Week 5 - Project Submission

Step 1 - Find a Dataset

For the purposes of this project, I am going to use Amazon Toys Review data.

Trying to understand Amazon reviews for toys is one of the main reasons I chose this course. During course lectures, we used Amazon Gift card data which has a similar structure. However the Toys data is more complex and larger with millions of reviews and thousands of products, and I conduct analysis that are significantly different than what was covered in course lectures. I will also use Pandas (instead of numpy) in many cases which is another key difference with what was covered in the lectures

Step 2 - Explore the Datsset (transform for analysis, clean etc.)

The data is in a tsv file in a gz zipped format. The data has quotes ("s) in the reviews itself. So, needed to apply a QUOTE_NONE option in the csv.reader function. Also, utf8 encoding needs to be specifically mentioned - else there are errors.

In [1]:
```
#Reading and Preprocessing the data
import gzip
import csv
import time
path = "/Users/prathap/Documents/Datasets/amazon_reviews_us_Toys_v1_00.t
sv.gz"
f = gzip.open(path, 'rt', encoding = 'utf8')
reader = csv.reader(f, delimiter = '\t', quoting = csv.QUOTE_NONE)
#To create keys with variable names, store them in "header"
header = next(reader)
dataset = []
for line in reader:
    #Apply header row/keys to all values and create dataset
    d = dict(zip(header,line))
    #Convert relevant strings to int
    for field in ['star_rating', 'helpful_votes', 'total_votes']:
        d[field] = int(d[field])
    #Convert relevant strings to boolean
    for field in ['vine', 'verified_purchase']:
        if d[field] == 'Y':
            d[field] = True
        else:
            d[field] = False
    #Convert relevant strings (review_date) to structured time object
    d['review_date'] = time.strptime(d['review_date'], "%Y-%m-%d")
    d['review_year'] = d['review_date'].tm_year
    d['review_month'] = d['review_date'].tm_mon
    d['review_mday'] = d['review_date'].tm_mday

    #Append each line to the dataset
    dataset.append(d)
```

The Amazon Toy Review dataset has millions of reviews of toys and games sold on Amazon.com in the United States. Below is a table explaining the different fields, their data types, measures and descriptions

Name	Data Type	Measure	Description
marketplace	nominal	US	only looking at the US market
customer_id	string	NA	ID of Customer
review_id	string	NA	ID of review
product_id	string	NA	ID of product
product_parent	string	NA	ID of product parent
product_title	string	Name	Title of Product
product_category	nominal	Toys	Product Category
star_rating	integer	1-5	1 being poor and 5 being excellent
helpful_votes	integer	Number	Number of votes that marked the review as helpful
total_votes	integer	Number	Total votes on the review
vine	boolean	Y/N	Is the review a part of vine program
verified_purchase	boolean	Y/N	Is the review after a verified purchase
review_headline	string	NA	Headline of the review
review_body	string	NA	Body of the Review
review_date	Time Structure	YYYY-MM-DD	Date of the Review
review_year	integer	YYYY	Year of the Review
review_month	integer	MM	Month of the Review
review_mday	integer	DD	Date in the month of the Review

*NA = Not Applicable

In [2]:
```
print("The dataset has", len(dataset), "reviews")
```

```
The dataset has 4864249 reviews
```

In [3]:
```
#Let's review the first value of the dataset
dataset[0]
```

Out[3]:
```
{'marketplace': 'US',
 'customer_id': '18778586',
 'review_id': 'RDIJS7QYB6XNR',
 'product_id': 'B00EDBY7X8',
 'product_parent': '122952789',
 'product_title': 'Monopoly Junior Board Game',
 'product_category': 'Toys',
 'star_rating': 5,
 'helpful_votes': 0,
 'total_votes': 0,
 'vine': False,
 'verified_purchase': True,
 'review_headline': 'Five Stars',
 'review_body': 'Excellent!!!',
 'review_date': time.struct_time(tm_year=2015, tm_mon=8, tm_mday=31, tm
_hour=0, tm_min=0, tm_sec=0, tm_wday=0, tm_yday=243, tm_isdst=-1),
 'review_year': 2015,
 'review_month': 8,
 'review_mday': 31}
```

Pandas and DataFrames are very useful in calculating summary statistics. So, create a dataframe from the existing dataset.

Using Python<3.6. So, order of columns will be lexographic by default. It is easier to maintain consistency with key value data structure. So, clarify order of columns as datset is being imported

In [4]:
```
import numpy as np
import pandas as pd
df = pd.DataFrame(dataset)
df = df[['marketplace', 'customer_id','review_id','product_id','product_
parent','product_title','product_category','star_rating','helpful_votes'
,'total_votes','vine','verified_purchase','review_headline','review_bod
y','review_date', 'review_year', 'review_month', 'review_mday']]
```

In [5]:
```
df.head()
```

Out[5]:

	marketplace	customer_id	review_id	product_id	product_parent	product_title	pro
0	US	18778586	RDIJS7QYB6XNR	B00EDBY7X8	122952789	Monopoly Junior Board Game	
1	US	24769659	R36ED1U38IELG8	B00D7JFOPC	952062646	56 Pieces of Wooden Train Track Compatible wit...	
2	US	44331596	R1UE3RPRGCOLD	B002LHA74O	818126353	Super Jumbo Playing Cards by S&S Worldwide	
3	US	23310293	R298788GS6I901	B00ARPLCGY	261944918	Barbie Doll and Fashions Barbie Gift Set	
4	US	38745832	RNX4EXOBBPN5	B00UZOPOFW	717410439	Emazing Lights eLite Flow Glow Sticks - Spinni...	

We now have two representations of Amazon Toys Customer review data

1. Key Value Pair data - "dataset"

2. Pandas data frame - "df"

We will mostly use Pandas data frame but will use key value pair data when more efficient

In [6]:
```
df.shape
```

Out[6]:
```
(4864249, 18)
```

Basic Summary Statistics - Mean, count etc.

In [7]:
```
df['star_rating'].describe()
```

Out[7]:
```
count    4.864249e+06
mean     4.211735e+00
std      1.263352e+00
min      1.000000e+00
25%      4.000000e+00
50%      5.000000e+00
75%      5.000000e+00
max      5.000000e+00
Name: star_rating, dtype: float64
```

Average rating across all years is 4.211 across a total of 4,864,249 reviews

Calculate the number of unique number of toys in the dataset

In [8]:
```
from collections import defaultdict
nProducts = defaultdict(int)
for x in dataset:
    nProducts[x['product_id']] += 1
```

In [9]:
```
print("The total number of toys and games reviewed in this dataset are",
len(nProducts), "! That is a lot of Toys and Games!!")
```

```
The total number of toys and games reviewed in this dataset are 664062
! That is a lot of Toys and Games!!
```

Lots of Toys!!

Most Reviewed item

In [10]:
```
MostReviewed = df.groupby(['product_id','product_title'])['star_rating']
.agg(['mean', 'count'])
print(MostReviewed.sort_values(by='count',ascending=False).head(100))
```

product_id	product_title	mean	count
B004S8F7QM	Cards Against Humanity	4.860138	24281
B005JFNE8G	Cards Against Humanity: First Expansion	4.826568	6060
B008JNPBYK	Cards Against Humanity: Second Expansion	4.861216	3963
B0053X62GK	VTech Sit-to-Stand Learning Walker	4.668185	3951
8499000606	Syma S107/S107G R/C Helicopter *Colors Vary	4.194132	3647
B00B3YT030	Cards Against Humanity: Third Expansion	4.886903	2741
B0039S7NO6	Spot It!	4.800000	2530
B003G4IM4S	Accoutrements Horse Head Mask	4.529627	2464
B00D3IN11Q	UDI U818A 2.4GHz 4 CH 6 Axis Gyro RC Quadcopter...	3.870167	2457
B004A8ZRB0	Syma S107/S107G R/C Helicopter with Gyro	4.183993	2424
B00C6Q1Z6E	Disney Frozen Sparkle Princess Elsa Doll(Discon...	4.538593	2332
B00F9F6OVK	Cards Against Humanity: Fourth Expansion	4.887293	2298
B00DMC6KAC	Rainbow Loom Crafting Kit includes Loom, Metal ...	3.853497	2116
0975277324	Ticket To Ride	4.804453	2066
B00F4WMAI4	Syma S107/S107G R/C Helicopter with Gyro	4.214811	2039
B000W7JWUA	The Settlers of Catan	4.732941	2037
B001BKX0V4	Little Tikes EasyScore Basketball Set	4.551892	1850
B007GE75HY	Mega Bloks 80-Piece Big Building Bag	4.758977	1838
B000OCEWGW	Liquid Ass	4.682340	1829
1933054395	MindWare Qwirkle Tile Game	4.709216	1747
B000EUHKUE	Moon In My Room Remote Control Wall Décor Night...	4.456929	1602
B000FK3WDC	Multi Voice Changer - 6.5", Colors May Vary	3.385344	1583
B000BNCA4K	Manhattan Toy Winkel Rattle and Sensory Teether...	4.560538	1561
B00000IV35	Five Crowns	4.835662	1503
B00CIXVITO	Snap Circuits Jr. SC-100	4.851035	1497
B00008BFZH	Snap Circuits Jr. SC-100	4.852881	1475
B00000ISC5	Insect Lore Live Butterfly Garden	4.345890	1460

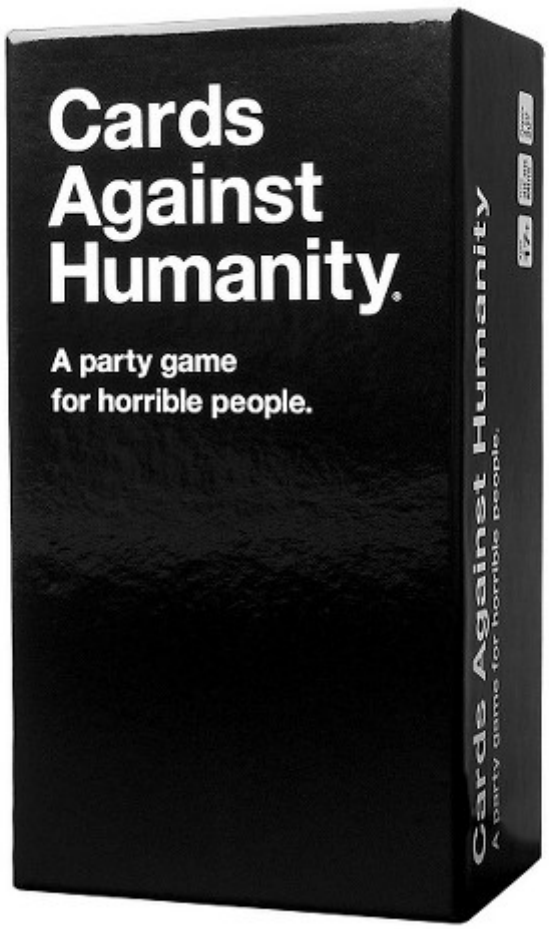
```
B00KI03U8Q Syma X5C Explorers 2.4G 4CH 6-Axis Gyro RC Quad...  4.246914
1458
B00000K3BR The Original Stomp Rocket                          4.519972
1452
B00006C29I Fisher-Price Ocean Wonders Aquarium Cradle Swing   4.401798
1446
...                                                                ...
...
B000OEUUG6 Joissu Flingshot Slingshot Flying Screaming Monkey  3.985782
1055
B003EIK136 Rory's Story Cubes                                 4.649240
1052
B001DAWY1Y Intex River Run II                                 3.686069
1048
B0057MGKO6 niceEshop(TM) Cute Velvet Animal Style Finger P...  4.299615
1038
B002JCS5JA Melissa & Doug Deluxe Standing Easel               4.486804
1023
B002SQW38Q Play Tent Princess Castle by Pockos - Features ...  4.322549
1020
B000N5QNSK Battat Take-A-Part Airplane                        4.435039
1016
B003AIM52A HedBanz Game - Edition may vary                    4.619329
1014
B000062XQ8 LeapFrog Learning Table                            4.715976
1014
B00ABA0ZOA Jenga Hardwood Game                                4.700197
1014
B0083IXKYY Fisher-Price Ocean Wonders Soothe and Glow Seah...  4.444773
1014
B00F3TKLRA Aukmont 1800 Pieces Loom Bands with 75 Clips       4.070647
1005
B0002AUWKG Melissa & Doug                                     4.432836
1005
B0006N8X3M Learning Resources Pretend & Play Teaching Cash...  4.234297
1003
B007U7M0WM LeapFrog LeapPad2 Recharger Pack (Works with al...  4.544367
1003
B004HKQI9G Little Tikes Princess Cozy Truck Ride-On           4.451000
1000
B00IFWSMMS Taboo Buzzd Game                                   4.478478
999
B00004TFT1 Power Wheels 12-Volt Rechargeable Battery          4.273374
984
B00A8UT562 Playskool Heroes Transformers Rescue Bots Energ...  4.742100
981
B001SN8GF4 Telestrations the Telephone Game Sketched Out!     4.817901
972
B000XR6MBQ The Elf on the Shelf: A Christmas Tradition wit...  4.550679
957
B0037UT3E4 Melissa & Doug Car Carrier Truck and Cars Woode...  4.587252
957
B003AVIO4K Radio Flyer 4-in-1 Stroll 'N Trike                 4.335430
954
B000641DPQ Step2  WaterWheel Activity Play Table              4.587185
952
B003621UT4 Little Tikes 2-in-1 Snug 'n Secure Swing Blue      4.617492
```

```
949
B000NOU54O My First Lab  Microscope                              4.483906
932
B000GL1EEE Lionel Polar Express Train Set - G-Gauge              3.813376
927
B00A8UT55I Playskool Heroes Transformers Rescue Bots Energ...    4.696641
923
B0089RPUHO LeapFrog LeapPad2 Explorer Kids' Learning Table...    3.903720
914
B0089W1IGG Fisher-Price Brilliant Basics Baby's First Blocks     4.437500
912

[100 rows x 2 columns]
```

The most reviewed item here is Cards Against Humanity with over 30,000 reviews (with expansions packs that are ranked 2 and 3) way ahead of the next game which has less than 4000 reviews.

Cards Against Humanity. No Suprise here!!!



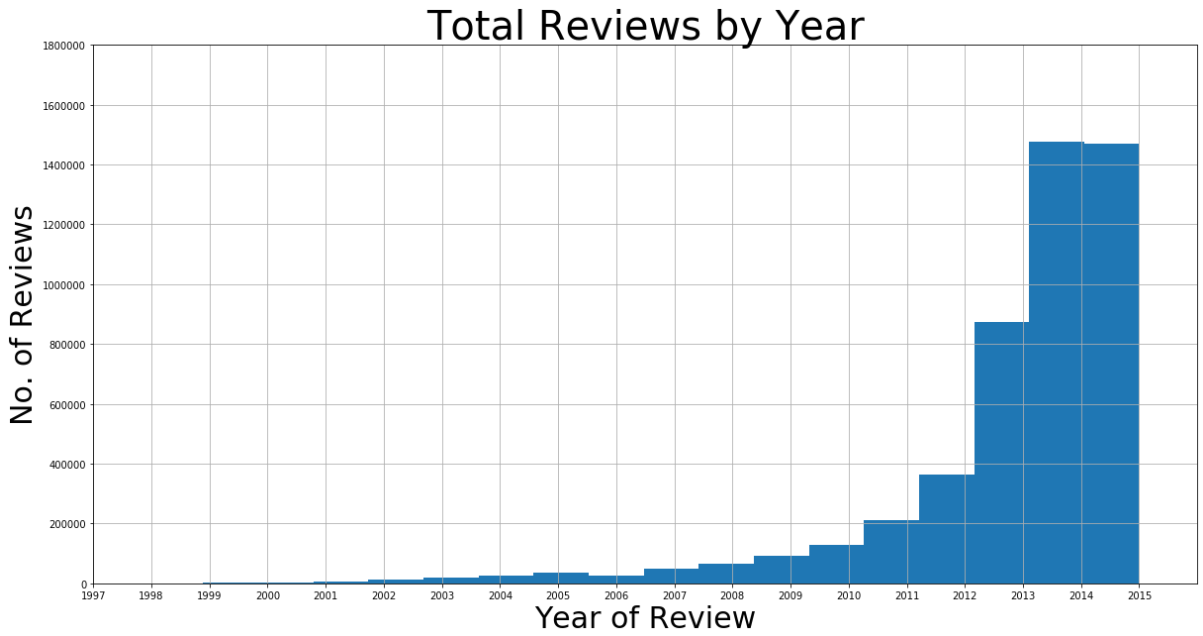
Calculate number of reviews per year - first as a table, and then as a graph

In [11]:
```
df['star_rating'].groupby(df['review_year']).count()
```

Out[11]:
```
review_year
1997          1
1998         40
1999       1974
2000       3278
2001       6184
2002      12676
2003      19165
2004      26080
2005      36113
2006      26869
2007      48233
2008      65554
2009      93176
2010     129181
2011     211454
2012     363790
2013     873984
2014    1476401
2015    1470096
Name: star_rating, dtype: int64
```

In [12]:
```
import matplotlib.pyplot as plt
from matplotlib import colors
```

In [13]:
```
fig, ax = plt.subplots(figsize=(20,10))
ax.hist(df['review_year'], bins = 19)
plt.xlabel("Year of Review", fontsize = 30)
plt.ylabel("No. of Reviews", fontsize = 30)
plt.title("Total Reviews by Year", fontsize = 40)
plt.grid(True)
plt.axis([1997,2016,0,1800000])
plt.xticks(np.arange(1997, 2016,1))
plt.show()
```



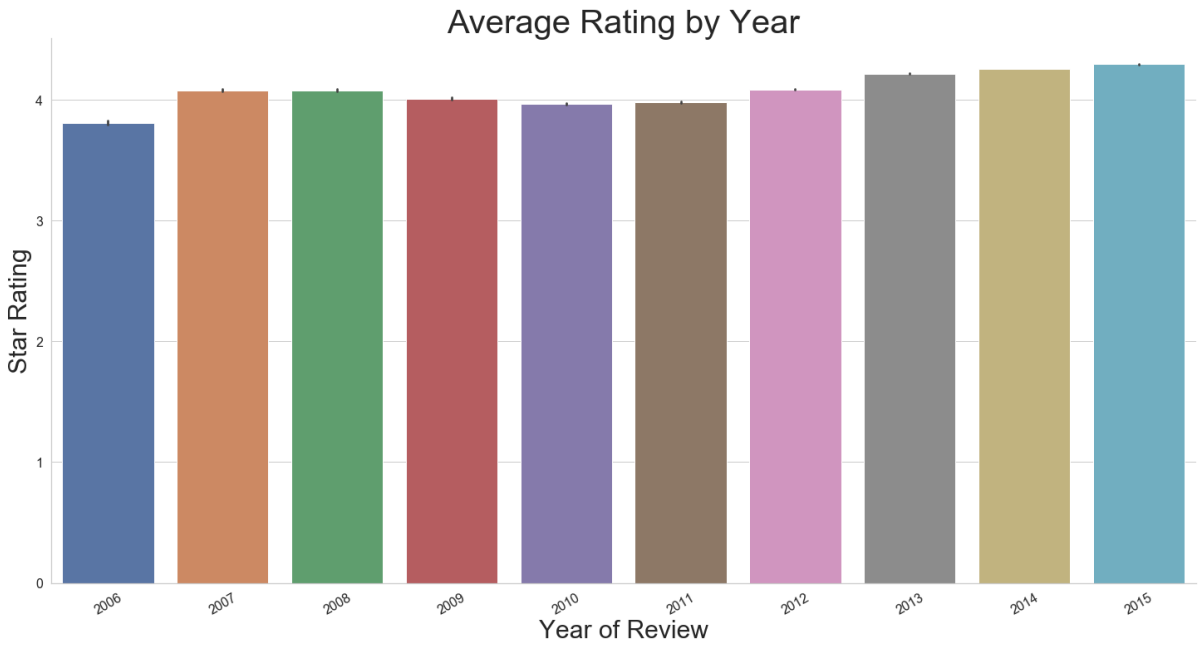
Reviews from 1997 to 2005 are very old and very few. So, Only keep reviews for 10 years - 2006 to 2015 - for further analysis

In [14]:
```
df_analysis = df[df.review_year>2005]
```

Steps 3 & 4 - Identify and plot interesting statistics

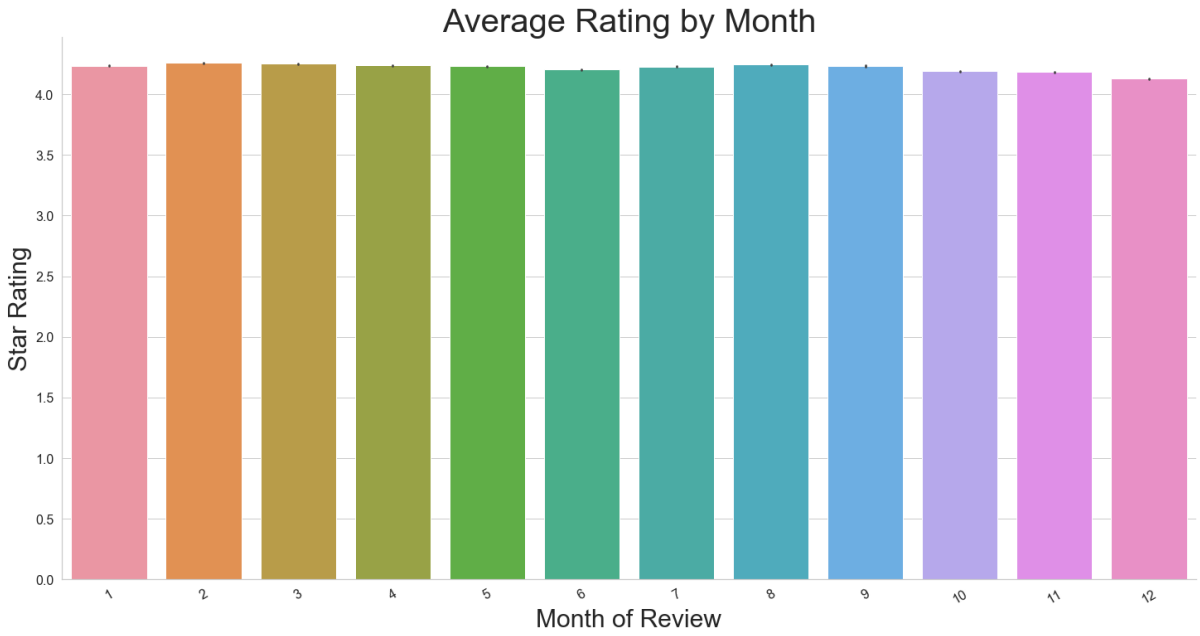
Let us first see if average rating moved by year as Amazon became more and more popular and was used by more and more customers

In [15]:
```
#create bar plot
import seaborn as sns
sns.set(style="whitegrid")
barplot = sns.catplot(x="review_year", y ="star_rating", kind = "bar", d
ata = df_analysis,height =10, aspect = 2)
barplot.set_xticklabels(rotation=30, fontsize = 16)
barplot.set_yticklabels(fontsize = 16)
plt.xlabel("Year of Review", fontsize = 30)
plt.ylabel("Star Rating", fontsize = 30)
plt.title("Average Rating by Year", fontsize = 40)
plt.show()
```



From the graph and the table, we can see upward movement of average reviews from 2010 to 2015. This needs further inquiry - it is not that people are rating the games higher, something to do with the selection of people rating the games of Amazon

In [16]:
```
import seaborn as sns
sns.set(style="whitegrid")
barplot = sns.catplot(x="review_month", y ="star_rating", kind = "bar",
data = df_analysis,height =10, aspect = 2)
barplot.set_xticklabels(rotation=30, fontsize = 16)
barplot.set_yticklabels(fontsize = 16)
plt.xlabel("Month of Review", fontsize = 30)
plt.ylabel("Star Rating", fontsize = 30)
plt.title("Average Rating by Month", fontsize = 40)
plt.show()
```



While it makes sense that the average review doesn't change that much by month, let's look at the number of reviews per month. Most sales of toys happen in the holiday season - Nov-Dec, and the summer - June-July. So, there should be a spike in reviews around that time - Nov-Jan for the holiday season, and July-Aug for the summer

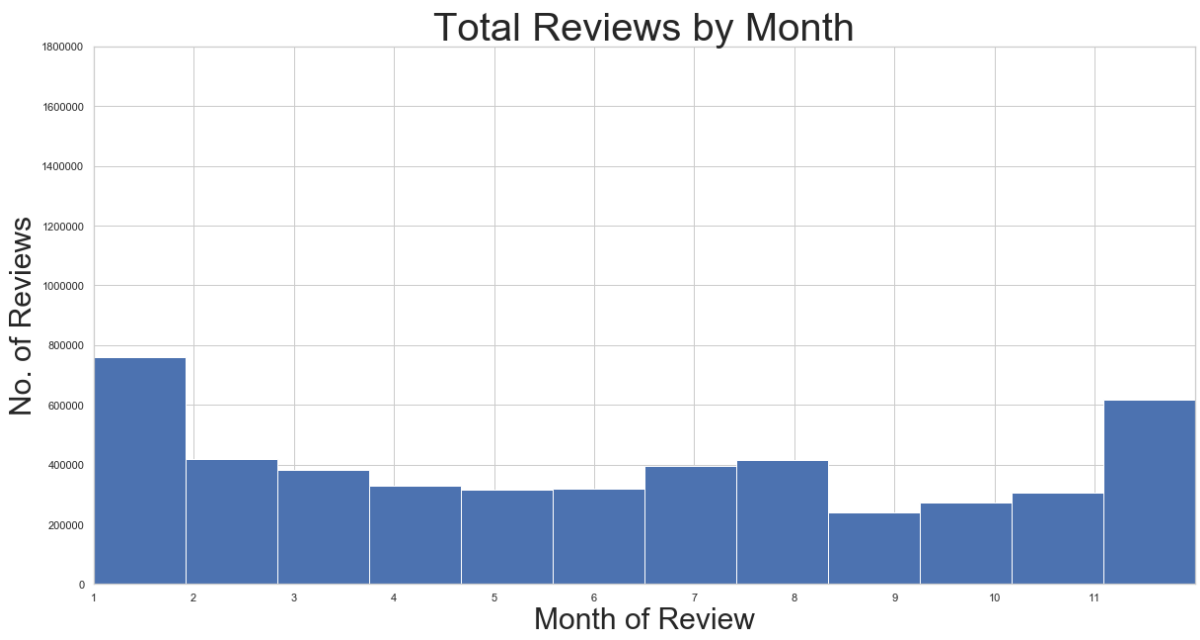
First, look at this in a tabular form

In [17]:
```
df_analysis['star_rating'].groupby(df_analysis['review_month']).count()
```

Out[17]:
```
review_month
1     757814
2     417209
3     380347
4     329066
5     314338
6     318222
7     393488
8     414208
9     239769
10    272195
11    304778
12    617304
Name: star_rating, dtype: int64
```

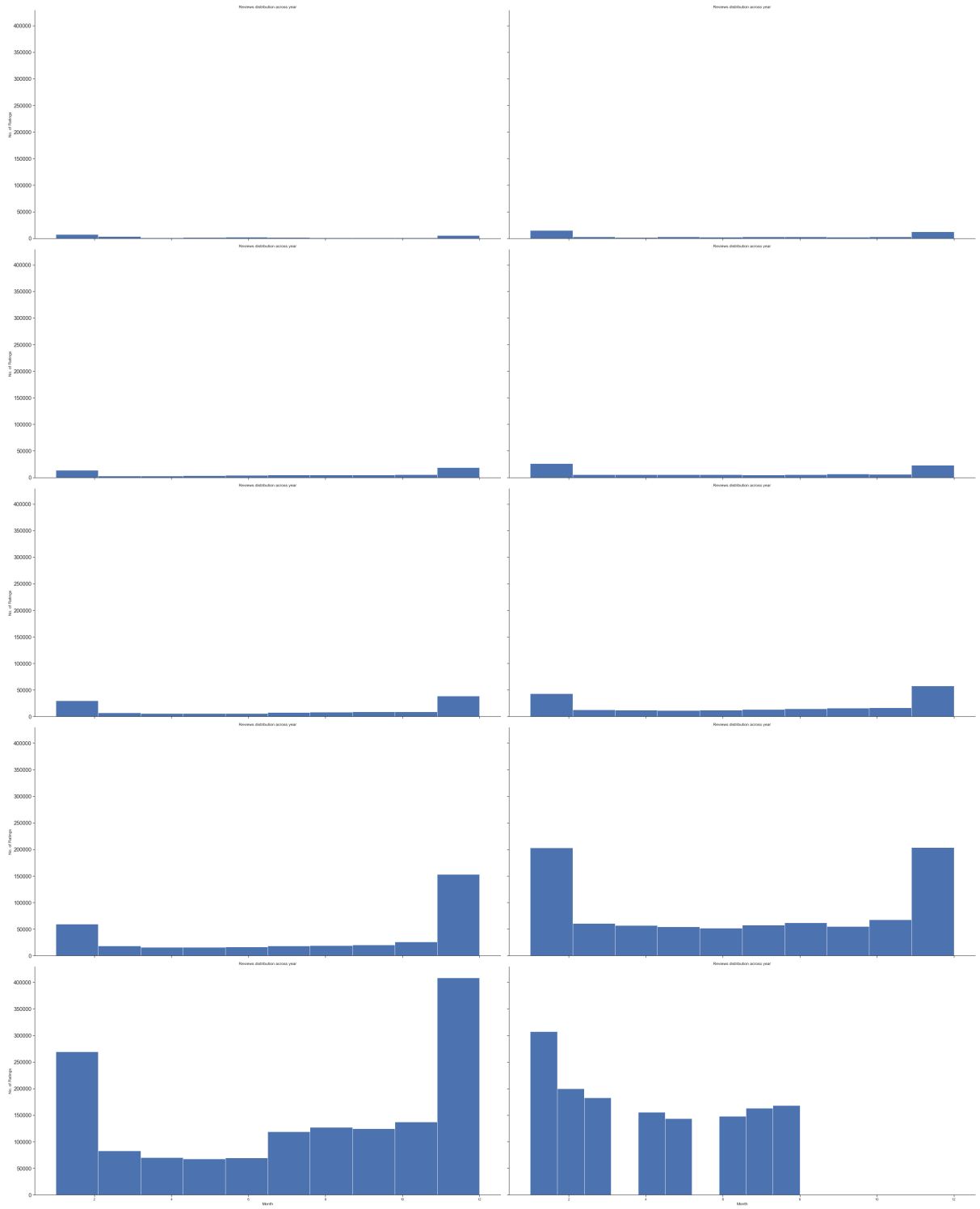
Now, look at this in a graphical form

In [18]:
```
fig, ax = plt.subplots(figsize=(20,10))
ax.hist(df_analysis['review_month'], bins = 12)
plt.xlabel("Month of Review", fontsize = 30)
plt.ylabel("No. of Reviews", fontsize = 30)
plt.title("Total Reviews by Month", fontsize = 40)
plt.grid(True)
plt.xticks(np.arange(1,12,1))
plt.axis([1,12,0,1800000])
plt.show()
```



From the table and the graph, we can clearly see the spike of reviews in Dec and Jan (for sales in holiday season) And July and Aug (for the summer). Now, let's see if this trend holds for all 10 years (2006-2015) under consideration

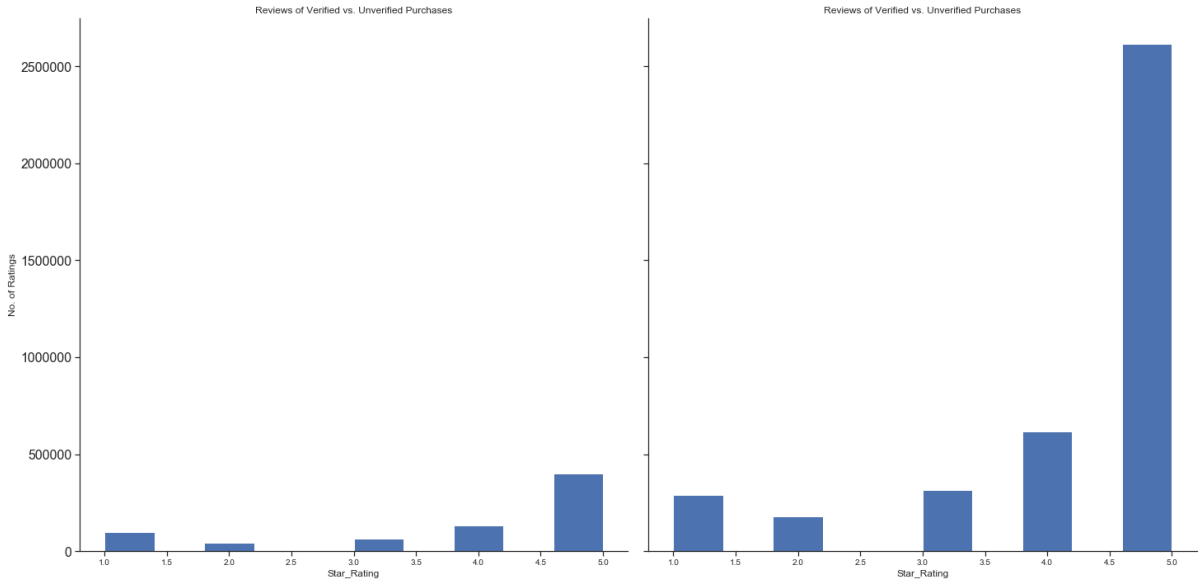
In [19]:
```
sns.set(style="ticks")
g = sns.FacetGrid(df_analysis, col = "review_year",col_wrap = 2, height
=10, aspect = 2)
g.map(plt.hist, 'review_month')
g.set_axis_labels("Month", "No. of Ratings")
g.set_titles("Reviews distribution across year", fontsize = 40)
g.set_xticklabels(fontsize = 10)
g.set_yticklabels(fontsize = 16)
plt.show()
```



The same trend - higher number of reviews in summer and holiday season holds across all the 10 years. For 2015, we only have data till August - which is what the data is empty after. This is pretty telling of the annual sales cycle for toys and games

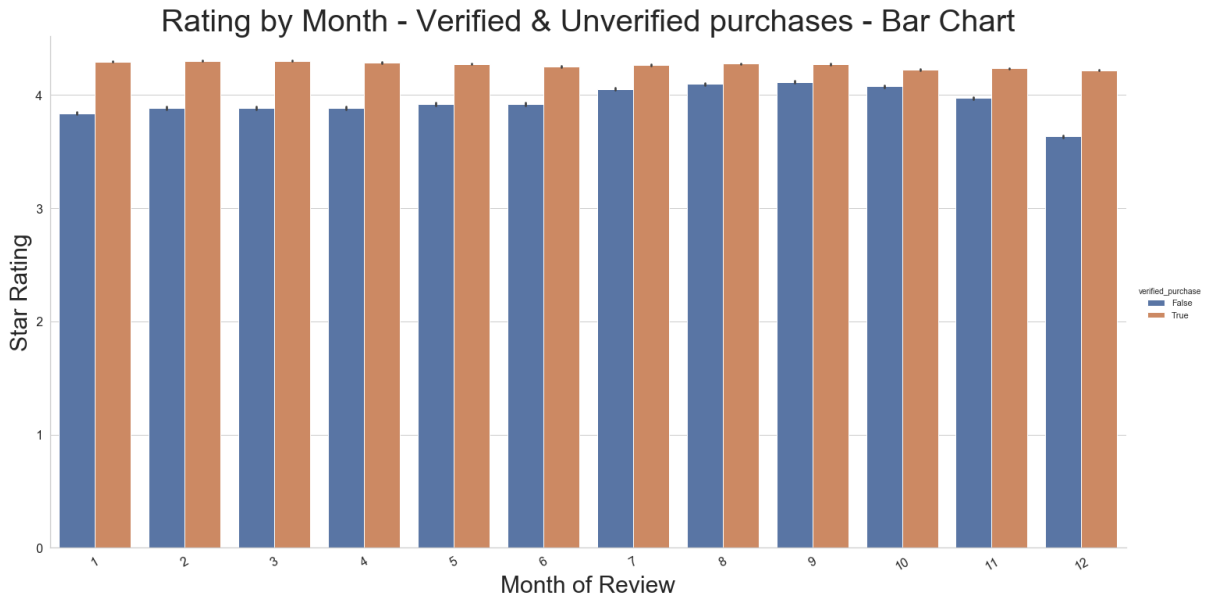
Let's look at the ratings of Verified vs. Unverified purchases

In [20]:
```
sns.set(style="ticks")
g = sns.FacetGrid(df_analysis, col = "verified_purchase",height =10, asp
ect = 1)
g.map(plt.hist, 'star_rating')
g.set_axis_labels("Star_Rating", "No. of Ratings")
g.set_titles("Reviews of Verified vs. Unverified Purchases", fontsize =
40)
g.set_xticklabels(fontsize = 10)
g.set_yticklabels(fontsize = 16)
plt.show()
```

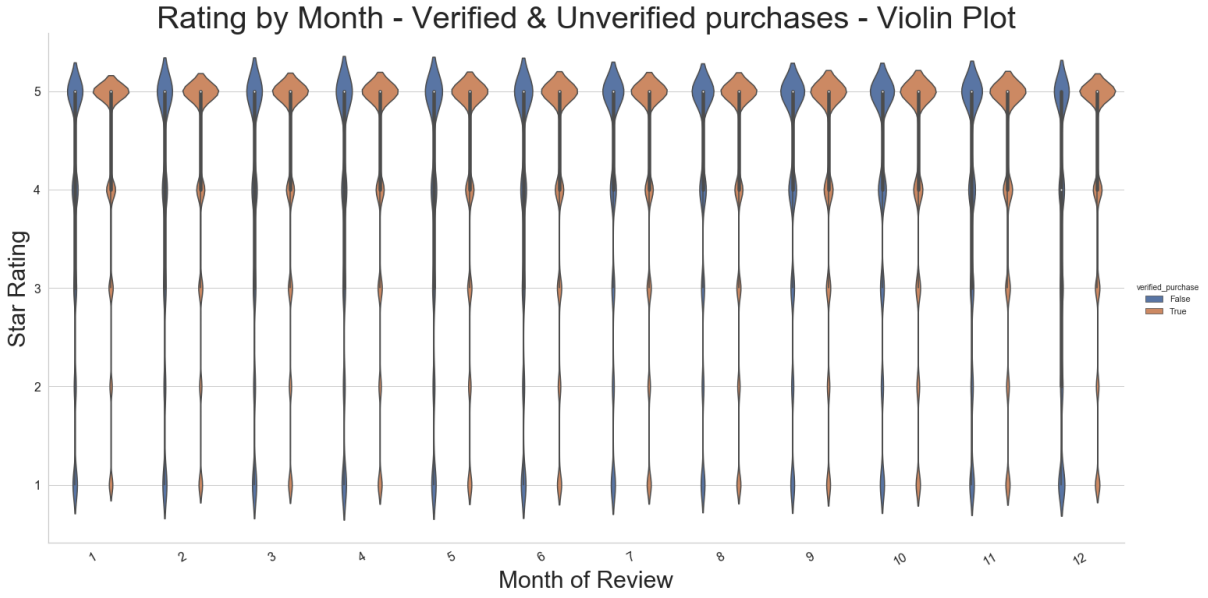


It seems most of the 5 star ratings and a higher percentage of 5 star ratings are coming from verified purchases. Let's look at the contrast between verified and unverified purchases now

In [21]:
```
import seaborn as sns
sns.set(style="whitegrid")
barplot = sns.catplot(x="review_month", y ="star_rating", hue = "verifie
d_purchase", kind = "bar", data = df_analysis,height =10, aspect = 2)
barplot.set_xticklabels(rotation=30, fontsize = 16)
barplot.set_yticklabels(fontsize = 16)
plt.xlabel("Month of Review", fontsize = 30)
plt.ylabel("Star Rating", fontsize = 30)
plt.title("Rating by Month - Verified & Unverified purchases - Bar Char
t", fontsize = 40)
plt.show()
```



In [22]:
```
import seaborn as sns
sns.set(style="whitegrid")
barplot = sns.catplot(x="review_month", y ="star_rating", hue = "verifie
d_purchase", kind = "violin", data = df_analysis,height =10, aspect = 2)
barplot.set_xticklabels(rotation=30, fontsize = 16)
barplot.set_yticklabels(fontsize = 16)
plt.xlabel("Month of Review", fontsize = 30)
plt.ylabel("Star Rating", fontsize = 30)
plt.title("Rating by Month - Verified & Unverified purchases - Violin Pl
ot", fontsize = 40)
plt.show()
```



It seems verified purchases result in a higher rating (Across all months). The difference seems to be particularly high for holiday sales (Dec-Jan reviews).

Identify top rated products to recommend: Products that have lower number of reviews shouldn't be considered at this stage as they may have too many high or low ratings. Our confidence in recommending a product that has fewer than, say 50 reviews, is low. So, we temporarily discard them for recommended product analysis

In [23]:
```
df_topRated = df_analysis.groupby('product_id').filter(lambda x: len(x)>
50)
```

In [24]:
```
df_topRated.shape
```

Out[24]: (2111018, 18)

Running this one filter by dropping products with less than or equal to 50 reviews reduced the number of reviews to almost 50%.

In [25]:
```
topRated = df_topRated.groupby(['product_id','product_title'])['star_rating'].agg(['mean', 'count'])
```

In [26]:
```
print(topRated.sort_values(by='mean',ascending=False).head(100))
```

product_id	product_title	mean count
B0094PF0EI	Wild Republic Cuddlekin Honey Badger 12" Plush	5.000000 51
B00ATSOSJM	1D One Direction 22 inch Two Tone Teddy Bear wi...	5.000000 62
B000KU5JO2	THE Original Flying Turtle Ride-On Scooter - No...	4.983871 62
B00EZIKSZK	Lalaloopsy Yuki Kimono Doll	4.981818 55
B004FVG3RY	ROLLICK! THE HYSTERICAL GAME OF CLUES AND COLLA...	4.971831 71
B00M0DHI4Q	WolVol (Set of 3) Push and Go Friction Powered ...	4.967742 62
B002T5TI98	Magformers Standard Super Magformers Set (30-pi...	4.966102 59
B00OHLY4CI	Flexifoil 2.05m Power Kite, Big Buzz Sport Foil...	4.965517 58
B004ID794A	Ravensburger Tropical Birds - 300 Pieces Large ...	4.964286 56
B007GEMQ2I	LEGO Creator 10224 Town Hall	4.963636 55
B00B4WJITW	Aurora World Dreamy Eyes Plush Flame Red Dragon...	4.963636 55
B00J4S6Y8G	LEGO Chima 70145 Maula's Ice Mammoth Stomper Bu...	4.962264 53
B00NHQGKHQ	LEGO Creator Red Creatures	4.962264 53
B00D3Y18WO	Bandai Tamashii Nations S.H. MonsterArts MFS-3 ...	4.961538 52
B00MZ6BQYQ	Monster High Haunted Student Spirits River Styx...	4.961165 103
B003UX1KVM	aBaby Child's Folding Chairs (Set of 2)	4.960784 51
B00IB7DEO8	LEGO Chima 70132 Scorm's Scorpion Stinger	4.960784 51
B003TSD930	Webkinz Signature Endangered Brown Bear	4.960784 51
B006UPHPRA	Aurora World Miyoni 11" Boxer Stuffed Dog	4.959459 74
B00T03U5J4	Monster High Draculaura Collector Doll (Discont...	4.955882 68
B007Q0OEUE	LEGO City Excavator Transport 4203	4.954545 88
B00LV1D0UE	Alex the Monkey - You Receive One & Another One...	4.954023 87
B00IVP13SO	Monster High Social Spots Creepateria Accessory	4.953125 64
B00DLTQH54	Good Smile Fire Emblem Awakening Tharja PVC Fig...	4.948276 58
B001J8HMPS	Pacific Play Tents One Touch Play Cabana No.19010	4.948276 58
B008RQ1PSK	DRINK-A-PALOOZA Party Board Game: combines "old...	4.947368 76
B0006N6ODS	Liquid Cement For Plastics 1 oz	4.946667 75

```
B00B9J2HPM Breyer Traditional Gypsy Vanner Horse Toy Model     4.946667
75
B00HUJHIBI #1 Best quality rainbow loom band refill pack -...  4.946429
56
B00IVFCLZS Monster High Ghoul Sports Clawdeen Wolf Doll        4.946429
56
...                                                                 ...
...
B00IHK4RUO Mezco Toyz Breaking Bad 6" Saul Goodman Bobbleh...  4.915254
59
B004ID5RY4 Ravensburger Secret Sanctuary - 300 Pieces Larg...  4.915254
59
B00SSSG6FE Kangaroo Stuffed Giraffe - Toy Plush Giraffe- 2...  4.914634
82
B00IGNZZN0 LEGO City Arctic Outpost 60035 Building Toy (Di...  4.914286
70
B00QD78PII AEG Pretty Pretty Smash Up Expansion                4.914286
70
B00005KBVD Step 2 Up & Down Roller Coaster                    4.913396
739
B001GOO0VO Breyer Traditional Big Chex to Cash Horse Toy M...  4.913043
92
B00K8FDQS0 Cuddle Barn Lil Bub Adorable Kitten Cat Plush T...  4.912621
103
B00PSZLHWM Minecraft Figure 4-pack Hostile Mobs Set            4.912281
57
B00705Z9LI Wild Republic 11041 Walrus Stuffed Animal, 15-Inch  4.912281
57
B00AZL34A2 Ty Beanie Boos Glamour Leopard Plush, Pink          4.910714
56
B00A850SGS LEGO City Fire Emergency 60003                      4.910569
123
B005PIHMZ8 Fisher Price Loving family Exclusive Holiday Do...  4.910112
89
B00000J50C Uncle Goose Classic ABC Blocks with Pull Wagon ...  4.909910
111
B001I8BZZC EzyRoller Classic Ride On                           4.909836
122
B005JT5P7A 20 Pairs - Neon Prism Diffraction Fireworks Gla...  4.908046
87
B001REQRS2 Armor of God High Relief Challenge Coin             4.908046
87
B009K8308C Yu-Gi-Oh! - Five-Headed Dragon (LC03-EN004) - L...  4.907407
108
B00DEHRNUG Kotobukiya DC Comics Superman for Tomorrow ArtF...  4.907407
54
B00CM6Y268 Ty Beanie Boos Swoops Brown Barn Owl Plush          4.905660
53
B00IVFCNQK Monster High Ghoul Sports Spectra Vondergeist Doll  4.905660
53
B001QIH6S4 Thomas And Friends Wooden Railway - Hiro            4.905405
74
B009Q3YX8W Melissa & Doug Slice and Bake Wooden Christmas ...  4.904762
63
B00B2B051A Learning Resources New Sprouts Dinner Foods Bas...  4.904255
188
B00FEN5MDS Disney Frozen Exclusive 12" Classic Doll Kristoff   4.903846
```

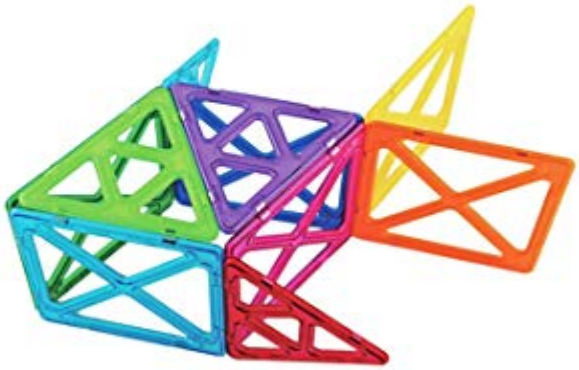
```
52
B00F8HK4PG Disney Lambie Plush - 11'' - Doc McStuffins         4.903614
83
B00GSPFD26 LEGO Movie Castle Cavalry 70806                     4.901961
51
B002KRDIIW Merrymakers Llama Llama Doll                        4.901961
51
B007EAIF6A Bandai Hobby #08 RX-178 Gundam MK II (AEUG) 1/1...  4.901961
51
B009MJU8P2 Schleich Hippopotamus Toy Figure                    4.901961
51

[100 rows x 2 columns]
```

Every year around the holiday season, blogs and websites recommend toys and games. Based on the above sorting, and a little bit of sbjectivity and for fun, here are some recommended toys and games (across all ages) for 2019 holiday season! Ideally we would divide this by age group, but this is not in this dataset, and perhaps we will be able to locate a dataset with age groups later on

1) Wild Republic Cuddlekin Honey Badger 12" Plush

2) Magformers Standard Super Magformers Set



3) Step 2 Up & Down Roller Coaster

4) Uncle Goose Classic ABC Blocks with Pull Wagon

5) Learning Resources New Sprouts Dinner Foods

Conclusion: We took a rather large dataset of 4.8 million Amazon customer reviews of toys and games and conducted analysis with histograms, bar charts and violin charts that showed clear patters around holiday shopping and reviews.

The tables and graphs also showed a higher rating for verified purchases - which also could mean that customers are using review data to make better choices. But this hypothesis needs further testing.