SI 201
Final Presentation

Presented by :
I Want To Go Home: Chih-Hsiang Chang & Xiwen Mark

Today’s Agenda

01 Introduction

02 Key Questions

03 APIs and Websites Sources

04 Gathering Data and Database

05 Calculation and Visualization

06 Conclusion

07 Challenges

08 Resources

Introduction

This project is aim to gather data from one or more APIs and a website (with Beautiful Soup) to answer questions about the preferences and correlations between books and their movie adaptations.

Goals

Initial:

Identify the preferences and correlations between books and their movie adaptations using two APIs.

Final:

Added an additional website to scrape a list of books that have been made into movies

Key Questions

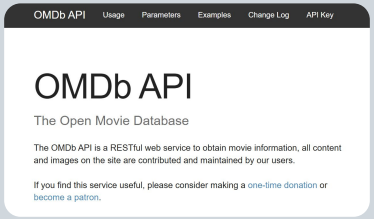
PROBLEM 1
Book vs Movie Preferences

Do people like the movie adaptions more?
Or do people love their original book compositions…?

PROBLEM 2
Book vs Movie Ratings Correlation

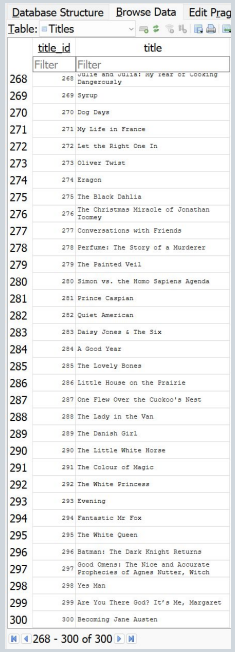
Do highly rated books lead to highly rated movies?

APIs & Web

Source 01	Source 02	Source 03
	Google Books APIs	goodreads
As seen in the homeworks… **–The almighty OMDb API!**	Google Books API	Goodreads Website

Gathering Data and Database

- Final Database in DB Browser



Database Structure | Browse Data | Edit Prag

Table: Titles

	title_id	title
	Filter	Filter
268	268	Julie and Julia: My Year of Cooking Dangerously
269	269	Syrup
270	270	Dog Days
271	271	My Life in France
272	272	Let the Right One In
273	273	Oliver Twist
274	274	Eragon
275	275	The Black Dahlia
276	276	The Christmas Miracle of Jonathan Toomey
277	277	Conversations with Friends
278	278	Perfume: The Story of a Murderer
279	279	The Painted Veil
280	280	Simon vs. the Homo Sapiens Agenda
281	281	Prince Caspian
282	282	Quiet American
283	283	Daisy Jones & The Six
284	284	A Good Year
285	285	The Lovely Bones
286	286	Little House on the Prairie
287	287	One Flew Over the Cuckoo's Nest
288	288	The Lady in the Van
289	289	The Danish Girl
290	290	The Little White Horse
291	291	The Colour of Magic
292	292	The White Princess
293	293	Evening
294	294	Fantastic Mr Fox
295	295	The White Queen
296	296	Batman: The Dark Knight Returns
297	297	Good Omens: The Nice and Accurate Prophecies of Agnes Nutter, Witch
298	298	Yes Man
299	299	Are You There God? It's Me, Margaret
300	300	Becoming Jane Austen

268 - 300 of 300

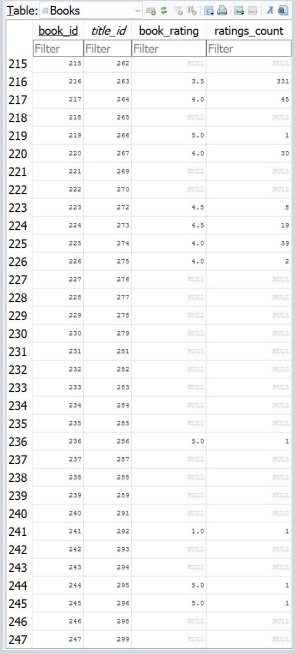
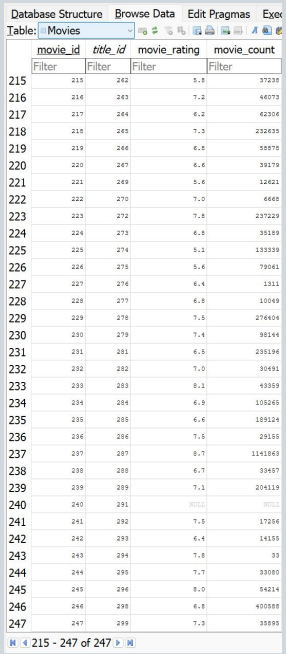


Table: Books

	book_id	title_id	book_rating	ratings_count
	Filter	Filter	Filter	Filter
215	215	262	NULL	NULL
216	216	263	3.5	331
217	217	264	4.0	45
218	218	265	NULL	NULL
219	219	266	5.0	1
220	220	267	4.0	30
221	221	269	NULL	NULL
222	222	270	NULL	NULL
223	223	272	4.5	8
224	224	273	4.5	19
225	225	274	4.0	39
226	226	275	4.0	2
227	227	276	NULL	NULL
228	228	277	NULL	NULL
229	229	278	NULL	NULL
230	230	279	NULL	NULL
231	231	281	NULL	NULL
232	232	282	NULL	NULL
233	233	283	NULL	NULL
234	234	284	NULL	NULL
235	235	285	NULL	NULL
236	236	286	5.0	1
237	237	287	NULL	NULL
238	238	288	NULL	NULL
239	239	289	NULL	NULL
240	240	291	NULL	NULL
241	241	292	1.0	1
242	242	293	NULL	NULL
243	243	294	NULL	NULL
244	244	295	5.0	1
245	245	296	5.0	1
246	246	298	NULL	NULL
247	247	299	NULL	NULL



Database Structure | Browse Data | Edit Pragmas | Exec

Table: Movies

	movie_id	title_id	movie_rating	movie_count
	Filter	Filter	Filter	Filter
215	215	262	5.8	37238
216	216	263	7.2	46073
217	217	264	6.2	62306
218	218	265	7.3	232635
219	219	266	6.8	58878
220	220	267	6.6	39179
221	221	269	5.6	12621
222	222	270	7.0	6668
223	223	272	7.8	237229
224	224	273	6.8	35189
225	225	274	5.1	133339
226	226	275	5.6	79061
227	227	276	6.4	1311
228	228	277	6.8	10049
229	229	278	7.5	276404
230	230	279	7.4	98144
231	231	281	6.5	235196
232	232	282	7.0	30491
233	233	283	8.1	43359
234	234	284	6.9	105265
235	235	285	6.6	189124
236	236	286	7.5	29155
237	237	287	8.7	1141863
238	238	288	6.7	93457
239	239	289	7.1	204119
240	240	291	NULL	NULL
241	241	292	7.5	17256
242	242	293	6.4	14155
243	243	294	7.8	33
244	244	295	7.7	33080
245	245	296	8.0	54214
246	246	298	6.8	400588
247	247	299	7.3	35895

215 - 247 of 247

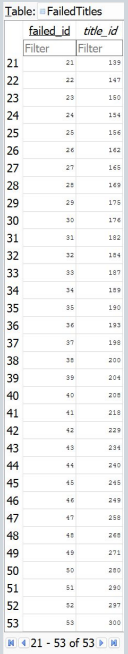


Table: FailedTitles

	failed_id	title_id
	Filter	Filter
21	21	139
22	22	147
23	23	150
24	24	154
25	25	156
26	26	162
27	27	165
28	28	169
29	29	175
30	30	176
31	31	182
32	32	184
33	33	187
34	34	189
35	35	190
36	36	193
37	37	198
38	38	200
39	39	204
40	40	208
41	41	218
42	42	229
43	43	234
44	44	240
45	45	245
46	46	249
47	47	258
48	48	268
49	49	271
50	50	280
51	51	290
52	52	297
53	53	300

21 - 53 of 53

Titles | Books | Movies | Failed Titles

Gathering Data and Database

- Preventing Duplicates

```
if gb_data is None:
    print("  Skipping: no valid Google Books data.")
    cur = conn.cursor()
    cur.execute(
        "INSERT OR IGNORE INTO FailedTitles (title_id) VALUES (?)",
        (title_id,)
    )

    conn.commit()
    continue
```

```
if omdb_data is None:
    print("  Skipping: no valid OMDb data.")
    cur = conn.cursor()
    cur.execute(
        "INSERT OR IGNORE INTO FailedTitles (title_id) VALUES (?)",
        (title_id,)
    )

    conn.commit()
    continue
```

Table: FailedTitles

	failed_id	*title_id*
	Filter	Filter
21	21	139
22	22	147
23	23	150
24	24	154
25	25	156
26	26	162
27	27	165
28	28	169
29	29	175
30	30	176
31	31	182
32	32	184
33	33	187
34	34	189
35	35	190
36	36	193
37	37	198
38	38	200
39	39	204
40	40	208
41	41	218
42	42	229
43	43	234
44	44	240
45	45	245
46	46	249
47	47	258
48	48	268
49	49	271
50	50	280
51	51	290
52	52	297
53	53	300

21 - 53 of 53

Gathering Data and Database

- Integer Key (title_id)


Database Structure | Browse Data | Edit Prag

Table: Titles

	title_id	title
	Filter	Filter
268	268	Julie and Julia: My Year of Cooking Dangerously
269	269	Syrup
270	270	Dog Days
271	271	My Life in France
272	272	Let the Right One In
273	273	Oliver Twist
274	274	Eragon
275	275	The Black Dahlia
276	276	The Christmas Miracle of Jonathan Toomey
277	277	Conversations with Friends
278	278	Perfume: The Story of a Murderer
279	279	The Painted Veil
280	280	Simon vs. the Homo Sapiens Agenda
281	281	Prince Caspian
282	282	Quiet American
283	283	Daisy Jones & The Six
284	284	A Good Year
285	285	The Lovely Bones
286	286	Little House on the Prairie
287	287	One Flew Over the Cuckoo's Nest
288	288	The Lady in the Van
289	289	The Danish Girl
290	290	The Little White Horse
291	291	The Colour of Magic
292	292	The White Princess
293	293	Evening
294	294	Fantastic Mr Fox
295	295	The White Queen
296	296	Batman: The Dark Knight Returns
297	297	Good Omens: The Nice and Accurate Prophecies of Agnes Nutter, Witch
298	298	Yes Man
299	299	Are You There God? It's Me, Margaret
300	300	Becoming Jane Austen

268 - 300 of 300

Titles

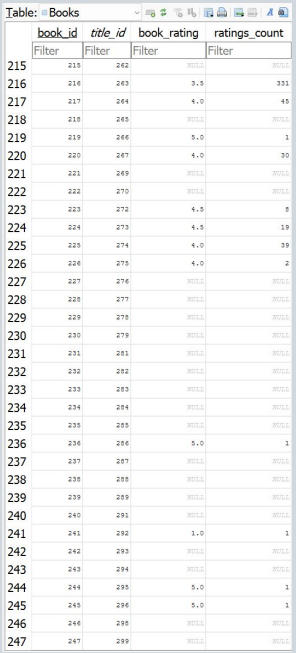


Table: Books

	book_id	title_id	book_rating	ratings_count
	Filter	Filter	Filter	Filter
215	215	262	NULL	NULL
216	216	263	3.5	331
217	217	264	4.0	45
218	218	265	NULL	NULL
219	219	266	5.0	1
220	220	267	4.0	30
221	221	269	NULL	NULL
222	222	270	NULL	NULL
223	223	272	4.5	8
224	224	273	4.5	19
225	225	274	4.0	39
226	226	275	4.0	2
227	227	276	NULL	NULL
228	228	277	NULL	NULL
229	229	278	NULL	NULL
230	230	279	NULL	NULL
231	231	281	NULL	NULL
232	232	282	NULL	NULL
233	233	283	NULL	NULL
234	234	284	NULL	NULL
235	235	285	NULL	NULL
236	236	286	5.0	1
237	237	287	NULL	NULL
238	238	288	NULL	NULL
239	239	289	NULL	NULL
240	240	291	NULL	NULL
241	241	292	1.0	1
242	242	293	NULL	NULL
243	243	294	NULL	NULL
244	244	295	5.0	1
245	245	296	5.0	1
246	246	298	NULL	NULL
247	247	299	NULL	NULL

Books

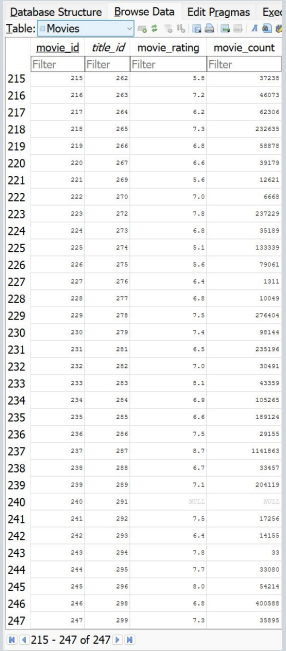

Database Structure | Browse Data | Edit Pragmas | Exec

Table: Movies

	movie_id	title_id	movie_rating	movie_count
	Filter	Filter	Filter	Filter
215	215	262	5.5	37235
216	216	263	7.2	46073
217	217	264	6.2	62306
218	218	265	7.3	232635
219	219	266	6.8	58878
220	220	267	6.6	39179
221	221	269	5.6	12621
222	222	270	7.0	6668
223	223	272	7.8	297229
224	224	273	6.8	35189
225	225	274	5.1	133339
226	226	275	5.6	79061
227	227	276	6.4	1311
228	228	277	6.8	10049
229	229	278	7.5	276404
230	230	279	7.4	98144
231	231	281	6.5	235196
232	232	282	7.0	30491
233	233	283	8.1	43359
234	234	284	6.9	105265
235	235	285	6.6	189124
236	236	286	7.5	29155
237	237	287	8.7	1161863
238	238	288	6.7	33457
239	239	289	7.1	206119
240	240	291	NULL	NULL
241	241	292	7.5	17256
242	242	293	6.4	14155
243	243	294	7.8	33
244	244	295	7.7	33080
245	245	296	8.0	54214
246	246	298	6.8	400588
247	247	299	7.3	35895

215 - 247 of 247

Movies

Gathering Data and Database

- Limits to 25 data stored

```python
def load_batch(conn, max_new=25):
    """
    Load up to max_new NEW adaptations in one run.
    This enforces the 25-items-per-run rule from the project spec.
    """
    candidate_rows = get_pending_titles(conn, max_new)

    inserted = 0
    for title_id, title in candidate_rows:
        if inserted >= max_new:
            break

        print(f"Processing '{title}' (title_id={title_id}) ...")
```

```python
max_pages = 5
max_new_per_run = 25

inserted = 0

for page_num in range(1, max_pages + 1):
    if inserted >= max_new_per_run or (current_count + inserted) >= max_count:
        break
```

API

Scrape

Select Data & JOIN

JOIN if Book and Movie has the same title_id

```python
def fetch_joined_data(conn):
    """
    Get joined rows: one per book-movie pair.
    """
    cur = conn.cursor()
    cur.execute("""
        SELECT
            T.title,
            B.book_rating,
            B.ratings_count,
            M.movie_rating,
            M.movie_count
        FROM Titles AS T
        JOIN Books  AS B ON T.title_id = B.title_id
        JOIN Movies AS M ON T.title_id = M.title_id
        WHERE B.book_rating IS NOT NULL
          AND M.movie_rating IS NOT NULL
    """)
    rows = cur.fetchall()

    data = []
    for row in rows:
        data.append({
            "title":        row[0],
            "book_rating":  row[1],
            "book_count":   row[2],
            "movie_rating": row[3],
            "movie_count":  row[4],
        })

    return data
```

Calculation

```python
def compute_preference_counts(data):
    books_better = 0
    movies_better = 0
    ties = 0
    total = 0

    for row in data:
        book_rating = row.get("book_rating")
        movie_rating_10 = row.get("movie_rating")
        movie_rating_5 = convert_movie_rating(movie_rating_10)

        if book_rating is None or movie_rating_5 is None:
            continue

        total += 1
        if movie_rating_5 > book_rating:
            movies_better += 1
        elif book_rating > movie_rating_5:
            books_better += 1
        else:
            ties += 1

    return books_better, movies_better, ties, total
```

```python
def preference_percentage(preferece_counts):
    books_better, movies_better, ties, total = preferece_counts

    if total == 0:
        return (0, 0, 0)

    books_better_pct = books_better / total
    movies_better_pct = movies_better / total
    ties_pct = ties / total
    return books_better_pct, movies_better_pct, ties_pct
```

Question 1:
Percentage of Book Preferred and Movie Preferred

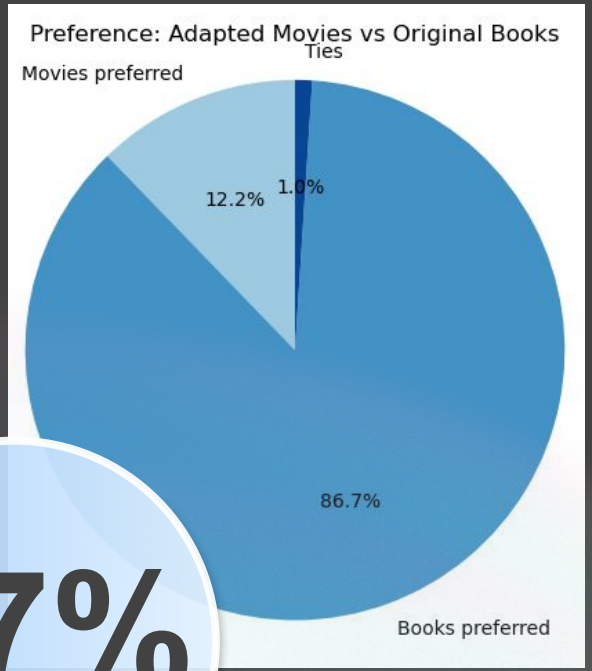
Visualization

Question 1:

Percentage of Book Preferred and Movie Preferred

- 86.7% → preference for books over movies
- 12.2% → preferring movies
- 1% → tie.

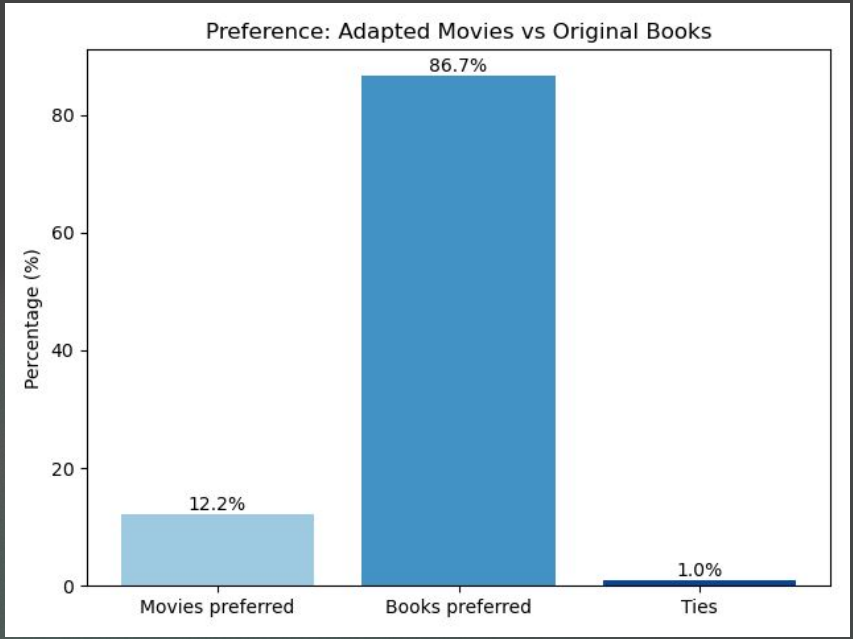
This suggests that the majority of people prefer the book over the movie in terms of book-to-movie adaptations.



87%

Preferred Books.

Visualization



Question 1:

Percentage of Book Preferred and Movie Preferred

Calculation

```python
def prepare_correlation_data(data):
    """
    Preparing data for further statistical computation.
    Convert data into x and y values [array]
    Input: filtered data (dict)
    Output: x (array), y (array)
    """
    x_values = []
    y_values = []

    for row in data:
        book_rating = row.get("book_rating")
        movie_rating_10 = row.get("movie_rating")
        movie_rating_5 = convert_movie_rating(movie_rating_10)

        # prevent null values
        if book_rating is None or movie_rating_5 is None:
            continue

        x_values.append(float(book_rating))
        y_values.append(float(movie_rating_5))

    return x_values, y_values
```

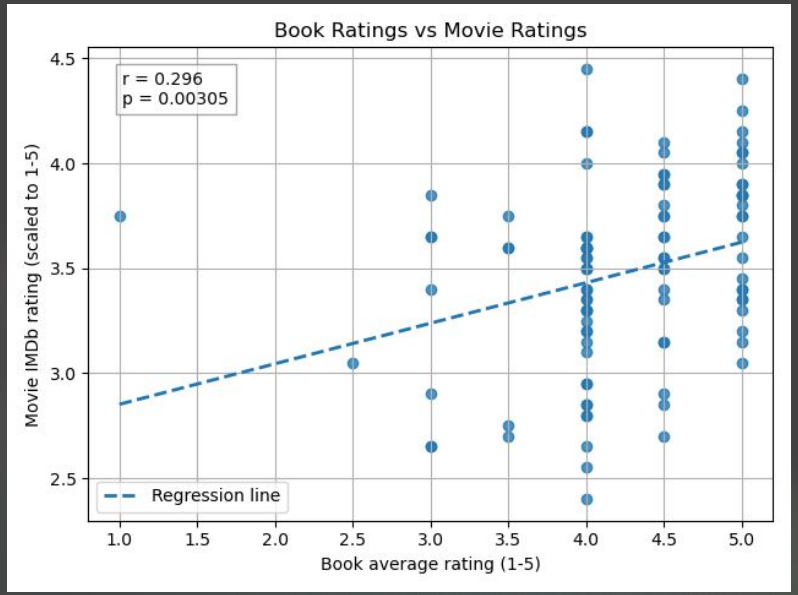
```python
def pearson_correlation(x, y):
    if len(x) != len(y) or len(x) < 3:
        return None

    r, p = pearsonr(np.array(x), np.array(y))
    return r, p

def linear_regression(x, y):
    if len(x) != len(y) or len(x) < 3:
        return None

    result = linregress(np.array(x), np.array(y))
    return {
        "slope": result.slope,
        "intercept": result.intercept,
        "r_value": result.rvalue,
        "p_value": result.pvalue,
        "stderr": result.stderr
    }
```

Question 2:
Books and Movies Rating correlations

Visualization

Question 2:
Books and Movies Rating correlations

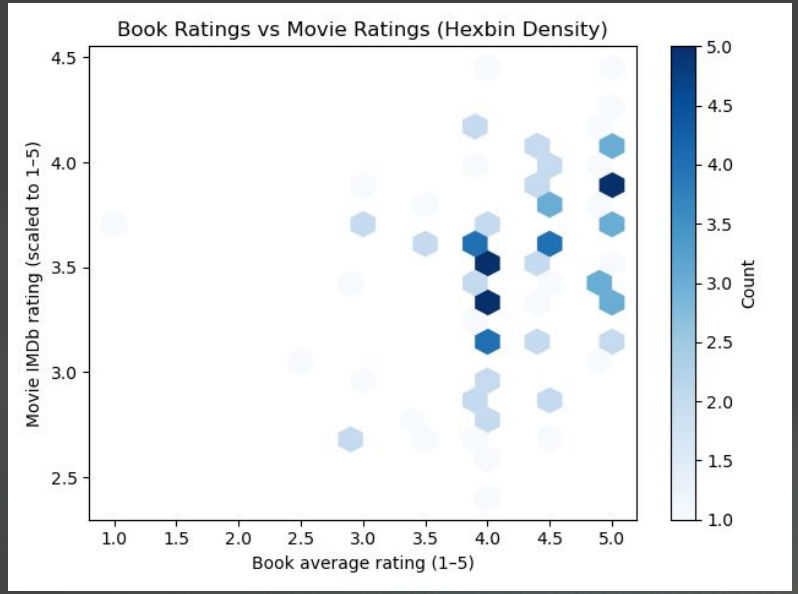
- The scatter plot shows a slightly upward trend
- Correlation coefficient of 0.296 and a p-value of 0.003 → correlation between the two variables is weak and statistically significant.



Visualization

Question 2:
Books and Movies Rating correlations

- Higher-rated books tend to have more movie adaptations.
- Higher book rating does not guarantee a higher movie adaptation rating.
- Adaptations rarely outperform low-rated books.
- Books typically receive slightly higher ratings than their movie adaptations.



Challenges

Chih-Hsiang:

- New API
- Structures
- Approach

Xiwen:

- Relational Database
- Calculation Threshold

Update

Fixed after presentation:

The title scraping wasn’t limited to 25, now it is fixed accordingly.

Thank You

Presented by :

Chih-Hsiang Chang & Xiwen Mark

(and I actually want to go home)