

Survival Analysis

Survival analysis

- So far, have seen:
 - ▶ response variable counted or measured (regression)
 - ▶ response variable categorized (logistic regression)
- But what if response is time until event (eg. time of survival after surgery)?
- Additional complication: event might not have happened at end of study (eg. patient still alive). But knowing that patient has “not died yet” presumably informative. Such data called *censored*.

... continued

- Enter *survival analysis*, in particular the “Cox proportional hazards model”.
- Explanatory variables in this context often called *covariates*.

Packages

- Install package survival if not done. Also use broom and marginalesffects from earlier.

```
library(tidyverse)
library(survival)
library(broom)
library(marginalesffects)
```

Example: still dancing?

- 12 women who have just started taking dancing lessons are followed for up to a year, to see whether they are still taking dancing lessons, or have quit. The “event” here is “quit”.
- This might depend on:
 - ▶ a treatment (visit to a dance competition)
 - ▶ woman's age (at start of study).

Data

Months	Quit	Treatment	Age
1	1	0	16
2	1	0	24
2	1	0	18
3	0	0	27
4	1	0	25
7	1	1	26
8	1	1	36
10	1	1	38
10	0	1	45
12	1	1	47

About the data

- `months` and `quit` are kind of combined response:
 - ▶ `Months` is number of months a woman was actually observed dancing
 - ▶ `quit` is 1 if woman quit, 0 if still dancing at end of study.
- `Treatment` is 1 if woman went to dance competition, 0 otherwise.
- Fit model and see whether `Age` or `Treatment` have effect on survival.
- Want to do predictions for probabilities of still dancing as they depend on whatever is significant, and draw plot.

Read data

- Column-aligned:

```
url <- "http://ritsokiguess.site/datafiles/dancing.txt"  
dance <- read_table(url)
```


The data

```
dance
```

```
# A tibble: 12 x 4
```

	Months <dbl>	Quit <dbl>	Treatment <dbl>	Age <dbl>
1	1	1	0	16
2	2	1	0	24
3	2	1	0	18
4	3	0	0	27
5	4	1	0	25
6	5	1	0	21
7	11	1	0	55
8	7	1	1	26
9	8	1	1	36
10	10	1	1	38
11	10	0	1	45
12	12	1	1	47

Fit model

- Response variable has to incorporate both the survival time (Months) and whether or not the event, quitting, happened (that is, if Quit is 1).
- This is made using Surv from survival package, with two inputs:
 - ▶ the column that has the survival times
 - ▶ something that is TRUE or 1 if the event happened.
- Easiest for us to create this when we fit the model, predicting response from explanatories:

```
dance.1 <- coxph(Surv(Months, Quit) ~ Treatment + Age,  
                 data = dance)
```

What does Surv output actually look like?

```
library(survival)
library(dplyr)

dance %>% mutate(y = Surv(Months, Quit)) %>%
  slice(1:6) # top 6 rows to fit
```

```
# A tibble: 6 x 5
```

	Months <dbl>	Quit <dbl>	Treatment <dbl>	Age <dbl>	y <Surv>
1	1	1	0	16	1
2	2	1	0	24	2
3	2	1	0	18	2
4	3	0	0	27	3+
5	4	1	0	25	4
6	5	1	0	21	5

Output looks a lot like regression

```
summary(dance.1)
```

Call:

```
coph(formula = Surv(Months, Quit) ~ Treatment + Age, data = dance)
```

n= 12, number of events= 10

	coef	exp(coef)	se(coef)	z	Pr(> z)
Treatment	-4.44915	0.01169	2.60929	-1.705	0.0882 .
Age	-0.36619	0.69337	0.15381	-2.381	0.0173 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
Treatment	0.01169	85.554	7.026e-05	1.9444
Age	0.69337	1.442	5.129e-01	0.9373

Concordance= 0.964 (se = 0.039)

Likelihood ratio test= 21.68 on 2 df, p=2e-05

Wald test = 5.67 on 2 df, p=0.06

Score (logrank) test = 14.75 on 2 df, p=6e-04

Conclusions

- Use $\alpha = 0.10$ here since not much data.
- Three tests at bottom like global F-test. Consensus that something predicts survival time (whether or not dancer quit and/or how long it took).
- Age (definitely), Treatment (marginally) both predict survival time.

Behind the scenes

- All depends on *hazard rate*, which is based on probability that event happens in the next short time period, given that event has not happened yet:
- X denotes time to event, δ is small time interval:
- $h(t) = P(X \leq t + \delta | X \geq t) / \delta$
- if $h(t)$ large, event likely to happen soon (lifetime short)
- if $h(t)$ small, event unlikely to happen soon (lifetime long).

Modelling lifetime

- want to model hazard rate
- but hazard rate always positive, so actually model *log* of hazard rate
- modelling how (log-)hazard rate depends on other things eg $X_1 =$ age, $X_2 =$ treatment, with the β being regression coefficients:
- Cox model $h(t) = h_0(t) \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots)$, or:
- $\log(h(t)) = \log(h_0(t)) + \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots$
- like a generalized linear model with log link.

Predictions with marginaleffects

- Predicted survival probabilities depend on:
 - ▶ the combination of explanatory variables you are looking at
 - ▶ the time at which you are looking at them (when more time has passed, it is more likely that the event has happened, so the “survival probability” should be lower).
- look at effect of age by comparing ages 20 and 40, and later look at the effect of treatment (values 1 and 0).
- Also have to provide some times to predict for, in Months.

Effect of age

```
new <- datagrid(model = dance.1, Age = c(20, 40), Months = c(3, 5, 7))
new
```

	Quit	Treatment	Age	Months	rowid
1	1	0	20	3	1
2	1	0	20	5	2
3	1	0	20	7	3
4	1	0	40	3	4
5	1	0	40	5	5
6	1	0	40	7	6

These are actually for women who *did not* go to the dance competition.

The predictions

```
cbind(predictions(dance.1, newdata = new, type = "survival"))  
  select(Age, Treatment, Months, estimate)
```

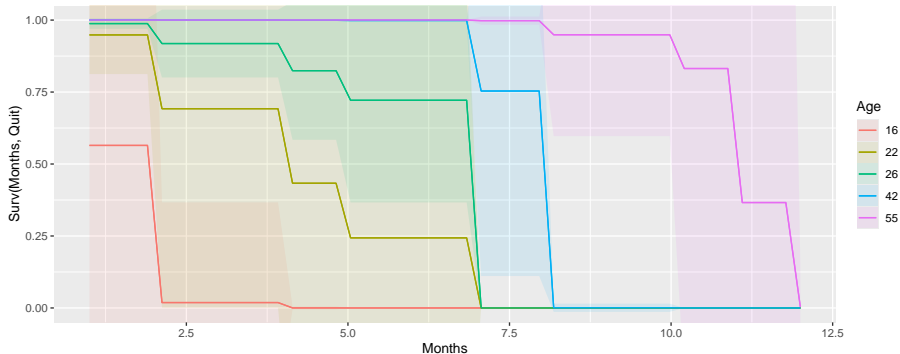
	Age	Treatment	Months	estimate
1	20	0	3	3.987336e-01
2	20	0	5	2.934959e-02
3	20	0	7	2.964394e-323
4	40	0	3	9.993936e-01
5	40	0	5	9.976749e-01
6	40	0	7	6.126327e-01

The estimated survival probabilities go down over time. For example a 20-year-old woman here has estimated probability 0.0293 of still dancing after 5 months.

A graph

We can plot the predictions over time for an experimental condition such as age. The key for `plot_predictions` is to put time *first* in the condition:

```
plot_predictions(dance.1, condition = c("Months", "Age"),  
                 type = "survival") +  
  coord_cartesian(ylim = c(0, 1)) # y-axis from 0 to 1
```



Comments

- The plot picks some representative ages.
- It is (usually) best to be up and to the right (has the highest chance of surviving longest).
- Hence the oldest women have the best chance to still be dancing longest (the youngest women are most likely to quit soonest).

The effect of treatment

The same procedure will get predictions for women who did or did not go to the dance competition, at various times:

```
new <- datagrid(model = dance.1, Treatment = c(0, 1), Months =  
new
```

	Quit	Age	Treatment	Months	rowid
1	1	31.5	0	3	1
2	1	31.5	0	5	2
3	1	31.5	0	7	3
4	1	31.5	1	3	4
5	1	31.5	1	5	5
6	1	31.5	1	7	6

The age used for predictions is the mean of all ages.

The predictions

```
cbind(predictions(dance.1, newdata = new, type = "survival"))  
  select(Age, Treatment, Months, estimate)
```

	Age	Treatment	Months	estimate
1	31.5	0	3	9.864573e-01
2	31.5	0	5	9.490195e-01
3	31.5	0	7	1.646297e-05
4	31.5	1	3	9.998406e-01
5	31.5	1	5	9.993886e-01
6	31.5	1	7	8.792014e-01

Women of this age have a high (0.879) chance of still dancing after 7 months if they went to the dance competition, but much lower (almost zero) if they did not.

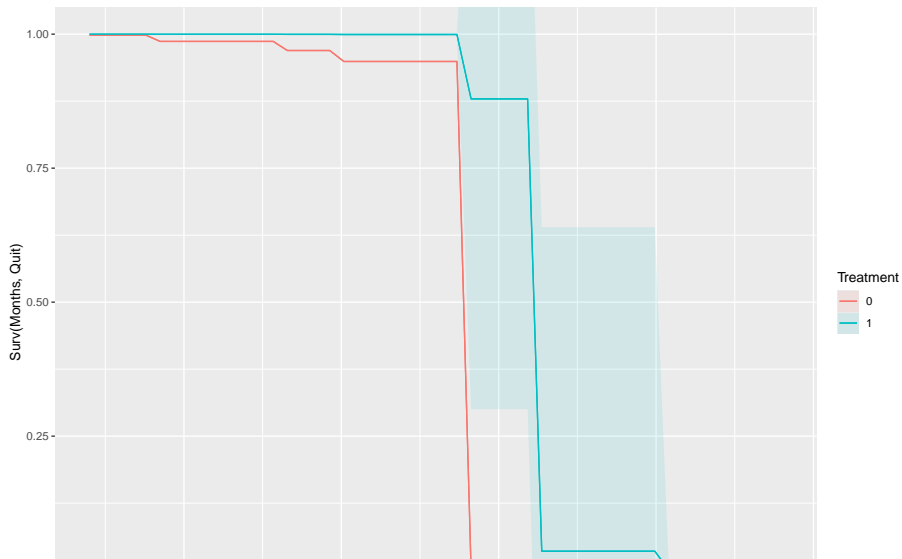
A graph

Again, time first, effect of interest second (as colours):

```
plot_predictions(dance.1,  
                 condition = c("Months", "Treatment"),  
                 type = "survival") +  
coord_cartesian(ylim = c(0, 1)) -> g
```

The graph

gg



Comments

- The survival curve for Treatment 1 is higher all the way along
- Hence at any time, the women who went to the dance competition have a higher chance of still dancing than those who did not.

The model summary again

```
summary(dance.1)
```

Call:

```
coxph(formula = Surv(Months, Quit) ~ Treatment + Age, data = c
```

```
n= 12, number of events= 10
```

	coef	exp(coef)	se(coef)	z	Pr(> z)
Treatment	-4.44915	0.01169	2.60929	-1.705	0.0882 .
Age	-0.36619	0.69337	0.15381	-2.381	0.0173 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
Treatment	0.01169	85.554	7.026e-05	1.9444
Age	0.69337	1.442	5.129e-01	0.9373

Comments

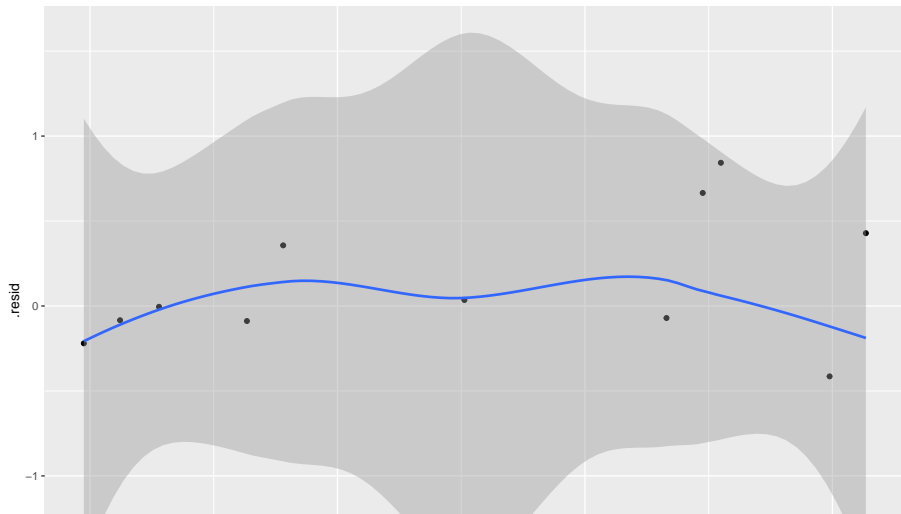
- The numbers in the `coef` column describe effect of that variable on log-hazard of quitting.
- Both numbers are negative, so a higher value on both variables goes with a lower hazard of quitting:
 - ▶ an older woman is less likely to quit soon (more likely to be still dancing)
 - ▶ a woman who went to the dance competition (`Treatment = 1`) is less likely to quit soon vs. a woman who didn't (more likely to be still dancing).

Model checking

- With regression, usually plot residuals against fitted values.
- Not quite same here (nonlinear model), but “martingale residuals” should have no pattern vs. “linear predictor”.
- Use broom ideas to get them, in `.resid` and `.fitted` as below.
- Martingale residuals can go very negative, so won't always look normal.

Martingale residuals

```
dance.1 %>% augment(dance) %>%  
  ggplot(aes(x = .fitted, y = .resid)) + geom_point() + geom_s
```



A more realistic example: lung cancer

- When you load in an R package, get data sets to illustrate functions in the package.
- One such is `lung`. Data set measuring survival in patients with advanced lung cancer.
- Along with survival time, number of “performance scores” included, measuring how well patients can perform daily activities.
- Sometimes high good, but sometimes bad!
- Variables below, from the data set help file (`?lung`).

The variables

Format

inst: Institution code
time: Survival time in days
status: censoring status 1=censored, 2=dead
age: Age in years
sex: Male=1 Female=2
ph.ecog: ECOG performance score (0=good 5=dead)
ph.karno: Karnofsky performance score (bad=0-good=100) rated by physician
pat.karno: Karnofsky performance score as rated by patient
meal.cal: Calories consumed at meals
wt.loss: Weight loss in last six months

Uh oh, missing values

```
lung %>% select(meal.cal, wt.loss)
```

	meal.cal	wt.loss
1	1175	NA
2	1225	15
3	NA	15
4	1150	11
5	NA	0
6	513	0
7	384	10
8	538	1
9	825	16
10	271	34
11	1025	27
12	NA	23
13	NA	5
14	1225	32
15	2600	60
16	NA	15
17	1150	-5
18	1025	22
19	238	10
20	1175	NA
21	1025	17
22	1175	-8

A closer look

```
summary(lung)
```

inst		time		status		age	
Min.	: 1.00	Min.	: 5.0	Min.	:1.000	Min.	:39.00
1st Qu.:	3.00	1st Qu.:	166.8	1st Qu.:	1.000	1st Qu.:	56.00
Median	:11.00	Median	: 255.5	Median	:2.000	Median	:63.00
Mean	:11.09	Mean	: 305.2	Mean	:1.724	Mean	:62.45
3rd Qu.:	16.00	3rd Qu.:	396.5	3rd Qu.:	2.000	3rd Qu.:	69.00
Max.	:33.00	Max.	:1022.0	Max.	:2.000	Max.	:82.00
NA's	:1						

sex		ph.ecog		ph.karno		pat.karno	
Min.	:1.000	Min.	:0.0000	Min.	: 50.00	Min.	: 30.00
1st Qu.:	1.000	1st Qu.:	0.0000	1st Qu.:	75.00	1st Qu.:	70.00
Median	:1.000	Median	:1.0000	Median	: 80.00	Median	: 80.00
Mean	:1.395	Mean	:0.9515	Mean	: 81.94	Mean	: 79.96
3rd Qu.:	2.000	3rd Qu.:	1.0000	3rd Qu.:	90.00	3rd Qu.:	90.00
Max.	:2.000	Max.	:3.0000	Max.	:100.00	Max.	:100.00
		NA's	:1	NA's	:1	NA's	:3

meal.cal		wt.loss	
Min.	: 96.0	Min.	: -24.000
1st Qu.:	635.0	1st Qu.:	0.000
Median	: 975.0	Median	: 7.000
Mean	: 928.8	Mean	: 9.832
3rd Qu.:	1150.0	3rd Qu.:	15.750
Max.	:2600.0	Max.	: 68.000
NA's	:47	NA's	:14

Remove obs with *any* missing values

```
lung %>% drop_na() -> lung.complete  
lung.complete %>%  
  select(meal.cal:wt.loss) %>%  
  slice(1:10)
```

	meal.cal	wt.loss
2	1225	15
4	1150	11
6	513	0
7	384	10
8	538	1
9	825	16
10	271	34
11	1025	27
15	2600	60
17	1150	-5

Check!

```
summary(lung.complete)
```

inst		time		status		age	
Min.	: 1.00	Min.	: 5.0	Min.	:1.000	Min.	:39.00
1st Qu.:	3.00	1st Qu.:	174.5	1st Qu.:	1.000	1st Qu.:	57.00
Median	:11.00	Median	: 268.0	Median	:2.000	Median	:64.00
Mean	:10.71	Mean	: 309.9	Mean	:1.719	Mean	:62.57
3rd Qu.:	15.00	3rd Qu.:	419.5	3rd Qu.:	2.000	3rd Qu.:	70.00
Max.	:32.00	Max.	:1022.0	Max.	:2.000	Max.	:82.00
sex		ph.ecog		ph.karno		pat.karno	
Min.	:1.000	Min.	:0.0000	Min.	: 50.00	Min.	: 30.00
1st Qu.:	1.000	1st Qu.:	0.0000	1st Qu.:	70.00	1st Qu.:	70.00
Median	:1.000	Median	:1.0000	Median	: 80.00	Median	: 80.00
Mean	:1.383	Mean	:0.9581	Mean	: 82.04	Mean	: 79.58
3rd Qu.:	2.000	3rd Qu.:	1.0000	3rd Qu.:	90.00	3rd Qu.:	90.00
Max.	:2.000	Max.	:3.0000	Max.	:100.00	Max.	:100.00
meal.cal		wt.loss					
Min.	: 96.0	Min.	: -24.000				
1st Qu.:	619.0	1st Qu.:	0.000				
Median	: 975.0	Median	: 7.000				
Mean	: 929.1	Mean	: 9.719				
3rd Qu.:	1162.5	3rd Qu.:	15.000				
Max.	:2600.0	Max.	: 68.000				

No missing values left.

Model 1: use everything except inst

```
names(lung.complete)
```

```
[1] "inst"      "time"      "status"    "age"       "sex"  
[6] "ph.ecog"   "ph.karno"  "pat.karno" "meal.cal"  "wt.loss"
```

- Event was death, goes with status of 2:

```
lung.1 <- coxph(  
  Surv(time, status == 2) ~ . - inst - time - status,  
  data = lung.complete  
)
```

“Dot” means “all the other variables”.

summary of model 1

```
summary(lung.1)
```

Call:

```
coph(formula = Surv(time, status == 2) ~ . - inst - time - status,  
      data = lung.complete)
```

n= 167, number of events= 120

	coef	exp(coef)	se(coef)	z	Pr(> z)
age	1.080e-02	1.011e+00	1.160e-02	0.931	0.35168
sex	-5.536e-01	5.749e-01	2.016e-01	-2.746	0.00603 **
ph.ecog	7.395e-01	2.095e+00	2.250e-01	3.287	0.00101 **
ph.karno	2.244e-02	1.023e+00	1.123e-02	1.998	0.04575 *
pat.karno	-1.207e-02	9.880e-01	8.116e-03	-1.488	0.13685
meal.cal	2.835e-05	1.000e+00	2.594e-04	0.109	0.91298
wt.loss	-1.420e-02	9.859e-01	7.766e-03	-1.828	0.06748 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
age	1.0109	0.9893	0.9881	1.0341
sex	0.5749	1.7395	0.3872	0.8534
ph.ecog	2.0950	0.4773	1.3479	3.2560
ph.karno	1.0227	0.9778	1.0004	1.0455
pat.karno	0.9880	1.0121	0.9724	1.0038
meal.cal	1.0000	1.0000	0.9995	1.0005
wt.loss	0.9859	1.0143	0.9710	1.0010

Concordance= 0.653 (se = 0.029)

Likelihood ratio test= 28.16 on 7 df, p=2e-04

Wald test = 27.5 on 7 df, p=3e-04

Score (logrank) test = 28.31 on 7 df, p=2e-04

Overall significance

The three tests of overall significance:

```
glance(lung.1) %>% select(starts_with("p.value"))
```

```
# A tibble: 1 x 4
```

	p.value.log	p.value.sc	p.value.wald	p.value.robust
	<dbl>	<dbl>	<dbl>	<dbl>
1	0.000205	0.000193	0.000271	NA

All strongly significant. *Something* predicts survival.

Coefficients for model 1

```
tidy(lung.1) %>% select(term, p.value) %>% arrange(p.value)
```

```
# A tibble: 7 x 2
  term      p.value
  <chr>      <dbl>
1 ph.ecog  0.00101
2 sex      0.00603
3 ph.karno 0.0457
4 wt.loss  0.0675
5 pat.karno 0.137
6 age      0.352
7 meal.cal 0.913
```

- sex and ph.ecog definitely significant here
- age, pat.karno and meal.cal definitely not
- Take out definitely non-sig variables, and try again.

Model 2

```
lung.2 <- update(lung.1, . ~ . - age - pat.karno - meal.cal)
summary(lung.2)
```

Call:

```
coxph(formula = Surv(time, status == 2) ~ sex + ph.ecog + ph.karno +
      wt.loss, data = lung.complete)
```

n= 167, number of events= 120

	coef	exp(coef)	se(coef)	z	Pr(> z)	
sex	-0.570881	0.565028	0.198842	-2.871	0.004091	**
ph.ecog	0.844660	2.327188	0.218644	3.863	0.000112	***
ph.karno	0.017877	1.018038	0.010887	1.642	0.100584	
wt.loss	-0.012048	0.988025	0.007495	-1.607	0.107975	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
sex	0.565	1.7698	0.3827	0.8343
ph.ecog	2.327	0.4297	1.5164	3.5700
ph.karno	1.018	0.982	0.982	1.018
wt.loss	0.988	1.012	0.988	0.988

Compare with first model:

```
anova(lung.2, lung.1)
```

Analysis of Deviance Table

Cox model: response is Surv(time, status == 2)

Model 1: ~ sex + ph.ecog + ph.karno + wt.loss

Model 2: ~ (inst + age + sex + ph.ecog + ph.karno + pat.karno

	loglik	Chisq	Df	Pr(> Chi)
--	--------	-------	----	------------

1	-495.67			
---	---------	--	--	--

2	-494.03	3.269	3	0.352
---	---------	-------	---	-------

- No harm in taking out those variables.

Model 3

Take out ph.karno and wt.loss as well.

```
lung.3 <- update(lung.2, . ~ . - ph.karno - wt.loss)
```

```
tidy(lung.3) %>% select(term, estimate, p.value)
```

```
# A tibble: 2 x 3
  term      estimate p.value
  <chr>      <dbl>    <dbl>
1 sex        -0.510  0.00958
2 ph.ecog     0.483  0.000266
```

```
summary(lung.3)
```

Call:

```
coxph(formula = Surv(time, status == 2) ~ sex + ph.ecog, data
```

```
n= 167, number of events= 120
```

Check whether that was OK

```
anova(lung.3, lung.2)
```

Analysis of Deviance Table

Cox model: response is Surv(time, status == 2)

Model 1: ~ sex + ph.ecog

Model 2: ~ sex + ph.ecog + ph.karno + wt.loss

	loglik	Chisq	Df	Pr(> Chi)
1	-498.38			
2	-495.67	5.4135	2	0.06675

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Just OK.

Commentary

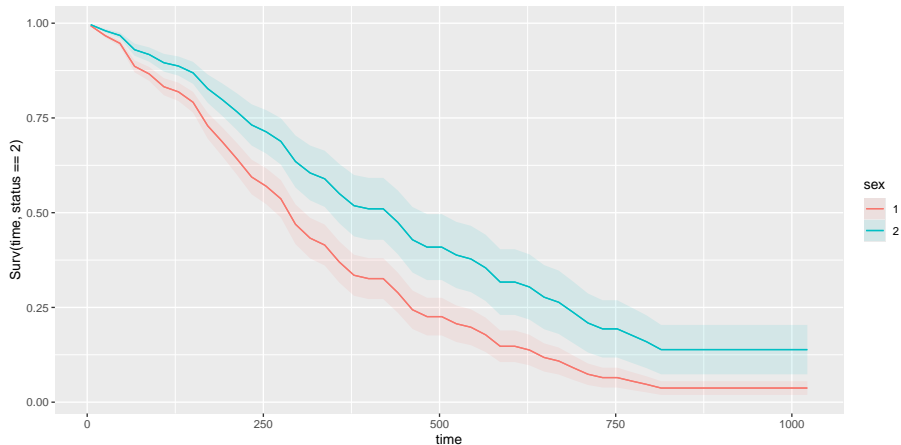
- OK (just) to take out those two covariates.
- Both remaining variables strongly significant.
- Nature of effect on survival time? Consider later.
- Picture?

Plotting survival probabilities

- Assess (separately) the effect of `sex` and `ph.ecog` score using `plot_predictions`
- Don't forget to add time (here actually called `time`) to the condition.

Effect of sex:

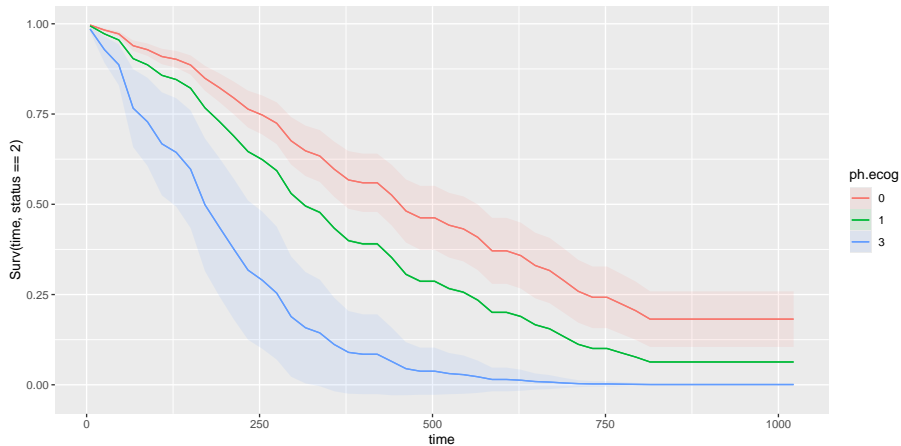
```
plot_predictions(lung.3, condition = c("time", "sex"),  
                 type = "survival")
```



- Females (sex = 2) have better survival than males.

Effect of ph.ecog score:

```
plot_predictions(lung.3, condition = c("time", "ph.ecog"),  
  type = "survival")
```



Comments

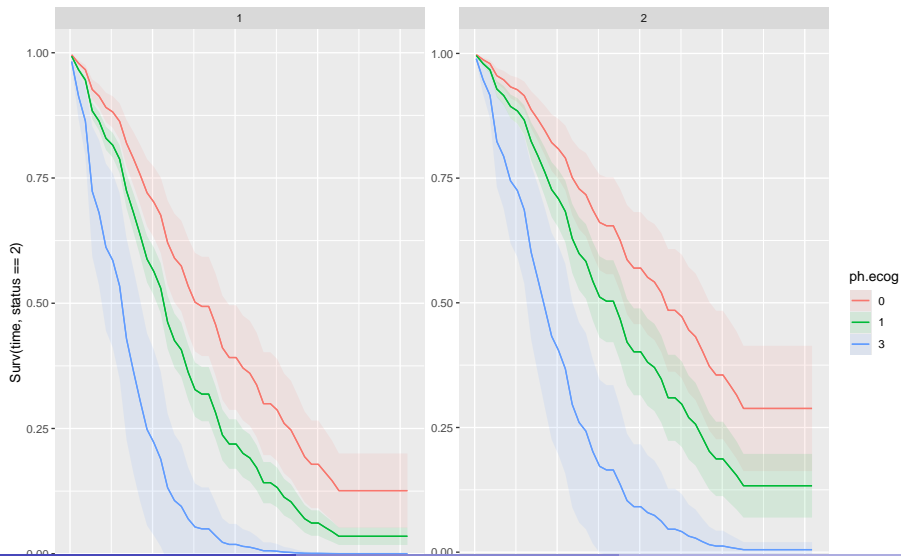
- A lower `ph.ecog` score is better.
- For example, a patient with a score of 0 has almost a 50-50 chance of living 500 days, but a patient with a score of 3 has almost no chance to survive that long.
- Is this for males or females? See over. (The comparison of scores is the same for both.) How many males and females did we observe?

```
lung %>% count(sex)
```

	sex	n
1	1	138
2	2	90

Sex and ph.ecog score

```
plot_predictions(lung.3, condition = c("time", "ph.ecog", "sex"))
```



Comments

- The previous graph was males. There were more males in the dataset (sex of 1).
- This pair of graphs shows the effect of `ph.ecog` score (above and below on each facet), and the effect of males (left) vs. females (right).
- The difference between males and females is about the same as 1 point on the `ph.ecog` scale (compare the red curve on the left facet with the green curve on the right facet).

The summary again

```
summary(lung.3)
```

Call:

```
coxph(formula = Surv(time, status == 2) ~ sex + ph.ecog, data = lung
```

```
      n= 167, number of events= 120
```

	coef	exp(coef)	se(coef)	z	Pr(> z)	
sex	-0.5101	0.6004	0.1969	-2.591	0.009579	**
ph.ecog	0.4825	1.6201	0.1323	3.647	0.000266	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
sex	0.6004	1.6655	0.4082	0.8832
ph.ecog	1.6201	0.6172	1.2501	2.0998

Concordance= 0.641 (se = 0.031)

Likelihood ratio test= 19.48 on 2 df, p=6e-05

Wald test= 19.25 on 2 df, p=6e-05

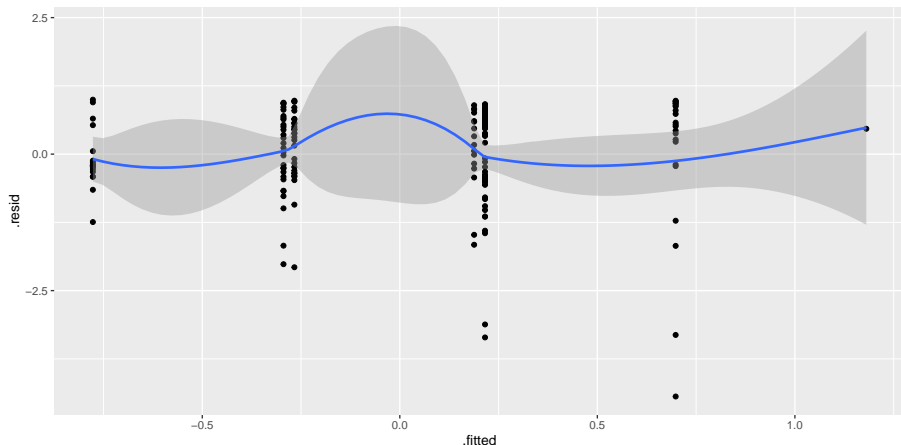
Comments

- A higher-numbered sex (female) has a lower hazard of death (negative coef). That is, females are more likely to survive longer than males.
- A higher `ph.ecog` score goes with a *higher* hazard of death (positive coef). So patients with a *lower* score are more likely to survive longer.
- These are consistent with the graphs we drew.

Martingale residuals for this model

No problems here:

```
lung.3 %>% augment(lung.complete) %>%  
  ggplot(aes(x = .fitted, y = .resid)) + geom_point() + geom_s
```



When the Cox model fails (optional)

- Invent some data where survival is best at middling age, and worse at high *and* low age:

```
age <- seq(20, 60, 5)
survtime <- c(10, 12, 11, 21, 15, 20, 8, 9, 11)
stat <- c(1, 1, 1, 1, 0, 1, 1, 1, 1)
d <- tibble(age, survtime, stat)
d %>% mutate(y = Surv(survtime, stat)) -> d
d
```

```
# A tibble: 9 x 4
```

	age	survtime	stat	y
	<dbl>	<dbl>	<dbl>	<Surv>
1	20	10	1	10
2	25	12	1	12
3	30	11	1	11
4	35	21	1	21
5	40	15	0	15+

Fit Cox model

```
y.1 <- coxph(y ~ age, data = d)
summary(y.1)
```

Call:

```
coxph(formula = y ~ age, data = d)
```

n= 9, number of events= 8

	coef	exp(coef)	se(coef)	z	Pr(> z)
age	0.01984	1.02003	0.03446	0.576	0.565

	exp(coef)	exp(-coef)	lower .95	upper .95
age	1.02	0.9804	0.9534	1.091

Concordance= 0.545 (se = 0.105)

Likelihood ratio test= 0.33 on 1 df, p=0.6

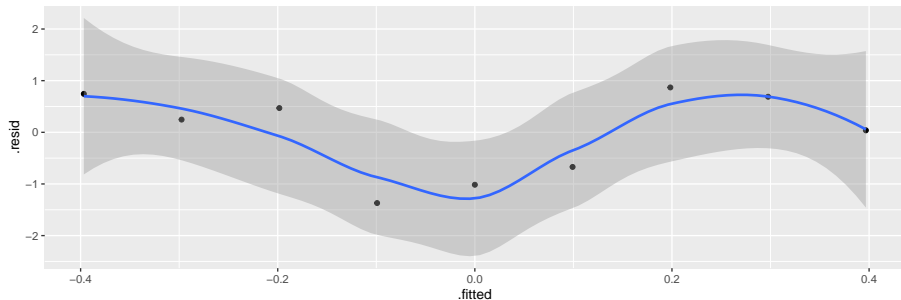
Wald test = 0.33 on 1 df, p=0.6

Score (logrank) test = 0.33 on 1 df, p=0.6

Martingale residuals

Down-and-up indicates incorrect relationship between age and survival:

```
y.1 %>% augment(d) %>%  
  ggplot(aes(x = .fitted, y = .resid)) + geom_point() + geom_s
```



Attempt 2

Add squared term in age:

```
y.2 <- coxph(y ~ age + I(age^2), data = d)
summary(y.2)
```

Call:

```
coxph(formula = y ~ age + I(age^2), data = d)
```

n= 9, number of events= 8

	coef	exp(coef)	se(coef)	z	Pr(> z)
age	-0.380184	0.683736	0.241617	-1.573	0.1156
I(age^2)	0.004832	1.004844	0.002918	1.656	0.0977 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
age	0.6837	1.4626	0.4258	1.098

Martingale residuals this time

Not great, but less problematic than before:

```
y.2 %>% augment(d) %>%  
  ggplot(aes(x = .fitted, y = .resid)) + geom_point() + geom_s
```

