**Maitrey Prajapati - Programming Assignment 3**

**Input file** : I changed the format of xlsx file to csv because pyspark doesn’t have a function to load excel file. And after loading the file in pandas and then creating spark dataframe gave error that there were multiple data types in a single column so it can’t be loaded into rdd.

**Output** :

[('Average Temperature (F)', 37.946348928727595)]

[('Average High Temperature (F)', 47.190852575488464)]

[('Average Low Temperature (F)', 28.799822301199466)]

[('Average Precipitation (in)', 34.4720404040404)]

**Program** :

from pyspark import SparkContext

from pyspark.sql import SQLContext

from pyspark.sql.types import IntegerType,FloatType

def calculate\_total(row\_dict):

tmp\_val = float(row\_dict['ANNUAL\xa0'])

cities = int(row\_dict['# CITIES\xa0'])

return (row\_dict['Alberta'],(tmp\_val\*cities,cities))

def calculate\_total\_temp\_cities(row1,row2):

return(row1[0]+row2[0],row1[1]+row2[1] )

def temp\_calc(row):

return(row[0],row[1][0]/row[1][1])

sc = SparkContext.getOrCreate()

sqlContext = SQLContext(sc)

df = sqlContext.read.format('csv').options(header=True).load('data.csv')

data\_df = df.dropna()

avg\_tmp = data\_df.filter(data\_df['Alberta'] == 'Average Temperature (F)').rdd.map(lambda row: row.asDict())

avg\_high = data\_df.filter(data\_df['Alberta'] == 'Average High Temperature (F)').rdd.map(lambda row: row.asDict())

avg\_low = data\_df.filter(data\_df['Alberta'] == 'Average Low Temperature (F)').rdd.map(lambda row: row.asDict())

avg\_pcp = data\_df.filter(data\_df['Alberta'] == 'Average Precipitation (in)').rdd.map(lambda row: row.asDict())

avg\_tmp = avg\_tmp.map(calculate\_total)

avg\_tmp =avg\_tmp.reduceByKey(calculate\_total\_temp\_cities)

avg\_tmp = avg\_tmp.map(temp\_calc)

avg\_tmp.collect()

avg\_high = avg\_high.map(calculate\_total)

avg\_high =avg\_high.reduceByKey(calculate\_total\_temp\_cities)

avg\_high = avg\_high.map(temp\_calc)

avg\_high.collect()

avg\_low = avg\_low.map(calculate\_total)

avg\_low =avg\_low.reduceByKey(calculate\_total\_temp\_cities)

avg\_low = avg\_low.map(temp\_calc)

avg\_low.collect()

avg\_pcp = avg\_pcp.map(calculate\_total)

avg\_pcp =avg\_pcp.reduceByKey(calculate\_total\_temp\_cities)

avg\_pcp = avg\_pcp.map(temp\_calc)

avg\_pcp.collect()

**Answers:**

'Average Temperature (F)' : 37.946348928727595

'Average High Temperature (F)' : 47.190852575488464

'Average Low Temperature (F)' : 28.799822301199466

'Average Precipitation (in)' : 34.4720404040404