Assignment 03

Julio Vargas

September 21, 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)  
df.count()

+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
| 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

[Stage 77:> (0 + 1) / 1]

72498

# 1. Casting salary and experience columns (1)

## 1.1 Computing medians

from pyspark.sql.functions import col, regexp\_replace, trim  
# 1.1 Casting salary and experience columns  
  
  
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"))  
#df.select("SALARY", "SALARY\_FROM", "SALARY\_TO", "MIN\_YEARS\_EXPERIENCE", "MAX\_YEARS\_EXPERIENCE").printSchema()  
df.select("SALARY", "SALARY\_FROM", "SALARY\_TO", "MIN\_YEARS\_EXPERIENCE", "MAX\_YEARS\_EXPERIENCE").show(5)

+-------+-----------+---------+--------------------+--------------------+  
| SALARY|SALARY\_FROM|SALARY\_TO|MIN\_YEARS\_EXPERIENCE|MAX\_YEARS\_EXPERIENCE|  
+-------+-----------+---------+--------------------+--------------------+  
| NULL| NULL| NULL| 2.0| 2.0|  
| NULL| NULL| NULL| 3.0| 3.0|  
| NULL| NULL| NULL| 5.0| NULL|  
| NULL| NULL| NULL| 3.0| NULL|  
|92500.0| 35000.0| 150000.0| NULL| NULL|  
+-------+-----------+---------+--------------------+--------------------+  
only showing top 5 rows

## 1.2 Computing medians

# 1.2 Computing medians  
def compute\_median(sdf, col\_name):  
 q = sdf.approxQuantile(col\_name, [0.5], 0.01) #50 percentile 1% error  
 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")

[Stage 81:> (0 + 1) / 1] [Stage 82:> (0 + 1) / 1] [Stage 83:> (0 + 1) / 1]

# 1.2 Output  
#the median\_from, median\_to , median\_salary respectively are:  
  
print("- Median SALARY\_FROM: $" + str(median\_from))  
print("- Median SALARY\_TO: $" + str(median\_to))  
print("- Median SALARY: $" + str(median\_salary))

- Median SALARY\_FROM: $87295.0  
- Median SALARY\_TO: $130042.0  
- Median SALARY: $115024.0

## 1.3 Imputing missing salaries

# 1.3 Imputing missing salaries  
df = df.fillna({  
 "SALARY\_FROM": median\_from,  
 "SALARY\_TO": median\_to,  
 "SALARY": median\_salary  
})  
  
# 1.3 Add new column Average\_Salary  
df = df.withColumn("Average\_Salary", (col("SALARY\_FROM") + col("SALARY\_TO")) / 2)  
  
export\_cols = ["Average\_Salary","SALARY","EDUCATION\_LEVELS\_NAME","REMOTE\_TYPE\_NAME",  
 "MAX\_YEARS\_EXPERIENCE","LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"]  
  
# 1.3 Output  
df.select(\*export\_cols).show(5, truncate=False)

+--------------+--------+-----------------------------+----------------+--------------------+----------------------------------+  
|Average\_Salary|SALARY |EDUCATION\_LEVELS\_NAME |REMOTE\_TYPE\_NAME|MAX\_YEARS\_EXPERIENCE|LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME|  
+--------------+--------+-----------------------------+----------------+--------------------+----------------------------------+  
|108668.5 |115024.0|[\n "Bachelor's degree"\n] |[None] |2.0 |General ERP Analyst / Consultant |  
|108668.5 |115024.0|[\n "No Education Listed"\n]|Remote |3.0 |Oracle Consultant / Analyst |  
|108668.5 |115024.0|[\n "Bachelor's degree"\n] |[None] |NULL |Data Analyst |  
|108668.5 |115024.0|[\n "No Education Listed"\n]|[None] |NULL |Data Analyst |  
|92500.0 |92500.0 |[\n "No Education Listed"\n]|[None] |NULL |Oracle Consultant / Analyst |  
+--------------+--------+-----------------------------+----------------+--------------------+----------------------------------+  
only showing top 5 rows

## 1.4 Cleaning Education column

#1.4 Cleaning Education column  
#remove the \n and \r  
df1 = df.withColumn("EDUCATION\_LEVELS\_NAME",  
 trim(  
 regexp\_replace(regexp\_replace(col("EDUCATION\_LEVELS\_NAME"),r"\n|\r", ""), #remove \n and \r  
 r"\[\s+\"", "[\"" ) #remove spaces.  
 )  
)  
# 1.4 Output  
df1.select(\*export\_cols).show(5, truncate=False)

+--------------+--------+-----------------------+----------------+--------------------+----------------------------------+  
|Average\_Salary|SALARY |EDUCATION\_LEVELS\_NAME |REMOTE\_TYPE\_NAME|MAX\_YEARS\_EXPERIENCE|LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME|  
+--------------+--------+-----------------------+----------------+--------------------+----------------------------------+  
|108668.5 |115024.0|["Bachelor's degree"] |[None] |2.0 |General ERP Analyst / Consultant |  
|108668.5 |115024.0|["No Education Listed"]|Remote |3.0 |Oracle Consultant / Analyst |  
|108668.5 |115024.0|["Bachelor's degree"] |[None] |NULL |Data Analyst |  
|108668.5 |115024.0|["No Education Listed"]|[None] |NULL |Data Analyst |  
|92500.0 |92500.0 |["No Education Listed"]|[None] |NULL |Oracle Consultant / Analyst |  
+--------------+--------+-----------------------+----------------+--------------------+----------------------------------+  
only showing top 5 rows

## 1.5 Exporting Cleaned Data

#1.5 Exporting Cleaned Data  
# Export to CSV  
df\_selected=df1.select(\*export\_cols)  
pdf = df\_selected.toPandas()  
pdf.to\_csv("data/lightcast\_cleaned.csv", index=False)  
  
print("Data cleaning complete. Rows retained:", len(pdf))

[Stage 86:> (0 + 1) / 1]

Data cleaning complete. Rows retained: 72498

# 2. Salary Distribution by Industry and Employment Type (2)

## 2.1 TEMPLATE

import plotly.graph\_objects as go  
import plotly.io as pio  
  
pio.templates["nike"] = go.layout.Template(  
 # LAYOUT  
 layout = {  
 # Fonts and colors  
 'title': {  
 'font': {'family': 'HelveticaNeue-CondensedBold, Helvetica, Sans-serif',  
 'size': 30,  
 'color': '#13007c'}   
 },  
 'font': {'family': 'Helvetica Neue, Helvetica, Sans-serif',  
 'size': 16,  
 'color': '#3b3b3b'},   
  
 'colorway': ['#fffb00', '#e010fc'],   
 # Adding others  
 'hovermode': 'x unified',  
 'plot\_bgcolor': '#E5ECF6',  
 'paper\_bgcolor': "#FFFFFF",  
   
 },  
 # DATA  
 data = {  
 # Default style applied to all bar charts  
 'bar': [go.Bar(  
 texttemplate = '%{value:$.2s}',  
 textposition = 'outside',  
 textfont = {'family': 'Helvetica Neue, Helvetica, Sans-serif',  
 'size': 20,  
 'color': '#ff6874'} # FFFFFF  
 )]  
 }  
)

## 2.2 Development of Question 2

#your code for first query  
import pandas as pd  
import polars as pl  
from IPython.display import display, HTML  
  
#2.2 Filter the dataset - Remove records where salary is missing or zero.  
df\_valid\_salaries = df.filter(df["SALARY"] > 0).select("NAICS2\_NAME","EMPLOYMENT\_TYPE\_NAME", "SALARY")  
  
#2.2 output - convert to pandas  
pdf = df\_valid\_salaries.toPandas()  
print("Data cleaning complete. Rows retained:", len(pdf))  
  
#2.3 Aggregate data - NAICS industry codes, employment type and compute salary distribution.  
  
# Clean employment type names for better readability  
pdf["EMPLOYMENT\_TYPE\_NAME"] = (pdf["EMPLOYMENT\_TYPE\_NAME"].astype(str)  
 .str.replace(r"[^\x00-\x7F]+", "", regex=True)  
)  
  
#2.3 output  
median\_salaries\_naics = pdf.groupby("NAICS2\_NAME")["SALARY"].median()  
median\_salaries\_employee = pdf.groupby("EMPLOYMENT\_TYPE\_NAME")["SALARY"].median()  
display(median\_salaries\_naics.to\_frame().head())  
display(median\_salaries\_employee.to\_frame().head())

[Stage 87:> (0 + 1) / 1]

Data cleaning complete. Rows retained: 72498

|  | SALARY |
| --- | --- |
| NAICS2\_NAME |  |
| Accommodation and Food Services | 115024.0 |
| Administrative and Support and Waste Management and Remediation Services | 115024.0 |
| Agriculture, Forestry, Fishing and Hunting | 115024.0 |
| Arts, Entertainment, and Recreation | 115024.0 |
| Construction | 115024.0 |

|  | 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 |

#2.4 Visualize results box plot   
# X-axis = NAICS2\_NAME || Y-axis = SALARY\_FROM || Group by EMPLOYMENT\_TYPE\_NAME.  
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.update\_layout(  
 template="nike",  
 height=700,  
 xaxis=dict(  
 title=dict(text="NAICS2\_NAME", standoff=40),   
 tickangle=45,  
 tickfont=dict(size=10),  
 automargin=True  
 ),  
 yaxis=dict(title=dict(text="Salary")),  
 margin=dict(b=150)   
   
)  
  
  
fig.show()  
fig.write\_image("output/Q2.svg", width=1000, height=600, scale=1)

[Stage 88:> (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

## 2.3 Explanation of Box Plot 2

This box plot gives a quick view of how salaries vary by industry, each box is where most of the pay sits, and the line inside is the median. Those dots floating above are the really high-paying jobs. For example in the Information industry, most salaries bunch around 110 k, but you can also see some offers way higher than that, those are the outliers and this job is among the top salaries at about 500k.

# 3. Salary Analysis by ONET Occupation Type (Bubble Chart) (3)

## 3.1 Development of Question 3

import plotly.express as px  
#3.1 Analyze how salaries differ across LOT\_OCCUPATION\_NAME occupation types.  
#ONET\_NAME CHANGE TO LOT\_OCCUPATION\_NAME  
  
#Aggregate Data  
  
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  
 WHERE LOT\_OCCUPATION\_NAME IS NOT NULL  
 GROUP BY LOT\_OCCUPATION\_NAME  
 ORDER BY Job\_Postings DESC  
 LIMIT 10  
""") #the result only has 6 results and a null, limit to 10 is not necessary  
  
salary\_pd = salary\_analysis.toPandas()  
display(salary\_pd.head())  
  
#Simple plot to Analyze  
figa = px.scatter(  
 salary\_pd,  
 x="Occupation\_name",  
 y="Median\_Salary",  
 size="Job\_Postings",  
 title="Salary Analysis by Occupation",  
 color="Occupation\_name"  
)  
figa.update\_xaxes(tickangle=45, automargin=True)  
figa.show()  
  
#3.2 Visualize results bubble chart  
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=900,  
 height=600,  
 color="Job\_Postings",  
 color\_continuous\_scale="Plasma"  
)  
#customize layout  
fig.update\_layout(  
   
 height=700,  
 font\_family="Arial",  
 font\_size=14,  
 title\_font\_size=25,  
 title\_font\_color="#13007c",  
 font\_color="#2e2e2e",   
 xaxis\_title="LOT Occupation",  
 yaxis\_title="Median Salary",  
 plot\_bgcolor="#FAFDFF",  
 xaxis=dict(  
 tickangle=-45,  
 showline=True,  
 linecolor="#444"  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="black"  
 )  
)  
  
  
fig.show()  
fig.write\_image("output/Q3.svg", width=1000, height=600, scale=1)

[Stage 89:> (0 + 1) / 1]

|  | Occupation\_name | Median\_Salary | Job\_Postings |
| --- | --- | --- | --- |
| 0 | Data / Data Mining Analyst | 95250.0 | 30057 |
| 1 | Business Intelligence Analyst | 125900.0 | 29445 |
| 2 | Computer Systems Engineer / Architect | 157600.0 | 8212 |
| 3 | Business / Management Analyst | 93650.0 | 4326 |
| 4 | Clinical Analyst / Clinical Documentation and ... | 89440.0 | 261 |

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

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

## 3.2 Explanation of Box Plot 3

This shows how salaries stack up by occupation. Each bubble is a job type, higher on the chart means a bigger median salary, we can see bigger bubbles which mean more job postings, and the color shows volume too. For example, Business Intelligence Analyst pay well and have tons of postings, while Market Research Analyst roles are smaller and pay less.

# 4. Salary by Education Level (4)

## 4.1 Development of Question 4

# 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 new column EDU\_GROUP   
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")  
)  
  
# Modyfying/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"))  
  
# df.select("MAX\_YEARS\_EXPERIENCE","Average\_Salary","EDU\_GROUP","EDUCATION\_LEVELS\_NAME").printSchema() #check schema changes  
  
# print(df.count()) #Total 72,498 after 8074  
  
# 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 EDU\_GROUP 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()  
pdf4=df.select("MAX\_YEARS\_EXPERIENCE","Average\_Salary","EDU\_GROUP","EDUCATION\_LEVELS\_NAME").toPandas()  
display(pdf4.head())

[Stage 92:> (0 + 1) / 1] [Stage 93:> (0 + 1) / 1]

|  | MAX\_YEARS\_EXPERIENCE | Average\_Salary | EDU\_GROUP | EDUCATION\_LEVELS\_NAME |
| --- | --- | --- | --- | --- |
| 0 | 2.0 | 108668.5 | Bachelor's or lower | [\n "Bachelor's degree"\n] |
| 1 | 3.0 | 108668.5 | Bachelor's or lower | [\n "No Education Listed"\n] |
| 2 | 7.0 | 108668.5 | Bachelor's or lower | [\n "No Education Listed"\n] |
| 3 | 2.0 | 92962.0 | Bachelor's or lower | [\n "Bachelor's degree",\n "Master's degree"\n] |
| 4 | 5.0 | 108668.5 | Bachelor's or lower | [\n "Associate degree",\n "Bachelor's degree... |

# Jittering / trimming  
df\_pd["MAX\_EXPERIENCE\_JITTER"] = df\_pd["MAX\_YEARS\_EXPERIENCE"] + np.random.uniform(-0.25, 0.25, size=len(df\_pd))  
df\_pd["AVERAGE\_SALARY\_JITTER"] = df\_pd["Average\_Salary"] + np.random.uniform(-2500, 2500, size=len(df\_pd))  
df\_pd = df\_pd.round(2)  
  
# Remove outlier higher than 399K  
df\_pd = df\_pd[df\_pd["AVERAGE\_SALARY\_JITTER"] <= 399000]  
  
df\_pd.head()

|  | ID | LAST\_UPDATED\_DATE | LAST\_UPDATED\_TIMESTAMP | DUPLICATES | POSTED | EXPIRED | DURATION | SOURCE\_TYPES | SOURCES | URL | ... | 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 | MAX\_EXPERIENCE\_JITTER | AVERAGE\_SALARY\_JITTER |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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... | ... | 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 | 2.07 | 110819.55 |
| 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] | ... | 5613 | Employment Services | 56132 | Temporary Help Services | 561320 | Temporary Help Services | 108668.5 | Bachelor's or lower | 2.94 | 108300.48 |
| 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... | ... | 9999 | Unclassified Industry | 99999 | Unclassified Industry | 999999 | Unclassified Industry | 108668.5 | Bachelor's or lower | 7.08 | 109648.17 |
| 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... | ... | 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 | 1.83 | 91939.49 |
| 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/... | ... | 6113 | Colleges, Universities, and Professional Schools | 61131 | Colleges, Universities, and Professional Schools | 611310 | Colleges, Universities, and Professional Schools | 108668.5 | Bachelor's or lower | 5.23 | 106881.94 |

#jittering and triming  
# Plot four groups  
fig1 = px.scatter(  
 df\_pd,  
 x="MAX\_EXPERIENCE\_JITTER",  
 y="AVERAGE\_SALARY\_JITTER",  
 color="EDU\_GROUP",  
 hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
 title="<b>Experience vs Salary by Education Level</b>",  
 opacity=1.0, #0.7  
 color\_discrete\_sequence=[  
 "#636EFA", # Blue  
 "#EF553B", # Red  
 "#00CC96", # Green  
 "#AB63FA" # Purple  
 ]  
)  
  
fig1.update\_traces(  
 marker=dict(size=10, line=dict(width=1, color="black"))  
)  
  
fig1.update\_layout(  
 plot\_bgcolor="#fcfcf0",  
 paper\_bgcolor="#f5d9b2",  
 font=dict(family="Segoe UI", size=14, color="#2b2b2b"),  
 title\_font=dict(size=22, color="#4b3832"),  
 xaxis\_title="Years of Experience",  
 yaxis\_title="Average Salary (USD)",  
 legend\_title="Education Group",  
 hoverlabel=dict(bgcolor="white", font\_size=13, font\_family="Arial"),  
 margin=dict(t=70, b=60, l=60, r=60),  
 xaxis=dict(  
 gridcolor="#e0e0e0",  
 tickmode="linear",  
 dtick=1  
 ),  
 yaxis=dict(gridcolor="#cccccc")  
)  
  
fig1.show()  
fig.write\_image("output/Q4.svg", width=1000, height=600, scale=1)

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

## 4.2 Explanation of Box Plot 4

This chart shows how pay changes with experience for two education levels. Blue dots are people with a bachelor s or lower, red dots are master’s or PhD. We can see most salaries cluster under 200k no matter the experience, but a few outliers pop way higher.

# 5. Salary by Remote Work Type (5)

## 5.1 Development of Question 5

from pyspark.sql.functions import when, col, trim  
  
#5.1 Split into three groups based on REMOTE\_TYPE\_NAME  
df = df.withColumn(  
 "REMOTE\_GROUP",  
 when(trim(col("REMOTE\_TYPE\_NAME")) == "Remote", "Remote")  
 .when(trim(col("REMOTE\_TYPE\_NAME")) == "Hybrid Remote", "Hybrid")  
 .when(trim(col("REMOTE\_TYPE\_NAME")) == "Not Remote", "Onsite")  
 .when(col("REMOTE\_TYPE\_NAME").isNull(), "Onsite")  
 .otherwise("Onsite")  
)  
  
#print(df.count())  
  
#5.1 Filter valid values  
df = df.filter(  
 col("MAX\_YEARS\_EXPERIENCE").isNotNull() & col("Average\_Salary").isNotNull() &  
 (col("MAX\_YEARS\_EXPERIENCE") > 0) & (col("Average\_Salary") > 0)  
)  
  
#5.1 Pandas  
df\_pd = df.select(  
 "MAX\_YEARS\_EXPERIENCE","Average\_Salary","LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME","REMOTE\_GROUP"  
 ).toPandas()  
  
df\_pd.head()  
  
# Jittering / trimming  
df\_pd["MAX\_EXPERIENCE\_JITTER"] = df\_pd["MAX\_YEARS\_EXPERIENCE"] + np.random.uniform(-0.15, 0.15, size=len(df\_pd))  
df\_pd["AVERAGE\_SALARY\_JITTER"] = df\_pd["Average\_Salary"] + np.random.uniform(-1000, 1000, size=len(df\_pd))  
df\_pd = df\_pd.round(2)  
  
# Remove outlier higher than 399K  
df\_pd = df\_pd[df\_pd["AVERAGE\_SALARY\_JITTER"] <= 399000]

[Stage 94:> (0 + 1) / 1]

# Plot four groups  
fig5 = px.scatter(  
 df\_pd,  
 x="MAX\_EXPERIENCE\_JITTER",  
 y="AVERAGE\_SALARY\_JITTER",  
 color="REMOTE\_GROUP",  
 hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
 title="<b>Experience vs Salary by Remote Work Type </b>",  
 opacity=1.0, #0.7  
 color\_discrete\_sequence=[  
 "#636EFA", # Blue  
 "#EF553B", # Red  
 "#00CC96", # Green  
 "#AB63FA" # Purple  
 ]  
)  
  
fig5.update\_traces(  
 marker=dict(size=10, line=dict(width=1, color="black"))  
)  
  
fig5.update\_layout(  
 plot\_bgcolor="#fcfcf0",   
 paper\_bgcolor="#f5d9b2",   
 font=dict(family="Segoe UI", size=14, color="#2b2b2b"),  
 title\_font=dict(size=22, color="#4b3832"),  
 xaxis\_title="Years of Experience",  
 yaxis\_title="Average Salary (USD)",  
 legend\_title="Education Group",  
 hoverlabel=dict(bgcolor="white", font\_size=13, font\_family="Arial"),  
 margin=dict(t=70, b=60, l=60, r=60),  
 xaxis=dict(  
 gridcolor="#e0e0e0",  
 tickmode="linear",  
 dtick=1   
 ),  
 yaxis=dict(gridcolor="#cccccc")  
)  
  
fig5.show()  
fig.write\_image("output/Q5.svg", width=1000, height=600, scale=1)

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

## 5.2 Explanation of Box Plot 5

This chart shows pay versus experience split by how people work — blue is onsite, red is remote, green is hybrid. Most salaries bunch under 200k no matter the setup, but you can spot a few high outliers across all three.