Übung 3

Computational Statistics
Sommersemester 2019
April 15, 2019
J. Groos (FBMN, h_da)

Name:

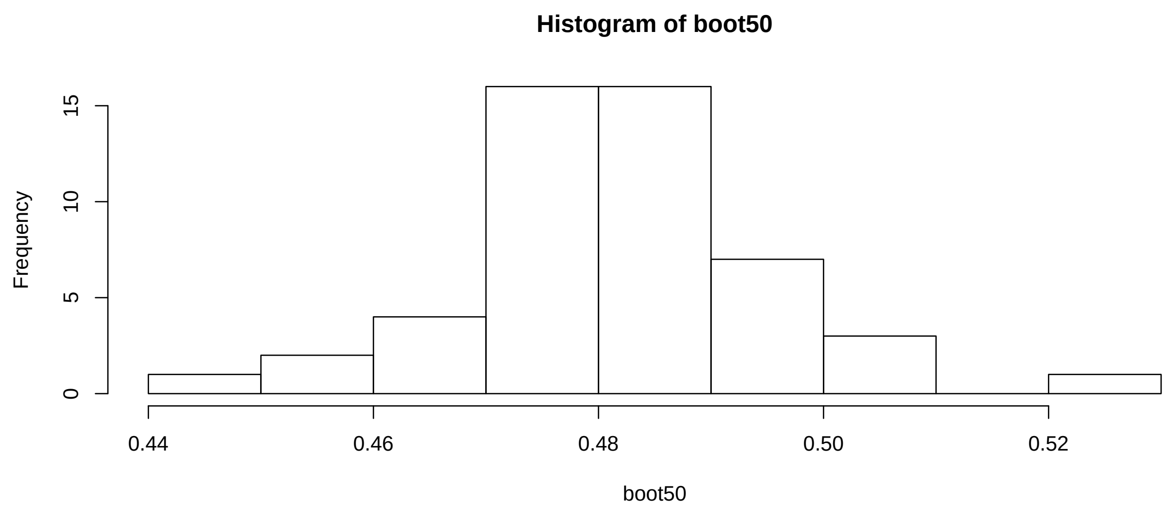
A 1

a)

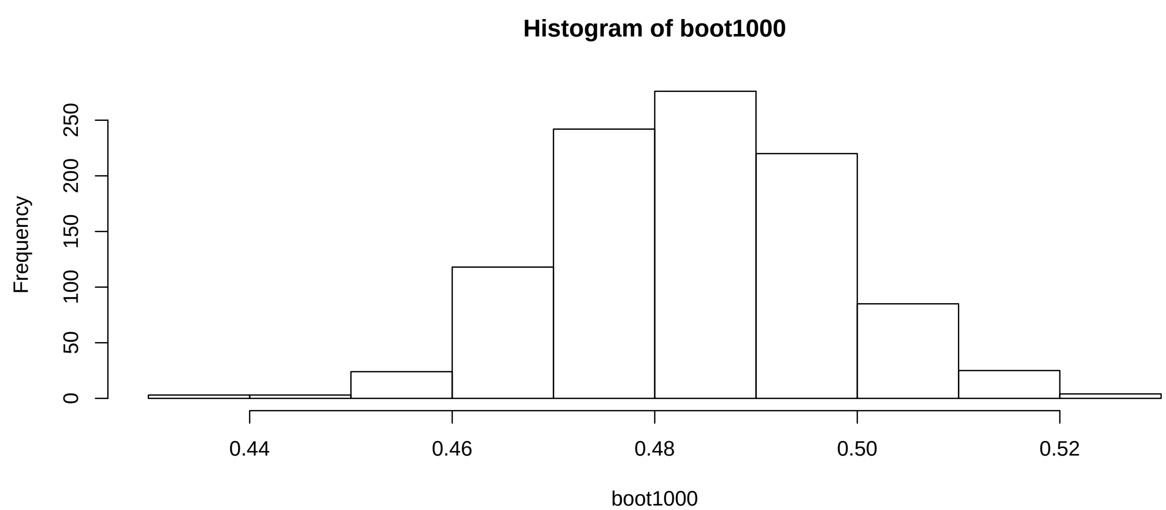
```
means50 = matrix(NA, nrow = 50, ncol = 1)
for (i in 1:50){
  summe = runif(500,0,1)
  means50[i,] = mean(summe)
}
means1000 = matrix(NA, nrow = 1000, ncol = 1)
for (i in 1:1000){
  summe = runif(500,0,1)
  means1000[i,] = mean(summe)
}
head(means1000)
#>           [,1]
#> [1,] 0.5027536
#> [2,] 0.4805654
#> [3,] 0.4894376
#> [4,] 0.4845473
#> [5,] 0.4941672
#> [6,] 0.4940287
```

b) & c)

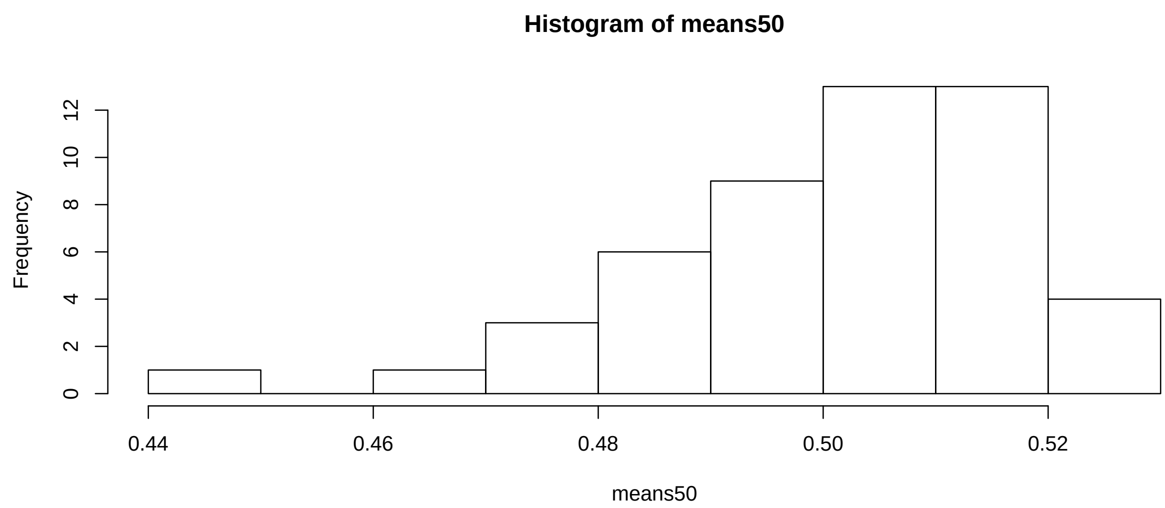
```
packageTest('boot')
bootdata <- runif(500,0,1)
bootfunc <- function(data,i) {
  d <- data[i]
  return(mean(d))
}
boot50 <- boot(bootdata, bootfunc, R = 50)
boot50 <- boot50$t
boot1000 <- boot(bootdata, bootfunc, R = 1000)
boot1000 <- boot1000$t
hist(boot50)
```



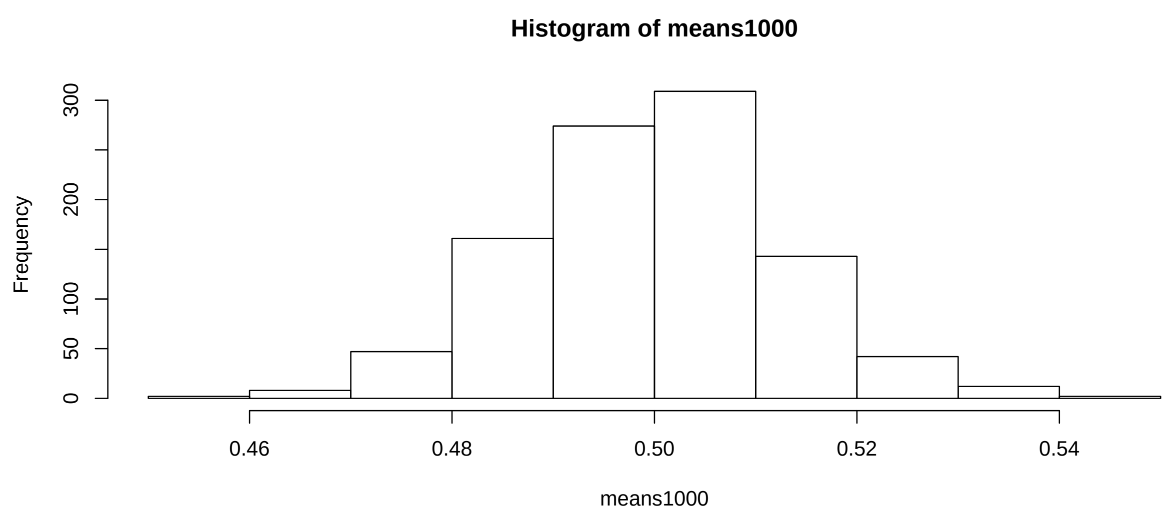

```
hist(boot1000)
```



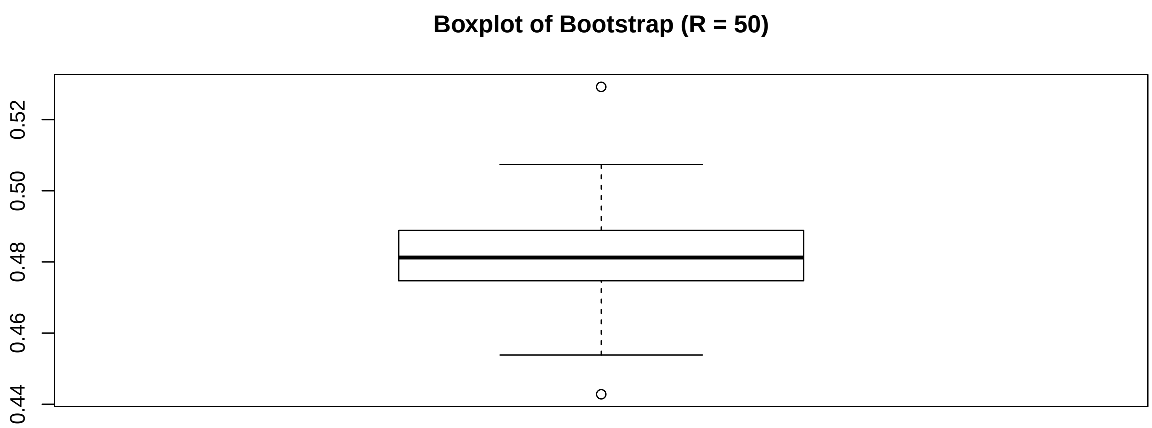

```
hist(means50)
```



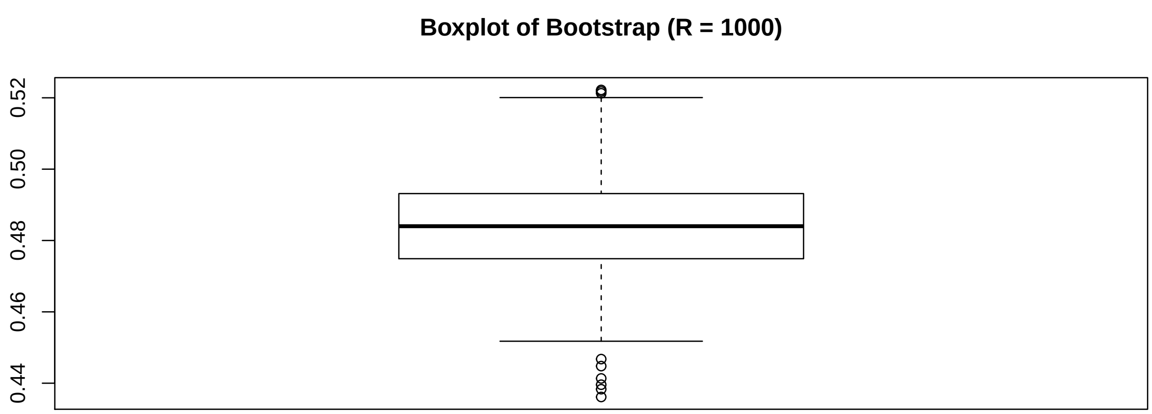

```r
hist(means1000)
```



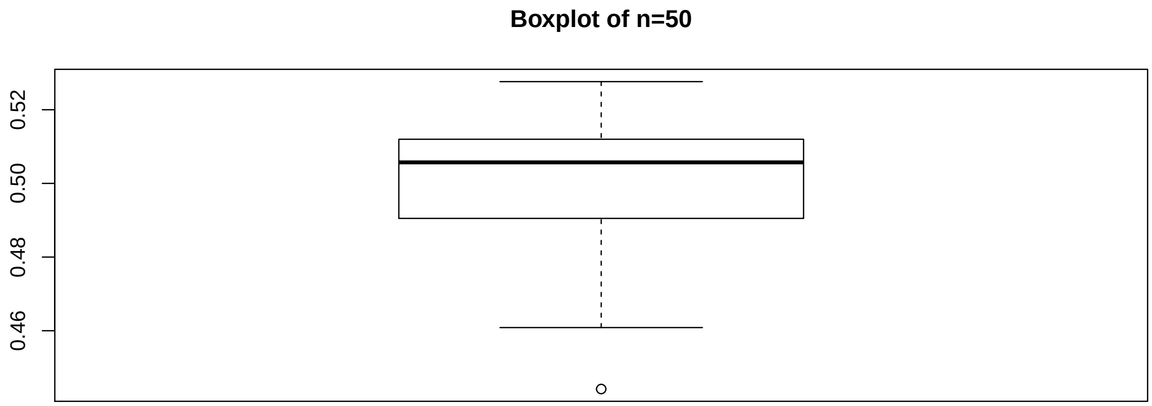

```r
boxplot(boot50, main='Boxplot of Bootstrap (R = 50)')
```



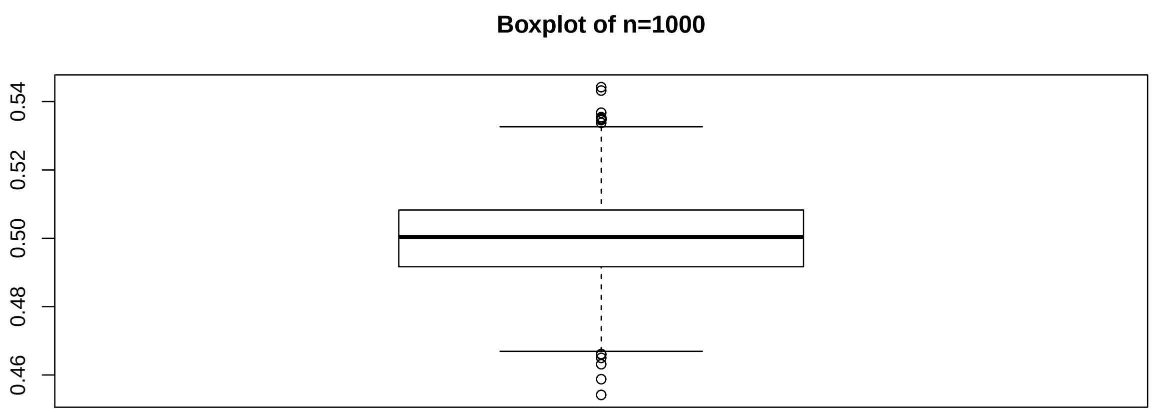

```r
boxplot(boot1000, main='Boxplot of Bootstrap (R = 1000)')
```



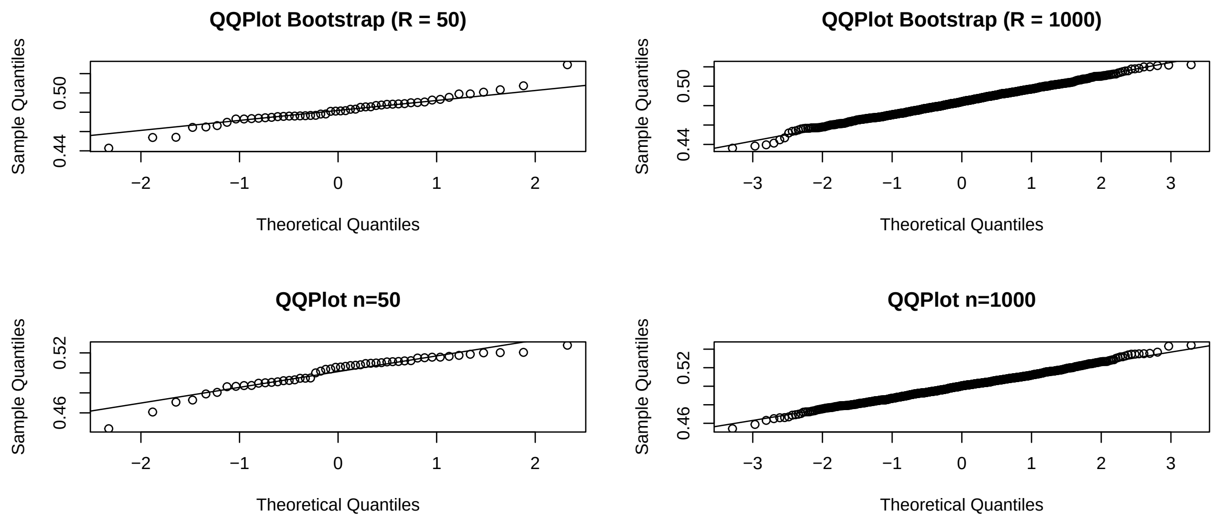

```r
boxplot(means50, main='Boxplot of n=50')
```




```r
boxplot(means1000, main='Boxplot of n=1000')
```




```r
par(mfrow=c(2,2))
qqnorm(boot50, main='QQPlot Bootstrap (R = 50)')
qqline(boot50)
qqnorm(boot1000, main='QQPlot Bootstrap (R = 1000)')
qqline(boot1000)
qqnorm(means50, main='QQPlot n=50')
qqline(means50)
qqnorm(means1000, main='QQPlot n=1000')
qqline(means1000)
```




```
shapiro.test(means50)
#>
#>  Shapiro-Wilk normality test
#>
#> data:  means50
#> W = 0.9233, p-value = 0.003109
shapiro.test(means1000)
#>
#>  Shapiro-Wilk normality test
#>
#> data:  means1000
#> W = 0.99833, p-value = 0.4477
shapiro.test(boot50)
#>
#>  Shapiro-Wilk normality test
#>
#> data:  boot50
#> W = 0.95889, p-value = 0.07985
shapiro.test(boot1000)
#>
#>  Shapiro-Wilk normality test
#>
#> data:  boot1000
#> W = 0.99779, p-value = 0.2045
```

Laut Shapiro-Wilks-Test kann für keine der vier Stichproben die Nullhypothese (Stichprobe ist normalverteilt) verworfen werden, auch wenn die Histogramme von boot50 und means50 nicht nach Normalverteilung aussehen.

d)

```
packageTest('car')

levene50 <- c(means50,boot50)
```

```r
levene1000 <- c(means1000,boot1000)

levgroup50 <- as.factor(c(rep(1, length(means50)), rep(2, length(boot50))))


levgroup1000 <- as.factor(c(rep(1, length(means1000)), rep(2, length(boot1000))))
leveneTest(levene50, levgroup50)
#> Levene's Test for Homogeneity of Variance (center = median)
#>       Df F value Pr(>F)
#> group  1  1.2861 0.2595
#>       98

leveneTest(levene1000, levgroup1000)
#> Levene's Test for Homogeneity of Variance (center = median)
#>         Df F value Pr(>F)
#> group    1   2.352 0.1253
#>       1998

t.test(means50, boot50, var.equal = TRUE)
#>
#>  Two Sample t-test
#>
#> data:  means50 and boot50
#> t = 6.0469, df = 98, p-value = 2.689e-08
#> alternative hypothesis: true difference in means is not equal to 0
#> 95 percent confidence interval:
#>  0.01260618 0.02492212
#> sample estimates:
#> mean of x mean of y
#> 0.5005581 0.4817939

t.test(means1000, boot1000, var.equal = TRUE)
#>
#>  Two Sample t-test
#>
#> data:  means1000 and boot1000
#> t = 27.282, df = 1998, p-value < 2.2e-16
#> alternative hypothesis: true difference in means is not equal to 0
#> 95 percent confidence interval:
#>  0.01475133 0.01703640
#> sample estimates:
#> mean of x mean of y
#> 0.4999347 0.4840408
```

Um t-Tests durchführen zu können, müssen die Gruppen erst auf Varianzhomogenität geprüft werden. Diese wird mit dem Levene-Test geprüft. Laut diesem ist die Varianzhomogenität sowohl für Means_{sim50} und Means_{boot50} und $\text{Means}_{sim1000}$ und $\text{Means}_{boot1000}$ gegeben.

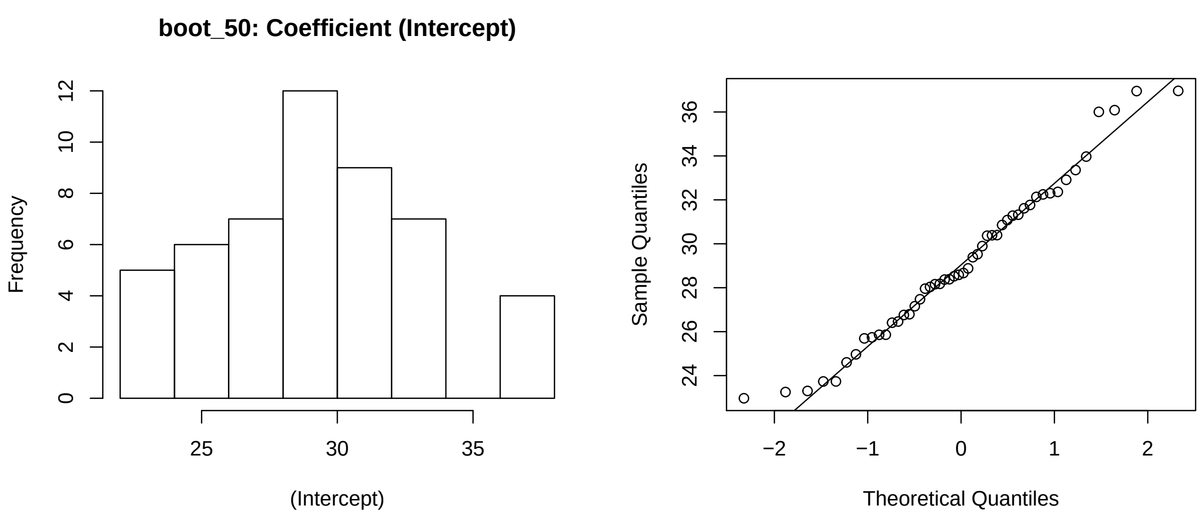
Der t-Test zur Überprüfung der Mittelwerte von Means_{sim50} und Means_{boot50} lieferte ein insignifikantes Ergebnis. Aus diesem Grund kann die Nullhypothese ("Differenz der Mittelwerte gleich 0.") nicht verworfen werden.

Für die Mittelwerte von $\text{Means}_{sim1000}$ und $\text{Means}_{boot1000}$ kann die Nullhypothese mit einem Siginifkanzniveau von 95% verworfen werden.

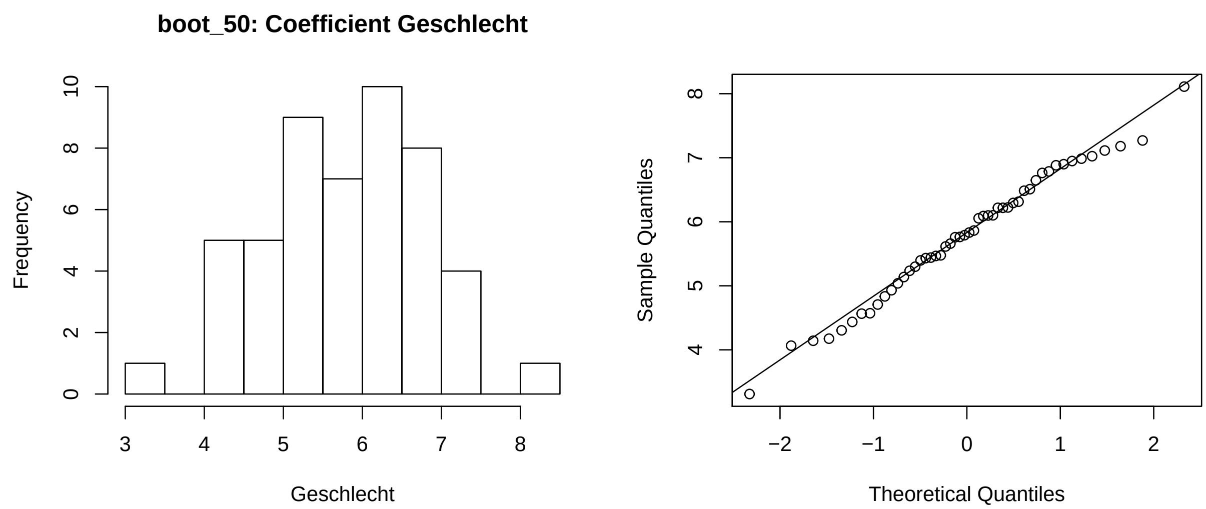
Aufgabe 2

a)

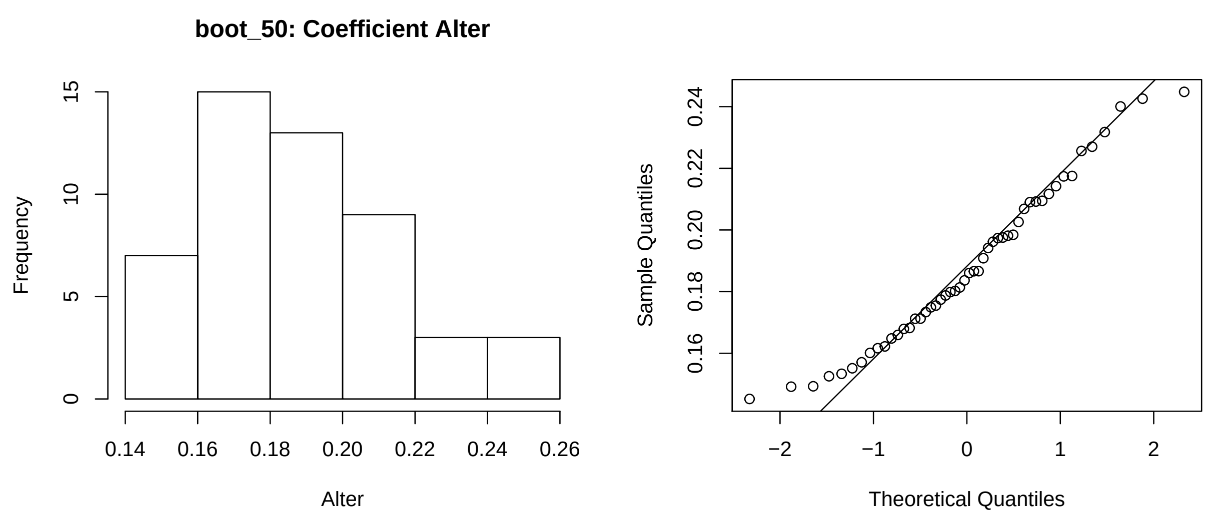
```r
load("Donald.RData")
vergleich <- lm(data = Donald_1, Trump ~ Geschlecht + Alter + Minderheit + Fremdenfeindlich + IQ)
theta <- function(formula, data, indices){
  d <- data[indices,]
  fit <- lm(formula, data = d)
  return(coef(fit))
}

coef_names <- names(vergleich$coefficients)

boot50_1 <- boot(data=Donald_1, statistic=theta, R=50, formula=Trump~.)
par(mfrow=c(1,2))
for (i in 1:6){
hist(boot50_1$t[,i], main=paste("boot_50: Coefficient", coef_names[i]), xlab = coef_names[i])
qqnorm(boot50_1$t[,i], main = NULL)
qqline(boot50_1$t[,i])
print(t.test(boot50_1$t[,i], mu=vergleich$coefficients[i]))
}
```




```
#>
#>  One Sample t-test
#>
#> data:  boot50_1$t[, i]
#> t = -1.1499, df = 49, p-value = 0.2558
#> alternative hypothesis: true mean is not equal to 29.73834
#> 95 percent confidence interval:
#>  28.13126 30.17565
#> sample estimates:
#> mean of x
#>  29.15346
```

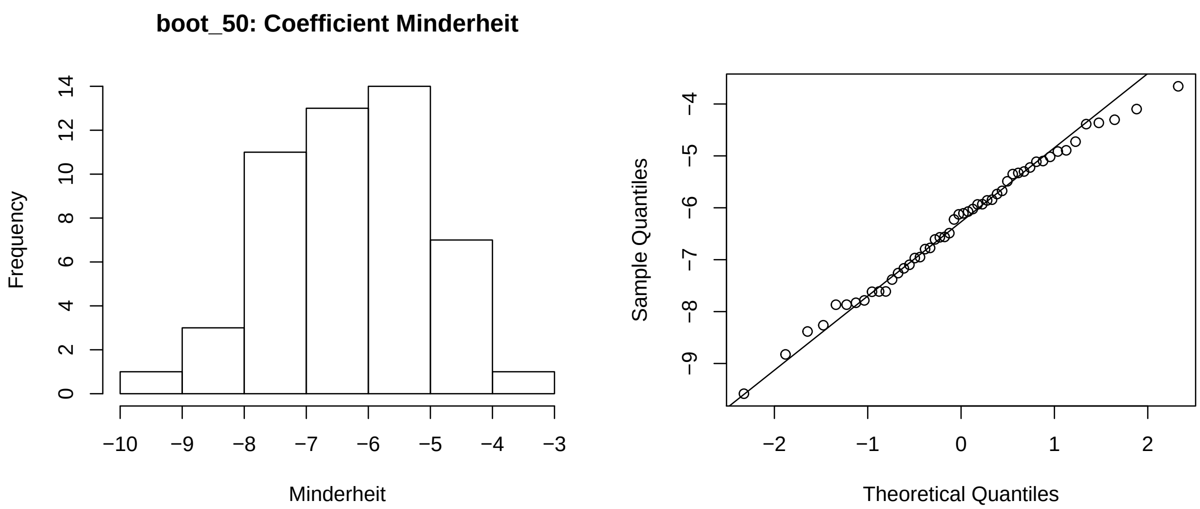



```
#>
#>  One Sample t-test
#>
#> data:  boot50_1$t[, i]
#> t = 0.23409, df = 49, p-value = 0.8159
#> alternative hypothesis: true mean is not equal to 5.75572
#> 95 percent confidence interval:
#>  5.503144 6.074898
#> sample estimates:
#> mean of x
#>  5.789021
```

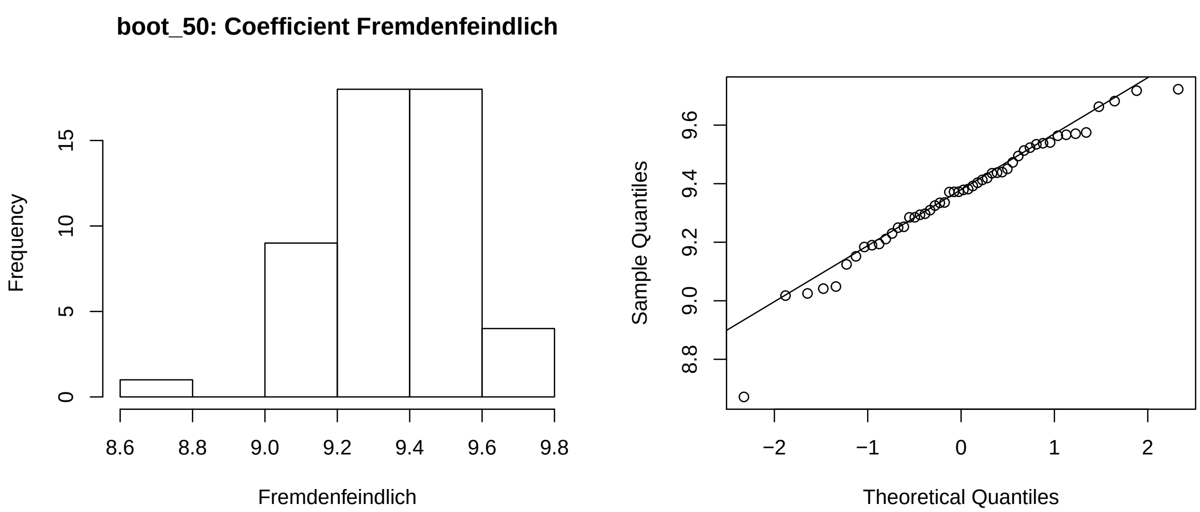



```
#>
#>  One Sample t-test
#>
#> data:  boot50_1$t[, i]
#> t = 1.748, df = 49, p-value = 0.08673
#> alternative hypothesis: true mean is not equal to 0.1815267
```

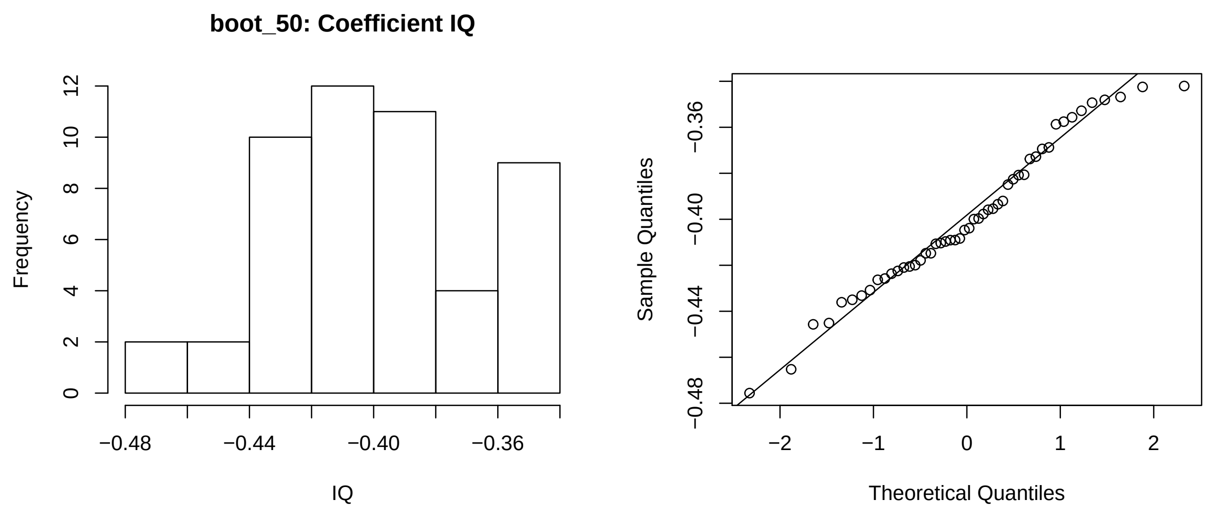
```
#> 95 percent confidence interval:
#>  0.1805527 0.1955171
#> sample estimates:
#> mean of x
#> 0.1880349
```



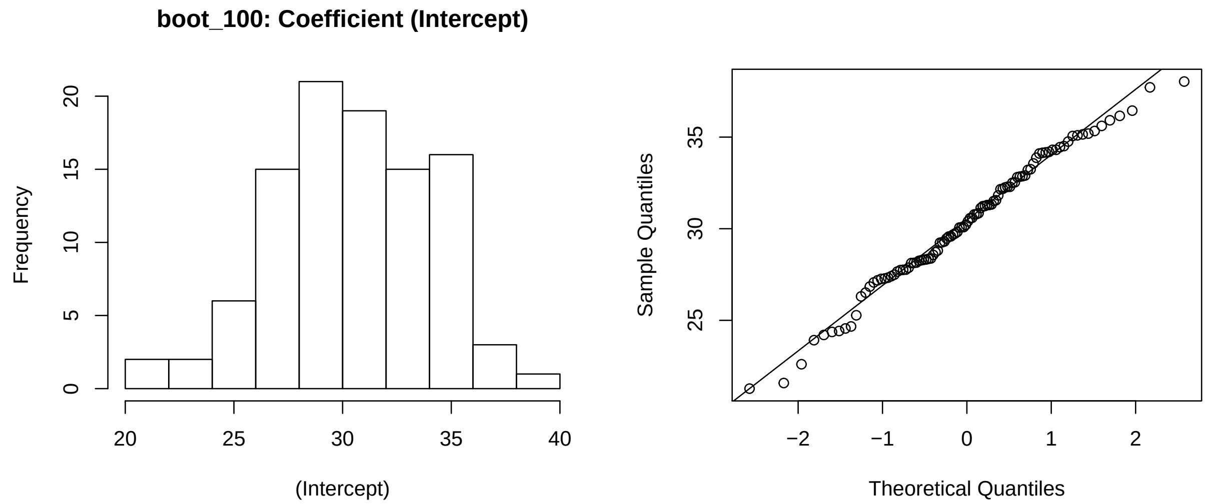

```
#>
#>  One Sample t-test
#>
#> data:  boot50_1$t[, i]
#> t = 1.5126, df = 49, p-value = 0.1368
#> alternative hypothesis: true mean is not equal to -6.575863
#> 95 percent confidence interval:
#>  -6.668653 -5.918275
#> sample estimates:
#> mean of x
#> -6.293464
```



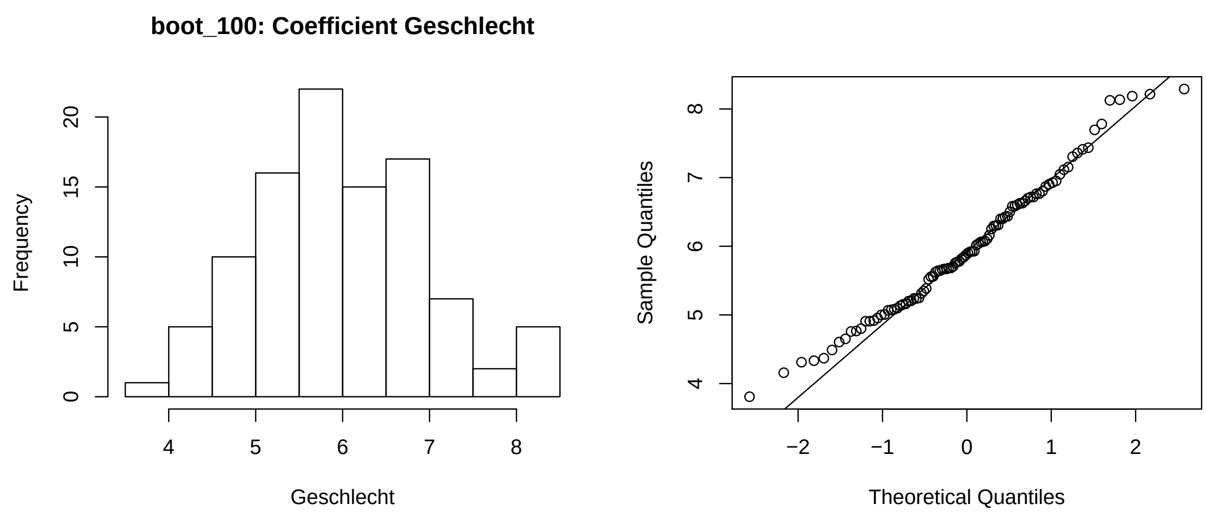
```
#>
#>  One Sample t-test
#>
#> data:  boot50_1$t[, i]
#> t = 0.35607, df = 49, p-value = 0.7233
#> alternative hypothesis: true mean is not equal to 9.349838
#> 95 percent confidence interval:
#>  9.302384 9.417730
#> sample estimates:
#> mean of x
#>  9.360057
```




```
#>
#>  One Sample t-test
#>
#> data:  boot50_1$t[, i]
#> t = 0.38054, df = 49, p-value = 0.7052
#> alternative hypothesis: true mean is not equal to -0.4013497
#> 95 percent confidence interval:
#>  -0.4087766 -0.3904530
#> sample estimates:
#>  mean of x
#> -0.3996148
boot100_1 <- boot(data=Donald_1, statistic=theta, R=100, formula=Trump~.)
par(mfrow=c(1,2))
for (i in 1:6){
hist(boot100_1$t[,i], main=paste("boot_100: Coefficient", coef_names[i]), xlab = coef_names[i])
qqnorm(boot100_1$t[,i], main = NULL)
qqline(boot100_1$t[,i])
print(t.test(boot100_1$t[,i], mu=vergleich$coefficients[i]))
}
```

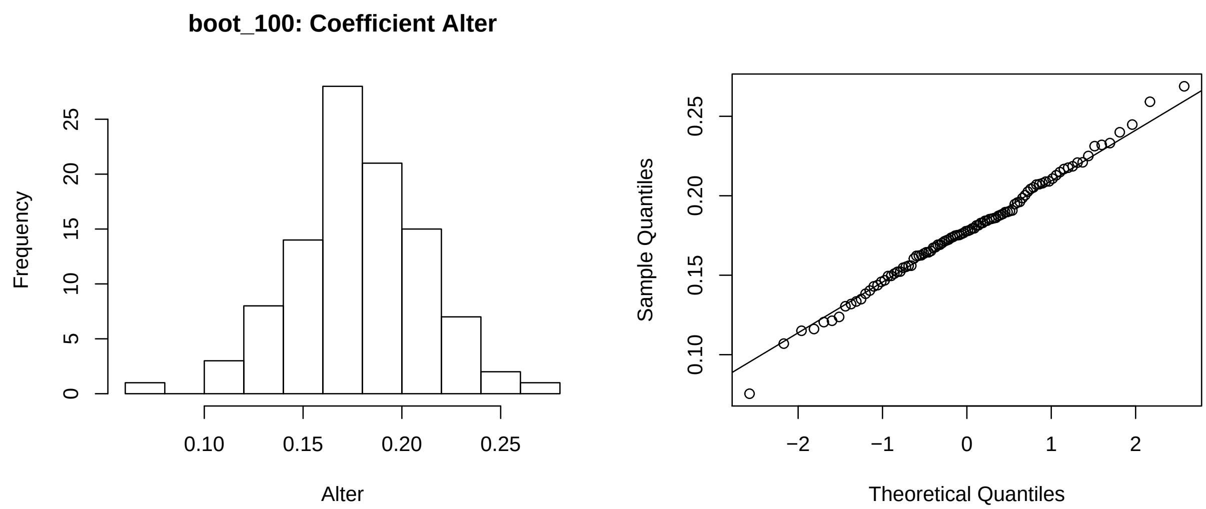



```
#>
#>  One Sample t-test
#>
#> data:  boot100_1$t[, i]
#> t = 1.7239, df = 99, p-value = 0.08784
#> alternative hypothesis: true mean is not equal to 29.73834
#> 95 percent confidence interval:
#>  29.64540 31.06252
#> sample estimates:
#> mean of x
#>  30.35396
```

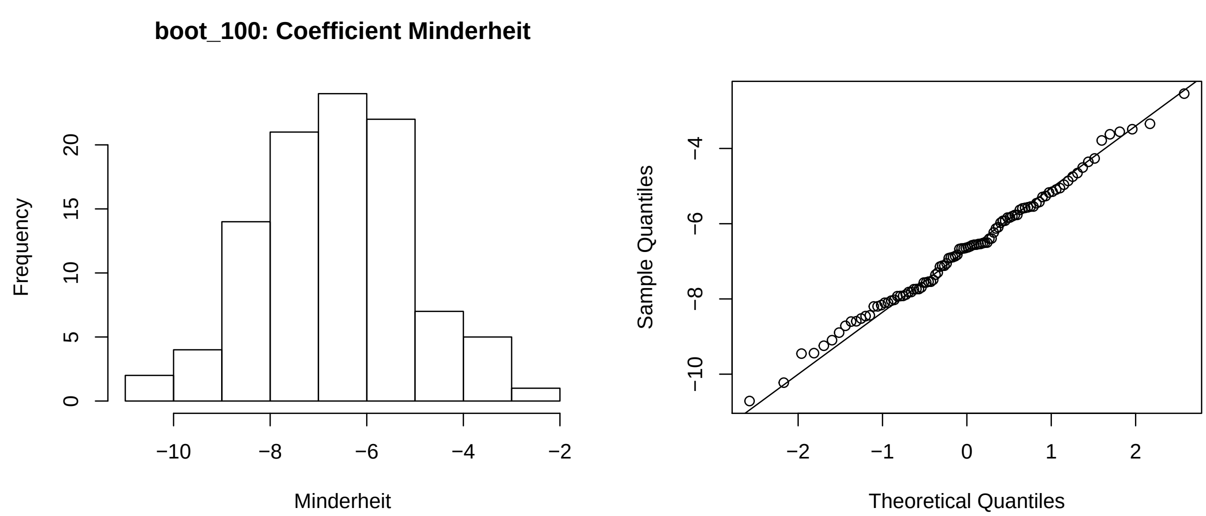



```
#>
#>  One Sample t-test
#>
#> data:  boot100_1$t[, i]
#> t = 2.2386, df = 99, p-value = 0.02742
#> alternative hypothesis: true mean is not equal to 5.75572
```

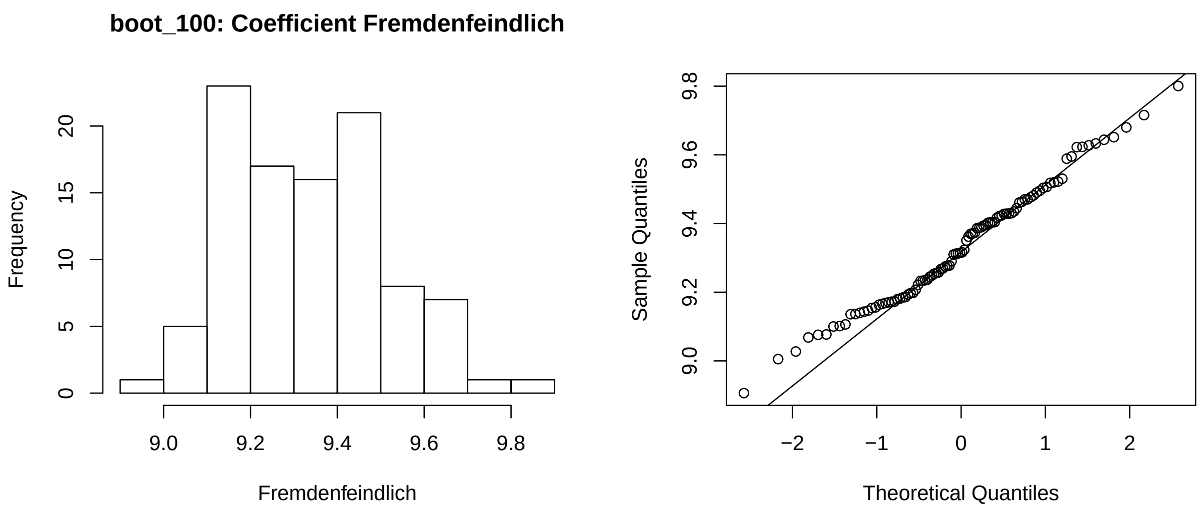
```
#> 95 percent confidence interval:
#>  5.780898 6.173650
#> sample estimates:
#> mean of x
#>  5.977274
```



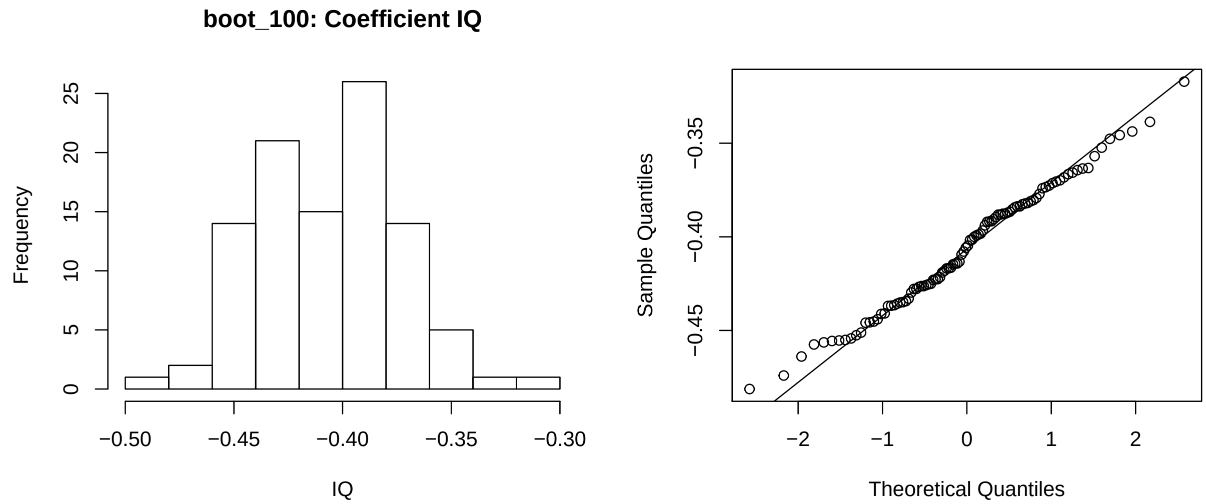

```
#>
#>  One Sample t-test
#>
#> data:  boot100_1$t[, i]
#> t = -1.0572, df = 99, p-value = 0.293
#> alternative hypothesis: true mean is not equal to 0.1815267
#> 95 percent confidence interval:
#>  0.1712778 0.1846506
#> sample estimates:
#> mean of x
#> 0.1779642
```



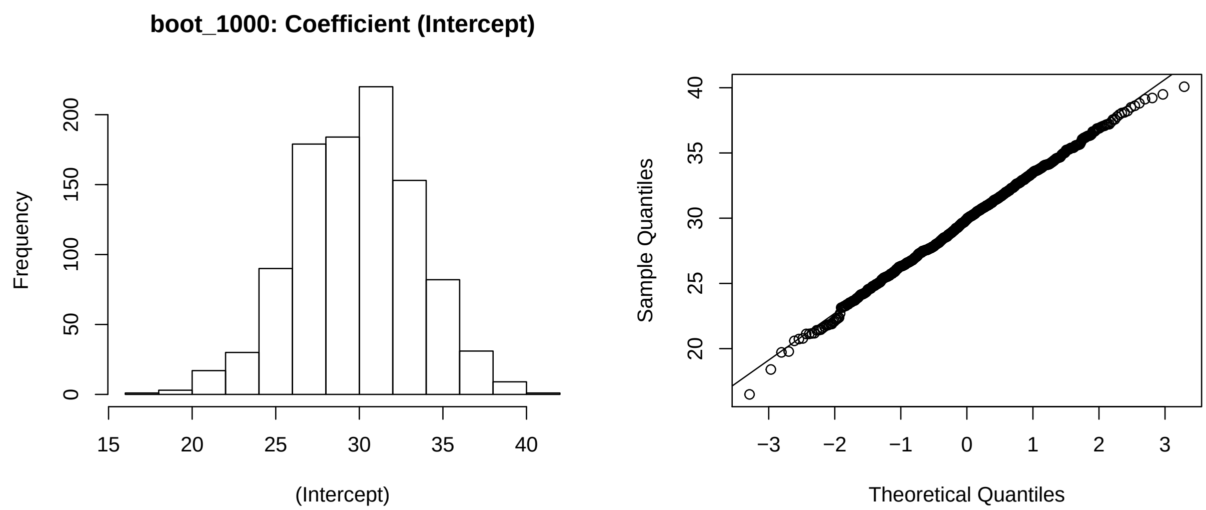
```
#>
#>  One Sample t-test
#>
#> data:  boot100_1$t[, i]
#> t = -0.5942, df = 99, p-value = 0.5537
#> alternative hypothesis: true mean is not equal to -6.575863
#> 95 percent confidence interval:
#>  -6.979361 -6.358339
#> sample estimates:
#> mean of x
#>  -6.66885
```



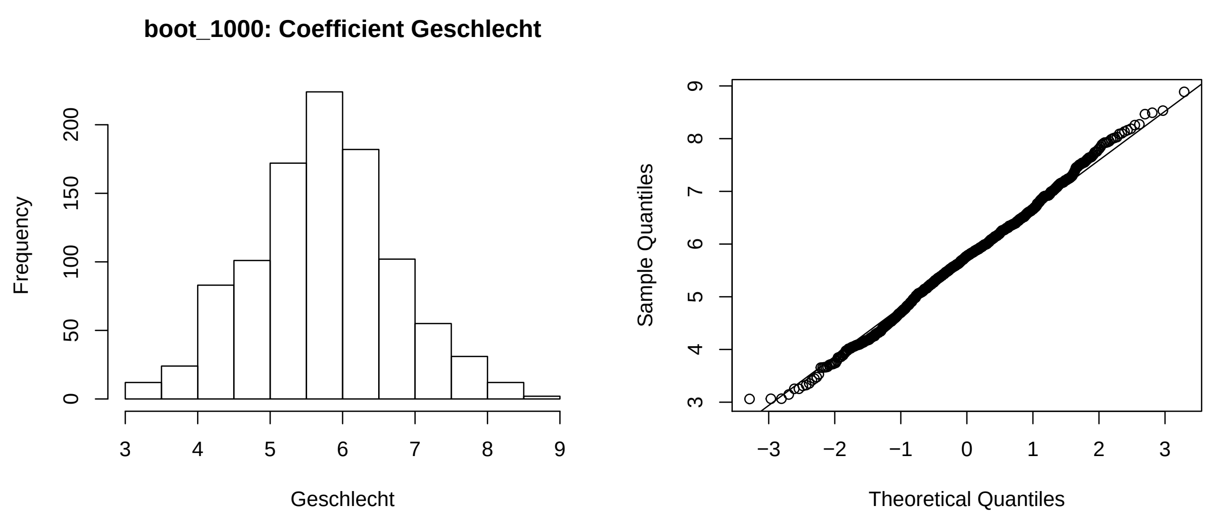

```
#>
#>  One Sample t-test
#>
#> data:  boot100_1$t[, i]
#> t = -0.94105, df = 99, p-value = 0.349
#> alternative hypothesis: true mean is not equal to 9.349838
#> 95 percent confidence interval:
#>  9.298257 9.368232
#> sample estimates:
#> mean of x
#>  9.333245
```




```
#>
#>  One Sample t-test
#>
#> data:  boot100_1$t[, i]
#> t = -1.406, df = 99, p-value = 0.1629
#> alternative hypothesis: true mean is not equal to -0.4013497
#> 95 percent confidence interval:
#>  -0.4126399 -0.3994242
#> sample estimates:
#> mean of x
#> -0.406032
boot1000_1 <- boot(data=Donald_1, statistic=theta, R=1000, formula=Trump~.)
par(mfrow=c(1,2))
for (i in 1:6){
hist(boot1000_1$t[,i], main=paste("boot_1000: Coefficient", coef_names[i]), xlab = coef_names[i])
qqnorm(boot1000_1$t[,i], main = NULL)
qqline(boot1000_1$t[,i])
print(t.test(boot1000_1$t[,i], mu=vergleich$coefficients[i]))
}
```

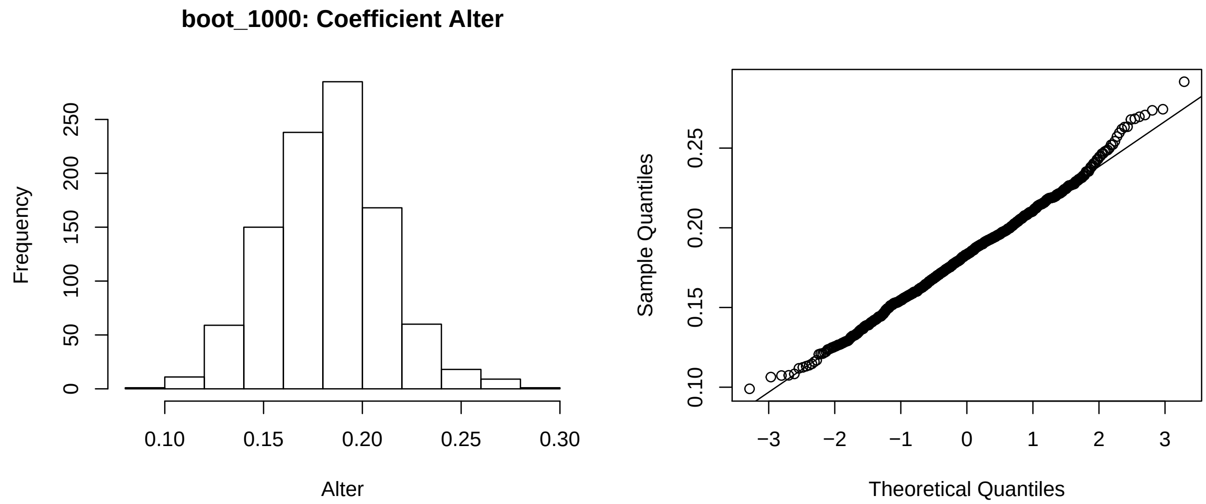



```
#>
#>  One Sample t-test
#>
#> data:  boot1000_1$t[, i]
#> t = 0.55418, df = 999, p-value = 0.5796
#> alternative hypothesis: true mean is not equal to 29.73834
#> 95 percent confidence interval:
#>  29.57863 30.02377
#> sample estimates:
#> mean of x
#>   29.8012
```

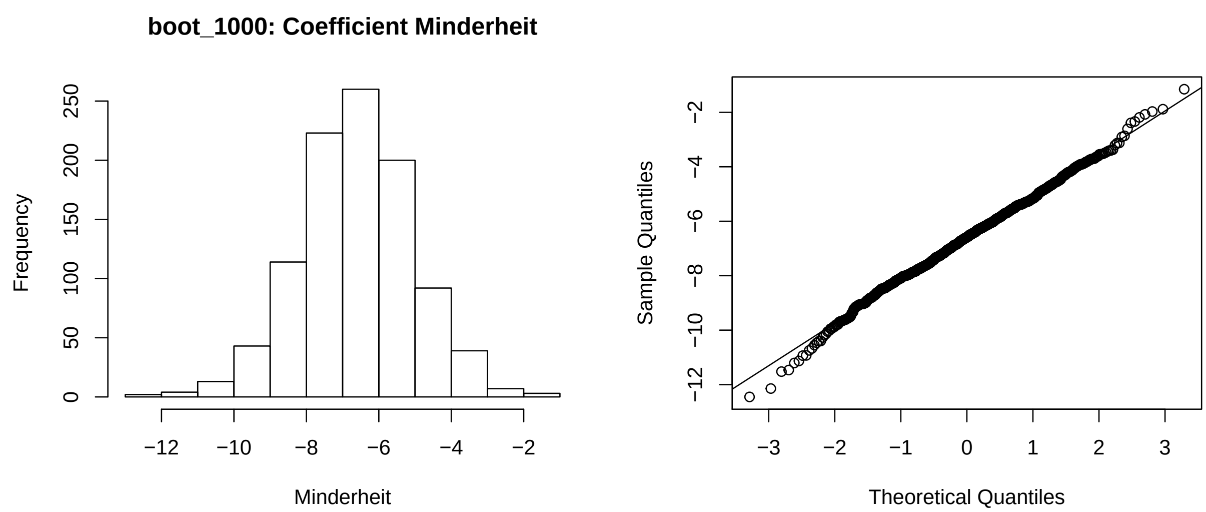



```
#>
#>  One Sample t-test
#>
#> data:  boot1000_1$t[, i]
#> t = -0.69897, df = 999, p-value = 0.4847
#> alternative hypothesis: true mean is not equal to 5.75572
```

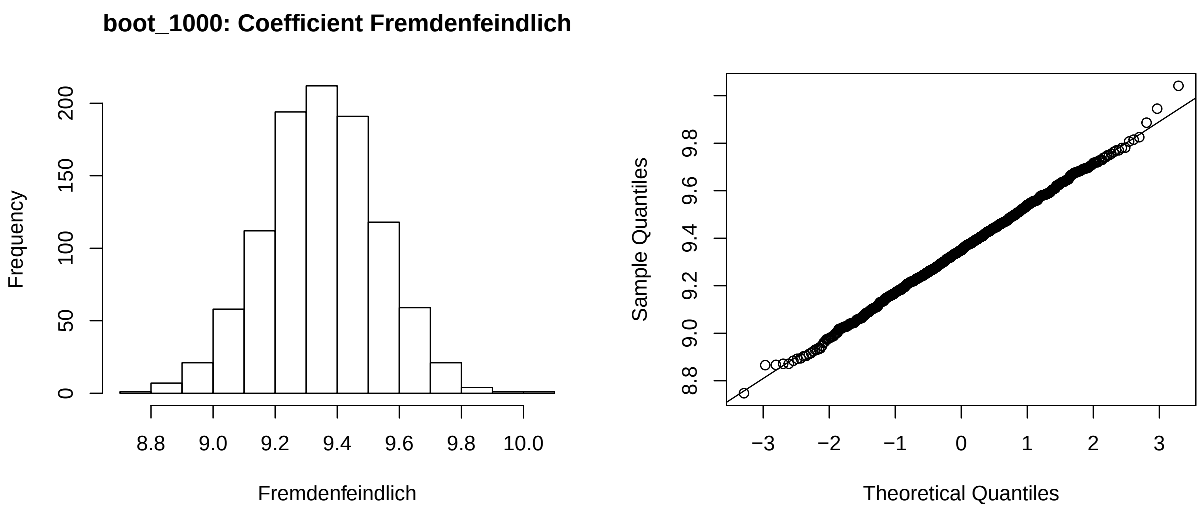
```
#> 95 percent confidence interval:
#>  5.672561 5.795196
#> sample estimates:
#> mean of x
#>  5.733879
```



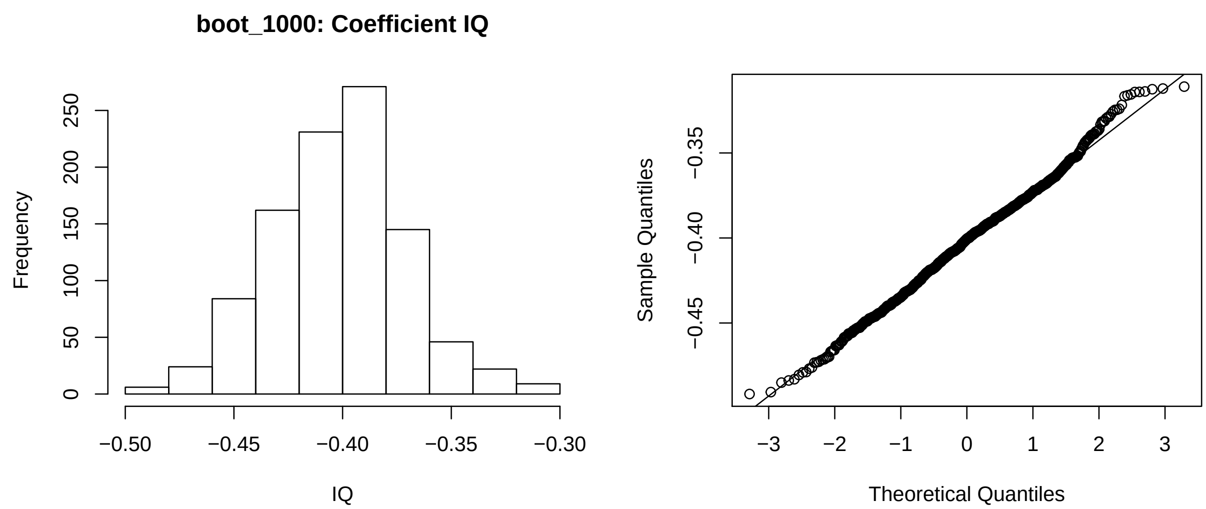

```
#>
#>  One Sample t-test
#>
#> data:  boot1000_1$t[, i]
#> t = 1.4219, df = 999, p-value = 0.1554
#> alternative hypothesis: true mean is not equal to 0.1815267
#> 95 percent confidence interval:
#>  0.1810316 0.1846272
#> sample estimates:
#> mean of x
#> 0.1828294
```



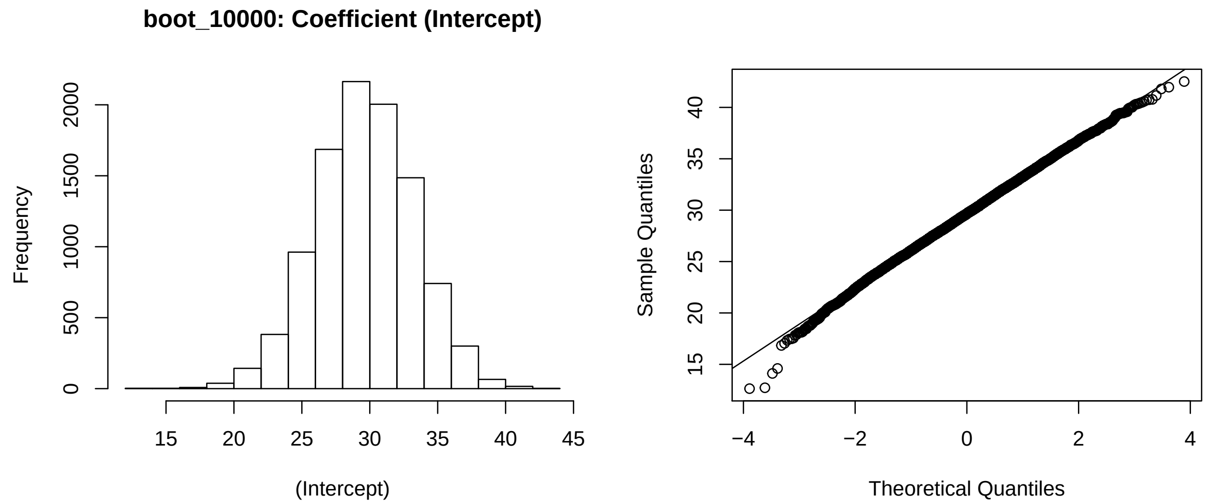
```
#>
#>  One Sample t-test
#>
#> data:  boot1000_1$t[, i]
#> t = -0.86717, df = 999, p-value = 0.3861
#> alternative hypothesis: true mean is not equal to -6.575863
#> 95 percent confidence interval:
#>  -6.715782 -6.521706
#> sample estimates:
#> mean of x
#> -6.618744
```



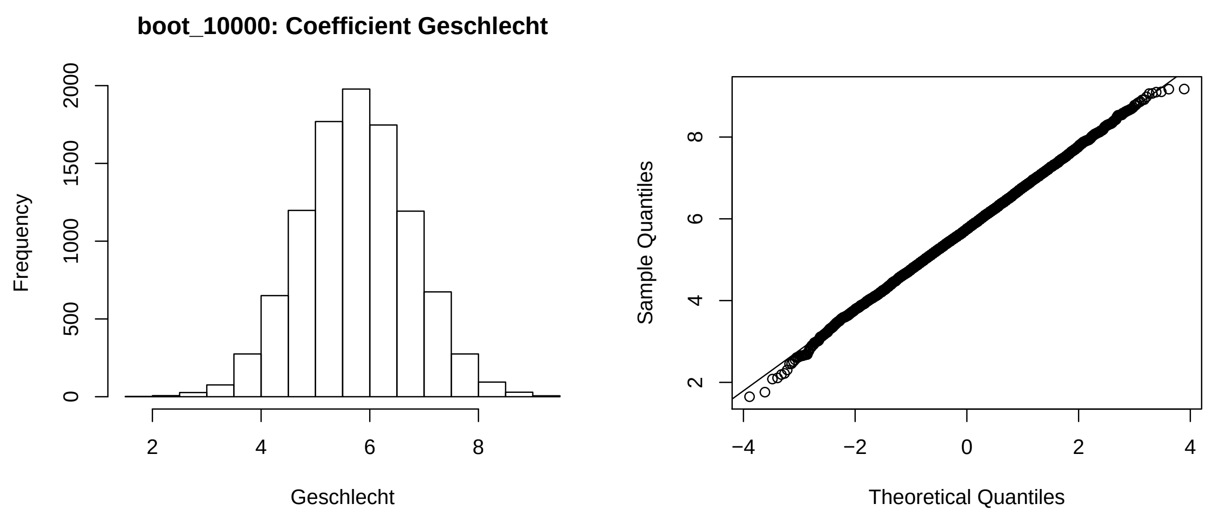

```
#>
#>  One Sample t-test
#>
#> data:  boot1000_1$t[, i]
#> t = 0.22419, df = 999, p-value = 0.8227
#> alternative hypothesis: true mean is not equal to 9.349838
#> 95 percent confidence interval:
#>  9.339718 9.362569
#> sample estimates:
#> mean of x
#>  9.351143
```




```
#>
#>  One Sample t-test
#>
#> data:  boot1000_1$t[, i]
#> t = -0.93855, df = 999, p-value = 0.3482
#> alternative hypothesis: true mean is not equal to -0.4013497
#> 95 percent confidence interval:
#>  -0.4042033 -0.4003426
#> sample estimates:
#>  mean of x
#> -0.4022729
boot10000_1 <- boot(data=Donald_1, statistic=theta, R=10000, formula=Trump~.)
par(mfrow=c(1,2))
for (i in 1:6){
hist(boot10000_1$t[,i], main=paste("boot_10000: Coefficient", coef_names[i]), xlab = coef_names[i])
qqnorm(boot10000_1$t[,i], main = NULL)
qqline(boot10000_1$t[,i])
print(t.test(boot10000_1$t[,i], mu=vergleich$coefficients[i]))
}
```

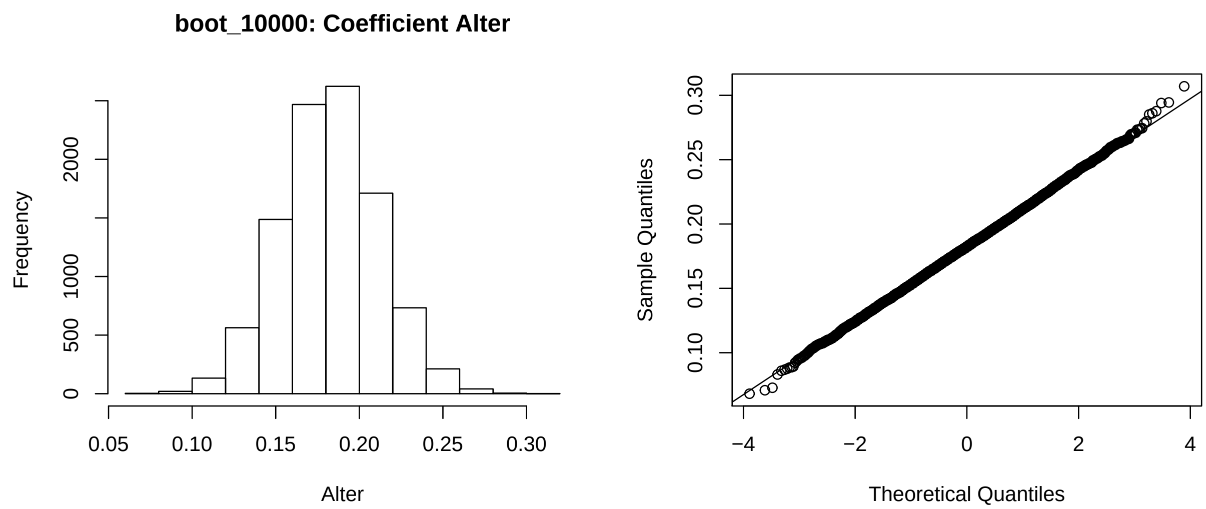



```
#>
#>  One Sample t-test
#>
#> data:  boot10000_1$t[, i]
#> t = -2.4243, df = 9999, p-value = 0.01536
#> alternative hypothesis: true mean is not equal to 29.73834
#> 95 percent confidence interval:
#>  29.58034 29.72162
#> sample estimates:
#> mean of x
#>  29.65098
```

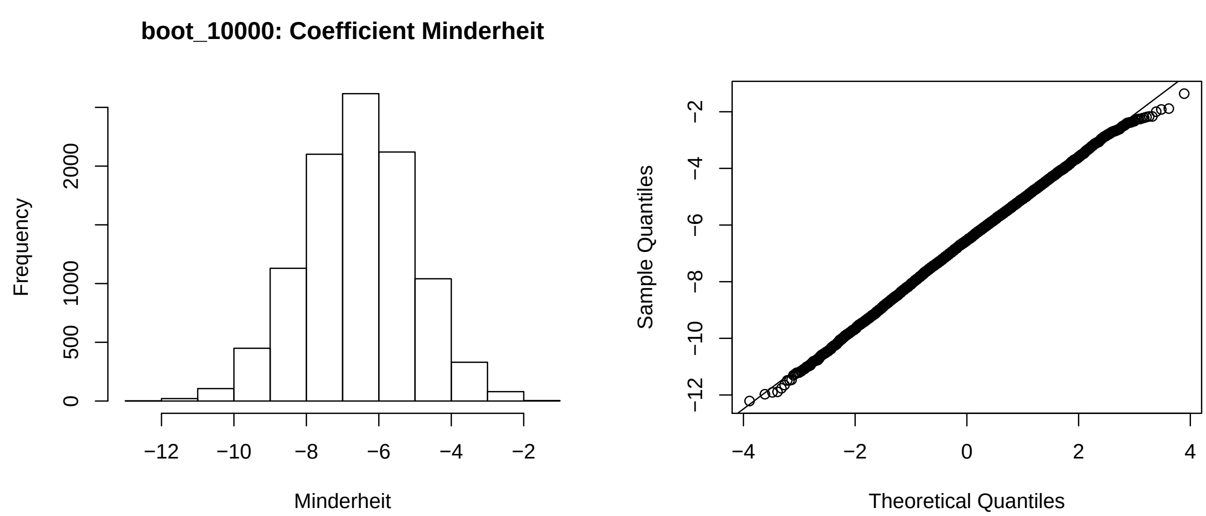



```
#>
#>  One Sample t-test
#>
#> data:  boot10000_1$t[, i]
#> t = 0.047807, df = 9999, p-value = 0.9619
#> alternative hypothesis: true mean is not equal to 5.75572
```

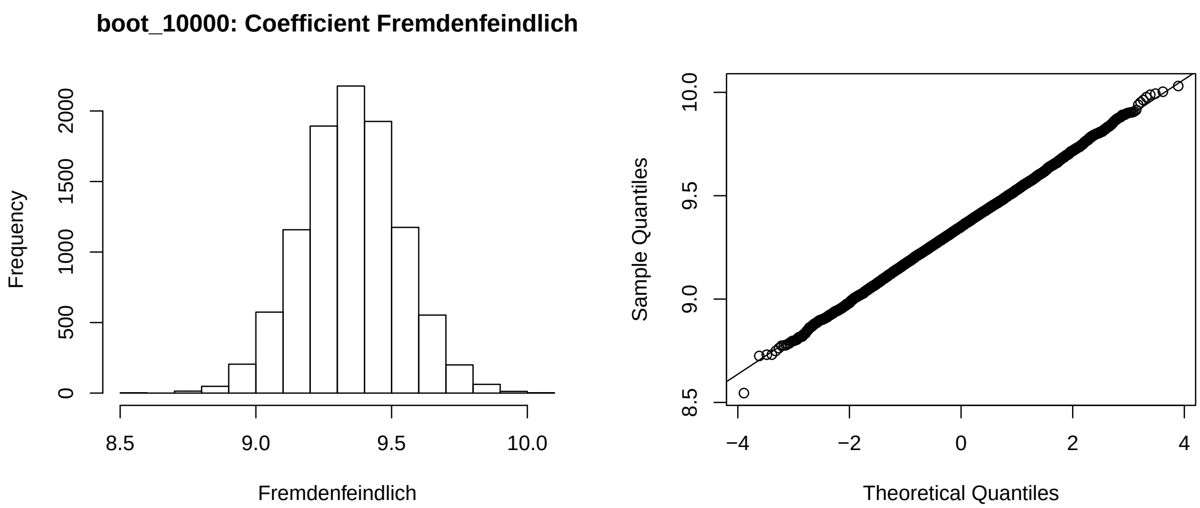
```
#> 95 percent confidence interval:
#>  5.736572 5.775825
#> sample estimates:
#> mean of x
#>  5.756198
```



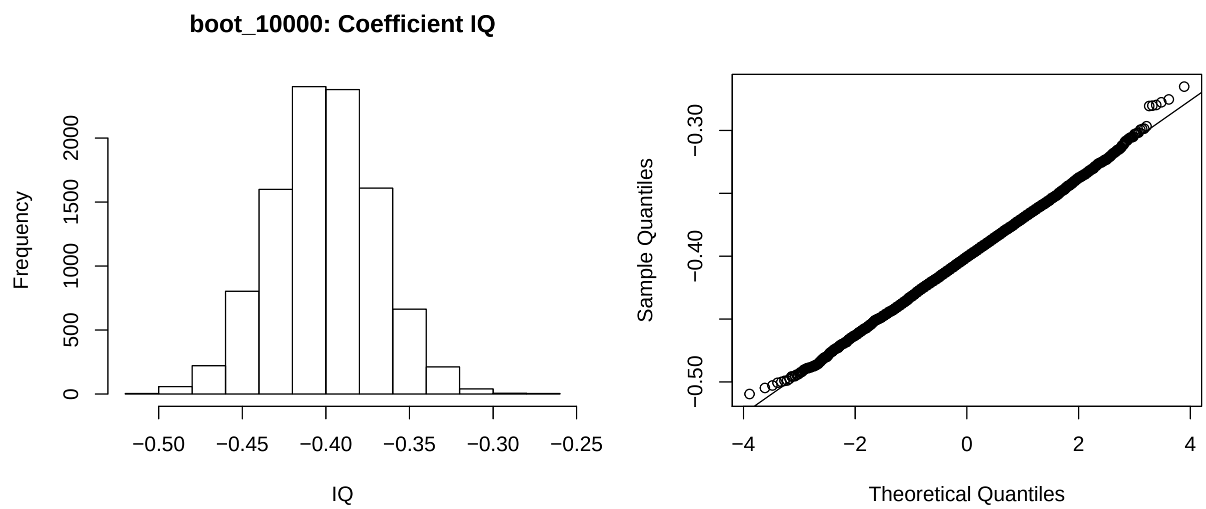

```
#>
#>  One Sample t-test
#>
#> data:  boot10000_1$t[, i]
#> t = 3.1769, df = 9999, p-value = 0.001493
#> alternative hypothesis: true mean is not equal to 0.1815267
#> 95 percent confidence interval:
#>  0.1818826 0.1830295
#> sample estimates:
#> mean of x
#> 0.1824561
```



```
#>
#>  One Sample t-test
#>
#> data:  boot10000_1$t[, i]
#> t = 0.52339, df = 9999, p-value = 0.6007
#> alternative hypothesis: true mean is not equal to -6.575863
#> 95 percent confidence interval:
#>  -6.597539 -6.538394
#> sample estimates:
#> mean of x
#> -6.567966
```




```
#>
#>  One Sample t-test
#>
#> data:  boot10000_1$t[, i]
#> t = 0.23136, df = 9999, p-value = 0.817
#> alternative hypothesis: true mean is not equal to 9.349838
#> 95 percent confidence interval:
#>  9.346706 9.353808
#> sample estimates:
#> mean of x
#>  9.350257
```




```
#>
#>  One Sample t-test
#>
#> data:  boot10000_1$t[, i]
#> t = 1.5753, df = 9999, p-value = 0.1152
#> alternative hypothesis: true mean is not equal to -0.4013497
#> 95 percent confidence interval:
#>  -0.4014698 -0.4002468
#> sample estimates:
#>  mean of x
#> -0.4008583
```

Hier untersuchen wir, ob wir durch das Bootstrappingverfahren auf das gleiche Ergebnis kommen wie die lineare Regression. Außerdem vergleichen wir die Verteilungen der Parameterschätzer für verschiedene Replikationsgrößen. Zu den Verteilungen lässt sich, wie bei Aufgabe 1, sagen, dass sich die Verteilungen mit mehr Replikationen immer mehr an die Normalverteilung annähern. Dies lässt sich an den Histogrammen, sowie den qqnorm-Plots ablesen. Die t-Tests sind tendenziell insignifikant (p-Wert > 0.05), die Nullhypothesen können deshalb nicht verworfen werden. Es gibt vereinzelt Ausnahmen (Beispiel: Bei boot_100 Geschlecht).

b)

```
Konfi50 <- matrix(NA,6,2)
rownames(Konfi50) <- c("Intercept","Geschlecht","Alter","Minderheit","Fremdenfeindlich","IQ")
colnames(Konfi50) <- c("2,5%","97,5%")
for(i in 1:6)
{
    Konfi50[i,1] <- boot.ci(boot50_1, type="basic", index = i)$basic[4]
    Konfi50[i,2] <- boot.ci(boot50_1, type="basic", index = i)$basic[5]
}
Konfi100 <- matrix(NA,6,2)
rownames(Konfi100) <- c("Intercept","Geschlecht","Alter","Minderheit","Fremdenfeindlich","IQ")
colnames(Konfi100) <- c("2,5%","97,5%")
for(i in 1:6)
{
    Konfi100[i,1] <- boot.ci(boot100_1, type="basic", index = i)$basic[4]
```

```r
    Konfi100[i,2] <- boot.ci(boot100_1, type="basic", index = i)$basic[5]
}
Konfi1000 <- matrix(NA,6,2)
rownames(Konfi1000) <- c("Intercept","Geschlecht","Alter","Minderheit","Fremdenfeindlich","IQ")
colnames(Konfi1000) <- c("2,5%","97,5%")
for(i in 1:6)
{
    Konfi1000[i,1] <- boot.ci(boot1000_1, type="basic", index = i)$basic[4]
    Konfi1000[i,2] <- boot.ci(boot1000_1, type="basic", index = i)$basic[5]
}
Konfi10000 <- matrix(NA,6,2)
rownames(Konfi10000) <- c("Intercept","Geschlecht","Alter","Minderheit","Fremdenfeindlich","IQ")
colnames(Konfi10000) <- c("2,5%","97,5%")
for(i in 1:6)
{
    Konfi10000[i,1] <- boot.ci(boot10000_1, type="basic", index = i)$basic[4]
    Konfi10000[i,2] <- boot.ci(boot10000_1, type="basic", index = i)$basic[5]
}
Konfi50
#>                        2,5%      97,5%
#> Intercept        22.5184241 36.4133808
#> Geschlecht        3.6848169  7.9463833
#> Alter             0.1190041  0.2165262
#> Minderheit       -9.3443496 -3.8251713
#> Fremdenfeindlich  8.9786910  9.9112550
#> IQ               -0.4605168 -0.3306015
Konfi100
#>                        2,5%      97,5%
#> Intercept        22.4812933 37.3189333
#> Geschlecht        3.3121834  7.2667820
#> Alter             0.1120515  0.2515150
#> Minderheit       -9.7259934 -3.3638260
#> Fremdenfeindlich  9.0042067  9.6819829
#> IQ               -0.4611898 -0.3343833
Konfi1000
#>                        2,5%      97,5%
#> Intercept        22.6128829 37.2159550
#> Geschlecht        3.7553528  7.7237071
#> Alter             0.1202946  0.2370037
#> Minderheit       -9.5232719 -3.3502610
#> Fremdenfeindlich  8.9929961  9.7164341
#> IQ               -0.4654710 -0.3394534
Konfi10000
#>                        2,5%      97,5%
#> Intercept        22.8489632 36.9531461
#> Geschlecht        3.7826102  7.6931125
#> Alter             0.1224300  0.2376156
#> Minderheit       -9.4877229 -3.5712765
#> Fremdenfeindlich  8.9888542  9.7058668
#> IQ               -0.4635636 -0.3408012
```

c)

```
confint(vergleich,level=0.95)
#>                        2.5 %     97.5 %
#> (Intercept)       22.6034487 36.8732397
#> Geschlecht         3.7795957  7.7318438
#> Alter              0.1212600  0.2417933
#> Minderheit       -10.1898944 -2.9618306
#> Fremdenfeindlich   9.0271567  9.6725195
#> IQ                -0.4609559 -0.3417435
Konfi50
#>                        2,5%      97,5%
#> Intercept        22.5184241 36.4133808
#> Geschlecht        3.6848169  7.9463833
#> Alter             0.1190041  0.2165262
#> Minderheit       -9.3443496 -3.8251713
#> Fremdenfeindlich  8.9786910  9.9112550
#> IQ               -0.4605168 -0.3306015
Konfi100
#>                        2,5%      97,5%
#> Intercept        22.4812933 37.3189333
#> Geschlecht        3.3121834  7.2667820
#> Alter             0.1120515  0.2515150
#> Minderheit       -9.7259934 -3.3638260
#> Fremdenfeindlich  9.0042067  9.6819829
#> IQ               -0.4611898 -0.3343833
Konfi1000
#>                        2,5%      97,5%
#> Intercept        22.6128829 37.2159550
#> Geschlecht        3.7553528  7.7237071
#> Alter             0.1202946  0.2370037
#> Minderheit       -9.5232719 -3.3502610
#> Fremdenfeindlich  8.9929961  9.7164341
#> IQ               -0.4654710 -0.3394534
Konfi10000
#>                        2,5%      97,5%
#> Intercept        22.8489632 36.9531461
#> Geschlecht        3.7826102  7.6931125
#> Alter             0.1224300  0.2376156
#> Minderheit       -9.4877229 -3.5712765
#> Fremdenfeindlich  8.9888542  9.7058668
#> IQ               -0.4635636 -0.3408012
```

Die Unterschiede der Konfidenzintervalle durch Bootstrapping mit dem Konfidenzintervall der linearen Regression sind verschwindend gering, deshalb ist eine Empfehlung schwierig. Wir würden uns für ein bootstrapping mit Replikationsgröße 10000 entscheiden, da hier die Unterschiede wohl am geringsten sind.

Interessanterweise ist bei boot_100 das Konfidenzintervall für den Parameterschätzer der Kovariaten "Geschlecht" am unterschiedlichsten zum Konfidenzinterall der linearen Regression. Im Gegensatz zu den anderen Replikationsgrößen ist dieses Intervall nach unten verschoben. Dies schneidet sich mit der in Aufgabe 2a) aufgetretetenen Anomalie.

d)

```r
Konfi1000perc <- matrix(NA,6,2)
rownames(Konfi1000perc) <- c("Intercept","Geschlecht","Alter","Minderheit","Fremdenfeindlich","IQ")
colnames(Konfi1000perc) <- c("2,5%","97,5%")
for(i in 1:6)
{
    Konfi1000perc[i,1] <- boot.ci(boot1000_1, type="perc", index = i)$perc[4]
    Konfi1000perc[i,2] <- boot.ci(boot1000_1, type="perc", index = i)$perc[5]
}
Konfi1000bca <- matrix(NA,6,2)
rownames(Konfi1000bca) <- c("Intercept","Geschlecht","Alter","Minderheit","Fremdenfeindlich","IQ")
colnames(Konfi1000bca) <- c("2,5%","97,5%")
for(i in 1:6)
{
    Konfi1000bca[i,1] <- boot.ci(boot1000_1, type="bca", index = i)$bca[4]
    Konfi1000bca[i,2] <- boot.ci(boot1000_1, type="bca", index = i)$bca[5]
}
Konfi1000
#>                        2,5%      97,5%
#> Intercept        22.6128829 37.2159550
#> Geschlecht        3.7553528  7.7237071
#> Alter             0.1202946  0.2370037
#> Minderheit       -9.5232719 -3.3502610
#> Fremdenfeindlich  8.9929961  9.7164341
#> IQ               -0.4654710 -0.3394534
Konfi1000perc
#>                        2,5%      97,5%
#> Intercept        22.2607334 36.8638055
#> Geschlecht        3.7877324  7.7560868
#> Alter             0.1260496  0.2427587
#> Minderheit       -9.8014640 -3.6284532
#> Fremdenfeindlich  8.9832421  9.7066801
#> IQ               -0.4632460 -0.3372284
Konfi1000bca
#>                        2,5%      97,5%
#> Intercept        22.3297795 36.8833535
#> Geschlecht        3.7402325  7.7323257
#> Alter             0.1237283  0.2353549
#> Minderheit       -9.8039293 -3.6347743
#> Fremdenfeindlich  8.9866019  9.7108741
#> IQ               -0.4700202 -0.3414887
```

Wir sehen hier keine großen Unterschiede in den Konfidenzintervallen.