

The `binomexp` package*

David Roderick
angel_ov_north at tiscali dot co dot uk

January 10, 2007

Abstract

Calculates and prints successive lines of Pascal's triangle..

$$\begin{array}{l} (f+s)^4 \left| \begin{array}{l} 1f^4s^0 \\ 4f^3s^1 \\ 6f^2s^2 \\ 4f^1s^3 \\ 1f^0s^4 \end{array} \right| \\ (f+s)^5 \left| \begin{array}{l} 1f^5s^0 \\ 5f^4s^1 \\ 10f^3s^2 \\ 10f^2s^3 \\ 5f^1s^4 \\ 1f^0s^5 \end{array} \right| \end{array}$$

and also will typeset the following proof

$$7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \qquad \binom{n}{r} = \frac{n!}{(n-r)! \cdot r!} = \frac{n!}{(n-r)! \cdot (n-(n-r))!} = \binom{n}{n-r}$$

$$\begin{aligned} \binom{n-1}{r-1} + \binom{n-1}{r} &= \frac{(n-1)!}{(r-1)! \cdot [(n-1)-(r-1)]!} + \frac{(n-1)!}{r! \cdot [(n-1)-r]!} \\ &= (n-1)! \cdot \left(\frac{1}{(r-1)! \cdot (n-r)!} + \frac{1}{r! \cdot [(n-r)-1]!} \right) \\ &= (n-1)! \cdot \frac{r+(n-r)}{r!(n-r)!} \\ &= \frac{n!}{r!(n-r)!} = \binom{n}{r} \end{aligned}$$

$$\frac{r}{r! \cdot (n-r)!} = \frac{1}{(r-1)! \cdot (n-r)!} \qquad \text{because} \qquad \frac{6}{6! \cdot (n-r)!} = \frac{1}{5! \cdot (n-r)!}$$

$$\begin{aligned} (r+1) \cdot \binom{n+1}{r+1} &= (r+1) \cdot \frac{(n+1)!}{((r+1)! \cdot ((n+1)-(r+1))!)} \\ &= (r+1) \cdot \frac{(n+1)!}{(r+1)! \cdot (n-r)!} \\ &= (n+1) \cdot \frac{n!}{r! \cdot (n-r)!} = (n+1) \cdot \binom{n}{r} \end{aligned}$$

1 Introduction

A very simple package with simple usage. Putting ‘binomexp’ (which is also typed exactly the same way than `\binomexp`) inside of the argumentative input of the `\usepackage` commands enables the user to do two extra things.

*This document corresponds to `binomexp` v1.0, dated 2007/01/07.

- print any successive rows of Pascal's triangle which will fit on the page up until the power as 31, at which point L^AT_EX runs out of brain power.
- Use a piece of code which Morten Høgholm wrote which allows the cells inside of an array or a tabular to be repeated in a similar way than those may be repeated inside of the initial description of said array or tabular.

2 Usage

Binomexp ought to load ifthen and calc by itself. If you have already loaded these packages using `\usepackage{calc,ifthen}` unload these therefore. You must then use the command as `\makeatletter` so to get the command names with the symbol as @ inside of those to function.

`\binomexp@putpascal` `\binomexp@putpascal {<number as lower power>} {<number as higher power>} {<symbol as first variable>} {<symbol as second variable>} {<symbol again as first variable>} {<symbol again as second variable>}`

`\binomexp@putpascal{7}{9}{f}{x}{f}{x}` will typeset the rows as 7, 8, and 9 of Pascal's triangle. The first column will have $(f + x)^{power}$. The reason why you have to input the symbol again is because the user might like to use a `\cdot` or whatever in the other columns except the first column. And that's it really.

`\binomexp@proof` `\binomexp@proof {<number as row variable>} {<number as column variable>}` will typeset the mathematical proof of Pascal's triangle, which is based upon the observation that the co-efficient is equal with the number of possible combinations of the column variable out of the row variable.

3 How I wrote it.

```
1 \RequirePackage{calc,ifthen}
Morten Høgholm wrote the following code.
2 \newcommand\binomexp@replicate[2]{%
3   \ifnum#1>\z@ \expandafter\@firstofone
4   \else
5     \expandafter\@gobble
6   \fi
7   {#2\expandafter\binomexp@replicate\expandafter{\number\numexpr#1-1\relax}{#2}}%
8 }
```

Morten's code allows the following.

```
\begin{document}
\makeatletter
\begin{tabular}{|*{6}{|c|}|}
something1 \binomexp@replicate{4}{& something2}Blah&stuff\\
something1 \binomexp@replicate{4}{& something2}Blah&stuff\\
Third row with line atop from second to fifth column:
\cline{2-5}something1 \binomexp@replicate{4}{& something2}Blah&stuff\\
\end{tabular}
\end{document}
```

You can invoke Morten's code either by loading the `\usepackage{binomexp}` within the preamble, and then by putting `\makeatletter`, or by including the following code somewhere (perhaps a preamble).

```

\makeatletter
\newcommand\binomexp@replicate[2]{%
  \ifnum#1>\z@ \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {#2\expandafter\binomexp@replicate\expandafter{\number\numexpr#1-1\relax}{#2}}%
}
\makeatother

```

`\binomexp@call` the `\newcommand` as `\binomexp@call` makes things nice and pretty within a cell
 9 `\newcommand{\binomexp@call}[1]{\rule[-0.125cm]{0mm}{0.5cm}\mbox{##1}}`

`\binomexp@up` the `\newcommand` as `\binomexp@up` is by the power of the series which ascends
 10 `\newcounter{binomexp@up}`
 11 `\newcommand{\binomexp@up}{\number\value{binomexp@up}}`
 12 `\addtocounter{binomexp@up}{1}`

`\binomexp@down` the `\newcommand` as `\binomexp@down` is by the power of the series which descends
 13 `\newcounter{binomexp@down}`
 14 `\newcommand{\binomexp@down}{\number\value{binomexp@down}}`
 15 `\addtocounter{binomexp@down}{-1}`

`\binomexp@columns` an array of so many columns
 16 `\newcounter{binomexp@columns}`

`\binomexp@power` $(f + s)^{power}$
 17 `\newcounter{binomexp@power}`

`\binomexp@pascalstart` the next 3 counters are used within the `\binomexp@putpascal` command
`\binomexp@pascalstop` 18 `\newcounter{binomexp@pascalstart}`
`\binomexp@emptytimes` 19 `\newcounter{binomexp@pascalstop}`
 20 `\newcounter{binomexp@emptytimes}`

`\binomexp@variable1` the following 3 counters are used within the process of calculation as `\binomexp@printpascal`
`\binomexp@variable2` 21 `\newcounter{binomexp@variable1}`
`\binomexp@answervar` 22 `\newcounter{binomexp@variable2}`
 23 `\newcounter{binomexp@answervar}`

`\binomexp@sub`
 24 `\newcounter{binomexp@sub}`

`\binomexp@printpascal` to calculate the coefficients of the Pascal's triangle
 25 `\protect\newcommand*\binomexp@printpascal{`
 26 `\addtocounter{binomexp@power}{1}`
 27 `\expandafter\edef\csname`
 28 `binomexp@morten\roman{binomexp@power}exporti\endcsname{1}`
 29 `\setcounter{binomexp@sub}{2}`

```

30 \setcounter{binomexp@variable1}{\numexpr\number\value{binomexp@power}+1\relax}
31 \whiledo{\number\numexpr\number\value{binomexp@power}+1\relax>
32 \value{binomexp@sub}}{
33 \setcounter{binomexp@variable1}{\numexpr\number\value{binomexp@sub}-1\relax}
34 \setcounter{binomexp@variable2}{\value{binomexp@sub}}
35 \setcounter{binomexp@answervar}{\number\numexpr\csname
36 binomexp@x\roman{binomexp@variable1}\endcsname\relax+\number\numexpr\csname
37 binomexp@x\roman{binomexp@variable2}\endcsname\relax}
38 \expandafter\edef\csname binomexp@y\roman{binomexp@sub}\endcsname
39 {\number\value{binomexp@answervar}}\relax
40 \addtocounter{binomexp@sub}{1}
41 }

```

TRANSFER PART set counter as binomexp@sub to 1

```
42 \setcounter{binomexp@sub}{2}
```

create a loop which shall get the binomexp@y values and put those into the appropriate binomexp@x values. Also export the y values by this same corresponding power into a length called binomexp@morten\roman{power}export\roman{binomexp@sub}

```

43 \whiledo{\numexpr\number\value{binomexp@power}+1\relax>\value{binomexp@sub}}{
44 \setcounter{binomexp@answervar}{\number\numexpr\csname
45 binomexp@y\roman{binomexp@sub}\endcsname\relax}
46 \expandafter\edef\csname binomexp@x\roman{binomexp@sub}\endcsname
47 {\number\value{binomexp@answervar}}

```

Here is how I exported the values to the table.

```

48 \expandafter\edef\csname
49 binomexp@morten\roman{binomexp@power}export\roman{binomexp@sub}\endcsname
50 {\number\value{binomexp@answervar}}

51 \addtocounter{binomexp@sub}{1}
52 }

53 \setcounter{binomexp@variable1}
54 {\numexpr\number\value{binomexp@power}+1\relax}
55 \expandafter\edef\csname
56 binomexp@x\roman{binomexp@variable1}\endcsname{1}
57 \expandafter\edef\csname
58 binomexp@morten\roman{binomexp@power}export\roman{binomexp@variable1}\endcsname{1}

```

To see what is happening add the following lines at this place.

```

power is \number\value{binomexp@power}\par
\setcounter{binomexp@variable2}{1}
\whiledo{\value{binomexp@variable2}<
\numexpr\number\value{binomexp@power}+2\relax}{
binomexp@morten\roman{binomexp@power}export\roman{binomexp@variable2} is
\csname binomexp@morten\roman{binomexp@power}export\roman{binomexp@%
variable2}\endcsname\relax\par\addtocounter{binomexp@variable2}{1}}

```

```
59 }
```

\binomexp@putpascal set binomexp@xi as 1

binomexp@xi never alters

```

60 \newcommand*\binomexp@putpascal [6] {\par
61 \expandafter\edef\csname binomexp@xi\endcsname{1}

```

```

set an eventuality for binomexp@xi by the power as zero
62 \expandafter\edef\csname binomexp@mortenexporti\endcsname{1}
we'll need to start power as zero by the way \binomexp@printpascal is transfig-
ured.
63 \setcounter{binomexp@power}{0}
wrap the chipolatas in stringy bacon.
64 \setcounter{binomexp@pascalstart}{#1}
65 \setcounter{binomexp@pascalstop}{#2+1}
now calculate all the co-efficients.
66 \setcounter{binomexp@emptytimes}{\value{binomexp@pascalstop}}
67 \whiledo{\value{binomexp@emptytimes}>1}{
68 \binomexp@printpascal \addtocounter{binomexp@emptytimes}{-1}
69 }
work out the number of columns
70 \setcounter{binomexp@columns}
71 {\numexpr\number\value{binomexp@pascalstop}+2\relax}
now the table
72 \begin{math} \begin{array}{@{}|c|*{\value{binomexp@columns}}{c}|@{}}
repeat the number of rows so many times
73 \whiledo{\value{binomexp@pascalstart}<
74 \numexpr\number\value{binomexp@pascalstop}-1\relax}{
prime the binomexp@up gun and cock.
75 \setcounter{binomexp@up}{0}
prime the binomexp@down gun and cock.
76 \setcounter{binomexp@down}{\value{binomexp@pascalstart}}
77 \binomexp@call{(#3+#4)^{\number\numexpr\number\value{binomexp@pascalstart}\relax}}
78 \binomexp@replicate{\numexpr\number\value{binomexp@pascalstart}+1\relax}
79 {\&\binomexp@call{\csname
80 binomexp@morten\romannumeral\numexpr\value{binomexp@pascalstart}\relax
81 export\romannumeral\numexpr\value{binomexp@up}+1\relax\endcsname
82 #5^{\binomexp@down} #6^{\binomexp@up}}\}
83 \addtocounter{binomexp@pascalstart}{1}
84 }
add one more row for luck
85 \setcounter{binomexp@up}{0}
86 \setcounter{binomexp@down}{\value{binomexp@pascalstart}}
87 \binomexp@call{(#3+#4)^{\number\numexpr\number\value{binomexp@pascalstart}\relax}}
88 \binomexp@replicate{\numexpr\number\value{binomexp@pascalstart}+1\relax}
89 {\&\binomexp@call{\csname
90 binomexp@morten\romannumeral\numexpr\value{binomexp@pascalstart}\relax
91 export\romannumeral\numexpr\value{binomexp@up}+1\relax\endcsname
92 #5^{\binomexp@down} #6^{\binomexp@up}}\}
93 \end{array} \end{math}
94 }

```

`\binomexp@proof` This command prints a mathematical proof of the Pascals's triangle based upon obervation.

```

95 \newcommand{\binomexp@proof}[2]{

```

```

96 \[ 7!=7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1 \hspace*{5em}
97 \{#1 \choose #2} = \frac{#1!}{(#1-#2)!\cdot #2!}=
98 \frac{#1!}{(#1-#2)!\cdot(#1-(#1-#2))!}={#1 \choose #1-#2}
99 \]
100 \begin{eqnarray*} \{#1 - 1 \choose #2 - 1} + \{#1 - 1 \choose #2}
101 &=& \frac{(#1 - 1)!}{(#2 - 1)!\cdot[(#1 - 1) - (#2 - 1)]!} +
102 \frac{(#1 - 1)!}{#2!\cdot[(#1 - 1) - #2]!}\backslash
103 &=& (#1 - 1)!\cdot\left(\frac{1}{(#2 - 1)!\cdot(#1 - #2)!} +
104 \frac{1}{#2!\cdot[(#1 - #2) - 1]!}\right) \backslash\backslash
105 &=& (#1 - 1)! \cdot\frac{#2 + (#1 - #2)}{#2! (#1 - #2)!} \backslash\backslash
106 &=& \frac{#1!}{#2!(#1 - #2)!} = \{#1 \choose #2}
107 \end{eqnarray*}
108 \[ \frac{#2}{#2!\cdot(#1-#2)!} = \frac{1}{(#2-1)!\cdot(#1-#2)!}
109 \hspace*{5em} \mbox{because} \hspace*{5em} \backslash\backslash
110 \frac{6}{6!\cdot(#1-#2)!} = \frac{1}{5!\cdot(#1-#2)!} \backslash\backslash
111 \begin{eqnarray*}
112 (#2 + 1)\cdot \{#1 + 1 \choose #2 + 1} &=& (#2 + 1)\cdot
113 \frac{(#1 + 1)!}{((#2 + 1)!\cdot((#1 + 1) - (#2 + 1))!)}\backslash\backslash
114 &=& (#2 + 1)\cdot \frac{(#1 + 1)!}{(#2 + 1)!\cdot(#1 - #2)!}\backslash\backslash
115 &=& (#1 + 1)\cdot \frac{#1!}{#2!\cdot(#1 - #2)!} = (#1 + 1)\cdot
116 \{#1 \choose #2}\backslash\backslash
117 \end{eqnarray*}
118 ]

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in **roman** refer to the code lines where the entry is used.

Symbols	<code>\binomexp@power</code> ... <u>17</u>	E
<code>\@firstofone</code> 3	<code>\binomexp@printpascal</code> <u>25</u> , 72	<code>\edef</code> 27, 38, 50, 52, 59, 61, 65, 66
<code>\@gobble</code> 5	<code>\binomexp@proof</code> .. <u>2</u> , <u>99</u>	<code>\else</code> 4
<code>\[</code> 100, 112	<code>\binomexp@putpascal</code> <u>1</u> , <u>64</u>	<code>\end</code> 42, 97, 111, 121
<code>\backslash</code> . 86, 106, 108, 109, 113, 117, 118, 120	<code>\binomexp@replicate</code> 2, 7, 82, 92	<code>\endcsname</code> 28, 36–38, 49, 50, 53, 60, 62, 65, 66, 85, 95
<code>\]</code> 103, 114	<code>\binomexp@sub</code> <u>24</u>	<code>\expandafter</code> 3, 5, 7, 27, 38, 50, 52, 59, 61, 65, 66
A	<code>\binomexp@up</code> . <u>10</u> , 86, 96	F
<code>\addtocounter</code> 12, 15, 26, 40, 55, 72, 87	<code>\binomexp@variable1</code> <u>21</u>	<code>\fi</code> 6
B	<code>\binomexp@variable2</code> <u>21</u>	<code>\frac</code> 101, 102, 105–110, 112, 114, 117–119
<code>\begin</code> .. 45, 76, 104, 115	C	H
<code>\binomexp@answer</code> var <u>21</u>	<code>\cdot</code> 100– 102, 105–109, 112, 114, 116–119	<code>\hspace</code> 100, 113
<code>\binomexp@call</code> <u>9</u> , 81, 83, 91, 93	<code>\choose</code> ... 101, 102, 104, 110, 116, 120	I
<code>\binomexp@columns</code> . <u>16</u>	<code>\csname</code> 27, 35, 36, 38, 48, 50, 52, 59, 61, 65, 66, 83, 93	<code>\ifnum</code> 3
<code>\binomexp@down</code> <u>13</u> , 86, 96		
<code>\binomexp@emptytimes</code> <u>18</u>		
<code>\binomexp@pascalstart</code> <u>18</u>		
<code>\binomexp@pascalstop</code> <u>18</u>		

	L		S
<code>\left</code>	107	78, 81, 82, 84, 85, 91, 92, 94, 95	<code>\setcounter</code>
	M	P	29, 30, 33–35, 46, 48, 57, 67–70, 74, 79, 80, 89, 90
<code>\mbox</code>	9, 113	<code>\par</code>	
	N	<code>\protect</code>	
<code>\newcommand</code>	2, 9, 11, 14, 25, 64, 99		V
<code>\newcounter</code>		R	<code>\value</code> 11, 14, 30–34, 39, 47, 51, 54, 58, 70, 71, 75– 78, 80–82, 84, 85, 90–92, 94, 95
.	10, 13, 16–24	<code>\relax</code>	
<code>\number</code>	7, 11, 14, 30, 31, 33, 35, 36, 39, 47, 48, 51, 54, 58, 75, 78, 81, 82, 91, 92	7, 30, 31, 33, 36, 37, 39, 47, 49, 58, 75, 78, 81, 82, 84, 85, 91, 92, 94, 95	
<code>\numexpr</code>	7, 30, 31, 33, 35, 36, 47, 48, 58, 75,	<code>\RequirePackage</code>	
		1	W
		<code>\right</code>	<code>\whiledo</code>
		108	31, 47, 71, 77
		<code>\roman</code>	Z
		28, 36–38, 49, 50, 53, 60, 62	<code>\z@</code>
		<code>\romannumeral</code>	3
		
		84, 85, 94, 95	
		<code>\rule</code>	
		9	