R-Code Tutorial: Revisiting linear regression to test agreement in continuous predicted-observed datasets

Adrian A. Correndo, Trevor Hefley, Dean P. Holzworth, and Ignacio A. Ciampitti

Contents

Libraries needed. 2
Error Metrics Definitions. 3
Decomposition metrics using SMA. 4
Illustrative dataset. Comparing Regressions. 6
APSIM Datasets. 15
APSIM examples. 16
Error Metrics. 16
A. Wheat Grain N 18
B. Barley Grain No. 20
C. Sorghum Grain No. 22
D. Chickpea Dry-mass. 24
REFERENCES 26

*Corresponding: e-mail: correndo@ksu.edu - ciampitti@ksu.edu.

Libraries needed.

```
#install.packages("easypackages") # To install packages
library(easypackages)
packages("tidyverse", "data.table", "fs", "readxl") # Data wrangling
packages("ggplot2", "ggpubr", "ggformula", "EnvStats",
         "ggExtra", "ggpmisc","ggthemes","magrittr",
         "hrbrthemes", "ggimage", "cowplot") # Figures
packages("smatr") # Regression Models
packages("kableExtra") # Tables
```

Error Metrics Definitions.

The following lines of code define functions for the calculation of error metrics described in Table 5, Correndo et al. (2021).

```
# Variance (uncorrected, for MSE and its decomposition)
SSx <- function(x){sum((x - mean(x))^2)/length(x)}

# Standard Deviation (uncorrected, for MSE and its decomposition)
sdev <- function(x){sqrt(SSx(x))}

# Mean Bias Error (Predicted compared to Observed)
MBErr <- function(Obs,Pre){(mean(Pre)-mean(Obs))}

# Mean Square Error
MSE <- function(Obs,Pre){sum((Obs-Pre)^2)/length(Obs)}

# Root Mean Square Prediction Error
RMSE <- function(Obs,Pre){sqrt(sum((Obs-Pre)^2)/length(Obs))}

# Accuracy component of CCC
Xa <- function(Obs,Pre){(2 / (sdev(Pre)/sdev(Obs) +
                                sdev(Obs)/sdev(Pre) +
                                (MBErr(Obs,Pre)^2/(sdev(Pre)*sdev(Obs)))) )}

# Concordance Correlation (Lin,1989), equivalent to r*Xa
CCC <- function(Obs, Pre) {cor(Obs,Pre) * Xa(Obs,Pre)}
```

Decomposition metrics using SMA.

The following lines of code cover the functions necessary to perform calculations presented in Section 2.5 (Error decomposition using linear regression), Correndo et al. (2021).

```
# Mean Lack of Accuracy (Systematic difference)
MLA <- function(Obs, Pre){sum ((
  Pre - ( (mean(Obs) - (sdev(Obs)/sdev(Pre)*mean(Pre))) +
            sdev(Obs)/sdev(Pre) * Pre))^2) / length(Obs) }

# Mean Lack of Precision (Unsystematic difference)
MLP <- function(Obs, Pre){
  sum (abs(Obs - ((mean(Obs) - (sdev(Obs)/sdev(Pre)*mean(Pre))) +
                    sdev(Obs)/sdev(Pre) * Pre)) *
         abs(Pre - ((mean(Pre) - (sdev(Pre)/sdev(Obs)*mean(Obs))) +
                      sdev(Pre)/sdev(Obs) * Obs) ) ) / length(Obs) }

# Kobayashi and Salam (2000) MSE segregation, also in Gupta et al. (2009)
# LCS = MLP; SB + SDSD = MLA
# However, authors consider LCS + SDSD as unsystematic (Mean Square Variation)
# We argue that SDSD is part of the "systematic" component
# Table 3, MLA = SB + SDSD, Correndo et al., (2021)

# Lack of correlation
LCS <- function(Obs, Pre){2 * sdev(Pre) * sdev(Obs) * (1 - cor(Obs,Pre))}
# Square difference between standard deviations
SDSD <- function(Obs, Pre){(sdev(Pre) - sdev(Obs))^2}
# Square bias
SB <- function(Obs, Pre){(mean(Obs) - mean(Pre))^2}

# Equivalence with Kobayashi and Salam (2000)
# MLA as the sum of SDSD + SB
MLA2 <- function(Obs, Pre){SDSD(Obs,Pre) + SB(Obs,Pre)}
# MLP as LCS
MLP2 <- function(Obs, Pre){ LCS(Obs,Pre) }

# Theil's partial inequalities, Smith & Rose (1995)
# Proportion of TSS related to mean bias (additive bias, part of PLA)
Ub <- function(Obs,Pre){100*
    (length(Obs)*(mean(Obs)-mean(Pre))^2) / sum((Obs-Pre)^2) }

# Lack of Consistency. Proportion of TSS related to proportional bias (PLA)
Uc <- function(Obs,Pre){100*
    (length(Obs)*(sdev(Obs)-sdev(Pre))^2) / sum((Obs-Pre)^2) }

# Proportion of TSS related to random error (= PLP)
Ue <- function(Obs,Pre){100*
    ( 2*length(Obs)* (1-cor(Obs,Pre))*sdev(Pre)*sdev(Obs) )  / sum((Obs-Pre)^2)}

# Equivalence with Smith & Rose (1995)
PLA <- function(Obs,Pre) {Ub(Obs,Pre) + Uc(Obs,Pre)}
PLP <- function(Obs,Pre) {Ue(Obs,Pre)}

# Percentage additive bias (contribution to MLA)
```

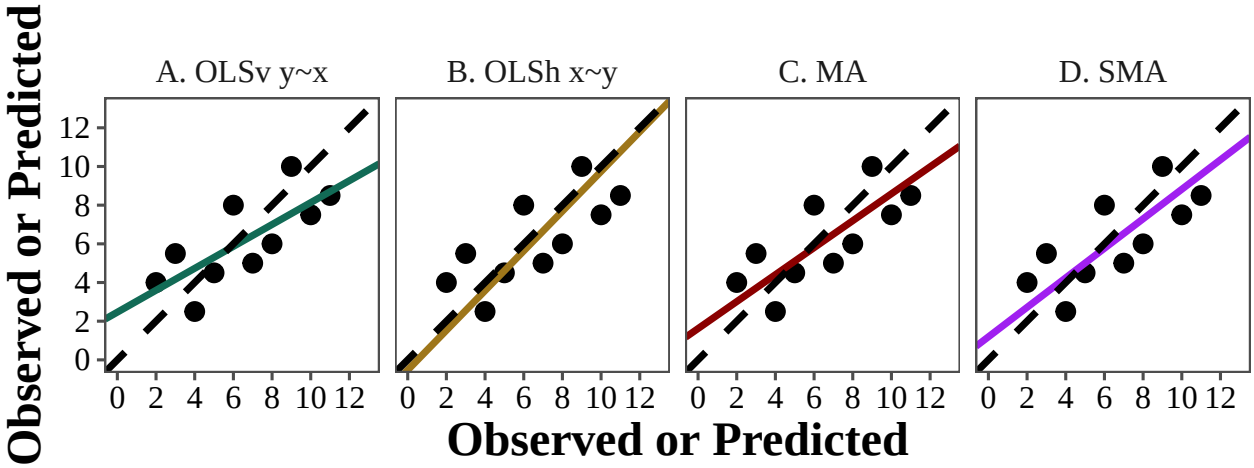
```r
PAB <- function(Obs,Pre) {100 * (SB(Obs,Pre)) / MSE(Obs,Pre)  }
# Percentage proportional bias (contribution to MLA)
PPB <- function(Obs,Pre) {100 * (SDSD(Obs,Pre)) / MSE(Obs,Pre)  }
```

Illustrative dataset. Comparing Regressions.

Simulated dataset provided in Figure 2, Correndo et al. (2021).

```
# Define example dataset
data = data.frame(x = c(2,3,4,5,6,7,8,9,10,11),
                  y = c(4,5.5,2.5,4.5,8,5,6,10,7.5,8.5))

#OLSv yx
# Extracting coeff. using LM
YX <- lm(y~x, data=data)
YX.slope <- YX %>% coef %>% .[[2]]
YX.intercept <- YX %>% coef %>% .[[1]]
# Manual alternative
#OLSv_slope = Cov(data$x,data$y)/SSx(data$x) # OLSv slope
#OLSv_int = mean(data$y) - (OLSv_slope*mean(data$x)) # OLSv Intercept

#OLSh xy
# Extracting coeff. using LM
XY <- lm(x~y, data=data)
XY.slope <- (1/XY$coef[2])
XY.intercept <- -(XY$coef[1]/XY$coef[2])
# Manual alternative
#OLSh_slope = SSx(data$y)/Cov(data$x,data$y) # OLSh slope
#OLSh_int = mean(data$y) - (OLSh_slope*mean(data$x)) # OLSh Intercept

# Major axis
# Extracting coeff. using smatr package (Warton et al., 2012)
MA = sma(y~x, method = "MA", data = data)
MA.slope <- MA %>% coef %>% .[[2]]
MA.intercept <- MA %>% coef %>% .[[1]]

# Standardized Major axis
# Extracting coeff. using smatr
SMA = sma(y~x, method = "SMA", data = data)
SMA.slope <- SMA %>% coef %>% .[[2]]
SMA.intercept <- SMA %>% coef %>% .[[1]]

data = data %>%
  mutate(#add the slopes/intercepts to the data frame:
         slope_YX=YX.slope,
         slope_XY=XY.slope,
         slope_MA=MA.slope,
         slope_SMA=SMA.slope,
         intercept_YX=YX.intercept,
         intercept_XY=XY.intercept,
         intercept_MA=MA.intercept,
         intercept_SMA=SMA.intercept,
         )%>%
  #transpose to a long form
  gather(key="key",value="value",-c(x,y)) %>%
  # have "yhat_YX", want two colums of "yhat" "YX"
  separate(key,c("type","line"),"_") %>%
  #then transpose to be fatter, so we have cols for xhat, yhat etc
```

```
  spread(key="type",value="value") %>%
  #relable the lines with more descriptive names, and reorder for plotting:
  mutate(line=case_when(
           line=="YX" ~ "A. OLSv y~x",
           line=="XY" ~ "B. OLSh x~y",
           line=="MA" ~ "C. MA",
           line=="SMA" ~ "D. SMA"),
         line=factor(line,levels=c("A. OLSv y~x","B. OLSh x~y","C. MA","D. SMA"))) %>%
  arrange(., line)

# Plot
data %>% filter(line %in% c("A. OLSv y~x","B. OLSh x~y","C. MA","D. SMA")) %>%
  ggplot() +
  geom_point(aes(x=x,y=y), color="black", size=3,show.legend=F)+
  geom_abline(aes(slope=slope,intercept=intercept,colour=line),size=1.25,show.legend=F)+
  geom_abline(linetype = "dashed", size = 1.25)+
  scale_colour_manual("", values = c("#126b56", "#9e761a","dark red", "purple"))+
  scale_x_continuous(limits = c(0,12.9), breaks = seq(0,12,by=2))+
  scale_y_continuous(limits = c(0,12.9), breaks = seq(0,12,by=2))+
  labs(x = "Predicted or Observed", y = "Predicted or Observed")+
  facet_wrap(~line, nrow = 1)+
  theme_few()+
  theme(axis.text.y = element_text(color="black", size = rel(1.25), family = "serif"),
        axis.text.x = element_text(color="black", size = rel(1.25), family = "serif"),
        axis.title = element_text(color = "black", size = rel(1.5),
                                  family = "serif",face = "bold",
                                  margin(t = 0, r = -2, b = 0, l = 0, unit = "inches")),
        strip.text = element_text(size = rel(1), family = "serif"),
        strip.background = element_blank(),
    plot.margin=unit(c(1,1,1,1), "pt"),
        aspect.ratio = 1)+
  labs(x = "Observed or Predicted", y = "Observed or Predicted")
```


A. OLSv y~x
B. OLSh x~y
C. MA
D. SMA
Observed or Predicted
Observed or Predicted
0 2 4 6 8 10 12

Error Decomposition with OLS, MA and SMA.

The following lines of code correspond to the error decompositions proposed in Section 2.5, and in Supplementary material of Correndo et al. (2021). The decompositions include the proposed by Willmott et al.(1981) using OLS, by Duveiller et al.(2006) using MA, and adapted from Ji and Gallo (2006) using SMA linear regressions.

```
# Example 1, Figure 2, Correndo et al. (2021) -Agr. Syst.-
df = data.frame(O_i = c(2,3,4,5,6,7,8,9,10,11),
                P_i = c(4,5.5,2.5,4.5,8,5,6,10,7.5,8.5),
                sim = rep(1)) %>% nest(data=c("O_i", "P_i"))

# Optional, Example 2, Table 9.1, Wallach et al. (2019)
# Remove the "#" to use this dataset
#df = data.frame(O_i = c(78,110,92,75,110,108,113,155,150),
#                P_i = c(126,126,126,105,105,105,147,147,147),
#                 sim = rep(1)) %>% nest(data=c("O_i", "P_i"))


# Estimate linear models and Sum of Unsyst and Syst differences
example = df %>%
  mutate(
    #MODELS
    model_ols = data %>% map(~lm(P_i~O_i, data=df %>% unnest())),
    model_ma = data %>% map(~sma(P_i~O_i,data=., method = "MA")),
    model_sma = data %>% map(~sma(P_i~O_i,data=.)),
    #Fitted P and O
    #Fitted P values as function of O using OLS,
    # To calculate MSEu and MSEs proposed by Willmott (1981)
    P_ols = map2(data, model_ols,
                 ~ .y %>% coef %>% .[[1]] + .y %>% coef %>% .[[2]] * .x$O_i),
    #Fitted P values as function of O using MA
    # To calculate 2h^2 proposed by Duveiller et al (2016)
    P_ma = map2(data, model_ma,
                 ~ .y %>% coef %>% .[[1]] + .y %>% coef %>% .[[2]] * .x$O_i),
    #Fitted P values as function of O using SMA
    # To calculate Unsyst and Syst proposed by Ji & Gallo (2006)
    # Equivalent to Kobayashi et al 2000 and Smith & Rose (1995)
    P_sma = map2(data, model_sma,
                 ~ .y %>% coef %>% .[[1]] + .y %>% coef %>% .[[2]] * .x$O_i),
    #Fitted O values as function of P using SMA, Ji & Gallo (2006)
    O_sma = map2(data, model_sma,
    # SMA slope_OP = 1/slope_PO
    ~ ((1/(.y %>% coef %>% .[[2]]))*.x$P_i) +
      (mean(.x$O_i)-((1/(.y %>% coef %>% .[[2]]))*
                       mean(.x$P_i) ))),
    # MA residuals (hi) (Table 1, Eq. (6))
    h_ma = map2(data, model_ma,
                ~sqrt((.x$P_i - (.y %>% coef %>% .[[1]] +
                        .y %>% coef %>%
                          .[[2]] * .x$O_i))^2 /
                          ((.y %>% coef %>% .[[2]])^2 + 1) ),
        ))
```

```r
# Estimating residuals from OLS, MA and SMA lines
example = example %>%
  # Remove models and keep only data of interest
  dplyr::select(-model_sma, -model_ma, -model_ols) %>%
  unnest(cols=c("data", "P_sma", "O_sma", "P_ols", "P_ma","h_ma")) %>%
  mutate(n = length(O_i),
         Sq_diff = (O_i-P_i)^2, # Square difference bw P_i & O_i
         O_45 = P_i, # Observed 45 line = Predicted
         P_45 = O_i, # Predicted 45 line = Observed
         Oi_45 = abs(O_i - O_45), # Diff. actual O (i) vs. 1:1-line
         Pi_45 = abs(P_i - P_45), # Diff. actual P (i) vs. 1:1-line
         UD_ols = (P_ols-P_i)^2, # Unsyst. Diff. OLS - Willmott (1981)
         SD_ols = (P_ols-O_i)^2, # Syst.Diff. OLS - Willmott (1981)
         Psma_i = abs(P_sma - P_i), # Diff. sma-line P (sma) vs. actual P (i)
         Osma_i = abs(O_sma - O_i), # Diff. sma-line O (sma) vs. actual O (i)
         P45_sma = abs(P_45 - P_sma), # Diff. Pi=O (1:1-line) vs. sma P (i)
         O45_sma = abs(O_45 - O_sma), # Diff. Oi=P (1:1-line) vs. sma O (i)
         PD = Oi_45 * Pi_45, # Product of Diff., O (i-45) by P (i-45)
         UD_ma = 2*h_ma^2, # Unsyst.Diff. MA - Adapted Duveiller et al (2016)
         UD_ma2 = ((2*h_ma)^2)/2, # Adapted Duveiller et al (2016)
         SD_ma = (P_ma-O_i)^2, # Systematic Diff. MA
         UD_sma = Osma_i * Psma_i, # Unsyst. Diff., SMA - Ji & Gallo (2006)
         SD_sma = P45_sma^2) %>% # Syst. Diff., SMA
  #group_by(sim) %>%
  mutate(#METRICS DERIVED FROM REGRESSION LINES
         TSS = sum(Sq_diff), # Total sum of squares REFERENCE
         SUD_ols = sum(UD_ols),# Sum of unsyst. diff. from OLS
         SSD_ols = sum(SD_ols), # Sum of syst. diff. from OLS
         TSS_ols = SUD_ols + SSD_ols, # *TSS as SUD_ols + SSD_ols
         SUD_ma = sum(UD_ma),# Sum of unsyst. diff. from MA
         SSD_ma = sum(SD_ma), # Sum of syst. diff. from MA
         TSS_ma = SUD_ma + SSD_ma, # *TSS as SUD_ma + SSD_ma
         SUD_sma = sum(UD_sma), # Sum of unsyst. diff. from SMA
         SSD_sma = sum(SD_sma), # Sum of syst. diff. from SMA
         TSS_sma = SUD_sma + SSD_sma, # TSS as SUD_sma + SSD_sma
         `SUD_%` = 100*SUD_sma/TSS,# Contribution of SUD to TSS = PLP
         `SSD_%` = 100*SSD_sma/TSS,# Contribution of SSD to TSS = PLA
         MSE_tss = TSS/length(O_i), # MSE as TSS/n
         MSEu = SUD_ols/length(O_i),
         MSEs = SSD_ols/length(O_i),
         MSE_willmott = MSEu + MSEs,
         MLP_sma = SUD_sma/n, # Mean Lack of Precision from SMA (SUD_sma/n)
         MLA_sma = SSD_sma/n, # Mean Lack of Accuracy from SMA (SUD_sma/n)
         MSE_sma = MLP_sma + MLA_sma, # MSE from SMA
         # METRICS FROM FUNCTIONS
         cor = cor(P_i,O_i), # Pearson correlation
         CCC = CCC(O_i, P_i), # Concordance correlation
         #Option from a package that also estimates ConfIntervals
         #CCC = CCC(O_i, P_i, ci = "z-transform", conf.level = 0.95)[[1]]$est,
         bias2 = MBErr(O_i,P_i)^2,
         MSE = MSE(P_i,O_i), # MSE from traditional formula
         MLP = MLP(O_i,P_i), # MLP proposed from formula
         MLA = MLA(O_i,P_i), # MLA proposed from formula
```

```
          RMLP = sqrt(MLP), # Root MLP
          RMLA = sqrt(MLA), # Root MLA
          PLP = 100*(MLP/MSE), # Percentage Lack of Precision
          PLA = 100*(MLA/MSE), # Percentage Lack of Accuracy
          PAB = PAB(O_i,P_i), # Percentage additive bias (to MSE)
          PPB = PPB(O_i,P_i) # Percentage proportional bias (to MSE)
          )

# Individual differences Willmott (1981)
ex.points.OLS = example %>%
  dplyr::select(O_i, P_i, P_ols, UD_ols, SD_ols)
kable(ex.points.OLS) %>%
  kable_styling(latex_options =c("striped"))
```

O_i	P_i	P_ols	UD_ols	SD_ols
2	4.0	3.600000	0.1600000	2.5600000
3	5.5	4.166667	1.7777778	1.3611111
4	2.5	4.733333	4.9877778	0.5377778
5	4.5	5.300000	0.6400000	0.0900000
6	8.0	5.866667	4.5511111	0.0177778
7	5.0	6.433333	2.0544444	0.3211111
8	6.0	7.000000	1.0000000	1.0000000
9	10.0	7.566667	5.9211111	2.0544444
10	7.5	8.133333	0.4011111	3.4844444
11	8.5	8.700000	0.0400000	5.2900000

```
# Individual differences Duveiller et al (2016)
ex.points.MA = example %>%
  dplyr::select(O_i, P_i, P_ma, h_ma, UD_ma, SD_ma)
kable(ex.points.MA) %>%
  kable_styling(latex_options =c("striped"))
```

O_i	P_i	P_ma	h_ma	UD_ma	SD_ma
2	4.0	3.013078	0.8096222	1.3109762	1.0263262
3	5.5	3.710171	1.4682865	4.3117307	0.5043435
4	2.5	4.407265	1.5646258	4.8961080	0.1658650
5	4.5	5.104359	0.4957863	0.4916081	0.0108908
6	8.0	5.801453	1.8035788	6.5057931	0.0394209
7	5.0	6.498547	1.2293335	3.0225219	0.2514552
8	6.0	7.195641	0.9808444	1.9241114	0.6469937
9	10.0	7.892735	1.7286959	5.9767790	1.2260365
10	7.5	8.589829	0.8940413	1.5986196	1.9885835
11	8.5	9.286922	0.6455521	0.8334750	2.9346348

```
# Individual differences Willmott (1981)
ex.points.SMA = example %>%
  dplyr::select(O_i, P_i, Psma_i, Osma_i, UD_sma, SD_sma)
kable(ex.points.SMA) %>%
  kable_styling(latex_options =c("striped"))
```

O_i	P_i	Psma_i	Osma_i	UD_sma	SD_sma
2	4.0	1.283359	1.6820600	2.158687	0.5135742
3	5.5	2.020390	2.6480646	5.350124	0.2300254
4	2.5	1.742578	2.2839447	3.979953	0.0588442
5	4.5	0.505547	0.6626051	0.334978	0.0000308
6	8.0	2.231484	2.9247391	6.526510	0.0535850
7	5.0	1.531484	2.0072703	3.074103	0.2195069
8	6.0	1.294453	1.6966005	2.196170	0.4977965
9	10.0	1.942578	2.5460787	4.945957	0.8884538
10	7.5	1.320390	1.7305958	2.285062	1.3914788
11	8.5	1.083359	1.4199260	1.538290	2.0068715

```
# Metrics summary using data points and line
# 1. MSEs + MSEu (Willmott, 1981) do not match MSE
ex.summary.line.OLS = example %>%
  dplyr::group_by(sim) %>% slice_head() %>% ungroup() %>%
  dplyr::select(TSS,SUD_ols,SSD_ols,TSS_ols,
                MSE_tss,MSEu,MSEs,MSE_willmott)
kable(ex.summary.line.OLS) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

TSS	SUD_ols	SSD_ols	TSS_ols	MSE_tss	MSEu	MSEs	MSE_willmott
38.25	21.53333	16.71667	38.25	3.825	2.153333	1.671667	3.825

```
# 2. SUD_ma (Duveiller, 2016) is NOT additive to the TSS (nor to MSE)
ex.summary.line.MA = example %>%
  dplyr::group_by(sim) %>% slice_head() %>% ungroup() %>%
  dplyr::select(TSS,SUD_ma,SSD_ma,TSS_ma)
kable(ex.summary.line.MA) %>%
  kable_styling(latex_options =c("striped"))
```

TSS	SUD_ma	SSD_ma	TSS_ma
38.25	30.87172	8.79455	39.66627

```
# 3. MLP_sma + MLA_sma = MSE
ex.summary.line.SMA = example %>%
  dplyr::group_by(sim) %>% slice_head() %>% ungroup() %>%
  dplyr::select(TSS,SUD_sma,SSD_sma,TSS_sma,
                MSE_tss,MLP_sma,MLA_sma,MSE_sma)

# Print summary SMA line
kable(ex.summary.line.SMA) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

TSS	SUD_sma	SSD_sma	TSS_sma	MSE_tss	MLP_sma	MLA_sma	MSE_sma
38.25	32.38983	5.860167	38.25	3.825	3.238983	0.5860167	3.825

```r
# Print example summary
kable(example %>% dplyr::select(1:8,10:11)) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

sim	O_i	P_i	P_ols	P_ma	P_sma	O_sma	h_ma	Sq_diff	O_45
1	2	4.0	3.600000	3.013078	2.716641	3.682060	0.8096222	4.00	4.0
1	3	5.5	4.166667	3.710171	3.479610	5.648065	1.4682865	6.25	5.5
1	4	2.5	4.733333	4.407265	4.242578	1.716055	1.5646258	2.25	2.5
1	5	4.5	5.300000	5.104359	5.005547	4.337395	0.4957863	0.25	4.5
1	6	8.0	5.866667	5.801453	5.768516	8.924739	1.8035788	4.00	8.0
1	7	5.0	6.433333	6.498547	6.531484	4.992730	1.2293335	4.00	5.0
1	8	6.0	7.000000	7.195641	7.294453	6.303400	0.9808444	4.00	6.0
1	9	10.0	7.566667	7.892735	8.057422	11.546079	1.7286959	1.00	10.0
1	10	7.5	8.133333	8.589829	8.820390	8.269404	0.8940413	6.25	7.5
1	11	8.5	8.700000	9.286922	9.583359	9.580074	0.6455521	6.25	8.5

```r
kable(example %>% dplyr::select(12:26)) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

P_45	Oi_45	Pi_45	UD_ols	SD_ols	Psma_i	Osma_i	P45_sma	O45_sma	PD	UD_ma	UD_ma2	SD_ma	UD_sma	SD_sma
2	2.0	2.0	0.1600000	2.5600000	1.283359	1.6820600	0.7166409	0.3179400	4.00	1.3109762	1.3109762	1.0263262	2.158687	0.5135742
3	2.5	2.5	1.7777778	1.3611111	2.020390	2.6480646	0.4796096	0.1480646	6.25	4.3117307	4.3117307	0.5043435	5.350124	0.2300254
4	1.5	1.5	4.9877778	0.5377778	1.742578	2.2839447	0.2425783	0.7839447	2.25	4.8961080	4.8961080	0.1658650	3.979953	0.0588442
5	0.5	0.5	0.6400000	0.0900000	0.505547	0.6626051	0.0055470	0.1626051	0.25	0.4916081	0.4916081	0.0108908	0.334978	0.0000308
6	2.0	2.0	4.5511111	0.0177778	2.231484	2.9247391	0.2314843	0.9247391	4.00	6.5057931	6.5057931	0.0394209	6.526510	0.0535850
7	2.0	2.0	2.0544444	0.3211111	1.531484	2.0072703	0.4685157	0.0072703	4.00	3.0225219	3.0225219	0.2514552	3.074103	0.2195069
8	2.0	2.0	1.0000000	1.0000000	1.294453	1.6966005	0.7055470	0.3033995	4.00	1.9241114	1.9241114	0.6469937	2.196170	0.4977965
9	1.0	1.0	5.9211111	2.0544444	1.942578	2.5460787	0.9425783	1.5460787	1.00	5.9767790	5.9767790	1.2260365	4.945957	0.8884538
10	2.5	2.5	0.4011111	3.4844444	1.320390	1.7305958	1.1796096	0.7694042	6.25	1.5986196	1.5986196	1.9885835	2.285062	1.3914788
11	2.5	2.5	0.0400000	5.2900000	1.083359	1.4199260	1.4166409	1.0800740	6.25	0.8334750	0.8334750	2.9346348	1.538290	2.0068715

```r
kable(example %>% dplyr::select(27:36) %>% slice_head()) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

TSS	SUD_ols	SSD_ols	TSS_ols	SUD_ma	SSD_ma	TSS_ma	SUD_sma	SSD_sma	TSS_sma
38.25	21.53333	16.71667	38.25	30.87172	8.79455	39.66627	32.38983	5.860167	38.25

```r
kable(example %>% dplyr::select(37:45) %>% slice_head()) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

SUD_%	SSD_%	MSE_tss	MSEu	MSEs	MSE_willmott	MLP_sma	MLA_sma	MSE_sma
84.6793	15.3207	3.825	2.153333	1.671667	3.825	3.238983	0.5860167	3.825

```r
kable(example %>% dplyr::select(46:57) %>% slice_head()) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

cor	CCC	bias2	MSE	MLP	MLA	RMLP	RMLA	PLP	PLA	PAB	PPB
0.7427129	0.7096774	0.1225	3.825	3.238983	0.5860167	1.799718	0.7655173	84.6793	15.3207	3.202614	12.11808

```r
# Metrics summary using defined functions
ex.summary.functions = example %>% dplyr::select(1,46:57) %>%
  group_by(sim) %>% slice_head() %>% ungroup()

# Print summary of functions
kable(ex.summary.functions) %>%
  kable_styling(latex_options =c("striped","scale_down"))
```

sim	cor	CCC	bias2	MSE	MLP	MLA	RMLP	RMLA	PLP	PLA	PAB	PPB
1	0.7427129	0.7096774	0.1225	3.825	3.238983	0.5860167	1.799718	0.7655173	84.6793	15.3207	3.202614	12.11808

APSIM Datasets.

The following four datasets are collections of point forecasts from multiple locations belonging to simulation modules at different stages of their calibration process of APSIM model (Holzworth et al., 2018), described in Table 4, by Correndo et al. (2021).

Disclaimer: *The data published here originated from the following GitHub repository: https://github.com/APSIMInitiative/ApsimX. This data has been provided for use for the development of APSIM. Any Improvements to APSIM are owned by the APSIM Initiative and are covered by the terms and conditions which can be accessed here: www.apsim.info.*

Note: *The next coding lines allow users to read multiple worksheets in a single excel file at once. Users could replace the provided with their own datasets. If the number of new datasets differs from 4, user must change the assigned "id" below*

```
# Read the worksheets in a xlsx file
# define the file path
path = "APSIM_datasets.xlsx"
sheet_names = path %>% excel_sheets()

# Read the multiple worksheets
mad = path %>% excel_sheets() %>% set_names(., sheet_names) %>% map(read_excel, path = path)

# Extracting dataframes
# create a tibble of necessary rows (or use the sheet names vector)
# mapping must be done with map using 1:n vector because the data is still under "list" format
APSIM =  as.tibble(sheet_names) %>%
  rename(., dataset = value) %>%
  mutate(examples = map(seq(1,4,by=1),~as.data.frame(mad[[.]])),
         # Assign "id", UPDATE if removing/adding datasets (as worksheets)
         id = seq(1,4,by=1))

# Unnest and select variables of interest
APSIM = APSIM %>% unnest() %>% dplyr::select(id, dataset, predicted, observed)

# Filtering datasets 1,2,3,4
APSIM_datasets = APSIM %>%
  # ID corresponding to the worksheet order
  filter(id == 1 | id == 2  | id == 3 | id == 4) %>%
  dplyr::select(id, observed, predicted) %>%
  # Renaming observed (O_i) and predicted columns (P_i)
  mutate(O_i = observed, P_i = predicted) %>%
  dplyr::select(id, O_i, P_i) %>% nest(data=c('O_i', 'P_i')) %>%
  # Assing unique letters to the datasets
  mutate(Dataset = rep(c("A","B","C","D"))) %>%
  dplyr::select(-id) %>% arrange(., Dataset)
```

APSIM examples.

Error Metrics.

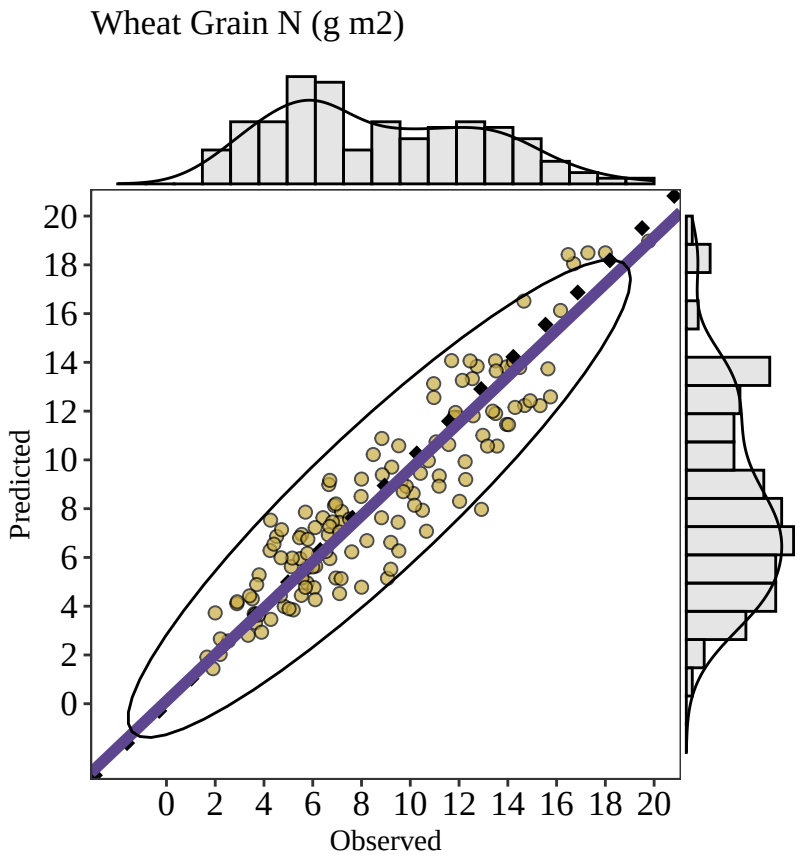
```
APSIM_metrics = APSIM_datasets %>%
  unnest(cols=c("data")) %>% # Unnest observed, predicted values
  group_by(Dataset) %>% # Grouping by dataset
## ESTIMATING METRICS
  mutate(mean.O = mean (O_i),
         mean.P = mean (P_i),
         cor = cor(P_i,O_i), # 1. Correlation Coeff.(Pearson)
         R2 = cor^2, # 2. Coeff. of determination R2
         MBE = MBErr(O_i,P_i), # 3. Mean Bias Error P to O
         MSE = MSE(O_i,P_i), # 4.a. MSE
         RMSE = RMSE(O_i, P_i), # 5.b. RMSE
         Xa = Xa(O_i,P_i), # 6.a. Accuracy component of CCC/Lambda
         CCC = CCC(O_i,P_i), # 6.b. Concordance Correlation Coeff. (Lin, 1989)
      # Proposed MSE segregation
         MLP = MLP(O_i, P_i),# Mean Lack of Precision
         MLA = MLA(O_i, P_i),# Mean Lack of Accuracy
         RMLP = sqrt(MLP), # Root Mean Lack of Precision
         RMLA = sqrt(MLA), # Root Mean Lack of Accuracy
         PLP = 100*(MLP/MSE),# Percentage Lack of Precision
         PLA = 100*(MLA/MSE),# Percentage Lack of Accuracy
         PAB = PAB(O_i,P_i),# Percentage Additive Bias
         PPB = PPB(O_i,P_i),# Percentage Proportional Bias
      # Kobayashi and Salam (2000) MSE segregation
         LCS = LCS(O_i,P_i), # Lack of Correlation weighed by SD
         SDSD = SDSD(O_i,P_i), # Square SD diff
         SB = SB(O_i,P_i), # Square bias
         MSE_kob = LCS + SDSD + SB, # MSE as sum of Kobashi's terms
      # Theils TSS segregation (Smith & Rose, 1995)
         Ub = Ub(O_i,P_i), # Theils's bias contribution (%)
         Uc = Uc(O_i,P_i), # Theil's consistency contribution (%)
         Ue = Ue(O_i,P_i), # Theil's unexplained variance contribution (%)
      # Equivalence with Theils
         PLP2 = PLP(O_i, P_i), # PLP as Theil's Ue
         PLA2 = PLA(O_i, P_i), # PLA as sum of Theil's Ub + Uc
      # Regression estimators
         SMA_slope = sd(P_i)/sd(O_i), # Slope SMA (P vs O)
         SMA_int = mean.P - (SMA_slope*mean.O), # SMA Intercept (P vs O)
         MA_slope = var(P_i) - var(O_i) +
           sqrt((var(P_i) - var(O_i))^2 +
                  4*(cov(O_i,P_i)^2))/(2*cov(O_i,P_i)), # Slope MA
         MA_int = mean.P - (MA_slope*mean.O), # SMA Intercept
         OLSv_slope = cov(O_i,P_i)/var(O_i), # OLSv slope
         OLSv_int = mean.P - (OLSv_slope*mean.O), # OLSv Intercept
         OLSh_slope = var(P_i)/cov(O_i,P_i), # OLSh slope
         OLSh_int = mean.P - (OLSh_slope*mean.O) # OLSh Intercept
         ) %>%
  mutate_if(.,is.numeric, round, digits = 2) %>%
  nest(data=c(P_i,O_i)) %>%
  filter(row_number()==1) %>%
```

```r
  mutate_if(.,is.numeric, round, digits = 2)

# Transpose and print table
Metrics = as_tibble(t(APSIM_metrics %>% dplyr::select(-data)), rownames = "Atribute") %>%
  setNames(., rep(c("Example", "Wheat","Barley",
                    "Sorghum", "Chickpea")) )

# Print Error metrics
kable(Metrics) %>%
  kable_styling(latex_options =c("striped","hold_position"))
```

Example	Wheat	Barley	Sorghum	Chickpea
Dataset	A	B	C	D
mean.O	8.73	18.78	24.73	692.34
mean.P	8.41	18.57	24.01	1059.10
cor	0.92	0.67	0.74	0.68
R2	0.85	0.45	0.55	0.46
MBE	-0.32	-0.21	-0.72	366.76
MSE	2.78	15.89	37.99	183206.79
RMSE	1.67	3.99	6.16	428.03
Xa	1.00	1.00	0.83	0.42
CCC	0.92	0.67	0.62	0.28
MLP	2.63	15.72	21.07	34314.25
MLA	0.14	0.17	16.92	148892.53
RMLP	1.62	3.96	4.59	185.24
RMLA	0.38	0.42	4.11	385.87
PLP	94.84	98.91	55.46	18.73
PLA	5.16	1.09	44.54	81.27
PAB	3.65	0.27	1.35	73.42
PPB	1.51	0.82	43.19	7.85
LCS	2.63	15.72	21.07	34314.25
SDSD	0.04	0.13	16.40	14379.87
SB	0.10	0.04	0.51	134512.66
MSE_kob	2.78	15.89	37.99	183206.79
Ub	3.65	0.27	1.35	73.42
Uc	1.51	0.82	43.19	7.85
Ue	94.84	98.91	55.46	18.73
PLP2	94.84	98.91	55.46	18.73
PLA2	5.16	1.09	44.54	81.27
SMA_slope	0.95	0.93	0.54	1.67
SMA_int	0.11	1.13	10.72	-99.01
MA_slope	-0.67	-2.58	-54.80	58634.37
MA_int	14.26	67.06	1379.20	-40593966.08
OLSv_slope	0.87	0.62	0.40	1.13
OLSv_int	0.78	6.85	14.11	274.85
OLSh_slope	1.03	1.38	0.72	2.47
OLSh_int	-0.61	-7.39	6.16	-651.10

A. Wheat Grain N

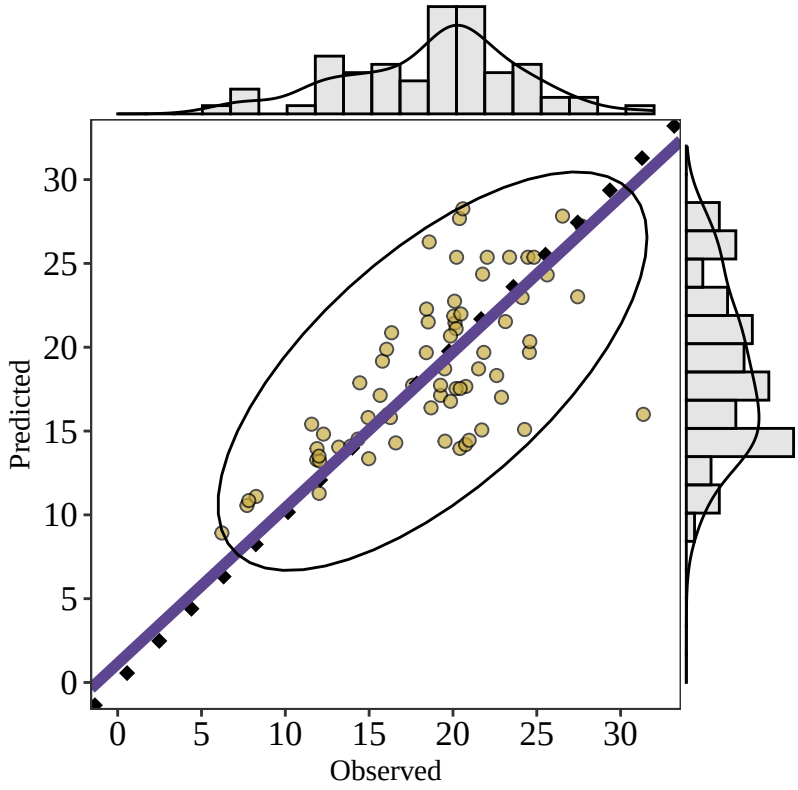
```
# Data
APSIM_A = APSIM_datasets %>% filter(Dataset == "A") %>% unnest(cols = c("data"))

# SMA regression (smatr, see Warton et al. 2012)
APSIM_A_sma = APSIM_datasets %>% filter(Dataset == "A") %>%
  unnest() %>% sma(P_i~O_i,data=.)

FIG5_A = APSIM_A %>%
  ggplot(aes(x = O_i, y=P_i))+
  geom_point(shape = 21, color = "black", fill="#c9ac40", alpha=0.7,
             size= 2,show.legend=F)+
  geom_abline(linetype = "dotted", size=2)+ # 1:1 line
  geom_abline(slope =  APSIM_A_sma %>% coef %>% .[[2]] ,
              intercept = APSIM_A_sma %>% coef %>% .[[1]],
              linetype="solid", col="#5d468f", size=2)+ # SMA regression line
  stat_ellipse(type="norm", level=0.95)+ # Data ellipse
  scale_x_continuous(limits = c(-2,20), breaks = seq(0,20, by=2))+
  scale_y_continuous(limits = c(-2,20), breaks = seq(0,20, by=2))+
  labs(title = "Wheat Grain N (g m2)", x = "Observed", y = "Predicted")+
  theme_bw()+
  theme(panel.grid = element_blank(),
        title = element_text(family = "serif", color="black",
                             size = rel(1)),
        axis.text.y = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        axis.text.x = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        strip.text = element_blank(),
        strip.background = element_blank(),
        aspect.ratio = 1)

plot.5A = ggMarginal(FIG5_A,type = "densigram",
    xparams = list(col="black",fill="grey",bins = 20, alpha=0.4, size = 0.5),
    yparams = list(col="black",fill="grey", bins = 20,alpha=0.4, size=0.5))


plot.5A
```


Wheat Grain N (g m2)
Predicted
Observed
0
2
4
6
8
10
12
14
16
18
20

B. Barley Grain No.

```
# Data
APSIM_B = APSIM_datasets %>% filter(Dataset == "B") %>%  unnest(cols = c("data"))

# SMA regression (smatr, see Warton et al. 2012)
APSIM_B_sma = APSIM_datasets %>% filter(Dataset == "B") %>%
  unnest() %>% sma(P_i~O_i,data=.)

FIG5_B = APSIM_B %>%
  ggplot(aes(x = O_i, y=P_i))+
  geom_point(shape = 21, color = "black", fill="#c9ac40", alpha=0.7,
             size= 2,show.legend=F)+
  geom_abline(linetype = "dotted", size=2)+
  geom_abline(slope =  APSIM_B_sma %>% coef %>% .[[2]],
              intercept = APSIM_B_sma %>% coef %>% .[[1]],
              linetype="solid", col="#5d468f", size=2)+
  stat_ellipse(type="norm", level=0.95)+
  scale_x_continuous(limits = c(0,32), breaks = seq(0,30, by=5))+
  scale_y_continuous(limits = c(0,32), breaks = seq(0,30, by=5))+
  labs(title = "Barley Grain No.", x = "Observed", y = "Predicted")+
  theme_bw()+
  theme(panel.grid = element_blank(),
        title = element_text(family = "serif", color="black",
                             size = rel(1)),
        axis.text.y = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        axis.text.x = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        strip.text = element_blank(),
        strip.background = element_blank(),
        aspect.ratio = 1)

plot.5B = ggMarginal(FIG5_B,type = "densigram",
      xparams = list(col="black",fill="grey",bins = 20, alpha=0.4, size = 0.5),
      yparams = list(col="black",fill="grey", bins = 20,alpha=0.4, size=0.5))

plot.5B
```

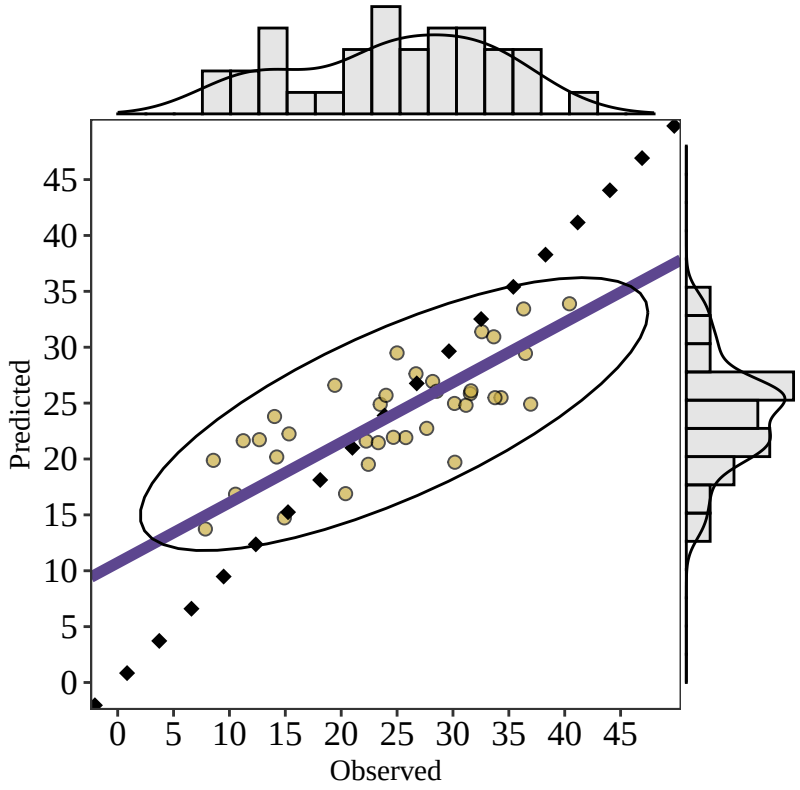
Barley Grain No.



C. Sorghum Grain No.

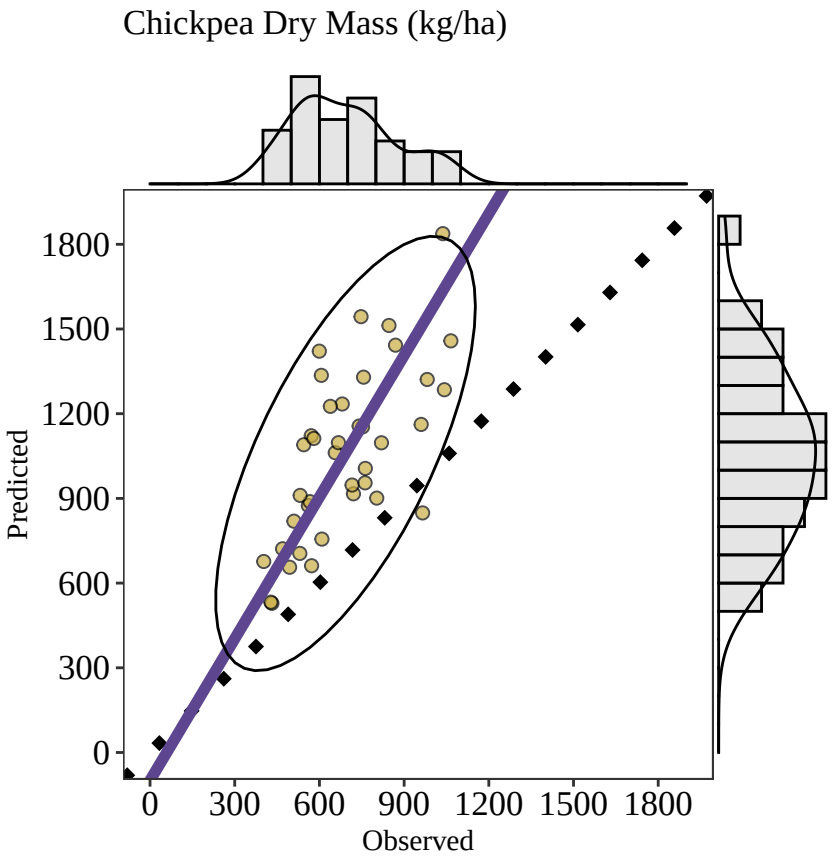
```
# Data
APSIM_C = APSIM_datasets %>% filter(Dataset == "C") %>%  unnest(cols = c("data"))

# SMA regression (smatr, see Warton et al. 2012)
APSIM_C_sma = APSIM_datasets %>% filter(Dataset == "C") %>%
  unnest() %>% sma(P_i~O_i,data=.)

FIG5_C = APSIM_C %>%
  ggplot(aes(x = O_i, y=P_i))+
  geom_point(shape = 21, color = "black", fill="#c9ac40", alpha=0.7,
             size= 2,show.legend=F)+
  geom_abline(linetype = "dotted", size=2)+
  geom_abline(slope =  APSIM_C_sma %>% coef %>% .[[2]],
              intercept = APSIM_C_sma %>% coef %>% .[[1]],
              linetype="solid", col="#5d468f", size=2)+
  stat_ellipse(type="norm", level=0.95)+
  scale_x_continuous(limits = c(0,48), breaks = seq(0,45, by=5))+
  scale_y_continuous(limits = c(0,48), breaks = seq(0,45, by=5))+
  labs(title = "Sorghum Grain No.", x = "Observed", y = "Predicted")+
  theme_bw()+
  theme(panel.grid = element_blank(),
        title = element_text(family = "serif", color="black",
                             size = rel(1)),
        axis.text.y = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        axis.text.x = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        strip.text = element_blank(),
        strip.background = element_blank(),
        aspect.ratio = 1)

plot.5C = ggMarginal(FIG5_C,type = "densigram",
      xparams = list(col="black",fill="grey",bins = 20, alpha=0.4, size = 0.5),
      yparams = list(col="black",fill="grey", bins = 20,alpha=0.4, size=0.5))

plot.5C
```

Sorghum Grain No.



D. Chickpea Dry-mass.

```
# Data
APSIM_D = APSIM_datasets %>% filter(Dataset == "D") %>%  unnest(cols = c("data"))

# SMA regression (smatr, see Warton et al. 2012)
APSIM_D_sma = APSIM_datasets %>% filter(Dataset == "D") %>%
  unnest() %>% sma(P_i~O_i,data=.)

FIG5_D = APSIM_D %>%
  ggplot(aes(x = O_i, y=P_i))+
  geom_point(shape = 21, color = "black", fill="#c9ac40",
             alpha=0.7, size= 2,show.legend=F)+
  geom_abline(linetype = "dotted", size=2)+
  geom_abline(slope =  APSIM_D_sma %>% coef %>% .[[2]],
              intercept = APSIM_D_sma %>% coef %>% .[[1]],
              linetype="solid", col="#5d468f", size=2)+
  stat_ellipse(type="norm", level=0.95)+
  scale_x_continuous(limits = c(0,1900), breaks = seq(0,1800, by=300))+
  scale_y_continuous(limits = c(0,1900), breaks = seq(0,1800, by=300))+
  labs(title = "Chickpea Dry Mass (kg/ha)", x = "Observed", y = "Predicted")+
  theme_bw()+
  theme(panel.grid = element_blank(),
        title = element_text(family = "serif", color="black",
                             size = rel(1)),
        axis.text.y = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        axis.text.x = element_text(family = "serif", color="black",
                                   size = rel(1.5)),
        strip.text = element_blank(),
        strip.background = element_blank(),
        aspect.ratio = 1)

plot.5D = ggMarginal(FIG5_D,type = "densigram",
      xparams = list(col="black",fill="grey",bins = 20, alpha=0.4, size = 0.5),
      yparams = list(col="black",fill="grey", bins = 20,alpha=0.4, size=0.5))

plot.5D
```


Chickpea Dry Mass (kg/ha)
Predicted
Observed
0
300
600
900
1200
1500
1800

REFERENCES

- Correndo, A.A., Hefley, T., Holzworth, D., Ciampitti, I.A., *Under-review.* Revisiting linear regression to test agreement in continuous predicted-observed datasets. *Agr. Syst.* x, xx-xx.
- Duveiller, G., Fasbender, D., Meroni, M., 2016. Revisiting the concept of a symmetric index of agreement for continuous datasets. *Sci. Rep.* 6, 1–14. https://doi.org/10.1038/srep19401
- Gupta, H. V., Kling, H., Yilmaz, K.K., Martinez, G.F., 2009. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* 377, 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003
- Holzworth, D., Huth, N.I., Fainges, J., Brown, H., Zurcher, E., Cichota, R., Verrall, S., Herrmann, N.I., Zheng, B., Snow, V., 2018. APSIM Next Generation: Overcoming challenges in modernising a farming systems model. *Environ. Model. Softw.* 103, 43–51. https://doi.org/10.1016/j.envsoft.2018.02.002
- Ji, L., Gallo, K., 2006. An agreement coefficient for image comparison. Photogramm. *Eng. Remote Sensing.* 7, July 2006, 823-833. https://doi.org/10.14358/PERS.72.7.823
- Kobayashi, K., Salam, M.U., 2000. Comparing simulated and measured values using mean squared deviation and its components. *Agron. J.* 92, 345–352. https://doi.org/10.2134/agronj2000.922345x
- Lin, L.I.-K., 1989. A concordance correlation coefficient to evaluate reproducibility. Biometrics. 45(1), 255-268. https://doi.org/10.2307/2532051
- Smith, E.P., Rose, K.A., 1995. Model goodness-of-fit analysis using regression and related techniques. *Ecol. Modell.* 77, 49–64. https://doi.org/10.1016/0304-3800(93)E0074-D
- Wallach, D., Makowski, D., Jones, J.W., Brun, F., 2019. Chapter 9: Model evaluation. In: D. Wallach, D. Makowski, J.W. Jones, F. Brun, eds. Working with dynamic crop models, 3rd edn. San Diego, CA, USA: Academic Press, 311- 373.
- Warton, D.I., Wright, I.J., Falster, D.S., Westoby, M., 2006. Bivariate line-fitting methods for allometry. *Biol. Rev. Camb. Philos. Soc.* 81, 259–291. https://doi.org/10.1017/S1464793106007007
- Warton, D.I., Duursma, R.A., Falster, D.S., Taskinen, S., 2012. smatr 3– an R package for estimation and inference about allometric lines. *Methods in Ecology and Evolution* 3: 257-259. https://doi.org/10.1111/j.2041-210X.2011.00153.x
- Willmott, C.J., 1981. On the validation of models. *Phys. Geogr.* 2, 184–194. https://doi.org/10.1080/02723646.1981.10642213