Biostat 209 Lab R Version

Repeated Measures 2

Lab Summary: In this lab we will learn about the use of mixed model and GEE analysis for numeric and binary outcomes and use of robust for GEE analyses.

Re-analysis of the fecal fat data.

First, load the fecal fat data (fecfat) and reproduce the results in lecture. Make sure to make pilltype a categorical variable first.

```
fecfat$pilltype <- factor(fecfat$pilltype)
fecfat_mixed <- lmer(fecfat ~ pilltype +(1|patid), data=fecfat)
summary(fecfat_mixed)
```

Derive the overall pilltype test using the anova command from the lmerTest package[1].

```
anova(fecfat_mixed)
```

The fecal fat data can also be analyzed using a two-way ANOVA, with factors of pilltype and patient. This can be performed using regression (as we learned in Biostat 208). Use the regression command to do this:

```
two_way_fit <- glm(fecfat~pilltype+factor(patid), data=fecfat)
```

followed by the `anova` command. How do the p-values for the overall pilltype tests compare?

The OAI data

Load the "oai thru 18 months" data. Our goal is to look over 18 months to see if the changes over time in the WOMAC pain score are the same in men and women. First generate some descriptive statistics and a graph. Tabulate results using the data.table package or other means:

```
setDT(oai_thru_18m)
tab_oai <-
    oai_thru_18m[,.(mean_womac=mean(womac_pain),
    N=sum(!is.na(womac_pain))), by=.(sex, visit)]
```

[1] Annoyingly sometimes R does not provide p-values for the t- or F-statistics and makes you compute them manually. The rationale is that the calculation is only valid for large sample sizes. Here is how to get an approximate p-value that is valid in large sample sizes: Take the F-value (which is 6.2574 in this example) and multiply it by the degrees of freedom (listed as "npar" in the output). Then use a chi-square distribution (pchisq calculates the lower tail area of a chi-square, so one minus that gives the upper tail p-value) with the given degrees of freedom to get an approximate p-value:

```
p_val <- 1-pchisq(3*6.2574,3)
```

In this case we clearly do not have large sample sizes, so we would have to take the p-value with a grain of salt. However, it is highly statistically significant, so even if the approximation is a poor one, the conclusions would be unchanged.

```
tab_oai
```

Next a graphical display. First separate the men and women:

```
male_mean <- tab_oai[sex==1]
female_mean <- tab_oai[sex==2]
```

Then graph:

```
      ggplot() +
geom_line(data = male_mean, aes(x = visit, y = mean_womac), color = "blue") +
geom_line(data = female_mean, aes(x = visit, y = mean_womac), color = "red") +
xlab("visit") +
ylab("mean") +
ggtitle("Mean by visit and sex")
```

Next fit a model and do a formal test:

```
womac_mix_fit <- lmer(womac_pain ~
factor(sex)+factor(visit)+factor(sex)*factor(visit)+(1|id)
,data=oai_thru_18m)
summary(womac_mix_fit)
anova(womac_mix_fit)
```

What are the interpretations of each of the coefficients in the model fit? It may help to get the predicted values and look at the values for the various combinations of sex and visit.

Is there a statistically significant difference in how the pain for men and women change over time?

Perform diagnostic checks of the assumptions. Are there serious violations of the assumptions that make a qualitative difference in the analysis?

Logistic regression for the backpain data

Load the backpain data from the web site. It is a hypothetical version of the data from the Korff, et al article described in class. The variables we will be interested in are:

1. Doctor ID number (`doctor`).
2. Whether or not the patient understood the treatment (`undrstnd`).
3. Age in years (`age`).
4. Education divided into three categories (0 means <12 years, 1 means 13-16 years, 2 means ≥ 16 years education). (`educ`).

Using undrstnd as the dependent variable, fit a logistic regression model with predictors age (as a continuous variable) and education (as categorical):

```
logistic_fit <-
      glm(undrstnd ~ age+factor(educ), data=backpain, family="binomial")
summary(logistic_fit)
```

Now let's take account of the clustering using the `gee` command.

```
gee_fit <- glmgee(undrstnd ~ age+factor(educ), data=backpain,
       id=doctor, family=binomial, corstr="exchangeable")
```

We do see a couple of important differences from the previous use of gee. The family= binomial option tells R that the data are binary and also invokes the logistic regression model. How do the results differ between the gee and logit commands?

Which working correlation structure (corstr) is the most reasonable one to use in this case? Use the `glmgee` command (and statistical significance) to decide which predictor variables to include in your final model (You don't have to consider interaction terms). Is practice style statistically significant? Provide an interpretation of the effect of practice style.

When you have reached a final model, compare the use of the logistic (glm) command. How do the coefficients compare? How does the test of practice style compare?

Logistic regression for the OAI data

Next we are going to analyze a binary outcome variable using BMI. Create an overweight and/or obese variable from bmi (defined as bmi > 25). Fit a model using gee and interpret the sex by visit interaction. It may be helpful to run the model separately by sex with the only predictor being a categorical visit variable.

The robust option in GEE analyses

Let's start with a simple regression analysis using the predictors birth order and age at first birth and using the birthweight data.

```
gab_regr <- glm(bweight ~ birthord + initage, data=GAbabies)
```

As we noted previously, this is incorrect because we have not accounted for the clustering by mom. Next, let's see how the gee command can mimic the regression command.

The gee command is given below, and the `corstr="independence"` portion of the command tells R to pretend the data are independent when calculating the estimates even though they are clustered by mom.

```
gab_gee_ind<-glmgee(bweight ~ birthord+initage,
      data=GAbabies, id=momid, family=gaussian, corstr="independence")
summary(gab_gee_ind)
```

How do the estimates and standard errors compare to the straight regression? What is the fit using as the assumed correlation structure (described as "working correlation" in the summary).

Now let's change the `corstr="independence"` portion of the command to exchangeable to see the impact.

```
gab_gee_exch<-glmgee(bweight ~ birthord+initage,
      data=GAbabies, id=momid, family=gaussian, corstr="exchangeable")
summary(gab_gee_exch)
```

What is the form of the correlation structure invoked with "exchangeable"? Did it make a difference with regard to the estimates or standard errors or tests?

By default, the `glmgee` procedure produces the robust standard errors (S.E.). When the robust option is used, the working correlation structure is used for some intermediate calculations (the coefficients), but the correlation within the clusters is estimated from the data. A nice feature of this option is that inferences remain valid even if the assumed correlation structure is wrong. It is not a free lunch however. It works best with lots of small clusters and can give inaccurate estimates when there are only a few clusters or the clusters are large.

The robust option is similar in spirit to assuming nothing about the correlation structure ("unstructured"). However, using "unstructured" first estimates all the pairwise correlations, which can be wasteful. Robust proceeds directly to calculating standard errors of coefficients without that intermediate calculation. Here is a fit using an unstructured working correlation:

```
gab_gee_uns<-glmgee(bweight ~ birthord+initage,
      data=GAbabies, id=momid, family=gaussian, corstr="unstructured")
summary(gab_gee_uns)
```

Finally, try the correlation structure "autoregressive" and specify order equal to 1.

```
gab_gee_AR1<-glmgee(bweight ~ birthord+initage,
      data=GAbabies, id=momid, family=gaussian, corstr="AR-M", Mv=1)
summary(gab_gee_AR1)
```

Although the `glmgee` package defaults to using robust standard errors, you can ask it to produce the model based standard errors as well. Go back to each of the model fits above and extract the model based standard errors. The vcov command will give you the variance-covariance matrix of the estimates. The standard errors are the square roots of the diagonal of that matrix. For example, the model-based and robust standard errors for the `gab_gee_exch` fit can be gotten with the following:

```
exch_model <- sqrt(diag(vcov(gab_gee_exch, type="model")))
exch_robust <- sqrt(diag(vcov(gab_gee_exch, type="robust")))
```

Now, stop and take an assessment of all these fits, focusing on the coefficients and standard errors of birthorder. Did the estimates vary much? Which methods made the most difference in the standard errors and tests? In particular, the unstructured fit might imply that the correlations aren't all the same. How did the exchangeable structure do compared to the unstructured or robust fits?