

# GC measurements

## Necessary libraries

## Read file

```
master <- read_excel("RES_LCMS_GC.xlsx", sheet=1)
master$Sensor <- as.factor(master$Sensor)

fan_off <- master %>%
  select(-LCMS_ppm_hr_fan_on, -LCMS_ppm_hr_3point) %>%
  na.omit()

fan_on <- master %>%
  select(-LCMS_ppm_hr, -LCMS_ppm_hr_3point) %>%
  na.omit()

three_point <- master %>%
  select(-LCMS_ppm_hr, -LCMS_ppm_hr_fan_on) %>%
  na.omit()

str(master)
```

```
## tibble [79 x 6] (S3: tbl_df/tbl/data.frame)
## $ Plot      : chr [1:79] "C1_Vincent_Aug1" "SMW2_Vincent_Aug1" "C1_SMW2_Vincent_Aug27" "C1_V
## $ Sensor    : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 3 3 2 2 2 ...
## $ GC_ppm_hr : num [1:79] 69.4 107.5 NA 83.1 63.2 ...
## $ LCMS_ppm_hr : num [1:79] 26.8 43.4 NA 25.6 33.2 ...
## $ LCMS_ppm_hr_fan_on: num [1:79] 64.2 86.4 NA NA NA ...
## $ LCMS_ppm_hr_3point: num [1:79] 25.6 38.8 NA 26.9 36.2 ...
```

## fan off

```
all_plotted_fan_off <-
ggplot(data=fan_off, aes(x=LCMS_ppm_hr, y=GC_ppm_hr))+
  geom_point(aes(color=Sensor))+
  scale_x_continuous(limits = c(-0, 149), expand = c(0, 0))+
  scale_y_continuous(limits = c(-0, 149), expand = c(0, 0))+
  geom_abline(intercept = 0, slope = 1)+
  stat_regline_equation(aes(x=LCMS_ppm_hr, y=GC_ppm_hr,
    label = paste(..rr.label..)),
    show.legend = FALSE,
```

```

    label.x = 75,
    label.y = 45)+
stat_regline_equation(aes(x=LCMS_ppm_hr, y=GC_ppm_hr,
    label = paste(..eq.label..)),
    show.legend = FALSE,
    label.x = 75,
    label.y = 50)+
geom_smooth(data=fan_off, aes(x=LCMS_ppm_hr, y=GC_ppm_hr),method="lm", level = 0.95)+
ggtitle("(a) Fan off")

```

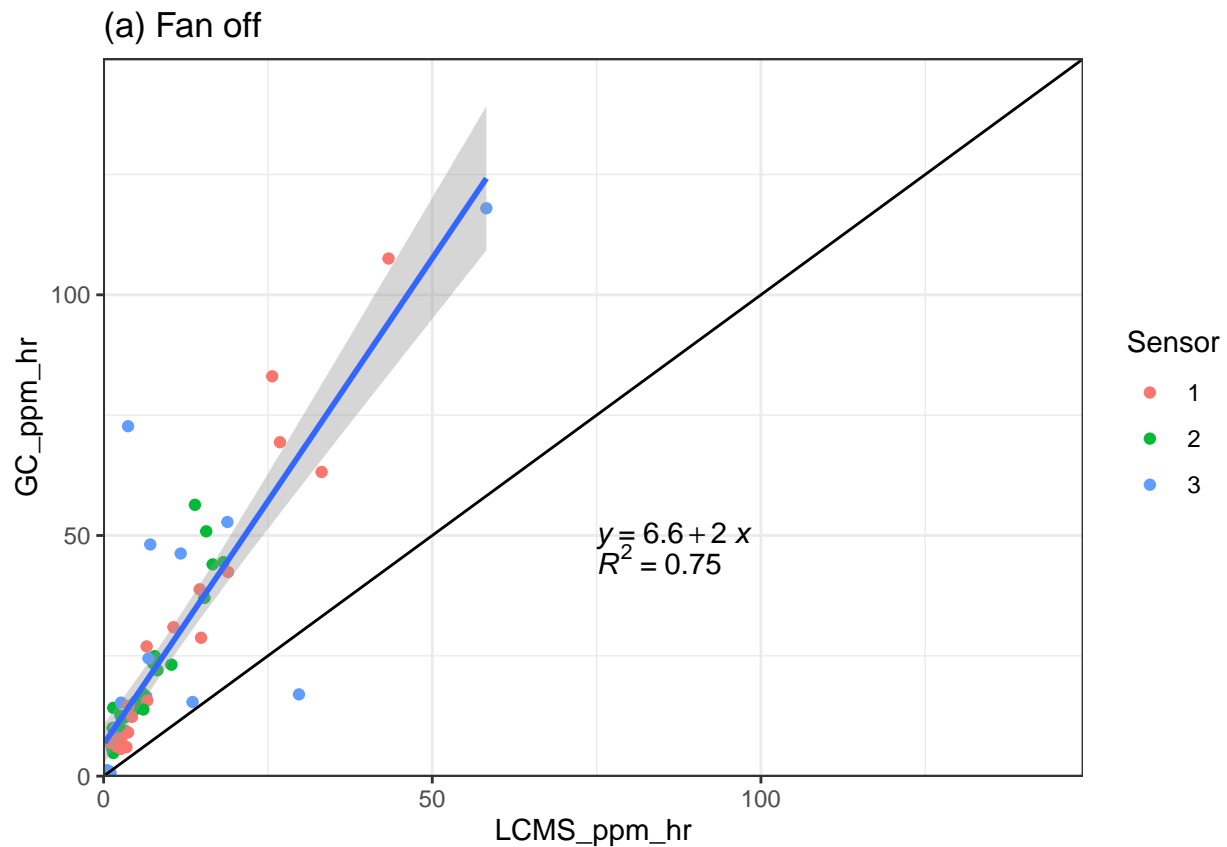
all\_plotted\_fan\_off

```

## Warning: The dot-dot notation ('..rr.label..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(rr.label)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
summary(lm(GC_ppm_hr~LCMS_ppm_hr, data =fan_off)) #coefficients are the same, y=mx+c and r2
```

```
##
```

```
## Call:
## lm(formula = GC_ppm_hr ~ LCMS_ppm_hr, data = fan_off)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -49.645  -4.912  -1.575   2.623  58.561
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.6319     2.0679   3.207  0.00214 **
## LCMS_ppm_hr    2.0189     0.1478  13.660 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.68 on 61 degrees of freedom
## Multiple R-squared:  0.7536, Adjusted R-squared:  0.7496
## F-statistic: 186.6 on 1 and 61 DF,  p-value: < 2.2e-16
```

## fan off transformed

```
lm_fan_off <- lm(GC_ppm_hr~LCMS_ppm_hr, data =fan_off)

fan_off$fan_off_predicted_ppm <- predict(lm_fan_off, fan_off)

all_plotted_fan_off_predicted <-
ggplot(data=fan_off, aes(x=fan_off_predicted_ppm, y=GC_ppm_hr))+
  geom_point(aes(color=Sensor))+
  scale_x_continuous(limits = c(-0, 149), expand = c(0, 0))+
  scale_y_continuous(limits = c(-0, 149), expand = c(0, 0))+
  geom_abline(intercept = 0, slope = 1)+
  geom_smooth(data=fan_off, aes(x=fan_off_predicted_ppm, y=GC_ppm_hr),method="lm", level = 0.95)+
  ggtitle("(b) Fan off predicted")
```

## fan on

```
all_plotted_fan_on <-
ggplot(data=fan_on, aes(x=LCMS_ppm_hr_fan_on, y=GC_ppm_hr))+
  geom_point(aes(color=Sensor))+
  scale_x_continuous(limits = c(-0, 149), expand = c(0, 0))+
  scale_y_continuous(limits = c(-0, 149), expand = c(0, 0))+
  geom_abline(intercept = 0, slope = 1)+
  stat_regline_equation(aes(x=LCMS_ppm_hr_fan_on, y=GC_ppm_hr,
    label = paste(..rr.label..)),
    show.legend = FALSE,
    label.x = 75,
    label.y = 45)+
  stat_regline_equation(aes(x=LCMS_ppm_hr_fan_on, y=GC_ppm_hr,
    label = paste(..eq.label..)),
```

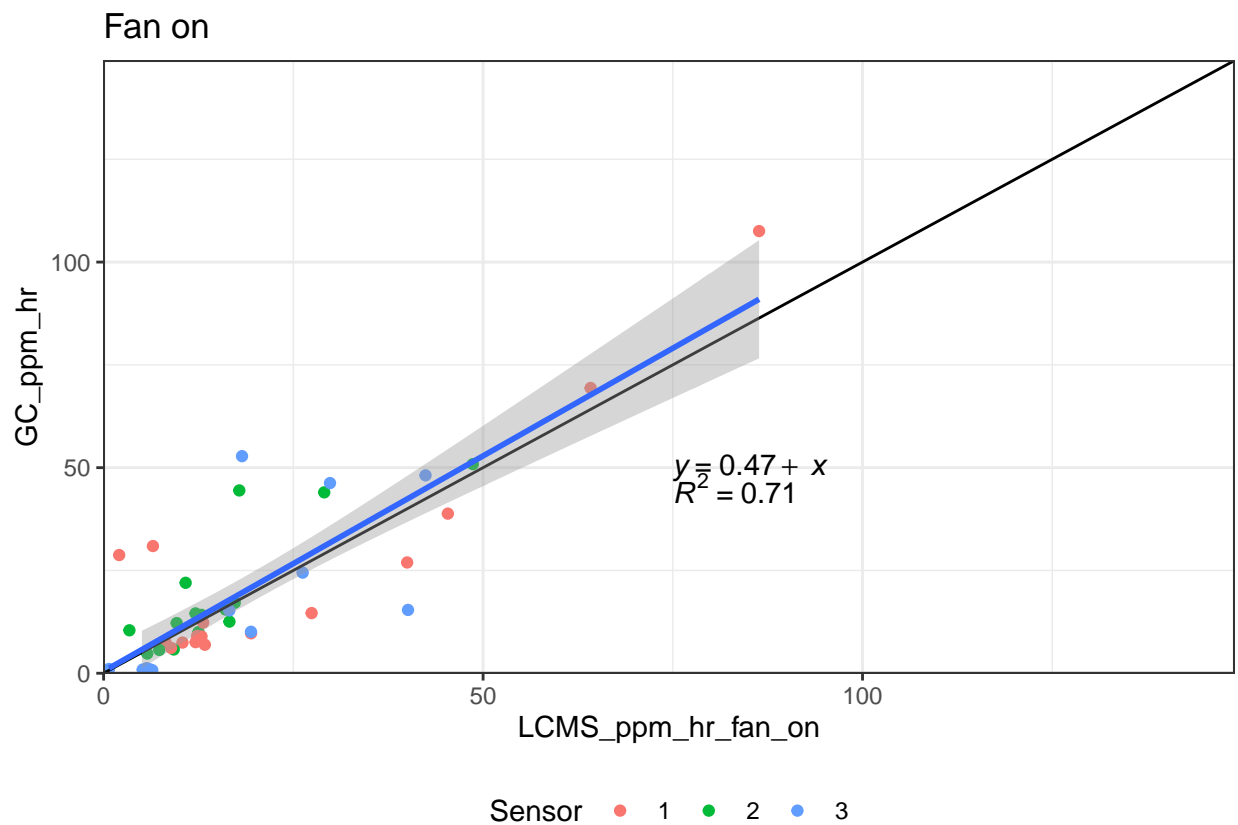
```

show.legend = FALSE,
label.x = 75,
label.y = 50)+
geom_smooth(data=fan_on, aes(x=LCMS_ppm_hr_fan_on, y=GC_ppm_hr),method="lm", level = 0.95)+
ggtitle("Fan on")+
theme(legend.position="bottom")

```

all\_plotted\_fan\_on

```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
summary(lm(GC_ppm_hr~LCMS_ppm_hr_fan_on, data =fan_on)) #coefficients are the same, y=mx+c and r2
```

```
##
## Call:
## lm(formula = GC_ppm_hr ~ LCMS_ppm_hr_fan_on, data = fan_on)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.096  -5.367  -2.521   1.643  33.215
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)

```

```
## (Intercept)          0.4735      2.6207    0.181    0.857
## LCMS_ppm_hr_fan_on   1.0477      0.1024   10.236 5.57e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.68 on 42 degrees of freedom
## Multiple R-squared:  0.7138, Adjusted R-squared:  0.707
## F-statistic: 104.8 on 1 and 42 DF,  p-value: 5.567e-13
```

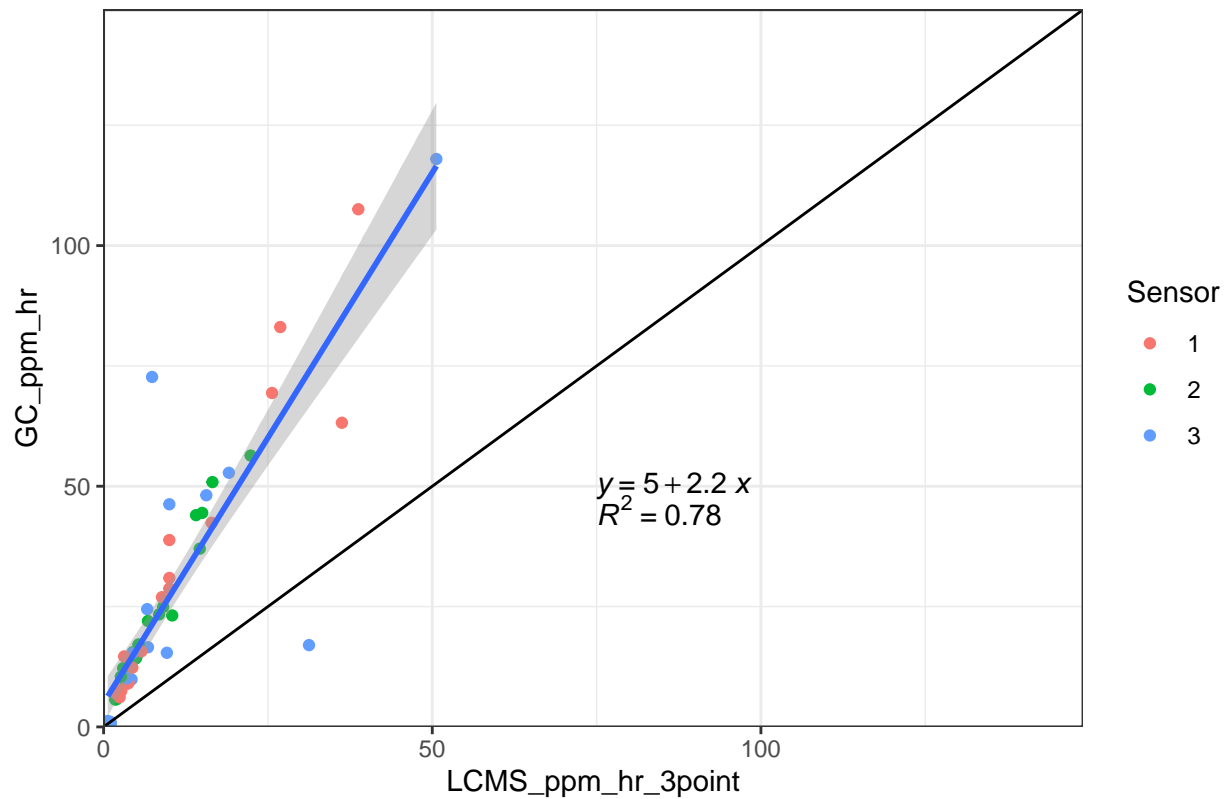
## three point

```
all_plotted_three_point <-
ggplot(data=three_point, aes(x=LCMS_ppm_hr_3point, y=GC_ppm_hr))+
  geom_point(aes(color=Sensor))+
  scale_x_continuous(limits = c(-0, 149), expand = c(0, 0))+
  scale_y_continuous(limits = c(-0, 149), expand = c(0, 0))+
  geom_abline(intercept = 0, slope = 1)+
  stat_regline_equation(aes(x=LCMS_ppm_hr_3point, y=GC_ppm_hr,
    label = paste(..rr.label..)),
    show.legend = FALSE,
    label.x = 75,
    label.y = 45)+
  stat_regline_equation(aes(x=LCMS_ppm_hr_3point, y=GC_ppm_hr,
    label = paste(..eq.label..)),
    show.legend = FALSE,
    label.x = 75,
    label.y = 50)+
  geom_smooth(data=three_point, aes(x=LCMS_ppm_hr_3point, y=GC_ppm_hr),method="lm", level = 0.95)+
  ggtitle("(c) Three point")

all_plotted_three_point
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

(c) Three point



```
summary(lm(GC_ppm_hr~LCMS_ppm_hr_3point, data =three_point)) #coefficients are the same,  $y=mx+c$  and  $r^2$ 
```

```
##
## Call:
## lm(formula = GC_ppm_hr ~ LCMS_ppm_hr_3point, data = three_point)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -56.851  -3.717  -1.393   2.260  51.454
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.0038     2.1567   2.32  0.0239 *
## LCMS_ppm_hr_3point 2.2028     0.1551  14.20 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.21 on 57 degrees of freedom
## Multiple R-squared:  0.7797, Adjusted R-squared:  0.7758
## F-statistic: 201.7 on 1 and 57 DF,  p-value: < 2.2e-16
```

## three point transformed

```
lm_three_point <- lm(GC_ppm_hr~LCMS_ppm_hr_3point, data =three_point)

three_point$three_point_predicted_ppm <- predict(lm_three_point, three_point)

all_plotted_three_point_predicted <-
ggplot(data=three_point, aes(x=three_point_predicted_ppm, y=GC_ppm_hr))+
  geom_point(aes(color=Sensor))+
  scale_x_continuous(limits = c(-0, 149), expand = c(0, 0))+
  scale_y_continuous(limits = c(-0, 149), expand = c(0, 0))+
  geom_abline(intercept = 0, slope = 1)+
  geom_smooth(data=three_point, aes(x=three_point_predicted_ppm, y=GC_ppm_hr),method="lm", level = 0.95,
  ggtitle("(d) Three point predicted")
```

## save the plots

```
combined <- ggarrange(all_plotted_fan_off,
                      all_plotted_fan_off_predicted,
                      all_plotted_three_point,
                      all_plotted_three_point_predicted,
                      nrow = 2,
                      ncol =2,
                      common.legend = TRUE,
                      legend= "bottom")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
```

```
ggsave(filename = "all_plotted.jpg", # Include the file extension here
        plot = combined,             # Specify the plot
        #path = "D:/Academics/UC Davis/School Work/Linguist Lab/Data/R stats/Agronomic paper/Figures",
        dpi = 400,
        height = 21, width = 20, units = "cm")

ggsave(filename = "fan_on.jpg", # Include the file extension here
        plot = all_plotted_fan_on,  # Specify the plot
        #path = "D:/Academics/UC Davis/School Work/Linguist Lab/Data/R stats/Agronomic paper/Figures",
        dpi = 400,
        height = 16, width = 15, units = "cm")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```