Assignment 03

Julio Vargas

September 23, 2025

from pyspark.sql import SparkSession  
from pyspark.sql import functions as F  
from pyspark.sql.functions import col, split, explode, regexp\_replace, transform, when  
from pyspark.sql.functions import col, monotonically\_increasing\_id  
from pyspark.sql.types import StructType # to/from JSON  
  
import json  
import re  
import numpy as np  
import pandas as pd  
  
import plotly.express as px  
import plotly.io as pio  
import plotly.graph\_objects as go  
  
  
np.random.seed(30) # set a fixed seed for reproducibility  
pio.renderers.default = "vscode+notebook" #  
# Initialize Spark Session  
spark = SparkSession.builder.appName("JobPostingsAnalysis").getOrCreate()  
# Load schema from JSON file  
with open("data/schema\_lightcast.json") as f:  
 schema = StructType.fromJson(json.load(f))  
  
# Load Data  
df = (spark.read  
 .option("header", "true")  
 .option("inferSchema", "false")  
 .schema(schema) # saved schema  
 .option("multiLine", "true")  
 .option("escape", "\"")  
 .csv("data/lightcast\_job\_postings.csv")  
 )  
  
df.createOrReplaceTempView("job\_postings")  
# Show Schema and Sample Data  
#df.printSchema()   
df.show(5)

+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
| 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|  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
|1f57d95acf4dc67ed...| 9/6/2024| 2024-09-06 20:32:...| 0|6/2/2024| 6/8/2024| 6| [\n "Company"\n]|[\n "brassring.c...|[\n "https://sjo...| []| NULL|Enterprise Analys...|31-May-2024\n\nEn...| 6/8/2024| 6| 894731| Murphy USA| Murphy USA| false| [\n 2\n]| [\n "Bachelor's ...| 2| Bachelor's degree| NULL| NULL| 1|Full-time (> 32 h...| 2| 2| false| NULL| 0| [None]| NULL| NULL| NULL|{\n "lat": 33.20...|RWwgRG9yYWRvLCBBUg==|El Dorado, AR| 5139| Union, AR|20980| El Dorado, AR| 5| Arkansas| 5139| Union, AR| 5139| Union, AR| 20980| El Dorado, AR| 20980| El Dorado, AR| 44| Retail Trade| 441|Motor Vehicle and...| 4413|Automotive Parts,...| 44133|Automotive Parts ...|441330|Automotive Parts ...|ET29C073C03D1F86B4|Enterprise Analysts|enterprise analys...|[\n "KS126DB6T06...|[\n "Merchandisi...|[\n "KS126DB6T06...| [\n "Merchandisi...| []| []|[\n "KS126706DPF...|[\n "Mathematics...|[\n "KS440W865GC...|[\n "SQL (Progra...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...|[\n "45.0601",\n...|[\n "Economics, ...|[\n "45.06",\n ...|[\n "Economics",...|[\n "45",\n "27...|[\n "Social Scie...| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231010|Business Intellig...| 23101011| General ERP Analy...| 2310| Business Intellig...| 23101011| General ERP Analy...| 231010| Business Intellig...| 2310| Business Intellig...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| [\n 7\n]| [\n "Artificial ...| 44| Retail Trade| 441|Motor Vehicle and...| 4413|Automotive Parts,...| 44133|Automotive Parts ...| 441330|Automotive Parts ...|  
|0cb072af26757b6c4...| 8/2/2024| 2024-08-02 17:08:...| 0|6/2/2024| 8/1/2024| NULL| [\n "Job Board"\n]| [\n "maine.gov"\n]|[\n "https://job...| []| NULL|Oracle Consultant...|Oracle Consultant...| 8/1/2024| NULL| 133098|Smx Corporation L...| SMX| true| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 1|Full-time (> 32 h...| 3| 3| false| NULL| 1| Remote| NULL| NULL| NULL|{\n "lat": 44.31...| QXVndXN0YSwgTUU=| Augusta, ME| 23011| Kennebec, ME|12300|Augusta-Watervill...| 23| Maine| 23011| Kennebec, ME| 23011| Kennebec, ME| 12300|Augusta-Watervill...| 12300|Augusta-Watervill...| 56|Administrative an...| 561|Administrative an...| 5613| Employment Services| 56132|Temporary Help Se...|561320|Temporary Help Se...|ET21DDA63780A7DC09| Oracle Consultants|oracle consultant...|[\n "KS122626T55...|[\n "Procurement...|[\n "KS122626T55...| [\n "Procurement...| []| []| []| []|[\n "BGSBF3F508F...|[\n "Oracle Busi...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231010|Business Intellig...| 23101012| Oracle Consultant...| 2310| Business Intellig...| 23101012| Oracle Consultant...| 231010| Business Intellig...| 2310| Business Intellig...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| NULL| NULL| 56|Administrative an...| 561|Administrative an...| 5613| Employment Services| 56132|Temporary Help Se...| 561320|Temporary Help Se...|  
|85318b12b3331fa49...| 9/6/2024| 2024-09-06 20:32:...| 1|6/2/2024| 7/7/2024| 35| [\n "Job Board"\n]|[\n "dejobs.org"\n]|[\n "https://dej...| []| NULL| Data Analyst|Taking care of pe...| 6/10/2024| 8|39063746| Sedgwick| Sedgwick| false| [\n 2\n]| [\n "Bachelor's ...| 2| Bachelor's degree| NULL| NULL| 1|Full-time (> 32 h...| 5| NULL| false| NULL| 0| [None]| NULL| NULL| NULL|{\n "lat": 32.77...| RGFsbGFzLCBUWA==| Dallas, TX| 48113| Dallas, TX|19100|Dallas-Fort Worth...| 48| Texas| 48113| Dallas, TX| 48113| Dallas, TX| 19100|Dallas-Fort Worth...| 19100|Dallas-Fort Worth...| 52|Finance and Insur...| 524|Insurance Carrier...| 5242|Agencies, Brokera...| 52429|Other Insurance R...|524291| Claims Adjusting|ET3037E0C947A02404| Data Analysts| data analyst|[\n "KS1218W78FG...|[\n "Management"...|[\n "ESF3939CE1F...| [\n "Exception R...|[\n "KS683TN76T7...|[\n "Security Cl...|[\n "KS1218W78FG...|[\n "Management"...|[\n "KS126HY6YLT...|[\n "Microsoft O...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231113|Data / Data Minin...| 23111310| Data Analyst| 2311| Data Analysis and...| 23111310| Data Analyst| 231113| Data / Data Minin...| 2311| Data Analysis and...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| NULL| NULL| 52|Finance and Insur...| 524|Insurance Carrier...| 5242|Agencies, Brokera...| 52429|Other Insurance R...| 524291| Claims Adjusting|  
|1b5c3941e54a1889e...| 9/6/2024| 2024-09-06 20:32:...| 1|6/2/2024|7/20/2024| 48| [\n "Job Board"\n]|[\n "disabledper...|[\n "https://www...| []| NULL|Sr. Lead Data Mgm...|About this role:\...| 6/12/2024| 10|37615159| Wells Fargo|Wells Fargo| false| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 1|Full-time (> 32 h...| 3| NULL| false| NULL| 0| [None]| NULL| NULL| NULL|{\n "lat": 33.44...| UGhvZW5peCwgQVo=| Phoenix, AZ| 4013| Maricopa, AZ|38060|Phoenix-Mesa-Chan...| 4| Arizona| 4013| Maricopa, AZ| 4013| Maricopa, AZ| 38060|Phoenix-Mesa-Chan...| 38060|Phoenix-Mesa-Chan...| 52|Finance and Insur...| 522|Credit Intermedia...| 5221|Depository Credit...| 52211| Commercial Banking|522110| Commercial Banking|ET2114E0404BA30075|Management Analysts|sr lead data mgmt...|[\n "KS123QX62QY...|[\n "Exit Strate...|[\n "KS123QX62QY...| [\n "Exit Strate...| []| []|[\n "KS7G6NP6R6L...|[\n "Reliability...|[\n "KS4409D76NW...|[\n "SAS (Softwa...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231113|Data / Data Minin...| 23111310| Data Analyst| 2311| Data Analysis and...| 23111310| Data Analyst| 231113| Data / Data Minin...| 2311| Data Analysis and...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| [\n 6\n]| [\n "Data Privac...| 52|Finance and Insur...| 522|Credit Intermedia...| 5221|Depository Credit...| 52211| Commercial Banking| 522110| Commercial Banking|  
|cb5ca25f02bdf25c1...| 6/19/2024| 2024-06-19 07:00:00| 0|6/2/2024|6/17/2024| 15|[\n "FreeJobBoar...|[\n "craigslist....|[\n "https://mod...| []| NULL|Comisiones de $10...|Comisiones de $10...| 6/17/2024| 15| 0| Unclassified| LH/GM| false| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 3|Part-time / full-...| NULL| NULL| false| 92500| 0| [None]| year| 150000| 35000|{\n "lat": 37.63...| TW9kZXN0bywgQ0E=| Modesto, CA| 6099|Stanislaus, CA|33700| Modesto, CA| 6|California| 6099| Stanislaus, CA| 6099| Stanislaus, CA| 33700| Modesto, CA| 33700| Modesto, CA| 99|Unclassified Indu...| 999|Unclassified Indu...| 9999|Unclassified Indu...| 99999|Unclassified Indu...|999999|Unclassified Indu...|ET0000000000000000| Unclassified|comisiones de por...| []| []| []| []| []| []| []| []| []| []|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231010|Business Intellig...| 23101012| Oracle Consultant...| 2310| Business Intellig...| 23101012| Oracle Consultant...| 231010| Business Intellig...| 2310| Business Intellig...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| NULL| NULL| 99|Unclassified Indu...| 999|Unclassified Indu...| 9999|Unclassified Indu...| 99999|Unclassified Indu...| 999999|Unclassified Indu...|  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
only showing top 5 rows

# Histogram of SALARY distribution (cast + filter)  
salary\_df = (  
 df.select(col("SALARY").cast("float"))  
 .filter(col("SALARY").isNotNull() & (col("SALARY") > 0))  
)  
  
fig = px.histogram(  
 salary\_df.toPandas(),  
 x="SALARY",  
 nbins=50,  
 title="Salary Distribution"  
)  
fig.update\_layout(bargap=0.1)  
fig

[Stage 14:> (0 + 1) / 1]

Unable to display output for mime type(s): text/html

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

# 1. Data Preparation

# Step 1: Casting salary and experience columns  
from pyspark.sql.functions import col  
  
df = df.withColumn("SALARY", col("SALARY").cast("float")) \  
 .withColumn("SALARY\_FROM", col("SALARY\_FROM").cast("float")) \  
 .withColumn("SALARY\_TO", col("SALARY\_TO").cast("float")) \  
 .withColumn("MIN\_YEARS\_EXPERIENCE", col("MIN\_YEARS\_EXPERIENCE").cast("float"))\  
 .withColumn("MAX\_YEARS\_EXPERIENCE", col("MAX\_YEARS\_EXPERIENCE").cast("float"))  
  
# Step 2: Computing medians for salary columns  
def compute\_median(sdf, col\_name):  
 q = sdf.approxQuantile(col\_name, [0.5], 0.01)  
 return q[0] if q else None  
  
median\_from = compute\_median(df, "SALARY\_FROM")  
median\_to = compute\_median(df, "SALARY\_TO")  
median\_salary = compute\_median(df, "SALARY")  
  
print("Medians:", median\_from, median\_to)  
  
# Step 3: Imputing missing salaries, but not experience  
df = df.fillna({  
 "SALARY\_FROM": median\_from,  
 "SALARY\_TO": median\_to,  
 "SALARY": median\_salary  
})  
  
# Step 5: Computing average salary  
df = df.withColumn("Average\_Salary",  
 (col("SALARY\_FROM") + col("SALARY\_TO")) / 2)  
  
# Step 6: Selecting required columns  
export\_cols = [  
 "EDUCATION\_LEVELS\_NAME",  
 "REMOTE\_TYPE\_NAME",  
 "MAX\_YEARS\_EXPERIENCE",  
 "Average\_Salary",  
 "SALARY",  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"  
]  
df\_selected = df.select(\*export\_cols)  
  
# Step 7: Saving to CSV  
pdf = df\_selected.toPandas()  
pdf.to\_csv("data/lightcast\_cleaned.csv", index=False)  
  
print("Data cleaning complete. Rows retained:", len(pdf))

[Stage 15:> (0 + 1) / 1] [Stage 16:> (0 + 1) / 1] [Stage 17:> (0 + 1) / 1]

Medians: 87295.0 130042.0

[Stage 18:> (0 + 1) / 1]

Data cleaning complete. Rows retained: 72498

## 1.1 Salary Distribution by Industry and Employment Type

* Compare salary variations across industries.

**Filter the dataset** - Remove records where **salary is missing or zero**.

**Aggregate Data** - Group by **NAICS industry codes** (e.g., NAICS2\_NAME). - Group by **employment type** (EMPLOYMENT\_TYPE\_NAME) and compute salary distribution. - Calculate **salary percentiles** (25th, 50th, 75th) for each group.

**Visualize results** - Create a **box plot** where: - **X-axis** = NAICS2\_NAME - **Y-axis** = SALARY\_FROM, or SALARY\_TO, or SALARY - **Group/color** = EMPLOYMENT\_TYPE\_NAME - Customize colors, fonts, and styles.

**Explanation:** Write two sentences about what the graph reveals (e.g., median differences across industries and dispersion by employment type).

# 2. Set up plotly template

import plotly.graph\_objects as go  
import plotly.io as pio  
  
pio.templates["nike"] = go.layout.Template(  
 # LAYOUT  
 layout = {  
 # Fonts  
 # Note - 'family' must be a single string, NOT a list or dict!  
 'title': {  
 'font': {'family': 'HelveticaNeue-CondensedBold, Helvetica, Sans-serif',  
 'size': 30,  
 'color': '#13007c'} # 333  
 },  
 'font': {'family': 'Helvetica Neue, Helvetica, Sans-serif',  
 'size': 16,  
 'color': '#3b3b3b'}, # 333  
  
 # Colorways ec7424  
 'colorway': ['#fffb00', '#e010fc'], # CC0000, a4adab  
 # Keep adding others as needed below  
 'hovermode': 'x unified'  
 },  
 # DATA  
 data = {  
 # Each graph object must be in a tuple or list for each trace  
 'bar': [go.Bar(  
 texttemplate = '%{value:$.2s}',  
 textposition = 'outside',  
 textfont = {'family': 'Helvetica Neue, Helvetica, Sans-serif',  
 'size': 20,  
 'color': '#ff6874'} # FFFFFF  
 )]  
 }  
)

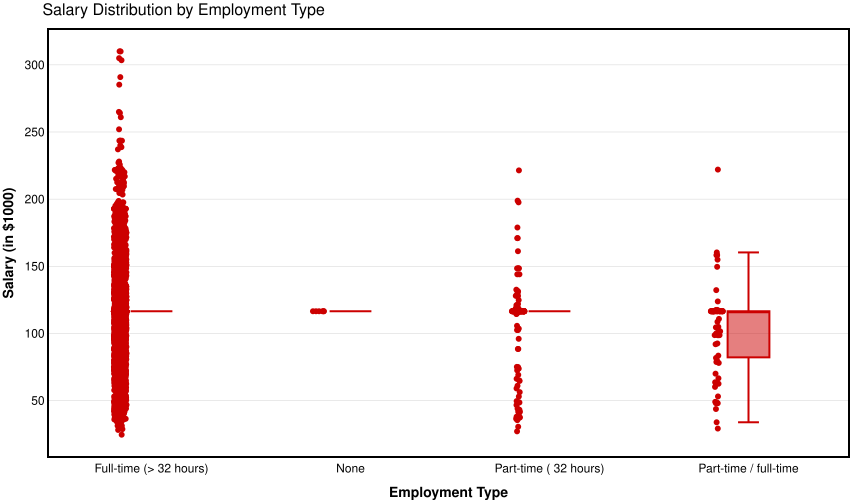
#your code for first query  
import pandas as pd  
import polars as pl  
from IPython.display import display, HTML  
  
# Filter out missing or zero salary values  
pdf = df.filter(df["SALARY"] > 0).select("EMPLOYMENT\_TYPE\_NAME", "SALARY").toPandas()  
  
  
# Clean employment type names for better readability  
pdf["EMPLOYMENT\_TYPE\_NAME"] = (  
 pdf["EMPLOYMENT\_TYPE\_NAME"]  
 .astype(str)  
 .str.replace(r"[^\x00-\x7F]+", "", regex=True)  
)  
  
#display(HTML(f"<div style='height:300px; overflow:auto'>{pdf.iloc[:10].to\_html(index=False)}</div>"))  
  
# Compute median salary for sorting  
median\_salaries = pdf.groupby("EMPLOYMENT\_TYPE\_NAME")["SALARY"].median()  
display(median\_salaries.to\_frame().head())  
  
  
# Sort employment types based on median salary in descending order  
sorted\_employment\_types = median\_salaries.sort\_values(ascending=False).index  
  
# Apply sorted categories  
pdf["EMPLOYMENT\_TYPE\_NAME"] = pd.Categorical(  
 pdf["EMPLOYMENT\_TYPE\_NAME"],  
 categories=sorted\_employment\_types,  
 ordered=True  
)  
  
# Create box plot with horizontal grid lines  
fig = px.box(  
 pdf,  
 x="EMPLOYMENT\_TYPE\_NAME",  
 y="SALARY",  
 title="Salary Distribution by Employment Type",  
 color\_discrete\_sequence=["#CC0000"], # Single neutral color  
 boxmode="group",  
 points="all" # Show all outliers  
)  
fig  
  
  
  
# Improve layout, font styles, and axis labels  
fig.update\_layout(  
 title=dict(  
 text="Salary Distribution by Employment Type",  
 font=dict(size=16, family="Helvetica", color="black") # Bigger & Bold Title  
 ),  
 xaxis=dict(  
 title=dict(text="Employment Type", font=dict(size=14, family="Helvetica", color="black",weight="bold")), # Bigger X-label  
 tickangle=0, # Rotate X-axis labels for readability  
 tickfont=dict(size=12, family="Helvetica", color="black"), # Bigger & Bold X-ticks  
 showline=True, # Show axis lines  
 linewidth=2, # Thicker axis lines  
 linecolor="black",  
 mirror=True,  
 showgrid=False, # Remove vertical grid lines  
 categoryorder="array",  
 categoryarray=sorted\_employment\_types.tolist()  
 ),  
 yaxis=dict(  
 title=dict(text="Salary (in $1000)", font=dict(size=14, family="Helvetica", color="black",weight="bold")), # Bigger Y-label  
 tickvals=[0, 50000, 100000, 150000, 200000, 250000, 300000, 350000, 400000, 450000, 500000],  
 ticktext=["0", "50", "100", "150", "200", "250", "300", "350", "400", "450", "500"],  
 tickfont=dict(size=12, family="Helvetica", color="black"), # Bigger & Bold Y-ticks  
 showline=True,  
 linewidth=2,  
 linecolor="black",  
 mirror=True,  
 showgrid=True, # Enable light horizontal grid lines  
 gridcolor="lightgray", # Light shade for the horizontal grid  
 gridwidth=0.5 # Thin grid lines  
 ),  
 font=dict(family="Helvetica", size=12, color="black"),  
 boxgap=0.7,  
 plot\_bgcolor="white",  
 paper\_bgcolor="white",  
 showlegend=False,  
 height=500,  
 width=850  
)  
  
# Show & export  
fig.show()  
fig.write\_html("output/Q1.html")  
fig.write\_image("output/Q1.svg", width=850, height=500, scale=1)

[Stage 19:> (0 + 1) / 1]

|  | SALARY |
| --- | --- |
| EMPLOYMENT\_TYPE\_NAME |  |
| Full-time (> 32 hours) | 115024.0 |
| None | 115024.0 |
| Part-time ( 32 hours) | 115024.0 |
| Part-time / full-time | 115024.0 |

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

/tmp/ipykernel\_17899/1235387320.py:91: DeprecationWarning:  
  
  
Support for Kaleido versions less than 1.0.0 is deprecated and will be removed after September 2025.  
Please upgrade Kaleido to version 1.0.0 or greater (`pip install 'kaleido>=1.0.0'` or `pip install 'plotly[kaleido]'`).



# 3. Salary Distribution by Industry and Employment Type

* Compare salary variations across industries.
* **Filter the dataset**
  + Remove records where **Salary is missing or zero**.
* **Aggregate Data**
  + Group by **NAICS industry codes**.
  + Group by **employment type** and compute salary-distribution.
* **Visualize results**
  + Create a **box plot** where:
    - **X-axis** = NAICS2\_NAME.
    - **Y-axis** = SALARY\_FROM.
    - Group by EMPLOYMENT\_TYPE\_NAME.
  + Customize colors, fonts, and styles.
* **Explanation:** Write two sentences about what the graph reveals.

pdf = df.select("NAICS2\_NAME", "SALARY").toPandas()  
fig = px.box(pdf, x="NAICS2\_NAME", y="SALARY", title="Salary Distribution by Industry",  
 color\_discrete\_sequence=["#EF553B"])  
 # add nike template  
fig.update\_layout(template="nike")  
  
fig.update\_xaxes(tickangle=45)  
fig.show()

[Stage 20:> (0 + 1) / 1]

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

# 4. Salary Analysis by ONET Occupation Type (Bubble Chart)

* Analyze how salaries differ across ONET occupation types.
* **Aggregate Data**
  + Compute *median salary* for each occupation in the **ONET taxonomy**.
* **Visualize results**
  + Create a **bubble chart** where:
    - **X-axis** = ONET\_NAME
    - **Y-axis** = Median Salary
    - **Size** = Number of job postings
  + Apply custom colors and font styles.
* **Explanation:** Write two sentences about what the graph reveals.

pdf = df.select("ONET\_NAME", "SALARY", "ID").toPandas()  
  
df\_onet = pdf.groupby(["ONET\_NAME"]).agg({"SALARY": "median", "ID": "count"}).reset\_index()  
fig = px.scatter(  
 df\_onet,  
 x="ONET\_NAME",  
 y="SALARY",  
 size="ID",  
 title="Salary Analysis by ONET Occupation",  
 color="ONET\_NAME"  
)  
fig.update\_layout(font\_family="Arial", title\_font\_size=16)  
fig.show()

[Stage 21:> (0 + 1) / 1]

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

# 5. Salary by Education Level

* Create two groups:
  + **Bachelor’s or lower** (Bachelor’s, GED, Associate, No Education Listed)
  + **Master’s or PhD** (Master’s degree, Ph.D. or professional degree)
* Plot scatter plots for each group using MAX\_YEARS\_EXPERIENCE (with jitter), Average\_Salary, LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME.
* Then, plot histograms overlaid with KDE curves for each group.
  + This would generate two scatter plots and two histograms.
* **After each graph, add a short explanation** of key insights.

# 6. Salary by Remote Work Type

* Split into three groups based on REMOTE\_TYPE\_NAME:
  + Remote
  + Hybrid
  + Onsite (includes [None] and blank)
* Plot scatter plots for each group using MAX\_YEARS\_EXPERIENCE (with jitter), Average\_Salary, LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME.
* Also, create salary histograms for all three groups.
* **After each graph, briefly describe any patterns or comparisons.**

# Submission Instructions

* Submit the Word Document (part of git repo) containing:
  + The **HTTPS URL** of your GitHub repository.
  + Answer to the questions.
  + **Visualizations** created using matplotlib, Seaborn or plotly (preferred).
  + Answers to the questions below.

# Step 1: Spark SQL - Median salary and job count per TITLE\_NAME  
salary\_analysis = spark.sql("""  
 SELECT  
 LOT\_OCCUPATION\_NAME AS Occupation\_name,  
 PERCENTILE(SALARY, 0.5) AS Median\_Salary,  
 COUNT(\*) AS Job\_Postings  
 FROM job\_postings  
 GROUP BY LOT\_OCCUPATION\_NAME  
 ORDER BY Job\_Postings DESC  
 LIMIT 10  
""")  
  
# Step 2: Convert to Pandas DataFrame  
salary\_pd = salary\_analysis.toPandas()  
salary\_pd.head()  
  
  
  
# Step 3: Bubble chart using Plotly  
  
import plotly.express as px  
  
fig = px.scatter(  
 salary\_pd,  
 x="Occupation\_name",  
 y="Median\_Salary",  
 size="Job\_Postings",  
 title="Salary Analysis by LOT Occupation Type (Bubble Chart)",  
 labels={  
 "Occupation\_name": "LOT Occupation",  
 "Median\_Salary": "Median Salary",  
 "Job\_Postings": "Number of Job Postings"  
 },  
 hover\_name="Occupation\_name",  
 size\_max=60,  
 width=1000,  
 height=600,  
 color="Job\_Postings",  
 color\_continuous\_scale="Plasma"  
)  
  
  
  
# Step 4: Layout customization  
fig.update\_layout(  
 font\_family="Arial",  
 font\_size=14,  
 title\_font\_size=25,  
 xaxis\_title="LOT Occupation",  
 yaxis\_title="Median Salary",  
 plot\_bgcolor="white",  
 xaxis=dict(  
 tickangle=-45,  
 showline=True,  
 linecolor="black"  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="black"  
 )  
)  
  
# Step 5: Show and export  
fig.show()  
fig.write\_image("output/Q7.svg", width=1000, height=600, scale=1)

[Stage 22:> (0 + 1) / 1]

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

/tmp/ipykernel\_17899/2086897635.py:65: DeprecationWarning:  
  
  
Support for Kaleido versions less than 1.0.0 is deprecated and will be removed after September 2025.  
Please upgrade Kaleido to version 1.0.0 or greater (`pip install 'kaleido>=1.0.0'` or `pip install 'plotly[kaleido]'`).

# Defining education level groupings  
lower\_deg = ["Bachelor's", "Associate", "GED", "No Education Listed", "High school"]  
higher\_deg = ["Master's degree", "PhD or professional degree"]  
  
# Adding EDU\_GROUP column  
df = df.withColumn(  
 "EDU\_GROUP",  
 when(col("EDUCATION\_LEVELS\_NAME").rlike("|".join([f"(?i){deg}" for deg in lower\_deg])), "Bachelor's or lower")  
 .when(col("EDUCATION\_LEVELS\_NAME").rlike("|".join([f"(?i){deg}" for deg in higher\_deg])), "Master's or PhD")  
 .otherwise("Other")  
)  
  
# Casting necessary columns to float  
df = df.withColumn("MAX\_YEARS\_EXPERIENCE", col("MAX\_YEARS\_EXPERIENCE").cast("float"))  
df = df.withColumn("Average\_Salary", col("Average\_Salary").cast("float"))  
  
# Filtering for non-null and positive values  
df = df.filter(  
 col("MAX\_YEARS\_EXPERIENCE").isNotNull() &  
 col("Average\_Salary").isNotNull() &  
 (col("MAX\_YEARS\_EXPERIENCE") > 0) &  
 (col("Average\_Salary") > 0)  
)  
  
# Filtering for just the two education groups  
df\_filtered = df.filter(col("EDU\_GROUP").isin("Bachelor's or lower", "Master's or PhD"))  
  
# Converting to Pandas for plotting  
df\_pd = df\_filtered.toPandas()  
df\_pd.head()

[Stage 25:> (0 + 1) / 1]

|  | ID | LAST\_UPDATED\_DATE | LAST\_UPDATED\_TIMESTAMP | DUPLICATES | POSTED | EXPIRED | DURATION | SOURCE\_TYPES | SOURCES | URL | ... | 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 | Average\_Salary | EDU\_GROUP |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1f57d95acf4dc67ed2819eb12f049f6a5c11782c | 9/6/2024 | 2024-09-06 20:32:57.352 | 0 | 6/2/2024 | 6/8/2024 | 6.0 | [\n "Company"\n] | [\n "brassring.com"\n] | [\n "https://sjobs.brassring.com/TGnewUI/Sear... | ... | 441 | Motor Vehicle and Parts Dealers | 4413 | Automotive Parts, Accessories, and Tire Retailers | 44133 | Automotive Parts and Accessories Retailers | 441330 | Automotive Parts and Accessories Retailers | 108668.5 | Bachelor's or lower |
| 1 | 0cb072af26757b6c4ea9464472a50a443af681ac | 8/2/2024 | 2024-08-02 17:08:58.838 | 0 | 6/2/2024 | 8/1/2024 | NaN | [\n "Job Board"\n] | [\n "maine.gov"\n] | [\n "https://joblink.maine.gov/jobs/1085740"\n] | ... | 561 | Administrative and Support Services | 5613 | Employment Services | 56132 | Temporary Help Services | 561320 | Temporary Help Services | 108668.5 | Bachelor's or lower |
| 2 | 5a843df632e1ff756fa19d80a0871262d51becc0 | 6/21/2024 | 2024-06-21 07:00:00.000 | 0 | 6/2/2024 | 6/20/2024 | 18.0 | [\n "Job Board"\n] | [\n "computerwork.com"\n] | [\n "http://computerwork.com/us/en/search-job... | ... | 999 | Unclassified Industry | 9999 | Unclassified Industry | 99999 | Unclassified Industry | 999999 | Unclassified Industry | 108668.5 | Bachelor's or lower |
| 3 | 229620073766234e814e8add21db7dfaef69b3bd | 10/9/2024 | 2024-10-09 18:07:44.758 | 0 | 6/2/2024 | 8/1/2024 | NaN | [\n "Company"\n] | [\n "3ds.com"\n] | [\n "https://www.3ds.com/careers/jobs/sr-mark... | ... | 541 | Professional, Scientific, and Technical Services | 5415 | Computer Systems Design and Related Services | 54151 | Computer Systems Design and Related Services | 541511 | Custom Computer Programming Services | 92962.0 | Bachelor's or lower |
| 4 | 138ce2c9453b47a9b33403c364d4fd80996caa4f | 8/10/2024 | 2024-08-10 19:36:49.244 | 5 | 6/2/2024 | 8/9/2024 | NaN | [\n "Job Board",\n "Education",\n "Recruite... | [\n "silkroad.com",\n "hercjobs.org",\n "di... | [\n "https://main.hercjobs.org/jobs/20166141/... | ... | 611 | Educational Services | 6113 | Colleges, Universities, and Professional Schools | 61131 | Colleges, Universities, and Professional Schools | 611310 | Colleges, Universities, and Professional Schools | 108668.5 | Bachelor's or lower |