

4. A Customized Content MathML for Probability and Statistics

1. [Introduction](#)
2. [Probability and Expected Value](#)
 - [Probability](#)
 - [Expected Value](#)
 - [Conditioning](#)
3. [Functions](#)
 - [Trigonometric and Related Functions](#)
 - [Cardinality](#)
 - [Determinant](#)
 - [New Special Functions](#)
 - [Chi-Square Quantiles](#)
 - [Indicator Functions](#)
 - [Positive and Negative Parts](#)
 - [Norm and Inner Product](#)
 - [Factorial](#)
 - [Binomial and Multinomial Coefficients](#)
4. [Relations](#)
 - [General Relations](#)
 - [Special Relations](#)
 - [Equivalence Class](#)
 - [Mapping](#)
5. [Operations](#)
 - [Convolution](#)
 - [Convolution Power](#)
 - [Composition Power](#)
 - [Plus-Minus](#)
 - [Power Set](#)
 - [Complement](#)
 - [Symmetric Difference](#)
 - [Union and Intersection](#)
6. [Infimum and Related Constructions](#)
 - [Infimum and Supremum](#)
 - [Limit Inferior and Limit Superior](#)
7. [Special Sets of Numbers](#)
8. [Hypothesis Testing](#)
9. [Playing Cards](#)
 - [Special Denominations](#)
 - [Suits](#)
10. [Miscellaneous Additions](#)

- [Et Cetera](#)
- [Odds](#)
- [Ordinals](#)

4. 1. Introduction

This section describes the customized and extended version of Content MathML developed for the project [Virtual Laboratories in Probability and Statistics](#). Thus, the information here should be considered a companion to the standard [Content MathML documentation](#) at [W3C](#). The customized Content MathML markup is translated into standard Presentation MathML via the XSL transformation file [stat.xsl](#) which in turn is based on the standard XSL transformation file [ctop.xsl](#) written by David Carlisle. Our stylesheet [stat.xsl](#) also corrects a number of bugs in the original [ctop.xsl](#). These corrections are due to Tom Leathrum.

4.2. Probability and Expected Value

Probability

The markup `<prob />` has been added to denote probability measure, and is rendered by the stylesheet with the special symbol \mathbb{P} . This markup is ordinarily used with the standard `<apply></apply>` construction, with one or more arguments. For example, the markup

```
<apply><prob />
  <apply><intersect />
    <ci>A</ci>
    <ci>B</ci>
  </apply>
</apply>
```

is rendered by the stylesheet as $\mathbb{P}(A \cap B)$.

Expected Value

Similarly, the markup `<expect />` has been added to denote the expected value operator, and is rendered by the stylesheet with the special symbol \mathbb{E} . This markup is ordinarily used with the standard `<apply></apply>` construction, with one or more arguments. For example, the markup

```
<apply><expect />
  <ci>X</ci>
  <apply><geq />
    <ci>Y</ci>
    <cn>0</cn>
  </apply>
</apply>
```

is rendered by the stylesheet as $\mathbb{E}(X, Y \geq 0)$

Conditioning

The notion of conditioning is of central importance in modern probability, so the markup `<given></given>` has been added for conditional probability and conditional expected value. This markup is used inside the `<apply></apply>` construction, and can have any number of arguments. For example, the markup

```
<apply><prob />
  <ci>A</ci>
  <given>
    <ci>B</ci>
    <ci>C</ci>
  </given>
</apply>
```

is rendered by the stylesheet as $\mathbb{P}(A|B, C)$. Similarly, the markup

```
<apply><expect />
  <ci>X</ci>
  <given>
    <apply><geq />
      <ci>Y</ci>
      <cn>0</cn>
    </apply>
  </given>
</apply>
```

is rendered by the stylesheet as $\mathbb{E}(X|Y \geq 0)$

4.3. Functions

Trigonometric and Related Functions

Our stylesheet uses the standard `<apply></apply>` construction with all of the trigonometric, logarithmic, hyperbolic functions and the inverses of these functions, that are in standard Content MathML. Thus, the stylesheet renders these functions in consistent notation, with arguments enclosed in parentheses after the function name. For example,

```
<apply><sin />
  <ci>&theta;</ci>
</apply>
```

is rendered by the stylesheet as $\sin(\theta)$ rather than $\sin \theta$.

Cardinality

We use function notation for cardinality. The markup `<card />` in standard Content MathML is rendered with our

stylesheet by #. This markup works in the usual way with the standard `<apply></apply>` construction. For example,

```
<apply><card />
  <ci>A</ci>
</apply>
```

is rendered by the stylesheet as $\#(A)$ rather than $|A|$.

Determinant

Similarly, we use function notation for determinants. The markup `<determinant />` in standard Content MathML is rendered by our stylesheet as det. This markup works in the usual way with the standard `<apply></apply>` construction. For example

```
<apply><determinant />
  <ci>A</ci>
</apply>
```

is rendered by the stylesheet as $\det(A)$ rather than $|A|$.

New Special Functions

Markup has been added for a number of special functions, particularly those related to probability and statistics.

Variance

The markup `<variance />` denotes variance and is rendered by the stylesheet as var.

Covariance

The markup `<covariance />` denotes covariance and is rendered by the stylesheet as cov.

Correlation

The markup `<correlation />` denotes correlation and is rendered by the stylesheet as cor.

Skewness

The markup `<skew />` denotes skewness and is rendered by the stylesheet as skew.

Kurtosis

The markup `<kurt />` denotes correlation and is rendered by the stylesheet as kurt.

Beta

The markup `<Beta />` denotes the beta function and is rendered by the stylesheet as B .

Gamma

The markup `<Gamma />` denotes the gamma function and is rendered by the stylesheet as Γ .

Zeta

The markup `<zeta />` denotes the zeta function and is rendered by the stylesheet as ζ .

Lebesgue measure

The markup `<lebesgue />` denotes Lebesgue measure and is rendered by the stylesheet as λ .

Closure

The markup `<closure />` denotes closure and is rendered by the stylesheet as cl.

Each of these works in the usual with the standard `<apply></apply>` construction. With this construction, `<covariance />` and `<correlation />` ordinarily have two arguments; the other functions in the list ordinarily have one argument. For example,

```
<apply><covariance />
  <ci>X</ci>
  <ci>Y</ci>
</apply>
```

is rendered by the stylesheet as $\text{cov}(X, Y)$ while

```
<apply><skew />
  <ci>X</ci>
</apply>
```

is rendered by the stylesheet as $\text{skew}(X)$.

Chi-Square Quantiles

Chi-square procedures are common in statistics, and involve some complicated notation. The markup `<chisquare />` has been added for the chi-square quantile function with a specified degrees of freedom. This markup should be used in the standard `<apply></apply>` construction, with two arguments. For example, the markup

```
<apply><chisquare />
  <ci>k</ci>
  <ci>p</ci>
</apply>
```

is rendered by the stylesheet as $\chi_k^2(p)$.

Indicator Functions

The markup `<indicate />` has been added for an indicator function, and is rendered by the stylesheet as **1**. This markup is used with the `<apply></apply>` construction. For example, the markup

```
<apply>
  <apply><indicate />
    <ci>A</ci>
  </apply>
  <ci>x</ci>
</apply>
```

is rendered by the stylesheet as $\mathbf{1}(A)(x)$.

Positive and Negative Parts

The markup `<positivepart />` and `<negativepart />` have been added for the positive and negative parts of a real number, respectively. These should be used with the standard `<apply></apply>` construction, with one argument. For example, the markup

```
<apply><positivepart />
  <ci>x</ci>
</apply>
```

is rendered by the stylesheet as x^+ , while the markup

```
<apply><negativepart />
  <ci>x</ci>
</apply>
```

is rendered by the stylesheet as x^- .

Norm and Inner Product

The markup `<norm />` has been added for the norm of a vector. The markup should be used with the `<apply></apply>` construction, with one argument. For example, the markup

```
<apply><norm />
  <ci>x</ci>
</apply>
```

is rendered by the stylesheet as $\|x\|$.

The markup `<innerproduct />` has been added for the inner product of two vectors. The markup should be used with the `<apply></apply>` construction, with two arguments. For example, the markup

```
<apply><innerproduct />
  <ci>x</ci>
  <ci>y</ci>
</apply>
```

is rendered by the stylesheet as $\langle x, y \rangle$. Standard Content MathML has the markup `<scalarproduct />` for scalar product (or dot product). Our stylesheet leaves the rendering of this element unchanged.

Factorial

The markup `<factorial />` for factorials has been extended to denote the “falling power” formula used for counting permutations and an even more general permutation formula. (For the definitions of these, see [Combinatorial Structures in Virtual Laboratories in Probability and Statistics](#)) This markup should be used in the standard `<apply></apply>` construction, with one, two, or three arguments. With one argument, the markup denotes the standard factorial, so that

```
<apply><factorial />
  <ci>n</ci>
```

</apply>

is rendered by the stylesheet as $n!$. With two arguments, the markup denotes the falling power factorial formula, so that

```
<apply><factorial />
  <ci>n</ci>
  <ci>k</ci>
</apply>
```

is rendered by the stylesheet as $n^{(k)}$. With three arguments, the markup denotes the general factorial formula, so that

```
<apply><factorial />
  <ci>n</ci>
  <ci>k</ci>
  <ci>c</ci>
</apply>
```

is rendered by the stylesheet as $n^{(k,c)}$

Binomial and Multinomial Coefficients

The markup <binomial /> has been added for binomial and multinomial coefficients. This markup should be used with the standard <apply></apply> construction; with two arguments it represents a binomial coefficient, and with more than two arguments it represents a multinomial coefficient. For example,

```
<apply><binomial />
  <ci>n</ci>
  <ci>k</ci>
</apply>
```

is rendered by the stylesheet as $\binom{n}{k}$, and

```
<apply><binomial />
  <ci>n</ci>
  <ci>i</ci>
  <ci>j</ci>
  <ci>k</ci>
</apply>
```

is rendered by the stylesheet as $\binom{n}{i,j,k}$.

4.4. Relations

General Relations

The standard <apply></apply> construction has been extended with the addition of a `type` attribute. The

`type="relation"` setting uses standard notation associated with a relation. For example,

```
<apply type="relation">
  <ci>R</ci>
  <ci>x</ci>
  <ci>y</ci>
</apply>
```

is rendered by the stylesheet as xRy , as opposed to

```
<apply>
  <ci>R</ci>
  <ci>x</ci>
  <ci>y</ci>
</apply>
```

which is rendered by the stylesheet in the usual functional way as $R(x, y)$.

Special Relations

Markup has been added for a number of special relation symbols, beyond the ones in standard Content MathML. For the most part, these relations are associated with partial orders, equivalence relations, and other transitive relations.

Weak precedes

`<preceq />` which is rendered by the stylesheet as \preceq .

Weak succeeds

`<succeq />` which is rendered by the stylesheet as \succeq .

Strong precedes

`<prec />` which is rendered by the stylesheet as $<$.

Strong succeeds

`<succ />` which is rendered by the stylesheet as $>$.

Parallel

`<parallel />` which is rendered by the stylesheet as \parallel .

Perpendicular

`<perp />` which is rendered by the stylesheet as \perp .

Weak approximation

`<approxpeq />` which is rendered by the stylesheet as \cong .

Congruent

`<congruent />` which is rendered by the stylesheet as \equiv .

Left arrow

`<left />` which is rendered by the stylesheet as \leftarrow .

Right arrow

`<right />` which is rendered by the stylesheet as \rightarrow .

Left-right arrow

`<leftright />` which is rendered by the stylesheet as \leftrightarrow .

Up arrow

`<up />` which is rendered by the stylesheet as \uparrow .

Down arrow

`<down />` which is rendered by the stylesheet as \downarrow .

Up-down arrow

`<updown />` which is rendered by the stylesheet as \updownarrow .

Of course, each of these should be used with the `<apply type="relation"></apply>` construction, with two arguments. For example,

```
<apply type="relation"><preeq />
  <ci>x</ci>
  <ci>y</ci>
</apply>
```

is rendered by the stylesheet as $x \preceq y$.

Equivalence Class

The markup `<equivalenceclass />` has been added to denote an equivalence class. This markup should be used with the standard `<apply></apply>` construction, with one argument. For example, the markup

```
<apply><equivalenceclass />
  <ci>x</ci>
</apply>
```

is rendered by the stylesheet as $[x]$.

Mapping

The `<mapsto />` markup has been added for the description of functions. This markup should be used with the standard `<apply></apply>` construction, with either two or three arguments. For example, the markup

```
<apply><mapsto />
  <ci>x</ci>
  <apply><power />
    <ci>x</ci>
    <cn>2</cn>
  </apply>
</apply>
```

is rendered by the stylesheet as $x \mapsto x^2$, while the markup

```
<apply><mapsto />
  <ci>f</ci>
  <ci>A</ci>
  <ci>B</ci>
</apply>
```

is rendered by the stylesheet as $f : A \rightarrow B$,

4.5. Operations

We have added several new operations, beyond the ones in standard Content MathML.

Convolution

The markup `<convolve />` denotes convolution, and is rendered by the stylesheet as $*$. This markup is ordinarily used with the `<apply></apply>` construction, with two or more arguments. For example, the markup

```
<apply><convolve />
  <ci>f</ci>
  <ci>g</ci>
  <ci>g</ci>
</apply>
```

is rendered by the stylesheet as $f * g * h$.

Convolution Power

The markup `<convolutionpower />` denotes convolution power (repeated convolution of a function with itself). It should be used with the `<apply></apply>` construction, with two arguments. For example, the markup

```
<apply><convolutionpower />
  <ci>f</ci>
  <ci>n</ci>
</apply>
```

is rendered by the stylesheet as f^{*n} .

Composition Power

The composition operation `<compose />` is present in standard Content MathML. We have added `<compositionpower />` to denote repeated composition of a function with itself. This markup should be used with the standard `<apply></apply>` construction, with two arguments. For example, the markup

```
<apply><compositionpower />
  <ci>f</ci>
  <ci>n</ci>
</apply>
```

is rendered by the stylesheet as $f^{\circ n}$.

Plus-Minus

The markup `<plusminus />` has been added for the plus-minus operator, which is rendered by the stylesheet as \pm . The plus-minus operator is both unary and binary, so the markup should be used with the `<apply></apply>` construction, with either one or two arguments. For example, the markup

```
<apply><plusminus />
  <ci>a</ci>
</apply>
```

is rendered by the stylesheet as $\pm a$, while the markup

```
<apply><plusminus />
  <ci>a</ci>
  <ci>b</ci>
</apply>
```

is rendered by the stylesheet as $a \pm b$.

Power Set

The markup `<powerset />` has been added for the power set operator, which is rendered by the stylesheet as \mathcal{P} . This markup is ordinarily used with the `<apply></apply>` construction, with one argument. For example, the markup

```
<apply><powerset />
  <ci>A</ci>
</apply>
```

is rendered by the stylesheet as $\mathcal{P}(A)$.

Complement

The markup `<complement />` has been added for the complement of a set. This markup is ordinarily used with the `<apply></apply>` construction, with one argument. For example, the markup

```
<apply><complement />
  <ci>A</ci>
</apply>
```

is rendered by the stylesheet as A^c .

Symmetric Difference

The markup `<symmdiff />` has been added for the symmetric difference operator, rendered by the stylesheet as Δ . This markup is ordinarily used with the `<apply></apply>` construction, with two arguments. For example, the markup

```
<apply><symmdiff />
  <ci>A</ci>
  <ci>B</ci>
</apply>
```

is rendered by the stylesheet as $A\Delta B$.

Union and Intersection

New markup expressions `<Union />` and `<Intersect />` have been added for unions and intersections, respectively, over collections. The distinction between the standard markup `<union />` and the new markup `<Union />` is analogous to the distinction between `<plus />` and `<sum />` in standard markup. For example, the markup

```
<apply><Union />
  <condition>
    <apply><in />
      <ci>i</ci>
      <ci>I</ci>
    </apply>
  </condition>
  <apply><selector />
    <ci>A</ci>
```

```

    <ci>i</ci>
  </apply>
</apply>

```

is rendered by the stylesheet as $\bigcup_{i \in I} A_i$, while the markup

```

<apply><Union />
  <bvar><ci>i</ci></bvar>
  <lowlimit><cn>1</cn></lowlimit>
  <uplimit><ci>n</ci></uplimit>
  <apply><selector />
    <ci>A</ci>
    <ci>i</ci>
  </apply>
</apply>

```

is rendered by the stylesheet as $\bigcup_{i=1}^n A_i$. The markup for intersection works in exactly the same way, with

`<Intersect />` replacing `<Union />`. Unions and intersections over collections will be a part of the new Content MathML 3.0 specification.

4.6. Infimum and Related Markup

Infimum and Supremum

Standard Content MathML has markup for minimums and maximums. We have added analogous markup for infima and suprema. The markup `<inf />` and `<sup />` are used with the standard `<apply></apply>` construction. Thus for example, the markup

```

<apply><inf />
  <ci>a</ci>
  <ci>b</ci>
  <ci>c</ci>
</apply>

```

is rendered by the stylesheet as $\inf\{a, b, c\}$, and the markup

```

<apply><inf />
  <bvar>
    <apply><selector />
      <ci>a</ci>
      <ci>n</ci>
    </apply>
  </bvar>
  <condition>
    <apply><in />
      <ci>n</ci>
      <naturalnumbers />

```

```

    </apply>
  </condition>
</apply>

```

is rendered by the stylesheet as $\inf\{a_n : n \in \mathbb{N}\}$. Of course, the markup for suprema is entirely analogous, with `<sup />` replacing `<inf />`.

Limit Inferior and Limit Superior

Standard Content MathML has markup for limits. We have added analogous markup for limit inferior and limit superior. The markup `<limitinf />` and `<limitsup />` are used with the standard `<apply></apply>` construction. Thus, for example, the markup

```

<apply><limitsup />
  <condition>
    <apply><tendsto />
      <ci>t</ci>
      <ci>x</ci>
    </apply>
  </condition>
<apply>
  <ci>F</ci>
  <ci>t</ci>
</apply>
</apply>

```

is rendered by the stylesheet as $\limsup_{t \rightarrow x} F(t)$. Of course, the markup for limit inferior is entirely analogous, with `<limitinf />` replacing `<limitsup />`.

4.7. Special Sets of Numbers

Standard Content MathML has markup for several special sets of numbers. We extend this to include

Positive integers

The markup `<positiveintegers />` denotes the set of positive integers, rendered by the stylesheet as \mathbb{N}_+ . In our project, the natural numbers (which has markup in standard Content MathML) refers to the nonnegative integers.

Positive rationals

The markup `<positiverationals />` denotes the set of positive rational numbers, rendered by the stylesheet as \mathbb{Q}_+ .

Algebraic numbers

The markup `<algebraicnumbers />` denotes the set of algebraic numbers, rendered by the stylesheet as \mathbb{A} .

4.8. Hypothesis Testing

Hypothesis testing is a core procedure in inferential statistics. Thus, we have added special markup for hypothesis testing, with a type attribute to distinguish between the null and alternative hypotheses. Thus,

`<hypothesis type="null" />` is rendered by the stylesheet as H_0 , while `<hypothesis type="alt" />` is rendered by the stylesheet as H_1 . This markup is ordinarily used with the `<apply></apply>` construction, with one argument (representing the hypothesis). For example, the markup

```
<apply><hypothesis type="null" />
  <apply><eq />
    <ci>&mu;</ci>
    <cn>0</cn>
  </apply>
</apply>
```

is rendered by the stylesheet as $H_0: \mu = 0$.

4.9. Playing Cards

Concepts and techniques in probability are often illustrated with examples involving playing cards. Such examples are useful because of their conceptual clarity and because of the rich variety of problems that can be posed.

Special Denominations

The markup `<ace />`, `<jack />`, `<queen />`, and `<king />` have been added to denote the denominations of ace, jack, queen, and king, respectively. The stylesheet renders these as 1, j, q, and k, respectively. These should be considered as special constants. Indeed, the other denominations are ordinarily denoted by the standard numbers 2, 3, 4, 5, 6, 7, 8, 9, and 10.

Suits

The markup `<spades />`, `<hearts />`, `<diamonds />`, and `<clubs />` have been added to denote the suits. These are rendered by the stylesheet as ♠, ♥, ♦, and ♣, respectively. The markup for a suit can also be used with the `<apply></apply>` construction, with one arguments, to represent a card in that suit. For example, the markup

```
<apply> <hearts />
  <queen />
</apply>
```

would be rendered by the stylesheet as q♥

4.10. Miscellaneous Additions

Et Cetera

The markup `<etc />` has been added to denote the ubiquitous concept of “and so forth”. The stylesheet renders this with three dots (an ellipsis), in either a lower position or a middle position, depending on a type attribute. For example,

the markup

```
<vector>
  <apply><selector />
    <ci>x</ci>
    <cn>1</cn>
  </apply>
  <apply><selector />
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <etc />
  <apply><selector />
    <ci>x</ci>
    <ci>n</ci>
  </apply>
</vector>
```

is rendered by the stylesheet as (x_1, x_2, \dots, x_n) , while the markup

```
<apply><plus />
  <apply><selector />
    <ci>x</ci>
    <cn>1</cn>
  </apply>
  <apply><selector />
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <etc type="middle" />
  <apply><selector />
    <ci>x</ci>
    <ci>n</ci>
  </apply>
</apply>
```

is rendered by the stylesheet as $x_1 + x_2 + \dots + x_n$.

Odds

The markup `<odds />` has been added to denote odds in gambling and probability. This markup should be used with the standard `<apply></apply>` construction, with two arguments. For example, the markup

```
<apply><odds />
  <ci>a</ci>
  <ci>b</ci>
</apply>
```

is rendered by the stylesheet as $a : b$.

Ordinals

The markup `<ordinal />` has been added for ordinal numbers. This markup should be used with the standard `<apply></apply>` construction, with one argument. The type of suffix is controlled by a type attribute, with *th* as the default. For example

```
<apply><ordinal />
  <ci>n</ci>
</apply>
```

is rendered by the stylesheet as n^{th} ;

```
<apply><ordinal type="st"/>
  <cn>1</cn>
</apply>
```

is rendered by the stylesheet as 1^{st} ;

```
<apply><ordinal type="nd"/>
  <cn>2</cn>
</apply>
```

is rendered by the stylesheet as 2^{nd} ; and

```
<apply><ordinal type="rd"/>
  <cn>3</cn>
</apply>
```

is rendered by the stylesheet as 3^{rd} .