August 2005
Stata Application Tutorial 3: Cox Model

Data Note: Code makes use of cabinet.dta and restrictiveadoption.dta. Both data sets are available on the Event History website. Code is based on Stata version 8.

Preliminaries: Basic issues with Cox model.
Let's estimate a model using cabinet duration data:

```
. stcox invest polar numst format postelec caretakr, efron nohr

         failure _d:  censor
   analysis time _t:  durat

Iteration 0:   log likelihood =  -1369.664
Iteration 1:   log likelihood = -1307.6939
Iteration 2:   log likelihood = -1287.7995
Iteration 3:   log likelihood = -1287.7389
Iteration 4:   log likelihood = -1287.7389
Refining estimates:
Iteration 0:   log likelihood = -1287.7389

Cox regression -- Efron method for ties

No. of subjects =          314                     Number of obs   =       314
No. of failures =          271
Time at risk    =       5789.5
                                                   LR chi2(6)      =    163.85
Log likelihood  =   -1287.7389                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      invest |   .3871388   .1371298     2.82   0.005     .1183693    .6559083
       polar |   .0233392   .0056193     4.15   0.000     .0123255    .0343528
       numst |  -.5826222   .1322266    -4.41   0.000    -.8417816   -.3234628
      format |    .130011   .0438699     2.96   0.003     .0440275    .2159945
    postelec |  -.8611202   .1406178    -6.12   0.000    -1.136726   -.5855144
    caretakr |   1.710397   .2828184     6.05   0.000     1.156084    2.264711
------------------------------------------------------------------------------
```

The coefficients above are unexponentiated (i.e. are hazard rates). As we've seen before, deriving the hazard ratio is simple: exponentiate the coefficient.

For the investiture covariate, the estimated hazard ratio is:

```
. display exp(_b[invest])
1.4727609
```

This implies the “risk” of a cabinet failure is about 1.47 (or about 1 ½ times greater when there is an investiture requirement than when there is no investiture requirement).

What about the post-election covariate. The hazard ratio is

```
. display exp(_b[postelec])
.42268833
```

This means that when a government is formed immediately after the election, the risk is about .42 that of the case when protracted negotiations are necessary to form the government. The risk is *lower*. Equivalently, taking the inverse

```
 . display 1/exp(_b[postelec])
2.3658093
```

gives us the risk for the case when postelec=0. Here, we see that when negotiations are required, the risk of failure is about 2.4 times greater than compared to the case when the government is immediately formed after the election (the two numbers convey different “sides to the same coin;” one is the inverse of the other).

Now, let me illustrate the proportional hazards property “in action.”

Consider the formation attempts covariate. It takes the values:

```
. tab format

  Formation |
   attempts |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |        179       57.01       57.01
          2 |         63       20.06       77.07
          3 |         36       11.46       88.54
          4 |         14        4.46       92.99
          5 |         12        3.82       96.82
          6 |          5        1.59       98.41
          7 |          1        0.32       98.73
          8 |          4        1.27      100.00
------------+-----------------------------------
      Total |        314      100.00
```

Now, let’s compute the hazard ratio for each level of the format covariate:

```
. gen format_ratio=exp(_b[format]*format)

. tab format_ratio

format_rati |
          o |      Freq.     Percent        Cum.
------------+-----------------------------------
   1.138841 |        179       57.01       57.01
   1.296959 |         63       20.06       77.07
    1.47703 |         36       11.46       88.54
   1.682102 |         14        4.46       92.99
   1.915646 |         12        3.82       96.82
   2.181616 |          5        1.59       98.41
   2.484514 |          1        0.32       98.73
   2.829466 |          4        1.27      100.00
------------+-----------------------------------
      Total |        314      100.00
```

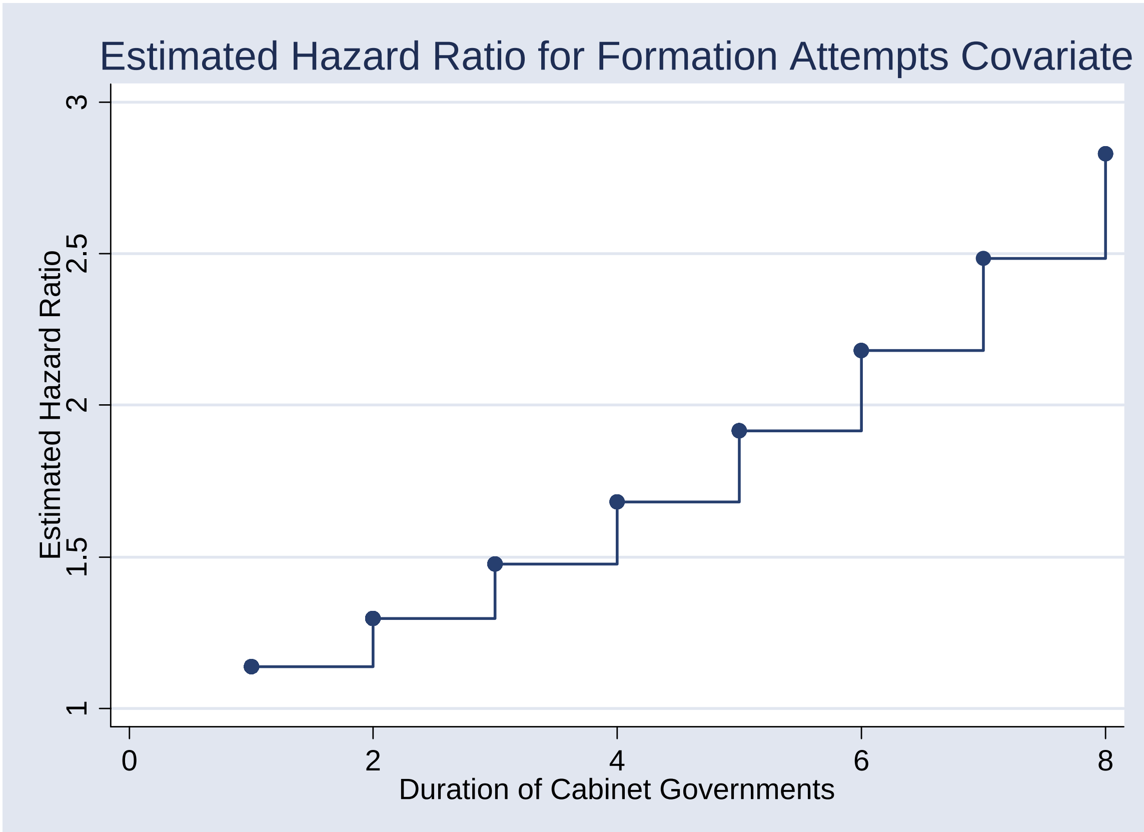
As in the case of the other PH models, the ratio of each adjacent value of the format covariate is *identical*:

```
. display 1.47703/1.296959
1.1388409

. display 1.682102/1.47703
1.1388408

. display 2.829466/2.484514
1.1388408
```

Graphing the hazard ratio for a covariate might be something we'd like to do:

```
. twoway (scatter format_ratio format, c(J)),
ytitle(Estimated Hazard Ratio) xtitle(Duration of Cabinet
Governments) title(Estimated Hazard Ratio for Formation
Attempts Covariate) saving(icpsr_hrcoxcab, replace)
```

This returns:



This shows how the estimated risk is increasing with increases in the covariate. This is sometimes a useful way to interpret your Cox model.

COX DIAGNOSTICS: The Proportional Hazards Property and Other Issues

The Link-Test

First, estimate a full Cox model.

```
. stcox south mooneymean asimean, efron nohr mgale(mg)
schoenfeld(sc*) scaledsch(ssc*) cluster(stcode)
```

```
         failure _d:  rev_even
   analysis time _t:  duration

Iteration 0:   log pseudolikelihood = -689.70194
Iteration 1:   log pseudolikelihood = -673.54338
Iteration 2:   log pseudolikelihood = -673.16918
Iteration 3:   log pseudolikelihood =  -673.1689
Refining estimates:
Iteration 0:   log pseudolikelihood =  -673.1689

Cox regression -- Efron method for ties

No. of subjects      =         2554                Number of obs   =      2554
No. of failures      =           98
Time at risk         =        23593
                                                   Wald chi2(3)    =     22.48
Log pseudolikelihood =    -673.1689                Prob > chi2     =    0.0001

                           (standard errors adjusted for clustering on stcode)
------------------------------------------------------------------------------
             |               Robust
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       south |   .4926406    .282232     1.75   0.081     -.060524    1.045805
  mooneymean |  -.1859751   .0688747    -2.70   0.007     -.320967   -.0509833
     asimean |   -.020338   .0078458    -2.59   0.010    -.0357154   -.0049606
------------------------------------------------------------------------------
```

Now, I'll compute the linear predictions and square them:

```
. predict xb, xb
```

```
. gen xb_2=xb^2
```

Now, let's reestimate the Cox model on these new covariates.

```
. stcox xb xb_2, efron nohr cluster(stcode) nolog

         failure _d:  rev_even
   analysis time _t:  duration

Cox regression -- Efron method for ties

No. of subjects      =         2554               Number of obs   =      2554
No. of failures      =           98
Time at risk         =        23593
                                                  Wald chi2(2)    =     20.04
Log pseudolikelihood =   -670.10644               Prob > chi2     =    0.0000

                           (standard errors adjusted for clustering on stcode)
------------------------------------------------------------------------------
             |               Robust
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          xb |   1.334259   .3071008     4.34   0.000     .7323523    1.936165
        xb_2 |  -.7748816   .4960223    -1.56   0.118    -1.747068    .1973043
------------------------------------------------------------------------------
```

What we're looking for is the coefficient for xb_2. If it is significantly different from 0, we may have "problems." A p-value of .12 is "close" to the conventional .10 level. Fortunately, the linktest is easy to do as a post-estimation command in Stata. After estimating the initial Cox model, we can perform the linktest:

```
. linktest, efron cluster(stcode) nolog

         failure _d:  rev_even
   analysis time _t:  duration

Cox regression -- Efron method for ties

No. of subjects      =         2554               Number of obs   =      2554
No. of failures      =           98
Time at risk         =        23593
                                                  Wald chi2(2)    =     20.04
Log pseudolikelihood =   -670.10644               Prob > chi2     =    0.0000

                           (standard errors adjusted for clustering on stcode)
------------------------------------------------------------------------------
             |               Robust
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        _hat |   1.334259   .3071008     4.34   0.000     .7323523    1.936165
      _hatsq |  -.7748816   .4960223    -1.56   0.118    -1.747068    .1973043
------------------------------------------------------------------------------
```

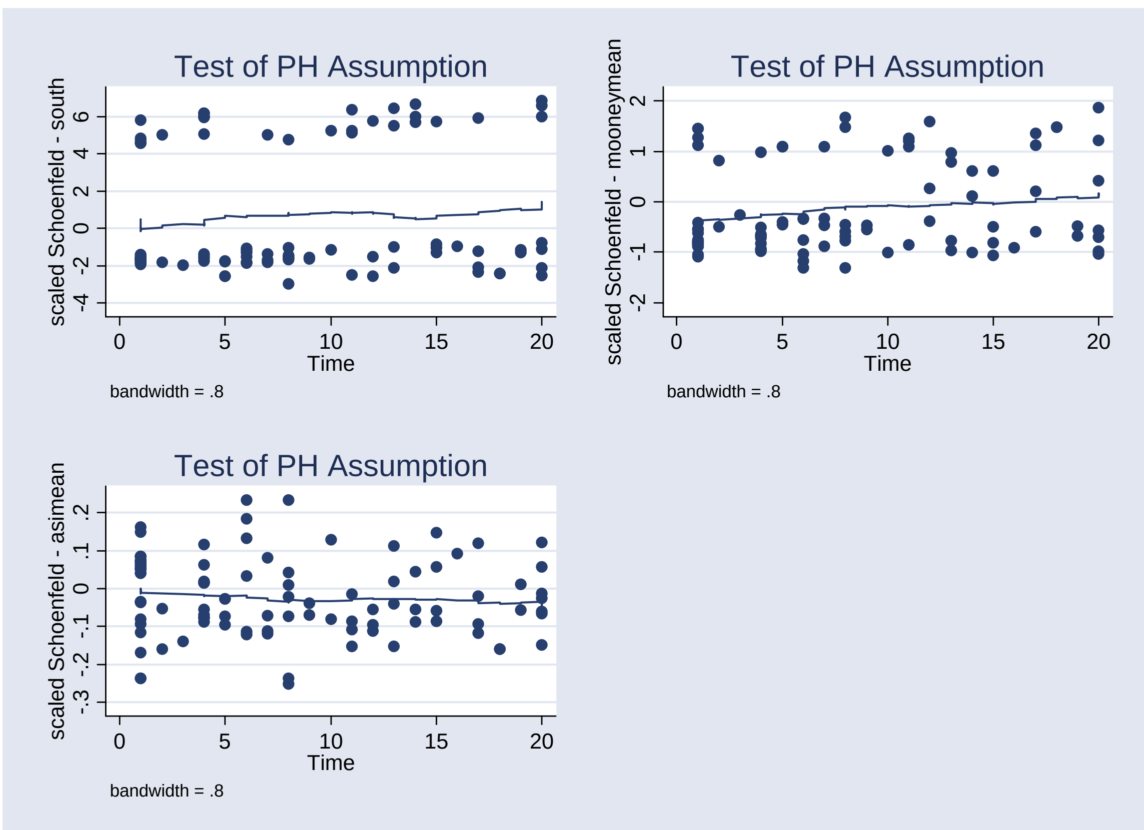
As they should be, the results are identical to our "by hand" linktest. Note that this diagnostic is not a test of the PH property per se. It is a general specification test (note its similarities to the White specification test). There are more formal tests of the PH property.

Residual Tests

In our Cox model from above, we calculated and saved several kinds of residuals. Among them were the Schoenfeld and scaled Schoenfeld residuals. These residuals can be used to evaluate the PH property. Graphs of the Schoenfeld residuals are useful diagnostic tools. To generate these, I did the following:

```
. stphtest, plot(south) saving(south)
(file south.gph saved)

. stphtest, plot(mooneymean) saving(mooneymean)
(file mooneymean.gph saved)

. stphtest, plot(asimean) saving(asimean)
(file asimean.gph saved)

. graph combine south.gph mooneymean.gph asimean.gph
```

This last command returns:



This is an “eyeball” test: are the slopes flat with respect to time? The Mooney coefficient may be problematic. As with any eyeball test, “truth” is in the eye of the beholder?

Perhaps a better test is a test based on the scaled Schoenfeld residuals. This test essentially asks whether or not the slope in the regression of time vs. residuals is flat (it should be if PH property holds; why?). Most software programs have this kind of residual regression test built in. In Stata, it entails the use of stphtest. For these data, I use the diagnostic:

```
. stphtest, detail

      Test of proportional hazards assumption

      Time:  Time
      ----------------------------------------------------------------
                  |       rho            chi2       df       Prob>chi2
      ------------+---------------------------------------------------
      south       |      0.13889         2.77        1         0.0960
      mooneymean  |      0.25704        11.04        1         0.0009
      asimean     |     -0.11105         2.13        1         0.1441
      ------------+---------------------------------------------------
      global test |                     15.91        3         0.0012
      ----------------------------------------------------------------

note: robust variance-covariance matrix used.
```

The test shows that the mooneymean covariate clearly violates the PH assumption. Depending on your *p*-value choice, you might also conclude the south covariate is problematic. For now, we’ll focus on the mooneymean coefficient.

What does the significant rho imply? It suggests that correlation between the scaled Schoenfeld residuals and the rank of the survival time (*t*) is significantly different from 0. The null is that rho is 0 (again, why?).

A variety of solutions have been suggested, chief among them is to interact the “offending covariate” with time. I do that.

First let’s try “linear” time and then log(t).

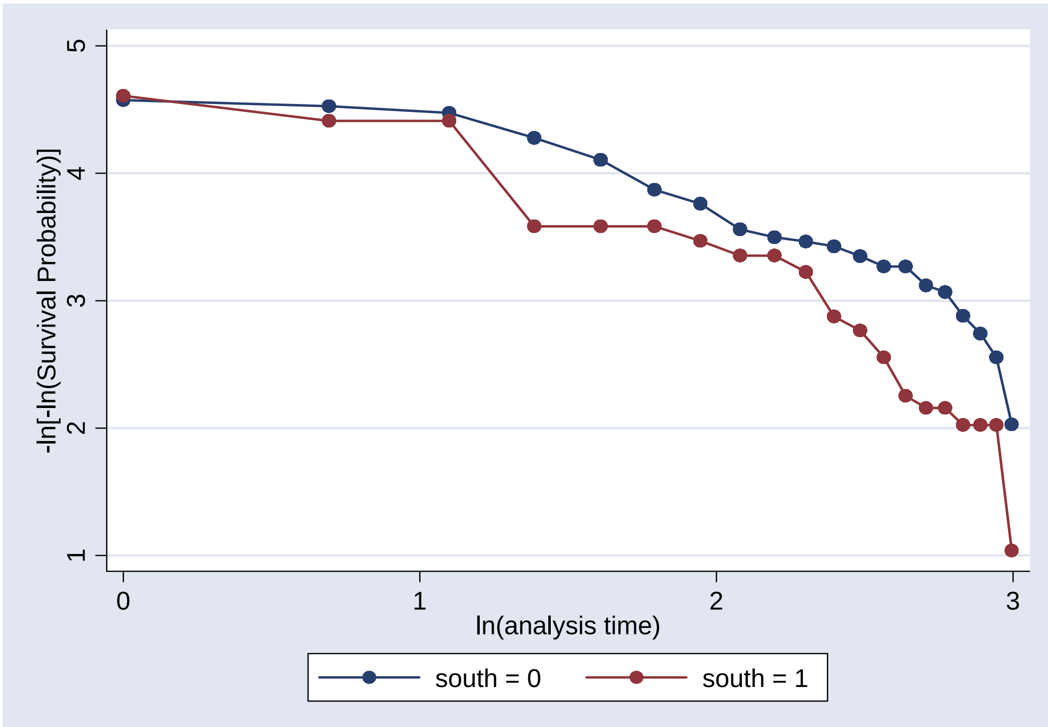
```
. gen mooney_t=mooneymean*_t

. gen logt=log(_t)

. gen mooney_logt=mooneymean*logt
```

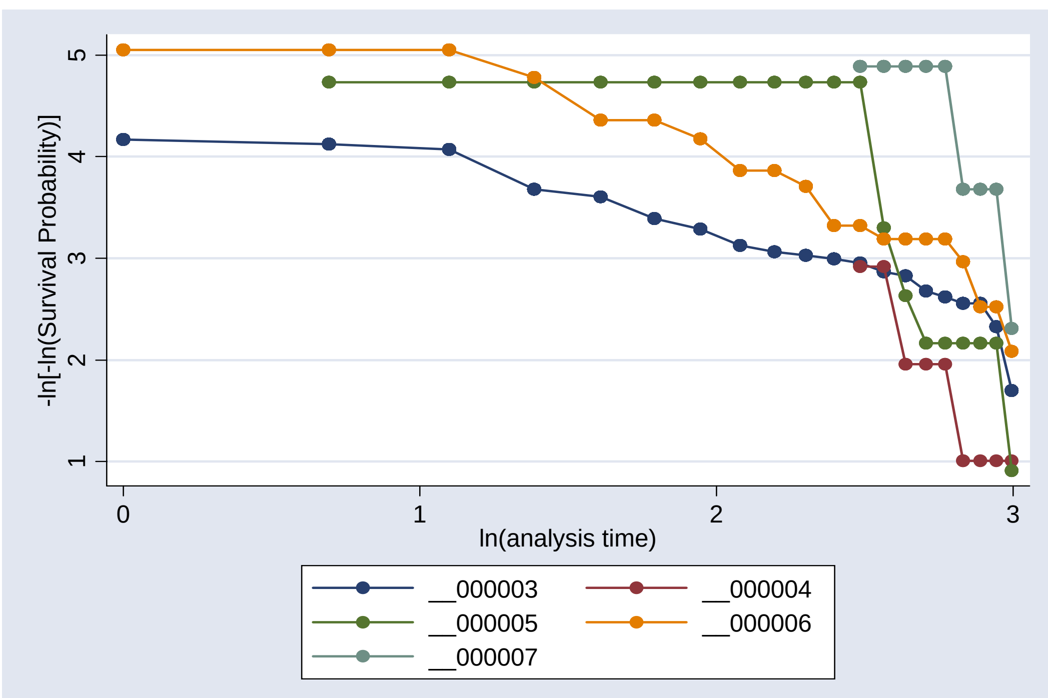
Let us rerun the regressions:

```
. stcox south mooneymean asimean mooney_t, efron nohr
cluster(stcode) nolog

         failure _d:  rev_even
   analysis time _t:  duration

Cox regression -- Efron method for ties

No. of subjects      =         2554                Number of obs   =      2554
No. of failures      =           98
Time at risk         =        23593
                                                   Wald chi2(4)    =     32.64
Log pseudolikelihood =   -652.15527                Prob > chi2     =    0.0000

                           (standard errors adjusted for clustering on stcode)
------------------------------------------------------------------------------
             |               Robust
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       south |   .5837776   .2790302     2.09   0.036     .0368884    1.130667
  mooneymean |  -1.201793   .5130948    -2.34   0.019     -2.20744   -.1961456
     asimean |  -.0221056   .0086217    -2.56   0.010    -.0390037   -.0052074
    mooney_t |   .0709123   .0300178     2.36   0.018     .0120786    .1297461
------------------------------------------------------------------------------


. stcox south mooneymean asimean mooney_logt, nolog efron nohr cluster(stcode)

         failure _d:  rev_even
   analysis time _t:  duration

Cox regression -- Efron method for ties

No. of subjects      =         2554                Number of obs   =      2554
No. of failures      =           98
Time at risk         =        23593
                                                   Wald chi2(4)    =     45.69
Log pseudolikelihood =   -636.60588                Prob > chi2     =    0.0000

                           (standard errors adjusted for clustering on stcode)
------------------------------------------------------------------------------
             |               Robust
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       south |   .5733904   .2624636     2.18   0.029     .0589712     1.08781
  mooneymean |  -2.102194   .7971802    -2.64   0.008    -3.664639     -.53975
     asimean |  -.0212366   .0089132    -2.38   0.017     -.038706   -.0037671
 mooney_logt |   .7540933   .2804719     2.69   0.007     .2043784    1.303808
------------------------------------------------------------------------------
```

The interaction with log(t) is preferred to linear(t). We see the interaction is statistically significantly imply that the effect of the mooneymean covariate is changing with respect to time. (If we were to redo the PH test from before, we would fail to reject the null that the PH property holds…which is what we want).

There are other graphics-based PH tests. Log-log plots are popular. The idea behind these kind of plots is simple. If the PH assumption holds, then the covariate effect over different levels of the covariate should be roughly "parallel." This means that the effects are not changing over time. For our purposes, I produce the log-log plot for the South and Mooneymean covariates.





The “parallel” property “holds” for the South covariate—barely; it does not hold for the Mooneymean covariate. Nicely, all of our tests have pointed to these general conclusions.

Other Residual Tests

Apart from the PH property, there are a variety of other diagnostics available for the Cox model.

Let’s turn to some model fit statistics and make use of the Cox-Snell residuals and martingale residuals. I will use the cabinet duration data.

First, estimate a model and save residuals:

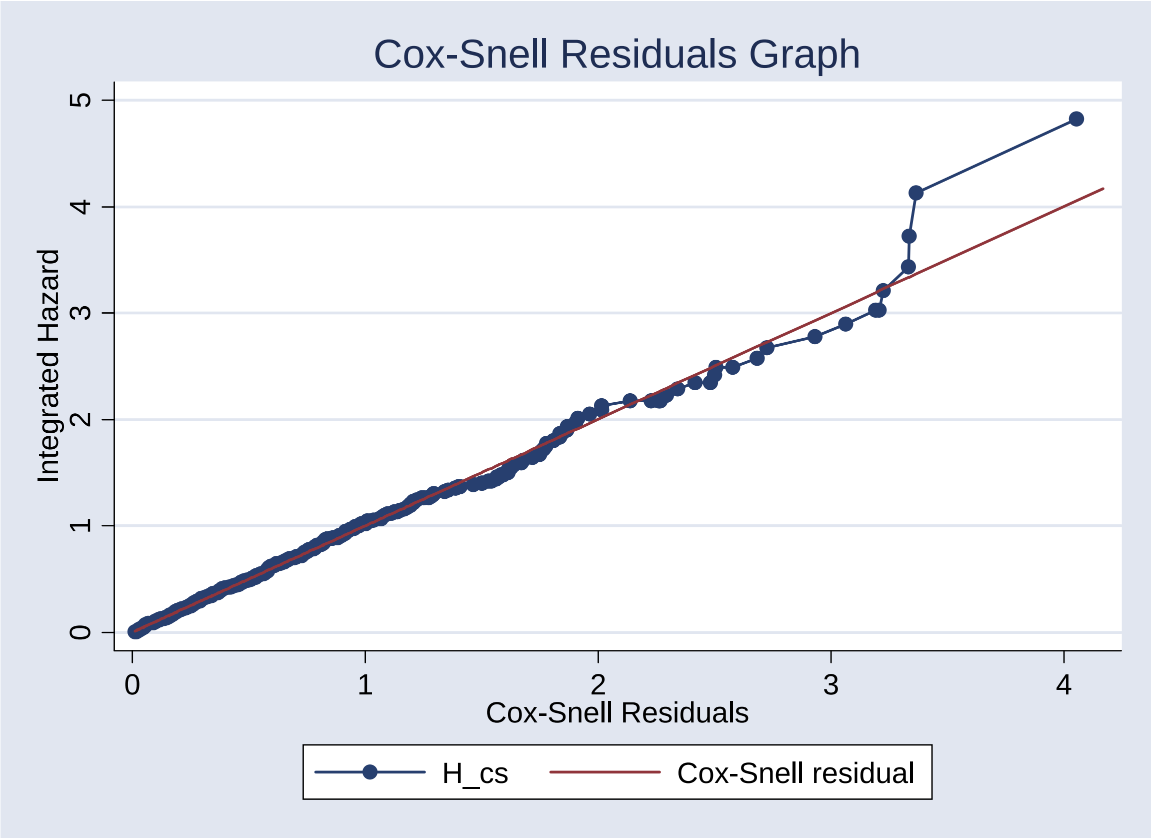
```
. stcox invest polar numst form postelec caretakr, nolog
nohr efron mgale(mg) robust

         failure _d:  censor
   analysis time _t:  durat

Cox regression -- Efron method for ties

No. of subjects      =          314               Number of obs   =        314
No. of failures      =          271
Time at risk         =       5789.5
                                                  Wald chi2(6)    =     181.81
Log pseudolikelihood =   -1287.7389               Prob > chi2     =     0.0000

------------------------------------------------------------------------------
             |               Robust
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      invest |   .3871388   .1331699     2.91   0.004     .1261305    .6481471
       polar |   .0233392   .0050912     4.58   0.000     .0133606    .0333177
       numst |  -.5826222   .1239658    -4.70   0.000    -.8255906   -.3396537
      format |    .130011   .0401728     3.24   0.001     .0512738    .2087481
    postelec |  -.8611202    .130801    -6.58   0.000    -1.117485   -.6047549
    caretakr |   1.710397   .2744381     6.23   0.000     1.172509    2.248286
------------------------------------------------------------------------------
```

Now I compute the Cox-Snell residuals and then compute the components for the residual graph (details will be given in class!) and then graph them:

```
. predict CoxSnell, csnell

. sts generate km=s

. gen double H_cs=-log(km)
```

```
. twoway (scatter H_cs CoxSnell, c(l)) (scatter CoxSnell
CoxSnell, c(l) s(i)), ytitle(Integrated Hazard )
xtitle(Cox-Snell Residuals) title(Cox-Snell Residuals
Graph) saving(icpsr_coxsnellresidcab, replace)
```

This returns:



What we're looking for is the slope of the integrated hazard. Ideally, the slope should be 1 (i.e. a 45° angle). These data mostly confirm the adequacy of the Cox model.

*****SIDETRIP TO PARAMETRIC-LAND*****

The Cox-Snell residuals can also be used in the parametric context in essentially the same way as for the Cox model. Let me illustrate.

First, I need to reset the data and then estimate a Weibull:

```
.  stset durat, failure(censor)

. streg invest polar  numst format postelec caretakr,
dist(weibull) time

         failure _d:  censor
   analysis time _t:  durat


Fitting constant-only model:

Iteration 0:   log likelihood = -500.17174
Iteration 1:   log likelihood = -500.04551
Iteration 2:   log likelihood =  -500.0455

Fitting full model:

Iteration 0:   log likelihood =  -500.0455
Iteration 1:   log likelihood = -441.92621
Iteration 2:   log likelihood = -415.20546
Iteration 3:   log likelihood = -414.07836
Iteration 4:   log likelihood = -414.07496
Iteration 5:   log likelihood = -414.07496

Weibull regression -- accelerated failure-time form

No. of subjects =          314                     Number of obs   =       314
No. of failures =          271
Time at risk    =       5789.5
                                                   LR chi2(6)      =    171.94
Log likelihood  =   -414.07496                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      invest |  -.2958188   .1059024    -2.79   0.005    -.5033838   -.0882538
       polar |   -.017943   .0042784    -4.19   0.000    -.0263285   -.0095575
       numst |   .4648894   .1005815     4.62   0.000     .2677533    .6620255
      format |  -.1023747   .0335853    -3.05   0.002    -.1682006   -.0365487
    postelec |   .6796125    .104382     6.51   0.000     .4750276    .8841974
    caretakr |   -1.33401   .2017528    -6.61   0.000    -1.729438   -.9385818
       _cons |   2.985428   .1281146    23.30   0.000     2.734328    3.236528
-------------+----------------------------------------------------------------
       /ln_p |    .257624   .0500578     5.15   0.000     .1595126    .3557353
-------------+----------------------------------------------------------------
           p |   1.293852   .0647673                      1.172939     1.42723
         1/p |   .7728858   .0386889                       .700658    .8525593
------------------------------------------------------------------------------
```

Now compute Cox-Snell residuals.

```
. predict double cs, csnel
```

Now restset the data to make the residuals “the data.”

```
. stset cs, failure(censor)
```

Now generate K-M estimates,

```
. sts generate km=s
```

and back out the estimated cumulative hazard:

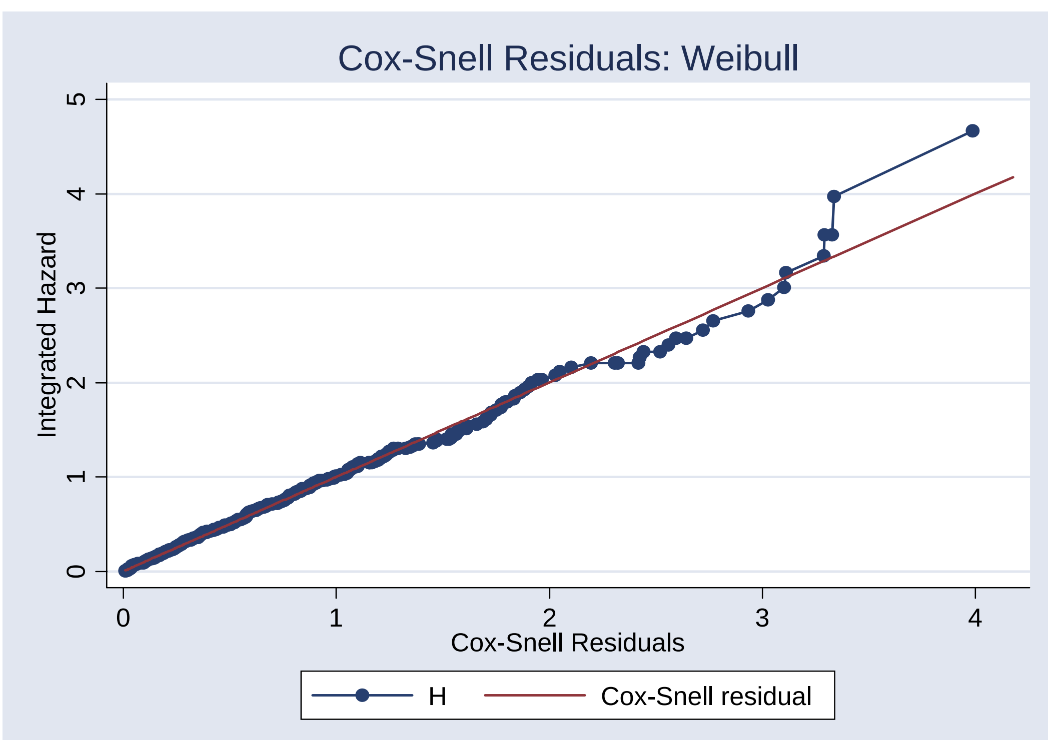
```
. gen double H=-log(km)
```

Now, I’ll graph it:

```
. sort cs
```

```
. twoway (scatter H cs, c(l)) (scatter cs cs, c(l) s(i)),
ytitle(Integrated Hazard ) xtitle(Cox-Snell Residuals) title(Cox-
Snell Residuals: Weibull) saving(icpsr_coxsnellresidweib,
replace)
```

This returns:

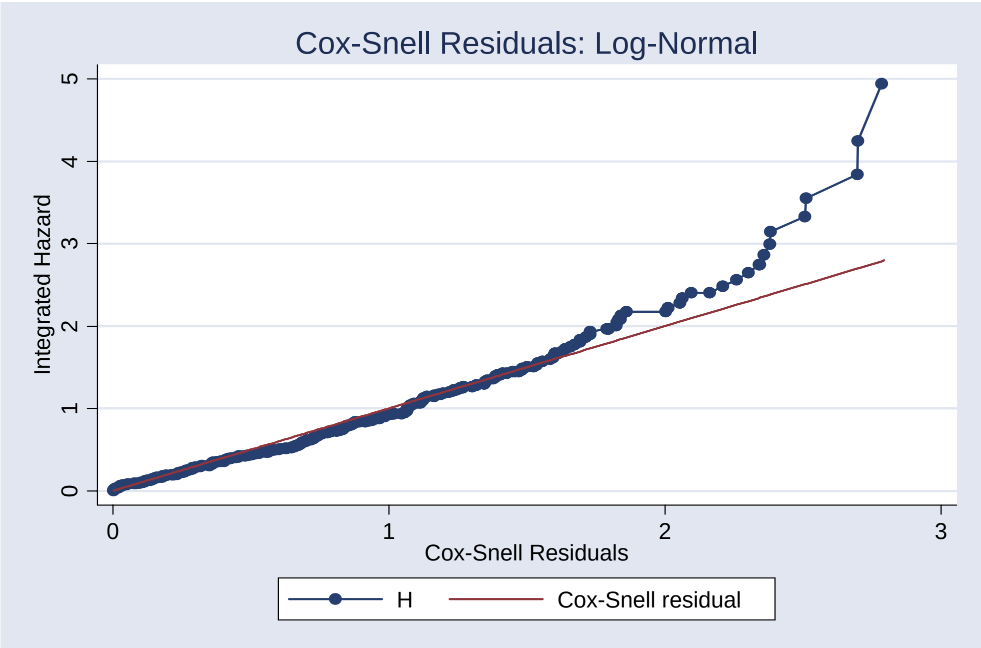


Again, what we're looking for is the slope of the integrated hazard to be 1 (i.e. a 45° angle). These data mostly confirm the adequacy of the Weibull model. Let's contrast this to a log-normal.

Below are the sequence of commands leading to the residual plot for the log-normal:

```
.  stset durat, failure(censor)

. streg invest polar  numst format postelec caretakr,
dist(lognorm) time  [output omitted]

. drop cs km H

. stset cs, failure(censor)

. sts generate km=s

. gen double H=-log(km)

. sort cs

. twoway (scatter H cs, c(l)) (scatter cs cs, c(l) s(i)),
ytitle(Integrated Hazard ) xtitle(Cox-Snell Residuals)
title(Cox-Snell Residuals: Log-Normal)
saving(icpsr_coxsnellresidlognorm, replace)
```

This returns:



This model seems to yield a greater departure from the 45° degree line. Importantly, recall from our parametric analyses of the cabinet data that both the generalized gamma and the AIC suggested the Weibull fit these data the best. It is comforting to find that residual tests also suggest that the Weibull seems to fit these data (I omit the other parametric models, but the conclusion is that the Weibull is the preferred model [see pp. 137—139 in our book]).

…Now back to the Cox model.

Martingale Residuals

Martingale residuals can be used to evaluate functional form of a covariate. This would seem to be a useful diagnostic. In class, I'll discuss martingales and the two approaches to evaluating functional form. Here, let me illustrate Approach 1.

First, I'll reset the data.

```
. stset durat, failure(censor)
```

Next, I'll estimate a Cox model (for illustrative purposes, I'm only going to use two covariates here):

```
. drop mg
. stcox format polar, exactp nohr mgale(mg) [output
omitted]

. lowess mg polar, ytitle(Martingale Residuals)
xtitle(Polarization Covariate) title(Approach 1)
saving(icpsr_martingalepolar1.gph, replace)

. lowess mg format, ytitle(Martingale Residuals)
xtitle(Formation Attempts Covariate) title(Approach 1)
saving(icpsr_martingaleformat1.gph, replace)
```

Now I'll illustrate Approach 2. First, let's look at the formation attempts model and graph its residuals against the polarization variable (the omitted covariate):

```
. drop mg

. stcox format, exactp nohr mgale(mg) nolog [output
omitted]

. lowess mg polar, ytitle(Martingale Residuals)
xtitle(Polarization Covariate) title(Approach 2)
saving(icpsr_martingalepolar2.gph, replace)
```

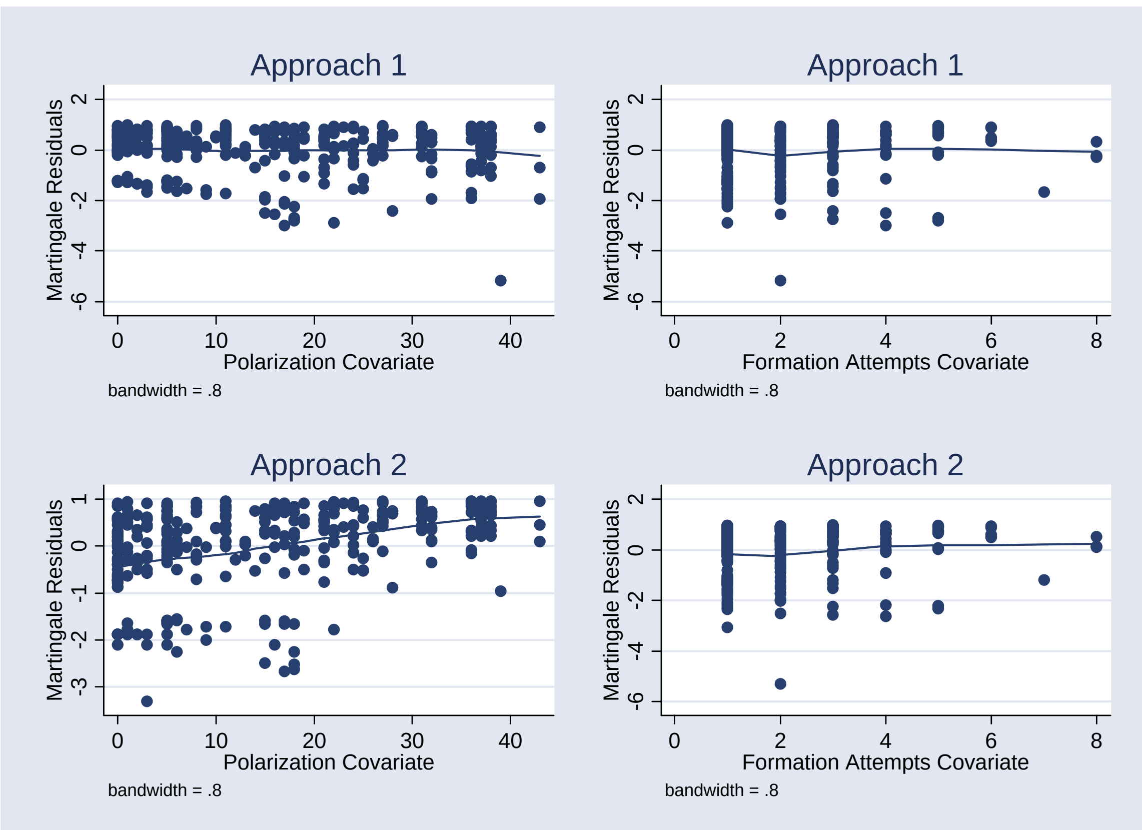
Now let's look at the polarization model and graph its residuals against the formation attempts variable (the omitted covariate):

```
. drop mg

. stcox polar, exactp nohr mgale(mg) nolog

. lowess mg format, ytitle(Martingale Residuals)
xtitle(Formation Attempts Covariate) title(Approach 2)
saving(icpsr_martingaleformat2.gph, replace)
```

Combining the four graphs from above, we obtain:

```
. graph combine icpsr_martingalepolar1.gph
icpsr_martingaleformat1.gph  icpsr_martingalepolar2.gph
icpsr_martingaleformat2.gph ,
saving(icpsr_combinedmartingales.gph, replace)
```

This returns:



What we're looking for is a slope of 0. Approach 2 for the polarization covariate seems to suggest some possible problems (though the slope ranges over a very, very small range). In general, Approach 1 and Approach 2 yield similar conclusions. If we thought there was a functional form problem, transforming the covariates or including interactions might be a useful thing to do.