TitanicV2

Titanic visualization assignment.

FINISHED

This assignment shows in a nice way the data set from Titanic passenger list

Took 0 sec. Last updated by anonymous at February 21 2020, 4:26:59 PM.

FINISHED

```
%spark

import org.apache.spark.sql.Dataset;
import org.apache.spark.sql.functions.expr

//import org.apache.spark.sql.types.{StructType, StructField, StringType, IntegerType};

// import org.apache.commons.io.IOUtils
// import java.net.URL
// import java.nio.charset.Charset
// import org.apache.spark.ml.classification.DecisionTreeClassificationModel;
// import org.apache.spark.ml.classification.DecisionTreeClassifier;
// import org.apache.spark.ml.feature.StringIndexer;
// import org.apache.spark.ml.feature.StringIndexerModel;
// import org.apache.spark.ml.feature.VectorAssembler;
// import org.apache.spark.mllib.evaluation.MulticlassMetrics;
// import org.apache.spark.sql.SparkSession;
// import org.apache.spark.sql.Row;
// import org.apache.spark.sql.functions;
// import org.apache.spark.sql.types.DataTypes;
// import org.apache.spark.sql.types.StructType;

// création du dataframe Spark
val df = spark.read.options(Map("inferSchema"->"true","delimiter"->",","header"->"true"))
  .csv("/Users/FXSUIRE/Google Drive/_00_Dev/Titanic/input.csv")

// application du schéma de données
// case class passenger_cls(Survived:Int,Pclass:Int,Name:String,Sex:String,Age:Float,Siblings:Int,ParentAboard:Int,Fare:Float,Farel
case class passenger_cls(Survived:Int,Pclass:Int,Name:String,Sex:String,Age:Float,Siblings:Int,ParentAboard:Int,Fare:Float)

//Ratio d'actualisation du prix du ticket en liver sterling 2020
// source: https://www.thisismoney.co.uk/money/bills/article-1633409/Historic-inflation-calculator-value-money-changed-1900.html
```

TitanicV2

```
33 val UpdatingRatio = 115.8
34
35 // val FareActualised = "Fare*UpdatingRatio"
36 // df.withColumn("Fare_2020",expr(FareActualised))
37 // val df2 = df.withColumn("FareUpdated", df("Fare") * UpdatingRatio)
38
39 // x0: Surived:Int
40 // x1: Pclass:Int
41 // x2: Name:String
42 // x3: Sex:String
43 // x4: Age:Float
44 // x5: Siblings:Int
45 // x6: ParentAboard:Int
46 // x7: Fare:Float
47 // x8: Actualised (2020) Fare: float
48
49 // mapping des colonnes
50 //val passenger = data1.map(x=>x.split(",")).map(x => passenger_cls(x(0).toInt,x(1).toInt,x(2),x(3),x(4).toFloat,x(5).toInt,x(6).t
51 val passenger = data1.map(x=>x.split(",")).map(x => passenger_cls(x(0).toInt,x(1).toInt,x(2),x(3),x(4).toFloat,x(5).toInt,x(6).toI
52
53 //enregistrement dans une table temporaire
54 passenger.registerTempTable("PASSENGER")
55
56 //sélection des données que l'on souhaite utiliser dans nos requêtes de visualisations
57
58 // Décès-survie du passager, classe, sex, prix du ticket
59 val SurvivalAgeClassSexFare = sqlContext.sql("select Survived, Pclass, Sex, Fare from PASSENGER")
60 SurvivalAgeClassSexFare.registerTempTable("SMALLPASSTABLE")
61
62 val BasicStats = sqlContext.sql("select min(fare), max(fare), avg(fare), stddev(fare) from PASSENGER where fare>0").toDF
63
64 val BasicStatsRenamed = BasicStats.withColumnRenamed("min(fare)","Minimum fare price")
65             .withColumnRenamed("max(fare)","Maximum fare price")
66             .withColumnRenamed("avg(fare)","Average fare price")
67             .withColumnRenamed("stddev(fare)","Standard deviation")
68
69 BasicStatsRenamed.registerTempTable("BASICFARETABLE")
70
71 // statistiques sur le prix du ticket actualisé, c'est à dire en GBP de 2020
72 // val BasicStats = sqlContext.sql("select min(FareUpdated), max(FareUpdated), avg(FareUpdated), stddev(FareUpdated) from PASSENGE
73
74 //val BasicStatsRenamed = BasicStats.withColumnRenamed("min(FareUpdated)","Minimum fare price")
75             //.withColumnRenamed("max(FareUpdated)","Maximum fare price")
76             //.withColumnRenamed("avg(FareUpdated)","Average fare price")
77             //.withColumnRenamed("stddev(FareUpdated)","Standard deviation")
```

```
78
79 // BasicStatsRenamed.registerTempTable("BASICFARETABLE")
80
81 //BasicStatsRenamed.printSchema()
```

TitanicV2

```
warning: there were three deprecation warnings; re-run with -deprecation for details
import sqlContext.implicits._
import org.apache.spark.sql.Dataset
import org.apache.spark.sql.functions.expr
df: org.apache.spark.sql.DataFrame = [Survived: int, Pclass: int ... 6 more fields]
defined class passenger_cls
UpdatingRatio: Double = 115.8
passenger: org.apache.spark.sql.DataFrame = [Survived: int, Pclass: int ... 6 more fields]
SurvivalAgeClassSexFare: org.apache.spark.sql.DataFrame = [Survived: int, Pclass: int ... 2 more fields]
```

Took 15 sec. Last updated by anonymous at February 21 2020, 4:51:13 PM.

Extract of Titanic passenger list

FINISHED

```
1 %spark
2
3 // we display the raw data
4     df.show(false)
5     df.printSchema()
```

```
+--------+------+------------------------------------------------------+------+----+----------------------+----------------------+----
---+
|Survived|Pclass|Name                                                  |Sex   |Age |Siblings/Spouses Aboard|Parents/Children Aboard|Fare
|
+--------+------+------------------------------------------------------+------+----+----------------------+----------------------+----
---+
|0       |3     |Mr. Owen Harris Braund                                |male  |22.0|1                      |0                      |7.25
|
|1       |1     |Mrs. John Bradley (Florence Briggs Thayer) Cumings    |female|38.0|1                      |0                      |71.2
833|
|1       |3     |Miss. Laina Heikkinen                                 |female|26.0|0                      |0                      |7.92
5  |
|1       |1     |Mrs. Jacques Heath (Lily May Peel) Futrelle           |female|35.0|1                      |0                      |53.1
|
|0       |3     |Mr. William Henry Allen                               |male  |35.0|0                      |0                      |8.05
|
|0       |3     |Mr. James Moran                                       |male  |27.0|0                      |0                      |8.45
```

Took 2 sec. Last updated by anonymous at February 21 2020, 4:27:08 PM.

TitanicV2

Number of deceased or living passenger by sex after the Shipwreck

FINISHED

```
%spark.sql
select Survived,Sex, count(Survived) from SMALLPASSTABLE group by Survived, Sex
```

settings

Survived	Sex
0	female
1	male
1	female
0	male

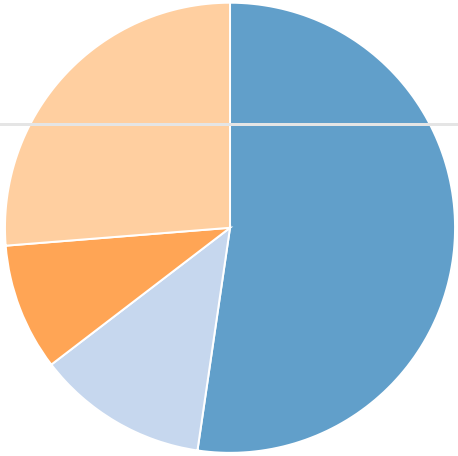
Took 1 sec. Last updated by anonymous at February 21 2020, 4:41:20 PM. (outdated)

Share of Passenger who survived or not by sex

FINISHED

settings

TitanicV2

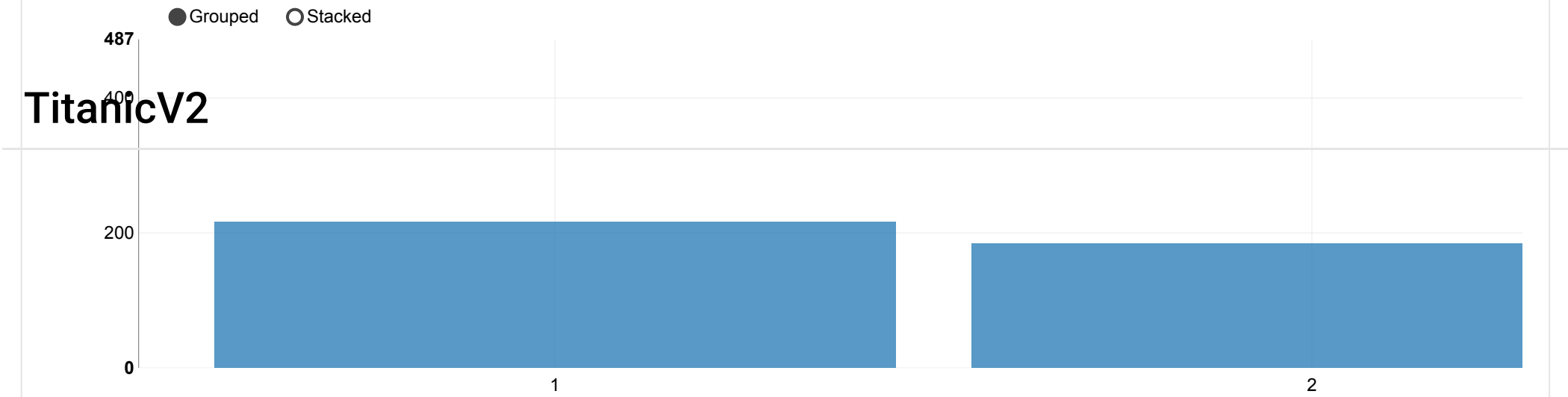


Took 0 sec. Last updated by anonymous at February 21 2020, 4:27:09 PM.

Titanic passenger class distribution

FINISHED

settings

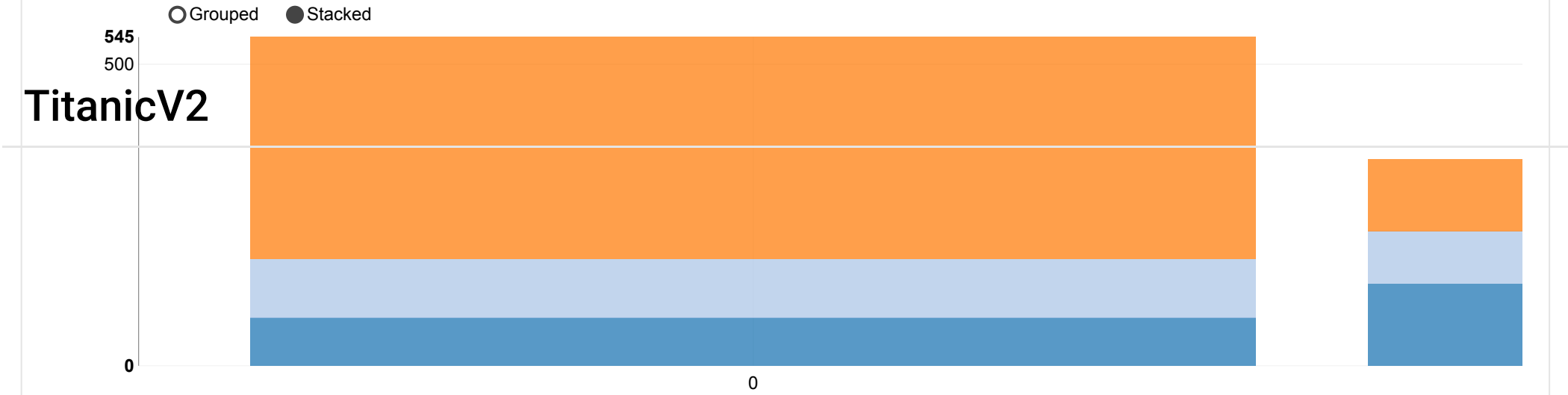


Took 1 sec. Last updated by anonymous at February 21 2020, 4:27:10 PM.

```
%spark.sql
select Survived,Pclass from SMALLPASSTABLE order by Pclass asc
```

FINISHED

settings



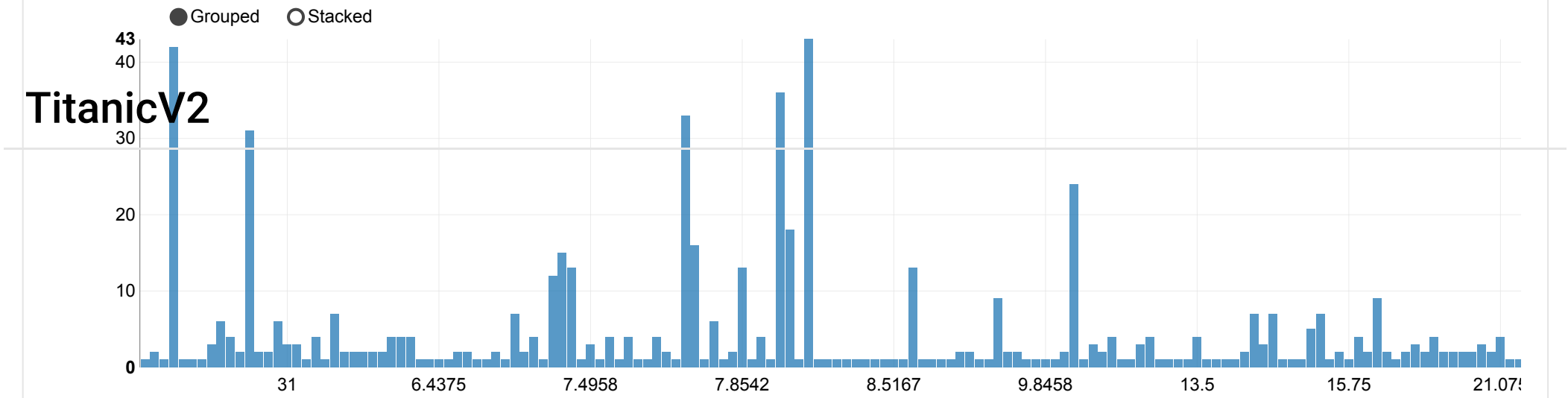
Took 0 sec. Last updated by anonymous at February 21 2020, 4:33:10 PM. (outdated)

Titanic fare price frequency distribution

FINISHED

```
%spark.sql
select fare, count(1) nInBin
from SMALLPASSTABLE
where fare>0
group by fare
order by fare
```

settings



Took 3 sec. Last updated by anonymous at February 21 2020, 4:27:13 PM. (outdated)

Basic statistics about Titanic fare price (in year 1912 GBP)

FINISHED

```
%spark.sql
select * from BASICFARETABLE
```

settings

Minimum fare price	Maximum fare price	Average fare price
4.0125	512.3292	32.86113275737937

TitanicV2

Took 0 sec. Last updated by anonymous at February 21 2020, 4:51:45 PM. (outdated)

```
%spark.sql
```

READY