



인터랙티브 심리학

팀프로젝트

2019204011 김혜원

2019204088 정호진

2021204018 오민성

2021204051 우지윤

2021204048 이시은

목차

1. SRT / CRT 란?

1- 1. 데이터 수집

1- 2. 데이터 분석

1- 2- 1. 과정

1- 2- 2. 결과

2. Sternberg's classical test 란?

2- 1. 데이터 수집

2- 2. 데이터 분석

2- 2- 1. 과정

2- 2- 2. 결과

1. SRT / CRT 란?

SRT(Simple Reaction Time)는 자극에 대하여 Mental Identification 없이 즉각적으로 반응하는 과제를 수행하는데 걸리는 시간이며, **CRT**(Choice Reaction Time)는 어떤 자극에 대하여 Mental Identification 뿐 아니라 Mental Selection 과정을 필요로 하는 과제를 수행하는데 걸리는 시간이다.

Hick's Law에 따르면 $\text{Reaction Time} = a + bH$ 라는 식이 성립한다. 이때 H 는 전송된 정보의 양이며, a 는 SRT, b 는 정보처리 속도이다. a 는 inter-subject difference, S-R Modality, Stimulus intensity, Expectancy 등에 영향을 받으며, b 는 S-R Compatibility, Learning, Age등에 영향을 받는다.

1 - 1. 데이터 수집

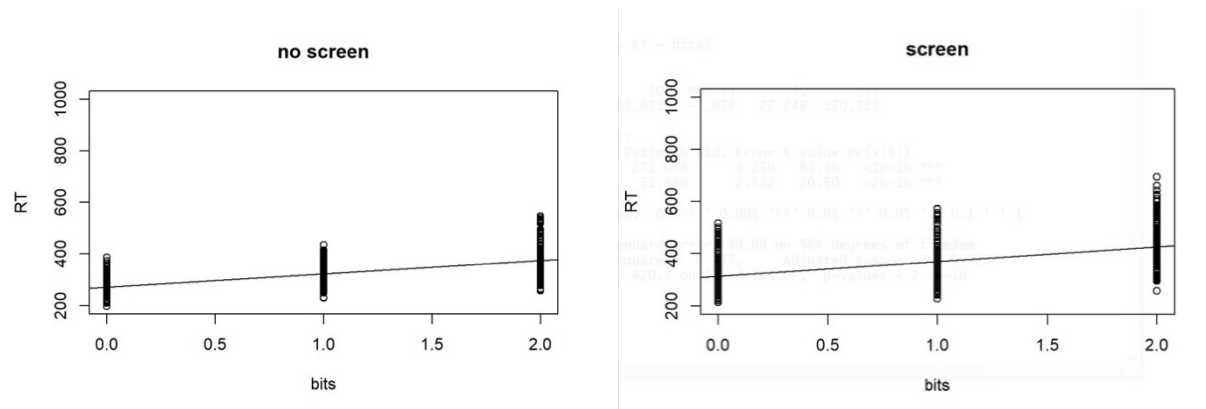
해당 데이터는 20대 20명의 데이터를 가지고 진행하였다. 해당 실험의 경우 시각적 방해요소에 따른 SRT변화와, 정보처리 속도의 변화를 확인해보기 위해 다음과 같은 데이터를 수집하였다. 실험자와 피실험자가 1:1로 마주앉아 실험을 진행하였고, 동일한 노트북, 동일한 프로그램으로 진행하여 최대한 noise가 없는 순수한 데이터를 얻기 위해 노력하였다. 또한 모든 피실험자에게 부담감으로 인한 부정적인 영향을 배제하기 위해 오답이 나오더라도 추후 예외 처리 가능함을 언급하였다.

1 - 2. 데이터 분석

1 - 2 - 1. 과정

데이터를 분석하기 위해 RStudio를 사용하였다. 실험 프로그램의 결과로 나온 텍스트 파일을 Dataset으로 삽입하여 데이터를 가공하였다. SRT/CRT 와 Sternberg's classical test의 데이터 전처리 과정이 매우 유사해 아래의 Sternberg's classical test 분석 과정에 설명했다.

1 - 2 - 2. 결과



```

      Estimate Std. Error t value Pr(>|t|)
(Intercept)  271.678     3.259   83.36 <2e-16 ***
bits         51.899     2.532   20.50 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 49.09 on 564 degrees of freedom
Multiple R-squared:  0.427,    Adjusted R-squared:  0.426
F-statistic: 420.3 on 1 and 564 DF, p-value: < 2.2e-16

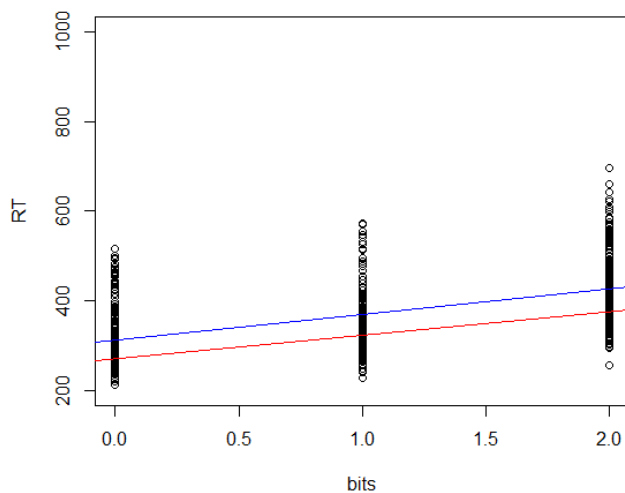
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  313.906     5.194   60.44 <2e-16 ***
bits         56.637     4.002   14.15 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 77.81 on 562 degrees of freedom
Multiple R-squared:  0.2627,    Adjusted R-squared:  0.2614
F-statistic: 200.3 on 1 and 562 DF, p-value: < 2.2e-16

```

No screen 의 경우 추정된 회귀식은 $271.678 + 51.899H(\text{Bit} = \text{제공된 정보량})$ 이며 회귀 계수의 유의성과 회귀 모형의 유의성 모두 0.05 미만으로 신뢰성이 높다. 이때 제공된 정보량이 SRT 변화에 주는 영향은 47.2% 로 추정할 수 있다.

Screen의 경우 추정된 회귀식은 $313.906 + 56.637H(\text{Bit})$ 이며 위의 식과 마찬가지로 회귀 계수의 유의성과 회귀 모형의 유의성 모두 0.05 미만으로 신뢰성이 높다. 정보량이 SRT 변화에 주는 영향은 26.27% 로 추정할 수 있다.



No screen 과 screen 그래프를 하나의 그래프로 합쳐서 나타내면 다음과 같다. (No screen – 붉은색, screen – 푸른색)

Hick's Law 에서 나타난 대로 정보처리 속도의 차이는 No screen은 51.899, screen은 56.637 로 시각적 방해요소가 있음에 따라 아주 조금 느려지는 것을 확인할 수 있다. 이는 실험 환경 설정에서 발생한 것으로 추정된다. 노트북 화면 중간점을 기준으로 영상과 실험 프로그램 창을 반으로 설정해주었고, 그로 인해서 다음과 같은 결과가 나왔을 수 있다. 다음과 같은 문제를 해결하기 위해서는, 실험 환경을 여러가지 시도를 해본 다음 실험을 진행하는 것이 좋은 해결책이 될 수 있다.

2. Sternberg's classical test 란?

무작위의 Positive Set을 제공하고, 임의의 숫자가 해당 Positive Set에 존재 하였는가에 대한 참, 거짓을 대답하는 test이다. 본 실험에서는 인간은 순차적으로 정보를 처리하기 때문에 Positive Set이 증가하면 Reaction Time(이하 RT)이 증가하며, Positive Set 안에 존재하지 않는 임의의 숫자가 제공되는 경우가 반대의 경우보다 더 큰 RT가 필요하다.

2 - 1. 데이터 수집

해당 데이터는 20대의 20명의 데이터를 가지고 진행하였다. 해당 실험의 경우 청각적 방해요소에 따른 RT 변화를 확인해보기 위해 다음과 같은 데이터를 수집하였다. 실험환경은 앞서 실험했던 환경과 동일하다.

2 - 2. 데이터 분석

2 - 2 - 1. 과정

```
path_dir = 'sternbergSound'

df_num3 = pd.DataFrame()
df_num4 = pd.DataFrame()
df_num5 = pd.DataFrame()
df_num6 = pd.DataFrame()

for file in file_list:
    if file[-5] == '3':
        sub_df = pd.read_csv(f'{path_dir}/{file}', sep = ',')
        df_num3 = pd.concat([df_num3, sub_df])
    if file[-5] == '4':
        sub_df = pd.read_csv(f'{path_dir}/{file}', sep = ',')
        df_num4 = pd.concat([df_num4, sub_df])
    if file[-5] == '5':
        sub_df = pd.read_csv(f'{path_dir}/{file}', sep = ',')
        df_num5 = pd.concat([df_num5, sub_df])
    if file[-5] == '6':
        sub_df = pd.read_csv(f'{path_dir}/{file}', sep = ',')
        df_num6 = pd.concat([df_num6, sub_df])
```

실험 프로그램의 결과로 나온 텍스트 파일을 Positive Set을 기준으로 각 DataFrame을 생성하였다. 각 데이터 프레임마다 아래의 코드를 통해 전처리를 진행하였다.

```
def rm_wrong(df):
    df.reset_index(inplace=True)
    df.drop(df[df['T/F'] == False].index, inplace=True)
    df.drop(['index', 'Index'], axis=1, inplace=True)
    return df
```

(오답 제거)

```

q3 = df_num3['Time(ms)'].quantile(0.75)
q1 = df_num3['Time(ms)'].quantile(0.25)

# 이상치 여부를 판별
def is_outlier(df):
    iqr = q3 - q1
    rt = df['Time(ms)']
    if rt > q3 + 1.5 * iqr or rt < q1 - 1.5 * iqr:
        return True
    else:
        return False

def rm_outlier(df):
    df['outlier'] = df.apply(is_outlier, axis = 1)
    # 이상치인 행은 제거하여 필터링
    df = df.loc[df['outlier'] == False].copy()
    # 이상치여부를 나타내는 열 제거
    del df['outlier']
    return df

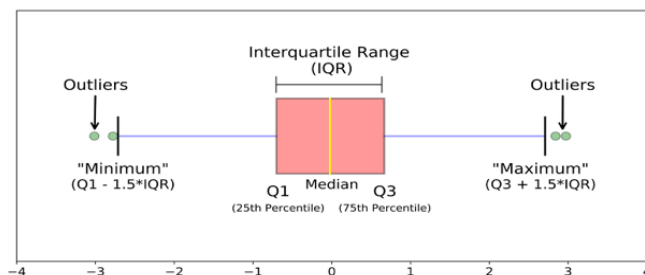
df_num3 = rm_outlier(df_num3)
print(df_num3)

```

(이상치 제거)

전처리에서는 오답 제거, 이상치 제거를 진행하였다. 오답을 제거한 이유는 다음과 같다. 올바르게 응답한 경우만을 이용한 이유는 해당 실험에서 확인하고자 하는 것은 Positive Set의 증가로 인해서 사람이 더 많은 정보의 처리에 따른 RT증가를 보고자 하는 것이며, 이때 실수나 올바르게 못한 정보처리 과정은 실험의 본질을 흐린다고 생각했기 때문이다.

이상치는 pandas의 quantile 함수를 통해 찾아서 제거하였다.



제한시간안에 응답을 하지 못한 Timeover가 존재하는지 확인을 하였으나, Timeove 데이터는 발생하지 않았다.

RStudio 를 이용하여 회귀 분석을 다음과 같이 진행하였다.

```

df_total <- read_csv("C:/users/minseong/Desktop/Sternberg-20201013T080132Z-001/df_total_Yessound.csv")
RT <- df_total$`Time(ms)`
pss <- df_total$pss
plot(pss, RT, main='Total Sound Graph', ylim=c(400,2000))
result <- lm(RT~pss)
summary(result)
abline(lm(RT~pss), lty=1, col='blue')

```

전처리한 csv파일을 불러와 독립변수를 RT, 종속변수를 pss(Positive Set)으로 설정하여 회귀하였다.

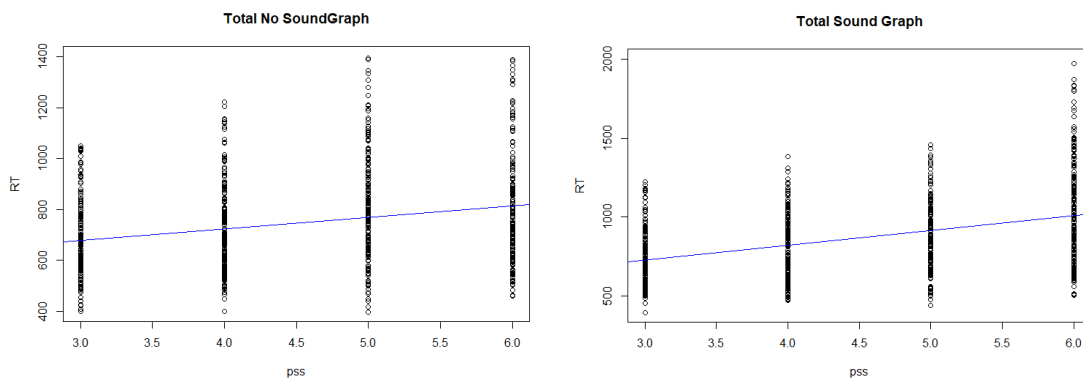
```
# yes로 대답한 데이터
df_yes <- read_csv("C:/Users/minseong/Desktop/Sternberg-20201013T080132Z-001/df_yes_total_Yessound.csv")
RT <- df_yes$`Time(ms)`
pss <- df_yes$pss
plot(pss,RT, ylim=c(400,2000))
result <- lm(RT~pss)
summary(result)
abline(lm(RT~pss),lty=1, col='blue')

# 그래프 겹치게 그리기
par(new=TRUE)

# no로 대답한 데이터
df_no <- read_csv("C:/Users/minseong/Desktop/Sternberg-20201013T080132Z-001/df_no_total_Yessound.csv")
RT <- df_no$`Time(ms)`
pss <- df_no$pss
plot(pss,RT,ylim=c(400,2000),main='Sound Graph')
result <- lm(RT~pss)
summary(result)
abline(lm(RT~pss),lty=1, col='red')
```

랜덤하게 나오는 임의의 숫자가 Positive Set 안에 존재하는 경우(이하 Yes)와 그렇지 않은 경우(이하 No)로 나누어서 회귀를 한 뒤, par(TRUE)를 통해 한 그래프에 시각화하였다.

2 - 2 - 2. 결과



```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    542.83      30.26   17.940 < 2e-16 ***
pss             45.21       6.50    6.955 7.9e-12 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

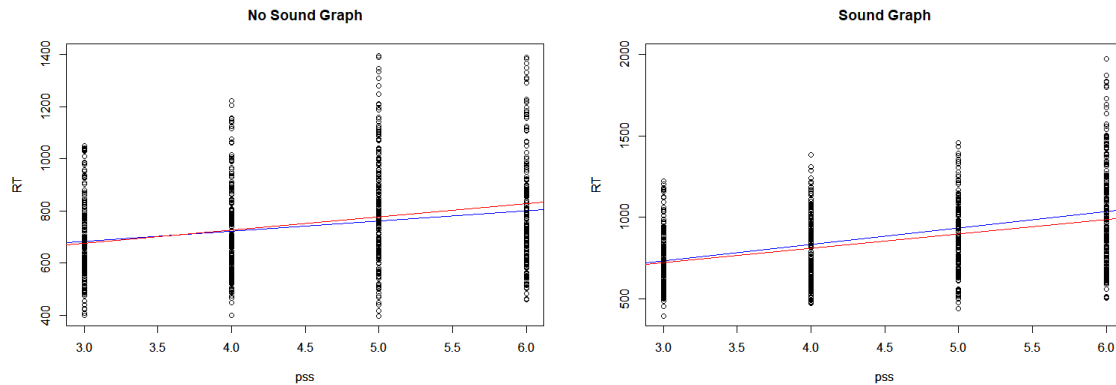
```
Residual standard error: 193.8 on 723 degrees of freedom
Multiple R-squared:  0.06271,    Adjusted R-squared:  0.06141
F-statistic: 48.37 on 1 and 723 DF, p-value: 7.904e-12
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    447.263     35.117   12.74 <2e-16 ***
pss             93.693     7.575   12.37 <2e-16 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 232.9 on 747 degrees of freedom
Multiple R-squared:  0.17,    Adjusted R-squared:  0.1689
F-statistic: 153 on 1 and 747 DF, p-value: < 2.2e-16
```

No Sound 의 경우 추정된 회귀식은 $542.83 + 45.21H(\text{Bit})$ 이며 위의 식과 마찬가지로 회귀 계수의 유의성과 회귀 모형의 유의성 모두 0.05 미만으로 신뢰성이 높다. Positive set이 이 RT 변화에 주는 영향은 6.271%로 추정할 수 있다.

Sound 의 경우 추정된 회귀식은 $447.263 + 93.693H$ 이며 회귀 계수의 유의성과 회귀 모형의 유의성 모두 0.05 미만으로 신뢰성이 높다. 이때 제공된 Positive set이 이 RT 변화에 주는 영향은 17%로 추정할 수 있다.



위의 두 그래프를 볼 수 있다. 각 그래프의 붉은색은 No, 푸른색은 Yes를 의미한다. 각 그래프에서 확인할 수 있듯이 Yes의 경우와 No의 경우 RT의 차이가 명확하지 않다. Yes의 경우와 No의 경우 RT의 차이가 명확하지 않은 이유는 다음과 같다.

Positive Set 은 숫자로 주어지기 때문에 순차적으로 정렬할 수 있다. 이렇게 순차적으로 정렬하게 되면 No의 경우 본래의 이론에서 가정한대로 모든 Positive Set의 모든 정보를 확인하지 않더라도 Positive Set 에 존재하지 않는다는 것을 확인할 수 있다. 실제로 필자도 다음과 같은 방식을 사용하였고, 피실험자들에게 실험이 끝난 뒤 확인하였을 때 대부분 같은 전략을 사용하였다. 이러한 문제를 해결하기 위해서는 Positive Set을 도형과 같이 정렬이 불가능한 것으로 진행하거나, 피실험자들이 Positive Set을 외우는 방식을 제한해야 한다.