

# Atmospheric Conductance Sensitivity Analysis

Elmera Azadpour, Danielle Sciafani, Desik Somasundaram

2022-04-21

## Setup

```
source(here("functions", "atmospheric_conductance.R"))
```

## Analysis

```
#test function  
atmos_conduct(1000, 250)
```

```
## $atmospheric_conductance  
## [1] 15.44228
```

```
# 4 parameters for sensitivity analysis  
factors = c("h", "k_d", "k_0", "v")
```

```
# Decide How many parameter sets to run  
nsets=100
```

```
# distributions for parameters along with known ranges for sampling
```

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

```
q.arg = list(list(min=950, max=1050), list(mean=0.7, sd=0.007), list(mean=0.1, sd=0.001), list(mean=250, sd=10))
```

```
# generate samples from LHS
```

```
sens_atmos_cond = LHS(NULL, factors, nsets, q, q.arg)
```

```
sens_pars = get.data(sens_atmos_cond)
```

```
head(sens_pars)
```

```
##           h           k_d           k_0           v  
## 1  994.5 0.7013238 0.09996239 278.0377  
## 2 1008.5 0.7080524 0.09988696 265.3022  
## 3 1006.5 0.7151906 0.09804004 287.6070  
## 4  950.5 0.6970170 0.10051007 273.6757  
## 5  953.5 0.6915975 0.10093459 245.8509  
## 6  973.5 0.6999123 0.10003761 249.6240
```

```
# run model for all of the parameters generated by LHS
```

```
# use map function that uses the actual names of input parameters
```

```
atmospheric_conductance = sens_pars %>% pmap(atmos_conduct)
```

```
head(atmospheric_conductance)
```

```
## [[1]]  
## [[1]]$atmospheric_conductance  
## [1] 17.17542  
##
```

```
##
## [[2]]
## [[2]]$atmospheric_conductance
## [1] 16.77161
##
##
## [[3]]
## [[3]]$atmospheric_conductance
## [1] 18.07217
##
##
## [[4]]
## [[4]]$atmospheric_conductance
## [1] 16.46485
##
##
## [[5]]
## [[5]]$atmospheric_conductance
## [1] 14.70041
##
##
## [[6]]
## [[6]]$atmospheric_conductance
## [1] 15.21743
```

## Results and Visualizations

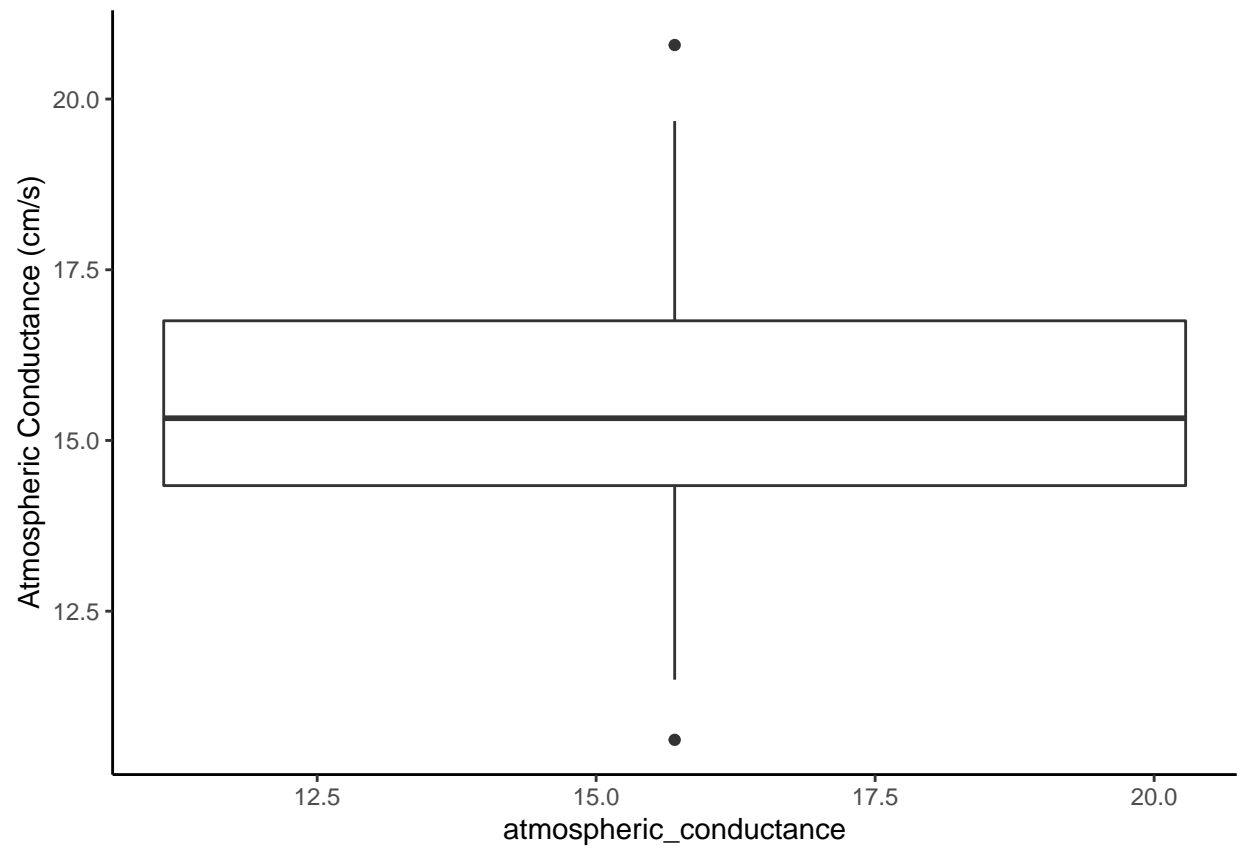
```
# turn results in to a dataframe for easy display/analysis
atmospheric_conductancedf = atmospheric_conductance %>% map_dfr(`[,c("atmospheric_conductance")])

# tell links output to original LHS object once results are in matrix
sens_atmos_cond = pse::tell(sens_atmos_cond, t(as.matrix(atmospheric_conductancedf)),
                             res.names=c("atmospheric_conductance"))
```

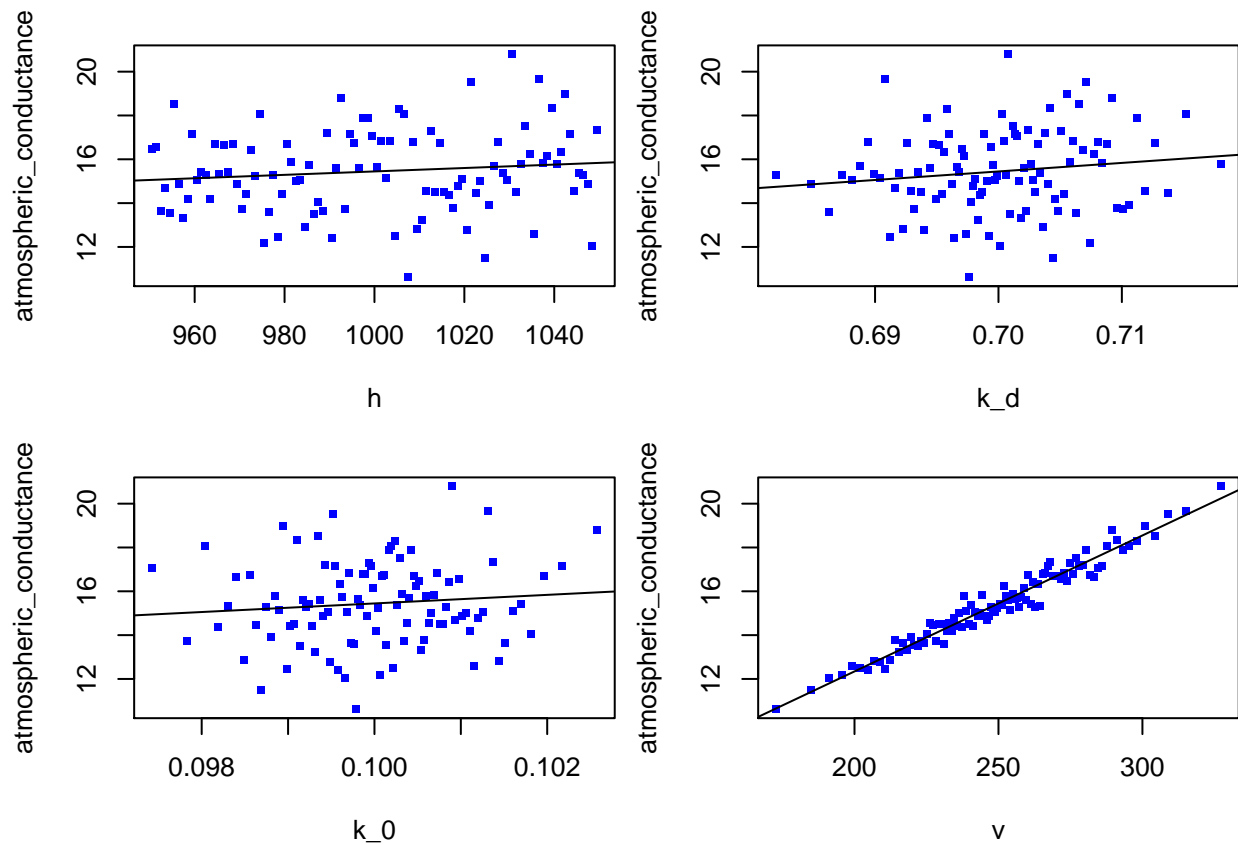
```
# add uncertainty bounds on our estimates (c)
tmp = atmospheric_conductancedf %>%
  mutate(type = "atmospheric_conductance") %>%
  rename(value = atmospheric_conductance)

ggplot(tmp, aes(atmospheric_conductance, value))+
  geom_boxplot()+
  labs(y="Atmospheric Conductance (cm/s)") +
  theme_classic()
```

```
## Warning in min(diff(sort(x))): no non-missing arguments to min; returning Inf
```



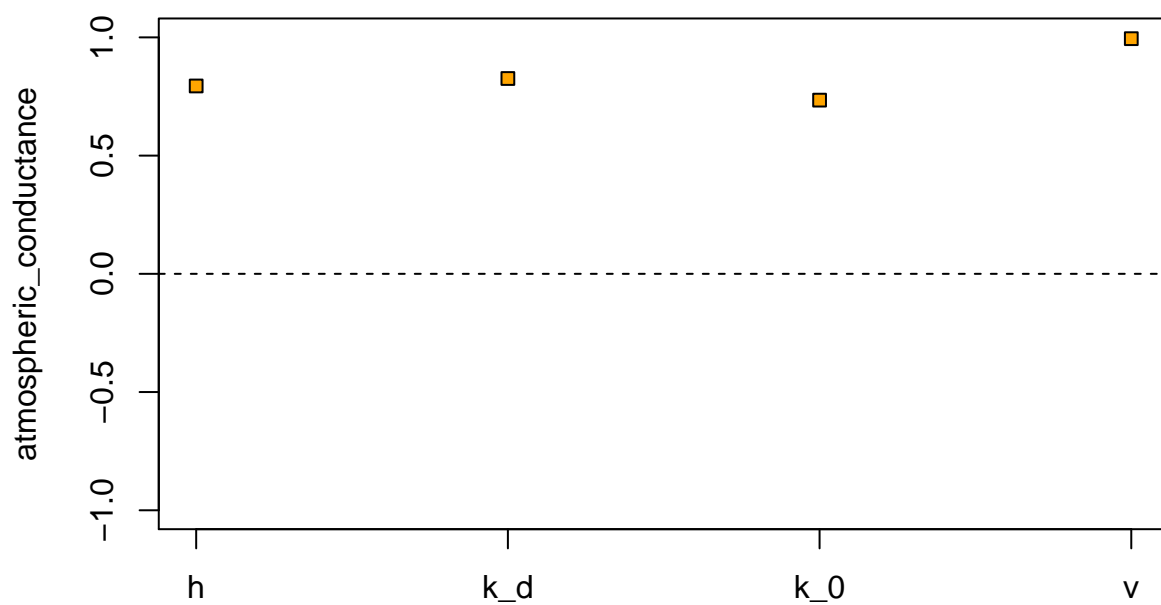
```
#use built in LHS functions to analyze parameter sensitivity (d)  
pse::plotscatter(sens_atmos_cond, col="blue", cex=5)
```



```
# Estimate the Partial Rank Correlation Coefficients (e)
# PSE: object has partial rank correlation coefficients

# prcc's automatically generated and easy to plot
pse::plotprcc(sens_atmos_cond)
```

## PRCC



```

# more negative/more positive the value, the more sensitive your object is to

# PRCC for all 3 output metrics
sens_atmos_cond$res.names

## [1] "atmospheric_conductance"

sens_atmos_cond$prcc

## [[1]]
##
## Call:
## pcc.default(X = L, y = r, rank = T, nboot = nboot)
##
## Partial Rank Correlation Coefficients (PRCC):
##      original
## h    0.7944352
## k_d  0.8259997
## k_0  0.7345424
## v    0.9945765

# correlation coefficient
# compare PRCC with first correlation coefficient
# recall
head(atmospheric_conductancedf)

## # A tibble: 6 x 1
##   atmospheric_conductance
##               <dbl>
## 1                   17.2
## 2                   16.8
## 3                   18.1
## 4                   16.5
## 5                   14.7
## 6                   15.2

sens_atmos_cond$prcc[1]

## [[1]]
##
## Call:
## pcc.default(X = L, y = r, rank = T, nboot = nboot)
##
## Partial Rank Correlation Coefficients (PRCC):
##      original
## h    0.7944352
## k_d  0.8259997
## k_0  0.7345424
## v    0.9945765

# we can still use our sens_pars data frame - rows of parameters will be
# match rows in the output from our use of pmap to run the model for all
# parameters
cor(atmospheric_conductancedf$atmospheric_conductance, sens_pars$h, method="spearman")

## [1] 0.1282928

```

```
cor(atmospheric_conductancedf$atmospheric_conductance, sens_pars$k_d, method="spearman")
## [1] 0.150099
cor(atmospheric_conductancedf$atmospheric_conductance, sens_pars$k_0, method="spearman")
## [1] 0.07793579
cor(atmospheric_conductancedf$atmospheric_conductance, sens_pars$v, method="spearman")
## [1] 0.9682208
```

## Discussion

*Discuss what your results tell you about how aerodynamic conductance varies with the different parameters? What does it suggest about what you should focus on if you want to reduce uncertainty in aerodynamic conductance estimates? Does this tell you anything about the sensitivity of plant water use to climate change? (f)*

Our results indicate that atmospheric conductance is most sensitive to  $v$  which is wind speed, followed by  $k_d$ ,  $k_0$ , and  $h$  (vegetation height). This means that getting more precise wind speed would help significantly reduce the uncertainty in aerodynamic conductance estimates. Climate change is expected to have a significant impact on weather patterns around the world with an expected decrease in mean wind speed in most regions. However, it is important to note that the changes will depend locally so it's hard to generalize without setting a more specific region of interest. Another factor to consider is the rise in stomatal conductance due to global rise in temperature which also affects plant water use.