

Assignment-3

Syed Muhammad Adeel Ibrahim

May 19, 2019

Question 1 [8 marks] Consider Fig. 1, which makes one of two parallel lines of equal length look shorter. Use R to reproduce the Figure as closely as possible.

```
x0 <- c(1, 3)
y0 <- c(1, 1)

x1 <- c(1, 3)
y1 <- c(3, 3)

plot(NULL, xlim = c(0, 4), ylim= c(0, 4), type = "l", axes=FALSE,
frame.plot=FALSE, ann=FALSE)
rect(0, 0, 4, 4, col = rgb(red = 227/255, green = 100/255, blue = 65/255),
border = NA)

arrows(x1[2], y1[2], x1=x1[1], y1=y1[1], lwd=8, length = .8, col='white',
angle = 135, code = 3, lend = 2, xpd = TRUE)
arrows(x0[2], y0[2], x1=x0[1], y1=y0[1], lwd=8, length = .8, col='white',
angle = 45, code = 3, lend = 2, xpd = TRUE)
```

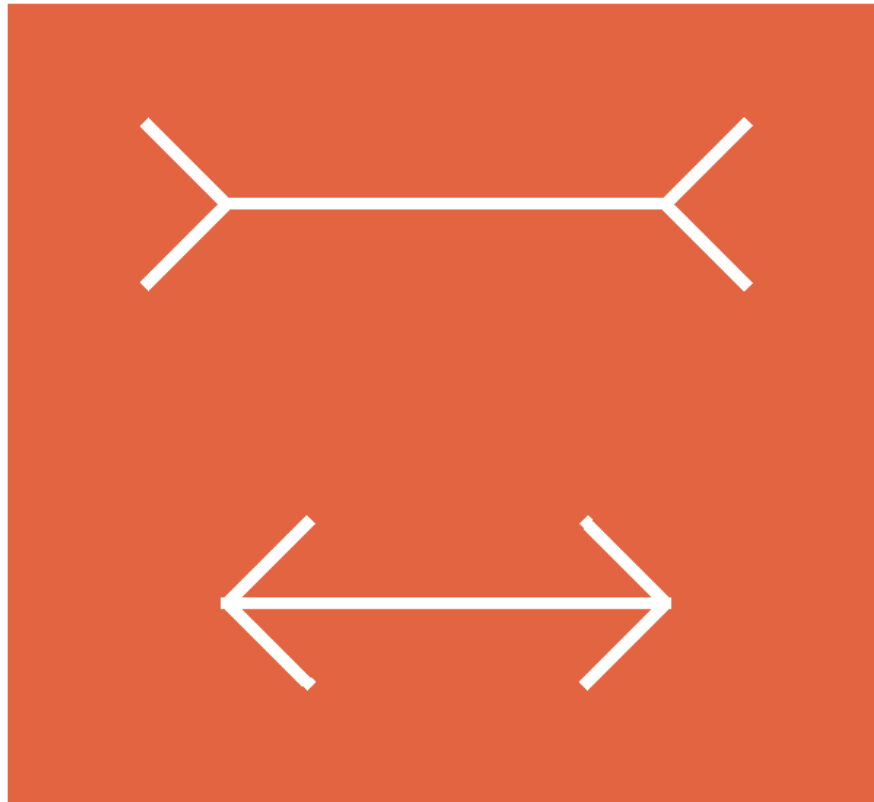


Figure 1: A figure that deceptively makes two parallel lines of equal length look of unequal length.

Question 2 2. [8 marks] Consider Fig. 2.

(a) Use R to reproduce the figure as closely as possible. [6 marks]

```
x <- seq(-4, 4, length = 1000)

xplot <- x[x < -0.862]
yplot <- dnorm(xplot, mean = 0, sd = 1)

plot(xplot, yplot, type = "l", col = "green4", lwd = 3, axes = FALSE,
frame.plot = FALSE, ann = FALSE, ylim = c(-.1, .5), xlim = c(-4, 4))

xline <- x[x >= -0.862]
yline <- dnorm(xline, mean = 0, sd = 1)
```

```
lines(x = xline, y = yline, col = "blue", lwd = 3)
```

```
meanOmega <- dnorm(-0.862, mean = 0, sd = 1)
```

```
abline(h = 0, col = "blue", lwd = 3)
```

```
lines(x = c(-.862, -.862), y = c(-.02, meanOmega), col = "purple", lty="6666",  
lwd = 3)
```

```
points(x = -.862, y = -.02, pch = 17, col = "purple", cex = 4)
```

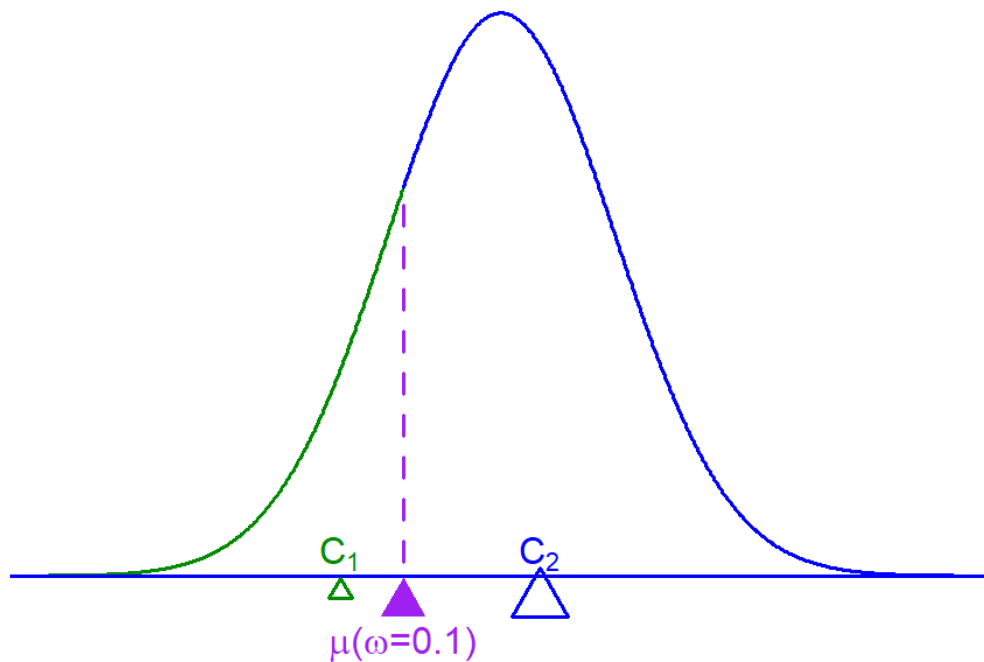
```
text(x = -.862, y = -.03, labels = expression(paste(mu, "(", omega, "=",  
"0.1)")), col = "purple", pos = 1, cex=2)
```

```
points(x = -1.415, y = -.011, pch = 2, col = "green4", cex = 2, lwd = 3)
```

```
text(x = -1.415, y = -.004, labels = expression('C'[1]), col = "green4", pos  
= 3, cex=2)
```

```
points(x = 0.342, y = -.018, pch = 2, col = "blue", cex = 5, lwd = 3)
```

```
text(x = 0.342, y = -.004, labels = expression('C'[2]), col = "blue", pos =  
3, cex=2)
```



(b)

Modify the plot by choosing nicer colours, etc. in order to improve it. This should be another plot. [2 marks]

```
# sequence for generating Bell graph
x <- seq(-4, 4, length = 1000)

# line plotting of elements below -0.0862
xplot <- x[x < -0.862]
yplot <- dnorm(xplot, mean = 0, sd = 1)
plot(xplot, yplot, type = "l", col=rgb(0, 1, 0, .3), lwd = 3, axes = FALSE,
frame.plot = FALSE, ann = FALSE, ylim = c(-.1, .5), xlim = c(-4, 4))

# Polygon filling of area below -0.862
m <- length(xplot)
x.poly <- c(xplot, xplot[m], xplot[1])
y.poly <- c(yplot, 0, 0)

# Adjoin two x-coordinates
# .. and the corresponding y-
coordinates
```

```

polygon(x.poly, y.poly, col=rgb(0, 1, 0, .3), border=NA)

# Line plotting of elements above or equal to -0.862
xline <- x[x >= -0.862]
yline <- dnorm(xline, mean = 0, sd = 1)
lines(x = xline, y = yline, col = rgb(0, 0, 1, .3), lwd = 3)

# Polygon filling of elements above or equal to -0.862
m <- length(xline)
x.poly <- c(xline, xline[m], xline[1])      # Adjoin two x-coordinates
y.poly <- c(yline, 0, 0)                    # .. and the corresponding y-
coordinates
polygon(x.poly, y.poly, col=rgb(0, 0, 1, .3), border=NA)

# Partitioning line
meanOmega <- dnorm(-0.862, mean = 0, sd = 1)
lines(x = c(-.862, -.862), y = c(-.02, meanOmega), col = "purple", lty="6666",
lwd = 3)

#plotting and text
points(x = -.862, y = -.02, pch = 17, col = "purple", cex = 4)
text(x = -.862, y = -.03, labels = expression(paste(mu, "(", omega, "=",
"0.1)")), col = "purple", pos = 1, cex=2)
text(x = -1.415, y = -.004, labels = expression('C'[1]), col = "white", pos
= 3, cex=4)
text(x = 0.342, y = -.004, labels = expression('C'[2]), col = "white", pos =
3, cex=4)

```

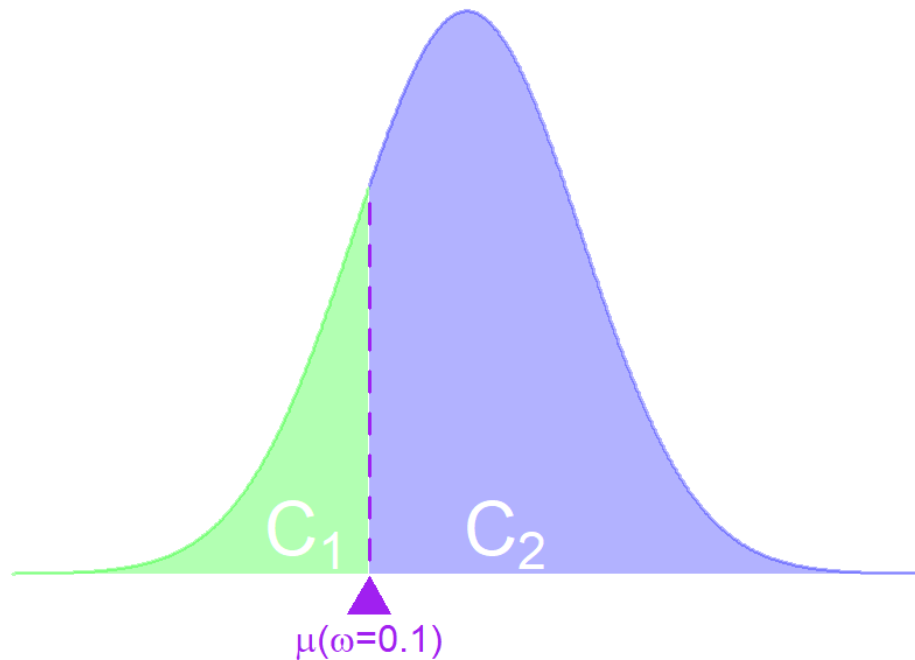


Figure 2: Illustration of the interpretation of expectiles in terms of centres of balance: the hollow triangles at positions c_1 and c_2 . Here, the vertical dashed line is at the 0.1-expectile, and the solid triangle at $\mu(\omega=0.1)$.

Question 3 [10 marks] Consider Fig. 3. Use R to reproduce the figure as closely as possible. The outer perimeter is a circle.

```
plot.new()
plot.window(xlim = c(-2, 2), ylim = c(-2, 2), asp = 1)

theta0 <- seq(0, 2 * pi, length = 150)

x <- 2 * cos(theta0)
y <- 2 * sin(theta0)

polygon(x, y, col = "darkseagreen", border = "black", lwd = 2)
```

```

theta <- seq(0, 6*pi, length = 150)

# four curves in circle
spiralCurve <- 1/(3*pi)

r1 <- spiralCurve * theta
r2 <- (spiralCurve + .021) * theta

x1 <- r1 * cos(theta)
y1 <- r1 * sin(theta)

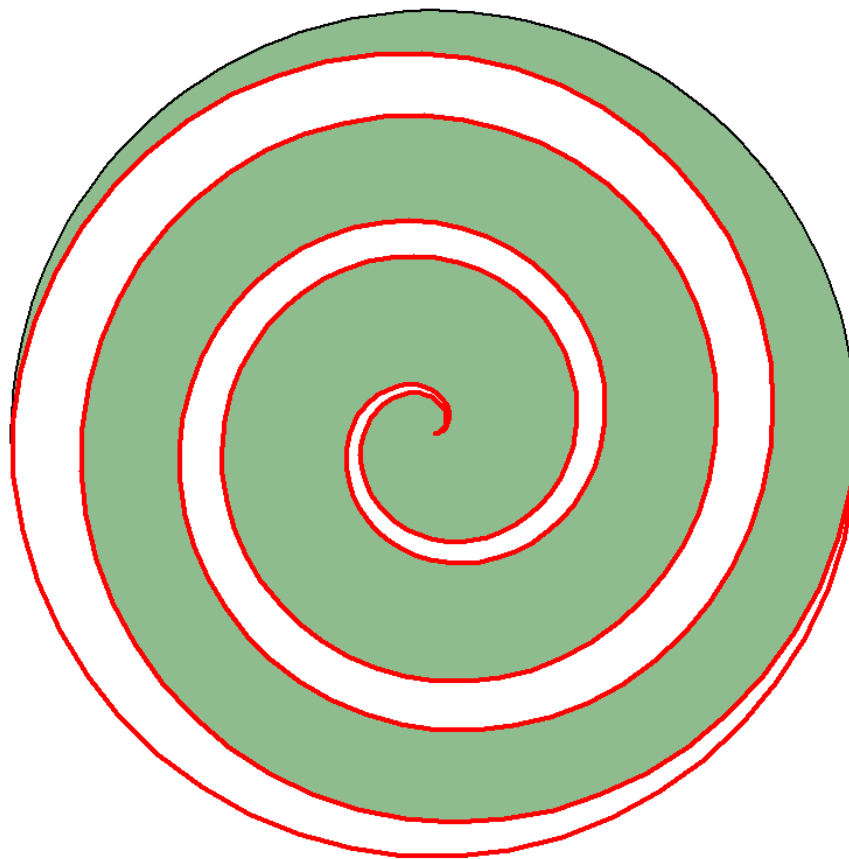
x2 <- r2 * cos(theta)
y2 <- r2 * sin(theta)

# verification of distance that should not exceed 2 units
for (i in 1:150) {
  if(sqrt((x2[i]^2) + (y2[i]^2)) > 2) {
    x2[i] <- 2 * cos(theta[i])
    y2[i] <- 2 * sin(theta[i])
  }
}

lines(x1, y1, col = "red", type = 'l')
lines(x2, y2, col = "red", type = 'l')

polygon(c(x1, rev(x2)), c(y1, rev(y2)), col = "white", border = "red", lwd =
4)

```



##Question 4. [24 marks] Consider Fig. 4, which displays a BBC News graphic obtained from an article on recent Australian heatwaves. ###(a) The data from which the plot was made was unavailable from the BBC website. Instead, I downloaded something very similar from <http://www.bom.gov.au/> and it is available to you. The data happens to come from Sydney (pronounced Seedney). Hence your plot won't be exactly the same data-wise, nevertheless you can reproduce it otherwise the same. Read the data into R. Then check the annual means column by computing the means over all months-you are checking the integrity of the data. Comment. [5 marks]

```
bbcData <- read.csv('IDCJAC0002-066062-Data12edited.csv')

bbcData$ComputedMean <- round(rowMeans(bbcData[,4:16], na.rm = TRUE), 2)

result <- any(bbcData$ComputedMean != bbcData$Annual)
if(is.na(result)) {
  'Some values are Not Available.'
} else if (result) {
```



```

    'Data has Integrity Issue'
  } else {
    'No problem with data.'
  }
}

## [1] "Data has Integrity Issue"

```

(b) Try to mimic the figure in R as closely as possible, except for the BBC logo at the bottom and that the data doesn't match exactly. [10 marks]

```

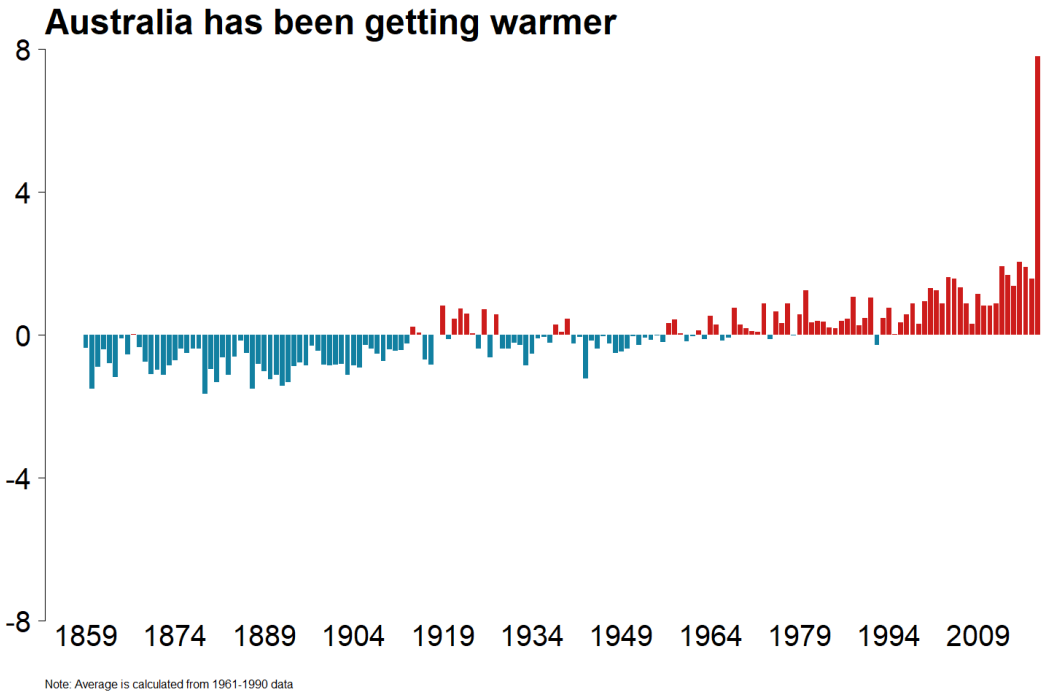
bbcData$meandiff <- round(bbcData$ComputedMean - mean(bbcData$ComputedMean),
2)

bbcData$color <- ifelse(bbcData$meandiff > 0, 1, 0)
bbcData$positive <- ifelse(bbcData$meandiff > 0, bbcData$meandiff, 0)
bbcData$negative <- ifelse(bbcData$meandiff < 0, bbcData$meandiff, 0)

mat <- matrix(bbcData$positive, nrow = 1, dimnames = list(c(), bbcData$Year))
mat <- rbind(mat, bbcData$negative)

barplot(
  height = mat,
  col = c('#cd1c1b', '#1280a1'),
  las = 1,
  border = NA,
  names.arg = colnames(mat),
  ylim = c(-8, 8),
  main = "Australia has been getting warmer",
  #mtext = "",
  sub = "Note: Average is calculated from 1961-1990 data",
  adj = 0,
  cex.main = 3,
  cex.names = 2.5,
  cex.axis = 2.5,
  yaxp = c(8, -8, 4),
  ann = FALSE
)

```



(c) Add in all years' data and obtain a similar plot, and use the same centring mean as in (b). Write a paragraph or two interpreting the plot, i.e., as a statistician, what do we learn from this plot? [5 marks]

```
bbcData$total <- round(rowSums(bbcData[,4:16], na.rm = TRUE), 2)

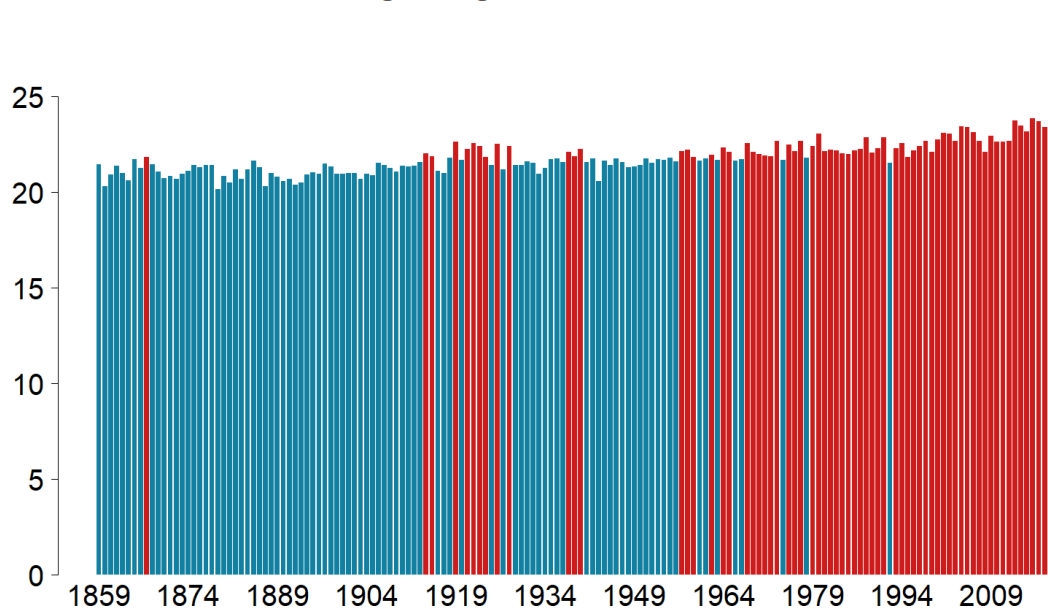
bbcData$positive <- ifelse(bbcData$ComputedMean > mean(bbcData$ComputedMean),
  bbcData$ComputedMean, 0)
bbcData$negative <- ifelse(bbcData$ComputedMean < mean(bbcData$ComputedMean),
  bbcData$ComputedMean, 0)

mat <- matrix(bbcData$positive, nrow = 1, dimnames = list(c(), bbcData$Year))
mat <- rbind(mat, bbcData$negative)

barplot(
  height = mat,
  col = c('#cd1c1b', '#1280a1'),
  las = 1,
  border = NA,
  names.arg = colnames(mat),
  #ylim = c(-8, 8),
  main = "Australia has been getting warmer",
  #mtext = "",
  sub = "Note: Average is calculated from 1961-1990 data",
  adj = 0,
  cex.main = 3,
  cex.names = 2.5,
```

```
cex.axis = 2.5,
#yaxp = c(8, -8, 4),
ann = FALSE
)
```

Australia has been getting warmer



Note: Average is calculated from 1961-1990 data

(d) Predict the annual mean temperature value in the year 2030, and try give a 95% prediction interval for this. Justify your method. [4 marks]

```
model <- lm(ComputedMean~Year, data = bbcData)
```

```
predict(model, data.frame(Year = 2030), interval="prediction")
```

```
##          fit      lwr      upr
## 1 23.16196 21.7637 24.56021
```

```
'Fitted Model is 23.16196, lower limit is 21.7637 and upper limit is 24.56021'
```

```
## [1] "Fitted Model is 23.16196, lower limit is 21.7637 and upper limit is 24.56021"
```

(e) Bonus. Suppose in Fig. 4 that temperatures stabilized beyond the year 2100 and the annual mean temperature above the average is exponentially distributed with unit mean independently each year. About how many years after 2100 would somebody have to wait for so that the probability of experiencing an annual maximum greater than 6.5???C above the average is at least 5%? [3 marks]