multivariate_t5

November 21, 2024

1 Multivariate statistics Test 5: Hierarchical Linear Modeling

Student: Aleksandr Jan Smoliakov, VU MIF Data Science MSc year 1
Date: 2024-11-21

Data: File `hsb12.sav`, variables

- `school` - school's id
- `student` - student's id
- `minority` - 1 if ethnical minority, 0 - if not
- `female` - 1 if female, 0 if male
- `ses` - social –economic status
- `cses` - centered social-economic status
- `meanses` - school's average ses
- `mathach` - mathematical achievements
- `size` - number of students at school
- `sector` - 1 for catholic school, 0 for the state school
- `pracad` - proportion of students in the academic track
- `himinty` - 1 if over 40% of students are from ethnical minorities, 0 if less than 40%

Task: Create an HLM model for `mathach`.

First of all, let's load the data and take a look.

[1]:
```
import pyreadstat
import pandas as pd
import statsmodels.formula.api as smf

pd.options.display.float_format = '{:.4f}'.format

df_hsb, metadata_hsb = pyreadstat.read_sav("data/hsb12.sav")

df_hsb.describe()
```

[1]:
```
         SCHOOL   STUDENT      CONS  MINORITY    FEMALE       SES   MEANSES  \
count 7185.0000 7185.0000 7185.0000 7185.0000 7185.0000 7185.0000 7185.0000
mean  5277.8978   24.5081    1.0000    0.2747    0.5282    0.0001    0.0061
std   2499.5778   15.2024    0.0000    0.4464    0.4992    0.7794    0.4136
min   1224.0000    1.0000    1.0000    0.0000    0.0000   -3.7580   -1.1880
25%   3020.0000   12.0000    1.0000    0.0000    0.0000   -0.5380   -0.3170
```

```
50%    5192.0000   23.0000     1.0000     0.0000     1.0000     0.0020     0.0380
75%    7342.0000   36.0000     1.0000     1.0000     1.0000     0.6020     0.3330
max    9586.0000   67.0000     1.0000     1.0000     1.0000     2.6920     0.8310

           CSES   MATHACH      SIZE    SECTOR    PRACAD   DISCLIM   HIMINTY
count 7185.0000 7185.0000 7185.0000 7185.0000 7185.0000 7185.0000 7185.0000
mean    -0.0060   12.7479 1056.8618    0.4931    0.5345   -0.1319    0.2800
std      0.6606    6.8782  604.1725    0.5000    0.2512    0.9440    0.4490
min     -3.6570   -2.8320  100.0000    0.0000    0.0000   -2.4160    0.0000
25%     -0.4540    7.2750  565.0000    0.0000    0.3200   -0.8170    0.0000
50%      0.0100   13.1310 1016.0000    0.0000    0.5300   -0.2310    0.0000
75%      0.4630   18.3170 1436.0000    1.0000    0.7000    0.4600    1.0000
max      2.8500   24.9930 2713.0000    1.0000    1.0000    2.7560    1.0000
```

There are no missing values in the dataset. The dataset has two additional columns not described in the task: `CONS` and `DISCLIM`. We are not going to remove them to avoid using them accidentally.

Additionally, we'll cast `SCHOOL` and `STUDENT` into integers and make the column names lowercase for convenience.

[2]:
```python
df_hsb = df_hsb.drop(["CONS", "DISCLIM"], axis=1)
df_hsb.columns = df_hsb.columns.str.lower()

df_hsb["school"] = df_hsb["school"].astype(int)
df_hsb["student"] = df_hsb["student"].astype(int)

df_hsb.head()
```

[2]:
```
   school  student  minority  female     ses  meanses    cses  mathach  \
0    1224        1    0.0000  1.0000 -1.5280  -0.4280 -1.1000   5.8760
1    1224        2    0.0000  1.0000 -0.5880  -0.4280 -0.1600  19.7080
2    1224        3    0.0000  0.0000 -0.5280  -0.4280 -0.1000  20.3490
3    1224        4    0.0000  0.0000 -0.6680  -0.4280 -0.2400   8.7810
4    1224        5    0.0000  0.0000 -0.1580  -0.4280  0.2700  17.8980

      size  sector  pracad  himinty
0 842.0000  0.0000  0.3500   0.0000
1 842.0000  0.0000  0.3500   0.0000
2 842.0000  0.0000  0.3500   0.0000
3 842.0000  0.0000  0.3500   0.0000
4 842.0000  0.0000  0.3500   0.0000
```

1.1 Unconditional model

1. Create an unconditional model and calculate ICC.

We will start by creating an unconditional model. The model will have two levels: students and schools.

Model:	MixedLM	Dependent Variable:	mathach
No. Observations:	7185	Method:	REML
No. Groups:	160	Scale:	39.1483
Min. group size:	14	Log-Likelihood:	-23558.3967
Max. group size:	67	Converged:	Yes
Mean group size:	44.9		

	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
Intercept	12.637	0.244	51.704	0.000	12.158	13.116
Group Var	8.615	0.174				

[13]:
```python
unconditional_model = smf.mixedlm(
    "mathach ~ 1",
    data=df_hsb,
    groups=df_hsb["school"],
    re_formula="~ 1"
)
unconditional_model_results = unconditional_model.fit()

unconditional_model_results.summary()
```

[13]:

[15]:
```python
group_variance = unconditional_model_results.cov_re.iloc[0, 0]
res_variance = unconditional_model_results.scale
print(f"Group variance: {group_variance:.4f}")
print(f"Residual variance: {res_variance:.4f}")

icc = group_variance / (group_variance + res_variance)
print(f"ICC: {icc:.4f}")
```

```
Group variance: 8.6148
Residual variance: 39.1483
ICC: 0.1804
```

The Intraclass Correlation Coefficient (ICC) is `0.1804`, which means that 18.04% of the variance in math achievement can be attributed to differences between schools.

1.2 Final model

2. Create a final model with at least three variables (at least one school-level variable).

1.2.1 Correlation analysis

[16]:
```python
cor = df_hsb.corr()
cor[cor > 0.5].stack().rename("corr").reset_index().query("level_0 < level_1")
```

[16]:

	level_0	level_1	corr
8	meanses	ses	0.5306
10	meanses	pracad	0.6373

```
11     cses       ses 0.8476
18   pracad    sector 0.6811
20  himinty  minority 0.5814
```

[17]:
```
cor[cor.index == "mathach"].stack().rename("corr").reset_index().
 ↪sort_values("corr", ascending=False, key=abs)
```

[17]:
```
    level_0   level_1    corr
7   mathach   mathach  1.0000
4   mathach       ses  0.3608
5   mathach   meanses  0.3437
10  mathach    pracad  0.2921
2   mathach  minority -0.2680
6   mathach      cses  0.2104
9   mathach    sector  0.2040
11  mathach   himinty -0.1731
3   mathach    female -0.1231
8   mathach      size -0.0506
1   mathach   student  0.0194
0   mathach    school -0.0029
```

We're going to create a final model with the significant variables from the correlation analysis.

[18]:
```
final_model = smf.mixedlm(
    "mathach ~ ses + meanses + pracad + minority + female",
    df_hsb,
    groups=df_hsb["school"],
    re_formula="~ minority",
)
final_model_results = final_model.fit()

final_model_results.summary()
```

```
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-packages/statsmodels/base/model.py:607:
ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check
mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-
packages/statsmodels/regression/mixed_linear_model.py:2200: ConvergenceWarning:
Retrying MixedLM optimization with lbfgs
  warnings.warn(
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-packages/statsmodels/base/model.py:607:
ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check
mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

Model:	MixedLM	Dependent Variable:	mathach
No. Observations:	7185	Method:	REML
No. Groups:	160	Scale:	34.7450
Min. group size:	14	Log-Likelihood:	-23324.0790
Max. group size:	67	Converged:	No
Mean group size:	44.9		

	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
Intercept	13.051	1.636	7.975	0.000	9.843	16.258
ses	1.872	0.108	17.335	0.000	1.660	2.084
meanses	1.879	1.947	0.965	0.335	-1.938	5.695
pracad	1.851	2.948	0.628	0.530	-3.926	7.628
minority	-2.947	0.317	-9.304	0.000	-3.568	-2.326
female	-1.180	0.165	-7.132	0.000	-1.504	-0.855
Group Var	59.900					
Group x minority Cov	8.682					
minority Var	6.181	0.177				

```
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-
packages/statsmodels/regression/mixed_linear_model.py:2200: ConvergenceWarning:
Retrying MixedLM optimization with cg
  warnings.warn(
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-packages/statsmodels/base/model.py:607:
ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check
mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-
packages/statsmodels/regression/mixed_linear_model.py:2206: ConvergenceWarning:
MixedLM optimization failed, trying a different optimizer may help.
  warnings.warn(msg, ConvergenceWarning)
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-
packages/statsmodels/regression/mixed_linear_model.py:2218: ConvergenceWarning:
Gradient optimization failed, |grad| = 114.844762
  warnings.warn(msg, ConvergenceWarning)
/home/aleks/.cache/pypoetry/virtualenvs/multivariate-
bR2SZf0l-py3.11/lib/python3.11/site-
packages/statsmodels/regression/mixed_linear_model.py:2261: ConvergenceWarning:
The Hessian matrix at the estimated parameter values is not positive definite.
  warnings.warn(msg, ConvergenceWarning)
```

[18]:

There are some fitting issues I had no time to fix.

1.2.2 Equations for both levels

[32]:
```
# equations for both levels

# level 1

# level 2
```

1.2.3 Combined model

[ ]:
```
# TODO
```

1.2.4 List of fixed and random effect variables

As seen in "Final model" section, the final model has the following fixed effects:

- `ses`
- `meanses`
- `pracad`
- `minority`
- `female`

And the following random effects:

- `minority`

1.2.5 Estimates for fixed parameters

The estimates for fixed parameters are given below:

[19]:
```
coef = final_model_results.fe_params.rename("coef")
coef.to_frame()
```

[19]:

	coef
Intercept	13.0508
ses	1.8721
meanses	1.8785
pracad	1.8513
minority	-2.9472
female	-1.1796

1.2.6 Combined model with parameter estimates

[22]:
```
random_effects = pd.DataFrame(final_model_results.random_effects).T

print(" + ".join([
    f"mathach_ij = {coef.Intercept:.3f}",
    f"{coef.ses:.3f}*ses_ij",
    f"{coef.meanses:.3f}*meanses_ij",
```

```
    f"{coef.pracad:.3f}*pracad_ij",
    f"{coef.minority:.3f}*minority_ij",
    f"{coef.female:.3f}*female_ij",
    "u_0j + r_ij"
]))
```

```
mathach_ij = 13.051 + 1.872*ses_ij + 1.879*meanses_ij + 1.851*pracad_ij +
-2.947*minority_ij + -1.180*female_ij + u_0j + r_ij
```

Here u_0j is the school-level random effect for minority and r_ij is the student-level random effect.

1.2.7 Change in chosen information index

We will calculate the change in Akaike criteria (AIC) for the final model. The AIC is given by the formula:

$$AIC = -2 \times \text{log-likelihood} + 2 \times \text{number of random-effect parameters}$$

[23]:
```
unconditional_aic = unconditional_model_results.aic
final_aic = final_model_results.aic
aic_change = final_aic - unconditional_aic

print(f"Unconditional AIC: {unconditional_aic:.4f}")
print(f"Final AIC: {final_aic:.4f}")
print(f"Change in AIC: {aic_change:.4f}")
```

```
Unconditional AIC: nan
Final AIC: nan
Change in AIC: nan
```

Strangely, AIC was not automatically computed.

1.2.8 Relative change in first level residual variance estimate

Change in the first level residual variance estimate is given by the formula:

$$\frac{\text{Old residual variance estimate} - \text{New residual variance estimate}}{\text{Old residual variance estimate}}$$

[95]:
```
unconditional_residual_var = unconditional_model_results.scale
final_residual_var = final_model_results.scale
variance_change = (unconditional_residual_var - final_residual_var) /
 ↪unconditional_residual_var
print(f"Relative Change in Residual Variance: {variance_change:.4f}")
```

```
Relative Change in Residual Variance: 0.1125
```

The relative change in the first level residual variance estimate is `0.1125`. This means that the final model explains 11.25% more of the variance in math achievement at the student level.

1.3 Forecasting

We will forecast `mathach` for a student with the following characteristics:

[29]:
```
forecast_data = {
    "minority": 1,
    "female": 1,
    "ses": 0,
    "cses": 0.4,
    "meanses": -0.4,
    "size": 800,
    "sector": 0,
    "pracad": 0.25,
    "himinty": 0,
}
```

[31]:
```
fixed_effects = coef.to_dict()

forecast_value = fixed_effects["Intercept"] + sum(
    [
        fixed_effects[var] * forecast_data[var]
        for var in forecast_data
        if var in fixed_effects
    ]
)

print(f"Forecasted mathach: {forecast_value:.4f}")
```

```
Forecasted mathach: 8.6354
```