Survival Models for the Social and Political Sciences
Week 7: Parametric Models

JEFF GILL
Professor of Political Science
Professor of Biostatistics
Professor of Surgery (Public Health Sciences)
Washington University, St. Louis

Checking Model Assumptions: Proportionality

- Return to the proportional hazards assumption, using the `births` dataset:

```
lapply(c("eha","survival"),library, character.only=TRUE)
data(births)
fert3 <- fert[fert$parity == 2,]    # PREVIOUS BIRTH WAS SECOND CHILD
fit4 <- coxreg(Surv(next.ivl, event) ~ ses + age + year + parish, data=fert3)
```

- Check the PH assumption with `cox.zph(fit4)`:

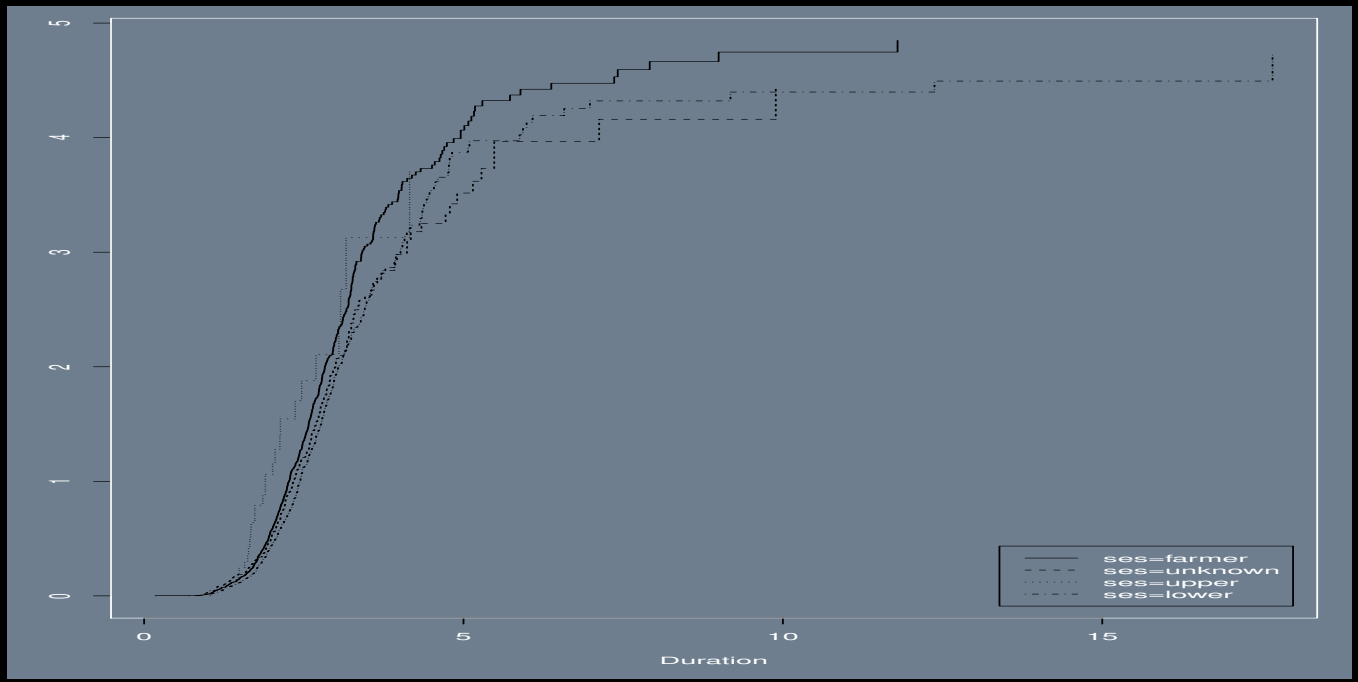
```
                rho   chisq       p
sesunknown -0.02773  1.1235 0.28918
sesupper   -0.05311  4.2624 0.03896
seslower    0.04774  3.4199 0.06441
age        -0.07322  6.8550 0.00884
year       -0.00315  0.0156 0.90071
parishNOR   0.00907  0.1234 0.72538
parishSKL   0.01883  0.5333 0.46524
GLOBAL           NA 20.1604 0.00523
```

where the p-values on `age` and `GLOBAL` are clearly in the tail showing that the PH assumption does not hold for those.

Checking Model Assumptions: Proportionality

- We can also see this graphically by looking at the categorical variable of `ses` (`age` is semi-continuous making this harder to test):

```
fit5 <- coxreg(Surv(next.ivl, event) ~ strata(ses)+age+year+parish,data=fert3)
par(mar=c(5,3,1,1),mfrow=c(1,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit5)
```



Checking Model Assumptions: Proportionality

- So we still have a problem with the PH issue.
- Check with the `cox.zph` function again:

```
cox.zph(fit5)
                rho   chisq       p
age       -0.074152 6.94200 0.00842
year       0.000807 0.00101 0.97469
parishNOR  0.010598 0.16842 0.68152
parishSKL  0.021465 0.69300 0.40515
GLOBAL           NA 9.51235 0.04949
```

- Showing that age is still an issue.
- Since age is continuous we need to categorize it to use `strata()`.

Checking Model Assumptions: Proportionality

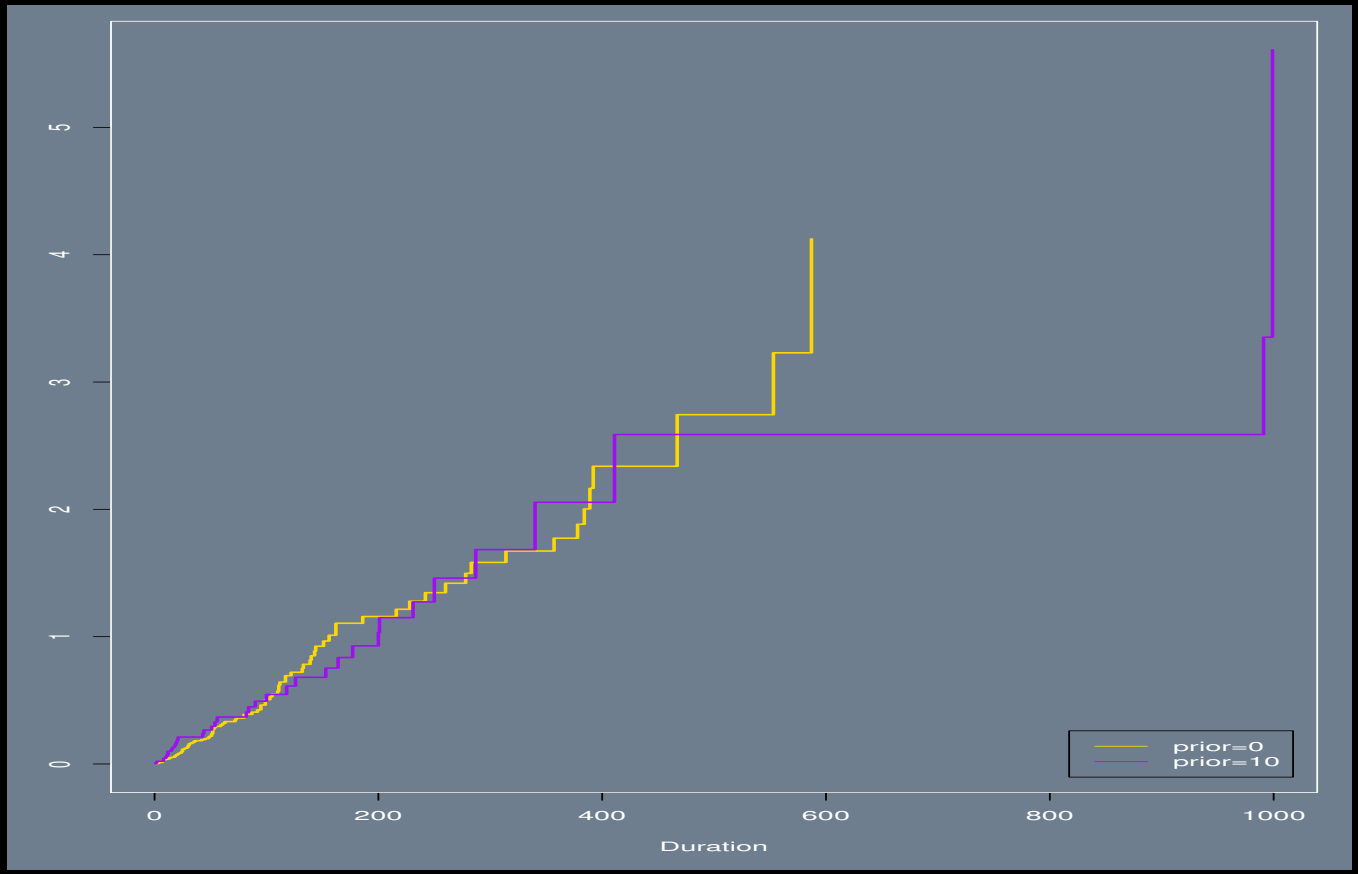
- Sometimes there are theoretical cutpoints and sometimes you have to search for good splits.
- The Broström book uses four:

```
fit6 <- coxreg(Surv(next.ivl, event) ~ strata(cut(fert3$age,4))
    + strata(ses) + year + parish, data=fert3)
cox.zph(fit6)
             rho  chisq     p
year      0.00489 0.0382 0.845
parishNOR 0.00526 0.0417 0.838
parishSKL 0.01607 0.3906 0.532
GLOBAL         NA 1.1273 0.770
```

- This indicates no evidence of a further problem.

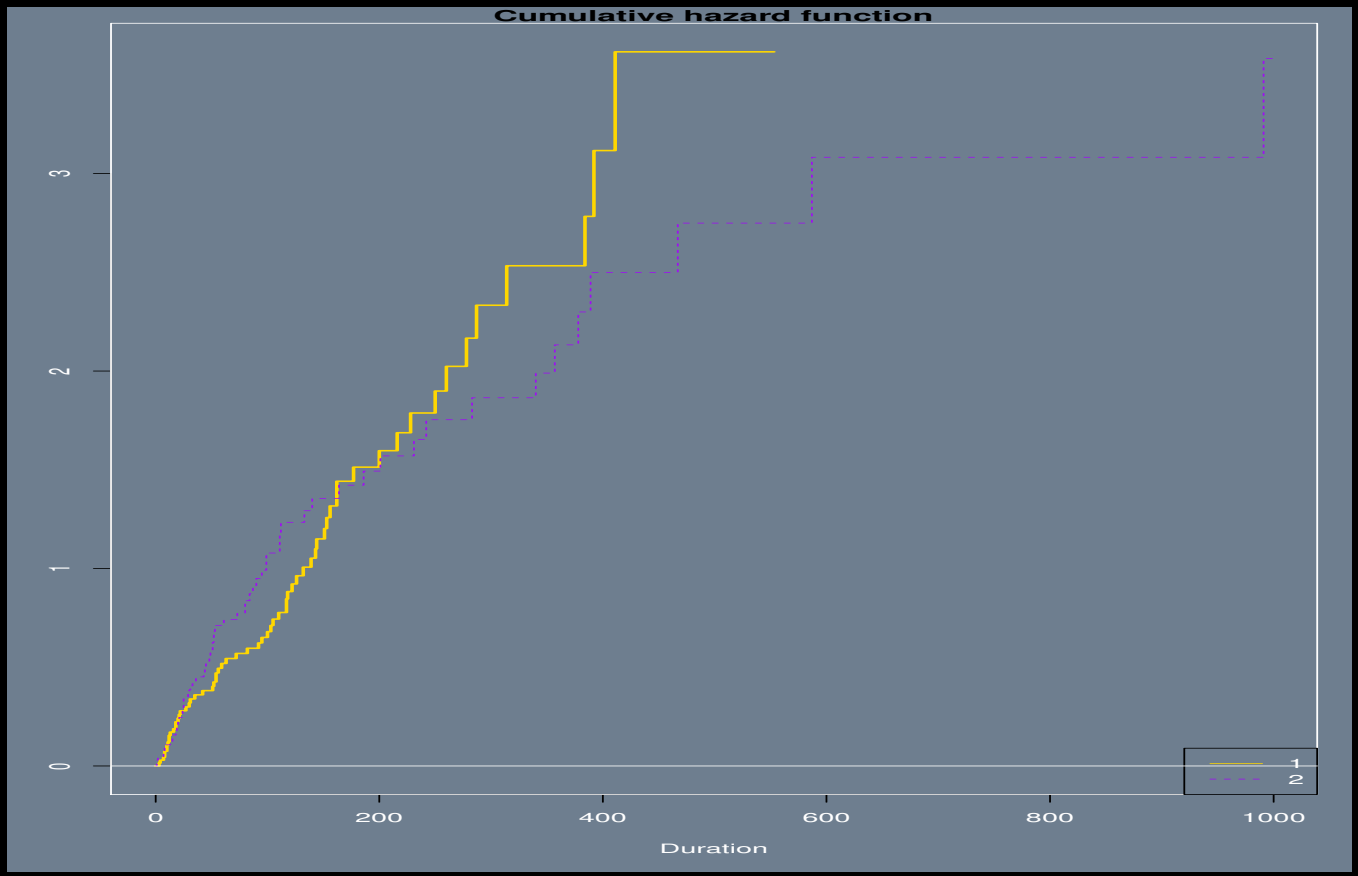
Checking Proportionality Example: VA Lung Cancer

```
postscript("Class.Survival/Images/veteran.strat.ps")
par(mar=c(5,3,1,1),mfrow=c(1,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(strat.fit2,fun="status",col=c("gold","purple"),lwd=2)
dev.off()
```



Checking Proportionality Example: VA Lung Cancer

```
postscript("Class.Survival/Images/veteran.strat2.ps")
par(mar=c(5,3,1,1),mfrow=c(1,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
with(veteran, plot(Surv(time,status), strat=trt,col=c("gold","purple"),lwd=2))
dev.off()
```



Checking Model Assumptions: Log-Linearity

- As we have seen the effect of the explanatory variables on the hazard is log-linear.
- This means that the link function between the linear additive component and the log of the hazard ratio is linear.
- Sometimes transformations are necessary as in regular LM or GLM modeling.
- Rarely the Martingale residuals plot can sometimes be helpful.
- Solution: lots of plots of explanatory variables against the outcome variable keeping track of the log-linear transformation.

Parametric Models

- All of the models we have studied from the Broström book so far are semiparametric models in that an explicitly underlying density was not directly specified.
- Here we look at the: Weibull, piece-wise constant hazards, lognormal, log-logistic, extreme value, and Gompertz distributions.
- These distributions can be extended and generalized.
- See also Appendix B in the Broström book.
- Note that the book incorrectly says "extremely significant" on pages 88 and 90, and "not that statistically significant" on page 122. Please don't say such things.
- Skip Broström Sections 6.2.3, 6.2.4.1, 6.2.5.

A Reminder On Proportional Hazards

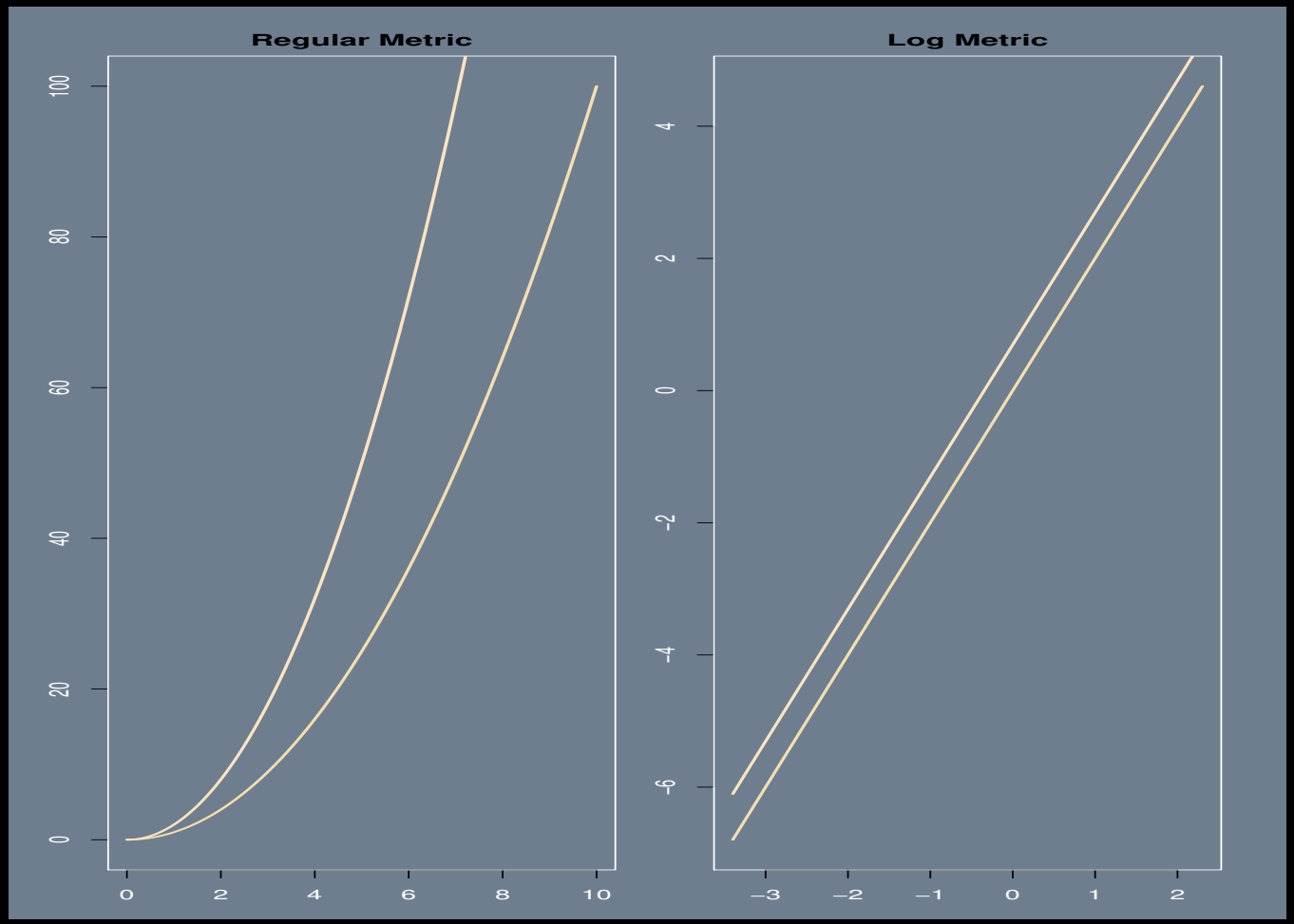
- This chapter is about moving beyond proportional hazards by imposing a distributional assumption.
- To increase our intuition, consider the simple functions:

$$h_0(t) = t^2, \quad h_1(t) = 2h_0(t) = 2t^2, \quad t \geq 0.$$

- Let's plot these on the regular and log scale:

```
postscript("Class.Survival/Images/ph.illustration.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),mfrow=c(1,2),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
dur <- seq(0,10,length=300)
plot(dur,dur^2,type="l",lwd=2,col="wheat", main="Regular Metric")
lines(dur,2*dur^2,lwd=2,col="bisque")
plot(log(dur),log(dur^2),type="l",lwd=2,col="wheat", main="Log Metric")
lines(log(dur),log(2*dur^2),lwd=2,col="bisque")
dev.off()
```

A Note On Proportional Hazards



The Weibull Model

- The Weibull distribution is described by:
  - PDF: $w(x|\gamma,\beta) = \frac{\gamma}{\beta}x^{\gamma-1}\exp\left(-\left(\frac{x}{\beta}\right)^{\gamma}\right)$ if $x \geq 0$ and 0 otherwise, where: $\gamma, \beta > 0$.
  - $E[\mathbf{X}_{ij}] = \beta\Gamma\left[1+\frac{1}{\gamma}\right]$.
  - $\text{Var}[X_{ij}] = \beta^2\left(\Gamma\left[1+\frac{2}{\gamma}\right] - \gamma\left[1+\frac{1}{\gamma}\right]^2\right)$.
- In the Broström book the Weibull hazard function is identified by:

$$h(t|p,\lambda) = \frac{p}{\lambda}\left(\frac{t}{\lambda}\right)^{p-1}, \quad t,p,\lambda > 0.$$

- Now specify:

$$h_1(t) = t^{p-1}, \quad p \geq 0, t > 0$$

with fixed p, leading to a proportional hazards *family*:

$$h_c = ct^{p-1} = \left(\frac{p}{\lambda^p}\right)t^{p-1} = \frac{p}{\lambda}\left(\frac{t}{\lambda}\right)^{p-1},$$

where for any fixed $p > 0$ a proportional hazards family is generated by altering λ (but not for altering p).

The Weibull Model

- Therefore the Weibull determines a collection of families of proportional hazards where each family is determined by the value of p, but varies by λ.
- The Weibull model is a *proportional hazards parametric survival model.*
- We get the regression model form by altering the previous statement according to:

$$h(t|\mathbf{x}, \lambda, p, \boldsymbol{\beta}) = \frac{p}{\lambda}\left(\frac{t}{\lambda}\right)^{p-1} \exp(\mathbf{x}\boldsymbol{\beta}), \quad t > 0$$

- This is supplied by the `phreg` function in `eha`.
- Returning to the Swedish fertility data:

```
library(eha)
data(fert)
fert1 <- fert[fert$parity == 1,]
fert1$Y <- Surv(fert1$next.ivl, fert1$event)
```

Weibull Model of the Fertility Data

- So we have added a Y variable that is `next.ivl` in survival format:

```
fert1[1:10,]
   id parity age year next.ivl event prev.ivl     ses parish       Y
2   1      1  25 1826   22.348     0    0.411  farmer    SKL 22.348+
4   2      1  19 1821    1.837     1    0.304 unknown    SKL  1.837
11  3      1  24 1827    2.051     1    0.772  farmer    SKL  2.051
21  4      1  35 1838    1.782     0    6.787 unknown    SKL  1.782+
23  5      1  28 1832    1.629     1    3.031  farmer    SKL  1.629
28  6      1  25 1829    1.730     1    0.819   lower    SKL  1.730
37  7      1  22 1826    0.988     1    0.465  farmer    SKL  0.988
41  8      1  23 1828    2.418     1    1.172  farmer    SKL  2.418
50  9      1  22 1827    1.125     1    0.882  farmer    SKL  1.125
65 10      1  19 1824    1.218     1    1.350  farmer    SKL  1.218
```

- So what the *matrix* `22.348+` means is that the interval between the first child and the event is 22.348 years, but the `+` means that the event was a right censoring because of the end of the study.

- Recall that `parity` is the order of previous birth and we selected on 1.

Weibull Model of the Fertility Data

- Previously we had `Surv()` inside the model statement but now we have it embedded into the data.
- The Weibull model is produced by:

```
fit.w1 <- phreg(Y ~ age + year + ses, data=fert1 dist="weibull")
summary(fit.w1)

Covariate          W.mean      Coef Exp(Coef)  se(Coef)    Wald p
age                27.151    -0.039     0.961     0.005     0.000
year             1858.664     0.005     1.005     0.002     0.022
ses
          farmer    0.449     0         1           (reference)
         unknown    0.190    -0.359     0.698     0.070     0.000
           upper    0.024    -0.382     0.683     0.174     0.028
           lower    0.336    -0.097     0.908     0.056     0.085

log(scale)                    6.825              2.830     0.016
log(shape)                    0.299              0.015     0.000

Events                    1657
Total time at risk        4500.5
Max. log. likelihood       -3140
LR test statistic         74.65
Degrees of freedom        5
Overall p-value           1.09912e-14
```

Weibull Model of the Fertility Data

- Everything is read as with previous proportional hazard models, except the parametric component.
- We read:

`log(scale)`	`6.825`	`2.830`	`0.016`
`log(shape)`	`0.299`	`0.015`	`0.000`

  in the following way:

  - Both terms are statistically significant at $\alpha = 0.05$.
  - Scale is λ, which is in the denominator of $h(t|\mathbf{x}, \lambda, p, \boldsymbol{\beta})$, so this is a reductive scaling *untied* to the progression of time.
  - Shape is p, and $\exp(0.299) = 1.34851$, which we can insert into the kernel to obtain $1.34851(t)^{0.34851}$, which means a declining instantaneous hazard over time.
  - So a woman is at declining risk of having a second baby over time.

Weibull Model of the Fertility Data

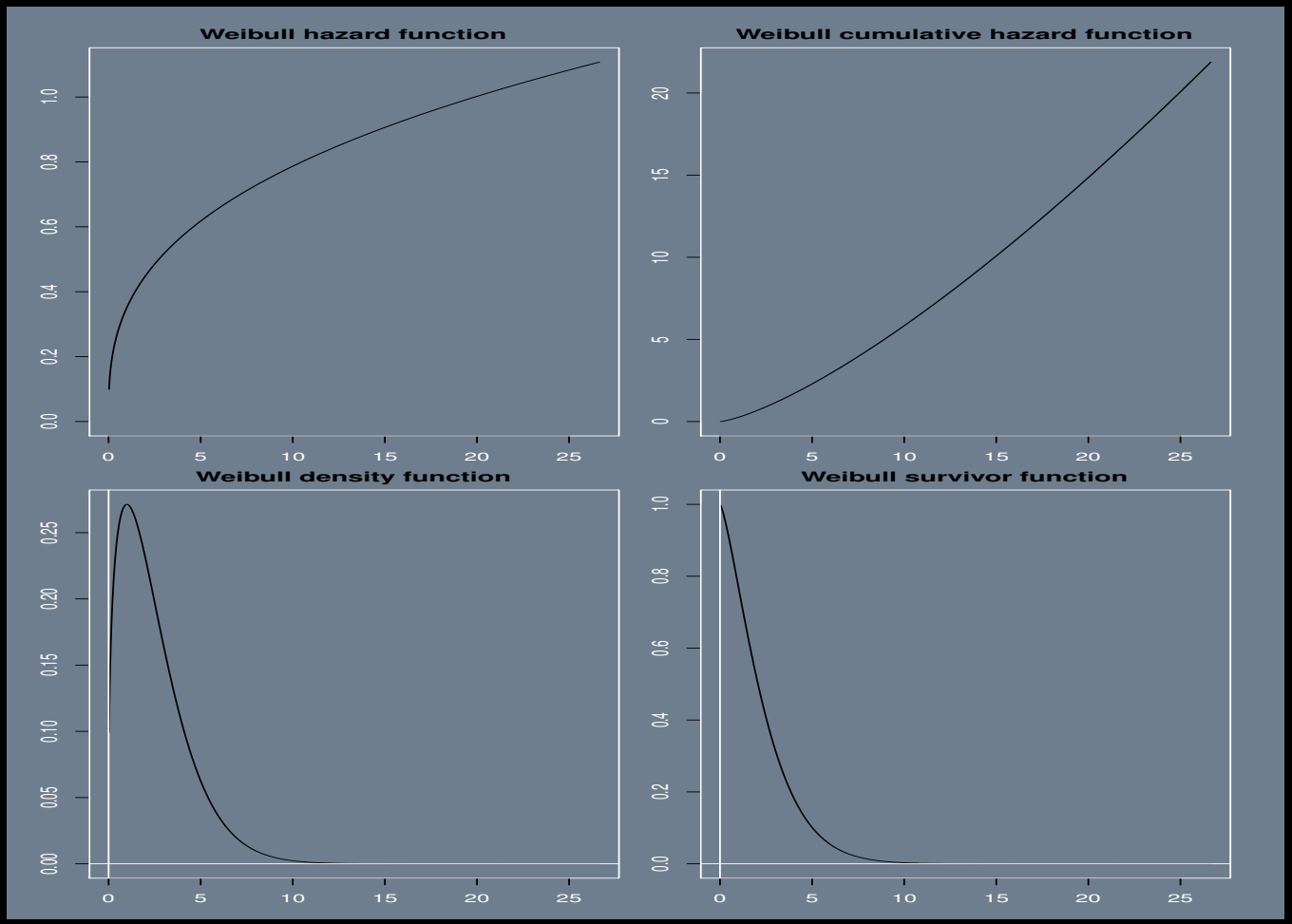
- The likelihood ratio tests are given by:

```
drop1(fit.w1,test="Chisq")
Single term deletions

Model:
Y ~ age + year + ses
       Df  AIC   LRT Pr(>Chi)
<none>    6290
age     1 6342 53.72 2.31e-13
year    1 6293  5.27   0.0217
ses     3 6314 30.07 1.34e-06
```

- We get a complete graphical view with one statement:

```
postscript("Class.Survival/Images/weibull.fert.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.w1)
dev.off()
```

Weibull Model of the Fertility Data



The Lognormal Model

- The lognormal is characterized by:
  - PDF: $\mathcal{LN}(x|\mu,\sigma) = (2\pi\sigma^2)^{-\frac{1}{2}}x^{-1}\exp[-(\log(x)-\mu)^2/2\sigma^2], \quad -\infty < \mu, x < \infty, 0 < \sigma^2$
  - $E[X] = \exp[\mu + \sigma^2/2]$
  - $\text{Var}[X] = \exp[2(\mu+\sigma^2)] - \exp[2\mu+\sigma^2]$.
- So if X is distributed normal, then $Y = \exp(X)$ is distributed lognormal.
- This version of a parametric model is not as "well-behaved" as the Weibull since it does not have closed-form solutions for the hazard and survivor functions.
- It is a three parameter family of distributions that changes family form by multiplying the hazard function with a positive constant: μ, σ^2, and the multiplier.

Lognormal Model of the Fertility Data

- Now run the same model from the fertility data with the lognormal specification:

```
fit.l1 <- phreg(Y ~ age + year + ses, data=fert1, dist="lognormal")
summary(fit.l1)
Covariate          W.mean      Coef Exp(Coef)  se(Coef)    Wald p
(Intercept)                  -5.784              3.853     0.133
age                27.151    -0.046     0.955     0.005     0.000
year             1858.664     0.003     1.003     0.002     0.102
ses
          farmer     0.449     0         1           (reference)
         unknown     0.190    -0.269     0.764     0.070     0.000
           upper     0.024    -0.234     0.792     0.173     0.177
           lower     0.336    -0.088     0.915     0.056     0.115

log(scale)                    0.466              0.049     0.000
log(shape)                    0.832              0.045     0.000

Events                    1657
Total time at risk        4500.5
Max. log. likelihood      -2584.1
LR test statistic         90.52
Degrees of freedom        5
Overall p-value           0
```

Lognormal Model of the Fertility Data

- The likelihood ratio tests are given by:

```
drop1(fit.l1,test="Chisq")
Single term deletions

Model:
Y ~ age + year + ses
       Df  AIC   LRT Pr(>Chi)
<none>    5178
age     1 5250 74.13  < 2e-16
year    1 5179  2.67 0.102183
ses     3 5189 16.27 0.000998
```

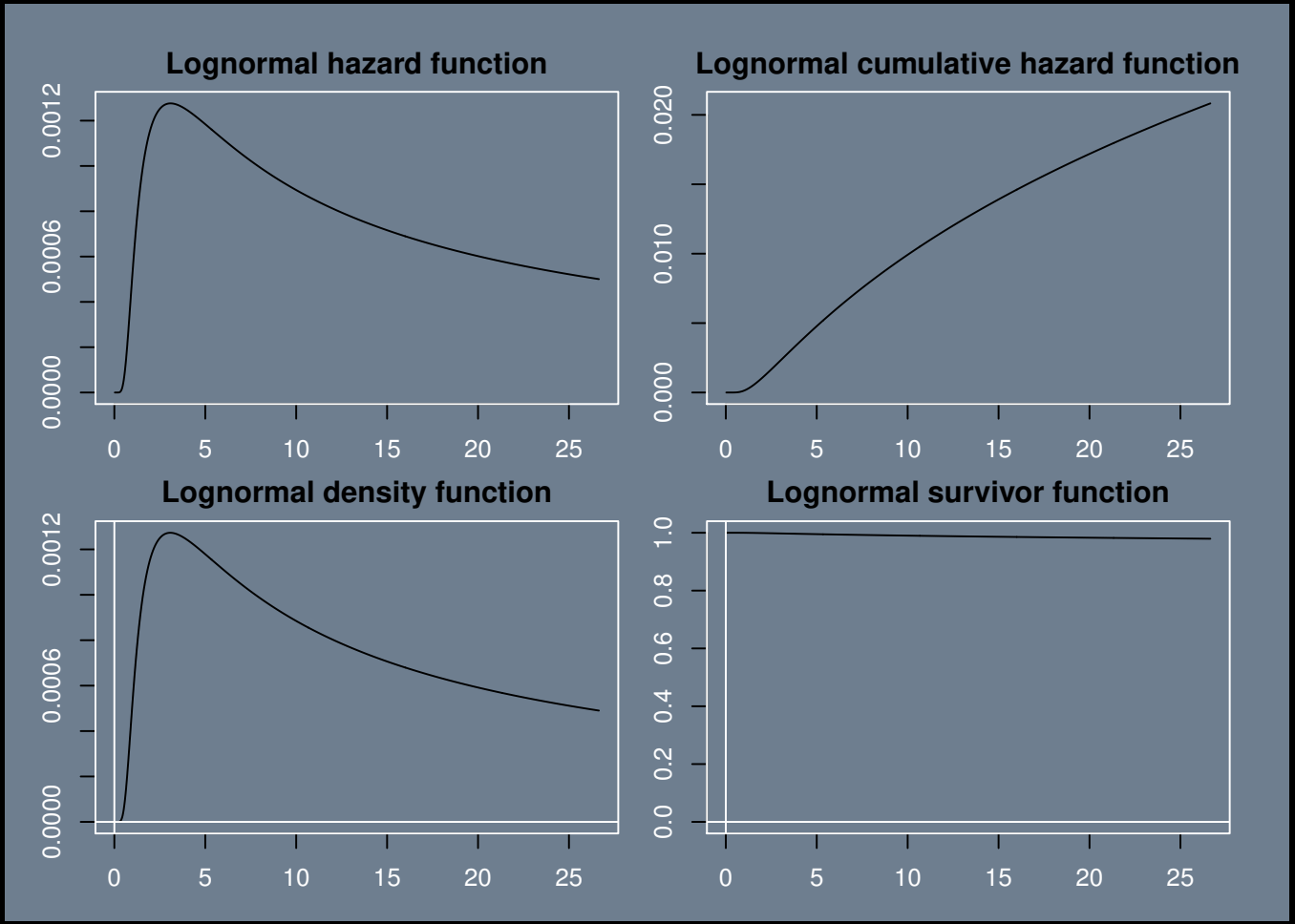
  which shows two of the three variables to be useful.

Lognormal Model of the Fertility Data

- As with the Weibull, risk declines with age, farmers' wives are more fertile, and the model fits well overall.
- However, unlike the Weibull, the hazard function does not have to be monotonic and here it increases until year 3 and then decreases.
- We get a complete graphical view with one statement (again):

```
postscript("Class.Survival/Images/lognormal.fert.ps",width=7.2,height=5.2)
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.l1)
dev.off()
```

Lognormal Model of the Fertility Data



The Piece-Wise Constant Hazards Model

- This model allows us to "slice" the time interval any way we like and then fit different constant hazards within the slices.
- We can do this to improve fit or because there are different epochs in the data.
- Returning to the subsetted fertility data, fit a PCH model:

```
fit.p1 <- phreg(Surv(next.ivl,event) ~ age + year + ses, data=fert1,
    dist="pch", cuts=c(4,8,12))
summary(fit.p1)
```

```
 Error in substr(covar.names[covar.no], 16) :
argument "stop" is missing, with no default
```

The Piece-Wise Constant Hazards Model

- So I wrote the summary from scratch:

```
fit.p1.null <- phreg(Surv(next.ivl,event) ~ 1, data=fert1, dist="pch",
    cuts=c(4,8,12))
tab1 <- cbind( fit.p1$coefficients, exp(fit.p1$coefficients),
    sqrt(diag(fit.p1$var)),
    2*(1-pnorm(abs(fit.p1$coefficients)/sqrt(diag(fit.p1$var)))) )
tab1 <- rbind(tab1[1:2,],c(0,1,NA,NA),tab1[3:5,])
dimnames(tab1)[[1]][3] <- "farmer"
dimnames(tab1)[[2]] <- c("Coef","Exp(Coef)","se(Coef)","Wald p")

tab2 <- data.frame(
"Events"=fit.p1$events,
"Total time at risk"=sum(fit.p1$y[,2]),
"Max log likelihood"=round(fit.p1$loglik[2],3),
"LR test statistic"=round(2*(fit.p1$loglik[2]-fit.p1.null$loglik[2]),3),
"Degrees of freedom"=fit.p1$df,
"Overall p-value"=pchisq(2*(fit.p1$loglik-fit.p1.null$loglik),
    df=fit.p1$df,lower.tail=FALSE)[2] )
```

The Piece-Wise Constant Hazards Model

- This results in:

```
tab1
                  Coef Exp(Coef)   se(Coef)      Wald p
age         -0.03101767  0.969458 0.00554863 2.26878e-08
year         0.00103979  1.001040 0.00210639 6.21564e-01
farmer       0.00000000  1.000000         NA          NA
sesunknown -0.11383577  0.892405 0.06970793 1.02461e-01
sesupper   -0.03289457  0.967641 0.17289134 8.49104e-01
seslower   -0.05118488  0.950103 0.05606762 3.61288e-01

t(tab2)
                          [,1]
Events               1.65700e+03
Total.time.at.risk   4.50046e+03
Max.log.likelihood -3.17024e+03
LR.test.statistic    3.92070e+01
Degrees.of.freedom   5.00000e+00
Overall.p.value      2.15744e-07
```

The Piece-Wise Constant Hazards Model

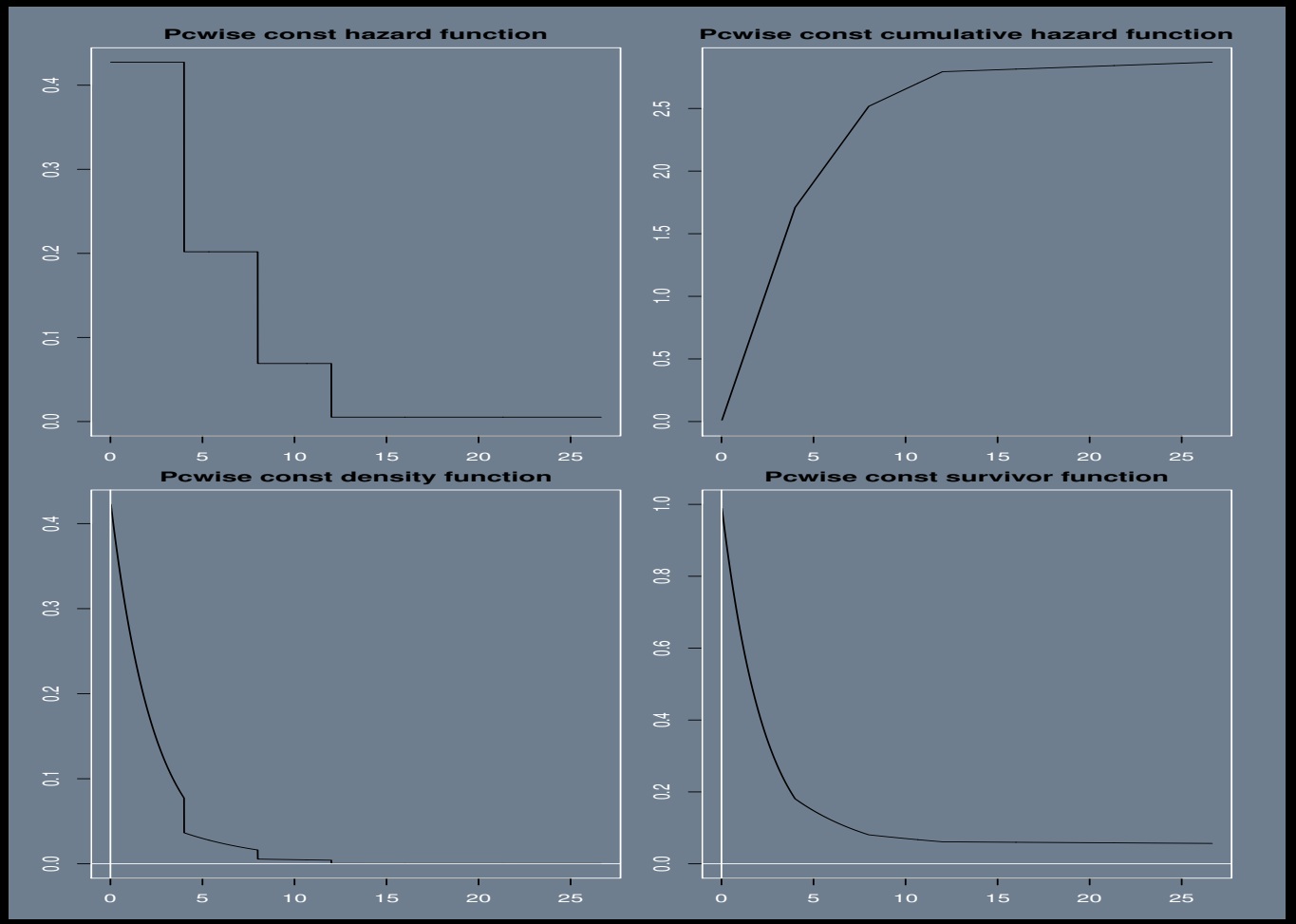
- The standard LRT approach gives only AIC values for the alternative models:

```
drop1(fit.p1)
 Single term deletions

Model:
Surv(next.ivl, event) ~ age + year + ses
       Df  AIC
<none>    6350
age     1 6381
year    1 6349
ses     3 6347
```

- The plots are created by:

```
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.p1)
```

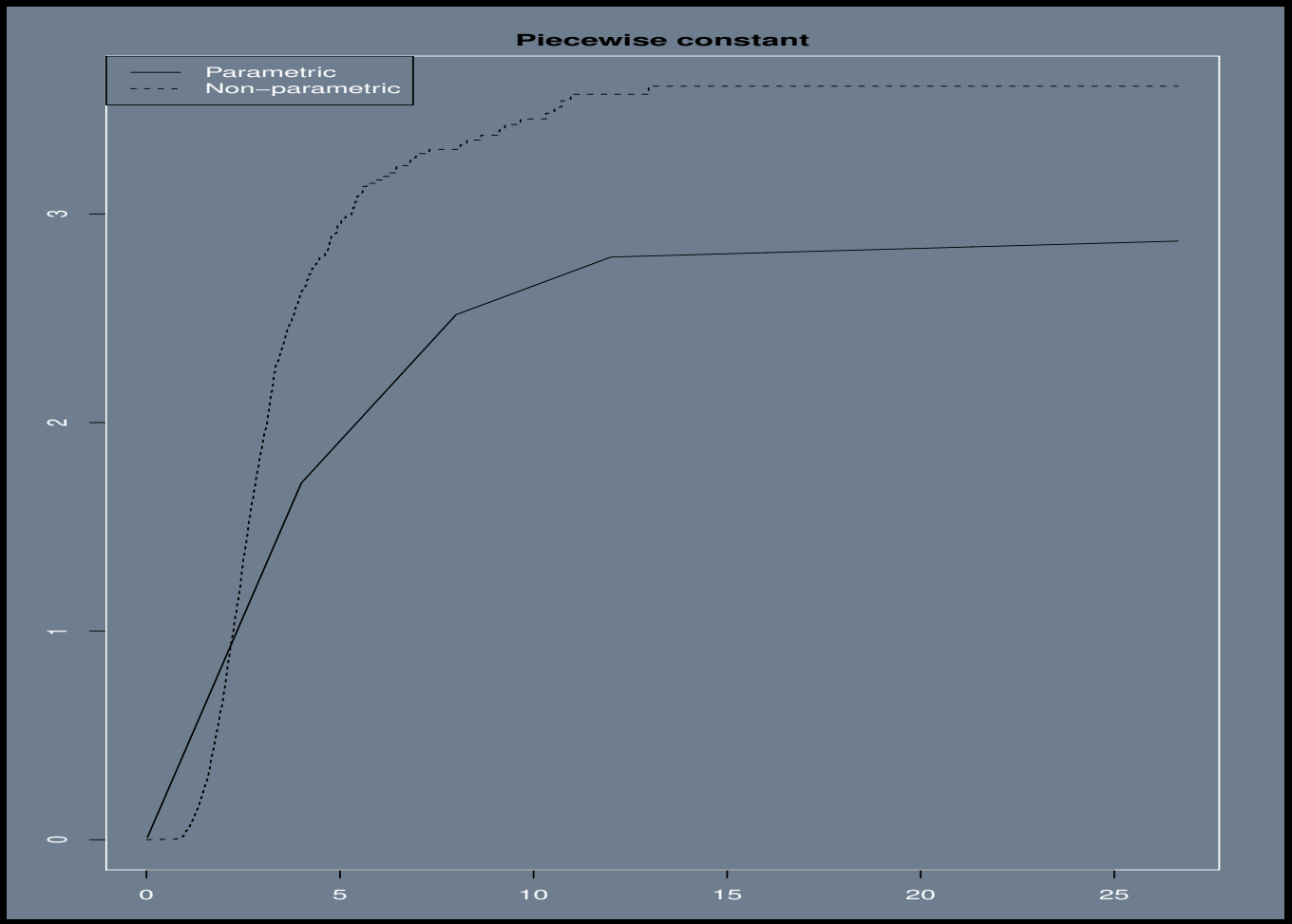
Piece-Wise Constant Hazards Model of the Fertility Data



Compare Piece-Wise Constant Hazard To Cox Model

```
fit.p1 <- phreg(Surv(next.ivl,event) ~ age + year + ses, data=fert1,
    dist="pch", cuts=c(4,8,12))
fit.p1.cox <- coxreg(Surv(next.ivl,event) ~ age + year + ses, data=fert1)

postscript("Class.Survival/Images/pch4.fert.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
check.dist(fit.p1.cox,fit.p1)
dev.off()
```

Compare Piece-Wise Constant Hazard To Cox Model



Piece-Wise Constant Hazards Model of the Fertility Data

- The Broström book notes dissatisfaction with the cutpoints stipulated so far particularly in the early phase, so cut every year with `cuts(1:13)`.
- This leads to the following model:

```
fit.p2 <- phreg(Surv(next.ivl,event) ~ age + year + ses, data=fert1,
    dist="pch", cuts=1:13)
summary(fit.p2)
```

```
 Error in substr(covar.names[covar.no], 16) :
argument "stop" is missing, with no default
```

- So once again. . . but let's make it a function.

Piece-Wise Constant Hazards Model of the Fertility Data

```
pch.sum <- function(in.list,null.loglik)  {
    tab1 <- cbind( in.list$coefficients, exp(in.list$coefficients),
        sqrt(diag(in.list$var)),
        2*(1-pnorm(abs(in.list$coefficients)/sqrt(diag(in.list$var)))) )
    tab1 <- rbind(tab1[1:2,],c(0,1,NA,NA),tab1[3:5,])
    dimnames(tab1)[[1]][3] <- "farmer"
    dimnames(tab1)[[2]] <- c("Coef","Exp(Coef)","se(Coef)","Wald p")
    tab2 <- data.frame(
        "Events"=in.list$events,
    "Total time at risk"=sum(in.list$y[,2]),
    "Max log likelihood"=round(in.list$loglik[2],3),
    "LR test statistic"=round(2*(in.list$loglik[2]-null.loglik),3),
    "Degrees of freedom"=in.list$df,
    "Overall p-value"=pchisq(2*(in.list$loglik-null.loglik),
        df=in.list$df,lower.tail=FALSE)[2] )
    return(list(tab1,t(tab2[1,])))
}
```

Piece-Wise Constant Hazards Model of the Fertility Data

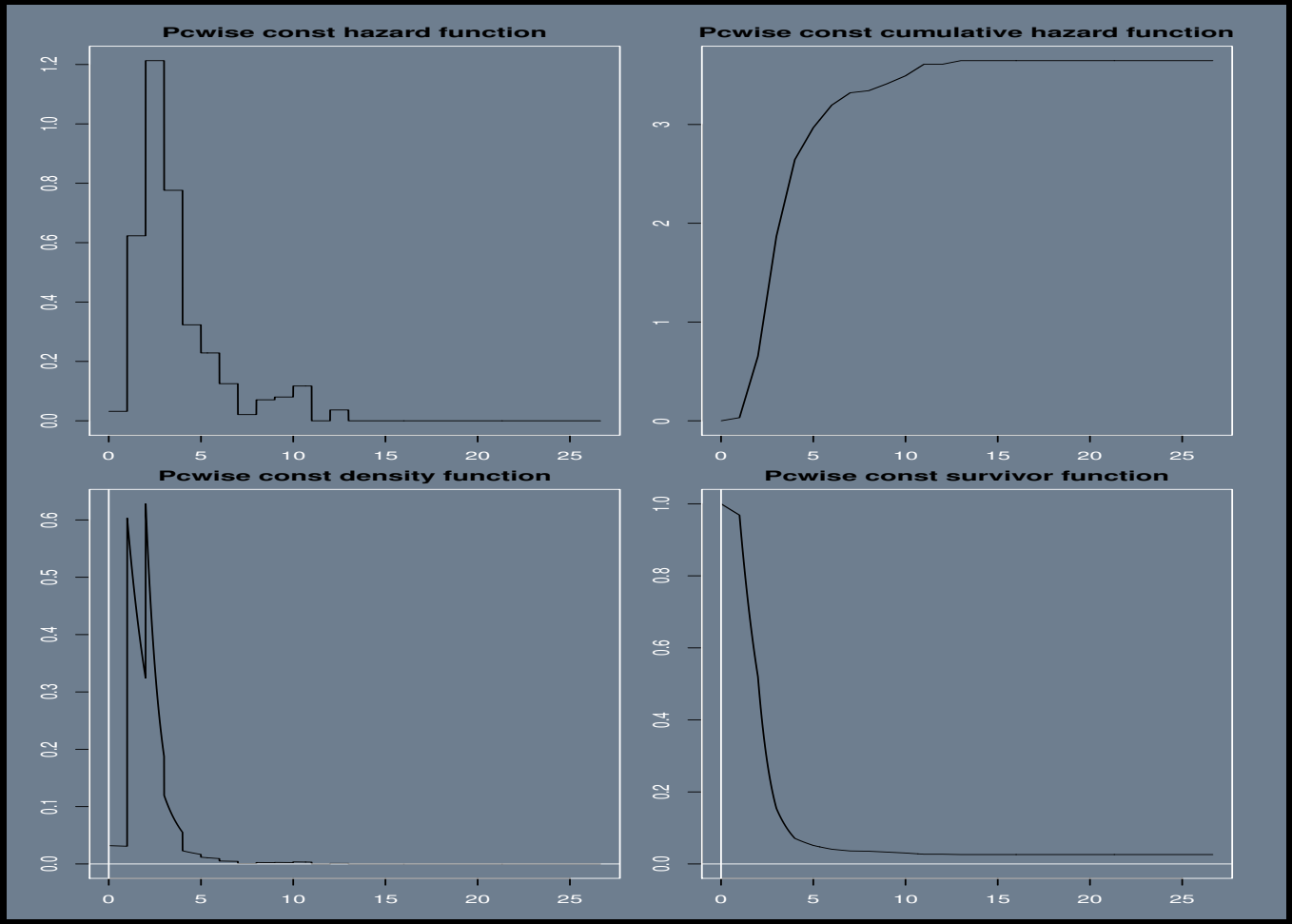
```
fit.null <- phreg(Surv(next.ivl,event) ~ 1, data=fert1,
    dist="pch", cuts=1:13)
pch.sum(fit.p2,fit.null$loglik)
```

```
                  Coef Exp(Coef)   se(Coef)      Wald p
age         -0.03994534  0.960842 0.00542533 1.80078e-13
year         0.00190674  1.001909 0.00213636 3.72115e-01
farmer       0.00000000  1.000000         NA          NA
sesunknown -0.11278210  0.893345 0.06973389 1.05809e-01
sesupper   -0.00576577  0.994251 0.17295751 9.73406e-01
seslower   -0.06993455  0.932455 0.05604554 2.12098e-01

Events              1.65700e+03
Total.time.at.risk  4.50046e+03
Max.log.likelihood -2.30629e+03
LR.test.statistic   6.78720e+01
Degrees.of.freedom  5.00000e+00
Overall.p.value     2.83982e-13
```

Piece-Wise Constant Hazards Model Plots

```
postscript("Class.Survival/Images/pch2.fert.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.p2)
dev.off()
```

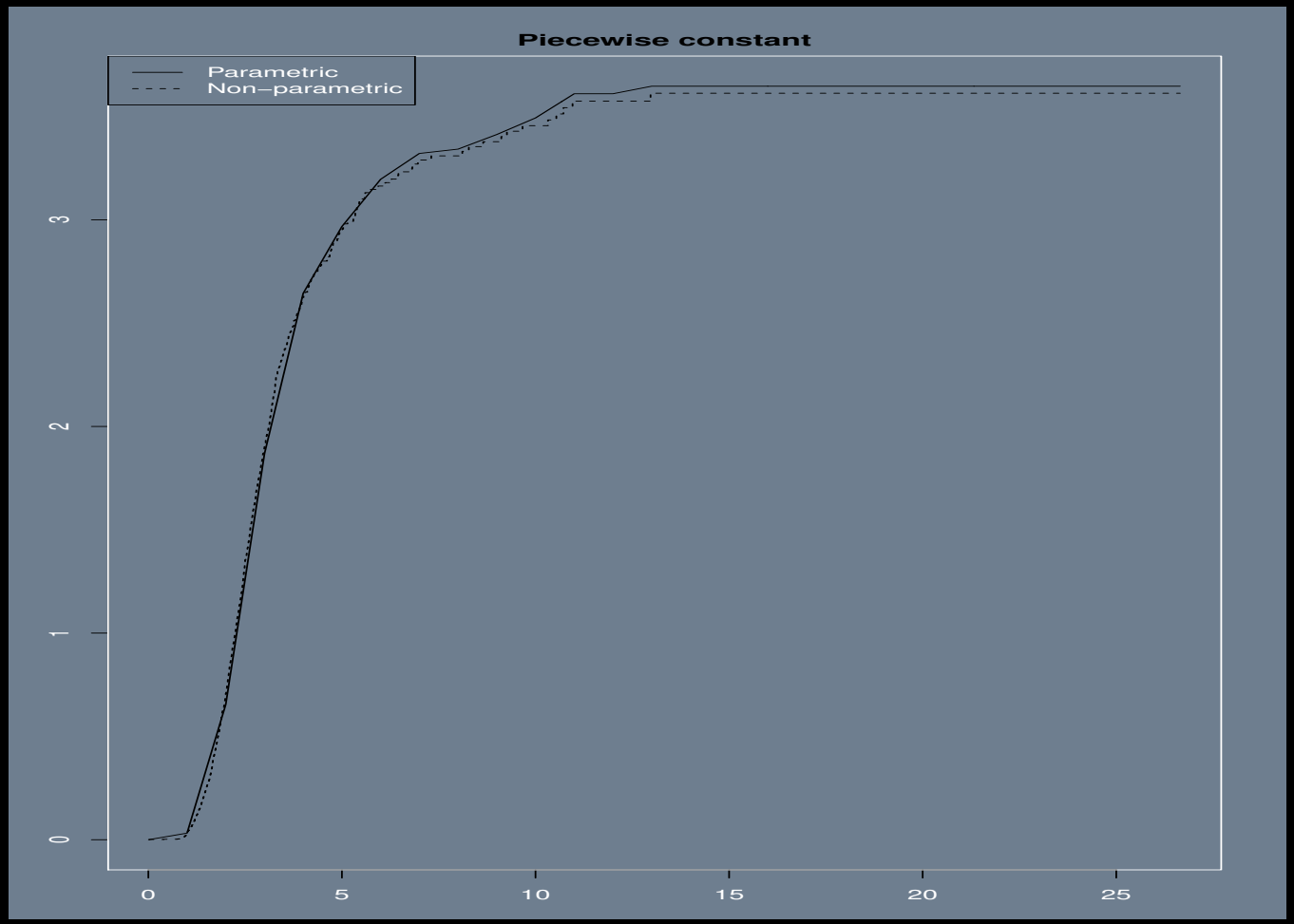
Piece-Wise Constant Hazards Model Plots



Compare Piece-Wise Constant Hazard To Cox Model

```
fit.p2 <- phreg(Surv(next.ivl,event) ~ age + year + ses, data=fert1,
    dist="pch", cuts=1:13)
fit.p2.cox <- coxreg(Surv(next.ivl,event) ~ age + year + ses, data=fert1)

postscript("Class.Survival/Images/pch3.fert.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
check.dist(fit.p2.cox,fit.p2)
dev.off()
```

Compare Piece-Wise Constant Hazard To Cox Model



Supreme Court Retirement Model

- In Section 8.5 BSJ introduce the example of retirements from the SCOTUS.
- Explanatory variables:
  - Republican Justice: 1 if Republican, 0 if Democrat.
  - Partisan Agreement: 1 for whether the justice is of the same party as his/her appointing president, 0 otherwise???
  - Age: chronological age.
  - Chief Justice: 1 if chief justice, 0 otherwise.
  - Southern Justice: 1 if from the South, 0 otherwise.
  - 20th Century Appointment: 1 if from the South, 0 otherwise.
  - Critical Nomination: 1 if their appointment causes the partisan balance to shift, or if their appointment moves the ideological balance to move from 6-3 to 5-4 or 5-4 to 6-3, 0 otherwise.

Supreme Court Retirement Model

Table 8.1: Cox and Piece-Wise Cox Models of Supreme Court Retirements

	Cox	Piece-Wise Divided at Median ($T = 14$)	
Republican Justice	0.56 (0.33)	0.56 (0.50)	0.29 (0.46)
Partisan Agreement	0.39 (0.47)	1.11 (1.08)	0.17 (0.57)
Age	0.08 (0.03)	0.02 (0.04)	0.12 (0.05)
Chief Justice	-0.40 (0.47)	-0.26 (0.63)	-0.31 (0.81)
Southern Justice	0.08 (0.37)	1.19 (0.53)	-0.25 (0.56)
20th Century Justice	0.53 (0.32)	0.70 (0.45)	0.74 (0.50)
Critical Nomination	-0.58 (0.40)	0.56 (0.81)	-0.43 (0.50)
N	107	52	55
Log-Likelihood	-181.71	-73.16	-75.72
χ^2	26.87 ($p < 0.001$)	7.72 (n.s)	26.35 ($p < 0.001$)

Supreme Court Retirement Model

Table 8.2: Stratified Cox Models of Supreme Court Retirement

	Age	Southern Justice	Critical Nomination
Republican Justice	0.51 (0.32)	0.73 (0.36)	0.60 (0.32)
Partisan Agreement	0.33 (0.48)	0.55 (0.50)	0.33 (0.49)
Age	n/a	0.08 (0.03)	0.08 (0.03)
Chief Justice	-0.22 (0.46)	-0.24 (0.48)	0.40 (0.47)
Southern Justice	0.13 (0.38)	n/a	0.11 (0.37)
20th Century Justice	0.75 (0.32)	0.49 (0.32)	0.48 (0.33)
Critical Nomination	-0.69 (0.41)	-0.54 (0.40)	n/a
N	107	107	107
Log-Likelihood	-152.94	-151.95	-161.90
χ^2	12.11 ($p = 0.06$)	26.16 ($p < 0.001$)	23.68 ($p < 0.001$)

Assessing Proportional Hazards for the Cox Model

- A simple way to test for non-PH is to graph the Schoenfeld Residuals against survival times and look for non-linearity.
- The Schoenfeld residual for the ith individual and the jth explanatory variable is:

$$r_{S_{ji}} = \delta_i(x_{ji} - \hat{a}_{ji}),$$

where:

$$\hat{a}_{ji} = \frac{\sum_{\ell \in R(t_i)} \exp(\mathbf{x}_{j\ell}\hat{\boldsymbol{\beta}})}{\sum_{\ell \in R(t_i)} \exp(\mathbf{x}_{\ell}\hat{\boldsymbol{\beta}})}$$

and $\ell \in R(t_j)$ is the risk-set at time t_i.

- The Grambsch & Therneau global proportionality test takes the maximum of the absolute cumulative summed Schoenfeld residuals to build a test statistic where being in the tail implies non-PH for the overall model.
- The Harrell's rho test uses the correlation between the Schoenfeld residuals for each covariate and the survival times, and has the same χ^2 interpretation.

Assessing Proportional Hazards for the Cox Model

Table 8.3: Nonproportionality Tests of Supreme Court Retirements

	Estimates ρ	χ^2 Statistic	p-value
Harrell's ρ			
Republican Justice	-0.213	2.224	0.13
Partisan Agreement	-1.117	0.71	0.40
Age	0.195	2.16	0.14
Chief Justice	-0.064	0.23	0.63
Southern Justice	-0.281	4.86	0.03
20th Century Justice	-0.009	0.01	0.94
Critical Nomination	0.142	1.01	0.31
G&T Proportionality Global Test	–	153.98	0.001

The Accelerated Failure Time Model

- The accelerated failure time model asserts that a treatment *accelerates* time to failure by some factor ϕ.
- This can be "good" if $\phi < 1$.
- Or it can hasten demise if $\phi > 1$.
- Positive values of ϕ shift the hazard function to the left for the treatment, meaning that the treatment causes more early events.
- If this is the correct specification, the hazards tend to coalesce at later times.
- See Figure 6.16 in the Broström book on page 113.

The Accelerated Failure Time Model

- Consider two groups with two survivor functions:
  - ▷ Control: $p(T \geq t) = S_0(t)$
  - ▷ Control: $p(T \geq t) = S_0(\phi t)$
  - ▷ $-\infty < \phi < \infty$

  meaning that the treatment accelerates failure time by ϕ.
- Values of $\phi > 0$ imply longer times to death, and vice-versa.
- As the name implies this model is useful when deaths occur rapidly at the beginning of the period and alternative models fail to capture the effect (see illustrations in the Broström book).

The Accelerated Failure Time Model

- State that T has standard survivor function $S(t) = p(T \geq t)$.
- Also that $T_c = T/c$, so $S(t_c) = p(T/c \geq t) = p(T \geq tc) = S(tc)$.
- Now stipulate that $Y = \log(T)$ and $Y_c - \log(T_c)$, so that:

$$Y_c = Y - \log(c).$$

where this is a linear model with substitutions:

$$(Y = Y_c*) = (\epsilon = Y) - (\log(c) = -\mathbf{x}\boldsymbol{\beta})$$
$$Y = \mathbf{x}\boldsymbol{\beta} + \epsilon$$

- Obviously this is a linear model but right censoring and left truncation preclude OLS estimation.
- The `survreg` function in the `survival` package runs AFT but not for left censoring or time-varying covariates.
- The author's `aftreg` function in `eha`.

The Accelerated Failure Time Model, Old Age Mortality

- The following model fits an AFT model with a Weibull distribution:

```
library(eha)
data(oldmort)
oldmort$Y <- Surv(oldmort$enter - 60, oldmort$exit - 60, oldmort$event)
fit.aft1 <- aftreg(Y ~ sex + civ + birthplace, data=oldmort)
fit.aft1

Covariate           W.mean      Coef Time-Accn  se(Coef)    Wald p
sex
             male    0.406     0          1           (reference)
           female    0.594    -0.147      0.863     0.028     0.000
civ
        unmarried    0.080     0          1           (reference)
          married    0.530    -0.270      0.764     0.049     0.000
            widow    0.390    -0.124      0.884     0.046     0.008
birthplace
           parish    0.574     0          1           (reference)
           region    0.226     0.032      1.033     0.033     0.321
           remote    0.200     0.036      1.037     0.035     0.299
```

The Accelerated Failure Time Model, Old Age Mortality

- continued...

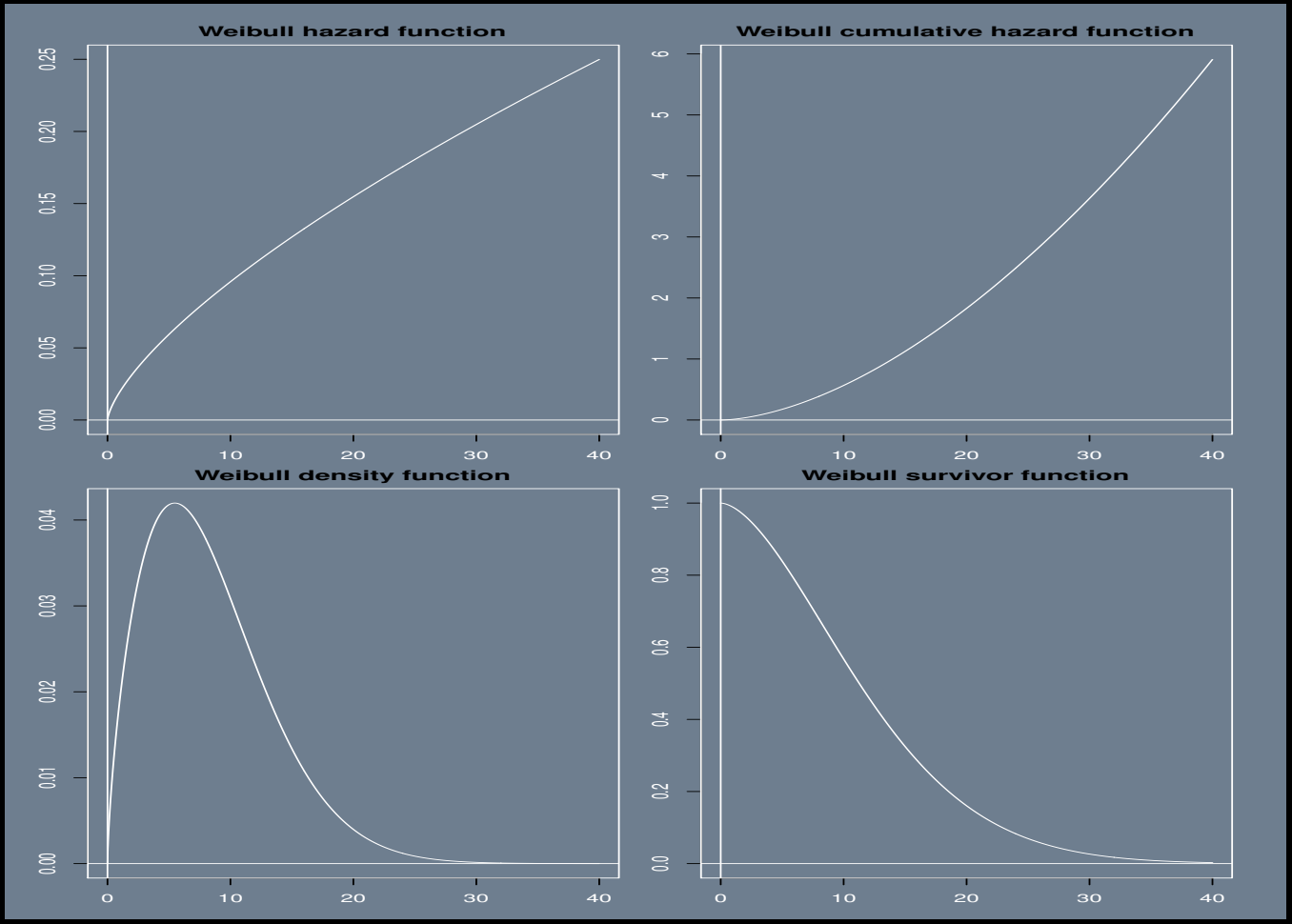
```
Baseline parameters:
log(scale)                  2.639    13.993     0.048     0.000
log(shape)                  0.526     1.691     0.019     0.000
Baseline life expectancy:

Events                    1971
Total time at risk        37824
Max. log. likelihood      -7410
LR test statistic         55.1
Degrees of freedom        5
Overall p-value           1.24228e-10
```

- This is almost the same as the regular Weibull model, except that the parameters are on a different scale.
- The log likelihood is also different.

Accelerated Failure Time Model, Plots

```
postscript("Class.Survival/Images/aft1.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.aft1)
dev.off()
```

Accelerated Failure Time Model, Plots



Gompertz Distribution for Failure Times

- Another distribution that can be used for parametric survival models is the Gompertz distribution.
- The PDF of the lifetime distribution for the random variable T is:

$$f(T = t|\alpha, \beta) = \alpha \exp[\beta t] \exp\left[-\left(\frac{\alpha}{\beta}\exp(\beta t) - \frac{\alpha}{\beta}\right)\right].$$

- The CDF of T is:

$$F(T < t|\alpha, \beta) = 1 - \exp\left[-\left(\frac{\alpha}{\beta}\exp(\beta t) - \frac{\alpha}{\beta}\right)\right].$$

- The survival function is:

$$S(t|\alpha, \beta) = \exp\left[-\left(\frac{\alpha}{\beta}\exp(\beta t) - \frac{\alpha}{\beta}\right)\right].$$

- The cumulative hazard function:

$$H(t|\alpha, \beta) = -\left(\frac{\alpha}{\beta}\exp(\beta t) - \frac{\alpha}{\beta}\right).$$

The Accelerated Failure Time Model, Old Age Mortality

- We can also run it with the Gompertz distribution instead of the Weibull:

```
fit.aft2 <- aftreg(Y ~ sex + civ + birthplace, dist="gompertz", data=oldmort)
fit.aft2
```

```
Covariate          W.mean      Coef Time-Accn  se(Coef)    Wald p
sex
            male    0.406     0         1           (reference)
          female    0.594    -0.081     0.922     0.020     0.000
civ
       unmarried    0.080     0         1           (reference)
         married    0.530    -0.152     0.859     0.034     0.000
           widow    0.390    -0.100     0.905     0.031     0.001
birthplace
          parish    0.574     0         1           (reference)
          region    0.226     0.022     1.022     0.023     0.340
          remote    0.200     0.038     1.039     0.025     0.132
```

The Accelerated Failure Time Model, Old Age Mortality

- continued...

```
Baseline parameters:
log(scale)                  2.222     9.224     0.042     0.000
log(shape)                 -1.584     0.205     0.075     0.000
Baseline life expectancy:  13.6

Events                    1971
Total time at risk         37824
Max. log. likelihood       -7280
LR test statistic         33
Degrees of freedom        5
Overall p-value           3.81949e-06
```

- This is only a slightly different fit here.

Comparing the Proportional Hazards Model to the Gompertz AFT Model

- Recall that the AIC for model comparison is $AIC = -2\ell() + 2p$.
- Since these two models have the same $2p$ this comes down to comparing just the likelihood functions.
- Actually, since these are not nested specifications, this is an approximation at best.
- That is, $LRT = 2\log\frac{L_\ell}{L_s} = -2\log\frac{L_s}{L_\ell}$ isn't appropriate here.
- Nonetheless, since Broström does it (page 116):

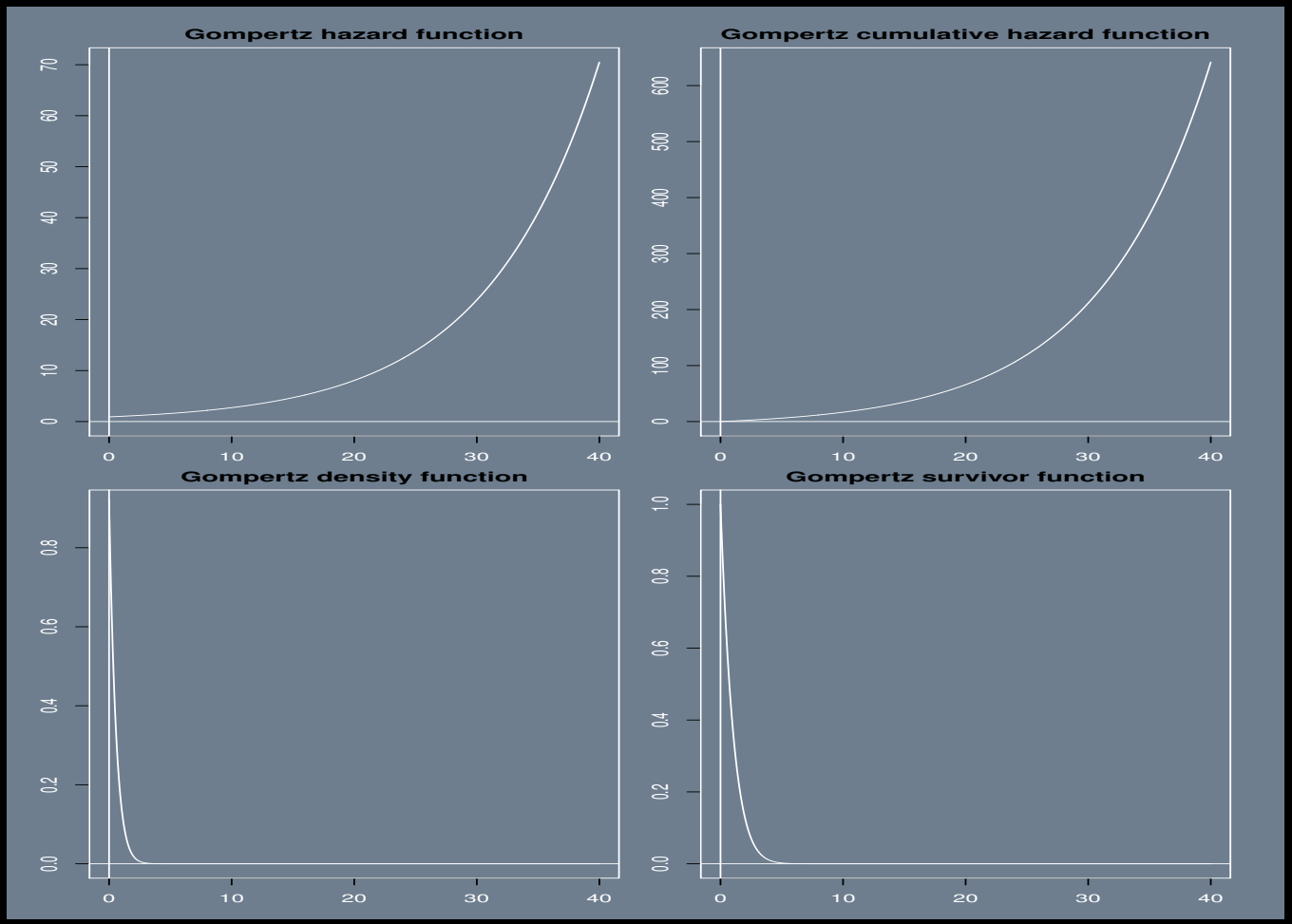
Regular Gompertz	-7273.701
AFT Gompertz	-7279.973

so we *might* prefer the regular Gompertz proportional hazards mode.

Accelerated Failure Time Model, Plots

```
postscript("Class.Survival/Images/aft2.ps")
par(oma=c(1,1,1,1),mar=c(2,2,2,1),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.aft2)
dev.off()
```

Accelerated Failure Time Model, Plots



Discrete Time Parametric Models

- In demography *register* data is frequently used and always discrete.
- So far we have only worked with *long* data in which there is one record (line) in the data for each person at each time (from `survSplit`).
- There is also *wide* data in which each individual gets only one line but with multiple time values recorded like traditional panel data.
- The `R` function `reshape` reshapes a data frame between wide format with repeated measurements in separate columns of the same record and long format with the repeated measurements in separate records.
- In the case of wide data time-varying variables of the same measure must end in a numeric value tying the variable to the time measured.
- Broström book example, the variable `civ` over three periods on the same record (line) would be `civ.1`, `civ.2`, and `civ.3`, where the "." is the default separator.

Discrete Time Parametric Example

- Reformat the old age mortality dataset into long format according to:

```
data(oldmort)
om <- oldmort[oldmort$enter == 60,] # RESTRICT ENTRY TO AGE 60
table(om$enter)
  60
  3223
om <- age.window(om, c(60,70))  # CUT THE AGE AT TWO POINTS

om$m.id <- om$f.id <- om$imr.birth <- om$birthplace <- NULL
om$birthdate <- om$ses.50 <- NULL # REMOVE UNUSED COLUMNS

# NOW MOVE TO counting process FORMAT
om1 <- survSplit(om, cut=61:69, start="enter", end="exit",
       event="event", episode="agegrp")
om1[1:3,]
         id enter exit event    sex     civ   region agegrp
1 800000625    60   61     0   male   widow    rural      0
2 800000631    60   61     0 female   widow industry      0
3 800000633    60   61     0   male married     town      0
```

Discrete Time Parametric Example

```
# FACTOR AND RELIABLE AGE GROUPS CREATED
table(om1$agegrp)
   0    1    2    3    4    5    6    7    8    9
3223 2823 2426 2075 1760 1453 1203 1011  803  657
om1$agegrp <- factor(om1$agegrp, labels=60:69)
table(om1$agegrp)
   60   61   62   63   64   65   66   67   68   69
3223 2823 2426 2075 1760 1453 1203 1011  803  657
```

Discrete Time Parametric Example

```
# SORT BY IDENTIFICATION THEN ENTER
om1 <- om1[order(om1$id, om1$enter),]
om1[1:10,]
```

	id	enter	exit	event	sex	civ	region	agegrp
1	800000625	60	61.000	0	male	widow	rural	0
3224	800000625	61	62.000	0	male	widow	rural	1
6447	800000625	62	63.000	0	male	widow	rural	2
9670	800000625	63	63.413	0	male	widow	rural	3
2	800000631	60	61.000	0	female	widow	industry	0
3225	800000631	61	62.000	0	female	widow	industry	1
6448	800000631	62	63.000	0	female	widow	industry	2
9671	800000631	63	64.000	0	female	widow	industry	3
12894	800000631	64	65.000	0	female	widow	industry	4
16117	800000631	65	66.000	0	female	widow	industry	5

Discrete Time Parametric Example

```
# MAKE THESE ID NUMBERS EASIER TO READ AND FACTORS
rownames(om1) <- 1:nrow(om1)
om1[1:10,]
          id enter   exit event    sex   civ   region agegrp
1  800000625    60 61.000     0   male widow    rural      0
2  800000625    61 62.000     0   male widow    rural      1
3  800000625    62 63.000     0   male widow    rural      2
4  800000625    63 63.413     0   male widow    rural      3
5  800000631    60 61.000     0 female widow industry      0
6  800000631    61 62.000     0 female widow industry      1
7  800000631    62 63.000     0 female widow industry      2
8  800000631    63 64.000     0 female widow industry      3
9  800000631    64 65.000     0 female widow industry      4
10 800000631    65 66.000     0 female widow industry      5
```

Discrete Time Parametric Example

```
om1$id <- as.numeric(as.factor(om1$id))
om1[1:10,]
   id enter   exit event    sex   civ birthplace   region agegrp
1   1    60 61.000     0   male widow     parish    rural     60
2   1    61 62.000     0   male widow     parish    rural     61
3   1    62 63.000     0   male widow     parish    rural     62
4   1    63 63.413     0   male widow     parish    rural     63
5   2    60 61.000     0 female widow     region industry     60
6   2    61 62.000     0 female widow     region industry     61
7   2    62 63.000     0 female widow     region industry     62
8   2    63 64.000     0 female widow     region industry     63
9   2    64 65.000     0 female widow     region industry     64
10  2    65 66.000     0 female widow     region industry     65
```

Discrete Time Parametric Example

- So each individual has as many records (lines) as their "presented ages."
- The first person has four record since they entered at age 60 and exited at age 63.413.
- The maximum number of presented ages is 10 by construction.
- We can check the distribution of these presented ages by:

```
table( tapply(om1$id, om1$id, length) )
  1   2   3   4   5   6   7   8   9  10
400 397 351 315 307 250 192 208 146 657
```

noticing the large number that survive until the last period.

Discrete Time Parametric Example

- Now turn the long format into the wide format, first getting rid of redundant variables `enter` and `exit` to make the new dataset easier to read.
- We will make wide the explanatory variables `event`, `civ`, `region`.
- We need to tell `reshape` which is the identification column, which is the time column, and that we want "wide."
- This is done by:

```
om1$exit <- om1$enter <- NULL
om2 <- reshape(om1, v.names=c("event", "civ", "region"),
        idvar="id", direction="wide", timevar="agegrp")
```

Discrete Time Parametric Example

- Looking at the results:

```
names(om2)
 [1] "id"       "sex"      "event.0"  "civ.0"    "region.0" "event.1"  "civ.1"
 [8] "region.1" "event.2"  "civ.2"    "region.2" "event.3"  "civ.3"    "region.3"
[15] "event.4"  "civ.4"    "region.4" "event.5"  "civ.5"    "region.5" "event.6"
[22] "civ.6"    "region.6" "event.7"  "civ.7"    "region.7" "event.8"  "civ.8"
[29] "region.8" "event.9"  "civ.9"    "region.9"
```

```
om2[1,]
        id  sex event.0 civ.0 region.0 event.1 civ.1 region.1 event.2 civ.2 region.2
1 800000625 male       0 widow    rural       0 widow    rural       0 widow    rural
  event.3 civ.3 region.3 event.4 civ.4 region.4 event.5 civ.5 region.5 event.6 civ.6
1       0 widow    rural      NA  <NA>     <NA>      NA  <NA>     <NA>      NA  <NA>
  region.6 event.7 civ.7 region.7 event.8 civ.8 region.8 event.9 civ.9 region.9
1     <NA>      NA  <NA>     <NA>      NA  <NA>     <NA>      NA  <NA>     <NA>
```

Discrete Time Parametric Example

- What about going the other way back to long format?

```
om3 <- reshape(om2, direction="long", idvar="id", varying=3:32)
om3[1:10,]
                  id    sex time event       civ   region
800000625.0 800000625   male    0     0     widow    rural
800000631.0 800000631 female    0     0     widow industry
800000633.0 800000633   male    0     0   married     town
800000641.0 800000641   male    0     0   married     town
800000644.0 800000644 female    0     0   married     town
800000645.0 800000645 female    0     0   married     town
800000652.0 800000652 female    0     0 unmarried    rural
800000663.0 800000663   male    0     0   married     town
800000686.0 800000686 female    0     0   married     town
800000691.0 800000691 female    0     0 unmarried industry
```

Discrete Time Parametric Example

- Now let's sort by `id` then `time`:

```
om3 <- om3[order(om3$id, om3$time),]
om3$id <- as.numeric(as.factor(om3$id))
dimnames(om3)[[1]] <- 1:nrow(om3)
om3[1:10,]
   id  sex time event   civ region
1   1 male    0     0 widow  rural
2   1 male    1     0 widow  rural
3   1 male    2     0 widow  rural
4   1 male    3     0 widow  rural
5   1 male    4    NA  <NA>   <NA>
6   1 male    5    NA  <NA>   <NA>
7   1 male    6    NA  <NA>   <NA>
8   1 male    7    NA  <NA>   <NA>
9   1 male    8    NA  <NA>   <NA>
10  1 male    9    NA  <NA>   <NA>
```

- Where all individuals get 10 records even if they didn't live that long like case 1.

Discrete Time Parametric Example

- So let's remove the cases (rows) that occur after the event.
- This is done by looking at the `NA` values:

```
nrow(om3)
[1] 32230
om3 <- om3[!is.na(om3$event),]
nrow(om3)
[1] 17434
```

- Revisiting `time`, we see that:

```
summary(om3$time)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    1.00    3.00    3.15    5.00    9.00
```

but this needs to be a factor to get a set of hazards by time, including the baseline:

```
om3$time <- as.factor(om3$time)
table(om3$time)
   0    1    2    3    4    5    6    7    8    9
3223 2823 2426 2075 1760 1453 1203 1011  803  657
```

Discrete Time Parametric Example

- All of this sets up a binomial model for each time period since there is a number at risk in each of the time periods and a subset of them die during this time period.
- Recall that with binomial models in R there are three typical choices of link function, but only `cloglog` preserves the proportional hazards assumption.
- Therefore run the model:

```
fit.clog <- glm(event ~ sex + civ + region + as.factor(time),
    family=binomial(link=cloglog), data=om3)
```

Discrete Time Parametric Example

- Summarizing:

```
summary(fit.clog)
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.426  -0.243  -0.218  -0.194   2.950

   Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -3.4213     0.2117  -16.16  < 2e-16
sexfemale        -0.3633     0.0993   -3.66  0.00025
civmarried       -0.3368     0.1560   -2.16  0.03085
civwidow         -0.1901     0.1681   -1.13  0.25817
regionindustry    0.1078     0.1444    0.75  0.45526
regionrural      -0.2242     0.1403   -1.60  0.11008
as.factor(time)1  0.0490     0.1851    0.26  0.79137
as.factor(time)2  0.4600     0.1740    2.64  0.00819
as.factor(time)3  0.0559     0.2020    0.28  0.78213
as.factor(time)4  0.4632     0.1888    2.45  0.01416
as.factor(time)5  0.3175     0.2085    1.52  0.12782
as.factor(time)6  0.4558     0.2123    2.15  0.03175
as.factor(time)7  0.9151     0.1955    4.68  2.9e-06
as.factor(time)8  0.6855     0.2258    3.04  0.00239
as.factor(time)9  0.6054     0.2490    2.43  0.01506

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4186.8  on 17433  degrees of freedom
Residual deviance: 4125.2  on 17419  degrees of freedom
AIC: 4155
```

Discrete Time Parametric Example

- Notice the uneven reliability of year coefficients.
- We can assess the value of the covariates' contributions again with the `drop1` function:

```
drop1(fit.clog, test="Chisq")
Single term deletions

Model:
event ~ sex + civ + region + as.factor(time)
                Df Deviance  AIC   LRT Pr(>Chi)
<none>               4125 4155
sex              1     4139 4167 13.30 0.000265
civ              2     4130 4156  4.98 0.082996
region           2     4136 4162 10.53 0.005181
as.factor(time)  9     4161 4173 36.01 3.96e-05
```

  where we see that `civ` is somewhat disappointing (don't use "not that statistically significant").

- The `eha` package also provides a function `glmmboot` that fits grouped GLMs with fixed group effects and supplies bootstrapped standard errors for testing.

Discrete Time Parametric Example

- Run this routine:

```
fit.clog2 <- glmmboot(event ~ sex + civ + region, cluster=time,
    family=binomial(link=cloglog), data=om3)
summary(fit.clog2)
                 coef se(coef)      z Pr(>|z|)
sexfemale      -0.363   0.0993 -3.659  0.00025
civmarried     -0.337   0.1560 -2.158  0.03100
civwidow       -0.190   0.1681 -1.131  0.26000
regionindustry  0.108   0.1445  0.746  0.46000
regionrural    -0.224   0.1404 -1.597  0.11000

Residual deviance: 4270 on 17419 degrees of freedom    AIC: 4300
```

- Notice that time is not in this summary; we need to ask for it separately to get the baseline hazards for the 10 periods:

```
plogis(fit.clog2$frail)
[1] 0.0316355 0.0331700 0.0492060 0.0333923 0.0493552 0.0429493
[7] 0.0490086 0.0754235 0.0608914 0.0564689
```

Discrete Time Parametric Example

- It is interesting, but hardly essential, to compare to the Cox proportional hazards model for the same conditioned dataset (remember that we did lots of processing here).
- This involves switching back to intervals:

```
om3$exit <- as.numeric(as.character(om3$time))
om3$enter <- om3$exit - 0.5
fit.cox <- coxreg(Surv(enter, exit, event) ~ sex + civ + region,
    method="ml", data=om3)
```

Discrete Time Parametric Example

- The results are given by: `summary(fit.cox)`:

```
Covariate           Mean       Coef     Rel.Risk   S.E.    Wald p
sex         male    0.404      0        1 (reference)
          female    0.596     -0.363    0.695     0.099    0.000
civ    unmarried    0.090      0        1 (reference)
         married    0.653     -0.337    0.714     0.156    0.031
           widow    0.257     -0.190    0.827     0.168    0.258
region      town    0.143      0        1 (reference)
        industry    0.307      0.108    1.114     0.144    0.455
           rural    0.551     -0.224    0.799     0.140    0.110

Events                    451
Total time at risk          8717
Max. log. likelihood     -2062.6
LR test statistic         27.23
Degrees of freedom        5
Overall p-value           5.14285e-05
```

which looks a lot like previous results.

Discrete Time Parametric Example

- Of course we like to look at the results graphically:

```
postscript("Class.Survival/Images/final.cox.ps",width=7.2,height=5.2)
par(oma=c(1,1,1,1),mar=c(2,2,2,1),mfrow=c(1,2),col.axis="white",
    col.lab="white",col.sub="white",col="white", bg="slategray")
plot(fit.cox,fn="cum")
plot(fit.cox,fn="surv")
dev.off()
```

Discrete Time Parametric Example Plot