Exploring data 2

Simple statistical tests in R

Statistical tests

R has many different functions for different statistical tests.

So far, I have not found a statistical test I wanted to perform that did not have a function to do it in R.

Statistical tests

We'll start with a pretty simple test called the "Shapiro-Wilk test of normality."

The **null hypothesis** for this test is that the data follow a normal distribution. If the p-value from running the test is lower than a specified threshold (often 0.05), then you reject the null hypothesis.

Statistical tests

Often, a parametric test will require the assumption that one or more variables follow a normal distribution. This test can help check that assumption.

If you find that the distribution does not seem to be normal, you can make adjustments, like transforming the variable or using a non-parametric test (see here for a nice overview).

Statistical tests

We'll use this test as an example, but it has two characteristics that are common to most simple statistical tests functions in R:

1. It inputs a vector (or vectors, for some tests), rather than a tidy dataframe.
2. It outputs a list object, rather than a dataframe.

We'll talk about both of these characteristics and how we can still work statistical tests into a "tidy" workflow, where we're keeping the data in a tidy dataframe format most of the time.

Statistical tests

To show how this test works, let's simulate some data that we know are normal.

There are several functions in R that let you simulate a vector of data from a specified distribution. For a normal distribution, the function is `rnorm`.

For this function, you'll specify how many values you want in the vector (`n`) and the values of the mean (`mean`) and standard deviation (`sd`) of the normal distribution.

Statistical tests

For example, you can run this to create a vector with a random sample of 1,000 values from a normal distribution with mean 200 and standard deviation 50:

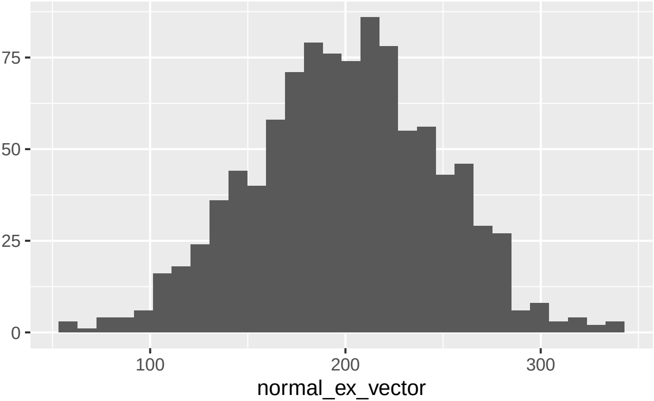
```
normal_ex_vector <- rnorm(n = 1000, mean = 200, sd = 50)
head(normal_ex_vector)
```

```
## [1] 136.5821 260.3263 239.6398 178.3469 219.4890 182.1706
```

Statistical tests

Let's check the histogram of this data. Since this data in in a vector, not a dataframe, we should use `qplot` from `ggplot2` instead of `ggplot`:

```
qplot(normal_ex_vector, geom = "histogram")
```



Statistical tests

It looks pretty normal, but let's run the test. The function to run the test is shapiro.test. Its only argument is x, the vector that you want to check.

```
shapiro.test(x = normal_ex_vector)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  normal_ex_vector
## W = 0.99834, p-value = 0.4574
```

Statistical tests

The default for this function is to print out the results in a way that's easy to read on the screen. However, you can also save the results from the test in a new object:

```
ex_sw_result <- shapiro.test(x = normal_ex_vector)
```

Statistical tests

This object has a special class called “htest”:

```
class(ex_sw_result)
```

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

Really, though, this is just a special kind of list:

```
is.list(ex_sw_result)
```

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

Statistical tests

If you run str on the object, you can see it has all the information that is usually printed by the function tucked away in different slots of the list:

```
str(ex_sw_result)
```

```
## List of 4
##  $ statistic: Named num 0.998
##   ..- attr(*, "names")= chr "W"
##  $ p.value  : num 0.457
##  $ method   : chr "Shapiro-Wilk normality test"
##  $ data.name: chr "normal_ex_vector"
##  - attr(*, "class")= chr "htest"
```

Statistical tests

There is a package called broom that can pull this information out and format it as a tidy dataframe. For example:

```r
library(broom)
tidy(ex_sw_result)
```

```
## # A tibble: 1 x 3
##   statistic p.value method
##       <dbl>   <dbl> <chr>
## 1     0.998   0.457 Shapiro-Wilk normality test
```

This seems unexciting for this example, but trust me, it turns out that being able to do this is **very** exciting.