Market Research for Custom Digital Prints on Etsy

Looking into current market for custom digital art that features minimalist drawings of customer photos

Examples of customer photos used in this work include couple photos for anniversaries and friend group photos/sibling photos for birthdays. Although this research is not extensive, I can still get a glimpse of the current market for this product. To conduct this work, I refrenced advice given by content creator Creative Hive in her video, How to Price Art Prints (https://youtu.be/d0KJnkw9U2E). Specific notes followed are listed at the end of this notebook.

Last Updated: March 22, 2023

Etsy Artists

Sampled five artists on Etsy that feature similar artwork. Took note of their pricing, production time, delivery time, sales, reviews and shop policies.

Notes on descriptions:

- My opinion on product quality (consider colors, details/texture vs price): wow!!, wow!, wow, meh
- Bestseller: Item has had a high sales volume over the past 6 months
- Etsy's Pick: Selected by Etsy's style and trend editors
- Star seller: The seller consistently earned 5-star reviews, shipped on time, and replied quickly to any messages they received
- Rave reviews: Average review rating is 4.8 or higher
- Descriptions say "one person", "two people", "more than two", etc. to refer to the amount of people featured in the drawings. Descriptions with "with eyes/eyes" and "background" mean the artist offers drawings that feature the eyes or background in the customer's photo.

EmmaDigitalStudio (wow!) *bestseller

- $13.35 for one person/ $20 for one person with eyes
- $18 for two people/ $26.69 for two people with eyes
- Each increase is about $4.50 - $6.00 ($10.31 for eyes), except from one to two persons which was $4.65 ($6.69 for eyes)
- Arrives Mar 24-28 (today is Mar 22)
- Ready in 6 hours
- Star seller
- 775 reviews for shop, 679 reviews for item
- Cost to ship: free
- Returns and exchanges accepted
  - She can make 3 revisions to final product
- Shop:
  - 2,580 sales
  - Rave reviews

PunoPrints (meh) *etsy's pick

- $18 for one person/ $26 for one person + background
- $24 for two people/ $32 for two people + background
- $6 increase from one to two people (both people and people + background), seems like $4 increase for any more than two
- Arrives Mar 24-30 (today is Mar 22)
- Ready in 6 hours
- Not star seller
- 8,000 reviews for shop, 4,009 reviews for item
- Cost to ship: free
- Returns and exchanges not accepted
- Shop:
  - 27,589 sales
  - Rave reviews

ZamaPrints (meh-wow)

- $11.96 for one person/ $15.22 for one person + background
- $15.76 for two people/ $19.03 for two people + background
- $3.80/ $3.81 increase from one to two people (both people and people + background). The same price increase stands for two to three people, three to four people, ...
- Arrives Mar 27-28 (today is Mar 22)
- Ready in 6 hours
- Not star seller
- 1,687 reviews for shop, 35 for item
- Cost to ship: free
- Returns accepted
- Shop:
  - 6,954 sales
  - Rave reviews

NLdigitalPortraits (wow!) *bestseller

- $11.40 for one person / $15.90 for one person + background
- $17.80 for two people/ $20.80 for two people + background
- $6.40 increase from one to two people/ $4.90 increase from one to two peole with background
- $3.10 - $1.90 increase from two to three people, three to four people, .../ $3 increase from two to three people with background, three to four people with background, ...
- Arrives Mar 24-27 (today is Mar 22)
- Ready in 12 hours
- Star seller
- 2,191 reviews for shop, 506 reviews for item
- Cost to ship: free
- Returns accepted
- Shop:
  - 8,652 sales
  - Rave reviews

IdyllicCreativeCo (wow!!) *bestseller

- $23.65 for one person/pet *digital image
- $29.64 for two people/pets digital *digital image
- Each increase is about $4.28, except from one to two people which was $5.99
- Arrives Mar 24-28 (today is Mar 22)
- Ready in 24 hours for digital print
- Star seller
- 416 reviews for shop, 155 reviews for item
- Cost to ship: free
- Returns and exchanges not accepted
- Shop:
  - 1,174 sales
  - Rave reviews

In [1]:
```
import numpy as np
import pandas as pd
# import matplotlib as plt
# import seaborn as sns
```

In [2]:
```
shops = ["EmmaDigitalStudio", "PunoPrints", "ZamaPrints", "NLdigitalPortraits", "IdyllicCreativeCo"]
shops_dict = {k:{} for k in shops}
shops_dict
```

Out[2]:
```
{'EmmaDigitalStudio': {},
 'PunoPrints': {},
 'ZamaPrints': {},
 'NLdigitalPortraits': {},
 'IdyllicCreativeCo': {}}
```

In [3]:

```
#bestseller: yes == 1, no == 0
#sales: total shop sales
#time: time in hours to deliver
#background_price: has NaN value if artist did not offer background drawings
#price_increase: range in price increases from one person to two people and two to three people.
#To keep it simple, it's based on price increases for images without backgrounds

shops_dict["EmmaDigitalStudio"] = {"bestseller": 1,
                                   "sales": 2580,
                                   "item_reviews": 679,
                                   "time": 6,
                                   "one_price": 13.35,
                                   "one_background_price": "NaN",
                                   "two_price": 18.00,
                                   "two_background_price": "NaN",
                                   "price_increase": [4.65, 6.00]
                                  }

shops_dict["PunoPrints"] = {"bestseller": 0,
                            "sales": 27589,
                            "item_reviews": 4009,
                            "time": 6,
                            "one_price": 18.00,
                            "one_background_price": 26.00,
                            "two_price": 24.00,
                            "two_background_price": 32.00,
                            "price_increase": [6.00, 4.00]
                           }

shops_dict["ZamaPrints"] = {"bestseller": 0,
                            "sales": 6954,
                            "item_reviews": 35,
                            "time": 6,
                            "one_price": 11.96,
                            "one_background_price": 15.22,
                            "two_price": 15.76,
                            "two_background_price": 19.03,
                            "price_increase": [3.80, 3.81]
                           }

shops_dict["NLdigitalPortraits"] = {"bestseller": 1,
                                    "sales": 8652,
                                    "item_reviews": 506,
                                    "time": 12,
                                    "one_price": 11.40,
                                    "one_background_price": 15.90,
                                    "two_price": 17.80,
                                    "two_background_price": 20.80,
                                    "price_increase": [3.10, 1.90]
                                   }

shops_dict["IdyllicCreativeCo"] = {"bestseller": 1,
                                   "sales": 1174,
                                   "item_reviews": 155,
                                   "time": 24,
                                   "one_price": 23.65,
                                   "one_background_price": "NaN",
                                   "two_price": 29.64,
                                   "two_background_price": "NaN",
                                   "price_increase": [5.99, 4.28]
                                  }
```

Determing price of digital art commissions

* At the moment, we will only focus on drawings that only feature people. We will not calculate prices for drawings that include a background or feature pets.

In [4]:
```
shops_df = pd.DataFrame(shops_dict)
shops_df
```

Out[4]:

	EmmaDigitalStudio	PunoPrints	ZamaPrints	NLdigitalPortraits	IdyllicCreativeCo
bestseller	1	0	0	1	1
sales	2580	27589	6954	8652	1174
item_reviews	679	4009	35	506	155
time	6	6	6	12	24
one_price	13.35	18.0	11.96	11.4	23.65
one_background_price	NaN	26.0	15.22	15.9	NaN
two_price	18.0	24.0	15.76	17.8	29.64
two_background_price	NaN	32.0	19.03	20.8	NaN
price_increase	[4.65, 6.0]	[6.0, 4.0]	[3.8, 3.81]	[3.1, 1.9]	[5.99, 4.28]

In [5]:
```
shops_df = shops_df.transpose()
shops_df
```

Out[5]:

	bestseller	sales	item_reviews	time	one_price	one_background_price	two_price	two_background_price	price_increase
EmmaDigitalStudio	1	2580	679	6	13.35	NaN	18.0	NaN	[4.65, 6.0]
PunoPrints	0	27589	4009	6	18.0	26.0	24.0	32.0	[6.0, 4.0]
ZamaPrints	0	6954	35	6	11.96	15.22	15.76	19.03	[3.8, 3.81]
NLdigitalPortraits	1	8652	506	12	11.4	15.9	17.8	20.8	[3.1, 1.9]
IdyllicCreativeCo	1	1174	155	24	23.65	NaN	29.64	NaN	[5.99, 4.28]

In [6]:
```
#Focus on bestsellers only
bestsellers_df = shops_df[shops_df["bestseller"] == 1]
bestsellers_df.replace("NaN", 0, inplace = True)
bestsellers_df
```

```
/Users/emilylopez/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py:5238: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  return super().replace(
```

Out[6]:

	bestseller	sales	item_reviews	time	one_price	one_background_price	two_price	two_background_price	price_increase
EmmaDigitalStudio	1	2580	679	6	13.35	0.0	18.00	0.0	[4.65, 6.0]
NLdigitalPortraits	1	8652	506	12	11.40	15.9	17.80	20.8	[3.1, 1.9]
IdyllicCreativeCo	1	1174	155	24	23.65	0.0	29.64	0.0	[5.99, 4.28]

In [7]:
```
#Find prices for bestsellers
calcs = {"avg":{}, "med":{}, "max":{}, "min":{}}
calcs["avg"]["one_person"] = round(np.mean(bestsellers_df["one_price"]), 2)
calcs["avg"]["one_background"] = round(np.mean(bestsellers_df["two_price"]),2)
calcs["avg"]["two_people"] = round(np.mean(bestsellers_df["two_price"]),2)
calcs["avg"]["two_background"] = round(np.mean(bestsellers_df["two_background_price"]),2)

calcs["med"]["one_person"] = np.median(bestsellers_df["one_price"])
calcs["med"]["one_background"] = np.median(bestsellers_df["one_background_price"])
calcs["med"]["two_people"] = np.median(bestsellers_df["two_price"])
calcs["med"]["two_background"] = np.median(bestsellers_df["two_background_price"])

calcs["max"]["one_person"] = np.max(bestsellers_df["one_price"])
calcs["max"]["one_background"] = np.max(bestsellers_df["one_background_price"])
calcs["max"]["two_people"] = np.max(bestsellers_df["two_price"])
calcs["max"]["two_background"] = np.max(bestsellers_df["two_background_price"])

calcs["min"]["one_person"] = np.min(bestsellers_df["one_price"])
calcs["min"]["one_background"] = np.min(bestsellers_df["one_background_price"])
calcs["min"]["two_people"] = np.min(bestsellers_df["two_price"])
calcs["min"]["two_background"] = np.min(bestsellers_df["two_background_price"])

calcs
```

Out[7]:
```
{'avg': {'one_person': 16.13,
  'one_background': 21.81,
  'two_people': 21.81,
  'two_background': 6.93},
 'med': {'one_person': 13.35,
  'one_background': 0.0,
  'two_people': 18.0,
  'two_background': 0.0},
 'max': {'one_person': 23.65,
  'one_background': 15.9,
  'two_people': 29.64,
  'two_background': 20.8},
 'min': {'one_person': 11.4,
  'one_background': 0.0,
  'two_people': 17.8,
  'two_background': 0.0}}
```

In [8]:
```
calcs_df = pd.DataFrame(calcs)
calcs_df = calcs_df.transpose()
calcs_df
```

Out[8]:

	one_person	one_background	two_people	two_background
avg	16.13	21.81	21.81	6.93
med	13.35	0.00	18.00	0.00
max	23.65	15.90	29.64	20.80
min	11.40	0.00	17.80	0.00

The dataframe above, `calcs_df` , features summary statistics for current prices of items made by bestselling artisits in our sample.

In [9]:
```
#See prices for shops that have good reviews vs sales: they have a lot of reviews compared to how much they sell
shops_df.sort_values(by = ["sales", "item_reviews"], ascending = False)
```

Out[9]:

	bestseller	sales	item_reviews	time	one_price	one_background_price	two_price	two_background_price	price_increase
PunoPrints	0	27589	4009	6	18.0	26.0	24.0	32.0	[6.0, 4.0]
NLdigitalPortraits	1	8652	506	12	11.4	15.9	17.8	20.8	[3.1, 1.9]
ZamaPrints	0	6954	35	6	11.96	15.22	15.76	19.03	[3.8, 3.81]
EmmaDigitalStudio	1	2580	679	6	13.35	NaN	18.0	NaN	[4.65, 6.0]
IdyllicCreativeCo	1	1174	155	24	23.65	NaN	29.64	NaN	[5.99, 4.28]

In [10]:
```
shops_df["reviews_v_sales"] = shops_df["item_reviews"]/shops_df["sales"]
rvs_df = shops_df.sort_values(by = "reviews_v_sales", ascending = False)
rvs_df
```

Out[10]:

	bestseller	sales	item_reviews	time	one_price	one_background_price	two_price	two_background_price	price_increase	reviews_v_sales
EmmaDigitalStudio	1	2580	679	6	13.35	NaN	18.0	NaN	[4.65, 6.0]	0.263178
PunoPrints	0	27589	4009	6	18.0	26.0	24.0	32.0	[6.0, 4.0]	0.145312
IdyllicCreativeCo	1	1174	155	24	23.65	NaN	29.64	NaN	[5.99, 4.28]	0.132027
NLdigitalPortraits	1	8652	506	12	11.4	15.9	17.8	20.8	[3.1, 1.9]	0.058484
ZamaPrints	0	6954	35	6	11.96	15.22	15.76	19.03	[3.8, 3.81]	0.005033

In [11]:
```
#Bestseller with highest reviews to sales ratio
rvs_df.head(1)
```

Out[11]:

	bestseller	sales	item_reviews	time	one_price	one_background_price	two_price	two_background_price	price_increase	reviews_v_sales
EmmaDigitalStudio	1	2580	679	6	13.35	NaN	18.0	NaN	[4.65, 6.0]	0.263178

In [12]:
```
#Find average price between avg and max for each category:
one_person_price = round((calcs_df["one_person"][0] + calcs_df["one_person"][2])/2, 2)
print("Price for one person image: $", one_person_price)
two_people_price = round((calcs_df["two_people"][0] + calcs_df["two_people"][2])/2, 2)
print("Price for two people image: $", two_people_price)
```

```
Price for one person image: $ 19.89
Price for two people image: $ 25.72
```

In [13]:
```
#check price increase from one to two people
price_increase = round(two_people_price - one_person_price, 2)
print("Price increase from one to two people: $", price_increase)
```

```
Price increase from one to two people: $ 5.83
```

In [14]:
```
#check price inrease from one to two people for the bestselling artist with the highest reviews to sales ratio
best_rvs = rvs_df.head(1)
avg_price_increase = round(np.mean(np.array(best_rvs["price_increase"][0])), 2)
avg_price_increase
```

Out[14]: 5.32

In [15]:
```
#check price increase for our product vs average price increase calculated above
round(avg_price_increase - price_increase, 2)
```

Out[15]: -0.51

Negative value means our price increase is higher than the avg price increase. We want prices that reflect the same price increase as the bestselling artist with the highest reviews to sales ratio.

In [16]:
```
#adjust prices:
new_one_person_price = round(one_person_price + 0.255, 2)
print("Price for one person image: $", new_one_person_price)
new_two_people_price = round(two_people_price - 0.255, 2)
print("Price for two people image: $", new_two_people_price)
```

```
Price for one person image: $ 20.14
Price for two people image: $ 25.46
```

In [17]:
```
#check new price increase
new_price_increase = round(new_two_people_price - new_one_person_price, 2)
print("Price increase from one to two people: $", new_price_increase)
```

```
Price increase from one to two people: $ 5.32
```

In [18]:
```
#check calculated price increase compared to avg price increase calculated eariler
round(avg_price_increase - new_price_increase, 2)
```

Out[18]: 0.0

Since it's 0.0, we know our calculated prices have similar price increases as our desired target number.

Discounted prices

In [19]:
```
discount_30 = round(new_two_people_price - (new_two_people_price * 0.30),2)
print("30% discount: $", discount_30)
discount_40 = round(new_two_people_price - (new_two_people_price * 0.40),2)
print("40% discount: $",discount_40)
discount_50 = round(new_two_people_price - (new_two_people_price * 0.50),2)
print("50% discount: $",discount_50)
```

```
30% discount: $ 17.82
40% discount: $ 15.28
50% discount: $ 12.73
```

In [20]:
```
one_discount_30 = round(new_one_person_price - (new_one_person_price * 0.30),2)
print("30% discount: $", one_discount_30)
one_discount_40 = round(new_one_person_price - (new_one_person_price * 0.40),2)
print("40% discount: $", one_discount_40)
one_discount_50 = round(new_one_person_price - (new_one_person_price * 0.50),2)
print("50% discount: $", one_discount_50)
```

```
30% discount: $ 14.1
40% discount: $ 12.08
50% discount: $ 10.07
```

Final Calculated Price

Price for drawing with one person: $ 20.14

```
30% discount: $ 14.10
40% discount: $ 12.08
50% discount: $ 10.07
```

Price for drawing with two people: $ 25.46

```
30% discount: $ 17.82
40% discount: $ 15.28
50% discount: $ 12.73
```

Advice from Creative Hive

My notes from their How to Price Art Prints (https://youtu.be/d0KJnkw9U2E) video:

- Look for similar types of pieces and look at what they're charging
  - Compare similar sizes and art materials to the ones you're doing
  - Also jot down how many sales that person has had to see what materials/sizes are popular
- Decide what options you definitely want to offer your customers and what you want to leave out
- Come up with a cost that takes into account per item cost, market research and profit
- Good idea to price your products at 2-4x, if not more, of the material costs
- $10 per item is the bare minimum recommended
- Consider: do you want to be known as an affordable shop or higher end shop (because you provide better quality and service)
- What the market will bear
  - You want to be at the higher end of what the market will bear
  - A consumer sees a high price, they perceive a better quality product
  - Situation: Market research reveals that you can't price product based on price you hoped to earn --> Maybe adjust your materials

In [ ]:
```
1
```