Session 4: Data visualization

Meklit Chernet, Turry Ouma,Ibnou Dieng’

To create a plot, specify the data in the ggplot() function and add the required layers: variables, aesthetic elements and the type of plot:

```
ggplot(data) +
aes(x = var_x, y = var_y) +
geom_x()
```

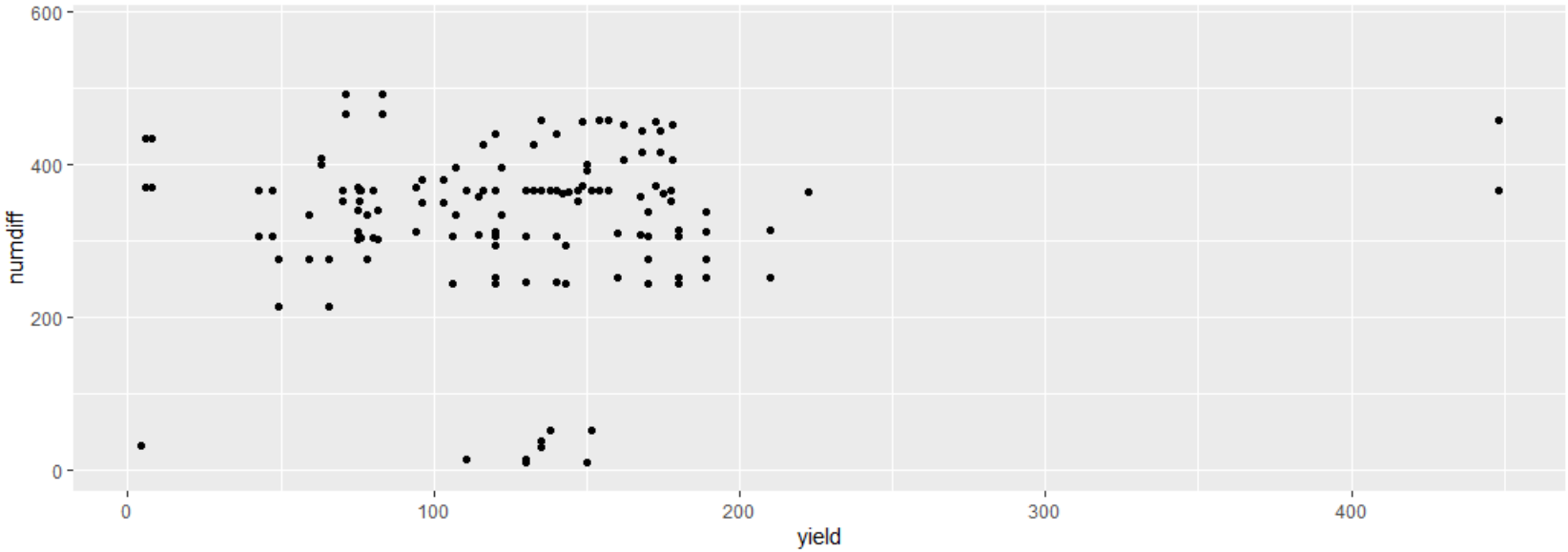
- data in ggplot() is the name of the data frame which contains the variables var_x and var_y.
- The + symbol used to indicate the different layers
- The layer aes() indicates the variables to be used in the plot and more generally, the aesthetic elements of the plot
- x in geom_x() represents the type of plot: geom_point(), geom_line(), etc.

Scatter plot

- A scatter plot is used to visualize the relation between two quantitative variables.

```
ggplot(rti) +   # data
  aes( x=yield,
y=numdiff) + # variables
  geom_point() # type of
plot
```

- Often used to visualize a potential correlation between the two variables
- We create a scatter plot using geom_point()



Look at the output. It is OK but we can do better: personalize the plot to make it more informative. We can add a title, subtitle, caption and edit axis labels with the labs() function:

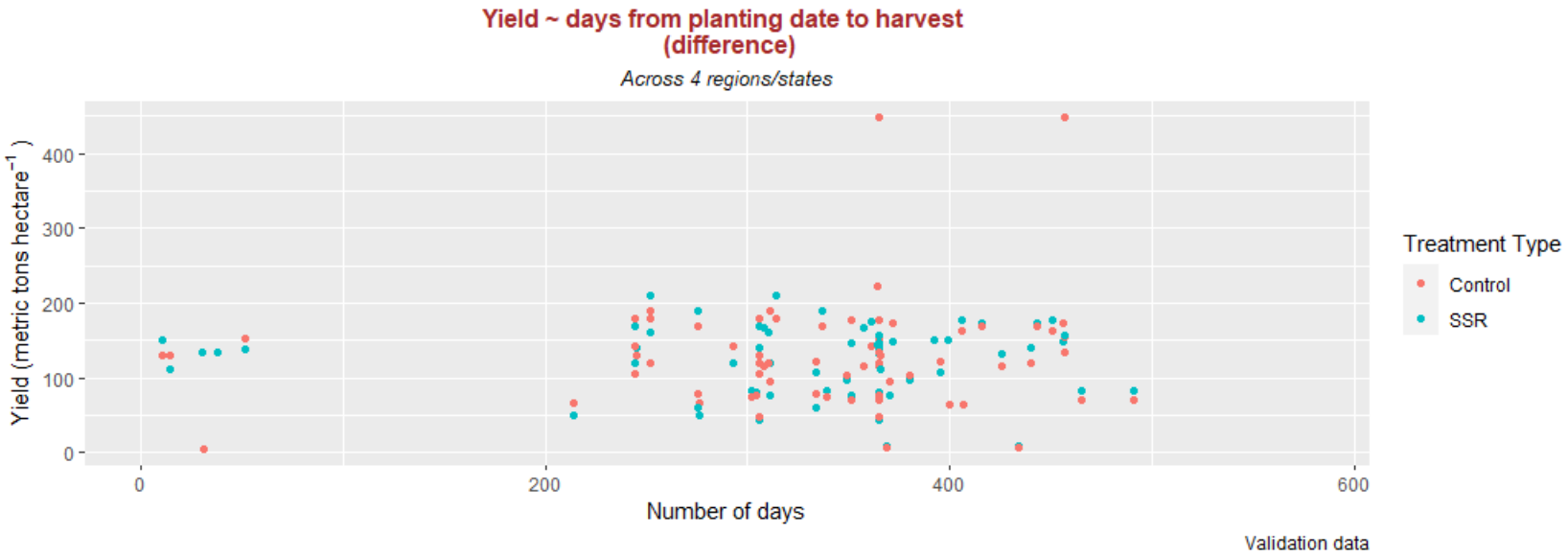
- It is possible to use mathematical equations instead of text strings. The quote() can be used for that. Read about the available options in ?plotmath
- Save the “main” plot in an object, and add more layers. You can edit the alignment, the size and the shape of the title and subtitle via the theme() layer and the element_text() function.
- If the title or subtitle is long, divide it into multiple lines, use \n

Let’s now see it in terms of the treatments applied:

```
p <- ggplot(rti) +   # data
aes( x=numdiff, y=yield,
color=harvest) + # variables
geom_point() # type of plot
```

We can do better by changing the title of the legend: “Treatment Type ” instead of “harvest”

```
p <- p + scale_color_discrete (name
="Treatment Type")
```



Let's change the default color:

The {RColorBrewer} package makes it easy to quickly load sensible color palettes

```
install.packages("RColorBrewer")
library(RColorBrewer)
```

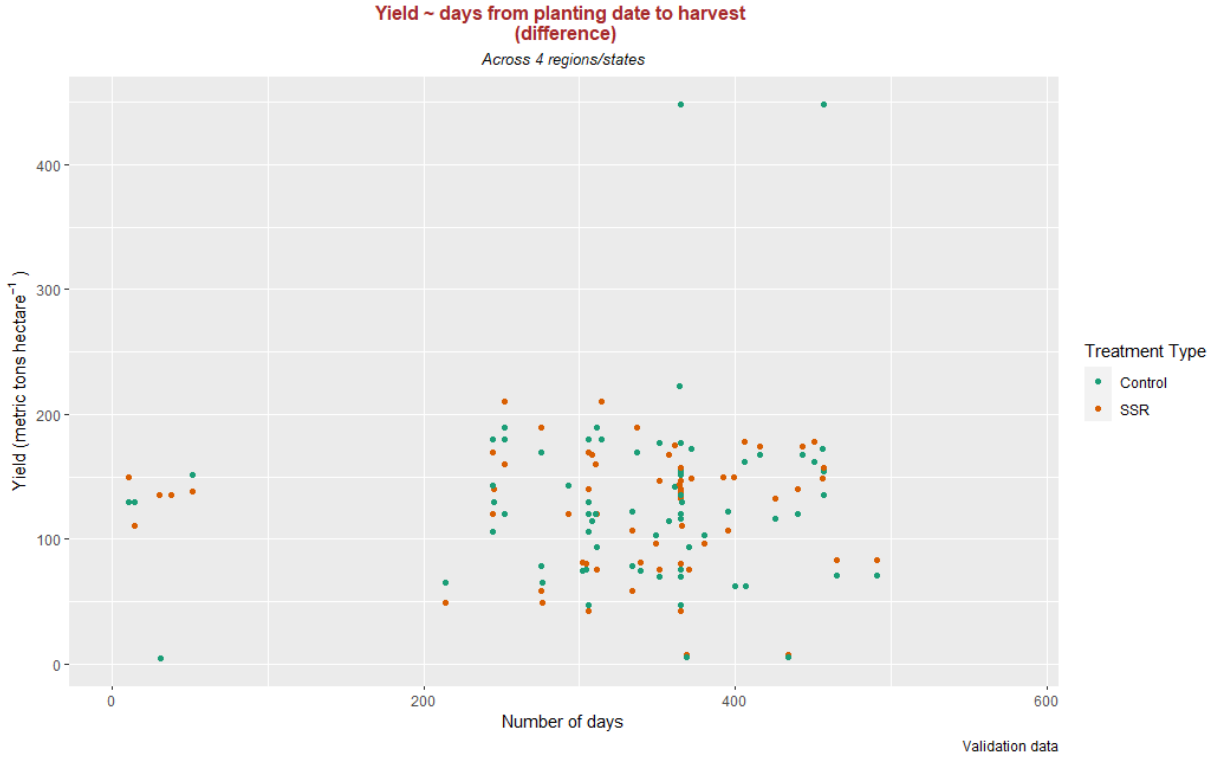
There are three types of palettes: sequential, diverging and qualitative.

- Sequential palettes are suited to ordered data that progress from low to high.
- Diverging palettes are suited to centered data with extremes in either direction.
- Qualitative palettes are suited to nominal or categorical data.

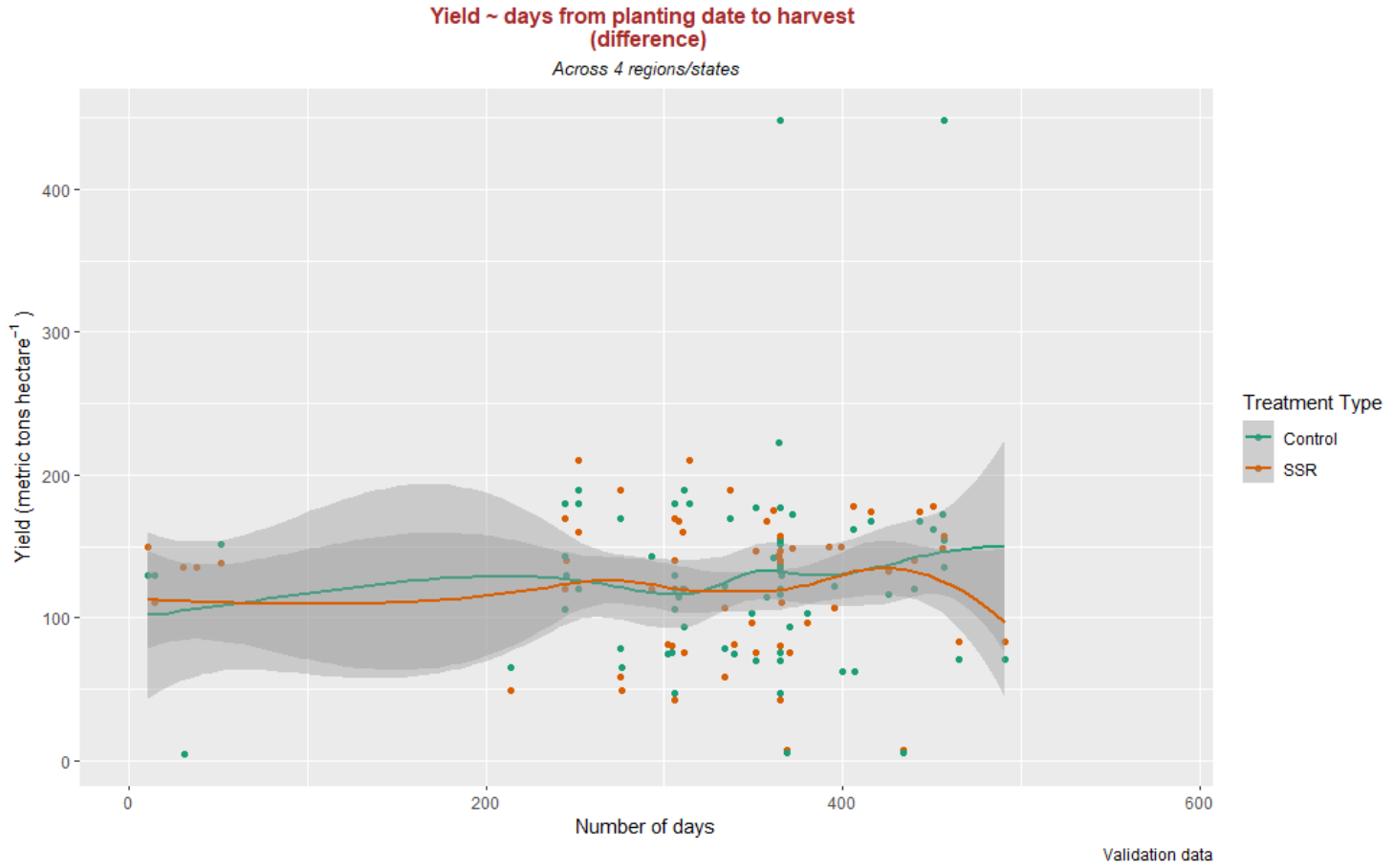
The display.brewer.all() will plot the available palettes

Let's use the scale_color_brewer() to apply the relevant palette:

```
p <- p + scale_color_brewer(name ="Treatment Type" ,palette="Dark2")
```

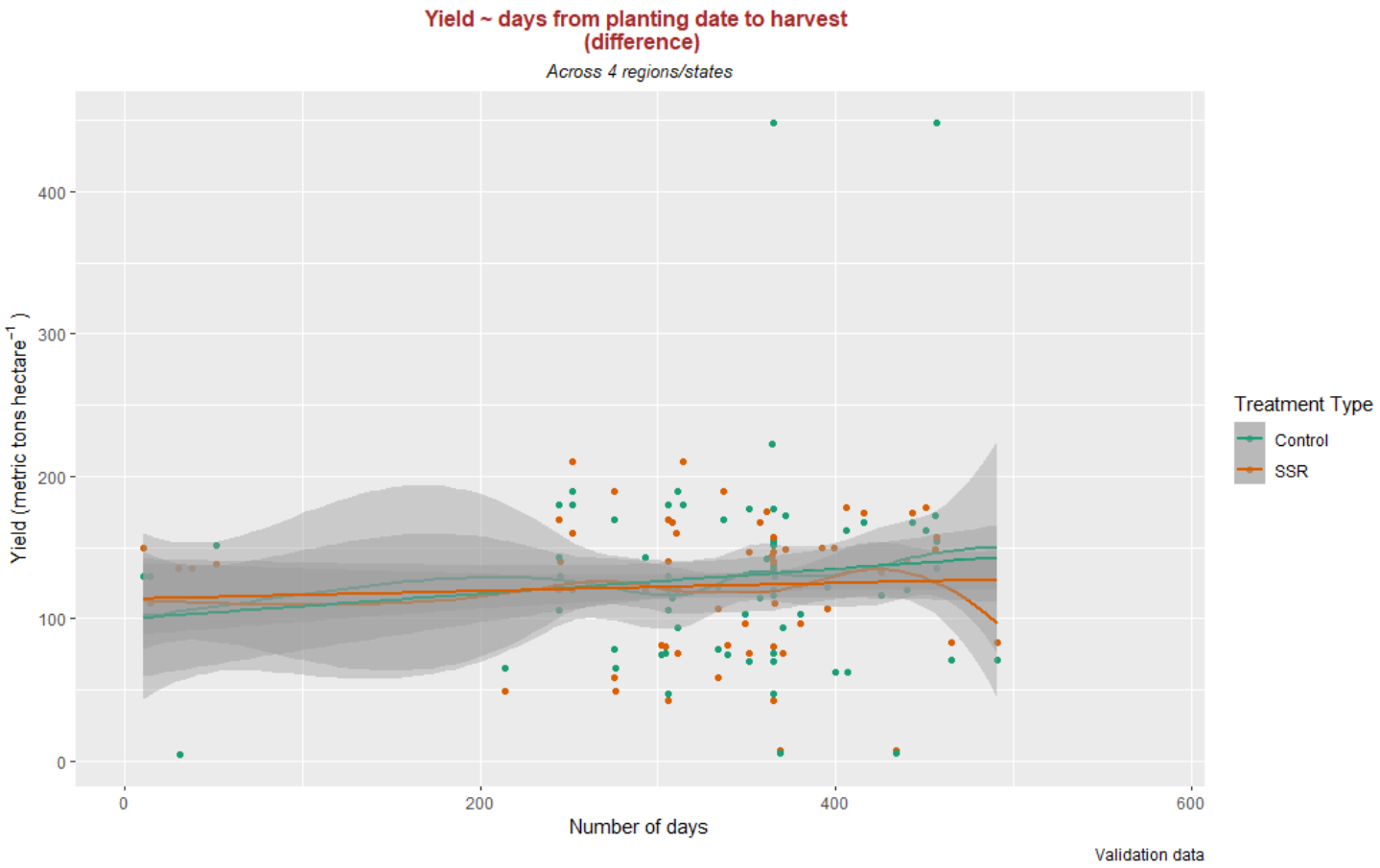


- It is also possible to add a smooth line fitted to the data:
- p <- p + geom_smooth()

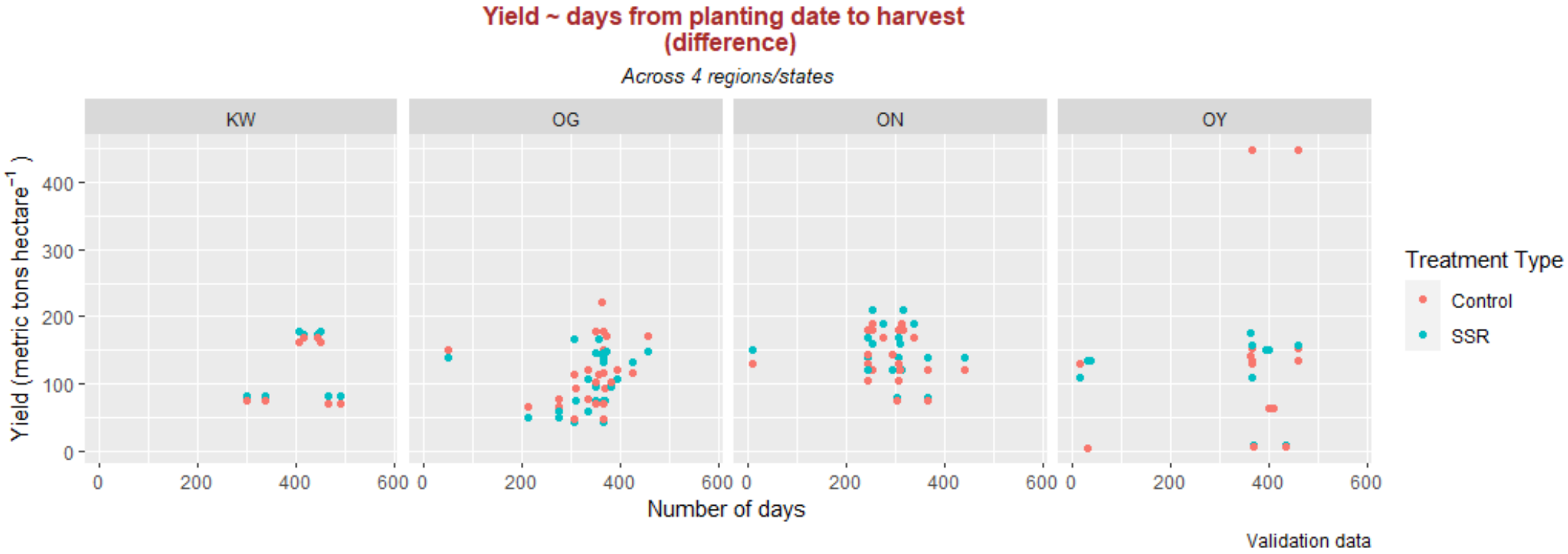


- In case of simple linear regression, it is possible to display the regression line on the plot.
- This can be done by adding method = lm in geom_smooth(), as below:

```
p <- p + geom_smooth(method = lm)
```



Facetting: Instead of plotting all the *regions* in the same plot and using color to differentiate them, we can use facet_grid() to divide the same graphic into several panels according to the values of one (*region*) or even two qualitative variables.



Here's the facetting code:

```
p <- ggplot(data = rti) +   # data
  aes( x=numdiff, y=yield, color=harvest) + # variables
  geom_point() + # type of plot
     labs(title= "Yield ~ days from planting date to harvest \n (difference)", #Add labels
     subtitle = "Across 4 regions/states",
     caption = "Validation data",
     x="Number of days",
     y = quote("Yield (metric tons" ~ "hectare"^{-1} ~ ")"))+
     scale_color_discrete(name="Treatment Type")+#Add mathematical equation
     facet_grid(. ~ region3)+ #facet by region
  theme(
  plot.title = element_text(hjust = 0.5, size = 12, color = "brown",face = "bold"),
  plot.subtitle = element_text(hjust = 0.5, size = 10,color = "black",face = "italic"))
p
```

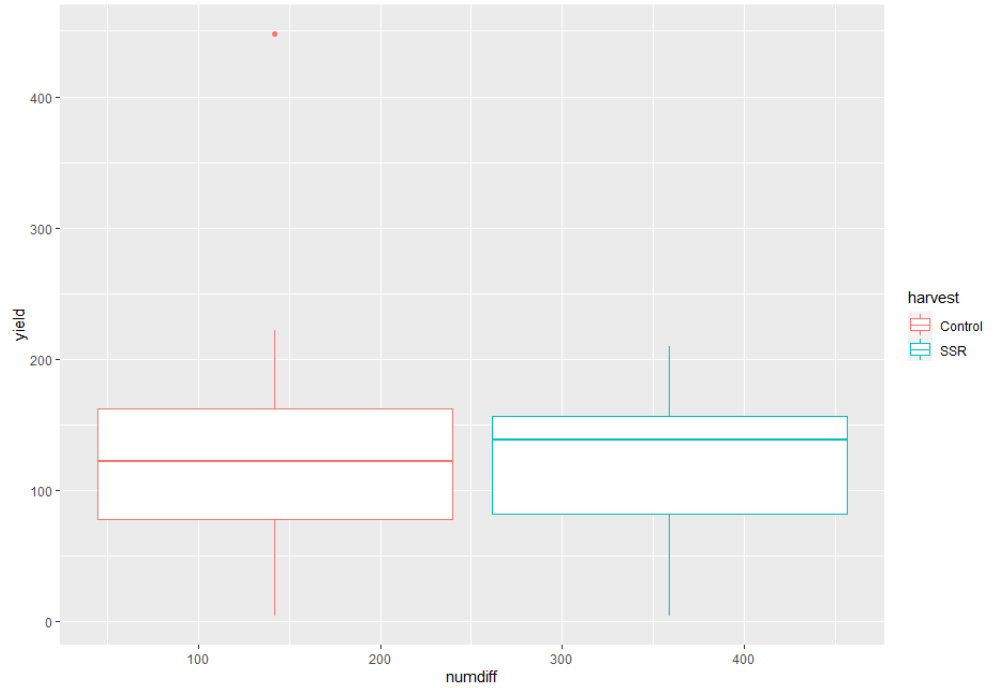
Boxplot

A boxplot can be plotted using geom_boxplot()

A boxplot graphically represents the distribution of a quantitative variable by visually displaying five common location summary (minimum, median, first/third quartiles and maximum) and any observation that was classified as a suspected outlier using the interquartile range (IQR) criterion.

Use this code to get the boxplot:

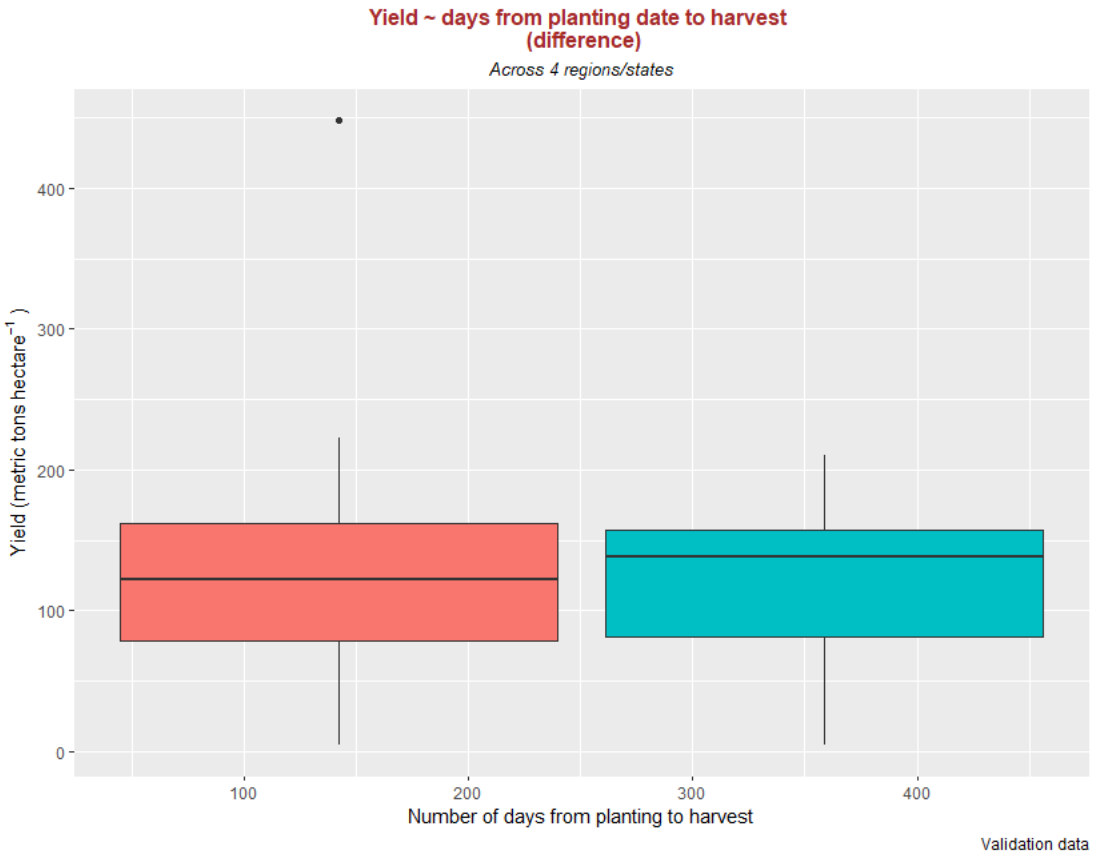
```
p <- ggplot(rti) +   # data
  aes( x=numdiff, y=yield, color=harvest) + # variables
  geom_boxplot() # type of plot

p <- p + labs(x = "",
              y = "Yield (metric tons/hectare)")
p + theme_classic()

p
```



AKILIMO

ACAI

Boxplot by factor using colors:



Boxplot by factor: divided into several panels

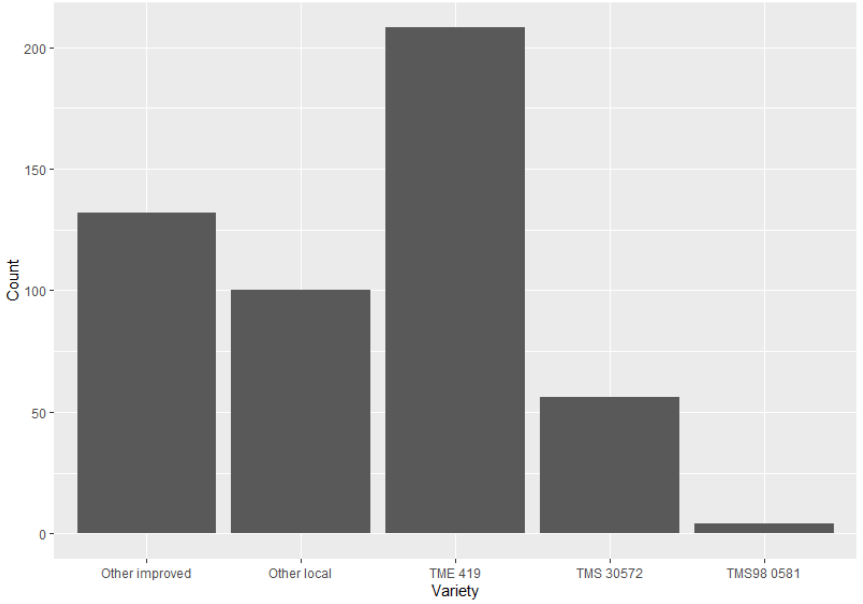
Barplot

- A barplot can be plotted using geom_bar(). A barplot is a tool to visualize the distribution of a qualitative variable.
- So, when you asses our data, what is(are) the qualitative(s) variable(s) we can use for barplot?

```
p <- ggplot(rti) +
aes(x = variety) +
geom_bar()
p <- p + labs(x = "Variety", y = "Count")
p + theme_classic()
```

The levels of variety are: "other_improved" "other_local" "TME419" "TMS30572"
"TMS98_0581". We can recode these using mutate() and case_when()

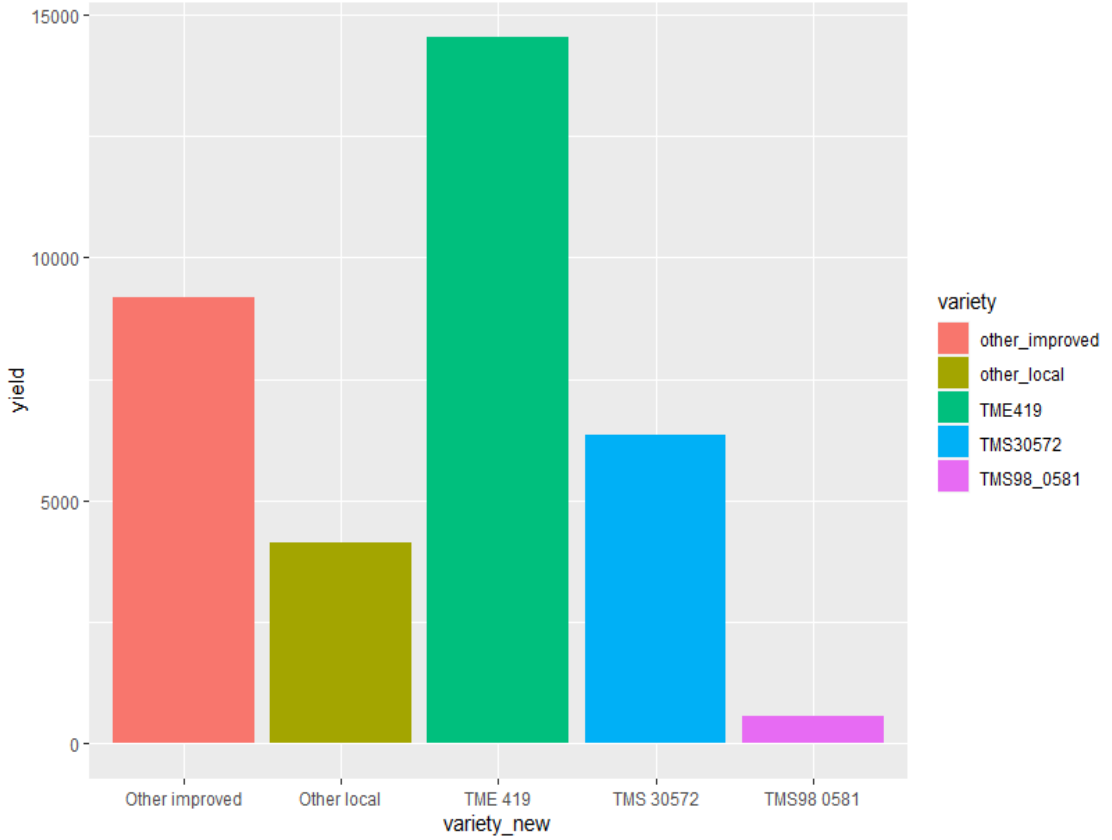
```
rti <- rti %>% mutate(
 variety_new = case_when(
  variety %in% "other_local"  ~ "Other local",
  variety %in% "other_improved" ~ "Other improved",
  variety %in% "TME419"  ~ "TME 419",
  variety %in% "TMS30572"  ~ "TMS 30572",
  variety %in% "TMS98_0581" ~ "TMS98 0581"
  )
 )
```



The output is rather dull. Let's asses the geom_bar() syntax to make it more presentable:

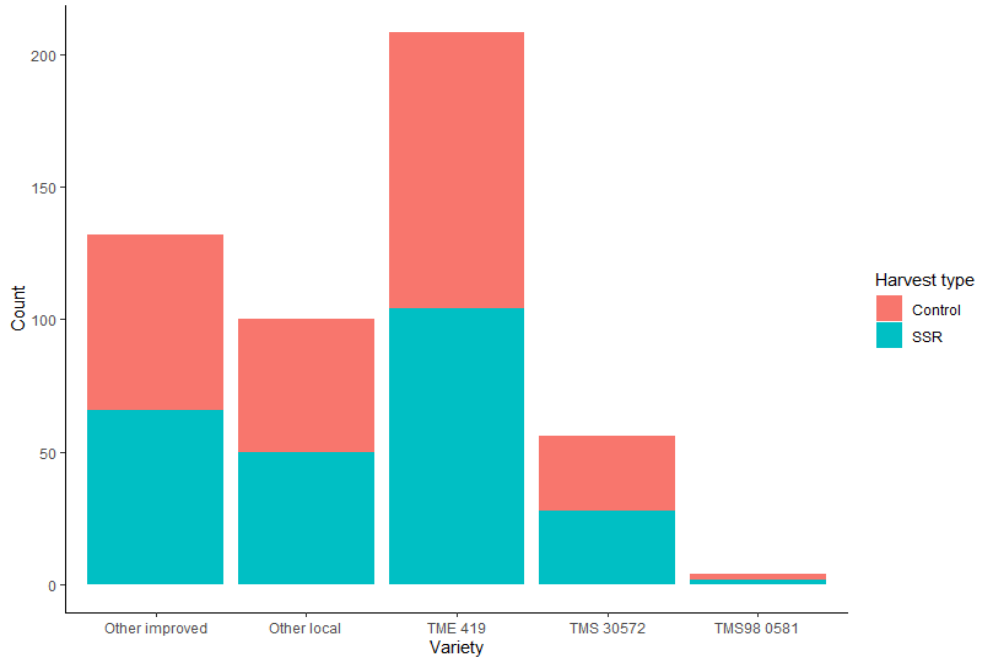
- geom_bar(stat, fill, color, width)
- Parameters :
- stat : Set the stat parameter to identify the mode.
- fill : Represents color inside the bars.
- color : Represents color of outlines of the bars.
- width : Represents width of the bars.

```
f <-ggplot(data=rti, aes(x=variety_new,
y=yield,fill=variety))+
  geom_bar(stat="identity")
f
```



Now let’s see a barplot with two qualitative variables

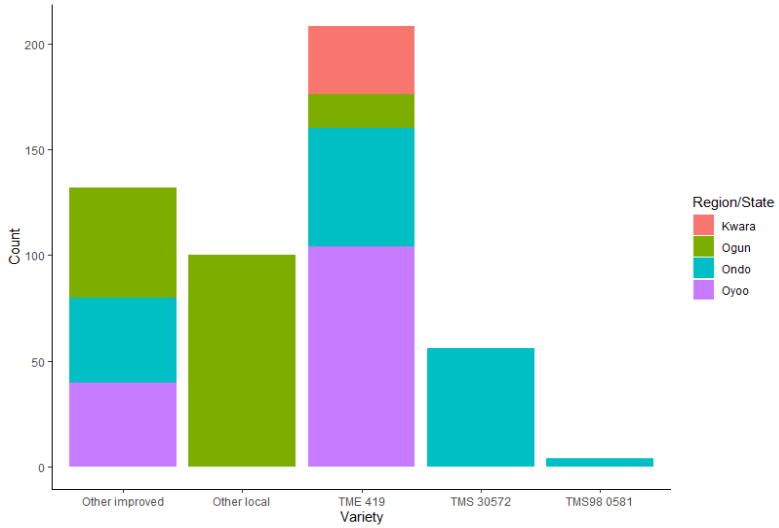
```
p <- ggplot(rti) +
  aes(x = variety_new, fill=factor(harvest)) +
  geom_bar()
p <- p + labs(x = "Variety", y = "Count")
p <- p + scale_fill_discrete(name="Year")
p + theme_classic()
```



Let's plot by region/state:

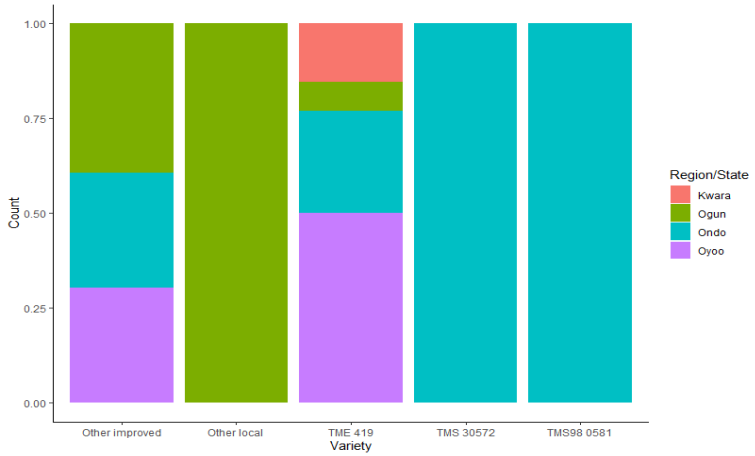
Example 1

```
p <- ggplot(rti) +
  aes(x = variety_new, fill=factor(region3)) +
  geom_bar()
p <- p + labs(x = "Variety", y = "Count")
p <- p + scale_fill_discrete(name="Region/State")
p + theme_classic()
```



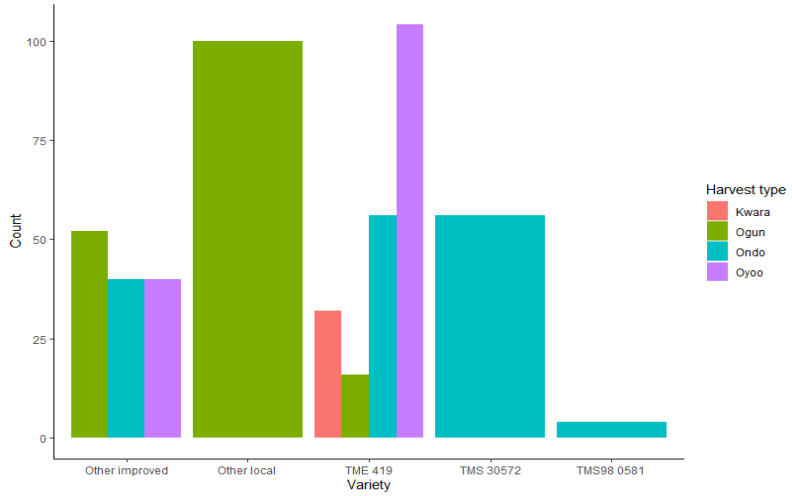
Example 2

```
p <- ggplot(rti) +
  aes(x = variety_new, fill=factor(region3 )) +
  geom_bar(position="fill")
p <- p + labs(x = "Variety", y = "Count")
p <- p + scale_fill_discrete(name="Region/State")
p + theme_classic()
```



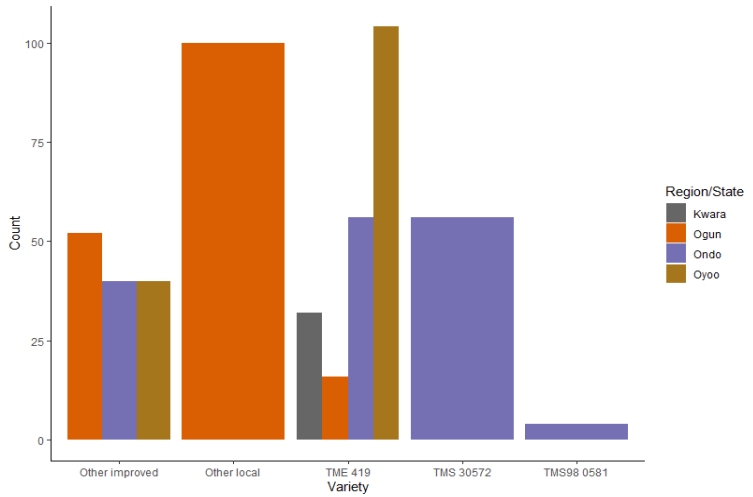
Here's a barplot with two qualitative variables:

```
#position dodge
p <- ggplot(rti) +
  aes(x = variety_new, fill=factor(harvest )) +
  geom_bar(position="dodge")
p <- p + labs(x = "Variety", y = "Count")
p <- p + scale_fill_discrete(name="Harvest type")
p + theme_classic()
```



Apply manual colors:

```
p <- ggplot(rti) +
  aes(x = variety_new, fill=factor(harvest )) +
  geom_bar(position="dodge")
p <- p + labs(x = "Variety", y = "Count")
p <- p + scale_fill_manual(values=c("#666666", "#D95F02", "#7570B3",
"#A6761D"),
                           name="Region/State")
p + theme_classic()
```



TASK

See example below and try to write a code that would give the output below: