HUDM6122 Homework_10

Chenguang Pan

2023-05-06

0.1 Github Address

All my latest homework can be found on Github: https://github.com/cgpan/hudm6122_homeworks . Thanks for checking if interested.

0.2 Homework Reference

Due to the heavy workload during this final and given the difficulty of the last two questions, part of the Ex 8.4 and Ex 8.5 solutions of this assignment has referred to the answers from this webpage(click here).

Ex. 8.1

The final model fitted to the timber data did not constrain the fitted curves to go through the origin, although this is clearly necessary. Fit an amended model where this constraint is satisfied, and plot the new predicted values.

MY SOLUTION:

To make this homework's layout beautiful, I wrote a separate R script to load all the required dataset.

```
> # import the R script containing the dataset
> source("hw_10_data_source.r")
> df <- timber()
> head(df)
        specimen slippage loads
spec1.1    spec1        0     0
spec2.1    spec2        0     0
spec3.1    spec3        0     0
spec4.1    spec4        0     0
spec5.1    spec5        0     0
spec6.1    spec6        0     0
> dim(df)
[1] 120   3
```

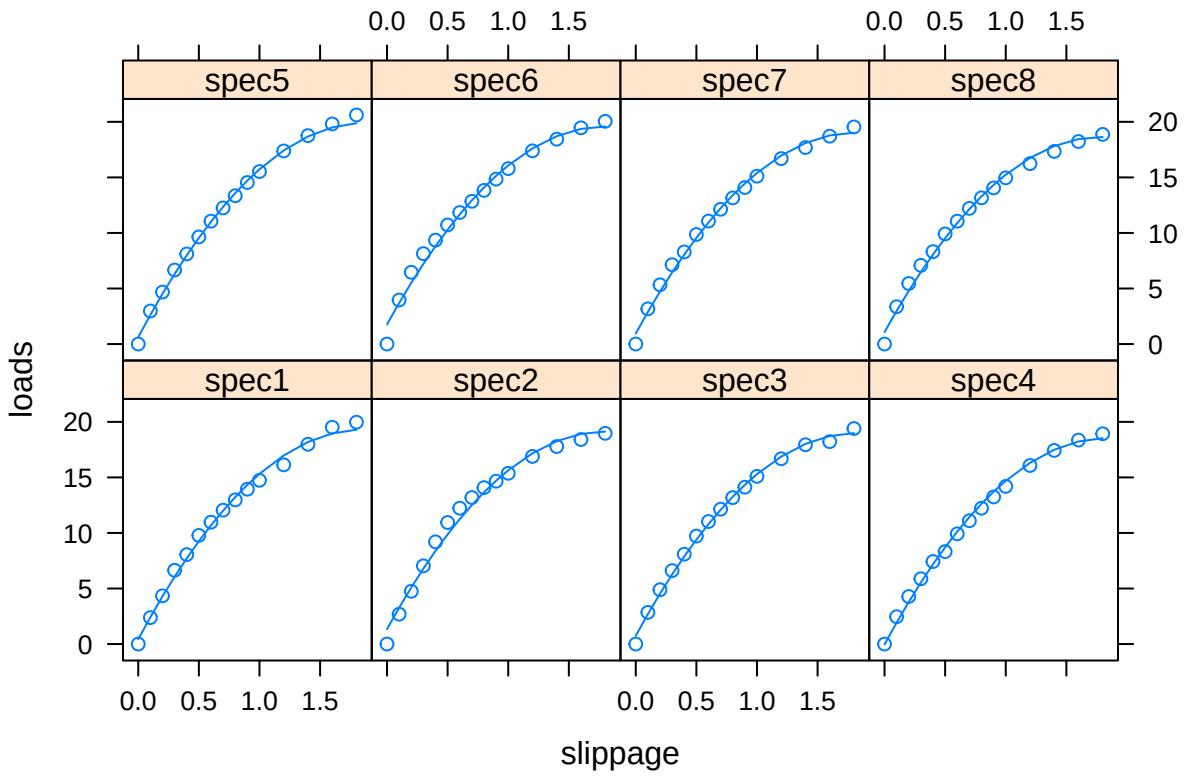
The data looks good. To constrain the fitted curve to go through the origin, one simply need to remove the intercept term from the model. Then, Y will be zero if all predictors are zero.

```
> library(nlme)
> # import the final model
> timber.lme2 <- lme(loads ~ slippage + I(slippage^2),
```

```
+                    random = ~slippage|specimen,
+                    data = df,method = "ML")
> # remove the intercept from the model or just set the intercept to 0
> timber.lme3 <- lme(loads ~ 0 + slippage + I(slippage^2),
+                    random = ~slippage|specimen,
+                    data = df,method = "ML")
> # see the result
> summary(timber.lme3)
Linear mixed-effects model fit by maximum likelihood
  Data: df
       AIC      BIC    logLik
  222.4795 239.2045 -105.2398

Random effects:
 Formula: ~slippage | specimen
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.0285296 (Intr)
slippage    0.5374447 -0.873
Residual    0.4970639

Fixed effects:  loads ~ 0 + slippage + I(slippage^2)
                  Value Std.Error  DF   t-value p-value
slippage      20.465438 0.2716921 111  75.32586       0
I(slippage^2) -5.513104 0.1624059 111 -33.94645       0
 Correlation:
              slippg
I(slippage^2) -0.913

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.54914093 -0.54581926  0.07432158  0.68613181  2.15422341

Number of Observations: 120
Number of Groups: 8
```

It looks good.

```
> # get the predicted value
> df$pred1 <- predict(timber.lme3)
> # plot the graph
> library(lattice)
> pfun <- function(x, y) {
+     panel.xyplot(x, y[1:length(x)])
+     panel.lines(x, y[1:length(x) + length(x)], lty = 1)
+ }
>
> plot(xyplot(cbind(loads, pred1) ~ slippage | specimen, data = df,
+      panel = pfun, layout = c(4, 2), ylab = "loads"))
```



Great! All fitted curve goes through the origin!

Ex. 8.2

Investigate a further model for the glucose challenge data that allows a random quadratic effect.

```
> # import the data
> plasma <- plasma()
> head(plasma)
       Subject   group time plasma
id01.1    id01 control    1    4.3
id02.1    id02 control    1    3.7
id03.1    id03 control    1    4.0
id04.1    id04 control    1    3.6
id05.1    id05 control    1    4.1
id06.1    id06 control    1    3.8
> dim(plasma)
[1] 264   4
```

The data looks good.

```
> # write a new model with random quadratic effect
> plasma.lme3 <- lme(plasma ~ time*group + I(time^2),
+                    random = ~time+I(time^2)|Subject,
+                    data = plasma, method = "ML")
> summary(plasma.lme3)
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: plasma
       AIC      BIC    logLik
  384.9571 427.8685 -180.4786

Random effects:
 Formula: ~time + I(time^2) | Subject
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 0.79150689 (Intr) time
time        0.23853208 -0.722
I(time^2)   0.02225533  0.599 -0.951
Residual    0.36633783

Fixed effects:  plasma ~ time * group + I(time^2)
                    Value  Std.Error  DF    t-value p-value
(Intercept)      4.631515 0.19331352 228  23.958566   0e+00
time            -0.753964 0.06354367 228 -11.865284   0e+00
groupobese       1.067254 0.25322087  31   4.214716   2e-04
I(time^2)        0.084668 0.00632243 228  13.391706   0e+00
time:groupobese -0.125195 0.03434255 228  -3.645471   3e-04
 Correlation:
                (Intr) time   gropbs I(t^2)
time            -0.724
groupobese      -0.516  0.144
I(time^2)        0.569 -0.941  0.000
time:groupobese  0.350 -0.213 -0.678  0.000

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-2.877281204 -0.544442854  0.002749287  0.560328140  2.929399256

Number of Observations: 264
Number of Groups: 33
```

The model looks good. Next, use Likelihood ratio test to compare the model fit.

```
> plasma.lme2 <- lme(plasma ~ time*group + I(time^2),
+                    random = ~time|Subject,
+                    data = plasma, method = "ML")
> anova(plasma.lme2,plasma.lme3)
            Model df      AIC      BIC    logLik   Test  L.Ratio p-value
plasma.lme2     1  9 383.3149 415.4984 -182.6575
plasma.lme3     2 12 384.9571 427.8685 -180.4786 1 vs 2 4.357781  0.2253
```

Note, the p-value associated with the likelihood ratio test is .2253, indicating that the random quadratic effect is not better than the nested model. Therefore, I prefer not to add the random quadratic effect into the model.

Ex. 8.3

Fit an independence model to the Beat the Blues data, and compare the estimated treatment effect confidence interval with that from the random intercept model described in the text.

```
> # import the data
> data("BtheB", package = "HSAUR2")
> BtheB$subject <- factor(rownames(BtheB))
> nobs <- nrow(BtheB)
> BtheB_long <- reshape(BtheB, idvar = "subject",
+     varying = c("bdi.2m", "bdi.3m", "bdi.5m", "bdi.8m"),
+     direction = "long")
> BtheB_long$time <- rep(c(2, 3, 5, 8), rep(nobs, 4))
> head(BtheB_long)
     drug length treatment bdi.pre subject time bdi
1.2m   No    >6m       TAU      29       1    2   2
2.2m  Yes    >6m     BtheB      32       2    2  16
3.2m  Yes    <6m       TAU      25       3    2  20
4.2m   No    >6m     BtheB      21       4    2  17
5.2m  Yes    >6m     BtheB      26       5    2  23
6.2m  Yes    <6m     BtheB       7       6    2   0
> dim(BtheB_long)
[1] 400   7
```

The data looks good. Next, fit an independence model. In an independence model, the observations at each time point are assumed to be independent of each other, meaning that there is no correlation between them.

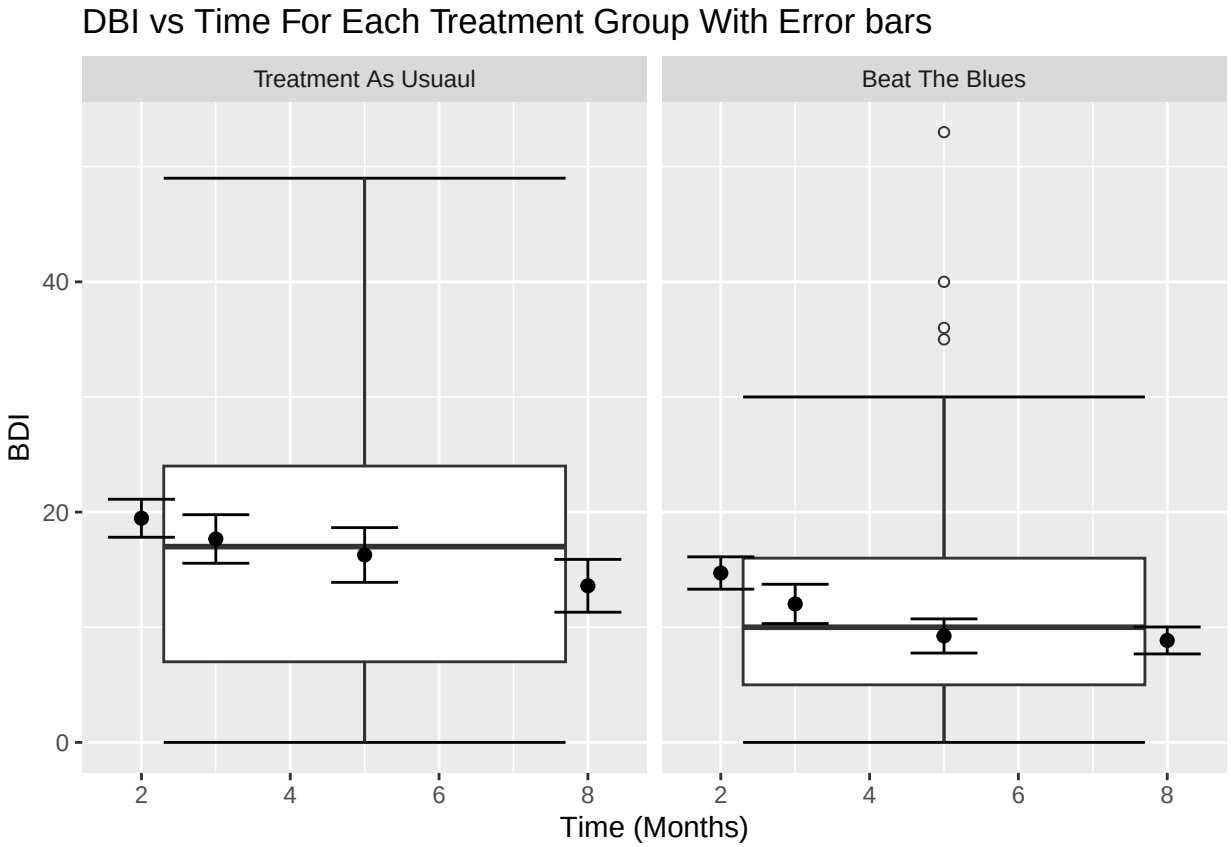
```
> BtheB_lme3 <- lm(bdi ~ bdi.pre + time + treatment+ drug+ length +subject,
+                  data = BtheB_long, na.action = na.omit)
> summary(BtheB_lme3)$coefficients["treatmentBtheB", c("Estimate","Std. Error",
+                                                      "t value", "Pr(>|t|)")]
   Estimate  Std. Error     t value    Pr(>|t|)
-17.1935329  11.7573196  -1.4623684   0.1453649
```

From the result, one can have the 95% confidence interval(CI) is $-17.19 \pm 1.96 \times 11.75 = (-40.24, 5.86)$. This treatment effect is not statistically significant. The 95% CI from the random intercept model is $-2.315 \pm 1.96 \times 1.72 = (-5.69, 1.06)$

Ex. 8.4

Construct a plot of the mean profiles of the two treatment groups in the Beat the Blues study showing also the predicted mean profiles under the model used in the chapter. Repeat the exercise with a model that includes only a time effect.

```
> library(ggplot2)
> treatmentlabels <-c('TAU'='Treatment As Usuaul', 'BtheB'='Beat The Blues')
> ggplot(BtheB_long, aes(time, bdi)) +
+   stat_boxplot(geom='errorbar', linetype=1, width=0.5) +
+   geom_boxplot( aes(time, bdi),outlier.shape=1) +
+   facet_grid(~treatment, labeller = as_labeller(treatmentlabels)) +
+   stat_summary(fun.y=mean, geom='point', size=2) +
+   stat_summary(fun.data=mean_se, geom='errorbar') +
+   labs(title='DBI vs Time For Each Treatment Group With Error bars',
+        x='Time (Months)',y='BDI')
```

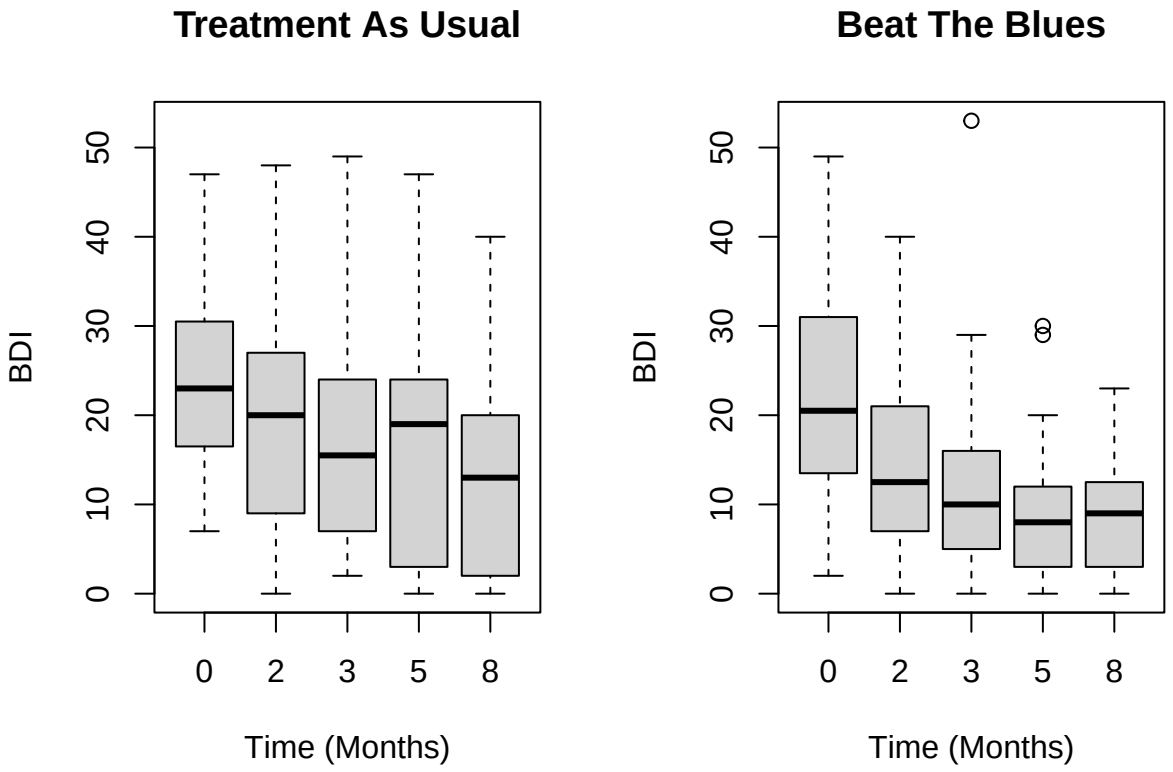


Ex. 8.5

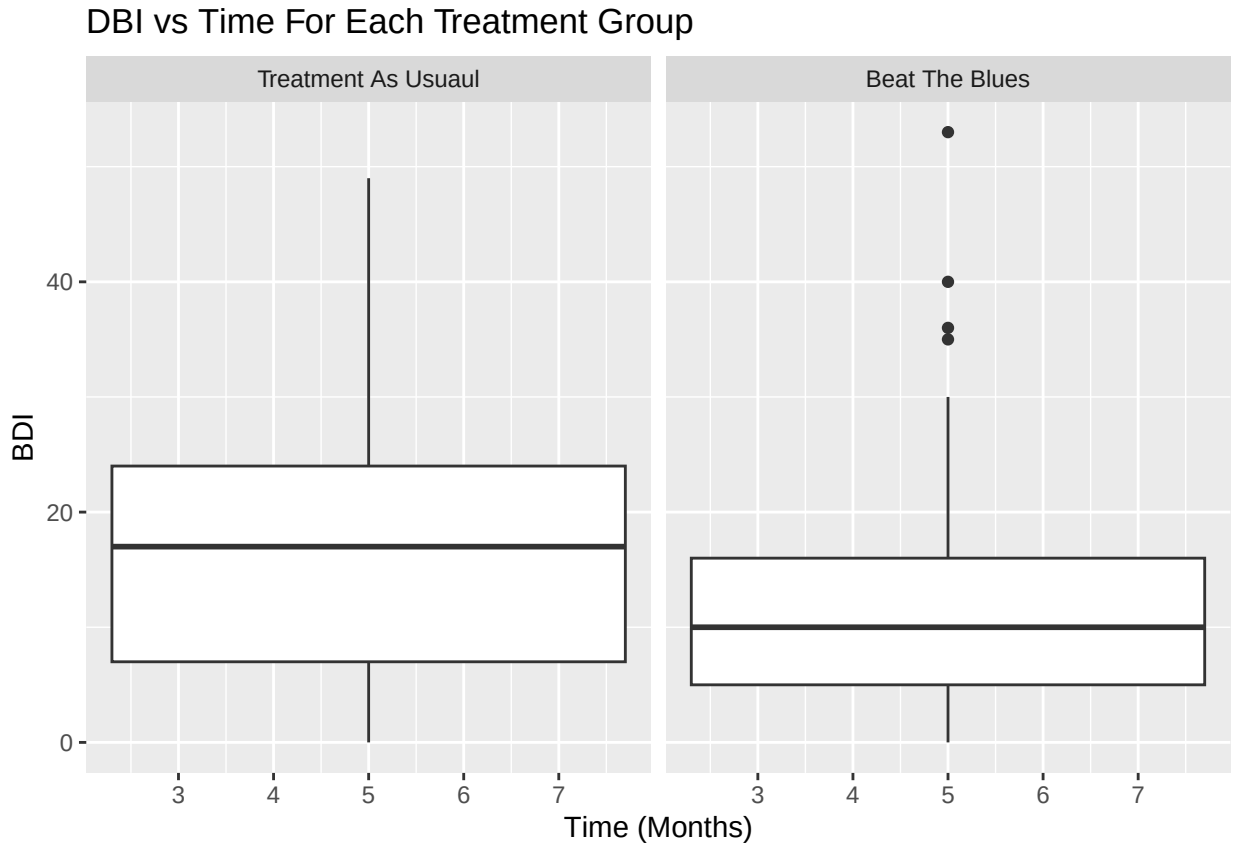
Investigate whether there is any evidence of an interaction between treatment and time for the Beat the Blues data.

Reference Note:
The following solution refers to the answers on this webpage. Please click the hyperlink for more detail.

```
> library("HSAUR3")
> library("ggplot2")
> library("lme4")
> layout(matrix(1:2, nrow=1))
> ylim <- range(BtheB[,grep('bdi', names(BtheB))],
+               na.rm=TRUE)
>
>
> #subsetting the data for treatment as usual group
> tau <- subset(BtheB, treatment=='TAU')[,grep('bdi', names(BtheB))]
> #developing box plots for each time interval for the treatment as usual group based on BDI values
> boxplot(tau, main='Treatment As Usual',xlab='Time (Months)', ylab='BDI', ylim=ylim,
+ names=c(0,2,3,5,8))
>
> #subseting the data for BtheB group
> btb <- subset(BtheB, treatment=='BtheB')[,grep('bdi', names(BtheB))]
> #developing box plots for each time interval for the treatment as usual group based on BDI values
> boxplot(btb, main='Beat The Blues', xlab='Time (Months)', ylab='BDI', ylim=ylim, names=c(0,2,3,5,8))
```




```
> #ggplot can now easily divide the categorical variable (treatment) into its groups TAU vs BtheB
> treatmentlabels <-c('TAU'='Treatment As Usuaul', 'BtheB'='Beat The Blues')
>
> ggplot(BtheB_long, aes(time,bdi)) + geom_boxplot() + facet_grid(~treatment, labeller = as_labeller(tr
```



Over time, the BDI values show a decreasing trend, while the dispersion of data in the TAU group appears to be wider than the BtheB group. To investigate the effect of treatment and its interaction with time, I will fit both a fixed and mixed model with the same covariates used in the previous models.

```
> #Fitting linear model
> btheb_lm2 <- lm(bdi ~ bdi.pre + time + drug + length + treatment + subject + treatment*time, data= Bt
>
> #Fitting mixed Model
> btheb_lmer2 <- lmer(bdi ~ bdi.pre + time + drug + length + treatment + treatment*time + (1 | subject)
> anova(btheb_lmer2, btheb_lm2)
Data: BtheB_long
Models:
btheb_lmer2: bdi ~ bdi.pre + time + drug + length + treatment + treatment * time + (1 | subject)
btheb_lm2: bdi ~ bdi.pre + time + drug + length + treatment + subject + treatment * time
            npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
btheb_lmer2    9 1886.7 1919.4 -934.35   1868.7
btheb_lm2    100 1773.0 2136.5 -786.51   1573.0 295.67 91  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the results, the fixed model is prefered. Next, checking the interacion coefficient,

```
> summary(btheb_lm2)$coefficients["time:treatmentBtheB", c("Estimate","Std. Error",
+                                                          "t value", "Pr(>|t|)")]
  Estimate Std. Error    t value   Pr(>|t|)
0.64357788 0.29756513 2.16281351 0.03186724
```

The interaction effect is significant in the fixed model.