Data Analysis

Comprehensive Data Cleaning & Exploratory Analysis of Job Market Trends

Yibei Yu, Fuhan Zhang, Jonathan Leon

import pandas as pd  
import missingno as msno  
import matplotlib.pyplot as plt  
import numpy as np  
import missingno as msno  
import matplotlib.pyplot as plt  
df = pd.read\_csv("data/lightcast\_job\_postings.csv")  
print(df.columns.tolist())  
  
columns\_to\_drop = [  
 "ID", "URL", "ACTIVE\_URLS", "DUPLICATES", "LAST\_UPDATED\_TIMESTAMP",  
 "NAICS2", "NAICS3", "NAICS4", "NAICS5", "NAICS6",  
 "SOC\_2", "SOC\_3", "SOC\_5"  
]  
df.drop(columns=columns\_to\_drop, inplace=True, errors="ignore")  
  
plt.figure(figsize=(10,6))  
msno.heatmap(df)  
plt.title("Missing Values Heatmap")  
plt.tight\_layout()  
thresh = len(df) \* 0.5  
df = df.loc[:, df.notna().sum() >= thresh]  
df.shape  
  
num\_cols = df.select\_dtypes(include=[np.number]).columns.tolist()  
cat\_cols = [c for c in df.columns if c not in num\_cols]  
for c in num\_cols:  
 df[c] = pd.to\_numeric(df[c], errors="coerce")  
df[num\_cols] = df[num\_cols].fillna(df[num\_cols].median())  
for c in cat\_cols:  
 df[c] = df[c].fillna("Unknown")  
salary\_candidates = ["SALARY", "Salary", "Average\_Salary", "AVERAGE\_SALARY"]  
salary\_col = next((c for c in salary\_candidates if c in df.columns), None)  
  
if salary\_col:  
 df[salary\_col] = pd.to\_numeric(df[salary\_col], errors="coerce")  
 q1, q3 = df[salary\_col].quantile([0.25, 0.75])  
 iqr = q3 - q1  
 lo, hi = q1 - 1.5\*iqr, q3 + 1.5\*iqr  
 df[salary\_col] = df[salary\_col].clip(lower=lo, upper=hi)  
dup\_keys = [c for c in ["TITLE", "COMPANY", "LOCATION", "POSTED"] if c in df.columns]  
if dup\_keys:  
 before = len(df)  
 df = df.drop\_duplicates(subset=dup\_keys, keep="first")  
 after = len(df)  
 print(f"Removed duplicates: {before - after}")  
else:  
 df = df.drop\_duplicates()  
if "Industry" in df.columns:  
 plt.figure(figsize=(10,5))  
 df["Industry"].value\_counts().head(15).plot(kind="bar")  
 plt.title("Job Postings by Industry (Top 15)")  
 plt.xlabel("Industry")  
 plt.ylabel("Count")  
 plt.tight\_layout()  
 plt.show()  
if salary\_col and "Industry" in df.columns:  
 plt.figure(figsize=(12,6))  
 top\_ind = df["Industry"].value\_counts().head(10).index  
 tmp = df[df["Industry"].isin(top\_ind)]  
 tmp.boxplot(column=salary\_col, by="Industry", rot=45)  
 plt.title(f"Salary Distribution by Industry ({salary\_col})")  
 plt.suptitle("")  
 plt.xlabel("Industry")  
 plt.ylabel("Salary")  
 plt.tight\_layout()  
 plt.show()  
remote\_col = next((c for c in ["REMOTE\_TYPE\_NAME","REMOTE\_TYPE","Remote\_Type","REMOTE\_GROUP"] if c in df.columns), None)  
if remote\_col:  
 plt.figure(figsize=(6,6))  
 df[remote\_col].fillna("Unknown").value\_counts().plot(kind="pie", autopct="%1.1f%%")  
 plt.title("Remote vs. On-Site Jobs")  
 plt.ylabel("")  
 plt.tight\_layout()  
 plt.show()

/tmp/ipykernel\_23161/636741187.py:7: DtypeWarning: Columns (19,30) have mixed types. Specify dtype option on import or set low\_memory=False.  
 df = pd.read\_csv("data/lightcast\_job\_postings.csv")

['ID', 'LAST\_UPDATED\_DATE', 'LAST\_UPDATED\_TIMESTAMP', 'DUPLICATES', 'POSTED', 'EXPIRED', 'DURATION', 'SOURCE\_TYPES', 'SOURCES', 'URL', 'ACTIVE\_URLS', 'ACTIVE\_SOURCES\_INFO', 'TITLE\_RAW', 'BODY', 'MODELED\_EXPIRED', 'MODELED\_DURATION', 'COMPANY', 'COMPANY\_NAME', 'COMPANY\_RAW', 'COMPANY\_IS\_STAFFING', 'EDUCATION\_LEVELS', 'EDUCATION\_LEVELS\_NAME', 'MIN\_EDULEVELS', 'MIN\_EDULEVELS\_NAME', 'MAX\_EDULEVELS', 'MAX\_EDULEVELS\_NAME', 'EMPLOYMENT\_TYPE', 'EMPLOYMENT\_TYPE\_NAME', 'MIN\_YEARS\_EXPERIENCE', 'MAX\_YEARS\_EXPERIENCE', 'IS\_INTERNSHIP', 'SALARY', 'REMOTE\_TYPE', 'REMOTE\_TYPE\_NAME', 'ORIGINAL\_PAY\_PERIOD', 'SALARY\_TO', 'SALARY\_FROM', 'LOCATION', 'CITY', 'CITY\_NAME', 'COUNTY', 'COUNTY\_NAME', 'MSA', 'MSA\_NAME', 'STATE', 'STATE\_NAME', 'COUNTY\_OUTGOING', 'COUNTY\_NAME\_OUTGOING', 'COUNTY\_INCOMING', 'COUNTY\_NAME\_INCOMING', 'MSA\_OUTGOING', 'MSA\_NAME\_OUTGOING', 'MSA\_INCOMING', 'MSA\_NAME\_INCOMING', 'NAICS2', 'NAICS2\_NAME', 'NAICS3', 'NAICS3\_NAME', 'NAICS4', 'NAICS4\_NAME', 'NAICS5', 'NAICS5\_NAME', 'NAICS6', 'NAICS6\_NAME', 'TITLE', 'TITLE\_NAME', 'TITLE\_CLEAN', 'SKILLS', 'SKILLS\_NAME', 'SPECIALIZED\_SKILLS', 'SPECIALIZED\_SKILLS\_NAME', 'CERTIFICATIONS', 'CERTIFICATIONS\_NAME', 'COMMON\_SKILLS', 'COMMON\_SKILLS\_NAME', 'SOFTWARE\_SKILLS', 'SOFTWARE\_SKILLS\_NAME', 'ONET', 'ONET\_NAME', 'ONET\_2019', 'ONET\_2019\_NAME', 'CIP6', 'CIP6\_NAME', 'CIP4', 'CIP4\_NAME', 'CIP2', 'CIP2\_NAME', 'SOC\_2021\_2', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4', 'SOC\_2021\_4\_NAME', 'SOC\_2021\_5', 'SOC\_2021\_5\_NAME', 'LOT\_CAREER\_AREA', 'LOT\_CAREER\_AREA\_NAME', 'LOT\_OCCUPATION', 'LOT\_OCCUPATION\_NAME', 'LOT\_SPECIALIZED\_OCCUPATION', 'LOT\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_OCCUPATION\_GROUP', 'LOT\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_SPECIALIZED\_OCCUPATION', 'LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION', 'LOT\_V6\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION\_GROUP', 'LOT\_V6\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_CAREER\_AREA', 'LOT\_V6\_CAREER\_AREA\_NAME', 'SOC\_2', 'SOC\_2\_NAME', 'SOC\_3', 'SOC\_3\_NAME', 'SOC\_4', 'SOC\_4\_NAME', 'SOC\_5', 'SOC\_5\_NAME', 'LIGHTCAST\_SECTORS', 'LIGHTCAST\_SECTORS\_NAME', 'NAICS\_2022\_2', 'NAICS\_2022\_2\_NAME', 'NAICS\_2022\_3', 'NAICS\_2022\_3\_NAME', 'NAICS\_2022\_4', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_5', 'NAICS\_2022\_5\_NAME', 'NAICS\_2022\_6', 'NAICS\_2022\_6\_NAME']  
Removed duplicates: 3300

<Figure size 960x576 with 0 Axes>



