Epi Lab color code

Software/Packages/Add-ins required

Software/Packages/Add-ins recommended

Description text

R code to copy/paste into console

R code to copy/paste into console that needs adjustment to your personal workspace

Websites where you can download requirements

Lab #11 requirements

- R - http://cran.r-project.org/bin/windows/base/
- R Studio - www.rstudio.com/ide/download/desktop
- Internet connection
- R packages "lattice" and "Cairo"

Lattice graphs - different data JDG

This is the second of the lattice graphing exercises. Here I go back to what started me on lattice graphs as mentioned last month. The data here is of multiple types - in this case we have a baseline mortality rates in ostrich chicks over a period of 26 weeks with parallel weather data of ambient temperature (maximum and minimum) and then rainfall. Why we did this initially was to identify whether weather subjectively influenced mortality. In this case there was a period of weather change and higher than normal mortality in week 13, and you'll see in the exercise that we show how to label these events.

This lab also introduces something that I should have learnt ages ago - but rather late than never. When plotting in R the plot in the window is manipulated by the size of the window. We now show how to directly export plots in pdf so that it'll look the same on whatever computer you are using

Functions and Code covered - Lab 11

xyplot in the lattice package; **CairoPDF** function in the **CairoPDF** package to generate pdf plots in working directory, a couple of new **lattice** options in the **lattice** framework including functions to manipulate individual panels; **tapply** to summarise data by group; **dir.create** to make directories on local machine

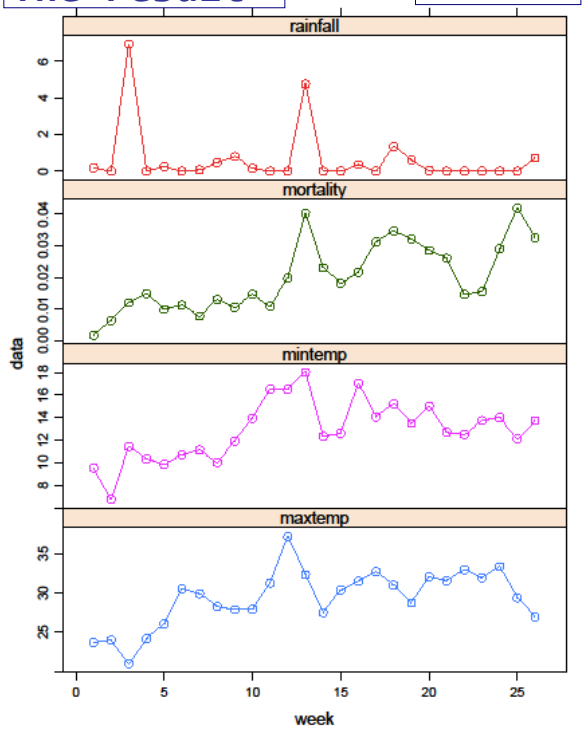
The code

```
#Install (if necessary) and load the libraries required
install.packages("Cairo")
library(Cairo)

install.packages("lattice")
library(lattice)

#Create a directory so that the plots you make are in one place
dir.create("C:/latticeexample")
dir.create("C:/latticeexample/graphs")

#Set the working directory to the newly created directory so the exports go directly there
setwd("C:/latticeexample/graphs")

#import the sample dataset and have a quick look at the data
aih7data <- read.csv("http://www.jdata.co.za/backpagelabs/backpagelabs_jdg_latticeai.csv")

head(aih7data,10)
tapply(X=aih7data$data,INDEX=aih7data$group,
FUN = summary)
#the "FUN" refers to the function you want to run on the grouped data. There are 4 groups of data - rainfall, max temp, min temp and mortality over the 26 weeks of data collected
#Again we'll get to the final lattice graph after going through a few steps each step will save the plot to your c: directory we have created so maybe open that so long to view your data - it will not show up in your plotting window this time

##Plot 1: BASIC PLOT#####################################
CairoPDF("plot1.pdf",6,8)
#This function makes a pdf 6 units wide and 8 high and after some experimentation this worked for this graph
xyplot(data~week|group, data = aih7data,
  groups = group, horizontal = FALSE, layout = c(1,4),
  scales=list(y=list(relation="free")), type = "o"
      )
dev.off()
#each CairoPDF plot needs to be turned off with a dev.off()
#the type = "o" refers to the type of xy graph you want - in our case a line graph does the trick

##PLOT 2 - RE_ORDER PANELS AND CHANGE Y AXIS LIMITS#############
CairoPDF("plot2.pdf",6,8)
xyplot(data~week|group, data = aih7data,
  groups = group, horizontal = FALSE, layout = c(1,4),
  scales=list(y=list(relation="free")), type = "o",
  index.cond = list(c(3,2,1,4)),
  ylim=list(c(20,40),c(0,20),c(0,0.05),c(0,10))
        )
dev.off()

# here the "index.cond" function forces the 3rd panel 1st (I think mortality should be on the bottom), the 2nd panel (min temp) stays 2nd, the 1st panel moves to 3rd (max temp) and the 4th stays where it is (rainfall).

#everything referring to the panels must be kept in this order, so the y axis limits per axis are first for the max temp panel, then the min temp, then mortality, then rainfall

##PLOT 3 - MANIPULATE THE PANEL STRIP STYLE AND ADD AXIS LABELS
CairoPDF("plot3.pdf",6,8)
xyplot(data~week|group, data = aih7data,
  groups = group, horizontal = FALSE, layout = c(1,4),
  scales=list(y=list(relation="free")), type = "o",
  index.cond = list(c(3,2,1,4)),
  ylim=list(c(20,40),c(0,20),c(0,0.05),c(0,10)),
  strip=strip.custom(style= 1,bg="gray",
      par.strip.text=list(col="black",cex=0.8),
      factor.levels=c("Maximum Temperature",
        "Minimum Temperature",
        "Mortality Rate",
        "Rainfall")),
  ylab=list(c("proportion",
      expression(paste("Temperature (",degree,"C)")),
      expression(paste("Temperature (",degree,"C)")),
      "mm per week"), cex=0.8),
  xlab = "Week of evaluation between 1 Oct 2012 and 25 March 2013"
  )
```

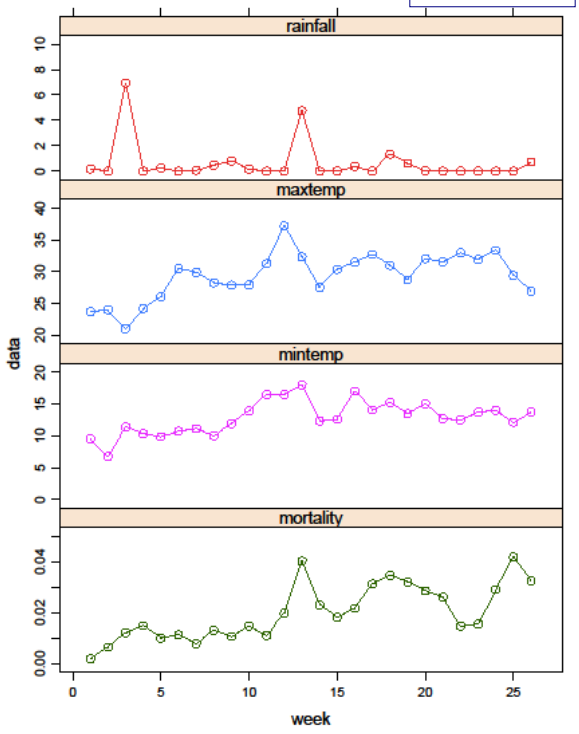
```
dev.off()

#note the colour of the strip is now grey with black writing
#remember cex  applies generally to size - so in this case text size
#remember that the y labels must still be put in order of the original plot, so max temp, then min temp, then mortality, then rainfall
#note the function to include a ° sign in R

#Plot 3 already gets us quite a long way, and you could probably stop right there however I think it would be nice to add some data point labels.
#so lets do 2 things here - 1. add the maximum data labels per group except for the min temp where the minimum would be valuable
#2. add the labels for week 13 where subjectively it looks like a lot happened in both weather and mortality

##PLOT 4 - ADD MAX AND MIN VALUES TO PLOT################
CairoPDF("plot4.pdf",6,8)
xyplot(data~week|group, data = aih7data,
  groups = group, horizontal = FALSE, layout = c(1,4),
  scales=list(y=list(relation="free")), type = "o",
  index.cond = list(c(3,2,1,4)),
  ylim=list(c(20,40),c(0,20),c(0,0.05),c(0,10)),
  strip=strip.custom(style= 1,bg="gray",
      par.strip.text=list(col="black",cex=0.8),
      factor.levels=c("Maximum Temperature",
         "Minimum Temperature",
         "Mortality Rate",
         "Rainfall")),
  ylab=list(c("proportion",
      expression(paste("Temperature (",degree,"C)")),
      expression(paste("Temperature (",degree,"C)")),
      "mm per week"), cex=0.8),
  xlab = "Week of evaluation between 1 Oct 2012 and 25 March 2013",

  panel=panel.superpose,
  panel.groups=function(x,y,...){
      panel.xyplot(x,y,t="o",...)
#here we generate variables which have relevance to the point we want to plot. The first is to identify the highest Y values and identify the x value for that y value - and all per group because we are in a panel.groups function
#so here xt = the x value (i.e. week) of the y value that's = the max y value
  xt <- x[y==max(y)]
#so here yt = the maximum y value per group
  yt <- y[y==max(y)]
#now similarly the minimum values for the min temp panel
#xm = the x value (i.e. week) of the y value that's = to the min y value
  xm <- x[y==min(y)]
#so here ym = minimum y value
  ym <- y[y==min(y)]
#now to plot Maximum values for panels 1,3 and 4
#In R the "|" sign indicates "or"
#so here we say for panels 1,3 or 4 plot the yt labels at the point on the panel of location xt (weeks) ,yt (max value)
  if(packet.number()==1|
  packet.number()==3|
  packet.number()==4){
    panel.text(xt, yt,
    labels=yt,
    pos=4, #shows labels on right side
    cex=0.8)
                        }#close maximum if
#now to plot Minimum values for panel 2
#Here as above just for panel 2 we want the minimum y plotted at the point xm,ym
  if(packet.number()==2){
    panel.text(xm, ym, labels = ym,
    pos=4,# show labels on right side
    cex=0.8)
                        } #close minimum if
                                } #close panel.groups
                                  #function
)#close xyplot
dev.off()

##PLOT 5: ADD A SERIES OF LABELS FOR WEEK 13 SPECIFICALLY#########

CairoPDF("plot5.pdf",6,8)
xyplot(data~week|group, data = aih7data,
  groups = group, horizontal = FALSE, layout = c(1,4),
  scales=list(y=list(relation="free")), type = "o",
  index.cond = list(c(3,2,1,4)),
  ylim=list(c(20,40),c(0,20),c(0,0.05),c(0,10)),
  strip=strip.custom(style= 1,bg="gray",
      par.strip.text=list(col="black",cex=0.8),
      factor.levels=c("Maximum Temperature",
         "Minimum Temperature",
         "Mortality Rate",
         "Rainfall")),
  ylab=list(c("proportion",
      expression(paste("Temperature (",degree,"C)")),
      expression(paste("Temperature (",degree,"C)")),
      "mm per week"), cex=0.8),
  xlab = "Week of evaluation between 1 Oct 2012 and 25 March 2013",

  panel=panel.superpose,
  panel.groups=function(x,y,...){
      panel.xyplot(x,y,t="o",...)
  xt <- x[y==max(y)]
  yt <- y[y==max(y)]
  xm <- x[y==min(y)]
  ym <- y[y==min(y)]

  if(packet.number()==1|
    packet.number()==3|
    packet.number()==4){
    panel.text(xt, yt, labels=yt, pos=4, cex=0.8)}

  if(packet.number()==2){
    panel.text(xm, ym, labels = ym, pos=4, cex=0.8)}
#Now we get the 13th week data and plot it on all panels
  x13 <- x[13] # find 13th x point
  y13 <- y[13] # find value at of y at the 13th point
#put this panel text in every plot at x13,y13
  panel.text(x13, y13, labels=y[13], pos=4, cex=0.8)
                                    }
)
dev.off()
```

The result

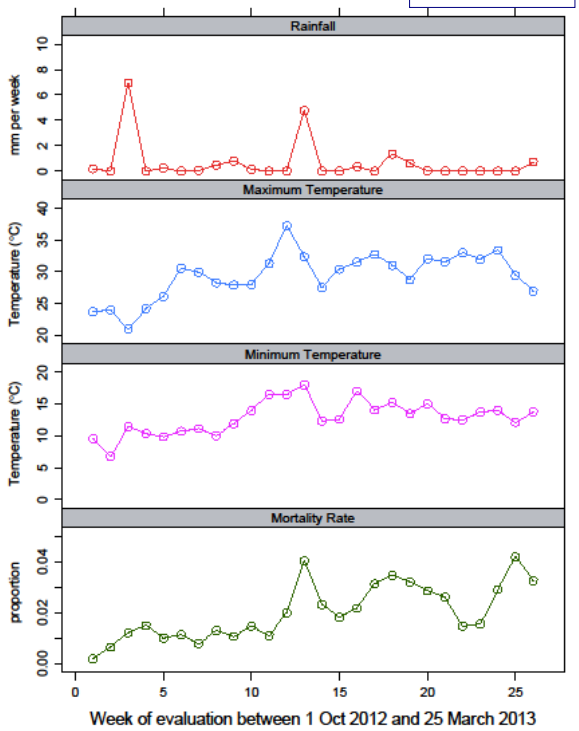
Plot 1



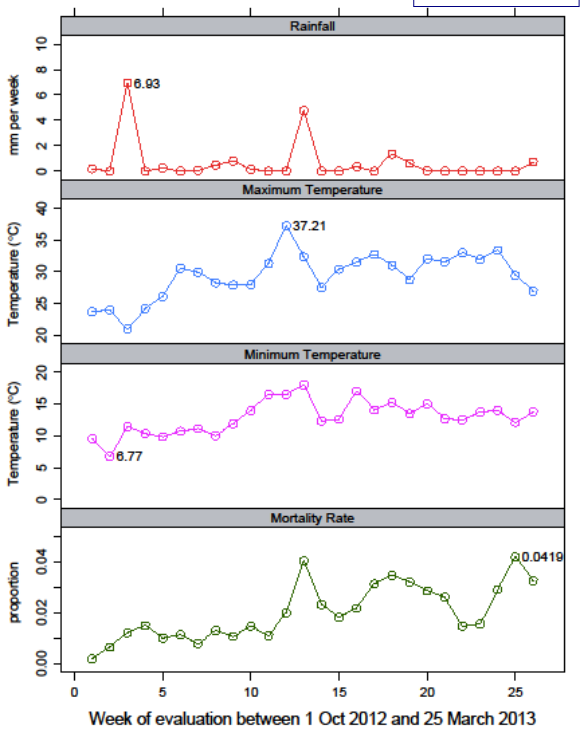
Plot 2



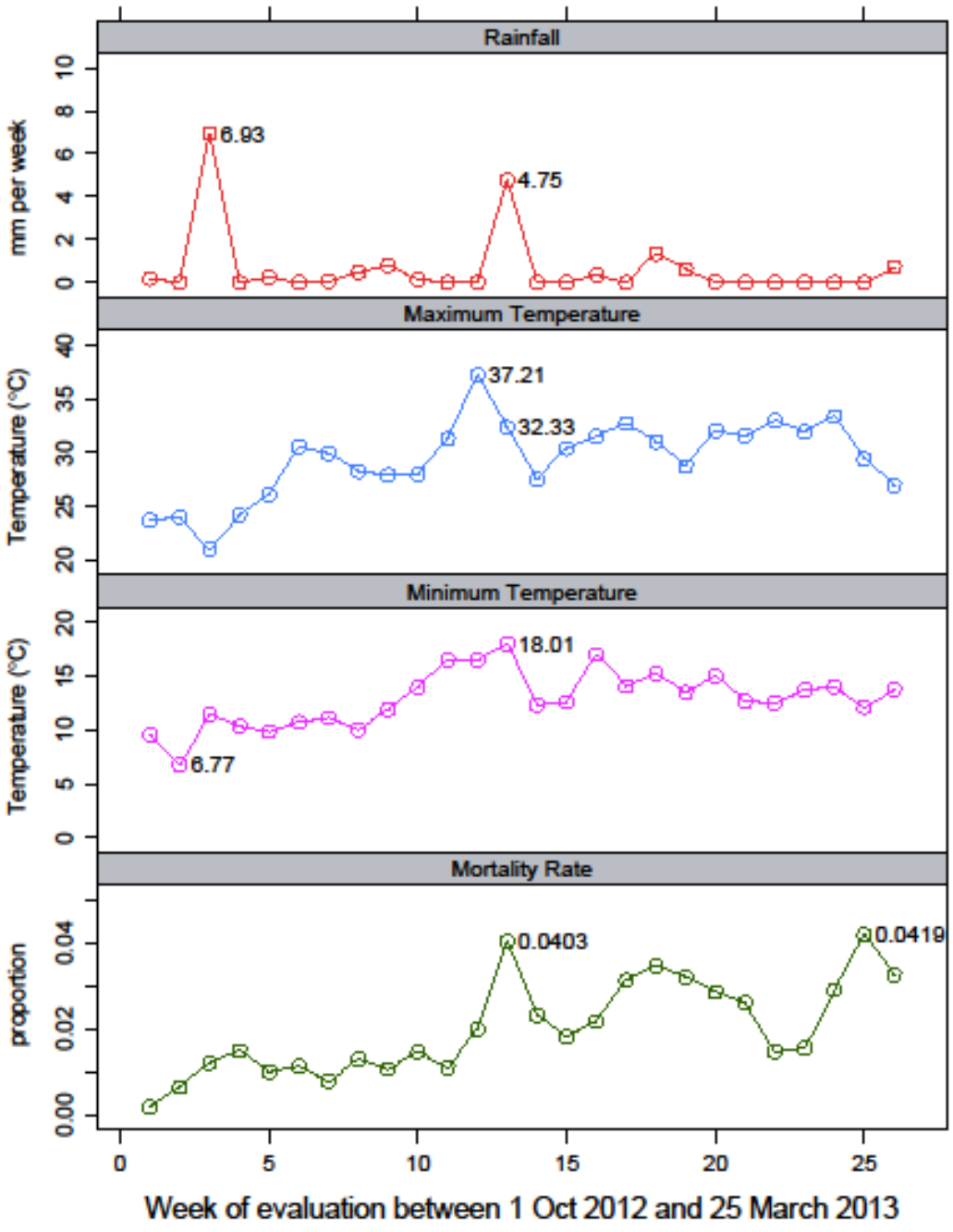
Plot 3



Plot 4



Plot 5



References

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Sarkar, Deepayan (2008) Lattice: Multivariate Data Visualization with R. Springer, New York. ISBN 978-0-387-75968-5

Simon Urbanek and Jeffrey Horner (2014). Cairo: R graphics device using cairo graphics library for creating high-quality bitmap (PNG, JPEG, TIFF), vector (PDF, SVG, PostScript) and display (X11 and Win32) output.. R package version 1.5-6. http://CRAN.R-project.org/package=Cairo