Survival Analysis

Survival analysis

- So far, have seen:
  - response variable counted or measured (regression)
  - response variable categorized (logistic regression)

and have predicted response from explanatory variables.

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

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.

Packages (for this section)

- Install packages `survival` and `survminer` if not done.
- Load `survival`, `survminer`, `broom` and `tidyverse`:

```
library(tidyverse)
library(survival)
library(survminer)
library(broom)
```

Read data

- Column-aligned:

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

```
##
## -- Column specification --------------------------------
## cols(
##   Months = col_double(),
##   Quit = col_double(),
##   Treatment = col_double(),
##   Age = col_double()
## )
```

The data

```
dance
```

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

Examine response and fit model

- Response variable:

```
dance %>% mutate(mth = Surv(Months, Quit)) -> dance
```

- Then fit model, predicting `mth` from explanatories:

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

Output looks a lot like regression

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

```
## Call:
## coxph(formula = mth ~ Treatment + Age, data = dance)
##
##   n= 12, number of events= 10
##
##               coef exp(coef) se(coef)      z
## Treatment -4.44915   0.01169  2.60929 -1.705
## Age       -0.36619   0.69337  0.15381 -2.381
##           Pr(>|z|)
## Treatment   0.0882 .
## Age         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
```

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 how long it took).
- `Age` (definitely), `Treatment` (marginally) both predict survival time.

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".
- `ggcoxdiagnostics` from package `survminer` makes plot, to which we add smooth. If smooth trend more or less straight across, model OK.
- Martingale residuals can go very negative, so won't always look normal.

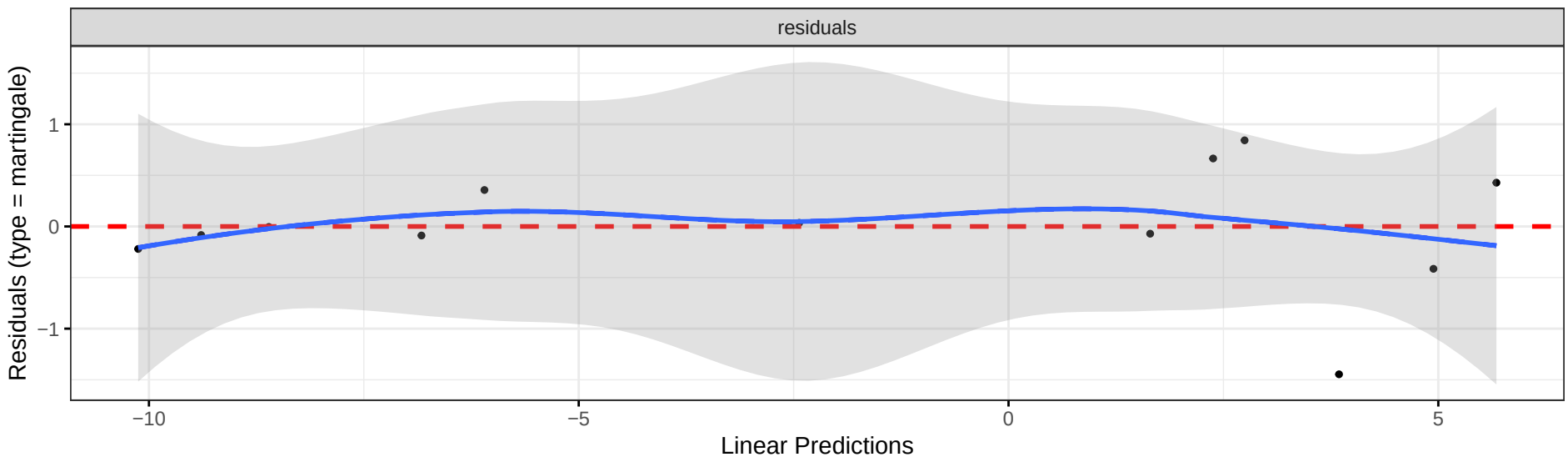
Martingale residual plot for dance data

This looks good (with only 12 points):

```
ggcoxdiagnostics(dance.1) + geom_smooth(se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Predicted survival probs

- The function we use is called `survfit`, though actually works rather like `predict`.
- First create a data frame of values to predict from. We'll do all combos of ages 20 and 40, treatment and not, using `crossing` to get all the combos:

```
treatments <- c(0, 1)
ages <- c(20, 40)
dance.new <- crossing(Treatment = treatments, Age = ages)
dance.new
```

Treatment	Age
0	20
0	40
1	20
1	40

The predictions

One prediction *for each time* for each combo of age and treatment in `dance.new`:

```
s <- survfit(dance.1, newdata = dance.new, data = dance)
summary(s)
```

```
## Call: survfit(formula = dance.1, newdata = dance.new, data = dance)
##
##  time n.risk n.event survival1 survival2 survival3 survival4
##     1     12       1  8.76e-01  1.00e+00  9.98e-01     1.000
##     2     11       2  3.99e-01  9.99e-01  9.89e-01     1.000
##     4      8       1  1.24e-01  9.99e-01  9.76e-01     1.000
##     5      7       1  2.93e-02  9.98e-01  9.60e-01     1.000
##     7      6       1 2.96e-323  6.13e-01  1.70e-04     0.994
##     8      5       1  0.00e+00  2.99e-06  1.35e-98     0.862
##    10      4       1  0.00e+00  0.00e+00  0.00e+00     0.000
##    11      2       1  0.00e+00  0.00e+00  0.00e+00     0.000
##    12      1       1  0.00e+00  0.00e+00  0.00e+00     0.000
```

Conclusions from predicted probs

- Older women more likely to be still dancing than younger women (compare "profiles" for same treatment group).
- Effect of treatment seems to be to increase prob of still dancing (compare "profiles" for same age for treatment group vs. not)
- Would be nice to see this on a graph. This is `ggsurvplot` from package `survminer`:

```
g <- ggsurvplot(s, conf.int = F)
```

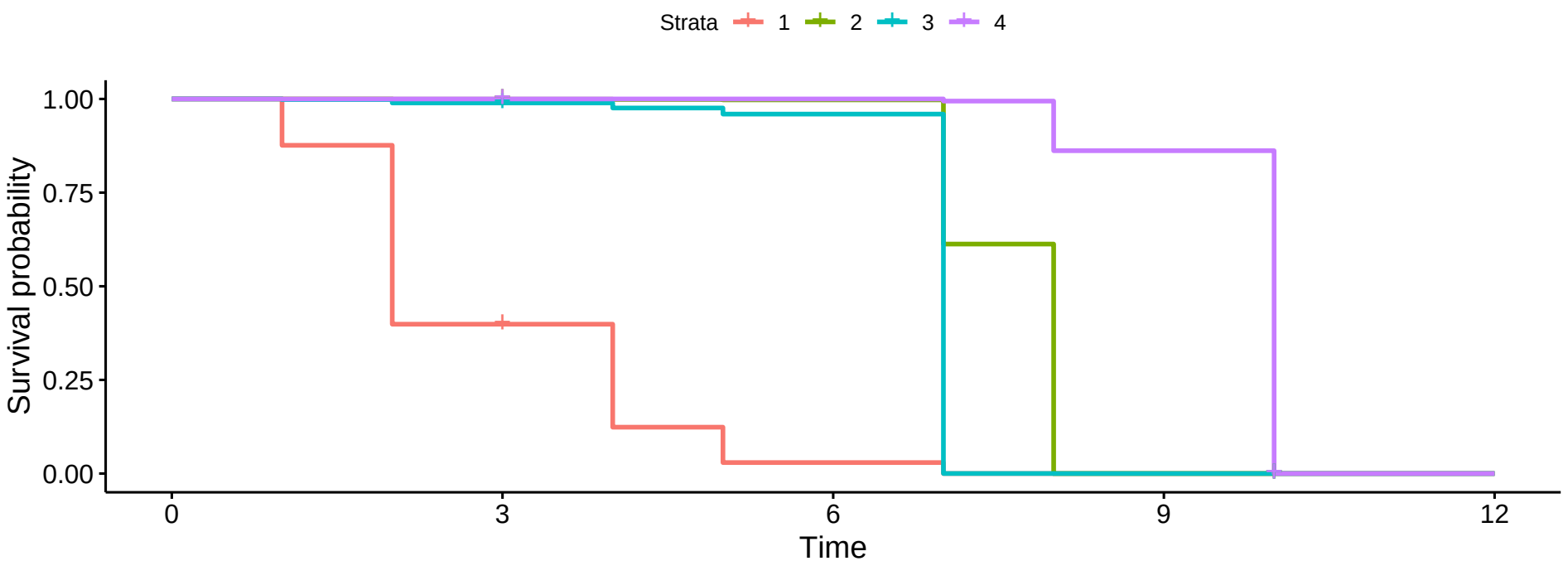
“Strata” (groups)

- uses “strata” thus (`dance.new`):

Treatment	Age
0	20
0	40
1	20
1	40

Plotting survival probabilities

```
gg
```



Discussion

- Survivor curve farther to the right is better (better chance of surviving longer).
- Best is age 40 with treatment, worst age 20 without.
- Appears to be:
  - age effect (40 better than 20)
  - treatment effect (treatment better than not)
  - In analysis, treatment effect only marginally significant.

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 %>% slice(1:16)
```

inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
3	306	2	74	1	1	90	100	1175	NA
3	455	2	68	1	0	90	90	1225	15
3	1010	1	56	1	0	90	90	NA	15
5	210	2	57	1	1	90	60	1150	11
1	883	2	60	1	0	100	90	NA	0
12	1022	1	74	1	1	50	80	513	0
7	310	2	68	2	2	70	60	384	10
11	361	2	71	2	2	60	80	538	1
1	218	2	53	1	1	70	80	825	16
7	166	2	61	1	2	70	70	271	34
6	170	2	57	1	1	80	80	1025	27
16	654	2	68	2	2	70	70	NA	23
11	728	2	68	2	1	90	90	NA	5
21	71	2	60	1	NA	60	70	1225	32
12	567	2	57	1	1	80	70	2600	60
1	144	2	67	1	1	80	90	NA	15

A closer look

```
summary(lung)
```

```
##       inst             time            status            age              sex
##  Min.   : 1.00   Min.   :   5.0   Min.   :1.000   Min.   :39.00   Min.   :1.000
##  1st Qu.: 3.00   1st Qu.: 166.8   1st Qu.:1.000   1st Qu.:56.00   1st Qu.:1.000
##  Median :11.00   Median : 255.5   Median :2.000   Median :63.00   Median :1.000
##  Mean   :11.09   Mean   : 305.2   Mean   :1.724   Mean   :62.45   Mean   :1.395
##  3rd Qu.:16.00   3rd Qu.: 396.5   3rd Qu.:2.000   3rd Qu.:69.00   3rd Qu.:2.000
##  Max.   :33.00   Max.   :1022.0   Max.   :2.000   Max.   :82.00   Max.   :2.000
##  NA's   :1
##     ph.ecog         ph.karno        pat.karno        meal.cal         wt.loss
##  Min.   :0.0000   Min.   : 50.00   Min.   : 30.00   Min.   :  96.0   Min.   :-24.000
##  1st Qu.:0.0000   1st Qu.: 75.00   1st Qu.: 70.00   1st Qu.: 635.0   1st Qu.:  0.000
##  Median :1.0000   Median : 80.00   Median : 80.00   Median : 975.0   Median :  7.000
##  Mean   :0.9515   Mean   : 81.94   Mean   : 79.96   Mean   : 928.8   Mean   :  9.832
##  3rd Qu.:1.0000   3rd Qu.: 90.00   3rd Qu.: 90.00   3rd Qu.:1150.0   3rd Qu.: 15.750
##  Max.   :3.0000   Max.   :100.00   Max.   :100.00   Max.   :2600.0   Max.   : 68.000
##  NA's   :1        NA's   :1        NA's   :3        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              sex       
##  Min.   : 1.00   Min.   :   5.0   Min.   :1.000   Min.   :39.00   Min.   :1.000  
##  1st Qu.: 3.00   1st Qu.: 174.5   1st Qu.:1.000   1st Qu.:57.00   1st Qu.:1.000  
##  Median :11.00   Median : 268.0   Median :2.000   Median :64.00   Median :1.000  
##  Mean   :10.71   Mean   : 309.9   Mean   :1.719   Mean   :62.57   Mean   :1.383  
##  3rd Qu.:15.00   3rd Qu.: 419.5   3rd Qu.:2.000   3rd Qu.:70.00   3rd Qu.:2.000  
##  Max.   :32.00   Max.   :1022.0   Max.   :2.000   Max.   :82.00   Max.   :2.000  
##     ph.ecog         ph.karno        pat.karno        meal.cal         wt.loss       
##  Min.   :0.0000   Min.   : 50.00   Min.   : 30.00   Min.   :  96.0   Min.   :-24.000  
##  1st Qu.:0.0000   1st Qu.: 70.00   1st Qu.: 70.00   1st Qu.: 619.0   1st Qu.:  0.000  
##  Median :1.0000   Median : 80.00   Median : 80.00   Median : 975.0   Median :  7.000  
##  Mean   :0.9581   Mean   : 82.04   Mean   : 79.58   Mean   : 929.1   Mean   :  9.719  
##  3rd Qu.:1.0000   3rd Qu.: 90.00   3rd Qu.: 90.00   3rd Qu.:1162.5   3rd Qu.: 15.000  
##  Max.   :3.0000   Max.   :100.00   Max.   :100.00   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"        "ph.eco
##  [8] "pat.karno" "meal.cal"   "wt.loss"
```

- Event was death, goes with `status` of 2:

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

"Dot" means "all the other variables".

summary of model 1: too tiny to see!

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

```
## Call:
## coxph(formula = resp ~ . - 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"))
```

p.value.log	p.value.sc	p.value.wald	p.value.robust
0.0002053	0.0001929	0.0002711	NA

All strongly significant. *Something* predicts survival.

Coefficients for model 1

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

term	p.value
ph.ecog	0.0010126
sex	0.0060268
ph.karno	0.0457479
wt.loss	0.0674829
pat.karno	0.1368514
age	0.3516810
meal.cal	0.9129766

- `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)
tidy(lung.2) %>% select(term, p.value)
```

term	p.value
sex	0.0040915
ph.ecog	0.0001119
ph.karno	0.1005838
wt.loss	0.1079748

Compare with first model:

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

loglik	Chisq	Df	P(>|Chi|)
-495.6689	NA	NA	NA
-494.0344	3.268999	3	0.3519808

- 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)
```

term	estimate	p.value
sex	-0.5100991	0.0095794
ph.ecog	0.4825185	0.0002656

Check whether that was OK

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

loglik	Chisq	Df	P(>|Chi|)
-498.3757	NA	NA	NA
-495.6689	5.413508	2	0.0667531

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

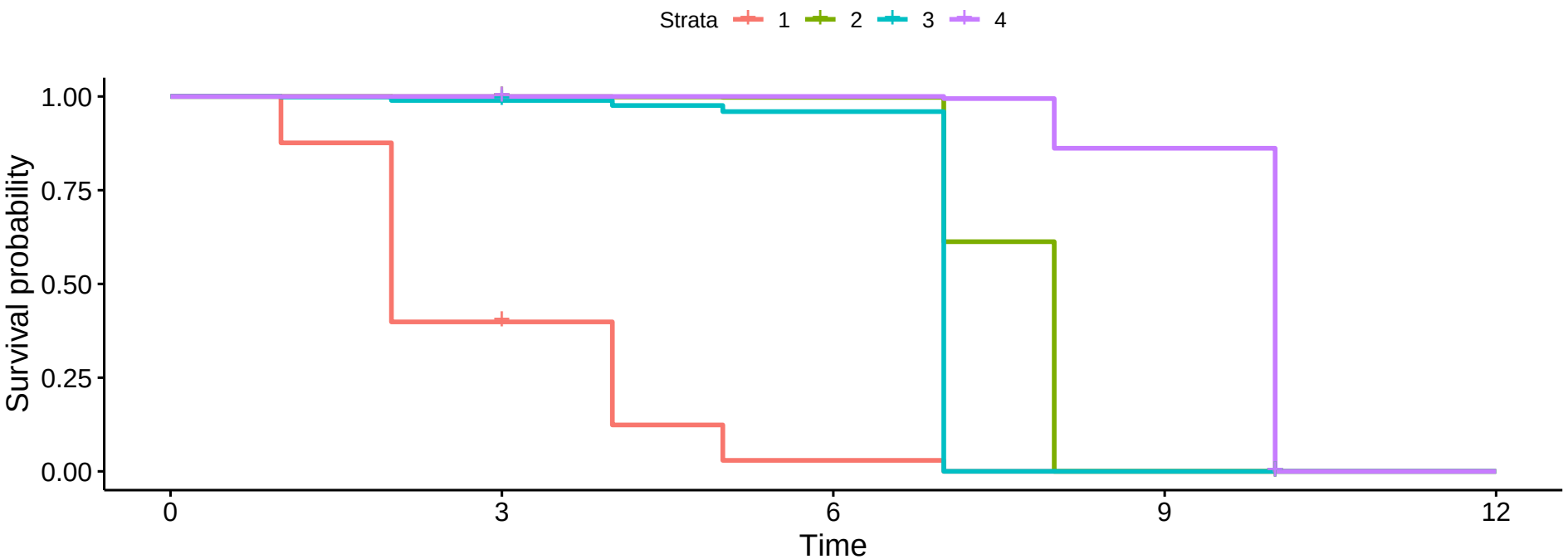
- Create new data frame of values to predict for, then predict:

```
sexes <- c(1, 2)
ph.ecogs <- 0:3
lung.new <- crossing(sex = sexes, ph.ecog = ph.ecogs)
lung.new
```

sex	ph.ecog
1	0
1	1
1	2
1	3
2	0
2	1
2	2
2	3

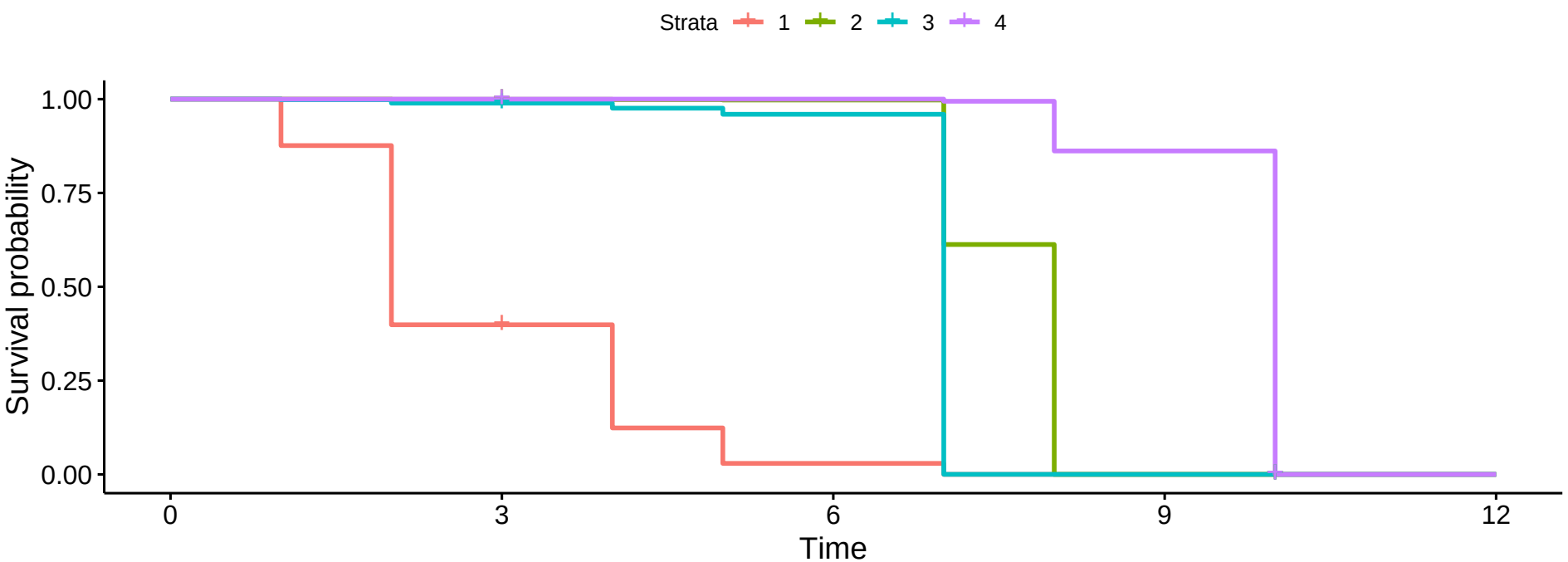
Making the plot

```
ggsurvplot(s, conf.int = F)
```



The plot

```
gg
```


Strata 1 2 3 4
1.00
0.75
0.50
0.25
0.00
Survival probability
0
3
6
9
12
Time

Discussion of survival curves

- Best survival is teal-blue curve, stratum 5, females with `ph.ecog` score 0.
- Next best: blue, stratum 6, females with score 1, and red, stratum 1, males score 0.
- Worst: green, stratum 4, males score 3.
- For any given `ph.ecog` score, females have better predicted survival than males.
- For both genders, a lower score associated with better survival.

The coefficients in model 3

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

term	estimate	p.value
sex	-0.5100991	0.0095794
ph.ecog	0.4825185	0.0002656

- `sex` coeff negative, so being higher `sex` value (female) goes with *less* hazard of dying.
- `ph.ecog` coeff positive, so higher `ph.ecog` score goes with *more* hazard of dying
- Two coeffs about same size, so being male rather than female corresponds to 1-point increase in `ph.ecog` score. Note how survival curves come in 3 pairs plus 2 odd.

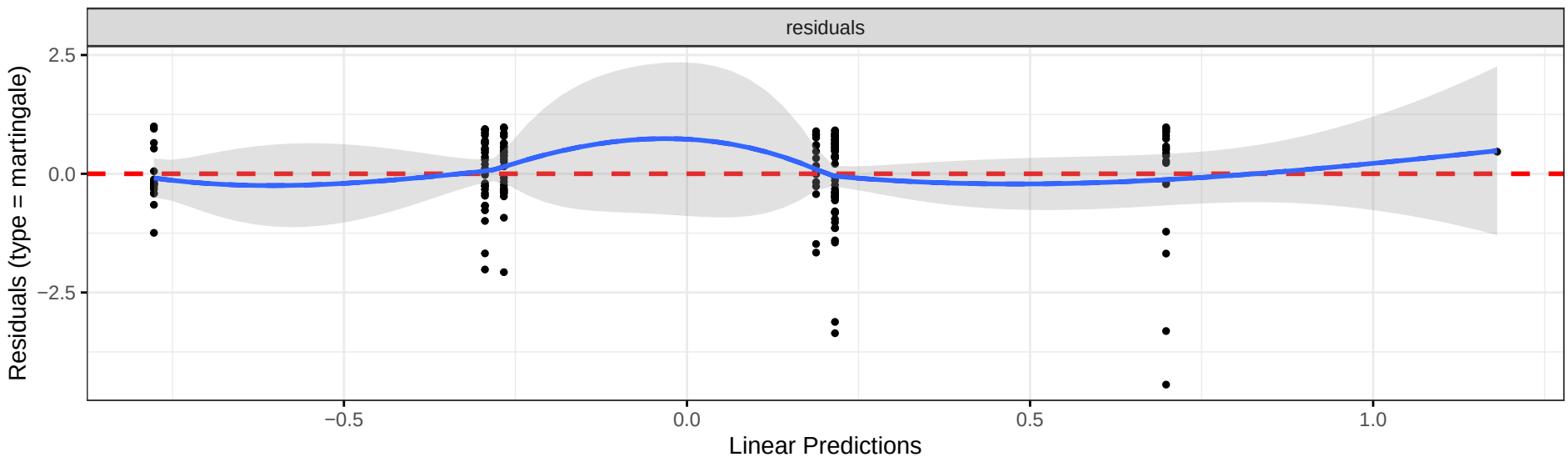
Martingale residuals for this model

No problems here:

```
ggcoxdiagnostics(lung.3) + geom_smooth(se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



When the Cox model fails

- 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
```

- Small survival time 15 in middle was actually censored, so would have been longer if observed.

Fit Cox model

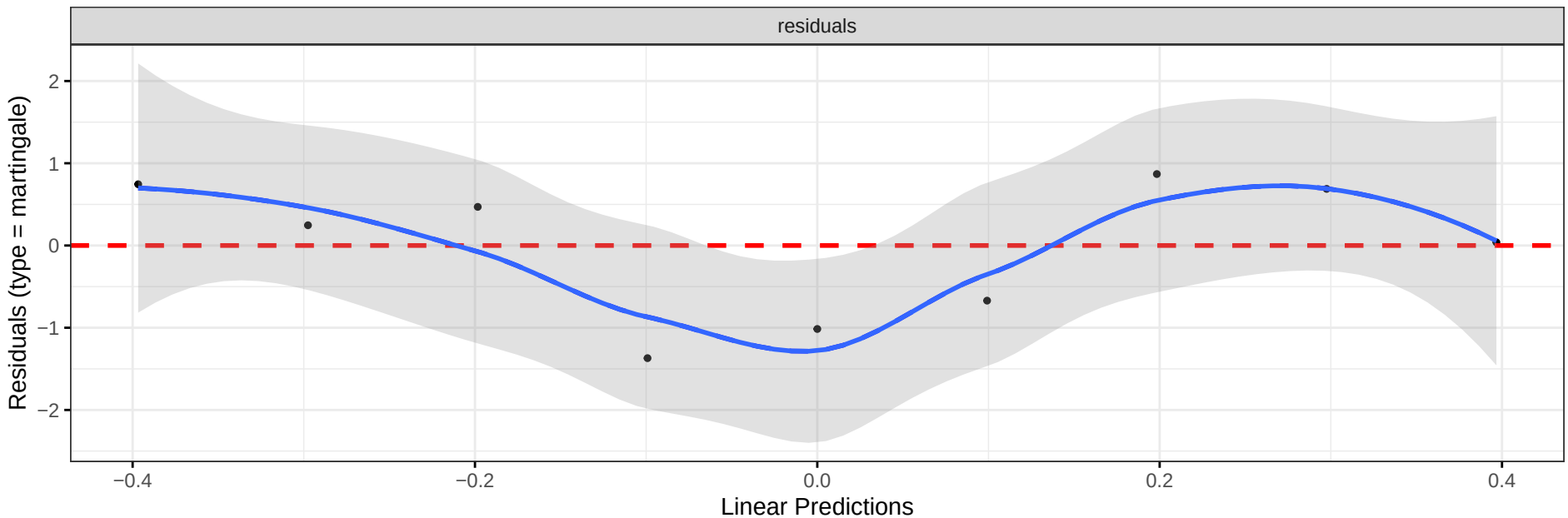
```
y.1 <- coxph(y ~ age, data = d)
summary(y.1)
```

```
## 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:

```
ggcoxdiagnostics(y.1) + geom_smooth(se = F)
```



Attempt 2

Add squared term in age:

```
y.2 <- coxph(y ~ age + I(age^2), data = d)
tidy(y.2) %>% select(term, estimate, p.value)
```

term	estimate	p.value
age	-0.3801838	0.1156031
I(age^2)	0.0048324	0.0976903

- (Marginally) helpful.

Martingale residuals this time

Not great, but less problematic than before:

```
ggcoxdiagnostics(y.2) + geom_smooth(se = F)
```