Biostatistics 208 Lab #6 2/13/14

The purpose of this lab is to give you practice in examining the assumptions of the linear model. The dataset is a 5% random sample from the Study of Osteoporotic Fractures (SOF), a very large prospective cohort study of community-dwelling older women. The rationale for using a relatively small subset of the SOF data is to give you practice in dealing with a dataset small enough that departures from the Normality assumption may be of serious concern. In addition, some of the graphical diagnostics are more challenging to use in small samples. The dataset includes predictors of bone mineral density (BMD) identified in a 1996 SOF paper (Orwoll et al, Axial bone mass in older women, *Ann Int Med*, 1996;124:187-96).

The dataset `lab6.dta` is on the website. The variables and values are already labeled.

Linearity

Linearity checks are simplest when only an unadjusted association with a single continuous predictor is of interest. In that case we can use a LOWESS smooth of the outcome against the single predictor. This estimates the regression line without making the assumption of linearity.

```
lowess bmd age
```

Question 1. Does there appear to be any departure from linearity in the relationship between BMD and age?

In the multi-predictor context, checking on linearity is a bit more difficult, because the univariate relationship between each predictor and the outcome may not accurately reflect what is going on after taking other predictors into account. For linear models, the solution is to smooth the residuals from the adjusted regression model against the continuous predictor, rather than smoothing the outcome itself.

To do this in Stata, we recommend using the component plus residual (CPR) plot, because it provides both a LOWESS smooth (unlike the basic residual-versus-predictor plot obtained using the `rvpplot` command). An additional advantage of the CPR plot is that the slope of the association shows up, unlike the RVP plot. The regression model needs to be run first; like other so-called post-estimation commands, `cprplot` uses information from the most recently estimated regression model.

```
reg bmd age weight
cprplot weight, lowess
```

The downward curvature of the smoothed line suggests using log-transformed weight instead of the measured value; the natural log of weight, variable `lweight`, is already on the dataset. The `nlcom` command is just for practice in interpretation.

```
reg bmd age lweight
cprplot lweight, lowess
nlcom _b[lweight]*log(1.1)
```

Question 2. Does the CPR plot for log-weight look better? If you think that the log-transformed predictor is a worthwhile improvement, how would you interpret the `nlcom` result?

Finally, check whether including a quadratic term in weight improves the fit. The square of weight, variable `weight2`, is already on the dataset. In this case we have given you code to compute the RVP

plot, because with quadratic and other polynomial fits, CPR plots are hard to interpret.

```
reg bmd age weight weight2
predict residuals, resid
rvpplot weight, yline(0) addplot(lowess residuals weight)
```

Stata notes: The `cprplot` command used above did not require us to obtain the residuals directly, but to add the lowess smooth to the RVP plot, we have to do this extra step. In general, the syntax of the command is `predict` *`newvarname, option`*, with the added option specifying what to predict – in this case the residuals. (Recall that the default with linear models is the value of the linear predictor (`xb`).)

Question 3. Does the RVP plot for the quadratic model look better?

Normality of residuals

The normality assumption model for the multi-predictor linear model really applies to the residuals. We can get the residuals from our regression of BMD on age and its square using the predict postestimation command, then use `qnorm` to obtain a normal quantile or Q-Q plot.

```
regress bmd age weight weight2
predict bmdresid, residual
qnorm bmdresid
kdensity bmdresid, normal
```

Question 4. Is there anything to worry about here, given that this is a moderately large dataset?

Exercise energy use (`eeu`) is another variable of interest on the dataset. Check the distribution of the residuals of regression of `eeu` on age, poor self-reported health (`poorhlth`), and gait speed (`gaitspd`).

```
reg eeu age poorhlth gaitspd
predict eeuresid, residual
qnorm eeuresid
kdensity eeuresid, normal
```

Question 5. What do you think about these residuals? Would your judgment be different if we had the full SOF dataset, with over 5,000 observations?

One solution to the right skewness of `eeu` would be to use a normalizing transformation. In datasets of less than 50 observations, transformation might be crucial to maintaining efficiency and making correct inferences. We can assess possible transformations using the `qladder` command. This command tries several different transformations (including the natural log) and presents normal qqplots for the results. To avoid losing the zero values (which will happen for the log and inverse transformations), first add 0.01 to the EEU variable.

```
replace eeu = eeu+0.01
qladder eeu
```

The `qladder` results suggest that log transformation does an OK job for `eeu`.

```
gen ln_eeu = log(eeu)
regress ln_eeu age poorhlth gaitspd
predict ln_eeuresid, residual
qnorm ln_eeuresid
```

Question 6. Does log transformation of EEU adequately address the departure from normality?

Constant variance

Use the following commands to verify the assumptions of constant variance for the regression of log EEU on age, poor health, and gait speed. The plotting commands allow us to check for funnel shapes in the spread of the residuals when we plot them against the fitted values and the two continuous predictors. In addition, we can check for equality of their standard deviations by level of the binary predictor `poorhlth`.

```
regress ln_eeu age poorhlth gaitspd
rvfplot
rvpplot age
rvpplot gaitspd
table poorhlth, c(n ln_eeuresid sd ln_eeuresid)
```

Question 7. Do you find any evidence of "heteroskedasticity" (i.e. non-constant variance)?

Influential points

After running a regression, the `dfbeta` command can be used to obtain a measure of the influence (called "dfbeta") of each data point on the coefficient estimates. We then recommend using a boxplot to detect outlying values among the dfbetas. Here is the code to do this:

```
reg bmd age lweight
dfbeta
graph box _dfbeta_1 _dfbeta_2
```

The boxplot suggests omitting observations with absolute dfbeta values greater than 0.2, as in the example in the book. To identify these observations so that their accuracy can be checked, use these commands:

```
list bmd age lweight _dfbeta_1 if abs(_dfbeta_1) > 0.2 &
~missing(_dfbeta_1)
list bmd age lweight _dfbeta_2 if abs(_dfbeta_2) > 0.2 &
~missing(_dfbeta_2)
```

Stata notes: In the `list` commands, the second if condition (i.e., that dfbeta 1 or dfbeta 2 is not equal to missing) is required because Stata stores missing values as very large numbers; without this condition, missing values would be included in the listing.

The listing shows that the observations with large dfbetas have outlying but reasonable values of either age or log-weight, so are probably not due to data errors. Assuming they are correct, we could rerun the model excluding the influential points as a sensitivity analysis:

```
reg bmd age lweight if abs(_dfbeta_1) <= .2 & abs(_dfbeta_2) <= 0.2
```

Question 8. Are your conclusions affected? How might you report the results of this sensitivity analysis?

Covariate overlap

Suppose we wished to evaluate the independent effect of current estrogen use (`estrogen`) on other outcomes including BMD and EEU. To assess overlap between the participants who do and do not report current estrogen use, we could first check overlap in terms of covariates one by one, as we might find in Table 1 of the typical research paper reporting the results of a randomized trial. To proceed with this, we first need to make a binary indicator current of estrogen use.

```
recode estrogen 1=0 2=1, gen(ht_current)
label values ht_current yesno
tab estrogen ht_current
foreach x in age bmi gaitspd has10 {
   table ht_current, c(mean `x' p5  `x' min `x' p95  `x' max `x')
   }
foreach x in usearms poorhlth calsupp etid nsfx2 {
   tab ht_current `x', row
   }
```

A more comprehensive assessment could be based on an examination of propensity scores. A logistic regression model with the estrogen use indictor as the outcome is used to estimate propensity scores, then the logit propensity scores (i.e. the estimated log odds of using estrogen) are obtained from the fitted logistic model using `predict`. Assume that the chosen model for current estrogen use is a reasonable choice. The Stata statements below fit the model, calculate and plot the scores, and check the difference between their means.

```
mkspline agesp = age, cubic
logistic ht_current agesp* calsupp
predict logit_pscore, xb
twoway (kdensity logit_pscore if estrogen==1, area(1)
lpattern(solid)) (kdensity logit_pscore if estrogen==0, area(1)
lpattern(longdash)), ytitle("Density") xtitle("Logit Propensity
Score") legend(order(1 "Estrogen users" 2 "Estrogen non-users"))
```

Question 9. How would you characterize the degree of overlap in the two estrogen groups? Is the overlap good enough to proceed as usual?

Optional: Linear and restricted cubic splines

In lecture we saw that categorization is one semi-satisfactory way to relax the assumption of linearity. While flexible, categorical transformations are unrealistic in that the fitted regression line is a so-called step-function, whereas nature generally doesn't turn corners.

An improvement is provided by linear splines, which are continuous rather than jumping at the cutpoints, and linear in between. (They do not avoid the problem of how many "knots" (i.e., cutpoints) to use and where to put them, a problem also shared by cubic splines.)

A more sophisticated answer is provided by restricted cubic splines, which are cubic between the

knots, smooth (have continuous first and second derivatives) at the knots, and are linear beyond the extreme knots, to minimize bad behavior in the tails. Use the following code to obtain unadjusted categorical, linear spline, restricted cubic spline, and LOWESS regressions of BMD on BMI.

```
* categorical fit
recode bmi min/18.5=1 18.5001/25=2 25.0001/30=3 30.00001/35=4
35.00001/max=5, gen(bmicat)
xi: reg bmd i.bmicat age i.estrogen
qui adjust age _Iestrogen_*, gen(fitted1) xb
* rerun using Version 12 notation to test for trend, departure from
linearity
qui reg bmd i.bmicat age i.estrogen
* test for linear trend across categories
contrast q(1).bmicat
* test for departure from linearity
contrast q(2/4).bmicat
* linear spline fit
mkspline bmi1 18.5 bmi2 25 bmi3 30 bmi4 35 bmi5 = bmi
xi: reg bmd bmi1-bmi5 age i.estrogen
qui adjust age _Iestrogen_*, gen(fitted2) xb
* test for departure from linearity
testparm bmi*, equal
* restricted cubic splines
capture drop bmisp*
mkspline bmisp = bmi, cubic
xi: reg bmd bmisp* age i.estrogen
qui adjust age _Iestrogen_*, gen(fitted3) xb
* test for overall BMI effect
testparm bmisp*
* test for departure from linearity
testparm bmisp2 bmisp3 bmisp4
```

Stata notes: For all these models, we need to use the earlier "`xi:`" prefix for including categorical variables in a regression model. This is because we want to use the `adjust` command to obtain fitted values after adjustment (i.e. we want to plot the predicted average outcomes as a function of BMI alone to examine linearity, so we need to estimate these values "adjusted" for the presence of the other predictors in the model). For the first model, treating BMI as categorical, we also run it using the new factor notation, so that we can use the new `contrast` command to test for linear trend as well as departure from it.

The first `mkspline` command creates the linear spline variables `bmi1-bmi5`. In the `regress` output for that model, the coefficients for these variables give the slope in each of the regions between the knots, which we placed at BMI values of 18.5, 25, 30, and 35 kg/m2. In the regression using the linear splines, we can assess departure from linearity by running a test of whether the slopes in the five segments of the linear spline regression line are equal, using the following command:

```
testparm, equal
```

Question 10. Is there evidence for changes in the linear spline slopes across regions of the plot?

The second `mkspline` command with the cubic option makes the restricted cubic spline variables, using the default five knots, at the default locations recommended by Harrell (both can be modified

using options to the command – see online help). The command `capture drop bmisp*` is included in case you already made the splines earlier in checking covariate overlap. The capture pre-command keeps things from stopping if `bmisp1-bmisp4` are not on the dataset. We can also test for departures from linearity in this model by assessing joint effects of all but the first spline, which is the linear component in the default Stata setup.

Question 11. Is there evidence for non-linearity in the cubic spline fit?

Now here is code to plot the various fits. As usual, please be careful not to include any carriage returns if you enter the long `twoway` plotting command directly on the command line.

```
twoway (scatter bmd bmi, msize(vtiny)) (line fitted1 bmi, sort c(J)
lp(longdash) lcol(red) lw(medthick)) (line fitted2 bmi, sort
lp(shortdash) lcol(green) lw(medthick)) (line fitted3 bmi, sort
lp(shortdash) lcol(black) lw(medthick)) (lowess bmd bmi, lp(solid)
lcol(blue) lw(medthick)), plotregion(style(none)) scheme(s1color)
ytitle("BMD (gm/cm^2)") legend(order(2 "Categorical" 3 "Linear
Spline" 4 "Cubic Spline" 5 "Lowess") rows(2)) caption(Adjusted for
age and estrogen use) name(splinefit, replace)
```

Question. How do the linear spline and cubic spline fits compare to the categorical and lowess fits?