

## Biostatistics 208, Lab #6 2/11/10

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, *Annals of Internal Medicine*, 1996;**124**:187-96.)

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

### 1 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
```

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. 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.

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

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 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`, with an added option specifying what to predict – in this case the residuals. The default is the value of the linear predictor (`xb`).

Now compute the fitted slope for weights of 40, 60, 80, and 100 kg. With a quadratic model like this the slope isn't a constant, unlike the other models we have considered. In brief, if

$$E[\text{BMD}|\mathbf{x}] = \beta_0 + \beta_1\text{age} + \beta_2\text{weight} + \beta_3\text{weight}^2$$

then the increase in  $E[\text{BMD}|\mathbf{x}]$  per unit increase in `weight` is given by the derivative of the regression function with respect to `weight`:

$$\frac{\partial E[\text{BMD}|\mathbf{x}]}{\partial \text{weight}} = \beta_2 + 2\beta_3\text{weight}$$

Here is the code to do that and to plot the fitted mean trajectory. Remember that the leading single quote is on the key at the upper left of the keyboard below the escape key; if you cut and paste from the handout into a do-file you will have to edit it to make this work.

```
forvalues wt = 40(20)100 {
    display ""
    display "Slope of regression line when weight = 'wt':"
    lincom weight + 2*'wt'*weight2
}
quietly adjust age, gen(fitted)
tway (scatter bmd weight) (line fitted weight, sort)
```

*STATA notes:* The `forvalues wt = 40(20)100` command sets the temporary variable `wt` equal to values from 40 to 100 in increments of 20. Then the command `lincom weight + 2*'wt'*weight2` is run plugging in each of those four values for `weight` in place of `'wt'`. Remember that in a `lincom` statement, variable names refer to the coefficient estimates, not the variables themselves. The two `display` commands are included to make the output easier to follow.

3. Does the RVP plot for the quadratic model look better? How would you present this results of this model?

## 2 Normality of $\epsilon$

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` post-estimation command, then use `qnorm` to obtain a normal quantile or Q-Q plot.

```
reg bmd age weight weight2
predict bmdresid, residual
qnorm bmdresid
```

1. 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
```

2. 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. The log transformation does a good job for `eeu`, though to retain one observation with `eeu = 0` we have to add a small number (say 0.01) to each observation before taking the log.

```
gen leeu = log(eeu + 0.01)
reg leeu age poorhlth gaitspd
predict leeuresid, residual
qnorm leeuresid
```

3. Does log transformation of `EEU` adequately address the departure from normality?

### 3 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`.

```
reg leeu age poorhlth gaitspd
rvfplot
rvpplot age
rvpplot gaitspd
table poorhlth, c(n leeuresid sd leeuresid)
```

1. Do you find any evidence of “heteroskedasticity” (i.e. non-constant variance)?

### 4 Influential points

After running a regression, the `predict` command can also be used to obtain `dfbetas` as a measure of influence of each data point. 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 & _dfbeta_1 ~= .
list bmd age lweight _dfbeta_2 if abs(_dfbeta_2) > 0.2 & _dfbeta_2 ~= .
```

*STATA Notes:* In the `list` commands, the second `if` condition (ie, 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. Ordinarily an ID number could be listed to identify an observation with out-of-range data, but for confidentiality reasons identifiers are not included on this dataset. In this case the observations with large `dfbetas` are so-called high-leverage

points with outlying but reasonable values of either age or log-weight.

Assuming the variables were found to be 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
```

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

## 5 Covariate overlap

Suppose we wished to evaluate the independent effect of poor self-reported health (`poorhlth`) on other outcomes including BMD and EEU. To assess overlap between the participants who do and do not report poor health, we could check overlap in terms of covariates one by one, as we might find in Table 1 of the typical clinical research paper:

```
foreach x in age bmi gaitspd has10 {
  tab poorhlth, sum('x')
}
foreach x in usearms estrogen calsupp etid {
  tab poorhlth 'x', row
}
```

Then we could use a logistic model to estimate propensity scores, and check overlap in the logit propensity scores. I provided code to plot the scores as well as checking the difference between their means. Remember to type the long plotting command without any carriage returns, or put it in a do-file using `///` at line breaks.

```
mkspline bmispl = bmi, cubic
logistic poorhlth age bmispl* gaitspd has10
predict logit_pscore, xb
tab poorhlth, sum(logit_pscore)
tway (kdensity logit_pscore if poorhlth==1, area(1) lpattern(solid))
      (kdensity logit_pscore if poorhlth==0, area(1) lpattern(longdash)),
      ytitle("Density") xtitle("Logit Propensity Score")
      legend(order(1 "Poor or fair health" 2 "Good to excellent health"))
```

1. Is the overlap between the two groups good enough to proceed as usual?

## 6 Optional: bootstrap confidence intervals

An additional diagnostic for departures from normality of the residuals is to compare model-based with bootstrap confidence intervals, and perhaps substitute the latter. The bootstrap repeatedly re-fits the model to samples of the same size drawn *with replacement* from the data. The sample-to-sample differences in the estimated coefficients provide a non-parametric estimate of the sampling variability of the model coefficient estimates. In the simpler version of this procedure, we use bootstrap to get a better estimate of the standard error of each coefficient, then compute a  $p$ -value and CI under a normality assumption; these will be shown in the `regress` output when we add the option `vce(bootstrap)`. The bootstrap  $p$ -value is from a  $Z$ -test of the coefficient estimate divided by its bootstrap SE. However, percentile-based “bias-corrected accelerated” (bca) CIs are more reliable, because they do not depend even to this extent on the normality assumption. We will include the option `bca` in the `bootstrap` command and request enough bootstrap repetitions to make these CIs reliable, then use the additional post-estimation `estat bootstrap, bca` command to obtain them.

Note that the second bootstrap method does not provide a  $p$ -value. Setting the “seed” ensures that the results match when you rerun the procedure; otherwise they will vary slightly from run to run. You can use any positive integer for the seed.

```
set seed 9896
reg eeu age poorhlth gaitspd, vce(bootstrap, reps(1000) bca)
estat bootstrap, bca
```

1. How closely do the model-based, bootstrap normal-based, and bootstrap bca CIs agree?

## 7 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 avoid 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 30.0001/35=4 35.0001/max=5, gen(bmicat)
xi: reg bmd i.bmicat
predict fitted, xb

* linear spline fit
mkspline bmi1 18.5 bmi2 25 bmi3 30 bmi4 35 bmi5 = bmi
reg bmd bmi1-bmi5
* test for heterogeneity in region-specific BMI trends
testparm bmi*, equal
* test for a linear pattern in the region-specific slopes
test 2*bmi1 + bmi2 = bmi4 + 2*bmi5
predict fitted2, xb

* restricted cubic splines
capture drop bmisp*
mkspline bmisp = bmi, cubic
reg bmd bmisp1-bmisp4
* test for any BMI effect
testparm bmisp*
* test for non-linearity
testparm bmisp2 bmisp3 bmisp4
predict fitted3, xb
```

*STATA notes:* 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  $kg/m^2$ . In the regression using the linear splines, we can run a test of whether the slopes in the five segments of the linear spline regression line are equal, using the `testparm, equal` command. In addition we can test for trend in the slopes (i.e., is the slope linearly decreasing or increasing across the five BMI regions?).

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.

*1. Is there evidence for changes in the linear spline slopes across regions of the plot? Is there evidence of a linear trend in those slopes? Is there evidence for non-linearity in the cubic spline fit?*

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. If you write it in a do-file, then run it, you can use `///` to split the lines.

```
twoway
  (scatter bmd bmi, msize(vtiny))
  (line fitted bmi, sort lpattern(longdash) lcolor(red) lwidth(medthick))
  (line fitted2 bmi, sort lpattern(shortdash) lcolor(green) lwidth(medthick))
  (line fitted3 bmi, sort lpattern(shortdash) lcolor(black) lwidth(medthick))
  (lowess bmd bmi, lpattern(solid) lcolor(blue) lwidth(medthick)),
  plotregion(style(none)) scheme(s1color) ytitle("BMD (gm/cm^2)")
  legend(label(2 "Categorical") label(3 "Linear Spline")
  label(4 "Cubic Spline") label(5 "Lowess") rows(2))
```

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