Cleaning HPWRA input data for HAPI Project

Kelsey Brock - Last Updated Feb 28, 2020

In [1]:

```
# ignore warnings to compile the final notebook
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
import pandas as pd
```

Cleaning HPWRA Data

In [3]:

```
# import csv file of HPWRA data with analysis friendly column headings

CompiledData = pd.read_csv("UltimateDataCompilation.csv", sep=',',
                  names = ["Filename",
                           "1.01_domesticated","1.02_naturalized_grown","1.03_weedy_races",
                           "2.01_clim_match","2.02_climmatch_qual",
                           "2.03_broad_clim","2.04_similar_clim","2.05_repeat_intro",
                           "3.01_beyond_native", "3.02_disturbance_weed",
                           "3.03_agri_forestry_weed", "3.04_enviro_weed",
                           "3.05_congener","4.01_spiny", "4.02_allelopathic",
                           "4.03_parasitic", "4.04_unpalatable", "4.05_toxic", "4.06_alternate_host
,
                           "4.07_allergies", "4.08_fire_hazard", "4.09_shade_tolerant", "4.10_toler
tes_soilcond",
                           "4.11_climber","4.12_forms_thickets","5.01_aquatic","5.02_grass",
                           "5.03_nitrogen_fixer",
"5.04_geophyte","6.01_repro_failure","6.02_viable_seed",
                           "6.03_hybridizes",
"6.04_selfcompatible","6.05_special_pollinators","6.06_vegetative_repro",
                           "6.07_miniumum_gen_time","7.01_unintentional_dispersal",
                           "7.02_intentional_dispersal","7.03_contaminant_dispersal",
                           "7.04_wind_dispersal","7.05_water_dispersal","7.06_bird_dispersal",
                           "7.07_animal_dispersal","7.08_survive_gut","8.01_prolific_seeder",
                           "8.02_propagule_bank","8.03_herbicide_controlled",
                           "8.04_tolerates_mutilation","8.05_local_enemies", "manual_score"])
```

In [4]:

```
# visual check to make sure everything loaded okay
CompiledData.head(5)
```

Out[4]:

	Filename	1.01_domesticated	1.02_naturalized_grown	1.03_weedy_races	2.01_clim_match	2.02_climmatch_qual	2.03_br
0	File_name	1.01	1.02	1.03	2.01	2.02	
1	Abelia_x_grandiflora.xls	y	n	n	1	2	
2	Acacia_auriculiformis.xls	N	Y	N	2	2	
3	Acacia_confusa.xls	N	Y	N	2	2	
4	Acacia_crassicarpa.xls	n	y	n	2	2	

5 rows × 51 columns

In [5]:

```
#removing the old names in the first column
```

```
HPWRA1 = CompiledData.iloc[1:]
```

In [12]:

```
HPWRA1.head(5)
```

Out[12]:

	Filename	1.01_domesticated	1.02_naturalized_grown	1.03_weedy_races	2.01_clim_match	2.02_climmatch_qual	2
1	Abelia_x_grandiflora.xls	Yes	n	n	1	2	
2	Acacia_auriculiformis.xls	No	Y	N	2	2	
3	Acacia_confusa.xls	No	Y	N	2	2	
4	Acacia_crassicarpa.xls	No	y	n	2	2	
5	Acacia_farnesiana.xls	No	y	n	2	2	
6	Acacia_longifolia.xls	No	y	n	1	1	
7	Acacia_mangium.xls	No	NaN	NaN	2	2	
8	Acacia_mearnsii.xls	No	y	n	1	2	
9	Acacia_melanoxylon.xls	No	y	n	1	2	
10	Acacia_nilotica.xls	No	y	NaN	2	2	
11	Acacia_parramattensis.xls	No	y	n	2	2	
12	Acacia_pycnantha.xls	NaN	NaN	NaN	1	2	
13	Acalypha_godseffiana.xls	No	n	n	2	1	
14	Acalypha_hispida.xls	No	y	n	2	2	
15	Acalypha_wilkesiana.xls	No	y	n	2	2	
16	Acmella_grandiflora.xls	No	NaN	NaN	2	2	
17	Acoelorraphe_wrightii.xls	No	n	n	2	2	
18	Adansonia_digitata.xls	No	n	n	2	2	
19	Adenanthera_pavonina.xls	No	y	n	2	2	
20	Adenium_obesum.xls	No	n	n	2	2	
21	Aechmea_blanchetiana.xls	No	n	n	2	2	
22	Aechmea_fasicata.xls	No	n	n	2	2	
23	Aeschynomene_americana.xls	No	y	n	2	2	
24	Afrocarpus_falcatus.xls	No	n	n	2	2	
25	Agapanthus_africanus.xls	No	NaN	NaN	1	1	

25 rows × 51 columns

In [7]:

```
# Let's return the # of rows
print( "Number of Assessments in Data Set = " + str(len(HPWRA1)))
```

```
Number of Assessments in Data Set = 2068
```

In [8]:

```
HPWRA1.shape
```

Out[8]:

```
(2068, 51)
```

Manual Data Cleaning

the following changes were made to the UltimateDataCompilation.csv file in Microsoft Excel:

- Some of the missing scores and misaligned data set were modified.
- re-entered Tabebuia berteroi.xls - was shifted one column to the left
- re-entered Brya ebenus.pdf - all missing data columns were skipped.
- re-entered Archontophoenix alexandrae.xls - contained 0s instead of y/n
- re-entered Eucalyptus gradis.xls - there was an error on the original file

In [27]:

```
# need a dictionary
yesno_dict = {'no':"No", 'n':"No", ' n':"No", 'n ':"No", 'N':"No", ' N':"No",'N ':"No", 'yes':"Yes"
,
              'y':"Yes", ' y':"Yes", 'y ':"Yes", 'Y':"Yes", ' Y':"Yes",'Y ':"Yes",
              ' ':None, '  ':None, '?':None, '0':None, 'None':None, ' None':None}
```

In [11]:

```
# Global plant history Q's: let's change y/n questions to binary 0s and 1s
HPWRA1["1.01_domesticated"].replace(yesno_dict, inplace=True)
```

In [13]:

```
HPWRA1["1.02_naturalized_grown"].replace(yesno_dict, inplace=True)
HPWRA1["1.03_weedy_races"].replace(yesno_dict, inplace=True)
```

In [14]:

```
#Check:
HPWRA1["1.01_domesticated"].unique()
```

Out[14]:

```
array(['Yes', 'No', nan], dtype=object)
```

In [15]:

```
HPWRA1["1.02_naturalized_grown"].unique()
```

Out[15]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [16]:

```
HPWRA1["1.03_weedy_races"].unique()
```

Out[16]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [17]:

```
# we need dictionary for questions about climate suitability
lowmedhigh_dict = {"Low": "Low", "Intermediate": "Intermediate", "int" : "Intermediate", "Int":"Int
ermediate","High":"High", "0":"Low", "1":"Intermediate", "2":"High"}

# Climate suitability Q's: let's change the answers to 0s, 1s and 2s.
HPWRA1["2.01_clim_match"].replace(lowmedhigh_dict, inplace=True)
HPWRA1["2.02_climmatch_qual"].replace(lowmedhigh_dict, inplace=True)
```

In [18]:

```
#Check: there should be only 0,1,2s and Nan left
HPWRA1["2.01_clim_match"].unique()
```

Out[18]:

```
array(['Intermediate', 'High', 'Low', nan], dtype=object)
```

In [19]:

```
HPWRA1["2.02_climmatch_qual"].unique()
```

Out[19]:

```
array(['High', 'Intermediate', 'Low', nan], dtype=object)
```

In [20]:

```
# More climate suitability Q's, but these ones are y/n: let's change y/n questions to 0s, 1s
HPWRA1["2.03_broad_clim"].replace(yesno_dict, inplace=True)
HPWRA1["2.04_similar_clim"].replace(yesno_dict, inplace=True)
HPWRA1["2.05_repeat_intro"].replace(yesno_dict, inplace=True)
```

In [21]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1["2.03_broad_clim"].unique()
```

Out[21]:

```
array(['Yes', 'No', nan], dtype=object)
```

In [22]:

```
HPWRA1["2.04_similar_clim"].unique()
```

Out[22]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [28]:

```
HPWRA1["2.05_repeat_intro"].unique()
```

Out[28]:

```
array(['Yes', 'No', nan, None], dtype=object)
```

In [24]:

```
# Q's about how a plant has behaved elsewhere in the world: let's change y/n to 1/0
HPWRA1["3.01_beyond_native"].replace(yesno_dict, inplace=True)
HPWRA1["3.02_disturbance_weed"].replace(yesno_dict, inplace=True)
HPWRA1['3.03_agri_forestry_weed'].replace(yesno_dict, inplace=True)
HPWRA1['3.04_enviro_weed'].replace(yesno_dict, inplace=True)
HPWRA1['3.05_congener'].replace(yesno_dict, inplace=True)
```

In [25]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1["3.01_beyond_native"].unique()
```

Out[25]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [26]:

```
HPWRA1["3.02_disturbance_weed"].unique()
```

Out[26]:

```
array(['No', 'Yes', nan, None], dtype=object)
```

In [29]:

```
HPWRA1["3.03_agri_forestry_weed"].unique()
```

Out[29]:

```
array(['No', 'Yes', nan, None], dtype=object)
```

In [30]:

```
HPWRA1['3.04_enviro_weed'].unique()
```

Out[30]:

```
array(['No', 'Yes', nan, None], dtype=object)
```

In [31]:

```
HPWRA1['3.05_congener'].unique()
```

Out[31]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [32]:

```
# Q's about undesireable traits: let's convert y/n to binary 1/0s
HPWRA1['4.01_spiny'].replace(yesno_dict, inplace=True)
HPWRA1['4.02_allelopathic'].replace(yesno_dict, inplace=True)
HPWRA1['4.03_parasitic'].replace(yesno_dict, inplace=True)
HPWRA1['4.04_unpalatable'].replace(yesno_dict, inplace=True)
HPWRA1['4.05_toxic'].replace(yesno_dict, inplace=True)
HPWRA1['4.06_alternate_host'].replace(yesno_dict, inplace=True)
HPWRA1['4.07_allergies'].replace(yesno_dict, inplace=True)
HPWRA1['4.08_fire_hazard'].replace(yesno_dict, inplace=True)
HPWRA1['4.09_shade_tolerant'].replace(yesno_dict, inplace=True)
HPWRA1['4.10_tolerates_soilcond'].replace(yesno_dict, inplace=True)
HPWRA1['4.11_climber'].replace(yesno_dict, inplace=True)
HPWRA1['4.12_forms_thickets'].replace(yesno_dict, inplace=True)
```

In [33]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1['4.01_spiny'].unique()
```

Out[33]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [34]:

```
HPWRA1['4.02_allelopathic'].unique()
```

Out[34]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [35]:

```
HPWRA1['4.03_parasitic'].unique()
```

Out[35]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [36]:

```
HPWRA1['4.04_unpalatable'].unique()
```

Out[36]:

```
array(['Yes', nan, 'No', None], dtype=object)
```

In [37]:

```
HPWRA1['4.05_toxic'].unique()
```

Out[37]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [38]:

```
HPWRA1['4.06_alternate_host'].unique()
```

Out[38]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [39]:

```
HPWRA1['4.07_allergies'].unique()
```

Out[39]:

```
array(['No', nan, 'Yes', None], dtype=object)
```

In [40]:

```
HPWRA1['4.08_fire_hazard'].unique()
```

Out[40]:

```
array(['No', nan, 'Yes', None], dtype=object)
```

In [41]:

```
HPWRA1['4.09_shade_tolerant'].unique()
```

Out[41]:

```
array(['Yes', nan, 'No', None], dtype=object)
```

In [42]:

```
HPWRA1['4.10_tolerates_soilcond'].unique()
```

Out[42]:

```
array(['Yes', 'No', nan], dtype=object)
```

In [43]:

```
HPWRA1['4.11_climber'].unique()
```

Out[43]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [44]:

```
HPWRA1['4.12_forms_thickets'].unique()
```

Out[44]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [45]:

```
# Q's about whether they'll alter habits: these are y/n and should be changed to 1/0
HPWRA1['5.01_aquatic'].replace(yesno_dict, inplace=True)
HPWRA1['5.02_grass'].replace(yesno_dict, inplace=True)
HPWRA1['5.03_nitrogen_fixer'].replace(yesno_dict, inplace=True)
HPWRA1['5.04_geophyte'].replace(yesno_dict, inplace=True)
```

In [46]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1['5.01_aquatic'].unique()
```

Out[46]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [47]:

```
HPWRA1['5.02_grass'].unique()
```

Out[47]:

```
array(['No', 'Yes'], dtype=object)
```

In [48]:

```
HPWRA1['5.03_nitrogen_fixer'].unique()
```

Out[48]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [49]:

```
HPWRA1['5.04_geophyte'].unique()
```

Out[49]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [50]:

```
# Q's about whether they'll establish in Hawaii: these are y/n and should be changed to 1/0
HPWRA1['6.01_repro_failure'].replace(yesno_dict, inplace=True)
HPWRA1['6.02_viable_seed'].replace(yesno_dict, inplace=True)
HPWRA1['6.03_hybridizes'].replace(yesno_dict, inplace=True)
HPWRA1['6.04_selfcompatible'].replace(yesno_dict, inplace=True)
HPWRA1['6.05_special_pollinators'].replace(yesno_dict, inplace=True)
HPWRA1['6.06_vegetative_repro'].replace(yesno_dict, inplace=True)
```

In [51]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1['6.01_repro_failure'].unique()
```

Out[51]:

```
array([nan, 'No', 'Yes'], dtype=object)
```

In [52]:

```
HPWRA1['6.02_viable_seed'].unique()
```

Out[52]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [53]:

```
HPWRA1['6.03_hybridizes'].unique()
```

Out[53]:

```
array([nan, 'Yes', 'No', None], dtype=object)
```

In [54]:

```
HPWRA1['6.04_selfcompatible'].unique()
```

Out[54]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [55]:

```
HPWRA1['6.05_special_pollinators'].unique()
```

Out[55]:

```
array([nan, 'No', 'Yes'], dtype=object)
```

In [56]:

```
HPWRA1['6.06_vegetative_repro'].unique()
```

Out[56]:

```
array(['No', 'Yes', nan], dtype=object)
```

Minimum generation time question

- This is a question about how long a species takes to meet maturity. I binned these values according to Gordon et al 2010 (Guidance for addressing the Australian Weed Risk Assessment Questions, Plant Protection Quarterly 25(2): 56-74) where 1 = 0-2 years; 2 = 2-4 years; 3 = 4-10 years; 4 = >10 years

In [58]:

```
# making the dictionary
lifespan_dict ={'0':"0-2 years", '<1':"0-2 years", '1':"0-2 years", '1 year':"0-2 years", '2-Jan':"
0-2 years", '1or 2':"0-2 years", '1.5-2':"0-2 years", '1.5-2.5':"0-2 years", '>1':"0-2 years",
               '2':"2-4 years", '3':"2-4 years", '2 or 3':"2-4 years", '2 or 3 ':"2-4 years", '2+':
"2-4 years", '>2':"2-4 years", '2 or 3 years':"2-4 years", '3-Feb':"2-4 years", '>3':"2-4 years", '
<4':"2-4 years", '5-Mar':"2-4 years",'3+':"2-4 years",
               '4':"4-10 years", '4+':"4-10 years", '>4':"4-10 years", '>4+':"4-10 years",  '4+ ':
"4-10 years", '5':"4-10 years", '5+':"4-10 years", '6':"4-10 years", '7':"4-10 years", '7+':"4-10
years", '8':"4-10 years", '9':"4-10 years",
               '10':">10 years", '15':">10 years", '19':">10 years",'20':">10 years", '30':">10 ye
ars",
               'n':None}
# binning the responses
HPWRA1['6.07_miniumum_gen_time'].replace(lifespan_dict, inplace=True)
```

In [59]:

```
#Check: make sure there's nothing but 1-4 and NaN left
HPWRA1['6.07_miniumum_gen_time'].unique()
```

Out[59]:

```
array([nan, '2-4 years', '4-10 years', '0-2 years', '>10 years'],
      dtype=object)
```

In [60]:

```
# Q's about how easily a plant is dispersed: these are y/n and should be changed to 1/0
HPWRA1['7.01_unintentional_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.02_intentional_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.03_contaminant_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.04_wind_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.05_water_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.06_bird_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.07_animal_dispersal'].replace(yesno_dict, inplace=True)
HPWRA1['7.08_survive_gut'].replace(yesno_dict, inplace=True)
```

In [61]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1['7.01_unintentional_dispersal'].unique()
```

Out[61]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [62]:

```
HPWRA1['7.02_intentional_dispersal'].unique()
```

Out[62]:

```
array(['Yes', 'No', nan], dtype=object)
```

In [63]:

```
HPWRA1['7.03_contaminant_dispersal'].unique()
```

Out[63]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [64]:

```
HPWRA1['7.04_wind_dispersal'].unique()
```

Out[64]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [65]:

```
HPWRA1['7.05_water_dispersal'].unique()
```

Out[65]:

```
array(['No', nan, 'Yes'], dtype=object)
```

In [66]:

```
HPWRA1['7.06_bird_dispersal'].unique()
```

Out[66]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [67]:

```
HPWRA1['7.07_animal_dispersal'].unique()
```

Out[67]:

```
array(['No', 'Yes', nan], dtype=object)
```

In [68]:

```
HPWRA1['7.08_survive_gut'].unique()
```

Out[68]:

```
array(['No', nan, 'Yes', None], dtype=object)
```

In [69]:

```
# Q's about how easily a plant is controlled: these are y/n and should be changed to 1/0
HPWRA1['8.01_prolific_seeder'].replace(yesno_dict, inplace=True)
HPWRA1['8.02_propagule_bank'].replace(yesno_dict, inplace=True)
HPWRA1['8.03_herbicide_controlled'].replace(yesno_dict, inplace=True)
HPWRA1['8.04_tolerates_mutilation'].replace(yesno_dict, inplace=True)
HPWRA1['8.05_local_enemies'].replace(yesno_dict, inplace=True)
```

In [70]:

```
#Check: make sure there's nothing but 0s,1s, and NaN left
HPWRA1['8.01_prolific_seeder'].unique()
```

Out[70]:

```
array(['No', None, nan, 'Yes'], dtype=object)
```

In [71]:

```
HPWRA1['8.02_propagule_bank'].unique()
```

Out[71]:

```
array(['No', 'Yes', nan, None], dtype=object)
```

In [72]:

```
HPWRA1['8.03_herbicide_controlled'].unique()
```

Out[72]:

```
array([nan, 'Yes', 'No', None], dtype=object)
```

In [73]:

```
HPWRA1['8.04_tolerates_mutilation'].unique()
```

Out[73]:

```
array(['Yes', nan, 'No'], dtype=object)
```

In [74]:

```
HPWRA1['8.05_local_enemies'].unique()
```

Out[74]:

```
array([nan, 'No', 'Yes', None], dtype=object)
```

Adding date of assessment and final risk category

The date each assessment was conducted, and the categorical risk assessment was not present on each of the scraped pdfs, so we need to add this information (downloadable here https://sites.google.com/site/weedriskassessment/home). Thankfully, each spreadsheet contains a "Filename" column, so we can match the datasets by these values.

In [75]:

```
# read in the necessary fields from the summary csv
#fields = ["Genus","Species", "Synonyms", "Common_name", "WRA_score", "WRA_rating",
"WRA_designation", "Date", "Filename"]
HPWRA_summary =pd.read_csv("All_HPWRA_Risk.csv", sep=',')#, usecols=fields)
```

In [76]:

```
# Check:make sure it loaded okay and that the newest assessed species are on there
HPWRA_summary.tail()
```

Out[76]:

	Family	Taxa	Genus	Species	Synonyms	Common name	WRA_score	WRA_rating	WRA_designation	I
2036	Solanaceae	Lycium barbarum	Lycium	barbarum	Lycium halimifolium, Lycium vulgare	goji berry, matrimony vine, Chinese boxthorn	15.0	High Risk	H (HPWRA)	8/28/2
2037	Apocynaceae	Strophanthus amboensis	Strophanthus	amboensis	Strophanthus gossweileri	elephant vine, knob-stemmed poisonrope	1.0	Evaluate	Evaluate	9/4/2
2038	Malvaceae	Abroma augusta	Abroma	augusta	Abroma fastuosum, Ambroma augustum	devil's cotton	4.0	Evaluate	Evaluate	9/9/2
2039	Dicksoniaceae	Dicksonia squarrosa	Dicksonia	squarrosa	Trichomanes squarrosum	harsh tree fern, rough tree fern, wheki	18.0	High Risk	H (HPWRA)	9/11/2
2040	Myrtaceae	Syzygium polyanthum	Syzygium	polyanthum	Eugenia polyantha	Indian bayleaf, Indonesian bayleaf	3.0	High Risk	H (HPWRA)	9/13/2

In [77]:

```
#Check: How many assessments have been completed?
print("Number of Species that Have been Assessed = " + str(len(HPWRA_summary)))
```

```
Number of Species that Have been Assessed = 2041
```

- The number of assessments scraped from pdf and xls does not match up
- This is because some species have more than one assessment if it has been updated in recent years
- Must be careful to ensure that the Assessment we use is the most recent

In [78]:

```
var1 = len(CompiledData) - len(HPWRA_summary)

print("Number of Species that have more than 1 assessment = " + str(var1))
```

```
Number of Species that have more than 1 assessment = 28
```

In [79]:

```
#merging the
HPWRAall = pd.merge(HPWRA_summary, HPWRA1, on="Filename", how="left")
```

In [80]:

```
HPWRAall.shape
```

Out[80]:

```
(2048, 62)
```

In [81]:

```
HPWRAall.to_csv('HPWRAlist.csv',encoding='utf-8-sig', index=False)
```

In [82]:

```
HPWRAall.head(5)
```

Out[82]:

	Family	Taxa	Genus	Species	Synonyms	Common name	WRA_score	WRA_rating	WRA_designation	Date	..
0	Fabaceae	Acacia auriculiformis	Acacia	auriculiformis	NaN	Darwin black wattle	13.0	High Risk	H (HPWRA)	10/7/2002	..
1	Fabaceae	Acacia confusa	Acacia	confusa	NaN	Formosan koa	10.0	High Risk	H (Hawaii)	10/7/2002	..
2	Fabaceae	Acacia melanoxylon	Acacia	melanoxylon	NaN	Australian blackwood	12.0	High Risk	H (HPWRA)	10/7/2002	..
3	Euphorbiaceae	Acalypha hispida	Acalypha	hispida	NaN	chenille plant	2.0	Low Risk	L (HPWRA)	10/7/2002	..
4	Euphorbiaceae	Acalypha wilkesiana	Acalypha	wilkesiana	NaN	beefsteak plant	-2.0	Low Risk	L (HPWRA)	10/7/2002	..

5 rows × 62 columns

In [83]:

```
# how many assessments didn't get scraped and aren't included in our dataset?
missing_assessments = HPWRAall[HPWRAall["manual_score"].isnull()]
missing_assessments.shape
```

Out[83]:

```
(4, 62)
```

- The following are species that have HPWRA that could not be included because the original data file is corrupted in some way.

In [84]:

```
missing_assessments.head(5)
```

Out[84]:

	Family	Taxa	Genus	Species	Synonyms	Common name	WRA_score	WRA_rating	WRA_designation	
393	Araceae	Anthurium hookeri	Anthurium	hookeri	NaN	birds nest anthurium	-6.0	Low Risk	L (HPWRA)	8/6
725	Zingiberaceae	Kaempferia galanga	Kaempferia	galanga	NaN	galanga	1.0	Low Risk	L (HPWRA)	7/2
1122	Fabaceae	Leucaena 'KX2'	Leucaena	KX2'	NaN	KX2	3.0	Evaluate	Evaluate	3/9
1123	Fabaceae	Leucaena 'Wondergraze'	Leucaena	Wondergraze'	NaN	Wondergraze	7.0	High Risk	H (HPWRA)	3/12

4 rows × 62 columns