Data Visualization and Modelling: Exercise 1

```
# Data Visualization: Exercise 1 ----------------------------------------
# Author: Alejandro Jiménez  Rico

library('combinat')
```

```
## Warning: package 'combinat' was built under R version 3.3.2
```

```
##
## Attaching package: 'combinat'
```

```
## The following object is masked from 'package:utils':
##
##     combn
```

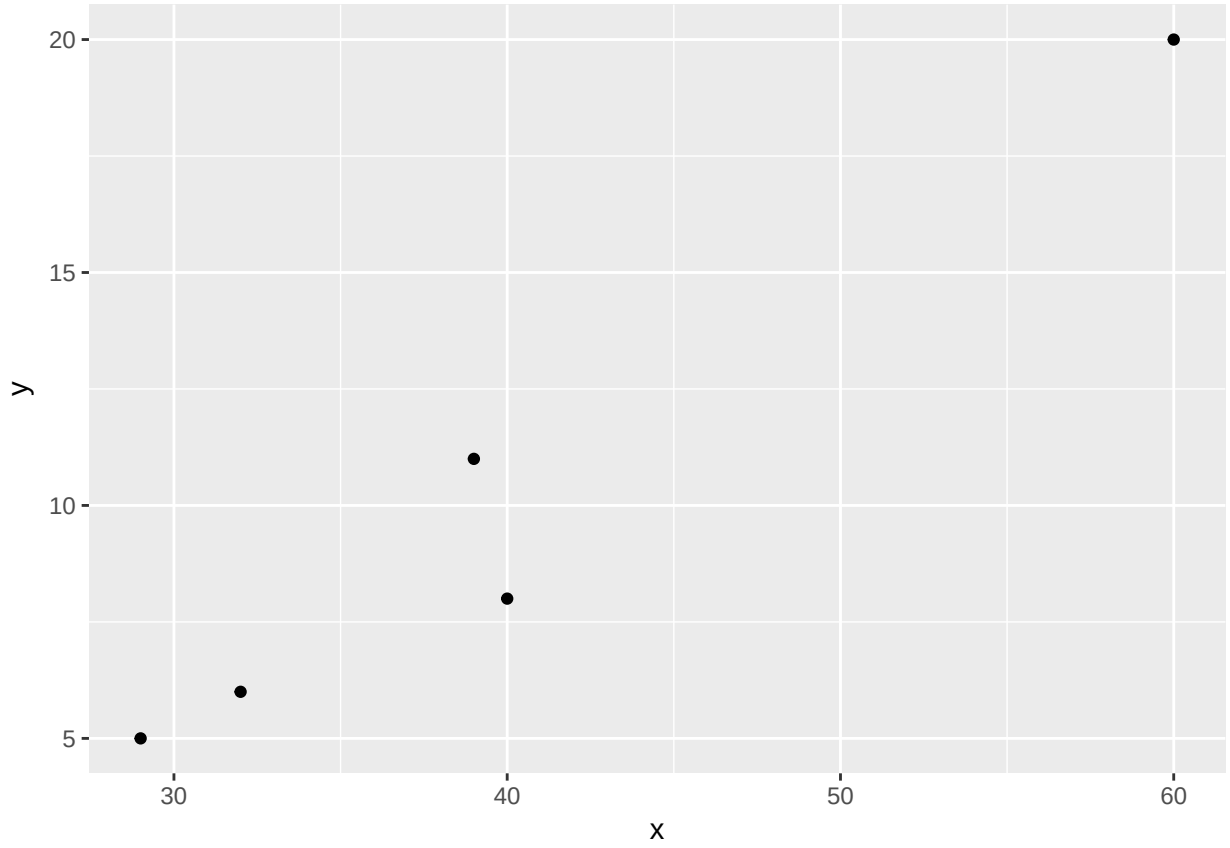
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

Volume of Air Experiment

The first row is chest circumference (in inches) of five subjects. Let us call this X. The second row is the respective total volumes of air that can be breathed in and out in one minute (in liters) for the same five subjects. Let us call this Y.

```
x <- c(39,29, 60, 40, 32)
y <- c(11, 5, 20, 8, 6)

# Taking a look at the data
ggplot() +
    geom_point(aes(x=x, y=y))
```



First of all, we must create a bunch of variables in order to perform the permutation test. We define the maximum error (alpha) we are able to tolerate.

```
spearson <- c()
sim.spear <- c()
df <- as.data.frame(x)
df$y <- y
df$dx <- 0
df$dy <- 0
alpha <- 0.05
```

Now, we compute the coefficients of our data set. To make this exercise more didactic, I have decided to compute the coefficients manually and try not to use the standard functions in R.

```
# Compute Pearson Coefficient
cov <- cov(x,y)
sdx <- sd(x)
sdy <- sd(y)
pearson <- cov/(sdx*sdy)

# Clear variables
rm(cov,sdx,sdy)

# Ranking values
for(i in 1:length(x)){
    a <- sort(x, decreasing = TRUE)[i]
    df[df$x == a, ]$dx <- i
    a <- sort(y, decreasing = TRUE)[i]
```

```
    df[df$y == a, ]$dy <- i
}

# Compute difference of ranks
df$diff <- df$dx - df$dy
df$diff2 <- df$diff^2

# Compute Spearman Coefficient
t <- table(df$diff)
t <- t[names(t) == 0]
t <- as.data.frame(t)
n <- length(df$diff2) - t$t
n <- length(df$diff2)
spearman <- 1 - 6*(sum(df$diff2))/(n^3-n)

# Clear variables
rm(a,i,t,n)
```

Now we perform the permutation test. The philosophy of this test is to generate random combinations of our actual data set and compute its coefficients; the general idea is to measure wether these simulated coefficients are similar enough to our original coefficients or not.

```
# Permutation test
cnt <- 0
cnt2 <- 0
w <- permn(y)
for (i in 1:length(w)){
    y <- as.vector(w[[i]])

    # Compute Pearson and Spearman Coefficients from randomized data
    cov <- cov(x,y)
    sdx <- sd(x)
    sdy <- sd(y)
    spearson[i] <- cov/(sdx*sdy)
    if (spearson[i] > pearson){cnt = cnt + 1}

    sspearman <- cor(x,y, method='spearman')
    sim.spear[i] <- sspearman
    if (sim.spear[i] > spearman){cnt2 = cnt2+1}
}

# Measure pvalue
pvalue1 <- cnt/length(w)
pvalue2 <- cnt2/length(w)
```

Now we have compared our coefficients with those obtained from randomized data. Taking a look at the distribution of the coefficients from the randomized data we can measure the oddness of our coefficients and distinguish them from the simulated ones. The question here to be answered is the following: 'Do our coefficients are odd enough to be distinguished from mere noise?' Or more technically: 'Are our coefficients stasticially different from those obtained from randomized data set?'

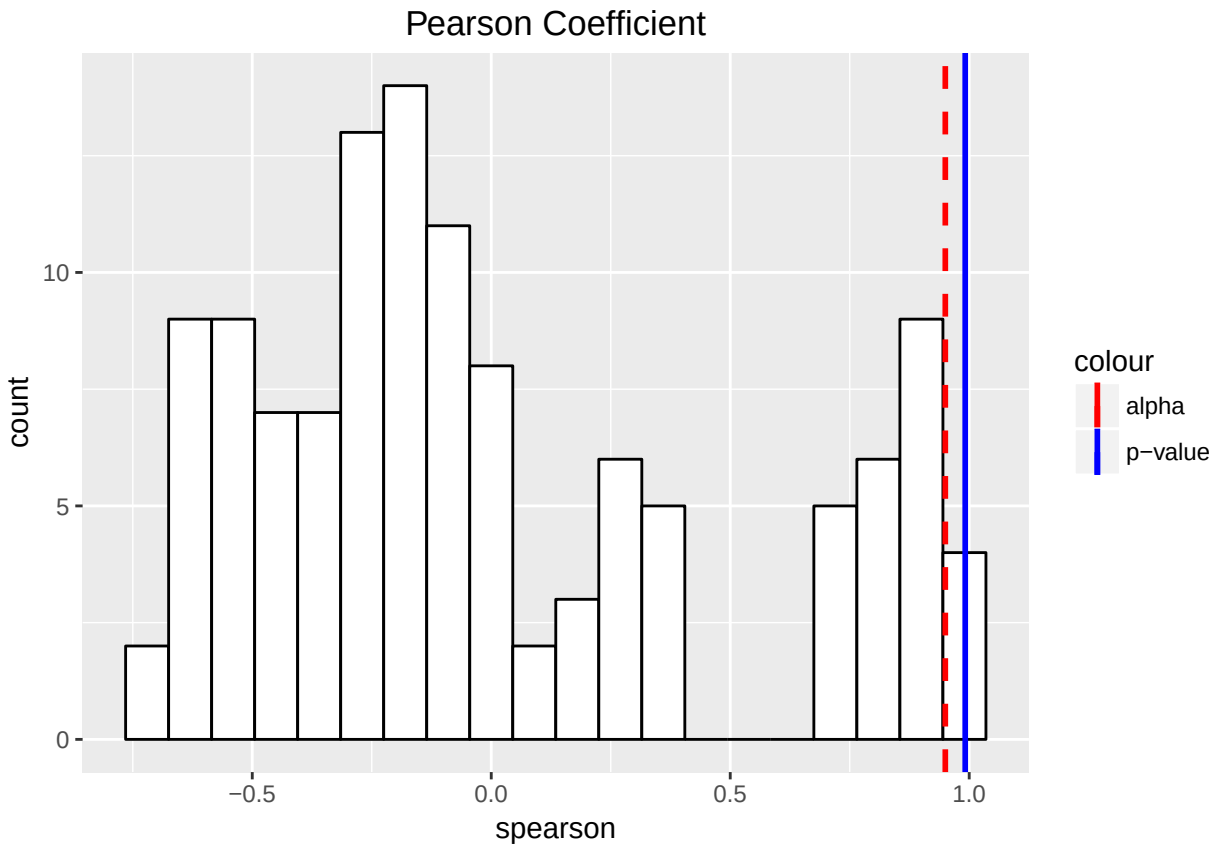
Why do we do it this way? Because a randomized data set delivers coefficients whose means tend to zero. Henceforth, if we can reject that our coefficients are equal to 0, we just need to reject that they are equal to coefficients obtained from random generated data sets.

```
pearson.hist <- ggplot() +
    geom_histogram(aes(x=spearson), binwidth = 0.09, colour='black', fill='white') +
    geom_vline(aes(xintercept=quantile(1-pvalue1), colour='p-value'),  size=1, linetype=1, show.legend='
    geom_vline(aes(xintercept=quantile(1-alpha), colour='alpha'), size=1, linetype=2, show.legend=TRUE)
    scale_colour_manual(values=c('red','blue')) +
    ggtitle('Pearson Coefficient') +
    theme(plot.title = element_text(hjust = 0.5))
```

The point here is to compare the pvalue obtained with the alpha estabilshed at the beginning of the experiment. If the pvalue is lower than the alpha, we can reject the *Null Hypothesis*. In this experiment, the *Null Hypothesis* is that the coefficient is equal to 0.

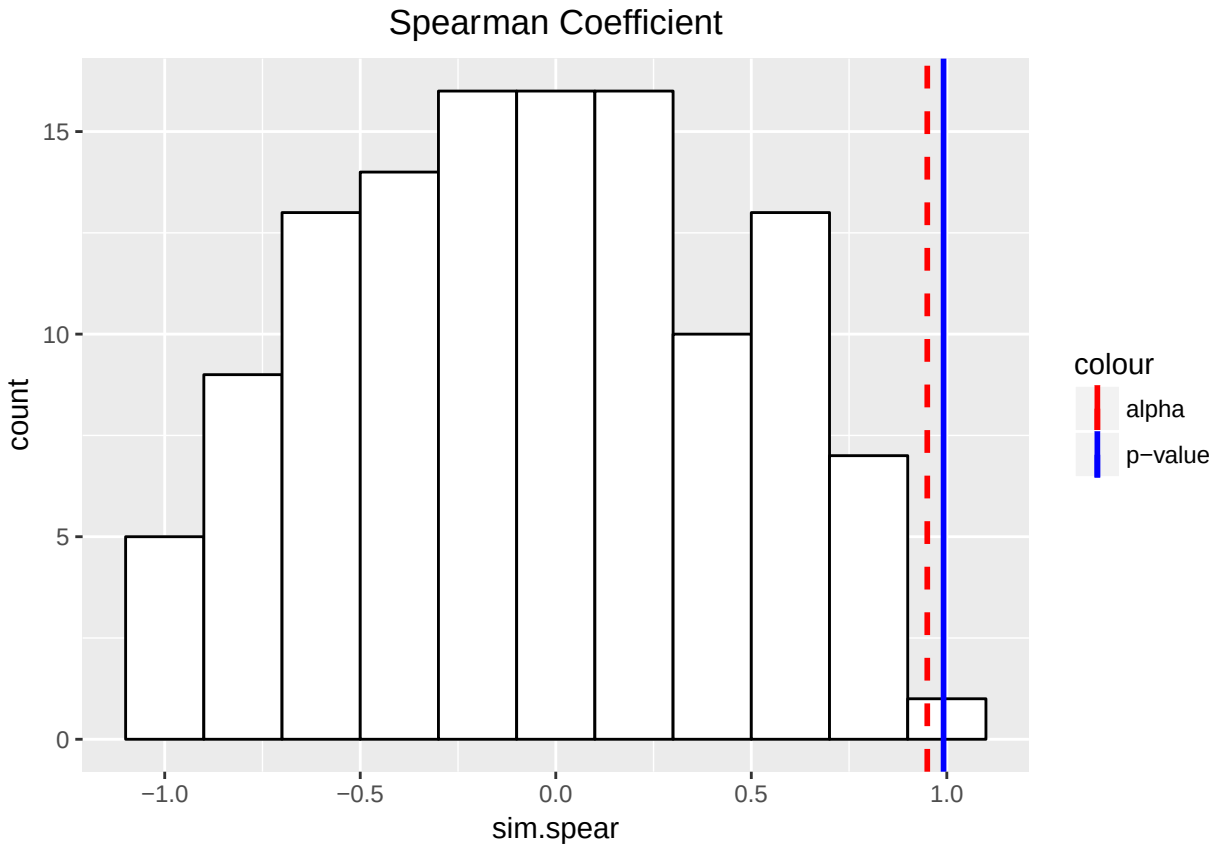
Being $\alpha = 0.05$ and pvalue $= 0.0083333$, we can say, with a 95% confidence, that the Pearson Coefficient is different from 0.

```
spearman.hist <- ggplot() +
    geom_histogram(aes(x=sim.spear), binwidth = 0.2, colour='black', fill='white') +
    geom_vline(aes(xintercept=quantile(1-pvalue2), colour='p-value'),  size=1, linetype=1, show.legend='
    geom_vline(aes(xintercept=quantile(1-alpha), colour='alpha'), size=1, linetype=2, show.legend=TRUE)
    scale_colour_manual(values=c('red','blue')) +
    ggtitle('Spearman Coefficient') +
    theme(plot.title = element_text(hjust = 0.5))

pearson.hist
```



Analogously, being $\alpha = 0.05$ and pvalue $= 0.0083333$, we can say, with a 95% confidence, that the Spearman Coefficient is different from 0.

```
spearman.hist
```




```
# Clear all variables
rm(list=ls(all=TRUE))
```

Weight Experiment

These are the increments of weight recorded in an experiment where a new additive has been added to a standard compound feed:

```
standard <- c(2.5,3.4,2.9,4.1,5.3,3.4,1.9,3.3,1.8)
additive <- c(3.5,6.3,4.2,4.3,3.8,5.7,4.4)
```

First of all, we must create a bunch of variables in order to perform the permutation test. We define the maximum error (alpha) we are able to tolerate.

```
nr <- 100000 # Number of rearrangements
st <- numeric(nr)

n1 <- length(standard)
n2 <- length(additive)
total <- n1+n2
alpha <- 0.05
```

Now, we compute the statistics of our data set. Considering that we want to test whether or not the mean of the *Additive* observations is greater than the mean of *Standard*, the statistic is going to be the difference of

the means between *Standard* and *Additive*. And the test will measure if its value is statistically different from 0.

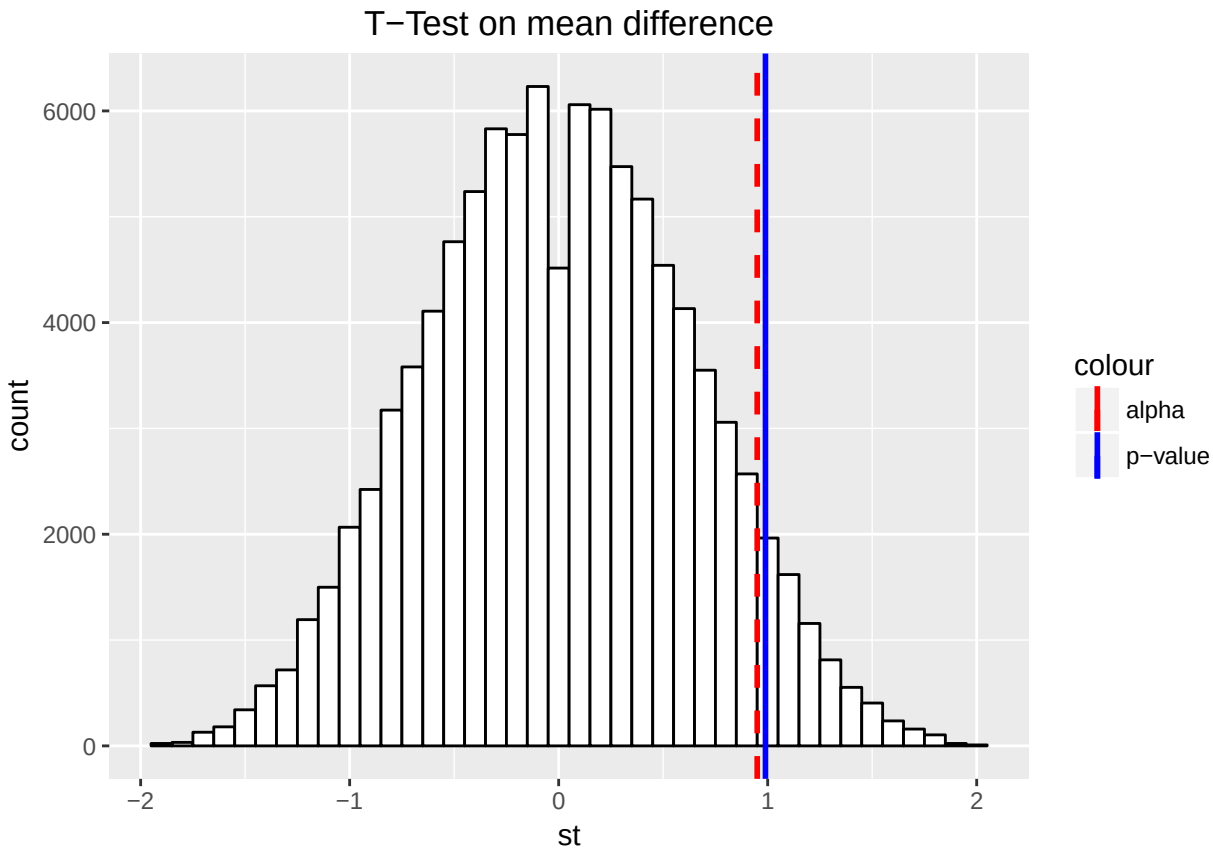
```
m.st <- mean(standard)
m.add <- mean(additive)

sttrue <- m.add - m.st
cnt <- 0
# Put both sets of observations in a single vector
vect <- c(standard, additive)
```

Conceptually similar to

```
# Perform permutations and compare means
for (i in 1:nr){
    d <- sample(vect, total)
    s.st<- d[1:n1]
    a <- n1+1
    s.add <- d[a:total]
    st[i] <- mean(s.add)-mean(s.st)
    if(st[i] > sttrue){cnt = cnt + 1}
}

pvalue <- cnt/nr
```

The pvalue that we have obtained from the permutation test is 0.01044. So we can reject the *Null Hypothesis*. In other words, we can say, with 95% confidence, that the mean of *Additive* is greater than the mean of *Standard*.

```
# Plot histogram
histogram <- ggplot() +
    geom_histogram(aes(x=st), binwidth = 0.1, colour='black', fill='white') +
    geom_vline(aes(xintercept=quantile(1-pvalue), colour='p-value'),  size=1, linetype=1, show.legend=TR
    geom_vline(aes(xintercept=quantile(1-alpha), colour='alpha'), size=1, linetype=2, show.legend=TRUE)
    scale_colour_manual(values=c('red','blue')) +
    ggtitle('T-Test on mean difference') +
    theme(plot.title = element_text(hjust = 0.5))

histogram
```



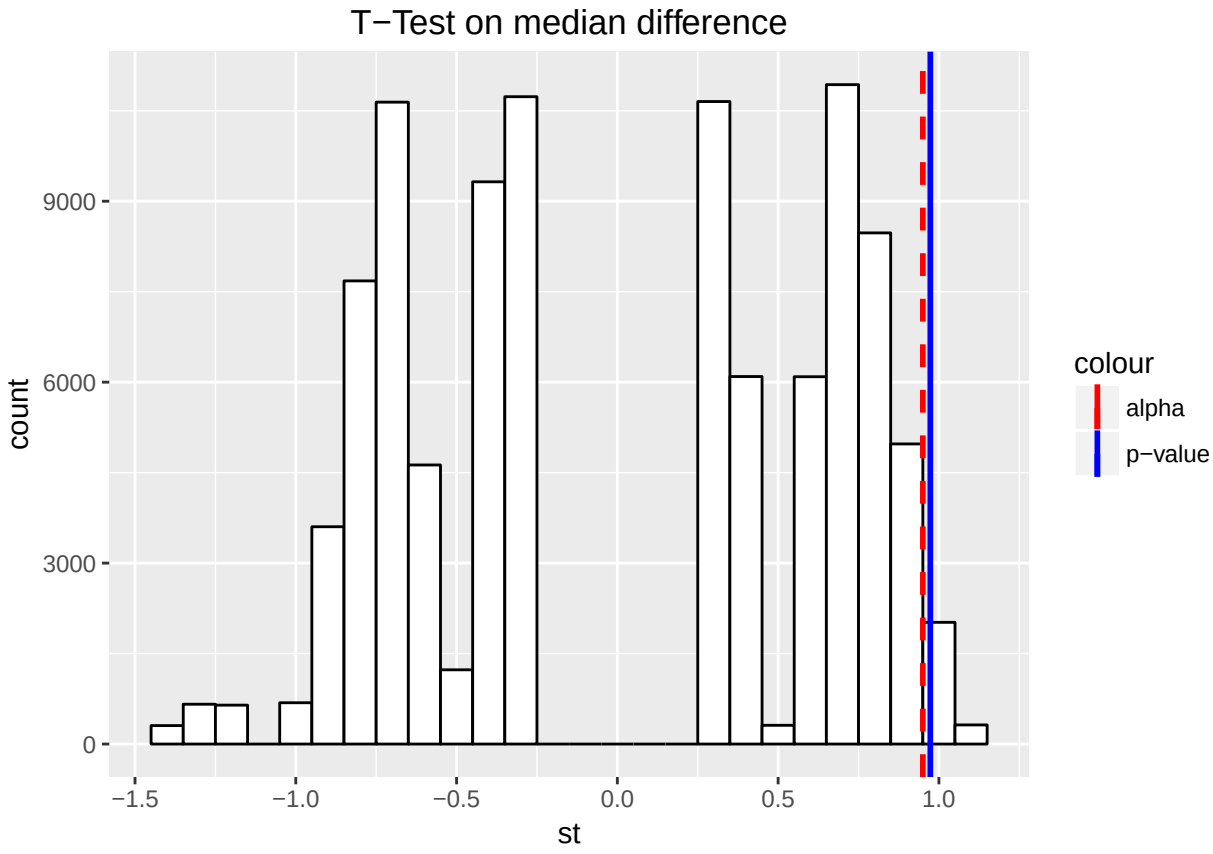
Analogously, we can repeat the experiment looking for different measures that can help us to understand the data. Another measure we can consider is the **median**. So we repeat the whole experiment measuring the median difference.

```
sttrue <- median(additive) - median(standard)

# Perform permutation test comparing medians
for (i in 1:nr){
    d <- sample(vect, total)
    s.st<- d[1:n1]
    a <- n1+1
    s.add <- d[a:total]
    st[i] <- median(s.add)-median(s.st)
    if(st[i] > sttrue){cnt = cnt + 1}
}

pvalue <- cnt/nr
```

The pvalue that we have obtained from the permutation test is 0.02674. So we can reject the *Null Hypothesis*. In other words, we can say, with 95% confidence, that the median of *Additive* is greater than the median of *Standard*.

```
# Plot histogram
histogram <- ggplot() +
    geom_histogram(aes(x=st), binwidth = 0.1, colour='black', fill='white') +
    geom_vline(aes(xintercept=quantile(1-pvalue), colour='p-value'),  size=1, linetype=1, show.legend=TR
    geom_vline(aes(xintercept=quantile(1-alpha), colour='alpha'), size=1, linetype=2, show.legend=TRUE)
    scale_colour_manual(values=c('red','blue')) +
```

```
    ggtitle('T-Test on median difference') +
    theme(plot.title = element_text(hjust = 0.5))

histogram
```



Conclusion: Is the additive efficient? If the purpose of the additive is to increase weight, **yes**.