Lab1 Report

Ninh (Vincent) DO

March 23, 2019

`birthwt` dataset

This package is part of the MASS library. Load those alongside ggplot2 and plyr

```
library(MASS)
library(plyr)
library(ggplot2)
```

A Little Bit of Pre-processing

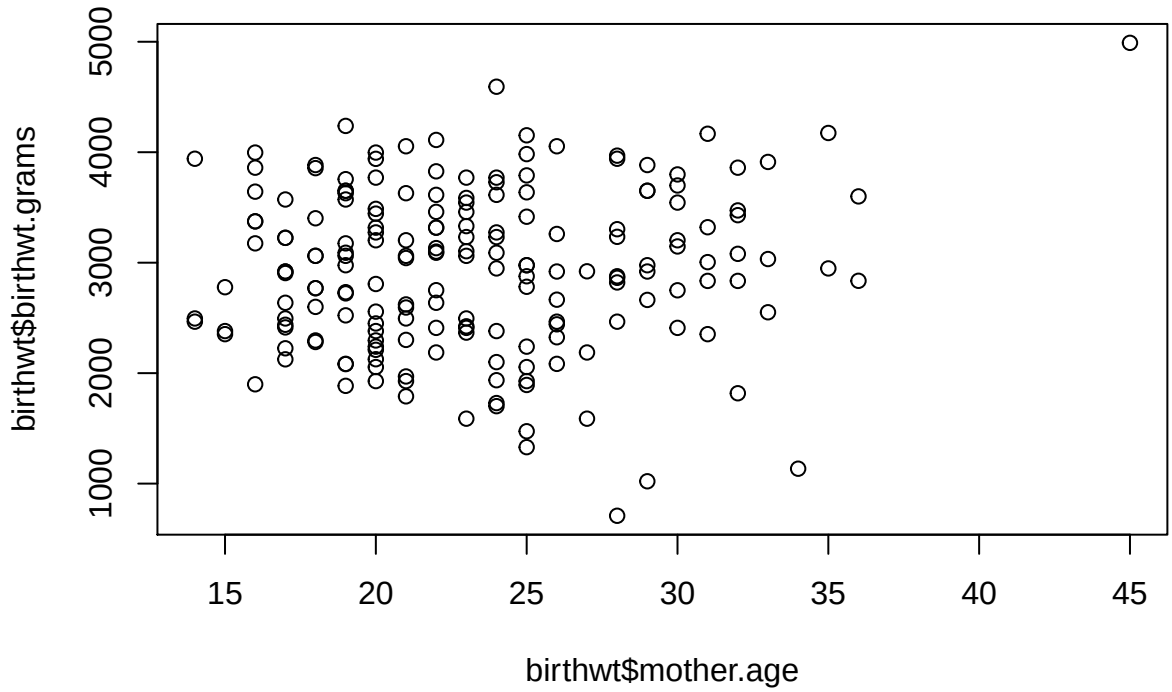
Here we just change some variable names and levels to more descriptive names and levels.

```
# Rename the columns to have more descriptive names
colnames(birthwt) <- c("birthwt.below.2500", "mother.age", "mother.weight",
    "race", "mother.smokes", "previous.prem.labor", "hypertension", "uterine.irr",
    "physician.visits", "birthwt.grams")

# Transform variables to factors with descriptive levels
birthwt <- mutate(birthwt,
                  race = as.factor(mapvalues(race, c(1, 2, 3),
                                             c("white","black", "other"))),
                  mother.smokes = as.factor(mapvalues(mother.smokes,
                                                      c(0,1), c("no", "yes"))),
                  hypertension = as.factor(mapvalues(hypertension,
                                                     c(0,1), c("no", "yes"))),
                  uterine.irr = as.factor(mapvalues(uterine.irr,
                                                    c(0,1), c("no", "yes")))
)
```

Doing some EDA.

```
# Plot histogram of mother ages to see if the survey is comprehensive,
# i.e. covering a uniformly wide range of mother ages
hist(birthwt$mother.age, breaks = max(birthwt$mother.age) - min(birthwt$mother.age) + 1)
```


```
# Scatterplot of mother ages and birth weights to have some idea about correlation
plot(birthwt$mother.age, birthwt$birthwt.grams)
```




```
# Calculate Pearson correlation coefficient
p = cor(birthwt$mother.age, birthwt$birthwt.grams)
p
```

```
## [1] 0.09031781
```

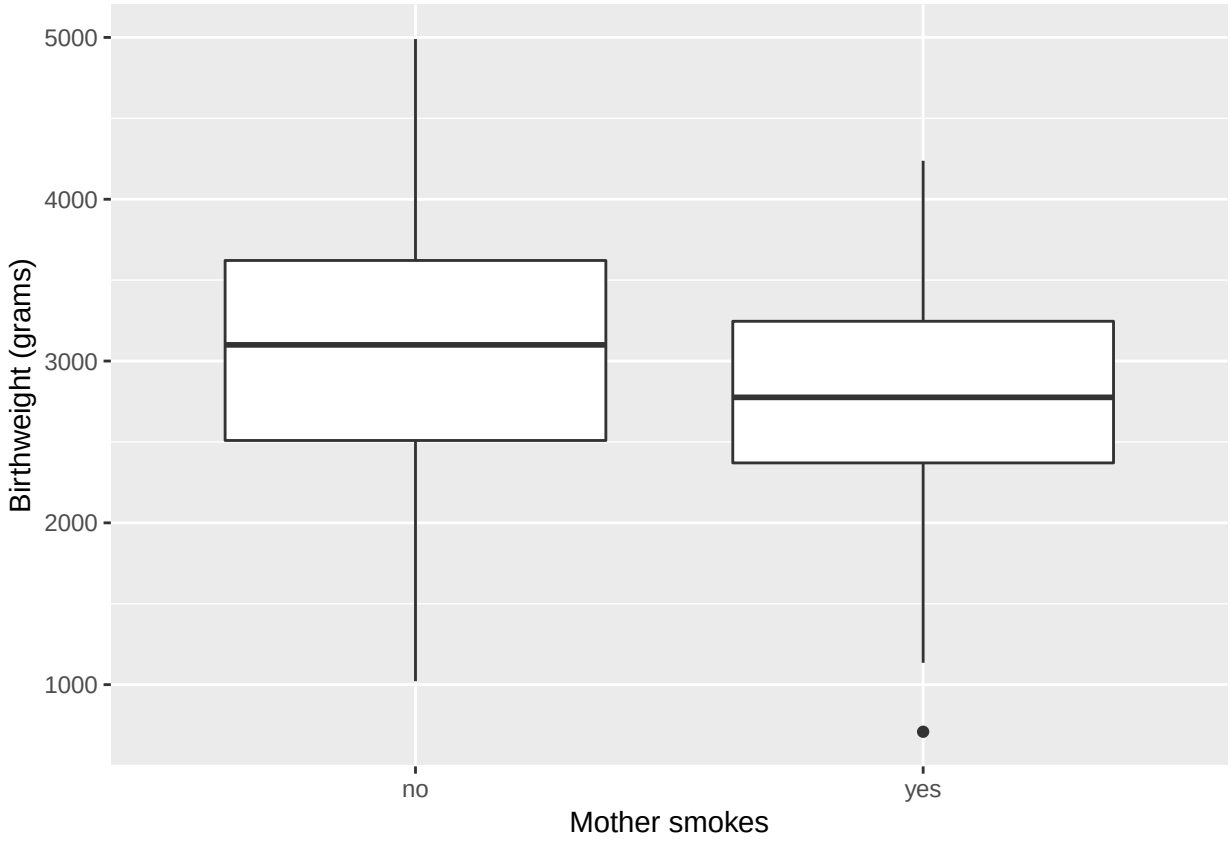
The plot and Pearson correlation coefficient r of mother ages and birth weights does not show any correlation between these two variables.

Testing differences in means

Let's compare the birth weight between smoking and non-smoking mothers.

First, create a boxplot of the two groups.

```
ggplot(birthwt, aes(x = mother.smokes, y = birthwt.grams)) +
  geom_boxplot() +
  labs(x = "Mother smokes", y = "Birthweight (grams)")
```



This plot suggests that smoking is associated with lower birth weight.

We can assess whether this difference is statistically significant.

Now create a summary table.

```
birthwt_smoke_summary = ddply(birthwt, c("mother.smokes"), summarise, mean.birthwt = mean(birthwt.grams)
birthwt_smoke_summary
```

```
##   mother.smokes mean.birthwt sd.birthwt
## 1            no     3055.696   752.6566
## 2           yes     2771.919   659.6349
```

To assess statistical significance we really want to have the standard error (which the standard deviation adjusted by the group size).

```
birthwt_smoke_summary_more = ddply(birthwt, c("mother.smokes"), summarise, group.size = length(birthwt.g
birthwt_smoke_summary_more
```

```
##   mother.smokes group.size mean.birthwt sd.birthwt se.mean.birthwt
## 1            no        115     3055.696   752.6566        70.18559
## 2           yes         74     2771.919   659.6349        76.68100
```

t-test via t.test()

This difference looks quite significant. Now run a two-sample t-test, using the `t.test()` function.

```
birthwt.t.test = t.test(birthwt.grams ~ mother.smokes, data = birthwt)
birthwt.t.test
```

```
##
##  Welch Two Sample t-test
##
## data:  birthwt.grams by mother.smokes
## t = 2.7299, df = 170.1, p-value = 0.007003
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   78.57486 488.97860
## sample estimates:
##  mean in group no mean in group yes
##          3055.696          2771.919
```

Conclusion: p-value << 0.5 -> statistically significant. The difference is real.

The `t.test()` function also outputs a confidence interval and other information.

```
names(birthwt.t.test)
```

```
## [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
## [6] "null.value"  "alternative" "method"      "data.name"
```

p-value:

```
birthwt.t.test$p.value
```

```
## [1] 0.007002548
```

sample estimates:

```
birthwt.t.test$estimate
```

```
##  mean in group no mean in group yes
##          3055.696          2771.919
```

95% confidence interval:

```
birthwt.t.test$conf.int
```

```
## [1]  78.57486 488.97860
## attr(,"conf.level")
## [1] 0.95
```

confidence level:

```
attr(birthwt.t.test$conf.int, "conf.level")
```

```
## [1] 0.95
```

Calculate difference in means between smoking and nonsmoking groups.

```
birthwt.t.test$estimate
```

```
##  mean in group no mean in group yes
##          3055.696          2771.919
```

```
birthwt.smoke.diff = birthwt.t.test$estimate[1] - birthwt.t.test$estimate[2]
```

Confidence level as a %

```
conf.level <- attr(birthwt.t.test$conf.int, "conf.level") * 100
conf.level
```

```
## [1] 95
```

We can inlay test results in our report without hardcoding. Example: Here's what happens when we knit the following paragraph.

Our study finds that birth weights are on average 283.7767333g higher in the non-smoking group compared to the smoking group (t-statistic 2.73, p=0.007, 95% CI [78.6, 489]g)

`t.test()` accepts input in multiple forms. Here's another way of specifying the same information.

```
with(birthwt, t.test(x=birthwt.grams[mother.smokes=="no"],
                     y=birthwt.grams[mother.smokes=="yes"]))
```

```
##
##  Welch Two Sample t-test
##
## data:  birthwt.grams[mother.smokes == "no"] and birthwt.grams[mother.smokes == "yes"]
## t = 2.7299, df = 170.1, p-value = 0.007003
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   78.57486 488.97860
## sample estimates:
## mean of x mean of y
##  3055.696  2771.919
```

Specifying x and y arguments to the `t.test` function runs a t-test to check whether x and y have the same mean.

```
t.test(birthwt$birthwt.grams[birthwt$mother.smokes == "no"], birthwt$birthwt.grams[birthwt$mother.smoke
```

```
##
##  Welch Two Sample t-test
##
## data:  birthwt$birthwt.grams[birthwt$mother.smokes == "no"] and birthwt$birthwt.grams[birthwt$mother
## t = 2.7299, df = 170.1, p-value = 0.007003
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   78.57486 488.97860
## sample estimates:
## mean of x mean of y
##  3055.696  2771.919
```

The Meaning of Tests of Significance.

Perform a simulation using the helper function simulateData provided below to create data from Null and Alternative with effect size of 1.5 and perform tests of significance of difference in means on both along with boxplots and density plots.

- First simulation (Null case): the treatment has no effect
- Second simulation (Alternative case): the treatment on average increases outcome

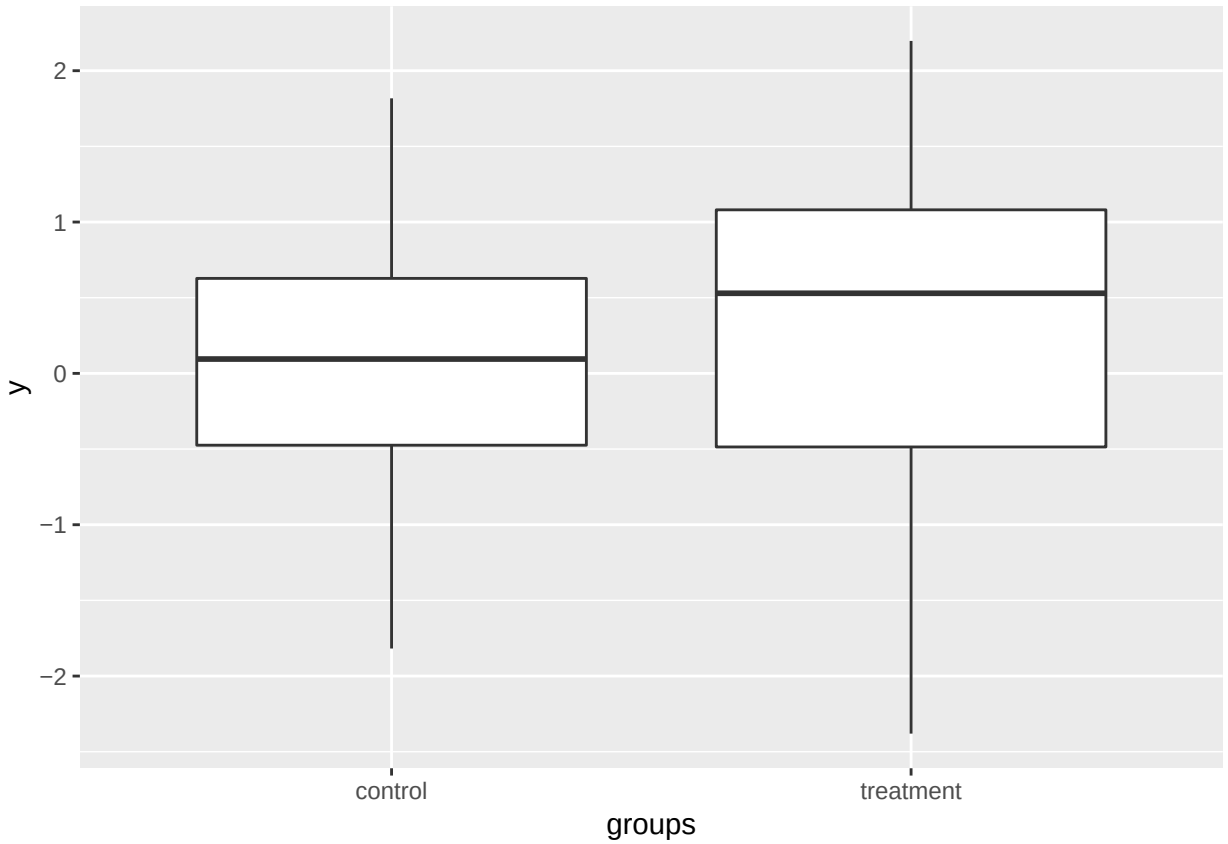
```
set.seed(12345)
# Function to generate data
simulateData <- function(n1, n2, mean.shift = 0) {
  y <- rnorm(n1 + n2) + c(rep(0, n1), rep(mean.shift, n2))
  groups <- c(rep("control", n1), rep("treatment", n2))

  return(data.frame(y = y, groups = groups))
}
```

Null

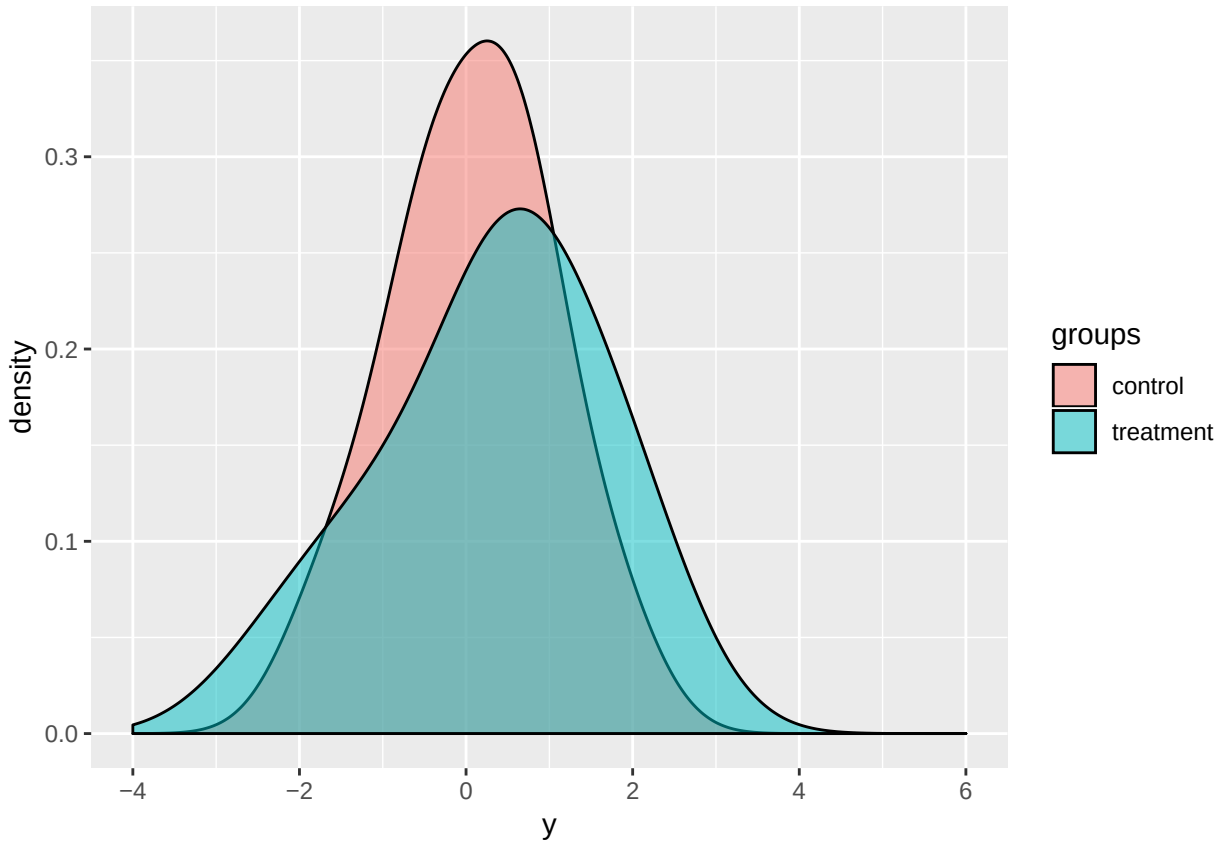
```
n1 = 30
n2 = 30

no_effect = simulateData(n1 = n1, n2 = n2)
head(no_effect)
```

```
##            y  groups
## 1  0.5855288 control
## 2  0.7094660 control
## 3 -0.1093033 control
## 4 -0.4534972 control
## 5  0.6058875 control
## 6 -1.8179560 control
```

```
ggplot(data = no_effect, aes(x = groups, y = y)) +
  geom_boxplot()
```




```
qplot(fill = groups, x = y, data = no_effect, geom = "density",
      alpha = I(0.5),
      adjust = 1.5,
      xlim = c(-4, 6),
      ylab = "density")
```

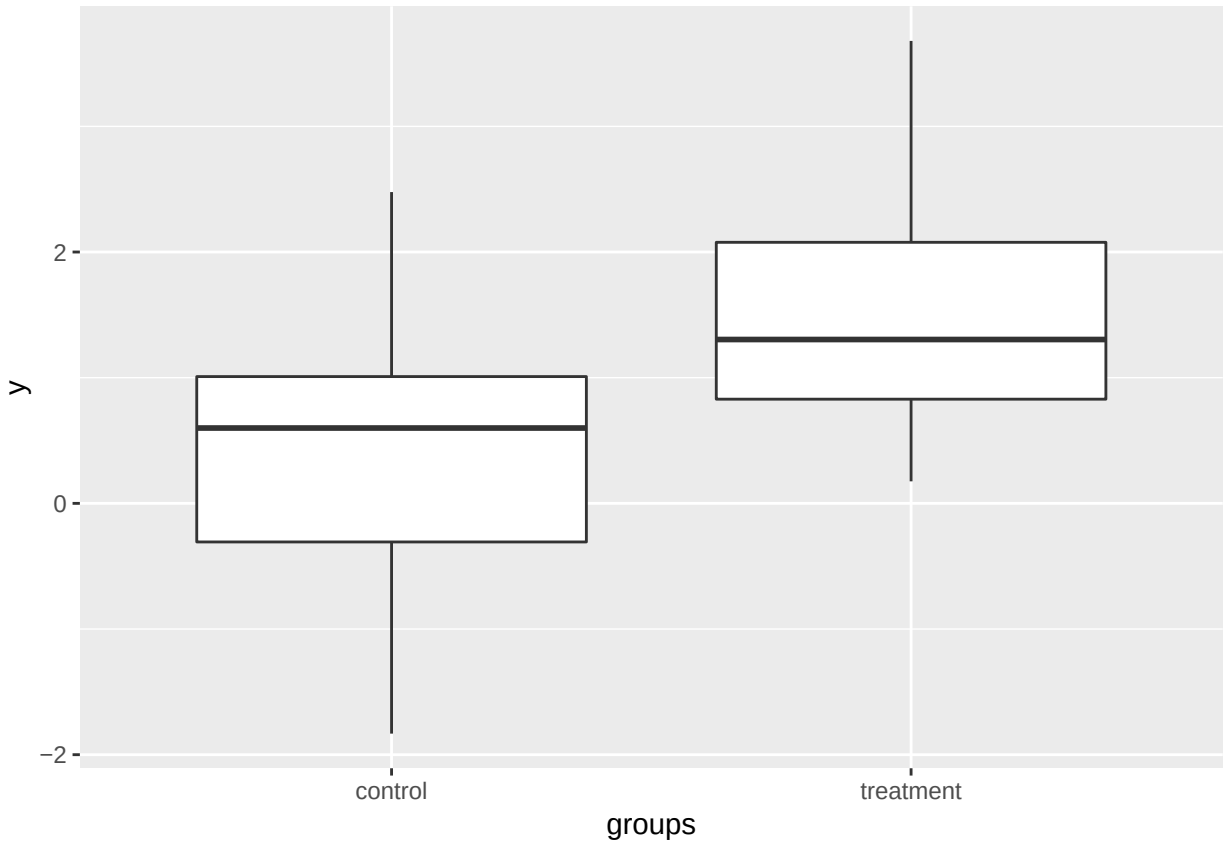



```
t.test(y ~ groups, data = no_effect)
```

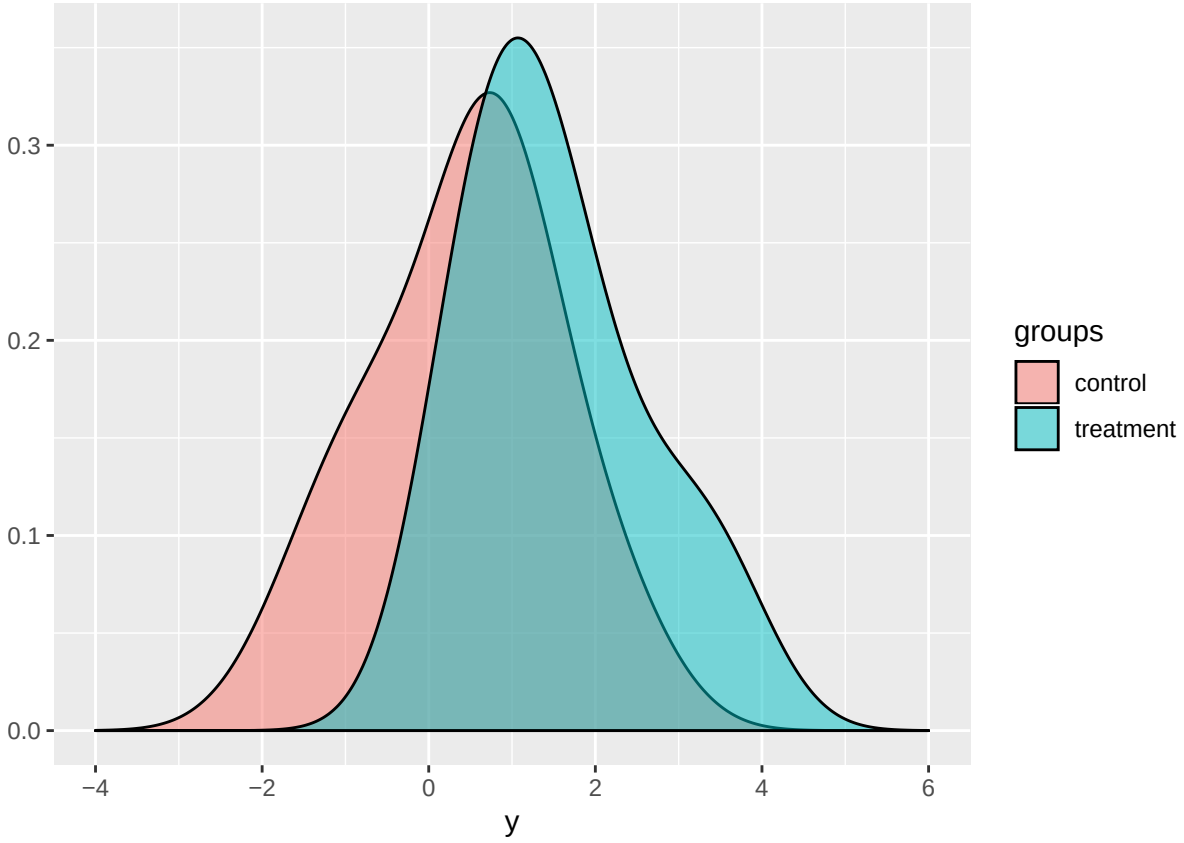
```
##
##  Welch Two Sample t-test
##
## data:  y by groups
## t = -0.80109, df = 53.069, p-value = 0.4267
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8154901  0.3499890
## sample estimates:
##   mean in group control mean in group treatment
##              0.07880701              0.31155756
```

Alternative

```
# Non-null case, very strong treatment effect
# Observation, null case
obs.data <- simulateData(n1 = n1, n2 = n2, mean.shift = 1.5)

# Box plots
qplot(x = groups, y = y, data = obs.data, geom = "boxplot")
```




```
# Density plots
qplot(fill = groups, x = y, data = obs.data, geom = "density",
      alpha = I(0.5),
      adjust = 1.5,
      xlim = c(-4, 6))
```




```
# t-test
t.test(y ~ groups, data = obs.data)
```

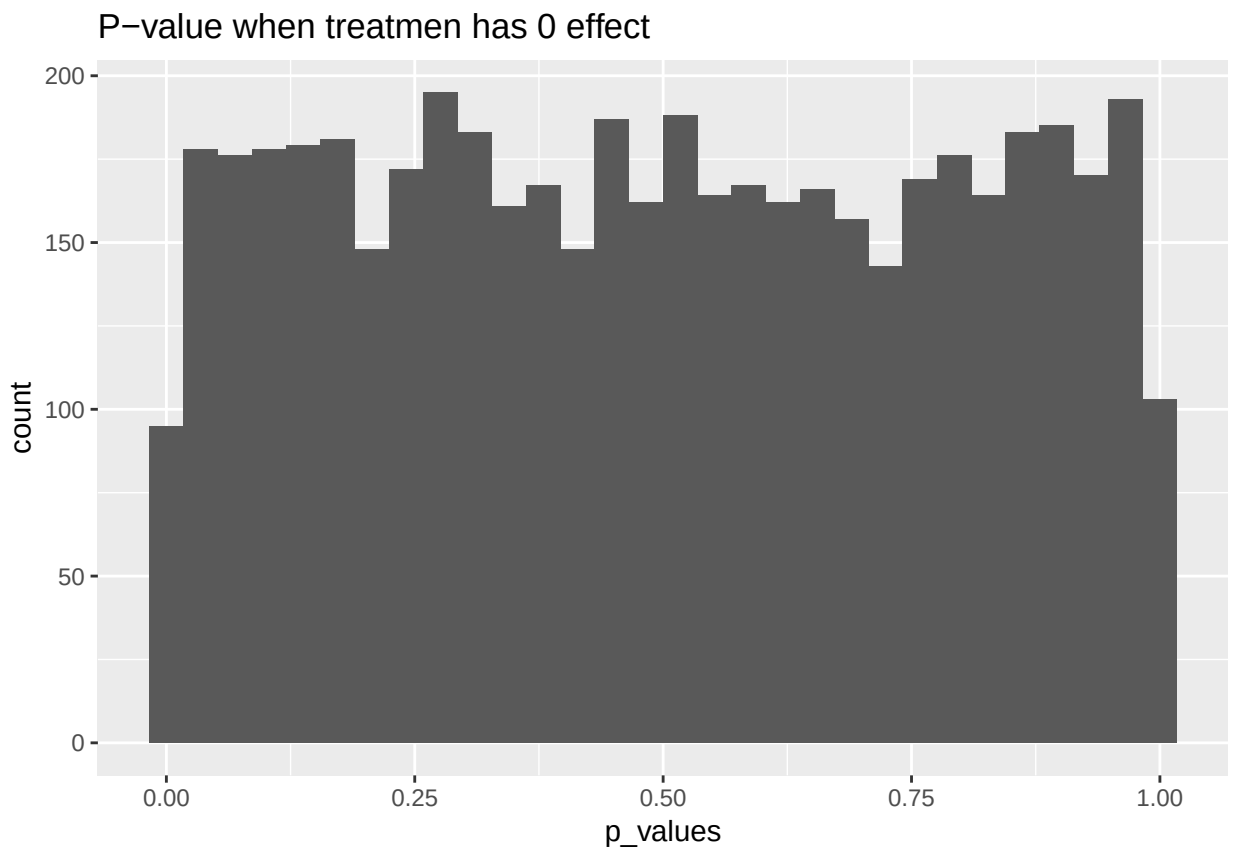
```
##
##  Welch Two Sample t-test
##
## data:  y by groups
## t = -3.979, df = 57.858, p-value = 0.0001952
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.6378573 -0.5414529
## sample estimates:
##   mean in group control mean in group treatment
##               0.4355869               1.5252421
```

Now repeat the above simulation 1e4 times and plot the p-values. Use a modest effect size of 0.5 for the alternative case.

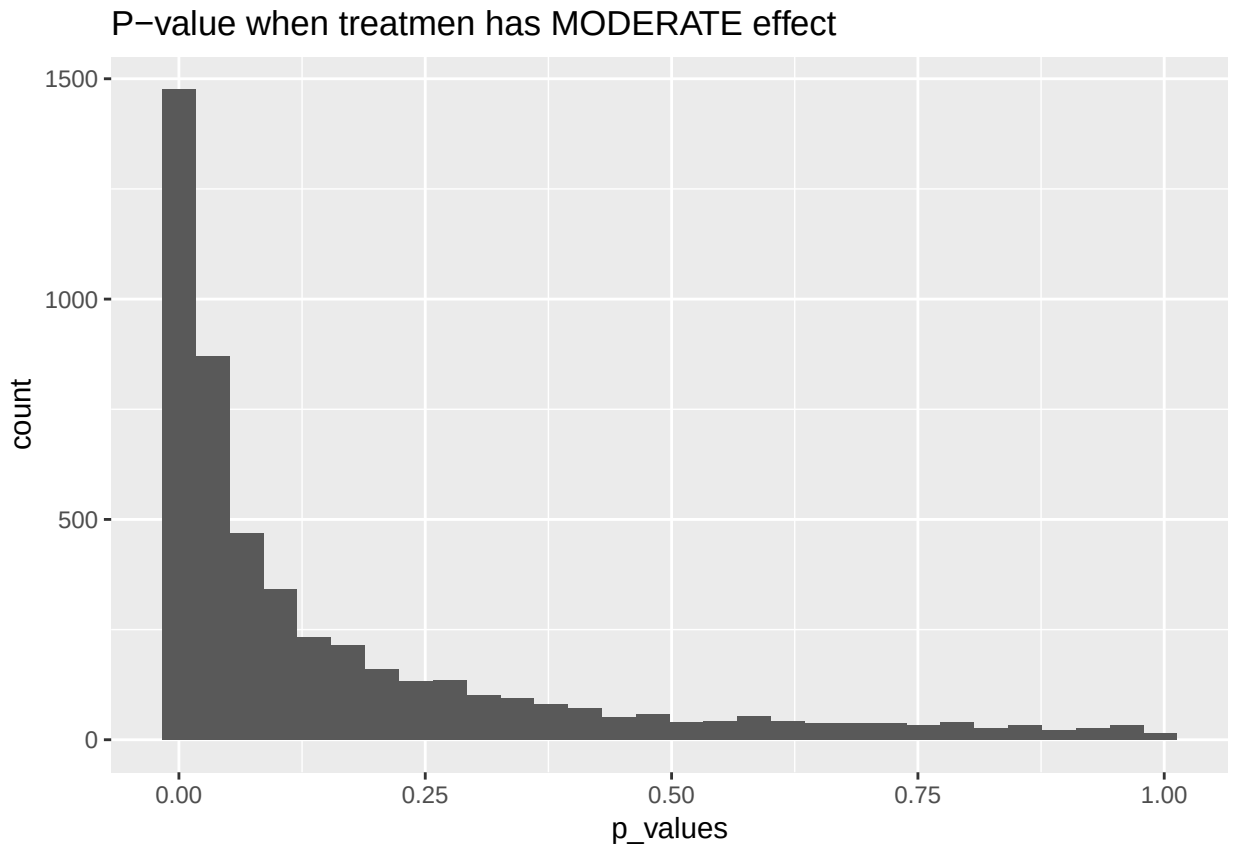
```
N = 1e4
null = c(rep(TRUE, N/2), rep(FALSE, N/2))
p_values = rep(0, N)

for (i in 1:N) {
  if (null[i]) {
    mean_shift = 0
  } else {
    mean_shift = 0.5
  }
```

```
  dat = simulateData(n1 = n1, n2 = n2, mean.shift = mean_shift)
  ttest_results = t.test(y ~ groups, data = dat)
  p_values[i] = ttest_results$p.value
}

p_df = data.frame(null = null, p_values = p_values)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
ggplot(data = p_df[p_df$null == TRUE, ], mapping = aes(x = p_values)) +
  stat_bin(bins = 30) +
  labs(title = "P-value when treatmen has 0 effect")
```

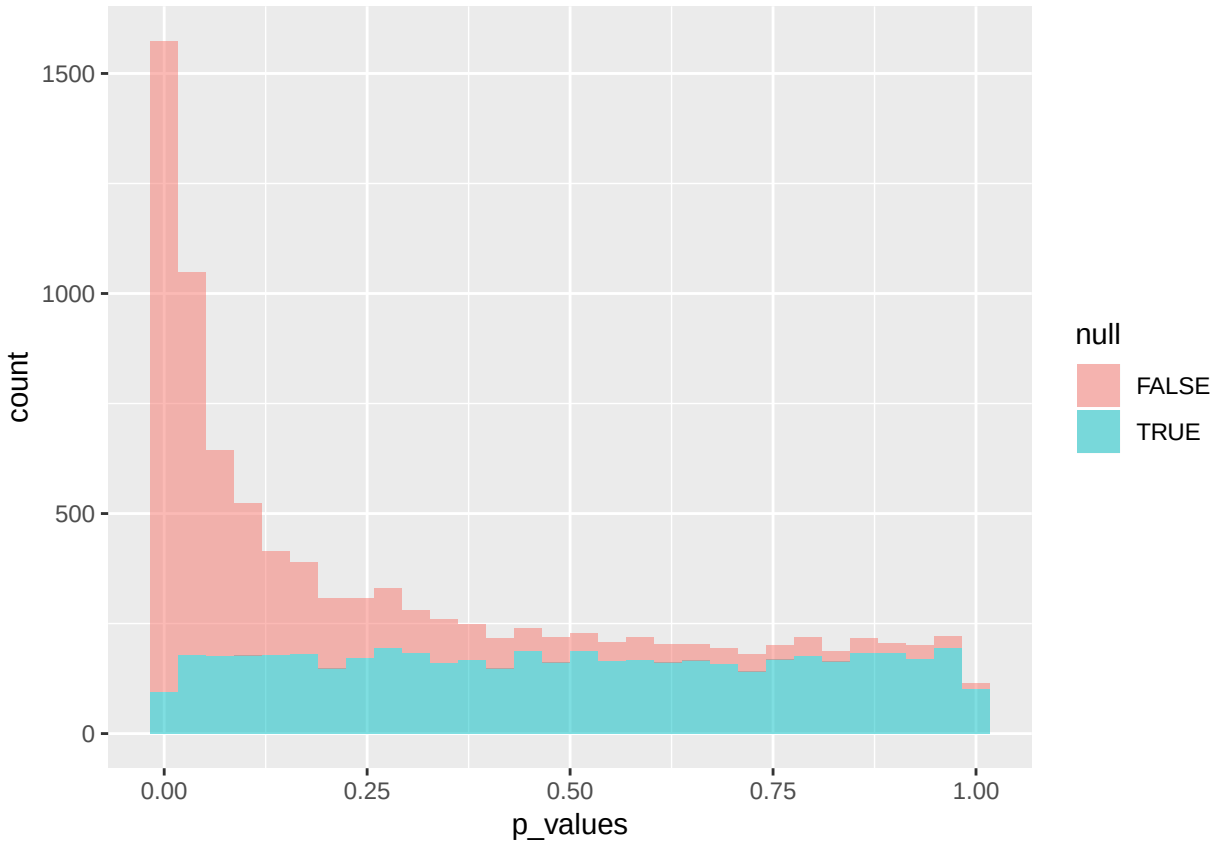



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
ggplot(data = p_df[p_df$null == FALSE, ], mapping = aes(x = p_values)) +
  stat_bin(bins = 30) +
  labs(title = "P-value when treatmen has MODERATE effect")
```



Comparing the two.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
# ggplot(mapping = aes(x = p_values)) +
#   stat_bin(data = p_df[p_df$null == TRUE, ], bins = 30, alpha = 0.5, fill = "blue") +
#   stat_bin(data = p_df[p_df$null == FALSE, ], bins = 30, alpha = 0.5, fill = "red")
ggplot(data = p_df, mapping = aes(x = p_values, fill = null)) +
  stat_bin(bins = 30, alpha = 0.5)
```



What if sample is small and data are non-Gaussian?

If your data is significantly non-gaussian, you would need a very large sample size for the t-statistic to actually be t-distributed.

In these cases, you can run a non-parametric test. Here's how we run a Mann-Whitney U test (aka Wilcoxon rank-sum test) using the `wilcox.test()` function.

```
birthwt.wilcox.test = wilcox.test(birthwt.grams ~ mother.smokes, data = birthwt, conf.int = TRUE)
birthwt.wilcox.test
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  birthwt.grams by mother.smokes
## W = 5249.5, p-value = 0.006768
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##   85.00004 512.00005
## sample estimates:
## difference in location
##               306.1846
```

Hypothesis tests in R return an object of class htest which has similar attributes to what we saw in the t-test.

```
class(birthwt.wilcox.test)
```

```
## [1] "htest"
```

Here's a summary of the attributes:

name	description
statistic	the value of the test statistic with a name describing it.
parameter	the parameter(s) for the exact distribution of the test statistic.
p.value	the p-value for the test.
null.value	the location parameter mu.
alternative	a character string describing the alternative hypothesis
method	the type of test applied.
data.name	a character string giving the names of the data.
conf.int	a confidence interval for the location parameter. (Only present if argument conf.int = TRUE.)
estimate	an estimate of the location parameter. (Only present if argument conf.int = TRUE.)

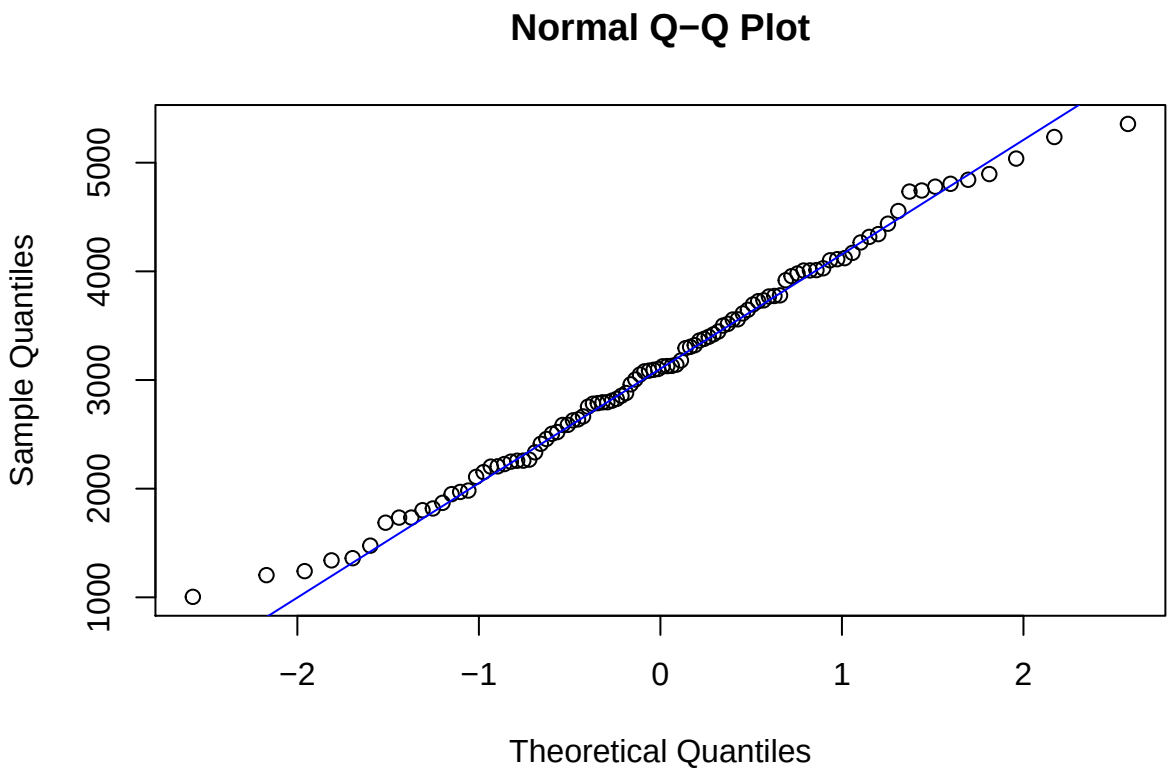
Is the data normal?

QQ-plots are a good way of assessing normality.

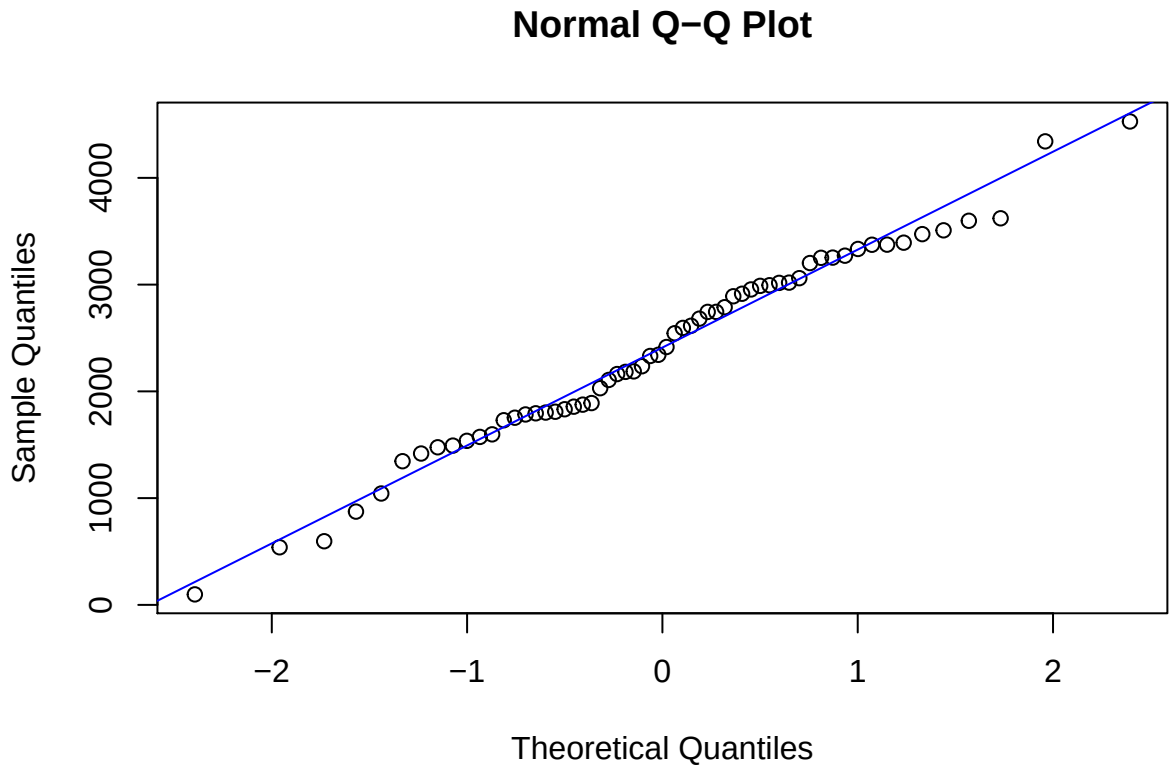
QQ-plot

The simplest thing to look at is a normal QQ-plot of the data. Use `qqnorm()` function to create the following plot.

```
normdat = rnorm(100) * 1000 + 3000
qqnorm(normdat)
qqline(normdat, col = "blue")
```

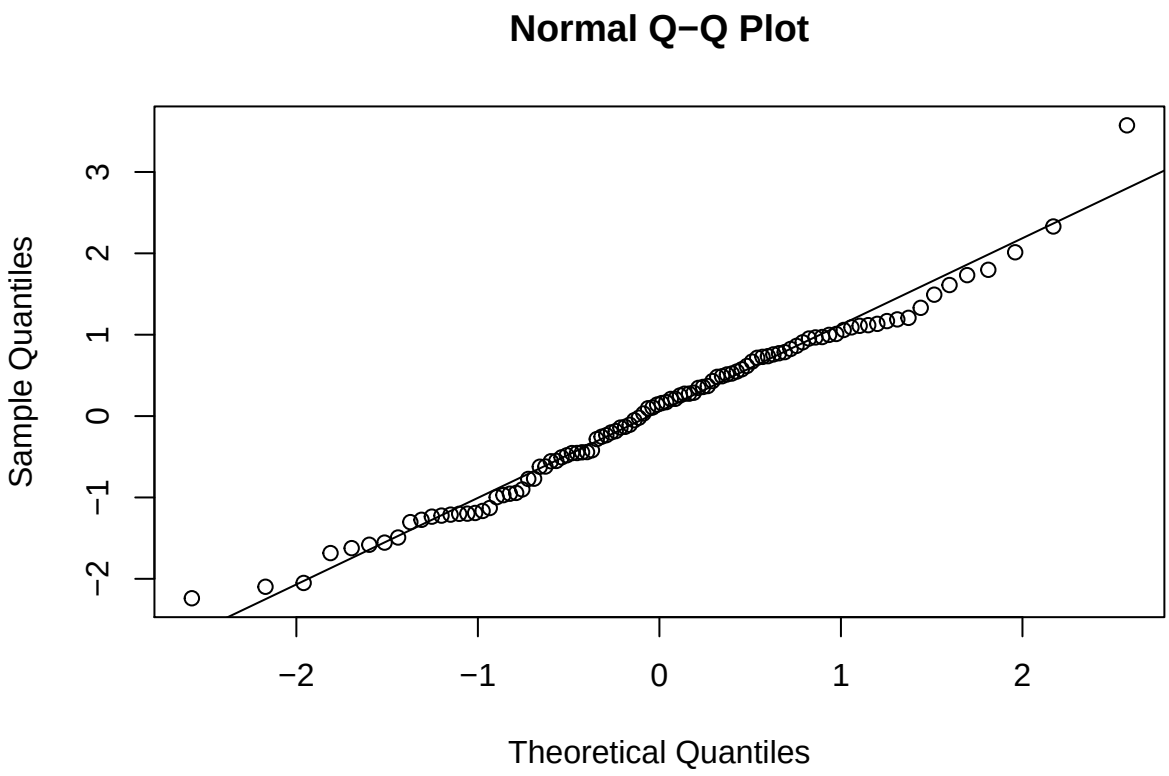


```r
normdat = rnorm(60) * 800 + 2500
qqnorm(normdat)
qqline(normdat, col = "blue")
```



Create a QQ-plot for a perfectly normal sample.

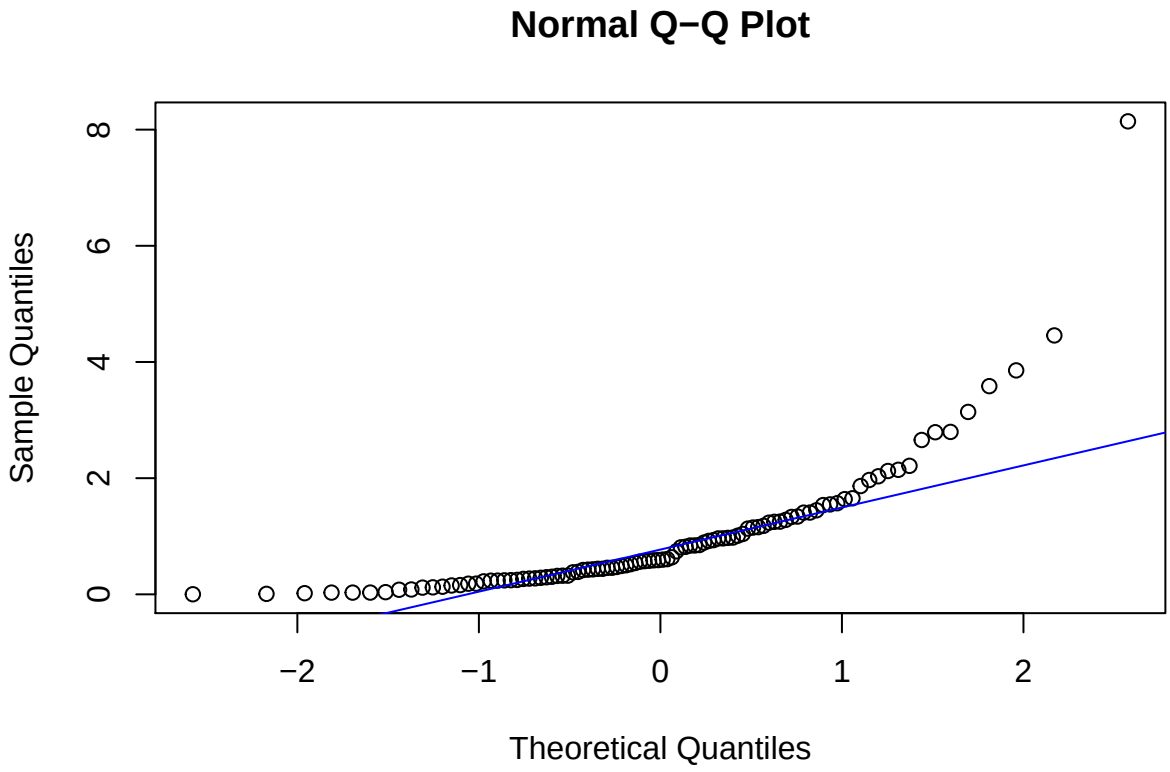
```r
normdat = rnorm(100)
qqnorm(normdat)
qqline(normdat)
```



If the data are exactly normal, you expect the points to lie on a straight line. The data we have here are pretty close to lying on a line.

Here's what we would see if the data were right-skewed

```
normdat = rexp(100)
qqnorm(normdat)
qqline(normdat, col = "blue")
```



Contingency Tables

Tests for 2x2 tables

Here's an example of a 2 x 2 table that we might want to run a test on. This one looks at low birthweight broken down by mother's smoking status. You can think of it as another approach to the t-test problem, this time looking at indicators of low birth weight, `birthwt.below.2500`, instead of the actual weights.

First, build a table using the `table()` function.

```
birthwt_smoke_contingency_table = table(birthwt[, c("birthwt.below.2500", "mother.smokes")])
birthwt_smoke_contingency_table
```

```
##                    mother.smokes
## birthwt.below.2500 no yes
##                  0 86  44
##                  1 29  30
```

We suspect a positive association between low birthweight and smoking status.

To test for significance, pass your 2 x 2 table into the appropriate function. Here's the result of using fisher's exact test by calling `fisher.test`

```
fishertest_results = fisher.test(birthwt_smoke_contingency_table)
fishertest_results
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  birthwt_smoke_contingency_table
```

```
## p-value = 0.03618
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.028780 3.964904
## sample estimates:
## odds ratio
##   2.014137
```

```
names(fishertest_results)
```

```
## [1] "p.value"     "conf.int"    "estimate"    "null.value"  "alternative"
## [6] "method"      "data.name"
```

```
class = class(fishertest_results)
```

As when using the t-test, we find that there is a significant association between smoking an low birth weight.

You can also use the chi-squared test via the `chisq.test` function.

```
chi2test_results = chisq.test(birthwt_smoke_contingency_table)
chi2test_results
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  birthwt_smoke_contingency_table
## X-squared = 4.2359, df = 1, p-value = 0.03958
```

You get essentially the same answer by running the chi-squared test, but the output isn't as useful. In particular, you're not getting an estimate or confidence interval for the odds ratio. You also might prefer `fisher.test()` for testing 2 x 2 tables.

Tests for j x k tables

Here's a small data set on party affiliation broken down by gender.

```
# Manually enter the data
politics <- as.table(rbind(c(762, 327, 468), c(484, 239, 477)))
dimnames(politics) <- list(gender = c("F", "M"),
                     party = c("Democrat","Independent", "Republican"))

politics # display the data
```

```
##       party
## gender Democrat Independent Republican
##      F      762         327        468
##      M      484         239        477
```

We may be interested in asking whether men and women have different party affiliations.

The answer will be easier to guess at if we convert the rows to show proportions instead of counts. Convert the table you created to this form:

```
politics.prop = prop.table(politics, 1)
politics.prop
```

```
##       party
## gender  Democrat Independent Republican
##      F 0.4894027   0.2100193  0.3005780
```

```
##      M 0.4033333   0.1991667  0.3975000
```

```
rowSums(politics.prop) # Check that columns sum to 1
```

```
## F M
## 1 1
```

By looking at the table we see that Female are more likely to be Democrats and less likely to be Republicans.

We still want to know if this difference is significant. To assess this we can use the chi-squared test. Will you use the counts or the proportions?

```
chisq.test(politics)
```

```
##
##  Pearson's Chi-squared test
##
## data:  politics
## X-squared = 30.07, df = 2, p-value = 2.954e-07
```

There isn't really a good one-number summary for general j x k tables the way there is for 2 x 2 tables.

Now remove the Independent category and just look at the 2 x 2 table showing the counts for the Democrat and Republican categories and perform a Fisher's Exact test.

```
politics_2x2 = politics[, c("Democrat", "Republican")]
politics_2x2
```

```
##       party
## gender Democrat Republican
##      F      762        468
##      M      484        477
```

```
fisher.test(politics_2x2)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  politics_2x2
## p-value = 6.806e-08
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.347341 1.910944
## sample estimates:
## odds ratio
##   1.604345
```

Conclusion: p-value is too small ($<< 0.05$) -> statistically significant.