Determotology Dataset

This dataset contains 34 attributes, All fetures are numerical features and the Problem is multiclass problem with target feature as multiclass

The differential diagnosis of erythemato-squamous diseases is a real problem in dermatology. They all share the clinical features of erythema and scaling, with very little differences. The diseases in this group are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. Usually a biopsy is necessary for the diagnosis but unfortunately these diseases share many histopathological features as well. Another difficulty for the differential diagnosis is that a disease may show the features of another disease at the beginning stage and may have the characteristic features at the following stages. Patients were first evaluated clinically with 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope.

In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values. Here is datasource (https://archive.ics.uci.edu/ml/datasets/dermatology)

After some precprocessing final data descrpition is

Data Preprocessing

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
data = pd.read_csv("Dermo_Disease.csv")
```

In [3]:
```
## Check the missing value in data
```

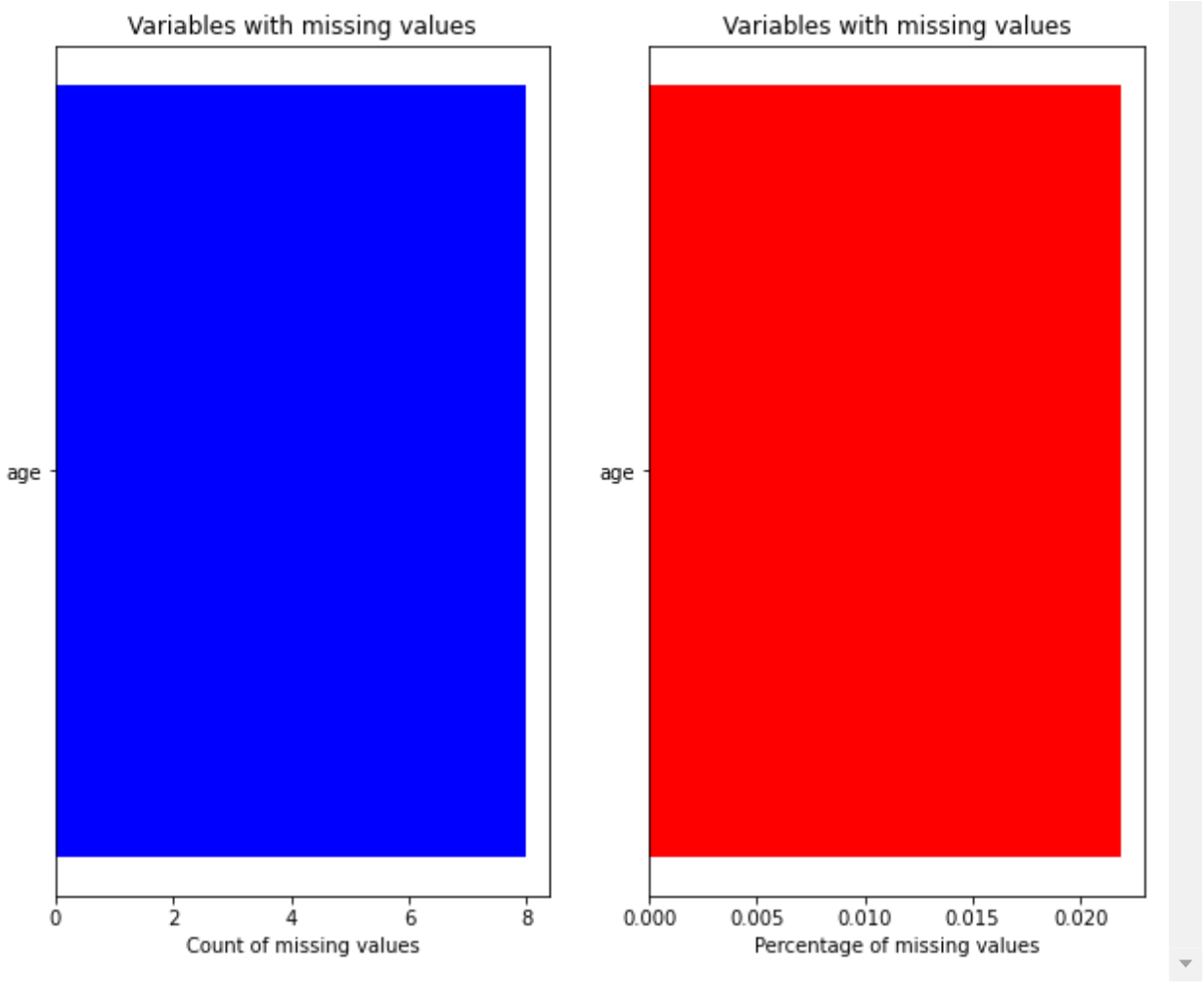
In [4]:
```
# Check which variables have missing values

columns_with_missing_values = data.columns[data.isnull().any()]
data[columns_with_missing_values].isnull().sum()
```

Out[4]:
```
age    8
dtype: int64
```

In [5]:
```
# viusalize the mssing pattern in data
```

In [6]:
```
# To hold variable names
labels = []

# To hold the count of missing values for each variable
valuecount = []

# To hold the percentage of missing values for each variable
percentcount = []

for col in columns_with_missing_values:
    labels.append(col)
    valuecount.append(data[col].isnull().sum())
    # data.shape[0] will give the total row count
    percentcount.append(data[col].isnull().sum()/data.shape[0])

ind = np.arange(len(labels))

fig, (ax1, ax2) = plt.subplots(1,2,figsize=(10,8))

rects = ax1.barh(ind, np.array(valuecount), color='blue')
ax1.set_yticks(ind)
ax1.set_yticklabels(labels, rotation='horizontal')
ax1.set_xlabel("Count of missing values")
ax1.set_title("Variables with missing values")

rects = ax2.barh(ind, np.array(percentcount), color='red')
ax2.set_yticks(ind)
ax2.set_yticklabels(labels, rotation='horizontal')
ax2.set_xlabel("Percentage of missing values")
ax2.set_title("Variables with missing values")
```

Out[6]:
```
Text(0.5, 1.0, 'Variables with missing values')
```



In [7]:
```
# W e have only one features that has missing value so we hve to remove that feat
```

In [8]:
```
data.drop(["age"], axis=1, inplace=True)
data
```

Out[8]:

	erythema	scaling	definite_borders	itching	koebner_phenomenon	polygonal_papules	follicul
0	2	2	0	3	0	0	
1	3	3	3	2	1	0	
2	2	1	2	3	1	3	
3	2	2	2	0	0	0	
4	2	3	2	2	2	2	
...	...	...	...	...	...	...	
361	2	1	1	0	1	0	
362	3	2	1	0	1	0	
363	3	2	2	2	3	2	
364	2	1	3	1	2	3	
365	3	2	2	0	0	0	

366 rows × 34 columns

In [ ]:
```
```