Alternative management

Marlese

2/9/2021

# librarires

library(openxlsx)  
library(readr)  
  
library(tidytext)  
library(tidyr)  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3 v dplyr 1.0.5  
## v tibble 3.1.1 v stringr 1.4.0  
## v purrr 0.3.4 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggthemes)  
library(ggplot2)  
library(rvest)

##   
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':  
##   
## guess\_encoding

library(xml2)  
library(dplyr)  
library(plyr)

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

library(agricolae)  
  
library(MASS) #ordered logistic regression

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

# logistic  
require(foreign)

## Loading required package: foreign

require(ggplot2)  
require(MASS)  
require(Hmisc)

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:plyr':  
##   
## is.discrete, summarize

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

require(reshape2)

## Loading required package: reshape2

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

library(nnet)  
  
library(wesanderson)

#import

manage\_raw = read.xlsx("C:\\Users\\2010088819\\Documents\\R Projects\\Scle\_management\\manage.xlsx", "temp")

# temperature

## modify

manage\_raw$size\_mm <- as.factor(manage\_raw$size\_mm)  
  
manage\_raw$weight\_g <- as.factor(manage\_raw$weight\_g)  
  
manage\_raw$rep <- as.factor(manage\_raw$rep)  
  
manage\_raw$depth\_cm <- as.factor(manage\_raw$depth\_cm)  
  
manage\_raw$time\_min <- as.factor(manage\_raw$time\_min)  
  
manage\_raw$temp <- as.factor(manage\_raw$temp)  
  
manage\_raw$myc <- as.character(manage\_raw$myc)  
  
manage\_raw$scle <- as.character(manage\_raw$scle)  
  
  
  
  
manage\_raw1 <- manage\_raw %>%  
 filter(temp %in% c("125", "155", "185", "200")) %>%   
 mutate(myc1 = case\_when(myc == "0" ~ "absent",  
 myc == "1" ~ "contamination",  
 myc == "2" ~ "present\_partial",  
 myc == "3" ~ "present\_full",  
 myc != "\*" ~ myc),  
 scle1 = case\_when(scle == "0" ~ "absent",  
 scle == "1" ~ "contamination",  
 scle == "2" ~ "present\_partial",  
 scle == "3" ~ "present\_full",  
 scle != "\*" ~ scle))

## histogram

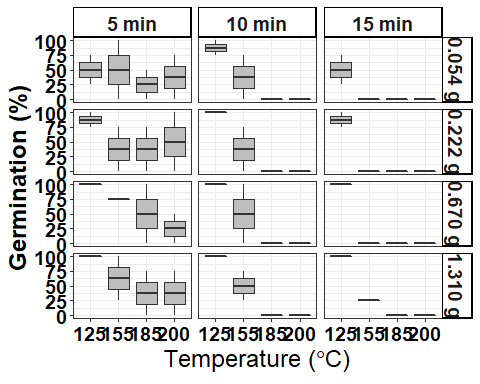
freq\_df <- manage\_raw %>%  
 filter(temp %in% c("125", "155", "185", "200")) %>%   
 mutate(binary = case\_when(myc == 0 ~ 0,  
 myc == 1 ~ 0,  
 myc == 2 ~ 1,  
 myc == 3 ~ 1)) %>%   
 group\_by(size\_mm, weight\_g, depth\_cm, time\_min, temp) %>%   
 dplyr::summarise(freq = sum(binary)/4\*100)

## `summarise()` has grouped output by 'size\_mm', 'weight\_g', 'depth\_cm', 'time\_min'. You can override using the `.groups` argument.

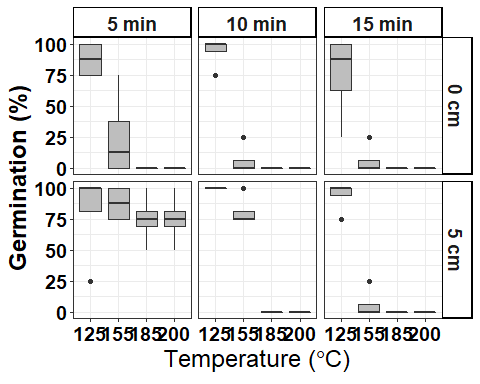
freq\_df <- manage\_raw %>%  
 filter(temp %in% c("125", "155", "185", "200")) %>%   
 mutate(binary = case\_when(myc == 0 ~ 0,  
 myc == 1 ~ 0,  
 myc == 2 ~ 1,  
 myc == 3 ~ 1)) %>%   
 group\_by(size\_mm, weight\_g, depth\_cm, time\_min, temp) %>%   
 dplyr::summarise(freq = sum(binary)/4\*100)

## `summarise()` has grouped output by 'size\_mm', 'weight\_g', 'depth\_cm', 'time\_min'. You can override using the `.groups` argument.

new\_label\_time <- c("5" = "5 min", "10" = "10 min", "15" = "15 min")  
new\_label\_depth <- c("0" = "0 cm", "5" = "5 cm")  
new\_label\_weight <- c("0.054" = "0.054 g", "0.222" = "0.222 g", "0.67" = "0.670 g", "1.31" = "1.310 g")  
  
  
freq\_df %>%   
 ggplot()+  
 geom\_boxplot(aes(temp, freq), fill = "grey")+  
 facet\_grid(~weight\_g~time\_min, labeller = labeller(weight\_g = new\_label\_weight, time\_min = new\_label\_time))+  
 theme\_bw()+  
 labs(x = expression(paste("Temperature (",degree,"C)")), y = "Germination (%)")+  
 theme(axis.title.x=element\_text(size=18,face="bold"),  
 axis.title.y=element\_text(size=18,face="bold"),  
 axis.text.y = element\_text(size = 14, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 14, color = "black", face = "bold"),  
 legend.text = element\_text(size = 16),  
 legend.title = element\_text(size = 18),  
 strip.background = element\_rect(fill="white", size=1, color="black"),  
 strip.text.x = element\_text(size=14, face="bold"),  
 strip.text.y = element\_text(size=14, face="bold"))+  
 ggsave("plots/variables\_weight.png", units = "cm", width = 60, height = 30, dpi = 300)



freq\_df %>%   
 ggplot()+  
 geom\_boxplot(aes(temp, freq), fill = "grey")+  
 facet\_grid(~depth\_cm~time\_min, labeller = labeller(depth\_cm = new\_label\_depth, time\_min = new\_label\_time))+  
 theme\_bw()+  
 labs(x = expression(paste("Temperature (",degree,"C)")), y = "Germination (%)")+  
 theme(axis.title.x=element\_text(size=18,face="bold"),  
 axis.title.y=element\_text(size=18,face="bold"),  
 axis.text.y = element\_text(size = 14, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 14, color = "black", face = "bold"),  
 legend.text = element\_text(size = 16),  
 legend.title = element\_text(size = 18),  
 strip.background = element\_rect(fill="white", size=1, color="black"),  
 strip.text.x = element\_text(size=14, face="bold"),  
 strip.text.y = element\_text(size=14, face="bold"))+  
 ggsave("plots/variables\_depth.png", units = "cm", width = 60, height = 30, dpi = 300)



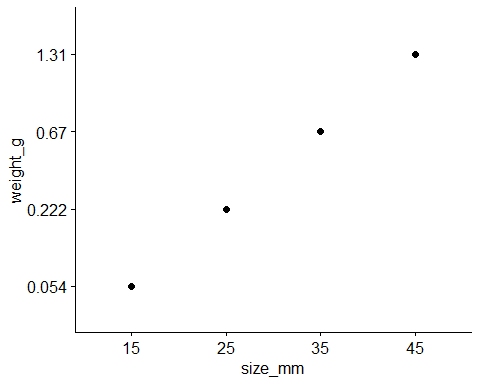
library(ggpubr)

##   
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':  
##   
## mutate

ggscatter(manage\_raw, x = "size\_mm", y = "weight\_g",  
 add = "reg.line",  
 conf.int = TRUE,  
 add.oarams= list(color = "blue",  
 fill = "gray"))+  
 stat\_cor(method = "pearson", label.x = 15, label.y = 1)

## `geom\_smooth()` using formula 'y ~ x'



## anova

aov.freq <- aov(freq ~ weight\_g\*temp + weight\_g\*time\_min + weight\_g\*depth\_cm + temp\*time\_min + temp\*depth\_cm + time\_min\*depth\_cm + temp\*time\_min\*depth\_cm + temp\*depth\_cm\*weight\_g + time\_min\*depth\_cm\*weight\_g, freq\_df)  
  
summary(aov.freq)

## Df Sum Sq Mean Sq F value Pr(>F)   
## weight\_g 3 3197 1066 5.299 0.003954 \*\*   
## temp 3 92624 30875 153.529 < 2e-16 \*\*\*  
## time\_min 2 15833 7917 39.367 8.69e-10 \*\*\*  
## depth\_cm 1 15632 15632 77.730 1.61e-10 \*\*\*  
## weight\_g:temp 9 4694 522 2.594 0.020348 \*   
## weight\_g:time\_min 6 729 122 0.604 0.725028   
## weight\_g:depth\_cm 3 384 128 0.637 0.596281   
## temp:time\_min 6 10833 1806 8.978 5.16e-06 \*\*\*  
## temp:depth\_cm 3 4707 1569 7.802 0.000387 \*\*\*  
## time\_min:depth\_cm 2 9115 4557 22.662 4.26e-07 \*\*\*  
## temp:time\_min:depth\_cm 6 12969 2161 10.748 7.84e-07 \*\*\*  
## weight\_g:temp:depth\_cm 9 840 93 0.464 0.888900   
## weight\_g:time\_min:depth\_cm 6 1198 200 0.993 0.444880   
## Residuals 36 7240 201   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(aov.freq)

## Analysis of Variance Table  
##   
## Response: freq  
## Df Sum Sq Mean Sq F value Pr(>F)   
## weight\_g 3 3197 1065.5 5.2986 0.0039543 \*\*   
## temp 3 92624 30874.6 153.5288 < 2.2e-16 \*\*\*  
## time\_min 2 15833 7916.7 39.3669 8.690e-10 \*\*\*  
## depth\_cm 1 15632 15631.5 77.7302 1.607e-10 \*\*\*  
## weight\_g:temp 9 4694 521.6 2.5935 0.0203484 \*   
## weight\_g:time\_min 6 729 121.5 0.6043 0.7250279   
## weight\_g:depth\_cm 3 384 128.0 0.6367 0.5962806   
## temp:time\_min 6 10833 1805.6 8.9784 5.165e-06 \*\*\*  
## temp:depth\_cm 3 4707 1569.0 7.8022 0.0003871 \*\*\*  
## time\_min:depth\_cm 2 9115 4557.3 22.6619 4.261e-07 \*\*\*  
## temp:time\_min:depth\_cm 6 12969 2161.5 10.7482 7.839e-07 \*\*\*  
## weight\_g:temp:depth\_cm 9 840 93.3 0.4640 0.8888999   
## weight\_g:time\_min:depth\_cm 6 1198 199.7 0.9928 0.4448802   
## Residuals 36 7240 201.1   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

write.xlsx(aov.freq, "anova\_manage.xlsx")  
  
#residual standard error  
  
sqrt(deviance(aov.freq)/df.residual(aov.freq))

## [1] 14.18096

## LSD

a <- LSD.test(aov.freq, "temp", DFerror = aov.freq$df.residual, MSerror = deviance(aov.freq)/aov.freq$df.residual, alpha = 0.05, group = TRUE)  
b <- LSD.test(aov.freq, "time\_min", DFerror = aov.freq$df.residual, MSerror = deviance(aov.freq)/aov.freq$df.residual, alpha = 0.05, group = TRUE)  
c <- LSD.test(aov.freq, "depth\_cm", DFerror = aov.freq$df.residual, MSerror = deviance(aov.freq)/aov.freq$df.residual, alpha = 0.05, group = TRUE)  
d <- LSD.test(aov.freq, "weight\_g", DFerror = aov.freq$df.residual, MSerror = deviance(aov.freq)/aov.freq$df.residual, alpha = 0.05, group = TRUE)  
  
ab = LSD.test(aov.freq,  
 c("temp", "time\_min"),  
 DFerror = aov.freq$df.residual,  
 MSerror = deviance(aov.freq)/aov.freq$df.residual,  
 alpha = 0.05,  
 group = TRUE,  
 console = TRUE)

##   
## Study: aov.freq ~ c("temp", "time\_min")  
##   
## LSD t Test for freq   
##   
## Mean Square Error: 201.0995   
##   
## temp:time\_min, means and individual ( 95 %) CI  
##   
## freq std r LCL UCL Min Max  
## 125:10 96.875 8.838835 8 86.706694 107.04331 75 100  
## 125:15 84.375 26.516504 8 74.206694 94.54331 25 100  
## 125:5 84.375 26.516504 8 74.206694 94.54331 25 100  
## 155:10 43.750 41.726148 8 33.581694 53.91831 0 100  
## 155:15 6.250 11.572751 8 -3.918306 16.41831 0 25  
## 155:5 56.250 41.726148 8 46.081694 66.41831 0 100  
## 185:10 0.000 0.000000 8 -10.168306 10.16831 0 0  
## 185:15 0.000 0.000000 8 -10.168306 10.16831 0 0  
## 185:5 37.500 42.257713 8 27.331694 47.66831 0 100  
## 200:10 0.000 0.000000 8 -10.168306 10.16831 0 0  
## 200:15 0.000 0.000000 8 -10.168306 10.16831 0 0  
## 200:5 37.500 42.257713 8 27.331694 47.66831 0 100  
##   
## Alpha: 0.05 ; DF Error: 36  
## Critical Value of t: 2.028094   
##   
## least Significant Difference: 14.38016   
##   
## Treatments with the same letter are not significantly different.  
##   
## freq groups  
## 125:10 96.875 a  
## 125:15 84.375 a  
## 125:5 84.375 a  
## 155:5 56.250 b  
## 155:10 43.750 bc  
## 185:5 37.500 c  
## 200:5 37.500 c  
## 155:15 6.250 d  
## 185:10 0.000 d  
## 185:15 0.000 d  
## 200:10 0.000 d  
## 200:15 0.000 d

ac = LSD.test(aov.freq,  
 c("temp", "depth\_cm"),  
 DFerror = aov.freq$df.residual,  
 MSerror = deviance(aov.freq)/aov.freq$df.residual,  
 alpha = 0.05,  
 group = TRUE,  
 console = TRUE)

##   
## Study: aov.freq ~ c("temp", "depth\_cm")  
##   
## LSD t Test for freq   
##   
## Mean Square Error: 201.0995   
##   
## temp:depth\_cm, means and individual ( 95 %) CI  
##   
## freq std r LCL UCL Min Max  
## 125:0 85.41667 22.50842 12 77.114279 93.719054 25 100  
## 125:5 91.66667 22.19063 12 83.364279 99.969054 25 100  
## 155:0 12.50000 22.61335 12 4.197613 20.802387 0 75  
## 155:5 58.33333 40.35824 12 50.030946 66.635721 0 100  
## 185:0 0.00000 0.00000 12 -8.302387 8.302387 0 0  
## 185:5 25.00000 38.43531 12 16.697613 33.302387 0 100  
## 200:0 0.00000 0.00000 12 -8.302387 8.302387 0 0  
## 200:5 25.00000 38.43531 12 16.697613 33.302387 0 100  
##   
## Alpha: 0.05 ; DF Error: 36  
## Critical Value of t: 2.028094   
##   
## least Significant Difference: 11.74135   
##   
## Treatments with the same letter are not significantly different.  
##   
## freq groups  
## 125:5 91.66667 a  
## 125:0 85.41667 a  
## 155:5 58.33333 b  
## 185:5 25.00000 c  
## 200:5 25.00000 c  
## 155:0 12.50000 d  
## 185:0 0.00000 e  
## 200:0 0.00000 e

bc = LSD.test(aov.freq,  
 c("time\_min", "depth\_cm"),  
 DFerror = aov.freq$df.residual,  
 MSerror = deviance(aov.freq)/aov.freq$df.residual,  
 alpha = 0.05,  
 group = TRUE,  
 console = TRUE)

##   
## Study: aov.freq ~ c("time\_min", "depth\_cm")  
##   
## LSD t Test for freq   
##   
## Mean Square Error: 201.0995   
##   
## time\_min:depth\_cm, means and individual ( 95 %) CI  
##   
## freq std r LCL UCL Min Max  
## 10:0 25.0000 41.83300 16 17.80992 32.19008 0 100  
## 10:5 45.3125 47.62593 16 38.12242 52.50258 0 100  
## 15:0 20.3125 36.76360 16 13.12242 27.50258 0 100  
## 15:5 25.0000 41.83300 16 17.80992 32.19008 0 100  
## 5:0 28.1250 40.69705 16 20.93492 35.31508 0 100  
## 5:5 79.6875 22.76465 16 72.49742 86.87758 25 100  
##   
## Alpha: 0.05 ; DF Error: 36  
## Critical Value of t: 2.028094   
##   
## least Significant Difference: 10.16831   
##   
## Treatments with the same letter are not significantly different.  
##   
## freq groups  
## 5:5 79.6875 a  
## 10:5 45.3125 b  
## 5:0 28.1250 c  
## 10:0 25.0000 c  
## 15:5 25.0000 c  
## 15:0 20.3125 c

ad = LSD.test(aov.freq,  
 c("temp", "weight\_g"),  
 DFerror = aov.freq$df.residual,  
 MSerror = deviance(aov.freq)/aov.freq$df.residual,  
 alpha = 0.05,  
 group = TRUE,  
 console = TRUE)

##   
## Study: aov.freq ~ c("temp", "weight\_g")  
##   
## LSD t Test for freq   
##   
## Mean Square Error: 201.0995   
##   
## temp:weight\_g, means and individual ( 95 %) CI  
##   
## freq std r LCL UCL Min Max  
## 125:0.054 62.500000 30.61862 6 50.7586512 74.24135 25 100  
## 125:0.222 91.666667 12.90994 6 79.9253178 103.40802 75 100  
## 125:0.67 100.000000 0.00000 6 88.2586512 111.74135 100 100  
## 125:1.31 100.000000 0.00000 6 88.2586512 111.74135 100 100  
## 155:0.054 29.166667 45.87120 6 17.4253178 40.90802 0 100  
## 155:0.222 25.000000 38.72983 6 13.2586512 36.74135 0 75  
## 155:0.67 41.666667 46.54747 6 29.9253178 53.40802 0 100  
## 155:1.31 45.833333 33.22900 6 34.0919845 57.57468 25 100  
## 185:0.054 8.333333 20.41241 6 -3.4080155 20.07468 0 50  
## 185:0.222 12.500000 30.61862 6 0.7586512 24.24135 0 75  
## 185:0.67 16.666667 40.82483 6 4.9253178 28.40802 0 100  
## 185:1.31 12.500000 30.61862 6 0.7586512 24.24135 0 75  
## 200:0.054 12.500000 30.61862 6 0.7586512 24.24135 0 75  
## 200:0.222 16.666667 40.82483 6 4.9253178 28.40802 0 100  
## 200:0.67 8.333333 20.41241 6 -3.4080155 20.07468 0 50  
## 200:1.31 12.500000 30.61862 6 0.7586512 24.24135 0 75  
##   
## Alpha: 0.05 ; DF Error: 36  
## Critical Value of t: 2.028094   
##   
## least Significant Difference: 16.60477   
##   
## Treatments with the same letter are not significantly different.  
##   
## freq groups  
## 125:0.67 100.000000 a  
## 125:1.31 100.000000 a  
## 125:0.222 91.666667 a  
## 125:0.054 62.500000 b  
## 155:1.31 45.833333 c  
## 155:0.67 41.666667 cd  
## 155:0.054 29.166667 de  
## 155:0.222 25.000000 ef  
## 185:0.67 16.666667 efg  
## 200:0.222 16.666667 efg  
## 185:0.222 12.500000 fg  
## 185:1.31 12.500000 fg  
## 200:0.054 12.500000 fg  
## 200:1.31 12.500000 fg  
## 185:0.054 8.333333 g  
## 200:0.67 8.333333 g

abc = LSD.test(aov.freq,  
 c("temp", "time\_min", "depth\_cm"),  
 DFerror = aov.freq$df.residual,  
 MSerror = deviance(aov.freq)/aov.freq$df.residual,  
 alpha = 0.05,  
 group = TRUE,  
 console = TRUE)

##   
## Study: aov.freq ~ c("temp", "time\_min", "depth\_cm")  
##   
## LSD t Test for freq   
##   
## Mean Square Error: 201.0995   
##   
## temp:time\_min:depth\_cm, means and individual ( 95 %) CI  
##   
## freq std r LCL UCL Min Max  
## 125:10:0 93.75 12.50000 4 79.369843 108.13016 75 100  
## 125:10:5 100.00 0.00000 4 85.619843 114.38016 100 100  
## 125:15:0 75.00 35.35534 4 60.619843 89.38016 25 100  
## 125:15:5 93.75 12.50000 4 79.369843 108.13016 75 100  
## 125:5:0 87.50 14.43376 4 73.119843 101.88016 75 100  
## 125:5:5 81.25 37.50000 4 66.869843 95.63016 25 100  
## 155:10:0 6.25 12.50000 4 -8.130157 20.63016 0 25  
## 155:10:5 81.25 12.50000 4 66.869843 95.63016 75 100  
## 155:15:0 6.25 12.50000 4 -8.130157 20.63016 0 25  
## 155:15:5 6.25 12.50000 4 -8.130157 20.63016 0 25  
## 155:5:0 25.00 35.35534 4 10.619843 39.38016 0 75  
## 155:5:5 87.50 14.43376 4 73.119843 101.88016 75 100  
## 185:10:0 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 185:10:5 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 185:15:0 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 185:15:5 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 185:5:0 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 185:5:5 75.00 20.41241 4 60.619843 89.38016 50 100  
## 200:10:0 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 200:10:5 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 200:15:0 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 200:15:5 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 200:5:0 0.00 0.00000 4 -14.380157 14.38016 0 0  
## 200:5:5 75.00 20.41241 4 60.619843 89.38016 50 100  
##   
## Alpha: 0.05 ; DF Error: 36  
## Critical Value of t: 2.028094   
##   
## least Significant Difference: 20.33661   
##   
## Treatments with the same letter are not significantly different.  
##   
## freq groups  
## 125:10:5 100.00 a  
## 125:10:0 93.75 ab  
## 125:15:5 93.75 ab  
## 125:5:0 87.50 ab  
## 155:5:5 87.50 ab  
## 125:5:5 81.25 ab  
## 155:10:5 81.25 ab  
## 125:15:0 75.00 b  
## 185:5:5 75.00 b  
## 200:5:5 75.00 b  
## 155:5:0 25.00 c  
## 155:10:0 6.25 cd  
## 155:15:0 6.25 cd  
## 155:15:5 6.25 cd  
## 185:10:0 0.00 d  
## 185:10:5 0.00 d  
## 185:15:0 0.00 d  
## 185:15:5 0.00 d  
## 185:5:0 0.00 d  
## 200:10:0 0.00 d  
## 200:10:5 0.00 d  
## 200:15:0 0.00 d  
## 200:15:5 0.00 d  
## 200:5:0 0.00 d

##### original grouping

## Original order of LSD$group   
  
ascend\_A = a$groups %>%  
 group\_by(rownames(a$groups)) %>%  
 arrange(rownames(a$groups))  
  
print(ascend\_A)

## # A tibble: 4 x 3  
## # Groups: rownames(a$groups) [4]  
## freq groups `rownames(a$groups)`  
## <dbl> <chr> <chr>   
## 1 88.5 a 125   
## 2 35.4 b 155   
## 3 12.5 c 185   
## 4 12.5 c 200

ascend\_B = b$groups %>%  
 group\_by(rownames(b$groups)) %>%  
 arrange(rownames(b$groups))  
  
print(ascend\_B)

## # A tibble: 3 x 3  
## # Groups: rownames(b$groups) [3]  
## freq groups `rownames(b$groups)`  
## <dbl> <chr> <chr>   
## 1 35.2 b 10   
## 2 22.7 c 15   
## 3 53.9 a 5

ascend\_C = c$groups %>%  
 group\_by(rownames(c$groups)) %>%  
 arrange(rownames(c$groups))  
  
print(ascend\_C)

## # A tibble: 2 x 3  
## # Groups: rownames(c$groups) [2]  
## freq groups `rownames(c$groups)`  
## <dbl> <chr> <chr>   
## 1 24.5 b 0   
## 2 50 a 5

ascend\_D = d$groups %>%  
 group\_by(rownames(d$groups)) %>%  
 arrange(rownames(d$groups))  
  
print(ascend\_D)

## # A tibble: 4 x 3  
## # Groups: rownames(d$groups) [4]  
## freq groups `rownames(d$groups)`  
## <dbl> <chr> <chr>   
## 1 28.1 b 0.054   
## 2 36.5 a 0.222   
## 3 41.7 a 0.67   
## 4 42.7 a 1.31

ascend\_AB = ab$groups %>%  
 group\_by(rownames(ab$groups)) %>%   
 arrange(freq)  
  
print(ascend\_AB)

## # A tibble: 12 x 3  
## # Groups: rownames(ab$groups) [12]  
## freq groups `rownames(ab$groups)`  
## <dbl> <chr> <chr>   
## 1 0 d 185:10   
## 2 0 d 185:15   
## 3 0 d 200:10   
## 4 0 d 200:15   
## 5 6.25 d 155:15   
## 6 37.5 c 185:5   
## 7 37.5 c 200:5   
## 8 43.8 bc 155:10   
## 9 56.2 b 155:5   
## 10 84.4 a 125:15   
## 11 84.4 a 125:5   
## 12 96.9 a 125:10

ascend\_AC = ac$groups %>%  
 group\_by(rownames(ac$groups)) %>%  
 arrange(freq)  
  
print(ascend\_AC)

## # A tibble: 8 x 3  
## # Groups: rownames(ac$groups) [8]  
## freq groups `rownames(ac$groups)`  
## <dbl> <chr> <chr>   
## 1 0 e 185:0   
## 2 0 e 200:0   
## 3 12.5 d 155:0   
## 4 25 c 185:5   
## 5 25 c 200:5   
## 6 58.3 b 155:5   
## 7 85.4 a 125:0   
## 8 91.7 a 125:5

ascend\_BC = bc$groups %>%  
 group\_by(rownames(bc$groups)) %>%  
 arrange(freq)  
  
print(ascend\_BC)

## # A tibble: 6 x 3  
## # Groups: rownames(bc$groups) [6]  
## freq groups `rownames(bc$groups)`  
## <dbl> <chr> <chr>   
## 1 20.3 c 15:0   
## 2 25 c 10:0   
## 3 25 c 15:5   
## 4 28.1 c 5:0   
## 5 45.3 b 10:5   
## 6 79.7 a 5:5

ascend\_AD = ad$groups %>%  
 group\_by(rownames(ad$groups)) %>%  
 arrange(freq)  
  
print(ascend\_AD)

## # A tibble: 16 x 3  
## # Groups: rownames(ad$groups) [16]  
## freq groups `rownames(ad$groups)`  
## <dbl> <chr> <chr>   
## 1 8.33 g 185:0.054   
## 2 8.33 g 200:0.67   
## 3 12.5 fg 185:0.222   
## 4 12.5 fg 185:1.31   
## 5 12.5 fg 200:0.054   
## 6 12.5 fg 200:1.31   
## 7 16.7 efg 185:0.67   
## 8 16.7 efg 200:0.222   
## 9 25 ef 155:0.222   
## 10 29.2 de 155:0.054   
## 11 41.7 cd 155:0.67   
## 12 45.8 c 155:1.31   
## 13 62.5 b 125:0.054   
## 14 91.7 a 125:0.222   
## 15 100 a 125:0.67   
## 16 100 a 125:1.31

ascend\_ABC = abc$groups %>%  
 group\_by(rownames(abc$groups)) %>%  
 arrange(freq)  
  
print(ascend\_ABC)

## # A tibble: 24 x 3  
## # Groups: rownames(abc$groups) [24]  
## freq groups `rownames(abc$groups)`  
## <dbl> <chr> <chr>   
## 1 0 d 185:10:0   
## 2 0 d 185:10:5   
## 3 0 d 185:15:0   
## 4 0 d 185:15:5   
## 5 0 d 185:5:0   
## 6 0 d 200:10:0   
## 7 0 d 200:10:5   
## 8 0 d 200:15:0   
## 9 0 d 200:15:5   
## 10 0 d 200:5:0   
## # ... with 14 more rows

##### Mean error and DF

meanSE\_a = freq\_df %>%  
 dplyr::group\_by(temp) %>%   
 dplyr::summarise(avg\_a = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))  
  
print(meanSE\_a)

## # A tibble: 4 x 3  
## temp avg\_a se  
## <fct> <dbl> <dbl>  
## 1 125 88.5 4.51  
## 2 155 35.4 8.09  
## 3 185 12.5 6.02  
## 4 200 12.5 6.02

meanSE\_b = freq\_df %>%  
 dplyr::group\_by(time\_min) %>%  
 dplyr::summarise(avg\_b = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))  
print(meanSE\_b)

## # A tibble: 3 x 3  
## time\_min avg\_b se  
## <fct> <dbl> <dbl>  
## 1 5 53.9 7.37  
## 2 10 35.2 8.01  
## 3 15 22.7 6.86

meanSE\_c = freq\_df %>%  
 dplyr::group\_by(depth\_cm) %>%  
 dplyr::summarise(avg\_c = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))  
print(meanSE\_c)

## # A tibble: 2 x 3  
## depth\_cm avg\_c se  
## <fct> <dbl> <dbl>  
## 1 0 24.5 5.64  
## 2 5 50 6.40

meanSE\_d = freq\_df %>%  
 dplyr::group\_by(weight\_g) %>%  
 dplyr::summarise(avg\_d = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))  
print(meanSE\_d)

## # A tibble: 4 x 3  
## weight\_g avg\_d se  
## <fct> <dbl> <dbl>  
## 1 0.054 28.1 7.72  
## 2 0.222 36.5 9.15  
## 3 0.67 41.7 9.71  
## 4 1.31 42.7 9.09

MeanSE\_AB = freq\_df %>%  
 dplyr::group\_by(temp, time\_min) %>%  
 dplyr::summarise(avg\_AB = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))

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

MeanSE\_AB = MeanSE\_AB %>%   
 arrange(avg\_AB)  
  
  
print(MeanSE\_AB)

## # A tibble: 12 x 4  
## # Groups: temp [4]  
## temp time\_min avg\_AB se  
## <fct> <fct> <dbl> <dbl>  
## 1 185 10 0 0   
## 2 185 15 0 0   
## 3 200 10 0 0   
## 4 200 15 0 0   
## 5 155 15 6.25 4.09  
## 6 185 5 37.5 14.9   
## 7 200 5 37.5 14.9   
## 8 155 10 43.8 14.8   
## 9 155 5 56.2 14.8   
## 10 125 5 84.4 9.38  
## 11 125 15 84.4 9.38  
## 12 125 10 96.9 3.12

MeanSE\_AC = freq\_df %>%  
 dplyr::group\_by(temp, depth\_cm) %>%  
 dplyr::summarise(avg\_AC = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))

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

MeanSE\_AC = MeanSE\_AC %>%   
 arrange(avg\_AC)  
  
print(MeanSE\_AC)

## # A tibble: 8 x 4  
## # Groups: temp [4]  
## temp depth\_cm avg\_AC se  
## <fct> <fct> <dbl> <dbl>  
## 1 185 0 0 0   
## 2 200 0 0 0   
## 3 155 0 12.5 6.53  
## 4 185 5 25 11.1   
## 5 200 5 25 11.1   
## 6 155 5 58.3 11.7   
## 7 125 0 85.4 6.50  
## 8 125 5 91.7 6.41

MeanSE\_BC = freq\_df %>%  
 dplyr::group\_by(time\_min, depth\_cm) %>%  
 dplyr::summarise(avg\_BC = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))

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

MeanSE\_BC = MeanSE\_BC %>%   
 arrange(avg\_BC)  
  
print(MeanSE\_BC)

## # A tibble: 6 x 4  
## # Groups: time\_min [3]  
## time\_min depth\_cm avg\_BC se  
## <fct> <fct> <dbl> <dbl>  
## 1 15 0 20.3 9.19  
## 2 10 0 25 10.5   
## 3 15 5 25 10.5   
## 4 5 0 28.1 10.2   
## 5 10 5 45.3 11.9   
## 6 5 5 79.7 5.69

MeanSE\_AD = freq\_df %>%  
 dplyr::group\_by(temp, weight\_g) %>%  
 dplyr::summarise(avg\_AD = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))

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

MeanSE\_AD = MeanSE\_AD %>%   
 arrange(avg\_AD)  
  
print(MeanSE\_AD)

## # A tibble: 16 x 4  
## # Groups: temp [4]  
## temp weight\_g avg\_AD se  
## <fct> <fct> <dbl> <dbl>  
## 1 185 0.054 8.33 8.33  
## 2 200 0.67 8.33 8.33  
## 3 185 0.222 12.5 12.5   
## 4 185 1.31 12.5 12.5   
## 5 200 0.054 12.5 12.5   
## 6 200 1.31 12.5 12.5   
## 7 185 0.67 16.7 16.7   
## 8 200 0.222 16.7 16.7   
## 9 155 0.222 25 15.8   
## 10 155 0.054 29.2 18.7   
## 11 155 0.67 41.7 19.0   
## 12 155 1.31 45.8 13.6   
## 13 125 0.054 62.5 12.5   
## 14 125 0.222 91.7 5.27  
## 15 125 0.67 100 0   
## 16 125 1.31 100 0

MeanSE\_ABC = freq\_df %>%  
 dplyr::group\_by(temp, time\_min, depth\_cm) %>%  
 dplyr::summarise(avg\_ABC = mean(freq),  
 se = sd(freq)/sqrt(length(freq)))

## `summarise()` has grouped output by 'temp', 'time\_min'. You can override using the `.groups` argument.

print(MeanSE\_ABC)

## # A tibble: 24 x 5  
## # Groups: temp, time\_min [12]  
## temp time\_min depth\_cm avg\_ABC se  
## <fct> <fct> <fct> <dbl> <dbl>  
## 1 125 5 0 87.5 7.22  
## 2 125 5 5 81.2 18.8   
## 3 125 10 0 93.8 6.25  
## 4 125 10 5 100 0   
## 5 125 15 0 75 17.7   
## 6 125 15 5 93.8 6.25  
## 7 155 5 0 25 17.7   
## 8 155 5 5 87.5 7.22  
## 9 155 10 0 6.25 6.25  
## 10 155 10 5 81.2 6.25  
## # ... with 14 more rows

### plot LSD

#### temp

p1 = ggplot(meanSE\_a, aes(x = temp,  
 y = avg\_a))+   
 geom\_bar(stat = "identity",  
 position = position\_dodge(width=0.9),  
 width = 0.6,  
   
 fill = "grey")+   
 geom\_errorbar(aes(ymax = avg\_a + se,  
 ymin = avg\_a - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+  
 labs(x = expression(paste("Temperature (",degree,"C)")),  
 y = "Mean germination (%)")+  
 geom\_text(aes(x = temp,  
 y = avg\_a + se,  
 label = as.matrix(ascend\_A$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/Temperature\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

#### time

p2 = ggplot(meanSE\_b, aes(x = time\_min,  
 y = avg\_b))+  
 geom\_bar(stat = "identity",  
 position = position\_dodge(width=0.9),  
 width = 0.6,  
 fill = "grey")+   
 geom\_errorbar(aes(ymax = avg\_b + se,  
 ymin = avg\_b - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+  
 labs(x = "Time (min)",  
 y = "Mean germination (%)")+  
 geom\_text(aes(x = time\_min,  
 y = avg\_b + se,  
 label = as.matrix(ascend\_B$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/time\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

#### depht

p3 = ggplot(meanSE\_c, aes(x = depth\_cm,  
 y = avg\_c))+  
 geom\_bar(stat = "identity",  
 position = position\_dodge(width=0.9),  
 width = 0.4,  
 fill = "grey")+   
 geom\_errorbar(aes(ymax = avg\_c + se,  
 ymin = avg\_c - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+  
 labs(x = "Depth (cm)",  
 y = "Mean germination (%)")+  
 geom\_text(aes(x = depth\_cm,  
 y = avg\_c + se,  
 label = as.matrix(ascend\_C$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/depth\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

#### weight

p4 = ggplot(meanSE\_d, aes(x = weight\_g,  
 y = avg\_d))+  
 geom\_bar(stat = "identity",  
 position = position\_dodge(width=0.9),  
 width = 0.6,  
 fill = "grey")+   
 geom\_errorbar(aes(ymax = avg\_d + se,  
 ymin = avg\_d - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+  
 geom\_text(aes(x = weight\_g,  
 y = avg\_d + se,  
 label = as.matrix(ascend\_D$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 labs(x = "Mean sclerotial Weight (g)",  
 y = "Mean germination (%)")+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/weight\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

#### interaction

##### temp x time

i1 = ggplot(MeanSE\_AB, aes(x = temp,  
 y = avg\_AB,  
 fill = factor(time\_min)))+   
 geom\_bar(stat = "identity",  
 color = "black",  
 position = position\_dodge(width=0.9))+   
 scale\_fill\_grey(start = 0.4, end = 0.8)+   
 geom\_errorbar(aes(ymax = avg\_AB + se,  
 ymin = avg\_AB - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+   
 labs(x = expression(paste("Temperature (",degree,"C)")),  
 y = "Mean germination (%)",  
 fill = "Time (min)")+   
 geom\_text(aes(x = temp,  
 y = avg\_AB + se,  
 label = as.matrix(ascend\_AB$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/temp\_time\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

##### temp x depth

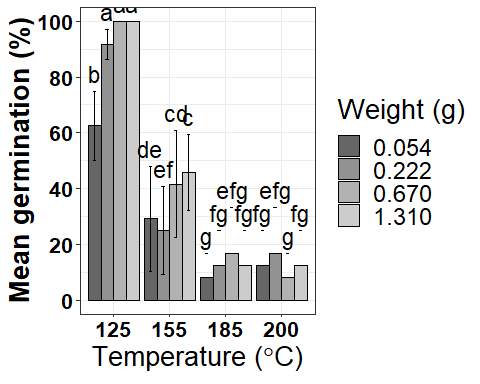
i2 = ggplot(MeanSE\_AC, aes(x = temp,  
 y = avg\_AC,  
 fill = factor(depth\_cm)))+   
 geom\_bar(stat = "identity",  
 color = "black",  
 position = position\_dodge(width=0.9))+   
 scale\_fill\_grey(start = 0.4, end = 0.8, labels = c("0", "5"))+   
 geom\_errorbar(aes(ymax = avg\_AC + se,  
 ymin = avg\_AC - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+   
 labs(x = expression(paste("Temperature (",degree,"C)")),  
 y = "Mean germination (%)",  
 fill = "Depth (cm)")+   
 geom\_text(aes(x = temp,  
 y = avg\_AC + se,  
 label = as.matrix(ascend\_AC$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/temp\_depth\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

##### time x depth

i3 = ggplot(MeanSE\_BC, aes(x = time\_min,  
 y = avg\_BC,  
 fill = factor(depth\_cm)))+   
 geom\_bar(stat = "identity",  
 color = "black",  
 position = position\_dodge(width=0.9))+   
 scale\_fill\_grey(start = 0.4, end = 0.8, labels = c("0", "5"))+   
 geom\_errorbar(aes(ymax = avg\_BC + se,  
 ymin = avg\_BC - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+   
 labs(x = "Time (min)",  
 y = "Mean germination (%)",  
 fill = "Depth (cm)")+   
 geom\_text(aes(x = time\_min,  
 y = avg\_BC + se,  
 label = as.matrix(ascend\_BC$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/time\_depth\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)

##### weight x temp

i4 <- ggplot(MeanSE\_AD, aes(x = temp,  
 y = avg\_AD,  
 fill = factor(weight\_g)))+   
 geom\_bar(stat = "identity",  
 color = "black",  
 position = position\_dodge(width=0.9))+   
 scale\_fill\_grey(start = 0.4, end = 0.8, labels = c("0.054", "0.222", "0.670", "1.310"))+   
 geom\_errorbar(aes(ymax = avg\_AD + se,  
 ymin = avg\_AD - se),   
 position = position\_dodge(width=0.9),   
 width = 0.25)+   
 labs(x = expression(paste("Temperature (",degree,"C)")),  
 y = "Mean germination (%)",  
 fill = "Weight (g)")+   
 geom\_text(aes(x = temp,  
 y = avg\_AD + se,  
 label = as.matrix(ascend\_AD$groups)),  
 size = 6,  
 position = position\_dodge(width = 0.9),  
 vjust = -(0.5))+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 theme\_bw()+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/temp\_weight\_lsd.png", units = "cm", width = 35, height = 20, dpi = 300)  
  
  
i4



# dung

## import

dung\_raw = read.xlsx("C:\\Users\\2010088819\\Documents\\R Projects\\Scle\_management\\manage.xlsx", "dung")  
  
  
dung\_raw$weight\_g <- as.factor(dung\_raw$weight\_g)

##ANOVA

weight\_g num germ\_num freq 0.030 53 2 3.773584906 0.262 24 0 0 0.328 16 2 12.5 0.400 8 0 0 0.610 3 0 0

### one way ANOVA

# 1 - modified  
  
weight\_g <- c(0.030, 0.262, 0.328, 0.400, 0.610)  
num <- c(53, 24, 16, 8, 3)  
germ\_num <- c(2, 0, 2, 0, 0)  
freq <- c(3.77, 0, 12.5, 0, 0)  
  
  
dung <- data.frame(  
 Y=c(weight\_g, num, freq),  
 variable = factor(rep(c("weight\_g", "num", "freq"),  
 times=c(length(weight\_g), length(num), length(freq)))))  
  
dung\_model <- aov(Y ~ variable, data = dung)  
anova(dung\_model)

## Analysis of Variance Table  
##   
## Response: Y  
## Df Sum Sq Mean Sq F value Pr(>F)   
## variable 2 1226.0 613.02 4.4089 0.03668 \*  
## Residuals 12 1668.5 139.04   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

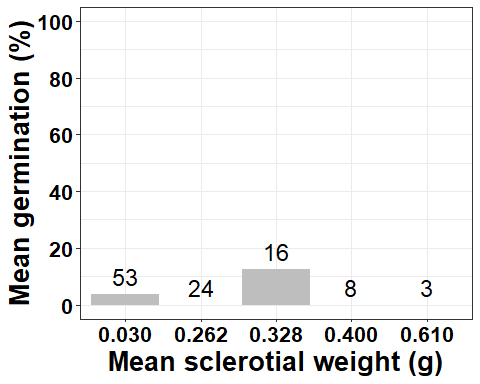
# 2 - raw  
  
aov\_d <- aov(germ\_num ~ weight\_g + num, data = dung\_raw)  
  
summary(aov\_d)

## Df Sum Sq Mean Sq  
## weight\_g 4 4.8 1.2

write.xlsx(dung\_model, "anova\_dung.xlsx")

## graphs

ggplot(aes(weight\_g, freq), data = dung\_raw)+  
 geom\_col( fill = "grey")+  
 geom\_text(aes(weight\_g, freq, label = num), hjust = 0.5, vjust = -0.5, size = 6)+  
 theme\_bw()+  
 labs(x = "Mean sclerotial weight (g)", y = "Mean germination (%)")+  
 scale\_y\_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 20))+  
 scale\_x\_discrete(labels=c("0.03" = "0.030", "0.262" = "0.262",  
 "0.328" = "0.328", "0.4" = "0.400", "0.61" = "0.610"))+  
 theme(axis.title.x=element\_text(size=20,face="bold"),  
 axis.title.y=element\_text(size=20,face="bold"),  
 axis.text.y = element\_text(size = 16, color = "black", face = "bold"),  
 axis.text.x = element\_text(size = 16, color = "black", face = "bold"),  
 legend.text = element\_text(size = 18),  
 legend.title = element\_text(size = 20),  
 strip.background = element\_rect(fill="white", size=1, color="black"))+  
 ggsave("plots/dung.png", units = "cm", width = 35, height = 20, dpi = 300)



,