```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils
import json
from pyspark.sql import SparkSession
from pyspark.sql.types import *
import datetime
import time
from pyspark.sql.functions import split
from pyspark.sql.functions import *
sc = SparkContext(appName='dezyre_test')
sc.setLogLevel('WARN')
spark = SparkSession(sc)

#Read streaming data from Kafka into Pyspark dataframe
dfCSV=spark.readStream.format('kafka').option('kafka.bootstrap.servers','loca
lhost:9092').option('subscribe',
'dezyre_data_csv').option("failOnDataLoss","false").option('startingOffsets',
'earliest').load().selectExpr("CAST(value AS STRING)")
dfCSV.printSchema()

#Define schema for the data
userSchema =StructType([
StructField('Global_new_confirmed',StringType()),
StructField('Global_new_deaths',StringType()),
StructField('Global_new_recovered',StringType()),
StructField('Global_total_confirmed',StringType()),
StructField('Global_total_deaths',StringType()),
StructField('Global_total_recovered',StringType()),
StructField('Country_code',StringType()),
StructField('Country_name',StringType()),
StructField('Country_new_deaths',StringType()),
StructField('Country_new_recovered',StringType()),
StructField('Country_newconfirmed',StringType()),
StructField('Country_slug',StringType()),
StructField('Country_total_confirmed',StringType()),
StructField('Country_total_deaths',StringType()),
StructField('Country_total_recovered',StringType()),
StructField('Extracted_timestamp',TimestampType())
])



#Parse the data
def parse_data_from_kafka_message(sdf, schema):
  from pyspark.sql.functions import split
  assert sdf.isStreaming == True, "DataFrame doesn't receive streaming data"
```

```
  col = split(sdf['value'], ',') #split attributes to nested array in one
Column
  #now expand col to multiple top-level columns
  for idx, field in enumerate(schema):
      sdf  = sdf.withColumn(field.name,
col.getItem(idx).cast(field.dataType))
  return sdf.select([field.name for field in schema])
dfCSV = parse_data_from_kafka_message(dfCSV, userSchema)


#Process the data
q=dfCSV.groupBy("Country_code","Country_name","Country_total_deaths","Extract
ed_timestamp").count()

#Write streaming data to output Kafka topic which can be consumed by
destination services like #HDFS, Nifi, etc.
q2=q.select(to_json(struct(
'Country_code',
'Country_name',
'Country_total_deaths','Extracted_timestamp')).alias('value')).writeStream.fo
rmat("kafka").outputMode("complete").option("failOnDataLoss","false").option(
'checkpointLocation','/home/ubuntu/checkpoint_out').option("kafka.bootstrap.s
ervers","ip-172-31-23-142.us-east-2.compute.internal:9092").option("topic",
"dezyre_out").start().awaitTermination()
```