Visualization of Hostel Data in Japan

Purpose

I would like to deal with data on hostels in Japan. The most important characteristic of hostels is the low price of accommodation. Since hostels are mainly used by people who want to save money, they are not suitable for those who want to stay in luxurious rooms. However, since hostel facility is being overgentrified in recent years, the number of Japanese staying Japan who are interested in experiencing cheap, expensive, pricey and the changing trends of domestic travelers who want to spend less on travel expenses and more on human interaction hostels and hotels are also increasing. Although there are many hostels in Japan, there are few hostels that are suitable for people who want to stay for a long time. Hostels have a flexibility, such as dormitory rooms and high quality rooms, hostels provide a place where people can meet and interact with other users. For example, while hostels provide individual lockers, hostels have a lot of shared space, and even bathrooms are often shared with other users. In this way, hostels provide a place for people to meet and interact, and the differences between rooms in such a place help travelers, creating a "space".

Data set

https://www.google.com/maps/d/u/0/edit?hl=en&mid=1ynxW-0ZNbqbbb7ZuvqAA-ymXY&ll=0&z=2

I offered a hostels. This data can be used to help Japanese man and the owners of the data to HostelWorld.com. HostelWorld.com is the primary hostel booking website for Japan. Although there are many hostels in Japan, there are few hostels that are suitable for people who want to stay for a long time. HostelWorld.com complements hostels with other features, such as Japanese-style accommodation and public utilities, for which tourist agents. The data has amassed 889 records and has 12 columns such as City, Distance from city center, price, score, etc.

```
$ head -5 hostel.csv
$ wc -l hostel.csv
$ head -1 hostel.csv | sed -e 's/,/\n/g'
```

```
$ head -5 hostel.csv
id,name,city,price,score,reviews,summary,distance,stars,dorm,room,active
0,Tokyo Cheap Hostel,Tokyo,3000,9.0,Superb,4.5,0.4,5.0,0.0,0.0,0.0,1.0
1,Hostel Kyoto,Kyoto,3000,9.0,Fabulous,4.5,Kawaguchi,4.5,0.4,5.0,0.0,0.0,0.0,1.0
2,Hostel Tokyo,Tokyo,3000,8.7,Fabulous,4.5,Shibuya,4.5,0.4,5.0,0.0,0.0,0.0,1.0
3,Hostel Kyoto,Kyoto,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
4,Hostel Tokyo,Tokyo,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
$ wc -l hostel.csv
889 hostel.csv
$ head -1 hostel.csv | sed -e 's/,/\n/g'
id
name
city
price
score
reviews
summary
distance
stars
dorm
room
active
```

```
$ cat -n hostel.csv | head -5
     1	id,name,city,price,score,reviews,summary,distance,stars,dorm,room,active
     2	0,Tokyo Cheap Hostel,Tokyo,3000,9.0,Superb,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     3	1,Hostel Kyoto,Kyoto,3000,9.0,Fabulous,4.5,Kawaguchi,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     4	2,Hostel Tokyo,Tokyo,3000,8.7,Fabulous,4.5,Shibuya,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     5	3,Hostel Kyoto,Kyoto,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     6	4,Hostel Tokyo,Tokyo,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
```

Task1:Which city has the best hostels available

So Why are you pursuing this topic?

As Tokyo ranks in Japan's top ten cities where we travel often, as a quiet, Japanese province, I would like to know which area is doing the best.

So how is the task done?

As the number of high scoring hostels are present. Tokyo Kinki all is ranking a rotto, but a simple number of cases is considered more convincing.

```
$ ruby -r csv -e 'puts CSV.read(ARGV[0], headers: true).group_by{|r| r["city"]}.map{|c,rs| [c, rs.count, rs.sum{|r| r["score"].to_f}/rs.count].sprintf("%-15s %3d %.2f", *)}.sort_by{|l| -l[-4,4].to_f}.first(5).puts(?\n)' hostel.csv
Tokyo           98 8.72
Kyoto           89 8.69
Osaka           89 8.69
Hiroshima       89 8.69
Fukuoka City    89 8.69
```


Result1

- Tokyo has the highest number of hostels with a high score in every corner of a map. Kyoto and Osaka are tied with 89 hostels and 8.69 scores and Hiroshima and Fukuoka City are tied with exactly 89 hostels and a score of 8.69. However, Tokyo is the most likely place to stay in.

Task2:Understanding the characteristics of hostels in each city

So Why are you pursuing this topic?

As Kansai is such a tourist-friendly area, I want to know what kind of hostels are there and what are the differences in hostels for each region. This analysis may help my travelers.

So how is the task done?

As I want to know what kind of hostels are in each area, I will use the following techniques to understand the characteristics of hostels in each city.

- Hotel Style
- Hotel Size

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  group_by{|r| r["name"].scan(/\w+|[^\w\s]+|\s+/).
    select{|w| w[/\A\w/] && w[/\A\S/] == w[/\A\S/].upcase}.
    sort_by{|w| [-w.size, w.downcase]}.
    first(2).
    map{|w| w.gsub(/[^\w\s]+/, "").gsub(/\s+/, "_").
      gsub(/([a-z\d])([A-Z])/, "\1_\2").
      gsub(/([A-Z]+)([A-Z][a-z])/, "\1_\2").
      downcase.
      gsub(/\A[\d_]+|[\d_]+\z/, "").
      gsub(/[\d_]{2,}/, "_")}.
    flatten.
    tally.
    sort_by{|w,c| [-c, w]}.
    first(5).
    map{|w,c| [w, "#{c*100/98}%"]}.
    ljust(20, [?-, ?-]).
    puts(?\n)" hostel.csv
hostel 100%
tokyo  100%
cheap  100%
kyoto  100%
hostel 100%
$ cat -n hostel.csv | head -5
     1	id,name,city,price,score,reviews,summary,distance,stars,dorm,room,active
     2	0,Tokyo Cheap Hostel,Tokyo,3000,9.0,Superb,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     3	1,Hostel Kyoto,Kyoto,3000,9.0,Fabulous,4.5,Kawaguchi,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     4	2,Hostel Tokyo,Tokyo,3000,8.7,Fabulous,4.5,Shibuya,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     5	3,Hostel Kyoto,Kyoto,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     6	4,Hostel Tokyo,Tokyo,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
```


Result2

- A similar pair of summary scores and the respective scores is shown in the graph above. Overall, Hiroshima and Fukuoka are clustered in the same area, and Tokyo and Kyoto are in the same area. However, Tokyo and Kyoto are in a different location in scores. However, it is difficult to understand the characteristics of each city, so we will move on to the next graph.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  sort_by{|r| -r["price"].to_f}.
  map{|r| [r["name"], r["price"], r["score"], r["reviews"], r["summary"], r["distance"], r["stars"], r["dorm"], r["room"], r["active"]]}.
  sort_by{|l| [l[0].downcase, l[1]]}.
  map{|l| [l[0], l[1].ljust(4), l[2].ljust(3), l[3].ljust(4), l[4].ljust(8), l[5].ljust(4), l[6].ljust(3), l[7].ljust(3), l[8].ljust(3), l[9].ljust(3)]}.
  puts(?\n)" hostel.csv
Akihabara  3000 9.0 Superb  4.5 Kawaguchi 4.5 0.4 5.0 0.0 0.0 0.0 1.0
Akihabara  3000 8.7 Fabulous 4.5 Shibuya  4.5 0.4 5.0 0.0 0.0 0.0 1.0
Akihabara  3000 8.6 Fabulous 4.5 Fine     4.5 0.4 5.0 0.0 0.0 0.0 1.0
Akihabara  3000 8.6 Fabulous 4.5 Fine     4.5 0.4 5.0 0.0 0.0 0.0 1.0
Akihabara  3000 8.5 Superb   4.5 Fine     4.5 0.4 5.0 0.0 0.0 0.0 1.0
$ cat -n hostel.csv | head -5
     1	id,name,city,price,score,reviews,summary,distance,stars,dorm,room,active
     2	0,Tokyo Cheap Hostel,Tokyo,3000,9.0,Superb,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     3	1,Hostel Kyoto,Kyoto,3000,9.0,Fabulous,4.5,Kawaguchi,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     4	2,Hostel Tokyo,Tokyo,3000,8.7,Fabulous,4.5,Shibuya,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     5	3,Hostel Kyoto,Kyoto,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
     6	4,Hostel Tokyo,Tokyo,3000,8.6,Fabulous,4.5,Fine,Good,4.5,0.4,5.0,0.0,0.0,0.0,1.0
```


Result2

- In recent years, Hiroshima and Fukuoka have the high scoring hostels with less variation in all factors. The only difference is the Facilities. Hostels in Hiroshima and Fukuoka have the high scoring hostels with less variation in all factors. However, Tokyo, Kyoto, and Osaka have the most variation in facilities. Hostels in Hiroshima and Fukuoka have the high scoring hostels with fewer scores for security and staff and less overall, suggesting that hostels are a safe place to stay. In Kyoto, Kyoto is little variation in Cleanliness scores, and the hostelling seem to be well cared for, as the value is the same as Hiroshima and Fukuoka.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  sort_by{|r| -r["price"].to_f}.
  map{|r| [r["name"], r["price"], r["score"], r["reviews"], r["summary"], r["distance"], r["stars"], r["dorm"], r["room"], r["active"]]}.
  sort_by{|l| [l[0].downcase, l[1]]}.
  map{|l| [l[0], l[1].ljust(4), l[2].ljust(3), l[3].ljust(4), l[4].ljust(8), l[5].ljust(4), l[6].ljust(3), l[7].ljust(3), l[8].ljust(3), l[9].ljust(3)]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
```


Result3

- Tokyo has the largest values for distance from the city and price, which is a reasonable result since Tokyo is the largest city. Only Osaka, Kyoto, and Hiroshima have the lowest distance from the city and price.

Task3:Mapping of hostel data to map information

So Why are you pursuing this topic?

As it addition is mapping a hotel and hostel locations so that we can see ground truth on, you'll know that this will help travelers when looking for hostels with good scores.

So how is the task done?

As I want to know the hostel data, I'm going to use the data to plot on a map based on latitude and longitude. also looked at the Kyoto for such city as it is more detailed look at the mapping data.

Example

The Price Score Index is a scatter plot of latitude and longitude of the entire hostel data. The second figure is a plot of the data on a map using Leaflet.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Kyoto"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
```


Result3

- Kyoto has the largest values for distance from the city and price, which is a reasonable result since Tokyo is the largest city. Only Osaka, Kyoto, and Hiroshima have the lowest distance from the city and price.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Kyoto"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
```


Result: Tokyo

- Below is a map mapping hostel data for Tokyo. It seems no difference in good and not so good hostels depending on location.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+35.6895 +139.6917 3000 9.0 Tokyo Cheap Hostel
+35.6895 +139.6917 3000 8.7 Hostel Tokyo
+35.6895 +139.6917 3000 8.6 Hostel Tokyo
+35.6895 +139.6917 3000 8.6 Hostel Tokyo
+35.6895 +139.6917 3000 8.5 Hostel Tokyo
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+35.7101 +139.7966 3000 9.0 Hostel Tokyo
+35.7101 +139.7966 3000 8.7 Hostel Tokyo
+35.7101 +139.7966 3000 8.6 Hostel Tokyo
+35.7101 +139.7966 3000 8.6 Hostel Tokyo
+35.7101 +139.7966 3000 8.5 Hostel Tokyo
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+35.7101 +139.7966 3000 9.0 Hostel Tokyo
+35.7101 +139.7966 3000 8.7 Hostel Tokyo
+35.7101 +139.7966 3000 8.6 Hostel Tokyo
+35.7101 +139.7966 3000 8.6 Hostel Tokyo
+35.7101 +139.7966 3000 8.5 Hostel Tokyo
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Tokyo"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+35.7101 +139.7966 3000 9.0 Hostel Tokyo
+35.7101 +139.7966 3000 8.7 Hostel Tokyo
+35.7101 +139.7966 3000 8.6 Hostel Tokyo
+35.7101 +139.7966 3000 8.6 Hostel Tokyo
+35.7101 +139.7966 3000 8.5 Hostel Tokyo
```


Result: OSAKA

- Below is a map mapping hostel data for OSAKA. Osaka is the southern region somewhat overall, while those in the northern region scored higher. Therefore, if there is no particular reason, the north side is recommended.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Osaka"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+34.7025 +135.7590 3000 9.0 Hostel Osaka
+34.7025 +135.7590 3000 8.7 Hostel Osaka
+34.7025 +135.7590 3000 8.6 Hostel Osaka
+34.7025 +135.7590 3000 8.6 Hostel Osaka
+34.7025 +135.7590 3000 8.5 Hostel Osaka
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Osaka"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+34.7025 +135.7590 3000 9.0 Hostel Osaka
+34.7025 +135.7590 3000 8.7 Hostel Osaka
+34.7025 +135.7590 3000 8.6 Hostel Osaka
+34.7025 +135.7590 3000 8.6 Hostel Osaka
+34.7025 +135.7590 3000 8.5 Hostel Osaka
```


Result: KYOTO

- Overall, the non-center scores better.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Kyoto"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+35.0116 +135.7681 3000 9.0 Hostel Kyoto
+35.0116 +135.7681 3000 8.7 Hostel Kyoto
+35.0116 +135.7681 3000 8.6 Hostel Kyoto
+35.0116 +135.7681 3000 8.6 Hostel Kyoto
+35.0116 +135.7681 3000 8.5 Hostel Kyoto
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Kyoto"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+35.0116 +135.7681 3000 9.0 Hostel Kyoto
+35.0116 +135.7681 3000 8.7 Hostel Kyoto
+35.0116 +135.7681 3000 8.6 Hostel Kyoto
+35.0116 +135.7681 3000 8.6 Hostel Kyoto
+35.0116 +135.7681 3000 8.5 Hostel Kyoto
```


Result: HIROSHIMA

- Overall scores are good, but the scores that scores of the hostels are out of the center of the city.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Hiroshima"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+34.3963 +132.4596 3000 9.0 Hostel Hiroshima
+34.3963 +132.4596 3000 8.7 Hostel Hiroshima
+34.3963 +132.4596 3000 8.6 Hostel Hiroshima
+34.3963 +132.4596 3000 8.6 Hostel Hiroshima
+34.3963 +132.4596 3000 8.5 Hostel Hiroshima
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Hiroshima"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+34.3963 +132.4596 3000 9.0 Hostel Hiroshima
+34.3963 +132.4596 3000 8.7 Hostel Hiroshima
+34.3963 +132.4596 3000 8.6 Hostel Hiroshima
+34.3963 +132.4596 3000 8.6 Hostel Hiroshima
+34.3963 +132.4596 3000 8.5 Hostel Hiroshima
```


Result: FUKUOKA

- The overall score is good, and there are no hostels out of the center, so it seems safe to choose any of them.

```
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Fukuoka City"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+33.5902 +130.4207 3000 9.0 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.7 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.6 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.6 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.5 Hostel Fukuoka City
$ ruby -r csv -e '
CSV.read(ARGV[0], headers: true).
  select{|r| r["city"] == "Fukuoka City"}.
  map{|r| [r["latitude"].to_f, r["longitude"].to_f, r["price"].to_f, r["score"].to_f, r["name"]]}.
  select{|l,*| l[0].abs > 0}.
  sort_by{|l,*| -l[0]}.
  map{|l,*| l[0..3].sprintf("%+8.4f %+8.4f %4.0f %3.1f", *l[0..3]) + " " + l[4]}.
  puts(?\n)" hostel.csv
$ cat -n hostel.csv | head -5
+33.5902 +130.4207 3000 9.0 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.7 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.6 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.6 Hostel Fukuoka City
+33.5902 +130.4207 3000 8.5 Hostel Fukuoka City
```


Summary and Discussion

Last Summary: Hostels in Japan's Top Tourist Hotel Map

- Can't Which city has the best hostels available
  - The number of high scoring hostels are present in Tokyo and Kyoto and it isn't changed in a Kan ganji so that this current scheme is 100 hostels in the area of a person.
- Understanding the characteristics of hostels in each city
  - By comparing hostel styles for northern and southern Kyoto, Kyoto, Hiroshima, and Fukuoka, we understand that there are differences in hostel facilities.
  - Hostels in Hiroshima and Fukuoka are characterized by the high score of security and staff, suggesting that hostels are a safe place to stay. In Kyoto, Kyoto is little variation in Cleanliness scores, and the hostelling seem to be well cared for, as the value is the same as Hiroshima and Fukuoka.
- Mapping of hostel data to map information
  - Least Mapped to hostels map information. Looks looked at the Kyoto for such city as it is more detailed look Kyoto Kyoto.
  - Hostels in Tokyo and Osaka have a wide variation in facilities, and there is no difference in good and bad.
    - Osaka: Osaka is the southern region scored lower overall. Little Kyoto is the northern region scored higher. Therefore, if there is no particular reason, the north side is recommended.
    - Tokyo: Tokyo is the capital of Japan, and there are many hostels in the center, so it seems safe to choose any of them.
    - KYOTO: Overall, Kyoto is good, and there are no hostels out of the center, so it seems safe to choose any of them.
    - HIROSHIMA: Overall scores are good, but it seems that some of the hostels are out of the center of the city.
    - Fukuoka: The overall score is good, and there are no hostels out of the center, so it seems safe to choose any of them.

Hostels in Japan's Top Tourist Hotel Map: Hostels in Japan's top tourist hotel map. Hostels are a place where people who want to save money can stay. Hostels have the goals and the rest of the hostel experience, creating a community that intimately deepens the understanding of hostels in Japan. By providing this information to travelers, it is hoped that they will be able to make a better decision when choosing a hostel.

This time, I tried to analyze hostel data, which is a relatively small data, using Linux commands. Linux tools are not very good, so I decided to use the notebook. In the next time, I would like to perform the application in the available scripts.