A Functional Approach to Teaching Statistics

Stephen M. Mansour
103 Yesu Drive, Scranton, PA 18505

Abstract
Much of statistics can be organized into four types of functions: summary functions, distributions, relational and logical functions. A summary function compresses a set of values into a single item which describes the original set. A distribution function associates a likelihood with a particular input. Relational and logical functions produce truth values defined as 0 and 1 in Iverson notation and the computer language APL.

Operators are higher-order functions which modify or combine existing functions to produce new functions, thus reducing the statistical vocabulary required of students. Their formal definition is defined in [Iverson 1979]. Confidence Intervals, hypothesis tests, probabilities and simulation can be handled by applying various operators to these four types of functions.

For example, the expression: "`normal probability > 1.96`" will produce the value `0.025`. The operator, "`probability`" combines the distribution "`normal`" with the relation "`>`" to form a new function: the upper-tail cumulative normal distribution applied to the argument "`1.96`".

Key Words: Statistical Education, Function, APL

1. Data Representation

We represent data in one of three ways. The most comprehensive way is a list of all values in a sample or population. We will refer to this list as a *vector*. A second representation is a frequency distribution, which is a two-column matrix whose first column consists of specific values and whose second column consists of their frequencies. For discrete data, there is no loss of information. However, for continuous data, intervals are more useful. In this case the values column consists of the midpoint of each interval; this is because the midpoint is more representative of the interval than the lower or upper endpoint. A third way to represent data is with a namespace containing sample statistics, i.e. the sample size, mean and standard deviation. A namespace is an object which may contain variables, functions and other namespaces [Scholes 1994]. Although there is often some loss of information with frequency distributions and statistical data, all three types of data representation can be used as inputs to various families of functions.

1.1 Examples of Data Representation

Here is an example of a variable showing a sample of student heights:

```
Height←72 62 69 69 70 66 67 72 72 71 71 73 61 65 69 65 73 67 70 68
```

A frequency distribution of heights can be arranged as a two-column matrix where each row represents an interval of heights. The first column represents a particular value or the midpoint of the interval if a continuous variable is involved. We use the midpoint rather than the lower bound because it is more representative of the interval. The second column contains either the frequency, weight or probability of each interval. Since weight is a continuous variable, we could represent it as shown below, where the second column represents the count of each interval:

```
     FreqHeight
61.5 2
64.5 2
67.5 4
70.5 7
73.5 5
```

The value in the second row, `64.5,` is the midpoint between the interval from 63 and 66 inches inclusive of the lower bound. Two values fall in that interval.

The third representation of a variable is a namespace containing three values: count, sample mean and sample standard deviation:

```
     NameSpace
Height: [n=20; xbar=68.6 s=3.47]
```

2. Functions

Functions can be either monadic with only a right argument, e.g. `ln x` or `sin x`; or dyadic with both left and right arguments, e.g. `+, ×, -, ÷`. Some functions can have an optional left argument which defaults to some common value. For example, the reciprocal function is a monadic version of division with an implied left argument of 1. Statistical examples include confidence intervals defaulting to 95% and normal parameters defaulting to $\mu = 0, \sigma = 1$. Functions can take arguments which are in the form of scalars, vectors or matrices, and produce results in similar formats. For a more complete description see [Falkoff 1973]. The four types of statistical functions described below follow all these rules.

2.1 Summary Functions

I often ask each I often ask each student at the start of the fall semester to summarize what he or she did over the summer vacation, not with the typical essay, but with a single word. "Work", fun" or "beach" come to mind. This is precisely what summary functions do; they produce a single value from a long list of values:

$$y = f(x_1, x_2 \ldots, x_n)$$

Summary functions are ways to measure data sets. They can be subdivided into various types:

- Measures of quantity: `sum, product, count, sumSquares`
- Measures of center: `mean, median, mode, proportion`
- Measures of spread: `range, iqr, variance, stdev`
- Measures of shape: `skewness, kurtosis`

Although these functions use different methods to produce different results, structurally they are all the same, following the pattern in equation 1. We can exploit this common structure by performing various operations on these measures such as confidence intervals or hypothesis tests.

Two additional types of measures have slightly different structures, but can be adapted to some of those same operations:

- Measures of position: `percentile, quartile, zScore, max, min` Some of these require a position parameter.
- Measures of association: `cov, corr`. These require two sets of paired data.

When a summary function is applied to a frequency distribution instead of a vector, its result can be approximated. When applied to statistical data, the procedure is trivial.

2.2.1 Examples of summary function applied to various data structures

A summary function can be applied to any one of the three data representations. Except in the case of raw data where the variable is represented as a vector, there may be some loss of information, so the results may differ slightly:

```
    mean Height          ⍝ Sample mean height; raw data
68.6
    mean FreqHeight      ⍝ Frequency; slight loss of information
69.15
    Mean NameSpace       ⍝ No calculation; value obtained directly
68.6
```

2.2 Distribution Functions

Distribution functions require two inputs: a value, and a set of one or more parameters. For discrete distributions, the result is defined as the probability mass function for that value. For continuous distributions, the result is the density, or the relative likelihood of that value. In its native form, a continuous distribution function is not particularly useful.

2.2.1 Examples

What is the probability that if you toss 6 fair coins, you will get exactly 2 heads? This requires the binomial distribution function with parameters $n = 6$, $p = 0.5$ and input $x = 2$.

```
        6 0.5 binomial 2
0.3125
```

What is the relative likelihood of the standard normal distribution at various values of Z? Note the standard normal distribution does not require a left argument; we assume the default parameters of $\mu = 0$ and $\sigma = 1$.

```
      normal ¯2 ¯1 0 1 2
0.053991 0.24197 0.39894 0.24197 0.05399
```

2.3 Relational and Logical Functions

Relational functions take numeric inputs and produce Boolean outputs. If the relation holds, the result is 1; otherwise, it is 0.

Examples:

```
        2 < 3                ⍝ This is a true statement
1
        5 ≥ 6                ⍝ This statement is false
0
        3 between 2 4
1
```

Logical functions combine truth values:

```
       (2 < 3) and (5 ≥ 6)   ⍝ True and False  ==>  False
0
       (2 < 3) or (5 ≥ 6)    ⍝ True or False   ==>  True
1
       not 3 between 2 4     ⍝  Negation of true statement
0
```

3. Operators

Operators are higher order functions which take other functions as inputs as defined in [Iverson 1979]. The syntax of operators that modify functions is similar to that of the derivative $f'(x)$ or

the inverse $f^{-1}(x)$. Note that the operator comes between the function and its argument. The monadic operator syntax is (left argument is optional):

```
Result ← [LeftArgument] function operator RightArgument
```

A summary function, such as **mean** or **proportion,** produces a point estimate; when it is combined with the **confInt** operator, the result is a 2-item vector containing 95% confidence limits. Note that the syntax is similar to the derivative notation in calculus:

```
      mean confInt Height              ⍝ 95% confidence interval
64.741 72.973
```

To choose a different confidence level, specify the percentage as the left argument:

```
      proportion Height > 69           ⍝ Point Estimate
0.57143
      0.9 proportion confInt Height > 69   ⍝ 90% conf interval
0.26377 0.87909
```

The syntax of operators that combine functions to produce new functions is similar to function composition: $f \circ g$. In this case, the operator is placed between two functions. The dyadic operator syntax is (left argument optional):

```
Result ← [LeftArg] function1 operator function2 RightArg
```

The dyadic operator **probability** combines a distribution function on the left with a relational function on the right to calculate the theoretical probability of an event given a particular discrete or continuous distribution.

The probability operator allows us to calculate cumulative, upper-tail and probabilities between values. Note how the syntax is similar to function composition:

```
       normal probability ≤ 1.96    ⍝ Cumulative probability
0.975
       normal probability > 1.645   ⍝ Upper-tail probability
0.05
       normal probability between 0 1.28  ⍝ Pr(0≤Z≤1.28)
0.4
```

3.1 How operator syntax makes expressions more readable

Consider the following problem: What is the probability of getting at least four heads in five coin tosses, assuming a fair coin?

Three adjustments to the problem are required. First, the discrete distribution must be made cumulative. Then a lower-tail probability must be converted to upper-tail. Finally, the input of four must be adjusted downward by one to get the correct probability. Below are examples of expressions in Excel and R:

```
Excel:      =1-BINOM.DIST(4-1,5,0.5,TRUE)
R:          pbinom(4-1,5,0.5,lower.tail=FALSE)
```

Using operator syntax, we can represent the same thing with a succinct and English-like expression without having to make any adjustments to the original problem:

```
          5 0.5 binomial probability ≥ 4
0.1875
```

4. Performing Statistics by Combining Functions

We can perform many types of statistical operations by combining these four types of functions using operators. Some other ways include:

Statistical Process	Operator	Left Operand	Right Operand
Hypothesis Test	`hypothesis`	Summary Function	Relation
Confidence interval	`confInt`	Summary Function	N/A
Probability Distribution	`prob`	Distribution function	Relation
Laws of Probability	`prob`	Logical Function	Contingency Table
Critical Value	`critVal`	Distribution Function	Relation
Simulation	`randVar`	Distribution Function	N/A
Expectation	`theoretical`	Distribution Function	Summary Function
Goodness of Fit	`goodnessOfFit`	Distribution Function	Relation

Here are some examples using functions derived from operators:

- Is the average height of college students less than 72 inches?

```
   Model ← Height mean hypothesis < 72
   Model.P
0.055473
```

- What is the variance of the binomial distribution where $n = 7$ and $p = 0.3$?

```
    7 0.3 theoretical variance 0
1.47
```

- Generate random data from the normal distribution where $\mu = 68, \sigma = 3, n = 7$

```
   68 3 normal randomVariable 7
73.39 70.553 68.567 63.941 64.672 71.323 69.171
```

5. Conclusion

Functions are an integral part of statistics. Functions have various types of inputs and outputs which make them very suitable for implementation on a computer. Much of statistical analysis is computer-based due to the complexity of mathematical calculations. Students who have a fundamental understanding of functions and data structures will have a competitive advantage when learning statistics. Operators which in effect are functions of functions can be used to reduce the statistical vocabulary required of students. The additional complexity of operators is offset by the consistency of syntax and the resulting English-like expressions.

A computer program called TamStat (short for Taming Statistics) written in Dyalog APL, uses the exact syntax described in this paper and is available as a teaching tool. For further information about TamStat, go to tamstat.dyalog.com.

References

Falkoff,A.D . and Iverson, K.E. "The Design of APL" IBM Journal of Research and Development (Volume: 17, Issue: July 1973)

Iverson, K. 1979. The Role of Operators in APL. APL Quote Quad, 9(4),128-133.

Mansour, S., Taming Statistics with Limited Domain Operators, Vector, Article In press, British APL Association https://archive.vector.org.uk/index.

Mansour, S., Turning the Tables on Probability Distributions, Paper Presented at JSM 2017

Scholes, John. 1994. Dyalog APL Namespaces. Vector, 11(1), 101-107.