Registry Spring Cleaning

Tess Jaffe, Registry WG Vice Chair

(after consulting *not at all* with better informed IVOA'ns at EuroVO and GAVO)

```
import os ●●●
```

Registry Spring Cleaning notebook

Following up on Markus' Confessions of a Registry Janitor, I propose some regular checks of the metadata. We already have checks of the validity of services, for instance, in the Operations group weather reports. This would be compolementary.

Check 1: spot check numbers between different registries

```
def compare( query ): ...
```

[18]:
```
compare("select * from rr.capability where standard_id like '%hips%'")
```

```
NAVO RegTAP finds 20
GAVO RegTAP finds 570
EUVO RegTAP finds 570
```

[3]:
```
compare("select * from rr.capability where standard_id like '%sia%'")
```

```
NAVO RegTAP finds 0
GAVO RegTAP finds 382
EUVO RegTAP finds 376
```

[4]:
```
compare("select * from rr.capability where standard_id like '%cone%' and ivoid not like '%vizier%'")
```

```
NAVO RegTAP finds 1948
GAVO RegTAP finds 1922
EUVO RegTAP finds 1962
```

[ ]:

Check 2: UCDs

- Check 2a: are the UCDs valid according to astropy.io.votable.ucd.check_ucd

```
from astropy.io.votable.ucd import check_ucd •••
```

```
Found 58 invalid UCDs
  The top 10 bad UCD values by number of instances are
                          : 210478
??                        : 30342
vox:image_filesize        : 122
????                      : 70
vox:image_mjdateobs       : 54
image?                    : 49
vox:bandpass_id           : 42
???                       : 36
vox:bandpass_hilimit      : 34
vox:bandpass_lolimit      : 34
```

```
colons = [] •••
```

```
culprits = [] •••
```

```
Found 797 that are not valid under UCD1+ controlled vocabulary
  The top 10 bad UCD values by number of instances are
: 210478
??: 30342
error: 13825
code_misc: 8538
phot_mag: 6291
fit_param: 4505
obs.field: 4296
number: 3090
id_number: 2695
phot_intensity_adu: 2512
```

Check 3: authors

Have

- Last F.
- Last F., Last2 F.
- Last, F.
- F. Last, Last2. F.

At least where there are commas they are used to separate two authors, rather than "Last, F" or something.

[8]:
```
names = gavo_regtap.search("select role_name, count(*) as cnt from rr.res_role where base_role = 'creator' gro
names
```

object	int32
Temaj D.	1
Beccari G.,Rood R.T.	1
Matarrese S.,Matthai F.	1
Dworak T.Z.	1
	11
Tello C.	1
Laurikainen, E.	2
Perez-Gonzalez P. G.	1
Lucas W.	1
...	...

[ ]:

Check 4: subjects and the UAT

[9]:
```
subjects = gavo_regtap.search("select res_subject, count(*) as cnt from rr.res_subject group by res_subject or
subjects
```

[9]: *Table length=1263*

res_subject	cnt
object	int32
Optical astronomy	6627
Galaxies	4141
Infrared photometry	4120
Spectroscopy	3836
Photometry	3535
Radial velocity	2652
Surveys	2547
Redshifted	2474

Check 4 continued

```
# Generator that goes through the nested JSON and looks for a key anywhere down in it •••
```

```
uat_name_list = [x.lower() for x in item_generator(uat,'name')] •••
```

```
culprits = [] •••
```

```
Found 872 Registry res_subject entries that are not in the UAT
  The top 10 bad subject values by number of instances are
Wide-band photometry: 2276
Survey Source: 651
Sky survey: 445
asteroid-dynamics: 194
: 153
visible-astronomy: 139
virtual-observatories: 138
extragalactic survey: 101
Observation: 81
Stars: 80
```

13]:
```
[None, None, None, None, None, None, None, None, None, None]
```

14]:
```
import re
result = [u for u in uat_name_list if re.search("^star.*",u)]
print(f"Found {len(result)} matches to 'star' such as")
print(result[0:10])
```

```
Found 17 matches to 'star' such as
['star-planet interactions', 'starburst galaxies', 'starburst galaxies', 'starburst galaxies', 'star atlases',
'star counts', 'star counts', 'star lore', 'starspots', 'starspots']
```

Check 4b: concepts

[15]:
```
reg_uat_concept_list = gavo_regtap.search("select distinct uat_concept from rr.subject_uat").to_table()["uat_co
print(f"There are {len(reg_uat_concept_list)} distinct uat_concept values in the registry's subject_uat table")
```

```
There are 451 distinct uat_concept values in the registry's subject_uat table
```

[16]:
```
bad=[]
for c in reg_uat_concept_list:
    # lower case and replace - with space
    if c.lower().replace("-"," ") not in uat_name_list:
        bad.append(c)
print(f"There are {len(bad)} concepts not found in the UAT such as:")
print(bad[0:10])
```

```
There are 47 concepts not found in the UAT such as:
['x-ray-binary-stars', 'hubble-space-telescope ', 'survey', 'early-type-stars', 'stellar-atmosphers', 'gamma-r
ay-bursts', 'post-asymptotic-giant-branch-stars', 'planetary-magnetosphere', 'kolmogorov-smirnov-test', 'pre-m
ain-sequence-stars']
```

To be expanded. Now what to do with this?

- Report cross-checks between registries to their admins.
- Compile a report of issues as above and advertise at IVOA Interop's Registry (or Ops?) session.
- Compile a report of issues found for each publisher and email them yearly to request updates.