StatR 101: Fall 2012
Homework 3 Solutions
Eli Gurarie, October 16, 2012

1. **Medians:** A function that uses an `if()` statement and the `%%` operator:

```
Median <- function(x)
{
  x <- na.omit(x)
  n <- length(x)
  x.sorted <- sort(x)
  if(n%%2 == 1)
    x.median <- x.sorted[(n+1)/2]
  else
    x.median <- (x.sorted[n/2] + x.sorted[n/2 + 1])/2
  x.median
}
```

Here's efficient code (from a student) that uses `floor` and `ceiling`, that relies on no logical testing:

```
Median <- function(X)
{
        values = sort(X[!is.na(X)])       # remove the NA values and sort
        a = floor((length(values)+1)/2)   # find the lower value to sum
        b = ceiling((length(values)+1)/2) # find the upper value to sum
        m = (values[a] + values[b])/2     # find the midpoint of the values.
        return(m)
}
```

Testing that is works:

```
> x <- round(rnorm(10)*5^2)
> median(x)
[1] 11.5
> Median(x)
[1] 11.5
```

2. **Skewness**

(a)
```
Skew <-  function(x)
{
  x <- na.omit(x)
  m3 <- mean((x-mean(x))^3)
  m2 <- sum((x - mean(x))^2)/n
  return(m3/m2^(3/2))
}
```

(b) We expect GDP to be right skewed, Literacy to be left skewed, and Birthrate to be also right skewed, but less strongly than GDP. The results confirm these predictions:

```
> Skew(GDP)
[1] 1.870465
> Skew(Literacy)
[1] -1.417821
> Skew(Birthrate)
[1] 0.7485703
```

(c)
```
> data.frame(Literacy = tapply(Literacy, Continent, Skew),
+            GDP = tapply(GDP, Continent, Skew),
+            Birthrate = tapply(Birthrate, Continent, Skew))

                Literacy       GDP   Birthrate
Africa        -0.4611914 2.0451059 -0.62540872
Asia          -1.3431745 1.7775283  0.47786680
Europe        -3.1042619 1.2085067  1.27219719
North America -1.0228632 1.3941595  0.76809298
Oceania       -0.9904140 0.9260776  0.09151849
South America  0.1837546 0.1406809  0.90355420
```

Skewness indirectly indicates extent of inequality. South America has the most "balanced" GDP distribution, meaning that about at many countries are above as below the mean GDP, and the mean is therefore a reasonable summary statistic for GDP in South America. In contrast, the high skewness in Africa and Asia suggest that there are more wealthy outliers relative to their neighbors in those countries. The negative skewness in Africa's birthrate is interesting, as it suggests that those countries with the lowest birthrate are the outliers

(d) **Boxplots:** The most minimal code is below.

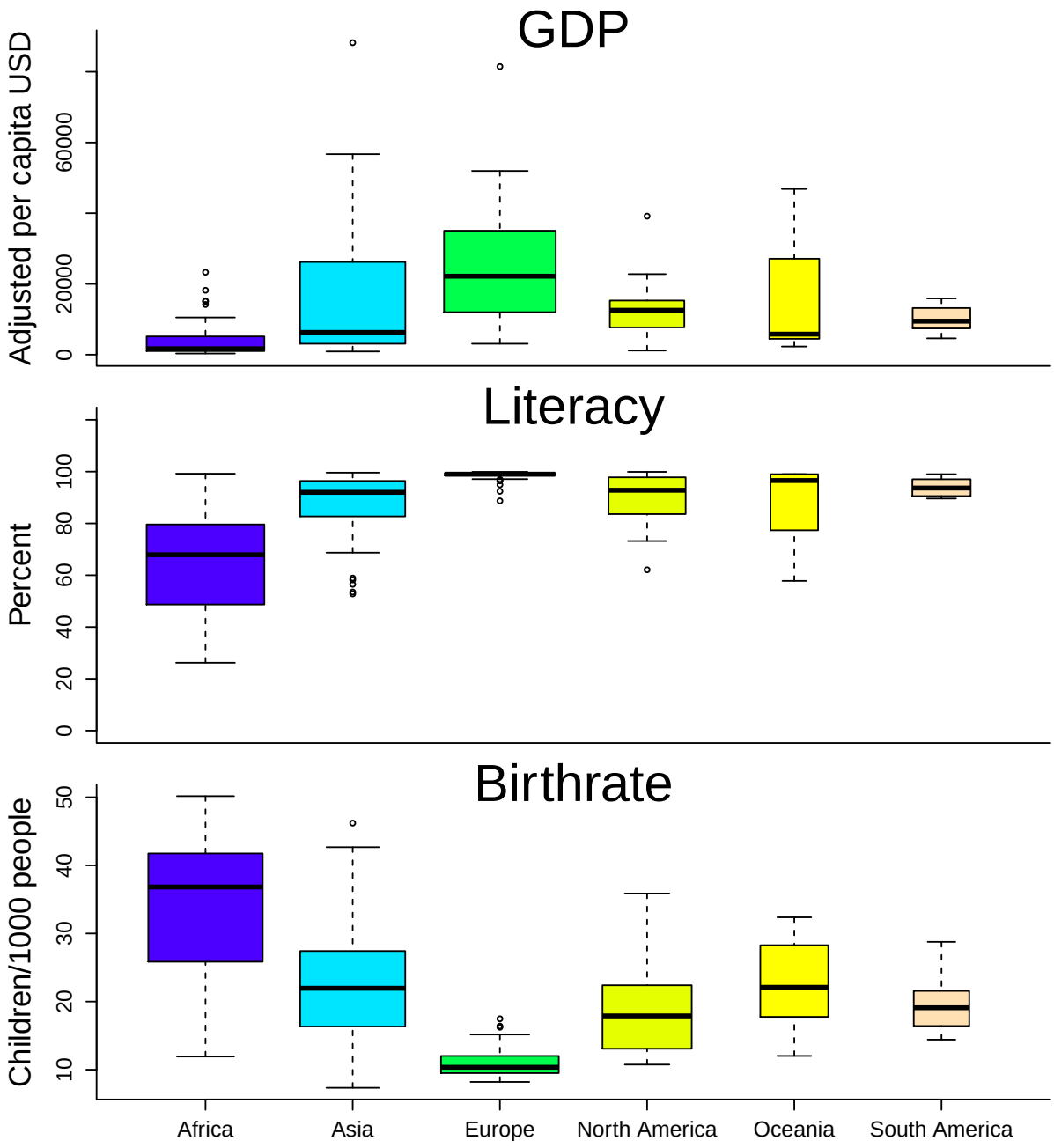
```
par(mfrow=c(3,1))
boxplot(GDP~Continent, main="GDP", varwidth=TRUE)
boxplot(Literacy~Continent, main="Literacy", varwidth=TRUE)
boxplot(Birthrate~Continent, main="Birthrate", varwidth=TRUE)
```

Note that even for a "minimal" boxplot, I like using the `varwidth=TRUE` option, because it is often helpful to be able to visualize easily which group has the most data points in it, though note that this is not an issue in a controlled, balanced experiment.

The complete code that I used for the figure on the following page is below. I used `mtext()` to produce titles that are below the typical title placement, and the adjustment of the margins so that the pictures are positioned symmetrically, but I only need to to label the bottom-most figure:

```
par(mfrow=c(3,1), bty="l", mar=c(0,5,4,2), cex.axis=1.25, cex.lab=1.5)
par(mfrow=c(3,1), bty="l", mar=c(0,5,4,2), cex.axis=1.25, cex.lab=1.75)
boxplot(GDP~Continent, data=C, xaxt="n", col=topo.colors(6),
        ylab="Adjusted per capita USD", varwidth=TRUE)
mtext(side=3, text="GDP", line=-1, cex=2)
par(mar=c(2,5,2,2))
boxplot(Literacy~Continent, data=C, xaxt="n", col=topo.colors(6),
        ylab="Percent", ylim=c(0,120), varwidth=TRUE)
mtext(side=3, text="Literacy", line=-1, cex=2)
par(mar=c(5,5,0,2))
boxplot(Birthrate~Continent, data=C, col=topo.colors(6),
        ylab="Children/1000 people", varwidth=TRUE)
mtext(side=3, text="Birthrate", line=-1, cex=2)
```

Skewness in the figure is most visible when the median line is close to the edge of a box. For example: the Oceania, Asia and Africa GDP's have among the highest positive skew, while literacy in Oceania is among the most negative skews.


GDP
Adjusted per capita USD
0
20000
60000
Literacy
Percent
0
20
40
60
80
100
Birthrate
Children/1000 people
10
20
30
40
50
Africa
Asia
Europe
North America
Oceania
South America

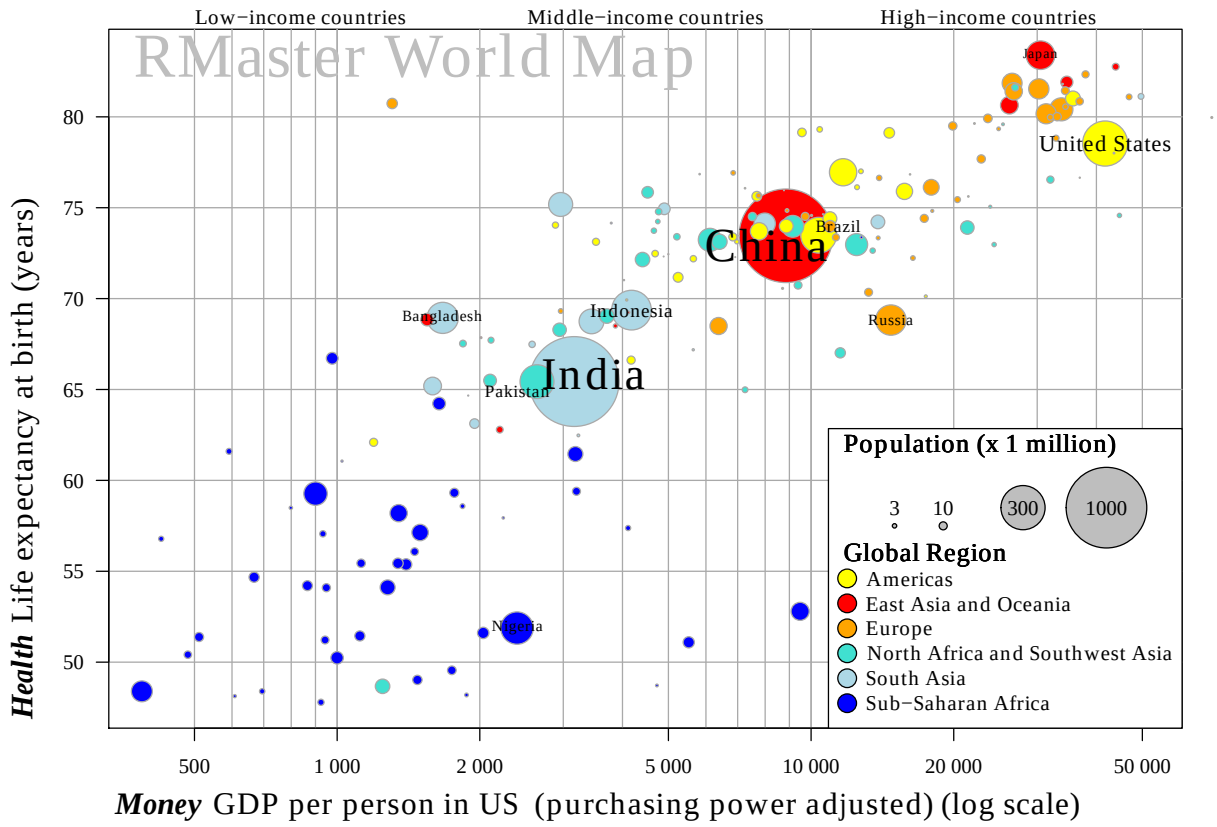
3. **Emulating Gapminder:** Solutions here vary quite a bit. I walk through generating a figure here, and provide some finalized code with the attached figure.

```
Data <- read.csv("GDPLifeExpectancyRegion2011.csv")
# Order your data by descreasing population size.
Data <- Data[order(Data$Population, decreasing=TRUE),]
  X <- Data$GDP
  Y <- Data$LifeExpectancy
  C <- Data$Country
  R <- Data$Region2
  P <- Data$Population
# specify colors corresponding to the regions.
# The list of regions is given by "> levels(R)":
# [1] "Americas"                          "East Asia and Oceania"
# [3] "Europe"                            "North Africa and Southwest Asia"
# [5] "South Asia"                        "Sub-Saharan Africa"
# We match them as follows:
  colors <- c("yellow", "red", "orange", "turquoise", "lightblue", "blue")
# Lets start with producing an "empty" plot of X and Y.
# It is important here to specify that the x-axis will be log-transformed.
# I also limited the range of the x-axis.
  plot(X, Y, log="x", type="n", xlab="", xaxt="n", ylab="", yaxt="n",
      xlim=c(400,50000))
# Our first task is gridlines:
  ygrid <- c(seq(500,1000,100), seq(1000,10000,1000), seq(10000,50000,10000))
  xgrid <- seq(45,85,5)
  abline(v = ygrid, col="darkgrey")
  abline(h = xgrid, col="darkgrey")
# Now we're just about ready for the real meat, which is plotting the data.
# This is a surprisingly short command. Note that we use "pch=21" for filled
# circles, and that I added the texttt{xpd=TRUE} which allows me to add points
# outside of the limits (for the outliers)
  points(X, Y, cex=sqrt(P/max(P))*10, bg=colors[R], pch=21,xpd=TRUE)
# We're 90% there, which just means we're 90% away from the end!
# We can add the x and y-axes  very simply with:
  axis(1)
  axis(2)
# Some people didn't like the fact that exponential notation (e.g. 5e4 for 50,000)
# was used on the x-axis.  Here, I "accidentally" solved that by having limited by
# x-range.  One way you could control it (and match the formatting of the
# GapMinder plot), is to fix the legend of the axis.  (Note, you have to rerun all
#  the code above without the last line to recreate the plot!) So:
  axis(1, at = c(5,10,20,50,100,200,500)*100,
          label = c("500", "1 000", "2 000", "5 000", "10 000", "20 000", "50 000"))
# the y-axis, with horizontal numbers, looks OK,
  axis(2, las=2)
# For marginal texts, we use the mtext() command
```

```
# Upper margins:  Note how I use "paste" to create a vector with all three words:
# ("High", "Low", and "Middle") spliced to "-income countries".
  mtext(paste(c("Low", "Middle", "High"), "-income countries", sep=""),
      side=3, line=0, adj=c(.1,.5,.9), cex=.75, font=6)
# the Lower and Left margins are tricky to totally control, because of their different
# fonts and sizes in a "single" string.  Here, some mucking around is necessary, and it
# ultimately depends on the size of the graphics device you are writing on. One nice
# compact solution (from a student) is to use the "expression()" function, which is used
# to typeset mathematical text.
  mtext(expression(paste(bold("Money"),italic("GDP per person in US (purchasing power adjusted)"))),
      side=1, line=2, adj=0.05)
mtext(expression(paste(bold("Health"),italic(" Life expectancy at birth (years)"))),
      side=2, line=2, adj=0.05)

# Legends can be the trickiest to get to look right, especially in this case.
# Note that you have to specify the "bg" for the points (not the legend box) with
# "pt.bg".  Similarly for the "cex" of the points, you use "pt.cex".
  legend("bottomright", pch=21, pt.bg=colors, legend=levels(R), bg="white",
         pt.cex=2, title="Region")
# To create the population size legend, we can space things with "NA"s
  sizes <- c(3, 10, NA, 300, NA, NA, 1000, NA, NA)
  pt.cex <- sqrt(c(3, 10, NA, 300, NA, NA, 1000, NA, NA)*1e6/max(P))*10
  legend("topleft", pt.cex=pt.cex, legend=sizes, pch=21,
          col="darkgrey", pt.bg="grey", bg="white",
          title="Population (x 1 million)")
# For another solution (e.g. if you want to put all the information in one box) is to
# create a long legend full of NA's, and add points and text point by point.  For an
# example, see the final code for the plot on the next page.

# Finally, text for countries.  You could go crazy and label ALL the countries, and
# scale the text to country size:
  text(X,Y, C, cex=sqrt(P/max(P)*3))
# But if you only wanted to do this with the ten largest countries, you know that they
# are the 1st ten in your data because the data are ordered:
  text(X[1:10], Y[1:10], C[1:10], cex=sqrt(P[1:10]/max(P[1:10])*3))
# There is some oveelap (betweem Pakistan and India, for example, and China and Brazil.
# We can "jitter" those, either systematically or randomly.
  dx <-  1.1^c(-1,1,0,0,1,-1,0,0,0,0)
  dy <- c(-1,1,0,0,1,-1,0,0,0,0)
  text(X[1:10]*dx, Y[1:10]+dy, C[1:10], cex=sqrt(P[1:10]/max(P[1:10])*3))
# That more or less does it.  Look at the final code for some small changes.
```



The code to product this version of the plot is below:

```
pdf("./Homework/RMasterPlot.pdf", width=10, height=7)
  pdfFonts("Palatino")
  par(mar=c(5,5,2,2), family="Palatino")
  colors <- c("yellow", "red", "orange", "turquoise", "lightblue", "blue")
# null plot
  plot(X, Y, log="x", type="n", xlab="", xaxt="n", ylab="", yaxt="n", xlim=c(400,50000))
# grid lines
  ygrid <- c(seq(500,1000,100), seq(1000,10000,1000), seq(10000,50000,10000))
  xgrid <- seq(45,85,5)
  abline(v = ygrid, col="darkgrey")
  abline(h = xgrid, col="darkgrey")
# data points
  points(X, Y, cex=sqrt(P/max(P))*10, bg=colors[R], pch=21,xpd=TRUE, col="darkgrey")
# axes
  axis(1, at = c(5,10,20,50,100,200,500)*100,
       label = c("500", "1 000", "2 000", "5 000", "10 000", "20 000", "50 000"))
  axis(2, las=2)
```

```
# margin text
  mtext(paste(c("Low", "Middle", "High"), "-income countries", sep=""),
  side=3, line=0, adj=c(.1,.5,.9))
  mtext(expression(paste(italic("Money")," GDP per person in US (purchasing power adjusted) (log scale)
        side=1, line=3, adj=0.05, cex=1.5)
  mtext(expression(paste(italic("Health")," Life expectancy at birth (years)")),
        side=2, line=2.5, adj=0.05, cex=1.5)
# legend
  pt.cex <- c(rep(NA,5), rep(2, 6))
  legend <- c(rep(NA,5), levels(R))
  pt.cols <- c(rep(NA,5), colors)
  legend("bottomright", pt.cex=pt.cex, legend=legend, pch=21, bg="white", pt.bg = pt.cols)
  # adding population points and text
  xs <- c(15000, 19000, 28000, 42000)
  ys <- rep(58,4)
  points(xs, ys + c(-.5,-.5,.5,.5), pch=21, bg="grey", cex=sqrt(c(3,10,300,1000)*1e6 / max(P))*10)
  text(xs, ys + c(+.5,+.5,.5,.5), c("3", "10", "300", "1000"))
  text(11000, ys + 3.9, "Population (x 1 million)", cex=1.25, pos=4)
  text(11000, ys - 2.1, "Global Region", cex=1.25, pos=4)
# adding 10 largest country text
  dx <-  1.1^c(-1,1,0,0,1,-1,0,0,0,0)
  dy <- c(-1,1,0,0,1,-1,0,0,0,0)/2
  text(X[1:10]*dx, Y[1:10]+dy, C[1:10], cex=sqrt(P[1:10]/max(P[1:10])*5))
# add a little stamp at the end
  text(350, 83, "RMaster World Map", cex=3, col="grey", pos=4)
dev.off()
```