E1 - Análise de dados e Regressão Linear: Preço de imóveis

Considere a base de dados "Ames Housing Dataset". Com o objetivo de desenvolver um modelo de predição do preço do imóvel, desenvolva os itens a seguir e entregue a análise em arquivo do tipo powerpoint ou pdf

- Importando as bibliotecas

```
import pandas as pd
import numpy as np
import statsmodels
import seaborn
from matplotlib import pyplot as plt
pd.options.display.max_columns = 100
```

```
/Users/karinseeder/anaconda3/lib/python3.7/site-
packages/pandas/compat/_optional.py:138: UserWarning: Pandas requires
version '2.7.0' or newer of 'numexpr' (version
'2.6.8' currently installed).
  warnings.warn(msg, UserWarning)
```

- Importando a base de dados

```
df = pd.read_csv('base_1ah.csv')
print(df.shape)
```

```
(1460, 81)
```

- Visualizando uma amostra da base_1ah

```
df.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotC
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Insid
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Insid
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corn
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2

```
df = df.set_index('Id')
```

1 - Análise descritiva de variáveis

1.1 Estatísticas descritivas: frequência, proporção, média ($\bar{x}$), desvio padrão (s), quartis ($Q1$, $\tilde{x}$, $Q3$) (1,0)

Avaliando o tipo das variáveis na base_1ah

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1460 entries, 1 to 1460
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   MSSubClass     1460 non-null   int64
 1   MSZoning       1460 non-null   object
 2   LotFrontage    1201 non-null   float64
 3   LotArea        1460 non-null   int64
 4   Street         1460 non-null   object
 5   Alley          91 non-null     object
 6   LotShape       1460 non-null   object
```

```
 7   LandContour    1460 non-null   object
 8   Utilities      1460 non-null   object
 9   LotConfig      1460 non-null   object
 10  LandSlope      1460 non-null   object
 11  Neighborhood   1460 non-null   object
 12  Condition1     1460 non-null   object
 13  Condition2     1460 non-null   object
 14  BldgType       1460 non-null   object
 15  HouseStyle     1460 non-null   object
 16  OverallQual    1460 non-null   int64
 17  OverallCond    1460 non-null   int64
 18  YearBuilt      1460 non-null   int64
 19  YearRemodAdd   1460 non-null   int64
 20  RoofStyle      1460 non-null   object
 21  RoofMatl       1460 non-null   object
 22  Exterior1st    1460 non-null   object
 23  Exterior2nd    1460 non-null   object
 24  MasVnrType     1452 non-null   object
 25  MasVnrArea     1452 non-null   float64
 26  ExterQual      1460 non-null   object
 27  ExterCond      1460 non-null   object
 28  Foundation     1460 non-null   object
 29  BsmtQual       1423 non-null   object
 30  BsmtCond       1423 non-null   object
 31  BsmtExposure   1422 non-null   object
 32  BsmtFinType1   1423 non-null   object
 33  BsmtFinSF1     1460 non-null   int64
 34  BsmtFinType2   1422 non-null   object
 35  BsmtFinSF2     1460 non-null   int64
 36  BsmtUnfSF      1460 non-null   int64
 37  TotalBsmtSF    1460 non-null   int64
 38  Heating        1460 non-null   object
 39  HeatingQC      1460 non-null   object
 40  CentralAir     1460 non-null   object
 41  Electrical     1459 non-null   object
 42  1stFlrSF       1460 non-null   int64
 43  2ndFlrSF       1460 non-null   int64
 44  LowQualFinSF   1460 non-null   int64
 45  GrLivArea      1460 non-null   int64
 46  BsmtFullBath   1460 non-null   int64
 47  BsmtHalfBath   1460 non-null   int64
 48  FullBath       1460 non-null   int64
 49  HalfBath       1460 non-null   int64
 50  BedroomAbvGr   1460 non-null   int64
 51  KitchenAbvGr   1460 non-null   int64
 52  KitchenQual    1460 non-null   object
 53  TotRmsAbvGrd   1460 non-null   int64
 54  Functional     1460 non-null   object
 55  Fireplaces     1460 non-null   int64
 56  FireplaceQu    770 non-null    object
 57  GarageType     1379 non-null   object
 58  GarageYrBlt    1379 non-null   float64
 59  GarageFinish   1379 non-null   object
 60  GarageCars     1460 non-null   int64
 61  GarageArea     1460 non-null   int64
 62  GarageQual     1379 non-null   object
 63  GarageCond     1379 non-null   object
 64  PavedDrive     1460 non-null   object
 65  WoodDeckSF     1460 non-null   int64
 66  OpenPorchSF    1460 non-null   int64
 67  EnclosedPorch  1460 non-null   int64
 68  3SsnPorch      1460 non-null   int64
 69  ScreenPorch    1460 non-null   int64
 70  PoolArea       1460 non-null   int64
 71  PoolQC         7 non-null      object
 72  Fence          281 non-null    object
 73  MiscFeature    54 non-null     object
 74  MiscVal        1460 non-null   int64
 75  MoSold         1460 non-null   int64
 76  YrSold         1460 non-null   int64
 77  SaleType       1460 non-null   object
 78  SaleCondition  1460 non-null   object
 79  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(34), object(43)
memory usage: 923.9+ KB
```

Análise descritiva para variáveis numéricas

```
df.describe()
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	Ma
count	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	Ma
mean	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.6
std	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.0
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.00
25%	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.00
50%	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.00
75%	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.0
max	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600

Análise descritiva das variáveis categóricas

```
df.describe(include=object)
```

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Co
count	1460	1460	91	1460	1460	1460	1460	1460	1460	146
unique	5	2	2	4	4	2	5	3	25	9
top	RL	Pave	Grvl	Reg	Lvl	AllPub	Inside	Gtl	NAmes	No
freq	1151	1454	50	925	1311	1459	1052	1382	225	126

```
for i, x in enumerate(df.dtypes):
  if x == 'object':
    print(pd.crosstab(index=df[df.columns[i]], columns='freq', dropna=False))
    print('')
```

```
col_0     freq
MSZoning
C (all)     10
FV          65
RH          16
RL        1151
RM         218

col_0   freq
Street
Grvl       6
Pave    1454

col_0  freq
Alley
Grvl     50
Pave     41

col_0     freq
LotShape
IR1        484
IR2         41
IR3         10
Reg        925

col_0        freq
LandContour
Bnk            63
HLS            50
Low            36
Lvl          1311

col_0      freq
Utilities
AllPub     1459
NoSeWa        1
```

```
col_0      freq
LotConfig
Corner      263
CulDSac      94
FR2          47
FR3           4
Inside     1052

col_0      freq
LandSlope
Gtl        1382
Mod          65
Sev          13

col_0         freq
Neighborhood
Blmngtn         17
Blueste          2
BrDale          16
BrkSide         58
ClearCr         28
CollgCr        150
Crawfor         51
Edwards        100
Gilbert         79
IDOTRR          37
MeadowV         17
Mitchel         49
NAmes          225
NPkVill          9
NWAmes          73
NoRidge         41
NridgHt         77
OldTown        113
SWISU           25
Sawyer          74
SawyerW         59
Somerst         86
StoneBr         25
Timber          38
Veenker         11

col_0       freq
Condition1
Artery        48
Feedr         81
Norm        1260
PosA           8
PosN          19
RRAe          11
RRAn          26
RRNe           2
RRNn           5

col_0       freq
Condition2
Artery         2
Feedr          6
Norm        1445
PosA           1
PosN           2
RRAe           1
RRAn           1
RRNn           2

col_0     freq
BldgType
1Fam      1220
2fmCon      31
Duplex      52
Twnhs       43
TwnhsE     114

col_0       freq
HouseStyle
1.5Fin       154
1.5Unf        14
1Story       726
2.5Fin         8
2.5Unf        11
2Story       445
SFoyer        37
SLvl          65

col_0      freq
RoofStyle
Flat         13
```

```
Gable      1141
Gambrel      11
Hip         286
Mansard       7
Shed          2

col_0     freq
RoofMatl
ClyTile      1
CompShg   1434
Membran      1
Metal        1
Roll         1
Tar&Grv     11
WdShake      5
WdShngl      6

col_0        freq
Exterior1st
AsbShng        20
AsphShn         1
BrkComm         2
BrkFace        50
CBlock          1
CemntBd        61
HdBoard       222
ImStucc         1
MetalSd       220
Plywood       108
Stone           2
Stucco         25
VinylSd       515
Wd Sdng       206
WdShing        26

col_0        freq
Exterior2nd
AsbShng        20
AsphShn         3
Brk Cmn         7
BrkFace        25
CBlock          1
CmentBd        60
HdBoard       207
ImStucc        10
MetalSd       214
Other           1
Plywood       142
Stone           5
Stucco         26
VinylSd       504
Wd Sdng       197
Wd Shng        38

col_0       freq
MasVnrType
BrkCmn        15
BrkFace      445
None         864
Stone        128

col_0      freq
ExterQual
Ex           52
Fa           14
Gd          488
TA          906

col_0      freq
ExterCond
Ex            3
Fa           28
Gd          146
Po            1
TA         1282

col_0       freq
Foundation
BrkTil       146
CBlock       634
PConc        647
Slab          24
Stone          6
Wood           3

col_0     freq
BsmtQual
```

```
Ex         121
Fa          35
Gd         618
TA         649

col_0     freq
BsmtCond
Fa          45
Gd          65
Po           2
TA        1311

col_0         freq
BsmtExposure
Av             221
Gd             134
Mn             114
No             953

col_0         freq
BsmtFinType1
ALQ            220
BLQ            148
GLQ            418
LwQ             74
Rec            133
Unf            430

col_0         freq
BsmtFinType2
ALQ             19
BLQ             33
GLQ             14
LwQ             46
Rec             54
Unf           1256

col_0    freq
Heating
Floor       1
GasA     1428
GasW       18
Grav        7
OthW        2
Wall        4

col_0      freq
HeatingQC
Ex          741
Fa           49
Gd          241
Po            1
TA          428

col_0       freq
CentralAir
N             95
Y           1365

col_0       freq
Electrical
FuseA         94
FuseF         27
FuseP          3
Mix            1
SBrkr       1334

col_0        freq
KitchenQual
Ex            100
Fa             39
Gd            586
TA            735

col_0       freq
Functional
Maj1          14
Maj2           5
Min1          31
Min2          34
Mod           15
Sev            1
Typ         1360

col_0        freq
FireplaceQu
Ex             24
```

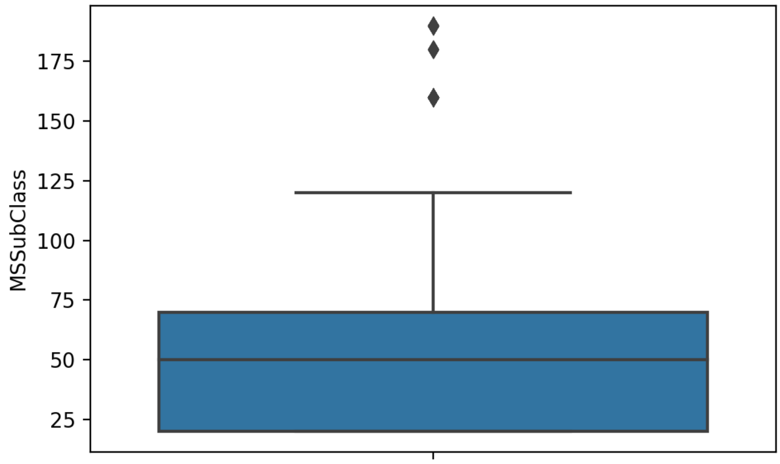
```
Fa             33
Gd            380
Po             20
TA            313

col_0       freq
GarageType
2Types         6
Attchd       870
Basment       19
BuiltIn       88
CarPort        9
Detchd       387

col_0         freq
GarageFinish
Fin            352
RFn            422
Unf            605

col_0       freq
GarageQual
Ex             3
Fa            48
Gd            14
Po             3
TA          1311

col_0       freq
GarageCond
Ex             2
Fa            35
Gd             9
Po             7
TA          1326

col_0       freq
PavedDrive
N             90
P             30
Y           1340

col_0   freq
PoolQC
Ex         2
Fa         2
Gd         3

col_0  freq
Fence
GdPrv    59
GdWo     54
MnPrv   157
MnWw     11

col_0        freq
MiscFeature
Gar2            2
Othr            2
Shed           49
TenC            1

col_0     freq
SaleType
COD         43
CWD          4
Con          2
ConLD        9
ConLI        5
ConLw        5
New        122
Oth          3
WD        1267

col_0          freq
SaleCondition
Abnorml         101
AdjLand           4
Alloca           12
Family           20
Normal         1198
Partial         125
```

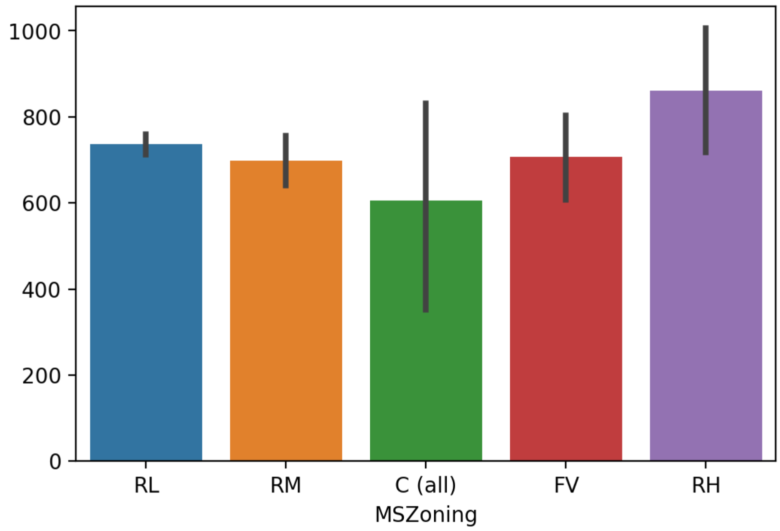
1.2 Gráficos como: Gráficos de colunas, BoxPlot, dispersão (2,0)

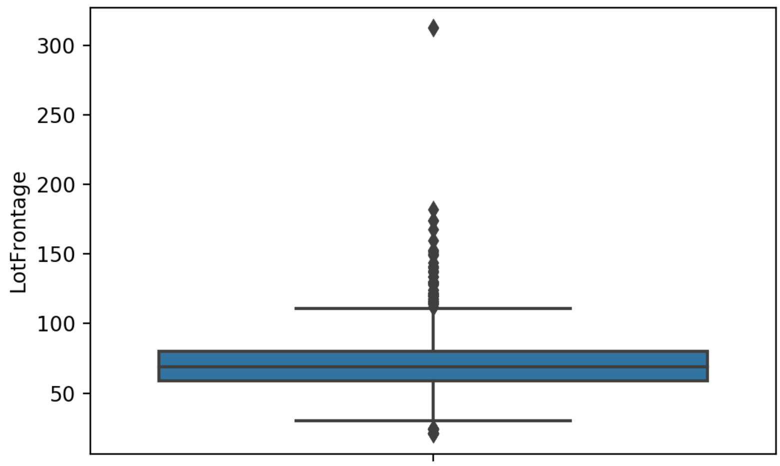
Para gráficos numéricos plotamos boxplot e para categóricos gráficos de barra.

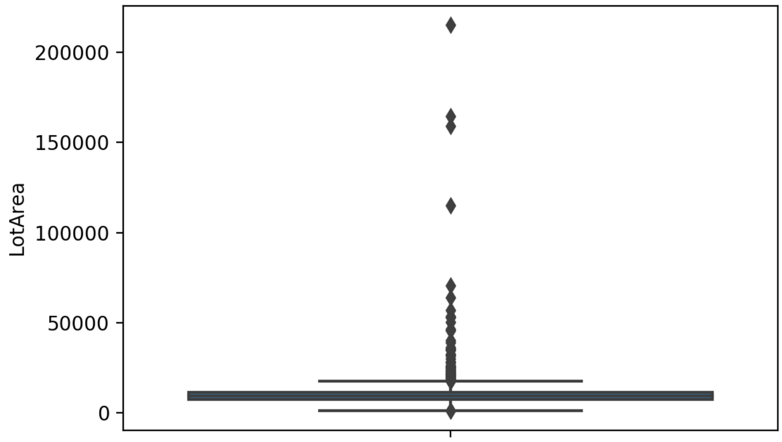
```
for i, x in enumerate(df.dtypes):
    if x == 'int64' or x == 'float64':
        plt.figure(i)
        seaborn.boxplot(y = df[df.columns[i]])
    elif x == 'int64' or x == 'object':
        plt.figure(i)
        seaborn.barplot(x = df[df.columns[i]], y = range(0,len(df)))
```

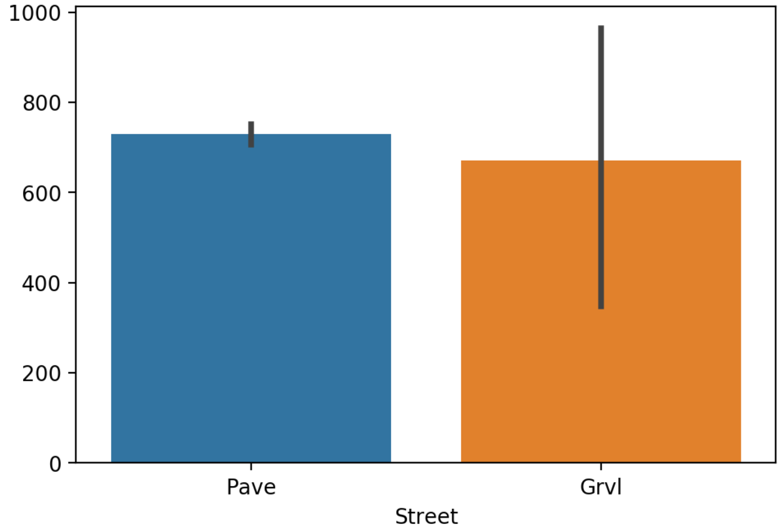
```
/Users/karinseeder/anaconda3/lib/python3.7/site-
packages/matplotlib/pyplot.py:514: RuntimeWarning: More than 20
figures have been opened. Figures created through the pyplot interface
(`matplotlib.pyplot.figure`) are retained until explicitly closed and
may consume too much memory. (To control this warning, see the rcParam
`figure.max_open_warning`).
  max_open_warning, RuntimeWarning)
```

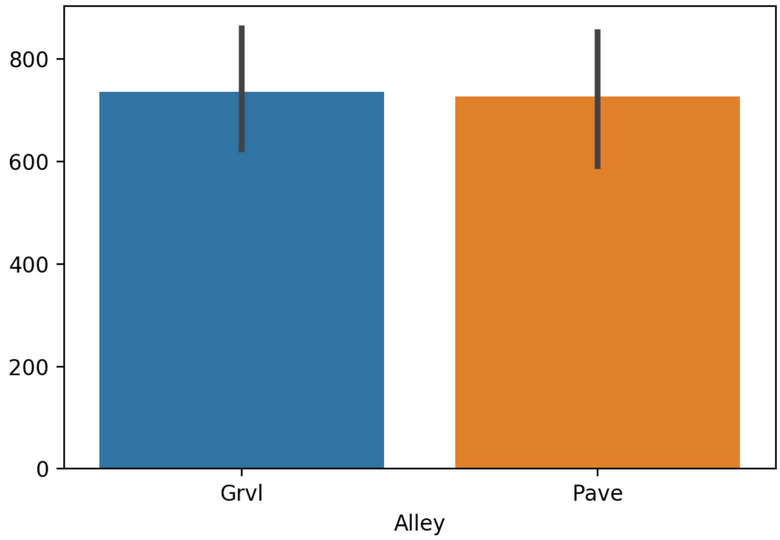


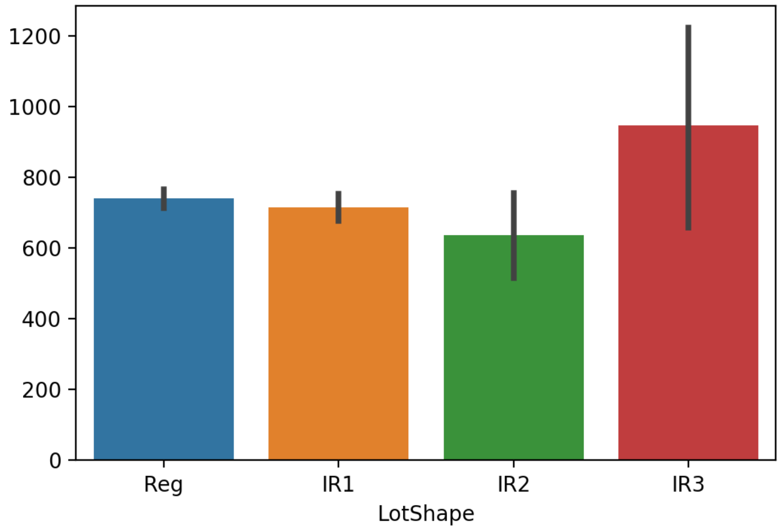


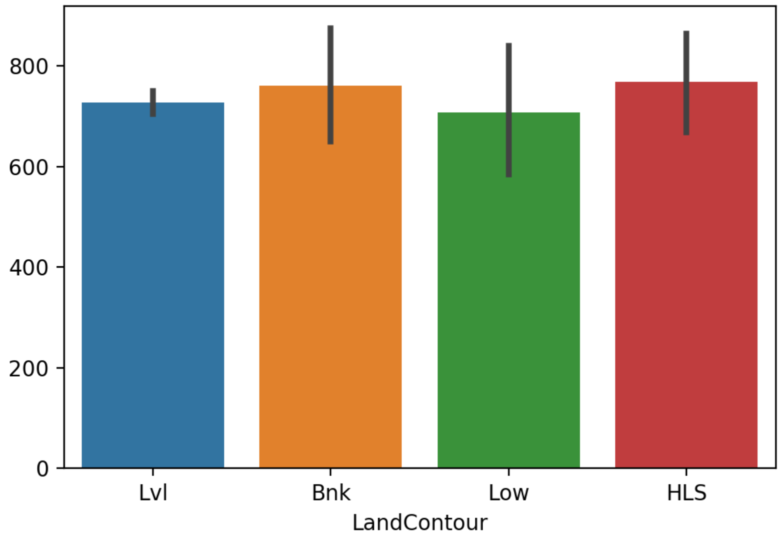


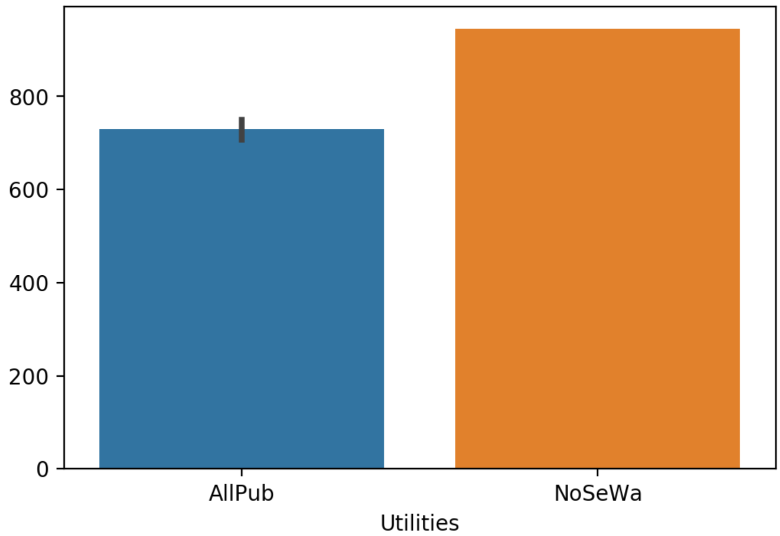

200000
150000
100000
50000
0
LotArea

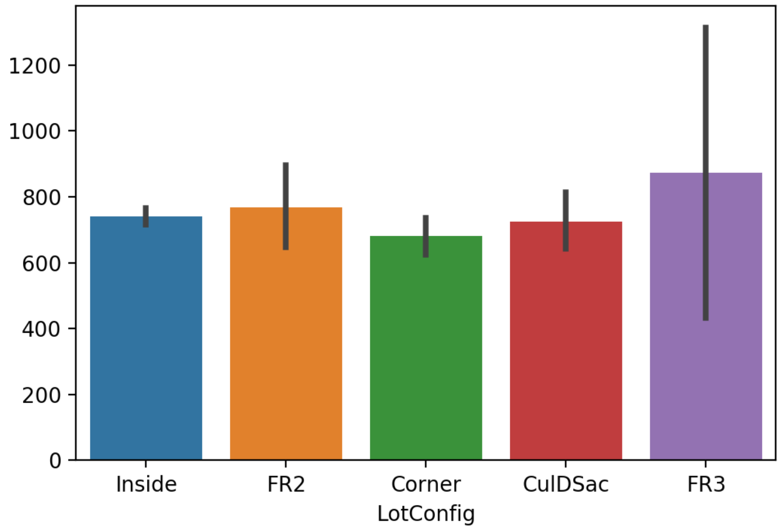

1000
800
600
400
200
0
Pave
Grvl
Street


800
600
400
200
0
Grvl
Pave
Alley

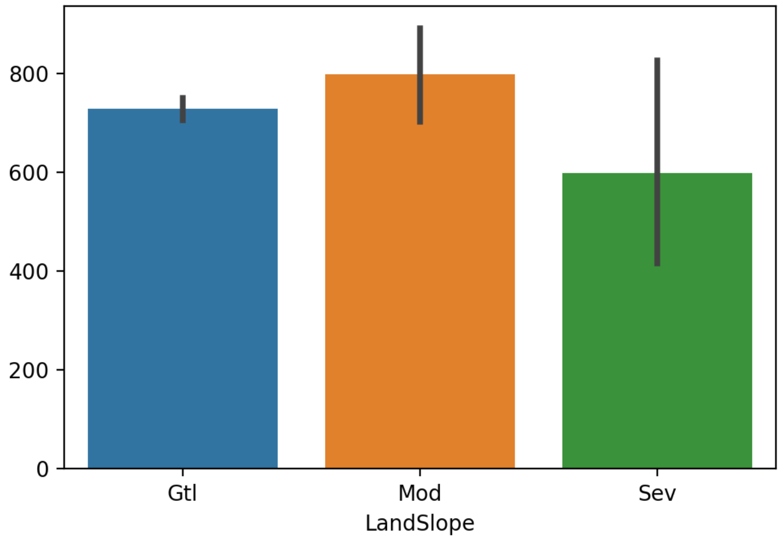

1200
1000
800
600
400
200
0
Reg
IR1
IR2
IR3
LotShape


800
600
400
200
0
Lvl
Bnk
Low
HLS
LandContour

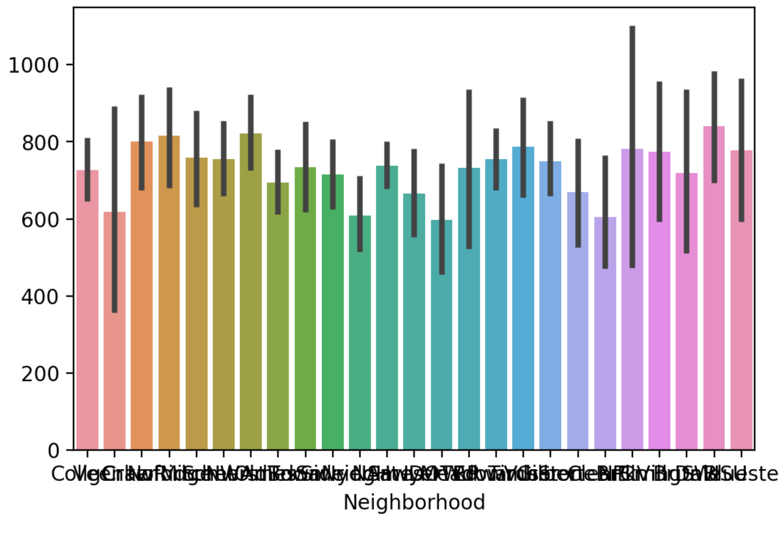

800
600
400
200
0
AllPub
NoSeWa
Utilities



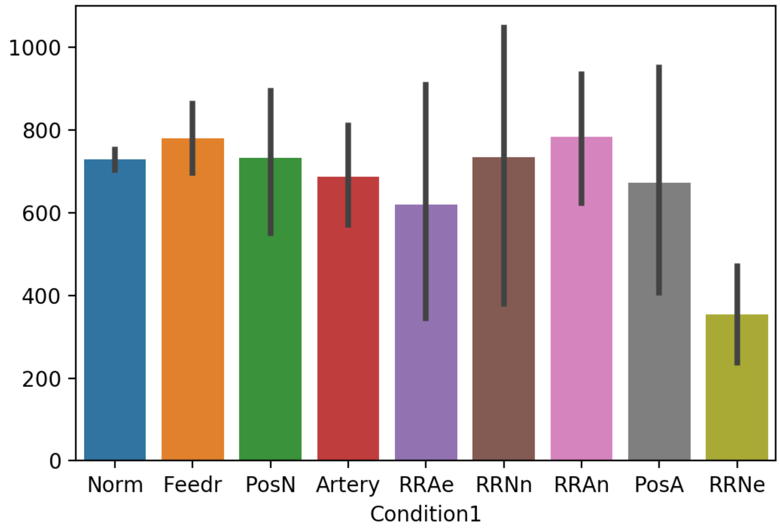
1200
1000
800
600
400
200
0
Inside
FR2
Corner
CulDSac
FR3
LotConfig

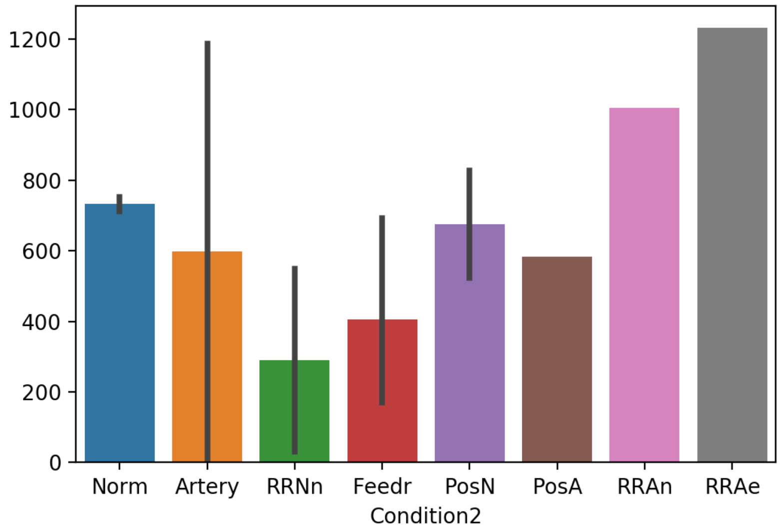


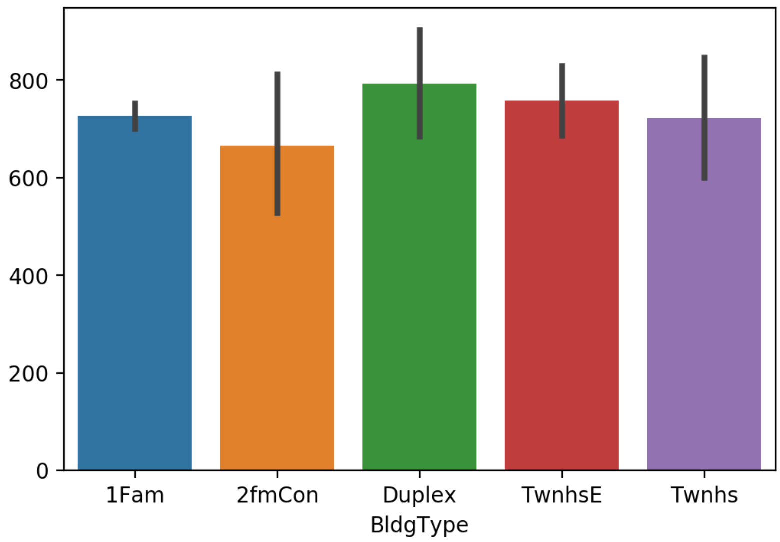
800
600
400
200
0
Gtl
Mod
Sev
LandSlope

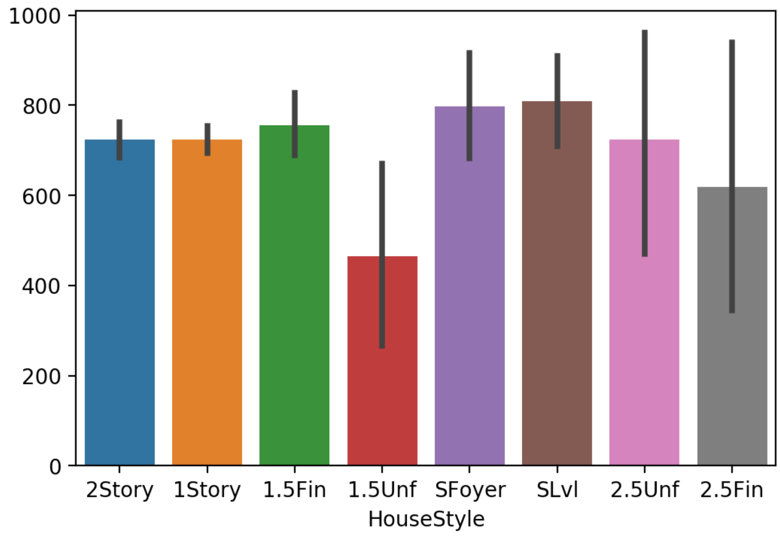


1000
800
600
400
200
0
Neighborhood

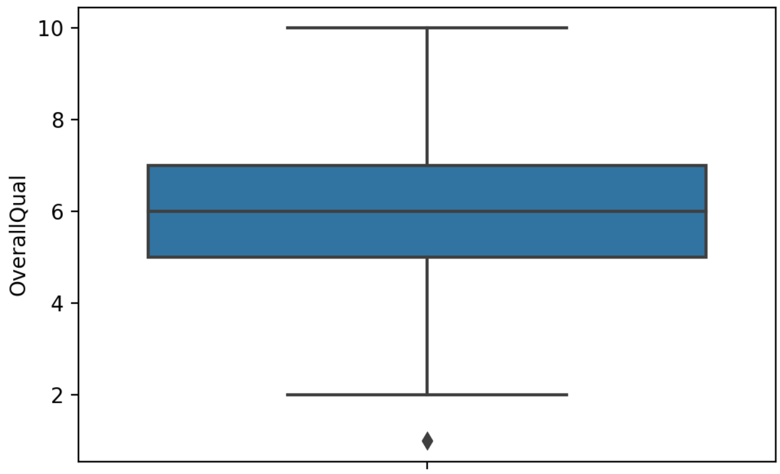

1000
800
600
400
200
0
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1


1200
1000
800
600
400
200
0
Norm
Artery
RRNn
Feedr
PosN
PosA
RRAn
RRAe
Condition2

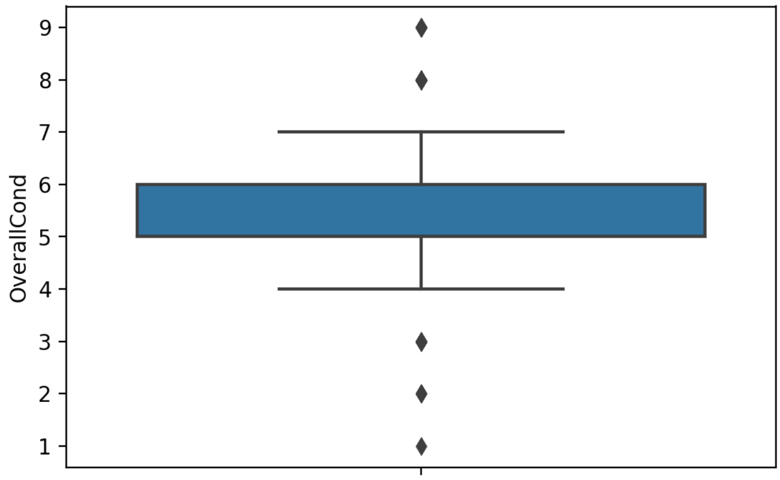

800
600
400
200
0
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
BldgType



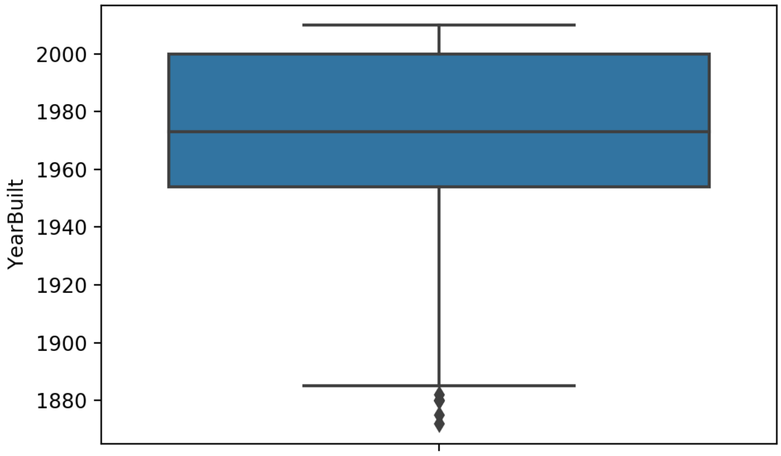
1000
800
600
400
200
0
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
HouseStyle



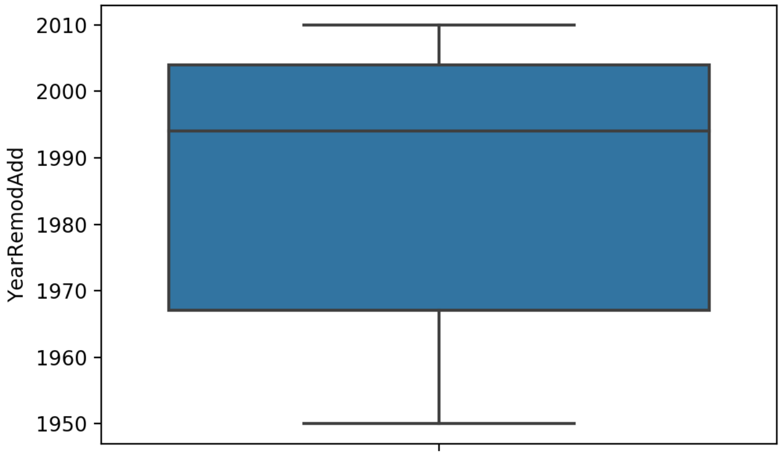
OverallQual
10
8
6
4
2



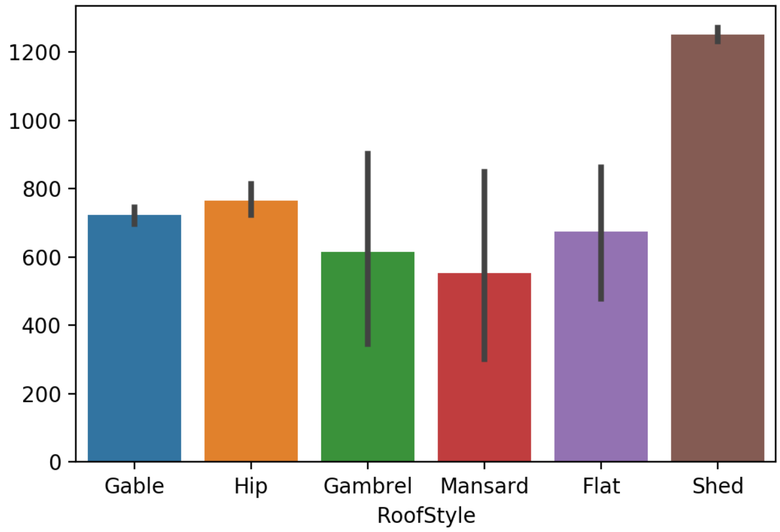
OverallCond
9
8
7
6
5
4
3
2
1



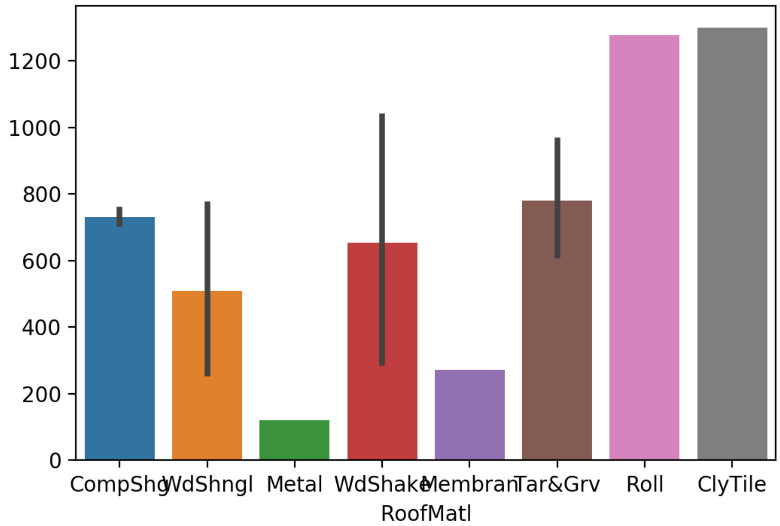
YearBuilt
2000
1980
1960
1940
1920
1900
1880



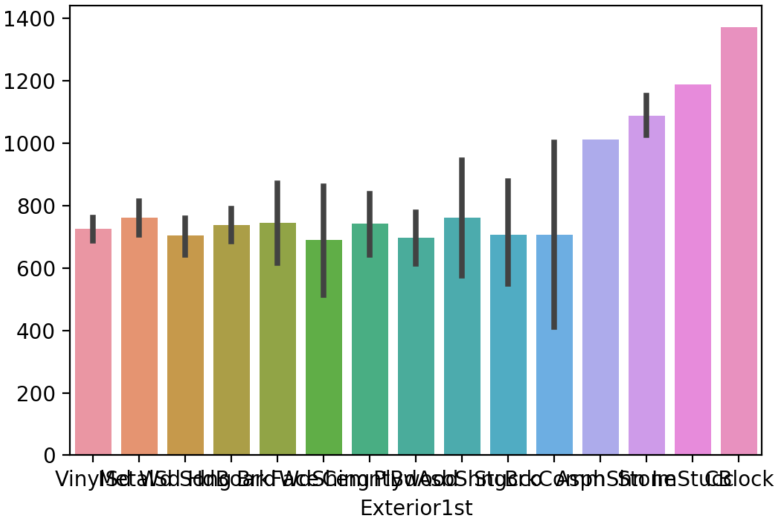
YearRemodAdd
2010
2000
1990
1980
1970
1960
1950

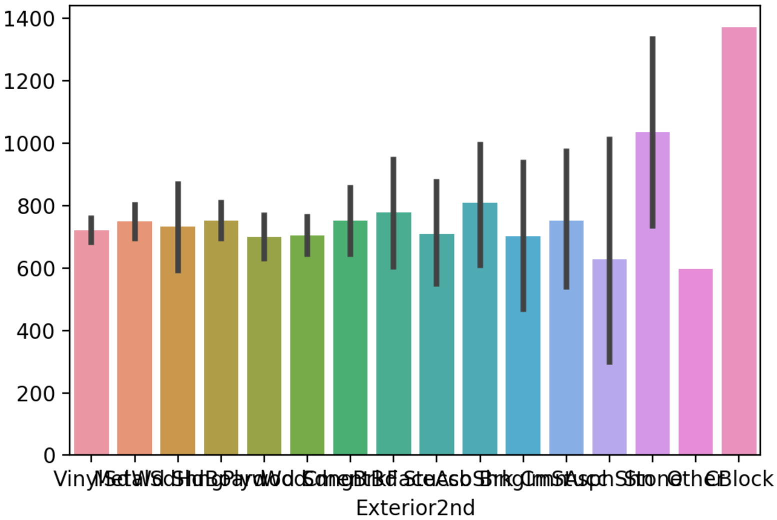


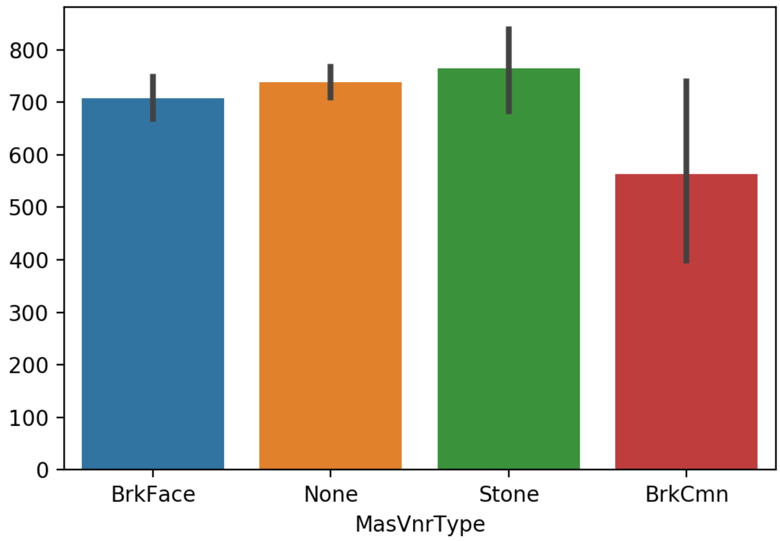
1200
1000
800
600
400
200
0
Gable
Hip
Gambrel
Mansard
Flat
Shed
RoofStyle

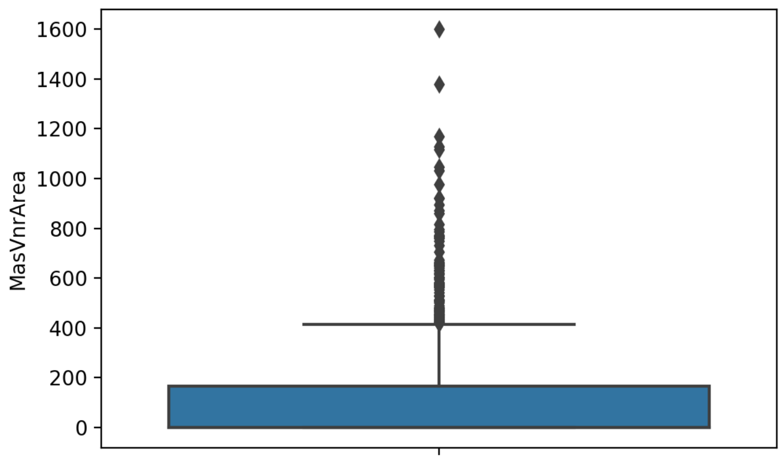


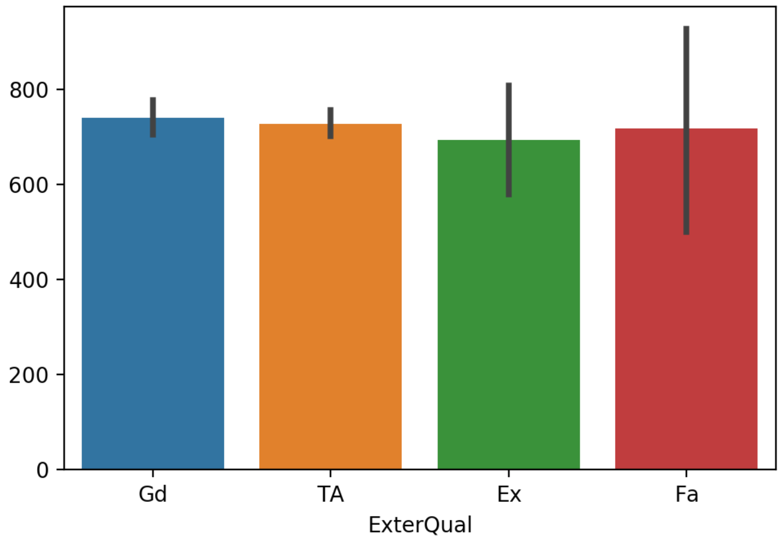
1200
1000
800
600
400
200
0
CompShg
WdShngl
Metal
WdShake
Membran
Tar&Grv
Roll
ClyTile
RoofMatl

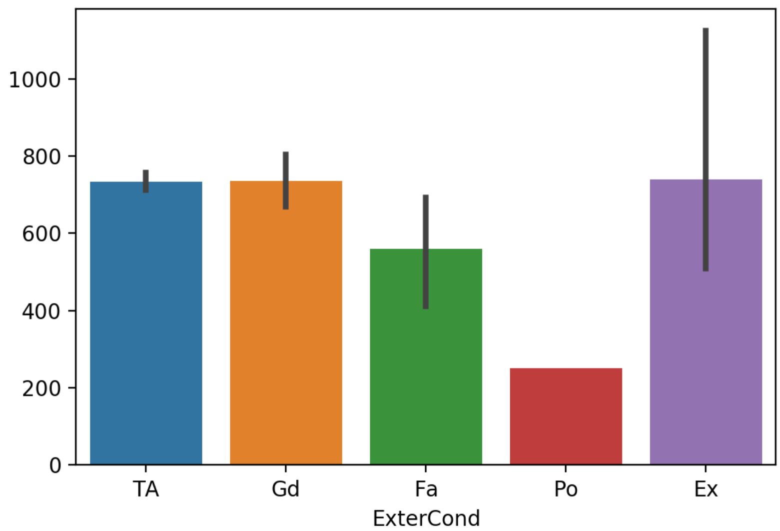

0
200
400
600
800
1000
1200
1400
Exterior1st

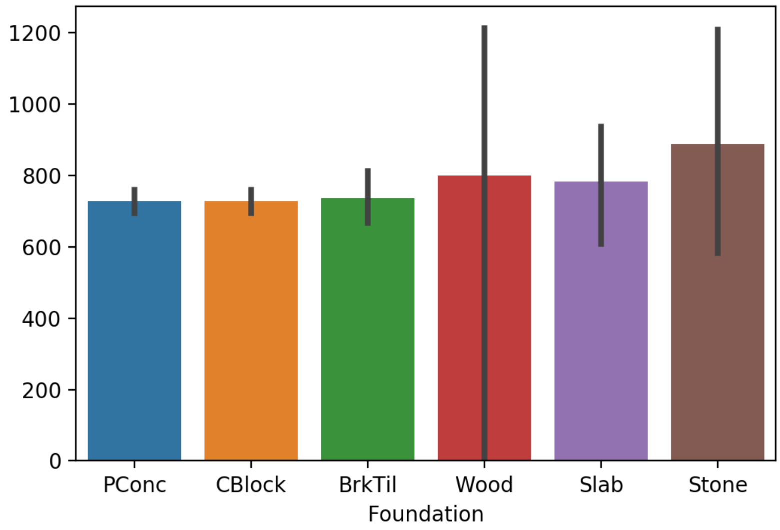

0
200
400
600
800
1000
1200
1400
Exterior2nd


0
100
200
300
400
500
600
700
800
BrkFace
None
Stone
BrkCmn
MasVnrType

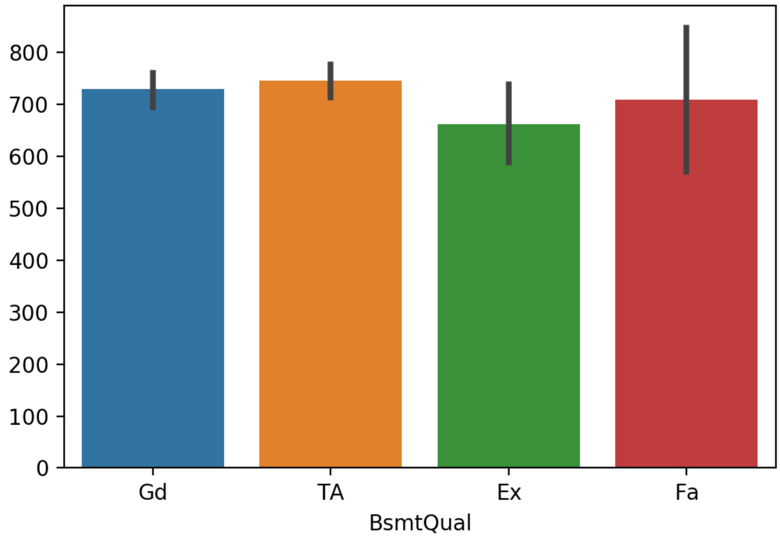

1600
1400
1200
1000
800
600
400
200
0
MasVnrArea


800
600
400
200
0
Gd
TA
Ex
Fa
ExterQual

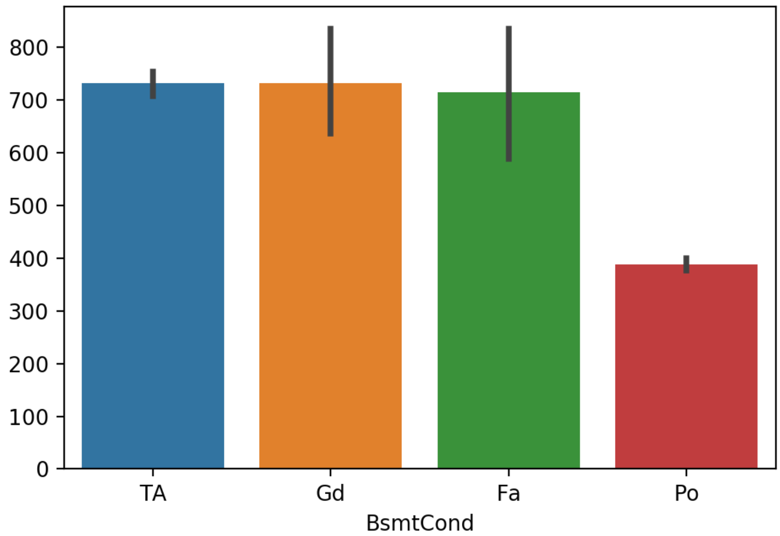

1000
800
600
400
200
0
TA
Gd
Fa
Po
Ex
ExterCond



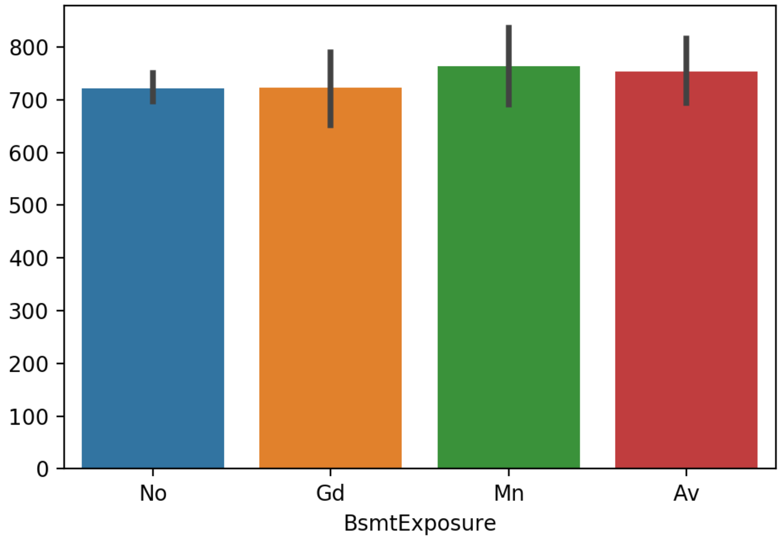
1200
1000
800
600
400
200
0
PConc
CBlock
BrkTil
Wood
Slab
Stone
Foundation



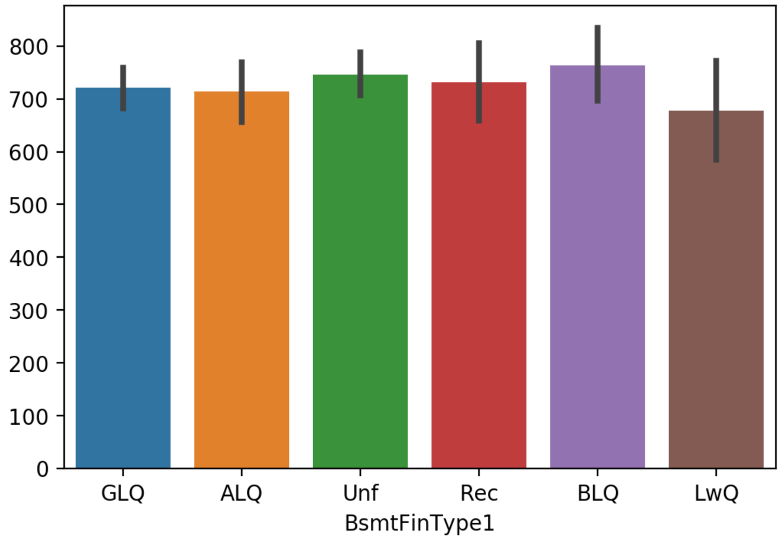
800
700
600
500
400
300
200
100
0
Gd
TA
Ex
Fa
BsmtQual



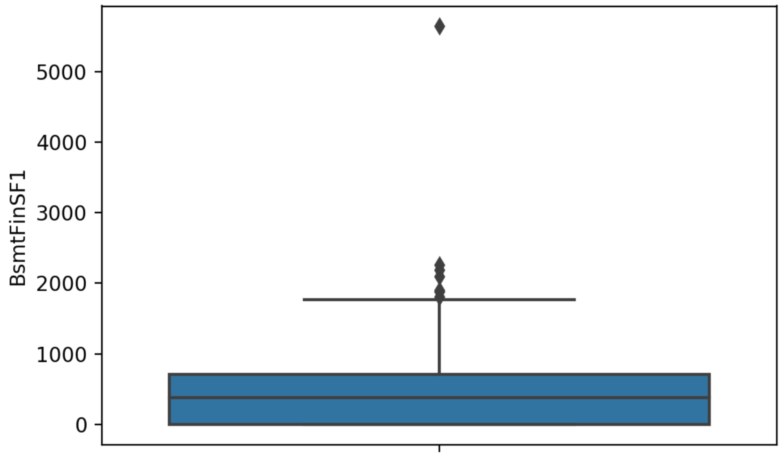
800
700
600
500
400
300
200
100
0
TA
Gd
Fa
Po
BsmtCond



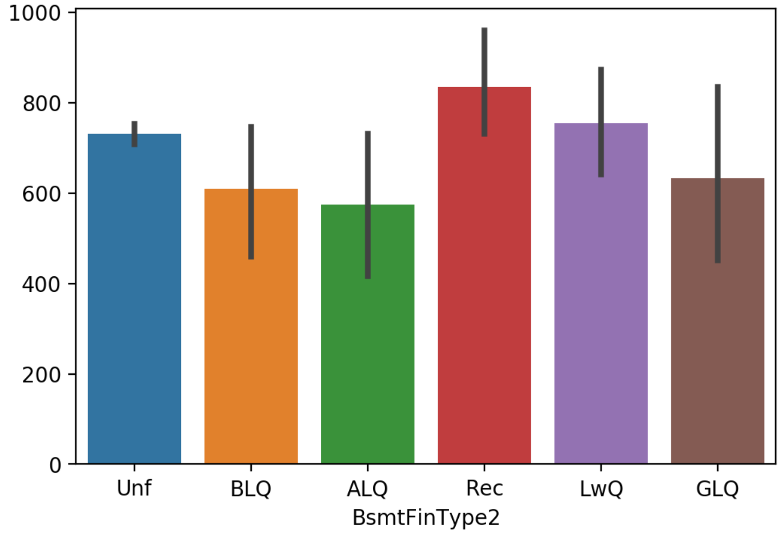
800
700
600
500
400
300
200
100
0
No
Gd
Mn
Av
BsmtExposure

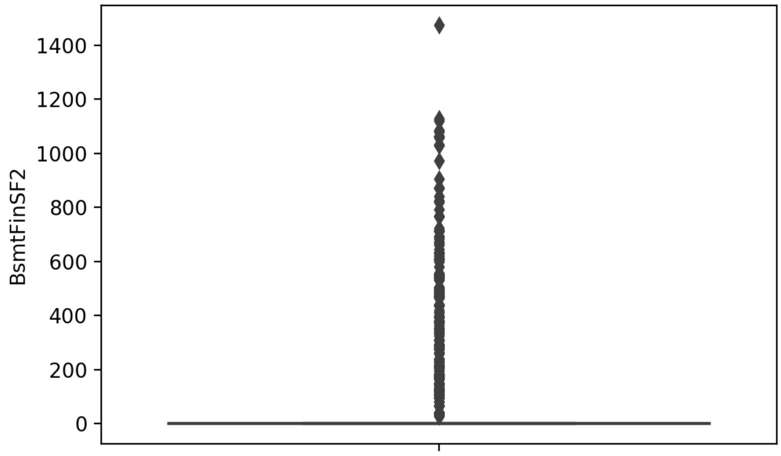


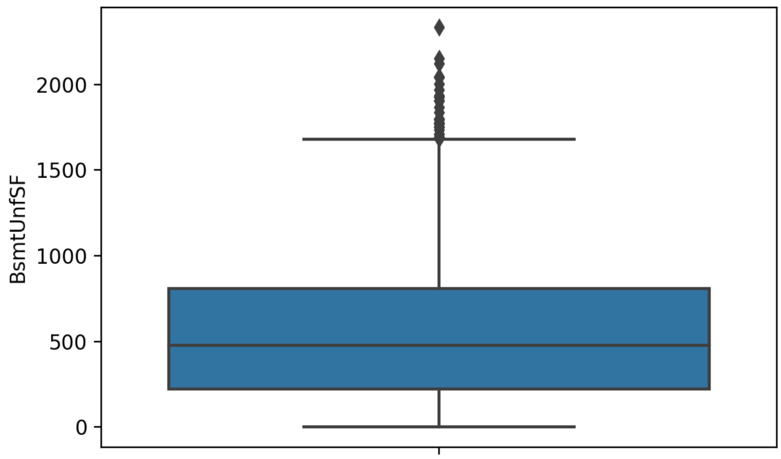
800
700
600
500
400
300
200
100
0
GLQ
ALQ
Unf
Rec
BLQ
LwQ
BsmtFinType1

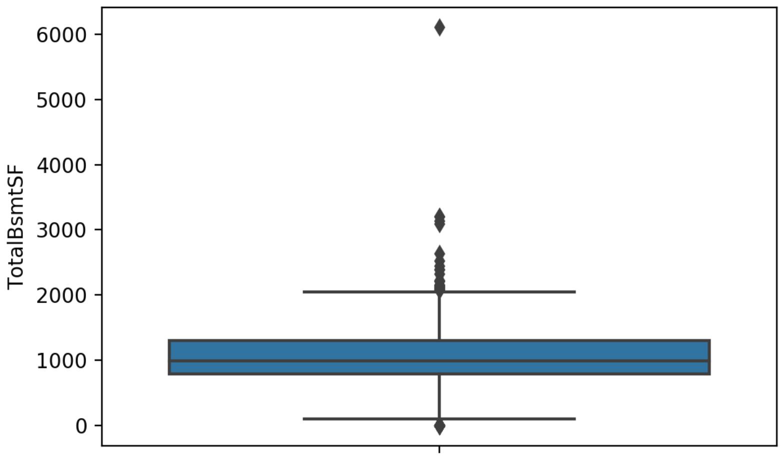


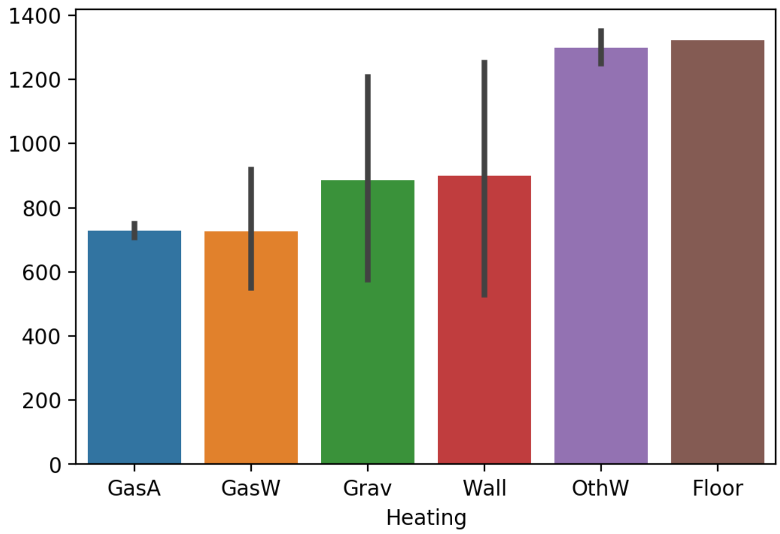
BsmtFinSF1
5000
4000
3000
2000
1000
0

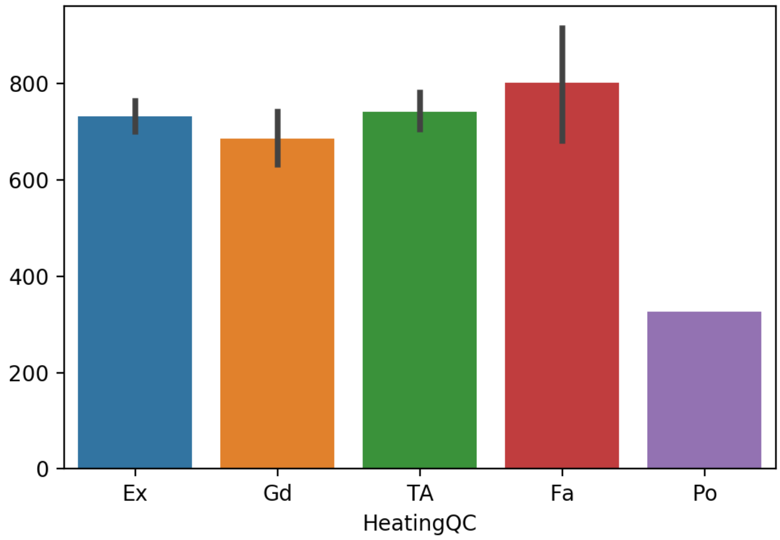

1000
800
600
400
200
0
Unf
BLQ
ALQ
Rec
LwQ
GLQ
BsmtFinType2

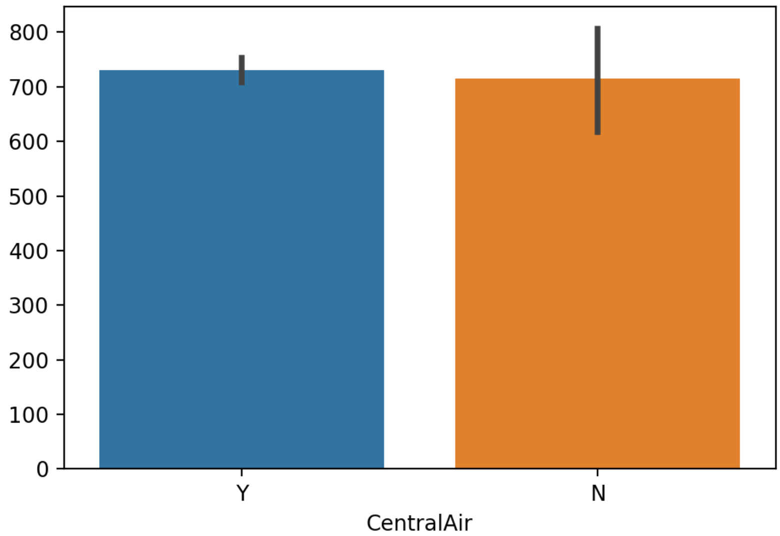

BsmtFinSF2
1400
1200
1000
800
600
400
200
0

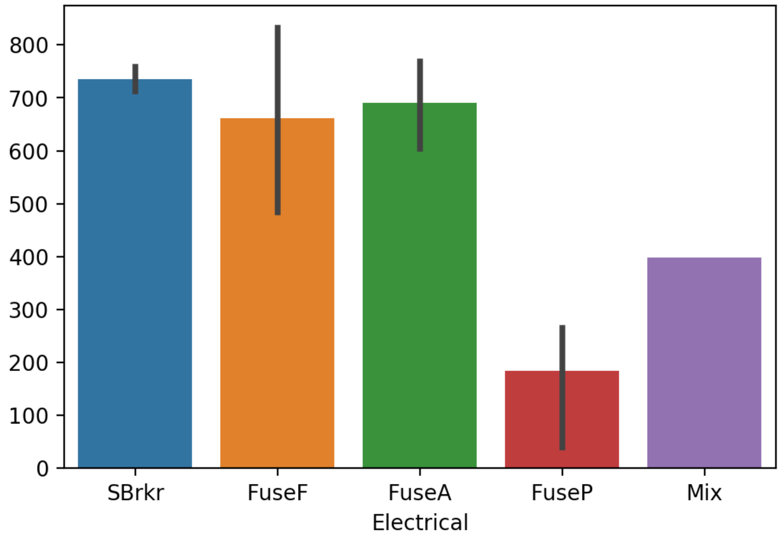

BsmtUnfSF
2000
1500
1000
500
0

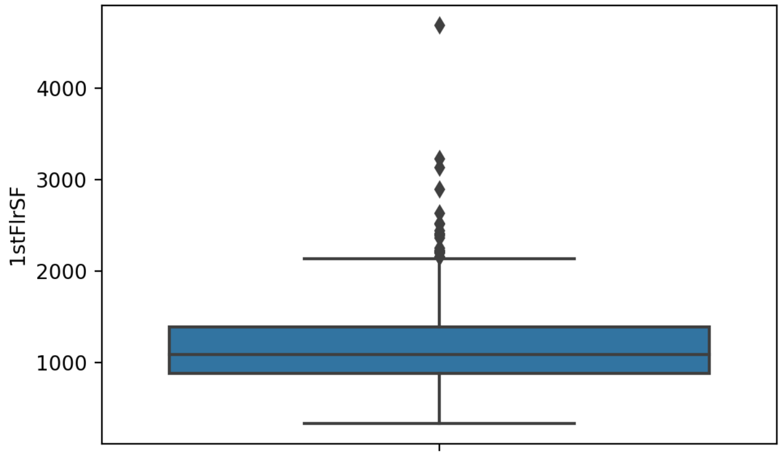

TotalBsmtSF
6000
5000
4000
3000
2000
1000
0

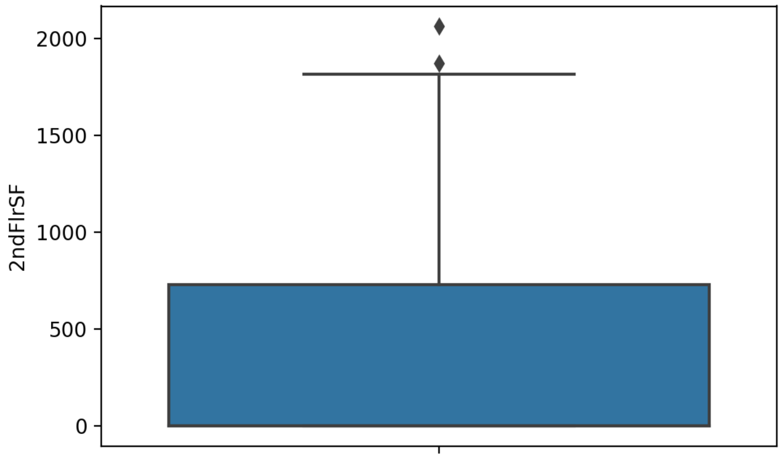

1400
1200
1000
800
600
400
200
0
GasA
GasW
Grav
Wall
OthW
Floor
Heating

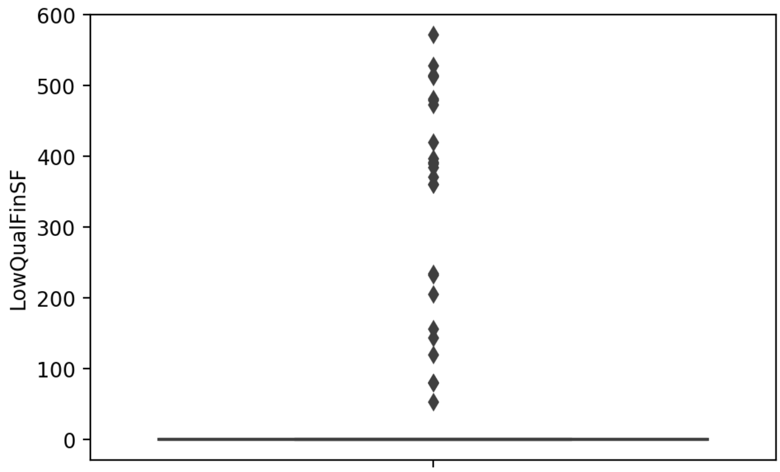

800
600
400
200
0
Ex
Gd
TA
Fa
Po
HeatingQC

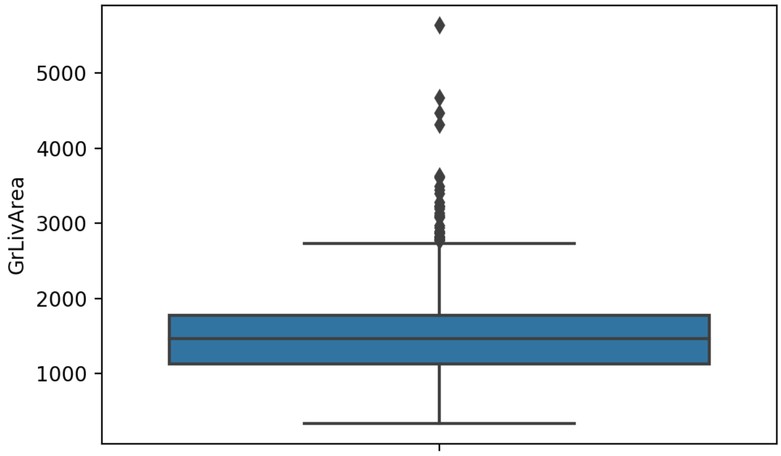

800
700
600
500
400
300
200
100
0
Y
N
CentralAir

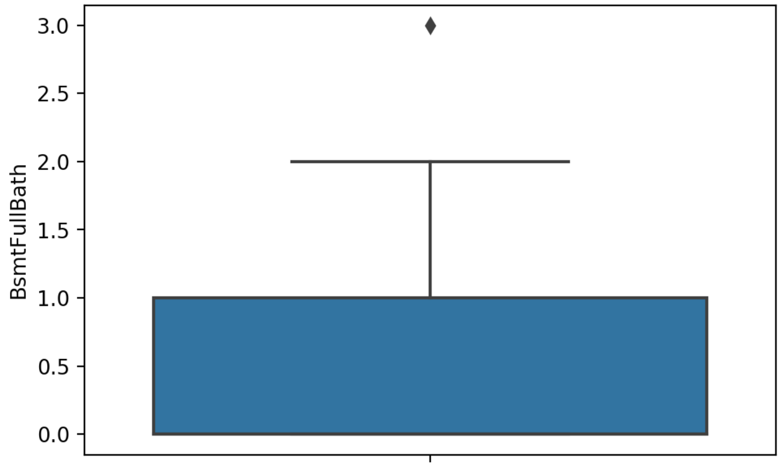

800
700
600
500
400
300
200
100
0
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical

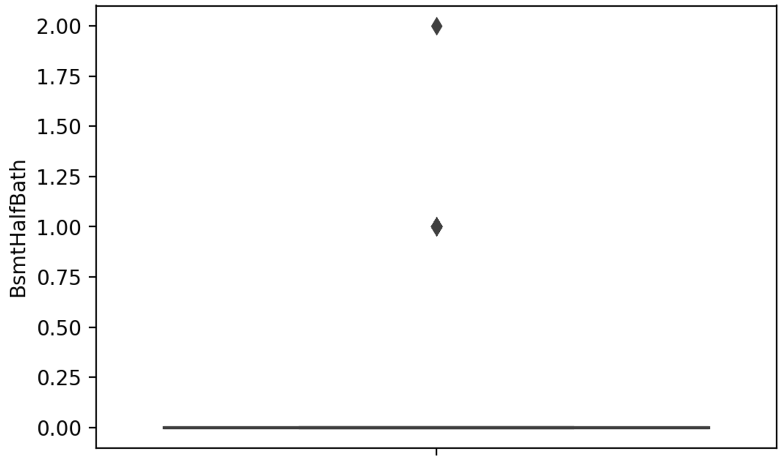

1stFlrSF
4000
3000
2000
1000

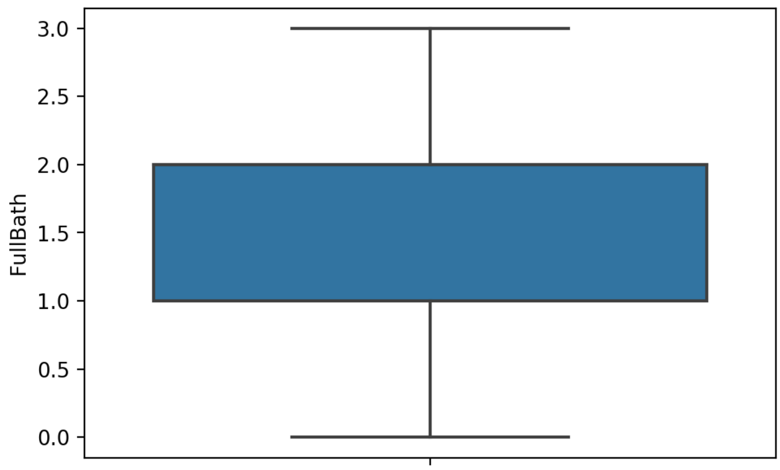

2ndFlrSF
2000
1500
1000
500
0

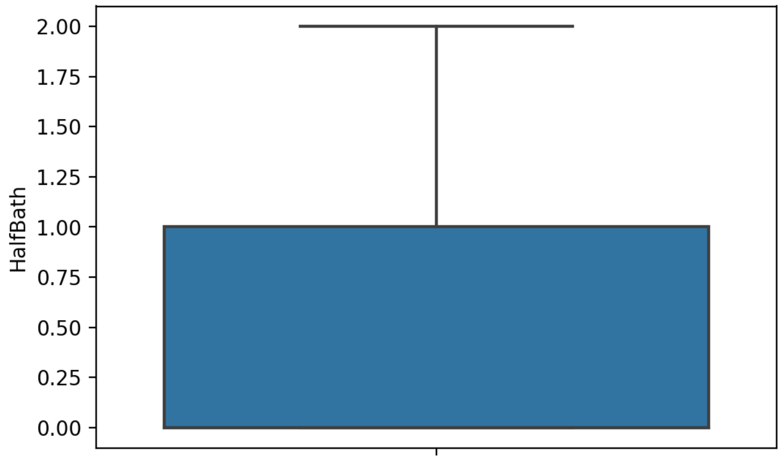

LowQualFinSF
600
500
400
300
200
100
0

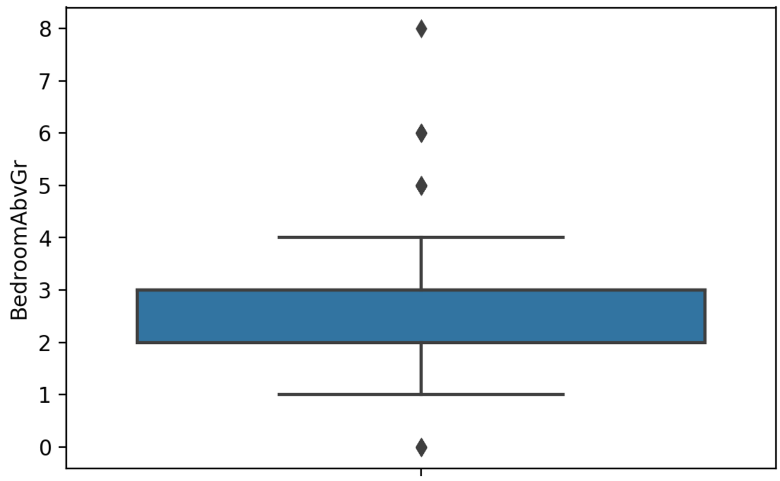

5000
4000
3000
2000
1000
GrLivArea

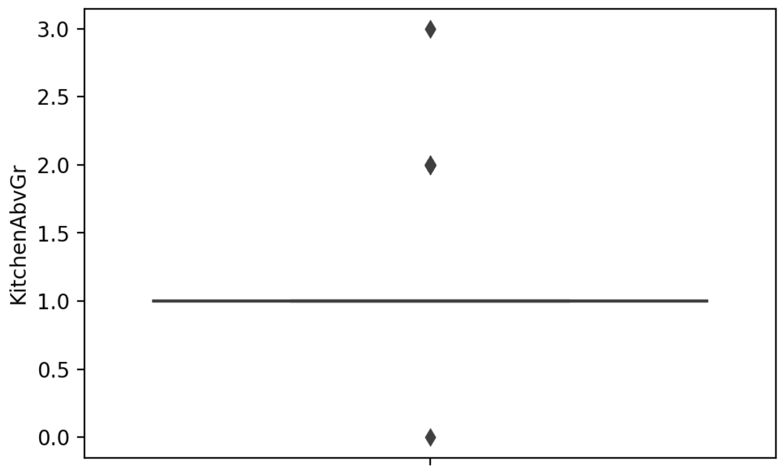

3.0
2.5
2.0
1.5
1.0
0.5
0.0
BsmtFullBath

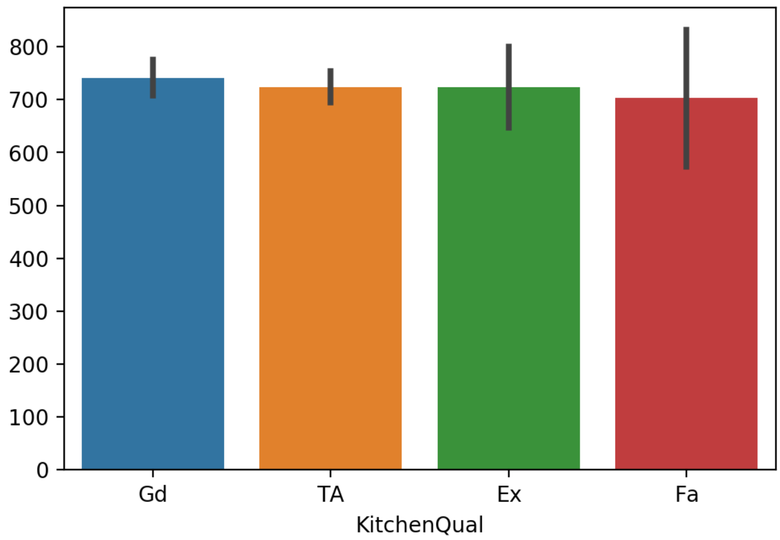

2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
BsmtHalfBath

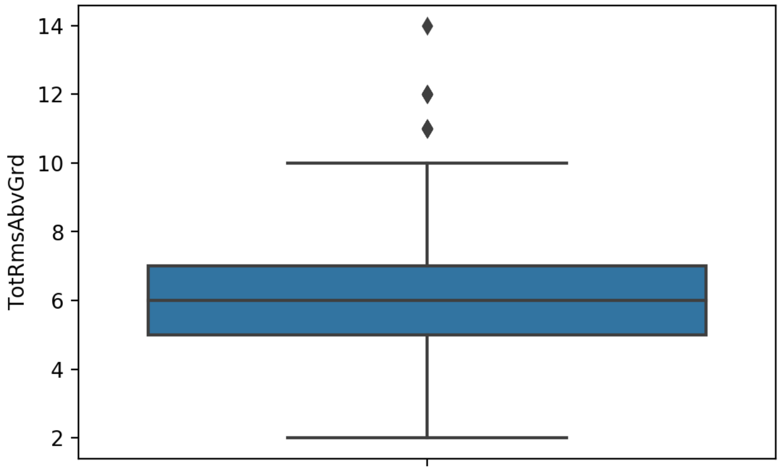

3.0
2.5
2.0
1.5
1.0
0.5
0.0
FullBath


HalfBath
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

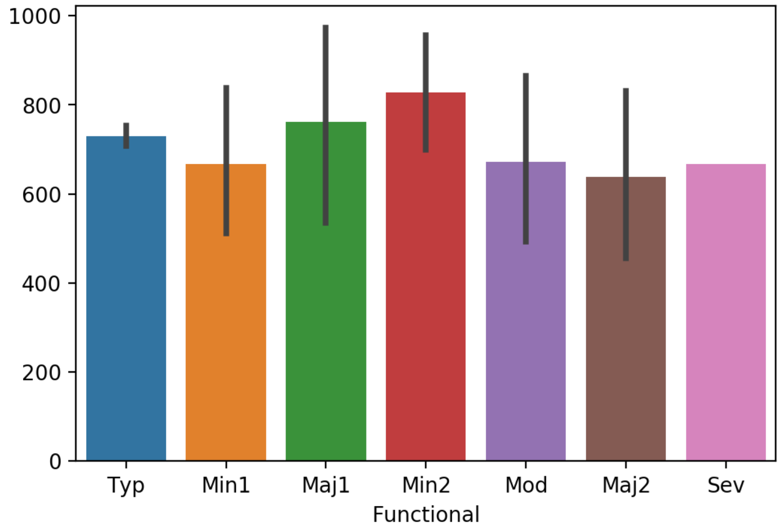

BedroomAbvGr
8
7
6
5
4
3
2
1
0


KitchenAbvGr
3.0
2.5
2.0
1.5
1.0
0.5
0.0

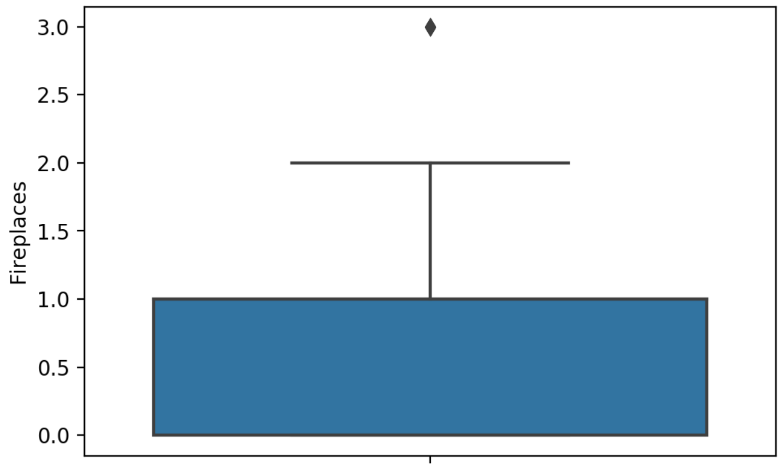

800
700
600
500
400
300
200
100
0
Gd
TA
Ex
Fa
KitchenQual



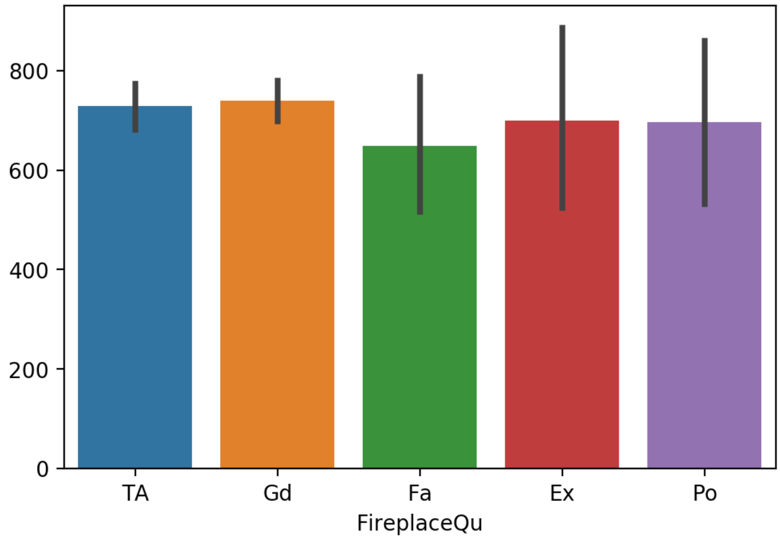
TotRmsAbvGrd
14
12
10
8
6
4
2



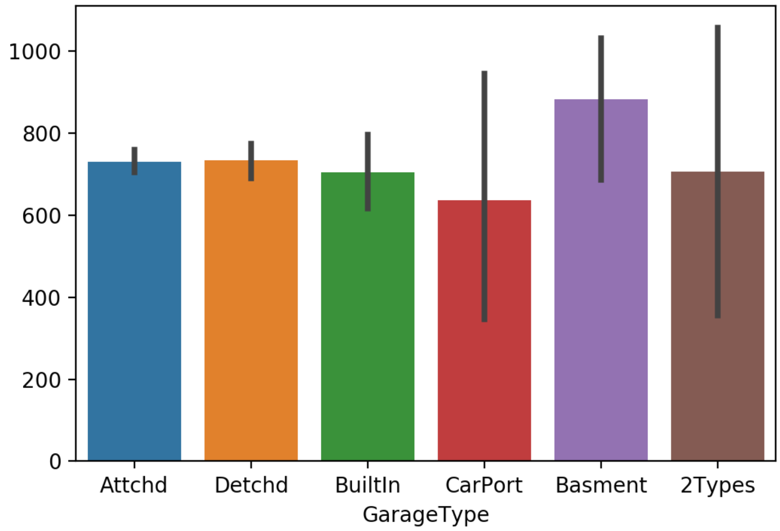
1000
800
600
400
200
0
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functional



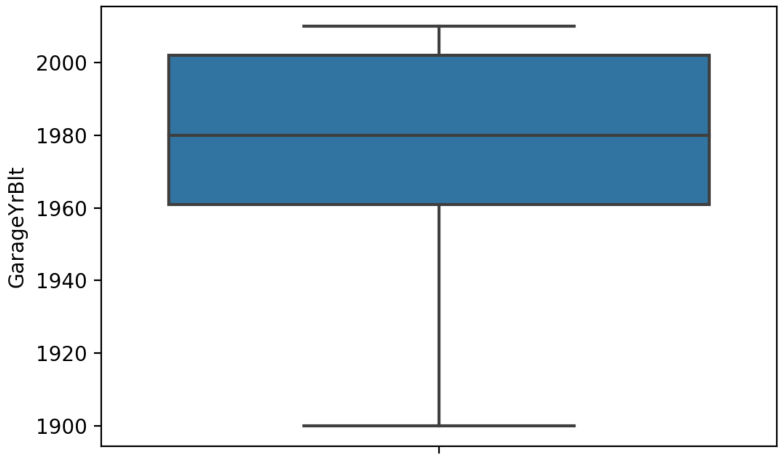
Fireplaces
3.0
2.5
2.0
1.5
1.0
0.5
0.0



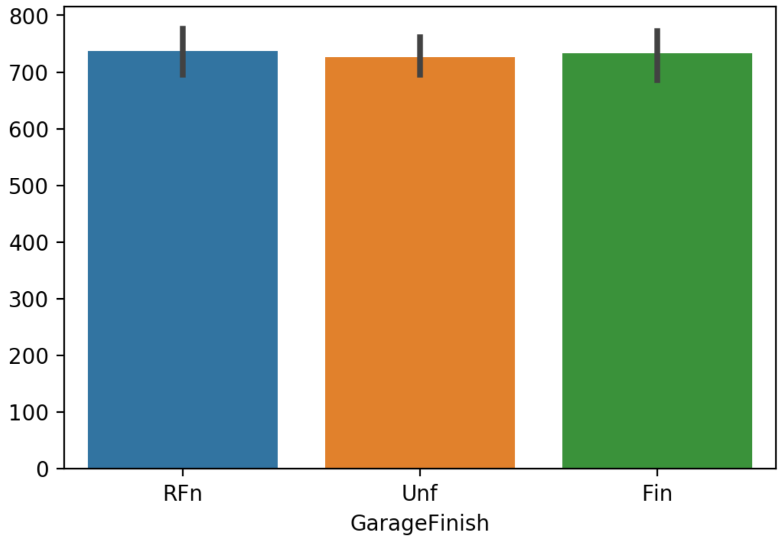
800
600
400
200
0
TA
Gd
Fa
Ex
Po
FireplaceQu

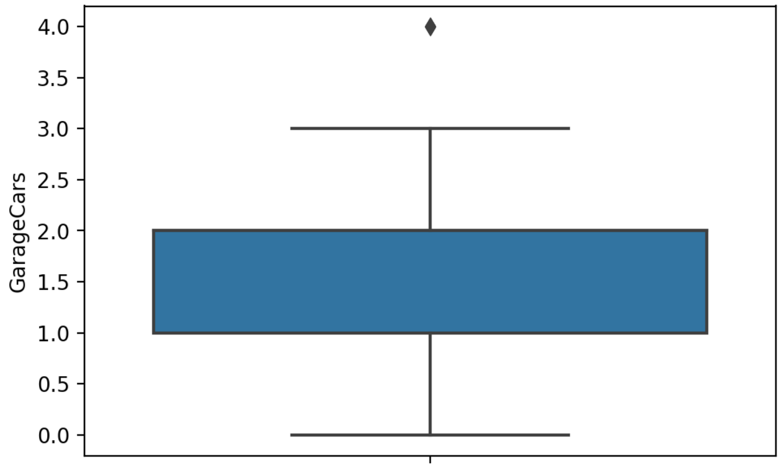


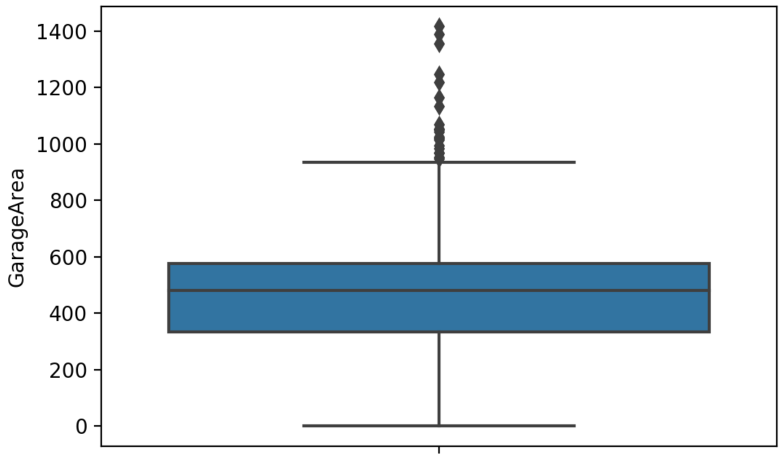
1000
800
600
400
200
0
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageType

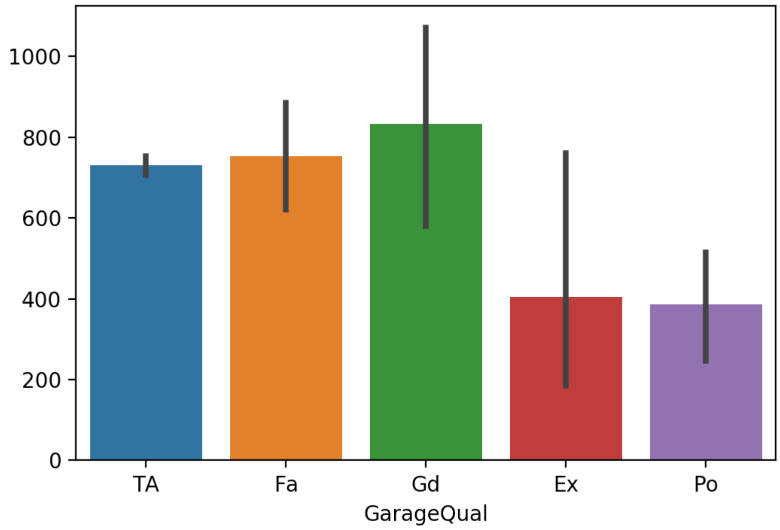


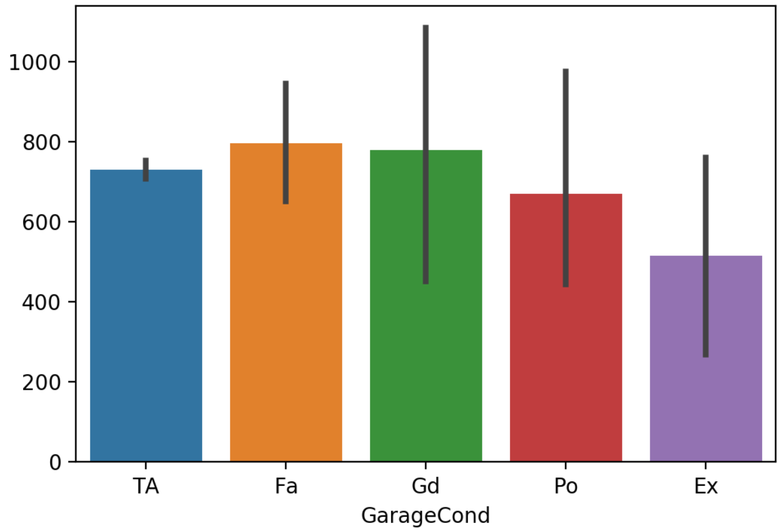
GarageYrBlt
2000
1980
1960
1940
1920
1900

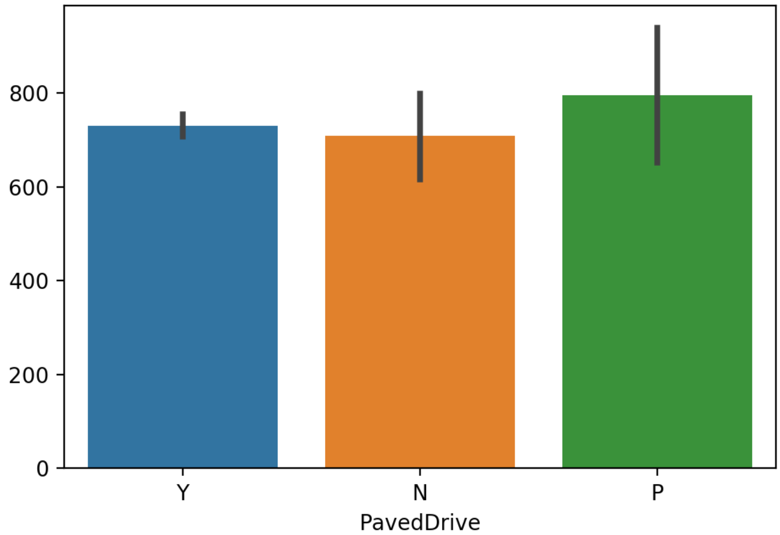

800
700
600
500
400
300
200
100
0
RFn
Unf
Fin
GarageFinish

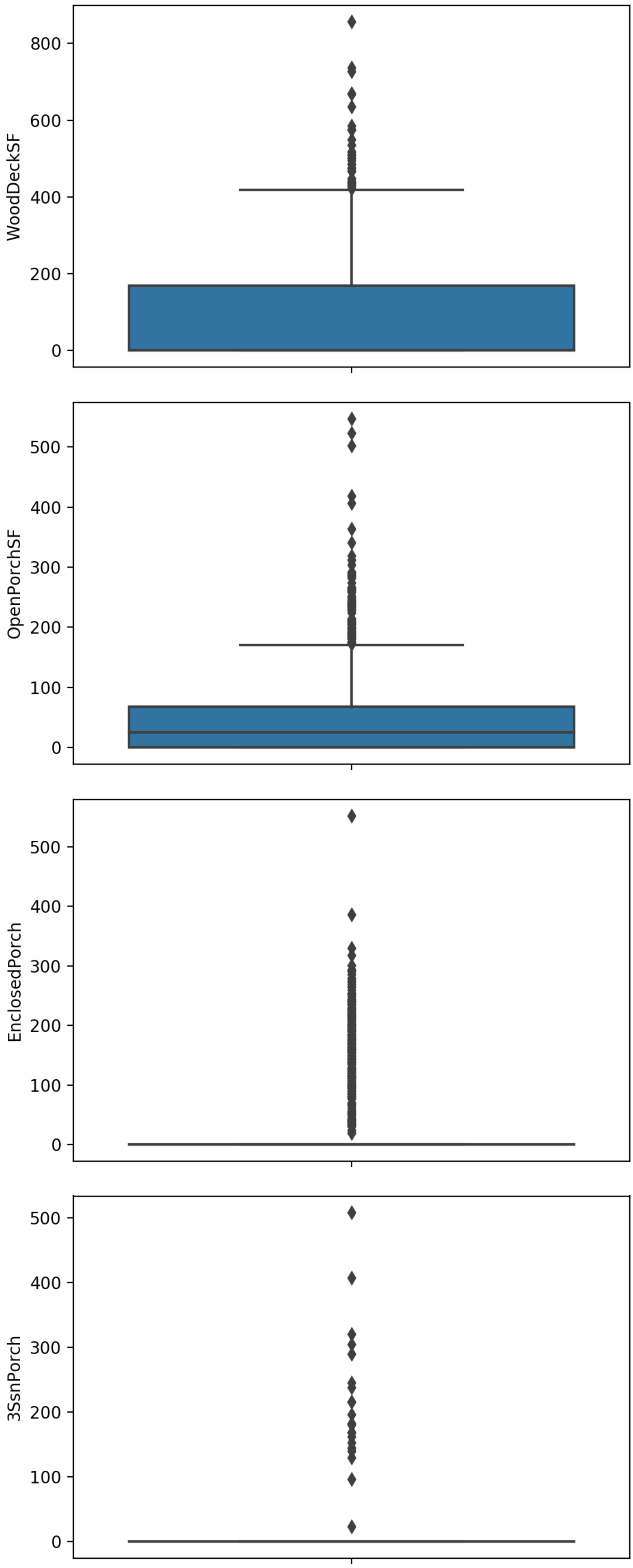

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
GarageCars

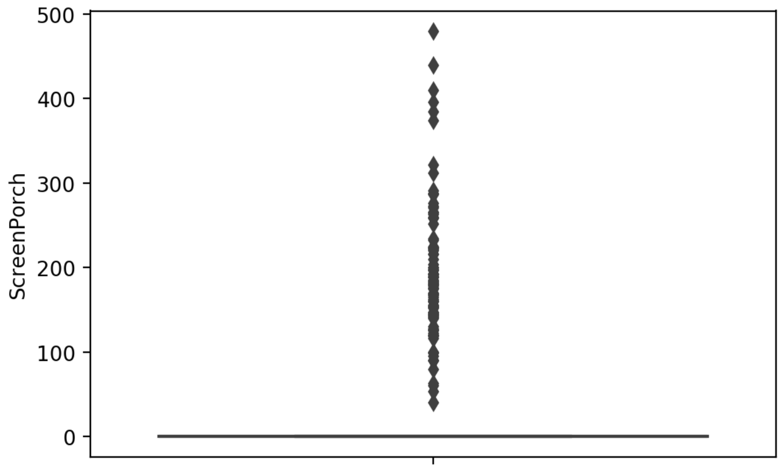

1400
1200
1000
800
600
400
200
0
GarageArea

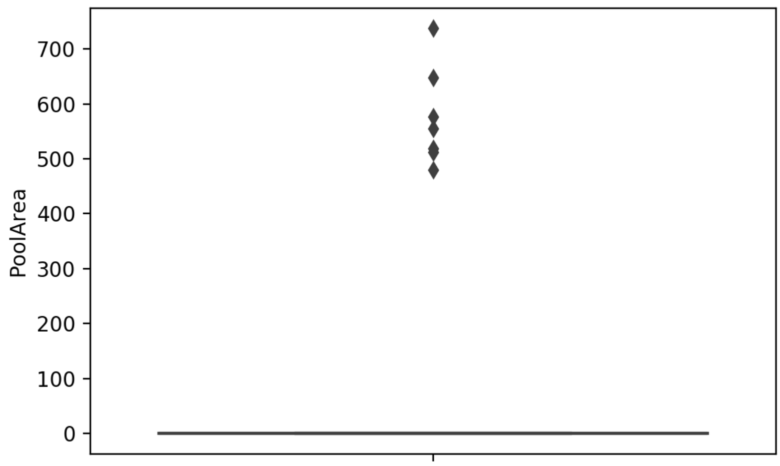

1000
800
600
400
200
0
TA
Fa
Gd
Ex
Po
GarageQual

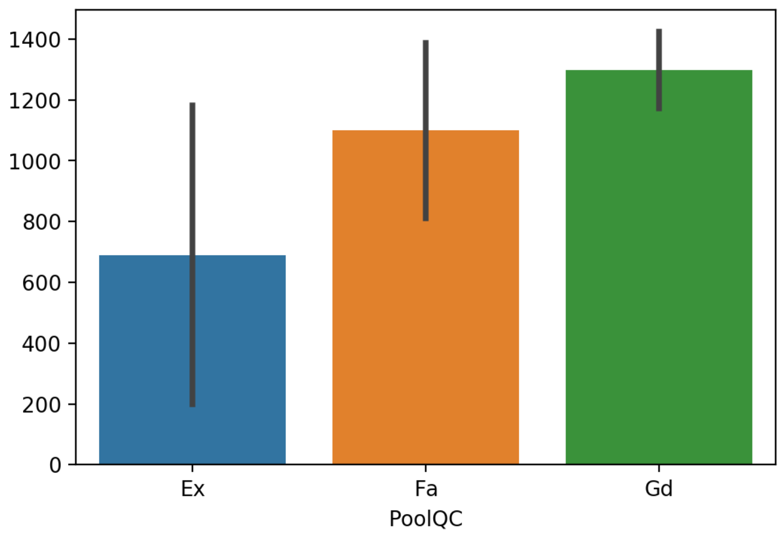

1000
800
600
400
200
0
TA
Fa
Gd
Po
Ex
GarageCond

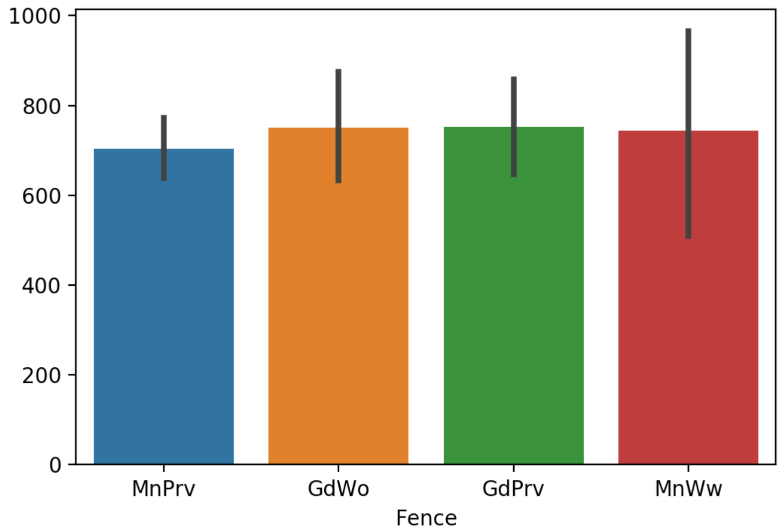

800
600
400
200
0
Y
N
P
PavedDrive

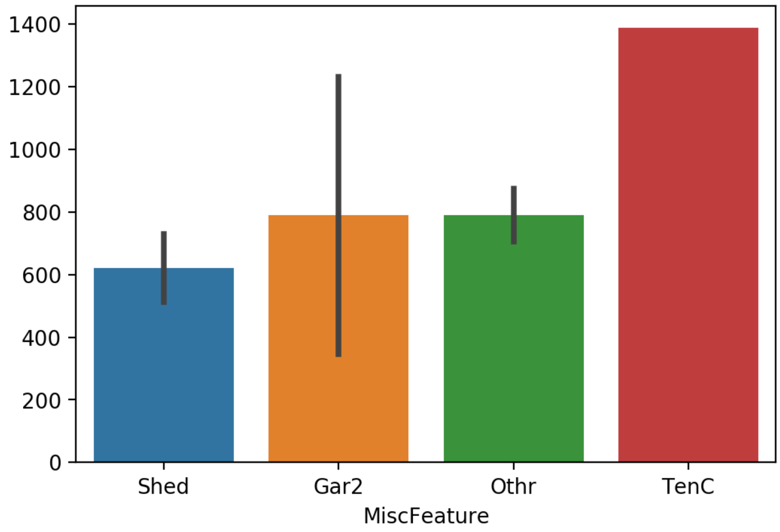

WoodDeckSF
0
200
400
600
800
OpenPorchSF
0
100
200
300
400
500
EnclosedPorch
0
100
200
300
400
500
3SsnPorch
0
100
200
300
400
500

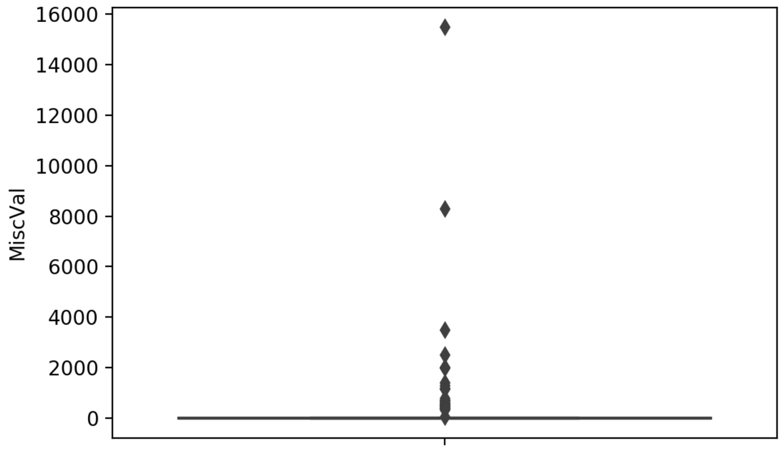

ScreenPorch
0
100
200
300
400
500

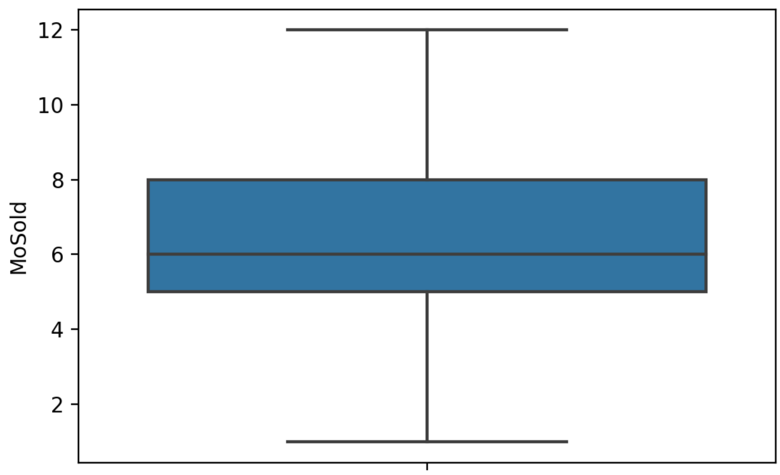

PoolArea
0
100
200
300
400
500
600
700

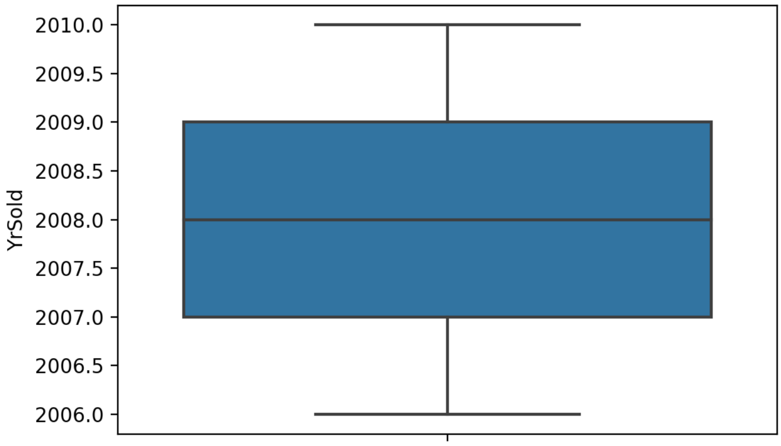

0
200
400
600
800
1000
1200
1400
Ex
Fa
Gd
PoolQC


1000
800
600
400
200
0
MnPrv
GdWo
GdPrv
MnWw
Fence

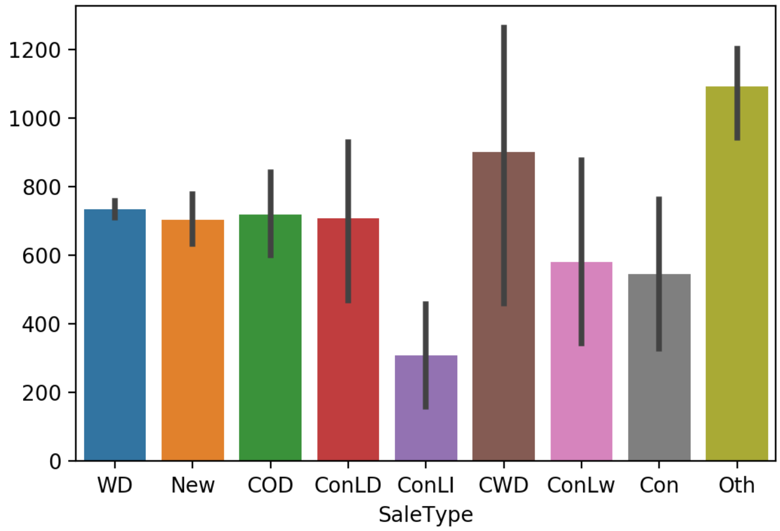

1400
1200
1000
800
600
400
200
0
Shed
Gar2
Othr
TenC
MiscFeature


16000
14000
12000
10000
8000
6000
4000
2000
0
MiscVal

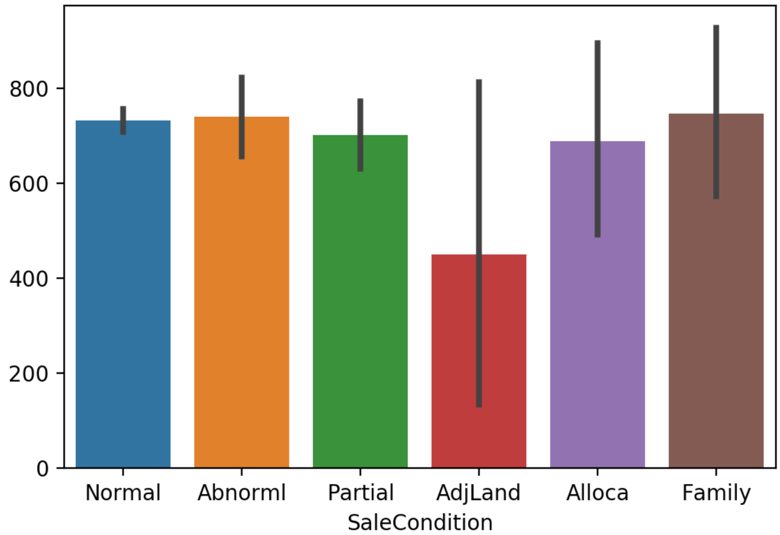

12
10
8
6
4
2
MoSold



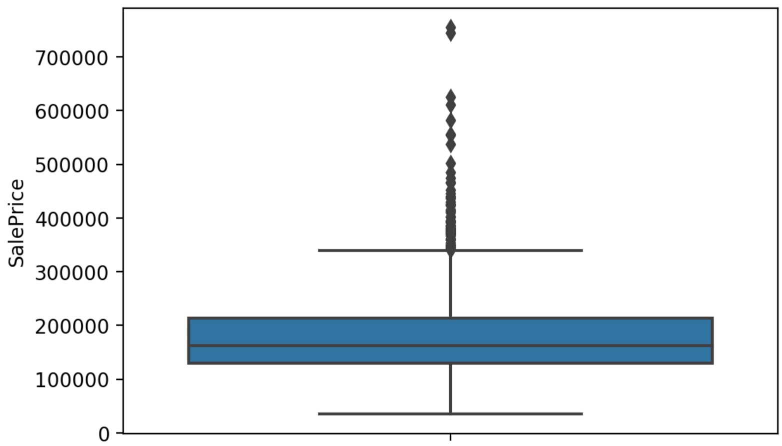
YrSold
2010.0
2009.5
2009.0
2008.5
2008.0
2007.5
2007.0
2006.5
2006.0



1200
1000
800
600
400
200
0
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleType

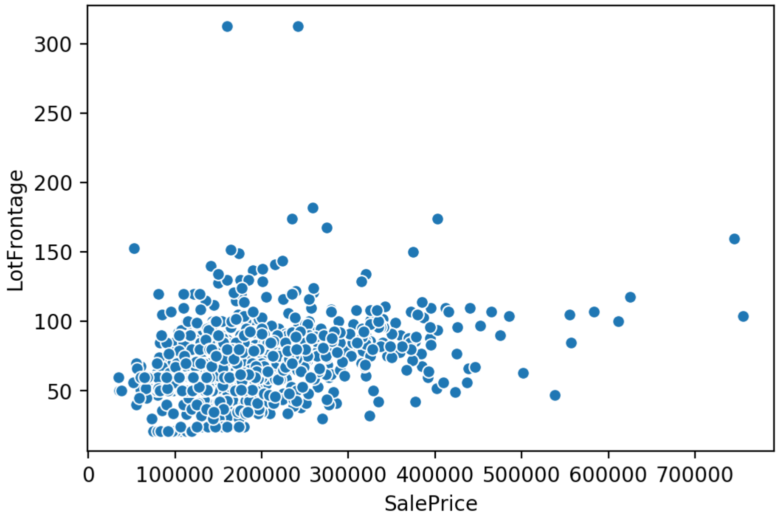


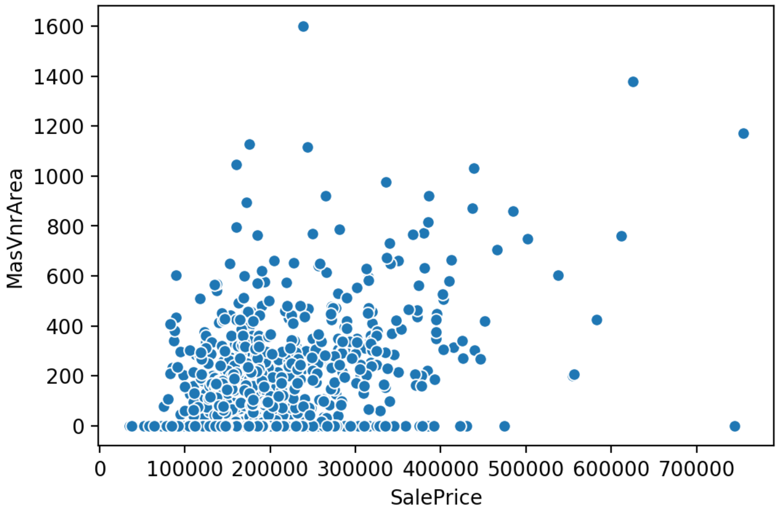
800
600
400
200
0
Normal
Abnorml
Partial
AdjLand
Alloca
Family
SaleCondition

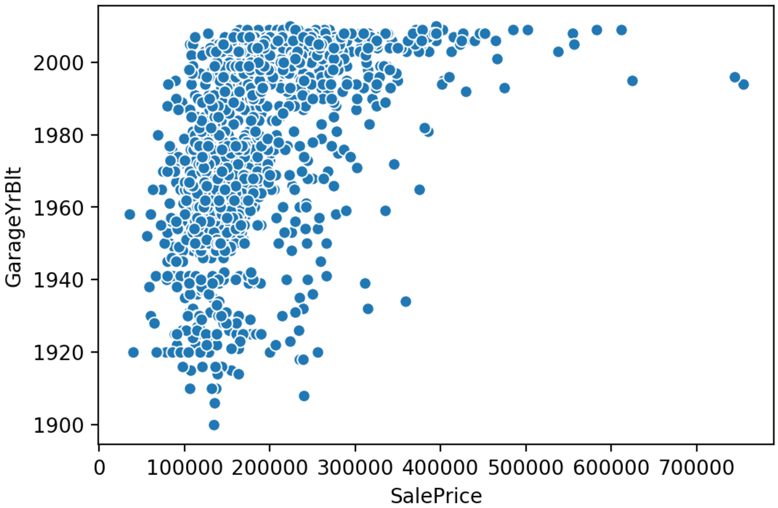


SalePrice
700000
600000
500000
400000
300000
200000
100000
0

```
for i, x in enumerate(df.dtypes):
    if x == 'float64':
        plt.figure(i)
        seaborn.scatterplot(x = df.SalePrice ,y = df[df.columns[i]])
```





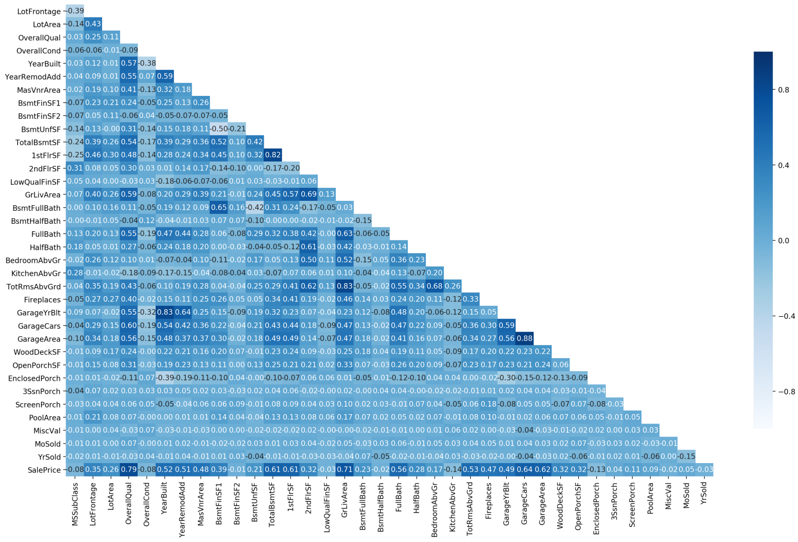


2 - Análise de correlações (rxi,xj)

2.1 Correlograma (1,5)

```
df_corr = df.corr()

import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(20, 12))
```

```
# mask
mask = np.triu(np.ones_like(df_corr, dtype=np.bool))
# adjust mask and df
mask = mask[1:, :-1]
corr = df_corr.iloc[1:,:-1].copy()
# plot heatmap
seaborn.heatmap(corr, mask=mask, annot=True, fmt=".2f", cmap='Blues',
           vmin=-1, vmax=1, cbar_kws={"shrink": .8})
# yticks
plt.yticks(rotation=0)
plt.show()
```



2.2 Análise sobre correlações significativas (1,5)

Correlações com a variável resposta

```
df.corr().loc['SalePrice']
```

```
MSSubClass      -0.084284
LotFrontage      0.351799
LotArea          0.263843
OverallQual      0.790982
OverallCond     -0.077856
YearBuilt        0.522897
YearRemodAdd     0.507101
MasVnrArea       0.477493
BsmtFinSF1       0.386420
BsmtFinSF2      -0.011378
BsmtUnfSF        0.214479
TotalBsmtSF      0.613581
1stFlrSF         0.605852
2ndFlrSF         0.319334
LowQualFinSF    -0.025606
GrLivArea        0.708624
BsmtFullBath     0.227122
BsmtHalfBath    -0.016844
FullBath         0.560664
HalfBath         0.284108
BedroomAbvGr     0.168213
KitchenAbvGr    -0.135907
TotRmsAbvGrd     0.533723
Fireplaces       0.466929
GarageYrBlt      0.486362
GarageCars       0.640409
GarageArea       0.623431
WoodDeckSF       0.324413
OpenPorchSF      0.315856
EnclosedPorch   -0.128578
3SsnPorch        0.044584
ScreenPorch      0.111447
PoolArea         0.092404
MiscVal         -0.021190
MoSold           0.046432
YrSold          -0.028923
SalePrice        1.000000
Name: SalePrice, dtype: float64
```

Podemos notar que as variáveis com maior correlação são:

-OverallQual 0.790982 - Faz todo sentido, dado que aqui é uma nota que dão para o imóvel

-TotalBsmtSF 0.613581 - Talvez faça sentido o tamanho do sotão, pois existe a possibilidade de virar um espaço para alugar

-1stFlrSF 0.605852 - Aqui quanto maior a metragem do primeiro andar, maior a área comum, logo um valor maior

-GrLivArea 0.708624 - Idem ao de cima, tamanho maior da área externa de convivência

-GarageCars 0.640409 - Quantidade de carro que cabem na garagem valorizam o imóvel

-GarageArea 0.623431 - Muita correlação com o de cima (0.88)

3 - Desenvolvimento de modelo de Regressão utilizando Regressão Linear com o método de mínimos quadrados ordinários. Apresente as características do desenvolvimento: amostras, medidas de avaliação do modelo...

Primeiramente realizamos um tratamento nas variaveis categóricas e nos missings

```
categorical_data = ['MSSubClass','MSZoning','Street','Alley','LotShape','LandContour','Utilities','LotConfig',
'LandSlope','Neighborhood','Condition1','Condition2','BldgType','HouseStyle','RoofStyle','RoofMatl','Exterior1st',
'Exterior2nd','MasVnrType','ExterQual','ExterCond','Foundation','BsmtQual','BsmtCond','BsmtExposure','BsmtFinType1',
'BsmtFinType2','Heating','HeatingQC','CentralAir','Electrical','KitchenQual','Functional','FireplaceQu','GarageType'
'GarageFinish','GarageQual','GarageCond','PavedDrive','PoolQC','Fence','MiscFeature','SaleType','SaleCondition']

num_data = ['LotFrontage','LotArea','OverallQual','OverallCond','MasVnrArea','BsmtFinSF1','BsmtFinSF2','BsmtUnfSF',
'TotalBsmtSF','1stFlrSF','2ndFlrSF','LowQualFinSF','GrLivArea','BsmtFullBath','BsmtHalfBath','FullBath','HalfBath',
'BedroomAbvGr','KitchenAbvGr','TotRmsAbvGrd','Fireplaces','GarageCars','GarageArea','WoodDeckSF','OpenPorchSF',
'EnclosedPorch','3SsnPorch','ScreenPorch','PoolArea','MiscVal']

drop_data = ['id']

date_data = ['YearBuilt','YearRemodAdd','GarageYrBlt','MoSold','YrSold']

Y = df.SalePrice

X_cat_df = pd.get_dummies(df[categorical_data].fillna('NA'))

X_num_data = df[num_data].fillna(0)

df['garageTime'] = df.YrSold - df.GarageYrBlt

df['timeToSell'] = df.YrSold - df.YearBuilt

X = pd.concat([X_cat_df, X_num_data, df.garageTime.fillna(0), df.timeToSell.fillna(0)], axis = 1)
```

Importando as bibliotecas de ML

```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

Aplicamos uma taxa de amostragem para teste de 30%, deixando 70% para treino

```
 X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=1234)
```

Para selecionar as variaveis mais relevantes fitamos uma radom forest e selecionamos as 20 variaveis mais relevantes.

```
rf = RandomForestRegressor(n_estimators=900,n_jobs=-1, max_depth=5)
rf.fit(X_train,y_train)
best_feat = list(pd.DataFrame(rf.feature_importances_, index=X_train.columns,columns=['Importance']).sort_values('Im
best_feat
```

```
['OverallQual',
 'GrLivArea',
 '2ndFlrSF',
 'TotalBsmtSF',
 '1stFlrSF',
 'BsmtFinSF1',
 'LotArea',
 'GarageArea',
 'TotRmsAbvGrd',
 'FullBath',
 'GarageCars',
 'timeToSell',
 'MasVnrArea',
 'WoodDeckSF',
 'LotFrontage',
 'Fireplaces',
```

```
 'GarageType_Detchd',
 'OpenPorchSF',
 'KitchenQual_Gd',
 'FireplaceQu_NA']
```

```
lr = LinearRegression()
lr.fit(X_train[best_feat],y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

```
print('Test R2 =', r2_score(y_test, lr.predict(X_test[best_feat])))
print('Train R2 =', r2_score(y_train, lr.predict(X_train[best_feat])))
```

```
Test R2 = 0.8226256109446395
Train R2 = 0.7782552445955451
```

```
print('Test MAE =', mean_absolute_error(y_test, lr.predict(X_test[best_feat])))
print('Train MAE =', mean_absolute_error(y_train, lr.predict(X_train[best_feat])))
```

```
Test MAE = 22034.64190126019
Train MAE = 23742.679130796634
```

```
print('Test RMSE =', np.sqrt(mean_squared_error(y_test, lr.predict(X_test[best_feat]))))
print('Train RMSE =', np.sqrt(mean_squared_error(y_train, lr.predict(X_train[best_feat]))))
```

```
Test RMSE = 29774.417592378973
Train RMSE = 38998.84589651374
```

Importando as bibliotecas para o teste de ANOVA

```
from statsmodels.api import OLS, add_constant,graphics
from scipy import stats
```

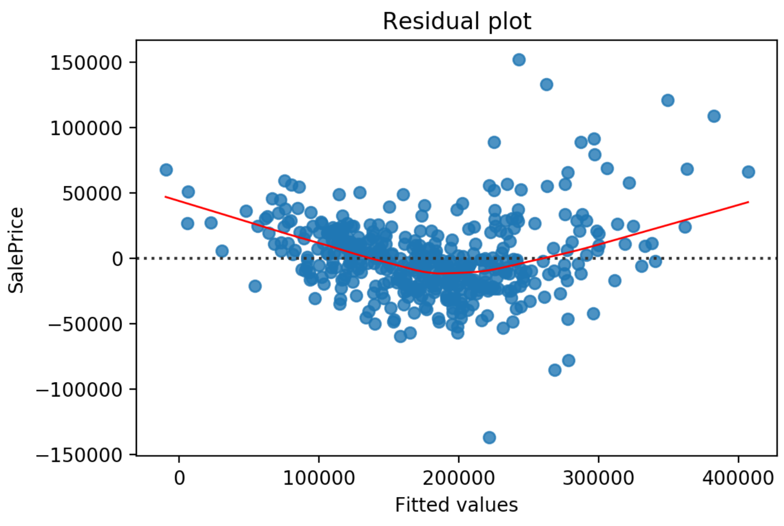
Avaliando a significancia dos coeficientes e analise de resíduos:

```
X2_train = add_constant(X_train[best_feat])
est = OLS(y_train, X2_train)
est2 = est.fit()
print(est2.summary())

seaborn.residplot(lr.predict(X_test[best_feat]),y_test, lowess=True,
                                  line_kws={'color': 'red', 'lw': 1, 'alpha': 1})
plt.xlabel("Fitted values")
plt.title('Residual plot')
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:
0.778
Model:                            OLS   Adj. R-squared:
0.774
Method:                 Least Squares   F-statistic:
175.7
Date:                Thu, 29 Jul 2021   Prob (F-statistic):
2.49e-310
Time:                        00:29:21   Log-Likelihood:
-12254.
No. Observations:                1022   AIC:
2.455e+04
Df Residuals:                    1001   BIC:
2.465e+04
Df Model:                          20
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|
[0.025      0.975]
-------------------------------------------------------------------------------------
const             -7.423e+04   1.16e+04     -6.421      0.000
-9.69e+04   -5.15e+04
OverallQual        2.219e+04   1503.530     14.759      0.000
1.92e+04    2.51e+04
GrLivArea            41.9901     26.280      1.598      0.110
-9.579      93.559
2ndFlrSF             -6.0037     26.479     -0.227      0.821
-57.964      45.957
TotalBsmtSF           8.3485      5.604      1.490      0.137
```

```
-2.648      19.345
1stFlrSF              -2.3063     26.917     -0.086      0.932
-55.127      50.515
BsmtFinSF1            14.9089      3.397      4.389      0.000
8.242      21.575
LotArea                0.3325      0.146      2.284      0.023
0.047       0.618
GarageArea            10.8056     12.868      0.840      0.401
-14.446      36.058
TotRmsAbvGrd        2143.3835   1407.089      1.523      0.128
-617.799    4904.566
FullBath            -400.3125   3426.292     -0.117      0.907
-7123.852    6323.227
GarageCars          1.142e+04   3890.672      2.934      0.003
3781.777    1.91e+04
timeToSell          -222.2222     66.547     -3.339      0.001
-352.810     -91.635
MasVnrArea            32.0146      7.761      4.125      0.000
16.785      47.244
WoodDeckSF            40.4396     10.432      3.877      0.000
19.969      60.911
LotFrontage            9.2440     37.745      0.245      0.807
-64.824      83.312
Fireplaces          7901.7778   4892.097      1.615      0.107
-1698.164    1.75e+04
GarageType_Detchd -5080.0312   3496.892     -1.453      0.147
-1.19e+04    1782.048
OpenPorchSF            6.2005     19.851      0.312      0.755
-32.753      45.154
KitchenQual_Gd     -2579.8473   2997.803     -0.861      0.390
-8462.546    3302.851
FireplaceQu_NA      3203.0148   6216.780      0.515      0.607
-8996.401    1.54e+04
==============================================================================
Omnibus:                      445.486   Durbin-Watson:
2.117
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
57093.269
Skew:                          -0.952   Prob(JB):
0.00
Kurtosis:                      39.567   Cond. No.
1.47e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
[2] The condition number is large, 1.47e+05. This might indicate that
there are
strong multicollinearity or other numerical problems.
/Users/karinseeder/anaconda3/lib/python3.7/site-
packages/statsmodels/tsa/tsatools.py:142: FutureWarning: In a future
version of pandas all arguments of concat except for the argument
'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

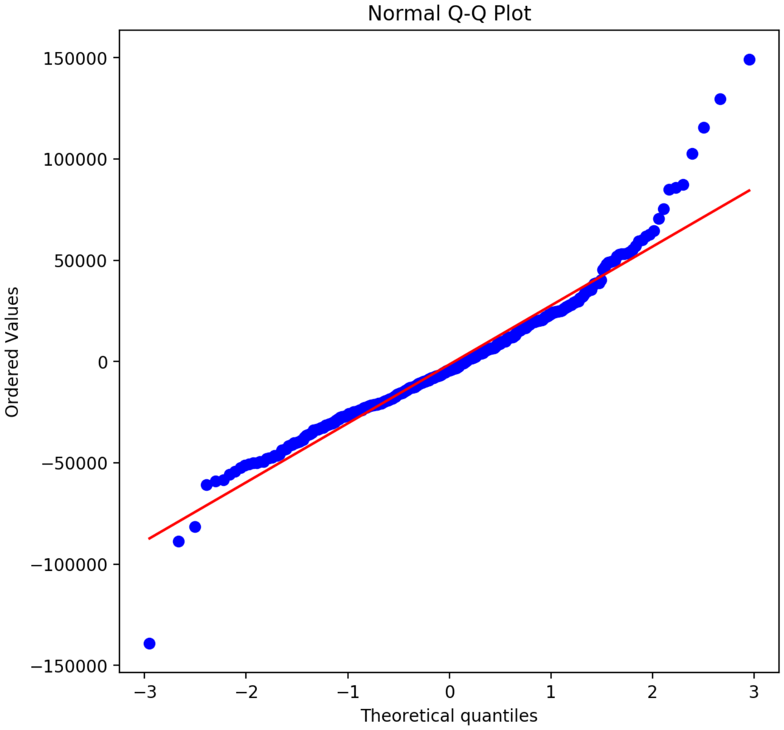
```
Text(0.5, 1.0, 'Residual plot')
```




```
residuals = y_test - lr.predict(X_test[best_feat])
residuals

plt.figure(figsize=(7,7))
```

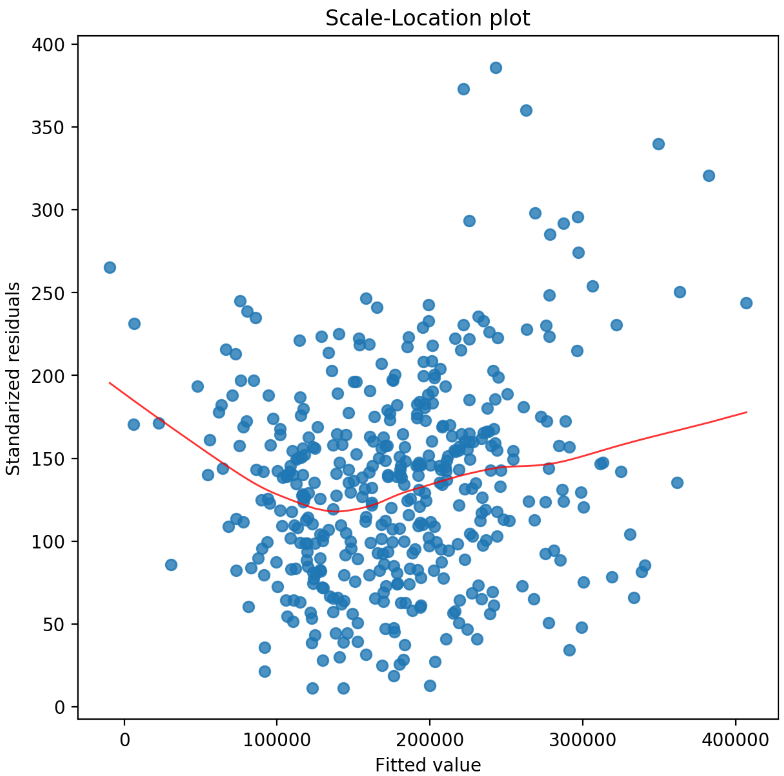
```
stats.probplot(residuals, dist="norm", plot=plt)
plt.title("Normal Q-Q Plot")
```

```
Text(0.5, 1.0, 'Normal Q-Q Plot')
```




```
model_norm_residuals_abs_sqrt=np.sqrt(np.abs(residuals))

plt.figure(figsize=(7,7))
seaborn.regplot(lr.predict(X_test[best_feat]), model_norm_residuals_abs_sqrt,
              scatter=True,
              lowess=True,
              line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.ylabel("Standarized residuals")
plt.xlabel("Fitted value")
plt.title('Scale-Location plot')
```

```
Text(0.5, 1.0, 'Scale-Location plot')
```



```python
from scipy import stats
shapiro_test = stats.shapiro(residuals)
print(shapiro_test)
```

```
ShapiroResult(statistic=0.9504468441009521,
pvalue=6.039371086563605e-11)
```

Conclusões

Analisando o R2 podemos verificar que o modelo explica 78% da variável resposta, ou seja, as variaveis escolhidas explicam 78% do preço da casa.

Avaliando a Prob-F, dado que ela é menor que 0.05 podemos rejeitar a hipótese H0 de que a regressão linear não existe.

Na métrica de Durbin-Watson tambem podemos concluir que não a correlação entre os resíduos.

Apenas com estas métricas poderiamos concluir que temos um bom modelo, porém ao continuar nossas análises podemos notar que tanto MAE e RMSE apresentam valores significantes e a diferença entre o RMSE e MAE nos permite supor que possa existir algum outlier e que os erros não são uniformes.

Pela analise de resíduo podemos verificar graficamente que os erros não são uniformemente distribuidos e que apresentam um leve formato de U. Aplicando o teste de shapiro com o resultado de p-value<0.05, isso nos leva a aceitar a hipotese nula de que os erros não seguem uma distribuição normal. O mesmo resultado pode ser obtido pela Prob Omnibus apresentada na tabela OLS Regression Results.

Portanto, podemos concluir que a regressão linear não é a melhor escolha.

Published from aula1.pmd using Pweave 0.30.3 on 29-07-2021.