CVRD Mask

June 19, 2020

1 CVRD MASK MARKET RESEARCH (Amazon.com)

1.1 By Julian Murillo

1.1.1 Julian@cvrdmask.org

1.1.2 Notes :

- The Data was scraped directly off of Amazon's Website using ParseHub. This is a scrape of first 10 pages with the most relevant results when using the search "Face mask for Virus" and "Childrens face mask".

- I had to create binary dummy variables (1 = positive, 0 = negative) for the variables 'childrens_mask', 'ear_loops', 'mask_pack', and 'reusable'.

- Example... if it was a childrens_mask there would be a 1, if it was an adult mask there would be a 0. Same for the other binary variables.

- Also, just ignore the code. Look only for the visuals and the captions/explanations at the end.

[59]:
```
# Import necessary packages



# General
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.preprocessing import StandardScaler #Z-score variables

# Cross-Validation
from sklearn.model_selection import KFold # k-fold cv
from sklearn.model_selection import LeaveOneOut #LOO cv
from sklearn.model_selection import cross_val_score # cross validation metrics
```

```
from sklearn.model_selection import cross_val_predict # cross validation metrics


# Decision Tree
from sklearn.tree import DecisionTreeClassifier


# Naive Bayes Algorithm
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB,␣
↪CategoricalNB


# Classification Model Performance
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import plot_confusion_matrix

# Correlation Matrix
import seaborn as sn


#Read-in data and import packages


# LOGISTIC Regression
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score


# LINEAR REGRESSION
from sklearn.linear_model import LinearRegression # Linear Regression Model
from sklearn.metrics import mean_squared_error, r2_score #model evaluation
import statsmodels.api as sm # Inferential
import statsmodels.formula.api as smf # Inferential

from sklearn.model_selection import train_test_split # simple TT split cv

# plots
from matplotlib import pyplot
import joypy


# Set Seed
import random
random.seed(1968)
```

```
%precision %.7g
```

[59]:
```
'%.7g'
```

[3]:
```
# "Face mask for virus" general search results
mask_df = pd.read_csv("/Users/julianjr./Desktop/CVRD Mask/AmazonScrape7 copy.
 ↪csv")
mask_df.head(5)
```

[3]:

	product_name	product_url	product_price	product_rating	product_shipping	product_reviews	childrens_mask	z_score_price	ear_loops	mask_pack	reusable
0	Buttonsmith White Adult Cotton Face Mask - Two…	https://www.amazon.com/Buttonsmith-White-Adult…	14.0	3.9	FREE: Orders over $25	22	0	-0.276143	0	0	1
1	Summer Face Mask Balaclava Protection from Dus…	https://www.amazon.com/Self-Pro-Balaclava-Prot…	14.0	4.5	FREE: Orders over $25	4940	0	-0.276143	0	0	0
2	Milcoast 3-Ply Layer Disposable Earloop Face M…	https://www.amazon.com/Milcoast-3-Ply-Layer-Di…	34.0	3.9	FREE Shipping by Amazon	239	0	0.854489	1	1	0
3	2 Pack Sponge Face Cover with Breathing Valve,…	https://www.amazon.com/Sponge-Cover-Breathing-…	19.0	1.9	FREE Shipping	62	0	0.006515	0	1	1
4	3 Pack Cloth Bandana | Face Mask for Dust & Su…	https://www.amazon.com/Cloth-Bandana-Protectio…	NaN	2.6	FREE: Orders over $25	38	0	NaN	0	1	0

[4]:
```
# "childrens face mask" search results

childrens_df = pd.read_csv("/Users/julianjr./Desktop/CVRD Mask/ChildrenMask␣
 ↪copy.csv")
childrens_df.head(5)
```

[4]:

	product_name
0	Cotton Unisex Face Shield Reusable for Cycling…
1	Cotton Face Bandanas, for Children, Haze Dust …

```
2  EXTSUD Dustproof Sunhat Cotton Packable Sun Ha…
3  Masks for children and adults-ideal (1 pack 2 …
4  30 pieces Mixed Color Elastic Ear Loops School…

                                        product_url  product_price  \
0  https://www.amazon.com/Cotton-Unisex-Reusable-…          12.0
1  https://www.amazon.com/Cotton-Bandanas-Animal-…           8.0
2  https://www.amazon.com/EXTSUD-Dustproof-Packab…          16.0
3  https://www.amazon.com/Masks-children-adults-i…          19.0
4  https://www.amazon.com/pieces-Mixed-Color-Elas…          25.0

   product_rating       product_shipping  product_reviews  childrens_mask  \
0             3.5  FREE: Orders over $25              62               1
1             2.0                 Charge              67               1
2             4.3  FREE: Orders over $25             148               1
3             3.9  FREE: Orders over $25              24               1
4             5.0           FREE Shipping              3               1

   z_score_price  ear_loops  mask_pack  reusable
0      -0.466446          0          1         1
1      -0.940741          0          1         0
2       0.007850          0          1         0
3       0.363571          0          1         0
4       1.075015          1          0         0
```

[5]:
```
print(f'General Mask Search data set has dimensions of {mask_df.shape} \nThe␣
 ↪Childrens Mask Search has dimensions of {childrens_df.shape}')

# mask_df.shape
# childrens_df.shape
# print(mask_df["Product_name"])
```

```
General Mask Search data set has dimensions of (338, 11)
The Childrens Mask Search has dimensions of (306, 11)
```

[6]:
```
# Merge Data Frames vertically
full_df = mask_df.append(childrens_df, ignore_index=True)
full_df.tail()
```

[6]:
```
                                        product_name  \
639  Unigear Full Face Snorkel Mask, Snorkeling Mas…
640  2 Pack Kids Neck Gaiters, Unisex Face Scarf Fo…
641  Weddingstar Protective Face Mask Travel Bag w/…
642  Aegend 2 Pack Fleece Neck Warmer for Kids (Age…
643  LHWY Multifunction Sport Scarves for Women Men…

                                          product_url  product_price  \
```

```
639  https://www.amazon.com/Unigear-Snorkel-Safety-…          30.0
640  https://www.amazon.com/Gaiters-Unisex-Outdoors…          17.0
641  https://www.amazon.com/Weddingstar-Protective-…           5.0
642  https://www.amazon.com/aegend-Pack-Fleece-Warm…          19.0
643  https://www.amazon.com/LHWY-Multifunction-Anti…          17.0

     product_rating         product_shipping  product_reviews  childrens_mask  \
639             4.7  FREE Shipping by Amazon             148               1
640             5.0    FREE: Orders over $25              15               1
641             NaN    FREE: Orders over $25               0               1
642             4.5    FREE: Orders over $25              80               1
643             3.3                   Charge               9               1

     z_score_price  ear_loops  mask_pack  reusable
639       1.667884          0          0         0
640       0.126424          0          1         0
641      -1.296463          0          0         0
642       0.363571          0          1         0
643       0.126424          1          1         1
```

[89]:
```
# Checking to see what kind of variables Python thinks we're working with
full_df.dtypes
```

[89]:
```
product_name         object
product_url          object
product_price       float64
product_rating      float64
product_shipping     object
product_reviews       int64
childrens_mask        int64
z_score_price       float64
ear_loops             int64
mask_pack             int64
reusable              int64
dtype: object
```

[8]:
```
# Check to see if we have missing values in our data

full_df.isnull().sum()
# We've got 88 missing values in product price
# Usually because of the "click for price" option in Amazon
```

[8]:
```
product_name          0
product_url           0
product_price        88
product_rating      144
product_shipping      0
```

```
product_reviews       0
childrens_mask        0
z_score_price        88
ear_loops             0
mask_pack             0
reusable              0
dtype: int64
```

[9]:
```
# Drop missing values
full_df = full_df.dropna()
full_df.shape
```

[9]:
```
(446, 11)
```

1.1.3 ——————————- Summary Statistics of Numeric Columns ——————————

[10]:
```
full_df.describe()
```

[10]:

	product_price	product_rating	product_reviews	childrens_mask	\
count	446.000000	446.000000	446.000000	446.000000	
mean	16.952915	3.899776	118.719731	0.540359	
std	12.310406	0.843015	484.310899	0.498928	
min	1.000000	1.000000	0.000000	0.000000	
25%	10.000000	3.400000	4.000000	0.000000	
50%	15.000000	4.000000	17.000000	1.000000	
75%	19.000000	4.500000	62.000000	1.000000	
max	174.000000	5.000000	6587.000000	1.000000	

	z_score_price	ear_loops	mask_pack	reusable
count	446.000000	446.000000	446.000000	446.000000
mean	-0.028810	0.147982	0.594170	0.405830
std	0.928858	0.355481	0.491603	0.491603
min	-1.770758	0.000000	0.000000	0.000000
25%	-0.585019	0.000000	0.000000	0.000000
50%	-0.219612	0.000000	1.000000	0.000000
75%	0.232641	0.000000	1.000000	1.000000
max	8.768916	1.000000	1.000000	1.000000

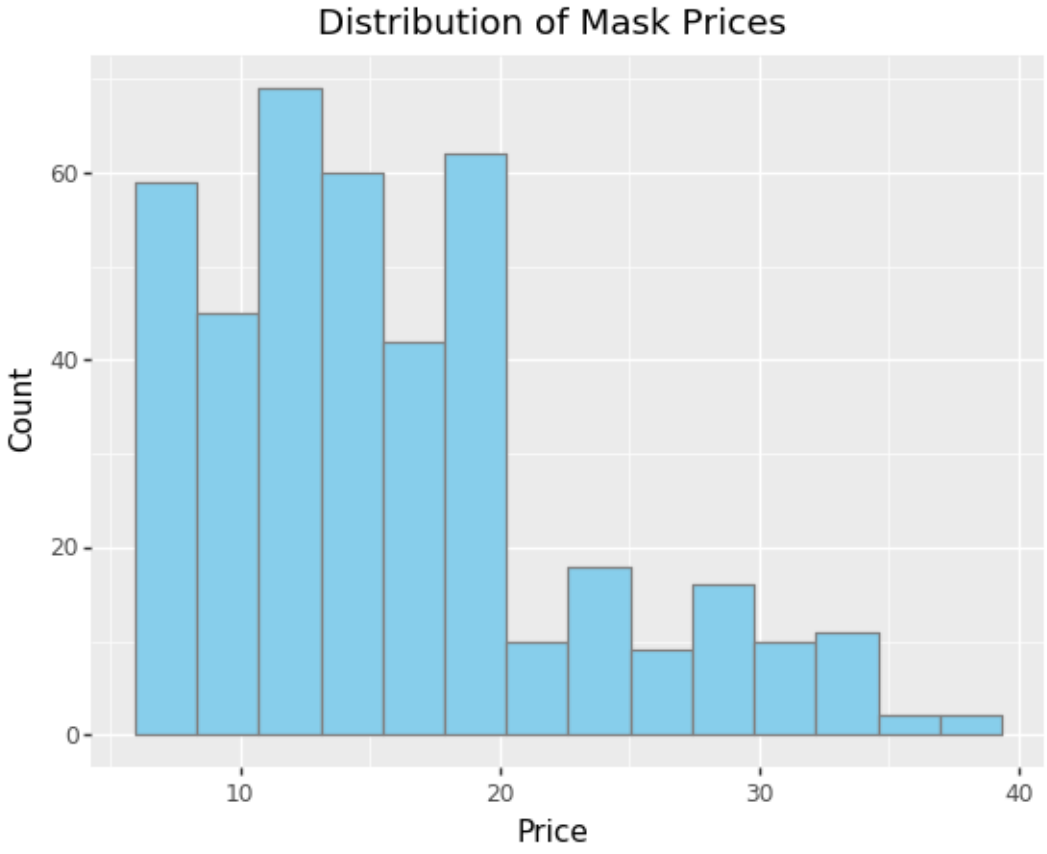
[11]:
```
# Remove Price Outliers
# Initially I was going to use z-scores... but decided to use quantiles instead
# Remove rows who's price points are above the 97% quantile and below the 3␣
 ↪quantile. These are probably errors.
max_thresh = full_df['product_price'].quantile(.97)
min_thresh = full_df['product_price'].quantile(.03)
```

[12]:
```
full_df = full_df[(full_df['product_price'] < max_thresh) &␣
↪(full_df['product_price'] > min_thresh)]
full_df.shape
```

[12]: (415, 11)

1.2 Explore Data: Visualizations

[13]:
```
# Histogram... no outliers
(ggplot(full_df, aes(x = 'product_price')) + geom_histogram(fill = 'skyblue',␣
↪color = 'gray') +
labs(x = 'Price', y = 'Count', title = 'Distribution of Mask Prices'))
```



[13]: <ggplot: (8775265268577)>

[92]:
```
print(f"Is skewed right and not normally dsitrubuted with a mean around:␣
↪{full_df['product_price'].mean()}")
```

```
Is skewed right and not normally dsitrubuted with a mean around:
15.778313253012048
```

1.2.1 Caption:

Keep prices around this mean (15.79) to stay competitive... on Amazon

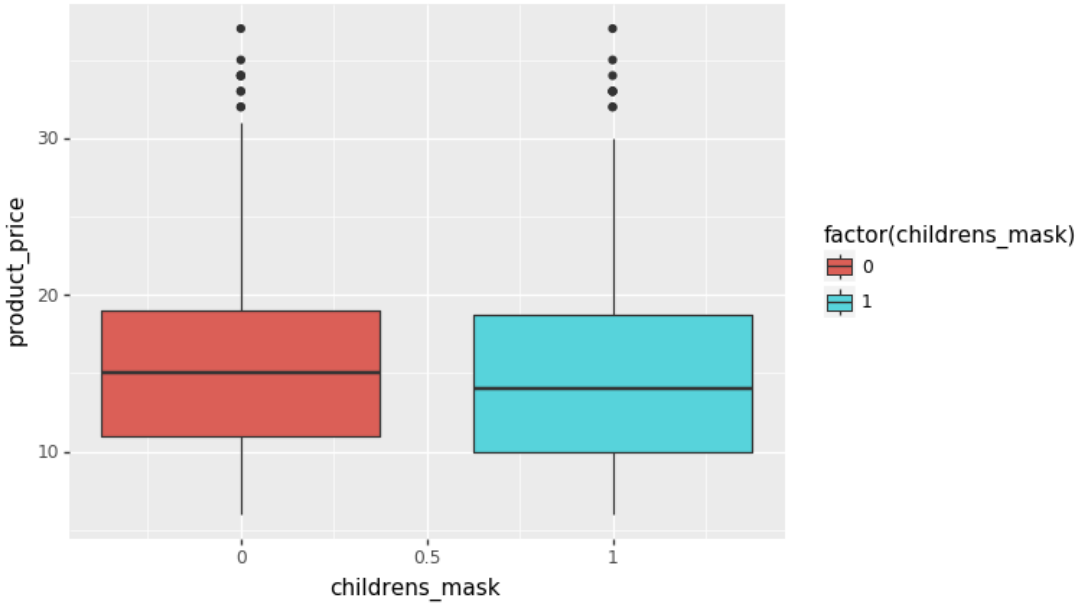
[14]:
```
(ggplot(full_df, aes(x = "product_price", y = "product_rating", color =␣
↪'product_shipping')) + geom_point() + stat_smooth(method='lm') +
facet_wrap("product_shipping") + labs(title = "Price Effect Rating?"))
```

[14]:
```
<ggplot: (8775265411357)>
```

1.2.2 Caption:

Price doesn't look to have any effect on rating, however shipments fulfilled by Amazon have much better ratings on average (lower left corner)

[15]:
```
(ggplot(full_df, aes(x = "childrens_mask",
                     y = "product_price", fill = "factor(childrens_mask)"))␣
↪+ geom_boxplot())
```

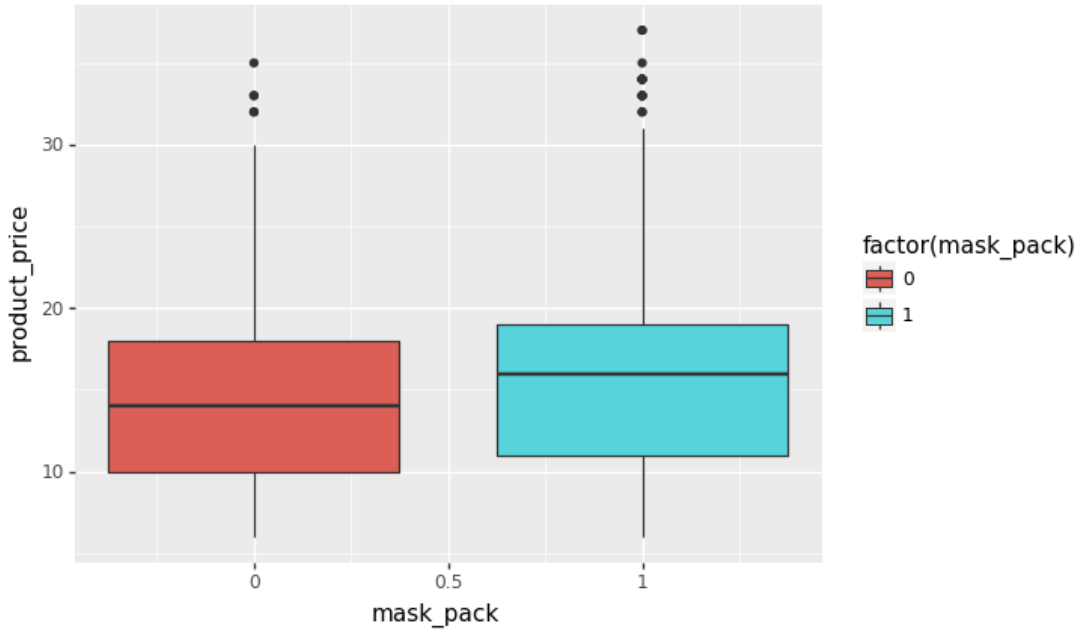


[15]: <ggplot: (8775266324061)>

1.2.3 Caption:

Not a crazy difference (~ 2 dollars) in price points for Childrens mask... which is good, high prices drive profit

[89]:
```
(ggplot(full_df, aes(x = "mask_pack",
                     y = "product_price", fill = "factor(mask_pack)")) +␣
↪geom_boxplot())
```

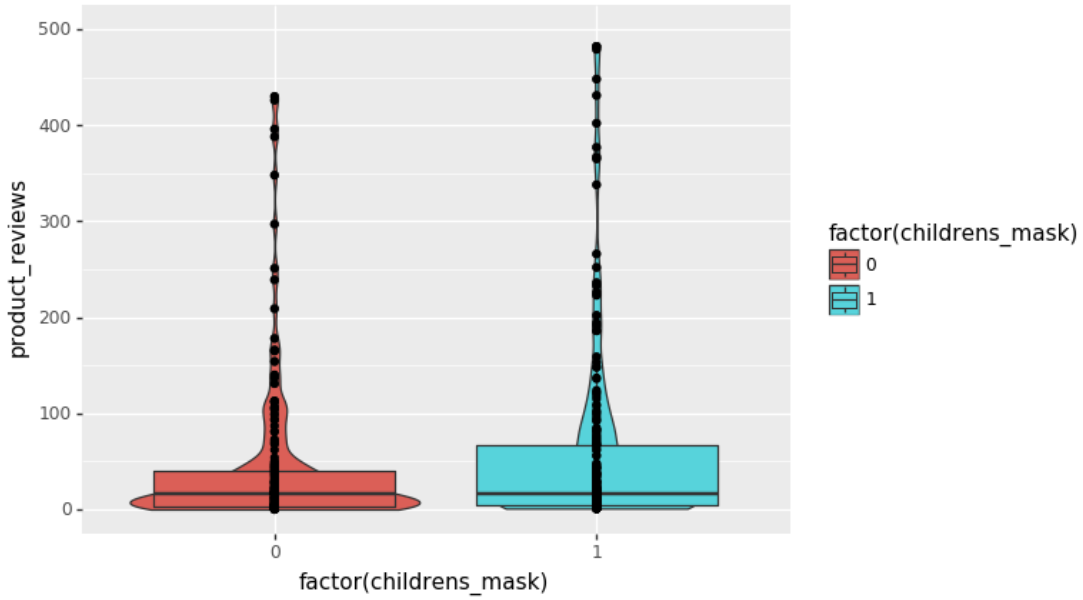


[89]: <ggplot: (8792153473529)>

1.2.4 Caption:

Packs of masks are typically more expensive... something we might expect

[90]:
```
(ggplot(full_df, aes(x = "factor(childrens_mask)", y = "product_reviews")) +
geom_violin(aes(fill = "factor(childrens_mask)")) +
geom_boxplot(aes(fill = "factor(childrens_mask)")) +
geom_point() +
 scale_y_continuous(limits = (0,500)))
```



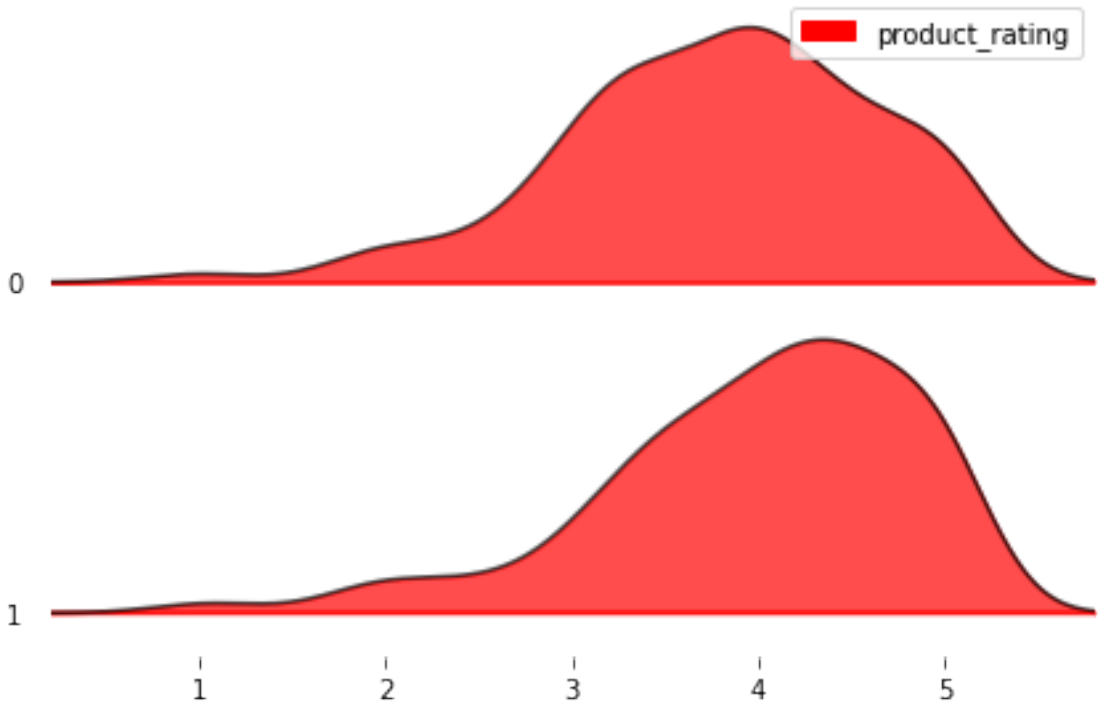
[90]: <ggplot: (8792147737921)>

1.2.5 Caption:

Not a massive difference in the number of reviews of childrens masks vs adult masks. But still some difference. There is a greater proportion of high-number-of-reviews for childrens masks vs adult masks. A violin plot shows not only the spread of the data in, but also the density around certain areas (median for instance)... the black line in the middle.

[112]:
```
joypy.joyplot(data = full_df, by = 'childrens_mask',
              column = ['product_rating'],
              color = ['r'],
              alpha = .7,
              legend = True,
             overlap = False)
```

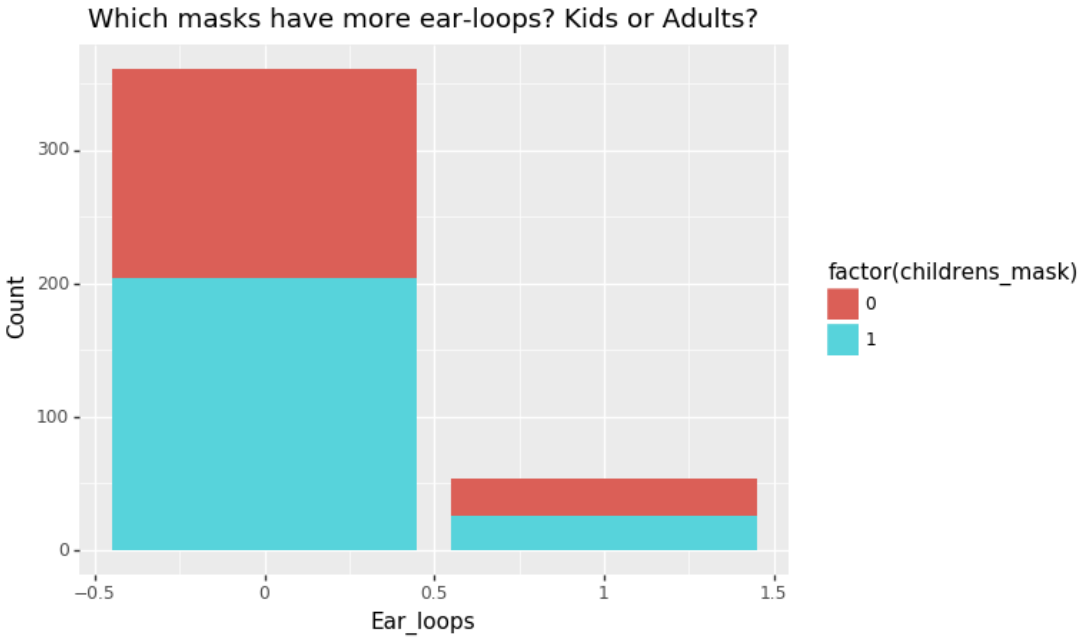
[112]:
```
(<Figure size 432x288 with 3 Axes>,
 [<matplotlib.axes._subplots.AxesSubplot at 0x7ff152f18710>,
  <matplotlib.axes._subplots.AxesSubplot at 0x7ff15544bc90>,
  <matplotlib.axes._subplots.AxesSubplot at 0x7ff155456d90>])
```



1.2.6 Caption:

Ratings for the masks (kids vs adults) are about the same as well... the y-axis here is population density (of the childrens mask 1, and adult mask 0)

[94]:
```
# Bar-Chart
(ggplot(full_df, aes(x = 'ear_loops')) + geom_bar(aes(fill =␣
 ↪'factor(childrens_mask)')) +
labs(x = 'Ear_loops', y = 'Count', title = 'Which masks have more ear-loops?␣
 ↪Kids or Adults?'))
```



[94]: <ggplot: (8792154375945)>

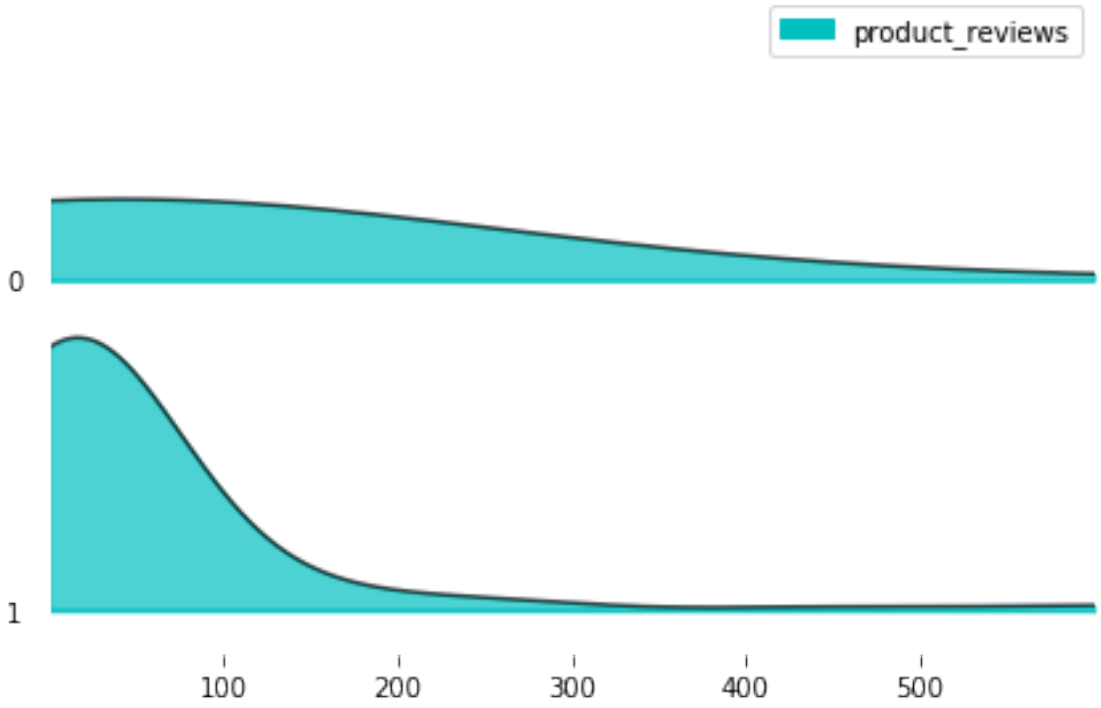
1.2.7 Caption:

Proportionally, it's about a 50/50 split in regard to which masks have more earloops (kids vs adults).

Also, because the item descrptions are formatted so terribly, it was hard to systematically identify which masks had loops and which didn't. I searched to see if the item descriptions contained "oops".

[113]:
```
joypy.joyplot(data = full_df, by = 'reusable',
              column = ['product_reviews'],
              color = ['c'],
              alpha = .7,
              legend = True,
              x_range=[0,600],
             overlap = False)
```

[113]:
```
(<Figure size 432x288 with 3 Axes>,
 [<matplotlib.axes._subplots.AxesSubplot at 0x7ff155424110>,
  <matplotlib.axes._subplots.AxesSubplot at 0x7ff155d75690>,
  <matplotlib.axes._subplots.AxesSubplot at 0x7ff155cc6c50>])
```



1.2.8 Caption:

Intersetingly, the disposable masks seem to have greater numbers of product reviews. Which isa bummer, means that Amazons algorithm will now tend to favor disposable masks in regard to reviews.

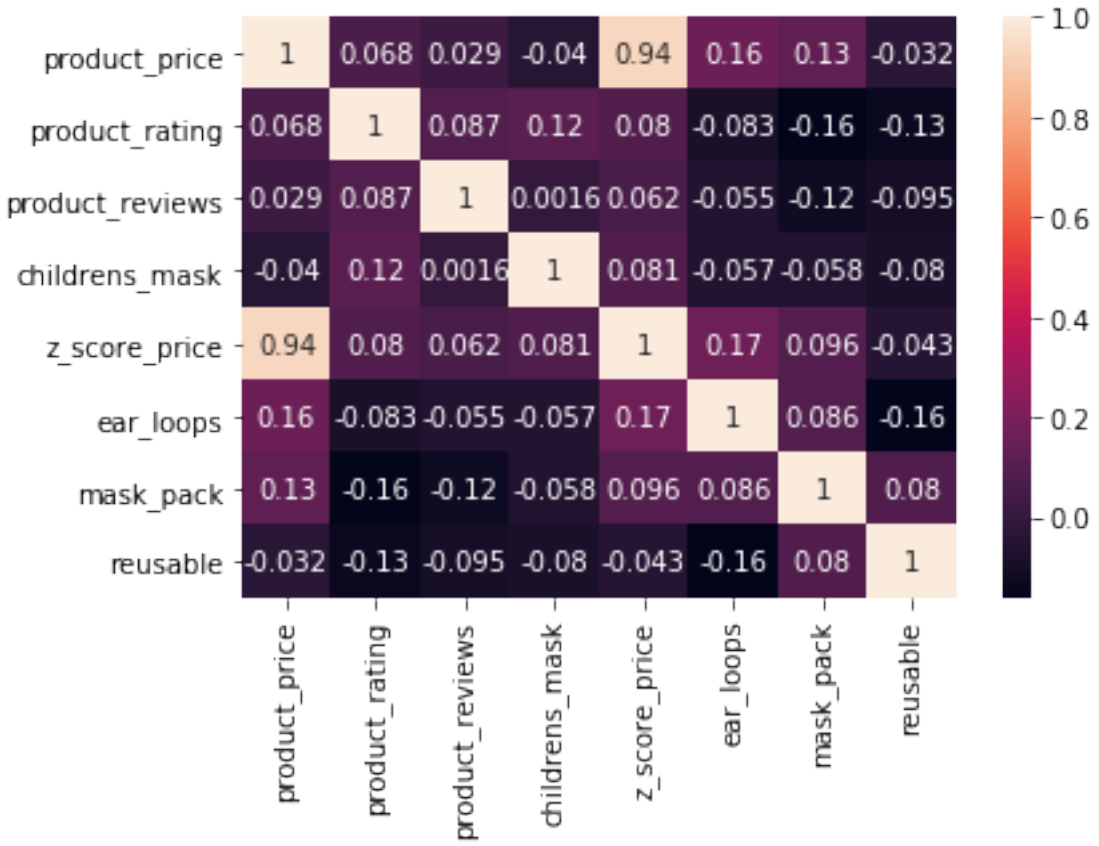
1.3 ————————- Begin Regression Analysis: Ordinary Least Squares (OLS)————————

[91]:
```
# Quick correlation matrix to verify Gauss Markov Assumptions

corrMatrix = full_df.corr()
sn.heatmap(corrMatrix, annot=True)

# Perfect... none of the predictor variables are super correlated with one␣
↪another
```

[91]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb26b444ad0>
```



[51]:
```
full_df.columns
```

[51]:
```
Index(['product_name', 'product_url', 'product_price', 'product_rating',
       'product_shipping', 'product_reviews', 'childrens_mask',
       'z_score_price', 'ear_loops', 'mask_pack', 'reusable'],
      dtype='object')
```

1.3.1 Inferential Regression Model

[68]:
```
lr_model = smf.ols(formula = "product_reviews ~ product_price + product_rating␣
↪+ childrens_mask + ear_loops + mask_pack + reusable" , data = full_df)
```

[69]:
```
# Model Summary
output1 = lr_model.fit()
output1.summary()
```

[69]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:        product_reviews   R-squared:                       0.030
Model:                            OLS   Adj. R-squared:                  0.016
Method:                 Least Squares   F-statistic:                     2.099
Date:                Fri, 19 Jun 2020   Prob (F-statistic):             0.0524
Time:                        11:39:18   Log-Likelihood:                -3161.9
No. Observations:                 415   AIC:                             6338.
Df Residuals:                     408   BIC:                             6366.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
==
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--
Intercept         73.3810    139.214      0.527      0.598    -200.286
347.048
product_price      3.1725      3.564      0.890      0.374      -3.833
10.178
product_rating    31.6528     30.182      1.049      0.295     -27.678
90.984
childrens_mask   -19.7341     49.665     -0.397      0.691    -117.366
77.898
ear_loops        -95.9198     75.113     -1.277      0.202    -243.577
51.737
mask_pack       -103.9034     51.081     -2.034      0.043    -204.319
-3.488
reusable         -92.5366     51.148     -1.809      0.071    -193.084
8.011
==============================================================================
Omnibus:                      647.791   Durbin-Watson:                   1.907
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           144760.667
Skew:                           8.608   Prob(JB):                         0.00
Kurtosis:                      92.863   Cond. No.                         104.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

1.3.2 Predictive Regression Model

[73]:
```
predictors = ['product_price','product_rating','childrens_mask', 'ear_loops',␣
↪'mask_pack', 'reusable']
X = full_df[predictors]
Y = full_df["product_reviews"]

model2 = LinearRegression()
model2.fit(X,Y)
```

[73]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

[86]:
```
# Generate Predictions..
reviews_pred = model2.predict(X)
reviews_pred[4:12]
```

[86]:
```
array([263.39877477,  19.45799899, 111.19167167,  13.12743275,
       209.54575047, 123.36855261,   6.79686651, 115.19586309])
```

[88]:
```
# Very high mean squared error lol

mean_squared_error(Y, reviews_pred)
```

[88]:
```
242817.53264493175
```

[77]:
```
r2_score(Y,reviews_pred)
```

[77]:
```
0.02994181589469702
```

[79]:
```
coefficients = pd.DataFrame({"Coef":model2.coef_,
              "Name": predictors})
coefficients = coefficients.append({"Coef": model2.intercept_,
                    "Name": "intercept"}, ignore_index = True)
coefficients
```

[79]:

	Coef	Name
0	3.172485	product_price
1	31.652831	product_rating
2	-19.734067	childrens_mask
3	-95.919834	ear_loops
4	-103.903352	mask_pack
5	-92.536572	reusable
6	73.380961	intercept

Interstingly, the two things that seems to drive the number of reviews a product gets is the price and (obviously) the product rating. If you increase the product price by

1 dollar, you will expect to see around 3 more reviews on average. Similarly, if you increase the product rating by 1 star, you can expect to see an increase of about 31 reviews on average.

Also, childrens masks, ear-loops, packs of masks, and reusable masks were inversely (negatively) correlated with the number of reviews a product had.

Overall, these variables combined account for about 3% of the variation in number of reviews... not a very solid estimate (pretty bad actually). We would need much more information/many more variables to generate a predictive model to really see what drives Amazon reviews.

1.4 Conclusion:

1.4.1 The higher the price is on your product, the more likely it is to get reviews, also if your product has more positive ratings, then it will tend to generate more product reviews. Children's masks, Earloops, and reusable material generally don't increase the number of reviews your product gets.