Package demo

William Shih, Ricardo Simpao, Nilay Varshney, Luke Yee

3/20/2020

Here is an example of how our package functions run. For our data set, we are using a "SGEMM GPU kernel performance Data Set," which measures the running times of a matrix-matrix product, given different parameter combinations.

```
library(devtools)
library(tidyverse)
library(STA141CFinal)

dat = read_csv("sgemm_product.csv")
```

```
#We specifiy a specific column set
y = dat$`Run1 (ms)`
x = dat[,1:(ncol(dat)-4)]

#linear model objects
fit = linear_reg_bs(x = x, y = y, s = 10, r = 1000)
#fit$bootstrap_coefficient_estimates

fit2 = linear_reg_bs_par(x = x, y = y, s = 10, r = 1000)
#fit2$bootstrap_coefficient_estimates
```

Linear Regression with blb

95 % Confidence Interval for Variable Coefficients

```
coef_CI(fit, alpha = 0.05)
```

```
##           Lower_Bounds     Estimates Upper_Bounds
## Intercept   -66.910978 218.738529060   513.421889
## MWG          -1.260043   0.006954464     1.288869
## NWG          -1.261053  -0.004824181     1.275299
## KWG          -6.341448   0.005717980     6.249883
## MDIMC        -6.667762  -0.093852118     6.856528
## NDIMC        -6.427916  -0.029233029     6.974446
## MDIMA        -5.362095   0.035059915     5.744351
## NDIMB        -5.403451   0.004079777     5.719580
## KWI         -16.113086  -0.093264542    15.951679
```

```
## VWM         -27.024223  -0.078427197    29.610773
## VWN         -26.135769   0.280068953    30.190972
## STRM        -94.809131   0.384049951    97.708665
## STRN        -95.747251  -0.164080409    96.485461
## SA          -95.888917  -1.007278073    95.918019
## SB          -96.725990   0.558883883    98.466003
```

```
coef_CI_par(fit,alpha = 0.05)
```

```
##           Lower_Bounds     Estimates Upper_Bounds
## Intercept   -66.910978 218.738529060   513.421889
## MWG          -1.260043   0.006954464     1.288869
## NWG          -1.261053  -0.004824181     1.275299
## KWG          -6.341448   0.005717980     6.249883
## MDIMC        -6.667762  -0.093852118     6.856528
## NDIMC        -6.427916  -0.029233029     6.974446
## MDIMA        -5.362095   0.035059915     5.744351
## NDIMB        -5.403451   0.004079777     5.719580
## KWI         -16.113086  -0.093264542    15.951679
## VWM         -27.024223  -0.078427197    29.610773
## VWN         -26.135769   0.280068953    30.190972
## STRM        -94.809131   0.384049951    97.708665
## STRN        -95.747251  -0.164080409    96.485461
## SA          -95.888917  -1.007278073    95.918019
## SB          -96.725990   0.558883883    98.466003
```

```
(b1 = bench::mark(
  coef_CI(fit, alpha = 0.05),
  coef_CI_par(fit,alpha = 0.05))
)
```

```
## # A tibble: 2 x 6
##   expression                          min   median `itr/sec` mem_alloc `gc/sec`
##   <bch:expr>                     <bch:tm> <bch:tm>     <dbl> <bch:byt>    <dbl>
## 1 coef_CI(fit, alpha = 0.05)       31.9ms   35.2ms      28.1    7.49MB     7.66
## 2 coef_CI_par(fit, alpha = 0.05)   53.6ms   54.9ms      17.5     7.8MB     8.76
```

Notice that `coef_CI_par` offers better memory allocation than `coef_CI`.

```
plan(multiprocess, workers = 4)
PI(fit, dat[1:3, 1:14], alpha = 0.05)
PI_par(fit, dat[1:3, 1:14], alpha = 0.05)

(b2 = bench::mark(
  PI(fit, x, alpha = 0.05),
  PI_par(fit, x, alpha = 0.05))
)
```

95 % Confindence Interval for Variance

```
s2_CI(fit, alpha = 0.05)
```

```
## Lower_Bound    Estimate Upper_Bound
##    621.7156   1282.3401   2133.3777
```

```
s2_CI_par(fit, alpha = 0.05)
```

```
## Lower_Bound    Estimate Upper_Bound
##    621.7156   1282.3401   2133.3777
```

```
(b3 = bench::mark(
  s2_CI(fit, alpha = 0.05),
  s2_CI_par(fit, alpha = 0.05))
)
```

```
## # A tibble: 2 x 6
##   expression                      min   median `itr/sec` mem_alloc `gc/sec`
##   <bch:expr>                  <bch:tm> <bch:tm>     <dbl> <bch:byt>    <dbl>
## 1 s2_CI(fit, alpha = 0.05)       2.16ms   2.34ms     398.      118KB     6.74
## 2 s2_CI_par(fit, alpha = 0.05)  30.62ms  32.55ms      29.6     413KB     7.39
```

Notice that `s2_CI_par` offers better memory allocation than `s2_CI`.