Assignment Three

Etana Disasa

11/14/2018

```
library(ggplot2)
library(reshape2)
#' @name assign_vector
#' @param data A vector of data to perform the t-test on.
#' @param n An integer indicating the number of t-tests to perform. Default is 1000
#' @return A data frame in "tall" format
assign_vector <- function(data, n = 1000) {
  # replicate the call to shapiro.test n times to build up a vector of p-values
  p.5 <- replicate(n=n, expr=shapiro.test(sample(my.data, 5, replace=TRUE))$p.value)
  p.10 <- replicate(n=n, expr=shapiro.test(sample(my.data, 10, replace=TRUE))$p.value)
  p.1000 <- replicate(n=n, expr=shapiro.test(sample(my.data, 1000, replace=TRUE))$p.value)
  #' Combine the data into a data frame,
  #' one column for each number of samples tested.
  p.df <- cbind(p.5, p.10, p.1000)
  p.df <- as.data.frame(p.df)
  colnames(p.df) <- c("5 samples","10 samples","1000 samples")
  #' Put the data in "tall" format, one column for number of samples
  #' and one column for the p-value.
  p.df.m <- melt(p.df)
  #' Make sure the levels are sorted correctly.
  p.df.m <- transform(p.df.m, variable = factor(variable, levels = c("5 samples","10 samples","1000 samp
  return(p.df.m)
}
```

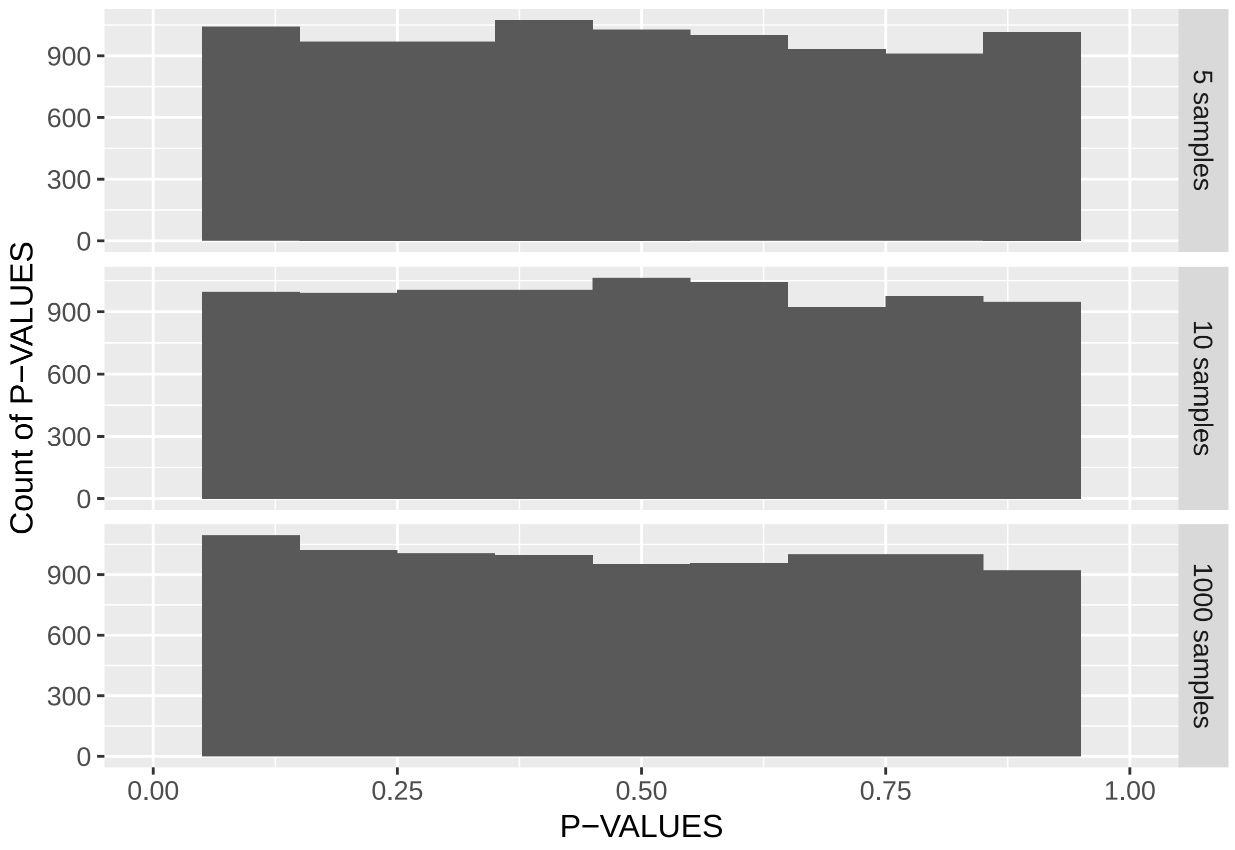
This function assign_vector does the following things. 1) It takes the p-values of a normal distribution of n populations with sample sizes of 5, 10, and 1000. 2) Merges and melts them into a single data frame with two 2 variable; variables(listing which sample size) and value(p-values). 3) Finalizes and creates p.df.m ready to be plotted in graphs.

```
n.rand <- 100000
n.test <- 10000
my.data <- rnorm(n.rand)
p.df.m <- assign_vector(my.data, n = n.test)
```

```
## No id variables; using all as measure variables
```

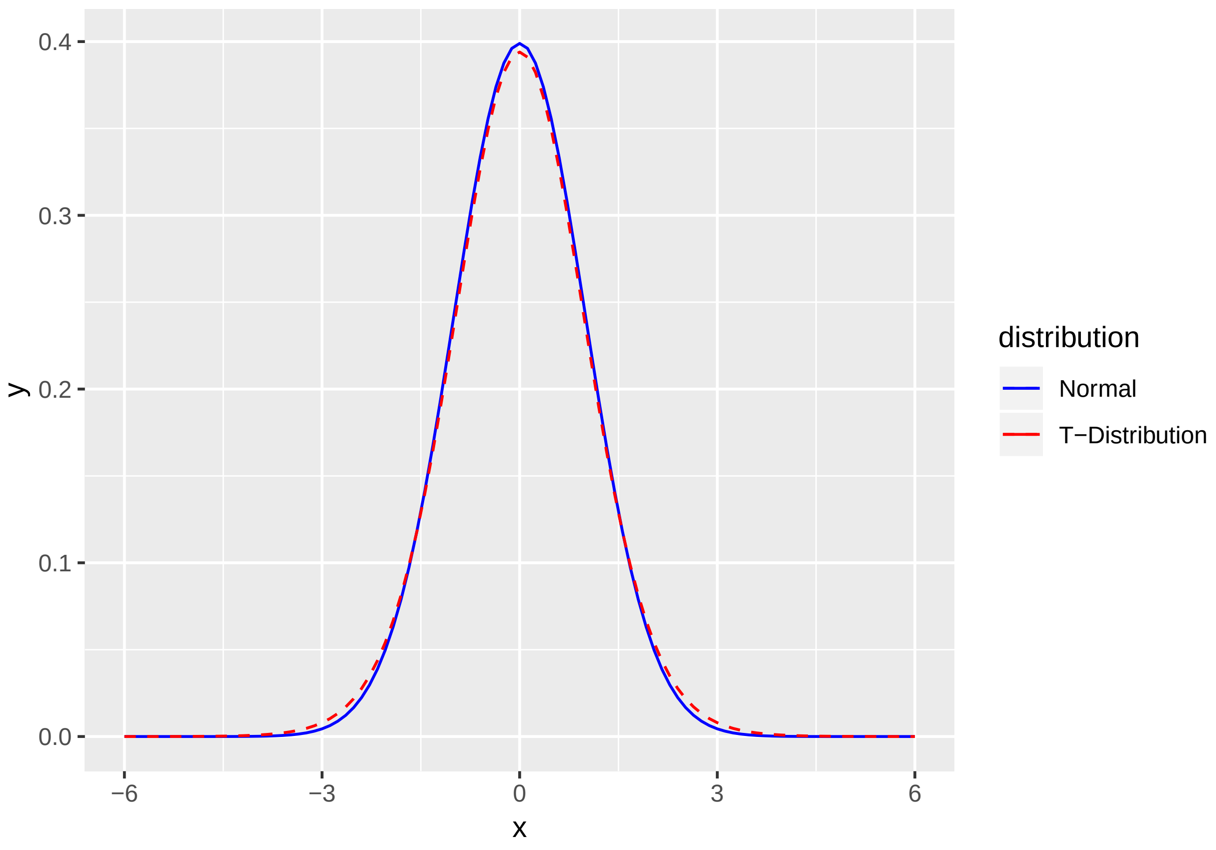
```
ggplot(p.df.m, aes(x = value)) +
  geom_histogram(binwidth = 1/10) +
  facet_grid(facets=variable ~ ., scales="free_y") +
  xlim(0,1) +
  ylab("Count of P-VALUES") +
  xlab("P-VALUES") +
  theme(text = element_text(size = 12))
```

```
## Warning: Removed 6 rows containing missing values (geom_bar).
```



Histogram of p-values for the normal distribution, for sample sizes 5, 10 and 1000.

```
ggplot(NULL, aes(x=x, colour = distribution)) +
  stat_function(fun=dnorm, data = data.frame(x = c(-6,6),
                distribution = factor(1))) +
  stat_function(fun=dt, args = list( df = 20), data = data.frame(x = c(-6,6),
                distribution = factor(2)), linetype = "dashed") +
  scale_colour_manual(values = c("blue","red"), labels = c("Normal","T-Distribution"))
```



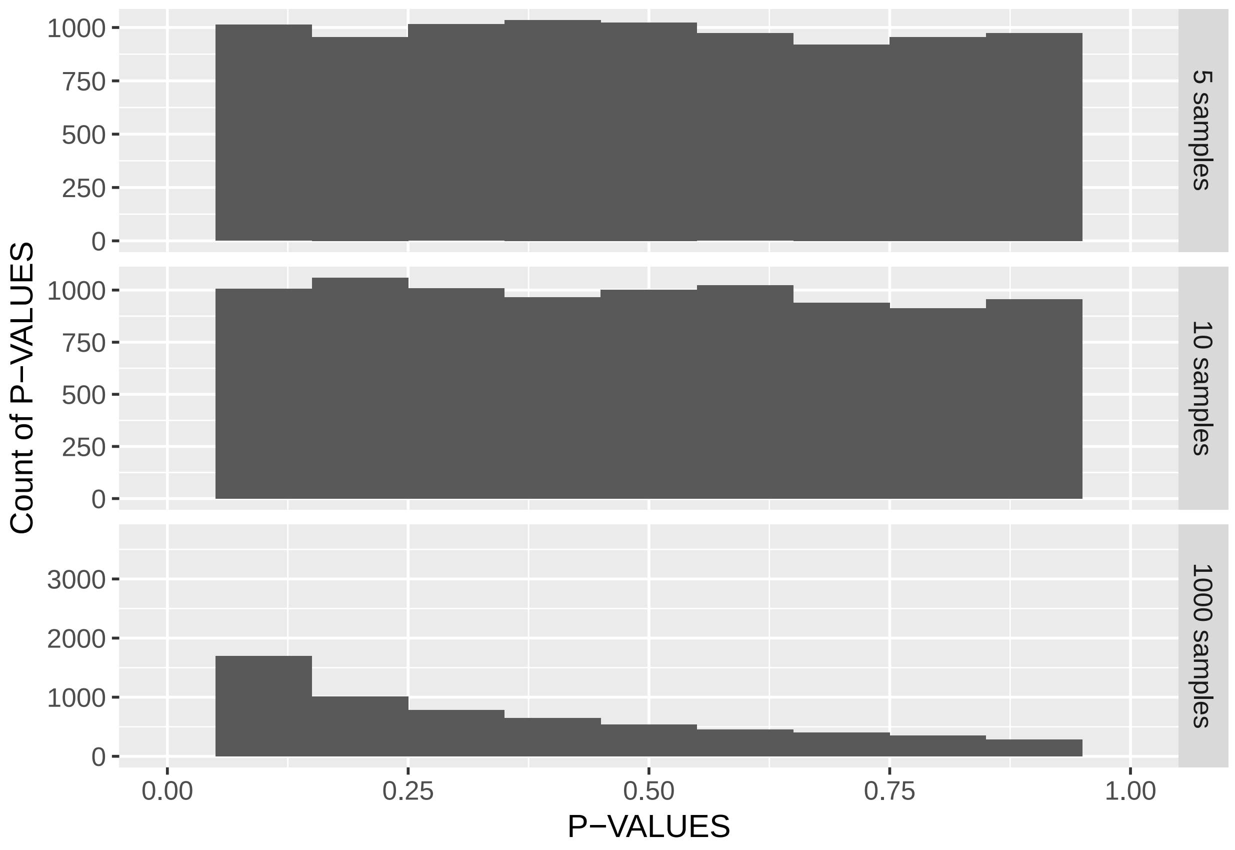
Density plot of normal and t distributions

```
my.data <- rt(n.rand, df = 20)
p.df.m <- assign_vector(my.data, n = n.test)
```

```
## No id variables; using all as measure variables
```

```
ggplot(p.df.m, aes(x = value)) +
  geom_histogram(binwidth = 1/10) +
  facet_grid(facets=variable ~ ., scales="free_y") +
  xlim(0,1) +
  ylab("Count of P-VALUES") +
  xlab("P-VALUES") +
  theme(text = element_text(size = 12))
```

```
## Warning: Removed 6 rows containing missing values (geom_bar).
```

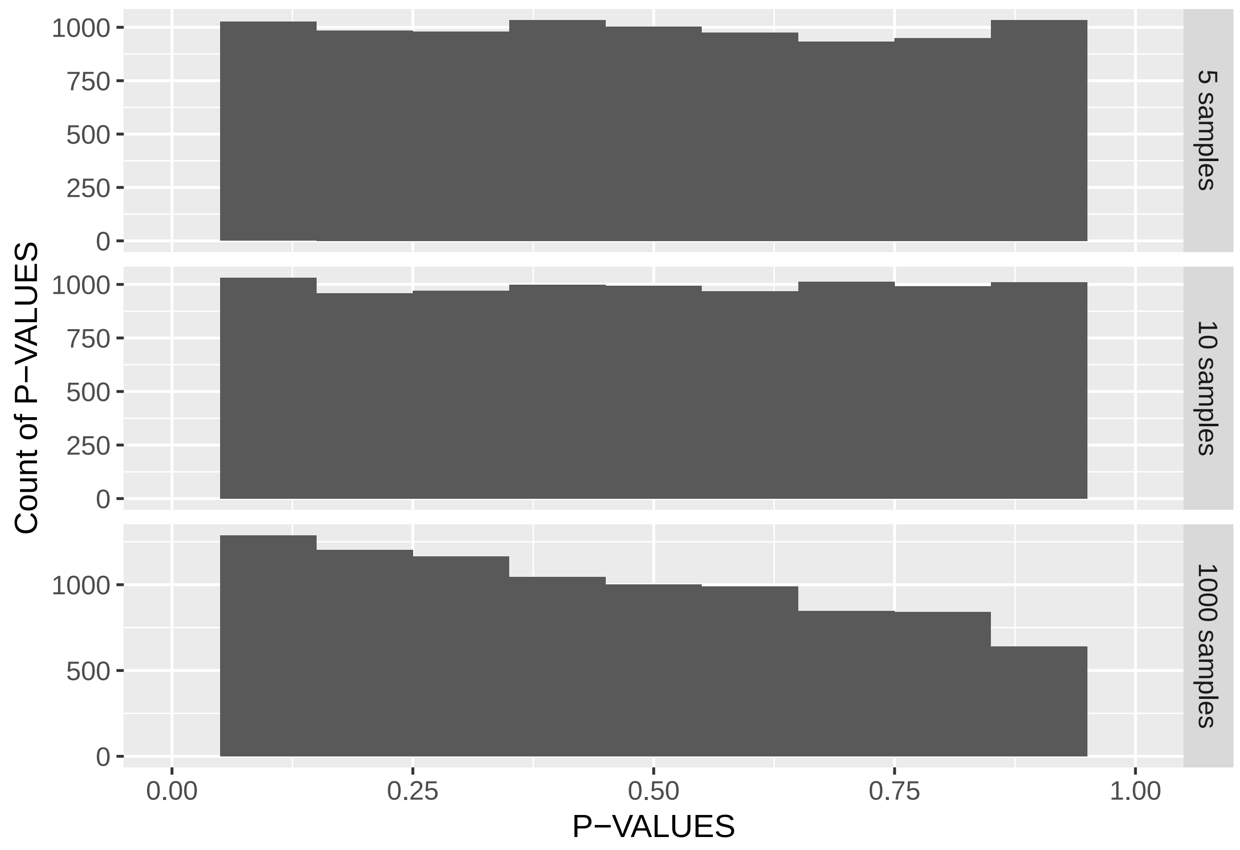


```
my.data <- rt(n.rand, df = 20)
my.data.2 <- rnorm(n.rand)
# Trim off the tails (-3 < my.data < 3)
my.data <- my.data[which(my.data < 3 & my.data > -3)]
# Add in tails from the other distribution (my.data.2 <-3 & my.data.2 >3)
my.data <- c(my.data, my.data.2[which(my.data.2 < -3 | my.data.2 > 3)])
p.df.m <- assign_vector(my.data, n = n.test)
```

```
## No id variables; using all as measure variables
```

```
ggplot(p.df.m, aes(x = value)) +
  geom_histogram(binwidth = 1/10) +
  facet_grid(facets=variable ~ ., scales="free_y") +
  xlim(0,1) +
  ylab("Count of P-VALUES") +
  xlab("P-VALUES") +
  theme(text = element_text(size = 12))
```

```
## Warning: Removed 6 rows containing missing values (geom_bar).
```

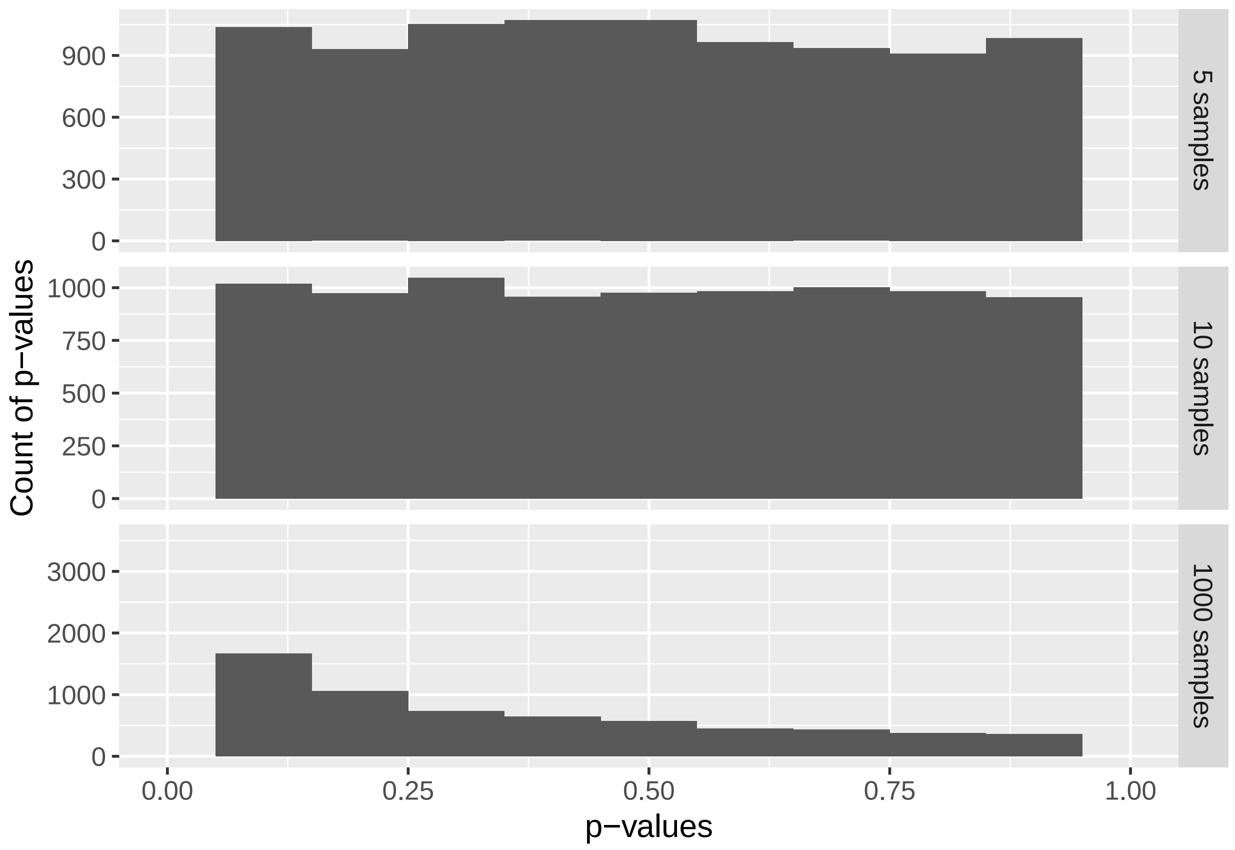


```
my.data <- rnorm(n.rand)
my.data.2 <- rt(n.rand, df = 20)
# Trim off the tails from similar to done above.
my.data <- my.data[which(my.data < 3 & my.data > -3)]
# Add in tails from the other distribution
my.data <- c(my.data, my.data.2[which(my.data.2 < -3 | my.data.2 > 3)])
p.df.m <- assign_vector(my.data, n = n.test)
```

```
## No id variables; using all as measure variables
```

```
ggplot(p.df.m, aes(x = value)) +
  geom_histogram(binwidth = 1/10) +
  facet_grid(facets=variable ~ ., scales="free_y") +
  xlim(0,1) +
  ylab("Count of p-values") +
  xlab("p-values") +
  theme(text = element_text(size = 12))
```

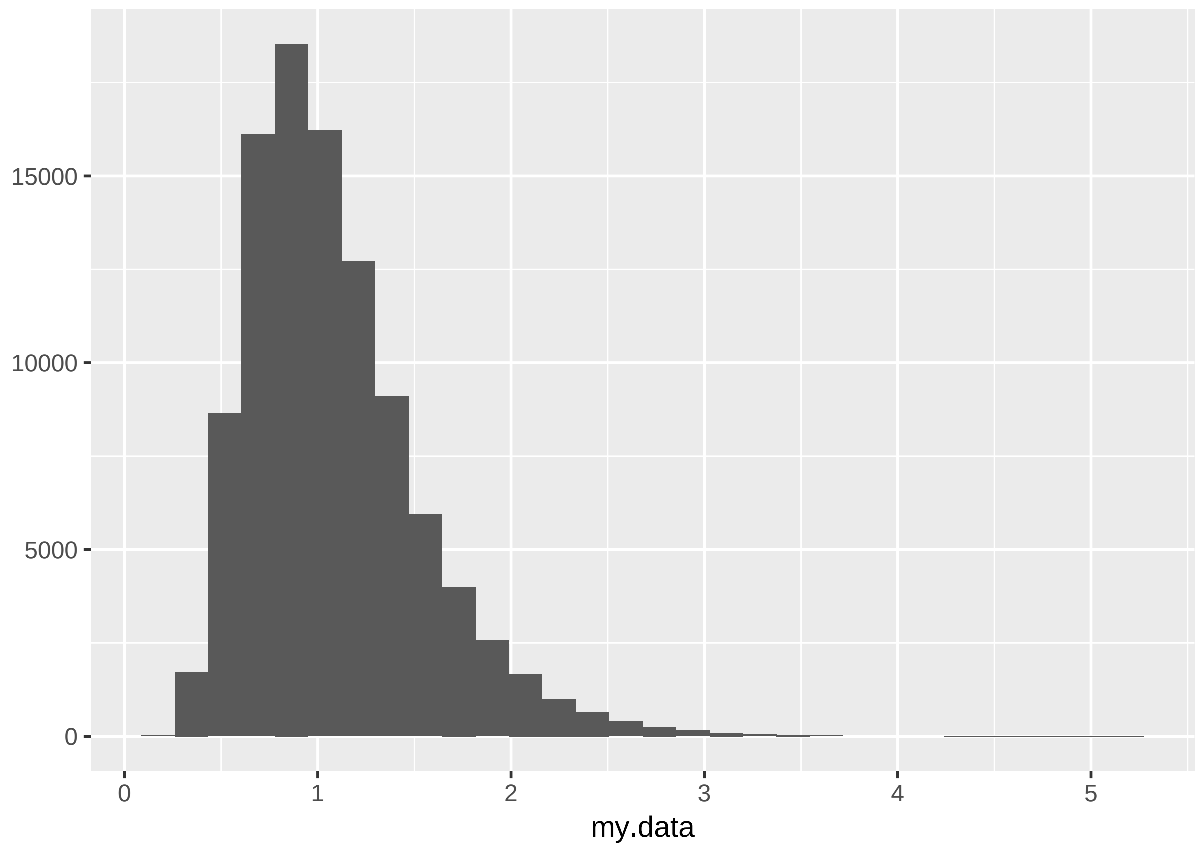
```
## Warning: Removed 6 rows containing missing values (geom_bar).
```



```
# Introducing a skewed data
my.data <- rlnorm(n.rand, 0, 0.4)
```

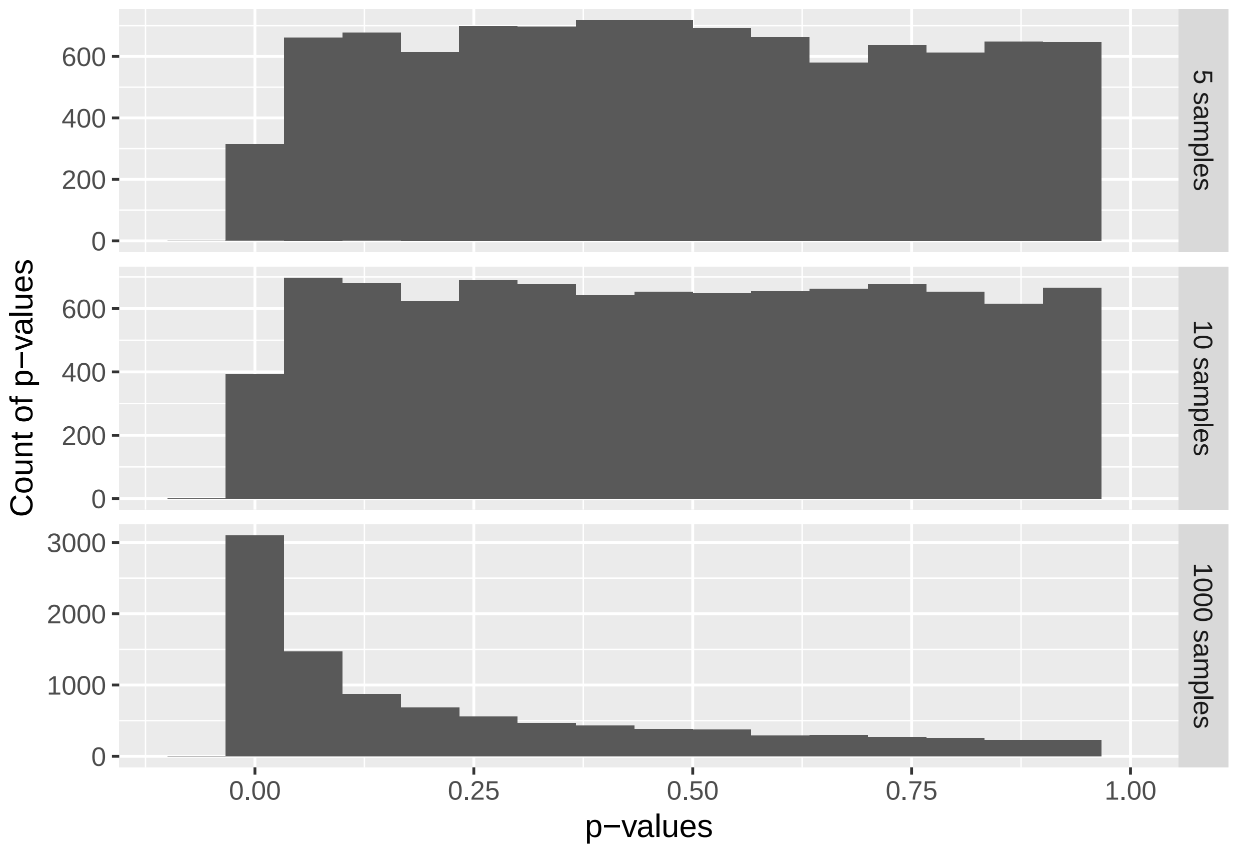
```
my.data <- rlnorm(n.rand, 0, 0.4)
#p.df.m <- assign_vector(my.data, n = n.test)
qplot(my.data, geom="histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(p.df.m, aes(x = value)) +
  geom_histogram(binwidth = 1/15) +
  facet_grid(facets=variable ~ ., scales="free_y") +
  xlim(-.1,1) +
  ylab("Count of p-values") +
  xlab("p-values") +
  theme(text = element_text(size = 12))
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```


Count of p−values
p−values
0
200
400
600
3000
2000
1000
0.00
0.25
0.50
0.75
1.00
5 samples
10 samples
1000 samples