Part A - Lego Analysis

Analysis of Lego

```
# Loading libraries

library(tidyverse)
library(tidytuesdayR)
library(ggplot2)
library(scales)
```

Reading in dataset

```
# Read data from tidytuesday
tuesdata = tidytuesdayR::tt_load('2022-09-06')

for (name in names(tuesdata))
{var_name = gsub("\\.csv$", "", name)
  assign(var_name, tuesdata[[name]])}

rm(tuesdata)
```

Explore data

```
head(colors)
```

```
# A tibble: 6 x 4
     id name          rgb     is_trans
  <dbl> <chr>         <chr>   <lgl>
1    -1 [Unknown]     0033B2  FALSE
2     0 Black         05131D  FALSE
3     1 Blue          0055BF  FALSE
```

```
4     2 Green          237841 FALSE
5     3 Dark Turquoise 008F9B FALSE
6     4 Red            C91A09 FALSE
```

```
head(elements)
```

```
# A tibble: 6 x 3
  element_id part_num color_id
       <dbl> <chr>       <dbl>
1    6300211 67906c01       14
2    4566309 2564            0
3    4275423 53657        1004
4    6194308 92926          71
5    6229123 26561           4
6    4241969 51035        1004
```

```
head(inventories)
```

```
# A tibble: 6 x 3
     id version set_num
  <dbl>   <dbl> <chr>
1     1       1 7922-1
2     3       1 3931-1
3     4       1 6942-1
4    15       1 5158-1
5    16       1 903-1
6    17       1 850950-1
```

```
head(inventory_minifigs)
```

```
# A tibble: 6 x 3
  inventory_id fig_num     quantity
         <dbl> <chr>          <dbl>
1            3 fig-001549         1
2            4 fig-000764         1
3           19 fig-000555         1
4           25 fig-000574         1
5           26 fig-000842         1
6           26 fig-008641         1
```

```
head(inventory_parts)
```

```
# A tibble: 6 x 6
  inventory_id part_num       color_id quantity is_spare img_url
         <dbl> <chr>             <dbl>    <dbl> <lgl>    <chr>
1            1 48379c01             72        1 FALSE    https://cdn.rebrickabl~
2            1 48395                 7        1 FALSE    https://cdn.rebrickabl~
3            1 stickerupn0077     9999        1 FALSE    <NA>
4            1 upn0342               0        1 FALSE    <NA>
5            1 upn0350              25        1 FALSE    <NA>
6            3 2343                 47        1 FALSE    https://cdn.rebrickabl~
```

```
head(inventory_sets)
```

```
# A tibble: 6 x 3
  inventory_id set_num quantity
         <dbl> <chr>      <dbl>
1           35 75911-1        1
2           35 75912-1        1
3           39 75048-1        1
4           39 75053-1        1
5           50 4515-1         1
6           50 4520-1         2
```

```
head(minifigs)
```

```
# A tibble: 6 x 4
  fig_num    name                                     num_parts img_url
  <chr>      <chr>                                        <dbl> <chr>
1 fig-000001 Toy Store Employee                               4 https://cdn.reb~
2 fig-000002 Customer Kid                                     4 https://cdn.reb~
3 fig-000003 Assassin Droid, White                            8 https://cdn.reb~
4 fig-000004 Man, White Torso, Black Legs, Brown Hair         4 https://cdn.reb~
5 fig-000005 Captain America with Short Legs                  3 https://cdn.reb~
6 fig-000006 Lloyd Avatar                                     5 https://cdn.reb~
```

```
head(part_categories)
```

```
# A tibble: 6 x 2
     id name
  <dbl> <chr>
1     1 Baseplates
2     3 Bricks Sloped
3     4 Duplo, Quatro and Primo
4     5 Bricks Special
5     6 Bricks Wedged
6     7 Containers
```

head(part_relationships)

```
# A tibble: 6 x 3
  rel_type child_part_num parent_part_num
  <chr>    <chr>          <chr>
1 P        3626cpr3662    3626c
2 P        87079pr9974    87079
3 P        3960pr9971     3960
4 R        98653pr0003    98086pr0003
5 R        98653pr0003    98088pat0003
6 R        98653pr0003    98089pat0003
```

head(parts)

```
# A tibble: 6 x 4
  part_num name                                 part_cat_id part_material
  <chr>    <chr>                                      <dbl> <chr>
1 003381   Sticker Sheet for Set 663-1                   58 Plastic
2 003383   Sticker Sheet for Sets 618-1, 628-2           58 Plastic
3 003402   Sticker Sheet for Sets 310-3, 311-1, 312-3    58 Plastic
4 003429   Sticker Sheet for Set 1550-1                  58 Plastic
5 003432   Sticker Sheet for Sets 357-1, 355-1, 940-1    58 Plastic
6 003434   Sticker Sheet for Set 575-2, 653-1, 460-1     58 Plastic
```

head(sets)

```
# A tibble: 6 x 6
  set_num name          year theme_id num_parts img_url
  <chr>   <chr>        <dbl>    <dbl>     <dbl> <chr>
1 001-1   Gears         1965        1        43 https://cdn.rebri~
```

```
2 0011-2   Town Mini-Figures        1979       67         12 https://cdn.rebri~
3 0011-3   Castle 2 for 1 Bonus Offer  1987     199          0 https://cdn.rebri~
4 0012-1   Space Mini-Figures       1979      143         12 https://cdn.rebri~
5 0013-1   Space Mini-Figures       1979      143         12 https://cdn.rebri~
6 0014-1   Space Mini-Figures       1979      143         12 https://cdn.rebri~
```

```
head(themes)
```

```
# A tibble: 6 x 3
      id name            parent_id
   <dbl> <chr>               <dbl>
1      1 Technic                NA
2      3 Competition             1
3      4 Expert Builder          1
4     16 RoboRiders              1
5     17 Speed Slammers          1
6     18 Star Wars               1
```

Considering the popularity of Lego, which is the most popular and the least popular themes
of Lego and how many parts did they have median

Methodology:

Many sets did not have a number of parts and were either 0 or NA, which affected the data
analysis and data manipulation. What we've done is filtered only the ones that we know of
how many parts as if they are not documented then there is no point in analysing it.

```
# Joining sets with themes and filtering out the sets with no parts due to error in data col]
set_themes = sets |>
  filter(!is.na(num_parts) & num_parts >0) |>
  left_join(themes |>
            select(id, theme_name = name),
            by = c(theme_id = "id"))
```
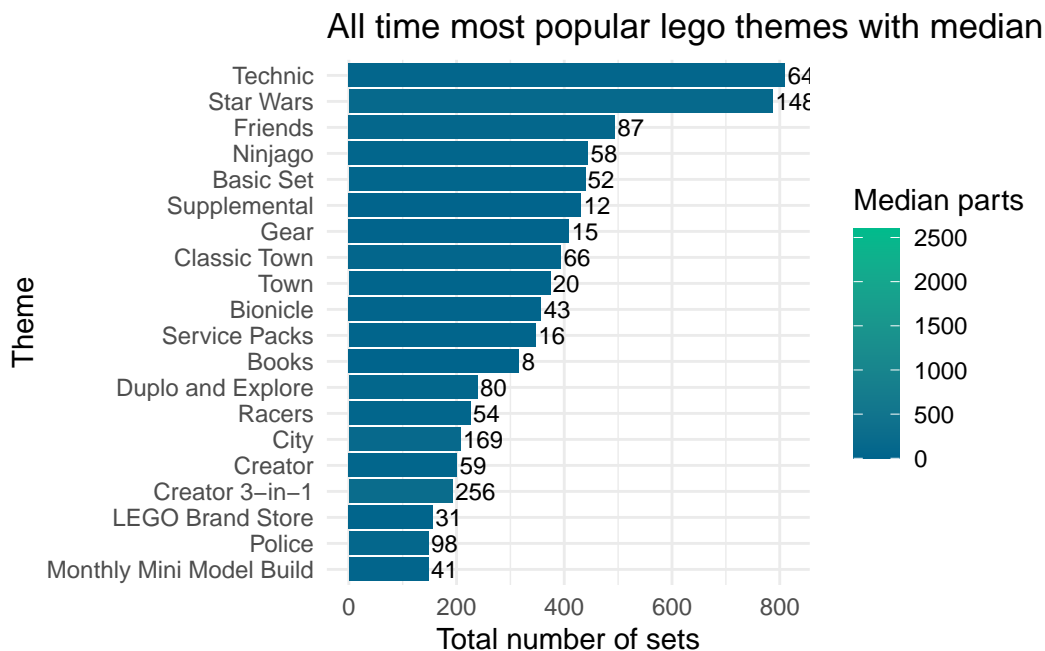
```
# Finding out the most popular themes
most_popular = set_themes |>
  group_by(theme_name) |>
  summarize(total_sets = n(),
            median_parts = round(median(num_parts, na.rm = TRUE)),
            .groups = "drop") |>
  arrange(desc(total_sets)) |>
  head(20) |>
```

```
ggplot(aes(x = total_sets, y = reorder(theme_name, total_sets))) +
  geom_bar(stat = "identity", aes(fill = median_parts)) +
  geom_text(aes(label = median_parts,
                x = total_sets + 5),
            hjust = 0, size = 3.2) +
  scale_fill_gradient(low = "#00648C", high = "#00BC8C", limits = c(0, 2600)) +
  labs(fill = "Median parts",
       title = "All time most popular lego themes with median parts",
       x = "Total number of sets",
       y = "Theme") +
  theme_minimal()

most_popular
```


All time most popular lego themes with median

```
# Finding out the least popular themes
least_popular = set_themes |>
  group_by(theme_name) |>
  summarize(total_sets = n(),
            median_parts = round(median(num_parts, na.rm = TRUE)),
            .groups = "drop") |>
  arrange(desc(total_sets)) |>
  tail(20) |>
```
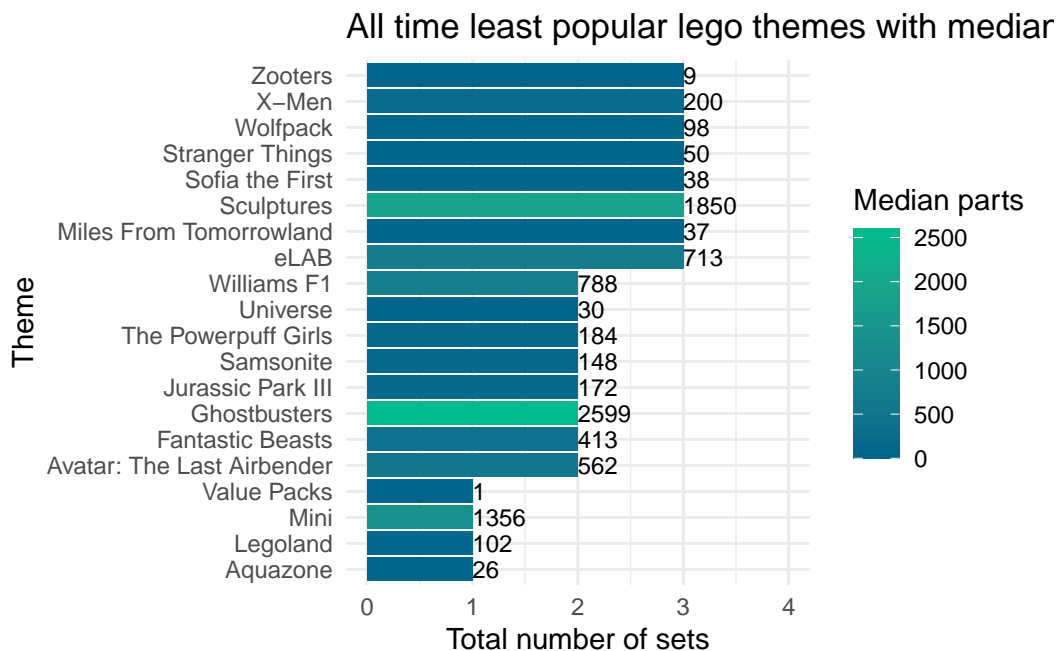
```r
ggplot(aes(x = total_sets, y = reorder(theme_name, total_sets))) +
geom_bar(stat = "identity", aes(fill = median_parts)) +
geom_text(aes(label = median_parts),
          hjust = 0, size = 3.2) +
scale_fill_gradient(low = "#00648C", high = "#00BC8C", limits = c(0, 2600)) +
labs(fill = "Median parts",
     title = "All time least popular lego themes with median parts",
     x = "Total number of sets",
     y = "Theme")+
xlim(0,4)+
theme_minimal()

least_popular
```


Most Popular LEGO Themes

1. Technic ranks as the most popular theme by total number of sets (~850) and features a high median part count of 846, indicating complex, advanced builds.

2. Star Wars follows with nearly 700 sets and a median of 648 parts, combining broad fan appeal with high build complexity.

3. Other popular themes like Friends, Ninjago, and Classic Town have moderate median part counts (50–80), suggesting simpler, more accessible builds for younger audiences.

4. While Gear and Basic Set were popular in total count in earlier analyses, they show much lower median parts, indicating smaller, simpler sets.

Least Popular LEGO Themes

1. Themes like Zooters, X-Men, Wolfpack, and Stranger Things each have fewer than 4 sets, clearly making them least popular.

2. Surprisingly, some of these rare themes have very high median part counts, such as:

   - Ghostbusters (2599 parts)

   - Sculptures (1850 parts)

   - Fantastic Beasts (413 parts)

3. These themes likely target niche or adult builders, offering large, detailed sets but lacking broad market traction.

4. Themes like Mini (1356 sets) are high in count but low in median parts, suggesting mass-produced, small builds.

Conclusion

Popular themes balance set quantity and moderate complexity to reach wide audiences. Unpopular themes may offer greater complexity, but limited releases or niche targeting reduce their market impact. LEGO's strategy spans both mass-market series and collector-grade sets, reflecting its diverse user base and flexible product design.

```
theme_name_summary = set_themes |>
  group_by(theme_name) |>
  summarize(n_sets = n(),
            median_parts = median(num_parts)) |>
  arrange(desc(n_sets)) |>
  filter(n_sets >= 100)

theme_name_summary
```

```
# A tibble: 30 x 3
   theme_name    n_sets median_parts
   <chr>          <int>        <dbl>
 1 Technic          810         64.5
 2 Star Wars        787        148
 3 Friends          494         87
 4 Ninjago          444         58
 5 Basic Set        439         52
 6 Supplemental     431         12
 7 Gear             409         15
 8 Classic Town     394         66.5
 9 Town             374         20
10 Bionicle         357         43
# i 20 more rows
```

Seeing the trends of Lego sets over the years, how complex have the Lego sets become.

```
part_year = set_themes |>
  group_by(year) |>
  summarise(median_parts = median(num_parts, na.rm = TRUE), .groups = "drop") |>
  ggplot(aes(x = year, y = median_parts)) +
  geom_line(color = "#00BC8C", linewidth = 1) +
  geom_point(color = "black") +
  geom_smooth(method = "loess", se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Median number of parts over the years",
       x = "Year",
       y = "Median parts per set")+
  theme_minimal()

part_year
```

```
`geom_smooth()` using formula = 'y ~ x'
```

Median number of parts over the years
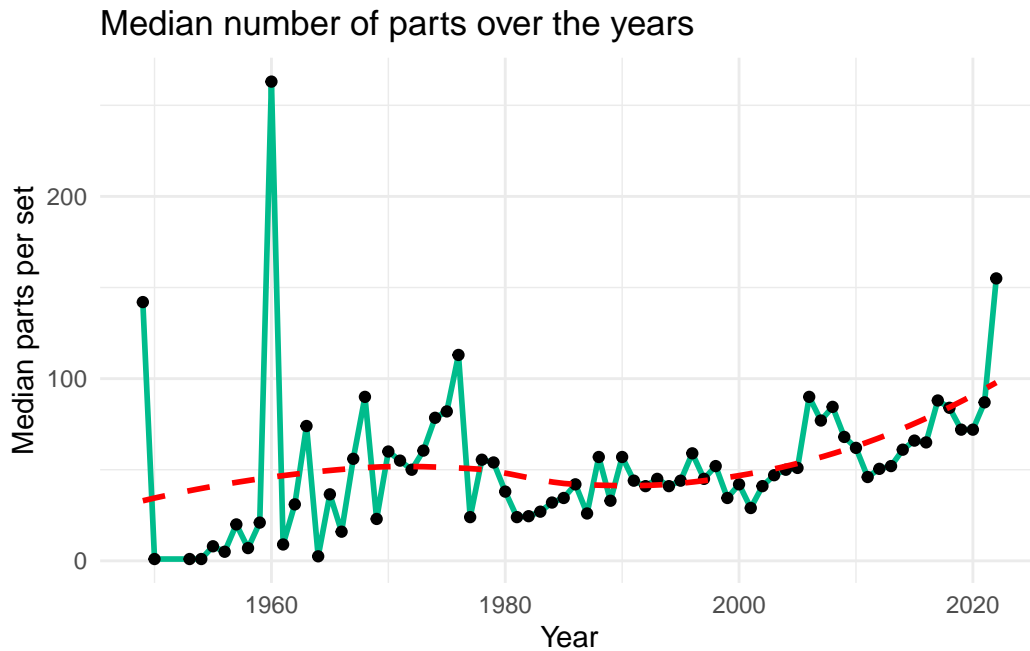
Which sets have the most parts

Analysis: Complexity of LEGO Sets Over the Years

1. Early Years (1950s–1960s): Irregular and Experimental

   The 1950s and early 1960s show sharp fluctuations in median part counts, including extreme spikes (e.g., one year with over 250 parts). This likely reflects LEGO's early experimentation with set design before standardizing product lines.

2. 1970s–1990s: Gradual Stabilization

   During these decades, the median number of parts remained relatively stable, averaging between 50 and 100 parts per set. This suggests LEGO was producing simpler sets aimed at children, focusing more on quantity and playability than complexity.

3. 2000s–2010s: Moderate Increase in Complexity

   A slight upward trend appears in the 2000s, with median parts slowly rising year over year. This reflects LEGO's expansion into more diverse themes (e.g., Star Wars, Technic) and the introduction of more detailed sets, appealing to both children and adult fans.

4. 2020–2023: Sharp Rise in Complexity

In the most recent years, the median number of parts has increased significantly, exceeding 150 parts per set in 2023. This may be due to the growing popularity of large-scale sets, modular buildings, and collector-focused themes.

5. Trendline Interpretation

The red dashed trendline confirms a clear long-term upward trajectory in set complexity. Despite some fluctuations, the overall pattern shows LEGO sets have consistently become more complex over the decades.

Conclusion

LEGO sets have evolved from basic, simple designs into intricate and detail-rich models. The increasing median part count, especially after 2010, demonstrates a deliberate shift toward higher complexity, aligning with the rise of adult fans and the collector market.