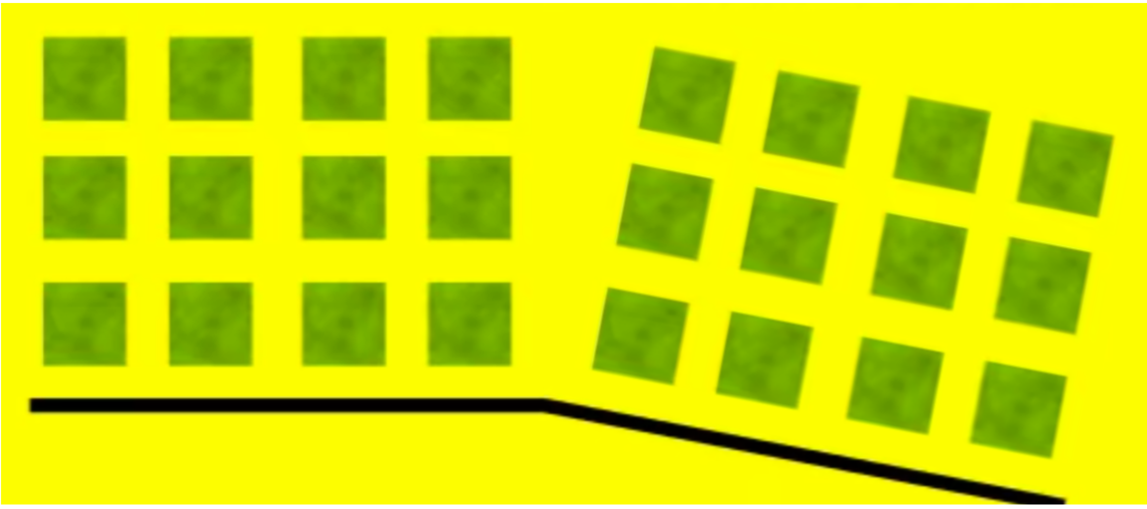
Two Factor Model : An Agricultural Example

Ananda Biswas

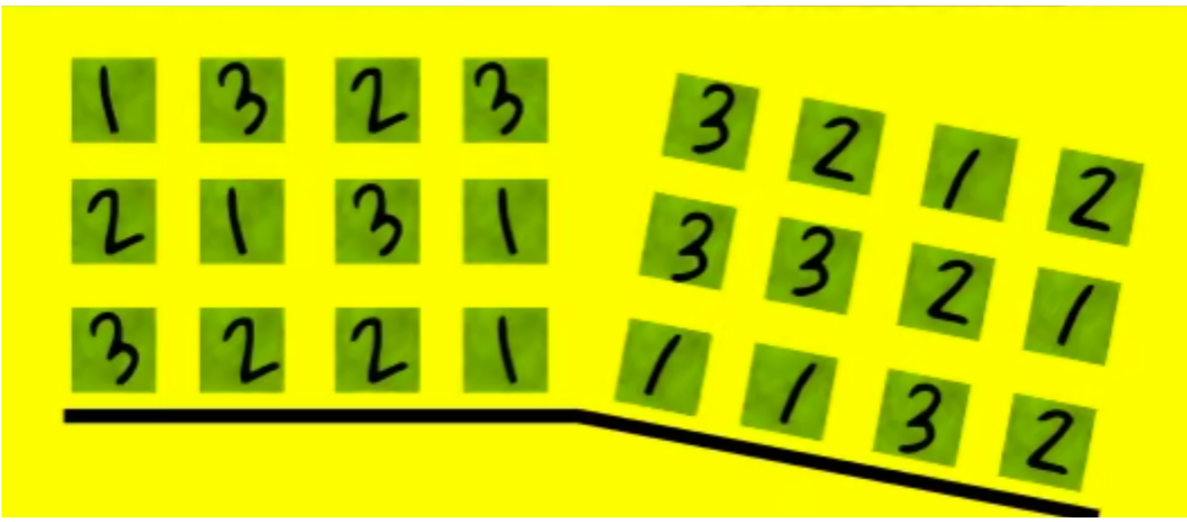
Suppose in a hilly area, we have 24 plots. The plots are as identical as possible. 12 of them are in a plain region and rest 12 of them are in a tilted region. We have 3 varieties of a crop and we want to study their yields and how it is affected by slope.



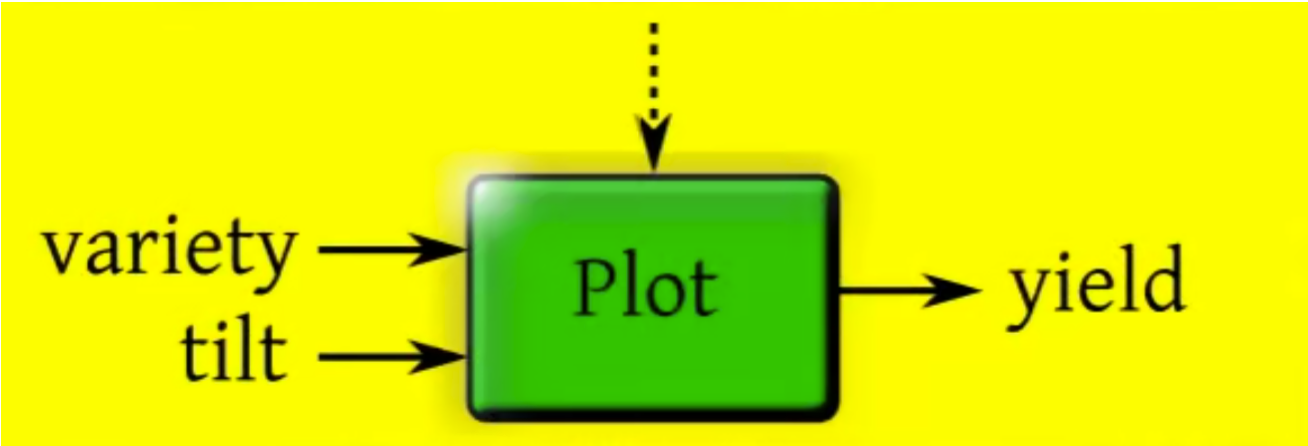
Among the 12 plots in the plain region, we randomly select 4 plots and sow seeds of variety **1**, then we again randomly select another 4 plots and sow seeds of variety **2** and in the rest 4 plots we sow seeds of variety **3**.

We do the same thing for the 12 plots in tilted region.

The allocation of various varieties in the plots is as follows :



Here our blackbox diagram is :



Here our linear model will be :

$$y_{ijk} = \alpha_i + \beta_j + \epsilon_{ijk}$$

where $\vec{\epsilon} \sim (\vec{0}, \sigma^2 I)$;

i is the index for variety;

j is the index for region (plane or tilted);

k is the index for plot of certain variety in certain region.

Here $i = 1, 2, 3$, $j = 1, 2$ and $k = 1, 2, 3, 4$.

Two Factor Model without Interaction (Additive Model)

Let us have a data set of yield of 3 varieties of paddy IR8, Jaya, Taichung in plain and slopy regions.

```
getwd()

## [1] "D:/Programming Languages/R/Linear Statistical Models - Arnab Chakraborty/005"
```

```
agri_dat = read.csv("agriculture_dataset.csv", as.is = FALSE)

agri_dat

##     variety  tilt yield
## 1       IR8 plain 254.2
## 2       IR8 plain 253.9
## 3       IR8 plain 254.4
## 4       IR8 plain 254.2
## 5       IR8 plain 254.0
## 6       IR8 slope 261.0
## 7       IR8 slope 260.4
## 8       IR8 slope 261.1
```

```
## 9       IR8 slope 260.9
## 10      IR8 slope 261.1
## 11     Jaya plain 264.5
## 12     Jaya plain 264.7
## 13     Jaya plain 264.3
## 14     Jaya plain 264.2
## 15     Jaya plain 264.3
## 16     Jaya slope 270.7
## 17     Jaya slope 271.3
## 18     Jaya slope 270.6
## 19     Jaya slope 271.2
## 20     Jaya slope 270.7
## 21 Taichung plain 284.2
## 22 Taichung plain 284.6
## 23 Taichung plain 284.6
## 24 Taichung plain 285.0
## 25 Taichung plain 284.4
## 26 Taichung slope 291.1
## 27 Taichung slope 291.1
## 28 Taichung slope 291.2
## 29 Taichung slope 291.2
## 30 Taichung slope 291.5
```

```
dim(agri_dat)
```

```
## [1] 30  3
```

```
names(agri_dat)
```

```
## [1] "variety" "tilt"    "yield"
```

```
head(agri_dat)
```

```
##   variety  tilt yield
## 1     IR8 plain 254.2
## 2     IR8 plain 253.9
## 3     IR8 plain 254.4
## 4     IR8 plain 254.2
## 5     IR8 plain 254.0
## 6     IR8 slope 261.0
```

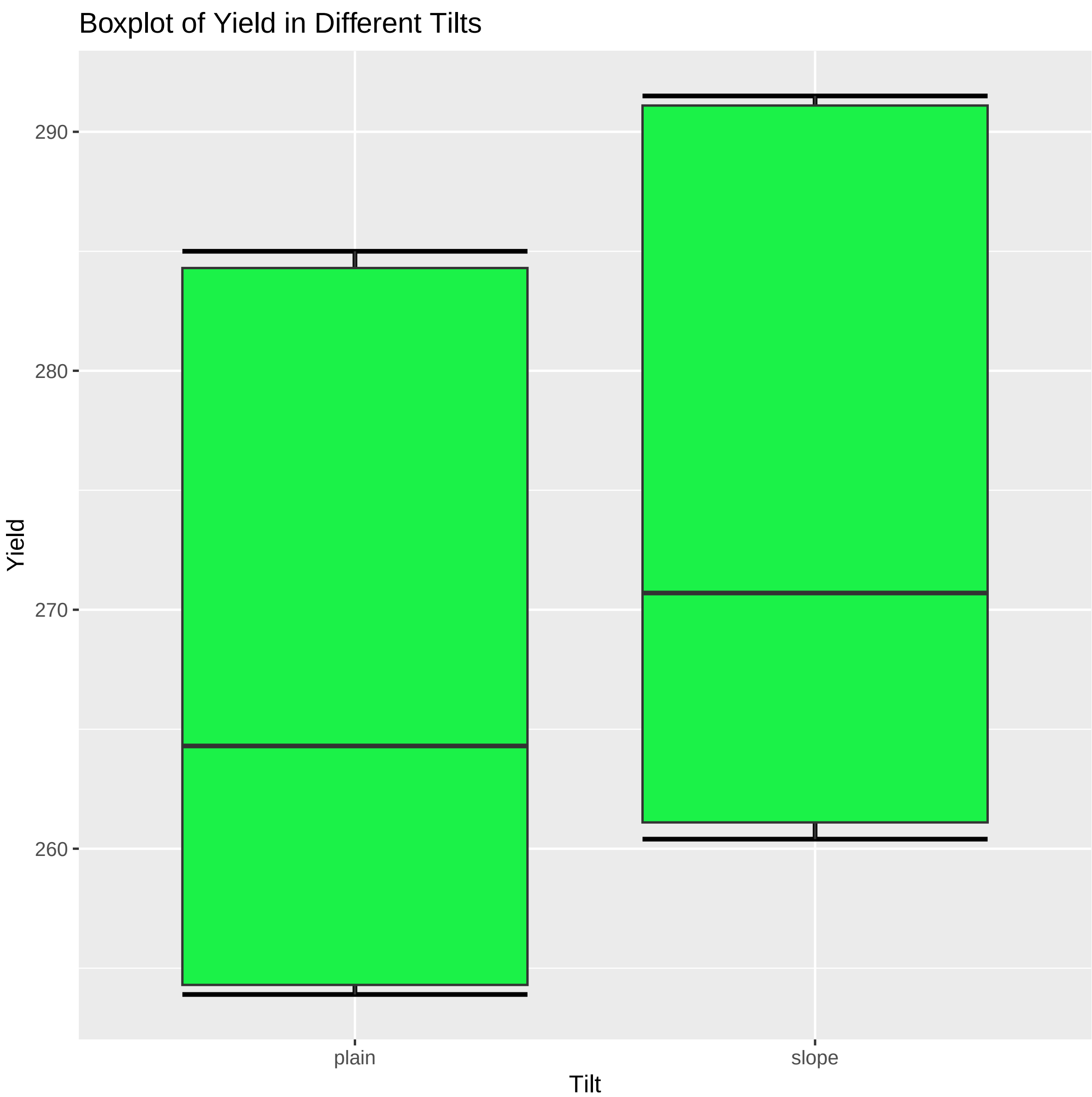
```
tail(agri_dat)
```

```
##     variety  tilt yield
## 25 Taichung plain 284.4
## 26 Taichung slope 291.1
## 27 Taichung slope 291.1
## 28 Taichung slope 291.2
## 29 Taichung slope 291.2
## 30 Taichung slope 291.5
```

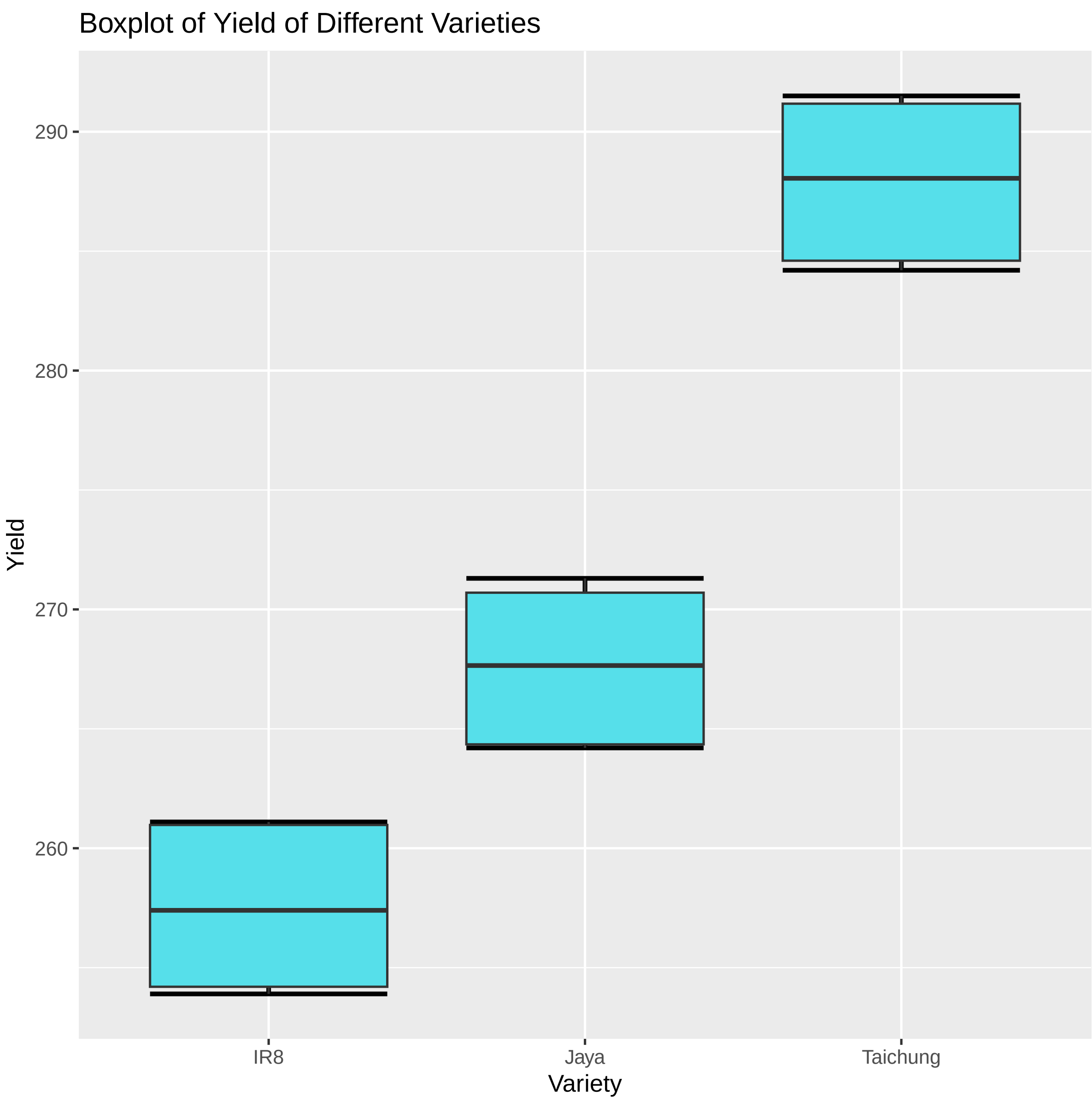
Setting *as.is = FALSE* tells R to read the strings in the csv file a factors, not characters.

```r
library(tidyverse)

## Warning:  package 'tidyverse' was built under R version 4.2.3
## Warning:  package 'ggplot2' was built under R version 4.2.2
## Warning:  package 'tibble' was built under R version 4.2.3
## Warning:  package 'tidyr' was built under R version 4.2.3
## Warning:  package 'readr' was built under R version 4.2.2
## Warning:  package 'purrr' was built under R version 4.2.3
## Warning:  package 'dplyr' was built under R version 4.2.3
## Warning:  package 'stringr' was built under R version 4.2.3
## Warning:  package 'forcats' was built under R version 4.2.2
## Warning:  package 'lubridate' was built under R version 4.2.2
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0
--
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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 errors
```

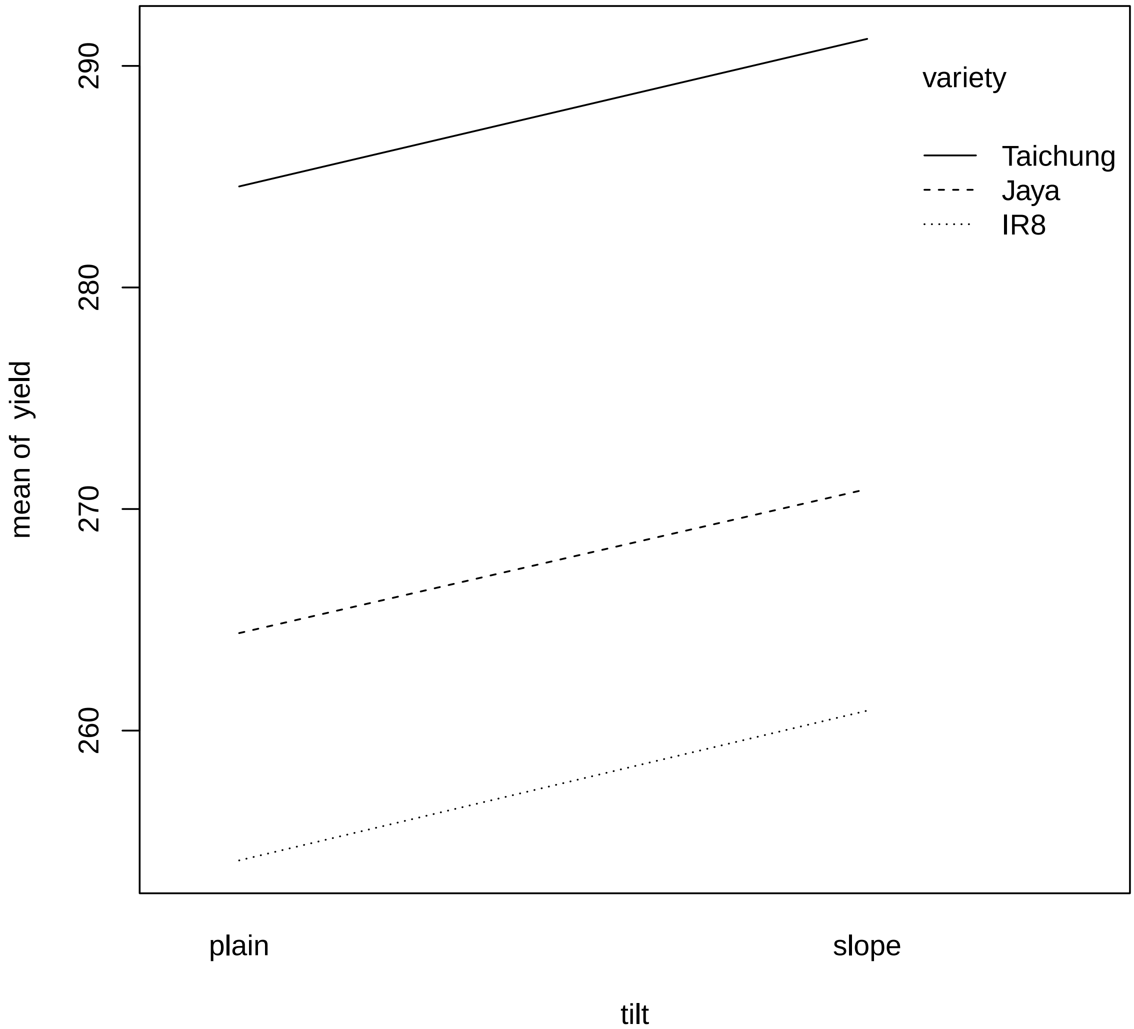
```r
agri_dat %>%
  ggplot(aes(x = tilt, y = yield)) +
  stat_boxplot(geom = "errorbar", linewidth = 1) +
  geom_boxplot(fill = "#1BF248") +
  labs(x = "Tilt", y = "Yield", title = "Boxplot of Yield in Different Tilts")
```



```
agri_dat %>%
  ggplot(aes(x = variety, y = yield)) +
  stat_boxplot(geom = "errorbar", linewidth = 1) +
  geom_boxplot(fill = "#56DFEA") +
  labs(x = "Variety", y = "Yield", title = "Boxplot of Yield of Different Varieties")
```

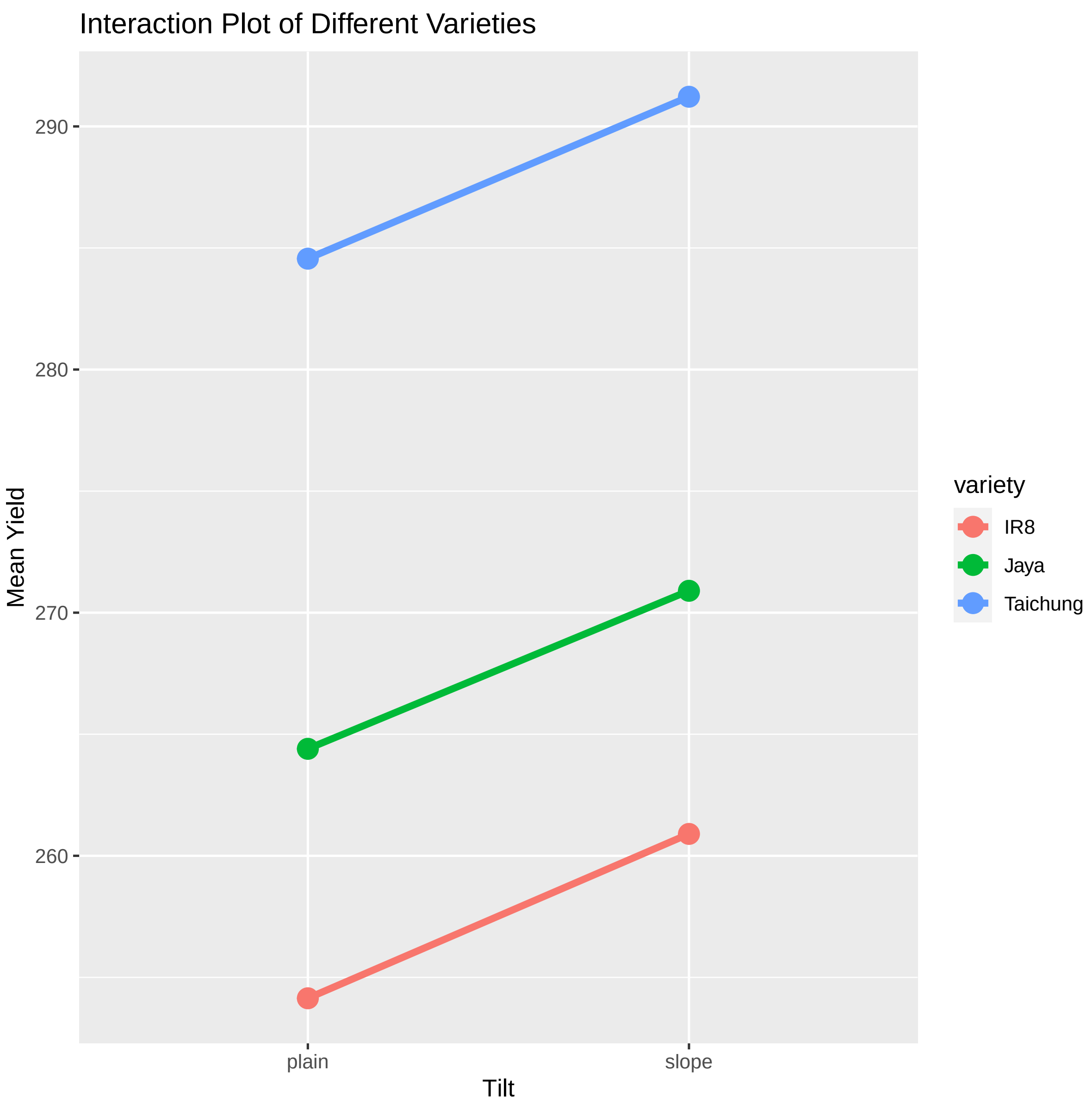


```r
with(agri_dat, interaction.plot(tilt, variety, yield))
```



For interaction plot, the first argument is the variable that I want in x-axis, the second argument is the variable that I want as profile and the third argument is the variable that I want in y-axis.

```r
df1 <- agri_dat %>%
  group_by(variety, tilt) %>%
  summarise(mean_yield = mean(yield))

## `summarise()` has grouped output by 'variety'.  You can override using the
## `.groups` argument.

df1 %>%
  ggplot(aes(x = tilt, y = mean_yield)) +
  geom_line(aes(group = variety, color = variety), linewidth = 1.5) +
  geom_point(aes(color = variety), size = 4) +
  labs(x = "Tilt", y = "Mean Yield", title = "Interaction Plot of Different Varieties")
```



Such an interaction plot translates to an additive model.
The different varieties IR8, Jaya, Taichung are often referred as *profiles*.

The boxplots verify that the homoscedasticity assumption is true.

```r
fit1 = lm(yield ~ tilt + variety, data = agri_dat)
```

```r
fit1
```

```
##
## Call:
## lm(formula = yield ~ tilt + variety, data = agri_dat)
##
## Coefficients:
##     (Intercept)        tiltslope      varietyJaya  varietyTaichung
##          254.20             6.64            10.13            30.37
```

Here the linear model is :

$$y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$$

where μ is the benchmark yield.

The estimates of *tiltplane* and *varietyIR8* have been forced to 0; *i.e.* $\alpha_1 = 0$ and $\beta_1 = 0$.

```r
summary(fit1)
```

```
##
## Call:
## lm(formula = yield ~ tilt + variety, data = agri_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4400 -0.1600 -0.0050  0.1925  0.4300
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     254.20000    0.09109 2790.64   <2e-16 ***
## tiltslope         6.64000    0.09109   72.89   <2e-16 ***
## varietyJaya      10.13000    0.11156   90.80   <2e-16 ***
## varietyTaichung  30.37000    0.11156  272.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2495 on 26 degrees of freedom
## Multiple R-squared:  0.9997,Adjusted R-squared:  0.9996
## F-statistic: 2.739e+04 on 3 and 26 DF,  p-value: < 2.2e-16
```

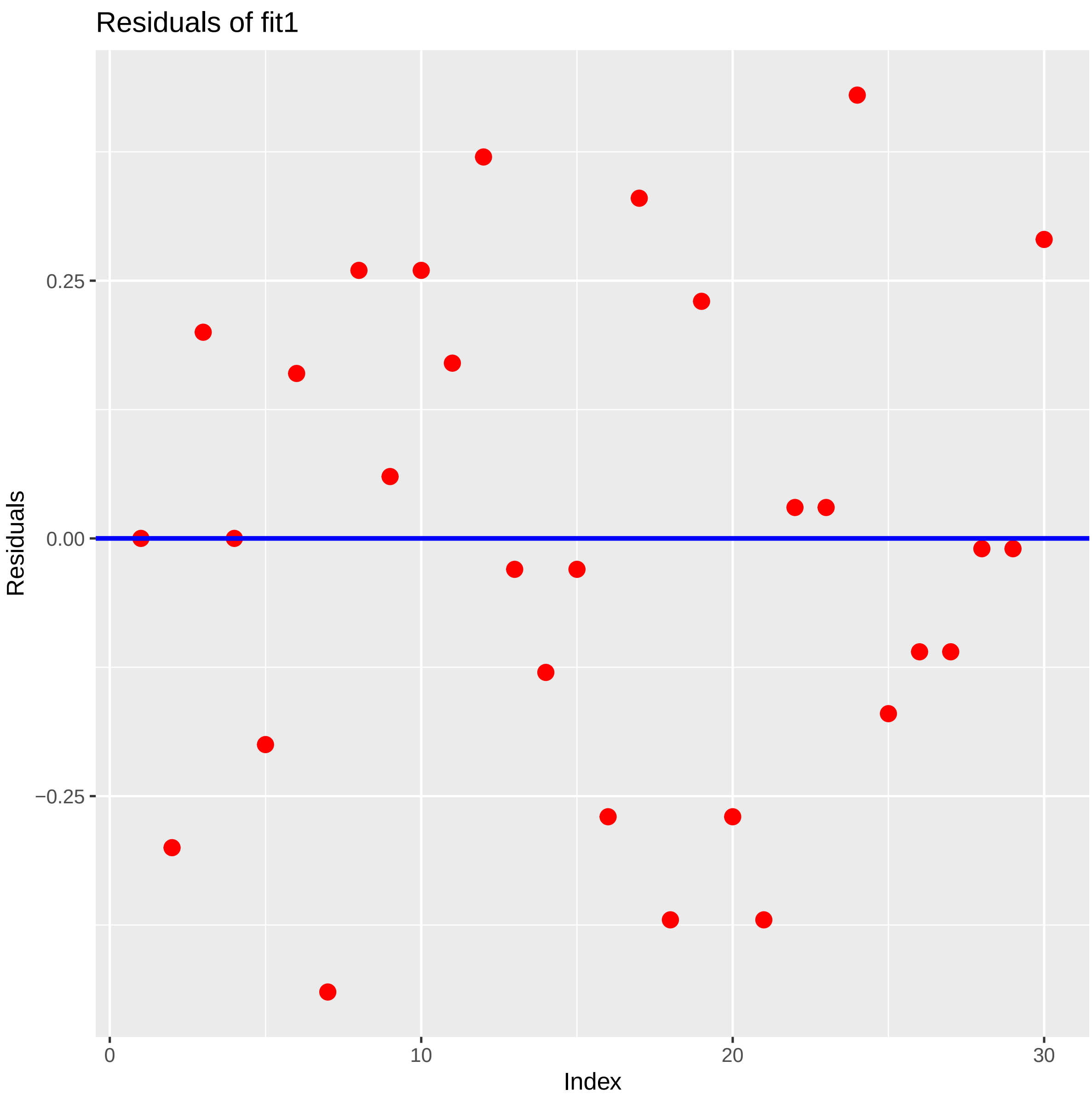
```r
model.matrix(fit1)
```

```
##    (Intercept) tiltslope varietyJaya varietyTaichung
## 1            1         0           0               0
## 2            1         0           0               0
## 3            1         0           0               0
## 4            1         0           0               0
```

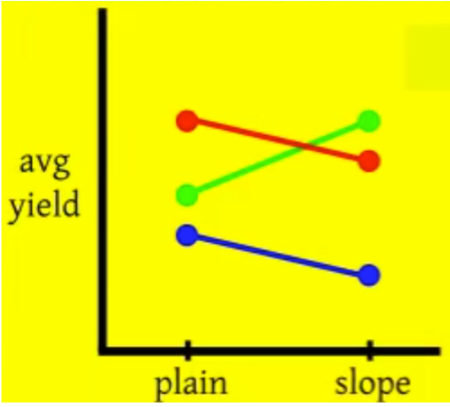
```
## 5            1         0           0               0
## 6            1         1           0               0
## 7            1         1           0               0
## 8            1         1           0               0
## 9            1         1           0               0
## 10           1         1           0               0
## 11           1         0           1               0
## 12           1         0           1               0
## 13           1         0           1               0
## 14           1         0           1               0
## 15           1         0           1               0
## 16           1         1           1               0
## 17           1         1           1               0
## 18           1         1           1               0
## 19           1         1           1               0
## 20           1         1           1               0
## 21           1         0           0               1
## 22           1         0           0               1
## 23           1         0           0               1
## 24           1         0           0               1
## 25           1         0           0               1
## 26           1         1           0               1
## 27           1         1           0               1
## 28           1         1           0               1
## 29           1         1           0               1
## 30           1         1           0               1
## attr(,"assign")
## [1] 0 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$tilt
## [1] "contr.treatment"
##
## attr(,"contrasts")$variety
## [1] "contr.treatment"
```

```
fit1$rank

## [1] 4
```

```r
temp_df <- data.frame(fit1$residuals)

temp_df %>%
  ggplot(aes(y = fit1.residuals, x = 1:length(fit1.residuals))) +
  geom_point(color = "red", size = 3) +
  geom_hline(yintercept = 0, color = "blue", linewidth = 1) +
  labs(x = "Index", y = "Residuals", title = "Residuals of fit1")
```



Two Factor Model with Interaction



If we have an interaction plot like the above *i.e.* at least two of the profiles are intersecting or not so parallel, we shall introduce a new linear model where we shall take count of the interaction of the two factors.

The linear model is :

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon_{ijk}$$

where γ_{ij}s take count of the interaction.

In practice, we shall first consider this model. We shall test whether γ_{ij}s are 0 or not. If all the γ_{ij}s are 0, then we shall resort to the **additive model**. If any one of the γ_{ij}s is non-zero, then we shall report that. But we shall never estimate γ_{ij}s.

For a statistician, interaction is bad news. Because, when there is no interaction, we can talk about the inputs separately. But interaction spoils the fun by saying, you cannot really say how the inputs are connected to the output, they are inextricable; and it is their combined influence which is effecting the output. So all that a statistician can say is *things are twisted* and nothing more.

- **Cell Means Model**

When we find any one of the γ_{ij}s is non-zero (*i.e.* there is some interaction), then we shall resort to this model :

$$y_{ijk} = \mu_{ij} + \epsilon_{ijk}$$

where μ_{ij} is the expected yield of variety i in tilt j and the fluctuations in y_{ijk} are due to the random error ϵ_{ijk}.

```
getwd()

## [1] "D:/Programming Languages/R/Linear Statistical Models - Arnab Chakraborty/005"
```

```
paddy_data = read.csv("agriculture_dataset_2.csv")

paddy_data
```

```
##     variety  tilt  yield
## 1        IR8 plain 250.20
## 2        IR8 plain 249.97
## 3        IR8 plain 250.08
## 4        IR8 plain 250.29
## 5        IR8 plain 250.27
## 6        IR8 slope 240.50
## 7        IR8 slope 240.82
## 8        IR8 slope 240.61
## 9        IR8 slope 240.30
## 10       IR8 slope 240.65
## 11      Jaya plain 260.75
## 12      Jaya plain 260.64
## 13      Jaya plain 260.57
## 14      Jaya plain 260.17
## 15      Jaya plain 260.52
## 16      Jaya slope 275.32
## 17      Jaya slope 275.56
## 18      Jaya slope 275.59
## 19      Jaya slope 275.24
## 20      Jaya slope 275.82
## 21 Taichung plain 280.76
## 22 Taichung plain 280.83
## 23 Taichung plain 281.29
## 24 Taichung plain 280.66
## 25 Taichung plain 280.62
## 26 Taichung slope 252.79
## 27 Taichung slope 252.92
## 28 Taichung slope 253.29
## 29 Taichung slope 253.25
## 30 Taichung slope 253.14
```

```
dim(paddy_data)
```

```
## [1] 30  3
```

```
names(paddy_data)
```

```
## [1] "variety" "tilt"    "yield"
```

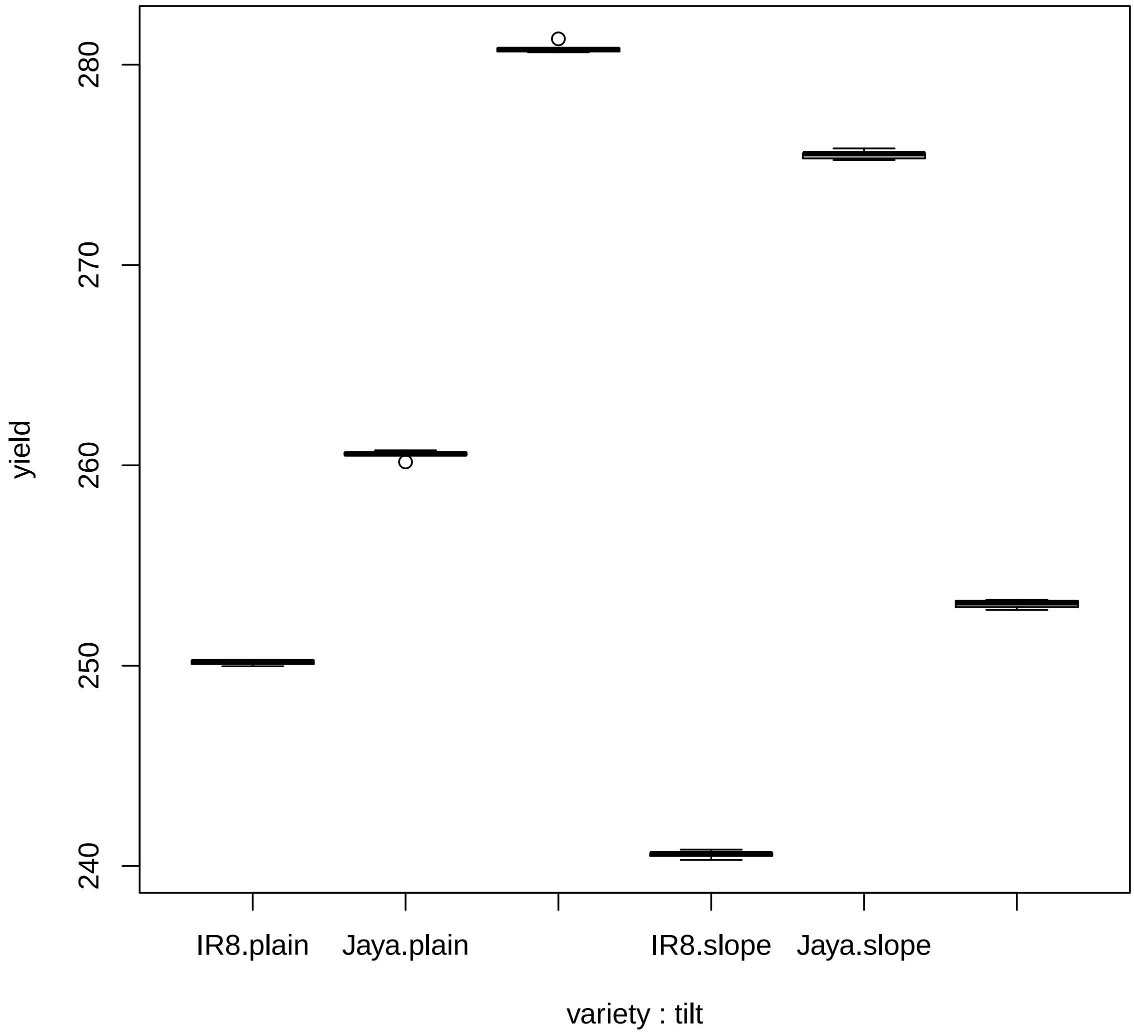
```
head(paddy_data)
```

```
##   variety  tilt  yield
## 1     IR8 plain 250.20
## 2     IR8 plain 249.97
## 3     IR8 plain 250.08
## 4     IR8 plain 250.29
## 5     IR8 plain 250.27
## 6     IR8 slope 240.50
```

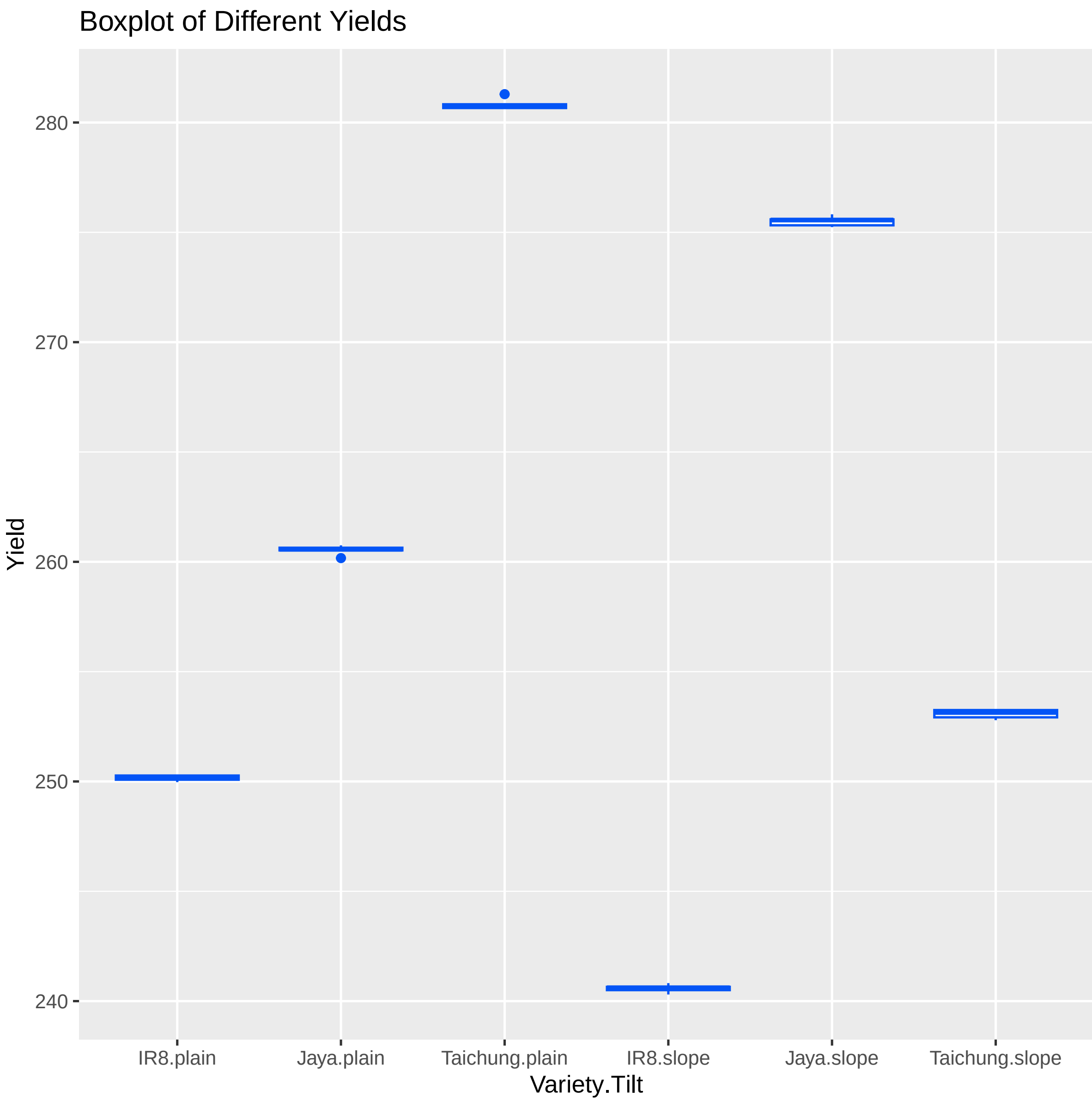
```
tail(paddy_data)
```

```
##     variety  tilt  yield
## 25 Taichung plain 280.62
## 26 Taichung slope 252.79
## 27 Taichung slope 252.92
## 28 Taichung slope 253.29
## 29 Taichung slope 253.25
## 30 Taichung slope 253.14
```

```r
with(data = paddy_data, boxplot(yield ~ variety:tilt))
```

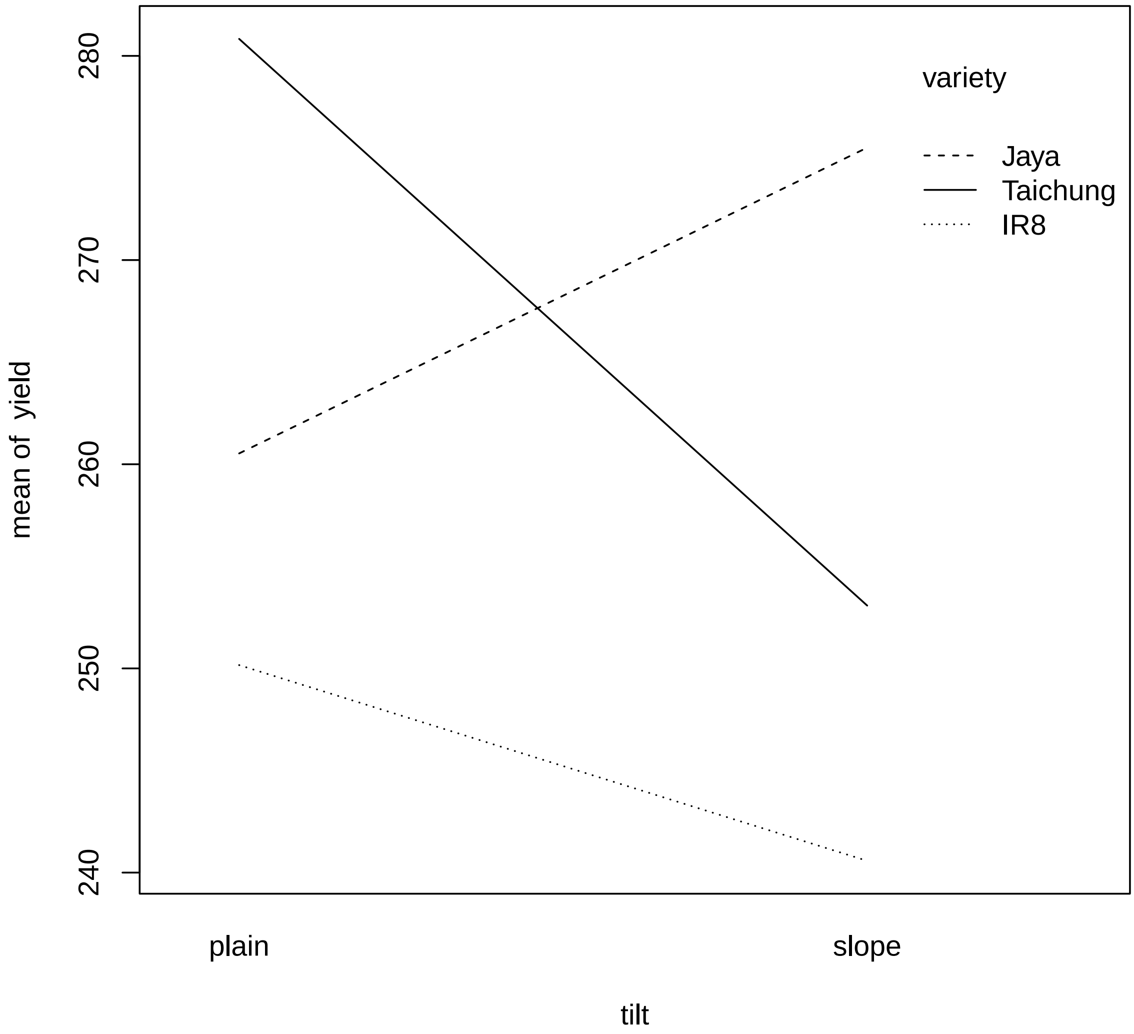


```r
paddy_data %>%
  ggplot(aes(x = interaction(variety, tilt), y = yield)) +
  geom_boxplot(col = "#0354F6") +
  labs(x = "Variety.Tilt", y = "Yield", title = "Boxplot of Different Yields")
```



The boxplots verify that the homoscedasticity assumption is true and it also gives an idea about the interaction plot.

```r
with(data = paddy_data, interaction.plot(tilt, variety, yield))
```



```r
df2 <- paddy_data %>%
  group_by(variety, tilt) %>%
  summarise(mean_yield = mean(yield))

## `summarise()` has grouped output by 'variety'.  You can override using the
## `.groups` argument.

df2 %>%
  ggplot(aes(x = tilt, y = mean_yield)) +
  geom_line(aes(group = variety, color = variety), linewidth = 1.5) +
  geom_point(aes(color = variety), size = 4) +
  labs(x = "Tilt", y = "Mean Yield", title = "Interaction Plot of Different Varieties")
```


```r
# fit2 = lm(yield ~ variety + tilt + variety:tilt, data = paddy_data)
# An abbrebriation of the above command is
fit2 = lm(yield ~ variety*tilt, data = paddy_data)
```

```r
fit2
```

```
##
## Call:
## lm(formula = yield ~ variety * tilt, data = paddy_data)
##
## Coefficients:
##               (Intercept)                varietyJaya
##                   250.162                     10.368
##          varietyTaichung                  tiltslope
##                    30.670                     -9.586
##     varietyJaya:tiltslope  varietyTaichung:tiltslope
##                    24.562                    -18.168
```

Here α_1, β_1 have been forced to 0.

Also 2 of the 6 interaction terms γ_{22} and γ_{32} have been reported and others have been dropped.

```r
model.matrix(fit2)
```

```
##    (Intercept) varietyJaya varietyTaichung tiltslope varietyJaya:tiltslope
## 1            1           0               0         0                     0
## 2            1           0               0         0                     0
## 3            1           0               0         0                     0
## 4            1           0               0         0                     0
## 5            1           0               0         0                     0
## 6            1           0               0         1                     0
## 7            1           0               0         1                     0
## 8            1           0               0         1                     0
## 9            1           0               0         1                     0
## 10           1           0               0         1                     0
## 11           1           1               0         0                     0
## 12           1           1               0         0                     0
## 13           1           1               0         0                     0
## 14           1           1               0         0                     0
## 15           1           1               0         0                     0
## 16           1           1               0         1                     1
## 17           1           1               0         1                     1
## 18           1           1               0         1                     1
## 19           1           1               0         1                     1
## 20           1           1               0         1                     1
## 21           1           0               1         0                     0
## 22           1           0               1         0                     0
## 23           1           0               1         0                     0
## 24           1           0               1         0                     0
## 25           1           0               1         0                     0
## 26           1           0               1         1                     0
## 27           1           0               1         1                     0
## 28           1           0               1         1                     0
## 29           1           0               1         1                     0
## 30           1           0               1         1                     0
```

```
##    varietyTaichung:tiltslope
## 1                           0
## 2                           0
## 3                           0
## 4                           0
## 5                           0
## 6                           0
## 7                           0
## 8                           0
## 9                           0
## 10                          0
## 11                          0
## 12                          0
## 13                          0
## 14                          0
## 15                          0
## 16                          0
## 17                          0
## 18                          0
## 19                          0
## 20                          0
## 21                          0
## 22                          0
## 23                          0
## 24                          0
## 25                          0
## 26                          1
## 27                          1
## 28                          1
## 29                          1
## 30                          1
## attr(,"assign")
## [1] 0 1 1 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$variety
## [1] "contr.treatment"
##
## attr(,"contrasts")$tilt
## [1] "contr.treatment"
```

```
fit2$rank

## [1] 6
```

Observe that, when the rank of the model matrix is 6, R will report only 6 values in fit2.

```
summary(fit2)

##
## Call:
## lm(formula = yield ~ variety * tilt, data = paddy_data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3600 -0.1685  0.0360  0.1095  0.4580
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               250.16200    0.09586 2609.68   <2e-16 ***
## varietyJaya                10.36800    0.13557   76.48   <2e-16 ***
## varietyTaichung            30.67000    0.13557  226.24   <2e-16 ***
## tiltslope                  -9.58600    0.13557  -70.71   <2e-16 ***
## varietyJaya:tiltslope      24.56200    0.19172  128.12   <2e-16 ***
## varietyTaichung:tiltslope -18.16800    0.19172  -94.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2143 on 24 degrees of freedom
## Multiple R-squared:  0.9998,Adjusted R-squared:  0.9998
## F-statistic: 2.604e+04 on 5 and 24 DF,  p-value: < 2.2e-16
```

```r
df3 <- data.frame(fit2$residuals)

df3 %>%
  ggplot(aes(x = 1:length(fit2.residuals), y = fit2.residuals)) +
  geom_point(color = "red", size = 3) +
  geom_hline(yintercept = 0, color = "blue", linewidth = 1.5) +
  labs(x = "Index", y = "Residuals", title = "Residuals of fit2")
```