Sensitivity Analysis - Latin Hypercube Sampling

Paloma Cartwright, Juliet Cohen, Julia Parish

2022-04-26

Sensitivity Analysis - Latin Hypercube Sampling

This environmental model was completed as an assignment for the course, Environmental Data Science 230 | Environmental Science & Management: Modeling Environmental Systems. The goal of this assignment was to code a function to compute atmosphere condcuctance and to conduct a formal sensitivity analysis using the Latin Hypercube Sampling random sampling method. This assignment focuses on developing skills to create a atmospheric conductance model function, utilize the Latin Hypercube Sampling (LHS) to generate near random sample of parameter values, and then plot the atmospheric conductance values.

1. Code a function to compute atmospheric conductance

```
source(here("R/atmcon.R"))
```

2. Run `atmcon` model and provide a single estimate of atmospheric conductance for this forest.

```
ac_forest <- atmcon(vm = 250, h = 1000)
ac_forest
```

```
## $ac
## [1] 15.44228
```

```
ac_forest_rounded <- round(ac_forest[[1]], 2)

print(paste0("The atmospheric conductance for this vegetation is ",
             ac_forest_rounded, " centimeters per second."))
```

```
## [1] "The atmospheric conductance for this vegetation is 15.44 centimeters per second."
```

3. Conduct a sensitivity analysis

Consider the sensitivity of estimates to uncertainty in the following parameters and inputs:

- *h*
- *kd*
- *k0*
- *v*

3.A Use LHS to generate parameter values for the 4 parameters

```
factors = c("vm", "h", "kd", "k")
```

```r
nsets = 100

q = c("qnorm", "qunif", "qnorm", "qnorm")

q.arg = list(list(mean = 250, sd = 30),
             list(min = 950, max = 1050),
             list(mean = 0.7, sd = 0.07),
             list(mean = 0.1, sd = 0.01))
q.arg
```

```
## [[1]]
## [[1]]$mean
## [1] 250
##
## [[1]]$sd
## [1] 30
##
##
## [[2]]
## [[2]]$min
## [1] 950
##
## [[2]]$max
## [1] 1050
##
##
## [[3]]
## [[3]]$mean
## [1] 0.7
##
## [[3]]$sd
## [1] 0.07
##
##
## [[4]]
## [[4]]$mean
## [1] 0.1
##
## [[4]]$sd
## [1] 0.01
```

```r
# generate samples
sens_ac = LHS(NULL, factors, nsets, q, q.arg)
#sens_ac
#summary(sens_ac)
#sens_ac$data
# NULL indicates there is no model
sens_pars <- get.data(sens_ac)
sens_pars
```

```
##           vm      h        kd          k
## 1   191.2011 1035.5 0.7260299 0.08560469
## 2   248.8718 1018.5 0.6398268 0.09785298
## 3   232.9585  976.5 0.7576726 0.11058122
## 4   242.7872 1020.5 0.6345787 0.08304602
```

```
## 5   264.4518 1011.5 0.7805245 0.10896473
## 6   246.6088  981.5 0.6538814 0.10453762
## 7   172.7251 1049.5 0.5196919 0.11959964
## 8   300.8619 1027.5 0.6938509 0.11695398
## 9   247.3647  958.5 0.6082595 0.09518273
## 10  295.4231  960.5 0.5628025 0.10934589
## 11  262.7844  961.5 0.7461186 0.10690309
## 12  253.3912  952.5 0.6701696 0.09402240
## 13  283.0919  975.5 0.8268337 0.10138304
## 14  244.3264 1028.5 0.7840251 0.09309691
## 15  261.9657 1031.5 0.7552434 0.09371994
## 16  297.9458 1014.5 0.6372469 0.12170090
## 17  260.3538 1041.5 0.8519063 0.09912155
## 18  273.6757 1048.5 0.7377185 0.10538836
## 19  245.8509  989.5 0.6194755 0.10371856
## 20  251.8812  991.5 0.6849709 0.09836342
## 21  204.5769  965.5 0.6776952 0.09277521
## 22  270.7093 1039.5 0.7279199 0.09886961
## 23  267.9328 1013.5 0.7439604 0.10481727
## 24  232.0672  957.5 0.6622815 0.11598193
## 25  241.2288  995.5 0.8371975 0.10597760
## 26  254.1491  962.5 0.6122504 0.08849651
## 27  256.4410  985.5 0.7528791 0.11253565
## 28  224.2115  996.5 0.7681880 0.11200359
## 29  239.6462 1019.5 0.7223048 0.09065411
## 30  287.6070 1036.5 0.7710655 0.10345126
## 31  286.0108 1015.5 0.6831702 0.10266311
## 32  219.5433 1022.5 0.6956105 0.09176106
## 33  261.1557 1008.5 0.6289345 0.09244585
## 34  208.8339  966.5 0.7061491 0.08188089
## 35  202.0542  993.5 0.6991227 0.10859617
## 36  225.2832  969.5 0.7186417 0.10823894
## 37  263.6129 1046.5 0.7168298 0.11811911
## 38  304.3573  977.5 0.6885439 0.09759574
## 39  210.6826 1012.5 0.7505735 0.11310579
## 40  237.2156 1002.5 0.6318120 0.12575829
## 41  291.1661  955.5 0.6920873 0.09654874
## 42  249.6240  997.5 0.7132383 0.11015222
## 43  230.2349 1023.5 0.7096813 0.09628144
## 44  195.6427  988.5 0.6039457 0.09962392
## 45  250.3760  964.5 0.6602364 0.10426148
## 46  275.7885  990.5 0.7601732 0.09103527
## 47  276.8942 1030.5 0.7917405 0.10062707
## 48  308.7989  987.5 0.7772144 0.09987467
## 49  220.7766 1038.5 0.7150291 0.07424171
## 50  227.3375 1017.5 0.6813583 0.10037608
## 51  293.1859  994.5 0.6560396 0.11439531
## 52  271.6744  953.5 0.6581568 0.09431949
## 53  236.3871 1044.5 0.6739701 0.10292375
## 54  215.4895 1021.5 0.6447566 0.09546238
## 55  269.7651 1000.5 0.7026326 0.10189118
## 56  184.8973  992.5 0.7298304 0.09810882
## 57  221.9623 1003.5 0.7079127 0.09025886
## 58  251.1282 1010.5 0.6682366 0.09341162
```

```
## 59  231.1598  983.5 0.6795338 0.10722479
## 60  248.1188  950.5 0.7740685 0.08984778
## 61  228.3256 1045.5 0.7008773 0.09707625
## 62  268.8402  963.5 0.7317634 0.10214702
## 63  278.0377 1025.5 0.7357051 0.09733689
## 64  218.2564  956.5 0.7483216 0.09210808
## 65  206.8141  954.5 0.8186778 0.10789192
## 66  216.9081 1024.5 0.7241588 0.10510073
## 67  245.0902 1005.5 0.5813222 0.09140383
## 68  315.1027  980.5 0.6494265 0.08627796
## 69  252.6353 1033.5 0.5731663 0.08941878
## 70  327.2749 1009.5 0.7114561 0.08799641
## 71  266.1651  974.5 0.8007672 0.10974114
## 72  272.6625  999.5 0.5940129 0.10163658
## 73  235.5482  959.5 0.8059871 0.09937293
## 74  267.0415 1032.5 0.6471209 0.10087845
## 75  284.5105 1026.5 0.6159749 0.09573852
## 76  274.7168  972.5 0.6516784 0.09489927
## 77  199.1381  984.5 0.7337209 0.08689421
## 78  233.8349 1006.5 0.6867617 0.10755415
## 79  258.7712  970.5 0.6903187 0.09861696
## 80  242.0107  973.5 0.7204662 0.11103063
## 81  238.8443 1042.5 0.6423274 0.08401807
## 82  259.5592 1001.5 0.6720801 0.11514102
## 83  243.5590  951.5 0.5881265 0.10658838
## 84  257.9893 1016.5 0.5480937 0.10113039
## 85  265.3022 1004.5 0.6973674 0.11372204
## 86  279.2234  979.5 0.6227856 0.07829910
## 87  223.1058 1040.5 0.8118735 0.09601145
## 88  257.2128 1043.5 0.7418432 0.11150349
## 89  226.3243  968.5 0.6259315 0.09681361
## 90  280.4567  978.5 0.7627531 0.08485898
## 91  255.6736 1037.5 0.6662791 0.10318639
## 92  212.3930  986.5 0.7043895 0.10628006
## 93  229.2907  998.5 0.6642949 0.10398855
## 94  234.6978  982.5 0.7654213 0.09461164
## 95  213.9892  971.5 0.7960543 0.08896937
## 96  238.0343 1047.5 0.7877496 0.10012533
## 97  240.4408 1029.5 0.6758412 0.10240426
## 98  254.9098  967.5 0.7397636 0.10568051
## 99  289.3174 1034.5 0.5992328 0.08040036
## 100 281.7436 1007.5 0.8803081 0.08746435
```

3.B Run the atmospheric conductance model, `atmcon`, for LHS derived parameters and return aerodynamic conductances

```
source(here("R/atmcon.R"))
ac_lhs <- pmap(sens_pars, atmcon)
head(ac_lhs[[2]])
```

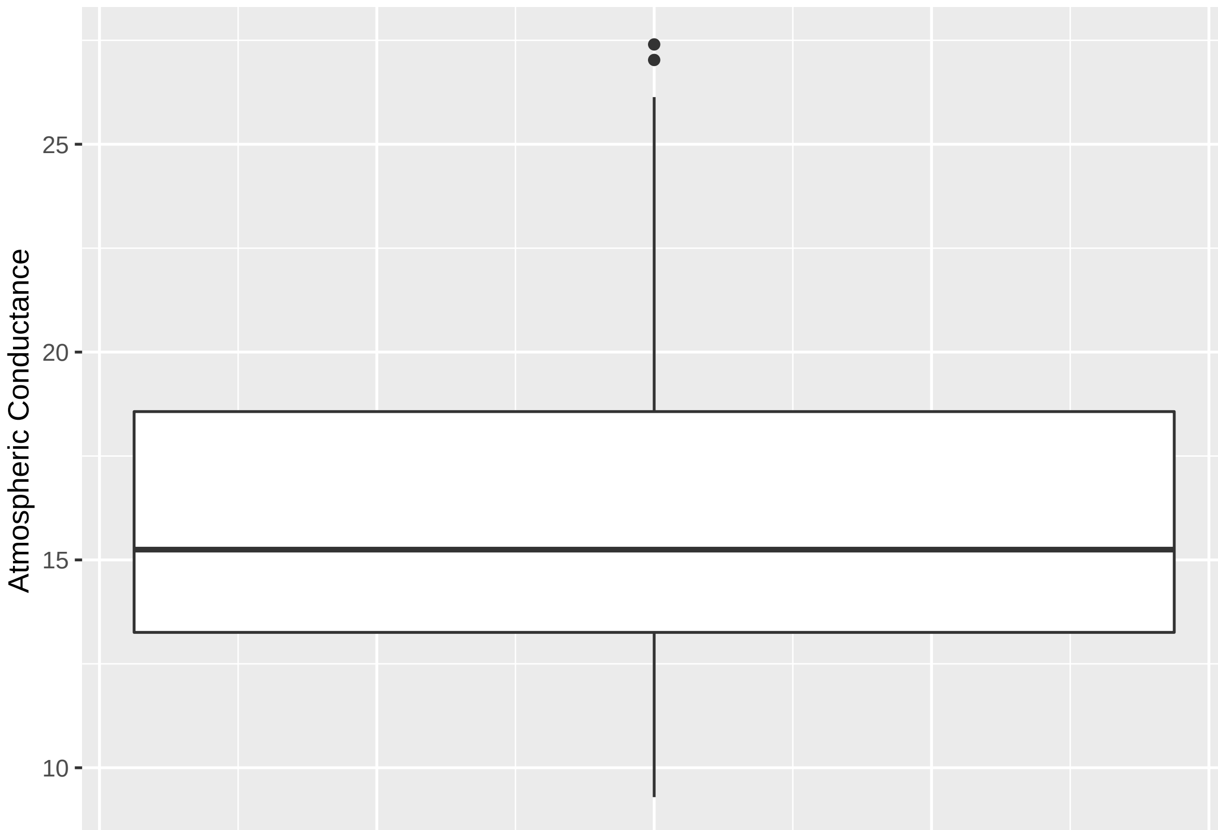
```
## $ac
## [1] 13.1783
```

```r
atmospheric_conductances <- ac_lhs %>%
  map_dfr(`[`, "ac")

atmospheric_conductances
```

```
## # A tibble: 100 x 1
##       ac
##    <dbl>
##  1 10.6
##  2 13.2
##  3 19.1
##  4 10.6
##  5 23.5
##  6 14.3
##  7  9.29
##  8 22.8
##  9 11.7
## 10 15.0
## # ... with 90 more rows
```

3.C. Plot conductance estimates in a way that accounts for parameter uncertainty

```r
#data <- atmospheric_conductances %>%
#  gather(value = "value", key = )

ggplot(data = atmospheric_conductances, aes(y = ac)) +
  geom_boxplot() +
  labs(y = "Atmospheric Conductance") +
  #theme(x = element_blank()) +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```



3.D. Plot conductance estimates against each of your parameters

```
#pse::plotscatter(sens_ac)
```

3.E. Estimate the Partial Rank Correlation Coefficients (PRCC)

3.F. Discussion

What do the results tell about how aerodynamic conductance?

What does it suggest about what you should focus on if you want to reduce uncertainty in aerodymaic conductance estimates?

Does this tell you anything about the sensitivity of plant water use to climate change?