# Week 3: Normality and Testing for Normality

Caution is needed when assuming normality in the populations that we analyze. ANOVA and t-tests assume normality in the population. There are both statistical and qualitative methods for checking for normality. However, qualitative methods have major human errors and statistical methods can be too sensitive to tails in the data.

This is a excerices using Shapiro-Wilk test for normality in R.

In [2]:

library(ggplot2)

library(reshape2)

Warning message:  
"package 'ggplot2' was built under R version 3.3.3"Warning message:  
"package 'reshape2' was built under R version 3.3.3"

In [3]:

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)

#Use cbind to combine the data and as.data.frame. to make the data into a data frame. Then define the column names

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 samples")))

return(p.df.m)

}

Create a set of random data

In [4]:

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

Because the data is random, we expect that there is equal probability in the p-values. 5% of the time we should see p-values less than or equal to 0.05.

Use a ggplot histogram to confirm this assumption.

In [5]:

p.df.m <- assign\_vector(my.data, n = n.test)

ggplot(p.df.m, aes(x = value)) **+**

geom\_histogram(binwidth = 1**/**10) **+**

facet\_grid(facets=variable **~** ., scales="free\_y") **+**

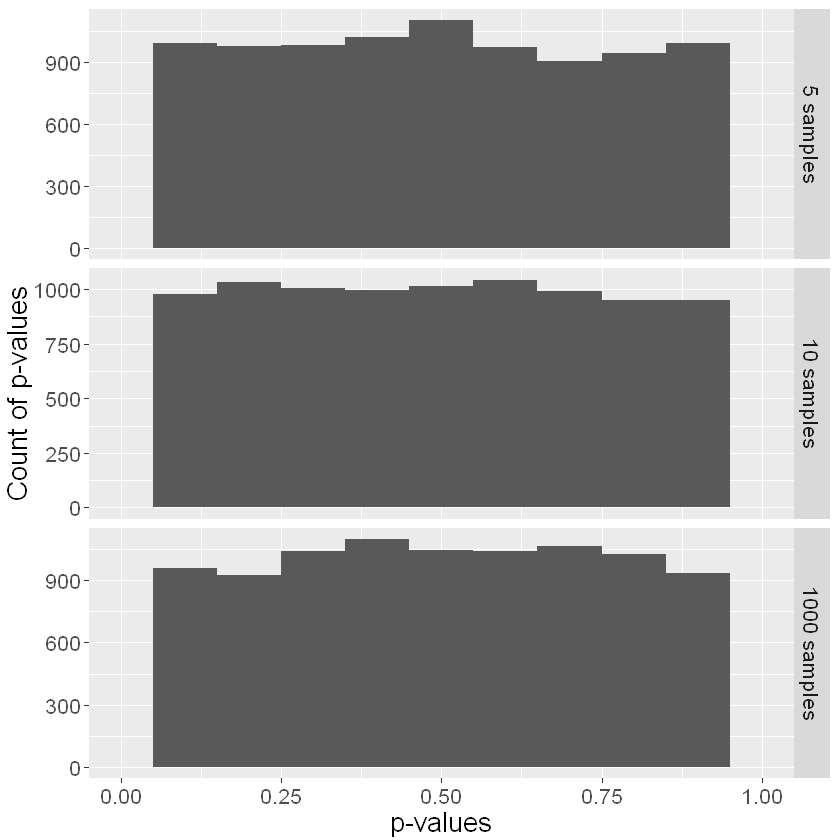
xlim(0,1) **+**

ylab("Count of p-values") **+**

xlab("p-values") **+**

theme(text = element\_text(size = 16))

No id variables; using all as measure variables



This confirms our assumption about the normal distribution as all the p-value counts are relatively equal.

Next, we look at a t distribution. To the naked eye the t-distribution looks normal, but it is not!

In [6]:

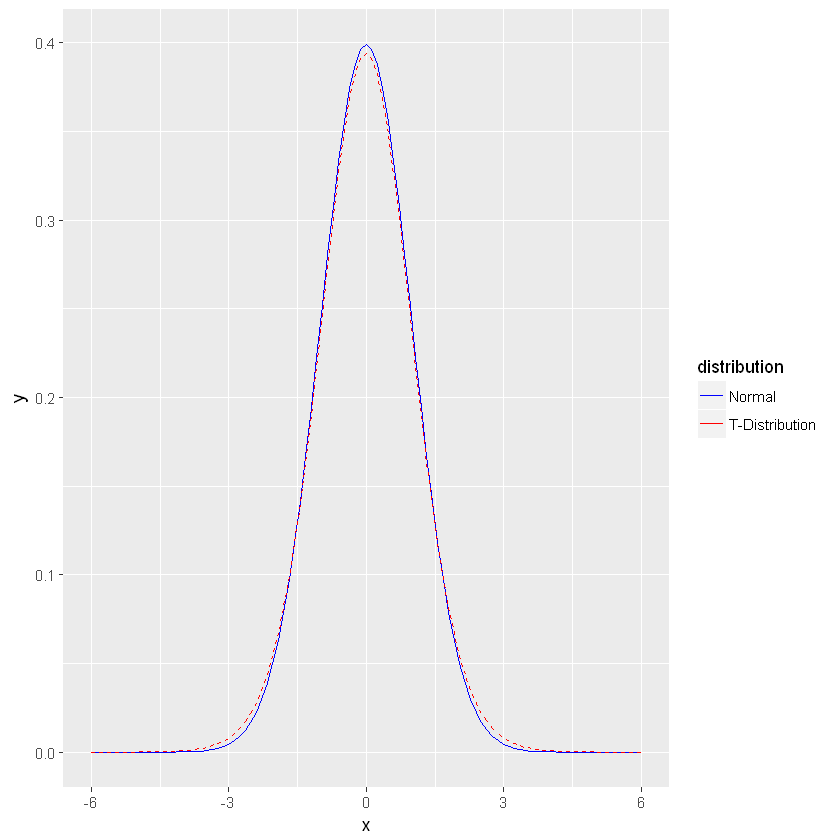
ggplot(NULL, aes(x=x, colour = distribution)) **+**

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"))

​



In [41]:

my.data <- rt(n.rand, df = 20)

p.df.m <- assign\_vector(my.data, n = n.test)

ggplot(p.df.m, aes(x = value)) **+**

geom\_histogram(binwidth = 1**/**50) **+**

facet\_grid(facets=variable **~** ., scales="free\_y") **+**

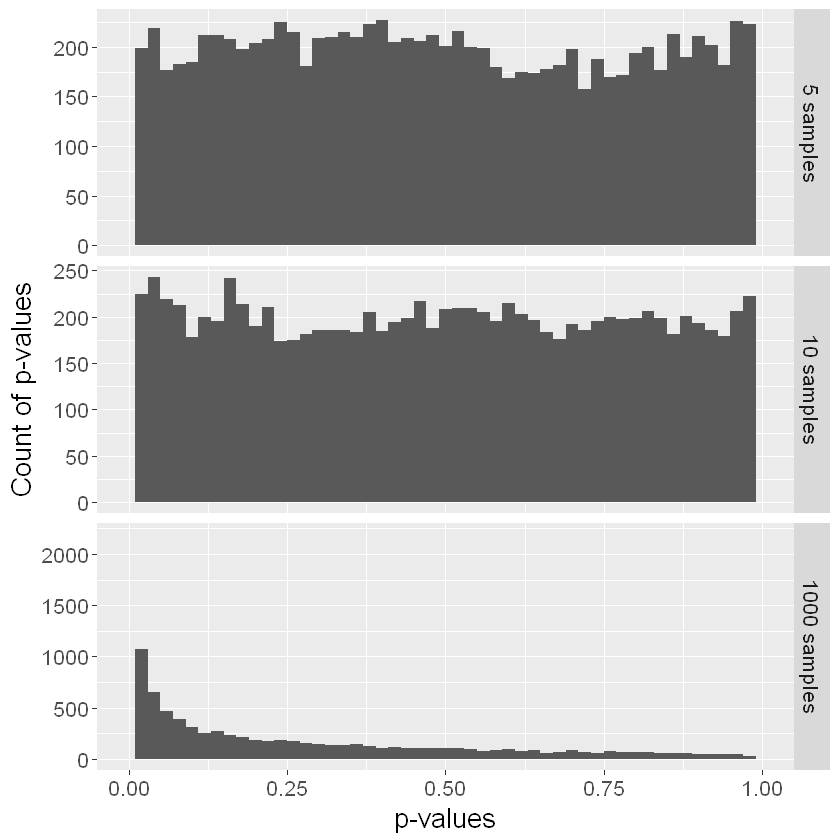
xlim(0,1) **+**

ylab("Count of p-values") **+**

xlab("p-values") **+**

theme(text = element\_text(size = 16))

No id variables; using all as measure variables



At the two small sample sizes it looks like the data is normalized. However in a bigger data set only fails the test for normality about 50% of the time (add up the frequencies for p-value > 0.05 to see this).

## Testing for tails

To test the tails, we can construct a data set that uses the t distribution for the middle 99% of the data, and the normal distribution for the tails.

In [42]:

my.data.2 <- rnorm(n.rand)

In [43]:

*# Trim off the tails*

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)])

In [44]:

p.df.m <- assign\_vector(my.data, n = n.test)

ggplot(p.df.m, aes(x = value)) **+**

geom\_histogram(binwidth = 1**/**50) **+**

facet\_grid(facets=variable **~** ., scales="free\_y") **+**

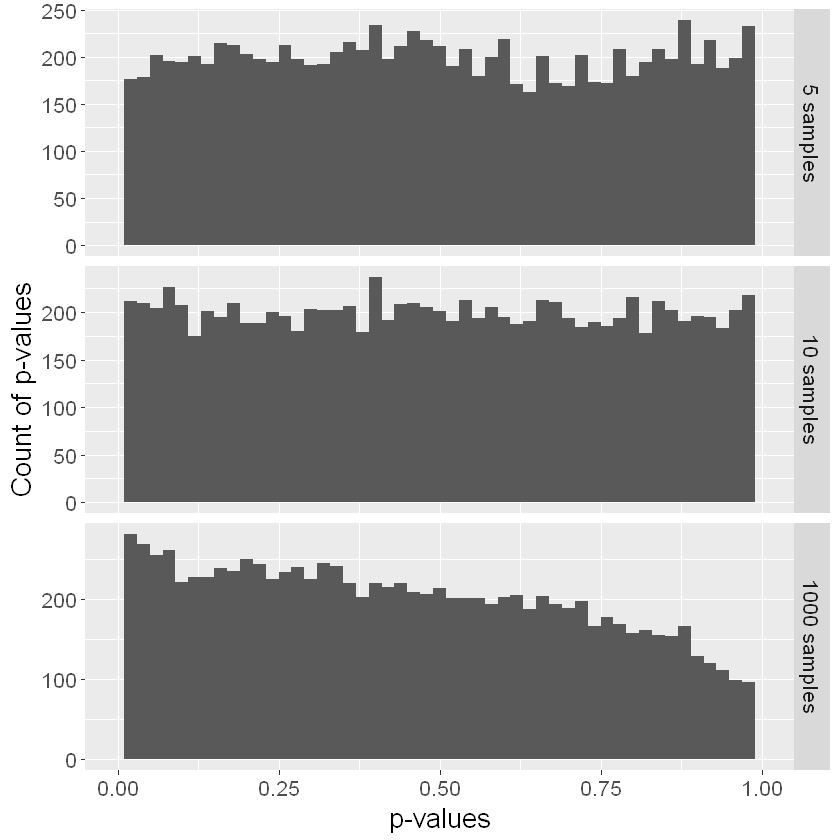
xlim(0,1) **+**

ylab("Count of p-values") **+**

xlab("p-values") **+**

theme(text = element\_text(size = 16))

No id variables; using all as measure variables



Removing the 1% of tails and only looking at the 99% of data in this t distribution the data looks much more normalized. This is proof that the tails have an impact on our analysis.

To look further into this problem, we can use the t distribution with “extreme” tails.

​

In [45]:

my.data <- rnorm(n.rand)

my.data.2 <- rt(n.rand, df = 20)

*# Trim off the tails*

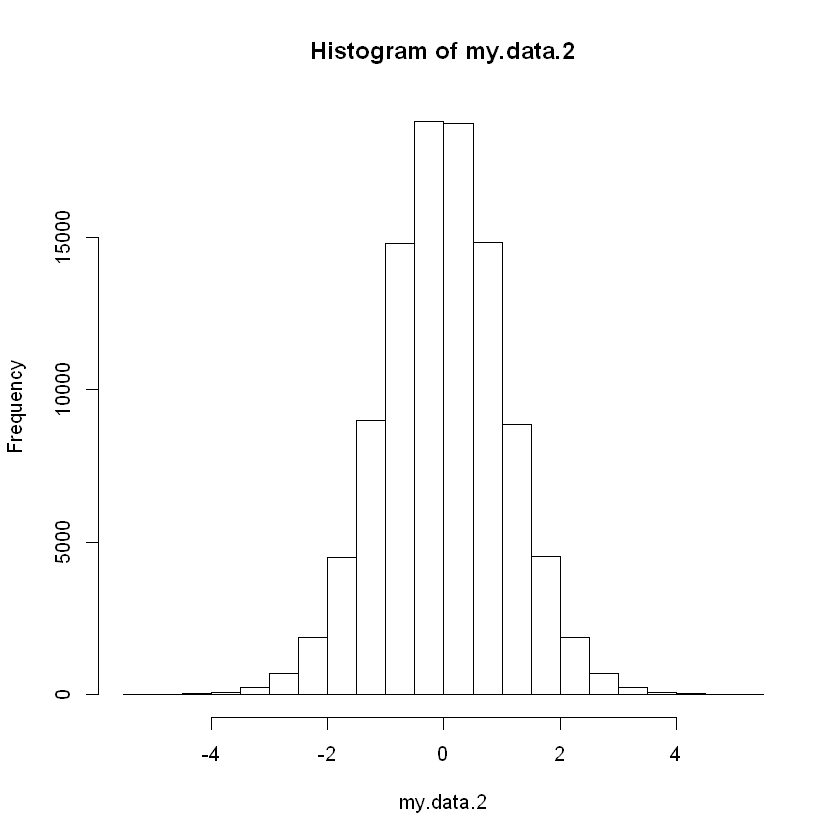
my.data <- my.data[which(my.data **<** 3 **&** my.data **> -**3)]

*# Add in tails from the other distribution*

my.data <- c(my.data, my.data.2[which(my.data.2 **< -**3 **|** my.data.2 **>** 3)])

In [49]:

hist(my.data.2)



In [46]:

p.df.m <- assign\_vector(my.data, n = n.test)

ggplot(p.df.m, aes(x = value)) **+**

geom\_histogram(binwidth = 1**/**50) **+**

facet\_grid(facets=variable **~** ., scales="free\_y") **+**

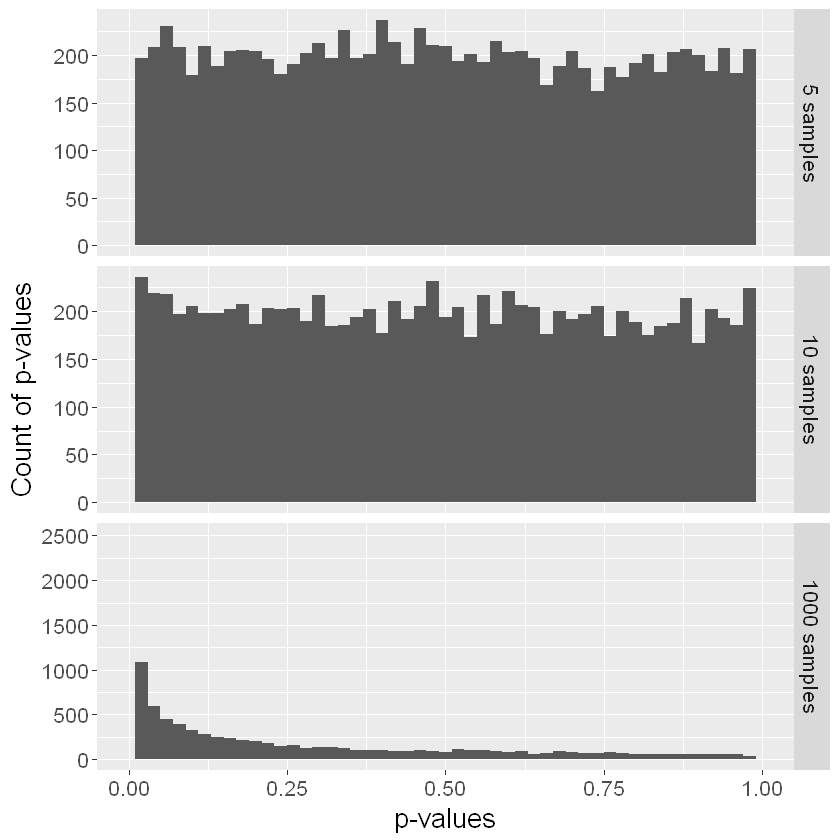
xlim(0,1) **+**

ylab("Count of p-values") **+**

xlab("p-values") **+**

theme(text = element\_text(size = 16))

No id variables; using all as measure variables



​“If you check the y-axis scales carefully, you’ll see that the chance of getting p-value ≤ 0.05 is a bit lower here than for the t distribution.”

Looking at even more skewed data:

In [47]:

my.data <- rlnorm(n.rand, 0, 0.4)

​

In [48]:

hist(my.data)

In [50]:

p.df.m <- assign\_vector(my.data, n = n.test)

ggplot(p.df.m, aes(x = value)) **+**

geom\_histogram(binwidth = 1**/**50) **+**

facet\_grid(facets=variable **~** ., scales="free\_y") **+**

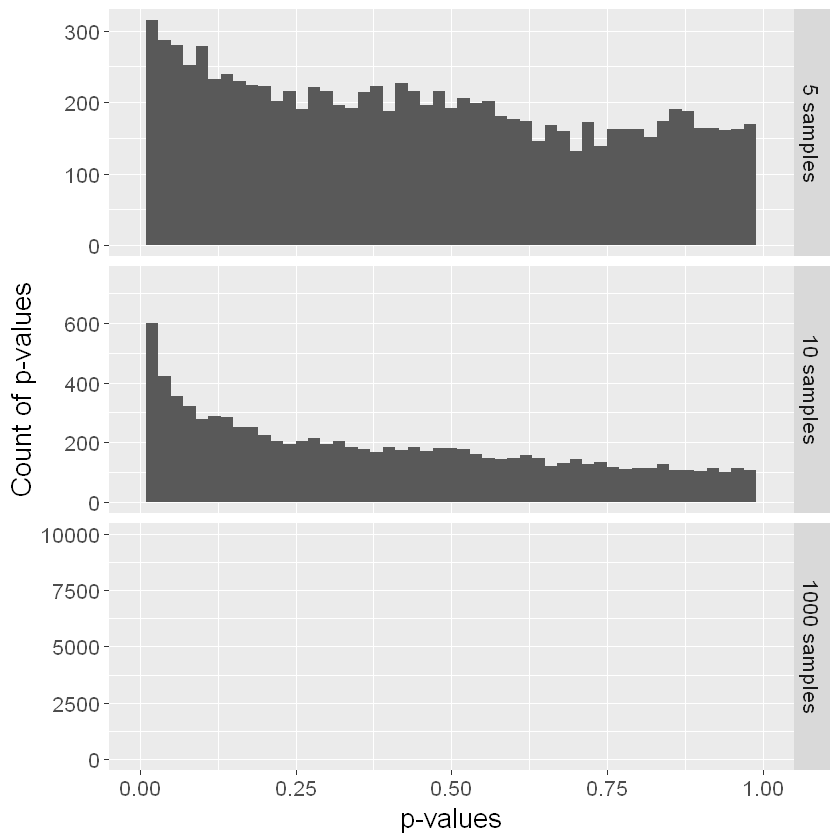
xlim(0,1) **+**

ylab("Count of p-values") **+**

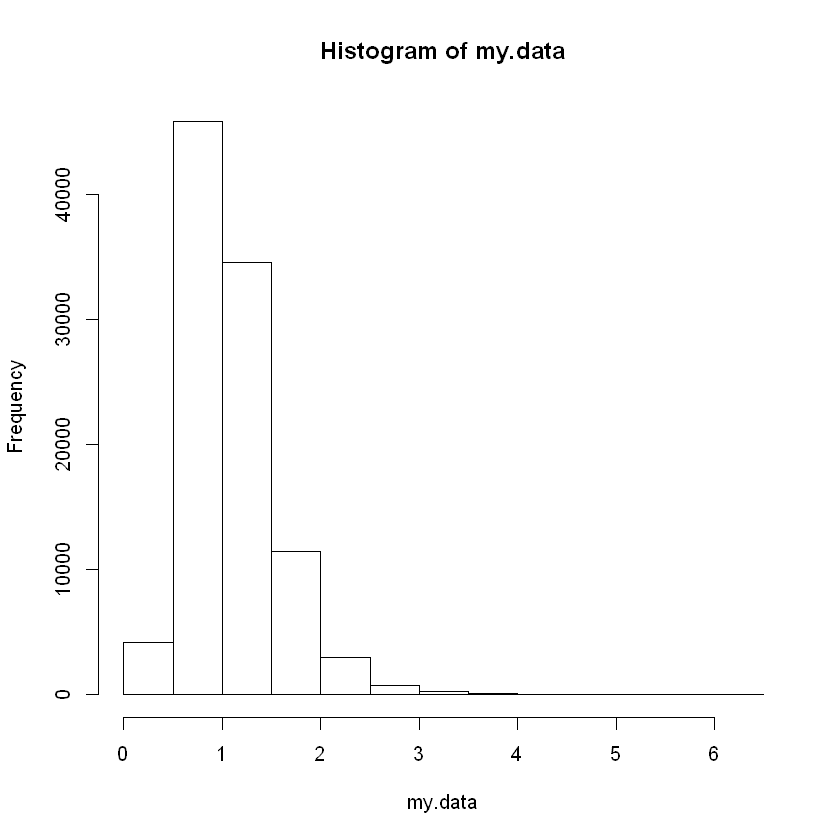
xlab("p-values") **+**

theme(text = element\_text(size = 16))

No id variables; using all as measure variables



For the small sample sizes these could possibly still pass a test for normality.



As we can see tails and small sample sizes cause major problems with testing for normality.