Contents

What is survival analysis?

- Survival analysis[1] is a class of statistical methods for studying the occurrence and timing of **events**, e.g., death, onset of disease, return of a tumor, equipment failures, customer churn/purchase, earthquakes, stock market crashes, recessions, revolutions, job terminations, divorces, promotions, retirements, re-arrests after parole, leaving a web site
- A.K.A. history analysis (sociology), reliability analysis (engineering), failure-time analysis (engineering), duration analysis (economics), transition analysis (economics)
- Survival analysis can model data with two features that are difficult to handle with conventional methods:
  - **censoring**: we may not observe the "event" for all observations
  - **time-dependent covariates**: predictor variables may change during the study

  All survival analysis methods allow for censoring. Many allow for time-dependent covariates

[1] Here's a short list of references:

- Allison, Paul (1995), *Survival Analysis using the SAS System*, SAS Institute (newer 2nd edition too)
- Cox, D.R. and Oakes, D. (1984), *Analysis of Survival Data*, Chapman and Hall
- Crowder, M.J., Kimber, A.C., Smith, R.L., and Sweeting, T.J. (1991), *Statistical Analysis of Reliability Data*, Chapman and Hall
- Kalbfleisch, J.D. and Prentice, R.L. (1980), *The Statistical Analysis of Failure Time Data*, Wiley

Survival Analysis Terms

- T random variable giving the time of the event
- *Cumulative distribution function*: $F(t) = \mathsf{P}(T \leq t)$
- *Surviver function*[2]: $S(t) = \mathsf{P}(T > t) = 1 - F(t)$
- *Probability density function* (PDF) is

$$f(t) = \frac{dF(t)}{dt} = -\frac{dS(t)}{dt}$$

- *Hazard function*:

$$h(t) = \lim_{\Delta t \to 0} \frac{\mathsf{P}(t \leq T < t + \Delta t | T > t)}{\Delta t} = \frac{f(t)}{S(t)}$$

  Indicates the "proneness to failure" in the instant after time t or "instantaneous risk that the event will occur"

- The *cumulative hazard function* is

$$H(t) = \int_0^t h(u)\, du = -\log S(t)$$

Survival analysis allows us to study these functions and how covariates affect their shape/level.

[2] Books are inconsistent in where to put the equals sign. Some define $F(t) = \mathsf{P}(T < t)$ and $S(t) = \mathsf{P}(T \geq t)$.

Example: Exponential Distribution

- The **PDF** is $f(t) = \lambda e^{-\lambda t}$, $t \geq 0$, $\lambda > 0$
- The **CDF** is $F(t) = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t}$
- The **survival function** is $S(t) = 1 - F(t) = e^{-\lambda t}$
- The **hazard function** is **constant** over time:

$$h(t) = \frac{f(t)}{S(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda$$

- **Memoryless property**: for exponentials, the probability of a unit that has survived until time t survives an additional u is independent of t:

$$\begin{aligned} \mathsf{P}(T > t + u | T > t) &= \frac{\mathsf{P}(T > t + u)}{\mathsf{P}(T > t)} \\ &= \frac{S(t + u)}{S(t)} = \frac{e^{\lambda(t+u)}}{e^{-\lambda t}} = e^{-\lambda u} \end{aligned}$$

- Here we assume the PDF and derive the other functions. Survival models estimate parameters (e.g., λ), or estimate functions without assuming a parametric form (like a histogram)

Example: Weibull Distribution

The *Weibull distribution* has survival function

$$S(t) = \exp\left[-\left(\frac{t}{\alpha}\right)^{\beta}\right], \quad \text{for } t \geq 0,$$

where $\beta > 0$ determines the **shape** and α is a **scaling** factor. We can derive the other functions: $F(t) = 1 - S(t)$,

$$f(t) = -\frac{dS(t)}{dt} = \beta\alpha^{-\beta}t^{\beta-1}S(t), \quad \text{and}$$
$$h(t) = \frac{f(t)}{S(t)} = \beta\alpha^{-\beta}t^{\beta-1}.$$

The **hazard** has the shape of a *power function*, e.g.,

- $\beta = 1 : h(t) = t^0/\alpha = 1/\alpha$, which is constant. Thus the exponential distribution is a special case of the Weibull.
- $\beta = 2 : h(t) = 2t^1/\alpha^2 = k_1 t$, which increases *linearly* with t. This is also called the *Rayleigh distribution*
- $\beta = 3 : h(t) = 3t^2/\alpha^3 = k_2 t^2$, which has increasing slope.
- $\beta = 1.5 : h(t) = 1.5t^{0.5}/\alpha^{1.5} = k_3\sqrt{t}$, which has decreasing slope.

Estimating Functions With Survival Analysis

Survival analysis is a set of methods for understanding questions about the time when some "event" occurs (T) such as:

Q1 When is the event, e.g., cancelation or repurchase, likely to occur? Are there times when the event is more or less likely to happen? How long until we expect the event to occur?

Q2 How do **static characteristics** of customers affect the probability of T? e.g., acquisition source, demographics, and the length of the initial contract. Are customers acquired from one channel (e.g., telemarketing) systematically more likely to cancel than those from another (e.g., direct mail)? Are young people more likely to cancel than old?

Q3 How do things that happen during the relationship (**time-dependent covariates**) affect the probability of T?

There are many "survival analysis" methods

- Nonparametric "product-moment" estimates, e.g., Kaplan-Meier (KM) (answers Q1)
- Accelerated failure-time (AFT) model (Q1, Q2)
- Discrete-time survival model (Q1, Q2, Q3)
- Cox Proportional Hazard model (Q1, Q2, Q3)

Nonparametric "Product-Moment" Estimates

- Let n_t be the number of customers "at risk" at time t
- Let d_t be the number of customers who cancel at time t
- The product-moment estimate of the survival function is

$$\hat{S}(t) = \prod_{i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

- The *Kaplan-Meier* (KM) method counts all people who are censored at t as being at risk
- Intuition: if we had discrete time periods (e.g., months) where the probability of retaining a customer in period t is π_t then $S(t) = \mathsf{P}(T > t) = \prod_{i=1}^{t} \pi_i$. Note $1 - d_i/n_i$ estimates π_i
- Available in R with the `survfit` function
- Specify dependent variable as `Surv(T, event)`, where `event` equals one if the event happened and 0 if censored, and `T` is the time of the event/censoring.

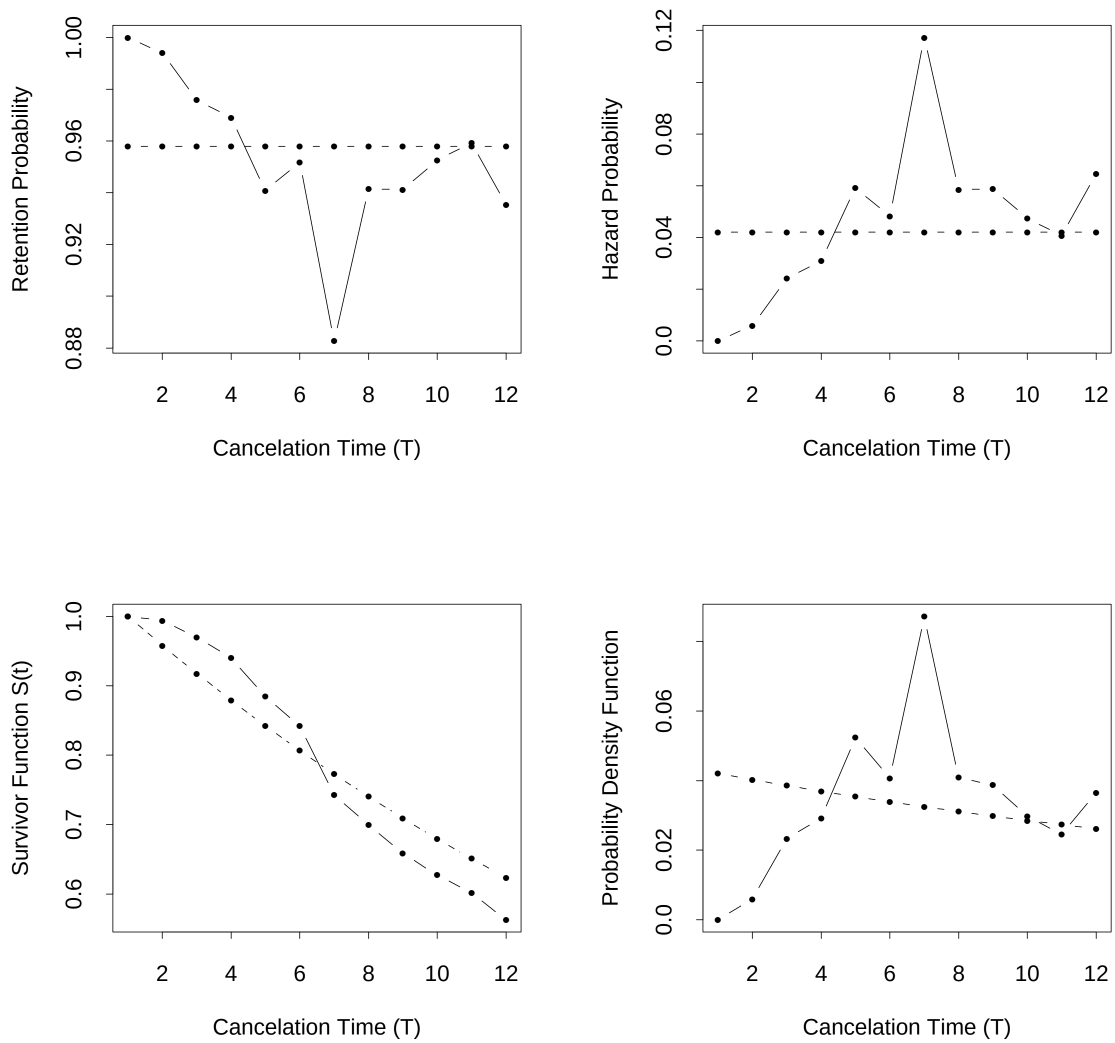
Educational Service Example

A Japanese cram school acquires junior high school children as subscribers to its service. For a sample of kids acquired within the past year, the following are the times of cancelation or censoring:

	Time of Cancelation/Censoring												
Censor	1	2	3	4	5	6	7	8	9	10	11	12	Total
No	0	4	16	20	37	28	61	24	19	13	10	13	245
Yes	3	0	2	1	7	33	49	63	30	16	34	188	426
Total	3	4	18	21	44	61	110	87	49	29	44	201	671

			Kaplan-Meier			Life Table	
t	Number Cancel d_t	Number Censor c_t	Number at Risk n_t	Retention Rate $1 - d_t/n_t$	Survivor Function $S(t)$	Number at Risk n_t	Survivor Function $S(t)$
1	0	3	671	1.0000	1.0000	669.5	1.0000
2	4	0	668	0.9940	0.9940	668	0.9940
3	16	2	664	0.9759	0.9701	663	0.9700
4	20	1	646	0.9690	0.9400	645.5	0.9400
5	37	7	625	0.9408	0.8844	621.5	0.8840
6	28	33	581	0.9518	0.8418	564.5	0.8402
7	61	49	520	0.8827	0.7430	495.5	0.7367
8	24	63	410	0.9415	0.6995	378.5	0.6900
9	19	30	323	0.9412	0.6584	308	0.6474
10	13	16	274	0.9526	0.6271	266	0.6158
11	10	34	245	0.9592	0.6015	228	0.5888
12	13	188	201	0.9353	0.5626	107	0.5173

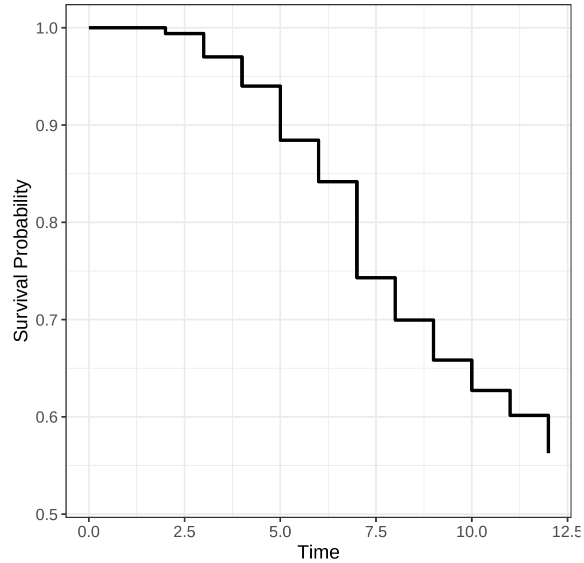
Educational Service Example Continued



Product-Moment Estimates in R

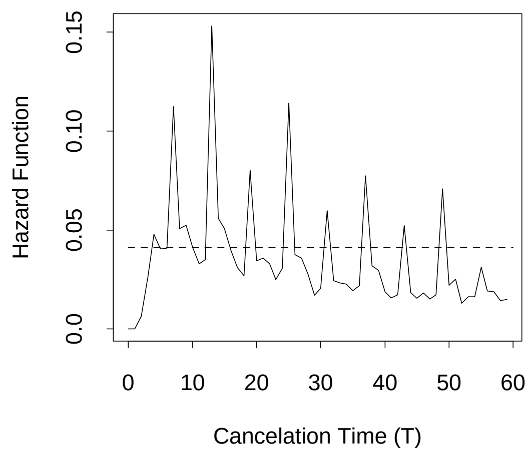
```
library(survival)
dat= read.table("service1yr.txt", header=T)
dat = data.frame(
  bigT = c(2:12, 1, 3:12),
  cancel = c(rep(1,11), rep(0,11)),
  count = c(4,16,20,37,28,61,24,19,13,10,13,3,2,1,7,33,49,63,30,16,34,188)
)
fit = survfit(Surv(bigT, cancel) ~ 1, data=dat, weight=count)
summary(fit)
Call: survfit(formula = Surv(bigT, cancel) ~ 1, data = dat, weights = count)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    2    668       4    0.994 0.00299        0.988        1.000
    3    664      16    0.970 0.00659        0.957        0.983
    4    646      20    0.940 0.00919        0.922        0.958
    5    625      37    0.884 0.01239        0.860        0.909
    6    581      28    0.842 0.01417        0.814        0.870
    7    520      61    0.743 0.01725        0.710        0.778
    8    410      24    0.700 0.01838        0.664        0.736
    9    323      19    0.658 0.01958        0.621        0.698
   10    274      13    0.627 0.02048        0.588        0.669
   11    245      10    0.602 0.02118        0.561        0.645
   12    201      13    0.563 0.02239        0.520        0.608
```

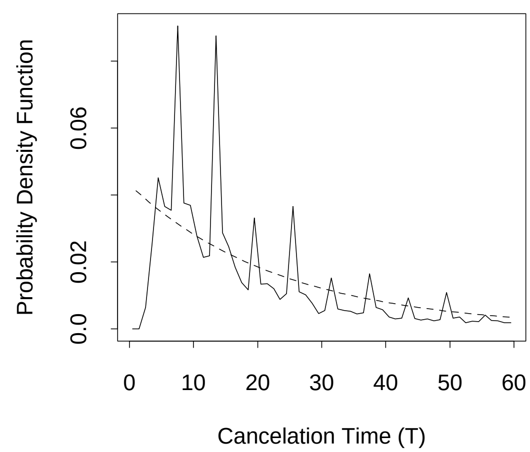
```
plot(fit) # basic version

library(ggsurvfit) # ggplot
fit %>%
  ggsurvfit(size = 1) +
  add_confidence_interval()
```

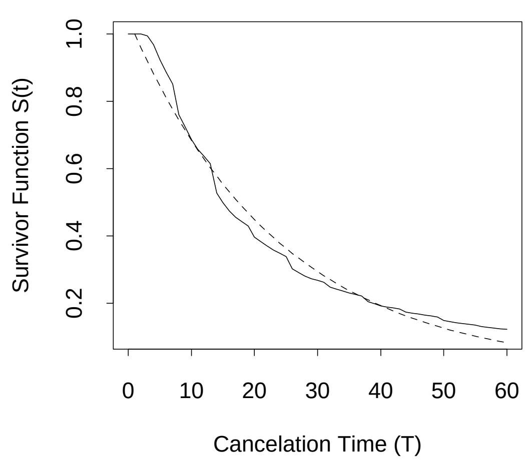


Service Provider, Five Years

```
service5yr = read.table("service5yr.txt", header=T)
fit2 = survfit(Surv(bigT, cancel) ~ 1, data=service5yr, weight=count)
```



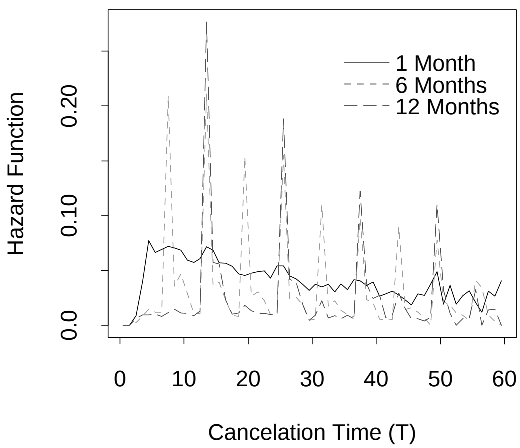


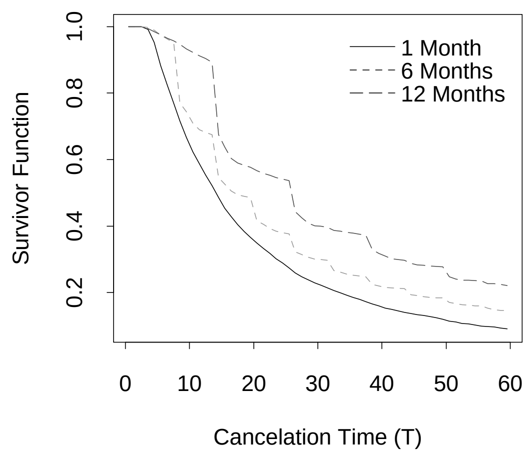


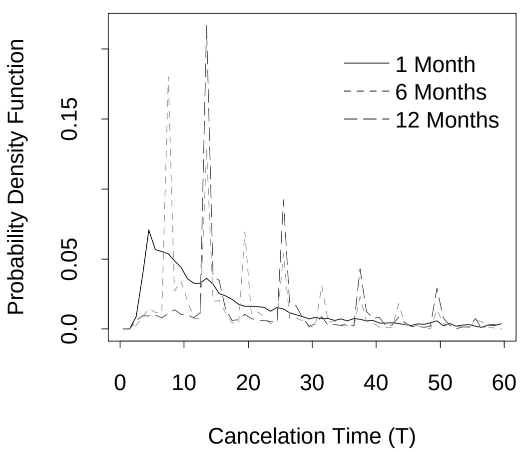
- Dashed line shows simple retention model (SRM).
- There is a distinct seasonal pattern, with large spikes every 12 months and smaller ones 6 months after the big ones.
- People sign 1-, 6- or 12-month contracts.
- The SRM fit of the hazard and PDF is poor, but the survival function looks OK.

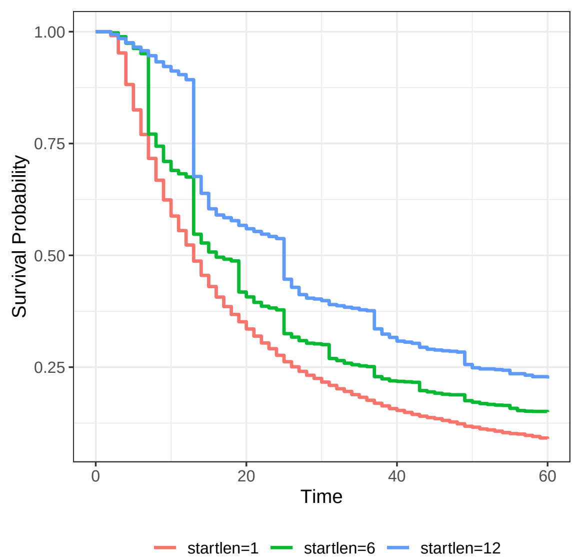
Service Provider, Five Years, Stratifying on Starting Contract Length

```
fit3 = survfit(Surv(bigT, cancel) ~ startlen, data=service5yr, weight=count)
fit3 %>% ggsurvfit(size = 1) + add_confidence_interval()
```









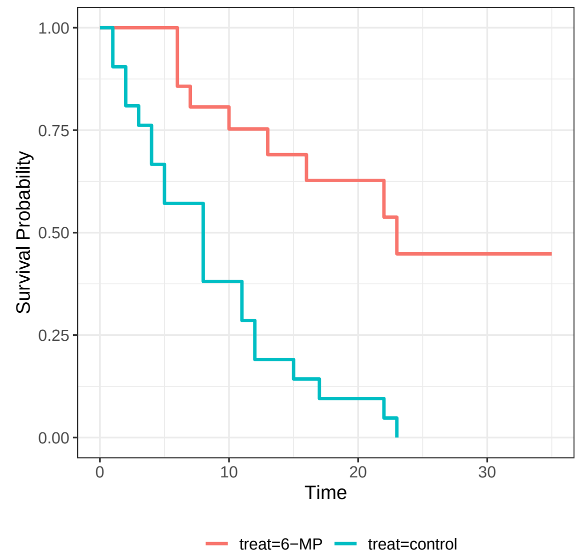
Gehan Leukemia Example

21 leukemia patients treated with new drug (6-mercaptopurine) and 21 matched controls (See Venables and Ripley, ch 13)

```
> library(survival)
> library(MASS)
> head(gehan) # cens poorly labeled, 1 if event happened and 0 for censored
  pair time cens   treat
1    1    1    1 control
2    1   10    1    6-MP
3    2   22    1 control
4    2    7    1    6-MP
5    3    3    1 control
6    3   32    0    6-MP
> fit = survfit(formula = Surv(time, cens) ~ treat, data = gehan)
> with(gehan, Surv(time, cens))
 [1]  1  10  22   7   3  32+ 12  23   8  22  17   6   2  16  11  34+  8  32+ 12
[20] 25+  2  11+  5  20+  4  19+ 15   6   8  17+ 23  35+  5   6  11  13   4   9+
[39]  1   6+  8  10+

> summary(fit)
Call: survfit(formula = Surv(time, cens) ~ treat, data = gehan)
                treat=6-MP
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    6     21       3    0.857  0.0764        0.720        1.000
    7     17       1    0.807  0.0869        0.653        0.996
   10     15       1    0.753  0.0963        0.586        0.968
   13     12       1    0.690  0.1068        0.510        0.935
   16     11       1    0.627  0.1141        0.439        0.896
   22      7       1    0.538  0.1282        0.337        0.858
   23      6       1    0.448  0.1346        0.249        0.807

                treat=control
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1     21       2   0.9048  0.0641      0.78754        1.000
    2     19       2   0.8095  0.0857      0.65785        0.996
    3     17       1   0.7619  0.0929      0.59988        0.968
    4     16       2   0.6667  0.1029      0.49268        0.902
    5     14       2   0.5714  0.1080      0.39455        0.828
    8     12       4   0.3810  0.1060      0.22085        0.657
   11      8       2   0.2857  0.0986      0.14529        0.562
   12      6       2   0.1905  0.0857      0.07887        0.460
   15      4       1   0.1429  0.0764      0.05011        0.407
   17      3       1   0.0952  0.0641      0.02549        0.356
   22      2       1   0.0476  0.0465      0.00703        0.322
   23      1       1   0.0000     NaN           NA           NA
```

Plotting and Inference on KM

```
> fit %>%
    ggsurvfit(size = 1) +
    add_confidence_interval()
```



- $\mathbb{V}(\hat{S}(t))$ can be estimated by **Greenwood's formula**:

$$\mathbb{V}[\hat{S}(t)] = [\hat{S}(t)]^2 \sum_{i:t_i \le t} \frac{d_i}{n_i(n - d_i)}$$

- A 95% CI for $S(t)$ is $\hat{S}(t) \pm 1.96\sqrt{\mathbb{V}[S(t)]}$
- **Log-rank test** evaluates $H_0 : S_1(t) = S_2(t), \forall t$.

```
> survdiff(Surv(time, cens)~treat, gehan)  # log-rank test of differences
Call: survdiff(formula = Surv(time, cens) ~ treat, data = gehan)
               N Observed Expected (O-E)^2/E (O-E)^2/V
treat=6-MP    21        9     19.3      5.46      16.8
treat=control 21       21     10.7      9.77      16.8

 Chisq= 16.8  on 1 degrees of freedom, p= 4.17e-05
```

The Proportional Hazard Model

$$h_i(t) = \lambda_0(t)\exp(\beta_1 x_{i1} + \cdots + \beta_p x_{ip})$$

- $h_i(t)$ is the hazard for individual i at time t
- $\lambda_0(t)$ is the unspecified baseline hazard function
- x_{ij} is the value of covariate j for individual i
- β_j is the effect of covariate j on the hazard function
- If we take the log of both sides, we get something that looks more like a linear regression model. Note that the (log) baseline hazard function determines the intercept.

$$\log[h_i(t)] = \log[\lambda_0(t)] + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

- The ratios of the hazards of two individuals i and i' is some constant (independent of time). This is called the *proportional hazards property.*

$$\frac{h_i(t)}{h_{i'}(t)} = \exp[\beta_1(x_{i1} - x_{i'1}) + \cdots + \beta_p(x_{ip} - x_{i'p})]$$

- With one predictor having two levels, the **hazard ratio** is

$$\frac{h(t|x=1)}{h(t|x=0)} = \frac{\lambda_0(t)\exp(\beta_1)}{\lambda_0(t)} = \exp(\beta_1)$$

Gehan Data with Cox PH

```
> fit4 = coxph(Surv(time, cens) ~ treat, gehan, method="exact") # cox
> plot(survfit(fit4),xlab="Remission (weeks)", ylab="Survival", cex=1.5)
> summary(fit4)
Call:
coxph(formula = Surv(time, cens) ~ treat, data = gehan, method = "exact")

  n= 42, number of events= 30

               coef exp(coef) se(coef)     z Pr(>|z|)
treatcontrol 1.6282    5.0949   0.4331 3.759  0.00017 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

             exp(coef) exp(-coef) lower .95 upper .95
treatcontrol     5.095     0.1963      2.18     11.91

Rsquare= 0.321   (max possible= 0.98 )
Likelihood ratio test= 16.25  on 1 df,   p=5.544e-05
Wald test            = 14.13  on 1 df,   p=0.0001704
Score (logrank) test = 16.79  on 1 df,   p=4.169e-05
```

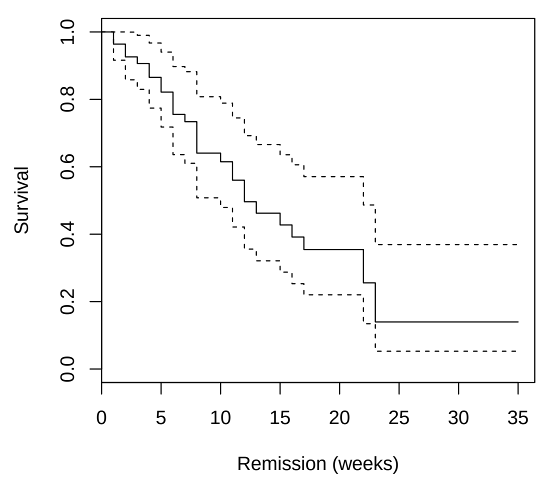
Let $x = 1$ for control, 0 for treatment

$$\log[h(t)] = \log[\lambda_0(t)] + 1.63x$$

$$h(t) = \lambda_0(t)\exp(1.63x)$$

Or, the hazard $h(t)$ is 5.095 times greater for those in the control group compared with the treatment group

KM estimate for average patient



Gehan Data with pairs as blocking variable

```
> fit5 = coxph(Surv(time, cens) ~ treat + factor(pair), gehan, method="exact") # cox
> summary(fit5)
Call:
coxph(formula = Surv(time, cens) ~ treat + factor(pair), data = gehan,
    method = "exact")

  n= 42, number of events= 30

                    coef exp(coef)  se(coef)      z Pr(>|z|)
treatcontrol    3.314679 27.513571  0.742620  4.463 8.06e-06 ***
factor(pair)2  -5.015219  0.006636  1.550131 -3.235 0.001215 **
factor(pair)3  -3.598195  0.027373  1.547371 -2.325 0.020053 *
....

               exp(coef) exp(-coef) lower .95 upper .95
treatcontrol   27.513571    0.03635 6.418e+00 117.94128
factor(pair)2   0.006636  150.68919 3.180e-04   0.13848
factor(pair)3   0.027373   36.53222 1.319e-03   0.56813
...
> anova(fit4, fit5)
Analysis of Deviance Table
 Cox model: response is  Surv(time, cens)
 Model 1: ~ treat
 Model 2: ~ treat + factor(pair)
   loglik  Chisq Df P(>|Chi|)
1 -74.543
2 -59.915 29.256 20   0.08283 .

> drop1(fit5, test="Chisq")
Single term deletions

Surv(time, cens) ~ treat + factor(pair)
             Df    AIC    LRT  Pr(>Chi)
<none>          161.83
treat         1 190.16 30.328 3.648e-08 ***
factor(pair) 20 151.09 29.256   0.08283 .
```

Discrete-Time Survival Analysis With Logistic Regression

- Assume discrete, equal-sized time intervals $t = 1, 2, \ldots$
- Let $\pi_{it} = \mathsf{P}(T = t | T > t - 1)$ be the probability that customer i cancels during period t, given the customer has survived $t - 1$ periods
- Objective: model π_{it} as a function of
  - *Baseline hazard function*
  - *Static covariates* such as the lenght of the initial contract, demographics, or acquisition source
  - *Time-dependent covariates* such as whether the contract is expiring in the current period, usage of the product/service, or complaints
- Example model: logistic regression

$$\log\left(\frac{\pi_{it}}{1 - \pi_{it}}\right) = \alpha_t + x_{i1t}\beta_1 + \cdots + x_{ipt}\beta_p, t = 1, 2, \ldots$$

  where x_{ijt} is the value of covariate j at time t for customer i.

- Approach:
  1. Identify study period and sample customers
  2. Define response variable

$$y_{it} = \begin{cases} 1 & \text{customer } i \text{ cancels in period } t \\ 0 & \text{otherwise} \end{cases}$$

  3. Develop analysis data set:
     - Each customer contributes varying numbers of observations
     - If customer i cancels in period 3, this customer gets 3 observations ($y_{i1} = 0$, $y_{i2} = 0$, $y_{i3} = 1$)
     - If customer never cancels, all $y_{it} = 0$ during the study period

Service Example

We start with data

```
> dat
   bigT cancel count
1     2      1     4
2     3      1    16
3     4      1    20
4     5      1    37
5     6      1    28
6     7      1    61
7     8      1    24
8     9      1    19
9    10      1    13
10   11      1    10
11   12      1    13
12    1      0     3
13    3      0     2
14    4      0     1
15    5      0     7
16    6      0    33
17    7      0    49
18    8      0    63
19    9      0    30
20   10      0    16
21   11      0    34
22   12      0   188
```

We convert it as follows to "long" format

```
> long=survSplit(data=dat, cut=0:12,
    end="bigT", event="cancel")
> head(long, 20)
   count tstart bigT cancel
1      4      0    1      0
2      4      1    2      1
3     16      0    1      0
4     16      1    2      0
5     16      2    3      1
6     20      0    1      0
7     20      1    2      0
8     20      2    3      0
9     20      3    4      1
10    37      0    1      0
11    37      1    2      0
12    37      2    3      0
13    37      3    4      0
14    37      4    5      1
15    28      0    1      0
16    28      1    2      0
17    28      2    3      0
18    28      3    4      0
19    28      4    5      0
20    28      5    6      1
...

78      3      0    1      0
79      2      0    1      0
80      2      1    2      0
81      2      2    3      0
82      1      0    1      0
83      1      1    2      0
84      1      2    3      0
85      1      3    4      0
86      7      0    1      0
...
```

Service Example: Intercept Model

```
> fit=glm(cancel ~ 1, binomial, long, weight=count)
> summary(fit)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.12622    0.06527  -47.89   <2e-16 ***
```

```
PROC LOGISTIC DATA=long DESCENDING;
  MODEL cancel = ;
  WEIGHT count;
  OUTPUT OUT=tmp PREDICTED=phat;
RUN;
PROC PRINT DATA=tmp(OBS=1);
RUN;
```

```
                              -2 Log L = 2032.447

                                             Standard          Wald
      Parameter    DF    Estimate      Error    Chi-Square    Pr > ChiSq
      Intercept     1     -3.1262     0.0653     2293.7927        <.0001


     Obs     count      censored     bigT     t     cancel     _LEVEL_       phat
      1        3            1          1      1        0          1       0.042038
```

In our discussion of the simple retention model we found

$$\hat{r} = 1 - \frac{n}{\sum d_t + \sum c_t} = 1 - \frac{245}{5828} = 0.957962$$

which equals the result given above

$$1 - 0.957962 = 0.042038 = \frac{1}{1 + e^{-(-3.1262)}}$$

Allowing Intercept to Vary over Time

```
PROC LOGISTIC DATA=long DESCENDING;
  CLASS t;
  MODEL cancel = t;
  WEIGHT count;
  OUTPUT OUT=tmp PREDICTED=phat;
RUN;
```

Criterion	Intercept Only	Intercept and Covariates
-2 Log L	2032.447	1871.360

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	161.0865	11	<.0001

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-4.3831	28.7290	0.0233	0.8787
t	1	1	-13.8113	316.0	0.0019	0.9651
t	2	1	-0.7289	28.7327	0.0006	0.9798
t	3	1	0.6818	28.7299	0.0006	0.9811
t	4	1	0.9395	28.7298	0.0011	0.9739
t	5	1	1.6173	28.7294	0.0032	0.9551
t	6	1	1.3999	28.7296	0.0024	0.9611
t	7	1	2.3649	28.7293	0.0068	0.9344
t	8	1	1.6053	28.7296	0.0031	0.9554
t	9	1	1.6105	28.7298	0.0031	0.9553
t	10	1	1.3835	28.7302	0.0023	0.9616
t	11	1	1.2261	28.7305	0.0018	0.9660

$$\log\left(\frac{\pi_{it}}{1-\pi_{it}}\right) = \alpha + \alpha_t, t = 1, 2, \ldots, 11$$

We reject $H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_{11} = 0$, implying the retention rate is not constant.

Service Example: Allowing Intercept to Vary over Time

```
PROC SQL;
  SELECT UNIQUE(t), phat FORMAT=7.4, 1-phat as r FORMAT=7.4
  FROM tmp
  ORDER BY t;

                Estimated
              t  Probability        r
       ------------------------------
              1       0.0000   1.0000
              2       0.0060   0.9940
              3       0.0241   0.9759
              4       0.0310   0.9690
              5       0.0592   0.9408
              6       0.0482   0.9518
              7       0.1173   0.8827
              8       0.0585   0.9415
              9       0.0588   0.9412
             10       0.0474   0.9526
             11       0.0408   0.9592
             12       0.0647   0.9353
```

			Kaplan-Meier			Life Table	
t	Number Cancel d_t	Number Censor c_t	Number at Risk n_t	Retention Rate $1 - d_t/n_t$	Survivor Function $S(t)$	Number at Risk n_t	Survivor Function $S(t)$
1	0	3	671	1.0000	1.0000	669.5	1.0000
2	4	0	668	0.9940	0.9940	668	0.9940
3	16	2	664	0.9759	0.9701	663	0.9700
4	20	1	646	0.9690	0.9400	645.5	0.9400
5	37	7	625	0.9408	0.8844	621.5	0.8840
6	28	33	581	0.9518	0.8418	564.5	0.8402
7	61	49	520	0.8827	0.7430	495.5	0.7367
8	24	63	410	0.9415	0.6995	378.5	0.6900
9	19	30	323	0.9412	0.6584	308	0.6474
10	13	16	274	0.9526	0.6271	266	0.6158
11	10	34	245	0.9592	0.6015	228	0.5888
12	13	188	201	0.9353	0.5626	107	0.5173

Service Example: Allowing Intercept to Vary over Time

The previous analysis is problematic because no one with $T = 1$ cancels because they are all censored. Since there is no variation in the dependent variable for this group, the standard errors are off. This can be fixed by dropping this group.

```
PROC LOGISTIC DATA=long DESCENDING;
  CLASS t;
  MODEL cancel = t;
  WEIGHT count;
  WHERE t>1;
RUN;
```

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-3.1275	0.0823	1444.3921	<.0001
t	2	1	-1.9845	0.4610	18.5284	<.0001
t	3	1	-0.5738	0.2433	5.5644	0.0183
t	4	1	-0.3161	0.2213	2.0399	0.1532
t	5	1	0.3617	0.1740	4.3208	0.0376
t	6	1	0.1443	0.1936	0.5560	0.4559
t	7	1	1.1093	0.1482	56.0195	<.0001
t	8	1	0.3497	0.2073	2.8453	0.0916
t	9	1	0.3549	0.2292	2.3981	0.1215
t	10	1	0.1279	0.2699	0.2247	0.6355
t	11	1	-0.0295	0.3034	0.0095	0.9225

Static and Time-Dependent Covariates

$$\log\left(\frac{\pi_{it}}{1-\pi_{it}}\right) = \alpha_t + x_{i1t}\beta_1 + \cdots + x_{ipt}\beta_p, t = 1, 2, \ldots$$

$$\log\left(\frac{\pi_{it}}{1-\pi_{it}}\right) = \lambda(t) + x_{i1t}\beta_1 + \cdots + x_{ipt}\beta_p, t = 1, 2, \ldots$$

- α_t or some function $\lambda(t)$ describe the shape of the *baseline hazard function*
- x_{ijt} is the value of covariate j at time t for customer i.
  - *Static covariate*: $x_{ij1} = x_{ij2} = \cdots = x_{ij}$ does not change over time, e.g., gender, starting contract length, acquisition source. Notice that the effect of the covariate is to shift the entire (logit of the) baseline hazard function up or down.
  - *Time-dependent covariates* change over time.

Service Example With Time-Dependent Covariates: lagged tests

Input Data

custid	pay0	pay1	pay2	pay3	pay4	pay5	pay6	pay7	pay8	pay9	pay10	pay11
137	6	6	6	6	6	6	6	6	6	6	6	6
143	6	6	6	6	6	6	0	0	0	0	0	0
160	6	6	6	6	6	6	0	0	0	0	0	0
163	1	1	1	1	1	0	0	1	1	1	1	1
165	1	1	1	1	1	0	0	0	0	0	0	0
5993	1	0	0	0	0	0	0	0	0	0	0	0
6030	12	12	12	0	0	0	0	0	0	0	0	0
6610	1	1	1	1	6	6	6	6	6	6	6	6

custid	test0	test1	test2	test3	test4	test5	test6	test7	test8	test9	test10	test11
137	4	4	4	4	4	4	0	4	4	4	4	0
143	4	4	4	4	4	0	0	0	0	0	0	0
160	4	4	4	4	4	4	0	0	0	0	0	0
163	4	4	4	4	4	0	0	4	4	4	4	4
165	0	0	0	0	0	0	0	0	0	0	0	0
5993	0	0	0	0	0	0	0	0	0	0	0	0
6030	4	4	4	0	0	0	0	0	0	0	0	0
6610	4	4	4	4	4	4	4	4	1	3	4	4

Data for Logistic Regression

custid=137			
T	startlen	cancelnow	lagnotest
1	6	0	0
2	6	0	0
3	6	0	0
4	6	0	0
5	6	0	0
6	6	0	0
7	6	0	1
8	6	0	0
9	6	0	0
10	6	0	0
11	6	0	0

custid=143			
T	startlen	cancelnow	lagnotest
1	6	0	0
2	6	0	0
3	6	0	0
4	6	0	0
5	6	0	0
6	6	1	1

custid=5993			
T	startlen	cancelnow	lagnotest
1	1	1	1

```
fit = glm(cancelnow ~ t*startlen + lagnotest, binomial, long)
```

Static Covariates

```
> long = read.csv("long1.csv")
> fit = glm(cancelnow ~ factor(t)+factor(startlen), binomial, long)
> summary(fit)
Call: glm(formula=cancelnow ~ factor(t) + factor(startlen), family=binomial, data=long)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.79486    0.19041 -25.182  < 2e-16 ***
factor(t)2          1.30803    0.21469   6.093 1.11e-09 ***
factor(t)3          2.02284    0.20323   9.953  < 2e-16 ***
factor(t)4          1.69775    0.20872   8.134 4.16e-16 ***
factor(t)5          1.80559    0.20766   8.695  < 2e-16 ***
factor(t)6          3.20476    0.19565  16.380  < 2e-16 ***
factor(t)7          2.28594    0.20371  11.222  < 2e-16 ***
factor(t)8          1.92716    0.21004   9.175  < 2e-16 ***
factor(t)9          1.64608    0.21696   7.587 3.28e-14 ***
factor(t)10         1.34490    0.22640   5.940 2.84e-09 ***
factor(t)11         1.29462    0.22906   5.652 1.59e-08 ***
factor(startlen)6  -0.54973    0.05454 -10.079  < 2e-16 ***
factor(startlen)12 -1.79223    0.12838 -13.961  < 2e-16 ***

    Null deviance: 14023  on 43441  degrees of freedom
Residual deviance: 12814  on 43429  degrees of freedom
```

Static Covariates with Separate Baseline Hazard Functions

```
> fit2 = glm(cancelnow ~ factor(t)*factor(startlen), binomial, long)
> summary(fit2)
Call: glm(cancelnow ~ factor(t) * factor(startlen), family=binomial, data=long)

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -4.71671    0.22460 -21.000  < 2e-16 ***
factor(t)2                     1.36958    0.25312   5.411 6.27e-08 ***
factor(t)3                     2.17056    0.23941   9.066  < 2e-16 ***
factor(t)4                     1.85187    0.24543   7.546 4.51e-14 ***
factor(t)5                     2.09170    0.24254   8.624  < 2e-16 ***
factor(t)6                     2.24824    0.24147   9.311  < 2e-16 ***
factor(t)7                     2.27169    0.24250   9.368  < 2e-16 ***
factor(t)8                     2.06760    0.24751   8.354  < 2e-16 ***
factor(t)9                     1.67816    0.25850   6.492 8.47e-11 ***
factor(t)10                    1.62410    0.26167   6.207 5.41e-10 ***
factor(t)11                    1.51559    0.26695   5.677 1.37e-08 ***
factor(startlen)6             -1.68243    0.61923  -2.717 0.006588 **
factor(startlen)12            -0.15543    0.50197  -0.310 0.756840
factor(t)2:factor(startlen)6   0.43220    0.67319   0.642 0.520859
factor(t)3:factor(startlen)6   0.08886    0.65315   0.136 0.891780
factor(t)4:factor(startlen)6   0.22401    0.66128   0.339 0.734797
factor(t)5:factor(startlen)6  -0.43459    0.67755  -0.641 0.521257
factor(t)6:factor(startlen)6   2.81119    0.62836   4.474 7.68e-06 ***
factor(t)7:factor(startlen)6   0.87997    0.64201   1.371 0.170484
factor(t)8:factor(startlen)6   0.39035    0.65958   0.592 0.553972
factor(t)9:factor(startlen)6   0.75780    0.66504   1.139 0.254506
factor(t)10:factor(startlen)6 -0.97674    0.80762  -1.209 0.226509
factor(t)11:factor(startlen)6 -0.30303    0.74021  -0.409 0.682256
factor(t)2:factor(startlen)12 -1.02233    0.64032  -1.597 0.110354
factor(t)3:factor(startlen)12 -1.35415    0.59273  -2.285 0.022337 *
factor(t)4:factor(startlen)12 -1.81601    0.68070  -2.668 0.007634 **
factor(t)5:factor(startlen)12 -2.96896    0.87288  -3.401 0.000671 ***
factor(t)6:factor(startlen)12 -2.01729    0.65430  -3.083 0.002049 **
factor(t)7:factor(startlen)12 -1.41675    0.59404  -2.385 0.017082 *
factor(t)8:factor(startlen)12 -2.21770    0.71725  -3.092 0.001988 **
factor(t)9:factor(startlen)12 -1.59684    0.68558  -2.329 0.019849 *
factor(t)10:factor(startlen)12 -1.53444    0.68679  -2.234 0.025468 *
factor(t)11:factor(startlen)12 -1.41752    0.68883  -2.058 0.039602 *

    Null deviance: 14023  on 43441  degrees of freedom
Residual deviance: 12324  on 43409  degrees of freedom
```

Time-Dependent Covariates with Separate Baseline Hazard Functions

```
> fit3 = glm(cancelnow ~ factor(t)*factor(startlen) + lagnotest, binomial, long)
> summary(fit3)
Call: glm(formula = cancelnow ~ factor(t) * factor(startlen) + lagnotest,
    family = binomial, data = long)

Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -6.83773    0.24467 -27.947  < 2e-16 ***
factor(t)2                      1.16608    0.25672   4.542 5.57e-06 ***
...
factor(t)11                     1.21158    0.27115   4.468 7.89e-06 ***
factor(startlen)6              -1.33989    0.62230  -2.153 0.031309 *
factor(startlen)12              0.45563    0.50971   0.894 0.371378
lagnotest                       3.40654    0.10090  33.763  < 2e-16 ***
factor(t)2:factor(startlen)6    0.42836    0.67771   0.632 0.527337
...
factor(t)11:factor(startlen)6  -0.63951    0.74443  -0.859 0.390309
factor(t)2:factor(startlen)12  -0.99217    0.65135  -1.523 0.127695
...
factor(t)11:factor(startlen)12 -2.02478    0.69723  -2.904 0.003684 **

    Null deviance: 14023.2  on 43441  degrees of freedom
Residual deviance:  9822.1  on 43408  degrees of freedom
```

Including Contract Up

Input Data

custid	pay0	pay1	pay2	pay3	pay4	pay5	pay6	pay7	pay8	pay9	pay10	pay11
137	6	6	6	6	6	6	6	6	6	6	6	6
143	6	6	6	6	6	6	0	0	0	0	0	0
160	6	6	6	6	6	6	0	0	0	0	0	0
163	1	1	1	1	1	0	0	1	1	1	1	1
165	1	1	1	1	1	0	0	0	0	0	0	0
5993	1	0	0	0	0	0	0	0	0	0	0	0
6030	12	12	12	0	0	0	0	0	0	0	0	0
6610	1	1	1	1	6	6	6	6	6	6	6	6

Data for Logistic Regression

custid=137				
T	lag notest	pay left	contract up	cancel now
1	0	5	0	0
2	0	4	0	0
3	0	3	0	0
4	0	2	0	0
5	0	1	0	0
6	0	0	1	0
7	1	5	0	0
8	0	4	0	0
9	0	3	0	0
10	0	2	0	0
11	0	1	0	0

custid=6610				
T	lag notest	pay left	contract up	cancel now
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	0	1	0
5	0	5	0	0
6	0	4	0	0
7	0	3	0	0
8	0	2	0	0
9	0	1	0	0
10	0	0	1	0
11	0	5	0	0

custid=6030				
T	lag notest	pay left	contract up	cancel now
1	0	11	0	0
2	0	10	0	0
3	0	9	0	1

Including Contract Up

```
> long2 = read.csv("long2.csv")
> fit = glm(cancelnow ~ factor(t) + lagnotest + contractup, binomial, long2)
> summary(fit)
Call: glm(formula = cancelnow ~ factor(t) + lagnotest + contractup,
    family = binomial, data = long2)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.94287    0.21891 -36.284  < 2e-16 ***
factor(t)2   1.11588    0.21818   5.115 3.14e-07 ***
factor(t)3   1.80597    0.20690   8.729  < 2e-16 ***
factor(t)4   1.26393    0.21203   5.961 2.50e-09 ***
factor(t)5   1.31435    0.21095   6.231 4.64e-10 ***
factor(t)6   2.49876    0.19912  12.549  < 2e-16 ***
factor(t)7   1.91132    0.20750   9.211  < 2e-16 ***
factor(t)8   1.55470    0.21382   7.271 3.56e-13 ***
factor(t)9   1.24398    0.22062   5.638 1.72e-08 ***
factor(t)10  1.09433    0.23027   4.752 2.01e-06 ***
factor(t)11  0.94036    0.23273   4.041 5.33e-05 ***
lagnotest    3.32263    0.10001  33.224  < 2e-16 ***
contractup   1.34556    0.07151  18.815  < 2e-16 ***

    Null deviance: 14023  on 43441  degrees of freedom
Residual deviance: 10075  on 43429  degrees of freedom
```