Lab 2

Beatrice Dang

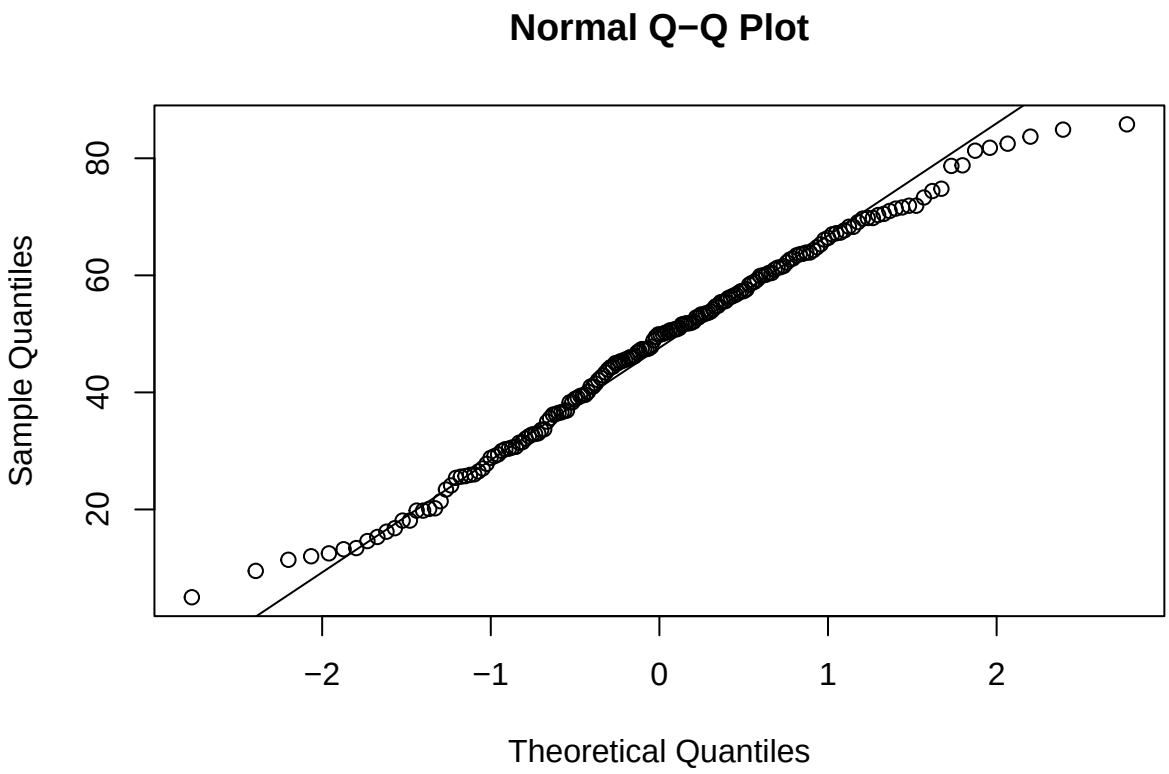
2024-09-20

Setup

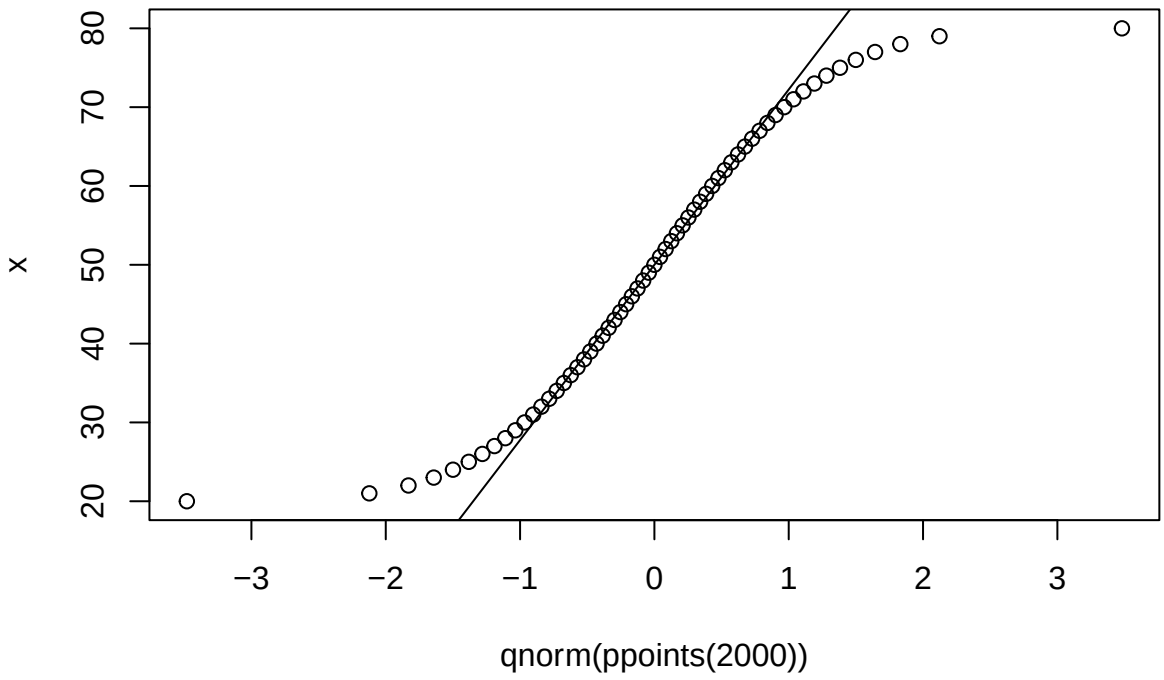
```
library(ggplot2)
population_data <- read.csv("C:\\Users\\bmd\\Downloads\\countries_populations_2023.csv")
epi_results <- read.csv("C:\\Users\\bmd\\Downloads\\epi2024results06022024.csv", header=TRUE)
epi_weights <- read.csv("C:\\Users\\bmd\\Downloads\\epi2024weights.csv")

attach(epi_results)
attach(epi_weights)
View(epi_results)
View(epi_weights)
```

Exercise 1: Fitting a Distribution Beyond Histograms

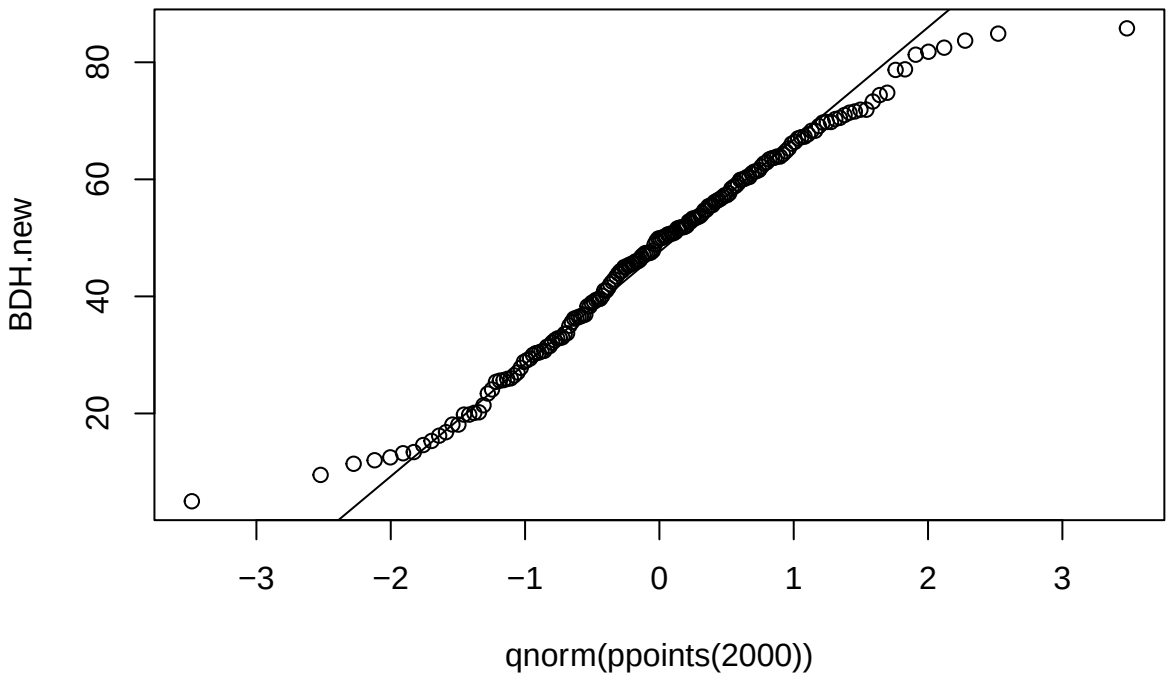
```
qqnorm(BDH.new); qqline(BDH.new)
```



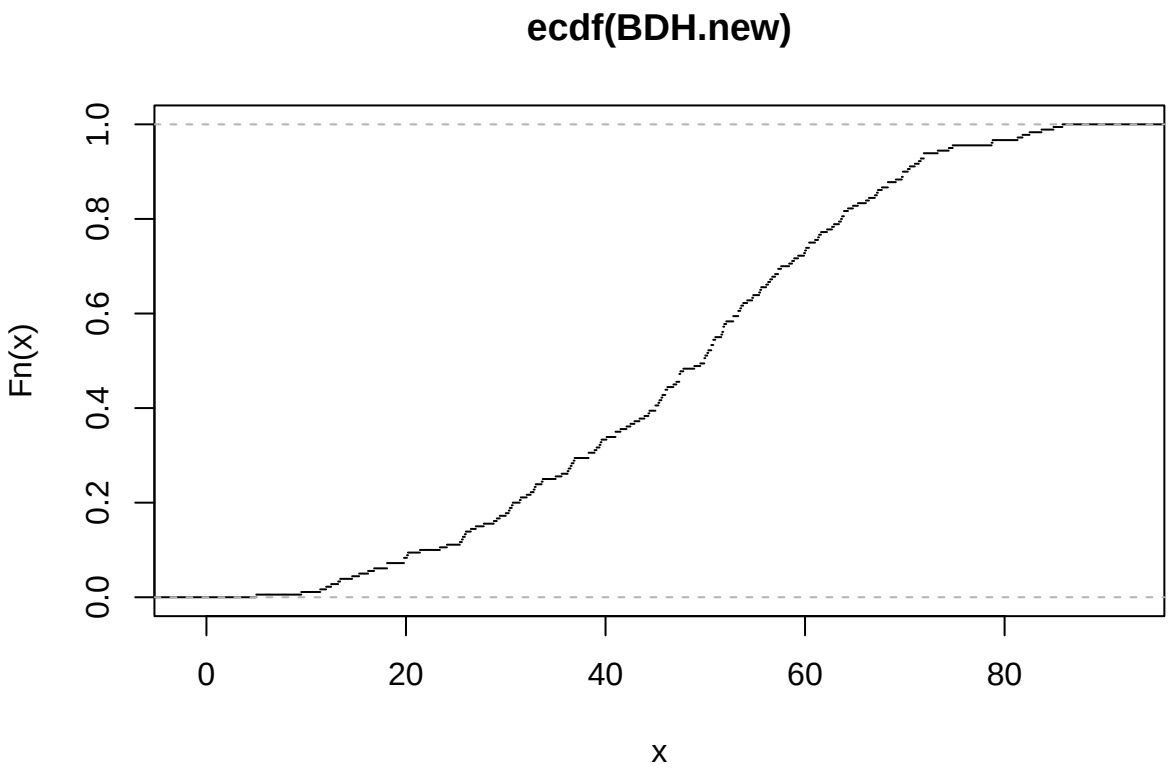

```
x <- seq(20., 80., 1.0)
qqplot(qnorm(ppoints(2000)), x)
qqline(x)
```



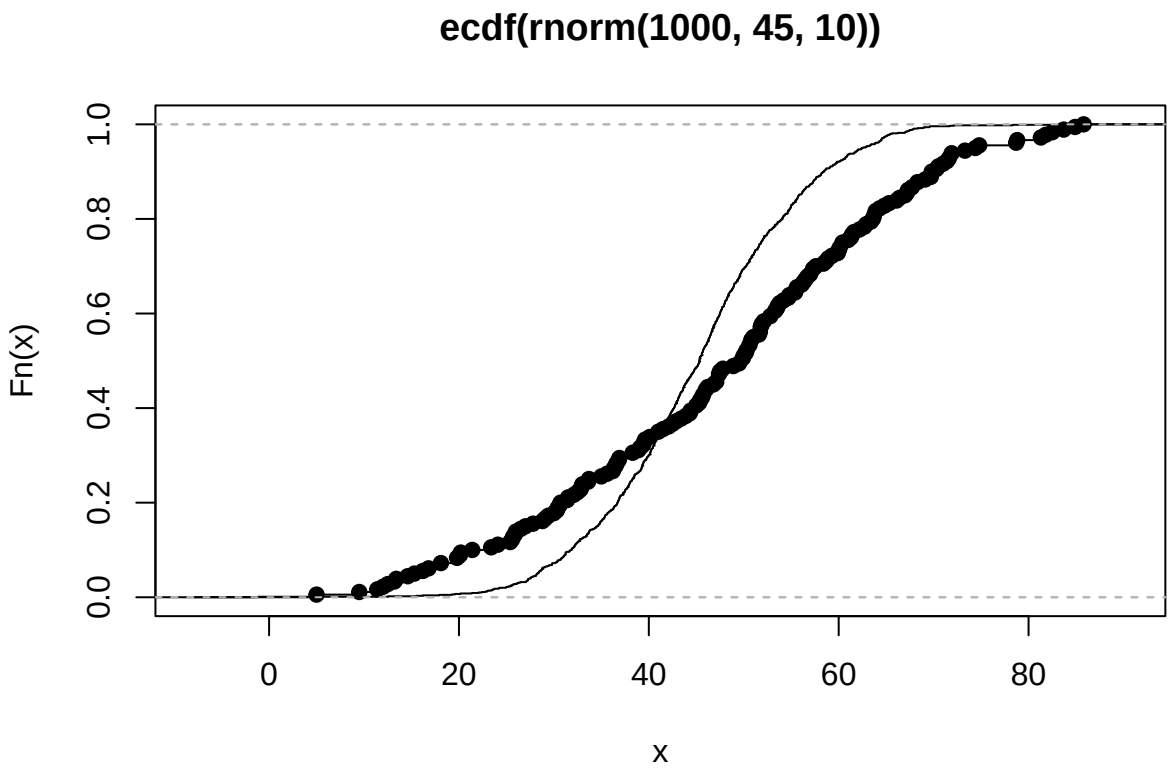

```
qqplot(qnorm(ppoints(2000)),BDH.new)
qqline(BDH.new)
```




```
plot(ecdf(BDH.new), do.points=FALSE)
```

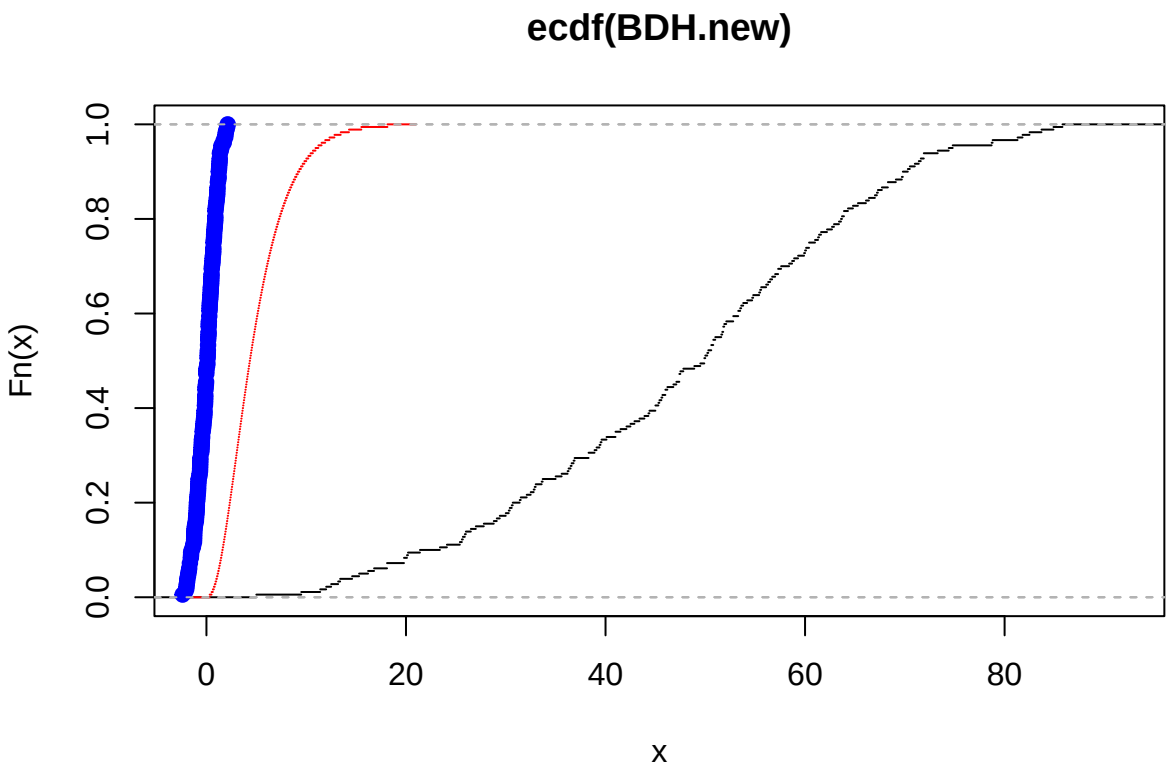



```
plot(ecdf(rnorm(1000, 45, 10)), do.points=FALSE)
lines(ecdf(BDH.new))
```



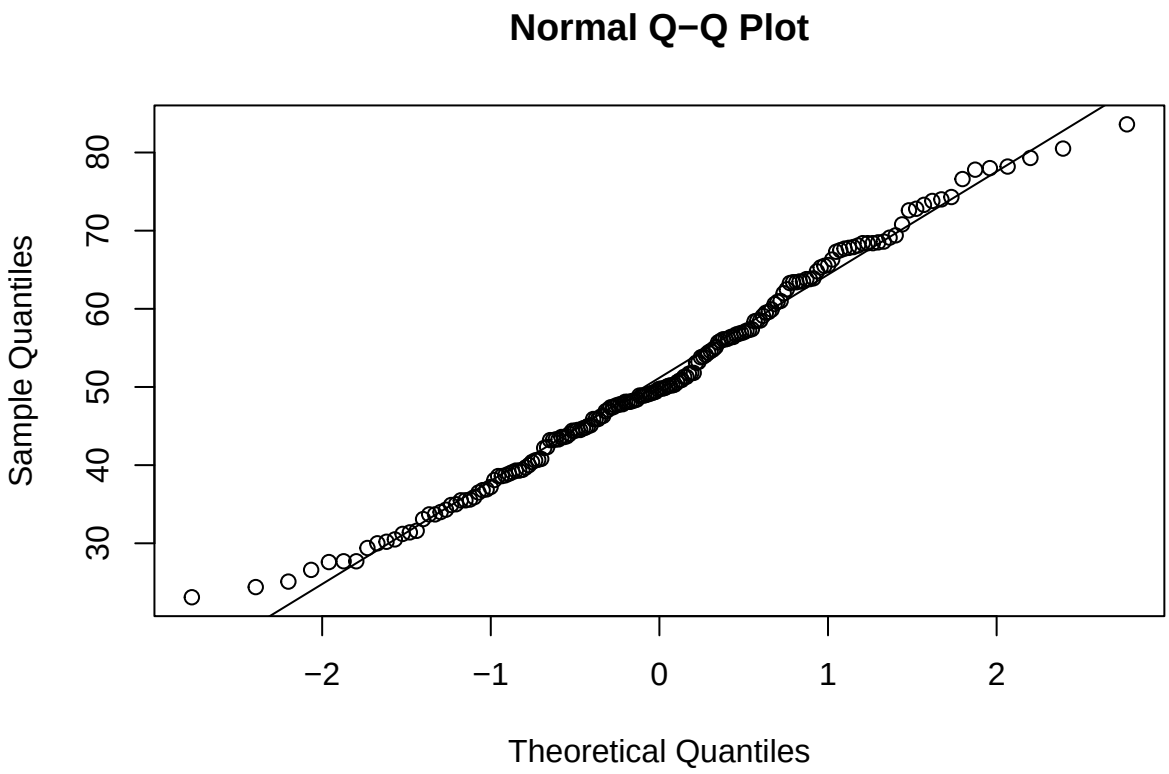
Exploration 1: qchisq Distribution for ECDF(BDH.new)

```
# ECDF of BDH.new
plot(ecdf(BDH.new), do.points=FALSE)

# Overlay ECDF of chi-squared distribution
p <- ppoints(length(BDH.new))  # Generate probabilities
df <- 5  # Example degrees of freedom
plot(ecdf(qchisq(p, df)), do.points=FALSE, col="red", add=TRUE)

# Add lines for BDH.new
lines(ecdf(scale(BDH.new)), col="blue")
```

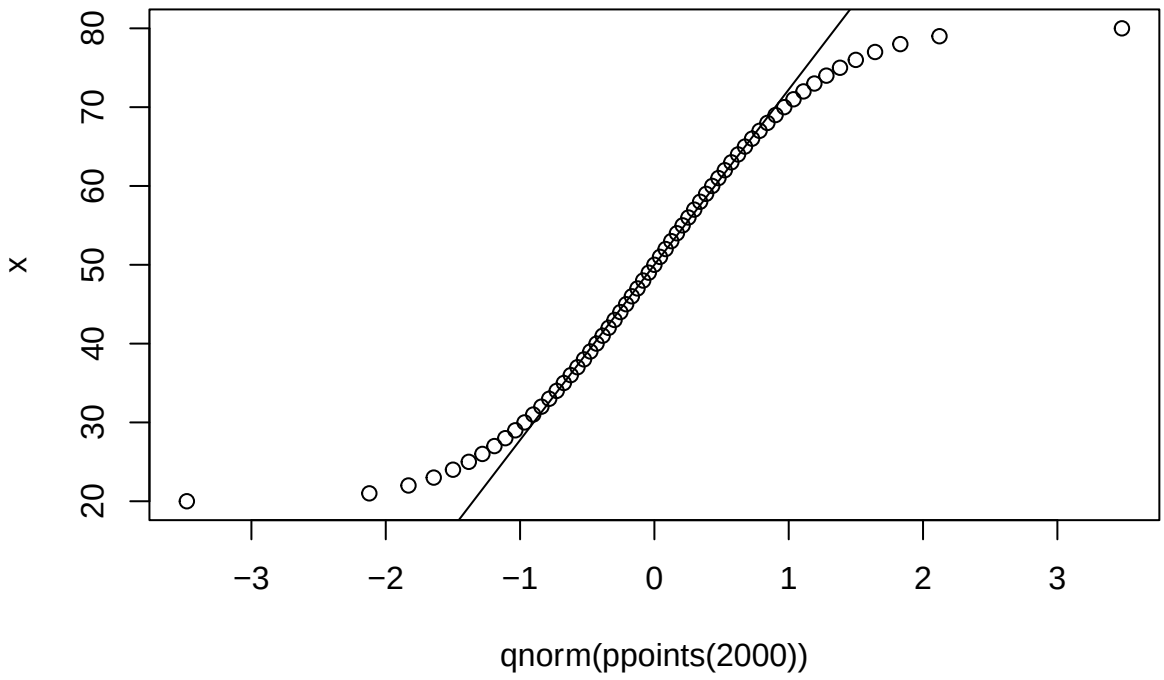


Exploration 2: qbeta Distribution for qqplot(ECO.new)

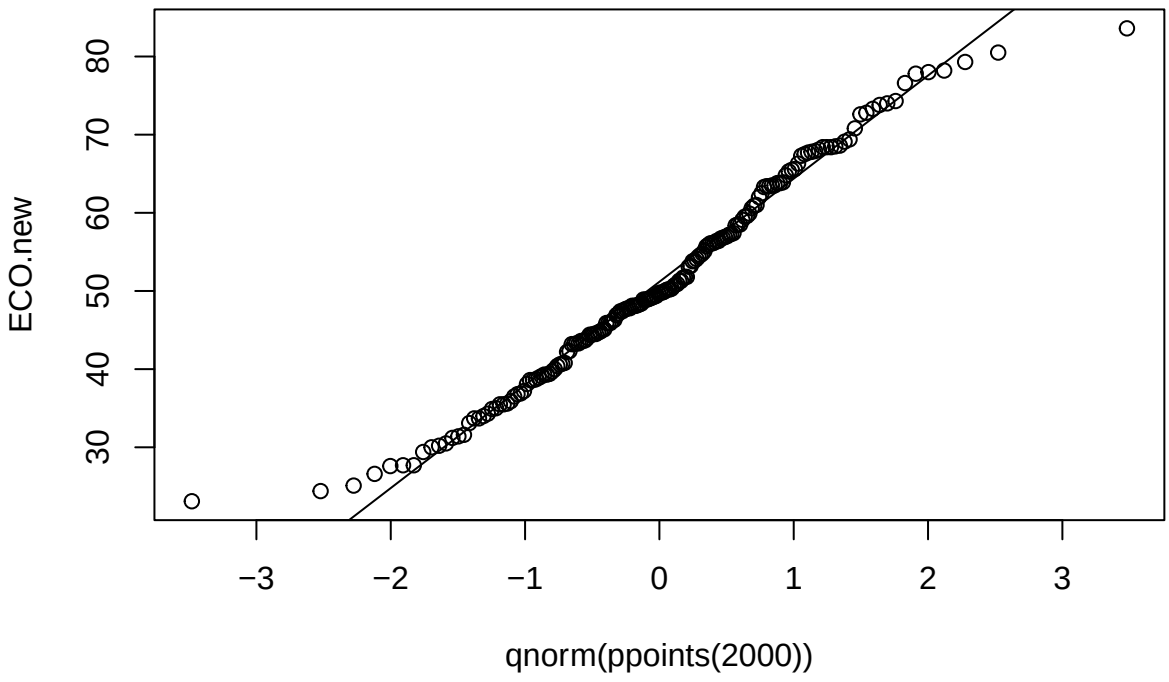
```
qqnorm(ECO.new); qqline(ECO.new)
```



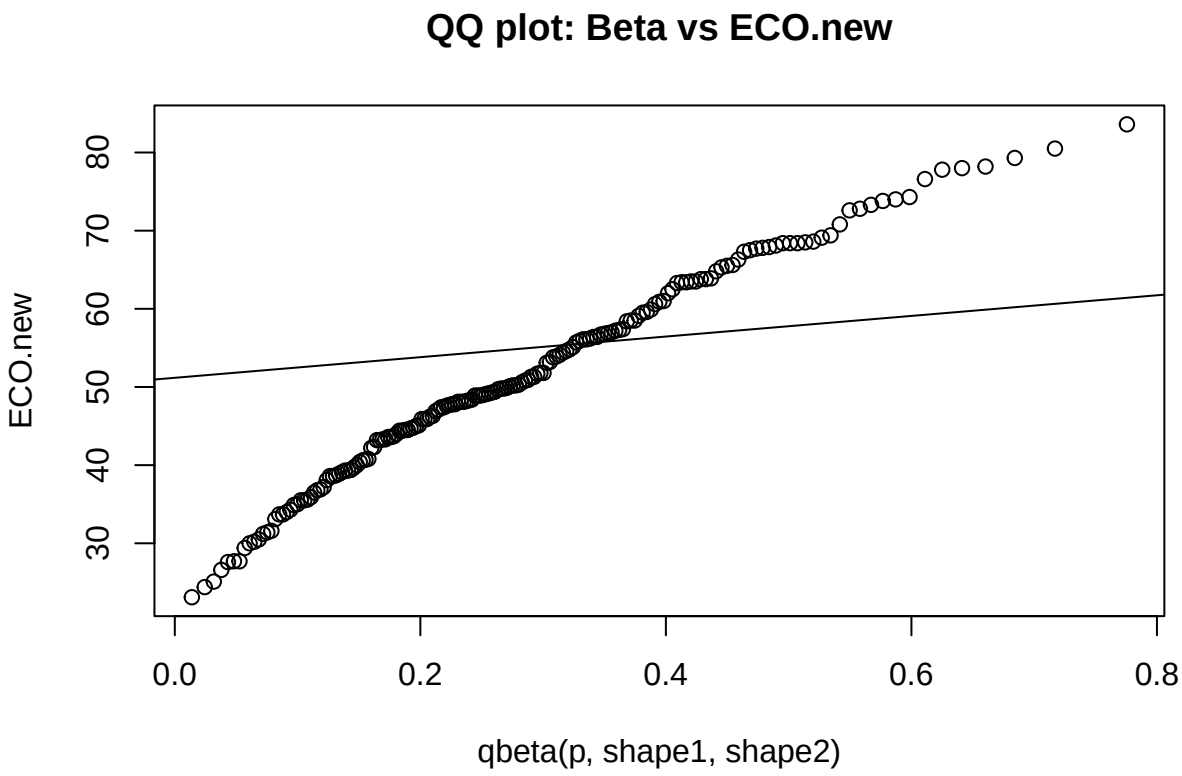

```
x <- seq(20., 80., 1.0)
qqplot(qnorm(ppoints(2000)), x)
qqline(x)
```




```
qqplot(qnorm(ppoints(2000)),ECO.new)
qqline(ECO.new)
```

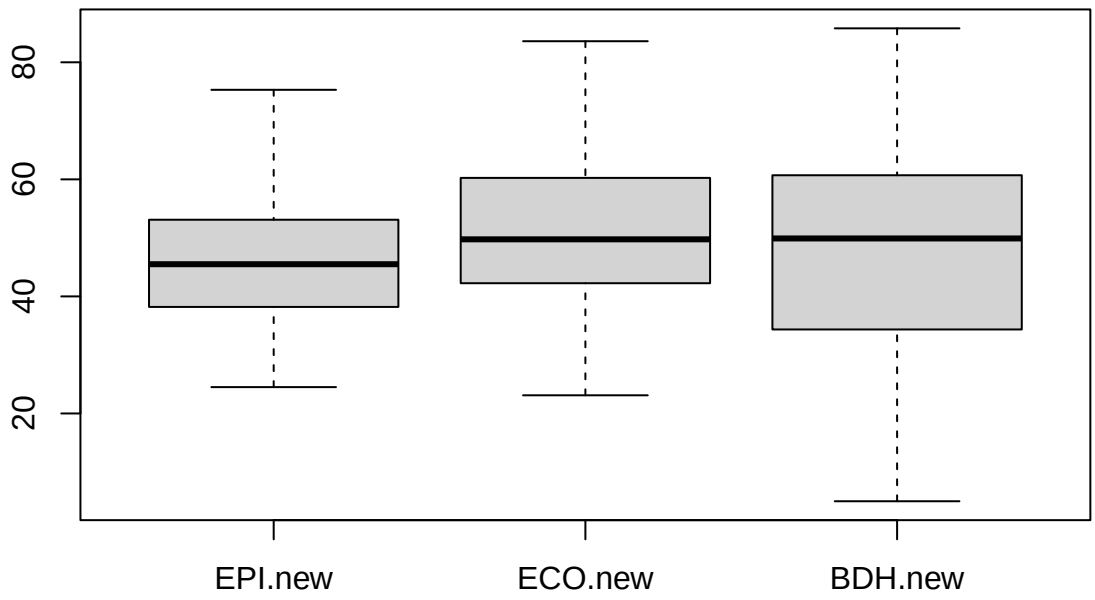



```r
# Now, generate a QQ plot comparing ECO.new to a Beta distribution
# Generate p points for comparison (same length as ECO.new)
p <- ppoints(length(ECO.new))

# Set the shape parameters for the Beta distribution (arbitrary, adjust as needed)
shape1 <- 2
shape2 <- 5

# QQ Plot: Beta quantiles vs ECO.new
qqplot(qbeta(p, shape1, shape2), ECO.new, main = "QQ plot: Beta vs ECO.new")
qqline(ECO.new)
```



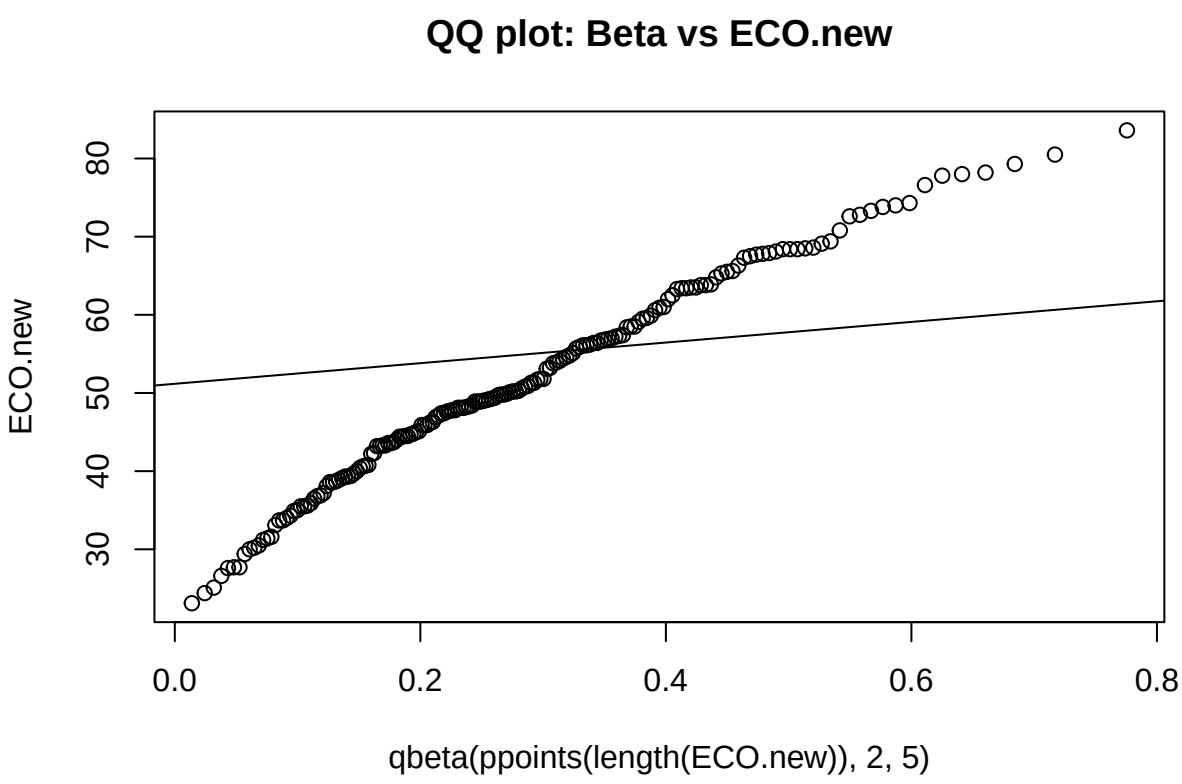
Boxplots Comparing 3 Variables

```
boxplot(EPI.new, ECO.new, BDH.new, names=c("EPI.new","ECO.new", "BDH.new"))
```

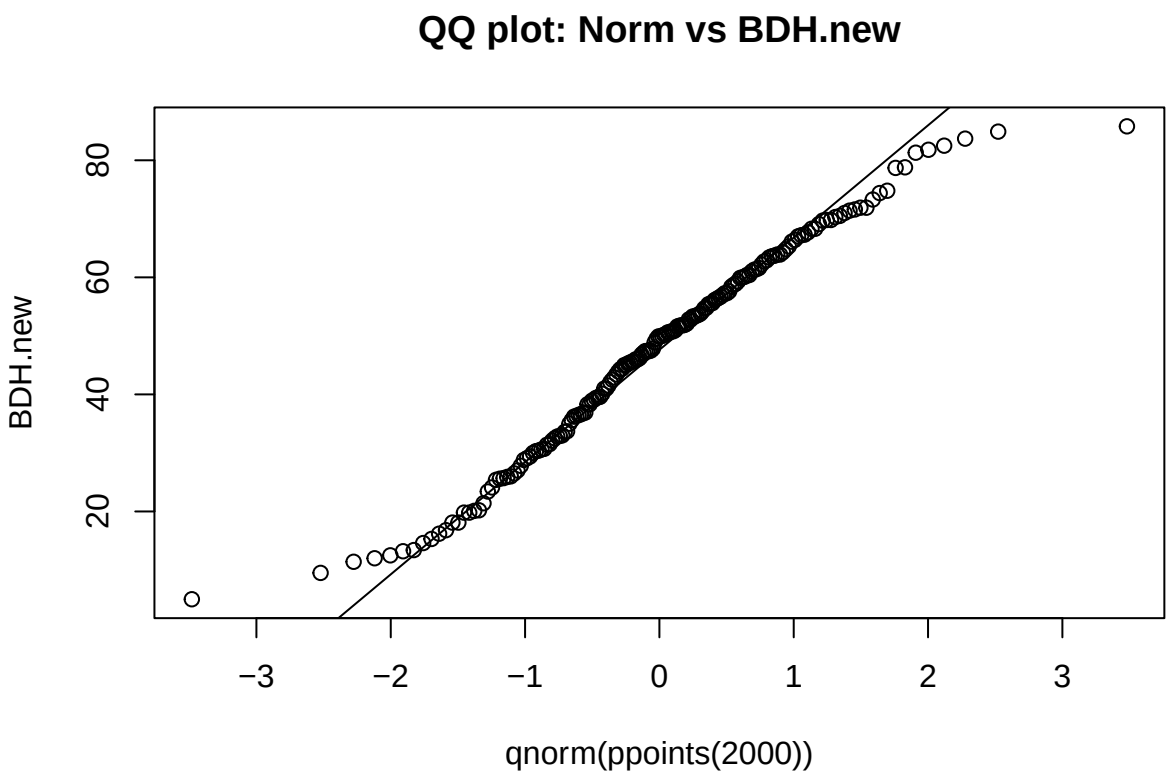


Q-Q plots for 3 Variables Compared to Known Distributions

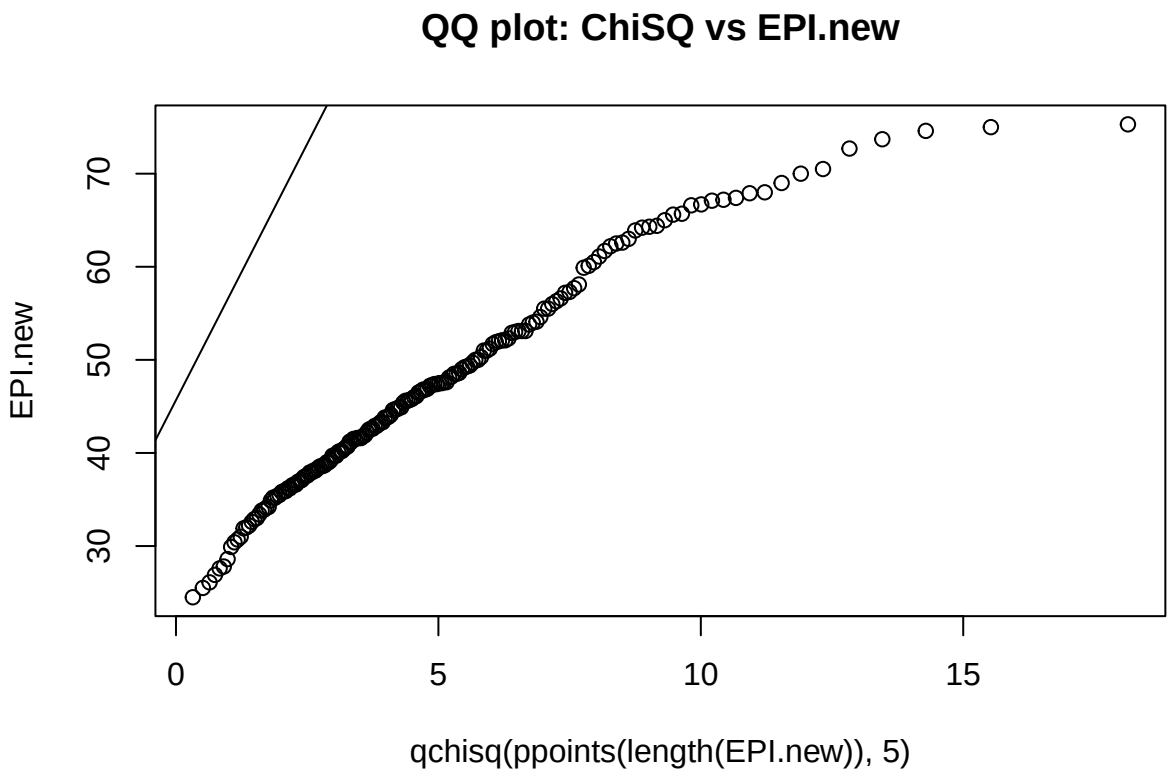
```
qqplot(qbeta(ppoints(length(ECO.new)), 2, 5), ECO.new, main = "QQ plot: Beta vs ECO.new")
qqline(ECO.new)
```




```
qqplot(qnorm(ppoints(2000)), BDH.new, main = "QQ plot: Norm vs BDH.new")
qqline(BDH.new)
```

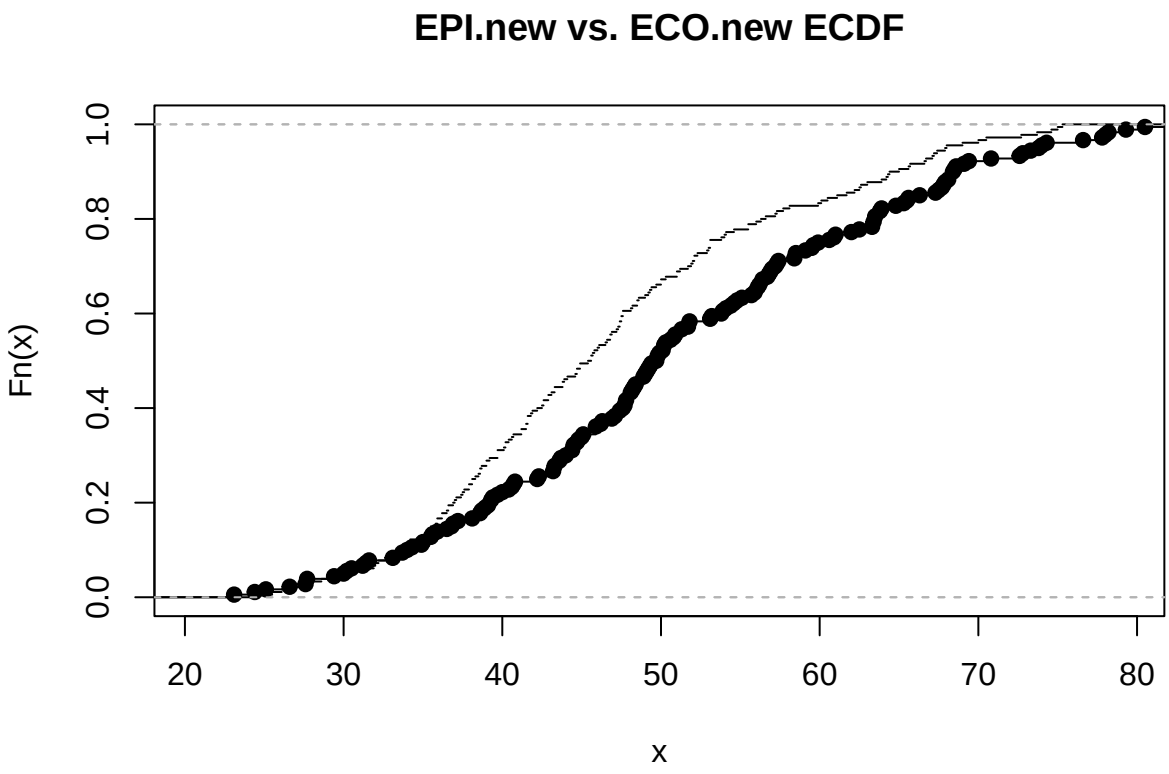



```
qqplot(qchisq(ppoints(length(EPI.new)), 5), EPI.new, main = "QQ plot: ChiSQ vs EPI.new")
qqline(EPI.new)
```

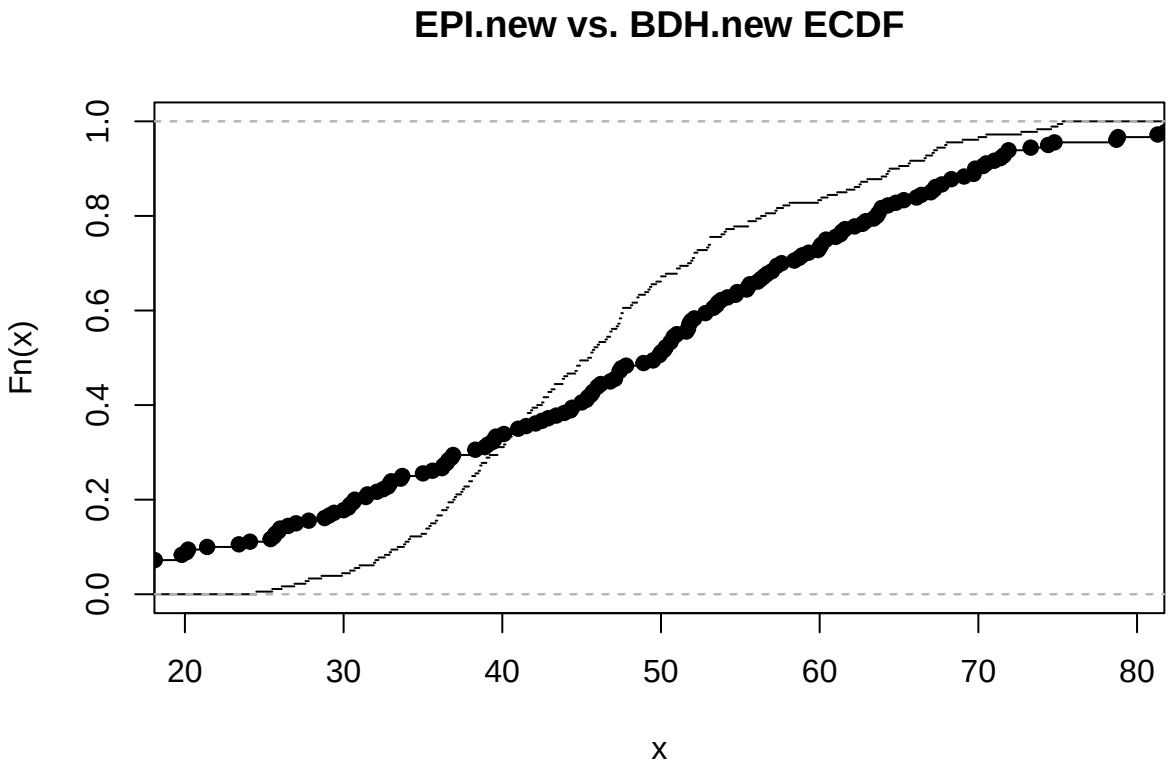


ECDF Plots for 3 Variables Compared to Each Other

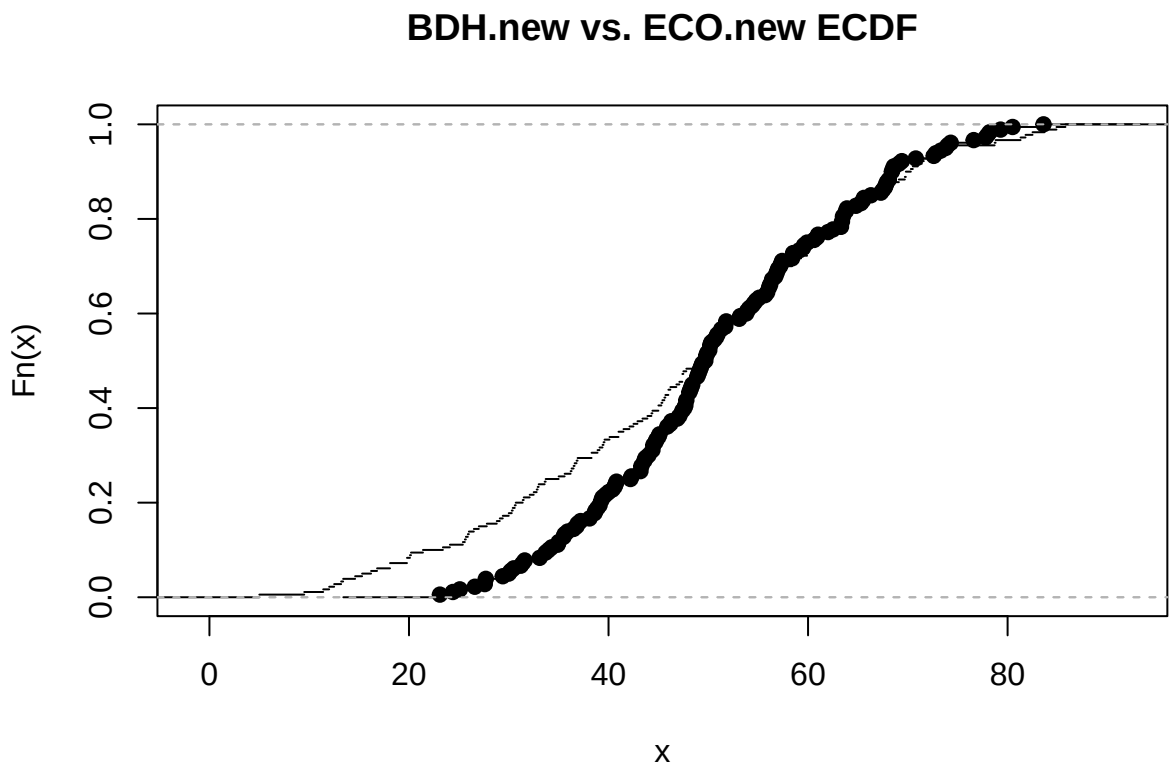
```
plot(ecdf(EPI.new), do.points=FALSE, main="EPI.new vs. ECO.new ECDF")
lines(ecdf(ECO.new))
```




```
plot(ecdf(EPI.new), do.points=FALSE, main="EPI.new vs. BDH.new ECDF")
lines(ecdf(BDH.new))
```



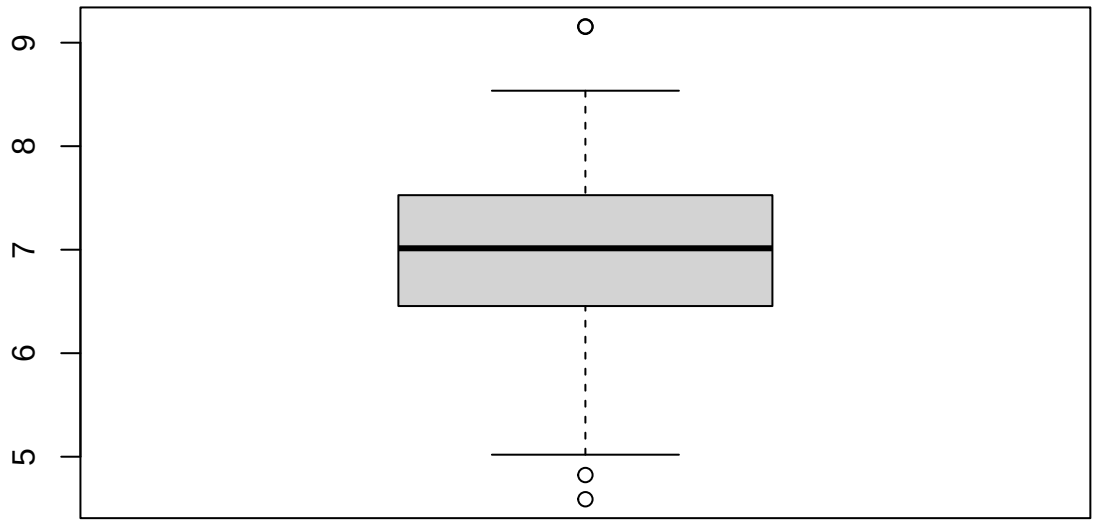

```
plot(ecdf(BDH.new), do.points=FALSE, main="BDH.new vs. ECO.new ECDF")
lines(ecdf(ECO.new))
```



Summary Stats and Select Plots from 3 Linear Models

setup

```
## drop country populations that don't exist in epi results
populations <- population_data[-which(!population_data$Country %in% epi_results$country),]

## sort populations by country name
populations <- populations[order(populations$Country),]

## drop country results that don't exist in populations
epi_results.sub <- epi_results[-which(!epi_results$country %in% populations$Country),]

## sort results by country name
epi_results.sub <- epi_results.sub[order(epi_results.sub$country),]

## only keep relevant columns
epi_results.sub <- epi_results.sub[,c("country","EPI.old","EPI.new", "ECO.new", "BDH.new")]

## convert to mnumeric
epi_results.sub$population <- as.numeric(populations$Population)

## compute population log
epi_results.sub$population_log <- log10(epi_results.sub$population)

boxplot(epi_results.sub$population_log)
```

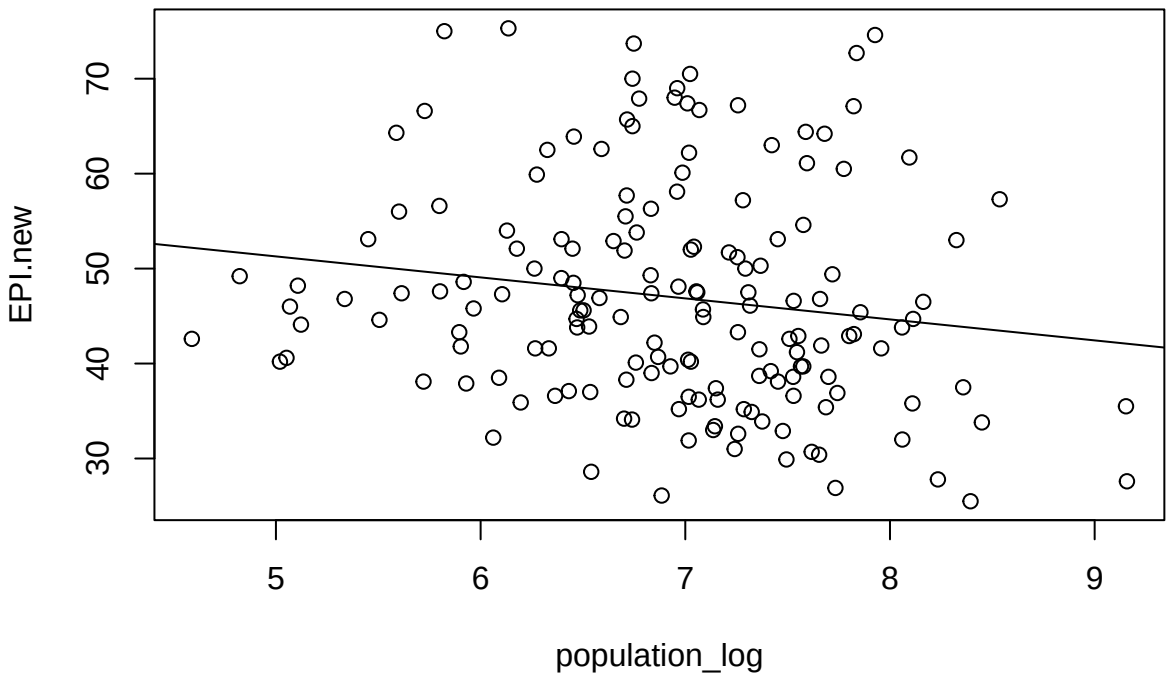



```
attach(epi_results.sub)
```

```
## The following objects are masked from epi_results:
##
##     BDH.new, country, ECO.new, EPI.new, EPI.old
```

Linear Model 1: EPI.new

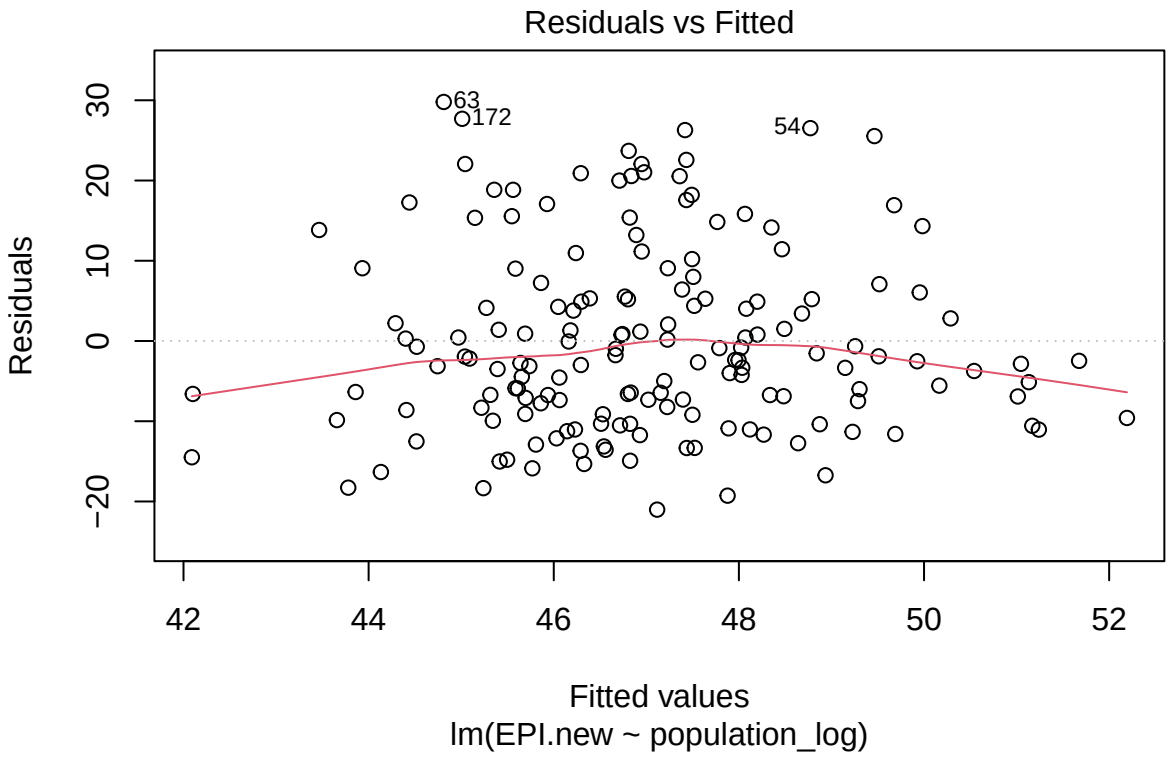
```
## created linear model of EPI.new = a(population_log) + b
lin.mod.epinew <- lm(EPI.new~population_log,epi_results.sub)

plot(EPI.new~population_log)
abline(lin.mod.epinew)
```

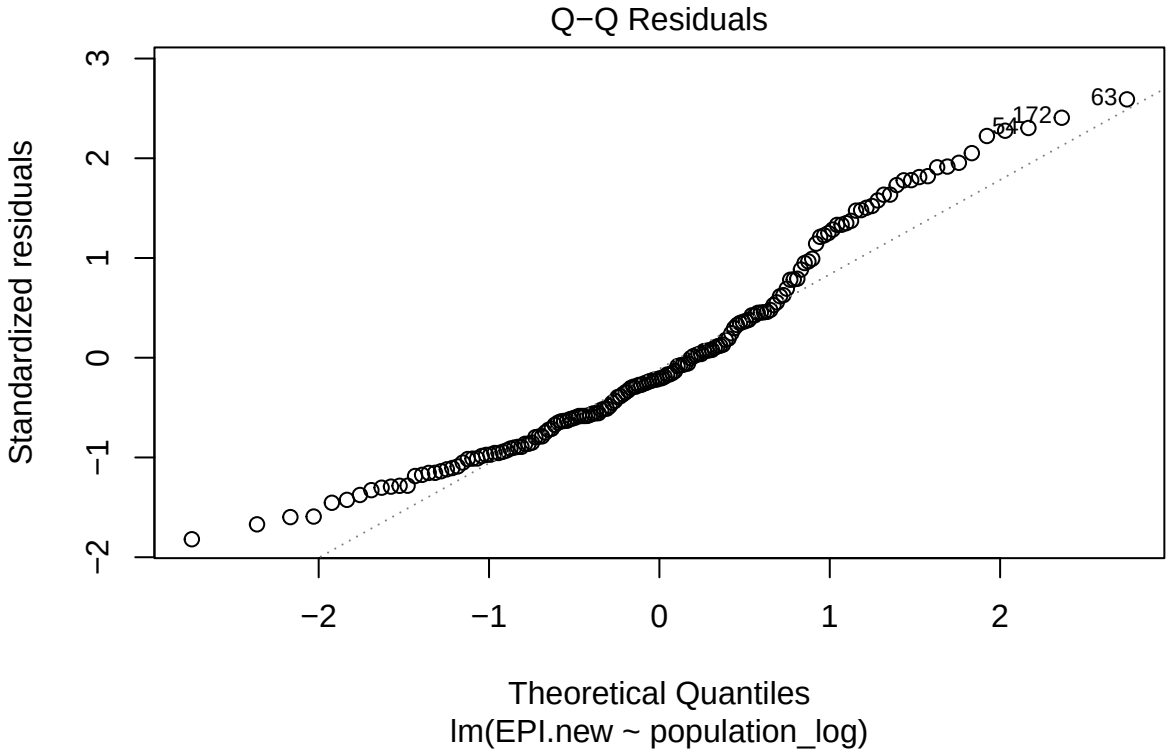


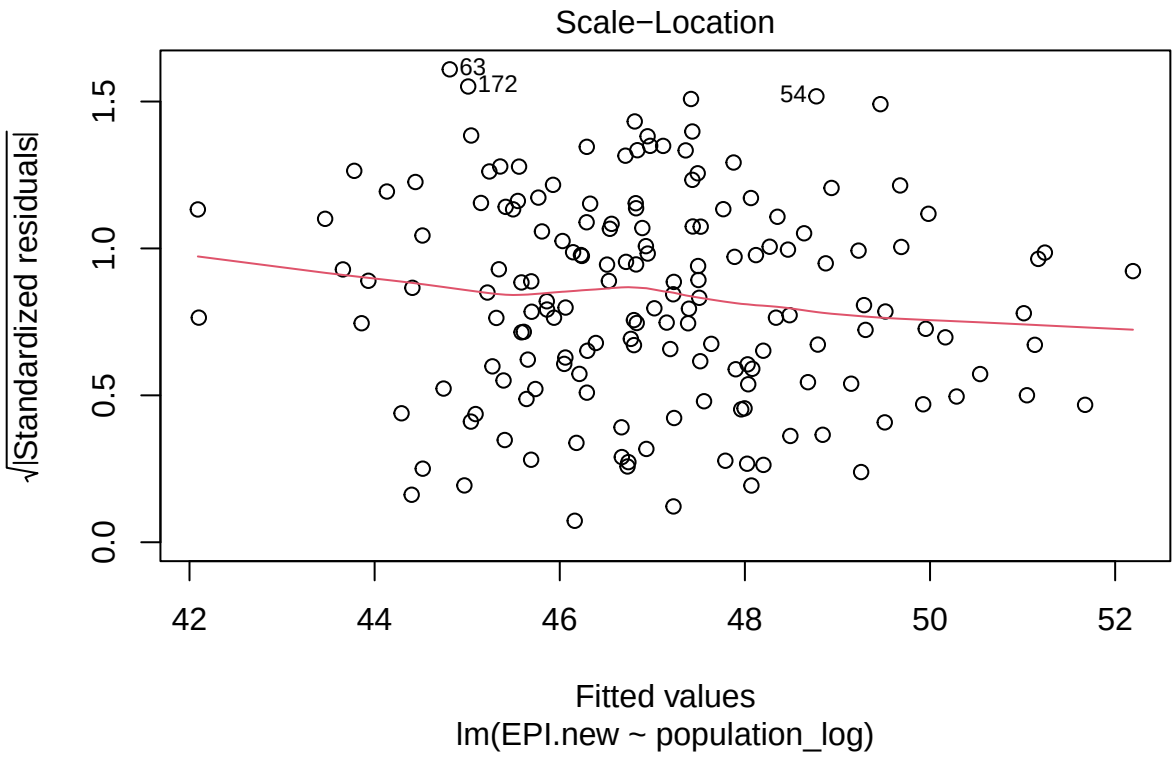

```
summary(lin.mod.epinew)
```

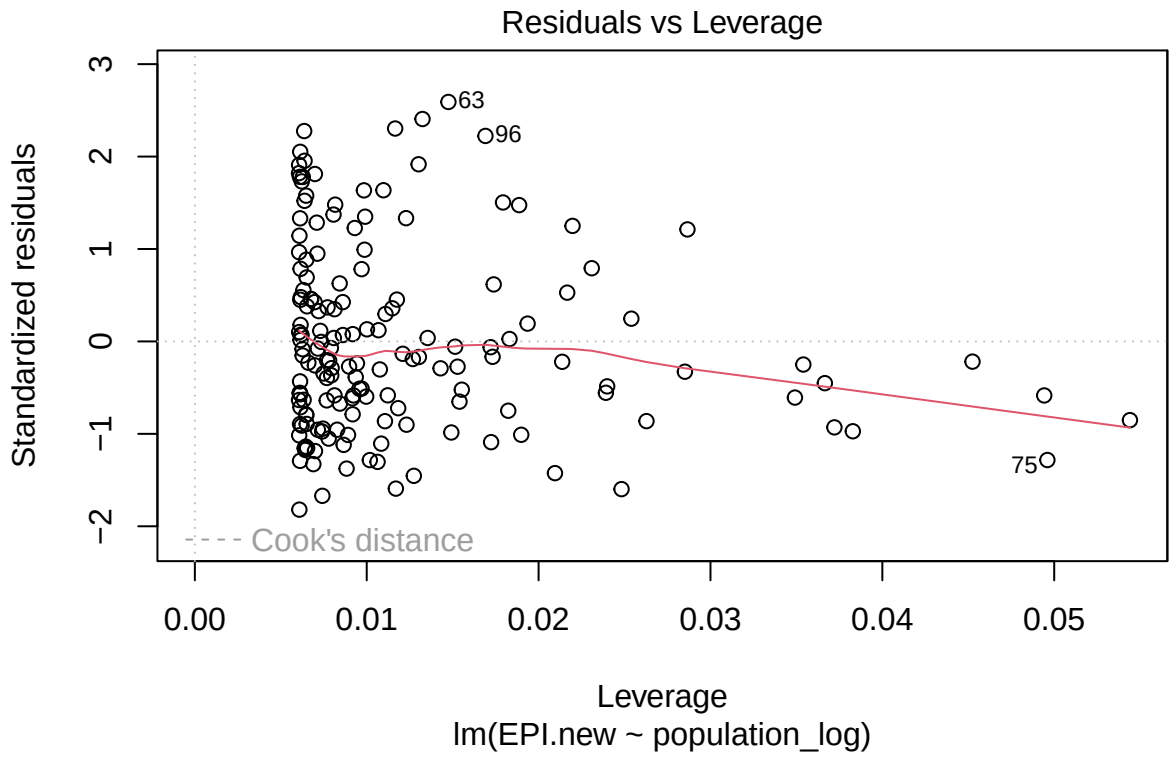
```
##
## Call:
## lm(formula = EPI.new ~ population_log, data = epi_results.sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.017  -8.608  -2.396   6.046  29.789
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      62.340      7.587   8.216 6.17e-14 ***
## population_log   -2.211      1.087  -2.035   0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.58 on 163 degrees of freedom
## Multiple R-squared:  0.02478,    Adjusted R-squared:  0.01879
## F-statistic: 4.141 on 1 and 163 DF,  p-value: 0.04348
```

```
plot(lin.mod.epinew)
```


Residuals vs Fitted
Residuals
30
20
10
0
−20
63
172
54
42
44
46
48
50
52
Fitted values
lm(EPI.new ~ population_log)

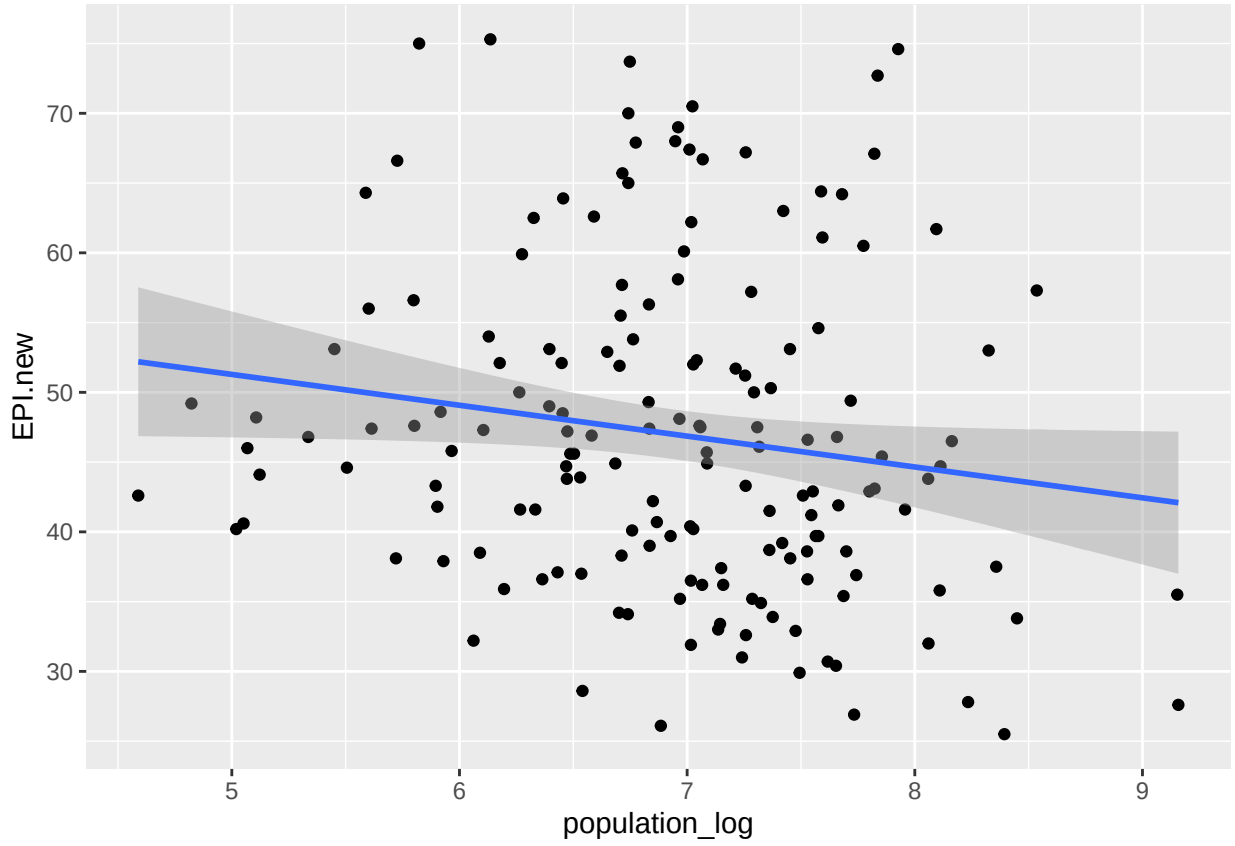

Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(EPI.new ~ population_log)
63
172
54


Scale−Location
63
172
54
√|Standardized residuals|
0.0
0.5
1.0
1.5
42
44
46
48
50
52
Fitted values
lm(EPI.new ~ population_log)

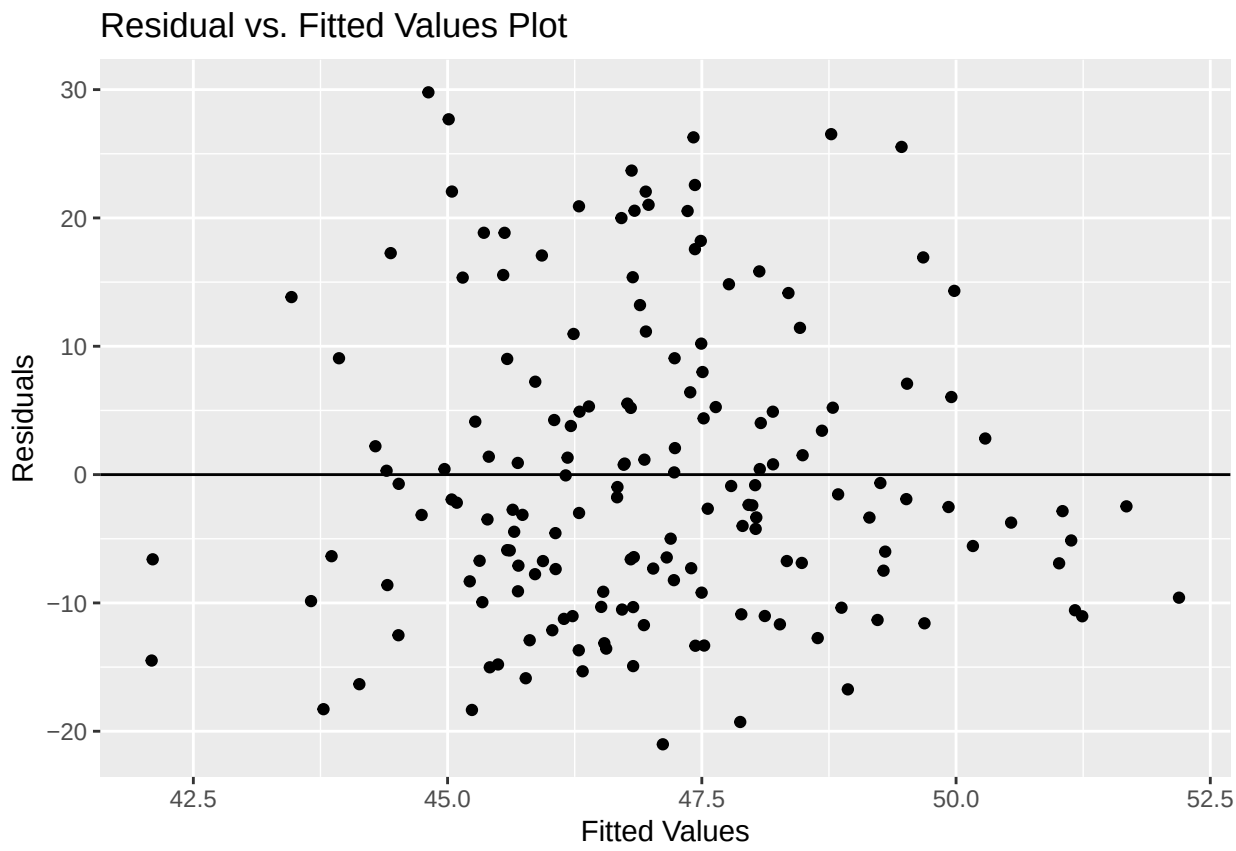



```
ggplot(epi_results.sub, aes(x = population_log, y = EPI.new)) +
  geom_point() +
  stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

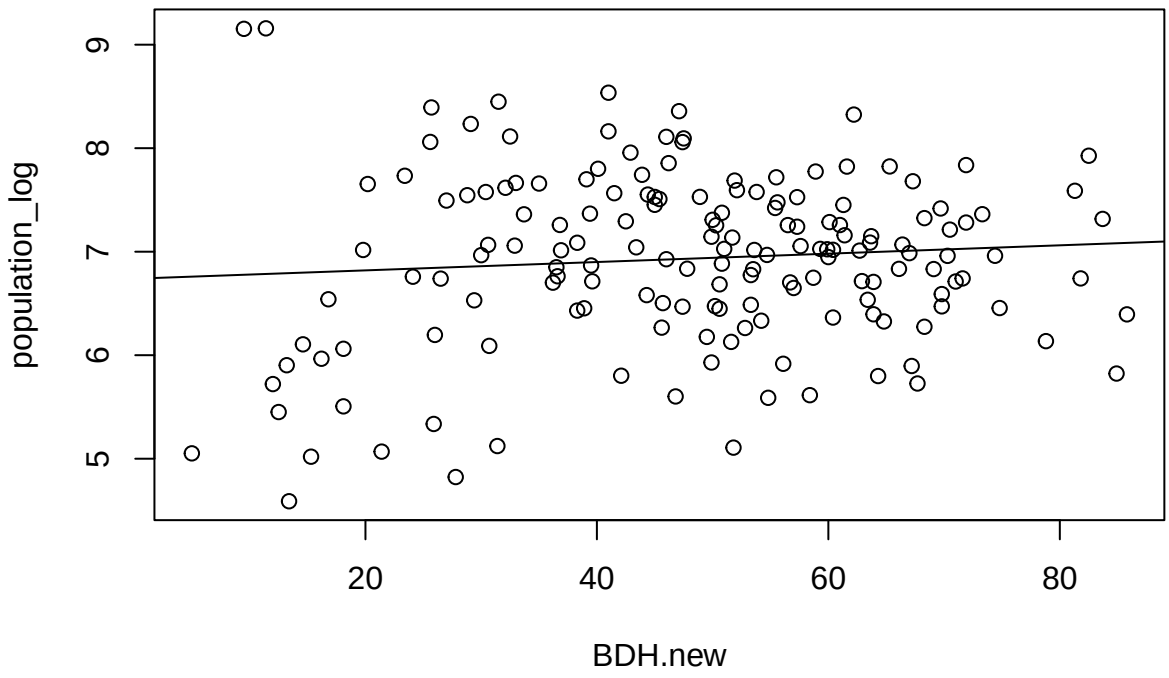



```
ggplot(lin.mod.epinew, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(title='Residual vs. Fitted Values Plot', x='Fitted Values', y='Residuals')
```



Linear Model 2:BDH.new

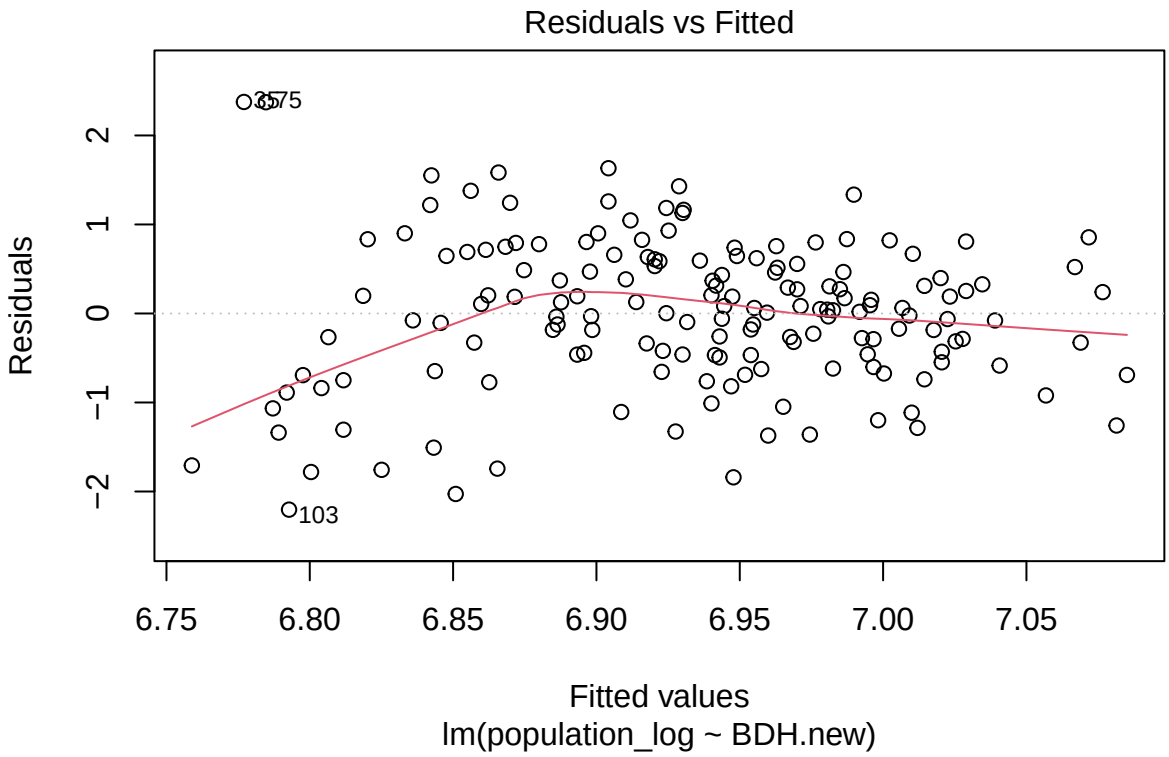
```
lin.mod.pop <- lm(population_log~BDH.new,epi_results.sub)
plot(population_log~BDH.new)
abline(lin.mod.pop)
```

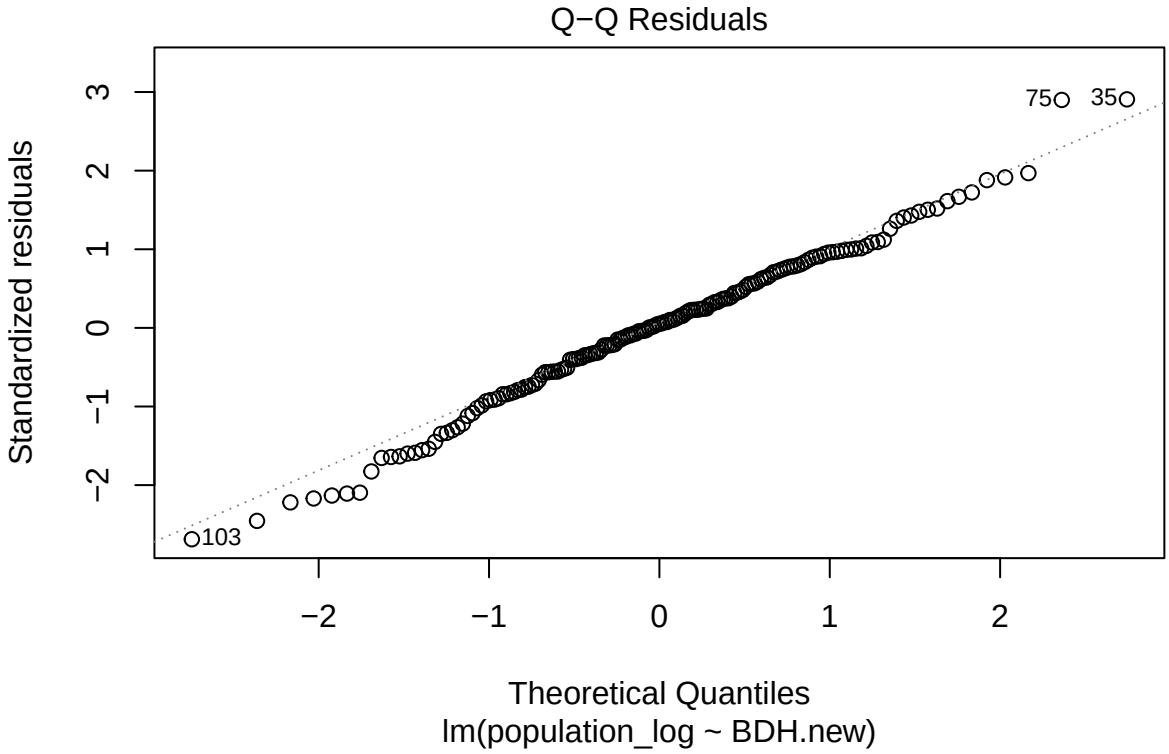


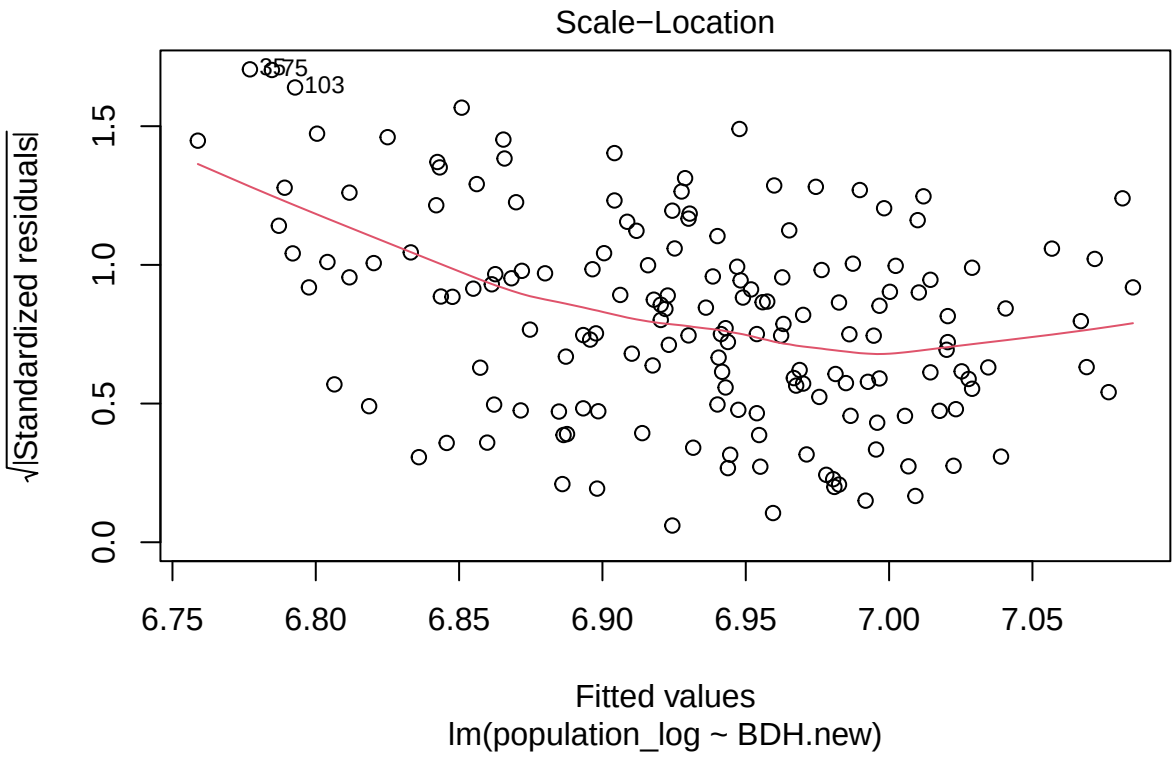

```
summary(lin.mod.pop)
```

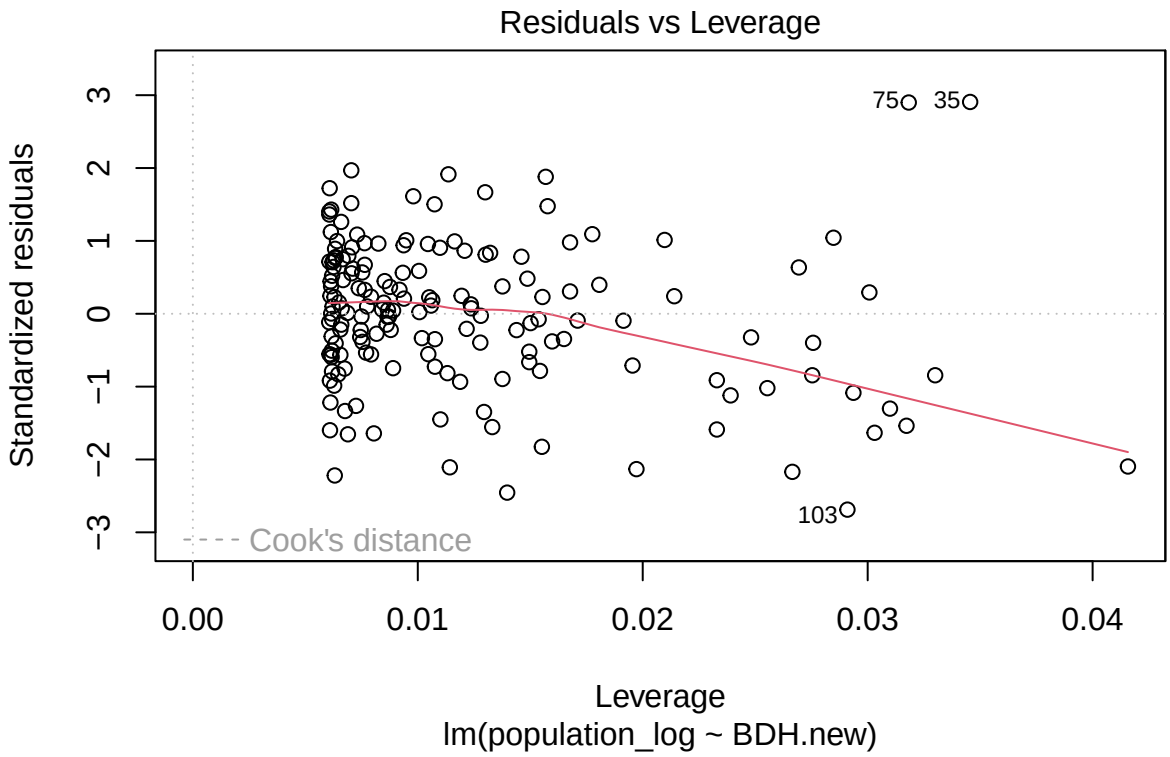
```
##
## Call:
## lm(formula = population_log ~ BDH.new, data = epi_results.sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20364 -0.46726  0.04282  0.58676  2.37605
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.738671   0.186513  36.130   <2e-16 ***
## BDH.new     0.004037   0.003630   1.112    0.268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8319 on 163 degrees of freedom
## Multiple R-squared:  0.007532,   Adjusted R-squared:  0.001443
## F-statistic: 1.237 on 1 and 163 DF,  p-value: 0.2677
```

```
plot(lin.mod.pop)
```


Residuals vs Fitted
Residuals
Fitted values
lm(population_log ~ BDH.new)
6.75
6.80
6.85
6.90
6.95
7.00
7.05
−2
−1
0
1
2
375
103

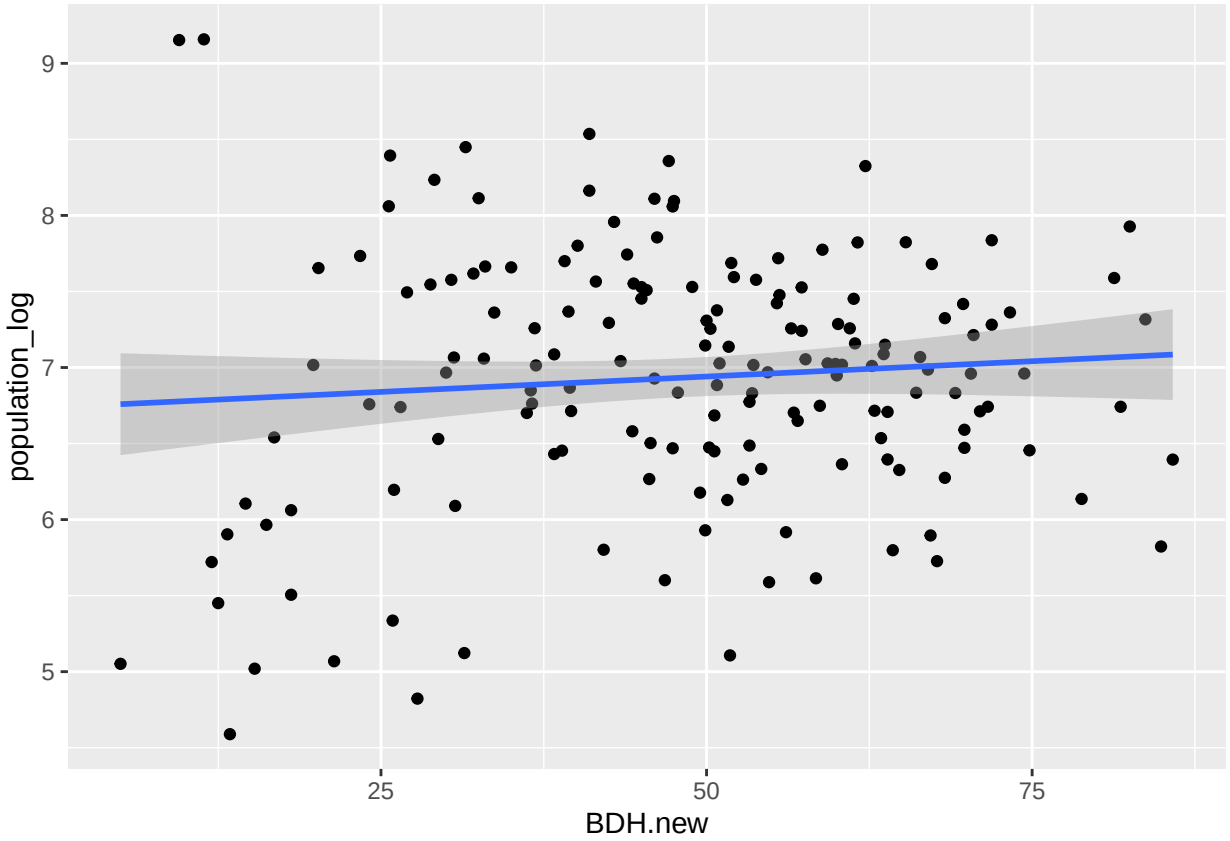

Q−Q Residuals
Standardized residuals
3
2
1
0
−1
−2
75
35
103
−2
−1
0
1
2
Theoretical Quantiles
lm(population_log ~ BDH.new)


Scale−Location
35
75
103
√|Standardized residuals|
0.0
0.5
1.0
1.5
6.75
6.80
6.85
6.90
6.95
7.00
7.05
Fitted values
lm(population_log ~ BDH.new)

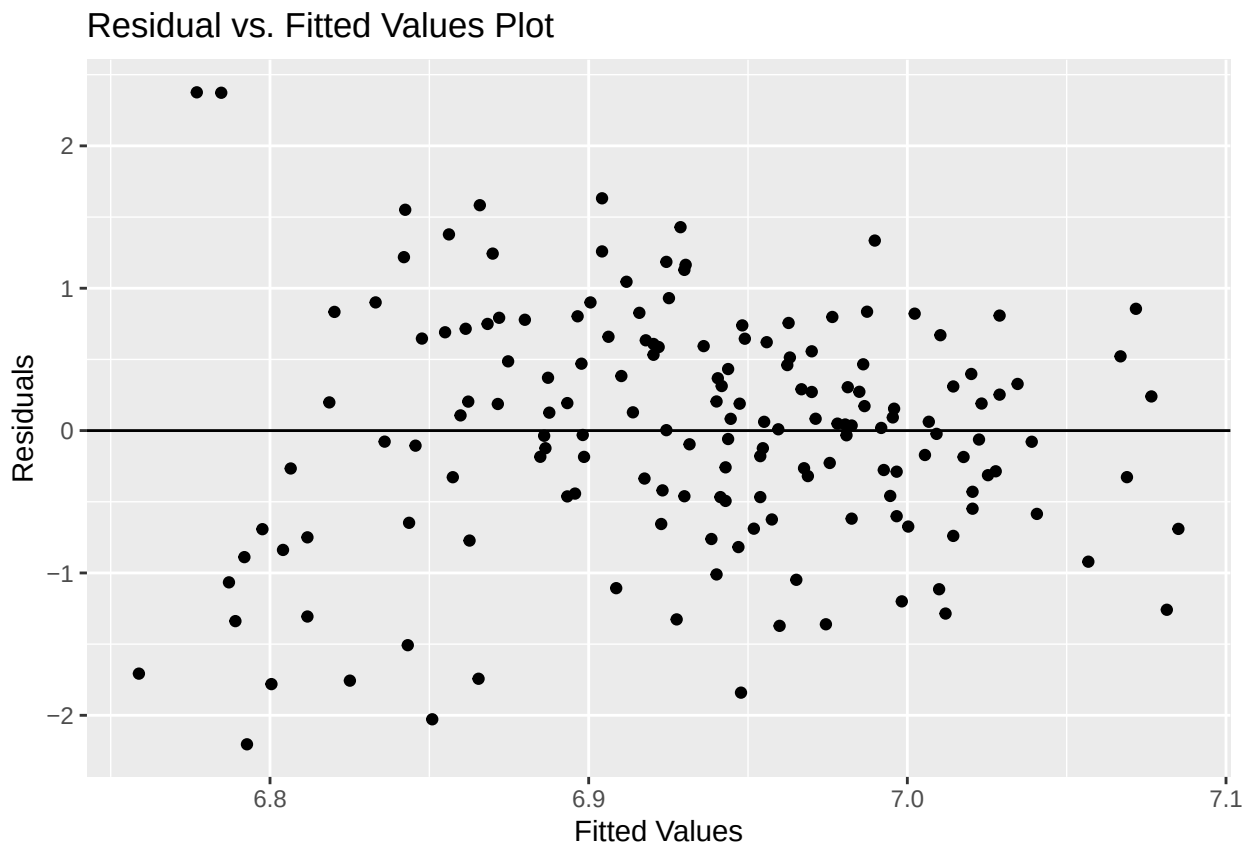



```
ggplot(epi_results.sub, aes(x = BDH.new, y = population_log)) +
  geom_point() +
  stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

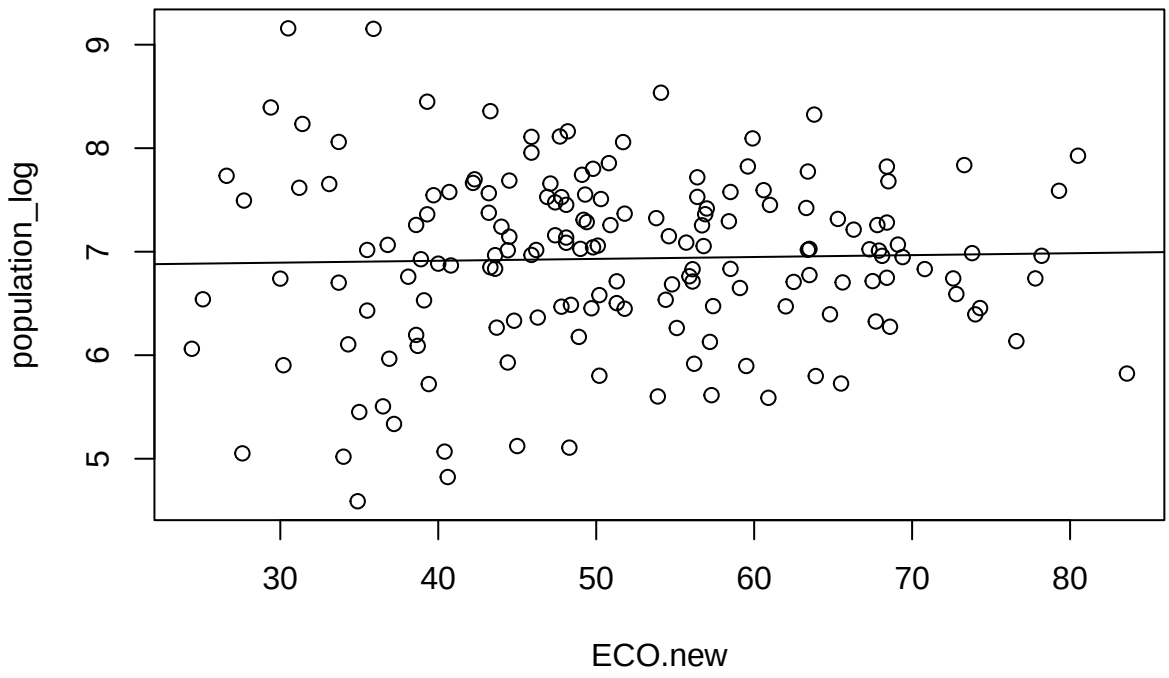



```
ggplot(lin.mod.pop, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(title='Residual vs. Fitted Values Plot', x='Fitted Values', y='Residuals')
```



Linear Model 3: ECO.new

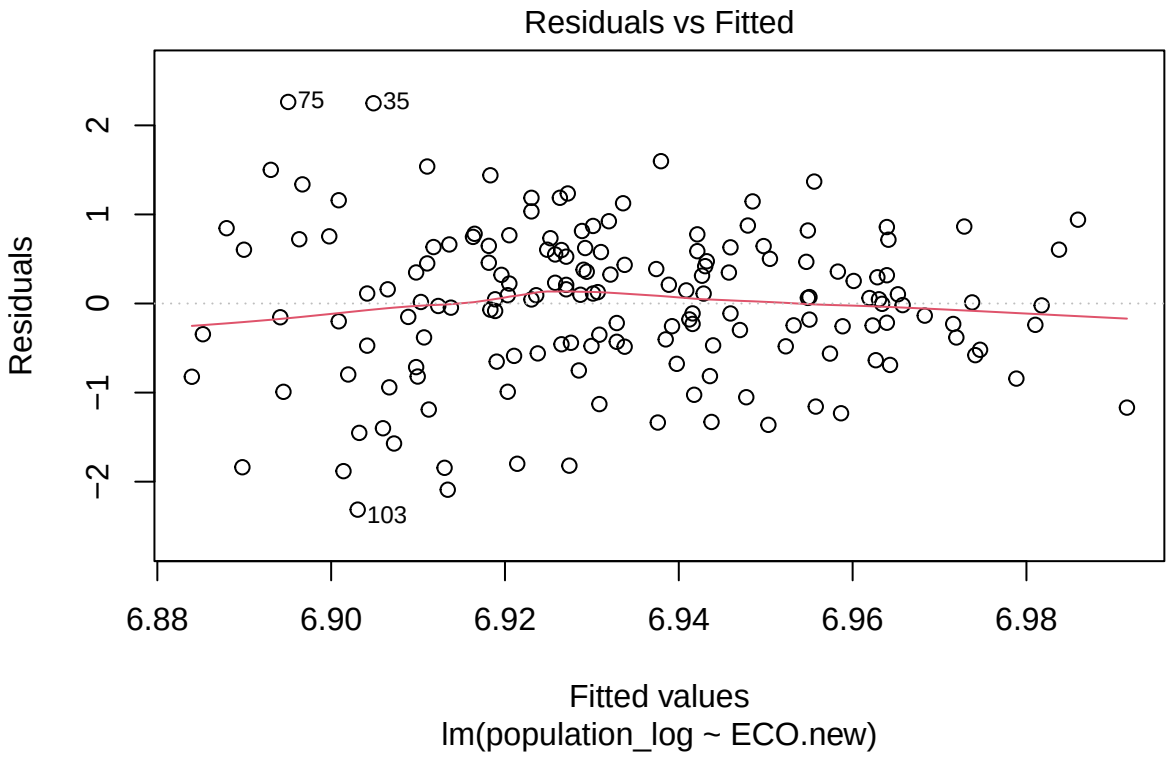
```
lin.mod.pop <- lm(population_log~ECO.new,epi_results.sub)
plot(population_log~ECO.new)
abline(lin.mod.pop)
```

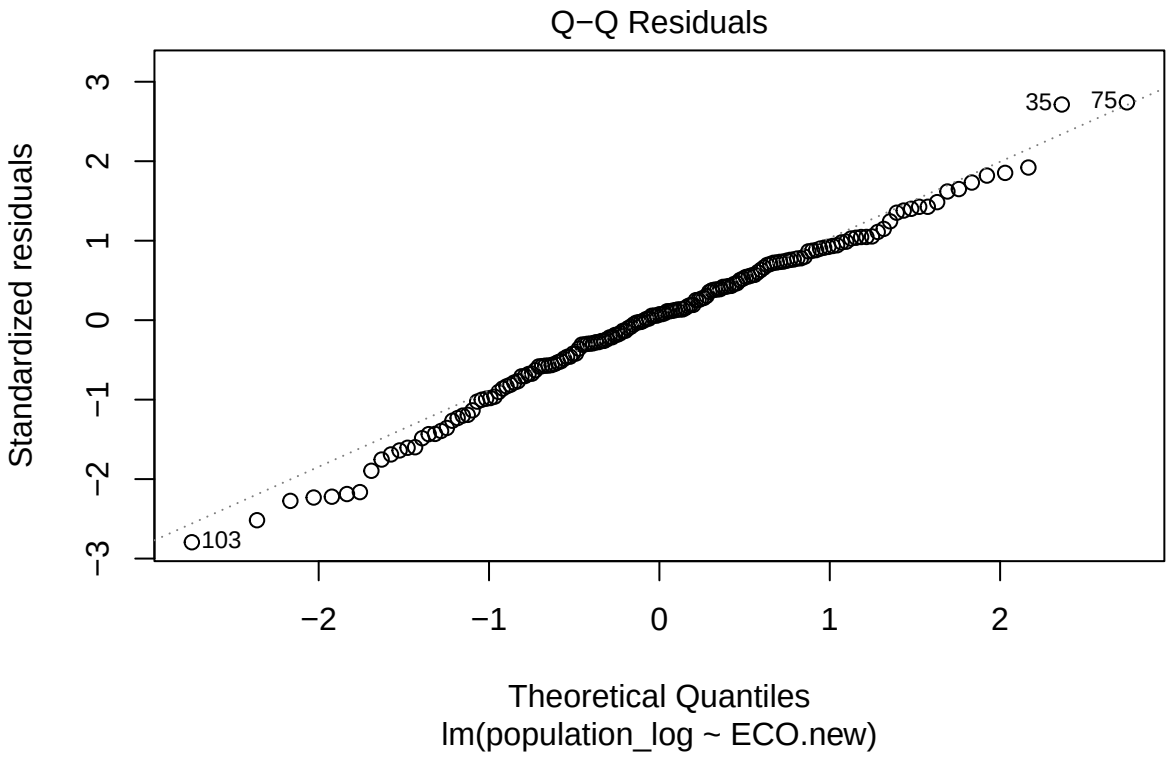


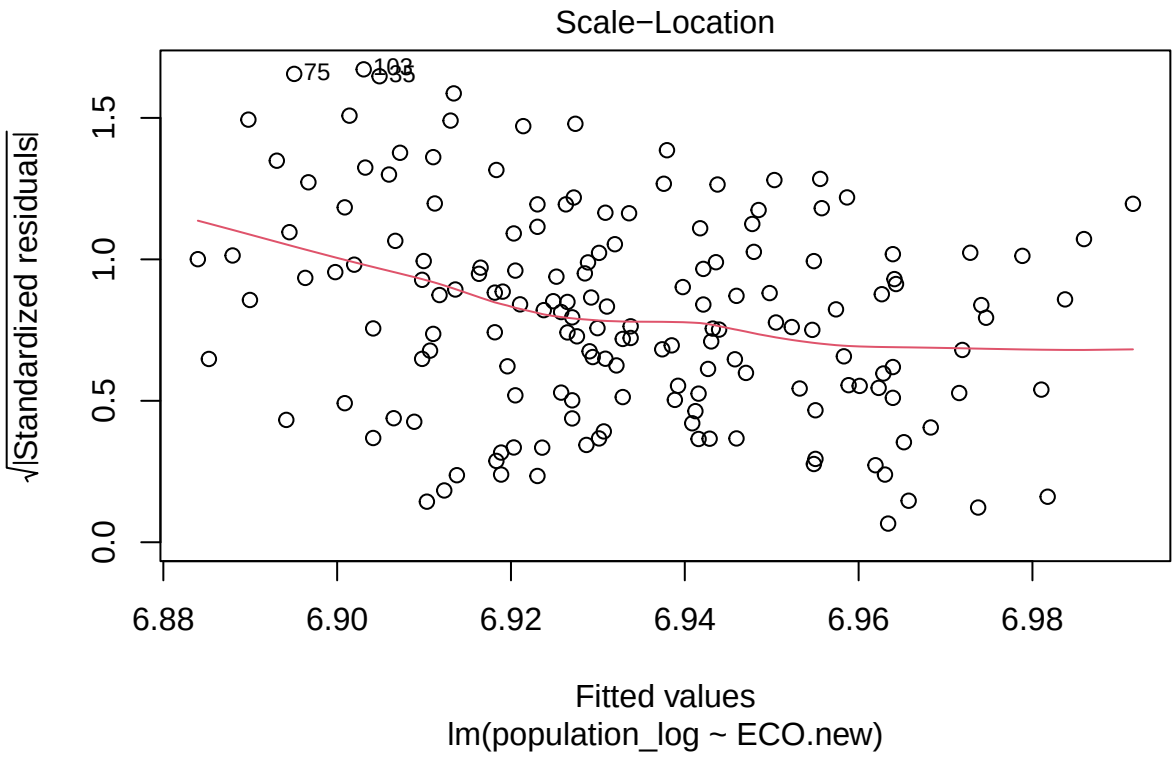

```
summary(lin.mod.pop)
```

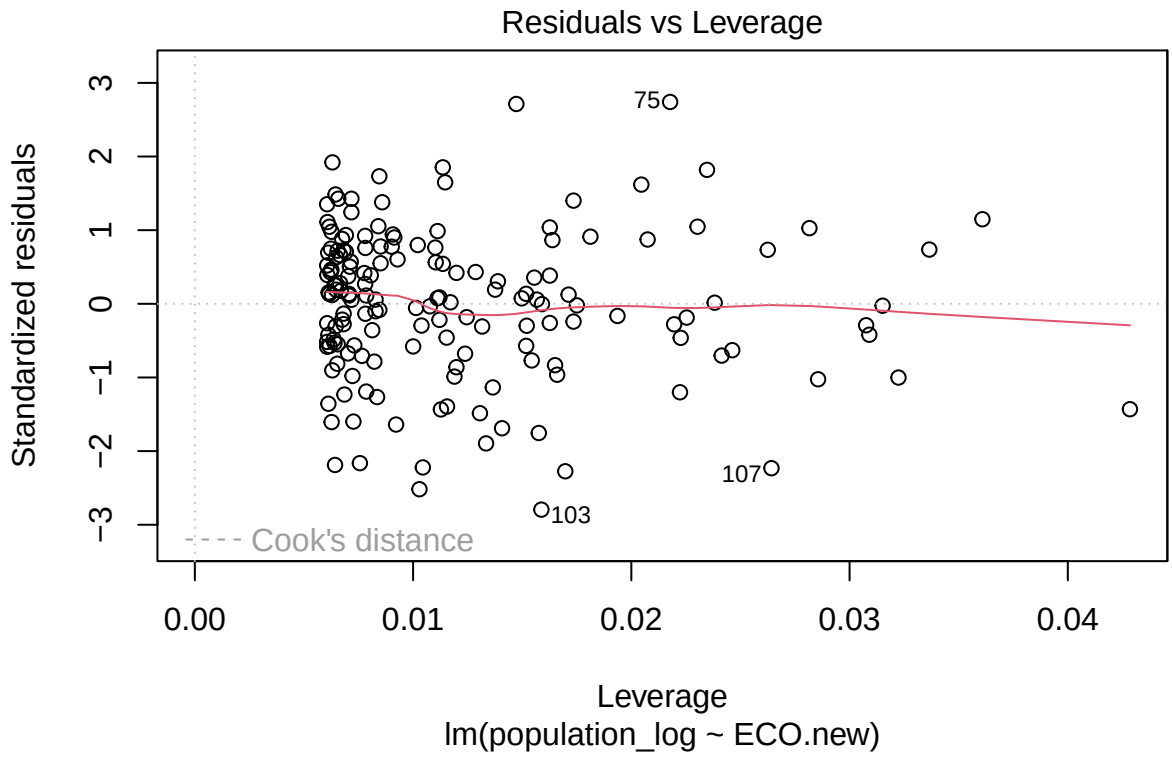
```
##
## Call:
## lm(formula = population_log ~ ECO.new, data = epi_results.sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.31392 -0.47667  0.06137  0.60029  2.26272
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.839624   0.264799  25.830   <2e-16 ***
## ECO.new     0.001817   0.004986   0.365    0.716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8347 on 163 degrees of freedom
## Multiple R-squared:  0.0008147,  Adjusted R-squared:  -0.005315
## F-statistic: 0.1329 on 1 and 163 DF,  p-value: 0.7159
```

```
plot(lin.mod.pop)
```


Residuals vs Fitted
75
35
103
Residuals
2
1
0
−1
−2
6.88
6.90
6.92
6.94
6.96
6.98
Fitted values
lm(population_log ~ ECO.new)

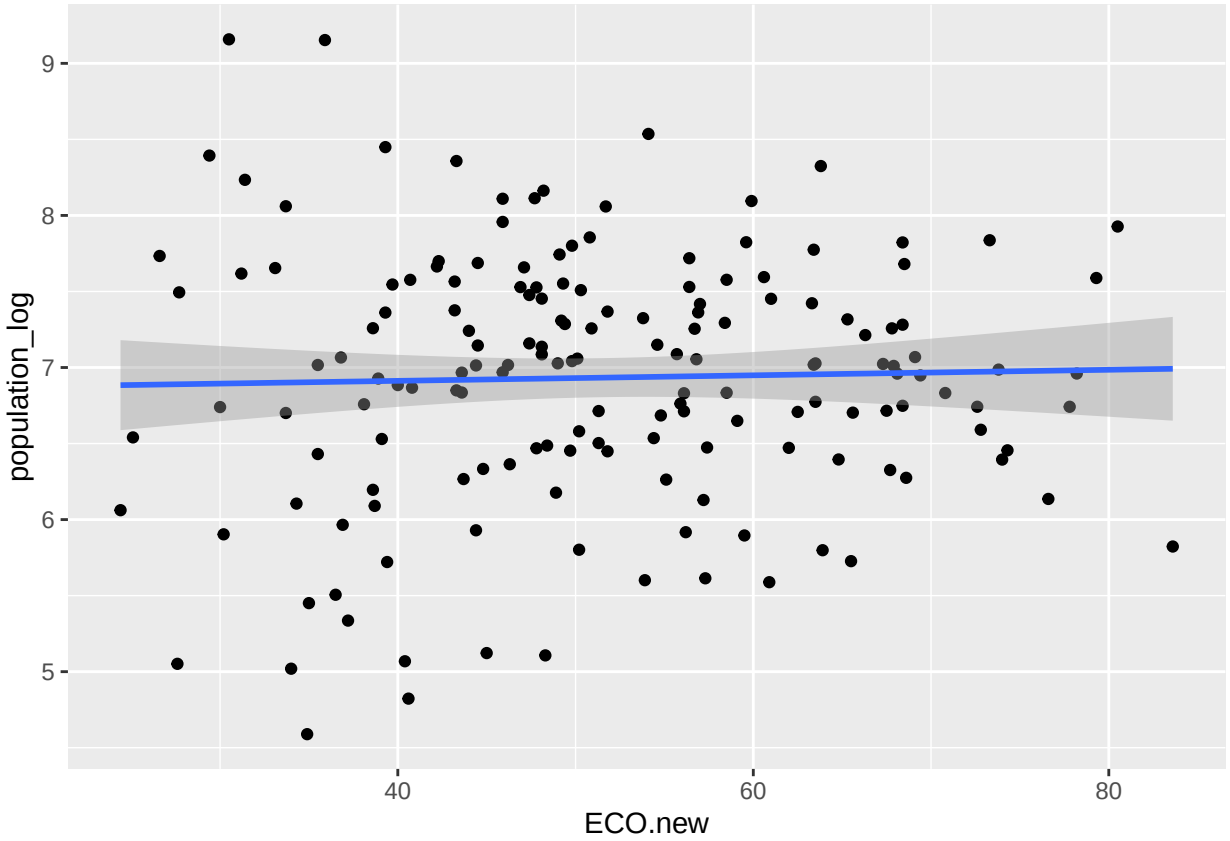

Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
103
35
75
−2
−1
0
1
2
Theoretical Quantiles
lm(population_log ~ ECO.new)


Scale−Location
75
103
35
√|Standardized residuals|
0.0
0.5
1.0
1.5
6.88
6.90
6.92
6.94
6.96
6.98
Fitted values
lm(population_log ~ ECO.new)

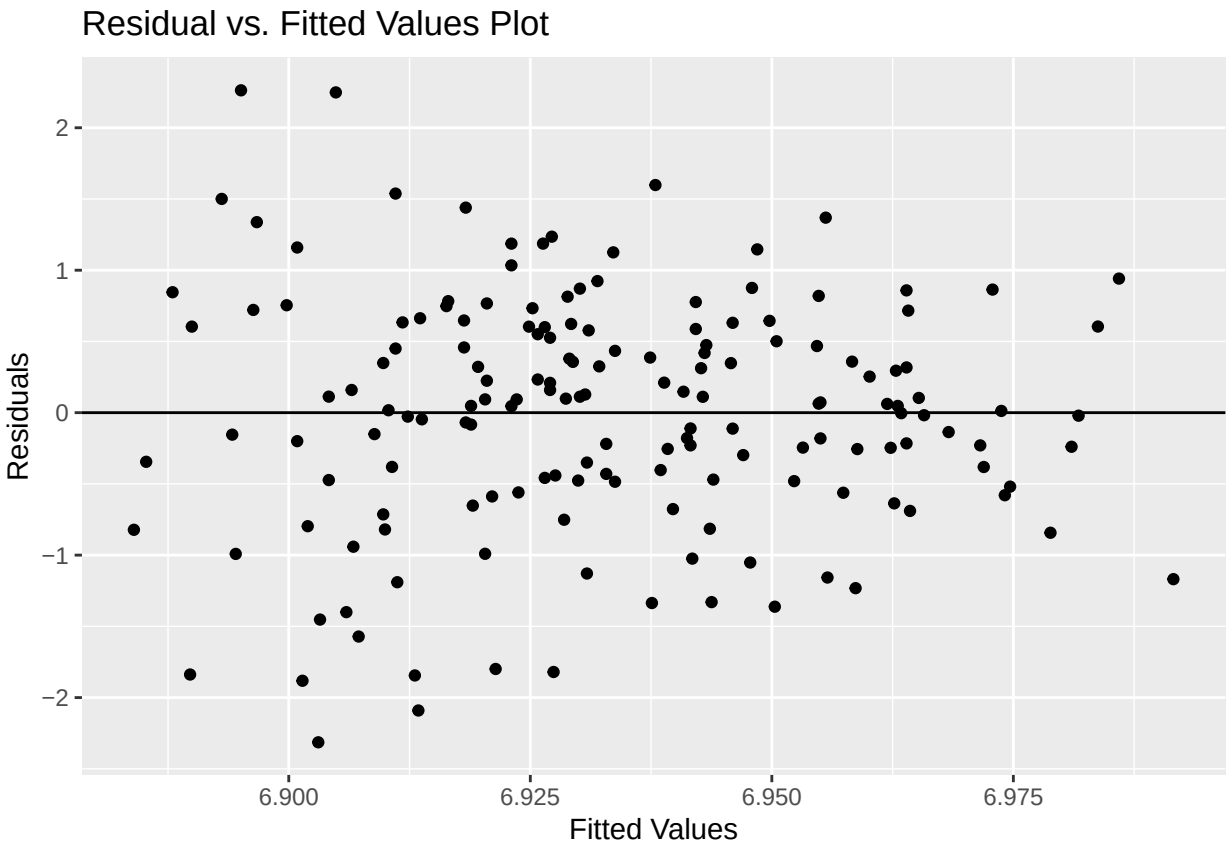



```
ggplot(epi_results.sub, aes(x = ECO.new, y = population_log)) +
  geom_point() +
  stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(lin.mod.pop, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(title='Residual vs. Fitted Values Plot', x='Fitted Values', y='Residuals')
```


Residual vs. Fitted Values Plot
Residuals
2
1
0
−1
−2
6.900
6.925
6.950
6.975
Fitted Values