Assignment 03

Makenzie Howard

September 22, 2025

# Load the Dataset

import pandas as pd  
import plotly.express as px  
import plotly.io as pio  
from pyspark.sql import SparkSession  
import re  
import numpy as np  
import plotly.graph\_objects as go  
from pyspark.sql.functions import col, split, explode, regexp\_replace, transform, when  
from pyspark.sql import functions as F  
from pyspark.sql.functions import col, monotonically\_increasing\_id  
  
np.random.seed(42)  
  
pio.renderers.default = "notebook"  
  
# Initialize Spark Session  
spark = SparkSession.builder.appName("LightcastData").getOrCreate()  
  
# Load Data  
df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","true").option("escape", "\"").csv("lightcast\_job\_postings.csv")  
df.createOrReplaceTempView("job\_postings")  
  
# Show Schema and Sample Data  
print("---This is Diagnostic check, No need to print it in the final doc---")  
  
df.printSchema() # comment this line when rendering the submission  
df.show(5)

[Stage 31:> (0 + 1) / 1]

---This is Diagnostic check, No need to print it in the final doc---  
root  
 |-- ID: string (nullable = true)  
 |-- LAST\_UPDATED\_DATE: string (nullable = true)  
 |-- LAST\_UPDATED\_TIMESTAMP: timestamp (nullable = true)  
 |-- DUPLICATES: integer (nullable = true)  
 |-- POSTED: string (nullable = true)  
 |-- EXPIRED: string (nullable = true)  
 |-- DURATION: integer (nullable = true)  
 |-- SOURCE\_TYPES: string (nullable = true)  
 |-- SOURCES: string (nullable = true)  
 |-- URL: string (nullable = true)  
 |-- ACTIVE\_URLS: string (nullable = true)  
 |-- ACTIVE\_SOURCES\_INFO: string (nullable = true)  
 |-- TITLE\_RAW: string (nullable = true)  
 |-- BODY: string (nullable = true)  
 |-- MODELED\_EXPIRED: string (nullable = true)  
 |-- MODELED\_DURATION: integer (nullable = true)  
 |-- COMPANY: integer (nullable = true)  
 |-- COMPANY\_NAME: string (nullable = true)  
 |-- COMPANY\_RAW: string (nullable = true)  
 |-- COMPANY\_IS\_STAFFING: boolean (nullable = true)  
 |-- EDUCATION\_LEVELS: string (nullable = true)  
 |-- EDUCATION\_LEVELS\_NAME: string (nullable = true)  
 |-- MIN\_EDULEVELS: integer (nullable = true)  
 |-- MIN\_EDULEVELS\_NAME: string (nullable = true)  
 |-- MAX\_EDULEVELS: integer (nullable = true)  
 |-- MAX\_EDULEVELS\_NAME: string (nullable = true)  
 |-- EMPLOYMENT\_TYPE: integer (nullable = true)  
 |-- EMPLOYMENT\_TYPE\_NAME: string (nullable = true)  
 |-- MIN\_YEARS\_EXPERIENCE: integer (nullable = true)  
 |-- MAX\_YEARS\_EXPERIENCE: integer (nullable = true)  
 |-- IS\_INTERNSHIP: boolean (nullable = true)  
 |-- SALARY: integer (nullable = true)  
 |-- REMOTE\_TYPE: integer (nullable = true)  
 |-- REMOTE\_TYPE\_NAME: string (nullable = true)  
 |-- ORIGINAL\_PAY\_PERIOD: string (nullable = true)  
 |-- SALARY\_TO: integer (nullable = true)  
 |-- SALARY\_FROM: integer (nullable = true)  
 |-- LOCATION: string (nullable = true)  
 |-- CITY: string (nullable = true)  
 |-- CITY\_NAME: string (nullable = true)  
 |-- COUNTY: integer (nullable = true)  
 |-- COUNTY\_NAME: string (nullable = true)  
 |-- MSA: integer (nullable = true)  
 |-- MSA\_NAME: string (nullable = true)  
 |-- STATE: integer (nullable = true)  
 |-- STATE\_NAME: string (nullable = true)  
 |-- COUNTY\_OUTGOING: integer (nullable = true)  
 |-- COUNTY\_NAME\_OUTGOING: string (nullable = true)  
 |-- COUNTY\_INCOMING: integer (nullable = true)  
 |-- COUNTY\_NAME\_INCOMING: string (nullable = true)  
 |-- MSA\_OUTGOING: integer (nullable = true)  
 |-- MSA\_NAME\_OUTGOING: string (nullable = true)  
 |-- MSA\_INCOMING: integer (nullable = true)  
 |-- MSA\_NAME\_INCOMING: string (nullable = true)  
 |-- NAICS2: integer (nullable = true)  
 |-- NAICS2\_NAME: string (nullable = true)  
 |-- NAICS3: integer (nullable = true)  
 |-- NAICS3\_NAME: string (nullable = true)  
 |-- NAICS4: integer (nullable = true)  
 |-- NAICS4\_NAME: string (nullable = true)  
 |-- NAICS5: integer (nullable = true)  
 |-- NAICS5\_NAME: string (nullable = true)  
 |-- NAICS6: integer (nullable = true)  
 |-- NAICS6\_NAME: string (nullable = true)  
 |-- TITLE: string (nullable = true)  
 |-- TITLE\_NAME: string (nullable = true)  
 |-- TITLE\_CLEAN: string (nullable = true)  
 |-- SKILLS: string (nullable = true)  
 |-- SKILLS\_NAME: string (nullable = true)  
 |-- SPECIALIZED\_SKILLS: string (nullable = true)  
 |-- SPECIALIZED\_SKILLS\_NAME: string (nullable = true)  
 |-- CERTIFICATIONS: string (nullable = true)  
 |-- CERTIFICATIONS\_NAME: string (nullable = true)  
 |-- COMMON\_SKILLS: string (nullable = true)  
 |-- COMMON\_SKILLS\_NAME: string (nullable = true)  
 |-- SOFTWARE\_SKILLS: string (nullable = true)  
 |-- SOFTWARE\_SKILLS\_NAME: string (nullable = true)  
 |-- ONET: string (nullable = true)  
 |-- ONET\_NAME: string (nullable = true)  
 |-- ONET\_2019: string (nullable = true)  
 |-- ONET\_2019\_NAME: string (nullable = true)  
 |-- CIP6: string (nullable = true)  
 |-- CIP6\_NAME: string (nullable = true)  
 |-- CIP4: string (nullable = true)  
 |-- CIP4\_NAME: string (nullable = true)  
 |-- CIP2: string (nullable = true)  
 |-- CIP2\_NAME: string (nullable = true)  
 |-- SOC\_2021\_2: string (nullable = true)  
 |-- SOC\_2021\_2\_NAME: string (nullable = true)  
 |-- SOC\_2021\_3: string (nullable = true)  
 |-- SOC\_2021\_3\_NAME: string (nullable = true)  
 |-- SOC\_2021\_4: string (nullable = true)  
 |-- SOC\_2021\_4\_NAME: string (nullable = true)  
 |-- SOC\_2021\_5: string (nullable = true)  
 |-- SOC\_2021\_5\_NAME: string (nullable = true)  
 |-- LOT\_CAREER\_AREA: integer (nullable = true)  
 |-- LOT\_CAREER\_AREA\_NAME: string (nullable = true)  
 |-- LOT\_OCCUPATION: integer (nullable = true)  
 |-- LOT\_OCCUPATION\_NAME: string (nullable = true)  
 |-- LOT\_SPECIALIZED\_OCCUPATION: integer (nullable = true)  
 |-- LOT\_SPECIALIZED\_OCCUPATION\_NAME: string (nullable = true)  
 |-- LOT\_OCCUPATION\_GROUP: integer (nullable = true)  
 |-- LOT\_OCCUPATION\_GROUP\_NAME: string (nullable = true)  
 |-- LOT\_V6\_SPECIALIZED\_OCCUPATION: integer (nullable = true)  
 |-- LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME: string (nullable = true)  
 |-- LOT\_V6\_OCCUPATION: integer (nullable = true)  
 |-- LOT\_V6\_OCCUPATION\_NAME: string (nullable = true)  
 |-- LOT\_V6\_OCCUPATION\_GROUP: integer (nullable = true)  
 |-- LOT\_V6\_OCCUPATION\_GROUP\_NAME: string (nullable = true)  
 |-- LOT\_V6\_CAREER\_AREA: integer (nullable = true)  
 |-- LOT\_V6\_CAREER\_AREA\_NAME: string (nullable = true)  
 |-- SOC\_2: string (nullable = true)  
 |-- SOC\_2\_NAME: string (nullable = true)  
 |-- SOC\_3: string (nullable = true)  
 |-- SOC\_3\_NAME: string (nullable = true)  
 |-- SOC\_4: string (nullable = true)  
 |-- SOC\_4\_NAME: string (nullable = true)  
 |-- SOC\_5: string (nullable = true)  
 |-- SOC\_5\_NAME: string (nullable = true)  
 |-- LIGHTCAST\_SECTORS: string (nullable = true)  
 |-- LIGHTCAST\_SECTORS\_NAME: string (nullable = true)  
 |-- NAICS\_2022\_2: integer (nullable = true)  
 |-- NAICS\_2022\_2\_NAME: string (nullable = true)  
 |-- NAICS\_2022\_3: integer (nullable = true)  
 |-- NAICS\_2022\_3\_NAME: string (nullable = true)  
 |-- NAICS\_2022\_4: integer (nullable = true)  
 |-- NAICS\_2022\_4\_NAME: string (nullable = true)  
 |-- NAICS\_2022\_5: integer (nullable = true)  
 |-- NAICS\_2022\_5\_NAME: string (nullable = true)  
 |-- NAICS\_2022\_6: integer (nullable = true)  
 |-- NAICS\_2022\_6\_NAME: string (nullable = true)  
  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
| 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

# 1. Data Preparation

# Step 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"))  
)  
  
# 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, median\_salary)  
  
# Step 3: Imputing missing salaries, but not experience  
df = df.fillna({  
 "SALARY\_FROM" : median\_from,  
 "SALARY\_TO" : median\_to,  
 "SALARY" : median\_salary  
})  
df\_filtered = df.filter(  
 (F.col("SALARY") > 0) &  
 F.col("EMPLOYMENT\_TYPE\_NAME").isNotNull() &  
 (F.trim(F.col("EMPLOYMENT\_TYPE\_NAME")) != F.lit("")) &  
 (F.lower(F.trim(F.col("EMPLOYMENT\_TYPE\_NAME"))) != F.lit("none"))  
)  
  
pdf = (  
 df\_filtered  
 .select("EMPLOYMENT\_TYPE\_NAME", "SALARY")  
 .toPandas()  
)  
  
  
# 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",  
 "median\_salary"  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME",  
]  
  
df\_selected = df.select(\*[c for c in export\_cols if c in df.columns])  
  
# Step 7: Saving to CSV  
pdf = df\_selected.toPandas() # OK for small/medium data  
pdf.to\_csv("./data/lightcast\_cleaned.csv", index=False)  
  
print("Data Cleaning Complete. Rows retained:", len(pdf))

[Stage 33:> (0 + 1) / 1] [Stage 34:> (0 + 1) / 1] [Stage 35:> (0 + 1) / 1]

Medians: 87295.0 130042.0 115024.0

[Stage 36:> (0 + 1) / 1] [Stage 37:> (0 + 1) / 1]

Data Cleaning Complete. Rows retained: 72498

# 2. Salary Distribution by Employment Type

# Salary Distribution by Industry and Employment Type  
  
import os, re  
import pandas as pd  
import plotly.express as px  
from IPython.display import display  
  
# Ensure output folder exists  
os.makedirs("output", exist\_ok=True)  
  
# 1) Filter out missing or zero salary values and bring to pandas  
pdf = (  
 df.filter(df["SALARY"] > 0) # PySpark filter (assumes SALARY is numeric)  
 .select("EMPLOYMENT\_TYPE\_NAME", "SALARY")  
 .toPandas()  
)  
  
# 2) Clean employment type names   
pdf["EMPLOYMENT\_TYPE\_NAME"] = (  
 pdf["EMPLOYMENT\_TYPE\_NAME"]  
 .astype(str)  
 .apply(lambda x: re.sub(r"[^\x00-\x7F]+", "", x))  
)  
  
# Make sure salary is numeric   
pdf["SALARY"] = pd.to\_numeric(pdf["SALARY"], errors="coerce")  
  
# If a stray pluralized column exists from earlier runs  
if "EMPLOYMENT\_TYPE\_NAMES" in pdf.columns:  
 pdf = pdf.drop(columns=["EMPLOYMENT\_TYPE\_NAMES"])  
  
# 3) Compute median salary for sorting  
median\_salaries = pdf.groupby("EMPLOYMENT\_TYPE\_NAME", dropna=False)["SALARY"].median()  
  
# 4) Sort employment types based on median salary (descending)  
sorted\_employment\_types = median\_salaries.sort\_values(ascending=False).index  
  
# 5) Apply sorted categories  
pdf["EMPLOYMENT\_TYPE\_NAME"] = pd.Categorical(  
 pdf["EMPLOYMENT\_TYPE\_NAME"],  
 categories=sorted\_employment\_types,  
 ordered=True  
)  
  
# 6) Create box plot   
fig = px.box(  
 pdf,  
 x="EMPLOYMENT\_TYPE\_NAME",  
 y="SALARY",  
 title="Salary Distribution by Employment Type",  
 points="all",  
 boxmode="group",  
   
)  
# Force black for markers and box lines  
fig.update\_traces(marker\_color="red", line\_color="black")  
  
# 7) Improve layout, font styles, and axis labels  
fig.update\_layout(  
 title=dict(text="Salary Distribution by Employment Type",  
 font=dict(size=30, family="Helvetica", color="black")),  
 xaxis=dict(  
 showline=True, linewidth=2, linecolor="black", mirror=True, showgrid=False,  
 categoryorder="array", categoryarray=list(sorted\_employment\_types)  
 ),  
 yaxis=dict(  
 title=dict(text="Salary (K $)", font=dict(size=24, family="Helvetica", color="black")),  
 tickvals=[0, 50000, 100000, 150000, 200000, 250000, 300000, 350000, 400000, 450000, 500000],  
 ticktext=["0","50K","100K","150K","200K","250K","300K","350K","400K","450K","500K"],  
 tickfont=dict(size=18, family="Helvetica", color="black"),  
 showline=True, linewidth=2, linecolor="black", mirror=True,  
 showgrid=True, gridcolor="lightgray", gridwidth=0.5  
 ),  
 font=dict(family="Helvetica", size=16, color="black"),  
 boxgap=0.7, boxmode="group",  
 plot\_bgcolor="white", paper\_bgcolor="white",  
 showlegend=False, height=500, width=850,  
)  
  
display(fig)  
  
# Save SVG   
try:  
 # pip install kaleido  
 fig.write\_image("output/Q1.svg", width=850, height=500, scale=1)  
except Exception:  
 fig.write\_html("output/Q1.html", include\_plotlyjs="cdn")

[Stage 38:> (0 + 1) / 1]

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

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

***Explanation*** This table represents salary distributions by employment type: full time, part-time, or a hybrid of both. According to the table, full-time employees make the highest salaries with part-time employees making the least. Due to its hybrid nature, full-time/part-time employees experience the broadest range of salaries.

# 3. Salary Distribution by Industry

# Show table + interactive box chart  
  
import plotly.express as px  
from IPython.display import display, HTML  
import pandas as pd, os  
  
# 1) Data to pandas  
pdf = df.select("NAICS2\_NAME", "SALARY").toPandas()  
  
# 1a) Force a visible HTML table preview  
display(HTML(pdf.head(15).to\_html(index=False)))  
  
# 2) Plotly box chart  
fig = px.box(  
 pdf,  
 x="NAICS2\_NAME",  
 y="SALARY",  
 title="Salary Distribution by Industry",  
 color\_discrete\_sequence=["red"]  
)  
fig.update\_layout(font\_family="Helvetica", title\_font\_size=16)  
fig.update\_xaxes(tickangle=45, tickfont=dict(size=12))  
  
fig.show()  
  
os.makedirs("output", exist\_ok=True)  
fig.write\_html("output/naics\_salary\_box.html", include\_plotlyjs="cdn")

[Stage 39:> (0 + 1) / 1]

| NAICS2\_NAME | SALARY |
| --- | --- |
| Retail Trade | 115024.0 |
| Administrative and Support and Waste Management and Remediation Services | 115024.0 |
| Finance and Insurance | 115024.0 |
| Finance and Insurance | 115024.0 |
| Unclassified Industry | 92500.0 |
| Information | 110155.0 |
| Manufacturing | 115024.0 |
| Finance and Insurance | 115024.0 |
| Unclassified Industry | 115024.0 |
| Professional, Scientific, and Technical Services | 92962.0 |
| Wholesale Trade | 107645.0 |
| Administrative and Support and Waste Management and Remediation Services | 115024.0 |
| Finance and Insurance | 115024.0 |
| Professional, Scientific, and Technical Services | 192800.0 |
| Finance and Insurance | 81286.0 |

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

***Explanation*** This table represents salary distribution by industry. According to the table, Healthcare Services, Waste Management Services, and Information Services have the largest ranges of salaries while also claiming the highest salaries.

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

#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={  
 "LOT\_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="Helvetica",  
 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="red"  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="red"  
 )  
)  
  
fig.show()  
  
# save artifact without kaleido  
import os  
os.makedirs("output", exist\_ok=True)  
  
try:  
 fig.write\_image("output/Q7.svg", width=1000, height=600, scale=1)  
except Exception:  
 # no kaleido? save interactive HTML instead  
 fig.write\_html("output/Q7.html", include\_plotlyjs="cdn")  
 print("Saved interactive HTML fallback to output/Q7.html (no kaleido).")

[Stage 40:> (0 + 1) / 1]

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

Saved interactive HTML fallback to output/Q7.html (no kaleido).

***Explanation*** This bubble chart shows the top ten job titles by number of job postings. We can see that Business Intelligence Analysts and Data Mining Analysts, although the most abundant, do not have the highest salaries in comparison to more technical groups such as Computer System Engineers. The least amount of job postings lie in the business analyst and market research analyst positions, while also paying the lowest salaries.

# 5. Salary by Education Level

from pyspark.sql.functions import when, col  
import plotly.express as px  
import pandas as pd  
import os, re  
from IPython.display import display, HTML  
  
SHOW\_PREVIEW = True  
  
# Build groups  
lower\_deg = ["Bachelor's", "Associate", "GED", "No Education Listed", "High School"]  
higher\_deg = ["Master's degree", "PHD or professional degree"]  
lower\_pat = "(?i)(" + "|".join(map(re.escape, lower\_deg)) + ")"  
higher\_pat = "(?i)(" + "|".join(map(re.escape, higher\_deg)) + ")"  
  
df2 = (  
 df.withColumn(  
 "EDU\_GROUP",  
 when(col("EDUCATION\_LEVELS\_NAME").rlike(lower\_pat), "Bachelor's or lower")  
 .when(col("EDUCATION\_LEVELS\_NAME").rlike(higher\_pat), "Master's or PhD")  
 .otherwise("Other")  
 )  
 .withColumn("MAX\_YEARS\_EXPERIENCE", col("MAX\_YEARS\_EXPERIENCE").cast("float"))  
 .withColumn("Average\_Salary", col("Average\_Salary").cast("float"))  
 .filter(  
 col("MAX\_YEARS\_EXPERIENCE").isNotNull() &  
 col("Average\_Salary").isNotNull() &  
 (col("MAX\_YEARS\_EXPERIENCE") > 0) &  
 (col("Average\_Salary") > 0)  
 )  
)  
  
df\_filtered = df2.filter(col("EDU\_GROUP").isin("Bachelor's or lower", "Master's or PhD"))  
  
# To pandas  
df\_pd = df\_filtered.toPandas()  
  
# Optional preview table  
if SHOW\_PREVIEW:  
 cols\_to\_show = [c for c in ["EDU\_GROUP","MAX\_YEARS\_EXPERIENCE","Average\_Salary",  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"] if c in df\_pd.columns]  
 display(HTML(df\_pd.loc[:, cols\_to\_show].head(10).to\_html(index=False)))  
  
# Scatter plot  
hover\_cols = [c for c in ["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"] if c in df\_pd.columns]  
fig1 = px.scatter(  
 df\_pd,  
 x="MAX\_YEARS\_EXPERIENCE",  
 y="Average\_Salary",  
 color="EDU\_GROUP",  
 hover\_data=hover\_cols,  
 title="<b>Experience vs Salary by Education Level</b>",  
 opacity=0.7,  
 color\_discrete\_sequence=["blue", "red"]  
)  
fig1.update\_traces(marker=dict(size=7, line=dict(width=1, color="black")))  
fig1.update\_layout(  
 plot\_bgcolor="#f9f9f9",  
 paper\_bgcolor="#FFF5DC",  
 font=dict(family="Helvetica", size=14),  
 title\_font=dict(size=22),  
 xaxis\_title="Years of Experience",  
 yaxis\_title="Average Salary (USD)",  
 legend\_title="Education Group",  
 hoverlabel=dict(bgcolor="white", font\_size=13, font\_family="Helvetica"),  
 margin=dict(t=70, b=60, l=60, r=60),  
 xaxis=dict(gridcolor="lightgrey", tickmode="linear", dtick=1),  
 yaxis=dict(gridcolor="lightgrey")  
)  
  
# ✅ Return fig1 so Quarto embeds THIS chart (don’t call fig.show())  
fig1.show()  
  
# Save interactive artifact  
os.makedirs("output", exist\_ok=True)  
fig1.write\_html("output/q\_1a\_Experience\_vs\_Salary\_by\_Education\_Level.html", include\_plotlyjs="cdn")

[Stage 43:> (0 + 1) / 1]

| EDU\_GROUP | MAX\_YEARS\_EXPERIENCE | Average\_Salary | LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME |
| --- | --- | --- | --- |
| Bachelor's or lower | 2.0 | 108668.5 | General ERP Analyst / Consultant |
| Bachelor's or lower | 3.0 | 108668.5 | Oracle Consultant / Analyst |
| Bachelor's or lower | 7.0 | 108668.5 | General ERP Analyst / Consultant |
| Bachelor's or lower | 2.0 | 92962.0 | Data Analyst |
| Bachelor's or lower | 5.0 | 108668.5 | Data Analyst |
| Bachelor's or lower | 3.0 | 108668.5 | Oracle Consultant / Analyst |
| Bachelor's or lower | 8.0 | 165000.0 | Enterprise Architect |
| Bachelor's or lower | 2.0 | 108668.5 | Data Analyst |
| Bachelor's or lower | 2.0 | 75026.0 | Oracle Consultant / Analyst |
| Bachelor's or lower | 1.0 | 108668.5 | Data Analyst |

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

***Explanation*** This table analyzes salary by education level. It indicates that there is not an extreme discrepancy surrounding bachelor’s degree holders versus higher education holders, but those with higher education beyond a bachelor’s degree do tend to have higher salaries. However, years of service does not seem to generate a higher salary. Alternatively, those who have the highest salaries are earning it during the 5-10 year mark of experience in their careers. Years of experience does not necessarily translate to higher salaries, but higher education can.

# 6. Salary by Remote Work Type

from pyspark.sql.functions import col, when, lower, trim  
import plotly.express as px  
import numpy as np, pandas as pd, os  
from IPython.display import Markdown, display  
  
os.makedirs("output", exist\_ok=True)  
np.random.seed(42)  
  
# 1) Build groups: Remote / Hybrid / Onsite (Onsite includes None/blank)  
df\_remote = (  
 df.withColumn(  
 "REMOTE\_GROUP",  
 when(lower(trim(col("REMOTE\_TYPE\_NAME"))) == "remote", "Remote")  
 .when(lower(trim(col("REMOTE\_TYPE\_NAME"))) == "hybrid", "Hybrid")  
 .otherwise("Onsite")  
 )  
 .withColumn("MAX\_YEARS\_EXPERIENCE", col("MAX\_YEARS\_EXPERIENCE").cast("float"))  
 .withColumn("Average\_Salary", col("Average\_Salary").cast("float"))  
 .filter(  
 col("MAX\_YEARS\_EXPERIENCE").isNotNull() &  
 col("Average\_Salary").isNotNull() &  
 (col("MAX\_YEARS\_EXPERIENCE") > 0) &  
 (col("Average\_Salary") > 0)  
 )  
 .select("REMOTE\_GROUP", "MAX\_YEARS\_EXPERIENCE", "Average\_Salary",  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME")  
)  
  
# 2) To pandas   
pdf = df\_remote.toPandas()  
pdf["EXP\_JITTER"] = (  
 pdf["MAX\_YEARS\_EXPERIENCE"].astype(float)  
 + np.random.uniform(-0.15, 0.15, len(pdf))  
).clip(lower=0)  
  
# 3) y axis  
def \_round\_up(x, base=50000):  
 return int(base \* np.ceil(max(x, 1) / base))  
ymax = \_round\_up(float(pdf["Average\_Salary"].max()) if len(pdf) else 50000, 50000)  
yticks = list(range(0, ymax + 1, 50000))  
yticktext = [("0" if v == 0 else f"{v//1000}k") for v in yticks]  
  
# 4) colors  
cmap = {"Onsite": "#4C6EF5", "Remote": "#E8590C", "Hybrid": "#2B8A3E"} # blue, orange, green  
  
# 5) Single consolidated scatter  
fig = px.scatter(  
 pdf,  
 x="EXP\_JITTER",  
 y="Average\_Salary",  
 color="REMOTE\_GROUP",  
 hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME", "MAX\_YEARS\_EXPERIENCE"],  
 title="Experience vs Salary by Remote Work Type",  
 opacity=0.7,  
 color\_discrete\_map=cmap  
)  
  
# marker outlines + layout   
fig.update\_traces(marker=dict(size=8, line=dict(width=1, color="black")))  
fig.update\_layout(  
 font=dict(family="Helvetica", size=16, color="#1c2a39"),  
 title\_font=dict(size=34),  
 legend\_title="Remote Work Type",  
 legend=dict(x=1.02, y=1, bgcolor="rgba(255,255,255,0.7)"),  
 xaxis\_title="Years of Experience",  
 yaxis\_title="Average Salary (USD)",  
 xaxis=dict(gridcolor="#e0e0e0", tickmode="linear", dtick=1),  
 yaxis=dict(gridcolor="#e0e0e0", tickvals=yticks, ticktext=yticktext),  
 plot\_bgcolor="#F3F5F7", # inner panel  
 paper\_bgcolor="#EAF6E8", # soft green page  
 margin=dict(t=90, r=140, b=70, l=80),  
)  
  
fig.show()  
  
fig.write\_html("output/remote\_scatter\_all.html", include\_plotlyjs="cdn")

[Stage 44:> (0 + 1) / 1]

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

***Explanation*** This table compares experience versus salary in onsite and remote workers. Onsite and remote workers do not have much difference in salary, however, there are more remote workers that have less experience than onsite workers with more experience. This is most likely caused by several different external factors and would require further analysis.