

# Final

*John Brandt*

*12/14/2017*

Modeling of bear weight based upon bodily measurements using mixed effects and generalized least squares modeling accounting for covariance-variance structures and within-group correlation.

Based upon predicted vs. observed, residuals, residual sum of squares, and AIC, this script determines that bear weight is modeled best by a cubic-splines transformed mixed-effect model in the form of:

```
lme(weight ~ chestGSc + lengthSc + headWSc + ageNA +
      splinepoints_chest + splinepoints_length,
      random = ~ 1 | name/age,
      weights=varExp(form=~as.vector(chestGSc + splinepoints_chest)),
      correlation=corCAR1(form = ~ age | name))
```

where:

- chestGSc = scaled chest girth
- lengthSc = scaled length
- headWSc = scaled head width
- ageNA = dummy variable, indicating whether or not age was imputed
- splinepoints\_chest = cubic splines transformation of chest girth, with knots at 7, 25, and 90 percent
- splinepoints\_length = cubic splines transformation of length, with knots at 10, 35, and 75 percent
- autocorrelation is modeled with an order of 1, with a continuous time covariate of age grouped by the name of each bear
- within-group variance is modeled using the exponential variance of scaled chest girth plus its spline transformation

## 1 Read in Data

```
require(pacman)
p_load(ggplot2, nlme, lme4, reshape2, dplyr, car)

bears <- read.fwf("Bears.full.dat", skip=20,
  col.names=c("id", "age", "month", "sex", "headL", "headW",
    "neckG", "length", "chestG", "weight", "obs", "name"),
  widths=c(9, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 10),
  strip.white=T, na.string = "")
```

Missing bear names are set to “Unnamed 1” and “Unnamed 2” and relevant columns are converted to factors

```
bears$name <- as.character(bears$name)
bears[87,12] <- "Unnamed1"
bears[88,12] <- "Unnamed2"
bears$name <- as.factor(bears$name)
bears$month <- as.factor(bears$month)
bears$sex <- as.factor(bears$sex)
attach(bears)
```

## 2 Correlations

Exploratory correlations show strong relationships between possible predictor variables and bear weight.

```
cor(weight, bears[,5:9])
```

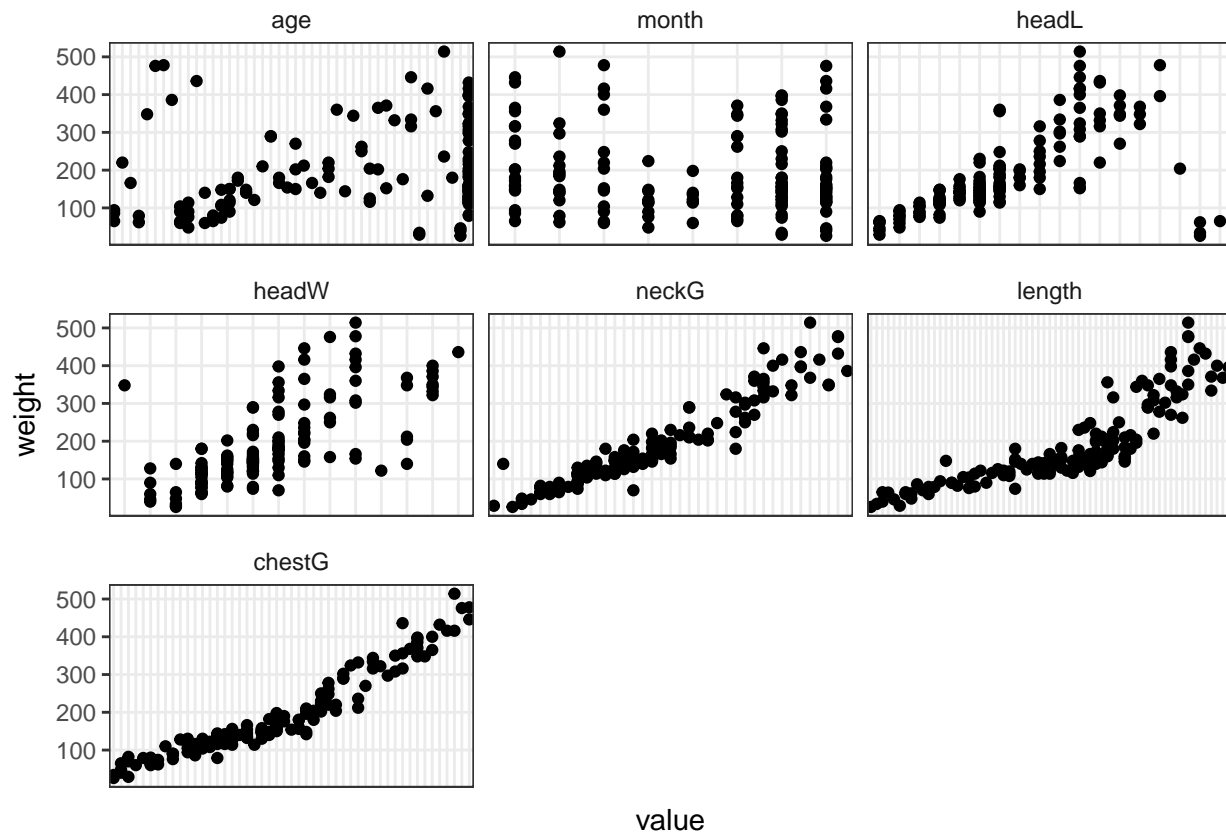
```
##           headL    headW    neckG    length    chestG
## [1,] 0.8333214 0.7556092 0.9432771 0.8746451 0.9659937
```

## 3 Initial plotting

Initial plots show nonlinear relationships for length, chest, and neck.

```
bearsplotdata <- bears[,0:12]
bearsplotdata <- bearsplotdata[,-c(1,4,11,12)]
bearsplotdata <- melt(bearsplotdata, id.vars=c("weight"))

bearsplot1 <- ggplot(data=bearsplotdata, aes(y=weight, x=value))+
  theme_bw()+
  geom_point()+
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        strip.background = element_blank(),
        panel.grid.minor = element_blank()) +
  facet_wrap(~variable, ncol=3, scales="free_x")
print(bearsplot1)
```



## 4 Transformations

Transform length, chest, and neck based upon observed relationships in the initial plots.

```
bears$lengthtransf <- 1/bears$length
bears$chestGtransf <- bears$chestG^2
bears$neckGtransf <- bears$neckG^2
```

## 5 Scale response and predictor variables

The measurements and weights that we are using in the model vary by more than an order of magnitude. In order for the model to converge, we need to make sure that the predictors and response variables have similar ranges.

All predictor variables were scaled by subtracting the mean and dividing by the standard deviation.

```
bears$headLSc <- scale(bears$headL)
bears$neckGSctransf <- scale(bears$neckGtransf)
bears$neckGSc <- scale(bears$neckG)
bears$headWSc <- scale(bears$headW)
bears$lengthSc <- scale(bears$length)
bears$lengthSctransf <- scale(bears$lengthtransf)
bears$chestGSc <- scale(bears$chestG)
bears$chestGSctransf <- scale(bears$chestGtransf)
```

## 6 Deal with missing age data

I chose a model-based imputation as outlined in Andrew Gelman’s 2012 textbook “Data Analysis Using Regression and Multilevel/Hierarchical Models”. I create a dummy variable for future models indicating whether or not the age data is imputed, and then infer the value based upon the relationship between age and head width.

I build a linear model relating age to scaled head width, resulting in an estimate of 21.532. This means that, for every 1 unit change in the standard deviation of head width, age increases by 21.532 months. I then apply this relationship to the missing age data.

As a means to check whether the imputation was successful, I compare the mean of the imputed age to the mean of the known age. I then compare the correlation between age and a variable not used to impute age (chest girth) for imputed and known age.

Age is similar between the two (43.4 vs 46.8), as are the correlations (0.734 vs. 0.731). Thus, I continue to use the imputed age data in the modeling process.

```
bears$ageNA <- 0
bears$ageNA[is.na(bears$age)] <- 1

summary(lm(bears$age[bears$ageNA==0] ~ bears$headWSc[bears$ageNA==0]))

##
## Call:
## lm(formula = bears$age[bears$ageNA == 0] ~ bears$headWSc[bears$ageNA ==
##      0])
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -63.441 -14.084  -6.891   9.156  79.171
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          45.893      2.808  16.341 < 2e-16 ***
## bears$headWSc[bears$ageNA == 0] 21.532      2.656   8.106 4.65e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.44 on 81 degrees of freedom
## Multiple R-squared:  0.4479, Adjusted R-squared:  0.441
## F-statistic: 65.7 on 1 and 81 DF,  p-value: 4.653e-12

meanage <- mean(bears$age[bears$ageNA==0])
bearschange <- bears$headWSc[bears$ageNA==1]*21.532

bears$age[bears$ageNA==1] <- meanage + bearschange
bears$age <- jitter(bears$age, amount=0.0001)

mean(bears$age[bears$ageNA==1])

## [1] 46.83521
mean(bears$age[bears$ageNA==0])

## [1] 43.43373
cor(bears$chestG[bears$ageNA==0], bears$age[bears$ageNA==0])

## [1] 0.7342368
cor(bears$chestG[bears$ageNA==1], bears$age[bears$ageNA==1])

## [1] 0.7314327
bears$agescale <- scale(bears$age)
```

## 7 Linear model

I start by building a simple linear model to get a base AIC score and model structure moving forward.

```
lm1 <- lm(weight ~ chestGSctransf + sex + neckGSctransf + lengthSctransf + headLSc
          + headWSc, data=bears)

summary(lm1)

##
## Call:
## lm(formula = weight ~ chestGSctransf + sex + neckGSctransf +
##     lengthSctransf + headLSc + headWSc, data = bears)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -58.473 -11.112  -1.404  11.803  74.150
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   192.8548     2.1479  89.788 < 2e-16 ***
## chestGSctransf 74.9734     5.4441  13.772 < 2e-16 ***
## sex2          -2.2553     4.1094  -0.549  0.5840
## neckGSctransf  31.8555     6.0351   5.278 5.03e-07 ***
## lengthSctransf -6.6563     3.6866  -1.806  0.0732 .
## headLSc       -2.0311     4.2655  -0.476  0.6347
## headWSc       -0.9657     2.9858  -0.323  0.7469
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.76 on 136 degrees of freedom
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.9647
## F-statistic: 648.2 on 6 and 136 DF,  p-value: < 2.2e-16
```

```
AIC(lm1)
```

```
## [1] 1282.124
```

## 7.1 Variable Inflation Factors

Variable inflation factors indicate that chestG and neckG are highly colinear.

```
vif(lm1)
```

```
## chestGSctransf      sex neckGSctransf lengthSctransf      headLSc
##          9.762581      1.193243      11.997357      4.476711      5.993064
##          headWSc
##          2.936599
```

Indeed, chestG and neckG have a 0.94 correlation.

```
cor(bears$chestGSctransf, bears$neckGSctransf)
```

```
##           [,1]
## [1,] 0.9406729
```

I keep chestG because it has a slightly higher correlation with weight. I remove neckG.

```
cor(bears$chestGSctransf, bears$weight)
```

```
##           [,1]
## [1,] 0.9773564
```

```
cor(bears$neckGSctransf, bears$weight)
```

```
##           [,1]
## [1,] 0.9532861
```

## 8 Mixed-effect model with power variance function

I first build a mixed-effects model with all of the potential predictor variables and no controls for heteroskedasticity or longitudinality.

I group the data by name and age, and allow the slope and intercept to vary by groups.

```
lme_init <- lme(weight ~ chestGSctransf + lengthSctransf + headWSc + headLSc,
  random = ~ 1 | name/age, data=bears)
```

Remove insignificant variable (headLSc) and begin trying various weighting choices.

I choose between the varPower and varExp variance function classes based upon visual inspections of the variance in the initial faceted plot.

```
lme.power <- lme(weight ~ chestGSctransf + lengthSctransf + headWSc,
  random = ~ 1 | name/age,
  weights=varPower(form=~as.vector(lengthSctransf)), data=bears)

lme.exp <- lme(weight ~ chestGSctransf + lengthSctransf + headWSc,
  random = ~ 1 | name/age,
  weights=varExp(form=~as.vector(lengthSctransf)), data=bears)
```

Choose the varExp function to model variance-covariance structure, based upon its lower AIC score.

```
AIC(lme.power, lme.exp)
```

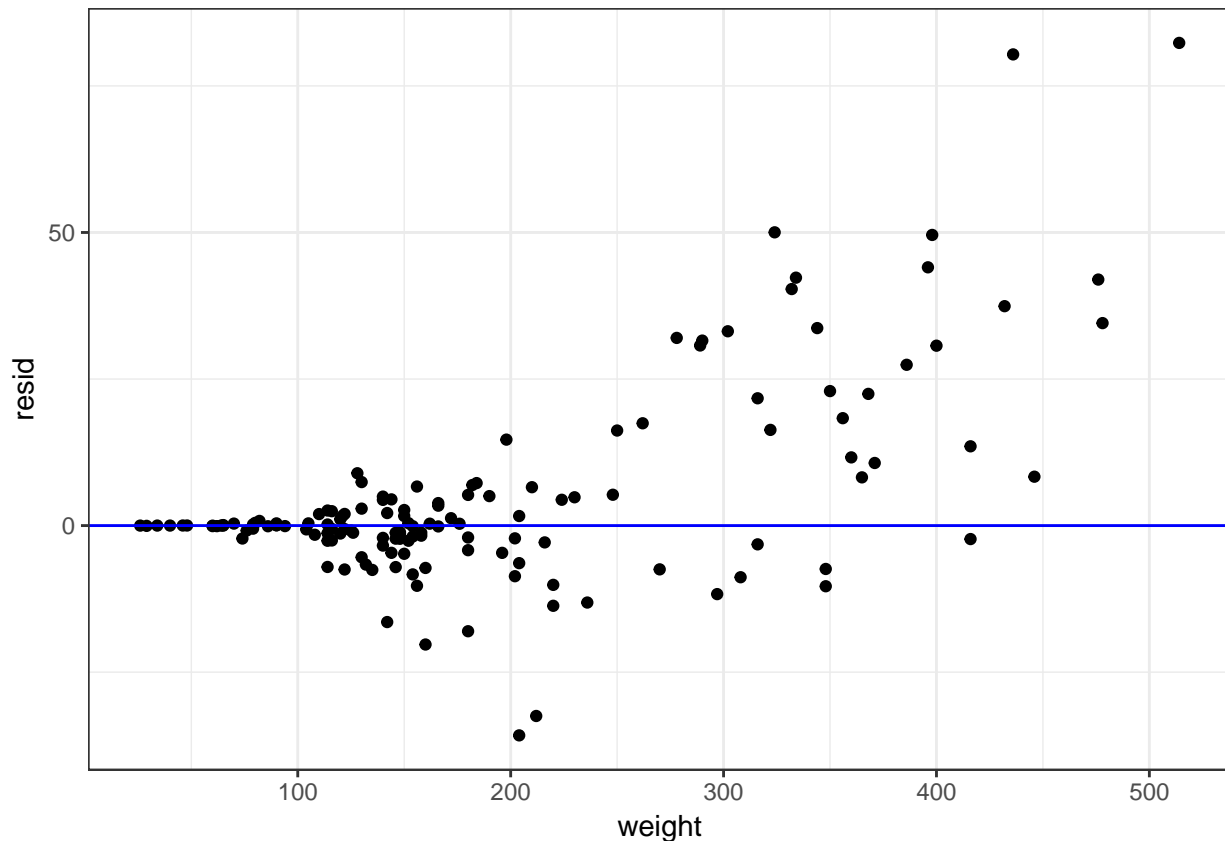
```
##           df      AIC
## lme.power  8 1296.343
## lme.exp    8 1279.278
```

## 8.1 Residuals

Residuals for the linear mixed-effect model.

```
resid <- resid(lme.exp)
residplot <- ggplot(data=bears, aes(y=resid, x=weight))+
  geom_point()+
  geom_hline(yintercept=0, color="blue")+
  theme_bw()

print(residplot)
```



## 8.2 Remove outliers

Based upon the residuals from the above linear model, I identified samples 37 and 132 as potential outliers. Sample 37 is the oldest bear by 25%, and sample 132 is the heaviest bear by nearly 10%.

There were no other reasons to remove sample 37 other than its abnormal residuals, and so it was kept in the data.

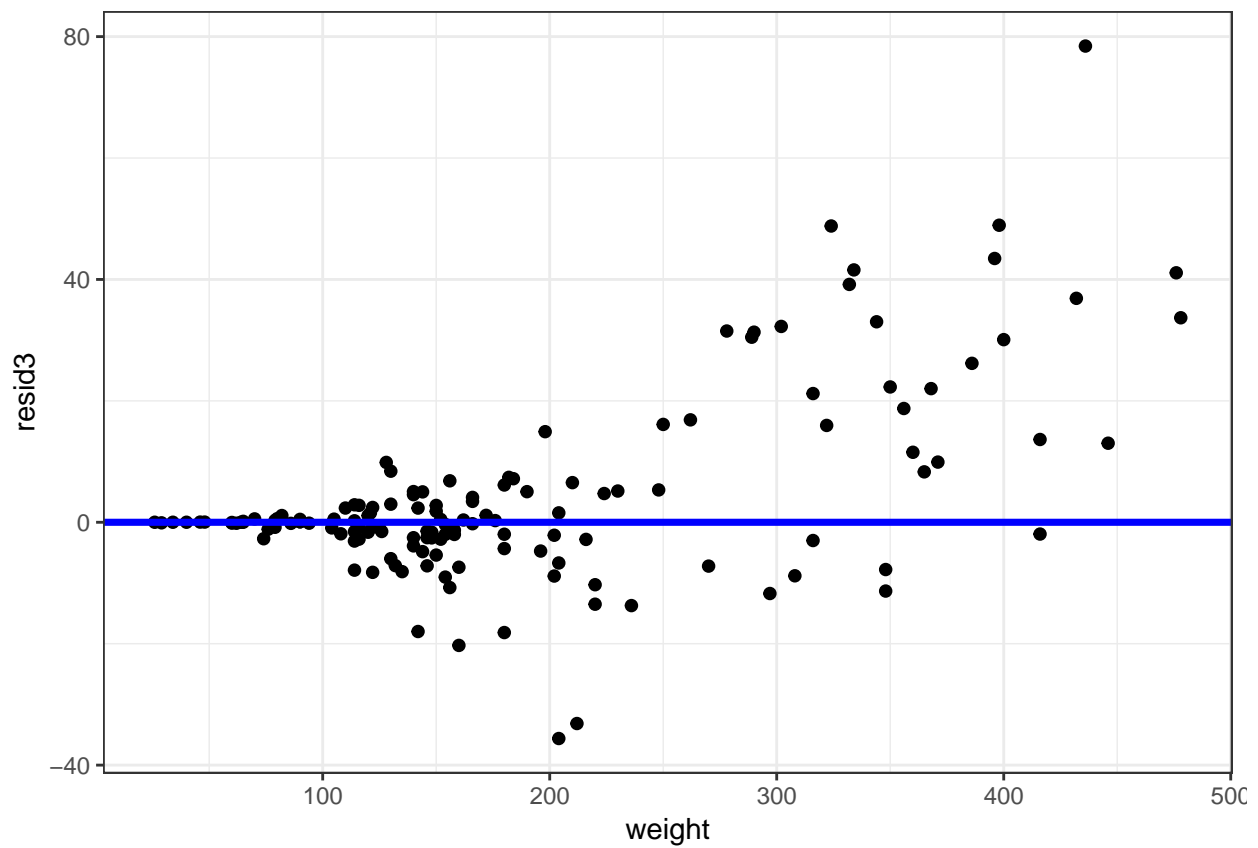
Sample 132 had been measured 10 months earlier, during which they gained 70 pounds while losing chest girth and length. Based on the unlikelihood of increasing weight by 20% while becoming smaller in size, I chose to remove sample 132.

```
bearsRM <- bears[-132,]
lme3 <- lme(weight ~ chestGSctransf + lengthSctransf + headWSc,
  random = ~ 1 |name/age, weights=varExp(form=~as.vector(lengthSctransf)),
  data=bearsRM)
```

## 8.3 Residuals

```
resid3 <- resid(lme3)
residplot3 <- ggplot(data=bearsRM, aes(y=resid3, x=weight))+
  geom_point(size=1.7)+
  geom_hline(yintercept=0, color="blue", size=1.25)+
  theme_bw()

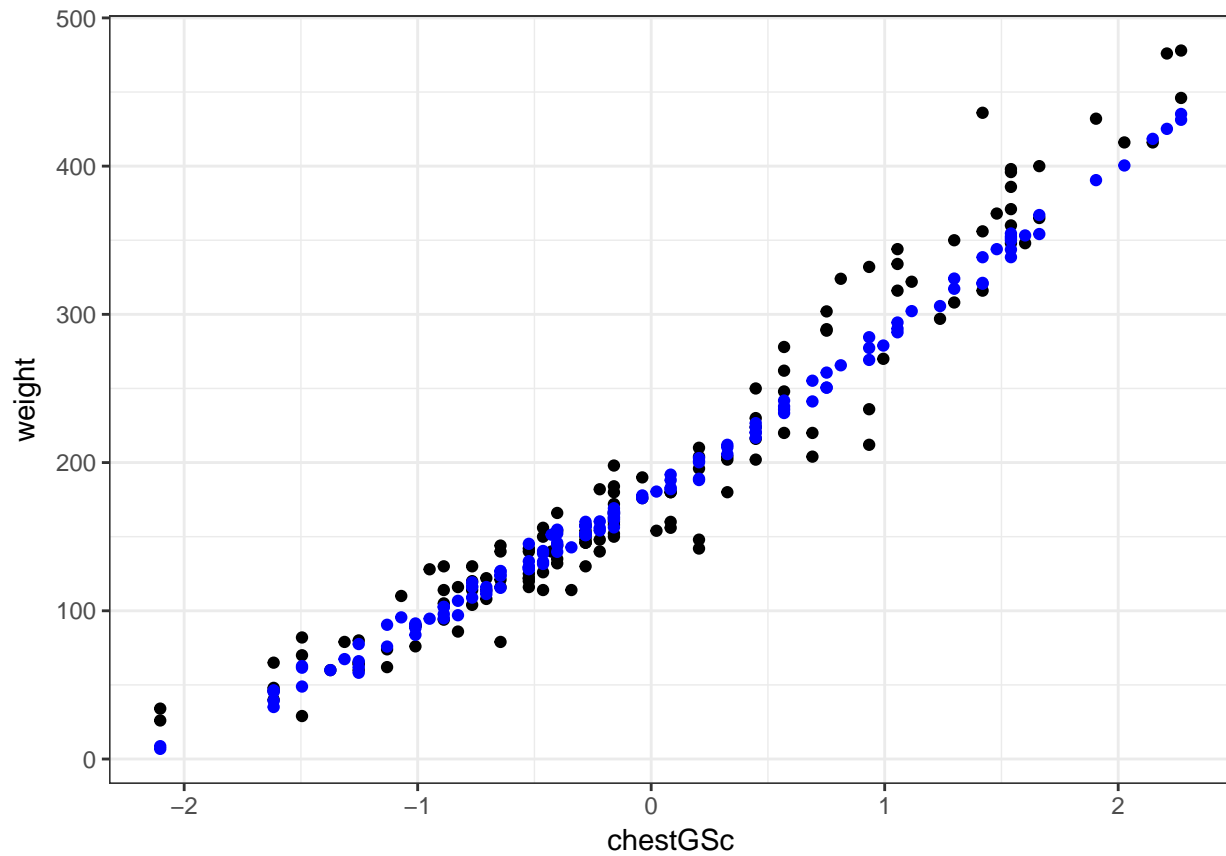
print(residplot3)
```



#### 8.4 Chest vs. weight - observed vs. predicted

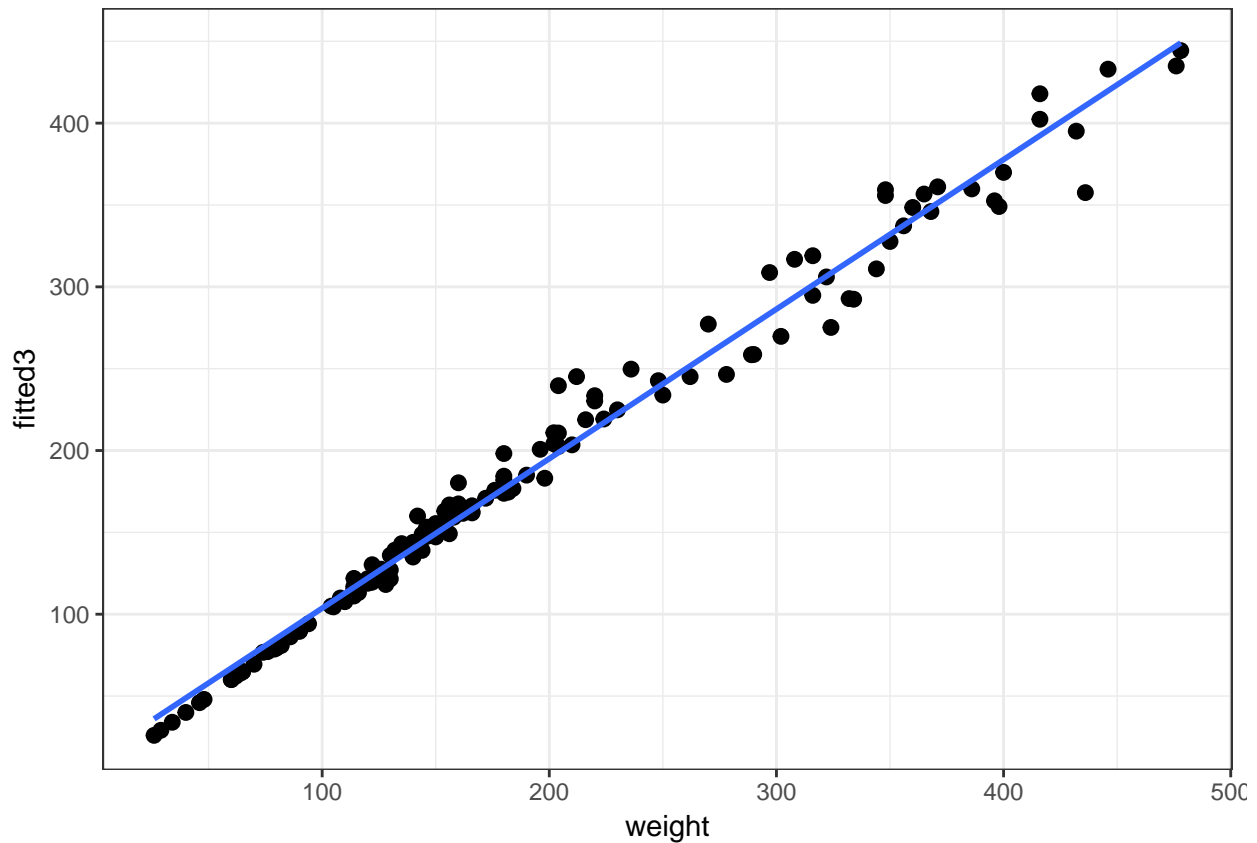
```
plot1 <- ggplot(data=bearsRM, aes(y=weight, x= chestGSc))+  
  geom_point()+  
  theme_bw()+  
  geom_point(aes(y=lme3$fitted[,1], x=chestGSc), color="blue")  
print(plot1)
```





## 8.5 Fitted vs. actual

```
fitted3 <- fitted(lme3)
fittedvsactual <- ggplot(data=bearsRM, aes(y=fitted3, x=weight))+
  theme_bw()+
  geom_point(size=2.3)+
  geom_smooth(method="lm", se=FALSE)
print(fittedvsactual)
```



## 9 Generalized least squares

Next, bear weight is modeled by generalized least squares.

Model one has no correlation structure, while models two and three do. Model two includes a dummy variable for whether age was imputed.

```
gls1 <- gls(weight ~ chestGSctransf + lengthSctransf + headWSc,
            weights=varExp(form=~as.vector(lengthSctransf)), data=bearsRM)

gls2 <- gls(weight ~ chestGSctransf + lengthSctransf + headWSc + ageNA,
            weights=varExp(form=~as.vector(lengthSctransf)),
            correlation=corCAR1(form = ~ age | name), data=bearsRM)

gls3 <- gls(weight ~ chestGSctransf + lengthSctransf + headWSc,
            weights=varExp(form=~as.vector(lengthSctransf)),
            correlation=corCAR1(form = ~age | name), data=bearsRM)
```

### 9.1 AIC

AIC suggests that model two (with correlation structure and the imputed dummy variable) is the best thus far.

```
AIC(gls1, gls2, gls3)
```

```
## Warning in AIC.default(gls1, gls2, gls3): models are not all fitted to the
```

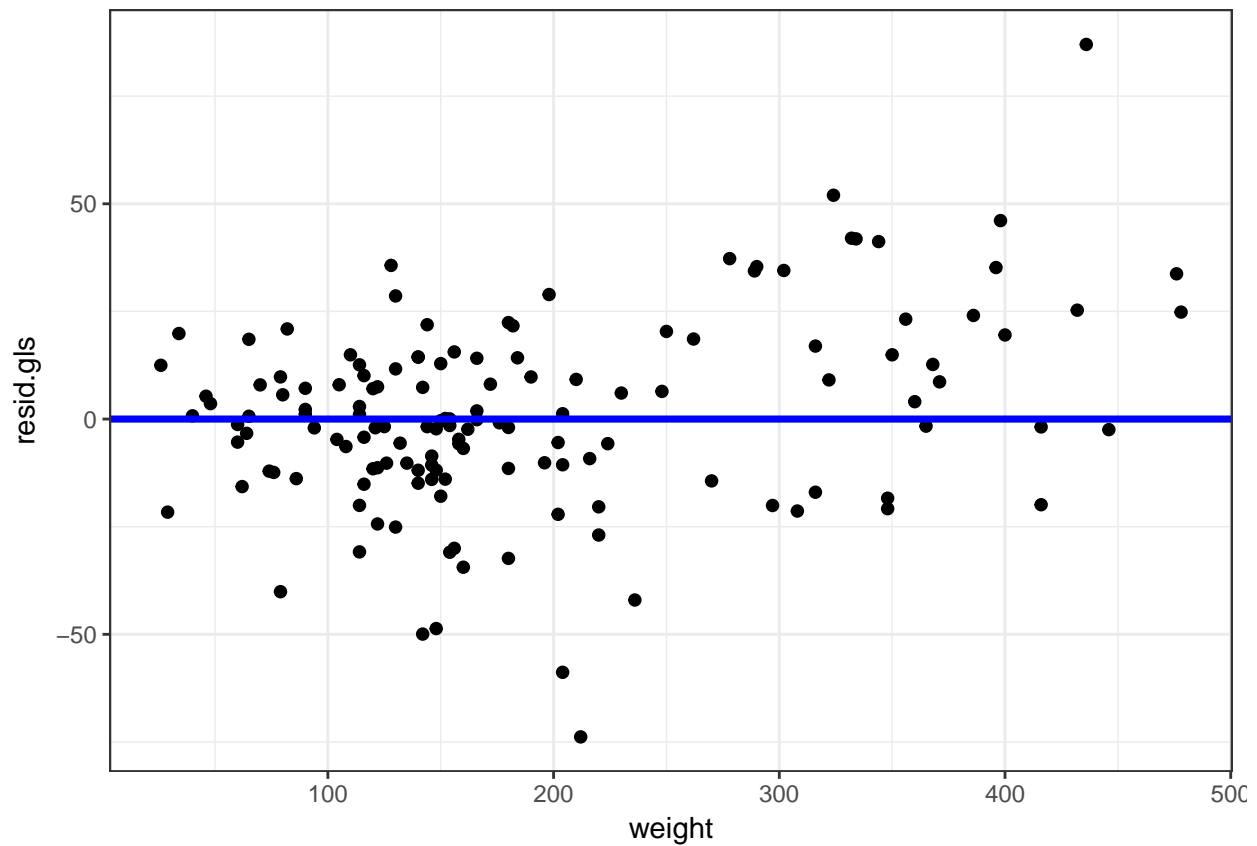
```
## same number of observations

##      df      AIC
## gls1  6 1268.551
## gls2  8 1267.498
## gls3  7 1270.551
```

## 9.2 Residuals

```
resid.gls <- resid(gls2)
residplot.gls <- ggplot(data=bearsRM, aes(y=resid.gls, x=weight))+
  geom_point(size=1.7)+
  geom_hline(yintercept=0, color="blue", size=1.25)+
  theme_bw()

print(residplot.gls)
```



## 10 Cubic Splines

Up until this point, chest girth and body length have been modeled using  $x^2$  and  $1/x$  transformation, respectively.

In this section, we replace these simple transformations with the cubic splines transformation, which allows for different transformations at different regions of the distribution of  $x$ .

## 10.1 Function

The following equation is Eq 2.25 in Harrell p. 20

```
natural.spline.comp<-function(X,t,j) {  
  k<-length(t)  
  pmax(0,X-t[j])^3-pmax(0,X-t[k-1])^3*(t[k]-t[j])/(t[k]-t[k-1])+  
    pmax(0,X-t[k])^3*(t[k-1]-t[j])/(t[k]-t[k-1])  
}
```

## 10.2 Chest Girth

### 10.2.1 Knots

To select proper knot placement for chest girth, I begin with 0.10, 0.50, and 0.90, and then tweak the placements and choose the placement that results in the lowest AIC score.

```
attach(bearsRM)  
knots <- quantile(chestGSc, probs=c(0.10,0.50,0.90), na.rm=TRUE)  
knots_2 <- quantile(chestGSc, probs=c(0.05,0.30,0.90), na.rm=TRUE)  
knots_3 <- quantile(chestGSc, probs=c(0.07,0.25,0.90), na.rm=TRUE)  
knots_4 <- quantile(chestGSc, probs=c(0.10,0.35,0.75), na.rm=TRUE)  
  
splinepoints <- natural.spline.comp(chestGSc, knots, 1)  
splinepoints_2 <- natural.spline.comp(chestGSc, knots_2, 1)  
splinepoints_3 <- natural.spline.comp(chestGSc, knots_3, 1)  
splinepoints_4 <- natural.spline.comp(chestGSc, knots_4, 1)
```

### 10.2.2 Spline point selection

- Model 1 = 0.10, 0.50, 0.90
- Model 2 = 0.05, 0.30, 0.90
- Model 3 = 0.07, 0.25, 0.90 << Best
- Model 4 = 0.10, 0.35, 0.75

```
gls_spline1 <- gls(weight ~ chestGSc + lengthSctransf + headWSc + ageNA + splinepoints,  
  weights=varExp(form=~as.vector(lengthSctransf)),  
  correlation=corCAR1(form = ~ age |name), data=bearsRM)  
  
gls_spline2 <- gls(weight ~ chestGSc + lengthSctransf + headWSc + ageNA + splinepoints_2,  
  weights=varExp(form=~as.vector(lengthSctransf)),  
  correlation=corCAR1(form = ~ age |name), data=bearsRM)  
  
gls_spline3 <- gls(weight ~ chestGSc + lengthSctransf + headWSc + ageNA + splinepoints_3,  
  weights=varExp(form=~as.vector(lengthSctransf)),  
  correlation=corCAR1(form = ~ age |name), data=bearsRM)  
  
gls_spline4 <- gls(weight ~ chestGSc + lengthSctransf + headWSc + ageNA + splinepoints_4,  
  weights=varExp(form=~as.vector(lengthSctransf)),  
  correlation=corCAR1(form = ~ age |name), data=bearsRM)
```

### 10.2.3 Spline point AIC

```
AIC(gls_spline1 ,gl_spline2 ,gl_spline3 ,gl_spline4)
```

```
##           df      AIC
## gls_spline1  9 1233.156
## gls_spline2  9 1282.236
## gls_spline3  9 1228.490
## gls_spline4  9 1232.279
```

## 10.3 Length - spline points

To select proper knot placement for chest girth, I begin with 0.10, 0.50, and 0.90, and then tweak the placements and choose the placement that results in the lowest AIC score.

```
knots_l_1 <- quantile(lengthSc, probs=c(0.10,0.50,0.90), na.rm=TRUE)
knots_l_2 <- quantile(lengthSc, probs=c(0.05,0.30,0.90), na.rm=TRUE)
knots_l_3 <- quantile(lengthSc, probs=c(0.07,0.25,0.90), na.rm=TRUE)
knots_l_4 <- quantile(lengthSc, probs=c(0.10,0.35,0.75), na.rm=TRUE)
```

```
splinepoints_l_1 <- natural.spline.comp(lengthSc, knots, 1)
splinepoints_l_2 <- natural.spline.comp(lengthSc, knots_2, 1)
splinepoints_l_3 <- natural.spline.comp(lengthSc, knots_3, 1)
splinepoints_l_4 <- natural.spline.comp(lengthSc, knots_4, 1)
```

- Model 1 = 0.10, 0.50, 0.90
- Model 2 = 0.05, 0.30, 0.90
- Model 3 = 0.07, 0.25, 0.90
- Model 4 = 0.10, 0.35, 0.75 << Best

```
gl_spline_l_1 <- gls(weight ~ chestGSc + lengthSc + headWSc +
                    ageNA + splinepoints_3 + splinepoints_l_1,
                    weights=varExp(form=~as.vector(lengthSctransf)),
                    correlation=corCAR1(form = ~ age |name), data=bearsRM)
```

```
gl_spline_l_2 <- gls(weight ~ chestGSc + lengthSc + headWSc +
                    ageNA + splinepoints_3 + splinepoints_l_2,
                    weights=varExp(form=~as.vector(lengthSctransf)),
                    correlation=corCAR1(form = ~ age |name), data=bearsRM)
```

```
gl_spline_l_3 <- gls(weight ~ chestGSc + lengthSc + headWSc +
                    ageNA + splinepoints_3 + splinepoints_l_3,
                    weights=varExp(form=~as.vector(lengthSctransf)),
                    correlation=corCAR1(form = ~ age |name), data=bearsRM)
```

```
gl_spline_l_4 <- gls(weight ~ chestGSc + lengthSc + headWSc +
                    ageNA + splinepoints_3 + splinepoints_l_4,
                    weights=varExp(form=~as.vector(lengthSctransf)),
                    correlation=corCAR1(form = ~ age |name), data=bearsRM)
```

```
AIC(gls_spline_l_1, gls_spline_l_2, gls_spline_l_3, gls_spline_l_4)
```

```
##           df      AIC
## gls_spline_l_1 10 1210.123
```

```
## gls_spline_l_2 10 1210.157
## gls_spline_l_3 10 1209.300
## gls_spline_l_4 10 1209.536
```

## 10.4 Restructure variance

Because the cubic splines transformation resulted in a lower AIC score and better model fit than the  $x^2$  and  $1/x$  transformation, we must revisit the manner in which within-group and between-group heteroskedasticity is being modeled.

Namely, we must consider that modeling this variance with a cubic splines transformation may also be better than modeling it with the  $x^2$  or  $1/x$  transformation.

To test for this, we build four potential models.

- Option 1 = model based upon splinepoints for chest girth
- Option 2 = model based upon splinepoints for length
- Option 3 = model based upon chest girth plus its splinepoints
- Option 4 = model based upon length plus its splinepoints

```
gls_spline_opt1 <- gls(weight ~ chestGSc + lengthSc + headWSc +
  ageNA + splinepoints_3 + splinepoints_l_4,
  weights=varExp(form=~as.vector(splinepoints_3)),
  correlation=corCAR1(form = ~ age | name), data=bearsRM)

gls_spline_opt2 <- gls(weight ~ chestGSc + lengthSc + headWSc +
  ageNA + splinepoints_3 + splinepoints_l_4,
  weights=varExp(form=~as.vector(splinepoints_l_4)),
  correlation=corCAR1(form = ~ age | name), data=bearsRM)

gls_spline_opt3 <- gls(weight ~ chestGSc + lengthSc + headWSc +
  ageNA + splinepoints_3 + splinepoints_l_4,
  weights=varExp(form=~as.vector(chestGSc + splinepoints_3)),
  correlation=corCAR1(form = ~ age | name), data=bearsRM)

gls_spline_opt4 <- gls(weight ~ chestGSc + lengthSc + headWSc +
  ageNA + splinepoints_3 + splinepoints_l_3,
  weights=varExp(form=~as.vector(lengthSc + splinepoints_l_3)),
  correlation=corCAR1(form = ~ age | name), data=bearsRM)

AIC(gls_spline_opt1, gls_spline_opt2, gls_spline_opt3, gls_spline_opt4)

##           df      AIC
## gls_spline_opt1 10 1203.178
## gls_spline_opt2 10 1212.882
## gls_spline_opt3 10 1201.739
## gls_spline_opt4 10 1211.680
gls_spline <- gls_spline_opt3
```

## 11 Restructure LME

Because the cubic splines transformation improved the model fit of the GLS so much, this section revisits the LME and applies the same cubic splines transformation to the best-fit LME model from above.

```
lme_final <- lme(weight ~ chestGSc + lengthSc + headWSc +
  ageNA + splinepoints_3 + splinepoints_1_4,
  random = ~ 1 | name/age,
  weights=varExp(form=~as.vector(chestGSc + splinepoints_3)),
  correlation=corCAR1(form = ~ age | name), data=bearsRM)
```

## 12 Comparison of models

### 12.1 AIC

```
cat(" AIC LM          ", round(AIC(lm1),2),
  "\n", "AIC GLS          ", round(AIC(gls2),2), "\n",
  "AIC GLS spline ", round(AIC(gls_spline), 2), "\n",
  "AIC LME spline ", round(AIC(lme_final),2),
  "\n")
```

```
## AIC LM          1282.12
## AIC GLS          1267.5
## AIC GLS spline  1201.74
## AIC LME spline  1202.34
```

### 12.2 Residuals

```
resid_lm <- resid(lm1)
resid_gls_spline <- resid(gls_spline)
resid_gls <- resid(gls2)
resid_lme <- resid(lme_final)

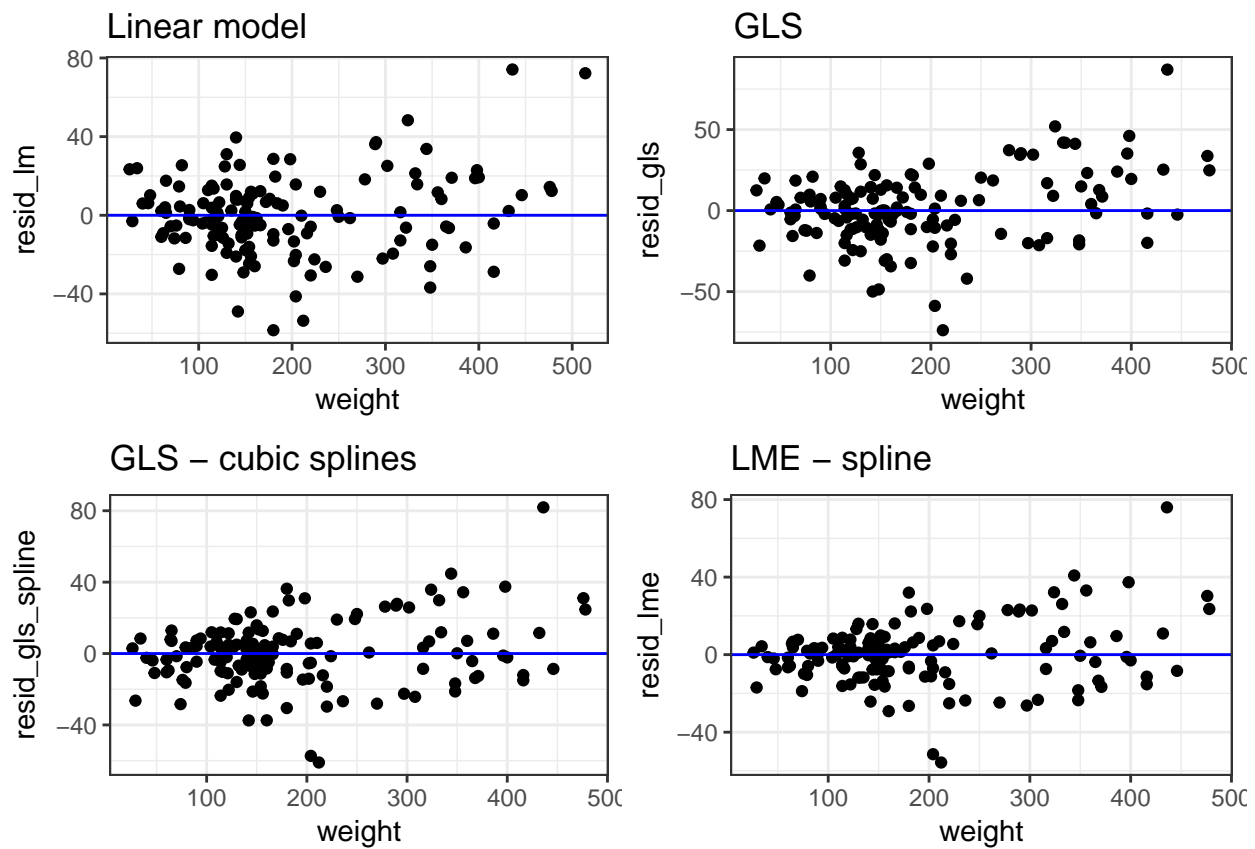
residplot_lm <- ggplot(data=bears, aes(y=resid_lm, x=weight))+
  geom_point()+
  geom_hline(yintercept=0, color="blue")+
  theme_bw()+
  ggtitle("Linear model")

residplot_gls_spline <- ggplot(data=bearsRM, aes(y=resid_gls_spline, x=weight))+
  geom_point()+
  geom_hline(yintercept=0, color="blue")+
  theme_bw()+
  ggtitle("GLS - cubic splines")

residplot_gls <- ggplot(data=bearsRM, aes(y=resid_gls, x=weight))+
  geom_point()+
  geom_hline(yintercept=0, color="blue")+
  theme_bw()+
  ggtitle("GLS")

residplot_lme <- ggplot(data=bearsRM, aes(y=resid_lme, x=weight))+
  geom_point()+
  geom_hline(yintercept=0, color="blue")+
  theme_bw()+
```

```
ggtitle("LME - spline")
multiplot(residplot_lm, residplot_gls_spline, residplot_gls, residplot_lme, cols=2)
```



### 12.3 Predicted vs observed

```
fitted_lm <- fitted(lm1)
fitted_gls_spline <- fitted(gls_spline)
fitted_gls <- fitted(gls2)
fitted_lme <- fitted(lme_final)

fitted_plot_lm <- ggplot(data=bears, aes(y=fitted_lm, x=weight))+
  theme_bw()+
  geom_point(size=1.7)+
  geom_smooth(method="lm", se=FALSE)+
  xlab("")+
  ylab("Fitted weight")+
  ggtitle("Linear model")+
  theme(panel.grid.minor=element_blank())

fitted_plot_gls_spline <- ggplot(data=bearsRM, aes(y=fitted_gls_spline, x=weight))+
  theme_bw()+
  geom_point(size=1.7)+
  geom_smooth(method="lm", se=FALSE)+
  ggtitle("GLS - spline")+
```



```

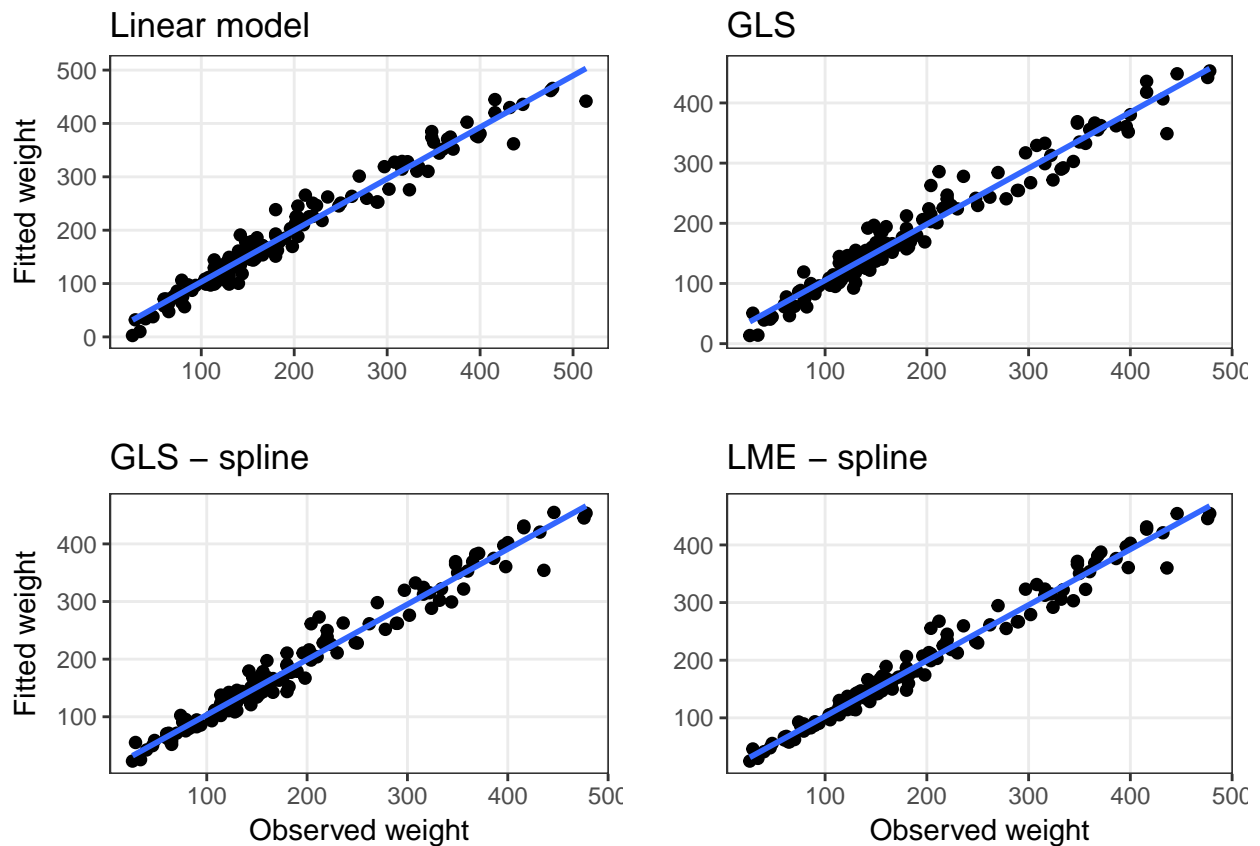
xlab("Observed weight")+
ylab("Fitted weight")+
theme(panel.grid.minor=element_blank())

fitted_plot_gls <- ggplot(data=bearsRM, aes(y=fitted_gls, x=weight))+
  theme_bw()+
  geom_point(size=1.7)+
  geom_smooth(method="lm", se=FALSE)+
  ggtitle("GLS")+
  xlab("")+
  ylab("")+
  theme(panel.grid.minor=element_blank())

fitted_plot_lme <- ggplot(data=bearsRM, aes(y=fitted_lme, x=weight))+
  theme_bw()+
  geom_point(size=1.7)+
  geom_smooth(method="lm", se=FALSE)+
  ggtitle("LME - spline")+
  ylab("")+
  xlab("Observed weight")+
  theme(panel.grid.minor=element_blank())

multiplot(fitted_plot_lm, fitted_plot_gls_spline, fitted_plot_gls, fitted_plot_lme, cols=2)

```



## 12.4 Residual sum of squares

```
RSS_lm <- sum(resid(lm1)^2)
RSS_gls <- sum(resid(gls2)^2)
RSS_gls_spline <- sum(resid(gls_spline)^2)
RSS_lme_spline <- sum(resid(lme_final)^2)

cat(" RSS LM          ", round(RSS_lm,2),
    "\n", "RSS GLS          ", round(RSS_gls,2), "\n",
    "RSS GLS spline ", round(RSS_gls_spline, 2), "\n",
    "RSQ LME spline ", round(RSS_lme_spline,2),
    "\n")

## RSS LM          58628.63
## RSS GLS         69851.49
## RSS GLS spline  49865.17
## RSQ LME spline  37288.98
```

## 13 Final model

The linear mixed effects model is chosen to be the best model for bear weight based upon its low AIC score and low residual sum of squares.

```
lme(weight ~ chestGSc + lengthSc + headWSc + ageNA +
     splinepoints_chest + splinepoints_length,
     random = ~ 1 | name/age,
     weights=varExp(form=~as.vector(chestGSc + splinepoints_chest)),
     correlation=corCAR1(form = ~ age | name))
```

```
AIC(lme_final)
```

```
## [1] 1202.342
```

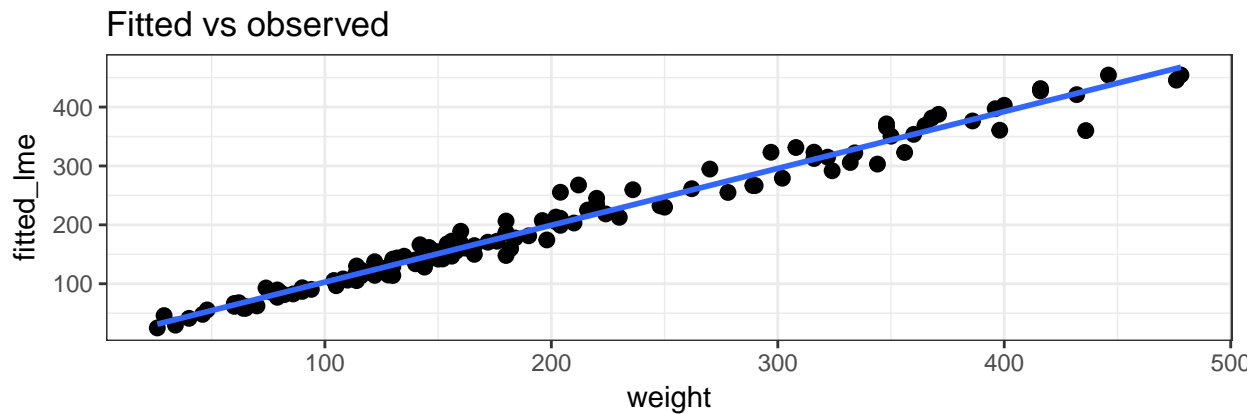
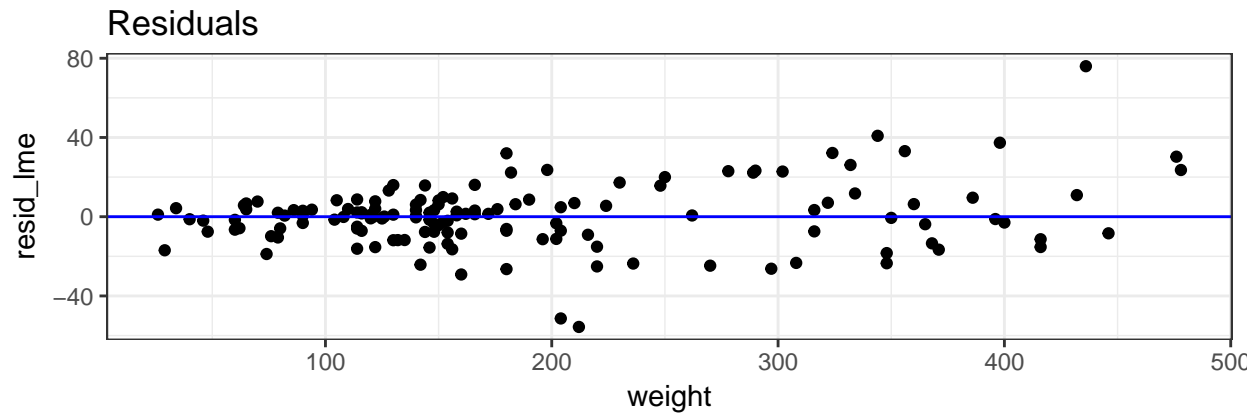
```
(RSS_lme_spline <- sum(resid(lme_final)^2))
```

```
## [1] 37288.98
```

```
fitted_plot_lme <- ggplot(data=bearsRM, aes(y=fitted_lme, x=weight))+
  theme_bw()+
  geom_point(size=2.3)+
  geom_smooth(method="lm", se=FALSE)+
  ggtitle("Fitted vs observed")
```

```
residplot_lme <- ggplot(data=bearsRM, aes(y=resid_lme, x=weight))+
  geom_point()+
  geom_hline(yintercept=0, color="blue")+
  theme_bw()+
  ggtitle("Residuals")
```

```
multiplot(residplot_lme, fitted_plot_lme)
```



```
summary(lme_final)
```

```
## Linear mixed-effects model fit by REML
## Data: bearsRM
##      AIC      BIC    logLik
## 1202.342 1237.206 -589.1712
##
## Random effects:
## Formula: ~1 | name
##      (Intercept)
## StdDev:    7.369669
##
## Formula: ~1 | age %in% name
##      (Intercept) Residual
## StdDev:  0.00178662 11.38655
##
## Correlation Structure: Continuous AR(1)
## Formula: ~age | name/age
## Parameter estimate(s):
## Phi
## 0.2
## Variance function:
## Structure: Exponential of variance covariate
## Formula: ~as.vector(chestGSc + splinepoints_3)
## Parameter estimates:
##      expon
## 0.09857261
```

```

## Fixed effects: weight ~ chestGSc + lengthSc + headWSc + ageNA + splinepoints_3 + splinepoints_1
##               Value Std.Error DF   t-value p-value
## (Intercept)   137.34347  4.600698 98 29.852749  0.0000
## chestGSc       18.39338  7.128357 37  2.580312  0.0140
## lengthSc       24.93949  5.021722 37  4.966323  0.0000
## headWSc         4.99183  2.174034 37  2.296115  0.0274
## ageNA           1.85700  3.131285 98  0.593048  0.5545
## splinepoints_3  17.38913  2.233030 37  7.787238  0.0000
## splinepoints_1_4 0.49507  1.937311 37  0.255544  0.7997
## Correlation:
##               (Intr) chstGS lngthS hedWSc ageNA spln_3
## chestGSc       0.527
## lengthSc      -0.066 -0.810
## headWSc        0.105 -0.095 -0.117
## ageNA         -0.417  0.066 -0.222 -0.064
## splinepoints_3 -0.459 -0.823  0.709 -0.086  0.001
## splinepoints_1_4 -0.166  0.511 -0.735  0.024  0.105 -0.741
##
## Standardized Within-Group Residuals:
##               Min           Q1           Med           Q3           Max
## -2.63632686 -0.53081533  0.04505409  0.48008005  2.50417096
##
## Number of Observations: 142
## Number of Groups:
##               name age %in% name
##               100      142

```