MT5757: Advanced Data Analysis
School of Mathematics and Statistics, University of St. Andrews

January 25, 2018

Contents

1 Introducing the data

We are going to use some environmental impact assessment (EIA) data to revise Generalized Linear Models (GLMs) and introduce Generalized Additive Models (GAMs; section 2) and Generalized Estimating Equations (GEEs; section 3).

1.1 Nysted Wind Farms I and II

The EIA data is collected from a site with two off-shore wind farms in Denmark. These wind farms are among the largest in the world (Figure 1) and generate large amounts of renewable energy.

Figure 1: Nysted I wind farm

Here are some details about the data:

- The data are collected from aircraft travelling along transects (pre-determined tracks) across the water. The data collected were counts of birds seen from the aircraft, within some distance of the transects (Figure 2).
- Environmental data (e.g. Depth) is also available from these spatial positions.
- The count data used as input for the models which follow, are already corrected for the animals that were missed due to the imperfect detection process (animals farthest from the plane are more difficult to see).
- This inflation for the numbers overlooked was undertaken using Distance Sampling; these methods are routinely used for this purpose but are outside the scope of this course.
- The data were collected in three phases:
  - A: before any farms were installed in the area
  - B: after one wind farm (Nysted I) was installed

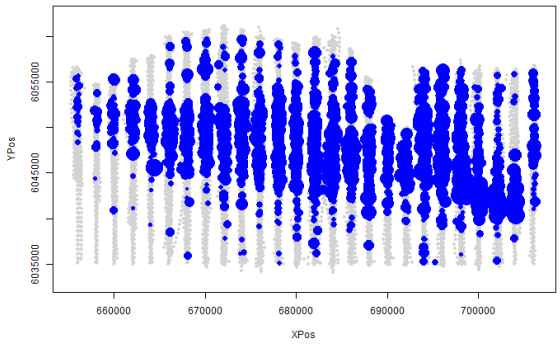


Figure 2: Transect locations (grey lines) and locations with non-zero counts (blue dots). The blue dots are scaled based on the associated count.

  - C: after two wind farms were installed (Nysted I and II)

- There were multiple transects surveyed, within survey days, within the three phases (A, B and C).
- The survey effort is highly uneven across phases, there were:
  - 11 survey days in Phase A,
  - 13 survey days in Phase B and
  - 5 days in Phase C.
- There are different areas associated with each count – the areas range from 0.00157 to 0.956 square kilometres.
- The counts are likely to be correlated in time/space:
  - i.e. counts collected close together within transects are likely to be similar to each other (e.g. 'popular transects' are likely to exhibit high counts for many (consecutive) sites along the transect)

1.2 Research Questions

We are going to ask some questions of the data in this section:

1. Does the abundance appear to have changed across phases?
2. What are the best predictors of abundance and what do these relationships look like?
3. Do the animals appear to have redistributed across phases? If so, what does that redistribution look like?

1.3 Exploratory Data Analysis

We will begin to address these questions using exploratory work.

In this case, the 95% confidence intervals (Figure 3) suggest there may be fewer animals on average in phase C (compared to phases A and B)

This drop in numbers could be due to lower effort in phase C compared with earlier phases (5 days compared with 11 and 13 days; page 5 & Figure 4).

```
> require(gplots)
> par(las=2)
> plotmeans(count/area ~ yearmonth, pch=20, xlab="Year/Month",
ylab="Estimated abundance per unit area", las=2, n.label=FALSE)
> abline(v=c(9.5, 22.5))
> legend(1,12, c("Phase A"), bty="n")
> legend(13,12, c("Phase B"), bty="n")
> legend(22,12, c("Phase C"), bty="n")
```

There don't appear to be any compelling differences in average numbers between phases A and B, since the confidence intervals share values (Figure 3).

```
> require(gplots)
> plotmeans(count ~ phase, pch=20, xlab="Phase",
ylab="Corrected Count")
```

The confidence intervals in Figure 3 should be treated with caution; we suspect the data are correlated along transects (and thus are unlikely to be independent of each other).
We also need to be sure that any differences in counts across phases (and thus stages in wind farm development) are not due to other predictors.

- For instance, if a species avoids deep waters and deep waters are over-sampled in phase C, this would lead to lower average numbers in phase C (compared to the other phases) for this reason alone.
- Ignoring these sampling issues might mean differences in average numbers across phases are incorrectly attributed to wind farm development.

For these and other reasons, it makes sense to consider several covariates simultaneously even if the primary interest lies in differences across phases.
In order to find good predictors of (estimated) abundance in this area and to examine what these relationships might look like, scatterplots and boxplots can also be useful (Figure 5):

```
> par(mfrow=c(3,2))
> plot(depth, count/area, xlab="Depth",
ylab="Estimated abundance per unit area")
> plot(X, count, xlab="X-coordinate",
ylab="Estimated abundance per unit area")
```

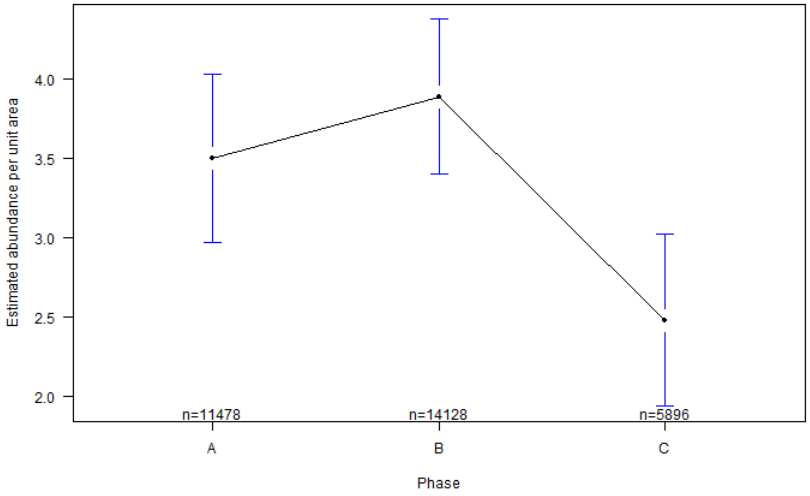


Figure 3: Average counts across phases

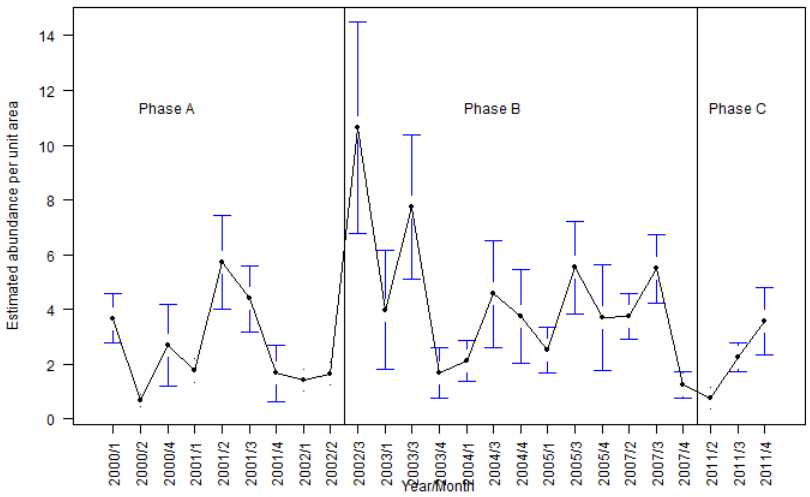


Figure 4: Average counts (with 95% CIs) across years and months (with phases indicated).

```
> plot(Y, count, xlab="Y-coordinate",
ylab="Estimated abundance per unit area")
> plot(phase, count, xlab="Phase",
ylab="Estimated abundance per unit area")
> plot(DistCoast, count, xlab="Distance from Coast",
ylab="Estimated abundance per unit area")
```

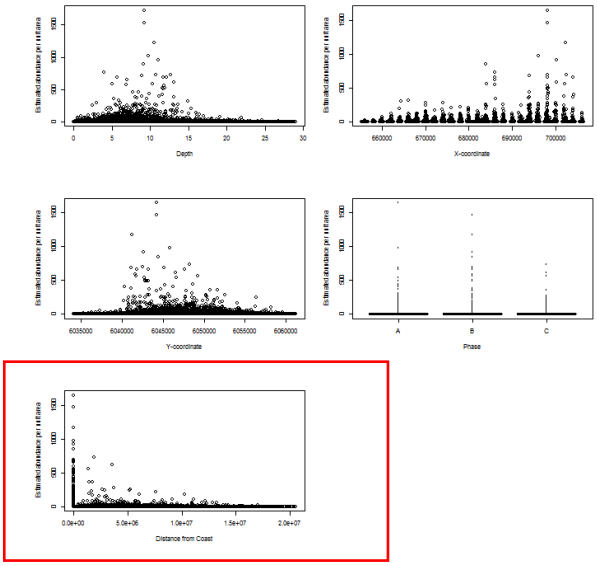


Figure 5: Exploratory Data Analysis for the covariates and estimated numbers.

The highest counts (Figure 5) seem to be associated with:

- moderate depths
- central values of the Y-coordinate
- large values of the X-coordinate
- locations near the coast

These plots tell us there may be some covariate relationships worth pursuing and provide a quick check for outliers in x and/or y.

While we can view the 'depth' and 'distance from coast' relationships easily in Figure 5, these can also be viewed spatially for the surveyed area (Figures 6 and 7).

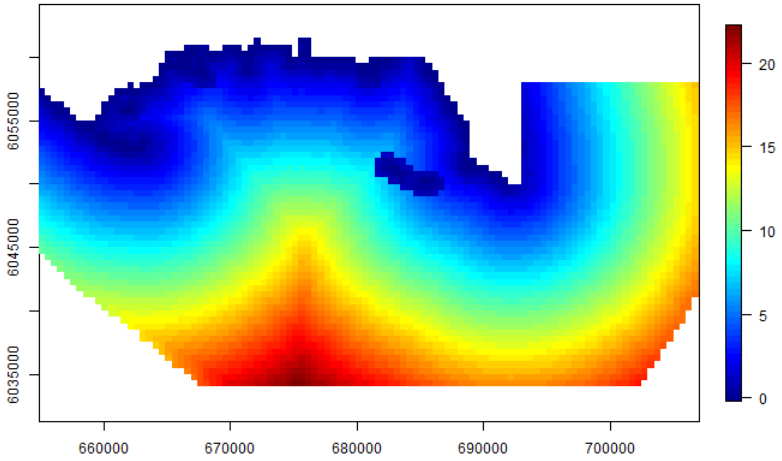


Figure 6: Distance from coast across the surveyed area.

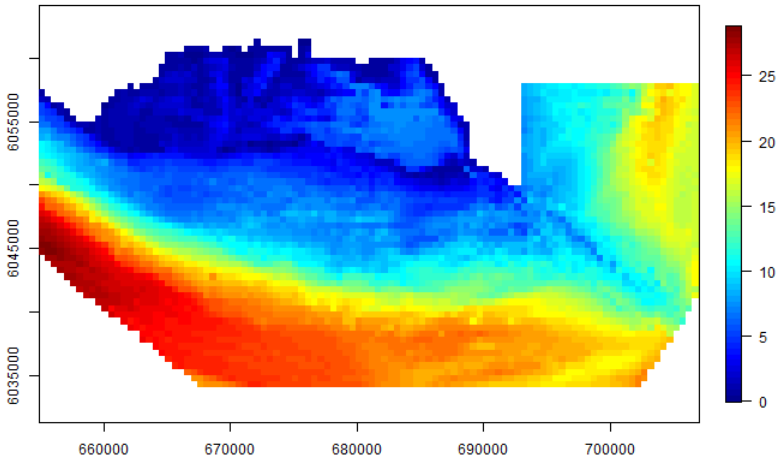


Figure 7: Depth across the surveyed area.

There also seems to be some evidence of redistribution across phases (Figure 8).
Notably the animals appear to be more widely dispersed in phases B and C compared with the more tightly clustered data in phase A.

Figure 8 tells us that we might want to consider models that permit redistribution in the fitted surfaces across phases (e.g. via interaction terms).

```
> par(mfrow=c(1,3))
> plot(X[phase=="A"], Y[phase=="A"], pch=20, col="grey",
main="Phase A", xlab="X co-ordinate", ylab="Y co-ordinate")
> points(X[phase=="A"], Y[phase=="A"], cex=log(count/area),
xlab="X co-ordinate", ylab="Y co-ordinate")
> plot(X[phase=="B"], Y[phase=="B"], pch=20, col="grey",
main="Phase B", xlab="X co-ordinate", ylab="Y co-ordinate")
> points(X[phase=="B"], Y[phase=="B"], cex=log(count/area),
xlab="X co-ordinate", ylab="Y co-ordinate")
> plot(X[phase=="C"], Y[phase=="C"], pch=20, col="grey",
main="Phase C", xlab="X co-ordinate", ylab="Y co-ordinate")
> points(X[phase=="C"], Y[phase=="C"], cex=log(count/area),
xlab="X co-ordinate", ylab="Y co-ordinate")
```

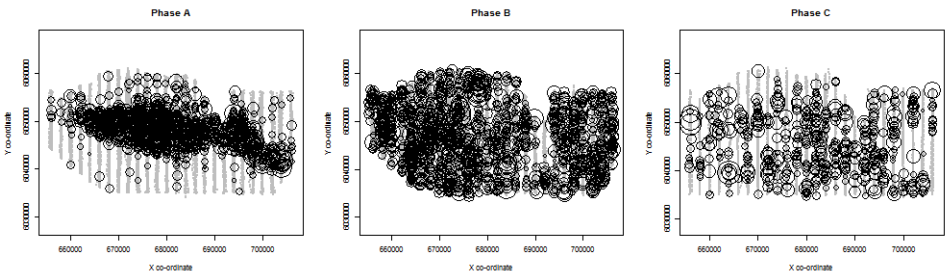


Figure 8: Counts of animals in each phase

1.4 Preliminary Model Specification

While exploratory work is useful at gleaning large-scale patterns from the data and identifying any outlying points, it is often necessary to interrogate the data with specific questions in mind. For instance, we may wish to formally revisit these questions:

1. Are there any differences in average numbers (per unit area) across phases?
2. What are the best predictors of abundance and what do these relationships look like?

3. Do the animals appear to have redistributed across phases? If so, what does this redistribution look like?

In order to answer these questions, we are going to fit some preliminary models:

- Poisson-based GLMs which permit overdispersion (Hilbe, 2014).

We will fit and diagnose each model and make (cautious) interpretations based on the results.

In this example we can think of the data as having $i = 1, ..., s$ panels (transects) and each panel has $t = 1, ...n_i$ potentially correlated observations within transects. N is the total number of observations across transects.

Each panel can have different numbers of observations.

Each model will assume the response Y_{it} belongs to a quasi-Poisson distribution with mean μ_{it} and variance proportional to the mean (via the dispersion parameter ϕ):

$$Y_{it} \sim Poisson(\mu_{it}, \phi\mu_{it}) \tag{1}$$

and the mean will be modelled using a log link and a linear predictor (η_{it}):

$$\mu_{it} = \exp(\eta_{it}) \tag{2}$$

To answer our research questions, the linear predictor will change with the question of interest. For instance:

1 Are there any differences in average numbers (per unit area) across phases?

If we only consider the `phase` term and disregard the other covariates, this model will contain a 3-level factor variable and attract two coefficients in addition to the baseline/intercept term):

$$\eta_{it} = \beta_0 + \beta_1 PhaseB_{it} + \beta_2 PhaseC_{it} + \log(area_{it}) \tag{3}$$

where β_0 is the intercept coefficient and β_1 and β_2 apply when `phase` is equal to B or C respectively (since $PhaseB_{it} = 1$ when phase=B and zero otherwise and $PhaseC_{it} = 1$ when phase=C and zero otherwise).

Area is fitted as an offset term and so does not attract a coefficient.

2 What are the best predictors of abundance?

If we consider several covariates simultaneously, and we are happy to assume these relationships are linear given the log link function[1], this question could

[1] we will relax this later

be addressed using a model with terms for `Depth`, `XPos`, `YPos`, `DistCoast` and `phase`:

$$\begin{aligned}\eta_{it} =&\beta_0 + \beta_1 Depth_{it} + \beta_2 XPos_{it} + \beta_3 YPos_{it} \\ &+\beta_4 DistCoast_{it} + \beta_5 PhaseB_{it} \\ &+ \beta_6 PhaseC_{it} + \log(area_{it})\end{aligned} \tag{4}$$

where β_0 is the intercept coefficient, β_1,...,β_4 are slope coefficients for `Depth`, `XPos, YPos` and `DistCoast` respectively. β_5 and β_6 apply when `phase` is equal to B or C respectively (phase A is part of the baseline). Area still appears as an offset.

3 Do the animals appear to have redistributed across phases? If so, what does this redistribution look like?

We can allow for a particular type of re-distribution across phases by permitting the `XPos` and `YPos` relationships to vary with phase[2].

This can be implemented using an interaction term between each of the spatial co-ordinates and phase. For example:

$$\begin{aligned}\eta_{it} =&\beta_0 + \beta_1 Depth_{it} + \beta_2 XPos_{it} + \beta_3 YPos_{it} \\ &+\beta_4 DistCoast_{it} + \beta_5 PhaseB_{it} + \beta_6 PhaseC_{it} \\ &+\beta_7 XPosPhaseB_{it} + \beta_8 XPosPhaseC_{it} + \beta_9 YPosPhaseB_{it} \\ &+\beta_{10} YPosPhaseC_{it} + log(area_{it})\end{aligned} \tag{5}$$

where β_7 and β_8 are associated with the `XPos` relationship in phase B and C respectively while β_9 and β_{10} are associated with the `YPos` relationship in phase B and C respectively; the other terms are as previously described.

1.4.1 Model Fitting

At this stage (for the purposes of comparison for later) we are going to assume the data are independent[3] and undertake parameter estimation as if the data come from a Poisson distribution.

For this we can use the 'pooled estimator' for Poisson data the log-likelihood function is:

$$\log(L) = \sum_{i=1}^{s} \sum_{t=1}^{n_i} [y_{it}(\eta_{it}) - \exp(\eta_{it}) - \log \Gamma(y_{it} + 1)] \tag{6}$$

- y_{it} are observed counts for the i-th transect at time point t

[2] we will do this more interestingly later in the course
[3] we will relax this later

- η_{it} is the linear predictor for the i-th transect at time point t under the model
- there are n_i observations per transect and s transects

Maximum likelihood is a powerful tool if the data come from a specific (and known) distribution. However, if the data are more variable than what's expected under a Poisson distribution (as is the case here), Quasi-likelihood (QL) estimation is preferable.

QL only assumes the variance of the data is a known function of the mean and that the observations (given the model) are independent.

QL lets us assume that the variance is some function of the mean (via ϕ) rather than related to the mean itself. We can estimate this dispersion parameter ϕ, using:

$$\hat{\phi} = \frac{1}{N-p-1} \sum_{i=1}^{s} \sum_{t=1}^{n_i} \frac{(y_{it} - \hat{\mu}_{it})^2}{V(\hat{\mu}_{it})} \tag{7}$$

1.4.2 Results

Are there any differences across phases?

We can fit the model in Equation 3 using:

```
> glmFit1<- glm(count ~ phase, offset=log(area),
family=poisson,  data=data)

> sum(data$count*(predict(glmFit1, type="link"))-
fitted(glmFit1)-lgamma(data$count+1))
[1] -338407.9

> logLik(glmFit1)
'log Lik.' -338407.9 (df=3)
```

The estimate of the dispersion parameter in this case is:

```
> (1/(nrow(data)-2-1))*sum(((data$count-fitted(glmFit1))^2)/
fitted(glmFit1))
[1] 211.4965
```

We can fit QL models easily in R:

```
> glmFitOD1<- glm(count ~ phase, family=quasipoisson,
offset=log(area), data=data)

> summary(glmFitOD1)$dispersion
[1] 211.4993
```

```
> Anova(glmFitOD1, test="F")
Analysis of Deviance Table (Type II tests)

Response: count
              SS    Df      F   Pr(>F)
phase       2405     2 5.6864 0.003395 **
Residuals 6661928 31499
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

What are the best predictors of abundance?

We are going to include the other candidate predictors in the model and examine if the covariates are significantly related to the response.

We can fit the model in Equation 4 using:

```
> glmFitOD2<- glm(count ~ depth+XPos+YPos+phase+DistCoast,
family=quasipoisson, offset=log(area), data=data)

> Anova(glmFitOD2, test="F")
Analysis of Deviance Table (Type II tests)

Response: count
              SS    Df        F    Pr(>F)
depth       89695     1 941.4310 < 2.2e-16 ***
XPos        27557     1 289.2334 < 2.2e-16 ***
YPos        46171     1 484.6105 < 2.2e-16 ***
phase         512     2   2.6892   0.06795 .
DistCoast    3167     1  33.2445 8.203e-09 ***
Residuals 3000679 31495
```

Do the animals appear to have redistributed across phases?

We can investigate this by fitting an interaction term between each spatial co-ordinate and 'phase':

We can fit the model in Equation 5 using:

```
> glmFitOD3<- glm(count ~ depth+XPos*phase+
YPos*phase+DistCoast, family=quasipoisson,
offset=log(area), data=data)

> Anova(glmFitOD3, test="F")
Analysis of Deviance Table (Type II tests)

Response: count
               SS    Df        F    Pr(>F)
depth        83672     1 851.6569 < 2.2e-16 ***
XPos         27131     1 276.1502 < 2.2e-16 ***
phase          512     2   2.6079   0.07371 .
YPos         46748     1 475.8223 < 2.2e-16 ***
```

```
DistCoast     3036     1  30.8973 2.743e-08 ***
XPos:phase    2092     2  10.6488 2.381e-05 ***
phase:YPos     243     2   1.2382   0.28993
Residuals  3093868 31491
```

From these preliminary results we can see:

- when phase is considered alone in a model, there appears to be genuine differences across phases
- differences across phases are no longer significant when other covariates are considered
- the relationship between average numbers and `Xpos` appears to change with phase, but this is not the case for the Y-coordinates.

We also notice it is important to account for extra-Poisson variation:

In this case, the dispersion parameter is substantially bigger than one ($\hat{\phi} = 98.24$) and this has consequences for model p-values.

In particular, while all terms are significant at the 5% level when $\phi = 1$, both the phase and `phase:Ypos` interaction terms are not ($p = 0.28993$) when ϕ is estimated;

```
> glmFit1<- glm(count ~ phase, family=poisson,
offset=log(area), data=data)

> glmFit2<- glm(count ~ depth+XPos+YPos+phase+DistCoast,
family=poisson,  offset=log(area), data=data)

> glmFit3<- glm(count ~ depth+XPos*phase+YPos*phase+DistCoast,
family=poisson, offset=log(area), data=data)

> require(car)
> Anova(glmFit1)
Analysis of Deviance Table (Type II tests)

Response: count
      LR Chisq Df Pr(>Chisq)
phase   2405.3  2  < 2.2e-16 ***

> Anova(glmFit2)
Analysis of Deviance Table (Type II tests)

Response: count
          LR Chisq Df Pr(>Chisq)
depth        89695  1  < 2.2e-16 ***
XPos         27557  1  < 2.2e-16 ***
YPos         46171  1  < 2.2e-16 ***
phase          512  2  < 2.2e-16 ***
DistCoast     3167  1  < 2.2e-16 ***


> Anova(glmFit3)
```

```
Analysis of Deviance Table (Type II tests)

Response: count
           LR Chisq Df Pr(>Chisq)
depth         83672  1  < 2.2e-16 ***
XPos          27131  1  < 2.2e-16 ***
phase           512  2  < 2.2e-16 ***
YPos          46748  1  < 2.2e-16 ***
DistCoast      3036  1  < 2.2e-16 ***
XPos:phase     2092  2  < 2.2e-16 ***
phase:YPos      243  2  < 2.2e-16 ***
```

1.4.3 Model Selection

- Thus far, we've not considered a range of possible models for the candidate covariates only the 'full' model both with and without the space-phase interactions. Some covariates may be valuable inclusions and some may not.
- We could adopt a forwards, backwards or stepwise approach to model selection but that might exclude some models from consideration, and so an all-possible-subsets approach is useful.

When data are overdispersed, the following quasi-AIC score can be used for model selection:

$$\text{QAIC}=-((2 \times \text{log-likelihood})/\text{dispersion parameter estimate})+2 \times \text{k}$$

where k=the number of model coefficients (e.g. intercept and slope coefficients +1 for the dispersion parameter estimate).
When we carry out all-possible-subsets selection on the overdispersed model using the QAIC (appropriate for quasi-Poisson selection), we find:

- the QAIC is better when the `phase*YPos` term is omitted from the model but `phase` is retained.
- the 'second-best' model (with weight=0.318) keeps the `phase*YPos` interaction however (at a cost of two additional parameters).
- the `phase*Xpos` interaction term is included in the 'best' model.

```
> require(MuMIn)
> dredge(glmFit3, rank="QAIC", chat=summary(glmFitOD3)$dispersion)
Global model call: glm(formula = count ~ depth + XPos * phase + YPos * phase + DistCoast,
    family = poisson, data = data, offset = log(area))
---
Model selection table
    (Int)    dpt      DsC        phs XPs       YPs        phs:XPs phs:YPs df logLik    QAIC   delta   weight
64  1252.000 -0.29000 -1.471e-07 +   3.660e-05 -2.105e-04 +              9 -275591.8 5630.2    0.00 0.682
128 1270.000 -0.28740 -1.785e-07 +   3.525e-05 -2.134e-04 +       +     11 -275470.1 5631.7    1.52 0.318
96  1117.000 -0.28620 -1.816e-07 +   5.367e-05 -1.901e-04        +      9 -276516.3 5649.0   18.82 0.000
32  1231.000 -0.29120 -1.137e-07 +   5.394e-05 -2.090e-04               7 -276965.9 5654.2   23.97 0.000
28  1222.000 -0.29320 -8.454e-08     5.414e-05 -2.076e-04               5 -277222.2 5655.4   25.19 0.000
126 1342.000 -0.30130            +   3.534e-05 -2.252e-04 +       +     10 -276987.9 5660.6   30.42 0.000
62  1211.000 -0.29830            +   4.019e-05 -2.042e-04 +              8 -277719.4 5671.5   41.31 0.000
94  1175.000 -0.29880            +   5.421e-05 -1.997e-04        +      8 -278003.6 5677.3   47.10 0.000
```

So, at this stage it appears that:

- Depth,
- Distance from coast,
- phase
- and the spatial co-ordinates

are all useful predictors of animal numbers.
Additionally, the relationship between average numbers and the X-coordinate appears to differ across phases. This suggests some redistribution in the X-direction across phases which is consistent with what we see in the data (Figure 8).

Model diagnostics
Before making any decisions based on our model, we need to be happy that model assumptions are met. We need to ensure:

1. the covariates in the model are not too similar to each other (i.e. we want to avoid 'collinearity')
2. the mean-variance relationship (variance is proportional to the mean) is appropriate
3. the data, given the model[4], are independent
4. the covariate relationships are well described as linear on the link scale

[4] so the 'left-overs' (residuals)

1.5 Collinearity

For a reliable model, we need to be sure that we don't include variables which are too similar.

In this case, Depth appears to have some sort of relationship with the Y-coordinate (Figure 7) since depth generally increases as the Y-coordinate decreases.

Collinearity:

- is a measure of the similarity between covariates and occurs when one of more of the explanatory variables can be written as a (near) linear combination of the other explanatory variables.
- relates only to near linearity in the explanatory variables; the response variable is not considered.
- is a problem because it amounts to trying to fit a plane to points while lie along a line.
  - This results in a very 'wobbly' plane since any plane supported by a line fits the data about as well as any other plane.
  - In statistical terms, this translates into (some of) the parameters defining the plane being highly uncertain (and thus having high variance).
- can be diagnosed by plotting the covariates with each other (Figure 9) and one or more numerical measures (e.g. VIFs).

1.5.1 Detecting collinearity

Variance Inflation Factors (VIFs) are one useful measure of collinearity. VIFs return one value per explanatory variable and are based on the R^2 you would get when regressing each explanatory variable on the others.

- The VIF for the jth explanatory variable is $\text{VIF}_j = \frac{1}{(1-R_j^2)}$, where R_j^2 is the squared multiple correlation coefficient (coefficient of determination) of the jth variable with the other variables.
- The closer R_j^2 is to 1, the higher the collinearity and the VIF.

If any of the covariates have more than one parameter associated with them, then generalized variance-inflation factors (GVIFs) Fox and Monette (1992) are calculated.

GVIFs represent the extent of the inflation of the confidence ellipse (the generalization of the confidence interval for multiple coefficients) for the collection of coefficients compared with what would be obtained if the covariates were unrelated to each other.

The GVIFs for a model without the interaction terms and a visual check of the support for the plane (Figure 10) reveal no collinearity issues.

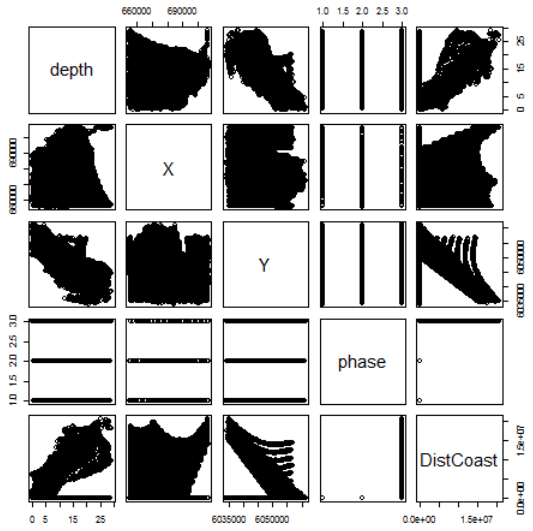


Figure 9: Pairwise scatterplots for the covariates of interest.

```
> vif(glmFitOD2)
              GVIF Df GVIF^(1/(2*Df))
depth     3.729755  1        1.931257
XPos      1.498642  1        1.224191
YPos      3.939489  1        1.984815
phase     2.327155  2        1.235112
DistCoast 2.309350  1        1.519655
```

In contrast, including Depth and a 'jittered' version of Depth (Depth with some noise added) illustrates the problem.

- Here, there is no support for the surface beyond the line of points obtained by plotting the two 'covariates' with each other (Figure 11).
- The associated GVIFs of 32.77971[5] confirm this collinearity problem and the associated problems with model stability.

Note that in a collinear situation the points can be well predicted by *any* plane running through their major axis (along the line). It is just the slope perpendicular to the major axis (which is a function of the β's) that can't be well predicted.

This is unsurprising since the points barely extend in the direction perpendicular to the line, and thus contain little information on the slope of the plane in this direction.

[5](output not shown)

Collinearity is not an issue if predictions are all that is required. However, if there is interest in the slope parameters (i.e. the β's) or the size of the standard errors about the coefficients are important, then collinearity should be avoided.

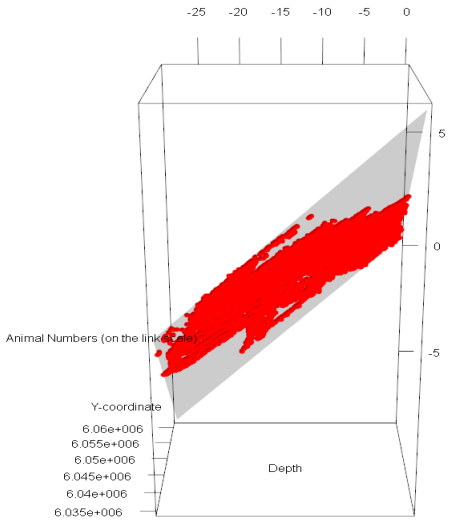


Figure 10: Fitted model (plane) on the scale of the link function when fitting Depth and the Y-coordinate to animal counts.

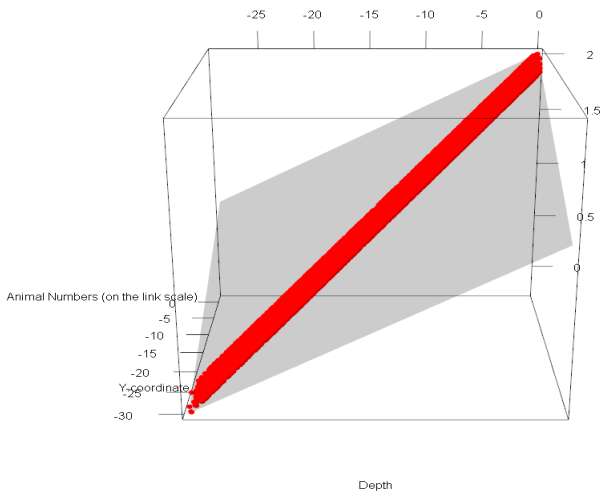


Figure 11: Fitted model (plane) on the scale of the link function when fitting Depth and a jitter version of Depth to animal counts.)

Collinearity is an issue when the interaction term is included since the XPos and phase*XPos terms are almost linear combinations of each other (phase just shifts the linear term up and down the axis on the link scale).

This results in GVIFs which are extremely high (~ 69). This can be remedied by centering the continuous covariate (XPos in this case) which reduces the GVIFs to within an acceptable range.

```
> glmFitOD4<- glm(count ~ depth+XPos*phase+YPos+DistCoast,
family=quasipoisson, offset=log(area), data=data)

> vif(glmFitOD4)
                    GVIF Df GVIF^(1/(2*Df))
depth       3.733122e+00  1        1.932129
XPos        3.048676e+00  1        1.746046
phase       2.305483e+07  2       69.293182
YPos        3.933083e+00  1        1.983200
DistCoast   2.850631e+00  1        1.688381
XPos:phase  2.353062e+07  2       69.647948
```

秃了哟提问是不是只是线性，其他关系无法体现
是的

```
> glmFitOD4<- glm(count ~ depth+scale(XPos, scale=F)*phase+
YPos+DistCoast, family=quasipoisson, offset=log(area),
data=data)

> vif(glmFitOD4)
                                    GVIF Df GVIF^(1/(2*Df))
depth                           3.733122  1        1.932129
scale(XPos, scale = F)          3.048676  1        1.746046
phase                           3.022438  2        1.318528
YPos                            3.933083  1        1.983200
DistCoast                       2.850631  1        1.688381
scale(XPos, scale = F):phase    5.835716  2        1.554260
```

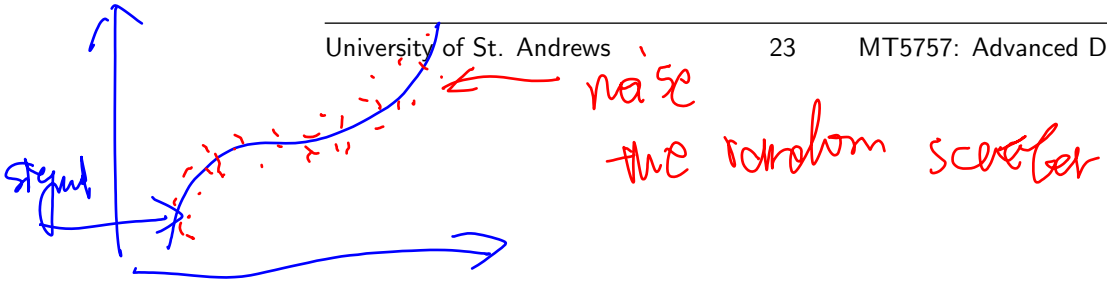
1.5.2 Dealing with collinearity

In many cases, centering the covariate doesn't help (e.g. Figure 11) and dealing with collinearity necessarily involves model selection:

- either variables are omitted completely from a model (e.g. all-possible subsets selection) or
- all variables are 'retained' in the model but one or more coefficients are down-weighted ('penalised') in some way, so some are more 'in' than others or
- some combination of both

Choosing model complexity
The response data contains elements of both signal and noise:

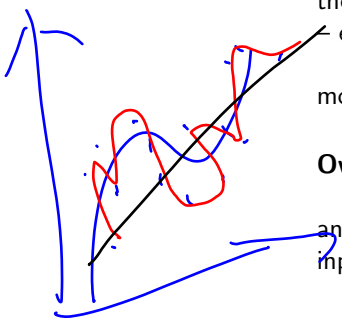
- The signal component describes the underlying mean function of the process



- The noise component represents the variability of the points about the underlying mean function

We want a model that describes the underlying mean function (the signal) well but leaves the noise element unmodelled.

red: too complex
black: too simple



We want a model which will return predictions which are as close as possible to the underlying mean function for several data sets generated from the same process – even for data which, solely by chance, was unseen by the model.

We can have the wrong mean model because the model is too complex or the model is too simple.

Overly complex models

An overly complex model has many parameters and so will model *both* the signal and the noise in the input data, and produce predictions which are very close to the input data.

Since the input data (signal + noise) come from a process with some mean (for a set of covariate values) these predictions will, on average (across many data sets generated from the same process), be close/equal to the true mean and therefore exhibit **low bias**.

While over-fitted models have low bias, they can return wildly different parameter estimates across different data sets from the same process since each new data set has a different noise element and the model accommodates the noise (to some extent) in each case.

For this reason, the parameter estimates will have **high variance** across different data sets sampled from the same process.

Overly simplistic models

A model which is too simple has too few parameters and models *neither* the signal nor the noise elements in the input data very well. This will produce predictions which are systematically wrong, and different from the underlying function on average and have **high bias**.

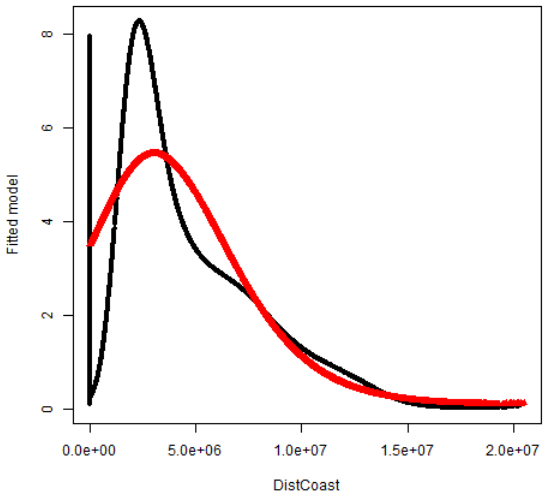
While underfitted models have high bias, simple models are very inflexible and so the parameter estimates don't vary much across data sets and will have **low variance** across different data sets sampled from the same process.

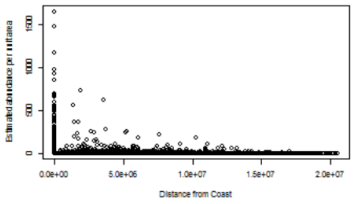
The bias-variance trade-off

For these reasons we wish to find a balance between overfitted models (with low bias and high variance) and underfitted models (with high bias and low variance)

The expected prediction error (EPE) is one way to quantify model performance for new data sampled from the same process but was unseen by the model.

A model based on sampled data can predict new data poorly for two main reasons:





P8

Figure 12: Examples of over-fitted (variable black line) and under-fitted (smoother red line) models to distance from coast covariate.

1 The model is wrong (and different from the underlying model). For example, we might have a model without all the necessary covariates or we might have all of the relevant covariates in the model but they are modelled incorrectly (too simply, or too complex).

This would return predictions which are poor for new data because the mean is not well described.

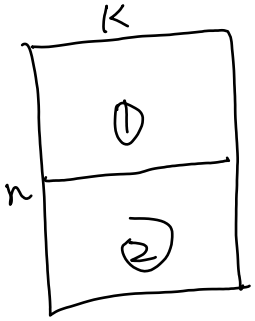
2 We have the correct model, but the process generating the data (with some mean function/model) is very noisy and this can return parameter estimates for any particular data set which are, just by chance, very different from their true values.

This would return predictions which are poor for new data because (just due to bad luck) the coefficients for the mean model are distant from the parameters and the new data are unlikely to be similarly different from the mean.

Validation set based selection

In data-rich situations, you can address this problem by dividing your sample data into two parts: a "training" sample and a "validation" sample.

Here,



Train : Fit ①

Validate: predict ②

in reality we dont use half and half

1. The training sample is used to fit the model.
2. The validation sample is used to estimate the expected prediction error for that model (since it is data unseen by the model).
3. A model is then chosen by finding the lowest expected prediction error across the range of candidate models (which will likely vary in the number and content of the covariates)

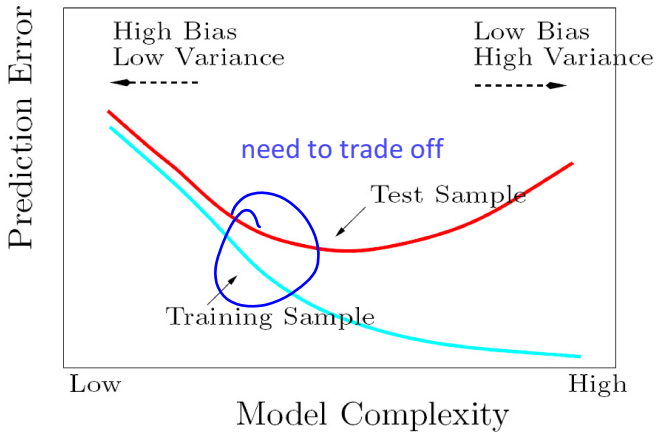
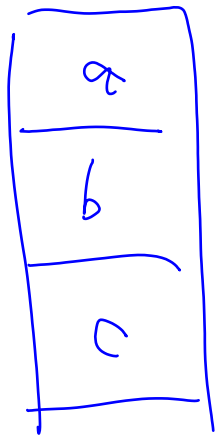


Figure 13: Behaviour of test sample and training sample error as the model complexity is varied (*Source: Hastie et al. (2001)*).

The model selection compromise (Figure 13), illustrates that:

- the prediction error on the training sample (i.e. how well the model fits the data) gets smaller as model complexity increases because a more complex model is better able to reproduce the training data.
- The prediction error on the validation/test sample (i.e. how well the model predicts new data) also decreases initially as model complexity increases, but beyond a point results in an increase in prediction error.
  - This is because overly complex models model the randomness (the noise) in the training sample (not just the features of the underlying function), and so predicts the validation/test sample less well.

QUestion:
choose the two or three part randomly or just devide into three part.



k-fold cross-validation (CV; Hastie et al., 2009, James et al., 2017) is a model selection procedure based on prediction error which involves:

- splitting the data into k samples (e.g. 10 folds)
- leaving out some subset of the data (e.g. 1 fold)

times Fit Validate
1 ab c MSE
2 a c b MSE
3 b c a MSE

$CV = \frac{\Sigma MSE}{3}$

有关correlation的没听到

- predicting the response values in the omitted set using the bulk of the data (e.g. 9 folds)

In this scheme, each fold is used once as a validation set and $k-1$ (e.g. 9) times as part of a larger training set.

Typically the expected prediction error (EPE) is calculated using the sum of the squared differences between the validation data (y^*_{it}) and predictions for the validation data based on the model fitted to the training data($\hat{y}_{it}$):

$$\sum_{i=1}^{s}\sum_{t=1}^{n_i}(y^*_{it}-\hat{y}_{it})^2 \tag{8}$$

When CV is too computationally expensive, and/or in data poor situations, the Akaike's Information Criterion (AIC) and Bayes' Information Criterion (BIC) can be used to try and balance the bias-variance trade-off without splitting samples into training and validation sets.

make it too simple

alpha = 0

1.5.3 Ridge Regression

panlise near 0 but not 0, retain all parameters

A popular alternative to modelling correlated covariates is to penalise the coefficients so they are closer to zero than they would be under ordinary least squares/ML based estimation (Hastie et al., 2009; James et al., 2017).

The benefit of doing this is that the variance of the fitted functions across several data sets from the same process is reduced, but at the cost of bias (i.e. predictions averaged over several samples from the same process will be systematically too large or too small).

pick the lambda first and then try a whole bunch of the and choose the best one (smallest CV score)

For GLMs we find the estimated coefficients (given some value for λ) by minimising the negative log likelihood ($\log(L)$) and the associated penalty (for the p coefficients):

fit penalty

lambda need to be found

$$-\log(L)+\lambda\sum_{j=1}^{p}\beta_j^2, \qquad \lambda>0 \tag{9}$$

So large values (and sum) of the coefficients will lead to a large penalty term and be discouraged; the extent of this depends on the size of λ and if large coefficients improve the fit to the data (which is signalled by larger, negative values of the log-likelihood).

Ridge regression for Poisson data (excluding any interaction terms) can be fitted using the glmnet function in the R library with the same name (with alpha=0, for reasons we see in section 1.5.5) and cross-validation can be used to find the best value for λ.

给出一系列的λ会得到最好的那一个

model.matrix(fit2)

```
> ridge<- glmnet(xmatrix,data$count,family="poisson",
offset=log(area), alpha=0)
> cvridge<- cv.glmnet(xmatrix,data$count,family="poisson",
alpha=0, offset=log(area), nfolds=10)
> par(mfrow=c(1,2))
> plot(ridge, xvar="lambda")
```

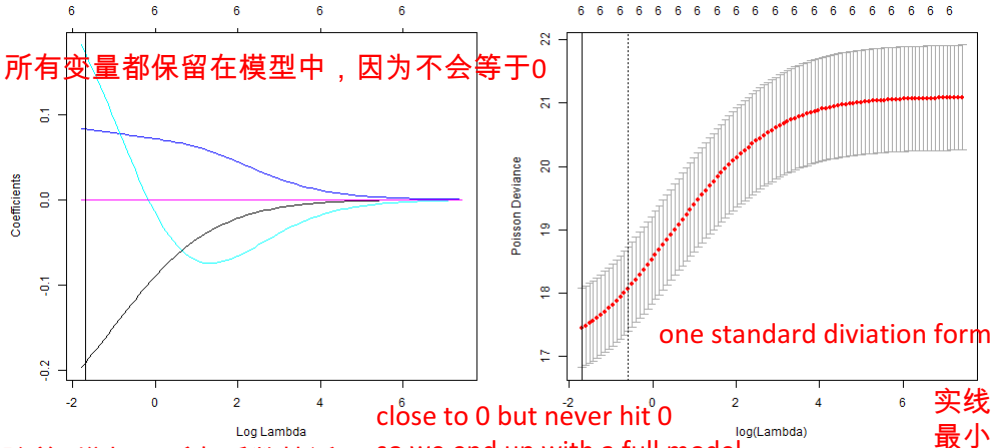
给出最好的λ

```
> abline(v=log(cvridge$lambda.min))
> plot(cvridge)
> abline(v=log(cvridge$lambda.min))
```

In this case, no penalty was really necessary (there was no collinearity concerns) and so the results are almost identical to the maximum likelihood solutions.

While ridge regression estimates are more stable (their variance is reduced relative to unpenalised approaches) this method does not 'select' predictors or give an easily interpretable model, since all covariates are always present in the model (to some extent).

For this reason, the Lasso method is often used instead.



所有变量都保留在模型中，因为不会等于0

红点是CV score uncertainty(灰色) 小好

one standard diviation form the solid line

实线是数据中最小λ: 是最小lambda还是取最小CV的lambda

随着λ增长，所有系数接近0

close to 0 but never hit 0
so we end up with a full model
all the time

Figure 14: Ridge regression based coefficient values and the Poisson-based deviance with λ. The solid vertical line in each plot represents the 'best' estimate for λ based on cross-validation.

alpha = 1

1.5.4 Lasso

The least absolute shrinkage and selection operator (Lasso; Hastie et al., 2009; James et al., 2017) undertakes both shrinkage and automatic variable selection at the same time[6].

In this case, the penalty contains the sum of the absolute value of the coefficients (rather than the sum of the squared coefficients).

For GLMs we find the estimated coefficients (given some value for λ) by minimising the sum of the negative log likelihood and the associated penalty (for the p coefficients):

[6] see 'http://stats.stackexchange.com/questions/74542/why-does-the-lasso-provide-variable-selection' for a theoretical discussion about how this occurs

$$-log(L) + \lambda \sum_{j=1}^{p} |\beta_j|, \qquad \lambda > 0 \tag{10}$$

So large values (and sum) of the coefficients will be discouraged depending on λ and how the size of the coefficients improve the fit to the data (quantified by the log-likelihood).

The estimated coefficients are shrunk towards zero and some may be estimated as *exactly* zero; this is not possible under ridge regression estimation.

Like all methods, the lasso has limitations:

- when $n > p$, ridge regression tends to return better predictions if there are high correlations amongst the covariates
- when $p > n$ it 'selects' n variables at most
- The lasso method does not 'group' predictors and so if there a group of highly correlated predictors, the lasso tends to 'select' one of these variables from this group, in an arbitrary way.

Lasso regression for Poisson data (excluding any interaction terms) can also be fitted using the `glmnet` function in the `R` library with the same name (with `alpha=1`, for reasons we see in section 1.5.5) and cross-validation can be used to find the best value for λ:

```
> lasso<-glmnet(xmatrix,data$count,family="poisson",
offset=log(area), alpha=1)
> cvlasso=cv.glmnet(xmatrix,data$count,family="poisson",
alpha=1, offset=log(area), nfolds=10)
> par(mfrow=c(1,2))
> plot(lasso, xvar="lambda")
> abline(v=log(cvlasso$lambda.min))
> plot(cvlasso)
> abline(v=log(cvlasso$lambda.min))
```

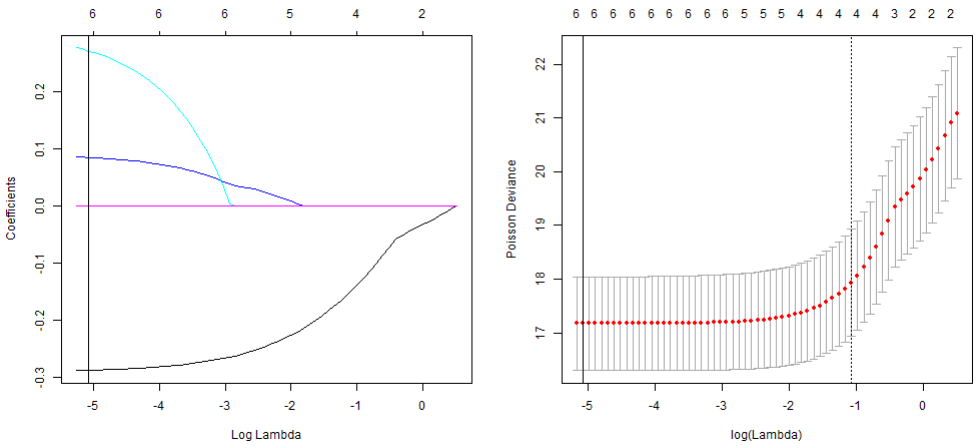


Figure 15: Lasso based coefficient values and the Poisson-based deviance with λ. The solid vertical line in each plot represents the 'best' estimate for λ based on cross-validation.

1.5.5 Elastic Net

The 'elastic net' method provides a way to group correlated predictors and is a combination of the lasso and ridge regression. As a result it has two penalty parameters.

For GLMs we obtain the coefficients by minimising the sum of the negative log likelihood and the two parameter penalty:

$$-log(L) + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{11}$$

The penalty can be re-written to involve just one parameter, α:

$$(1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \tag{12}$$

and so when $\alpha = 1$ the penalty reduces to the lasso and when $\alpha = 0$ the penalty reduces to the ridge regression solution. When $0 < \alpha < 1$ it returns the elastic net solution.

```
> attach(data)
> xmatrix<-model.matrix(glmFitOD2)
> cvenet<- cv.glmnet(xmatrix,data$count,family="poisson",
offset=log(area), nfolds=10)
```

default alpha is 1(lasso)

In this case, the penalty chosen under cross-validation was very weak (Figure 16) and so the coefficients under the elastic net approach are very similar to (and within the 95% confidence intervals for) the overdispersed Poisson-based model (Figure 17).

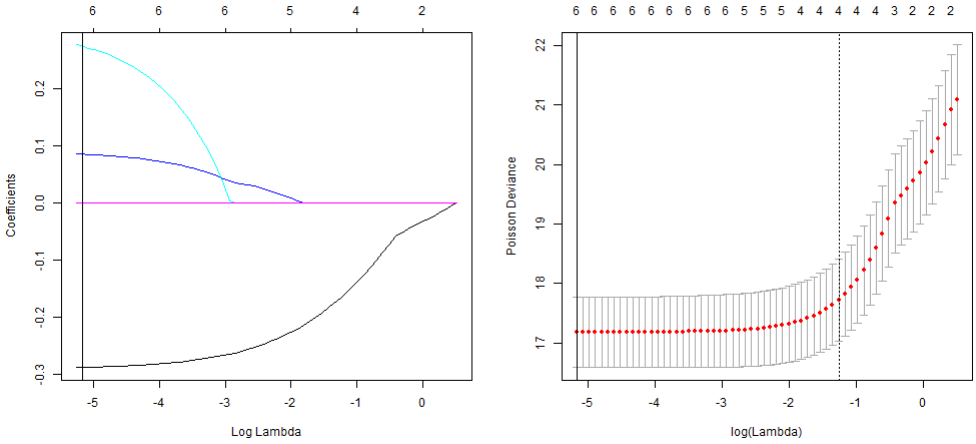


Figure 16: Elastic net based coefficient values and the Poisson-based deviance with λ (defined as α in Equation 12). The solid vertical line in each plot represents the 'best' estimate for α in Equation 12 based on cross-validation.

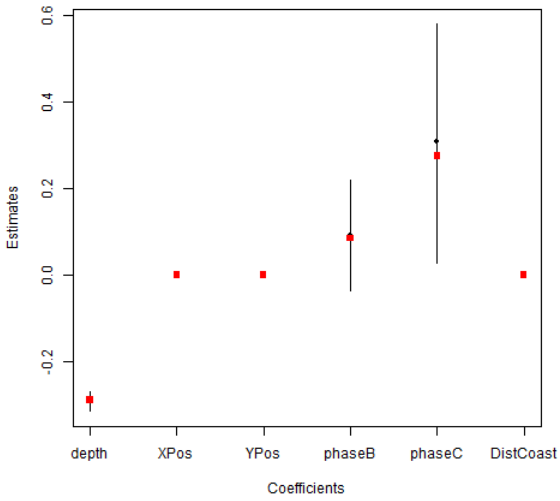


Figure 17: Coefficients for the elastic net model with penalty chosen using cross-validation (red squares) compared with the Poisson-based model (black circles). The 95% confidence intervals for the Poisson-based models are shown as the vertical black lines.

只有非常多covariates的时候需要，老师不太用得到，但是以后可能会遇到这样的问题

怎样选alpha：可以给出一系列alpha from 0 to 1 然后比较MSE(均方差)

1.6 Temporal autocorrelation

In this case we have assumed the data come from an overdispersed Poisson distribution with the variance proportional to the mean (and $\hat{\phi} \approx 98$). We have also assumed the errors are independent.

If the model does not capture the patterns in the response data, then some of this pattern will remain in model residuals and present as positive/negative autocorrelation.

- For instance, animals along transects are often clustered together and counts observed close together in time are more similar than counts distant in time/space (Figure 18).

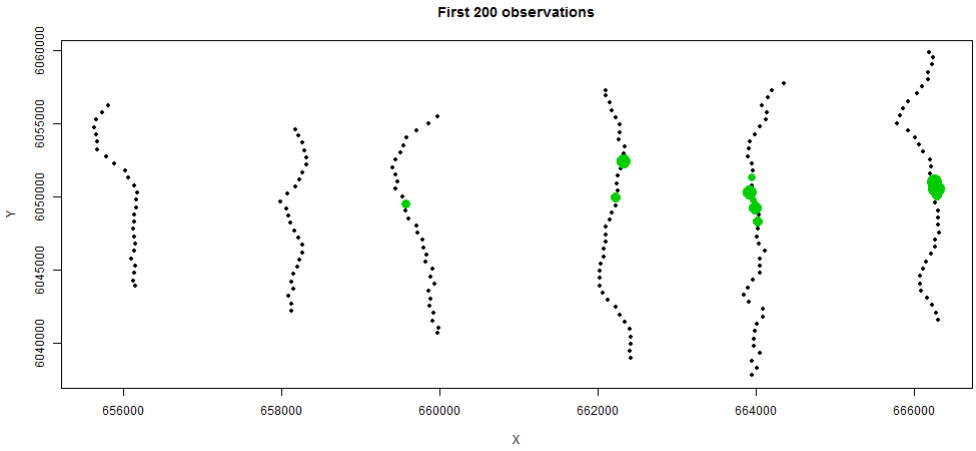


Figure 18: Response data along the transects for the first 200 observations in the `glmFitOD3` model. The size of the points indicates relative numbers.

- To some extent, model covariates will be able to explain the similarity in these counts along transects (left-hand plot; Figure 19), however some of this pattern tends to remain unexplained by the model and is found in model residuals (right-hand plot; Figure 19).

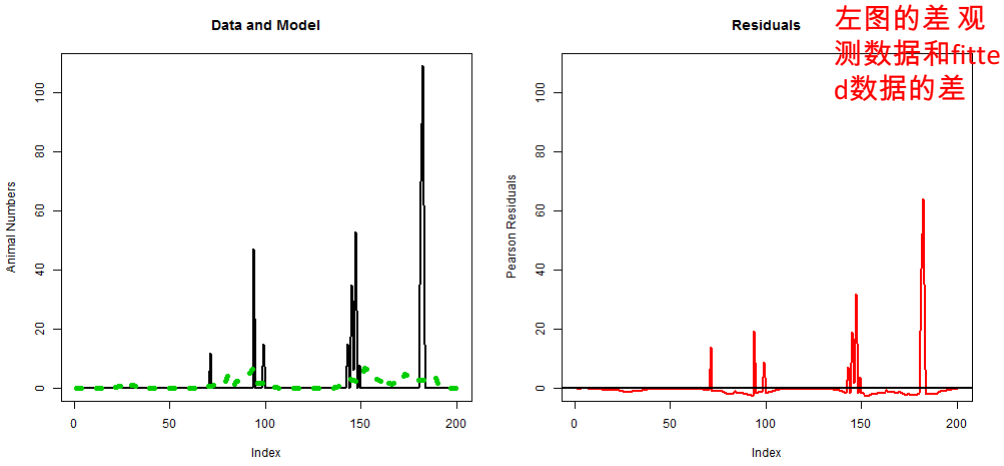


Figure 19: Response data (black line; left-hand plot) and fitted values (green dotted line; left-hand plot) and the corresponding Pearsons residuals (right-hand plot) for the first 200 observations in the `glmFitOD4` model

Diagnosing correlation using the Runs Test 绝对值大于2则有correlation问题

大p好，
没有correlation

Non-independence in model residuals can be diagnosed using a Wald-Wolfowitz runs test (Wald and Wolfowitz, 1943):

all positve -> 1
all negative -> -1

- This test assigns a 'sign' to each residual: positive residuals are labelled $+1$'s and negative residuals are labelled -1's
- The number of uninterrupted strings ('runs') of positive and negative residuals when looked at in sequence (e.g. time order) is then calculated
- Too few (long) runs provide evidence of positive correlation, while too many (short) runs provide evidence of negative correlation

run: 转变次数

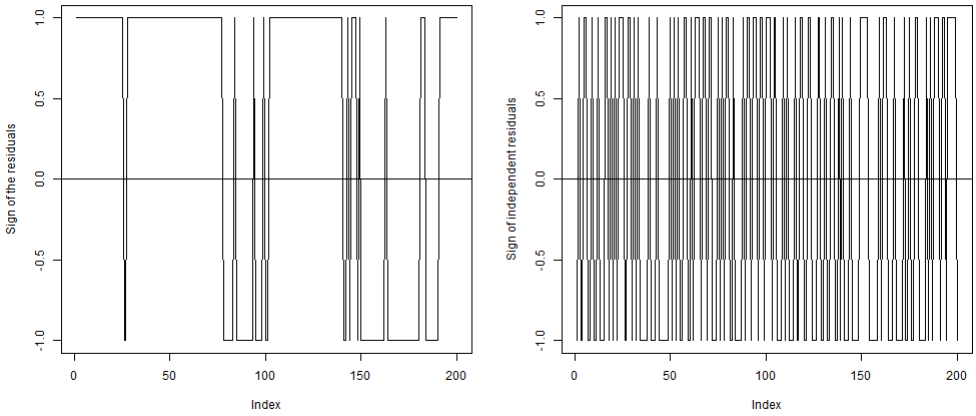


Figure 20: The sign of model residuals for the first 200 observations (left-hand plot) and the sign of 200 independent residuals (right-hand plot).

We can see that residuals are correlated across time and exhibit fewer, longer, runs (left-hand plot, figure 20) than independent residuals which show lots of short runs (right-hand plot, figure 20). This correlation over time is confirmed using a runs-test.

The runs test compares the observed number of runs (T) with what's expected under independence (E(T))[7] and adjusted for the variance (V(T))[8] to give a test

[7]
$$E(T) = \frac{2n_p n_n}{n_p + n_n} + 1$$
, where n_p is the number of positive residuals and n_n is the number of negative residuals.

[8]
$$V(T) = \frac{2n_p n_n (2n_p n_n - n_p - n_n)}{(n_p + n_n)^2 (n_p + n_n - 1)}$$

statistic W_z[9] which has a standard Normal(0,1) distribution.

So, values more extreme than ± 2 are considered compelling evidence against independence and consistent with positive ($W_Z < -2$) or negative correlation ($W_Z > 2$).

This test is based on the order of model residuals and thus has many uses. This test can be used to check for:

- time dependence (by calculating runs based on residuals in time order) or
- poorly specified covariate relationships (by calculating runs based on covariate value).

Positive correlation affects model selection

Positive correlation in model residuals can lead us to conclude that irrelevant variables are important. To illustrate this, we've made two types of Poisson data:

The first type (based on η_1 and μ_1) was created using x1 and the intercept (β_0) and slope (β_1) parameters.

```
> x1<- rep(1:10, times=100)
> b0<-0.1
> b1<-0.1
> eta1<-b0+b1*x1
> mu1<- exp(eta1)
> y1<- rpois(1000,mu1)
```

The second type (based on η_2 and μ_2) was also created using x1, β_0 and β_1 and also a sine wave/function (to mimic factors that exist in the environment that affect animal numbers).

```
> x1<- rep(1:10, times=100)
> b0<-0.1
> b1<-0.1
> eta2<- b0+b1*x1+sin(1:1000)*2
> mu2<- exp(eta2)
> y2<- rpois(1000,mu2)
```

The same Poisson-based model was fitted to several sets of data from each model type (generated from a Poisson distribution with mean mu1 or mu2). This model includes the x1 covariate but excludes the sine wave (used to generate the y2 outcomes) in order to mimic the absence of important covariates for the second set of models:

[9] $W_Z = \frac{T - E(T)}{\sqrt{V(T)}}$ (what we see ← what we expect)

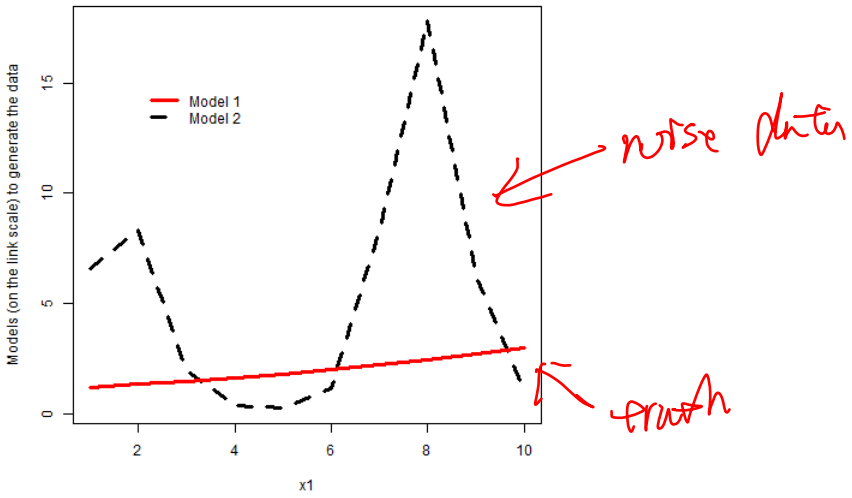


Figure 21: The two mean models used to illustrate positive correlation.

```
> fit1<- glm(y1 ~ x1, family=poisson)
> fit2<- glm(y2 ~ x1, family=poisson)
```

For one realisation we see that the residuals for the `fit2` model are correlated in observation order (Figure 22). This is in-keeping with how `mu2` was created; the runs test confirms residual-based correlation for the `fit2` model:

```
> require(lawstat)
> runs.test(residuals(fit1, type="pearson"))
Runs Test - Two sided
data:  residuals(fit1, type = "pearson")
Standardized Runs Statistic = -1.3649, p-value = 0.1723

> runs.test(residuals(fit2, type="pearson"))
Runs Test - Two sided
data:  residuals(fit2, type = "pearson")
Standardized Runs Statistic = -10.757, p-value < 2.2e-16
```

For these two data sets, we can also see that a covariate which is unrelated to the response and entirely made of randomly generated noise (`x2`) is found to be significantly related to the `y2` response variable (in the `fit2b` model).

This is not the case for the `y1` response variable (in the `fit1b` model):

residual by observation order

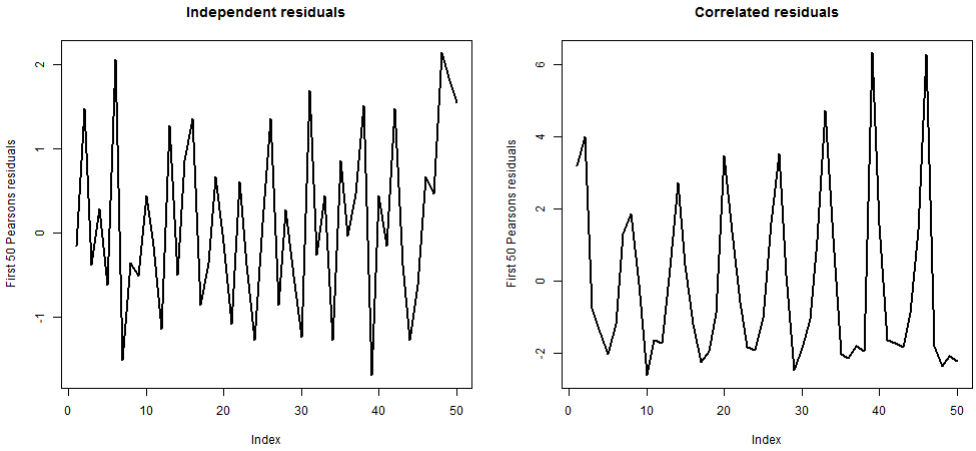


Figure 22: The first 50 Pearsons residuals for independent and correlated residuals.

```
> x2<- rnorm(1000,0,1)
> fit1b<-glm(y1 ~ x1+x2, family=poisson)
> fit2b<- glm(y2 ~ x1+x2, family=poisson)
> Anova(fit1b)
Analysis of Deviance Table (Type II tests)

Response: y1
   LR Chisq Df Pr(>Chisq)
x1  163.248  1     <2e-16 ***
x2    0.468  1     0.4941
---

> Anova(fit2b)
Analysis of Deviance Table (Type II tests)

Response: y2
   LR Chisq Df Pr(>Chisq)
x1   396.78  1    < 2e-16 ***
x2     5.59  1    0.01809 *
---
```

To ensure this result was not a peculiar event, this process was repeated 5000 times and the long-run behaviour examined. We see:

For the data generated using μ_1,

- the unrelated covariate (x_2) was falsely identified as statistically significant just 5.32% of the time (in line with the expected false positive rate of 5%)
- the runs test over-reported the presence of correlation in model residuals (even when the correct model was fitted);

the runs-test indicated the residuals were non-independent for 20% of the simulated sets using the 5% level to determine statistical significance (and we would expect this to be 5%).

- This over-reporting reduced to a more respectable 6.4% if 0.01 was used as a threshold instead.

For the data generated using μ_2,

- the unrelated covariate (x_2) was falsely identified as statistically significant 62.60% of the time
- the runs test suggested the residuals were positively correlated in every case (regardless if the 5% or 1% level was used as a threshold to determine significance).

Positive residual correlation is also apparent for our working model:

```
> require(lawstat)
> runs.test(residuals(glmFitOD3, type="pearson"))

Runs Test - Two sided

data:  residuals(glmFitOD3, type = "pearson")
Standardized Runs Statistic = -119.5029, p-value < 2.2e-16
```

and so, at this point the reported standard errors and p-values are likely to be too small.

We can update this model to incorporate the autocorrelation in model residuals using Generalized Estimating Equations (GEEs, section 3).

1.7 Nonlinearity on the link scale

At this stage, the model assumes the relationships between the covariates and the response on the link scale are linear. If this is unreasonable, model predictions (and any associated confidence intervals) can be poor.

Recall, if the relationship is not well described by the model, partial residual plots should exhibit curvature. To aid visual inspection, a (quadratic) term can be fitted to each numeric predictor and overlaid on each plot.

A quadratic relationship can also be fitted between the Pearsons residuals (r_{pit}[10]) and a squared term for each (j-th) covariate.

A quadratic term can also be fitted for each predictor (e.g. $Depth^2$) in the regression model directly as a test for nonlinearity. This provides a z/t-test statistic

[10] $(y_{it} - \hat{y}_{it})/(\sqrt{\hat{y}_{it}})$

and associated p-value for each quadratic term in the model, as a formal test (Tukey's test for non-additivity) for nonlinearity.

While this test will not be appropriate for all types of departures from linearity, small p-values for this test indicate nonlinearities on the link scale.

These partial plots can also be augmented with a more heavily data-based curve. The details of these smooth curves will be covered in section 2, but can be thought of as a kind of running mean - where a window is moved across the covariate range the average of the values inside each window is found.

These average values are then joined across windows (in clever ways) to make a fitted curve. These fitted curves are heavily based on the data rather than rely heavily on model(s) specified in advance.

Both quadratic and smoother-based curves were fitted the residual plots for our working model. In this case, there appears to be some nonlinearities on the link scale:

- Based on both curve types, curvature is apparent in all model covariates (Figures 23 – 26).
- The test statistics[11] for the coefficients associated with the quadratic term (x_j^2) for each covariate are all extremely large and the associated p-values are small and significant at the 1% level.

[11] Tukey's test for non-additivity

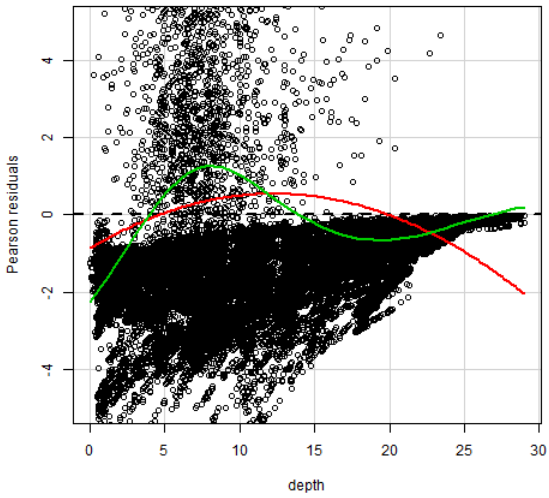


Figure 23: Residual plots for the `glmFitOD3` model: Depth covariate

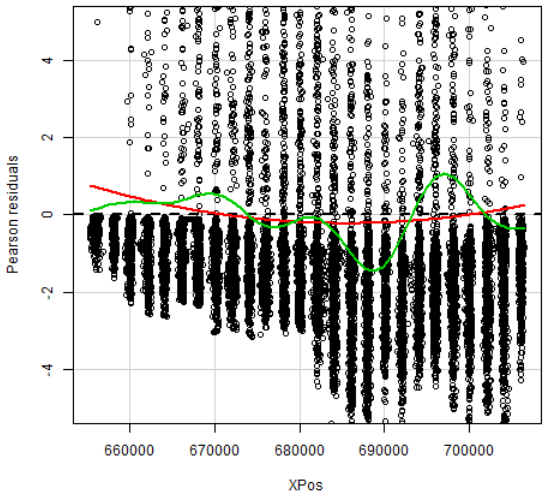


Figure 24: Residual plots for the `glmFitOD3` model: XPos covariate

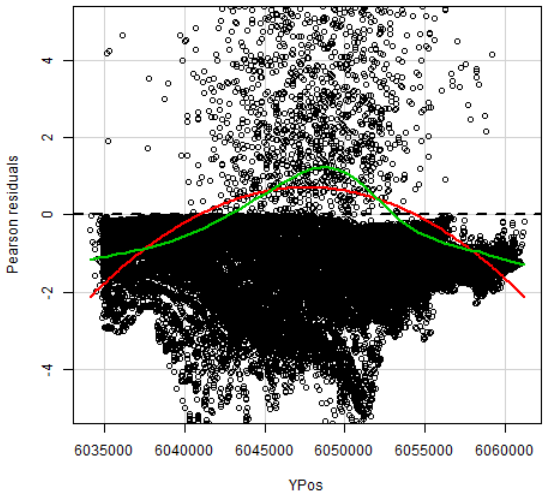


Figure 25: Residual plots for the `glmFitOD3` model: YPos covariate

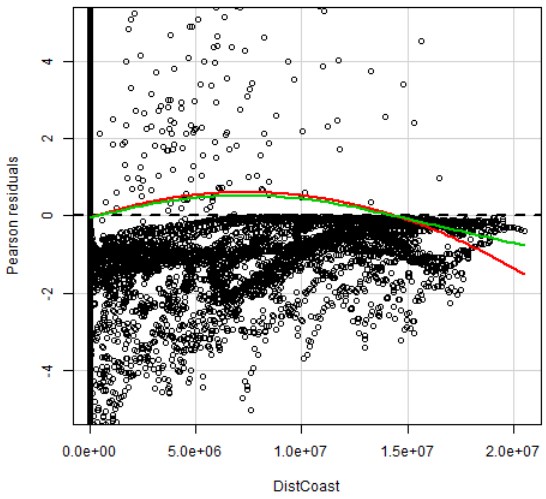


Figure 26: Residual plots for the `glmFitOD3` model: DistCoast covariate

```
> require(car)
```

```
> residualPlots(glmFitOD3, quadratic=T, type = "pearson")
           Test stat Pr(>|t|)
depth      43443.666        0
XPos        2202.237        0
phase             NA       NA
YPos       27759.164        0
DistCoast   2057.174        0
```

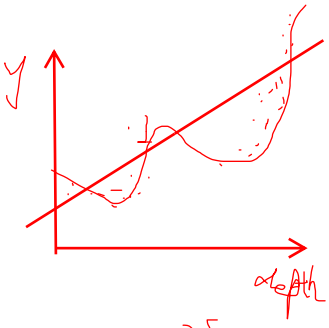
如果quadratic term is useful, you will get a small p-value

NA 是因为phase 是 factor variable

We are going to update the model to permit (data-driven) smooth functions on the link scale for all covariates using Generalized Additive Models (GAMs; section 2).

2 Generalized Additive models

Generalized Additive Models (GAMs) are an extension of GLMs that can implement a wide range of nonlinear relationships between the response variable and the covariates.

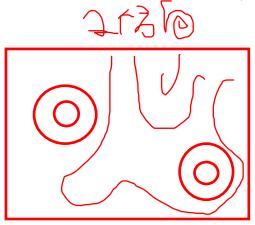


We first discuss basis functions for one-dimensional smoothers then for two dimensional smoothers, fitting models to the data as we proceed.

Some examples where you can find some more information on GAMs and smoothing methods are James et al. (2017) and Wood (2017). MGCV

2.1 Polynomials

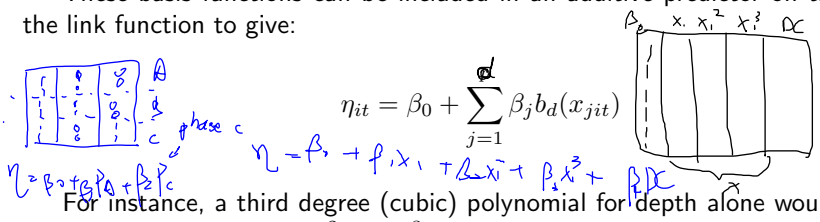
2.1.1 Model specification



Polynomials can be thought of as a basis expansion with as many columns as the degree of the polynomial:

$$b_d(x_{jit}) = (x_{jit})^d \tag{13}$$

These basis functions can be included in an additive predictor on the scale of the link function to give:



$$\eta_{it} = \beta_0 + \sum_{j=1}^{d} \beta_j b_d(x_{jit}) \tag{14}$$

For instance, a third degree (cubic) polynomial for depth alone would result in a basis matrix with x and x^2 and x^3 as columns and as many rows (N) as we have in the data set. More covariates could be added in the usual way to the additive predictor.

The columns for the $d \times N$ polynomial basis matrix for the depth covariate are plotted in Figure 27.

2.1.2 Model fitting & selection

Model fitting proceeds via least-squares/Maximum Likelihood/Quasi-Likelihood (depending on the mean-variance relationship assumed under the model) to give fitted curves.

For example, Poisson-based ML was used to return fitted curves for the count data (containing depth alone) with d=$1, 2, 3, \& \ 4$ polynomials. The results on the link and response scales are shown in Figure 28. The plot on the left-hand side represent the fitted function on the (log) link scale for $d = 1, ..., 4$:

$$\hat{\eta}_{it} = \hat{\beta}_0 + \hat{\beta}_1 x_{it} + \log(area_{it}) \quad (15)$$
$$\hat{\eta}_{it} = \hat{\beta}_0 + \hat{\beta}_1 x_{it} + \hat{\beta}_2 x_{it}^2 + \log(area_{it}) \quad (16)$$
$$\hat{\eta}_{it} = \hat{\beta}_0 + \hat{\beta}_1 x_{it} + \hat{\beta}_2 x_{it}^2 + \hat{\beta}_3 x_{it}^3 + \log(area_{it}) \quad (17)$$
$$\hat{\eta}_{it} = \hat{\beta}_0 + \hat{\beta}_1 x_{it} + \hat{\beta}_2 x_{it}^2 + \hat{\beta}_3 x_{it}^3 + \hat{\beta}_4 x_{it}^4 + \log(area_{it}) \quad (18)$$

while the plot on the right hand side of contains the fitted values on the response scale for $d = 1, ..., 4$:

$$\hat{\mu}_{it} = \exp(\hat{\beta}_0 + \hat{\beta}_1 x_{it}) \times area_{it} \quad (19)$$
$$\hat{\mu}_{it} = \exp(\hat{\beta}_0 + \hat{\beta}_1 x_{it} + \hat{\beta}_2 x_{it}^2) \times area_{it} \quad (20)$$
$$\hat{\mu}_{it} = \exp(\hat{\beta}_0 + \hat{\beta}_1 x_{it} + \hat{\beta}_2 x_{it}^2 + \hat{\beta}_3 x_{it}^3) \times area_{it} \quad (21)$$
$$\hat{\mu}_{it} = \exp(\hat{\beta}_0 + \hat{\beta}_1 x_{it} + \hat{\beta}_2 x_{it}^2 + \hat{\beta}_3 x_{it}^3 + \hat{\beta}_4 x_{it}^4) \times area_{it} \quad (22)$$

In this case, the fitted functions on the link scale are very similar for depth values < 20m for polynomial-based curves with $d \geq 2$.

After 20 metres, the curves notably diverge and standard model selection criteria would help discriminate between different degree polynomials.

Under ML, AIC or BIC scores could be used to choose between polynomials of different degrees, while QAIC scores can assist here if overdispersion was considered and QL employed for model fitting.

Fit and residual plots can also help the user choose between competing models.

2.1.3 Limitations

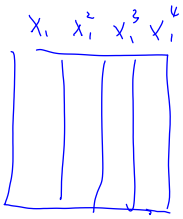
While polynomials are easy to understand and easy to implement, they have some undesirable properties:

- The numbers in the basis matrix get very large very quickly (page 47), which can make estimation difficult
- Polynomials can exhibit oscillatory behaviour in the fitted curves for high degree polynomials (regardless of the underlying covariate relationship)
- Polynomials are also 'evenly' smooth; the user is unable to allow more flexibility in some areas of the curve than in others
- Polynomials are global functions - every row of the data set (regardless of the depth value) contributes in some way to every column in the basis matrix.

  - So, the fitted curve for the very high depth values are also affected to some extent by the relationship for the very low depth values and vice versa

This global property can be a real issue when you have even moderately uneven smoothness in the underlying function.

- For instance, the response data was modified to include a peak in counts per unit area at 5 metres along with a peak in the same metric at 10 metres (Figure 29).
- This curve was poorly modelled by the cubic polynomial (it introduced a peak between 5 and 10 metres; red curve in Figure 29).
- The curve based on a 5th degree polynomial characterised the underlying curve somewhat better then the cubic, but still failed to rise sufficiently high for the peaks and fall sufficiently low between the peaks (green curve in Figure 29).
- A further increase in flexibility was also not the answer; the 10th degree polynomial instead overestimated the height of the peaks at 5 and 10 metres and failed to fall as low as required between the peaks (dark blue curve in Figure 29).
- The 20th degree polynomial did not converge and consequently returned a nonsense curve (light blue curves in Figure 29).
- For this reason (and the reasons listed above), more locally defined (or locally acting) basis functions are often preferred over polynomials.



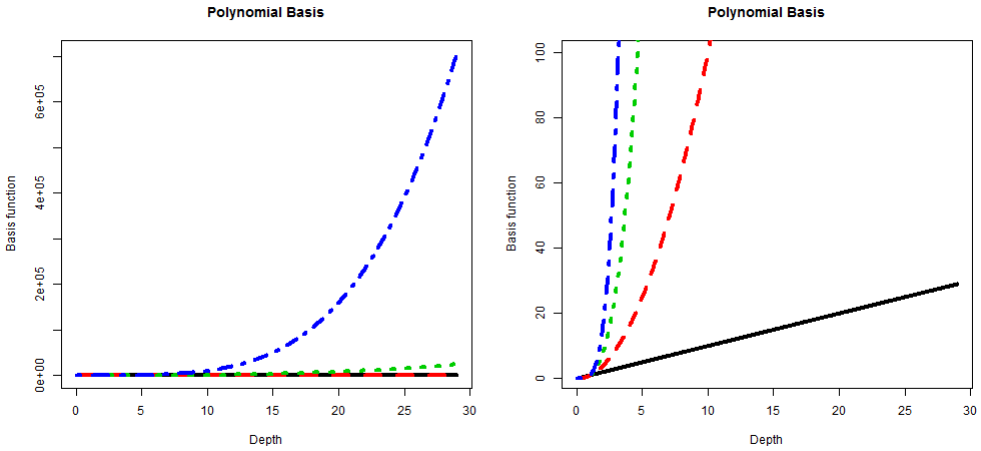


Figure 27: Polynomial basis function for depth with $d = 4$ and thus, 4 parameters. The black solid line is the first column of the $n \times 4$ matrix for the polynomial basis expansion for depth (x), the red dashed line is the second column (x^2), the green dotted line is the third column (x^3) and the blue dotted & dashed line is the fourth column of this same matrix (x^4) (see also Equation 18). Note: the right hand plot illustrates only the lower lower end of the y-value range for clarity.

```
> head(cbind(depth, depth**2, depth**3, depth**4))
[1,] 27.42 751.8564 20615.90 565288.0
[2,] 27.59 761.2081 21001.73 579437.8
[3,] 28.52 813.3904 23197.89 661603.9
[4,] 28.05 786.8025 22069.81 619058.2
[5,] 27.60 761.7600 21024.58 580278.3
[6,] 27.34 747.4756 20435.98 558719.8
> depth2<- depth**2
> depth3<- depth**3
> depth4<- depth**4
> poly4<- glm(count ~ depth+depth2+depth3+depth4,
data=data, family=poisson)
```

```
> vif(poly4)
depth    depth2    depth3    depth4
252.7578 2514.3719 3352.8331  563.5955
> depth2<- poly(depth, degree=4)[,2]
> depth3<- poly(depth, degree=4)[,3]
> depth4<- poly(depth, degree=4)[,4]
> poly4<- glm(count ~ depth+depth2+depth3+depth4,
data=data, family=poisson)
> vif(poly4)
depth    depth2    depth3    depth4
8.223004 13.860734  6.474782  5.321257
```

也可以用poly function, 有现成函数

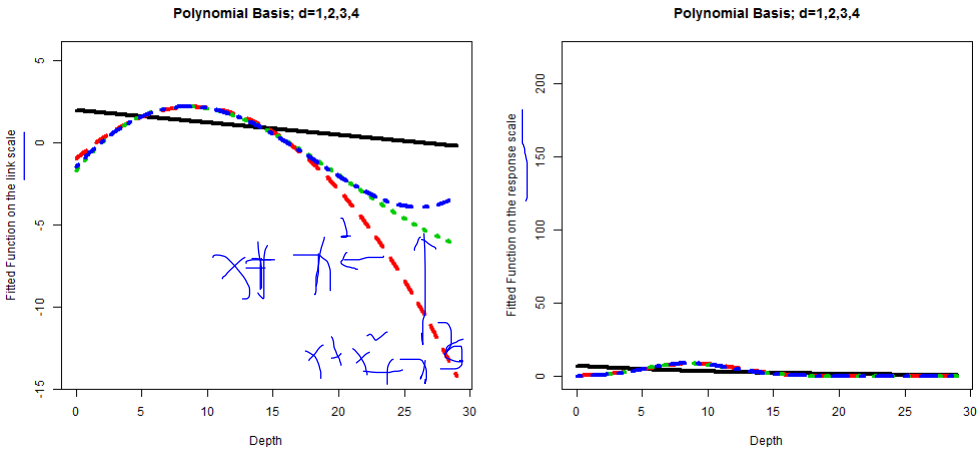


Figure 28: Polynomial fitted functions for depth with $d = 1, 2, 3$ & 4 and thus, either 2, 3, 4 or 5 parameters including the intercept term. The black solid line is the fitted relationship on the link scale (left-hand plot) and response scale (right-hand plot) for a polynomial of degree=1 (linear term; Equations 15 & 19). The red dashed line is for a degree 2 polynomial (quadratic) on both scales (Equations 16 & 20), the green dotted line represents a cubic polynomial (Equations 17 & 21) and the blue dotted & dashed line is the degree 4 polynomial (Equations 18 & 22) fitted relationship on both scales.

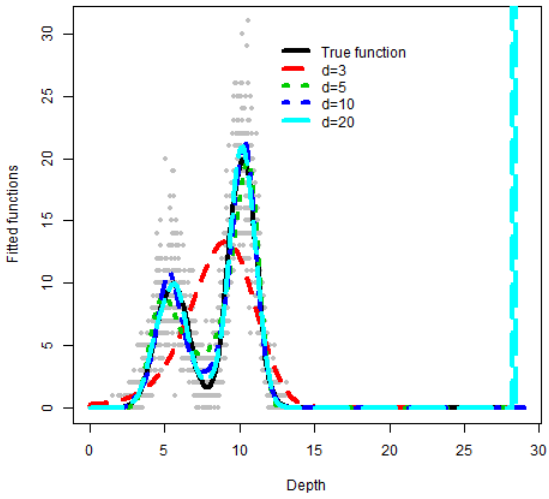


Figure 29: A variety of polynomial-based fitted curves for the count per unit area data with a peak in numbers at both 5 metres and 10 metres.

2.2 Truncated power basis

2.2.1 Model Specification

Truncated power basis functions are more 'locally defined' since each column only has non-zero values between a certain x-value (the j-th 'knot') and the maximum x-value.

In other words, the jth basis function sets all the x-values that are less than the value of the jth knot to zero, subtracts the value of the jth knot from the remaining values and then raises the resulting values to a given power:

$$b_j(x_{it}) = (x_{it} - k_j)_+^d \tag{23}$$

where,

$$\begin{aligned} k_j &= \text{the } j\text{th knot} \\ d &= \text{the power of the basis function} \\ (x_{it} - k_j)_+ &= \text{the truncated power basis function} = \begin{cases} 0, & x_{it} < k_j; \\ (x_{it} - k_j), & x_{it} \geq k_j. \end{cases} \end{aligned}$$

The number of truncated power series basis functions is equal to the number of knots ($j = 1, ..., K$ knots and $K = 4$ in this case).

These basis functions can be included in an additive predictor (like we saw for generalized (non)linear models) on the scale of the link function to give:

$$\eta_{it} = \beta_0 + \sum_{m=1}^{d} \beta_m x_{it}^m + \sum_{j=1}^{K} \beta_{(d+j)} (x_{it} - k_j)_+^d \tag{24}$$

Figure 30 shows the basis functions for $d = 1, ...4$ with $K = 4$ knots (equally spaced using quantiles) in each case.

So, for a TPB with $d = 3$ and two knots ($K = 2$, with k_1 and k_2), then:

$$\eta_{it} = \beta_0 + \beta_1 x_{it} + \beta_2 x_{it}^2 + \beta_3 x_{it}^3 + \beta_4 (x_{it} - k_1)_+^3 + \beta_5 (x_{it} - k_2)_+^3 \tag{25}$$

2.2.2 Model fitting

As for polynomials, TPB models can be fitted using ML/QL. For example, Poisson-based ML was used to return $d = 1, ..., 4$ based truncated power basis models to the count data.

The results on the link and response scales are shown in Figure 32. The plot on the left-hand side represent the fitted functions on the (log) link scale for $d = 1, ..., 4$ and $K = 4$:

$$\hat{\eta}_{it} = \hat{\beta}_0 + \beta_1 x_{it} + \hat{\beta}_2 (x_{it} - k_1)_+ + \hat{\beta}_3 (x_{it} - k_2)_+ + \hat{\beta}_4 (x_{it} - k_3)_+ + \hat{\beta}_5 (x_{it} - k_4)_+ + \log(area_{it}) \quad (26)$$

$$\hat{\eta}_{it} = \hat{\beta}_0 + \beta_1 x_{it} + \beta_2 x_{it}^2 + \hat{\beta}_3 (x_{it} - k_1)_+^2 + \hat{\beta}_4 (x_{it} - k_2)_+^2 + \hat{\beta}_5 (x_{it} - k_3)_+^2 + \hat{\beta}_6 (x_{it} - k_4)_+^2 + \log(area_{it}) \quad (27)$$

$$\hat{\eta}_{it} = \hat{\beta}_0 + \beta_1 x_{it} + \beta_2 x_{it}^2 + \beta_3 x_{it}^3 + \hat{\beta}_4 (x_{it} - k_1)_+^3 + \hat{\beta}_5 (x_{it} - k_2)_+^3 + \hat{\beta}_6 (x_{it} - k_3)_+^3 + \hat{\beta}_7 (x_{it} - k_4)_+^3 + \log(area_{it}) \quad (28)$$

$$\hat{\eta}_{it} = \hat{\beta}_0 + \beta_1 x_{it} + \beta_2 x_{it}^2 + \beta_3 x_{it}^3 + \beta_4 x_{it}^4 + \hat{\beta}_5 (x_{it} - k_1)_+^4 + \hat{\beta}_6 (x_{it} - k_2)_+^4 \hat{\beta}_7 (x_{it} - k_3)_+^4 + \hat{\beta}_8 (x_{it} - k_4)_+^4 + \log(area_{it}) \quad (29)$$

while the plot on the right hand side of contains the fitted values on the response scale for $m = 1, ..., 4$ ($d = 4$), which in each case is $\hat{\mu}_{it} = \exp(\hat{\eta}_{it})$. The values used for the knots (k) are listed in the caption below Figure 32.

2.2.3 Model Selection

In keeping with the polynomial results, the fitted functions on the link scale (regardless of d) are very similar for depth values < 20m. However on the link scale after 20 metres, the curves notably diverge – particularly with $d \geq 3$.

The differences between the fitted curves are easier to see on the response scale (after the exponentiation; right-hand plot of Figure 31) even for the smallest depth values, with the peak in the curve shifting to larger depth values as d increases from 1 to 4.

As for the polynomial results, standard model selection criteria could be used to choose between models with different d values and/or different numbers of knots and their locations (e.g. between models in equations 26 and 29).

2.2.4 Limitations

While truncated basis function expansions are more local acting than polynomials, they also have some undesirable properties:

- The numbers in the basis matrix get very large very quickly, especially as d increases. This can make estimation difficult.
- The columns of the basis matrix can also be prohibitively collinear.
- They are still global functions beyond the (k_j) knot locations (they apply until the maximum value of x).

  For this reason, the very high depth values can still be affected to some extent by the relationship for the very low depth values and vice versa

For instance, the method was subjected to a curve with two peaks (also fitted using polynomials) and this method failed to return either an appropriate drop between the peaks or rise high enough to capture the peaks in either case (Figure 32).

A more locally acting function might be preferred in this and many other cases.

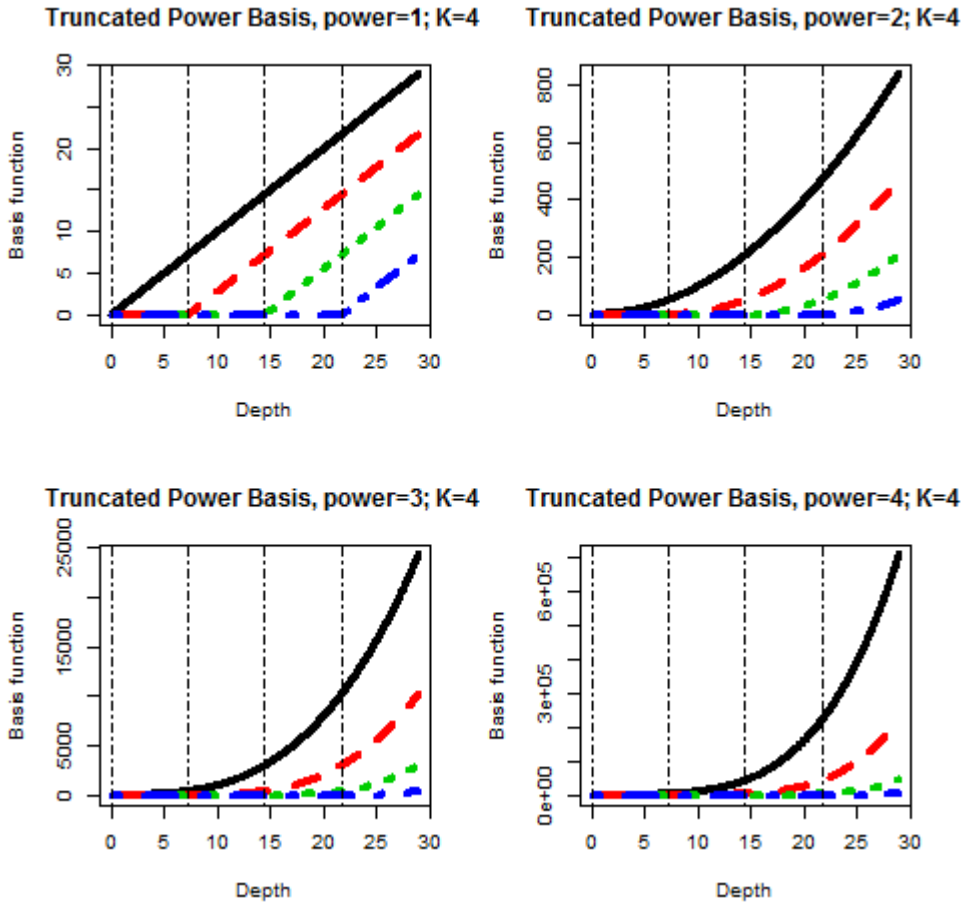


Figure 30: Truncated power bases for depth with $d = 1, ..., 4$ for one set of knots: k=(0.00, 7.25, 14.50, 21.75) shown using black vertical lines. The black solid line is the first column of the $n \times 4$ matrix for depth (associated with the first knot), the red dashed line is the second column (associated with the second knot), the green dotted line is the third column and the blue dotted & dashed line is the fourth column of this same matrix.

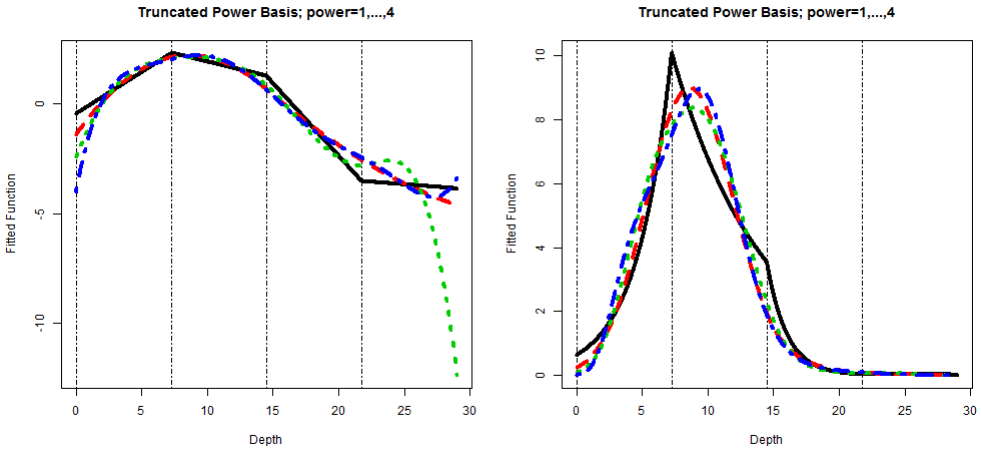


Figure 31: Truncated power based model fits for depth with $d = 1, ..., 4$ for one set of knots: k=(0.00, 7.25, 14.50, 21.75) shown using black vertical lines. The black solid line is the truncated power basis model with $d = 1$ (Equation 26), the red dashed line is the truncated power basis model with $d = 2$ (Equation 27), the green dotted line is the truncated power basis model with $d = 3$ (Equation 28) and the blue dotted & dashed line is the truncated power basis model with $d = 4$ (Equation 29).

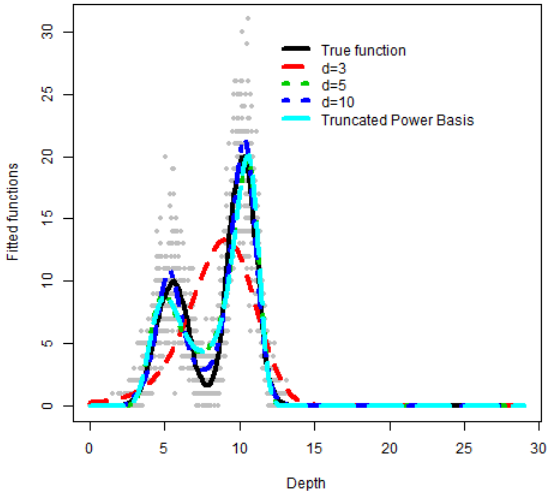


Figure 32: Truncated power based model fits for depth with $d = 4$ for one set of knots: k=(0.00, 7.25, 14.50, 21.75) shown using the light blue long dashed line. The remaining curves are as described in the legend and in Figure 29.

2.3 B-splines

B-splines are piecewise polynomial functions which join smoothly at 'knots'. This basis expansion is 'locally defined' since each column only has non-zero values between certain x-values, determined by the knot locations.
The specification (and flexibility) of a B-spline basis depends on:

1. the degree of the basis function: the higher the degree (d), the more flexible the function can be ($d = 2$ and $d = 3$ are most common)
2. the number of knots: the more knots the more flexible the function can be.
3. the location of these ($k = 1, ..., K$) knots; specifying knots close together results in a more flexible function, in and around these covariate values

Knots are best placed at locations (covariate values):

- with support from the data (e.g. in areas with observations)
- where the biggest changes are likely to occur in the covariate relationship
- so there are *at least* 3 unique covariate values between them (to enable estimation of the curve between knots)

In contrast to the truncated power series, B-splines have some attractive properties:

- estimation issues (encountered with truncated power basis and polynomial expansions) are avoided since the values in the basis matrix are constrained to lie between zero and one
- prohibitively high collinearity is avoided because the columns of the basis only contribute non-zero values for parts of the x-range

2.3.1 Model specification: Basis creation

B-splines are created using a recursive formula starting with $degree = 1$; a piecewise linear function[12] and higher degree B-splines are found using an extension of Equation 30[13].

While the user must choose the interior knots, there are always boundary knots automatically allocated (at the maximum and minimum of the x-range) for B-splines.

- For this reason, there are always more columns than interior knots specified and this number increases with the degree:

Number of basis columns in the basis matrix = Degree + Number of internal knots

- For instance, the quadratic basis for depth (with 3 internal knots) in Figure 33 has five columns (2+3), cubic has six (3+3) and so on.

  B-spline columns only contain values between 0 and 1, and the sum of the unscaled basis function values for any x-value in the predictor range is equal to 1:

For instance, a piecewise linear B-spline basis (degree=1) for depth has the following:

```
> head(bs(depth, knots=k, degree=1))
             1           2 3 4 5
[1,] 1.000000 0.000000000 0 0 0
[2,] 0.995996 0.004004004 0 0 0
[3,] 0.991992 0.008008008 0 0 0
[4,] 0.987988 0.012012012 0 0 0
[5,] 0.983984 0.016016016 0 0 0
[6,] 0.979980 0.020020020 0 0 0
```

[12]
$$B_{j,d=1}(x_{it}) = \frac{(x_{it} - k_j)}{k_{(j+1)} - k_j}_{x \in [k_j, k_{j+1})}(x_{it}) + \frac{(k_{j+2} - x_{it})}{k_{j+2} - k_{j+1}}_{x \in [k_{j+1}, k_{j+2})}(x_{it}) \quad (30)$$

with $j = -1, 2, ..., K$

[13]
$$B_{j,d}(x_{it}) = \frac{(x_{it} - k_j)}{k_{(j+d)} - k_j} B_{i,d-1}(x_{it}) + \frac{(k_{j+d+1} - x_{it})}{k_{(j+d+1)} - k_{j+1}} B_{i+1,d-1}(x_{it}) \quad (31)$$

B-spline basis expansions for $degree = 1, ..., 4$ are shown in Figure 33 with three interior knots at equally spaced quantiles (25th, 50th and 75th percentiles).

Here, the extra complexity possible for the fitted curve as $degree$ increases is apparent from the increased flexibility of (overlapping) columns available for coupling with coefficients in an additive predictor.

For example a basis expansion for depth alone would have an additive predictor for the columns of the basis matrix:

$$\eta_{it} = \beta_0 + \sum_{p=1}^{P} \beta_p B_p(x_{it}) + \log(area_{it}) \tag{32}$$

where P is the number of coefficients in the model associated with the $j = 1, ..., K$ interior knot locations (at covariate values, k_j) and the boundary knots for the B-spline basis.

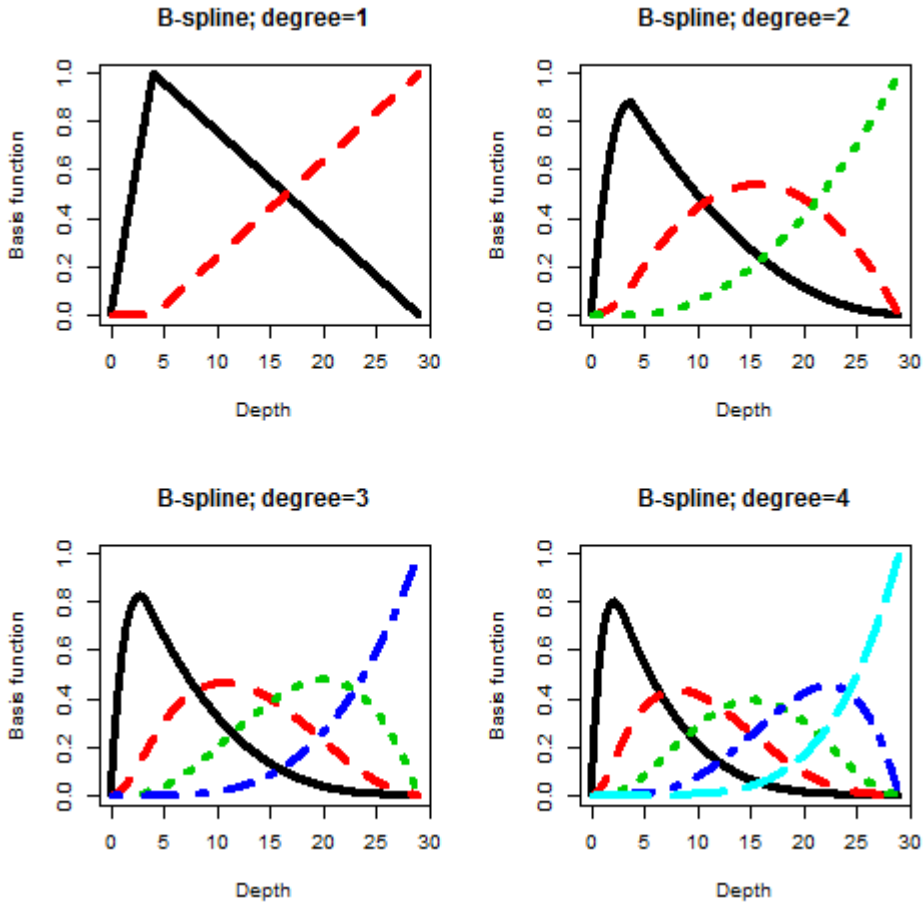


Figure 33: B-spline bases for depth with $d = 1, ..., 4$ for one set of knots: j=(0.00, 7.25, 14.50, 21.75) shown using black vertical lines.

2.4 Model selection & fitting

The flexibility of the fitted B-spline curves can be fully dictated by the number and location of the knots (specified by the user), or jointly determined by these features and a penalty term.

Model flexibility for regression splines (RS) is fully determined using knots (and their locations), while penalised regression splines (PRS) and smoothing splines (SS) also employ a penalty during the fitting process.

2.4.1 Regression splines

Manual knot allocation requires specifying the degree of the basis, the number of knots and the covariate values of the knot locations. The latter two features of RS specification are the most important.

The benefits of manual specification is that covariate relationships which are unevenly smooth can easily be accommodated:

- several knots can be allocated to the covariate range where the flexibility requirements are greatest.
- few knots can be allocated to the covariate range where the relationship is easily approximated by a quadratic/cubic function (depending on the order of the basis used).

While there are clear benefits in some cases, this widens the model selection task. As for the previous basis expansions, standard model selection criteria could be used to choose between models with different d values and knot (number and location) choices.

RS fitting

B-splines can return highly nonlinear curves, but are linear in their parameters (Equation 32) and thus can be fitted using a ML or QL fitting engine.
For example, Poisson-based ML was used to return B-splines based fits for $d = 1, ..., 4$ to the count data. The results on the link and response scales are shown in Figure 34. The plot on the left-hand side represents the fitted functions on the (log) link scale for $d = 1, ..., 4$ and $K = 3$ in all cases.

In keeping with previous results, the fitted functions on the link scale (for $d \geq 2$) are very similar for depth values between 2m and 20m (Figure 34).

However on the link scale after 20 metres, the curves notably diverge – particularly with $d = 3$. The differences between the fitted curves are less pronounced on the response scale (after the exponentiation; right-hand plot of Figure 34).

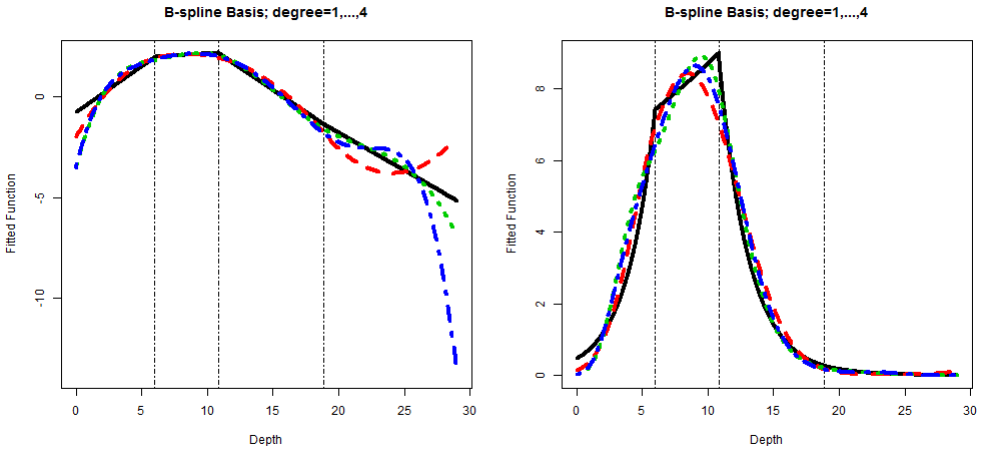


Figure 34: The B-spline basis fits for depth with $d = 1, ..., 4$ for one set of knots placed at the 25th, 50th and 75th percentiles shown using black vertical lines. The black solid line is the fitted model with $d = 1$, the red dashed line is the basis with $d = 2$, the green dotted line is the basis with $d = 3$ and the blue dotted & dashed line is the basis with $d = 4$.

B-spline bases are more locally defined than the expansions discussed previously and thus, can better cope will with abrupt changes in the underlying function as long as the knots are sensibly placed.

- For instance the two-peak curve was better approximated using B-splines when the knots were carefully placed (top left, Figure 35) but was similar to the truncated power basis results when they were arbitrarily placed (top right, Figure 35).
- While a pleasing curve was produced with 17 arbitrary placed knots (bottom left, Figure 35), this model has many more parameters than the 3 knot version (21 coefficients compared to just 7 for the 3 knot version) and so this highlights the need for sensibly placed knots.

The model selection task can be automated (like many model selection approaches) and the spatially adaptive local smoothing algorithm (SALSA **?**) provides one way to automatically select the number and location of knots.

SALSA has been shown to work well especially for functions that require different amounts of flexibility in different parts of the covariate range (i.e. they are unevenly smooth).

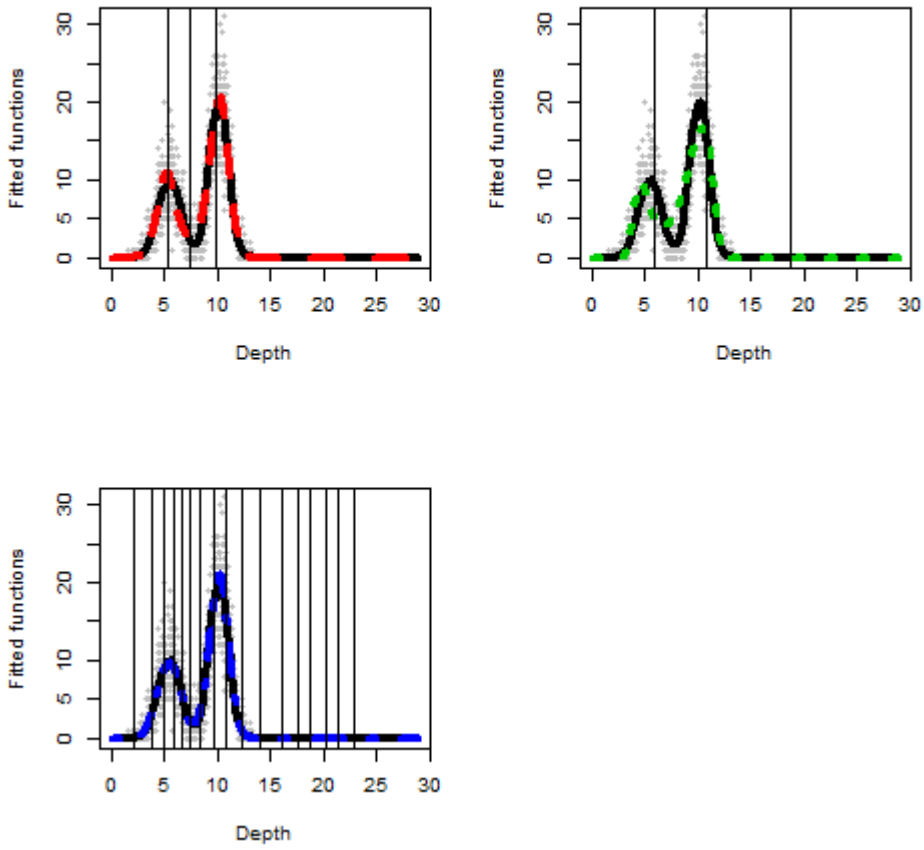


Figure 35: B-spline based model fits for depth with $d = 3$ and variously placed knots (indicated by the vertical lines). The solid black curve is the true function in each case.

2.4.2 Penalized Regression splines

Model specification

In contrast to regression splines, PRS make some arbitrary choices for the number and location of knots and instead rely on a penalty term to control model flexibility.

The flexibility of a curve fitted using penalised regression splines (PRS) is determined using:

- the degree of the basis function ($d = 3$ is typical for B-splines and truncated

power bases)

- a 'large' number of knots (e.g. 10 knots is routine) which are invariably equally spaced using quantiles
- a penalty term to control curve complexity

The penalty term includes:

- a smoothing parameter (λ) which must be chosen by the user or based on some objective fit score.
- the coefficients from the model

In this way very wiggly curves (indicated by large coefficients) are controlled.

Model fitting

The following objective function (for a single smooth term) is minimised to obtain the parameter estimates and associated fitted curve:

$$-log(L) + \lambda \sum_{j=2}^{J-1} (\beta_{j-1} - 2\beta_j + \beta_{j+1})^2 \tag{33}$$

where J is the number of coefficients associated with the smooth term.
Note:

- $\lambda = 0$ results in an interpolating function (i.e. any curvature is tolerated)
- $\lambda = \infty$ results in a straight line (i.e. no curvature is tolerated)

Equation 33 easily extends to more covariates and each covariate attracts its own smoothing parameter.

Model selection

PRS models require one or more smoothing parameters to be chosen, and these are typically chosen using generalized cross-validation (GCV, Equation 34).

$$GCV = \frac{N \times Deviance}{(N - edf)^2} \tag{34}$$

GCV avoids the computationally intensive cross-validation and is a function of the fit to the data (the deviance[14]) based on an associated set of smoothing parameters and model complexity (edf).

[14]

$$Deviance = 2 \sum_{i=1}^{s} \sum_{t=1}^{n_i} y_{it} \log \left(\frac{y_{it}}{\hat{y}_{it}} \right) - (y_{it} - \hat{y}_{it}) \tag{35}$$

Here the y_{it} are the observed values, $\hat{y}_{it}$ are the fitted values for the t-th observation for the i-th transect.

The effective degrees of freedom (edf) quantifies the number of parameters used to fit the model[15]. This is not typically a whole number due to the penalty term employed and each penalized coefficient contributes less than 1 df.

So for a set of λ values the deviance and GCV scores are calculated and the best set of λ values (for multiple covariates) with the best GCV score is chosen.

PRS fitting in R

These can easily be fitted in `R` using the `gam` function in the `mgcv` library (Wood, 2016). This function:

- uses 10 interior knots by default for each covariate
- estimates a different smoothing parameter for each covariate
- produces output assessing the significance of each smooth function (compared to a model without each smooth term)

```
> require(mgcv)
> pen_reg<- gam(count ~ s(depth)+s(X)+s(Y)+s(DistCoast)+phase,
family=quasipoisson, data=data, offset=log(area))
> summary(pen_reg)

Family: quasipoisson
Link function: log

Formula:
count ~ s(depth) + s(X) + s(Y) + s(DistCoast) + phase

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.12434    0.08517  -1.460  0.14432
phaseB       0.09439    0.02584   3.653  0.00026 ***
phaseC      -1.67857    0.32600  -5.149 2.63e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
               edf Ref.df      F p-value
s(depth)     8.413  8.784 300.68  <2e-16 ***
s(X)         8.880  8.995 239.30  <2e-16 ***
s(Y)         8.944  8.999 204.18  <2e-16 ***
s(DistCoast) 7.903  8.493  57.94  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.049   Deviance explained =   30%
GCV score = 14.791  Scale est. = 14.773    n = 31502
```

[15] and is found using the trace of the smoother/hat matrix

In this case:

- between 7.9 and 8.9 effective degrees of freedom were used to fit the continuous covariates (see output and Figure 36)
- the associated p-values suggest that the smooth terms are justified for each covariate.
- average numbers appear to have increased in Phase B (relative to A) but decreased in Phase C (relative to A). We cannot tell from this output how numbers compare from phase B to C.

The estimated smoothing parameters also vary across the covariates and are very small:

```
> pen_reg$sp
    s(depth)          s(X)          s(Y) s(DistCoast)
3.958026e-05 7.048372e-05 1.608575e-05 4.448229e-05
```

Partial plots (Figure 36) are easily created using the following code but when plotted without partial residuals these have little value, and plotting these 'squashes' the range on the y-axis and so are omitted here:

```
> par(mfrow=c(2,2))
> plot(pen_reg, shade=T)
```

while the fitted values for Phase A, B and C (Figure 37) can quickly be evaluated using:

```
> require(fields)
> par(mfrow=c(1,1))
> quilt.plot(data$X[data$phase=="A"], data$Y[data$phase=="A"],
 fitted(pen_reg)[data$phase=="A"], nrow=25, ncol=60,
zlim=range(fitted(pen_reg)))
> quilt.plot(data$X[data$phase=="B"], data$Y[data$phase=="B"],
 fitted(pen_reg)[data$phase=="B"], nrow=25, ncol=60,
zlim=range(fitted(pen_reg)))
> quilt.plot(data$X[data$phase=="C"], data$Y[data$phase=="C"],
 fitted(pen_reg)[data$phase=="C"], nrow=25, ncol=60,
zlim=range(fitted(pen_reg)))
```

PRS models for non-normal data are more difficult to fit with SAS but the GAM procedure employs a similar method (smoothing splines) for this purpose.

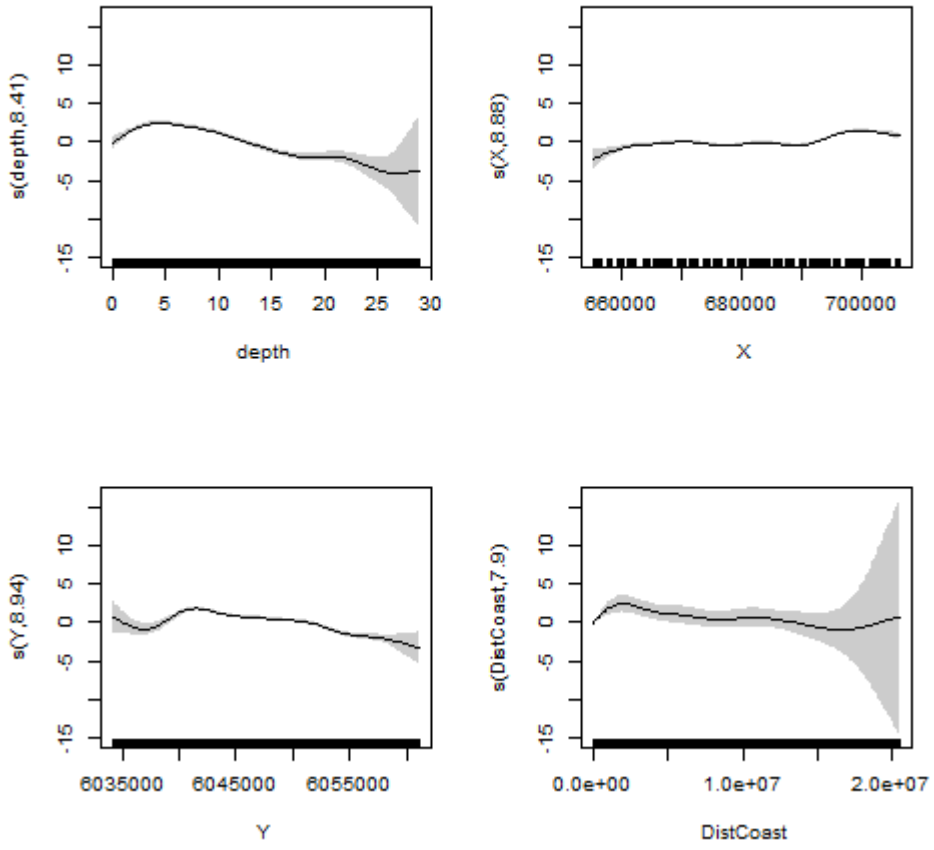


Figure 36: Penalised regression spline fit for the four covariates of interest. The solid black line is the fit on the link scale while the dotted lines are upper and lower 95% confidence intervals.

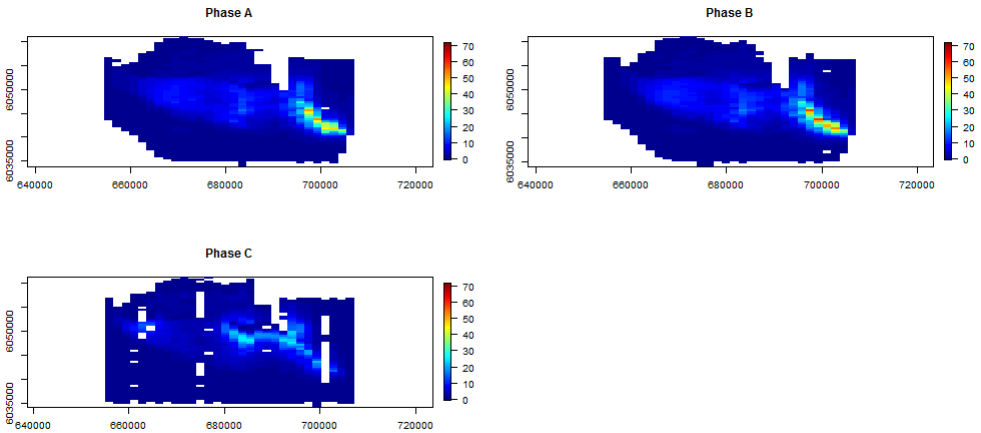


Figure 37: Penalised regression spline-based predictions on the link scale for each Phase shown in a spatial context.

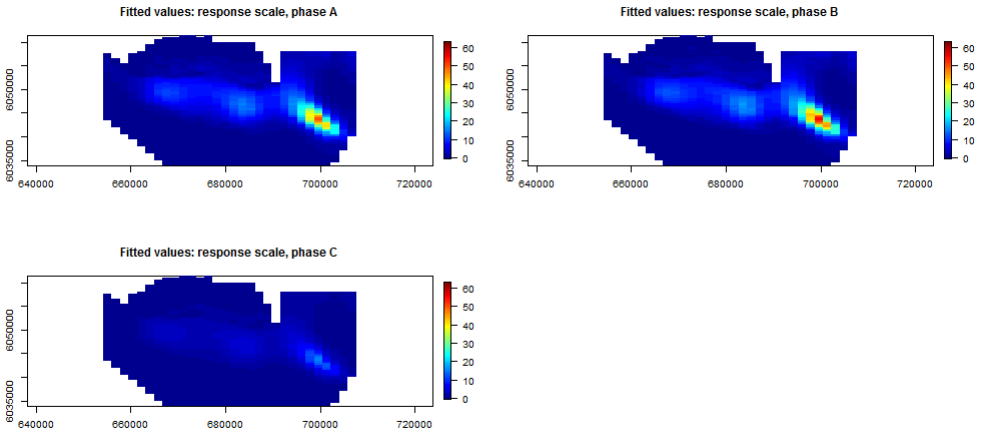


Figure 38: Penalised regression spline-based fitted values for each Phase shown in a spatial context.

Limitations

Penalized splines can be quick and easy to fit, but the fitted curves may be sub-optimal if the flexibility required in the covariate relationship is uneven across the covariate range(s) (e.g. Figure 39).

This is because the covariate-specific global smoothing parameter applies for all values of the covariate of interest. In this case the method does reasonably well, but there are other cases where the effects of the inability of the smoothing paramter to vary across the covariate range are more striking.

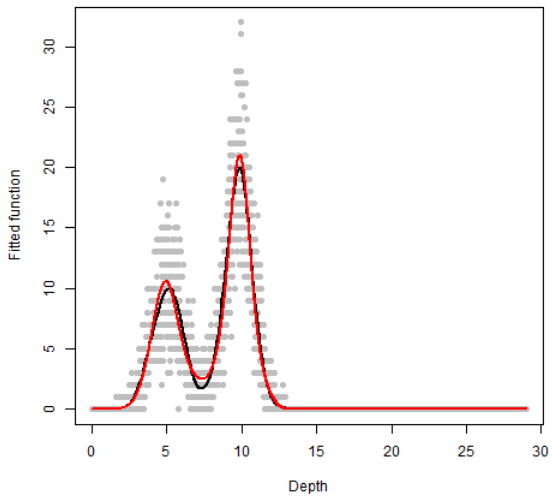


Figure 39: Penalised regression spline-based fitted values for the 'two peak' curve.

2.4.3 Smoothing splines

In keeping with penalized regression splines, smoothing splines avoid the knot selection issue by allocating each unique x-value a knot[16] and control flexibility at the fitting stage using a penalty term.

Model specification

The penalty differs from the PRS penalty however, in that it involves the squared second derivative of each curve. So, for one covariate:

[16] although in practice this is usually limited to 200 knots distributed equally in the covariate range

$$-log(L) + \lambda \int_{x_{min}}^{x_{max}} (f''(x))^2 dx \tag{36}$$

The penalty term involves a smoothing parameter (λ) and the integral of the squared second derivative over the range of the covariate.

The integrated squared second derivative describes the 'roughness' of the curve. For example,

- a function which has a bigger amplitude, is more wiggly, or has a 'rougher' curve. Function A (Figure 40) generates larger first and second derivatives than a function with a smaller amplitude (Function B in Figure 40).
- The slope of functions (and the rate of change of these slopes) is generally steeper/larger for functions with larger amplitudes (Figure 40)
振幅
- The squared second derivative is also larger for functions with larger amplitudes (compare the shaded areas, the integrals, in Figure 40 associated with the squared second derivative in each case).
- A penalty of this sort ensures that positive and negative second derivatives count equally.

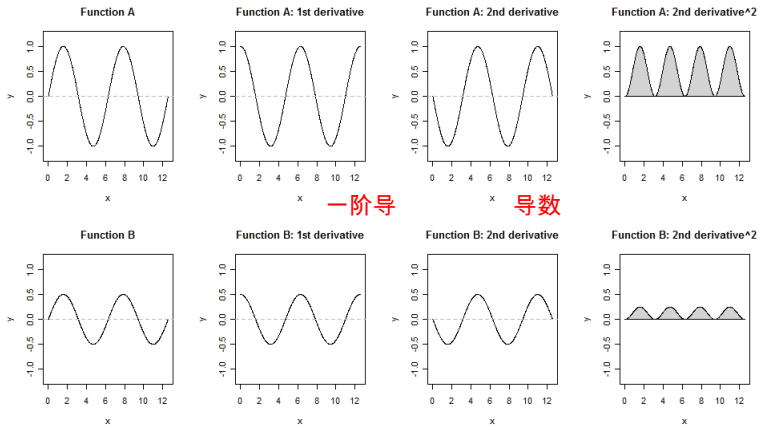


Figure 40: Two functions A and B, and their associated first and second derivatives. The square of the second derivatives is also shown along with the integral (shaded areas).

Model fitting

To fit these models we need to estimate λ. This can be done using generalized cross-validation (Equation 34) as a part of the model fitting routine.

Fitting occurs via a 'local scoring algorithm' which starts all covariates at some initial position and then updates each smooth term (for a given λ) separately while fixing the other covariates in the model at their current position.

This iterative approach is carried out for each covariate in turn, until the fit fails to improve.

Smoothing splines in SAS

Smoothing spline based GAMs can be fitted in `SAS` using the GAM procedure. We can either:

- set the number of parameters to estimate for each covariate (using `df=4` in this case), or
- use `method=GCV` in the options part of the `model` statement to estimate each smoothing parameter.

```
#specifying df=4;
proc gam data=nysted;
class phase;
model count=param(phase) spline(depth,df=4) /dist = poisson;
output out=estimate p;
run;
```

A model containing only phase and depth results in the output in Figures 41 and 42. A model with more terms did not converge.

in SAS the last level (C) is the baseline

We can also ask SAS to use GCV to choose each smoothing parameter:

```
#using GCV to choose the smoothing parameter;
proc gam data=nysted;
class phase;
model count=param(phase) spline(depth) spline(X) spline(Y)
spline(DistCoast)/dist = poisson method=GCV;
output out=estimate p;
run;
```

Smoothing splines in R

This can also be done in `R` using the gam library (you must detach `mgcv` and restart `R` if you want to use the gam function from the gam library):

detach(package:mgcv)
search()

```
> require(gam)
> summary(ss_reg)

Call: gam(formula = count ~ s(depth) + s(X) + s(Y) + s(DistCoast) +
    phase, family = quasipoisson, data = data)
```

```
Deviance Residuals:
     Min       1Q   Median       3Q      Max
-10.6374  -2.4280  -1.0539  -0.3087 101.5494

(Dispersion Parameter for quasipoisson family taken to be 82.9459)

    Null Deviance: 666428.3 on 31501 degrees of freedom
Residual Deviance: 469632.9 on 31422.07 degrees of freedom
AIC: NA

Number of Local Scoring Iterations: 30
```

This output shows results for the linear component for each term (and associated tests of significance):

```
Anova for Parametric Effects
               Df  Sum Sq Mean Sq  F value    Pr(>F)
s(depth)        1     392     392   4.7296   0.02965 *
s(X)            1   35249   35249 424.9623 < 2.2e-16 ***
s(Y)            1   21267   21267 256.3926 < 2.2e-16 ***
s(DistCoast)    1    1320    1320  15.9145 6.642e-05 ***
phase           2     649     324   3.9097   0.02006 *
Residuals   31422 2606332      83
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

a linear term rather than a non-linear term

and results for the nonlinear component for each term (and associated tests of significance):

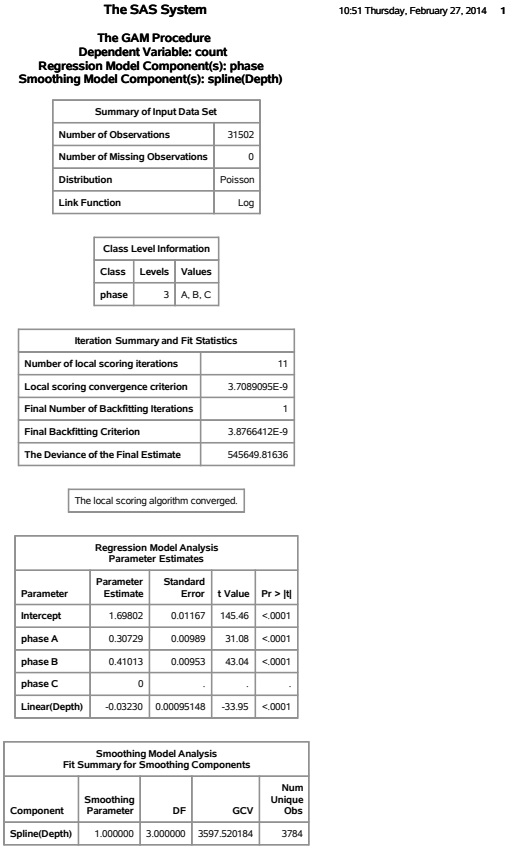
```
Anova for Nonparametric Effects
            Npar Df Npar F     Pr(F)
(Intercept)
s(depth)         3.0 87.916 < 2.2e-16 ***
s(X)             3.0 46.882 < 2.2e-16 ***
s(Y)             3.0 23.609 2.887e-15 ***
s(DistCoast)    63.9  0.995    0.4881
phase
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

> par(mfrow=c(3,2))
> plot(ss_reg, se=T)
```

The partial plots (Figure 43) show a large amount of flexibility is allocated to the DistCoast covariate ($df = 63.9$) and this is not significantly nonlinear ($p = 0.4881$). This term might best be re-fitted as linear on the link scale.

There is also compelling evidence for differences across phases ($p = 0.02006$). Phase A and B appear to have significantly more animals on average than Phase C (the baseline in SAS; Figure 41).

There are no major distributional shifts in the fitted numbers across phases under the model (Figure 44).



The SAS System 10:51 Thursday, February 27, 2014 1

The GAM Procedure
Dependent Variable: count
Regression Model Component(s): phase
Smoothing Model Component(s): spline(Depth)

Summary of Input Data Set	
Number of Observations	31502
Number of Missing Observations	0
Distribution	Poisson
Link Function	Log

Class Level Information		
Class	Levels	Values
phase	3	A, B, C

Iteration Summary and Fit Statistics	
Number of local scoring iterations	11
Local scoring convergence criterion	3.7089095E-9
Final Number of Backfitting Iterations	1
Final Backfitting Criterion	3.8766412E-9
The Deviance of the Final Estimate	545649.81636

The local scoring algorithm converged.

Regression Model Analysis Parameter Estimates				
Parameter	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1.69802	0.01167	145.46	<.0001
phase A	0.30729	0.00989	31.08	<.0001
phase B	0.41013	0.00953	43.04	<.0001
phase C	0	.	.	.
Linear(Depth)	-0.03230	0.00095148	-33.95	<.0001

Smoothing Model Analysis Fit Summary for Smoothing Components				
Component	Smoothing Parameter	DF	GCV	Num Unique Obs
Spline(Depth)	1.000000	3.000000	3597.520184	3784

Figure 41: Smoothing spline results using the GAM procedure in SAS for a model containing only phase and depth.

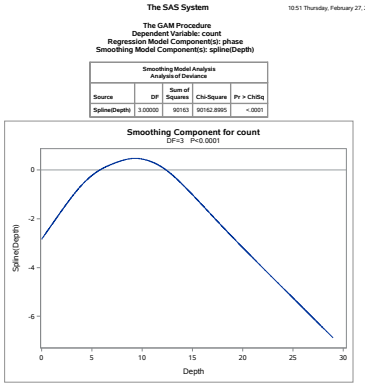


Figure 42: Smoothing spline results using the GAM procedure in SAS for a model containing only phase and depth. Including additional covariates fails to converge.

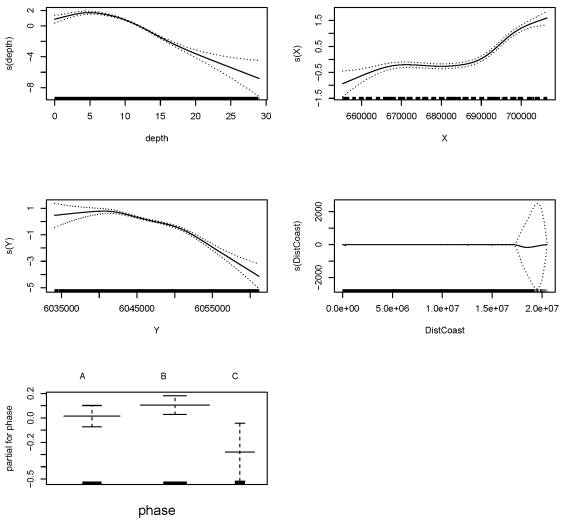
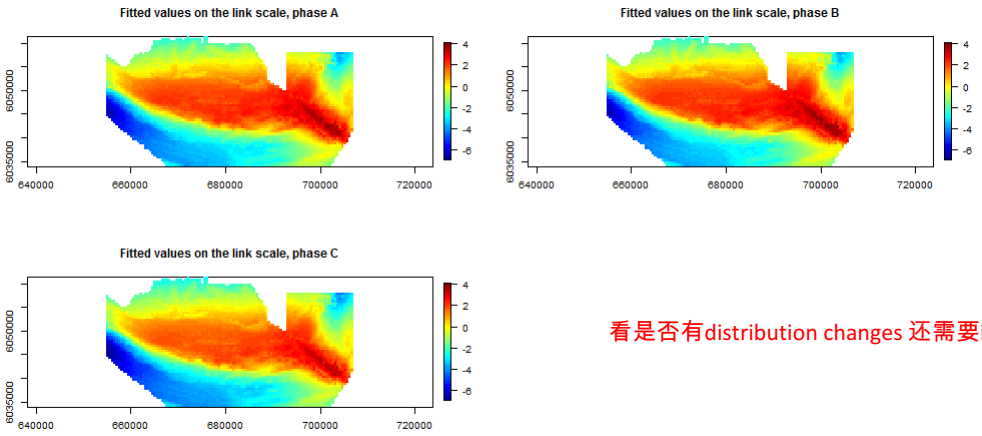


Figure 43: Smoothing spline results in R. Note including `DistCoast` results in a smooth term with $df = 63.8$, even though it should only allocate $df = 3$ to the smooth component.



看是否有distribution changes 还需要interaction

Figure 44: Smoothing spline-based fitted values for each Phase A shown in a spatial context.

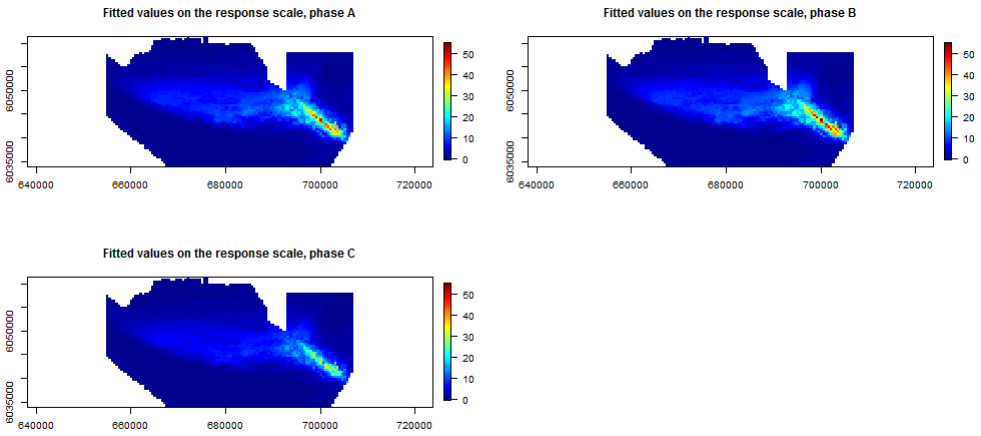


Figure 45: Smoothing spline-based fitted values for each Phase shown in a spatial context.

Limitations

Integrating the squared second derivative can be a large computational burden (especially with many covariates) and models don't often converge. smooth will take a long time

For this reason, penalized splines (which use a penalty which is easier to calculate) or regression splines (fitted without penalty) might return results when smoothing splines do not.

As for penalized splines, the fitted curves may also be sub-optimal if the flexibility required in the covariate relationship is uneven across the covariate range(s) since the smoothing parameter applies for all values of the covariate.

2.4.4 Regression splines with automated model selection via SALSA

Model specification

Instead of side-stepping knot selection and using a global penalty, we can specify the model manually and choose both the number of knots and their locations.

Targeting flexibility

The benefits of doing this is that covariate relationships which are unevenly smooth can easily be accommodated.

For example, several knots can be allocated to the covariate range where the flexibility requirements are greatest and fewer knots where the relationship is easily approximated by a quadratic/cubic function (depending on the order of the basis used).

While there are clear benefits in some cases, this widens the model selection task. As for the previous basis expansions, standard model selection criteria can be used to choose between models with different degrees for the basis and the number and location of knots.

Automated model selection via SALSA

The model selection task can be automated (like many model selection approaches) and the spatially adaptive local smoothing algorithm (SALSA; Walker et al., 2011) provides one way to automatically select the number and location of knots.

SALSA has been shown to work well especially for functions that require different amounts of flexibility in different parts of the covariate range (i.e. they are unevenly smooth).

Model selection: SALSA overview

SALSA is simple in concept and 'chooses' the number and location of knots starting from evenly spaced knot positions (using quantiles) and some given starting number of knots.

SALSA evaluates the potential benefits of making local knot moves to adjacent positions (based on objective fit criteria) and to locations which are poorly fitted by the provisional model (indicated by the maximum Pearson residual).

SALSA also tries to improve model fit in a more radical way by switching each current knot location to the point returning the maximum Pearson residual.

In addition to this, the model fit obtained by adding a knot at the point returning the maximum (Pearson) residual is compared with the working model and models obtained by dropping each knot one-by-one.

We'll now look in closer detail at the specification of these models using the `MRSea` package in R (Scott-Hayward et al., 2017).

Candidate knot locations & knot numbers

Up to 200 unique covariate values are candidate knot locations[17] and the user sets the range of the number of knots considered (`minKnots_1d` & `maxKnots_1d`) however this can be a wide range.

The starting number of knots is also specified: `startKnots_1d`.

We can also specify a gap between knots (`gap`) or set this to be zero and enforce no gap at all and like any spline-based analysis, the degree of the basis function (`degree`) must also be chosen.

Objective fit criteria

To choose between different spline-based models, we can use standard objective fit criteria (`fitnessMeasure` in SALSA). At the time of writing, AIC, AICc, BIC, QAIC, QBIC, QAICc and AIC_h are currently implemented.

K-fold cross-validation based on omitting correlated blocks (transects in this case) is also used to choose between models with and without each covariate.

Helpfully, this process compares models with each covariate as a smoother-based term, a linear term and omitting each term altogether.

Model fitting: Which predictors are useful?

```
#load the package (version 1.0.01)
> require(MRSea)
#create the variable 'response' for SALSA
> windfarm$response<- windfarm$count
> windfarm$panels<-as.numeric(windfarm$unique.transect.label)
> windfarm$foldid<-getCVids(windfarm, 10, 'panels')

#create the variable 'response' for SALSA
> windfarm$response<- windfarm$count
#set initial model without the spline-based terms
initialModel<- glm(response ~ phase  + as.factor(month) +
offset(log(area)),
family='quasipoisson', data=windfarm)

#set some input info for SALSA
> factorlist<-c('phase','month')
> salsa1dlist<-list(fitnessMeasure='QAIC', minKnots_1d = c(1,1,1,1),
    maxKnots_1d=c(4,4,4,4), startKnots_1d = c(1,1,1,1), degree=c(2,2,2,2),
    maxIterations=100, gaps=c(0,0,0,0))
```

[17] these are 'space-filled' if the number of unique points exceeds 200

```
#run SALSA
> salsa1dout<-runSALSA1D(initialModel, salsa1dlist, varlist, factorlist,
varlist_cyclicSplines = NULL, splineParams = NULL,
datain=windfarm, suppress.printout = FALSE, removal=FALSE,
panelid=NULL)

# store best model
bestModel1D<-salsa1dout$bestModel

#summary output
> anova(bestModel1D, test='F')
Analysis of Deviance Table (Type II tests)
Marginal Testing

Response: response
Error estimate based on Pearson residuals

                      SS    Df        F    Pr(>F)
phase               1877     2   8.3688 0.0002325 ***
as.factor(month)   12271     3  36.4671 < 2.2e-16 ***
s(depth)           46463     3 138.0746 < 2.2e-16 ***
s(X)               32735     6  48.6401 < 2.2e-16 ***
s(Y)               31145     4  69.4164 < 2.2e-16 ***
s(DistCoast)        6494     3  19.2998 1.695e-12 ***
Residuals        3531039 31480
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Model fitting: Checking for redistribution

```
> SALSAMod<- update(salsa1dOutput$bestModel, . ~ .
+ bs(X, knots = splineParams[[2]]$knots, degree = splineParams[[2]]$degree,
Boundary.knots = splineParams[[2]]$bd):phase
+ bs(Y, knots = splineParams[[3]]$knots, degree = splineParams[[3]]$degree,
Boundary.knots = splineParams[[3]]$bd):phase)

#need to use this function in this case - duplicate names are currently an issue with the current package.
> Anova(SALSAMod, test="F")
Analysis of Deviance Table (Type II tests)

Response: response
Error estimate based on Pearson residuals

                                                                                                                SS    Df        F    Pr(>F)
phase                                                                                                         1877     2   8.5932 0.0001858 ***
as.factor(month)                                                                                             12284     3  37.4838 < 2.2e-16 ***
bs(depth, knots = splineParams[[2]]$knots, degree = splineParams[[2]]$degree, Boundary.knots = splineParams[[2]]$bd)      45022     3 137.3817 < 2.2e-16 ***
bs(X, knots = splineParams[[3]]$knots, degree = splineParams[[3]]$degree, Boundary.knots = splineParams[[3]]$bd)          32465     6  49.5322 < 2.2e-16 ***
bs(Y, knots = splineParams[[4]]$knots, degree = splineParams[[4]]$degree, Boundary.knots = splineParams[[4]]$bd)          30736     4  70.3423 < 2.2e-16 ***
bs(DistCoast, knots = splineParams[[5]]$knots, degree = splineParams[[5]]$degree, Boundary.knots = splineParams[[5]]$bd)   1049     3   3.2023 0.0222328 *
phase:bs(X, knots = splineParams[[3]]$knots, degree = splineParams[[3]]$degree, Boundary.knots = splineParams[[3]]$bd)     3615    12   2.7579 0.0009395 ***
phase:bs(Y, knots = splineParams[[4]]$knots, degree = splineParams[[4]]$degree, Boundary.knots = splineParams[[4]]$bd)     3352     8   3.8358 0.0001605 ***
Residuals                                                                                                  3436641 31460
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

At this point we can see compelling evidence for redistribution (of a particular kind) in the X and Y co-ordinates, and compelling evidence for both depth and distance from coast relationships.

2.5 Two dimensional smoothers

Models fitted to the data so far, assume that the relationship between the x-coordinate and average animal numbers holds (i.e. is the same) regardless of the value of the y-coordinate.

This idea is often unrealistic and the approach is unable to identify any local surface changes, such as peaks or depressions in parts of the surface (e.g. locally popular areas with the animals: 'hot-spots').

For this reason, it is often wise to permit the smooth relationship between one covariate and the response to vary with another covariate in a flexible way - include interaction terms in the model.

This can be done using two dimensional smoothers and is a more realistic way of including spatial information in a model.

The most commonly implemented two-dimensional smoother is the thin plate spline smoother (TPS) which employs a global smoothing parameter that applies across the two dimensional surface.

In keeping with one-dimensional smoothers, thin-plate splines are based on a set of knots which are used to make basis functions and form columns of the basis matrix.

This basis matrix is then fitted using ML/QL either with or without a penalty (and the associated smoothing parameter).

2.5.1 Thin plate splines: TPS

Model specification
In general, the process is as follows:

- A set of knots are chosen from a set of spatial co-ordinates
- The euclidean/straight line distances between each set of spatial co-ordinates and these knot locations is found; this creates one column of distances (with N rows) for each knot (these are the $d_{j,it}$)

For illustration consider one randomly chosen knot location (at observation number 4448) and two randomly chosen spatial coordinates with observation numbers: 19052 and 29278.

```
> ps
[1]  4448 19052 29278
> cbind(data$X[ps], data$Y[ps])
         [,1]    [,2]
[1,] 680104.2 6044690
[2,] 659969.8 6041891
[3,] 662071.7 6036317
```

To find the distances between the knot and the spatial location for row 19052 (Figure 46) we use:

$$= \sqrt{(x_{4448} - x_{19052})^2 + (y_{4448} - y_{19052})^2} \quad (37)$$

$$= \sqrt{(680104.2 - 659969.8)^2 + (6044690 - 6041891)^2} \quad (38)$$

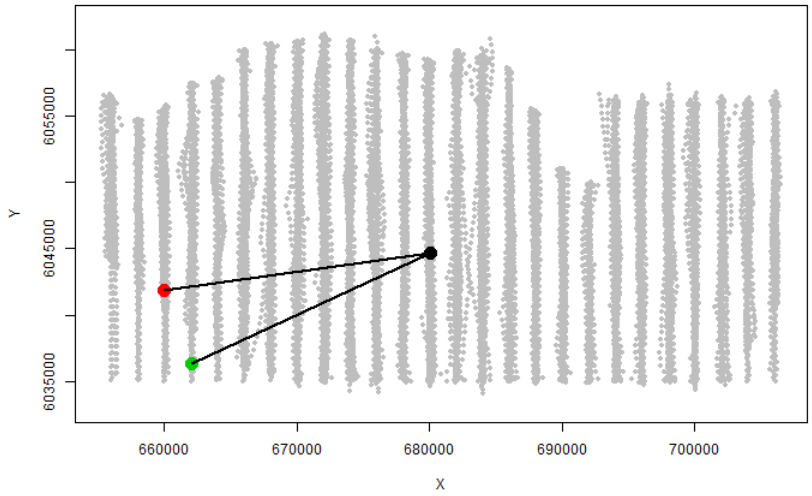


Figure 46: Euclidean distance between a randomly chosen knot and two locations.

- These straight-line distances are then used to create columns of the basis matrix $(B_{j,it})$ – one column per knot:

$$B_{j,it} = d_{j,it}^2 \log d_{j,it} \tag{39}$$

The values inside these 'radial' basis functions increase in value with distance from each knot location (Figure 47).

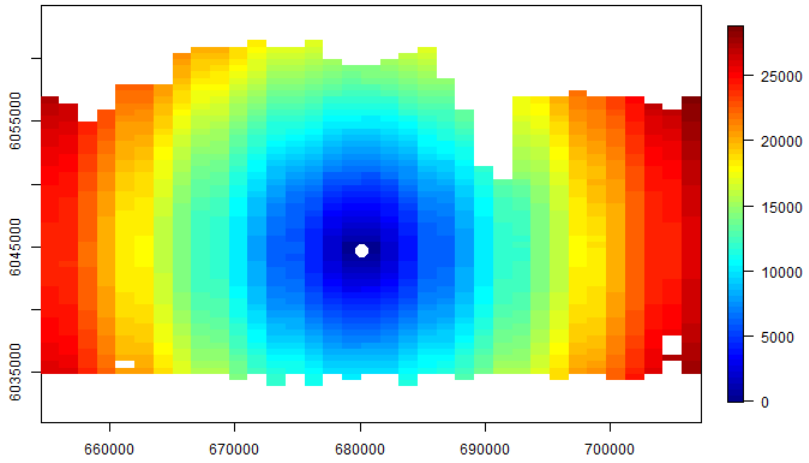


Figure 47: Thin plate spline-based values for a single knot (indicatd by the white dot).

- These columns of the basis matrix for the j-th knot, i-th transect and t-th observation ($B_{j,it}$) are used in the linear predictor alongside the linear terms for the spatial co-ordinates (called $XPos$ and $YPos$ in Equation 40) and coefficients attached to any other covariates.

$$\eta_{j,it} = \beta_0 + \beta_1 XPos_{it} + \beta_2 YPos_{it} + \sum_{j=1}^{K} \beta_{2+j} B_{j,it} \tag{40}$$

- Estimation can proceed unpenalised or with a penalty term.

Model Fitting

积分的

In SAS, the penalty term is the integral of the squared second derivative and the user can specify the degrees of freedom to use for each covariate (which determines λ(s)) or GCV can be used to choose the smoothing parameter(s).
These models can be fitted using:

```
proc gam data=nysted;
class phase;
model count=param(phase) spline(depth,df=4) spline2(X,Y,df=10)
spline(DistCoast, df=4)/dist = poisson offset=larea;
output out=estimate p;
run;
```

log of area

In R, the penalty term is related to the size of the coefficients and the user can specify the degrees of freedom to use for each covariate (which determines λ(s)) or GCV can be used to choose the smoothing parameter(s).

one dimensional smooth of depth

```
pen_reg_2D<- gam(count ~ s(depth)+s(X,Y)+s(DistCoast)+
phase, family=quasipoisson, data=data, offset=log(area))
```

Model selection

Non-nested models[18] can be chosen by comparing relevant fit scores (AIC/BIC) but this is typically a manual process; automatic model selection is unable as yet.

The dredge function in R provides an all subsets approach but does not consider linear versions for any of the terms (instead of smooth terms in the full model).

if you have a smooth term the thing you do is to fit a smooth term in a linear model.
重新拟合

Results

The results of the model with and without a phase-related interaction are as follows:

```
> summary(pen_reg_2D)

Family: quasipoisson
Link function: log

Formula:
count ~ s(depth) + s(X, Y) + s(DistCoast) + phase
```

[18] models which are not a special case of each other

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.28410    0.08741  -3.250 0.001154 **
phaseB       0.09568    0.02563   3.733 0.000189 ***
phaseC      -1.21835    0.31767  -3.835 0.000126 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1


Approximate significance of smooth terms:
                edf Ref.df      F p-value
s(depth)      8.645  8.917  33.03  <2e-16 ***
s(X,Y)       28.726 28.984 154.34  <2e-16 ***
s(DistCoast)  7.647  8.279  54.00  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.0549   Deviance explained = 31.2%
GCV score = 14.559  Scale est. = 14.537    n = 31502


> summary(pen_reg_2DwithInt)

Family: quasipoisson
Link function: log

Formula:
count ~ s(depth) + s(X, Y, by = phase) + s(DistCoast) + phase

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.3834     0.6432  -2.151   0.0315 *
phaseB        0.7879     0.4317   1.825   0.0680 .
phaseC        1.4398     2.7531   0.523   0.6010
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
                 edf Ref.df      F  p-value
s(depth)       8.807  8.972  3.834 7.58e-05 ***
s(X,Y):phaseA 28.720 28.987  7.701  < 2e-16 ***
s(X,Y):phaseB 28.669 28.977 12.360  < 2e-16 ***
s(X,Y):phaseC 26.149 28.059  2.155 0.000367 ***
s(DistCoast)   7.812  8.424  1.756 0.076648 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.0614   Deviance explained = 33.9%
GCV = 14.032  Scale est. = 116.16    n = 31502
```

We will interpret the results for the model with the phase-related interaction term, since the GCV score is lower. Based on the model without the interaction, average numbers:

- have increased from Phase A to Phase B

- have decreased from Phase A to Phase C

Based on the model with the interaction, assessment of the outputs shows:

- that whilst some non-linearity has been estimated (edf=8.9) average numbers appear to change little with depth, with the highest uncertainty estimated at depths greater than 25m (Figure 48).

- that average numbers appear be higher closer to the coast and decline with distance from coast (Figure 49). This relationship is highly uncertain however and nonlinearities (edf = 7.8) appear unjustified. This term could be refitted as linear and/or omitted from the model (whichever is determined 'best').

- The spatial distribution appears to be similar in phases A and B but with an increase in numbers in phase B. In phase C there are very large peaks in numbers in the east and west (Figure 50).

- All relationships except distance from coast appear to be significantly nonlinear (evidenced by very small p-values for all terms).

- About 34% of the deviance is explained by the model.

- There is some evidence of wind farm avoidance under the interaction based model for the first windfarm (animals tend to move away from the footprint in phase B). There is also evidence of attraction in phase C; there are increased numbers in and around the windfarms in phase C.

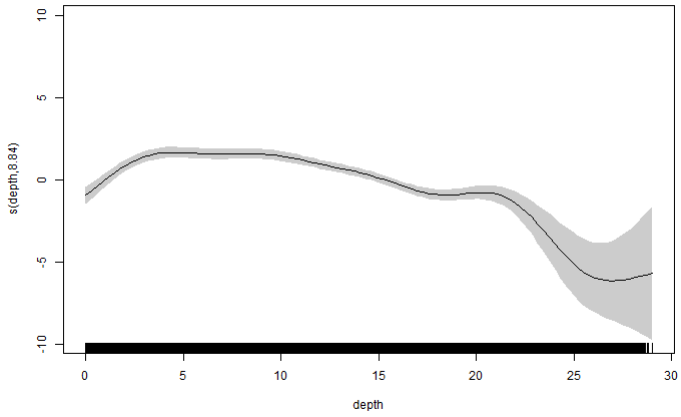


Figure 48: An illustration of the depth relationship for the TPS based model. The grey shading is the 95% confidence interval.

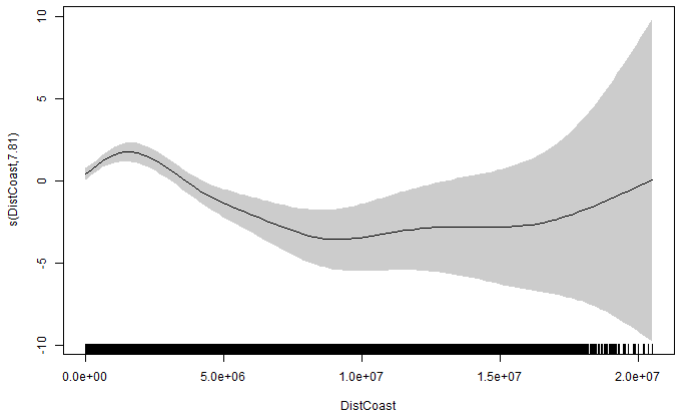


Figure 49: An illustration of the DistCoast relationship for the TPS based model. The grey shading is the 95% confidence interval.

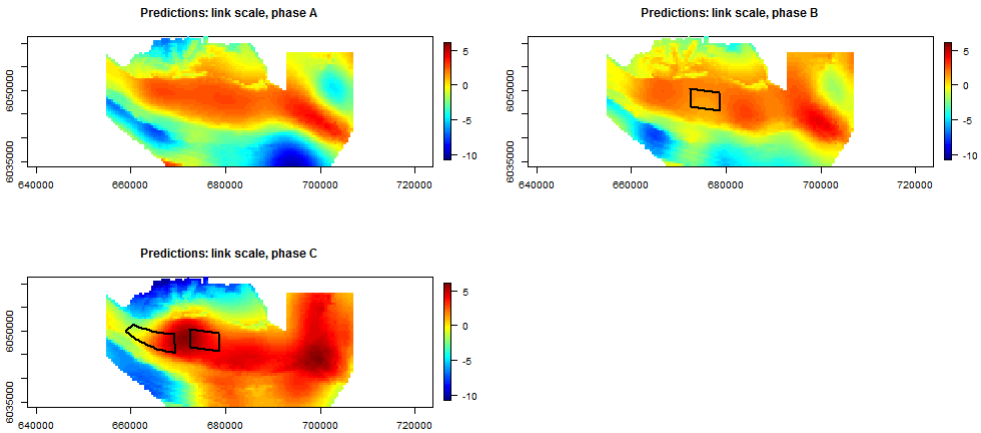
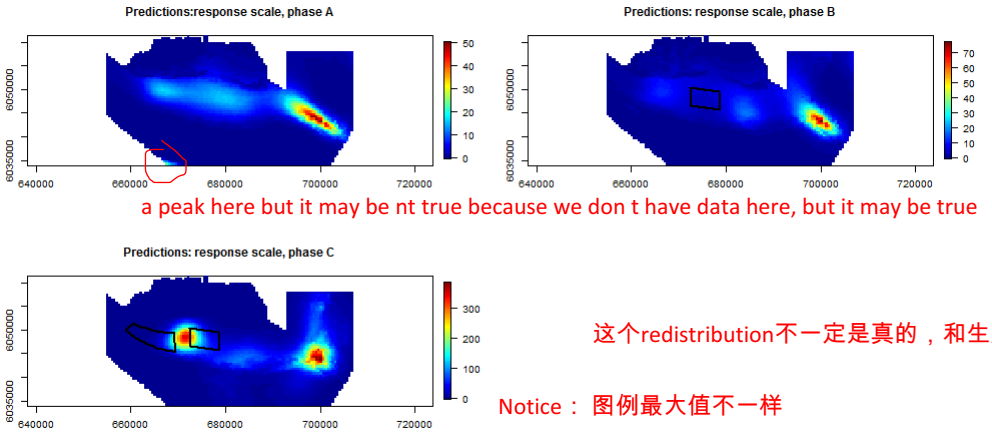


Figure 50: Fitted values on the link scale for the TPS based model with an interaction for each phase.



a peak here but it may be nt true because we don t have data here, but it may be true

这个redistribution不一定是真的，和生成模型的数据不符

Notice： 图例最大值不一样

如果能每个phase画出lower and upper CI 的图
或者动画显示所有可能

Figure 51: Fitted values for the TPS based model with an interaction for each phase.

- For instance, in this (and many) cases the spatial distribution of animals across the survey area is patchy and the distribution of animals is highly uneven across the site. 不调和的
- In some cases, survey effort can also be patchy across the area with some areas

frequently visited (e.g. on the transect lines) and some areas un-surveyed (e.g. between transects).

This uneveness in both the presence of data and their response values (when they are seen) can cause grief for methods which allocate flexibility evenly/systematically across the site.

This systematic allocation can mean interesting surface features are missed (and left unmodelled) and/or parts of the area poorly estimated due to data paucity.

欧几里得

These euclidean based smoothers can also return silly answers when there are internal exclusion zones - areas where, in this case, the animals cannot be found (e.g. marine mammals and islands).

This can make the straight line distances between knots and points an unsuitable metric (Figure 52) and in those cases the genuine distances across these areas are much bigger than the euclidean distances would suggest.

These basis functions are also global (e.g. Figure 47) and their increasing values with distance from the knots mean you can see wild predictions at the boundaries.

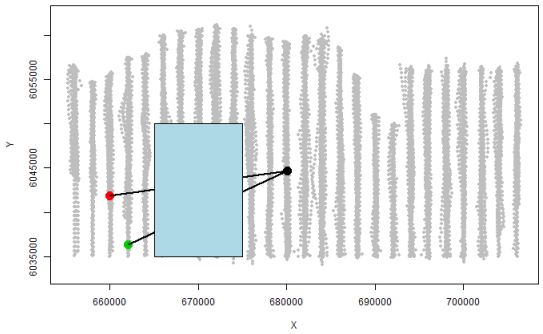


Figure 52: Euclidean distance between a randomly chosen knot and two locations and an interior exclusion zone.

In general we want to use two-dimensional smoothers which accommodate:

- patterns which apply across large parts of the survey area (i.e. global patterns)
- patterns which only apply in parts of the survey area (i.e. local patterns)
- internal exclusion zones

2.5.2 Complex Region Spatial Smoother: CReSS

Instead of using a penalty term in the estimation process, CReSS-based smoothers rely on sensible allocation of knots and their locations and the coefficients are estimated without penalty.

Model specification

The model specification process is similar: knots are chosen, basis functions created (Equation 41) and included in the linear predictor alongside other covariates (either as linear or smoother-based terms; Equation 42).

The CReSS basis function is very different from the TPS basis and, in particular the values in the columns decline with distance from each knot (Figure 53).

There is also a range parameter, r, to be chosen to help specify the effective range of the bases (and this can vary across knots).

$$B_{j,it} = \exp^{(-d_{j,it}r^2)} \tag{41}$$

$$\eta_{j,it} = \beta_0 + \beta_1 \text{Depth}_{it} + \beta_2 \text{Distcoast}_{it} + \sum_{j=1}^{K} \beta_{2+j} B_{j,it} \tag{42}$$

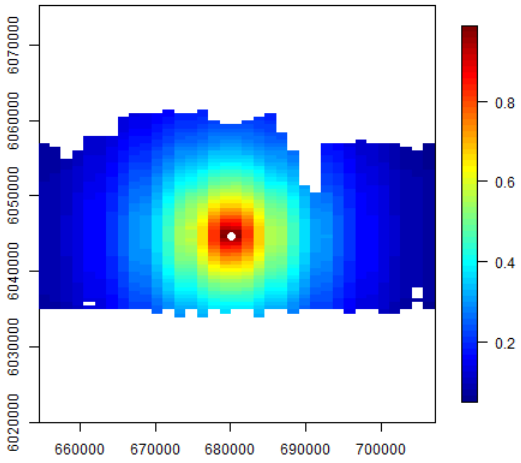


Figure 53: An illustration of one column from a CReSS basis matrix.

CV时间会更久

Model selection

Model selection for the knots, their locations and the range parameter can be automated and governed by your favourite objective fit criteria.

The spatially adaptive local smoothing algorithm (SALSA) was developed for spatially adaptive smoothing in one dimension (for regression splines) but has been extended to help with two dimensional smoother specification.

A SALSA2D overview

SALSA2D works very simply by choosing the number and location of knots starting from a set of available knots and some given starting number of knots.

SALSA2D then evaluates the potential benefits of making local knot moves (based on objective fit criteria and the 5 nearest 'legal' knots) to locations which are poorly fitted by the provisional model (indicated by the maximum Pearson residual).

SALSA2D also tries to improve model fit in a more radical way by switching each current knot location to the point returning the maximum Pearson residual.

In addition to this, the model fit obtained by adding a knot at the point returning the maximum (Pearson) residual is compared with the working model and models obtained by dropping each knot one-by-one.

Candidate knot locations

It is preferable to have each basis function centred about an observed spatial location for maximum support from the data.

To reduce computation time for very large data sets, rather than allowing a knot location at every unique data location, we choose 300 candidate knots from the full set using a 'space-filling' algorithm (Figure 54). This approach ensures maximum coverage for a given number of knots.

We typically space-fill 300 knots but this number can naturally be varied depending on the computation time available. We have found this to be adequate even for very large spatial areas.

Candidate knot numbers

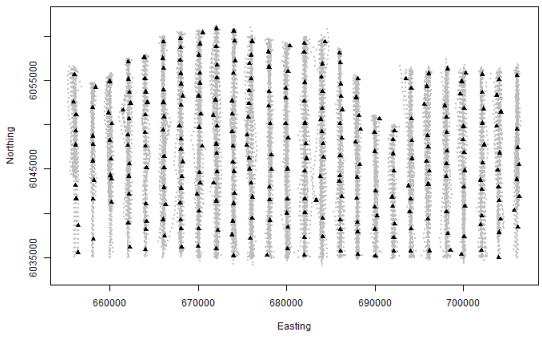
While the user sets the range of the number of knots trialled[19], the algorithm tends to 'converge' on a similar number of knots for the final model (regardless of the starting number of knots).

However, in some cases the analysis might benefit from varying the number of starting knots.

If a variety of starting knot numbers is desired, multiple R sessions can typically be run in parallel and so this does not noticeably add to the analysis time.

From there, objective fit criteria can easily be used to choose between the different final models (which start with different numbers of knots).

[19] `startKnots`, `minKnots` & `maxKnots` in SALSA2D)



can only move to the place with triangle

Figure 54: Knotgrid for the windfarm model. Grey dots are data locations and black triangles are 300 candidate knot locations

The 'effective range' of each knot

A range of values for the effective range of each knot (the r_k's) are also trialled as part of the model selection process.

Further, since we have found modelling results to be relatively insensitive to the choice of r_k, a single value (the median of the candidate values) is used while the knot locations are being chosen.

The sequence of candidate values trialled by SALSA2D ensures the effective range of the basis includes at least one point and possibly all points.

For this reason, the distance between each point and the candidate knots is found and a sequence of 8 values based on the minimum to the maximum of these distances is found.

From there, model fit (based on objective fit criteria) for a set of chosen knots is used to choose the best r_k-value for each knot location.

Objective fit criteria

To choose the location and/or number of knots we can use standard objective fit criteria (`fitnessMeasure` in SALSA2D).

At the time of writing, AIC, AICc, BIC, QAIC, QBIC, QAICc, cross-validation (and at least one other which we'll describe in the next section) are currently implemented.

K-fold cross-validation based on omitting correlated blocks (transects in this case) can be used to choose between models, although this approach can be considerably more time consuming depending on the size of the data set and model complexity.

For this reason, a CV-based approach might be best used to choose between models with different numbers of knots, rather than when trialling knot locations for a particular knot number.

Gaps between knots

While the user must also set a minimum distance between knots trialled (gap in SALSA), in practice the surfaces don't seem to be sensitive to this parameter and can be set to zero (no gap).

Further, there will necessarily be gaps between the candidate knots due to the observed spatial locations and the space filling approach to these.

Model fitting

CReSS based models are linear in their parameters and thus can be fitted using a routine fitting engine and SALSA2D can be used to choose model flexibility. This is implemented inside the `MRSea` package (Scott-Hayward et al., 2017).

Fitting the 1D smoother-based model

```
#OneD model MRSea v1.0.01
> require(MRSea)
#reserved names
> windfarm$x.pos<- windfarm$X
> windfarm$y.pos<- windfarm$Y



#  fit the initial model for this process
> initialModel<- glm(response ~ phase  + as.factor(month) +
                     offset(log(area)),

# create objects containing smooth and factor covariates.
# note X and Y are removed as these will be part of the 2D smooth
> factorlist<-c('phase','month')

> varlist<-c('depth')

# set some inputs for SALSA
> salsa1dlist<-list(fitnessMeasure='QAIC', minKnots_1d = c(1), maxKnots_1d=c(4),
                  startKnots_1d = c(1), degree=c(2), maxIterations=100, gaps=c(0))

> # run SALSA
> salsa1dout<-runSALSA1D(initialModel, salsa1dlist, varlist, factorlist,
                        varlist_cyclicSplines = NULL, splineParams = NULL,
                        datain=windfarm, suppress.printout = FALSE)

> # store best model
> bestModel1D<-salsa1dout$bestModel
```

Fitting the 2D smoother-based model with a phase interaction

Candidate knot locations:

```
> knotgrid<- getKnotgrid(coordData = cbind(windfarm$x.pos, windfarm$y.pos))
```

```
#make distance matrix
> distMats <- makeDists(cbind(windfarm$x.pos, windfarm$y.pos),
    na.omit(knotgrid))


# make parameter set for running salsa2d
> salsa2dlist<-list(fitnessMeasure = 'QAIC', knotgrid = knotgrid,
                    startKnots=6, minKnots=2, maxKnots=20,  gap=0,
                    interactionTerm="phase")



> salsa2dOutput<-runSALSA2D(bestModel1D, salsa2dlist, d2k=distMats$dataDist,
                            k2k=distMats$knotDist, splineParams=NULL, tol=0, chooserad=F,
                            panels=NULL, suppress.printout=FALSE)

> bestModel<-salsa2dOutput$bestModel

> anova(bestModel, test='F')
Analysis of Deviance Table (Type II tests)
Marginal Testing

Response: response
Error estimate based on Pearson residuals

                            SS    Df       F    Pr(>F)
phase                     3039     2  15.122 2.727e-07 ***
as.factor(month)         12179     3  40.405 < 2.2e-16 ***
s(depth)                 44078     4 109.678 < 2.2e-16 ***
s(x.pos, y.pos)          63859     6 105.934 < 2.2e-16 ***
s(x.pos, y.pos):phase    14899    12  12.358 < 2.2e-16 ***
Residuals              3162190 31474
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Results

Under the model, average numbers appear to:

- increase with depth until about 5 metres, after which average numbers generally decline (Figure 55)
- The spatial distribution appears to be complicated with peaks in numbers in the south-east in phases A and B and a move centrally in phase C (Figures 56 & 57).
- All relationships appear to be significantly nonlinear (evidenced by very small p-values for all terms).
- There is spatially explicit change near the wind farm site, and some evidence of wind farm avoidance under the CReSS-based GAM (Figure 58).
- These predictions are only part of the story since we require confidence intervals about these differences (for any model) to make any conclusions about whether these changes are statistically significant - or could be due to chance.

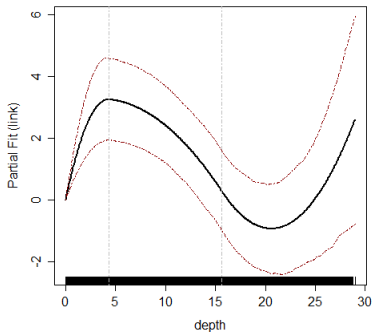


Figure 55: An illustration of the depth relationship for the SALSA based model.

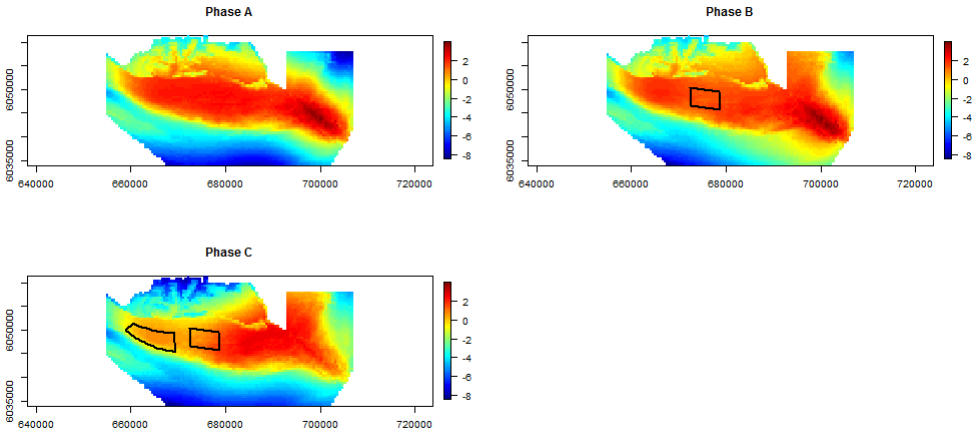


Figure 56: An illustration of the spatial relationship for the CReSS/SALSA based model for each phase.

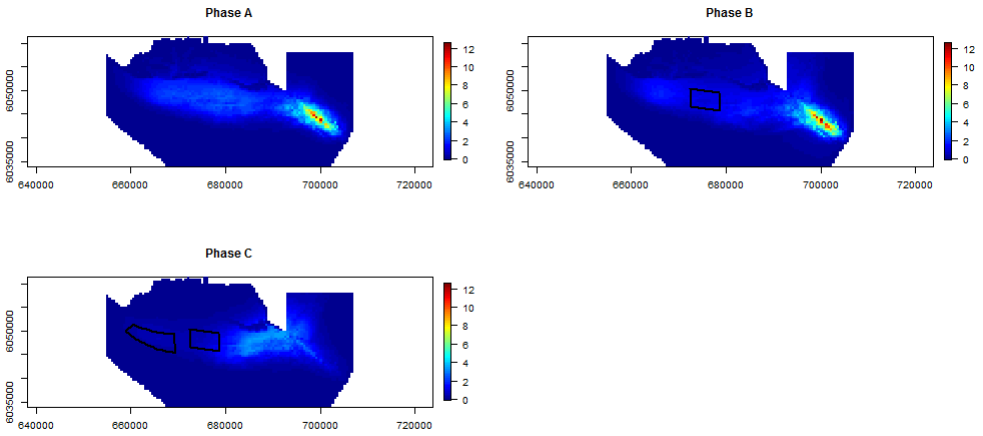


Figure 57: An illustration of the spatial relationship for the CReSS/SALSA based model for each phase.

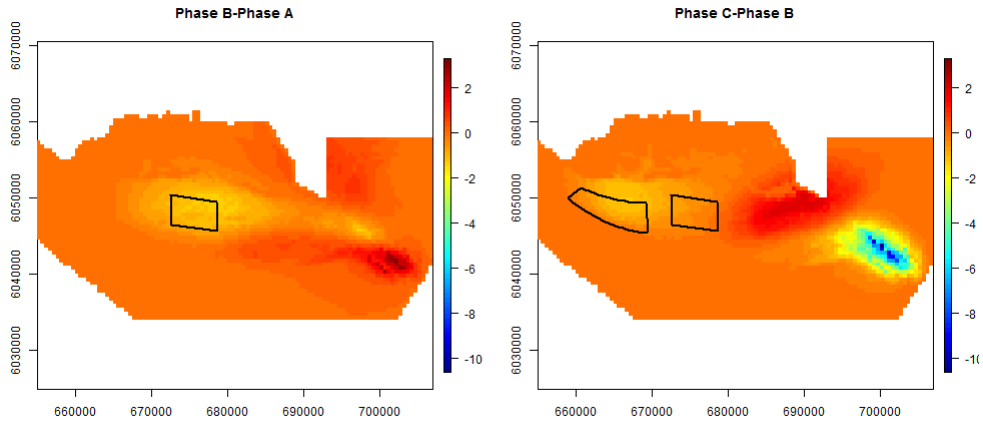


Figure 58: Predicted differences across phases B and A (left) and phases C and B (right).

Limitations

The model selection element for the spatially adaptive smoothing can be more time consuming, compared to some other approaches, however recent work suggests the surfaces produced are good approximations even to highly uneven two dimensional functions.

What next?

We have not assessed our assumption of residual independence for this model and so all model conclusions could be wrong.

We will use the empirical runs test (Appendix B; Mackenzie et al., 2017) and an acf plot for assessment.

- The ordinary runs test has been shown to be inadequate for overdispersed models.
- For the empirical runs test, we manually obtain the reference distribution (under independence) rather than using the Normal distribution.

```
> require(MRSeaPower)
> nsim<-500
> d<-as.numeric(summary(bestModel)$dispersion)

> # simulate some independent data:
> newdat<-generateNoise(nsim, fitted(bestModel), family='poisson', d=d)

> # generate the empirical distribution of runs test statistics:
> empdistribution<-getEmpDistribution(n.sim = nsim, simData=newdat, model = bestModel,
    data = windfarm, plot=FALSE, returnDist = TRUE, dots=FALSE)

> runsTest(residuals(bestModel, type='pearson'), emp.distribution = empdistribution)

Runs Test - Two sided; Empirical Distribution

data:  residuals(bestModel, type = "pearson")
Standardized Runs Statistic = -120.43, p-value = 0.06
```

Conclusion

The result of the runs test show that the assumption of residual independence is not violated ($p > 0.05$) and so our conclusions are valid. However the result of the test is marginal and the acf plot (Figure 59) appears to show some correlation.

For this reason, we will use a method that allows for correlation: **robust standard errors**. These are covered in the next chapter, however some code for calculating these in using the MRSea package in R is shown below.

```
> # define a block structure
> windfarm$panels<-as.numeric(windfarm$unique.transect.label)

> # update gamMRSea model
> bestModelcor<-make.gamMRSea(bestModel, panelid=windfarm$panels)
```

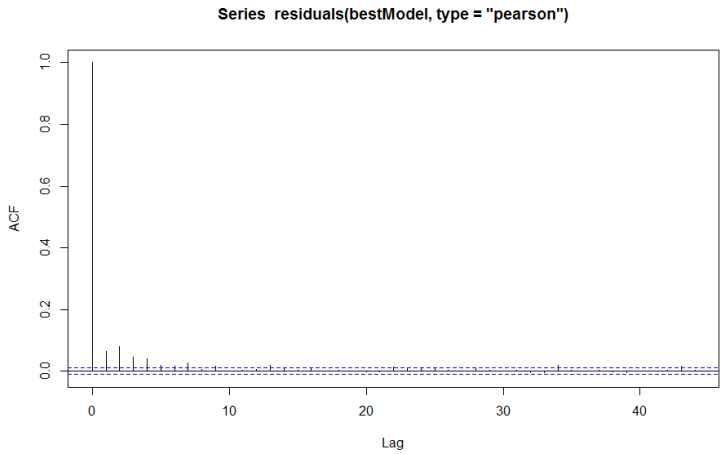


Figure 59: ACF plot

Model summary showing robust s.e.

```
> summary(bestModelcor)
Call:
gamMRSea(formula = response ~ phase + as.factor(month) + bs(depth,
    knots = splineParams[[2]]$knots, degree = splineParams[[2]]$degree,
    Boundary.knots = splineParams[[2]]$bd) + LRF.g(radiusIndices,
    dists, radii, aR) + phase:LRF.g(radiusIndices, dists, radii,
    aR) + offset(log(area)), family = quasipoisson(link = log),
    data = windfarm, splineParams = splineParams)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-14.280   -2.282   -0.817   -0.241   88.787

Coefficients:
                      Estimate Std. Error Robust S.E. t value Pr(>|t|)
(Intercept)          -12.2153     2.8159      5.0143  -2.436 0.014853 *
phaseB                -1.2891     3.4081      5.8635  -0.220 0.825989
phaseC                 2.7227     5.6154      6.0275   0.452 0.651486
as.factor(month)2      0.1267     0.1277      0.1797   0.705 0.480803
as.factor(month)3      0.8132     0.1000      0.1646   4.941 7.83e-07 ***
as.factor(month)4      0.1672     0.1086      0.1741   0.961 0.336766
s(depth)1              3.3021     0.6728      0.5312   6.217 5.14e-10 ***
s(depth)2              2.9664     0.6067      0.4094   7.246 4.38e-13 ***
s(depth)3             -3.0335     0.8207      0.8969  -3.382 0.000720 ***
s(depth)4              2.5566     1.6300      1.0627   2.406 0.016145 *


s(x.pos, y.pos)b1      6.0185     1.2357      1.5187   3.963 7.42e-05 ***
s(x.pos, y.pos)b2     15.5466     1.9223      1.7391   8.940  < 2e-16 ***
s(x.pos, y.pos)b3      4.8508     2.1774      4.3507   1.115 0.264875
s(x.pos, y.pos)b4     -1.4209     3.7999      3.7618  -0.378 0.705638
```

```
s(x.pos, y.pos)b5         17.6886     5.2882      6.3317   2.794 0.005215 **
s(x.pos, y.pos)b6         -4.2383     7.9472      8.0048  -0.529 0.596485
s(x.pos, y.pos)b7        -13.8801     4.0480      4.2433  -3.271 0.001072 **
phaseB:s(x.pos, y.pos)b1   0.9734     1.5322      1.9775   0.492 0.622562
phaseC:s(x.pos, y.pos)b2   3.0290     2.8624      2.5202   1.202 0.229413
phaseB:s(x.pos, y.pos)b3  -7.9847     2.2951      2.1769  -3.668 0.000245 ***
phaseC:s(x.pos, y.pos)b4   2.4990     4.2313      5.0058   0.499 0.617629
phaseB:s(x.pos, y.pos)b5   0.6255     2.6435      4.8977   0.128 0.898380
phaseC:s(x.pos, y.pos)b6  -8.4552     4.3833      5.1656  -1.637 0.101676
phaseB:s(x.pos, y.pos)b7  13.0226     4.3756      4.8530   2.683 0.007291 **
phaseC:s(x.pos, y.pos)b1  11.0411     6.0568      5.0509   2.186 0.028826 *
phaseB:s(x.pos, y.pos)b2   9.7232     6.3813      8.4482   1.151 0.249778

phaseC:s(x.pos, y.pos)b3   0.4565     9.9548      8.8959   0.051 0.959073
phaseB:s(x.pos, y.pos)b4 -25.5350     9.2730     10.7334  -2.379 0.017365 *
phaseC:s(x.pos, y.pos)b5 -11.7324    13.1463     11.0168  -1.065 0.286905
phaseB:s(x.pos, y.pos)b6  19.5655     4.5450      5.1243   3.818 0.000135 ***
phaseC:s(x.pos, y.pos)b7   5.6756     6.2386      6.8391   0.830 0.406613
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasipoisson family taken to be 111.6477)

    Null deviance: 664249  on 31501  degrees of freedom
Residual deviance: 448330  on 31471  degrees of freedom
AIC:  NA

Max Panel Size = 54; Number of panels = 731
Number of Fisher Scoring iterations: 8
```

ANOVA using robust s.e.

```
> anova(bestModelcor)
Analysis of 'Wald statistic' Table
Model: quasipoisson, link: log
Response: response
Marginal Testing
Max Panel Size = 54; Number of panels = 731

                      Df      X2 P(>|Chi|)
phase                  2   0.584    0.7466
as.factor(month)       3  30.533 1.066e-06 ***
s(depth)               4 133.867 < 2.2e-16 ***
s(x.pos, y.pos)        7 115.526 < 2.2e-16 ***
s(x.pos, y.pos):phase 14  80.654 2.140e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

3 Models for correlated data

3.1 Introducing the data

Corneometry is a technology that is used to measure the hydration of the outer layer of the skin.

Specifically, a sensor (corneometer) is pressed against the skin and the epidermal hydration level is typically measured before and repeatedly (e.g. hourly) after the application of a pharmaceutical or cosmetic product.

This process gives rise to so-called 'repeated-measures' data, which will be used to illustrate modelling methods for correlated data.

The data we will consider consists of:

- 25 hourly measurements ($n_i = 25$) each from
- 20 individuals ($s = 20$) over time;
- there are 500 observations in total ($N = 500$).

The first measurement for each subject was a baseline measurement (before application of a product) and the 24 hourly measurements followed the application of a skin cream.

In this section, we are going to examine the scenario where average hydration behaviour over time post-application is the same for all individuals, however unmeasured individual level-covariates also influence the person-specific hydration levels over the surveyed period (Figure 60).

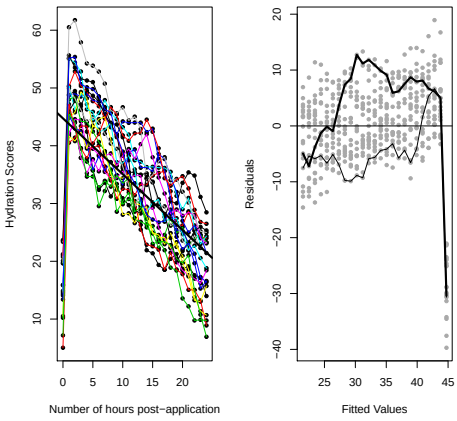


Figure 60: LHS: Scatter plot showing hydration scores over time at baseline and following application of a skin cream. The lines link data from individual subjects over time. RHS: Fitted vs residual plot (based on a linear model) with the residuals linked for a subject 1 (lower solid line) and subject 4 (upper bolded solid line).

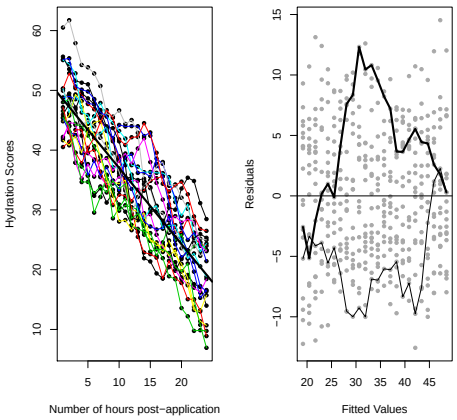


Figure 61: LHS: Scatter plot showing hydration scores over time post-application of a skin cream. The lines link data from individual subjects over time. RHS: Fitted vs residual plot (based on a linear model) with the residuals linked for a subject 1 (lower solid line) and subject 4 (upper bolded solid line).

Since the covariate information which influences why the person-specific responses behave as they do is unknown to us (and is not available to be included in the model), these subject-level patterns over time create identifiable correlated patterns in the noise component (Right-hand plot, Figures 60 and 61).

For illustration with analysis results later, let's ignore the repeated measures nature of the data and pretend the data are independent.
We are going to fit standard linear models to the post application data using Maximum Likelihood(ML)/ Least-Squares:

$$y_{it} = \beta_0 + \beta_1 x_{1it} + e_{it} \tag{43}$$

Here:

- y_{it} is the observation at time point t for the $i-th$ subject in the data set
- x_{1it} is the first (and only in this case) explanatory variable value for subject i at time t
- β_0 is the intercept parameter
- β_1 is the slope parameter
- e_{it} is the error associated with subject i at time t. At this point we will assume these errors are Normally distributed with mean zero and some constant variance term (σ_e^2)

3.2 Fitting Linear Models in SAS using the GENMOD procedure

There are many procedures in SAS that fit standard linear models, but we will be fitting repeated measures models to these data and the GENMOD procedure is useful for this (Littell et al., 2006). For this reason, we will use the GENMOD procedure here, to fit standard linear models:

```
proc genmod data=course.data PLOTS=ALL;
class subject;
model y1=hours /type3;
output out=Results PREDICTED=PREDICTED
LOWER=LOWER UPPER=UPPER RESCHI=RESCHI
RESRAW=RESRAW STDRESCHI=STDRESCHI;
run;
```

Note, the syntax in SAS is not case sensitive - it doesn't matter if you use capital letters or small letters. A bit more about the syntax:

- We are analysing the data set (called DATA) in the course library (DATA=COURSE.DATA)

- We are going to make some fit plots to accompany the fitted model (PLOTS=ALL)
- The response in the data set is called y1 and the explanatory variable is hours (hence the MODEL statement has y1=hours)

We want SAS to supply an output data set (work.Results) which contains (amongst other things):

- the fitted values: $\hat{y}_{it}$ (PREDICTED)
- upper and lower 95% confidence limits for the fitted values (LOWER, UPPER)
- the model residuals: $r_{it} = (y_{it} - \hat{y}_{it})$ (RESRAW) and standardised Pearsons residuals (STDRESCHI):

  $r_{it} = \frac{(y_{it} - \hat{y}_{it})}{\sqrt{v_{it}(1-h_{it})}}$ to give us residuals with equal variance, and a variance of 1 (for perspective)

The GENMOD procedure returns the standard linear model output (Figure 62):

- Estimates for the Intercept ($\hat{\beta}_0 = 51.6728$), Slope ($\hat{\beta}_1 = -1.2748$) and Standard Deviation (SD) of the errors ($\hat{\sigma} = 5.0056$)
- Standard errors for these estimates (0.4716, 0.0330 and 0.1616 for the estimates respectively)
- Upper and lower 95% confidence limits for model parameters: $(50.7485, 52.5972)$, $(-1.3394, -1.2101)$ and $(4.6988, 5.3325)$

The SAS System 13:02 Wednesday, March 12, 2014 **1**

The GENMOD Procedure

Model Information	
Data Set	COURSE.DATA
Distribution	Normal
Link Function	Identity
Dependent Variable	y1

Number of Observations Read	480
Number of Observations Used	480

Class Level Information		
Class	**Levels**	**Values**
Subject	20	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Criteria For Assessing Goodness Of Fit			
Criterion	**DF**	**Value**	**Value/DF**
Deviance	478	12026.9876	25.1611
Scaled Deviance	478	480.0000	1.0042
Pearson Chi-Square	478	12026.9876	25.1611
Scaled Pearson X2	478	480.0000	1.0042
Log Likelihood		-1454.1598	
Full Log Likelihood		-1454.1598	
AIC (smaller is better)		2914.3197	
AICC (smaller is better)		2914.3701	
BIC (smaller is better)		2926.8410	

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	**DF**	**Estimate**	**Standard Error**	**Wald 95% Confidence Limits**		**Wald Chi-Square**	**Pr > ChiSq**
Intercept	1	51.6728	0.4716	50.7485	52.5972	12004.7	<.0001
Hours	1	-1.2748	0.0330	-1.3394	-1.2101	1491.64	<.0001
Scale	1	5.0056	0.1616	4.6988	5.3325		

Note: The scale parameter was estimated by maximum likelihood.

LR Statistics For Type 3 Analysis			
Source	**DF**	**Chi-Square**	**Pr > ChiSq**
Hours	1	678.16	<.0001

Figure 62: SAS Output for a standard linear model.

Based on this output we expect:

- the average hydration level to be somewhere between 50.75 and 52.60 units pre-application (but this is not based on the pre-application data, so should not be interpreted in this case)
- average hydration levels to decrease between 1.21 and 1.34 units per hour post application of the product.

Fit measures for this model are also provided. These include the deviance (D_{normal}), scaled deviance ($D_{scaled;normal}$), AIC, AICc and BIC:

$$D_{normal} = \sum_{i=1}^{s}\sum_{t=1}^{n_i}(y_{it} - \hat{y}_{it})^2 = 12026.9876 \tag{44}$$

$$\begin{aligned} D_{scaled;normal} =& \frac{\sum_{i=1}^{s}\sum_{t=1}^{n_i}(y_{it} - \hat{y}_{it})^2)}{\sigma_e^2} && (45)\\ =& \frac{12026.9876}{5.0056^2} && (46)\\ =& 480 && (47) \end{aligned}$$

$$\begin{aligned} AIC =& -2log - likelihood + 2p && (48)\\ =& -2 \times -1454.1598 + 2 \times 3 && (49)\\ =& 2914.32 && (50) \end{aligned}$$

$$\begin{aligned} AICc =& -2log - likelihood + 2p\frac{n}{n-p-1} && (51)\\ =& -2 \times -1454.1598 + 2 \times 3 \times (480/(480-3-1)) && (52)\\ =& 2914.37 && (53) \end{aligned}$$

$$\begin{aligned} BIC =& -2log - likelihood + \log(N) \times p && (54)\\ =& -2 \times -1454.1598 + \log(480) \times 3 && (55)\\ =& 2926.841 && (56) \end{aligned}$$

These are all relative measures of fit, and so will be useful for comparison with other models later.

The GENMOD procedure also returns a p-value corresponding to a Chi-square test for each parameter, $H_0 : \beta_j = 0$ (for the j-th parameter).

For example, the slope parameter has the following test statistic:

$$\chi^2 = \left(\frac{\text{estimate}}{\text{standard error}}\right)^2 \tag{57}$$

$$= \left(\frac{-1.2748}{0.0330}\right)^2 \tag{58}$$

$$= 1492 \tag{59}$$

In this case, the p-values are both $p < 0.0001$ for the intercept and slope parameters (Figure 62), suggesting the hourly change in the response post application is non-zero.

As a part of a normal analysis, it is advisable to check model assumptions: linearity, constant error variance, independence and normality.

In particular, knowing about the sampling regime, the independence assumption will be suspect, at best.

4 Population Averaged Models; Correlated Errors

4.1 Modelling repeated measures data using GEEs

In order to get reliable uncertainty estimates, we can choose a more appropriate model that acknowledges the correlated nature of the data (Hardin and Hilbe, 2013).

We can still use a linear model (Equation 43) to model the mean hydration scores across a population of individuals, however we will recognize the data structure (repeated measures on individuals) in the errors.

To recap, the linear model can be written in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{60}$$

where

- $\mathbf{y}$ is a $N \times 1$ vector of response values ('stacked' from subject 1 measured at time 1, to the last subject at the last measurement)
- $\mathbf{X}$ is the 'design matrix' (which includes a set of covariate values) which is $N \times p$; p is 2 in this case: one column for the intercept and one column for the 'hour' variable

- $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients
- $\mathbf{e}$ is the vector of errors; this has dimension $N \times 1$

Recall for independent data, the error term: $\mathbf{e} \sim Normal(0, \sigma_e^2)$ is the same as $Normal(0, \sigma_e^2 \mathbf{I})$ where $\mathbf{I}$ is a $N \times N$ diagonal matrix (which means it has a diagonal of 1's and zeros in the off diagonals):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & ..... & 0 \\ 0 & 1 & 0 & 0 & 0 & ..... & 0 \\ 0 & 0 & 1 & 0 & 0 & ..... & 0 \\ 0 & 0 & 0 & 1 & 0 & ..... & 0 \\ 0 & 0 & 0 & 0 & 1 & ..... & 0 \\ . & . & . & . & . & ..... & . \\ . & . & . & . & . & ..... & . \\ 0 & 0 & 0 & 0 & 0 & ..... & 1 \end{bmatrix}$$

Here, the variance multiplies into the diagonal of the matrix (so each error has the same variance) but there is no correlation between the error terms at any time point - the off diagonals in the correlation matrix are zeros.

While simple, this model for the noise is often unsuitable for repeated measures data:

- there are often subject-level variables operating which also influence the response of interest and are correlated with time
- these subject-level variables (and relationships with the response) are often unknown to us (or are at least missing from the model) which means these subject-level patterns are necessarily allocated to the errors
- This results in correlated patterns in the noise term which is assumed to be patternless and independent in a standard linear model

For example:

- air humidity can influence skin hydration in a predictable way (increased humidity results in greater skin hydration) and the effect of air humidity on skin moisture is likely to vary across individuals in the study at different times
- humidity in the air is typically correlated with time. For instance, humidity values in the clinic taken 5 minutes apart are likely to be more similar than humidity values in the clinic taken 5 hours apart

- since humidity and skin hydration are related, and humidity values are correlated with time, humidity-related contributions to the response are also correlated with time. i.e. consecutive values of higher or lower than average humidity values result in higher or lower than average skin hydration scores

For this reason, residuals (which are estimates for the errors) for correlated data are rarely independent. For example, while independent data give a good mix of positive and negative residuals, correlated data show long uninterrupted sequences of positive and negative residuals (Figures 63 and 64).

It is these long uninterrupted sequences of positive and negative residuals which are useful in diagnosing correlation in the errors using, for example, the Runs test.

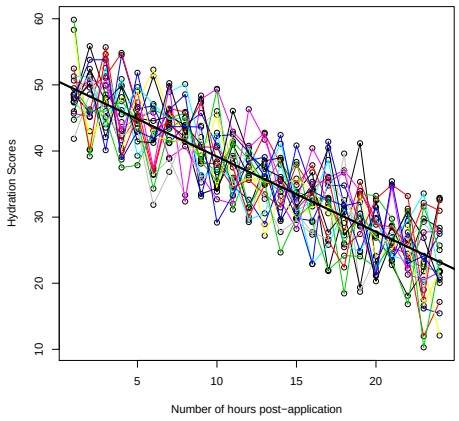


Figure 63: Scatter plots of the number of hours post-application vs hydration scores for independent data and with correlated errors for 20 subjects.

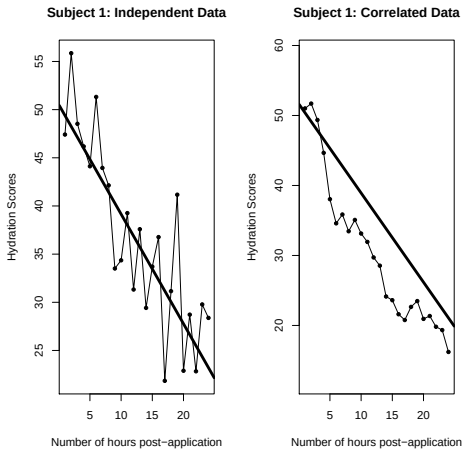


Figure 64: Scatter plots of the number of hours post-application vs hydration scores for independent data and with correlated errors for one subject ($i = 1$).

4.1.1 Different models for the noise

This pattern in the errors within subjects (e.g. Figure 64) can easily be accommodated using a so-called $n_i \times n_i$ 'block-diagonal' structure for the $N \times N$ error matrix ($\mathbf{e}$).

- In this case, the $N \times N$ block-diagonal error matrix will consist of 20 $n_i \times n_i$ blocks (since we have 20 individuals)
- Each block has 24 rows and 24 columns since we have 24 observations for each individual ($n_i = 24$ for all subjects)
- In this case, the error matrix has dimensions: 480×480

After we define our block (or panel) structure, we need to choose a model for the contents of each block.

To investigate the correlation between residuals (over time) within blocks/panels/subjects, empirical autocorrelation functions can be useful (e.g. using the `acf` function in R[20]).

For example, the average autocorrelation between residuals may decay with the time interval between them, as is the case here (Figure 65). In this case,

- the correlation between measurements up to 3 hours apart is significantly non-zero (the 95% confidence intervals at time lags 1 and 2 do not include zero; Figure 65)
- The correlation decays with time and independence (a correlation of zero) is within the bounds of uncertainty for measurements 3 or more hours apart
- Since residual correlation within subjects decays with time (Figure 65), an AR(1) model for the correlation structure is a good starting point.

For illustration, imagine we have just 4 observations for an individual ($n_i = 4$), and an AR(1) model is a sensible representation of the correlated pattern within individuals. This would give a $n_i \times n_i$ AR(1) block ($n_i = 4$):

$$\begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

[20] Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer-Verlag.

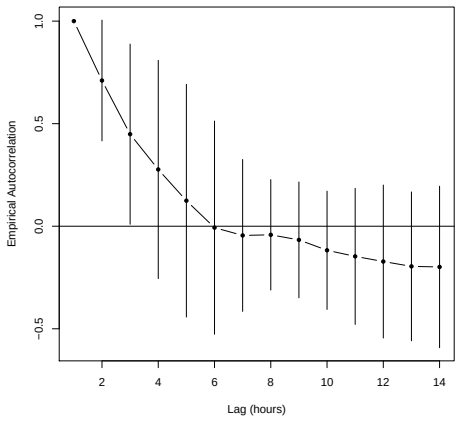


Figure 65: Mean empirical autocorrelation given different lags (hours) across subjects. 95% confidence intervals for the mean at each lag is also shown.

Under this model:

- the correlation between measurements 1 hour apart is the correlation coefficient ρ,
- the correlation 2 hours apart is ρ^2 and
- the correlation for observations 3 hours apart is ρ^3

So, if our data set contained just 2 individuals and 4 observations for each, the 8×8 block-diagonal correlation matrix would look like:

$$\begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 & 0 & 0 & 0 & 0 \\ \rho & 1 & \rho & \rho^2 & 0 & 0 & 0 & 0 \\ \rho^2 & \rho & 1 & \rho & 0 & 0 & 0 & 0 \\ \rho^3 & \rho^2 & \rho & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \rho & \rho^2 & \rho^3 \\ 0 & 0 & 0 & 0 & \rho & 1 & \rho & \rho^2 \\ 0 & 0 & 0 & 0 & \rho^2 & \rho & 1 & \rho \\ 0 & 0 & 0 & 0 & \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$$

Notice, there are two 4×4 blocks inside this block-diagonal matrix. This AR(1) model for the correlation structure sees the correlation between residuals decay as the time interval between observations increases.

For instance, if we assume $\rho = 0.9$:

- this implies a correlation of 0.9 for errors one hour apart and
- $0.9^2 = 0.81$ for residuals 2 hours apart, and
- a correlation of $0.9^3 = 0.729$ for residuals 3 hours apart etc

We can estimate the correlation coefficient (ρ) for the AR(1) structure based on the data, using the `TYPE=AR(1)` option in the `REPEATED` statement in SAS.

There are other correlation structures one can choose to set up the block diagonals.

For instance, if the data scenario was different and a group of randomly selected individuals was measured just once each (e.g. patients visiting a medical centre), then we might suspect that all observations within that group (medical centre) have some common correlation (due to, for example, the staff and facilities available at that centre).

This would lead us to consider:

- A common correlation between measurements within blocks/panels
- an 'exchangeable' or 'compound symmetry' correlation structure
- This structure also only has one associated parameter to estimate (ρ)

e.g. For two medical centres, each with 4 individuals measured just once each, an exchangeable/compound symmetry structure looks like:

$$\begin{bmatrix} 1 & \rho & \rho & \rho & 0 & 0 & 0 & 0 \\ \rho & 1 & \rho & \rho & 0 & 0 & 0 & 0 \\ \rho & \rho & 1 & \rho & 0 & 0 & 0 & 0 \\ \rho & \rho & \rho & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \rho & \rho & \rho \\ 0 & 0 & 0 & 0 & \rho & 1 & \rho & \rho \\ 0 & 0 & 0 & 0 & \rho & \rho & 1 & \rho \\ 0 & 0 & 0 & 0 & \rho & \rho & \rho & 1 \end{bmatrix}$$

We can estimate the common correlation parameter (ρ) using the `TYPE=CS` option in the `REPEATED` statement in SAS.

Let's look at this model in more detail. We now have two components we need to estimate for this model:

- the model for the mean and

- the model for the noise (which includes variance and correlation parameters)

We are going to fit this model using Generalized Estimating Equations (GEEs).

GEEs model the mean response and the association between repeated measures within individuals separately, and so the interpretation of the parameter estimates holds as for standard GLMs

- e.g. the slope coefficient describes how the mean response in the population is expected to change with a one unit increase in the covariate

Here's what the model looks like:

1 The mean of the response vector ($\mathbf{y}_i$) for the i-th individual (with length n_i) is assumed to depend on the $n_i \times p$ covariate matrix ($\mathbf{X}_i$) through a known link function $g(.)$:

$$E(\mathbf{y}_i|\mathbf{X}_i) = \boldsymbol{\mu}_i \tag{61}$$

$$g(\boldsymbol{\mu}_i) = \boldsymbol{\eta}_i = \mathbf{X}_i\boldsymbol{\beta} \tag{62}$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector of population-average (marginal) regression coefficients and $\boldsymbol{\eta}_i$ is the linear predictor for the i-th individual. In this case, the link is the 'identity' and so,

$$g(\boldsymbol{\mu}_i) = \boldsymbol{\mu}_i = \boldsymbol{\eta}_i = \mathbf{X}_i\boldsymbol{\beta} \tag{63}$$

2 The covariance matrix for each individual (which can contain both variance and correlation parameters; $\mathbf{V}_i$) can depend on the mean in some way and some working correlation matrix $\mathbf{R}(\boldsymbol{\alpha})$ (which has some associated coefficients; $\boldsymbol{\alpha}$)

$$\mathbf{V}_i = \phi\mathbf{A}_i^{\frac{1}{2}}\mathbf{W}_i^{-\frac{1}{2}}\hat{\mathbf{R}}_i(\boldsymbol{\alpha})\mathbf{W}_i^{-\frac{1}{2}}\mathbf{A}_i^{\frac{1}{2}} \tag{64}$$

Where:

- ϕ is a scale parameter that is either fixed (and known) or estimated
  - In this case, the errors are assumed Normal and so $\phi = \sigma_e^2$
- $\mathbf{A}_i$ is an $n_i \times n_i$ diagonal matrix with $v(\mu_{it})$ as the j-th diagonal element
  - In this case, the errors are assumed to be Normal and so, $v(\mu_{it}) = 1$

- $\mathbf{W}_i$ is a $n_i \times n_i$ weight matrix which is specified by the user using the `WEIGHT` statement.
  - In this case, all observations have equal weight and so this defaults to a matrix of 1's
- $\mathbf{R}_i$ is the $n_i \times n_i$ correlation matrix. For example, an AR(1) correlation structure.

4.1.2 Obtaining coefficients

In practice, the GEE estimation method alternates between solutions for the fixed effects and for the variance parameters:

1 A solution for the regression coefficients is found assuming independence

2 The working correlations are calculated, based on the standardised residuals:

$$\mathbf{r}_i = \frac{\mathbf{y}_i - \hat{\mathbf{y}}_i}{\sqrt{\phi v(\boldsymbol{\mu}_i)/\mathbf{w}_i}} \tag{65}$$

the current solution for the regression coefficients, and some assumed structure for the correlation matrix, $\mathbf{R}_i(\boldsymbol{\alpha})$

3 The covariance is then re-calculated:

$$\mathbf{V}_i = \phi \mathbf{A}_i^{\frac{1}{2}} \mathbf{W}_i^{-\frac{1}{2}} \widehat{\mathbf{R}_i}(\boldsymbol{\alpha}) \mathbf{W}_i^{-\frac{1}{2}} \mathbf{A}_i^{\frac{1}{2}}$$

4 The regression coefficients are then updated based on the new estimates of the covariance.

5 Steps 2-4 are repeated until convergence

4.1.3 Obtaining standard errors

We can estimate 'model-based' standard errors using the assumed correlation structure, but this is risky unless we know this correlation structure is reasonable.

For example, if the assumed correlation structure is wrong this could return standard errors which are systematically too small or too large.

Alternatively we can use so-called 'robust' (modified sandwich based) standard errors which are instead based on the product of the sums of the residuals within panels.

In this way,

- negative within-panel correlation results in a smaller product of sums than would be obtained under independence while
- positive within-panel correlation results in a larger product of sums than would be obtained under independence.

Note: the 'robustness' property of this 'sandwich' variance estimator is an asymptotic property and works best for balanced longitudinal data[21] with lots of subjects and relatively few repeated measures for each.

We should also note that if the assumed correlation structure is correct, then it is best to specify this correlation structure explicitly in the model.

This returns model-based standard errors which are smaller than the robust alternatives in this case and this reduction in size is appropriate

Note: It is not appropriate to choose one model over the other just because the standard errors are smaller!

4.1.4 Using SAS to fit repeated measures models

The GENMOD procedure can be used to fit Normal errors models with a correlation structure:

```
proc genmod data=course.data PLOTS=ALL;
class subject time ;
model y1=hours ;
repeated subject=subject /corrw covb modelse within=time;

output out=Results DFBETA=DFBETA COOKSD=COOKSD
PREDICTED=PREDICTED LOWER=LOWER UPPER=UPPER
RESCHI=RESCHI
STDRESCHI=STDRESCHI RESRAW=RESRAW CLEVERAGE=CLEVERAGE
CLUSTERCOOKSD=CLUSTERCOOKSD DFBETAC=_all_ ;
effectplot fit;
assess var=(hours) / resample seed=603708000;
run;
```

A bit more about the syntax:

- We have repeated measures within subjects (`REPEATED`): in this case the 'subject' variable contains the subject identifier
- We want to view the fitted correlation matrix for the within subject residuals, based on some assumed structure (`corrw`).

  When `TYPE` is not specified, independence is the working correlation structure for the errors.

[21] where each subject has the same number of observations

Note: if we use 'empirical' standard errors then we don't need to rely on the model to give us good standard errors (in many cases).

So, in the absence of any good choices (or information about) the 'best' correlation structure, we can base our conclusions on the 'empirical' confidence intervals for each parameter.

Of course if the correct correlation structure is available, then this should be specified as the working correlation structure:

- the coefficients depend, in part, on the working correlation structure

- the model-based standard errors will be slightly smaller (and thus better) than the empirical versions if the correct correlation structure is used

At the very least, the empirical standard errors are available for comparison with the model-based results.

If we modify the `REPEATED` statement to include the `TYPE` option we can include AR(1) or unstructured errors:

- ```
  repeated subject=subject/ type=AR(1)
    corrw covb modelse within=time;
  ```
- ```
  repeated subject=subject/ type=un
    corrw covb modelse within=time;
  ```

Note: If we have missing values (e.g. if some subjects didn't show up every hour for 24 hours post application) then we need the `within=time` option.

This ensures that any gaps in the data in time are respected, and contribute to the correlation matrix appropriately.

Back to the syntax:

- We want to see the covariance matrix for the coefficients under the chosen model (the `modelse` option in the `REPEATED` statement) and the empirical version of the covariance matrix (the `covb` option in the `REPEATED` statement)

Note: This analysis assumes equal spaces between measurements; we need a different correlation structure (and different procedure) to accommodate unequally spaced measurements in SAS.

4.1.5 Comparing results under different correlation structures

- The working correlation matrix is clearly diagonal when the SAS default is specified (Figure 67)

- The correlation coefficient estimated for the AR(1) correlation is 0.9004 (Figure 71)

- The correlation coefficients estimated for the unstructured correlation are much smaller: 0.6165, 0.5448, 0.4576, 0.4257 and so on (Figure 75).

The model-based standard errors (SEs) were often very different to the empirical standard errors (ESEs):

- The independence based SE's for the intercept and slope coefficients are almost two and a half times the size of the ESE (Figures 68 & 69)

- the AR(1) based SE is closer to the ESE for the intercept parameter (the SE is 85% of the size of the ESE), and there is very little difference between the SE and ESE for the slope coefficient. The SE is 99% of the size of the ESE for the slope coefficient (Figures 72 & 73).

- the unstructured-based SEs were the most different to the ESE for both the intercept parameter (the SE is just 16% of the size of the ESE), and there is little resemblance between the SE and ESE for the slope coefficient (the SE is just 8% of the size of the ESE)(Figures 76 & 77).

These specified 'correlation' structures may not be consistent with the data, but we can use model selection procedures to help us choose both the correlation structure and model for the mean (e.g. model covariates).

The SAS System 11:56 Wednesday, March 26, 2014 1

The GENMOD Procedure

Model Information	
Data Set	COURSE.DATA
Distribution	Normal
Link Function	Identity
Dependent Variable	y1

Number of Observations Read	480
Number of Observations Used	480

Class Level Information		
Class	**Levels**	**Values**
Subject	20	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
time	24	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	Hours

Algorithm converged.

GEE Model Information	
Correlation Structure	Independent
Within-Subject Effect	time (24 levels)
Subject Effect	Subject (20 levels)
Number of Clusters	20
Correlation Matrix Dimension	24
Maximum Cluster Size	24
Minimum Cluster Size	24

Covariance Matrix (Model-Based)		
	Prm1	**Prm2**
Prm1	0.22335	-0.01367
Prm2	-0.01367	0.001094

Figure 66: SAS Output for a GEE Model - Independent working correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 3

The GENMOD Procedure

Working Correlation Matrix

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13
Row1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row3	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row4	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row5	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row6	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
Row10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
Row11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Row12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
Row13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
Row14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 67: SAS Output for a GEE Model - Independent working correlation matrix.

The GENMOD Procedure

Working Correlation Matrix											
	Col14	Col15	Col16	Col17	Col18	Col19	Col20	Col21	Col22	Col23	Col24
Row1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row14	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row15	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row16	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row17	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row18	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row19	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Row20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
Row21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
Row22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Row23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
Row24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

GEE Fit Criteria	
QIC	510.6668
QICu	482.0000

Analysis Of GEE Parameter Estimates						
Empirical Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	51.6728	1.0928	49.5309	53.8147	47.28	<.0001
Hours	-1.2748	0.0813	-1.4341	-1.1154	-15.68	<.0001

Figure 68: SAS Output for a GEE Model - Independent working correlation matrix.

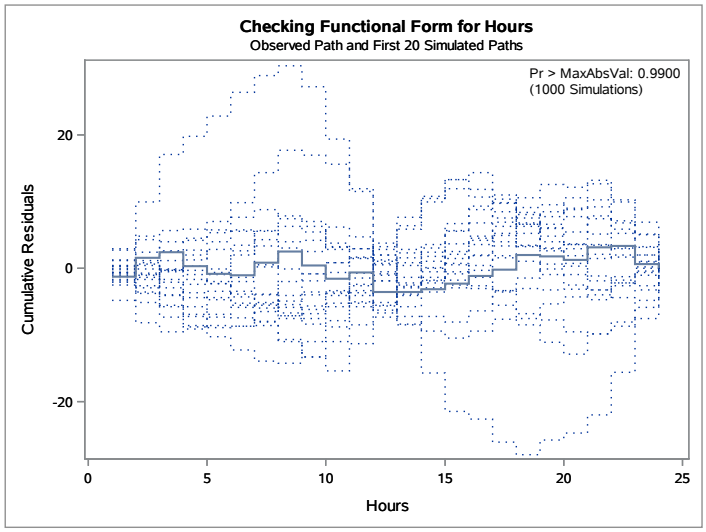
The SAS System 11:56 Wednesday, March 26, 2014 5

The GENMOD Procedure

Analysis Of GEE Parameter Estimates						
Model-Based Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	51.6728	0.4726	50.7465	52.5991	109.34	<.0001
Hours	-1.2748	0.0331	-1.3396	-1.2099	-38.54	<.0001
Scale	5.0161	.	.	.	.	.

Note: The scale parameter for GEE estimation was computed as the square root of the normalized Pearson's chi-square.

Assessment Summary				
Assessment Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
Hours	3.6384	1000	603708000	0.9900



Score Statistics For Type 3 GEE Analysis			
Source	DF	Chi-Square	Pr > ChiSq
Hours	1	18.50	<.0001

Figure 69: SAS Output for a GEE Model - Independent working correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 1

The GENMOD Procedure

Model Information	
Data Set	COURSE.DATA
Distribution	Normal
Link Function	Identity
Dependent Variable	y1

Number of Observations Read	480
Number of Observations Used	480

Class Level Information		
Class	**Levels**	**Values**
Subject	20	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
time	24	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	Hours

Algorithm converged.

GEE Model Information	
Correlation Structure	AR(1)
Within-Subject Effect	time (24 levels)
Subject Effect	Subject (20 levels)
Number of Clusters	20
Correlation Matrix Dimension	24
Maximum Cluster Size	24
Minimum Cluster Size	24

Covariance Matrix (Model-Based)		
	Prm1	**Prm2**
Prm1	1.23717	-0.05333
Prm2	-0.05333	0.004266

Figure 70: SAS Output for a GEE Model - AR(1) correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 **3**

The GENMOD Procedure

Working Correlation Matrix

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13
Row1	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839
Row2	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153
Row3	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502
Row4	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889
Row5	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319
Row6	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797
Row7	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328
Row8	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917
Row9	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572
Row10	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299
Row11	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107
Row12	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004
Row13	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000
Row14	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004
Row15	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107
Row16	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299
Row17	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572
Row18	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917
Row19	0.1512	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797	0.5328
Row20	0.1362	0.1512	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319	0.4797
Row21	0.1226	0.1362	0.1512	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889	0.4319
Row22	0.1104	0.1226	0.1362	0.1512	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502	0.3889
Row23	0.0994	0.1104	0.1226	0.1362	0.1512	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153	0.3502
Row24	0.0895	0.0994	0.1104	0.1226	0.1362	0.1512	0.1680	0.1866	0.2072	0.2301	0.2556	0.2839	0.3153

Figure 71: SAS Output for a GEE Model - AR(1) correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 4

The GENMOD Procedure

Working Correlation Matrix											
	Col14	Col15	Col16	Col17	Col18	Col19	Col20	Col21	Col22	Col23	Col24
Row1	0.2556	0.2301	0.2072	0.1866	0.1680	0.1512	0.1362	0.1226	0.1104	0.0994	0.0895
Row2	0.2839	0.2556	0.2301	0.2072	0.1866	0.1680	0.1512	0.1362	0.1226	0.1104	0.0994
Row3	0.3153	0.2839	0.2556	0.2301	0.2072	0.1866	0.1680	0.1512	0.1362	0.1226	0.1104
Row4	0.3502	0.3153	0.2839	0.2556	0.2301	0.2072	0.1866	0.1680	0.1512	0.1362	0.1226
Row5	0.3889	0.3502	0.3153	0.2839	0.2556	0.2301	0.2072	0.1866	0.1680	0.1512	0.1362
Row6	0.4319	0.3889	0.3502	0.3153	0.2839	0.2556	0.2301	0.2072	0.1866	0.1680	0.1512
Row7	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839	0.2556	0.2301	0.2072	0.1866	0.1680
Row8	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839	0.2556	0.2301	0.2072	0.1866
Row9	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839	0.2556	0.2301	0.2072
Row10	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839	0.2556	0.2301
Row11	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839	0.2556
Row12	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153	0.2839
Row13	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502	0.3153
Row14	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889	0.3502
Row15	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319	0.3889
Row16	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797	0.4319
Row17	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328	0.4797
Row18	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917	0.5328
Row19	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572	0.5917
Row20	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299	0.6572
Row21	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107	0.7299
Row22	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004	0.8107
Row23	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000	0.9004
Row24	0.3502	0.3889	0.4319	0.4797	0.5328	0.5917	0.6572	0.7299	0.8107	0.9004	1.0000

GEE Fit Criteria	
QIC	503.6467
QICu	482.0000

Analysis Of GEE Parameter Estimates						
Empirical Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	51.5193	0.9465	49.6642	53.3743	54.43	<.0001
Hours	-1.2694	0.0657	-1.3982	-1.1406	-19.32	<.0001

Figure 72: SAS Output for a GEE Model - AR(1) correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 5

The GENMOD Procedure

Analysis Of GEE Parameter Estimates						
Model-Based Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	51.5193	1.1123	49.3393	53.6993	46.32	<.0001
Hours	-1.2694	0.0653	-1.3974	-1.1414	-19.43	<.0001
Scale	5.0170	.	.	.	.	.

Note: The scale parameter for GEE estimation was computed as the square root of the normalized Pearson's chi-square.

Assessment Summary				
Assessment Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
Hours	12.3541	1000	603708000	0.8920

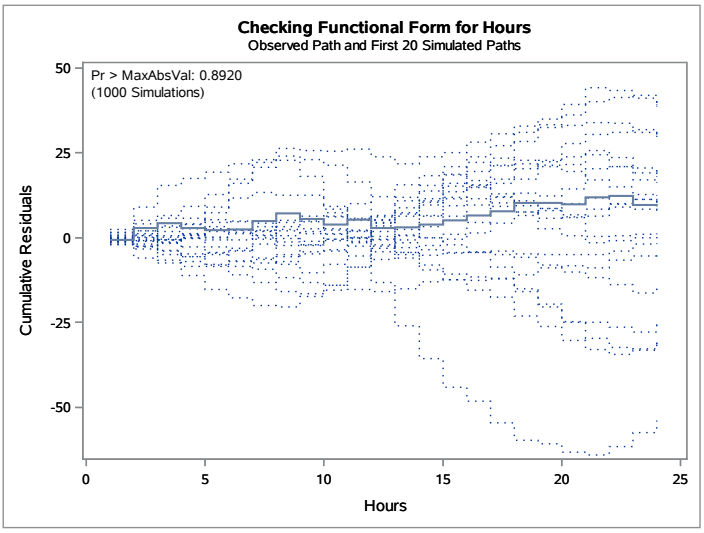


Figure 73: SAS Output for a GEE Model - AR(1) correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 1

The GENMOD Procedure

Model Information	
Data Set	COURSE.DATA
Distribution	Normal
Link Function	Identity
Dependent Variable	y1

Number of Observations Read	480
Number of Observations Used	480

Class Level Information		
Class	**Levels**	**Values**
Subject	20	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
time	24	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Parameter Information	
Parameter	**Effect**
Prm1	Intercept
Prm2	Hours

Algorithm converged.

GEE Model Information	
Correlation Structure	Unstructured
Within-Subject Effect	time (24 levels)
Subject Effect	Subject (20 levels)
Number of Clusters	20
Correlation Matrix Dimension	24
Maximum Cluster Size	24
Minimum Cluster Size	24

Covariance Matrix (Model-Based)		
	Prm1	**Prm2**
Prm1	0.08879	0.003541
Prm2	0.003541	0.0001604

Figure 74: SAS Output for a GEE Model - Unstructured working correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 3

The GENMOD Procedure

Working Correlation Matrix

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13
Row1	1.0000	0.6165	0.5448	0.4576	0.4257	0.3947	0.3923	0.3082	0.2953	0.2714	0.1587	-0.0281	-0.0889
Row2	0.6165	1.0000	0.6791	0.5771	0.5307	0.4973	0.4436	0.2987	0.2421	0.2729	0.1480	-0.0372	-0.0766
Row3	0.5448	0.6791	1.0000	0.7615	0.6358	0.6968	0.5730	0.3505	0.2744	0.3105	0.1925	-0.0011	-0.0812
Row4	0.4576	0.5771	0.7615	1.0000	0.7127	0.7626	0.6246	0.3862	0.2955	0.2969	0.1389	-0.0362	-0.1069
Row5	0.4257	0.5307	0.6358	0.7127	1.0000	0.7626	0.6778	0.4622	0.3784	0.3428	0.1524	-0.0044	0.0116
Row6	0.3947	0.4973	0.6968	0.7626	0.7626	1.0000	0.7626	0.5934	0.4944	0.4702	0.3048	0.1671	0.1857
Row7	0.3923	0.4436	0.5730	0.6246	0.6778	0.7626	1.0000	0.6184	0.5240	0.4918	0.3639	0.1924	0.2461
Row8	0.3082	0.2987	0.3505	0.3862	0.4622	0.5934	0.6184	1.0000	0.4865	0.4147	0.3395	0.2328	0.2643
Row9	0.2953	0.2421	0.2744	0.2955	0.3784	0.4944	0.5240	0.4865	1.0000	0.5042	0.3550	0.2880	0.3669
Row10	0.2714	0.2729	0.3105	0.2969	0.3428	0.4702	0.4918	0.4147	0.5042	1.0000	0.4582	0.3880	0.4225
Row11	0.1587	0.1480	0.1925	0.1389	0.1524	0.3048	0.3639	0.3395	0.3550	0.4582	1.0000	0.4923	0.4937
Row12	-0.0281	-0.0372	-0.0011	-0.0362	-0.0044	0.1671	0.1924	0.2328	0.2880	0.3880	0.4923	1.0000	0.6355
Row13	-0.0889	-0.0766	-0.0812	-0.1069	0.0116	0.1857	0.2461	0.2643	0.3669	0.4225	0.4937	0.6355	1.0000
Row14	-0.1083	-0.1691	-0.1699	-0.1257	0.0842	0.2912	0.3653	0.4089	0.4877	0.4626	0.5209	0.5370	0.7537
Row15	-0.0795	-0.1130	-0.0969	-0.0617	0.0909	0.3020	0.3535	0.4147	0.4584	0.4632	0.5050	0.5826	0.7500
Row16	0.0354	-0.0453	0.0100	0.0149	0.1414	0.3787	0.4080	0.4851	0.5131	0.4648	0.5187	0.5201	0.6690
Row17	0.1083	0.0147	0.0541	0.0623	0.1480	0.3544	0.3901	0.4593	0.5082	0.4349	0.4650	0.4306	0.6168
Row18	0.0731	-0.0222	0.0415	0.0479	0.1076	0.2694	0.3166	0.4198	0.4463	0.2945	0.3492	0.3469	0.4698
Row19	-0.0112	-0.1156	-0.0726	-0.1026	-0.0395	0.1100	0.2058	0.3118	0.3280	0.1794	0.2718	0.3082	0.4191
Row20	-0.0759	-0.1889	-0.1673	-0.1298	-0.0098	0.1035	0.2091	0.3431	0.4264	0.2056	0.2321	0.2781	0.4260
Row21	0.0352	-0.0379	-0.0511	-0.0073	0.1133	0.2276	0.3540	0.4656	0.5927	0.3360	0.3068	0.2821	0.4838
Row22	0.0161	-0.0446	-0.1101	-0.0579	0.0332	0.0892	0.1797	0.3267	0.4544	0.2244	0.1991	0.2334	0.3351
Row23	0.0192	-0.0722	-0.1136	-0.0467	-0.0103	0.0141	0.0969	0.2430	0.3669	0.1344	0.0904	0.1137	0.1900
Row24	0.0072	-0.0580	-0.0667	-0.0057	0.0583	0.1121	0.2047	0.3084	0.3921	0.1957	0.1181	0.1291	0.2639

Figure 75: SAS Output for a GEE Model - Unstructured working correlation matrix.

The SAS System 11:56 Wednesday, March 26, 2014 **4**

The GENMOD Procedure

Working Correlation Matrix

	Col14	Col15	Col16	Col17	Col18	Col19	Col20	Col21	Col22	Col23	Col24
Row1	-0.1083	-0.0795	0.0354	0.1083	0.0731	-0.0112	-0.0759	0.0352	0.0161	0.0192	0.0072
Row2	-0.1691	-0.1130	-0.0453	0.0147	-0.0222	-0.1156	-0.1889	-0.0379	-0.0446	-0.0722	-0.0580
Row3	-0.1699	-0.0969	0.0100	0.0541	0.0415	-0.0726	-0.1673	-0.0511	-0.1101	-0.1136	-0.0667
Row4	-0.1257	-0.0617	0.0149	0.0623	0.0479	-0.1026	-0.1298	-0.0073	-0.0579	-0.0467	-0.0057
Row5	0.0842	0.0909	0.1414	0.1480	0.1076	-0.0395	-0.0098	0.1133	0.0332	-0.0103	0.0583
Row6	0.2912	0.3020	0.3787	0.3544	0.2694	0.1100	0.1035	0.2276	0.0892	0.0141	0.1121
Row7	0.3653	0.3535	0.4080	0.3901	0.3166	0.2058	0.2091	0.3540	0.1797	0.0969	0.2047
Row8	0.4089	0.4147	0.4851	0.4593	0.4198	0.3118	0.3431	0.4656	0.3267	0.2430	0.3084
Row9	0.4877	0.4584	0.5131	0.5082	0.4463	0.3280	0.4264	0.5927	0.4544	0.3669	0.3921
Row10	0.4626	0.4632	0.4648	0.4349	0.2945	0.1794	0.2056	0.3360	0.2244	0.1344	0.1957
Row11	0.5209	0.5050	0.5187	0.4650	0.3492	0.2718	0.2321	0.3068	0.1991	0.0904	0.1181
Row12	0.5370	0.5826	0.5201	0.4306	0.3469	0.3082	0.2781	0.2821	0.2334	0.1137	0.1291
Row13	0.7537	0.7500	0.6690	0.6168	0.4698	0.4191	0.4260	0.4838	0.3351	0.1900	0.2639
Row14	1.0000	0.7626	0.7626	0.7626	0.6502	0.5757	0.6519	0.7562	0.5685	0.4209	0.4732
Row15	0.7626	1.0000	0.7626	0.7626	0.6961	0.5990	0.6392	0.7266	0.5559	0.4103	0.4801
Row16	0.7626	0.7626	1.0000	0.7626	0.7626	0.7540	0.7498	0.7626	0.6623	0.5291	0.5552
Row17	0.7626	0.7626	0.7626	1.0000	0.7626	0.7497	0.7220	0.7626	0.6364	0.5229	0.5653
Row18	0.6502	0.6961	0.7626	0.7626	1.0000	0.7399	0.7526	0.7626	0.6568	0.5348	0.5024
Row19	0.5757	0.5990	0.7540	0.7497	0.7399	1.0000	0.7439	0.7626	0.6241	0.5261	0.4807
Row20	0.6519	0.6392	0.7498	0.7220	0.7526	0.7439	1.0000	0.7626	0.7626	0.6896	0.6207
Row21	0.7562	0.7266	0.7626	0.7626	0.7626	0.7626	0.7626	1.0000	0.7626	0.7626	0.7569
Row22	0.5685	0.5559	0.6623	0.6364	0.6568	0.6241	0.7626	0.7626	1.0000	0.7626	0.7626
Row23	0.4209	0.4103	0.5291	0.5229	0.5348	0.5261	0.6896	0.7626	0.7626	1.0000	0.7626
Row24	0.4732	0.4801	0.5552	0.5653	0.5024	0.4807	0.6207	0.7569	0.7626	0.7626	1.0000

GEE Fit Criteria	
QIC	537.3719
QICu	482.0000

Analysis Of GEE Parameter Estimates

Empirical Standard Error Estimates

Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	50.7815	1.8176	47.2191	54.3438	27.94	<.0001
Hours	-1.2614	0.1628	-1.5804	-0.9424	-7.75	<.0001

Figure 76: SAS Output for a GEE Model - Unstructured working correlation matrix.

The GENMOD Procedure

Analysis Of GEE Parameter Estimates						
Model-Based Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	50.7815	0.2980	50.1975	51.3655	170.43	<.0001
Hours	-1.2614	0.0127	-1.2862	-1.2366	-99.59	<.0001
Scale	5.0692	.	.	.	.	.

Note: The scale parameter for GEE estimation was computed as the square root of the normalized Pearson's chi-square.

Assessment Summary				
Assessment Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
Hours	77.7618	1000	603708000	0.4840

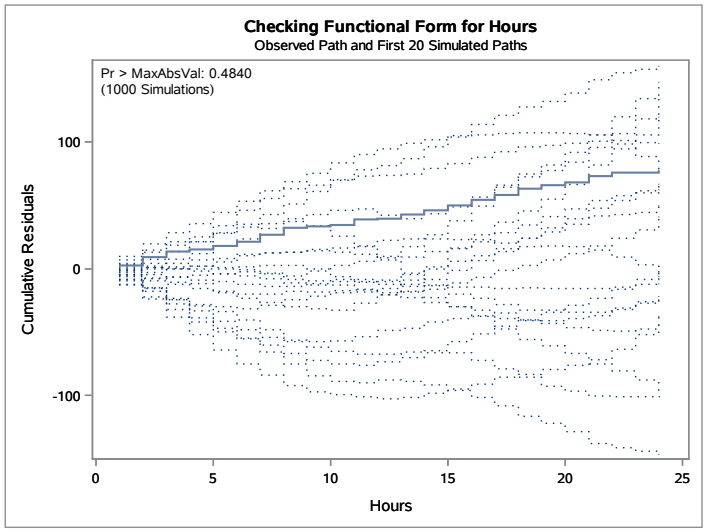


Figure 77: SAS Output for a GEE Model - Unstructured working correlation matrix.

4.2 Model Selection for GEEs

4.2.1 Choosing model covariates

Generalized Linear Models (GLMs) are Maximum Likelihood (ML) based, and so ML-based fit statistics (e.g. AIC) are often used to choose between models with different covariates. In contrast to GLMs, GEEs are Quasi-Likelihood (QL) based and so, QL-based QIC scores must be used instead (Hardin and Hilbe, 2013).

The form of the quasi-likelihood (QL) component depends on the nature of the response being modelled. Common choices are:

Normal	$-\frac{1}{2}\sum_{i=1}^{s}\sum_{t=1}^{n_i} w_{it}(y_{it}-\mu_{it})^2$
Binomial (k)	$\sum_{i=1}^{s}\sum_{t=1}^{n_i} w_{it}\left\{y_{it}\log\left(\frac{\mu_{it}}{1-\mu_{it}}\right)+\log(1-\mu_{it})\right\}$
Poisson	$\sum_{i=1}^{s}\sum_{t=1}^{n_i} w_{it}\{y_{it}\log\mu_{it}-\mu_{it}\}$

where w_{it} are the weights for any given observation (these are all equal to one in this case), and the y_{it} and μ_{it} are the observed values and mean of interest in each case.

Note, these QL contributions to the QIC scores (which will be defined in a moment) are calculated assuming independence in the errors (so the coefficients which generate the fitted values ($\hat{\mu}_{it}$) under each model are fitted assuming independence).

The quasi-likelihood (QL) under the independence model information criterion (QIC) are used to choose between GEE-based models, and there are two sorts of QIC statistic: the QIC_u and $QIC(\boldsymbol{R})$.

- The QIC_u statistic can be used to choose between models with different sets of covariates while

- the $QIC(\boldsymbol{R})$ can be used to choose between models with the same covariates but different correlation structures

The QIC_u is calculated using:

$$QIC_u = -2QL(g^{-1}(\mathbf{X}\boldsymbol{\beta}_{\mathbf{R}})) + 2p \tag{66}$$

and **smaller is better**. This statistic is composed of:

- the QL component, which is specific to the error distribution assumed
- g, which is the link function and
- the $\mathbf{X}$ is the $N \times p$ design matrix (set of covariates) for the model
- The regression coefficients ($\boldsymbol{\beta}_{\mathbf{R}}$) are fitted assuming some correlation structure ($\mathbf{R}$) and
- $2p$ is the penalty term and p is the number of parameters incurred by fitting the model

4.2.2 Choosing a correlation structure

The $QIC(\boldsymbol{R})$ statistic accounts for the correlation structure more explicitly and can be used to discriminate between models with different correlation structures; **smaller is better**. This statistic is calculated using:

$$QIC(R) = -2QL(g^{-1}(\mathbf{X}\boldsymbol{\beta}_{\mathbf{R}})) + 2\ trace(\mathbf{V}_{\mathbf{I}}^{-1}\mathbf{V}_{M,\mathbf{R}}) \tag{67}$$

where:

- QL is the quasi-likelihood component (based on the error distribution assumed for the noise)
- g is the link function
- $\mathbf{X}$ is the $N \times p$ design matrix for the model

- $\boldsymbol{\beta}_{\mathbf{R}}$ is the set of coefficients obtained under some specified correlation structure $\mathbf{R}$

Of course the coefficients are estimated, so we use: $\hat{\mu}_{it} = g^{-1}(\mathbf{X}_{it}\hat{\boldsymbol{\beta}}_{\mathbf{R}})$ to calculate the $QIC(\mathbf{R})$ score based on some working correlation structure. This statistic also contains:

- the trace of a matrix (which is the sum of the diagonal elements)

- $\mathbf{V}_{\mathbf{I}}$ is the variance matrix obtained by fitting an independence model and
- $\mathbf{V}_{M,\mathbf{R}}$ is the model-based estimate of variance from the model with some chosen correlation structure $\mathbf{R}$.

Some general guidance about the choice of correlation structure is also provided by Hardin and Hilbe (2013).

- If the size of the panels is small and the data are complete, use the unstructured specification (where possible)
- If the observations within panels are collected across time use a correlation structure that also includes a time dependence

- If the observations are clustered (but not collected over time) use the exchangeable/compound symmetry structure

- If the numbers of panels is small, then the independence model may be best. However, use the empirical estimate of variance when hypothesis testing and/or interpreting model coefficients

Correlation Structure	QIC(R)
Independent	510.6668
AR(1)	503.6467
Unstructured	537.3719

The QIC_u and $QIC(\boldsymbol{R})$ scores are shown by default in SAS and are labelled QICu and QIC respectively (e.g. Figures 68, 72 and 76)

In this case, the QIC_u score for this model with one covariate (regardless of correlation structure) is 482, while the $QIC(\mathbf{R})$ scores for the hydration data naturally differ by correlation structure:

In this case the AR(1) model looks preferable; the $QIC(\mathbf{R})$ is considerably lower than the independence case and the larger number of parameters for the unstructured model appear unjustified.

The QIC score is also a relative measure of fit, and so none of these models might be suitable!

For example, we saw the empirical standard errors were quite different to the model-based SEs for both the independence and unstructured models, indicating the models for the standard errors were inappropriate.

We will use the working independence model to obtain the coefficients and the empirical standard errors as the basis for any model conclusions (e.g. based on p-values).

We could also use the AR(1) model with model based standard errors, as the estimates of the coefficients (intercept and slope) are almost identical to that of the independence model

4.3 Model Assessment for GEEs

In keeping with GLMs, it is necessary to assess the fit of the model for the signal using, in this case:

- the fitted relationship with hour (with standard error bars) (Figure 78);

  this provides an idea about the predicted values across time and the uncertainty about the fitted relationship

- It is also useful to examine the raw and Pearsons residuals in observation order, in this case since individuals are followed over time.

In this case, the residuals appear to be correlated in time (Figure 79) which indicates that independence is unreasonable and a correlation structure is likely to be necessary. This can easily be confirmed by the runs test (and is).

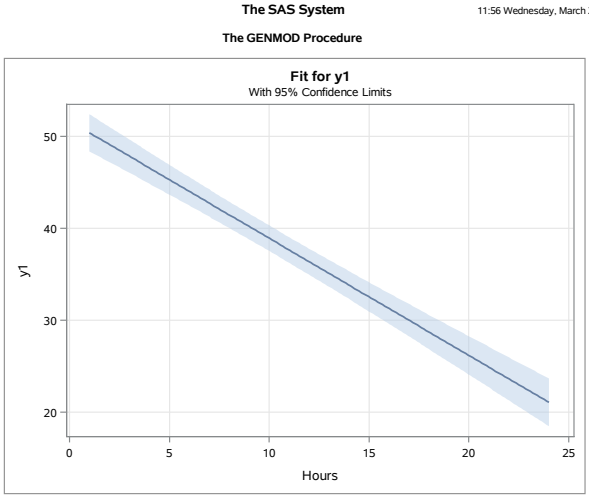


Figure 78: SAS Output for a GEE Model - Independent working correlation matrix.

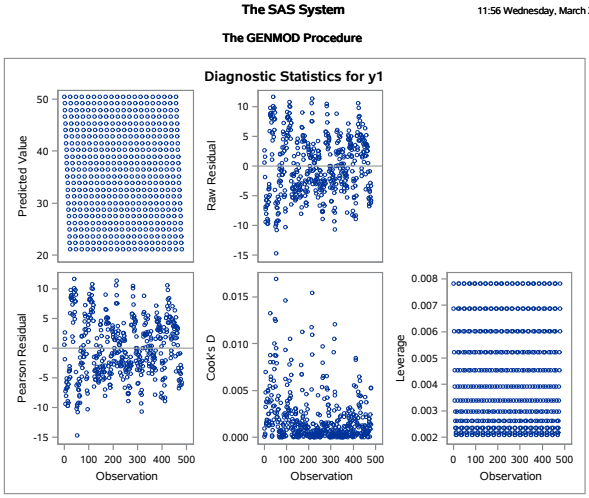


Figure 79: SAS Output for a GEE Model - Independent working correlation matrix.

We can also assess if the form of the relationship is appropriate using cumulative sums of residuals over a set of co-ordinates (e.g. covariates or linear predictors)

Here, systematic over or under-prediction across the covariate being considered tells us we have a problem with linearity.

The process works as follows:

- The cumulative residuals based on the model (the 'observed path') are compared with (simulated) sets of cumulative residuals which are likely to be obtained if the linear model is appropriate

- In general, a discrepancy between the simulated sets (based on the linear model being correct) and the cumulative residuals for our model, give evidence that the model is inappropriate

- This discrepancy between model residuals and those expected under a linear model return a p-value; small p-values signify a problem with the linear model

The `ASSESS` statement helps us determine if our linear model is a reasonable description of the relationship between hours and the response:

```
assess var=(hours) / resample ;
```

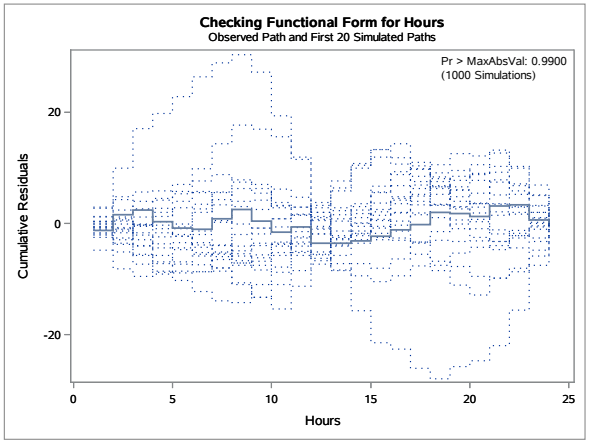
The SAS System 11:56 Wednesday, March 26, 2014 5

The GENMOD Procedure

Analysis Of GEE Parameter Estimates						
Model-Based Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	51.6728	0.4726	50.7465	52.5991	109.34	<.0001
Hours	-1.2748	0.0331	-1.3396	-1.2099	-38.54	<.0001
Scale	5.0161	.	.	.	.	.

Note: The scale parameter for GEE estimation was computed as the square root of the normalized Pearson's chi-square.

Assessment Summary				
Assessment Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
Hours	3.6384	1000	603708000	0.9900



Score Statistics For Type 3 GEE Analysis			
Source	DF	Chi-Square	Pr > ChiSq
Hours	1	18.50	<.0001

Figure 80: SAS Output for a GEE Model - Independent working correlation matrix.

In this case, the cumulative residuals do not appear to become systematically negative or positive with time (Figure 80), which implies no problems with linearity ($p = 0.99$).

4.3.1 Assessing influential points and/or individuals

As with GLMs, it is also important to ensure model results are not dictated by a few particularly strange points.

In this case however, we have repeated measures data on a set of individuals and equally, we don't want a couple of strange individuals heavily influencing model results.

For this reason, it is useful to assess the influence of individual data points and influential individuals using **deletion diagnostics**:

- We find the influence of individual data points[22] by measuring the difference in the fitted regression coefficients by deleting a single observation from the data and re-fitting the model.

Here, relatively large values signify influential points. These values are calculated in SAS by specifying the `DFBETA` option in the `output` statement (returning Figure 81).

- We can find the influence of individuals[23] by measuring the difference in the fitted regression coefficients by deleting an individual (and all associated observations) from the data and re-fitting the model.

This can easily be done in SAS by specifying the `DFBETAC` option in the output statement (Figure 82).

[22] $$\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[\boldsymbol{it}]} \approx DFBETA_{it} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}_i'\mathbf{W}_i^{\frac{1}{2}}(1-h_i)^{-\frac{1}{2}}r_{pi}$$
, where h_i is the i-th diagonal of the hat matrix.

[23] $$DFBETACS_i = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}_i'(\mathbf{W}_{ei}^{-1} - \mathbf{Q}_i)^{-1}\mathbf{E}_i$$

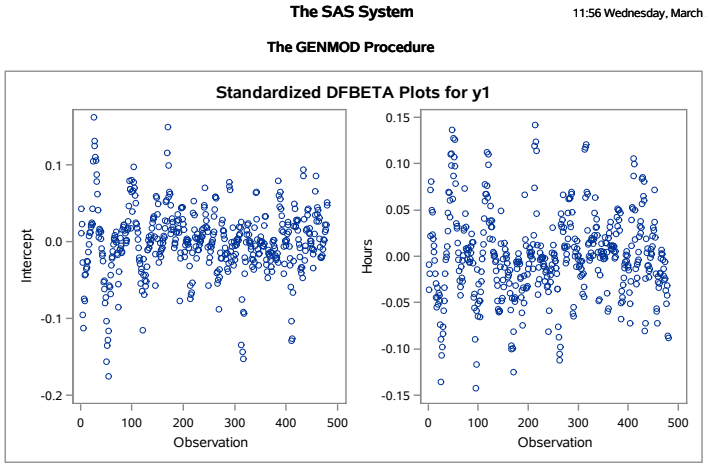


Figure 81: SAS Output for a GEE Model - Independent working correlation matrix.

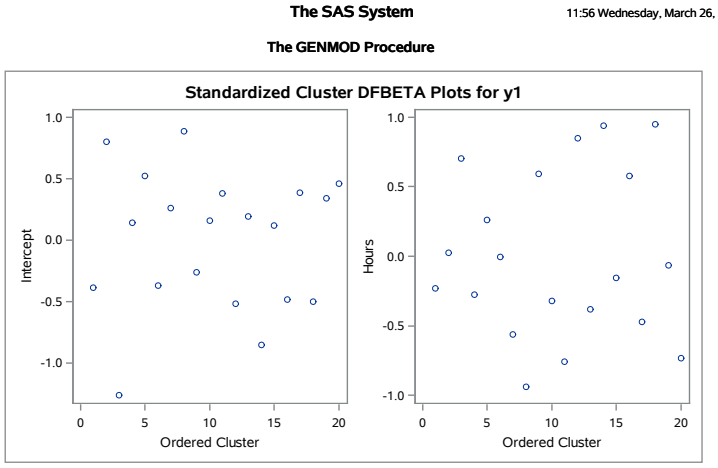


Figure 82: SAS Output for a GEE Model - Independent working correlation matrix.

We can also use Cook's distance measures to assess points which are outlying in their outcome.

Cook's distance is a scaled measure of the distance between the coefficient vectors when the k-th group of observations is deleted from the analysis.

- Cooks distance values based on leaving out single-observations[24] can be obtained by specifying the `COOKSD` option in the `OUTPUT` statement (returning one of the plots in Figure 79).
- Cooks distance, calculated by leaving out entire subjects/panels[25] are produced when you specify the `CLUSTERCOOKSD` option in the `OUTPUT` statement (returning one of the plots in Figure 83).

The independence working model doesn't appear to have any wildly large Cook's distance values (Figure 79), and this is also the case for the cluster-level Cook's distance plot (Figure 83).

To alleviate any concerns about any influential individuals, one or more individuals can be removed from the data, the model re-fitted and the results compared.

[24] $Cook_{it} = \frac{\mathbf{S}_{it}^2 \mathbf{Q}_{it}}{p\hat{\phi}(\mathbf{W}_{it}^{-1} - \mathbf{Q}_{it})^2}$ where $\mathbf{Q} = \mathbf{X}(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T$ and $\mathbf{H} = \mathbf{Q}\mathbf{W}$

[25] $Cook_i = \frac{1}{p\hat{\phi}}\mathbf{S}_i^T(\mathbf{W}_i^{-1} - \mathbf{Q}_i)^{-1}\mathbf{Q}_i(\mathbf{W}_i^{-1} - \mathbf{Q}_i)^{-1}\mathbf{S}_i$

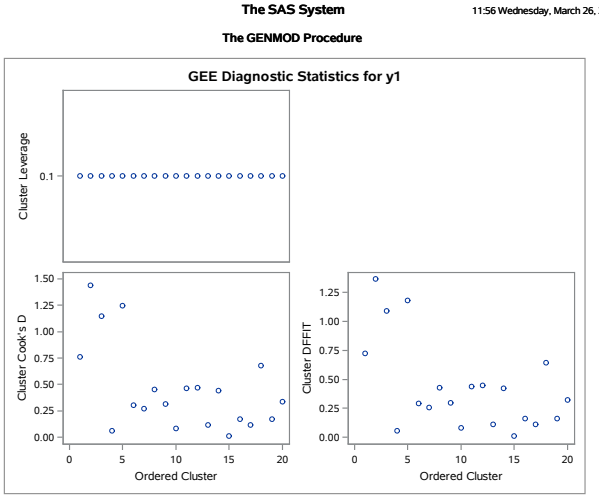


Figure 83: SAS Output for a GEE Model - Independent working correlation matrix.

4.3.2 Assessing predictive power

The predictive power of a model can be assessed using an R^2 value or concordance correlation.

An extension of the R^2 measure is calculated using:

$$R^2_{MARG} = 1 - \frac{\sum_{i=1}^{s}\sum_{t=1}^{n_i}(Y_{it} - \hat{Y}_{it})^2}{\sum_{i=1}^{s}\sum_{t=1}^{n_i}(Y_{it} - \bar{Y})^2}$$

but the concordance correlation has also been developed for GEEs:

$$r_c = \frac{2\sum_{i=1}^{s}\sum_{t=1}^{n_i}(Y_{it} - \bar{Y}..)(\hat{Y}_{it} - \bar{\hat{Y}}..)}{\sum_{i=1}^{s}\sum_{t=1}^{n_i}(Y_{it} - \bar{Y}..)^2 + \sum_{i=1}^{s}\sum_{t=1}^{n_i}(\hat{Y}_{it} - \bar{\hat{Y}}..)^2}$$

While these aren't routinely provided by SAS, they can be calculated using a freely available SAS macro (GOF). For example:

```
proc genmod data=course.data PLOTS=ALL;
class subject time;
model y1=hours;
repeated subject=subject/ corrw covb modelse within=time ecovb mcovb;
output out=Results DFBETA=DFBETA COOKSD=COOKSD

PREDICTED=PREDICTED
LOWER=LOWER UPPER=UPPER RESCHI=RESCHI
STDRESCHI=STDRESCHI RESRAW=RESRAW CLEVERAGE=CLEVERAGE
CLUSTERCOOKSD=CLUSTERCOOKSD DFBETAC=_all_ ;
effectplot fit;
assess var=(hours) / resample seed=603708000;
ODS OUTPUT GEEEmpPEst=GEEEmpPEst ;
ODS OUTPUT GEENcov=GEENcov;
ODS OUTPUT GEERcov=GEERcov;
run;

%GOF(where,proc=genmod,
           parms=GEEEmpPEst,
           covparms=,
           data=results,
           subject=subject,
           fitstats=,
           omega=,
           omega_r=,
```

```
response=y_1,
pred_ind=,
pred_avg=predicted,
title=,
ref=,
opt=noprint,
printopt=print);
```

We can see here that values are reasonably high for these data (Figure 84).

The SAS System 11:56 Wednesday, March 26, 2014 **14**

R-Square Type Goodness-of-Fit Information

Results based on predicted values from SAS procedure GENMOD

MODEL FITTING INFORMATION

DESCRIPTION	VALUE
Total Observations	480
N (number of subjects)	20
Number of Total Parameters	2
Number of Fixed-Effects Regression Parameters	2
Average Model R-Square:	0.756548
Average Model Adjusted R-Square:	0.755529
Average Model Concordance Correlation:	0.861403
Average Model Adjusted Concordance Correlation:	0.860823
Conditional Model R-Square:	0.756548
Conditional Model Adjusted R-Square:	0.755529
Conditional Model Concordance Correlation:	0.861403
Conditional Model Adjusted Concordance Correlation:	0.860823

Figure 84: SAS Output from the GOF macro

4.4 Parameter Interpretation & Hypothesis Testing

Since these models are population-averaged models, parameter interpretation works in the same way as standard linear models, in this case (Figure 85):

- the intercept parameter estimate represents the expected value of the hydration score when $x_{it} = 0$ (so average hydration score for zero hours post-application: at baseline)
- the estimate for the slope parameter represents the expected change in the average hydration score for every hour post-application

To test if one of the slope coefficients (β_j) is zero, we can use a Wald test (H_0: $\hat{\beta}_j = 0$):

$$\frac{\text{estimate}}{\text{standard error}} = \frac{\hat{\beta}_j}{\sqrt{\widehat{Var}(\hat{\beta}_j)}} \sim N(0,1)$$

where $\hat{\beta}_j$ is the estimate of interest and $\widehat{Var}(\hat{\beta}_j)$ is the diagonal element of the variance-covariance matrix for the parameter estimates, which can either be model-based or using empirical standard error estimates.

Wald tests are routinely supplied as part of the `GENMOD` output. Note: We need to be careful that there are more 'panels' than parameters to estimate, if we want to use the `EMPIRICAL` option.

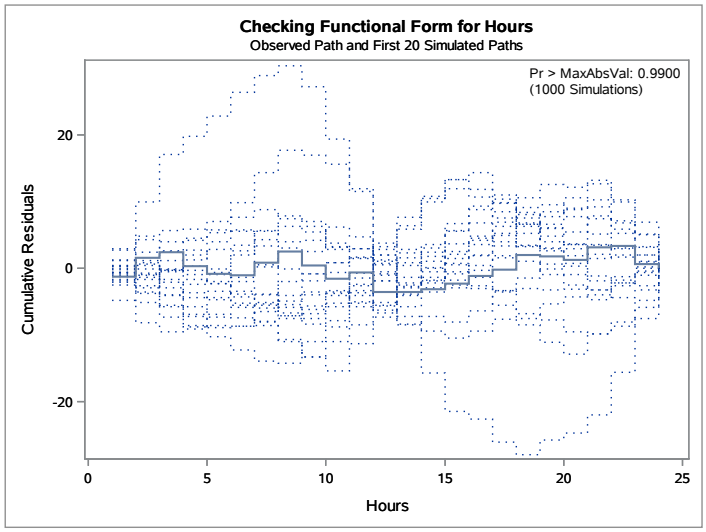
The SAS System 11:56 Wednesday, March 26, 2014 5

The GENMOD Procedure

Analysis Of GEE Parameter Estimates						
Model-Based Standard Error Estimates						
Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	51.6728	0.4726	50.7465	52.5991	109.34	<.0001
Hours	-1.2748	0.0331	-1.3396	-1.2099	-38.54	<.0001
Scale	5.0161	.	.	.	.	.

Note: The scale parameter for GEE estimation was computed as the square root of the normalized Pearson's chi-square.

Assessment Summary				
Assessment Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
Hours	3.6384	1000	603708000	0.9900



Score Statistics For Type 3 GEE Analysis			
Source	DF	Chi-Square	Pr > ChiSq
Hours	1	18.50	<.0001

Figure 85: SAS Output from the Independent working model

5 Random intercept mixed effect models

In this chapter, we are going to consider that average hydration behaviour is the same for all individuals, however the baseline hydration levels differ across individuals.

We will use **mixed effects** models in SAS and you can find more information in Littell et al. (2006).

We will also accommodate individual level-covariates which also influence the hydration levels for each person over the surveyed period (Figure 86).

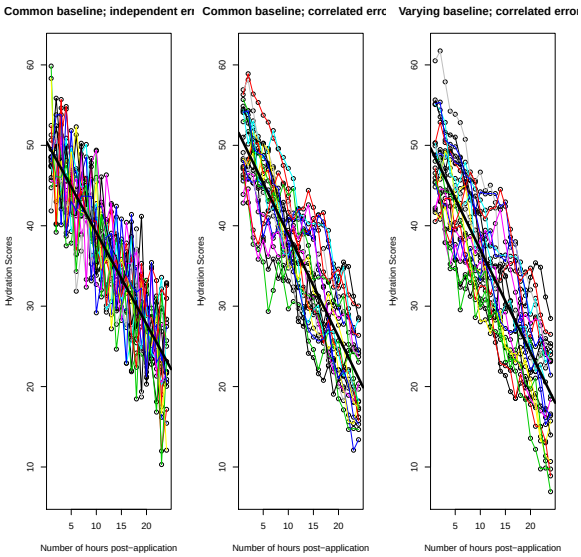


Figure 86: Raw input data for comparison. The bold lines indicate a standard linear model fit.

As is usual, the covariate information which describes why baseline hydration differs across individuals and why hydration levels fluctuate over time within individuals is missing from the model.

For this reason, these time-related patterns and diversity in baseline levels are unexplained and contribute to the noise component of the model. We are going to consider these features in our model.

5.1 Setting up the model

This scenario has sets of 24 observations ($n_i = 24$) collected from 20 individuals ($s = 20$) over time (hourly for a 24 hour period, including a baseline measurement); $N = 480$.

In this case, the baseline hydration (intercept term) has some mean level common to all individuals (the familiar β_0; note this is without a subscript i),

but each individual is assigned it's own contribution to the response (u_{0i}); note the subscript i which means the value can change for each individual, i:

$$y_{it} = \beta_0 + u_{0i} + \beta_1 x_{1it} + e_{it}$$

As before, the i indexes the individuals ($i = 1, ..., s$) and t indexes the time point corresponding to each observation ($t = 1, ..., n_i$); the number of observations can still vary across individuals (n_i).

Specifically:

- y_{it} is the observation at time point t for the $i - th$ subject in the data set (variable baseline hydration and correlated errors)
- x_{it} is the explanatory variable value for subject i at time t
- β_0 is the intercept parameter which is common to all individuals (the average/common intercept)
- u_{0i} is the subject-specific 'adjustment' to the baseline/intercept parameter
  - This component of the model has special properties: the u_{0i} (which vary across individuals) are deemed to vary from each other in a particular way

  - They are assumed to come from some distribution with mean zero (since the mean baseline hydration is described using β_0) and with some variance

- – A normal distribution is typically assumed for this distribution: $\mathbf{u}_0 \sim N(0, \sigma^2_{u_0})$
  - So, including this term attracts a variance component for the way baseline hydration levels vary from each other, $\sigma^2_{u_0}$, which must be estimated from the data

- β_1 is the slope parameter which is common to all individuals
- e_{it} is the error associated with subject i at time t

For this data type, we can still assume the errors to be Normal with some correlation within individuals (as seen previously).

This correlation can still be included via the block-diagonal structures described earlier.

5.2 Model Fitting in theory

The mixed effects model can be written in matrix form using:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{68}$$

where $\mathbf{y}$ is an $N \times 1$ response vector, $\mathbf{X}$ is an $N \times p$ design matrix of covariates ($+$ intercept), $\boldsymbol{\beta}$ is an $p \times 1$ vector of fixed effects coefficients and $\mathbf{e}$ is an $N \times 1$ vector of errors.

Here, $\mathbf{Z}$ is a $N \times q$ design matrix for the q random effects, and $\mathbf{u}$ is a $q \times 1$ vector of random effects parameters.

The random effects and errors have distributional properties:

$$\mathbf{u} \sim \mathbf{N}(\mathbf{0}, \mathbf{G}) \tag{69}$$

$$\mathbf{e} \sim \mathbf{N}(\mathbf{0}, \mathbf{R}) \tag{70}$$

and so the variability in the response has contributions from both the random effects and error term[26].

[26] $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$

This added variability with a variable baseline can be seen by contrasting data with a common and variable baseline (e.g. middle and right hand plot in Figure 86).

The $\mathbf{G}$ matrix is $q \times q$ (where q is the number of random effects in the model), so in this case is a single number ($\sigma^2_{u_0}$; a 1×1 matrix).

5.2.1 Maximum likelihood (ML)

The log-likelihood function for these models is based on a multivariate Normal distribution:

$$\log(L) = K - \frac{1}{2}[\log|\mathbf{V}| + (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})] \quad (71)$$

where K is a constant: $K = -\frac{1}{2}n\log(2\pi)$, and so can be ignored in the estimation process and n is the number of observations.

ML estimation for these models works as follows:

1 This procedure maximises the log likelihood (Equation 71) for the variance parameters first while treating the fixed effects as known constants.

2 Once the variance estimates are obtained, the fixed effects estimates are found by treating the variance parameters as fixed and finding the values for the $\boldsymbol{\beta}$ which maximise the log-likelihood.

This method produces variance estimates which are too small, to some degree. This bias is greatest for small samples, or for small samples relative to the number of parameters requiring estimation.

5.2.2 Restricted Maximum likelihood (REML)

REML works differently to ML by eliminating the fixed effects from the likelihood in order that it is only defined in terms of the variance parameters; the equation which finds the solution for the fixed effects appears instead[27] of $\boldsymbol{\beta}$ itself.

[27] $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$

In recognising that the fixed effects are parameters, rather than known constants, REML estimation includes an extra term[28]. This extra term (Equation 72) results in variance estimates which are unbiased (rather than too small as seen with ML).

The REML based log-likelihood function for these models is:

$$\log(L) = K - \frac{1}{2}[\log|\mathbf{V}| - \log|\mathbf{X}'\mathbf{V}\mathbf{X}|^{-1} + (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})] \quad (72)$$

The ability of REML to produce unbiased variance estimates, means REML estimation is often preferred over ML estimation.

5.3 Model Fitting in practice

We are going to use a very general fitting procedure for this data (`GLIMMIX`). We could also use the `MIXED` procedure which has slightly more functionality than the `GLIMMIX` procedure for these data, but this procedure can cope with a variety of data types (e.g. proportions/binary response types and count data).

```
proc glimmix data=course.data
PLOTS=residualpanel(conditional marginal) ic=pq ;
class subject;
model y2=hours/SOLUTION ddfm=kenwardroger cl;
random int /subject=subject cl ;

random _residual_/type=ar(1) subject=subject cl;
output out=Results
PREDICTED=pred
PREDICTED(NOBLUP)=predPA
RESIDUAL=res
RESIDUAL(NOBLUP)=resPA
PEARSON=pearson_res
PEARSON(NOBLUP)=pearson_resPA
LCL=LCL UCL=UCL
LCL(NOBLUP)=LCLPA UCL(NOBLUP)=UCLPA;
run;
```

A bit about the syntax. We are asking for:

- some residual diagnostic plots (`PLOTS=residualpanel(conditional marginal)`)

[28] $\log|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}|^{-1}$

- results and fit criteria based on restricted maximum likelihood estimation: `METHOD=RSPL` by default

- a Kenward-Roger based correction to the degrees of freedom for the hypothesis testing; more importantly this correction provides an adjustment to model standard errors to make them more realistic (more on this later). This is provided by `ddfm=kenwardroger`.

- confidence limits for fixed and random effects `cl` in the `MODEL` and `RANDOM` statements
- a random intercept term: `int` in the `RANDOM` statement

- autocorrelated errors within `subject` which follow an AR(1) process: `type=ar(1)` in the second `RANDOM` statement
- an output data set (`out=Results`) which contains:
  - the 'subject-specific' predictions (which include the random effects estimates): `PREDICTED=pred`
  - the 'population-average' predictions (which assume the random effects are zero): `PREDICTED(NOBLUP)=predPA`
  - the subject-specific residuals: `RESIDUAL=res`
  - the population-average residuals: `resPA`
  - Pearson-type residuals for the subject-specific and population averaged residuals (`pearson_res` and `pearson_resPA`) respectively
  - Upper and lower confidence limits for the subject-specific and population averaged predictions (`UCL`, `LCL`, `UCLPA` & `LCLPA`)

5.4 Model Selection

5.4.1 Nested Models

We can formally compare **nested** models using likelihood ratio tests (LRTs) using either ML or REML results (under certain conditions).

For LRTs, the reduced model must be a special case of the full model and for instance, can be obtained by imposing constraints on the full model.

For example, setting one or more covariance parameters to zero (e.g. the AR(1) correlation coefficient and/or a variance component).

$$-2\left[LogLikelihood_{FULL} - LogLikelihood_{REDUCED}\right] \sim \chi^2_{df=d} \qquad (73)$$

where d is the difference in the number of parameters between the full and reduced models.

For ML-based models (`METHOD=MSPL`) this comparison is valid for models with different fixed effects and/or covariance structures.

For REML-based models (`METHOD=RSPL`) this comparison is based on the Restricted-Likelihood values and :

- can be used to discriminate between models with different covariance structures
- **only** works for models with the same fixed effects structure and cannot be used to compare models with different $\mathbf{X}$ matrices.

5.4.2 Non-nested Models

To compare non-nested models you can use the AIC (Equation 48) or AICc statistics (Equation 51).

- These measures depend on the sample size (n) and the number of parameters (p) fitted in the model (including any variance parameters)

- The sample size issue is interesting. For correlated data the effective sample size (N) is not the same as the apparent sample size ($N = 480$ in this case)

- The effective sample size is somewhere between:
  - one per subject: i.e. perfect correlation within subjects ($N = s$)
  - the apparent sample size: i.e. independence ($N = \sum_{i=1}^{s} n_i$)

This is an area of current research, but in SAS the apparent sample size $N = 500$ is used to calculate the AICc.

By specifying `IC=PQ` in the `PROCEDURE` statement, the penalty term p contains the number of fixed effects parameters and covariance parameters regardless of whether ML or REML is used.

Eg. for the random intercept model with AR(1) errors: There are two fixed effects parameters (β_0 and β_1) and three covariance parameters (σ_{u_0}, σ_e, ρ).
The -2 Restricted log-likelihood is 2180.04 and $N = 480$:

$$AICc = 2180.04 + 2 \times 5 \times (480/(480 - 5 - 1)) \quad (74)$$
$$= 2190.167 \quad (75)$$

which is reported in the output (Figure 87).

Note the use of the AIC/AICc scores depends on whether ML or REML estimation is used:

- Under ML, the AIC/AICc statistics can be used to compare models with different fixed effects and random effects models
- Under REML, you can only use AIC/AICc scores to compare models with the same fixed effects structure

The SAS System 15:22 Monday, April 14, 2014 **2**

The GLIMMIX Procedure

Iteration History					
Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	2183.9819614	.	102.8156
1	0	4	2182.6708908	1.31107054	5.027819
2	0	7	2180.3984891	2.27240175	32.45079
3	0	3	2180.1877576	0.21073152	35.90242
4	0	2	2180.098324	0.08943358	24.84243
5	0	2	2180.0410768	0.05724717	1.374073
6	0	3	2180.0408169	0.00025988	0.009612
7	0	3	2180.0408169	0.00000002	0.000157

Convergence criterion (GCONV=1E-8) satisfied.

Fit Statistics	
-2 Res Log Likelihood	2180.04
AIC (smaller is better)	2190.04
AICC (smaller is better)	2190.17
BIC (smaller is better)	2210.89
CAIC (smaller is better)	2215.89
HQIC (smaller is better)	2198.24
Generalized Chi-Square	12797.82
Gener. Chi-Square / DF	26.77

REML information criteria are adjusted for fixed effects and covariance parameters.

Covariance Parameter Estimates			
Cov Parm	Subject	Estimate	Standard Error
Intercept	Subject	6.3710	8.1045
AR(1)	Subject	0.9010	0.03014
Residual		26.7737	8.1174

Solutions for Fixed Effects								
Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Intercept	49.6071	1.2761	42.33	38.87	<.0001	0.05	47.0325	52.1818
Hours	-1.2694	0.06912	34.01	-18.36	<.0001	0.05	-1.4098	-1.1289

Figure 87: Random intercept, AR(1) errors output fitted using REML.

- We can use REML based AICc scores to compare across different error structures (e.g. independence/AR(1)/continous AR(1)) since the fixed effects (model for the mean) are the same.

- We can fit models with and without random intercepts (by simply omitting the `RANDOM` statement) and with and without AR(1) errors (by omitting the `random _residual_` statement)

- In this case, we find the model with AR(1) errors (Model 3; Table 1) is preferable to the independent errors results (Models 1 & 2; Table 1), even if a random intercept term is included (Model 2).

Model	Fixed Effects	Variance Components	Correlation terms	AICC
1	β_0, β_1	σ_e	-	3025.64
2	β_0, β_1	σ_e, σ_{u_0}	-	2736.53
3	β_0, β_1	σ_e, σ_{u_0}	AR(1): ρ	2190.17

Table 1: Fit statistics for REML based models with different error structures and with and without a random intercept.

5.5 Parameter interpretation

The AR(1) model with a random intercept term returns the following point estimates for the fixed effects and variance/correlation terms (Figure 87):

- $\hat{\beta}_0 = 49.6071$, $\hat{\beta}_1 = -1.2694$
- $\hat{\sigma}^2_{u_0} = 6.3710$, $\hat{\rho} = 0.9010$, $\hat{\sigma}^2_e = 26.7737$

Further, since we are assuming Normal errors (and no link function) in this case, the parameter estimates have both population average and conditional/subject-specific interpretations. i.e:

- Under the model, the expected change in the response across a population of individuals for a one hour increase post-application of the product is $\hat{\beta}_1$ *and*

- Under the model, the expected change in the response for the 'average' individual and a one hour increase post-application of the product is $\hat{\beta}_1$

For example, the prediction for 10 hours post application obtained by averaging predictions over a **very** large number of random effect values from the assumed distribution (Normal(0,$\hat{\sigma}^2_e$)) gives:

```
> mean(49.6071 +rnorm(100000000,0, sqrt(6.3710))-1.2694*10)
[1]  36.91293
```

while the prediction for the average individual based on the 'mean' random effect (of zero) is:

$$\begin{aligned}\hat{y}_{it} =& \hat{\beta}_0 + \hat{u}_{0i} + \hat{\beta}_1 x_{it} && (76)\\ =& 49.6071 + 0 - 1.2694 \times 10 && (77)\\ =& 36.9131 && (78)\end{aligned}$$

In this case, a huge number of samples is required to give a population averaged value close to the subject-specific mixed model prediction: 36.9131.

This is due to the size of the estimated standard deviation ($\sqrt{6.3710}$); fewer samples from the Normal distribution are required if the variance is small:

```
> mean(49.6071 +rnorm(1000,0, sqrt(0.0063710))-1.2694*10)
[1] 36.91227
```

This coincidence between population average and subject-specific interpretations only works for Normal errors models with identity link function (i.e. no link function).

If we were, instead, working with binomial data (and a logit link) and a different set of parameters (e.g. $\hat{\beta}_0 = 0.2$, $\hat{\beta}_1 = 1.0845$), the prediction based on assuming a 'zero' random effect (the mean of the distribution) at **one hour** post application would be found using:

$$\eta_{it} = 0.2 + 0 + 1.0845 \times 1$$

and:

$$p_{it} = \frac{\exp(\eta_{it})}{1 + \exp(\eta_{it})}$$

to give a prediction of 0.783 at one hour post-application of the product.

The result is quite different, however, when you calculate the average prediction across a population of individuals (based on an estimated random intercept distribution: N$(0, \sqrt{35.9557})$).

This gives a population averaged prediction of 0.582 - quite a difference!

```
> eta_SS<-0.2 +1.0845*1
> exp(eta_SS)/(1+exp(eta_SS))
[1] 0.7832148
> eta_PA<-0.2 +rnorm(10000,0, sqrt(35.9557))+1.0845*1
> mean(exp(eta_PA)/(1+exp(eta_PA)))
[1] 0.5826408
```

5.5.1 Random effects predictions

The random effect prediction(s) for each subject in the data can be thought of as the 'best guess' for the location within the normal distribution which best reflects the difference between each subject and the population average.

The random effects predictions are found by finding the difference between the data and fitted values for that subject and adjusting these differences for the variance estimated under the model:

$$\hat{\mathbf{u}} = \mathbf{G}\mathbf{Z}'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \tag{79}$$

where, the variance has contributions from both the random effects and errors:

$$\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R} \tag{80}$$

In this case, the random effect prediction is the best guess for the difference between the average baseline hydration score and that for a particular subject.

- The random effects are 'shrunk' towards the mean:
  - The random effects predictions will be less widely spread than the coefficients obtained by fitting the effects as fixed (e.g. by fitting 'subject' as a factor variable in the model)
- The extent that shrinkage occurs depends, in part, on the size of the random effects variance:
  - If the random effects variance is **small** then the shrinkage is large since the spread of subject-specific intercepts are known to be very similar to each other and close to zero.
  - If the random effects variance is **large** then the shrinkage is small since the spread of subject-specific intercepts are known to be relatively diffuse.

- The extent that shrinkage occurs depends on the sample size within individuals (n_i):
  - Shrinkage is less for subjects with many repeated measurements (compared to subjects with fewer observations) since we have more information on subjects with larger samples

Given fixed and random effects estimates, the fitted value for the response for subject i at time t, has contributions from the population average intercept term, subject-specific intercept term and the population slope term:

$$\hat{y}_{it} = \hat{\beta}_0 + \hat{u}_{0i} + \hat{\beta}_1 x_{it} \tag{81}$$

So, the prediction at 10 hours for subject 1, is: $\hat{y}_{it} = 49.6071 - 1.4258 - 1.2694 * 10 = 35.4873$.

Subject-specific predictions for all subjects are provided in SAS using the `PREDICTED` option in the `OUTPUT` statement.

The prediction for the population average/average individual is:

$$\hat{y}_{PAit} = \hat{\beta}_0 + \hat{\beta}_1 x_{it} \tag{82}$$

So, the prediction at 10 hours for the 'average subject', is: $\hat{y}_{it} = 49.6071 - 1.2694 \times 10 = 36.9131$.

Population averaged predictions are provided in SAS using the `PREDICTED(NOBLUP)` option in the `OUTPUT` statement.

5.6 Parameter inference

The model based standard errors for the fixed and random effects are based on a formula that assumes the variance is known (and not estimated)[29] which can cause the standard errors to be too small.

This downwards bias can be small, but there is no way to know how big the bias is. This bias is particularly large when:

- the variance parameters are highly uncertain
- the ratio of the variance parameters compared with the error variance is small

[29] $var(\hat{\boldsymbol{\beta}}) = (\mathbf{X}\mathbf{V}^{-1}\mathbf{X})^{-1}$

- there is a large imbalance in the data (lots of missing values for some subjects)

- One of the suggested corrections is the Kenward-Roger adjustment[30]. This is implemented using the
  DDFM=KENWARDROGER option in the model statement.
- We can also use the 'robust' variance estimator[31] (also called the empirical sandwich estimator) which uses the variance of the residuals directly in the calculation of the standard errors.

Rather than rely on correctly specifying the correlation structure in the errors, this approach takes into account the observed covariance in the resisudals (via $cov(\mathbf{y})$[32]) and can be specified using the `EMPIRICAL` option in the procedure statement.

In this case, the standard errors for the fixed effects based on REML estimation[33] are larger than those based on ML estimation[34], and those based on the observed residual variance (using the EMPIRICAL option) are a bit smaller in both cases (Table 2).

Parameter	Estimate	Standard Error
β_0	49.6071 (49.6100) [49.6071]	1.2761 (1.2483) [1.0741]
β_1	-1.2694 (-1.2694) [-1.2694]	0.06912 (0.06785) [0.06569]

Table 2: Estimates for the fixed effects under a model with a random intercept and AR(1) errors fitted using REML[37], (ML[38]) and using the [EMPIRICAL] option.

5.6.1 Confidence intervals and Hypothesis tests

Confidence intervals for the fixed effects are calculated in the standard way:

$$\text{estimate} \pm t - \text{multiplier} \times \text{standard error}$$

and the t-multiplier is chosen using the degrees of freedom method specified by the user.

[30] Kenward, M. G. and Roger, J. H. (1997), Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood, Biometrics, 53, 983-997.

[31] $var(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}cov(\mathbf{y})\mathbf{V}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}$

[32] $cov(\mathbf{y}) = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'$

[33] with the Kenward-Roger adjustment

[34] with the Kenward-Roger adjustment

Upper and lower confidence limits for the predictions (of all types) can also be obtained using the following options in the `OUTPUT` statement: `LCL`, `LCL(NOBLUP)`, `UCL`, `UCL(NOBLUP)`.

Significance tests for the fixed effects parameters are reported as Wald tests:

$$\text{test-statistic} = \frac{\text{parameter estimate}}{\text{standard error}}$$

which compare this test statistic with a reference (t) distribution with degrees of freedom chosen by the user.

Here, the `DDFM=KENWARDROGER` option in the `MODEL` statement is used since it also provides better variance estimates.

In this case:

```
> -1.2694/0.06912
[1] -18.36516
> 2*pt(-18.36516,df=34.01)
[1] 3.159459e-19
```

5.7 Model Assessment

5.7.1 Assessing predictive power

As for GLMs, we can assess the fit of the model by comparing observed versus fitted values and by checking the distributional assumptions about model residuals. It is more difficult to check the assumptions about the random effect.

Plotting the observed response data versus the population average and subject-specific fitted values can be done using the output data set (`work.Results` in this case).

Model residuals (at both levels: r_{it} and r_{PAit}) can also be easily obtained using the `RESIDUAL` and `RESIDUAL(NOBLUP)` options in the `OUTPUT` statement:

$$r_{it} = y_{it} - \hat{y}_{it} \tag{83}$$

and

$$r_{PAit} = y_{it} - \hat{y}_{PAit} \tag{84}$$

5.7.2 Residual analysis

Graphical analysis using the subject-specific residuals (r_{it}) should show these are Normally distributed and patternless when considered against any covariates.

Scaled residuals[39] can be used to assess the variance and correlation structure of the model;these are adjusted for the variance and correlation structure assumed under the model and should be Normally distributed and uncorrelated[40].

- In this case, the residuals appear Normal as evidenced by the histogram, boxplot, and quantile-quantile plot (Figure 88).

- We also note that the subject-specific residuals are smaller on average than the population-level residuals (Figures 88 & 89) since they accommodate the individual response data more closely.

Scaled residuals (Figure 90 and 91) are relatively new and can only be obtained using the `PLOTS` option as a part of the `MIXED` procedure in `SAS`:

```
ods graphics on;
proc mixed data=course.data PLOTS=VCIRYPANEL(UNPACK)
PLOTS=RESIDUALPANEL(UNPACK);
class subject;
model y2=hours/SOLUTION  cl influence
residual vciry;
random int /subject=subject  cl;
repeated /type=ar(1) subject=subject;
run;
ods graphics off;
```

[39] http://www2.sas.com/proceedings/sugi29/189-29.pdf
[40] approximately, since the variance is unknown and estimated

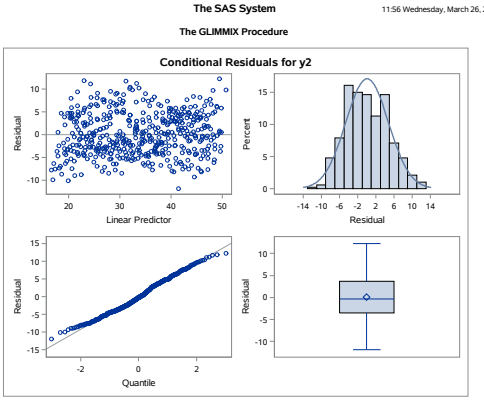


Figure 88: Graphical analysis of subject-specific residuals

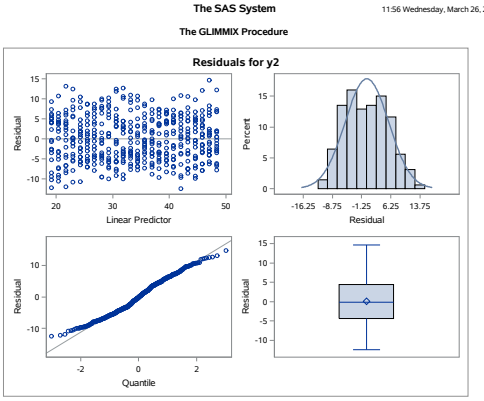


Figure 89: Graphical analysis of population-level residuals

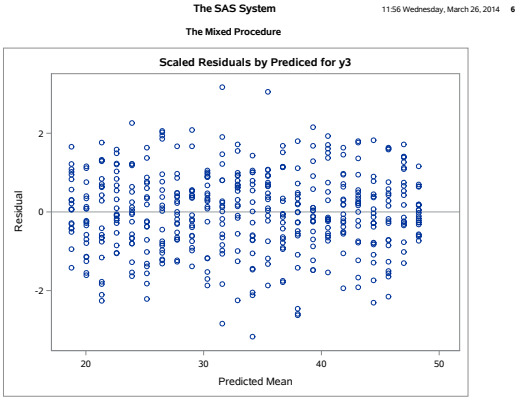


Figure 90: Scaled residuals for the random intercept model with AR(1) errors.

Checking that the random effects predictions come from a Normal distribution is more difficult, but it can help to plot the random effects predictions (Figure 92).

For normal errors models (such as this), only the variance estimates are typically affected (while the fixed effects estimates are unbiased) if the random effects are non-normal.

5.7.3 Influence diagnostics

Checking for influential subjects in the model works in a similar way to standard linear models:

1. Fit a model to the data and obtain estimates of all parameters
2. Remove one or more data points from the analysis and compute updated parameter estimates

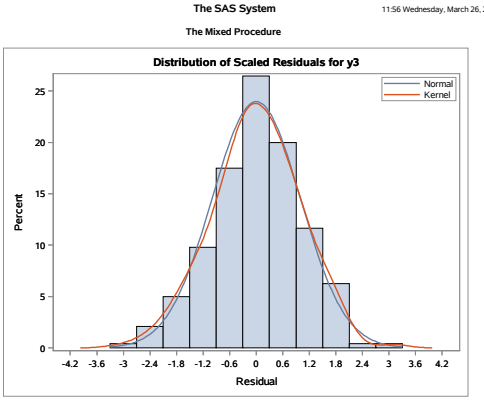


Figure 91: Scaled residuals for the random intercept model with AR(1) errors.

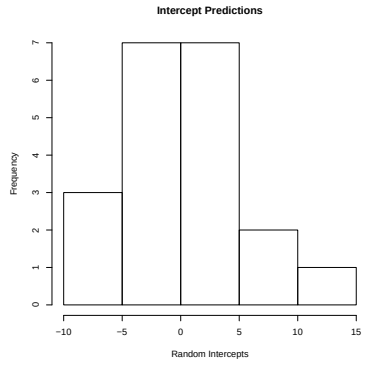


Figure 92: Histogram showing random effects predictions for the intercept.

3. Contrast model results based on full and reduced data estimates

Overall influence is measured using likelihood distance or likelihood displacement. This gives the amount the (restricted) log-likelihood value based on the full data changes if the reduced data parameter estimates[41] are used instead.

In this case, one observation (observation 398) seems particularly influential (Figure 93) but it is important to find out **how** this observation influences model results (if at all).

- If they largely affect the estimates of the fixed effects the Cook's D (`D`) statistic will be relatively large.
  - This is not the case for any of the observations; there are no particuarly large values for the Cook's D statistic (Figure 94).
- If observations unduely influence the precision of the fixed effects, the `COVRATIO` statistic will be notably different from one:
  - In this case, the observation 398 has a relatively low `COVRATIO` value (Figure 94).
- If observations unduly influence the fitted values, then the `PRESS` residual and `DFFITS` scores will be relatively large:
  - In this case, observation 398 does not show a particularly large `PRESS` residual or `DFFITS` score (Figure 95).

Even better (since these have a reference (t) distribution), the externally studentized residuals help us identify unusual observations.

Here, values greater than ± 2 signal a problem and observation 398 has a value of just 0.5414 (Figure 95).

[41] (rather than the full data parameter estimates)

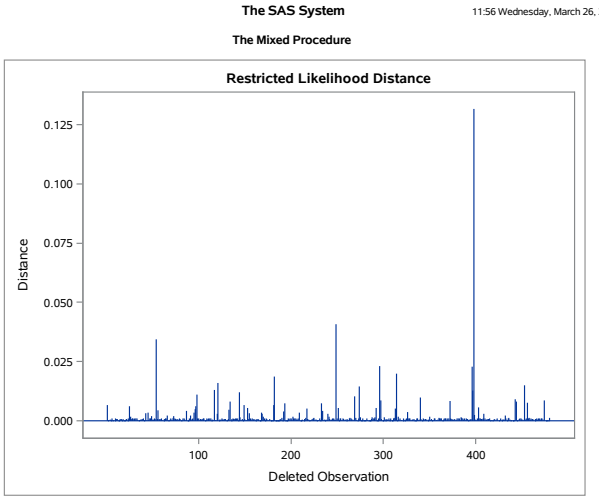


Figure 93: Overall influence values for the random intercept model with AR(1) errors.

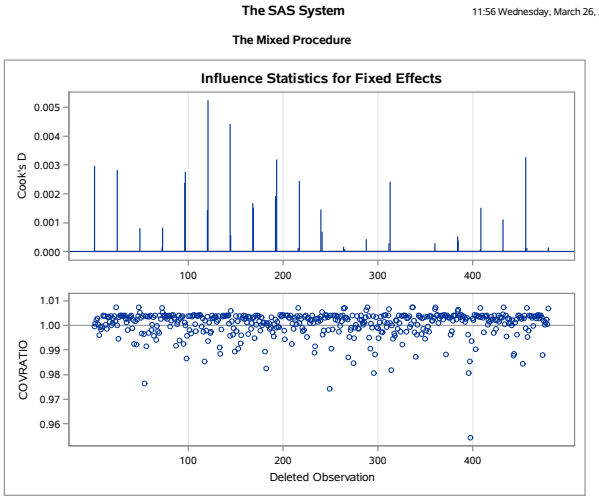


Figure 94: Overall influence values for the random intercept model with AR(1) errors.

The SAS System 11:56 Wednesday, March 26, 2014 36

The Mixed Procedure

Influence Diagnostics

Deleted Obs. Index	Observed Value	Predicted Value	Residual	Leverage	PRESS Residual	Internally Studentized Residual	RMSE without deleted obs	Externally Studentized Residual	Cook's D	DFFITS	COVRATIO
375	36.232837957	30.307	5.926	0.00050	5.925	0.5729	10.5587	0.5725	0.00000	-0.00021	1.0030
376	38.746117131	29.025	9.721	0.00051	9.721	0.9402	10.5588	0.9394	0.00000	0.00021	1.0031
377	39.529582885	27.743	11.787	0.00052	11.788	1.1403	10.5407	1.1414	0.00000	0.00057	0.9962
378	35.643567139	26.461	9.183	0.00053	9.182	0.8888	10.5448	0.8893	0.00000	-0.00052	0.9978
379	35.961336883	25.179	10.783	0.00054	10.783	1.0442	10.5559	1.0437	0.00000	0.00031	1.0020
380	33.826215939	23.896	9.930	0.00056	9.930	0.9622	10.5616	0.9612	0.00000	-0.00005	1.0041
381	32.070308916	22.614	9.456	0.00058	9.456	0.9170	10.5612	0.9160	0.00000	0.00010	1.0040
382	29.569373014	21.332	8.237	0.00060	8.237	0.7994	10.5617	0.7986	0.00000	-0.00003	1.0042
383	27.271176226	20.050	7.221	0.00062	7.221	0.7014	10.5571	0.7010	0.00000	-0.00029	1.0025
384	27.157898983	18.768	8.390	0.0440	8.450	0.8158	10.5581	0.8152	0.00053	0.02538	1.0060
385	55.64071566	48.258	7.383	0.0440	7.435	0.7179	10.5590	0.7173	0.00039	0.02186	1.0064
386	53.356167819	46.975	6.381	0.00062	6.381	0.6198	10.5617	0.6191	0.00000	-0.00002	1.0042
387	51.243610944	45.693	5.550	0.00060	5.550	0.5387	10.5592	0.5382	0.00000	-0.00021	1.0033
388	50.741157954	44.411	6.330	0.00058	6.330	0.6138	10.5617	0.6132	0.00000	-0.00001	1.0042
389	50.285692247	43.129	7.157	0.00056	7.157	0.6935	10.5565	0.6931	0.00000	0.00030	1.0022
390	47.501938529	41.847	5.655	0.00054	5.655	0.5477	10.5610	0.5471	0.00000	-0.00011	1.0039
391	45.608210898	40.565	5.044	0.00053	5.044	0.4882	10.5604	0.4877	0.00000	0.00014	1.0037
392	42.550958508	39.282	3.268	0.00052	3.268	0.3162	10.5546	0.3161	0.00000	-0.00033	1.0015
393	42.222163964	38.000	4.222	0.00051	4.222	0.4083	10.5576	0.4080	0.00000	0.00025	1.0026
394	39.82748166	36.718	3.109	0.00050	3.110	0.3006	10.5587	0.3004	0.00000	0.00021	1.0030
395	35.658098913	35.436	0.222	0.00050	0.223	0.0215	10.5410	0.0215	0.00000	0.00056	0.9963
396	26.8486267	34.154	-7.305	0.00050	-7.307	-0.7061	10.4993	-0.7096	0.00000	-0.00097	0.9807
397	26.081697438	32.872	-6.790	0.00050	-6.792	-0.6563	10.5120	-0.6588	0.00000	-0.00087	0.9854
398	32.491891208	31.589	0.902	0.00050	0.905	0.0872	10.4281	0.0883	0.00000	0.00143	0.9543
399	27.150220523	30.307	-3.157	0.00050	-3.158	-0.3053	10.5340	-0.3057	0.00000	-0.00065	0.9937
400	27.174622997	29.025	-1.851	0.00051	-1.850	-0.1790	10.5581	-0.1788	0.00000	0.00023	1.0028
401	25.2658558	27.743	-2.477	0.00052	-2.477	-0.2397	10.5561	-0.2395	0.00000	0.00030	1.0021
402	20.942508568	26.461	-5.518	0.00053	-5.518	-0.5341	10.5617	-0.5336	0.00000	0.00001	1.0042
403	16.53245384	25.179	-8.646	0.00054	-8.648	-0.8373	10.5251	-0.8394	0.00000	-0.00078	0.9903
404	18.282991376	23.896	-5.613	0.00056	-5.613	-0.5440	10.5615	-0.5434	0.00000	0.00006	1.0041
405	19.591163621	22.614	-3.023	0.00058	-3.023	-0.2932	10.5614	-0.2929	0.00000	-0.00007	1.0041
406	21.441342941	21.332	0.109	0.00060	0.110	0.0106	10.5599	0.0106	0.00000	0.00018	1.0035
407	21.93865071	20.050	1.889	0.00062	1.889	0.1835	10.5601	0.1833	0.00000	0.00018	1.0036
408	21.125258328	18.768	2.357	0.0440	2.380	0.2292	10.5612	0.2290	0.00008	0.00970	1.0072

Figure 95: Overall influence values for the random intercept model with AR(1) errors.

6 Random coefficients models

We are now going to consider that both the baseline hydration level and hydration behaviour across time differs across individuals. Individual level-covariates which also influence the hydration levels for each person are also present.

As before, the covariate information which describes why baseline and time-dependent hydration behaviour differs across individuals and why hydration levels fluctuate over time within individuals is missing from the model.

For this reason, these time-related patterns and diversity in baseline levels and time-dependent behaviour are unexplained and contribute to the noise component of the model. We are going to include these features in our model.

6.1 Setting up the model

This scenario has sets of 24 observations ($n_i = 24$) collected from 20 individuals ($s = 20$) over time (hourly for a 24 hour period); $N = 480$ (Figure 96).

In this case, the baseline hydration (intercept term) has some mean level common to all individuals (the familiar β_0 and each individual is it's own contribution to the intercept and slope values (u_{0i}, u_{1i}).

$$y_{it} = \beta_0 + u_{0i} + \beta_1 x_{1it} + u_{1i} x_{1it} + e_{it} \tag{85}$$

As before, the i indexes the individuals ($i = 1, ..., s$) and t indexes the time point corresponding to each observation ($t = 1, ..., n_i$); the number of observations can still vary across individuals (n_i).

Specifically:

- y_{it} is the observation at time point t for the $i - th$ subject
- x_{it} is the explanatory variable value for subject i at time t
- β_0 is the intercept parameter which is common to all individuals (the average/common intercept)
- β_1 is the slope parameter which is common to all individuals
- e_{it} is the error associated with subject i at time t
- u_{0i} is the subject-specific 'adjustment' to the baseline/intercept parameter
- u_{1i} is the subject-specific 'adjustment' to the slope parameter

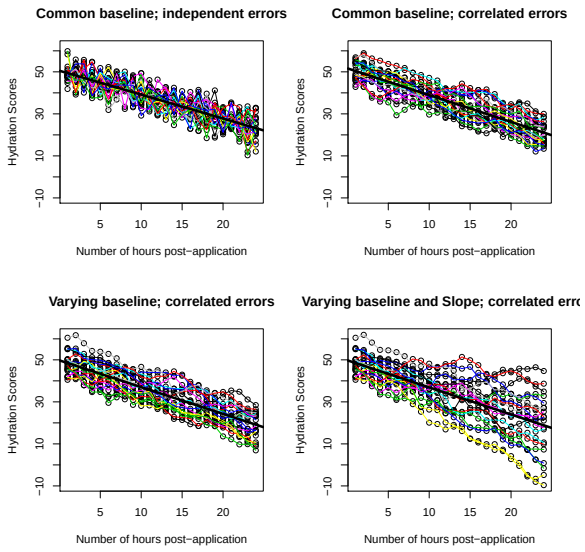


Figure 96: Raw input data for comparison.

The random effects (which vary across individuals) are assumed to come from a Normal distribution with mean zero and with some variance-covariance matrix:

$$[\mathbf{u}_0, \mathbf{u}_1] \sim \text{Normal}\left(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\sigma^2_{u_0} & \rho\sigma_{u_0}\sigma_{u_1}\\ \rho\sigma_{u_0}\sigma_{u_1} & \sigma^2_{u_1}\end{bmatrix}\right)$$

Including these random effects terms attracts two variance components ($\sigma^2_{u_0}, \sigma^2_{u_1}$) and potentially a correlation parameter, ρ, which describes the way the intercept and slope parameter values vary from (and with) each other in the population.

For this data type, we can still assume the errors to be Normal with some correlation within individuals (as seen in for the previous data type). This correlation can still be included via the block-diagonal structure described earlier.

6.2 Model Fitting and Selection

Model fitting routines are the same as for the random intercept models; ML and REML are typical choices and REML is preferred for more realistic variance estimates.

Model selection works in the same way as for random intercept models. i.e. REML and ML-based fit criteria need to be used appropriately here also:

- REML-based fit criteria can only used to discriminate between models with different random effects/variance components (the fixed effects part of the models being compared must be the same)

- ML-based fit criteria can be used to compare all models with each other (provided the input/response data is the same)

For these models, we also need to ask if the random effects estimates are independent or correlated (i.e. decide if ρ is necessary or if the off-diagonal component is zero)

In SAS, the default is to assume independence between the random effects terms - e.g. if the intercept random effect is predicted to be high/low, that doesn't lead us to suspect the slope random effect will also be high/low.

The default in SAS is:

```
random int hours/subject=subject cl;
```

This assumption might be unreasonable. For instance,

- subjects with relatively dry skin (and thus low baseline hydration scores) might respond more quickly to the product and exhibit steeper slopes.

- This would imply a negative correlation between random intercepts and slopes

- If we want to allow some unspecified correlation between the random intercept and slopes, it is best to use `type=chol`:

```
ods html close; /* close previous */
ods html; /* open new */
ods pdf file="C:\MT5757\SASOutputRandomIntSlope.pdf";
proc glimmix data=course.data1 PLOTS=ALL IC=PQ;
class subject;
model y3=hours/SOLUTION ddfm=kenwardroger cl;
random int hours/subject=subject  cl type=un;
output out=Results PREDICTED=PREDICTED STDERR=STDERR
 RESIDUAL=RESIDUAL PEARSON=PEARSON LCL=LCL UCL=UCL;
run;
ods pdf close;
```

Model	Random effects	Errors	AICC
1	None	Independent	3503.31
2	None	AR(1)	2254.73
3	Slope	AR(1)	2223.08
4	Intercept	AR(1)	2254.73
5	Intercept & Slope (Indep)	Independent	2582.32
6	Intercept & Slope (Indep)	AR(1)	2223.08
7	Intercept & Slope (Un)	Independent	2575.46
8	Intercept & Slope (Un)	AR(1)	2226.56

Table 3: Fit statistics for the various models fitted to this response. These were all fitted using REML and the penalty term includes both fixed effects and variance components.

Various models were fitted to this data (with the same fixed effects models) including those with and without random intercept and slope terms (Table 3).

Additionally, the random effects were specified to be independent (Indep) or unstructured (Un) (see also Table 3).

From these results we can see:

- the residuals look to be autocorrelated, even when intercept and slope random effects are fitted in the model.
- The model with a random slope (but a fixed intercept) looks to fit best if AR(1) errors are fitted; Table 3.
- The random effects appear to be correlated if independent errors are assumed; compare the AICc scores for Models 5 and 7 (Table 3)
- The random effects structure doesn't appear to justify the correlation term however if AR(1) errors are assumed; compare the AICc scores for Models 6 and 8 (Table 3)
- While the random effects variance component estimates affect model standard errors, the differences are slight in this case; the AR(1) errors seem to have the biggest impact on model-based SEs

6.3 Parameter Interpretation, Inference and Prediction

Since our model still has an identity link function, the parameters have both population average and subject specific interpretations.

Additionally, Wald-based tests are routine for models with both random intercepts and slopes are routinely outputted.

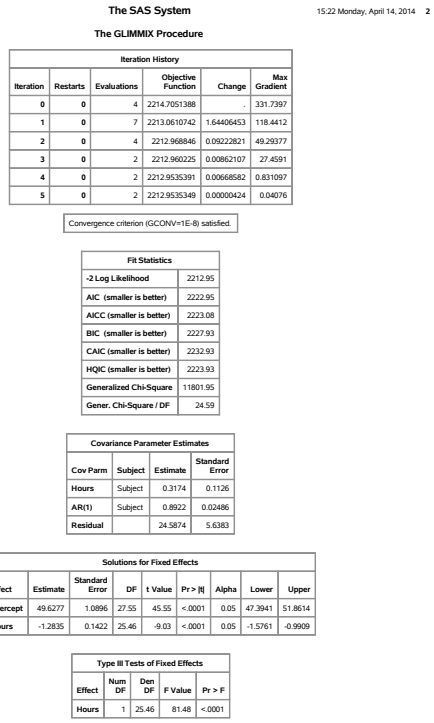
We can obtain predictions for the population average (Equation 86) and subject-specific predictions (based on the estimated random effects for each subject; Equation 87).

$$\hat{y}_{PAit} = \hat{\beta}_0 + \hat{\beta}_1 x_{1it} \tag{86}$$

$$\hat{y}_{SSit} = \hat{\beta}_0 + \hat{u}_{0i} + \hat{\beta}_1 x_{1it} + \hat{u}_{1i} x_{1it} \tag{87}$$

The subject-specific predictions will track the observed data much more closely than the population averaged predictions and the fitted subject-specific lines have variable intercept and slope since the starting point and gradient changes (potentially) for each individual.

SAS also helpfully provides the upper and lower limits for the 95% confidence intervals for the subject-specific estimates and population average line.



The SAS System 15:22 Monday, April 14, 2014 2

The GLIMMIX Procedure

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	2214.7051388	.	331.7397
1	0	7	2213.0610742	1.64406453	118.4412
2	0	4	2212.968846	0.09222821	49.29377
3	0	2	2212.960225	0.00862107	27.4591
4	0	2	2212.9535391	0.00668582	0.831097
5	0	2	2212.9535349	0.00000424	0.04076

Convergence criterion (GCONV=1E-8) satisfied.

Fit Statistics

-2 Log Likelihood	2212.95
AIC (smaller is better)	2222.95
AICC (smaller is better)	2223.08
BIC (smaller is better)	2227.93
CAIC (smaller is better)	2232.93
HQIC (smaller is better)	2223.93
Generalized Chi-Square	11801.95
Gener. Chi-Square / DF	24.59

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Hours	Subject	0.3174	0.1126
AR(1)	Subject	0.8922	0.02486
Residual		24.5874	5.6383

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Intercept	49.6277	1.0896	27.55	45.55	<.0001	0.05	47.3941	51.8614
Hours	-1.2835	0.1422	25.46	-9.03	<.0001	0.05	-1.5761	-0.9909

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Hours	1	25.46	81.48	<.0001

Figure 97: GLIMMIX output for the random slope model (fixed intercept) with AR(1) errors.

The GLIMMIX Procedure

Solution for Random Effects

Effect	Subject	Estimate	Std Err Pred	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Hours	Subject 1	0.8962	0.2221	42.89	4.04	0.0002	0.05	0.4483	1.3442
Hours	Subject 2	-0.3386	0.2221	42.89	-1.52	0.1347	0.05	-0.7866	0.1093
Hours	Subject 3	0.4199	0.2221	42.89	1.89	0.0654	0.05	-0.02799	0.8679
Hours	Subject 4	0.2665	0.2221	42.89	1.20	0.2367	0.05	-0.1814	0.7145
Hours	Subject 5	-0.03499	0.2221	42.89	-0.16	0.8756	0.05	-0.4829	0.4130
Hours	Subject 6	0.2095	0.2221	42.89	0.94	0.3509	0.05	-0.2385	0.6574
Hours	Subject 7	-0.8940	0.2221	42.89	-4.03	0.0002	0.05	-1.3419	-0.4461
Hours	Subject 8	0.2707	0.2221	42.89	1.22	0.2295	0.05	-0.1772	0.7187
Hours	Subject 9	0.5962	0.2221	42.89	2.68	0.0103	0.05	0.1482	1.0441
Hours	Subject 10	0.1962	0.2221	42.89	0.88	0.3819	0.05	-0.2517	0.6442
Hours	Subject 11	-0.4505	0.2221	42.89	-2.03	0.0488	0.05	-0.8984	-0.00253
Hours	Subject 12	0.4522	0.2221	42.89	2.04	0.0479	0.05	0.004289	0.9002
Hours	Subject 13	-0.3240	0.2221	42.89	-1.46	0.1519	0.05	-0.7719	0.1240
Hours	Subject 14	0.009953	0.2221	42.89	0.04	0.9645	0.05	-0.4380	0.4579
Hours	Subject 15	-1.0116	0.2221	42.89	-4.55	<.0001	0.05	-1.4596	-0.5637
Hours	Subject 16	0.3137	0.2221	42.89	1.41	0.1650	0.05	-0.1342	0.7617
Hours	Subject 17	-0.01875	0.2221	42.89	-0.08	0.9331	0.05	-0.4667	0.4292
Hours	Subject 18	0.7999	0.2221	42.89	3.60	0.0008	0.05	0.3519	1.2478
Hours	Subject 19	-0.6906	0.2221	42.89	-3.11	0.0033	0.05	-1.1385	-0.2426
Hours	Subject 20	-0.6680	0.2221	42.89	-3.01	0.0044	0.05	-1.1160	-0.2201

Figure 98: GLIMMIX output for the random slope model (fixed intercept) with AR(1) errors.

6.4 Model Assessment

Internally studentized residuals are simply:

$$\frac{y_i - \hat{y}_i}{\sqrt{Var[e_i]}} \tag{88}$$

where $\hat{y}_i$ can either be marginal or conditional fitted values (depending on if they are calculated with or without the random effects predictions).

The internally studentized residuals are similar to the externally studentized residuals, except the latter are found using the error variance calculated without the set of observations of interest (e.g. those in group U):

$$\frac{y_i - \hat{y}_i}{\sqrt{Var[e_{i(U)}]}} \tag{89}$$

PRESS residuals measure the difference between the observed and fitted values which are obtained without one or more observations (those in group U):

$$r_{i(U)} = y_i - \hat{y}_{i(U)} \tag{90}$$

$$= y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{(U)} \tag{91}$$

and the PRESS statistic is the sum of the squared PRESS residuals which are in group U.

The 'leverage' is the weight of the observation in contributing to its own predicted value, and so large values indicate influential points.

The DFFITS values measures the change in predicted values due to removal of data points. If this change is standardized by the externally estimated standard error of the predicted value in the full data (i.e. calculated with the set of observations omitted):

$$\frac{y_i - y_{i(U)}}{ese(\hat{y}_i)} \tag{92}$$

Correlation can be striking in the population-level residuals but should be less obvious in the subject specific residuals. It is these subject-level residuals which are assumed to be Normally distributed with an AR(1) structure (and common variance in this case, but this can be relaxed).

In this case, the residuals appear to have come from a Normal distribution with a common variance (Figure 99 and 100)

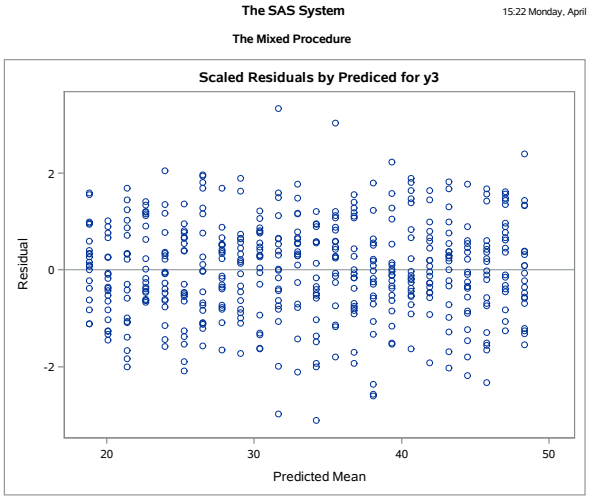


Figure 99: Scaled residuals for the random slope model with AR(1) errors.

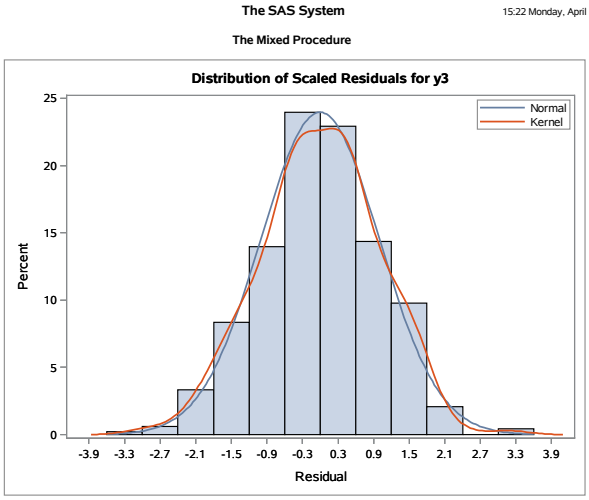


Figure 100: Histogram of scaled residuals for the random slope model with AR(1) errors.

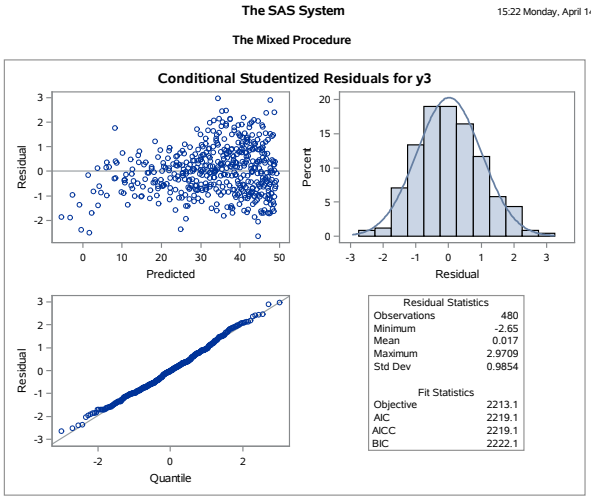


Figure 101: Conditional studentized residuals for the random slope model with AR(1) errors.

6.4.1 Influence diagnostics

Observation 398 returns a very different likelihood value when omitted (Restricted Likelihood distance: 0.1373; Figures 102 and 103). While this observation wasn't particularly distinct with respect to Cooks Distance (Figure 104) it returned a relatively low COVRATIO score (also Figure 104).

The SAS System 15:22 Monday, April 14, 2014 **37**

The Mixed Procedure

Influence Diagnostics	
Deleted Obs. Index	Restricted Likelihood Distance
375	0.0006
376	0.0005
377	0.0010
378	0.0003
379	0.0002
380	0.0010
381	0.0009
382	0.0010
383	0.0003
384	0.0009
385	0.0028
386	0.0010
387	0.0007
388	0.0010
389	0.0003
390	0.0009
391	0.0008
392	0.0001
393	0.0004
394	0.0005
395	0.0009
396	0.0251
397	0.0142
398	0.1373
399	0.0026
400	0.0005
401	0.0003
402	0.0010
403	0.0065
404	0.0010
405	0.0010
406	0.0007
407	0.0007
408	0.0009

Figure 102: Overall influence values for the random slope model with AR(1) errors.

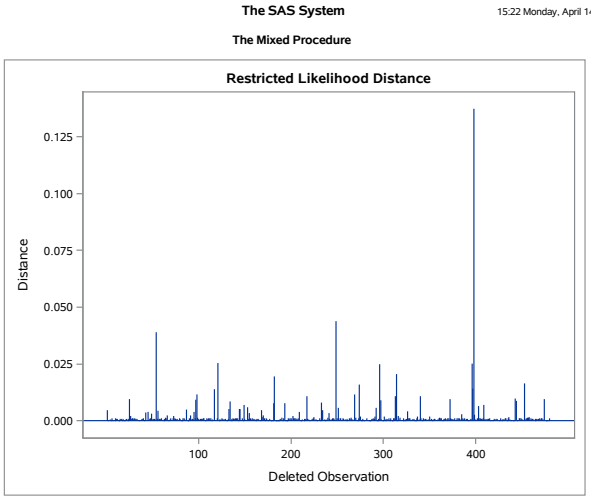


Figure 103: Overall influence values for the random slope model with AR(1) errors.

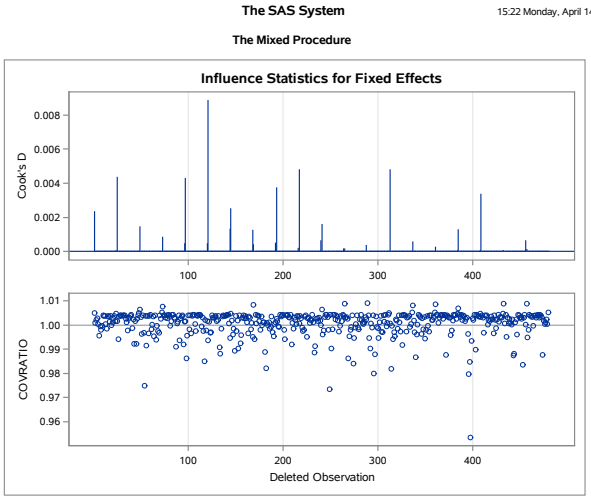


Figure 104: Overall influence values for the random slope model with AR(1) errors.

References

Fox, J. and Monette, G. (1992). Generalized collinearity diagnostics. *Journal of the American Statistical Association*, 87(417):178–183.

Hardin, J. and Hilbe, J. (2013). *Generalised Estimating Equations (2nd Ed)*. CRC Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics, second edition.

Hilbe, J. (2014). *Modelling Count Data*. Cambridge University Press.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2017). *An Introduction to Statistical Learning with Applications in R*. Springer Series in Statistics.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Oliver, S. (2006). *SAS for Mixed Models, Second Edition*. SAS Publishing.

Mackenzie, M., Scott-Hayward, L., Paxton, C., and Burt, M. (2017). Quantifying the power to detect change: methodological development and implementation using the r package mrseapower. Technical report, Report number: CREEM-13804-2016-1. Provided to the Scottish Government and Marine Scotland Science (USA/012/15).

Scott-Hayward, L., Oedekoven, C., and Mackenzie, M. (2017). MRS*ea Package (version 1.0beta) Statistical modelling of bird and cetacean distrubutions in offshore renewables development areas*.

Wald, A. and Wolfowitz, J. (1943). An exact test for randomness in the non-parametric case based on serial correlation. *Annals of Mathematical Statistics*, 14(4):378–388.

Walker, C. G., Mackenzie, M. L., Donovan, C. R., and O'Sullivan, M. J. (2011). Salsa a spatially adaptive local smoothing algorithm. *Journal of Statistical Computation and Simulation*, 81(2):179–191.

Wood, S. (2016). *mgcv: Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation*. R package version 1.8-12.

Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, second edition edition.