Estimation of Prediction Error in Linear Models

Alexander McLain

August 30, 2021

```
## Registered S3 method overwritten by 'printr':
##   method                from
##   knit_print.data.frame rmarkdown
```

This example will use the bodyfat data from the textbook. First we will read in the data, then look at some summaries. Here's a link to some info about it here.

```
bf_dat <- read.csv("bodyfat2.csv")
bf_df <- data.frame(bf_dat)
head(bf_df)
```

density	bodyfat	age	weight	height	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist
1.0708	12.3	23	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1
1.0853	6.1	22	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2
1.0414	25.3	22	154.00	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6
1.0751	10.4	26	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2
1.0340	28.7	24	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7
1.0502	20.9	24	210.25	74.75	39.0	104.5	94.4	107.8	66.0	42.0	25.6	35.7	30.6	18.8

Second, we'll look at the correlation matrix of the data:

```
round(cor(bf_df)[1:8,1:8],3)
```

	density	bodyfat	age	weight	height	neck	chest	abdomen
density	1.000	-0.999	-0.290	-0.613	0.019	-0.491	-0.703	-0.812
bodyfat	-0.999	1.000	0.291	0.612	-0.025	0.491	0.703	0.813
age	-0.290	0.291	1.000	-0.013	-0.245	0.114	0.176	0.230
weight	-0.613	0.612	-0.013	1.000	0.487	0.831	0.894	0.888
height	0.019	-0.025	-0.245	0.487	1.000	0.321	0.227	0.190
neck	-0.491	0.491	0.114	0.831	0.321	1.000	0.785	0.754
chest	-0.703	0.703	0.176	0.894	0.227	0.785	1.000	0.916
abdomen	-0.812	0.813	0.230	0.888	0.190	0.754	0.916	1.000

Third, we'll fit a simple linear model to the data

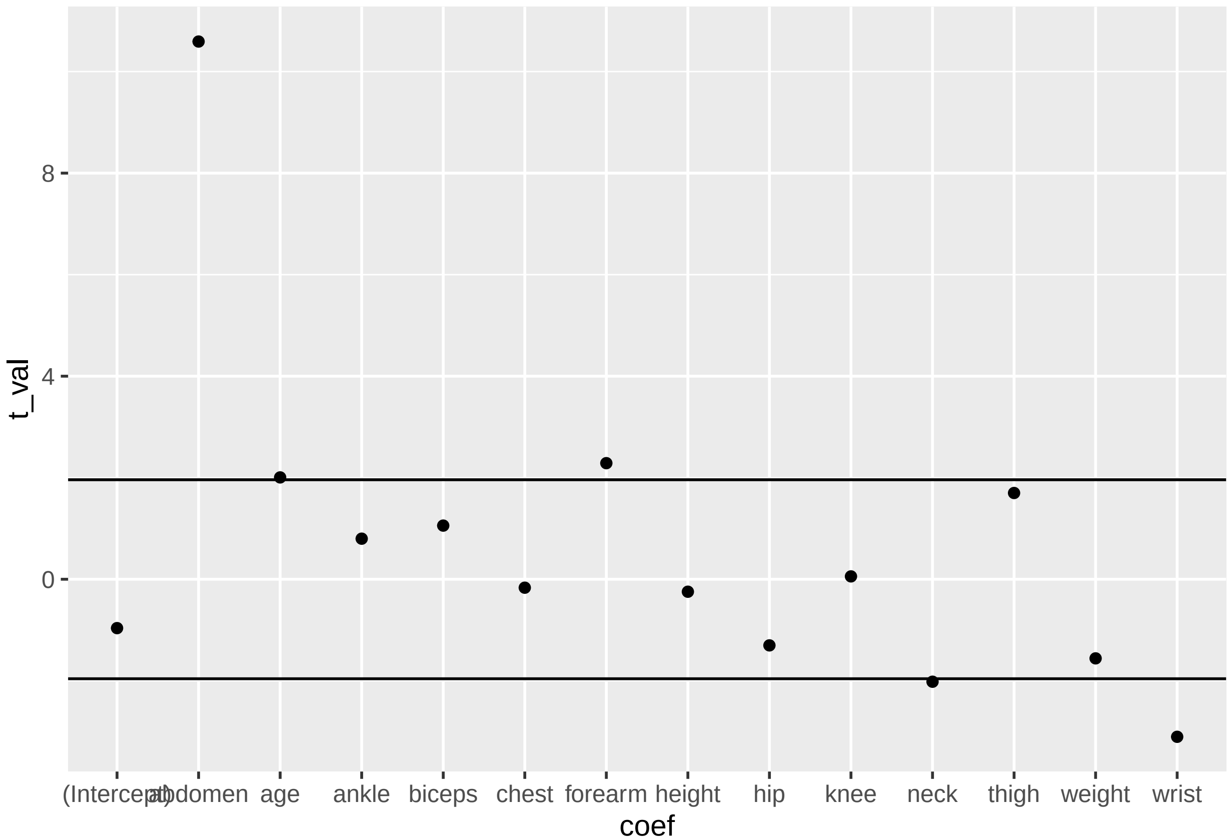
```
bf_mod <- lm(bodyfat~age + weight + height + neck + chest + abdomen +
               hip + thigh + knee + ankle + biceps + forearm + wrist,data = bf_df)
summary(bf_mod)
```

```
##
## Call:
## lm(formula = bodyfat ~ age + weight + height + neck + chest +
```

```
##     abdomen + hip + thigh + knee + ankle + biceps + forearm +
##     wrist, data = bf_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1966  -2.8824  -0.1111   3.1901   9.9979
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.35323   22.18616  -0.962  0.33680
## age           0.06457    0.03219   2.006  0.04601 *
## weight       -0.09638    0.06185  -1.558  0.12047
## height       -0.04394    0.17870  -0.246  0.80599
## neck         -0.47547    0.23557  -2.018  0.04467 *
## chest        -0.01718    0.10322  -0.166  0.86792
## abdomen       0.95500    0.09016  10.592  < 2e-16 ***
## hip          -0.18859    0.14479  -1.302  0.19401
## thigh         0.24835    0.14617   1.699  0.09061 .
## knee          0.01395    0.24775   0.056  0.95516
## ankle         0.17788    0.22262   0.799  0.42505
## biceps        0.18230    0.17250   1.057  0.29166
## forearm       0.45574    0.19930   2.287  0.02309 *
## wrist        -1.65450    0.53316  -3.103  0.00215 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.309 on 238 degrees of freedom
## Multiple R-squared:  0.7486, Adjusted R-squared:  0.7348
## F-statistic:  54.5 on 13 and 238 DF,  p-value: < 2.2e-16
```

```
t_vals <- data.frame(t_val = bf_mod$coefficients/(coef(summary(bf_mod))[,2]),
                     coef = names(bf_mod$coefficients))
```

Plot the resulting T-values:



Now let's calculate the coefficient for 'wrist' using the GS algorithm

```
wrist_mod <- lm(wrist~age + weight + height + neck + chest + abdomen +
               hip + thigh + knee + ankle + biceps + forearm,data = bf_df)
z <- wrist_mod$residuals
bf_wrist_mod <- lm(bf_df$bodyfat ~ z -1)
summary(bf_wrist_mod)
```

```
##
## Call:
## lm(formula = bf_df$bodyfat ~ z - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.329 12.728 19.389 25.173 47.281
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## z   -1.654      2.588  -0.639    0.523
##
## Residual standard error: 20.92 on 251 degrees of freedom
## Multiple R-squared:  0.001626,   Adjusted R-squared:  -0.002352
## F-statistic: 0.4088 on 1 and 251 DF,  p-value: 0.5232
```

```
## Ratio of estimated RMSE
20.92/4.309
```

```
## [1] 4.854955
```

```r
## Ratio of estimated Std Err of wrist
2.588/0.53316
```

```
## [1] 4.854078
```

Estimating Prediction Error

Here RSS_n $= \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n}$.

```r
RSS_n <- mean(bf_mod$residuals^2)
RSS_n
```

```
## [1] 17.53994
```

Let see how that compares to the CV estimate. To do this we will:

- create a function that will do K-fold CV sampling of the data ($K = 2, 3, \ldots, n$).
- execute a linear model for each of the K-fold samples
- estimate the prediction error for each of the K-fold samples

Here is the function to do the K-fold sampling:

$\vdots$

$\vdots$

$\vdots$

Where'd it go? Let's see it work.

```r
CV_ids <- CV_sampl(bf_df,10)
CV_ids$ids[1:20]
```

```
##  [1] 2 4 8 9 4 3 5 2 7 6 4 7 3 8 2 1 3 7 2 3
```

Now to do the CV for each model:

```r
#Which CV's will we do:
CV <- c(3,5,10,20,length(bf_df[,1]))
#Set the seed so we can replicate
set.seed(4)
PE_est <- RSS_n
for(k in CV){
  #Get which group each subject is in.
  ids <-  CV_sampl(bf_df,k)$ids
  t_PE_est <- NULL
  for(j in 1:k){
    #Get jth leaning and test datasets, and estimate LM
    learning_data <- bf_df[ids!=j,]
    test_data <- bf_df[ids==j,]
    bf_mod_CV <- lm(bodyfat ~ age + weight + height + neck + chest +
                      abdomen + hip + thigh + knee + ankle + biceps +
                      forearm + wrist,data = learning_data)
    #Predict Y_hat for the new data.
    new_Yhat <- predict(bf_mod_CV,test_data)
    #Estimate the error.
    new_RSS_n <- mean((test_data$bodyfat - new_Yhat)^2)
```

```r
    t_PE_est <- c(t_PE_est,new_RSS_n)
  }
  PE_est <- c(PE_est,mean(t_PE_est))
}
PE_res <- data.frame(K = c(0,CV), EPE = PE_est)
PE_res
```

K	EPE
0	17.53994
3	21.30789
5	19.72081
10	20.83760
20	21.10300
252	20.29476