Covariance Pattern Analysis

Alexander McLain

Unstructured analysis

First, let's load in the lead data and turn it from wide to long.

```
library(tidyr)
wide_lead <- read.csv("wide_lead.csv",header = TRUE, na.strings = "",
                      stringsAsFactors = FALSE)
long_lead <- pivot_longer(wide_lead, cols = starts_with("PB"), names_to = "time",
                          names_prefix = "PB", values_to = "PB",
                          values_drop_na = TRUE)
head(long_lead,10)
```

ID	TRT	time	PB
1	P	1	30.8
1	P	2	26.9
1	P	3	25.8
1	P	4	23.8
2	A	1	26.5
2	A	2	14.8
2	A	3	19.5
2	A	4	21.0
3	A	1	25.8
3	A	2	23.0

To implement the a covariance pattern analysis with an unstructed covariance were going to use **gls** function in the **nlme** package.

```
library(nlme)
```

We need to discuss some of the basic quantities for how this function works. The first thing we need to specify a formula. The formula will specify how we are modeling the mean or average. We'll discuss this as we continue through the semester.

A formula will look like the following (these are just examples):

```
formula1 <- Y ~ X1 + X2
formula2 <- PB ~ TRT
```

The second thing we need is the set the correlation matrix. We can set the correlation matrix to any of the following

```
?corClasses
```

```
## Correlation Structure Classes
##
## Value:
```

```
##
##      Available standard classes:
##
##   corAR1: autoregressive process of order 1.
##
##  corARMA: autoregressive moving average process, with arbitrary orders
##           for the autoregressive and moving average components.
##
##  corCAR1: continuous autoregressive process (AR(1) process for a
##           continuous time covariate).
##
## corCompSymm: compound symmetry structure corresponding to a constant
##           correlation.
##
##   corExp: exponential spatial correlation.
##
##  corGaus: Gaussian spatial correlation.
##
##   corLin: linear spatial correlation.
##
## corRatio: Rational quadratics spatial correlation.
##
## corSpher: spherical spatial correlation.
##
##  corSymm: general correlation matrix, with no additional structure.
```

To use the correlation classes you have to give it the ID in the following fasion:

```
cor_fun <- corSymm(value = ~ 1|ID)
```

This tells the function that the 'ID' variable is what were modeling the correlation within.

The second thing we need is the set the is if the residual variance is heterogeneous or homogeneous. If you want a homogeneous variance you don't need to do anything (that is the default). If we want a heterogenous residual matrix you want to use:

```
var_fun <- varIdent(value = ~1|time)
```

This tells the function that we want a different variance for each time point. Once we specify the formula, correlation and variance we can run our analysis in the **gls** function.

```
?gls
```

```
## Fit Linear Model Using Generalized Least Squares
##
## Usage:
##
##      gls(model, data, correlation, weights, subset, method, na.action,
##          control, verbose)
##      ## S3 method for class 'gls'
##      update(object, model., ..., evaluate = TRUE)
```

The unstructured covariance model as follows:

```
# Set the formula, correlation and variance
formula2<- PB ~ TRT + time + TRT*time
cor_fun <- corSymm(form = ~ 1|ID)
var_fun <- varIdent(form = ~ 1|time)
```

```r
# Run the model
lm_unstruc <- gls(model = formula2, data = long_lead, correlation = cor_fun,
                  weights = var_fun)
summary(lm_unstruc)
```

```
## Generalized least squares fit by REML
##   Model: formula2
##   Data: long_lead
##        AIC      BIC    logLik
##   2452.076 2523.559 -1208.038
##
## Correlation Structure: General
##  Formula: ~1 | ID
##  Parameter estimate(s):
##  Correlation:
##   1     2     3
## 2 0.571
## 3 0.570 0.775
## 4 0.577 0.582 0.581
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | time
##  Parameter estimates:
##        1        2        3        4
## 1.000000 1.325887 1.370454 1.524826
##
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept)  26.540 0.7102888  37.36508  0.0000
## TRTP         -0.268 1.0045001  -0.26680  0.7898
## time2       -13.018 0.7919194 -16.43854  0.0000
## time3       -11.026 0.8149169 -13.53021  0.0000
## time4        -5.778 0.8885251  -6.50291  0.0000
## TRTP:time2   11.406 1.1199432  10.18445  0.0000
## TRTP:time3    8.824 1.1524665   7.65662  0.0000
## TRTP:time4    3.152 1.2565643   2.50843  0.0125
##
##  Correlation:
##            (Intr) TRTP   time2  time3  time4  TRTP:2 TRTP:3
## TRTP       -0.707
## time2      -0.218  0.154
## time3      -0.191  0.135  0.680
## time4      -0.096  0.068  0.386  0.385
## TRTP:time2  0.154 -0.218 -0.707 -0.481 -0.273
## TRTP:time3  0.135 -0.191 -0.481 -0.707 -0.272  0.680
## TRTP:time4  0.068 -0.096 -0.273 -0.272 -0.707  0.386  0.385
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.1756392 -0.6849960 -0.1515546  0.5294172  5.6327405
##
## Residual standard error: 5.0225
## Degrees of freedom: 400 total; 392 residual
```

```r
getVarCov(lm_unstruc)
```

```
## Marginal variance covariance matrix
##        [,1]   [,2]   [,3]   [,4]
## [1,] 25.226 19.107 19.699 22.202
## [2,] 19.107 44.346 35.535 29.675
## [3,] 19.699 35.535 47.377 30.620
## [4,] 22.202 29.675 30.620 58.652
##   Standard Deviations: 5.0225 6.6593 6.8831 7.6584
```

```r
anova(lm_unstruc)
```

	numDF	F-value	p-value
(Intercept)	1	2583.784243	0.0000000
TRT	1	4.226606	0.0404567
time	3	61.493511	0.0000000
TRT:time	3	35.928996	0.0000000

The independent covariance model as follows:

```r
# Set the formula, correlation and variance
cor_fun <- corIdent(form = ~ 1|ID)
# Run the model
lm_indep <- gls(model = formula2, data = long_lead, correlation = cor_fun)
summary(lm_indep)
```

```
## Generalized least squares fit by REML
##   Model: formula2
##   Data: long_lead
##        AIC      BIC    logLik
##   2644.255 2679.997 -1313.128
##
## Correlation Structure: Independent
##  Formula: ~1 | ID
##  Parameter estimate(s):
## numeric(0)
##
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept)  26.540 0.9370175 28.323912  0.0000
## TRTP         -0.268 1.3251428 -0.202242  0.8398
## time2       -13.018 1.3251428 -9.823847  0.0000
## time3       -11.026 1.3251428 -8.320613  0.0000
## time4        -5.778 1.3251428 -4.360285  0.0000
## TRTP:time2   11.406 1.8740349  6.086333  0.0000
## TRTP:time3    8.824 1.8740349  4.708557  0.0000
## TRTP:time4    3.152 1.8740349  1.681932  0.0934
##
##  Correlation:
##            (Intr) TRTP   time2  time3  time4  TRTP:2 TRTP:3
## TRTP       -0.707
## time2      -0.707  0.500
## time3      -0.707  0.500  0.500
## time4      -0.707  0.500  0.500  0.500
```

```
## TRTP:time2  0.500 -0.707 -0.707 -0.354 -0.354
## TRTP:time3  0.500 -0.707 -0.354 -0.707 -0.354  0.500
## TRTP:time4  0.500 -0.707 -0.354 -0.354 -0.707  0.500  0.500
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.5147478 -0.6973588 -0.1498706  0.5542799  6.5106945
##
## Residual standard error: 6.625714
## Degrees of freedom: 400 total; 392 residual
```

```r
anova(lm_indep)
```

	numDF	F-value	p-value
(Intercept)	1	4359.34511	0
TRT	1	70.86206	0
time	3	24.85017	0
TRT:time	3	15.41672	0

The compound symmetric covariance model as follows:

```r
# Set the formula, correlation and variance
cor_fun <- corCompSymm(form = ~ 1|ID)
# Run the model
lm_CS <- gls(model = formula2, data = long_lead, correlation = cor_fun)
summary(lm_CS)
```

```
## Generalized least squares fit by REML
##   Model: formula2
##   Data: long_lead
##        AIC      BIC    logLik
##   2480.621 2520.334 -1230.311
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | ID
##  Parameter estimate(s):
##       Rho
## 0.5954401
##
## Coefficients:
##                Value Std.Error    t-value p-value
## (Intercept)   26.540 0.9370175  28.323911  0.0000
## TRTP          -0.268 1.3251428  -0.202242  0.8398
## time2        -13.018 0.8428574 -15.445080  0.0000
## time3        -11.026 0.8428574 -13.081691  0.0000
## time4         -5.778 0.8428574  -6.855252  0.0000
## TRTP:time2    11.406 1.1919804   9.568950  0.0000
## TRTP:time3     8.824 1.1919804   7.402807  0.0000
## TRTP:time4     3.152 1.1919804   2.644339  0.0085
##
##  Correlation:
##            (Intr) TRTP   time2  time3  time4  TRTP:2 TRTP:3
## TRTP       -0.707
## time2      -0.450  0.318
```

```
## time3      -0.450  0.318  0.500
## time4      -0.450  0.318  0.500  0.500
## TRTP:time2  0.318 -0.450 -0.707 -0.354 -0.354
## TRTP:time3  0.318 -0.450 -0.354 -0.707 -0.354  0.500
## TRTP:time4  0.318 -0.450 -0.354 -0.354 -0.707  0.500  0.500
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.5147478 -0.6973588 -0.1498706  0.5542799  6.5106944
##
## Residual standard error: 6.625714
## Degrees of freedom: 400 total; 392 residual
```

```r
getVarCov(lm_CS)
```

```
## Marginal variance covariance matrix
##       [,1]  [,2]  [,3]  [,4]
## [1,] 43.90 26.14 26.14 26.14
## [2,] 26.14 43.90 26.14 26.14
## [3,] 26.14 26.14 43.90 26.14
## [4,] 26.14 26.14 26.14 43.90
##   Standard Deviations: 6.6257 6.6257 6.6257 6.6257
```

```r
anova(lm_CS)
```

	numDF	F-value	p-value
(Intercept)	1	1564.55268	0e+00
TRT	1	25.43213	7e-07
time	3	61.42519	0e+00
TRT:time	3	38.10738	0e+00

The heterogeneous compound symmetric covariance model as follows:

```r
# Set the formula, correlation and variance
cor_fun <- corCompSymm(form = ~ 1|ID)
var_fun <- varIdent(form = ~ 1|time)
# Run the model
lm_HCS <- gls(model = formula2, data = long_lead, correlation = cor_fun,
                   weights = var_fun)
summary(lm_HCS)
```

```
## Generalized least squares fit by REML
##   Model: formula2
##   Data: long_lead
##       AIC      BIC   logLik
##   2459.96 2511.587 -1216.98
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | ID
##  Parameter estimate(s):
##       Rho
## 0.6102796
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | time
```

```
##  Parameter estimates:
##        1        2        3        4 
## 1.000000 1.279651 1.323192 1.519195 
## 
## Coefficients:
##               Value Std.Error   t-value p-value
## (Intercept)  26.540 0.7238070  36.66723  0.0000
## TRTP         -0.268 1.0236177  -0.26182  0.7936
## time2       -13.018 0.7506742 -17.34174  0.0000
## time3       -11.026 0.7713906 -14.29367  0.0000
## time4        -5.778 0.8726861  -6.62094  0.0000
## TRTP:time2   11.406 1.0616137  10.74402  0.0000
## TRTP:time3    8.824 1.0909111   8.08865  0.0000
## TRTP:time4    3.152 1.2341645   2.55395  0.0110
## 
##  Correlation: 
##            (Intr) TRTP   time2  time3  time4  TRTP:2 TRTP:3
## TRTP       -0.707                                          
## time2      -0.211  0.149                                   
## time3      -0.181  0.128  0.402                            
## time4      -0.060  0.043  0.383  0.383                     
## TRTP:time2  0.149 -0.211 -0.707 -0.285 -0.270              
## TRTP:time3  0.128 -0.181 -0.285 -0.707 -0.271  0.402       
## TRTP:time4  0.043 -0.060 -0.270 -0.271 -0.707  0.383  0.383
## 
## Standardized residuals:
##        Min         Q1        Med         Q3        Max 
## -2.1429194 -0.6927682 -0.1528875  0.5263104  5.5480289 
## 
## Residual standard error: 5.118088 
## Degrees of freedom: 400 total; 392 residual
```

```r
getVarCov(lm_HCS)
```

```
## Marginal variance covariance matrix
##        [,1]   [,2]   [,3]   [,4]
## [1,] 26.195 20.457 21.153 24.286
## [2,] 20.457 42.894 27.068 31.078
## [3,] 21.153 27.068 45.863 32.135
## [4,] 24.286 31.078 32.135 60.456
##   Standard Deviations: 5.1181 6.5494 6.7722 7.7754
```

```r
anova(lm_HCS)
```

	numDF	F-value	p-value
(Intercept)	1	2438.048614	0.0000000
TRT	1	7.116307	0.0079553
time	3	80.549368	0.0000000
TRT:time	3	46.929235	0.0000000

Let's check the AIC of all of these models

```r
AIC(lm_unstruc)
```

```
## [1] 2452.076
```

```r
AIC(lm_indep)
```

```
## [1] 2644.255
```

```r
AIC(lm_CS)
```

```
## [1] 2480.621
```

```r
AIC(lm_HCS)
```

```
## [1] 2459.96
```

Here, we're going to use an unstructured covariance matrix, but allow the variance to be different by group. Note that the residual variance differs by group, but the correlation does not. There is actualy no way to have different correlation matrices by TRT group when using **gls**. This is something that can be done in **SAS**.

```r
# Set the formula, correlation and variance
formula2<- PB ~ TRT + time + TRT*time
cor_fun <- corSymm(form = ~ 1|ID)
var_fun <- varIdent(form = ~ 1|time*TRT)
# Run the model
lm_unstruc2 <- gls(model = formula2, data = long_lead, correlation = cor_fun,
                   weights = var_fun)
summary(lm_unstruc2)
```

```
## Generalized least squares fit by REML
##   Model: formula2
##   Data: long_lead
##        AIC      BIC    logLik
##   2386.174 2473.541 -1171.087
##
## Correlation Structure: General
##  Formula: ~1 | ID
##  Parameter estimate(s):
##  Correlation:
##   1     2     3
## 2 0.683
## 3 0.689 0.818
## 4 0.667 0.674 0.733
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | time * TRT
##  Parameter estimates:
##      1*P      2*P      3*P      4*P      1*A      2*A      3*A      4*A
## 1.000000 1.088074 1.127801 1.120538 1.391921 2.122833 2.245127 2.572218
##
## Coefficients:
##               Value Std.Error    t-value p-value
## (Intercept)  26.540 0.8609004  30.828188  0.0000
## TRTP         -0.268 1.0600422  -0.252820  0.8005
## time2       -13.018 0.9592411 -13.571145  0.0000
## time3       -11.026 1.0109941 -10.906097  0.0000
## time4        -5.778 1.2015623  -4.808740  0.0000
## TRTP:time2   11.406 1.0893518  10.470447  0.0000
## TRTP:time3    8.824 1.1387400   7.748915  0.0000
## TRTP:time4    3.152 1.3169567   2.393397  0.0172
```

```
##
##  Correlation:
##             (Intr) TRTP   time2  time3  time4  TRTP:2 TRTP:3
## TRTP        -0.812
## time2        0.038 -0.031
## time3        0.095 -0.077  0.655
## time4        0.167 -0.136  0.400  0.513
## TRTP:time2 -0.033 -0.058 -0.881 -0.577 -0.353
## TRTP:time3 -0.084 -0.002 -0.582 -0.888 -0.455  0.661
## TRTP:time4 -0.153  0.055 -0.365 -0.468 -0.912  0.410  0.518
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.0703549 -0.7306997 -0.1597098  0.5598887  4.0105219
##
## Residual standard error: 4.373442
## Degrees of freedom: 400 total; 392 residual
```

```r
anova(lm_unstruc2)
```

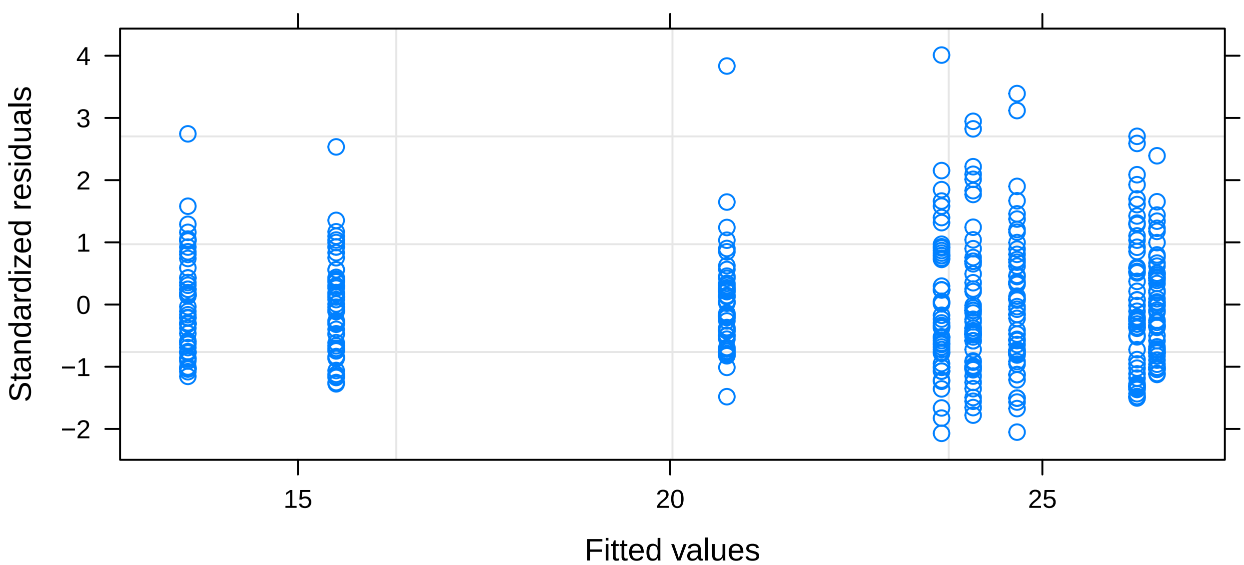
	numDF	F-value	p-value
(Intercept)	1	2911.85502	0.0000000
TRT	1	2.79556	0.0953237
time	3	34.55329	0.0000000
TRT:time	3	39.29392	0.0000000

```r
AIC(lm_unstruc2)
```

```
## [1] 2386.174
```

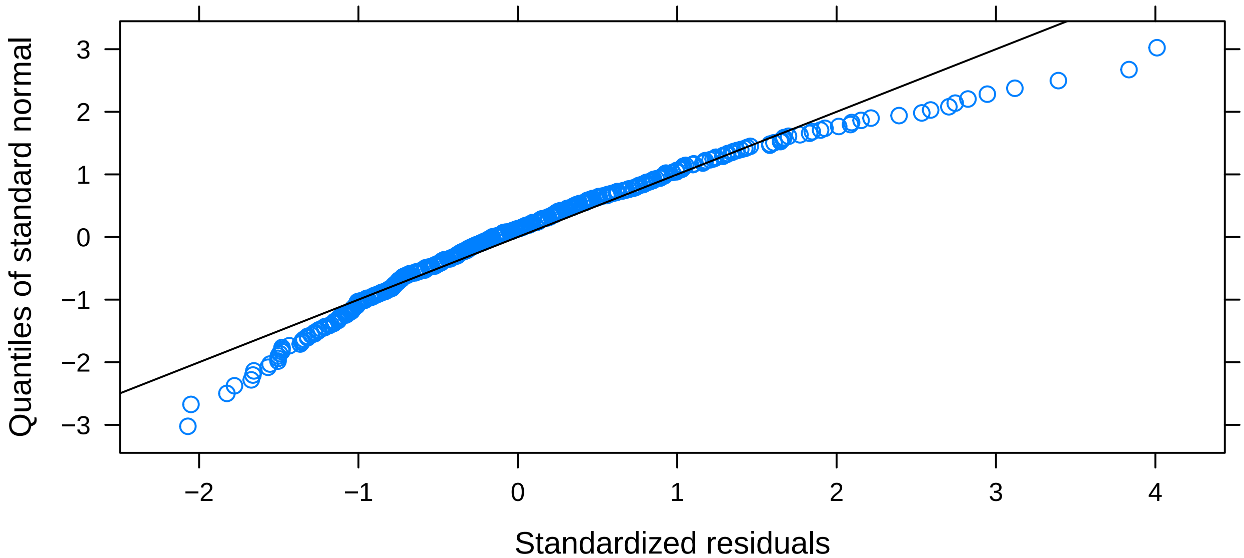
Plot Person residuals vs. predicted values

```r
plot(lm_unstruc2, resid(. , type="p",level=1) ~ fitted(.,level=1) )
```



QQ plot of Person residuals

```r
qqnorm(lm_unstruc2, ~ resid(. , type="p",level=1),abline=c(0,1))
```



A quick example with an exponential correlation matrix with and without a "**nugget**".

First, we have to create a variable that has time as a continuous **numeric** variable.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v dplyr   1.0.2
## v tibble  3.0.4     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::collapse() masks nlme::collapse()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
```

```r
long_lead <- long_lead %>% mutate(time_c = as.numeric(time)) %>%
  mutate(time_c = replace(time_c, time_c == 1, 0)) %>%
  mutate(time_c = replace(time_c, time_c == 2, 1)) %>%
  mutate(time_c = replace(time_c, time_c == 4, 6)) %>%
  mutate(time_c = replace(time_c, time_c == 3, 4))
head(long_lead)
```

ID	TRT	time	PB	time_c
1	P	1	30.8	0
1	P	2	26.9	1
1	P	3	25.8	4
1	P	4	23.8	6
2	A	1	26.5	0
2	A	2	14.8	1

Now, we'll fit the models.

```r
# Set the formula, correlation and variance
formula2<- PB ~ TRT + time + TRT*time
var_fun <- varIdent(form = ~ 1|time*TRT)

cor_fun <- corExp(form = ~ time_c|ID)
# An exponential correlation matrix
lm_exp <- gls(model = formula2, data = long_lead, correlation = cor_fun,
                   weights = var_fun)
getVarCov(lm_exp)
```

```
## Marginal variance covariance matrix
##          [,1]    [,2]   [,3]    [,4]
## [1,] 22.4090 19.2440 10.546  7.5138
## [2,] 19.2440 23.8160 13.052  9.2989
## [3,] 10.5460 13.0520 21.408 15.2520
## [4,]  7.5138  9.2989 15.252 22.5690
##   Standard Deviations: 4.7338 4.8802 4.6268 4.7507
```

```r
AIC(lm_exp)
```

```
## [1] 2441.982
```

```r
cor_fun <- corExp(form = ~ time_c|ID, nugget = TRUE)
# An exponential correlation matrix with a nugget
lm_exp_nug <- gls(model = formula2, data = long_lead, correlation = cor_fun,
                   weights = var_fun)
getVarCov(lm_exp_nug)
```

```
## Marginal variance covariance matrix
##         [,1]   [,2]   [,3]   [,4]
## [1,] 18.957 14.985 15.014 14.895
## [2,] 14.985 22.204 16.400 16.270
## [3,] 15.014 16.400 23.558 17.228
## [4,] 14.895 16.270 17.228 24.055
##   Standard Deviations: 4.3539 4.7121 4.8536 4.9046
```

```r
AIC(lm_exp_nug)
```

```
## [1] 2391.733
```