Question & Answers

1-) Andrew curves are not so specific like parallel plots. In scaling we have found more dispersed results and changing size, allows us to see in greater detail how the variables are accumulated and directed towards. While Andrew`s curves do not exactly give there are correlations, they are good to know whether there is something worth to look at it. Then it is better to check with parallel plots.

|  |  |
| --- | --- |
| Code | Result |
| ```{r}  library ( DAAG )  library(MASS)  attach ( crabs )  library(andrews)    andrews(crabs[,c(4:8)],clr=5,ymax=4.2)  andrews(crabs[,c(4:8)],type=4,clr=5,ymax=2.5)  ``` | Before Adjusting      After Adjusting: |
| library(GGally)  library(MASS)  s.cale <- c('globalminmax','std')  o.rder <- c("anyClass", 'allClass', 'skewness' )  d.ata <- crabs[seq(5, 200, 2), ] | Without Scaling: |
| ggparcoord(data = d.ata,  columns = 4:8, # Columns to select  alphaLines = 0.4, # Transparency of the lines )  showPoints = TRUE, # Adding points  scale = s.cale[1], # Scaling Parameter  order = o.rder[1], # Ordering parameter  groupColumn = "sex") + scale\_color\_brewer(palette = "Set2") + facet\_wrap(~ sp) + coord\_flip() |
| ggparcoord(data = d.ata,  columns = 4:8, # Columns to select  alphaLines = 0.4, # Transparency of the lines )  showPoints = TRUE, # Adding points  scale = s.cale[2], # Scaling Parameter  order = o.rder[1], # Ordering Parameter  groupColumn = "sex") + scale\_color\_brewer(palette = "Set2") + facet\_wrap(~ sp) + coord\_flip() | With Scaling |
| #Implementation of same codes with this size difference:  d.ata <- crabs[seq(5, 200, 2), ] | Without Scaling  Scaled |

2-) We found the cdf and change boundaries for pdf function. We get the sample and means. Then we estimate the variance in terms of confidence intervals.

|  |  |
| --- | --- |
| Code | Results |
| # Our Cdf Function  s.cdf <- function(x) {  ((1/sqrt(2\*pi))\*exp( -(t^2)/2) )  }  # We need to have pdf and change boundries  # Then we are going to estimate means from it  Varience\_Estimation<-function(x,n){  u<-runif(n)  if (x<0) {  # After conversion, our pdf function to create sample in which x is lower than 0  samples<- (0.5)-(-x\*exp(-x^2\*u^2/2))/sqrt(2\*pi)  estimate\_for\_sample<-mean(samples)  } else {  # After conversion, our pdf function to create sample in which x is higher than 0  samples<-x\*exp(-x^2\*u^2/2)/sqrt(2\*pi)+.5  estimate\_for\_sample<-mean(samples)  }  v<-var(samples)/n  # Confidence Intervals  Lower\_Bound<-estimate\_for\_sample-1.96\*sqrt(v)  Upper\_Bound<-estimate\_for\_sample+1.96\*sqrt(v)  cat("We estimate the pdf of Z=",x,"to be",estimate\_for\_sample,"with variance",v,"when n=",n,".\n And a 95% Confidence Interval for Z=",x," is (",Lower\_Bound,",",Upper\_Bound,").")  }  Varience\_Estimation(2,1000) |  |

3-) Both methods allow us to reduce variance in great amount. Control variate method shows greater performance than Antithetic method, but differences are so small from each other.

|  |  |  |
| --- | --- | --- |
| Code | Result | |
| ```{r}  g <- function(t) {sin(t)}  n <- 10^4  a <- 0; b <- pi/3    x <- runif(n = n, min = a , max = b)    g.Xbar = mean(sapply(x ,g))  estimate <- (b-a)\*g.Xbar  estimate  integrate(g, a, b)  ``` | Mean results of simple MC implementation and integral in given bounds.  metin içeren bir resim  Açıklama otomatik olarak oluşturuldu | |
| ```{r}  var.estimate <- var(sapply(x,g))  var.estimate <- mean(sapply(x,g)^2) - (mean(sapply(x, g)))^2  var.estimate  ``` | Varience Estimation | |
| ```{r}  f <- function(t) { cos(90-t) } # sqrt(1 - (cos(t))^2 )    range <- seq(0,360,by=0.01)  ff <- sapply(range ,f)  gg <- sapply(range ,g)   hist(ff)  hist(gg)  ``` | Basic function and control variate function. | |
| # Control Variate For Variance Reduction | |  | |
| ```{r}  n <- 10^4  a <- 0; b <- pi/3   # main function  g.cont <- function(t) {sin(t)}  # control function  f.cont <- function(t) { cos(90-t) }   # Calculation of correlation  cat('correlation is high', cor(g.cont(x),f.cont(x)),collapse='\n') # correlation is high, we satisfy the correlation between f and g.    c\_star <- -cov(g.cont(x),f.cont(x)) / var(f.cont(x))  cat('c\_star level:', c\_star,collapse='\n')    estimate\_cont = mean(g.cont(x) + c\_star \* (f.cont(x) - (1/(b-a))\*integrate(f.cont, a, b)$value))\*(b-a)    cat("Estimate mean using control variates is", estimate\_cont,collapse='\n')    var.cont.estimate <- var(g.cont(x) + c\_star \* (f.cont(x) - integrate(f.cont, a, b)$value))  cat("Varience using control variates is", var.cont.estimate)  ``` | | metin içeren bir resim  Açıklama otomatik olarak oluşturuldu | |
| ```{r}  result <- matrix(c(estimate,estimate\_cont, var.estimate, var.cont.estimate),nrow=2,  dimnames = list(c("classical approach","control variates"),  c("estimated mean","variance")))  result  ``` | metin içeren bir resim  Açıklama otomatik olarak oluşturuldu | | |
| ```{r}  var\_reduction = 100 \* ((var.estimate - var.cont.estimate ) / var.estimate)    cat("The control variates approach achived",round(var\_reduction,2), "%", "reduction in variance.")  ``` |  | | |

|  |  |
| --- | --- |
| # Antithetic |  |
| ```{r}  n=10^4  a <- 0; b <- pi/3    f <- function(t) {sin(t)}  g <- function(t) { cos(90-t) }    x <- runif(n/2, a, b)    # Step 3: Obtain Antithetic variables    z1 <- f(x)  z2 <- f(b-x)    antithetic <- (z1 + z2)/2    # Step 3: Compute (b-a)\*g(Xbar)    theta1 <- mean(z1) \* (b-a)  theta2 <- mean(z2) \* (b-a)    ant.estimate <- mean(antithetic)\*(b-a)    cat("Antithetic theta hat is ", ant.estimate,"\n")  var.ant.estimate <- (var(z1)+var(z2)+2\*cov(z1,z2)) / 4  cat("Antithetic variance of theta hat is ", var.ant.estimate,collapse='\n')    var\_reduction = 100 \* ((var.theta.hat-var.ant.estimate)/var.theta.hat)  cat("The antithetic variable approach achived",collapse='\n',  round(var\_reduction,2), "%", "reduction in variance.")  ``` |  |