PSet1_Q1_ARE213

October 2, 2023

[12]:
```
import pandas as pd
import numpy as np
#import data packages
```

[13]:
```
file = pd.read_csv('Pset1.csv')
#read in the data
```

[14]:
```
#i understand that this is ugly and i'm so sorry. so sorry.
file['cardiac'] = file['cardiac'].replace({9: None} )
file['lung'] = file['lung'].replace({9: None} )
file['diabetes'] = file['diabetes'].replace({9: None} )
file['herpes'] = file['herpes'].replace({9: None} )
file['herpes'] = file['herpes'].replace({8: None} ) #could be this problem
file['chyper'] = file['chyper'].replace({9: None} )
file['phyper'] = file['phyper'].replace({9: None} )
file['pre4000'] = file['pre4000'].replace({9: None} )
file['preterm'] = file['preterm'].replace({9: None} )
file['tobacco'] = file['tobacco'].replace({9: None} )
file['cigar6'] = file['cigar6'].replace({6: None} )
file['alcohol'] = file['alcohol'].replace({9: None})
file['wgain'] = file['wgain'].replace({99:None})
file['drink5'] = file['drink5'].replace({5:None})
#replace missing data codes with NaN values
```

[15]:
```
#replace [1,2] indicators with [0,1] indicators
indic_vars = ['rectype', 'pldel3', 'dmar', 'csex', 'anemia', 'cardiac', 'lung',␣
 ↪'diabetes', 'herpes',
       'chyper', 'phyper', 'pre4000', 'preterm', 'tobacco', 'alcohol']
for it in indic_vars:
    file[it] = file[it].replace([1,2], [1,0])
```

[16]:
```
#create dfs of category-> indicator variables
mrace3ind = pd.get_dummies(file['mrace3'], prefix = 'mrace3')
file['orfathhis'] = file['orfath'].replace([1,2,3,4,5], [1,1,1,1,1])
file['ormothhis'] = file['ormoth'].replace([1,2,3,4,5], [1,1,1,1,1])
file['educ_0.0'] = file['dmeduc'].
 ↪replace([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],␣
 ↪[0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0])
```

```
file['educ_1.0'] = file['dmeduc'].
 ↪replace([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],␣
 ↪[0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0])
file['educ_2.0'] = file['dmeduc'].
 ↪replace([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],␣
 ↪[0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1])
adind = pd.get_dummies(file['adequacy'], prefix = 'adeq')
livebirind = pd.get_dummies(file['isllb10'], prefix = 'live')
totalordind = pd.get_dummies(file['totord9'], prefix = 'tot')
pluralind = pd.get_dummies(file['dplural'], prefix = 'plur')

#concatenate indicator variables to main dataframe
data_clean = pd.concat([file, mrace3ind, adind, cntocind, livebirind,␣
 ↪totalordind, pluralind], axis=1)
```

[17]:
```
#create dataframe for analysis with dropped nulls
data_clean_a = data_clean.dropna()
```

[18]:
```
#create dataframe for balance table comparison to check for random (or␣
 ↪nonrandom) missing values
cols = ['variable', 'no_null_mean', 'no_null_sd', 'all_mean', 'all_sd', 'diff',␣
 ↪'se_diff']
balance_t = pd.DataFrame(columns = cols)

interesting_vars = ['pldel3', 'dmage', 'dmeduc', 'dmar', 'adequacy', 'dgestat',␣
 ↪'csex', 'dbrwt',
                    'dplural', 'omaps', 'fmaps', 'alcohol', 'pre4000',␣
 ↪'preterm', 'mrace3_1', 'mrace3_2', 'mrace3_3',]

#append means, standard errors, and differences in both from the null and␣
 ↪non-null datasets
for name in interesting_vars:
    vals = []
    vals.append(name)
    vals.append(data_clean_a[name].mean())
    vals.append(data_clean_a[name].std())
    vals.append(data_clean[name].mean())
    vals.append(data_clean[name].std())
    vals.append(data_clean_a[name].mean()-data_clean[name].mean())
    vals.append(np.sqrt(((float(data_clean_a[name].std())**2)/
 ↪float(data_clean_a[name].notnull().size)+((float(data_clean[name].std())**2)/
 ↪float(data_clean_a[name].notnull().size)))))
    balance_t = balance_t.append(pd.DataFrame([vals], columns = cols),␣
 ↪ignore_index = True)
#show table
print(balance_t)
```

```
#table to latex

#proving we can print a latex table in python; all following will come from stata
#print(balance_t.to_latex(index=False,

                  #formatters={"variable": str.upper},

                  #float_format="{:.1f}".format,))
```

	variable	no_null_mean	no_null_sd	all_mean	all_sd	diff	\
0	pldel3	0.981904	0.133300	0.980135	0.139535	0.001768	
1	dmage	27.756662	5.698714	27.272290	5.844527	0.484371	
2	dmeduc	13.210828	2.272132	12.939282	2.293448	0.271546	
3	dmar	0.748861	0.433670	0.678667	0.466990	0.070194	
4	adequacy	1.297060	0.546108	1.362046	0.610569	-0.064986	
5	dgestat	39.152901	2.444981	39.028313	2.708495	0.124588	
6	csex	0.514501	0.499792	0.513199	0.499827	0.001303	
7	dbrwt	3373.290760	585.174817	3335.836640	612.878055	37.454119	
8	dplural	1.028104	0.174365	1.028038	0.173817	0.000066	
9	omaps	8.117416	1.259542	8.077086	1.340189	0.040330	
10	fmaps	9.009214	0.706982	8.991048	0.820058	0.018166	
11	alcohol	0.009694	0.097979	0.031476	0.174600	-0.021782	
12	pre4000	0.014484	0.119475	0.013481	0.115324	0.001003	
13	preterm	0.014135	0.118048	0.016269	0.126509	-0.002134	
14	mrace3_1	0.860518	0.346450	0.827330	0.377963	0.033188	
15	mrace3_2	0.019815	0.139365	0.020769	0.142610	-0.000954	
16	mrace3_3	0.119667	0.324573	0.151901	0.358926	-0.032235	

	se_diff
0	0.000570
1	0.024112
2	0.009536
3	0.001882
4	0.002420
5	0.010778
6	0.002088
7	2.503029
8	0.000727
9	0.005433
10	0.003198
11	0.000591
12	0.000490
13	0.000511
14	0.001515
15	0.000589
16	0.001429

[19]:
```
cols = ['variable', 'mean', 'sd', 'mean no smoking', 'sd no smoking', 'mean with␣
 ↪smoking', 'sd with smoking']
balance_t = pd.DataFrame(columns = cols)

#append means, standard errors, and differences in both from the smoking and␣
 ↪non-smoking sections
for name in interesting_vars:
    vals = []
    vals.append(name)
    vals.append(data_clean_a[name].mean())
    vals.append(data_clean_a[name].std())
    vals.append(data_clean_a[data_clean_a['tobacco']==1][name].mean())
    vals.append(data_clean_a[data_clean_a['tobacco']==1][name].std())
    vals.append(data_clean_a[data_clean_a['tobacco']==0][name].mean())
    vals.append(data_clean_a[data_clean_a['tobacco']==0][name].std())
    balance_t = balance_t.append(pd.DataFrame([vals], columns = cols),␣
 ↪ignore_index = True)
print(balance_t)

#proving we can print a latex table in python; all following will come from stata
#print(balance_t.to_latex(index=False,

                #formatters={"variable": str.upper},

                #float_format="{:.1f}".format,))
```

```
    variable         mean          sd  mean no smoking  sd no smoking  \
0     pldel3     0.981904    0.133300         0.996715       0.057220
1      dmage    27.756662    5.698714        26.173437       5.605766
2     dmeduc    13.210828    2.272132        11.986587       1.633245
3       dmar     0.748861    0.433670         0.517847       0.499695
4   adequacy     1.297060    0.546108         1.411311       0.629802
5    dgestat    39.152901    2.444981        39.046808       2.709646
6       csex     0.514501    0.499792         0.518012       0.499689
7      dbrwt  3373.290760  585.174817      3171.139166     572.084454
8    dplural     1.028104    0.174365         1.022556       0.150682
9      omaps     8.117416    1.259542         8.102759       1.265606
10     fmaps     9.009214    0.706982         9.009088       0.707029
11   alcohol     0.009694    0.097979         0.034983       0.183742
12   pre4000     0.014484    0.119475         0.008431       0.091435
13   preterm     0.014135    0.118048         0.024581       0.154849
14  mrace3_1     0.860518    0.346450         0.869156       0.337239
15  mrace3_2     0.019815    0.139365         0.003778       0.061347
16  mrace3_3     0.119667    0.324573         0.127067       0.333057

    mean with smoking  sd with smoking
0            0.979096         0.143065
```

1	28.056828	5.666528
2	13.442934	2.301660
3	0.792660	0.405404
4	1.275399	0.525961
5	39.173015	2.390986
6	0.513836	0.499811
7	3411.616977	579.731321
8	1.029156	0.178482
9	8.120194	1.258376
10	9.009238	0.706977
11	0.004899	0.069822
12	0.015631	0.124046
13	0.012154	0.109575
14	0.858881	0.348146
15	0.022856	0.149444
16	0.118264	0.322922

[20]:
```
#export dataset as csv
data_clean_a.to_csv('clean_pset1.csv')
```

[ ]:
```

```