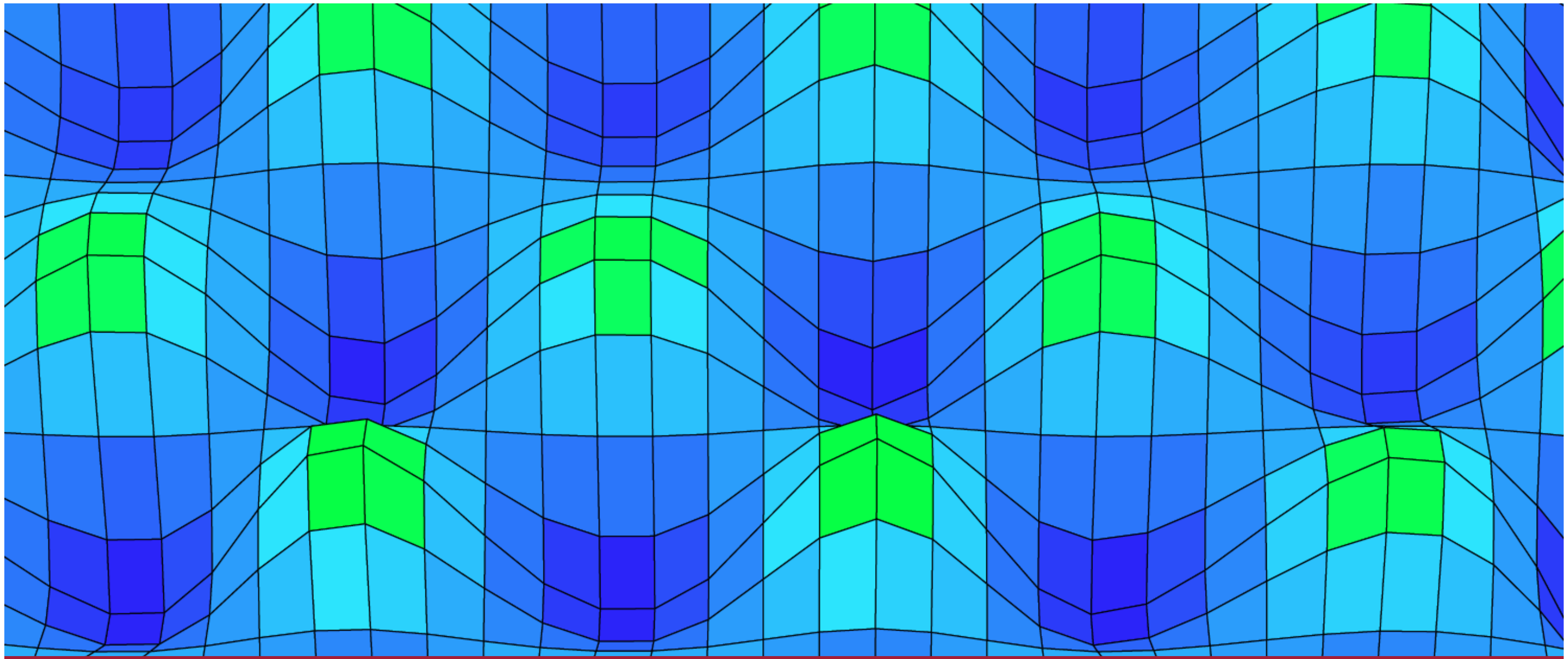
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Philosophische Fakultät
Seminar für Sprachwissenschaft



Advanced regression models: introduction to survival analysis

Peter Hendrix

Survival analysis

Slides:

http://www.peterhendrix.com/ARMsurvival.pdf

Slides and data:

http://www.peterhendrix.com/ARMsurvival.zip

Survival analysis

What is survival analysis?

Survival analysis

- Statistical techniques that model the time until an event of interest occurs
- Events of interest:
    - death (medicine)
    - failure of a mechanical device (engineering)
    - recidivism (sociology)

Survival analysis

- Linguistic data suitable for survival analysis:
  - survival of words in a language
  - reaction time studies
  - eye-movement patterns
  - ...

Survival analysis

- Why?
  - ability to model predictor effects as a function of time, even when the dependent variable is uni-dimensional
  - insight into the temporal development of language processing

Survival analysis

- Functions of interest:
  - survival function
  - hazard function
  - cumulative hazard function

Load data

```r
# Load data
load("data/blp-arm.rda")

# Show dimensions
dim(blp)
# [1] 17303     9
```

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Load data

```
# Show head
head(blp)
#        word       rt status logfrequency length  logold20
# 1     aback 728.3750      1   -1.2238354      5 0.6151856
# 2     abbey 627.5526      1   -0.1251098      5 0.6678294
# 3     abbot 873.8148      1   -1.0415705      5 0.6678294
# 4    abduct 684.7368      1   -0.3482817      6 0.8754687
# 5     abhor 752.2857      1   -1.4468940      5 0.7884574
# 6 abhorred 808.8846      1   -1.9855870      8 1.0473190
#   summedbigramfrequency    sem20 nsyl
# 1               1132695 11.91440    2
# 2               1181314 10.59987    2
# 3               1298460 11.12063    2
# 4                540440 12.28923    2
# 5               1841601 11.34443    2
# 6               4587386 11.40531    2
```

Survival functions

- Functions of interest:
  - survival function
  - hazard function
  - cumulative hazard function

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Survival function

- The survival function describes the probability of the time at which the event of interest occurs being greater than a given time t:

$$S(t) = P(T > t)$$

  where T is the time at which the event of interest occurs

- The survival function can be derived from the probability density function:

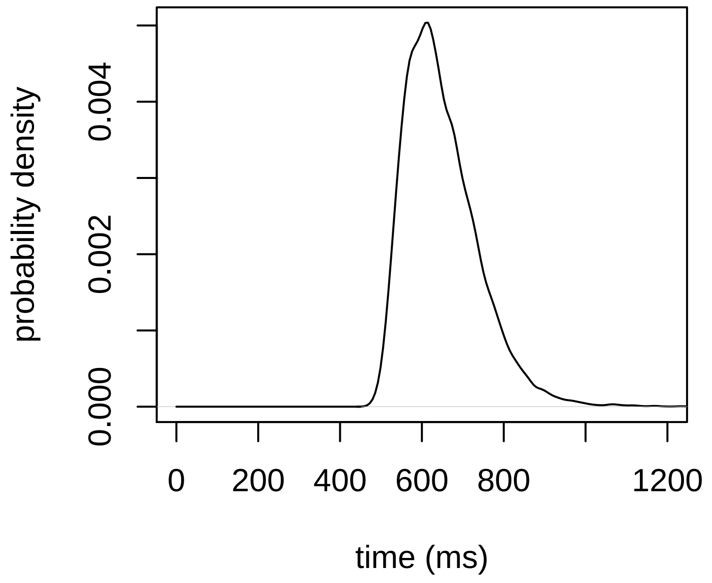
$$S(t) = \int_t^\infty f(x)dx = 1 - F(t)$$

Survival function

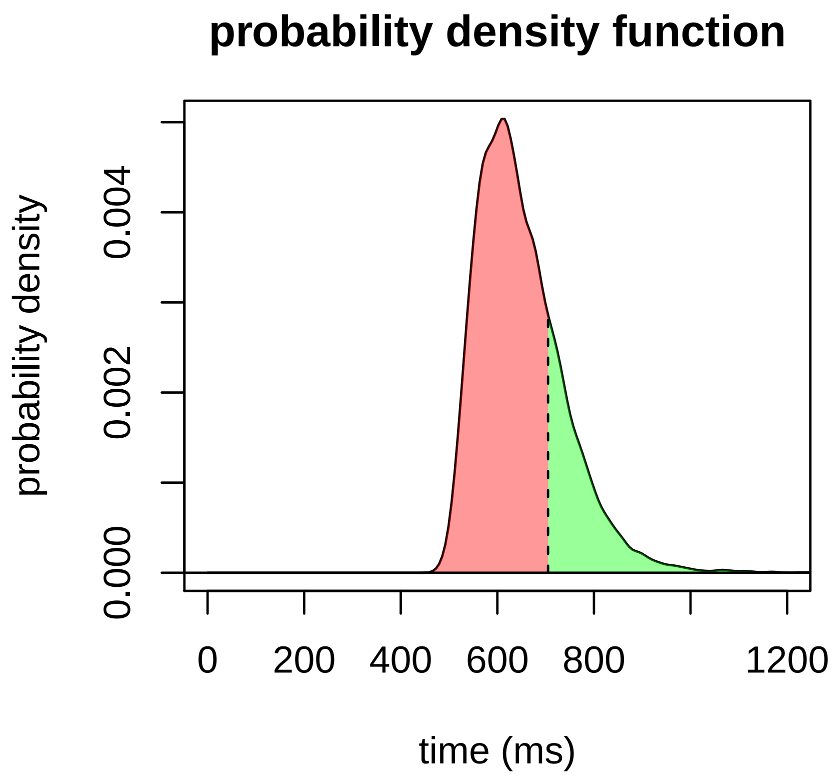
What is S(700)?

Probability density function

```r
# Plot density
plot(density(blp$rt), xlab = "time (ms)", ylab = "probability density",
     main = "", xlim = c(0, 1200))
```

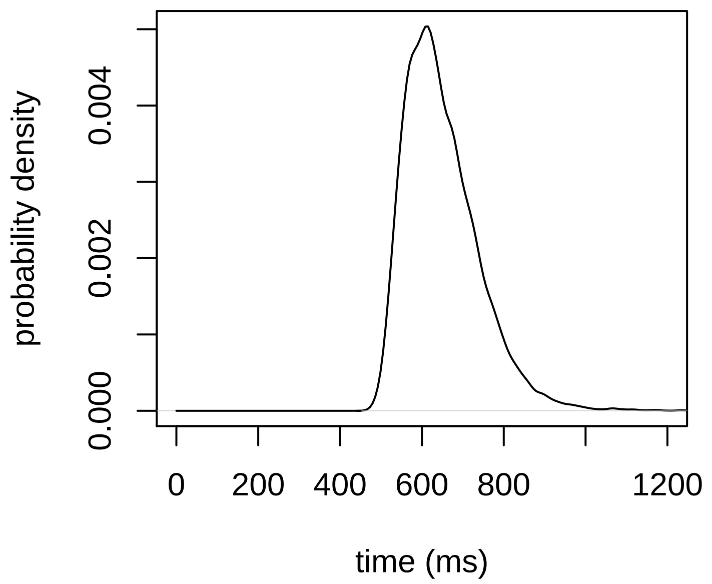


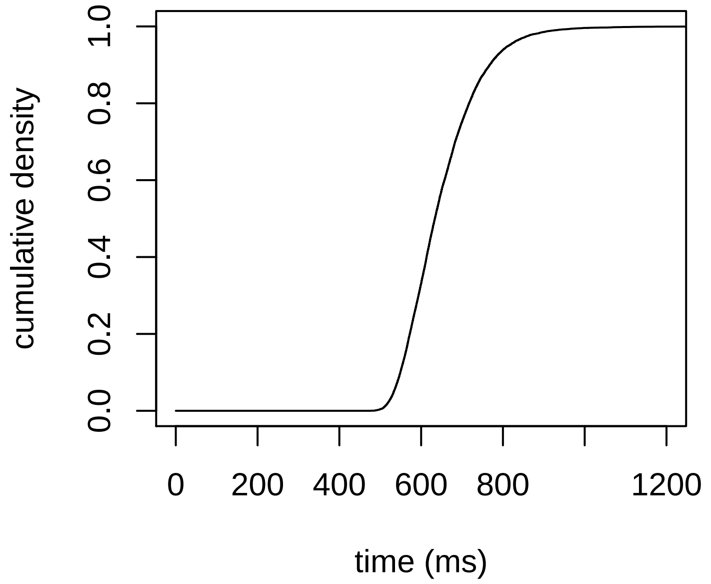
Probability density function



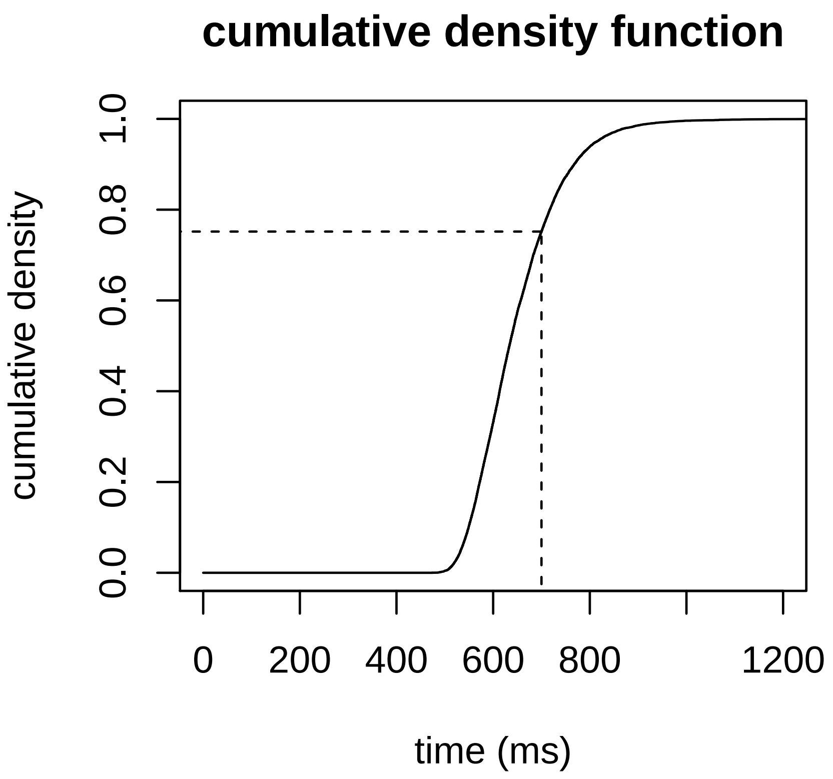
Cumulative density function

```r
# Plot cumulative density
cdf = ecdf(blp$rt)
plot(sort(blp$rt), cdf(sort(blp$rt)), type = "l", xlab = "time (ms)",
     ylab = "cumulative density", xlim = c(0, 1200))
segments(0, 0, min(blp$rt), 0)
```



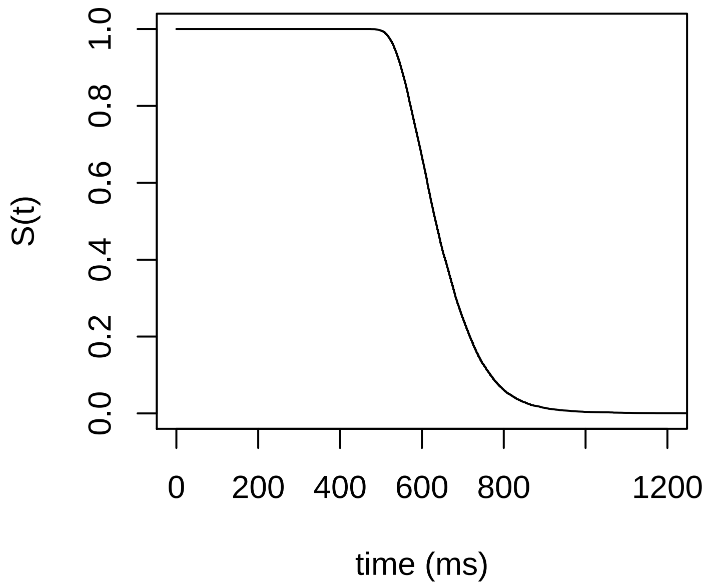


Cumulative density function

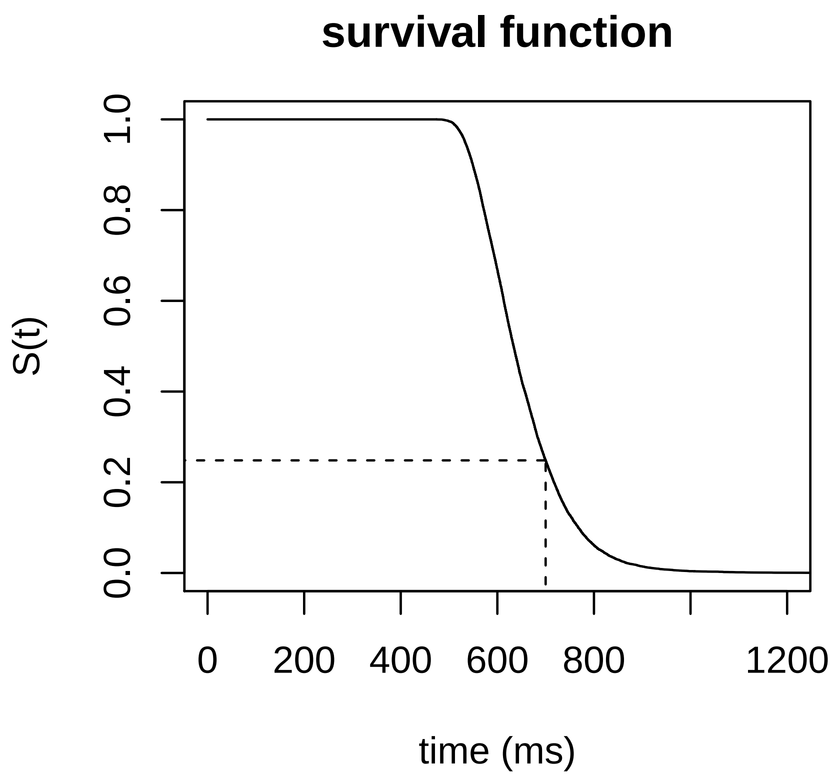


Survival function

```
# Plot survival function
plot(sort(blp$rt), 1 - cdf(sort(blp$rt)), type = "l",
     xlab = "time (ms)", ylab = "S(t)", xlim = c(0, 1200))
segments(0, 1, min(blp$rt), 1)
```



Survival function



Survival function

$$S(700) = 0.248$$

Survival functions

- Functions of interest:
    - survival function
    - hazard function
    - cumulative hazard function

Hazard function

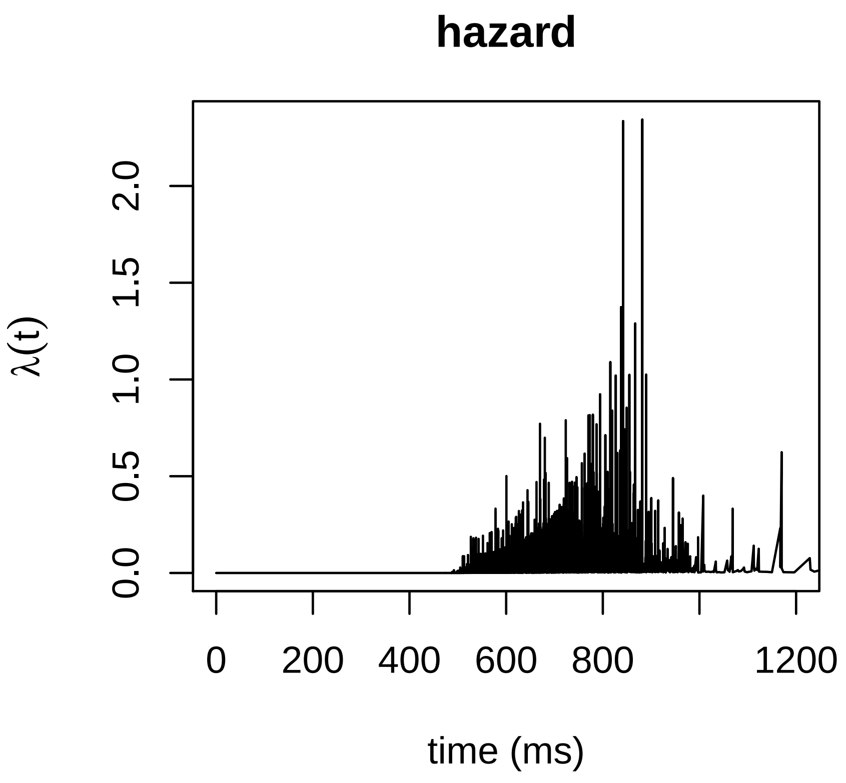
- The hazard function describes the instantaneous rate of the event of interest occurring, given that it has not occurred thus far:

$$\begin{aligned}\lambda(t) &= \lim_{dt \to 0} \frac{P(t \leq T \leq t + d \mid T \geq t)}{dt} \\ &= -\frac{d}{dt} \log(S(t))\end{aligned}$$

Hazard function

```r
# Define t
t = sort(blp$rt)

# Define S
S = 1 - cdf(sort(blp$rt))

# Calculate hazard rate
hr = diff(-log(S))/diff(t)
hr = c(hr, NA)

# Plot
plot(t, hr, type = "l", xlab = "time (ms)",
     ylab = expression(lambda(t)), main = "hazard",
     xlim = c(0, 1200))
```

Hazard function



Survival functions

- Functions of interest:
  - survival function
  - hazard function
  - cumulative hazard function

Cumulative hazard function

- The cumulative hazard function is obtained by integrating over the hazard function:

$$\Lambda(t) = \int_0^t \lambda(x)dx = -\log S(t)$$

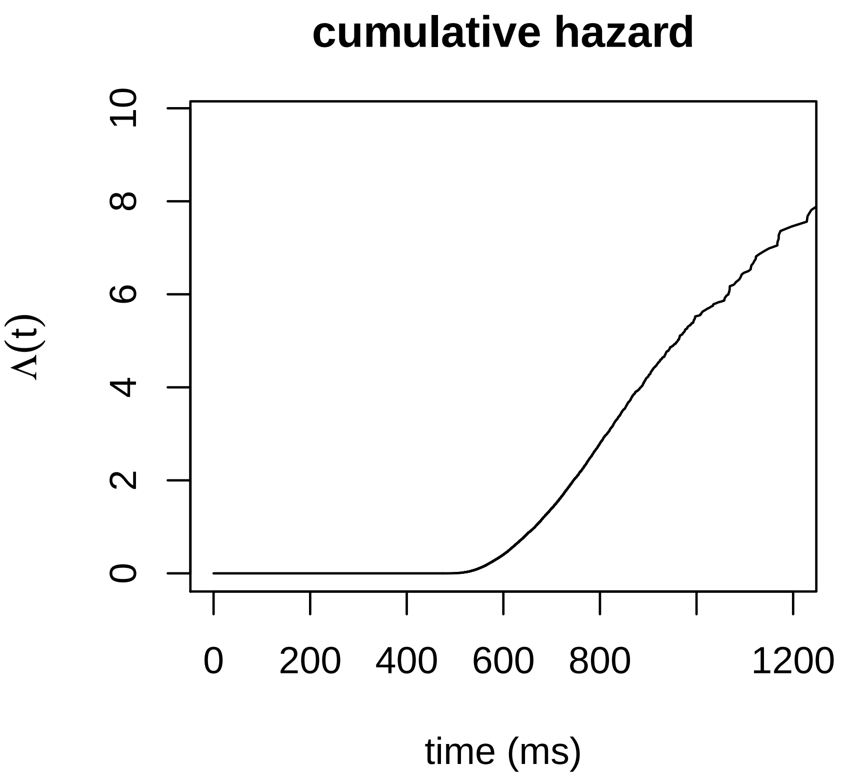
- No straightforward interpretation
- Mathematically useful

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Cumulative hazard function

```r
# Define t
t = sort(blp$rt)

# Define S
S = 1 - cdf(sort(blp$rt))

# Calculate hazard rate
cumh = -log(S)

# Plot
plot(t, cumh, type = "l", xlab = "time (ms)",
     ylab = expression(Lambda(t)), main = "cumulative hazard",
     xlim = c(0, 1200))
```

Cumulative hazard function



Estimation of survival functions

- The data provide an empirical survival function for a sample from the population
- How to estimate the survival function for the population?

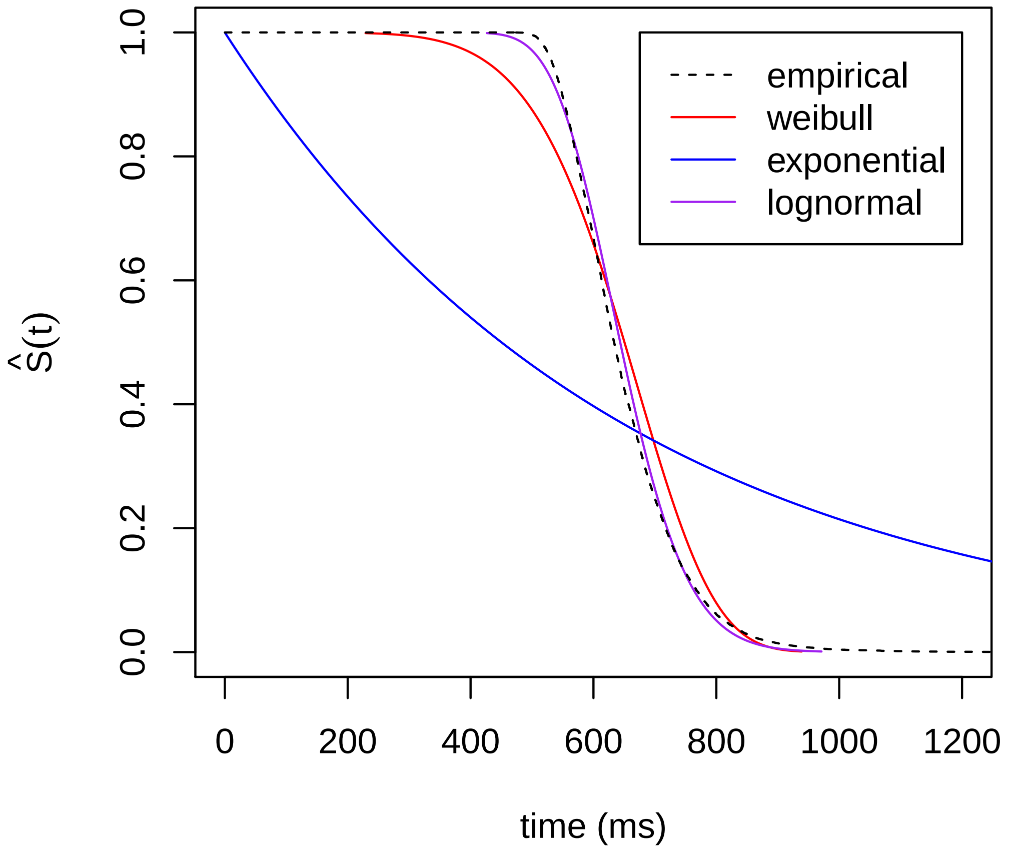
Estimation of survival functions

- The survival function can be estimated through two types of methods:
  - parametric methods
  - non-parametric methods

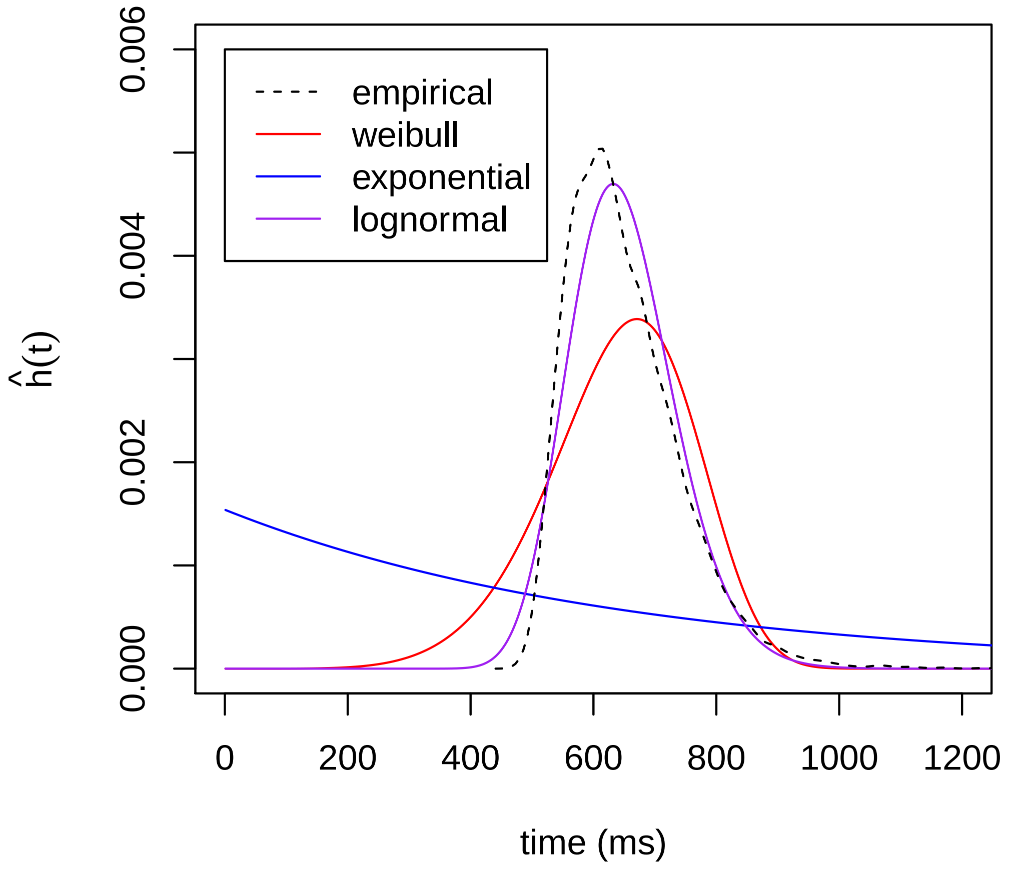
Estimation of survival function

- Parametric methods assume that the survival curve fits a specific distribution
- Typical distributions:
  - exponential
    (constant hazard rate)
  - Weibull
    (monotonically increasing or decreasing hazard rate)
  - lognormal
    (increasing and then decreasing hazard rate)

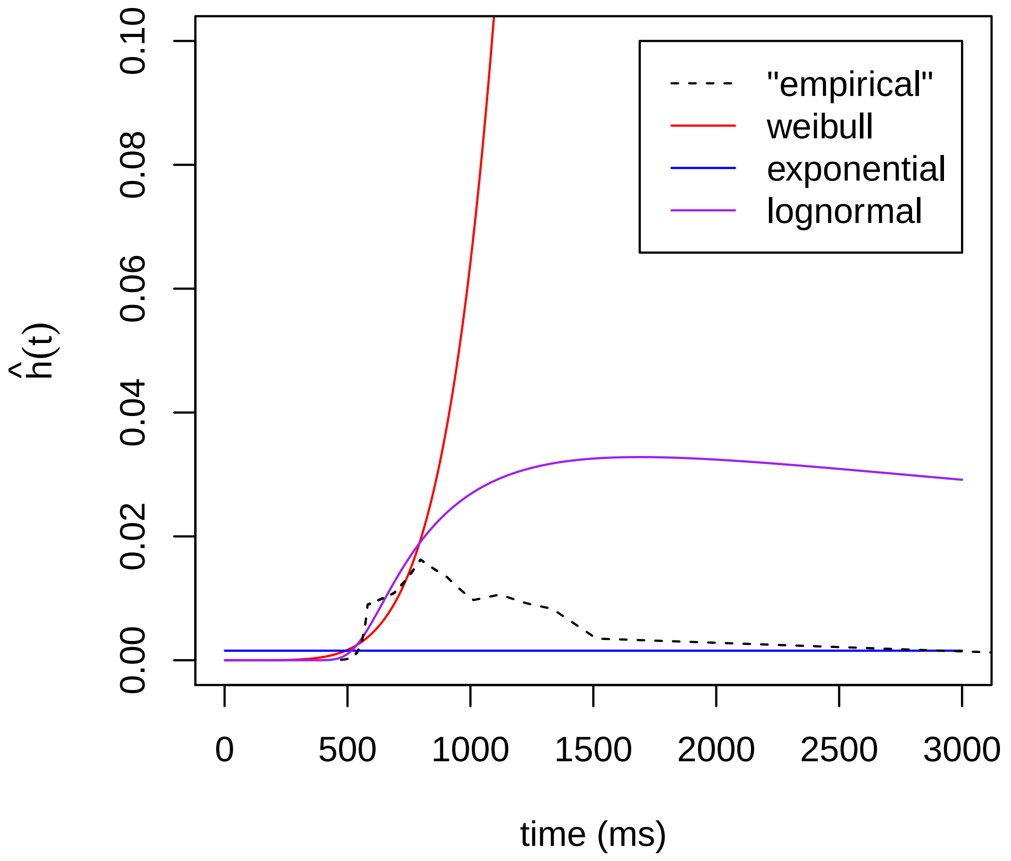
Survival function estimation



Probability density function estimation



Hazard function estimation



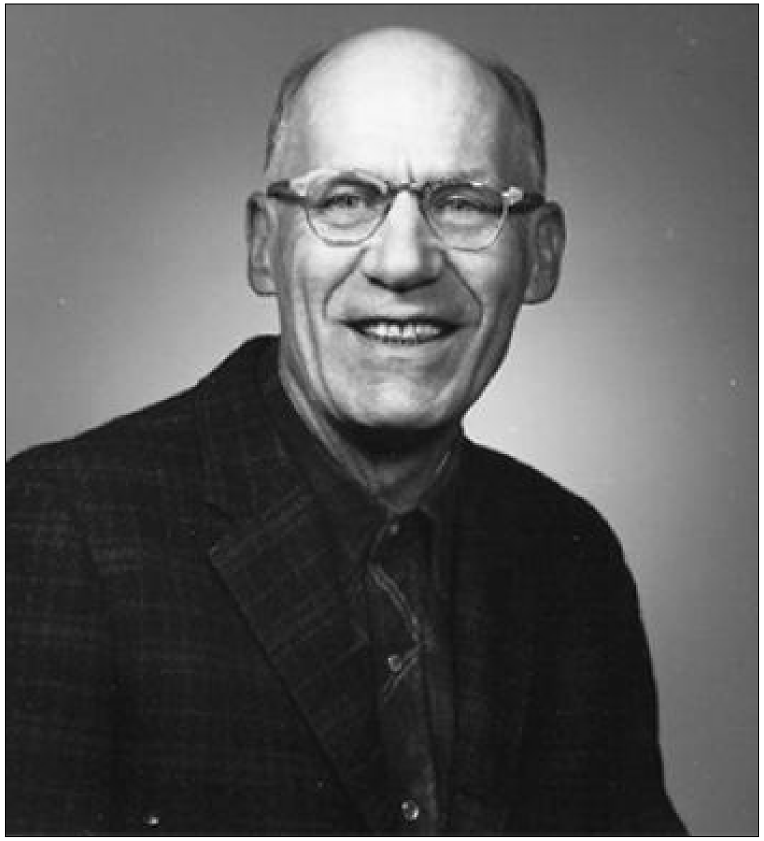
Estimation of survival functions

What if the data do not fit a theoretical distribution?

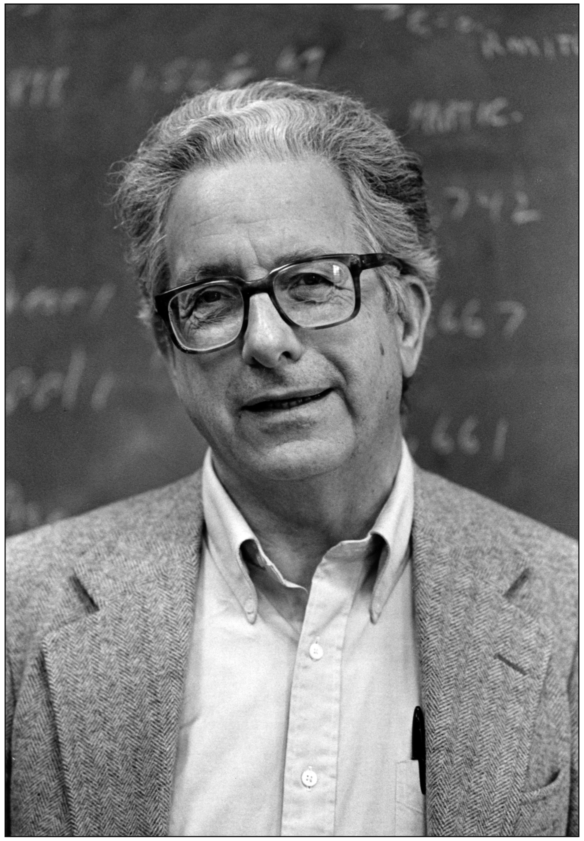
Estimation of survival functions

- Non-parametric methods do not make any assumptions about the underlying distribution
- More common in survival analysis than in many other areas of statistics
- Non-parametric methods:
  - life tables (survival)
  - Kaplan-Meier estimate (survival)
  - Nelson-Aalen estimator (cumulative hazard)

Kaplan-Meier estimate



Kaplan-Meier estimate



Censoring

- Censoring is a key concept in survival analysis
- Data are censored when only partial information is available for an observation
- Right censoring is most common in linguistic applications
- Examples of right censoring:
  - words still exist in the language
  - time outs in reaction time experiments
  - . . .

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

What is $S(630)$?

Kaplan-Meier estimate

word	**rt**	**status**
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$$S(630) = P(T > 630) = 3/5$$

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

What is $S(750)$?

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

We don’t know if word 3 was responded to before 700 ms

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$S(750) = 1/5$? $S(750) = 2/5$?

Neither!

Kaplan-Meier estimate

- The Kaplan-Meier estimate of the survival function is defined as:

$$\hat{S}(t) = \prod_{i:\, t_i \leq t} \left(1 - \frac{e_i}{n_i}\right)$$

  where t_i is a time at which at least one event happened,
  e_i is the number of events that happened at time t_i, and
  n_i is the number of observations at risk at time t_i

- Observations are at risk at time t_i when:
  - no event occurred prior to time t_i
  - no censoring took place before time t_i

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$$S(530) = 1 - 1/5 = 4/5$$

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Kaplan-Meier estimate

word	**rt**	**status**
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$$S(610) = 4/5 * (1 - 1/4) = 4/5 * 3/4 = 3/5$$

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$$S(640) = 3/5 * (1 - 0/3) = 3/5$$

Kaplan-Meier estimate

word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$$S(730) = 3/5 * (1 - 1/2) = 3/5 * 1/2 = 3/10$$

Kaplan-Meier estimate

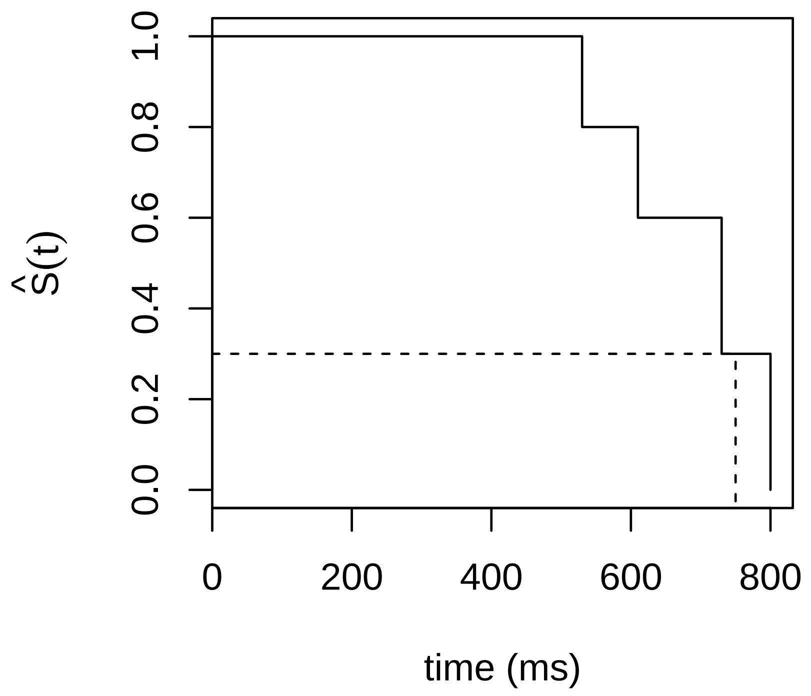
word	rt	status
word 1	530	1
word 2	610	1
word 3	640	0
word 4	730	1
word 5	800	1

$$S(750) = S(730) = 3/10$$

Kaplan-Meier estimate

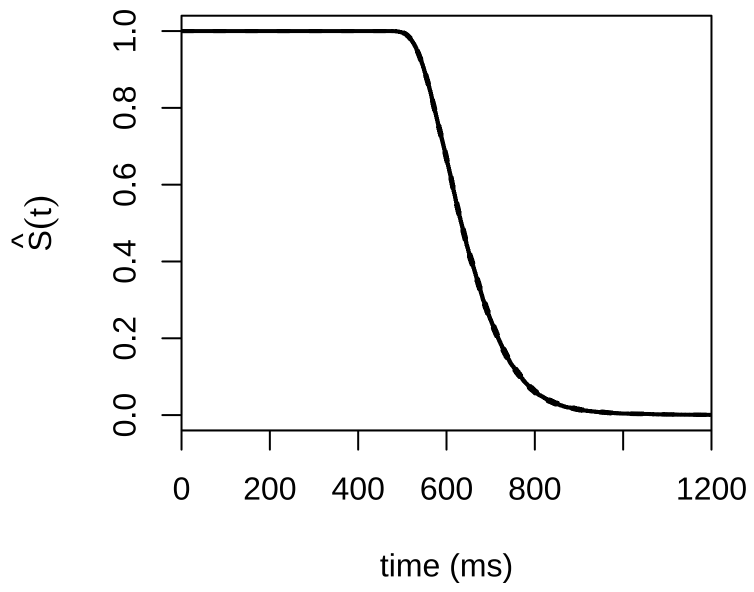
```r
dat = data.frame("word" = paste("word", 1:5),
                 "t" = c(530, 610, 640, 730, 800),
                 "status" = c(1, 1, 0, 1, 1))
# Kaplan-Meier estimate
km = survfit(Surv(t, status) ~ 1, data = dat)

# Plot
plot(km, xlab = "time (ms)", ylab = expression(hat(S)(t)))
```

Kaplan-Meier estimate

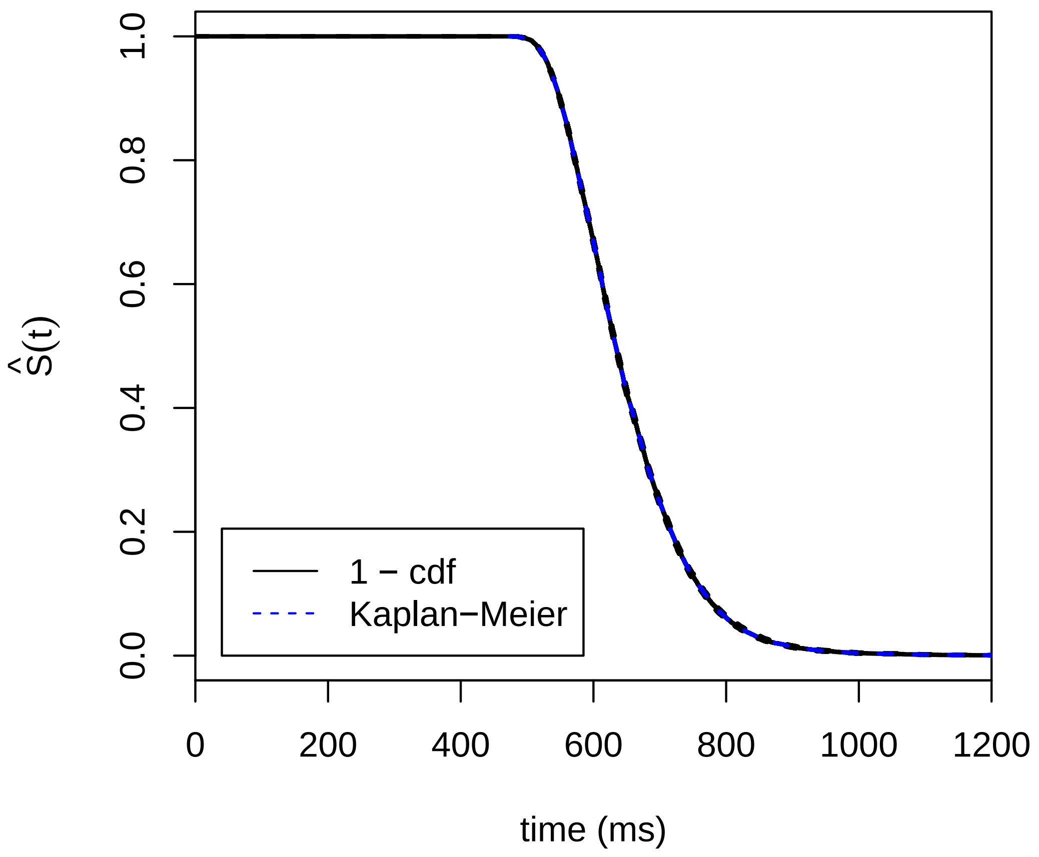


Kaplan-Meier estimate

```
# Kaplan-Meier estimate
km = survfit(Surv(rt, status) ~ 1, data = blp)
# Plot
plot(km, xlab = "time (ms)", ylab = expression(hat(S)(t)),
     xlim = c(0, 1200),lwd=2)
```



Kaplan-Meier estimate



Survival analysis

- Typically, we are interested in whether or not survival curves differ as a function of one or more predictors
- Types of predictors:
  - categorical predictors
  - numerical predictors

Survival analysis

- Types of predictors:
  - categorical predictors
  - numerical predictors

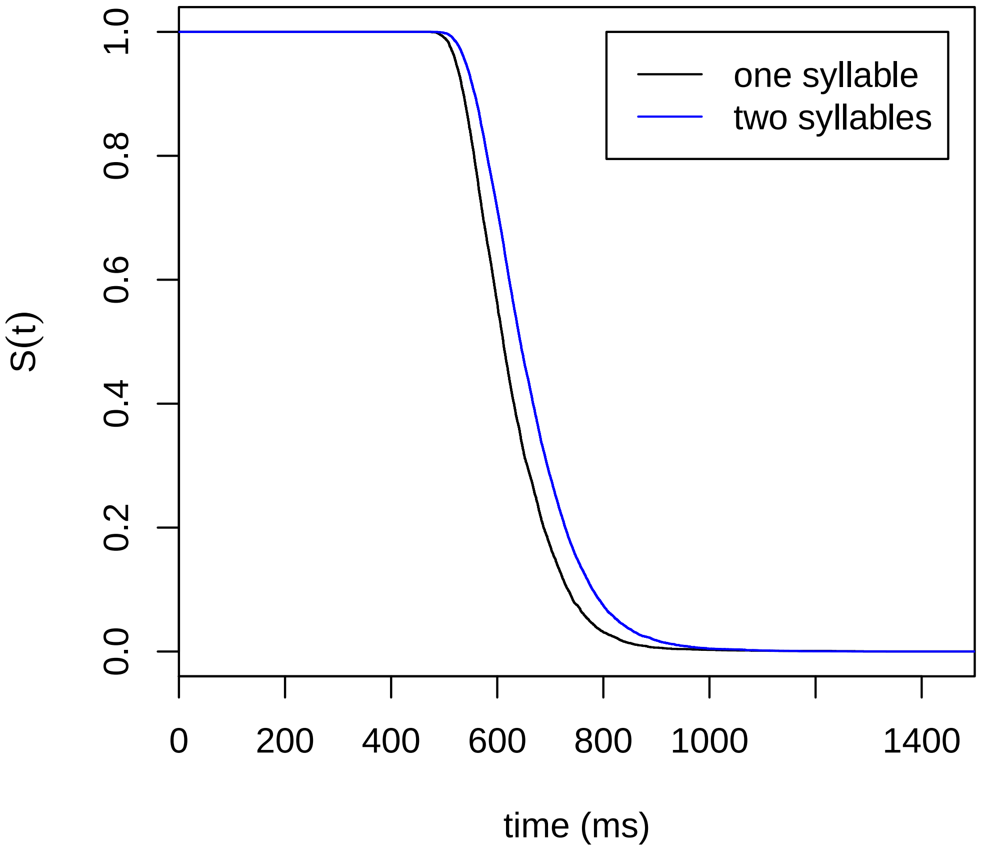
Survival analysis

```
# Show table of number of syllables
table(blp$nsyl)
#
#      1     2     3
#   5267 11908    48
#
# Create a temporary dataframe
tmp = blp[which(blp$nsyl < 3),]
tmp$nsyl = as.factor(tmp$nsyl)
```

Survival analysis

```r
# Kaplan-Meier estimate
km = survfit(Surv(rt, status) ~ nsyl, data = tmp)

# Plot
plot(km, xlab = "time (ms)", ylab = expression(hat(S)(t)),
     xlim = c(0, 1500), col = c("black", "blue"))
```

Survival analysis



Survival analysis

```
# Carry out a log-rank test
surv_diff <- survdiff(Surv(rt, status) ~ nsyl, data = tmp)
surv_diff
# Call:
# survdiff(formula = Surv(rt, status) ~ nsyl, data = tmp)
#
#              N Observed Expected (O-E)^2/E (O-E)^2/V
# nsyl=1  5267     5267     4050       365       483
# nsyl=2 11908    11908    13125       113       483
#
#  Chisq= 483  on 1 degrees of freedom, p= 0
```

Survival analysis

- Types of predictors:
    - categorical predictors
    - numerical predictors

Survival analysis

- Cox proportional hazards model
- Models the hazard function as:

$$\lambda_i(t) = \lambda_0(t)e^{\beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}}$$

where λ_0 is the baseline hazard

Cox proportional hazards model

- Cox model can be reformulated as a linear model for the log-hazard:

$$\log(\lambda_i(t)) = \log(\lambda_0(t)) + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}$$

- No specific distribution is assumed for the baseline hazard (λ_0)
- The Cox model thus is a semi-parametric model

Cox proportional hazards model

- The coefficient β_i describes the difference in log hazard per unit change for predictor X_i
- For $t = 1$ and $X_1 = a$:

$$\log(\lambda_i(t = 1, X_1 = a)) = \log(\lambda_0(1)) + \beta_1 a + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}$$

- For $t = 1$ and $X_1 = a + 1$:

$$\log(\lambda_i(t = 1, X_1 = a+1)) = \log(\lambda_0(1)) + \beta_1 (a+1) + \beta_2 X_{i2} + \ldots + \beta_k X_{ik}$$

Cox proportional hazards model

$$\cancel{\log(\lambda_0(1))} + \beta_1 a \cancel{+ \beta_2 X_{i2} + \ldots + \beta_k X_{ik}}$$

$$\cancel{\log(\lambda_0(1))} + \beta_1 (a + 1) \cancel{+ \beta_2 X_{i2} + \ldots + \beta_k X_{ik}}$$

$$\log(\lambda_i(t = 1, X_1 = a+1)) - \log(\lambda_i(t = 1, X_1 = a)) = \beta_1 (a+1) - \beta_1 a = \beta_1$$

Cox proportional hazards model

- The quantity:

$$\log(\lambda_i(t, X_1 = a + 1)) - \log(\lambda_i(t, X_1 = a))$$

  thus is independent of time

- The log-hazard for $X_1 = a + 1$ is the log-hazard for $X_1 = a$ plus β_1
- This is the proportional hazards assumption

Cox proportional hazards model

- The quantity e^{β_i} is the hazard ratio (HR) for predictor X_i
- Interpretation:
  - HR = 1 $\rightarrow$ no effect
  - HR < 1 $\rightarrow$ reduction in hazard
  - HR > 1 $\rightarrow$ increase in hazard
- A HR of 0.81 means that there is a reduction of 19% in hazard rate for every unit change in the predictor

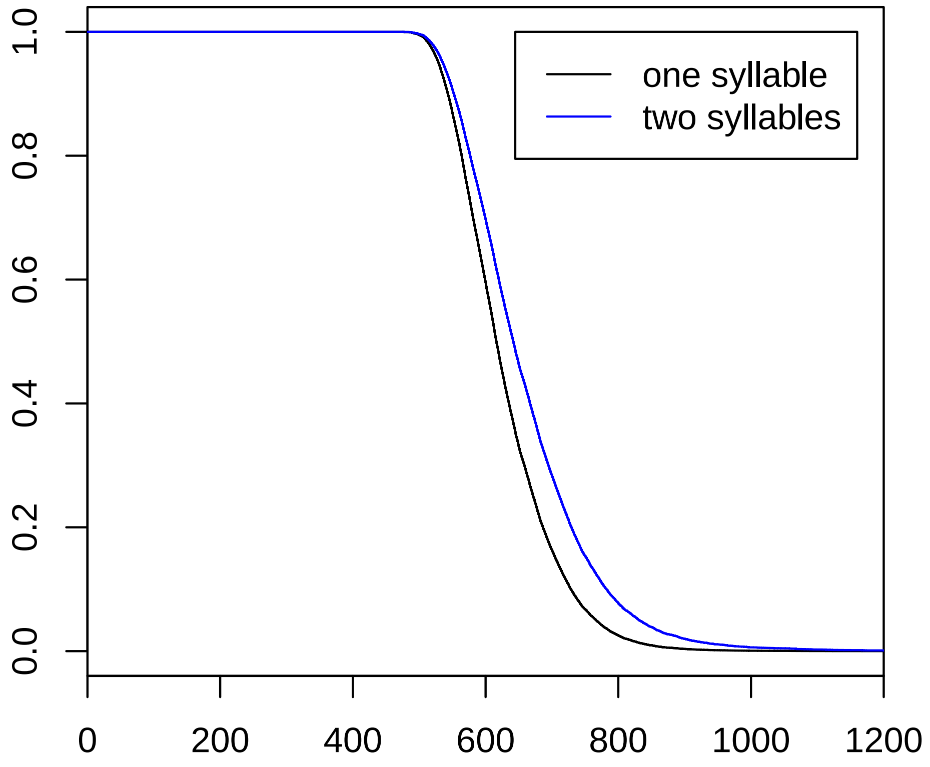
Cox proportional hazards model

```
# Fit a Cox model
cox.nsyl = coxph(Surv(rt, status) ~ nsyl, data = tmp)
cox.nsyl
# Call:
# coxph(formula = Surv(rt, status) ~ nsyl, data = tmp)
#
#           coef exp(coef) se(coef)     z      p
# nsyl2 -0.3637    0.6951   0.0166 -21.9 <2e-16
#
# Likelihood ratio test=455  on 1 df, p=0
# n= 17175, number of events= 17175
```

Cox proportional hazards model

```r
# Make a new dataframe
newdat = data.frame("nsyl" = c("1", "2"))

# Fit Cox model to new data
fit = survfit(cox.nsyl, newdata = newdat)

# Plot
plot(fit, xlim = c(0, 1200), col = c("black", "blue"))
```

Survival analysis



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

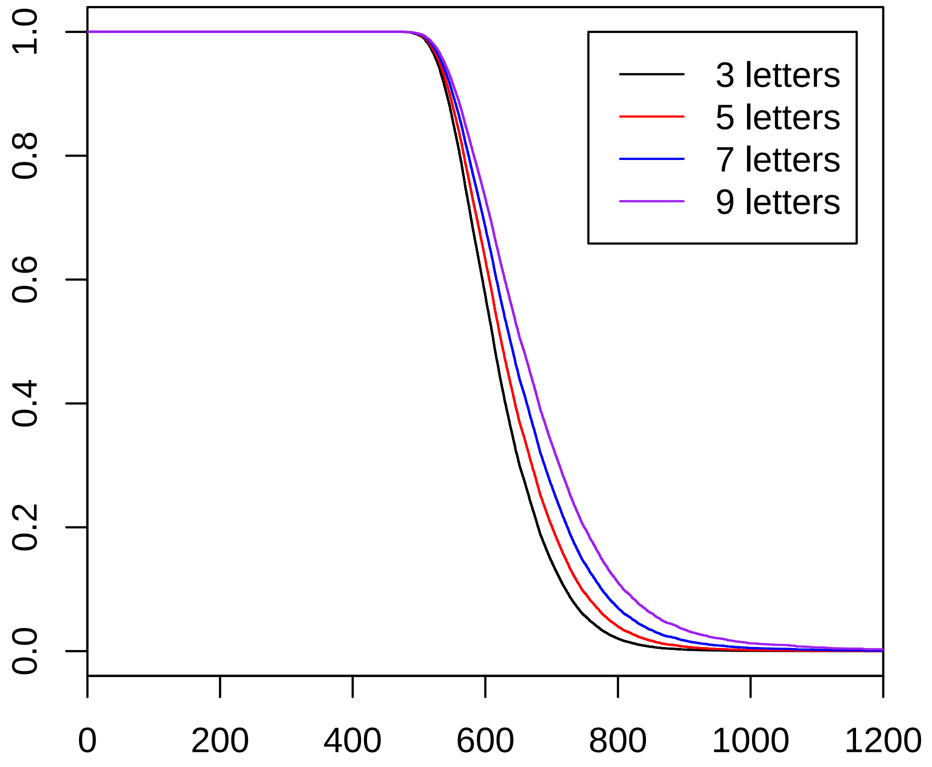
Cox proportional hazards model

```
# Fit a Cox model
cox.length = coxph(Surv(rt, status) ~ length, data = blp)
cox.length
# Call:
# coxph(formula = Surv(rt, status) ~ length, data = blp)
#
#             coef exp(coef) se(coef)     z      p
# length -0.09540   0.90901  0.00508 -18.8 <2e-16
#
# Likelihood ratio test=353  on 1 df, p=0
# n= 17303, number of events= 17303
```

Cox proportional hazards model

```r
# Make a new dataframe
newdat = data.frame("length" = c(3, 5, 7, 9))

# Fit Cox model to new data
fit = survfit(cox.length, newdata = newdat)

# Plot
plot(fit, xlim=c(0, 1200), col = c("black", "red", "blue", "purple"))
```

Cox proportional hazards model



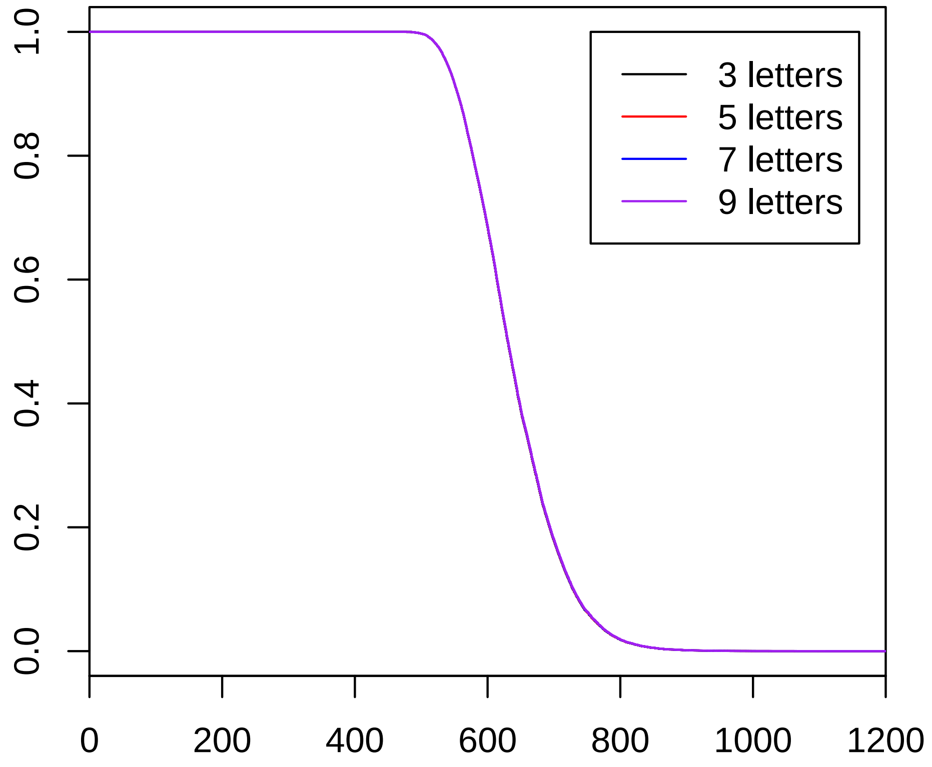
Cox proportional hazards model

```
# Fit a Cox model
cox.mult = coxph(Surv(rt, status) ~ logfrequency + length, data = blp)
cox.mult
# Call:
# coxph(formula = Surv(rt, status) ~ logfrequency + length, data = blp)
#
#                    coef  exp(coef) se(coef)     z       p
# logfrequency  0.45857   1.58181  0.00523 87.72 <2e-16
# length       -0.00107   0.99893  0.00512 -0.21   0.83
#
# Likelihood ratio test=7120  on 2 df, p=0
# n= 17303, number of events= 17303
```

Cox proportional hazards model

```
# Make a new dataframe
newdat = data.frame("length" = c(3, 5, 7, 9),
                    "logfrequency" = mean(blp$logfrequency))

# Fit Cox model to new data
fit = survfit(cox.mult, newdata = newdat)

# Plot
plot(fit, xlim=c(0, 1200), col = c("black", "red", "blue", "purple"))
```

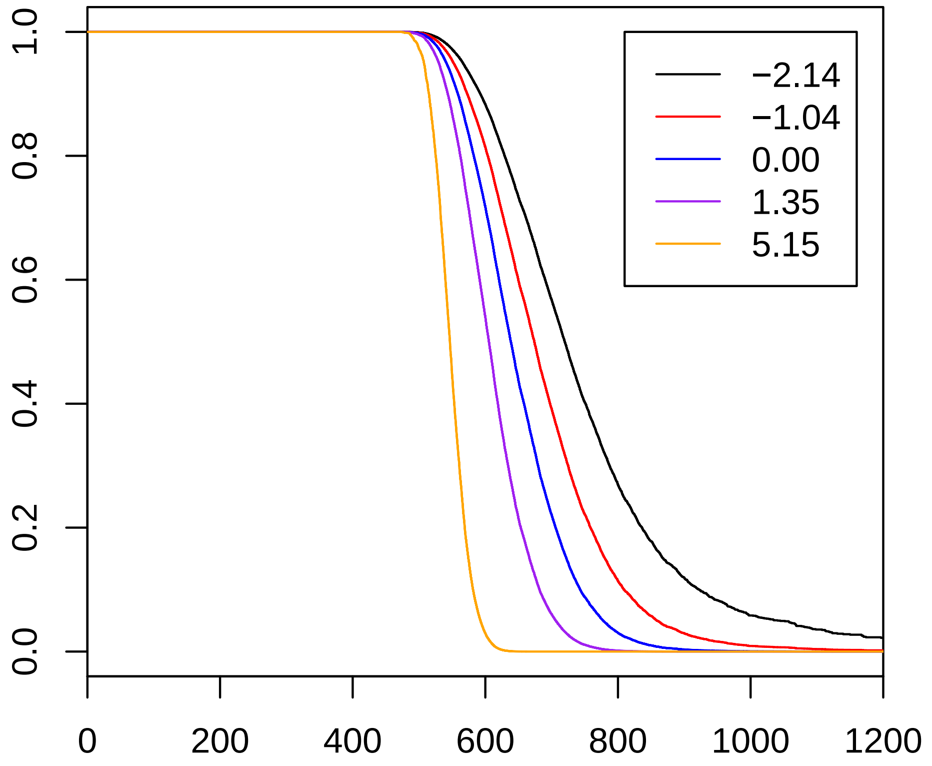
Cox proportional hazards model



Cox proportional hazards model

```r
# Make a new dataframe
newdat = data.frame("length" = mean(blp$length),
                    "logfrequency" = quantile(blp$logfrequency))

# Fit Cox model to new data
fit = survfit(cox.mult, newdata = newdat)

# Plot
plot(fit, xlim=c(0, 1200),
     col = c("black", "red", "blue", "purple", "orange"))
```

Cox proportional hazards model



Cox proportional hazards model

- The proportional hazards assumption is a strong assumption
- Test if it is reasonable

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Cox proportional hazards model

- Inspect residuals
- There is no straightforward equivalent of residuals in standard linear regression in the Cox model
- Schoenfeld residuals:
    - set of residuals for each observation
    - one residual per predictor
    - observed predictor value minus expected predictor value

Cox proportional hazards model

- Schoenfeld residuals should be independent of time
- Correlate the (scaled) Schoenfeld residuals with time to find out if the proportional hazards assumption holds

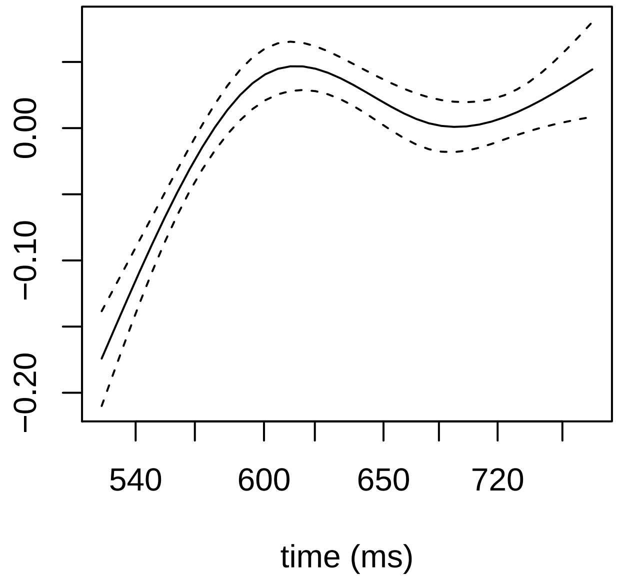
Cox proportional hazards model

```
# Test proportional hazards assumption
ph.test = cox.zph(cox.mult)
ph.test
#                  rho  chisq        p
# logfrequency -0.2755 1048.6 0.00e+00
# length        0.0476   37.6 8.88e-10
# GLOBAL            NA 1269.8 0.00e+00
```

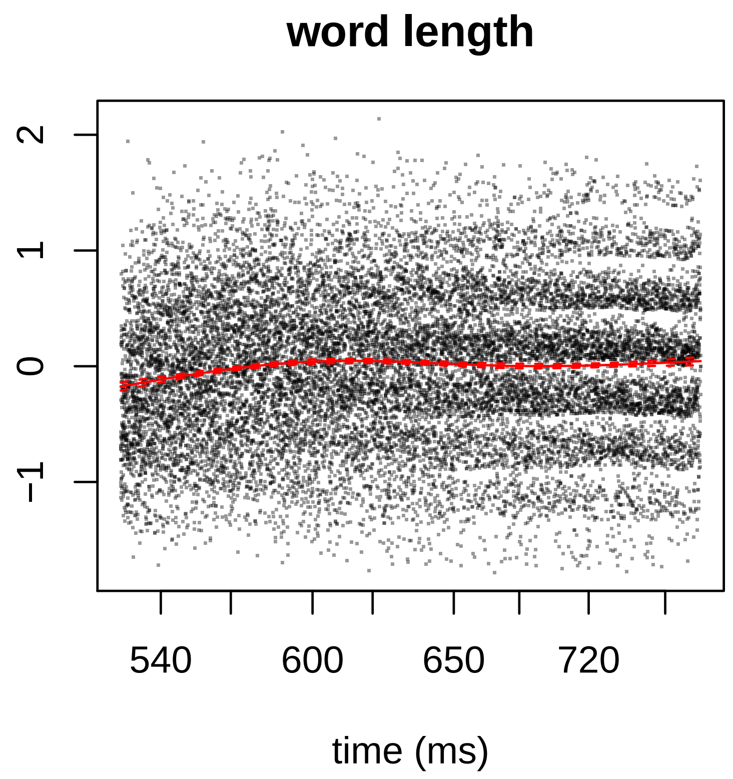
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Cox proportional hazards model

```r
# Plot scaled Schoenfeld residuals
plot(ph.test, var = "length", resid = FALSE,
     xlab = "time (ms)", ylab = "Schoenfeld residuals")
```



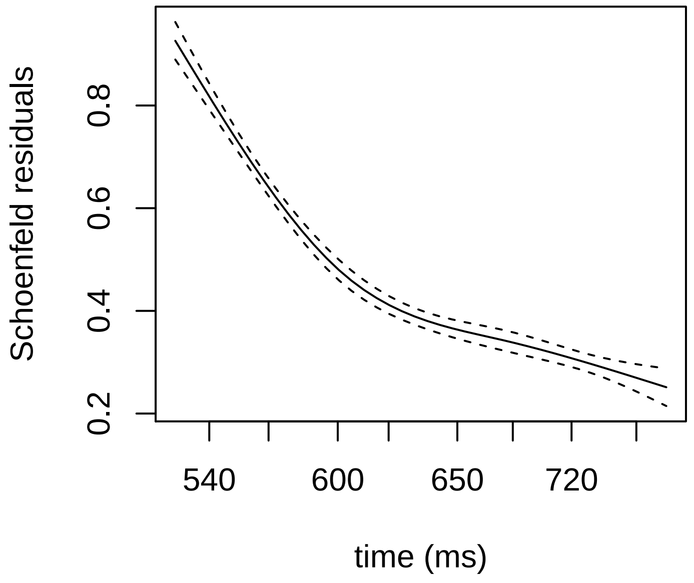
Cox proportional hazards model



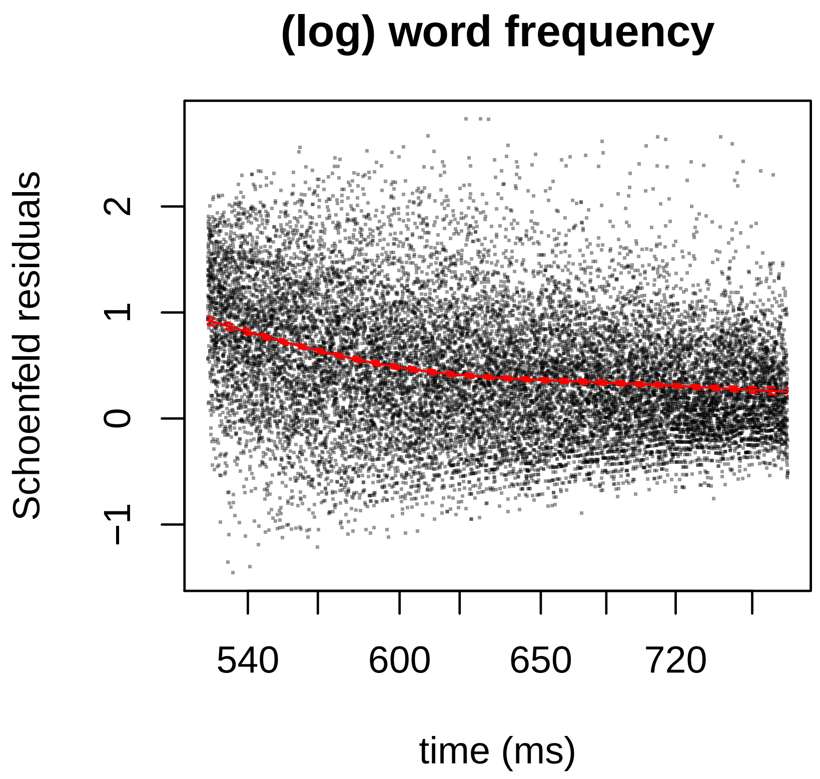
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Cox proportional hazards model

```r
# Plot scaled Schoenfeld residuals
plot(ph.test, var = "logfrequency", resid = FALSE,
     xlab = "time (ms)", ylab = "Schoenfeld residuals")
```



Cox proportional hazards model



Cox proportional hazards model

- Limitations of the Cox model:
    - Violations of the proportional hazards assumptions are common
    - The relation between the log-hazard and predictors need not be linear
    - Random effects are not available
- However, workarounds do exist and extensions of the Cox model have been developed

Survival analysis

PAMM

- Piece-wise exponential generalized additive model (PAMM)
- Advantages:
  - non-linear predictor effects
  - non-linear development over time, no proportional hazards assumption
  - random effects
- Combination of the piece-wise exponential model (PEM) and the generalized-additive mixed-effect model (GAMM)

PEM

- Idea behind the piece-wise exponential model (PEM):
  - partition the period $(0, t_{max})$ into a number of intervals
  - cut-points k are borders between time intervals
  - assume that the hazard rate is constant in each interval (hence piece-wise exponential)

PEM

- Cox proportional hazards model:

$$\lambda_i(t) = \lambda_0(t) e^{\beta^\top X}$$

  where λ_0 is the baseline hazard

- Piece-wise exponential model:

$$\lambda_i(t) = \lambda_j e^{\beta^\top X} \quad \forall\, t \in (k_{j-1}, k_j]$$

  where λ_j is the baseline hazard for interval j and k_j is the cut-point for time interval j

PEM

- The maximum likelihood of the PEM is proportional to the maximum likelihood of a Poisson regression model (if the data are in the right format)
- Both models yield the same β estimates
- We can therefore model a PEM through a Poisson regression model

PEM

```
# Load the pammtools library
library(pammtools)

# Define cut_points
cut_points = seq(0,3000,length.out=11)
cut_points
#  [1]    0  300  600  900 1200 1500 1800 2100 2400 2700 3000
```

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

PEM

```
# Put data in correct format
ped_blp = split_data(Surv(rt, status) ~ word, data = blp,
                     cut = cut_points)


# Inspect
head(ped_blp, n = 5)
#   id tstart tend  interval   offset ped_status  word
# 1  1      0  300   (0,300] 5.703782          0 aback
# 2  1    300  600 (300,600] 5.703782          0 aback
# 3  1    600  900 (600,900] 4.854956          1 aback
# 4  2      0  300   (0,300] 5.703782          0 abbey
# 5  2    300  600 (300,600] 5.703782          0 abbey
#
# RT for "aback" is 728.3750 ms
# Offset:
600 + exp(4.854956)
# [1] 728.375
```

PEM

```r
# Run PEM
pem = glm(ped_status ~ interval, data = ped_blp, offset = offset,
          family = "poisson")
```

PEM

```
# Create dataframe with interval information
int_blp = int_info(ped_blp)

# Add cumulative hazard
int_blp = add_cumu_hazard(int_blp, pem)

# Show dimensions
dim(int_blp)
# [1] 10  8
```

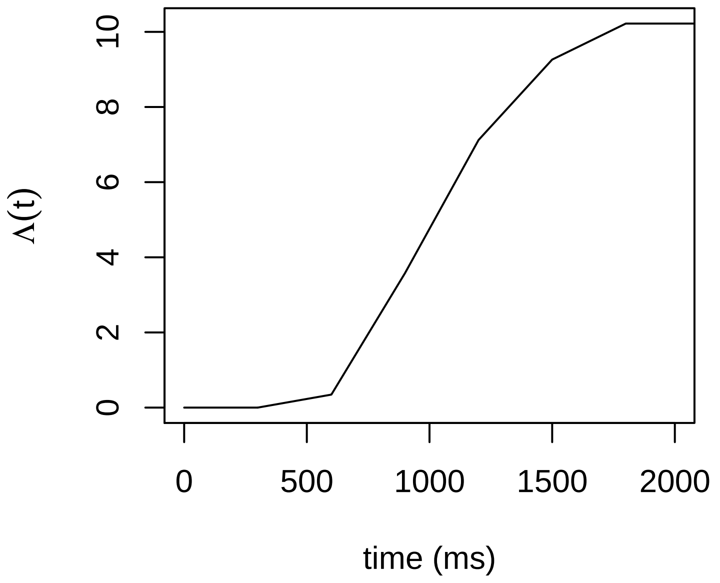
PEM

```
# Show head
head(int_blp)
#   tstart tend intlen intmid    interval  cumu_hazard   cumu_lower
# 1      0  300    300    150     (0,300] 4.139938e-09 2.351154e-71
# 2    300  600    300    450   (300,600] 3.454347e-01 3.364236e-01
# 3    600  900    300    750   (600,900] 3.577462e+00 3.508279e+00
# 4    900 1200    300   1050  (900,1200] 7.122409e+00 6.625552e+00
# 5   1200 1500    300   1350 (1200,1500] 9.261995e+00 7.680515e+00
# 6   1500 1800    300   1650 (1500,1800] 1.022046e+01 7.810229e+00
#     cumu_upper
# 1 7.289648e+53
# 2 7.289648e+53
# 3 7.289648e+53
# 4 7.289648e+53
# 5 7.289648e+53
# 6 7.289648e+53
```

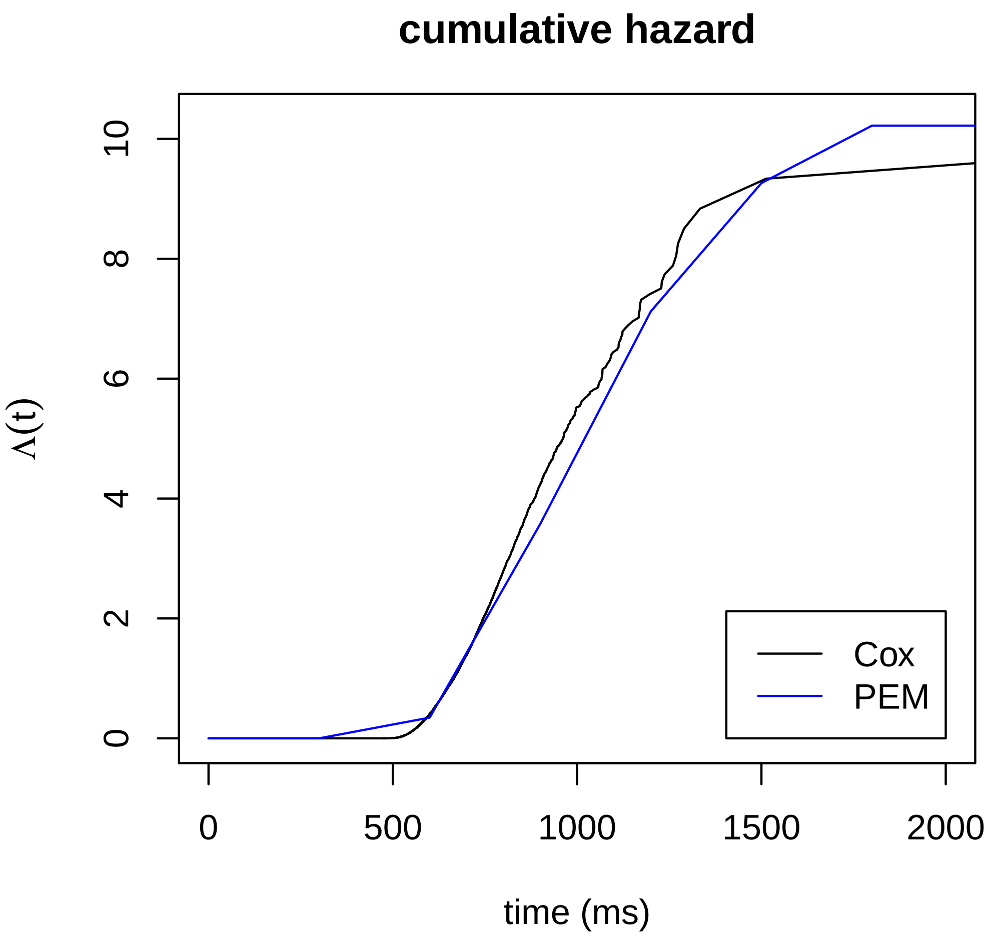
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

PEM

```
plot(int_blp$tend, int_blp$cumu_hazard, type = "l",
     xlab = "time (ms)", ylab = expression(Lambda(t)),
     xlim = c(0, 2000))
segments(0, 0, 300, 0)
```



PEM

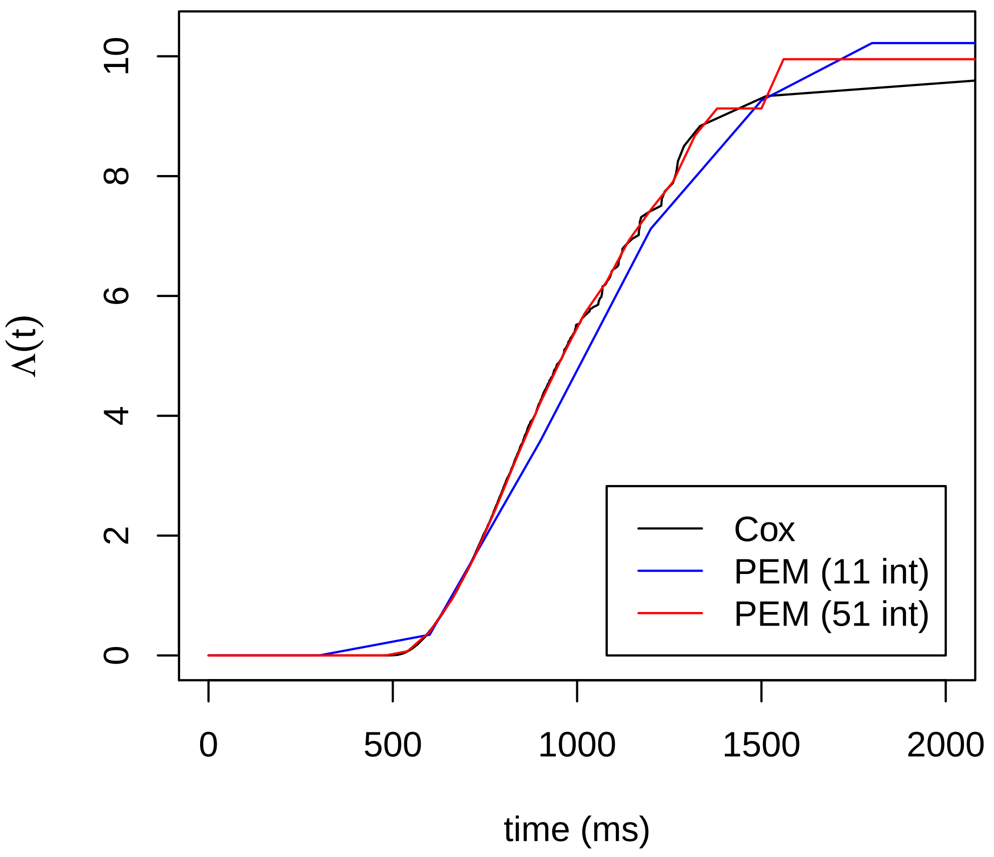

cumulative hazard
Λ(t)
0
2
4
6
8
10
0
500
1000
1500
2000
time (ms)
Cox
PEM

PEM

```
# Define more cut-points
cut_points = seq(0,3000,length.out=51)
cut_points
#  [1]    0   60  120  180  240  300  360  420  480  540  600  660
# [13]  720  780  840  900  960 1020 1080 1140 1200 1260 1320 1380
# [25] 1440 1500 1560 1620 1680 1740 1800 1860 1920 1980 2040 2100
# [37] 2160 2220 2280 2340 2400 2460 2520 2580 2640 2700 2760 2820
# [49] 2880 2940 3000
#
# Repeat other steps and run new Poisson regression model...
```

PEM

cumulative hazard



PEM

- PEMs were temporarily popular when computational implementations of Poisson regression were more available than dedicated techniques for survival analysis
- Criticism:
  - the choice of the number of intervals and the cut-points is arbitrary
  - too few intervals $\rightarrow$ inaccurate estimation of the hazard rate
  - too many interval $\rightarrow$ unstable estimates and overfitting
- Dedicated techniques such as the Cox proportional hazards model dominate the literature today

PAMM

- Piece-wise exponential generalized additive mixed model:

$$\lambda_i(t) = \exp\left(f_0(t_j) + \sum_{k=1}^{p} f_k(x_{i,k}, t_j) + b_{\ell_i} \right) \quad \forall\, t \in (k_{j-1}, k_j]$$

  where $f_0(t_j)$ is the log-baseline hazard for interval j, $f_k(x_i, t_j)$ are potentially non-linear and non-linear time-varying effects of predictors X, and b_ℓ are random effects associated with group $\ell = 1, \ldots, L$ to which observation i belongs

- As was the case for the Cox model, this can be reformulated as a linear model for the log-hazard

PAMM

- Piece-wise exponential generalized additive mixed model:

$$\lambda_i(t) = \exp\left(f_0(t_j) + \sum_{k=1}^{p} f_k(x_{i,k}, t_j) + b_{\ell_i} \right) \quad \forall\, t \in (k_{j-1}, k_j]$$

- The baseline hazard $f_0(t_j)$ is modeled as a regression spline over time
- Nonetheless, it remains piece-wise constant
- A large number of intervals is chosen
- Overfitting and instability are prevented by penalization

PAMM

- Piece-wise exponential generalized additive mixed model:

$$\lambda_i(t) = \exp\left(f_0(t_j) + \sum_{k=1}^{p} f_k(x_{i,k}, t_j) + b_{\ell_i} \right) \quad \forall\, t \in (\kappa_{j-1}, \kappa_j]$$

- The term $\sum_{k=1}^{p} f_k(x_i, t_j)$ can represent a variety of effect types:
  - linear, time-constant effects (x_i, β_k)
  - non-linear, time-constant effects $(f_k(x_{i,k}))$
  - non-linear, time-varying effects $(f_k(x_{i,k}, t_j))$

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

PAMM

- Piece-wise exponential generalized additive mixed model:

$$\lambda_i(t) = \exp\left(f_0(t_j) + \sum_{k=1}^{p} f_k(x_{i,k}, t_j) + b_{\ell_i} \right) \quad \forall\, t \in (\kappa_{j-1}, \kappa_j]$$

- The term b_{ℓ_i} models random intercepts for group $\ell = 1, \ldots, L$ to which observation i belongs
- More complex random effects can be modeled as well

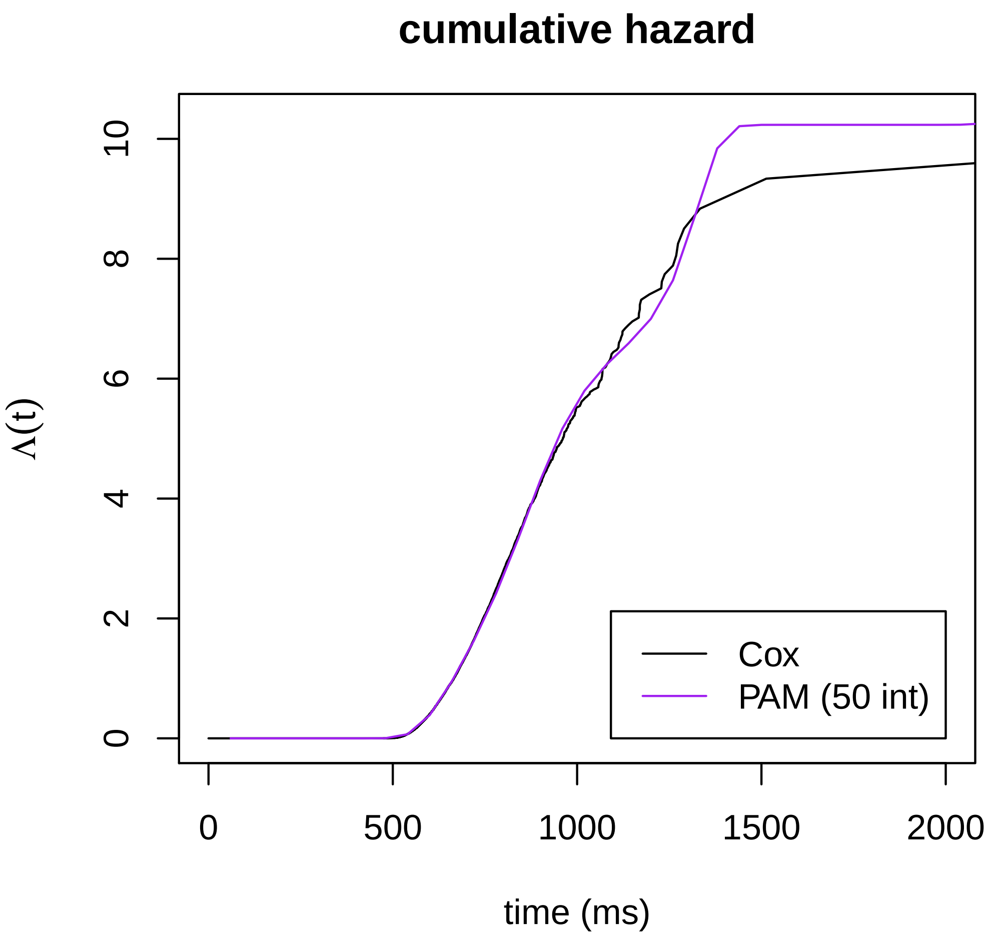
PAMM

```
# Prepare data
cut_points = seq(0,3000,length.out=51)
ped_blp = split_data(Surv(rt, status) ~ word, data = blp,
                     cut = cut_points)

# Get intervals
int_blp = int_info(ped_blp)

# Load library
library(mgcv)

# Run PAMM
pam = gam(ped_status ~ s(tend), data = ped_blp, offset = offset,
          family = "poisson")
```

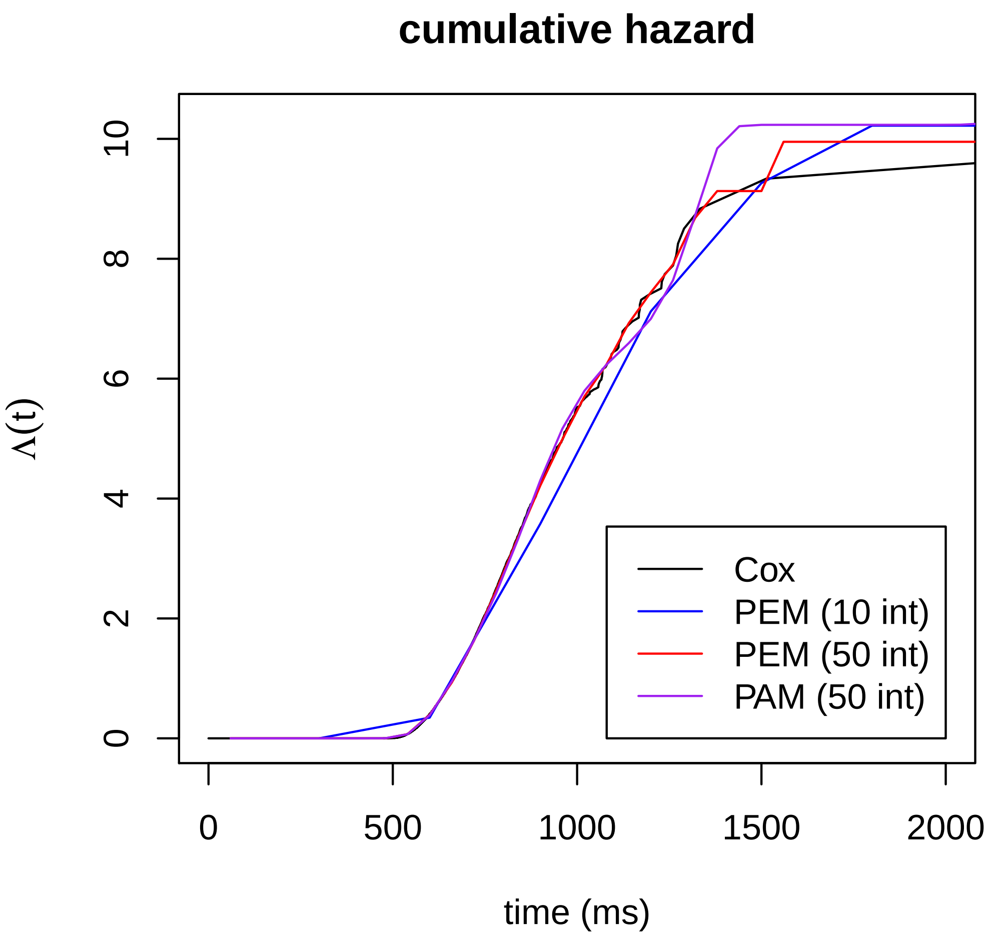
PAMM

```
# Extract hazard
int_blp$pamhaz = predict(pam, newdata = int_blp, type = "response")

# Convert to cumulative hazard
int_blp$pamch = cumsum(int_blp$pamhaz * int_blp$intlen)

# Show head
head(int_blp)
#   tstart tend intlen intmid  interval       pamhaz        pamch
# 1      0   60     60     30    (0,60] 2.220446e-16 1.332268e-14
# 2     60  120     60     90  (60,120] 2.220446e-16 2.664535e-14
# 3    120  180     60    150 (120,180] 2.220446e-16 3.996803e-14
# 4    180  240     60    210 (180,240] 2.220446e-16 5.329071e-14
# 5    240  300     60    270 (240,300] 2.220446e-16 6.661338e-14
# 6    300  360     60    330 (300,360] 5.669334e-12 3.402267e-10
```

PAMM


cumulative hazard
Λ(t)
0
2
4
6
8
10
0
500
1000
1500
2000
time (ms)
Cox
PAM (50 int)

PAMM


cumulative hazard
Λ(t)
0
2
4
6
8
10
0
500
1000
1500
2000
time (ms)
Cox
PEM (10 int)
PEM (50 int)
PAM (50 int)

PAMM

```
# Get quantiles of rt distribution
quantile(blp$rt, seq(0.975, 1, by = 0.005))
#      97.5%       98%     98.5%       99%     99.5%      100%
#   860.0596  875.4967  896.2844  927.6559  981.6830 3705.3500
#
# Define cut-points
cut_points = as.numeric(quantile(blp$rt[which(blp$rt <= 900)],
                                 seq(0, 1, by = 0.02)))

# Put data in right format
ped_blp = split_data(Surv(rt, status) ~ ., data = blp,
                     cut = cut_points)

# Show dimensions
dim(ped_blp)
# [1] 464660     13
```

PAMM

```
# Show head
head(ped_blp, n = 4)
#   id   tstart     tend                         interval
# 1  1   0.0000 474.0556               (0,474.055555555556]
# 2  1 474.0556 518.1579 (474.055555555556,518.157894736842]
# 3  1 518.1579 529.0250            (518.157894736842,529.025]
# 4  1 529.0250 536.6571            (529.025,536.657142857143]
#     offset ped_status  word logfrequency length  logold20
# 1 6.161325          0 aback    -1.223835      5 0.6151856
# 2 3.786513          0 aback    -1.223835      5 0.6151856
# 3 2.385740          0 aback    -1.223835      5 0.6151856
# 4 2.032369          0 aback    -1.223835      5 0.6151856
#   summedbigramfrequency   sem20 nsyl
# 1               1132695 11.9144    2
# 2               1132695 11.9144    2
# 3               1132695 11.9144    2
# 4               1132695 11.9144    2
```

PAMM

- How to model non-linear time-varying effects?
- Tensor product (`te(time, predictor)`)
- Split out in three components for increased interpretability:
  - main effect of time (`s(tend)`)
  - main effect of predictor (`s(predictor)`)
  - partial interaction between time and predictor (`ti(time, predictor)`)
- Time-varying effects are still assumed to be piece-wise constant in each interval

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

PAMM

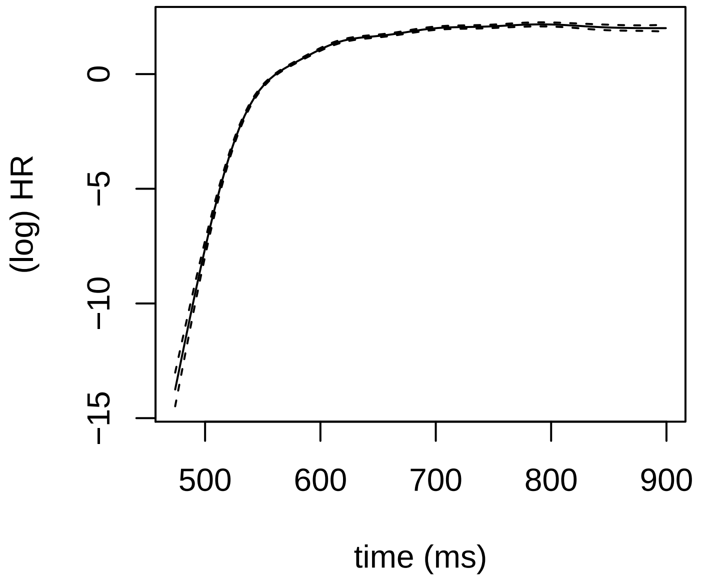
```r
# Run PAMM
pam.mult = gam(ped_status ~ s(tend) +
               s(logfrequency, k = 4) +
               ti(tend, logfrequency, k = c(4, 4)) +
               s(length, k = 4) +
               ti(tend, length,k = c(4,4)) +
               s(sem20, k = 4) +
               ti(tend, sem20, k = c(4,4)) +
               s(summedbigramfrequency, k = 4) +
               ti(tend, summedbigramfrequency,k = c(4, 4)) +
               s(logold20, k = 4) +
               ti(tend, logold20, k = c(4,4)),
               data = ped_blp, offset = offset, family = "poisson")
```

PAMM

```
# Show results
round(summary(pam.mult)$s.table,3)
#                                  edf Ref.df    Chi.sq p-value
# s(tend)                        8.991  9.000 4771.467   0.000
# s(logfrequency)                2.914  2.985 3898.210   0.000
# ti(tend,logfrequency)          5.030  5.765 1205.448   0.000
# s(length)                      1.003  1.007    7.094   0.008
# ti(tend,length)                4.651  5.360  138.960   0.000
# s(sem20)                       2.973  2.998   49.753   0.000
# ti(tend,sem20)                 2.581  2.916   25.662   0.000
# s(summedbigramfrequency)       1.028  1.055   89.744   0.000
# ti(tend,summedbigramfrequency) 2.529  2.827   26.875   0.000
# s(logold20)                    2.974  2.997  105.954   0.000
# ti(tend,logold20)              7.439  8.174   62.688   0.000
#
summary(pam.mult)$dev.expl
# [1] 0.3429811
```

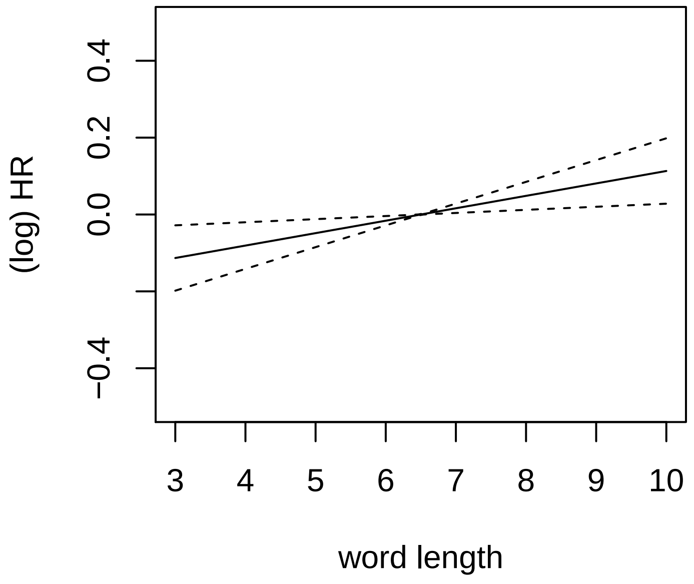
PAMM

```r
# Plot main effect of time
plot.gam(pam.mult, select = 1, xlab = "time (ms)",
         ylab = "(log) HR")
```



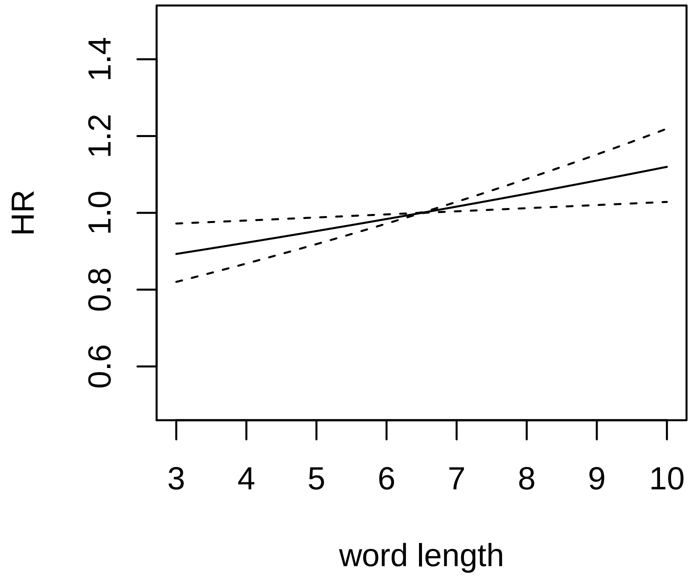
PAMM

```r
# Plot main effect of word length
plot.gam(pam.mult, select = 4, xlab = "word length",
         ylab = "(log) HR", ylim = c(-0.5, 0.5))
```



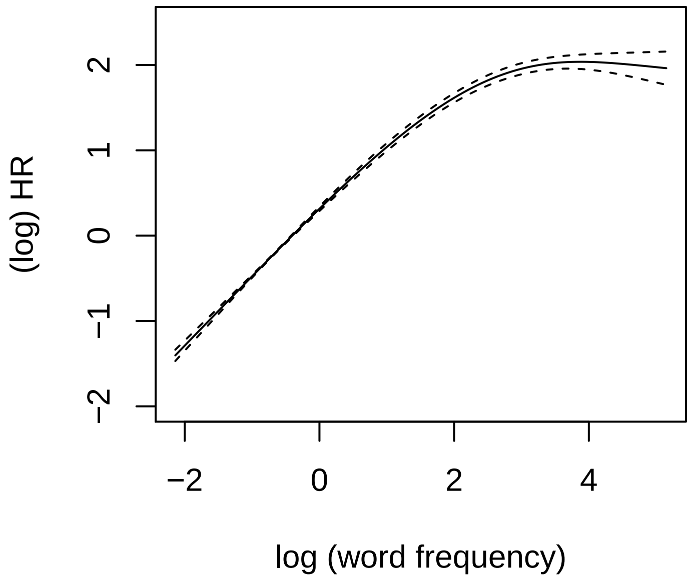
PAMM

```r
# Plot main effect of word length on HR scale
plot.gam(pam.mult, select = 4, trans = exp, xlab = "word length",
         ylab = "HR", ylim = c(0.5, 1.5))
```



PAMM

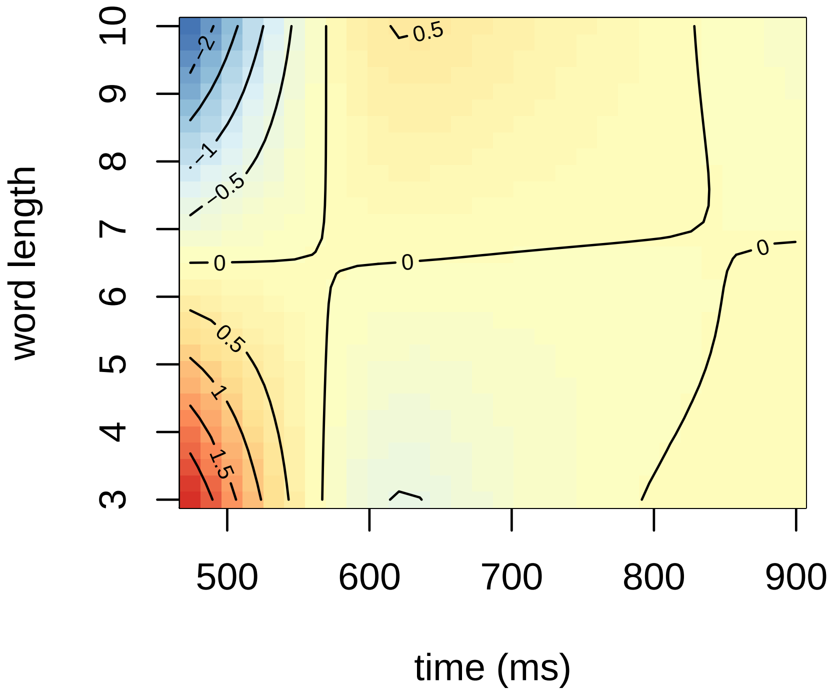
```r
# Plot main effect of word frequency
plot.gam(pam.mult, select = 2, xlab = "log (word frequency)",
         ylab = "(log) HR", ylim = c(-2, 2.5))
```



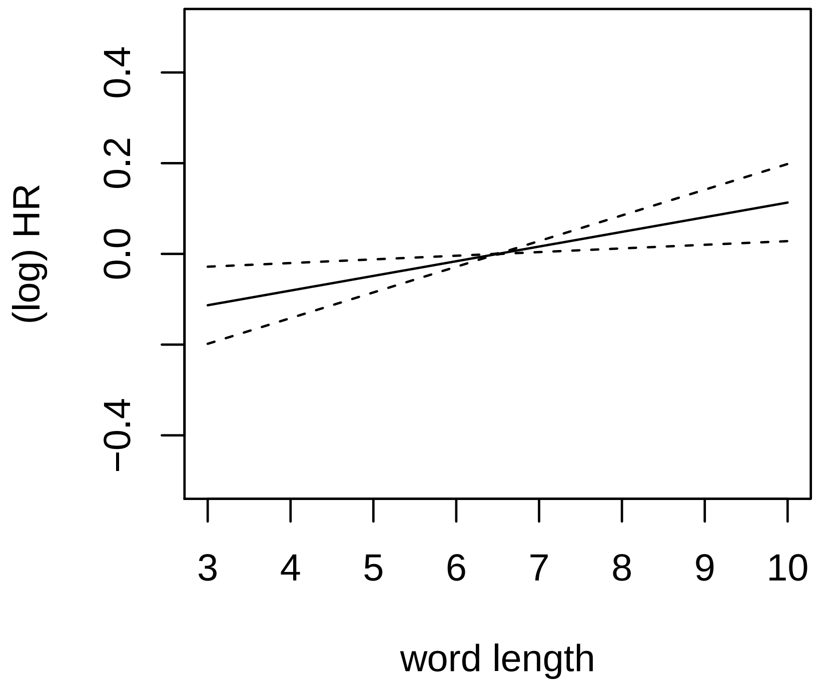
PAMM

```
# Load libraries
library(itsadug)
library(RColorBrewer)

# Define color palette
palette = colorRampPalette(rev(brewer.pal(n = 7,
              name = "RdYlBu")))(500)

# Plot partial effect of interaction of time and length
pvisgam(pam.mult, view = c("tend", "length"), color = palette,
        xlab = "time (ms)", ylab = "word length",
        print.summary = FALSE)
```

PAMM


word length
time (ms)
500
600
700
800
900
3
4
5
6
7
8
9
10
-2
-1
-0.5
0
0.5
1
1.5
0.5
0
0

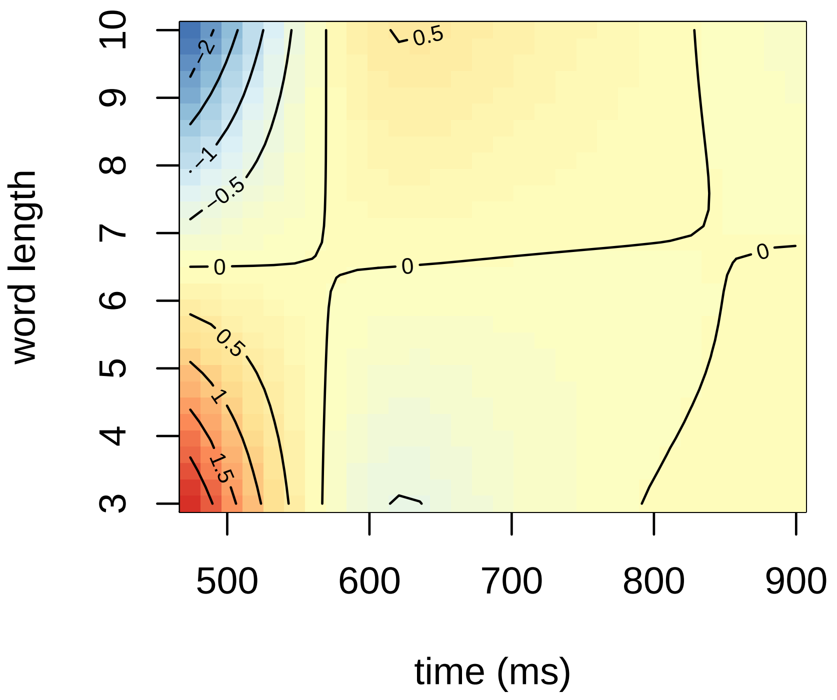
PAMM



PAMM

+

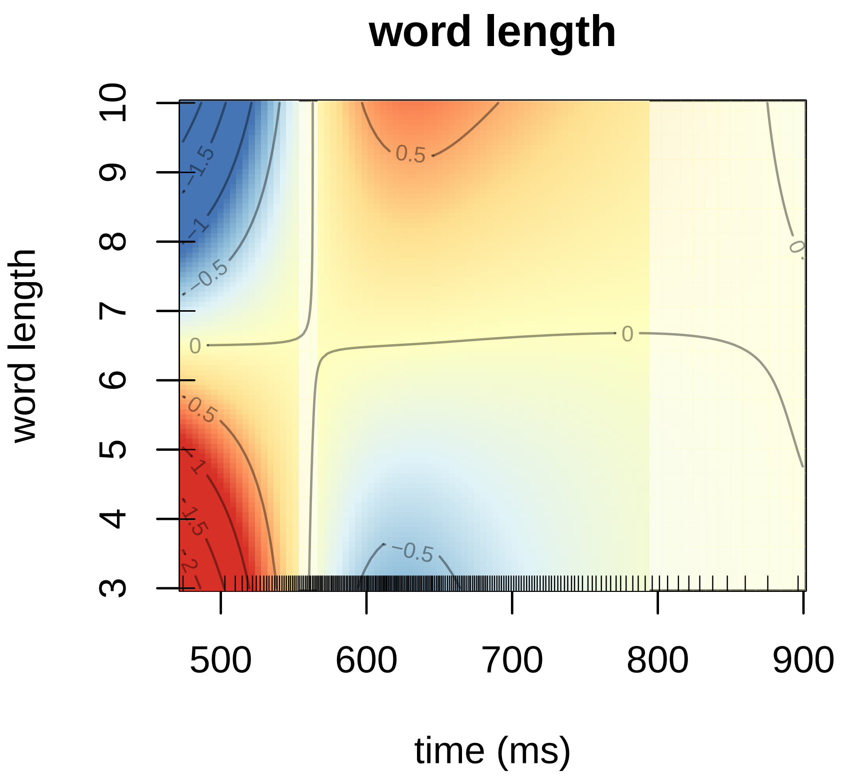
PAMM


word length
10
9
8
7
6
5
4
3
500
600
700
800
900
time (ms)
-2
-1
-0.5
0
0.5
1
1.5

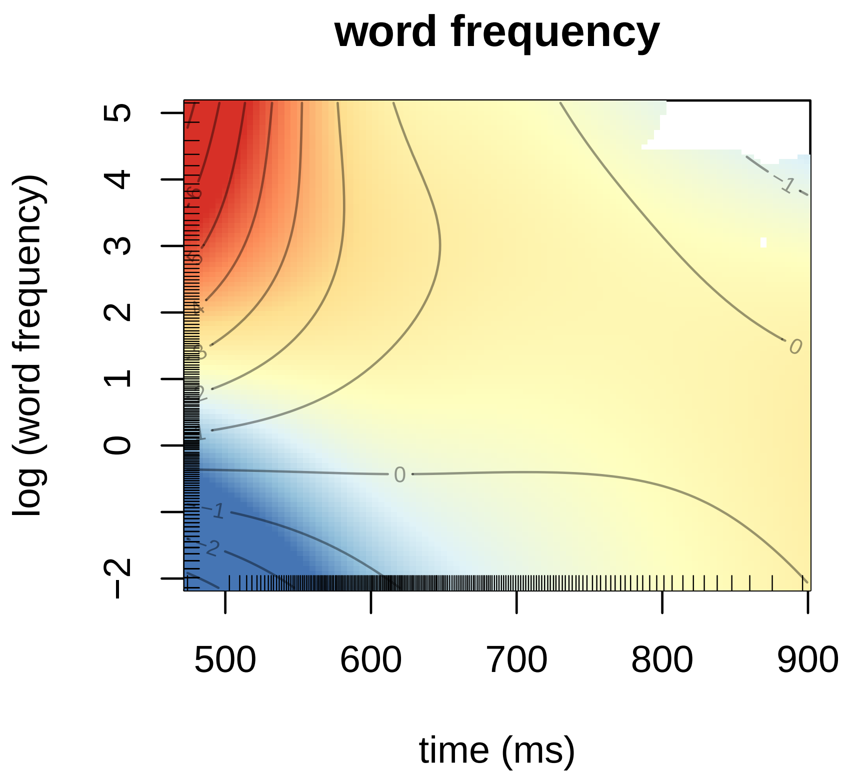
PAMM

$$=$$

PAMM


word length
word length
10
9
8
7
6
5
4
3
time (ms)
500
600
700
800
900
0.5
0
-0.5
-1
-1.5
-2

PAMM

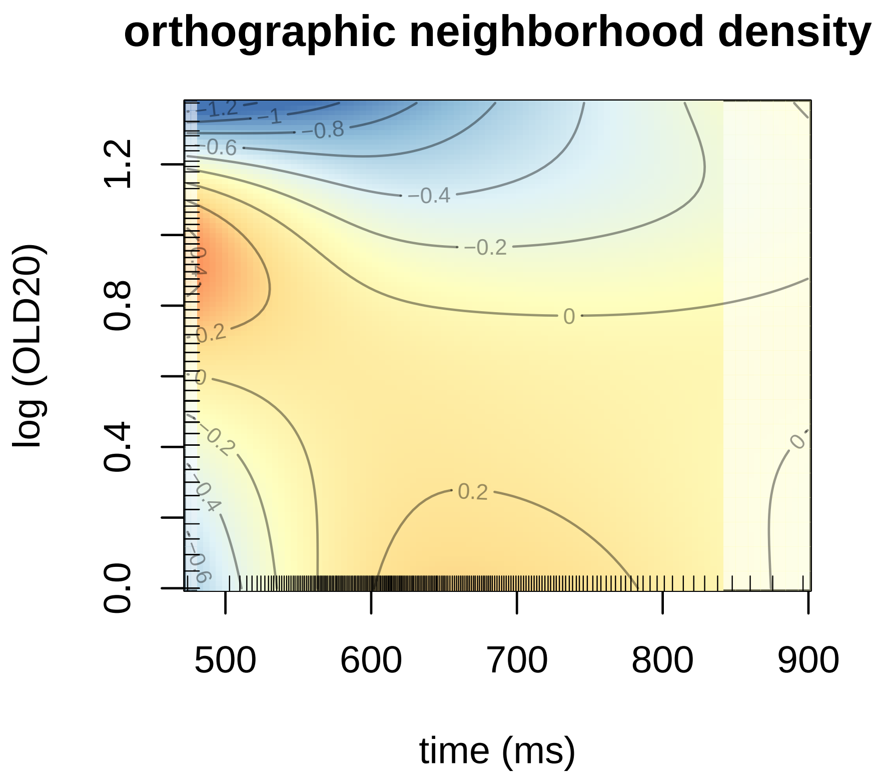

word frequency
log (word frequency)
time (ms)
5
4
3
2
1
0
−2
500
600
700
800
900
0
-1
-2
1
2
3
4
5
6

PAMM

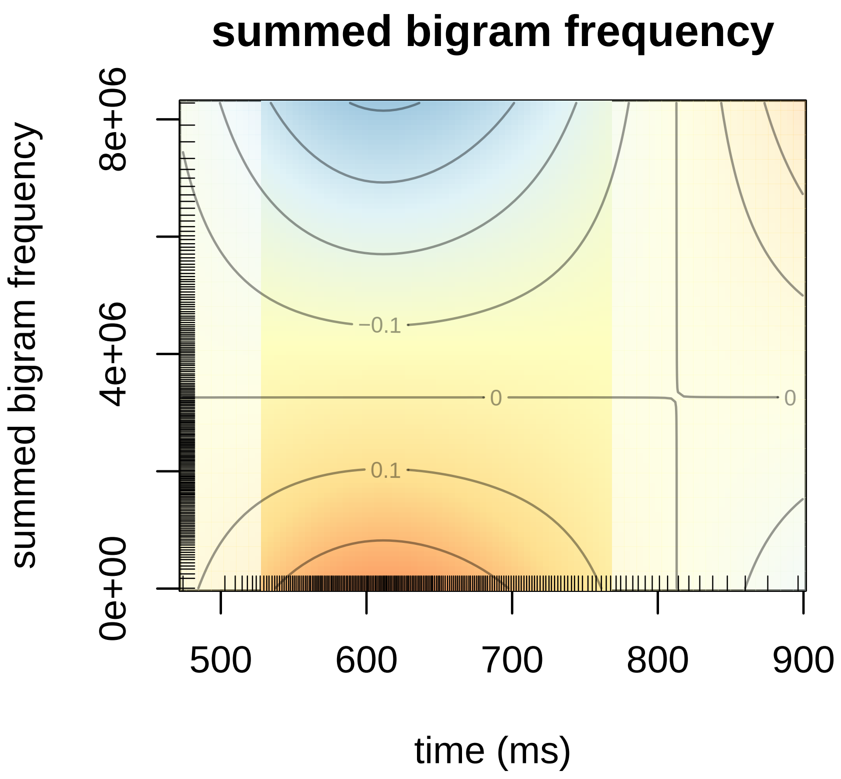
- Orthographic neighborhood density
- Levenshtein distance:
  - bear - pear $\rightarrow$ 1
  - bear - bar $\rightarrow$ 1
  - bear - cat $\rightarrow$ 3
- OLD20 is the average Levenshtein distance of the 20 closest orthographic neighbors
- OLD20 is low for words from dense orthographic neighborhoods

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

PAMM



PAMM


summed bigram frequency
summed bigram frequency
0e+00
4e+06
8e+06
500
600
700
800
900
time (ms)
-0.1
0
0.1
0

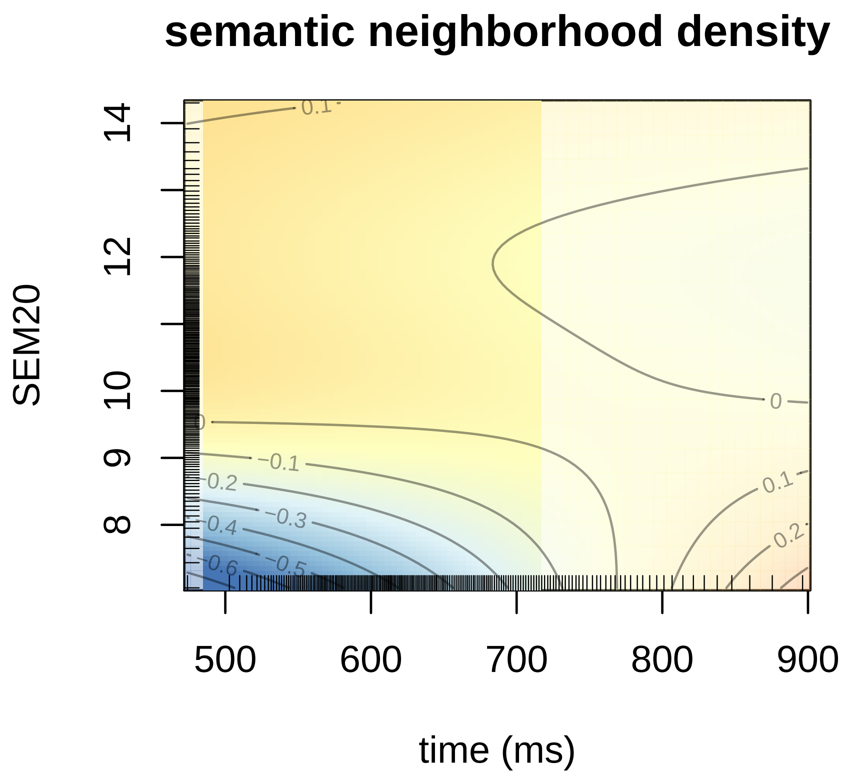
PAMM

- Semantic neighborhood density
- Based on vector semantics (fastText)
- Steps:
  - Calculate cosine similarity between a word and all other words
  - Select the 20 closest semantic neighbors
  - Sum the cosine similarities with the target word for these 20 words
- SEM20 is high for words from dense semantic neighborhoods

PAMM

```
# Semantic neighbors for monkey:
#   monkeys       ape    baboon  squirrel     chimp      apes
# 0.7531590 0.6818730 0.6241196 0.5961598 0.5720144 0.5603850
#    rhesus       cat       rat    parrot    chimps     panda
# 0.5549141 0.5491605 0.5464469 0.5462143 0.5360519 0.5347258
#   baboons     lemur    simian   raccoon  primates    donkey
# 0.5245028 0.5126457 0.5101910 0.5078749 0.5063482 0.5058308
#     sloth     mouse
# 0.5053900 0.5039323
#
# Mean:
# 0.556597
```

PAMM


semantic neighborhood density
SEM20
time (ms)
500
600
700
800
900
8
9
10
12
14
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
0.1
0.2

PAMM

- Piece-wise exponential generalized additive mixed models (PAMMs) allow for survival analysis within the framework of generalized additive mixed-effect models (GAMMs)
- Possibility to model non-linear predictors effects that vary non-linearly over time
- Insight into the temporal development of predictor effects in studies with uni-dimensional dependent variables

Survival analysis

Thank you!