432 Class 19 Slides

thomaselove.github.io/432

2022-03-24

Today's Agenda

1. Cox models for time-to-event data
    - Returning to the breast cancer trial
    - Using `cph` from `rms` to fit a Cox model
2. Fitting Robust Linear Models
    - Using Huber weights
    - Using bisquare weights
    - Using Quantile Regression

Cox Models

Preliminaries for Cox Regression Models

```
library(here); library(janitor); library(magrittr)
library(broom); library(knitr); library(rms)

library(survival); library(survminer)

library(tidyverse)

theme_set(theme_bw())
```

```
brca <- read_csv(here("data", "brca.csv")) %>%
    type.convert(as.is = FALSE)
```

Recap of What We Did Tuesday

We're working with data from a trial of three treatments for breast cancer

- Main tibble is `brca` containing `treat` = S_CT, S_IT, S_Both and age at baseline
- Time to event data are gathered in `trial_weeks` and `last_alive` which we used to create a survival object we named `S`.
- Created Kaplan-Meier estimate, `kmfit` to compare the `treat` results
- Then built a Cox model for treatment, called `mod_T` using `coxph`.

Now, we'll

- incorporate the covariate (`age`) into the model
- use `cph` from the `rms` package to fit a Cox model that incorporates some non-linearity

Create survival object

- `trial_weeks`: time in the study, in weeks, to death or censoring
- `last_alive`: 1 if alive at last follow-up (and thus censored), 0 if dead

So `last_alive` = 0 if the event (death) occurs.

```
brca$S <- with(brca, Surv(trial_weeks, last_alive == 0))

head(brca$S)
```

```
[1] 102  192   73   58+  48+ 182+
```

Fit Cox Model mod_T: Treatment alone

```r
mod_T <- coxph(S ~ treat, data = brca)
mod_T
```

```
Call:
coxph(formula = S ~ treat, data = brca)

            coef exp(coef) se(coef)     z     p
treatS_CT 0.8313    2.2963   0.6547 1.270 0.204
treatS_IT 0.2481    1.2816   0.6740 0.368 0.713

Likelihood ratio test=1.75  on 2 df, p=0.4164
n= 31, number of events= 15
```

Fit Cox Model `mod_AT`: Age + Treatment

```r
mod_AT <- coxph(S ~ age + treat, data = brca)
mod_AT
```

```
Call:
coxph(formula = S ~ age + treat, data = brca)

              coef exp(coef) se(coef)     z      p
age        0.07807   1.08119  0.03672 2.126 0.0335
treatS_CT 0.59960   1.82139  0.65741 0.912 0.3617
treatS_IT 0.28799   1.33375  0.68566 0.420 0.6745

Likelihood ratio test=6.99  on 3 df, p=0.07224
n= 31, number of events= 15
```

Interpreting the Coefficients of `mod_AT`

```
tidy(mod_AT, exponentiate = TRUE, conf.int = TRUE) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
age	1.08	0.04	1.01	1.16
treatS_CT	1.82	0.66	0.50	6.61
treatS_IT	1.33	0.69	0.35	5.11

- If Harry and Sally receive the same `treat` but Harry is one year older, the model estimates Harry will have 1.08 times the hazard of Sally (95% CI 1.01, 1.16).

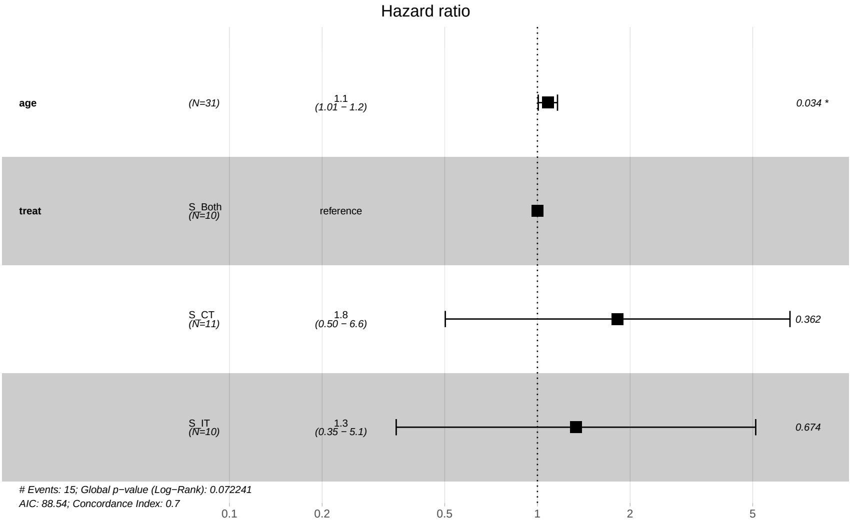
Interpreting the Coefficients of `mod_AT`

```
tidy(mod_AT, exponentiate = TRUE, conf.int = TRUE) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
age	1.08	0.04	1.01	1.16
treatS_CT	1.82	0.66	0.50	6.61
treatS_IT	1.33	0.69	0.35	5.11

- If Harry receives S_CT and Sally receives S_Both, and they are the same age, the model estimates Harry will have 1.82 times the hazard of Sally (95% CI 0.50, 6.61).
- If Cyrus receives S_IT and Sally receives S_Both, and they are the same age, the model estimates Cyrus will have 1.33 times the hazard of Sally (95% CI 0.33, 5.11).

ggforest(mod_AT, data = brca)



Comparing the Two Models

$n = 31$, $nevent = 15$ for each model.

```
bind_rows(glance(mod_T), glance(mod_AT)) %>%
    mutate(model = c("mod_T", "mod_AT")) %>%
    select(model, p.value.log, concordance, r.squared,
           max_r2 = r.squared.max, AIC, BIC) %>%
  kable(digits = c(0,3,3,3,3,1,1))
```

model	p.value.log	concordance	r.squared	max_r2	AIC	BIC
mod_T	0.416	0.577	0.055	0.944	91.8	93.2
mod_AT	0.072	0.701	0.202	0.944	88.5	90.7

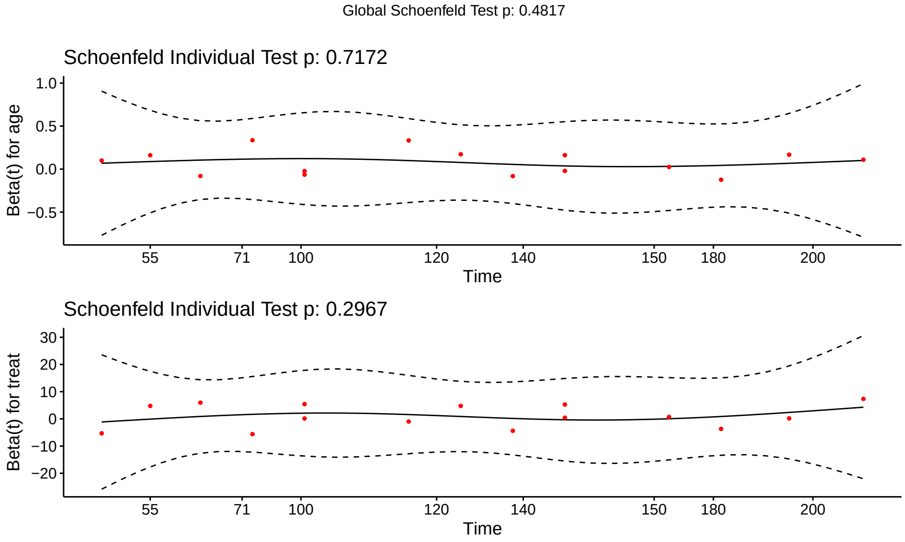
What do the `glance` results indicate?

Significance Test via Likelihood Ratio ANOVA

```
anova(mod_AT, mod_T)
```

```
Analysis of Deviance Table
 Cox model: response is  S
 Model 1: ~ age + treat
 Model 2: ~ treat
   loglik Chisq Df P(>|Chi|)
1 -41.268
2 -43.886 5.237  1   0.02211 *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Graphical PH Check `ggcoxzph(cox.zph(mod_AT))`



Using `cph` from the `rms` package

Using `rms::cph` to fit a fancier `AxT`

```
brca <- read_csv(here("data", "brca.csv")) %>%
    type.convert(as.is = FALSE) # reload without S

d <- datadist(brca)
options(datadist="d")

brca$S <- with(brca, Surv(trial_weeks, last_alive == 0))

cph_AxT <- cph(S ~ rcs(age, 4) + treat + age %ia% treat,
               data = brca,
               x = TRUE, y = TRUE, surv = TRUE)
```

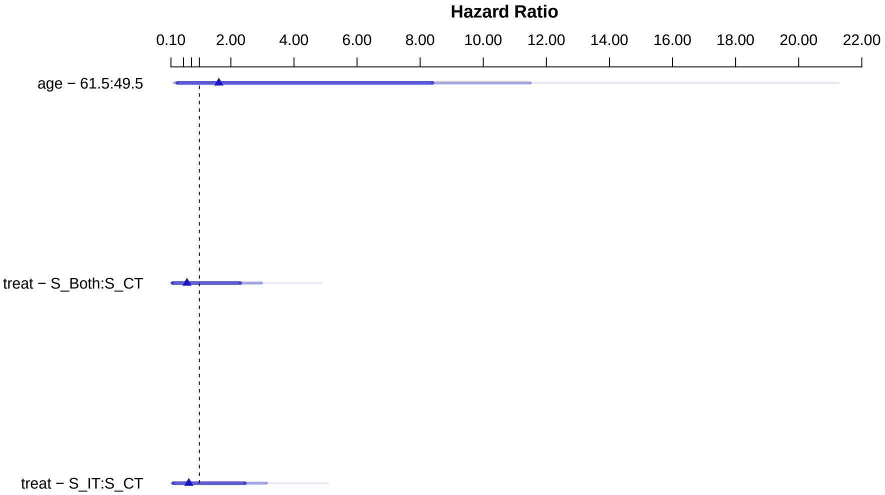
cph_AxT results

```
> cph_AxT
Cox Proportional Hazards Model

 cph(formula = S ~ rcs(age, 4) + treat + age %ia% treat, data = brca,
     x = TRUE, y = TRUE, surv = TRUE)

                      Model Tests          Discrimination
                                              Indexes
 Obs           31    LR chi2      11.66    R2        0.332
 Events        15    d.f.             7    Dxy       0.488
 Center 14.2906      Pr(> chi2) 0.1123     g         1.980
                     Score chi2  11.89     gr        7.245
                     Pr(> chi2) 0.1042

                     Coef    S.E.   Wald Z Pr(>|Z|)
 age                  0.3011 0.2330  1.29  0.1963
 age'                -1.2521 0.7528 -1.66  0.0963
 age''                2.7316 1.5490  1.76  0.0778
 treat=S_CT          -4.9327 6.6650 -0.74  0.4592
 treat=S_IT           0.1210 4.8180  0.03  0.9800
 age * treat=S_CT     0.1006 0.1157  0.87  0.3846
 age * treat=S_IT    -0.0005 0.0835 -0.01  0.9949
```

summary(cph_AxT)

```
             Effects              Response : S

 Factor                  Low  High Diff. Effect   S.E.
 age                     49.5 61.5 12     0.48200 0.99998
  Hazard Ratio           49.5 61.5 12     1.61930      NA
 treat - S_Both:S_CT      2.0  1.0 NA    -0.49745 0.80805
  Hazard Ratio            2.0  1.0 NA     0.60808      NA
 treat - S_IT:S_CT        2.0  3.0 NA    -0.40504 0.78888
  Hazard Ratio            2.0  3.0 NA     0.66695      NA
 Lower 0.95 Upper 0.95
 -1.47790     2.4419
  0.22811    11.4950
 -2.08120     1.0863
  0.12478     2.9633
 -1.95120     1.1411
  0.14210     3.1303
```

plot(summary(cph_AxT))



```r
set.seed(432)
validate(cph_AxT)
```

```
Divergence or singularity in 1 samples

      index.orig training    test optimism index.corrected
Dxy       0.4883   0.5965  0.3693   0.2273          0.2610
R2        0.3320   0.4741  0.2061   0.2680          0.0640
Slope     1.0000   1.0000  0.3819   0.6181          0.3819
D         0.1191   0.2078  0.0650   0.1428         -0.0237
U        -0.0223  -0.0226  1.0971  -1.1196          1.0973
Q         0.1414   0.2303 -1.0321   1.2624         -1.1210
g         1.9803   4.2315  1.2169   3.0145         -1.0342
       n
Dxy   39
R2    39
Slope 39
D     39
U     39
Q     39
```

ANOVA for `cph_AxT` model

```
> anova(cph_AxT)
                Wald Statistics          Response: S

 Factor                                     Chi-Square d.f. P
 age   (Factor+Higher Order Factors)        7.71       5    0.1727
  All Interactions                          0.96       2    0.6175
  Nonlinear                                 3.73       2    0.1548
 treat  (Factor+Higher Order Factors)       2.58       4    0.6297
  All Interactions                          0.96       2    0.6175
 age * treat  (Factor+Higher Order Factors) 0.96       2    0.6175
 TOTAL NONLINEAR + INTERACTION              3.74       4    0.4423
 TOTAL                                      8.55       7    0.2868
```

survplot in rms (code)

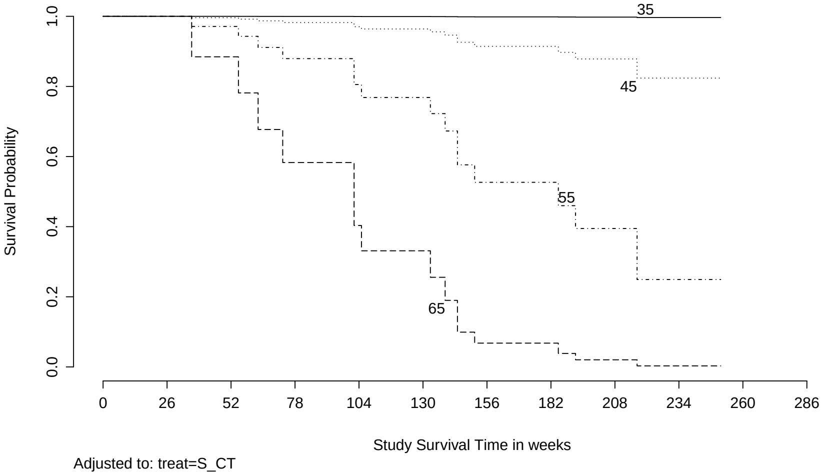
For age comparison:

```
survplot(cph_AxT,
         age = c(35, 45, 55, 65),
         time.inc = 26,
         type = "kaplan-meier",
         xlab = "Study Survival Time in weeks")
```

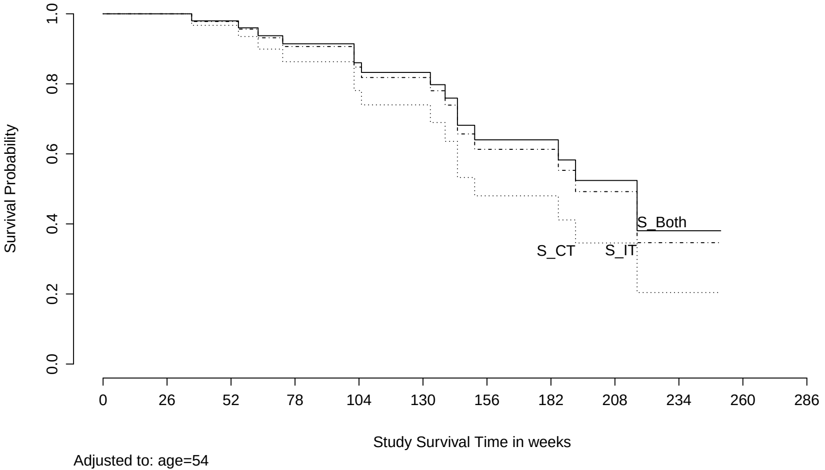
For treat comparison:

```
survplot(cph_AxT,
         treat,
         time.inc = 26,
         type = "kaplan-meier",
         xlab = "Study Survival Time in weeks")
```

survplot in rms (Result)

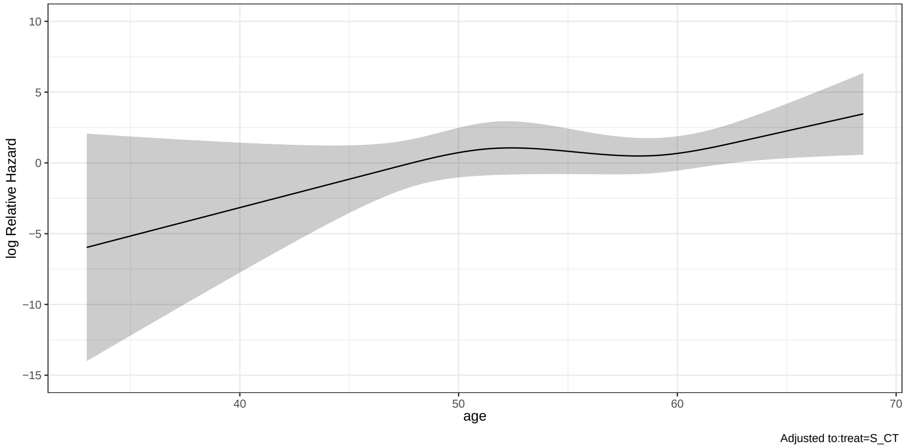


survplot for treat in rms (Result)



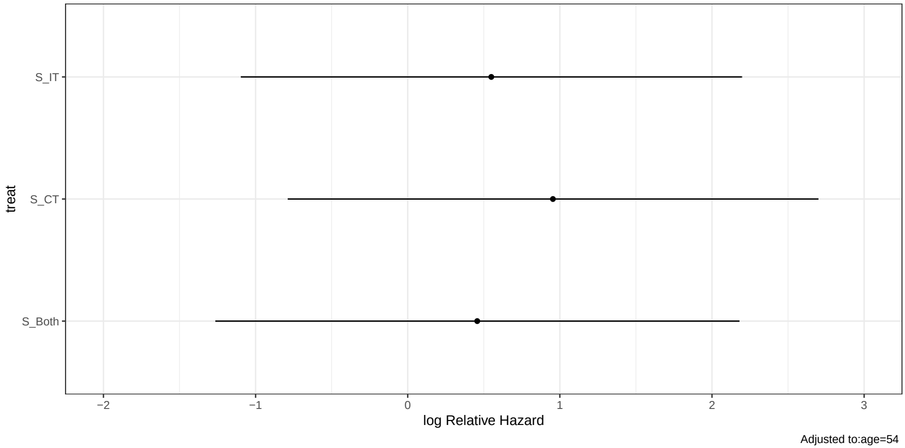
Plotting age effect implied by cph_AxT model

```
ggplot(Predict(cph_AxT, age))
```



Plotting `treat` effect implied by `cph_AxT` model

```
ggplot(Predict(cph_AxT, treat))
```

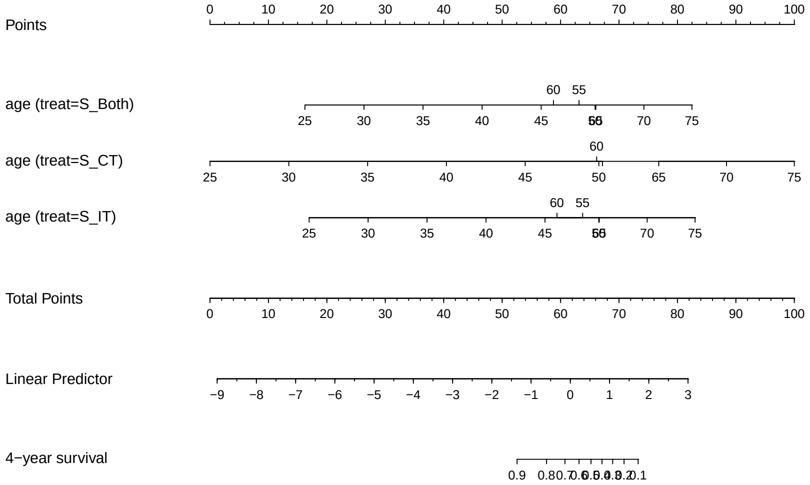


cph_AxT nomogram (code)

Suppose I want to show 4-year survival rates at the bottom of the nomogram...

```
sv <- Survival(cph_AxT)
surv4 <- function(x) sv(208, lp = x)

plot(nomogram(cph_AxT,
              fun = surv4,
              funlabel = c("4 year survival")))
```

cph_AxT nomogram (Results)

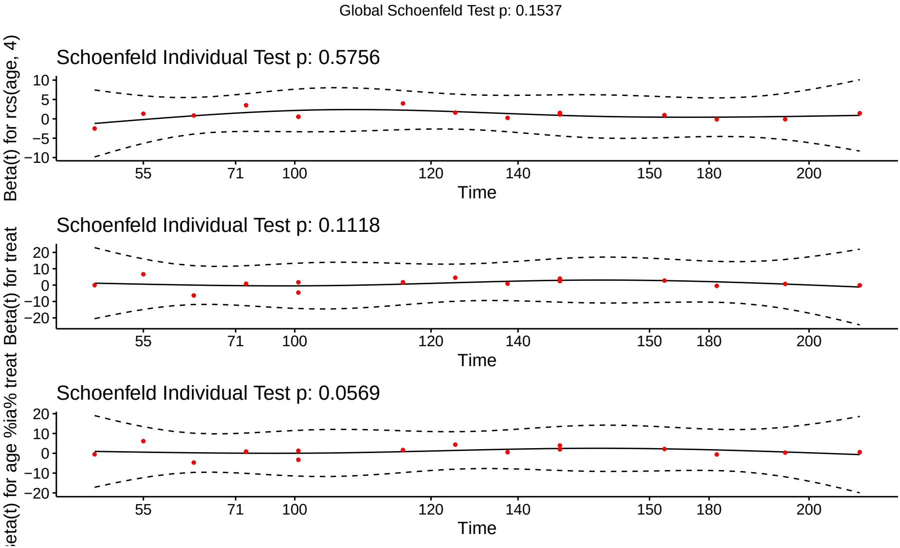


Checking the Proportional Hazards Assumption

```r
cox.zph(cph_AxT, transform = "km", global = TRUE)
```

```
                chisq df     p
rcs(age, 4)      1.98  3 0.576
treat            4.38  2 0.112
age %ia% treat   5.73  2 0.057
GLOBAL          10.67  7 0.154
```

```
ggcoxzph(cox.zph(cph_AxT))
```

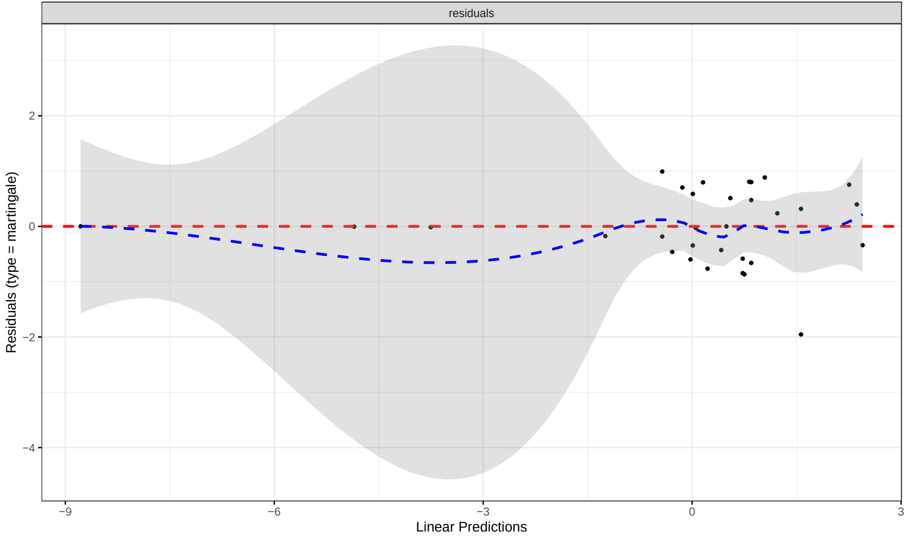


Additional Diagnostic Plots for your Cox model?

- `survminer` has a function called `ggcoxdiagnostics()` which plots different types of residuals as a function of time, linear predictor or observation id.
  - See the default graph (which shows martingale residuals) on the next slide.
- Available types of diagnostics that this can plot are specified with the `type` parameter, that takes any of the following options.

```
type = c("martingale", "deviance", "score", "schoenfeld",
         "dfbeta", "dfbetas", "scaledsch", "partial")
```

```
ggcoxdiagnostics(cph_AxT)
```



New Topic: An Introduction to Robust Linear Regression Methods

Robust Linear Regression Methods

- The `crimestat` data
- Robust Linear Regression Methods
    - with Huber weights
    - with bisquare weights (biweights)
    - Quantile Regression on the Median

Additional Packages for this work

```
library(MASS); library(robustbase); library(boot)
library(quantreg); library(lmtest); library(sandwich)
library(conflicted)

conflict_prefer("select", "dplyr")
conflict_prefer("summarize", "dplyr")

library(tidyverse)
```

The `crimestat` data

For each of 51 states (including the District of Columbia), we have the state's ID number, postal abbreviation and full name, as well as:

- **crime** - the violent crime rate per 100,000 people
- **poverty** - the official poverty rate (% of people living in poverty in the state/district) in 2014
- **single** - the percentage of households in the state/district led by a female householder with no spouse present and with her own children under 18 years living in the household in 2016

The crimestat data set

```
crimestat <- read_csv("data/crimestat.csv")
crimestat
```

```
# A tibble: 51 x 6
     sid state crime poverty single state.full
   <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>
 1     1 AL     427.    19.2   9.02 Alabama
 2     2 AK     636.    11.4   7.63 Alaska
 3     3 AZ     400.    18.2   8.31 Arizona
 4     4 AR     480.    18.7   9.41 Arkansas
 5     5 CA     396.    16.4   7.25 California
 6     6 CO     309.    12.1   6.75 Colorado
 7     7 CT     237.    10.8   8.04 Connecticut
 8     8 DE     489.    13     6.52 Delaware
 9     9 DC    1244.    18.4   8.41 District of Columbia
10    10 FL     540.    16.6   8.29 Florida
# ... with 41 more rows
```

Modeling `crime` with `poverty` and `single`

Our main goal will be to build a linear regression model to predict **crime** using centered versions of both **poverty** and **single**.

```
crimestat <- crimestat %>%
    mutate(pov_c = poverty - mean(poverty),
           single_c = single - mean(single))
```

Our original (OLS) model

Note the sneaky trick with the outside parentheses...

```
(mod1 <- lm(crime ~ pov_c + single_c, data = crimestat))
```

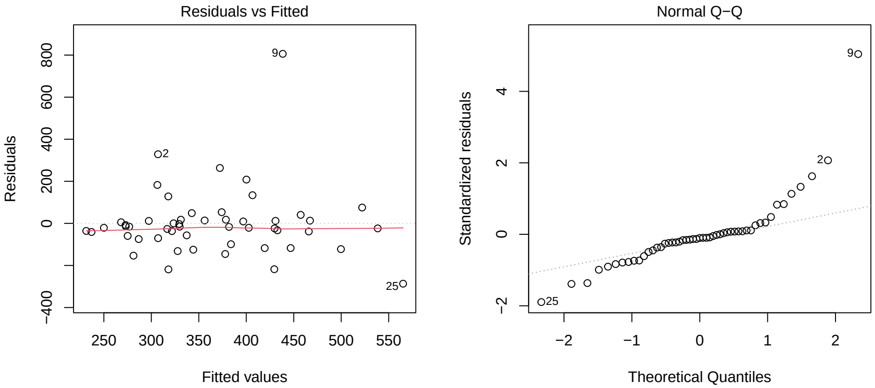
```
Call:
lm(formula = crime ~ pov_c + single_c, data = crimestat)

Coefficients:
(Intercept)        pov_c     single_c
     364.41        16.11        23.84
```

Coefficients?

```
tidy(mod1, conf.int = TRUE) %>%
  select(term, estimate, std.error,
         p.value, conf.low, conf.high) %>%
  kable(digits = 3)
```

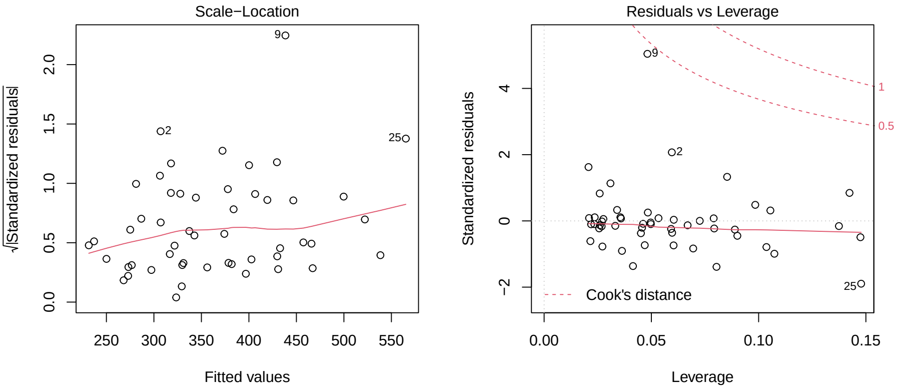
term	estimate	std.error	p.value	conf.low	conf.high
(Intercept)	364.406	22.933	0.000	318.297	410.515
pov_c	16.115	9.616	0.100	-3.219	35.448
single_c	23.843	18.384	0.201	-13.121	60.807

OLS Residuals



Which points are highlighted here?

Remaining Residual Plots from OLS



So which points are of special interest?

Which points are those?

```
crimestat %>%
  slice(c(2, 9, 25))
```

```
# A tibble: 3 x 8
    sid state crime poverty single state.full pov_c single_c
  <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>      <dbl>    <dbl>
1     2 AK     636.    11.4   7.63 Alaska     -3.47  -0.0588
2     9 DC    1244.    18.4   8.41 District ~  3.53   0.721
3    25 MS     278.    21.9  11.4  Mississip~  7.03   3.67
```

Robust Linear Regression with Huber weights

There are several ways to do robust linear regression using M-estimation, including weighting using Huber and bisquare strategies.

- Robust linear regression here will make use of a method called iteratively re-weighted least squares (IRLS) to estimate models.
- M-estimation defines a weight function which is applied during estimation.
- The weights depend on the residuals and the residuals depend on the weights, so an iterative process is required.

We'll fit the model, using the default weighting choice: what are called Huber weights, where observations with small residuals get a weight of 1, and the larger the residual, the smaller the weight.

Our robust model (using `MASS::rlm`)

```
rob.huber <- rlm(crime ~ pov_c + single_c, data = crimestat)
```

Summary of the robust (Huber weights) model

```
tidy(rob.huber) %>%
  kable(digits = 3)
```

term	estimate	std.error	statistic
(Intercept)	343.798	13.131	26.182
pov_c	11.910	5.506	2.163
single_c	30.987	10.527	2.944

Now, *both* predictors appear to have estimates that exceed twice their standard error. So this is a very different result than ordinary least squares gave us.

Glance at the robust model (vs. OLS)

```
glance(mod1)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic p.value    df
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>
1     0.197         0.163  164.      5.88 0.00518     2
# ... with 6 more variables: logLik <dbl>, AIC <dbl>,
#   BIC <dbl>, deviance <dbl>, df.residual <int>,
#   nobs <int>
```

```
glance(rob.huber)
```

```
# A tibble: 1 x 7
  sigma converged logLik      AIC   BIC deviance  nobs
  <dbl> <lgl>     <logLik>  <dbl> <dbl>    <dbl> <int>
1  59.1 TRUE      -331.3785  671.  678. 1314784.    51
```

Understanding the Huber weights a bit

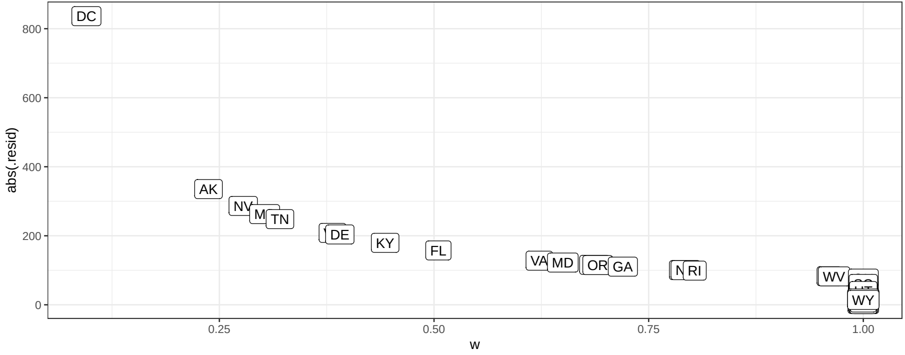
Let's augment the data with results from this model, including the weights.

```
crime_with_huber <- augment(rob.huber, crimestat) %>%
    mutate(w = rob.huber$w) %>% arrange(w)

crime_with_huber %>%
  select(sid, state, w, crime, pov_c, single_c, everything())
  head(., 3)
```

```
# A tibble: 3 x 15
    sid state       w crime  pov_c single_c poverty single
  <dbl> <chr>   <dbl> <dbl>  <dbl>    <dbl>   <dbl>  <dbl>
1     9 DC     0.0951 1244.  3.53    0.721     18.4   8.41
2     2 AK     0.237   636. -3.47   -0.0588    11.4   7.63
3    29 NV     0.278   636.  0.527  -0.0288    15.4   7.66
# ... with 7 more variables: state.full <chr>,
#   .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>,
#   .cooksd <dbl>, .std.resid <dbl>
```

Are cases with large residuals down-weighted?

```
ggplot(crime_with_huber, aes(x = w, y = abs(.resid))) +
    geom_label(aes(label = state))
```



Conclusions from the Plot of Weights

- District of Columbia will be down-weighted the most, followed by Alaska and then Nevada and Mississippi.
- But many of the observations will have a weight of 1.
- In ordinary least squares, all observations would have weight 1.
- So the more cases in the robust regression that have a weight close to one, the closer the results of the OLS and robust procedures will be.

summary(rob.huber)

```
Call: rlm(formula = crime ~ pov_c + single_c, data = crimestat
Residuals:
     Min       1Q   Median       3Q      Max
-262.751  -45.641    1.762   36.732  836.244

Coefficients:
            Value    Std. Error t value
(Intercept) 343.7982  13.1309    26.1823
pov_c        11.9098   5.5058     2.1631
single_c     30.9868  10.5266     2.9437

Residual standard error: 59.14 on 48 degrees of freedom
```

Robust Linear Regression with the biweight

As mentioned there are several possible weighting functions - we'll next try the **biweight**, also called the bisquare or Tukey's bisquare, in which all cases with a non-zero residual get down-weighted at least a little. Here is the resulting fit. . .

```
(rob.biweight <- rlm(crime ~ pov_c + single_c,
                     data = crimestat, psi = psi.bisquare))
```

```
Call:
rlm(formula = crime ~ pov_c + single_c, data = crimestat, psi
Converged in 13 iterations

Coefficients:
(Intercept)       pov_c    single_c
  336.17015    10.31578    34.70765

Degrees of freedom: 51 total; 48 residual
Scale estimate: 67.3
```

Coefficients and Standard Errors

```
tidy(rob.biweight) %>% kable(digits = 3)
```

term	estimate	std.error	statistic
(Intercept)	336.170	12.673	26.526
pov_c	10.316	5.314	1.941
single_c	34.708	10.160	3.416

Understanding the biweights weights a bit

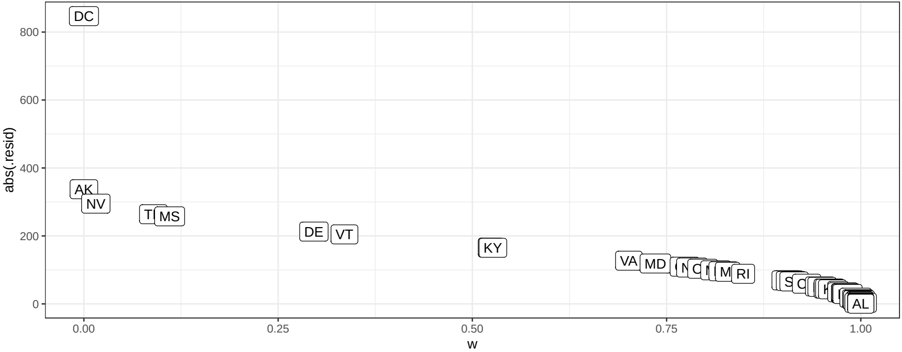
Let's augment the data, as above

```
crime_with_biweights <-
  augment(rob.biweight, newdata = crimestat) %>%
  mutate(w = rob.biweight$w) %>%
  arrange(w)

head(crime_with_biweights, 3)
```

```
# A tibble: 3 x 11
    sid state crime poverty single state.full            pov_c
  <dbl> <chr> <dbl>   <dbl>  <dbl> <chr>                 <dbl>
1     2 AK     636.    11.4   7.63 Alaska               -3.47
2     9 DC    1244.    18.4   8.41 District of Colum~   3.53
3    29 NV     636.    15.4   7.66 Nevada                0.527
# ... with 4 more variables: single_c <dbl>, .fitted <dbl>,
#   .resid <dbl>, w <dbl>
```

Relationship of Weights and Residuals

```r
ggplot(crime_with_biweights, aes(x = w, y = abs(.resid))) +
    geom_label(aes(label = state))
```



Conclusions from the biweights plot

Again, cases with large residuals (in absolute value) are down-weighted generally, but here, Alaska and Washington DC receive no weight at all in fitting the final model.

- We can see that the weight given to DC and Alaska is dramatically lower (in fact it is zero) using the bisquare weighting function than the Huber weighting function and the parameter estimates from these two different weighting methods differ.
- The maximum weight (here, for Alabama) for any state using the biweight is still slightly smaller than 1.

summary(rob.biweight)

```
Call: rlm(formula = crime ~ pov_c + single_c, data = crimestat
Residuals:
    Min      1Q  Median      3Q     Max
-257.58  -40.53    8.01   45.30  846.81

Coefficients:
            Value    Std. Error t value
(Intercept) 336.1702  12.6733    26.5259
pov_c        10.3158   5.3139     1.9413
single_c     34.7077  10.1598     3.4162

Residual standard error: 67.27 on 48 degrees of freedom
```

Comparing OLS and the two weighting schemes

```
glance(mod1) %>% select(1:6)
```

```
# A tibble: 1 x 6
  r.squared adj.r.squared sigma statistic p.value    df
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>
1     0.197         0.163  164.      5.88 0.00518     2
```

```
glance(mod1) %>% select(7:12)
```

```
# A tibble: 1 x 6
  logLik   AIC   BIC deviance df.residual  nobs
   <dbl> <dbl> <dbl>    <dbl>       <int> <int>
1  -331.  670.  677. 1287405.          48    51
```

Comparing OLS and the two weighting schemes

```
glance(rob.biweight) # biweights
```

```
# A tibble: 1 x 7
  sigma converged logLik      AIC   BIC deviance  nobs
  <dbl> <lgl>     <logLik>  <dbl> <dbl>    <dbl> <int>
1  67.3 TRUE      -331.8601  672.  679. 1339850.    51
```

```
glance(rob.huber) # Huber weights
```

```
# A tibble: 1 x 7
  sigma converged logLik      AIC   BIC deviance  nobs
  <dbl> <lgl>     <logLik>  <dbl> <dbl>    <dbl> <int>
1  59.1 TRUE      -331.3785  671.  678. 1314784.    51
```

Quantile Regression on the Median

We can use the `rq` function in the `quantreg` package to model the **median** of our outcome (violent crime rate) on the basis of our predictors, rather than the mean, as is the case in ordinary least squares.

```
rob.quan <- rq(crime ~ pov_c + single_c, data = crimestat)

glance(rob.quan)
```

```
# A tibble: 1 x 5
    tau logLik       AIC   BIC df.residual
  <dbl> <logLik>  <dbl> <dbl>       <int>
1   0.5 -315.7569  638.  643.           48
```

summary(rob.quan)

```
Call: rq(formula = crime ~ pov_c + single_c, data = crimestat)

tau: [1] 0.5

Coefficients:
            coefficients lower bd  upper bd
(Intercept) 344.75658    336.94534 366.23603
pov_c        10.54757      3.06714  28.95962
single_c     32.27249      4.45889  48.18925
```

Estimating a different quantile (tau = 0.70)

In fact, if we like, we can estimate any quantile by specifying the `tau` parameter (here `tau` = 0.5, by default, so we estimate the median.)

```
(rob.quan70 <- rq(crime ~ pov_c + single_c, tau = 0.70,
                  data = crimestat))
```

```
Call:
rq(formula = crime ~ pov_c + single_c, tau = 0.7, data = crime

Coefficients:
(Intercept)       pov_c    single_c
  379.72818    19.30376    32.15827

Degrees of freedom: 51 total; 48 residual
```

Comparing our Four Models

Estimating the Mean

Fit	Intercept CI	`pov_c` CI	`single_c` CI
OLS	(318.6, 410.2)	(-3.13, 35.35)	(-12.92, 60.60)
Robust (Huber)	(320.0, 367.6)	(0.89, 22.93)	(9.93, 52.05)
Robust (biweight)	(310.7, 361.5)	(-0.30, 20.94)	(14.39, 55.03)

Note: CIs estimated for OLS and Robust methods as point estimate $\pm$ 2 standard errors

Estimating the Median

Fit	Intercept CI	`pov_c` CI	`single_c` CI
Quantile (Median) Reg	(336.9, 366.2)	(3.07, 28.96)	(4.46, 48,19)

Comparing AIC and BIC

Fit	AIC	BIC
OLS	669.7	677.4
Robust (Huber)	670.8	678.5
Robust (biweight)	671.7	679.4
Quantile (median)	637.5	643.3

Some General Thoughts

1. When comparing the results of a regular OLS regression and a robust regression for a data set which displays outliers, if the results are very different, you will most likely want to use the results from the robust regression.
    - Large differences suggest that the model parameters are being highly influenced by outliers.
2. Different weighting functions have advantages and drawbacks.
    - Huber weights can have difficulties with really severe outliers.
    - Bisquare weights can have difficulties converging or may yield multiple solutions.
    - Quantile regression approaches have some nice properties, but describe medians (or other quantiles) rather than means.