**Python Scripts**

* **Combined Zillow & Realtor Housing Data and cleaned up the columns (cleaned\_housing\_data.csv)**

import pandas as pd

from functools import reduce

print("Script started")

# 1. Load data

realtor = pd.read\_csv('Realtor raw data.csv')

print("Realtor data loaded")

zillow\_days = pd.read\_csv('Zillow Days on Market - Metro & U.S - raw data.csv')

zillow\_days.columns = zillow\_days.columns.str.strip()

print("Zillow Days on Market loaded")

zillow\_for\_sale = pd.read\_csv('Zillow For Sale Listings - Metro & U.S. - raw data.csv')

zillow\_for\_sale.columns = zillow\_for\_sale.columns.str.strip()

print("Zillow For Sale Listings loaded")

zillow\_home\_values = pd.read\_csv('Zillow Home Values - Metro & U.S - raw data.csv')

zillow\_home\_values.columns = zillow\_home\_values.columns.str.strip()

print("Zillow Home Values loaded")

zillow\_sales = pd.read\_csv('Zillow Sales - Metro & U.S - raw data.csv')

zillow\_sales.columns = zillow\_sales.columns.str.strip()

print("Zillow Sales loaded")

# 2. Melt Zillow files from wide to long format (only date columns)

def melt\_zillow(df, value\_name):

id\_vars = ['StateName', 'RegionName', 'RegionType', 'RegionID', 'SizeRank']

id\_vars = [col for col in id\_vars if col in df.columns]

value\_vars = [col for col in df.columns if col not in id\_vars and col[:4].isdigit()]

df\_long = df.melt(id\_vars=id\_vars, value\_vars=value\_vars, var\_name='Date', value\_name=value\_name)

df\_long['Date'] = pd.to\_datetime(df\_long['Date'], errors='coerce')

if 'StateName' in df\_long.columns:

result = df\_long[['StateName', 'Date', value\_name]]

elif 'state' in df\_long.columns:

result = df\_long[['state', 'Date', value\_name]].rename(columns={'state': 'StateName'})

else:

raise ValueError("No StateName or state column found in Zillow file after melting.")

return result

zillow\_days\_long = melt\_zillow(zillow\_days, 'avg\_days\_on\_market\_zillow')

print("Zillow Days melted")

zillow\_for\_sale\_long = melt\_zillow(zillow\_for\_sale, 'avg\_active\_listing\_count\_zillow')

print("Zillow For Sale Listings melted")

zillow\_home\_values\_long = melt\_zillow(zillow\_home\_values, 'avg\_listing\_price\_zillow')

print("Zillow Home Values melted")

zillow\_sales\_long = melt\_zillow(zillow\_sales, 'avg\_pending\_listing\_count\_zillow')

print("Zillow Sales melted")

# 3. Drop NaNs and aggregate to state level before merging (using mean for average)

for df in [zillow\_days\_long, zillow\_for\_sale\_long, zillow\_home\_values\_long, zillow\_sales\_long]:

df.dropna(subset=['StateName', 'Date'], inplace=True)

zillow\_days\_long = zillow\_days\_long.groupby(['StateName', 'Date'], as\_index=False).mean(numeric\_only=True)

zillow\_for\_sale\_long = zillow\_for\_sale\_long.groupby(['StateName', 'Date'], as\_index=False).mean(numeric\_only=True)

zillow\_home\_values\_long = zillow\_home\_values\_long.groupby(['StateName', 'Date'], as\_index=False).mean(numeric\_only=True)

zillow\_sales\_long = zillow\_sales\_long.groupby(['StateName', 'Date'], as\_index=False).mean(numeric\_only=True)

# 4. Merge all Zillow datasets on StateName and Date

zillow\_merged = reduce(

lambda left, right: pd.merge(left, right, on=['StateName', 'Date'], how='outer'),

[zillow\_days\_long, zillow\_for\_sale\_long, zillow\_home\_values\_long, zillow\_sales\_long]

)

print("Zillow datasets merged")

# 5. Prepare Realtor data

realtor = realtor.rename(columns={'month\_date\_yyyymm': 'Date', 'state': 'StateName'})

realtor['Date'] = pd.to\_datetime(realtor['Date'], format='%Y%m', errors='coerce')

print("Realtor data prepared")

# Rename Realtor columns for consistency (change to avg)

realtor = realtor.rename(columns={

'median\_days\_on\_market': 'avg\_days\_on\_market\_realtor',

'pending\_listing\_count': 'avg\_pending\_listing\_count\_realtor',

'active\_listing\_count': 'avg\_active\_listing\_count\_realtor',

'median\_listing\_price': 'avg\_listing\_price\_realtor'

})

# 6. Merge Realtor with aggregated Zillow data

final\_df = pd.merge(

realtor, zillow\_merged, on=['StateName', 'Date'],

how='outer', suffixes=('\_realtor', '\_zillow')

)

print("Realtor and Zillow data merged")

# 7. Keep only the columns you want

cols\_to\_keep = (

['Date', 'StateName'] +

[

'avg\_days\_on\_market\_zillow', 'avg\_pending\_listing\_count\_zillow', 'avg\_active\_listing\_count\_zillow', 'avg\_listing\_price\_zillow',

'avg\_days\_on\_market\_realtor', 'avg\_pending\_listing\_count\_realtor', 'avg\_active\_listing\_count\_realtor', 'avg\_listing\_price\_realtor'

]

)

cols\_to\_keep = [col for col in cols\_to\_keep if col in final\_df.columns]

final\_df = final\_df[cols\_to\_keep]

# 8. Drop rows where all Zillow & Realtor columns are blank except Date/StateName

value\_cols = [

'avg\_days\_on\_market\_zillow', 'avg\_pending\_listing\_count\_zillow', 'avg\_active\_listing\_count\_zillow', 'avg\_listing\_price\_zillow',

'avg\_days\_on\_market\_realtor', 'avg\_pending\_listing\_count\_realtor', 'avg\_active\_listing\_count\_realtor', 'avg\_listing\_price\_realtor'

]

value\_cols = [col for col in value\_cols if col in final\_df.columns]

final\_df = final\_df.dropna(subset=value\_cols, how='all')

print("Blank rows dropped")

# 9. Save cleaned data with null for missing values

final\_df.to\_csv('cleaned\_housing\_data.csv', index=False, na\_rep='null')

print("Cleaned data saved as cleaned\_housing\_data.csv (null for blanks)")

# 10. Show first 5 rows with all Zillow columns present (no NaN in any Zillow column)

zillow\_cols = [col for col in value\_cols if col.endswith('\_zillow')]

zillow\_rows = final\_df.dropna(subset=zillow\_cols, how='any')

print("\nFirst 5 rows with Zillow values (all Zillow columns present):")

print(zillow\_rows.head())

# 11. Show first 5 rows with all Realtor columns present (no NaN in any Realtor column)

realtor\_cols = [col for col in value\_cols if col.endswith('\_realtor')]

realtor\_rows = final\_df.dropna(subset=realtor\_cols, how='any')

print("\nFirst 5 rows with Realtor values (all Realtor columns present):")

print(realtor\_rows.head())