Project 1: My SQL & Knowledge Discovery

MAT 259 - Winter 2023
Brianna Griffin

Introduction & Description:

For the first project, I will be looking at irregularities in the data set through outliers. Specifically, I will be looking at checkout time in days for the first book in the Twilight series by Stephenie Meyer, namely "Twilight". Published in 2005, the book is based in Forks, Washington around 100 miles away from Seattle. Thus, it has a lot of prevalence within the Seattle Public Library with a lot of years of checkout and checkin data. Specifically, I would like to answer the following questions:

- Are there any observations that do not fit the rest of the data set?
- When and why do these observations occur?
- How do the outliers affect statistics of the sample?

I will address the above questions in my analysis below. This will include SQL codes, corresponding CSV outputs, descriptions, and visualizations.

SPL Library Search: Twilight Book

• How many times has the Twilight book been checked out and returned?

```
SELECT
    *
FROM
    spl_2016.inraw
WHERE
    title = 'Twilight'
        AND itemtype LIKE '%bk'
        AND callNumber = 'YA MEYER'
```

- Data & Data Results : num_of_checkouts.csv

From the query, we get 8,065 rows returned. This means that the book *Twilight* has been checkout out and returned a total of 8,065 times. This number is significant to use in the following assignment and data analysis.

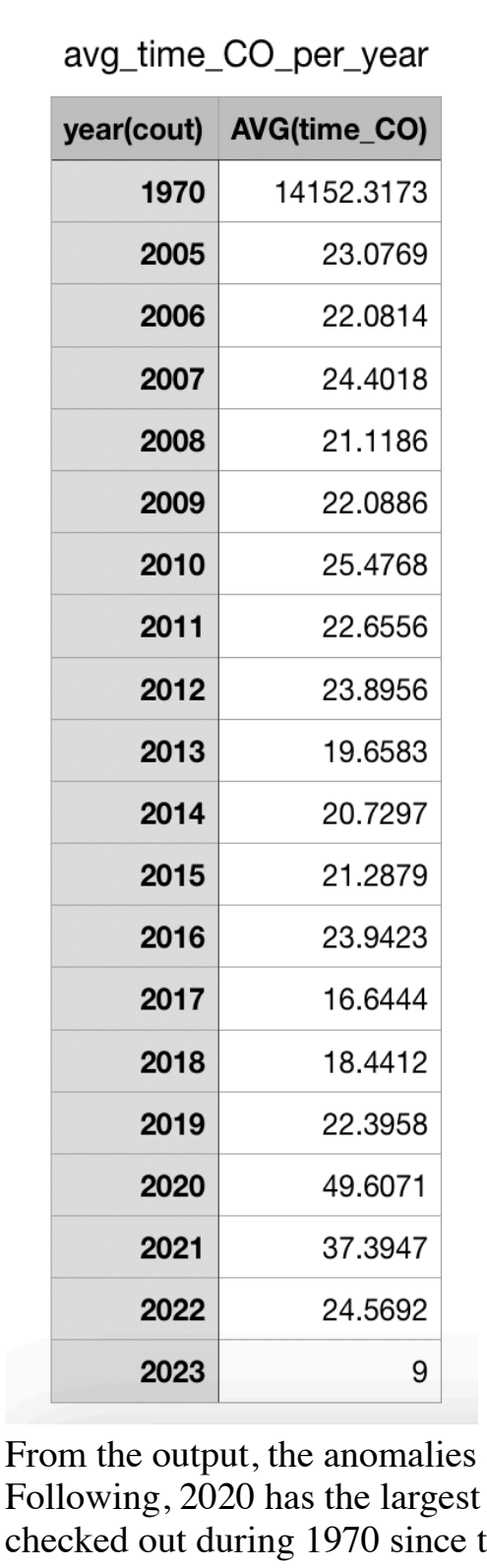
Also, notice that each has a the item type of 'acbk' which is Adult/YA Book.

• Average time checked out in days

First, I am going to aggregate the average time checkout out per year.

```
SELECT
    YEAR(cout), AVG(time_CO)
FROM
    (SELECT
        cout, cin, DATEDIFF(cin, cout) AS time_CO
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER') sub
GROUP BY YEAR(cout)
ORDER BY YEAR(cout) ASC;
```

- Data & Data Results :



avg_time_CO_per_year

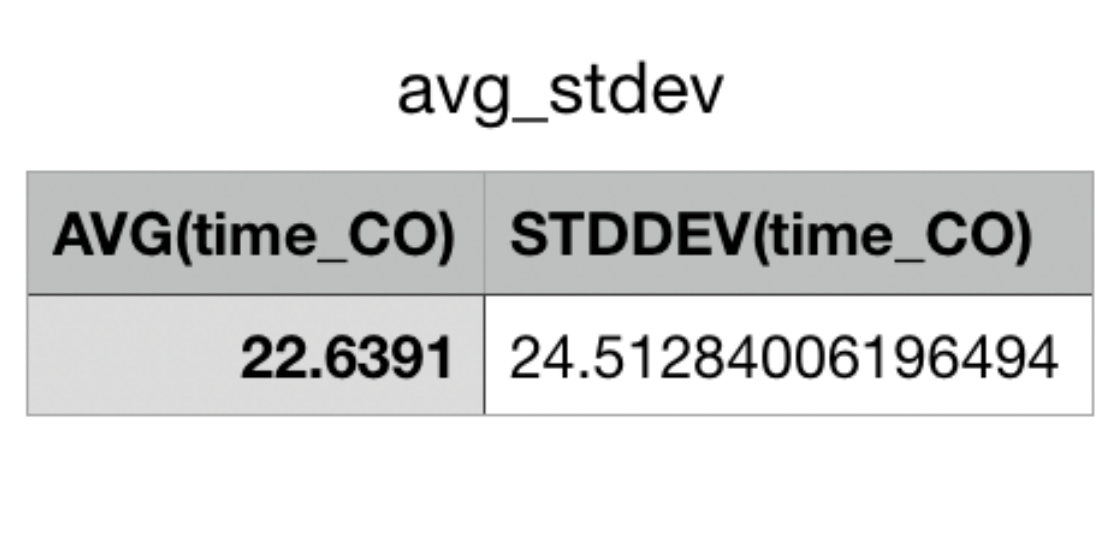
year(cout)	AVG(time_CO)
1970	14152.3173
2005	23.0769
2006	22.0814
2007	24.4018
2008	21.1186
2009	22.0886
2010	25.4768
2011	22.6556
2012	23.8956
2013	19.6583
2014	20.7297
2015	21.2879
2016	23.9423
2017	16.6444
2018	18.4412
2019	22.3958
2020	49.6071
2021	37.3947
2022	24.5692
2023	9

From the output, the anomalies of books being checkout out during 1970 is made clear. We know this cannot be accurate due to the nature of the database. Following, 2020 has the largest average of 49.61 days while 2023 has the minimum number of days for checkout; only 9. Moving on, I will exclude books checked out during 1970 since those will throw off the average checkout time significantly. However, they are clearly outliers but due to a data entry error! I am looking for outliers steemed from human behavior.

• Average & Standard Deviation Over all time

```
SELECT
        AVG(time_CO), STDDEV(time_CO)
FROM
    (SELECT
        cout, cin, DATEDIFF(cin, cout) AS time_CO
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER'
            AND YEAR(cout) != 1970) sub;
```

- Data & Data Results :



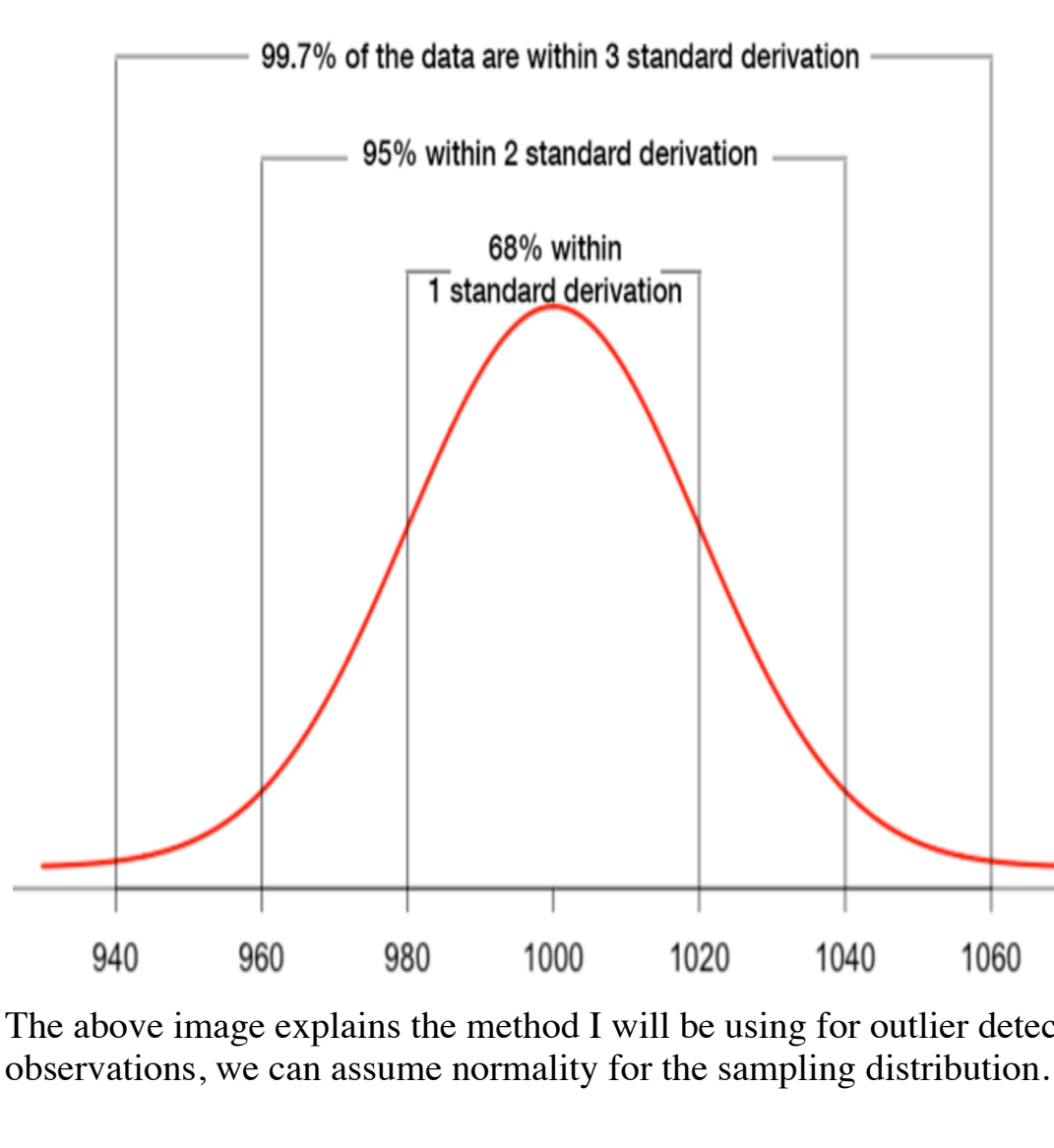
avg_stdev

AVG(time_CO)	STDDEV(time_CO)
22.6391	24.51284006196494

The average amount of time the the first twilight book was checked out from the SPL is 22.64 days and the standard deviation is 24.51. This is the average and standard deviation that I will be using through the rest of my analysis to determine outliers in the dataset.

• Finding Outliers Using the Statistics

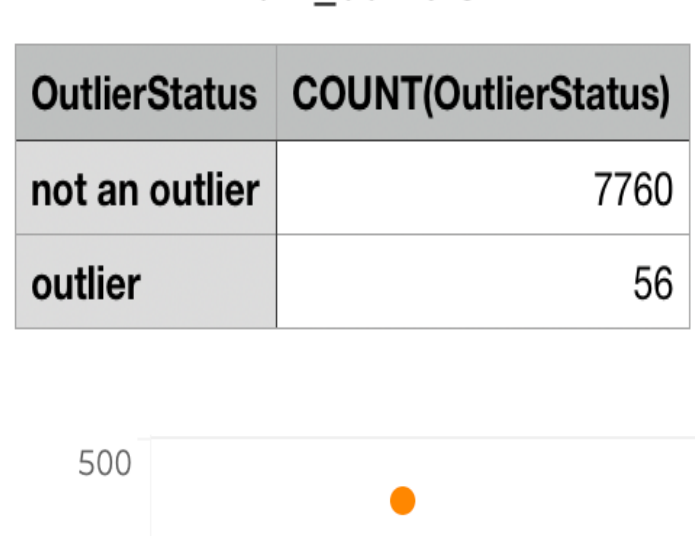
I will be detecting outliers using by detecting observations that are outside 3 standard deviations of the mean. Since it is impossible for a book to be checked out for a negative period of days, I will only consider outliers that are 3 standard deviations above the mean. Also, I will round this upper quantile to the nearest whole number since the DATEDIFF() calculation in SQL is taking the difference between the cout and cin date in number of days.


99.7% of the data are within 3 standard derivation
95% within 2 standard derivation
68% within
1 standard derivation
940
960
980
1000
1020
1040
1060

The above image explains the method I will be using for outlier detection. By the Central Limit Theorem, and the fact that out sample has 8,065 observations, we can assume normality for the sampling distribution. The below query will find the outliers in the desired sample.

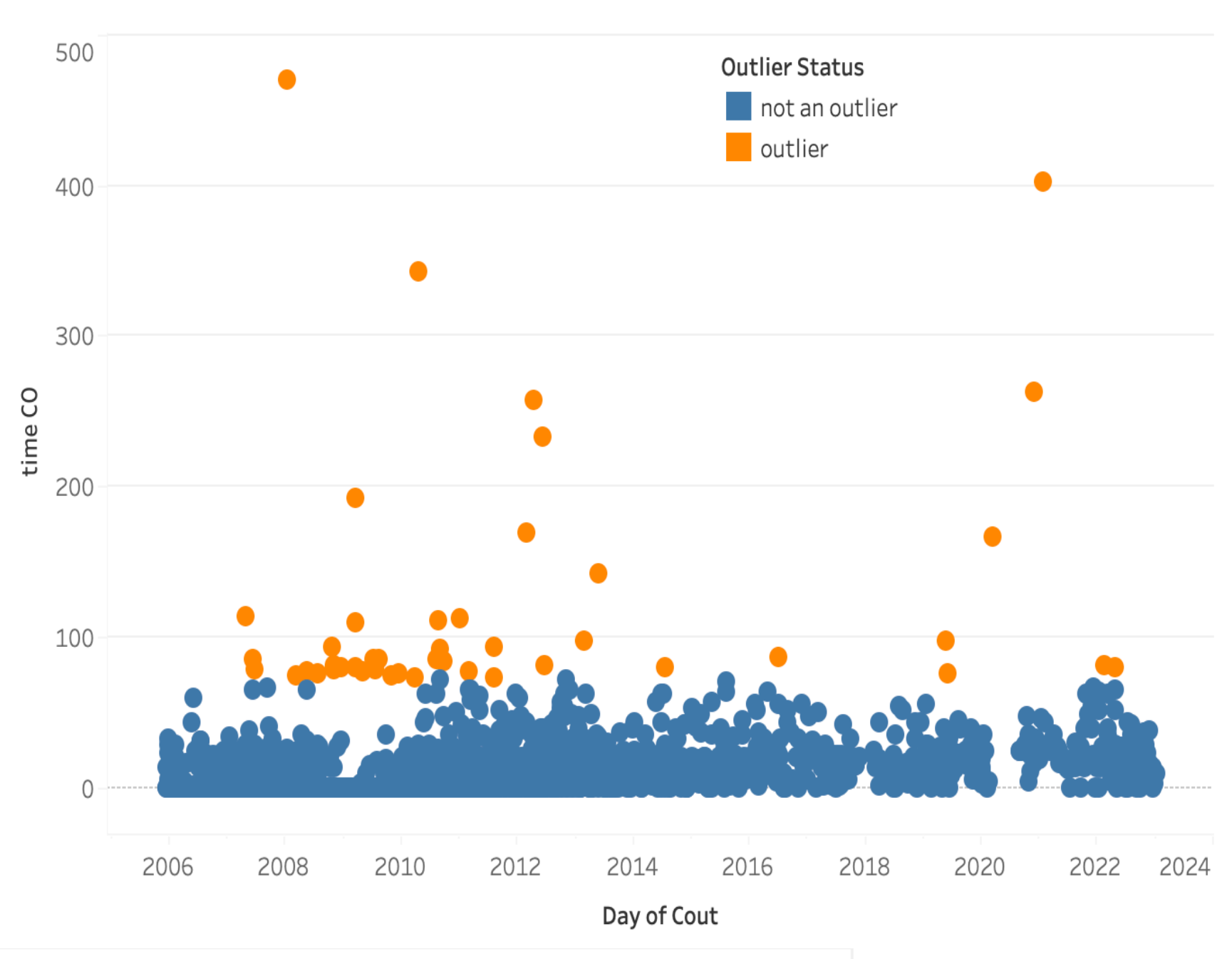
```
SELECT
    OutlierStatus, COUNT(OutlierStatus)
FROM
    (SELECT
        time_CO,
            CASE
                WHEN time_CO > 96 THEN 'outlier'
                ELSE 'not an outlier'
            END AS OutlierStatus
    FROM
        (SELECT
        cout, cin, DATEDIFF(cin, cout) AS time_CO
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER'
            AND YEAR(cout) != 1970) sub) sub2
GROUP BY OutlierStatus;
```

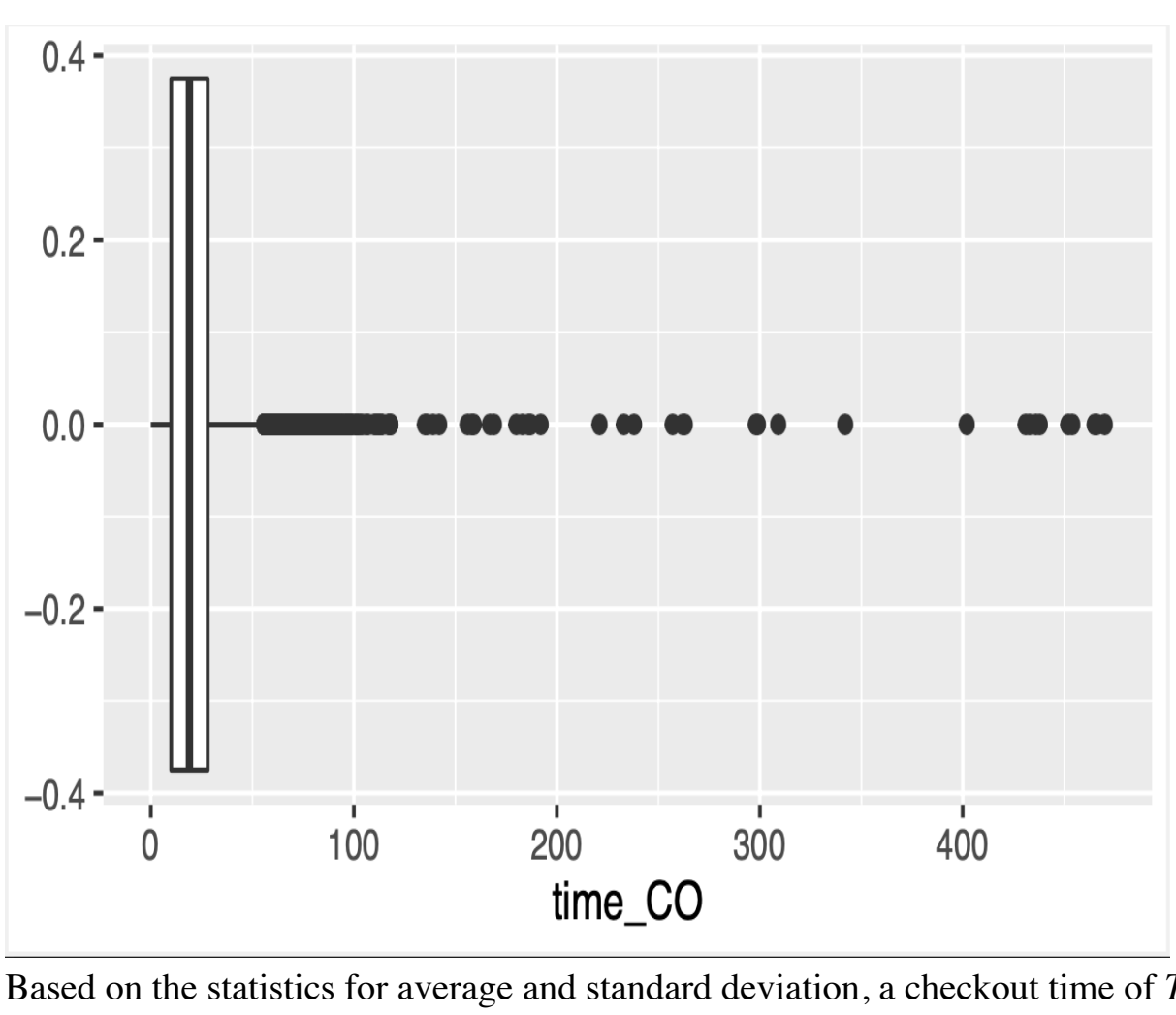
- Data & Data Results :



num_outliers

OutlierStatus	COUNT(OutlierStatus)
not an outlier	7760
outlier	56





Based on the statistics for average and standard deviation, a checkout time of *Twilight* is deemed an outlier if it is checked out for longer than 96 days. There are 56 outliers and 7,760 non-outliers. From the outliers, we see that a majority of the outliers ranged from 2007 to 2013. There were none during 2017 and few during the time period of 2019-2022.

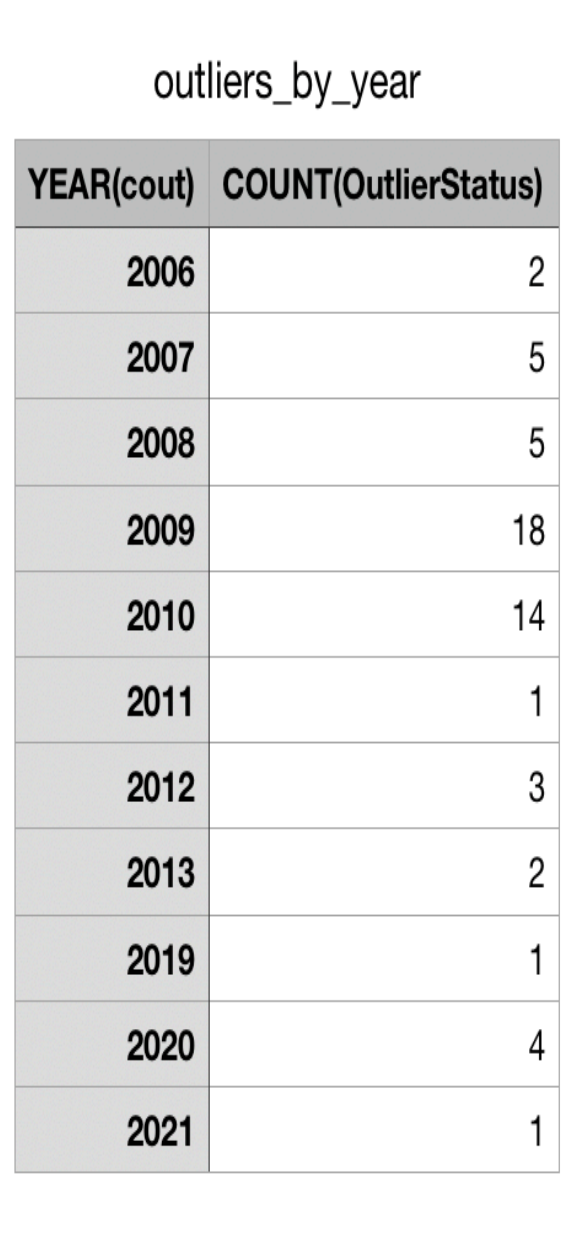
• Collective Outliers:

It is noticed that a majority of the outliers are grouped together. Also, a lot of the data points on the scatter plot do not seem "too bad" or rather "too far away" to be considered outliers. These are called **collective outliers**. "The individual data points within a collective outlier may not seem like a point outlier or a contextual outlier. It's when the data points are considered as a collection that anomalous patterns are observed." Essentially, a group of outlier points are occurred at the same time.

• Outlier Correlation with Movie Release?

Twilight by Stephanie Meyers was released on October 5, 2005. The movie for this book was released on November 17, 2008. Given these key years, I am going to filter the number of outliers by year of checkout to see if there are any patterns to observer here.

```
SELECT
    YEAR(cout), COUNT(OutlierStatus)
FROM
    (SELECT
        cout,
            time_CO,
            CASE
                WHEN time_CO > 96 THEN 'outlier'
                ELSE 'not an outlier'
            END AS OutlierStatus
    FROM
        (SELECT
        cout, cin, DATEDIFF(cin, cout) AS time_CO
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER'
            AND YEAR(cout) != 1970) sub) sub2
WHERE
    OutlierStatus = 'outlier'
GROUP BY 1
ORDER BY 1 ASC;
```



outliers_by_year

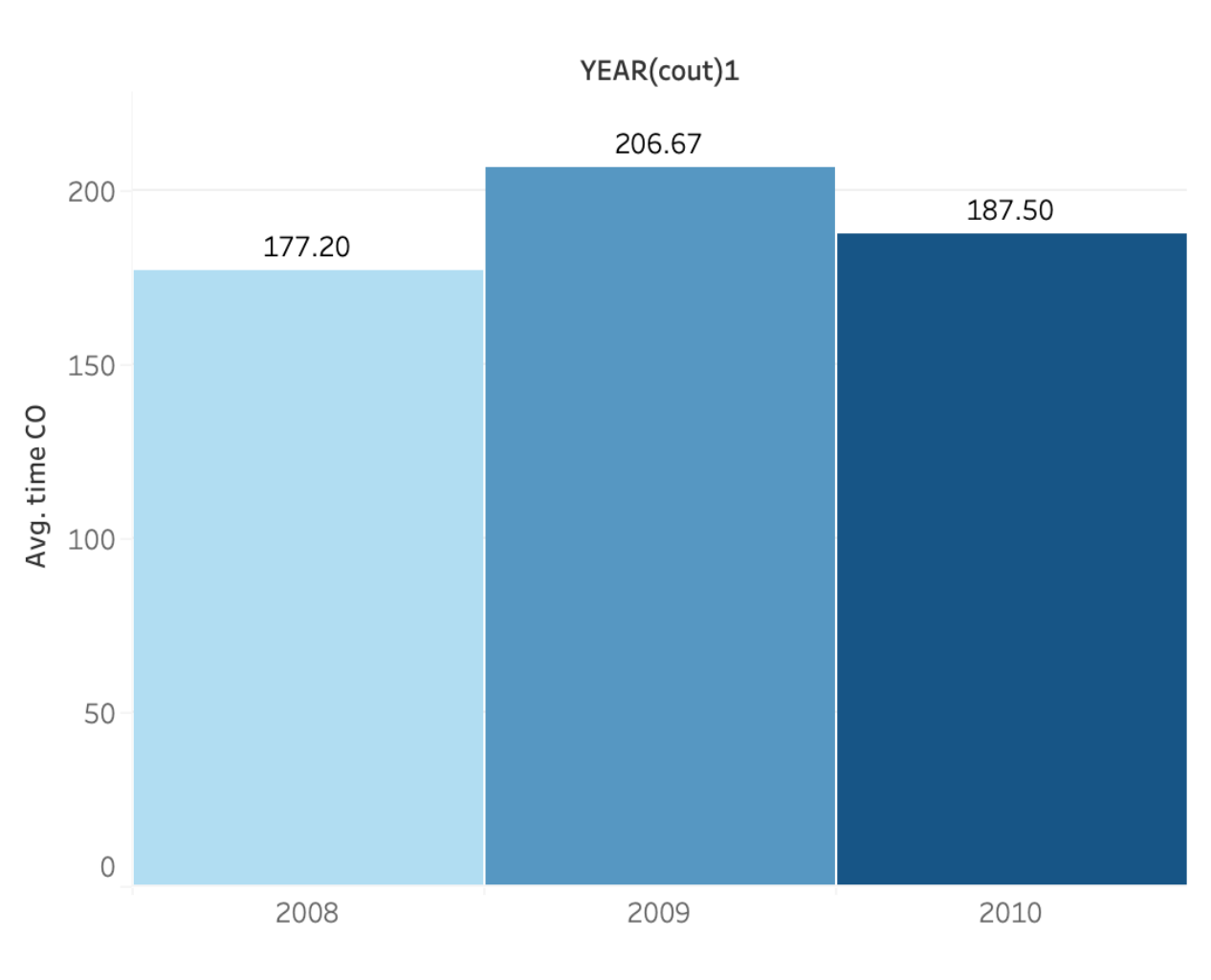
YEAR(cout)	COUNT(OutlierStatus)
2006	2
2007	5
2008	5
2009	18
2010	14
2011	1
2012	3
2013	2
2019	1
2020	4
2021	1

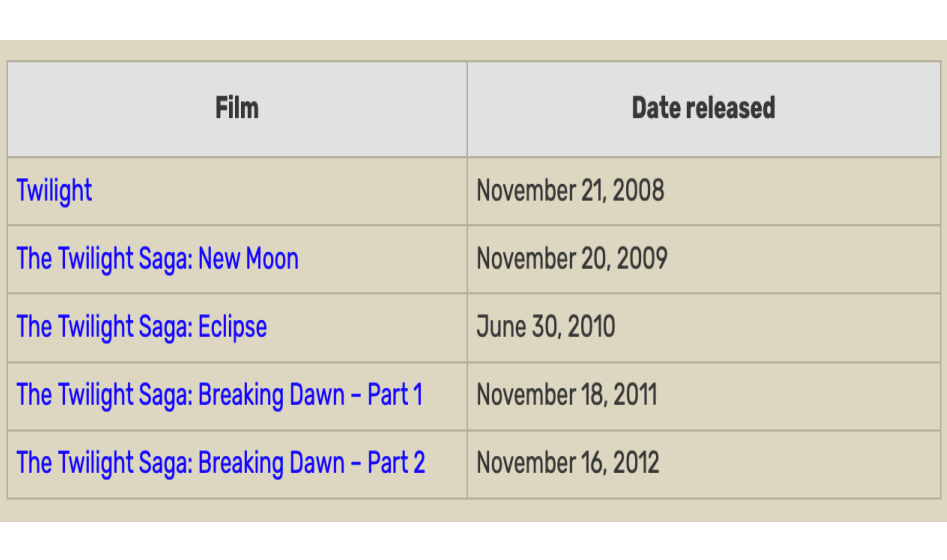
Most of the outliers are checkouts occurring in 2009 and 2010. The highest number is seen in 2009 which is also the year that had the highest number of checkouts over the course of the data base. 2008-2010 is the time period in which the movie had been released to the general public and to theaters, but not yet to the SPL.

Next, I will take a look closer at how long these checkouts were for. I will visualize the results for better clarity.

```
SELECT
    YEAR(cout), time_CO
FROM
    (SELECT
        cout,
            time_CO,
            CASE
                WHEN time_CO > 96 THEN 'outlier'
                ELSE 'not an outlier'
            END AS OutlierStatus
    FROM
        (SELECT
        cout, cin, DATEDIFF(cin, cout) AS time_CO
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER'
            AND YEAR(cout) != 1970) sub) sub2
WHERE
    OutlierStatus = 'outlier'
        AND YEAR(cout) >= 2008
        AND YEAR(cout) <= 2010
ORDER BY YEAR(cout) ASC;
```

- Data & Data Results : outliers_08_09_10.csv





Film	Date released
Twilight	November 21, 2008
The Twilight Saga: New Moon	November 20, 2009
The Twilight Saga: Eclipse	June 30, 2010
The Twilight Saga: Breaking Dawn – Part 1	November 18, 2011
The Twilight Saga: Breaking Dawn – Part 2	November 16, 2012

The graph shows the average checkout time for each year (outliers only!). 2009 has the highest average of 206.74 days. I also included the movie release dates. Although the book series was very popular, the movies coming out in theatre's brought a lot of fame to the storyline. The high average in 2009 can be a result of many people being drawn to the story line and checking out the book. They were drawn to the fandom, but never got a chance to read the book most likely leaving it as a long checkout.

It is also to note that the library borrow time is only 3 weeks or 21 days for books checked out. You can renew the item once this period ends, but as far as I know our database does not register renewals. Below is their policy on late/lost items:

"When an item is 31 days overdue we consider it lost and charge you for the replacement cost. We remove the charge for replacement when you return the late item."

The website also states, "We will temporarily suspend your Library account when an item is 14 days overdue."

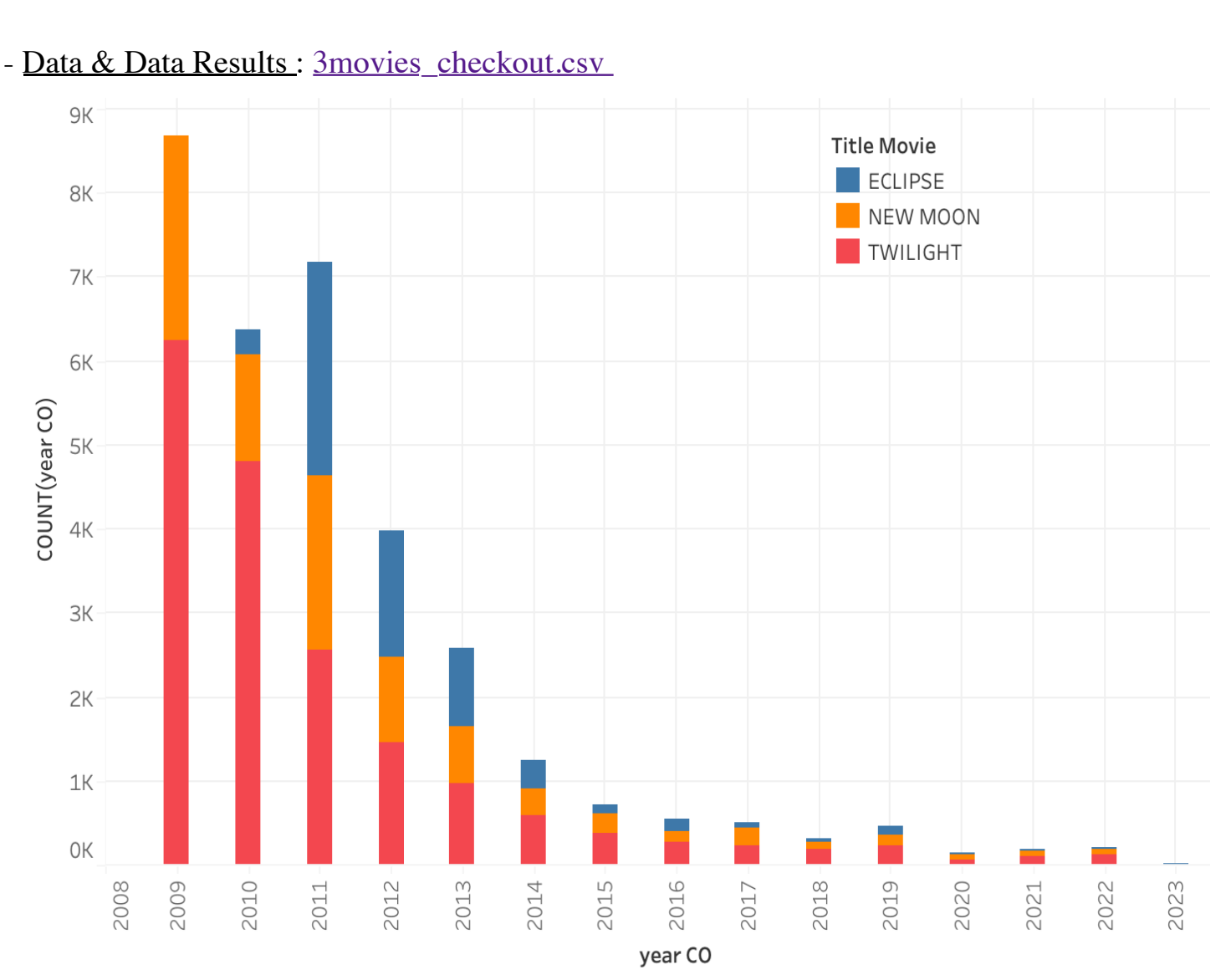
Since the SPL was able to eliminate late fines due to Washington legislature, there is not much of a backfire for holding on to an item for long periods of time. The people who checked out the book *Twilight* for an extended period of time likely did not face any financial burden. Rather, their library card was just put on hold until they returned the book.

(Reference Link)

Furthermore, I was curious about how the movies affected the set of outliers in checkout time. I decided to investigate the popularity in checkouts of the first three movies. Those were the ones that came out during the cluster of outliers that we are observing. Below is a query, data table, and visualization displaying results.

```
 SELECT
    year_CO, title_movie, COUNT(year_CO)
FROM
    (SELECT
        YEAR(cout) AS year_CO, UPPER(title) as title_movie
    FROM
        spl_2016.outraw
    WHERE
        title = 'Twilight' OR title = 'New Moon'
            OR title = 'Eclipse'
            AND itemtype LIKE '%dvd') sub
GROUP BY 1 , 2
ORDER BY 1 ASC;
```

- Data & Data Results : 3movies_checkout.csv



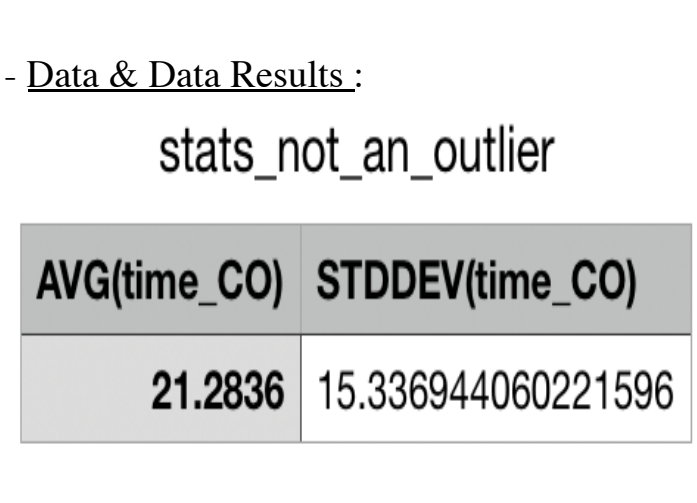
It is seen that the checkouts for the movies were the most popular during 2009, 2010, and 2011. That is a 1 year delay of the 3 year cluster we see of the outlier cluster. This can give us some intellectual insight. It is possible that people checked out the book then simply forgot about the physical copy once the movies came to the library. With 1,000s of checkouts at the SPL, it is clear they were a popular item to checkout.

• Average & Standard Deviation Removing Outliers

Now, I will run a query calculation the Average and Standard Deviation of the population (*with the outliers removed*). This will show how the outliers have affected the major statistics that are often used to categorize a population.

```
SELECT
    AVG(time_CO), STDDEV(time_CO)
FROM
    (SELECT
        time_CO
    FROM
        (SELECT
        cout,
            time_CO,
            CASE
                WHEN time_CO > 96 THEN 'outlier'
                ELSE 'not an outlier'
            END AS OutlierStatus
    FROM
        (SELECT
        cout, cin, DATEDIFF(cin, cout) AS time_CO
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER'
            AND YEAR(cout) != 1970) sub) sub2
    WHERE
        OutlierStatus = 'not an outlier') sub3;
```

- Data & Data Results :



stats_not_an_outlier

AVG(time_CO)	STDDEV(time_CO)
21.2836	15.336944060221596

With the outliers included in the sample, the **Average was 22.64** and the **Standard Deviation was 24.51**. Therefore, the average did not change much when removing the outliers. However, the standard deviation decreased significantly. This means that the deviation of the sample has decreased, which makes sense.

• Why are the outliers in checkout time occurring?

Below are 3 possible hypotheses that I have for why there are 56 outliers in the sample for checkout time. Recall that these outliers are checkout periods that exceeded 96 days.

→ Movie Correlation:

The outliers occurred at the same time that the movie checkout spikes. Thus, may be some correlation between long checkout times and movie checkouts between SPL library customers.

→ 2009 Library Shutdown:

In 2009, from Monday, August 31 through Sunday, September 6 the Seattle Public Library shut down. The shut down was due to city wide budget cuts. In order to stay on budget (a budget that was reduced by $1 million), the SPL was forced to close all of their locations for 1 week. During the closure no books could be checked out nor returned (even by book drop off). Furthermore, the website was non-accessible and there was no access to the online catalog to search, place holds, or check one's library record.

The one week library shutdown occurred during the same year as the year with the maximum number of outliers (2009). It is a possibility that readers checked out their books sometime in 2009 and meant to return it during the week of the shut down. Furthermore, they were forced to return them further on in the year causing a large gap in checkout periods. It is plausible that the closure caused confusion for SPL customers in when they can return their items back to the library.

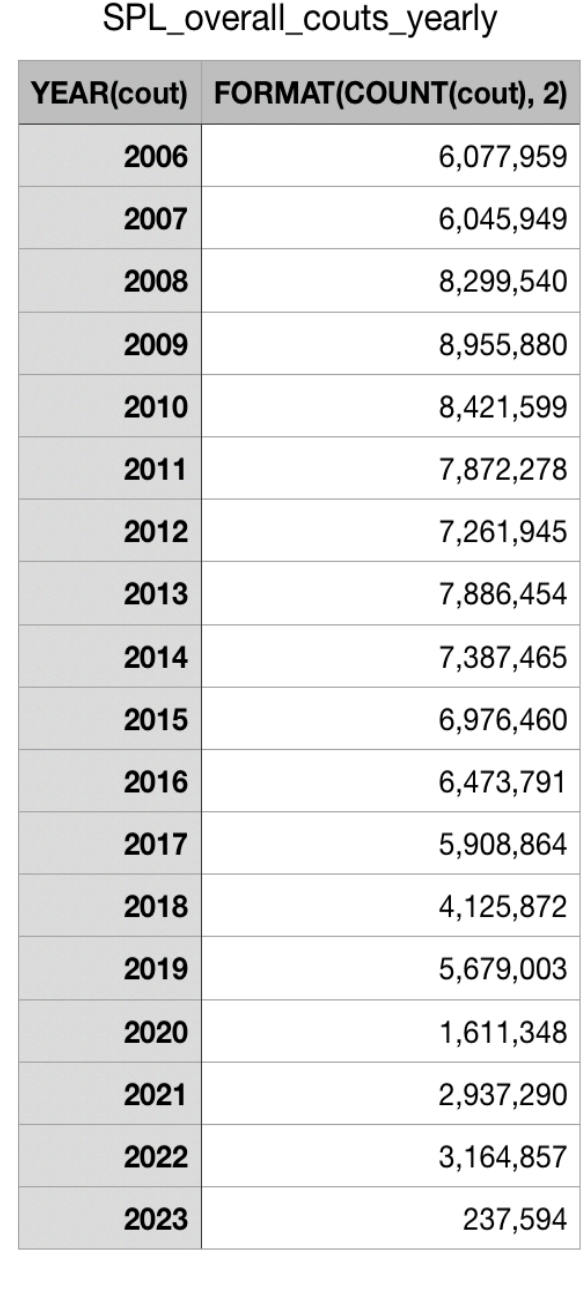
News Article

→ Library Popularity:

I am going to query for the number of checkouts and checkins (returns) per year. This will measure popularity at the SPL over the years.

```
SELECT
    YEAR(cout), FORMAT(COUNT(cout), 2)
FROM
    spl_2016.outraw
GROUP BY 1
ORDER BY 1 ASC;
```

- Data & Data Results : (Checkouts)

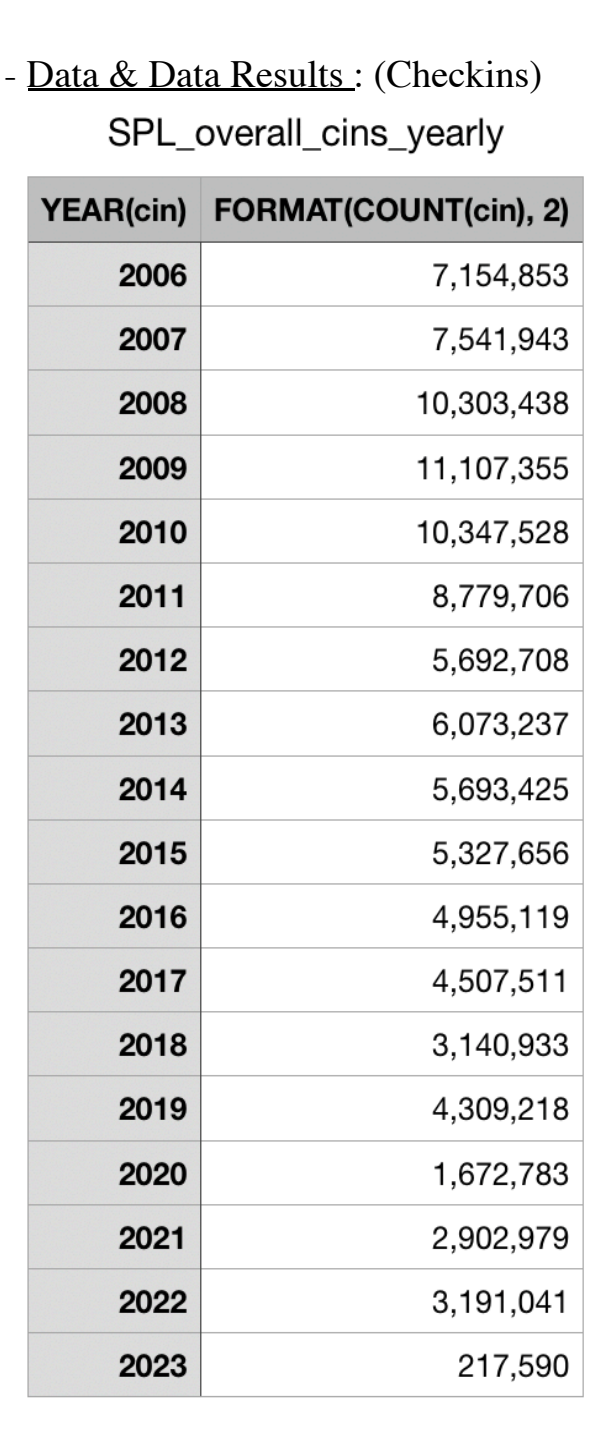


SPL_overall_couts_yearly

YEAR(cout)	FORMAT(COUNT(cout), 2)
2006	6,077,959
2007	6,045,949
2008	8,299,540
2009	8,955,880
2010	8,421,599
2011	7,872,278
2012	7,261,945
2013	7,886,454
2014	7,387,465
2015	6,976,460
2016	6,473,791
2017	5,908,864
2018	4,125,872
2019	5,679,003
2020	1,611,348
2021	2,937,290
2022	3,164,857
2023	237,594

```
SELECT
    YEAR(cin), FORMAT(COUNT(cin), 2)
FROM
    spl_2016.inraw
WHERE YEAR(cin) >= 2006
GROUP BY 1
ORDER BY 1 ASC;
```

- Data & Data Results : (Checkins)



SPL_overall_cins_yearly

YEAR(cin)	FORMAT(COUNT(cin), 2)
2006	7,154,853
2007	7,541,943
2008	10,303,438
2009	11,107,355
2010	10,347,528
2011	8,779,706
2012	5,692,708
2013	6,073,237
2014	5,693,425
2015	5,327,656
2016	4,955,119
2017	4,507,511
2018	3,140,933
2019	4,309,218
2020	1,672,783
2021	2,902,979
2022	3,191,041
2023	217,590

From the above two query outputs, it is seen that 2009 has the highest number of checkouts and checkins, followed by 2008 and 2010. This is the same result as the number of outliers in our sample. It could be that these years have more outliers since the library was the most popular during those years. However, the outliers distribution and the checkout distribution do not follow the same trend overall.

It is likely that the true reason of the outliers' occurance is not solely one of these, but a combination of the above 3 hypotheses.

★ Moving forward, how do I think we can stop these outliers from occurring?

The outliers are data points occurring from very long checkout times. In order to stop them from occurring, I think that the SPL would need to implement a stronger policy for long borrow times. Not much bad happens if an item is checked out for such a long period of time. If something strict were to be put into place, these outliers would not happen in the future.

• Preparing for 3D Visualization

In preparation for the remainder of the course, I want to use this research idea to produce data that can be used to create a 3D visualization in the future. This means that the output data will need to have 5 columns. Below are the 5 columns that I will query for using MySQL.

1. cout (Date of checkout from SPL) - axis position
2. cin (Date of return to SPL) - axis position
3. OutlierStatus ('outlier' or 'not an outlier') - color
4. ItemNumber (unique # per physical item based on acquisition date) - axis position
5. barcode (tracker placed by SPL to be scanned)- labeling each individual cell

```
SELECT
    DATE(cout),
    DATE(cin),
    CASE
        WHEN time_CO > 96 THEN 'outlier'
        ELSE 'not an outlier'
    END AS OutlierStatus,
    itemNumber,
    barcode
FROM
    (SELECT
        cout,
            cin,
            DATEDIFF(cin, cout) AS time_CO,
            itemNumber,
            barcode
    FROM
        spl_2016.inraw
    WHERE
        title = 'Twilight'
            AND itemtype LIKE '%bk'
            AND callNumber = 'YA MEYER'
            AND YEAR(cout) != 1970) sub;
```

- Data & Data Results : 3D_visualization_prep_5columns.csv

There is not much to analyze from the output csv, I just wanted to practice writing such query in preparation for future projects.

• Discussion/Analysis of Results

In conclusion, this project allowed me to detect outliers from a sample in the SPL data base. I learned about a certain type of outliers, *collective outliers*, that explained what was going on situationally. Through further research and querying I answered why and where these outliers occurred in checkout periods of the book *Twilight* by Stephanie Meyer. The outliers in checkout time were those that were for a significantly long amount of days, majority occurring between 2008 and 2010. I believe that these outliers occurred due to the movie's release, library shutdown, and library's popularity during the same years. Looking forward, I ended the project by creating a 5 column query that can be used in a 3D visualization.

• References

https://sql-snippets.count.co/t/identify-outliers-in-bigquery/169
https://seattle.bibliocommons.com/v2/search?query=twilight&searchType=keyword&f_FORMAT=BK
https://total-movies.fandom.com/wiki/The_Twilight_Saga_(Film_series)
https://www.geeksforgeeks.org/types-of-outliers-in-data-mining/
https://www.seldon.io/outlier-detection-and-analysis-methods