EXPERIMENT – 6

WRITE A PROGRAM IN PYTHON TO PERFORM AUDIO SIGNAL PROCESSING

Aim: - The aim of this program is to perform Audio signal processing using python.

Procedure:

```
from pydub import AudioSegment

# import the audio file

wav_file = AudioSegment.from_file(file="sample-6s.wav",format="wav")

# data type for the file

print(type(wav_file))

#  To find frame rate of song/file

print(wav_file.frame_rate)

print(wav_file.channels)

print(wav_file.sample_width)

print(wav_file.max)

print(len(wav_file))

wav_file_new = wav_file.set_frame_rate(50)

print(wav_file_new.frame_rate)
```

Output:

```
<class 'pydub.audio_segment.AudioSegment'>
44100
2
2
22298
6391
50
```

Result: - We successfully executed the audio file signal processing.

EXPERIMENT – 7

WRITE A PROGRAM IN PYTHON TO PERFORM VIDEO SIGNAL PROCESSING

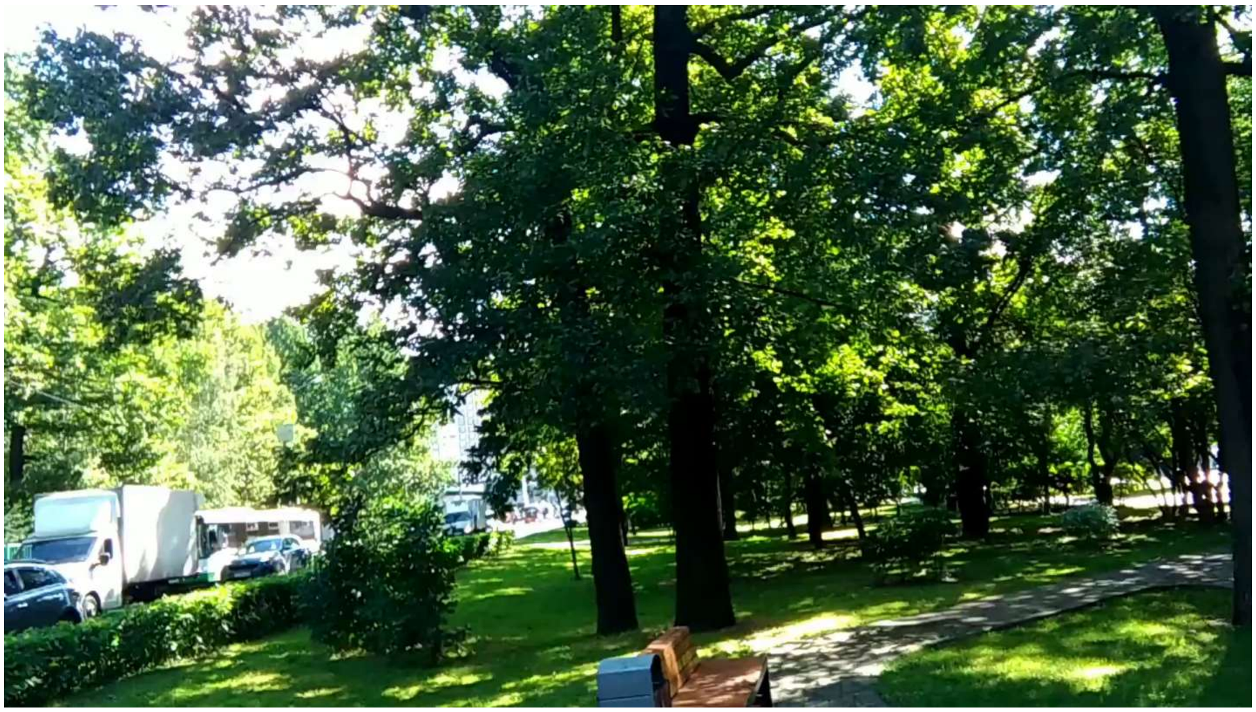
Aim: - The aim of this program is to perform video signal processing using python opencv library.

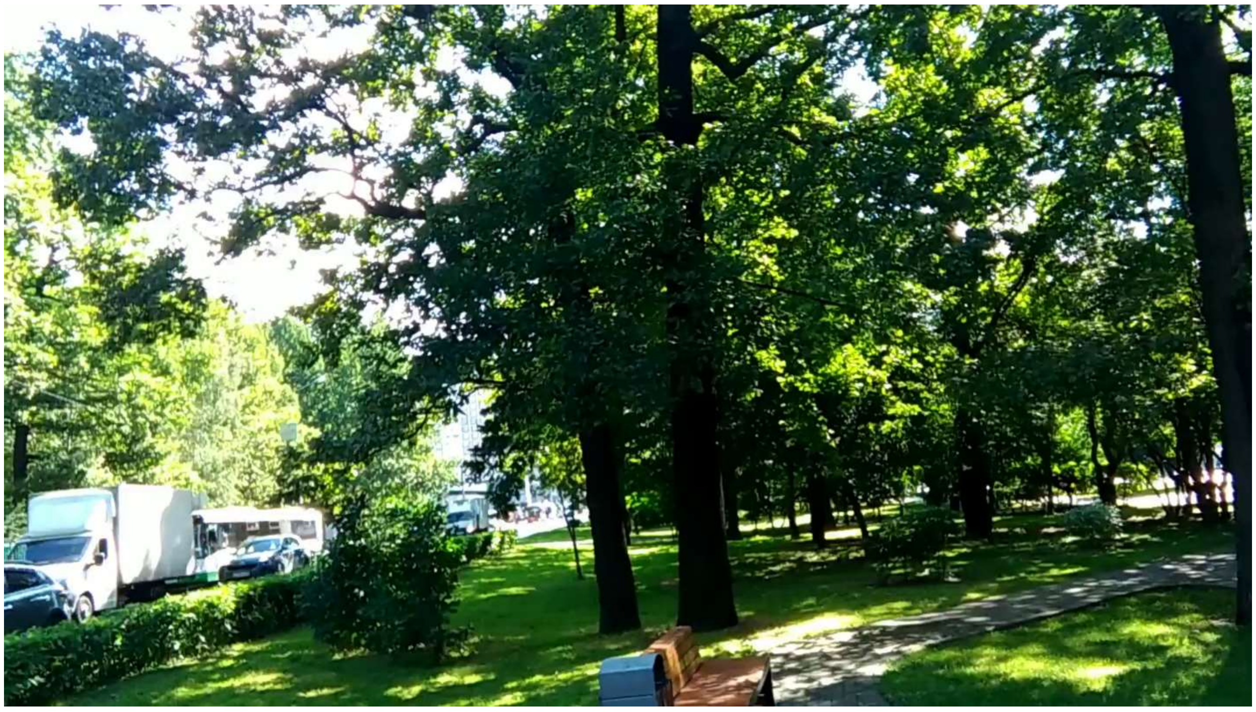
Procedure:

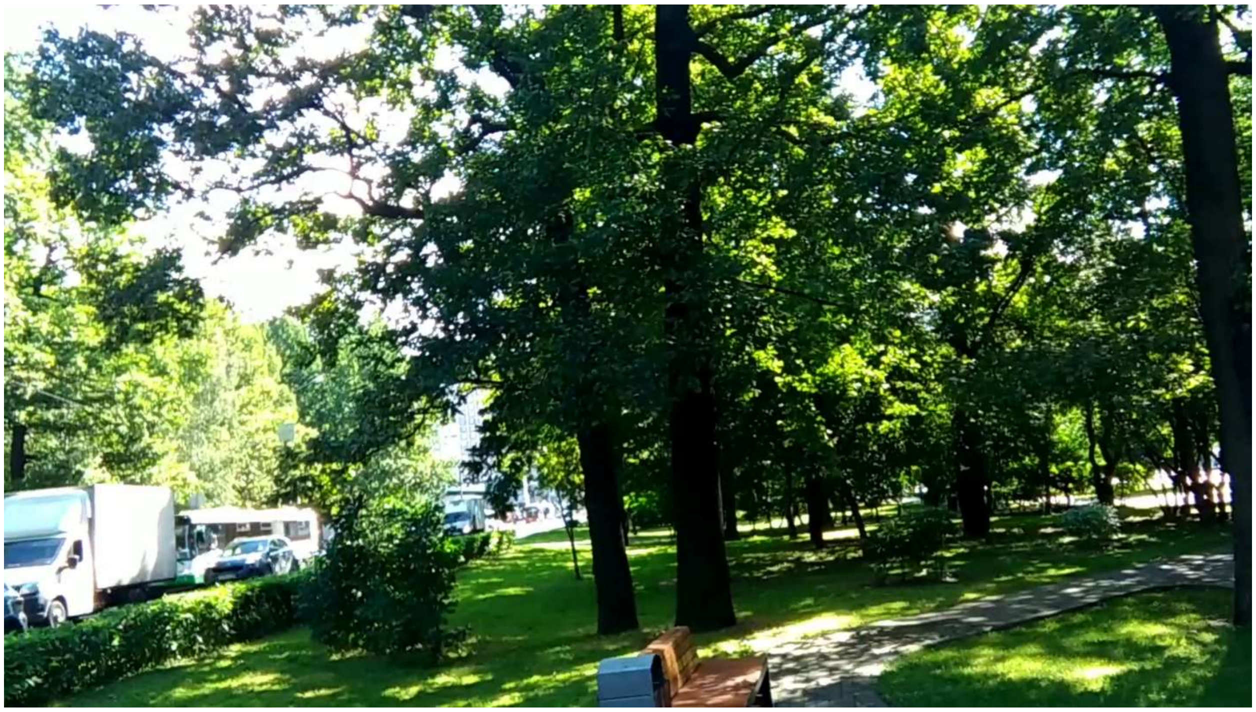
```
# importing libraries
import cv2
import numpy as np
# Create a VideoCapture object and read from input file
cap = cv2.VideoCapture('sample-5s.mp4')
# Check if camera opened successfully
if (cap.isOpened()== False):
  print("Error opening video file")
# Read until video is completed
while(cap.isOpened()):
# Capture frame-by-frame
  ret, frame = cap.read()
  if ret == True:
  # Display the resulting frame
    cv2_imshow(frame)
    # Press Q on keyboard to exit
    if cv2.waitKey(25) & 0xFF == ord("q"):
      break
# Break the loop
  else:
    break
# When everything done, release # the video capture object
cap.release()
```

```
# Closes all the frames
cv2.destroyAllWindows()
```

<u>Output:</u>







Result: **-** We successfully executed video signal processing program.

EXPERIMENT – 8

WRITE A PROGRA, THAT DEMONSTRATE CREATION OF RDD FROM TEXT FILE

Aim: - The aim of this program is the creation of rdd from txt file using pyspark.

Procedure:

```
import findspark
findspark.init()


from pyspark import SparkContext
from pyspark.sql import SparkSession


spark = SparkSession.builder.master('local').appName('Exercise').getOrCreate()
rdd = spark.sparkContext.textFile('sample.txt')
print(rdd.take(2))
```

Output:

```
['Hello World!', 'This is a PySpark Program.']
```

Result: - We successfully created an rdd from text file.

EXPERIMENT – 9

WRITE A PROGRA, THAT DEMONSTRATE CREATION OF RDD FROM CSV FILE

Aim: - The aim of this program is the creation of rdd from csv file using pyspark.

Procedure:

```
import findspark
findspark.init()


from pyspark.sql import SparkSession


spark = SparkSession.builder.master('local').appName('CSV_FILE').getOrCreate()
rdd = spark.sparkContext.textFile('Sample.csv')
print(rdd.take(2))
```

Output:

```
['1,"Eldon Base for stackable storage shelf, platinum",Muhammed MacIntyre,3,-
213.25,38.94,35,Nunavut,Storage & Organization,0.8', '2,"1.7 Cubic Foot Compact ""Cube""
Office Refrigerators",Barry French,293,457.81,208.16,68.02,Nunavut,Appliances,0.58']
```

Result: - We successfully created an rdd from csv file.

EXPERIMENT – 10

CREATE A PYSPARK PROGRAM THAT READ THE TEXT FILE AND PERFORM A WORD COUNT ON THE CONTENTS

Aim: - The aim of this program is to read the text file and perform a word count on the contents.

Procedure:

```
import findspark

findspark.init()

from pyspark import SparkContext

from pyspark.sql import SparkSession


spark = SparkSession.builder.master('local').appName('Exercise').getOrCreate()

rdd = spark.sparkContext.textFile('sample.txt')

word_counts = rdd.flatMap(lambda line: line.split(" ")) \

                .map(lambda word: (word, 1)) \

                .reduceByKey(lambda a, b: a + b)

for word, count in word_counts.collect():

    print(f"{word}: {count}")
```

Output:

```
Data: 2
Science: 2
Interview,: 1
Hello: 1
Sir,: 1
I: 2
am: 2
a: 2
coder: 1
and: 1
Hello,: 1
Student: 1
```

Result: - We successfully executed program to read text file and perform word count.

EXPERIMENT – 11

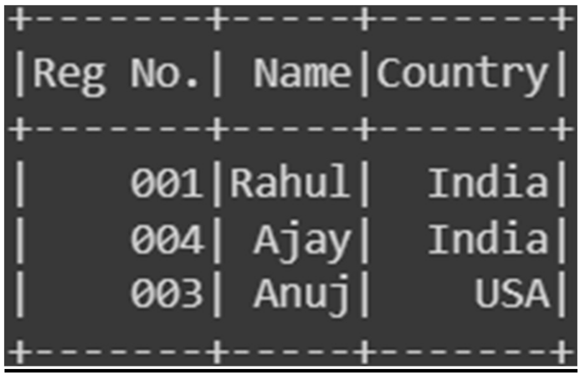
WRITE A PROGRAM IS TO DEMONSTRATE CREATION OF DATAFRAME FROM LIST

Aim: - The aim of this program is to demonstrate creation of dataframe from list.

Procedure:

```
import findspark

findspark.init()

from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

data = [('001','Rahul','India'),('004','Ajay','India'),('003','Anuj','USA')]

col = ['Reg No.','Name','Country']

df = spark.createDataFrame(data,schema=col)

df.show()
```

Output:



```
+-------+-----+-------+
|Reg No.| Name|Country|
+-------+-----+-------+
|    001|Rahul|  India|
|    004| Ajay|  India|
|    003| Anuj|    USA|
+-------+-----+-------+
```

Result: - We successfully created dataframe from list.

EXPERIMENT – 12

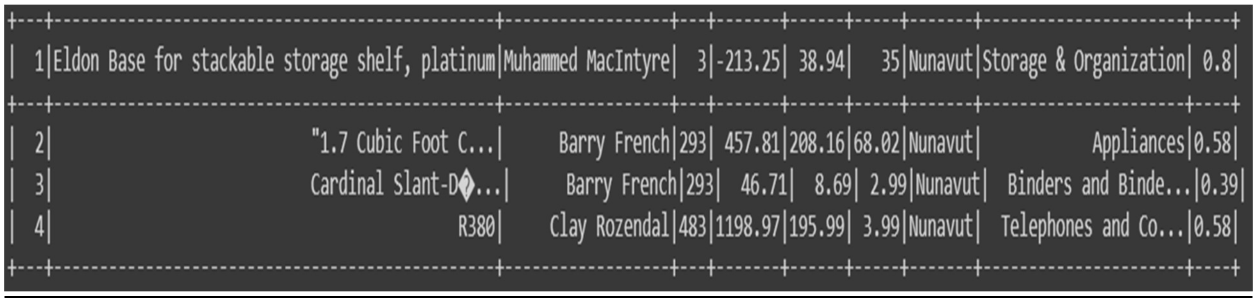
WRITE A PROGRAM THAT DEMONSTRATES CREATION OF DATAFRAME FROM CSV FILE

Aim: - The aim of this program is to create a dataframe from csv file.

Procedure:

```
import findspark

findspark.init()

from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

df_csv = spark.read.csv('Sample.csv',header='true',inferSchema='true')

df_csv.show()
```

Output:



```
+---+------------------------------------------------+------------------+---+-------+------+-----+-------+----------------------+----+
|  1|Eldon Base for stackable storage shelf, platinum|Muhammed MacIntyre|  3|-213.25| 38.94|   35|Nunavut|Storage & Organization| 0.8|
+---+------------------------------------------------+------------------+---+-------+------+-----+-------+----------------------+----+
|  2|                            "1.7 Cubic Foot C...|      Barry French|293| 457.81|208.16|68.02|Nunavut|            Appliances|0.58|
|  3|                            Cardinal Slant-D�...|      Barry French|293|  46.71|  8.69| 2.99|Nunavut|  Binders and Binde...|0.39|
|  4|                                            R380|     Clay Rozendal|483|1198.97|195.99| 3.99|Nunavut|  Telephones and Co...|0.58|
+---+------------------------------------------------+------------------+---+-------+------+-----+-------+----------------------+----+
```

Result: - We successfully created a dataframe from csv file.

EXPERIMENT – 13

WRITE A PROGRAM THAT DEMONSTRATE THE USE OF ORDERBY( ) AND SORT ( ) FUNCTION.

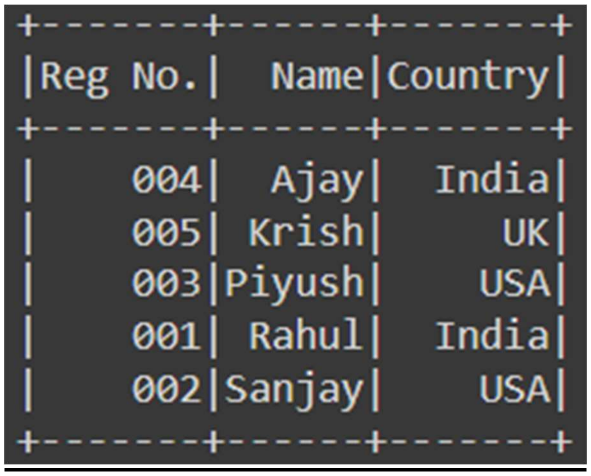
Aim: - The aim of this program is to demonstrate the use of orderBy() and sort() function.

Procedure:

```
import findspark

findspark.init()

from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

data =
[('001','Rahul','India'),('004','Ajay','India'),('003','Piyush','USA'),('002','Sanjay','USA'),('005','Kris
h','UK')]

col = ['Reg No.','Name','Country']

df = spark.createDataFrame(data,schema=col)

# by using orderBy()

df.orderBy('Name','Country').show()

# by using sort()

df.sort('Country').show()
```

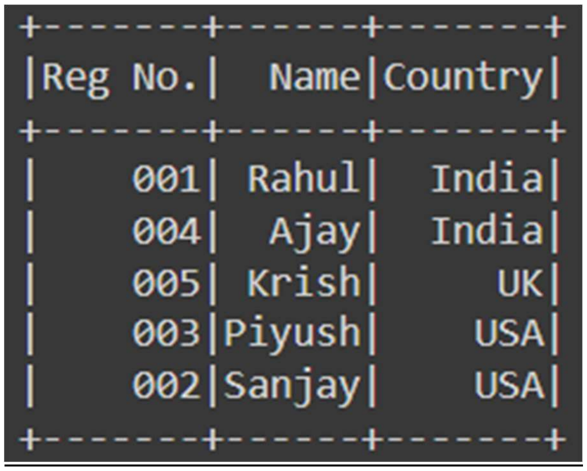
Output:

Output of orderBy() function



```
+-------+------+-------+
|Reg No.|  Name|Country|
+-------+------+-------+
|    004|  Ajay|  India|
|    005| Krish|     UK|
|    003|Piyush|    USA|
|    001| Rahul|  India|
|    002|Sanjay|    USA|
+-------+------+-------+
```

Output of sort() function



```
+-------+------+-------+
|Reg No.|  Name|Country|
+-------+------+-------+
|    001| Rahul|  India|
|    004|  Ajay|  India|
|    005| Krish|     UK|
|    003|Piyush|    USA|
|    002|Sanjay|    USA|
+-------+------+-------+
```

Result: - We successfully demonstrate the use of orderBy() and sort() function on dataframe.

EXPERIMENT – 14

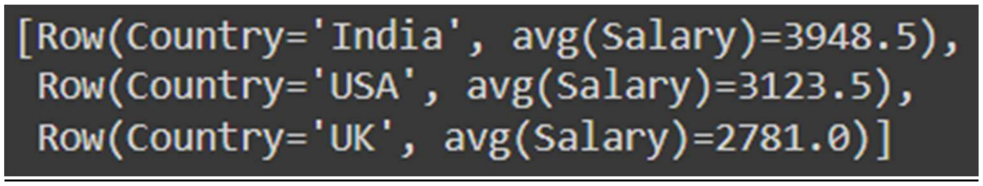
WRITE A PROGRAM THAT DEMONSTRATE THE USE OF GROUPBY ( ) FUNCTION.

Aim: - The aim of this program is to demonstrate the use of groupBy() function.

Procedure:

```
import findspark

findspark.init()

from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

data =
[('Rohit','India',4000),('Rahul','India',3897),('Amit','USA',2569),('Sanjay','USA',3678),('Anmol','UK',3781),('Anuj','UK',1781)]

col = ['Emp_Name','Country','Salary']

df = spark.createDataFrame(data,schema=col)

df.groupBy('Country').agg({'Salary': 'mean'}).collect()
```

Output:



```
[Row(Country='India', avg(Salary)=3948.5),
 Row(Country='USA', avg(Salary)=3123.5),
 Row(Country='UK', avg(Salary)=2781.0)]
```

Result: - We successfully demonstrate the use of groupBy() function on dataframe.

EXPERIMENT – 15

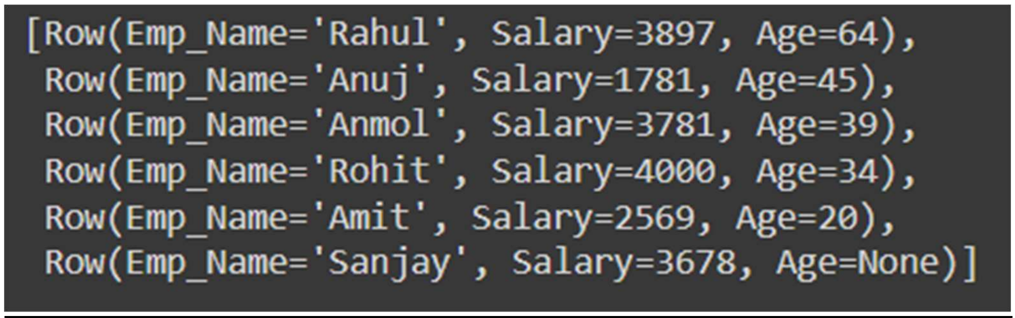
WRITE A PROGRAM THAT DEMONSTRATE THE USE OF JOIN ( ) FUNCTION.

Aim: - The aim of this program is to demonstrate the use of join() function.

Procedure:

```
import findspark

findspark.init()

from pyspark.sql import SparkSession

from pyspark.sql.functions import desc

spark = SparkSession.builder.getOrCreate()

data =
[('Rohit','India',4000),('Rahul','India',3897),('Amit','USA',2569),('Sanjay','USA',3678),('Anmol','
UK',3781),('Anuj','UK',1781)]

col = ['Emp_Name','Country','Salary']

# first dataframe

df = spark.createDataFrame(data,schema=col)

data2 =
[('Rohit','India',34),('Rahul','India',64),('Amit','USA',20),('Sanjay','USA',None),('Anmol','UK',39),
('Anuj','UK',45)]

col2 = ['Emp_Name','Country','Age']

# second dataframe

df2 = spark.createDataFrame(data2,schema=col2)

df.join(df2, 'Emp_Name', 'inner').select('Emp_Name','Salary', 'Age').sort(desc("Age")).collect()
```

Output:



```
[Row(Emp_Name='Rahul', Salary=3897, Age=64),
 Row(Emp_Name='Anuj', Salary=1781, Age=45),
 Row(Emp_Name='Anmol', Salary=3781, Age=39),
 Row(Emp_Name='Rohit', Salary=4000, Age=34),
 Row(Emp_Name='Amit', Salary=2569, Age=20),
 Row(Emp_Name='Sanjay', Salary=3678, Age=None)]
```

Result: - We successfully demonstrate the use of join() function on two dataframes.

EXPERIMENT – 16

WRITE A PROGRAM THAT DEMONSTRATE THE USE OF TRANSFORM ( ) FUNCTION.

Aim: - The aim of this program is to demonstrate the use of transform() function.

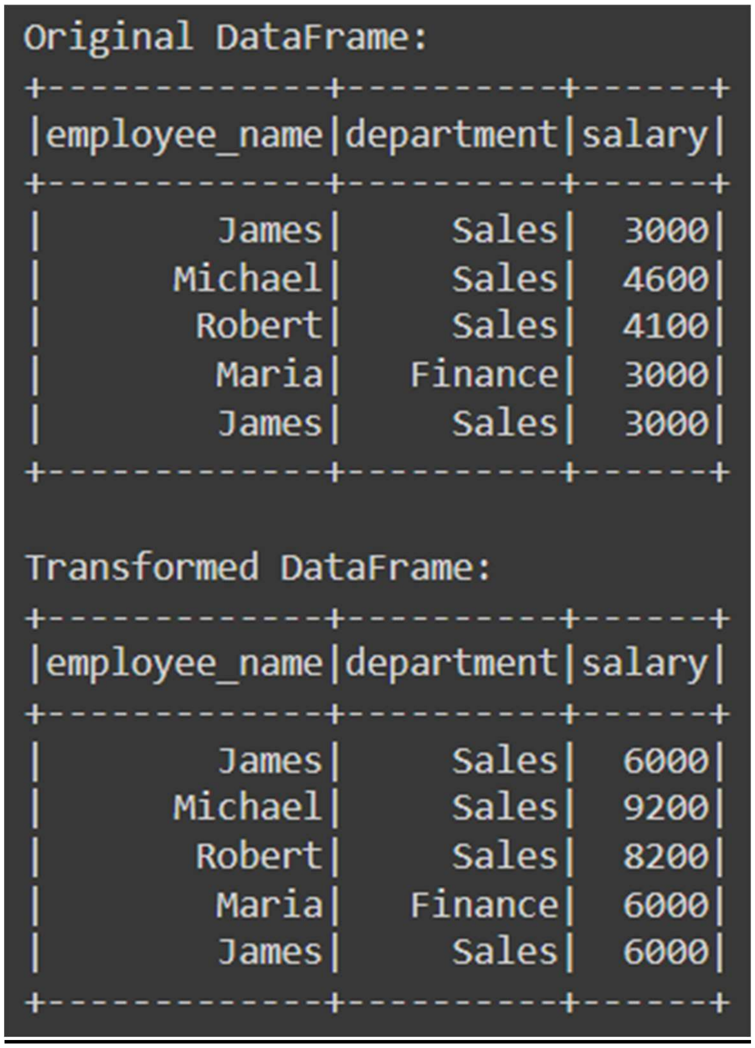
Procedure:

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col

spark = SparkSession.builder.appName("transformExample").getOrCreate()

# Create a sample DataFrame
data = [("James", "Sales", 3000),
        ("Michael", "Sales", 4600),
        ("Robert", "Sales", 4100),
        ("Maria", "Finance", 3000),
        ("James", "Sales", 3000)]

columns= ["employee_name", "department", "salary"]
df = spark.createDataFrame(data = data, schema = columns)

print("Original DataFrame:")
df.show()

def transform_salary(df):
    return df.withColumn("salary", col("salary") * 2)

df_transformed = df.transform(transform_salary)
```

print("Transformed DataFrame:")

df_transformed.show()

Output:



```
Original DataFrame:
+-------------+----------+------+
|employee_name|department|salary|
+-------------+----------+------+
|        James|     Sales|  3000|
|      Michael|     Sales|  4600|
|       Robert|     Sales|  4100|
|        Maria|   Finance|  3000|
|        James|     Sales|  3000|
+-------------+----------+------+

Transformed DataFrame:
+-------------+----------+------+
|employee_name|department|salary|
+-------------+----------+------+
|        James|     Sales|  6000|
|      Michael|     Sales|  9200|
|       Robert|     Sales|  8200|
|        Maria|   Finance|  6000|
|        James|     Sales|  6000|
+-------------+----------+------+
```

Result: - We successfully demonstrate the transform() function on dataframe.