```
!pip install pyspark==3.0.1 py4j==0.10.9
```

```
Collecting pyspark==3.0.1
  Downloading pyspark-3.0.1.tar.gz (204.2 MB)
     |████████████████████████████████| 204.2 MB 39 kB/s
Collecting py4j==0.10.9
  Downloading py4j-0.10.9-py2.py3-none-any.whl (198 kB)
     |████████████████████████████████| 198 kB 78.2 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.0.1-py2.py3-none-any.whl size=2046122
  Stored in directory: /root/.cache/pip/wheels/5e/34/fa/b37b5cef503fc5148b478b249504
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9 pyspark-3.0.1
```

```
from pyspark.sql import SparkSession

spark = SparkSession\
        .builder\
        .master("local[4]")\
        .appName('lab_4')\
        .getOrCreate()


csv_file = r"/content/IHME_GDP_1960_2050_Y2021M09D22.CSV"
data = spark.read.csv(csv_file, header=True)

data.show(5)
```

```
+-----------+-------------+----+------+----+----------------+----------------+-------
|location_id|location_name|iso3| level|year|    gdp_ppp_mean|   gdp_ppp_lower|   gdp_
+-----------+-------------+----+------+----+----------------+----------------+-------
|          1|       Global|   G|Global|1960|17483449774122.9|16019146112388.8|191158
|          1|       Global|   G|Global|1961|18135370554950.5|16595371585758.2|198249
|          1|       Global|   G|Global|1962|18953278607513.5|17390391432341.6|206147
|          1|       Global|   G|Global|1963|19656620517295.9|18117057797516.5|213499
|          1|       Global|   G|Global|1964|21005747228643.4|19356640986099.7|227679
+-----------+-------------+----+------+----+----------------+----------------+-------
only showing top 5 rows
```

```
data = spark.read.csv('/content/IHME_GDP_1960_2050_Y2021M09D22.CSV'
, sep=',', header=True)
data.printSchema()
```

```
root
 |-- location_id: string (nullable = true)
 |-- location_name: string (nullable = true)
 |-- iso3: string (nullable = true)
 |-- level: string (nullable = true)
 |-- year: string (nullable = true)
 |-- gdp_ppp_mean: string (nullable = true)
```

```
 |-- gdp_ppp_lower: string (nullable = true)
 |-- gdp_ppp_upper: string (nullable = true)
 |-- gdp_usd_mean: string (nullable = true)
 |-- gdp_usd_lower: string (nullable = true)
 |-- gdp_usd_upper: string (nullable = true)
```

```
from pyspark.sql.types import *
data_schema = [
    StructField('location_id', IntegerType(), True),
    StructField('location_name', StringType(), True),
    StructField('iso3', StringType(), True),
    StructField('level', StringType(), True),
    StructField('year', IntegerType(), True),
    StructField('gdp_ppp_mean', FloatType(), True),
    StructField('gdp_ppp_lower', FloatType(), True),
    StructField('gdp_ppp_upper', FloatType(), True),
    StructField('gdp_usd_mean', FloatType(), True),
    StructField('gdp_usd_lower', FloatType(), True),
    StructField('gdp_usd_upper', FloatType(), True),
    ]

final_struc = StructType(fields = data_schema)

data2 = spark.read.csv(csv_file, header=True, schema=final_struc)
data2.printSchema()
```

```
root
 |-- location_id: integer (nullable = true)
 |-- location_name: string (nullable = true)
 |-- iso3: string (nullable = true)
 |-- level: string (nullable = true)
 |-- year: integer (nullable = true)
 |-- gdp_ppp_mean: float (nullable = true)
 |-- gdp_ppp_lower: float (nullable = true)
 |-- gdp_ppp_upper: float (nullable = true)
 |-- gdp_usd_mean: float (nullable = true)
 |-- gdp_usd_lower: float (nullable = true)
 |-- gdp_usd_upper: float (nullable = true)
```

```
data.dtypes
```

```
[('location_id', 'string'),
 ('location_name', 'string'),
 ('iso3', 'string'),
 ('level', 'string'),
 ('year', 'string'),
 ('gdp_ppp_mean', 'string'),
 ('gdp_ppp_lower', 'string'),
 ('gdp_ppp_upper', 'string'),
 ('gdp_usd_mean', 'string'),
 ('gdp_usd_lower', 'string'),
 ('gdp_usd_upper', 'string')]
```

```
data2.head(2)
```

```
[Row(location_id=1, location_name='Global', iso3='G', level='Global', year=1960, gdp_
 Row(location_id=1, location_name='Global', iso3='G', level='Global', year=1961, gdp_
```

```
data2.tail(2)
```

```
[Row(location_id=44578, location_name='Low income', iso3=None, level='World Bank Inco
 Row(location_id=44578, location_name='Low income', iso3=None, level='World Bank Inco
```

```
res = data2.withColumn('Nowa kolumna', data2.year*0 + 1000)
```

```
res = res.withColumnRenamed ('Nowa kolumna', 'col')
```

```
res = data2.drop ('col')
```

```
from pyspark.sql.functions import udf
```

```
i = -1
def incr ():
    global i
    i = i+1
    return i

newCol = udf(incr , IntegerType ())

# dodanie nowej kolumny
data3 = data2.withColumn ('id', newCol ())

data3.show (5)
```

```
+-----------+-------------+----+------+----+-------------+-------------+-------------
|location_id|location_name|iso3| level|year| gdp_ppp_mean|gdp_ppp_lower|gdp_ppp_upper
+-----------+-------------+----+------+----+-------------+-------------+-------------
|          1|       Global|   G|Global|1960|1.74834498E13|1.60191459E13|1.91158634E13
|          1|       Global|   G|Global|1961|1.81353715E13| 1.6595372E13|1.98249273E13
|          1|       Global|   G|Global|1962|1.89532796E13|1.73903918E13|2.06147714E13
|          1|       Global|   G|Global|1963|1.96566204E13|1.81170571E13|2.13499343E13
|          1|       Global|   G|Global|1964|2.10057476E13|1.93566417E13| 2.2767911E13
+-----------+-------------+----+------+----+-------------+-------------+-------------
only showing top 5 rows
```

```
# poczatkowa liczba rekordow
data3.count()
```

```
19838
```

```
# usuniecie wierszy bez danych ( sposob 1.)
data4 = data3.na.drop()


data4.count()
```

```
     18655
```

```
# wstawienie zera w miejsce braku danych ( sposob 2.)
data5 = data3.na.fill(data3.select(0 * data3.year).collect()[0][0])

data5.count()
```

```
     19838
```

```
data5.select(['year', 'location_id', 'location_name']).show(5)
```

```
     +----+-----------+-------------+
     |year|location_id|location_name|
     +----+-----------+-------------+
     |1960|          1|       Global|
     |1961|          1|       Global|
     |1962|          1|       Global|
     |1963|          1|       Global|
     |1964|          1|       Global|
     +----+-----------+-------------+
     only showing top 5 rows
```

```
from pyspark.sql.functions import col

data5 .filter(( col('year') >= 2000) & (col('location_id') > 20)).select (['year',
'location_name', 'location_id']).show (5)
```

```
     +----+-------------+-----------+
     |year|location_name|location_id|
     +----+-------------+-----------+
     |2000|         Fiji|         22|
     |2001|         Fiji|         22|
     |2002|         Fiji|         22|
     |2003|         Fiji|         22|
     |2004|         Fiji|         22|
     +----+-------------+-----------+
     only showing top 5 rows
```

```
# dodanie kolumny zawierajacej wynik sprawdzenia ,
#czy rok jest wiekszy niz 2000
from pyspark . sql import functions as f

data5.select('year', 'location_id', 'location_name'
, f.when(data5.year > 2000, '21st century').otherwise ('20 th century')
.alias ('century')).show (5)
```

```
# dodanie kolumny zawierajacej wynik sprawdzenia ,
#czy nazwa kraju zaczyna sie od litery ’A’
data5.select('year', 'location_id', 'location_name',
             data5.location_name.rlike('^F').alias('Acountry')). show (5)
```

```
+----+-----------+-------------+-------------+
|year|location_id|location_name|      century|
+----+-----------+-------------+-------------+
|1960|          1|       Global|20 th century|
|1961|          1|       Global|20 th century|
|1962|          1|       Global|20 th century|
|1963|          1|       Global|20 th century|
|1964|          1|       Global|20 th century|
+----+-----------+-------------+-------------+
only showing top 5 rows
```

```
+----+-----------+-------------+--------+
|year|location_id|location_name|Acountry|
+----+-----------+-------------+--------+
|1960|          1|       Global|   false|
|1961|          1|       Global|   false|
|1962|          1|       Global|   false|
|1963|          1|       Global|   false|
|1964|          1|       Global|   false|
+----+-----------+-------------+--------+
only showing top 5 rows
```

```
# pogrupowanie danych wg kraju
from pyspark.sql.functions import mean, count, min, max


data5.select(['year', 'location_id', 'location_name']).groupBy('location_name')\
      .agg(
          count('year').alias('number of countries'),
          mean('location_id').alias('number'),
          min('year').alias('min year'),
          max('year').alias('max year')
      ).show(5)
```

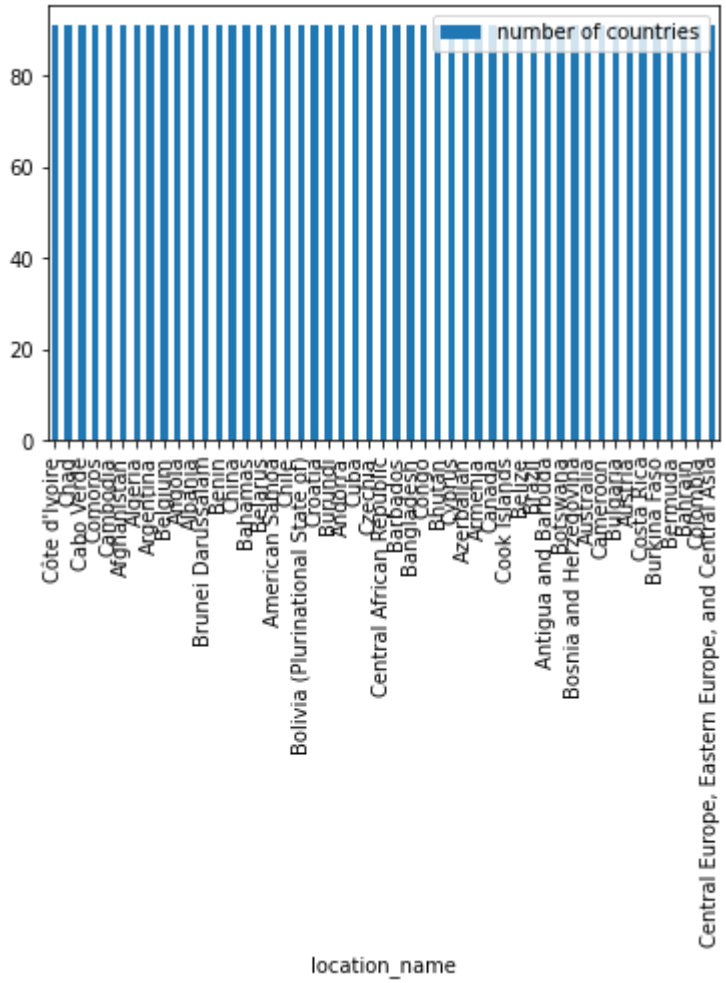
```
+--------------------+-------------------+------+--------+--------+
|       location_name|number of countries|number|min year|max year|
+--------------------+-------------------+------+--------+--------+
|          South Asia|                182| 158.5|    1960|    2050|
|       Côte d'Ivoire|                 91| 205.0|    1960|    2050|
|Micronesia (Feder...|                 91|  25.0|    1960|    2050|
|                Chad|                 91| 204.0|    1960|    2050|
|            Paraguay|                 91| 136.0|    1960|    2050|
+--------------------+-------------------+------+--------+--------+
only showing top 5 rows
```

```
from matplotlib import pyplot as plt

res = data5.filter(data5.location_name.rlike('^[ABC]'))\
.select(['year', 'location_id', 'location_name']).groupBy('location_name')\
```

```
        .agg(
            count('year').alias('number of countries'),
            mean('location_id').alias('number'),
            mean('year').alias('mean year'),
            max('year').alias('max year')
        ).toPandas()

res.plot(kind='bar', x='location_name', y='number of countries')
plt.show()
```



✓ 1 s ukończono o 10:26