venkatsv@hadoop-gate-0:~$ hdfs dfs -ls /data/weather/

venkatsv@hadoop-gate-0:~$ pyspark

>>> from pyspark.sql.functions import col, split

>>> from pyspark.sql.types import \*

>>> from \_\_future\_\_ import division

>>> file\_path1 = "/data/weather/2010"

>>> file\_path2 = "/data/weather/2011"

>>> file\_path3 = "/data/weather/2012"

>>> file\_path4 = "/data/weather/2013"

>>> file\_path5 = "/data/weather/2014"

>>> file\_path6 = "/data/weather/2015"

>>> file\_path7 = "/data/weather/2016"

>>> file\_path8 = "/data/weather/2017"

>>> file\_path9 = "/data/weather/2018"

>>> file\_path10 = "/data/weather/2019"

>>> inTextData1 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path1)

20/04/12 20:19:05 WARN SharedInMemoryCache: Evicting cached table partition meta data from memory due to size constraints (spark.sql.hive.filesourcePartitionFile CacheSize = 262144000 bytes). This may impact query planning performance.

>>> inTextData2 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path2)

>>> inTextData3 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path3)

>>> inTextData4 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path4)

>>> inTextData5 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path5)

>>> inTextData6 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path6)

>>> inTextData7 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path7)

20/04/12 20:21:04 WARN SharedInMemoryCache: Evicting cached table partition meta data from memory due to size constraints (spark.sql.hive.filesourcePartitionFile CacheSize = 262144000 bytes). This may impact query planning performance.

>>> inTextData8 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path8)

>>> inTextData9 = spark.read.format("csv").option("header", "true").option("deli miter","\t").load(file\_path9)

>>> inTextData10 = spark.read.format("csv").option("header", "true").option("del imiter","\t").load(file\_path10)

>>> inTextData = inTextData1.union(inTextData2).union(inTextData3).union(inTextD ata4).union(inTextData5).distinct()inTextData6.union(inTextData7).union(inTextData8).union(inTextD ata9).union(inTextData10).distinct()

>>> name\_list = inTextData.schema.names

>>> name\_list = str(name\_list).strip("['']").split(' ')

>>> names = []

>>> for item in name\_list:

... if len(item)>0:

... names.append(item)

...

>>> rdd1 = inTextData.rdd

>>> rdd2 = rdd1.map(lambda x: str(x).split('=')[1])

>>> rdd3 = rdd2.map(lambda x: ' '.join(x.split()))

>>> rdd4 = rdd3.map(lambda x: x[2:-2])

>>> rdd4.saveAsTextFile('/user/venkatsv/assignmenttest3'+'temp')

>>> newInData = spark.read.csv('/user/venkatsv/assignmenttest3temp',header=False,sep=' ')

>>> cleanData = newInData.drop('\_c1','\_c4','\_c6','\_c8','\_c10','\_c12','\_c14')

>>> cleanData = cleanData.withColumnRenamed('\_c0','STN').withColumnRenamed('\_c2','YEARMODA')\

... .withColumnRenamed('\_c3','TEMP').withColumnRenamed('\_c5','DEWP')\

... .withColumnRenamed('\_c7','SLP').withColumnRenamed('\_c9','STP')\

... .withColumnRenamed('\_c11','VISIB').withColumnRenamed('\_c13','WDSP')\

... .withColumnRenamed('\_c15','MXSPD').withColumnRenamed('\_c16','GUST')\

... .withColumnRenamed('\_c17','MAX').withColumnRenamed('\_c18','MIN')\

... .withColumnRenamed('\_c19','PRCP').withColumnRenamed('\_c20','SNDP')\

... .withColumnRenamed('\_c21','FRSHTT')

>>> cleanData1=cleanData.withColumn("MIN", split(col("MIN"), "\\\*").getItem(0)).withColumn("col3", split(col("MIN"), "\\\*").getItem(1))

>>> cleanData1=cleanData1.withColumn("MAX", split(col("MAX"), "\\\*").getItem(0)).withColumn("col3", split(col("MAX"), "\\\*").getItem(1))

>>> cleanData1=cleanData1.drop('col2','col3')

>>> cleanData1 = cleanData1.withColumn("MAX\_NEW", cleanData1["MAX"].cast("double"))

>>> cleanData1 = cleanData1.withColumn("MIN\_NEW", cleanData1["MIN"].cast("double"))

>>> cleanData1=cleanData1.drop('MAX','MIN')

>>> cleanData1=cleanData1.withColumnRenamed("MAX\_NEW","MAX")

>>> cleanData1=cleanData1.withColumnRenamed("MIN\_NEW","MIN")

>>> cleanData1.createOrReplaceTempView("cleanData")

>>> cleanData1=cleanData1.withColumn("PRCP", split(col("PRCP"), "[A-Z]").getItem(0)).withColumn("P1", split(col("PRCP"), "[A-Z]").getItem(1))

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2010') and LEFT(YEARMODA,4) ='2010' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where Min in (Select MIN(MIN) from cleanData where Min <>9999.9 and LEFT(YEARMODA,4)='2010') and LEFT(YEARMODA,4) ='2010' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2011') and LEFT(YEARMODA,4) ='2011' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where MIN in (Select MIN(MIN) from cleanData where MIN <>9999.9 and LEFT(YEARMODA,4)='2011') and LEFT(YEARMODA,4) ='2011' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2012') and LEFT(YEARMODA,4) ='2012' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where MIN in (Select MIN(MIN) from cleanData where MIN <>9999.9 and LEFT(YEARMODA,4)='2012') and LEFT(YEARMODA,4) ='2012' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2013') and LEFT(YEARMODA,4) ='2013' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where MIN in (Select MIN(MIN) from cleanData where MIN <>9999.9 and LEFT(YEARMODA,4)='2013') and LEFT(YEARMODA,4) ='2013' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2014') and LEFT(YEARMODA,4) ='2014' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where MIN in (Select MIN(MIN) from cleanData where MIN <>9999.9 and LEFT(YEARMODA,4)='2014') and LEFT(YEARMODA,4) ='2014' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE,MAX from cleanData where MAX = (Select MAX(MAX) from cleanData where MAX <>9999.9)and LEFT(YEARMODA,4)=2015").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where Min in (Select MIN(MIN) from cleanData where Min <>9999.9 and LEFT(YEARMODA,4)='2015') and LEFT(YEARMODA,4) ='2015' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where Min in (Select MIN(MIN) from cleanData where Min <>9999.9 and LEFT(YEARMODA,4)='2016') and LEFT(YEARMODA,4) ='2016' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2016') and LEFT(YEARMODA,4) ='2016' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2017') and LEFT(YEARMODA,4) ='2017' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where Min in (Select MIN(MIN) from cleanData where Min <>9999.9 and LEFT(YEARMODA,4)='2017') and LEFT(YEARMODA,4) ='2017' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2018') and LEFT(YEARMODA,4) ='2018' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where Min in (Select MIN(MIN) from cleanData where Min <>9999.9 and LEFT(YEARMODA,4)='2018') and LEFT(YEARMODA,4) ='2018' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX in (Select MAX(MAX) from cleanData where MAX <>9999.9 and LEFT(YEARMODA,4)='2019') and LEFT(YEARMODA,4) ='2019' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where Min in (Select MIN(MIN) from cleanData where Min <>9999.9 and LEFT(YEARMODA,4)='2019') and LEFT(YEARMODA,4) ='2019' ").show()

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MIN from cleanData where MIN IN(SELECT MIN(MIN) FROM cleanData where MIN <> 9999.9) ").show()

Hive Session ID = 1f115fd9-1106-4ed4-b6c5-cb67c24c51c1

>>> spark.sql("SELECT distinct(STN), LEFT(YEARMODA,4) as YEAR, RIGHT(YEARMODA,2) as DATE, MAX from cleanData where MAX IN(SELECT MAX(MAX) FROM cleanData where MAX <> 9999.9) ").show()

>>> spark.sql("Select STN,RIGHT(YEARMODA,2) as DATE, LEFT(YEARMODA,4) as YEAR, PRCP from cleanData1 where PRCP in (SELECT MAX(PRCP) from cleanData1 where PRCP <>'99.99')")

>>> spark.sql("Select STN,RIGHT(YEARMODA,2) as DATE, LEFT(YEARMODA,4) as YEAR, PRCP from cleanData1 where PRCP in (SELECT MIN(PRCP) from cleanData1 where PRCP <>'99.99')").show()

>>> table\_miss=cleanData1.where("STP='9999.9'")

>>> lost\_value=(float)(table\_miss.count())

>>> total=(float)(cleanData1.count())

>>> count\_percentage=(lost\_value\*100)/(total)

>>> print(count\_percentage)

>>> spark.sql("Select STN,RIGHT(YEARMODA,2)as DATE,LEFT(YEARMODA,4) as YEAR,GUST from newdata where GUST in (Select MAX(GUST) from newdata where GUST <>'999.9')").show()