Webexercises template

This is a Web Exercise template created by the psychology teaching team at the University of Glasgow, based on ideas from Software Carpentry. This template shows how instructors can easily create interactive web documents that students can use in self-guided learning.

The {webexercises} package provides a number of functions that you use in inline R code or through code chunk options to create HTML widgets (text boxes, pull down menus, buttons that reveal hidden content). Examples are given below. Render this file to HTML to see how it works.

NOTE: To use the widgets in the compiled HTML file, you need to have a JavaScript-enabled browser.

Example Questions

Fill-In-The-Blanks (`fitb()`)

Create fill-in-the-blank questions using `fitb()`, providing the answer as the first argument.

- 2 + 2 is __

You can also create these questions dynamically, using variables from your R session.

```
x <- sample(2:8, 1)
```

- The square root of 16 is: __

The blanks are case-sensitive; if you don't care about case, use the argument `ignore_case = TRUE`.

- What is the letter after D? __

If you want to ignore differences in whitespace use, use the argument `ignore_ws = TRUE` (which is the default) and include spaces in your answer anywhere they could be acceptable.

- How do you load the tidyverse package? _____

You can set more than one possible correct answer by setting the answers as a vector.

- Type a vowel: _

You can use regular expressions to test answers against more complex rules.

- Type any 3 letters: ____

Multiple Choice (`mcq()`)

- "Never gonna give you up, never gonna:

- (A) let you go

- (B) turn you down

- (C) run away

- (D) let you down

" - "I

- (A) bless the rains

- (B) guess it rains

- (C) sense the rain

down in Africa" -Toto

True or False (`torf()`)

- True or False? You can permute values in a vector using `sample()`. TRUE / FALSE

Longer MCQs (`longmcq()`)

When your answers are very long, sometimes a drop-down select box gets formatted oddly. You can use `longmcq()` to deal with this. Since the answers are long, It's probably best to set up the options inside an R chunk with `echo=FALSE`.

What is a p-value?

```
opts_p <- c(
    "the probability that the null hypothesis is true",
    answer = "the probability of the observed, or more extreme, data, under the assump
    "the probability of making an error in your conclusion"
)
```

- (A) the probability that the null hypothesis is true

- (B) the probability of the observed, or more extreme, data, under the assumption that the null-hypothesis is true

- (C) the probability of making an error in your conclusion

What is true about a 95% confidence interval of the mean?

```
# use sample() to randomise the order
opts_ci <- sample(c(
  answer = "if you repeated the process many times, 95% of intervals calculated in th
  "there is a 95% probability that the true mean lies within this range",
  "95% of the data fall within this range"
))
```

- (A) if you repeated the process many times, 95% of intervals calculated in this way contain the true mean

- (B) 95% of the data fall within this range

- (C) there is a 95% probability that the true mean lies within this range

Checked sections

Create sections with the class `webex-check` to add a button that hides feedback until it is pressed. Add the class `webex-box` to draw a box around the section (or use your own styles).

I am going to learn a lot: TRUE / FALSE

```r
opts <- c(
    "the probability that the null hypothesis is true",
    answer = "the probability of the observed, or more extreme, data, under the assumpt
    "the probability of making an error in your conclusion"
)

cat("What is a p-value?", longmcq(opts))
```

What is a p-value?

- (A) the probability that the null hypothesis is true

- (B) the probability of the observed, or more extreme, data, under the assumption that
  the null-hypothesis is true

- (C) the probability of making an error in your conclusion

Hidden solutions and hints

You can fence off a solution area that will be hidden behind a button using `hide()` before the solution and `unhide()` after, each as inline R code. Pass the text you want to appear on the button to the `hide()` function.

If the solution is a code chunk, instead of using `hide()` and `unhide()`, simply set the `webex.hide` chunk option to TRUE, or set it to the string you wish to display on the button.

Recreate the scatterplot below, using the built-in `cars` dataset.

```r
with(cars, plot(speed, dist))
```
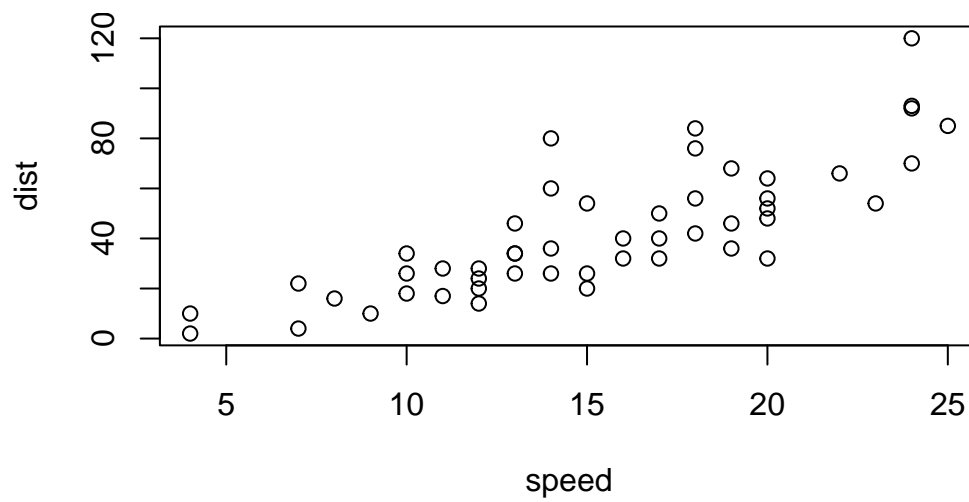
I need a hint

See the documentation for `plot()` (`?plot`)

Click here to see the solution

```
plot(cars$speed, cars$dist)
```