Lego Sets Past, Present and Future

Specialized Models: Time Series and Survival Analysis

Table of Content

Main Objectives

When most of today's grandparents were children LEGO was not yet a household name. For many years there were only basic building blocks to be bought, and if they or their children played with LEGO this was mostly sufficient.

In today's world the situation has become much more complex. Children already suffer from the consequences if their parents don't buy or can't afford branded articles. It's no longer sufficient to wear trainers, they have to be "in" and manufactured by a certain brand. LEGO, too, follows these trends and over the years the number of sets based on films, brands or other themes important to the young today. They will be laughed at if their LEGO stash doesn't follow these trends and the don't own at least some of what their friends have. So unless their offspring or grandchildren are asking for a specific set, this makes it complicated to buy a present for them., especially for first-time buyers.

The results from this analysis are intended to help shopkeepers and (grand-) parents to chose an appropriate set, knowing which sorts of sets have been trending over the years and which are predicted to trend over the next couple of years. It can also help shopkeepers to identify shelf warmers and sell them off at a lower price before interest in them has waned completely.

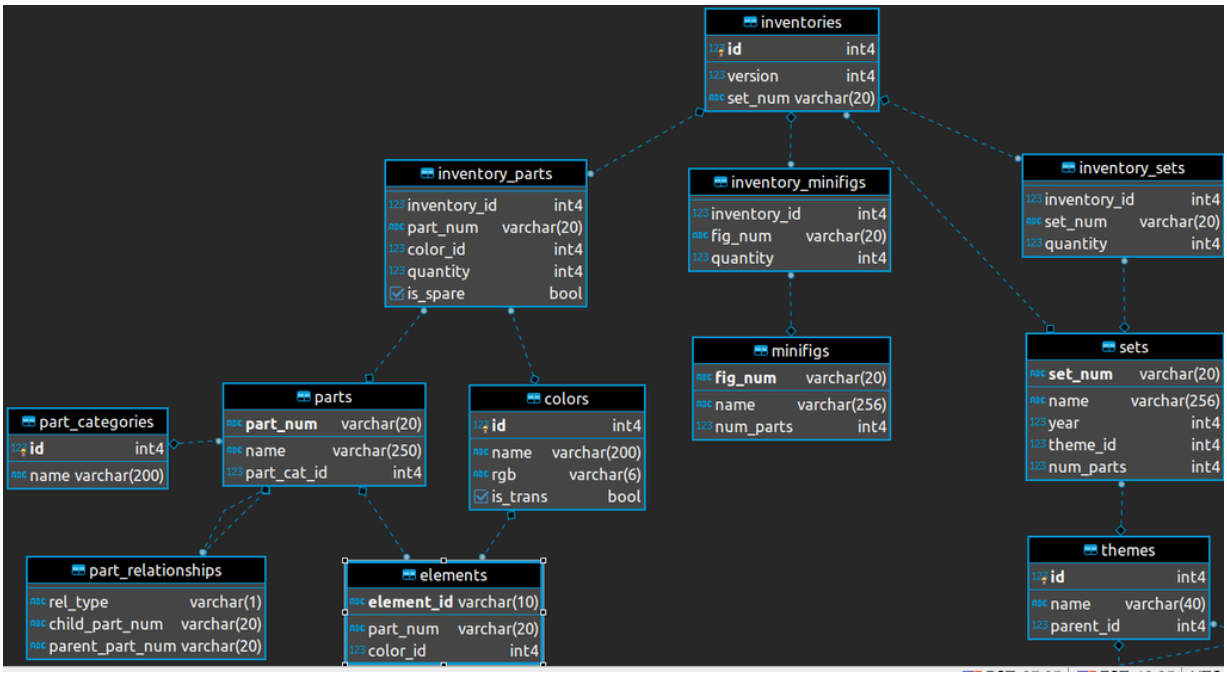
This report focuses on time series analysis by building a model that reproduces the past data accurately without overfitting it and is able to produce a forecast from its findings about the future trend in the next two decades. This is a rather large time frame for the future but is intended to also highlight a general trend with regard to re-selling all the building blocks.

The time series analysis uses the statsmodels library exclusively. This means that most people familiar with python programming will be able to reproduce the analysis at a later point in time. This analysis is mainly aimed at families and shops who are not likely to have access to less common libraries without investing more time than is available to them on top of their daily (work) lives. They also usually don't have state-of-the-art hardware and may find it difficult to install other libraries in their environment.

The Dataset

The dataset used for this analysis has been downloaded from rebrickable.com who keep an up-to-date database of anything LEGO at https://rebrickable.com/downloads/. This database contains only official LEGO items. There are a couple of competitors on the market now whose items are fully compatible with LEGO but these are not part of the database. The database was last updated on August 21, 2021.

To show how all files in the LEGO database are connected, rebrickable.com provides a database schema:



Source: www.rebrickable.com/downloads

Out of the 12 csv-files provided for each database "table", only two are of interest here. The main one, which could be used all by itself, is **sets.csv**. This lists all sets ever published, including, name, publication year, number of parts per set and also a more global theme id. This is were the second file, **themes.csv**, comes in. This not only contains the theme ids but also their names. Using this file is mainly intended to make the data exploration section more easily palatable although it will also be used for identifying the data for building models.

Data Exploration and Cleaning

Getting to Know the Data

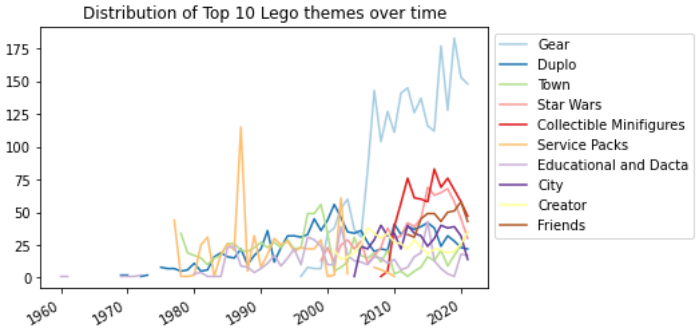
At the time of starting this analysis 18,021 LEGO sets have been published.

There are 535 different themes out of which 59 have been categorized as "parent themes" which comprise from just one to 38 sub-themes. The themes in the sets dataset only point to individual theme ids but doesn't give any indication whether these are parent themes or sub-themes. In order to get an idea of the popularity of themes it is necessary to merge the two datasets into one which adds the parent id of the theme and can be used to group by overall themes then rather than by sub-themes. These are the top 10 themes after doing so:

```
Gear                        2415
Duplo                       1218
Town                         904
Star Wars                    827
Collectible Minifigures      757
Service Packs                637
Educational and Dacta        583
City                         527
Creator                      519
Friends                      452
```

Many will say that Duplo is not LEGO because it has bigger building blocks than the rest and is for younger children. Since the models will look at specific themes there is no need to drop Duplo, and perhaps Creator as well, from the dataset.

This analysis is concerned with the past, present and future so here's a plot of how the number of sets per theme developed over time for the Top 10:

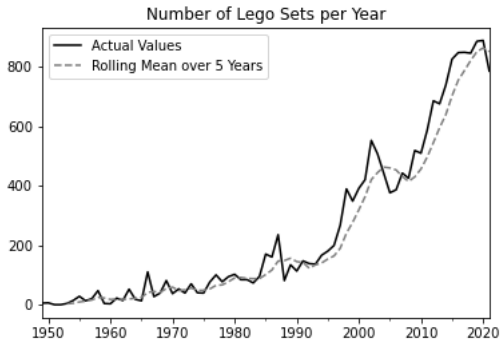


The oldest dedicated theme for LEGO sets is "Town", introduced before 1980, followed by "Star Wars" around 20 years later.

Both "Gear" and "Collectible Minifigures" skyrocketed in the early 2000s but interest in minifigures seems to be in the last couple of years while "Gear" still has an upward trend.

"Duplo" for young children has a fairly stable trend as has "Creator" for older children, teenagers and adults. It will be interesting to see how this develops once the covid-19 pandemic is in the past.

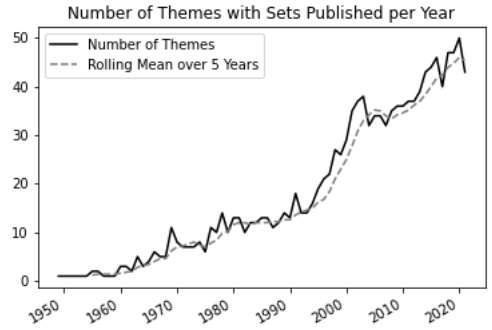
Just from looking at this one plot there's a trend of a rising number of sets over the years visible, both by churning out more sets per theme for some of them as well as introducing more themes in parallel. This remains the trend when just looking at the number of sets per year without taking themes into consideration.



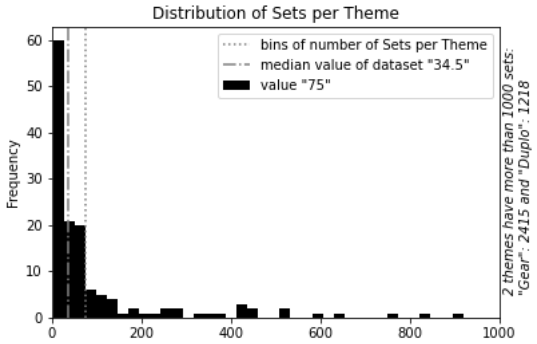
The downward trend in 2021 is probably due to the fact that there are still four more months to go before the end of the year.

Plotting the rolling mean over 5 years shows a bit more clearly that this is an exponential growth trend which has no observable seasonality despite two drops in 1988 and from 2003 through 2005.

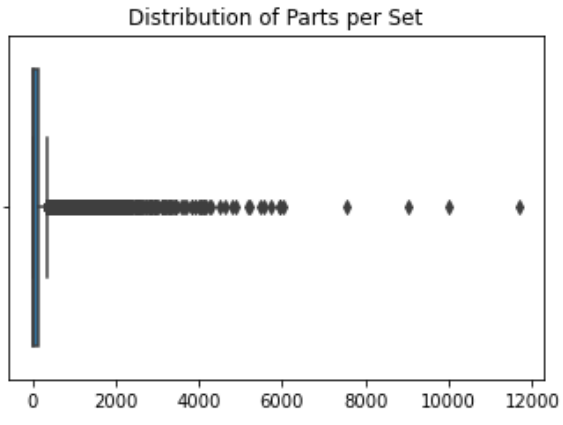
Looking at the development of themes for which sets were published in any given year the distribution shows a similar upward trend which is somewhat nearer to linear growth though.

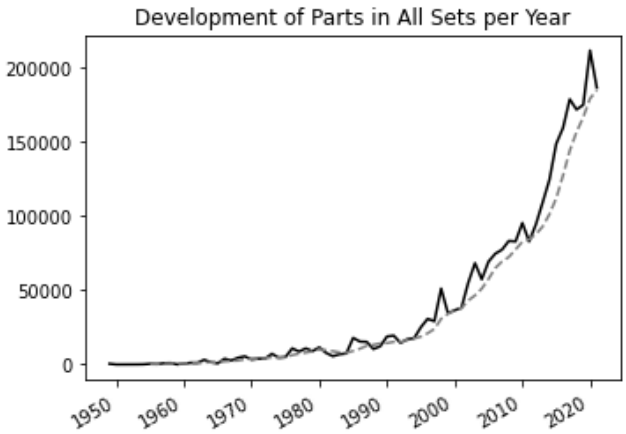


With sets for up to 50 different themes published each year families are faced with a difficult decision every time a present needs to be bought. On the other hand, the vast majority of themes has had less than 75 sets published and 50% of the themes consist of 34 or less sets.



On top of the themes and sets there is also the number of parts per set provided in the database. As might be expected most sets have relatively few parts but the span is very wide in order to provide a challenge for more experienced builders and experts. It might also be guessed from the plots above that the number of parts increases every year.





If you are looking for challenge, here are the ten sets with the most parts:

number of parts	set number	name of set	name of theme	year
5462	SWMP-1	Star Wars / M&M Mosaic - Promo Set	Star Wars	2005
5549	75978-1	Diagon Alley	Harry Potter	2020
5709	71741-1	NINJAGO City Gardens	Ninjago	2021
5922	10189-1	Taj Mahal	Creator	2008
5923	10256-1	Taj Mahal	Creator	2017
6020	71043-1	Hogwarts Castle	Harry Potter	2018
7541	75192-1	UCS Millennium Falcon	Star Wars	2017
9036	10276-1	Colosseum	Creator	2020
9987	BIGBOX-1	The Ultimate Battle for Chima	Legends of Chima	2015
11695	31203-1	World Map	LEGO Art	2021

Data Cleaning - Missing Values and Outliers

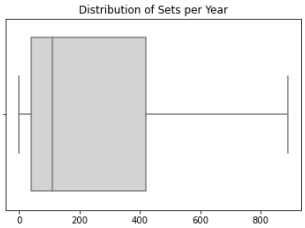
In order to get a feel for the data missing values were not important since all plots used so far can handle both missing values and missing years in the index, if a datetime index is provided. For further analysis and model building it is essential that there is no missing data in the relevant columns.

Overall development of sets over time

The original dataframe is just a list of all sets that were ever published, including the year of publications. To get the number of sets published each year the data needs to be grouped by year. Just looking at the head of this series makes it obvious that some years are missing:

```
1949-01-01     5
1950-01-01     6
1953-01-01     4
1954-01-01    14
1955-01-01    28
```

This can be addressed easily by setting the frequency of the data to "YS" (year start) and filling the missing values with 0 since there were no sets published or they would have been recorded in the original database. Checking for years without any sets only shows two years: 1951 and 1952 as was already visible from looking at the head of the series. The number of sets per year is rising every year but despite two peaks and declines there are no outliers in the data as evidenced by a boxplot:



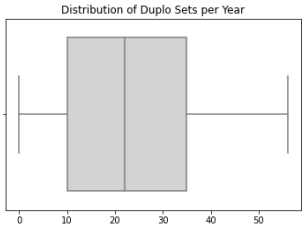
The plot shows whiskers to reflect the growing number of sets but no outliers. Standard parameters have been used.

The theme "Duplo"

Filtering the original dataset for the theme "Duplo" reduces the number of sets to 1218. No missing data is found in this dataframe.

Grouping this data by year finds that in a span from 1969 to 2021 one year is missing as the length of the series is 51, not 52. Again this can be solved by setting the frequency to "YS" and filling the missing value with 0. No set was published in 1971. Plotting the result series as a boxplot also shows no outliers.

```
1969-01-01    2
1970-01-01    2
1971-01-01    0
1972-01-01    1
1973-01-01    2
```



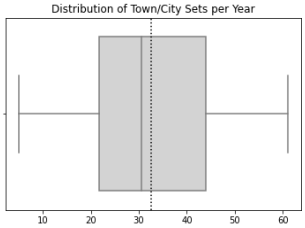
The themes "Town" and "City" combined

The first set for the theme "Town" came out in 1969. This set had 797 parts so it probably provided a lot to play with. After that there was a long drought. The next set was only published nine years later in 1978, and that was flatbed truck. More motor vehicles followed.

In 2004 the theme was extended by adding a new theme "City". The first of the sets was a city airport consisting of 914 parts. The next year a police station followed.

Both themes have remained constant favourites with many sets added in 2021, including a town centre (791 parts), a shopping street (533), a family house (388) and a fire command unit (380).

With a nine-year gap between the first two sets for "Town" it's clear that there are years missing from this series. Rather than filling the missing values with 0 it makes more sense to drop the first set from this series and continue on with data from 1978 onwards where there are no more missing values. It's also important to set the frequency to "YS" as this makes modelling easier although models can deduce the frequency on their own. This series doesn't show any outliers.

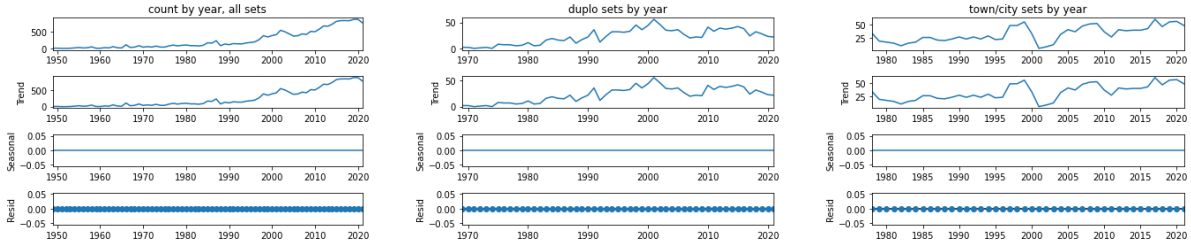


It also seems to be fairly evenly distributed around a mean value as shown by the dotted line in the centre. The whiskers are also of a comparable length. More on this later.

Trend, Seasonality and Noise

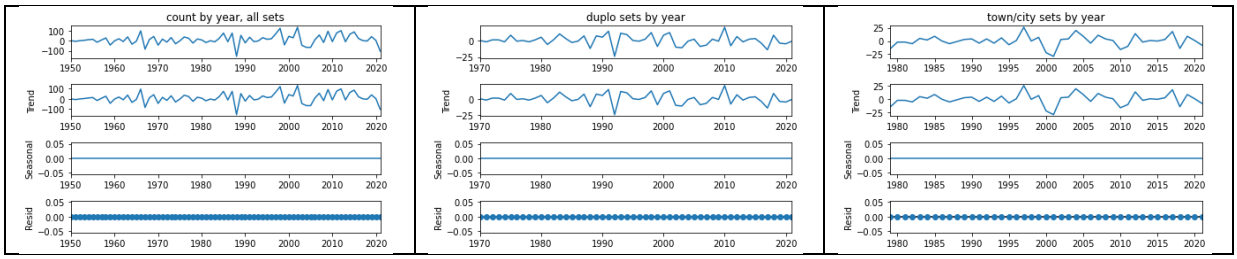
In general it can be said without having to decompose the data that neither set has any seasonality, i.e. repeating patterns over time. It also seems clear that they do have a

trend of increasing numbers of sets per year. However, there is no need to rely on line plots and the interpretation thereof as this can be plotted easily enough.



All three datasets don't show either seasonality or excessive noise. The trends are not really confirmed or rejected as it looks just the same as the original data. Since there is no seasonality in either of the datasets there is no need to use a an SARIMA model to account for it.

Taking the yearly difference between the values takes out the trend entirely for all three series but doesn't introduce seasonality or noise. This points to an ARIMA rather than a pure AR, MA or ARMA model.



Stationarity

Any time series data with a trend is unlikely to be stationary. This can confirmed using the Augmented Dickey-Fuller Test, or ADF Test, which is a statistical method provided by the statsmodels library.

The test returns three relevant statistic metrics:

1. adf: The more negative this value is, the mote likely it is that that the data is stationary.
2. p-value: The null hypothesis for the ADF Test is that the data is non-stationary. If the p-value is below a threshold of 0.05 this hypothesis can be rejected and the data is, in fact, stationary.
3. critical values: These are not entirely necessary to know but help to pin down a good adf-value. Any adf-value should be below any of the critical values.

These are the statistics returned for the three series:

sets per year	"Duplo"	"Town" & "City"
0.7483810484437359 0.9907555543990169 {'1%': -3.526004646825607 '5%': -2.9032002348069774 '10%': -2.5889948363419957}	-2.2659029306957246 0.1832008154723237 {'1%': -3.562878534649522 '5%': -2.918973284023669 '10%': -2.597393446745562}	-0.418907994312531 0.9068936312573241 {'1%': -3.6209175221605827 '5%': -2.9435394610388332 '10%': -2.6104002410518627}

As expected neither of the series is stationary. All three p-value are well above 0.05 and the null hypothesis can't be rejected. The data for "Duplo" comes closes to being stationary but even so the adf-value is also still higher than any of the critical values. Even so, the adf value is still higher than the critical values so it's necessary to transform all three series to make them stationary before fitting a model.

There are different methods for making the data stationary, the easiest and most common being to take the difference between any adjacent values. This is also used in the ARIMA models. After applying this technique all three series have become stationary:

sets per year	"Duplo"	"Town" & "City"
-9.601989949510328 1.9097051767289004e-16 {'1%': -.526004646825607 '5%': -.9032002348069774 '10%': -.5889948363419957}	-7.02098305449864 6.5444894413067e-10 {'1%': -3.568485864 '5%': -2.92135992 '10%': -2.5986616}	-5.5447915862218355 1.6702962111142508e-06 {'1%': -.6209175221605827 '5%': -2.9435394610388332 '10%': -.6104002410518627}

Please note that the p-values are now written in scientific notation corresponding to $1.9 * 10^{-16}$, $6.5 * 10^{-10}$ and $1.7 * 10^{-6}$. All three are significantly lower than 0.05.

It can now be concluded that a good, if not optimal, model will be an ARIMA(p, 1, q).

Taking the second difference only improves the metrics for the "Duplo" data. In fact, the other two have higher adf- and p-values than with taking just the first difference, making this a worse choice. Whether any ARIMA model performs better for the "Duplo" data with I = 2 or ARIMA(p, 2, q) remains to be seen when modelling.

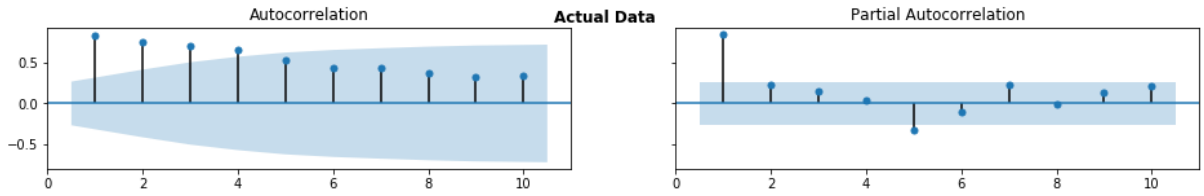
Models

In order to keep this paper to a reasonable length this and the following section only deals with the data pertaining to the "Duplo" sets. The techniques used here can be ported to any of the other set themes as well.

(Partial) autocorrelation

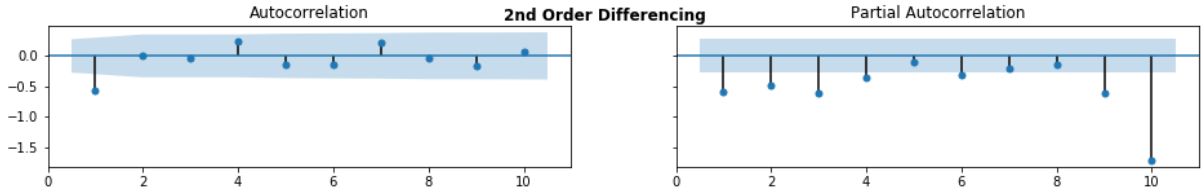
One method for finding the orders of the AR- and MA models is to use autocorrelation and partial autocorrelation plots. In the sections above it was found that model working with differenced data rather than the actual values might work better and for "Duplo" data it might even become necessary to difference twice to find an optimal model.

Model orders found from the autocorrelation and partial autocorrelation plots for the actual data cannot be ported to the differenced data so it becomes necessary to make plots for all three data series.



These plots for the actual data show that the autocorrelation is trailing off towards 0 and so is, in part, the plot for partial autocorrelation. There is also a sharp drop-off after lag 1, suggesting that an ARMA(1, 0) or ARIMA(1, 0, 0) model might be appropriate here.

The plots after taking the first difference don't show any clear pattern. Neither of the autocorrelations is trailing off and few values are statistically significant (don't fall into the shaded area). Many of the lags are also negatively correlated. There is no sharp drop-off for partial autocorrelation so this points to an ARIMA(0, 1, 1) model albeit only slightly.



Taking the difference twice doesn't make the plots much clearer. With just the first lag being statistically significant for autocorrelation an order of 1 might be assumed for the MA-part of the model. However, all lags for partial autocorrelation are negatively correlated and the only one that stands out is at lag 10. This seems an unlikely value for the AR-part of the model.

Finding reasonable order parameters

There are different methods of finding good parameters to use for the models. A reliable one is simply running a nested for-loop for each order parameter, building and fitting the model and writing the relevant statistics to a dataframe.

The relevant statistical metrics here are the aic and bic that are shown as part of the results summary of the fitted model. By itself the numbers tell us nothing but they can be compared across multiple models. The aic shows which model fits the data best while the bic indicates whether the forecast is thought to be reliable.

Actual data

model	aic	bic
ARIMA(1,0,1)	368.272305	374.183181
ARIMA(3,0,2)	368.349819	380.171571
ARIMA(4,0,2)	368.404611	382.196655

model	aic	bic
ARIMA(1,0,1)	368.272305	374.183181
ARIMA(2,0,0)	369.985291	375.896166
ARIMA(1,0,0)	372.381445	376.322029

1st order differencing

model	aic	bic
ARIMA(0,1,1)	357.949738	361.852225
ARIMA(2,1,2)	358.273988	368.030207
ARIMA(0,1,5)	358.679146	370.386608

model	aic	bic
ARIMA(0,1,1)	357.949738	361.852225
ARIMA(1,1,0)	359.893813	363.796301
ARIMA(0,1,0)	362.741510	364.692754

2nd order differencing

model	aic	bic
ARIMA(0,2,2)	358.541863	364.337340
ARIMA(3,2,3)	358.633245	372.156025
ARIMA(2,2,3)	358.696356	370.287310

model	aic	bic
ARIMA(0,2,2)	358.541863	364.337340
ARIMA(1,2,1)	360.257346	366.052823
ARIMA(0,2,1)	362.622892	366.486543

For this data the nested for-loops finds the same best order parameters for aic and bic for each order of differencing, making the decision which to use fairly easy. For the actual data both values are higher than for the differenced data, once more pointing to a differenced model.

Looking at the autocorrelation plots suggested an ARIMA(1, 0, 0) model but it turns out that this is only the third-best model found by the for-loops for bic and even worse for aic.

Looking at the (partial) autocorrelation plots suggests an ARIMA(0, 1, 1) model for first order differencing. This is confirmed by the findings of the for-loop.

The plots for second order differencing were unclear and the best orders found by the for-loop is hard to verify by looking at the plots. There is also the fact that the three best aic values are really close together so there may be little to pick between these top three models.

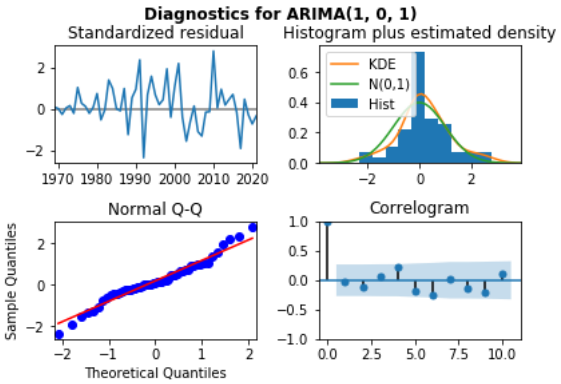
Building, plotting and reviewing three models

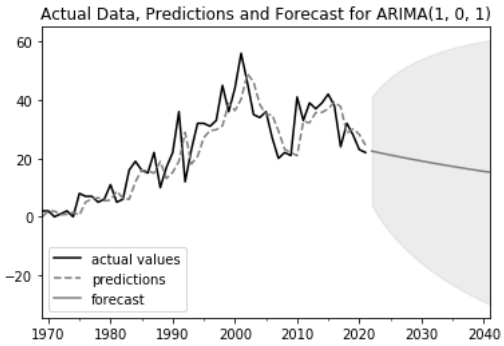
The models were build using the SARIMAX class of the statsmodels library. This was not strictly necessary as there is no seasonal data either in the actual values or after differencing. However, it doesn't hurt the model to simply leave out the parameters referring to seasonal orders and the models will need less tweaking in future if a seasonal trend in introduced over the next decades.

In the following the orders derived at from the for-loops are used, the best of each order of differencing. For the models using differenced data a constant trend was also included as a parameter since this data no longer follows the trend the actual data did.

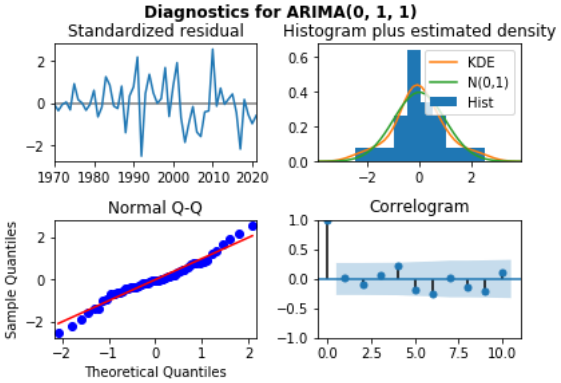
The following three double plots summarize the findings from each of models and are commented on in "A combined look at all three model predictions and forecasts" following the plots for each model.

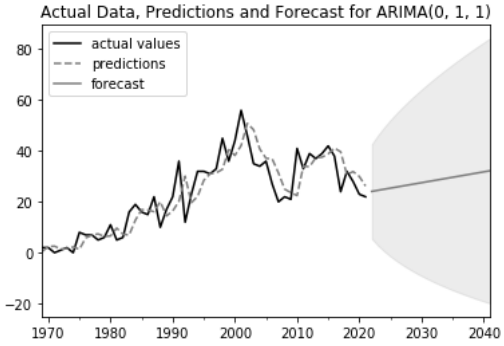
ARIMA(1, 0, 1)



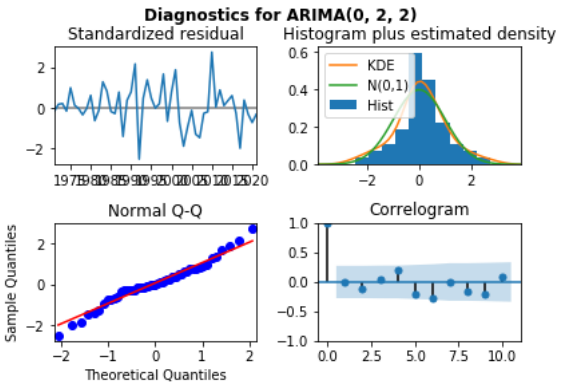


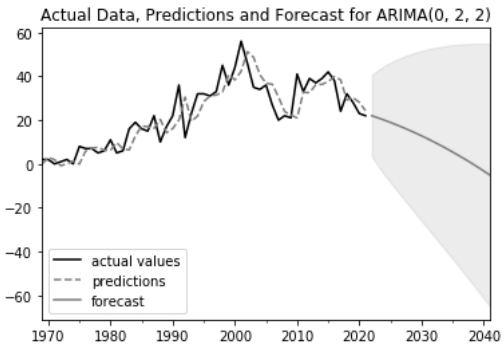
ARIMA(0, 1, 1)





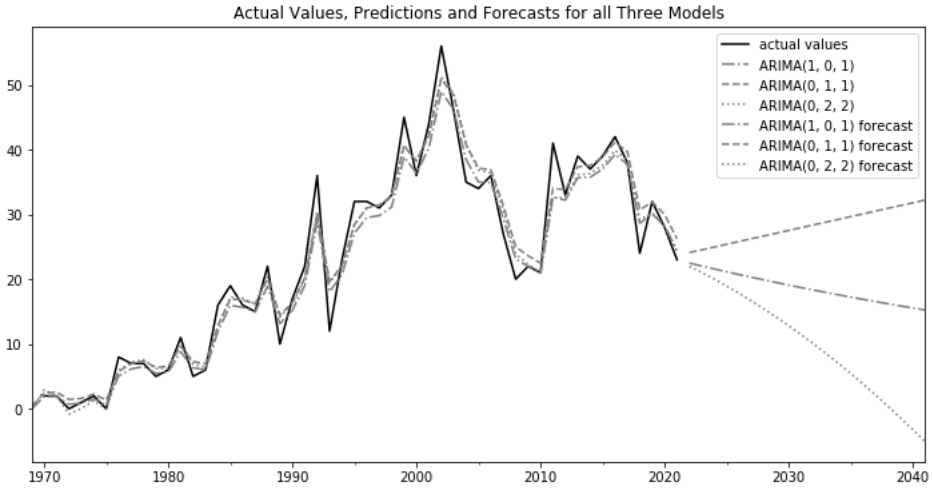
ARIMA(0, 2, 2)





A combined look at all three model predictions and forecasts

With all the models being quite similar except for the predictions it is rather difficult to see the differences between the models and their accurateness. It is almost impossible to find which model best fits the past values. Plotting all three models in one plots shows how very close they are to each other. Note: The actual data has been shifted by one year in this plot because taking the first difference also shifts the data by one year, by necessity (there can be no difference for the first value).



Neither of the three curves exactly follows the actual data, meaning that the risk that either is over-fitted is low. All more or less hit the lows and peaks albeit the low around 1993 was predicted as being less severe and the peak in 2001 was predicted as being smaller. In general though, the predictions follow the actual data well enough.

To judge the models it is necessary to turn to statistics. These are shown as plots on the left for each model. They deal with the residuals which are the differences between the actual and the predicted values for each point in time. These residuals should just represent white noise and there a guidelines for judging a good model.

The mean values of the residuals of the three models are:
ARIMA(1, 0, 1): `1.4019633978152939`
ARIMA(0, 1, 1): `0.015267265604687157`
ARIMA(0, 2, 2): `0.5457844560562354`
The residuals should be centred around 0, and the model with 1st order differencing comes really close.

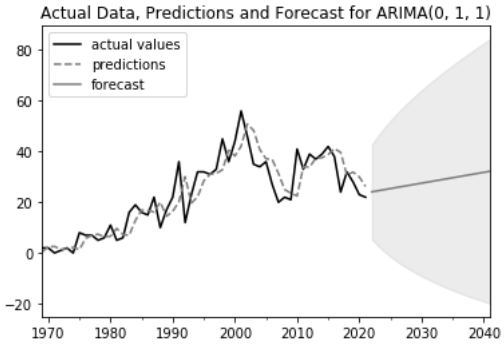
The standard deviation of the residuals of the three models are:
ARIMA(1, 0, 1): `7.145795477059154`
ARIMA(0, 1, 1): `7.233207178785556`
ARIMA(0, 2, 2): `7.272862621443911`
There isn't much to chose between them, but the model without differencing is clearly the worst. First order differencing performs slightly better than second order.

The residuals should also be normally distributed which is better seen in the plots above than adding more numbers. The model without differencing is clearly off here, with the orange bell curve in the histogram/kde plot being furthest off from the green one. The data points are also not following the straight line of normal distribution in the Q-Q plot but looks wavy. For the other two models there isn't so much to chose between them but the kde plot for the model using first differencing follows the normal distribution better than it does after differencing one more time. The deviations from the red line in the Q-Q plots are harder to judge. Neither is following the line entirely but then the models use real world data not perfectly simulated data.

The last rule of thumb for judging that the residuals are, indeed, white noise is looking at the autocorrelation plots. There should be no correlation over time, and there isn't as shown in the correlograms, for any of the models.

Best Model Recommendation

From the findings in the last section of this report it can be concluded that the model using first order differencing performs best. Its residuals are distributed around a mean of 0 and approach normal distribution the closest, which confirms that the residuals are mainly white noise. This model also forecasts a slight upward trend, confirming the overall upwards trend. This line is almost parallel to a imaginary straight line drawn across the actual data, waiving the peaks for the moment.



Summary

Starting from the database downloaded from rebrickable.com there was only a comprehensive list of which set was released when and which theme it belonged to. It also detailed how many parts each set was made up of. This was not ordered in any way and with 18,021 sets across 535 themes it was almost impossible to gain any insight into the data by just looking at the lines and lines of the list.

Doing some simple preliminary analysis found that the number of sets as well as the number of parts released during a year was growing fast, from just a few basic sets to over 800 in recent years. This analysis also found that the number of themes for which sets were released grew over the years, adding to the complexity of the data. There were times when less sets were released than in previous year but these don't correspond clearly to financial or social crises and can't be commented upon here.

The growth was more or less linear until the mid-nineties of the last century. After that growth could be considered exponential with almost steadily rising numbers of sets per year. In order to build any model from this the data needed to be transformed to take out the trend. For this data this was achieved by taking the differencing between adjoining values which is both the easiest way and part of the models used. It is also the method that loses the least data which is a serious consideration as the database only has yearly data and the loss of each data point might make the model less well able to perform.

Seeing as the main objective of this analysis was to provide some guidance on which sets and themes to buy or stock the data relating to "Duplo" sets was chosen for demonstration purposes because it is the longest-lived and as such has the most data points. The analysis could be replicated easily for any other theme of sets.

As expected from the linear overall growth merging into exponential growth the model using first order differencing and a constant trend performed best. It matched the actual data nicely without overfitting it and the forecast found that the number of sets will keep rising moderately following a decline which might be attributed to the current covid-19 crises. This rise also fits the overall trend if it was simply shown as a line following the overall development.

It is likely that any of the other longer-lived themes will follow the same trend but it is out of scope of this analysis to follow up on this assumption. It may be necessary to combine two related themes as was done earlier with "Town" and "City" where one was going down while the other was going up. Despite the different names they are essential the same theme so it would make sense to analyse them as one theme rather than separate ones.

Outlook

The goal of this paper was to provide some guidance for shopkeepers and (grand-) parents when choosing which sets or themes of sets to stock or buy and which are likely to be shelf warmers. Due to the limited scope of this paper the analysis should be repeated for other themes as well to offer a more comprehensive view. The analysis should also be repeated after a period of, say, five years. It seems not unlikely that a drought is coming up in the near future because children and enthusiastic adults have bought all the sets they could want for a while during the covid-19 pandemic and are also turning to other hobbies again when the restrictions are and stay lifted. It is not recommended to repeat the analysis every year based on this database because the trend won't be visible that quickly. On top of that only one data point per year is added and will not improve the models significantly.

It would also improve the models if monthly data could be obtained from the manufacturer rather than the yearly data compiled by rebrickable.com. Doing so would provide the models with twelve times more data points, thus increasing the ability to find patterns greatly. It might even turn out that the publication of sets is seasonal after all, with more sets being released in the weeks before Easter and Christmas and only a few in the warm summer months when children and adults alike are much more likely to be outside and use other toys/games.

Another way to improve forecasts would be to turn attention to the competitors and analyse which of their sets and themes are doing well and look likely to continue doing so. It may well be that they orientate themselves by what LEGO is showing an interest in but it may just as well happen that LEGO has missed a trend and might be following their competitors if their sets are doing well or even better than LEGO's own.