In [1]:
```
import os
import sys
spark_home = os.environ['SPARK_HOME'] = '/home/cloudera/Downloads/spark-2.
if not spark_home:
    raise ValueError('SPARK_HOME enviroment variable is not set')
sys.path.insert(0,os.path.join(spark_home,'python'))
sys.path.insert(0,os.path.join(spark_home,'python/lib/py4j-0.8.2.1-src.zip
execfile(os.path.join(spark_home,'python/pyspark/shell.py'))
```

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.0.1
      /_/

Using Python version 2.7.11 (default, Dec  6 2015 18:08:32)
SparkSession available as 'spark'.
```

ET 11

In [5]:
```
import numpy as np
from scipy.sparse import coo_matrix
import scipy.stats

control=[.5, .5, 3, 3, 3, 3, 3, 3, 4]
control.extend([0]*(100000-len(control)))
control=np.asarray(control)

# treatment=[ 0, 0, 0, 0, 0, 0, 0, 0, 0]
treatment=[1.5, .5, 0, 3, 4]
treatment.extend([0]*(100000-len(treatment)))
treatment=np.asarray(treatment)

#nb 2-sided test
scipy.stats.ttest_ind(control, treatment), scipy.stats.ttest_ind(control.a
```

Out[5]:
```
(Ttest_indResult(statistic=1.4142505069117979, pvalue=0.157289919856191
09),
 Ttest_indResult(statistic=1.3867952958906298, pvalue=0.165505737579078
8))
```

ET 14

In [4]:
```
RDD1 = sc.parallelize([(1, 2), (3, 4), (3, 6)])
RDD2 = sc.parallelize([(3, 9), (3, 6)])
print RDD1.join(RDD2).collect()
```

```
[(3, (4, 9)), (3, (4, 6)), (3, (6, 9)), (3, (6, 6))]
```

ET 16

In [6]:
```
rawSales = sc.textFile('beerSales.txt', use_unicode=False)
sales = (rawSales
         .map(lambda x: x.split('\t'))
         .filter(lambda x: x[0] != 'Week')
         .map(lambda x: float(x[5]))
        )

sales18Pk = sales.collect()

# Mean model is just mean of cases18pk sales
meanModel = sum(sales18Pk)/len(sales18Pk)
```

In [7]:
```
mape = (sales
        .map(lambda x: 100*abs(x - mm.value) / x)
       )

```

```
200.44422333
```

ET 17

In [9]:
```
from math import log

logSales = (sales
            .map(lambda x: log(x))
           )
```

```
[6.0844994130751715, 4.584967478670572, 4.248495242049359, 3.9512437185
814275, 4.1588830833596715]
```

In [10]:
```
logSales18Pk = logSales.collect()
meanModelLog = sum(logSales18Pk)/len(logSales18Pk)
```

In [11]:
```
mapeLog = (logSales
           .map(lambda x: 100*abs(x - mml.value) / x)
          )

```

```
19.5849342896
```

ET 18

In [22]:
```
from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD
from math import log

# Load and parse the data
def parsePoint(line):
    values = [x for x in line.split('\t')]
    if values[0] != 'Week':
        return LabeledPoint(log(float(values[5])), [log(float(values[1])),

data = sc.textFile('beerSales.txt')
parsedData = data.map(parsePoint).filter(lambda x: x != None)

print parsedData.take(10)

# Build the model
model = LinearRegressionWithSGD.train(parsedData, intercept=True)
```

```
[LabeledPoint(6.08449941308, [2.99473177322,2.64617479738,2.72063731661
]), LabeledPoint(4.58496747867, [2.99473177322,2.92584614609,2.72063731
661]), LabeledPoint(4.24849524205, [2.99473177322,2.92584614609,2.62972
823433]), LabeledPoint(3.95124371858, [2.99473177322,2.92584614609,2.55
178617863]), LabeledPoint(4.15888308336, [2.99473177322,2.92584614609,2
.5771819259]), LabeledPoint(4.27666611902, [2.99473177322,2.92584614609
,2.72063731661]), LabeledPoint(3.85014760171, [2.99473177322,2.92584614
609,2.63332665491]), LabeledPoint(4.44265125649, [3.00071981507,2.93012
651646,2.66861613186]), LabeledPoint(4.07753744391, [3.00171434523,2.93
119375242,2.62684014568]), LabeledPoint(4.14313472639, [3.00221123965,2
.93119375242,2.67414864943])]
```

In [21]:
```
valuesAndPreds = parsedData.map(lambda p: (p.label, model.predict(p.featur

mape18 = valuesAndPreds.map(lambda (v, p): 100 * abs(v - p) / v).reduce(la
print mape18
```

```
4.24146308056
```

In [ ]:

In [ ]: