resids.sas: Explanation of code

Goals of code:

- Draw QQ plots for each group and overall
- Save residuals for each observation for future use

Note: the explanation focuses on the new things in this code. Ask if you have any questions about the data step and the proc sort.

SAS provides two ways for drawing QQ plots (and other model diagnostic plots). I use the hamburger data in hamburger.csv to illustrate both ways. The two ways are:

1. Use plotting options associated with an analysis proc
2. Save residuals from an analysis, then plot those

Using plotting options
To get QQ plots for each group of observations: `proc ttest`: Includes QQ plots for each group by default. These are QQ plots of the Y values, but when considering one group in isolation, the residuals are just the observations shifted by their average.

To get one QQ plot for all observations: `proc glm plots=diagnostics(unpack)`. This needs to be done using the residuals, as explained in lecture. This requires shifting from proc ttest to proc glm, which is a more general model fitting procedure. This option gives you lots of plots. The fourth plot is the QQ plot. We will discuss many of the others later in the semester. If you omit `(unpack)`, you get all the plots in a much more compact panel display.

Writing a t-test as a proc glm model: To save residuals, you need to use one of the more general procedures for data analysis. We will use proc glm extensively in a few weeks. That fits all sorts of more complicated models. We will use it today to fit the t-test model because glm provides options to save all sorts of results, including the residuals and the predicted values.

proc glm can fit both a regression model (discussed in a few weeks) and an ANOVA model (comparison of two or more group means, discussed in a week or so). The t-test assuming equal variance is an analysis of variance (ANOVA) with just two groups. What model proc glm fits depends on two statements, the class statement and the model statement. The class statement must come first, but it is easier to discuss the model statement first.

model cfu = treatment; Define the response variable and the model
The term to the left of the = identifies the response variable (cfu). The stuff to the right of the = defines the model you want SAS to fit. This model can have many terms and can include regression

effects or group mean effects. To get a t-test, we want to fit a model with one mean for the control treatment and a second mean for the active treatment. Treatment here and in the class statement says that the mean cfu depends on the treatment group.

class treatment; treatment defines groups
When you include a variable in a class statement, SAS will use it to define groups and will estimate a mean for each group. If you omit the class statement or name the wrong variable(s), any reference to trt in the model statement will define a regression (straight line).

`class treatment` says that values of the treatment variable identify groups. glm will fit a model with one mean for each group.
`model cfu = treatment` identifies the response variable (cfu) on the left-hand side of the = sign. The right-hand side identifies the model.

Saving residuals from a proc glm model: `proc glm; output out= r=;`
To get residuals saved, you need to add an output statement to the proc glm. This has at least two parts: `data=resids` identifies a data set in which to store the results. That data set is called by the name you give (here resids). `r=resid` requests the residuals (`r=`) and names the variable to store them (here resid). I also generally request predicted values (`p=yhat`) in a second variable. We don't need predicted values here; we will use both residuals and predicted values later.

The new data set (here called resids) includes the requested variables and all the variables in the input data set. That means that the resids data set includes the treatment variable.

Plotting a QQ plot of the residuals: `proc univariate; qqplot resid; run;` Requests a qqplot of the variable named on the qqplot statement. The default is a QQ plot that assesses a normal distribution.