# Seminar

## Giới thiệu dữ liệu

Dữ liệu “Students Performance in Exams” được lấy từ roycekimmons.com với 1000 quan trắc và 8 biến như sau:

1. **gender:** Giới tính.
2. **race/ethnicity:** Chủng tộc / Dân tộc.
3. **parental level of education:** Trình độ học vấn của cha mẹ.
4. **lunch:** Bữa ăn trưa.
5. **test preparation course:** Khóa luyện thi.
6. **math score:** Điểm toán.
7. **reading score:** Điểm đọc.
8. **writing score:** Điểm viết.

### Thư viện

```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.3.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(ggplot2)
library(MASS)
library(nortest)
library(bayestestR)
```

```
## Warning: package 'bayestestR' was built under R version 4.3.2
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.3.2
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:ggpubr':
##
##     mutate
```

```
library("numDeriv")
```

### Đọc dữ liệu, kiểm tra dữ liệu khuyết và trực quan hóa dữ liệu

```
data <- read.csv("exams.csv")
data <- rename(data, c("math.score" = "Math", "reading.score" = "Read", "writing.score" = "Writing"))
str(data)
```

```
## 'data.frame':    1000 obs. of  8 variables:
##  $ gender                     : chr  "male" "female" "male" "female" ...
##  $ race.ethnicity             : chr  "group C" "group E" "group D" "group C" ...
##  $ parental.level.of.education: chr  "high school" "some college" "some college" "associate's degree
##  $ lunch                      : chr  "standard" "standard" "free/reduced" "standard" ...
##  $ test.preparation.course    : chr  "none" "completed" "completed" "none" ...
##  $ Math                       : int  75 67 81 85 56 50 50 74 58 82 ...
##  $ Read                       : int  57 67 77 97 57 67 62 76 62 100 ...
##  $ Writing                    : int  54 70 76 90 56 67 65 73 59 100 ...
```

```r
f<-function(data){any(is.na(data))}
apply(data, 2, f)
```

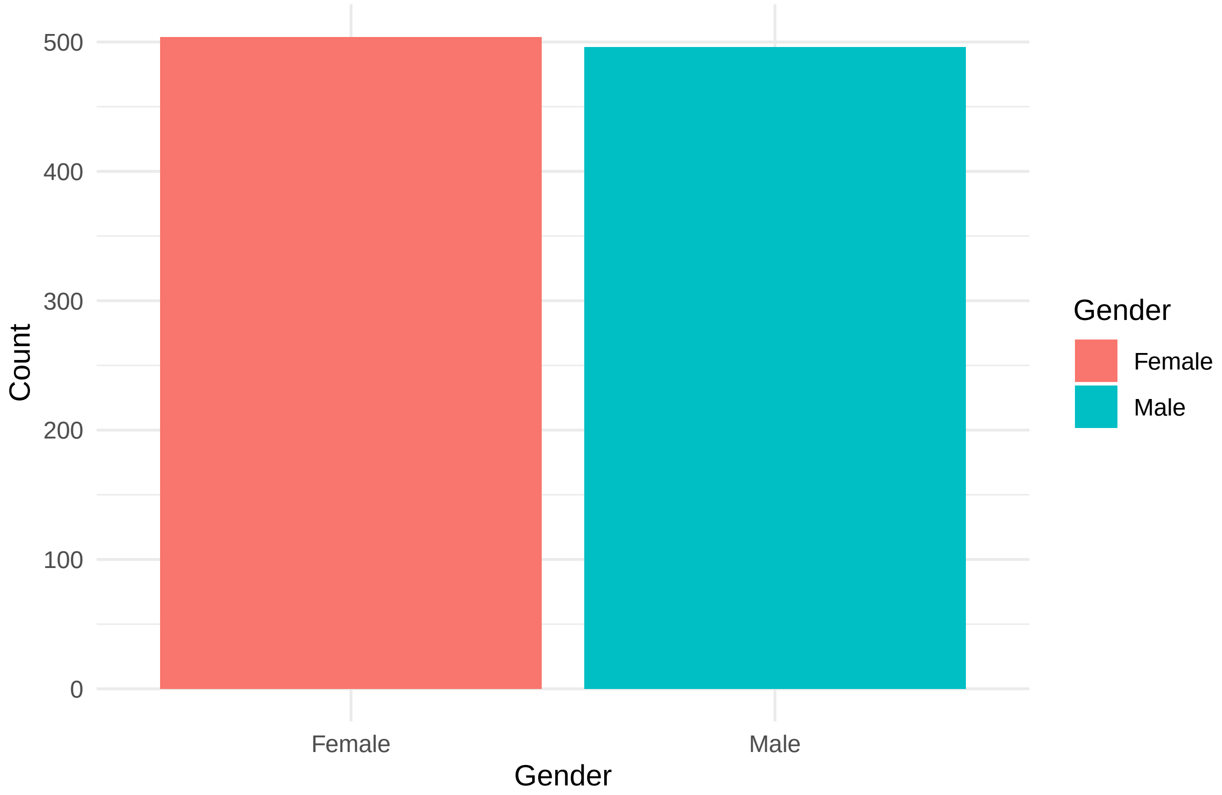
```
##                      gender              race.ethnicity
##                       FALSE                       FALSE
## parental.level.of.education                       lunch
##                       FALSE                       FALSE
##     test.preparation.course                        Math
##                       FALSE                       FALSE
##                        Read                     Writing
##                       FALSE                       FALSE
```

```r
for (i in 1:5){
  print(table(data[i]))
}
```

```
## gender
## female   male
##    504    496
## race.ethnicity
## group A group B group C group D group E
##      84     180     326     282     128
## parental.level.of.education
## associate's degree  bachelor's degree        high school    master's degree
##                207                128                193                 62
##       some college   some high school
##                221                189
## lunch
## free/reduced     standard
##          331          669
## test.preparation.course
## completed      none
##       311       689
```

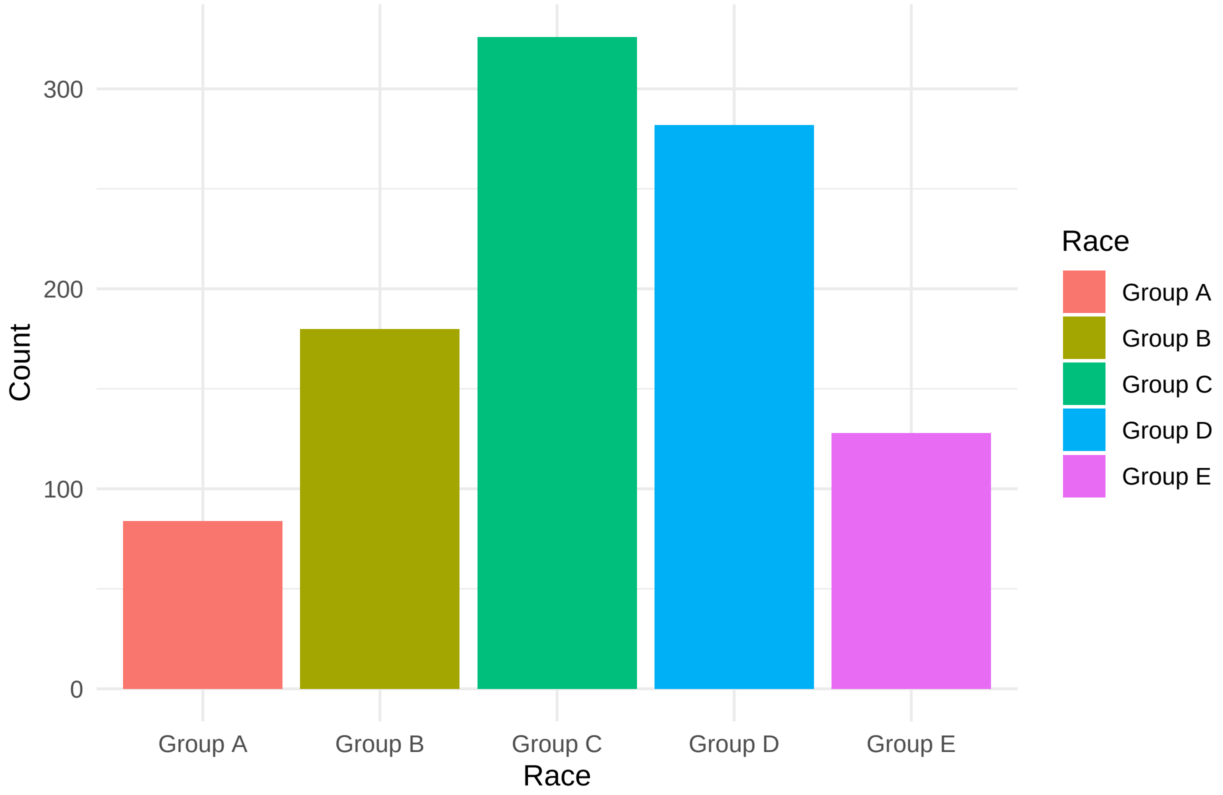
### Biểu đồ cột theo giới tính

```r
gender_data <- data.frame(Gender = c("Female", "Male"), Count = c(504, 496))

ggplot(gender_data, aes(x = Gender, y = Count, fill = Gender)) +
  geom_bar(stat = "identity") +
  labs(title = "", x = "Gender", y = "Count") +
  theme_minimal()
```



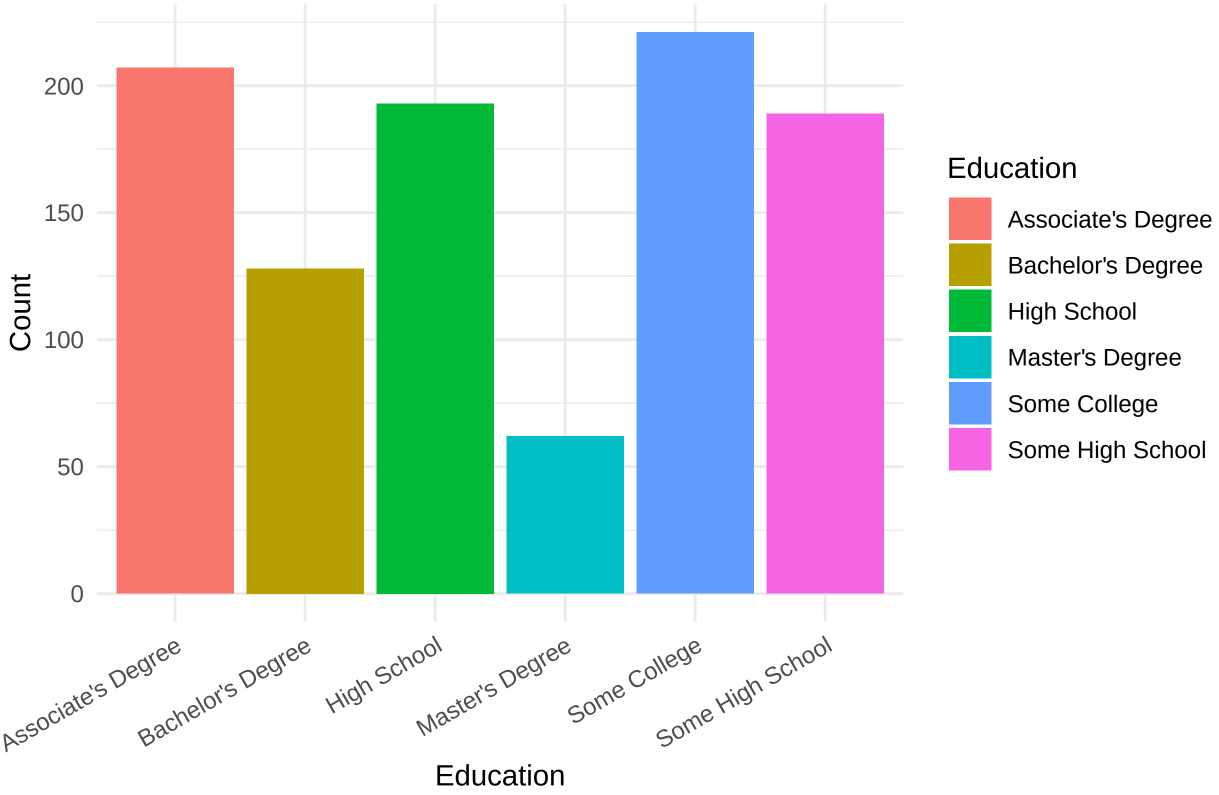
### Biểu đồ cột theo dân tộc / chủng tộc

```
race_data <- data.frame(
  Race = c("Group A", "Group B", "Group C", "Group D", "Group E"),
  Count = c(84, 180, 326, 282, 128)
)

ggplot(race_data, aes(x = Race, y = Count, fill = Race)) +
  geom_bar(stat = "identity") +
  labs(title = "", x = "Race", y = "Count") +
  theme_minimal()
```



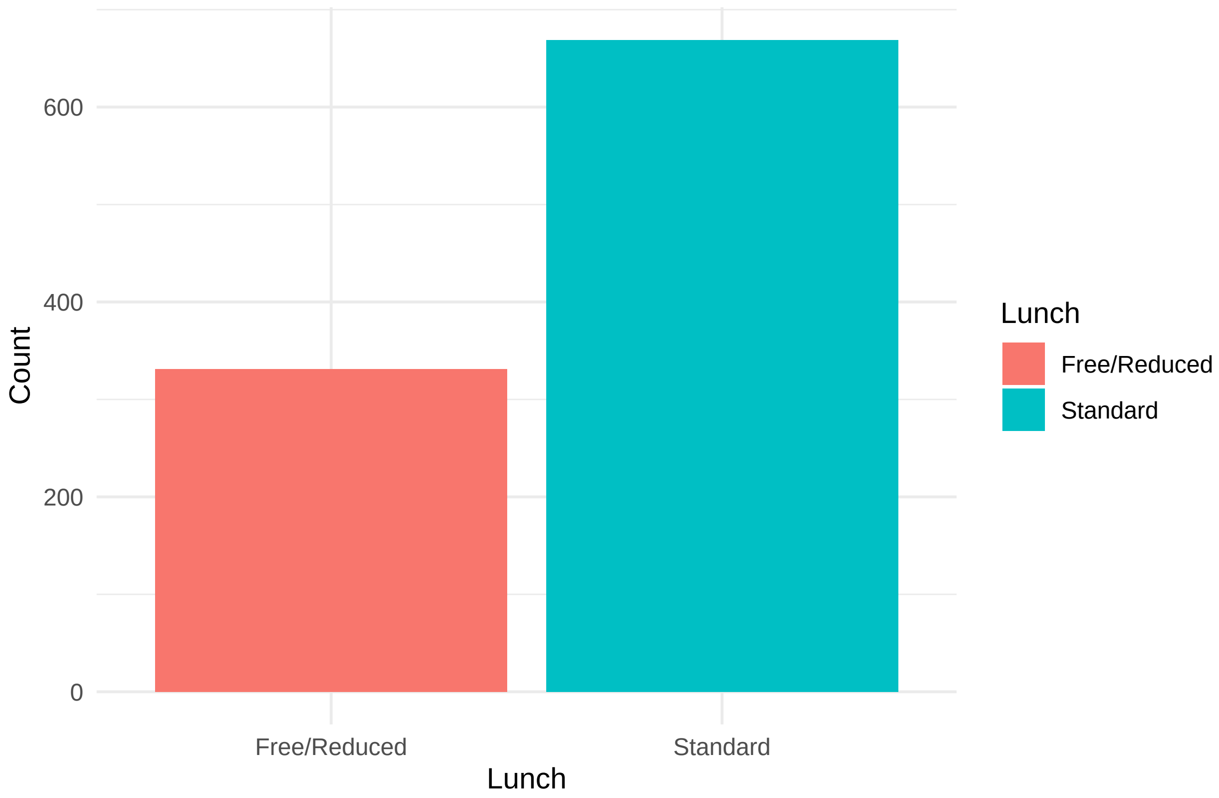
### Biểu đồ cột theo trình độ học vấn của cha mẹ

```r
education_data <- data.frame(
  Education = c("Associate's Degree", "Bachelor's Degree", "High School", "Master's Degree", "Some Coll
  Count = c(207, 128, 193, 62, 221, 189)
)

ggplot(education_data, aes(x = Education, y = Count, fill = Education)) +
  geom_bar(stat = "identity") +
  labs(title = "", x = "Education", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



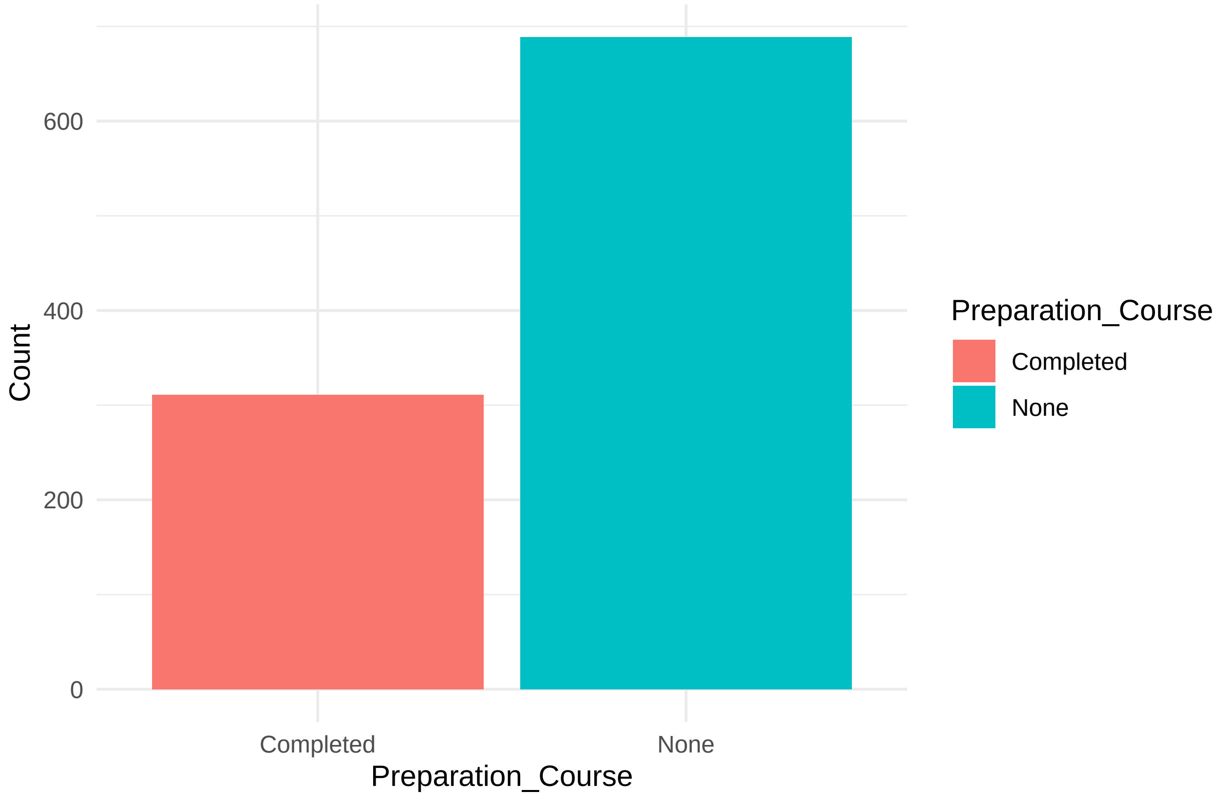
### Biểu đồ cột theo loại bữa trưa

```
lunch_data <- data.frame(
  Lunch = c("Free/Reduced", "Standard"),
  Count = c(331, 669)
)

ggplot(lunch_data, aes(x = Lunch, y = Count, fill = Lunch)) +
  geom_bar(stat = "identity") +
  labs(title = "", x = "Lunch", y = "Count") +
  theme_minimal()
```



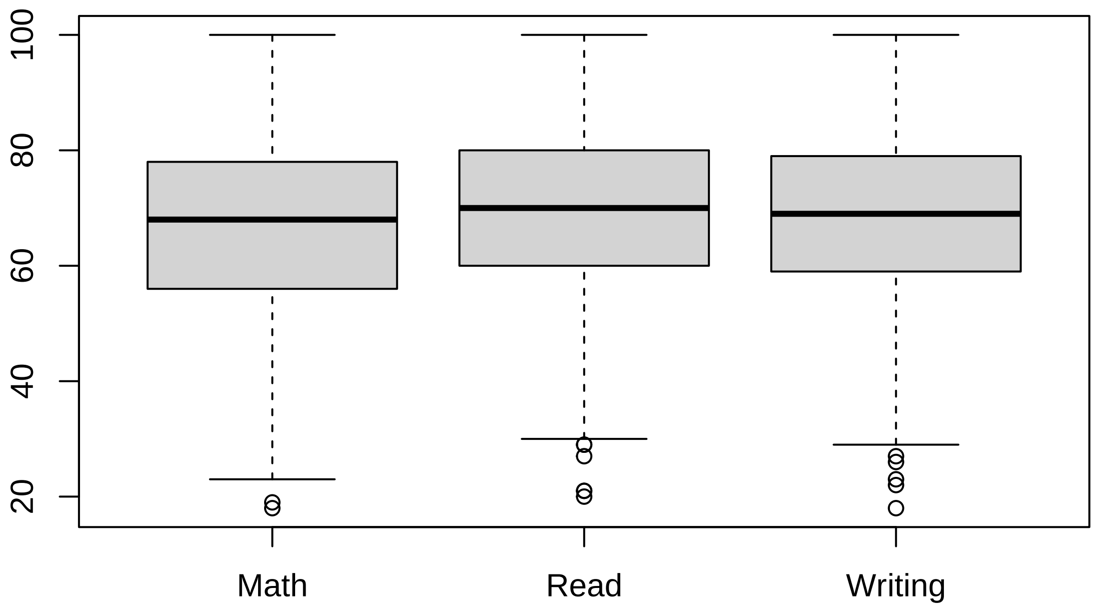
**Biểu đồ cột cho việc hoàn thành khóa luyện thi**

```r
prep_course_data <- data.frame(
  Preparation_Course = c("Completed", "None"),
  Count = c(311, 689)
)

ggplot(prep_course_data, aes(x = Preparation_Course, y = Count, fill = Preparation_Course)) +
  geom_bar(stat = "identity") +
  labs(title = "", x = "Preparation_Course", y = "Count") +
  theme_minimal()
```



### Boxplot

```r
boxplot(data[,c(6:8)])
```



**Ngoại lai và cực ngoại lai**

```
outliers <- function(x, i){ # 3 la cuc ngoai lai, 1.5 la ngoai lai
  # 1st and 3rd quantiles
  q75 = quantile(x, 0.75)
  q25 = quantile(x, 0.25)
  IQR = q75-q25
  # lower bound
  lower_bound = q25 - i * IQR
  # upper bound
  upper_bound = q75 + i * IQR
  # outliers
  outlier_ind <- which(x < lower_bound | x > upper_bound)
  if (length(outlier_ind) == 0){
    return (0)
  }
  return(outlier_ind)
}
for (i in 6:8) {
  cat("Cực ngoại lai -", names(data)[i], ": ", outliers(data[, i], 3), "\n")
  cat("Ngoại lai -", names(data)[i], ": ", outliers(data[, i], 1.5), "\n")
}
```

```
## Cực ngoại lai - Math :  0
## Ngoại lai - Math :  26 478
## Cực ngoại lai - Read :  0
## Ngoại lai - Read :  26 231 478 529 592 751
## Cực ngoại lai - Writing :  0
## Ngoại lai - Writing :  26 529 783 913 956
```

### Hàm định nghĩa biến đổi box-cox

```
boxcox_trans <- function(x, lambda){
  if (any(x < 0)) stop("x must be positive!")
  if (lambda == 0) return(log(x))
  else return((x^lambda - 1)/lambda)
}
```

### Định nghĩa hàm profile log-likelihood

```
llog_lambda <- function(lambda, x1, x2){ # x chua bien doi box-cox
  n1 <- length(x1)
  n2 <- length(x2)
  x1_lambda <- boxcox_trans(x = x1, lambda = lambda)
  x2_lambda <- boxcox_trans(x = x2, lambda = lambda)
  mu_x1_lambda <- mean(x1_lambda)
  mu_x2_lambda <- mean(x2_lambda)
  ll <- -0.5 * n1 *log(sum((x1_lambda - mu_x1_lambda)^2)/n1) - 0.5 * n2 *log(sum((x2_lambda - mu_x2_lam
    (lambda - 1)*(sum(log(x1)) + sum(log(x2)))
  return(ll)
}
```

### Ước lượng lambda bằng hàm optim()

```
out_lambda_math <- function(x1, x2) {
  optim(par = 1, fn = llog_lambda, method = "BFGS",
        control = list(fnscale = -1), x1 = x1, x2 = x2)
}
```

### Overlap Coefficient

```
povl <- function(bcmath, bcmaths) {
  mu1 <- mean(bcmath)
  sigma1 <- sd(bcmath)
  mu2 <- mean(bcmaths)
  sigma2 <- sd(bcmaths)
  x1 <- bcmath
  x2 <- bcmaths
  y1 <- dnorm(bcmath, mean = mu1, sd = sigma1) # dist1
  y2 <- dnorm(bcmaths, mean = mu2, sd = sigma2) # dist2

  # Create a function that returns the minimum density of two normal distributions for a given x value
  overlap_coef <- function(x) {
    pmin(dnorm(x, mu1, sigma1), dnorm(x, mu2, sigma2))
  }
  # Calculate the overlap coefficient between the two distributions
  oc <- integrate(overlap_coef, lower = -Inf, upper = Inf)$value
  cat("Overlap Coefficient: ", oc, "\n")
  # Create a data frame for the normal distributions and the overlap area
  df <- data.frame(x = c(x1, x2),
                   y = c(y1, y2),
                   group = factor(c(rep("Group 1", length(y1)), rep("Group 2", length(y2))),
                                  levels = c("Group 1", "Group 2")))
  df_overlap <- df %>%
    arrange(x) %>% #sap xep data.frame
    mutate(overlap = overlap_coef(x)) #tao them cot overlap_coef cho data.frame
  # Create the plot
  ggplot(df) +
    geom_area(data = df_overlap,
              aes(x = x, y = overlap, fill = "Overlap Area"),
              alpha = 0.2, position = "identity") +
    geom_line(size = 1, aes(x = x, y = y, color = group)) +
    scale_color_manual(values = c("blue", "red"),
                       name = "Group") +
    scale_fill_manual(values = "green",
                      name = "",
                      guide = guide_legend(override.aes = list(fill = "green", alpha = 0.2))) +
    theme_classic() +
    labs(title = "Overlap Coefficient of Two Normal Distributions",
         x = "X", y = "Probability Density",
         fill = paste0("Overlap Coefficient: ", round(oc, 2)))
}
```

```
ovl_normal <- function(par){
  mu1 <- par[1]
  mu2 <- par[2]
  sigma1 <- par[3]
  sigma2 <- par[4]
  a <- (sigma1^2 - sigma2^2)
  b <- (mu1*sigma2^2 - mu2*sigma1^2)
  c <- sigma1*sigma2
  d <- (mu1 - mu2)^2
  e <- log(sigma1^2/sigma2^2)
  x1 = (b - c*sqrt(d + a * e)) / -a
  x2 = (b + c*sqrt(d + a * e)) / -a
  ovl <- 1 + pnorm(x1, mu1, sigma1) - pnorm(x1, mu2, sigma2) - pnorm(x2, mu1, sigma1) + pnorm(x2, mu2, s
  return (ovl)
}
```

```
IC <- function(x, y) {
  mu1 <- mean(x)
  mu2 <- mean(y)
  sigma1 <- sd(x)
  sigma2 <- sd(y)
  par_s <- c(mu1, mu2, sigma1, sigma2)
  z <- jacobian(ovl_normal, par_s)
  varmu1 <- sigma1^2/length(x)
  varmu2 <- sigma2^2/length(y)
  varsigma1 <- 2 *sigma1^2 / length(x)
  varsigma2 <- 2 *sigma2^2 / length(y)
  var_OVL_s <- z[1] ^ 2 * varmu1 + z[2] ^2 * varmu2 + z[3]^2 * varsigma1 + z[4]^2 * varsigma2
  alpha <- 0.05
  z_normal <- qnorm(1 - alpha / 2)
  upper <- ovl_normal(par_s) + z_normal* sqrt(var_OVL_s)
  lower <- ovl_normal(par_s) - z_normal* sqrt(var_OVL_s)
  cat("Confidence interval for overlap coefficient: [", round(lower, 7), ",", round(upper, 7), "]\n")
}
```

Phân loại gender thành 2 nhóm gồm nữ (female) (487) và nam (male) (513).

```
math1 <- data$Math[data$gender == "female"]
math2 <- data$Math[data$gender == "male"]
lambda1 <- out_lambda_math(math1, math2)$par
bcmath1 <- boxcox_trans(math1, lambda1)
bcmath2 <- boxcox_trans(math2, lambda1)
shapiro.test(bcmath1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bcmath1
## W = 0.99566, p-value = 0.1767
```

```
shapiro.test(bcmath2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bcmath2
## W = 0.99308, p-value = 0.02204
```

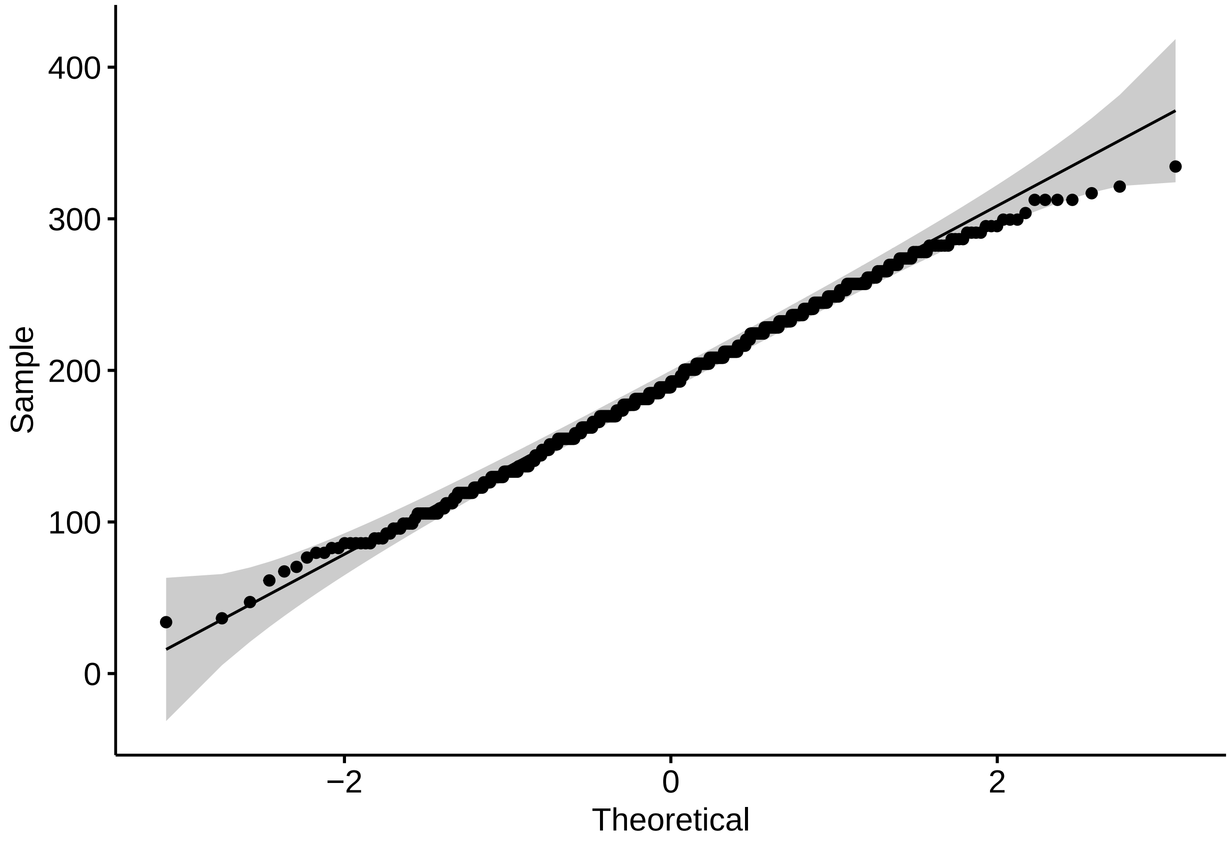
```
ad.test(bcmath1)
```

```
##
##  Anderson-Darling normality test
##
## data:  bcmath1
## A = 0.45582, p-value = 0.2663
```

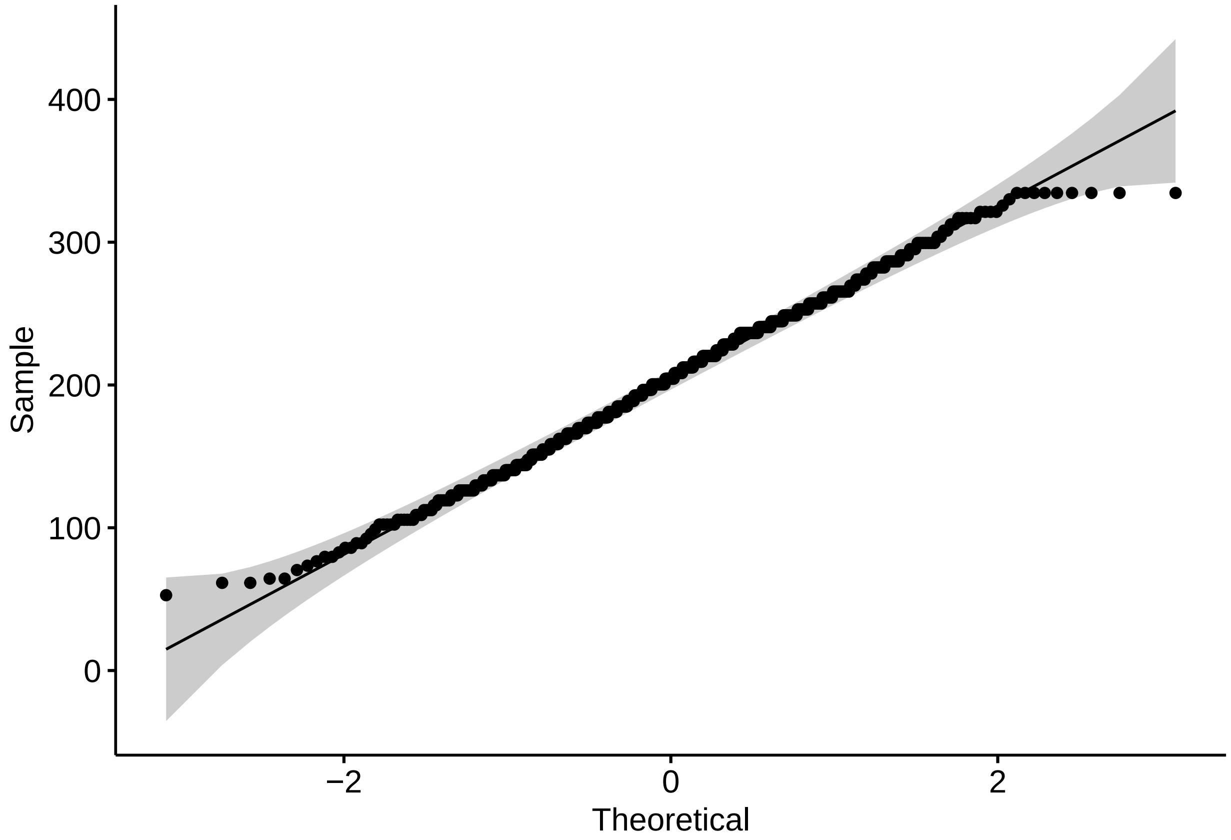
```
ad.test(bcmath2)
```

```
##
##  Anderson-Darling normality test
##
## data:  bcmath2
## A = 0.52618, p-value = 0.179
```

```
ggqqplot(bcmath1)
```



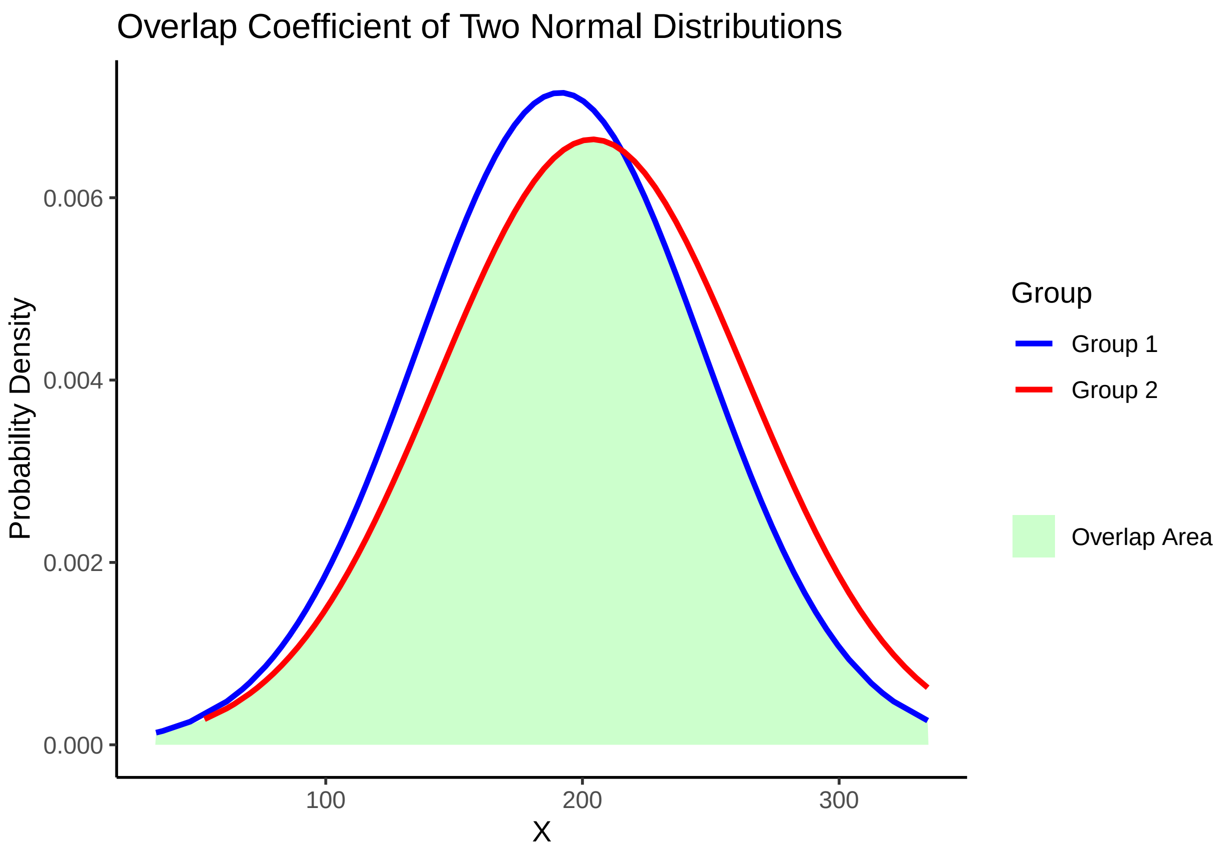
```
ggqqplot(bcmath2)
```



```
povl(bcmath1, bcmath2)
```

```
## Overlap Coefficient:  0.9087613
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



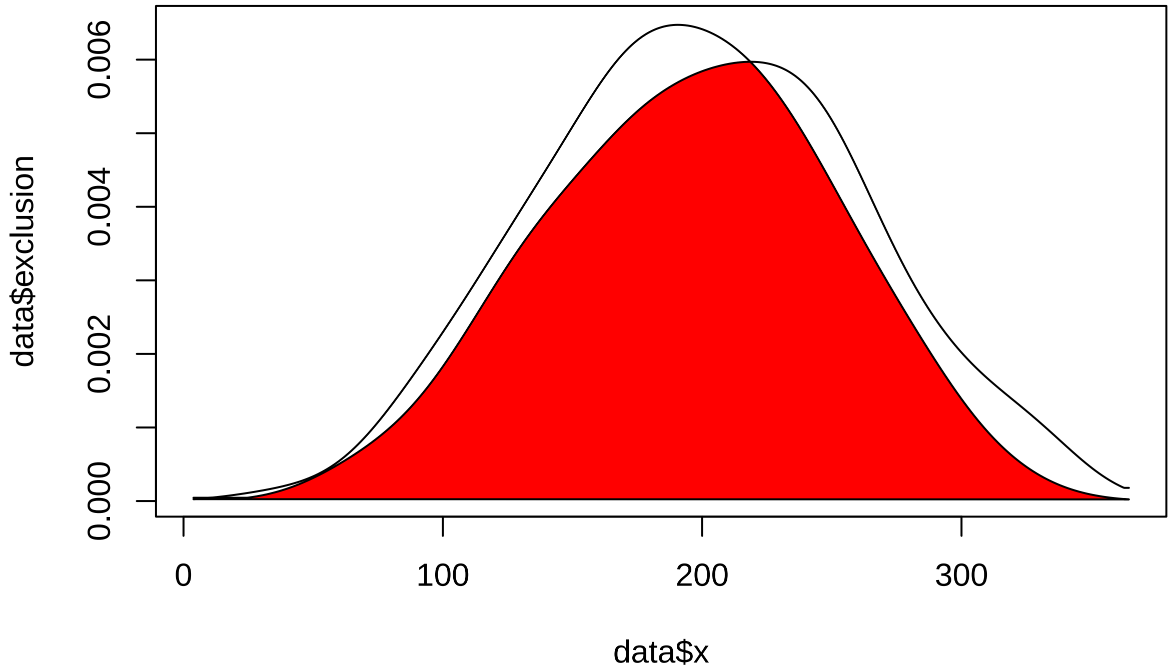
```
IC(bcmath1, bcmath2)
```

```
## Confidence interval for overlap coefficient: [ 0.8569935 , 0.9605244 ]
```

```
overlap(bcmath1, bcmath2)
```

```
## # Overlap
##
## 0.91
```

```
plot(overlap(bcmath1, bcmath2))
```



Phân loại parental.level.of.education thành 2 nhóm gồm cha mẹ có học thức cao (associate's degree, bachelor's degree, high school, master's degree) (594) và cha mẹ chưa có học thức thấp (some college, some high school)(406).

```
math3 <- data$Math[data$parental.level.of.education == "associate's degree"
                   | data$parental.level.of.education == "bachelor's degree"
                   | data$parental.level.of.education == "master's degree"
                   | data$parental.level.of.education == "high school"]
math4 <- data$Math[data$parental.level.of.education == "some college"
                   | data$parental.level.of.education == "some high school" ]
ad.test(math3)
```

```
##
##  Anderson-Darling normality test
##
## data:  math3
## A = 1.6684, p-value = 0.000277
```

```
ad.test(math4)
```

```
##
##  Anderson-Darling normality test
##
## data:  math4
## A = 0.39201, p-value = 0.377
```

```
lambda2 <- out_lambda_math(math3, math4)$par
bcmath3 <- boxcox_trans(math3, lambda2)
```

```r
bcmath4 <- boxcox_trans(math4, lambda2)
shapiro.test(bcmath3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bcmath3
## W = 0.99225, p-value = 0.003644
```

```r
shapiro.test(bcmath4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bcmath4
## W = 0.997, p-value = 0.6569
```

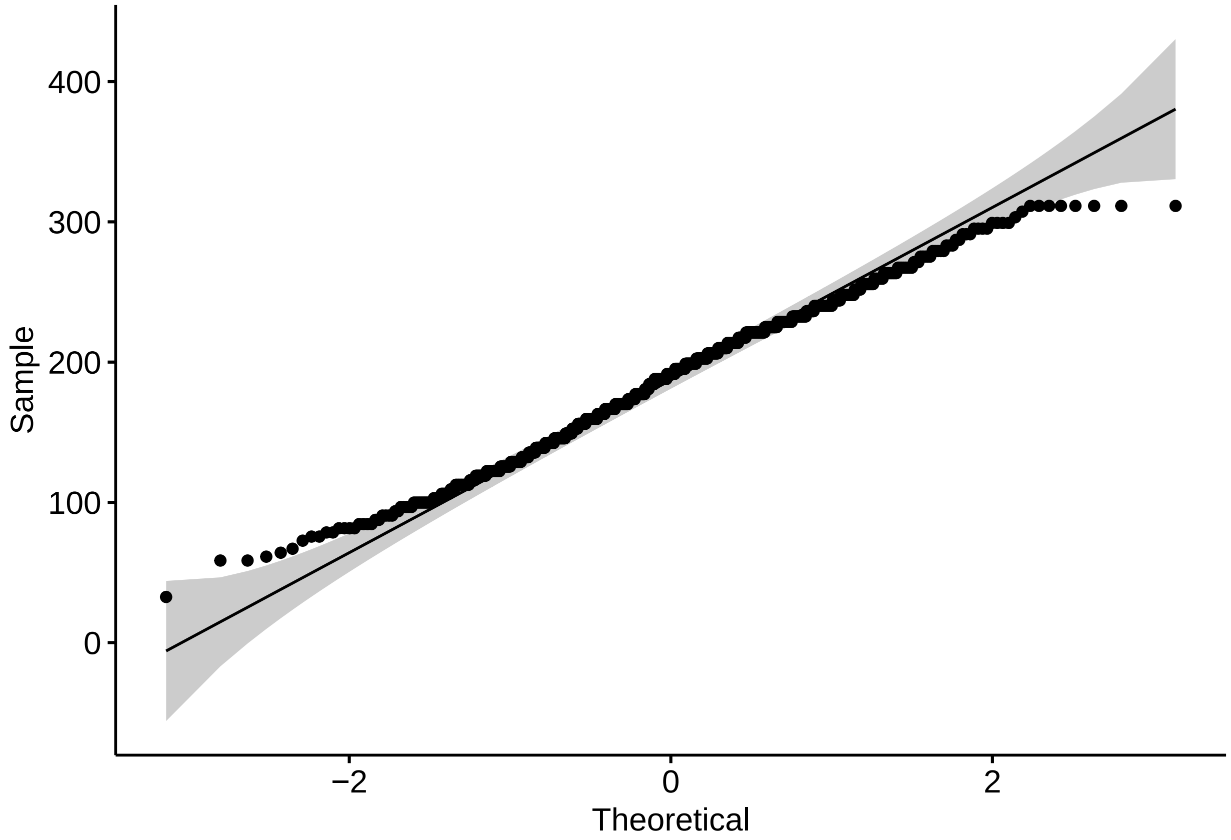
```r
ad.test(bcmath3)
```

```
##
##  Anderson-Darling normality test
##
## data:  bcmath3
## A = 1.1159, p-value = 0.006333
```

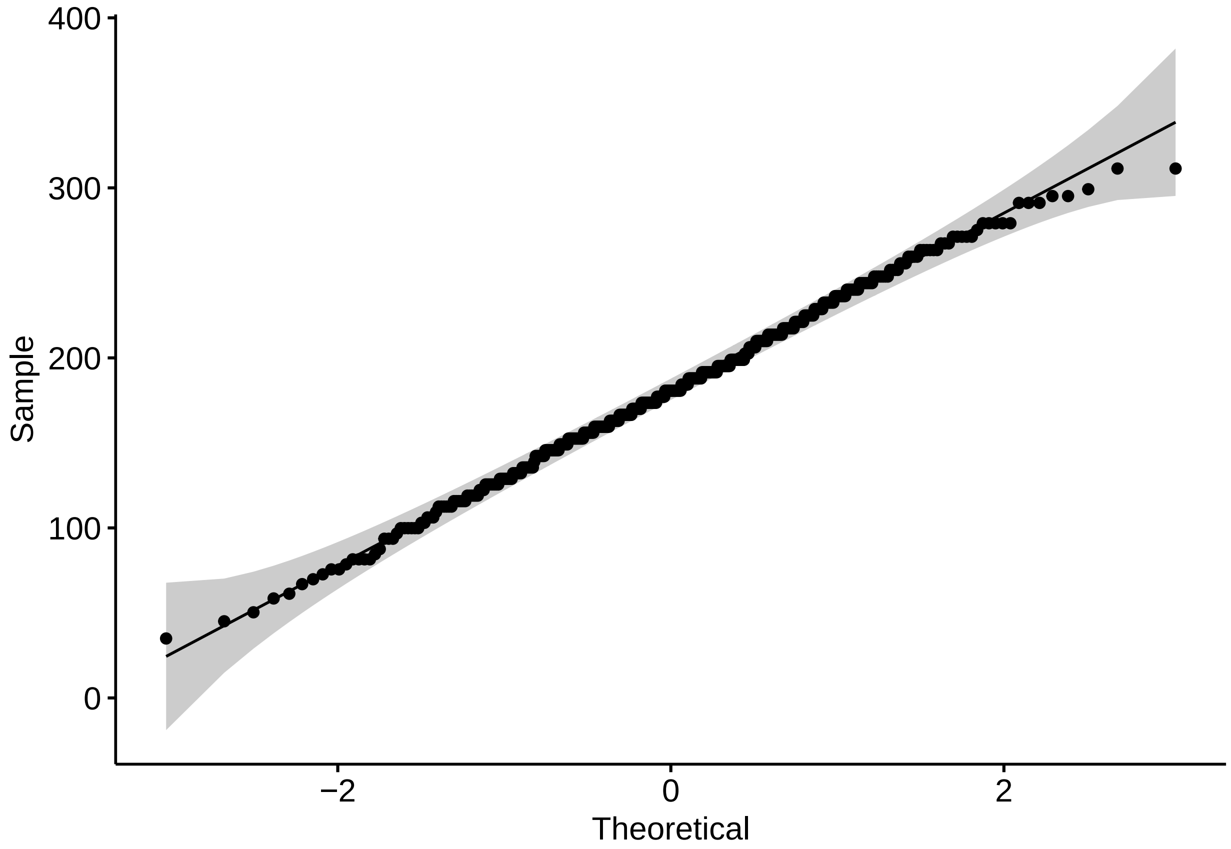
```r
ad.test(bcmath4)
```

```
##
##  Anderson-Darling normality test
##
## data:  bcmath4
## A = 0.25973, p-value = 0.7107
```

```r
ggqqplot(bcmath3)
```

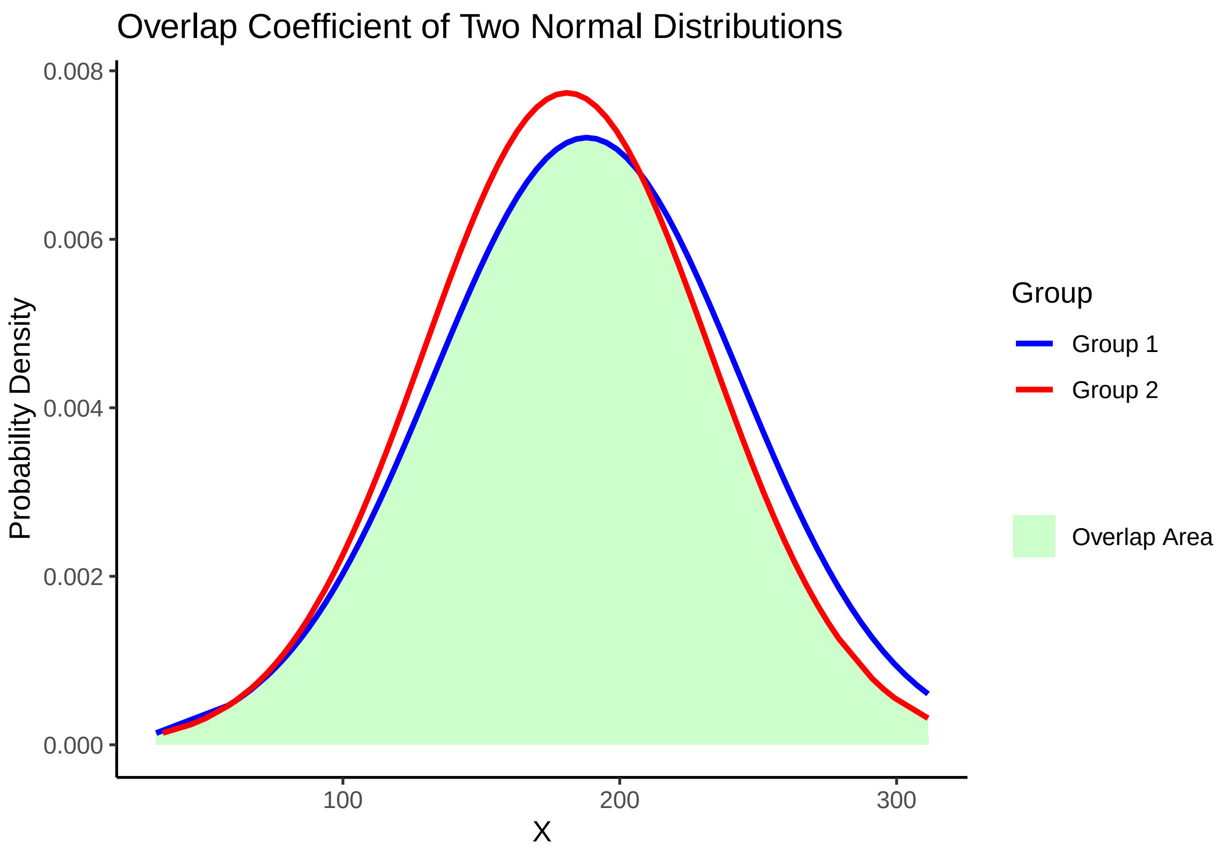


```
ggqqplot(bcmath4)
```



```
povl(bcmath3, bcmath4)
```

```
## Overlap Coefficient:  0.9394142
```



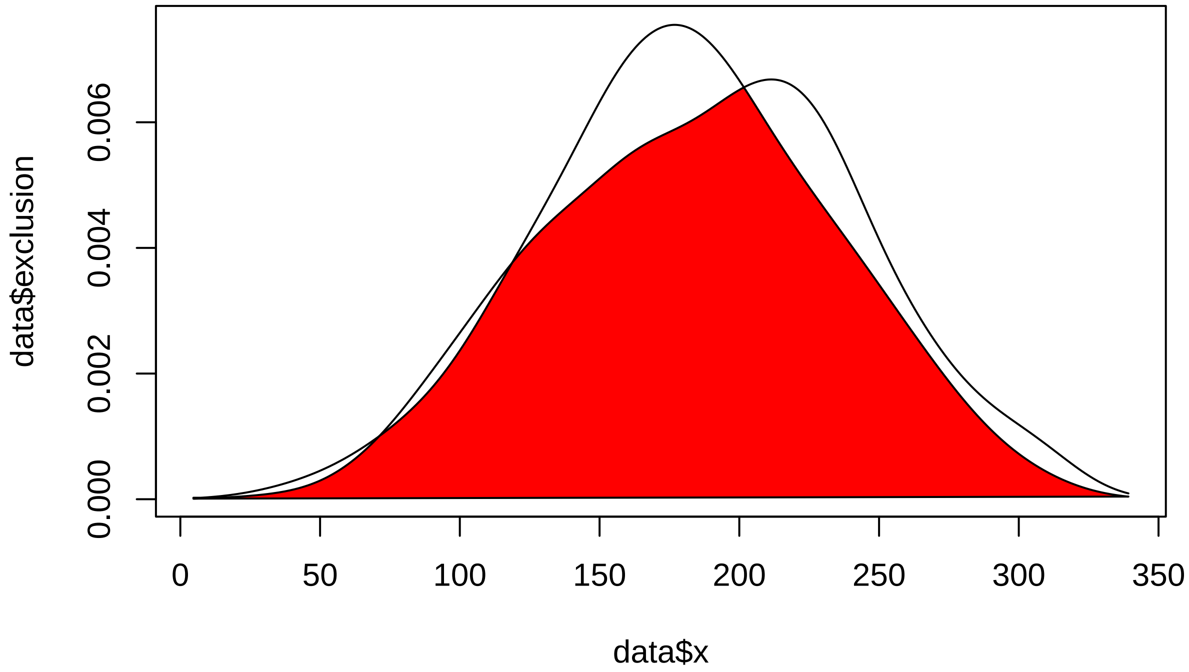
```
IC(bcmath3, bcmath4)
```

```
## Confidence interval for overlap coefficient: [ 0.8814869 , 0.9973536 ]
```

```
overlap(bcmath3, bcmath4)
```

```
## # Overlap
##
## 0.91
```

```
plot(overlap(bcmath3, bcmath4))
```



Phân loại lunch thành 2 nhóm gồm miễn phí/giảm tiêu (352) và chuẩn (648).

```
math5 <- data$Math[data$lunch == "free/reduced"]
math6 <- data$Math[data$lunch == "standard"]
ad.test(math5)
```

```
##
##  Anderson-Darling normality test
##
## data:  math5
## A = 0.37988, p-value = 0.4019
```

```
ad.test(math6)
```

```
##
##  Anderson-Darling normality test
##
## data:  math6
## A = 0.84819, p-value = 0.02899
```

```
lambda3 <- out_lambda_math(math5, math6)$par
bcmath5 <- boxcox_trans(math5, lambda3)
bcmath6 <- boxcox_trans(math6, lambda3)
shapiro.test(bcmath5)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bcmath5
## W = 0.9945, p-value = 0.2809
```

```
shapiro.test(bcmath6)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bcmath6
## W = 0.99542, p-value = 0.04543
```

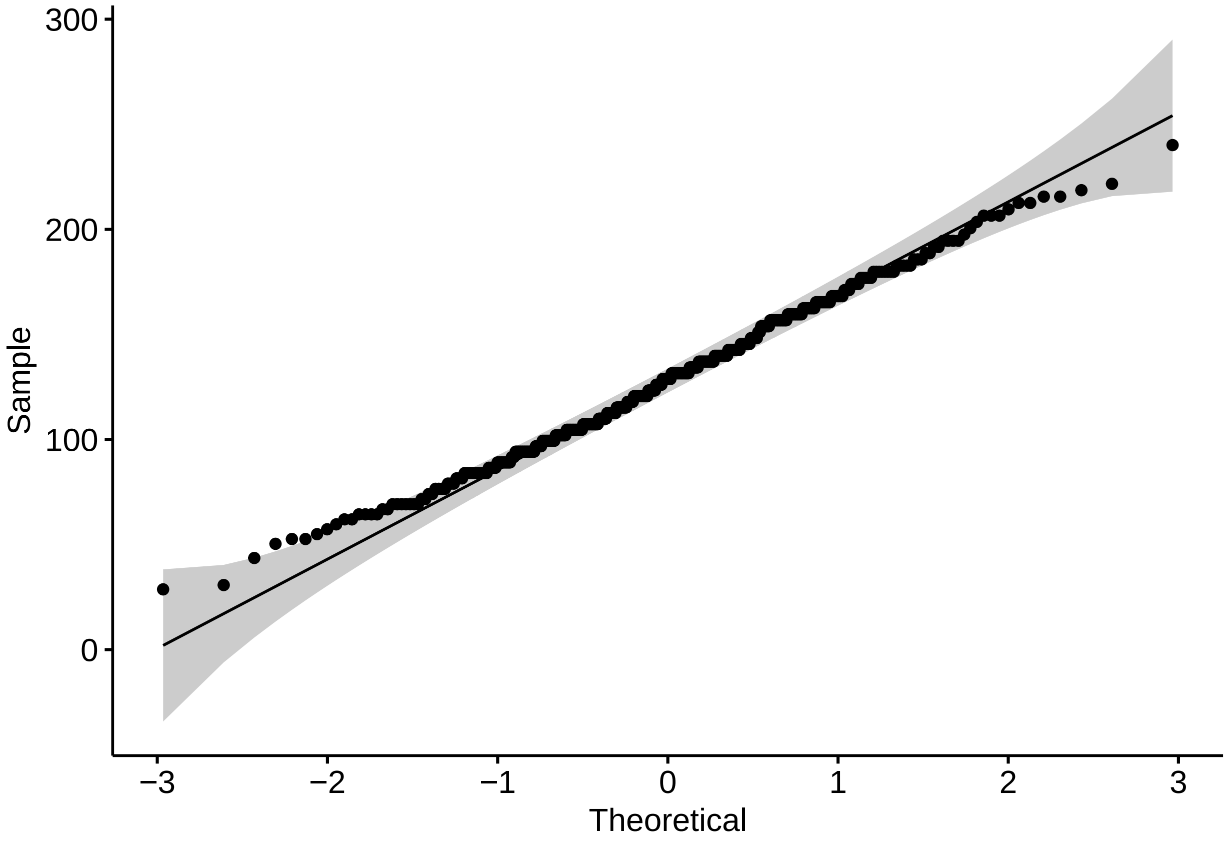
```
ad.test(bcmath5)
```

```
##
##  Anderson-Darling normality test
##
## data:  bcmath5
## A = 0.52179, p-value = 0.1831
```

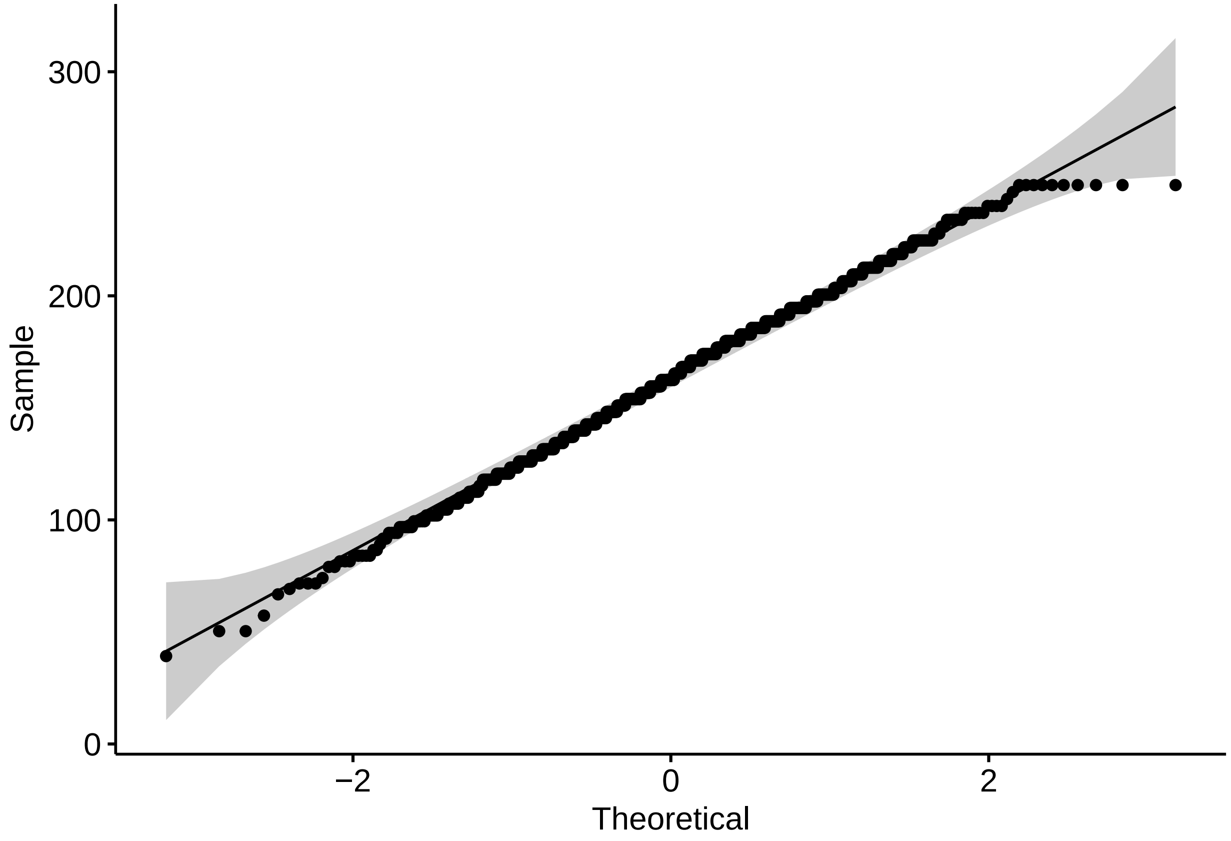
```
ad.test(bcmath6)
```

```
##
##  Anderson-Darling normality test
##
## data:  bcmath6
## A = 0.46455, p-value = 0.2539
```

```
ggqqplot(bcmath5)
```

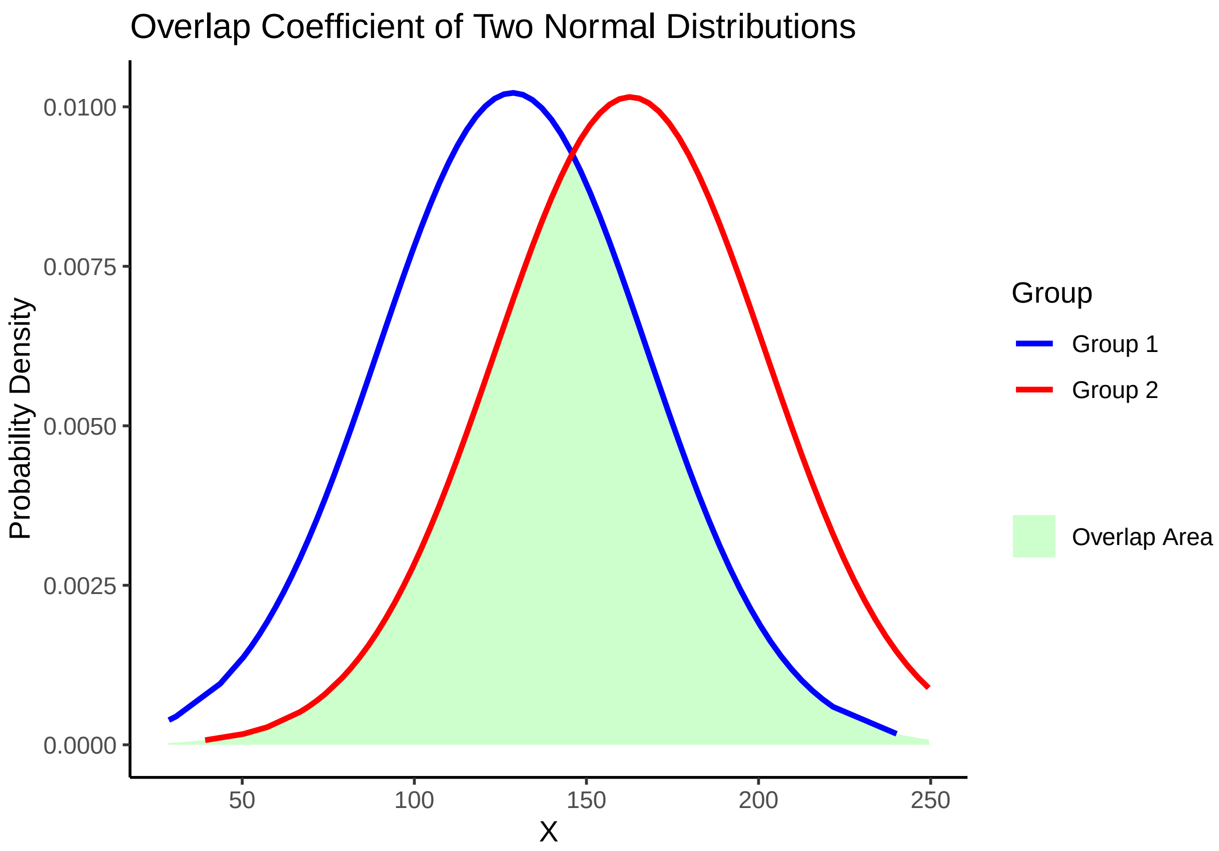


```
ggqqplot(bcmath6)
```



```
povl(bcmath5, bcmath6)
```

```
## Overlap Coefficient:  0.6626122
```



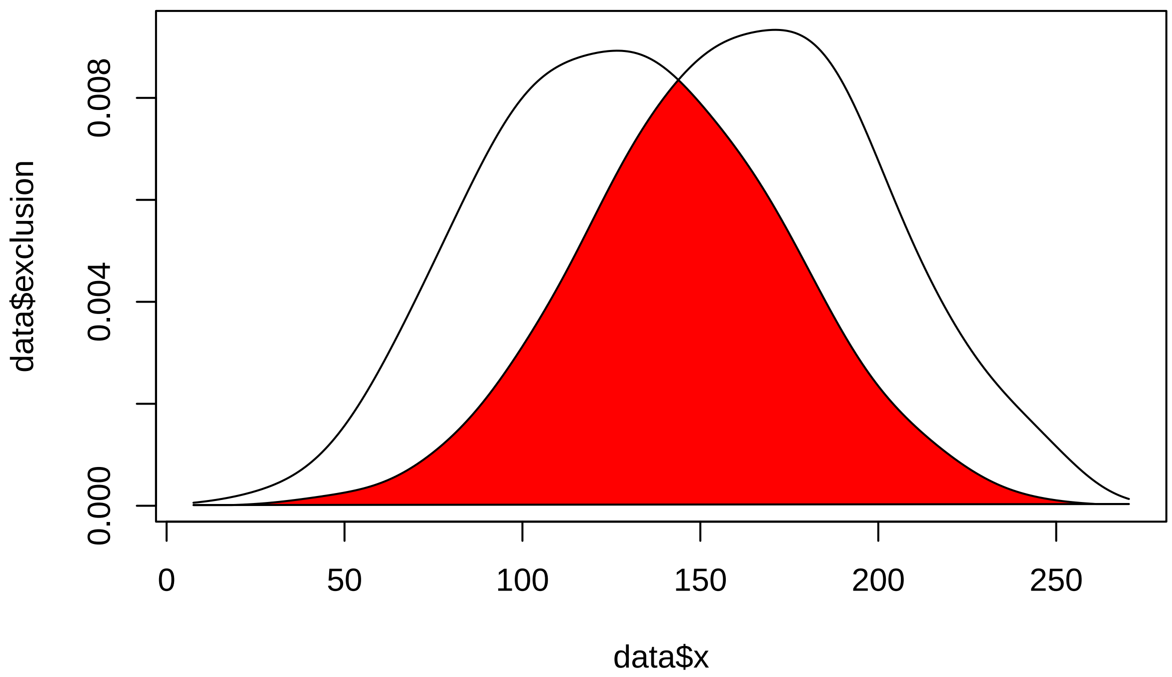
```
IC(bcmath5, bcmath6)
```

```
## Confidence interval for overlap coefficient: [ 0.6064487 , 0.7187759 ]
```

```
overlap(bcmath5, bcmath6)
```

```
## # Overlap
##
## 0.68
```

```
plot(overlap(bcmath5, bcmath6))
```


data$exclusion
0.000
0.004
0.008
0
50
100
150
200
250
data$x