Workshop Notebook 5: Quality Assurance with ExifTool

Mandatory Disclosures

1. This is a whirlwind introduction, not exhaustive instruction
2. All images are by courtesy of the University Archives at Texas State University: http://www.univarchives.txstate.edu (http://www.univarchives.txstate.edu)
3. img_qc_workshop is licensed under the GNU General Public License v3.0, https://github.com/photosbyjeremy/img_qc_workshop/blob/master/LICENSE (https://github.com/photosbyjeremy/img_qc_workshop/blob/master/LICENSE)
4. *Any and all code provided is done so without any warranty or expectation of support by Jeremy Moore, Todd Peters, or Texas State University*

```
In [ ]: # importing
        from pathlib import Path
        from PIL import Image
        import matplotlib.pyplot as plt

        # import Jeremy's code from img_qc/img_qc.py
        import img_qc.img_qc as img_qc  # don't want to have to type it twice so import it as img_qc
```

```
In [ ]: # matplotlib options

        # magic that lets us plot directly in the notebook
        %matplotlib inline

        # parameters for matplotlib to increase our default figure size -- NOTE: figure sizes are in INCHES
        plt.rcParams["figure.figsize"] = (12,12)  # set as needed for your screen and eyes

        # on a high-dpi monitor this will increase the quality of plots on-screen
        %config InlineBackend.figure_format = 'retina'
```

```
In [ ]: # set QC directory
        qc_directory = 'data/workshop-5/undergrad_catalogs/'
```

```
In [ ]: # run Jeremy's function to load metadata using ExifTool into
        # a DataFrame (think Excel: rows & colummns of data)
        images_df = img_qc.get_images_df(qc_directory, 'tif')

        images_df
```

```
In [ ]: # each column name is the key to get the values for that column
        images_df['File:FileName']
```

```
In [ ]: # we can get a the unique values in a column with the method DataFrame[column_name].unique()
        images_df['EXIF:XResolution'].unique()
```

```
In [ ]: # we can get a DataFrame where the DataFrame == a certain value
        dpi600_df = images_df[images_df['EXIF:XResolution'] == 600]

        dpi600_df
```

```
In [ ]: # unique widths
        dpi600_df['EXIF:ImageWidth'].unique()
```

```
In [ ]: # file path where the width == 3808
        dpi600_df['SourceFile'][dpi600_df['EXIF:ImageWidth'] == 3808]
```

```
In [ ]: # import pandas so we can adjust options
        import pandas as pd

        # set max_colwidth for a wider column so data isn't truncated
        pd.set_option('max_colwidth', 240)

        # file path where the width == 3808
        dpi600_df['SourceFile'][dpi600_df['EXIF:ImageWidth'] == 3808]
```

```
In [ ]: # unique widths for all images

        images_df['EXIF:ImageWidth'].unique()
```

```
In [ ]: # get the horizontal images; where width > height

        horizontal_df = images_df[images_df['EXIF:ImageWidth'] > images_df['EXIF:ImageHeight']]

        for sourcefile in horizontal_df['SourceFile']:

            print(sourcefile)
```

```
In [ ]: # loop over sourcefiles and display them

        for sourcefile in horizontal_df['SourceFile']:

            # open image
            horizontal_image = Image.open(sourcefile)

            plt.figure()
            # display image
            plt.imshow(horizontal_image)
```

```
In [ ]: # get all of the directories

        images_df['File:Directory'].unique()

        # get directories_df

        image_directory_df = images_df[images_df['File:Directory'] == 'data/workshop-5/undergrad_catalogs/1912-ann
        ual']

        image_directory_df['File:FileName'}
```