Project Report

# GitHub URL

https://github.com/asharkman/UCDPA\_paulmason

# Abstract

This is a project to analyse real world scenarios and generate valuable insights.

Being completely honest, I wouldn’t read this doc. Everything that’s in here, is better laid out in the jupyter notebook section. I literally just spotted this before I was uploading it and squeezed what I could onto it.

# Introduction

This is a topic I've been interested in since I was a child and now that I have a son we're both enjoying it. I felt that Lego sets have changed since I was a child. I was interested to see how they had changed. Were the pieces smaller? Were there more sets to choose from? What colours were used now compared to the past. Were there more colours used? And how big were sets then compared to now? Are there more themed sets now? Once I started to look I found the Lego data set on Kaggle. It looked like a nice complete data set. It was well broken down and laid out nice. Kaggle is excellent because you can see the reviews and comments people made on the dataset. This data set is actually originally from rebrickable.com. There is a newer dataset there but I thought it better to go with one that was tried and tested.

# Dataset

Once I started to look I found the Lego data set on Kaggle. It looked like a nice complete data set. It was well broken down and laid out nice. Kaggle is excellent because you can see the reviews and comments people made on the dataset. This data set is actually originally from rebrickable.com. There is a newer dataset there but I thought it better to go with one that was tried and tested.

# Implementation Process

First I imported the data from the 8 csv files. The ERD diagram that came with the database proved extremely helpful in identify how everything was hung together.

We could see that most table shad an ID column, those that didn’t I added a column to them in case they were needed later.

I printed the head of all of the newly imported csv files. This prints the first few columns of every dataframe and includes data types for each column.

The data looks great. Super well formed and linked together nicely with a correctly functioning primary and foreign key system. We can find out the dimensions of each set of data that we have.

Next we printed the shape of each dataframe. This gives you an idea of the columns and rows for each dataframe.

I then did a describe on the colour dataset.

Looking at this we can see our main dataset is probably going to be Inventory Parts (inv\_parts). Which makes sense, it's a dataset that contains what part belongs in which inventory. An inventory is the bag that comes in a box of Lego. Sometimes there's more than one bag(inventory) in a box(set). One non obvious column is parent\_ID in themes. This is infact the parent ID of that theme. Which is self referenceing. So forinstnace, you could have a "Space" theme, ID 126, but "Ice Planet 2002" ID 133 is also a theme, but it's parent theme is "Space".

Next we look at the datatypes used.

The data looks pretty straightforward. it's mostly integers, with the occasional float and then objects as text.

We then did a sets by year to examime how many sets were released every year.

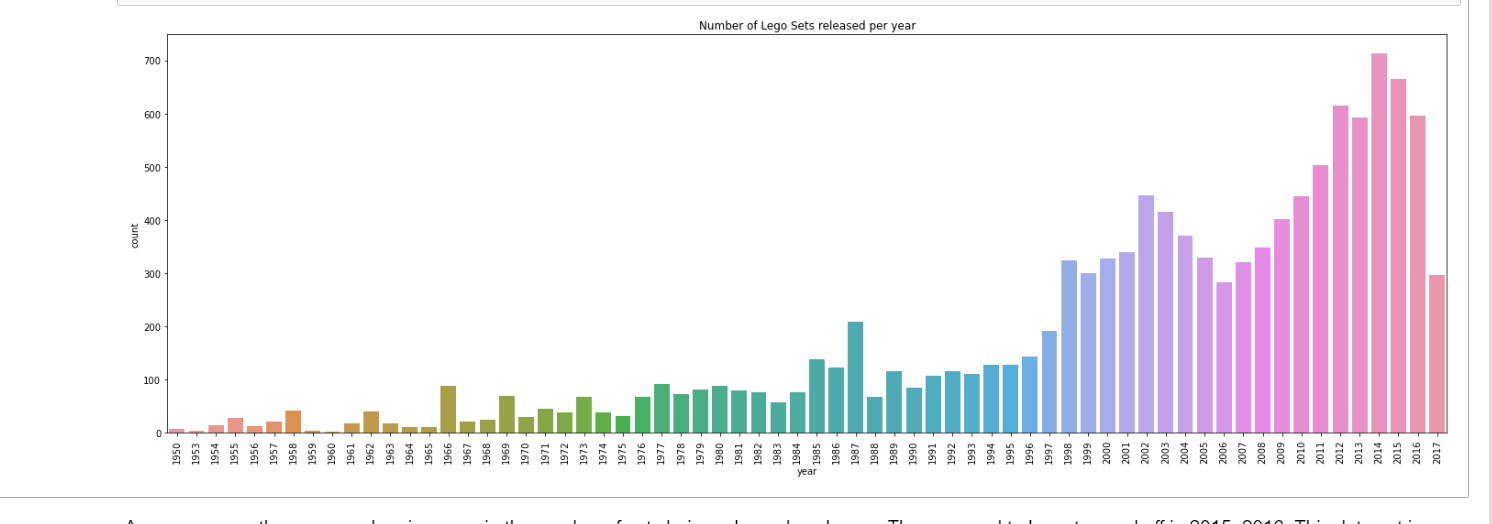
We then did a graph to make it easier to examine the values.

# Results

(Include the charts and describe them)

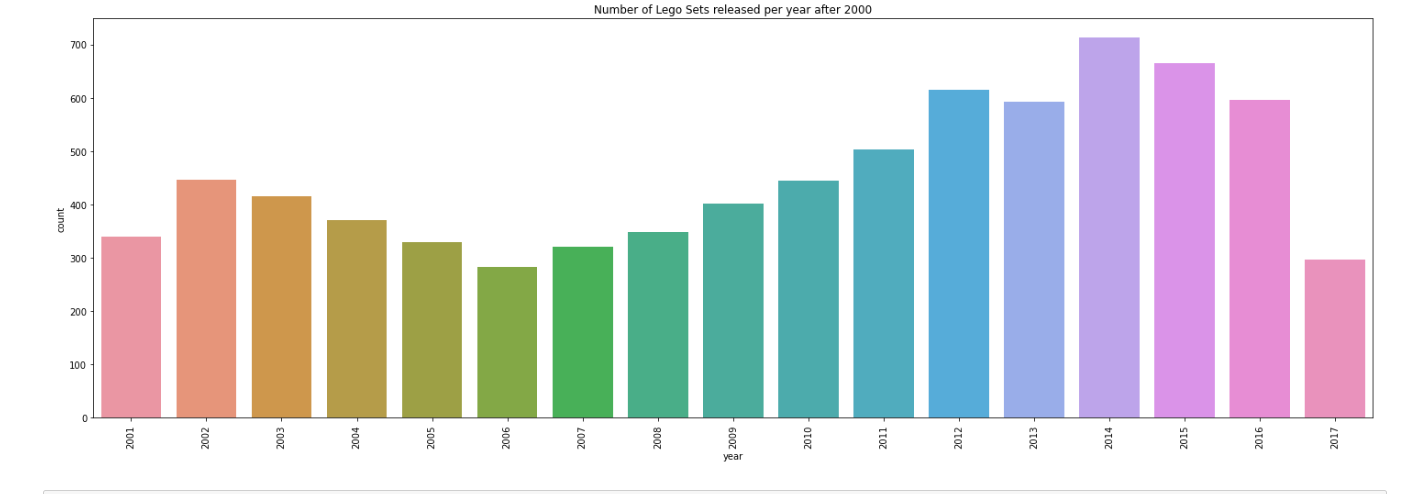
# Insights

1. Lego sets are increasing in size over the years. They took a slight dip in 2015 and 2016 (we're ignoring 2017 as it may not be a full years data)
2. If you look at historical data going back to 1950 it looks like the sets have been increasing at a rapid pace. However, if you zoom in on the data from the last 20 years the set numbers is actually quite stable.
3. The super sized sets aka sets with over 3000 pieces are all from the past 20 years so this is a new sized product that Lego have created. You can see that from the dictionary section.
4. There are peaks and throughs with the releses of sets. You can see every few years there's a peak and then it tapers off. This may relate to the production cycle or maybe the most popular sets are focused on after a large number of sets are released.
5. The lego eco system is very complicated, you can see this with the sheer number of pieces, sets, themes and then linking them all together is the inventories. Another interesting fact is that these are all traceable back to 1960. So the same system has been unaltered since then. Which says alot about the quality of the system and it's ability to be modified and improved on rather than wholesale replacement.

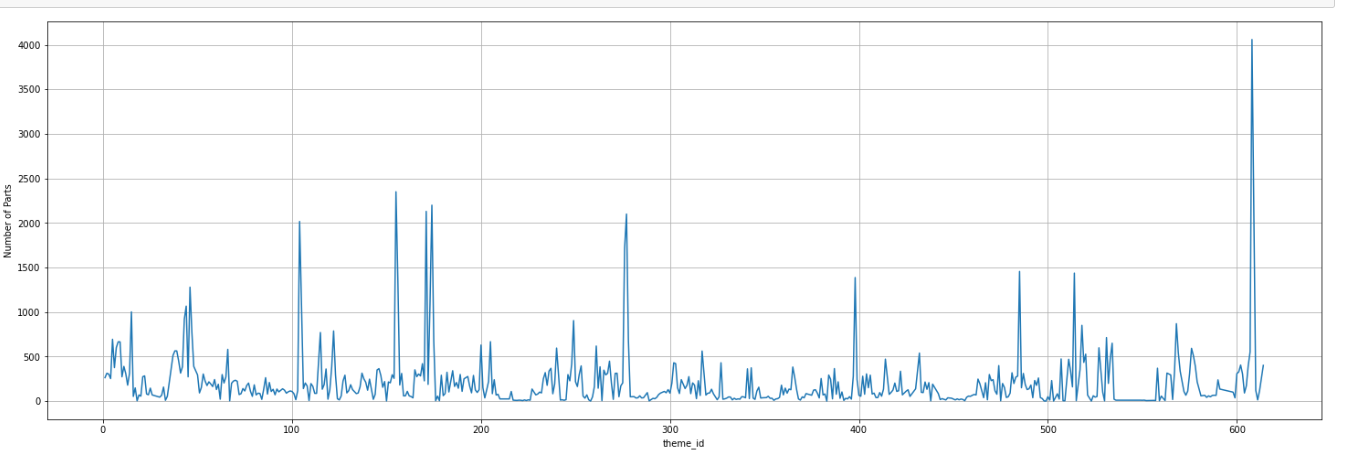


As you can see there was a clear increase in the number of sets being released each year. They seemed to have tapered off in 2015, 2016. This data set is from 2017, we don't know when, but it's possible it was mid year. So 2017 can be ignored.

Maybe we have a closer look at data from after the year 2000



I added an ID column to inv\_parts and inv\_sets. The tables have a composite key which would make them difficult to deal with later.



What i would have liked to do here is create a new dataset with the parents of each theme. Each theme acutally has sub themes. So for instance, Technic theme actually has 5 themes, it has Techinc, Artic Technic, Competition, Expert Builder and Model. So what I'd have liked to do with more time, is create a super theme column, and if a theme has a parent theme, then use that as the super theme, if it doesn't have a parent theme, then just use the ID as the super theme Value. Then build a graph of of that.

## Custom functions

We use custom functions to execute a piece of code that we'd like to reuse all the time. We could multiply part number values by any value you want. You just pass the value you want multiplied, and how much you want it multipied by.

In [43]:



**def** multiplyby(numtobemultiplied **=** 0, multiplyBy **=** 0):

total **=** numtobemultiplied **\*** multiplyBy

**return** total

​

result **=** multiplyby(2, 4)

print(result)

8

## Numpy

NumPy can be used to perform a wide variety of mathematical operations on arrays. Here we'll create a numpy array using the sets data set. We'll convert it to a numpy array first. We'll concentrate on the num\_parts field. Using numpy we'll easily be able to gather information on pieces per set.

## Merging

We'll add Suffixes for good practice. it's only added when two columns are the same. There are a few different ways to merge, but i went with this option because it gave me more control over what i was doing and made it easier to see what was happening.

# 6 Insights

1. Lego sets are increasing in size over the years. They took a slight dip in 2015 and 2016 (we're ignoring 2017 as it may not be a full years data)
2. If you look at historical data going back to 1950 it looks like the sets have been increasing at a rapid pace. However, if you zoom in on the data from the last 20 years the set numbers is actually quite stable.
3. The super sized sets aka sets with over 3000 pieces are all from the past 20 years so this is a new sized product that Lego have created. You can see that from the dictionary section.
4. There are peaks and throughs with the releses of sets. You can see every few years there's a peak and then it tapers off. This may relate to the production cycle or maybe the most popular sets are focused on after a large number of sets are released.
5. The lego eco system is very complicated, you can see this with the sheer number of pieces, sets, themes and then linking them all together is the inventories. Another interesting fact is that these are all traceable back to 1960. So the same system has been unaltered since then. Which says alot about the quality of the system and it's ability to be modified and improved on rather than wholesale replacement.

# 7. Machine Learning

Lego: Machine learning models can be employed to predict the popularity or demand of LEGO sets in the future. By analyzing historical sales data, customer reviews, set characteristics (such as theme, size, and price) these models can learn patterns and make predictions about the potential popularity of new LEGO sets. This information can assist LEGO in optimizing production.

Traffic Flow Prediction: Machine learning and deep learning models can be used to predict traffic flow patterns in the future. By analyzing historical traffic data, including variables such as traffic volume, weather conditions, time of day, and day of the week, these models can learn patterns and make predictions about future traffic congestion or traffic flow in specific areas.

In both cases, it's crucial to have high-quality and relevant data to ensure the accuracy and reliability of the predictions.

I would use regression methods, they are commonly used when you want to predict a continuous or numerical value based on input features. Here are some scenarios where regression methods are typically applied:

1. Predictive Modeling: Regression methods are often used in predictive modeling tasks where the goal is to estimate or forecast a numerical value. What we dealt with in this example is very numeric based. We've sets per year, we don't have sale prices or sales per year. But with that information you could build a prediction model based on historical data and market trends, or predicting the lifespan of a product based on how long it's been selling well.
2. Trend Analysis: Regression analysis can be employed to analyze trends and relationships between variables. For example, you can use regression to determine the impact of advertising expenditure on sales revenue, obviously, in our example you'd need to include this data. Advertising revenue per theme maybe.

# References

(Include any references if required)