In [1]:
```
a = 7
```

In [2]:
```
b = 8
```

In [3]:
```
a + b
```

Out[3]:
```
15
```

In [4]:
```
import numpy as numpy
import matplotlib.pyplot as plot
x = numpy.arange(-10, 10, 0.01)
def sigmoid(x):
 return 1/(1+numpy.exp(-x))
plot.plot(x, sigmoid(x))
plot.title('Sigmoid')
plot.grid(True)
plot.show()
```

```
<Figure size 640x480 with 1 Axes>
```

In [5]:
```
import pyspark
context = pyspark.SparkContext('local[*]')
print(context)
print(context.version)
```

```
<SparkContext master=local[*] appName=pyspark-shell>
2.4.0
```

In [6]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
print(spark)
print(spark.catalog.listTables())
```

```
<pyspark.sql.session.SparkSession object at 0x7febd9b266d8>
[]
```

In [7]:
```
import pandas as pd
import numpy as np
pandasdf = pd.DataFrame(np.random.random(10))
sparkdf = spark.createDataFrame(pandasdf)
sparkdf.createOrReplaceTempView("random")
print(spark.catalog.listTables())
```

```
[Table(name='random', database=None, description=None, tableType='TEMPORARY',
isTemporary=True)]
```

In [8]:
```
sparkdf.printSchema()
print(sparkdf.head(2))
print()
sparkdf.show(2,truncate=True)
print()
print(sparkdf.count())
```

```
root
 |-- 0: double (nullable = true)

[Row(0=0.5457280211743398), Row(0=0.770876244301722)]

+------------------+
|                 0|
+------------------+
|0.5457280211743398|
| 0.770876244301722|
+------------------+
only showing top 2 rows


10
```

In [9]:
```
filepath = '/opt/conda/lib/python3.7/site-packages/'
filepath += 'bokeh/sampledata/_data/us_marriages_divorces.csv'
divorces = spark.read.csv(filepath, header=True)
divorces.show(n=5)
divorces.filter('Year >= 1960').show(n=5)
divorces.filter('Year >= 2000').show(n=5)

filepath = '/opt/conda/lib/python3.7/site-packages/'
filepath += 'bokeh/sampledata/_data/us_marriages_divorces.csv'
filtered = spark.read.csv(filepath, header=True)
filtered.show(n=5)
filtered.filter('Year >= 1960').show(n=5)
filtered.filter('Year >= 2000').show(n=5)
```

```
+----+---------+--------+----------+------------------+-----------------+
|Year|Marriages|Divorces|Population|Marriages_per_1000|Divorces_per_1000|
+----+---------+--------+----------+------------------+-----------------+
|1867|   357000|   10000|  36970000|               9.7|              0.3|
|1868|   345000|   10000|  37885000|               9.1|              0.3|
|1869|   348000|   11000|  38870000|                 9|              0.3|
|1870|   352000|   11000|  39905000|               8.8|              0.3|
|1871|   359000|   12000|  41010000|               8.8|              0.3|
+----+---------+--------+----------+------------------+-----------------+
only showing top 5 rows

+----+---------+--------+----------+------------------+-----------------+
|Year|Marriages|Divorces|Population|Marriages_per_1000|Divorces_per_1000|
+----+---------+--------+----------+------------------+-----------------+
|1960|  1523000|  393000| 180760000|               8.4|              2.2|
|1961|  1548000|  414000| 183742000|               8.4|              2.3|
|1962|  1577000|  413000| 186590000|               8.5|              2.2|
|1963|  1654000|  428000| 189300000|               8.7|              2.3|
|1964|  1725000|  450000| 191927000|                 9|              2.3|
+----+---------+--------+----------+------------------+-----------------+
only showing top 5 rows

+----+---------+--------+----------+------------------+-----------------+
|Year|Marriages|Divorces|Population|Marriages_per_1000|Divorces_per_1000|
+----+---------+--------+----------+------------------+-----------------+
|2000|  2315000|  944000| 282398000|               8.2|              3.3|
|2001|  2326000|  940000| 285225000|               8.2|              3.3|
|2002|  2290000|  955000| 287955000|                 8|              3.3|
|2003|  2245000|  927000| 290626000|               7.7|              3.2|
|2004|  2279000|  879000| 293262000|               7.8|                3|
+----+---------+--------+----------+------------------+-----------------+
only showing top 5 rows

+----+---------+--------+----------+------------------+-----------------+
|Year|Marriages|Divorces|Population|Marriages_per_1000|Divorces_per_1000|
+----+---------+--------+----------+------------------+-----------------+
|1867|   357000|   10000|  36970000|               9.7|              0.3|
|1868|   345000|   10000|  37885000|               9.1|              0.3|
|1869|   348000|   11000|  38870000|                 9|              0.3|
|1870|   352000|   11000|  39905000|               8.8|              0.3|
|1871|   359000|   12000|  41010000|               8.8|              0.3|
+----+---------+--------+----------+------------------+-----------------+
only showing top 5 rows

+----+---------+--------+----------+------------------+-----------------+
|Year|Marriages|Divorces|Population|Marriages_per_1000|Divorces_per_1000|
+----+---------+--------+----------+------------------+-----------------+
|1960|  1523000|  393000| 180760000|               8.4|              2.2|
|1961|  1548000|  414000| 183742000|               8.4|              2.3|
|1962|  1577000|  413000| 186590000|               8.5|              2.2|
|1963|  1654000|  428000| 189300000|               8.7|              2.3|
|1964|  1725000|  450000| 191927000|                 9|              2.3|
+----+---------+--------+----------+------------------+-----------------+
only showing top 5 rows

+----+---------+--------+----------+------------------+-----------------+
|Year|Marriages|Divorces|Population|Marriages_per_1000|Divorces_per_1000|
```

```
+----+---------+--------+----------+------------------+-----------------+
|2000|  2315000|  944000| 282398000|               8.2|              3.3|
|2001|  2326000|  940000| 285225000|               8.2|              3.3|
|2002|  2290000|  955000| 287955000|                 8|              3.3|
|2003|  2245000|  927000| 290626000|               7.7|              3.2|
|2004|  2279000|  879000| 293262000|               7.8|                3|
+----+---------+--------+----------+------------------+-----------------+
only showing top 5 rows
```

In [10]:
```
print(type(divorces))
print(type(divorces))
```

```
<class 'pyspark.sql.dataframe.DataFrame'>
<class 'pyspark.sql.dataframe.DataFrame'>
```

In [11]:
```
words = context.parallelize('one fish two fish red fish blue fish'.split())
print(type(words))
print(words.count())
print(words.collect())
```

```
<class 'pyspark.rdd.RDD'>
8
['one', 'fish', 'two', 'fish', 'red', 'fish', 'blue', 'fish']
```

In [12]:
```
notFishWords = words.filter(lambda s: s != 'fish')
print(notFishWords.collect())
```

```
['one', 'two', 'red', 'blue']
```

In [13]:
```
capWords = words.map(lambda s: s.capitalize())
print(capWords.collect())
```

```
['One', 'Fish', 'Two', 'Fish', 'Red', 'Fish', 'Blue', 'Fish']
```

In [14]:
```
title = words.reduce(lambda s,t: s + ' ' + t)
print(title)
```

```
one fish two fish red fish blue fish
```

In [15]:
```
shortWords = words.filter(lambda s: len(s) == 3)
smallWords = shortWords.map(lambda s: s.upper())
print(smallWords.collect())
```

```
['ONE', 'TWO', 'RED']
```

In [16]:
```
from pyspark.sql import Row
customerList = [
(10010,'Ramas' ,'Alfred'),
(10011,'Dunne' ,'Leona' ),
(10012,'Smith' ,'Kathy' ),
(10013,'Olowski' ,'Paul' ),
(10014,'Orlando' ,'Myron' ),
(10015,'O''Brian','Amy' ),
(10016,'Brown' ,'James' ),
(10017,'Williams','George'),
(10018,'Farriss' ,'Anne' ),
(10019,'Smith' ,'Olette')
]
customerRdd = context.parallelize(customerList)
print(type(customerRdd))
customersRowsRdd = customerRdd.map(
 lambda x: Row(cust_id=x[0], cust_lastname=x[1], cust_firstname=x[2])
)
print(type(customersRowsRdd))
customersdf = spark.createDataFrame(customersRowsRdd)
print(type(customersdf))
```

```
<class 'pyspark.rdd.RDD'>
<class 'pyspark.rdd.PipelinedRDD'>
<class 'pyspark.sql.dataframe.DataFrame'>
```

In [17]:
```
customersdf.printSchema()
customersdf.show()
```

```
root
 |-- cust_firstname: string (nullable = true)
 |-- cust_id: long (nullable = true)
 |-- cust_lastname: string (nullable = true)

+--------------+-------+-------------+
|cust_firstname|cust_id|cust_lastname|
+--------------+-------+-------------+
|        Alfred|  10010|        Ramas|
|         Leona|  10011|        Dunne|
|         Kathy|  10012|        Smith|
|          Paul|  10013|      Olowski|
|         Myron|  10014|      Orlando|
|           Amy|  10015|       OBrian|
|         James|  10016|        Brown|
|        George|  10017|     Williams|
|          Anne|  10018|      Farriss|
|        Olette|  10019|        Smith|
+--------------+-------+-------------+
```

In [18]:
```
customersRowsRdd = customerRdd.map(lambda x: Row(cust_name=x[1]+ " " +x[2], cu
st_id=x[0]))
customersdf = spark.createDataFrame(customersRowsRdd)
customersdf.printSchema()
customersdf.show()
```

```
root
 |-- cust_id: long (nullable = true)
 |-- cust_name: string (nullable = true)

+-------+---------------+
|cust_id|      cust_name|
+-------+---------------+
|  10010|   Ramas Alfred|
|  10011|    Dunne Leona|
|  10012|    Smith Kathy|
|  10013|   Olowski Paul|
|  10014|  Orlando Myron|
|  10015|     OBrian Amy|
|  10016|    Brown James|
|  10017|Williams George|
|  10018|   Farriss Anne|
|  10019|   Smith Olette|
+-------+---------------+
```

In [19]:
```
bachelorsWomen = spark.read.csv(
 "/opt/conda/lib/python3.7/site-packages/bokeh/sampledata/_data/percent-bachelors-degrees-women-usa.csv",
 header = True, inferSchema = True)
bachelorsWomen.printSchema()
print(bachelorsWomen.columns)
```

```
root
 |-- Year: integer (nullable = true)
 |-- Agriculture: double (nullable = true)
 |-- Architecture: double (nullable = true)
 |-- Art and Performance: double (nullable = true)
 |-- Biology: double (nullable = true)
 |-- Business: double (nullable = true)
 |-- Communications and Journalism: double (nullable = true)
 |-- Computer Science: double (nullable = true)
 |-- Education: double (nullable = true)
 |-- Engineering: double (nullable = true)
 |-- English: double (nullable = true)
 |-- Foreign Languages: double (nullable = true)
 |-- Health Professions: double (nullable = true)
 |-- Math and Statistics: double (nullable = true)
 |-- Physical Sciences: double (nullable = true)
 |-- Psychology: double (nullable = true)
 |-- Public Administration: double (nullable = true)
 |-- Social Sciences and History: double (nullable = true)

['Year', 'Agriculture', 'Architecture', 'Art and Performance', 'Biology', 'Business', 'Communications and Journalism', 'Computer Science', 'Education', 'Engineering', 'English', 'Foreign Languages', 'Health Professions', 'Math and Statistics', 'Physical Sciences', 'Psychology', 'Public Administration', 'Social Sciences and History']
```

In [20]:
```
print('Rows: %d' % bachelorsWomen.count())
print('Columns: %d' % len(bachelorsWomen.columns))
```

```
Rows: 42
Columns: 18
```

In [21]:
```
bachelorsWomen.describe('Computer Science').show()
```

```
+-------+------------------+
|summary|  Computer Science|
+-------+------------------+
|  count|                42|
|   mean| 25.80952380952381|
| stddev|6.6887531932272015|
|    min|              13.6|
|    max|              37.1|
+-------+------------------+
```

In [22]:
```
bachelorsWomen.registerTempTable('bw')
bachelorsWomenTech = spark.sql(
 'select AVG(Year), AVG(`Computer Science`), AVG(`Engineering`), AVG(`Physical Sciences`)'
 ' from bw'
)
bachelorsWomenTech.show()
```

```
+---------+--------------------+------------------+----------------------+
|avg(Year)|avg(Computer Science)|  avg(Engineering)|avg(Physical Sciences)|
+---------+--------------------+------------------+----------------------+
|   1990.5|    25.80952380952381|12.892857142857142|    31.304761904761904|
+---------+--------------------+------------------+----------------------+
```

In [ ]: