C2M4_autograded

February 18, 2022

1 Module 4 - Autograded

1.0.1 Outline:

Here are the objectives of this assignment:

1.

Here are some general tips:

1. Read the questions carefully to understand what is being asked.
2. When you feel that your work is completed, feel free to hit the `Validate` button to see your results on the *visible* unit tests. If you have questions about unit testing, please refer to the "Module 0: Introduction" notebook provided as an optional resource for this course. In this assignment, there are hidden unit tests that check your code. **Do not misinterpret the feedback from visible unit tests as all possible tests for a given question–write your code carefully!**
3. Before submitting, we recommend restarting the kernel and running all the cells in order that they appear to make sure that there are no additional bugs in your code.

[20]:
```
# Load Required Libraries
library(testthat)
```

2 Problem 1

This question is to get you thinking about different parts of each experiment. You will answer multiple choice questions about a hypothetical situation. For each question #, fill in the corresponding `prob.1.#` with the correct corresponding answer number. Consider a the following experiment:

You want to evaluate the effects of water from different streams on fish lesions. You set up 2 sets of 3 aquaria (6 total), each with 50 fish. One set of 3 aquaria will have Red Fish, and the other set of 3 aquaria will have Blue fish. Other than the type of fish, the sets are identical. You randomly assign a water treatment (Stream A vs. Stream B vs. control) to each of the aquaria in each set. After 1 month, you catch 10 fish from each aquarium and count the number of lesions.

2.0.1 1. (a) Experimental Unit

In the above experiment, what is the experimental unit?

1. A fish
2. An aquarium
3. The water used
4. 10 fish
5. None of the above

[21]:
```
prob.1.a = NA

# your code here
prob.1.a = 2
```

[22]:
```
# Test Cell
if (!expect_is(prob.1.a, "numeric")){
    print("Make sure your answers are numeric.")
}
# This cell has hidden test cases that will run after submission.
```

2.0.2 1. (b) Sample Unit

What is the sample unit in the experiment?

1. A fish
2. An aquarium
3. The water used
4. 10 fish
5. None of the above

[34]:
```
prob.1.b = NA

# your code here
prob.1.b = 1
```

[24]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

2.0.3 1. (c) Replication

How many times was this experiment replicated?

1. 1 time
2. 2 times
3. 3 times
4. 0 times.
5. None of the above.

[38]:
```
prob.1.c = NA

# your code here
prob.1.c = 4
```

[26]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

2.0.4 1. (d) Experimental Design

What kind of experiment is this?

1. Completely Randomized Design
2. Randomized Complete Block Design
3. Factorial Design
4. None of the above

[39]:
```
prob.1.d = NA

# your code here
prob.1.d = 2
```

[28]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

3 Problem 2

In the 1800's, some physicists proposed that all electromagnetic and graviational forces required a "medium" for the forces to be applied through. This medium was called "the aether" and filled all space. The Michelson-Morley experiment was one of the first experiments that aimed to test this theory. In the experiment, the scientists measured the speed of light shown in perpendicular directions, one in the direction of the aether and one against the aether. We can can use their data, along with our new understanding of blocking, to determine if there actually is an aether!

The data from that experiment is loading in below. Information about this dataset can be found here.

[29]:
```
# Load in and clean the data
data(morley)
head(morley)

morley$Expt = as.factor(morley$Expt)
morley$Run = as.factor(morley$Run)
```

A data.frame: 6 × 3

	Expt <int>	Run <int>	Speed <int>
001	1	1	850
002	1	2	740
003	1	3	900
004	1	4	1070
005	1	5	930
006	1	6	850

3.0.1 2. (a) Testing Aether Thoery

The Michelson-Morley data can be viewed as a randomized block experiment with `Run` as the treatment factor and `Expt` as the blocking factor. Model this relationship and store your model as `morley.lmod`.

Determine if there is a difference between the speed of light among different runs. Save your answer in `is.different`, being `TRUE` if there is a noticable difference among runs and `FALSE` if there is not.

[45]:
```
morley.lmod = NA
is.different = NA

# your code here
morley.lmod= lm(Speed ~ Expt + Run, data=morley)
summary(morley.lmod)
```

```
Call:
lm(formula = Speed ~ Expt + Run, data = morley)

Residuals:
    Min      1Q  Median      3Q     Max
-206.60  -37.35    4.90   44.27  132.90

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.506e+02  3.599e+01  26.413  < 2e-16 ***
Expt2       -5.300e+01  2.323e+01  -2.281 0.025325 *
Expt3       -6.400e+01  2.323e+01  -2.755 0.007343 **
Expt4       -8.850e+01  2.323e+01  -3.810 0.000281 ***
Expt5       -7.750e+01  2.323e+01  -3.336 0.001317 **
Run2        -5.200e+01  4.646e+01  -1.119 0.266588
Run3        -2.800e+01  4.646e+01  -0.603 0.548545
Run4         6.000e+00  4.646e+01   0.129 0.897591
Run5        -7.600e+01  4.646e+01  -1.636 0.106032
Run6        -1.040e+02  4.646e+01  -2.238 0.028125 *
Run7        -1.000e+02  4.646e+01  -2.152 0.034551 *
Run8        -4.000e+01  4.646e+01  -0.861 0.391996
Run9        -1.000e+01  4.646e+01  -0.215 0.830167
```

```
Run10        -3.800e+01  4.646e+01  -0.818 0.415992
Run11         4.000e+00  4.646e+01   0.086 0.931621
Run12        -7.111e-14  4.646e+01   0.000 1.000000
Run13        -3.600e+01  4.646e+01  -0.775 0.440851
Run14        -9.400e+01  4.646e+01  -2.023 0.046576 *
Run15        -6.000e+01  4.646e+01  -1.291 0.200492
Run16        -6.600e+01  4.646e+01  -1.420 0.159552
Run17        -6.000e+00  4.646e+01  -0.129 0.897591
Run18        -3.800e+01  4.646e+01  -0.818 0.415992
Run19        -5.000e+01  4.646e+01  -1.076 0.285271
Run20        -4.400e+01  4.646e+01  -0.947 0.346641
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 73.46 on 76 degrees of freedom
Multiple R-squared:  0.3363,Adjusted R-squared:  0.1355
F-statistic: 1.675 on 23 and 76 DF,  p-value: 0.04956
```

[44]:
```
is.different = FALSE
anova(morley.lmod)
```

A anova: 3 × 5

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
Expt	4	94514	23628.500	4.378144	0.003070589
Run	19	113344	5965.474	1.105348	0.363209341
Residuals	76	410166	5396.921	NA	NA

[46]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```

3.0.2 2. (b) Relative Efficiency

Relative efficiency is defined as:

$$e = \widehat{\sigma}^2_{CRD} \Big/ \widehat{\sigma}^2_{RCBD}$$

This value tells us how many more observations we would need for a CRD to obtain the same level of precision as a RCBD. For example, if $e = 1.66$, then a CRD would require 66% more observations to obtain the same level of precision as an RCBD.

Calculate the the relative efficiency for your model. Store your answer in a variable called e.

[47]:
```
e = NA

# your code here
```

```
lmod2 = lm(Speed ~ Run, data=morley)
anova(lmod2)
```

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
Run	19	113344	5965.474	0.9456247	0.531329
Residuals	80	504680	6308.500	NA	NA

[48]:
```
e = 6308.500 / 5396.921
e
```

1.16890723432861

[49]:
```
summary(lmod2)
```

```
Call:
lm(formula = Speed ~ Run, data = morley)

Residuals:
   Min     1Q Median     3Q    Max 
-174.0  -48.5   -4.0   48.5  170.0 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.940e+02  3.552e+01  25.169   <2e-16 ***
Run2        -5.200e+01  5.023e+01  -1.035   0.3037    
Run3        -2.800e+01  5.023e+01  -0.557   0.5788    
Run4         6.000e+00  5.023e+01   0.119   0.9052    
Run5        -7.600e+01  5.023e+01  -1.513   0.1342    
Run6        -1.040e+02  5.023e+01  -2.070   0.0416 *  
Run7        -1.000e+02  5.023e+01  -1.991   0.0499 *  
Run8        -4.000e+01  5.023e+01  -0.796   0.4282    
Run9        -1.000e+01  5.023e+01  -0.199   0.8427    
Run10       -3.800e+01  5.023e+01  -0.756   0.4516    
Run11        4.000e+00  5.023e+01   0.080   0.9367    
Run12        2.187e-14  5.023e+01   0.000   1.0000    
Run13       -3.600e+01  5.023e+01  -0.717   0.4757    
Run14       -9.400e+01  5.023e+01  -1.871   0.0650 .  
Run15       -6.000e+01  5.023e+01  -1.194   0.2358    
Run16       -6.600e+01  5.023e+01  -1.314   0.1926    
Run17       -6.000e+00  5.023e+01  -0.119   0.9052    
Run18       -3.800e+01  5.023e+01  -0.756   0.4516    
Run19       -5.000e+01  5.023e+01  -0.995   0.3226    
Run20       -4.400e+01  5.023e+01  -0.876   0.3837    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 79.43 on 80 degrees of freedom
```

```
Multiple R-squared:  0.1834,Adjusted R-squared:  -0.01055
F-statistic: 0.9456 on 19 and 80 DF,  p-value: 0.5313
```

[33]:
```
# Test Cell
# This cell has hidden test cases that will run after submission.
```