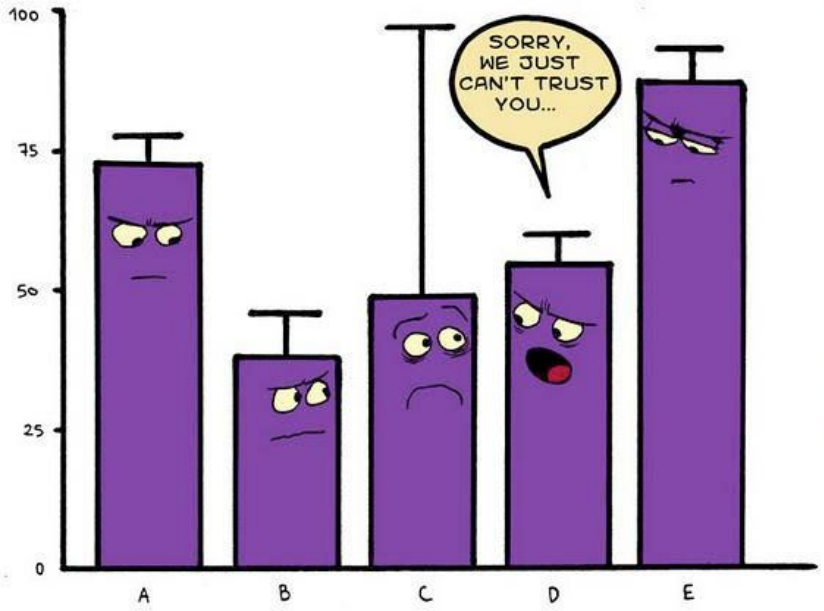
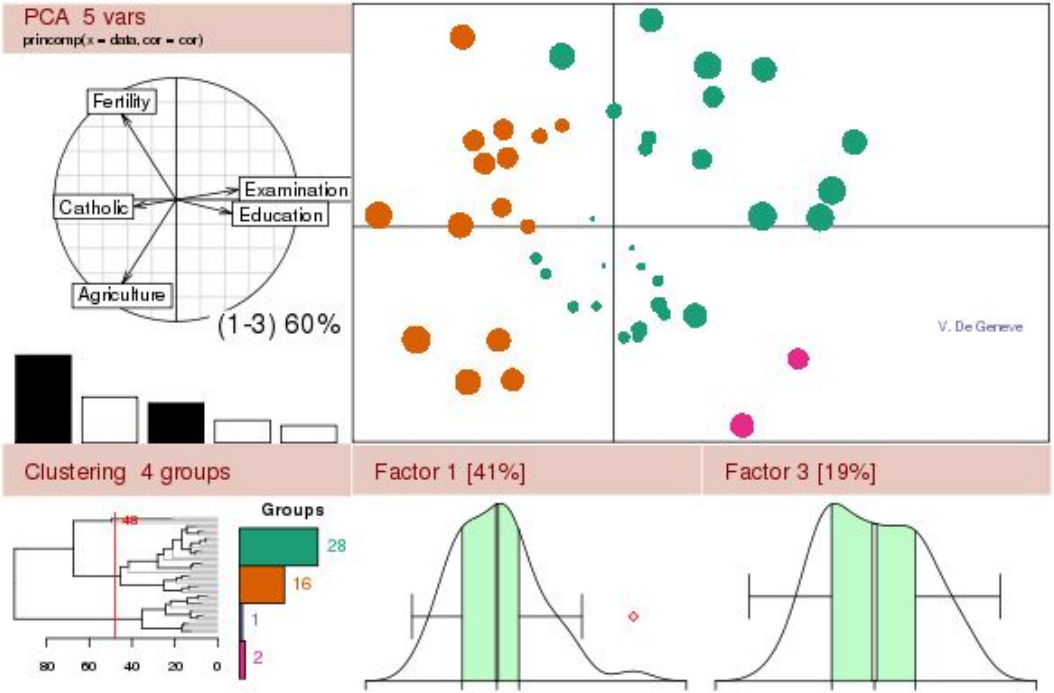
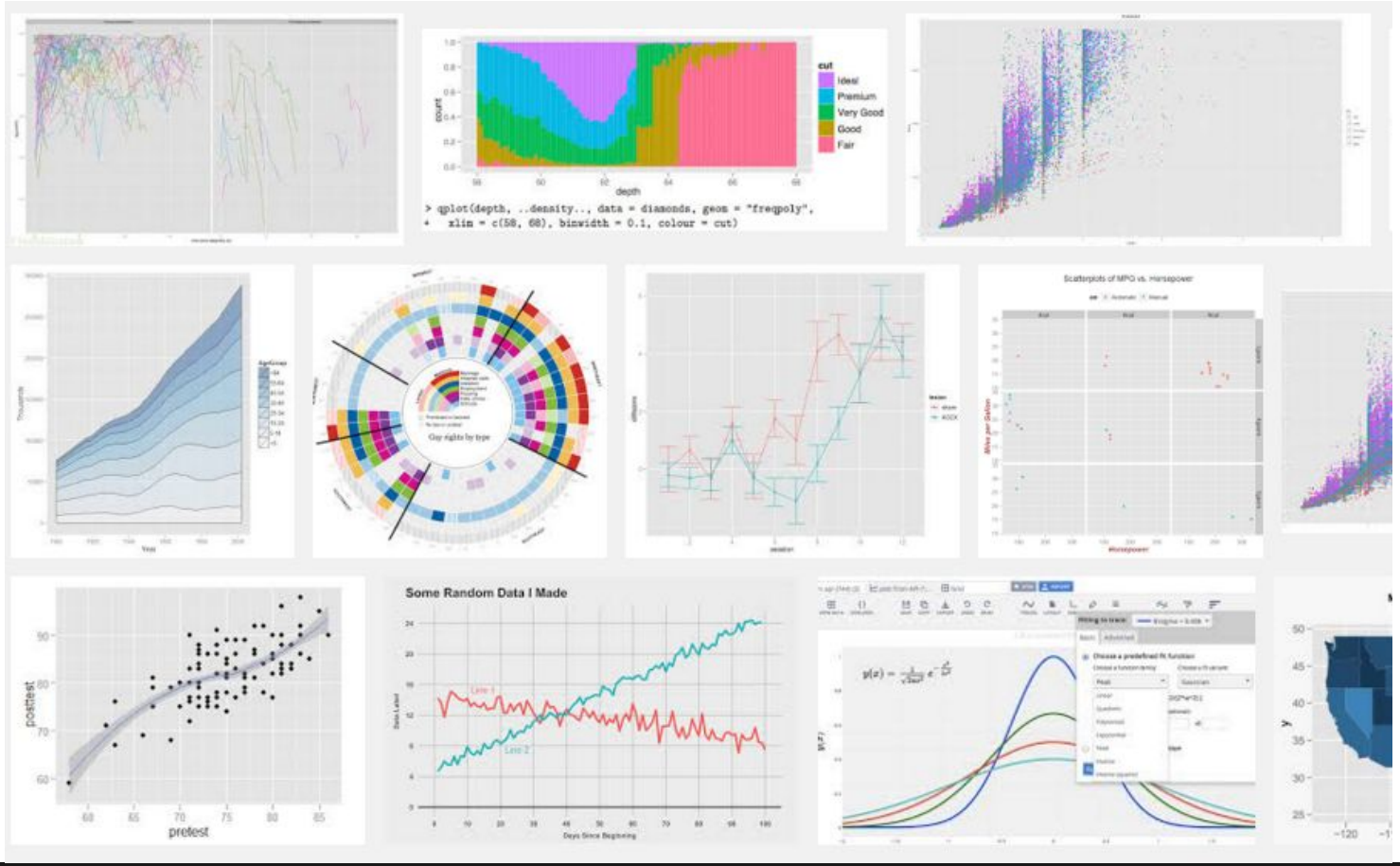
Presentación de Resultados

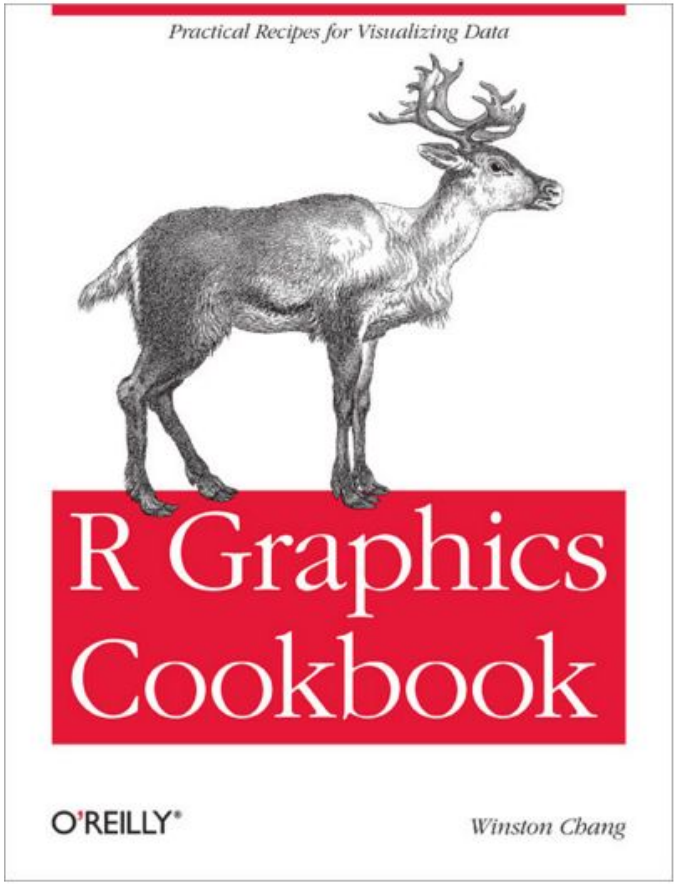




¿Por qué crear imágenes de tus datos?


cut
Ideal
Premium
Very Good
Good
Fair
depth
count
> qplot(depth, ..density.., data = diamonds, geom = "freqpoly",
+ xlim = c(58, 68), binwidth = 0.1, colour = cut)
Scatterplots of MPG vs. Horsepower
Some Random Data I Made
Line 1
Line 2
Days Since Beginning
pretest
posttest

ggplot2



Graphs with ggplot2

1. Bar and line graphs (ggplot2)
2. Plotting means and error bars (ggplot2)
3. Plotting distributions (ggplot2) - Histograms, density curves, boxplots
4. Scatterplots (ggplot2)
5. Titles (ggplot2)
6. Axes (ggplot2) - Control axis text, labels, and grid lines.
7. Legends (ggplot2)
8. Lines (ggplot2) - Add lines to a graph.
9. Facets (ggplot2) - Slice up data and graph the subsets together in a grid.
10. Multiple graphs on one page (ggplot2)
11. Colors (ggplot2)

1. **`geom_point()`** begins with the **`mpg`** data set

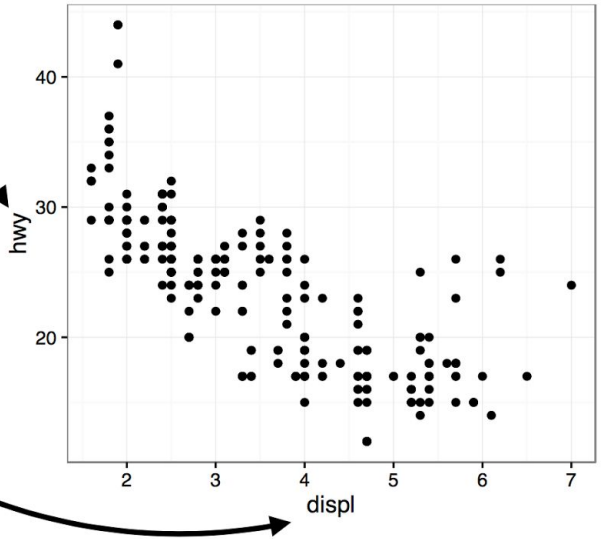
2. **`geom_point()`** transforms the data with the "identity" stat, which returns an identical copy of the data set.

3. **`geom_point()`** uses the identical copy to build the plot. displ is mapped to the x axis, hwy is mapped to the y axis.

manufacturer	model	year	displ	hwy	...
audi	a4	1999	1.8	29	...
audi	a4	1999	1.8	29	...
audi	a4	2008	2.0	31	...
audi	a4	2008	2.0	30	...
audi	a4	1999	2.8	26	...
...	...	...	...	...	...

stat_identity()

manufacturer	model	year	displ	hwy	...
audi	a4	1999	1.8	29	...
audi	a4	1999	1.8	29	...
audi	a4	2008	2.0	31	...
audi	a4	2008	2.0	30	...
audi	a4	1999	2.8	26	...
...	...	...	...	...	...



1. **`geom_bar()`** begins with the **`diamonds`** data set

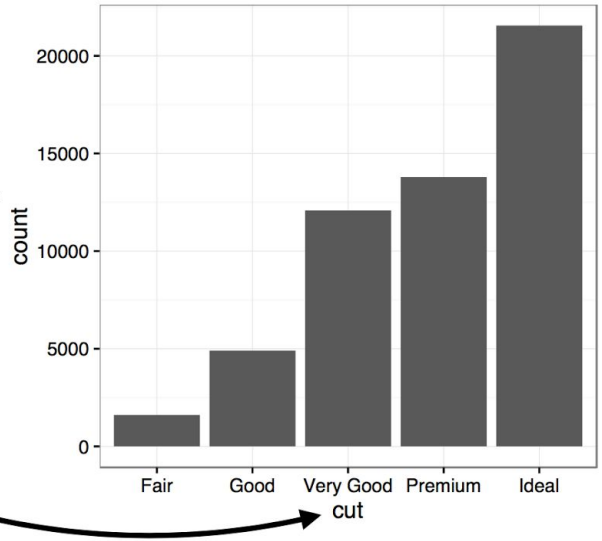
2. **`geom_bar()`** transforms the data with the "count" stat, which returns a data set of cut values and counts.

3. **`geom_bar()`** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

stat_count()

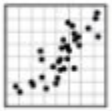
cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1



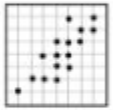
These geoms are useful for visualizing the relationship between two variables.

Continuous X, Continuous Y

geom_label(check_overlap = TRUE, nudge_x = 1, nudge_y = 1**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



geom_jitter(height = 2, width = 2**)**
x, y, alpha, color, fill, shape, size

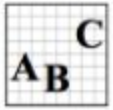


geom_point()
x, y, alpha, color, fill, shape, size, stroke

geom_quantile()
x, y, alpha, color, group, linetype, size, weight

geom_rug(sides = "bl"**)**
x, y, alpha, color, linetype, size

geom_smooth(method = lm, se = FALSE**)**
x, y, alpha, color, fill, group, linetype, size, weight



geom_text(check_overlap = TRUE**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Continuous Bivariate Distribution

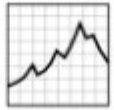
geom_bin2d(binwidth = c(5, 50)**)**
x, y, alpha, color, fill, linetype, size, weight

geom_density2d()
x, y, alpha, colour, group, linetype, size

geom_hex()
x, y, alpha, colour, fill, size

Continuous Function

geom_area()
x, y, alpha, color, fill, linetype, size



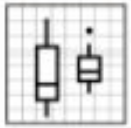
geom_line()
x, y, alpha, color, group, linetype, size

geom_step(direction = "hv"**)**
x, y, alpha, color, group, linetype, size

Discrete X, Continuous Y

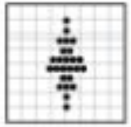
geom_bar(stat = "identity"**)**

x, y, alpha, color, fill, linetype, size, weight



geom_boxplot()

x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight



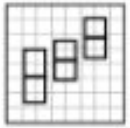
geom_dotplot(binaxis = "y", stackdir = "up"**)**

x, y, alpha, color, fill, group

geom_violin(scale = "area"**)**

x, y, alpha, color, fill, group, linetype, size, weight

Visualizing error

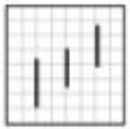


geom_crossbar(fatten = 2**)**

x, y, ymax, ymin, alpha, color, fill, group, linetype, size

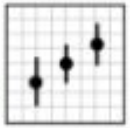
geom_errorbar()

x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)



geom_linerange()

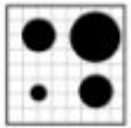
x, ymin, ymax, alpha, color, group, linetype, size



geom_pointrange()

x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

Discrete X, Discrete Y



geom_count()

x, y, alpha, color, fill, shape, size, stroke

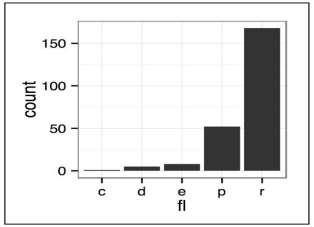
Maps

geom_map(map = map_data("state")**)**

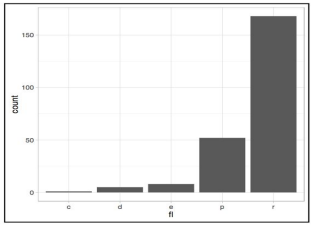
map_id, alpha, color, fill, linetype, size

Themes

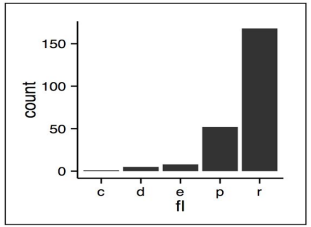
Theme functions change the appearance of your plot.



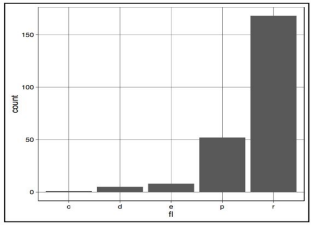
theme_bw()
White background with grid lines



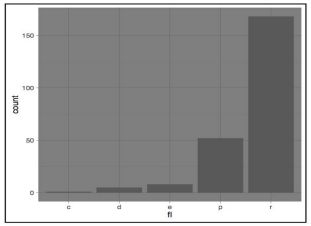
theme_light()
Light axes and grid lines



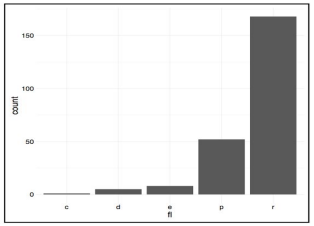
theme_classic()
Classic theme, axes but no grid lines



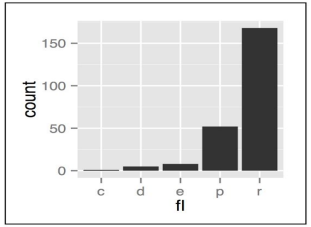
theme_linedraw()
Only black lines



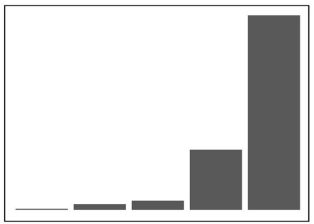
theme_dark()
Dark background for contrast



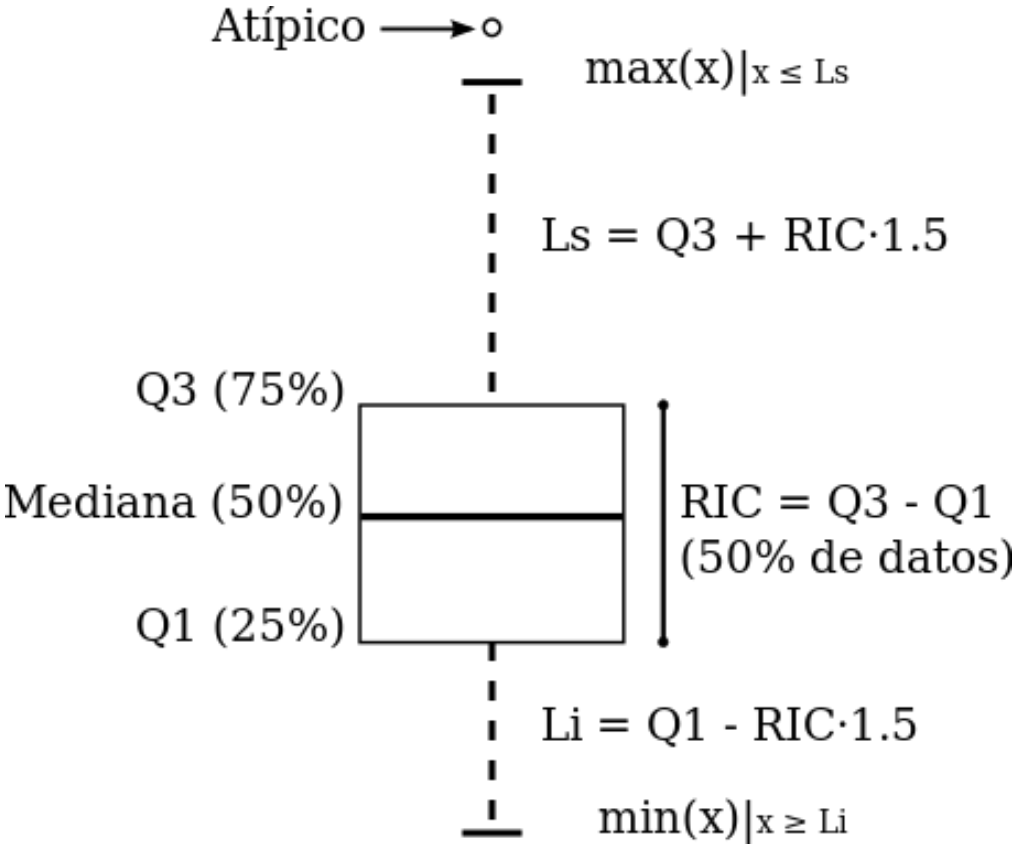
theme_minimal()
Minimal theme, no background



theme_gray()
Grey background (default theme)



theme_void()
Empty theme, only geoms are visible


Atípico
max(x)|x ≤ Ls
Ls = Q3 + RIC·1.5
Q3 (75%)
Mediana (50%)
Q1 (25%)
RIC = Q3 - Q1
(50% de datos)
Li = Q1 - RIC·1.5
min(x)|x ≥ Li

Boxplot

SE TRATA DE DIVIDIR LOS DATOS EN CUATRO GRUPOS IGUALES Y OBSERVAR LA DISTANCIA QUE SEPARA LOS GRUPOS EXTREMOS.

EL RECORRIDO INTERCUARTÍLICO (IQR) ES LA DISTANCIA (O DIFERENCIA) QUE HAY ENTRE ELLOS:

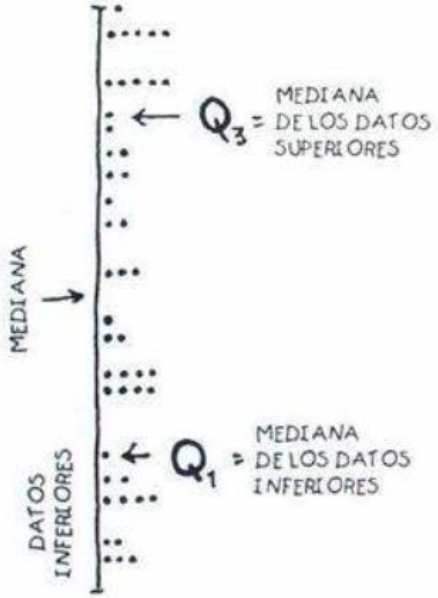
$$IQR = Q_3 - Q_1$$

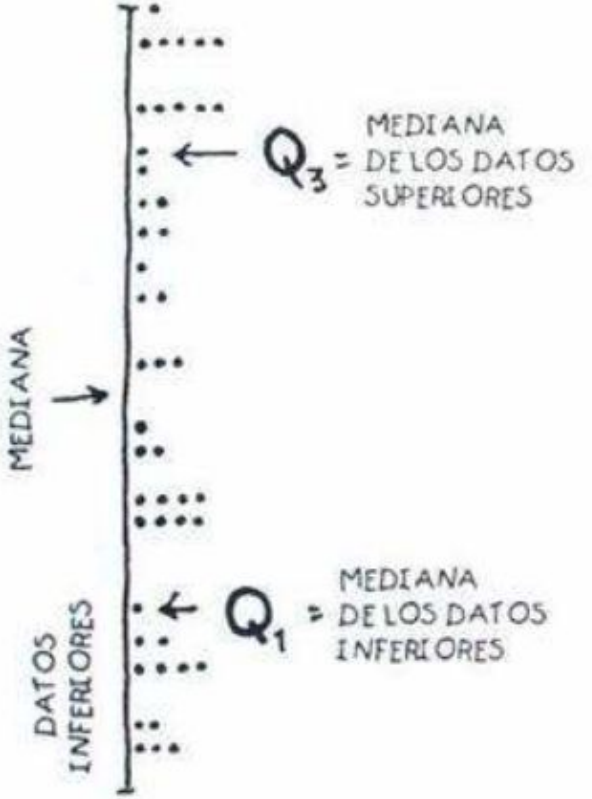
1) ORDENA LOS DATOS NUMÉRICAMENTE.

2) DIVIDE LOS DATOS POR LA MEDIANA EN DOS GRUPOS IGUALES (SI LA MEDIANA COINCIDE CON UN DATO, INCLÚYELO EN LOS DOS GRUPOS).

3) CALCULA LA MEDIANA DEL GRUPO INFERIOR. ÉSE ES EL PRIMER CUARTIL, O Q_1.

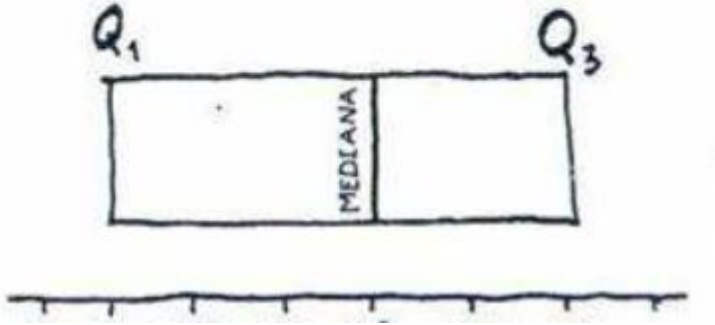
4) LA MEDIANA DEL GRUPO SUPERIOR ES EL TERCER CUARTIL, O Q_3.



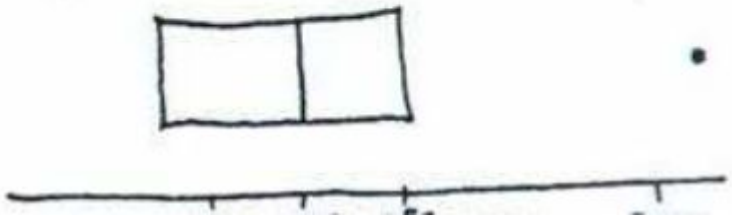


Recorrido Intercuartílico

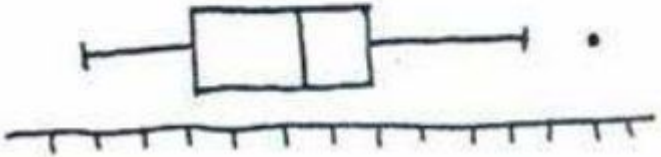
JOHN TUKEY INVENTÓ OTRO TIPO DE REPRESENTACIÓN PARA MOSTRAR EL IQR. EL GRÁFICO DE CAJA. LOS EXTREMOS DE LA CAJA SON LOS CUARTILES Q_1 Y Q_3. LA MEDIANA SE DIBUJA DENTRO DE LA CAJA.



SI UN PUNTO SE ENCUENTRA A MÁS DE 1,5 IQR DE LOS EXTREMOS DE LA CAJA, SE CONSIDERA QUE ES UNA OBSERVACIÓN ATÍPICA, Y SE REPRESENTA INDIVIDUALMENTE.



POR ÚLTIMO, EXTENDEMOS LÍNEAS HASTA LOS PUNTOS MÁS ALEJADOS QUE NO SON OBSERVACIONES ATÍPICAS (ES DECIR, QUE SE ENCUENTRAN A MENOS DE 1,5 IQR DE LOS CUARTILES).

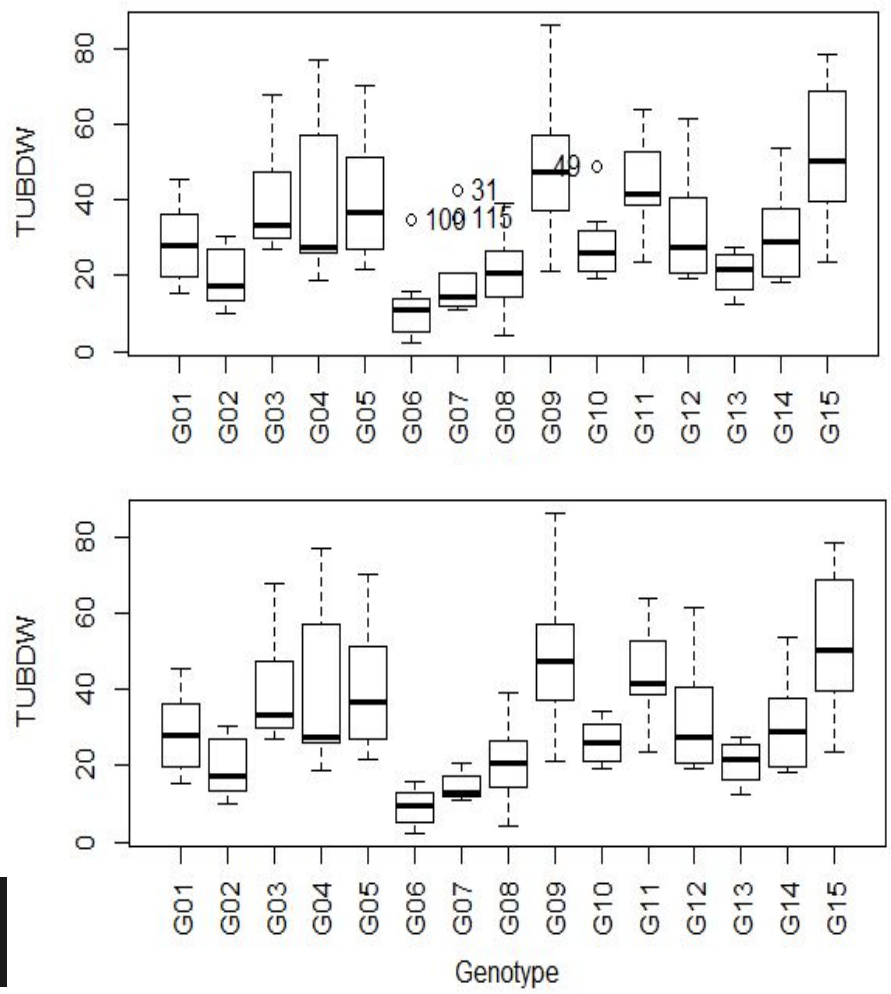


Boxplot

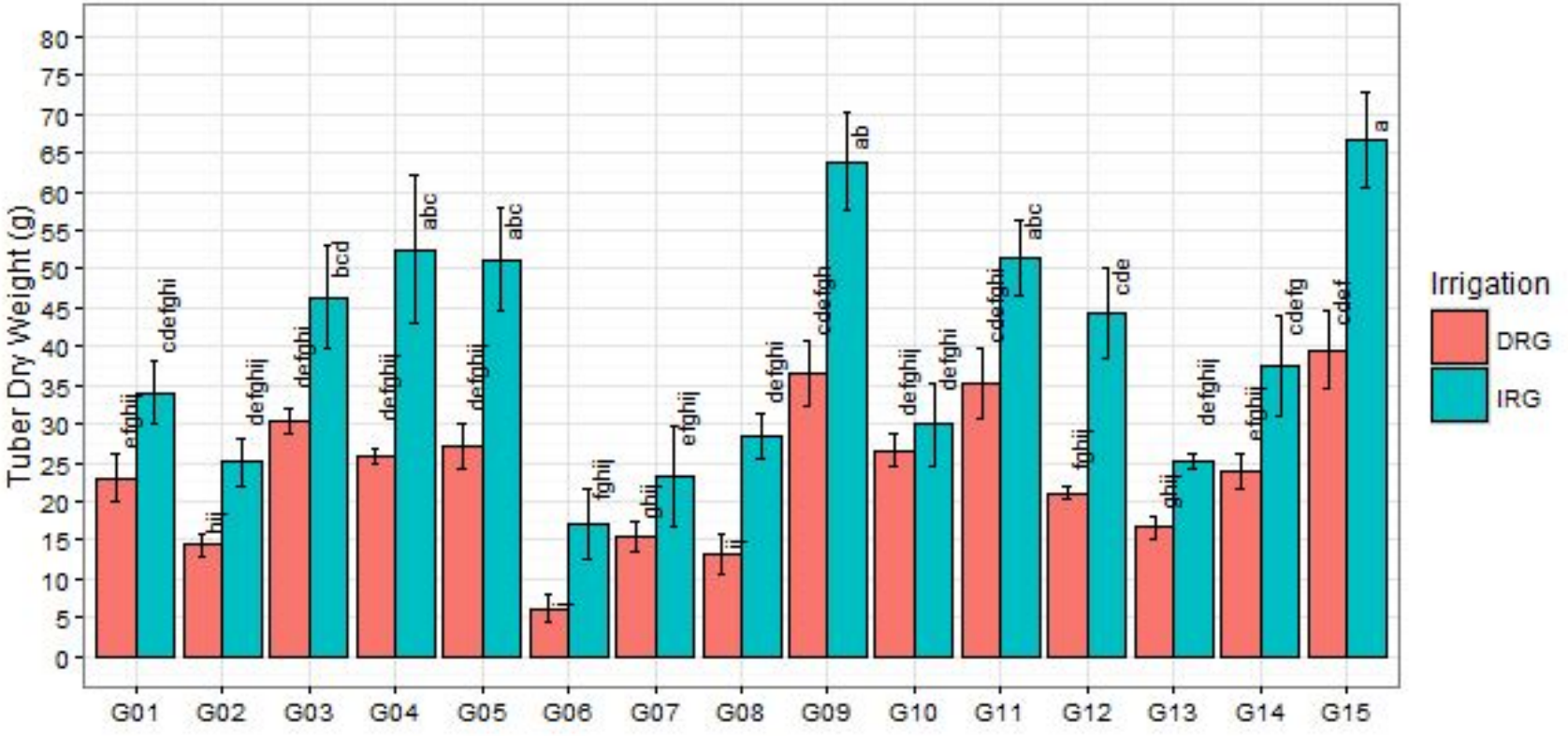
RClass.R

Source on Save | Run | Source

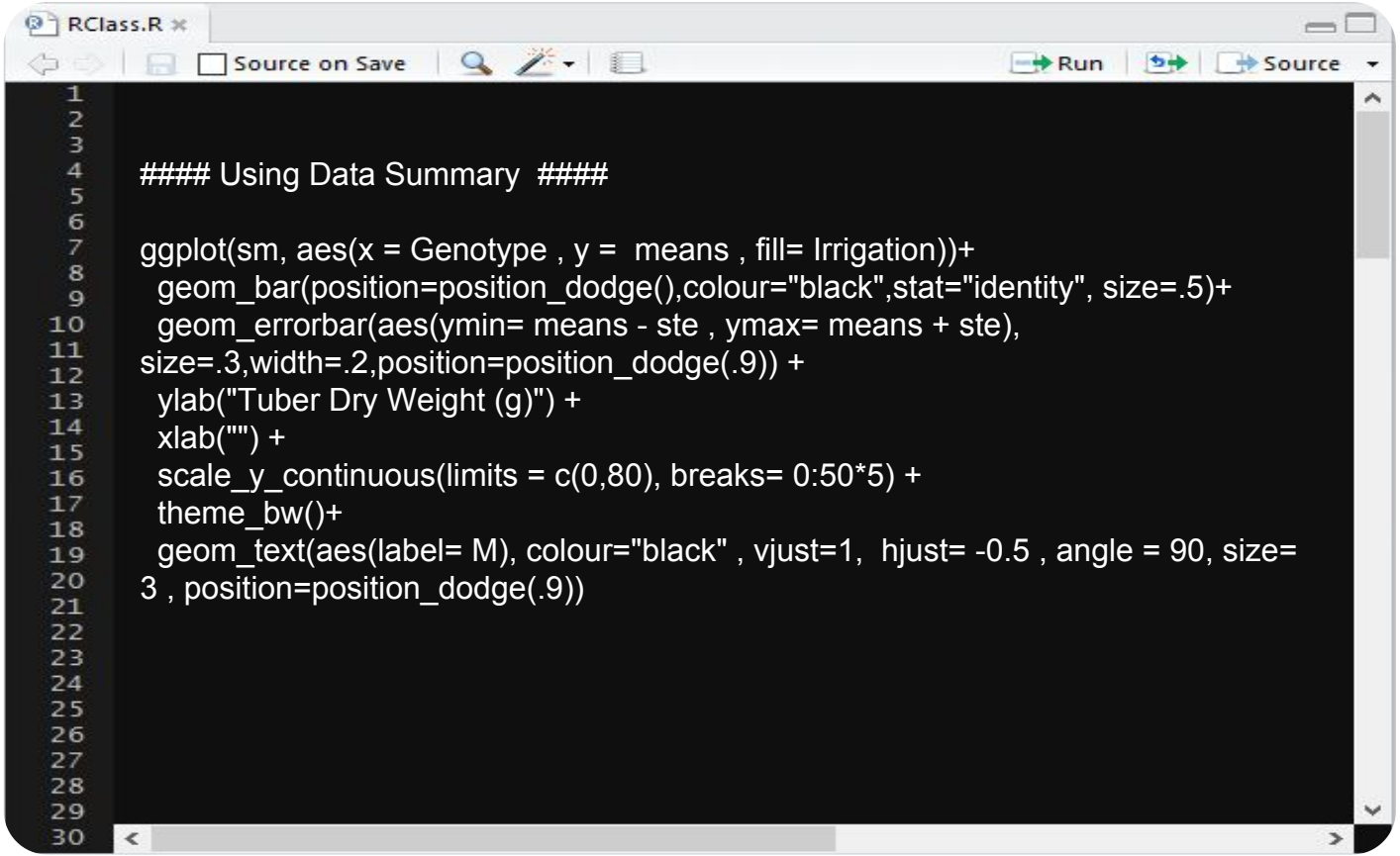
```
#### Using Linear Model  ####

bp <- Boxplot( TUBDW  ~ Genotype, fb, las = 3, id.n=Inf )

fb$TUBDW[as.numeric(bp)]  <- NA

bp <- Boxplot( TUBDW  ~ Genotype, fb, las = 3, id.n=Inf )
```



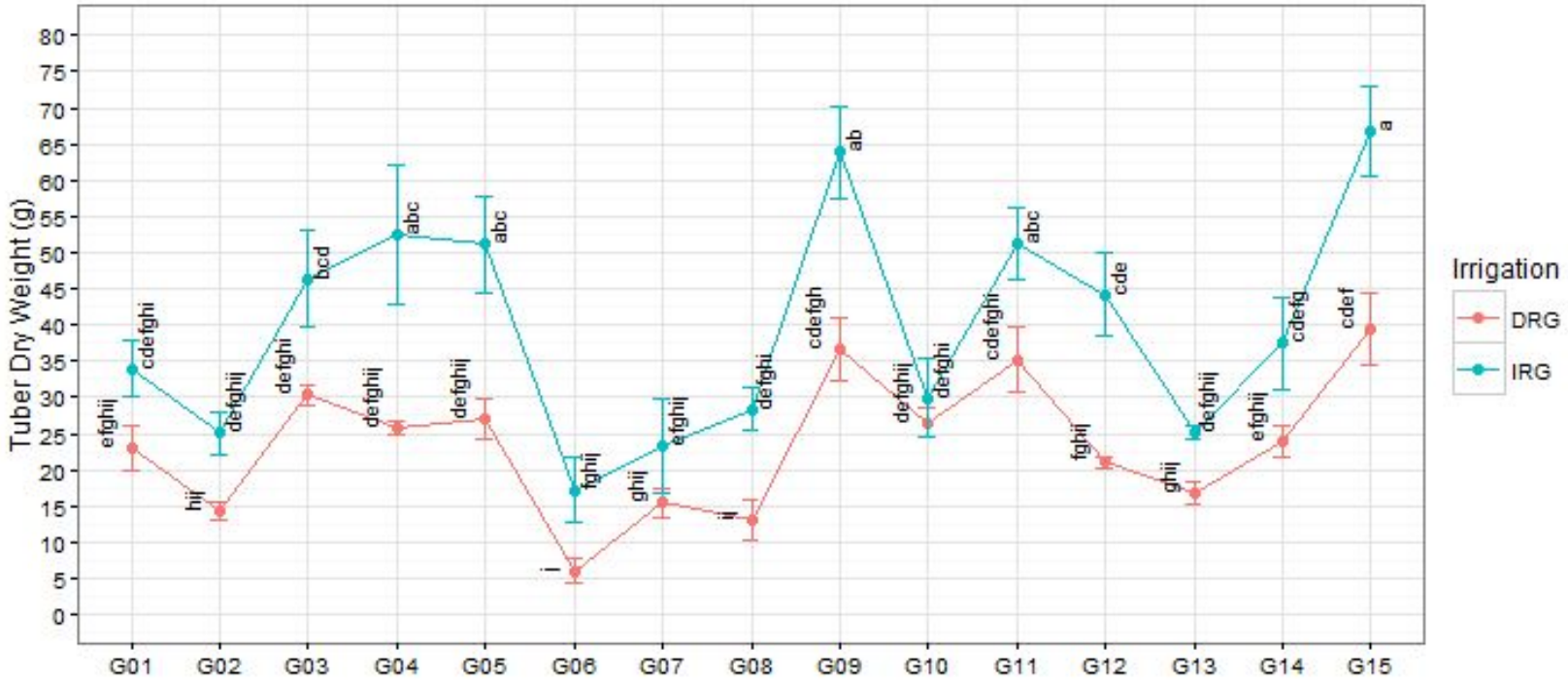
Outlier remove


Tuber Dry Weight (g)
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
efghij
hij
defghij
defghij
defghij
defghij
abc
abc
ij
fghij
ghij
efghij
defghi
cdefgh
ab
defghij
defghi
cdefghi
abc
fghij
cde
ghij
defghij
efghij
cdefg
cdef
a
G01
G02
G03
G04
G05
G06
G07
G08
G09
G10
G11
G12
G13
G14
G15
Irrigation
DRG
IRG

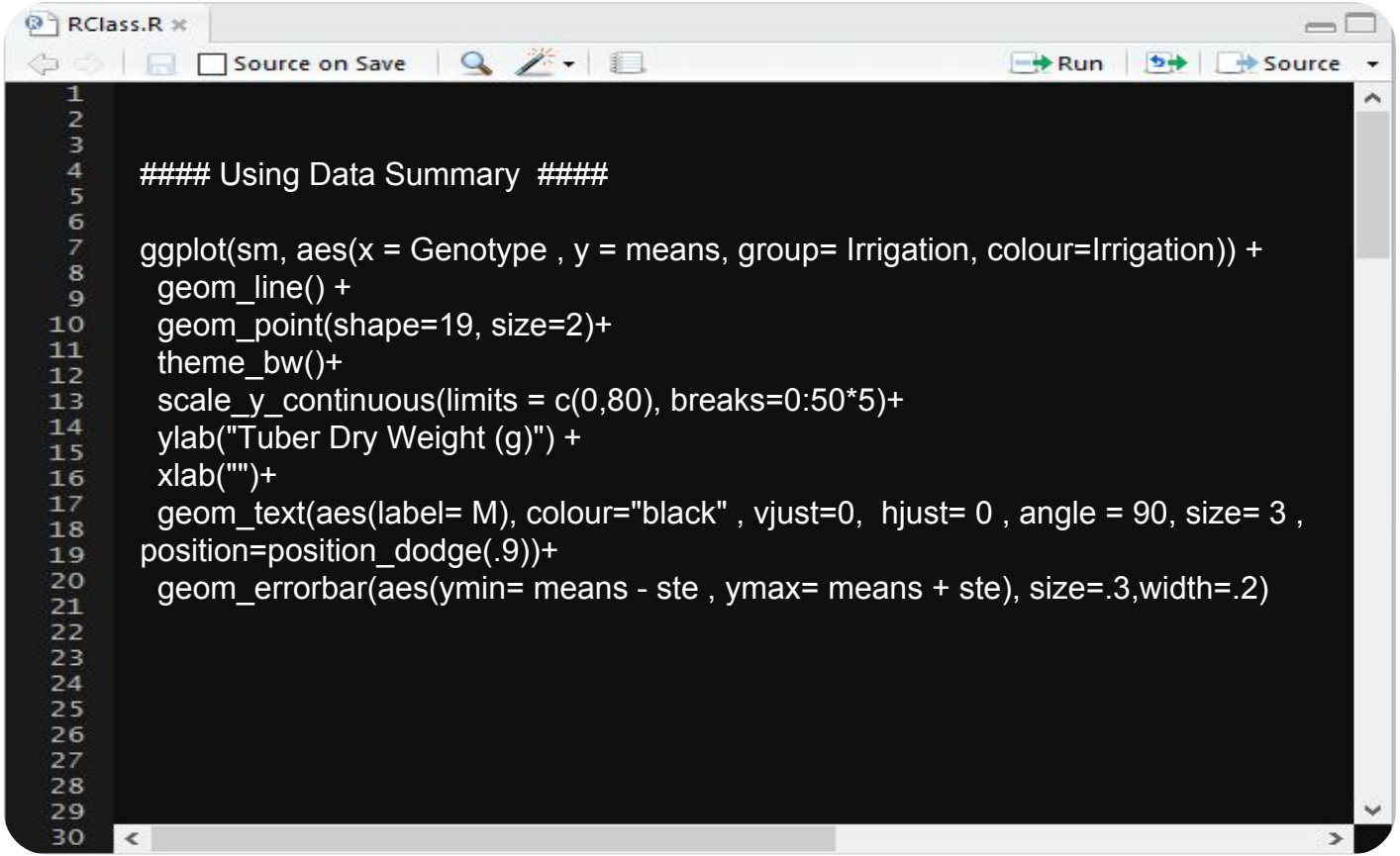
Bar Plot



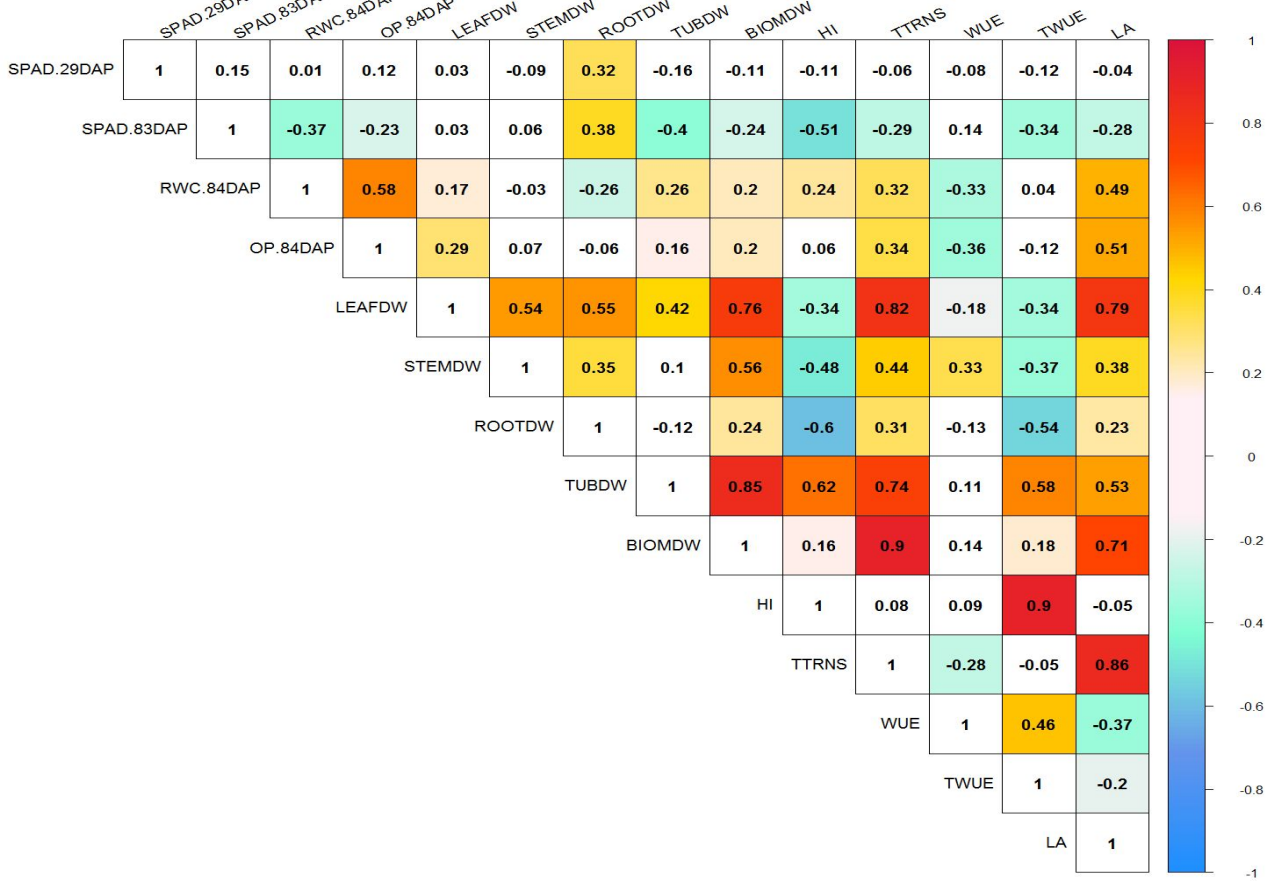
Bar Plot Code


Tuber Dry Weight (g)
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
G01
G02
G03
G04
G05
G06
G07
G08
G09
G10
G11
G12
G13
G14
G15
efghij
cdefghi
hij
defghij
defghi
bcd
defghij
abc
defghij
abc
i
fghij
ghij
efghij
ii
defghi
cdefgh
ab
defghij
defghi
cdefghi
abc
fghij
cde
ghij
defghij
efghij
cdefg
cdef
a
Irrigation
DRG
IRG

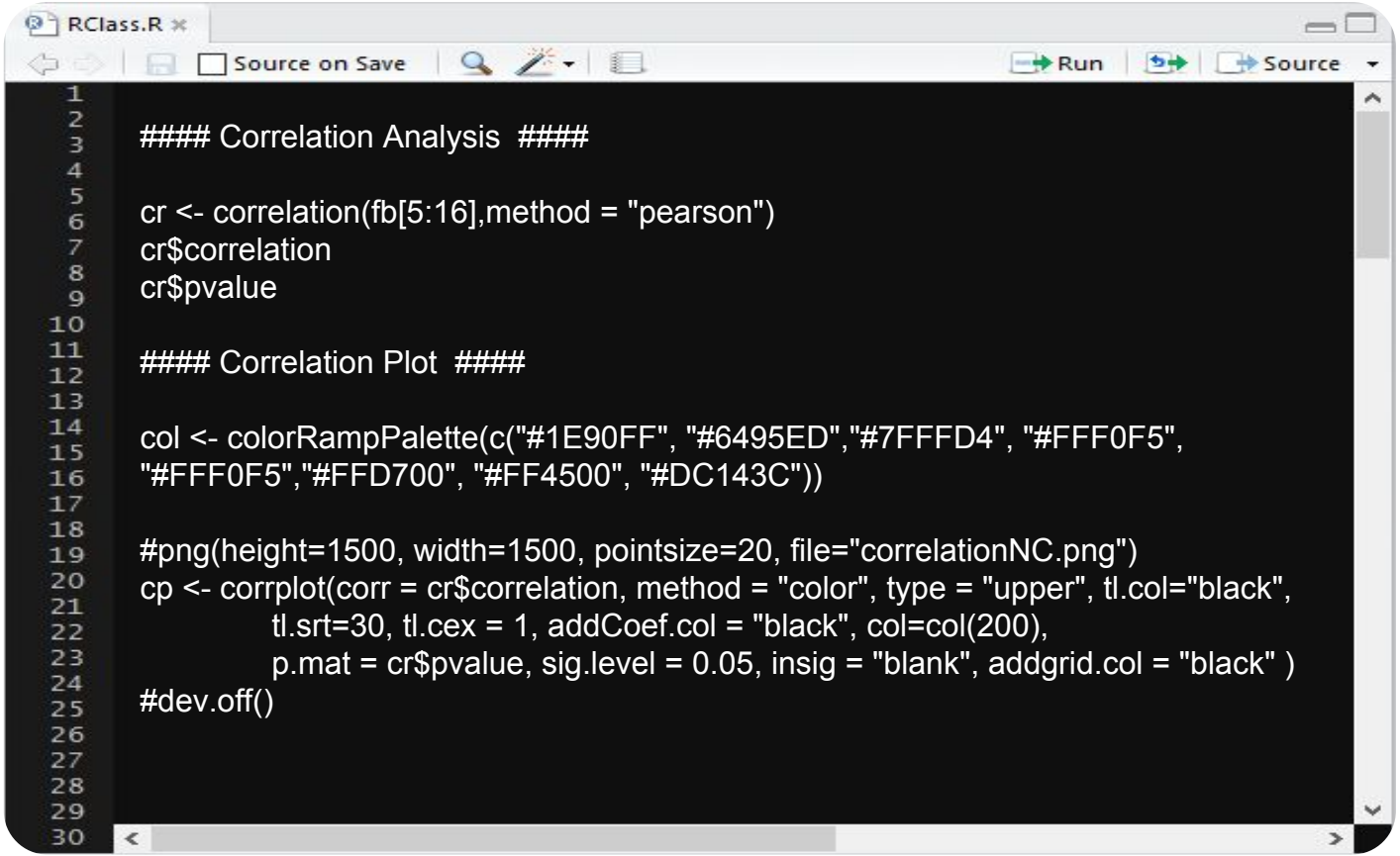
Line Plot



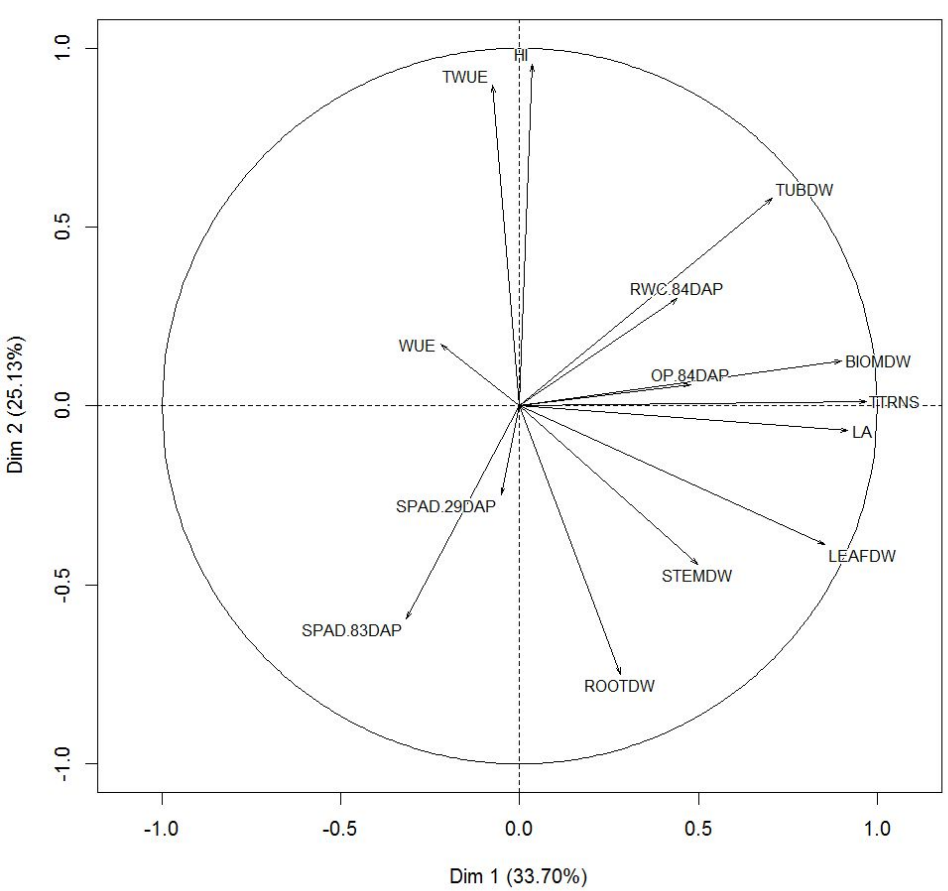
Line Plot Code

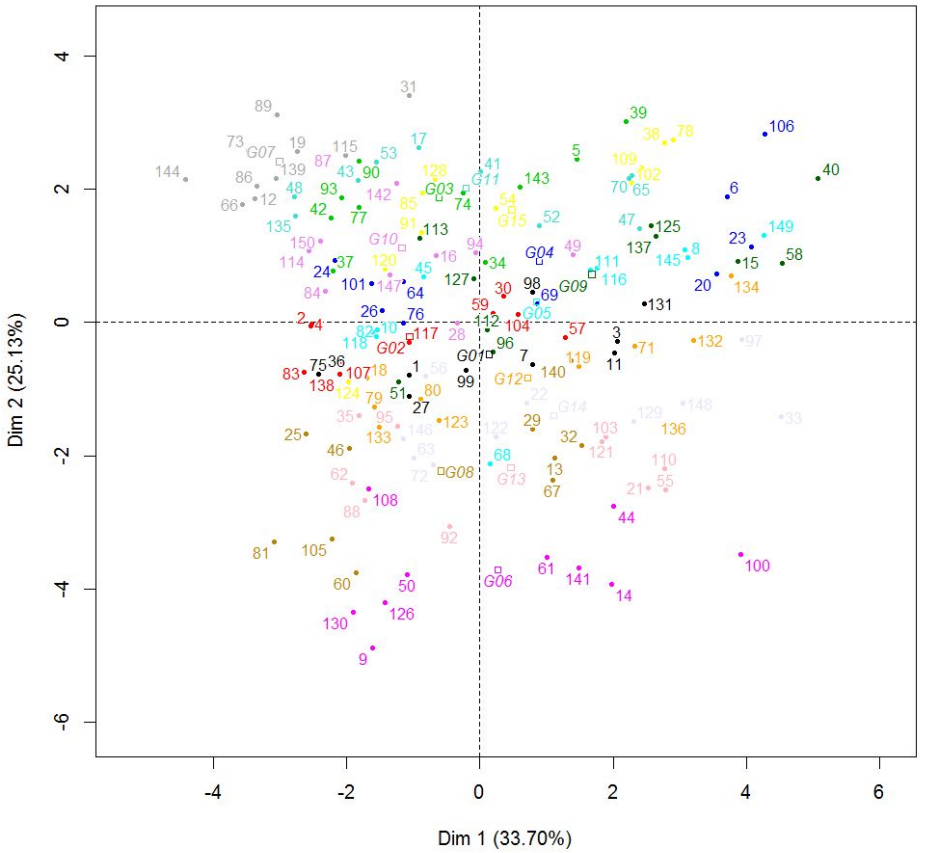


Correlation Plot

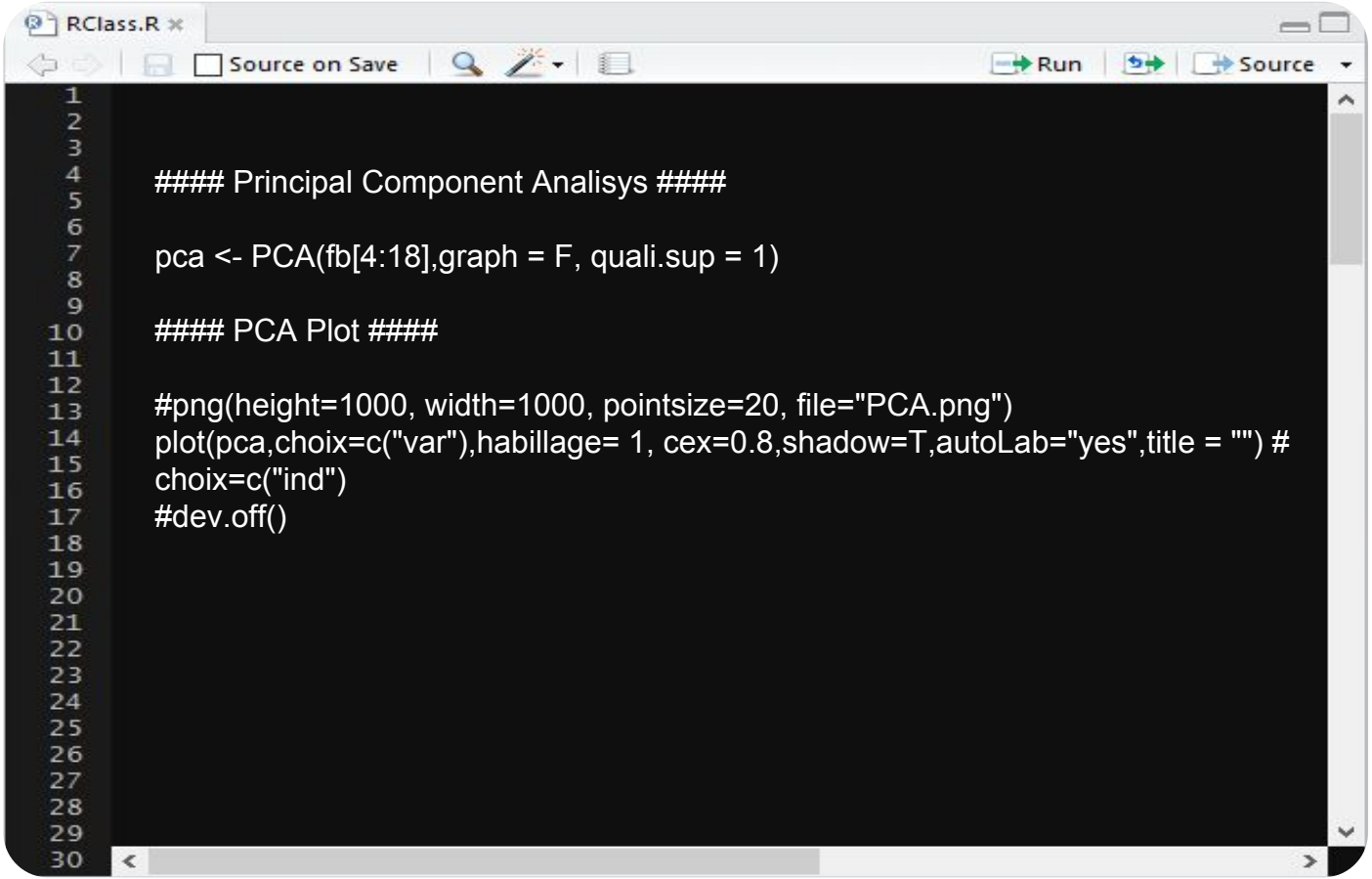


Correlation Plot Code

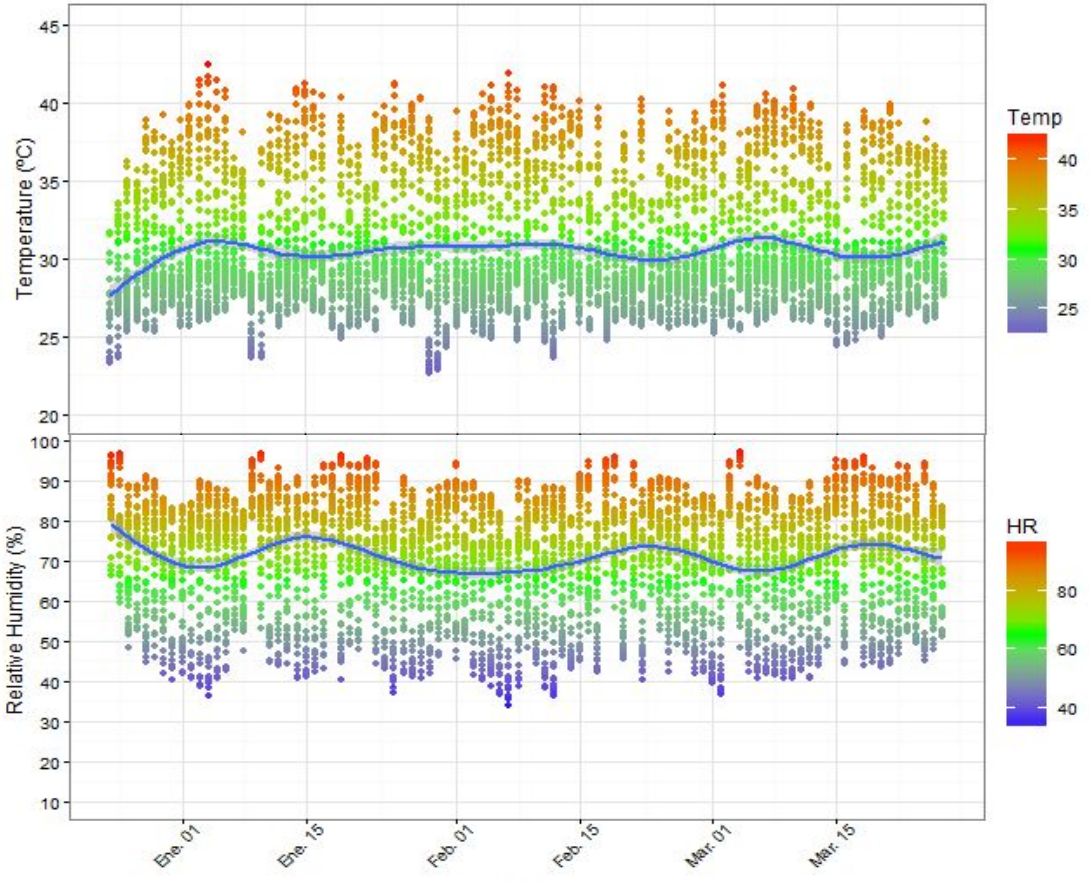




Principal Component Analysis (PCA)


RClass.R
Source on Save
Run
Source
Principal Component Analisys
pca <- PCA(fb[4:18],graph = F, quali.sup = 1)
PCA Plot
#png(height=1000, width=1000, pointsize=20, file="PCA.png")
plot(pca,choix=c("var"),habillage= 1, cex=0.8,shadow=T,autoLab="yes",title = "") #
choix=c("ind")
#dev.off()

PCA code



Meteorological Information Plot

RClass.R

Source on Save | Run | Source

```
#### Meteorological Data  ####

url <- "https://docs.google.com/spreadsheets/d/1sfy6XaCAYKqU1siOr9VP_I5wAzRBjjMmdr1nK387J0s/edit#gid=225298935"
md <- gsheet2tbl(url)
md <- as.data.frame(md)
md$DATE <- as.Date(md$DATE, format = "%m/%d/%y")
str(md)

ggplot(md, aes(x = DATE, y = Temp)) +
  geom_point(aes(colour = Temp)) +
  scale_colour_gradient2(low = "blue", mid = "green" , high = "red", midpoint = 31) +
  geom_smooth() +
  scale_y_continuous(limits = c(20,45), breaks=0:100*5) +
  theme_bw()+
  theme(axis.text.x = element_text(angle = 45, hjust = 0.8))+
  xlab("Date") +  ylab ("Temperature (ºC)")
```

Temperature & Humidity Plot Code