Understanding Lego sets popularity

Background

You recently applied to work as a data analyst intern at the famous Lego Group in Denmark. As part of the job interview process, you received the following take-home assignment:

You are asked to use the provided dataset to understand the popularity of different Lego sets and themes. The idea is to become familiarized with the data to be ready for an interview with a business stakeholder.

erd

The data

You received access to a database with the following tables. You can also see above a visualization of how the tables are related to each other. (source):

inventory_parts

- "inventory_id" - id of the inventory the part is in (as in the inventories table)
- "part_num" - unique id for the part (as in the parts table)
- "color_id" - id of the color
- "quantity" - the number of copies of the part included in the set
- "is_spare" - whether or not it is a spare part

parts

- "part_num" - unique id for the part (as in the inventory_parts table)
- "name" - name of the part
- "part_cat_id" - part category id (as in part_catagories table)

part_categories

- "id" - part category id (as in parts table)
- "name" - name of the category the part belongs to

colors

- "id" - id of the color (as in inventory_parts table)
- "name" - color name
- "rgb" - rgb code of the color
- "is_trans" - whether or not the part is transparent/translucent

inventories

- "id" - id of the inventory the part is in (as in the inventory_sets and inventory_parts tables)
- "version" - version number
- "set_num" - set number (as in sets table)

inventory_sets

- "inventory_id" - id of the inventory the part is in (as in the inventories table)
- "set_num" - set number (as in sets table)
- "quantity" - the quantity of sets included

sets

- "set_num" - unique set id (as in inventory_sets and inventories tables)
- "name" - the name of the set
- "year" - the year the set was published
- "theme_id" - the id of the theme the set belongs to (as in themes table)
- num-parts - the number of parts in the set

themes

- "id" - the id of the theme (as in the sets table)
- "name" - the name of the theme
- "parent_id" - the id of the larger theme, if there is one

__Acknowledgments__: Rebrickable.com

Challenge

Create a report to summarize your findings. Include:

1. What is the average number of Lego sets released per year?
2. What is the average number of Lego parts per year?
3. Create a visualization for item 2.
4. What are the 5 most popular colors used in Lego parts?
5. [Optional] What proportion of Lego parts are transparent?
6. [Optional] What are the 5 rarest lego bricks?
7. Summarize your findings.

Checklist before publishing

- Rename your workspace to make it descriptive of your work. N.B. you should leave the notebook name as notebook.ipynb.
- **Remove redundant cells** like the introduction to data science notebooks, so the workbook is focused on your story.
- Check that all the cells run without error.

Import modules and datasets

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

```python
themes = pd.read_csv('themes.csv')
sets = pd.read_csv('sets.csv')
parts = pd.read_csv('parts.csv')
part_categories = pd.read_csv('part_categories.csv')
inventory_sets = pd.read_csv('inventory_sets.csv')
inventory_parts = pd.read_csv('inventory_parts.csv')
inventories = pd.read_csv('inventories.csv')
colors = pd.read_csv('colors.csv')
```

Rename & Merge Columns

In [2]:
```python
# renaming individual dataframe columns for merging and differentiation
inventories.rename(columns={'id':'inventory_id'}, inplace=True)
inventory_sets.rename(columns={'quantity':'quantity_inv_sets'}, inplace=True)
sets.rename(columns={'name':'name_sets', 'num_parts':'num_parts_sets'}, inplace=True)
themes.rename(columns={'id':'theme_id', 'name':'name_themes'}, inplace=True)
inventory_parts.rename(columns={'quantity':'inv_parts_quantity'}, inplace=True)
colors.rename(columns={'id':'color_id', 'name':'name_colors'}, inplace=True)
part_categories.rename(columns={'id':'part_cat_id', 'name':'name_part_cat'}, inplace=Tru
```

In [3]:
```python
# merge inventory_sets with sets as df on common columns
df = pd.merge(inventory_sets, sets, on='set_num')
```

In [4]:
```python
# merge df with inventories via left side
df = df.merge(inventories, on=['inventory_id', 'set_num'], how='left')
```

In [5]:
```python
df = df.merge(themes, on='theme_id', how='left')
```

In [6]:
```python
df = df.merge(inventory_parts, on='inventory_id', how='left')
```

In [7]:
```python
parts.rename(columns={'name':'name_parts'}, inplace=True)

df = df.merge(parts, on='part_num')
```

In [8]:
```python
df = df.merge(part_categories, on='part_cat_id')
```

In [9]:
```python
df = df.merge(colors, on='color_id')
```

In [10]:
```python
# check first 2 rows of df
df.head(2)
```

Out[10]:

	inventory_id	set_num	quantity_inv_sets	name_sets	year	theme_id	num_parts_sets	
0	311	8593-1	1	Makuta	2003	324	199	https://cdn.rebrickable.com/
1	311	8596-1	1	Takanuva	2003	324	199	https://cdn.rebrickable.com/

In [11]:
```python
# drop irrelevant columns
df.drop(['img_url', 'version', 'is_spare', 'part_material', 'rgb', 'parent_id'], axis=1,
# reset index
df.reset_index(drop=True, inplace=True)
```

In [12]:
```
# inspect for duplicate rows
df.duplicated().sum()
```

Out[12]:
```
0
```

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 725 entries, 0 to 724
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   inventory_id         725 non-null    int64  
 1   set_num              725 non-null    object 
 2   quantity_inv_sets    725 non-null    int64  
 3   name_sets            725 non-null    object 
 4   year                 725 non-null    int64  
 5   theme_id             725 non-null    int64  
 6   num_parts_sets       725 non-null    int64  
 7   name_themes          725 non-null    object 
 8   part_num             725 non-null    object 
 9   color_id             725 non-null    float64
 10  inv_parts_quantity   725 non-null    float64
 11  name_parts           725 non-null    object 
 12  part_cat_id          725 non-null    int64  
 13  name_part_cat        725 non-null    object 
 14  name_colors          725 non-null    object 
 15  is_trans             725 non-null    object 
dtypes: float64(2), int64(6), object(8)
memory usage: 90.8+ KB
```

- df has a total of 725 rows (including headers), and 16 columns
- There appears to be no missing values
- most of the 'int64' columns may need to be converted into string for feature engineering

In [15]:
```
# confirm null values
df.isnull().any()
```

Out[15]:
```
inventory_id          False
set_num               False
quantity_inv_sets     False
name_sets             False
year                  False
theme_id              False
num_parts_sets        False
name_themes           False
part_num              False
color_id              False
inv_parts_quantity    False
name_parts            False
part_cat_id           False
name_part_cat         False
name_colors           False
is_trans              False
dtype: bool
```

In [16]:
```
# df.to_excel('mergedlego.xlsx')
```

Exploratory Data Analysis

In [17]:
```python
# view statistics of df
df.describe(include='all')
```

Out[17]:

	inventory_id	set_num	quantity_inv_sets	name_sets	year	theme_id	num_parts_sets	name_the
count	725.000000	725	725.000000	725	725.000000	725.000000	725.000000	
unique	NaN	167	NaN	166	NaN	NaN	NaN	
top	NaN	8580-1	NaN	Kraata	NaN	NaN	NaN	Star '
freq	NaN	76	NaN	76	NaN	NaN	NaN	
mean	29691.263448	NaN	1.135172	NaN	2002.754483	374.782069	126.787586	
std	37415.223658	NaN	0.346157	NaN	9.255303	172.163434	192.856806	
min	311.000000	NaN	1.000000	NaN	1969.000000	18.000000	0.000000	
25%	9820.000000	NaN	1.000000	NaN	1998.000000	236.000000	14.000000	
50%	11554.000000	NaN	1.000000	NaN	2003.000000	324.000000	59.000000	
75%	24973.000000	NaN	1.000000	NaN	2009.000000	532.000000	170.000000	
max	136912.000000	NaN	3.000000	NaN	2022.000000	744.000000	1412.000000	

- The top sets name is Kraata appearing x76 ('name_sets' - the name of the set)
- There are 56 unique theme names, with the top themes name; Star Wars appearing x113 ('name_themes' - the name of the theme)
- There are a total of 167 unique set ids ('set_num' - unique set id)
- 166 unique set names ('name_sets' - the name of the set)
- 56 unique theme names ('name_themes' - the name of the theme)
- 337 unique id for parts ('part_num' - unique id for the part)
- 336 unique names of parts ('name_part' - name of the part)
- there are 45 types of colors

In [18]:
```python
# convert int values to string
df[['inventory_id', 'theme_id', 'color_id', 'part_cat_id']].apply(str)
```

Out[18]:
```
inventory_id     0         311\n1         311\n2       16117\n3      ...
theme_id         0       324\n1       324\n2       324\n3       324...
color_id         0         0.0\n1         0.0\n2         0.0\n3      ...
part_cat_id      0       41\n1       41\n2       41\n3       41\n4 ...
dtype: object
```

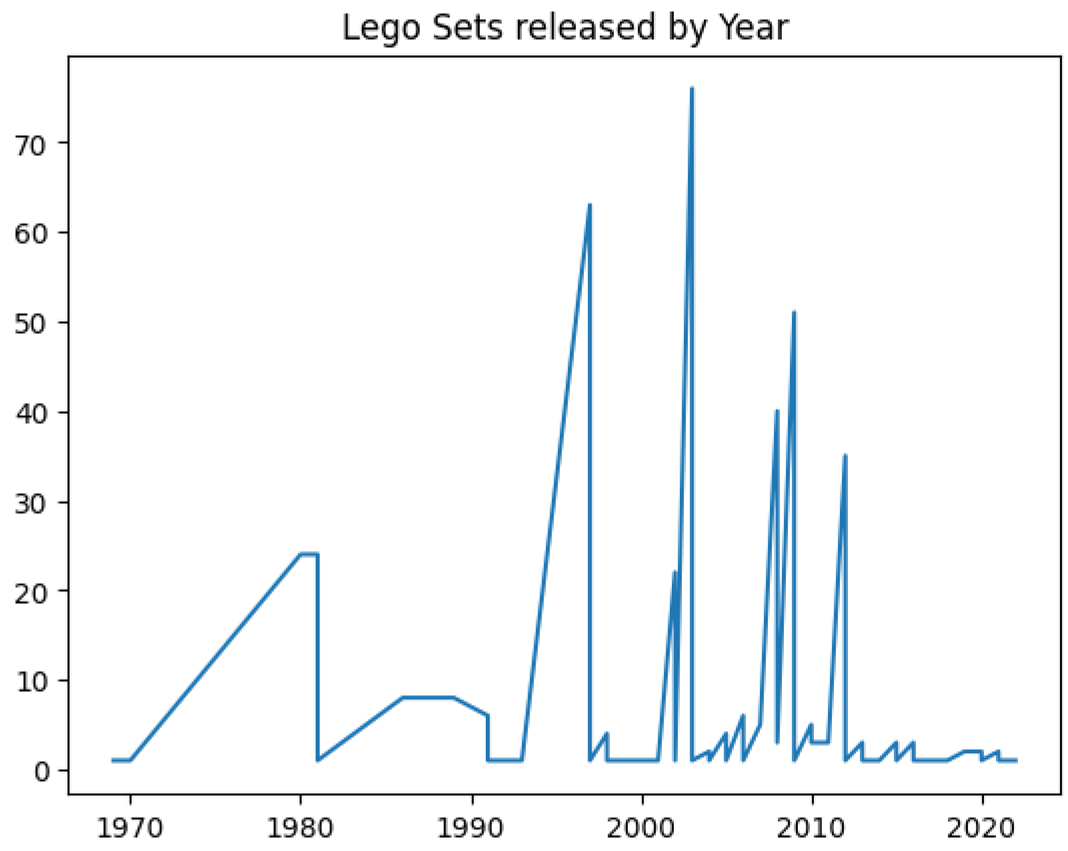
Average total of Lego Sets released per Year

In [39]:
```python
# create dataframe groupedby 'year' with 'set_num' value counts
year_sets = df.groupby(['year'], as_index=False)['set_num'].value_counts()

# calculate the average
avg_year_sets = round(sum(year_sets['count'])/len(year_sets['year']), 0)
```

```
print('The average total of Lego sets released per year is:', avg_year_sets)
```

```
The average total of Lego sets released per year is: 4.0
```

In [41]:
```
# plot total of lego sets released per year
plt.plot(year_sets['year'], year_sets['count'])
plt.title('Lego Sets released by Year')
plt.show()
```



In [61]:
```
# view highest 5 sets released count
year_sets.sort_values(by='count', ascending=False)[:5]
```

Out[61]:

	year	set_num	count
54	2003	8580-1	76
16	1997	9847-1	63
118	2009	20009-1	51
113	2008	20004-1	40
128	2012	5000062-1	35

In [63]:
```
#view smallest 3 sets released counts
year_sets.nsmallest(3, ['count'])
```

Out[63]:

	year	set_num	count
0	1969	344-1	1
1	1969	346-2	1
2	1970	603-3	1

- 1997 and 2003 recorded the highest amount of Lego sets released
- The trend has remained below 10 sets since 2012

Average Total of Lego Parts released per Year

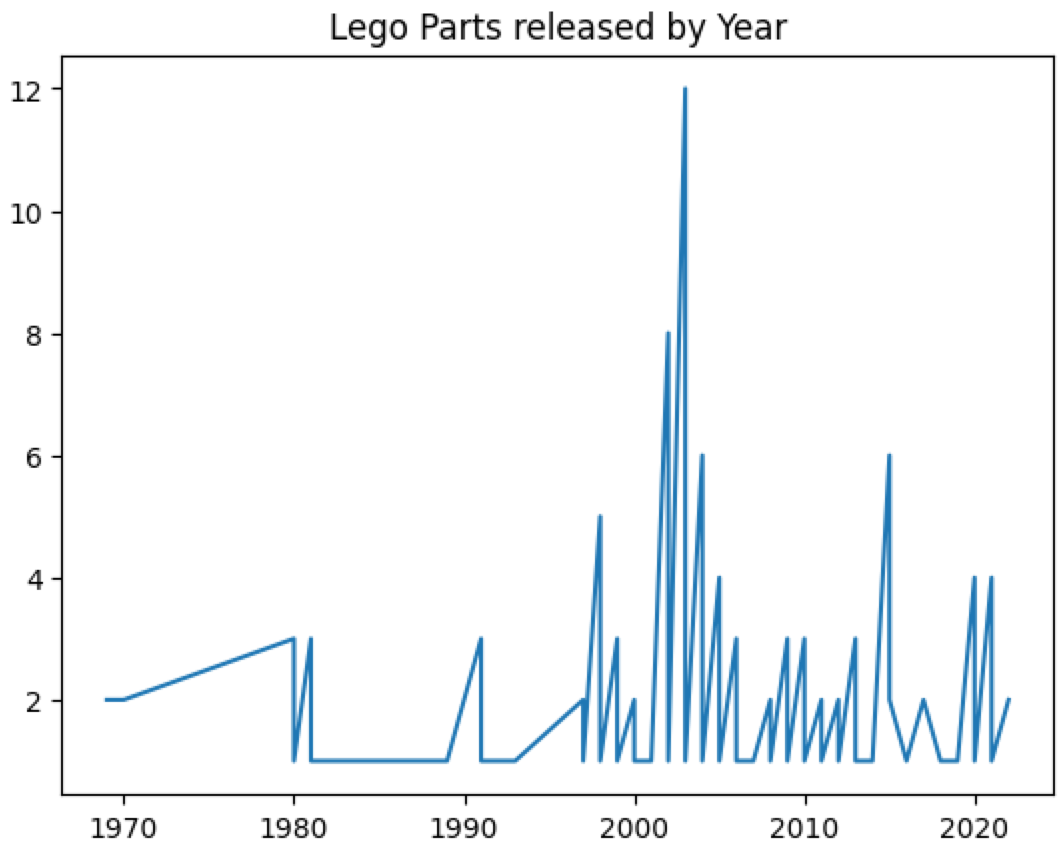
In [36]:

```
# create dataframe groupedby 'year' and 'part_num' value counts
year_parts = df.groupby(['year'], as_index=False)['part_num'].value_counts()

# calculate average
avg_year_parts = round(sum(year_parts['count'])/len(year_parts['part_num']), 0)

print('The average total of Lego parts released per year is:', avg_year_parts)
```

```
The average total of Lego parts released per year is: 2.0
```

In [42]:

```
# plot total Lego parts released per year
plt.plot(year_parts['year'], year_parts['count'])
plt.title('Lego Parts released by Year')
plt.show()
```



In [59]:

```
# view top 5 parts released count
year_parts.sort_values(by='count', ascending=False)[:5]
```

Out[59]:

	year	part_num	count
176	2003	43559	12
161	2002	74747	8
169	2002	70931	8
167	2002	6035	8
166	2002	5306c01	8

```
In [64]: # view smallest 3 parts released count
         year_parts.nsmallest(3, ['count'])
```

Out[64]:

	year	part_num	count
4	1980	3623	1
5	1980	3633	1
6	1980	3849	1

- 2003 recorded the highest amount of Lego parts released, peaking at 12
- Lego parts release count has remained below 7 after the peak, until 2020

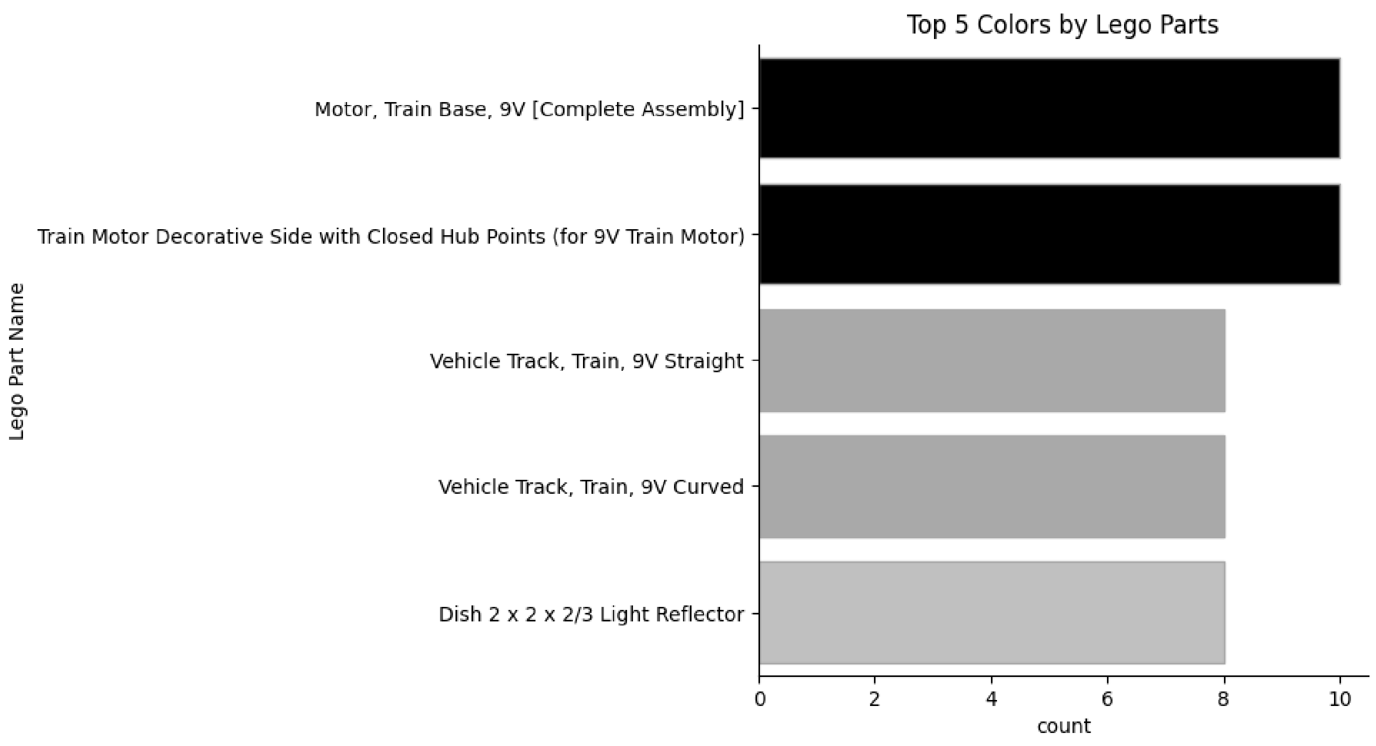
5 Most Popular Colors used in Lego Parts

```
In [23]: # create groupby dataframe with name_parts and Value_counts of colors
         part_color = df.groupby(['name_parts'], as_index=False)['name_colors'].value_counts().so
         part_color
```

Out[23]:

	name_parts	name_colors	count
165	Motor, Train Base, 9V [Complete Assembly]	Black	10
377	Train Motor Decorative Side with Closed Hub Po...	Black	10
384	Vehicle Track, Train, 9V Straight	Dark Gray	8
382	Vehicle Track, Train, 9V Curved	Dark Gray	8
65	Dish 2 x 2 x 2/3 Light Reflector	Chrome Silver	8

```
In [52]: plt.figure(figsize=(12,7))
         g = sns.catplot(part_color, y='name_parts', x='count', kind='bar',
                         palette=['black', 'black', 'darkgrey', 'darkgrey', 'silver'],
                        edgecolor='darkgrey')

         plt.title('Top 5 Colors by Lego Parts')
         plt.ylabel('Lego Part Name')
         plt.show()

         <Figure size 1200x700 with 0 Axes>
```



The most popular Colors Overall

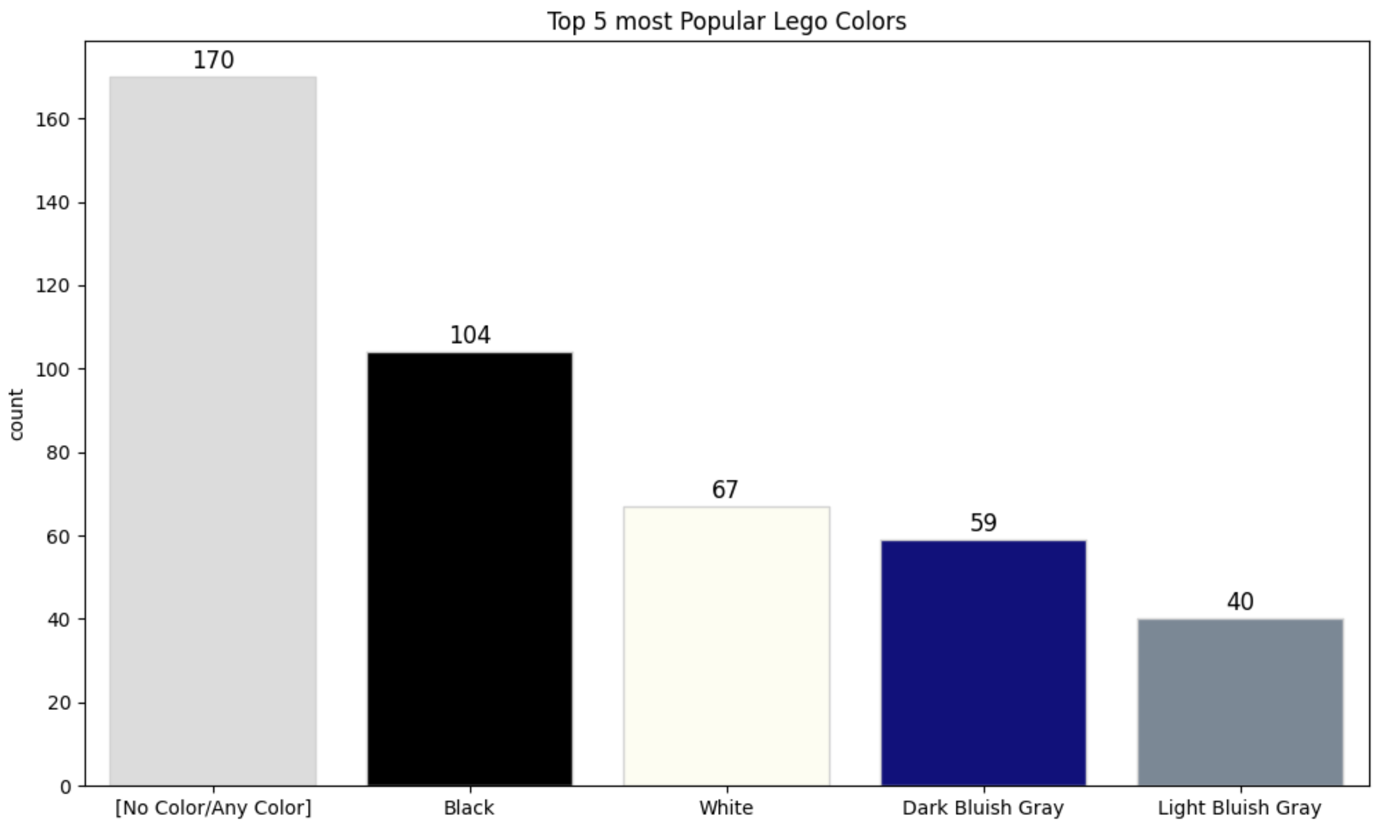
In [25]:
```
bricks = df.groupby(['name_colors'], as_index=False)['name_colors'].value_counts()
```

In [26]:
```
pop_bricks = bricks.nlargest(5, ['count'])
pop_bricks
```

Out[26]:

	name_colors	count
43	[No Color/Any Color]	170
0	Black	104
41	White	67
6	Dark Bluish Gray	59
14	Light Bluish Gray	40

In [27]:
```
# create plot figure
plt.figure(figsize=(12,7))
plots = sns.barplot(pop_bricks, x='name_colors', y='count', palette=['gainsboro', 'black

# annotate bars
for bar in plots.patches:
    plots.annotate(format(bar.get_height(), '.0f'),
                   (bar.get_x() + bar.get_width() / 2,
                    bar.get_height()), ha='center', va='center',
                   size=12, xytext=(0, 8),
                   textcoords='offset points')

plt.title('Top 5 most Popular Lego Colors')
plt.xlabel('')
plt.show()
```



What proportion of Lego is transparent?

```
In [28]: # value counts of is_trans
         df.is_trans.value_counts()
```

```
Out[28]: f    703
         t     22
         Name: is_trans, dtype: int64
```

```
In [29]: # create dataframe with 'is_trans' value counts
         trans_lego = df.groupby(['is_trans'], as_index=False)['is_trans'].value_counts()
```

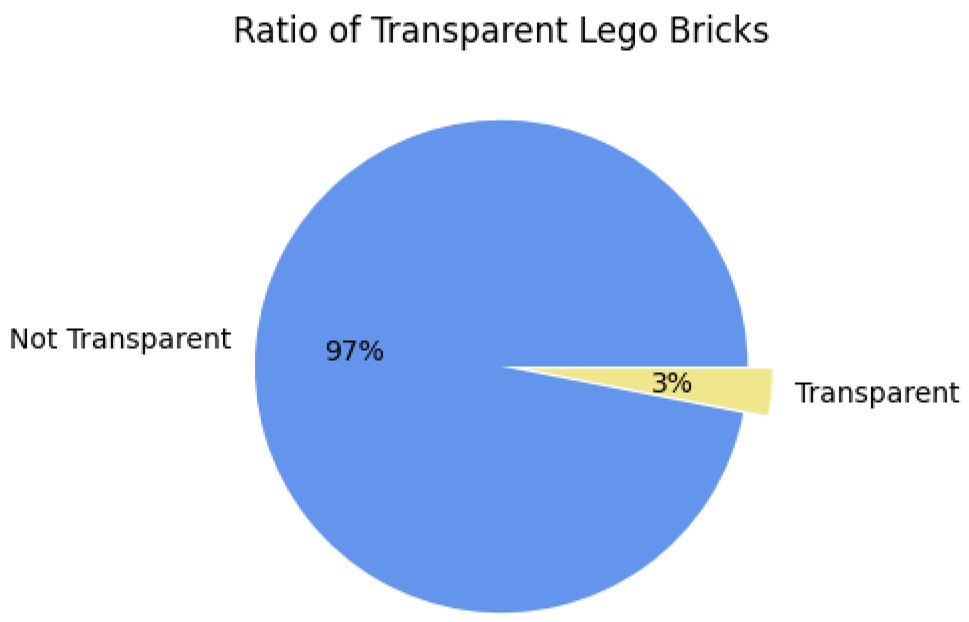
```
In [30]: # add percent column
         trans_lego['percent'] = (trans_lego['count']/sum(trans_lego['count'])*100).round(1)
```

```
In [31]: # rename values
         trans_lego['is_trans'] = trans_lego.is_trans.replace(['f', 't'], ['Not Transparent', 'Tr
         trans_lego
```

Out[31]:

	is_trans	count	percent
0	Not Transparent	703	97.0
1	Transparent	22	3.0

```
In [32]: # create pie plot of transarent proportions
         plt.figure(figsize=(4,4))
         plt.pie(x= trans_lego['count'], labels=trans_lego['is_trans'],
                autopct='%.0f%%', explode=[0,0.1], colors=['cornflowerblue', 'khaki'])
         plt.title('Ratio of Transparent Lego Bricks')
         plt.show()
```



- The proportion of transparent Lego is 3%

What are the 5 Rarest Lego Bricks

The 5 rarest Lego Bricks by Lego Parts

In [33]:
```
# create dataframe of name_parts of Lego bricks count
brick_names = df.groupby(['name_parts'], as_index=False)['name_parts'].value_counts()

# add percent column
brick_names['percent'] = (brick_names['count']/sum(brick_names['count'])*100).round(2)

# select the lowest 5 values
brick_names.sort_values(by='percent', ascending=True)[:5]
```

Out[33]:

	name_parts	count	percent
0	Activity Book, EV3 Space Challenge	1	0.14
200	Slope 10° 6 x 8	1	0.14
201	Slope 18° 4 x 1	1	0.14
202	Slope 18° 4 x 2	1	0.14
205	Slope 45° 2 x 1 Triple with Ovoid Bottom Pin	1	0.14

The 5 rarest Lego Bricks by Theme Name

In [54]:
```
rare_theme_bricks = df.groupby(['name_themes'], as_index=False)['name_themes'].value_cou
rare_theme_bricks
```

Out[54]:

	name_themes	count
27	Ninja	1
53	Western	1

50	Vehicle	1
49	UFO	1
36	Role Play Toys and Costumes	1

The 5 rarest Lego Bricks by Color

In [34]:
```
# select the 5 lowest count of Lego brick color
rarest_bricks = bricks.nsmallest(5, ['count'])
rarest_bricks
```

Out[34]:

	name_colors	count
3	Brown	1
7	Dark Brown	1
12	Glow In Dark Opaque	1
18	Metallic Gold	1
21	Pearl Blue	1

Summary of Analysis

Average Total of Lego Sets released by Year

- There are a total of 167 unique set_num and 166 unique name_sets, which suggests that there is a common name_set shared between 2 set_num
- The average total of sets released between 1970 and 2020 is 4 sets per year.
- 2003 marked the highest sets released with a 76 count of set_num 8580-1, followed by year 1997 with a 63 count of set_num 9847-1.
- The set_num released has remained below count of 10 since 2012 until 2020

Average Total of Lego Parts released by Year

- The average number of Lego parts released per year is 2 parts.
- 2003 recorded the highest amount of Lego parts released of 12 parts for part_num 43559
- The next highest parts released occurred in 2003, counting 8 parts for part_num 74747, 70931, 6035 and 5306c01
- The trendline indicates a volatile release of parts throughout 2005 and 2020, consistent to remaining under 7 parts.
- The lowest parts released occurred in 1980

Most Popular Lego Parts by Color

- The most popular Lego parts by color in 'Black' at 10 counts are for parts; Motor, Train Base, 9V [Complete Assembly], and Train Motor Decorative Side with Closed Hub Points.
- Followed by ‘Dark Gray’ for both Vehicle Track, Train, 9V Straight and Curved, at 8 counts.
- ‘Chrome Silver’ for the part Dish 2 x 2 x 2/3 Light Reflector at 8 counts.

Proportion of Transparent Lego

- The proportion of transparent Lego is 3%, indicating non-tranparent are far more popular

Rarest Lego Bricks

- The rarest Leo bricks by parts are Activity Book, EV3 Space Challenge, and the Slope ranges Slope 10° 6 x 8, Slope 18° 4 x 1, Slope 18° 4 x 2 and Slope 45° 2 x 1 Triple with Ovoid Bottom Pin
- The rarest Lego bricks by theme name are Ninja, Western, Vehicle, UFO and Role Play Toys and Costumes, at 1 count.
- The rarest Lego bricks by color are Brown, Dark Brown, Glow in the Dark Opaque, Metallic Gold and Pearl Blue.

```
In [ ]:
```