```
>>> rdd1 = sc.textFile("/home/uttam/futurense-datengg-bootcamp/dataset/bankmarketing.txt")
eader = rdd1.first()
rdd2 = rdd1.filter(lambda x : x != header).filter(lambda x : len(x) > 1)
rdd3 = rdd2.map(lambda x : x.split(';'))
>>> header = rdd1.first()
>>> rdd2 = rdd1.filter(lambda x : x != header).filter(lambda x : len(x) > 1)
>>> rdd3 = rdd2.map(lambda x : x.split(';'))
>>> rdd3.take(5)
[['58', 'management', 'married', 'tertiary', 'no', '2143', 'yes', 'no', 'unknown', '5', 'may', '261', '1', '-1', '0', 'unknown', 'no'], ['44', 'technician', 'single', 'secondary', 'no', '29', 'yes', 'no
', 'unknown', '5', 'may', '151', '1', '-1', '0', 'unknown', 'no'], ['33', 'entrepreneur', 'married', 'secondary', 'no', '2', 'yes', 'yes', 'unknown', '5', 'may', '76', '1', '-1', '0', 'unknown', 'no'],
['47', 'blue-collar', 'married', 'unknown', 'no', '1506', 'yes', 'no', 'unknown', '5', 'may', '92', '1', '-1', '0', 'unknown', 'no'], ['33', 'unknown', 'single', 'unknown', 'no', '1', 'no', 'no', 'unkno
wn', '5', 'may', '198', '1', '-1', '0', 'unknown', 'no']]
>>> no_subscribed = rdd3.filter(lambda x : x[16] == 'yes').count()
total_count = rdd3.count()
success_rate = (no_subscribed / total_count) * 100>>> total_count = rdd3.count()
>>> success_rate = (no_subscribed / total_count) * 100
>>> success_rate
11.698480458295547
>>> failure_rate = 100 - success_rate
>>> failure_rate
88.3015194170445
>>> ages = rdd3.map(lambda x : int(x[0]))
age = ag>>> max_age = ages.max()
age = ages.min()
avg_age = ages.mean()>>> min_age = ages.min()
>>> avg_age = ages.mean()
>>> print("Max Age :
  File "<stdin>", line 1
    print("Max Age :
          ^
SyntaxError: unterminated string literal (detected at line 1)
>>> print("Max Age : ", max_age)
Max Age :  95
>>> print("Min Age : ", min_age)
Min Age :  18
>>> print("Avg Age : ", avg_age)
Avg Age :  40.93621021432832
>>> print("Avg Age : ", round(avg_age,4))
Avg Age :  40.9362
>>>
```

```
>>> print(f"Average Balance : {avg_balance} \nMedian Balance: {median}")
Average Balance : 1362.2720576850763
Median Balance: 239.0
>>> age_rdd = rdd3.map(lambda x :(x[0],1))
_rdd_gro>>> age_rdd_grouped = age_rdd.reduceByKey(lambda a,b : a + b)
e_rdd_grouped_sorted = age_rdd_grouped.sortBy(lambda x : x[1], ascending=False)>>> age_rdd_grouped_sorted = age_rdd_grouped.sortBy(lambda x : x[1], ascending=False)
>>> age_rdd_grouped_sorted.take(10)
[('32', 2085), ('31', 1996), ('33', 1972), ('34', 1930), ('35', 1894), ('36', 1806), ('30', 1757), ('37', 1696), ('39', 1487), ('38', 1466)]
```

```
>>> def fun(x):
   if x ...     if x <20:
...         return 'Teenager'
...     elif x >=20 and x <40:
...         return 'Youngster'
  elif x...     elif x >=40 and x<60:
...         return 'MiddleAge'
se:
   ...     else:
...         return 'Senior'
...
>>> rdd.take(5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'rdd' is not defined. Did you mean: 'rdd1'?
>>> rdd3.take(5)
[['58', 'management', 'married', 'tertiary', 'no', '2143', 'yes', 'no', 'unknown', '5', 'may', '261', '1', '-1', '0', 'unknown', 'no'], ['44', 'technician', 'single', 'secondary', 'no', '29', 'yes', 'no
', 'unknown', '5', 'may', '151', '1', '-1', '0', 'unknown', 'no'], ['33', 'entrepreneur', 'married', 'secondary', 'no', '2', 'yes', 'yes', 'unknown', '5', 'may', '76', '1', '-1', '0', 'unknown', 'no'],
['47', 'blue-collar', 'married', 'unknown', 'no', '1506', 'yes', 'no', 'unknown', '5', 'may', '92', '1', '-1', '0', 'unknown', 'no'], ['33', 'unknown', 'single', 'unknown', 'no', '1', 'no', 'no', 'unkno
wn', '5', 'may', '198', '1', '-1', '0', 'unknown', 'no']]
>>> ages.take(5)
[58, 44, 33, 47, 33]
>>> ages_category = ages.map(lambda x : (fun(x), 1))
>>> ages_category.take(10)
[('MiddleAge', 1), ('MiddleAge', 1), ('Youngster', 1), ('MiddleAge', 1), ('Youngster', 1), ('Youngster', 1), ('Youngster', 1), ('MiddleAge', 1), ('MiddleAge', 1), ('MiddleAge', 1)]
>>> ages_category_grouped = ages_category.reduceByKey(lambda x, y : x + y)
>>> ages_category_grouped.collect()
[('Youngster', 23315), ('Teenager', 47), ('MiddleAge', 20065), ('Senior', 1784)]
>>> ages_category_grouped_sorted = ages_category_grouped.sortBy(lambda x : x[1], ascending=False)
>>> ages
ages                    ages_category           ages_category_grouped   ages_category_grouped_sorted
>>> ages_category_grouped_sorted
PythonRDD[38] at RDD at PythonRDD.scala:53
>>> ages_category_grouped_sorted.collect()
[('Youngster', 23315), ('MiddleAge', 20065), ('Senior', 1784), ('Teenager', 47)]
```

```
>>> marital_rdd = rdd3.map(lambda x :(x[2],1))
>>> marital_rdd_grouped = marital_rdd.reduceByKey(lambda a,b : a + b)
tal_rdd_grouped_sorted = marital_rdd_>>> marital_rdd_grouped_sorted = marital_rdd_grouped.sortBy(lambda x : x[1], ascending=False)
>>> marital_rdd_grouped_sorted.take(10)
[('married', 27214), ('single', 12790), ('divorced', 5207)]
>>> marital_age_rdd = rdd3.map(lambda x :((x[0],x[2]),1))
rital_age_rdd_>>> marital_age_rdd_grouped = marital_age_rdd.reduceByKey(lambda a,b : a + b)
l_age_rdd_groupe>>> marital_age_rdd_grouped_sorted = marital_age_rdd_grouped.sortBy(lambda x : x[1], ascending=False)
>>> marital_age_rdd_grouped.take(10)
[(('44', 'single'), 170), (('33', 'married'), 1075), (('33', 'single'), 746), (('42', 'divorced'), 184), (('41', 'divorced'), 175), (('29', 'single'), 683), (('53', 'married'), 657), (('57', 'married'),
 642), (('45', 'single'), 146), (('60', 'married'), 465)]
>>> marital_age_rdd_grouped_sorted.take(10)
[(('34', 'married'), 1131), (('35', 'married'), 1077), (('36', 'married'), 1076), (('33', 'married'), 1075), (('37', 'married'), 1073), (('31', 'single'), 1017), (('30', 'single'), 1012), (('32', 'marri
ed'), 1007), (('39', 'married'), 960), (('32', 'single'), 941)]
>>>
```