Assignment Report

- This report is a summary of technical approach. For more detailed discussion of the data and methodology please see notes in accompanying Python notebook and R script.
- Because of the iterative nature of data analysis, although the steps documented here do reflect a sequential process, it was by no means a strictly linear one, and this report should be read with that in mind.

<table>
<tr><td>Action</td><td>Evidence</td><td>Notes</td></tr>
<tr><td colspan="3">Thorough exploration of the data</td></tr>
<tr><td>Basic initial data exploration:</td><td><pre>
# Any missing values?
reviews.info()
# No missing values

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 11 columns):
 #   Column   Non-Null Count Dtype
--- ------   -------------- -----
 0   gender   2000 non-null   object
 1   age   2000 non-null   int64
 2   remuneration (k£)   2000 non-null   float64
 3   spending_score (1-100) 2000 non-null   int64
 4   loyalty_points   2000 non-null   int64
 5   education   2000 non-null   object
 6   language   2000 non-null   object
 7   platform   2000 non-null   object
 8   product   2000 non-null   int64
 9   review   2000 non-null   object
 10 summary   2000 non-null   object
</pre></td><td>

missing value checks
descriptive statistics
missing data check
data conversions
comparison with metadata
exploratory visualisations

</td></tr>
<tr><td></td><td><pre>
t_sales.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 352 entries, 0 to 351
Data columns (total 9 columns):
 #   Column   Non-Null Count Dtype
--- ------   -------------- -----
 0   Ranking   352 non-null   int64
 1   product   352 non-null   int64
 2   Platform   352 non-null   object
 3   Year   350 non-null   float64
 4   Genre   352 non-null   object
 5   Publisher   352 non-null   object
 6   NA_Sales   352 non-null   float64
 7   EU_Sales   352 non-null   float64
 8   Global_Sales 352 non-null   float64
dtypes: float64(4), int64(2), object(3)
memory usage: 24.9+ KB
</pre></td><td></td></tr>
</table>

Detailed exploration of the data

	age	remuneration (k£)	spending_score (1-100)	loyalty_points	product
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	39.495000	48.079060	50.000000	1578.032000	4320.521500
std	13.573212	23.123984	26.094702	1283.239705	3148.938839
min	17.000000	12.300000	1.000000	25.000000	107.000000
25%	29.000000	30.340000	32.000000	772.000000	1589.250000
50%	38.000000	47.150000	50.000000	1276.000000	3624.000000
75%	49.000000	63.960000	73.000000	1751.250000	6654.000000
max	72.000000	112.340000	99.000000	6847.000000	11086.000000

```
1 t_sales.describe()
```

	Ranking	product	Year	NA_Sales	EU_Sales	Global_Sales
count	352.000000	352.000000	350.000000	352.000000	352.000000	352.000000
mean	1428.017045	3607.227273	2006.985714	2.515966	1.643778	5.334688
std	2743.580936	2360.239834	6.750343	3.409479	2.025752	6.264962
min	1.000000	107.000000	1982.000000	0.000000	0.000000	0.010000
25%	88.750000	1945.000000	2003.000000	0.477500	0.390000	1.115000
50%	176.500000	3340.000000	2009.000000	1.820000	1.170000	4.320000
75%	1439.750000	5435.750000	2012.000000	3.125000	2.160000	6.435000
max	16096.000000	9080.000000	2016.000000	34.020000	23.800000	67.850000

Explore the data by columns

Spend Score

Metadata states: "A score is assigned to the customer by Turtle Games based on the customer's spending nature and behaviour. The value rang
and 100."

Checking the structure of the dataset (normality, artificiality, etc)

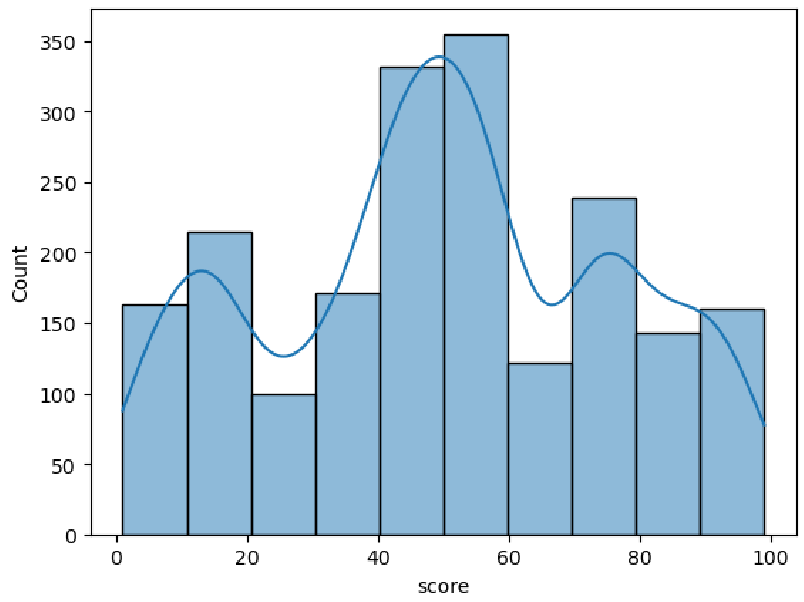
Because we know that the score ranges from 1-100 it is interesting that the mean is 50 and the std deviation is not far from 25 - indicating that the
an effort at play, to put the cutomers into quartiles that see them evenly spread across the score range, in which case we may expect that even th
have an even count of customers. Or perhaps that distribution in this dataset is due to these reviews being selected deliberatly from an even rang
scores? Investigate this further.

Sort by score (descending) to get a sense for the spending range associated with a top score, and any other easiy observable patterns (points, b
gender, language, etc).

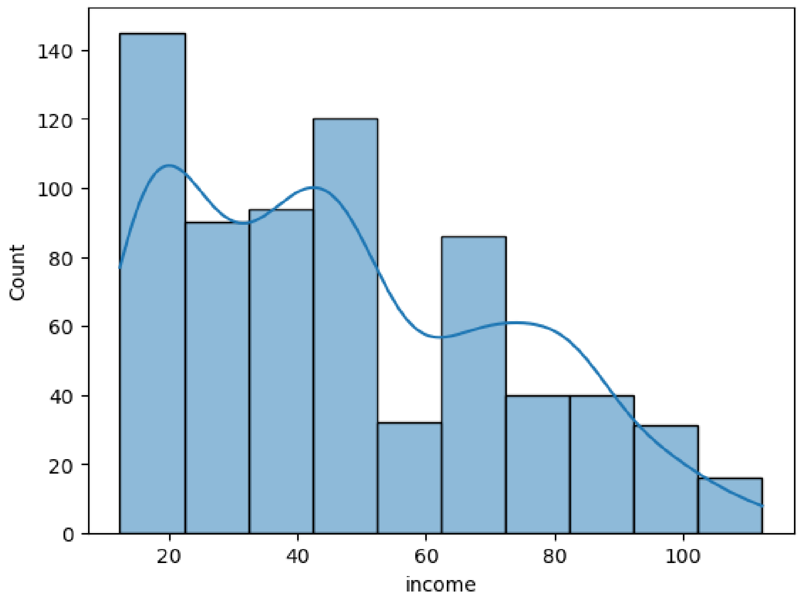
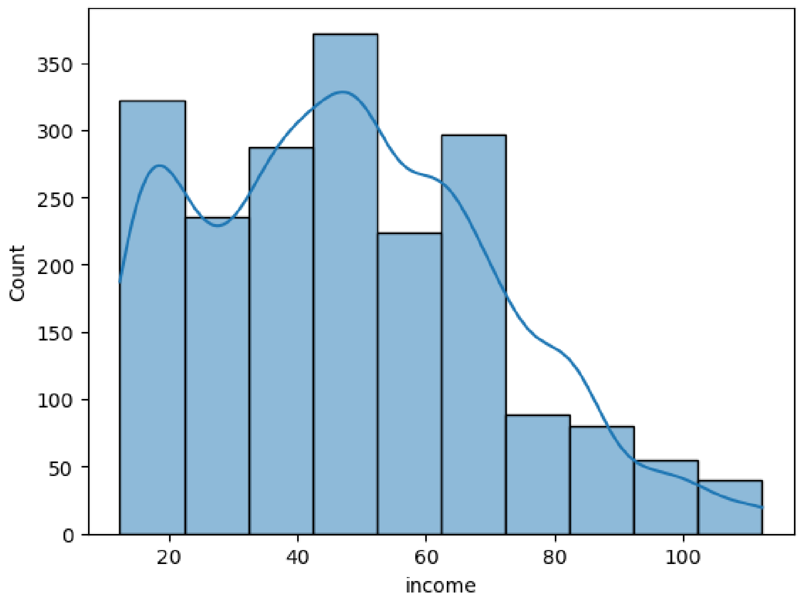
Querying why loyalty points are being chosen over spending score, when the goal is to increase overall sales: ie, why not just use spending score
why not just use overall spend?

- By columns, and then by relationships between columns, and then by relationships between the datasets.
- Anomalies investigated, e.g. distribution of spend score variable
- exploration driven by the business objective (increase sales)

Check normality of distribution of the score data



- numerical variables have some aspects of normal distribution but tests determine they are ultimately not normally distributed
  - however spend score is roughly symmetrical
  - income has an exaggerated right tail

Attempt to find normal distributions within subsets of the data		
		Income after normalising the data for gender and education groups, has improved. This is especially clear with the 50-60k income bin, and the 20-40k income bins. The absence of the left tail is not overly concerning regarding normality as [many if not most countries have a minimum wage]() (especially noting EU and NA countries / states, as these dominate TG's sales figures), Income represents annual aggregations of such, accounting for the absence of a left tail. There are also no customers under 17, and all customers in the dataset have an income. This is to be expected in a commercial dataset, where income is required to engage.

visual inspections of dataframes

```
top_50 = sales[sales['Ranking_Bracket'] == 0]

top_50.sort_values('Ranking', ascending = True, inplace = True)
top_50.head(50)
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_11380\1838584909.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  top_50.sort_values('Ranking', ascending = True, inplace = True)
```

Out[283]:

	Ranking	product	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Global_Sales	count	product2	Ranking_Bracket
0	1	107	Wii	2006.0	Sports	Nintendo	34.02	23.80	67.85	1	1070	0
1	2	123	NES	1985.0	Platform	Nintendo	23.85	2.94	33.00	2	1230	0
2	3	195	Wii	2008.0	Racing	Nintendo	13.00	10.56	29.37	1	1950	0
3	4	231	Wii	2009.0	Sports	Nintendo	12.92	9.03	27.06	1	2310	0
4	5	249	GB	1996.0	Role-Playing	Nintendo	9.24	7.29	25.72	1	2490	0
5	6	254	GB	1989.0	Puzzle	Nintendo	19.02	1.85	24.81	2	2540	0
6	7	263	DS	2006.0	Platform	Nintendo	9.33	7.57	24.61	1	2630	0
7	8	283	Wii	2006.0	Misc	Nintendo	11.50	7.54	23.80	1	2830	0
8	9	291	Wii	2009.0	Platform	Nintendo	11.96	5.79	23.47	1	2910	0
9	10	326	NES	1984.0	Shooter	Nintendo	22.08	0.52	23.21	1	3260	0
10	11	399	DS	2005.0	Simulation	Nintendo	7.44	9.02	20.30	1	3990	0
11	12	405	DS	2005.0	Racing	Nintendo	8.04	6.21	19.20	1	4050	0
12	13	453	GB	1999.0	Role-Playing	Nintendo	7.38	5.07	18.94	1	4530	0
13	14	466	Wii	2007.0	Sports	Nintendo	7.33	6.58	18.63	1	4660	0
14	15	486	Wii	2009.0	Sports	Nintendo	7.45	7.04	18.04	1	4860	0
15	16	504	X360	2010.0	Misc	Microsoft Game Studios	12.28	4.05	17.89	1	5040	0
16	17	515	PS3	2013.0	Action	Take-Two Interactive	5.75	7.60	17.55	5	5150	0
17	18	518	PS2	2004.0	Action	Take-Two Interactive	7.73	0.33	17.06	4	5180	0
18	19	535	SNES	1990.0	Platform	Nintendo	10.48	3.08	16.90	2	5350	0
19	20	577	DS	2005.0	Misc	Nintendo	3.90	7.59	16.58	1	5770	0
20	21	615	DS	2006.0	Role-Playing	Nintendo	5.26	3.71	15.06	1	6150	0
21	22	618	GB	1989.0	Platform	Nintendo	8.88	2.22	14.87	1	6180	0

Throughout this analysis, including use of .head() and .info() methods to check the effectiveness and completeness of data transformations.

Anomalous features of the dataset identified

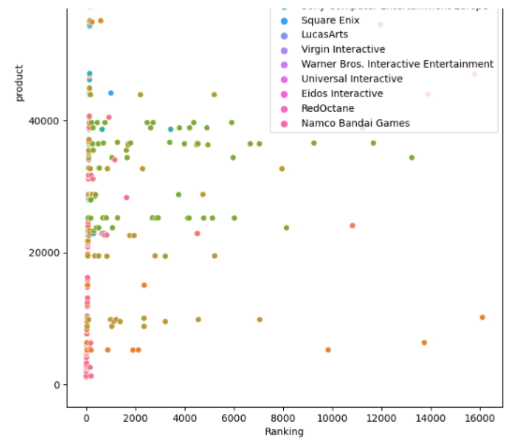
In [202]:

```
# Determine the product2 values
t_sales_grouped['product2']= t_sales_grouped['product'].astype(str) + (t_sales_grouped.groupby\
                                                            ('product').cumcount()).astype(str)

t_sales_grouped.head(20)
```

Out[202]:

	Ranking	product	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Global_Sales	count	product2
0	1	107	Wii	2006.0	Sports	Nintendo	34.02	23.80	67.85	1	1070
1	2	123	NES	1985.0	Platform	Nintendo	23.85	2.94	33.00	2	1230
2	3	195	Wii	2008.0	Racing	Nintendo	13.00	10.56	29.37	1	1950
3	4	231	Wii	2009.0	Sports	Nintendo	12.92	9.03	27.06	1	2310
4	5	249	GB	1996.0	Role-Playing	Nintendo	9.24	7.29	25.72	1	2490
5	6	254	GB	1989.0	Puzzle	Nintendo	19.02	1.85	24.81	2	2540
6	7	263	DS	2006.0	Platform	Nintendo	9.33	7.57	24.61	1	2630
7	8	283	Wii	2006.0	Misc	Nintendo	11.50	7.54	23.80	1	2830
8	9	291	Wii	2009.0	Platform	Nintendo	11.96	5.79	23.47	1	2910
9	10	326	NES	1984.0	Shooter	Nintendo	22.08	0.52	23.21	1	3260
10	11	399	DS	2005.0	Simulation	Nintendo	7.44	9.02	20.30	1	3990
11	12	405	DS	2005.0	Racing	Nintendo	8.04	6.21	19.20	1	4050
12	13	453	GB	1999.0	Role-Playing	Nintendo	7.38	5.07	18.94	1	4530

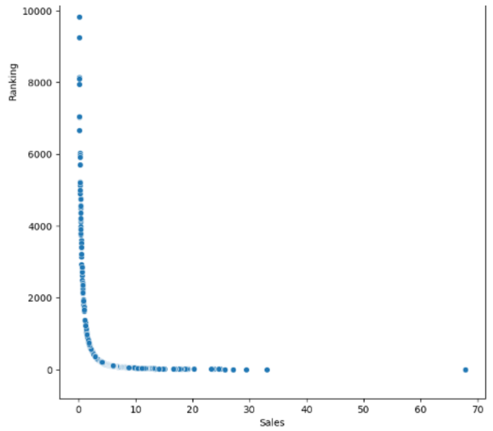
- product values are not unique to individual products. They are likely shared by several versions of a product, across several platforms (but sharing the same genre and publisher). They each have unique sales figures indicating the need to allocate each version of these likely ‘cross platform games’, their own product number, additional to the original (so as not to interfere with any significance the original product number may reveal)
- new numbers allocated as ‘ones’ digits after multiplying the original product value by ten, thus protecting the original coding system. This tested with corr() table.



No discernable relationship here between ranking and product number, but there seem to be some publisehrs that dominate in low rankings (ie more popular games). To be investigated further.

There are some anomalous gaps there, eg around 5000 (the original 500). This may be useful as it may indicate a grouping / system.

- plot variables sequentially and methodically, to identify any patterns or anomalies
  - eg “There are some anomalous gaps there, eg around 5000 (the original 500). This may be useful as it may indicate a grouping / system.”

<table>
<tr>
<td>correlation identified:
- ranking and global sales</td>
<td>


Strong correlation here between sales and ranking - this is not surprising, but has implocations:

- the higher ranked products are more desirable and sell better
- are they available at other sales outlets? Can we lower the price of some of these to draw customers in? Earn points for use with specific other prioducts? ie depends on competition, and other factors. Increase sales mean numbers or revenue?

The outlier has twice as many sales. How to cehck the reliability of it? Check publisher products only, and low ranking proucts.

Identify which publishers make the best ranking products.
</td>
<td>

- Significance of this is ranking is a global figure, which according to metadata is not determined by TG or its product sales, but a broader read of the market (although we are not told what that is, we can be reasonably certain that TG does know where the ranking data is sourced, and indeed these figures are accessible from many sources). Therefore, we can use ranking to predict global sales of TG's products.
- other implications and cautions considered (eg questioning whether sales / ranking correlations are true for all markets)
- decision to subset the dataset between low (popular) and high ranking products to better attempt to identify any distinct dynamics that are operating on them.
- Importance of selecting the correct cut-off point.

0 10 20 30 40 50 60 70
Sales

Strong correlation here between sales and ranking - this is not surprising, but has implocations:

- the higher ranked products are more desirable and sell better
- are they available at other sales outlets? Can we lower the price of some of these to draw customers in? Earn points for use with specific other prioducts? ie depends on competition, and other factors. Increase sales mean numbers or revenue?

The outlier has twice as many sales. How to cehck the reliability of it? Check publisher products only, and low ranking proucts.

Identify which publishers make the best ranking products.

Where should the cut off be? We know there is a strong alignment between Global sales and ranking (though with caution that this doesn't necessarily align with all markets ie NA =/= EU =/=Other)
</td>
</tr>
<tr>
<td>New dfs:
low ranking / high sales</td>
<td>

```
]:  1 # Make a new df which is just the low ranking / sales above 6million
    2 low_rank = sales[sales['Global_Sales']>6].copy()
    3 low_rank.corr()
```

]:

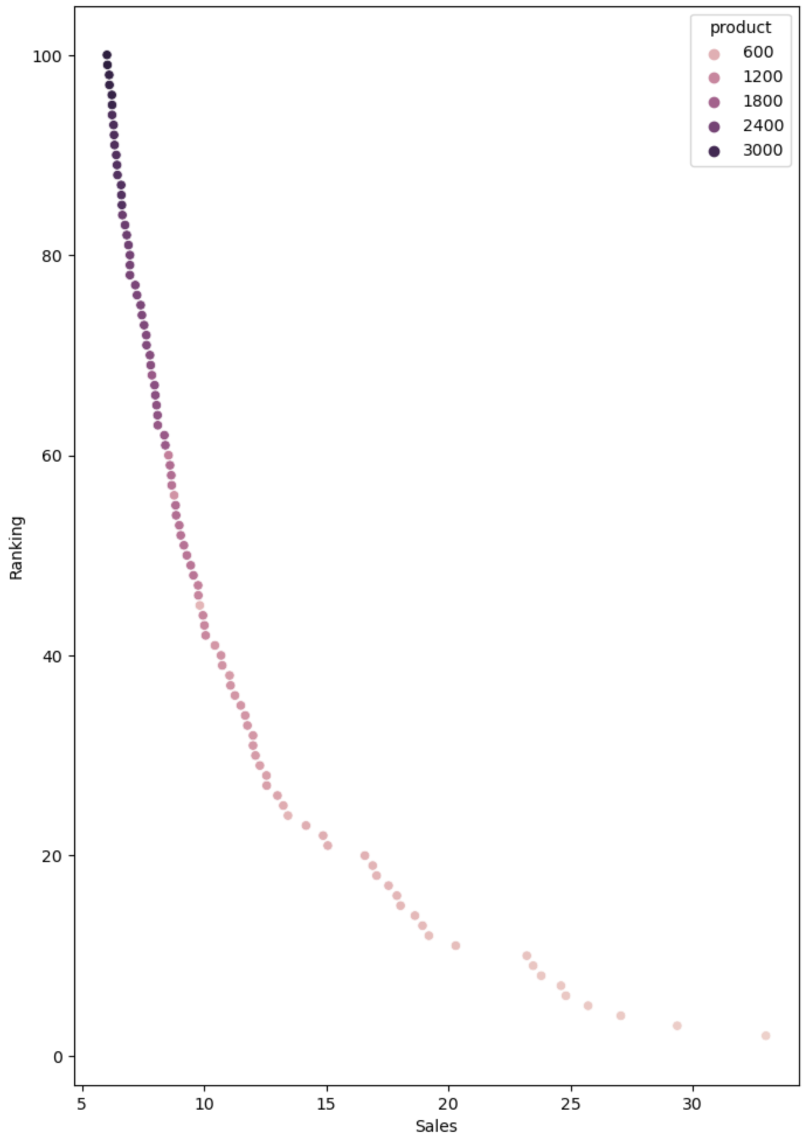
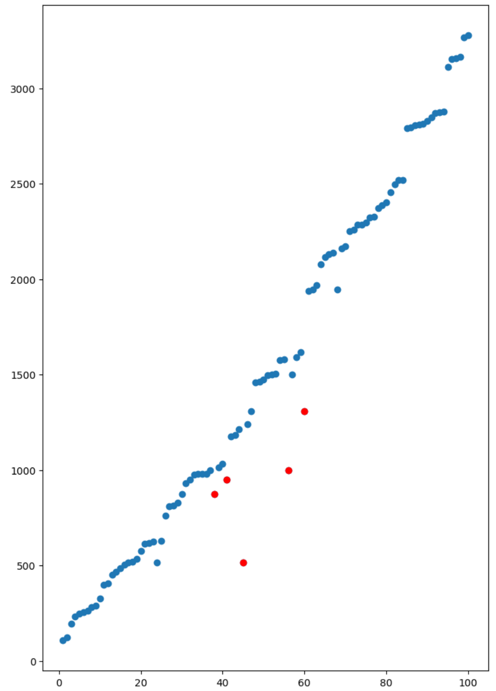
	Ranking	product	Year	NA_Sales	EU_Sales	Global_Sales	count	product2
Ranking	1.000000	0.985155	0.118663	-0.646128	-0.550839	-0.738424	0.172785	0.985161
product	0.985155	1.000000	0.081300	-0.594376	-0.524320	-0.686850	0.124346	1.000000
Year	0.118663	0.081300	1.000000	-0.321248	0.186159	-0.156577	0.381236	0.081339
NA_Sales	-0.646128	-0.594376	-0.321248	1.000000	0.569189	0.901044	-0.152323	-0.594385
EU_Sales	-0.550839	-0.524320	0.186159	0.569189	1.000000	0.823889	-0.025617	-0.524319
Global_Sales	-0.738424	-0.686850	-0.156577	0.901044	0.823889	1.000000	-0.188425	-0.686861
count	0.172785	0.124346	0.381236	-0.152323	-0.025617	-0.188425	1.000000	0.124437
product2	0.985161	1.000000	0.081339	-0.594385	-0.524319	-0.686861	0.124437	1.000000

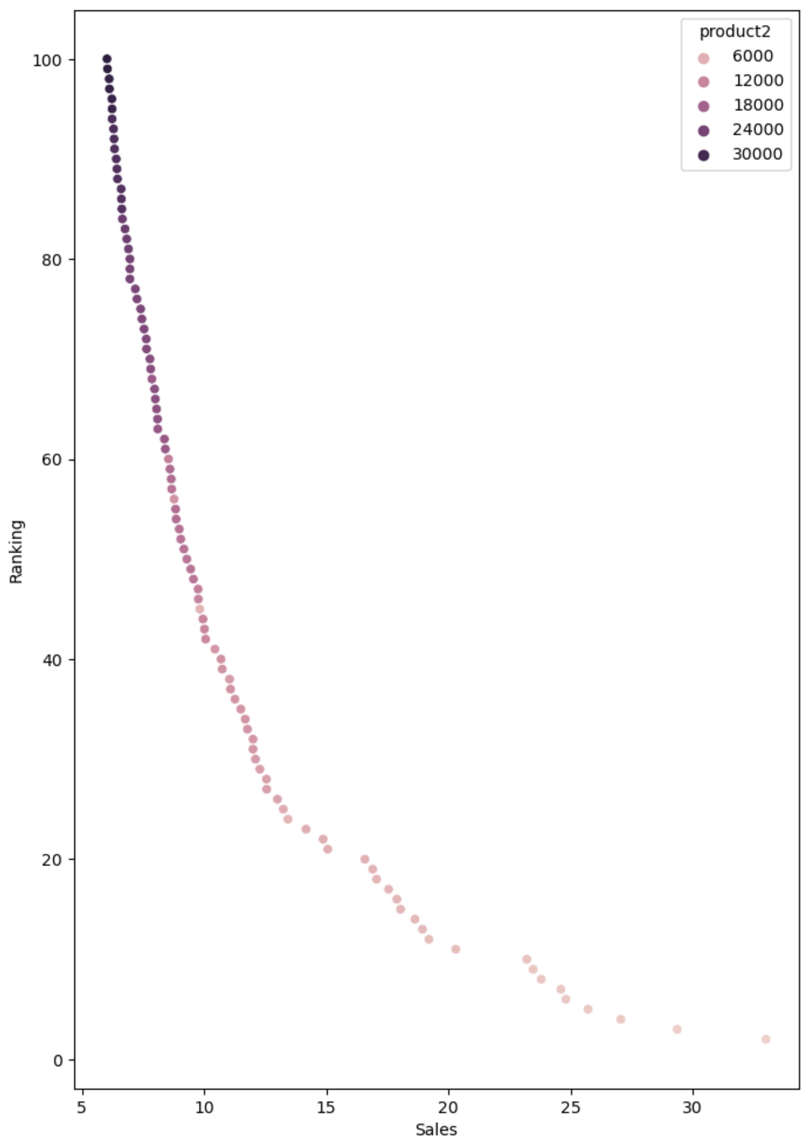
```
]:  1 # Identify the unique publishers in this dataset
    2 print(low_rank['Publisher'].unique())

['Take-Two Interactive' 'Nintendo' 'Microsoft Game Studios'
 'Sony Computer Entertainment' 'Ubisoft' 'Activision' 'Bethesda Softworks'
 'Electronic Arts' 'Sega' 'SquareSoft' 'Atari']
```
</td>
<td>

- sales outlier (Nintendo product) included because Nintendo proves to be far and beyond the most popular publisher, so it's not inconceivable that its figures are correct.
- It is a sports game and its popularity may not last, if for example it is popular because it's come out at the time of the football world cup etc.
- Ranking will inform TG of the expected sales of the product, not its existing popularity
</td>
</tr>
<tr>
<td>Best ranking (and best selling) Publishers identified</td>
<td>

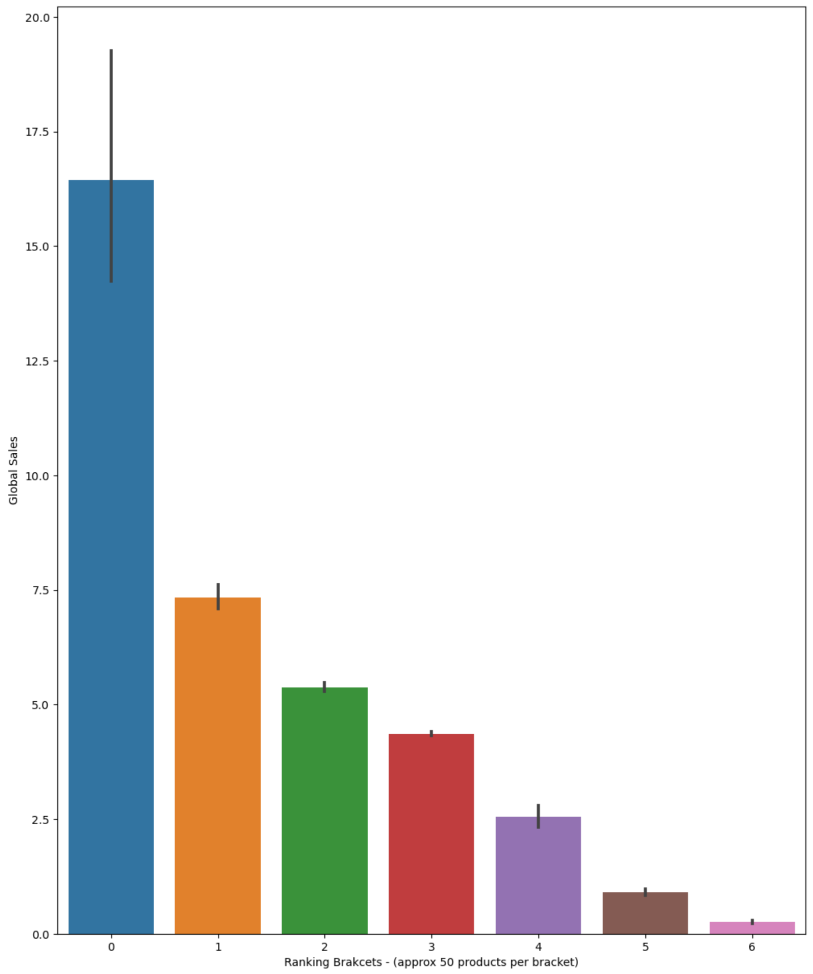
```
In [758]:  1 # Identify the unique publishers in this dataset
           2 print(low_rank['Publisher'].unique())

['Take-Two Interactive' 'Nintendo' 'Microsoft Game Studios'
 'Sony Computer Entertainment' 'Ubisoft' 'Activision' 'Bethesda Softworks'
 'Electronic Arts' 'Sega' 'SquareSoft' 'Atari']

In [763]:  1 # scatter plot
           2 plt.figure(figsize = (8, 12))
```
</td>
<td>

- Can be used by the marketing team when choosing which games to promote.
- Consideration of groups / preferences follows later in the analysis
</td>
</tr>
</table>

Product / ranking relationship identified		• not consistent across the whole dataset (limits significance) • not directly something that the client wants to know about • unsuccessful in identifying a pattern beyond this subset of the data, though this was attempted (for example see below, where an attempt was made to identify the dynamic by separating the outliers in the relationship between ranking and sales 

		• confirming that the product2 code has not interfered with any pattern we might discover

<table>
<tr>
<td>Ranking brackets created to make the relationship clearer to see (ie simpler</td>
<td>



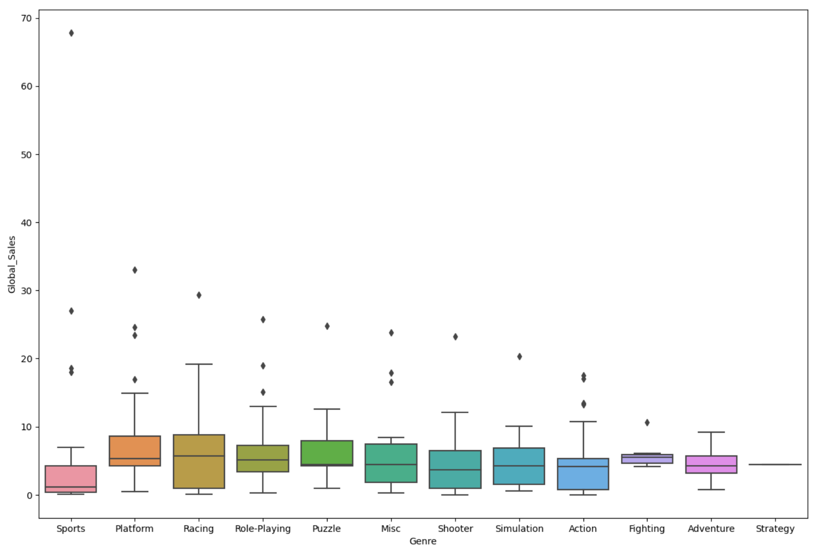
</td>
<td>

Counts in brackets are all very similar, validating the significance of the low ranking group as double that of the next one.

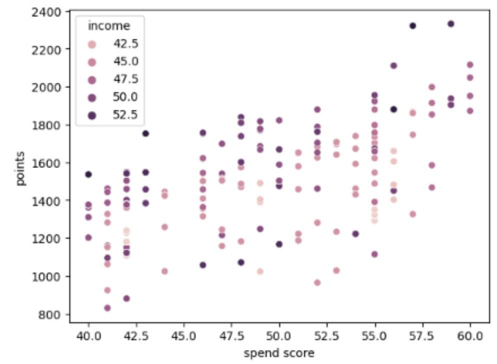
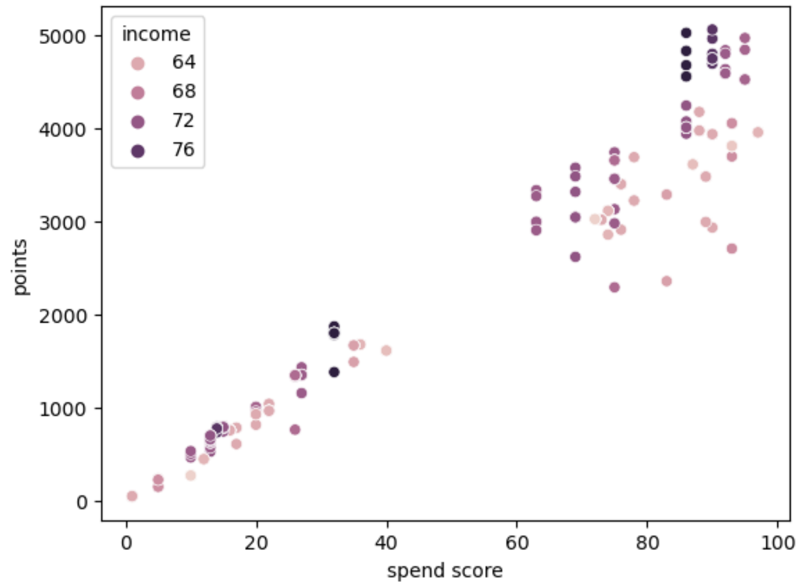
```
# Bracket the data by Ranking and replot, to show how the best ranking products dominate sales.

# Define the number of brackets
# num_brackets = 50
# num_brackets = 18
num_brackets = 7
# Create the new column
sales['Ranking_Bracket'] = pd.qcut(sales['Ranking'], q = num_brackets, labels = False)

# Count the number of occurences in each ranking bracket.
ranking_counts = sales['Ranking_Bracket'].value_counts().sort_index()
print(ranking_counts)
```

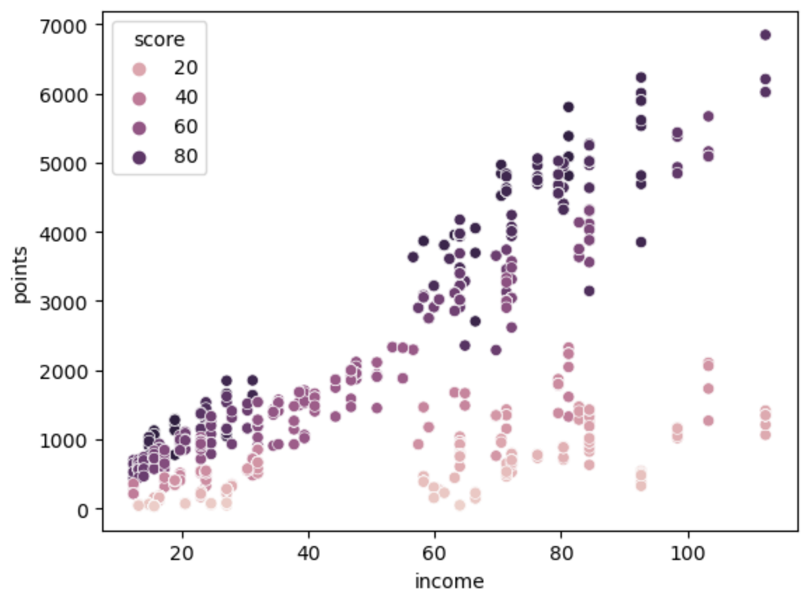
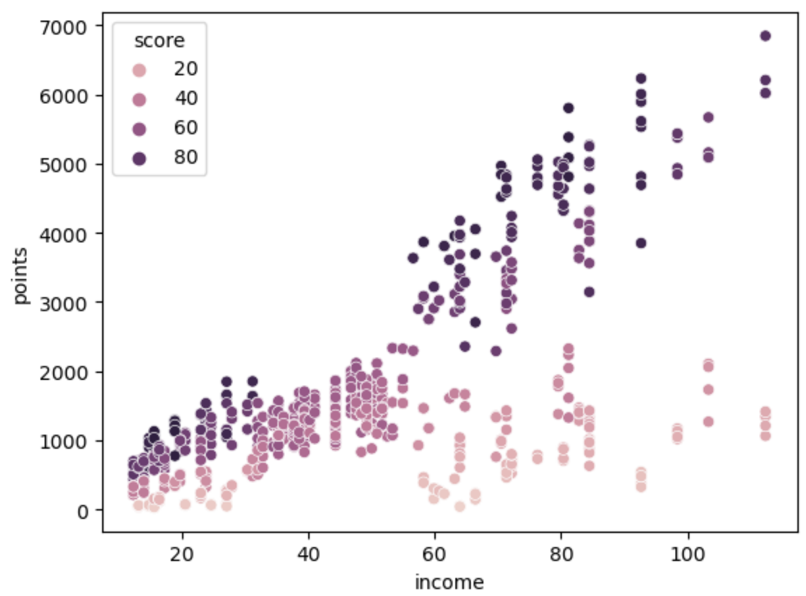
```
0    51
1    50
2    50
3    50
4    50
5    50
6    51
Name: Ranking_Bracket, dtype: int64
```

</td>
</tr>
<tr>
<td></td>
<td>

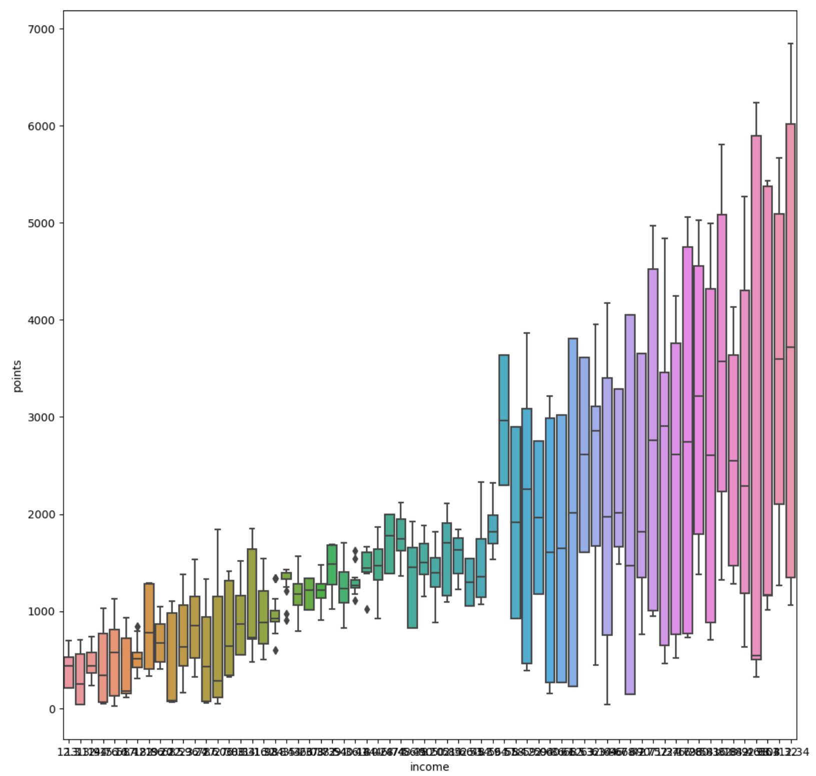
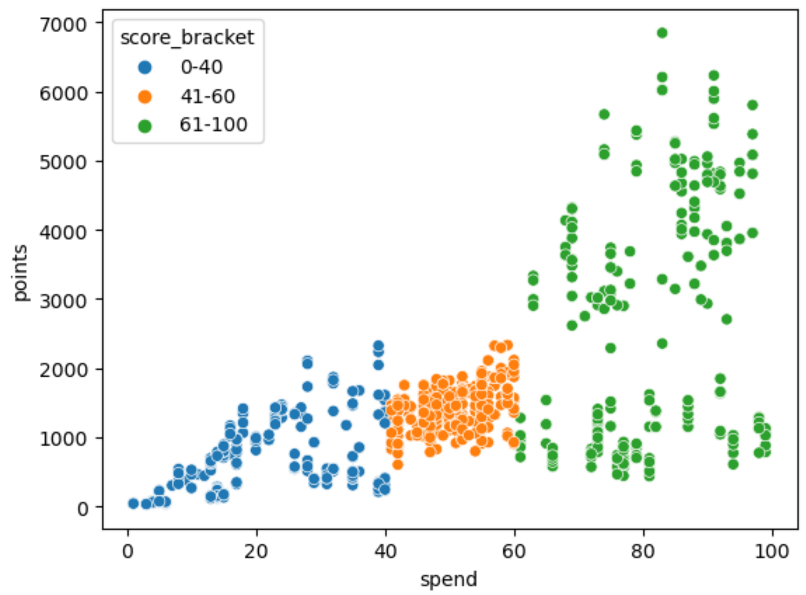


</td>
<td>We won't include genre in the presentation because it doesn't have a big impact on sales - the mean here is not very different across all the genres. However it is worth noting that sports genre has a lower than normal mean, in spite of the outlier success of the top selling game - indicating that that one game is even more of an anomaly, and indeed if its success is due to a contextual reason such as a work series event, we can expect that to drop. However we checked the date of this product and determined that it's from 2006, and not a 'late model' even in terms of this slightly dated dataset.</td>
</tr>
</table>

identified the anomaly in the score / income / points relationship	[1059]: `# scatter plot`
	`sns.scatterplot(data = income_41_to_56, x = 'score', y = 'loyalty_points', hue = 'income')`
	`plt.xlabel('spend score')`
	`plt.ylabel('points')`
	:[1059]: `Text(0, 0.5, 'points')`
	
	In [ ]:
	
		Anomaly: customers in a specific income band (40-55k) can only access a spend score of between 40-60. This in turn affects their access to loyalty points, because loyalty points are mostly a product of income and score. Furthermore certain incomes can't access spend scores between about 40-60

<table>
<tr>
<td></td>
<td>
</td>
<td>This is what the general pattern looks like - except there are gaps where those mid-scores should be because we’ve taken them out. When we put them back in….</td>
</tr>
<tr>
<td></td>
<td>
</td>
<td>You can see that there’s strange behvaviour in that bracket. It’s caused by a programming error or if it’s deliberate it would be good to know why!

I attempted to see if it was affecting purchasing behaviour but if so it’s beyond me how.</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
</td>
<td>This is the same data on a box plot, showing the strange dynamic</td>
</tr>
<tr>
<td>groups aggregated across clusters, based on score</td>
<td>
</td>
<td>to attempt to find out what determines score, by then exploring the data for commonalities or patterns within each cluster bracket</td>
</tr>
<tr>
<td>NLP: smaller (original) dataset used
(DF3)</td>
<td>Requirement already satisfied: colorama in c:\users

[364]:
# Load the data set as df3.
DF3=pd.read_csv("reviews.csv")

# View DataFrame.
DF3.head()

t[364]:
<table><tr><th></th><th>gender</th><th>age</th><th>income</th><th>score</th><th>points</th><th>education</th><th>product</th></tr><tr><td>0</td><td>Female</td><td>37</td><td>15.58</td><td>99</td><td>1067</td><td>postgraduate</td><td>1497</td></tr></table></td>
<td>client wants to know about the most neg and most pos reviews, and this can be achieved with the dataset
any concatenation can be conducted later by joining the new columns (sentiment score etc) to the larger, normalised df, because the latter is made or duplicating rows from the original, and they share many similar columns.
the larger dataframe would have been very slow to run the NLP on.</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>

2. Prepare the data for NLP

2a) Change to lower case and join the elements in each of t

In [348]:
```
# Review: Change all to lower case and join with a space.
DF3['review'] = DF3['review'].apply(lambda x: " ".join(x.lowe
```

In [349]:
```
# Review: Change all to lower case and join with a space.
DF3['summary'] = DF3['summary'].apply(lambda x: " ".join(x.lo
```

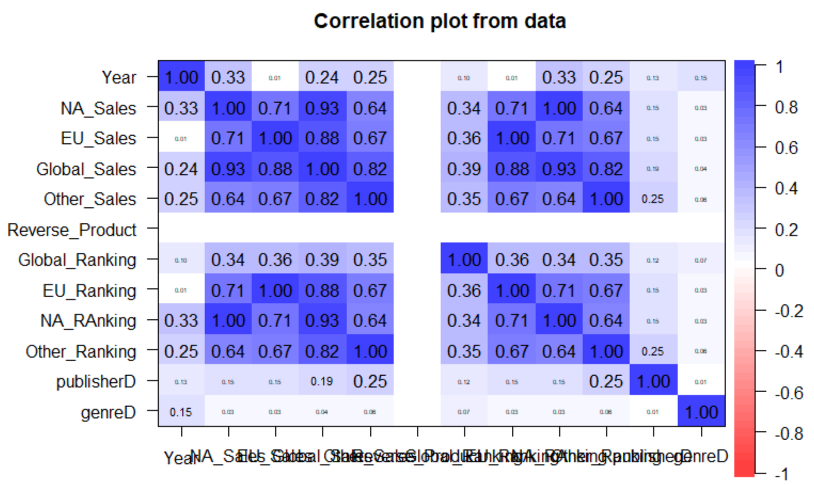
In [350]:
```
DF3.head()
```

Out[350]:

level_0	index	gender	age	income	score	points	education	product

nice

</td>
<td></td>
</tr>
<tr>
<td>Forecasting:</td>
<td>



</td>
<td>

- character variables converted to dummies
- correlations determined
- manipulation of age and ranking to make the correlations positive.
- summary() method used to determine usefulness of variables, and the usefulness of the models, eg R-means adjusted, t-score and p-values

</td>
</tr>
</table>