# Study about traveling sales person problem

## 1. Introduction

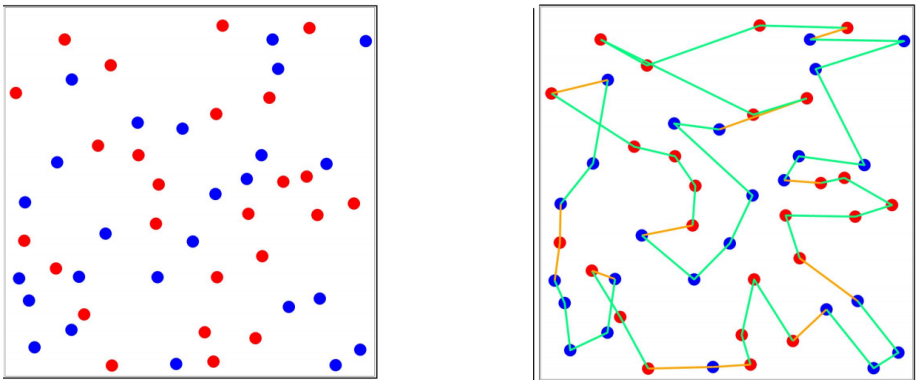
Traveling sales person problem (TSP) is one of the most famous optimization problems. TSP is NP-hard, therefore there is no algorithm that can find an optimal solution in linear time (unless P=NP). Surprisingly humans are able to solve the visual TSP in linear time and near optimally. Now, the natural question is what algorithms human use to solve this problem and if we can apply this algorithm in solving optimization problems.

It is long that cognitive scientists are trying to design human- subject experiments on TSP to find the underlining human produced algorithms. There is a known hypothesis that says TSP is solved in human visual system and it does not engage other parts of problem solving in the brain. If TSP is solved in visual system then human produced algorithms should follow the hierarchical shape of the visual system, and it should follow the limitations of the visual system. Human visual system is not connected to Dopamine neurons; therefore, we do not observe learning in TSP. Previous human-subject TSP experiments showed this case and they observe no improvement in performance after practice. In the experiment that we are analyzing in this project, we are following another hypothesis in this topic: If humans use their visual system for solving TSP, they should only be able to solve Euclidean TSP near optimally (because our visual system is evolved in a Euclidean space and works more efficiently in this space). Therefore, if we change TSP into a non-Euclidean space human performance will drop because visual system is no longer able to perform well in processing distances. Furthermore, what we are looking at in this project is as follows: if the visual system no longer is able to perform well in solving this non-Euclidean version of TSP, then other parts of the brain likely to engage in solving this type of TSP. Therefore, unlike the Euclidean TSP, we might be able to learn and improve our performance through consequative trials and by receiving feedback.

Experiment:

32 participant subjects.

There are 30 maps each map has 50 points (some points are blue and some are red) (See Figure 1A).



A B

Figure 1

Each map has an ID and each subject solve all thirty maps consequatively in random order. One sample of solved map is shown in Figure 1B. (These graphs are related to an old experiment and they are not produced by R, but the rest of graphs and analysis are done in R)

Goal: answering to the following questions:

1. How does the performance of subjects change with time passing.
2. Are there apparently difference in performance among subjects?
3. Are maps (which are random 50 points) different in difficulty?

We divide the coding of this project in two parts:

1. Tidy data
2. Analyzing data and Making appropriate graphs.

# 2. Tidy Data (code is attached in appendix A)

For convenient in further analysis, we should provide a wide table for statistic approaches next step. Each row will describes a trial, which taken by a specific subject on a specific map in specific time order. Every trial could yields a real score for goodness, which based on euclidean distance with weight adjustment for color switching in a two-dimensional surface. Map has a baseline best score in theory, therefore we can calculate the difference score between real score and best score. To describe a real person's performance on a map, we defined an error ratio value from difference score to baseline score, which is key value in our next step analysis.

The example of wide table:

| map_id <fctr> | subject_id <fctr> | real_score <dbl> | best_score <dbl> | err_ratio <dbl> | order_1_to_30 <dbl> | block_1_to_6 <fctr> |
|---|---|---|---|---|---|---|
| 1 | 1 | 9206.366 | 7646 | 0.20407607 | 4 | 1 |
| 2 | 1 | 8554.393 | 6476 | 0.32093777 | 28 | 6 |
| 3 | 1 | 9017.831 | 7172 | 0.25736627 | 22 | 5 |
| 4 | 1 | 10974.350 | 7028 | 0.56151819 | 25 | 5 |
| 5 | 1 | 8411.277 | 6999 | 0.20178267 | 10 | 2 |
| 6 | 1 | 7707.977 | 6866 | 0.12262987 | 26 | 6 |
| 7 | 1 | 7897.058 | 7239 | 0.09090452 | 6 | 2 |
| 8 | 1 | 7665.555 | 7095 | 0.08041651 | 29 | 6 |
| 9 | 1 | 8222.953 | 7026 | 0.17036056 | 20 | 4 |
| 10 | 1 | 10643.419 | 7208 | 0.47661191 | 16 | 4 |

Figure 1: Snapshot for wide table

We have 4 steps to get the above table. That is, read real score from trial data files for every subjects on each maps, read best score for each maps, calculate error ratio, read the experiment map arrangements order for each subjects and divide them into 6 groups.

Finally, I will show the graphically impressions for scores from different map and different subjects.

## 2.1 Read real score

There are 9 computers that subjects sit and solve problems on. In each computer, a folder with the name result_(date)_(subject number)_(computer number) is saved which contains information about all 30 maps that subjects solve and in each map's tour file(TSPX) the path that the subject follows is determined (Figure 2)

```
<?xml version="1.0" encoding="utf-8"?>
<TspDocument>
    <MapSize width="900" height="900" />
    <Cities />
    <Tours>
        <Tour Name="User Tour" IsOptimal="False">
            <City x="19" y="668" group="1" />
            <City x="44" y="724" group="1" />
            <City x="58" y="842" group="1" />
            <City x="151" y="798" group="1" />
            <City x="183" y="759" group="0" />
            <City x="253" y="888" group="0" />
            <City x="487" y="804" group="0" />
            <City x="615" y="819" group="0" />
            <City x="509" y="878" group="0" />
            <City x="415" y="884" group="1" />
            <City x="699" y="740" group="1" />
            <City x="817" y="887" group="1" />
            <City x="879" y="848" group="1" />
            <City x="777" y="719" group="1" />
            <City x="368" y="665" group="1" />
```

Figure 2: Example for tspx file

For example in Figure 2, it shows that the subject on a map followed the following path.

1. Starts with Point with coordinates (19,668), that has color 1(red).
2. Then Point (44,734) with color red
3. And so on

Therefore, we need follow this process to be able to extract all the raw data we need.

1. Finding all folders with the format "result_(date)_(subject number)_(computer number)": because it was the only folder with that contains ''result'' word. Then by using grep we found all these folders.
2. We read each tspsx file (map's tour file), and we found all digits inside (it is some digits on first lines that we need, then it is digits related to coordinates and a digit (1 or 0) that represent the color of points)
3. Then, we need to read from the file that contains optimal length.

Firstly, we need go through the directories and match file name and get map_id and subject_id by file name pattern.

We use grep(), list.dirs(), readChar(), paste() xmlChildren(), xmlAttrs() methods in tidyverse/XML library, to handle the data reading. Data.frame/vector collections such as used for data store used. Functions used to organize the control flow and well constructed the structure for our logic process. Also, loop used to go though all the files.

## 2.2 Read best score

Then every best score in a single file, we read it to build a mapping relationship from map id to its best score.

## 2.3 Calculate the error ratio

The score has a fumula that:

$$\text{score} = \sum_{\text{line i in link order}} [\sqrt{\delta_x^2 + \delta_y^2} * w(i)]$$

$w(i) = 1$ if color$(i)$ same as color$(i-1)$; 2 if color$(i)$ not same as color$(i-1)$

After extracting all the required information about points and optimal tours from initial files, now we calculate the length of the tours for each subject for each map. We take to consideration to double the distance if the two connected points were not in the same color. The output of this part is two excel files, both files have 30 rows (representing 30 maps), and 32 columns(representing subjects ), In one file, we save the length of the tour that each subject take for each map, and in the other file, we saved the corresponding accuracy of each subject(measured as performance), which is calculated as:

$$\text{error ratio} = \frac{\text{real score} - \text{best score}}{\text{best score}}$$

## 2.4 Calculate the time order as block

We read the map order for subjects, so we divide them into 6 groups, we'll consider it as a potential factor which maybe effects the error ratio.

## 2.5 Graphically impression

The core question we care is that, how subject and map detriments the final performance of a trial, thus, I will show the part of real map and subject's real solutions as well as their scores and best scores for each maps.

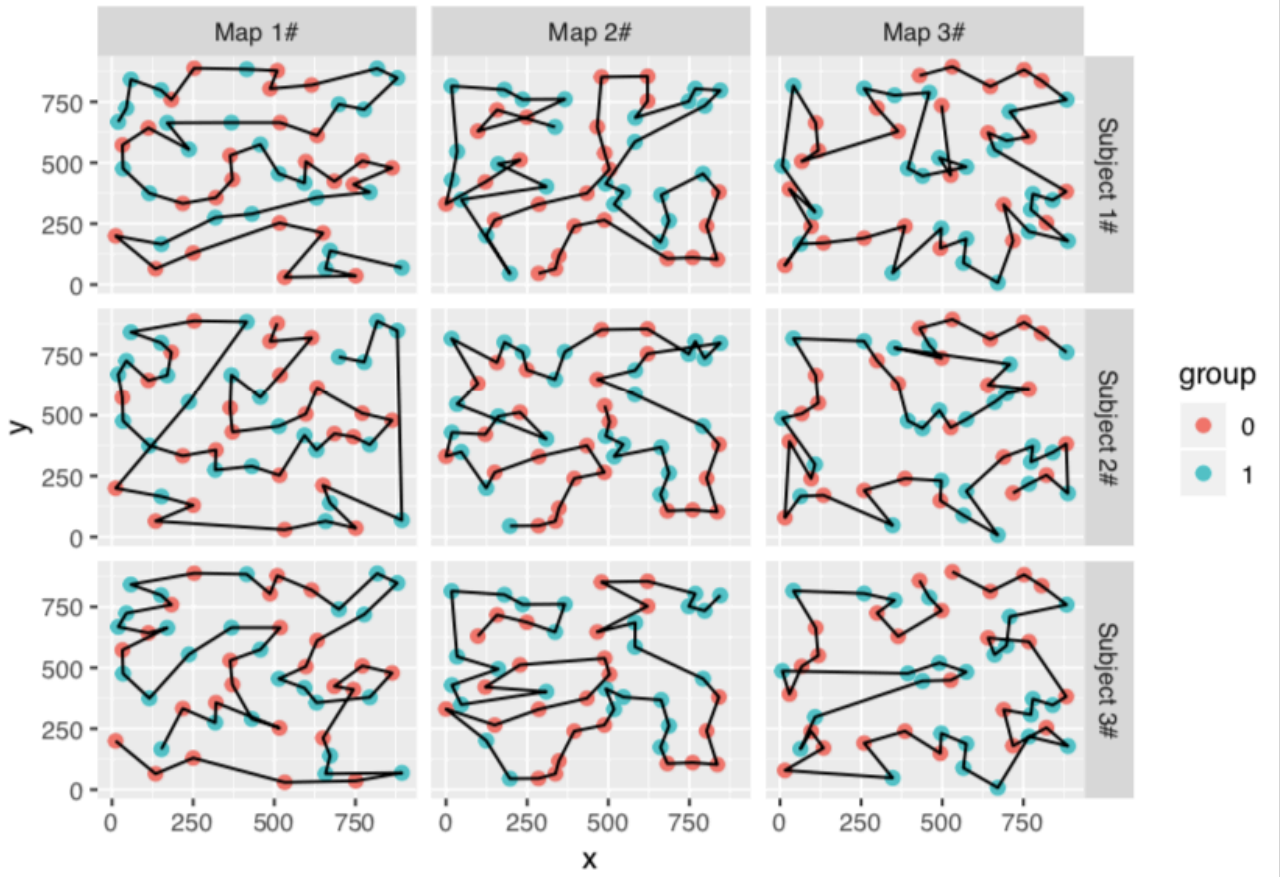



Figure 3: Real solutions for Subject 1 to 3 on Map 1 to 3

We use can see a map for each column, for example, map #1 in column 1. Three different linkage solution drawn by 3 subject. For sure, they will get different scores.

| | Map #1 | Map #2 | Map #3 |
|---|---|---|---|
| Subject #1 | 9206.366 | 8554.393 | 9017.831 |
| Subject #2 | 11073.808 | 9203.150 | 10059.082 |
| Subject #3 | 9606.208 | 8770.109 | 8182.654 |

Figure 4: Scores for Subject 1 to 3 on Map 1 to 3

Each map have different best score for maps, we should compare to.

| | Map #1 | Map #2 | Map #3 |
|---|---|---|---|
| Best Score in Theory | 7646 | 6476 | 7172 |

Figure 5: Best Scores for Map 1 to 3

Finally we have the error ratio for them:

| | Map #1 | Map #2 | Map #3 |
|---|---|---|---|
| Subject #1 | 0.2040761 | 0.3209378 | 0.2573663 |
| Subject #2 | 0.4483138 | 0.4211164 | 0.4025491 |
| Subject #3 | 0.2563704 | 0.3542478 | 0.1409167 |

Figure 6: Error Ratios for Subject 1 to 3 on Map 1 to 3

## 3. Data Analysis (code is attached in appendix B)

First, we divide data to blocks. Each 5 maps that subjects solve will be averaged and considered as one block. It help us to have more clear graphs with less variance. With the block number increasing, it could measure the performance change of people as time passing.

## 3.1. Draw conclusion from plot

The goal is answering to the following questions:

1. How does the performance of subjects change in blocks.

This graph (see Figure 3)should be helpful in understanding if subjects in average could improve their performance (reducing error) from the first block to the last block. The red line means the average performance of each block, so it shows that the error does not change a lot in average for subjects. And the performance shows a slowly down trend from block 1 to block 3, while shows a up trend from block 3 to block 4, and down trend from block 4 to block 6. That means the subjects' performance is increasing from when reading the first 15 maps, decreasing when reading the 16$^{th}$ to 20$^{th}$ images, but still increasing when reading the remaining images. But the trends are all not so apparently.

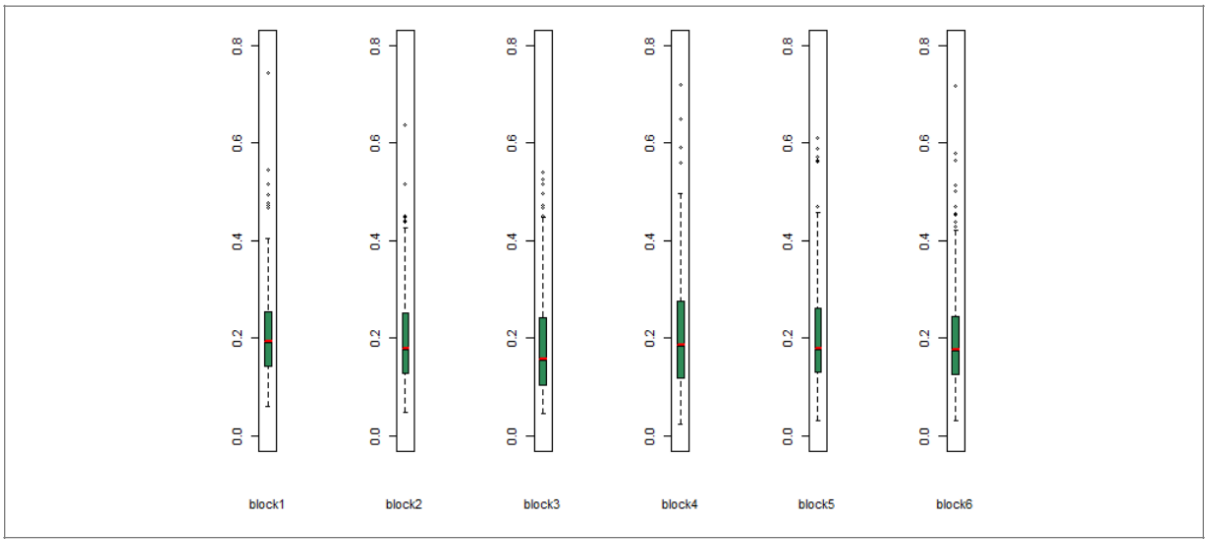


Figure 3. Box-plot of performance in Block 1-6

2. Do we have subjects who learn and the ones who does not.

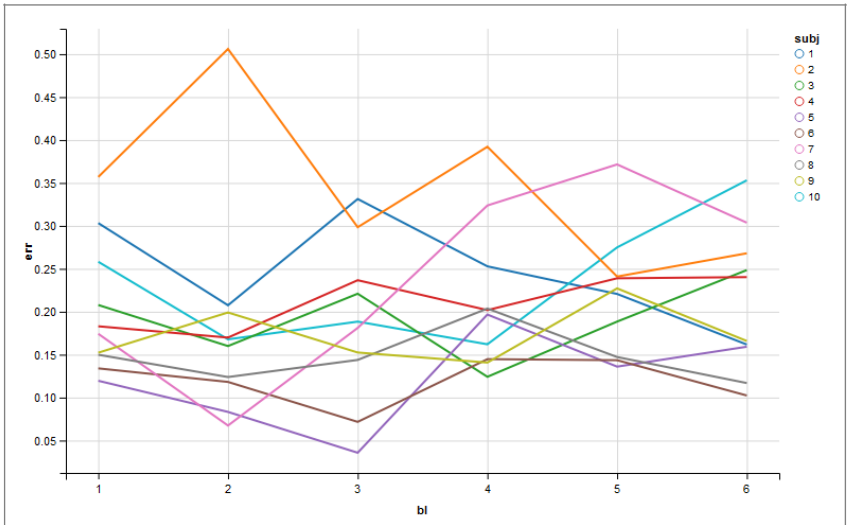



Figure 4. the performance of subjects

We draw performance responding based on block line for each subject (see Figure 4), and as sample, we chose first 10 subjects and draw their corresponding error lines (error in each block for each subject). If they learned the line should show a negative slope, but we do not see a descending trend, therefore it is not true that some subjects learned and some don't.

We also make another graph for each subject; we assigned a lighter color to first blocks that subjects solved and a darker to the later blocks. So, we have six points for each subject that shows how much error they made in each block (Figure 5). If we saw darker colors on bottom and lighter on top, it would suggested that subjects learned and they made less error in final maps that they saw. However, as can be seen in Figure 5, we cannot see a pattern for colors, therefore this graph also shows that people have not learned.

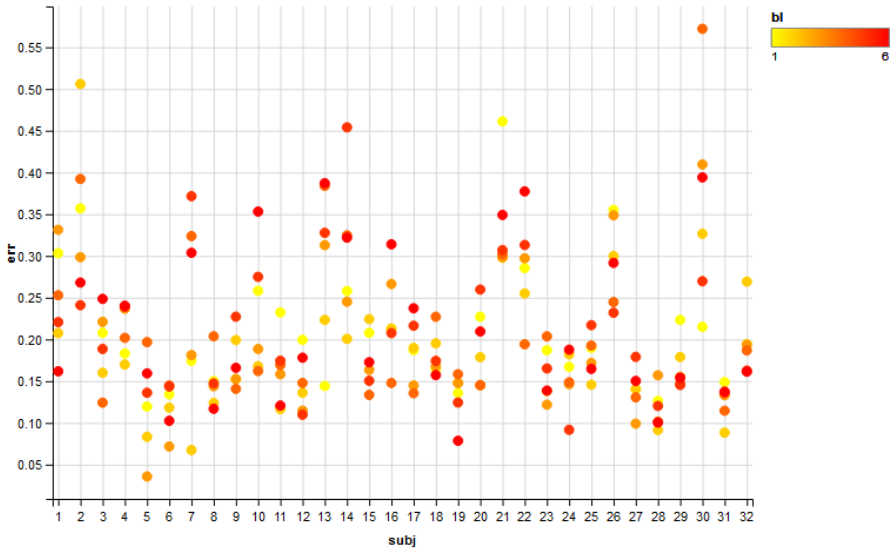



Figure 5. Performance of subjects

3. Do we have maps that are learnable and subject did well on them in last trials in comparison to first trials?

For answering this question we need to extract information on each single map in each block for all subjects. The result is shown in Figure 6. Similar to Figure 5, we cannot see a pattern of color here, therefore, we cannot see any difference in maps. Maps were generated randomly of 50 points and colors were randomly assigned to each point, therefore, we also expect not to see any significant difference between maps, as we see here.

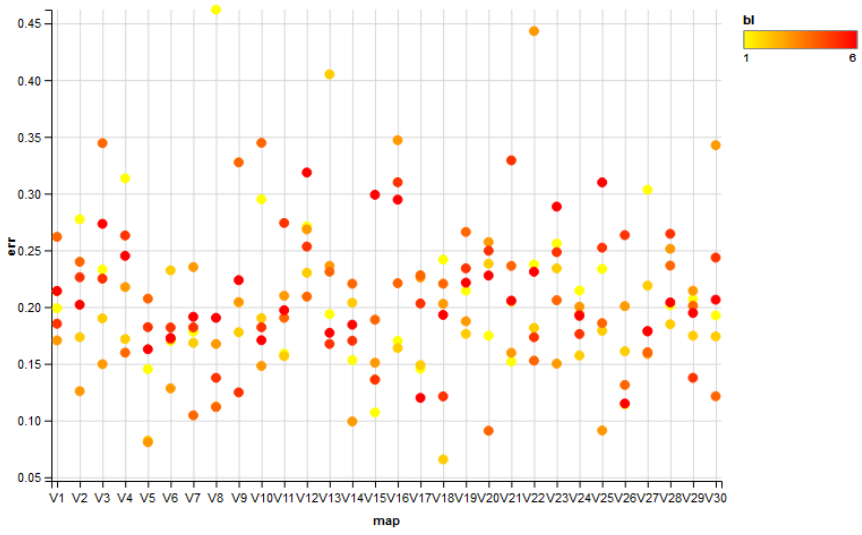



Figure 6. Performance difference among maps

We also tried to see the performance of all maps as lines in one graph as can be seen below:

Each separated color in Figure 7 represents one map, and for each map we have 6 points that they are connected to each other. Each point represent the average accuracy of the map when it appears in a block.

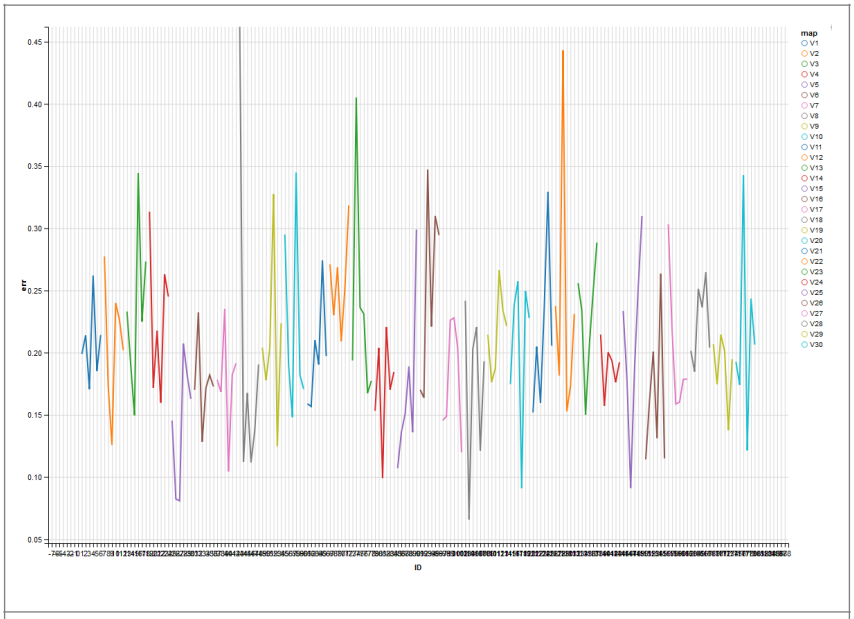



Figure 7. Performance for each subjects

Here, also we cannot see lines that are descending, instead we see lines with high variance. Therefore, we cannot make any conclusion about any map and claim that some maps were learnt and some did not.

## 3.2 Statistical modeling

### 3.2.1 Linear regression

We want to see if statistical model could help us draw conclusion about performance across subjects and maps.

$$\log(Y_{ijk}) = u + Subject_i + Map_j + block_k + \varepsilon_{ijk}$$

$$\text{Where i} = 1,2\ldots,32; \text{j} = 1,2,\ldots,30; \text{k} = 1,2,3,4,5,6;\ \varepsilon_{ijk} \sim N(0,\sigma^2)$$

```
              Df Sum Sq Mean Sq F value  Pr(>F)
subject_id    29  4.448 0.15338  13.131 < 2e-16 ***
map_id        29  0.624 0.02152   1.842 0.00465 **
block_1_to_6   5  0.061 0.01222   1.046 0.38914
Residuals    836  9.765 0.01168
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the above test result, we could know that the block effect are not significant, while the map and subject effect are significant. That means there are no difference in subjects' performance with time passing, but difference among maps will affect the subjects' performance, and different subjects have different performance.

### 3.2.2 Classification

We want to detect whether the maps are different in difficulty. Therefore, we used KNN algorithm to classify the maps with subjects' performance. We randomly select 7/10 subjects' data as train data, and 3/10 subjects' data as test data. The test data's prediction result is inaccuracy. It seems that only use one feature to classify maps is inappropriate. Then, we used K-means algorithm to classify performance into 30 types, and see weather the cluster id match the maps id. From the Figure 8, we could see they are not match basically. Therefore, we could not get the conclusion about the difficulty of maps .

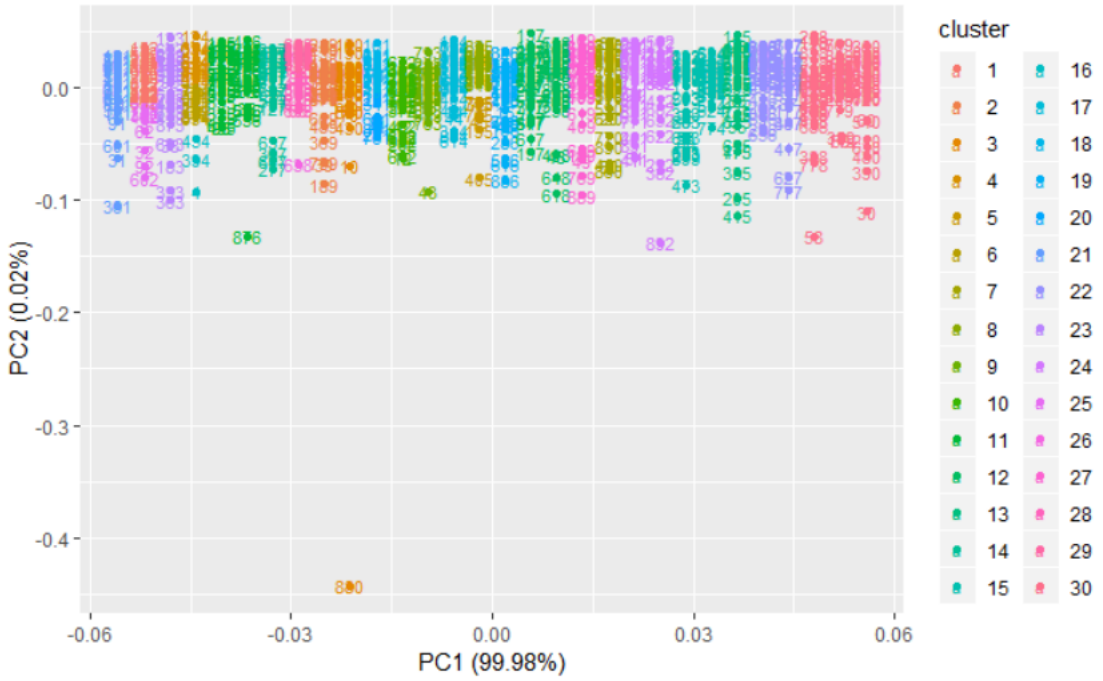



Figure 8. Classify Performance by k-means

## 4. Conclusion

From above analysis, we could conclude as follows:

1) With time passing, we could not see apparently performance of subjects change. That means no apparently learn effect with time.

2) The subjects' performance are affected by different type of maps, but we could not judge the difficulty of maps only by the subjects' performance in this dataset.

3) There exist huge variance on subjects' performance. Different subjects have significantly different performance on the same map.

# Appendix A:

```
library(FNN)
library(XML)
library(tidyverse)
library(ggfortify)
library(ggplot2)

rm(list = ls())

get_subject_order_map <- function() {
  subject_order_map <- read.table("order_arrange")
  subject_order_map + 1
}

get_map_score <- function() {
  if (str_detect(Sys.info()['sysname'], "Darwin")) {
    path_data <- "resource/res_analyz_spr_10/"
  } else {
    path_data <- "E:\\baiduyun\\github\\tsp_experiment\\tsp_experiment\\resource\
\res_analyz_spr_10"
  }
  best_value <- as.numeric(readLines(file.path(path_data,"ColorOptimalLength.txt"), 30))
  best_value
}

# 32 subjects
get_subject_dir_map <- function() {
  if (str_detect(Sys.info()['sysname'], "Darwin")) {
    path_data <- "resource/res_analyz_spr_10/TSP_spr_19/colored"
  } else {
```

```
    path_data <- "E:\\baiduyun\\github\\tsp_experiment\\tsp_experiment\\resource\
\res_analyz_spr_10\\TSP_spr_19/colored"
  }

  dirsAll <- list.dirs(path =file.path(path_data) , full.names = TRUE, recursive = TRUE)
  dirsAll <- grep("results", dirsAll, value = TRUE)
  subj_count <- length(dirsAll) #
  dirsAll
}

get_distance_from_two_record <- function(point1, point2) {
  distance <- sqrt((point1-point2)[1]^2 + (point1-point2)[2]^2)
  if(point1[3] != point2[3]) {
     distance <- 2 * distance
  }
  distance
}

get_file_content <- function(map_id, subject_id, subject_dir_map) {
  if (str_detect(Sys.info()['sysname'], "Darwin")) {
    file <- paste(subject_dir_map[subject_id],"/TSPMap_RandomPoints_Color_SwitchCost_",
toString(map_id-1),"-CollectedTour.tspx",sep = "")
  } else {
    file <- paste(subject_dir_map[subject_id],"\\TSPMap_RandomPoints_Color_SwitchCost_",
toString(map_id-1),"-CollectedTour.tspx",sep = "")
  }

  file_content = readChar(file, file.info(file)$size)
  file_content
}

get_real_record_distance <- function(map_id, subject_id, subject_dir_map) {
```

```
  file_content = get_file_content(map_id, subject_id, subject_dir_map)

  if (0 == length(file_content)) {
    warning("There are empty file, should check!")
  }

  tmp_data <- NULL
  first_data <- NULL
  distance_all <- 0
  count <- 0
  for(city in xmlChildren(xmlChildren(xmlChildren(xmlRoot(xmlParse(file_content)))$Tours)$Tour))
{
    count <- count + 1

    data<-c(as.numeric(xmlAttrs(city)['x']), as.numeric(xmlAttrs(city)['y']),
as.numeric(xmlAttrs(city)['group']))

    if(is.null(first_data)) { first_data <- data; }

    if (count != 1) {
      distance_all <- distance_all + get_distance_from_two_record(tmp_data, data)
      if (count == 50) {
        distance_all <- distance_all + get_distance_from_two_record(first_data, data)
      }
    }
    tmp_data <- data
  }

  distance_all
}

map_score <- get_map_score()
```

```
subject_dir_map <- get_subject_dir_map()
subject_order_map <- get_subject_order_map()

experiment_table <- data.frame()

for(subject_id in 1:30) {
  for(map_id in 1:30) {
    real_score <- get_real_record_distance(map_id, subject_id, subject_dir_map)
    best_score <- map_score[map_id]
    err_ratio <- (real_score-best_score)/best_score
    order_1_to_30 <- subject_order_map[subject_id, map_id]
    block_1_to_6 <- floor((order_1_to_30 - 1) / 5) + 1
    record <- data.frame(map_id, subject_id, real_score, best_score, err_ratio, order_1_to_30,
block_1_to_6)
    experiment_table <- rbind(experiment_table, record)
  }
}
experiment_table
# That is what we need for next step!!
#experiment_table

# write.csv(experiment_table, file='resource/experiment_table.csv', row.names = FALSE)
```

```{r plot_the_trace}
get_experiment_coord <- function(map_id, subject_id, subject_dir_map) {
  file_content = get_file_content(map_id, subject_id, subject_dir_map)
  if (0 == length(file_content)) {
    warning("There are empty file, should check!")
  }
  coord <- data.frame()
```

```
  count <- 0
  for(city in xmlChildren(xmlChildren(xmlChildren(xmlRoot(xmlParse(file_content)))$Tours)$Tour))
{
    count <- count + 1
    data<-c(map_id, subject_id, count,as.numeric(xmlAttrs(city)['x']), as.numeric(xmlAttrs(city)
['y']), as.numeric(xmlAttrs(city)['group']))
    coord <- rbind(coord, data)
  }
  colnames(coord) <- c("map_id","subject_id","index","x", "y", "group")
  coord
}



coord_1_1 <- get_experiment_coord(map_id=1, subject_id=1, subject_dir_map)
coord_1_2 <- get_experiment_coord(map_id=1, subject_id=2, subject_dir_map)
coord_1_3 <- get_experiment_coord(map_id=1, subject_id=3, subject_dir_map)
coord_2_1 <- get_experiment_coord(map_id=2, subject_id=1, subject_dir_map)
coord_2_2 <- get_experiment_coord(map_id=2, subject_id=2, subject_dir_map)
coord_2_3 <- get_experiment_coord(map_id=2, subject_id=3, subject_dir_map)
coord_3_1 <- get_experiment_coord(map_id=3, subject_id=1, subject_dir_map)
coord_3_2 <- get_experiment_coord(map_id=3, subject_id=2, subject_dir_map)
coord_3_3 <- get_experiment_coord(map_id=3, subject_id=3, subject_dir_map)


coord<-
rbind(coord_1_1,coord_1_2,coord_1_3,coord_2_1,coord_2_2,coord_2_3,coord_3_1,coord_3_2,coo
rd_3_3)
coord$group <- as.factor(coord$group)
coord$map_id <- as.character(coord$map_id)
coord$subject_id <- as.character(coord$subject_id)
```


```{r }
```

```
hp <- ggplot(coord, aes(x, y)) +
  geom_point(aes(color = group), size = 2) + geom_path()


map_labels <- c("1"="Map 1#", "2"="Map 2#", "3"="Map 3#")
subject_labels <- c("1"="Subject 1#", "2"="Subject 2#", "3"="Subject 3#")


# Histogram of total_bill, divided by sex and smoker
hp + facet_grid(subject_id ~ map_id, labeller=labeller(map_id = map_labels, subject_id =
subject_labels))


library(knitr)
sub_table <- experiment_table[experiment_table$map_id %in% 1:3 &
experiment_table$subject_id %in% 1:3,]
xtable_sub <- xtabs( real_score ~ subject_id +map_id , sub_table, drop.unused.levels=TRUE)
colnames(xtable_sub)<-c("Map #1","Map #2","Map #3")
rownames(xtable_sub)<-c("Subject #1","Subject #2","Subject #3")


kable(xtable_sub, caption = "Scores for Subject 1:3 on Map 1:3")


xtable_best<-data.frame(map_score[1],map_score[2],map_score[3])
colnames(xtable_best)<-c("Map #1","Map #2","Map #3")
rownames(xtable_best)<-c("Best Score in Theory")


kable(xtable_best, caption = "Best Scores for Map 1:3")


error_ratio <- xtable_sub
xtable_best1<-rbind(xtable_best,xtable_best,xtable_best)
error_ratio2<-(error_ratio-xtable_best1)/xtable_best1
rownames(error_ratio2)<-c("Subject #1","Subject #2","Subject #3")
kable(error_ratio2, caption = "Error Ratios for Subject 1 to 3 on Map 1 to 3”)
```

# Appendix B：

```{r, 2_model_regression}
experiment_table_lm <- experiment_table
experiment_table$map_id <- as.factor(experiment_table$map_id)
experiment_table$block_1_to_6 <- as.factor(experiment_table$block_1_to_6)
experiment_table$subject_id <- as.factor(experiment_table$subject_id)

anomod <- aov(log(err_ratio) ~ subject_id + map_id + block_1_to_6, data=experiment_table)
summary(anomod)

fit.lm <- lm(log(err_ratio) ~ subject_id + map_id + block_1_to_6, data=experiment_table)
summary(fit.lm)
qqnorm(fit.lm$residuals)

#prediction

```

## Classification prediction

```{r}
## boxplot
par(mar=c(4,4,2,4),mfrow=c(1,6))
boxplot(experiment_table[experiment_table$block_1_to_6==1,]$err_ratio,col="seagreen",ylim=c(0,0.8),xlab="block1")
lines(x=c(0.8,1.2),y=c(median(experiment_table[experiment_table$block_1_to_6==1,]$err_ratio),median(experiment_table[experiment_table$block_1_to_6==1,]$err_ratio)),col='red',lwd=2)
boxplot(experiment_table[experiment_table$block_1_to_6==2,]$err_ratio,col="seagreen",ylim=c(0,0.8),xlab="block2")
```

```
lines(x=c(0.8,1.2),y=c(median(experiment_table[experiment_table$block_1_to_6==2,]$err_ratio),
median(experiment_table[experiment_table$block_1_to_6==2,]$err_ratio)),col='red',lwd=2)

boxplot(experiment_table[experiment_table$block_1_to_6==3,]$err_ratio,col="seagreen",ylim=c(
0,0.8),xlab="block3")

lines(x=c(0.8,1.2),y=c(median(experiment_table[experiment_table$block_1_to_6==3,]$err_ratio),
median(experiment_table[experiment_table$block_1_to_6==3,]$err_ratio)),col='red',lwd=2)

boxplot(experiment_table[experiment_table$block_1_to_6==4,]$err_ratio,col="seagreen",ylim=c(
0,0.8),xlab="block4")

lines(x=c(0.8,1.2),y=c(median(experiment_table[experiment_table$block_1_to_6==4,]$err_ratio),
median(experiment_table[experiment_table$block_1_to_6==4,]$err_ratio)),col='red',lwd=2)

boxplot(experiment_table[experiment_table$block_1_to_6==5,]$err_ratio,col="seagreen",ylim=c(
0,0.8),xlab="block5")

lines(x=c(0.8,1.2),y=c(median(experiment_table[experiment_table$block_1_to_6==5,]$err_ratio),
median(experiment_table[experiment_table$block_1_to_6==5,]$err_ratio)),col='red',lwd=2)

boxplot(experiment_table[experiment_table$block_1_to_6==6,]$err_ratio,col="seagreen",ylim=c(
0,0.8),xlab="block6")

lines(x=c(0.8,1.2),y=c(median(experiment_table[experiment_table$block_1_to_6==6,]$err_ratio),
median(experiment_table[experiment_table$block_1_to_6==6,]$err_ratio)),col='red',lwd=2)




res <-
as.data.frame(cbind(experiment_table$map_id,experiment_table$subject_id,experiment_table$
err_ratio))

names(res) <- c("map_id","subject_id","err_ratio")

ggplot(res,aes(x=map_id, y = log(err_ratio),colour=subject_id)) + geom_point()

ggplot(res,aes(x=subject_id, y = log(err_ratio),colour=map_id)) + geom_point()


#train


train_data <- res[which(res$subject_id<22),]

test_data <- res[which(res$subject_id >21),]

train_data$index <- c(1:630)


get_distance <- function(x,y){
```

```
  distance <- abs(x-y)
  distance
}
get_knn <- function(k,training_data,test_point){
  my_dist <- lapply(training_data, get_distance, test_point)
  my_dist_sort <- sort(unlist(my_dist),index.return = TRUE)
  my_dist_sort$ix[1:k]
}



get_lable <- function(k,training_data,test_point){
  index <- get_knn(k,training_data,test_point)
  train_data[index,]$map_id
}

get_lable(10,train_data$err_ratio,0.25245273)[1]
# lapply(test_data$err_ratio)
#plot kmeans
autoplot(kmeans(res[,c(1,3)], 30),data=res,label=TRUE, label.size=3, frame=FALSE)
```