STAT 331 Lab 11

Instructions

Submit an HTML document by the beginning of class.

Exercise

Now that we're officially equipped with IF-statements, let's create a more robust and powerful hypothesis test function!

Your task is to write a function *hyp_test* which performs a one-sample hypothesis test about either a mean or a proportion. Your function should take the following arguments:

- data - a vector of numeric or factor values (this is the sample of data). This sample can have missing (NA) values in it. If numeric, the function will perform a one-sample t-test for a mean. If factor, the function will perform a one-sample z-test for a proportion. For a z-test, the sample proportion will be the proportion of data in the first factor level.
- null - a single numeric value (this is the hypothesized value).
- alpha - a single numeric value (this is the significance level). This should default to 0.05.
- alternative - a character string specifying the form of the alternative hypothsis ("less", "greater", "two-sided"). This should default to "two-sided".

Value

Your function should ignore any missing values in the data and return a list with the following components:

- statistic : the value of the z- or t-statistic
- df : degrees of freedom if appropriate
- p.value : the p-value for the test
- conf.int : a confidence interval for the proportion (or mean) appropriate to the specified alpha
- estimate : the estimated proportion (or mean) based on the data
- null.value : the specified hypothesized value of the proportion (or mean)
- alpha : the specified significance level

Display

Besides returning the items listed above, your function should print the following:

- The null hypothesis
- The value of the test statistic and the p-value
- The confidence interval for the proportion (or mean)

NOTE 1: Your function should perform a check to make sure the null hypothesis value is between 0 and 1 for the one proportion test; otherwise, your function should return an error.

NOTE 2: You may NOT use R's t.test() except to check your work.

Example

Test your code in AT LEAST the following 5 ways.

```
#TEST 1
data <- c(NA, 5:25)
hyp_test(data, null = 16, alpha = .05, alternative = "two-sided")
```

```
## Ho: mu = 16
```

```
## Test Statistic: -0.74 , p-value: 0.4688
```

```
## Confidence Interval: (12.18,17.82)
```

```
## $statistic
## [1] -0.7385489
##
## $df
## [1] 20
##
## $p.value
## [1] 0.4687599
##
## $conf.int
## [1] 12.17559 17.82441
##
## $estimate
## [1] 15
##
## $null.value
## [1] 16
##
## $alpha
## [1] 0.05
```

```
#TEST 2
data <- factor(c(NA, rep("a", 60), rep("b", 40)))
hyp_test(data, null = .5, alpha = .01, alternative = "greater")
```

```
## Ho: p = 0.5
```

```
## Test Statistic: 2 , p-value: 0.0228
```

```
## Confidence Interval: (0.4738,0.7262)
```

```
## $statistic
## [1] 2
##
## $p.value
## [1] 0.02275013
##
```

```
## $conf.int
## [1] 0.4738107 0.7261893
##
## $estimate
## [1] 0.6
##
## $null.value
## [1] 0.5
##
## $alpha
## [1] 0.01
```

```
# TEST 3
data <- factor(c(NA, rep("a", 60), rep("b", 40)))
hyp_test(data, null = 1.4, alpha = .01, alternative = "greater")
```

```
## Error: invalid hypothesized value. Must be between 0 and 1
```

```
## [1] NA
```

```
# TEST 4
data <- 1:10
hyp_test(data, null = 6, alpha = .101, alternative = "greater")
```

```
## Ho: mu = 6
```

```
## Test Statistic: -0.52 , p-value: 0.6929
```

```
## Confidence Interval: (3.75,7.25)
```

```
## $statistic
## [1] -0.522233
##
## $df
## [1] 9
##
## $p.value
## [1] 0.6929414
##
## $conf.int
## [1] 3.750928 7.249072
##
## $estimate
## [1] 5.5
##
## $null.value
## [1] 6
##
## $alpha
## [1] 0.101
```

```
# TEST 5
data <- factor(c(NA, rep("a", 60), rep("b", 40)))
hyp_test(data, null = 0.70, alpha = .02, alternative = "less")
```

```
## Ho: p = 0.7
```

```
## Test Statistic: -2.18 , p-value: 0.0145
```

```
## Confidence Interval: (0.486,0.714)
```

```
## $statistic
## [1] -2.182179
##
## $p.value
## [1] 0.01454817
##
## $conf.int
## [1] 0.4860327 0.7139673
##
## $estimate
## [1] 0.6
##
## $null.value
## [1] 0.7
##
## $alpha
## [1] 0.02
```