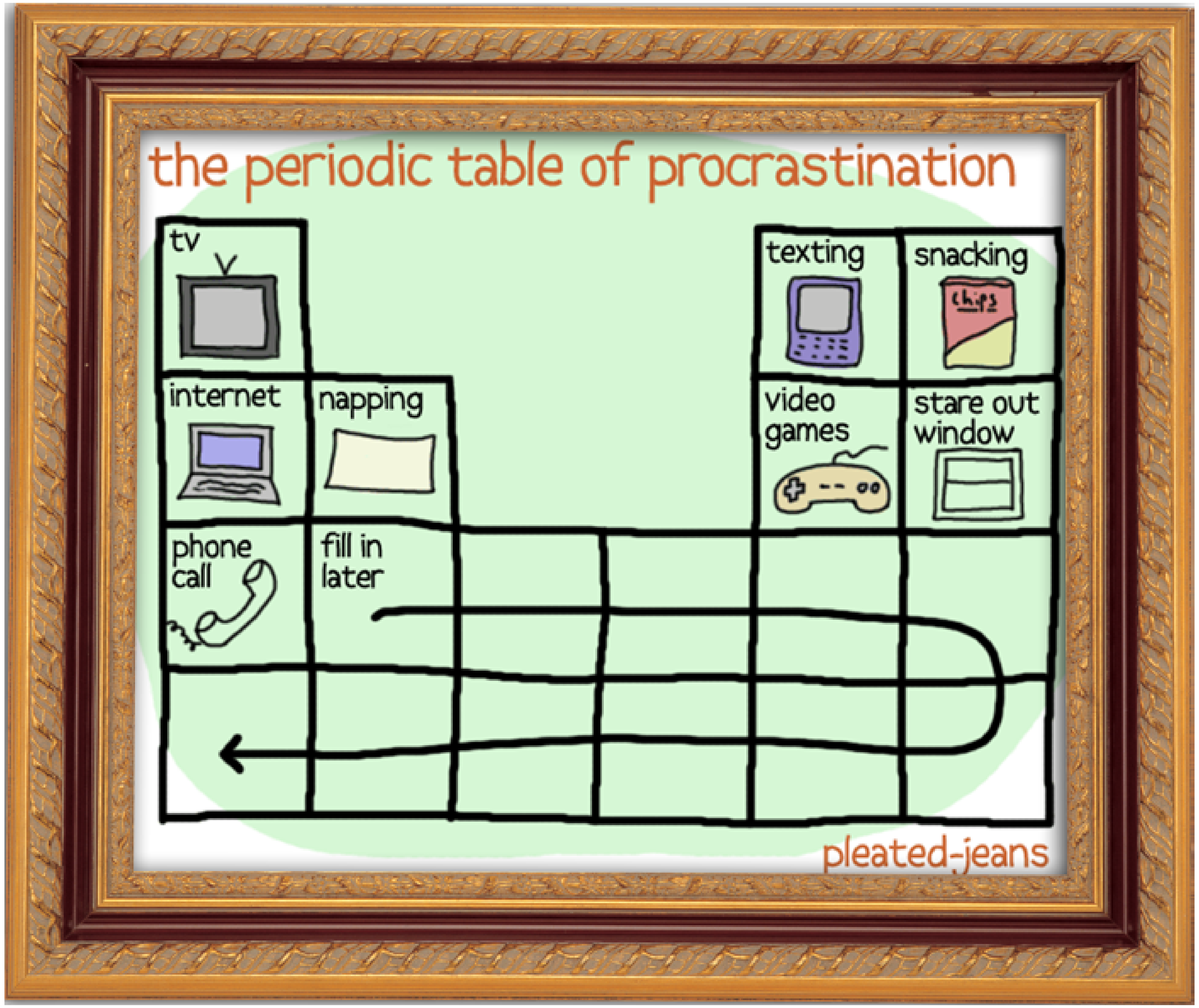
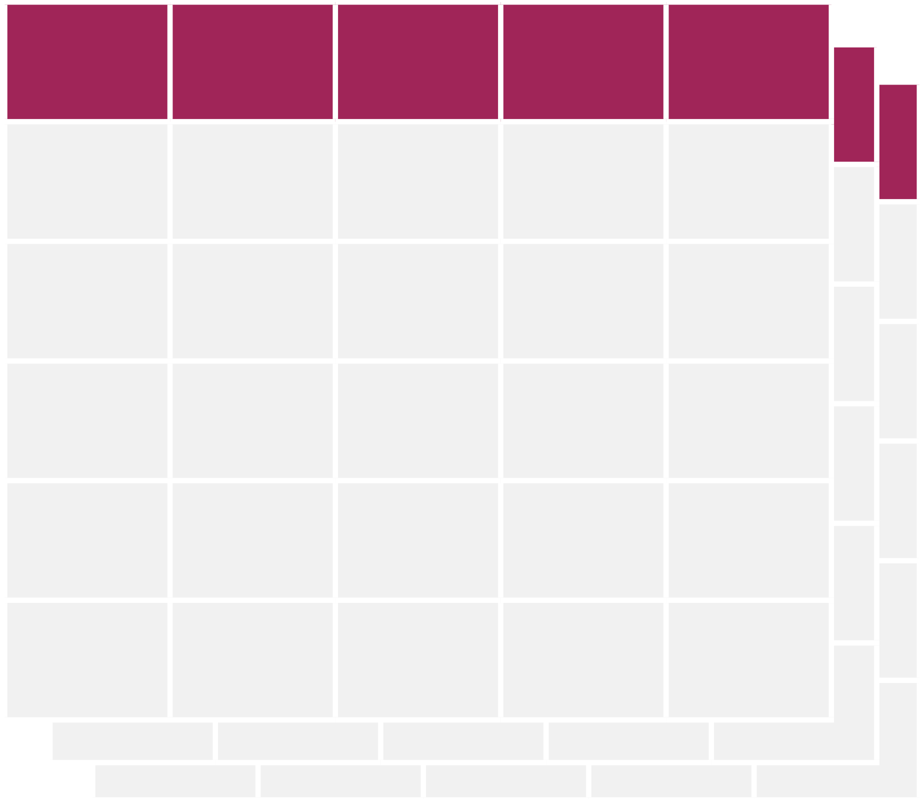
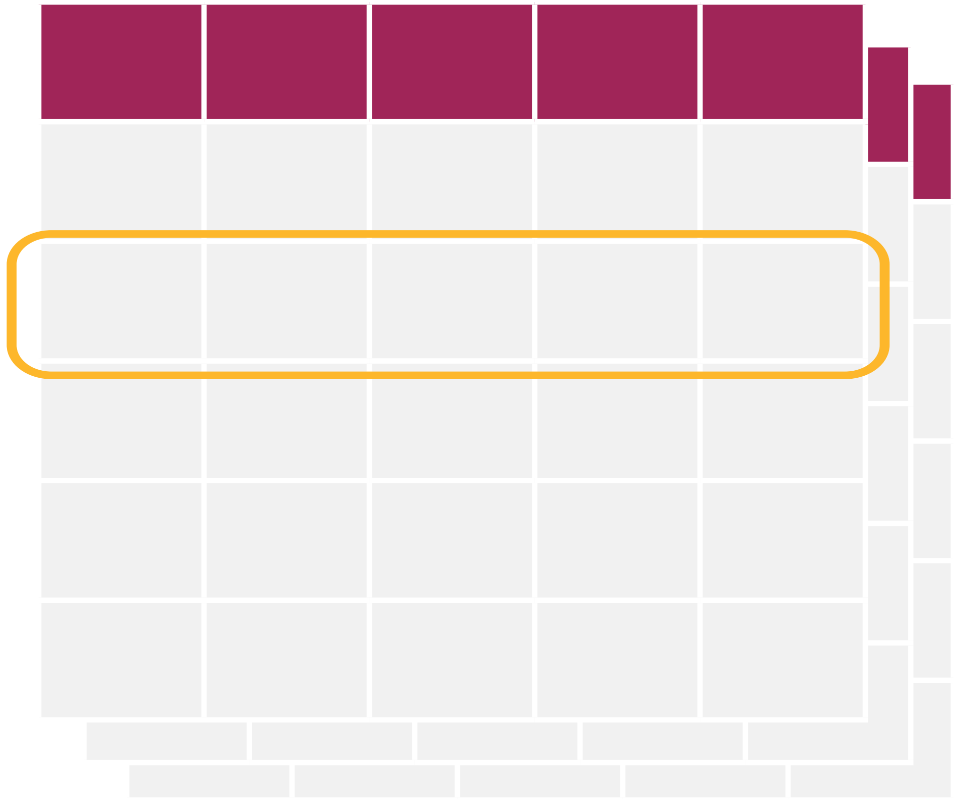
Introducing Spark SQL

What is Pandas DataFrame and how to create it

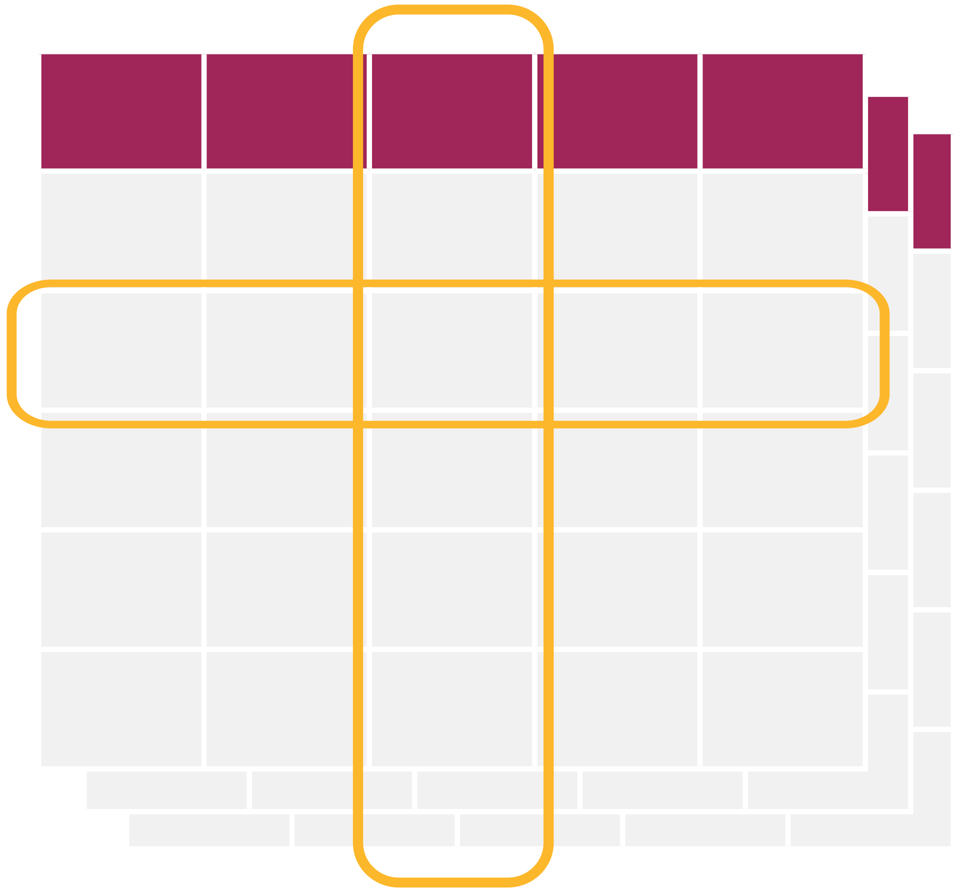

the periodic table of procrastination
tv
texting
snacking
chips
internet
napping
video games
stare out window
phone call
fill in later
pleated-jeans



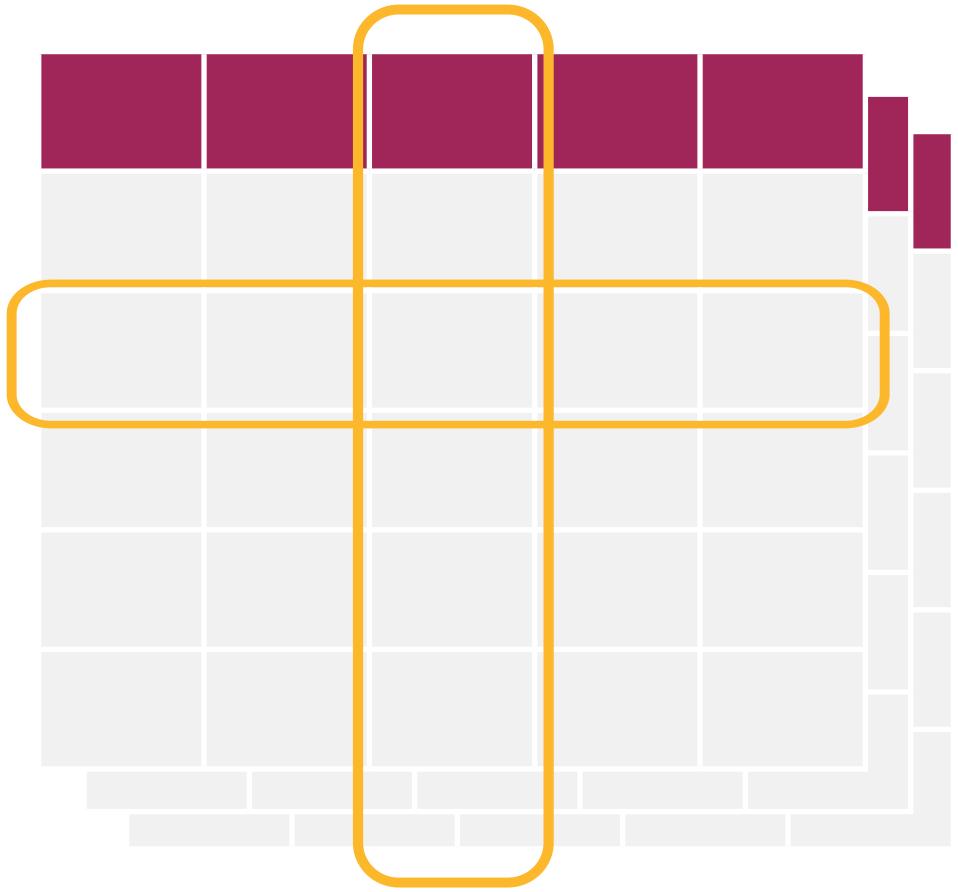
DataFrame



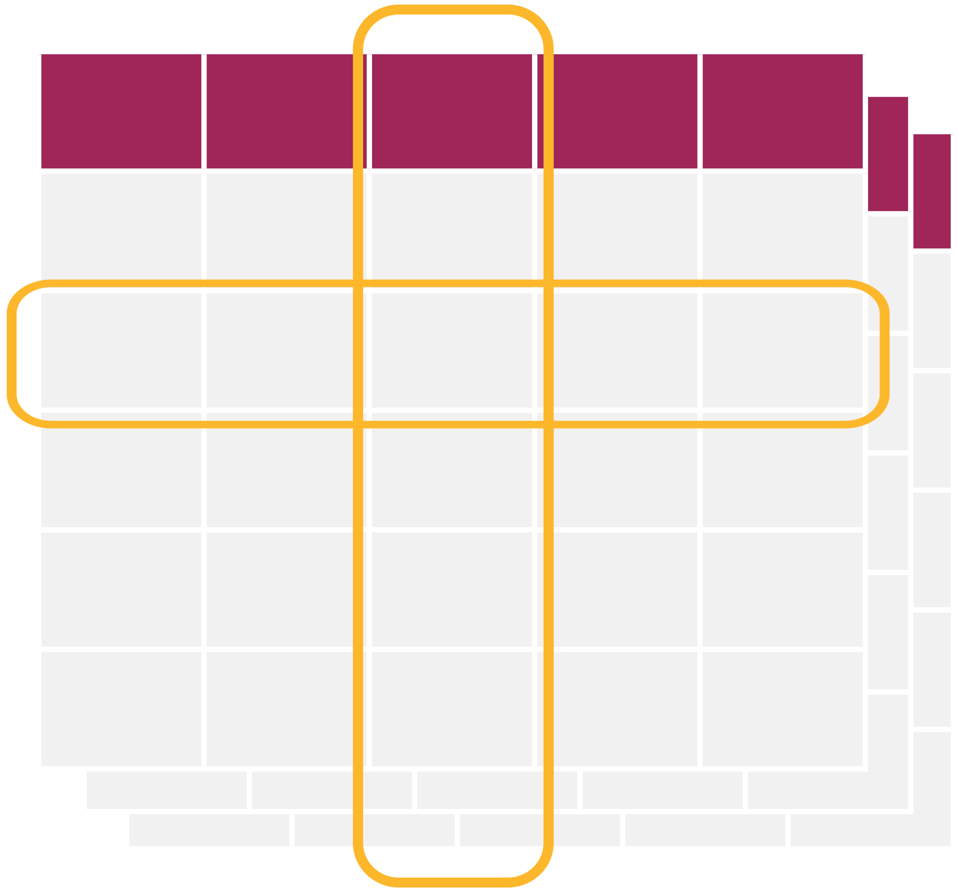
DataFrame



DataFrame



DataFrame



DataFrame = RDD + schema

```
In: from pyspark.sql import SparkSession
    spark_session = SparkSession.builder\
                                .enableHiveSupport()\
                                .appName("spark sql")\
                                .master("local")\
                                .getOrCreate()
```

```
In: from pyspark.sql import SparkSession
    spark_session = SparkSession.builder\
                                .enableHiveSupport()\
                                .appName("spark sql")\
                                .master("local")\
                                .getOrCreate()
```

```
In: geoip_rdd = spark_session\
                   .sparkContext\
                   .textFile("/user/pmezentsev/geoip")
```

```
In: from pyspark.sql import SparkSession
    spark_session = SparkSession.builder\
                                .enableHiveSupport()\
                                .appName("spark sql")\
                                .master("local")\
                                .getOrCreate()
```

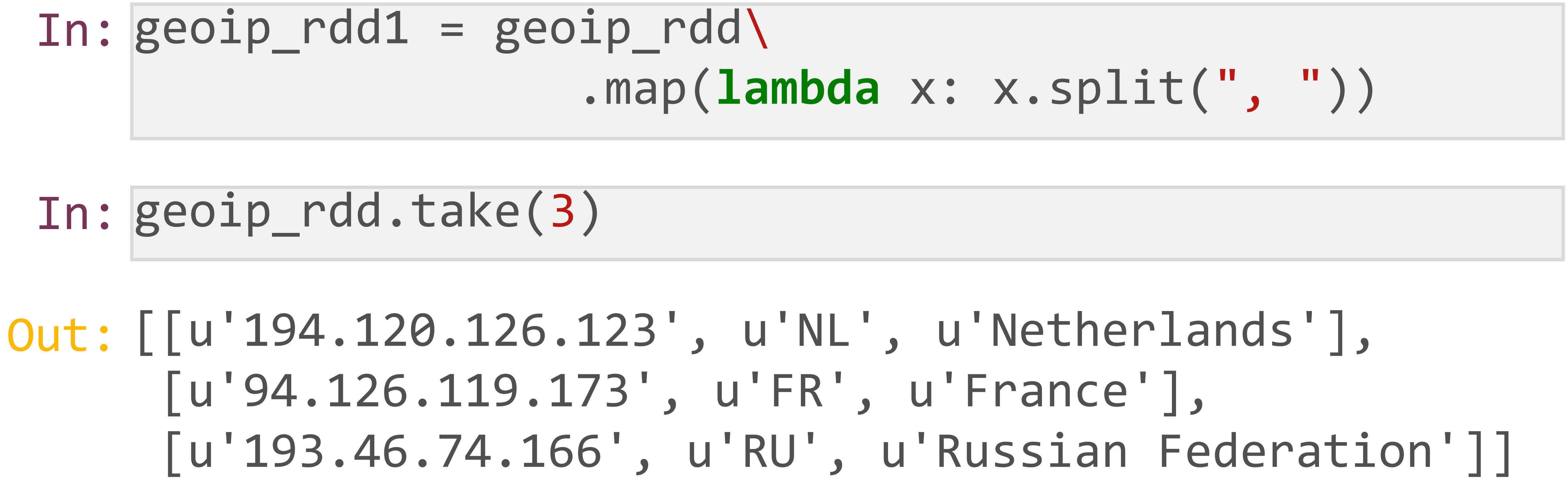
```
In: geoip_rdd = spark_session\
                   .sparkContext\
                   .textFile("/user/pmezentsev/geoip")
```

```
In: geoip_rdd.take(3)
```

```
Out: [u'194.120.126.123, NL, Netherlands',
      u'94.126.119.173, FR, France',
      u'193.46.74.166, RU, Russian Federation']
```

In:

```
geoip_rdd1 = geoip_rdd\
                .map(lambda x: x.split(", "))
```



```
In: geoip_rdd1 = geoip_rdd\
                    .map(lambda x: x.split(", "))
```

```
In: geoip_rdd.take(3)
```

```
Out: [[u'194.120.126.123', u'NL', u'Netherlands'],
      [u'94.126.119.173', u'FR', u'France'],
      [u'193.46.74.166', u'RU', u'Russian Federation']]
```

```
ip STRING,
code STRING,
country STRING
```

ip **STRING**,
code **STRING**,
country **STRING**

```
In: from pyspark.sql.types import *
    schema = StructType().add("ip",      StringType())\
                         .add("code",    StringType())\
                         .add("country", StringType())
```

ip **STRING**,
code **STRING**,
country **STRING**

```
In: from pyspark.sql.types import *
    schema = StructType().add("ip",      StringType())\
                         .add("code",    StringType())\
                         .add("country", StringType())
```

```
In: geoip_df = spark_session\
                   .createDataFrame(geoip_rdd1, schema)
```

ip **STRING**,
code **STRING**,
country **STRING**

```
In: from pyspark.sql.types import *
    schema = StructType().add("ip",      StringType())\
                         .add("code",    StringType())\
                         .add("country", StringType())
```

```
In: geoip_df = spark_session\
                 .createDataFrame(geoip_rdd1, schema)
```

```
In: geoip_df
```

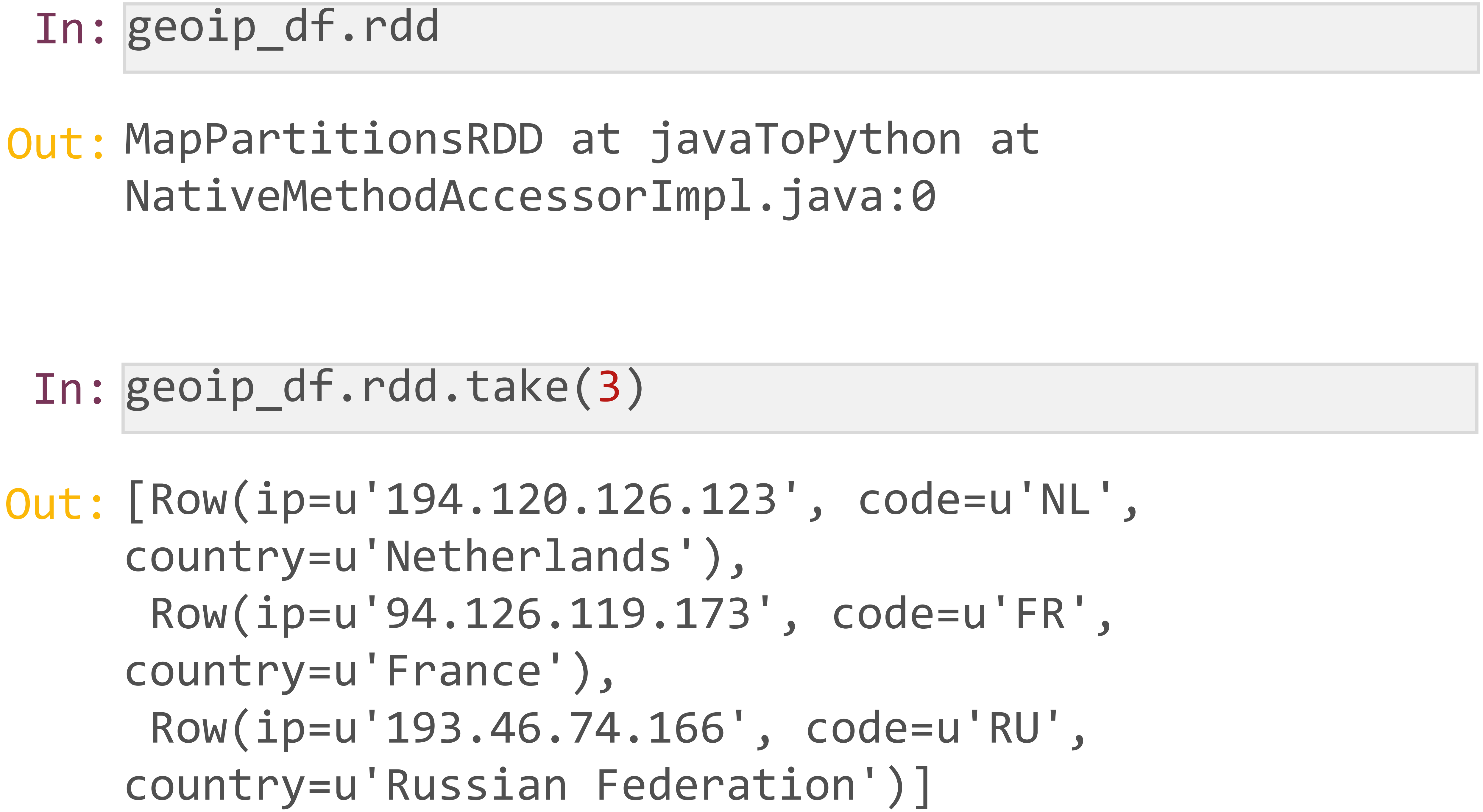
```
Out: DataFrame[ip: string, code: string, country: string]
```

```
In: geoip_df.show(3)
```

```
+---------------+----+------------------+
|             ip|code|           country|
+---------------+----+------------------+
|194.120.126.123|  NL|       Netherlands|
| 94.126.119.173|  FR|            France|
|  193.46.74.166|  RU|Russian Federation|
+---------------+----+------------------+
only showing top 3 rows
```

```
In:  geoip_df.rdd
```

```
Out: MapPartitionsRDD at javaToPython at
     NativeMethodAccessorImpl.java:0
```



```
In:  geoip_df.rdd

Out: MapPartitionsRDD at javaToPython at
     NativeMethodAccessorImpl.java:0
```

```
In:  geoip_df.rdd.take(3)

Out: [Row(ip=u'194.120.126.123', code=u'NL',
     country=u'Netherlands'),
      Row(ip=u'94.126.119.173', code=u'FR',
     country=u'France'),
      Row(ip=u'193.46.74.166', code=u'RU',
     country=u'Russian Federation')]
```

```
In: geoip_df.printSchema()
```

```
root
 |-- ip: string (nullable = true)
 |-- code: string (nullable = true)
 |-- country: string (nullable = true)
```

APACHE
Spark
TM

pandas

$y_it = \beta' x_{it} + \mu_i + \epsilon_{it}$

```
In: geoip_pd = geoip_df.toPandas()
```

pandas

$y_it = \beta' x_{it} + \mu_i + \epsilon_{it}$

```
In: geoip_pd = geoip_df.toPandas()
```

```
In: geoip_pd.head(3)
```

Out:

	ip	code	country
0	194.120.126.123	NL	Netherlands
1	94.126.119.173	FR	France
2	193.46.74.166	RU	Russian Federation

APACHE
Spark
TM

In:

```
geoip_01_df = spark_session.createDataFrame(geoip_pd)
```

```
In: geoip_01_df = spark_session.createDataFrame(geoip_pd)
```

```
In: geoip_01_df
```

```
Out: DataFrame[ip: string, code: string, country: string]
```

What have we learned:

- What is spark dataframe
- How to create it from RDD
- What is dataframe's schema
- How to convert pandas DataFrame to spark DataFrame and vice versa