Assignment 03

Dakota Alder - dkalder

September 22, 2025

## 0.1 Loading 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("data/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)

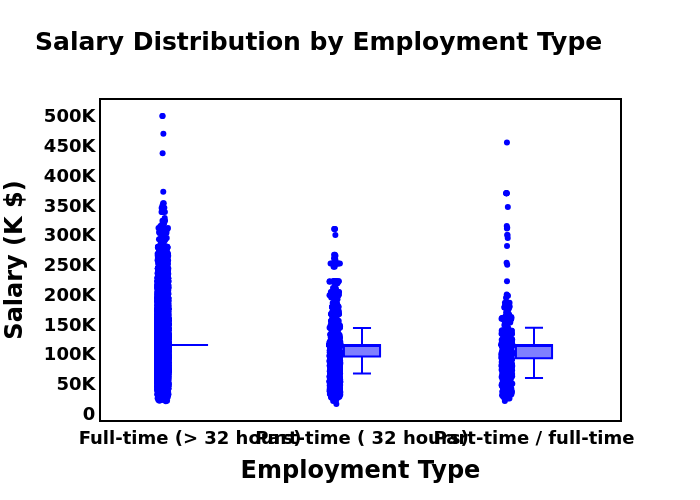
WARNING: Using incubator modules: jdk.incubator.vector  
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
25/09/25 03:44:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
25/09/25 03:44:38 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.  
[Stage 1:> (0 + 1) / 1] 25/09/25 03:44:51 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.

## 0.2 Data Preparation

[Stage 2:> (0 + 1) / 1] [Stage 3:> (0 + 1) / 1] [Stage 4:> (0 + 1) / 1] [Stage 5:> (0 + 1) / 1]

## 0.3 Salary Distribution by Employment Type

[Stage 6:> (0 + 1) / 1]

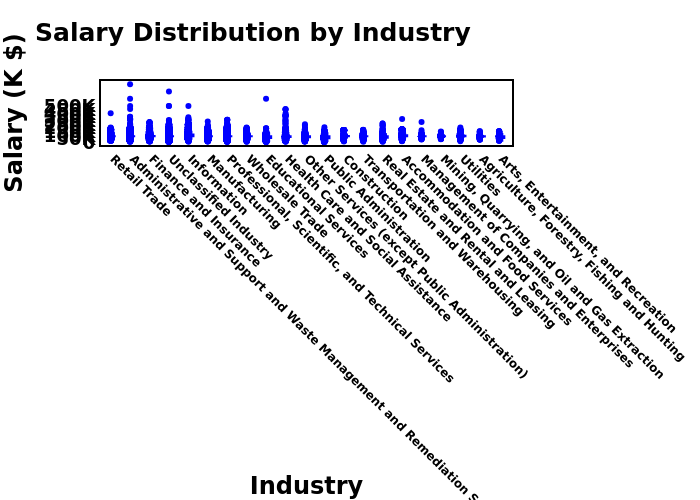


# 1. This graph shows that full time employees will have a higher salary on average than part time or mixed employment types. There is a higher range for salary for full-time employees vs. others.

## 1.1 Salary Distribution by Industry

#Question 2 Code  
  
#Select Industry and Salary Columns  
pdf = df.select("NAICS2\_NAME", "SALARY\_FROM").toPandas()  
  
  
#Create box plot with Horizontal grid lines  
fig = px.box(  
 pdf,  
 x="NAICS2\_NAME",  
 y="SALARY\_FROM",  
 title="Salary Distribution by Industry",  
 color\_discrete\_sequence=["blue"],  
 boxmode="group",  
 points="all",  
)  
  
  
  
fig.update\_layout(  
 title=dict(  
 text="Salary Distribution by Industry",  
 font=dict(size=25, family="Arial", color="black", weight="bold")  
 ),  
 xaxis=dict(  
 title=dict(text="Industry", font=dict(size=24, family="Arial", color="black", weight="bold")),  
 tickangle=45,  
 tickfont=dict(size=12, family="Arial", color="black",weight="bold"),  
 showline=True,  
 linewidth=2,  
 linecolor="black",  
 mirror=True,  
 showgrid=False,  
 categoryorder="array",  
 categoryarray=sorted\_employment\_types.tolist()  
 ),  
 yaxis=dict(  
 title=dict(text="Salary (K $)", font=dict(size=24, family="Arial", color="black", weight="bold")),  
 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="Arial", color="black",weight="bold"),  
 showline=True,  
 linewidth=2,  
 linecolor="black",  
 mirror=True,  
 showgrid=False,  
 gridcolor="lightgray",  
 gridwidth=.5  
 ),  
 font=dict(family="Arial", size=16, color="black"),  
 boxgap=0.7,  
 plot\_bgcolor="white",  
 paper\_bgcolor="white",  
 showlegend=False,  
 height=1200,  
 width=1000,  
)  
fig.show(renderer="png")  
fig.write\_html("output/Q2.html")  
#fig.write\_image("output/Q2.svg", width=850, height=500, scale=1)

[Stage 7:> (0 + 1) / 1]

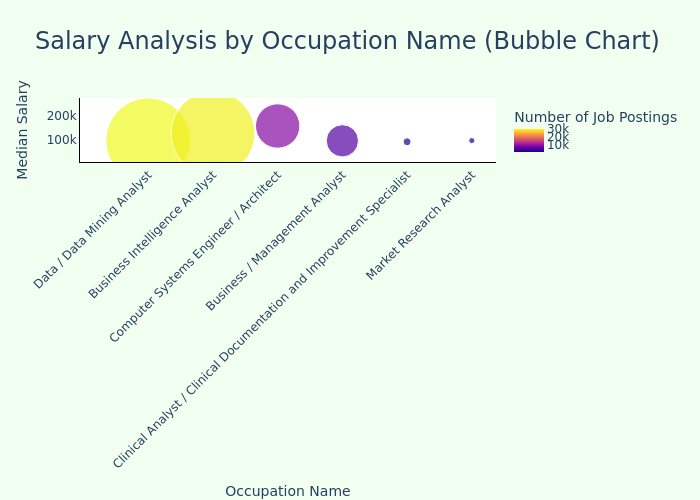


# 2. This visual shows that industries in the arts, agriculture, mining, and more blue collar work tend to have lower salaries then professional services, administration, more white collar jobs.

## 2.1 Question 2: Salary Analysis by ONET Occupation Type

#Query the required data with Spark SQL  
  
onet\_type = 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  
 """)  
  
onet\_pd = onet\_type.toPandas()  
  
fig = px.scatter(  
 onet\_pd,  
 x="Occupation\_Name",  
 y="Median\_Salary",  
 size="Job\_Postings",  
 title="Salary Analysis by Occupation Name (Bubble Chart)",  
 labels={  
 "Occupation\_Name": "Occupation Name",  
 "Median\_Salary": "Median Salary",  
 "Job\_Postings": "Number of Job Postings"  
 },  
 hover\_name="Occupation\_Name",  
 size\_max=60,  
 width=1000,  
 height=1000,  
 color="Job\_Postings",  
 color\_continuous\_scale="Plasma"  
)  
  
fig.update\_layout(  
 font\_family="Arial",  
 font\_size=12,  
 title\_font\_size=24,  
 xaxis\_title="Occupation Name",  
 yaxis\_title="Median Salary",  
 plot\_bgcolor="white",  
 paper\_bgcolor="#f0fff0",  
 xaxis=dict(  
 tickangle=-45,  
 showline=True,  
 linecolor="black"  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="black"  
 )  
)  
fig.show(renderer="png")  
fig.write\_html("output/Q3.html")

[Stage 8:> (0 + 1) / 1]

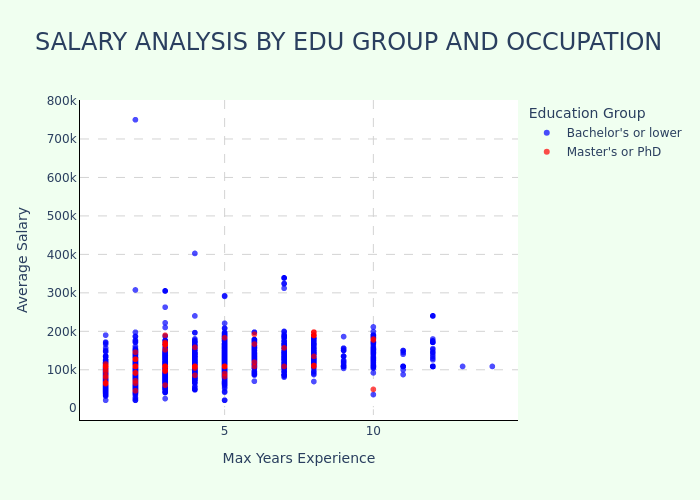


# 3. This figure shows that Data and Business Analysts not only have higher salaries than market research and clinical analysts, but there are also a lot more openings in the field.

## 3.1 Salary by Education Level

#Create 2 groups of Education Levels  
lower\_deg = ["Bachelor's", "Associate", "GED", "No Education Listed", "High School"]  
higher\_deg = ["Master's degree", "PhD or professional degree"]  
  
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")  
)  
  
  
df = df.withColumn("Average\_Salary", col("Average\_Salary").cast("float"))  
  
df = df.filter(  
 col("MAX\_YEARS\_EXPERIENCE").isNotNull() &  
 col("Average\_Salary").isNotNull() &  
 (col("MAX\_YEARS\_EXPERIENCE") > 0) &  
 (col("Average\_Salary") > 0)  
)  
  
  
df\_filtered = df.filter(col("EDU\_GROUP").isin("Bachelor's or lower", "Master's or PhD"))  
  
edu\_cols = [  
 "EDU\_GROUP",  
 "MAX\_YEARS\_EXPERIENCE",  
 "Average\_Salary",  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"  
]  
  
df\_filtered = df\_filtered.select(\*edu\_cols)  
edu\_pd = df\_filtered.toPandas()  
  
fig = px.scatter(  
 edu\_pd,  
 x="MAX\_YEARS\_EXPERIENCE",  
 y="Average\_Salary",  
 color="EDU\_GROUP",  
 title="SALARY ANALYSIS BY EDU GROUP AND OCCUPATION",  
 opacity=.7,  
 color\_discrete\_sequence=["blue", "red"],  
 hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
 width=1000,  
 height=600,  
)  
  
fig.update\_layout(  
 font\_family="Arial",  
 font\_size=12,  
 title\_font\_size=24,  
 xaxis\_title="Max Years Experience",  
 yaxis\_title="Average Salary",  
 legend\_title="Education Group",  
 plot\_bgcolor="white",  
 paper\_bgcolor="#f0fff0",  
 xaxis=dict(  
 showline=True,  
 linecolor="black",  
 showgrid=True,  
 gridcolor="lightgray",  
 gridwidth=1,  
 griddash="dash"  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="black",  
 showgrid=True,  
 gridcolor="lightgray",  
 gridwidth=1,  
 griddash="dash"  
 )  
)  
fig.show(renderer="png")  
fig.write\_html("output/Q4.html")

[Stage 11:> (0 + 1) / 1]

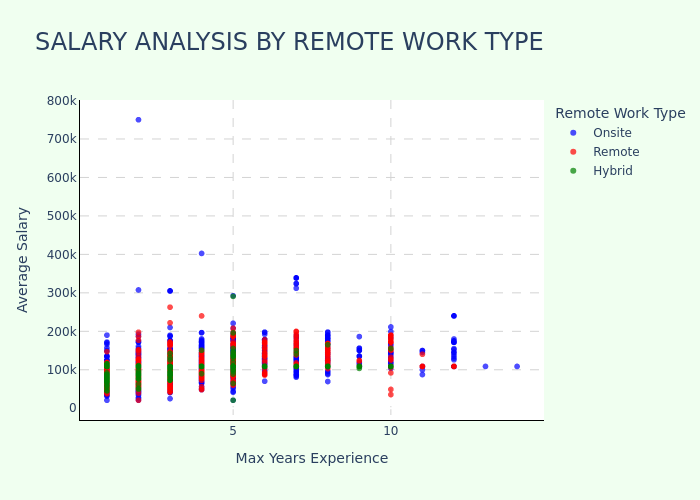


# 4. This visualization is showing that salaries don’t tend to go much higher, even with higher education. It seems that years of experience tends to increase salaries.

## 4.1 Salary by Remote Work Type

# Group Remote Types into 3 groups  
df = df.withColumn(  
 "REMOTE\_GROUP",  
 when(col("REMOTE\_TYPE\_NAME").rlike("(?i)hybrid"), "Hybrid")  
 .when(col("REMOTE\_TYPE\_NAME").rlike("(?i)remote"), "Remote")  
 .otherwise("Onsite")  
)  
  
df = df.withColumn("Average\_Salary", col("Average\_Salary").cast("float"))  
  
df = df.filter(  
 col("MAX\_YEARS\_EXPERIENCE").isNotNull() &  
 col("Average\_Salary").isNotNull() &  
 (col("MAX\_YEARS\_EXPERIENCE") > 0) &  
 (col("Average\_Salary") > 0)  
)  
  
  
remote\_filtered = df.filter(col("REMOTE\_GROUP").isin("Remote", "Hybrid", "Onsite"))  
  
remote\_cols = [  
 "REMOTE\_GROUP",  
 "MAX\_YEARS\_EXPERIENCE",  
 "Average\_Salary",  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"  
]  
  
remote\_filtered = remote\_filtered.select(\*remote\_cols)  
remote\_pd = remote\_filtered.toPandas()  
  
fig = px.scatter(  
 remote\_pd,  
 x="MAX\_YEARS\_EXPERIENCE",  
 y="Average\_Salary",  
 color="REMOTE\_GROUP",  
 title="SALARY ANALYSIS BY REMOTE WORK TYPE",  
 opacity=.7,  
 color\_discrete\_sequence=["blue", "red", "green"],  
 hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
 width=1000,  
 height=600,  
)  
  
fig.update\_layout(  
 font\_family="Arial",  
 font\_size=12,  
 title\_font\_size=24,  
 xaxis\_title="Max Years Experience",  
 yaxis\_title="Average Salary",  
 legend\_title="Remote Work Type",  
 plot\_bgcolor="white",  
 paper\_bgcolor="#f0fff0",  
 xaxis=dict(  
 showline=True,  
 linecolor="black",  
 showgrid=True,  
 gridcolor="lightgray",  
 gridwidth=1,  
 griddash="dash"  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="black",  
 showgrid=True,  
 gridcolor="lightgray",  
 gridwidth=1,  
 griddash="dash"  
 )  
)  
fig.show(renderer="png")  
fig.write\_html("output/Q5.html")

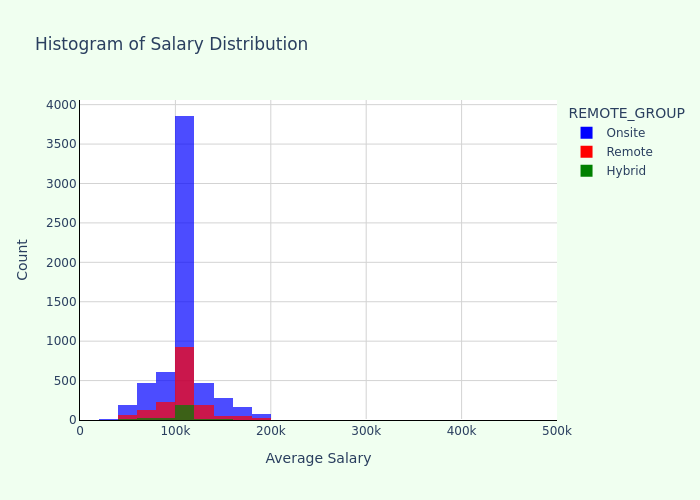
[Stage 12:> (0 + 1) / 1]



# 5. This graph shows that more outliers in the salary ranges are onsite workers.

## 5.1 Salary Histograms for Remote Type Work

#Histograms  
  
fig = px.histogram(  
 remote\_pd,  
 x = "Average\_Salary",  
 color="REMOTE\_GROUP",  
 nbins=40,  
 opacity=0.7,  
 barmode="overlay",  
 color\_discrete\_sequence=["blue","red","green"],  
 title="Histogram of Salary Distribution"  
)  
  
fig.update\_layout(  
 xaxis\_title="Average Salary",  
 yaxis\_title="Count",  
 plot\_bgcolor="white",  
 paper\_bgcolor="#f0fff0",  
 xaxis=dict(  
 showline=True,  
 linecolor="black",  
 showgrid=True,  
 gridcolor="lightgray",  
 gridwidth=1  
 ),  
 yaxis=dict(  
 showline=True,  
 linecolor="black",  
 showgrid=True,  
 gridcolor="lightgray",  
 gridwidth=1  
 )  
)  
fig.update\_xaxes(range=[0, 500000])  
  
  
fig.show(renderer="png")  
fig.write\_html("output/Q6.html")



# 6. This graph shows that regardless of remote work type, there is a correlation on salary and whether an employee is a remote, hybrid, or onsite worker.