Statistics Study

Facraft Regression

Warren Luo
Kevin Ren

June 14, 2018

1. Abstract

This research discovers the relationship between speed (in FPS) of *Facraft*[1] and the players' score. We assumed that the score will decrease when the speed of game increases. We used support vector regression to analyzed the data we obtained from game records and it seemed that the speed of the game does affect the score.

2. Problem Statement

We conducted the experiment to test our hypothesis that there is a negative relationship between game speed and players' score.

In our daily gaming[2] experience, we found that an increment of the speed (or pace) of a game often leads to the increment of the reaction speed required to play the game, and thus leads to a increment of the difficulty of the game, reflected by a lower score gained by the players. However, it is said that human can perform a lot more better under a higher pressure; if so, the score of a game might not decrease when the speed increases; it might even increase. To understand more about the phenomenon, we conducted this experiment.

3. Background Research

On the Internet, we found several researches with similar topic; e.g., *An EEG*[3] *Study on Numerical Error Monitoring under Performance Pressure* from Schillinger et al. In their research, the impact of students' academic pressure on their test score was studied. Eighteen participants performed numerical Stroop task under different pressures, and the result was analyzed. They conclude that performance is affected under a high pressure.

In our research, the relationship between pressure and performance also matters because the speed of the game affects the pressure of the player and the score of the game is an indicator of the player's performance.

[1] A Simple but Cool Game Made to "Worship" the Great BRS Principal Guangfa Wang, AKA Bro Fa
[2] Playing Computer Games, Not Gambling
[3] Electroencephalograph

4. Study Procedure

4.1 Experiment Design

4.1.1 Sampling

We asked for the population[4] data from Ms.Xin, the Homeroom Teacher of Class 2. The population data is documented in an excel file; students from the same class are recorded in the same column. We cleaned the file by deleting all irrelevant information and removing labels.

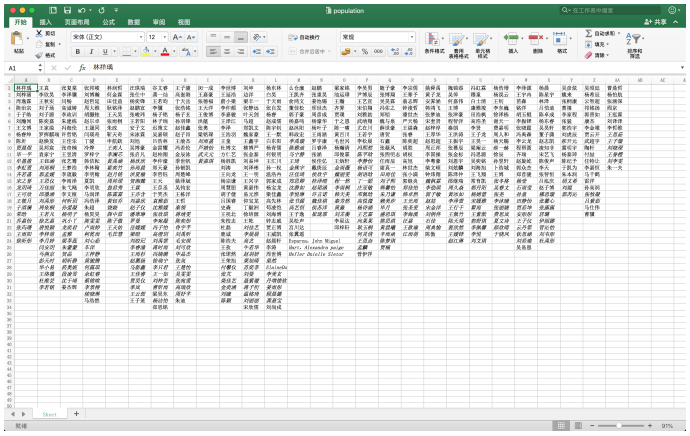


Figure 1: Cleaned Data

In order to take samples from the population, we used a python[5] script to process the data and perform simple random sampling method.

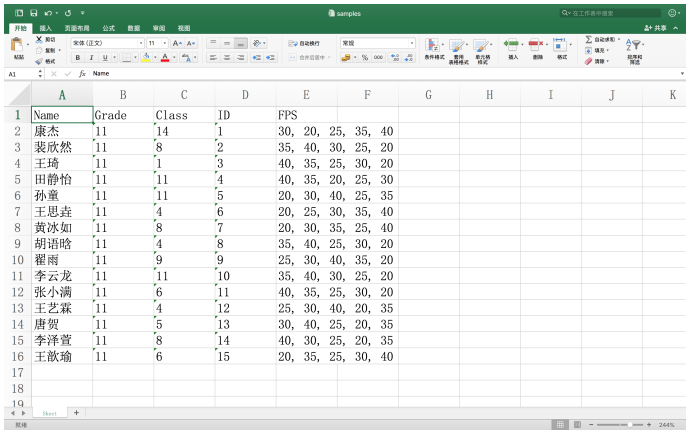
```
import openpyxl as xl
from random import shuffle

sample_size = int(input('Sample Size: '))

# load excel file
wb = xl.load_workbook('population.xlsx')
ws = wb.get_active_sheet()

# excel to list
population = []
for i in range(1,34):
    for j in range(1, 27):
        if type(ws.cell(row=i, column=j).value)==str:
            if j<13:
                grd = '10'
                cls = str(j)
            else:
```

[4]All Students in Grade 10 and 11

[5]A Programming Language That Lets Users Work Quickly and Integrate Systems More Effectively

```
                grd = '11'
                cls = str(j-12)
            population.append({'name': ws.cell(row=i, column=j).value.
                                                    strip(), 'grd': grd, '
                                                    cls': cls})

# shuffle list
shuffle(population)

# generate excel of samples and corresponding infos
wb = xl.Workbook()
ws = wb.active
ws['A1'].value = 'Name'
ws['B1'].value = 'Grade'
ws['C1'].value = 'Class'
ws['D1'].value = 'ID'
ws['E1'].value = 'FPS'
for i in range(sample_size):
    ws['A'+str(i+2)].value = population[i]['name']
    ws['B'+str(i+2)].value = population[i]['grd']
    ws['C'+str(i+2)].value = population[i]['cls']
    ws['D'+str(i+2)].value = str(i+1)
    fps = [str(_*5) for _ in range(4, 9)]
    shuffle(fps)
    ws['E'+str(i+2)].value = ', '.join(fps)
wb.save('samples.xlsx')
```

The script fetches the data from the excel file and covert the data into a list; the list is then shuffled and the samples are taken from the front of the list. A random sequence of FPS[6] of 20, 25, 30, 35 and 40 is generated and assigned to each student.



	A	B	C	D	E	F	G	H	I	J	K
1	Name	Grade	Class	ID	FPS						
2	康杰	11	14	1	30, 20, 25, 35, 40						
3	裴欣然	11	8	2	35, 40, 30, 25, 20						
4	王琦	11	1	3	40, 35, 25, 30, 20						
5	田静怡	11	11	4	40, 35, 20, 25, 30						
6	孙童	11	11	5	20, 30, 40, 25, 35						
7	王思垚	11	4	6	20, 25, 30, 35, 40						
8	黄冰如	11	8	7	20, 30, 35, 25, 40						
9	胡语晗	11	4	8	35, 40, 25, 30, 20						
10	翟雨	11	9	9	25, 30, 40, 35, 20						
11	李云龙	11	11	10	35, 40, 30, 25, 20						
12	张小满	11	6	11	40, 35, 25, 30, 20						
13	王艺霖	11	4	12	25, 30, 40, 20, 35						
14	唐贺	11	5	13	30, 40, 25, 20, 35						
15	李泽萱	11	8	14	40, 30, 25, 20, 35						
16	王歆瑜	11	6	15	20, 35, 25, 30, 40						
17											
18											
19											

Figure 2: Sample File Generated

[6] Frames Per Second

4.1.2 Gathering Data

Each student selected has a chance to to briefly try *Facraft* at a FPS of 30; then, they are asked to play the game for five times with the assigned FPS. The player, FPS, and score of each game is automatically recorded in a *SQLite*[7] database when the game is finished.

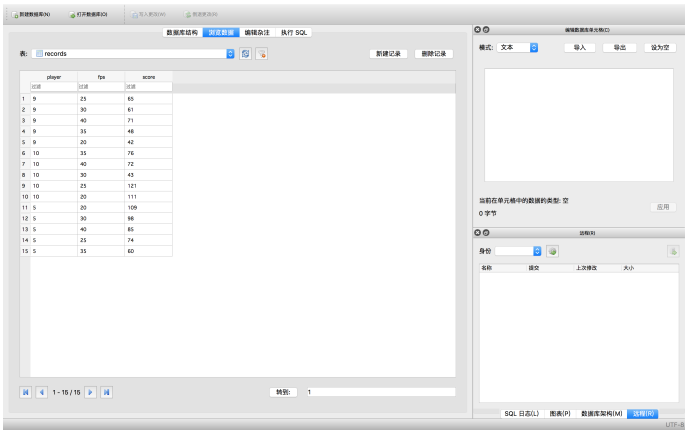


Figure 3: Data in SQLite Database Browser

4.1.3 Adjustments

After we found 11 students from the sample, we realized that it is too costly for us to gather enough data in this way because 8 out of the 11 students refused to play the game due to the fact that they had other works to do. We also found that it is difficult to get a perfect regression line when using data with only five different values of fps. Due to these problems, we have to slightly alter our plan.

We found several students to play the game for much more than five times; a random fps between 20 to 40 is generated each time. At last, we remove the records that are biased due to accidents; e.g., sometimes the player had to abort a game when it was time for class.

[7] A Self-contained, High-reliability, Embedded, Full-featured, Public-domain, SQL Database Engine

4.2 Data

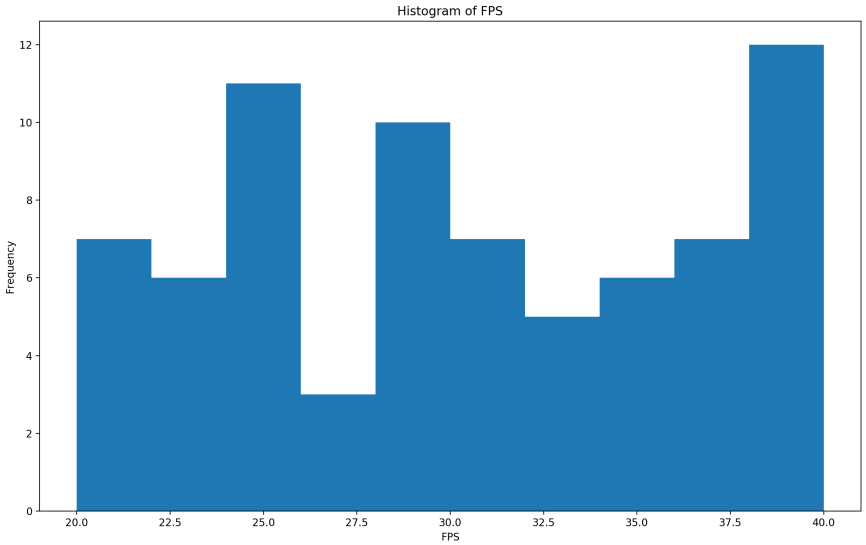
The final data we used for regression is shown bellow.

ID	FPS	SCORE		ID	FPS	SCORE		ID	FPS	SCORE
17	27	117		3	35	124		13	24	148
17	29	128		3	28	109		13	22	81
17	22	79		3	35	118		13	28	102
17	23	126		3	30	104		13	38	53
17	25	133		3	37	104		13	37	68
17	26	109		3	28	105		13	31	62
17	36	102		3	31	125		9	24	70
17	40	99		3	40	98		9	39	64
17	25	126		3	28	136		9	33	69
17	29	104		3	24	184		9	40	65
17	32	129		6	40	65		9	20	79
17	20	158		6	36	116		9	22	94
17	30	174		6	29	106		9	21	85
17	22	81		6	29	127		9	29	165
17	21	126		6	32	78		10	35	76
17	24	134		2	20	141		10	40	72
17	24	101		2	23	128		10	30	43
17	36	35		2	38	133		10	25	121
17	36	115		13	30	115		10	20	111
16	26	152		13	32	103		5	20	109
16	34	129		13	24	103		5	30	98
16	40	116		13	38	32		5	40	85
16	28	138		13	25	129		5	25	74
16	38	145		13	36	129		5	35	60
3	35	105		13	33	136				

The .db data file can be found on *GitHub*.

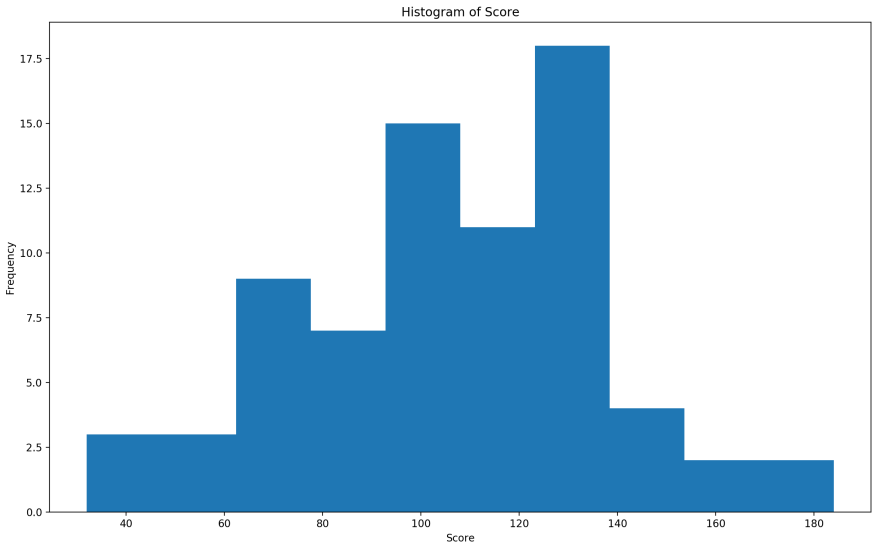
4.3 Statistics

4.3.1 FPS:



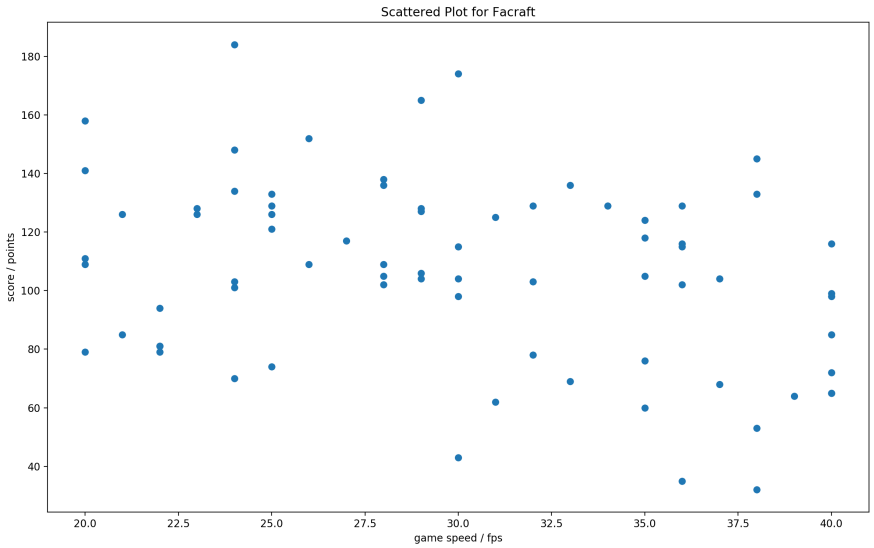
Mean FPS: 29.959
Median FPS: 29.5
Standard Deviation: 6.272
Description:
Obviously, the distribution is not normal; it appears to be uniform.

4.3.2 Score:



Mean Score: 106.257
Median Score: 109
Standard Deviation of Score: 31.481
Description:
The distribution is approximately normal.

4.3.3 Pattern Between the Two Variables



There is a week negative pattern between the game speed and the score. It is difficult to tell whether the relationship is linear or curved.

4.4 Regression

4.4.1 Method and Tools

We used python scripts to fetch data from database, make calculations, and draw plots. The essential part of the code is shown bellow.

```
import numpy as np
import sqlite3 as sql
from matplotlib import pyplot as plt
from sklearn.svm import SVR
from fetch_data import fetch_data as fd
from random_dark_colors import random_dark_colors as rdc

kernel = input('Which kernel to use, polynomial or linear ([p]/l)?    ')
if len(kernel)>0 and kernel[0]=='l':
    kernel = 'linear'
else:
    kernel = 'poly'

# get players
conn = sql.connect('records.db')
c = conn.cursor()
players = list({_[0] for _ in c.execute('select player from records').
                                        fetchall()})
c.close()
conn.close()

# prepare a list of dark colors
colors = rdc(len(players))

# draw scattered plot and regression line for each player
svr = SVR(kernel=kernel, degree=2)
for i in range(len(players)):
    fps, score = fd(player=players[i])
    p = svr.fit(fps, score).predict(np.array([[_] for _ in range(20, 41)]
                                        ))
    plt.scatter(fps, score, color=colors[i])
    plt.plot(np.array([[_] for _ in range(20, 41)]), p, color=colors[i],
                                        label='Player '+str(i+1))

# draw regression line for all players
fps, score = fd()
p = svr.fit(fps, score).predict(np.array([[_] for _ in range(20, 41)]))
plt.plot(np.array([[_] for _ in range(20, 41)]), p, color='black', label=
                                        'all players')

# draw labels
plt.xlabel('game speed / fps')
plt.ylabel('score / points')
if kernel=='linear':
    plt.title('Support Vector Linear Regression for Facraft')
else:
    plt.title('Support Vector Polynomial Regression for Facraft')
```

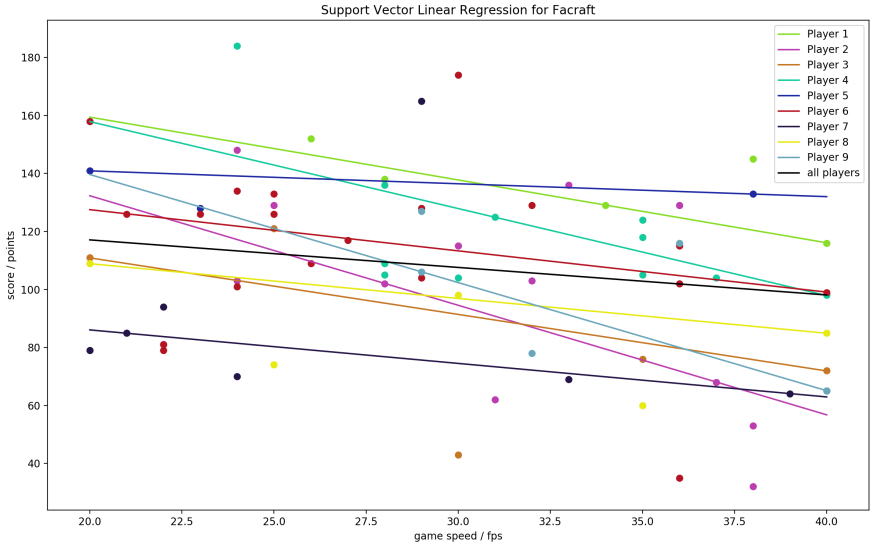
```
plt.legend()

plt.show()

# draw residual plot
p = svr.fit(fps, score).predict(fps)
res = score - p
plt.plot([15, 45], [0, 0], linestyle='--', color='black', lw=1)
plt.scatter(fps, res)
plt.xlim(15, 45)

# draw labels
if kernel=='linear':
    plt.title('Residual Plot for Linear Regression')
else:
    plt.title('Residual Plot for Polynomial Regression')
plt.xlabel('fps')
plt.ylabel('residual ( $score - \widehat{score}$ )')
plt.show()

# draw residual hiostogram
plt.hist(res)

# draw labels
if kernel=='linear':
    plt.title('Histogram of Residuals for Linear Regression')
else:
    plt.title('Histogram of Residuals for Polynomial Regression')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.show()
```

We used *svm.SVR* from *scikit-learn* for the regression. We also used other python packages (*sqlite3*, *numpy*, and *matplotlib*).

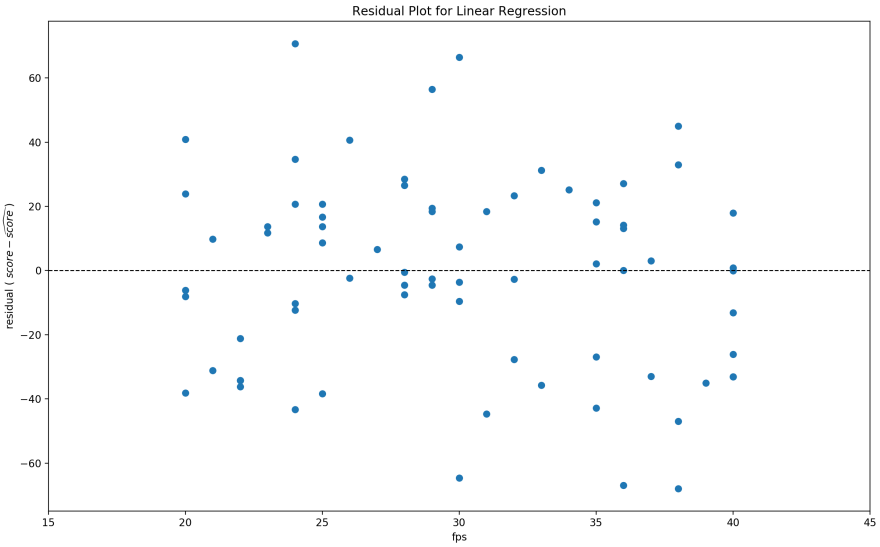
You can find the full code on *GitHub*.

4.4.2 Linear Regression

At first, we tried linear regression because the general pattern seems to be linear.

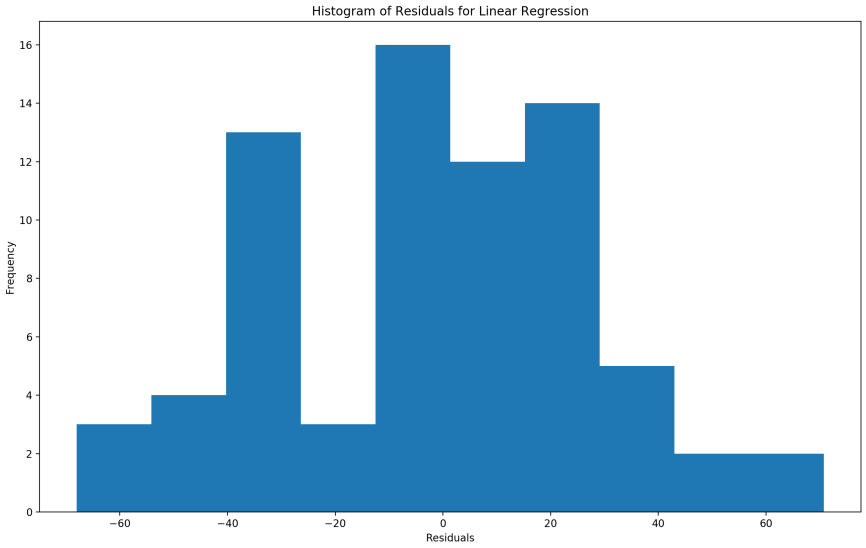


Overall, the slope of the regression line is slightly negative. The fact that the slope of regression line varies for each player reflexes different capabilities to cope with increasing speed.



Since the games are played by different players who have different gaming skills, there are significant difference between the scores generated. Thus, it is reasonable to see such

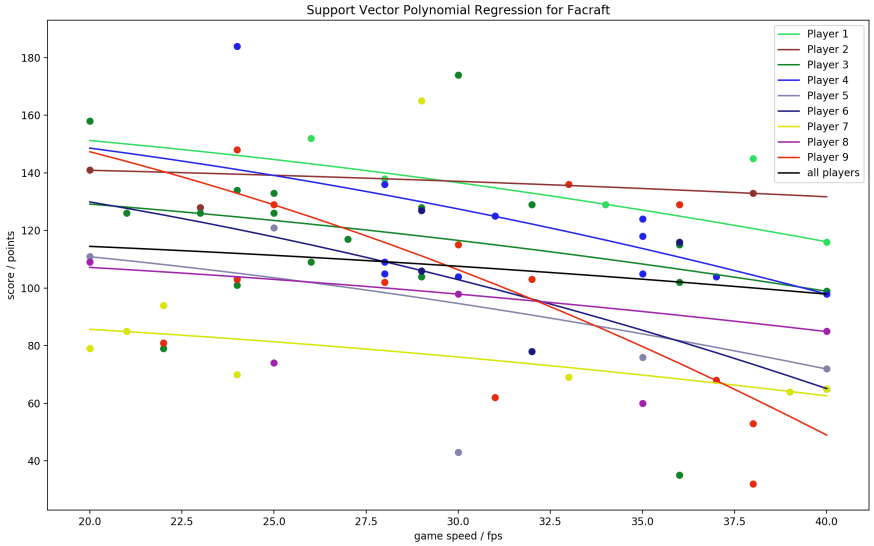
large residuals as long as no special pattern exists.



The distribution of residuals is approximately normal.

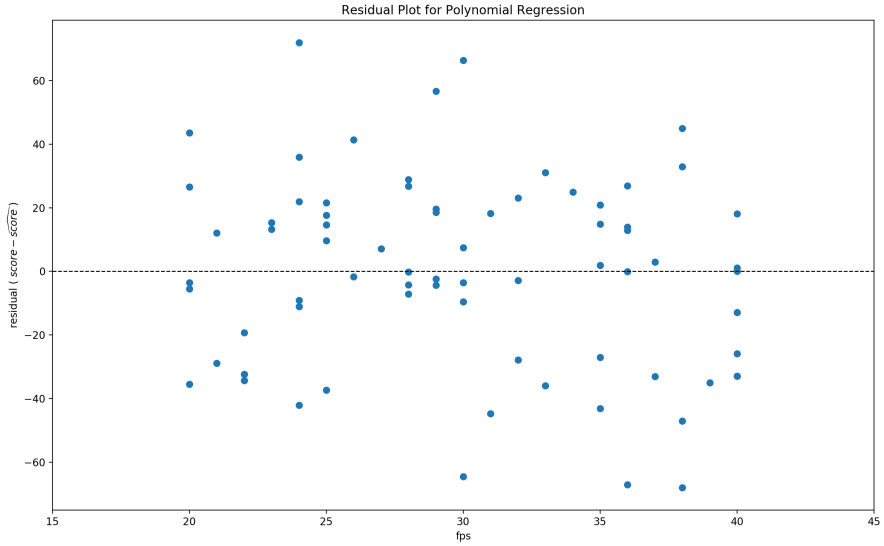
4.4.3 Polynomial Regression

In the linear regression, although the regression line reflexes the overall pattern, the data of players like player 9 cannot be predicted appropriately. Therefore, we used polynomial regression afterwards.

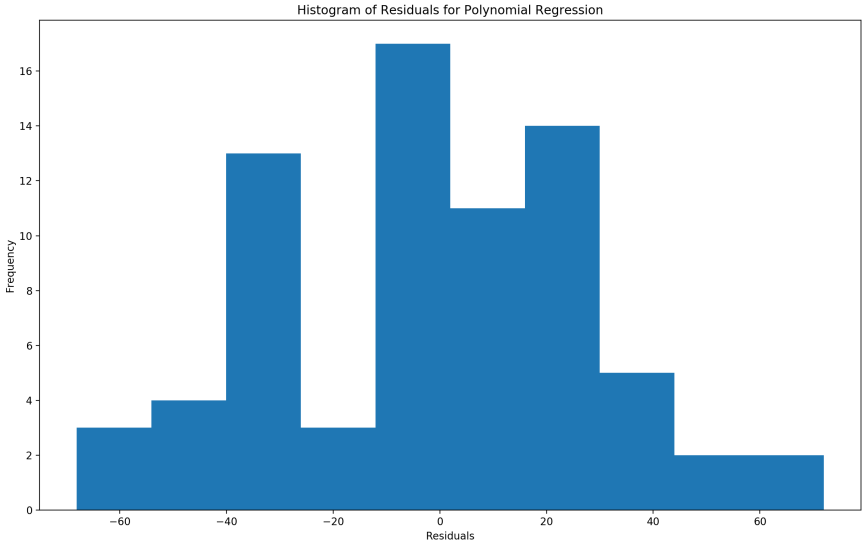


From the graph, we can see a week curvature downwards for all players. There can be two different interpretations of the curvature:

- The correlation between the speed and difficulty is not linear. The game becomes much more difficult as the speed increases further more.
- Hormones like adrenaline improve the quick reaction ability of the players. However, the improvement does not work when the speed becomes too high.



The residuals are almost equally larges as those from the linear regression. There are no special patterns.



The distribution of residuals is approximately normal.

5. Discussion & Conclusion

5.1 Our Weakness

In the study, we had to make adjustments (4.1.3) due to several limitations; these adjustments might affect the accuracy of the results. In addition, the players played the game under different circumstances; thus, environmental factors like noise and temperature might act as confounding variables.

5.2 Extrapolation

Extrapolating from the results, players generally gain a lower score when the game gets faster. We believe that this extrapolation can be generalized to all Chinese students who have physical and psychological development levels and gaming experiences equivalent to those of normal students in BRS[8] because physical and psychological development levels and gaming experiences are the main factors that affect gaming ability.

5.3 Further Work

There are a lot more to do other than what we did. In order to study the relationship between game speed and score, more different games can be studied and experiments can be conducted in better environments.

5.4 Conclusion

After analysis of the results of the regression, we concluded that when playing games, for most high-school students, an increment of game speed will generally lead to a lowers score.

[8] Our School — Beijing Royal School