Dr. Özlem İLK DAĞ

Fall 2023-2024

**STAT 545- LONGITUDINAL DATA ANALYSIS**

**HOMEWORK 1 SOLUTION**

**Part 2.** I simulated a data set, and post it at ODTUClass under the name sim.data.

Consider **Y0** as a baseline measurement and as **a covariate** for this dataset.

1. Is this data in long or wide form? **Wide form**
2. Fill in the blank: The disadvantage of such data form is **that it looks awkwardly wide when we have many time-dependent variables.**
3. Convert it into the other form and call this new dataset sim.data2. Give the first 5 lines of sim.data2. What is the dimension of sim.data2?

**I used the following codes, but there is more than one way:**

**> hw1.sim=read.csv("sim.data.csv")**

**> dim(hw1.sim)**

**[1] 500 8**

**> library(reshape2)**

**> sim.data2 <- melt(hw1.sim,**

**+ id.vars=c("id","X1","Y0","X2","X3"),**

**+ # The source columns**

**+ measure.vars=c("Y1","Y2","Y3"),**

**+ variable.name="FollowUpNo",**

**+ value.name="Y"**

**+ )**

**> head(sim.data2)**

**id X1 Y0 X2 X3 FollowUpNo Y**

**1 1 4.439524 9.398107 19.83624 0 Y1 35.59993**

**2 2 4.769823 9.006301 20.82480 0 Y1 36.81546**

**3 3 6.558708 11.026785 23.05724 1 Y1 48.80235**

**4 4 5.070508 10.751061 22.74696 0 Y1 40.97143**

**5 5 5.129288 8.490833 20.07681 0 Y1 36.58793**

**6 6 6.715065 9.904853 21.13707 1 Y1 49.52075**

**> dim(sim.data2)**

**[1] 1500 7**

1. Provide the spaghetti plot of response vs time. Interpret your Figure with just 1-2 sentences.

**> library(epiDisplay)**

**> sim.data2$year=rep(1:3,each=500)**

**> head(sim.data2)**

**id X1 Y0 X2 X3 FollowUpNo Y year**

**1 1 4.439524 9.398107 19.83624 0 Y1 35.59993 1**

**2 2 4.769823 9.006301 20.82480 0 Y1 36.81546 1**

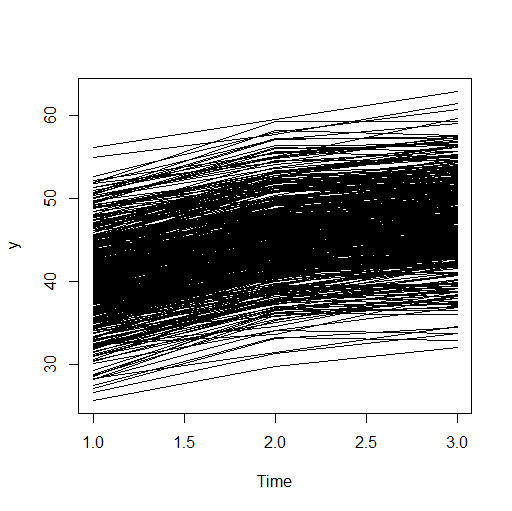
**3 3 6.558708 11.026785 23.05724 1 Y1 48.80235 1**

**4 4 5.070508 10.751061 22.74696 0 Y1 40.97143 1**

**5 5 5.129288 8.490833 20.07681 0 Y1 36.58793 1**

**6 6 6.715065 9.904853 21.13707 1 Y1 49.52075 1**

**>followup.plot(sim.data2$id,sim.data2$year,sim.data2$Y,line.col = "black",xlab="Time",ylab="y")**



**The response increases over time for almost all subjects. It seems to range between 25 and 65.**

**e)** Provide the trellis display of response vs each of the covariates separately. Be careful/neat about the labels, layout etc. of these displays (i.e., make them look nice). Interpret your Figures with just 1-2 sentences.

**> library(lattice)**

**> time=sim.data2$year**

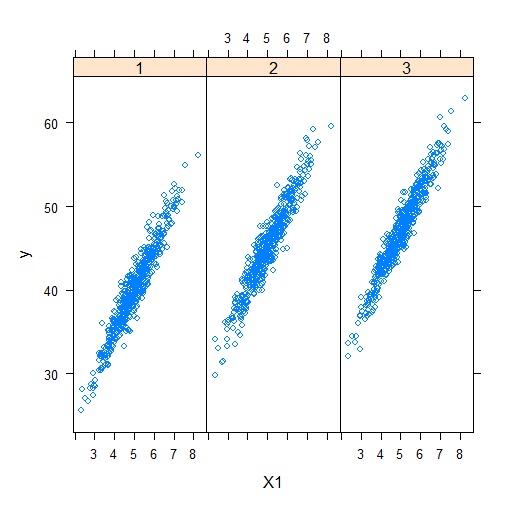
**> xyplot(sim.data2$Y~sim.data2$X1|factor(time),xlab="X1",ylab="y",layout=c(3,1))**

**> xyplot(sim.data2$Y~sim.data2$X2|factor(time),xlab="X2",ylab="y",layout=c(3,1))**

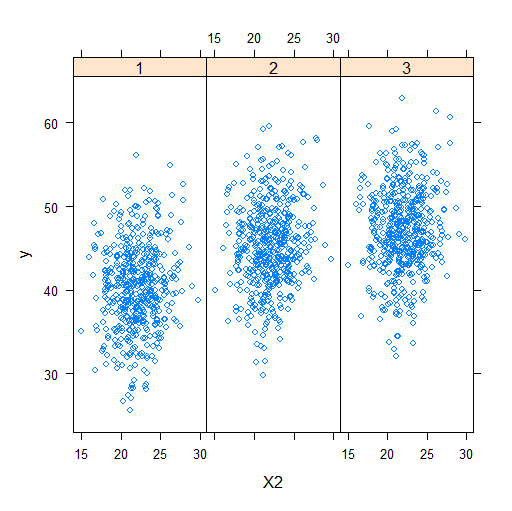
**> xyplot(sim.data2$Y~sim.data2$X3|factor(time),xlab="X3",ylab="y",layout=c(3,1))**

**> xyplot(sim.data2$Y~sim.data2$Y0|factor(time),xlab="Y0",ylab="y",layout=c(3,1))**

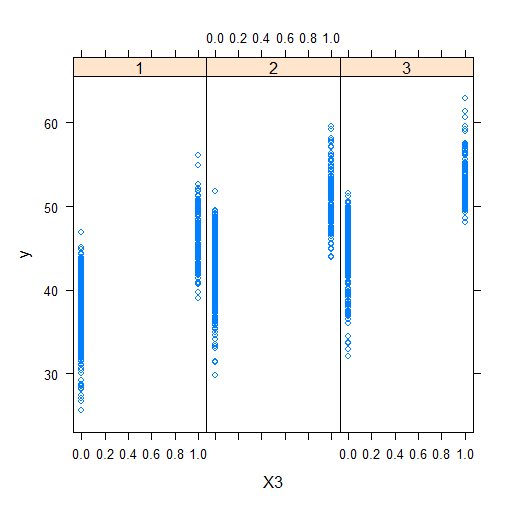
**Note the use of factor and layout options to make figures look nicer.**



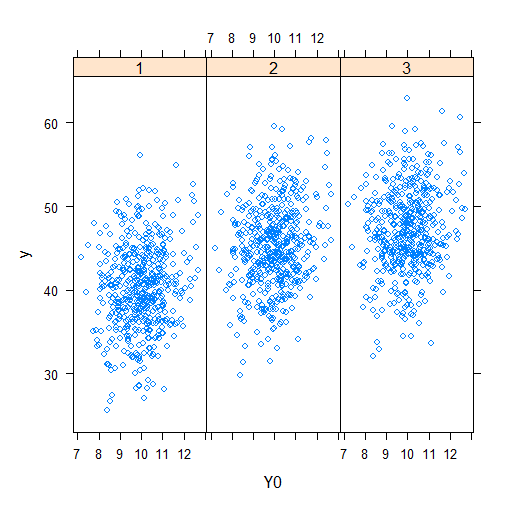
**There is a positive linear relation between response and X1 in all 3 years. We see the increasing trend of response from these plots as well.**



**I don’t expect any significant relation between Y and X2.**



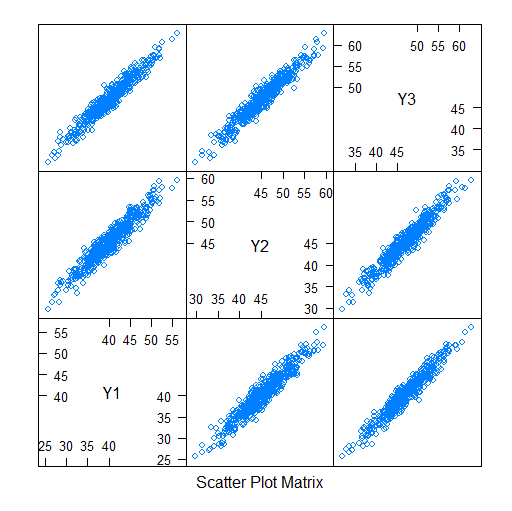
**X3 is taking only 2 values. Higher response values are observed when X3=1 compared to when it is 0.**



**There is no clear relation btw response at baseline and response at later years. Y0 values are much smaller than the Y values at later years.**

**f)** Provide the Draftman’s display. Interpret your Figure with just 1-2 sentences.

> splom(~hw1.sim[,4:6])



**I expect a positive high autocorrelation among responses at different time points**.

**g)** Provide the covariance and correlation matrices for the response at different years. What do you see? Suggest a known variance-covariance structure to be used while modeling this data. Explain why in 1-2 sentences.

**> cor(hw1.sim[,4:6])**

**Y1 Y2 Y3**

**The correlations are almost equal at different lags. The variances at different time points are also almost equal. Because of these, I would suggest exchangeable correlation structure.**

**Y1 1.0000000 0.9535378 0.9635342**

**Y2 0.9535378 1.0000000 0.9606095**

**Y3 0.9635342 0.9606095 1.0000000**

**> cov(hw1.sim[,4:6])**

**Y1 Y2 Y3**

**Y1 25.82273 24.14166 24.36764**

**Y2 24.14166 24.82312 23.81882**

**Y3 24.36764 23.81882 24.76797**

**h)** Now, delete some observations to create a dataset with missing data at random. Let the missing proportion be 10%. Provide code or steps of how you created this dataset. Explore if missingness is random. Report your findings. Apply some imputation techniques (it is up to you which ones you apply) that we learned in the class. Investigate if these methods work nicely for this situation.

**> sim.data.r.na=sim.data2**

**> dim(sim.data.r.na)**

**[1] 1500 8**

**> set.seed(123)**

**> s.na=sample(dim(sim.data2)[1],dim(sim.data2)[1]\*0.10)**

**> sim.data.r.na[s.na,]$Y=NA**

**> sum(is.na(sim.data.r.na))**

**[1] 150**

**> library(finalfit)**

**> explanatory=c("X1","X2","X3","Y0")**

**> dependent="Y"**

**> sim.data.r.na %>% missing\_compare(dependent,explanatory)**

**Missing data analysis: Y Not missing Missing p**

**X1 Mean (SD) 5.0 (1.0) 5.0 (1.0) 0.988**

**Summary stats for missing and non-missing Y are similar. All p-values are >5%. It might be safe to assume that missingness is random.**

**X2 Mean (SD) 22.1 (2.5) 21.8 (2.4) 0.302**

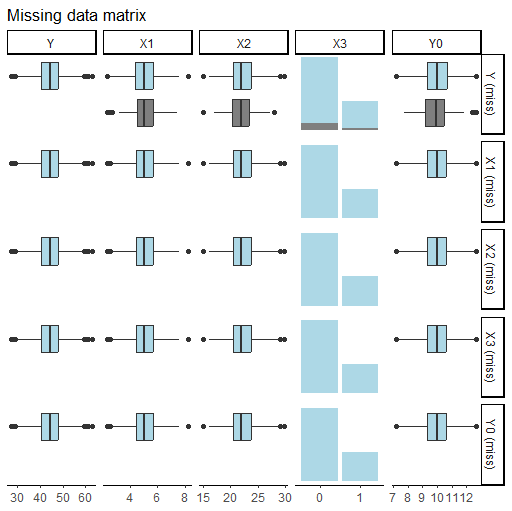
**X3 0 964 (90.0) 107 (10.0) 1.000**

**1 386 (90.0) 43 (10.0)**

**Y0 Mean (SD) 10.0 (1.0) 9.9 (1.0) 0.202**

**> sim.data.r.na$X3=as.factor(sim.data.r.na$X3)**

**> sim.data.r.na %>% missing\_pairs(dependent, explanatory)**



**Distributions of X1 (similarly, X2, X3 and Y0) are very similar for those Y is missing and nonmissing. Again, this implies we might assume missingness is random.**

**> sim.data.r.na.comp=sim.data.r.na[complete.cases(sim.data.r.na),]**

**> dim(sim.data.r.na)**

**Complete case analysis causes loss of info.**

**[1] 1500 7**

**> dim(sim.data.r.na.comp)**

**[1] 1350 7**

**> # converting to wide form with dcast function**

**> sim.data.r.na.wide=dcast(sim.data.r.na, id ~ FollowUpNo, value.var="Y")**

**> head(sim.data.r.na.wide)**

**id Y1 Y2 Y3**

**1 1 35.59993 40.77474 43.08413**

**2 2 36.81546 42.54816 45.09235**

**3 3 48.80235 52.91823 55.27874**

**4 4 40.97143 46.73067 49.32283**

**5 5 36.58793 45.25763 46.31141**

**6 6 49.52075 55.60739 NA**

**> library(longitudinalData)**

**> sim.data.r.na.wide=as.matrix(sim.data.r.na.wide)**

**> Imp1 <- imputation(sim.data.r.na.wide[,2:4],method="locf")**

**> head(Imp1)**

**[,1] [,2] [,3]**

**[1,] 35.59993 40.77474 43.08413**

**[2,] 36.81546 42.54816 45.09235**

**[3,] 48.80235 52.91823 55.27874**

**[4,] 40.97143 46.73067 49.32283**

**[5,] 36.58793 45.25763 46.31141**

**[6,] 49.52075 55.60739 55.60739**

**> error=Imp1-hw1.sim[,4:6]**

**> range(error)**

**[1] -8.490901 8.659303**

**> sum((error)\*\*2) # sum of square error**

**[1] 3368.542**

**> n=sum(abs(error)>0)**

**> n**

**[1] 150**

**> sum((error)\*\*2)/n # mean square error**

**[1] 22.45695**

**> # crossMean: occasion mean**

**> Imp2 <- imputation(sim.data.r.na.wide[,2:4],method="crossMean")**

**> error=Imp2-hw1.sim[,4:6]**

**> range(error)**

**[1] -12.39318 15.46192**

**> sum((error)\*\*2) # sum of square error**

**[1] 4366.475**

**> sum((error)\*\*2)/n # mean square error**

**[1] 29.10983**

**> # trajMean: subject mean**

**> Imp3 <- imputation(sim.data.r.na.wide[,2:4],method="trajMean")**

**> error=Imp3-hw1.sim[,4:6]**

**> range(error)**

**[1] -7.980909 8.659303**

**> sum((error)\*\*2) # sum of square error**

**[1] 3613.44**

**> sum((error)\*\*2)/n # mean square error**

**[1] 24.0896**

**> library(mice)**

**> # first run a naive and initial mice to change the $pred and $meth attributes easily:**

**> ini <- mice(sim.data.r.na, maxit = 0)**

**> pred <- ini$pred**

**> pred["Y", ] <- c(-2, 1, 1, 1, 1, 2, 0)**

**> # 1 for fixed vrs, 2 for random variables, 0 for vr to impute, -2 for ID (ie class)**

**> meth <- ini$meth**

**> hist(sim.data.r.na$Y)**

**> meth <- c("", "", "", "", "", "","2l.norm")**

**> # now lets run the updated mice with the appropriate $pred and $meth**

**> imp.upd <- mice(sim.data.r.na, pred = pred, meth=meth, print = FALSE,m=1)**

**> error=complete(imp.upd,1)$Y-sim.data2$Y**

**I tried many of the methods we covered. I am not expecting you to do all these. Mice (here, conditional imputation, i.e. regression analysis since m=1) gave the best result with the smallest MSE.**

**> range(error)**

**[1] -4.770134 5.878092**

**> sum((error)\*\*2) # sum of square error**

**[1] 631.1811**

**> sum((error)\*\*2)/n # mean square error**

**[1] 4.207874**

**i)** Now, delete some observations to create a dataset with missing data in a non-random fashion. Let the missing proportion be 10%. Provide code or steps of how you created this dataset. Explore if missingness is random. Report your findings. Apply some imputation techniques (it is up to you which ones you apply) that we learned in the class. Investigate if these methods work nicely for this situation.

**> # non-random missingness creation**

**> q90=quantile(sim.data2$Y, probs = 0.9)**

**> q90**

**90%**

**51.5871**

**> sim.data.non.r.na=sim.data2**

**> dim(sim.data.non.r.na)**

**[1] 1500 7**

**> sim.data.non.r.na$Y=ifelse(sim.data2$Y>q90,NA,sim.data2$Y)**

**> sum(is.na(sim.data.non.r.na))**

**[1] 150**

**There are significant differences in the summary stats of all covariates depending on whether Y is missing or not. Clearly, we have a non-random missingness case.**

**> range(sim.data.non.r.na$Y,na.rm=T)**

**[1] 25.58363 51.58151**

**> explanatory=c("X1","X2","X3","Y0")**

**> dependent="Y"**

**> sim.data.non.r.na %>% missing\_compare(dependent,explanatory)**

**Missing data analysis: Y Not missing Missing p**

**X1 Mean (SD) 4.9 (0.9) 6.5 (0.6) <0.001**

**X2 Mean (SD) 22.0 (2.4) 22.5 (2.7) 0.022**

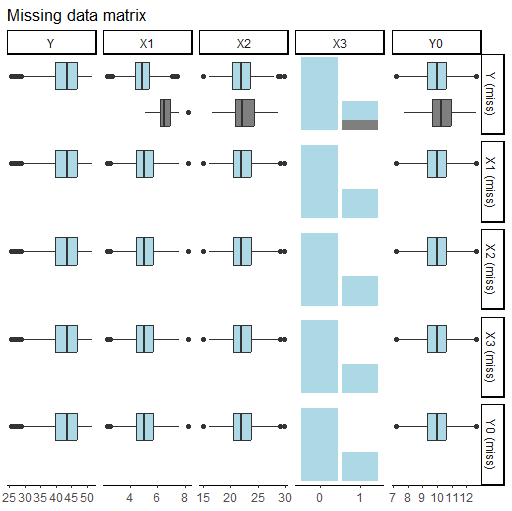
**X3 0 1070 (99.9) 1 (0.1) <0.001**

**1 280 (65.3) 149 (34.7)**

**Y0 Mean (SD) 10.0 (1.0) 10.3 (1.1) <0.001**

**> sim.data.non.r.na$X3=as.factor(sim.data.non.r.na$X3)**

**> sim.data.non.r.na %>% missing\_pairs(dependent, explanatory)**

****

**There are obvious differences in the distribution of covariates when Y is missing or not. This is especially true for X1 and X3.**

**> # converting to wide form with dcast function**

**> sim.data.non.r.na.wide=dcast(sim.data.non.r.na, id ~ FollowUpNo, value.var="Y")**

**> sim.data.non.r.na.wide=as.matrix(sim.data.non.r.na.wide)**

**> # crossMean: occasion mean**

**> Imp2 <- imputation(sim.data.non.r.na.wide[,2:4],method="crossMean")**

**> error=Imp2-hw1.sim[,4:6]**

**> range(error)**

**[1] -17.39792 0.00000**

**> sum((error)\*\*2) # sum of square error**

**[1] 14255.7**

**> n=sum(abs(error)>0)**

**> sum((error)\*\*2)/n # mean square error**

**[1] 95.03798**

**> ini <- mice(sim.data.non.r.na, maxit = 0)**

**> pred <- ini$pred**

**> pred["Y", ] <- c(-2, 1, 1, 1, 1, 2, 0)**

**> # 1 for fixed vrs, 2 for random variables, 0 for vr to impute, -2 for ID (ie class)**

**> meth <- ini$meth**

**> meth <- c("", "", "", "", "", "","2l.norm")**

**> # now lets run the updated mice with the appropriate $pred and $meth**

**> imp.upd <- mice(sim.data.non.r.na, pred = pred, meth=meth, print = FALSE,m=1)**

**> error=complete(imp.upd,1)$Y-sim.data2$Y**

**Conditional mean imputation provided much closer imputed values to true values, compared to occasion mean imputation. LOCF and subject mean imputation was not valid for this data since all of the Y values were missing for some subjects.**

**> range(error)**

**[1] -18.33011 5.70458**

**> sum((error)\*\*2) # sum of square error**

**[1] 3168.113**

**> n=sum(abs(error)>0)**

**> sum((error)\*\*2)/n # mean square error**

**[1] 21.12075**

Note: Your solution is expected to be <=10 pages. I know mine is longer, but I didn’t delete the qs and included as much info as possible for a compact answer.