

Image color picker

Goal

pick out colors of article of clothing and convert to readable form (eg purple, magenta).

First experiment - KMeans clustering of color

Based on: <https://stackoverflow.com/questions/37022787/color-detection-of-object-in-image>
(<https://stackoverflow.com/questions/37022787/color-detection-of-object-in-image>)

In [1]:

```
!pip install pillow
```

Requirement already satisfied: pillow in c:\program files\python36\lib\site-packages

In [2]:

```
import matplotlib.image as mpimg

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import colors as mcolors
from matplotlib import patches
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin
from sklearn.datasets import load_sample_image
from sklearn.utils import shuffle
from time import time
from PIL import Image
```

In [3]:

```
img = mpimg.imread('./red-skirt.jpg')
pimg = Image.open('./red-skirt.jpg')
img.shape
```

Out[3]:

```
(440, 333, 3)
```

Transform image to list of individual pixels for clustering

In [4]:

```
image_array = np.array(img, dtype=np.float64) / 255 # convert to float and scale 0 - 1
w, h, d = original_shape = tuple(image_array.shape) # width, height, color depth
image_array = np.reshape(image_array, (w * h, d)) # reshape into list of individual pixels

image_array[:5]
```

Out[4]:

```
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.],
       [ 1.,  1.,  1.],
       [ 1.,  1.,  1.],
       [ 1.,  1.,  1.]])
```

Fit KMeans Model

In [5]:

```
n_colors = 3 ##

t0 = time()
image_array_sample = shuffle(image_array, random_state=0)[:1000]
kmeans = KMeans(n_clusters=n_colors, random_state=0).fit(image_array_sample)
centroids = kmeans.cluster_centers_ ## image average colors
t1 = time()

print('fit time: {0} s'.format(t1 - t0))

centroids
```

fit time: 0.03352069854736328 s

Out[5]:

```
array([[ 0.99771678,  0.99719971,  0.99711256],
       [ 0.8921686 ,  0.02874251,  0.09613714],
       [ 0.92499379,  0.14167287,  0.22670638]])
```

centeroids represent 3 average colors in the image

In [6]:

```
fig = plt.figure()

plt.axis('off')
ax = fig.add_subplot(111)

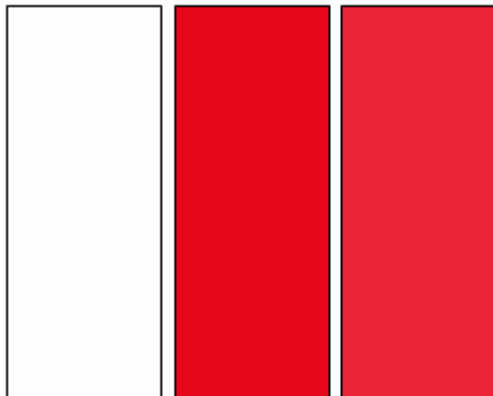
x_from = 0.00

for cluster_center in kmeans.cluster_centers_:
    ax.add_patch(patches.Rectangle( (x_from, 0.05), 0.23, 0.9, alpha=None, facecolor=tuple(cluster_center), edgecolor='black' ) )
    x_from = x_from + 0.25

plt.show()
plt.imshow(pimg)
plt.show()
```

c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



Evaluate colors

In [14]:

```
def find_n_best_colors(n, image_array):
    image_array = np.array(image_array, dtype=np.float64) / 255 # convert to float and
    scale 0 - 1
    w, h, d = original_shape = tuple(image_array.shape) # width, height, color depth
    image_array = np.reshape(image_array, (w * h, d)) # reshape into list of individual

    image_array_sample = shuffle(image_array, random_state=0)[:1000] # sample 1000 random pixels
    kmeans = KMeans(n_clusters=n, random_state=0).fit(image_array_sample) # find n clusters
    centroids = kmeans.cluster_centers_

    return centroids

def plot_colors_and_image(colors, img):
    fig = plt.figure()

    plt.axis('off')
    ax = fig.add_subplot(111)

    x_from = 0.00

    n = len(colors)

    for c in colors:
        ax.add_patch(patches.Rectangle( (x_from, 0.05), ((1/n) - 0.02), 0.9, alpha=None
, facecolor=tuple(c), edgecolor='black') )
        x_from = x_from + (1/n)

    if img is None:
        return

    plt.show()
    plt.imshow(img)
    plt.show()
```

In [15]:

```
image_array = mpimg.imread('./dark-red-dress.jpg')
pil_img = Image.open('./dark-red-dress.jpg')

colors = find_n_best_colors(5, image_array)
colors
```

Out[15]:

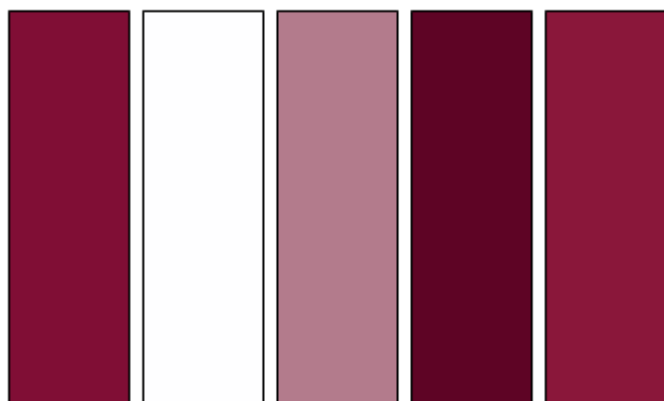
```
array([[ 0.50103605,  0.05570682,  0.20816851],
       [ 0.99791119,  0.99774948,  0.99813355],
       [ 0.70196078,  0.48039216,  0.54754902],
       [ 0.36993464,  0.01477124,  0.14457516],
       [ 0.54151409,  0.0888456 ,  0.22708739]])
```

In [16]:

```
plot_colors_and_image(colors, pil_img)
```

```
c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
```

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



Convert color to text

In [17]:

```
safe = mcolors.CSS4_COLORS
```

```
safe
```

Out[17]:

```
{'aliceblue': '#F0F8FF',
 'antiquewhite': '#FAEBD7',
 'aqua': '#00FFFF',
 'aquamarine': '#7FFFD4',
 'azure': '#F0FFFF',
 'beige': '#F5F5DC',
 'bisque': '#FFE4C4',
 'black': '#000000',
 'blanchedalmond': '#FFEBCD',
 'blue': '#0000FF',
 'blueviolet': '#8A2BE2',
 'brown': '#A52A2A',
 'burlywood': '#DEB887',
 'cadetblue': '#5F9EA0',
 'chartreuse': '#7FFF00',
 'chocolate': '#D2691E',
 'coral': '#FF7F50',
 'cornflowerblue': '#6495ED',
 'cornsilk': '#FFF8DC',
 'crimson': '#DC143C',
 'cyan': '#00FFFF',
 'darkblue': '#00008B',
 'darkcyan': '#008B8B',
 'darkgoldenrod': '#B8860B',
 'darkgray': '#A9A9A9',
 'darkgreen': '#006400',
 'darkgrey': '#A9A9A9',
 'darkkhaki': '#BDB76B',
 'darkmagenta': '#8B008B',
 'darkolivegreen': '#556B2F',
 'darkorange': '#FF8C00',
 'darkorchid': '#9932CC',
 'darkred': '#8B0000',
 'darksalmon': '#E9967A',
 'darkseagreen': '#8FBC8F',
 'darkslateblue': '#483D8B',
 'darkslategray': '#2F4F4F',
 'darkslategrey': '#2F4F4F',
 'darkturquoise': '#00CED1',
 'darkviolet': '#9400D3',
 'deeppink': '#FF1493',
 'deepskyblue': '#00BFFF',
 'dimgray': '#696969',
 'dimgrey': '#696969',
 'dodgerblue': '#1E90FF',
 'firebrick': '#B22222',
 'floralwhite': '#FFFAF0',
 'forestgreen': '#228B22',
 'fuchsia': '#FF00FF',
 'gainsboro': '#DCDCDC',
 'ghostwhite': '#F8F8FF',
 'gold': '#FFD700',
 'goldenrod': '#DAA520',
 'gray': '#808080',
 'green': '#008000',
 'greenyellow': '#ADFF2F',
 'grey': '#808080',
 'honeydew': '#F0FFF0',
 'hotpink': '#FF69B4',
 'indianred': '#CD5C5C'}
```

```
indianred': '#CD5C5C',  
'indigo': '#4B0082',  
'ivory': '#FFFFF0',  
'khaki': '#F0E68C',  
'lavender': '#E6E6FA',  
'lavenderblush': '#FFF0F5',  
'lawngreen': '#7CFC00',  
'lemonchiffon': '#FFFFACD',  
'lightblue': '#ADD8E6',  
'lightcoral': '#F08080',  
'lightcyan': '#E0FFFF',  
'lightgoldenrodyellow': '#FAFAD2',  
'lightgray': '#D3D3D3',  
'lightgreen': '#90EE90',  
'lightgrey': '#D3D3D3',  
'lightpink': '#FFB6C1',  
'lightsalmon': '#FFA07A',  
'lightseagreen': '#20B2AA',  
'lightskyblue': '#87CEFA',  
'lightslategray': '#778899',  
'lightslategrey': '#778899',  
'lightsteelblue': '#B0C4DE',  
'lightyellow': '#FFFFE0',  
'lime': '#00FF00',  
'limegreen': '#32CD32',  
'linen': '#FAF0E6',  
'magenta': '#FF00FF',  
'maroon': '#800000',  
'mediumaquamarine': '#66CDAA',  
'mediumblue': '#0000CD',  
'mediumorchid': '#BA55D3',  
'mediumpurple': '#9370DB',  
'mediumseagreen': '#3CB371',  
'mediumslateblue': '#7B68EE',  
'mediumspringgreen': '#00FA9A',  
'mediumturquoise': '#48D1CC',  
'mediumvioletred': '#C71585',  
'midnightblue': '#191970',  
'mintcream': '#F5FFFA',  
'mistyrose': '#FFE4E1',  
'moccasin': '#FFE4B5',  
'navajowhite': '#FFDEAD',  
'navy': '#000080',  
'oldlace': '#FDF5E6',  
'olive': '#808000',  
'olivedrab': '#6B8E23',  
'orange': '#FFA500',  
'orangered': '#FF4500',  
'orchid': '#DA70D6',  
'palegoldenrod': '#EEE8AA',  
'palegreen': '#98FB98',  
'paleturquoise': '#AFEEEE',  
'palevioletred': '#DB7093',  
'papayawhip': '#FFEFD5',  
'peachpuff': '#FFDAB9',  
'peru': '#CD853F',  
'pink': '#FFC0CB',  
'plum': '#DDA0DD',  
'powderblue': '#B0E0E6',  
'purple': '#800080',  
'rebeccapurple': '#663399',  
'red': '#FF0000',
```



```
'rosybrown': '#BC8F8F',  
'royalblue': '#4169E1',  
'saddlebrown': '#8B4513',  
'salmon': '#FA8072',  
'sandybrown': '#F4A460',  
'seagreen': '#2E8B57',  
'seashell': '#FFF5EE',  
'sienna': '#A0522D',  
'silver': '#C0C0C0',  
'skyblue': '#87CEEB',  
'slateblue': '#6A5ACD',  
'slategray': '#708090',  
'slategrey': '#708090',  
'snow': '#FFFAFA',  
'springgreen': '#00FF7F',  
'steelblue': '#4682B4',  
'tan': '#D2B48C',  
'teal': '#008080',  
'thistle': '#D8BFD8',  
'tomato': '#FF6347',  
'turquoise': '#40E0D0',  
'violet': '#EE82EE',  
'wheat': '#F5DEB3',  
'white': '#FFFFFF',  
'whitesmoke': '#F5F5F5',  
'yellow': '#FFFF00',  
'yellowgreen': '#9ACD32'}
```

In [18]:

```
def hex_to_rgb(value):
    value = value.lstrip('#')
    lv = len(value)
    rgb_int = tuple(int(value[i:i+lv//3], 16) for i in range(0, lv, lv//3)) ## 255, 25
5, 255
    rgb_fl = tuple(i / 255 for i in rgb_int)

    return rgb_fl

safe_map = { hex_to_rgb(v):k for (k,v) in safe.items()}
safe_map
```

Out[18]:

```
{(0.0, 0.0, 0.0): 'black',
 (0.0, 0.0, 0.5019607843137255): 'navy',
 (0.0, 0.0, 0.5450980392156862): 'darkblue',
 (0.0, 0.0, 0.803921568627451): 'mediumblue',
 (0.0, 0.0, 1.0): 'blue',
 (0.0, 0.39215686274509803, 0.0): 'darkgreen',
 (0.0, 0.5019607843137255, 0.0): 'green',
 (0.0, 0.5019607843137255, 0.5019607843137255): 'teal',
 (0.0, 0.5450980392156862, 0.5450980392156862): 'darkcyan',
 (0.0, 0.7490196078431373, 1.0): 'deepskyblue',
 (0.0, 0.807843137254902, 0.8196078431372549): 'darkturquoise',
 (0.0, 0.9803921568627451, 0.6039215686274509): 'mediumspringgreen',
 (0.0, 1.0, 0.0): 'lime',
 (0.0, 1.0, 0.4980392156862745): 'springgreen',
 (0.0, 1.0, 1.0): 'cyan',
 (0.09803921568627451,
 0.09803921568627451,
 0.4392156862745098): 'midnightblue',
 (0.11764705882352941, 0.5647058823529412, 1.0): 'dodgerblue',
 (0.12549019607843137,
 0.6980392156862745,
 0.6666666666666666): 'lightseagreen',
 (0.13333333333333333, 0.5450980392156862, 0.13333333333333333): 'forest
green',
 (0.1803921568627451, 0.5450980392156862, 0.3411764705882353): 'seagree
n',
 (0.1843137254901961,
 0.30980392156862746,
 0.30980392156862746): 'darkslategrey',
 (0.19607843137254902, 0.803921568627451, 0.19607843137254902): 'limegre
en',
 (0.23529411764705882,
 0.7019607843137254,
 0.44313725490196076): 'mediumseagreen',
 (0.25098039215686274, 0.8784313725490196, 0.8156862745098039): 'turquoi
se',
 (0.2549019607843137, 0.4117647058823529, 0.8823529411764706): 'royalblu
e',
 (0.27450980392156865, 0.5098039215686274, 0.7058823529411765): 'steelbl
ue',
 (0.2823529411764706,
 0.23921568627450981,
 0.5450980392156862): 'darkslateblue',
 (0.2823529411764706, 0.8196078431372549, 0.8): 'mediumturquoise',
 (0.29411764705882354, 0.0, 0.5098039215686274): 'indigo',
 (0.3333333333333333,
 0.4196078431372549,
 0.1843137254901961): 'darkolivegreen',
 (0.37254901960784315, 0.6196078431372549, 0.6274509803921569): 'cadetbl
ue',
 (0.39215686274509803,
 0.5843137254901961,
 0.9294117647058824): 'cornflowerblue',
 (0.4, 0.2, 0.6): 'rebeccapurple',
 (0.4, 0.803921568627451, 0.6666666666666666): 'mediumaquamarine',
 (0.4117647058823529, 0.4117647058823529, 0.4117647058823529): 'dimgre
y',
 (0.41568627450980394, 0.35294117647058826, 0.803921568627451): 'slatebl
ue',
 (0.4196078431372549, 0.5568627450980392, 0.13725490196078432): 'olivedr

```

```
(0.4190078431372549, 0.5508027450980392, 0.13725490196078433): 'olive  
ab',  
(0.4392156862745098, 0.5019607843137255, 0.5647058823529412): 'slategre  
y',  
(0.4666666666666667, 0.5333333333333333, 0.6): 'lightslategrey',  
(0.4823529411764706,  
0.40784313725490196,  
0.9333333333333333): 'mediumslateblue',  
(0.48627450980392156, 0.9882352941176471, 0.0): 'lawngreen',  
(0.4980392156862745, 1.0, 0.0): 'chartreuse',  
(0.4980392156862745, 1.0, 0.8313725490196079): 'aquamarine',  
(0.5019607843137255, 0.0, 0.0): 'maroon',  
(0.5019607843137255, 0.0, 0.5019607843137255): 'purple',  
(0.5019607843137255, 0.5019607843137255, 0.0): 'olive',  
(0.5019607843137255, 0.5019607843137255, 0.5019607843137255): 'grey',  
(0.5294117647058824, 0.807843137254902, 0.9215686274509803): 'skyblue',  
(0.5294117647058824, 0.807843137254902, 0.9803921568627451): 'lightskyb  
lue',  
(0.5411764705882353, 0.16862745098039217, 0.8862745098039215): 'bluevio  
let',  
(0.5450980392156862, 0.0, 0.0): 'darkred',  
(0.5450980392156862, 0.0, 0.5450980392156862): 'darkmagenta',  
(0.5450980392156862, 0.27058823529411763, 0.07450980392156863): 'saddle  
brown',  
(0.5607843137254902, 0.7372549019607844, 0.5607843137254902): 'darkseag  
reen',  
(0.5647058823529412, 0.9333333333333333, 0.5647058823529412): 'lightgre  
en',  
(0.5764705882352941, 0.4392156862745098, 0.8588235294117647): 'mediumpu  
rple',  
(0.5803921568627451, 0.0, 0.8274509803921568): 'darkviolet',  
(0.596078431372549, 0.984313725490196, 0.596078431372549): 'palegreen',  
(0.6, 0.19607843137254902, 0.8): 'darkorchid',  
(0.6039215686274509, 0.803921568627451, 0.19607843137254902): 'yellowgr  
een',  
(0.6274509803921569, 0.3215686274509804, 0.17647058823529413): 'sienn  
a',  
(0.6470588235294118, 0.16470588235294117, 0.16470588235294117): 'brow  
n',  
(0.6627450980392157, 0.6627450980392157, 0.6627450980392157): 'darkgre  
y',  
(0.6784313725490196, 0.8470588235294118, 0.9019607843137255): 'lightblu  
e',  
(0.6784313725490196, 1.0, 0.1843137254901961): 'greenyellow',  
(0.6862745098039216, 0.9333333333333333, 0.9333333333333333): 'paleturq  
uoise',  
(0.6901960784313725,  
0.7686274509803922,  
0.8705882352941177): 'lightsteelblue',  
(0.6901960784313725, 0.8784313725490196, 0.9019607843137255): 'powderbl  
ue',  
(0.6980392156862745, 0.13333333333333333, 0.13333333333333333): 'firebr  
ick',  
(0.7215686274509804,  
0.5254901960784314,  
0.043137254901960784): 'darkgoldenrod',  
(0.7294117647058823, 0.3333333333333333, 0.8274509803921568): 'mediumor  
chid',  
(0.7372549019607844, 0.5607843137254902, 0.5607843137254902): 'rosybrow  
n',  
(0.7411764705882353, 0.7176470588235294, 0.4196078431372549): 'darkkhak  
i',
```

```
(0.7529411764705882, 0.7529411764705882, 0.7529411764705882): 'silver',
(0.7803921568627451,
 0.08235294117647059,
 0.5215686274509804): 'mediumvioletred',
(0.803921568627451, 0.3607843137254902, 0.3607843137254902): 'indianred',
(0.803921568627451, 0.5215686274509804, 0.24705882352941178): 'peru',
(0.8235294117647058, 0.4117647058823529, 0.11764705882352941): 'chocolate',
(0.8235294117647058, 0.7058823529411765, 0.5490196078431373): 'tan',
(0.8274509803921568, 0.8274509803921568, 0.8274509803921568): 'lightgrey',
(0.8470588235294118, 0.7490196078431373, 0.8470588235294118): 'thistle',
(0.8549019607843137, 0.4392156862745098, 0.8392156862745098): 'orchid',
(0.8549019607843137, 0.6470588235294118, 0.12549019607843137): 'goldenrod',
(0.8588235294117647, 0.4392156862745098, 0.5764705882352941): 'palevioletred',
(0.8627450980392157, 0.0784313725490196, 0.23529411764705882): 'crimson',
(0.8627450980392157, 0.8627450980392157, 0.8627450980392157): 'gainsboro',
(0.8666666666666667, 0.6274509803921569, 0.8666666666666667): 'plum',
(0.8705882352941177, 0.7215686274509804, 0.5294117647058824): 'burlywood',
(0.8784313725490196, 1.0, 1.0): 'lightcyan',
(0.9019607843137255, 0.9019607843137255, 0.9803921568627451): 'lavender',
(0.9137254901960784, 0.5882352941176471, 0.47843137254901963): 'darksalmon',
(0.9333333333333333, 0.5098039215686274, 0.9333333333333333): 'violet',
(0.9333333333333333, 0.9098039215686274, 0.6666666666666666): 'palegoldenrod',
(0.9411764705882353, 0.5019607843137255, 0.5019607843137255): 'lightcoral',
(0.9411764705882353, 0.9019607843137255, 0.5490196078431373): 'khaki',
(0.9411764705882353, 0.9725490196078431, 1.0): 'aliceblue',
(0.9411764705882353, 1.0, 0.9411764705882353): 'honeydew',
(0.9411764705882353, 1.0, 1.0): 'azure',
(0.9568627450980393, 0.6431372549019608, 0.3764705882352941): 'sandybrown',
(0.9607843137254902, 0.8705882352941177, 0.7019607843137254): 'wheat',
(0.9607843137254902, 0.9607843137254902, 0.8627450980392157): 'beige',
(0.9607843137254902, 0.9607843137254902, 0.9607843137254902): 'whitesmoke',
(0.9607843137254902, 1.0, 0.9803921568627451): 'mintcream',
(0.9725490196078431, 0.9725490196078431, 1.0): 'ghostwhite',
(0.9803921568627451, 0.5019607843137255, 0.4470588235294118): 'salmon',
(0.9803921568627451, 0.9215686274509803, 0.8431372549019608): 'antiquewhite',
(0.9803921568627451, 0.9411764705882353, 0.9019607843137255): 'linen',
(0.9803921568627451,
 0.9803921568627451,
 0.8235294117647058): 'lightgoldenrodyellow',
(0.9921568627450981, 0.9607843137254902, 0.9019607843137255): 'oldlace',
(1.0, 0.0, 0.0): 'red',
(1.0, 0.0, 1.0): 'magenta',
(1.0, 0.0784313725490196, 0.5764705882352941): 'deeppink',
(1.0, 0.27058823529411763, 0.0): 'orangered',
```

```
(1.0, 0.38823529411764707, 0.2784313725490196): 'tomato',  
(1.0, 0.4117647058823529, 0.7058823529411765): 'hotpink',  
(1.0, 0.4980392156862745, 0.3137254901960784): 'coral',  
(1.0, 0.5490196078431373, 0.0): 'darkorange',  
(1.0, 0.6274509803921569, 0.47843137254901963): 'lightsalmon',  
(1.0, 0.6470588235294118, 0.0): 'orange',  
(1.0, 0.7137254901960784, 0.7568627450980392): 'lightpink',  
(1.0, 0.7529411764705882, 0.796078431372549): 'pink',  
(1.0, 0.8431372549019608, 0.0): 'gold',  
(1.0, 0.8549019607843137, 0.7254901960784313): 'peachpuff',  
(1.0, 0.8705882352941177, 0.6784313725490196): 'navajowhite',  
(1.0, 0.8941176470588236, 0.7098039215686275): 'moccasin',  
(1.0, 0.8941176470588236, 0.7686274509803922): 'bisque',  
(1.0, 0.8941176470588236, 0.8823529411764706): 'mistyrose',  
(1.0, 0.9215686274509803, 0.803921568627451): 'blanchedalmond',  
(1.0, 0.9372549019607843, 0.8352941176470589): 'papayawhip',  
(1.0, 0.9411764705882353, 0.9607843137254902): 'lavenderblush',  
(1.0, 0.9607843137254902, 0.9333333333333333): 'seashell',  
(1.0, 0.9725490196078431, 0.8627450980392157): 'cornsilk',  
(1.0, 0.9803921568627451, 0.803921568627451): 'lemonchiffon',  
(1.0, 0.9803921568627451, 0.9411764705882353): 'floralwhite',  
(1.0, 0.9803921568627451, 0.9803921568627451): 'snow',  
(1.0, 1.0, 0.0): 'yellow',  
(1.0, 1.0, 0.8784313725490196): 'lightyellow',  
(1.0, 1.0, 0.9411764705882353): 'ivory',  
(1.0, 1.0, 1.0): 'white'}
```

Calculate distance of input to each color point in the list and sort

In [19]:

```

from scipy.spatial import distance

def get_distance(x,y):

    if len(x) == 4:
        x = (x[0], x[1], x[2])

    if len(y) == 4:
        y = (y[0], y[1], y[2])

    return distance.euclidean(x,y)

def get_closest_point(p, pxs):
    d_min = None
    p_min = None

    for px in pxs:
        d = get_distance(p, px)

        if d_min is None or d < d_min:
            d_min = d
            p_min = px

    return p_min

def get_closest_color(c, colormap):
    named_c = get_closest_point(c, list(colormap.keys()))
    return colormap[named_c]

```

In [20]:

```
get_closest_color(c, safe_map)
```

```

-----
-
NameError                                Traceback (most recent call las
t)
<ipython-input-20-c9c1af11a4a7> in <module>()
----> 1 get_closest_color(c, safe_map)

```

NameError: name 'c' is not defined

In [35]:

```
image_array = mpimg.imread('./dark-red-dress.jpg')
pil_img = Image.open('./dark-red-dress.jpg')

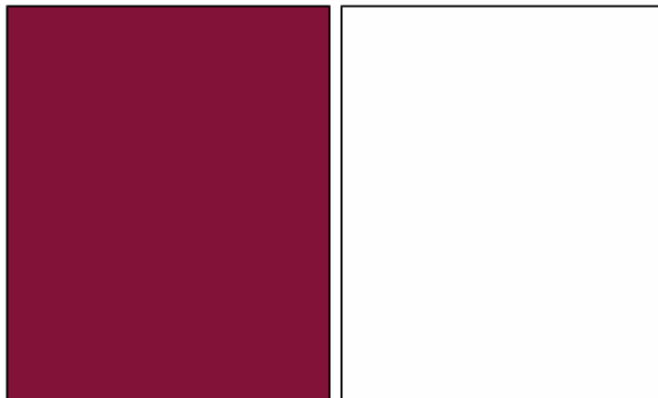
colors = find_n_best_colors(2, image_array)
plot_colors_and_image(colors, pil_img)

named_colors = [get_closest_color(c, safe_map) for c in colors]

named_colors
```

c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



Out[35]:

```
['brown', 'white']
```


In [22]:

```
def evaluate_image(path, n):
    image_array = mpimg.imread(path)
    pil_img = Image.open(path)

    colors = find_n_best_colors(n, image_array)
    plot_colors_and_image(colors, pil_img)

    named_colors = [get_closest_color(c, safe_map) for c in colors]
    print(named_colors)

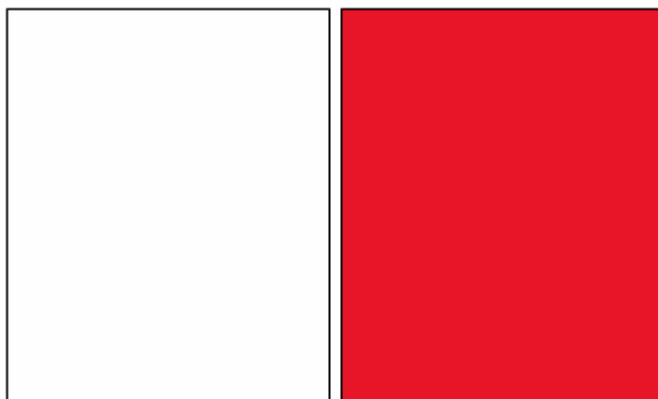
    return (named_colors, colors, pil_img)
```

In [23]:

```
res = evaluate_image('./red-skirt.jpg', 2)
```

c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



```
['white', 'crimson']
```

In [24]:

```
res = evaluate_image('./black-jacket.jpg', 2)
```

```
c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
```

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



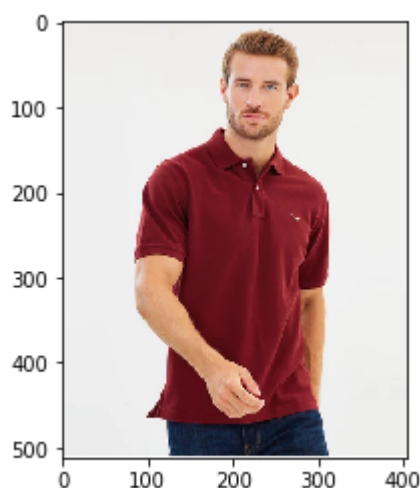
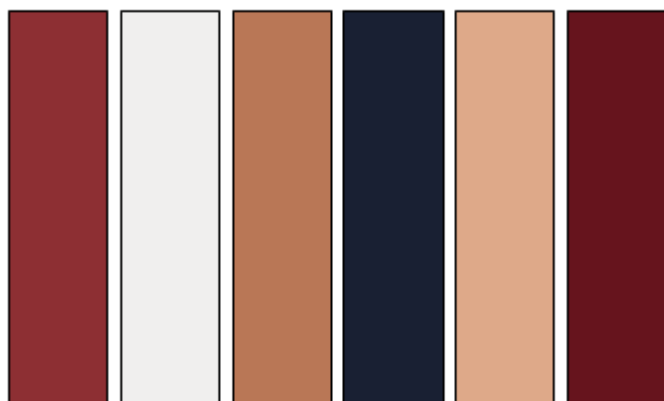
```
['black', 'white']
```

In [25]:

```
res = evaluate_image('./red-shirt.jpg', 6)
```

```
c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
```

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



```
['brown', 'whitesmoke', 'peru', 'darkslategrey', 'burlywood', 'maroon']
```

NOTE- fix for different color depths

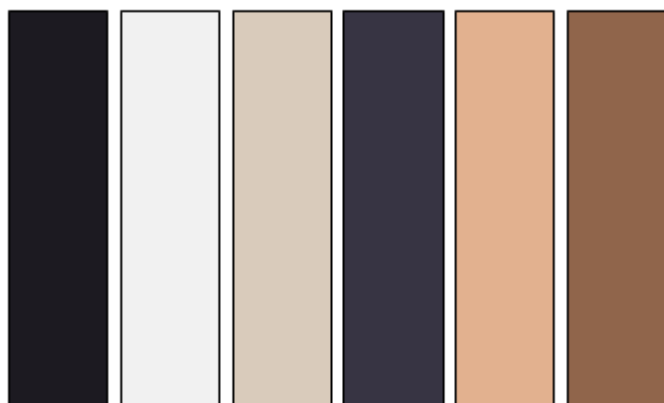
figure out how to map alpha color correctly

In [28]:

```
res = evaluate_image('./navy-shirt.jpeg', 6)
```

```
c:\program files\python36\lib\site-packages\matplotlib\cbook\deprecation.py:106: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
```

```
warnings.warn(message, mplDeprecation, stacklevel=1)
```



```
['black', 'whitesmoke', 'lightgrey', 'darkslategrey', 'burlywood', 'sienna']
```

Map colors to names - RAL

In [33]:

```
ral_colors = pd.read_csv('ral_standard.csv')
ral_colors.head(20)
ral_colors.values
```

Out[33]:

```
array([[ 'RAL 1000', '190-189-127', '#BED7F', ..., 'Beige verdoso',
        'Beige\xa0verdaastro', 'Groenbeige'],
       [ 'RAL 1001', '194-176-120', '#C2B078', ..., 'Beige', 'Beige',
        'Beige'],
       [ 'RAL 1002', '198-166-100', '#C6A664', ..., 'Amarillo\xa0arena',
        'Giallo sabbia', 'Zandgeel'],
       ...,
       [ 'RAL 9018', '215-215-215', '#D7D7D7', ..., 'Blanco papiro',
        'Bianco\xa0papiro', 'Papyruswit'],
       [ 'RAL 9022', '156-156-156', '#9C9C9C', ..., 'Gris claro perlado',
        'Grigio chiaro perlato', 'Parelmoer-lichtgrijs'],
       [ 'RAL 9023', '130-130-130', '#828282', ..., 'Gris oscuro perlado',
        'Grigio scuro perlato', 'Parelmoer-donker grijs']], dtype=object)
```

In [34]:

```
def ral_convert():
```

```
    ral_map = { row for row in ral_colors.values}
```

```
    File "<ipython-input-34-510e0a4cc8ce>", line 5
        ral_map = { for row in ral_colors.values}
                    ^
```

IndentationError: expected an indented block