Exploring data 2

Simple statistical tests in a tidy framework

Statistical tests

Nex, let's take a look at how the same process would work if you were starting with a vector that was a column in a tidy dataframe. Since you're using tidyverse tools, you'll probably find you want to do this often.

We'll create a very simple dataframe with only this column:

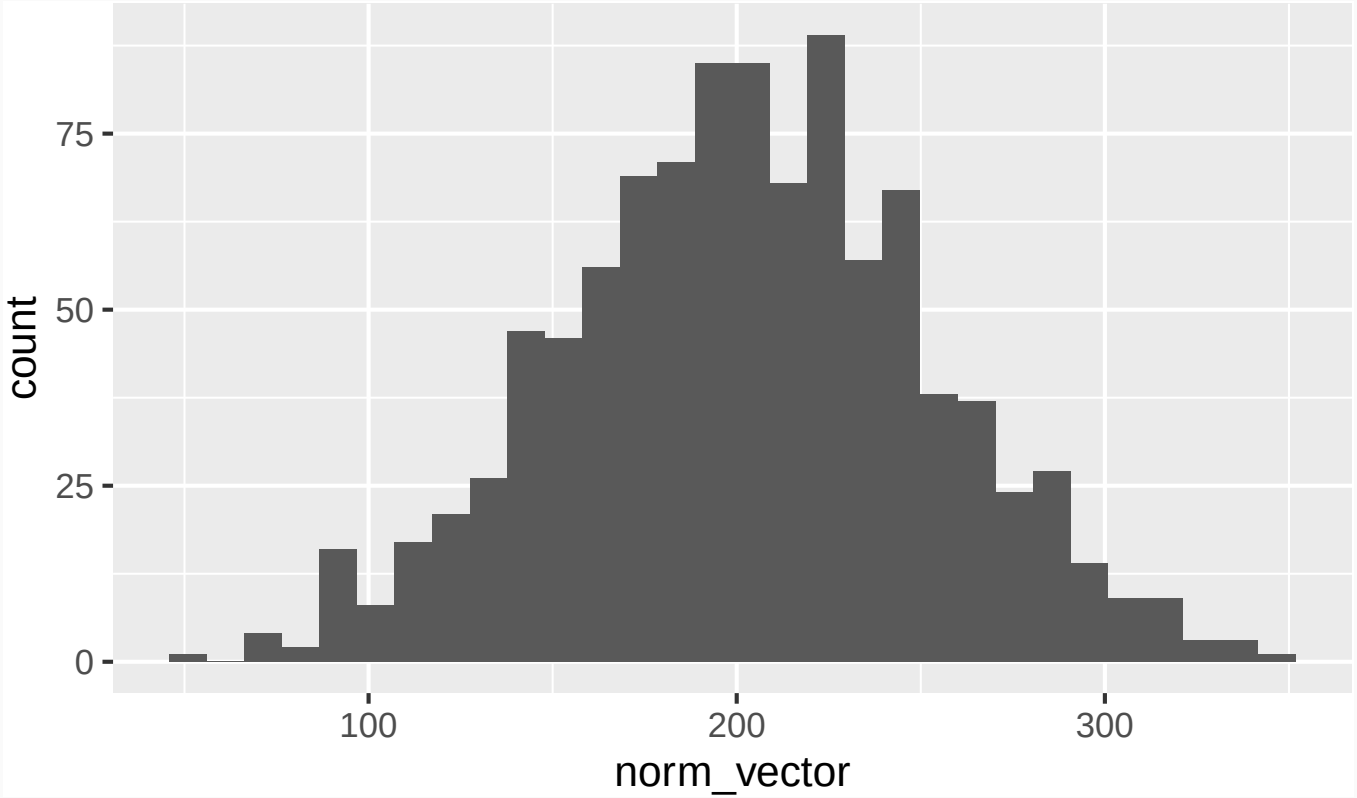
```
normal_ex_vector <- rnorm(n = 1000, mean = 200, sd = 50)
ex_df <- tibble(norm_vector = normal_ex_vector)
ex_df %>%
  slice(1:3)
```

```
## # A tibble: 3 x 1
##   norm_vector
##         <dbl>
## 1        268.
## 2        190.
## 3        136.
```

Statistical tests

Now you can use ggplot to make the histogram:

```
ggplot(ex_df, aes(x = norm_vector)) +
  geom_histogram()
```



Statistical tests

To fit the test, you'll need to be able to pull this vector out of the dataframe. To do that, you can use the `pull` function from the `dplyr` package in a pipeline. That function "pulls" out a single column as a vector. For example:

```
ex_df %>%
  pull("norm_vector") %>%
  head()
```

```
## [1] 267.9013 189.5794 136.2943 180.1419 311.5319 134.5041
```

Statistical tests

With that function, you can pipe right into the Shapiro-Wilk test function:

```
ex_df %>%
  pull("norm_vector") %>%
  shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.99868, p-value = 0.6741
```

Statistical tests

Now just add on the `tidy` function to get the test output in a tidy dataframe, and you're back to your typical format!

```
library(broom)
ex_df %>%
  pull("norm_vector") %>%
  shapiro.test() %>%
  tidy()
```

```
## # A tibble: 1 x 3
##   statistic p.value method
##       <dbl>   <dbl> <chr>
## 1     0.999   0.674 Shapiro-Wilk normality test
```

Statistical tests

Now let's look at some real data. The variable dataframe of the `atlas1006` dataset in the `microbiome` library has a column on `diversity`.

We might want to test if diversity is different by gender, nationality, or other factors. To pick which statistical tests to use to check those questions, though, it will help to know if this variable is normally distributed.

Statistical tests

The `atlas1006` data is stored in a `phyloseq` object (think of it as a fancy type of list). To extract a dataframe with characteristics of the samples, you'll need to use `get_variable` (which we can pipe into if we want):

```
library(microbiome)
data(atlas1006)
atlas1006 %>%
  get_variable() %>%
  slice(1:3)
```

```
##          age    sex nationality DNA_extraction_method project
## Sample-1  28   male          US                   <NA>       1
## Sample-2  24 female          US                   <NA>       1
## Sample-3  52   male          US                   <NA>       1
##            bmi_group subject time   sample
## Sample-1 severeobese       1    0 Sample-1
## Sample-2       obese       2    0 Sample-2
## Sample-3        lean       3    0 Sample-3
```

Statistical tests

There are a few people that they measure several times, so there are more rows than the number of people they measure:

```
atlas1006 %>%
  get_variable() %>%
  nrow()
```

```
## [1] 1151
```

Statistical tests

We probably just want to work with the first measurement from each person, so let's use `filter` to filter to samples with a "time" value of 0 (first measurement):

```
atlas1006 %>%
  get_variable() %>%
  filter(time == 0) %>%
  nrow()
```

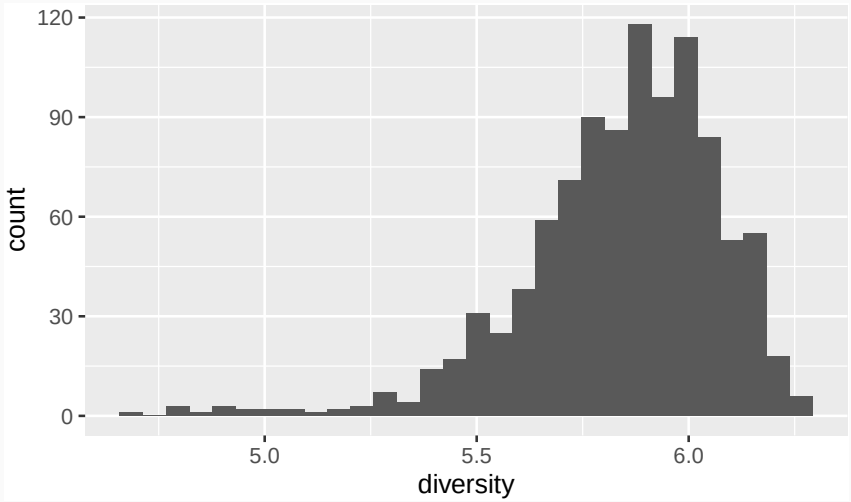
```
## [1] 1006
```

This looks right.

Statistical tests

We can use a histogram to visually check the normality:

```
atlas1006 %>%
  get_variable() %>%
  filter(time == 0) %>%
  ggplot(aes(x = diversity)) +
  geom_histogram()
```



Statistical tests

To extract the column on `diversity` as a vector, we can use `pull`:

```
atlas1006 %>%
  get_variable() %>%
  filter(time == 0) %>%
  pull("diversity") %>%
  head()
```

```
## [1] 5.76 6.06 5.50 5.87 5.89 5.53
```

Statistical tests

Now add on the Shapiro test function:

```
atlas1006 %>%
  get_variable() %>%
  filter(time == 0) %>%
  pull("diversity") %>%
  shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.93439, p-value < 2.2e-16
```

Statistical tests

And finally add on the `tidy` function:

```
atlas1006 %>%
  get_variable() %>%
  filter(time == 0) %>%
  pull("diversity") %>%
  shapiro.test() %>%
  tidy()
```

```
## # A tibble: 1 x 3
##   statistic  p.value method
##       <dbl>    <dbl> <chr>
## 1     0.934 1.29e-20 Shapiro-Wilk normality test
```

Find out more about statistical tests in R

I won't be teaching in this course how to find the correct statistical test. That's something you'll hopefully learn in a statistics course.

There are also a variety of books that can help you with this, including some that you can access free online through CSU's library. One servicable introduction is "Statistical Analysis with R for Dummies".