Assignment 03

Ryan Martin

September 24, 2025

https://github.com/met-ad-688/assignment-03-ryanmt1998.git

from pyspark.sql import SparkSession  
import pandas as pd  
import plotly.express as px  
import plotly.io as pio  
pio.renderers.default = "svg"  
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("./data/LightcastData").getOrCreate()  
  
# Load Data  
df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","true").option("escape", "\"").csv("./data/lightcast\_job\_postings.csv")  
  
# 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 01:49:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
[Stage 1:> (0 + 1) / 1]

## 1 1. Data Preperation

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")) \  
   
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)  
  
df = df.fillna({  
 "SALARY\_FROM": median\_from,  
 "SALARY\_TO": median\_to,  
 "SALARY": median\_salary  
})  
  
df= df.withColumn("Average\_Salary", (col("SALARY\_FROM") + col("SALARY\_TO")) / 2)  
  
  
export\_cols = [  
 "EDUCATION\_LEVELS\_NAME",  
 "REMOTE\_TYPE\_NAME",  
 "MAX\_YEARS\_EXPERIENCE",  
 "Average\_Salary",  
 "SALARY",  
 "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"  
]   
  
df\_selected = df.select(\*export\_cols)  
  
pdf = df\_selected.toPandas()  
pdf.to\_csv("./data/lightcast\_cleaned.csv", index=False)  
  
  
print("Data Cleaning Complete. Rows retained:", len(pdf))

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

Medians: 87295.0 130042.0 115024.0  
Data Cleaning Complete. Rows retained: 72498

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

## 3 Salary Distribution by Employment Type

The salary distribution by employment type graph emphasizes that full-time employment attracts higher salaries and has a broader range of payment. The part-time jobs appear to offer lower wages, but because of their few outliers, they represent more consistent salaries. The part-time/full-time mixed job has lower wages, similar to those in the part-time category, and outliers, such as the full-time category. The wages of full-time jobs demonstrate the benefits of an experienced job.

# import pandas as pd  
  
  
# pdf = df.filter(df["SALARY"] > 0).select("EMPLOYMENT\_TYPE\_NAME", "SALARY").toPandas()  
# import re  
# pdf["EMPLOYMENT\_TYPE\_NAME"] = pdf["EMPLOYMENT\_TYPE\_NAME"].apply(  
# lambda x: re.sub(r"[^\x00-\x7F]+", "", str(x)).strip() if pd.notnull(x) else ""  
# )  
  
# pdf = pdf[pdf["EMPLOYMENT\_TYPE\_NAME"] != ""]  
  
# median\_salaries = pdf.groupby("EMPLOYMENT\_TYPE\_NAME")["SALARY"].median()  
# median\_salaries.head()  
  
# sorted\_employment\_types = median\_salaries.sort\_values(ascending=False).index  
  
# pdf["EMPLOYMENT\_TYPE\_NAME"] = pd.Categorical(  
# pdf["EMPLOYMENT\_TYPE\_NAME"],  
# categories=sorted\_employment\_types,  
# ordered=True  
# )  
# fig = px.box(  
# pdf,  
# x="EMPLOYMENT\_TYPE\_NAME",  
# y="SALARY",  
# title="Salary Distribution by Employment Type",  
# color\_discrete\_sequence=["#1f77b4"],   
# boxmode="group",  
# points="all",  
# )  
  
# fig.update\_layout(  
# title=dict(  
# text="Salary Distribution by Employment Type",  
# font=dict(size=20, family="Helvetica", color="black", weight = "bold")  
# ),  
   
# xaxis=dict(  
# title=dict(text="Employment Type", font=dict(size=14, family = "Helvetica", color = "black", weight = "bold")),  
# tickangle=0,   
# 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=14, family="Helvetica", 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=12, family="Helvetica", color="black", weight="bold"),  
# showline=True,  
# linewidth=2,  
# linecolor="black",  
# mirror=True,  
# showgrid=False,  
# gridcolor="lightgray",  
# gridwidth=0.5,  
# ),  
# font=dict(family="Helvetica", size = 12, color = "black"),  
# boxgap=0.7,  
# plot\_bgcolor="white",  
# paper\_bgcolor="white",  
# showlegend=False,  
# height=500,  
# width=800,  
  
  
# )  
  
  
  
# fig.show()  
# fig.write\_html("output/Q1.html")  
# fig.write\_image("output/Q1.svg", width=850, height=500, scale=1)

## 4 Salary Distribution by Industry

The distribution plot displays the significant pay gap between certain sections of the job industry. The highest-paying jobs, which seem to pay around or over 200k, are in health, finance, and technology. The industries with the lowest salaries are mostly making around 50k, which includes jobs in the agricultural, retail, and food service industries.

# import pandas as pd  
# import re  
# import plotly.express as px  
  
# pdf = df.filter(df["SALARY"] > 0).select("NAICS2\_NAME", "SALARY").toPandas()  
  
# pdf["NAICS2\_NAME"] = pdf["NAICS2\_NAME"].apply(  
# lambda x: re.sub(r"[^\x00-\x7F]+", "", str(x)).strip() if pd.notnull(x) else ""  
# )  
  
# pdf = pdf[pdf["NAICS2\_NAME"] != ""]  
  
# median\_salaries = pdf.groupby("NAICS2\_NAME")["SALARY"].median()  
# sorted\_industries = median\_salaries.sort\_values(ascending=False).index  
  
# pdf["NAICS2\_NAME"] = pd.Categorical(  
# pdf["NAICS2\_NAME"],  
# categories=sorted\_industries,  
# ordered=True  
# )  
  
# fig = px.box(  
# pdf,  
# x="NAICS2\_NAME",  
# y="SALARY",  
# title="Salary Distribution by Industry",  
# color\_discrete\_sequence=["#EF553B"], # bright red for boxes  
# boxmode="group",  
# points="all",  
# )  
  
# fig.update\_layout(  
# title=dict(  
# text="Salary Distribution by Industry",  
# font=dict(size=30, family="Georgia", color="#990000", weight="bold") # dark red title  
# ),  
# xaxis=dict(  
# title=dict(text="Industry", font=dict(size=14, family="Georgia", color="#B22222", weight="bold")), # firebrick red  
# tickangle=45,  
# tickfont=dict(size=12, family="Georgia", color="#B22222", weight="bold"),  
# showline=True,  
# linewidth=2,  
# linecolor="#B22222",  
# mirror=True,  
# showgrid=False,  
# categoryorder="array",  
# categoryarray=sorted\_industries.tolist()  
# ),  
# yaxis=dict(  
# title=dict(text="Salary (K $)", font=dict(size=14, family="Georgia", color="#800000", weight="bold")), # maroon  
# tickvals=[100000, 200000, 300000, 400000, 500000],  
# ticktext=["100K", "200K", "300K", "400K", "500K"],  
# tickfont=dict(size=12, family="Georgia", color="#800000", weight="bold"),  
# showline=True,  
# linewidth=2,  
# linecolor="#800000",  
# mirror=True,  
# showgrid=False,  
# gridcolor="#F5B7B1",  
# gridwidth=0.5,  
# ),  
# font=dict(family="Georgia", size=12, color="#800000"),  
# boxgap=0.7,  
# plot\_bgcolor="#FFF0F0",  
# paper\_bgcolor="#FFF5F5",  
# showlegend=False,  
# height=900,  
# width=1100,  
# )  
  
# fig.show()  
# fig.write\_html("output/Q2.html")  
# fig.write\_image("output/Q2.svg", width=1100, height=900, scale=1)

## 5 Salary Analysis by ONET Occupation Type (Bubble Chart)

# df.createOrReplaceTempView("Job\_Postings")

The salaries for occupations in the ONET taxonomy range from around 100k to 120k, with a few outliers being present. Occupations such as Data Governance Analysts and ERP Business Analysts offer good salaries but have smaller job markets, indicating that they have fewer job postings. On the other hand, Data Analysts have many more job postings, mostly because it is a much broader topic of study compared to the alternatives. The biggest takeaway is that more specialized jobs demand higher salaries, and data analysts have a strong grasp of this job market.

# salary\_analysis = spark.sql("""  
# Select  
# TITLE\_NAME AS ONET\_NAME,  
# PERCENTILE(SALARY, 0.5) AS Median\_Salary,  
# COUNT(\*) AS Job\_Postings  
# FROM job\_Postings  
# GROUP BY TITLE\_NAME  
# ORDER BY Job\_Postings DESC  
# LIMIT 10  
# """)  
  
# salary\_pd = salary\_analysis.toPandas()  
# salary\_pd.head()  
  
# import plotly.express as px  
  
# fig = px.scatter(  
# salary\_pd,  
# x = "ONET\_NAME",  
# y="Median\_Salary",  
# size="Job\_Postings",  
# title="Median Salary by ONET Occupation Type (Bubble Chart)",  
# labels = {"ONET\_NAME": "ONET Occupation", "Median\_Salary": "Median Salary", "Job\_Postings": "Number of Job Postings"  
# },  
# hover\_name = "ONET\_NAME",  
# size\_max = 60,  
# width=1000,  
# height=600,  
# color="Job\_Postings",  
# color\_continuous\_scale="Jet"  
# )  
# fig.update\_layout(  
# font\_family="Cambria",  
# font\_size = 14,  
# title\_font\_size=25,  
# xaxis\_title = "ONET Occupation",  
# yaxis\_title= "Median Salary",  
# plot\_bgcolor="white",  
# xaxis=dict(  
# tickangle=-45,  
# showline=True,  
# linecolor="black"  
# ),  
# yaxis=dict(  
# showline=True,  
# linecolor="black"  
# )  
# )  
  
  
# fig.show()  
# fig.write\_html("output/Q3.html")  
# fig.write\_image("output/Q3.svg", width=1100, height=900, scale=1)

## 6 Salary by Education Level

# lower\_deg = ["Bachelor's", "Associate's", "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("MAX\_YEARS\_EXPERIENCE", col("MAX\_YEARS\_EXPERIENCE").cast("float"))  
# 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"))  
  
# df\_pd = df\_filtered.toPandas()  
  
# import numpy as np  
  
  
# jitter\_amount = 0.15   
# df\_pd['MAX\_YEARS\_EXPERIENCE\_JITTER'] = (  
# df\_pd['MAX\_YEARS\_EXPERIENCE'] +  
# np.random.uniform(-jitter\_amount, jitter\_amount, len(df\_pd))  
# )

## 7 Scatter Plot 1

For individuals with a Bachelor’s or lower, the average salary ranges from around 50k to 200k. People with greater job experience tend to earn a salary closer to 200k with each additional year of experience. For example, starting with 6 years of experience, salaries were consistently above 100k. A Bachelor’s degree can earn you a well-paying job, but additional experience can earn you an even greater salary than someone with less experience.

# df\_bachelor = df\_pd[df\_pd["EDU\_GROUP"]=="Bachelor's or lower"]  
# fig1 = px.scatter(  
# df\_bachelor,  
# x="MAX\_YEARS\_EXPERIENCE\_JITTER",  
# y="AVERAGE\_SALARY",  
# color = "EDU\_GROUP",  
# hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
# title="<b>Experience vs Salary by Education Level</b>",  
# opacity=0.7,  
# color\_discrete\_sequence=["#636efa"]  
  
# )  
  
# 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="Segoe UI", 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="Roboto"),  
# margin=dict(t=70,b=60, l=60, r=60),  
# xaxis=dict(  
# gridcolor="lightgrey",  
# tickmode='linear',  
# dtick=1  
# ),  
# yaxis=dict(gridcolor="lightgrey")  
# )  
# fig1.show()  
# fig1.write\_html("output/q\_1a\_Experience\_vs\_Salary\_by\_Education\_Level\_Bachelors.html")  
# fig1.write\_image("output/q\_1a\_Experience\_vs\_Salary\_by\_Education\_Level\_Bachelors.svg", width=1100, height=900, scale=1)

## 8 Scatter Plot 2

This graph emphasizes the benefits of having a Master’s or PhD in addition to a Bachelor’s degree. While the sample size may be smaller, the benefits of a Master’s or PHD are apparent. There is a positive relationship between having this level of education and an increasing number of years of experience. Salaries appear to average around 150k a year, with the outliers below that decreasing each year. Comparing this to the scatter plot for groups with a Bachelor’s degree or below, people with a Master’s or PhD are expected to earn more, even with as little as 1 year of experience.

# df\_master = df\_pd[df\_pd["EDU\_GROUP"]=="Master's or PHD"]  
# fig2 = px.scatter(  
# df\_master,  
# x="MAX\_YEARS\_EXPERIENCE\_JITTER",  
# y="AVERAGE\_SALARY",  
# color = "EDU\_GROUP",  
# hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
# title="<b>Experience vs Salary by Education Level</b>",  
# opacity=0.7,  
# color\_discrete\_sequence=["#FFC0CB"]  
  
# )  
  
# fig2.update\_traces(marker=dict(size=7, line=dict(width=1, color="black")))  
  
# fig2.update\_layout(  
# plot\_bgcolor="#f9f9f9",  
# paper\_bgcolor="#FFF5DC",  
# font=dict(family="Segoe UI", 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="Roboto"),  
# margin=dict(t=70,b=60, l=60, r=60),  
# xaxis=dict(  
# gridcolor="lightgrey",  
# tickmode='linear',  
# dtick=1  
# ),  
# yaxis=dict(gridcolor="lightgrey")  
# )  
# fig2.show()  
# fig2.write\_html("output/q\_1a\_Experience\_vs\_Salary\_by\_Education\_Level\_Masters.html")  
# fig2.write\_image("output/q\_1a\_Experience\_vs\_Salary\_by\_Education\_Level\_Masters.svg", width=1100, height=900, scale=1)

## 9 Histogram 1

The histogram appears right-skewed, with most people earning a salary of around 100k. The density peak is also around the same amount, further solidifying that many people with a Bachelor’s or below earn around this amount. It is very rare for a person to earn above 150k, but those who do most likely have many years of experience.

# import seaborn as sns  
# import matplotlib.pyplot as plt  
  
# sns.histplot(data=df\_pd[df\_pd['EDU\_GROUP'] == "Bachelor's or lower"],  
# x='AVERAGE\_SALARY',  
# bins=30,  
# kde=True,  
# color='#FFFF00',  
# stat='density',  
# alpha=0.5)  
  
# plt.title("Bachelor's or Lower Histogram")  
# plt.xlabel('Average Salary (USD)')  
# plt.ylabel('Density')  
# plt.savefig("output/Bachelors\_or\_Lower\_Histogram.svg", bbox\_inches='tight')  
  
# plt.show()

## 10 Histogram 2

The histogram generated creates a bimodal distribution, with most people earning around 110k. There is a significant increase at around 190k, highlighting that some people earn that much, demonstrating the benefits of having a higher education. Comparing this with the previous histogram, the salary distribution is larger for this one. Higher salaries are more abundant for people with a Master’s or a PhD.

# import seaborn as sns  
# import matplotlib.pyplot as plt  
  
# sns.histplot(data=df\_pd[df\_pd['EDU\_GROUP'] == "Master's or PHD"],  
# x='AVERAGE\_SALARY',  
# bins=30,  
# kde=True,  
# color='#636efa',  
# stat='density',  
# alpha=0.5)  
  
# plt.title("Master's or PHD Histogram")  
# plt.xlabel('Average Salary (USD)')  
# plt.ylabel('Density')  
# plt.savefig("output/Masters\_or\_PHD\_Histogram.svg", bbox\_inches='tight')  
# plt.show()

## 11 Salary by Remote Work Type

# from pyspark.sql.functions import when, col, trim  
  
# 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")  
# )  
  
# df = df.filter(  
# col("MAX\_YEARS\_EXPERIENCE").isNotNull() &  
# col("AVERAGE\_SALARY").isNotNull() &  
# (col("MAX\_YEARS\_EXPERIENCE") > 0) &  
# (col("AVERAGE\_SALARY") > 0)   
# )  
  
# df\_pd = df.select(  
# "MAX\_YEARS\_EXPERIENCE", "Average\_Salary",  
# "LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME", "REMOTE\_GROUP"  
# ).toPandas()  
  
# import numpy as np  
  
  
# jitter\_amount = 0.15   
# df\_pd['MAX\_YEARS\_EXPERIENCE\_JITTER'] = (  
# df\_pd['MAX\_YEARS\_EXPERIENCE'] +  
# np.random.uniform(-jitter\_amount, jitter\_amount, len(df\_pd))  
# )

## 12 Scatter Plot 1

For individuals with remote work experience, the average salary increases with each additional year of experience. People with 1 to 3 years of experience earn an average salary ranging from 50k to just under 150k. After that, there is a large amount of variability for the average salary for people who earn a greater wage. Some people with fewer years of experience can earn a salary above 100k, but most people with more than 3 years are expected to earn above that amount.

# remote\_df = df\_pd[df\_pd['REMOTE\_GROUP'] == 'Remote']  
# fig1 = px.scatter(  
# remote\_df,  
# x="MAX\_YEARS\_EXPERIENCE\_JITTER",  
# y="Average\_Salary",  
# color = "REMOTE\_GROUP",  
# hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
# title="<b>Experience vs Salary by Remote Work Type</b>",  
# opacity=0.7,  
# color\_discrete\_sequence=["#636efa"]  
  
# )  
  
# 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="Segoe UI", size=14),  
# title\_font=dict(size=22),  
# xaxis\_title="Years of Experience",  
# yaxis\_title="Average Salary (USD)",  
# legend\_title="Remote Work Type",  
# hoverlabel=dict(bgcolor="white", font\_size=13, font\_family="Garamond"),  
# margin=dict(t=70,b=60, l=60, r=60),  
# xaxis=dict(  
# gridcolor="lightgrey",  
# tickmode='linear',  
# dtick=1  
# ),  
# yaxis=dict(gridcolor="lightgrey")  
# )  
# fig1.show()  
# fig1.write\_html("output/Experience\_vs\_Salary\_by\_Remote\_Work\_Type.html")  
# fig1.write\_image("output/Experience\_vs\_Salary\_by\_Remote\_Work\_Type.svg", width=1100, height=900, scale=1)

## 13 Scatter Plot 2

Compared to remote workers, people with a hybrid job do not see an increase in their average salary with each year of experience. Most salaries cluster around 100k, despite a higher experience level. There is no upward trend present in the graph. There are also a large number of people who earn less than 100k, which was not present among those with remote jobs. There appears to be a form of salary cap in this type of work.

# hybrid\_df = df\_pd[df\_pd['REMOTE\_GROUP'] == 'Hybrid']  
# fig2 = px.scatter(  
# hybrid\_df,  
# x="MAX\_YEARS\_EXPERIENCE\_JITTER",  
# y="Average\_Salary",  
# color = "REMOTE\_GROUP",  
# hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
# title="<b>Experience vs Salary by Remote Work Type</b>",  
# opacity=0.7,  
# color\_discrete\_sequence=["#00CC96"]  
  
# )  
  
# fig2.update\_traces(marker=dict(size=7, line=dict(width=1, color="black")))  
  
# fig2.update\_layout(  
# plot\_bgcolor="#f9f9f9",  
# paper\_bgcolor="#FFF5DC",  
# font=dict(family="Segoe UI", size=14),  
# title\_font=dict(size=22),  
# xaxis\_title="Years of Experience",  
# yaxis\_title="Average Salary (USD)",  
# legend\_title="Remote Work Type",  
# hoverlabel=dict(bgcolor="white", font\_size=13, font\_family="Garamond"),  
# margin=dict(t=70,b=60, l=60, r=60),  
# xaxis=dict(  
# gridcolor="lightgrey",  
# tickmode='linear',  
# dtick=1  
# ),  
# yaxis=dict(gridcolor="lightgrey")  
# )  
# fig2.write\_html("output/Experience\_vs\_Salary\_by\_Remote\_Work\_Type\_Hybrid.html")  
# fig2.write\_image("output/Experience\_vs\_Salary\_by\_Remote\_Work\_Type\_Hybrid.svg", width=1100, height=900, scale=1)  
# fig2.show()

## 14 Scatter Plot 3

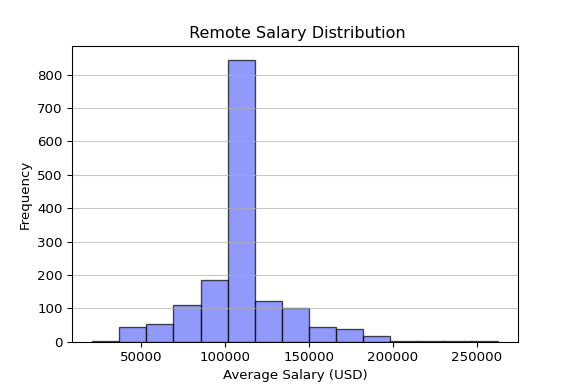
From the start, there appears to be a broader range for average salary compared to the other two types of work. People with an on-site job earn between 50k and around 750k. There is an upward trend in salaries with each additional year of experience, with most clustering around 50k to 200k. The higher salaries typically occur after 4 years of experience; in one case, a large number of people earned over 300k with 7 years of experience. On-site jobs offer a wide range of salary benefits that the other two job types can not match.

# onsite\_df = df\_pd[df\_pd['REMOTE\_GROUP'] == 'Onsite']  
# fig3 = px.scatter(  
# onsite\_df,  
# x="MAX\_YEARS\_EXPERIENCE\_JITTER",  
# y="Average\_Salary",  
# color = "REMOTE\_GROUP",  
# hover\_data=["LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME"],  
# title="<b>Experience vs Salary by Remote Work Type</b>",  
# opacity=0.7,  
# color\_discrete\_sequence=["#FFA500"]  
  
# )  
  
# fig3.update\_traces(marker=dict(size=7, line=dict(width=1, color="black")))  
  
# fig3.update\_layout(  
# plot\_bgcolor="#f9f9f9",  
# paper\_bgcolor="#FFF5DC",  
# font=dict(family="Segoe UI", size=14),  
# title\_font=dict(size=22),  
# xaxis\_title="Years of Experience",  
# yaxis\_title="Average Salary (USD)",  
# legend\_title="Remote Work Type",  
# hoverlabel=dict(bgcolor="white", font\_size=13, font\_family="Georgia"),  
# margin=dict(t=70,b=60, l=60, r=60),  
# xaxis=dict(  
# gridcolor="lightgrey",  
# tickmode='linear',  
# dtick=1  
# ),  
# yaxis=dict(gridcolor="lightgrey")  
# )  
# fig3.show()  
# fig3.write\_html("output/Experience\_vs\_Salary\_by\_Remote\_Work\_Type\_Onsite.html")  
# fig3.write\_image("output/Experience\_vs\_Salary\_by\_Remote\_Work\_Type\_Onsite.svg", width=1100, height=900, scale=1)

## 15 Histogram 1

The histogram for remote jobs is bell-shaped with signs of being right-skewed. Most jobs pay around 100k a week on average, as previously stated. There is a steady decrease in the average salary after 100k, indicating that very few people who have remote jobs earn higher salaries. On the other hand, there are very few people who earn wages on the lower end of the histogram. The typical salary range for people in a remote setting is 100k.

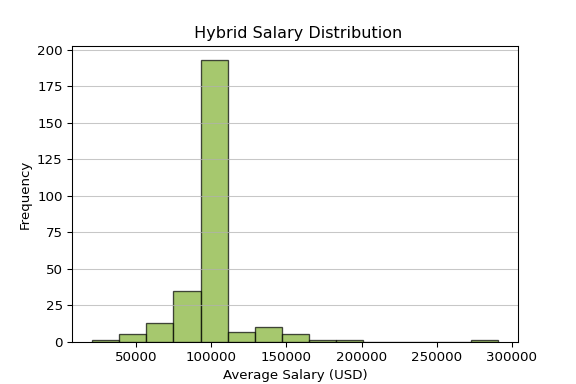
# import matplotlib.pyplot as plt  
  
# remote\_salaries = df\_pd[df\_pd['REMOTE\_GROUP'] == 'Remote']['Average\_Salary']  
  
# plt.figure(figsize=(6, 4))  
# plt.hist(remote\_salaries, bins=15, color='#636efa', alpha=0.7, edgecolor='black')  
# plt.title(' Remote Salary Distribution')  
# plt.xlabel('Average Salary (USD)')  
# plt.ylabel('Frequency')  
# plt.grid(axis='y', alpha=0.7)  
# plt.savefig("output/Salary\_Distribution\_Remote.png")  
# plt.show()



## 16 Histogram 2

The hybrid salary distribution is right-skewed, indicating that most people earn an average salary of 100k, with a frequency of less than 200. Remote jobs had a frequency of over 800 for people making 100k. Very few salaries are below 80k, but the same can be said for jobs earning above 150k. The histograms support the idea that remote jobs seemingly pay more than hybrid ones.

# import matplotlib.pyplot as plt  
  
# remote\_salaries = df\_pd[df\_pd['REMOTE\_GROUP'] == 'Hybrid']['Average\_Salary']  
  
# plt.figure(figsize=(6, 4))  
# plt.hist(remote\_salaries, bins=15, color='#80b131ff', alpha=0.7, edgecolor='black')  
# plt.title(' Hybrid Salary Distribution')  
# plt.xlabel('Average Salary (USD)')  
# plt.ylabel('Frequency')  
# plt.grid(axis='y', alpha=0.7)  
# plt.savefig("output/Salary\_Distribution\_Hybrid.png")  
# plt.show()



## 17 Histogram 3

Looking at this histogram for onsite jobs, average salaries frequently range around 100k, with a frequency of over 4000. There is a larger bar after 100k, showing that a large number of people earn closer to 200k. The onsite salary distribution has a much higher frequency than the remote and hybrid ones, and is also more concentrated. The histogram supports the idea that an onsite job will provide a more consistent and beneficial average salary.

# import matplotlib.pyplot as plt  
  
# remote\_salaries = df\_pd[df\_pd['REMOTE\_GROUP'] == 'Onsite']['Average\_Salary']  
  
# plt.figure(figsize=(6, 4))  
# plt.hist(remote\_salaries, bins=15, color='#e21b1eff', alpha=0.7, edgecolor='black')  
# plt.title(' Onsite Salary Distribution')  
# plt.xlabel('Average Salary (USD)')  
# plt.ylabel('Frequency')  
# plt.grid(axis='y', alpha=0.7)  
# plt.savefig("output/Salary\_Distribution\_Onsite.png")  
# plt.show()

