1 Dataset to download

Please download the datasets 175RNA_fup.dat and 175status.dat from the course web page after reading the 175.info files.

I purchased the data from the virology substudy of an AIDS clinical trial (ACTG 175) from the National Technical Information Service (NTIS). The data were provided as 9 separate files on a floppy disk. I was interested in relating the longitudinal trajectories of patients' RNA concentrations to their clinical disease status.

Today we will use the data file that contains only the follow-up values of RNA. (The baseline values are contained in the baseline data file. To evaluate HIV-1 concentration, blood was drawn from the patient at each follow-up visit. The specimen was then kept frozen. At certain times during the study, specimens were thawed and assayed. We wish to determine the average number of days between blood collection and assay performance.

2 Reading the dataset and doing arithmetic with dates

Internally, SAS stores dates as numeric variables. This makes it easy to calculate the elapsed time between two dates. The value for any date is the number of days from ??/??/?? until that date. Date *informs* and *formats* enable SAS to correctly read dates that are stored in different forms in data files, and to print dates in ways that are meaningful to people.

On p. 102, your textbook has a table of SAS date *informs* and their meanings.

The documentation that came with the disk states:

"The variables in RNA_fup.dat are in the following locations:

*1 pidnum
*7 specdate date7.
*15 assaydt date7.
*23 rna;

Let's try reading in the dataset using their specifications. In addition, we will compute the number of days that each specimen was frozen.

```
options linesize = 75
formchar = "|----|+|---+=|</->*/";
data rnafup ;

* infile '/group/ftp/pub/kcowles/datasets/175rna_fup.dat' ;
 * uses a built-in SAS informat to
 * enable SAS to read the dates as they
 * appear in the file ;
timefrz = assaydt - specdate ;  * calculate elapsed days;
format specdate assaydt mmddyy8. ;  * use a built-in SAS date format to
control printing of dates in output ;
run ;
```

Let's check the log. Uh-oh. The problem is that the pidnums toward the end of the data file are 6 digits long. This means that there is no space between the last digit in the pidnum and the start of the specdate. SAS thinks the specdate is just a continuation of the pidnum and complains because of the invalid characters (the slashes). We can solve this by telling SAS that the pidnum variable is only 6 digits long.

```
data rnafup ;
*infile '/group/ftp/pub/kcowles/datasets/175rna_fup.dat' ;
* uses a built-in SAS informat to
* enable SAS to read the dates as they
* appear in the file ;
timefrz = assaydt - specdate ;  * calculate elapsed days;
format specdate assaydt mmddyy8. ;  * use a built-in SAS date format to
control printing of dates in output ;
run ;
```

```
proc print data = rnafup (obs=25) ;
run ;
```
3 Selecting a single observation for each subject

Datasets like this one, with multiple records for each subject, are common. For example, your bank probably has a file with records on every transaction for every account number. Certain kinds of reports or analyses require extracting either the first record or the most recent record for each subject.

In SAS we can do this in two steps. First we must sort the dataset in such a way that all the records for each subject are grouped together, and that within subject, the records are in date order.

```
proc sort data = rnafup ;
  by pidnum specdate ;
run ;
```

Next we need to create a new dataset that contains only the most recent (last in date order) record for each patient.

```
data recent ;
  * name the new dataset ;
set rnafup ;
  * find the data for the new dataset in the existing dataset called patients ;
by pidnum ;
  * recognize that the incoming data is sorted by pidnum ;
if last.pidnum ;
  * keep only the last record from each pidnum group ;
run ;

proc print data = recent (obs = 25) ;
run ;
```

4 Getting summary statistics on each patient

Suppose we want to create a dataset that has a single record for each patient, but we want that record to contain the number of RNA measurements and the average RNA value from all his/her records. We can use `proc means` to create such a dataset.

First, let’s just have `proc means` display the appropriate output.

```
proc means data = rnafup mean maxdec = 2 ;
class pidnum ;
var rna ;
run ;
```

```
The MEANS Procedure
Analysis Variable : rna

               N    Mean
     pidnum  Obs
               ------------------------------------
     10341    5  1905.80
     10343    5  1340.40
     10361    2 19394.50
     10364    2  81539.00
     10378    5  1453.60
     10386    4  20716.00
     10389    3  89694.33
```

Now we will get SAS to create the corresponding dataset, that has `pidnum` as the unit of
observation instead of visit.

```
proc means data = rnafup noprint nway ;
  * noprint: don't print results of proc ;
  * nway: give results only for individual
    pidnums; don't give grand mean ;
  class pidnum ;
  var rna ;
  output out = rnasum * out = < name of new dataset > ;
    * which summary stats to include,
    and what to name variables containing
    them ;
run ;
```

```
proc print data = rnasum ;
  title 'rnasum dataset' ;
run ;
```

```
rnasum dataset 18
Obs pidnum _TYPE_ _FREQ_ m_rna
1 10341 1 5 1905.80
2 10343 1 5 1340.40
3 10361 1 2 19394.50
4 10364 1 2 81539.00
5 10378 1 5 1453.60
6 10386 1 4 20716.00
7 10389 1 3 89694.33
8 10432 1 5 27132.40
9 10456 1 5 214.20
10 10462 1 5 0.00
11 10478 1 2 2950.80
12 10889 1 2 2549.50
13 10894 1 5 2950.80
14 10896 1 2 1752.50
15 10896 1 2 1752.50
```

The new _FREQ_ variable gives the number of observations in the original dataset for each value of the class variable.

We can have proc means create output datasets with more than one summary statistic and/or more than one variable summarized. For example, suppose we wanted the sample means and sample standard deviations of both rna and timefrz.

```
proc means data = rnafup noprint nway ;
  var rna timefrz ;
  output out = rnasum
    mean = m_rna m_frz stdev = sd_rna sd_frz ;
run ;
```

```
proc print data = rnasum (obs = 15);
run ;
```

```nasum dataset 19
14:04 Sunday, June 15, 2003
Obs pidnum _TYPE_ _FREQ_ m_rna m_frz sd_rna sd_frz
1 10341 1 5 1905.80 812.6 954.27 281.003
2 10343 1 5 1340.40 833.2 1139.15 292.497
3 10361 1 2 19394.50 1112.0 2693.37 49.497
4 10364 1 2 81539.00 1129.5 16564.68 60.104
5 10378 1 5 1453.60 845.8 525.53 282.224
6 10386 1 4 20716.00 942.0 36018.97 231.318
7 10389 1 3 89694.33 1071.0 21191.53 174.860
8 10432 1 5 27132.40 857.2 16464.42 276.460
9 10456 1 5 214.20 871.8 325.47 285.361
10 10462 1 5 0.00 855.2 0.00 300.345
11 10478 1 2 370633.00 1157.0 141097.50 59.397
12 10478 1 5 3123.60 811.6 1521.21 280.420
13 10889 1 5 2549.50 931.0 2284.66 435.578
14 10894 1 5 2950.80 898.0 2452.18 288.628
15 10896 1 2 1752.50 1042.0 2478.41 237.588
```

Combining data from two separate datafiles in order to perform a statistical test

Suppose we wanted to test whether the mean RNA during follow-up was significantly different in patients in different treatment groups. The mean RNA value for each patient is in our rnasum dataset, but the treatment assignment is not.

We need to merge the rnasum dataset with data from the status data file.

First, we need to read in the 175status.dat datafile. Again we can look at the documentation for guidance in the input statement. We already know we'll have to add a numeric informat for the pidnum variable.

```
data status ;
  infile '/group/ftp/pub/kcowles/datasets/175status.dat' ;
  input @1 pidnum 6.
```

The new _FREQ_ variable gives the number of observations in the original dataset for each value of the class variable.

We can have proc means create output datasets with more than one summary statistic and/or more than one variable summarized. For example, suppose we wanted the sample means and sample standard deviations of both rna and timefrz.
@7 strtdt date7.
@15 crsdt date7.
@23 randdt date7.
@31 offtrdt date7.
@39 offtrsn
@42 otdesc1
@114 otdesc2
@186 offsdt date7.
@193 offfrsn
@196 osdesc
@247 deathdt date7.
@255 trt
@258 crstart
@260 lstcd4dt date7.
@269 crsftype;
format strtdt crsdt randdt offtrdt offsdt deathdt mmddyy10. ;
run ;
proc print data = status (obs = 15) ;
title 'first try at status dataset' ;
run ;
So we have to try to figure out where the variables we need actually are. Here's what I came up with after quite a bit of effort. Note that the last two variables that supposedly are in the dataset were cut off.
data status ;
infile '/group/ftp/pub/kcowles/datasets/175status.dat' ;
input
$1 pidnum 6.
@7 strtdt date7.
@15 crsdt date7.
@23 randdt date7.
@31 offtrdt date7.
/*@39 offtrsn $3.
@42 otdesc1 $71. */
@39 otdesc1 $70.
@112 otdesc2 $70.
@181 offsdt date7.
@189 offfrsn $2.
@191 osdesc $51.
@242 deathdt date7.
@250 trt $2.
@252 crsftype $2.
/*@260 lstcd4dt date7.
@269 crsftype; */
format strtdt crsdt randdt offtrdt offsdt deathdt mmddyy10. ;
run ;
proc print data = status (obs = 15) ;
title 'second try at status dataset' ;
run ;
Much better! The variables we need the most, pidnum and trt are read just fine.
Now we need to merge the rnasum dataset and the status dataset by matching the pidnum value. First we have to tell SAS that both datasets are already in order by pidnum. (If this were not the case, we would have to sort them by pidnum.
data rnasum ;
set rnasum ;
by pidnum ;
run ;
data status ;
set status ;
by pidnum ;
run ;
Now we can do the merge to create a new dataset.
data combo ;
merge status (keep = pidnum trt) rnasum ;
by pidnum ;
run ;
proc print data = combo (obs = 15);
title 'combo' ;
run ;
combo 15:11 Sunday, June 15, 2003 23
Obs pidnum trt _TYPE_ _FREQ_ m_rna m_frz sd_rna sd_frz
1 10198 D . . . . . .
2 10341 D 1 5 1905.80 812.6 954.27 281.003
3 10343 C 1 5 1340.40 833.2 1139.15 292.497
4 10361 B 1 2 19394.50 1112.0 2693.37 49.497
5 10364 C 1 2 81539.00 1129.5 16564.68 60.104
6 10378 A 1 5 1453.60 845.8 525.53 282.224
7 10386 C 1 4 20716.00 942.0 36018.97 231.318
8 10389 B 1 3 89694.33 1071.0 21191.53 174.860
A check of the log and a look at the first few observations convinces us that things look OK.

### 6 Doing the analysis

We have 4 different treatment groups. What statistical procedure could we use to determine whether this sample data suggests that the population means of follow-up RNA are the same in the four populations whom these patients are drawn from?