Assignment 3

Anna Calle and Jamie Miller

4/22/2020

Adjust your almond model to output ONLY the mean almond yield anomaly IF the users sets parameter (e.g mean_only = TRUE))

Perform a sensitivity analysis of how mean anomaly varies ALL of the parameters used in the yield model

Assume parameters are normally distributed with standard deviation of 20% mean value

Rank the parameters in term of their sensitivity

Graph uncertainty in mean yield anomaly across all parameter uncertainty (boxplot and cumulative distribution of the output).

Repeat using the LHS and Sobol methods

Repeat using twice as many parameter sets as you did in your first sensitivity analysis - and look at how this changes the sensitivity results

Submit R markdown and short write up describing what you learned from the sensitivity analysis. Please submit your markdown as an .html or PDF.

```
# read in the input data
clim <- read.table("clim.txt",  sep=" ", header=T)

# source function
source("almond_yield.R")

almond_yield(mean_only = TRUE)
```

```
## [1] 181.4453
```

LHS

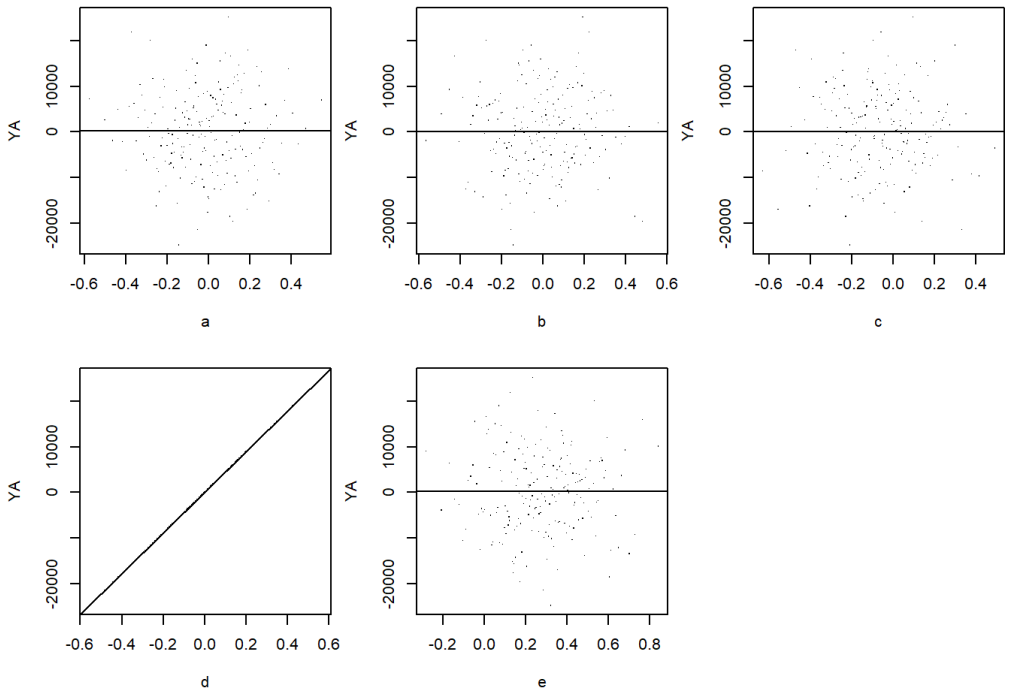
```
# names of our parameters: a = "Tmincoeff1", b = "Tmincoeff2", c = "Precipcoeff1", d = "Precipcoeff2", e = "Intercept"
factors = c("a", "b", "c", "d", "e")

# type of distributions they arise from
q = c("qnorm", "qnorm", "qnorm", "qnorm", "qnorm")

# parameters mean
a = -0.015
b = -0.0046
c = -0.07
d = 0.0043
e = 0.28

# parameters for those distributions
q.arg = list(list(mean=a, sd=0.2),
             list(mean=b, sd=0.2),
             list(mean=c, sd=0.2),
             list(mean=d, sd=0.2),
             list(mean=e, sd=0.2))

nsets=200
sens_YA = LHS(NULL,factors,nsets,q,q.arg)

# save parameter values
sens.pars = get.data(sens_YA)
head(sens.pars)
```

```
##             a           b           c            d         e
## 1 -0.22883093 -0.15072770 -0.18508615 -0.068729850 0.2202157
## 2 -0.05922374  0.04477628 -0.36148435 -0.223357716 0.2887760
## 3 -0.19242931  0.06575687  0.07287349 -0.009497392 0.2787467
## 4 -0.02377601 -0.20555716  0.01112994  0.189286892 0.1878561
## 5  0.54640675 -0.23225772 -0.10400258  0.157055449 0.5714843
## 6 -0.08535687 -0.04370039 -0.11422374 -0.135356673 0.4574293
```

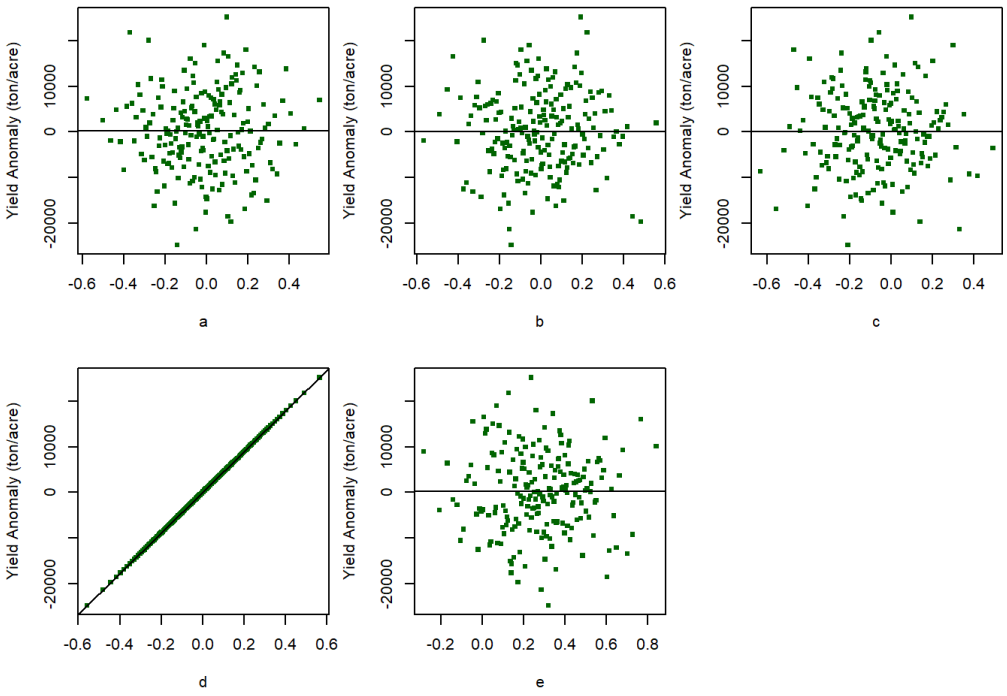
```
# run model
sens_results= mapply(FUN=almond_yield, a=sens.pars$a, b=sens.pars$b, c=sens.pars$c, d=sens.pars$d, e=sens.pars$e, MoreArgs=l
ist(clim_data = clim, mean_only=TRUE))
head(sens_results)
```

```
## [1] -3094.8689 -9966.4315  -407.3693  8390.1440  6946.3775 -6033.1266
```

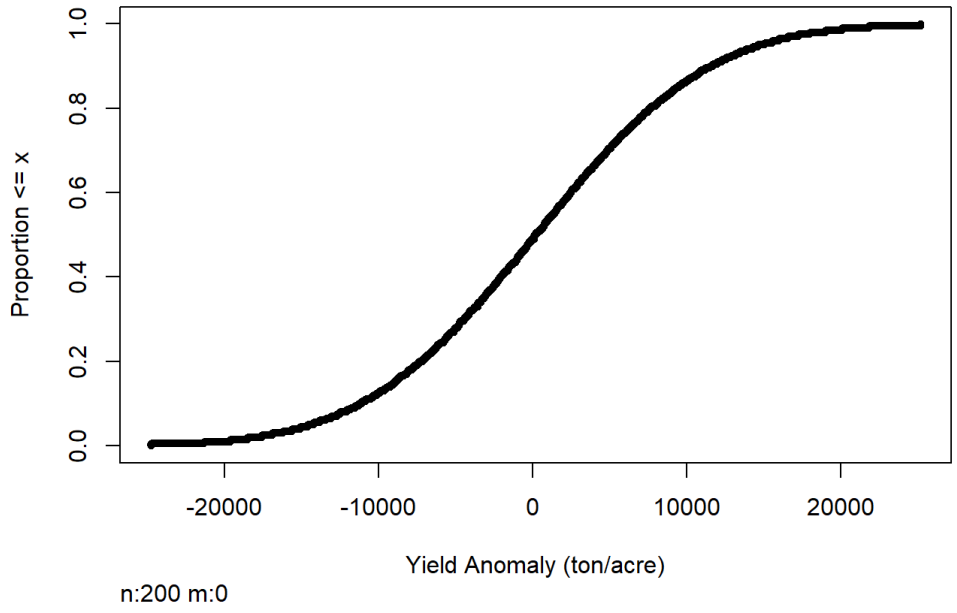
```
# use unlist to get a matrix
sens_res = matrix((unlist(sens_results)), ncol=1, byrow=TRUE)

sens_YA = pse::tell(sens_YA, t(sens_res), res.names="YA")
plotscatter(sens_YA)
```



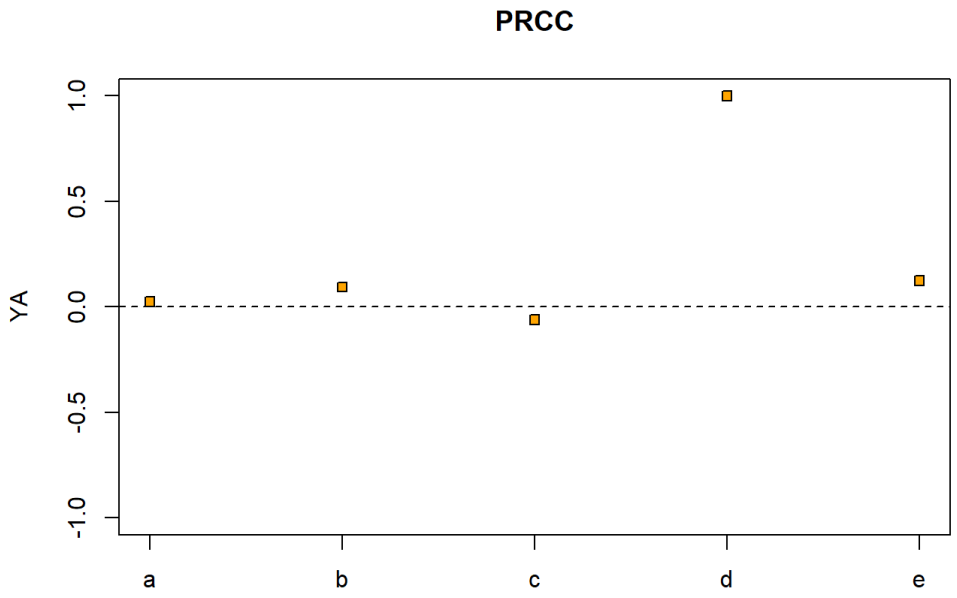

```
# note it can take standard R plotting parameters
plotscatter(sens_YA, col="darkgreen", cex=5, ylab="Yield Anomaly (ton/acre)")
```




```
# whats is the range of results
plotecdf(sens_YA, col="red", lwd=5, xlab="Yield Anomaly (ton/acre)")
```



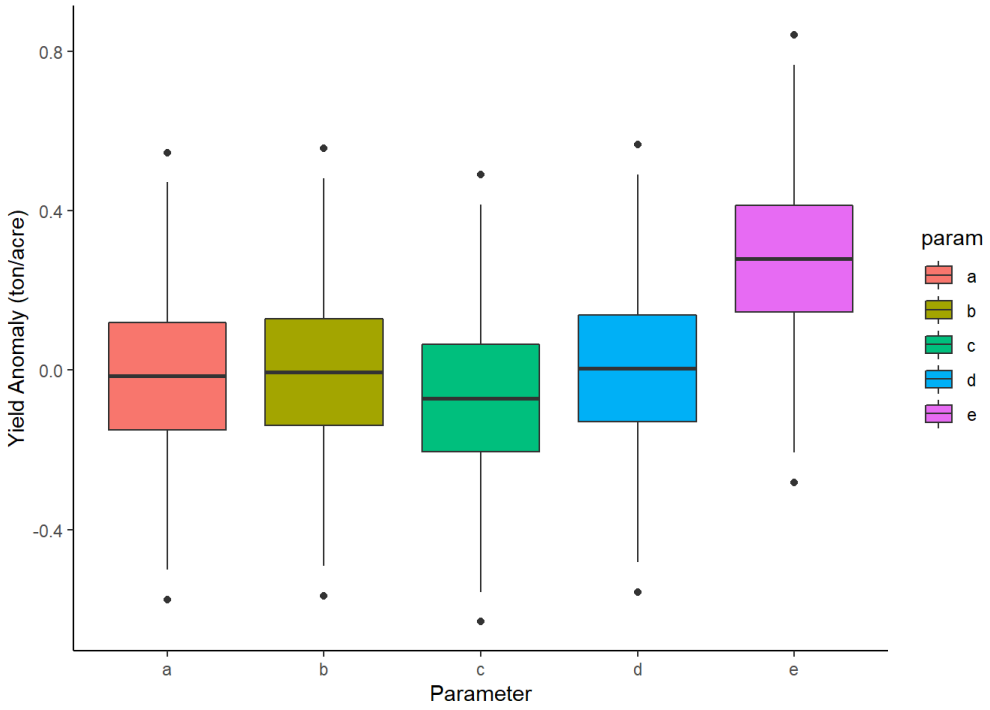

```
# we can also plot partial correlation coefficients
plotprcc(sens_YA)
```




```
# rank parameters
sens_YA_prcc <- sens_YA$prcc[[1]]$PRCC %>%
  mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

colnames(sens_YA_prcc) <-  c("PRCC", "Parameter")
sens_YA_prcc
```

```
##          PRCC Parameter
## 1  1.00000000         d
## 2  0.12449837         e
## 3  0.09377268         b
## 4 -0.06114488         c
## 5  0.02510748         a
```

```
# boxplot
sens_YA_df <- sens_YA$data %>%
  gather(param, YA) %>%
  mutate(method = "LHS")

lhs_boxplot <- ggplot(sens_YA_df, aes(param, YA, fill=param)) +
  geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  theme_classic()

lhs_boxplot
```



Sobol

```
# names of our parameters
# number of parameters

np=200
a = rnorm(mean=a, sd=0.2, n=np)
b = rnorm(mean=b, sd=0.2, n=np)
c = rnorm(mean=c, sd=0.2, n=np)
d = rnorm(mean=d, sd=0.2, n=np)
e = rnorm(mean=e, sd=0.2, n=np)

# generate two examples of random number from parmeter distributions

X1 = cbind.data.frame(a, b, c, d, e)

# repeat sampling
a = rnorm(mean=a, sd=0.2, n=np)
b = rnorm(mean=b, sd=0.2, n=np)
c = rnorm(mean=c, sd=0.2, n=np)
d = rnorm(mean=d, sd=0.2, n=np)
e = rnorm(mean=e, sd=0.2, n=np)

X2 = cbind.data.frame(a, b, c, d, e)

sens_YA_sobel = sobol2007(model = NULL, X1, X2, nboot = 100)


# run model for all parameter sets
res = mapply(FUN=almond_yield,  a=sens_YA_sobel$X$a,
             b=sens_YA_sobel$X$b,
             c=sens_YA_sobel$X$c,
             d=sens_YA_sobel$X$d,
             e=sens_YA_sobel$X$e,
             MoreArgs=list(clim_data = clim, mean_only=TRUE))


sens_YA_sobel = sensitivity::tell(sens_YA_sobel,res, res.names="YA")

# first-order indices (main effect without co-variance)
sens_YA_sobel$S
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a  3.024782e-05 -4.654492e-06 2.893335e-05 -1.765473e-05 9.438004e-05
## b  2.363611e-04 -4.682006e-05 2.865200e-04 -2.744107e-04 7.893564e-04
## c -1.506499e-06 -3.574017e-05 3.013451e-04 -4.717173e-04 8.082293e-04
## d  1.254827e+00  1.927029e-02 1.950143e-01  7.227101e-01 1.530511e+00
## e -3.665323e-06  2.707592e-08 2.482678e-06 -8.478476e-06 8.616172e-07
```

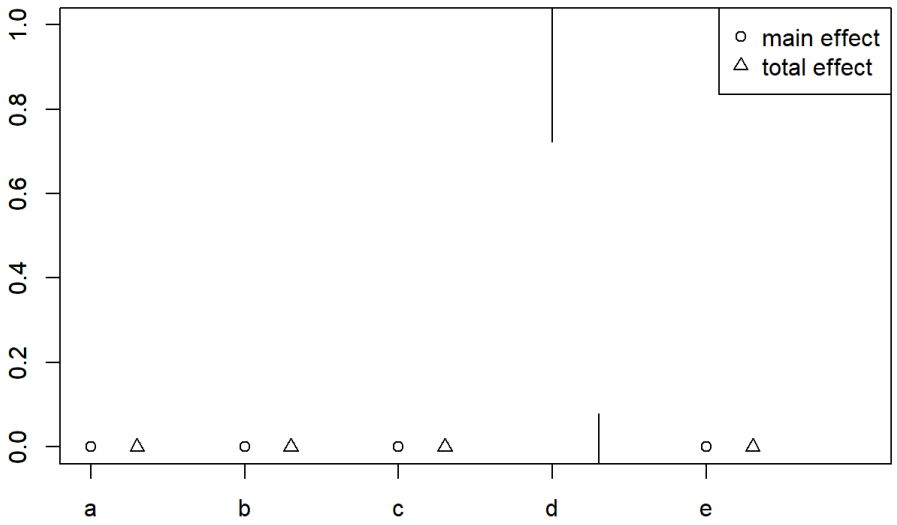
```
# total sensitivity index -note that this partitions the output variance - so values sum to 1
sens_YA_sobel$T
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a -3.892448e-06  4.795675e-06 1.661919e-05 -4.590681e-05 2.363092e-05
## b -2.421573e-04  2.409083e-05 1.774055e-04 -7.175371e-04 8.851531e-05
## c  7.621526e-05  4.357152e-05 2.046075e-04 -4.407685e-04 4.450157e-04
## d -8.871306e-02 -1.576804e-03 7.455461e-02 -2.375267e-01 7.927502e-02
## e  1.648905e-06  1.691964e-07 1.762263e-06 -2.094772e-06 5.065208e-06
```

```
# The difference between the main effect and total effect can tell us something about how the parameter influences results
# so in the main effect we include interactions with other parameters
print(sens_YA_sobel)
```

```
##
## Call:
## sobol2007(model = NULL, X1 = X1, X2 = X2, nboot = 100)
##
## Model runs: 1400
##
## First order indices:
##        original          bias   std. error     min. c.i.    max. c.i.
## a  3.024782e-05 -4.654492e-06 2.893335e-05 -1.765473e-05 9.438004e-05
## b  2.363611e-04 -4.682006e-05 2.865200e-04 -2.744107e-04 7.893564e-04
## c -1.506499e-06 -3.574017e-05 3.013451e-04 -4.717173e-04 8.082293e-04
## d  1.254827e+00  1.927029e-02 1.950143e-01  7.227101e-01 1.530511e+00
## e -3.665323e-06  2.707592e-08 2.482678e-06 -8.478476e-06 8.616172e-07
##
## Total indices:
##        original          bias   std. error     min. c.i.    max. c.i.
## a -3.892448e-06  4.795675e-06 1.661919e-05 -4.590681e-05 2.363092e-05
## b -2.421573e-04  2.409083e-05 1.774055e-04 -7.175371e-04 8.851531e-05
## c  7.621526e-05  4.357152e-05 2.046075e-04 -4.407685e-04 4.450157e-04
## d -8.871306e-02 -1.576804e-03 7.455461e-02 -2.375267e-01 7.927502e-02
## e  1.648905e-06  1.691964e-07 1.762263e-06 -2.094772e-06 5.065208e-06
```

```
plot(sens_YA_sobel)
```




```
# compare with LHS and PRCC
sens_YA$prcc
```

```
## [[1]]
##
## Call:
## pcc.default(X = L, y = r, rank = T, nboot = nboot)
##
## Partial Rank Correlation Coefficients (PRCC):
##      original
## a  0.02510748
## b  0.09377268
## c -0.06114488
## d  1.00000000
## e  0.12449837
```

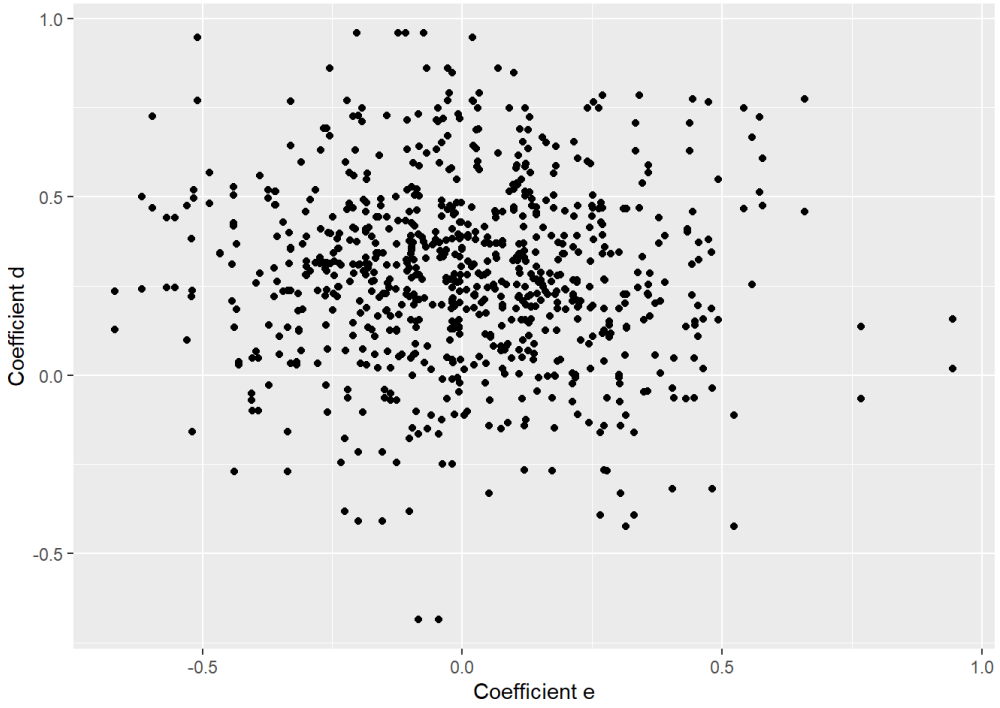
```
sens_YA_sobel$S
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a  3.024782e-05 -4.654492e-06 2.893335e-05 -1.765473e-05 9.438004e-05
## b  2.363611e-04 -4.682006e-05 2.865200e-04 -2.744107e-04 7.893564e-04
## c -1.506499e-06 -3.574017e-05 3.013451e-04 -4.717173e-04 8.082293e-04
## d  1.254827e+00  1.927029e-02 1.950143e-01  7.227101e-01 1.530511e+00
## e -3.665323e-06  2.707592e-08 2.482678e-06 -8.478476e-06 8.616172e-07
```

```
sens_YA_sobel$T
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a -3.892448e-06  4.795675e-06 1.661919e-05 -4.590681e-05 2.363092e-05
## b -2.421573e-04  2.409083e-05 1.774055e-04 -7.175371e-04 8.851531e-05
## c  7.621526e-05  4.357152e-05 2.046075e-04 -4.407685e-04 4.450157e-04
## d -8.871306e-02 -1.576804e-03 7.455461e-02 -2.375267e-01 7.927502e-02
## e  1.648905e-06  1.691964e-07 1.762263e-06 -2.094772e-06 5.065208e-06
```

```
# make a data frame for plotting
both = cbind.data.frame(sens_YA_sobel$X, gs=sens_YA_sobel$y)

# look at response of conductance to the two most important variables
 ggplot(both, aes(d, e))+geom_point()+labs(y="Coefficient d", x="Coefficient e")
```



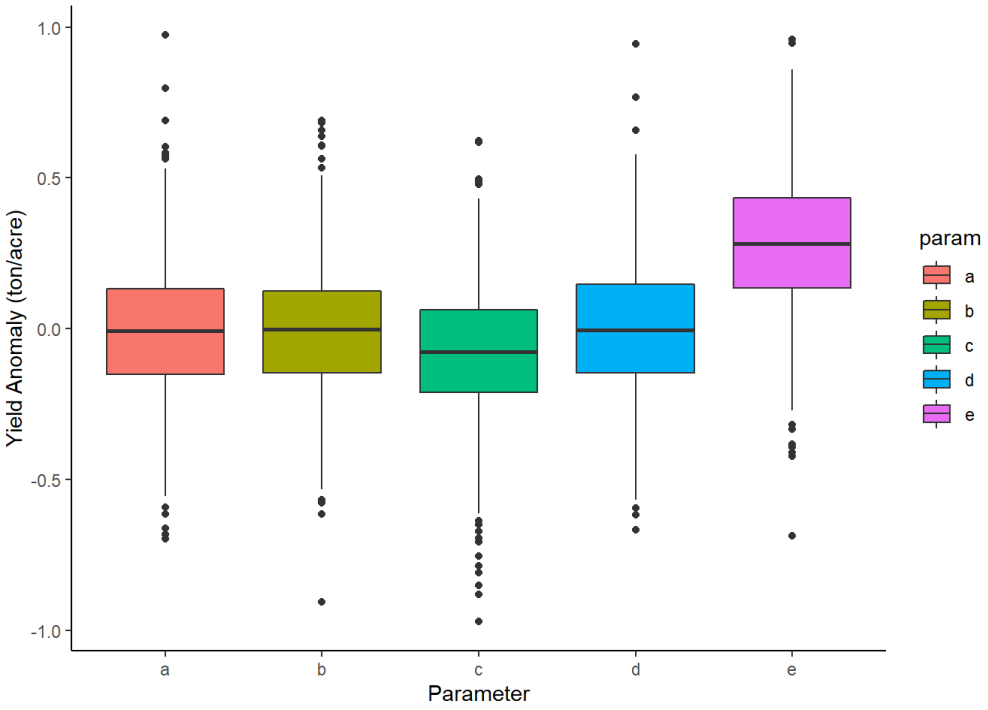

```
 # rank S
 rank_S <- sens_YA_sobel$S %>%
   select(original) %>%
   mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

colnames(rank_S) <-  c("PRCC", "Parameter")
rank_S
```

```
##            PRCC Parameter
## 1  1.254827e+00         d
## 2  2.363611e-04         b
## 3  3.024782e-05         a
## 4 -3.665323e-06         e
## 5 -1.506499e-06         c
```

```
 # rank T
 rank_T <- sens_YA_sobel$T %>%
   select(original) %>%
   mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

colnames(rank_T) <-  c("PRCC", "Parameter")
rank_T
```

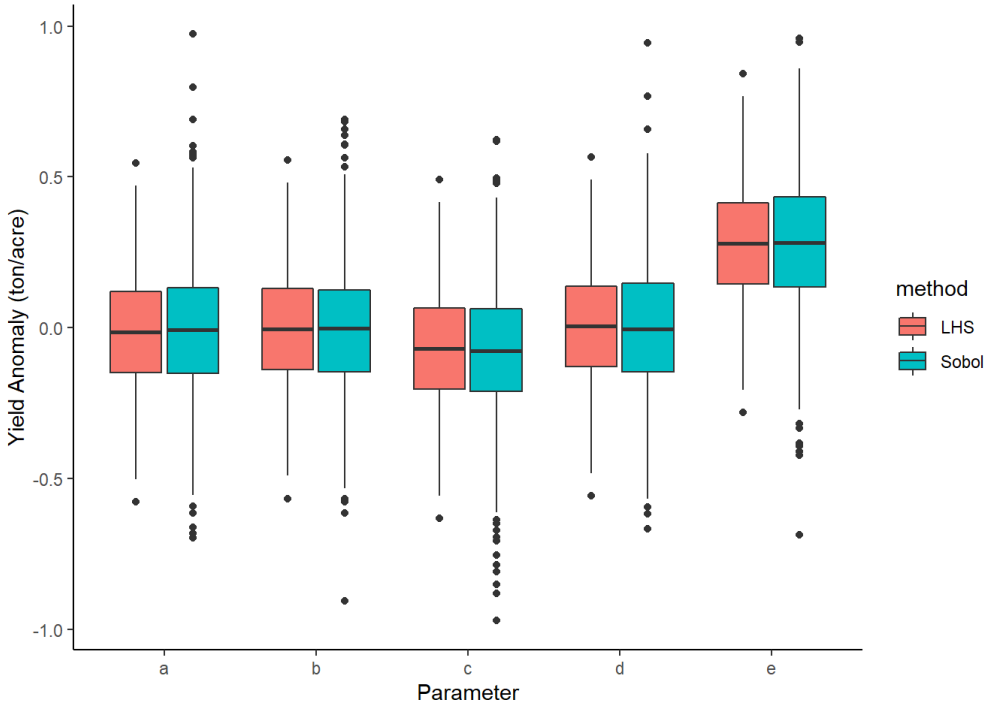
```
##            PRCC Parameter
## 1 -8.871306e-02         d
## 2 -2.421573e-04         b
## 3  7.621526e-05         c
## 4 -3.892448e-06         a
## 5  1.648905e-06         e
```

```
 # boxplot, comparing uncertainty of estimates
sens_YA_sobel_df <- sens_YA_sobel$X %>%
  gather(param, YA) %>%
  mutate(method = "Sobol")

sobol_boxplot <- ggplot(sens_YA_sobel_df, aes(param, YA, fill=param)) +
  geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  theme_classic()

sobol_boxplot
```



Comparing boxplots

```
# merging results for both methods
sens_both_df <- rbind(sens_YA_df, sens_YA_sobel_df)

# boxplot, comparing uncertainty of estimates
both_boxplot <- ggplot(sens_both_df, aes(x=param, y=YA, fill=method)) +
    geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  theme_classic()

both_boxplot
```



Doubling parameter sets

LHS

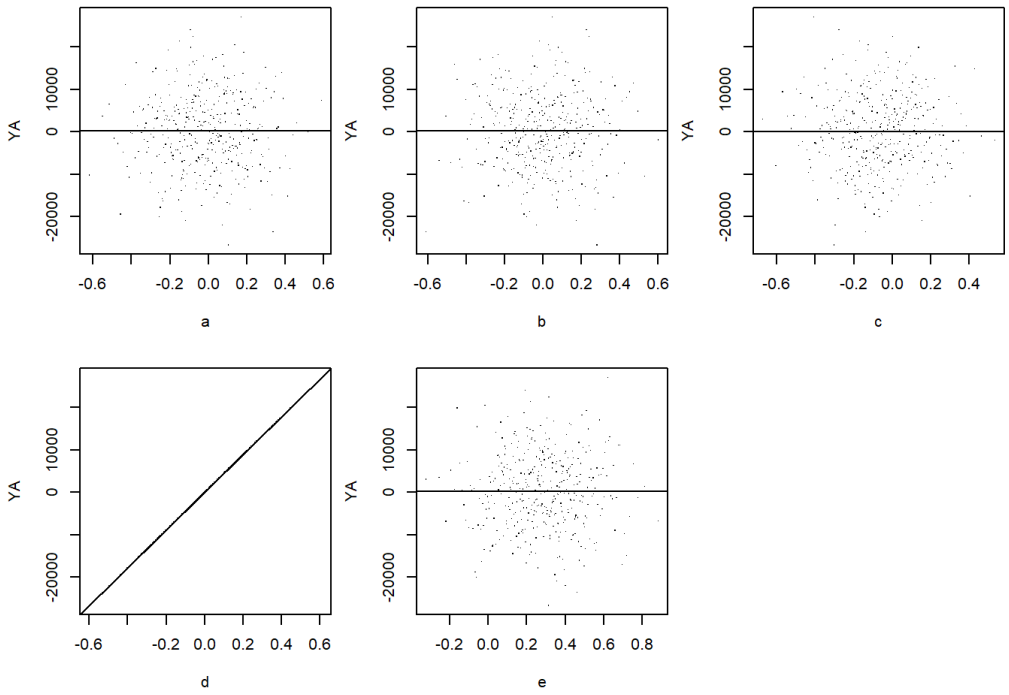
```
# names of our parameters: a = "Tmincoeff1", b = "Tmincoeff2", c = "Precipcoeff1", d = "Precipcoeff2", e = "Intercept"
factors = c("a", "b", "c", "d", "e")

# type of distributions they arise from
q = c("qnorm", "qnorm", "qnorm", "qnorm", "qnorm")

# parameters mean
a = -0.015
b = -0.0046
c = -0.07
d = 0.0043
e = 0.28

# parameters for those distributions
q.arg = list(list(mean=a, sd=0.2),
             list(mean=b, sd=0.2),
             list(mean=c, sd=0.2),
             list(mean=d, sd=0.2),
             list(mean=e, sd=0.2))

nsets=400
sens_YA = LHS(NULL,factors,nsets,q,q.arg)

# save parameter values
sens.pars = get.data(sens_YA)
head(sens.pars)
```

```
##             a           b           c           d         e
## 1 -0.05986614 -0.079643155 -0.282724388 -0.27175629 0.5528429
## 2  0.01709676 -0.283942529 -0.401419267 -0.05745308 0.3784621
## 3 -0.22994405 -0.003973342 -0.204112274  0.11134335 0.2630600
## 4 -0.01312000 -0.039238530 -0.151810809 -0.14101054 0.1213046
## 5 -0.22336190  0.079943880 -0.160058374 -0.31760646 0.2234843
## 6  0.12226760 -0.301633091 -0.004291099 -0.15785322 0.1521007
```

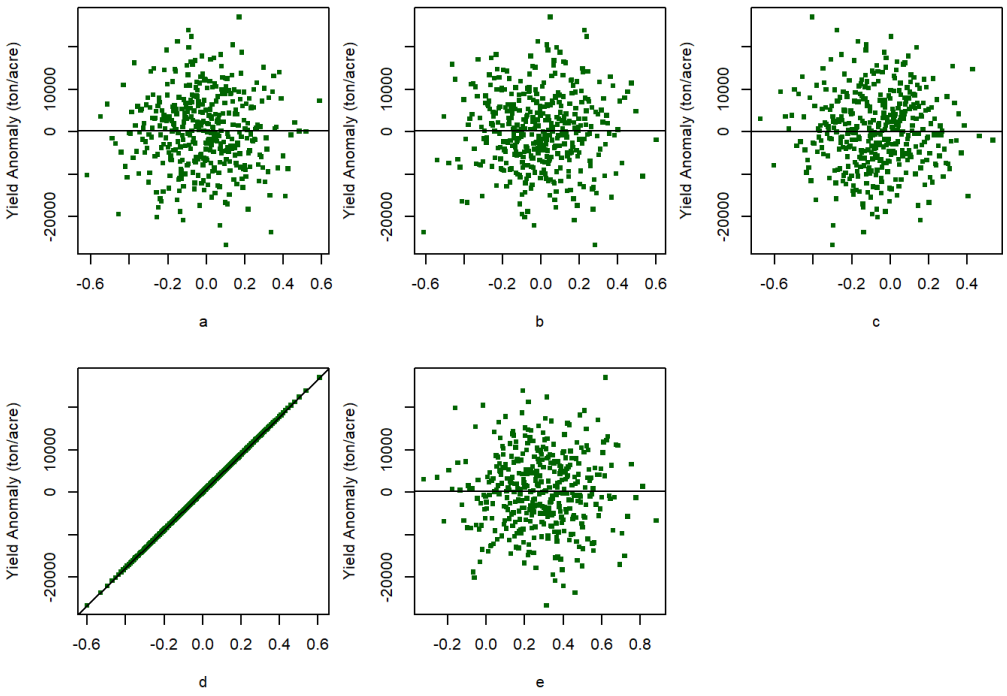
```
# run model
sens_results= mapply(FUN=almond_yield, a=sens.pars$a, b=sens.pars$b, c=sens.pars$c, d=sens.pars$d, e=sens.pars$e, MoreArgs=l
ist(clim_data = clim, mean_only=TRUE))
head(sens_results)
```

```
## [1] -12118.261  -2633.169   4917.082  -6288.453 -14125.025  -7041.914
```

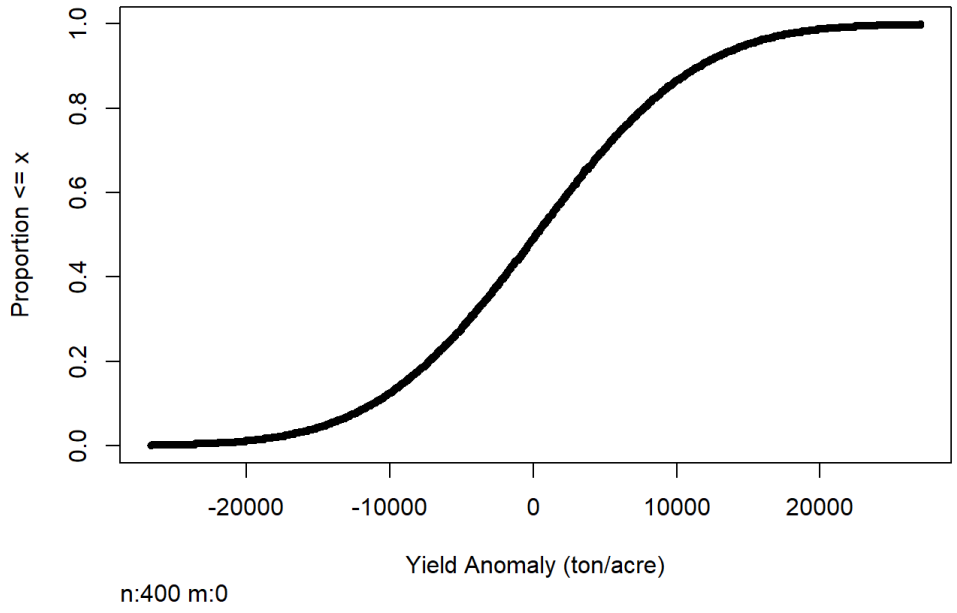
```
# use unlist to get a matrix
sens_res = matrix((unlist(sens_results)), ncol=1, byrow=TRUE)

sens_YA = pse::tell(sens_YA, t(sens_res), res.names="YA")
plotscatter(sens_YA)
```



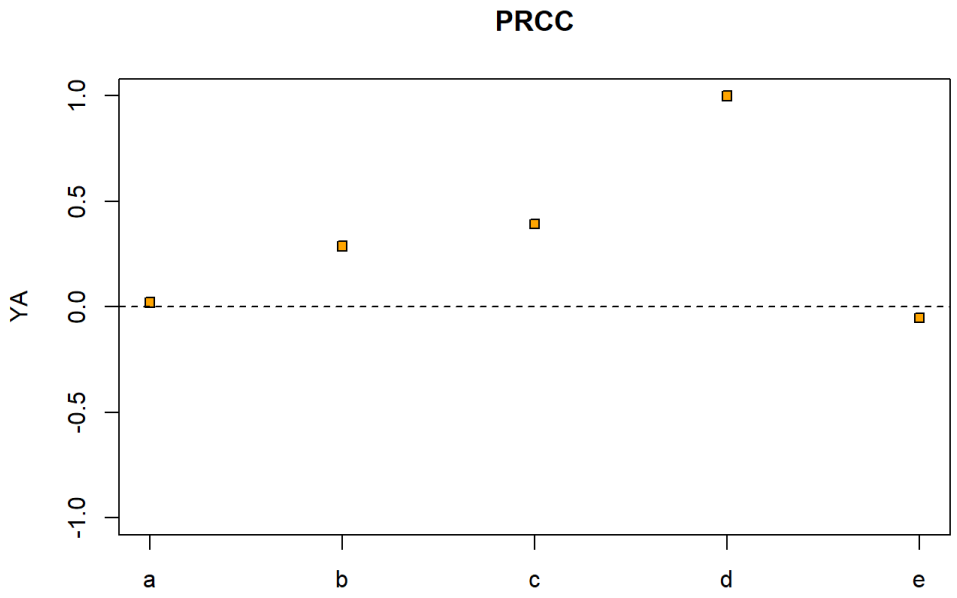

```
# note it can take standard R plotting parameters
plotscatter(sens_YA, col="darkgreen", cex=5, ylab="Yield Anomaly (ton/acre)")
```




```
# whats is the range of results
plotecdf(sens_YA, col="red", lwd=5, xlab="Yield Anomaly (ton/acre)")
```



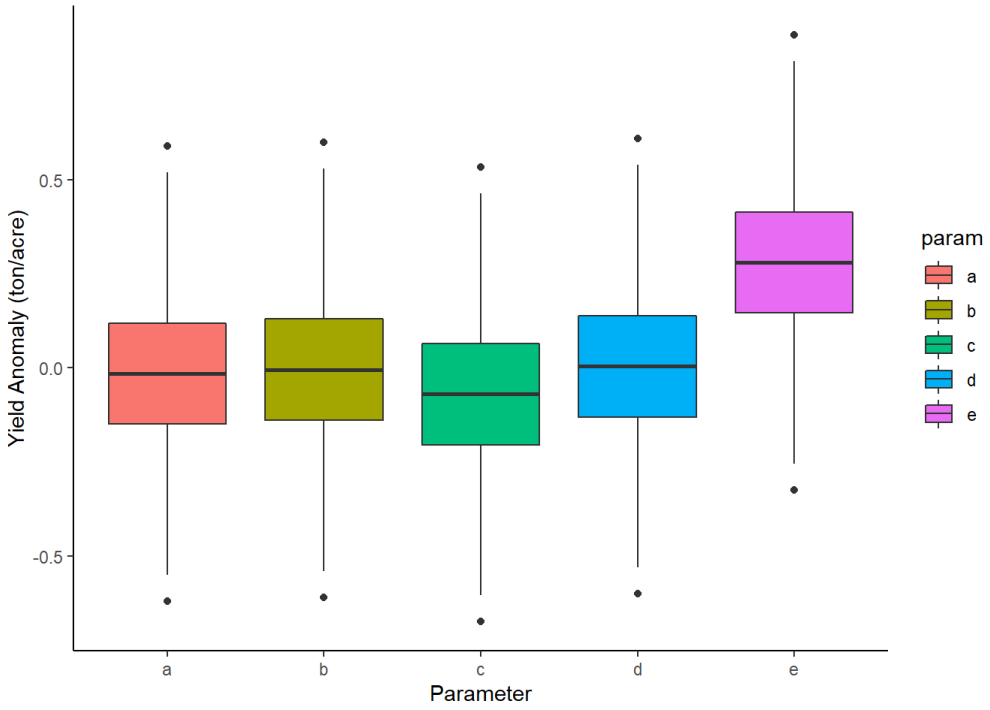

```
# we can also plot partial correlation coefficients
plotprcc(sens_YA)
```




```
# rank parameters
sens_YA_prcc <- sens_YA$prcc[[1]]$PRCC %>%
  mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

colnames(sens_YA_prcc) <-  c("PRCC", "Parameter")
sens_YA_prcc
```

```
##          PRCC Parameter
## 1  0.99999528         d
## 2  0.39287685         c
## 3  0.28881156         b
## 4 -0.05241764         e
## 5  0.02020351         a
```

```
# boxplot
sens_YA_df <- sens_YA$data %>%
  gather(param, YA) %>%
  mutate(method = "LHS")

lhs_boxplot <- ggplot(sens_YA_df, aes(param, YA, fill=param)) +
  geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  theme_classic()

lhs_boxplot
```



Sobol

```
# names of our parameters
# number of parameters

np=200
a = rnorm(mean=a, sd=0.2, n=np)
b = rnorm(mean=b, sd=0.2, n=np)
c = rnorm(mean=c, sd=0.2, n=np)
d = rnorm(mean=d, sd=0.2, n=np)
e = rnorm(mean=e, sd=0.2, n=np)

# generate two examples of random number from parmeter distributions

X1 = cbind.data.frame(a, b, c, d, e)

# repeat sampling
a = rnorm(mean=a, sd=0.2, n=np)
b = rnorm(mean=b, sd=0.2, n=np)
c = rnorm(mean=c, sd=0.2, n=np)
d = rnorm(mean=d, sd=0.2, n=np)
e = rnorm(mean=e, sd=0.2, n=np)

X2 = cbind.data.frame(a, b, c, d, e)

sens_YA_sobel = sobol2007(model = NULL, X1, X2, nboot = 100)


# run model for all parameter sets
res = mapply(FUN=almond_yield,  a=sens_YA_sobel$X$a,
             b=sens_YA_sobel$X$b,
             c=sens_YA_sobel$X$c,
             d=sens_YA_sobel$X$d,
             e=sens_YA_sobel$X$e,
             MoreArgs=list(clim_data = clim, mean_only=TRUE))


sens_YA_sobel = sensitivity::tell(sens_YA_sobel,res, res.names="YA")

# first-order indices (main effect without co-variance)
sens_YA_sobel$S
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a -9.294298e-07  5.163762e-06 2.947995e-05 -7.048233e-05 5.442798e-05
## b  2.121670e-04  1.493383e-05 2.676267e-04 -2.931960e-04 7.708079e-04
## c  3.803396e-04  9.814336e-06 3.465966e-04 -3.119395e-04 1.111084e-03
## d  1.043703e+00  3.068168e-02 1.655724e-01  5.941222e-01 1.306262e+00
## e -1.143324e-06 -3.687396e-07 2.303156e-06 -5.042050e-06 4.165153e-06
```

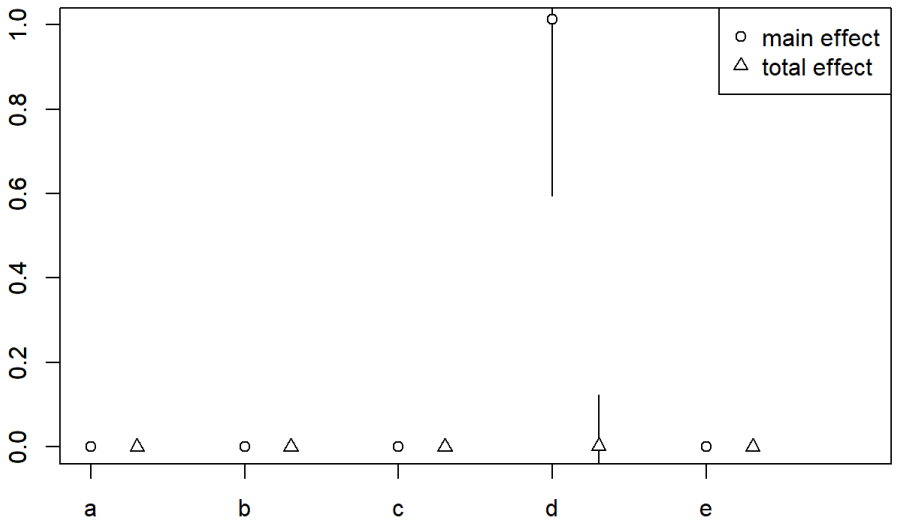
```
# total sensitivity index -note that this partitions the output variance - so values sum to 1
sens_YA_sobel$T
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a  3.936153e-07 -1.281999e-07 2.218707e-05 -4.185466e-05 4.368821e-05
## b -5.739246e-05 -2.224006e-05 1.598422e-04 -4.171752e-04 2.924882e-04
## c -3.011468e-04 -1.935396e-05 2.127891e-04 -7.803752e-04 1.201245e-04
## d  2.128521e-04 -2.013219e-03 6.452268e-02 -1.188779e-01 1.226902e-01
## e  5.258790e-07  1.553457e-07 1.612391e-06 -3.399229e-06 3.826086e-06
```

```
# The difference between the main effect and total effect can tell us something about how the parameter influences results
# so in the main effect we include interactions with other parameters
print(sens_YA_sobel)
```

```
##
## Call:
## sobol2007(model = NULL, X1 = X1, X2 = X2, nboot = 100)
##
## Model runs: 1400
##
## First order indices:
##        original          bias   std. error     min. c.i.    max. c.i.
## a -9.294298e-07  5.163762e-06 2.947995e-05 -7.048233e-05 5.442798e-05
## b  2.121670e-04  1.493383e-05 2.676267e-04 -2.931960e-04 7.708079e-04
## c  3.803396e-04  9.814336e-06 3.465966e-04 -3.119395e-04 1.111084e-03
## d  1.043703e+00  3.068168e-02 1.655724e-01  5.941222e-01 1.306262e+00
## e -1.143324e-06 -3.687396e-07 2.303156e-06 -5.042050e-06 4.165153e-06
##
## Total indices:
##        original          bias   std. error     min. c.i.    max. c.i.
## a  3.936153e-07 -1.281999e-07 2.218707e-05 -4.185466e-05 4.368821e-05
## b -5.739246e-05 -2.224006e-05 1.598422e-04 -4.171752e-04 2.924882e-04
## c -3.011468e-04 -1.935396e-05 2.127891e-04 -7.803752e-04 1.201245e-04
## d  2.128521e-04 -2.013219e-03 6.452268e-02 -1.188779e-01 1.226902e-01
## e  5.258790e-07  1.553457e-07 1.612391e-06 -3.399229e-06 3.826086e-06
```

```
plot(sens_YA_sobel)
```




```
# compare with LHS and PRCC
sens_YA$prcc
```

```
## [[1]]
##
## Call:
## pcc.default(X = L, y = r, rank = T, nboot = nboot)
##
## Partial Rank Correlation Coefficients (PRCC):
##      original
## a  0.02020351
## b  0.28881156
## c  0.39287685
## d  0.99999528
## e -0.05241764
```

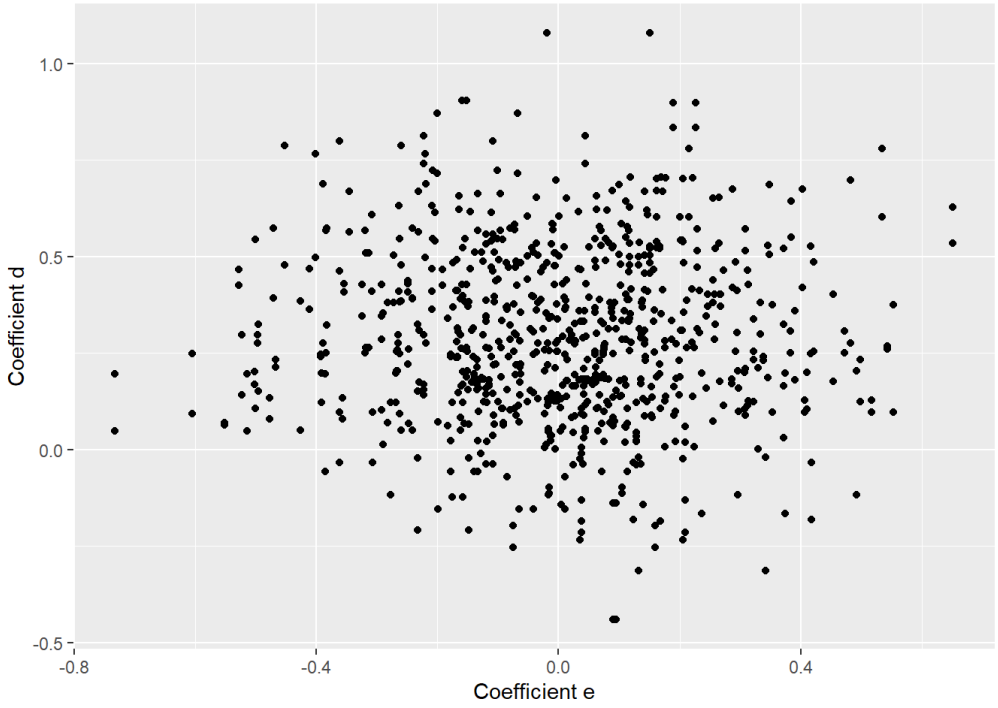
```
sens_YA_sobel$S
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a -9.294298e-07  5.163762e-06 2.947995e-05 -7.048233e-05 5.442798e-05
## b  2.121670e-04  1.493383e-05 2.676267e-04 -2.931960e-04 7.708079e-04
## c  3.803396e-04  9.814336e-06 3.465966e-04 -3.119395e-04 1.111084e-03
## d  1.043703e+00  3.068168e-02 1.655724e-01  5.941222e-01 1.306262e+00
## e -1.143324e-06 -3.687396e-07 2.303156e-06 -5.042050e-06 4.165153e-06
```

```
sens_YA_sobel$T
```

```
##        original          bias   std. error     min. c.i.    max. c.i.
## a  3.936153e-07 -1.281999e-07 2.218707e-05 -4.185466e-05 4.368821e-05
## b -5.739246e-05 -2.224006e-05 1.598422e-04 -4.171752e-04 2.924882e-04
## c -3.011468e-04 -1.935396e-05 2.127891e-04 -7.803752e-04 1.201245e-04
## d  2.128521e-04 -2.013219e-03 6.452268e-02 -1.188779e-01 1.226902e-01
## e  5.258790e-07  1.553457e-07 1.612391e-06 -3.399229e-06 3.826086e-06
```

```
# make a data frame for plotting
both = cbind.data.frame(sens_YA_sobel$X, gs=sens_YA_sobel$y)

# look at response of conductance to the two most important variables
 ggplot(both, aes(d, e))+geom_point()+labs(y="Coefficient d", x="Coefficient e")
```



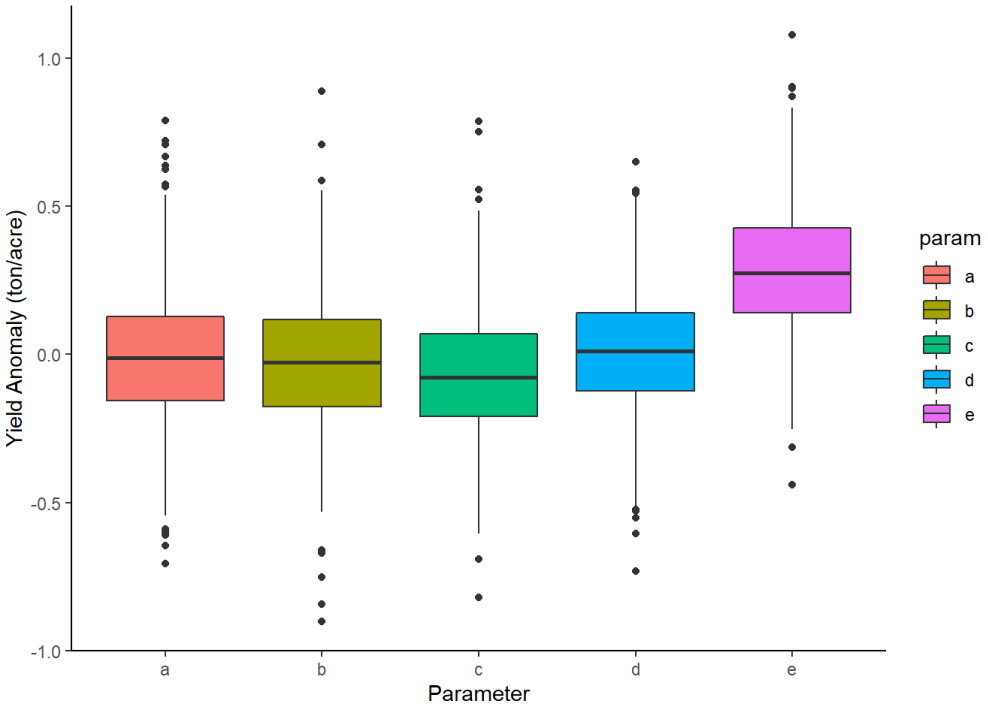

```
 # rank S
 rank_S <- sens_YA_sobel$S %>%
   select(original) %>%
   mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

colnames(rank_S) <-  c("PRCC", "Parameter")
rank_S
```

```
##            PRCC Parameter
## 1  1.043703e+00         d
## 2  3.803396e-04         c
## 3  2.121670e-04         b
## 4 -1.143324e-06         e
## 5 -9.294298e-07         a
```

```
 # rank T
 rank_T <- sens_YA_sobel$T %>%
   select(original) %>%
   mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

colnames(rank_T) <-  c("PRCC", "Parameter")
rank_T
```

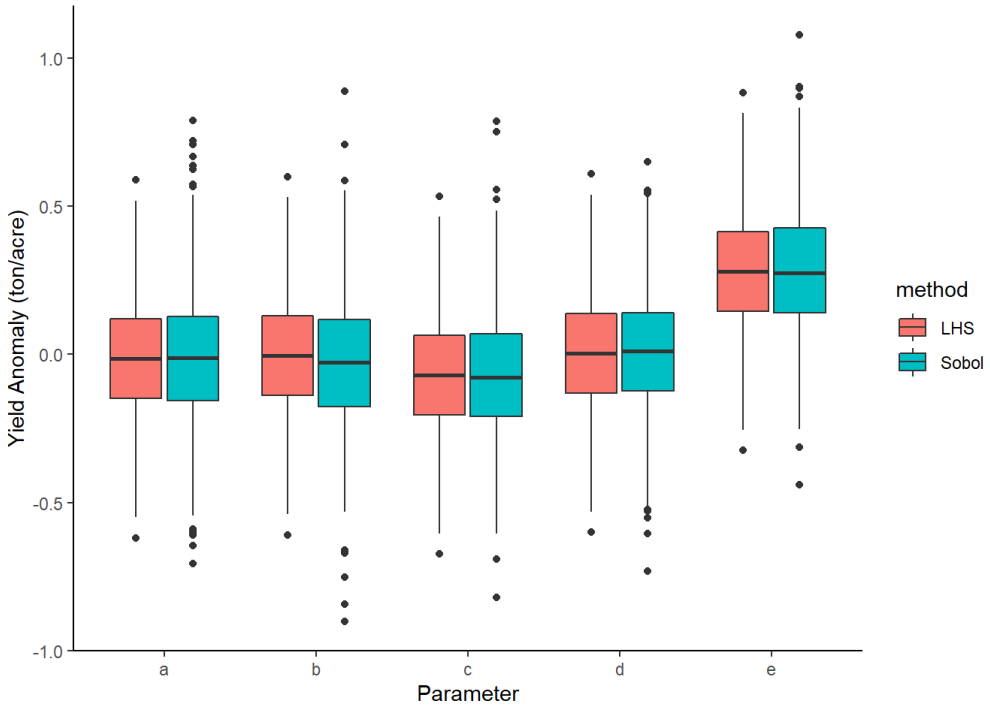
```
##            PRCC Parameter
## 1 -3.011468e-04         c
## 2  2.128521e-04         d
## 3 -5.739246e-05         b
## 4  5.258790e-07         e
## 5  3.936153e-07         a
```

```
 # boxplot, comparing uncertainty of estimates
sens_YA_sobel_df <- sens_YA_sobel$X %>%
  gather(param, YA) %>%
  mutate(method = "Sobol")

sobol_boxplot <- ggplot(sens_YA_sobel_df, aes(param, YA, fill=param)) +
  geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  theme_classic()

sobol_boxplot
```



Comparing boxplots

```
# merging results for both methods
sens_both_df <- rbind(sens_YA_df, sens_YA_sobel_df)

# boxplot, comparing uncertainty of estimates
both_boxplot <- ggplot(sens_both_df, aes(x=param, y=YA, fill=method)) +
    geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  theme_classic()

both_boxplot
```



There was no significant difference in model sensitivities between the LHS and Sobel methods. Almond yield was the most sensitive to changes in variable D (Precipcoeff2). The partial rank regression coefficients (PRCC) were highest for parameter D, and lowest for parameter E. This effect was nearly identical between the LHS and Sobol methods. Parameter D is the squared value of the amount of January precipitation in the current harvest year. It makes sense that almond yield would be the most sensitive to this parameter because winter rainfall can have a significant impact on plant productivity in the current year. Prior year precipitation will impact individual plant growth from the prior season so may give a plant more branches on which to grow fruit. However, current season precipitation dictates how much energy that plant can dedicate to fruit production right now. Minimum temperatures can have significant impacts on plant productivity if there is a hard freeze, but most these event are not common and most farmers have management strategies to mitigate sub-optimal temperatures.

Doubling the parameter sets reduced the uncertainty in the estimates, but did not appear to change the estimated sensitivities themselves.