figure.s3

wang xiaoqian

2025-05-20

```r
#Figure S3

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(reshape2)
```

```
##
##    'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(data.table)
```

```
##
##    'data.table'
##
## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
```

```
##     transpose
```

```r
library(metafor)
```

```
##      Matrix
##
##    'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##      metadat
##      numDeriv
##
## Loading the 'metafor' package (version 4.8-0). For an
## introduction to the package please type: help(metafor)
```

```r
library(metagear)
```

```
## ** metagear 0.7, for installing/troubleshooting help see:
## **     http://lajeunesse.myweb.usf.edu/metagear/metagear_basic_vignette.html
## ***** External dependencies check:
## ***** setup supports GUIs [ TRUE ]
## ***** setup supports data extraction from plots/figures [ FALSE ]
## *****       NOTE: EBImage package (Bioconductor) will be installed only
## *****             when a figure_* function is used.
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
# read data
d1 <- readxl::read_xlsx('D:/coursework/information/figure/Source Data.xlsx',sheet = "FigureS3")
d1 <- as.data.table(d1)

d2<-d1
CV_nuet_bar<-mean(d2$nuet_sd[is.na(d2$nuet_sd)==FALSE]/d2$nuet_mean[is.na(d2$nuet_sd)==FALSE])
d2$nuet_sd[is.na(d2$nuet_sd)==TRUE]<-d2$nuet_mean[is.na(d2$nuet_sd)==TRUE]*1.25*CV_nuet_bar

CV_nuec_bar<-mean(d2$nuec_sd[is.na(d2$nuec_sd)==FALSE]/d2$nuec_mean[is.na(d2$nuec_sd)==FALSE])
d2$nuec_sd[is.na(d2$nuec_sd)==TRUE]<-d2$nuec_mean[is.na(d2$nuec_sd)==TRUE]*1.25*CV_nuec_bar

# clean up column names
d2 <- as.data.table(d2)
setnames(d2,gsub('\\/','_',gsub(' |\\(|\\)','',colnames(d2))))
setnames(d2,tolower(colnames(d2)))

#Supplement and update the missing values for n_dose and p_dose
d2[is.na(n_dose), n_dose := median(d2$n_dose,na.rm=TRUE)]

# update the database (g_crop_type)
d2[g_crop_type=='maize', g_crop_type := 1]
d2[g_crop_type=='wheat', g_crop_type := 2]
d2[g_crop_type=='rice', g_crop_type := 3]
```

```r
#Conversion of factored data to numeric data
d2$g_crop_type <- as.numeric(d2$g_crop_type)
str(d2)
```

```
## Classes 'data.table' and 'data.frame':   2436 obs. of  15 variables:
##  $ studyid    : num  1 2 2 2 2 3 3 4 4 4 ...
##  $ mat        : num  14.2 14.2 14.2 14.2 14.2 ...
##  $ map        : num  495 495 495 495 495 ...
##  $ clay       : num  26.7 26.7 26.7 26.7 26.7 ...
##  $ soc        : num  11.5 11.5 11.5 11.5 11.5 ...
##  $ ph         : num  7.79 7.79 7.79 7.79 7.79 ...
##  $ g_crop_type: num  2 1 1 1 1 1 1 1 1 1 ...
##  $ management : chr  "EE" "CF" "CF" "RES" ...
##  $ n_dose     : num  120 150 150 150 150 337 349 150 150 150 ...
##  $ replication: num  2 3 3 3 3 4 4 3 3 3 ...
##  $ nue_type   : chr  "REN" "REN" "REN" "REN" ...
##  $ nuet_mean  : num  78.3 35.7 25.3 35.7 43.9 ...
##  $ nuet_sd    : num  10.17 4.64 3.29 4.64 5.7 ...
##  $ nuec_mean  : num  72.5 30.6 30.6 30.6 30.6 ...
##  $ nuec_sd    : num  9.89 4.17 4.17 4.17 4.17 ...
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "index")= int(0)
```

```r
# update the database (tillage)
d2[management=='ROT', management := 7]
d2[management=='CC', management := 8]
d2[management=='RES', management := 9]
d2[management=='RFR', management := 10]
d2[management=='RFP', management := 11]
d2[management=='RFT', management := 12]
d2[management=='CF', management := 13]
d2[management=='OF', management := 14]
d2[management=='RT', management := 15]
d2[management=='NT', management := 16]
d2[management=='EE', management := 17]

#Conversion of factored data to numeric data
d2$management <- as.numeric(d2$management)

# Estimate meta-analytical response measure (ROM Method)

# calculate effect size (NUE)
es21 <- escalc(measure = "ROM", data = d2,
               m1i = nuet_mean, sd1i = nuet_sd, n1i = replication,
               m2i = nuec_mean, sd2i = nuec_sd, n2i = replication )

# make forest plots per group treatments

# convert to data.tables
d02 <- as.data.table(es21)

d3 <- d02[,c("mat","map", "clay", "soc", "ph", "n_dose", "g_crop_type")] #Extraction correlation column

#Standardize data to prevent large gaps in values
```

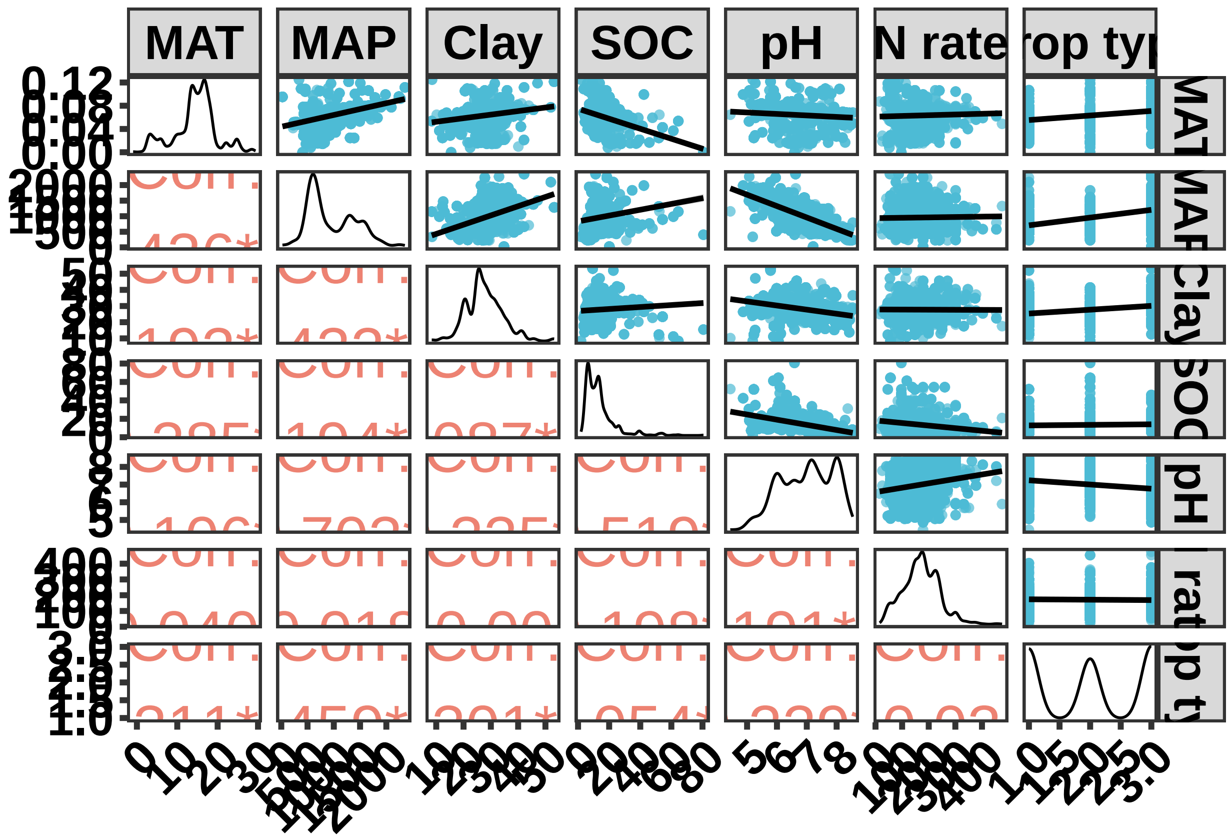
```
df=scale(d3[,1:7],center=TRUE,scale=TRUE)

#Change column name
names(d3)<-c("MAT","MAP", "Clay", "SOC", "pH", "N rate", "Crop type")
head(d3)
```

```
##          MAT       MAP     Clay      SOC       pH N rate Crop type
##        <num>     <num>    <num>    <num>    <num>  <num>     <num>
## 1: 14.23171  494.9222 26.72143 11.50714 7.792857    120         2
## 2: 14.23171  494.9222 26.72143 11.50714 7.792857    150         1
## 3: 14.23171  494.9222 26.72143 11.50714 7.792857    150         1
## 4: 14.23171  494.9222 26.72143 11.50714 7.792857    150         1
## 5: 14.23171  494.9222 26.72143 11.50714 7.792857    150         1
## 6: 17.87691 1281.7686 34.41429  9.05000 5.700000    337         1
```

```
str(d3)
```

```
## Classes 'data.table' and 'data.frame':   2436 obs. of  7 variables:
##  $ MAT      : num  14.2 14.2 14.2 14.2 14.2 ...
##  $ MAP      : num  495 495 495 495 495 ...
##  $ Clay     : num  26.7 26.7 26.7 26.7 26.7 ...
##  $ SOC      : num  11.5 11.5 11.5 11.5 11.5 ...
##  $ pH       : num  7.79 7.79 7.79 7.79 7.79 ...
##  $ N rate   : num  120 150 150 150 150 337 349 150 150 150 ...
##  $ Crop type: num  2 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "digits")= Named num [1:9] 4 4 4 4 4 4 4 4 4
##   ..- attr(*, "names")= chr [1:9] "est" "se" "test" "pval" ...
##  - attr(*, "yi.names")= chr "yi"
##  - attr(*, "vi.names")= chr "vi"
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
#Calculate correlation coefficients and scatter plots and fit linearity
ggpairs(d3, lower = list(continuous = wrap("cor", size = 8,color="#E64B35B2")),
        upper = list(continuous = wrap("smooth", size =1.2,color="#4DBBD5B2")))+
  theme_bw(base_line_size = 1.05,base_rect_size = 1.05)+
  theme(panel.grid.major=element_blank(),panel.grid.minor=element_blank())+
  theme(axis.text=element_text(colour='black',size=18, face="bold"), strip.text = element_text(color="bl
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```



```
ggsave(file = "D:/coursework/information/figure/picture/Figure_S3.png",width = 410,height = 297, units =
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

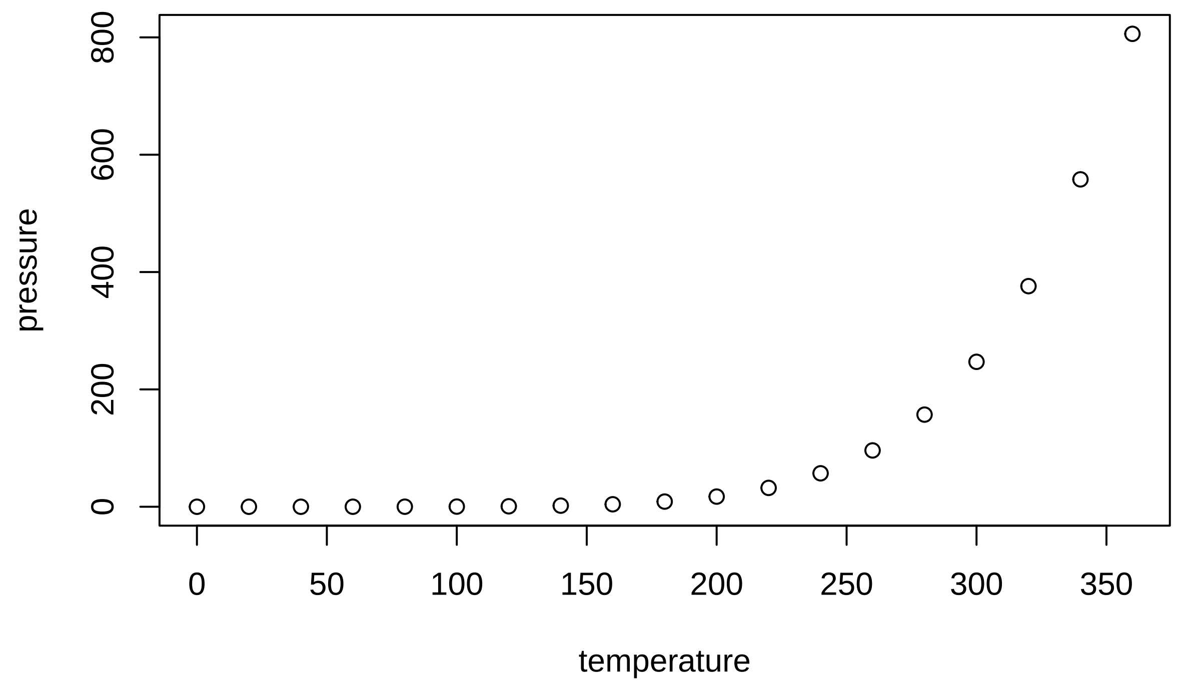
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.