Scraping the Web with Python

We will use Python to scrape data from the MakeupAlley and Sephora websites. BeautifulSoup can be used on the MakeupAlley.com, while Selenium can be used on Sephora.com as the Sephora website is Javascript rendered (BeautifulSoup will not work here).

Please refer to my GitHub for the Python code I wrote to scrape these websites. I have also uploaded the complete data sets there.

For the analysis below, we will need to import Pandas, Numpy, and Regular Expressions for wrangling with the data, and Bokeh for visualizations.

In [1]:
```
import pandas as pd
import numpy as np
import regex as re

from bokeh.charts import Histogram, output_notebook, show
from bokeh.layouts import row
from bokeh.plotting import figure, output_notebook, show

output_notebook()
```

Loading BokehJS ...

Initializing the Data

Next, we will load the scraped data into DataFrames. Printing out the head of each dataframe shows us whether the DataFrame has been set up properly.

By printing the average rating of each DataFrame, we can see off the bat that the average product rating on Sephora is 4.25 vs MakeupAlley 3.84. We can also see that MakeupAlley has a much higher number of total reviews and products. It is important to note that MakeupAlley hosts reviews for any products in existence, while Sephora only hosts reviews for products that they carry - thus explaining the greater number of reviews and products on MakeupAlley.

In [2]:
```
sites = ["MakeupAlley","Sephora"]

df = {name: pd.DataFrame() for name in sites}

df["MakeupAlley"] = pd.read_csv("/users/rosannelai/Downloads/MakeupAlley
_Ratings_All.csv", sep="\t", encoding = "utf-8").dropna().drop_duplicat
es(subset="Product Name")
df["Sephora"] = pd.read_csv("/users/rosannelai/Downloads/Sephora_Ratings
_All.csv", sep="\t", encoding = "utf-8").dropna().drop_duplicates(subse
t="Product Name")

for name in df:
    print name
    print df[name].head()
    print "\n"
```

```
    print "Total Average Rating: "+str((df[name]["Average Rating"]* df[name]["Number of Reviews"]).sum()/df[name]["Number of Reviews"].sum())
    print "Total Number of Reviews: " + str(df[name]["Number of Reviews"].sum())
    print "Total Number of Products: " + str(len(df[name]))
    print "\n"
```

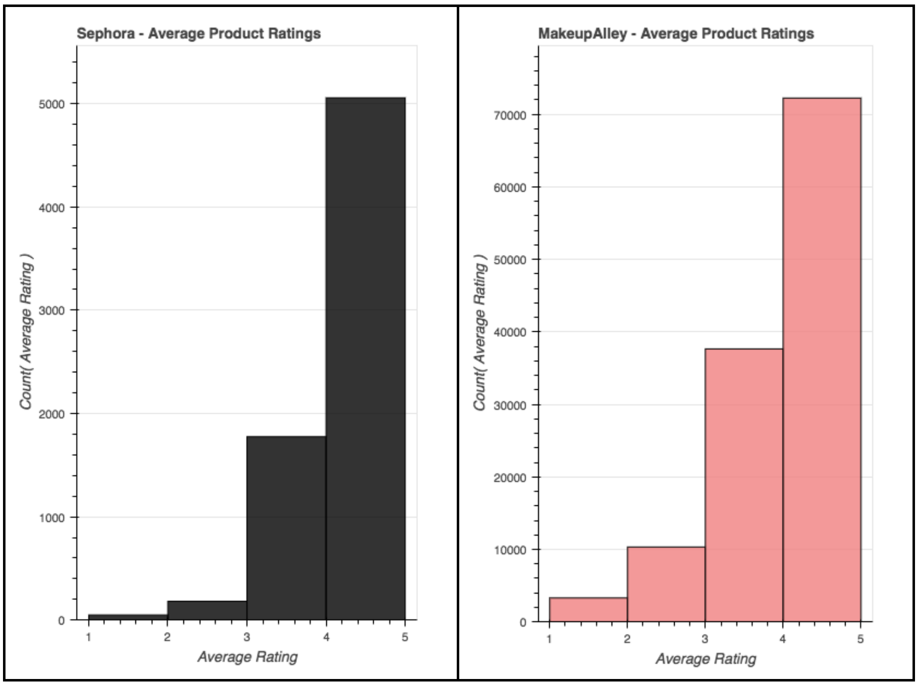
```
Sephora
    Brand Name                                          Product Name  \
0  DERMAdoctor                        DERMAdoctor KP Duty® Body Scrub
1   L’Occitane             L’Occitane Almond Eco-Refill Combo Pack
2   L’Occitane  L’Occitane Cleansing And Softening Shower Oil ...
3       boscia                             boscia Baby Soft Foot Peel
4    Herbivore        Herbivore Coco Rose Coconut Oil Body Polish

            Category  Average Rating  Number of Reviews
0  Bath-and-Body-Soap          4.5039             1020.0
1  Bath-and-Body-Soap          5.0000                2.0
2  Bath-and-Body-Soap          4.4568             1285.0
3  Bath-and-Body-Soap          4.2281              172.0
4  Bath-and-Body-Soap          4.5234              107.0


Total Average Rating: 4.252080413
Total Number of Reviews: 1573814.0
Total Number of Products: 7776


MakeupAlley
  Brand Name                        Product Name          Category  \
0    Anasazi   Anasazi Bee Pollen Conditioner        Conditioner
2     Arcona           Arcona Magic White Ice       Moisturizers
3     Arcona                  Arcona Eye Dew   Treatments (Eye)
4     Arcona              Arcona Desert Mist    Skincare - Face
5     Arcona           Arcona Hydrating Serum  Treatments (Face)

   Average Rating  Number of Reviews % Buy Again
0             4.0                1.0        100%
2             3.6               56.0         60%
3             3.8               18.0         72%
4             3.5               24.0         62%
5             4.2               13.0         69%


Total Average Rating: 3.83505482315
Total Number of Reviews: 2406830.0
Total Number of Products: 123552
```

Visualizing the Data As Is

Let's take a look at the distribution of average ratings across all products. A quick histogram plot shows that the there are far fewer products with a below-4 rating than on MakeupAlley. We can see that the distribution of products with a 2 or 3 rating on Sephora is significant lower than of MakeupAlley.

Could a fewer number of total reviews on Sephora cause the average product rating to be skewed higher than MakeupAlley? Perhaps a higher number of reviews on MakeupAlley causes the average rating of products to regress towards the average.

In [ ]:
```
hist_Sephora = Histogram(df["Sephora"]["Average Rating"][df["Sephora"]["
Number of Reviews"]>0], values = "Average Rating", bins = [1,2,3,4,5],
title = "Sephora - Average Product Ratings", color = "black", plot_width
=400)

hist_MakeupAlley = Histogram(df["MakeupAlley"]["Average Rating"][df["Mak
eupAlley"]["Number of Reviews"]>0], values = "Average Rating", bins = [1
,2,3,4,5], title = "MakeupAlley - Average Product Ratings", color = "li
ghtcoral", plot_width=400)

show (row(hist_Sephora, hist_MakeupAlley))
```



Comparing Sephora vs MakeupAlley by Brand

To answer the question above, let's aggregate the data by brand to compare. One would expect that the same brand be rated similarly between Sephora and MakeupAlley.

Here we will set up DataFrames aggregating the rating information by brand. Unlike above, we will calculate the average rating of each brand as the average rating of all products by the brand, weighted by the number of review for that product out of the total reviews for all products by the brand.

In [4]:
```
df_Brand = {name: pd.DataFrame() for name in sites}

def wavg(group, avg_name, weight_name):
    d = group[avg_name]
    w = group[weight_name]
    try:
        return (d * w).sum() / w.sum()
    except ZeroDivisionError:
        return d.mean()

for name in df_Brand:
```

```
    df_Brand[name]= pd.pivot_table(df[name], index="Brand Name",aggfunc
=np.sum)

    df_Brand[name]["Average Rating"] = df[name].groupby("Brand Name").a
pply(wavg, "Average Rating", "Number of Reviews")

    df_Brand[name]["Number of Products"] = df[name].groupby("Brand Name"
).size()

    print name
    print df_Brand[name].head()
    print "\n"
```

```
Sephora
                 Average Rating  Number of Reviews  Number of Products
Brand Name
AERIN                  4.348780              453.0                  33
AHAVA                  4.186429               59.0                  43
ALTERNA Haircare       4.250588             6554.0                  67
AMOREPACIFIC           4.427296             3092.0                  19
Acqua Di Parma         4.353047              325.0                  39


MakeupAlley
                 Average Rating  Number of Reviews  Number of Products
Brand Name
& Other Stories        3.625000                8.0                   8
100 Percent Pure       3.846443             1462.0                 201
1000HOUR               4.600000               27.0                   1
2 Grrrls               4.400000               35.0                  28
29 Cosmetics           4.400000               10.0                   6
```

Based on the total number of reviews written for each brand, we can determine the most popular brands on the Sephora website.

The top 10 most popular brands on Sephora are as follows:

In [5]:
```
print df_Brand["Sephora"].nlargest(10, "Number of Reviews")
```

```
                        Average Rating  Number of Reviews  Number of Prod
ucts
Brand Name

SEPHORA COLLECTION            4.152363           115470.0
387
Urban Decay                   4.366541            90717.0
99
Benefit Cosmetics             4.073919            77528.0
85
CLINIQUE                      4.257686            76157.0
205
NARS                          4.388070            70961.0
104
Too Faced                     4.147143            58546.0
53
tarte                         4.182035            57226.0
137
Kat Von D                     4.196836            56051.0
```

```
39
MAKE UP FOR EVER              4.163935          53630.0
173
Anastasia Beverly Hills        4.387240          48785.0
40
```

To look at the corresponding data for these brands from the MakeupAlley website, we will first need to set up a dictionary for the lookup of brand names due to small nuances. We will use Regular Expressions for this to find the corresponding names on MakeupAlley - which may contain an extra space or different capitalization than that on Sephora.

In [6]:

```
dict_Brand = {}

for n in df_Brand["MakeupAlley"].index:
    for element in df_Brand["Sephora"].index:
        if re.match(n, element, re.IGNORECASE):
            dict_Brand[element] = n
            break
        elif re.match(n+".", element, re.IGNORECASE):
            dict_Brand[element] = n
            break
        else:
            0
dict_Brand["Anastasia Beverly Hills"] = "Anastasia Of Beverly Hills "

print dict_Brand
```

```
{u'kate spade new york': u'Kate Spade', u'Acqua Di Parma': u'Acqua di Pa
rma', u'Buxom': u'Buxom', u'BECCA': u'Becca', u'Peter Thomas Roth': u'Pe
ter Thomas Roth', u'Urban Decay': u'Urban Decay', u'Juicy Couture': u'Ju
icy Couture', u'shu uemura': u'Shu Uemura', u'Chosungah 22': u'Chosungah
22', u'LAVANILA': u'Lavanila', u'Drunk Elephant': u'Drunk Elephant', u'P
AT McGRATH LABS': u'Pat McGrath Labs', u'Cinema Secrets': u'Cinema Secre
ts', u'Juliette Has a Gun': u'Juliette has a Gun', u'Jack Black': u'Jack
Black', u'SEPHORA COLLECTION': u'Sephora ', u'Biotherm': u'Biotherm', u'
Koh Gen Do': u'Koh Gen Do', u'Algenist': u'Algenist', u'Giorgio Armani B
eauty': u'Giorgio Armani', u'Drybar': u'Drybar', u'CLEAN': u'Clean', u'E
vian': u'Evian', u'ILIA': u'ILIA', u'Too Faced': u'Too Faced', u'Murad':
u'Murad', u'Comptoir Sud Pacifique': u'Comptoir Sud Pacifique', u'BALENC
IAGA': u'Balenciaga', u'Moschino': u'Moschino', 'Anastasia Beverly Hills
': 'Anastasia Of Beverly Hills ', u'NUDE Skincare': u'Nude Skincare', u'
DERMAdoctor': u'DERMAdoctor', u'Viktor & Rolf': u'Viktor & Rolf', u'Hana
e Mori': u'Hanae Mori', u'stila': u'Stila', u'Jurlique': u'Jurlique', u'
Clarins': u'Clarins', u'Salvatore Ferragamo': u'Salvatore Ferragamo', u'
JIMMY CHOO': u'Jimmy Choo', u'Smashbox': u'Smashbox', u'Eve Lom': u'Eve
Lom', u'NARS': u'NARS', u'Kat Von D': u'Kat Von D', u'Dior': u'Dior', u'
Deborah Lippmann': u'Deborah Lippmann', u'Omorovicza': u'Omorovicza', u'
Formula X': u'Formula X', u'DevaCurl': u'DevaCurl', u'Origins': u'Origin
s', u'TOCCA': u'Tocca', u'Atelier Cologne': u'Atelier Cologne', u'Cartie
r': u'Cartier', u'Moroccanoil': u'Moroccanoil', u'Jean Paul Gaultier': u
'Jean Paul Gaultier', u'Hugo Boss': u'Hugo Boss', u'Calvin Klein': u'Cal
vin Klein', u'Blinc': u'Blinc', u'Bobbi Brown': u'Bobbi Brown', u'Elizab
eth and James': u'Elizabeth and James', u'Clarisonic': u'Clarisonic', u'
B. Kamins': u'B. Kamins', u'beautyblender': u'beautyblender', u'First Ai
d Beauty': u'First Aid Beauty', u'Dr. Brandt Skincare': u'Dr. Brandt', u
'rms beauty': u'rms beauty', u'Caudalie': u'Caudalie', u'REN': u'Ren', u
'Stella McCartney': u'Stella McCartney', u'CLINIQUE': u'Clinique', u'Erb
orian': u'Erborian', u'Yves Saint Laurent': u'Yves Saint Laurent', u'Ver
sace': u'Versace', u'surratt beauty': u'Surratt', u'COVER FX': u'Cover F
X', u'Darphin': u'Darphin', u'Kenzo': u'Kenzo', u'Escada': u'Escada', u'
```

```
Laura Mercier': u'Laura Mercier', u'Diamancel': u'Diamancel', u'Guerlain
': u'Guerlain', u"Etat Libre d'Orange": u"Etat Libre D'Orange", u'Gucci'
: u'Gucci', u'Dr. Jart+': u'Dr. Jart+', u'GLAMGLOW': u'GLAMGLOW', u'DECI
EM': u'Deciem', u'KEVYN AUCOIN': u'Kevyn Aucoin', u'Phyto': u'Phyto', u'
Kate Somerville': u'Kate Somerville', u'TOM FORD': u'Tom Ford', u'Skyn I
celand': u'Skyn Iceland', u'Amazing Cosmetics': u'Amazing Cosmetics', u'
Hourglass': u'Hourglass', u'Marc Jacobs Beauty': u'Marc Jacobs', u'Tatch
a': u'Tatcha', u'BURBERRY': u'Burberry', u'Tria': u'tria', u'amika': u'A
mika', u'tarte': u'Tarte', u'Prada': u'Prada', u'Sally Hershberger 24K':
u'Sally Hershberger', u'Laneige': u'Laneige', u'Bite Beauty': u'Bite Bea
uty', u'AHAVA': u'Ahava', u'Too Cool For School': u'Too Cool For School'
, u'Givenchy': u'Givenchy', u'FARS\xc1LI': u'Fa', u'Benefit Cosmetics':
u'BeneFit Cosmetics', u'Living Proof': u'Living Proof', u'SUNDAY RILEY':
u'Sunday Riley', u'Oscar Blandi': u'Oscar Blandi', u'Paco Rabanne': u'Pa
co Rabanne', u'philosophy': u'Philosophy', u'T3': u'T3', u'boscia': u'Bo
scia', u'Perfekt': u'Perfekt', u'Tweezerman': u'Tweezerman', u'Juice Bea
uty': u'Juice Beauty', u'Fresh': u'Fresh', u'Farmacy': u'Farmacy', u'Nin
a Ricci': u'Nina Ricci', u'Caolion': u'Caolion', u'Issey Miyake': u'Isse
y Miyake', u'ALTERNA Haircare': u'Alterna', u'Serge Lutens': u'Serge Lut
ens', u'MAKE UP FOR EVER': u'Make Up For Ever', u'ghd': u'GHD', u'Perric
one MD': u'Perricone'}
```

Now, we can set up comparisons of the average ratings by brand between Sephora and MakeupAlley - and calculate the difference.

Similar to the overall rating difference we saw above, the average brand rating in all 10 instances of the most popular brands is significantly higher on Sephora than on MakeupAlley. We can see that the average rating difference of the top 10 brands ranges from 0.19 for Anastasia Beverly Hills to a whopping 0.59 for Clinique. Across the 10 brands, the average rating difference between Sephora and MakeupAlley is 0.33.

Interestingly, the total number of reviews on Sephora for each brand is actually higher than that of MakeupAlley Therefore, we can attribute the overall difference in the total number of reviews to the larger population of brands and products reviewed on MakeupAlley. The number of reviews does not appear to be the cause for the higher skewed rating on Sephora vs MakeupAlley.

The reason for the higher number of products by Brand on MakeupAlley is due to the fact that MakeupAlley often breaks out reviews by shade selection for each product.

In [7]:

```python
df_Compare = {name: pd.DataFrame() for name in df_Brand["Sephora"].nlargest(10, "Number of Reviews").index}
sum_Difference = 0

for name in df_Compare:
    df_Compare[name]["Sephora"] = df_Brand["Sephora"].loc[name]
    try:
        df_Compare[name]["MakeupAlley"] = df_Brand["MakeupAlley"].loc[dict_Brand[name]]
    except KeyError, e:
        print repr(e)
    df_Compare[name]["Difference"] = df_Compare[name]["Sephora"] - df_Compare[name]["MakeupAlley"]
    print name
    print df_Compare[name]
    sum_Difference = sum_Difference +  df_Compare[name]["Difference"].loc["Average Rating"]
    print "\n"
```

```
print "Average Difference in Rating Across the Top 10 Brands: " + str(su
m_Difference/10)
```

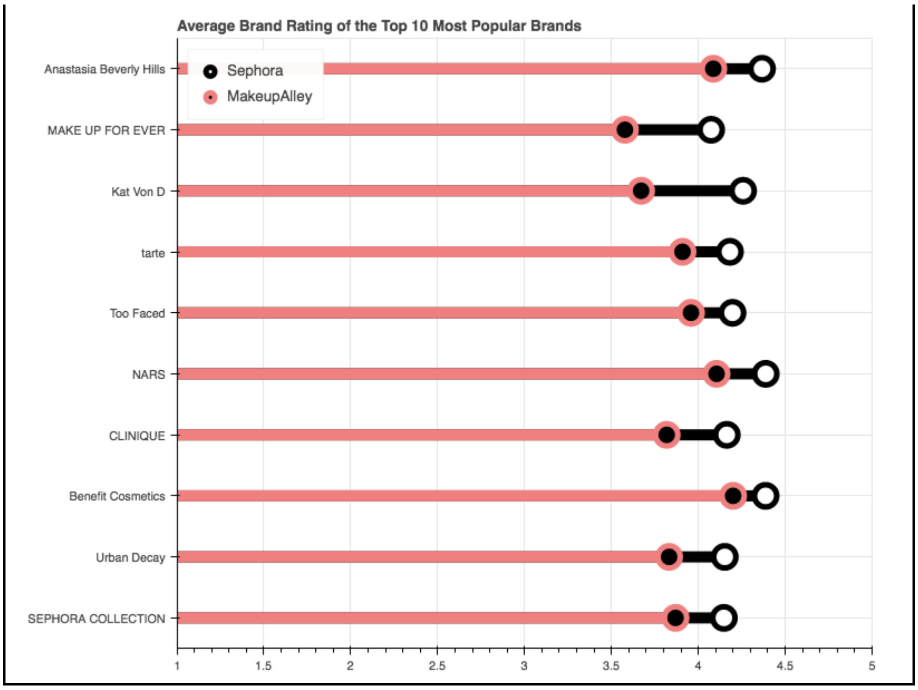
```
Too Faced
                        Sephora   MakeupAlley   Difference
Average Rating         4.147143      3.869193      0.27795
Number of Reviews  58546.000000  14776.000000  43770.00000
Number of Products    53.000000    570.000000   -517.00000


SEPHORA COLLECTION
                         Sephora   MakeupAlley     Difference
Average Rating          4.152363      3.832045       0.320319
Number of Reviews  115470.000000  11047.000000  104423.000000
Number of Products    387.000000   1004.000000    -617.000000


Anastasia Beverly Hills
                       Sephora  MakeupAlley    Difference
Average Rating         4.38724     4.200299      0.186941
Number of Reviews  48785.00000  3341.000000  45444.000000
Number of Products     40.00000   144.000000   -104.000000


MAKE UP FOR EVER
                        Sephora   MakeupAlley    Difference
Average Rating         4.163935      3.818159      0.345777
Number of Reviews  53630.000000  12121.000000  41509.000000
Number of Products   173.000000    447.000000   -274.000000


NARS
                       Sephora   MakeupAlley    Difference
Average Rating         4.38807      4.105116      0.282954
Number of Reviews  70961.00000  40933.000000  30028.000000
Number of Products   104.00000    938.000000   -834.000000


Kat Von D
                        Sephora  MakeupAlley    Difference
Average Rating         4.196836      3.95795      0.238886
Number of Reviews  56051.000000   3912.00000  52139.000000
Number of Products    39.000000    179.00000   -140.000000


tarte
                        Sephora   MakeupAlley    Difference
Average Rating         4.182035      3.909133      0.272902
Number of Reviews  57226.000000  11563.000000  45663.000000
Number of Products   137.000000    543.000000   -406.000000


CLINIQUE
                        Sephora   MakeupAlley   Difference
Average Rating         4.257686      3.670936      0.58675
Number of Reviews  76157.000000  58416.000000  17741.00000
Number of Products   205.000000   1000.000000   -795.00000


Benefit Cosmetics
                        Sephora   MakeupAlley    Difference
```

```
Average Rating            4.073919      3.577885      0.496034
Number of Reviews     77528.000000  38789.000000  38739.000000
Number of Products       85.000000    597.000000   -512.000000


Urban Decay
                           Sephora   MakeupAlley    Difference
Average Rating            4.366541      4.086297      0.280244
Number of Reviews     90717.000000  38233.000000  52484.000000
Number of Products       99.000000    946.000000   -847.000000


Average Difference in Rating Across the Top 10 Brands: 0.328875521537
```

Let's visualize the brand rating differences that we have calculated above.

In [ ]:

```
df_figBrand = pd.DataFrame()

for name in df_Compare:
    df_figBrand = df_figBrand.append (df_Compare[name].loc["Average Rating",["MakeupAlley","Sephora"]])

df_figBrand["Brand Name"] = df_Brand["Sephora"].nlargest(10, "Number of Reviews").index

factors = df_figBrand["Brand Name"].tolist()

df_figBrand.set_index("Brand Name", drop=True ,inplace = True)

x0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
x1 =  df_figBrand["MakeupAlley"]
x = df_figBrand["Sephora"]

p1 = figure(title="Average Brand Rating of the Top 10 Most Popular Brands", tools="resize,save", y_range=factors, x_range=[1,5],plot_width=800)

p1.segment(x0, factors, x, factors, line_width=10, line_color="black")
p1.circle(x, factors, size=20, fill_color="white", line_color="black", line_width=5, legend = "Sephora")
p1.segment(x0, factors, x1, factors, line_width=10, line_color="lightcoral")
p1.circle(x1, factors, size=20, fill_color="black", line_color="lightcoral", line_width=5, legend = "MakeupAlley")

p1.legend.location = "top_left"

show(p1)
```



Comparing Sephora vs MakeupAlley by Product

It would be interesting to see if the rating differences between Sephora and MakeupAlley are also true at the lowest level of aggregation - by product.

Let's take a look at the most popular products by number of reviews.

The 10 most popular products on Sephora are as follows:

In [9]:
```
print df["Sephora"].nlargest(10,"Number of Reviews")
```

```
                   Brand Name  \
728                      NARS
2291              Urban Decay
2287        Benefit Cosmetics
7511                    Buxom
7494                Kat Von D
1182               philosophy
2284                Kat Von D
2281  Anastasia Beverly Hills
2282                Too Faced
3457                Kat Von D

                                           Product Name     Category  \
728                                          NARS Blush  Cheek-Makeup
2291               Urban Decay 24/7 Glide-On Eye Pencil    Eye-Makeup
2287  Benefit Cosmetics They're Real! Lengthening & ...    Eye-Makeup
7511                          Buxom Full-On™ Lip Polish   Lips-Makeup
7494              Kat Von D Everlasting Liquid Lipstick   Lips-Makeup
1182             philosophy Purity Made Simple Cleanser      Cleanser
2284                            Kat Von D Tattoo Liner    Eye-Makeup
2281                 Anastasia Beverly Hills Brow Wiz     Eye-Makeup
2282                Too Faced Better Than Sex Mascara     Eye-Makeup
3457                       Kat Von D Lock-It Foundation  Face-Makeup

      Average Rating  Number of Reviews
```

```
728         4.6707          16498.0
2291        4.4198          14343.0
2287        4.1479          13150.0
7511        4.6353          11159.0
7494        4.2996          10449.0
1182        4.5431          10409.0
2284        4.2534           9993.0
2281        4.5010           9677.0
2282        3.7345           9276.0
3457        3.9572           9251.0
```

In [10]:
```
for name in df:
    df[name].set_index("Product Name", drop=True ,inplace = True)

df_Compare = {name: pd.DataFrame() for name in df["Sephora"].nlargest(1
0,"Number of Reviews").index}

for name in df_Compare:
    df_Compare[name]["Sephora"] = df["Sephora"].loc[name,["Average Rati
ng","Number of Reviews"]]
```

Again, we can set up comparisons of the average ratings by product between Sephora and MakeupAlley - and calculate the difference.

Yet again, the average brand rating in all 10 instances of the most popular products is significantly higher on Sephora than on MakeupAlley. We can see that the average rating difference of the top 10 products ranges from 0.10 for Anastasia Beverly Hills Brow Wiz to 0.84 for philosophy Purity Made Simple Cleanser.

While Sephora seems to be consistently honest about Anastasia, the other obvious differences between websites are now making me a bit more skeptical about the sincerity of Sephora reviews. It would be good to remember to take the shining product reviews on Sephora with a grain of salt!

Across the 10 products, the average rating difference between Sephora and MakeupAlley is 0.41.

In [11]:
```
dict_Product = {}
dict_Product["NARS Blush"] = ["NARS","Blush"]
dict_Product["Urban Decay 24/7 Glide-On Eye Pencil"] = ["Urban Decay","
Eyeliner"]
dict_Product["Kat Von D Everlasting Liquid Lipstick"] = ["Kat Von D","Li
pstick"]
dict_Product["Benefit Cosmetics They’re Real! Lengthening & Volumizing M
ascara".decode("utf-8")] = [" BeneFit Cosmetics They're Real"]
dict_Product["Buxom Full-On™ Lip Polish".decode("utf-8")] = ["Buxom","L
ip Gloss"]
dict_Product["philosophy Purity Made Simple Cleanser"] = [" Philosophy P
urity Made Simple (Real Purity Cleanser)"]
dict_Product["Kat Von D Tattoo Liner"] = [" Kat Von D Tattoo Liner"]
dict_Product["Anastasia Beverly Hills Brow Wiz"] = [" Anastasia Of Bever
ly Hills  Brow Wiz"]
dict_Product["Too Faced Better Than Sex Mascara"] = [" Too Faced Better
Than Sex Mascara"]
dict_Product["Kat Von D Lock-It Foundation"] = [" Kat Von D Lock-It Tat
too Foundation"]

sum_Difference = 0

for name in df_Compare:
    if name in ("Benefit Cosmetics They’re Real! Lengthening & Volumizin
```

```
g Mascara".decode("utf-8"),"philosophy Purity Made Simple Cleanser","Ka
t Von D Tattoo Liner","Anastasia Beverly Hills Brow Wiz","Kat Von D Lock
-It Foundation","Too Faced Better Than Sex Mascara"):
        df_Compare[name]["MakeupAlley"] = df["MakeupAlley"].loc[dict_Pro
duct[name][0],["Average Rating","Number of Reviews"]]
    else:
        try:
            df_Compare[name]["MakeupAlley"] = df["MakeupAlley"][(df["Mak
eupAlley"]["Brand Name"]==dict_Product[name][0])&(df["MakeupAlley"]["Ca
tegory"]==dict_Product[name][1])]["Number of Reviews"].sum()
            df_Compare[name]["MakeupAlley"]["Average Rating"] = (df["Mak
eupAlley"][(df["MakeupAlley"]["Brand Name"]==dict_Product[name][0])&(df
["MakeupAlley"]["Category"]==dict_Product[name][1])]["Average Rating"]*
df["MakeupAlley"][(df["MakeupAlley"]["Brand Name"]==dict_Product[name][
0])&(df["MakeupAlley"]["Category"]==dict_Product[name][1])]["Number of
Reviews"]).sum()/df_Compare[name]["MakeupAlley"]["Number of Reviews"]
        except KeyError, e:
            print repr(e)
    df_Compare[name]["Difference"] = df_Compare[name]["Sephora"] - df_Co
mpare[name]["MakeupAlley"]
    print name
    print df_Compare[name]
    sum_Difference = sum_Difference +  df_Compare[name]["Difference"].lo
c["Average Rating"]
    print "\n"

print "Average Difference in Rating Across the Top 10 Products: " + str(
sum_Difference/10)
```

```
/usr/local/lib/python2.7/site-packages/ipykernel/__main__.py:21: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-do
cs/stable/indexing.html#indexing-view-versus-copy
```

```
Kat Von D Everlasting Liquid Lipstick
                  Sephora  MakeupAlley Difference
Average Rating     4.2996     3.987968   0.311632
Number of Reviews   10449   748.000000       9701



NARS Blush
                  Sephora  MakeupAlley Difference
Average Rating     4.6707       4.2192     0.4515
Number of Reviews   16498   15047.0000       1451



Buxom Full-On™ Lip Polish
                  Sephora  MakeupAlley Difference
Average Rating     4.6353     4.328352   0.306948
Number of Reviews   11159  1238.000000       9921



Kat Von D Tattoo Liner
                  Sephora MakeupAlley Difference
Average Rating     4.2534         4.1     0.1534
Number of Reviews    9993         495       9498



philosophy Purity Made Simple Cleanser
                  Sephora MakeupAlley Difference
```

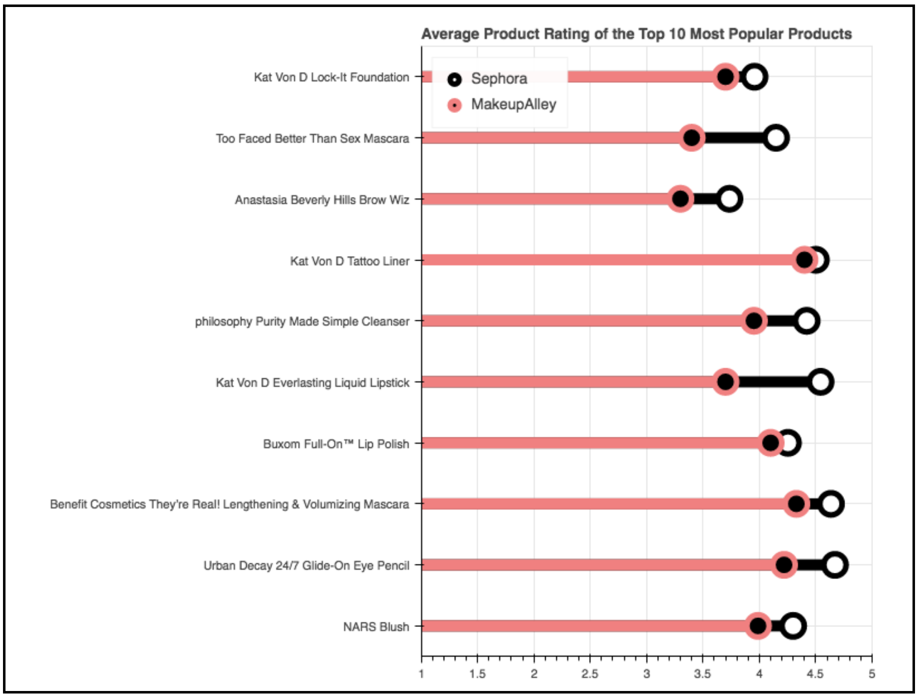
```
Average Rating     4.5431         3.7     0.8431
Number of Reviews   10409        2630       7779


Urban Decay 24/7 Glide-On Eye Pencil
                 Sephora  MakeupAlley Difference
Average Rating     4.4198     3.952098   0.467702
Number of Reviews   14343  4981.000000       9362


Anastasia Beverly Hills Brow Wiz
                 Sephora MakeupAlley Difference
Average Rating      4.501         4.4      0.101
Number of Reviews    9677         537       9140


Too Faced Better Than Sex Mascara
                 Sephora MakeupAlley Difference
Average Rating     3.7345         3.3     0.4345
Number of Reviews    9276         861       8415


Benefit Cosmetics They're Real! Lengthening & Volumizing Mascara
                 Sephora MakeupAlley Difference
Average Rating     4.1479         3.4     0.7479
Number of Reviews   13150        2393      10757


Kat Von D Lock-It Foundation
                 Sephora MakeupAlley Difference
Average Rating     3.9572         3.7     0.2572
Number of Reviews    9251         844       8407


Average Difference in Rating Across the Top 10 Products: 0.407488209183
```

Here are the product rating differences visualized.

In [ ]:

```
df_figProduct = pd.DataFrame()

for name in df_Compare:
    df_figProduct = df_figProduct.append (df_Compare[name].loc["Average Rating",["MakeupAlley","Sephora"]])

df_figProduct["Product Name"] = df["Sephora"].nlargest(10,"Number of Reviews").index

factors = df_figProduct["Product Name"].tolist()

df_figProduct.set_index("Product Name", drop=True ,inplace = True)

x0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
x1 =  df_figProduct["MakeupAlley"]
x = df_figProduct["Sephora"]

p1 = figure(title="Average Product Rating of the Top 10 Most Popular Products", tools="resize,save", y_range=factors, x_range=[1,5], plot_width=800)

p1.segment(x0, factors, x, factors, line_width=10, line_color="black")
```

```
p1.circle(x, factors, size=20, fill_color="white", line_color="black", line_width=5, legend = "Sephora")
p1.segment(x0, factors, x1, factors, line_width=10, line_color="lightcoral")
p1.circle(x1, factors, size=20, fill_color="black", line_color="lightcoral", line_width=5, legend = "MakeupAlley")

p1.legend.location = "top_left"

show(p1)
```



In [13]:
```
borderline = len(df["Sephora"]["Average Rating"][(df["Sephora"]["Average Rating"]*100 < 441)&(df["Sephora"]["Average Rating"]*100 > 400)])

print "Number of products rated above 4 but below 4.41 on Sephora: " + str(borderline)
print "These products as a percentage of all products rated above 4 :" + str(100*borderline/len(df["Sephora"]["Average Rating"][df["Sephora"]["Average Rating"]*100 > 400])) + "%"
```

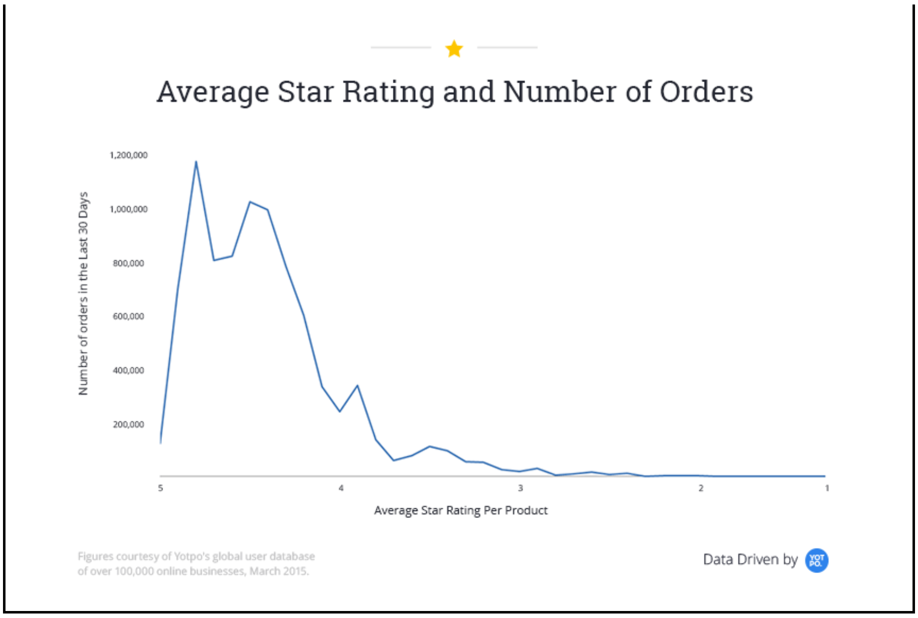
```
Number of products rated above 4 but below 4.41 on Sephora: 2171
These products as a percentage of all products rated above 4 :45%
```

Why Do We Care?

So what if Sephora's review ratings are a bit overstated? What does Sephora stand to gain from a 0.41 point difference?

1. People do not buy products rated less than a 4.

Yotpo conducted a study based on one million reviews and 8.6 million purchases, and found that 94% of purchases were made for products with a rating of 4 stars and above. Products with a rating below 4 only contributed to 6% of purchases.



2. 45% of products rated 4 or above on Sephora are within 0.41 points of that 4 star cutoff.

We calculated above that the average rating difference between Sephora and MakeupAlley for the top 10 products was 0.41 points. 45% of all products that are rated 4 or above on the Sephora website fall on the upper end of the 0.41 point range from the 4-star cutoff. Without any cost, Sephora has effectively expanded their offering of 4-star + products by 180% via the 0.41 point rating difference.

3. It's Strategic.

People trust user content more than brand/retailer content. User content invokes a psychological response known as "social proof" - we are hardwired to learn from others to help us avoid harmful choices. According to a survey by BrightLocal, 88 percent of consumers trust online reviews as much as a personal recommendation. More and more retailers are leveraging user content marketing strategies (ie. user reviews and photos) instead of spending on traditional avenues.