Kyle\_Gibson\_assignment.R

kyle

Tue Sep 20 10:13:13 2022

#### Graphs R us assignment (Kyle Daniel Gibson)  
library(grid)  
library(gridExtra)  
library(ggplot2)  
library(ggplotify)  
library(gridGraphics)  
library(gridBase)  
library(reshape2)  
library(dplyr) # Loading installed Graphs R us packages

## Warning: package 'dplyr' was built under R version 3.5.3

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(raster)

## Loading required package: sp

##   
## Attaching package: 'raster'

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

library(ggmap)  
library(ggthemes)  
library(GGally)

##   
## Attaching package: 'GGally'

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

library(viridis)

## Loading required package: viridisLite

library(viridisLite)  
library(colorRamps)  
library(jpeg)  
library(png)  
library(mapproj)

## Loading required package: maps

library(maptools)

## Checking rgeos availability: TRUE

library(RColorBrewer)  
library(RStoolbox)  
library(mapdata)  
library(broom)  
library(rmapshaper)  
library(ggridges)

##   
## Attaching package: 'ggridges'

## The following object is masked from 'package:ggplot2':  
##   
## scale\_discrete\_manual

library(devtools)  
library(gapminder)  
library(ggpubr)

## Loading required package: magrittr

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:ggmap':  
##   
## inset

## The following object is masked from 'package:raster':  
##   
## extract

##   
## Attaching package: 'ggpubr'

## The following object is masked from 'package:raster':  
##   
## rotate

library(ggridges)  
library(ggiraph)  
library(knitr)  
library(rmarkdown)  
  
  
  
setwd("C:\\AUT\\Graphs R us") # Set working directory  
getwd() # Checking working directory location is correct

## [1] "C:/AUT/Graphs R us"

cs <- read.csv("chilling\_sensitivity.csv") # Reading in datasets from my working directory  
de <- read.csv("drought\_elevation.csv")   
  
#### Data exploration of chilling sensitivity ####  
  
str(cs) # Checking overall dataframe type and variable type and amount

## 'data.frame': 168 obs. of 5 variables:  
## $ plant: int 1 1 1 1 1 1 1 2 2 2 ...  
## $ spec : Factor w/ 4 levels "Cynosurus cristatus",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ treat: Factor w/ 2 levels "chilled","nonchilled": 2 2 2 2 2 2 2 2 2 2 ...  
## $ conc : int 95 175 250 350 500 675 1000 95 175 250 ...  
## $ photo: num 16 30.4 34.8 37.2 35.3 39.2 39.7 13.6 27.3 37.1 ...

head(cs) # Viewing columns of first 6 rows

## plant spec treat conc photo  
## 1 1 Lolium perenne nonchilled 95 16.0  
## 2 1 Lolium perenne nonchilled 175 30.4  
## 3 1 Lolium perenne nonchilled 250 34.8  
## 4 1 Lolium perenne nonchilled 350 37.2  
## 5 1 Lolium perenne nonchilled 500 35.3  
## 6 1 Lolium perenne nonchilled 675 39.2

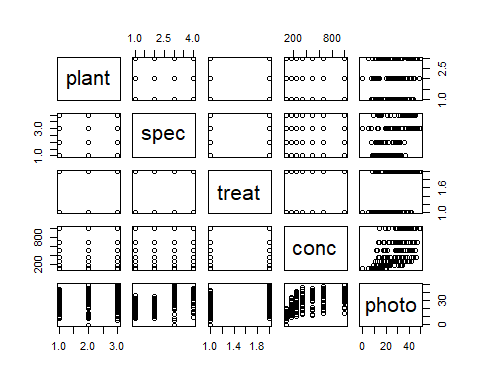
tail(cs) # Viewing columns of last 6 rows

## plant spec treat conc photo  
## 163 3 Cynosurus cristatus chilled 175 19.81030  
## 164 3 Cynosurus cristatus chilled 250 18.56421  
## 165 3 Cynosurus cristatus chilled 350 19.53616  
## 166 3 Cynosurus cristatus chilled 500 19.04634  
## 167 3 Cynosurus cristatus chilled 675 17.52267  
## 168 3 Cynosurus cristatus chilled 1000 21.72192

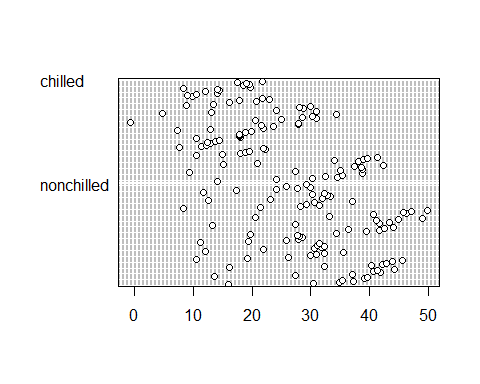
summary(cs) # Viewing summary statistics

## plant spec treat conc   
## Min. :1 Cynosurus cristatus:42 chilled :84 Min. : 95   
## 1st Qu.:1 Dactylis glomerata :42 nonchilled:84 1st Qu.: 175   
## Median :2 Holcus mollis :42 Median : 350   
## Mean :2 Lolium perenne :42 Mean : 435   
## 3rd Qu.:3 3rd Qu.: 675   
## Max. :3 Max. :1000   
## photo   
## Min. :-0.6157   
## 1st Qu.:17.8941   
## Median :27.7478   
## Mean :26.4240   
## 3rd Qu.:34.8500   
## Max. :49.8769

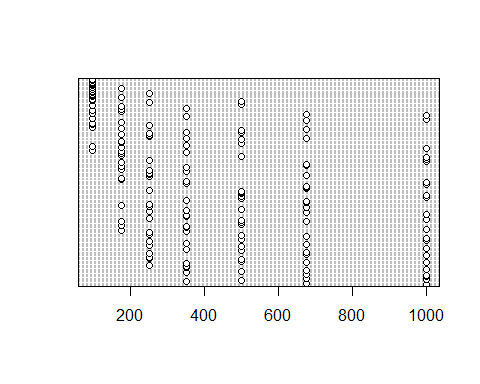
View(cs) # Complete view of dataset  
plot(cs) # Looking for obvious patterns between variables



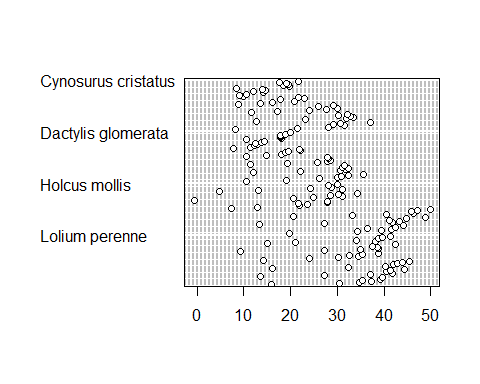
dotchart(x = cs$photo, groups = cs$treat) # Exploratory dot chart view



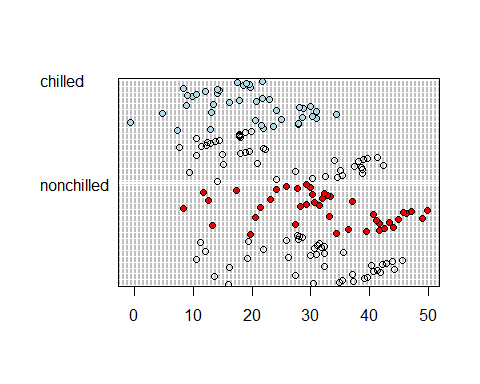
dotchart(x = cs$conc, groups = cs$photo)



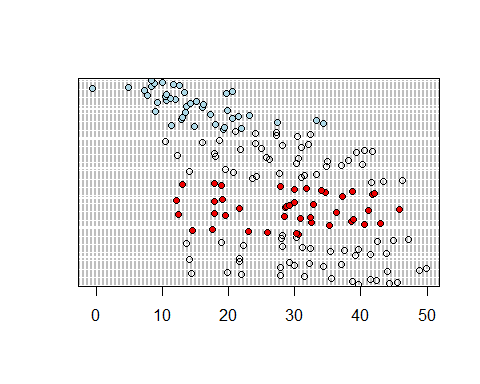
dotchart(x = cs$photo, groups = cs$spec)



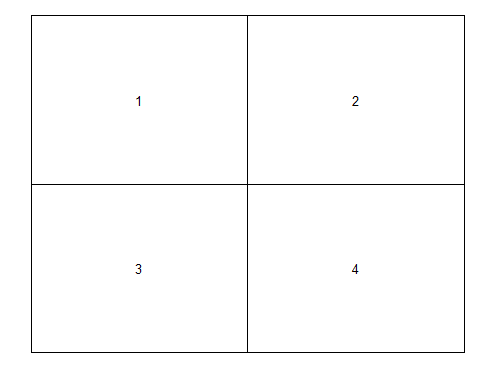
colo <- c("lightblue", "red2")  
dotchart(x = cs$photo, groups = cs$treat, # Exploring relationship between species, treatment and photosynthetic rate  
 bg = colo[as.numeric(cs$spec)])



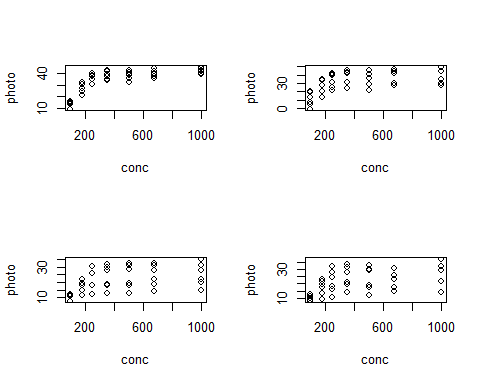
dotchart(x = cs$photo, groups = cs$conc, # Exploring relationship between species, carbon dioxide concentration and photosynthetic rate  
 bg = colo[as.numeric(cs$spec)])



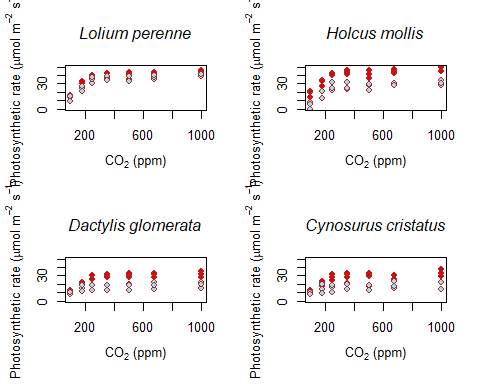
op <- par(mar = rep(1.6, 4), oma = c(2, 2, 1, 1)) # Turning off margins and creating custom margins  
layout(rbind(c(1, 2), # Creating custom 4 plot layout  
 c(3, 4)),  
 heights = c(1, 1, 1, 1), widths = c(1, 1, 1, 1))  
layout.show(4) # Checking layout in plot window



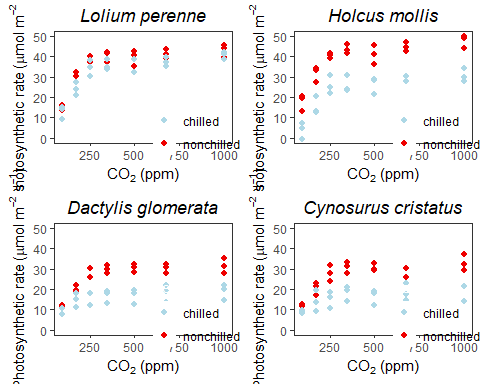
par(op)  
  
  
subspec <- subset(cs, spec == "Lolium perenne") # subsetting dataframe to individual species including all replicas and all columns  
subspec1 <- subset(cs, spec == "Holcus mollis")  
subspec2 <- subset(cs, spec == "Dactylis glomerata")  
subspec3 <- subset(cs, spec == "Cynosurus cristatus")  
  
subspec <- droplevels(subspec)   
subspec1 <- droplevels(subspec1) # Removing species rows that are not needed to clean sub-setted data   
subspec2 <- droplevels(subspec2)  
subspec3 <- droplevels(subspec3)  
  
plot(photo ~ conc, data = subspec) # Plotting each species to check scatter plot visually   
plot(photo ~ conc, data = subspec1)  
plot(photo ~ conc, data = subspec2)  
plot(photo ~ conc, data = subspec3)



#### Multi-panel scatter plot (Traditional graphics) ####  
  
#pdf("scatter.pdf", width = 7, height = 7) # Exporting multi-panel plot  
op <- par(mar = rep(1.6, 4), oma = c(2, 2, 1, 1)) # Turning off margins and creating custom margins  
layout(rbind(c(1, 2), # Creating custom 4 plot layout  
 c(3, 4)),  
 heights = c(1, 1, 1, 1), widths = c(1, 1, 1, 1))  
   
par(op)  
  
plot(subspec$conc, subspec$photo, pch = 21, col = "brown", bg = colo[as.numeric(subspec$treat)], ylab = NA, xlab = NA, ylim = c(0, 50)) # Plot concentration and photosynthetic rate for Lolium perenne with colour representing chilled and nonchilled  
legend(x = 746, y = 8, legend = c("Nonchilled", "Chilled"), pch = c(22, 22), # Adding legend to show control and treatment colours  
 pt.bg = c("red2", "lightblue"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
mtext(expression(paste("Photosynthetic rate (", mu, "mol m"^-2, " s"^-1, ")")), side = 2, line = 2.5, cex = 0.8)  
mtext(expression(paste("CO"[2], " (ppm)")), side = 1, line = 2.75, cex = 0.8) # Adding custom labels with correct symbols and font  
mtext("Lolium perenne", font = 3, side = 3, line = 1.5, cex = 1)  
  
plot(subspec1$conc, subspec1$photo, pch = 21, col = "brown", bg = colo[as.numeric(subspec1$treat)], ylab = NA, xlab = NA, ylim = c(0, 50)) # y limit set to allow better visual comparison between each plot  
legend(x = 746, y = 8, legend = c("Nonchilled", "Chilled"), pch = c(22, 22),   
 pt.bg = c("red2", "lightblue"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
mtext(expression(paste("Photosynthetic rate (", mu, "mol m"^-2, " s"^-1, ")")), side = 2, line = 2.5, cex = 0.8) # Plots repeated for all species  
mtext(expression(paste("CO"[2], " (ppm)")), side = 1, line = 2.75, cex = 0.8)   
mtext("Holcus mollis", font = 3, side = 3, line = 1.5, cex = 1)  
  
plot(subspec2$conc, subspec2$photo, pch = 21, col = "brown", bg = colo[as.numeric(subspec2$treat)], ylab = NA, xlab = NA, ylim = c(0, 50)) # data point outline colour changed so each point is easily seen  
legend(x = 746, y = 8, legend = c("Nonchilled", "Chilled"), pch = c(22, 22),   
 pt.bg = c("red2", "lightblue"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
mtext(expression(paste("Photosynthetic rate (", mu, "mol m"^-2, " s"^-1, ")")), side = 2, line = 2.5, cex = 0.8)   
mtext(expression(paste("CO"[2], " (ppm)")), side = 1, line = 2.75, cex = 0.8)   
mtext("Dactylis glomerata", font = 3, side = 3, line = 1.5, cex = 1)  
  
plot(subspec3$conc, subspec3$photo, pch = 21, col = "brown", bg = colo[as.numeric(subspec3$treat)], ylab = NA, xlab = NA, ylim = c(0, 50))   
legend(x = 746, y = 8, legend = c("Nonchilled", "Chilled"), pch = c(22, 22),   
 pt.bg = c("red2", "lightblue"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
mtext(expression(paste("Photosynthetic rate (", mu, "mol m"^-2, " s"^-1, ")")), side = 2, line = 2.5, cex = 0.8)  
mtext(expression(paste("CO"[2], " (ppm)")), side = 1, line = 2.75, cex = 0.8)   
mtext("Cynosurus cristatus", font = 3, side = 3, line = 1.5, cex = 1)



#dev.off() # Stop exporting  
  
  
#### Multi-panel scatter plot (ggplot) ####  
  
  
p1 <- ggplot(data = subspec, aes(x = conc, y = photo, color = treat)) + geom\_point(size = 3, shape = 20)+  
 scale\_colour\_manual(values=c("lightblue", "red2"))+ coord\_cartesian(ylim = c(0, 50))+ # Swapping point colours to 'chilled = blue' to make more sense to the reader, setting y axis limit to provide the same area for each plot  
labs(x = expression(paste("CO"[2], " (ppm)")), y = expression(paste("Photosynthetic rate", " (", mu, "mol m"^-2, " s"^-1, ")")))+  
theme\_bw()+theme(panel.grid = element\_blank(), legend.title = element\_text(colour = "white"), plot.title = element\_text(hjust = 0.5, face = "italic"), legend.background = element\_rect(fill="transparent"))+ # Move plot title to center and change font to italic  
 ggtitle("Lolium perenne")+ theme(legend.position = c(0.8, 0.2))  
  
p2 <- ggplot(data = subspec1, aes(x = conc, y = photo, color = treat)) + geom\_point(size = 3, shape = 20)+  
 scale\_colour\_manual(values=c("lightblue", "red2"))+ coord\_cartesian(ylim = c(0, 50))+   
 labs(x = expression(paste("CO"[2], " (ppm)")), y = expression(paste("Photosynthetic rate", " (", mu, "mol m"^-2, " s"^-1, ")")))+  
 theme\_bw()+theme(panel.grid = element\_blank(), legend.title = element\_text(colour = "white"), plot.title = element\_text(hjust = 0.5, face = "italic"), legend.background = element\_rect(fill="transparent"))+   
 ggtitle("Holcus mollis")+ theme(legend.position = c(0.8, 0.2))  
   
p3 <- ggplot(data = subspec2, aes(x = conc, y = photo, color = treat)) + geom\_point(size = 3, shape = 20)+  
 scale\_colour\_manual(values=c("lightblue", "red2"))+ coord\_cartesian(ylim = c(0, 50))+   
 labs(x = expression(paste("CO"[2], " (ppm)")), y = expression(paste("Photosynthetic rate", " (", mu, "mol m"^-2, " s"^-1, ")")))+  
 theme\_bw()+theme(panel.grid = element\_blank(), legend.title = element\_text(colour = "white"), plot.title = element\_text(hjust = 0.5, face = "italic"), legend.background = element\_rect(fill="transparent"))+   
 ggtitle("Dactylis glomerata")+ theme(legend.position = c(0.8, 0.2))  
  
p4 <- ggplot(data = subspec3, aes(x = conc, y = photo, color = treat)) + geom\_point(size = 3, shape = 20)+  
 scale\_colour\_manual(values=c("lightblue", "red2"))+ coord\_cartesian(ylim = c(0, 50))+   
 labs(x = expression(paste("CO"[2], " (ppm)")), y = expression(paste("Photosynthetic rate", " (", mu, "mol m"^-2, " s"^-1, ")")))+  
 theme\_bw()+theme(panel.grid = element\_blank(), legend.title = element\_text(colour = "white"), plot.title = element\_text(hjust = 0.5, face = "italic"), legend.background = element\_rect(fill="transparent"))+   
 ggtitle("Cynosurus cristatus")+ theme(legend.position = c(0.8, 0.2))  
  
#pdf("scatterggplot.pdf", width = 7, height = 7) # Exporting the below multi-panel plot  
grid.arrange(p1, p2, p3, p4, ncol=2) # Arranging my plots into grids to create a four panel plot (2 rows+2columns)



#dev.off() # Stop exporting  
  
#### Data exploration of drought elevation ####  
  
str(de) # Checking overall dataframe type and variable type and amount

## 'data.frame': 180 obs. of 4 variables:  
## $ spec : Factor w/ 3 levels "A. australis",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ elevation: Factor w/ 3 levels "high","low","mid": 2 2 2 2 2 2 2 2 2 2 ...  
## $ treat : Factor w/ 2 levels "ctrl","drought": 1 1 1 1 1 1 1 1 1 1 ...  
## $ height : num 24.9 25.3 25.7 33 31.2 25.5 20.8 28.6 26.8 24.5 ...

head(de) # Viewing columns of first 6 rows

## spec elevation treat height  
## 1 A. australis low ctrl 24.9  
## 2 A. australis low ctrl 25.3  
## 3 A. australis low ctrl 25.7  
## 4 A. australis low ctrl 33.0  
## 5 A. australis low ctrl 31.2  
## 6 A. australis low ctrl 25.5

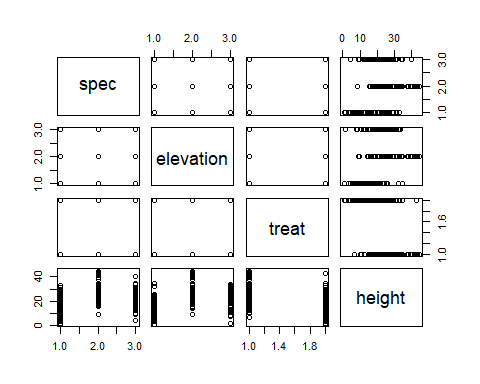
tail(de) # Viewing columns of last 6 rows

## spec elevation treat height  
## 175 B. tawa high drought 16.1  
## 176 B. tawa high drought 18.9  
## 177 B. tawa high drought 23.1  
## 178 B. tawa high drought 20.0  
## 179 B. tawa high drought 8.7  
## 180 B. tawa high drought 25.4

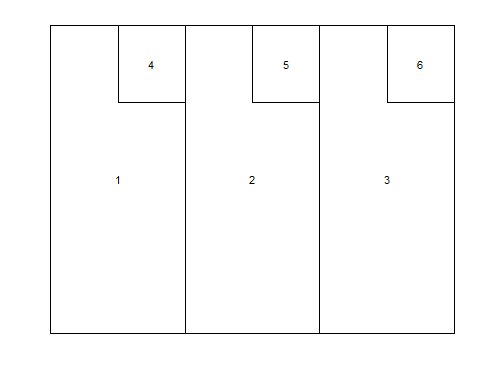
summary(de) # Viewing summary statistics

## spec elevation treat height   
## A. australis:60 high:60 ctrl :90 Min. : 0.60   
## B. tawa :60 low :60 drought:90 1st Qu.:15.97   
## P. totara :60 mid :60 Median :22.20   
## Mean :22.16   
## 3rd Qu.:27.23   
## Max. :44.70

View(de) # Complete view of dataset  
plot(de) # View of all factors



se <- function(x, na.rm = FALSE)  
{  
 if(na.rm == TRUE)  
 {  
 sqrt(var(x, na.rm = T)/length(na.omit(x))) # Creating standard error function  
 }  
 else  
 {  
 sqrt(var(x)/length(x))  
 }  
}  
  
  
  
op <- par(mar = rep(0, 4), oma = c(4, 4, 2, 2)) # Changing the plot window margins   
  
## Create the layout  
layout(rbind(c(1, 4, 2, 5, 3, 6),  
 c(1, 1, 2, 2, 3, 3), # Creating layout of 3 columns with one inset per column  
 c(1, 1, 2, 2, 3, 3),  
 c(1, 1, 2, 2, 3, 3)))  
  
 layout.show(n = 6)



par(op) # Back to original parameters (plot margins default)  
   
 de1 <- subset(de, elevation == "low") # subsetting dataframe to individual elevations including all replicas and all columns  
 de2 <- subset(de, elevation == "mid")  
 de3 <- subset(de, elevation == "high")  
   
 summary(de1)

## spec elevation treat height   
## A. australis:20 high: 0 ctrl :30 Min. : 8.80   
## B. tawa :20 low :60 drought:30 1st Qu.:22.62   
## P. totara :20 mid : 0 Median :27.25   
## Mean :28.07   
## 3rd Qu.:34.10   
## Max. :44.70

summary(de2) # Checking rows and columns

## spec elevation treat height   
## A. australis:20 high: 0 ctrl :30 Min. : 1.70   
## B. tawa :20 low : 0 drought:30 1st Qu.:17.90   
## P. totara :20 mid :60 Median :22.70   
## Mean :22.04   
## 3rd Qu.:26.50   
## Max. :33.50

summary(de3)

## spec elevation treat height   
## A. australis:20 high:60 ctrl :30 Min. : 0.600   
## B. tawa :20 low : 0 drought:30 1st Qu.: 9.875   
## P. totara :20 mid : 0 Median :16.750   
## Mean :16.377   
## 3rd Qu.:20.750   
## Max. :34.800

de1 <- droplevels(de1)   
 de2 <- droplevels(de2) # Removing elevation rows that are not needed to clean sub-setted data   
 de3 <- droplevels(de3)  
  
   
de4 <- group\_by(.data = de1, spec, treat) %>% summarise(ps\_av = mean(height), pos = mean(height) + se(height), neg = mean(height) - se(height)) # Creating new dataset with average height and standard error for plotting  
de4

## # A tibble: 6 x 5  
## # Groups: spec [3]  
## spec treat ps\_av pos neg  
## <fct> <fct> <dbl> <dbl> <dbl>  
## 1 A. australis ctrl 26.6 27.7 25.5  
## 2 A. australis drought 17.5 19.2 15.8  
## 3 B. tawa ctrl 40.2 41.2 39.2  
## 4 B. tawa drought 30.1 32.1 28.2  
## 5 P. totara ctrl 31.1 32.7 29.5  
## 6 P. totara drought 22.9 24.4 21.4

summary(de4) # Checking new dataset with averages

## spec treat ps\_av pos   
## A. australis:2 ctrl :3 Min. :17.46 Min. :19.17   
## B. tawa :2 drought:3 1st Qu.:23.84 1st Qu.:25.24   
## P. totara :2 Median :28.38 Median :29.90   
## Mean :28.07 Mean :29.55   
## 3rd Qu.:30.84 3rd Qu.:32.53   
## Max. :40.23 Max. :41.22   
## neg   
## Min. :15.75   
## 1st Qu.:22.44   
## Median :26.85   
## Mean :26.60   
## 3rd Qu.:29.15   
## Max. :39.24

de5 <- group\_by(.data = de2, spec, treat) %>% summarise(ps\_av = mean(height), pos = mean(height) + se(height), neg = mean(height) - se(height))   
de5

## # A tibble: 6 x 5  
## # Groups: spec [3]  
## spec treat ps\_av pos neg  
## <fct> <fct> <dbl> <dbl> <dbl>  
## 1 A. australis ctrl 22.1 23.8 20.4  
## 2 A. australis drought 14.0 16.3 11.8  
## 3 B. tawa ctrl 29.4 30.3 28.5  
## 4 B. tawa drought 22.2 23.0 21.4  
## 5 P. totara ctrl 25.6 26.9 24.2  
## 6 P. totara drought 18.9 20.7 17.1

summary(de5)

## spec treat ps\_av pos   
## A. australis:2 ctrl :3 Min. :14.05 Min. :16.30   
## B. tawa :2 drought:3 1st Qu.:19.68 1st Qu.:21.24   
## P. totara :2 Median :22.16 Median :23.39   
## Mean :22.04 Mean :23.49   
## 3rd Qu.:24.71 3rd Qu.:26.10   
## Max. :29.43 Max. :30.34   
## neg   
## Min. :11.80   
## 1st Qu.:17.93   
## Median :20.92   
## Mean :20.58   
## 3rd Qu.:23.54   
## Max. :28.52

de6 <- group\_by(.data = de3, spec, treat) %>% summarise(ps\_av = mean(height), pos = mean(height) + se(height), neg = mean(height) - se(height))   
de6

## # A tibble: 6 x 5  
## # Groups: spec [3]  
## spec treat ps\_av pos neg  
## <fct> <fct> <dbl> <dbl> <dbl>  
## 1 A. australis ctrl 14.6 16.0 13.1   
## 2 A. australis drought 6.22 7.19 5.25  
## 3 B. tawa ctrl 25.8 27.9 23.8   
## 4 B. tawa drought 18.5 19.9 17.1   
## 5 P. totara ctrl 20.2 21.3 19.1   
## 6 P. totara drought 12.9 14.5 11.4

summary(de6)

## spec treat ps\_av pos   
## A. australis:2 ctrl :3 Min. : 6.22 Min. : 7.193   
## B. tawa :2 drought:3 1st Qu.:13.35 1st Qu.:14.886   
## P. totara :2 Median :16.54 Median :17.980   
## Mean :16.38 Mean :17.799   
## 3rd Qu.:19.78 3rd Qu.:20.932   
## Max. :25.82 Max. :27.873   
## neg   
## Min. : 5.247   
## 1st Qu.:11.809   
## Median :15.100   
## Mean :14.954   
## 3rd Qu.:18.623   
## Max. :23.767

#### Multi-panel barplots with insets ####  
  
  
#pdf("barplot.pdf", width = 6, height = 8) # Exporting multi-panel plot  
op <- par(mar = rep(0, 4), oma = c(3, 6, 2, 4) + 0.1) # Changing the plot window margins   
layout(rbind(c(1, 1, 1, 0, 4),  
 c(1, 1, 1, 0, 1),  
 c(0, 0, 0, 0, 0), # Changed layout for better viewing  
 c(2, 2, 2, 0, 5),  
 c(2, 2, 2, 0, 2),  
 c(0, 0, 0, 0, 0),  
 c(3, 3, 3, 0, 6),  
 c(3, 3, 3, 0, 3)))  
  
#layout.show(n = 6) # Layout shown  
  
  
  
bp <- barplot(height = de4$ps\_av, las = 1, axes = T, ylim = c(0, 45), xlim = c(0, 10), col = c("lightblue", "darkgrey"), main = " Low altitude", xpd = NA)  
arrows(x0 = bp, y0 = de4$pos, x1 = bp, y1 = de4$neg, angle = 90, length = 0.04, code = 3)  
mtext("Growth height (cm)", side = 2, line = 2.5, cex = 0.8)  
text(x = 1.9, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 0.68, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)   
text(x = 3.1, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 4.3, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 5.5, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 6.7, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 1.2, y = -1, labels = "A. australis", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
text(x = 3.6, y = -1, labels = "B. tawa", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
text(x = 6, y = -1, labels = "P. totara", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
legend(x = 0, y = 40, legend = c("Control", "Rain exclusion"), pch = c(22, 22),   
 pt.bg = c("lightblue", "darkgrey"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
  
  
  
de5 <- group\_by(.data = de2, spec, treat) %>% summarise(ps\_av = mean(height), pos = mean(height) + se(height), neg = mean(height) - se(height))   
de5

## # A tibble: 6 x 5  
## # Groups: spec [3]  
## spec treat ps\_av pos neg  
## <fct> <fct> <dbl> <dbl> <dbl>  
## 1 A. australis ctrl 22.1 23.8 20.4  
## 2 A. australis drought 14.0 16.3 11.8  
## 3 B. tawa ctrl 29.4 30.3 28.5  
## 4 B. tawa drought 22.2 23.0 21.4  
## 5 P. totara ctrl 25.6 26.9 24.2  
## 6 P. totara drought 18.9 20.7 17.1

summary(de5)

## spec treat ps\_av pos   
## A. australis:2 ctrl :3 Min. :14.05 Min. :16.30   
## B. tawa :2 drought:3 1st Qu.:19.68 1st Qu.:21.24   
## P. totara :2 Median :22.16 Median :23.39   
## Mean :22.04 Mean :23.49   
## 3rd Qu.:24.71 3rd Qu.:26.10   
## Max. :29.43 Max. :30.34   
## neg   
## Min. :11.80   
## 1st Qu.:17.93   
## Median :20.92   
## Mean :20.58   
## 3rd Qu.:23.54   
## Max. :28.52

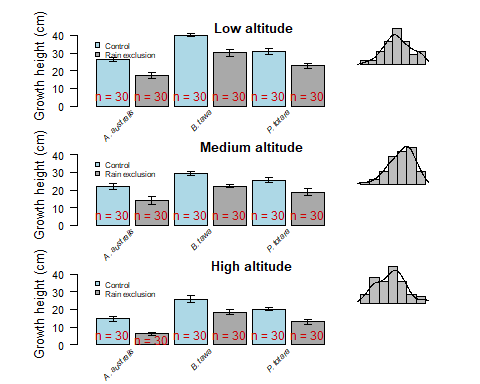
bp <- barplot(height = de5$ps\_av, las = 1, axes = T, ylim = c(0, 45), xlim = c(0, 10), col = c("lightblue", "darkgrey"), main = "Medium altitude", xpd = NA)  
arrows(x0 = bp, y0 = de5$pos, x1 = bp, y1 = de5$neg, angle = 90, length = 0.04, code = 3)  
mtext("Growth height (cm)", side = 2, line = 2.5, cex = 0.8)  
text(x = 1.9, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 0.68, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)   
text(x = 3.1, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 4.3, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 5.5, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 6.7, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 1.2, y = -1, labels = "A. australis", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
text(x = 3.6, y = -1, labels = "B. tawa", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
text(x = 6, y = -1, labels = "P. totara", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
legend(x = 0, y = 40, legend = c("Control", "Rain exclusion"), pch = c(22, 22),   
 pt.bg = c("lightblue", "darkgrey"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
  
  
de6 <- group\_by(.data = de3, spec, treat) %>% summarise(ps\_av = mean(height), pos = mean(height) + se(height), neg = mean(height) - se(height))   
de6

## # A tibble: 6 x 5  
## # Groups: spec [3]  
## spec treat ps\_av pos neg  
## <fct> <fct> <dbl> <dbl> <dbl>  
## 1 A. australis ctrl 14.6 16.0 13.1   
## 2 A. australis drought 6.22 7.19 5.25  
## 3 B. tawa ctrl 25.8 27.9 23.8   
## 4 B. tawa drought 18.5 19.9 17.1   
## 5 P. totara ctrl 20.2 21.3 19.1   
## 6 P. totara drought 12.9 14.5 11.4

summary(de6)

## spec treat ps\_av pos   
## A. australis:2 ctrl :3 Min. : 6.22 Min. : 7.193   
## B. tawa :2 drought:3 1st Qu.:13.35 1st Qu.:14.886   
## P. totara :2 Median :16.54 Median :17.980   
## Mean :16.38 Mean :17.799   
## 3rd Qu.:19.78 3rd Qu.:20.932   
## Max. :25.82 Max. :27.873   
## neg   
## Min. : 5.247   
## 1st Qu.:11.809   
## Median :15.100   
## Mean :14.954   
## 3rd Qu.:18.623   
## Max. :23.767

bp <- barplot(height = de6$ps\_av, las = 1, axes = T, ylim = c(0, 45), xlim = c(0, 10), col = c("lightblue", "darkgrey"), main = "High altitude", xpd = NA)  
arrows(x0 = bp, y0 = de6$pos, x1 = bp, y1 = de6$neg, angle = 90, length = 0.04, code = 3)  
mtext("Growth height (cm)", side = 2, line = 2.5, cex = 0.8)  
text(x = 1.9, y = 3, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 0.68, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)   
text(x = 3.1, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 4.3, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 5.5, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 6.7, y = 6, labels = "n = 30", srt = 0, col = "red3", cex = 1.1)  
text(x = 1.2, y = -1, labels = "A. australis", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
text(x = 3.6, y = -1, labels = "B. tawa", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
text(x = 6, y = -1, labels = "P. totara", srt = 45, cex = 0.8, adj = c(1, 1), xpd = NA, font = 3)  
legend(x = 0, y = 40, legend = c("Control", "Rain exclusion"), pch = c(22, 22),   
 pt.bg = c("lightblue", "darkgrey"), bty = "n", y.intersp = 0.9, pt.cex = 1, cex = 0.8)  
  
#### Insets ####  
 # Compute and add a probability density curve  
hist(de1$height, col = "grey", main = NA, xlab = NA, las = 1, ylab = NA, freq = F, axes = F)  
dens1 <- density(de1$height)  
lines(dens1)  
  
 # Compute and add a probability density curve  
hist(de2$height, col = "grey", main = NA, xlab = NA, ylab = NA, axes = F, las = 1, freq = F)  
dens2 <- density(de2$height)  
lines(dens2)  
  
 # Compute and add a probability density curve  
hist(de3$height, col = "grey", main = NA, xlab = NA, ylab = NA, axes = F, las = 1, freq = F)  
dens3 <- density(de3$height)  
lines(dens3)



par(op) # Back to original parameters (plot margins return to default)  
#dev.off() # Stop exporting  
  
#### Multi-panel map ####  
  
#### Exploration ####  
  
aus <- getData(name = "GADM", country = "Australia", level = 1)  
nrow(aus)

## [1] 11

summary(aus)

## Object of class SpatialPolygonsDataFrame  
## Coordinates:  
## min max  
## x 112.92111 159.109222  
## y -55.11694 -9.142176  
## Is projected: FALSE   
## proj4string :  
## [+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0]  
## Data attributes:  
## GID\_0 NAME\_0 GID\_1   
## Length:11 Length:11 Length:11   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
## NAME\_1 VARNAME\_1 NL\_NAME\_1   
## Length:11 Length:11 Length:11   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
## TYPE\_1 ENGTYPE\_1 CC\_1   
## Length:11 Length:11 Length:11   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
## HASC\_1   
## Length:11   
## Class :character   
## Mode :character

unique(aus$NAME\_1)

## [1] "Ashmore and Cartier Islands" "Australian Capital Territory"  
## [3] "Coral Sea Islands Territory" "Jervis Bay Territory"   
## [5] "New South Wales" "Northern Territory"   
## [7] "Queensland" "South Australia"   
## [9] "Tasmania" "Victoria"   
## [11] "Western Australia"

aus$NAME\_1

## [1] "Ashmore and Cartier Islands" "Australian Capital Territory"  
## [3] "Coral Sea Islands Territory" "Jervis Bay Territory"   
## [5] "New South Wales" "Northern Territory"   
## [7] "Queensland" "South Australia"   
## [9] "Tasmania" "Victoria"   
## [11] "Western Australia"

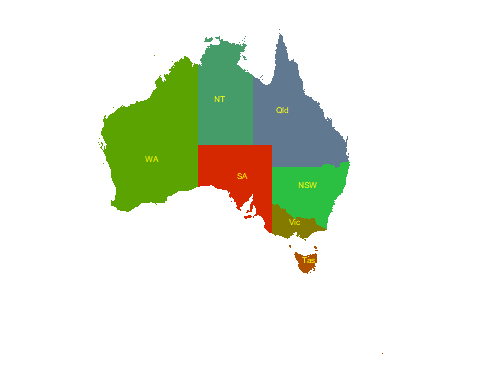
aus1 <- aus[aus$NAME\_1 %in% c( "New South Wales", "Northern Territory", "Queensland", "South Australia", "Tasmania", "Victoria", "Western Australia"), ] # Choosing the variables required for dataset and excluding the rest  
aus1$NAME\_1 # Dataset with states only

## [1] "New South Wales" "Northern Territory" "Queensland"   
## [4] "South Australia" "Tasmania" "Victoria"   
## [7] "Western Australia"

#### Map ####  
  
cols <- colorRampPalette(colors = c("red", "green", "purple", "orange", "brown")) # Creating colour function to use as a visual difference between each state  
  
map2 <- (ggplot(data = aus, aes(x = long, y = lat, group = group, map\_id = id)) +  
 geom\_polygon(aes(fill = id), size = 0.2) + # Creating a map of Australia with state borders and state labels with the corresponding abbreviations  
 coord\_quickmap() +   
 guides(fill = F) +   
 scale\_fill\_manual(values = cols(n = 26)) +  
annotate(geom = "text", x = 146.9211, y = -31.2532, label = "NSW", col = "yellow", size = 2) +  
annotate(geom = "text", x = 132.5510, y = -19.4914, label = "NT", size = 2, col = "yellow") +   
annotate(geom = "text", x = 142.7028, y = -20.9176, label = "Qld", size = 2, col = "yellow") +  
annotate(geom = "text", x = 136.2092, y = -30.0002, label = "SA", size = 2, col = "yellow") +  
annotate(geom = "text", x = 147, y = -41.4545, label = "Tas", size = 2, col = "yellow") +  
annotate(geom = "text", x = 144.7852, y = -36.3, label = "Vic", size = 2, col = "yellow") +  
annotate(geom = "text", x = 121.6283, y = -27.6728, label = "WA", size =2, col = "yellow") +  
theme\_map())

## Regions defined for each Polygons

map2 # Viewing map



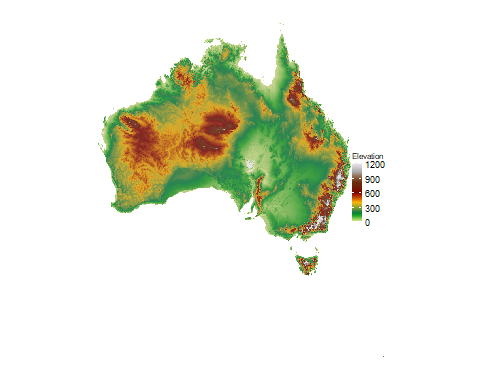
#### Digital elevation model ####  
  
alt.aus <- getData(name = "alt", country = "AUS") # Import Australian altitude data  
  
## Compute the slope, aspect and other terrain characteristics from raster object  
  
terr.aus1 <- terrain(x = alt.aus, opt = c("slope", "aspect"))  
  
  
## Compute the hill shade from the slope and aspect layer   
hill.aus1 <- hillShade(slope = terr.aus1$slope, aspect = terr.aus1$aspect)  
  
  
## Merge the two data sets  
hill.aus2 <- merge(hill.aus1, terr.aus1, overlap = T)   
alt.aus

## class : RasterLayer   
## dimensions : 5496, 5568, 30601728 (nrow, ncol, ncell)  
## resolution : 0.008333333, 0.008333333 (x, y)  
## extent : 112.8, 159.2, -54.9, -9.1 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0   
## data source : C:\AUT\Graphs R us\AUS\_msk\_alt.grd   
## names : AUS\_msk\_alt   
## values : -43, 2143 (min, max)

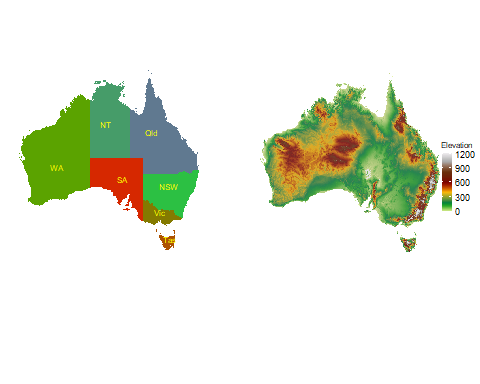
aoi <- extent(c(xmin = 112.8, xmax = 159.2, ymin = -54.9, ymax = -9.1))   
# aoi = area of interest  
  
## Use aoi as reference to crop the hill shade raster  
hill.aus3 <- crop(x = hill.aus2, y = aoi)  
  
## Same for the elevations  
alt.aus3 <- crop(x = alt.aus, y = aoi)  
  
  
  
## Colour palette for the DEM  
library(ggplot2)  
library(raster)  
library(RColorBrewer)  
  
schwarzwald <- colorRampPalette(c(  
 # rgb(174, 239, 213, max = 255), # Schwarzwald colour palette function created  
 # rgb(175, 240, 211, max = 255),  
 # rgb(176, 242, 208, max = 255),  
 # rgb(176, 242, 202, max = 255),  
 # rgb(177, 242, 196, max = 255),  
 # rgb(176, 243, 190, max = 255),  
 # rgb(176, 244, 186, max = 255),  
 # rgb(178, 246, 181, max = 255),  
 # rgb(181, 246, 178, max = 255),  
 # rgb(186, 247, 178, max = 255),  
 # rgb(192, 247, 178, max = 255),  
 # rgb(198, 248, 178, max = 255),  
 # rgb(204, 249, 178, max = 255),  
 # rgb(210, 250, 177, max = 255),  
 # rgb(217, 250, 178, max = 255),  
 # rgb(224, 251, 178, max = 255),  
 # rgb(231, 252, 178, max = 255),  
 # rgb(238, 252, 179, max = 255),  
 # rgb(245, 252, 179, max = 255),  
 # rgb(250, 252, 178, max = 255),  
 # rgb(248, 249, 172, max = 255),  
 # rgb(238, 244, 162, max = 255),  
 rgb(226, 240, 151, max = 255),  
 rgb(213, 235, 140, max = 255),  
 rgb(198, 228, 128, max = 255),  
 rgb(184, 222, 118, max = 255),  
 rgb(170, 216, 108, max = 255),  
 rgb(154, 211, 98, max = 255),  
 rgb(140, 205, 89, max = 255),  
 rgb(125, 199, 82, max = 255),  
 rgb(110, 194, 74, max = 255),  
 rgb( 94, 188, 66, max = 255),  
 rgb( 77, 182, 57, max = 255),  
 rgb( 62, 176, 50, max = 255),  
 rgb( 49, 171, 44, max = 255),  
 rgb( 39, 165, 42, max = 255),  
 rgb( 30, 160, 43, max = 255),  
 rgb( 24, 154, 46, max = 255),  
 rgb( 18, 148, 49, max = 255),  
 rgb( 14, 142, 52, max = 255),  
 rgb( 9, 137, 56, max = 255),  
 rgb( 7, 132, 60, max = 255),  
 rgb( 12, 130, 63, max = 255),  
 rgb( 24, 130, 63, max = 255),  
 rgb( 40, 132, 61, max = 255),  
 rgb( 52, 136, 60, max = 255),  
 rgb( 64, 140, 59, max = 255),  
 rgb( 76, 142, 59, max = 255),  
 rgb( 87, 146, 56, max = 255),  
 rgb( 99, 148, 54, max = 255),  
 rgb(110, 150, 52, max = 255),  
 rgb(120, 154, 50, max = 255),  
 rgb(128, 156, 48, max = 255),  
 rgb(137, 160, 46, max = 255),  
 rgb(147, 162, 43, max = 255),  
 rgb(156, 164, 41, max = 255),  
 rgb(166, 166, 39, max = 255),  
 rgb(176, 170, 36, max = 255),  
 rgb(187, 173, 34, max = 255),  
 rgb(197, 176, 30, max = 255),  
 rgb(207, 177, 28, max = 255),  
 rgb(218, 179, 24, max = 255),  
 rgb(228, 180, 20, max = 255),  
 rgb(238, 182, 14, max = 255),  
 rgb(246, 182, 8, max = 255),  
 rgb(248, 176, 4, max = 255),  
 rgb(244, 166, 2, max = 255),  
 rgb(238, 155, 2, max = 255),  
 rgb(232, 144, 2, max = 255),  
 rgb(226, 132, 2, max = 255),  
 rgb(220, 122, 2, max = 255),  
 rgb(216, 111, 2, max = 255),  
 rgb(211, 102, 2, max = 255),  
 rgb(206, 92, 2, max = 255),  
 rgb(200, 84, 2, max = 255),  
 rgb(192, 74, 2, max = 255),  
 rgb(186, 66, 2, max = 255),  
 rgb(180, 58, 2, max = 255),  
 rgb(174, 49, 2, max = 255),  
 rgb(169, 42, 2, max = 255),  
 rgb(163, 36, 2, max = 255),  
 rgb(157, 30, 2, max = 255),  
 rgb(151, 23, 2, max = 255),  
 rgb(146, 18, 1, max = 255),  
 rgb(141, 14, 1, max = 255),  
 rgb(135, 8, 0, max = 255),  
 rgb(130, 5, 0, max = 255),  
 rgb(125, 4, 0, max = 255),  
 rgb(122, 8, 2, max = 255),  
 rgb(119, 13, 2, max = 255),  
 rgb(118, 16, 2, max = 255),  
 rgb(117, 18, 4, max = 255),  
 rgb(117, 20, 4, max = 255),  
 rgb(117, 21, 4, max = 255),  
 rgb(116, 22, 4, max = 255),  
 rgb(116, 24, 5, max = 255),  
 rgb(114, 26, 6, max = 255),  
 rgb(114, 29, 6, max = 255),  
 rgb(112, 31, 7, max = 255),  
 rgb(111, 33, 8, max = 255),  
 rgb(110, 35, 8, max = 255),  
 rgb(110, 36, 8, max = 255),  
 rgb(109, 38, 9, max = 255),  
 rgb(108, 40, 10, max = 255),  
 rgb(108, 40, 10, max = 255),  
 rgb(108, 42, 10, max = 255),  
 rgb(107, 44, 11, max = 255),  
 rgb(106, 44, 12, max = 255),  
 rgb(106, 46, 12, max = 255),  
 rgb(107, 48, 14, max = 255),  
 rgb(110, 52, 18, max = 255),  
 rgb(113, 57, 23, max = 255),  
 rgb(116, 62, 28, max = 255),  
 rgb(118, 66, 32, max = 255),  
 rgb(121, 70, 37, max = 255),  
 rgb(125, 74, 43, max = 255),  
 rgb(128, 79, 50, max = 255),  
 rgb(131, 85, 56, max = 255),  
 rgb(135, 90, 63, max = 255),  
 rgb(138, 96, 69, max = 255),  
 rgb(140, 101, 76, max = 255),  
 rgb(144, 106, 84, max = 255),  
 rgb(147, 111, 90, max = 255),  
 rgb(150, 116, 96, max = 255),  
 rgb(152, 122, 104, max = 255),  
 rgb(156, 129, 112, max = 255),  
 rgb(158, 135, 120, max = 255),  
 rgb(160, 141, 130, max = 255),  
 rgb(163, 147, 139, max = 255),  
 rgb(166, 154, 147, max = 255),  
 rgb(167, 160, 156, max = 255),  
 rgb(170, 167, 164, max = 255),  
 rgb(172, 172, 171, max = 255),  
 rgb(174, 174, 174, max = 255),  
 rgb(178, 178, 178, max = 255),  
 rgb(181, 181, 181, max = 255),  
 rgb(184, 184, 184, max = 255),  
 rgb(188, 188, 188, max = 255),  
 rgb(192, 192, 192, max = 255),  
 rgb(196, 196, 196, max = 255),  
 rgb(200, 200, 200, max = 255),  
 rgb(204, 204, 204, max = 255),  
 rgb(208, 206, 208, max = 255),  
 rgb(212, 210, 212, max = 255),  
 rgb(216, 214, 216, max = 255),  
 rgb(218, 216, 218, max = 255),  
 rgb(221, 219, 221, max = 255),  
 rgb(225, 223, 225, max = 255),  
 rgb(229, 227, 229, max = 255),  
 rgb(233, 231, 233, max = 255),  
 rgb(235, 233, 235, max = 255)))  
  
aus.dem <- ggR(hill.aus3) + ggR(alt.aus, geom\_raster = T, ggLayer = T, alpha = 0.75, # Creating digital elevation map of Australia with elevation colour coded to gradient  
 ggObj = T) +  
 scale\_fill\_gradientn(colours = schwarzwald(n = 1500), name = "Elevation", # Schwarzwald colour palette function used  
 na.value = "transparent", limits = c(0, 1200),   
 expand = c(0, 0),   
 guide = guide\_colourbar(nbin = 2000, barwidth = 0.5,   
 barheight = 3,   
 title.theme = element\_text(size = 6),   
 label.theme = element\_text(size = 6.5),   
 draw.ulim = F)) +   
 coord\_quickmap() + theme\_map() +  
 theme(legend.position = c(0.83, 0.4), legend.background = element\_rect(fill = "transparent"))

## Coordinate system already present. Adding new coordinate system, which will replace the existing one.

aus.dem # Viewing map



#### State border map and DEM ####  
map1 <- aus.dem  
grid.arrange(map2, map1, nrow = 1) # Arranging maps side by side

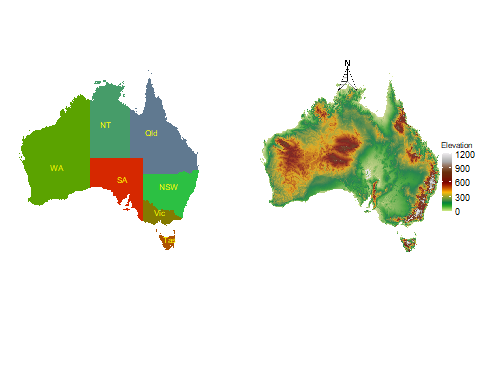


## Add North arrow  
library(ggsn)

##   
## Attaching package: 'ggsn'

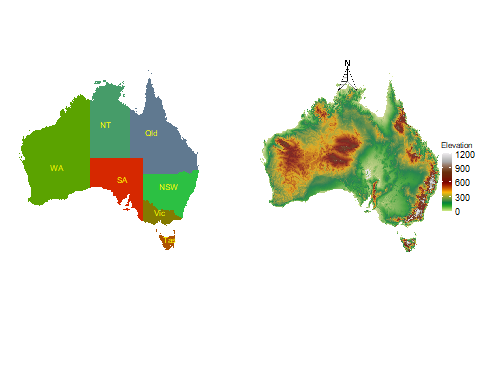
## The following object is masked from 'package:raster':  
##   
## scalebar

north2(ggp = grid.arrange(map2, map1, nrow = 1), x = 0.72, y = 0.8, symbol = 3)



## TableGrob (1 x 2) "arrange": 2 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (1-1,2-2) arrange gtable[layout]

#pdf("maps.pdf", width = 5, height = 5) # Exporting to pdf  
north2(ggp = grid.arrange(map2, map1, nrow = 1), x = 0.72, y = 0.8, symbol = 3)



## TableGrob (1 x 2) "arrange": 2 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (1-1,2-2) arrange gtable[layout]

#dev.off() # Finish exporting