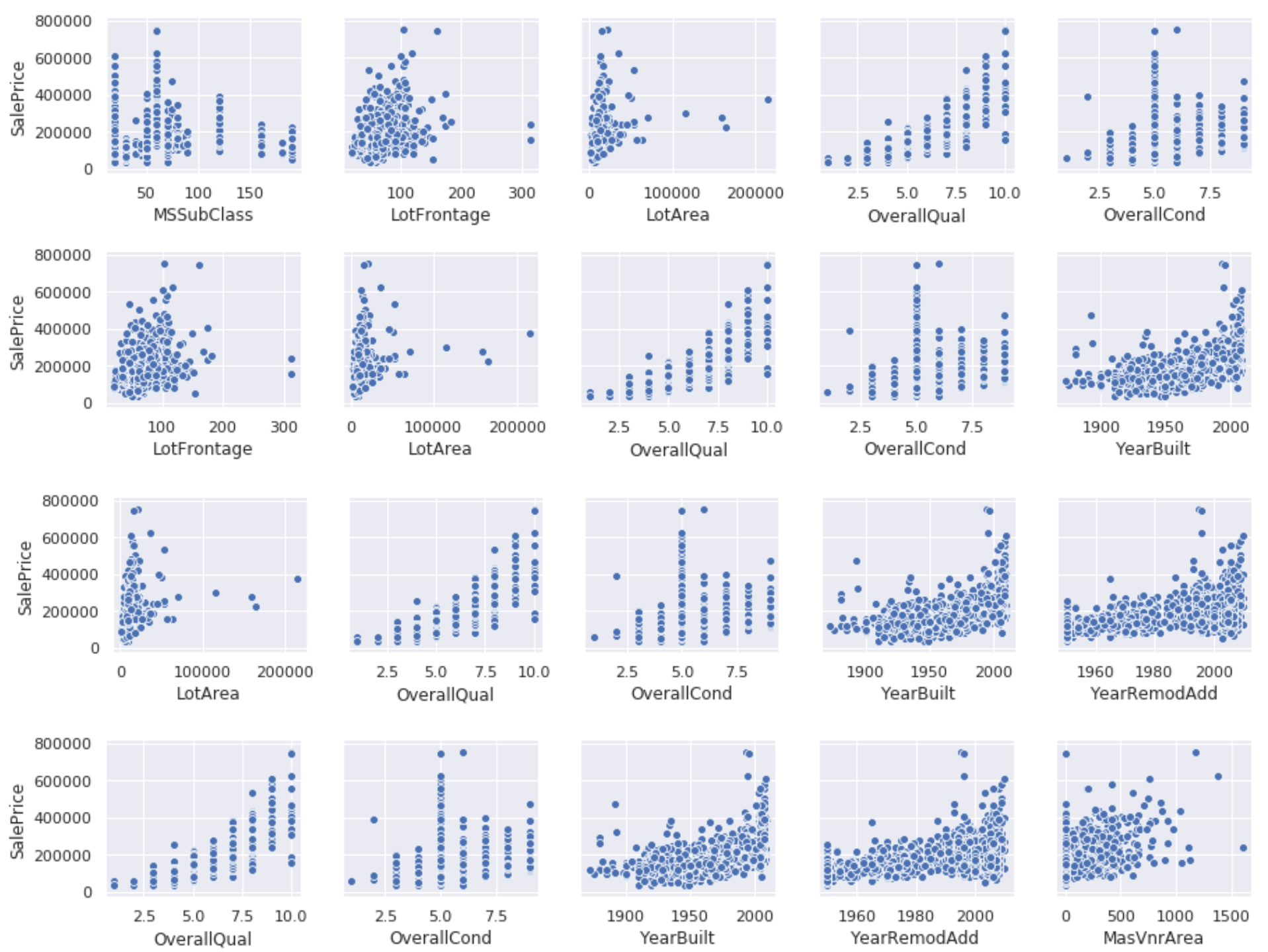
Break

# House Sales Data Exploration

- Plot all features scatter plot
  - `pairplot`

```
1 for i in range(0, len(df_num.columns), 5):  
2     | |     sns.pairplot(data=df_num, x_vars=df_num.columns[i:i+5], y_vars=['SalePrice'])
```

- check if our correlated values have a **linear relationship** to the Sale Price.



# House Sales Data Exploration

- Plotting all the numerical features in a seaborn pairplot will be hard to interpret.
  - **Heatmap**

```
1 corr = df_num.corr()  
2 plt.figure(figsize=(24,20))  
3 sns.heatmap(corr, annot=True)
```



MSSubClass	1	-0.39	-0.14	0.033	-0.059	0.028	0.041	0.023	-0.07	-0.066	-0.14	-0.24	-0.25	0.31	0.046	0.075	0.0035	-0.0023	0.13	0.18	-0.023	0.28	0.04	-0.046	0.085	-0.04	-0.099	-0.013	-0.0061	-0.012	-0.044	-0.026	0.0083	-0.0077	-0.014	-0.021	-0.084
LotFrontage	-0.39	1	0.43	0.25	-0.059	0.12	0.089	0.19	0.23	0.05	0.13	0.39	0.46	0.08	0.038	0.4	0.1	-0.0072	0.2	0.054	0.26	-0.0061	0.35	0.27	0.07	0.29	0.34	0.089	0.15	0.011	0.07	0.041	0.21	0.0034	0.011	0.0074	0.35
LotArea	-0.14	0.43	1	0.11	-0.0056	0.014	0.014	0.1	0.21	0.11	-0.0026	0.26	0.3	0.051	0.0048	0.26	0.16	0.048	0.13	0.014	0.12	-0.018	0.19	0.27	-0.025	0.15	0.18	0.17	0.085	-0.018	0.02	0.043	0.078	0.038	0.0012	-0.014	0.26
OverallQual	0.033	0.25	0.11	1	-0.092	0.57	0.55	0.41	0.24	-0.059	0.31	0.54	0.48	0.3	-0.03	0.59	0.11	-0.04	0.55	0.27	0.1	-0.18	0.43	0.4	0.55	0.6	0.56	0.24	0.31	-0.11	0.03	0.065	0.065	-0.031	0.071	-0.027	0.79
OverallCond	-0.059	-0.059	-0.0056	-0.092	1	-0.38	0.074	-0.13	-0.046	0.04	-0.14	-0.17	-0.14	0.029	0.025	-0.08	-0.055	0.12	-0.19	-0.061	0.013	-0.087	-0.058	-0.024	-0.32	-0.19	-0.15	-0.0033	-0.033	0.07	0.026	0.055	-0.002	0.069	-0.0035	0.044	-0.078
YearBuilt	0.028	0.12	0.014	0.57	-0.38	1	0.59	0.32	0.25	-0.049	0.15	0.39	0.28	0.01	-0.18	0.2	0.19	-0.038	0.47	0.24	-0.071	-0.17	0.096	0.15	0.83	0.54	0.48	0.22	0.19	-0.39	0.031	-0.05	0.0049	-0.034	0.012	-0.014	0.52
YearRemodAdd	0.041	0.089	0.014	0.55	0.074	0.59	1	0.18	0.13	-0.068	0.18	0.29	0.24	0.14	-0.062	0.29	0.12	-0.012	0.44	0.18	-0.041	-0.15	0.19	0.11	0.64	0.42	0.37	0.21	0.23	-0.19	0.045	-0.039	0.0058	-0.01	0.021	0.036	0.51
MasVnrArea	0.023	0.19	0.1	0.41	-0.13	0.32	0.18	1	0.26	-0.072	0.11	0.36	0.34	0.17	-0.069	0.39	0.085	0.027	0.28	0.2	0.1	-0.038	0.28	0.25	0.25	0.36	0.37	0.16	0.13	-0.11	0.019	0.061	0.012	-0.03	-0.006	-0.0082	0.48
BsmtFinSF1	-0.07	0.23	0.21	0.24	-0.046	0.25	0.13	0.26	1	-0.05	-0.5	0.52	0.45	-0.14	-0.065	0.21	0.65	0.067	0.059	0.0043	-0.11	-0.081	0.044	0.26	0.15	0.22	0.3	0.2	0.11	-0.1	0.026	0.062	0.14	0.0036	-0.016	0.014	0.39
BsmtFinSF2	-0.066	0.05	0.11	-0.059	0.04	-0.049	-0.068	-0.072	-0.05	1	-0.21	0.1	0.097	-0.099	0.015	-0.0096	0.16	0.071	-0.076	-0.032	-0.016	-0.041	-0.035	0.047	-0.088	-0.038	-0.018	0.068	0.0031	0.037	-0.03	0.089	0.042	0.0049	-0.015	0.032	-0.011
BsmtUnfSF	-0.14	0.13	-0.0026	0.31	-0.14	0.15	0.18	0.11	-0.5	-0.21	1	0.42	0.32	0.0045	0.028	0.24	-0.42	-0.096	0.29	-0.041	0.17	0.03	0.25	0.052	0.19	0.21	0.18	-0.0053	0.13	-0.0025	0.021	-0.013	-0.035	-0.024	0.035	-0.041	0.21
TotalBsmtSF	-0.24	0.39	0.26	0.54	-0.17	0.39	0.29	0.36	0.52	0.1	0.42	1	0.82	-0.17	-0.033	0.45	0.31	-0.00031	0.32	-0.049	0.05	-0.069	0.29	0.34	0.32	0.43	0.49	0.23	0.25	-0.095	0.037	0.084	0.13	-0.018	0.013	-0.015	0.61
1stFlrSF	-0.25	0.46	0.3	0.48	-0.14	0.28	0.24	0.34	0.45	0.097	0.32	0.82	1	-0.2	-0.014	0.57	0.24	0.002	0.38	-0.12	0.13	0.068	0.41	0.41	0.23	0.44	0.49	0.24	0.21	-0.065	0.056	0.089	0.13	-0.021	0.031	-0.014	0.61
2ndFlrSF	0.31	0.08	0.051	0.3	0.029	0.01	0.14	0.17	-0.14	-0.099	0.0045	-0.17	-0.2	1	0.063	0.69	-0.17	-0.024	0.42	0.61	0.5	0.059	0.62	0.19	0.071	0.18	0.14	0.092	0.21	0.062	-0.024	0.041	0.081	0.016	0.035	-0.029	0.32
LowQualFinSF	0.046	0.038	0.0048	-0.03	0.025	-0.18	-0.062	-0.069	-0.065	0.015	0.028	-0.033	-0.014	0.063	1	0.13	-0.047	-0.00580	0.00710	0.027	0.11	0.0075	0.13	-0.021	-0.036	-0.094	-0.068	-0.025	0.018	0.061	-0.0043	0.027	0.062	-0.0038	-0.022	-0.029	-0.026
GrLivArea	0.075	0.4	0.26	0.59	-0.08	0.2	0.29	0.39	0.21	-0.0096	0.24	0.45	0.57	0.69	0.13	1	0.035	-0.019	0.63	0.42	0.52	0.1	0.83	0.46	0.23	0.47	0.47	0.25	0.33	0.0091	0.021	0.1	0.17	-0.0024	0.05	-0.037	0.71
BsmtFullBath	0.0035	0.1	0.16	0.11	-0.055	0.19	0.12	0.085	0.65	0.16	-0.42	0.31	0.24	-0.17	-0.047	0.035	1	-0.15	-0.065	-0.031	-0.15	-0.042	-0.053	0.14	0.12	0.13	0.18	0.18	0.067	-0.05	-0.00011	0.023	0.068	-0.023	-0.025	0.067	0.23
BsmtHalfBath	-0.00230	0.072	0.048	-0.04	0.12	-0.038	-0.012	0.027	0.067	0.071	-0.096	0.00310	0.002	-0.024	-0.0058	-0.019	-0.15	1	-0.055	-0.012	0.047	-0.038	-0.024	0.029	-0.077	-0.021	-0.025	0.04	-0.025	-0.0086	0.035	0.032	0.02	-0.0074	0.033	-0.047	-0.017
FullBath	0.13	0.2	0.13	0.55	-0.19	0.47	0.44	0.28	0.059	-0.076	0.29	0.32	0.38	0.42	-0.0007	0.63	-0.065	-0.055	1	0.14	0.36	0.13	0.55	0.24	0.48	0.47	0.41	0.19	0.26	-0.12	0.035	-0.0081	0.05	-0.014	0.056	-0.02	0.56
HalfBath	0.18	0.054	0.014	0.27	-0.061	0.24	0.18	0.2	0.0043	-0.032	-0.041	-0.049	-0.12	0.61	-0.027	0.42	-0.031	-0.012	0.14	1	0.23	-0.068	0.34	0.2	0.2	0.22	0.16	0.11	0.2	-0.095	-0.005	0.072	0.022	0.0013	-0.009	-0.01	0.28
BedroomAbvGr	-0.023	0.26	0.12	0.1	0.013	-0.071	-0.041	0.1	-0.11	-0.016	0.17	0.05	0.13	0.5	0.11	0.52	-0.15	0.047	0.36	0.23	1	0.2	0.68	0.11	-0.065	0.086	0.065	0.047	0.094	0.042	-0.024	0.044	0.071	0.0078	0.047	-0.036	0.17
KitchenAbvGr	0.28	-0.0061	-0.018	-0.18	-0.087	-0.17	-0.15	-0.038	-0.081	-0.041	0.03	-0.069	0.068	0.059	0.0075	0.1	-0.042	-0.038	0.13	-0.068	0.2	1	0.26	-0.12	-0.12	-0.051	-0.064	-0.09	-0.07	0.037	-0.025	-0.052	-0.015	0.062	0.027	0.032	-0.14
TotRmsAbvGrd	0.04	0.35	0.19	0.43	-0.058	0.096	0.19	0.28	0.044	-0.035	0.25	0.29	0.41	0.62	0.13	0.83	-0.053	-0.024	0.55	0.34	0.68	0.26	1	0.33	0.15	0.36	0.34	0.17	0.23	0.0042	-0.0067	0.059	0.084	0.025	0.037	-0.035	0.53
Fireplaces	-0.046	0.27	0.27	0.4	-0.024	0.15	0.11	0.25	0.26	0.047	0.052	0.34	0.41	0.19	-0.021	0.46	0.14	0.029	0.24	0.2	0.11	-0.12	0.33	1	0.047	0.3	0.27	0.2	0.17	-0.025	0.011	0.18	0.095	0.0014	0.046	-0.024	0.47
GarageYrBlt	0.085	0.07	-0.025	0.55	-0.32	0.83	0.64	0.25	0.15	-0.088	0.19	0.32	0.23	0.071	-0.036	0.23	0.12	-0.077	0.48	0.2	-0.065	-0.12	0.15	0.047	1	0.59	0.56	0.22	0.23	-0.3	0.024	-0.075	-0.015	-0.032	0.0053	-0.001	0.49
GarageCars	-0.04	0.29	0.15	0.6	-0.19	0.54	0.42	0.36	0.22	-0.038	0.21	0.43	0.44	0.18	-0.094	0.47	0.13	-0.021	0.47	0.22	0.086	-0.051	0.36	0.3	0.59	1	0.88	0.23	0.21	-0.15	0.036	0.05	0.021	-0.043	0.041	-0.039	0.64
GarageArea	-0.099	0.34	0.18	0.56	-0.15	0.48	0.37	0.37	0.3	-0.018	0.18	0.49	0.49	0.14	-0.068	0.47	0.18	-0.025	0.41	0.16	0.065	-0.064	0.34	0.27	0.56	0.88	1	0.22	0.24	-0.12	0.035	0.051	0.061	-0.027	0.028	-0.027	0.62
WoodDeckSF	-0.013	0.089	0.17	0.24	-0.0033	0.22	0.21	0.16	0.2	0.068	-0.0053	0.23	0.24	0.092	-0.025	0.25	0.18	0.04	0.19	0.11	0.047	-0.09	0.17	0.2	0.22	0.23	0.22	1	0.059	-0.13	-0.033	-0.074	0.073	-0.0096	0.021	0.022	0.32
OpenPorchSF	-0.0061	0.15	0.085	0.31	-0.033	0.19	0.23	0.13	0.11	0.0031	0.13	0.25	0.21	0.21	0.018	0.33	0.067	-0.025	0.26	0.2	0.094	-0.07	0.23	0.17	0.23	0.21	0.24	0.059	1	-0.093	-0.0058	0.074	0.061	-0.019	0.071	-0.058	0.32
EnclosedPorch	-0.012	0.011	-0.018	-0.11	0.07	-0.39	-0.19	-0.11	-0.1	0.037	-0.0025	-0.095	-0.065	0.062	0.061	0.0091	-0.05	-0.0086	-0.12	-0.095	0.042	0.037	0.0042	-0.025	-0.3	-0.15	-0.12	-0.13	-0.093	1	-0.037	-0.083	0.054	0.018	-0.029	-0.0099	-0.13
3SsnPorch	-0.044	0.07	0.02	0.03	0.026	0.031	0.045	0.019	0.026	-0.03	0.021	0.037	0.056	-0.024	-0.0043	0.021	-0.00011	0.035	0.035	-0.005	-0.024	-0.025	-0.0067	0.011	0.024	0.036	0.035	-0.033	-0.0058	-0.037	1	-0.031	-0.0080	0.00035	0.029	0.019	0.045
ScreenPorch	-0.026	0.041	0.043	0.065	0.055	-0.05	-0.039	0.061	0.062	0.089	-0.013	0.084	0.089	0.041	0.027	0.1	0.023	0.032	-0.0081	0.072	0.044	-0.052	0.059	0.18	-0.075	0.05	0.051	-0.074	0.074	-0.083	-0.031	1	0.051	0.032	0.023	0.011	0.11
PoolArea	0.0083	0.21	0.078	0.065	-0.002	0.0049	0.0058	0.012	0.14	0.042	-0.035	0.13	0.13	0.081	0.062	0.17	0.068	0.02	0.05	0.022	0.071	-0.015	0.084	0.095	-0.015	0.021	0.061	0.073	0.061	0.054	-0.008	0.051	1	0.03	-0.034	-0.06	0.092
MiscVal	-0.00770	0.034	0.038	-0.031	0.069	-0.034	-0.01	-0.03	0.0036	0.0049	-0.024	-0.018	-0.021	0.016	-0.0038	-0.0024	-0.023	-0.0074	-0.014	0.0013	0.0078	0.062	0.025	0.0014	-0.032	-0.043	-0.027	-0.0096	-0.019	0.018	0.00035	0.032	0.03	1	-0.0065	0.0049	-0.

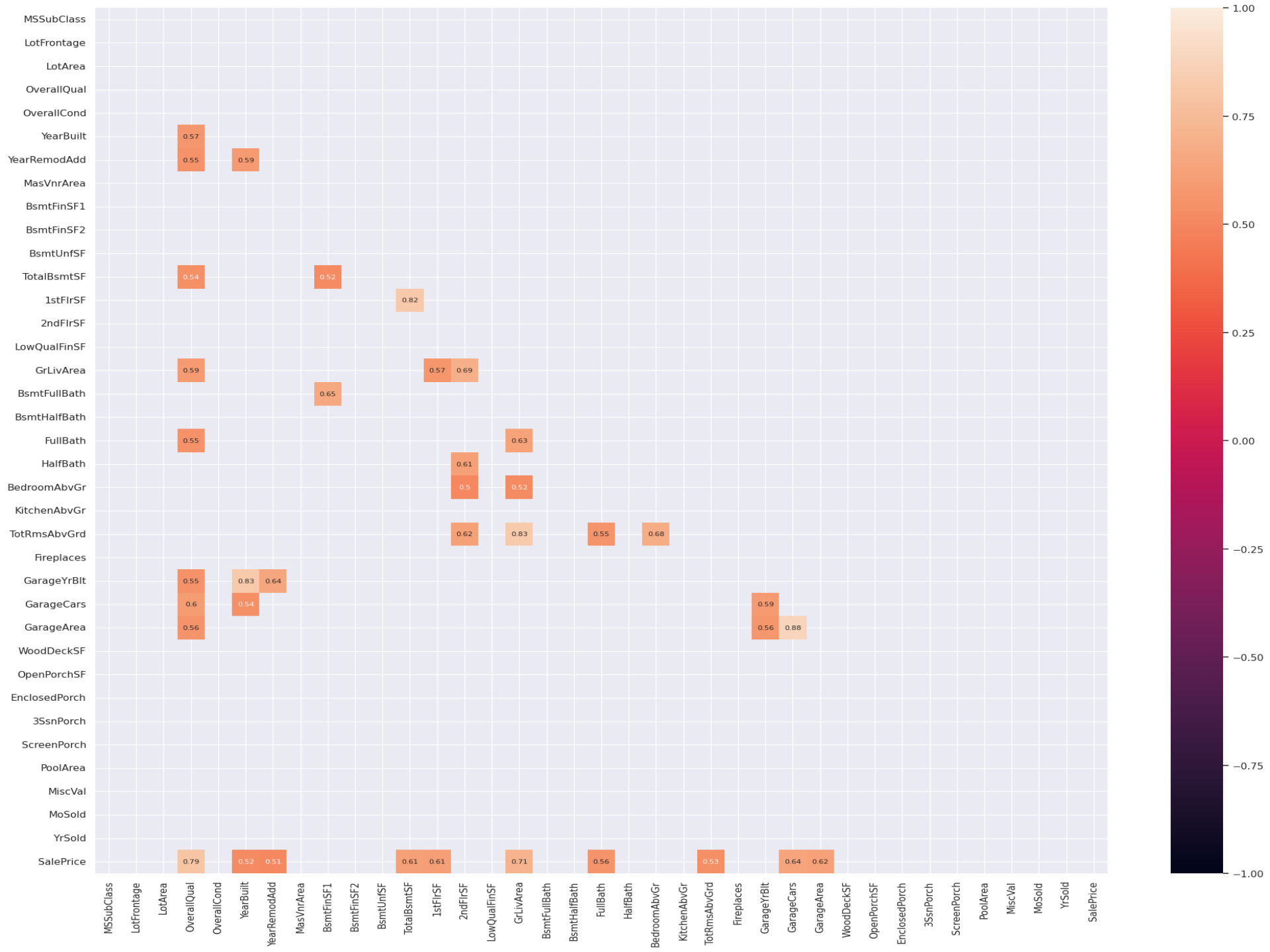
# House Sales Data Exploration

- Even this heatmap is difficult to interpret
- Some optimization will help!!

```
1 corr = df_num.corr()
2 mask = np.triu(np.ones_like(corr, dtype=bool))
3 plt.figure(figsize=(24, 20))
4 sns.heatmap(corr[(corr>=0.5)|(corr<=-0.5)],
5             mask=mask,
6             annot=True,
7             annot_kws={'size':8},
8             vmin=-1,
9             vmax=1)
```

- See the heatmap and state your observations?
  - There is a **strong negative** correlation between BsmtUnfSF and BsmtFinSF2
  - **Others?**





# House Sales Data Exploration



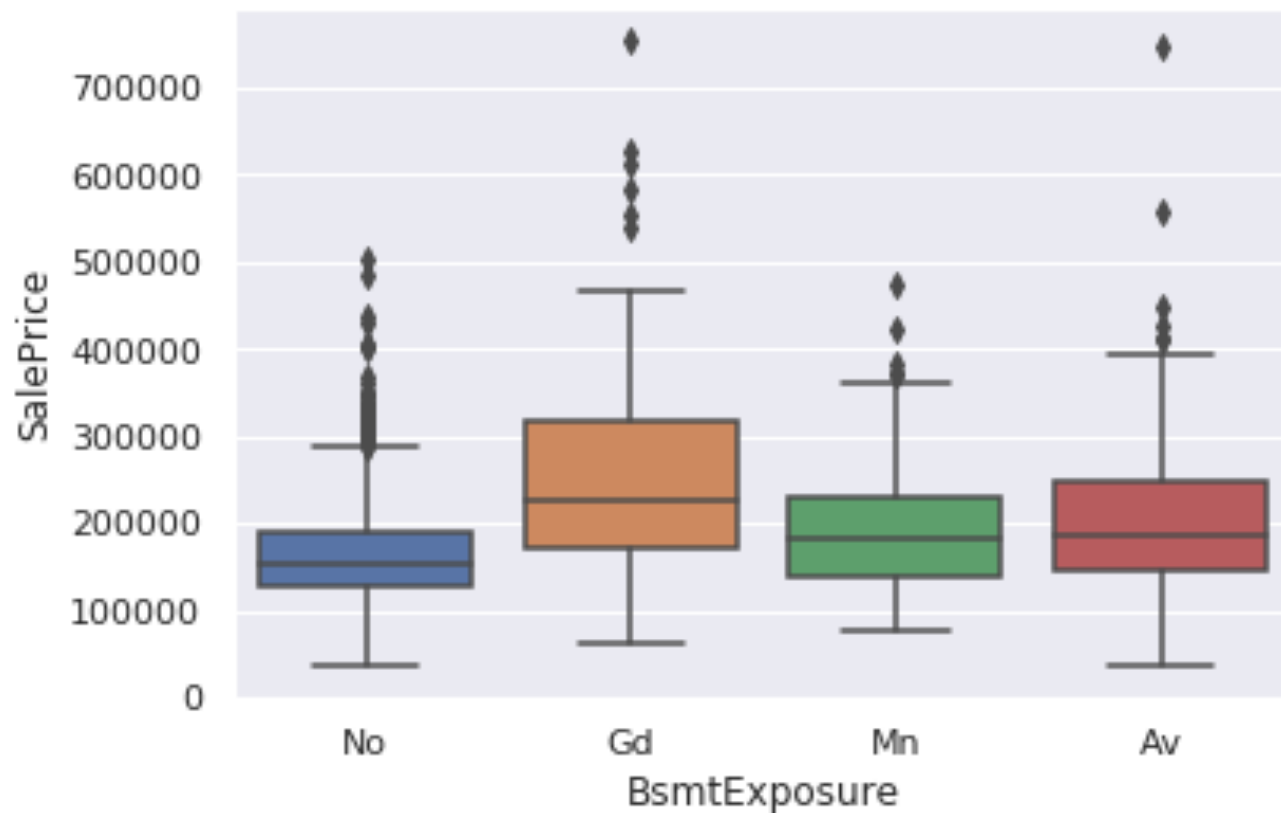
- Can we decrease the number of features based on our previous observations?

# House Sales Data Exploration

- Let us look at some **categorical** features of our dataset and see if we can find some insight in them.
  - Consider (BsmtExposure) feature
- Use boxplot

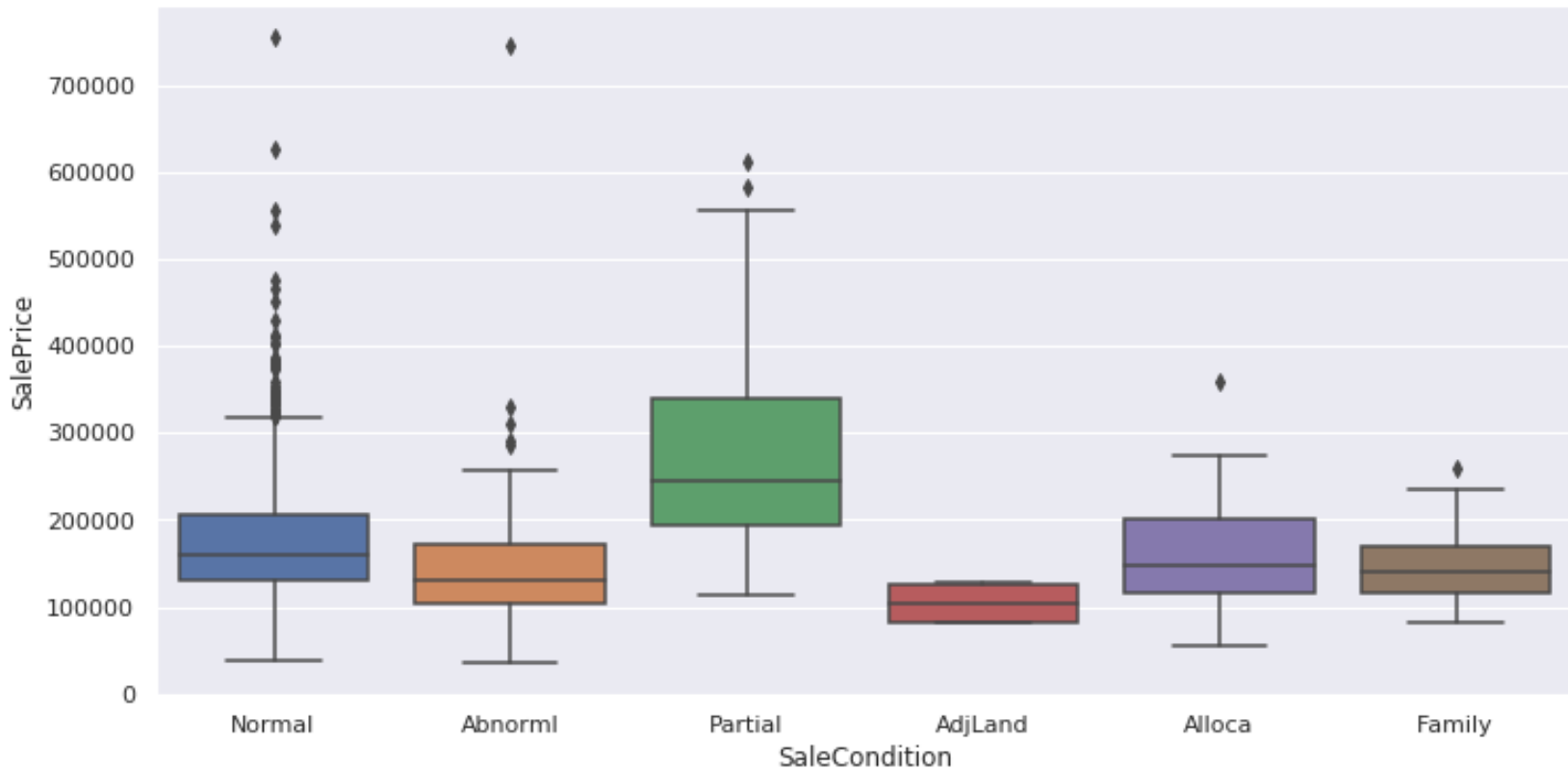
```
1 sns.boxplot(x='BsmtExposure', y='SalePrice', data=df)
```

# House Sales Data Exploration



# House Sales Data Exploration

- Check Sales Condition feature?



# Exercise

# Iris Dataset

# Iris dataset

- Introduced by the British statistician / biologist Ronald Fisher in his 1936
- It is a multi-class classification problem.

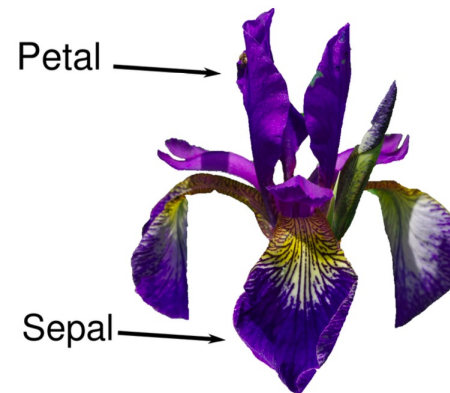
Number of Instances:	150	Area:	Life
Number of Attributes:	4	Date Donated	1988-07-01
Missing Values?	No	Number of Web Hits:	4457610

<https://archive.ics.uci.edu/ml/datasets/Iris>



# Iris dataset (cont'd)

- Balanced dataset
- 150 observations
- 4 numeric variables
  - Sepal length in cm
  - Sepal width in cm
  - Petal length in cm
  - Petal width in cm
- 1 target variable
  - **Class** (Iris Setosa, Iris Versicolour, Iris Virginica)



# Iris dataset (cont'd)

- No need to download it.



[https://scikit-learn.org/stable/datasets/toy\\_dataset.html](https://scikit-learn.org/stable/datasets/toy_dataset.html)

<code>load_boston(*[, return_X_y])</code>	DEPRECATED: <code>load_boston</code> is deprecated in 1.0 and will be removed in 1.2.
<code>load_iris(*[, return_X_y, as_frame])</code>	Load and return the iris dataset (classification).
<code>load_diabetes(*[, return_X_y, as_frame])</code>	Load and return the diabetes dataset (regression).
<code>load_digits(*[, n_class, return_X_y, as_frame])</code>	Load and return the digits dataset (classification).
<code>load_linnerud(*[, return_X_y, as_frame])</code>	Load and return the physical exercise Linnerud dataset.
<code>load_wine(*[, return_X_y, as_frame])</code>	Load and return the wine dataset (classification).
<code>load_breast_cancer(*[, return_X_y, as_frame])</code>	Load and return the breast cancer wisconsin dataset (classification).

# Iris dataset (cont'd)

```
from sklearn.datasets import load_iris
iris_dataset = load_iris()
```

return a **Dictionary** like object

## Returns

data : `~sklearn.utils.Bunch`  
Dictionary-like object, with the following attributes.

```
import seaborn as sns
iris = sns.load_dataset('iris')
```

return a **DataFrame**

## Returns

df : `pandas.DataFrame`  
Tabular data, possibly with some preprocessing applied.

```
sns.load_dataset("iris")
```

# Iris data Vs. Real data



# Exercise



Perform EDA on Iris dataset by doing:

- Descriptive statistics
- Removing duplicate data entries
- Compare between various species based on petal length and width.
  - Write your data insights in a notebook text
- Boxplot petal width distribution over species
  - Write your data insights in a notebook text