BDA Lab

word count

```
# mapper
import sys
for line in sys.stind:
    line = line.strip()
    words = line.split()
    for word in words:
        print("%s/t%s", %(word,1))

# reducer
from operator import itemgetter
import sys
word_count = 0
current_word = None
word = None
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('/t', 1)
    try:
        count = int(count)
    except ValueError:
        continue
    if current_word == word:
        current_count += count
    else:
        if current_word:
            print(f"{current_word}/t{current_count}")
        current_word = word
        current_count = count
```

```
if current_word == word:
    print(f"{current_word}/t{current_count}")
```

average salary

```
#mapper
import sys
for line in sys.stdin:
    fields = line.strip().split()
    if len(fields)!= 3:
        continue
    salary = int(fields[2])
    print("{0}/t{1}" .format(salary,1))

# reducer
import sys
total_sal = 0
employee_count = 0
for line in sys.stdin:
    salary, count = line.strip().split('/t')
    total_sal += int(salary) * int(count)
    employee_count += int(count)
average_sal = total_sal / employee_count
print("Average Salary {0}" .format{average_sal})
```

maximum salary

```
# mapper
import sys
for line in sys.stdin:
    line = line.strip()
    column = line.split()
    salary = column[2]
    print("%s/t%s" %(None, salary))


# reducer
```

```
import sys
max_sal = 0
for line in sys.stdin:
    key, value = line.strip().split('/t', 1)
    try:
        salary = int(value)
    except ValueError:
        continue
    if salary > max_sal:
        max_sal = salary
print(f"Maximum Salary : {max_sal}")
```

printing salary

```
# mapper.py
import sys
import csv
for line in csv.reader(iter(sys.stdin.readline, '')):
    year = line[0]
    sales = line
    print(f"{year}\t{sales}")


# reducer.py
import sys
current_year = None
total_sales = 0
for line in sys.stdin:
    year, sales = line.strip().split('\t')
    if current_year is None:
        current_year = year
    if year == current_year:
        total_sales += int(sales)
    else:
        print(f"{current_year}\t{total_sales}")
        current_year = year
        total_sales = int(sales)
```

```
if current_year is not None:
    print(f"{current_year}\t{total_sales}")
```

inverted

```
# Mapper.py
import sys
def mapper():
    for line in sys.stdin:
        line = line.strip()
        document, words = line.split(" ", 1)
        word_list = words.split()
        for word in word_list:
            print(f"{word}\t{document}")

if __name__ == "__main__":
    mapper()

# Reducer.py
import sys

def reducer():
    current_word = None
    doc_list = []

    for line in sys.stdin:
        line = line.strip()
        word, document = line.split("\t", 1)

        if current_word == word:
            doc_list.append(document)
        else:
            if current_word:
                print(f"{current_word}\t{', '.join(set(doc_list))}")
            doc_list = [document]
            current_word = word
```

```
    if current_word:
        print(f"{current_word}\t{', '.join(set(doc_list))}")

if __name__ == "__main__":
    reducer()


"""
Input.txt

doc1 apple banana
doc2 banana orange
doc3 apple mango
doc4 banana apple
"""
```

Inverted - new

```
# mapper
import sys

def mapper():
    for line in sys.stdin:
        line = line.strip()
        doc, *words = line.split()
        for word in words:
            print(f"{word}\t{doc}")

if __name__ == "__main__":
    mapper()


# reducer
import sys

def reducer():
    current_word, doc_list = None, set()
```

```
    for line in sys.stdin:
        word, doc = line.strip().split("\t")

        if word == current_word:
            doc_list.add(doc)
        else:
            if current_word:
                print(f"{current_word}: {', '.join(sorted(doc_list))}")
            current_word, doc_list = word, {doc}

    if current_word:
        print(f"{current_word}: {', '.join(sorted(doc_list))}")

if __name__ == "__main__":
    reducer()

-----------------------------------------------------------
# input.txt
doc1 apple banana
doc2 banana orange
doc3 apple mango
doc4 banana apple

# mapper output , reducer input
apple   doc1
apple   doc3
apple   doc4
banana    doc1
banana    doc2
banana    doc4
mango    doc3
orange    doc2

# reducer output
apple: doc1, doc3, doc4
banana: doc1, doc2, doc4
```

```
orange: doc2
mango: doc3
```

---

word count using spark shell

```
```

Hierchical

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import scipy.cluster.hierarchy as shc
from sklearn.cluster import AgglomerativeClustering

customer_data = pd.read_csv('hierarchical-clustering.csv')
#customerid, gender, age, income, spending score

customer_data.shape
customer_data.head()

data = customer_data.iloc[:, 3:5].values  #selecting annual income and spe
nding score
data

plt.figure(figsize=(10, 7))
plt.title("Customer Dendrogram")
shc.dendrogram(shc.linkage(data, method='ward'))
plt.show()


cluster = AgglomerativeClustering(n_clusters=5, metric='euclidean', linkag
e='ward')
# affinity = 'euclidean'
labels_ = cluster.fit_predict(data)

plt.figure(figsize=(10, 7))
```

```
plt.scatter(data[:, 0], data[:, 1], c=labels_, cmap='rainbow')
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Clusters of Customers")
plt.show()
```

DBSCAN

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN

data = pd.read_csv('hierarchical-clustering.csv')

data = data.dropna(subset=['Annual Income (k$)','Spending Score (1-10
0)'])

df= data.iloc[:, 3:5].values  #selecting annual income and spending score

plt.scatter(df[:,0], df[:,1], s=10, c="black")
plt.title("Scatter Plot of Data")
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.show()


wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters= i,init = 'k-means++', max_iter= 300, n_init
= 10,random_state=42)
    kmeans.fit(df)
    wcss.append(kmeans.inertia_)


plt.plot(range(1,11), wcss,marker='o',linestyle='--')
```

```
plt.title("The Elbow Method")
plt.xlabel("Number of clusters")
plt.ylabel("WCSS")
plt.grid()
plt.show()



dbscan = DBSCAN(eps=5, min_samples=5)
labels = dbscan.fit_predict(df)
print(np.unique(labels))

unique_labels = np.unique(labels)
colors = ['blue', 'red', 'green', 'brown', 'pink', 'yellow', 'silver']

# Loop over unique labels and assign colors to clusters
for label in unique_labels:
    # For noise points (-1), color them black
    if label == -1:
        plt.scatter(df[labels == label, 0], df[labels == label, 1], s=10, c='black',
label='Noise')
    else:
        # For other clusters, assign colors from the list
        color = colors[label % len(colors)]  # Ensures it works if there are mor
e than 6 clusters
        plt.scatter(df[labels == label, 0], df[labels == label, 1], s=10, c=color, la
bel=f'Cluster {label}')

# Adding labels and legend
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.legend()
plt.show()
```

OPTICS

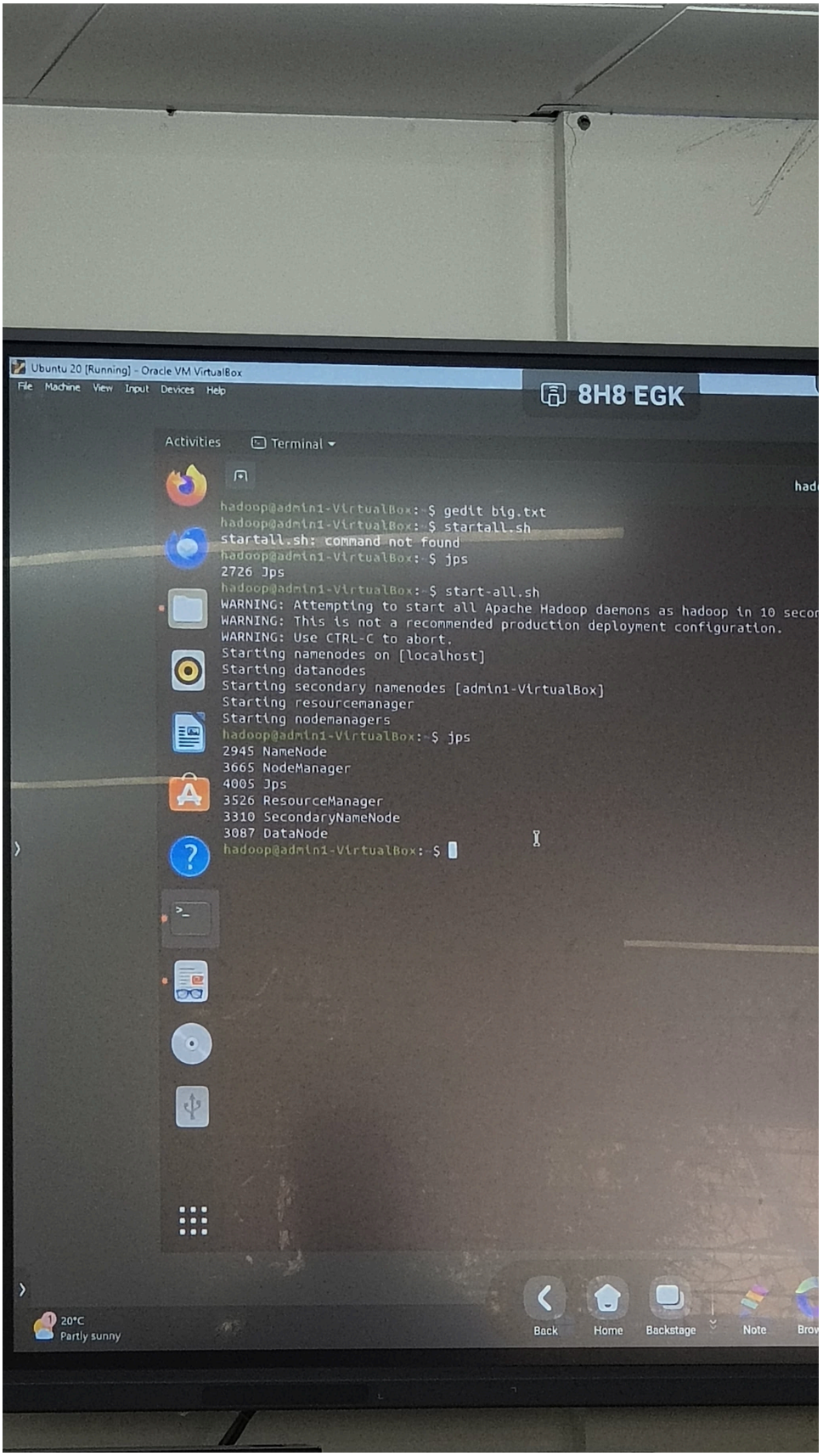
```
from sklearn.datasets import make_blobs
from sklearn.cluster import OPTICS
```

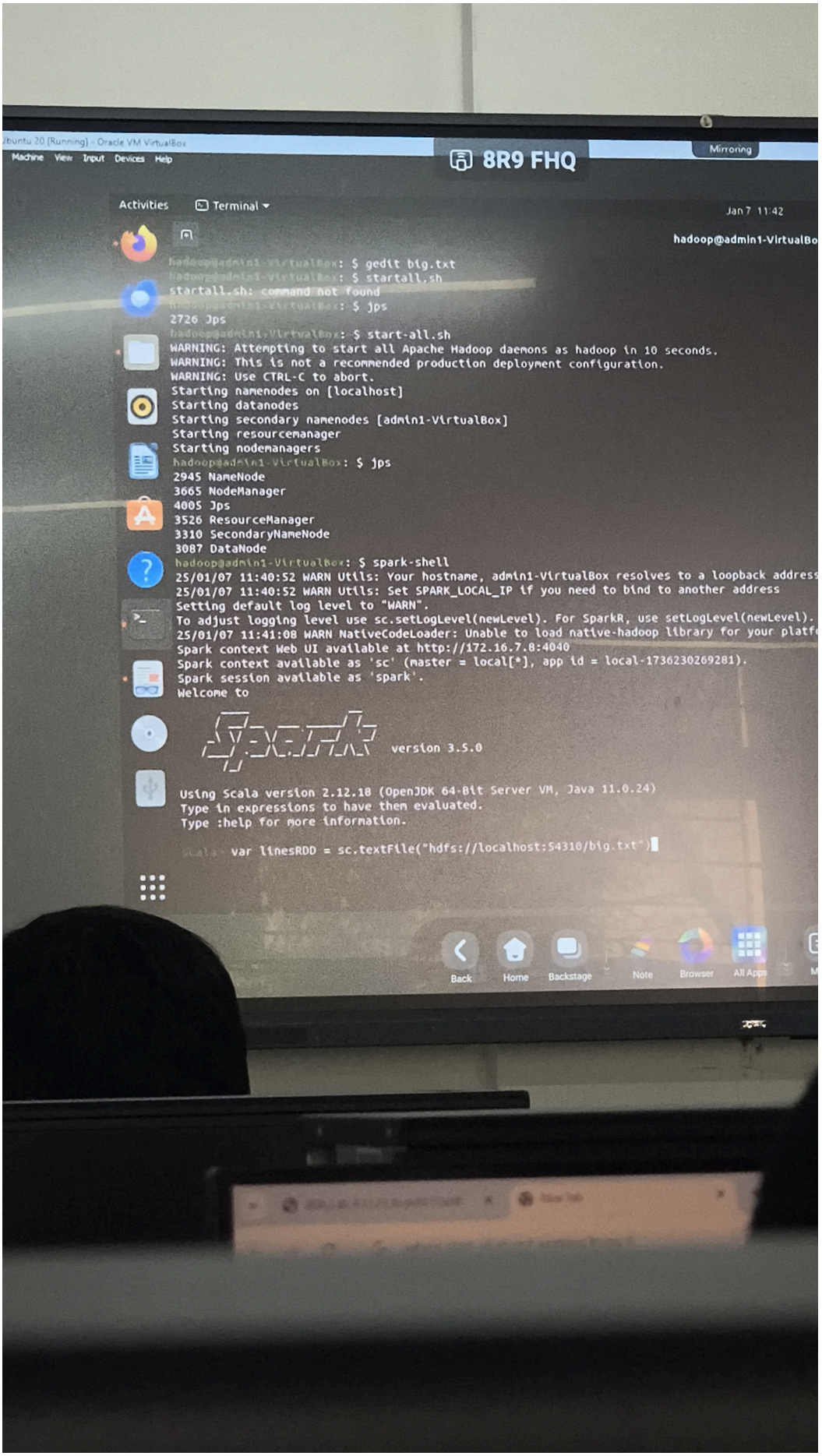
```
import numpy as np
import matplotlib.pyplot as plt

X, y = make_blobs(n_samples=1000, centers=[(3, 3), (7, 7)], n_features=2,
cluster_std=0.5, random_state=42)

# Apply OPTICS clustering
db = OPTICS(max_eps=2.0, min_samples=22, cluster_method='xi', metric
='minkowski').fit(X)
labels = db.labels_
no_clusters = len(np.unique(labels[labels != -1]))  # Exclude noise points
no_noise = np.sum(labels == -1)

print(f'Estimated no. of clusters: {no_clusters}')
print(f'Estimated no. of noise points: {no_noise}')

# Plot clustering results
plt.figure(figsize=(8, 6))
colors = ['blue' if label == 0 else 'red' if label == 1 else 'gray' for label in lab
els]
plt.scatter(X[:, 0], X[:, 1], c=colors, marker="o")
plt.title('OPTICS clustering')
plt.xlabel('Axis X[0]')
plt.ylabel('Axis X[1]')
plt.show()

# Generate reachability plot
plt.figure(figsize=(8, 6))
plt.plot(db.reachability_[db.ordering_])
plt.title('Reachability plot')
plt.xlabel('Sample Index')
plt.ylabel('Reachability Distance')
plt.show()
```

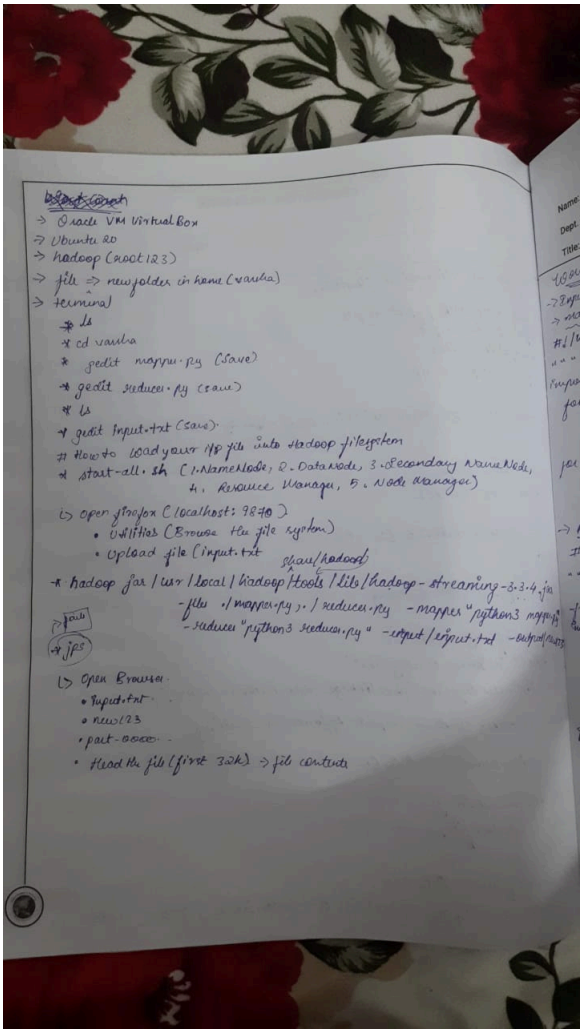
visualisation

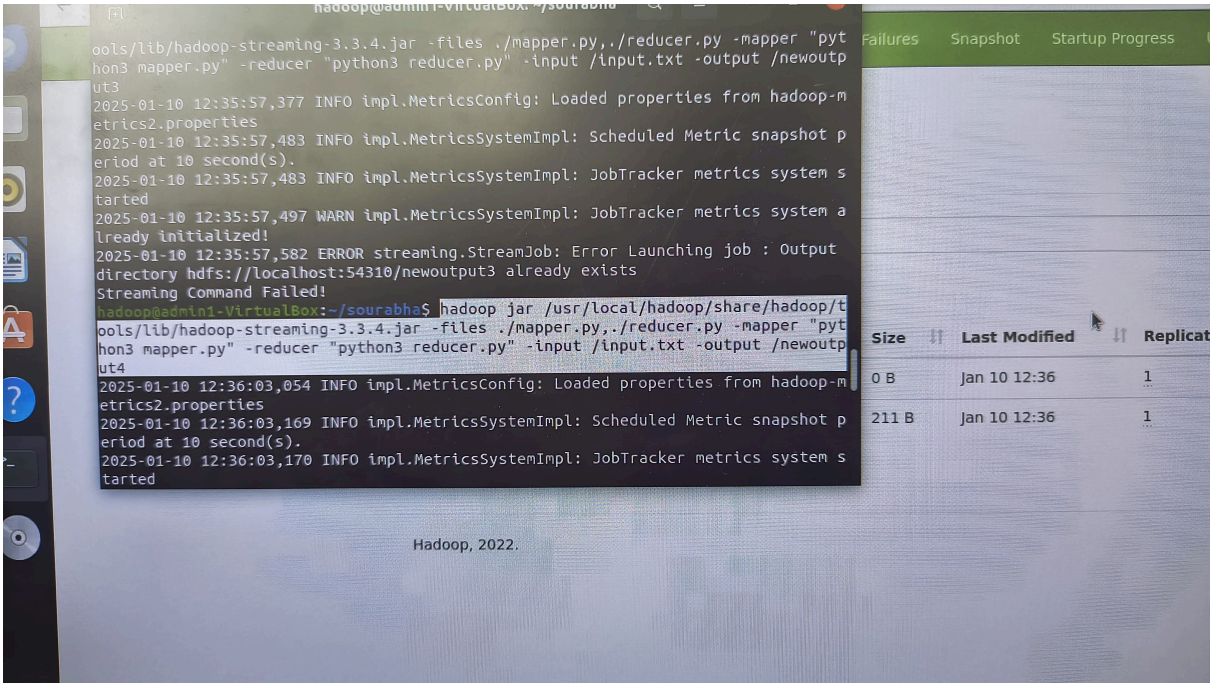
```
import plotly.express as px
df = px.data.iris()
fig = px.line(df, y="sepal_width",)
fig.show()

fig = px.line(df, y="sepal_width", line_group='species')
fig.show()

df = px.data.tips()
# Creating the bar chart
fig = px.bar(df, x='day', y="total_bill")
fig.show()

df = px.data.tips()
# plotting the histogram
fig = px.histogram(df, x="total_bill")
fig.show()


df = px.data.tips()
# plotting the histogram
fig = px.histogram(df, x="total_bill", color='sex', nbins=50, histnorm='perce
nt',barmode='overlay')
fig.show()

df = px.data.tips()
fig = px.pie(df, values="total_bill", names="day")
fig.show()

df = px.data.tips()
fig = px.box(df, x="day", y="tip")
fig.show()

df = px.data.tips()
# plotting the violin plot
fig = px.violin(df, x="day", y="tip")
fig.show()
```

```
df = px.data.tips()
fig = px.scatter_3d(df, x="total_bill", y="sex", z="tip")
fig.show()
```


Ubuntu 20 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
8H8 EGK
Activities
Terminal
hadoop@admin1-VirtualBox:~$ gedit big.txt
hadoop@admin1-VirtualBox:~$ startall.sh
startall.sh: command not found
hadoop@admin1-VirtualBox:~$ jps
2726 Jps
hadoop@admin1-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 secon
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [admin1-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hadoop@admin1-VirtualBox:~$ jps
2945 NameNode
3665 NodeManager
4005 Jps
3526 ResourceManager
3310 SecondaryNameNode
3087 DataNode
hadoop@admin1-VirtualBox:~$
20°C
Partly sunny
Back
Home
Backstage
Note


Machine View Input Devices Help
8R9 FHQ
Mirroring
Activities
Terminal
Jan 7 11:42
hadoop@admin1-VirtualBo
hadoop@admin1-VirtualBox: $ gedit big.txt
hadoop@admin1-VirtualBox: $ startall.sh
startall.sh: command not found
hadoop@admin1-VirtualBox: $ jps
2726 Jps
hadoop@admin1-VirtualBox: $ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [admin1-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hadoop@admin1-VirtualBox: $ jps
2945 NameNode
3665 NodeManager
4005 Jps
3526 ResourceManager
3310 SecondaryNameNode
3087 DataNode
hadoop@admin1-VirtualBox: $ spark-shell
25/01/07 11:40:52 WARN Utils: Your hostname, admin1-VirtualBox resolves to a loopback address
25/01/07 11:40:52 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/01/07 11:41:08 WARN NativeCodeLoader: Unable to load native-hadoop library for your platf
Spark context Web UI available at http://172.16.7.8:4040
Spark context available as 'sc' (master = local[*], app id = local-1736230269281).
Spark session available as 'spark'.
Welcome to
version 3.5.0
Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.24)
Type in expressions to have them evaluated.
Type :help for more information.
scala> var linesRDD = sc.textFile("hdfs://localhost:54310/big.txt")
Back
Home
Backstage
Note
Browser
All Apps


Oracle VM VirtualBox
Ubuntu 20
hadoop (root123)
file => new folder in home (vaisha)
terminal
ls
cd vaisha
gedit mapper.py (save)
gedit reducer.py (save)
ls
gedit input.txt (save)
How to load your txt file into Hadoop filesystem
start-all.sh (1.NameNode, 2.DataNode, 3.SecondaryNameNode, 4. Resource Manager, 5. Node Manager)
open firefox (localhost: 9870)
Utilities (Browse the file system)
Upload file (input.txt
share/hadoop/
hadoop jar /usr/local/hadoop/tools/lib/hadoop-streaming-3.3.4.jar
-file ~/mapper.py ~/reducer.py -mapper "python3 mapper.py"
-reducer "python3 reducer.py" -input /input.txt
jps
jps
Open Browser
input.txt
new123
part-0000
Read the file (first 32k) => file contents


ools/lib/hadoop-streaming-3.3.4.jar -files ./mapper.py,./reducer.py -mapper "pyt
hon3 mapper.py" -reducer "python3 reducer.py" -input /input.txt -output /newoutp
ut3
2025-01-10 12:35:57,377 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-01-10 12:35:57,483 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
eriod at 10 second(s).
2025-01-10 12:35:57,483 INFO impl.MetricsSystemImpl: JobTracker metrics system s
tarted
2025-01-10 12:35:57,497 WARN impl.MetricsSystemImpl: JobTracker metrics system a
lready initialized!
2025-01-10 12:35:57,582 ERROR streaming.StreamJob: Error Launching job : Output
directory hdfs://localhost:54310/newoutput3 already exists
Streaming Command Failed!
hadoop@admin1-VirtualBox:~/sourabha$ hadoop jar /usr/local/hadoop/share/hadoop/t
ools/lib/hadoop-streaming-3.3.4.jar -files ./mapper.py,./reducer.py -mapper "pyt
hon3 mapper.py" -reducer "python3 reducer.py" -input /input.txt -output /newoutp
ut4
2025-01-10 12:36:03,054 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-01-10 12:36:03,169 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
eriod at 10 second(s).
2025-01-10 12:36:03,170 INFO impl.MetricsSystemImpl: JobTracker metrics system s
tarted
Failures
Snapshot
Startup Progress
Size
Last Modified
Replicat
0 B
Jan 10 12:36
1
211 B
Jan 10 12:36
1
Hadoop, 2022.