nhanes_bp

February 24, 2020

Fit mixed models to NHANES blood pressure data. There are two blood pressure measurement types (systolic and diastolic), with up to 4 repeated measures for each type.

https://wwwn.cdc.gov/Nchs/Nhanes/2011-2012/DEMO_G.XPT
https://wwwn.cdc.gov/Nchs/Nhanes/2011-2012/BPX_G.XPT https://wwwn.cdc.gov/Nchs/Nhanes/2011-2012/BMX_G.XPT

[1]:
```
import statsmodels.api as sm
import pandas as pd
import numpy as np
```

```
/nfs/kshedden/python3/lib/python3.7/site-
packages/statsmodels/compat/pandas.py:23: FutureWarning: The Panel class is
removed from pandas. Accessing it from the top-level namespace will also be
removed in the next version
  data_klasses = (pandas.Series, pandas.DataFrame, pandas.Panel)
```

Load and merge the data sets

[2]:
```
demog = pd.read_sas("../data/DEMO_G.XPT")
bpx = pd.read_sas("../data/BPX_G.XPT")
bmx = pd.read_sas("../data/BMX_G.XPT")
df = pd.merge(demog, bpx, left_on="SEQN", right_on="SEQN")
df = pd.merge(df, bmx, left_on="SEQN", right_on="SEQN")
```

Convert from wide to long

[3]:
```
syvars = ["BPXSY%d" % j for j in (1,2,3,4)]
divars = ["BPXDI%d" % j for j in (1,2,3,4)]
vvars = syvars + divars
idvars = ['SEQN', 'RIDAGEYR', 'RIAGENDR', 'BMXBMI']
dx = pd.melt(df, id_vars=idvars, value_vars=vvars,
             var_name='bpvar', value_name='bp')
```

A bit of data cleanup

[4]:
```
dx = dx.sort_values(by='SEQN')
dx = dx.reset_index(drop=True)
dx['SEQN'] = dx.SEQN.astype(np.int)
dx = dx.dropna()

# Blood pressure type (systolic or diastolic)
```

```
dx["bpt"] = dx.bpvar.str[3:5]

dx["bpi"] = dx.bpvar.str[5].astype(np.int)
dx["female"] = (dx.RIAGENDR == 2).astype(np.int)

di_mean = dx.loc[dx.bpt=="DI", :].groupby("SEQN")["bp"].aggregate(np.mean)
di_mean.name = "di_mean"
dx = pd.merge(dx, di_mean, left_on="SEQN", right_index=True)

print(dx.head())
```

```
    SEQN  RIDAGEYR  RIAGENDR  BMXBMI   bpvar     bp bpt  bpi  female  \
0  62161      22.0       1.0    23.3  BPXSY1  110.0  SY    1       0
2  62161      22.0       1.0    23.3  BPXDI2   68.0  DI    2       0
3  62161      22.0       1.0    23.3  BPXSY3  118.0  SY    3       0
4  62161      22.0       1.0    23.3  BPXDI3   74.0  DI    3       0
5  62161      22.0       1.0    23.3  BPXSY2  104.0  SY    2       0

     di_mean
0  74.666667
2  74.666667
3  74.666667
4  74.666667
5  74.666667
```

Subsample to make the script run faster

[5]:
```
dx = dx.iloc[0:10000, :]
```

Fit a linear mean structure model using OLS. The variance structure of this model is misspecified.

[6]:
```
model1 = sm.OLS.from_formula("bp ~ RIDAGEYR + female + C(bpt) + BMXBMI", dx)
result1 = model1.fit()
print(result1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     bp   R-squared:                       0.764
Model:                            OLS   Adj. R-squared:                  0.764
Method:                 Least Squares   F-statistic:                     8082.
Date:                Mon, 24 Feb 2020   Prob (F-statistic):               0.00
Time:                        00:13:49   Log-Likelihood:                -41438.
No. Observations:               10000   AIC:                         8.289e+04
Df Residuals:                    9995   BIC:                         8.292e+04
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
```

```
--------------------------------------------------------------------------------
Intercept       45.3352      0.624     72.668      0.000      44.112      46.558
C(bpt)[T.SY]    52.3704      0.305    171.623      0.000      51.772      52.969
RIDAGEYR         0.3017      0.007     41.102      0.000       0.287       0.316
female          -3.0799      0.307    -10.044      0.000      -3.681      -2.479
BMXBMI           0.4038      0.023     17.790      0.000       0.359       0.448
==============================================================================
Omnibus:                      935.341   Durbin-Watson:                   1.177
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6588.569
Skew:                          -0.114   Prob(JB):                         0.00
Kurtosis:                       6.970   Cond. No.                         212.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

Fit a mixed model to systolic data with a simple random intercept per subject. Then calculate ICC.

[7]:
```
ds2 = dx.loc[dx.bpt == "SY"]
model2 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + BMXBMI",
                                 groups="SEQN", data=ds2)
result2 = model2.fit()
icc2 = result2.cov_re / (result2.cov_re + result2.scale)
print(result2.summary())
print("icc=%f\n" % icc2.values.flat[0])
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:            MixedLM Dependent Variable: bp
No. Observations: 5000    Method:             REML
No. Groups:       1675    Scale:              15.8606
Min. group size:  1       Likelihood:         -17109.8829
Max. group size:  3       Converged:          Yes
Mean group size:  3.0
---------------------------------------------------------
               Coef.  Std.Err.    z   P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept      96.079    1.424 67.454 0.000 93.288 98.871
RIDAGEYR        0.424    0.017 24.502 0.000  0.390  0.457
female         -3.969    0.722 -5.493 0.000 -5.385 -2.553
BMXBMI          0.307    0.053  5.742 0.000  0.202  0.411
SEQN Var      211.111    2.305
=========================================================

icc=0.930121
```

Partial out the mean diastolic blood pressure per subject

[8]:
```
model3 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + BMXBMI + di_mean",
                                 groups="SEQN", data=ds2)
result3 = model3.fit()
icc3 = result3.cov_re / (result3.cov_re + result3.scale)
print(result3.summary())
print("icc=%f\n" % icc3.values.flat[0])
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:            MixedLM Dependent Variable: bp
No. Observations: 5000    Method:             REML
No. Groups:       1675    Scale:              15.8604
Min. group size:  1       Likelihood:         -17027.5692
Max. group size:  3       Converged:          Yes
Mean group size:  3.0
---------------------------------------------------------
               Coef.  Std.Err.    z    P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept      80.987    1.762 45.958 0.000 77.533 84.441
RIDAGEYR        0.365    0.017 21.502 0.000  0.332  0.399
female         -3.273    0.689 -4.752 0.000 -4.623 -1.923
BMXBMI          0.145    0.052  2.774 0.006  0.042  0.247
di_mean         0.322    0.024 13.383 0.000  0.275  0.369
SEQN Var      190.267    2.083
=========================================================

icc=0.923056
```

Fit a mixed model to diastolic data only with simple random intercept per subject. Then calculate ICC.

[9]:
```
ds3 = dx.loc[dx.bpt == "DI"]
model4 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + BMXBMI",
                                 groups="SEQN", data=ds3)
result4 = model4.fit()
icc4 = result4.cov_re / (result4.cov_re + result4.scale)
print(result4.summary())
print("icc=%f\n" % icc4.values.flat[0])
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:            MixedLM Dependent Variable: bp
No. Observations: 5000    Method:             REML
No. Groups:       1675    Scale:              36.5421
Min. group size:  1       Likelihood:         -18442.0035
Max. group size:  3       Converged:          Yes
```

```
Mean group size:  3.0
--------------------------------------------------------
               Coef.  Std.Err.   z    P>|z| [0.025 0.975]
--------------------------------------------------------
Intercept      46.900    1.378 34.037 0.000 44.199 49.601
RIDAGEYR        0.181    0.017 10.794 0.000  0.148  0.213
female         -2.161    0.699 -3.091 0.002 -3.530 -0.791
BMXBMI          0.503    0.052  9.736 0.000  0.402  0.605
SEQN Var      190.243    1.421
========================================================

icc=0.838869
```

Fit a mixed model to diastolic data only with simple random intercept per subject (also using subset of data).

[10]:

```
ds3 = dx.loc[dx.bpt == "DI"]
model5 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + BMXBMI + bpi",
                                 groups="SEQN", re_formula="1+bpi",
                                 data=ds3)
result5 = model5.fit()
print(result5.summary())
```

```
/nfs/kshedden/python3/lib/python3.7/site-packages/statsmodels/base/model.py:512:
ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check
mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/nfs/kshedden/python3/lib/python3.7/site-
packages/statsmodels/regression/mixed_linear_model.py:2059: ConvergenceWarning:
Retrying MixedLM optimization with lbfgs
  ConvergenceWarning)

          Mixed Linear Model Regression Results
=========================================================
Model:              MixedLM Dependent Variable: bp
No. Observations:   5000    Method:             REML
No. Groups:         1675    Scale:              34.1496
Min. group size:    1       Likelihood:         -18393.5906
Max. group size:    3       Converged:          Yes
Mean group size:    3.0
---------------------------------------------------------
                Coef.  Std.Err.   z    P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept       47.989    1.339 35.848 0.000 45.365 50.613
RIDAGEYR         0.190    0.016 11.744 0.000  0.158  0.222
female          -2.179    0.675 -3.227 0.001 -3.503 -0.855
BMXBMI           0.487    0.050  9.733 0.000  0.389  0.585
bpi             -0.501    0.107 -4.681 0.000 -0.711 -0.292
```

```
SEQN Var       143.301    1.861
SEQN x bpi Cov   7.899    0.340
bpi Var          2.576    0.168
==========================================================
```

Fit a mixed model to both types of BP with simple random intercept per subject.

[11]:
```
model6 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + C(bpt) + BMXBMI",
                                 groups="SEQN", data=dx)
result6 = model6.fit()
print(result6.summary())
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:             MixedLM Dependent Variable: bp
No. Observations: 10000   Method:              REML
No. Groups:        1675    Scale:               115.2323
Min. group size:   2       Likelihood:          -39569.3038
Max. group size:   6       Converged:           Yes
Mean group size:   6.0
---------------------------------------------------------
              Coef.  Std.Err.    z    P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept     45.311    1.140  39.759 0.000 43.078 47.545
C(bpt)[T.SY]  52.370    0.215 243.932 0.000 51.950 52.791
RIDAGEYR       0.302    0.014  21.929 0.000  0.275  0.329
female        -3.068    0.575  -5.332 0.000 -4.196 -1.940
BMXBMI         0.405    0.043   9.511 0.000  0.321  0.488
SEQN Var     117.916    0.485
=========================================================
```

Fit a mixed model to both types of BP with subject random intercept and unique random effect per BP type with common variance.

[12]:
```
model7 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + C(bpt) + BMXBMI",
                                 groups="SEQN", re_formula="1",
                                 vc_formula={"bpt": "0+C(bpt)"},
                                 data=dx)
result7 = model7.fit()
print(result7.summary())
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:             MixedLM Dependent Variable: bp
No. Observations: 10000   Method:              REML
No. Groups:        1675    Scale:               26.2016
Min. group size:   2       Likelihood:          -35818.3134
```

```
Max. group size:  6       Converged:          Yes
Mean group size:  6.0
---------------------------------------------------------
              Coef.  Std.Err.    z    P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept     45.306    1.155  39.221 0.000 43.042 47.570
C(bpt)[T.SY]  52.363    0.433 120.854 0.000 51.514 53.213
RIDAGEYR       0.302    0.014  21.931 0.000  0.275  0.329
female        -3.064    0.576  -5.324 0.000 -4.192 -1.936
BMXBMI         0.405    0.043   9.519 0.000  0.322  0.489
SEQN Var      58.744    1.088
bpt Var      148.396    1.188
=========================================================
```

Fit a mixed model to both types of BP with subject random intercept and unique random effect per BP type with unique variance.

[13]:
```
dx["sy"] = (dx.bpt == "SY").astype(np.int)
dx["di"] = (dx.bpt == "DI").astype(np.int)
model8 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + C(bpt) + BMXBMI",
                                 groups="SEQN", re_formula="1",
                                 vc_formula={"sy": "0+sy", "di": "0+di"},
                                 data=dx)
result8 = model8.fit()
print(result8.summary())
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:            MixedLM Dependent Variable: bp
No. Observations: 10000   Method:             REML
No. Groups:       1675    Scale:              26.2024
Min. group size:  2       Likelihood:         -35817.2945
Max. group size:  6       Converged:          Yes
Mean group size:  6.0
---------------------------------------------------------
              Coef.  Std.Err.    z    P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept     45.383    1.154  39.335 0.000 43.121 47.644
C(bpt)[T.SY]  52.364    0.433 120.853 0.000 51.514 53.213
RIDAGEYR       0.296    0.014  20.628 0.000  0.268  0.324
female        -3.021    0.576  -5.245 0.000 -4.150 -1.892
BMXBMI         0.410    0.043   9.608 0.000  0.326  0.493
SEQN Var      58.734    1.087
di Var       140.838    1.533
sy Var       155.963    1.620
=========================================================
```

Fit a mixed model to both types of BP with subject random intercept and unique random effect per BP type with unique variance, and heteroscedasticity by BP type.

[14]:

```
dx["sy1"] = (dx.bpvar == "BPXSY1").astype(np.int)
dx["sy2"] = (dx.bpvar == "BPXSY2").astype(np.int)
dx["sy3"] = (dx.bpvar == "BPXSY3").astype(np.int)
dx["di1"] = (dx.bpvar == "BPXDI1").astype(np.int)
dx["di2"] = (dx.bpvar == "BPXDI2").astype(np.int)
dx["di3"] = (dx.bpvar == "BPXDI3").astype(np.int)
model9 = sm.MixedLM.from_formula("bp ~ RIDAGEYR + female + C(bpt) + BMXBMI",
                                 groups="SEQN", re_formula="1",
                                 vc_formula={"sy": "0+sy", "di": "0+di",
                                             "dye": "0+di1+di2+di3"},
                                 data=dx)
result9 = model9.fit()
print(result9.summary())
```

```
          Mixed Linear Model Regression Results
=========================================================
Model:             MixedLM Dependent Variable: bp
No. Observations:  10000   Method:             REML
No. Groups:        1675    Scale:              15.7773
Min. group size:   2       Likelihood:         -35522.5285
Max. group size:   6       Converged:          Yes
Mean group size:   6.0
---------------------------------------------------------
              Coef.  Std.Err.    z    P>|z| [0.025 0.975]
---------------------------------------------------------
Intercept     45.423    1.155  39.327 0.000 43.159 47.687
C(bpt)[T.SY]  52.349    0.434 120.675 0.000 51.499 53.199
RIDAGEYR       0.296    0.014  20.592 0.000  0.268  0.324
female        -3.027    0.576  -5.251 0.000 -4.157 -1.897
BMXBMI         0.409    0.043   9.585 0.000  0.326  0.493
SEQN Var      58.784    1.426
di Var       138.227    2.057
dye Var       21.215    0.323
sy Var       159.491    2.206
=========================================================
```