Log-normal Regression with R[*]

```
> rm(list=ls());  options(scipen=999)
> # install.packages("survival",dependencies=TRUE) # Only need to do this once
> library(survival) # Do this every time
> # install.packages("asaur",dependencies=TRUE) # Only need to do this once
> library(asaur)
> # summary(pharmacoSmoking)
> attach(pharmacoSmoking)
> # Make patch only the reference category
> contrasts(grp) = contr.treatment(2,base=2)
> colnames(contrasts(grp)) = c('Combo') # Names of dummy vars -- just one
> DayOfRelapse = Surv(ttr+1,relapse) # Day of relapse starts with one.
> # Collapse race categories
> Race = as.character(race) # Small r race is a factor. This is easier to modify.
> Race[Race!='white'] = 'blackOther'; Race=factor(Race)
>
> wmod =  survreg(DayOfRelapse ~ grp + age + employment , dist='weibull')
> summary(wmod) # This was model full2 in an earlier analysis

Call:
survreg(formula = DayOfRelapse ~ grp + age + employment, dist = "weibull")
                  Value Std. Error     z             p
(Intercept)      1.4957     0.8414  1.78 0.07545324261
grpCombo         1.1023     0.3793  2.91 0.00365915983
age              0.0643     0.0186  3.45 0.00055474131
employmentother -1.2880     0.4617 -2.79 0.00527676297
employmentpt    -1.2123     0.5616 -2.16 0.03088499029
Log(scale)       0.5454     0.0894  6.10 0.00000000105

Scale= 1.73

Weibull distribution
Loglik(model)= -464.3   Loglik(intercept only)= -476.5
      Chisq= 24.31 on 4 degrees of freedom, p= 0.000069
Number of Newton-Raphson Iterations: 5
n= 125

>
> lognorm = survreg(DayOfRelapse ~ grp + age + employment,dist='lognormal')
> summary(lognorm)

Call:
survreg(formula = DayOfRelapse ~ grp + age + employment, dist = "lognormal")
                  Value Std. Error     z                          p
(Intercept)      0.8923     0.8719  1.02 0.306084649658437957242541
grpCombo         1.1248     0.4055  2.77 0.005534818036902075522043
age              0.0569     0.0182  3.13 0.001748007563975885296503
employmentother -1.0495     0.4766 -2.20 0.027666798779144716885492
employmentpt    -0.8757     0.6456 -1.36 0.174941022226478948287109
Log(scale)       0.7671     0.0798  9.61 0.000000000000000000000743

Scale= 2.15

Log Normal distribution
Loglik(model)= -460.4   Loglik(intercept only)= -470.4
      Chisq= 20 on 4 degrees of freedom, p= 0.0005
Number of Newton-Raphson Iterations: 3
n= 125

>
```

[*] Copyright information is on the last page.

```
> # Now predict the day of relapse for a 50-year-old in the patch-only condition
> # who is working part-time. Use the estimated median exp(muhat) as a prediction.
>
> oldguy = data.frame(grp='patchOnly',age=50,employment='pt')
> pred1 = predict(lognorm,newdata=oldguy,type='linear',se=TRUE) ; pred1

$fit
       1
2.861025

$se.fit
        1
0.6206626
```

se of $\widehat{y}_{n+1}$ is $\sqrt{\mathbf{x}_{n+1}^\top\widehat{\mathbf{C}}_n\mathbf{x}_{n+1}}$. Want *se* of $(y_{n+1}-\widehat{y}_{n+1}) = \sqrt{\widehat{\sigma}^2 + \mathbf{x}_{n+1}^\top\widehat{\mathbf{C}}_n\mathbf{x}_{n+1}}$.

```
>
> # Construct prediction interval
> yhat = pred1$fit
> sigmasqhat = lognorm$scale^2
> se = sqrt(sigmasqhat+pred1$se^2)
> L = yhat - 1.96*se; U = yhat + 1.96*se
> that= exp(yhat)
> lowerr95 = exp(L); upper95 = exp(U)
> pi = c(that,lower95,upper95)
> names(pi) = c('t-hat','lower95','upper95')
> pi
       t-hat      lower95      upper95
  17.4794407    0.2161495 1413.5165098
```

So with 95% confidence, the old guy will be able to hold out between 5.2 hours and 3.9 years. Thank you very much.

For the record, the LaTeX code is

```
$se$ of $\widehat{y}_{n+1}$ is
$\sqrt{\mathbf{x}_{n+1}^\top\widehat{\mathbf{C}}_n\mathbf{x}_{n+1}}$.
Want $\sqrt{\widehat{\sigma}^2 + \mathbf{x}_{n+1}^\top
\widehat{\mathbf{C}}_n\mathbf{x}_{n+1}}$
```

Predicting Mark in University Calculus using Normal Regression

```
> good = lm(ucalc ~ diagtest + hsgpa + hscalc + hsengl + frstlang); summary(good)

Call:
lm(formula = ucalc ~ diagtest + hsgpa + hscalc + hsengl + frstlang)

Residuals:
    Min      1Q  Median      3Q     Max 
-50.441  -6.719   1.546   9.036  32.288 

Coefficients:
              Estimate Std. Error t value         Pr(>|t|)    
(Intercept)   -71.4517    11.1072  -6.433 0.00000000053717 ***
diagtest        0.9721     0.2518   3.860         0.000141 ***
hsgpa           1.5892     0.2209   7.195 0.00000000000569 ***
hscalc          0.2176     0.1021   2.132         0.033908 *  
hsengl         -0.3002     0.1227  -2.447         0.015017 *  
frstlangOther   4.6966     2.1264   2.209         0.028003 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.6 on 281 degrees of freedom
  (292 observations deleted due to missingness)
Multiple R-squared:  0.4635,	Adjusted R-squared:  0.454 
F-statistic: 48.56 on 5 and 281 DF,  p-value: < 0.00000000000000022

> round( predict(good,interval='prediction'), 1)
      fit  lwr   upr
1    44.6 17.6  71.6
2    30.4  3.2  57.7
3    59.5 32.6  86.4
4    62.5 35.5  89.5
5    69.2 42.3  96.1
6    43.3 16.2  70.5
7    68.8 41.7  96.0
8    44.6 17.6  71.6
9    46.9 19.8  74.0
10   78.2 51.2 105.2
14   72.2 45.2  99.3
17   61.5 34.3  88.7
18   80.0 53.0 107.0
19   52.4 25.5  79.3
21   59.4 32.2  86.5
22   68.5 41.6  95.5
23   59.8 32.6  86.9
25   64.8 37.8  91.8
26   48.8 21.9  75.7
28   63.3 36.3  90.3
29   59.8 32.9  86.7
32   46.9 20.0  73.9
33   58.9 31.5  86.3
34   69.6 42.6  96.6
36   50.5 23.2  77.8
43   71.0 43.7  98.3
45   65.5 38.3  92.7
46   39.4 12.4  66.3
49   55.1 28.2  81.9
50  100.4 72.5 128.3
51   62.1 35.1  89.0
52   47.9 21.0  74.8
53   85.1 58.0 112.2
55   80.7 53.5 107.9
57   59.3 32.1  86.6
```

```
65   43.9 16.7  71.1
67   57.1 30.3  84.0
71   56.4 29.5  83.2




511  57.2 30.3  84.0
513  68.2 41.2  95.2
517  56.0 29.1  82.8
520  57.7 30.6  84.9
522  64.1 37.1  91.1
526  79.4 52.4 106.4
527  58.3 31.2  85.4
528  52.9 25.8  80.1
531  48.6 21.5  75.6
534  58.8 31.9  85.6
536  57.1 30.1  84.2
537  46.1 18.7  73.5
538  73.8 46.8 100.7
539  37.8 10.8  64.8
542  38.8 11.7  65.9
543  58.3 31.5  85.2
544  50.5 23.6  77.3
549  66.9 39.9  93.8
551  44.8 17.9  71.7
552  59.2 32.4  86.1
556  76.3 49.1 103.6
559  51.6 24.6  78.6
560  47.2 20.3  74.2
561  52.9 26.0  79.7
562  84.2 57.2 111.3
563  54.2 27.3  81.1
564  69.1 42.2  96.1
565  57.3 30.4  84.2
566  78.2 51.2 105.3
567  74.3 47.1 101.5
568  65.1 38.2  92.1
570  70.8 43.8  97.8
571  52.6 25.6  79.6
572  68.2 41.3  95.2
574  56.7 29.7  83.7
576  64.7 37.6  91.7
577  50.5 23.4  77.6
579  57.6 30.7  84.5

Warning message:
In predict.lm(good, interval = "prediction") :
  predictions on current data refer to _future_ responses
--------------------------------------------------------------------------------
```