Case Study Pt. 2

Ian Hua, InHee Ho, Sonia Xu

September 9, 2017

# Case Study 1: Write Up 2

To continue our exploration of modelling the data, we decided to split the cleaned dataset from our first write-up into test and training data. We will fit the model on the training data to see how well it fits our test data. Our test data is 1/10 of the full dataset. Another goodness-of-fit measurement we will use is Root Mean Squared Error (RMSE), which measures the difference between the predicted response from a model and the true response the dataset. The lower the RMSE, the better the model fits the data. However, although a good indicator, it is important to note that relying solely on RMSE to assess the goodness-of-fit of a model is not enough; We must rely on multiple measurements like AIC, Adjusted R-Squared, ANOVA tests, etc to compare models. We sampled/fit the test and training data for 100 iterations for each model in an effort to normalize the performance of the models. By understanding how well the models predict the blotted uterus weight, we can see if there is improvement from utilizing more complex models with fixed and random effects in comparison to the simple linear model.

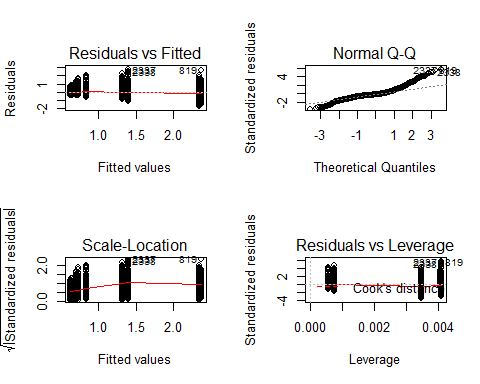
We believe that the best model predicts the most similar distribution of the test dataset compared to the true test dataset. Thus, the best models should be the most similar to this quantile:

quantile(test$wblot)

## 0% 25% 50% 75% 100%   
## 0.2625000 0.5051803 0.6704120 1.5944449 3.6455696

# Simple Linear Model

lm1 <- lm(wblot ~ 1+ poly(dose1,2) + poly(dose2,2), data = dat)  
par(mfrow=c(2,2))  
plot(lm1)



Checking the basic assumptions of normality for the simple linear model, we can clearly see problems in the model. Comparing its prediction accuracy to the true dataset, we obtain 0.6180028, 0.6254355, 0.7132286, 1.3050943, 2.3818359, which has a similar distribution mean. However, the tails of the predicted test dataset are wider than the true test dataset, which suggests that this model can be improved. Again, using the test data, the predicted response's Root Mean Squared Error (RMSE) is 0.4943245.

# Proposed Model from Write-up 1

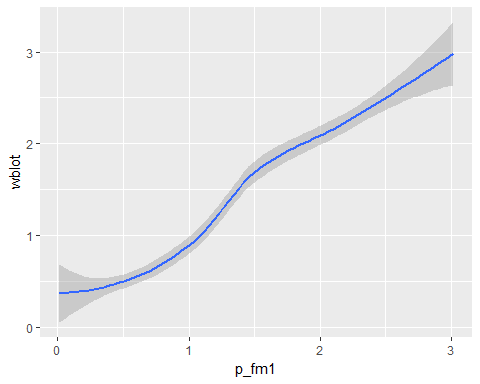
fm1 <- lmer(wblot ~ 1 + (proto|lab) + body + poly(dose1,2) + poly(dose2,2), dat)

## Warning: Some predictor variables are on very different scales: consider  
## rescaling

summary(fm1)

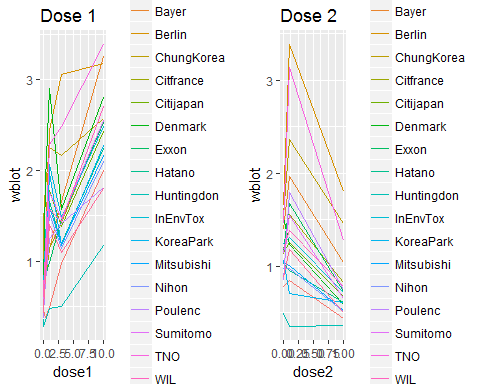
## Linear mixed model fit by REML ['lmerMod']  
## Formula: wblot ~ 1 + (proto | lab) + body + poly(dose1, 2) + poly(dose2,   
## 2)  
## Data: dat  
##   
## REML criterion at convergence: 2828.2  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.8496 -0.6123 -0.0756 0.3914 6.6255   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## lab (Intercept) 0.06703 0.2589   
## protoB 0.02928 0.1711 -0.14   
## protoC 0.05636 0.2374 -0.82 0.68   
## protoD 0.06945 0.2635 -0.83 0.67 1.00  
## Residual 0.16170 0.4021   
## Number of obs: 2677, groups: lab, 19  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 1.310205 0.067797 19.33  
## body -0.002174 0.000250 -8.69  
## poly(dose1, 2)1 30.946695 0.418625 73.92  
## poly(dose1, 2)2 -25.885510 0.587567 -44.06  
## poly(dose2, 2)1 -23.975696 0.518999 -46.20  
## poly(dose2, 2)2 11.729581 0.499744 23.47  
##   
## Correlation of Fixed Effects:  
## (Intr) body p(1,2)1 p(1,2)2 p(2,2)1  
## body -0.910   
## ply(ds1,2)1 -0.071 0.078   
## ply(ds1,2)2 0.037 -0.041 -0.197   
## ply(ds2,2)1 0.027 -0.029 -0.229 0.623   
## ply(ds2,2)2 -0.030 0.033 0.216 -0.585 -0.375   
## fit warnings:  
## Some predictor variables are on very different scales: consider rescaling

## `geom\_smooth()` using method = 'loess'

 Based on our intuitive assumptions from Case Study 1, above is our proposed model from write-up 1. The plot demonstrates how well our proposed model's prediction of the test data fits the true test data--a linear trend indicates perfect prediction. Overall, the predictive power of our intuitive model seems to work pretty well. The distribution of our proposed model more closely follows that of the true dataset compared to the Simple Linear Model, since the RMSE (0.4089548) is lower by 0.0853696 in comparison to the baseline model.

# Minimum Model

However, we were informed by a more experienced Statistician (our TA) about the necessity to model the data with a minimum model that contained random effects for protocol, dose1, and dose 2 for each lab because there is high variance in the effect of dose1 and dose2 for each lab.

The plots below demonstrate these differences:  In order to capture these random effects of dose1 and dose2, below is our minimum model.

#dose response curve  
minmod <- lmer(wblot ~ (1 + proto + poly(dose1,2) + poly(dose2,2)| lab), dat) #minimum model

## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 \*  
## length(par)^2 is not recommended.

## Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp), :  
## convergence code 1 from bobyqa: bobyqa -- maximum number of function  
## evaluations exceeded

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control  
## $checkConv, : unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control  
## $checkConv, : Model failed to converge: degenerate Hessian with 9 negative  
## eigenvalues

summary(minmod)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: wblot ~ (1 + proto + poly(dose1, 2) + poly(dose2, 2) | lab)  
## Data: dat  
##   
## REML criterion at convergence: 2274.4  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.1661 -0.5782 -0.1335 0.3423 7.7103   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## lab (Intercept) 0.07474 0.2734   
## protoB 0.01541 0.1241 -0.10   
## protoC 0.08497 0.2915 0.38 -0.26   
## protoD 0.08526 0.2920 0.31 -0.29 1.00   
## poly(dose1, 2)1 185.56613 13.6223 -0.16 0.40 -0.23 -0.23   
## poly(dose1, 2)2 153.35477 12.3836 -0.12 -0.25 -0.19 -0.17 -0.81  
## poly(dose2, 2)1 103.67722 10.1822 0.12 -0.37 0.08 0.08 -0.92  
## poly(dose2, 2)2 66.85543 8.1765 -0.02 0.36 -0.06 -0.07 0.39  
## Residual 0.12160 0.3487   
##   
##   
##   
##   
##   
##   
##   
## 0.92   
## -0.42 -0.55  
##   
## Number of obs: 2677, groups: lab, 19  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 1.21675 0.05245 23.2  
## convergence code: 1  
## unable to evaluate scaled gradient  
## Model failed to converge: degenerate Hessian with 9 negative eigenvalues  
## maxfun < 10 \* length(par)^2 is not recommended.

The Minimum Model performs better than our Proposed Model, with an average RMSE of 0.3525729 after 100 iterations, which is 0.0563819 lower than our Proposed Model. Furthermore, the AIC for the Minimum Model (2058.5225526) is lower and thus better than the Proposed Model's AIC (2536.1037174), which implies that this model is an improvement from the last.

With the Minimum Model as our baseline, we continued to explore different models. After much exploration, this was our Final Model:

# Final Model

finmod <- lmer(wblot ~ (1 + poly(dose1,2) + poly(dose2,2) |lab)+poly(dose1,2)+poly(dose2,2)+poly(body,2)+(proto\*lab), dat) #final model

## fixed-effect model matrix is rank deficient so dropping 35 columns / coefficients

summary(finmod)

## Linear mixed model fit by REML ['lmerMod']  
## Formula:   
## wblot ~ (1 + poly(dose1, 2) + poly(dose2, 2) | lab) + poly(dose1,   
## 2) + poly(dose2, 2) + poly(body, 2) + (proto \* lab)  
## Data: dat  
##   
## REML criterion at convergence: 2110.3  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.2513 -0.5889 -0.1083 0.3806 7.7797   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## lab (Intercept) 0.07871 0.2805   
## poly(dose1, 2)1 55.30645 7.4368 0.23   
## poly(dose1, 2)2 140.10202 11.8365 0.01 -0.73   
## poly(dose2, 2)1 55.96697 7.4811 -0.13 -0.70 0.95   
## poly(dose2, 2)2 65.03241 8.0643 0.17 -0.04 -0.39 -0.62  
## Residual 0.11567 0.3401   
## Number of obs: 2677, groups: lab, 19  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 0.354428 0.258647 1.370  
## poly(dose1, 2)1 31.265366 1.755195 17.813  
## poly(dose1, 2)2 -25.018384 2.776556 -9.011  
## poly(dose2, 2)1 -23.146742 1.782438 -12.986  
## poly(dose2, 2)2 9.815022 1.912302 5.133  
## poly(body, 2)1 -32.281934 5.018409 -6.433  
## poly(body, 2)2 7.962466 1.237298 6.435  
## protoB 0.061557 0.059204 1.040  
## protoC 0.873191 0.230870 3.782  
## protoD 1.373855 0.202903 6.771  
## labBayer 0.149291 0.357880 0.417  
## labBerlin 0.939123 0.359108 2.615  
## labChungKorea 0.935203 0.352196 2.655  
## labCitfrance 0.394689 0.355078 1.112  
## labCitijapan 0.317029 0.350639 0.904  
## labDenmark 0.366497 0.360410 1.017  
## labExxon 0.395591 0.355434 1.113  
## labHatano 0.200756 0.349890 0.574  
## labHuntingdon 0.334026 0.360317 0.927  
## labInEnvTox 0.531214 0.350073 1.517  
## labKoreaPark 0.462509 0.362530 1.276  
## labMitsubishi 0.293980 0.349760 0.841  
## labNihon 0.176111 0.349775 0.503  
## labPoulenc 0.285785 0.356523 0.802  
## labSumitomo 0.366880 0.350472 1.047  
## labTNO 0.441524 0.354105 1.247  
## labWIL 0.170249 0.352401 0.483  
## labZeneca 0.433600 0.350491 1.237  
## protoB:labChungKorea 0.184060 0.083823 2.196  
## protoB:labCitijapan 0.061662 0.083728 0.736  
## protoC:labCitijapan 0.295197 0.105939 2.786  
## protoB:labHatano 0.117473 0.083980 1.399  
## protoC:labHatano 0.381309 0.098707 3.863  
## protoD:labHatano -0.083569 0.086176 -0.970  
## protoB:labInEnvTox -0.002785 0.083742 -0.033  
## protoC:labInEnvTox -0.029063 0.126368 -0.230  
## protoD:labInEnvTox -0.433572 0.086059 -5.038  
## protoB:labKoreaPark -0.459628 0.113544 -4.048  
## protoB:labMitsubishi 0.138603 0.083794 1.654  
## protoC:labMitsubishi 0.326216 0.101387 3.218  
## protoD:labMitsubishi -0.165050 0.085672 -1.927  
## protoB:labNihon 0.150643 0.083781 1.798  
## protoC:labNihon 0.466314 0.111013 4.201  
## protoB:labSumitomo 0.126498 0.083756 1.510  
## protoC:labSumitomo 0.281010 0.108261 2.596  
## protoB:labTNO 0.261240 0.084138 3.105  
## protoB:labWIL -0.036273 0.083728 -0.433

##   
## Correlation matrix not shown by default, as p = 47 > 12.  
## Use print(x, correlation=TRUE) or  
## vcov(x) if you need it

## fit warnings:  
## fixed-effect model matrix is rank deficient so dropping 35 columns / coefficients

Overall, our Final Model's average RMSE (0.3465542) is again lower than the Minimum Model's average RMSE after 100 iterations. Comparing our Final Model's AIC against the Minimum Models' AIC, again, our Final Model has a lower AIC (1924.8597038) than our Minimum Model (2058.5225526). Furthermore, when conducting an ANOVA test between our Final Model and the Minimum Model, the Final Model is significantly better.

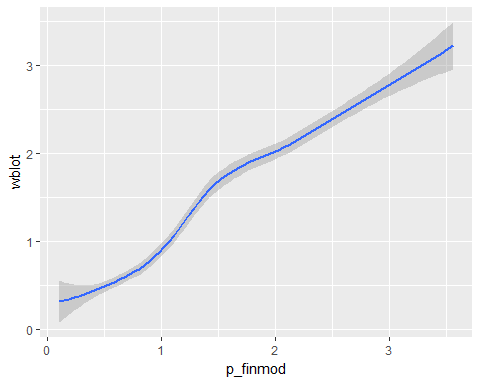
anova(finmod, minmod)

## refitting model(s) with ML (instead of REML)

## Data: train  
## Models:  
## minmod: wblot ~ (1 + proto + poly(dose1, 2) + poly(dose2, 2) | lab)  
## finmod: wblot ~ (1 + poly(dose1, 2) + poly(dose2, 2) | lab) + poly(dose1,   
## finmod: 2) + poly(dose2, 2) + poly(body, 2) + (proto \* lab)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)   
## minmod 38 2107.6 2327.5 -1015.81 2031.6   
## finmod 63 1885.5 2250.1 -879.74 1759.5 272.15 25 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Below is a plot of how well our final model predicts the true data. Overall, the trend is linear, which indicates a relatively accurate prediction.

## `geom\_smooth()` using method = 'loess'



After exploring and comparing multiple models, our final model looks like this:

where and $.

We noticed that adding dose1 and dose2 as both fixed and random effects better fits the data. The fixed effect captures the population effects of dose1 and dose2 across all labs and random effect captures the variation in the effects of dose1 and dose2 between labs. Similarly, we also assumed that different types of protocol have a different effect for each labs. After fitting the model with the interaction between protocol and lab (proto\*lab) and with the random effects of protocol on each labs (proto|lab) respectively, we observed that the former better fits the actual dataset (see Appendix). We decided to transform body weight, dose1, and dose2 to a polynomial with a degree of 2 because the scatterplot indicated a nonlinear trend between body weight and the response, the weighted blot.

# Appendix

This ANOVA test demonstrates why we chose to use proto\*lab as a fixed effect instead of a random effect.

fm1 <- lmer(wblot ~ (1 + poly(dose1,2) + poly(dose2,2) |lab) + poly(dose1,2) + poly(dose2,2) + poly(body,2) + (proto\*lab), dat) # interaction

## fixed-effect model matrix is rank deficient so dropping 35 columns / coefficients

fm2 <- lmer(wblot ~ (1 + proto + poly(dose1,2) + poly(dose2,2) |lab) + poly(dose1,2) + poly(dose2,2) + poly(body,2), dat) # random effect  
  
anova(fm1, fm2)

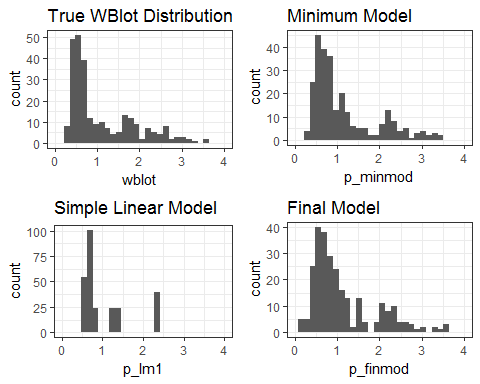
## refitting model(s) with ML (instead of REML)

## Data: dat  
## Models:  
## fm2: wblot ~ (1 + proto + poly(dose1, 2) + poly(dose2, 2) | lab) +   
## fm2: poly(dose1, 2) + poly(dose2, 2) + poly(body, 2)  
## fm1: wblot ~ (1 + poly(dose1, 2) + poly(dose2, 2) | lab) + poly(dose1,   
## fm1: 2) + poly(dose2, 2) + poly(body, 2) + (proto \* lab)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)   
## fm2 44 2166.2 2425.4 -1039.08 2078.2   
## fm1 63 2066.8 2438.0 -970.38 1940.8 137.41 19 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

These plots illustrate the distributions of the predicted response, wblot. The more similar the distribution to the original wblot, the more accurate the prediction and the better the model.

test\_p <- ggplot(data = test, aes(x = wblot)) + geom\_histogram() + lims(x = c(0,4)) + theme\_bw() + ggtitle("True WBlot Distribution")  
lm\_p <- ggplot(test,aes(p\_lm1)) + geom\_histogram() + lims(x = c(0,4)) + theme\_bw() + ggtitle("Simple Linear Model")  
min\_p <- ggplot(test,aes(p\_minmod)) + geom\_histogram() + lims(x = c(0,4)) + theme\_bw() + ggtitle("Minimum Model")  
fin\_p <- ggplot(test,aes(p\_finmod)) + geom\_histogram() + lims(x = c(0,4)) + theme\_bw() + ggtitle("Final Model")  
multiplot(test\_p, lm\_p, min\_p, fin\_p, cols = 2)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Contributions

Sonia Xu - goodness-of-fit for every model (RMSE, test/training), wrote case study 2, explored models InHee Ho - explored various models, edited & added to case study 2 Ian Hua - explored various models, edited case study 2