```
%%bash
apt-get install openjdk-8-jdk-headless -qq > /dev/null
[ ! -e "$(basename spark-3.1.2-bin-hadoop2.7.tgz)" ] && wget  http://apache.osuosl.org/spark/
tar xf spark-3.1.2-bin-hadoop2.7.tgz
pip install -q findspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.1.2-bin-hadoop2.7"
```

```
import findspark
findspark.init()
from pyspark.sql import SparkSession, Row
from pyspark.sql.functions import *
from pyspark import SparkContext,SparkConf

# get a spark session.
spark = SparkSession.builder.master("local[*]").getOrCreate()
```

```
import pandas as pd
import numpy as np
import plotly.graph_objs as go
import matplotlib.pyplot as plt
```

```
indeed = pd.read_csv('indeed_jobs.csv')
indeed = indeed.drop(columns='Unnamed: 0', axis=1)

# Remove non-ascii characters
indeed.location.replace({r'[^\x00-\x7F]+':' '}, regex=True, inplace=True)

# Clean location; specify remote
indeed['location_type'] = np.where(indeed['jobtype'].str.find('Remote') >= 0, 'Remote', 'Phys
indeed['location_type'] = np.where(indeed['location'] == "United States", 'Remote', indeed['l

# Clean location; city-state for non-remote
indeed['location_city'] = np.where(indeed['location_type'] == "Physical Location", indeed['lo
indeed['location_state'] = np.where(indeed['location_type'] == "Physical Location", indeed['l
# Clean location: city-state for non-remote: split on first instance of + or ' ' in order to
indeed['location_state'] = np.where(indeed['location_type'] == "Physical Location", indeed['l
indeed['location_state'] = np.where(indeed['location_type'] == "Physical Location", indeed['l

# Salary Cleaning
```

```
# hourly or yearly
indeed['salary_type'] = np.where(indeed['salary'].str.find("hour") >= 0, 'Hourly', indeed['sa
indeed['salary_type'] = np.where(indeed['salary'].str.find("year") >= 0, 'Yearly', indeed['sa

# remove "a year" "an hour"
indeed['salary'] = indeed['salary'].str.split(' a').str[0]

# remove "Up to "
indeed['salary'] = indeed['salary'].str.replace("Up to ","")

# remove other characters
indeed['salary'] = indeed['salary'].str.replace("$","")
indeed['salary'] = indeed['salary'].str.replace(",","")

# split into min and max
indeed[['salary_min','salary_max']] = indeed['salary'].str.split(" - ", expand=True)

#convert hourly number to yearly number
indeed['salary_min'] = np.where(indeed['salary_type'] == 'Hourly', indeed['salary_min'].astyp
indeed['salary_max'] = np.where(indeed['salary_type'] == 'Hourly', indeed['salary_max'].astyp

# get final average or min salary
indeed['salary'] = np.where(indeed['salary_max'].isna(), indeed['salary_min'], (indeed['salar

indeed.drop(['salary_min', 'salary_max'], axis=1, inplace=True)
indeed.drop_duplicates()

indeed.head(50)
```

	title	salary	jobtype	description	location	location_t
0	Data Scientist/Modeler - Remote Based, Teleflora	NaN	Remote	Experience with eCommerce data analysis is a p...	Delano, CA Remote	Rem
1	Data Scientist	NaN	Remote	NaN	Remote Remote	Rem
2	VP, Data Science	NaN	Remote	Experience supporting data scientists across t...	Remote Remote	Rem
3	Data Engineer, Operations Decision Science	NaN	NaN	Define and execute the data engineering roadma...	Atlanta, GA	Phys Loca
4	data scientist - Remote	NaN	Remote	Understands how to break down the data, extrac...	Seattle, WA Remote	Rem
5	Technical Sourcer, Data Science	NaN	Remote	Utilizes Data for reporting and influencing hi...	Remote+8 locations Remote	Rem
6	Data Analyst I	NaN	Remote	Integrate data from multiple data sets, analyz...	Remote+1 location Remote	Rem
7	Data Analyst	NaN	NaN	Contribute to data models and designs for the ...	Los Angeles, CA	Phys Loca
8	Data Scientist	NaN	Remote	Demonstrated professional experience in Tablea...	Cleveland, OH 44143 Remote	Rem
9	Data Science & Analytics	NaN	NaN	NaN	South Jordan, UT 84095	Phys Loca
10	Machine Learning Data Associate	NaN	NaN	This role focuses on language data, primarily ...	Santa Barbara, CA	Phys Loca
11	Data Scientist	NaN	NaN	Partner with data engineers to build pipelines...	Burlingame, CA 94010	Phys Loca
12	Data Product Associate	NaN	NaN	Proactive communication : Serves as a project ...	San Francisco, CA (North Waterfront area)	Phys Loca
				As part of the	San Francisco	

13	Data Science	70000.0	NaN	As part of the Residency program	Francisco, CA 94105+7	Phys

```
simply = pd.read_csv('simplyhired.csv')
simply = simply.drop(columns='Unnamed: 0', axis=1)
# Replace non-ascii chars
simply.location.replace({r'[^\x00-\x7F]+':' '}, regex=True, inplace=True)

# location type
simply['location_type'] = np.where(simply['location'] == 'Remote', 'Remote', 'Physical Locati

# Adjust location columns
simply[['location_city','location_state']] = simply['location'].str.split(", ",expand=True)
simply['location_state'] = simply['location_state'].str.split(' +').str[0]
simply['location_state'] = np.where(simply['location_city'] == 'Remote', 'Remote', simply['lo

# Salary Cleaning
# hourly or yearly
simply['salary_type'] = np.where(simply['salary'].str.find("hour") >= 0, 'Hourly', simply['sa
simply['salary_type'] = np.where(simply['salary'].str.find("year") >= 0, 'Yearly', simply['sa

# remove "a year" "an hour"
simply['salary'] = simply['salary'].str.split(' a').str[0]

# remove other characters
simply['salary'] = simply['salary'].str.replace("$","",regex=False)
simply['salary'] = simply['salary'].str.replace(",","")

# split into min and max
simply[['salary_min','salary_max']] = simply['salary'].str.split(" - ", expand=True)

#convert hourly number to yearly number
simply['salary_min'] = np.where(simply['salary_type'] == 'Hourly', simply['salary_min'].astyp
simply['salary_max'] = np.where(simply['salary_type'] == 'Hourly', simply['salary_max'].astyp

# get final average or min salary
simply['salary'] = np.where(simply['salary_max'].isna(), simply['salary_min'], (simply['salar

simply.drop(['salary_min', 'salary_max'], axis=1, inplace=True)
simply.drop(['salary_type'],axis=1,inplace=True)
```

```
simply.drop_duplicates(inplace=True)
simply.head()
```

	company	description	location	salary	title	location_type	location_city	l
0	ACI Federal™	5+ years data science analyst experience. Crea...	Falls Church, VA	78760.0	Data Science Analyst	Physical Location	Falls Church	
1	Pensa Systems	A solid understanding of data mining & big dat...	Remote	NaN	Data Analyst	Remote	Remote	

```
ssimply=spark.createDataFrame(simply)
ssimply.createOrReplaceTempView('simply')

statesal=spark.sql("""
SELECT location_state, salary from simply
Where location_state != '%Remote%'
order by location_state;
""")



df6=statesal.na.drop("any")
statedf=df6.orderBy('salary')



avgst= statedf.groupby('location_state').avg()
avgstate=avgst.select('location_state',round('avg(salary)',0)).withColumnRenamed('round(avg(s

avgstate.show()
b=avgstate.toPandas()
```

```
+--------------+--------+
|location_state|  salary|
+--------------+--------+
|            AK|115576.0|
|            AZ|115278.0|
|            CA|117694.0|
|            CO|112420.0|
|            CT|120000.0|
|            DC|118140.0|
|            DE| 90000.0|
|            FL| 91326.0|
|            GA|100450.0|
|            HI| 64858.0|
|            IL| 95446.0|
|            IN| 67000.0|
|            KY| 71729.0|
|            MA| 69377.0|
```

```
|            MD| 93210.0|
|            MI|105250.0|
|            MO|100198.0|
|            NC| 76576.0|
|            NJ| 99800.0|
|            NY|107974.0|
+--------------+--------+
only showing top 20 rows
```

```
b.to_csv('AvgStateSalary.csv')
```

```
from pyspark.sql.functions import split, explode, desc
simply1= spark.read.csv("simplyhired.csv",header='true',
                        inferSchema='true')
dfWords1 = simply1.select(explode(split('description', '\\s+')).alias('word')) \
                    .groupBy('word').count().orderBy(desc('word'))
dfWords2 = simply1.select(explode(split('title', '\\s+')).alias('word')) \
                    .groupBy('word').count().orderBy(desc('word'))
```

```
simplycount=dfWords1.orderBy(desc('count')).toPandas()
simplycounttitle=dfWords2.orderBy(desc('count')).toPandas()
```

```
simplycount.head()
simplycounttitle.head()
```

	word	count
0	Data	1947
1	Scientist	895
2	Analyst	828
3	Senior	474
4	Programmer	420

```
simplycounttitle.to_csv('titlecount.csv')
simplycounttitle.to_csv('simplywordcount.csv')
```