All solutions for Final Project

I. **Part 1: create 2 jobs.**
   a. Job 1: return (k, v) = (code treatment, relative variance).
      i. Mapper: read and split data, return a list of all bill request by grouping each code treatment.
      ii. Reducer: calculate the relative variance by formula per each code treatment.
   b. Job 2: sort output Job 1 by relative variance descending.
      i. Mapper: read and split data, return relative variance as key and code treatment as value.
      ii. CustomComparabale: sort DoubleWritable descending.
      iii. Reducer: return top 3 code after sorting.
   c. Output: only 3 code treatment.

II. **Part 2: create 3 jobs**.
   a. Job 1: return (k, v) = (code insurance, bill request).
      i. Mapper: read and split data, return a list of all composite value (includes bill request and code insurance) by grouping each code treatment.
      ii. Reducer: find maximum of bill request of each code treatment as output value and return corresponding code insurance as output key.
   b. Job 2: return (k, v) = (code insurance, number of appearance of each code).
      i. Mapper: read and split data, return a list of value equals 1 when grouping each code insurance.
      ii. Reducer: count the number of appearance per each code insurance.
   c. Job 3: sort output Job 2 by the number of appearance.
      i. Mapper: read and split data, return number of appearance as key and code insurance as value.
      ii. CustomComparable: sort IntWritable descending.
      iii. Reducer: return top 3 code after sorting.
   d. Output: only 3 code insurance.

III. **Part 3: create 4 jobs**.
   a. Job 1: return (k, v) = (composite key, average of bill request). Composite key includes code treatment, city name and state name.
      i. Mapper: read and split data, return a list of all bill request by grouping composite key (by code treatment, city, state).
      ii. Reducer: calculation the average of bill requests per each composite key.

b. Job 2: return (k, v) = (composite key, value). Composite key includes city and state name. Value equals 1 corresponding with maximum value of all average bill requests.
    i. Mapper: read and split data, return a list of all average bill request by grouping composite key (by city and state).
    ii. Reducer: find maximum of average bill requests, return 1 as value and corresponding city and state as key.
c. Job 3: return (k, v) = (composite key, number of appearance of each composite key). Composite key includes city and state name.
    i. Mapper: read and split data, return a list of value equals 1 when grouping each composite key.
    ii. Reducer: count the number of appearance per each composite key.
d. Job 4: sort output Job 3 by the number of appearance.
    i. Mapper: read and split data, return number of appearance as key and composite key (city and state name) as value.
    ii. CustomComparable: sort IntWritable descending.
    iii. Reducer: return top 3 composite key (city and state name) after sorting.
e. Output: only 3 city and state name.

IV. Part 4: create 3 jobs.

a. Job 1: return (k, v) = (code treatment, composite value). Composite value includes the max of bill gap (the gap between bill request and bill actual) per each code treatment with corresponding code insurance.
    i. Mapper: read and split data, return a list of all bill gaps by grouping code treatment.
    ii. Reducer: calculation the maximum of gap between bill request and bill actual per each code treatment and return it with corresponding code insurance as output value. The output key is code treatment.
b. Job 2: return (k, v) = (code insurance, number of appearance of each code insurance).
    i. Mapper: read and split data, return a list of value equals 1 when grouping each code insurance.
    ii. Reducer: count the number of appearance per each code insurance.
c. Job 3: sort output Job 2 by the number of appearance.
    i. Mapper: read and split data, return number of appearance as key and code insurance as value.
    ii. CustomComparable: sort IntWritable descending.
    iii. Reducer: return top 3 code insurance after sorting
d. Output: only 3 code insurance.