# BSC04

Pan Gaojie

2023-11-14

## 目录

## Improve the stochastic gradient descent function.

```
sgd.lm <- function(X, y, beta.init, n.samples = 1, tol = 1e-05, max.iter = 100) {
    n <- length(y)
    beta.old <- beta.init
    J <- betas <- list()
    sto.sample <- sample(1:n, n.samples, replace = TRUE)
    alpha <- best.alpha(X, y, beta.old,sto.sample)
    # print(alpha)
    betas[[1]] <- beta.old
    J[[1]] <- sgd.lm.cost(X, y, beta.old)
    beta.new <- beta.old - alpha * sgd.lm.cost.grad(X[sto.sample, ], y[sto.sample],
        beta.old)
    betas[[2]] <- beta.new
    J[[2]] <- sgd.lm.cost(X, y, beta.new)
    iter <- 0
    n.best <- 0
    while ((abs(sgd.lm.cost(X, y, beta.new) - sgd.lm.cost(X, y, beta.old)) > tol) & (iter +
        2 < max.iter)) {
      beta.old <- beta.new
      sto.sample <- sample(1:n, n.samples, replace = TRUE)
      alpha <- best.alpha(X, y, beta.old,sto.sample)
      # print(alpha)
      beta.new <- beta.old - alpha * sgd.lm.cost.grad(X[sto.sample, ], y[sto.sample],
          beta.old)
      iter <- iter + 1
      betas[[iter + 2]] <- beta.new
      J[[iter + 2]] <- sgd.lm.cost(X, y, beta.new)
    }
    if (abs(sgd.lm.cost(X, y, beta.new) - sgd.lm.cost(X, y, beta.old)) > tol) {
        cat("Could not converge. \n")
    } else {
        cat("Converged. \n")
        cat("Iterated", iter + 1, "times.", "\n")
        cat("Coef: ", beta.new, "\n")
        return(list(coef = betas, cost = J, niter = iter + 1))
    }
}

## Make the cost function
sgd.lm.cost <- function(X, y, beta) {
    n <- length(y)
    if (!is.matrix(X)) {
        X <- matrix(X, nrow = 1)
    }
    loss <- sum((X %*% beta - y)^2)/(2 * n)
    return(loss)
}
## Calculate the gradient
sgd.lm.cost.grad <- function(X, y, beta) {
    n <- length(y)
    if (!is.matrix(X)) {
        X <- matrix(X, nrow = 1)
    }
    t(X) %*% (X %*% beta - y)/n
}
```

During the regression progress, I found that the initial learning ratio $\alpha = 0.5$ is not suitable for "bodyfat" dataset. The sgd.lm() function yield such error information:

It seems that the value of $\alpha$ do matters a lot. Here I coded a function to determine the best $\alpha$ automatically, in effort to avoid such error occurring again.

```
best.alpha <- function(X, y, beta.old,sto.sample){
  alpha <- optim(0.1, function(alpha) {
    sgd.lm.cost(X, y, beta.old - alpha * sgd.lm.cost.grad(X[sto.sample, ], y[sto.sample],
        beta.old))}, lower=0,upper=1,method = "L-BFGS-B")
  if (alpha$convergence == 0) {
    alpha <- alpha$par
  } else {
    alpha <- 0.1
  }
  if(alpha<0.01){
    alpha <- 0.01
  }
  return(alpha)
}
```

## Variable Selection.

Import the "BAS" package which contains the dataset "bodyfat".

```
if (!requireNamespace("BAS", quietly = TRUE)) install.packages("BAS")
library(BAS)
```

```
## Warning: 程辑包'BAS'是用R版本4.3.2 来建造的
```

```
data(bodyfat)
head(bodyfat,3)
```

```
##   Density Bodyfat Age Weight Height Neck Chest Abdomen  Hip Thigh Knee Ankle
## 1  1.0708    12.3  23 154.25  67.75 36.2  93.1    85.2 94.5  59.0 37.3  21.9
## 2  1.0853     6.1  22 173.25  72.25 38.5  93.6    83.0 98.7  58.7 37.3  23.4
## 3  1.0414    25.3  22 154.00  66.25 34.0  95.8    87.9 99.2  59.6 38.9  24.0
##   Biceps Forearm Wrist
## 1   32.0    27.4  17.1
## 2   30.5    28.9  18.2
## 3   28.8    25.2  16.6
```

Pre-modelling on dataset “bodyfat”. Select significant variables by using lm() and Backward Selection.

```
bod <- scale(bodyfat[,-c(1,2)])
bod <- as.data.frame(cbind(Bodyfat=bodyfat$Bodyfat,bod))
lm.fit <- lm(Bodyfat ~.,data=bod)
lm.step<-step(lm.fit,direction = "backward")
```

```
## Start:  AIC=749.36
## Bodyfat ~ Age + Weight + Height + Neck + Chest + Abdomen + Hip +
##     Thigh + Knee + Ankle + Biceps + Forearm + Wrist
##
##            Df Sum of Sq    RSS    AIC
## - Knee      1      0.07 4411.5 747.36
## - Chest     1      1.07 4412.5 747.42
## - Height    1      9.74 4421.2 747.91
## - Ankle     1     11.44 4422.9 748.01
## - Biceps    1     20.87 4432.3 748.55
## <none>                  4411.4 749.36
## - Hip       1     37.50 4448.9 749.49
## - Thigh     1     49.58 4461.0 750.17
## - Weight    1     50.61 4462.1 750.23
## - Age       1     68.26 4479.7 751.23
## - Neck      1     75.96 4487.4 751.66
## - Forearm   1     95.51 4507.0 752.76
## - Wrist     1    170.12 4581.6 756.89
## - Abdomen   1   2260.95 6672.4 851.63
##
## Step:  AIC=747.36
## Bodyfat ~ Age + Weight + Height + Neck + Chest + Abdomen + Hip +
##     Thigh + Ankle + Biceps + Forearm + Wrist
##
##            Df Sum of Sq    RSS    AIC
## - Chest     1      1.13 4412.7 745.43
## - Height   1      9.66 4421.2 745.91
## - Ankle    1     12.09 4423.6 746.05
## - Biceps   1     20.81 4432.3 746.55
## <none>                 4411.5 747.36
## - Hip      1     37.43 4448.9 747.49
## - Weight   1     53.08 4464.6 748.38
## - Thigh    1     54.88 4466.4 748.48
## - Age      1     74.06 4485.6 749.56
## - Neck     1     78.44 4490.0 749.80
## - Forearm  1     96.77 4508.3 750.83
## - Wrist    1    170.55 4582.1 754.92
## - Abdomen  1   2269.88 6681.4 849.97
##
## Step:  AIC=745.43
## Bodyfat ~ Age + Weight + Height + Neck + Abdomen + Hip + Thigh +
##     Ankle + Biceps + Forearm + Wrist
##
##           Df Sum of Sq    RSS    AIC
## - Height   1      8.68 4421.3 743.92
## - Ankle    1     12.41 4425.1 744.13
## - Biceps   1     20.12 4432.8 744.57
## <none>                 4412.7 745.43
## - Hip      1     36.30 4449.0 745.49
## - Thigh    1     60.09 4472.7 746.83
## - Weight   1     70.84 4483.5 747.44
## - Age      1     73.84 4486.5 747.61
## - Neck     1     79.48 4492.1 747.93
## - Forearm  1     95.64 4508.3 748.83
## - Wrist    1    169.98 4582.6 752.95
## - Abdomen  1   2879.44 7292.1 870.01
##
## Step:  AIC=743.92
## Bodyfat ~ Age + Weight + Neck + Abdomen + Hip + Thigh + Ankle +
##     Biceps + Forearm + Wrist
##
##           Df Sum of Sq    RSS    AIC
## - Ankle    1      13.3 4434.6 742.68
## - Biceps   1      22.4 4443.7 743.19
## - Hip      1      30.4 4451.8 743.65
## <none>                 4421.3 743.92
## - Thigh    1      68.8 4490.1 745.81
## - Neck     1      77.1 4498.4 746.27
## - Age      1      81.3 4502.6 746.51
## - Forearm  1      98.1 4519.4 747.45
## - Weight   1     119.6 4540.9 748.65
## - Wrist    1     181.3 4602.6 752.05
## - Abdomen  1    3178.5 7599.9 878.43
##
## Step:  AIC=742.68
## Bodyfat ~ Age + Weight + Neck + Abdomen + Hip + Thigh + Biceps +
##     Forearm + Wrist
##
##           Df Sum of Sq    RSS    AIC
## - Biceps   1      20.7 4455.3 741.85
## - Hip      1      31.7 4466.4 742.47
## <none>                 4434.6 742.68
## - Thigh    1      72.3 4506.9 744.75
## - Age      1      77.6 4512.2 745.05
## - Neck     1      87.3 4521.9 745.59
## - Forearm  1      97.4 4532.0 746.15
## - Weight   1     107.2 4541.8 746.69
## - Wrist    1     168.0 4602.6 750.05
## - Abdomen  1    3182.0 7616.7 876.98
##
## Step:  AIC=741.85
## Bodyfat ~ Age + Weight + Neck + Abdomen + Hip + Thigh + Forearm +
##     Wrist
##
##           Df Sum of Sq    RSS    AIC
## <none>                 4455.3 741.85
## - Hip      1      36.5 4491.8 741.91
## - Neck     1      79.1 4534.4 744.29
## - Age      1      83.8 4539.1 744.55
## - Weight   1      93.0 4548.3 745.05
## - Thigh    1     100.7 4556.0 745.48
## - Forearm  1     140.5 4595.8 747.67
## - Wrist    1     166.8 4622.2 749.12
## - Abdomen  1    3163.0 7618.3 875.04
```

Now we obtained the model: Bodyfat ~ Age + Weight + Neck + Abdomen + Hip + Thigh + Forearm + Wrist.

## Linear regression using stochastic gradient descent method.

```r
summary(lm.step)
```

```
##
## Call:
## lm(formula = Bodyfat ~ Age + Weight + Neck + Abdomen + Hip +
##     Thigh + Forearm + Wrist, data = bod)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9757  -2.9937  -0.1644   2.9766  10.2244
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.1508     0.2697  70.999  < 2e-16 ***
## Age           0.8290     0.3878   2.137  0.03356 *
## Weight       -2.6407     1.1728  -2.252  0.02524 *
## Neck         -1.1342     0.5460  -2.077  0.03884 *
## Abdomen      10.1880     0.7757  13.134  < 2e-16 ***
## Hip          -1.4001     0.9920  -1.411  0.15940
## Thigh         1.5875     0.6775   2.343  0.01992 *
## Forearm       1.0421     0.3765   2.768  0.00607 **
## Wrist        -1.4346     0.4756  -3.017  0.00283 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.282 on 243 degrees of freedom
## Multiple R-squared:  0.7466, Adjusted R-squared:  0.7382
## F-statistic: 89.47 on 8 and 243 DF,  p-value: < 2.2e-16
```

Performing regression on the model above using the Stochastic Gradient Descent (SGD) method.

```
y <- as.matrix(bod$Bodyfat)
x <- scale(bod[,c(2,3,5,7,8,9,13,14)])
X <- as.matrix(cbind(intercept=1,x))
init <- rep(0,ncol(X))
sgd.bodyfat <- sgd.lm(X, y, beta.init = init, tol = 1e-05, max.iter = 10000)
```

```
## Converged.
## Iterated 1513 times.
## Coef:  19.21415 0.8997107 -0.875843 -1.301384 9.551706 -1.535313 0.7551379 0.9921229 -1.538117
```

The results yielded by SGD are close to those obtained by lm().

## Ploting the regression progress.

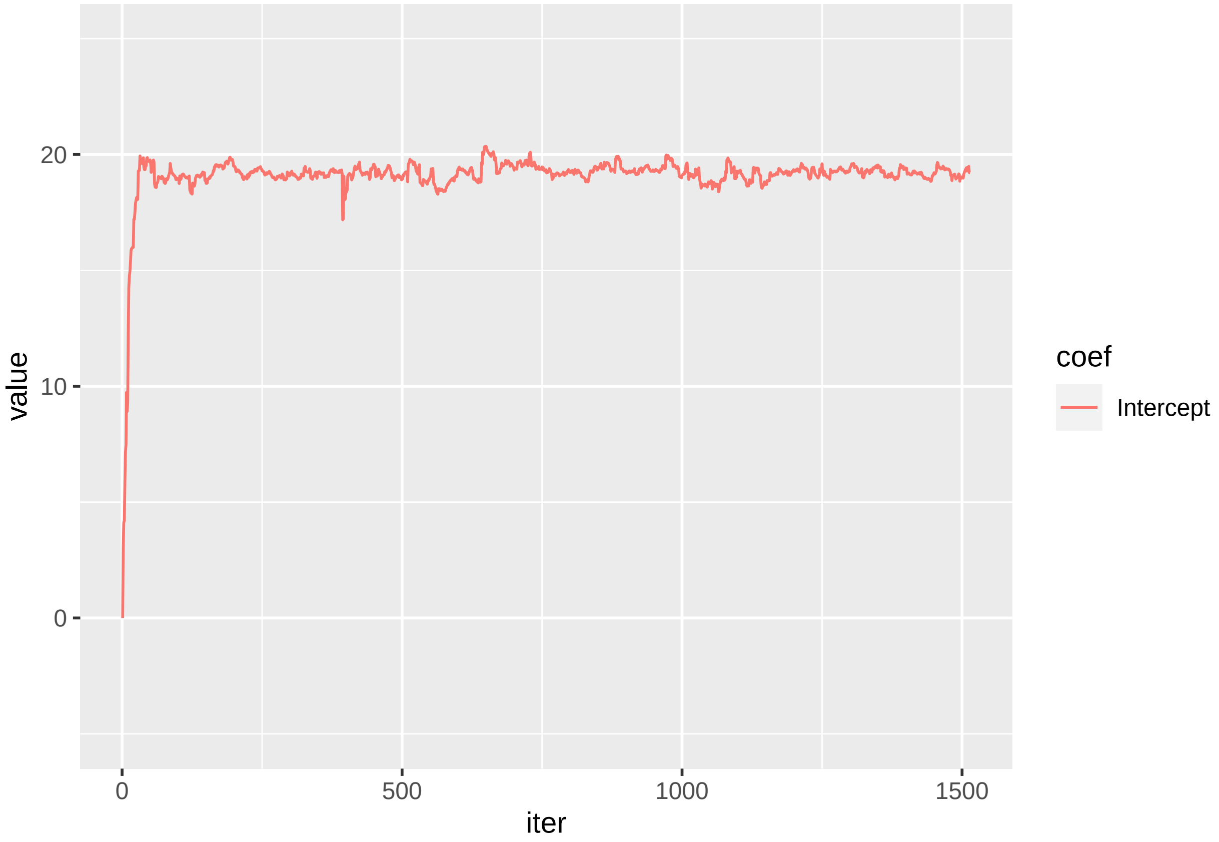
```
library(tidyverse)
```

```
## Warning: 程辑包'tidyverse'是用R版本4.3.1 来建造的

## Warning: 程辑包'ggplot2'是用R版本4.3.1 来建造的

## Warning: 程辑包'readr'是用R版本4.3.1 来建造的

## Warning: 程辑包'purrr'是用R版本4.3.1 来建造的

## Warning: 程辑包'dplyr'是用R版本4.3.1 来建造的

## Warning: 程辑包'forcats'是用R版本4.3.1 来建造的

## Warning: 程辑包'lubridate'是用R版本4.3.1 来建造的

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

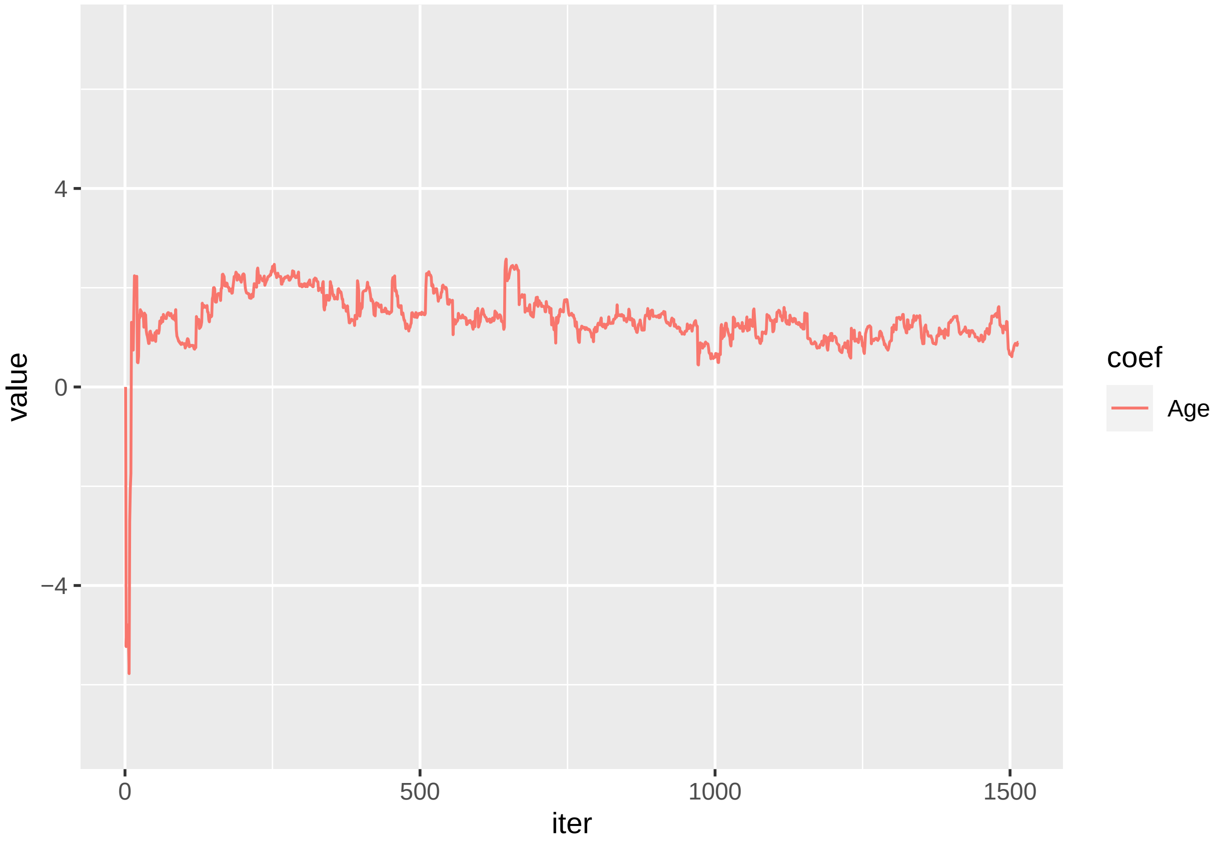
```
library(reshape2)
```

```
## Warning: 程辑包'reshape2'是用R版本4.3.2 来建造的

##
## 载入程辑包: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

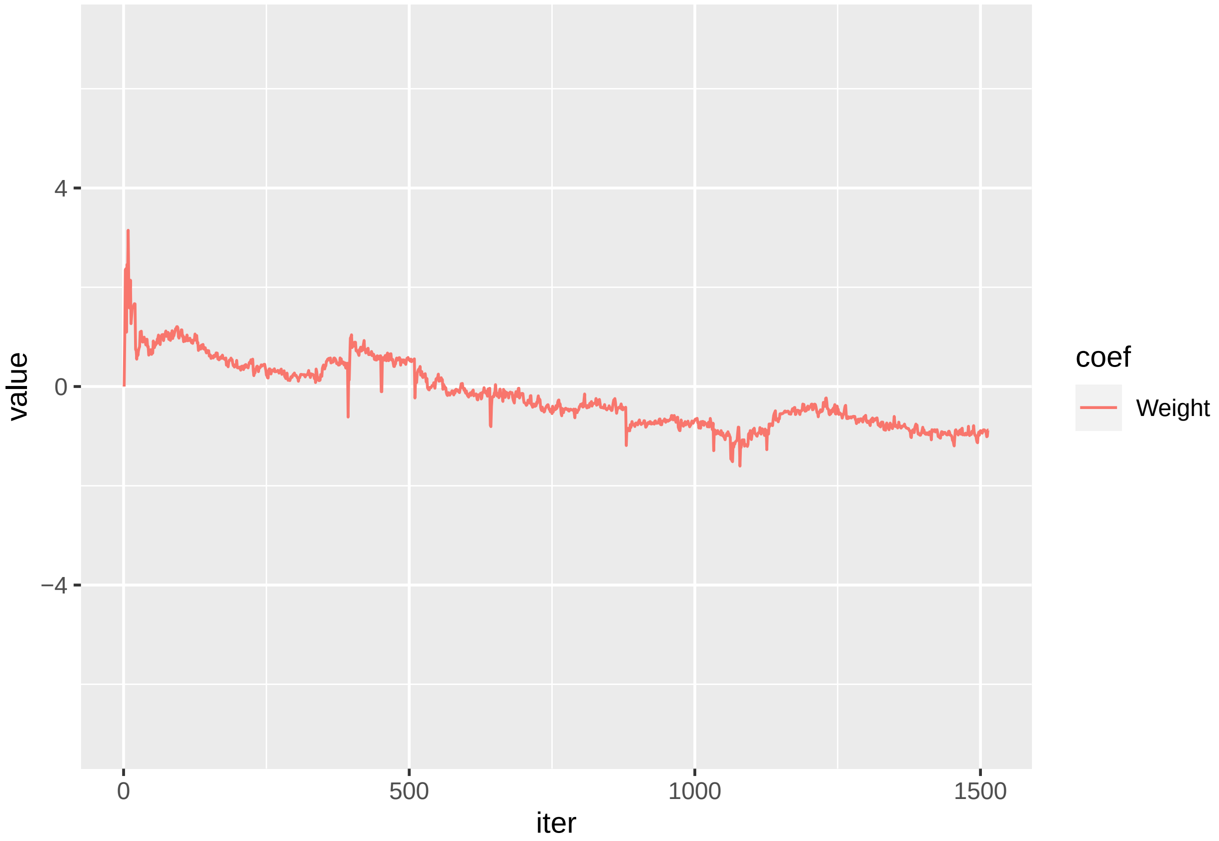
```
beta <- as.data.frame(t(do.call(cbind, sgd.bodyfat$coef)))

betas <- beta %>% select(Intercept=V1) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-5, 25))
```



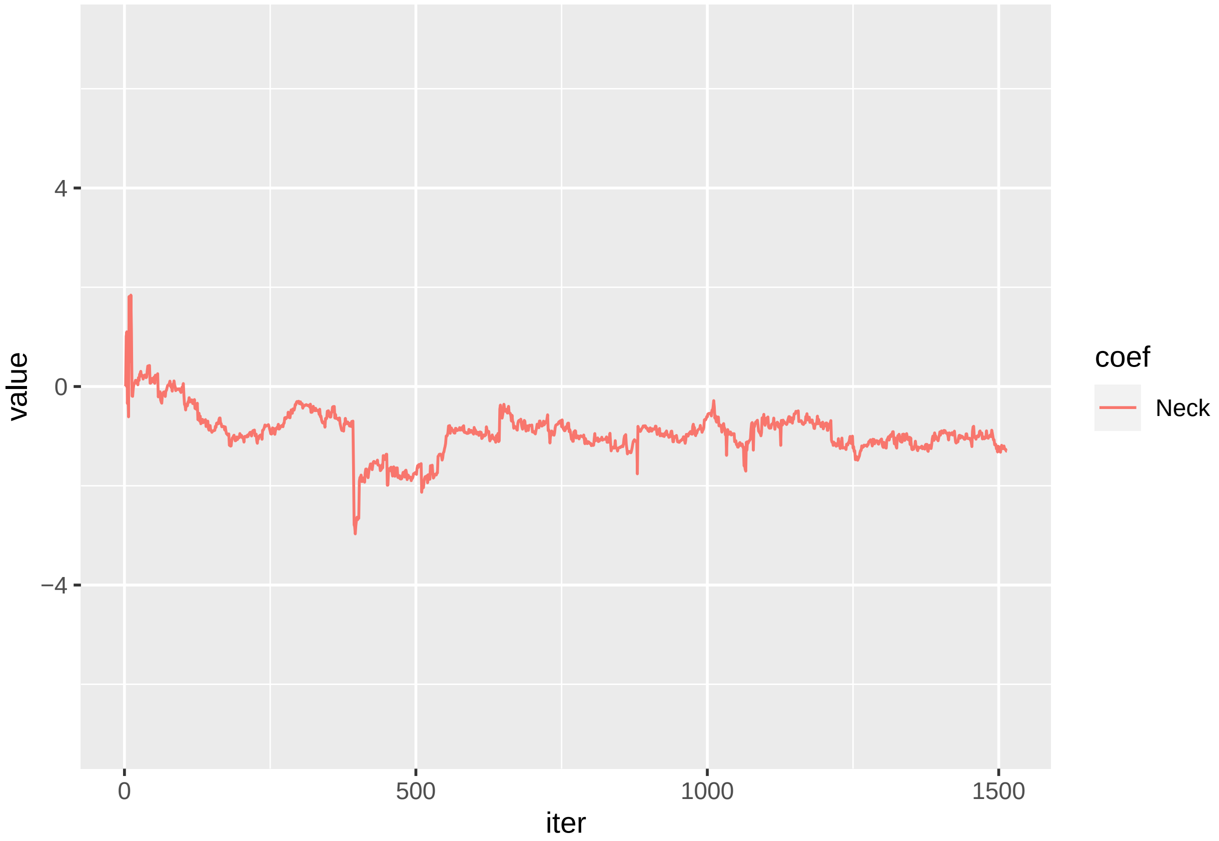
```r
betas <- beta %>% select(Age=V2) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



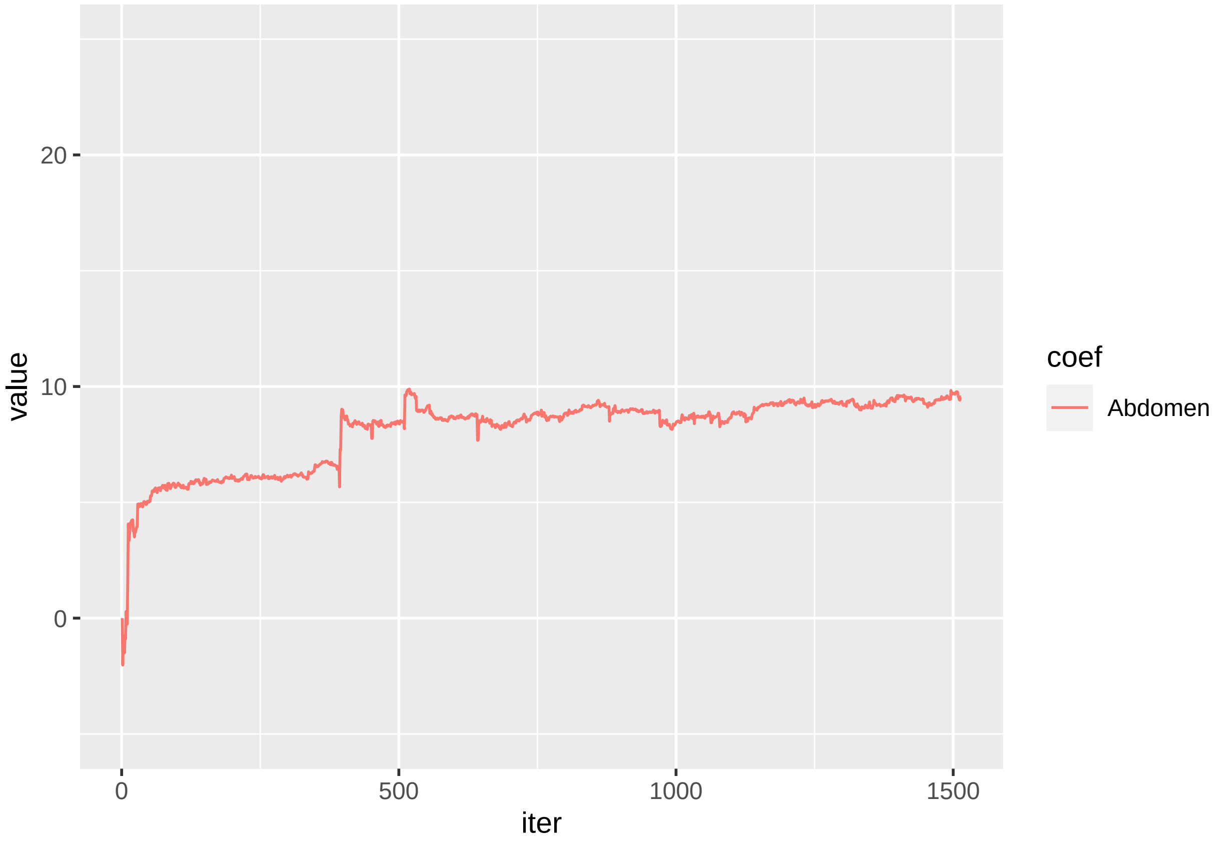
```r
betas <- beta %>% select(Weight=V3) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



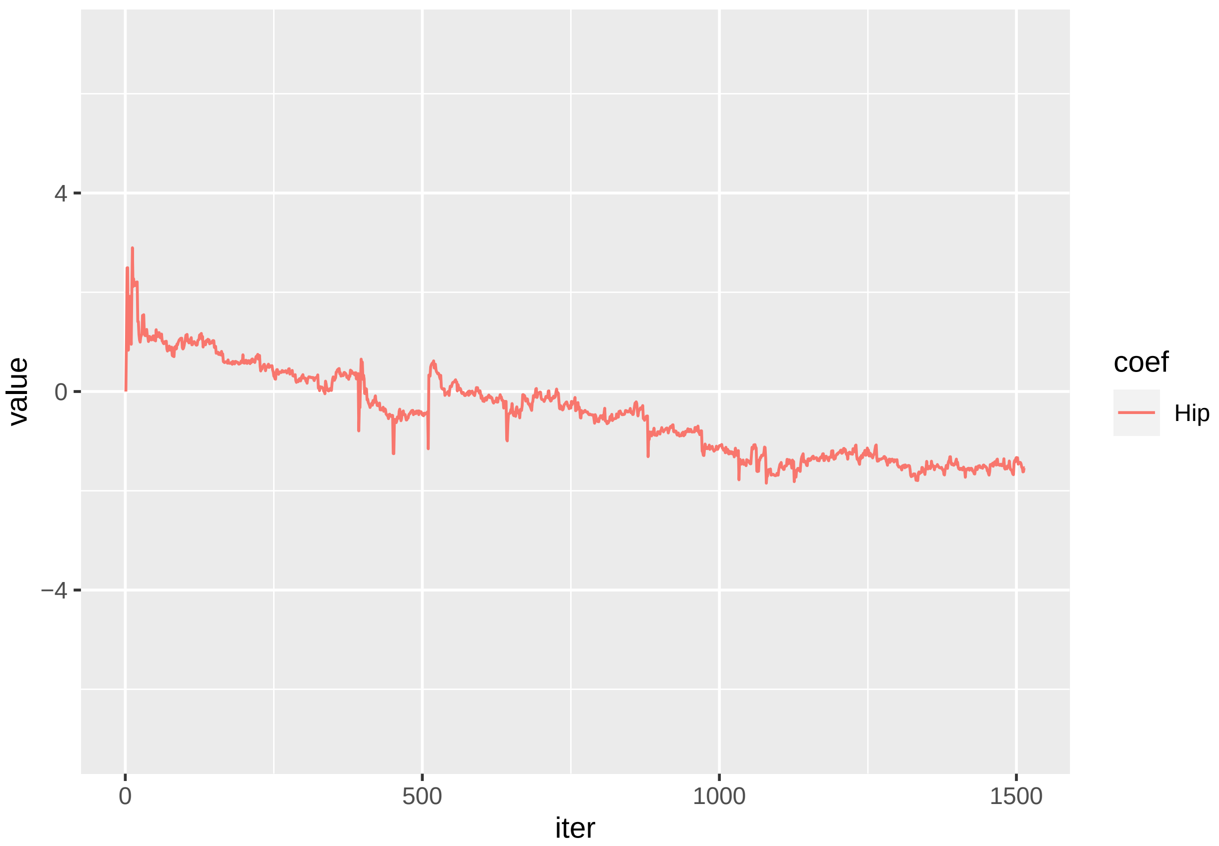
```
betas <- beta %>% select(Neck=V4) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



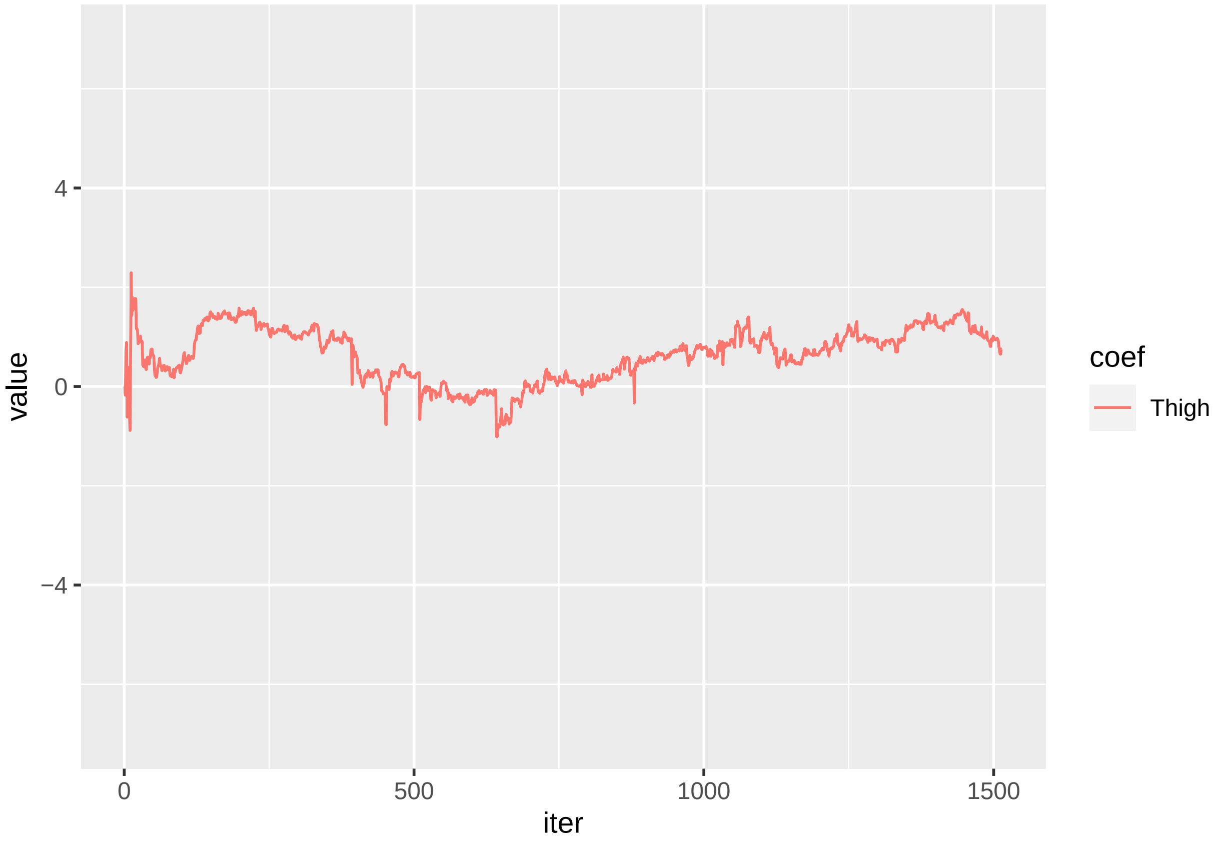
```
betas <- beta %>% select(Abdomen=V5) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-5, 25))
```



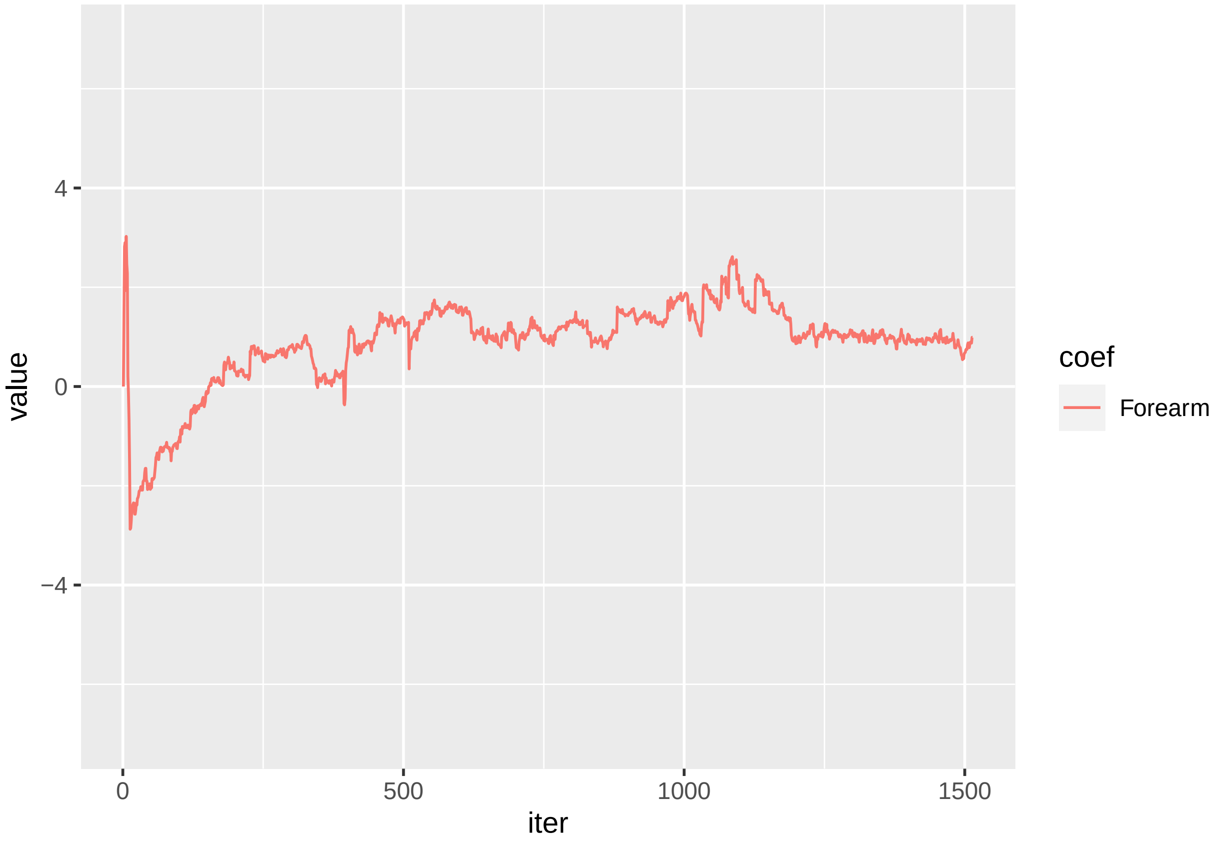
```
betas <- beta %>% select(Hip=V6) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



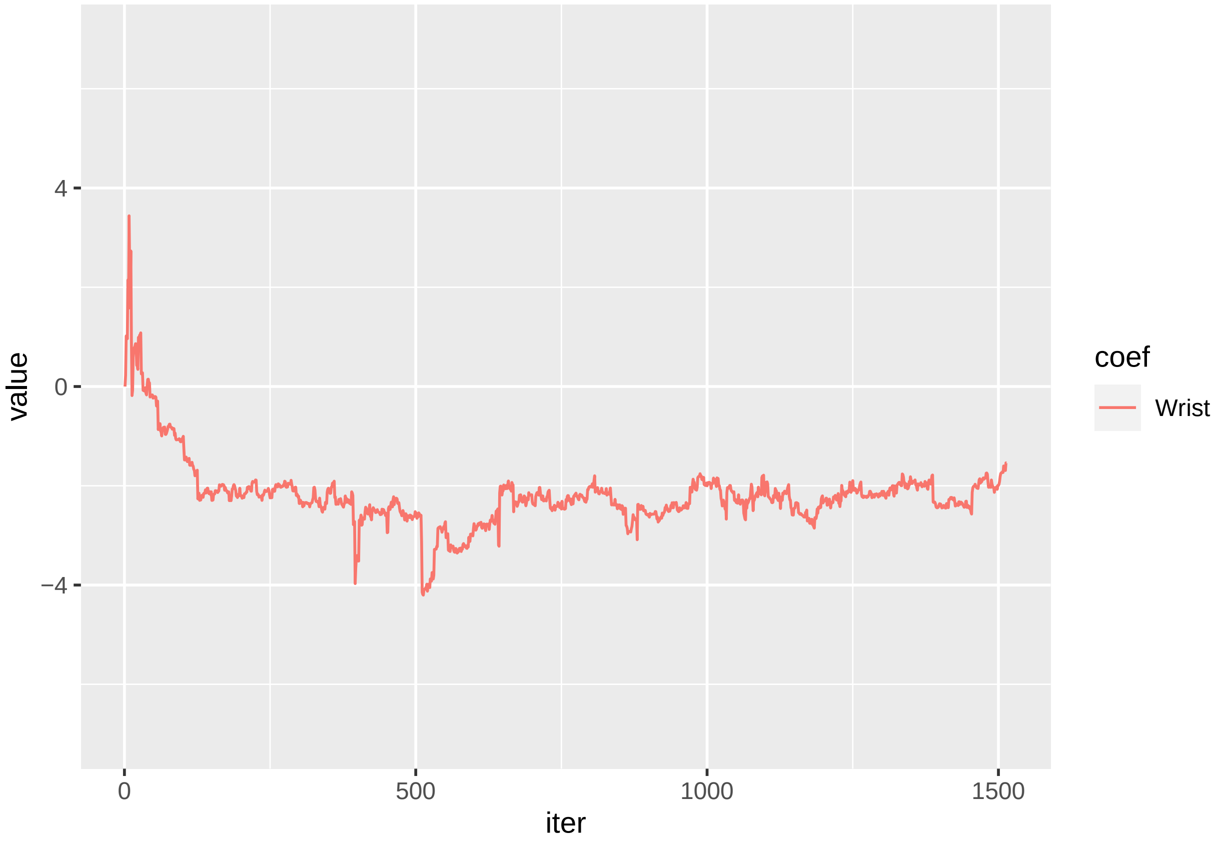
```
betas <- beta %>% select(Thigh=V7) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



```
betas <- beta %>% select(Forearm=V8) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



```
betas <- beta %>% select(Wrist=V9) %>%mutate(iter = 1:nrow(beta))
betas <- melt(betas, id.vars = "iter", variable.name = "coef")
ggplot(betas, aes(iter, value)) + geom_line(aes(colour = coef)) + ylim(c(-7, 7))
```



Although the trace plots of SGD are not smooth, the coefficient eventually fluctuates between the overall sample estimates.