In [14]: **import** pandas **as** pd import numpy as np import matplotlib.pyplot as plt In [15]: df=pd.read_csv('Downloads/Housing3/Transformed_Housing_Data.csv') In [16]: df.head() Out[16]: Flat Living Area Lot Area Area of the Age of No of Condition **Basement** House No of No of Area Lot Area No of Waterfront House from after after Ever Years Since of the ... ID Times Area (in Zipcode Latitude Longitude Price Bedrooms Bathrooms (in Sqft) Floors (in View **Basement** (in Renovation Renovation Renovated Renovation Visited House Sqft) Sqft) (in Sqft) Years) (in Sqft) (in Sqft) **0** 7129300520 221900.0 3 63 98178.0 47.5112 -122.257 1.00 1180.0 5650.0 1.0 No 0 Fair ... 1180.0 0 1340.0 5650 No 0 1 6414100192 538000.0 2.25 2570.0 2170.0 400 3 7242.0 2.0 67 98125.0 47.7210 -122.319 1690.0 7639 26 No 0 Fair ... Yes 2 5631500400 180000.0 2 1.00 770.0 10000.0 1.0 No 0 Fair ... 770.0 0 85 98028.0 47.7379 -122.233 2720.0 8062 No 0 **3** 2487200875 604000.0 3.00 1960.0 5000.0 1.0 0 Excellent ... 1050.0 910 53 98136.0 47.5208 -122.393 1360.0 5000 No 0 No 3 4 1954400510 510000.0 2.00 1680.0 8080.0 1.0 No Fair ... 1680.0 0 31 98074.0 47.6168 -122.045 1800.0 7503 No 0 5 rows × 21 columns In [5]: df.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 21609 entries, 0 to 21608 Data columns (total 21 columns): Column Non-Null Count Dtype --------0 ID 21609 non-null int64 Sale Price 21609 non-null float64 1 2 No of Bedrooms 21609 non-null int64 No of Bathrooms 21609 non-null float64 3 21609 non-null float64 Flat Area (in Sqft) 4 Lot Area (in Sqft) 21609 non-null float64 21609 non-null float64 No of Floors 7 Waterfront View 21609 non-null object 8 No of Times Visited 21609 non-null int64 9 Condition of the House 21609 non-null object 10 Overall Grade 21609 non-null int64 Area of the House from Basement (in Sqft) 21609 non-null float64 11 Basement Area (in Sqft) 21609 non-null int64 Age of House (in Years) 21609 non-null int64 14 Zipcode 21609 non-null float64 21609 non-null float64 15 Latitude 16 Longitude 21609 non-null float64 17 Living Area after Renovation (in Sqft) 21609 non-null float64 18 Lot Area after Renovation (in Sqft) 21609 non-null int64 19 Ever Renovated 21609 non-null object 21609 non-null int64 20 Years Since Renovation dtypes: float64(10), int64(8), object(3) memory usage: 3.5+ MB In [6]: df.drop(columns=['ID'],inplace=True) In [8]: df.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 21609 entries, 0 to 21608 Data columns (total 20 columns): Non-Null Count Dtype # Column 21609 non-null float64 0 Sale Price No of Bedrooms 21609 non-null int64 21609 non-null float64 No of Bathrooms Flat Area (in Sqft) 21609 non-null float64 3 Lot Area (in Sqft) 21609 non-null float64 No of Floors 21609 non-null float64 Waterfront View 21609 non-null object No of Times Visited 21609 non-null int64 Condition of the House 21609 non-null object 8 21609 non-null int64 Overall Grade 9 Area of the House from Basement (in Sqft) 21609 non-null float64 10 11 Basement Area (in Sqft) 21609 non-null int64 21609 non-null int64 12 Age of House (in Years) 21609 non-null float64 13 Zipcode 14 Latitude 21609 non-null float64 15 Longitude 21609 non-null float64 16 Living Area after Renovation (in Sqft) 21609 non-null float64 17 Lot Area after Renovation (in Sqft) 21609 non-null int64 21609 non-null object 18 Ever Renovated 19 Years Since Renovation 21609 non-null int64 dtypes: float64(10), int64(7), object(3) memory usage: 3.3+ MB In [8]: df['Condition of the House'].head() Out[8]: 0 Fair Fair 1 2 Fair 3 Excellent Fair Name: Condition of the House, dtype: object In [9]: df['Condition of the House'].unique() Out[9]: array(['Fair', 'Excellent', 'Good', 'Bad', 'Okay'], dtype=object) In [12]: df['Condition of the House'].value_counts() Out[12]: Condition of the House Fair 14028 Good 5678 Excellent 1701 172 0kay Bad 30 Name: count, dtype: int64 In [13]: df.groupby('Condition of the House')['Sale Price'].mean().plot(kind='bar') Out[13]: <Axes: xlabel='Condition of the House'> 500000 400000 300000 200000 100000 Excellent Condition of the House In [14]: df.groupby('Condition of the House')['Sale Price'].mean().sort_values().plot(kind='bar') Out[14]: <Axes: xlabel='Condition of the House'> 500000 400000 300000 200000 100000 Bad Fair Excellent Condition of the House In [16]: df.groupby('Waterfront View')['Sale Price'].mean().sort_values().plot(kind='bar') Out[16]: <Axes: xlabel='Waterfront View'> 1.0 0.8 0.6 0.4 0.2 0.0 Waterfront View In [17]: df.groupby('Ever Renovated')['Sale Price'].mean().sort_values().plot(kind='bar') Out[17]: <Axes: xlabel='Ever Renovated'> 600000 500000 400000 300000 200000 100000 2 Ever Renovated In [18]: df.groupby('Zipcode')['Sale Price'].mean().sort_values().plot(kind='bar') Out[18]: <Axes: xlabel='Zipcode'> 1e6 1.0 0.8 0.6 0.4 Zipcode In [17]: **from** statsmodels.formula.api **import** ols import statsmodels.api as sm In [29]: | df=df.rename(columns={'Condition of the House':'Condition_of_the_House'}) df=df.rename(columns={'Ever Renovated':'Ever_Renovated'}) In [31]: df=df.rename(columns={'Waterfront View':'Waterfront_View'}) In [32]: df=df.rename(columns={'Sale Price': 'Sale_Price'}) In [33]: df.head() Out[33]: Area of Age Flat Living Lot Area No of Lot Basement of Area No of House Area after Times Condition_of_the_House House Zipcode Latitude Longitude ID Sale_Price Area (in Waterfront_View Ever_Renovated Area (in Bedrooms Bathrooms (in **Floors** from Renovation Renovation Sqft) Visited Sqft) Sqft) Basement (in Sqft) (in Sqft) Years) (in Sqft) **0** 7129300520 221900.0 1.00 1180.0 5650.0 1.0 0 63 98178.0 47.5112 -122.257 1340.0 5650 3 Fair 1180.0 0 No No 538000.0 2.25 2570.0 7639 1 6414100192 3 7242.0 2.0 No Fair .. 2170.0 400 67 98125.0 47.7210 -122.319 1690.0 Yes **2** 5631500400 180000.0 2 1.00 770.0 10000.0 1.0 No 0 Fair .. 770.0 0 85 98028.0 47.7379 -122.233 2720.0 8062 No 3.00 1960.0 53 98136.0 47.5208 **3** 2487200875 604000.0 5000.0 1.0 No Excellent 1050.0 910 -122.393 1360.0 5000 No **4** 1954400510 510000.0 3 2.00 1680.0 0.0808 1.0 0 1680.0 0 31 98074.0 47.6168 -122.045 1800.0 7503 No No Fair 5 rows × 21 columns In [22]: df['Waterfront_View'].unique() Out[22]: array(['No', 'Yes'], dtype=object) mod=ols('Sale_Price ~ Condition_of_the_House', data=df).fit() In [35]: Anova_Table=sm.stats.anova_lm(mod, tyo=2) In [36]: print(Anova_Table) df F \ sum_sq mean_sq Condition_of_the_House 4.0 1.441831e+13 3.604579e+12 58.25563 21604.0 1.336752e+15 6.187520e+10 Residual PR(>F) Condition_of_the_House 5.447605e-49 In [38]: df=pd.get_dummies(df,columns=['Condition_of_the_House'],drop_first=True) In [39]: df.head() Out[39]: Flat Lot Area Living Years Lot No of No of No of Area No of Overall Area after ID Sale_Price Waterfront_View Since Area (in ... Latitude Longitude Ever_Renovated Condition_of_the_House_Excellent Condition_o Times Bedrooms Bathrooms Floors Grade Renovation Renovation Visited Sqft) Renovation Sqft) (in Sqft) (in Sqft) 3 5650.0 7 ... 47.5112 -122.257 1340.0 5650 No 0 False **0** 7129300520 221900.0 1.00 1180.0 1.0 No 0 1 6414100192 538000.0 3 2.25 2570.0 7242.0 2.0 No 0 7 ... 47.7210 -122.319 1690.0 7639 Yes 26 False 0 **2** 5631500400 180000.0 2 1.00 770.0 10000.0 1.0 No 0 6 ... 47.7379 -122.233 2720.0 8062 No False **3** 2487200875 604000.0 3.00 1960.0 5000.0 1.0 7 ... 47.5208 -122.393 1360.0 5000 0 No True No 3 8 ... 47.6168 -122.045 7503 0 **4** 1954400510 510000.0 2.00 1680.0 8080.0 1.0 1800.0 No False No

5 rows × 24 columns