t-Test in R

MD MAHFUJUL KARIM SHEIKH

2025-02-12

Contents

Required Packages

install.packages("reshape2")

library(reshape2)

```r
library(ggplot2)
library(reshape2)
library(car)
library(ggpubr)
```

One sample t-test

Manual Calculation

```
set.seed(123)
sample_data <- rnorm(30, mean = 50, sd = 10)
mean(sample_data)
```

```
## [1] 49.52896
```

```
sd(sample_data)
```

```
## [1] 9.810307
```

```
#mu_0=55
SE <- sd(sample_data)/sqrt(length(sample_data))
t_cal <- (mean(sample_data) - 55)/SE
print(t_cal)
```

```
## [1] -3.054553
```

```
t_crit <- qt(0.05/2, df = length(sample_data)-1, lower.tail=TRUE)
t_crit
```

```
## [1] -2.04523
```

```
t_cal <- t_crit
```

using function———-

```
t.test(sample_data, mu=55)
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -3.0546, df = 29, p-value = 0.004797
## alternative hypothesis: true mean is not equal to 55
## 95 percent confidence interval:
##  45.86573 53.19219
## sample estimates:
## mean of x
##  49.52896
```

```r
t.test(sample_data, mu=55, conf.level=0.99)
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -3.0546, df = 29, p-value = 0.004797
## alternative hypothesis: true mean is not equal to 55
## 99 percent confidence interval:
##  44.59198 54.46595
## sample estimates:
## mean of x
##  49.52896
```


```r
t.test(sample_data, mu=55, conf.level=0.99)
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -3.0546, df = 29, p-value = 0.004797
## alternative hypothesis: true mean is not equal to 55
## 99 percent confidence interval:
##  44.59198 54.46595
## sample estimates:
## mean of x
##  49.52896
```


Hypothesis:
$H_0 : \mu = \mu_0$
$H_1 : \mu \neq \mu_0$

```r
# mu_0 = 55
SE <- sd(sample_data)/sqrt(length(sample_data))
t_cal <- (mean(sample_data) - 55)/SE
print(t_cal)
```

```
## [1] -3.054553
```

```r
t_crit <- qt(0.05/2, df = length(sample_data)-1, lower.tail = TRUE)
t_crit
```

```
## [1] -2.04523
```

```r
t_cal <= t_crit  # decision: reject null
```

```
## [1] TRUE
```

```r
pt(t_cal, df = 29, lower.tail = TRUE) + pt(-t_cal, df = 29, lower.tail = FALSE)
```

```
## [1] 0.0047971
```

```r
mean(sample_data)+c(-1, 1)* abs(t_crit)*SE
```

```
## [1] 45.86573 53.19219
```

Calculating confidence interval:

```r
mean(sample_data) + c(-1, 1) * abs(t_crit) * SE
```

```
## [1] 45.86573 53.19219
```

Using Function

Generate sample data:

```r
set.seed(123)
sample_data <- rnorm(30, mean = 50, sd = 10)
```

Testing normality using Shapiro-Wilk test:

```r
shapiro.test(sample_data)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample_data
## W = 0.97894, p-value = 0.7966
```

H_0 : Data follows normal distribution.
H_1 : Data does not follow normal distribution.

Since the p-value is greater than the level of significance ($\alpha = 0.05$), we do not have enough statistical evidence to reject the null hypothesis.

One-sample t-test:

Hypothesis:
$H_0 : \mu = 50$
$H_1 : \mu \neq 50$

```r
t.test(sample_data, mu = 50)
```

```
##
##  One Sample t-test
##
## data:  sample_data
```

```
## t = -0.26299, df = 29, p-value = 0.7944
## alternative hypothesis: true mean is not equal to 50
## 95 percent confidence interval:
##  45.86573 53.19219
## sample estimates:
## mean of x
##  49.52896
```

Using function, perform the two tailed t-test:

```r
t.test(sample_data, mu = 53, conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -1.9379, df = 29, p-value = 0.06242
## alternative hypothesis: true mean is not equal to 53
## 95 percent confidence interval:
##  45.86573 53.19219
## sample estimates:
## mean of x
##  49.52896
```

If p-value < 0.05, then reject null. Decision is not rejected.

```r
t.test(sample_data, mu = 53, conf.level = 0.99)
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -1.9379, df = 29, p-value = 0.06242
## alternative hypothesis: true mean is not equal to 53
## 99 percent confidence interval:
##  44.59198 54.46595
## sample estimates:
## mean of x
##  49.52896
```

If p-value < 0.01, then reject null. Decision is not rejected.

```r
t.test(sample_data, mu = 53, conf.level = 0.90)
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -1.9379, df = 29, p-value = 0.06242
## alternative hypothesis: true mean is not equal to 53
## 90 percent confidence interval:
```

```
##  46.48564 52.57228
## sample estimates:
## mean of x
##  49.52896
```

If p-value < 0.10, then reject null. Decision is rejected.

Hypothesis:
$H_0 : \mu <= 40$
$H_1 : \mu > 40$

```r
t.test(sample_data, mu = 40, alternative = "greater")
```

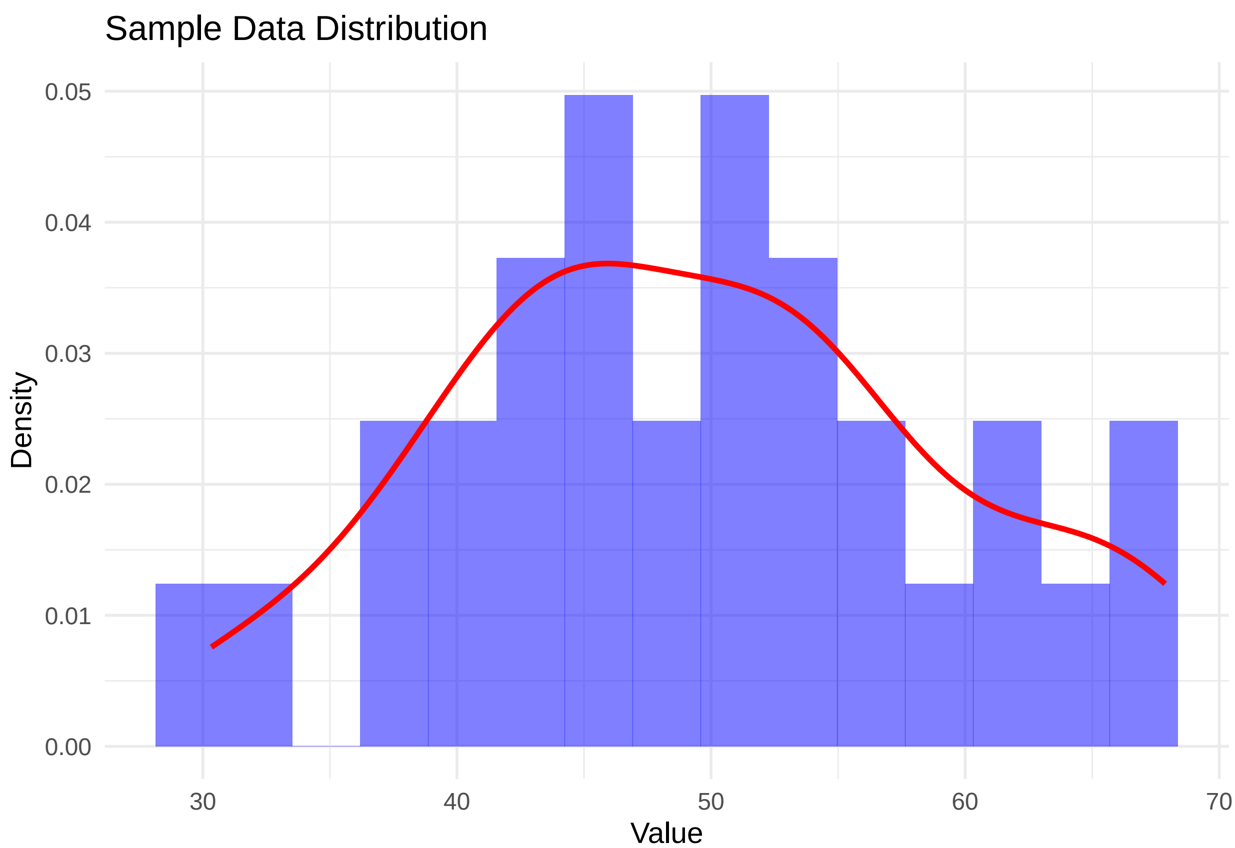
```
##
##  One Sample t-test
##
## data:  sample_data
## t = 5.3201, df = 29, p-value = 5.21e-06
## alternative hypothesis: true mean is greater than 40
## 95 percent confidence interval:
##  46.48564      Inf
## sample estimates:
## mean of x
##  49.52896
```

Hypothesis:
$H_0 : \mu >= 58$
$H_1 : \mu < 58$

```r
t.test(sample_data, mu = 58, alternative = "less")
```

```
##
##  One Sample t-test
##
## data:  sample_data
## t = -4.7295, df = 29, p-value = 2.689e-05
## alternative hypothesis: true mean is less than 58
## 95 percent confidence interval:
##      -Inf 52.57228
## sample estimates:
## mean of x
##  49.52896
```

```r
ggplot(data.frame(Value = sample_data), aes(x = Value)) +
  geom_histogram(aes(y = after_stat(density)), bins = 15, fill = "blue", alpha = 0.5) +
  geom_density(color = "red", linewidth = 1) +
  labs(title = "Sample Data Distribution", x = "Value", y = "Density") +
  theme_minimal()
```



Two-samples t-test

Use built-in dataset: mtcars (comparing mpg for automatic vs manual cars):

```
data(mtcars)
```

Split into two groups based on transmission type":

```
auto_mpg <- mtcars$mpg[mtcars$am == 0]     # Automatic
manual_mpg <- mtcars$mpg[mtcars$am == 1]   # Manual
```

H_0: Automatic cars and manual cars have equal average mpg.
H_1: Automatic cars and manual cars have unequal average mpg.

```
mean(auto_mpg)
```

```
## [1] 17.14737
```

```
mean(manual_mpg)
```

```
## [1] 24.39231
```

```r
var(auto_mpg)
```

```
## [1] 14.6993
```

```r
var(manual_mpg)
```

```
## [1] 38.02577
```

Normality test for both groups:

```r
shapiro.test(auto_mpg)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  auto_mpg
## W = 0.97677, p-value = 0.8987
```

```r
shapiro.test(manual_mpg)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  manual_mpg
## W = 0.9458, p-value = 0.5363
```

Check variance homogeneity (Levene's test):

```r
leveneTest(mpg ~ factor(am), data = mtcars, center = "mean")
```

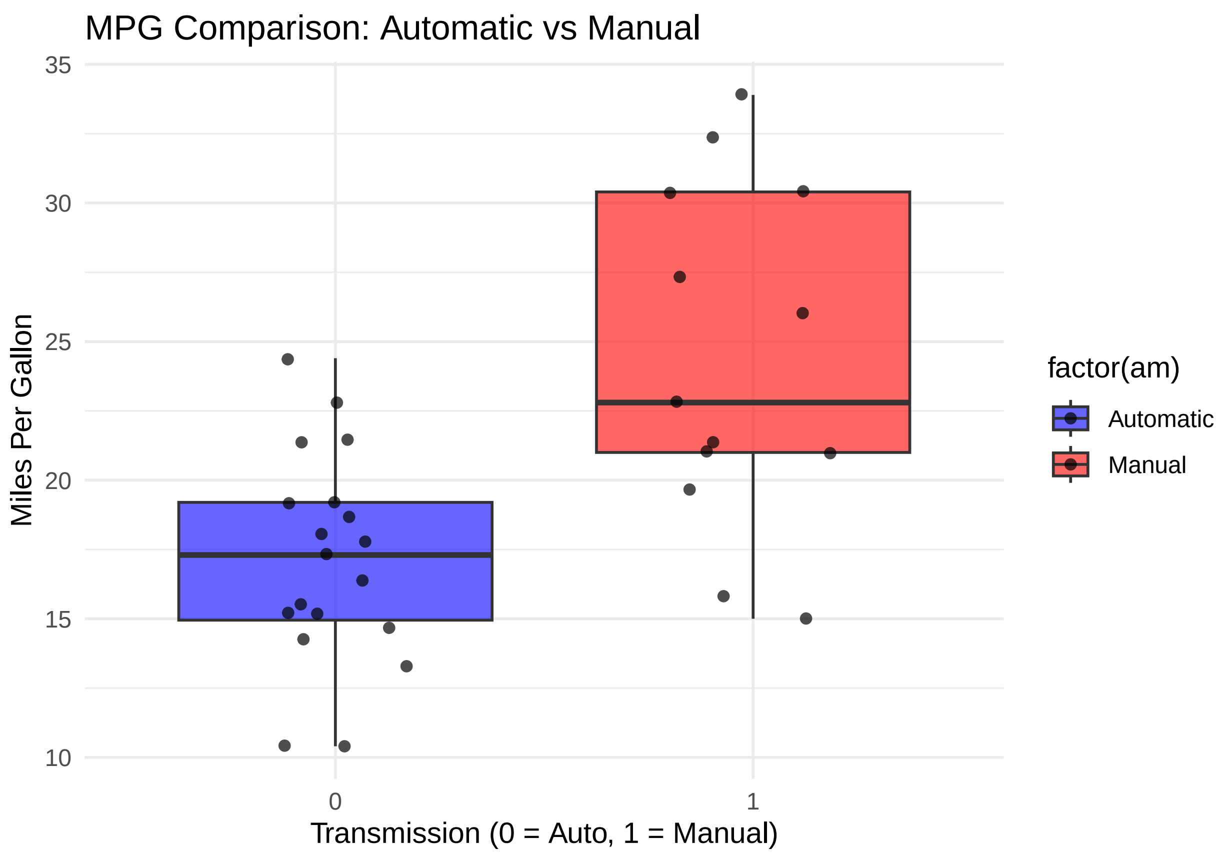
```
## Levene's Test for Homogeneity of Variance (center = "mean")
##       Df F value  Pr(>F)
## group  1   5.921 0.02113 *
##       30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Perform two-sample t-test:

```r
t.test(auto_mpg, manual_mpg, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  auto_mpg and manual_mpg
## t = -3.7671, df = 18.332, p-value = 0.001374
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.280194  -3.209684
## sample estimates:
## mean of x mean of y
##  17.14737  24.39231
```

```r
# Visualize the data
ggplot(mtcars, aes(x = factor(am), y = mpg, fill = factor(am))) +
  geom_boxplot(alpha = 0.6) +
  geom_jitter(width = 0.2, alpha = 0.7) +
  labs(title = "MPG Comparison: Automatic vs Manual",
       x = "Transmission (0 = Auto, 1 = Manual)",
       y = "Miles Per Gallon") +
  scale_fill_manual(values = c("blue", "red"),
                    labels = c("Automatic", "Manual")) +
  theme_minimal()
```



Paired samples t-test

Generate 30 observations:

```r
set.seed(123)
before <- round(rnorm(30, mean = 300, sd = 10), 0)
summary(before)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   280.0   293.2   299.5   299.6   305.0   318.0
```

```r
# after a course
after <- before + round(rnorm(30, mean = 5, sd = 5), 0) # Simulating a increase
summary(after)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   283.0   298.2   305.0   305.5   312.2   325.0
```

```r
df <- data.frame(
  ID = 1:30,
  Before = before,
  After = after
)
```

```r
mean(after) - mean(before)
```

```
## [1] 5.933333
```

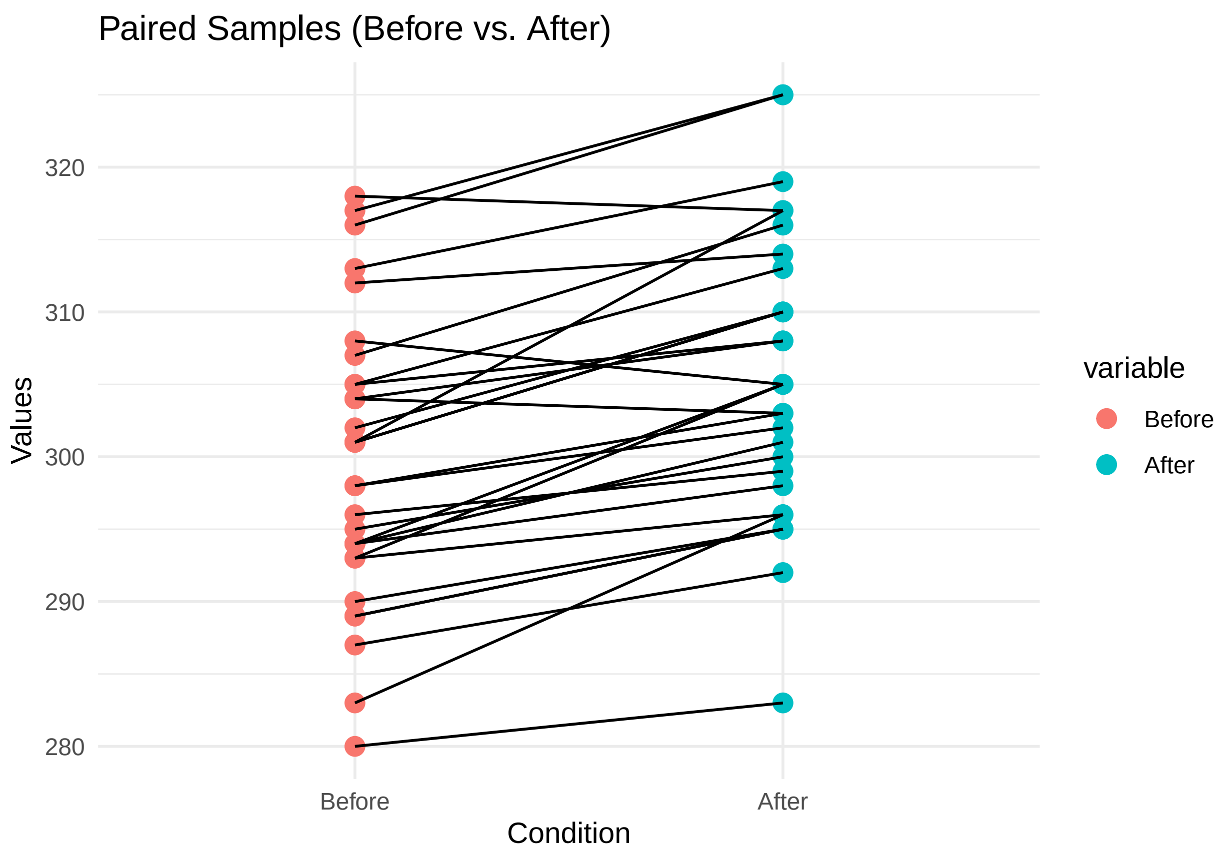
H_0 : Before and after the course the true GRE average score of the students stays the same.
H_1 : Before and after the course the true GRE average score of the students does not remain the same.

Perform Paired t-test

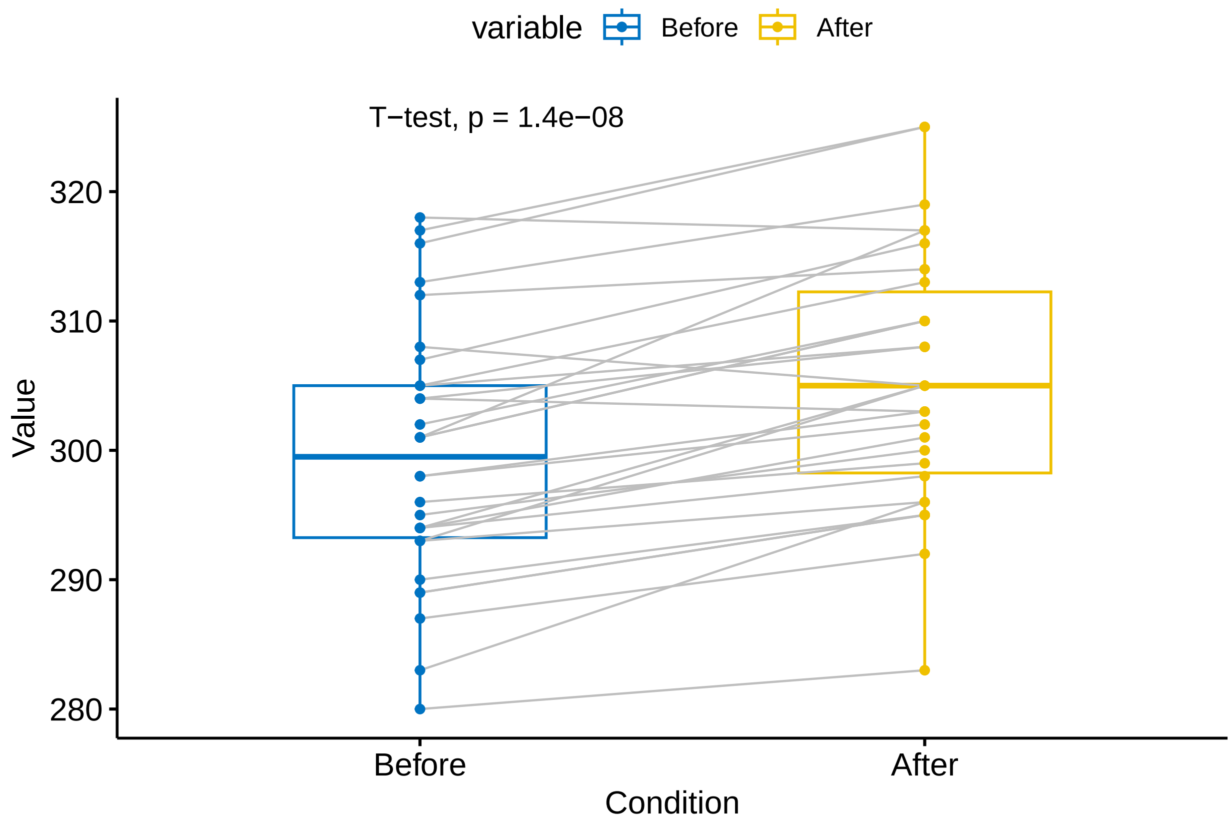
```r
t.test(x = after, y = before, paired = TRUE, alternative = "two.sided")
```

```
##
##  Paired t-test
##
## data:  after and before
## t = 7.7811, df = 29, p-value = 1.398e-08
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  4.373779 7.492888
## sample estimates:
## mean difference
##        5.933333
```

```r
# Visualize the differences
df <- data.frame(
  ID = 1:30,
  Before = before,
  After = after
  )
df_long <- melt(df, id.vars = "ID")
ggplot(df_long, aes(x = variable, y = value, group = ID)) +
  geom_point(aes(color = variable), size = 3) +
  geom_line() +
  labs(title = "Paired Samples (Before vs. After)",
       y = "Values", x = "Condition") +
  theme_minimal()
```




```
ggpaired(df_long,
         x = "variable",
         y = "value",
         color = "variable",
         line.color = "gray",
         line.size = 0.4,
         palette = "jco") +
  stat_compare_means(paired = TRUE, method = "t.test")
```


variable
Before
After
T−test, p = 1.4e−08
320
310
300
290
280
Value
Before
After
Condition