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.

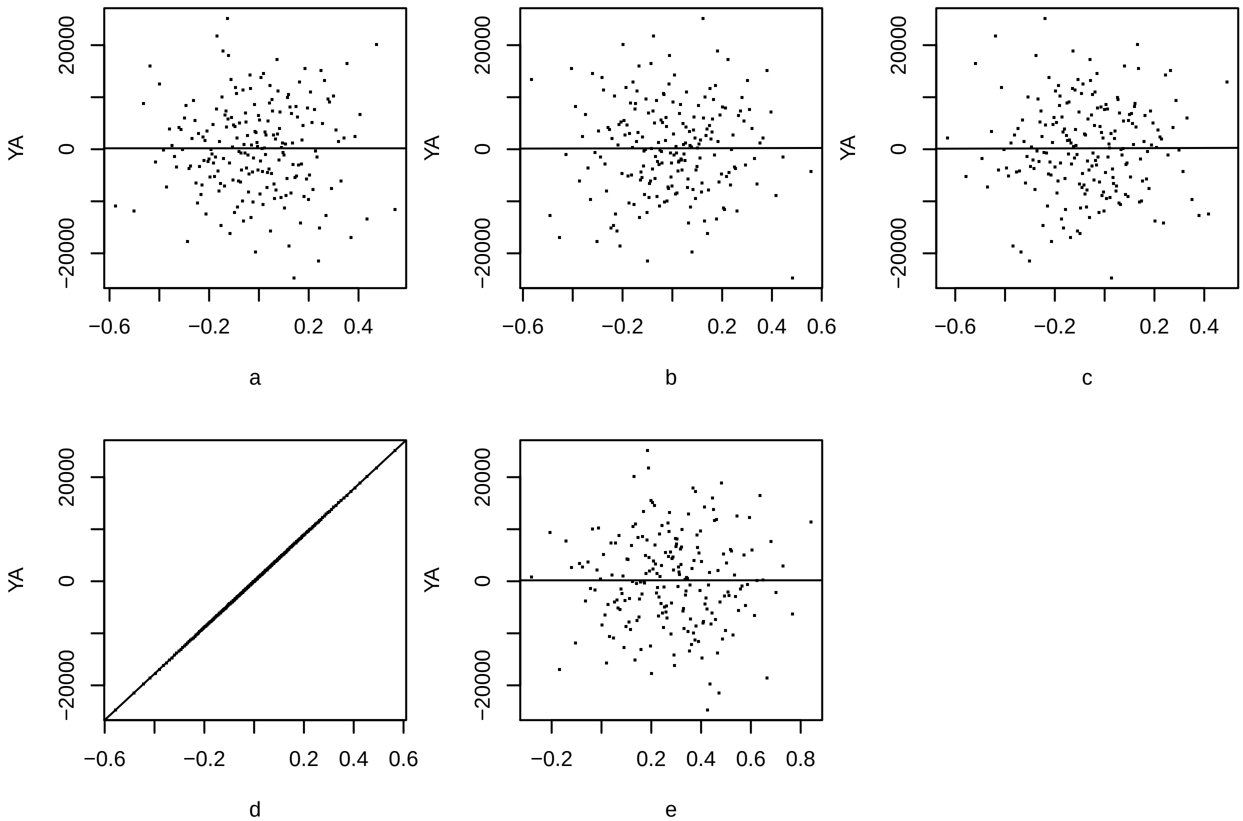
```r
# Adjusted almond yield function
almond_yield <- function( clim_data = clim,
                          a=-0.015,
                          b=-0.0046,
                          c=-0.07,
                          d=0.0043,
                          e=0.28,
                          mean_only = TRUE) {

  # Add in some error checking into the function

  # Make sure the climate data input is a dataframe
  if(class(clim_data) != "data.frame") return("Climate data input must be a data frame")

  # Make sure the climate data input contains the columns year, month, precip, tmin_c and tmax_c
  if(!all(has_name(clim_data, c("month", "year",  "month", "precip", "tmin_c", "tmax_c")))) return("Cli

  # Make sure that the input for precipitation is larger than 0
  clim_data$precip = ifelse(clim_data$precip < 0, return("Input for precipitation must be a value large

  # Make sure that the maximum tempertaure will be larger than the minimum temperature
  clim_data$tmin_c = ifelse(clim_data$tmin_c > clim_data$tmax_c, return("Input for maximum temperature 

  # Average monthly maximum daily temperature, and monthly precipitation from a data frame called clim

  clim_month <-  clim_data %>%
    group_by(month, year) %>%
    dplyr::summarise(meantmin = mean(tmin_c),
              meantmax = mean(tmax_c),
              precip=sum(precip))
```

```r
  # Filter Jan and Feb data
  jan <- clim_month %>%
    filter(month==1)

  feb <- clim_month %>%
    filter(month==2)

  # Change column names
  colnames(jan) <- c("month", "year", "Tn", "Tm", "P")
  colnames(feb) <- c("month", "year", "Tn", "Tm", "P")

  # Data structure for yield annomalies
  yield_df <- data.frame(year = jan$year, YA = NA)

  # Loop through each year
  for (i in 1:length(yield_df$year)) {
    yield_df$YA[i] = a*feb$Tn[i] + b*(feb$Tn[i]^2) +c*jan$P[i] + d*(jan$P[i]^2) + e
  }

  # Calculate max and min yields
  max_yield <- yield_df %>%
    arrange(-abs(YA)) %>%
    head(1)

  min_yield <- yield_df %>%
    arrange(abs(YA)) %>%
    head(1)

  # Change column names of max and min yields
  colnames(max_yield) <- c("Year", "Maximum Yield Anomaly")
  colnames(min_yield) <- c("Year", "Minimum Yield Anomaly")

  # Create list with three elements
  yield_list <- list(yield_df, max_yield, min_yield)

  # Mean anomaly
  mean_anom <- mean(yield_df$YA)

  # Return list
  ifelse(mean_only == TRUE, return (mean_anom), return(yield_list))
}
```

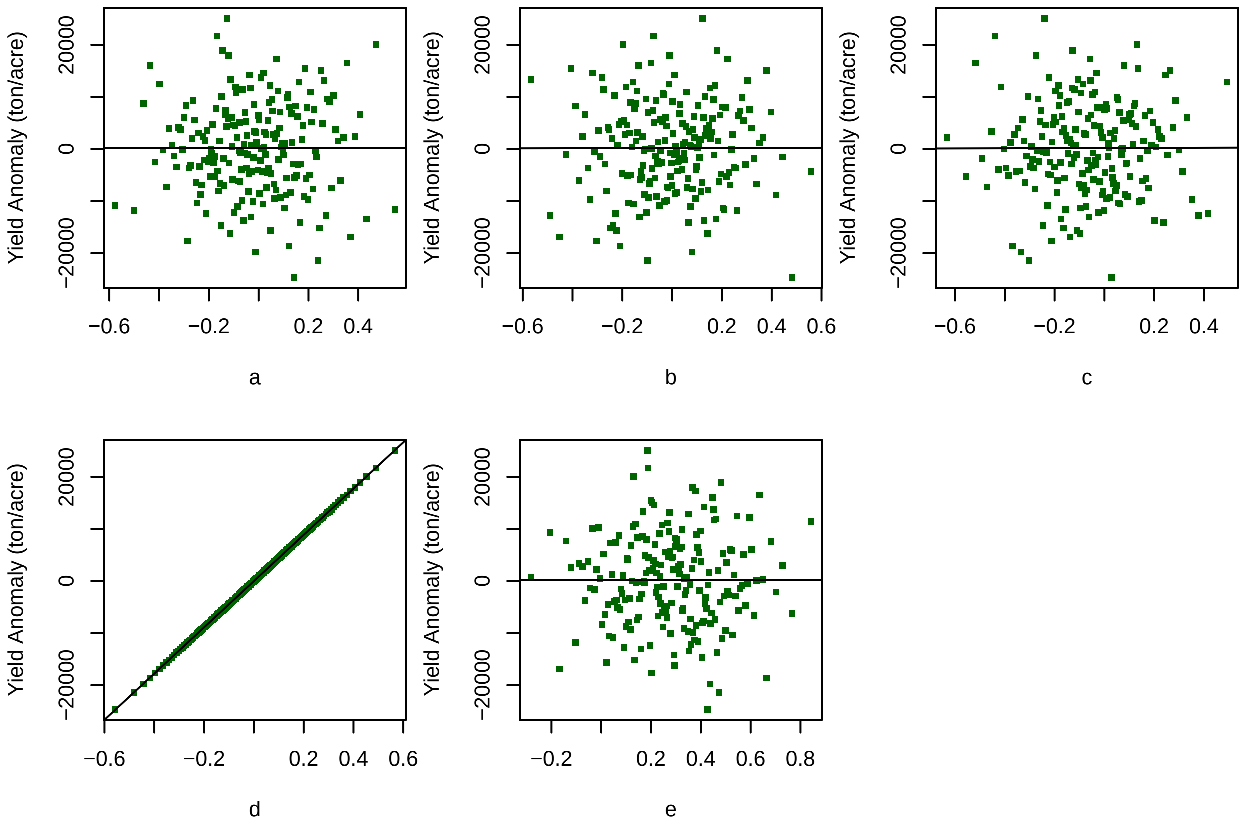
LHS

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

# source function
source("almond_yield.R")

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

# type of distributions they arise from
```

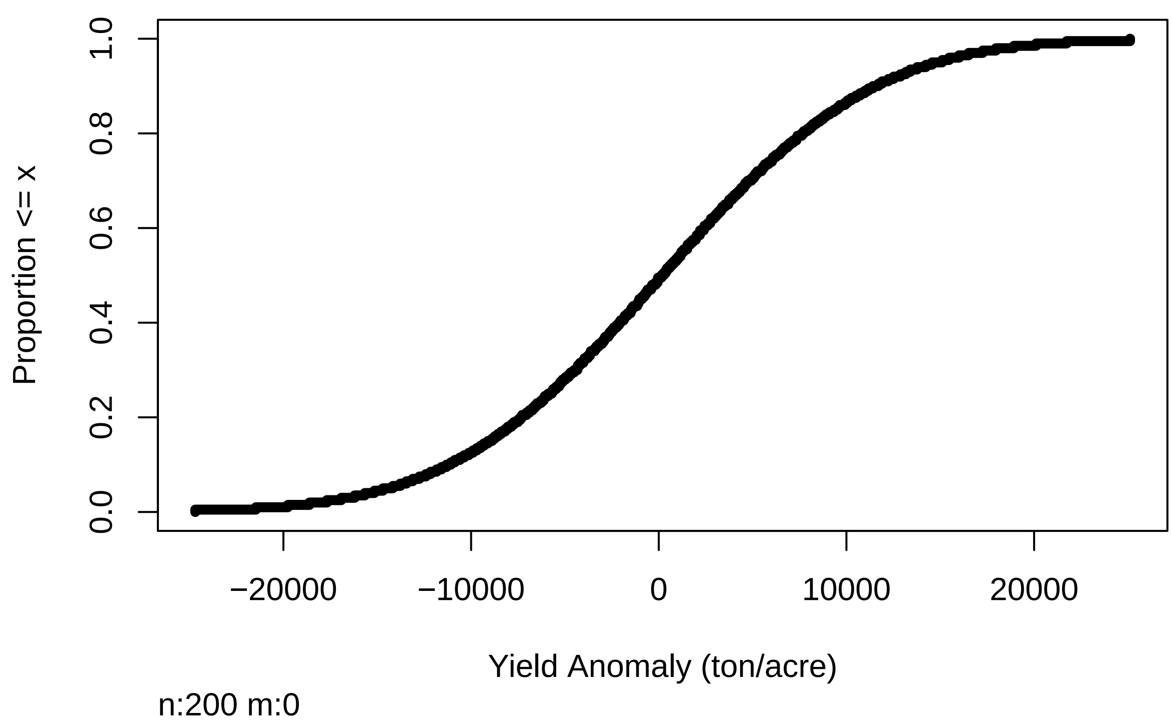
```r
q = c("qnorm", "qnorm", "qnorm", "qnorm", "qnorm")

# 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)

# 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=sen

# 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)
```



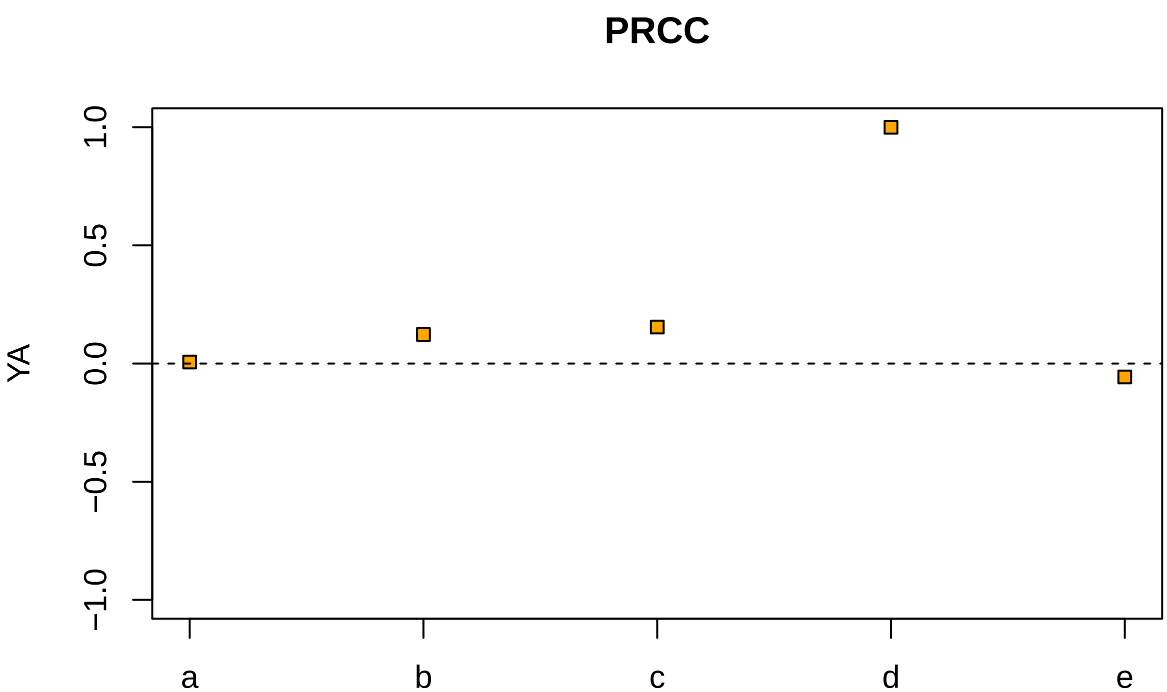

```
# plot parameters
plotscatter(sens_YA, col="darkgreen", cex=5, ylab="Yield Anomaly (ton/acre)")
```




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



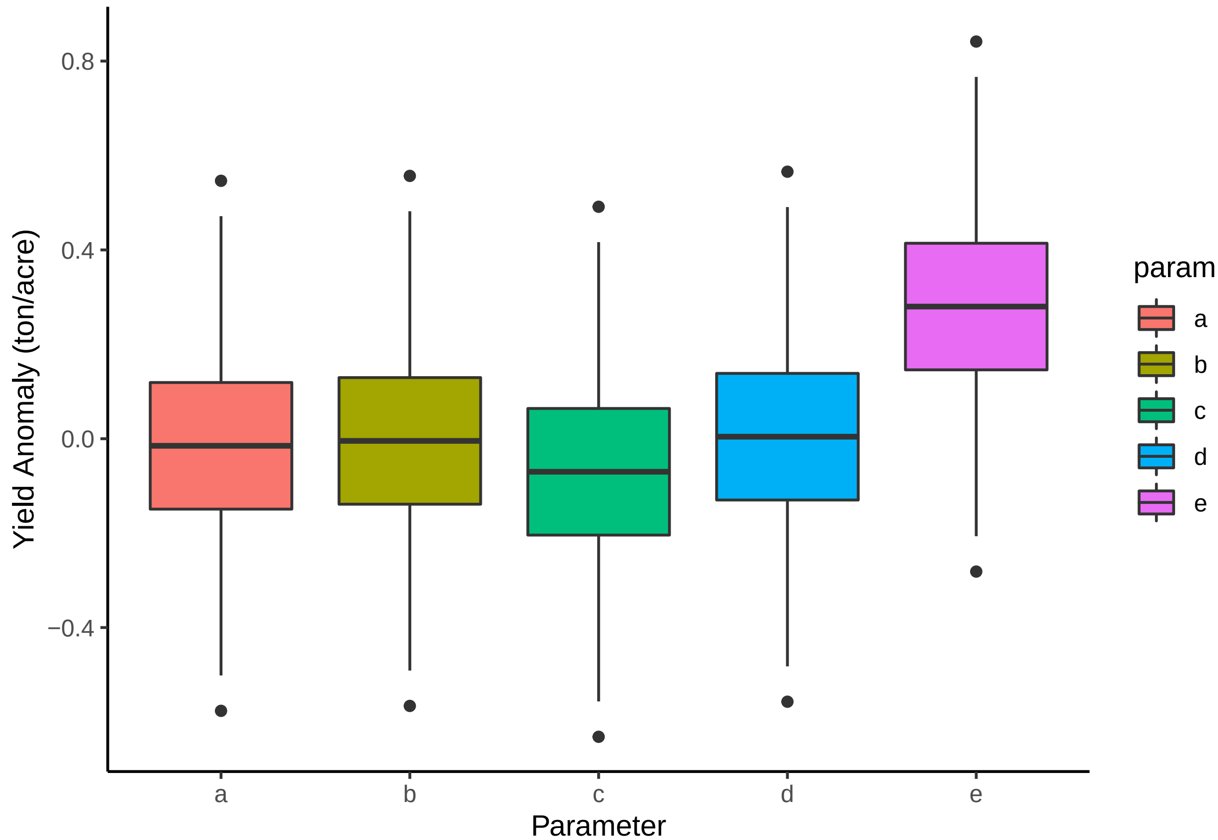

```r
# partial correlation coefficients
plotprcc(sens_YA)
```




```r
# 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  0.999998563         d
## 2  0.154484308         c
## 3  0.123130008         b
## 4 -0.056919643         e
## 5  0.006381968         a
```

```r
# 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

```
# 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,
```

```r
             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 -7.843722e-07 -5.119799e-06 2.593014e-05 -5.847805e-05 5.804058e-05
## b  2.264118e-04  1.994662e-05 2.597544e-04 -3.145564e-04 6.780189e-04
## c  9.456913e-05 -5.839788e-05 3.840581e-04 -6.104142e-04 9.465878e-04
## d  1.095622e+00  3.106956e-02 1.970467e-01  5.793173e-01 1.369586e+00
## e -2.494831e-06 -4.962021e-07 2.186955e-06 -6.384093e-06 3.127397e-06
```

```r
# total sensitivity index
sens_YA_sobel$T
```

```
##         original          bias   std. error     min. c.i.    max. c.i.
## a -5.046446e-06  3.702125e-07 1.655239e-05 -3.682814e-05 3.487342e-05
## b -1.898806e-04 -9.065372e-06 1.515950e-04 -4.956258e-04 1.122764e-04
## c -6.119480e-06  6.338073e-05 2.187590e-04 -5.049939e-04 3.191783e-04
## d -5.406868e-02 -1.738735e-03 6.051915e-02 -1.817092e-01 6.343230e-02
## e  9.272380e-08 -6.272184e-08 1.499005e-06 -3.246148e-06 3.559567e-06
```

```r
# 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  0.006381968
## b  0.123130008
## c  0.154484308
## d  0.999998563
## e -0.056919643
```

```r
sens_YA_sobel$S
```

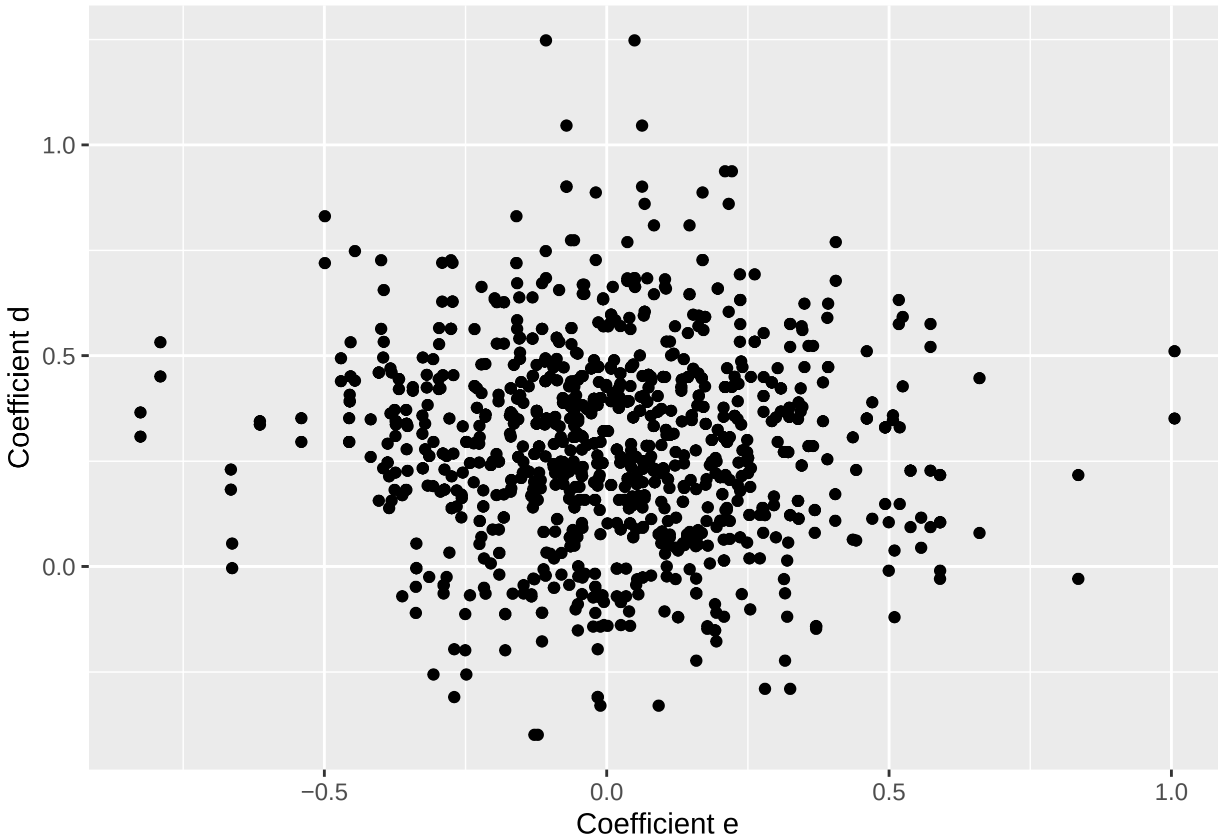
```
##         original          bias   std. error     min. c.i.    max. c.i.
## a -7.843722e-07 -5.119799e-06 2.593014e-05 -5.847805e-05 5.804058e-05
## b  2.264118e-04  1.994662e-05 2.597544e-04 -3.145564e-04 6.780189e-04
## c  9.456913e-05 -5.839788e-05 3.840581e-04 -6.104142e-04 9.465878e-04
## d  1.095622e+00  3.106956e-02 1.970467e-01  5.793173e-01 1.369586e+00
## e -2.494831e-06 -4.962021e-07 2.186955e-06 -6.384093e-06 3.127397e-06
```

```r
sens_YA_sobel$T
```

```
##         original          bias   std. error     min. c.i.    max. c.i.
```

```
## a -5.046446e-06  3.702125e-07 1.655239e-05 -3.682814e-05 3.487342e-05
## b -1.898806e-04 -9.065372e-06 1.515950e-04 -4.956258e-04 1.122764e-04
## c -6.119480e-06  6.338073e-05 2.187590e-04 -5.049939e-04 3.191783e-04
## d -5.406868e-02 -1.738735e-03 6.051915e-02 -1.817092e-01 6.343230e-02
## e  9.272380e-08 -6.272184e-08 1.499005e-06 -3.246148e-06 3.559567e-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")
```



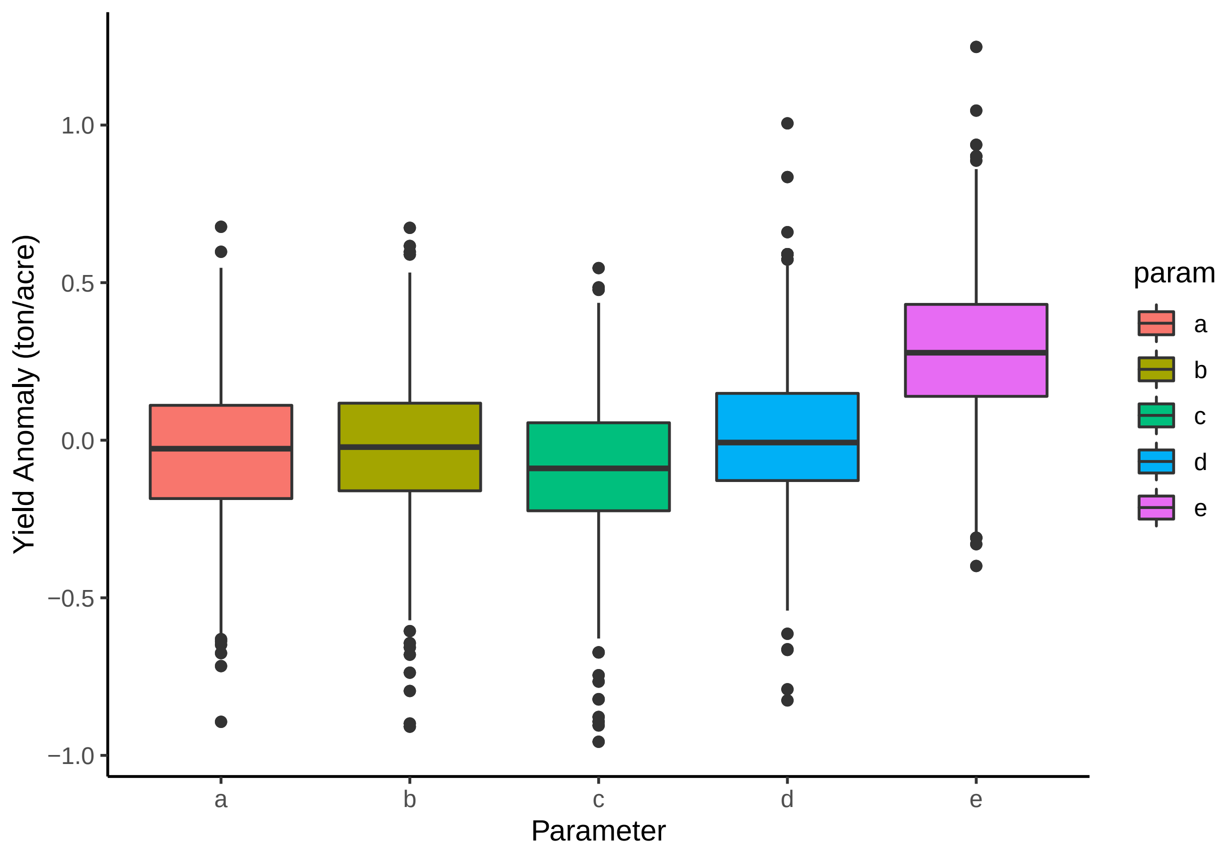

```
 # rank S
 rank_S <- sens_YA_sobel$S %>%
   select(original) %>%
   mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

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

```
##            PRCC Parameter
## 1  1.095622e+00         d
## 2  2.264118e-04         b
## 3  9.456913e-05         c
## 4 -2.494831e-06         e
## 5 -7.843722e-07         a
```

```r
# 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 -5.406868e-02         d
## 2 -1.898806e-04         b
## 3 -6.119480e-06         c
## 4 -5.046446e-06         a
## 5  9.272380e-08         e
```

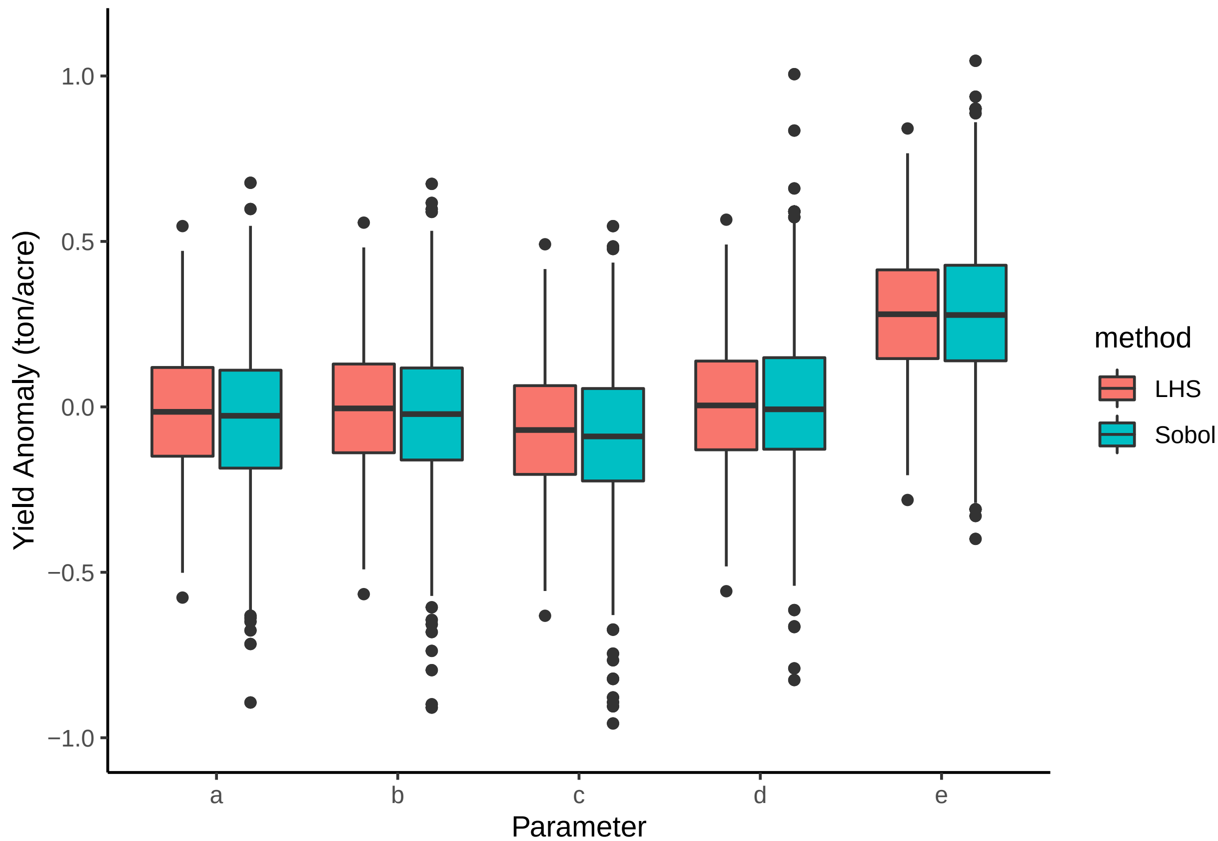
```r
# boxplot, comparing uncertainty of estimates
sens_YA_sobel_df <- sens_YA_sobel$X %>%
  gather(param, YA) %>%
  mutate(method = "Sobol")

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

```r
# 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") +
  ylim(-1, 1.1) +
  theme_classic()

both_boxplot
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```



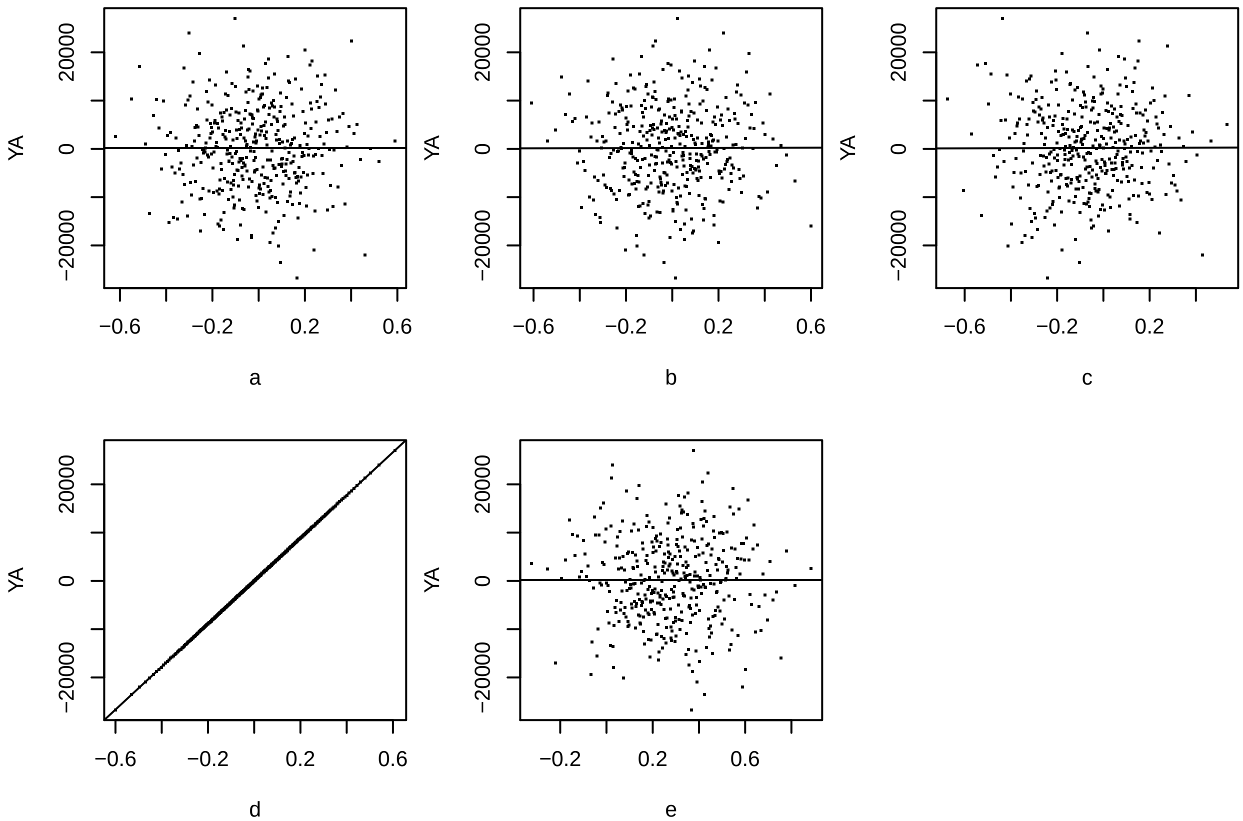
Double number of parameters LHS

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

# source function
source("almond_yield.R")

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

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

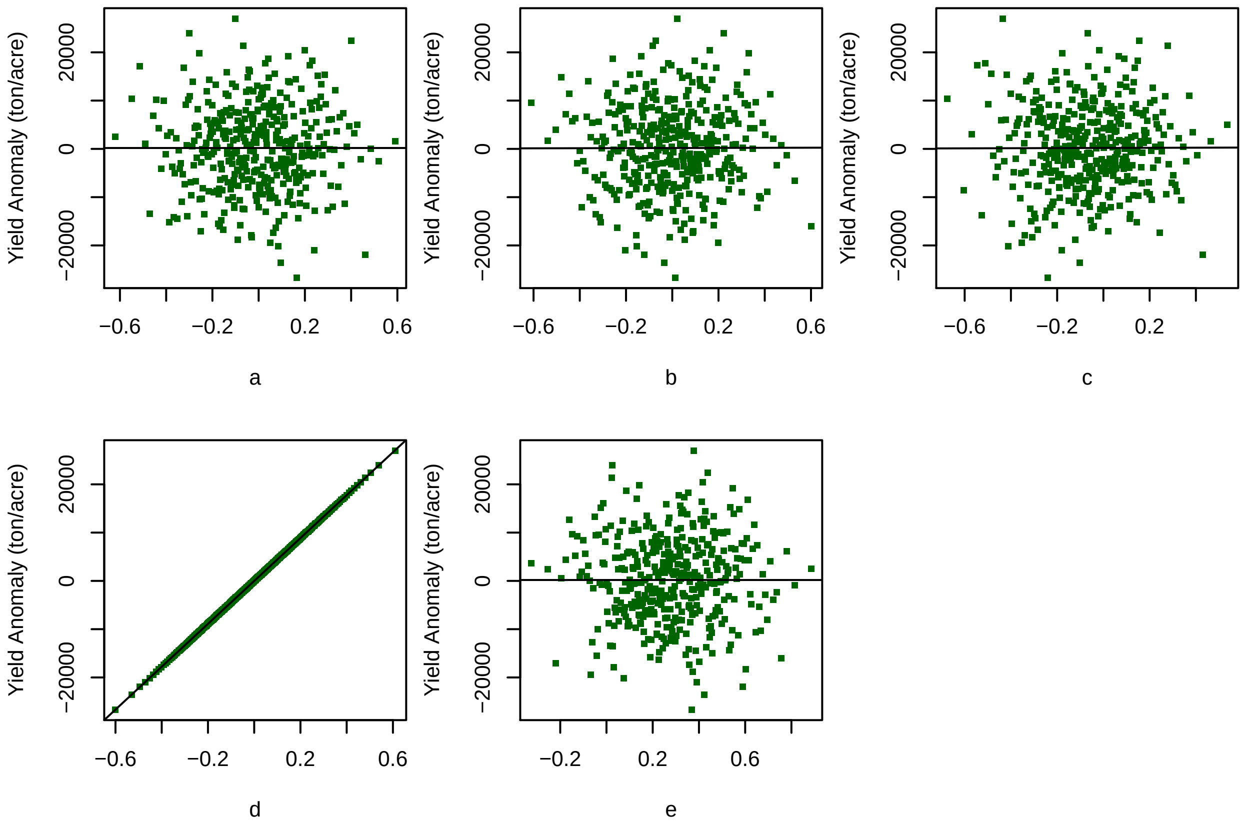
```r
# parameters mean
a = -0.015
b = -0.0046
c = -0.07
d = 0.0043
e = 0.28

# 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)

# 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=sen
head(sens_results)
```

```
## [1] -5831.75132  8243.25063  8226.08006  7463.39896    84.38123  -406.84979
```

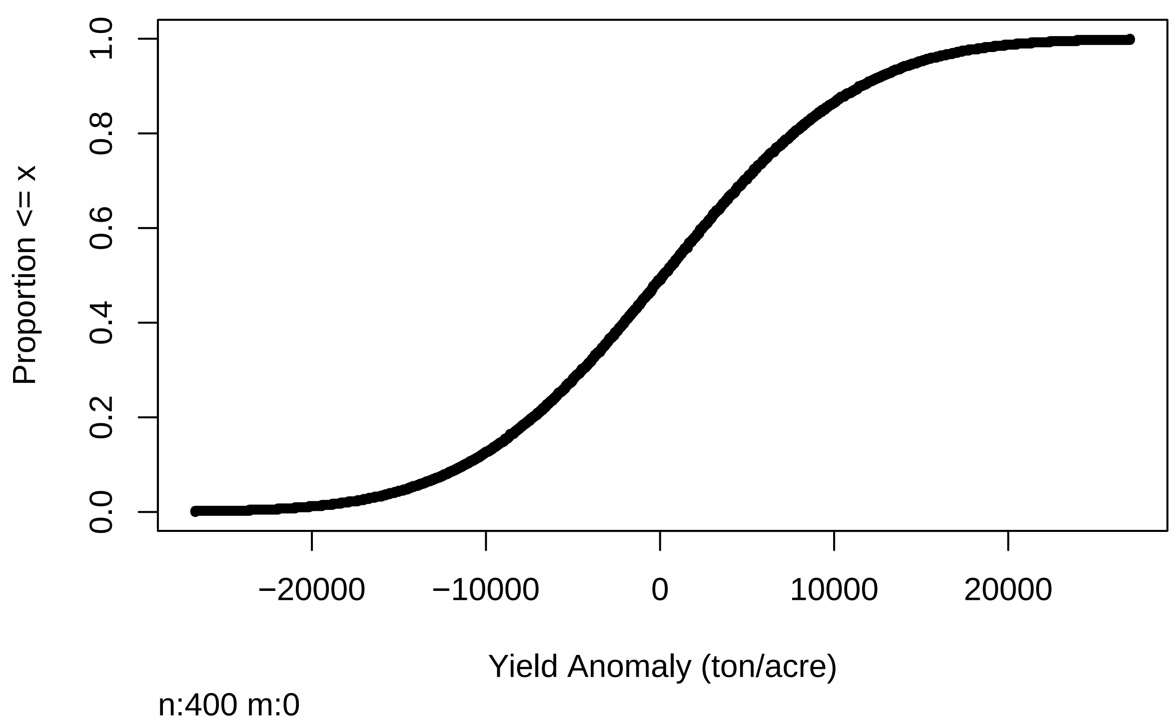
```r
# use unlist to get a matrix
sens_res = matrix((unlist(sens_results)), ncol=1, byrow=TRUE)

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



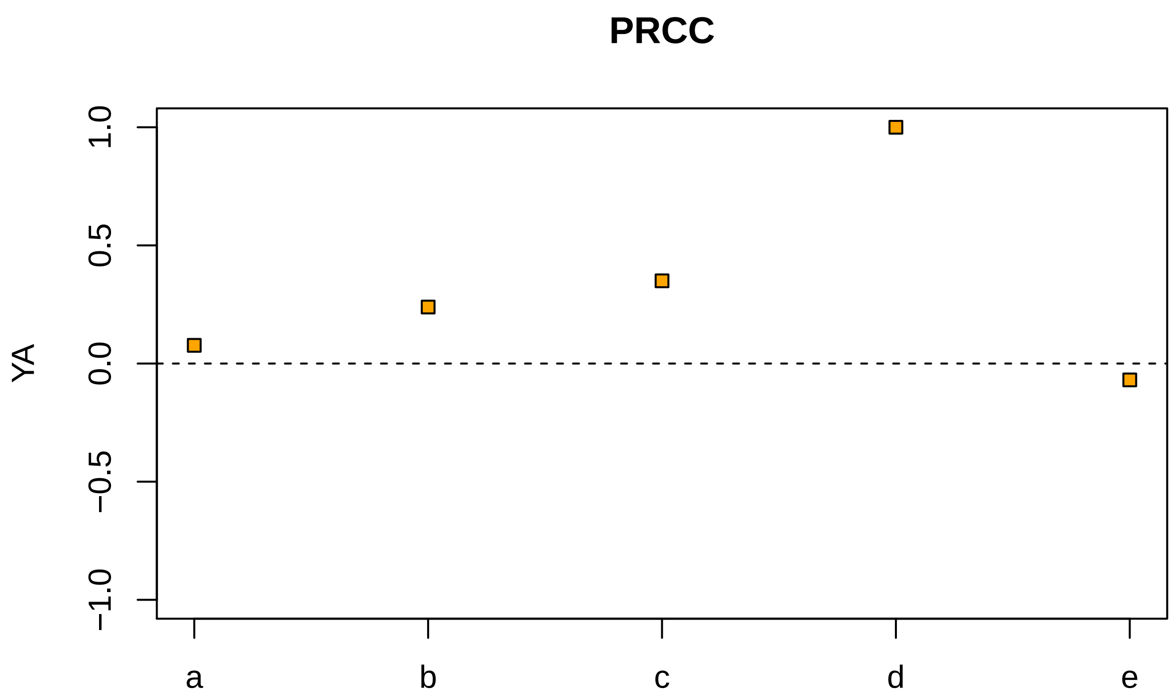

```
# plot parameters
plotscatter(sens_YA, col="darkgreen", cex=5, ylab="Yield Anomaly (ton/acre)")
```




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



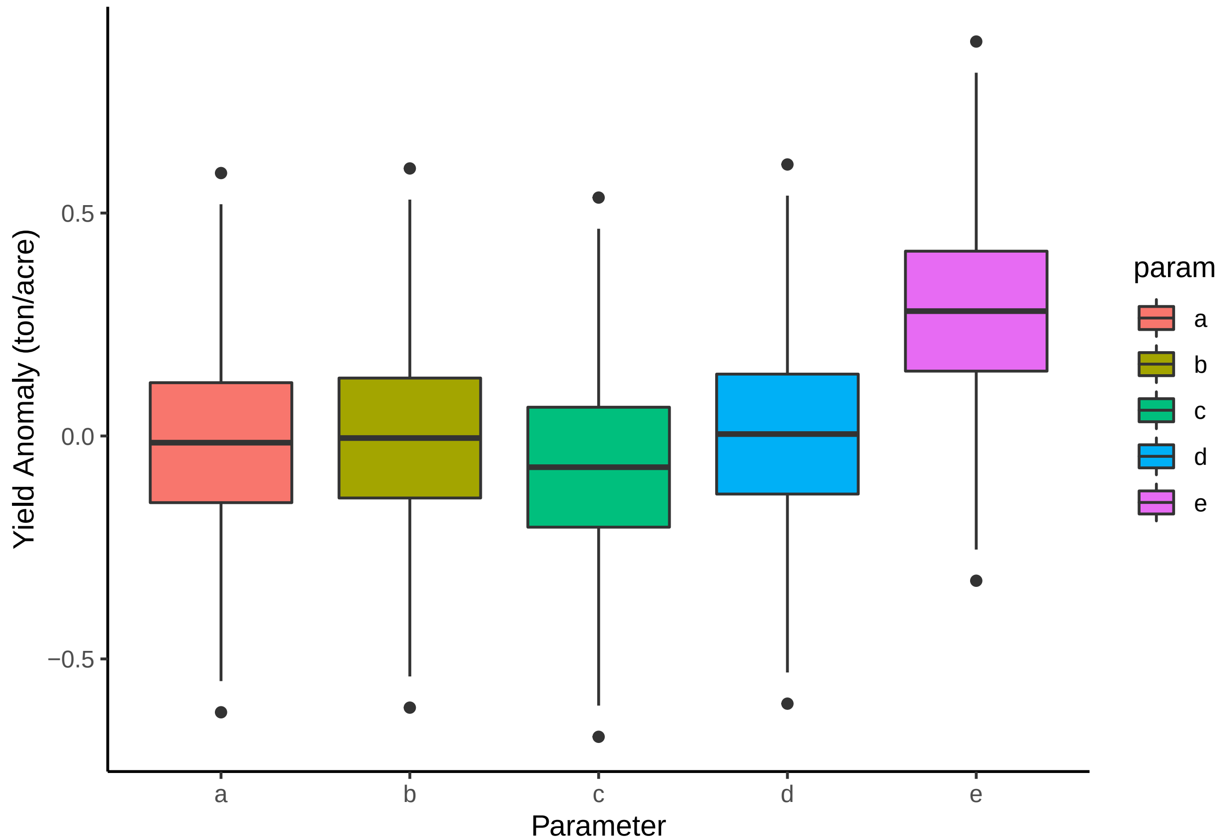

```r
# partial correlation coefficients
plotprcc(sens_YA)
```




```r
# 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  0.99999628         d
## 2  0.35003489         c
## 3  0.23907533         b
## 4  0.07693689         a
## 5 -0.06934321         e
```

```r
# 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
```



Double number of parameters Sobol

```
# number of parameters

np=400
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 -4.629122e-05  3.911209e-07 1.476824e-05 -7.380379e-05 -1.377356e-05
## b  2.042926e-04  2.384177e-05 1.658430e-04 -1.827327e-04  5.384685e-04
## c  1.257443e-05  7.338154e-06 1.951051e-04 -4.601070e-04  3.676919e-04
## d  1.009775e+00  1.993244e-02 1.027651e-01  7.553131e-01  1.190468e+00
## e -1.442032e-06 -7.558292e-08 1.721376e-06 -4.836948e-06  2.141129e-06
```

```
# total sensitivity index
sens_YA_sobel$T
```

```
##        original          bias   std. error     min. c.i.     max. c.i.
## a  2.297090e-05 -1.297264e-06 9.347268e-06  7.107071e-06  4.272909e-05
## b -2.471483e-04  2.112393e-06 1.112914e-04 -4.976843e-04 -3.709156e-06
## c  2.146906e-04 -7.705261e-06 1.669163e-04 -9.670880e-05  5.849519e-04
## d  3.252571e-03 -8.447569e-03 5.768698e-02 -1.034217e-01  1.259233e-01
## e  5.538434e-07  6.139580e-08 1.248547e-06 -1.770524e-06  2.878567e-06
```

```
# 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  0.07693689
## b  0.23907533
## c  0.35003489
## d  0.99999628
## e -0.06934321
```

```
sens_YA_sobel$S
```

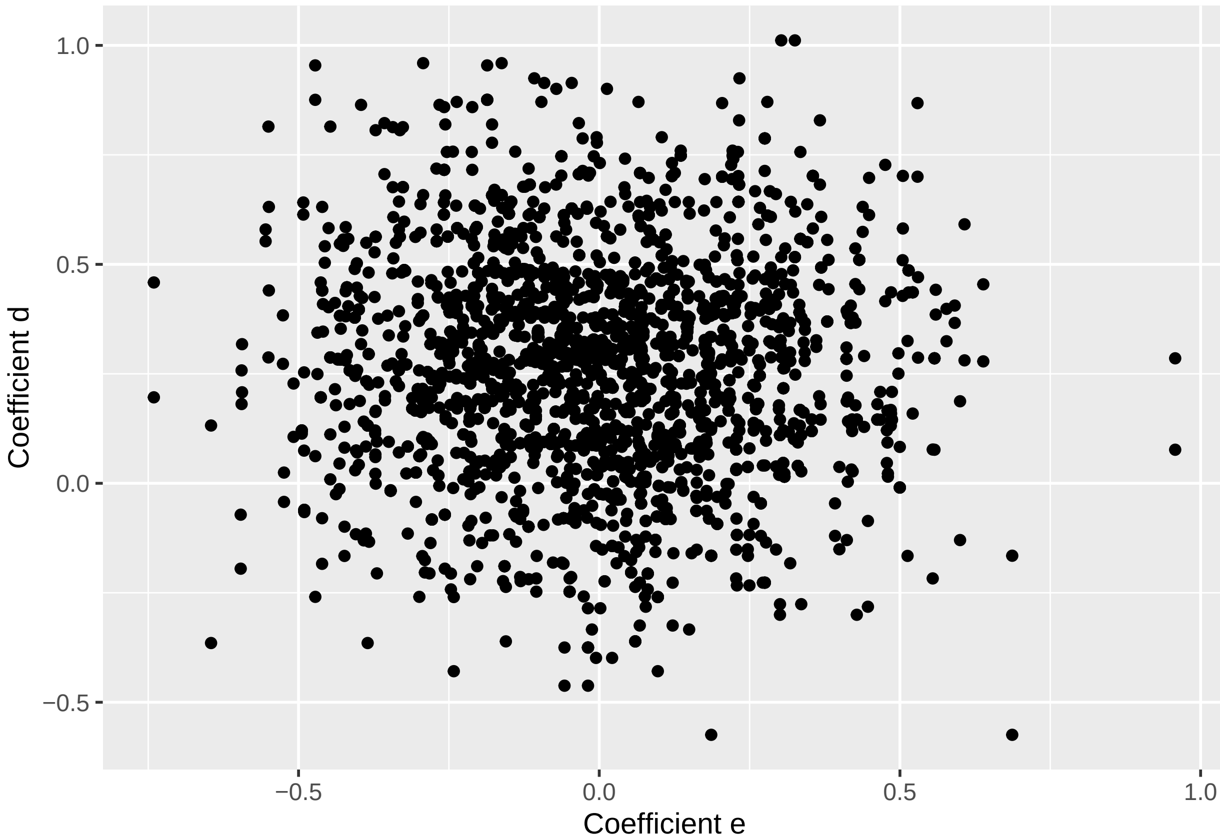

```
##        original          bias   std. error     min. c.i.     max. c.i.
## a -4.629122e-05  3.911209e-07 1.476824e-05 -7.380379e-05 -1.377356e-05
## b  2.042926e-04  2.384177e-05 1.658430e-04 -1.827327e-04  5.384685e-04
## c  1.257443e-05  7.338154e-06 1.951051e-04 -4.601070e-04  3.676919e-04
## d  1.009775e+00  1.993244e-02 1.027651e-01  7.553131e-01  1.190468e+00
## e -1.442032e-06 -7.558292e-08 1.721376e-06 -4.836948e-06  2.141129e-06
```


```
sens_YA_sobel$T
```

```
##        original          bias   std. error     min. c.i.     max. c.i.
```

```
## a  2.297090e-05 -1.297264e-06 9.347268e-06  7.107071e-06  4.272909e-05
## b -2.471483e-04  2.112393e-06 1.112914e-04 -4.976843e-04 -3.709156e-06
## c  2.146906e-04 -7.705261e-06 1.669163e-04 -9.670880e-05  5.849519e-04
## d  3.252571e-03 -8.447569e-03 5.768698e-02 -1.034217e-01  1.259233e-01
## e  5.538434e-07  6.139580e-08 1.248547e-06 -1.770524e-06  2.878567e-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")
```



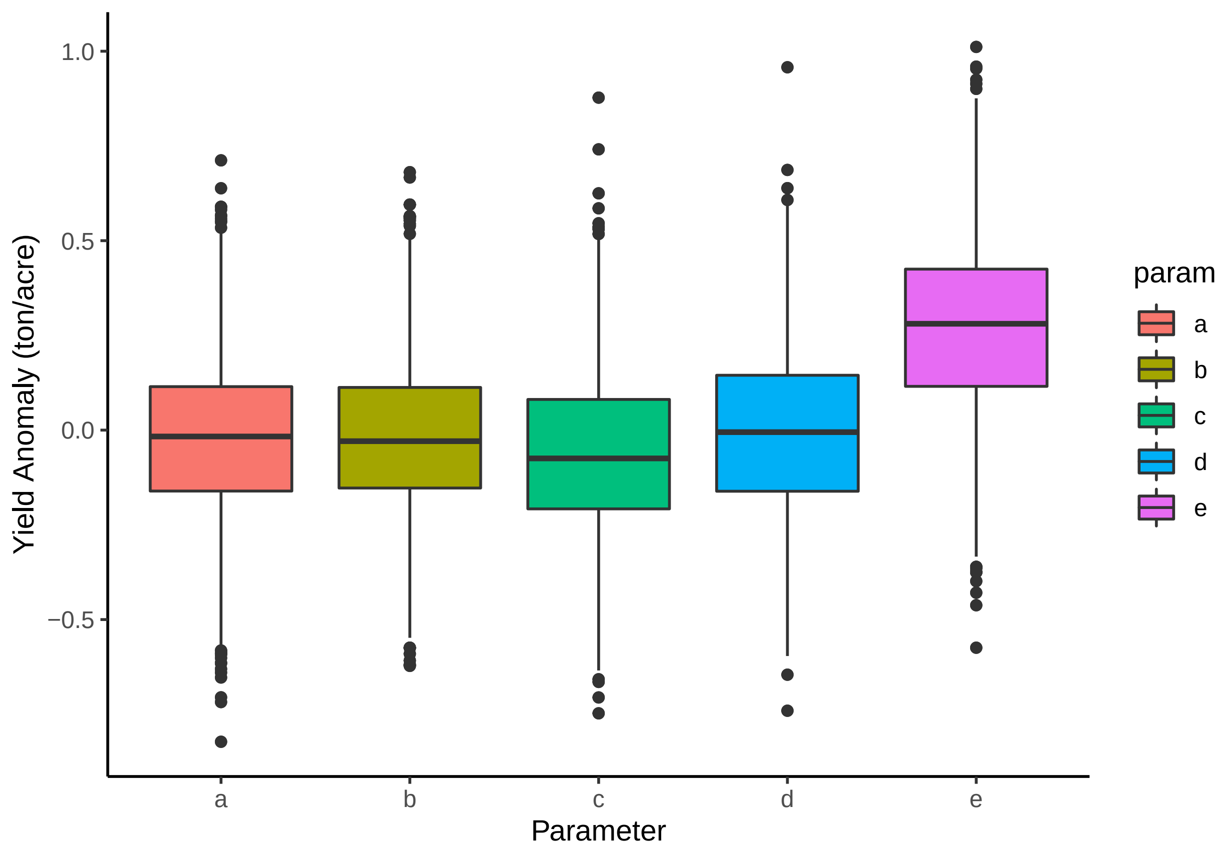

```
 # rank S
 rank_S <- sens_YA_sobel$S %>%
   select(original) %>%
   mutate(param = c("a", "b", "c", "d", "e")) %>%
  arrange(-(abs(original)))

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

```
##            PRCC Parameter
## 1  1.009775e+00         d
## 2  2.042926e-04         b
## 3 -4.629122e-05         a
## 4  1.257443e-05         c
## 5 -1.442032e-06         e
```

```r
 # 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
```

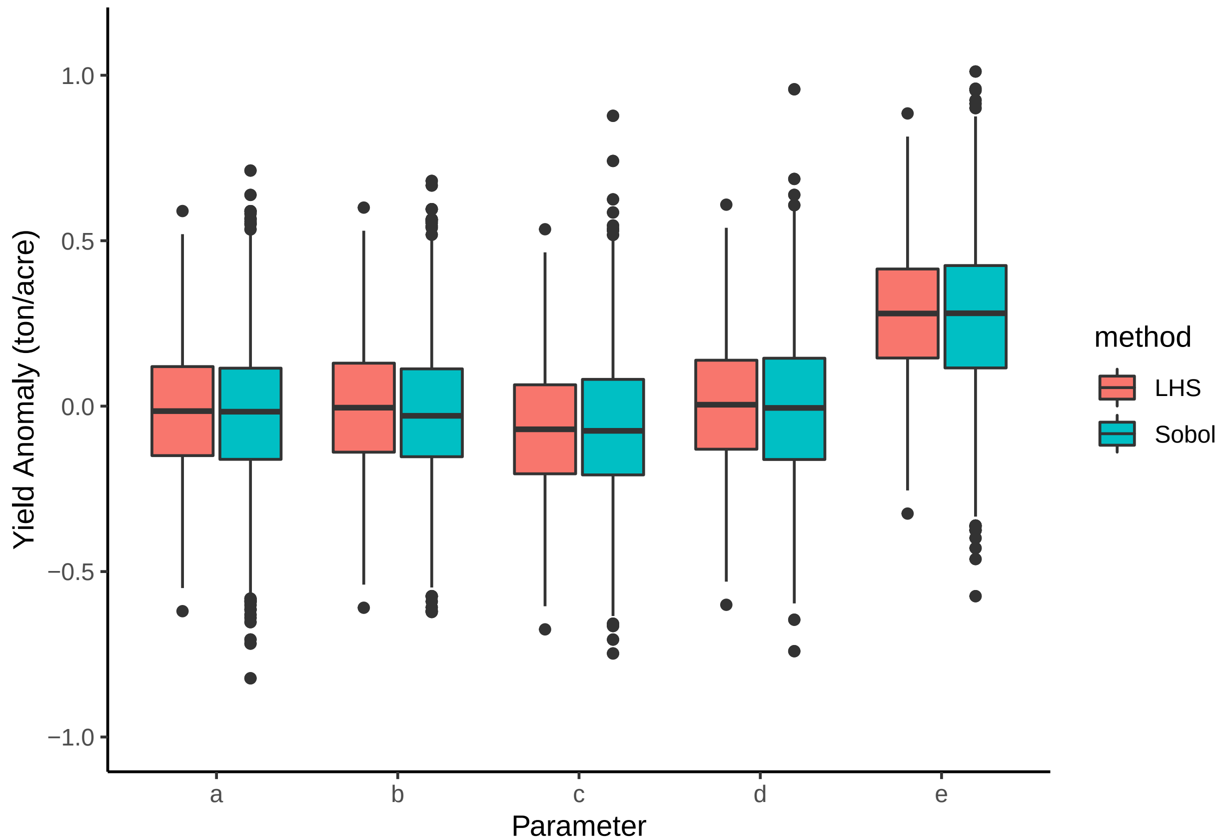
```
##              PRCC Parameter
## 1  3.252571e-03         d
## 2 -2.471483e-04         b
## 3  2.146906e-04         c
## 4  2.297090e-05         a
## 5  5.538434e-07         e
```

```r
 # 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
```



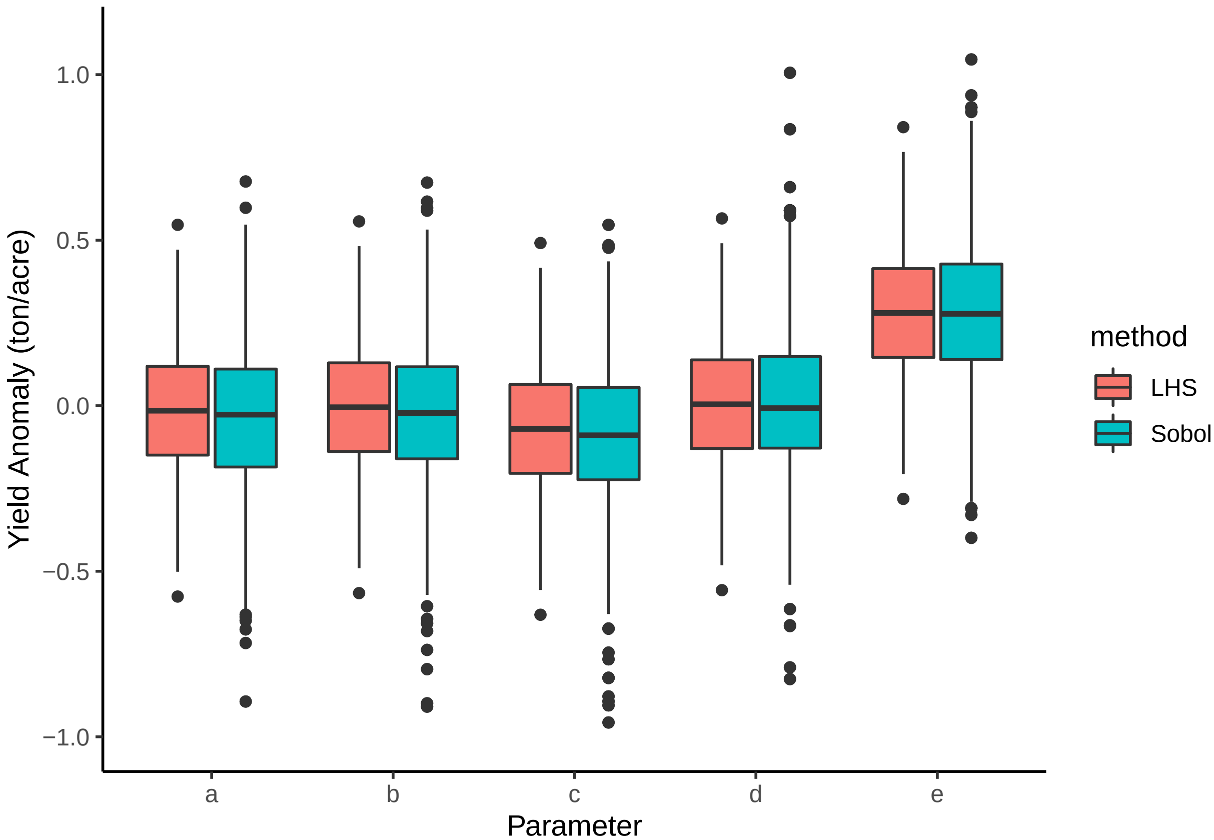
Comparing boxplots

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

# boxplot, comparing uncertainty of estimates
both_boxplot2 <- ggplot(sens_both_df, aes(x=param, y=YA, fill=method)) +
    geom_boxplot() +
  labs(y="Yield Anomaly (ton/acre)", x = "Parameter") +
  ylim(-1, 1.1) +
  theme_classic()

both_boxplot2
```




```
both_boxplot
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```



There was no significant difference in model sensitivities between the LHS and Sobol 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 number of parameters used in the sensitivity analysis did not significantly change the uncertainty in mean yield annomaly across the different parameters