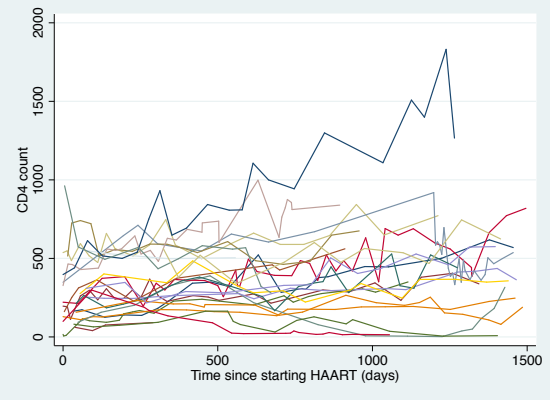
Longitudinal Data
Fall 2015

Chapter 2

Exploratory (Graphical) Analysis of Longitudinal Data



Instructors

Nick Jewell (*jewell@berkeley.edu*)

GSI

Robin Mejia (*mejia@nasw.org*)

Longitudinal Data Graphics

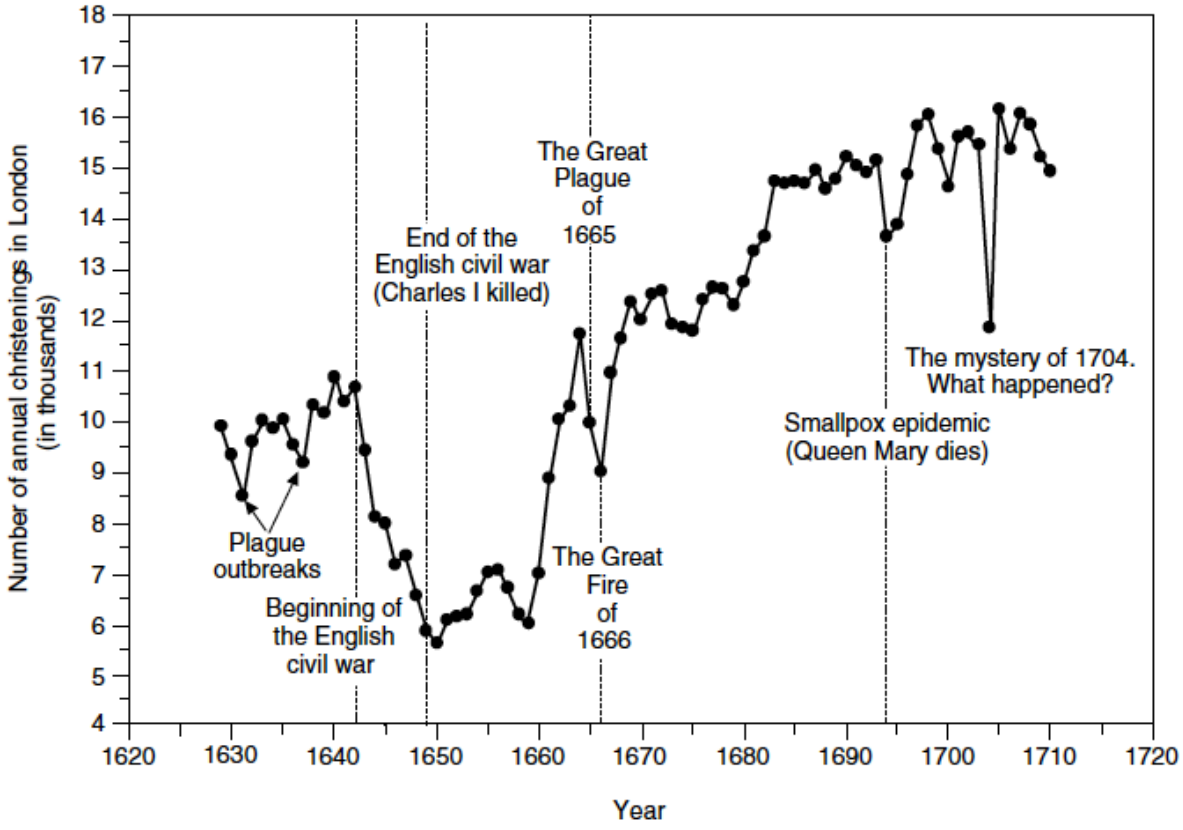


Figure 1 A plot of the annual christenings in London between 1630 and 1710 from the London Bills of Mortality. These data were taken from a table published by John Arbuthnot in 1710

Longitudinal Data Graphics, a History

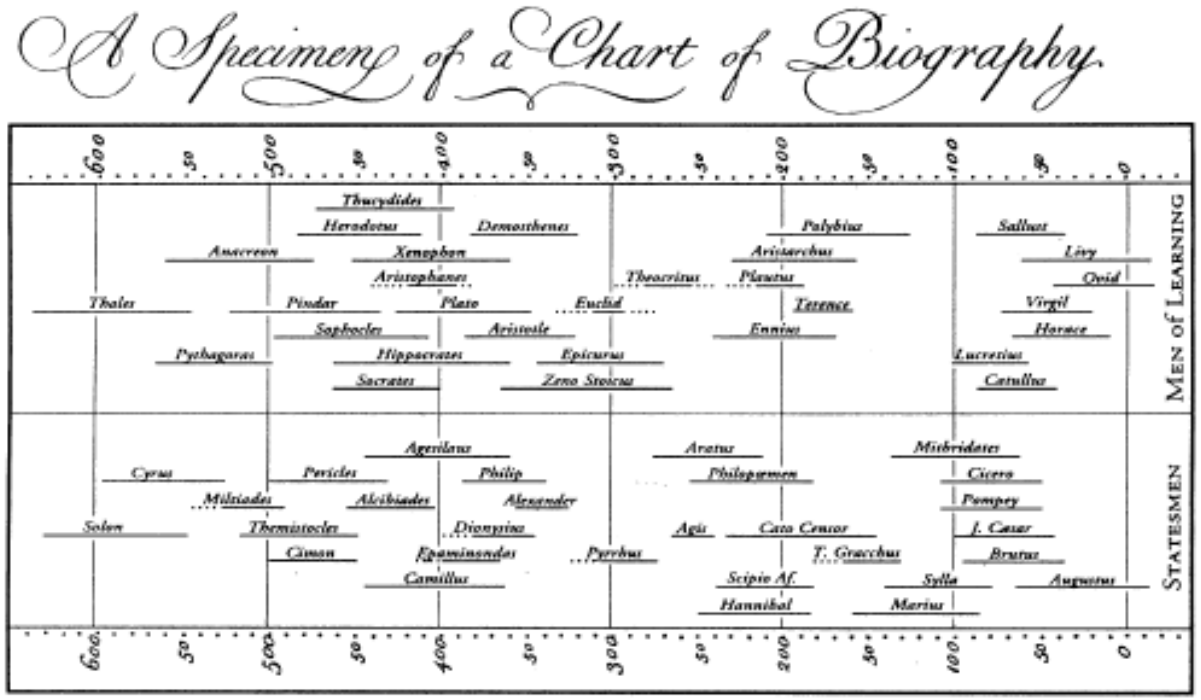


Figure 3 Lifespans of 59 famous people in the six centuries before Christ (Priestley, [6]). Its principal innovation is the use of the horizontal axis to depict time. It also uses dots to show the lack of precise information on the birth and/or death of the individual shown

Longitudinal Data Graphics, History

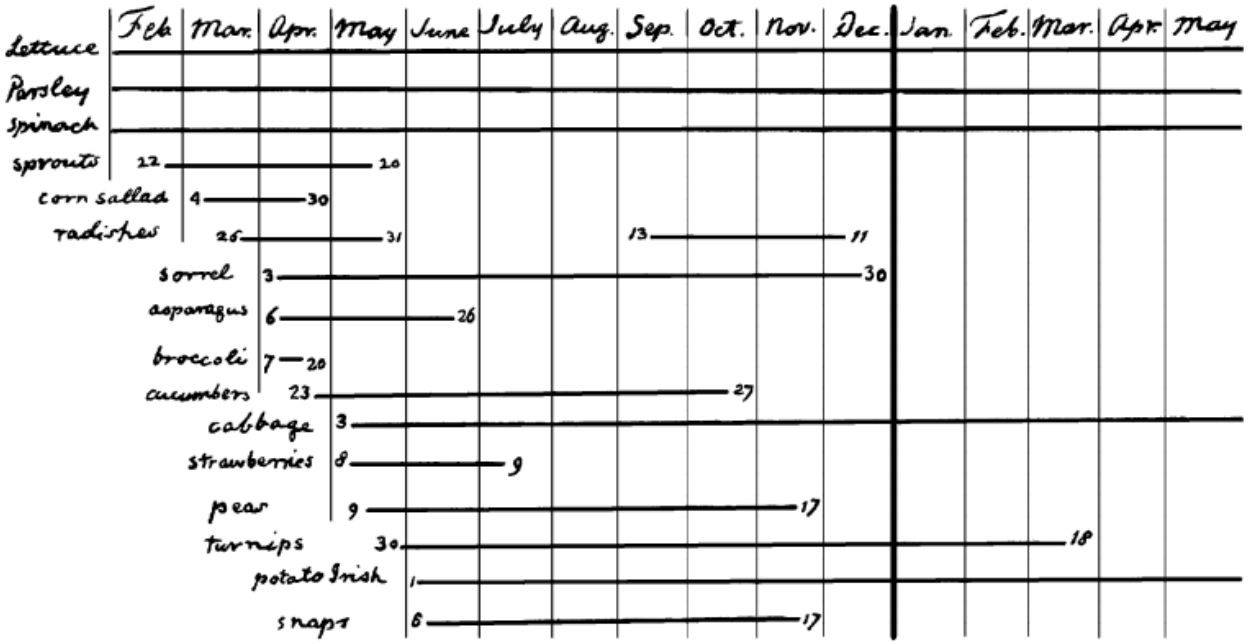
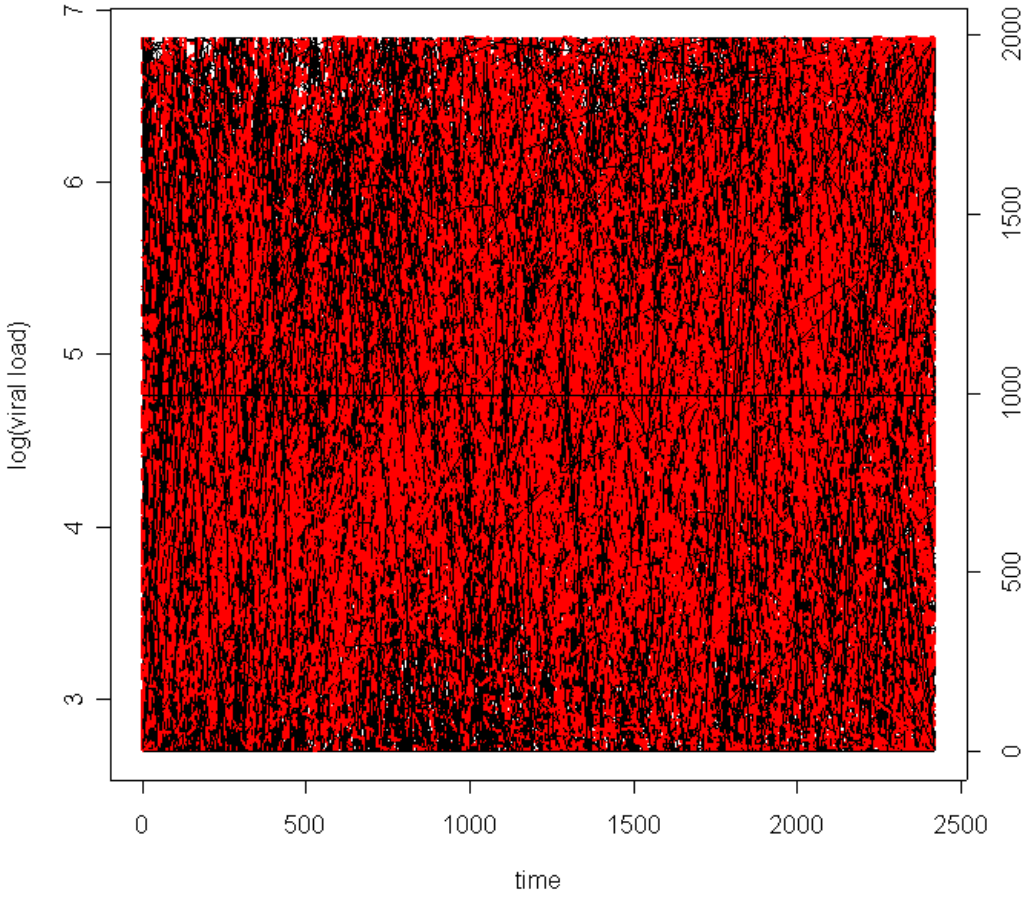


Figure 5 An excerpt from a plot by Thomas Jefferson showing the availability of 16 vegetables in the Washington market during 1802. This figure is reproduced, with permission, from Froncek ([3], p. 101)

Viral load and CD4 count – San Francisco General Hospital

- Data from AIDS clinic at SFGH.
- Collected from 1996-2000.
- Data on 483 subjects that were followed from beginning of HAART.
- Both viral load and CD4 count were measured irregularly over an average of 3 years – number of measurements varied from 11 to 89.

Plotting All the Data (both CD4 and viral load)



Graphical display of longitudinal repeated measures data

- Displaying longitudinal data can present a greater challenge than the analysis of such data.
- Standard methods exist for survival data (see Chapter 8), such as plotting Kaplan-Meier curves, and we may discuss those in some detail later.
- We concentrate mainly today graphical display of repeated measures data - that is when an outcome (such as CD4 count) is measured repeatedly over time on an individual.

Graphical display of longitudinal repeated measures data

- The challenge is to highlight potentially meaningful patterns among messy and abundant data.
- Because the data is longitudinal, interest will often focus on trends in outcomes over time.
- However, other relationships are also of interest (such as changes in outcomes versus changes in explanatory variables).
- The optimal graph will be a function of the question being addressed, and thus there is no universally best plot to display longitudinal data.

Plotting all of the Data (just CD4 this time) using STATA

- Use a STATA user-written program called overlay (not part of the package but must be installed).
- First we just look at CD4 count versus time (days after beginning of HAART).
- We want to plot the CD4 values versus time by id, connecting the points.
- In notation, plots of Y_i vs. T_i (time), for a random subset of i.

A portion of the data set

```
list id etime vl cd4 if etime >=0
          (i)     (Tij)            (Xij)     (Yij)
          id      etime              vl       cd4

 3.        1         39             500        45
 4.        1        137           83370       119
 5.        1        147               .       113
 6.        1        179           79580        74
 7.        1        187               .        95
 8.        1        214               .       120
31.        2          0          239148       196
32.        2          7            4256       369
33.        2         13            6379       353
34.        2         27            1789       474
35.        2         55             623       425
36.        2        111              20       493
37.        2        139              20       464
38.        2        167             139       448
39.        2        195              20       427
40.        2        223              20       460
66.        3         84          501300         .
67.        3        146          260500        41
68.        3        189           99360        53
69.        3        244          217700        31
70.        3        286          460800        32
71.        3        377          457100        26
84.        4        212          104700        79
85.        4        237           84880        81
86.        4        303          177700        59
```

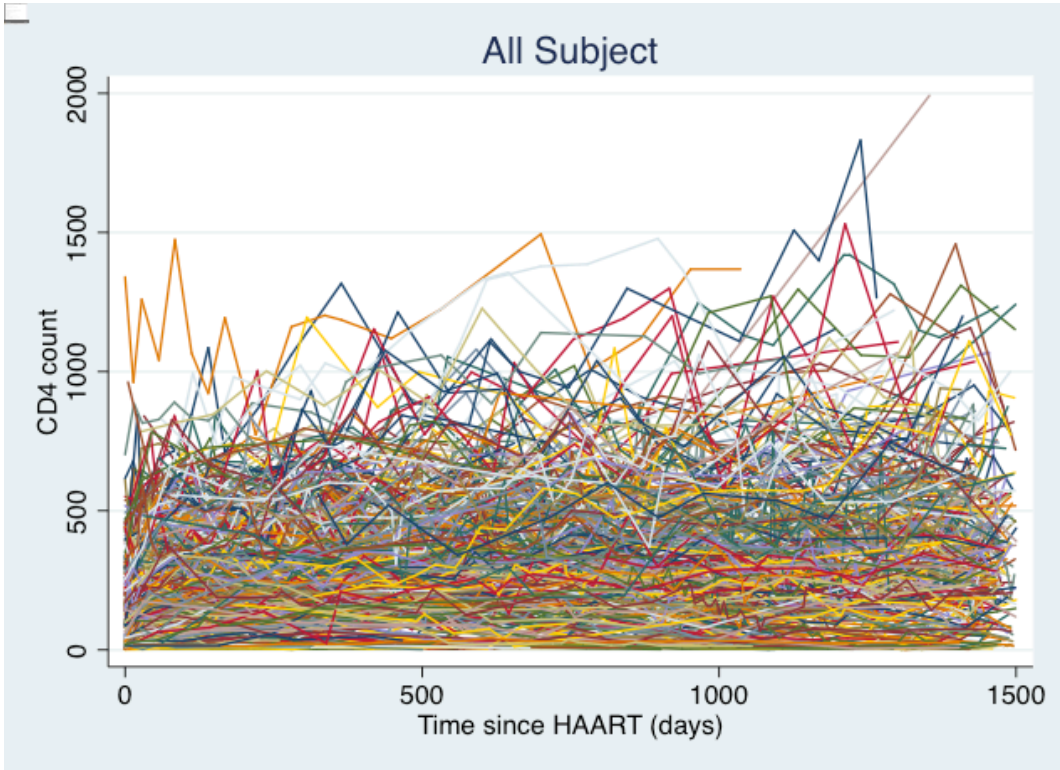
Plotting all of the Data, cont.

- STATA command is:

```
#delimit ;
xtline cd4, i(id) overlay t(etime)
xlab(0(500)1500) ylab(0(500)2000)
  ti("All Subject") legend(off);
```

- i(id) defines the unit
- cd4 is the outcome (y-axis)
- t(etime) indicates the time (x-axis)

Resulting Graph



Data Reduction Methods

- The first idea is to select a relatively small number of subjects whose data provide a good summary of the patterns of interest.
- The simplest method is simply a random sample of the original subjects.
- Most statistical programs have a way to generate a random sample

Syntax for random set of data (5% of id's)

- Below just one way to do it.
- Sort by time within id

```
sort id etime
```

- Generates a random uniform number only in first row of each id

```
by id: gen rr = uniform() if _n==1
```

- Get's the rank of the random number only in first row of each id

```
egen rankrr = rank(rr)
```

- Trick to fill-in all the other rows for the id with the rank of the random number in the first row

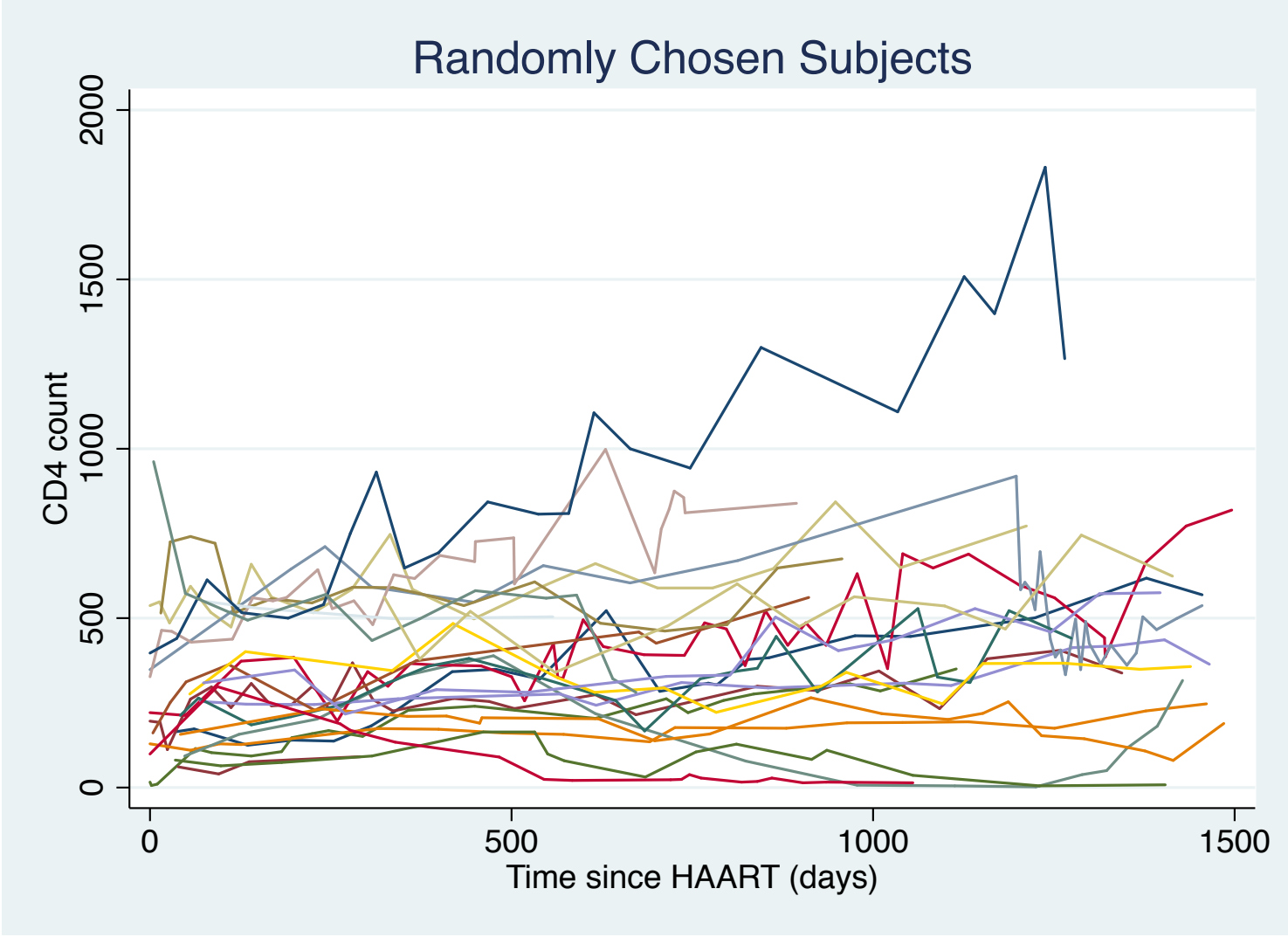
```
egen mnrr = mean(rankrr), by(id)
```

Syntax for random set of data

- Only plots those where the rank is < 5% of the total number of id's - There are 483 ID's, and 5% of 483 = 24

```
#delimit ;
xtline cd4 if mnrr < 24, i(id) overlay
  t(etime) xlab(0(500)1500) ylab(0(500)2000)
  ti("Randomly Chosen Subjects") legend(off);
```

Random subset of subjects plotted



Selecting to represent quantiles of a summary parameter.

- The problem with a small random draw of subjects is that it might not (evenly) represent the set of responses with time.
- For instance, one might want to represent those subjects with lowest average CD4, middle average CD4 ....
- Does not have to be based on average CD4 – could rank subjects by estimated median (or other quantiles), area under the curve, variability, etc.

Plotting subjects evenly spread over average CD4 count

- Get the average CD4 count for each subject.
- Rank subjects based on the average CD4 count (smallest to largest).
- If you want to plot only *k* of the subjects and there are *m* subjects, then only take only every (*m/k*)th subject on list. For instance, if you want to plot 20 subjects and there are *m=200* subjects, take every 200/20 or 10th subject in ranked list and plot their CD4 count versus time.

Syntax for plotting based on subjects ranked by mean(CD4)

- Gets the average mean count CD4 by subject:

$$\hat{\mu}_i = \bar{Y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij} = \frac{1}{n_i}\mathbf{1}^T\mathbf{Y}_i$$

```
egen avecd4 = mean(cd4), by(id)
```

- Finds ranks of subjects by average cd4:

```
sort id etime

quietly by avecd4 id: gen idcnt = _n

egen ravecd4 = rank(avecd4) if idcnt==1,
  unique
```

Syntax for plotting based on subjects ranked by mean(CD4), cont.

- Defines a variable that chooses only every *m/k* = 483/20≈24 subjects in ranked list (resulting in a total of 20 subjects).

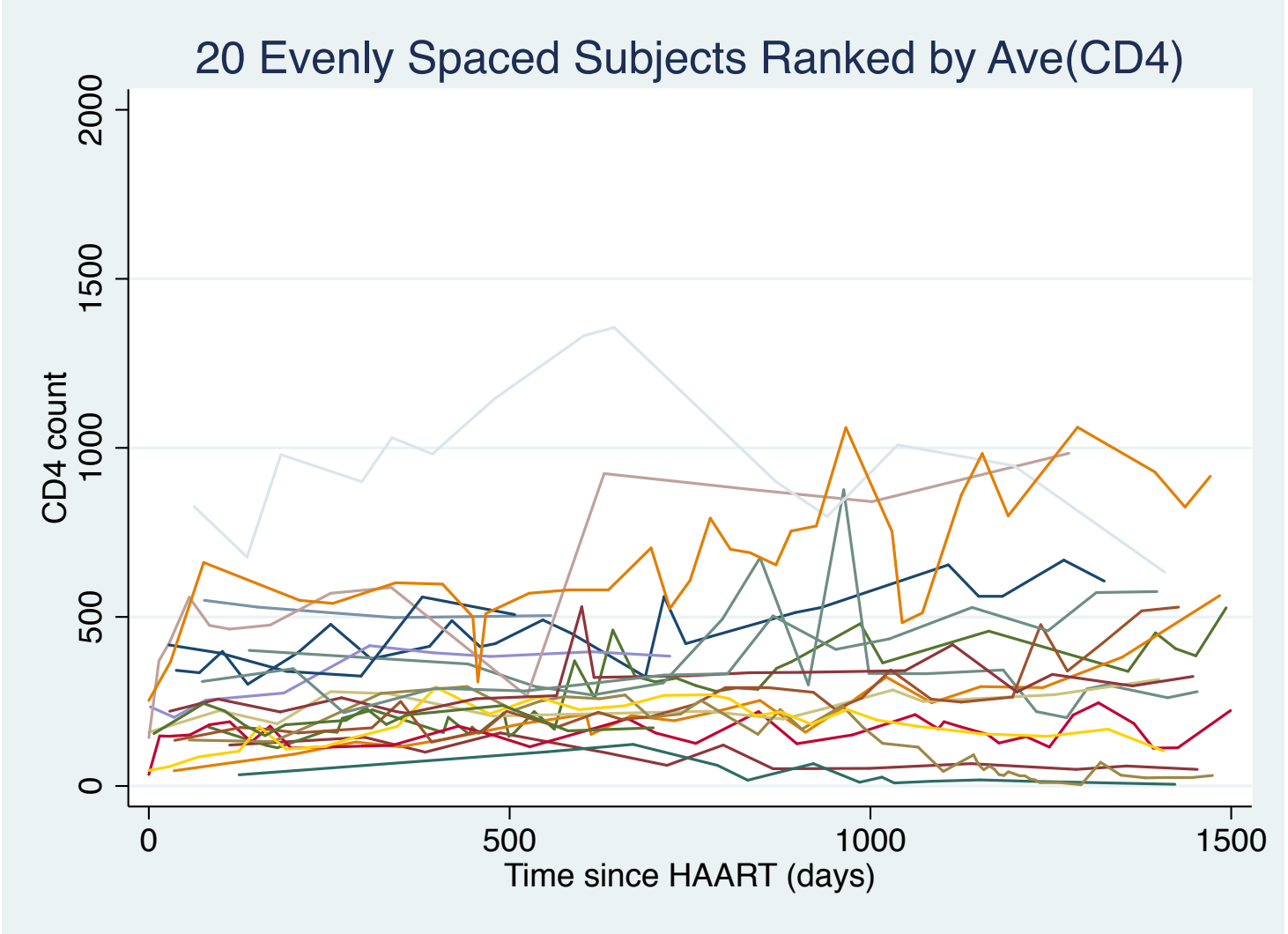
```
* A trick to get the rank for a subject on all
  lines (observations) for that subject

egen mr = mean(ravecd4), by(id)

* Maximum rank (just number of subjects, or m)
egen maxrmr = max(mr)

* m/k
gen ii = int(maxrmr/20)

* A trick to only get every m/k subjects plotted
gen i2 = int(mr/ii)-mr/ii
gen bb = abs(i2) < 0.0001
* or equivalently
capture drop bb
gen bb = mod(mr,ii)==0
```

Syntax for plotting based on subjects ranked by mean(CD4), cont.

- Plotting commands

```
#delimit ;
xtline cd4 if bb==1, i(id) overlay t(etime)
  xlab(0(500)1500) ylab(0(500)2000) ti("20
  Evenly Spaced Subjects Ranked by Ave(CD4)")
  legend(off);
```

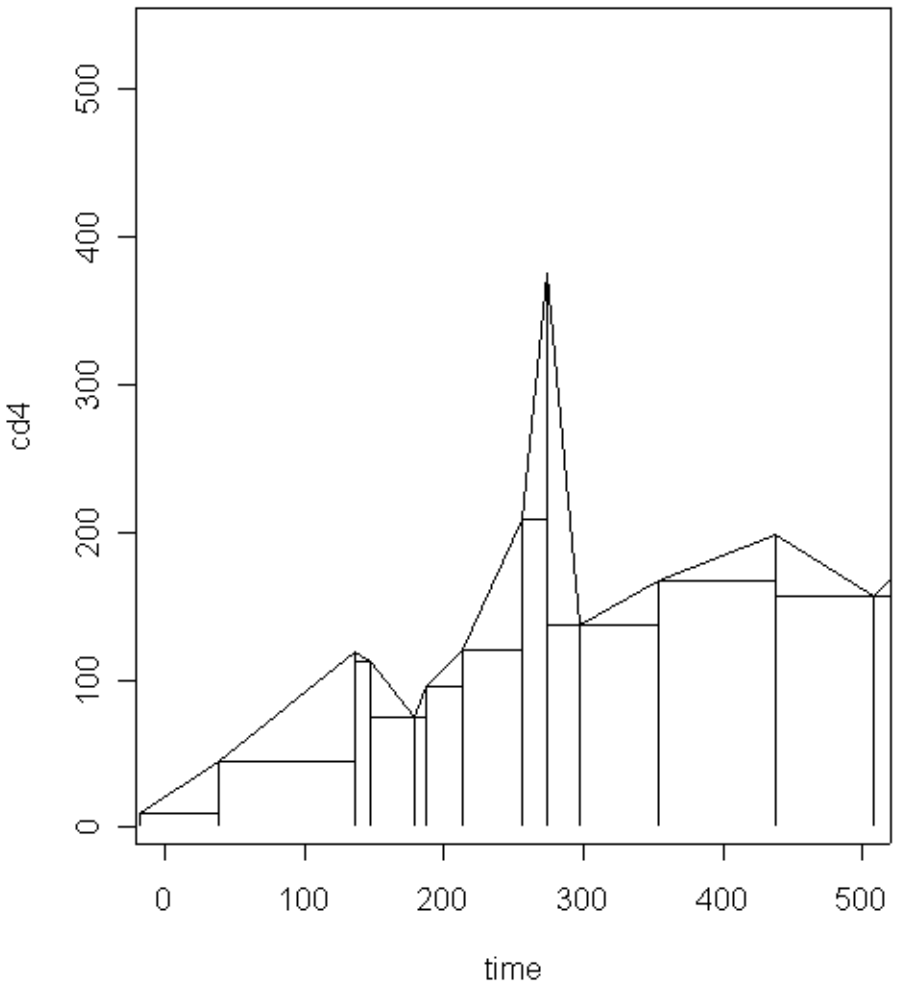
Plot of 20 evenly spaced (on rank of average CD4) subjects



Plotting subjects evenly spread over area under the curve (AUC)

- Same general algorithm as CD4 count, but now rank on AUC.
- Different ways to calculate AUC – below use simple trapezoid technique.
- Connect every point (time,cd4) with straight line and add up area underneath by simply adding triangle and rectangles

Showing polygons for calculating AUC



Syntax for plotting based on subjects ranked by AUC

- Add AUC from 0 to 1500 days for each subject. First, need to add points 0 and 1500 if subject does not already have them.

```
sort id etime
quietly by id: gen cntid = _n
quietly by id: gen totid = _N
expand 2 if cntid==totid | cntid==1
sort id etime
quietly by id: replace etime=1500 if _n==_N
quietly by id: replace cd4=. if _n==1
quietly by id: replace etime=0 if _n==1
quietly by id: replace cd4=. if _n==_N
```

- Linear interpolation to fill in CD4 at either 0 or 1500 days (or both).

```
id: ipolate cd4 etime, gen(newcd4)
```

Syntax for plotting based on subjects ranked by AUC

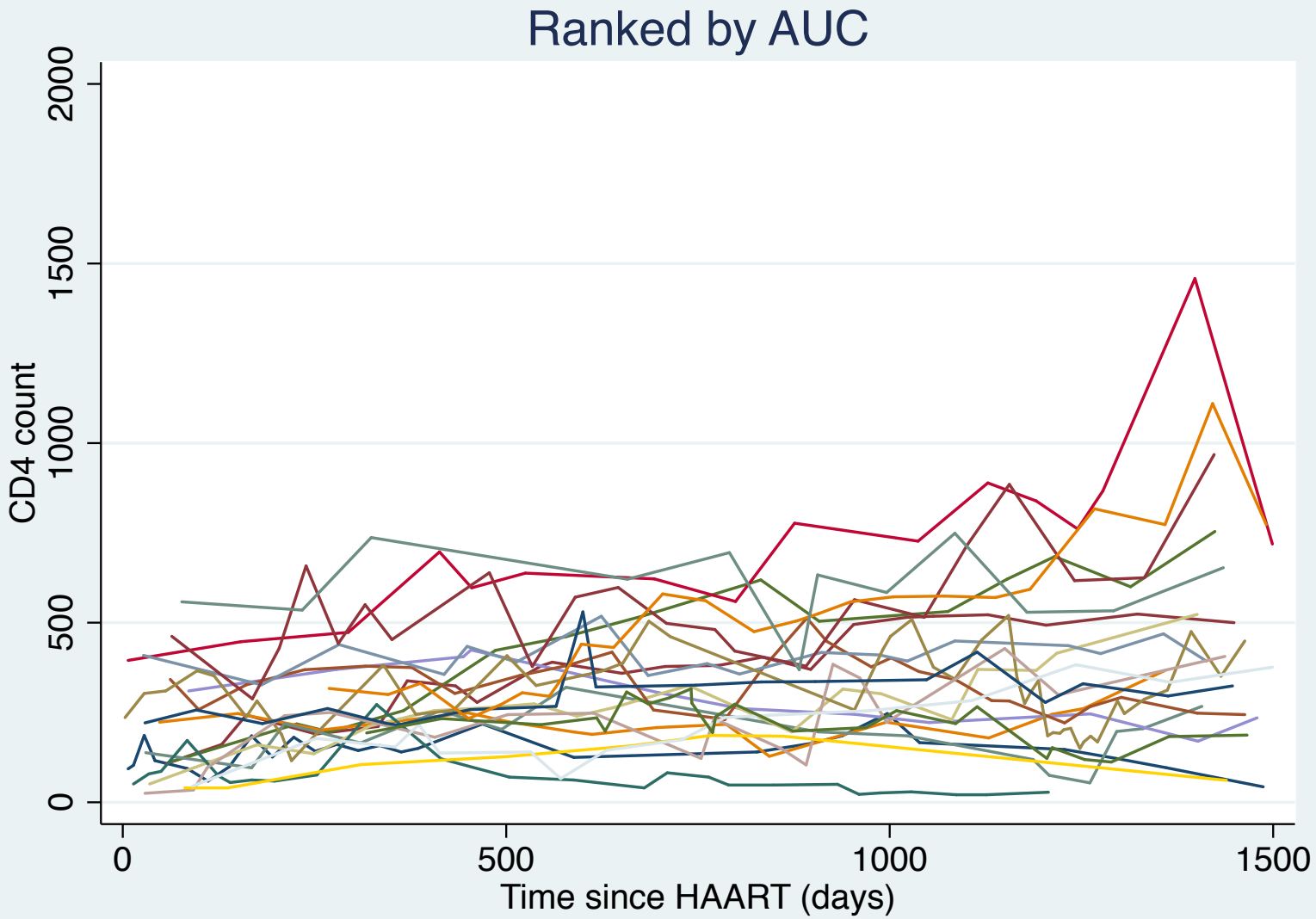
```
** Drop all points > 1500 days
drop if newcd4==. | etime > 1500
** Difference along x-axis
sort id etime
** Difference between successive points along
  x-axis
quietly by id: gen xdiff = etime[_n+1]-
  etime[_n]
** Difference between successive points along
  y-axis
quietly by id: gen ydiff = newcd4[_n+1]-
  newcd4[_n]
** Area of each rectangle+triangle
gen area = newcd4*xdiff+0.5*xdiff*ydiff
** Total AUC for subject
egen auc = sum(area), by(id)
```

Syntax for plotting based on subjects ranked by AUC

- Same algorithm to rank subjects by AUC and choose evenly spaced subjects (in terms of ranks) to get small subset to plot.

```
capture drop idcnt
sort id etime
quietly by id: gen idcnt = _n
capture drop rauc
egen rauc = rank(auc) if idcnt==1, unique
capture drop mr
egen mr = mean(rauc), by(id)
capture drop maxrmr
egen maxrmr = max(mr)
gen ii = int(maxrmr/20)
gen bb = mod(mr,ii)==0
```

Plot of 20 evenly spaced (on rank of AUC CD4) subjects



Fitting model by individual

- In addition to examining raw data one can also fit statistical models by subject.
- The type of fit will depend on many things including:
  - How much data per subject
  - Hypotheses about trends.
- In this case, we going to assume a simple linear model per subject.

Fitting linear models by individual

- In notation, for each subject *i* from *i=1,..,m* we use STATA to fit the linear models:

$$E\left[Y_{ij} \mid T_{ij}\right] = \beta_{0i} + \beta_{1i} T_{ij}$$

- Where Y_{ij} is the CD4 count on the *j*th measurement of subject *i* made at time T_{ij}.

- After the fit, we get estimates of the slopes and intercepts, $\hat{\beta}_{0i}, \hat{\beta}_{1i}$, for each subject, *i*.

Code to Fit Linear Models by ID

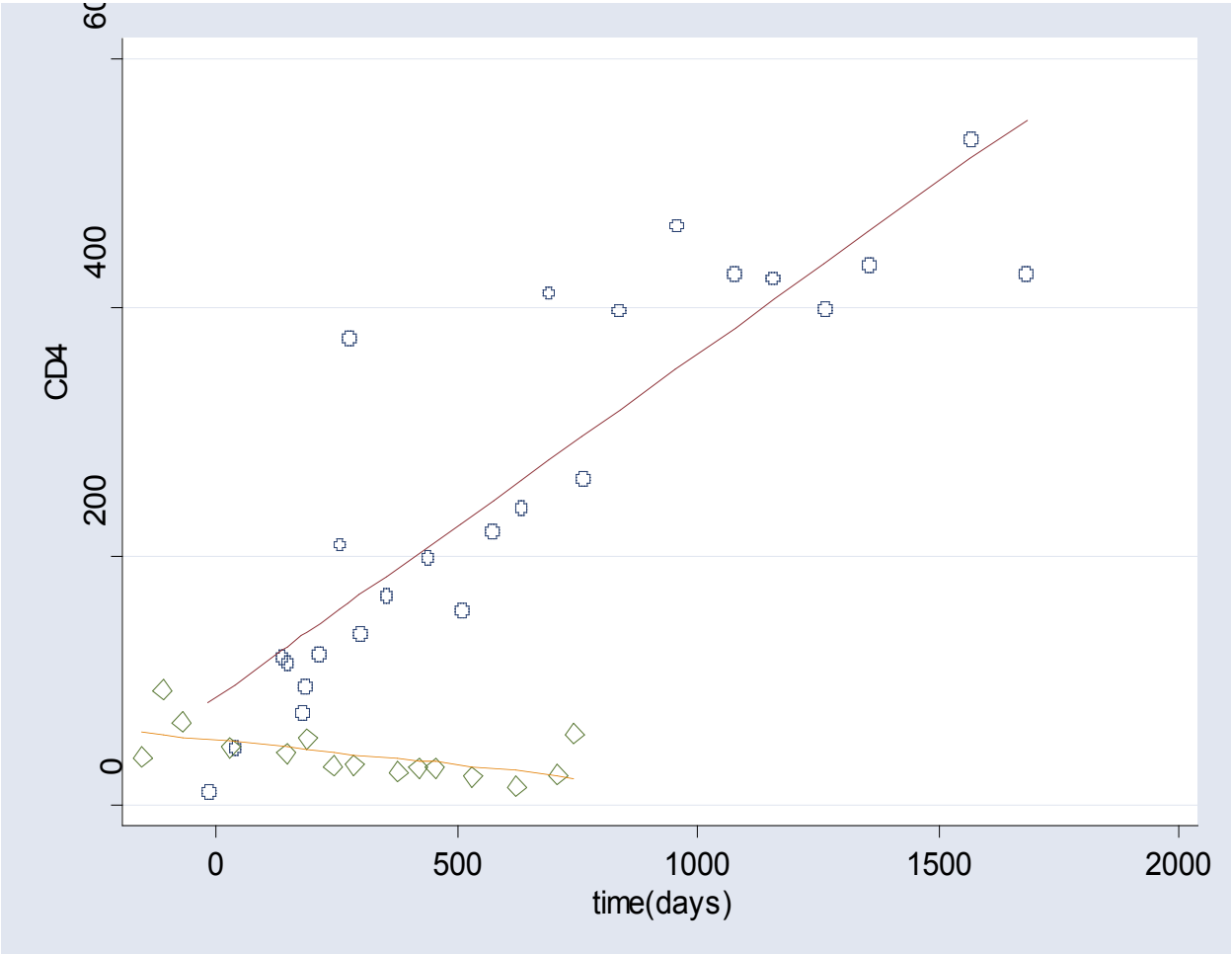
- Need to write a little STATA program:

```
* Blank variables to hold results
gen predcd4 = .
gen slope = .
gen intercept = .

* Define Program that does regression by ID and saves
* predicted values, slopes and intercepts
program define regbyid, byable(recall)
syntax [varlist] [if] [in]
marksample touse
capture matrix drop beta
regress `varlist' if `touse'
matrix beta = get(_b)
replace slope = beta[1,1] if `touse'
replace intercept = beta[1,2] if `touse'
capture drop predt
predict predt
replace predcd4 = predt if `touse'
end
```

Code to Fit Linear Models by ID

```
* Runs regression program by id
sort id
quietly by id: regbyid cd4 etime


*Plots example of fits and raw data for just 2
*ids
#delimit ;
scatter cd4 etime if id==1, ms(O) c(.) || scatter
    predcd4 etime if id==1,ms(i) c(l) ||
scatter cd4 etime if id==3, ms(D) c(.) || scatter
    predcd4 etime if id==3,ms(i) c(l)
legend(off) ytitle("CD4") xtitle("time(days)");
```

Example of fitting linear models by individual for two of the id's



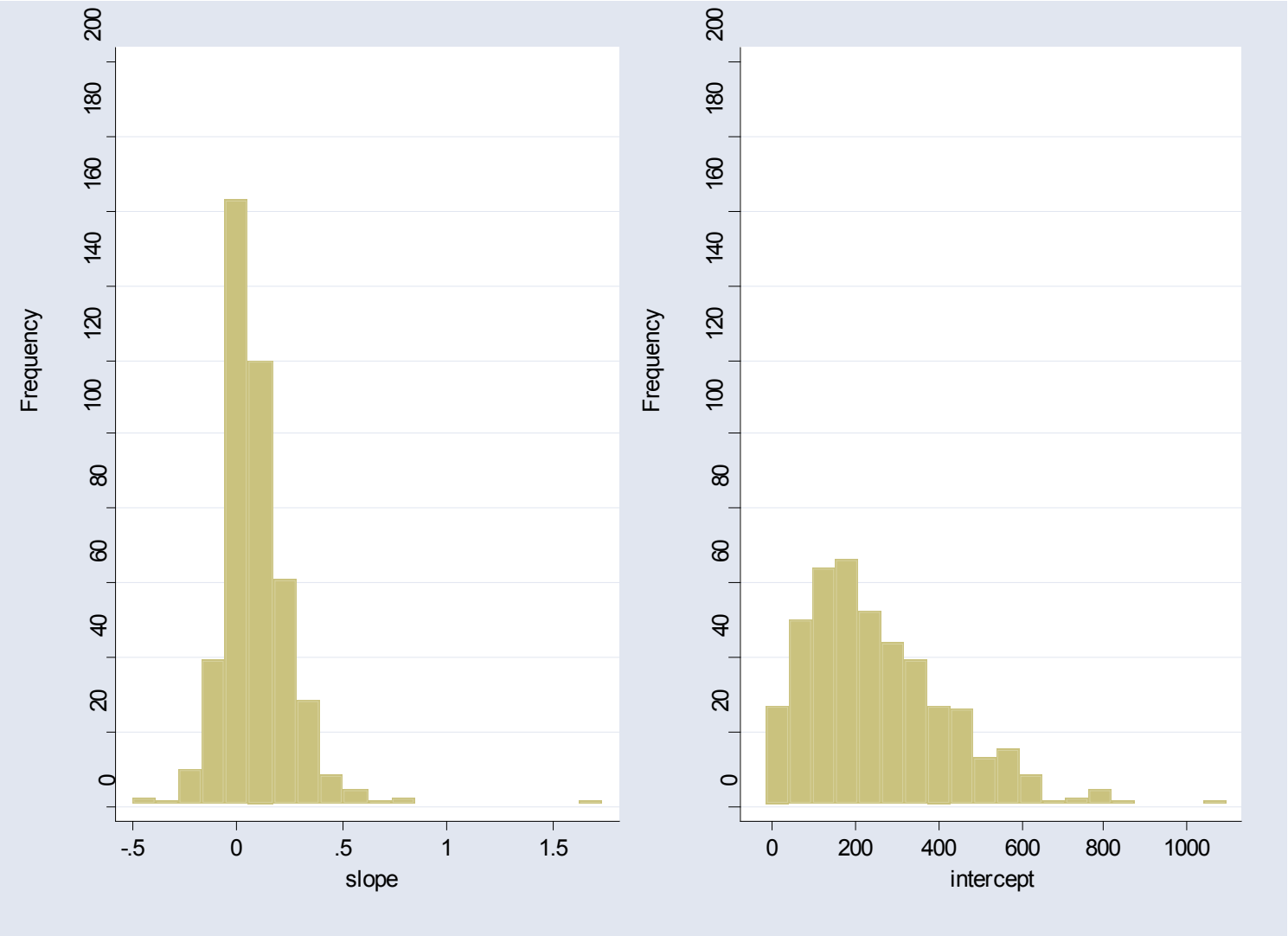
What is the distribution of Slopes and Intercepts?

- Want to see an estimate of the distribution of slopes and intercepts of CD4 count versus time.

- Can do so by using histograms.

```
** Create a variable that counts observations within an id
   sorted by time
capture drop idcnt
sort id etime
quietly by id: gen idcnt = _n
twoway histogram slope if idcnt==1, freq saving(graph1,
   replace) yscale(range(0 200))  ylabel(0(20)200, grid)

twoway histogram intercept if idcnt==1, freq saving(graph2,
   replace) yscale(range(0 200)) ylabel(0(20)200, grid)

gr combine graph1.gph graph2.gph
```

Histograms of Slope and Intercepts



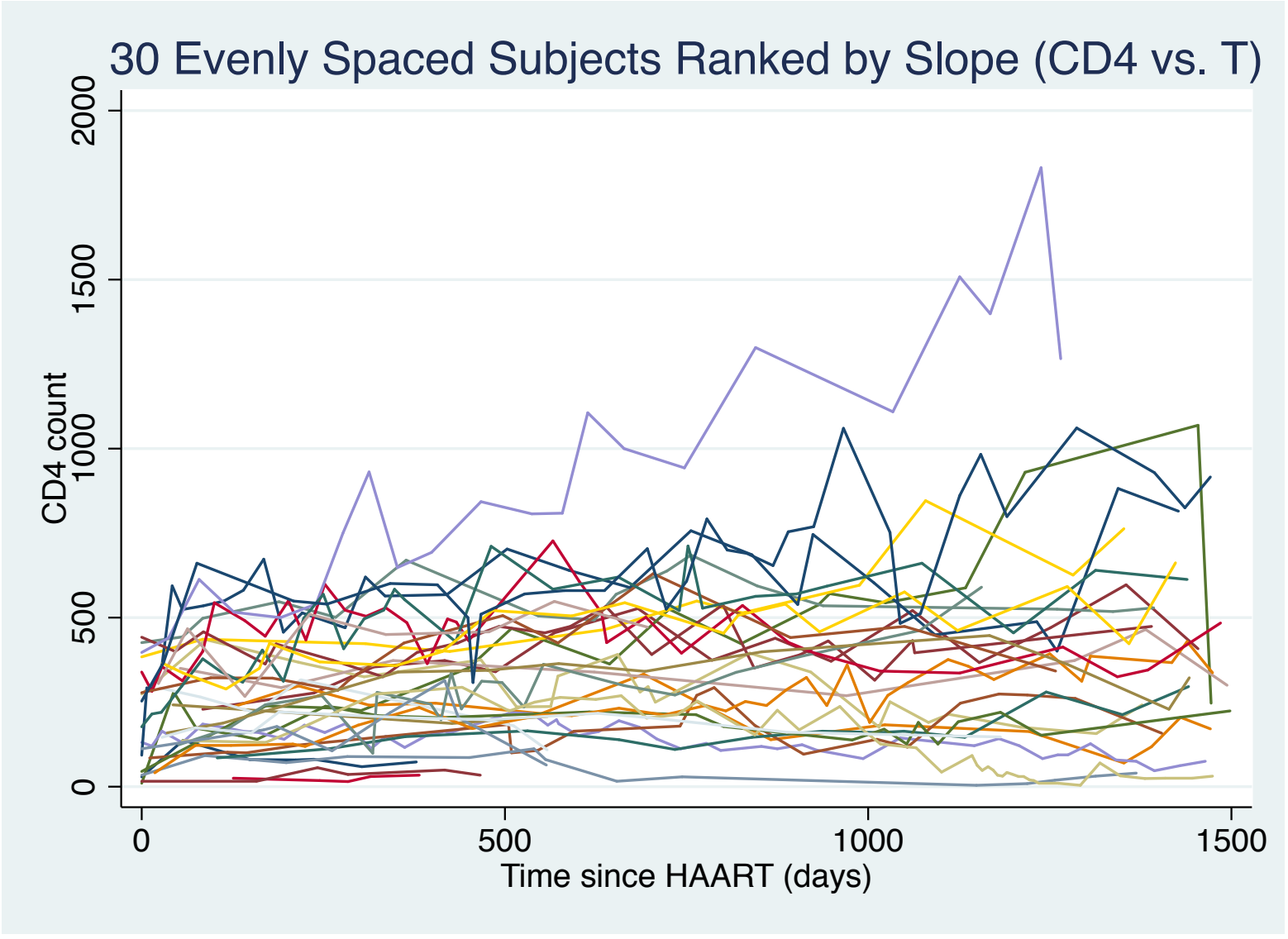
Display a subset of subjects based on slope (using both raw data and fit)

- Same principal as already examined for mean(CD4) and AUC.
- Sort subjects by slope (from most negative to most positive).
- Plot either the raw data or fitted lines for a subset of subjects evenly distributed with respect to the estimated slope.

Plot Raw Data, selected by Slope

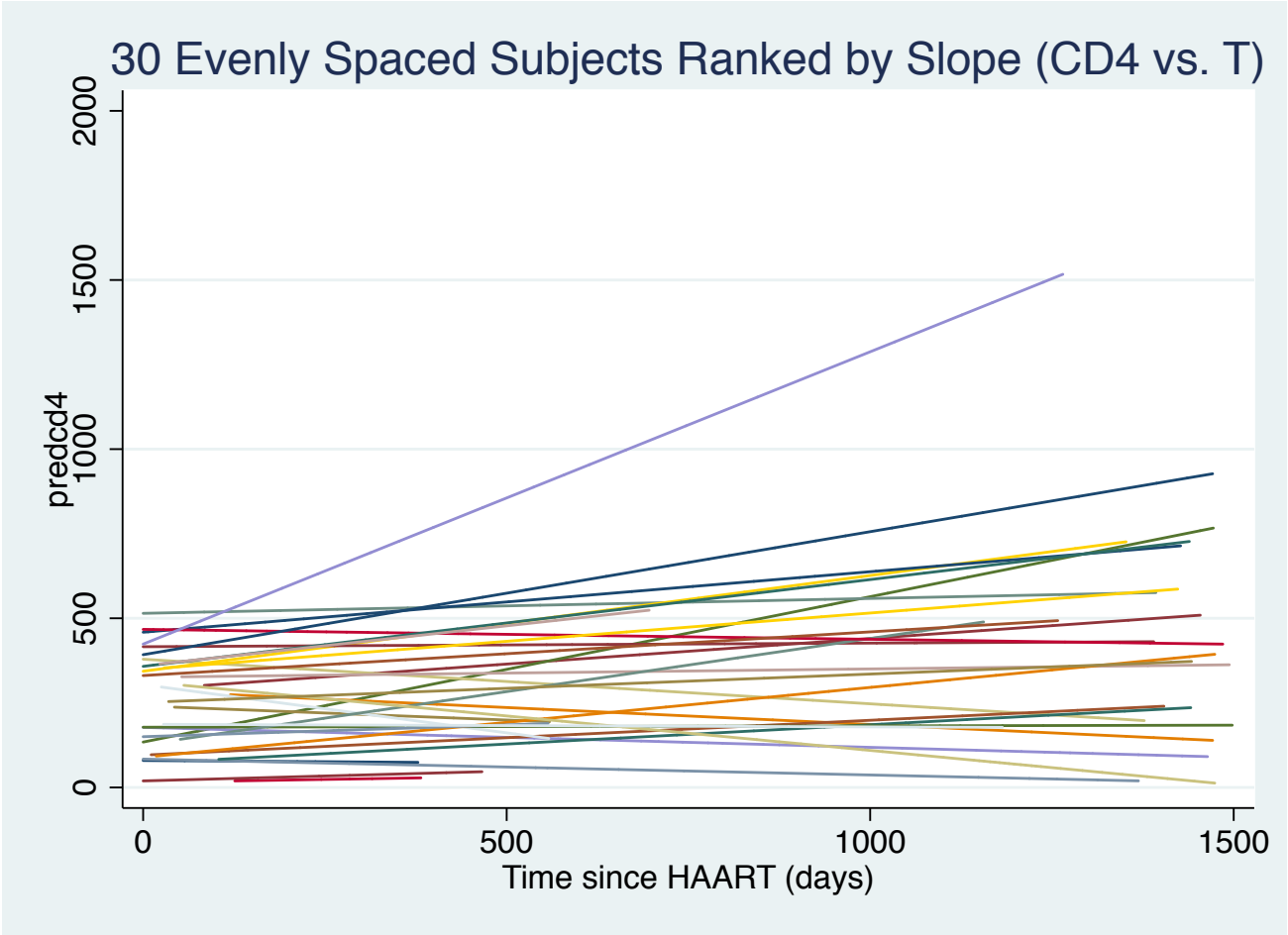
```
** Plot raw data evenly distributed w.r.t. slopes
capture drop rslope
egen rslope = rank(slope) if idcnt==1, unique
capture drop mr
** Trick just to get the rank at id level on every
  observation for that id
egen mr = mean(rslope), by(id)
capture drop maxrmr
** Maximum rank (or total number of subjects)
egen maxrmr = max(mr)
capture drop ii
* m/k
gen ii = int(maxrmr/30)
capture drop bb
gen bb = mod(mr,ii)==0
sort id etime
** Plot Results
#delimit ;
xtline cd4 if bb==1, i(id) overlay t(etime)
xlab(0(500)1500) ylab(0(500)2000) ti("30 Evenly Spaced
  Subjects Ranked by Slope (CD4 vs. T)") legend(off);
```

30 subjects evenly distributed with respect to slope of CD4 vs. time



Fitted values for same subjects

```
#delimit ;
xtline predcd4 if bb==1, i(id) overlay t(etime)
xlab(0(500)1500) ylab(0(500)2000) ti("30 Evenly
  Spaced Subjects Ranked by Slope (CD4 vs. T)")
  legend(off) saving(graph1, replace);
```



Smoothing

- One way to examine mean trends over time is to assume no particular form (linear, quadratic, etc.), but "smooth" the data (perform smooth regression).

- Many types of smoothing (moving average, kernel density, smoothing splines, local linear, etc.).

- Let the value of the smooth at time *t* be *m(t)* - almost all of the techniques to estimate *m(t)* can be understood as local (weighted) averages of the *Y*'s in a neighbor around an *t*.

Smooth regression

- Ignore the repeated measures structure of the data for now.
- Assume we have *Y*'s and *T*'s (CD4 and time).
- The way smooth regression works is to:
  - make a small grid of points *t* along the time axis (in our case, say *t=0,2,4,...,2000*).
  - for each t, estimate $E(Y|T=t)$.

Kernel smoothing regression

- Estimate is weighted average in the neighborhood:

$$\hat{m}(t) = \frac{\sum_{i=1}^{n} Y_i * w[(T_i - t)/h]}{\sum_{i=1}^{n} w[(T_i - t)/h]}$$

where $w[T_i\text{-}t]/h]$ is a weight which gets smaller as the distance between T_i and t gets larger (and how quickly it gets smal depends on h called the bandwidth)

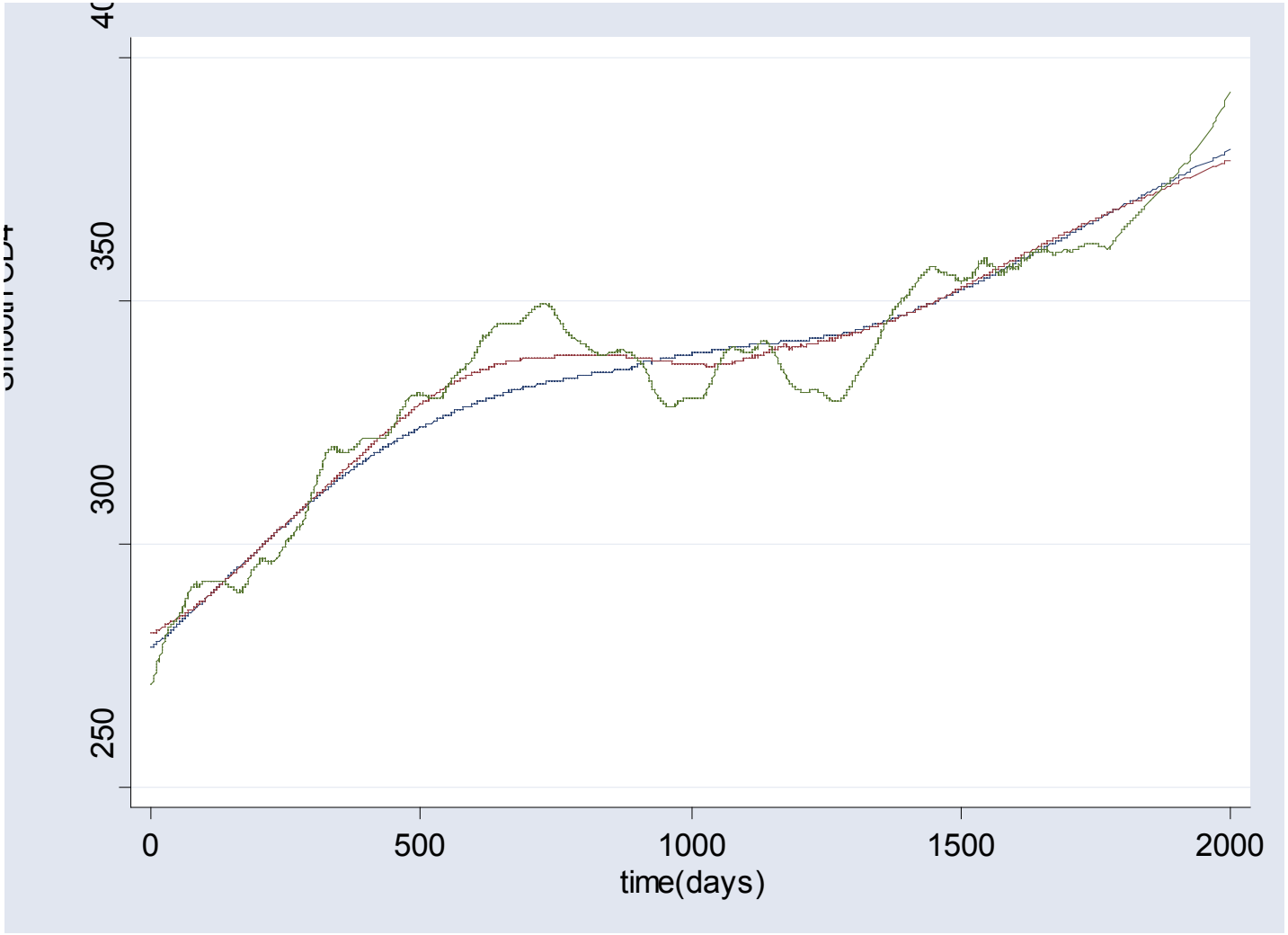
Kernel Smoothing, cont.

- Making a scatter plot (connecting points) of $\hat{m}(t)$ vs t.
- The bigger h is, the smoother the regression:
  - if h is really big, then at every point t you just get the average Y (a straight line).
  - if h is really small, just connect the raw data (no smoothing).
- There are ways to select the "best" h using model selection techniques (such as cross-validation).

Smoothing all data (CD4 vs. Time) using STATA

```
** At bandwidth that is 80% of data (gen generates a new
    variable which is m(t) at each t, or etime)
lowess cd4 etime if etime > 0 & etime < 2000,
    gen(smooth1) nograph bwidth(0.8)
** At bandwidth that is 50% of data
lowess cd4 etime if etime > 0 & etime < 2000,
    gen(smooth2) nograph bwidth(0.5)
** At bandwidth that is 10% of data
lowess cd4 etime if etime > 0 & etime < 2000,
    gen(smooth3) nograph bwidth(0.1)
sort etime
** Plot all together
#delimit;
scatter smooth1 etime if etime > 0 & etime < 2000, ms(i)
    c(l) || scatter smooth2 etime if etime > 0 & etime <
    2000, ms(i) c(l) ||
scatter smooth3 etime if etime > 0 & etime < 2000, ms(i)
    c(l) legend(off) ytitle("Smooth CD4")
    xtitle("time(days)") ysc(r(250 400));
```

Smoothed CD4 vs. Time for different bandwidths using STATA



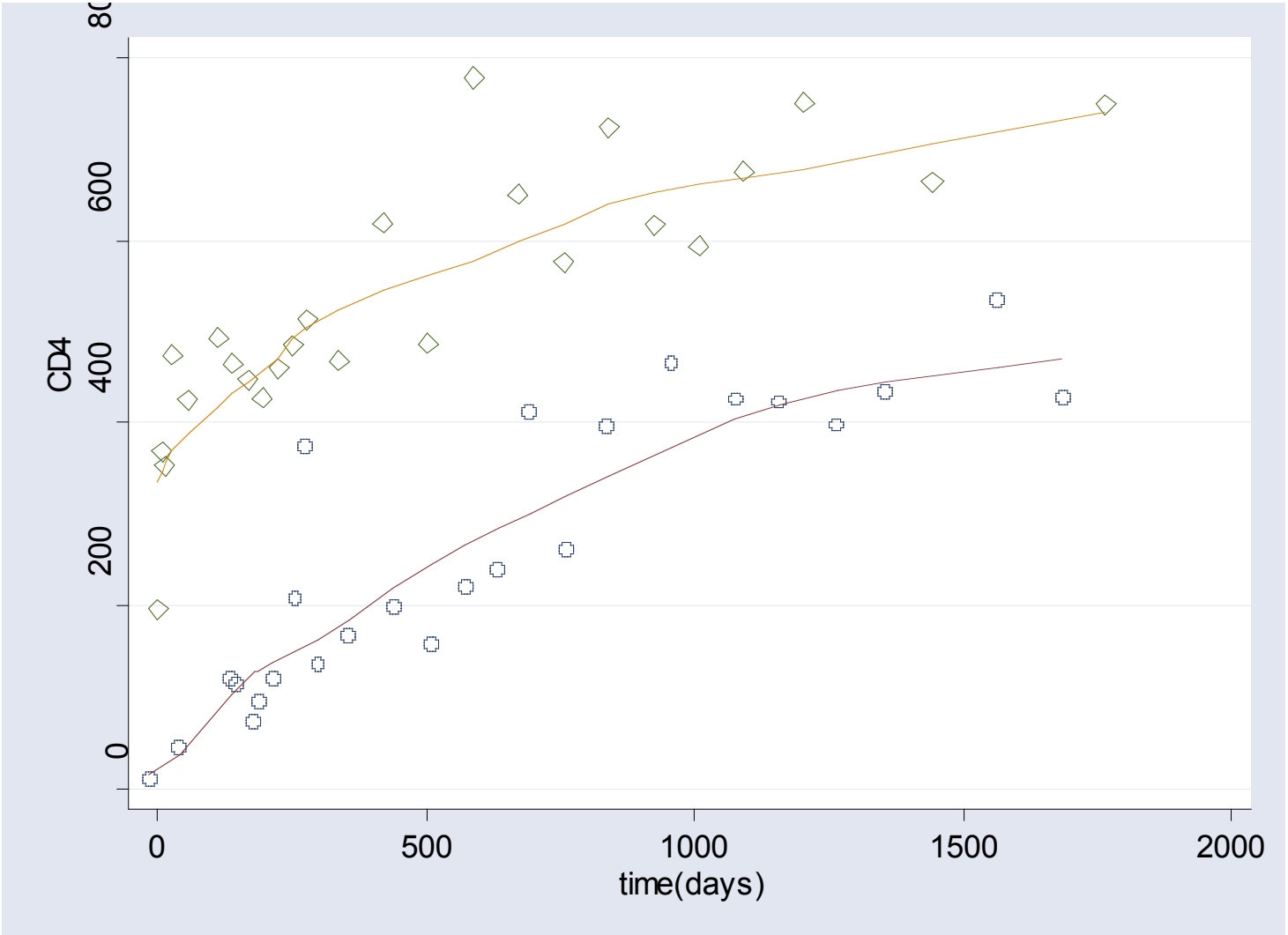
Smoothing one subject at a time

- Can make a similar plot to one showing the fitted linear models by id using the “smooths” instead.

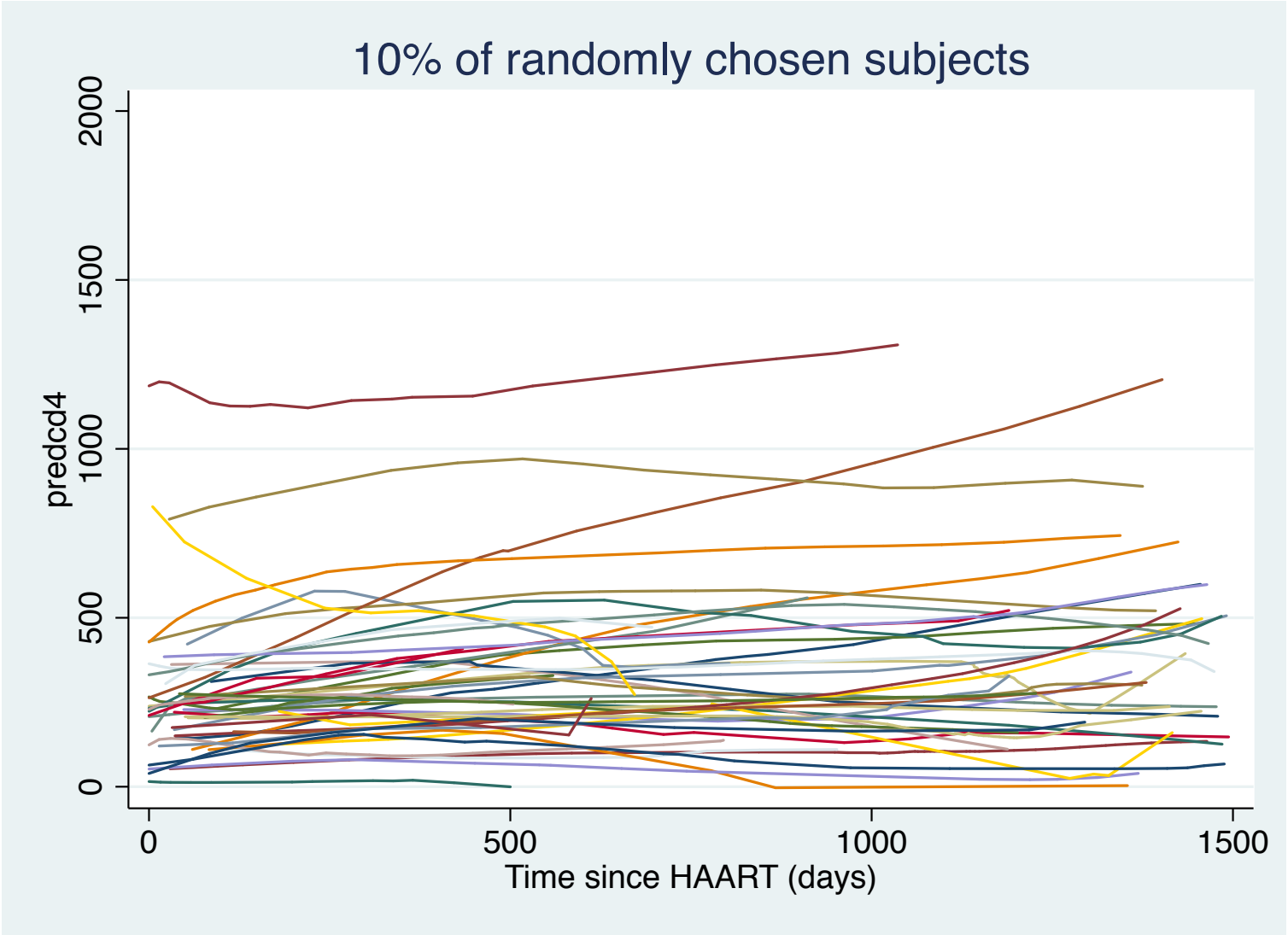
- Like the linear model example, need to right a little STATA program first.

```
***** Program to estimate smooth by id
capture drop predcd4
gen predcd4 = .
capture program drop smthbyid
program define smthbyid, byable(recall)
syntax [varlist] [if] [in]
marksample touse
capture drop predt
lowess `varlist' if `touse', gen(predt) bandwidth(0.5)
    nograph
replace predcd4 = predt if `touse'
end
```

Plot an example of two subjects of the smooth CD4 vs. time



Smooths plotted for a random Sample of 10% of the subjects



Clustering subjects based on similarity of CD4 versus time

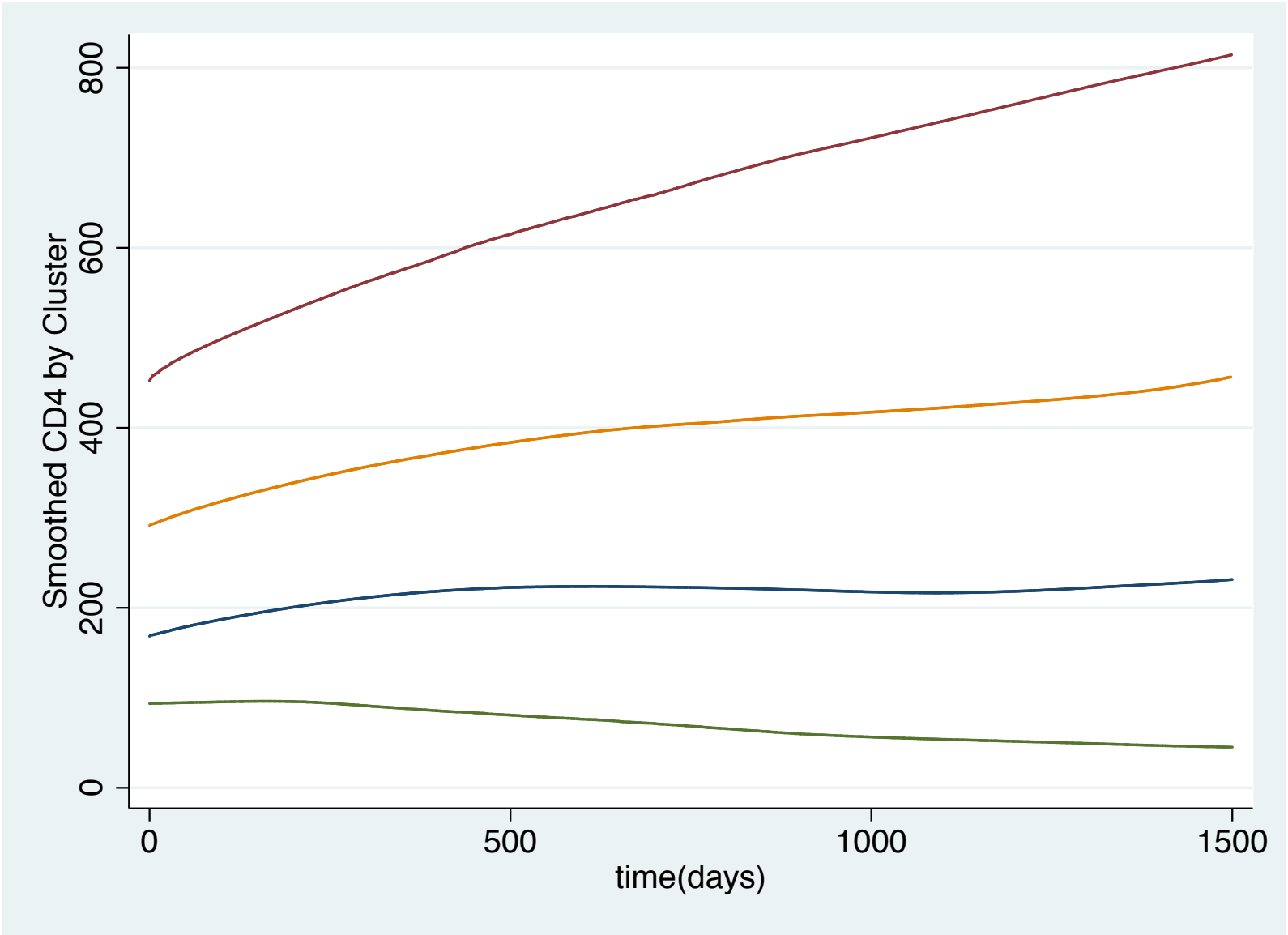
- Treat the CD4 observations at different times as simply different variables one measures on a subject.

- Technique requires all subjects to have the measurements at same times, so we need to interpolate at a fixed number of times for each subject so that everyone has identical times.

Clustering subjects based on similarity of CD4 versus time, cont

- Then, after putting the data in wide format, one simply clusters subjects based on these interpolated times.
- Then, after going back to long format, plot smooths by cluster assignment (see chapter 2 dofile for details).

Clustering subjects based on similarity of CD4 versus time



Outcomes vs. Explanatory Variables

- Besides time, we might also be interested in the relationship of an outcome, Y_{ij} to an explanatory variable, X_{ij}.
- The situation is relatively straightforward if examining variation in the trends in by differences in baseline covariates.
- Consider examining the trends in time CD4 for groups defined by baseline viral load.

CD4 vs. time by baseline Viral Load

- Perform smooth regression by Baseline Viral Load

```
**** Plotting CD4 vs. time for strata defined by viral load
** Make categorical variable for baseline viral load
gen catvl = vl500
recode catvl min/70000=0 70001/220000=1 220001/max=2
label define catvl 0 "<=70000" 1 "70001-220000" 2 ">220000"
label values catvl catvl
label variable catvl "Viral Load"
*** Replace catvl with dummy variable if not time 0
    (baseline)
replace catvl = -1 if etime !=0
*** Trick to assign baseline viral load category for an id to
    all observations
capture drop scatvl
egen scatvl = max(catvl), by(id)
```

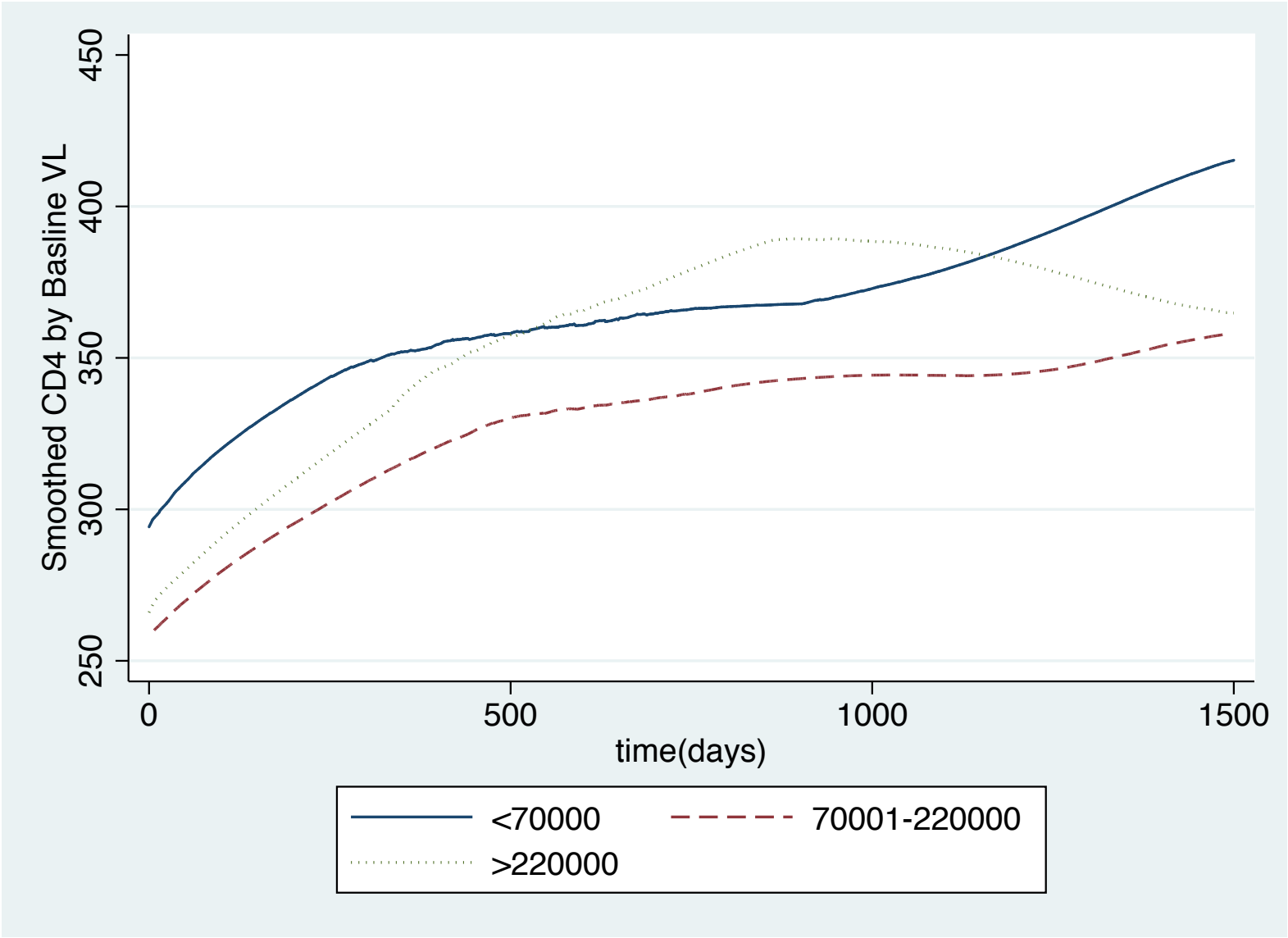
CD4 vs. time by baseline Viral Load, cont.

```
** Same program to smooth by baseline viral load
gen predcd4 = .
capture program drop smthbyid
program define smthbyid, byable(recall)
syntax [varlist] [if] [in]
marksample touse
capture drop predt
lowess `varlist' if `touse', gen(predt) bandwidth(0.5)
    nograph
replace predcd4 = predt if `touse'
end

** Do smooths
sort scatvl etime
quietly by scatvl: smthbyid cd4 etime
capture drop cntvl
quietly by scatvl etime: gen cntvl = _n
```

CD4 vs. time by baseline Viral Load, cont.

```
** Plot
gen vlow = predcd4
gen vmed = predcd4
gen vhigh = predcd4
replace vlow = . if scatvl !=0
replace vmed = . if scatvl !=1
replace vhigh = . if scatvl !=2
label variable vlow "<70000"
label variable vmed "70001-220000"
label variable vhigh ">220000"

** Plot
#delimit ;
scatter vlow etime if cntvl==1, ms(i) c(l)
    clpattern(solid) || scatter vmed etime if cntvl==1,
ms(i) c(l) clpattern(dash) || scatter vhigh etime if
    cntvl==1, ms(i) c(l) clpattern(dot)
ytitle("Smoothed CD4 by Basline VL") xtitle("time(days)");
```

Smooth regression CD4 vs. time by baseline Viral Load



Outcomes vs. Explanatory Variable Graphs

Looking at longitudinal effects: Change in Y vs. Change in X (change in CD4 vs. change in viral load)

- In this case, define a new outcome variable, say Y^*_{ij}, that is the change in CD4 count from last measurement (time *j-1*):

$$Y_{ij}^{*} = Y_{ij} - Y_{i(j-1)}$$

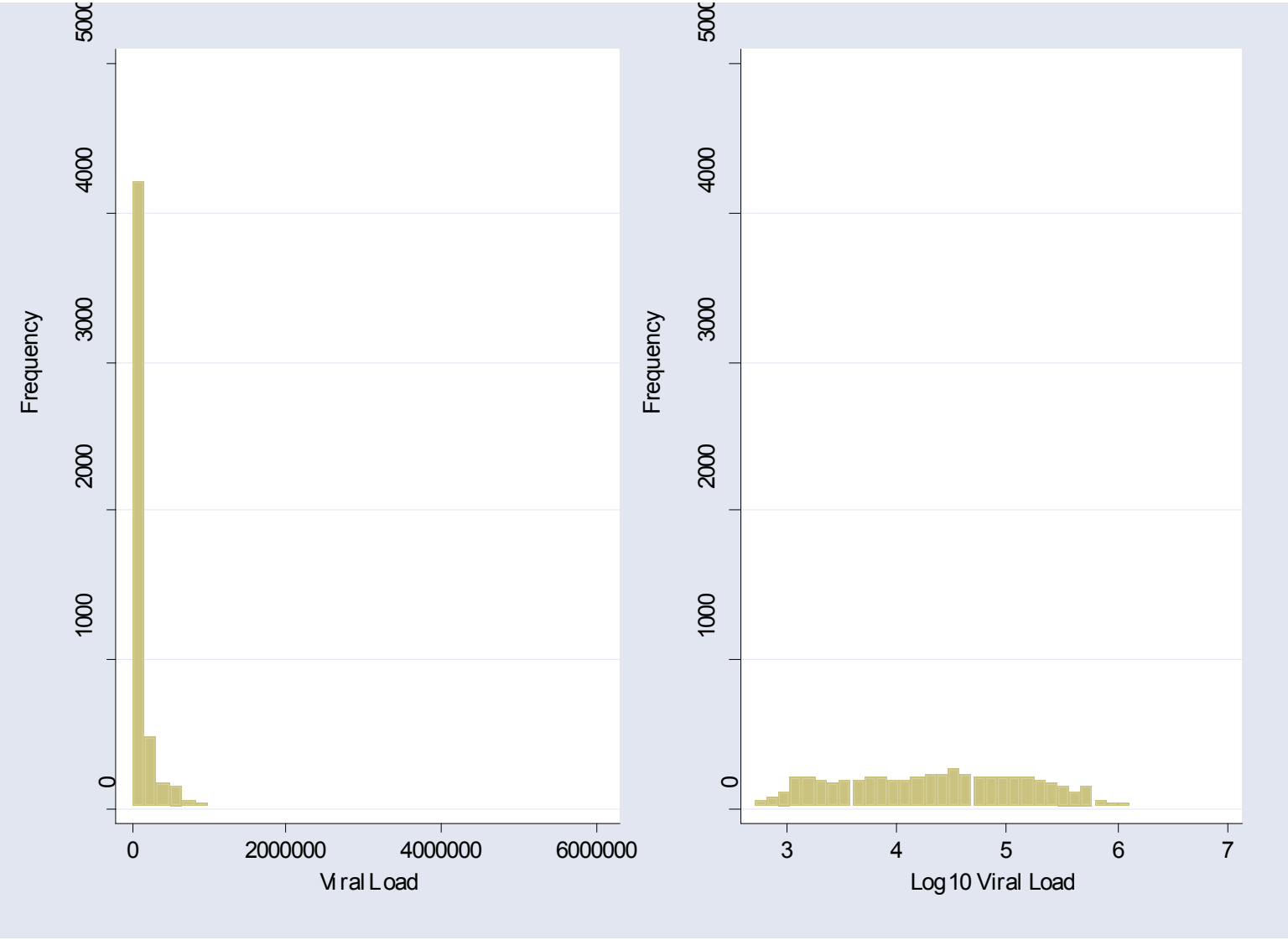
- Likewise, define a new explanatory variable that is the change in $\log_{10}$(viral load):

$$X_{ij}^{*} = X_{ij} - X_{i(j-1)}$$

Change in Y vs. Change in X (change in CD4 vs. change in viral load)

- Use graphical techniques already discussed for CD4 versus time to this new relationship, Y^*_{ij} vs X^*_{ij}

Distribution of Viral Load and $\log_{10}$(viral load) – eliminating non-detects (VL <500)



Change in CD4 vs. Change in log10(viral load)

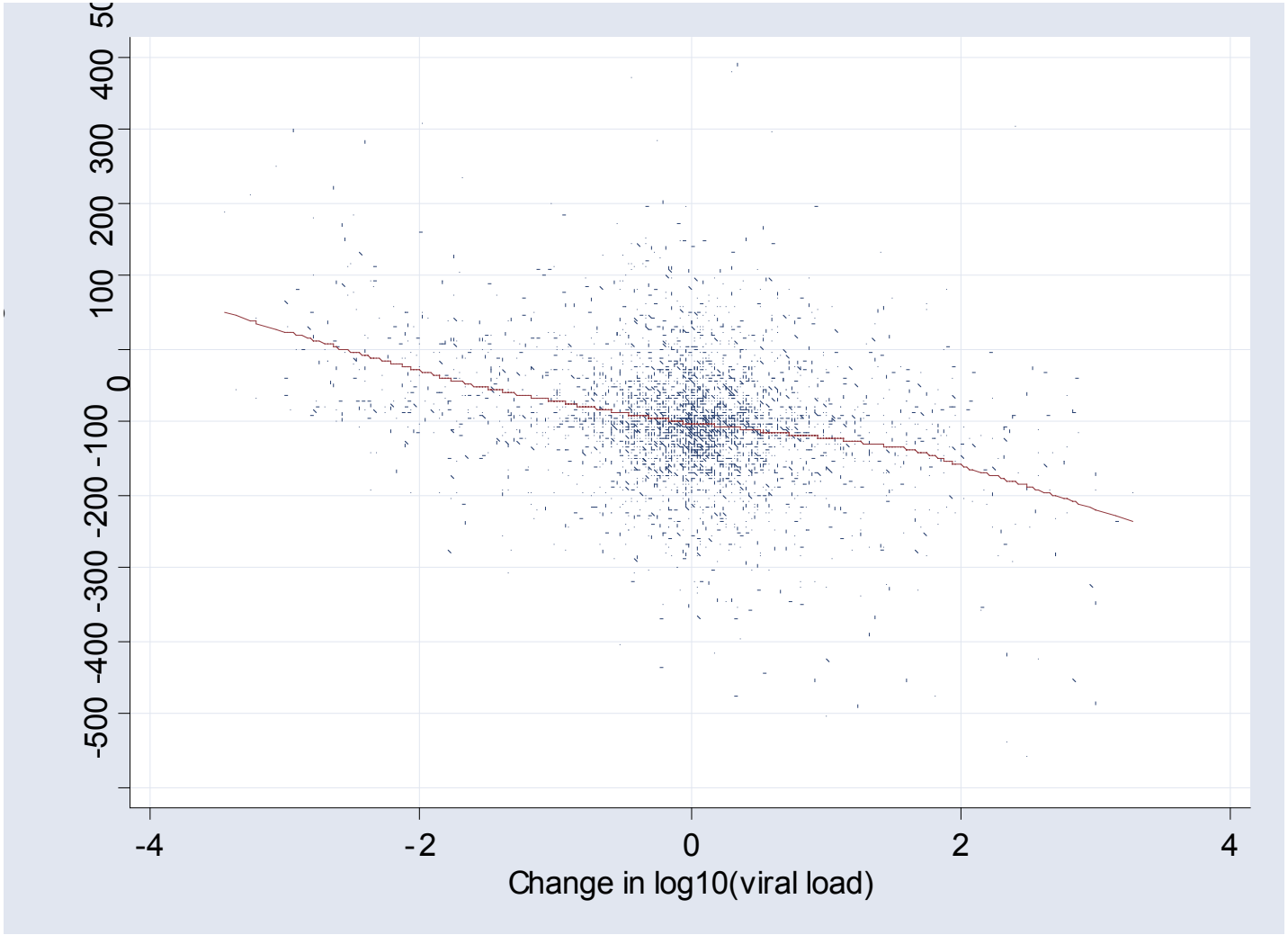
```
** Create X*ij, Y*ij
sort id etime
capture drop xdiff
quietly by id: gen xdiff = logvl[_n]-logvl[_n-1]
capture drop ydiff
quietly by id: gen ydiff = cd4[_n]-cd4[_n-1]
** If Viral load is <=500 twice in a row, make
** observation blank
by id: replace ydiff = . if vl500[_n-1]<=500 & xdiff==0
** Smooth Y*ij vs. X*ij
capture drop smthcd4vl
lowess ydiff xdiff, nograph gen(smthcd4vl)
** Only want to plot smooth at unique X*ij
sort xdiff
capture drop cntx
quietly by xdiff: gen cntx = _n
```

Change in CD4 vs. Change in $\log_{10}$(viral load)

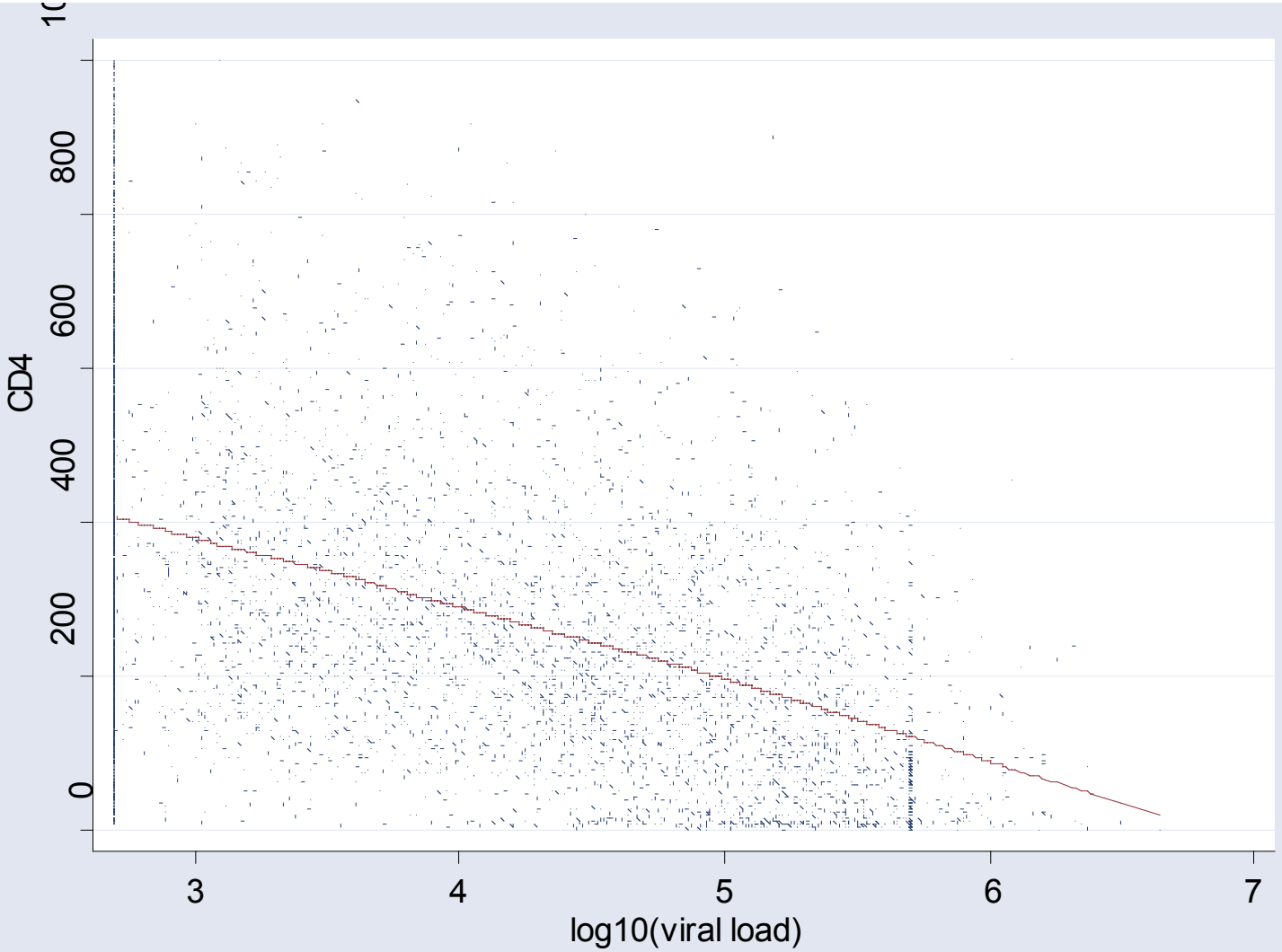
**** Plot both Smooth and Raw Data**

```
#delimit;
scatter ydiff xdiff if ydiff <500 & ydiff >-500, ms(p)
    c(.) ||  scatter smthcd4vl xdiff if cntx==1, ms(i)
    c(l)  legend(off)
ytitle("Change in CD4") xtitle("Change in log10(viral
    load)") xlabel(-4(2)4, grid);
```

Change in CD4 vs. Change in log10(viral load)



Just plain CD4 vs. log10(viral load)
x-sectional analysis of viral load data



Summary

- Raw Data Plots of Y_{ij} vs. T_{ij}
  - Random sample of subjects
  - Evenly distributed with respect to mean, AUC, median...
- Parametric (regression) models of Y_{ij} vs. T_{ij}
  - Histograms of the distributions of coefficients (slopes, intercepts,...)
  - Raw data plotted for subjects evenly distributed with respect to coefficients
  - Sample of subjects fitted data, i.e.,

$$\hat{E}[Y_{ij} \mid T_{ij}], \quad j = 1,..,n_i$$

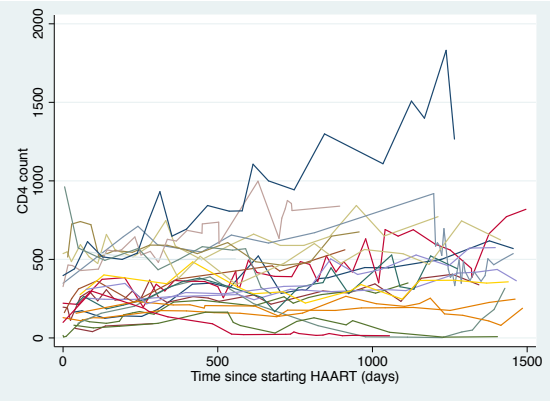
for a subset of subjects, i.

Summary

- Plots of Y_{ij} vs. T_{ij} for subjects stratified by baseline explanatory variable(s).
- Change in Y_{ij} vs. change in X_{ij} .
- Many more possible (e.g., surface plots of Y_{ij} vs. X_{ij}, T_{ij}).

Supplement

Chapter 2 in Weiss



Summaries Across Individuals

$$\bar{Y}_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n} Y_{ij}$$

and the sample standard deviations

$$s_{jj} = \left[\frac{1}{n-1}\sum_{i=1}^{n}(Y_{ij} - \bar{Y}_{\cdot j})^2\right]^{1/2}$$

of the Y_{ij} at a specific time j. We want to know if these means and standard deviations are increasing, constant, or decreasing over time. The ratio

$$\gamma_{ij} = \frac{Y_{ij} - Y_{i(j-1)}}{t_{ij} - t_{i(j-1)}} \qquad (2.1)$$

Chapter 2 in Weiss (2005)

Questions Plots Could Address

- the population mean response at a particular time,
- the population variance or standard deviation of the responses at a particular time,
- the correlations between observations within subjects, and
- the effects of covariates on these quantities.

Chapter 2 in Weiss (2005)

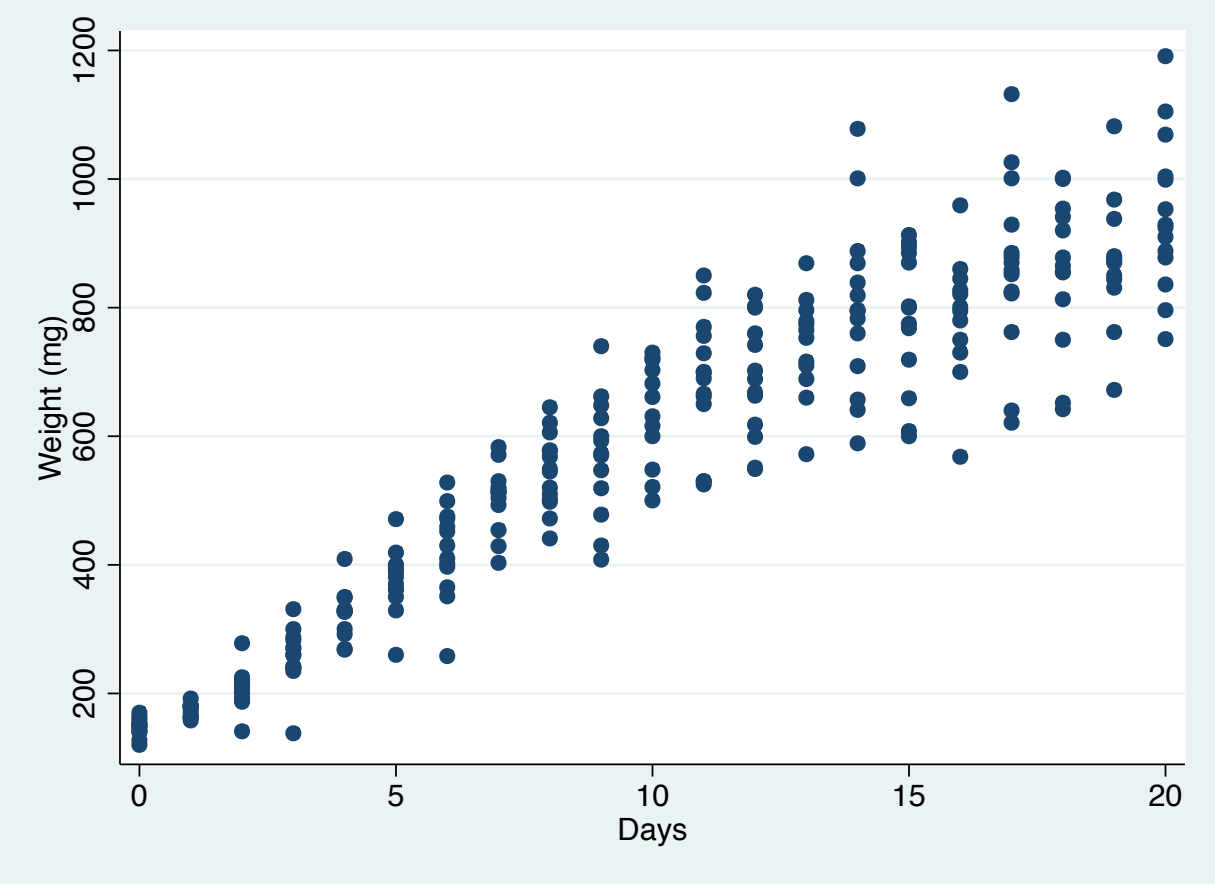
Big Mice Data

plots the longitudinal response against time. The obvious first plot we might consider plots all responses Y_{ij} against time t_{ij}. Figure 2.1 shows this plot for the *Big Mice* data. The response is the weight in milligrams for $n = 35$ mice with each mouse contributing observations from various days starting at birth, day 0, through day 20. Thirty-three of the mice were weighed every three days for a total of seven observations each. Eleven mice in group 1 were weighed beginning on day 0, ending on day 18; group 2 has 10 mice weighed beginning on day 1 ending on day 19; and group 3 has 12 mice weighed beginning on day 2 ending on day 20. The last two mice are in group 4 and were weighed daily from day 0 to day 20. A subset of the Big

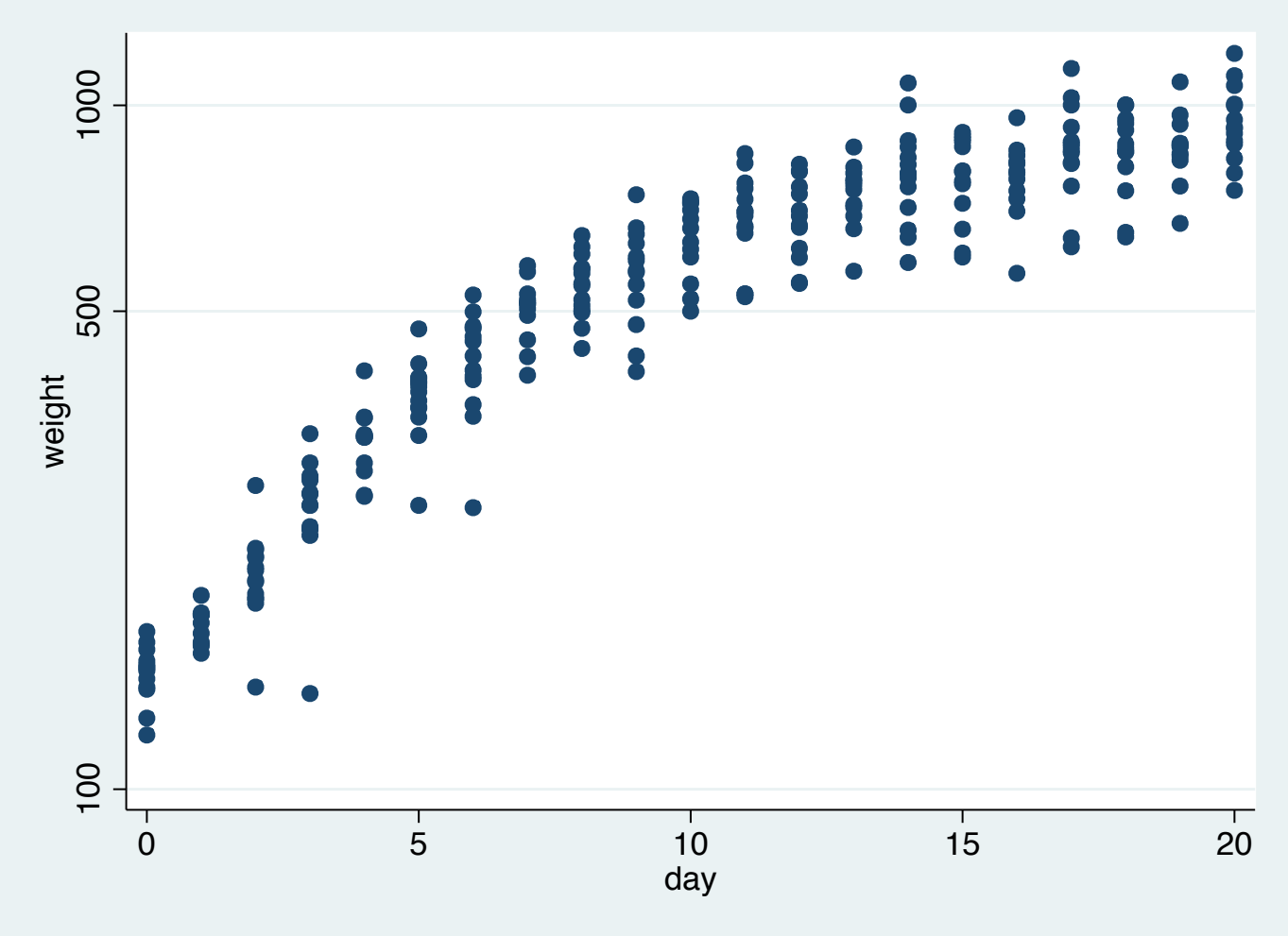
Get data from http://rem.ph.ucla.edu//mld/data.html

Scatter of Raw Data

```
* Read Data from Website
insheet using http://rem.ph.ucla.edu//mld/data/tabdelimiteddata/
    bigmice.txt
label variable weight "Weight (mg)"
label variable day "Days"
scatter weight day
```

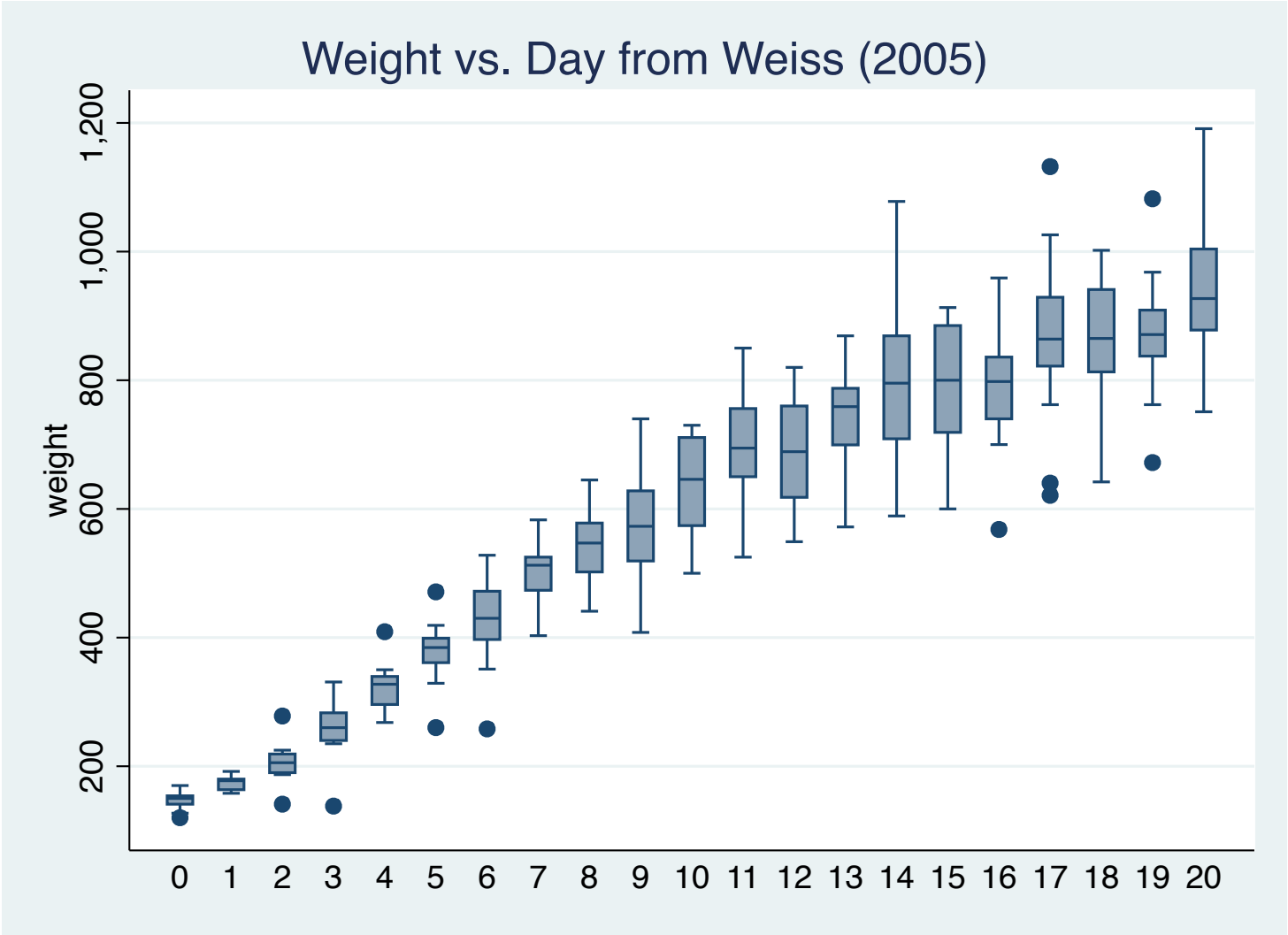


Variance Stabilizing Transformations



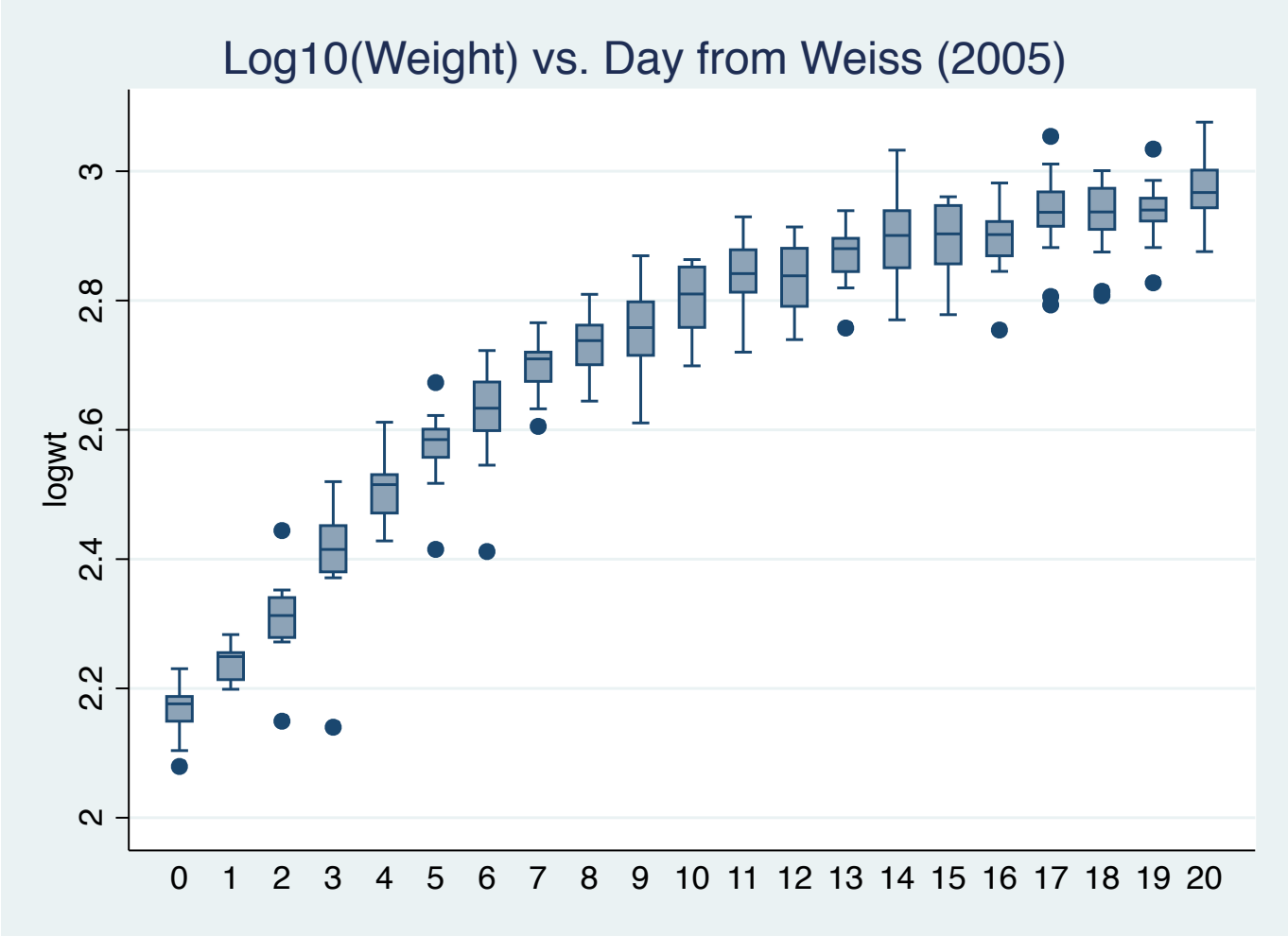

```
scatter weight day, yscale(log) ylabel(100 500 1000)
```

Distribution by Time




```
scatter weight day, yscale(log) ylabel(100 500 1000)
```

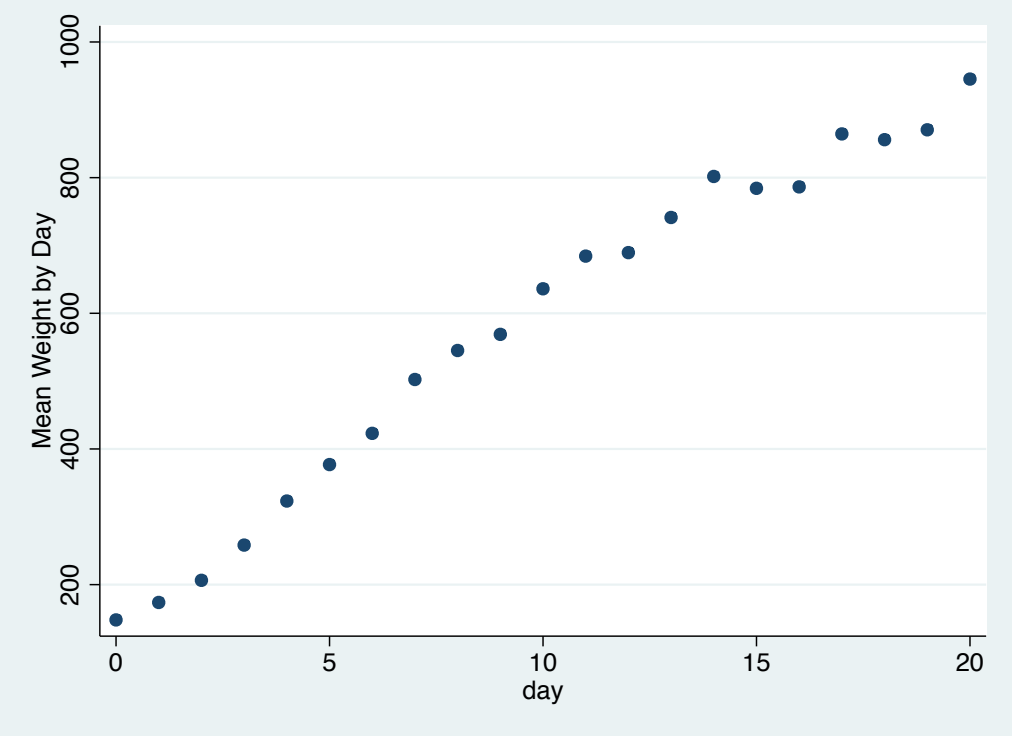
Distribution by Time ($\log_{10}$ wt)




```
gen logwt = log10(weight)
graph box logwt, over(day)  title("Log10(Weight) vs. Day from Weiss (2005)", span)
```

Mean by Time

```
capture drop meanbyday
egen meanbyday = mean(weight), by(day)
capture drop cntday
sort day
by day: gen cntday = _n
label variable meanbyday "Mean Weight by
      Day"
scatter meanbyday day if cntday==1
```

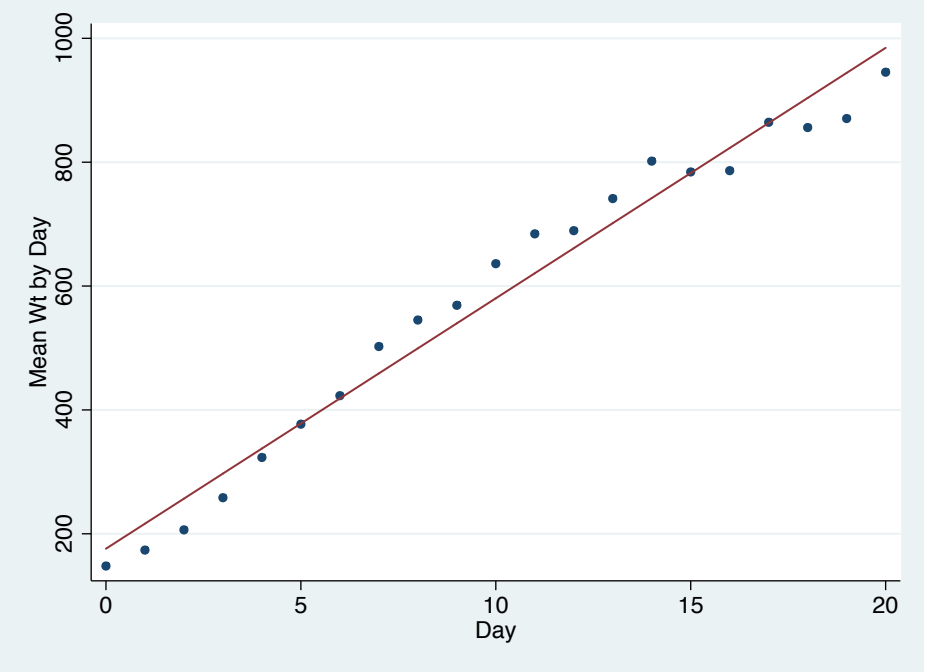


$$E\left[Y_{ij} \mid T_{ij}\right] = \beta_0 + \beta_1 T_{ij}$$

Inference For Regression of Wt by Day

- Pretend Days are Indep.

```
regress meanbyday day if
   cntday==1
capture drop py
predict py
sort day id
#delimit ;
scatter meanbyday day if
   cntday==1 , ms(o) c(.) ||
scatter py day if
   cntday==1,ms(i) c(l)
   legend(off) ytitle("Mean Wt
   by Day") xtitle("Day");
```



Inference (biased) For Regression of Wt by Day

```
. regress meanbyday day if cntday==1

      Source |       SS       df       MS              Number of obs =      21
-------------+------------------------------           F(  1,    19) =  675.22
       Model |  1259110.38     1  1259110.38           Prob > F      =  0.0000
    Residual |  35430.0041    19  1864.73706           R-squared     =  0.9726
-------------+------------------------------           Adj R-squared =  0.9712
       Total |  1294540.39    20  64727.0194           Root MSE      =  43.183

------------------------------------------------------------------------------
   meanbyday |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         day |   40.43771   1.556193     25.99   0.000     37.18056    43.69486
       _cons |    175.862    18.1926      9.67   0.000     137.7844    213.9395
------------------------------------------------------------------------------
```

Bootstrapping

- In a perfect statistical world, one could derive the inference on the estimate of by simply performing the same experiment (say random sample of id's with replacement).

- For instance, one would do this B times, and derive the inference, just by calculating the sample variance of the estimated:

$$\hat{\text{var}}(\hat{\beta}_1) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}_1^b - ave(\hat{\beta}_1^b)\right)^2$$

where $\hat{\beta}_1^b$ is the estimate made on the *b*th sample.

Inference using (Clustered) bootstrapping, cont.

- Bootstrapping mimics by sampling repeatedly (with replacement) not from the target population, but from the data itself.
- Works by re-sampling the independent statistical units (the rows) with replacement and creating new data sets of the same number of id's, *m*.
  - In this case, id's are randomly sampled with replacement, not observations.
- Then, the inference can be derived directly from the empirical distribution of these simulated estimates treating the data as the whole target pop^{n}.

Bootstrap SE

$$\text{boot } SE(\hat{\beta}_1) = \text{estimated SD} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}_1^b - \hat{\beta}_1\right)^2}$$

Correct Inference: Clustered Bootstrap

```
program bootwtreg, rclass
 quietly capture drop bmeanbyday
 quietly egen bmeanbyday = mean(weight), by(day)
 quietly capture drop bcntday
 quietly sort day
 quietly by day: gen bcntday = _n
 quietly regress bmeanbyday day if bcntday==1
 matrix coefs = get(_b)
 return scalar beta1 = coefs[1,1]
end

bootstrap beta1=r(beta1), cluster(id) rep(1000) nodrop: bootwtreg
                                    (Replications based on 35 clusters in id)
-------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+-----------------------------------------------------------------
       beta1 |   40.43771   1.341163    30.15   0.000     37.80908    43.06634
-------------------------------------------------------------------------------
```

Original Naïve Approach (from slide 12)

```
------------------------------------------------------------------------------
   meanbyday |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         day |   40.43771   1.556193    25.99   0.000     37.18056    43.69486
       _cons |    175.862    18.1926     9.67   0.000     137.7844    213.9395
------------------------------------------------------------------------------
```

Nonparametric Bootstrap Distribution of slope