PC3b – Additional Exercises

1. R-function:

```
geometric <- function(r,n){
  # calculates the sum of the geometric series
  #  1 + r + r^2 + r^3 + ... r^n
  v <- 0:n
  terms <- r^v
  return(sum(terms))
}
```

2. R-script:

```
Euler <- function(n){
  i <- 1:n
  return( sum((1/i)^2) )
}

cat(sprintf("pi^2/6 is %.4f\n", pi^2/6))
n <- 50
cat( sprintf("The sum of the first %d terms
is %.4f\n",n,Euler(n)) )
```

3. R-function:

```
maxmat <- function(mat){
  r=nrow(mat)
  c=ncol(mat)
  maxcol <- mat[1,]
  for(j in 1:c){
    for(i in 2:r){
      if(mat[i,j]>maxcol[j]){
        maxcol[j] <- mat[i,j]
      }
    }
  }

  maxrow <- mat[,1]
  for(i in 1:r){
    for(j in 2:c){
      if (mat[i,j] > maxrow[i]){
        maxrow[i] <- mat[i,j]
      }
    }
  }

  maxtot <- mat[1,1]
  for(i in 1:r){
    for(j in 1:c){
      if(mat[i,j] > maxtot){
        maxtot <- mat[i,j]
      }
    }
  }
  return(list(maxcol=maxcol,maxrow=maxrow,maxtot=maxtot))
}

set.seed(321)
x<-matrix(sample(1:30,3*5,replace=T),3,5); x
results <- maxmat(x)
results
```

4. R-function with subfunction:

```
areatri <- function(x1,y1,x2,y2,x3,y3){
  # Calculates the area of the triangle

    dist <- function(x1,y1,x2,y2){
    # Calculates the distance between any two points
    return( sqrt((x1-x2)^2 + (y1-y2)^2) )
  }

  a = dist(x1,y1,x2,y2);
  b = dist(x2,y2,x3,y3);
  c = dist(x3,y3,x1,y1);
  s = 0.5*(a+b+c);
  return( sqrt(s*(s-a)*(s-b)*(s-c)) )
}
```

Answer area:

```
> areatri(0,0,4,0,4,3)
[1] 6
```

5. R-script:

```
Approx <- function(x,M){
  taylor <- rep(0,length(x))
  for(n in 1:M){
    taylor=taylor+2*x^(2*n-1)/(2*n-1);
  }
  return(taylor)
}

M<-5
x <- seq(-0.99,0.99,by=0.03)
f <- log((1+x)/(1-x))
g <- Approx(x,M)
library("ggplot2")
ggplot(mapping=aes(x=x)) +
  theme(text = element_text(size = 18)) +
  geom_line(aes(y=f),size=1,col="blue") +
  geom_point(aes(y=g),shape=8,size=2,col="red") +
  xlab("x") +
  ylab("f(x) = solid blue line") +
  labs(title = "Function and its Taylor approximation")

cat('  x     log((1+x)/(1-x))      Taylor\n');
cat('--------------------------------------\n');
for(i in 1:5){
  xx=x[i];
  cat( sprintf('%.2f  %14.6f   %14.6f\n',xx,log((1+xx)/(1-
xx)),g[i]) )
}

cat('log((1+x)/(1-x))~2(');
for(n in 1:M){
  cat( sprintf('x^%d/%d',(2*n-1),(2*n-1)) )
  if(n<M)
    cat('+')
  else
    cat(')\n')
}
```