Name____________________________

Section__________________________

DEPARTMENT OF STATISTICS
**The Wharton School**
**University of Pennsylvania**

Statistics 405/705 Spring 2019

# QUIZ #4

## Instructions

1. Place your name and section at the top of this page.
2. Keep this side up until you are told to begin.
3. This is a closed-book quiz. You may use a calculator.
4. Circle the single best answer to each question.
5. You will have exactly 10 minutes to complete the quiz.

1. The vector `v` of length 100 contains logical values (`TRUE/FALSE)` with no `NAs`. What's the difference in terms of functionality between the two R expressions:

```
any(v)
```

and

```
sum(v) > 0
```

A. The first expression is suitable as a condition for an "`if`" statement, whereas the second one is not.

B. The second expression will throw an error if there are any FALSE's in the vector `v`.

**C. They are functionally equivalent.**

D. The second expression will return a numeric value, whereas the first returns a logical value.

2. Below is a simple if/else code snippet. Which statement below is a valid comment on the code?

```
if(x > 10){
    print("This number is big")
}
else{
    print("This number is small")
}
```

A. So long as x is numeric, there will be no problem with running the code.

**B. It's going to throw an error when evaluated.**

C. For the special case x = NA, the else block will always be evaluated.

D. The code will run, but neither block will be evaluated.

3. Which of the following commands will immediately exit from a `for` loop?

A. `next`

B. `goto`

C. `jump`

**D. `break`**

4. In a Monte-Carlo simulation, what is a key difference between the use of a `whil`e loop and a `for` loop?

A. `for` loops run faster.

B. `while` loops are less likely to get caught in an infinite loop.

**C. You know the maximum number of iterations of a `for` loop, but you don't know the maximum number of iterations for a `while` loop.**

D. `for` loops iterate over vectors, whereas `while` loops iterate over lists.

5. Which of the following three expressions always results in exactly the same answer each time it is run, even after having set a different random number seed? (The function `sort`, sorts a vector into ascending order.)

```
(i)   sort(sample(x = 10))
(ii)  order(sample(x = 10))
(iii) sort(sample(x = 10, replace = TRUE))
```

**A. (i) Not that sample with a single argument returns a random permutation, but sorting that will always give the same answer, the numbers 1 through 10.**

B. (i) and (iii)

C. (iii)

D. (ii)