Long & Short Term Trends: Additive Mixed Model

David Wells

11/07/2019

Summary

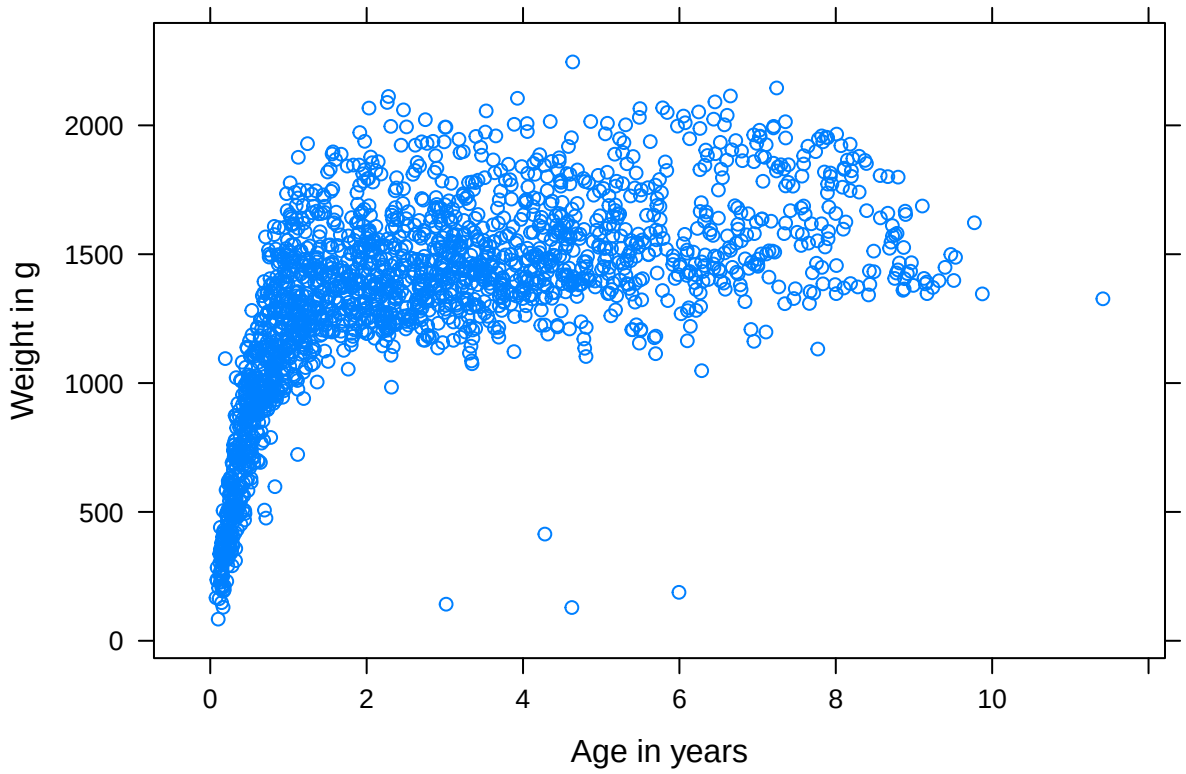
When working with temporal data it is important to capture both long and short term trends. In this series I'm modelling the relationship between the weight of banded mongooses over the course of their lives. The purpose is to compare several the performance of several model at this task.

This time the focus is on Additive Mixed Modelling. The relationship with time is modelled using a smoother (in this case two smoothers, one for long and one for short term effects). We also fit a random intercept to account for differences between individuals. Finally we also fit constraints on the residuals to account for heterogeneity and auto-correlation.

Data Exploration

First lets look at our data.

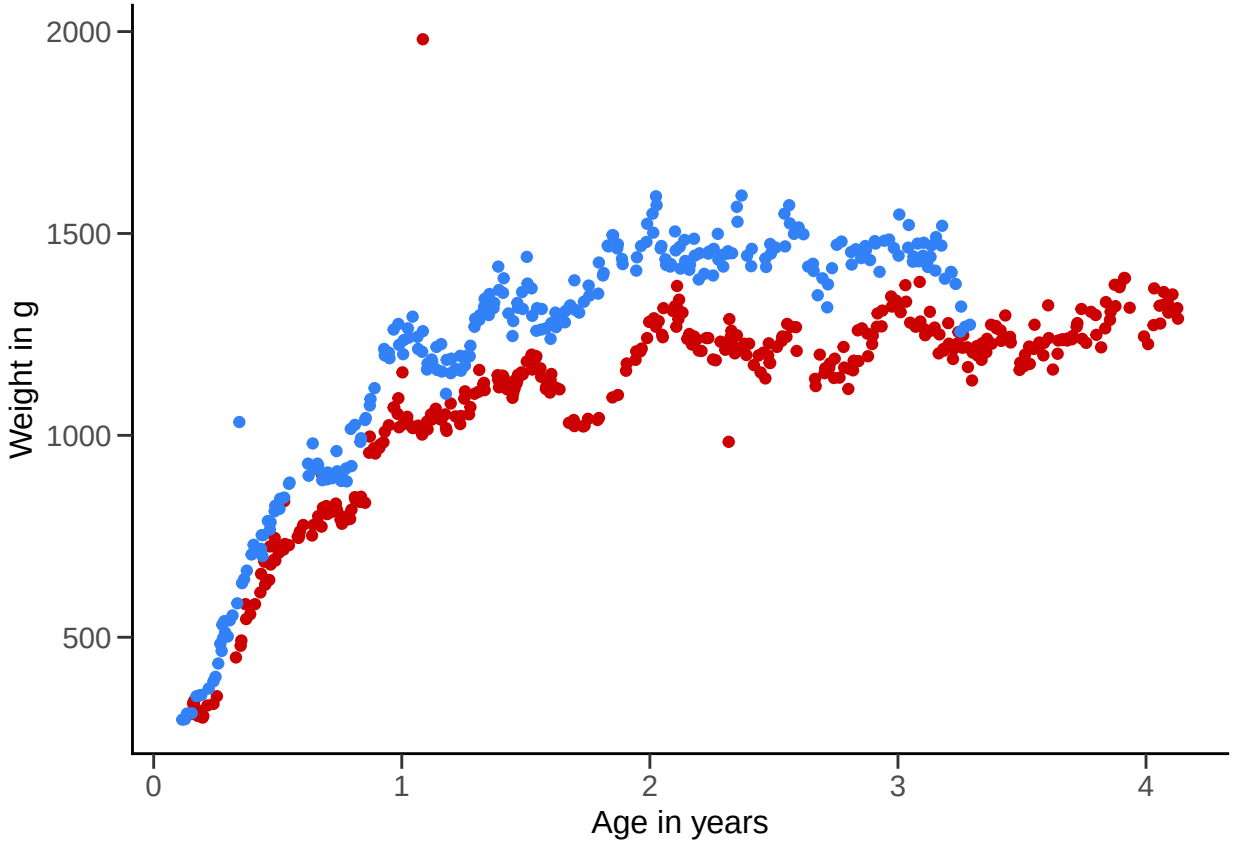
```
xyplot(train$weight ~ train$agen/365,
       xlab="Age in years", ylab="Weight in g")
```



This first plot shows us some of the complexities of our dataset, the relationship with age is clearly non-monotonic and variance in weight increases over approximately the first 2 years.

Next lets plot the data for only two mongooses to see the individual trends.

```
x<-filter(dsr,indiv %in% c("FM105","FM142"))
ggplot(x, aes(x=age/365, y=weight, colour=indiv)) +
geom_point() +
xlab("Age in years") + ylab("Weight in g") +
  #Just tidying up the graph
theme(panel.background=element_rect(fill="white",colour=NA),panel.grid.major=element_blank(),
panel.grid.minor=element_blank(),axis.line=element_line(colour="black",size=0.5,lineend="square"),
axis.ticks.length=unit(0.2,"cm"),axis.text=element_text(size=11),
axis.title=element_text(size=12))+scale_colour_manual(values=c('#cc0000','#3281f7'))+
scale_fill_manual(values=c('#cc0000','#3281f7'))+guides(colour=F)
```



From this we can see there is clearly autocorrelation within individuals and possibly some short term ocillations in weight. In this case autocorrelation is both obvious and expected, the weight on day `t`is clearly related to their weight on day `t-1`. In a less clear cut situation we could calculate the correlation or covariance at various lags to evaluate whether it is necessary to incorporate it into the residual structure.

Fitting models

Next we fit an additive model with a smoother on age and on the month of the year to capture long and short term trends respectively. For month we fit a cyclical smoother so that it loops back on itself and January follows on from December.

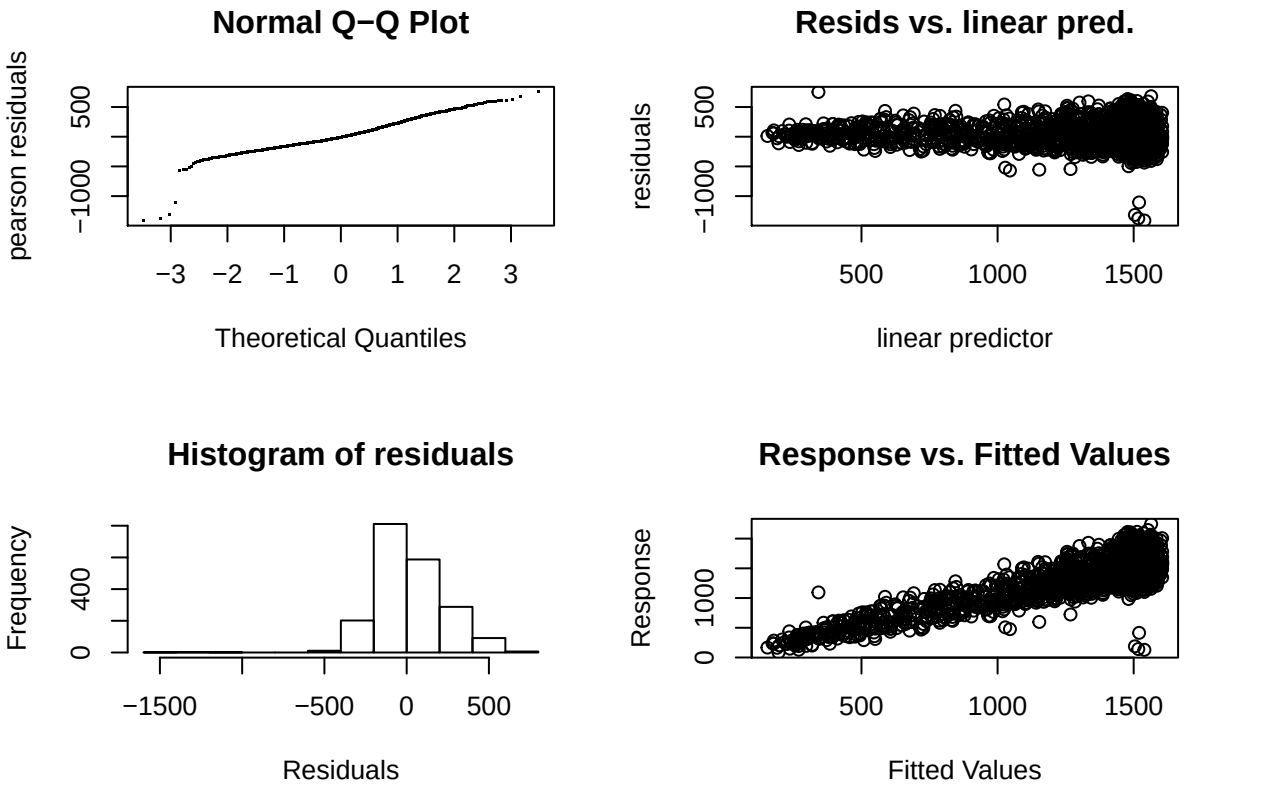
A random intercept is also included to allow individuals to differ in their mean weight and to estimate the variance which can be attributed to differences between individuals.

The final part of the model includes autocorrelation in the residuals, here we use AR-1. In this case AR-1 means that for measurements of the same individual at different times, the correlation between the

residuals depends on the time difference between measurements. Specifically, if phi is the correlation between measurements 1 unit time apart, then the correlation between measurements `n` time units apart is phi^n. We could extend the autocorrelation strucure to some thing more complex such as ARMA(p,q) but in this case it does not improve the model fit.

```
m6 <- gamm(weight~
#Smoother on age.
s(agen, k=20) +
#Cyclic smoother on month (treated as numeric).
s(monthn, bs="cc"),
#Random intercept for each individual.
random=list(indiv=~1),
#AR-1 correlation structure applied separately for each individual.
correlation = corAR1(form = ~agen|indiv), data=train)

gam.check(m6$gam, type="pearson")
```

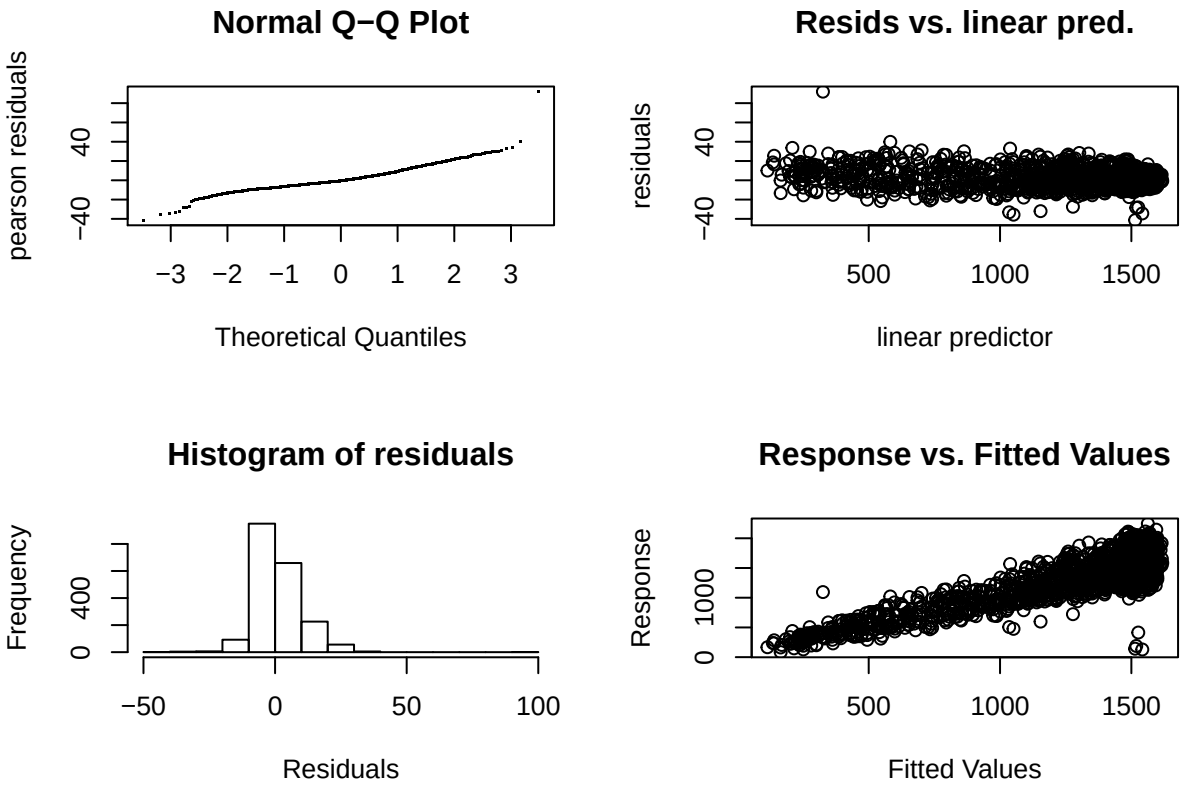


The plot produced by `gam.check()` is useful for validating the model. Both the QQ plot and histogram of residuals support the assumption of normality but the two plots on the right hand side show evidence of heterogeneity. Specifically, the plots fan out to the right indicating that our model makes less accurate predictions for larger individuals.

There are several ways we could solve this, here I choose to allow the residuals to increase linearly with age. This is done with the weights argument.

```
m7 <- gamm(weight~s(agen, k=20) + s(monthn, bs="cc"), random=list(indiv=~1),
correlation = corAR1(form = ~agen|indiv),  data=train,
#The varFixed function takes a formula proportional to the residuals.
weights = varFixed(~agen))
```

```
gam.check(m7$gam, type="pearson")
```



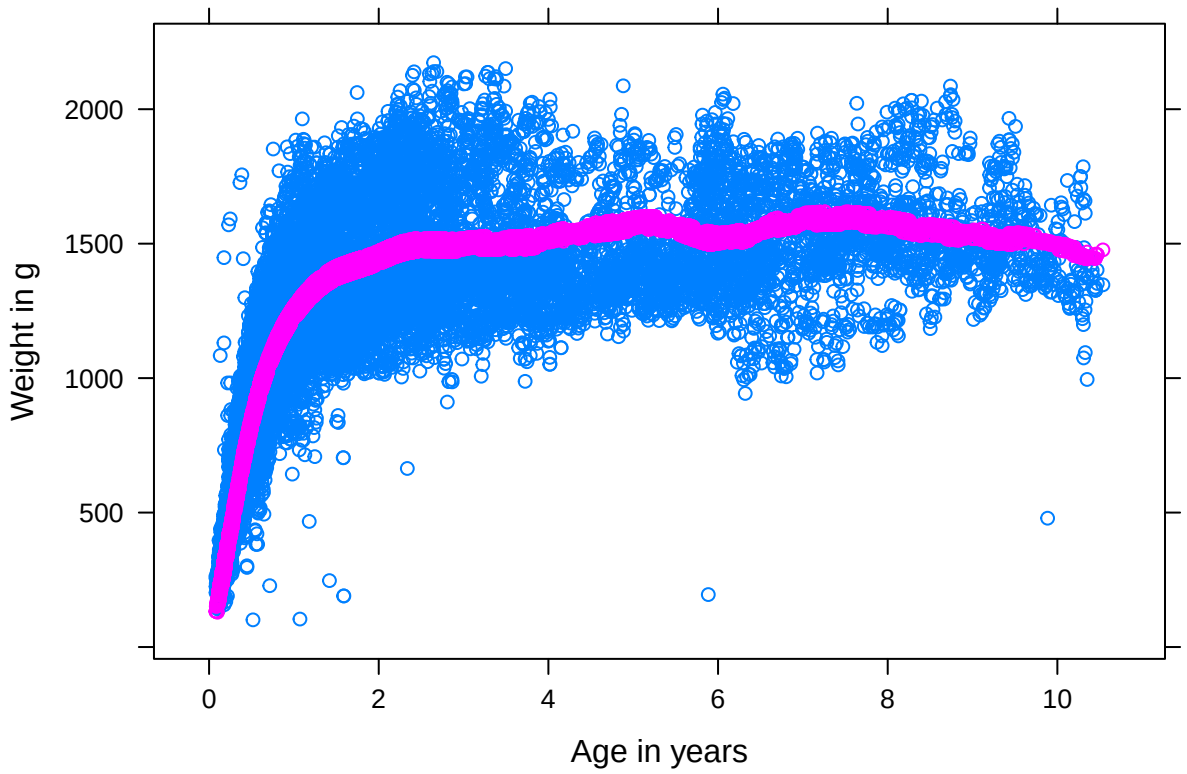
The bottom right plot, prediction vs observed, shows that the model is still less accurate for heavier weights. However, this is now accounted for as seen in the top right plot of residuals vs predictions. These are Pearson residuals which means they have been scaled by the mean/variance relationship of our model.

Predictions

We can use this fitted model to make predictions. The plot below shows the predicted weight in pink against the observed weight for new data. The vertical spread within the predictions is due to monthly changes but is hard to see because of overplotting.

```
pred <- predict(m7$gam, type="response", newdata=val)
#Predictions ignoring random effects.

xyplot(val$weight + pred ~ val$agen/365, xlab="Age in years", ylab="Weight in g")
```



The random effect has estimated an intercept for individuals in the training data and the variance of these intercepts (`sigma^2`). Therefore for new individuals we expect them to have an unknown intercept drawn from a normal distribution with `mean = 0`and `var = sigma^2`. This should be incorporated into the uncertainty of predictions for unknown individuals.

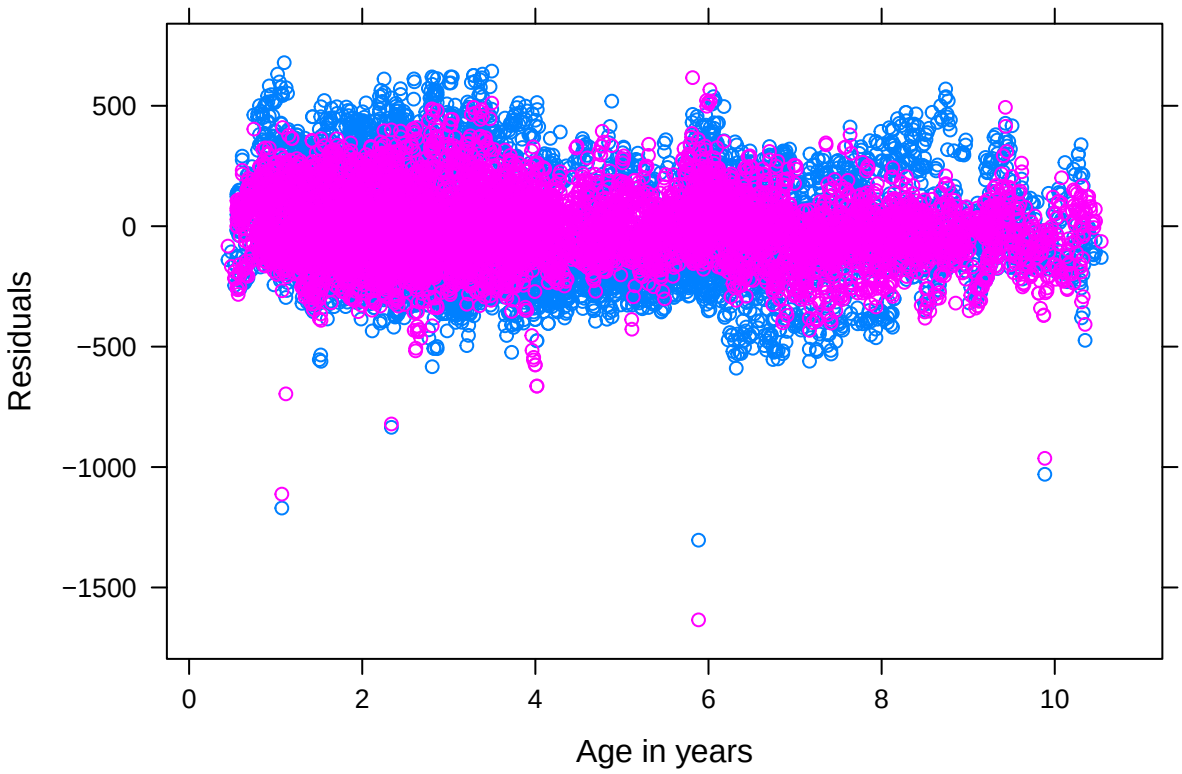
But for known individuals we can use that intercept to improve predictions. The plot below shows the raw residuals when ignoring individual identity in blue behind the improvment of including identity in pink.

```
E <- val$weight - pred

#Get the random effect intercept
ids <- paste("1/1/", val$indiv, sep="")
id_intercept <- random.effects(m7$lme)$indiv[ids,]
unknown_id <- is.na(id_intercept)
id_intercept[unknown_id]<-0

Er <- val$weight - (pred+id_intercept)

#Compare residuals with/without random intercept where known.
xyplot(E[!unknown_id] + Er[!unknown_id] ~ val$agen[!unknown_id]/365,
xlab="Age in years", ylab="Residuals")
```



Possible extensions

There are a few ways we could explore to imporve this additive mixed model further. As discussed above a more complex autocorrelation structure might be better for some datasets.

There are several alternative ways to capture short term trends. Making use of domain knowledge such as the timing of seasons can be very informative. The month or season effect could have been fit as a factor if consecutive months were drastically different. If the monthly effect altered over an individuals life a two dimensional smoother could have improved predictions `s(agen, monthn)`.

Finally for real time forcasting we could take advantage of the autocorrelation structure. To do this we take the residual from the most recent measurement we have data for $E_{t\text{-}n}$ and multiply that by phi^n (where n is the time difference) and add the result to our prediction.

For our model `phi = 0.93` which is a high correlation but after a month Phi^{30} gives a correlation of only slightly over 0.1. Therefore this is only useful when making relatively short predictions. This can be visualised by plotting the correlation matrix which is especially helpful for more complex correlation structures. This plot shows the correlation matrix for three individuals with records lasting 14, 30 and 60 days.