housing_data_preprocessing

April 28, 2021

1 Housing Price Model - Data Preprocessing

Author: **Paul A. Beata**
GitHub: pbeata

The original data set comes from the Ames, Iowa housing data on Kaggle.

Description of the columns (features): Each column represents a different feature of the home. Some of the features are numerical (such as "lot area" and "year built) and others are categorical (e.g., "neighboorhood" and "Building Type"). There are some features with absolute measurements, such as areas or distances, but there are also subjective ones like overall quality at the time of sale. A description each feature is provided in the "Ames_Housing_Data_Feature_Description.txt" file included with this project and referenced in this notebook.

Description of the targets: The target is the sale price of the house. Our ultimate goal is to build regression models to predict the sale price of a home based on its features. In this notebook, we will only focus on the data preprocessing.

Description of the rows: Each row of the data set is a single observation (i.e., a single house). Each house is identified in the "PID" column which is a unique identifier for each property.

Summary of data preprocessing performed in this notebook: Since this data set has missing values and outliers, we will first perform the following data preprocessing steps before building the regression models in a separate notebook.

1. Identify the potential outliers in the data set
2. Handle missing data using row-wise and/or column-wise solutions as appropriate
3. Create data checkpoints throughout the preprocessing
4. Handle the categorical data by making dummy variables
5. Save the final (cleaned) data set at the end of the notebook

1.1 Part 0: Load the Original Data

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

```
%config Completer.use_jedi = False
```

Here we can see a detailed description of the features (columns) in the data set before we begin preprocessing:

[2]:
```
with open('./data_original/Ames_Housing_Data_Feature_Description.txt', 'r') as␣
↪f:
    print(f.read())
```

```
MSSubClass: Identifies the type of dwelling involved in the sale.

        20      1-STORY 1946 & NEWER ALL STYLES
        30      1-STORY 1945 & OLDER
        40      1-STORY W/FINISHED ATTIC ALL AGES
        45      1-1/2 STORY - UNFINISHED ALL AGES
        50      1-1/2 STORY FINISHED ALL AGES
        60      2-STORY 1946 & NEWER
        70      2-STORY 1945 & OLDER
        75      2-1/2 STORY ALL AGES
        80      SPLIT OR MULTI-LEVEL
        85      SPLIT FOYER
        90      DUPLEX - ALL STYLES AND AGES
       120      1-STORY PUD (Planned Unit Development) - 1946 & NEWER
       150      1-1/2 STORY PUD - ALL AGES
       160      2-STORY PUD - 1946 & NEWER
       180      PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
       190      2 FAMILY CONVERSION - ALL STYLES AND AGES

MSZoning: Identifies the general zoning classification of the sale.

       A        Agriculture
       C        Commercial
       FV       Floating Village Residential
       I        Industrial
       RH       Residential High Density
       RL       Residential Low Density
       RP       Residential Low Density Park
       RM       Residential Medium Density

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

       Grvl     Gravel
       Pave     Paved
```

Alley: Type of alley access to property

Grvl	Gravel
Pave	Paved
NA	No alley access

LotShape: General shape of property

Reg	Regular
IR1	Slightly irregular
IR2	Moderately Irregular
IR3	Irregular

LandContour: Flatness of the property

Lvl	Near Flat/Level
Bnk	Banked - Quick and significant rise from street grade to building
HLS	Hillside - Significant slope from side to side
Low	Depression

Utilities: Type of utilities available

AllPub	All public Utilities (E,G,W,& S)
NoSewr	Electricity, Gas, and Water (Septic Tank)
NoSeWa	Electricity and Gas Only
ELO	Electricity only

LotConfig: Lot configuration

Inside	Inside lot
Corner	Corner lot
CulDSac	Cul-de-sac
FR2	Frontage on 2 sides of property
FR3	Frontage on 3 sides of property

LandSlope: Slope of property

Gtl	Gentle slope
Mod	Moderate Slope
Sev	Severe Slope

Neighborhood: Physical locations within Ames city limits

Blmngtn	Bloomington Heights
Blueste	Bluestem
BrDale	Briardale
BrkSide	Brookside

```
       ClearCr  Clear Creek
       CollgCr  College Creek
       Crawfor  Crawford
       Edwards  Edwards
       Gilbert  Gilbert
       IDOTRR   Iowa DOT and Rail Road
       MeadowV  Meadow Village
       Mitchel  Mitchell
       Names    North Ames
       NoRidge  Northridge
       NPkVill  Northpark Villa
       NridgHt  Northridge Heights
       NWAmes   Northwest Ames
       OldTown  Old Town
       SWISU    South & West of Iowa State University
       Sawyer   Sawyer
       SawyerW  Sawyer West
       Somerst  Somerset
       StoneBr  Stone Brook
       Timber   Timberland
       Veenker  Veenker

Condition1: Proximity to various conditions

       Artery   Adjacent to arterial street
       Feedr    Adjacent to feeder street
       Norm     Normal
       RRNn     Within 200' of North-South Railroad
       RRAn     Adjacent to North-South Railroad
       PosN     Near positive off-site feature--park, greenbelt, etc.
       PosA     Adjacent to postive off-site feature
       RRNe     Within 200' of East-West Railroad
       RRAe     Adjacent to East-West Railroad

Condition2: Proximity to various conditions (if more than one is present)

       Artery   Adjacent to arterial street
       Feedr    Adjacent to feeder street
       Norm     Normal
       RRNn     Within 200' of North-South Railroad
       RRAn     Adjacent to North-South Railroad
       PosN     Near positive off-site feature--park, greenbelt, etc.
       PosA     Adjacent to postive off-site feature
       RRNe     Within 200' of East-West Railroad
       RRAe     Adjacent to East-West Railroad

BldgType: Type of dwelling
```

```
       1Fam     Single-family Detached
       2FmCon   Two-family Conversion; originally built as one-family dwelling
       Duplx    Duplex
       TwnhsE   Townhouse End Unit
       TwnhsI   Townhouse Inside Unit

HouseStyle: Style of dwelling

       1Story   One story
       1.5Fin   One and one-half story: 2nd level finished
       1.5Unf   One and one-half story: 2nd level unfinished
       2Story   Two story
       2.5Fin   Two and one-half story: 2nd level finished
       2.5Unf   Two and one-half story: 2nd level unfinished
       SFoyer   Split Foyer
       SLvl     Split Level

OverallQual: Rates the overall material and finish of the house

       10       Very Excellent
       9        Excellent
       8        Very Good
       7        Good
       6        Above Average
       5        Average
       4        Below Average
       3        Fair
       2        Poor
       1        Very Poor

OverallCond: Rates the overall condition of the house

       10       Very Excellent
       9        Excellent
       8        Very Good
       7        Good
       6        Above Average
       5        Average
       4        Below Average
       3        Fair
       2        Poor
       1        Very Poor

YearBuilt: Original construction date

YearRemodAdd: Remodel date (same as construction date if no remodeling or
additions)
```

```
RoofStyle: Type of roof

       Flat     Flat
       Gable    Gable
       Gambrel  Gabrel (Barn)
       Hip      Hip
       Mansard  Mansard
       Shed     Shed

RoofMatl: Roof material

       ClyTile  Clay or Tile
       CompShg  Standard (Composite) Shingle
       Membran  Membrane
       Metal    Metal
       Roll     Roll
       Tar&Grv  Gravel & Tar
       WdShake  Wood Shakes
       WdShngl  Wood Shingles

Exterior1st: Exterior covering on house

       AsbShng  Asbestos Shingles
       AsphShn  Asphalt Shingles
       BrkComm  Brick Common
       BrkFace  Brick Face
       CBlock   Cinder Block
       CemntBd  Cement Board
       HdBoard  Hard Board
       ImStucc  Imitation Stucco
       MetalSd  Metal Siding
       Other    Other
       Plywood  Plywood
       PreCast  PreCast
       Stone    Stone
       Stucco   Stucco
       VinylSd  Vinyl Siding
       Wd Sdng  Wood Siding
       WdShing  Wood Shingles

Exterior2nd: Exterior covering on house (if more than one material)

       AsbShng  Asbestos Shingles
       AsphShn  Asphalt Shingles
       BrkComm  Brick Common
       BrkFace  Brick Face
       CBlock   Cinder Block
       CemntBd  Cement Board
```

```
        HdBoard  Hard Board
        ImStucc  Imitation Stucco
        MetalSd  Metal Siding
        Other    Other
        Plywood  Plywood
        PreCast  PreCast
        Stone    Stone
        Stucco   Stucco
        VinylSd  Vinyl Siding
        Wd Sdng  Wood Siding
        WdShing  Wood Shingles

MasVnrType: Masonry veneer type

        BrkCmn   Brick Common
        BrkFace  Brick Face
        CBlock   Cinder Block
        None     None
        Stone    Stone

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

        Ex       Excellent
        Gd       Good
        TA       Average/Typical
        Fa       Fair
        Po       Poor

ExterCond: Evaluates the present condition of the material on the exterior

        Ex       Excellent
        Gd       Good
        TA       Average/Typical
        Fa       Fair
        Po       Poor

Foundation: Type of foundation

        BrkTil   Brick & Tile
        CBlock   Cinder Block
        PConc    Poured Contrete
        Slab     Slab
        Stone    Stone
        Wood     Wood

BsmtQual: Evaluates the height of the basement
```

Ex	Excellent (100+ inches)
Gd	Good (90-99 inches)
TA	Typical (80-89 inches)
Fa	Fair (70-79 inches)
Po	Poor (<70 inches
NA	No Basement

BsmtCond: Evaluates the general condition of the basement

Ex	Excellent
Gd	Good
TA	Typical - slight dampness allowed
Fa	Fair - dampness or some cracking or settling
Po	Poor - Severe cracking, settling, or wetness
NA	No Basement

BsmtExposure: Refers to walkout or garden level walls

Gd	Good Exposure
Av	Average Exposure (split levels or foyers typically score average or above)
Mn	Mimimum Exposure
No	No Exposure
NA	No Basement

BsmtFinType1: Rating of basement finished area

GLQ	Good Living Quarters
ALQ	Average Living Quarters
BLQ	Below Average Living Quarters
Rec	Average Rec Room
LwQ	Low Quality
Unf	Unfinshed
NA	No Basement

BsmtFinSF1: Type 1 finished square feet

BsmtFinType2: Rating of basement finished area (if multiple types)

GLQ	Good Living Quarters
ALQ	Average Living Quarters
BLQ	Below Average Living Quarters
Rec	Average Rec Room
LwQ	Low Quality
Unf	Unfinshed
NA	No Basement

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

Floor	Floor Furnace
GasA	Gas forced warm air furnace
GasW	Gas hot water or steam heat
Grav	Gravity furnace
OthW	Hot water or steam heat other than gas
Wall	Wall furnace

HeatingQC: Heating quality and condition

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

CentralAir: Central air conditioning

N	No
Y	Yes

Electrical: Electrical system

SBrkr	Standard Circuit Breakers & Romex
FuseA	Fuse Box over 60 AMP and all Romex wiring (Average)
FuseF	60 AMP Fuse Box and mostly Romex wiring (Fair)
FuseP	60 AMP Fuse Box and mostly knob & tube wiring (poor)
Mix	Mixed

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

```
       Ex       Excellent
       Gd       Good
       TA       Typical/Average
       Fa       Fair
       Po       Poor
```

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

Functional: Home functionality (Assume typical unless deductions are warranted)

```
       Typ      Typical Functionality
       Min1     Minor Deductions 1
       Min2     Minor Deductions 2
       Mod      Moderate Deductions
       Maj1     Major Deductions 1
       Maj2     Major Deductions 2
       Sev      Severely Damaged
       Sal      Salvage only
```

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

```
       Ex       Excellent - Exceptional Masonry Fireplace
       Gd       Good - Masonry Fireplace in main level
       TA       Average - Prefabricated Fireplace in main living area or Masonry
Fireplace in basement
       Fa       Fair - Prefabricated Fireplace in basement
       Po       Poor - Ben Franklin Stove
       NA       No Fireplace
```

GarageType: Garage location

```
       2Types   More than one type of garage
       Attchd   Attached to home
       Basment  Basement Garage
       BuiltIn  Built-In (Garage part of house - typically has room above
garage)
```

```
       CarPort  Car Port
       Detchd   Detached from home
       NA       No Garage

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

       Fin      Finished
       RFn      Rough Finished
       Unf      Unfinished
       NA       No Garage

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

       Ex       Excellent
       Gd       Good
       TA       Typical/Average
       Fa       Fair
       Po       Poor
       NA       No Garage

GarageCond: Garage condition

       Ex       Excellent
       Gd       Good
       TA       Typical/Average
       Fa       Fair
       Po       Poor
       NA       No Garage

PavedDrive: Paved driveway

       Y        Paved
       P        Partial Pavement
       N        Dirt/Gravel

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet
```

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
NA	No Pool

Fence: Fence quality

GdPrv	Good Privacy
MnPrv	Minimum Privacy
GdWo	Good Wood
MnWw	Minimum Wood/Wire
NA	No Fence

MiscFeature: Miscellaneous feature not covered in other categories

Elev	Elevator
Gar2	2nd Garage (if not described in garage section)
Othr	Other
Shed	Shed (over 100 SF)
TenC	Tennis Court
NA	None

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

SaleType: Type of sale

WD	Warranty Deed - Conventional
CWD	Warranty Deed - Cash
VWD	Warranty Deed - VA Loan
New	Home just constructed and sold
COD	Court Officer Deed/Estate
Con	Contract 15% Down payment regular terms
ConLw	Contract Low Down payment and low interest
ConLI	Contract Low Interest
ConLD	Contract Low Down
Oth	Other

```
SaleCondition: Condition of sale

       Normal   Normal Sale
       Abnorml  Abnormal Sale -  trade, foreclosure, short sale
       AdjLand  Adjoining Land Purchase
       Alloca   Allocation - two linked properties with separate deeds,
typically condo with a garage unit
       Family   Sale between family members
       Partial  Home was not completed when last assessed (associated with New
Homes)
```

[3]:
```python
# load the original raw data set
df = pd.read_csv('./data_original/Ames_Housing_Data.csv')
df
```

[3]:
```
            PID  MS SubClass MS Zoning  Lot Frontage  Lot Area Street Alley  \
0     526301100           20        RL         141.0     31770   Pave   NaN
1     526350040           20        RH          80.0     11622   Pave   NaN
2     526351010           20        RL          81.0     14267   Pave   NaN
3     526353030           20        RL          93.0     11160   Pave   NaN
4     527105010           60        RL          74.0     13830   Pave   NaN
…           …          …       …            …        …    …    …
2925  923275080           80        RL          37.0      7937   Pave   NaN
2926  923276100           20        RL           NaN      8885   Pave   NaN
2927  923400125           85        RL          62.0     10441   Pave   NaN
2928  924100070           20        RL          77.0     10010   Pave   NaN
2929  924151050           60        RL          74.0      9627   Pave   NaN

     Lot Shape Land Contour Utilities  … Pool Area Pool QC  Fence  \
0          IR1          Lvl    AllPub  …         0     NaN    NaN
1          Reg          Lvl    AllPub  …         0     NaN  MnPrv
2          IR1          Lvl    AllPub  …         0     NaN    NaN
3          Reg          Lvl    AllPub  …         0     NaN    NaN
4          IR1          Lvl    AllPub  …         0     NaN  MnPrv
…        …            …      …  …   …     …    …
2925       IR1          Lvl    AllPub  …         0     NaN  GdPrv
2926       IR1          Low    AllPub  …         0     NaN  MnPrv
2927       Reg          Lvl    AllPub  …         0     NaN  MnPrv
2928       Reg          Lvl    AllPub  …         0     NaN    NaN
2929       Reg          Lvl    AllPub  …         0     NaN    NaN

     Misc Feature Misc Val Mo Sold Yr Sold  Sale Type  Sale Condition  \
0             NaN        0       5    2010        WD           Normal
1             NaN        0       6    2010        WD           Normal
2            Gar2    12500       6    2010        WD           Normal
```

```
3            NaN      0       4    2010        WD          Normal
4            NaN      0       3    2010        WD          Normal
…              …      …       …       …     …          …
2925         NaN      0       3    2006        WD          Normal
2926         NaN      0       6    2006        WD          Normal
2927        Shed    700       7    2006        WD          Normal
2928         NaN      0       4    2006        WD          Normal
2929         NaN      0      11    2006        WD          Normal

      SalePrice
0        215000
1        105000
2        172000
3        244000
4        189900
…           …
2925     142500
2926     131000
2927     132000
2928     170000
2929     188000

[2930 rows x 81 columns]
```

[4]:
```
df.describe()
```

[4]:
```
                PID  MS SubClass  Lot Frontage       Lot Area  Overall Qual  \
count  2.930000e+03  2930.000000   2440.000000    2930.000000   2930.000000
mean   7.144645e+08    57.387372     69.224590   10147.921843      6.094881
std    1.887308e+08    42.638025     23.365335    7880.017759      1.411026
min    5.263011e+08    20.000000     21.000000    1300.000000      1.000000
25%    5.284770e+08    20.000000     58.000000    7440.250000      5.000000
50%    5.354536e+08    50.000000     68.000000    9436.500000      6.000000
75%    9.071811e+08    70.000000     80.000000   11555.250000      7.000000
max    1.007100e+09   190.000000    313.000000  215245.000000     10.000000

       Overall Cond   Year Built  Year Remod/Add  Mas Vnr Area  BsmtFin SF 1  \
count   2930.000000  2930.000000     2930.000000   2907.000000   2929.000000
mean       5.563140  1971.356314     1984.266553    101.896801    442.629566
std        1.111537    30.245361       20.860286    179.112611    455.590839
min        1.000000  1872.000000     1950.000000      0.000000      0.000000
25%        5.000000  1954.000000     1965.000000      0.000000      0.000000
50%        5.000000  1973.000000     1993.000000      0.000000    370.000000
75%        6.000000  2001.000000     2004.000000    164.000000    734.000000
max        9.000000  2010.000000     2010.000000   1600.000000   5644.000000

       …  Wood Deck SF  Open Porch SF  Enclosed Porch   3Ssn Porch  \
```

```
count  …   2930.000000   2930.000000    2930.000000  2930.000000
mean   …     93.751877     47.533447      23.011604     2.592491
std    …    126.361562     67.483400      64.139059    25.141331
min    …      0.000000      0.000000       0.000000     0.000000
25%    …      0.000000      0.000000       0.000000     0.000000
50%    …      0.000000     27.000000       0.000000     0.000000
75%    …    168.000000     70.000000       0.000000     0.000000
max    …   1424.000000    742.000000    1012.000000   508.000000

       Screen Porch    Pool Area      Misc Val      Mo Sold      Yr Sold  \
count   2930.000000  2930.000000   2930.000000  2930.000000  2930.000000
mean      16.002048     2.243345     50.635154     6.216041  2007.790444
std       56.087370    35.597181    566.344288     2.714492     1.316613
min        0.000000     0.000000      0.000000     1.000000  2006.000000
25%        0.000000     0.000000      0.000000     4.000000  2007.000000
50%        0.000000     0.000000      0.000000     6.000000  2008.000000
75%        0.000000     0.000000      0.000000     8.000000  2009.000000
max      576.000000   800.000000  17000.000000    12.000000  2010.000000

           SalePrice
count    2930.000000
mean   180796.060068
std     79886.692357
min     12789.000000
25%    129500.000000
50%    160000.000000
75%    213500.000000
max    755000.000000

[8 rows x 38 columns]
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 81 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   PID              2930 non-null   int64
 1   MS SubClass      2930 non-null   int64
 2   MS Zoning        2930 non-null   object
 3   Lot Frontage     2440 non-null   float64
 4   Lot Area         2930 non-null   int64
 5   Street           2930 non-null   object
 6   Alley            198 non-null    object
 7   Lot Shape        2930 non-null   object
 8   Land Contour     2930 non-null   object
```

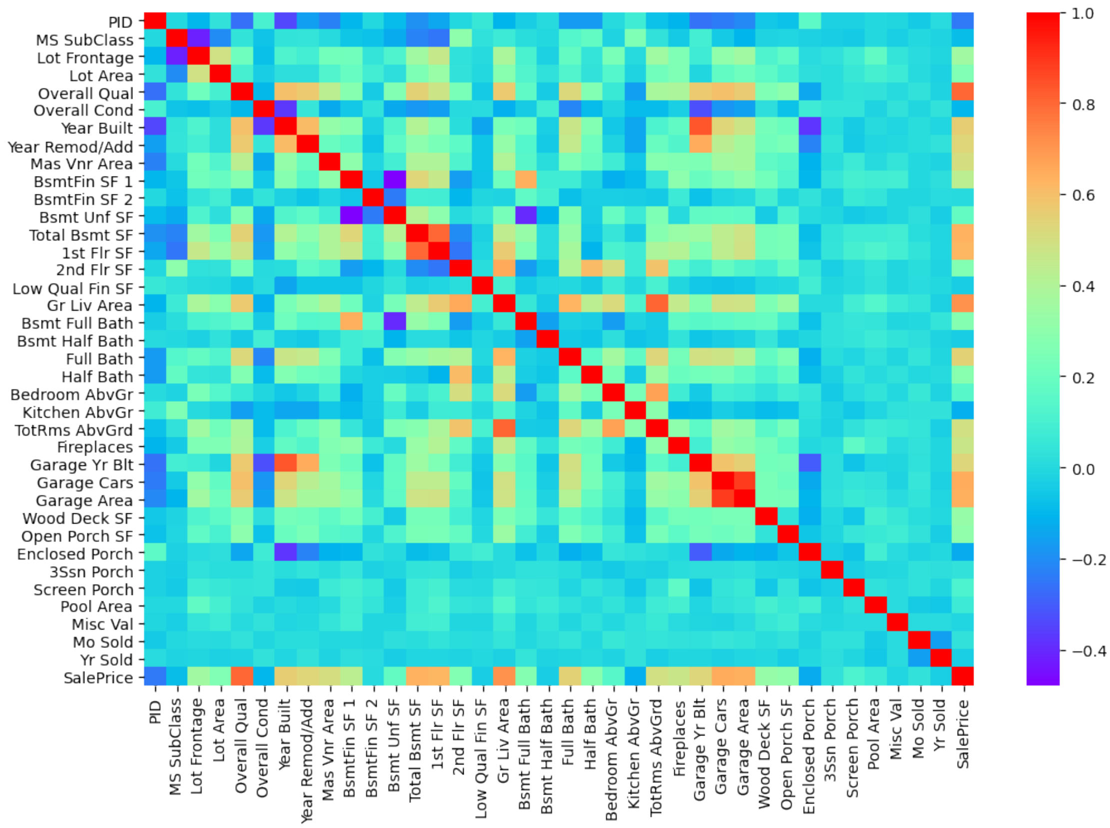
```
9   Utilities        2930 non-null   object
10  Lot Config       2930 non-null   object
11  Land Slope       2930 non-null   object
12  Neighborhood     2930 non-null   object
13  Condition 1      2930 non-null   object
14  Condition 2      2930 non-null   object
15  Bldg Type        2930 non-null   object
16  House Style      2930 non-null   object
17  Overall Qual     2930 non-null   int64
18  Overall Cond     2930 non-null   int64
19  Year Built       2930 non-null   int64
20  Year Remod/Add   2930 non-null   int64
21  Roof Style       2930 non-null   object
22  Roof Matl        2930 non-null   object
23  Exterior 1st     2930 non-null   object
24  Exterior 2nd     2930 non-null   object
25  Mas Vnr Type     2907 non-null   object
26  Mas Vnr Area     2907 non-null   float64
27  Exter Qual       2930 non-null   object
28  Exter Cond       2930 non-null   object
29  Foundation       2930 non-null   object
30  Bsmt Qual        2850 non-null   object
31  Bsmt Cond        2850 non-null   object
32  Bsmt Exposure    2847 non-null   object
33  BsmtFin Type 1   2850 non-null   object
34  BsmtFin SF 1     2929 non-null   float64
35  BsmtFin Type 2   2849 non-null   object
36  BsmtFin SF 2     2929 non-null   float64
37  Bsmt Unf SF      2929 non-null   float64
38  Total Bsmt SF    2929 non-null   float64
39  Heating          2930 non-null   object
40  Heating QC       2930 non-null   object
41  Central Air      2930 non-null   object
42  Electrical       2929 non-null   object
43  1st Flr SF       2930 non-null   int64
44  2nd Flr SF       2930 non-null   int64
45  Low Qual Fin SF  2930 non-null   int64
46  Gr Liv Area      2930 non-null   int64
47  Bsmt Full Bath   2928 non-null   float64
48  Bsmt Half Bath   2928 non-null   float64
49  Full Bath        2930 non-null   int64
50  Half Bath        2930 non-null   int64
51  Bedroom AbvGr    2930 non-null   int64
52  Kitchen AbvGr    2930 non-null   int64
53  Kitchen Qual     2930 non-null   object
54  TotRms AbvGrd    2930 non-null   int64
55  Functional       2930 non-null   object
56  Fireplaces       2930 non-null   int64
```

```
 57  Fireplace Qu     1508 non-null   object
 58  Garage Type      2773 non-null   object
 59  Garage Yr Blt    2771 non-null   float64
 60  Garage Finish    2771 non-null   object
 61  Garage Cars      2929 non-null   float64
 62  Garage Area      2929 non-null   float64
 63  Garage Qual      2771 non-null   object
 64  Garage Cond      2771 non-null   object
 65  Paved Drive      2930 non-null   object
 66  Wood Deck SF     2930 non-null   int64
 67  Open Porch SF    2930 non-null   int64
 68  Enclosed Porch   2930 non-null   int64
 69  3Ssn Porch       2930 non-null   int64
 70  Screen Porch     2930 non-null   int64
 71  Pool Area        2930 non-null   int64
 72  Pool QC          13 non-null     object
 73  Fence            572 non-null    object
 74  Misc Feature     106 non-null    object
 75  Misc Val         2930 non-null   int64
 76  Mo Sold          2930 non-null   int64
 77  Yr Sold          2930 non-null   int64
 78  Sale Type        2930 non-null   object
 79  Sale Condition   2930 non-null   object
 80  SalePrice        2930 non-null   int64
dtypes: float64(11), int64(27), object(43)
memory usage: 1.8+ MB
```

1.1.1 Observe Correlations

Here we look at the correlation of the various features to see which ones have large positive or negative correlations with the overall house price. In this heat map, we only look at the continuous variables for now since we cannot compute the correlation of categorical values.

[6]:
```python
# heatmap showing the correlations among all features
plt.figure(figsize=(12,8), dpi=100)
sns.heatmap(df.corr(), cmap='rainbow');
```



[7]:
```python
# sort the correlation values to see which features are more correlated with␣
↪sale price
df.corr()['SalePrice'].sort_values()
```

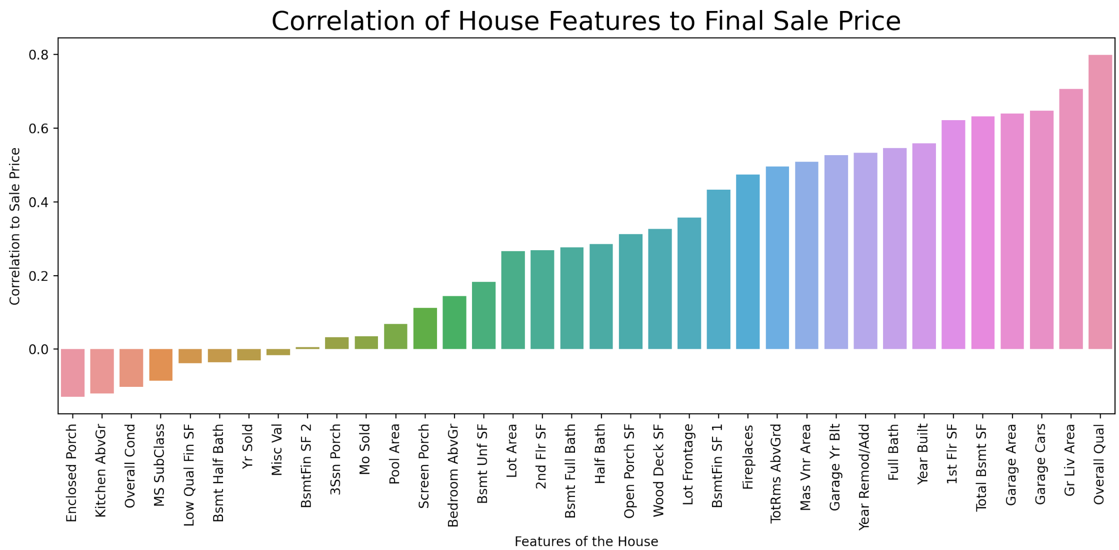
[7]:
```
PID               -0.246521
Enclosed Porch    -0.128787
Kitchen AbvGr     -0.119814
Overall Cond      -0.101697
MS SubClass       -0.085092
Low Qual Fin SF   -0.037660
Bsmt Half Bath    -0.035835
Yr Sold           -0.030569
Misc Val          -0.015691
BsmtFin SF 2       0.005891
3Ssn Porch         0.032225
Mo Sold            0.035259
Pool Area          0.068403
Screen Porch       0.112151
Bedroom AbvGr      0.143913
Bsmt Unf SF        0.182855
Lot Area           0.266549
```

```
2nd Flr SF          0.269373
Bsmt Full Bath      0.276050
Half Bath           0.285056
Open Porch SF       0.312951
Wood Deck SF        0.327143
Lot Frontage        0.357318
BsmtFin SF 1        0.432914
Fireplaces          0.474558
TotRms AbvGrd       0.495474
Mas Vnr Area        0.508285
Garage Yr Blt       0.526965
Year Remod/Add      0.532974
Full Bath           0.545604
Year Built          0.558426
1st Flr SF          0.621676
Total Bsmt SF       0.632280
Garage Area         0.640401
Garage Cars         0.647877
Gr Liv Area         0.706780
Overall Qual        0.799262
SalePrice           1.000000
Name: SalePrice, dtype: float64
```

Same correlation analysis but with a few adjustments: 1. Show the correlation values above in a bar chart 2. Sale price is 100% correlated to itself, so we remove it from the plot 3. PID is simply the property ID number, so we remove it from the plot

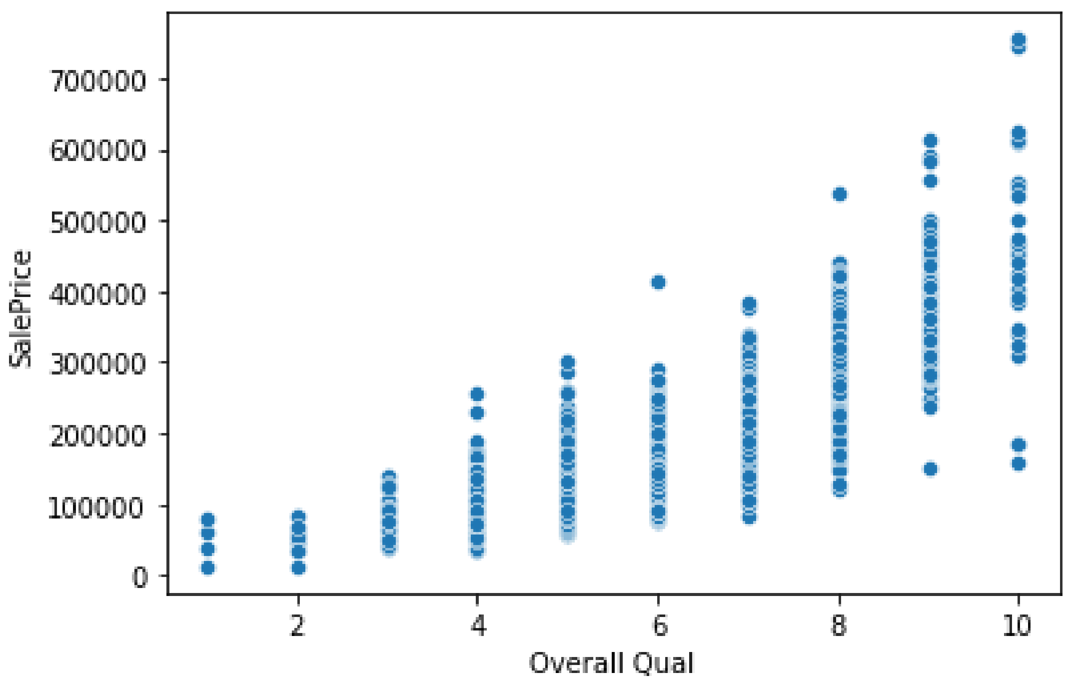
In the correlation plot below, we can see the numerical features with the highest positive (to the right) and negative (to the left) correlation to the final sale price.

[8]:
```python
# bar chart showing feature correlations to sale price
corr_sorted = df.corr()['SalePrice'].sort_values()[1:-1]
plt.figure(figsize=(12,6), dpi=200)
sns.barplot(x=corr_sorted.index, y=corr_sorted)

plt.xticks(rotation=90)
plt.xlabel('Features of the House')
plt.ylabel('Correlation to Sale Price')
plt.title('Correlation of House Features to Final Sale Price', fontsize=20)

plt.tight_layout()
plt.savefig('./house_feature_correlation.svg')
plt.savefig('./house_feature_correlation.png', dpi=300)
plt.show()
```

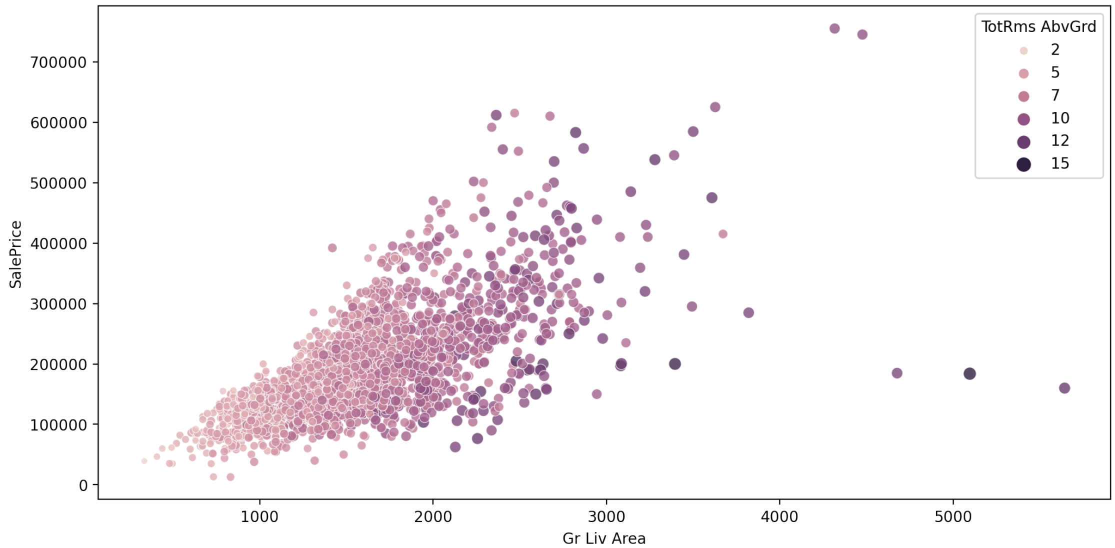


1.2 Part 1: Handling Outliers

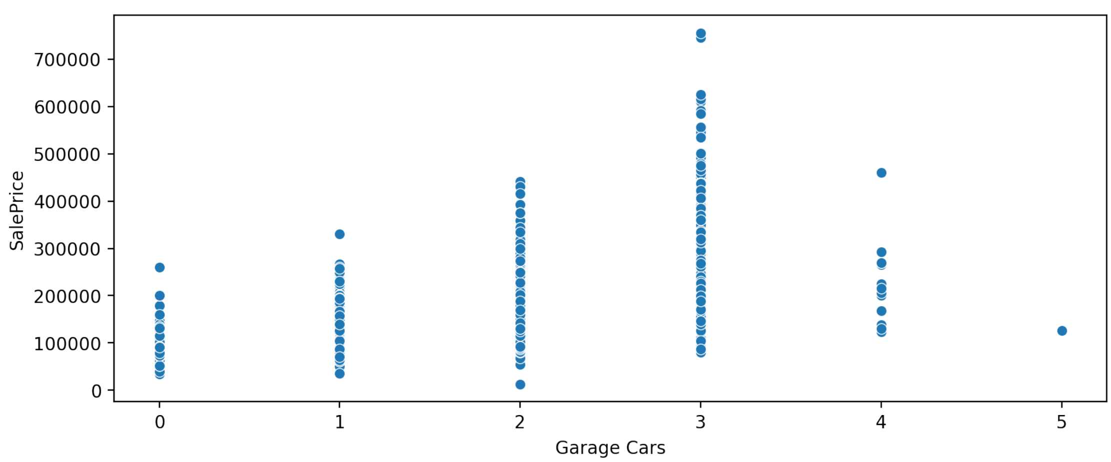
[9]:
```
# overall quality rating of the home versus the sale price
sns.scatterplot(x='Overall Qual', y='SalePrice', data=df);
```



[10]:
```
# overall above-ground living area of the home versus the sale price
plt.figure(figsize=(12,6), dpi=200)
sns.scatterplot(x='Gr Liv Area', y='SalePrice', data=df,
                hue='TotRms AbvGrd',
#                 style='TotRms AbvGrd',
                size='TotRms AbvGrd',
#                 palette='deep',
                alpha=0.8);
```

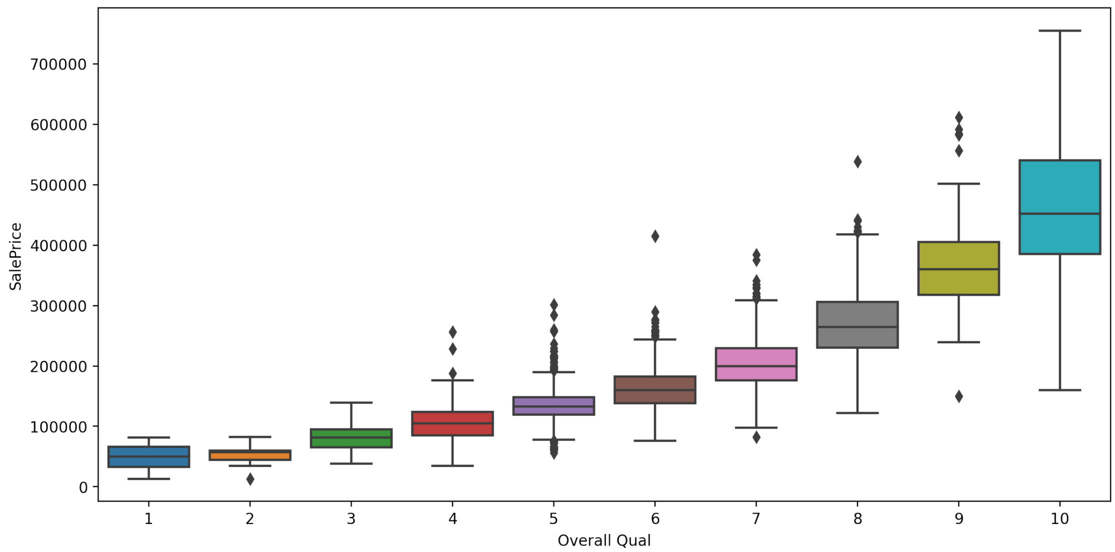


[11]:
```
# size of garage versus the sale price
plt.figure(figsize=(10,4), dpi=200)
sns.scatterplot(x='Garage Cars', y='SalePrice', data=df);
```



From the scatter plots above, we can see that there are potentially a few outliers in the original data set. For example, there are some houses with an overall rating of 9 or 10, but a sale price less than $200,000. Clearly there are other factors affecting the prices of these homes, but these few data points could be considered outliers since the general trend seen above is: the greater the rating, the greater the sale price. We can confirm this here with a box plot (below).

[12]:
```
plt.figure(figsize=(12,6), dpi=200)
sns.boxplot(data=df, x='Overall Qual', y='SalePrice')
plt.show()
```



[13]:
```
df[(df['Overall Qual'] > 8) & (df['SalePrice'] < 200000)]
```

[13]:
```
            PID  MS SubClass MS Zoning  Lot Frontage  Lot Area Street Alley  \
1182  533350090           60        RL           NaN     24572   Pave   NaN
1498  908154235           60        RL         313.0     63887   Pave   NaN
2180  908154195           20        RL         128.0     39290   Pave   NaN
2181  908154205           60        RL         130.0     40094   Pave   NaN

     Lot Shape Land Contour Utilities  … Pool Area Pool QC Fence  \
1182       IR1          Lvl    AllPub  …         0     NaN   NaN
1498       IR3          Bnk    AllPub  …       480      Gd   NaN
2180       IR1          Bnk    AllPub  …         0     NaN   NaN
2181       IR1          Bnk    AllPub  …         0     NaN   NaN

     Misc Feature Misc Val Mo Sold Yr Sold  Sale Type  Sale Condition  \
1182          NaN        0       6    2008        WD           Family
1498          NaN        0       1    2008        New          Partial
2180         Elev    17000      10    2007        New          Partial
2181          NaN        0      10    2007        New          Partial
```

```
      SalePrice
1182     150000
1498     160000
2180     183850
2181     184750

[4 rows x 81 columns]
```

[14]:
```
df[(df['Gr Liv Area'] > 4000) & (df['SalePrice'] < 400000)]
```

[14]:
```
            PID  MS SubClass MS Zoning  Lot Frontage  Lot Area Street Alley  \
1498  908154235           60        RL         313.0     63887   Pave   NaN
2180  908154195           20        RL         128.0     39290   Pave   NaN
2181  908154205           60        RL         130.0     40094   Pave   NaN

     Lot Shape Land Contour Utilities  … Pool Area Pool QC Fence  \
1498       IR3          Bnk    AllPub  …       480      Gd   NaN
2180       IR1          Bnk    AllPub  …         0     NaN   NaN
2181       IR1          Bnk    AllPub  …         0     NaN   NaN

     Misc Feature Misc Val Mo Sold Yr Sold  Sale Type  Sale Condition  \
1498          NaN        0       1    2008        New         Partial
2180         Elev    17000      10    2007        New         Partial
2181          NaN        0      10    2007        New         Partial

      SalePrice
1498     160000
2180     183850
2181     184750

[3 rows x 81 columns]
```

[15]:
```
df[(df['Gr Liv Area'] > 4000) & (df['SalePrice'] < 400000)].index
```

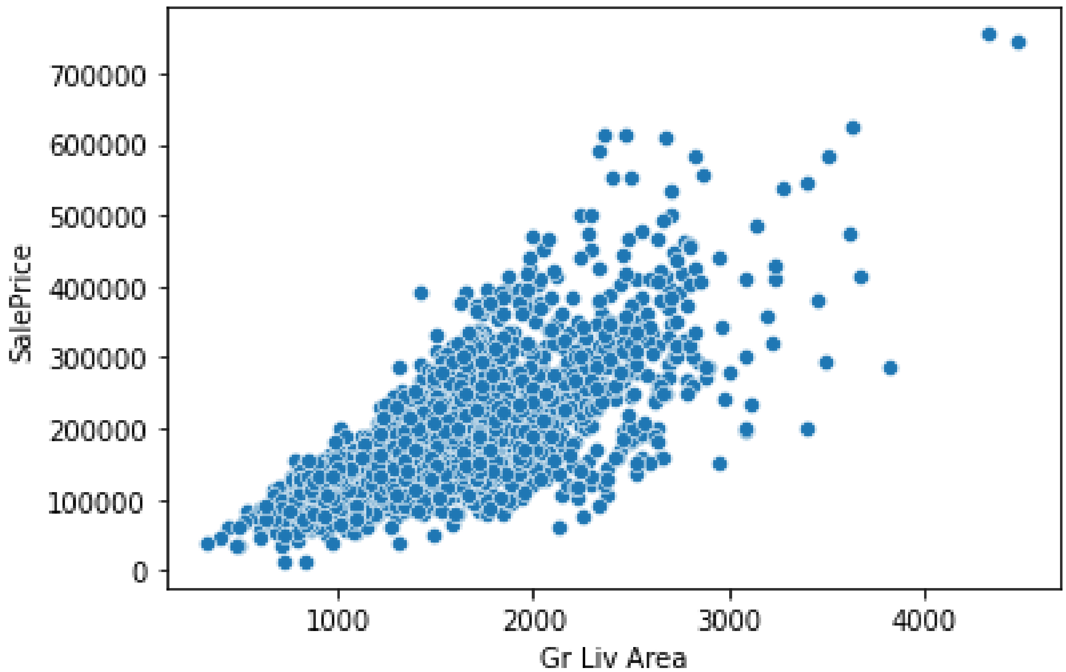
[15]:
```
Int64Index([1498, 2180, 2181], dtype='int64')
```

We choose to drop the observations that have a living area greater than 4000 square feet and a final sale price of less than \$400,000. This corresponds to 3 rows being dropped out of 2930 total rows (0.1% of the data being dropped).

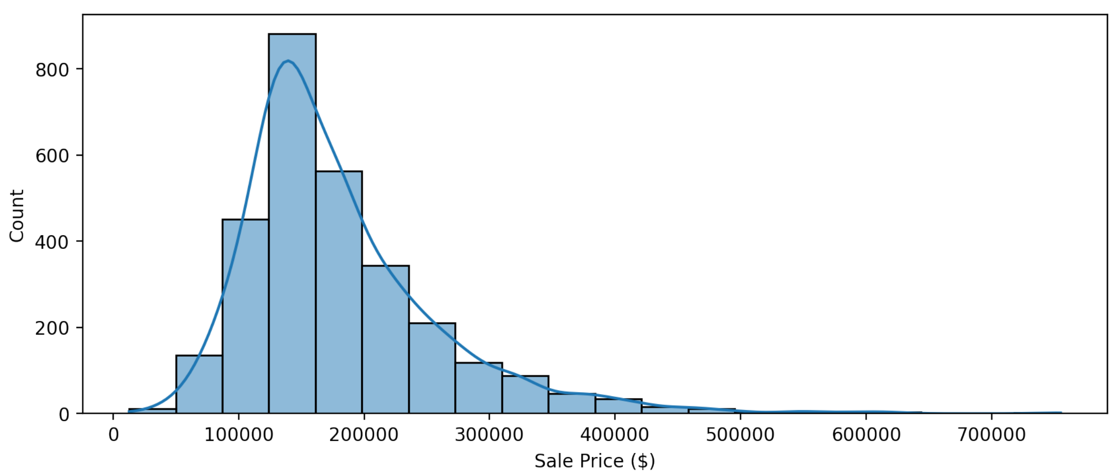
[16]:
```
drop_index = df[(df['Gr Liv Area'] > 4000) & (df['SalePrice'] < 400000)].index
```

[17]:
```
df = df.drop(drop_index, axis=0)
```

[18]:
```
sns.scatterplot(x='Gr Liv Area', y='SalePrice', data=df);
```



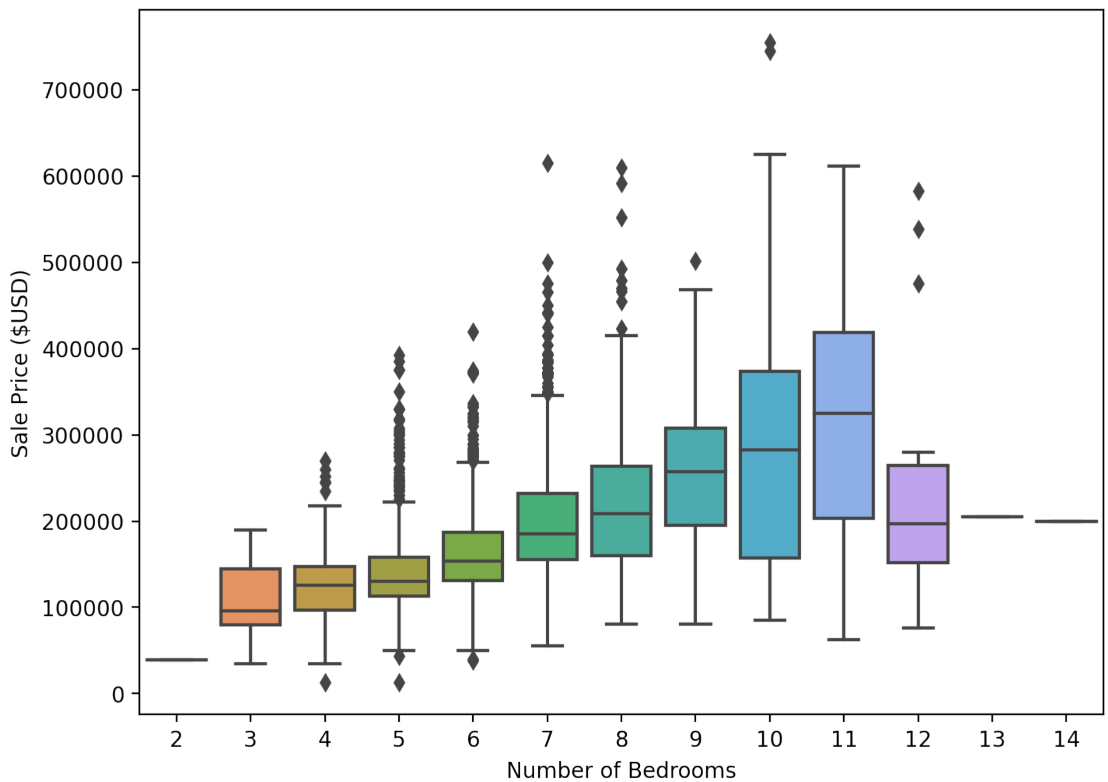
[19]:
```
price = np.array(df['SalePrice'])

plt.figure(figsize=(10,4), dpi=200)
sns.histplot(x=price, bins=20, kde=True)
plt.xlabel('Sale Price ($)')

plt.show()
```



[20]:
```
# as a quick calculation, we can also check the 25th and 75th percentiles for
  ↪the house prices:
q25, q75 = np.percentile(price, [25, 75])
print(q25, q75)
```

```
129500.0 213500.0
```

[21]:
```
plt.figure(figsize=(8,6), dpi=200)
sns.boxplot(y='SalePrice', x='TotRms AbvGrd', data=df)
plt.xlabel('Number of Bedrooms')
plt.ylabel('Sale Price ($USD)')

plt.show()
```



[22]:
```
# CHECKPOINT #1
df_drop_outliers = df.copy()
```

[23]:
```
# df = df_drop_outliers
```

1.3 Part 2: Handling Missing Data

[24]:
```python
# we can drop the PID column because it is simply the property ID value and has␣
↪no predictive power
df = df.drop("PID", axis=1)
```

[25]:
```python
# by checking the descriptive stats, it is clear that we have some missing␣
↪values to handle
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2927 entries, 0 to 2929
Data columns (total 80 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   MS SubClass      2927 non-null   int64
 1   MS Zoning        2927 non-null   object
 2   Lot Frontage     2437 non-null   float64
 3   Lot Area         2927 non-null   int64
 4   Street           2927 non-null   object
 5   Alley            198 non-null    object
 6   Lot Shape        2927 non-null   object
 7   Land Contour     2927 non-null   object
 8   Utilities        2927 non-null   object
 9   Lot Config       2927 non-null   object
 10  Land Slope       2927 non-null   object
 11  Neighborhood     2927 non-null   object
 12  Condition 1      2927 non-null   object
 13  Condition 2      2927 non-null   object
 14  Bldg Type        2927 non-null   object
 15  House Style      2927 non-null   object
 16  Overall Qual     2927 non-null   int64
 17  Overall Cond     2927 non-null   int64
 18  Year Built       2927 non-null   int64
 19  Year Remod/Add   2927 non-null   int64
 20  Roof Style       2927 non-null   object
 21  Roof Matl        2927 non-null   object
 22  Exterior 1st     2927 non-null   object
 23  Exterior 2nd     2927 non-null   object
 24  Mas Vnr Type     2904 non-null   object
 25  Mas Vnr Area     2904 non-null   float64
 26  Exter Qual       2927 non-null   object
 27  Exter Cond       2927 non-null   object
 28  Foundation       2927 non-null   object
 29  Bsmt Qual        2847 non-null   object
 30  Bsmt Cond        2847 non-null   object
 31  Bsmt Exposure    2844 non-null   object
```

32	BsmtFin Type 1	2847 non-null	object
33	BsmtFin SF 1	2926 non-null	float64
34	BsmtFin Type 2	2846 non-null	object
35	BsmtFin SF 2	2926 non-null	float64
36	Bsmt Unf SF	2926 non-null	float64
37	Total Bsmt SF	2926 non-null	float64
38	Heating	2927 non-null	object
39	Heating QC	2927 non-null	object
40	Central Air	2927 non-null	object
41	Electrical	2926 non-null	object
42	1st Flr SF	2927 non-null	int64
43	2nd Flr SF	2927 non-null	int64
44	Low Qual Fin SF	2927 non-null	int64
45	Gr Liv Area	2927 non-null	int64
46	Bsmt Full Bath	2925 non-null	float64
47	Bsmt Half Bath	2925 non-null	float64
48	Full Bath	2927 non-null	int64
49	Half Bath	2927 non-null	int64
50	Bedroom AbvGr	2927 non-null	int64
51	Kitchen AbvGr	2927 non-null	int64
52	Kitchen Qual	2927 non-null	object
53	TotRms AbvGrd	2927 non-null	int64
54	Functional	2927 non-null	object
55	Fireplaces	2927 non-null	int64
56	Fireplace Qu	1505 non-null	object
57	Garage Type	2770 non-null	object
58	Garage Yr Blt	2768 non-null	float64
59	Garage Finish	2768 non-null	object
60	Garage Cars	2926 non-null	float64
61	Garage Area	2926 non-null	float64
62	Garage Qual	2768 non-null	object
63	Garage Cond	2768 non-null	object
64	Paved Drive	2927 non-null	object
65	Wood Deck SF	2927 non-null	int64
66	Open Porch SF	2927 non-null	int64
67	Enclosed Porch	2927 non-null	int64
68	3Ssn Porch	2927 non-null	int64
69	Screen Porch	2927 non-null	int64
70	Pool Area	2927 non-null	int64
71	Pool QC	12 non-null	object
72	Fence	572 non-null	object
73	Misc Feature	105 non-null	object
74	Misc Val	2927 non-null	int64
75	Mo Sold	2927 non-null	int64
76	Yr Sold	2927 non-null	int64
77	Sale Type	2927 non-null	object
78	Sale Condition	2927 non-null	object
79	SalePrice	2927 non-null	int64

```
dtypes: float64(11), int64(26), object(43)
memory usage: 1.9+ MB
```

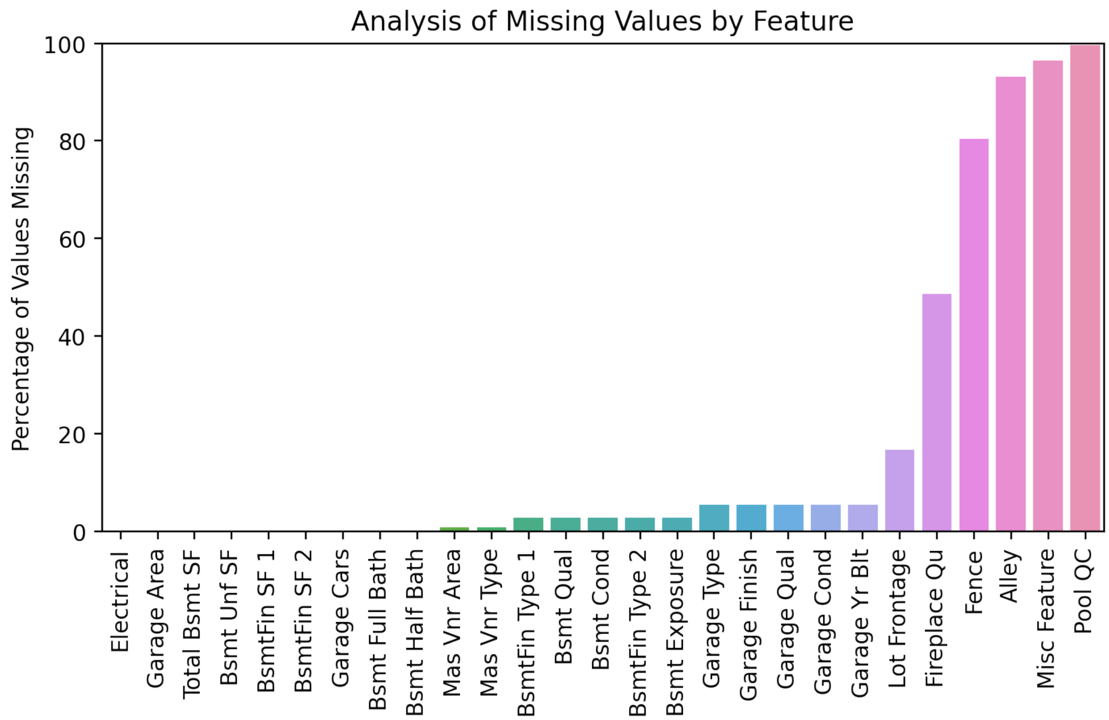
In order to get a better idea of the missing values that we have in the data set, we can use a function that will compute the percentage of missing values for each feature. The percentages computed by the function are sorted and filtered such that we only focus on features (columns) where there are at least 1 missing value.

[26]:
```
# function to compute the percentage of missing values in a DataFrame object
def percent_missing (df):
    percent_nan = 100 * df.isnull().sum() / len(df)
    percent_nan = percent_nan[percent_nan > 0].sort_values()
    return percent_nan
```

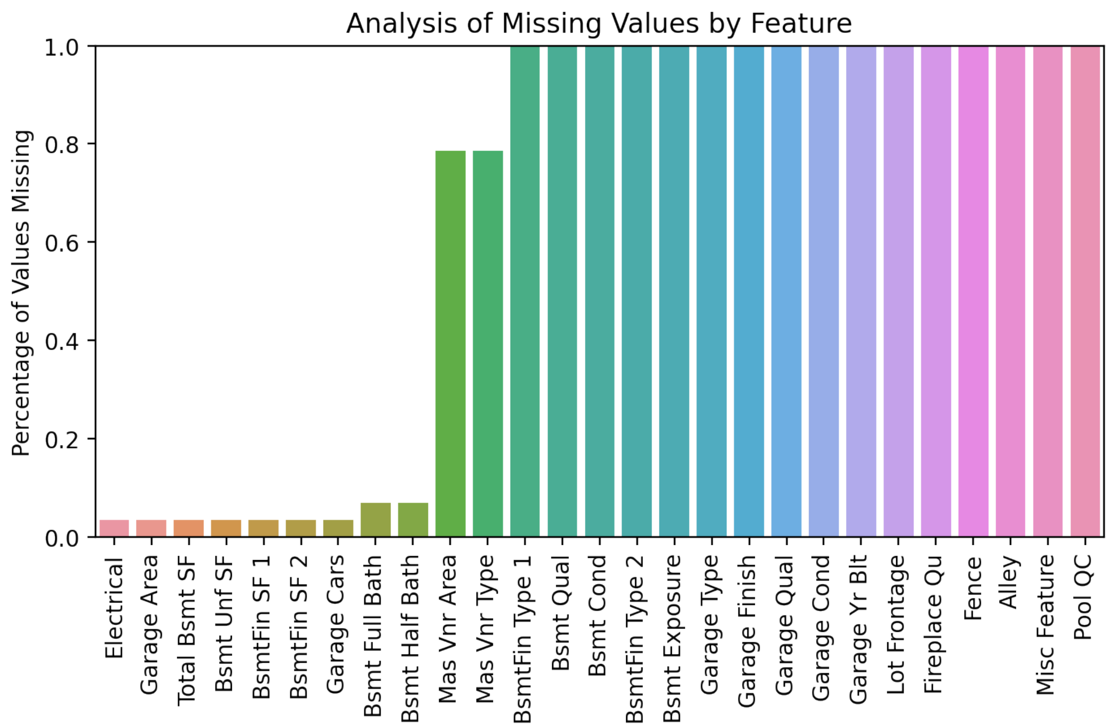
[27]:
```
# compute the percent of missing values in the housing data set (range: 0% to␣
 ↪100%)
percent_nan = percent_missing(df)
percent_nan
```

[27]:
```
Electrical         0.034165
Garage Area        0.034165
Total Bsmt SF      0.034165
Bsmt Unf SF        0.034165
BsmtFin SF 1       0.034165
BsmtFin SF 2       0.034165
Garage Cars        0.034165
Bsmt Full Bath     0.068329
Bsmt Half Bath     0.068329
Mas Vnr Area       0.785787
Mas Vnr Type       0.785787
BsmtFin Type 1     2.733174
Bsmt Qual          2.733174
Bsmt Cond          2.733174
BsmtFin Type 2     2.767339
Bsmt Exposure      2.835668
Garage Type        5.363854
Garage Finish      5.432183
Garage Qual        5.432183
Garage Cond        5.432183
Garage Yr Blt      5.432183
Lot Frontage      16.740690
Fireplace Qu      48.582166
Fence             80.457807
Alley             93.235395
Misc Feature      96.412709
Pool QC           99.590024
dtype: float64
```

[28]:
```
# bar plot of the percentages of missing values
plt.figure(figsize=(8,4), dpi=200)
sns.barplot(x=percent_nan.index, y=percent_nan)
plt.xticks(rotation=90)
plt.ylim(0,100)
plt.title('Analysis of Missing Values by Feature')
plt.ylabel('Percentage of Values Missing')

plt.show()
```



[29]:
```
# bar plot of the percentages of missing values: ZOOMED IN
plt.figure(figsize=(8,4), dpi=200)
sns.barplot(x=percent_nan.index, y=percent_nan)
plt.xticks(rotation=90)
plt.ylim(0, 1)
plt.title('Analysis of Missing Values by Feature')
plt.ylabel('Percentage of Values Missing')

plt.show()
```



[30]:
```
# features that are only missing less than 1% of the data
percent_nan[percent_nan < 1]
```

[30]:
```
Electrical        0.034165
Garage Area       0.034165
Total Bsmt SF     0.034165
Bsmt Unf SF       0.034165
BsmtFin SF 1      0.034165
BsmtFin SF 2      0.034165
Garage Cars       0.034165
Bsmt Full Bath    0.068329
Bsmt Half Bath    0.068329
Mas Vnr Area      0.785787
Mas Vnr Type      0.785787
dtype: float64
```

[31]:
```
# shows that one row is missing Garage Area and there is another (different)␣
↪row is missing Electrical

# df[df['Electrical'].isnull()]['Electrical']
df[df['Electrical'].isnull()]['Garage Area']
# df[df['Garage Area'].isnull()]['Electrical']
```

```
[31]: 1577    400.0
      Name: Garage Area, dtype: float64
```

```
[32]: df.columns
```

```
[32]: Index(['MS SubClass', 'MS Zoning', 'Lot Frontage', 'Lot Area', 'Street',
             'Alley', 'Lot Shape', 'Land Contour', 'Utilities', 'Lot Config',
             'Land Slope', 'Neighborhood', 'Condition 1', 'Condition 2', 'Bldg Type',
             'House Style', 'Overall Qual', 'Overall Cond', 'Year Built',
             'Year Remod/Add', 'Roof Style', 'Roof Matl', 'Exterior 1st',
             'Exterior 2nd', 'Mas Vnr Type', 'Mas Vnr Area', 'Exter Qual',
             'Exter Cond', 'Foundation', 'Bsmt Qual', 'Bsmt Cond', 'Bsmt Exposure',
             'BsmtFin Type 1', 'BsmtFin SF 1', 'BsmtFin Type 2', 'BsmtFin SF 2',
             'Bsmt Unf SF', 'Total Bsmt SF', 'Heating', 'Heating QC', 'Central Air',
             'Electrical', '1st Flr SF', '2nd Flr SF', 'Low Qual Fin SF',
             'Gr Liv Area', 'Bsmt Full Bath', 'Bsmt Half Bath', 'Full Bath',
             'Half Bath', 'Bedroom AbvGr', 'Kitchen AbvGr', 'Kitchen Qual',
             'TotRms AbvGrd', 'Functional', 'Fireplaces', 'Fireplace Qu',
             'Garage Type', 'Garage Yr Blt', 'Garage Finish', 'Garage Cars',
             'Garage Area', 'Garage Qual', 'Garage Cond', 'Paved Drive',
             'Wood Deck SF', 'Open Porch SF', 'Enclosed Porch', '3Ssn Porch',
             'Screen Porch', 'Pool Area', 'Pool QC', 'Fence', 'Misc Feature',
             'Misc Val', 'Mo Sold', 'Yr Sold', 'Sale Type', 'Sale Condition',
             'SalePrice'],
            dtype='object')
```

```
[33]: df[df['Garage Area'].isnull()][['Garage Type', 'Garage Cars', 'Garage Qual']]
```

```
[33]:      Garage Type  Garage Cars Garage Qual
      2236      Detchd          NaN         NaN
```

```
[34]: df['Garage Area'].describe()
```

```
[34]: count    2926.000000
      mean      472.123377
      std       213.939485
      min         0.000000
      25%       320.000000
      50%       480.000000
      75%       576.000000
      max      1488.000000
      Name: Garage Area, dtype: float64
```

```
[35]: # there are two observations (rows) where the basement half bath is missing
      df[df['Bsmt Half Bath'].isnull()]
```

[35]:
```
      MS SubClass MS Zoning  Lot Frontage  Lot Area Street Alley Lot Shape  \
1341           20        RM          99.0      5940   Pave   NaN       IR1
1497           20        RL         123.0     47007   Pave   NaN       IR1

     Land Contour Utilities Lot Config  … Pool Area Pool QC  Fence  \
1341          Lvl    AllPub        FR3  …         0     NaN  MnPrv
1497          Lvl    AllPub     Inside  …         0     NaN    NaN

     Misc Feature Misc Val Mo Sold  Yr Sold  Sale Type  Sale Condition  \
1341          NaN        0       4     2008      ConLD         Abnorml
1497          NaN        0       7     2008         WD          Normal

      SalePrice
1341      79000
1497     284700

[2 rows x 80 columns]
```

[36]:
```
# our goal is to keep as much data as possible and not lose many observations␣
↪(rows)...

# but, we will start by dropping a single row at a time since we cannot provide␣
↪a reasonable estimate for these two features:
df = df.dropna(axis=0, subset=['Electrical', 'Garage Area'])
```

[37]:
```
percent_nan = percent_missing(df)
percent_nan[percent_nan < 1]
```

[37]:
```
Bsmt Unf SF       0.034188
Total Bsmt SF     0.034188
BsmtFin SF 2      0.034188
BsmtFin SF 1      0.034188
Bsmt Full Bath    0.068376
Bsmt Half Bath    0.068376
Mas Vnr Type      0.786325
Mas Vnr Area      0.786325
dtype: float64
```

By looking at the percent of missing values above (where the percent missing is less than 1%), we can see that many of these features are related to basements: the smallest six values in the cell above are all associated with basements (”Bsmt”). It seems that these missing values are related to the fact that some homes do not have basements. We can use this knowledge to fill in such missing values with zero (e.g., for features like total basement area: if there is no basement, then the area can logically be set to zero).

[38]:
```
df[df['Bsmt Half Bath'].isnull()]
```

[38]:
```
      MS SubClass MS Zoning  Lot Frontage  Lot Area Street Alley Lot Shape  \
1341           20        RM          99.0      5940   Pave   NaN       IR1
1497           20        RL         123.0     47007   Pave   NaN       IR1

     Land Contour Utilities Lot Config  … Pool Area Pool QC  Fence  \
1341          Lvl    AllPub        FR3  …         0     NaN  MnPrv
1497          Lvl    AllPub     Inside  …         0     NaN    NaN

     Misc Feature Misc Val Mo Sold  Yr Sold  Sale Type  Sale Condition  \
1341          NaN        0       4     2008      ConLD         Abnorml
1497          NaN        0       7     2008        WD           Normal

      SalePrice
1341      79000
1497     284700

[2 rows x 80 columns]
```

[39]:
```
df[df['Bsmt Full Bath'].isnull()]['Total Bsmt SF']
```

[39]:
```
1341    NaN
1497    0.0
Name: Total Bsmt SF, dtype: float64
```

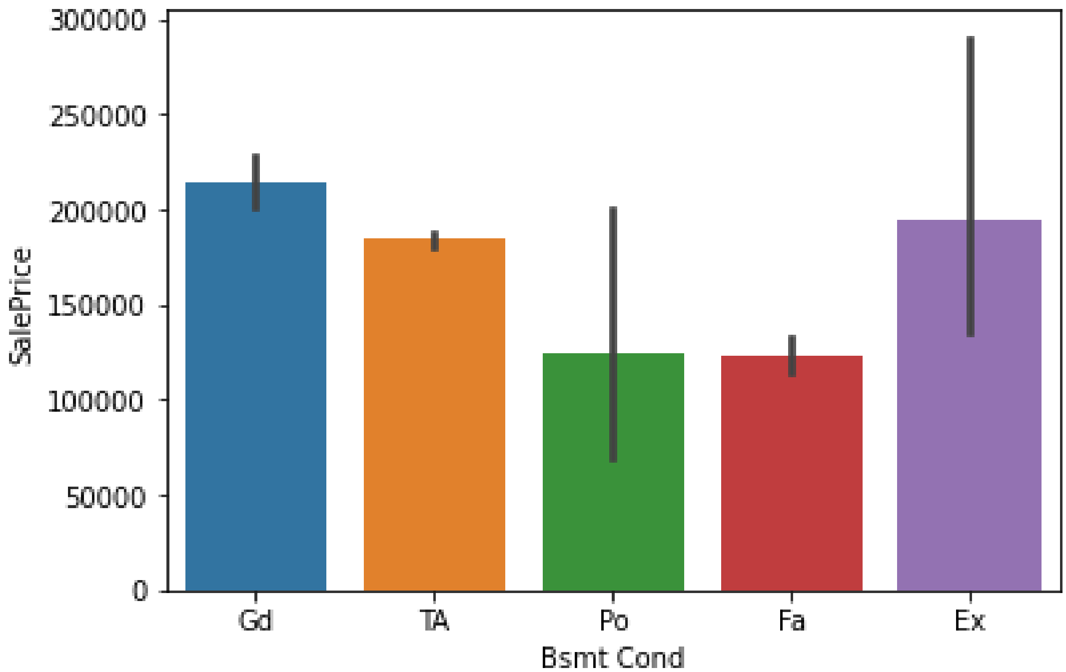
[40]:
```
df[df['Bsmt Cond'].isnull()]['Total Bsmt SF'].isnull().sum()
```

[40]:
```
1
```

[41]:
```
df[df['Bsmt Unf SF'].isnull()]
```

[41]:
```
      MS SubClass MS Zoning  Lot Frontage  Lot Area Street Alley Lot Shape  \
1341           20        RM          99.0      5940   Pave   NaN       IR1

     Land Contour Utilities Lot Config  … Pool Area Pool QC  Fence  \
1341          Lvl    AllPub        FR3  …         0     NaN  MnPrv

     Misc Feature Misc Val Mo Sold  Yr Sold  Sale Type  Sale Condition  \
1341          NaN        0       4     2008      ConLD         Abnorml

      SalePrice
1341      79000

[1 rows x 80 columns]
```

[42]:
```
sns.barplot(data=df, y='SalePrice', x='Bsmt Cond');
```



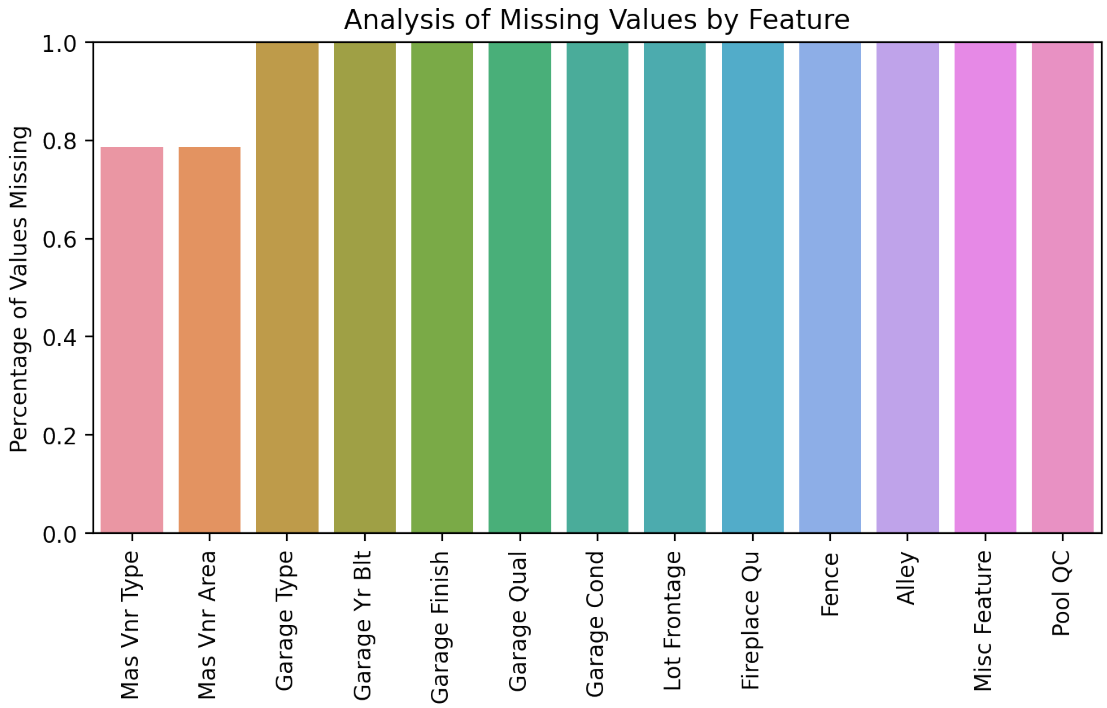
[43]:
```
bsmt_cat_cols = ['Bsmt Qual', 'Bsmt Cond', 'Bsmt Exposure', 'BsmtFin Type 1',␣
↪'BsmtFin Type 2']
df['Bsmt Qual'].unique()
```

[43]:
```
array(['TA', 'Gd', 'Ex', nan, 'Fa', 'Po'], dtype=object)
```

[44]:
```
# BASEMENT NUMERIC COLUMNS --> fillna 0
bsmt_num_cols = ['BsmtFin SF 1', 'BsmtFin SF 2', 'Bsmt Unf SF', 'Total Bsmt␣
↪SF', 'Bsmt Full Bath', 'Bsmt Half Bath']
df[bsmt_num_cols] = df[bsmt_num_cols].fillna(0)

# BASEMENT STRING COLUMNS
bsmt_cat_cols = ['Bsmt Qual', 'Bsmt Cond', 'Bsmt Exposure', 'BsmtFin Type 1',␣
↪'BsmtFin Type 2']
df[bsmt_cat_cols] = df[bsmt_cat_cols].fillna('None')
```

[45]:
```
# bar plot of the percentages of missing values: ZOOMED IN

# UPDATE MISSING VALUES
percent_nan = percent_missing(df)

plt.figure(figsize=(8,4), dpi=200)
sns.barplot(x=percent_nan.index, y=percent_nan)
plt.xticks(rotation=90)
```

```
plt.ylim(0, 1)
plt.title('Analysis of Missing Values by Feature')
plt.ylabel('Percentage of Values Missing')

plt.show()
```



[46]:
```
# we do a similar data cleaning here for these masonry features
# (we assume a missing value for these features means that the home does not␣
↪have them)
df['Mas Vnr Type'] = df['Mas Vnr Type'].fillna("None")
df['Mas Vnr Area'] = df['Mas Vnr Area'].fillna(0)
```

[47]:
```
# bar plot of the percentages of missing values
def plot_missing_values(data=df, y_min=0, y_max=100):

    # UPDATE MISSING VALUES
    percent_nan = percent_missing(data)

    plt.figure(figsize=(8,4), dpi=200)
    sns.barplot(x=percent_nan.index, y=percent_nan)
    plt.xticks(rotation=90)
    plt.ylim(y_min, y_max)
    plt.title('Analysis of Missing Values by Feature')
    plt.ylabel('Percentage of Values Missing')
```

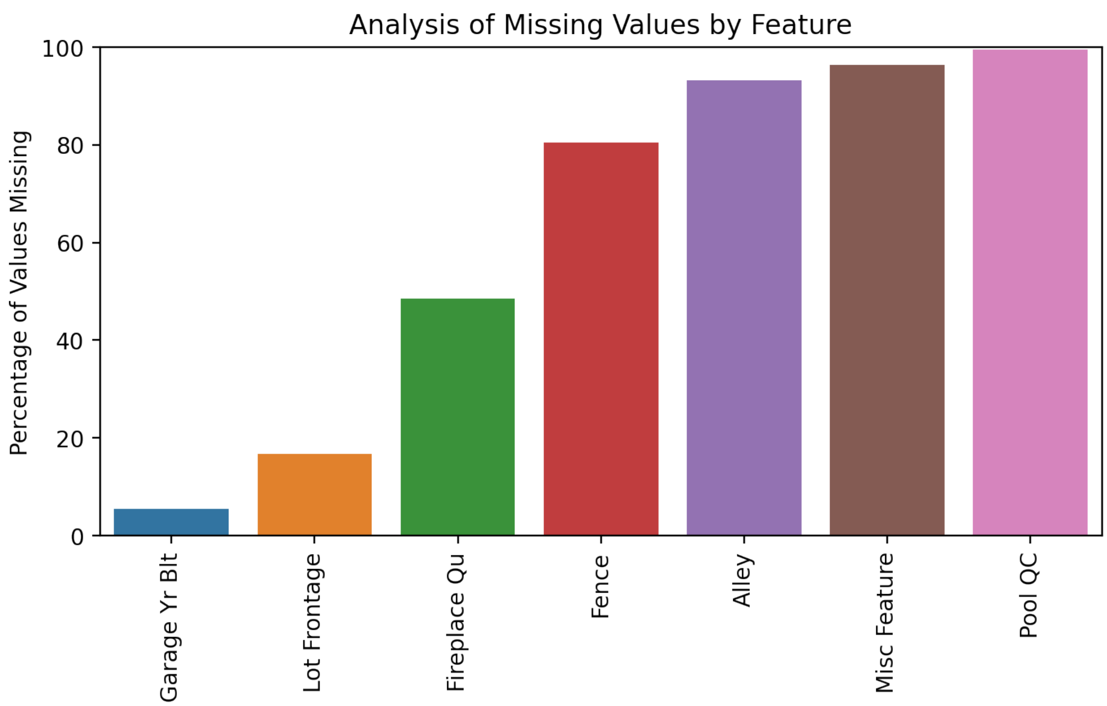
```
plt.show()
```

1.3.1 Fixing Data in Columns (Features)

Two approaches to consider: * Fill in missing values * Drop the feature column altogether

[48]:
```
# For the garage features, it seems that the missing values correspond to␣
↪houses that do not have garages.
# Therefore, we can replace the missing categorical data related to garages␣
↪with the value "None"
# to indicate that the home does not have a garage.
garage_cat_cols = ['Garage Type', 'Garage Finish', 'Garage Qual', 'Garage Cond']
df[garage_cat_cols] = df[garage_cat_cols].fillna('None')
```

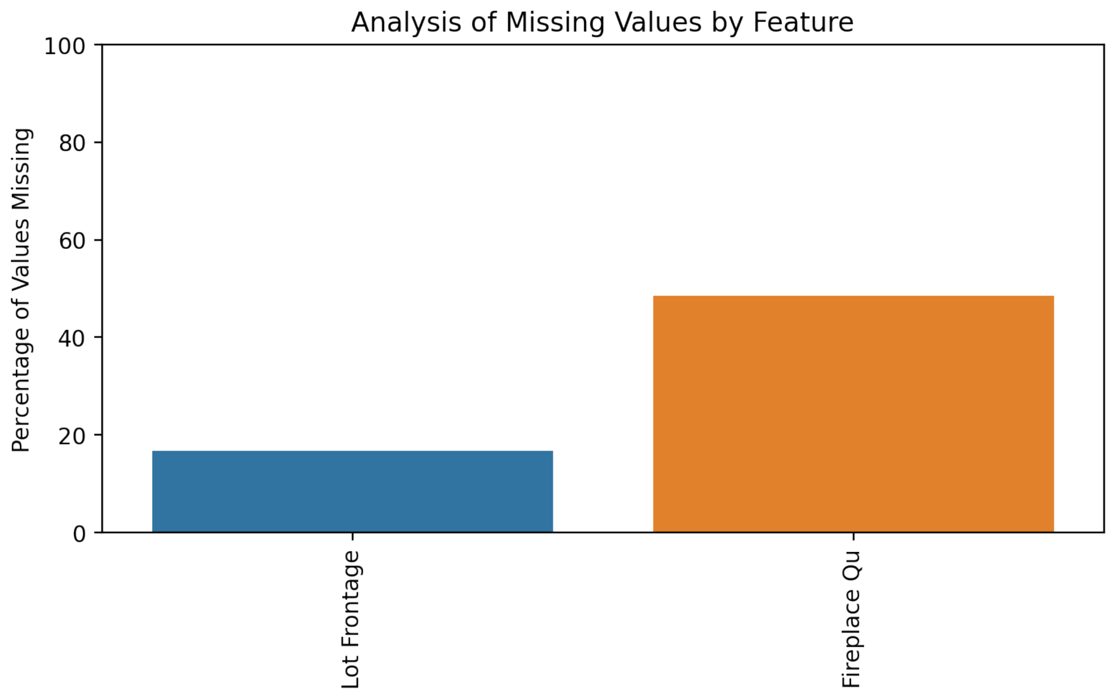
[49]:
```
plot_missing_values(data=df)
```



[50]:
```
# What should we do with the Garage Year Built?

# We can use Year = 0
# OR use the average Garage Year Built from the other data
df['Garage Yr Blt'] = df['Garage Yr Blt'].fillna(0)
```

[51]:
```
# Since we have such a large percentage of missing values in the following␣
↪features:
#   Pool QC      ~100%
#   Misc Feature  >95%
#   Alley         >90%
#   Fence         >80%
#... we decide to DROP these features from the data set here
df = df.drop(['Pool QC', 'Misc Feature', 'Alley', 'Fence'], axis=1)
```

[52]:
```
plot_missing_values(df)
```



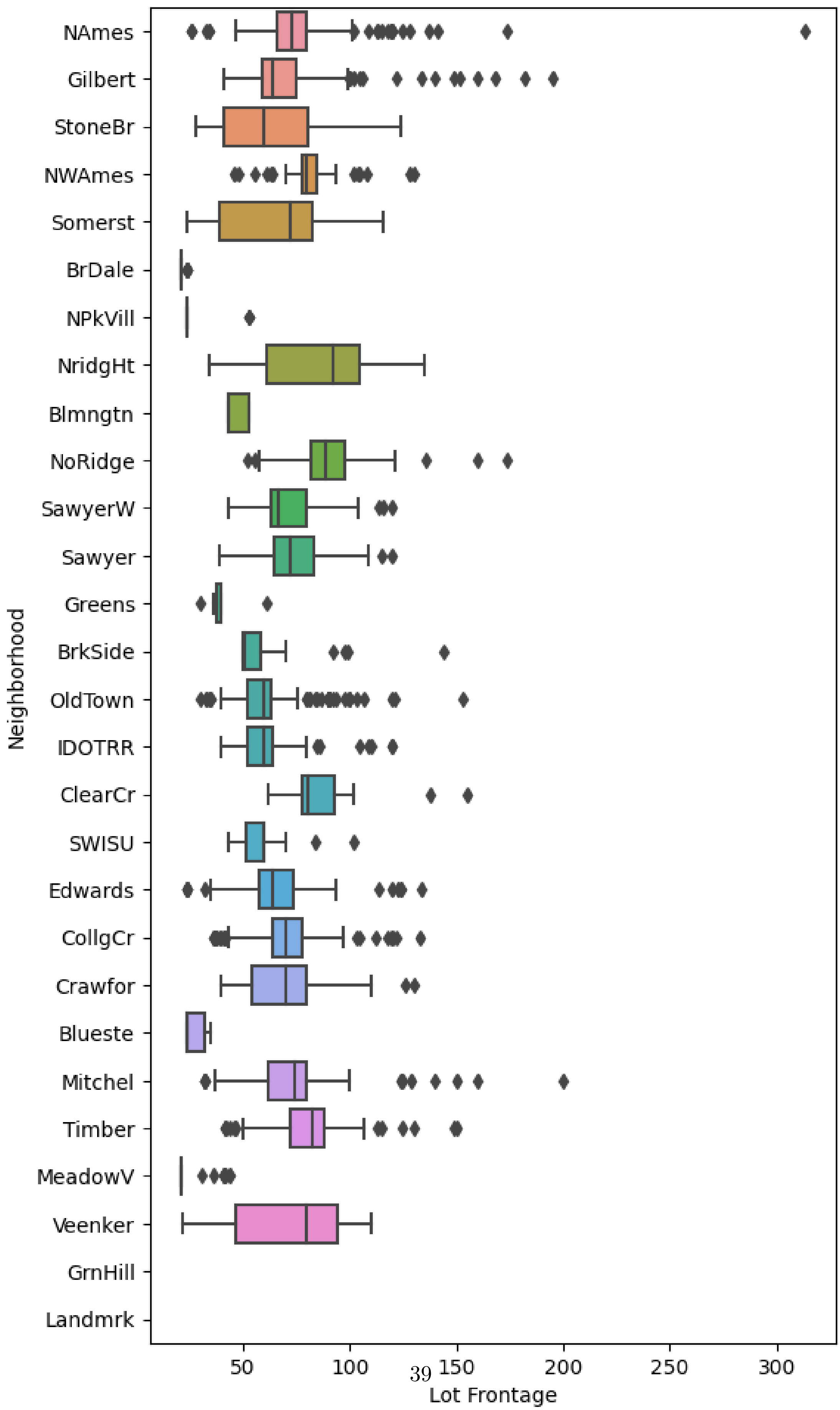
[53]:
```
df['Fireplace Qu'].value_counts()
```

[53]:
```
Gd    741
TA    600
Fa     75
Po     46
Ex     43
Name: Fireplace Qu, dtype: int64
```

[54]:
```
# the null values for fireplace must correspond to "No Fireplace" (as seen in␣
↪the description of the data set earlier)
df['Fireplace Qu'] = df['Fireplace Qu'].fillna('None')
```

The final feature with missing values now is the ”lot frontage” feature.

[55]:
```
# here we can see the lot frontage values according to neighborhood
plt.figure(figsize=(6,12), dpi=100)
sns.boxplot(x='Lot Frontage', y='Neighborhood', data=df, orient='h');
```

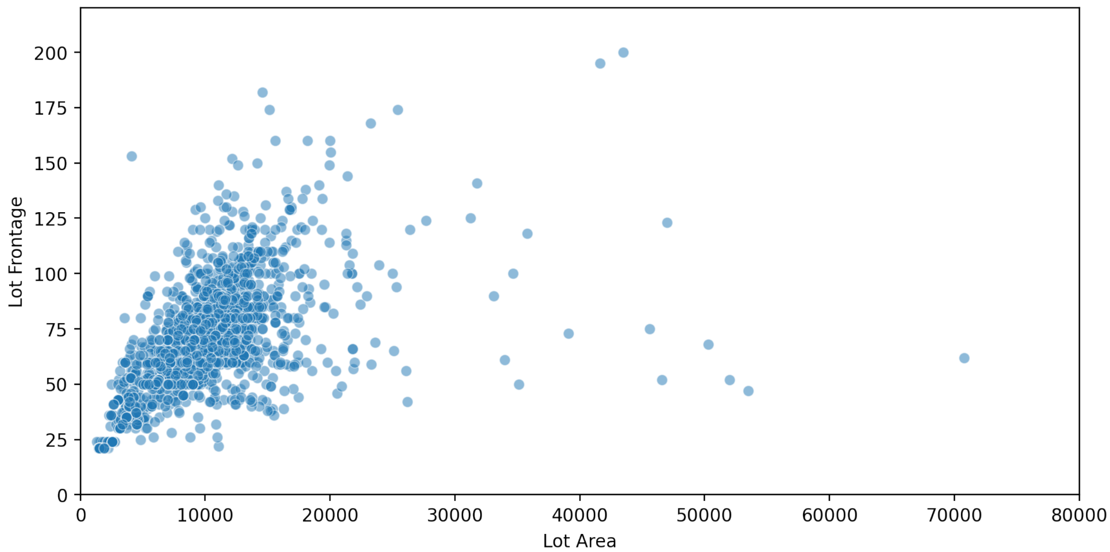

NAmes
Gilbert
StoneBr
NWAmes
Somerst
BrDale
NPkVill
NridgHt
Blmngtn
NoRidge
SawyerW
Sawyer
Greens
BrkSide
OldTown
IDOTRR
ClearCr
SWISU
Edwards
CollgCr
Crawfor
Blueste
Mitchel
Timber
MeadowV
Veenker
GrnHill
Landmrk
Neighborhood
50
100
150
200
250
300
Lot Frontage

[56]:
```
# mean lot frontage by neighborhood
df.groupby('Neighborhood')['Lot Frontage'].mean()
```

[56]:
```
Neighborhood
Blmngtn    46.900000
Blueste    27.300000
BrDale     21.500000
BrkSide    55.789474
ClearCr    88.150000
CollgCr    71.336364
Crawfor    69.951807
Edwards    64.794286
Gilbert    74.207207
Greens     41.000000
GrnHill          NaN
IDOTRR     62.383721
Landmrk          NaN
MeadowV    25.606061
Mitchel    75.144444
NAmes      75.210667
NPkVill    28.142857
NWAmes     81.517647
NoRidge    91.629630
NridgHt    84.184049
OldTown    61.777293
SWISU      59.068182
Sawyer     74.551020
SawyerW    70.669811
Somerst    64.549383
StoneBr    62.173913
Timber     81.303571
Veenker    72.000000
Name: Lot Frontage, dtype: float64
```

For the missing values in the lot frontage area, we will use the average frontage area for homes in the same neighborhood.

[57]:
```
plt.figure(figsize=(10,5), dpi=200)
sns.scatterplot(data=df, x='Lot Area', y='Lot Frontage', alpha=0.5)
plt.xlim(0, 80000)
plt.ylim(0, 220)
plt.show()
```



[58]:
```
df_temp = df[df['Lot Frontage'].notnull()]
df_temp.corr()['Lot Frontage'].sort_values()
```

[58]:
```
MS SubClass       -0.430521
Overall Cond      -0.072866
Bsmt Half Bath    -0.028655
Yr Sold           -0.007396
Low Qual Fin SF    0.006049
Kitchen AbvGr      0.006890
Misc Val           0.014022
Mo Sold            0.016471
Enclosed Porch     0.016562
2nd Flr SF         0.021647
3Ssn Porch         0.029928
Half Bath          0.034289
BsmtFin SF 2       0.048845
Screen Porch       0.080470
Year Remod/Add     0.087098
Bsmt Full Bath     0.094725
Garage Yr Blt      0.103244
Wood Deck SF       0.114204
Year Built         0.116581
Bsmt Unf SF        0.118046
Pool Area          0.125000
Open Porch SF      0.141202
BsmtFin SF 1       0.165814
Full Bath          0.182655
Overall Qual       0.200698
```

```
Mas Vnr Area       0.201185
Fireplaces         0.244824
Bedroom AbvGr      0.246874
Garage Cars        0.312195
Total Bsmt SF      0.312418
TotRms AbvGrd      0.341440
Garage Area        0.345994
Gr Liv Area        0.355336
SalePrice          0.367518
1st Flr SF         0.432625
Lot Area           0.468168
Lot Frontage       1.000000
Name: Lot Frontage, dtype: float64
```

[59]:
```
# For missing lot frontage values, we will fill in the null values with the␣
↪mean lot frontage for that neighborhood.

# use pandas.DataFrame.transform: combine groupby and apply methods
df['Lot Frontage'] = df.groupby('Neighborhood')['Lot Frontage'].
↪transform(lambda val: val.fillna(val.mean()))
```

[60]:
```
df.isnull().sum()
```

[60]:
```
MS SubClass       0
MS Zoning         0
Lot Frontage      3
Lot Area          0
Street            0
                 ..
Mo Sold           0
Yr Sold           0
Sale Type         0
Sale Condition    0
SalePrice         0
Length: 76, dtype: int64
```

[61]:
```
# There are 3 remaining values missing for the lot frontage:
# So we will fill these with the mean lot frontage value for the full data set..
↪.

mean_lot_frontage = df['Lot Frontage'].mean()
df['Lot Frontage'] = df['Lot Frontage'].fillna(mean_lot_frontage)
```

[62]:
```
percent_missing(df)
```

[62]:
```
Series([], dtype: float64)
```

[63]:
```
df.reset_index(inplace=True)
```

[64]:
```
df.columns

# note: if there is an extra column like 'level_0' or 'index' inserted here,␣
↪you can drop it the next cell.
```

[64]:
```
Index(['index', 'MS SubClass', 'MS Zoning', 'Lot Frontage', 'Lot Area',
       'Street', 'Lot Shape', 'Land Contour', 'Utilities', 'Lot Config',
       'Land Slope', 'Neighborhood', 'Condition 1', 'Condition 2', 'Bldg Type',
       'House Style', 'Overall Qual', 'Overall Cond', 'Year Built',
       'Year Remod/Add', 'Roof Style', 'Roof Matl', 'Exterior 1st',
       'Exterior 2nd', 'Mas Vnr Type', 'Mas Vnr Area', 'Exter Qual',
       'Exter Cond', 'Foundation', 'Bsmt Qual', 'Bsmt Cond', 'Bsmt Exposure',
       'BsmtFin Type 1', 'BsmtFin SF 1', 'BsmtFin Type 2', 'BsmtFin SF 2',
       'Bsmt Unf SF', 'Total Bsmt SF', 'Heating', 'Heating QC', 'Central Air',
       'Electrical', '1st Flr SF', '2nd Flr SF', 'Low Qual Fin SF',
       'Gr Liv Area', 'Bsmt Full Bath', 'Bsmt Half Bath', 'Full Bath',
       'Half Bath', 'Bedroom AbvGr', 'Kitchen AbvGr', 'Kitchen Qual',
       'TotRms AbvGrd', 'Functional', 'Fireplaces', 'Fireplace Qu',
       'Garage Type', 'Garage Yr Blt', 'Garage Finish', 'Garage Cars',
       'Garage Area', 'Garage Qual', 'Garage Cond', 'Paved Drive',
       'Wood Deck SF', 'Open Porch SF', 'Enclosed Porch', '3Ssn Porch',
       'Screen Porch', 'Pool Area', 'Misc Val', 'Mo Sold', 'Yr Sold',
       'Sale Type', 'Sale Condition', 'SalePrice'],
      dtype='object')
```

[65]:
```
# only run this if necessary ...

# drop_these_cols = ['level_0', 'index']
drop_these_cols = ['index']

df.drop(drop_these_cols, axis=1, inplace=True)
```

At this point, we have now handled all the missing values in the Ames housing data set.

Original Data: 2930 rows with 81 columns

NEW Data with No Missing Values: 2925 rows with 76 columns

[66]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2925 entries, 0 to 2924
Data columns (total 76 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   MS SubClass      2925 non-null   int64
 1   MS Zoning        2925 non-null   object
```

2	Lot Frontage	2925	non-null	float64
3	Lot Area	2925	non-null	int64
4	Street	2925	non-null	object
5	Lot Shape	2925	non-null	object
6	Land Contour	2925	non-null	object
7	Utilities	2925	non-null	object
8	Lot Config	2925	non-null	object
9	Land Slope	2925	non-null	object
10	Neighborhood	2925	non-null	object
11	Condition 1	2925	non-null	object
12	Condition 2	2925	non-null	object
13	Bldg Type	2925	non-null	object
14	House Style	2925	non-null	object
15	Overall Qual	2925	non-null	int64
16	Overall Cond	2925	non-null	int64
17	Year Built	2925	non-null	int64
18	Year Remod/Add	2925	non-null	int64
19	Roof Style	2925	non-null	object
20	Roof Matl	2925	non-null	object
21	Exterior 1st	2925	non-null	object
22	Exterior 2nd	2925	non-null	object
23	Mas Vnr Type	2925	non-null	object
24	Mas Vnr Area	2925	non-null	float64
25	Exter Qual	2925	non-null	object
26	Exter Cond	2925	non-null	object
27	Foundation	2925	non-null	object
28	Bsmt Qual	2925	non-null	object
29	Bsmt Cond	2925	non-null	object
30	Bsmt Exposure	2925	non-null	object
31	BsmtFin Type 1	2925	non-null	object
32	BsmtFin SF 1	2925	non-null	float64
33	BsmtFin Type 2	2925	non-null	object
34	BsmtFin SF 2	2925	non-null	float64
35	Bsmt Unf SF	2925	non-null	float64
36	Total Bsmt SF	2925	non-null	float64
37	Heating	2925	non-null	object
38	Heating QC	2925	non-null	object
39	Central Air	2925	non-null	object
40	Electrical	2925	non-null	object
41	1st Flr SF	2925	non-null	int64
42	2nd Flr SF	2925	non-null	int64
43	Low Qual Fin SF	2925	non-null	int64
44	Gr Liv Area	2925	non-null	int64
45	Bsmt Full Bath	2925	non-null	float64
46	Bsmt Half Bath	2925	non-null	float64
47	Full Bath	2925	non-null	int64
48	Half Bath	2925	non-null	int64
49	Bedroom AbvGr	2925	non-null	int64

```
 50  Kitchen AbvGr    2925 non-null   int64
 51  Kitchen Qual     2925 non-null   object
 52  TotRms AbvGrd    2925 non-null   int64
 53  Functional       2925 non-null   object
 54  Fireplaces       2925 non-null   int64
 55  Fireplace Qu     2925 non-null   object
 56  Garage Type      2925 non-null   object
 57  Garage Yr Blt    2925 non-null   float64
 58  Garage Finish    2925 non-null   object
 59  Garage Cars      2925 non-null   float64
 60  Garage Area      2925 non-null   float64
 61  Garage Qual      2925 non-null   object
 62  Garage Cond      2925 non-null   object
 63  Paved Drive      2925 non-null   object
 64  Wood Deck SF     2925 non-null   int64
 65  Open Porch SF    2925 non-null   int64
 66  Enclosed Porch   2925 non-null   int64
 67  3Ssn Porch       2925 non-null   int64
 68  Screen Porch     2925 non-null   int64
 69  Pool Area        2925 non-null   int64
 70  Misc Val         2925 non-null   int64
 71  Mo Sold          2925 non-null   int64
 72  Yr Sold          2925 non-null   int64
 73  Sale Type        2925 non-null   object
 74  Sale Condition   2925 non-null   object
 75  SalePrice        2925 non-null   int64
dtypes: float64(11), int64(26), object(39)
memory usage: 1.7+ MB
```

[67]:
```
# CHECKPOINT #2
df_clean = df.copy()
```

1.4 Part 3: Handling Categorical Data

[68]:
```
# After reviewing our data set, we see that the MS SubClass feature is entered␣
↪as "numeric" data,
# but in reality, these numbers do not have a linear relationship: they act as␣
↪categories.
df['MS SubClass'] = df['MS SubClass'].apply(str)
```

[69]:
```
# split our data into categorical and numerical
df_categorical = df.select_dtypes(include='object')
df_numerical = df.select_dtypes(exclude='object')
```

[70]:
```python
# create dummy variables from the categorical data only
df_dummies = pd.get_dummies(df_categorical, drop_first=True)
df_dummies
```

[70]:
```
      MS SubClass_150  MS SubClass_160  MS SubClass_180  MS SubClass_190  \
0                   0                0                0                0
1                   0                0                0                0
2                   0                0                0                0
3                   0                0                0                0
4                   0                0                0                0
…                 …                …                …                …
2920                0                0                0                0
2921                0                0                0                0
2922                0                0                0                0
2923                0                0                0                0
2924                0                0                0                0

      MS SubClass_20  MS SubClass_30  MS SubClass_40  MS SubClass_45  \
0                  1               0               0               0
1                  1               0               0               0
2                  1               0               0               0
3                  1               0               0               0
4                  0               0               0               0
…                …               …               …               …
2920               0               0               0               0
2921               1               0               0               0
2922               0               0               0               0
2923               1               0               0               0
2924               0               0               0               0

      MS SubClass_50  MS SubClass_60  …  Sale Type_ConLw  Sale Type_New  \
0                  0               0  …                0              0
1                  0               0  …                0              0
2                  0               0  …                0              0
3                  0               0  …                0              0
4                  0               1  …                0              0
…                …               …  …              …            …
2920               0               0  …                0              0
2921               0               0  …                0              0
2922               0               0  …                0              0
2923               0               0  …                0              0
2924               0               1  …                0              0

      Sale Type_Oth  Sale Type_VWD  Sale Type_WD   Sale Condition_AdjLand  \
0                 0              0              1                       0
1                 0              0              1                       0
2                 0              0              1                       0
```

```
3                 0             0             1                       0
4                 0             0             1                       0
…            …            …            …                    …
2920              0             0             1                       0
2921              0             0             1                       0
2922              0             0             1                       0
2923              0             0             1                       0
2924              0             0             1                       0

      Sale Condition_Alloca  Sale Condition_Family  Sale Condition_Normal  \
0                         0                      0                      1
1                         0                      0                      1
2                         0                      0                      1
3                         0                      0                      1
4                         0                      0                      1
…                      …                      …                      …
2920                      0                      0                      1
2921                      0                      0                      1
2922                      0                      0                      1
2923                      0                      0                      1
2924                      0                      0                      1

      Sale Condition_Partial
0                          0
1                          0
2                          0
3                          0
4                          0
…                       …
2920                       0
2921                       0
2922                       0
2923                       0
2924                       0

[2925 rows x 238 columns]
```

[71]:
```python
df_final = pd.concat([df_numerical, df_dummies], axis=1)
df_final
```

[71]:
```
      Lot Frontage  Lot Area  Overall Qual  Overall Cond  Year Built  \
0       141.000000     31770             6             5        1960
1        80.000000     11622             5             6        1961
2        81.000000     14267             6             6        1958
3        93.000000     11160             7             5        1968
4        74.000000     13830             5             5        1997
…              …         …            …            …          …
```

```
2920   37.000000   7937            6            6        1984
2921   75.144444   8885            5            5        1983
2922   62.000000  10441            5            5        1992
2923   77.000000  10010            5            5        1974
2924   74.000000   9627            7            5        1993

      Year Remod/Add  Mas Vnr Area  BsmtFin SF 1  BsmtFin SF 2  Bsmt Unf SF  \
0               1960         112.0         639.0           0.0        441.0
1               1961           0.0         468.0         144.0        270.0
2               1958         108.0         923.0           0.0        406.0
3               1968           0.0        1065.0           0.0       1045.0
4               1998           0.0         791.0           0.0        137.0
…                  …            …            …            …            …
2920            1984           0.0         819.0           0.0        184.0
2921            1983           0.0         301.0         324.0        239.0
2922            1992           0.0         337.0           0.0        575.0
2923            1975           0.0        1071.0         123.0        195.0
2924            1994          94.0         758.0           0.0        238.0

      …  Sale Type_ConLw  Sale Type_New  Sale Type_Oth  Sale Type_VWD  \
0     …                0              0              0              0
1     …                0              0              0              0
2     …                0              0              0              0
3     …                0              0              0              0
4     …                0              0              0              0
…   …              …             …             …             …
2920  …                0              0              0              0
2921  …                0              0              0              0
2922  …                0              0              0              0
2923  …                0              0              0              0
2924  …                0              0              0              0

      Sale Type_WD   Sale Condition_AdjLand  Sale Condition_Alloca  \
0                 1                       0                      0
1                 1                       0                      0
2                 1                       0                      0
3                 1                       0                      0
4                 1                       0                      0
…              …                      …                     …
2920              1                       0                      0
2921              1                       0                      0
2922              1                       0                      0
2923              1                       0                      0
2924              1                       0                      0

      Sale Condition_Family  Sale Condition_Normal  Sale Condition_Partial
0                         0                      1                       0
```

```
1                        0                      1                        0
2                        0                      1                        0
3                        0                      1                        0
4                        0                      1                        0
…                     …                     …                      …
2920                     0                      1                        0
2921                     0                      1                        0
2922                     0                      1                        0
2923                     0                      1                        0
2924                     0                      1                        0

[2925 rows x 274 columns]
```

[72]:
```
# We can now check which features have the highest correlation with the sale␣
↪price
# (including categorical data as dummy variables)

# df_final.corr()['SalePrice'].sort_values()
np.abs(df_final.corr()['SalePrice']).nlargest(10)
```

[72]:
```
SalePrice        1.000000
Overall Qual     0.802637
Gr Liv Area      0.727279
Total Bsmt SF    0.660983
Garage Cars      0.648488
1st Flr SF       0.645635
Garage Area      0.644368
Exter Qual_TA    0.591459
Year Built       0.559165
Full Bath        0.546645
Name: SalePrice, dtype: float64
```

[73]:
```
# SAVE FINAL DATAFRAME AS OUTPUT
df_final.to_csv('./data_processed/Ames_Housing_Data_Clean_Dummies.csv',␣
↪index=False)
```