```
@cT datastream 1
$syntaxlevel 2

fine    500,500
rescale TRUE, TRUE, FALSE, TRUE
define  button:        nextbutton
        button: prevbutton
        button: slidebuttona, slidebuttonb

define  MAXPMF = 10
define  screen: scr1
define  group,screenstatus:
        i: nextscreen
        i: prevscreen
        i: showcalcs

define  TITLESCREEN = 1
define  EQUATIONSCREEN = 2
define  STATSCREEN = 3
define  SLIDERSCREEN = 4

define  group, convolutionvars:      $$ slider vars for the last scree@
n
        slider: slidenuma, slidenumb
        i: multipliera, multiplierb
        slider: slidemulta, slidemultb

define  group, statvars:
        slider: slidec
        f: scalerx, scalery
        i: highlightc

define  group, graphs:  $$ values for all the graphs
        i: xorigin(4), yorigin(4)
        i: xpos(4), xneg(4), ypos(4), yneg(4)
        i: xmax(4), ymax(4), xmin(4)
        f: barwidth(4)
        i: barcolor(4)
        f: stripsize(3)
        f: stripscale
        button: editA, editB

define  group, editors: $$ sliders to manually set the values of the PM@
Fs
        slider: editslider1(0:20)
        slider: editslider2(0:20)
        button: upbutton1(0:20)
        button: downbutton1(0:20)
        button: upbutton2(0:20)
        button: downbutton2(0:20)
        button: OKbutton

define  group, pmfs:
        i: numa
        i: numb
        i: numc
        f: pmfa(0:20)
        f: pmfb(0:20)
        f: pmfc(0:40)

        slider: Pslider
        slider: Nslider
        slider: Rslider
        f: curP
        i: curN, curR

define  group, slidervars:
        slider: graphslider
        i: slidewhich
        f: curvalue, lastvalue


$$$$$$$$$$$$$$$$$$$$$$
$$ Initialize all variables, here!
$$$$$$$$$$$$$$$$$$$$$$@1ff   1
unit    main
        @1ff8000merge, pmfs:
        merge, slidervars:
        merge, screenstatus:
        merge, convolutionvars:

calc    multipliera:= 1
calc    multiplierb:= 1
        @1ff   1
@1ff8000do      init_pmf(1,4)
do      init_pmf(2,3)

do      generate_random(2)

calc    curP:= 0
calc    curN:= 0
calc    curR:= 0
calc    slidewhich:= 2
calc    curvalue:= -1
calc    lastvalue:= -1
calc    showcalcs:=  FALSE
do      init_graphs

jump    title_page

$$$$$$$$$$$$$$$$$$$$$$$$
$$ Title page, this shows up, first.
$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    title_page@1ff8000

fontp   zserif, 128
$$text  200,200;  300,300;
at      200, 200
write   @1fs  28@1ff   1Convolution!@1fs8000@1ff8000
@1pl4002$$@1fs  28Convolution!@1fs8000
@1pl4006$$\
do      set_screen(1)   $$ set screenvars to show that you’re on titlescreen
font    zserif, 24
do      menubar
do      next_button
do      prev_button

$$$$$$$$$$$$$$$$$$$$$$$$
$$ Calculates the size of each of the strips, used for the slider page
$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    size_of_strip@1ff8000
@1pl4007        merge, pmfs:
        merge, graphs:

calc    stripscale:= (xpos(4)-xneg(4)) / (xmax(4)-xmin(4))

calc    stripsize(1):= stripscale * numa
```

```
calc    stripsize(2):= stripscale * numb
calc    stripsize(3):= stripscale * numc

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Convolves multa*pmfa and multb*pmfb, and stores the result in pmfc
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    convolve(multa, multb)@1ff8000
@1pl4007        merge, pmfs:
        merge, global:
        i: index
        i: counter
        i: multa, multb

calc    numc:= numa+numb-1
loop    index:= 0,numc-1
        calc    pmfc(index):= 0
        loop    counter:= 0,index
                if      counter<numa & index-counter < numb
                        calc    pmfc(index):= pmfc(index) + pmfa(counter) * pmfb(index-c
ounter@
)
                endif
        endloop
endloop

loop    index:= numc, MAXPMF
        calc    pmfc(index):= 0
endloop

do      round_pmf(3)

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Rounds pmfa and pmfb to the nearest tenth, rounds pmfc to the neare@
st 100th
$$ fixes the last value of the pmf so that the probabilities will add @
up to 1.
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    round_pmf(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        i: index
        f: temp, sum

calc    sum:= 0
case    which
1
        loop    index:= 0,numa-1
                calc    temp:= pmfa(index)*10
                calc    pmfa(index):= round(temp)/10
                calc    sum:= sum + pmfa(index)
        endloop
        loop    index:= numa, 20
                calc    pmfa(index):= 0
        endloop
        calc    pmfa(numa-1):= pmfa(numa-1)+(1-sum)
2
        loop    index:= 0,numb-1
                calc    temp:= pmfb(index)*10
                calc    pmfb(index):= round(temp)/10
                calc    sum:= sum + pmfb(index)
        endloop
        loop    index:= numb, 20
                calc    pmfb(index):= 0
        endloop
        calc    pmfb(numb-1):= pmfb(numb-1)+(1-sum)
3
        loop    index:= 0,numc-1
                calc    temp:= pmfc(index)*100
                calc    pmfc(index):= round(temp)/100
        endloop
        loop    index:= numc, 40
                calc    pmfc(index):= 0
        endloop
endcase

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Jumps to the procedure to edit the pmfs
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    edit_pmf(which)@1ff8000
@1pl4007        i: which

jump    edit_pmf2(which)

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Updates the display of the probability of the value of a pmf, after@
 while it is editted
$$ index = the value that is being changed
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    update_graph(which, index)@1ff8000
        i: which, index
        merge, graphs:
        merge, pmfs:
        f: scalera, scalerb, sum
        i: index2

calc    scalera:= (xpos(1)-xneg(1))/(xmax(1)-xmin(1))     $$ how far to s@
pace each value printed
calc    scalerb:= (xpos(2)-xneg(2))/(xmax(2)-xmin(2))

if      which=1
        calc    sum:= 0
        pattern zpatterns, 16
        erase   xorigin(1)+(index-xmin(1)-.25)*scalera, 40; xorigin(1)+(index-x@
min(1)-.25)*scalera+15, 56
        erase   xorigin(1)+(numa-xmin(1)-.25)*scalera, 40; xorigin(1)+(numa-xmi@
n(1)-.25) *scalera+15,56
        erase   xorigin(3), 40; 499, 55
        at      xorigin(1)+(index-xmin(1)-.25)*scalera, 40
        show    pmfa(index)
        loop    index2:= 0, numa-1
                calc    sum:= sum + pmfa(index2)
        endloop
        at      xorigin(1)+(numa-xmin(1)-.25)*scalera, 40
$$      show    sum
        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                do      display_pmfvalue(3)
        endif
else
        calc    sum:= 0
        pattern zpatterns, 16
        erase   xorigin(2)+(index-xmin(2)-.25)*scalerb, 40; xorigin(2)+(index-x@
min(2)-.25)*scalerb+15, 55
        erase   xorigin(2)+(numb-xmin(2)-.25)*scalerb, 40; xorigin(2)+(numb-xmi@
n(2)-.25) *scalerb+15,55
        erase   xorigin(3), 40; 499, 55
        at      xorigin(2)+(index-xmin(2)-.25)*scalerb, 40
```

```
        show    pmfb(index)
        loop    index2:= 0, numb-1
                calc    sum:= sum + pmfb(index2)
        endloop
        at      xorigin(2)+(numa-xmin(2)-.25)*scalerb, 40
$$      show    sum
        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                do      display_pmfvalue(3)
        endif
endif

$$$$$$$$$$$$$$$$$$$$$$$$
$$ Edit the 2 pmfs, by dragging directly on the graphs.
$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    edit_pmf2(which)@1ff8000
        i: which, index, index2
        merge, editors:
        merge, graphs:
        merge, screenstatus:
        merge, slidervars:
        merge, pmfs:
        f: scalera, scalerb, sum
        i: oldx, oldy, newx, newy
        button: okbutton

calc    sum:= 0

do      update_pmf(1)   $$ display the 3 graphs
do      update_pmf(2)
do      update_pmf(3)

$$if    nextscreen-1=STATSCREEN & showcalcs=TRUE
        $$do    stat_page_stats
$$elseif        nextscreen-1=SLIDERSCREEN
        $$do    slide_graphs(FALSE)
$$endif

font    zserif, 48
at      xorigin(1)+20, yorigin(1)-ypos(1)-50
write   @1ff   1x@1ff8000
at      xorigin(2)-20, yorigin(2)-ypos(2)-50
write   @1fs  3c@1ff   1+@1fs8000@1ff8000
at      xorigin(2)+20,yorigin(2)-ypos(2)-50
write   @1ff   1y@1ff8000
at      xorigin(3)-20, yorigin(3)-ypos(3)-50
write   @1fs  3c@1ff   1=@1fs8000@1ff8000
at      xorigin(3)+20, yorigin(3)-ypos(3)-50
write   @1ff   1w@1ff8000

font    zserif, 16
do      display_pmfvalue(1)
do      display_pmfvalue(2)
do      display_pmfvalue(3)

calc    scalera:= (xpos(1)-xneg(1))/(xmax(1)-xmin(1))
calc    scalerb:= (xpos(2)-xneg(2))/(xmax(2)-xmin(2))

button  editA
pattern zpatterns, 16
erase   230, 460; 270, 499
box     230, 460; 270, 499
at      242, 473
write   @1fs  14@1ff   1OK@1fs8000@1ff8000
loop
        pause   keys=touch

        if      230<ztouchx<270 & 460<ztouchy<499       $$ TRUE means OK has been clicke@
d
                pause   keys=touch(left: up)
                if      230<ztouchx<270 & 460<ztouchy<499
                        do      check_valid_pmfs
                endif
        endif

        loop    index:= 0, numa-1
                calc    oldx:= ztouchx
                calc    oldy:= ztouchy
                if      xorigin(1)+(-xmin(1)+index-     barwidth(1)/2)*scalera<oldx<xorig
in(1@
)+(-xmin(1)+index+barwidth(1)/2)*scalera & yorigin(1)-ypos(1)<oldy<yor@
igin(1)
                        loop    zkey~=zk(left: up)      $$ go around till button is relea
sed
                                pause   keys=touch(left: move,up)
                                pattern zpatterns, 16
                                erase   170, 250; 300, 270
                                calc    newx:= ztouchx
                                calc    newy:= ztouchy
                                if      newy<oldy-5&pmfa(index)<1
                                        calc    oldy:= newy
                                        calc    pmfa(index):= pmfa(index)+.1
                                        do      update_pmf(1)
                                        pattern zpatterns, 16

                                elseif  newy>oldy+5&pmfa(index)>0
                                        calc    oldy:= newy
                                        calc    pmfa(index):= pmfa(index)-.1
                                        do      update_pmf(1)
                                endif
                                do      update_graph(1,index)
                                $$if    nextscreen-1=SLIDERSCREEN
                                $$      do      update_pmfslide(slidewhich, curvalue)
                                $$
                                $$      do      slide_graphs(FALSE)
                                $$endif
                        endloop

                endif
        endloop

        loop    index:= 0, numb-1
                calc    oldx:= ztouchx
                calc    oldy:= ztouchy
                if      xorigin(2)+(-xmin(2)+index-     barwidth(2)/2)*scalerb<oldx<xorig
in(2@
)+(-xmin(2)+index+barwidth(2)/2)*scalerb & yorigin(2)-ypos(2)<oldy<yor@
igin(2)
                        loop    zkey~=zk(left: up)
                                pause   keys=touch(left: move,up)
                                pattern zpatterns, 16
                                erase   170, 250; 300, 270
                                calc    newx:= ztouchx
                                calc    newy:= ztouchy
                                if      newy<oldy-5&pmfb(index)<1
                                        calc    oldy:= newy
                                        calc    pmfb(index):= pmfb(index)+.1
```

```
                                        do      update_pmf(2)
                                elseif  newy>oldy+5&pmfb(index)>0
                                        calc    oldy:= newy
                                        calc    pmfb(index):= pmfb(index)-.1
                                        do      update_pmf(2)
                                endif
                                do      update_graph(2,index)
                                $$if    nextscreen-1=SLIDERSCREEN
                                $$      do      update_pmfslide(slidewhich, curvalue)
                                $$      do      slide_graphs(FALSE)
                                $$endif
                        endloop
                endif
        endloop
endloop

do      jump_to_page(nextscreen-1)      $$ go back to the screen edit was called@
 from

$$$$$$$$$$$$$$$$$$$$$$$$
$$ Alternate method of editting pmfs, consisting of up and down arrow @
buttons
$$ NOT USED CURRENTLY, change edit_pmf to call this instead of calling@
 edit_pmf2
$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    main_edit_pmf(which)@1ff8000
@1pl4007        i: which, index
        merge, editors:
        merge, graphs:
        merge, pmfs:
        f: scaler, sum

calc    sum:= 0
do      update_pmf(1)
do      update_pmf(2)

if      which=1
        calc    scaler:= (xpos(which)-xneg(which))/(xmax(which)-xmin(which))
        loop    index:= 0,numa-1
                calc    sum:= sum + pmfa(index)
$$              slider  editslider1(index); xorigin(1)+(index-xmin(1)-.25)*scaler, @
140; xorigin(1)+(index-xmin(1)-.25)*scaler+20, 450; set_pmfa(index); v@
alue, pmfa(index); range, 0,1; line, .01; page, .1
                button  upbutton1(index); xorigin(1)+(index-xmin(1)-.25)*scaler+4, 15@
5; xorigin(1)+(index-xmin(1)-.25)*scaler+16, 175;uppmfa(index); "U"
                button  downbutton1(index); xorigin(1)+(index-xmin(1)-.25)*scaler+4, @
175; xorigin(1)+(index-xmin(1)-.25)*scaler+16, 195;downpmfa(index); "D@
"
                at      xorigin(1)+(index-xmin(1)-.25)*scaler, 140
                show    pmfa(index)
                at      xorigin(1)+(index-xmin(1)+.5)*scaler, 140
                if      index<numa-1
                        write   +
                else
                        write   =
                endif
        endloop
        at      xorigin(1)+(numa-xmin(1)-.25)*scaler,140
        show    sum
        calc    sum:= 0
        calc    scaler:= (xpos(2)-xneg(2))/(xmax(2)-xmin(2))
        loop    index:= 0, numb-1
                calc    sum:= sum + pmfb(index)
$$              slider  editslider2(index); xorigin(2)+(index-xmin(2)-.25)*scaler, @
140; xorigin(2)+(index-xmin(1)-.25)*scaler+20, 450; set_pmfb(index); v@
alue, pmfb(index); range, 0,1; line, .01; page, .1
                button  upbutton2(index); xorigin(2)+(index-xmin(2)-.25)*scaler+4, 15@
5; xorigin(2)+(index-xmin(2)-.25)*scaler+16, 175;uppmfb(index); "U"
                button  downbutton2(index); xorigin(2)+(index-xmin(2)-.25)*scaler+4, @
175; xorigin(2)+(index-xmin(2)-.25)*scaler+16, 195;downpmfb(index); "D@
"
                at      xorigin(2)+(index-xmin(2)-.25)*scaler, 140
                show    pmfb(index)
                at      xorigin(2)+(index-xmin(2)+.5)*scaler, 140
                if      index<numb-1
                        write   +
                else
                        write   =
                endif
        endloop
        at      xorigin(2)+(numb-xmin(2)-.25)*scaler, 140
        show    sum

        calc    sum:= 0
        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                calc    scaler:= (xpos(3)-xneg(3))/(xmax(3)-xmin(3))
                loop    index:= 0, numc-1
                        calc    sum:= sum + pmfc(index)
                        at      xorigin(3)+(index-xmin(3)-.25)*scaler, 140
                        show    pmfc(index)
                        at      xorigin(3)+(index-xmin(3)+.5)*scaler, 140
                        if      index<numc-1
                                write   +
                        else
                                write   =
                        endif
                endloop
                at      xorigin(3)+(numc-xmin(3)-.25)*scaler, 140
                show    sum
        endif
endif

$$ button       OKbutton; 170,400; 260,499; check_valid_pmfs; "OK"

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$
$$ Callback functions to be called when up arrow or down arrows are cl@
icked when editting pmfs
$$  NOT USED CURRENTLY
$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    uppmfa(curwhich)@1ff8000
@1pl4007        i: curwhich
        merge, pmfs:
        merge, graphs:
        merge, editors:
        f: scaler, sum, temp
        i: index

calc    sum:= 0
calc    scaler:= (xpos(1)-xneg(1))/(xmax(1)-xmin(1))

if      pmfa(curwhich)<1
        calc    temp:= pmfa(curwhich)+.1
        calc    pmfa(curwhich):=temp
        do      update_pmf(1)
        pattern zpatterns, 16
        erase   xorigin(1)+(curwhich-xmin(1)-.25)*scaler, 140; xorigin(1)+(curw@
```

```
hich-xmin(1)-.25)*scaler+15, 155
        erase   xorigin(1)+(numa-xmin(1)-.25)*scaler, 140; xorigin(1)+(numa-xmi@
n(1)-.25) *scaler+15,155
        erase   xorigin(3), 140; 499, 155
        do      update_pmf(1)
        at      xorigin(1)+(curwhich-xmin(1)-.25)*scaler, 140
        show    pmfa(curwhich)
        loop    index:= 0, numa-1
                calc    sum:= sum + pmfa(index)
        endloop
        at      xorigin(1)+(numa-xmin(1)-.25)*scaler, 140
        show    sum
        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                calc    scaler:= (xpos(3)-xneg(3))/(xmax(3)-xmin(3))
                loop    index:= 0, numc-1
                        calc    sum:= sum + pmfc(index)
                        at      xorigin(3)+(index-xmin(3)-.25)*scaler, 140
                        show    pmfc(index)
                        at      xorigin(3)+(index-xmin(3)+.5)*scaler, 140
                        if      index<numc-1
                                write   +
                        else
                                write   =
                        endif
                endloop
                at      xorigin(3)+(numc-xmin(3)-.25)*scaler, 140
                show    sum
        endif

endif
@1ff   1@1pl4006unit    uppmfb(curwhich)@1ff8000
@1pl4007        i: curwhich
        merge, pmfs:
        merge, graphs:
        merge, editors:
        f: scaler, sum, temp
        i: index

calc    sum:= 0
calc    scaler:= (xpos(2)-xneg(2))/(xmax(2)-xmin(2))

if      pmfb(curwhich)<1
        calc    temp:= pmfb(curwhich)+.1
        calc    pmfb(curwhich):=temp
        do      update_pmf(2)
        pattern zpatterns, 16
        erase   xorigin(2)+(curwhich-xmin(2)-.25)*scaler, 140; xorigin(2)+(curw@
hich-xmin(2)-.25)*scaler+15, 155
        erase   xorigin(2)+(numa-xmin(2)-.25)*scaler, 140; xorigin(2)+(numa-xmi@
n(2)-.25) *scaler+15,155
        erase   xorigin(3), 140; 499, 155
        do      update_pmf(1)
        at      xorigin(2)+(curwhich-xmin(2)-.25)*scaler, 140
        show    pmfb(curwhich)
        loop    index:= 0, numb-1
                calc    sum:= sum + pmfb(index)
        endloop
        at      xorigin(2)+(numb-xmin(2)-.25)*scaler, 140
        show    sum

        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                calc    scaler:= (xpos(3)-xneg(3))/(xmax(3)-xmin(3))
                loop    index:= 0, numc-1
                        calc    sum:= sum + pmfc(index)
                        at      xorigin(3)+(index-xmin(3)-.25)*scaler, 140
                        show    pmfc(index)
                        at      xorigin(3)+(index-xmin(3)+.5)*scaler, 140
                        if      index<numc-1
                                write   +
                        else
                                write   =
                        endif
                endloop
                at      xorigin(3)+(numc-xmin(3)-.25)*scaler, 140
                show    sum
        endif

endif

@1ff   1@1pl4006unit    downpmfa(curwhich)@1ff8000
@1pl4007        i: curwhich
        merge, pmfs:
        merge, graphs:
        merge, editors:
        f: scaler, sum, temp
        i: index

calc    sum:= 0
calc    scaler:= (xpos(1)-xneg(1))/(xmax(1)-xmin(1))

if      pmfa(curwhich)>0
        calc    temp:= pmfa(curwhich)-.1
        calc    pmfa(curwhich):=temp
        do      update_pmf(1)
        pattern zpatterns, 16
        erase   xorigin(1)+(curwhich-xmin(1)-.25)*scaler, 140; xorigin(1)+(curw@
hich-xmin(1)-.25)*scaler+15, 155
        erase   xorigin(1)+(numa-xmin(1)-.25)*scaler, 140; xorigin(1)+(numa-xmi@
n(1)-.25) *scaler+15,155
        erase   xorigin(3), 140; 499, 155
        do      update_pmf(1)
        at      xorigin(1)+(curwhich-xmin(1)-.25)*scaler, 140
        show    pmfa(curwhich)
        loop    index:= 0, numa-1
                calc    sum:= sum + pmfa(index)
        endloop
        at      xorigin(1)+(numa-xmin(1)-.25)*scaler, 140
        show    sum

        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                calc    scaler:= (xpos(3)-xneg(3))/(xmax(3)-xmin(3))
                loop    index:= 0, numc-1
                        calc    sum:= sum + pmfc(index)
                        at      xorigin(3)+(index-xmin(3)-.25)*scaler, 140
                        show    pmfc(index)
                        at      xorigin(3)+(index-xmin(3)+.5)*scaler, 140
                        if      index<numc-1
                                write   +
                        else
                                write   =
                        endif
                endloop
```

```
                at      xorigin(3)+(numc-xmin(3)-.25)*scaler, 140
                show    sum
        endif

endif

@1ff   1@1pl4006unit    downpmfb(curwhich)@1ff8000
@1pl4007        i: curwhich
        merge, pmfs:
        merge, graphs:
        merge, editors:
        f: scaler, sum, temp
        i: index

calc    sum:= 0
calc    scaler:= (xpos(2)-xneg(2))/(xmax(2)-xmin(2))

if      pmfb(curwhich)>0
        calc    pmfb(curwhich):= pmfb(curwhich)-.1
        pattern zpatterns, 16
        erase   xorigin(2)+(curwhich-xmin(2)-.25)*scaler, 140; xorigin(2)+(curw@
hich-xmin(2)-.25)*scaler+15, 155
        erase   xorigin(2)+(numa-xmin(2)-.25)*scaler, 140; xorigin(2)+(numa-xmi@
n(2)-.25) *scaler+15,155
        erase   xorigin(3), 140; 499, 155
        do      update_pmf(2)
        at      xorigin(2)+(curwhich-xmin(2)-.25)*scaler, 140
        show    pmfb(curwhich)
        loop    index:= 0, numb-1
                calc    sum:= sum + pmfb(index)
        endloop
        at      xorigin(2)+(numb-xmin(2)-.25)*scaler, 140
        show    sum

        do      convolve(1,1)
        do      update_pmf(3)
        if      numc<6
                calc    scaler:= (xpos(3)-xneg(3))/(xmax(3)-xmin(3))
                loop    index:= 0, numc-1
                        calc    sum:= sum + pmfc(index)
                        at      xorigin(3)+(index-xmin(3)-.25)*scaler, 140
                        show    pmfc(index)
                        at      xorigin(3)+(index-xmin(3)+.5)*scaler, 140
                        if      index<numc-1
                                write   +
                        else
                                write   =
                        endif
                endloop
                at      xorigin(3)+(numc-xmin(3)-.25)*scaler, 140
                show    sum
        endif

endif


@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Make sure the PMFs add up to 1; if not, give error message
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    check_valid_pmfs@1ff8000
        i: which
        merge, pmfs:
        merge, screenstatus:
        i: index
        f: sum

calc    sum:= 0
loop    index:= 0, numa-1
        calc    sum:= sum + pmfa(index)
endloop
if      sum =1
        calc    sum:= 0
        loop    index:= 0, numb-1
                calc    sum:= sum + pmfb(index)
        endloop
        if      sum=1
                do      jump_to_page(nextscreen-1)
        endif
endif

erase   170, 250; 300, 270
box     170, 250; 300, 270
box     172, 252; 298, 268
at      174, 250
write   PMFs must sum to 1!
$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Callback functions for a 3rd way of setting the PMFs, also not in u@
se.
$$ (a screen which brings up sliders to edit)
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    set_pmfa(curwhich)@1ff8000
@1pl4007        i: curwhich    $$$ curwhich == slider being changed
        f: scaler
        i: index
        f: sum
        merge, pmfs:
        merge, graphs:
        merge, editors:

calc    sum:= 0
slider  reset, editslider1(curwhich); value, round(100*zvalue(editslide@
r1(curwhich)))/100
calc    scaler:= (xpos(1)-xneg(1))/(xmax(1)-xmin(1))
pattern zpatterns, 16
erase   170,250; 270,400
erase   xorigin(1)+(curwhich-xmin(1)-.25)*scaler, 458; xorigin(1)+(curwh@
ich-xmin(1)-.25)*scaler+24, 499
erase   xorigin(1)+(numa-xmin(1)-.25)*scaler, 458; xorigin(1)+(numa-xmin@
(1)-.25) *scaler+24,499
calc    pmfa(curwhich):= zvalue(editslider1(curwhich))
do      update_pmf(1)
at      xorigin(1)+(curwhich-xmin(1)-.25)*scaler, 458
show    pmfa(curwhich)
loop    index:= 0, numa-1
        calc    sum:= sum + pmfa(index)
endloop
at      xorigin(1)+(numa-xmin(1)-.25)*scaler, 458
show    sum

@1ff   1@1pl4006unit    set_pmfb(curwhich)@1ff8000
@1pl4007        i: curwhich    $$$ curwhich == slider being changed
        f: scaler
        i: index
        f: sum
        merge, pmfs:
        merge, graphs:
        merge, editors:
```

```
calc    sum:= 0
slider  reset, editslider2(curwhich); value, round(100*zvalue(editslide@
r2(curwhich)))/100
calc    scaler:= (xpos(2)-xneg(2))/(xmax(2)-xmin(2))
pattern zpatterns, 16
erase   170,250; 270,400
erase   xorigin(2)+(curwhich-xmin(2)-.25)*scaler, 458; xorigin(2)+(curwh@
ich-xmin(2)-.25)*scaler+24, 499
erase   xorigin(2)+(numb-xmin(2)-.25)*scaler, 458; xorigin(2)+(numb-xmin@
(2)-.25) *scaler+24,499
calc    pmfb(curwhich):= zvalue(editslider2(curwhich))
do      update_pmf(2)
at      xorigin(2)+(curwhich-xmin(2)-.25)*scaler, 458
show    pmfb(curwhich)
loop    index:= 0, numb-1
        calc    sum:= sum + pmfb(index)
endloop
at      xorigin(2)+(numb-xmin(2)-.25)*scaler, 458
show    sum

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Set the number of values of pmf
$$ USE THIS INSTEAD OF calc     numa:= xx
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    set_num(which, newnum)@1ff8000
@1pl4007        i: which, newnum
        merge, pmfs:
        merge, graphs:

case    which
1
        calc    xmax(1):= newnum+1
        calc    numa:= newnum
        do      set_num(3,numa+numb-1)
2
        calc    xmax(2):= newnum+1
        calc    numb:= newnum
        do      set_num(3,numa+numb-1)
3
        calc    xmax(3):= newnum+1

        calc    numc:= newnum
endcase
@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Initializes the pmfs, n= # of values
$$ makes an equal probability for each of the values
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    init_pmf(which, n)@1ff8000
@1pl4007        i: n
        i: which                $$ init pmfa or pmfb
        i: index
        merge, pmfs:

do      set_num(which,n)
if      which=1
        loop    index:= 0,n-1
                calc    pmfa(index):= 1/n
        endloop
else
        loop    index:= 0,n-1
                calc    pmfb(index):= 1/n
        endloop
endif
do      round_pmf(which)
@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Initializes the variables for size/position/scale of graphs
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    init_graphs@1ff8000
@1pl4007        merge, graphs:

calc    xorigin(1):= 15
calc    yorigin(1):= 140
calc    xorigin(2):= 170
calc    yorigin(2):= 140
calc    xorigin(3):= 350
calc    yorigin(3):= 140
calc    xorigin(4):= 15
calc    yorigin(4):= 275

calc    xpos(1):= 130
calc    xneg(1):= -0
calc    ypos(1):= 80
calc    yneg(1):= -0

calc    xpos(2):= 130
calc    xneg(2):= -0
calc    ypos(2):= 80
calc    yneg(2):= -0

calc    xpos(3):= 145
calc    xneg(3):= -0
calc    ypos(3):= 80
calc    yneg(3):= -0

calc    xpos(4):= 365
calc    xneg(4):= -0
calc    ypos(4):= 80
calc    yneg(4):= -0

calc    xmax(1):= 20
calc    xmax(2):= 5
calc    xmax(3):= 20
calc    xmax(4):= 5
calc    ymax(1):= 1
calc    ymax(2):= 1
calc    ymax(3):= 1
calc    ymax(4):= 1

calc    xmin(1):= -1
calc    xmin(2):= -1
calc    xmin(3):= -1
calc    xmin(4):= -4

calc    barwidth(1):= .6
calc    barwidth(2):= .75
calc    barwidth(3):= .7

calc    barcolor(1):= 14
calc    barcolor(2):= 6
calc    barcolor(3):= 10

calc    stripscale:= (xpos(4)-xneg(4)) / (xmax(4)-xmin(4))

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Draws a bar graph for the appropriate pmf on the appropriate graph.
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    bar_graph(which)
```

```
@1ff8000@1pl4007        i: which@1ff   1
        @1ff8000merge, pmfs:
        merge, graphs:
        i: index

case    which
1
        loop    index:= 0,numa-1
                if      pmfa(index)<=1
                        if      pmfa(index)>=0
                                vbar    index,pmfa(index)
                        else
                                vbar    index, 0
                        endif
                else
                        vbar    index,1
                endif
        endloop
2
        loop    index:= 0,numb-1
                if      pmfb(index)<=1
                        if      pmfb(index)>=0
                                vbar    index,pmfb(index)
                        else
                                vbar    index, 0
                        endif
                else
                        vbar    index,1
                endif
        endloop
3
        loop    index:= 0,numc-1
                if      pmfc(index)<=1
                        if      pmfc(index)>=0
                                vbar    index,pmfc(index)
                        else
                                vbar    index, 0
                        endif
                else
                        vbar    index,1
                endif
        endloop
endcase

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Updates the graphs of the pmfs on the slider graph on the slider sc@
reen
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    update_pmfslide(which, offset)@1ff8000
@1pl4007        merge, graphs:
        merge, pmfs:
        i: which
        f: offset

do      set_graph(4)
pattern zpatterns,16
gerase  -3.99,0.01; xmax(4)+1,1
axes
pattern zpatterns,barcolor(1)
delta   barwidth(1)

if      which=1
        do      reverse_graph(1,offset)
        do      reverse_strip(1,offset)
else
        do      bar_graph(1)
        $$do    draw_strip(1)

endif
pattern zpatterns,barcolor(2)
delta   barwidth(2)
if      which=2
        do      reverse_graph(2,offset)
        do      reverse_strip(2,offset)
else
        do      bar_graph(2)
        $$do    draw_strip(2)
endif

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Draw the strip of probabilities for a PMF, for the slider screen
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    draw_strip(which)@1ff8000
        i: which
        merge, pmfs:
        merge, graphs:
        i: index

pattern zpatterns, 16
case    which
1
        erase   0,350; 405, 380
        at      xorigin(4)-(xmin(4)+1.5)*stripscale+15, 350
        font    zserif, 48
        write   @1ff   1x@1ff8000
        font    zserif, 16
        box     xorigin(4)-(xmin(4)+.5)*stripscale+5, 350; xorigin(4)+ stripscale@
*(numa-xmin(4)-.5), 380
        loop    index:= 0, numa-1
                at      xorigin(4)-(xmin(4)-index)*stripscale, 350
                show    pmfa(index)
        endloop
2
        erase   0,385; 405, 415
        at      xorigin(4)-(xmin(4)+1.5)*stripscale+15, 385
        font    zserif, 48
        write   @1ff   1y@1ff8000
        font    zserif, 16
        box     xorigin(4)-(xmin(4)+.5)*stripscale, 385; xorigin(4)+stripscale*(n@
umb-xmin(4)-.5), 415
        loop    index:= 0, numb-1
                at      xorigin(4)-(xmin(4)-index)*stripscale,385
                show    pmfb(index)
        endloop

endcase

$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Draw the reverse strip of probabilities for a PMF, for the slider s@
creen
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    reverse_strip(which,offset)@1ff8000
        i: which
        f: offset
        merge, pmfs:
        merge, graphs:
        i: index
```

```
pattern zpatterns,16
if      which=1
        erase   0,350; 408, 380
        at      xorigin(4)+(offset-numa-xmin(4)-1)*stripscale+25, 350
        font    zserif, 48
        showt   offset,1,0
        write    @1ff   1- x@1ff8000
        font    zserif, 16
        box     xorigin(4)+(offset-numa-xmin(4)+.5)*stripscale, 350; xorigin(4)+(@
offset-xmin(4)+.5)*stripscale, 380
        loop    index:= 0, numa-1
                at      xorigin(4)+(offset-xmin(4)-index)*stripscale, 350
                show    pmfa(index)
        endloop
else
        erase   0, 385; 408, 415
        at      xorigin(4)+(offset-numb-xmin(4)-1)*stripscale+25, 385
        font    zserif, 48
        showt   offset,1,0
        write   @1ff   1 - y@1ff8000
        font    zserif, 16
        box     xorigin(4)+(offset-numb-xmin(4)+.5)*stripscale, 385; xorigin(4)+(@
offset-xmin(4)+.5)*stripscale, 415
        loop    index:= 0, numb-1
                at      xorigin(4)+(offset-xmin(4)-index)*stripscale, 385
                show    pmfb(index)
        endloop
endif

@1ff   1unit    reverse_graph(which, offset)@1ff8000
@1pl4007        i: which
        f: offset
        merge,pmfs:
        merge,graphs:
        i: index

if      which=1
        loop    index:= 0, numa-1
                if      offset-index>xmin(4)+barwidth(1)
                        vbar    offset-index,pmfa(index)
                endif
        endloop
else
        loop    index:= 0, numb-1
                if      offset-index>xmin(4)+barwidth(2)
                        vbar    offset-index,pmfb(index)
                endif
        endloop
endif

@1pl4006$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Sets up the menus
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    menubar@1ff8000
menu    Screens; Title: jump_to_page(1)
menu    Screens; Convolution Equations: jump_to_page(2)
menu    Screens; Adding Independent Random Variables: jump_to_page(3)
menu    Screens; How Convolution Works: jump_to_page(4)
menu    Screens; Demo of Convolution: jump_to_page(5)
menu    Screens; Credits: jump_to_page(6)

menu    Edit A; Random: main_random(1)
menu    Edit A; Bernoulli: main_bernoulli(1)
$$menu  Edit A; Binomial: main_binomial(1)
$$menu  Edit A; Geometric
$$menu  Edit A; Pascal: main_pascal(1)
menu    Edit A; PMF C: generate_pmfc(1)

menu    Edit B; Random: main_random(2)
menu    Edit B; Bernoulli: main_bernoulli(2)
$$menu  Edit B; Binomial: main_binomial(2)
$$menu  Edit B; Geometric
$$menu  Edit B; Pascal: main_pascal(2)
menu    Edit B; PMF C: generate_pmfc(2)

menu    Help; About This Screen: help_page
$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Main units for generating preset PMFs.
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    generate_pmfc(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        merge, screenstatus:
        i: index

if      which=1
        if      numc<= 20
                do      set_num(1,numc)
        else
                do      set_num(1,20)
        endif
        loop    index:= 0, numa-1
                calc    pmfa(index):= pmfc(index)
        endloop
        do      convolve(1,1)
        do      update_pmf(1)
        do      update_pmf(3)
else
        if      numc<= 20
                do      set_num(2,numc)
        else
                do      set_num(2,20)
        endif
        loop    index:= 0, numb-1
                calc    pmfb(index):= pmfc(index)
        endloop
        do      convolve(1,1)
        do      update_pmf(2)
        do      update_pmf(3)
endif

do      jump_to_page(nextscreen-1)

@1ff   1@1pl4006unit    generate_bernoulli(which)@1ff8000
        i: which
        merge, pmfs:
        merge, graphs:

erase
font    zserif, 16
do      set_bernoulli(which+5)
do      update_pmf(1)
do      update_pmf(2)
do      update_pmf(3)

at      310, 220
write   Bernoulli Distribution, for PMF
if      which=1
```

```
        write   A
else
        write   B
endif

slider  Pslider; 310, 250; 330, 490; set_bernoulli(which); range, 0,1; @
line, .1; page, .1; value, curP

@1ff   1unit    generate_binomial(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        merge, graphs:

erase
do      set_binomial(which+5)
do      update_pmf(1)
do      update_pmf(2)
do      update_pmf(3)

at      310, 220
write   Binomial Distribution, for PMF
if      which=1
        write   A
else
        write   B
endif

slider  Pslider; 310, 250; 330, 490; set_binomial(which); range, 0,1; l@
ine, .05; page, .25; value, curP
slider  Nslider; 360, 250; 380, 490; set_binomial(which); range, 1,15; @
line, 1; page, 5; value, curN

@1ff   1@1pl4006unit    generate_pascal(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        merge, graphs:

erase
do      set_pascal(which+5)

do      update_pmf(1)
do      update_pmf(2)
do      update_pmf(3)

at      310, 220
write   Pascal Distribution, for PMF
if      which=1
        write   A
else
        write   B
endif
slider  Pslider; 310, 250; 330, 490; set_pascal(which); range, 0,1; lin@
e, .05; page, .25; value, curP
slider  Rslider; 365, 250; 385, 490; set_pascal(which); range, 1, 15; l@
ine, 1; page, 3; value, curN


@1pl4006
$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Units to jump to the main units
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    main_bernoulli(which)@1ff8000
@1pl4007        i: which

jump    generate_bernoulli(which)

@1ff   1@1pl4006unit    main_binomial(which)@1ff8000
@1pl4007        i: which

jump    generate_binomial(which)

@1ff   1@1pl4006unit    main_pascal(which)@1ff8000
@1pl4007        i: which

jump    generate_pascal(which)

@1pl4006
$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Callback functions for generating preset PMFs
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    set_bernoulli(which)@1ff8000
        i: which, init
        merge, pmfs:
        i: index

slider  reset, Pslider; value, round(zvalue(Pslider)*10)/10
if      which>3
        calc    init:=TRUE
        calc    which:= which-5
        slider  reset, Pslider; value, curP
else
        calc    init:= FALSE
        slider  reset, Pslider; value, round(zvalue(Pslider)*10)/10
        calc    curP:= zvalue(Pslider)
endif

if      which=1
        $$do    set_num(1,2)
        calc    numa:= 2
        calc    pmfa(1):= zvalue(Pslider)
        calc    pmfa(0):= 1-pmfa(1)
        do      round_pmf(1)
        loop    index:= 2, 20
                calc    pmfa(index):= 0
        endloop
        do      update_pmf(1)
else
        do      set_num(2,2)
        calc    pmfb(1):= curP
        calc    pmfb(0):= 1-pmfb(1)
        do      round_pmf(2)
        loop    index:= 2, 20
                calc    pmfb(index):= 0
        endloop
        do      update_pmf(2)
endif

do      convolve(1,1)
do      update_pmf(3)

@1ff   1unit    set_binomial(which)@1ff8000
@1pl4007        i: which, index
        merge, pmfs:

if      which>3
        calc    which:= which-5
        slider  reset, Pslider; value, curP
        slider  reset, Nslider; value, curN
```

```
else
        calc    curP:= zvalue(Pslider)
        calc    curN:= zvalue(Nslider)
endif

pattern zpatterns, 16
erase   340,260; 360,290
at      340,260
show    curP
erase   390,260;420,290
at      390,260
show    curN

if      which=1
        loop    index:= 0, numa-1
                if      index<=curN
                        if      curP=0 | curP=1
                                calc    pmfa(index):= 0
                                if      index=0 & curP = 0
                                        calc    pmfa(index):= gamma(curN+1) / (gamma(ind
ex+1) * gamma(curN-i@
ndex+1))*exp((curN-index)*ln(1-curP))
                                elseif  curN-index=0 & curP = 1
                                        calc    pmfa(index):= gamma(curN+1)/ (gamma(inde
x+1) * gamma(curN-in@
dex+1))*exp(index*ln(curP))
                                endif

                        else
                                calc    pmfa(index):= gamma(curN+1) / (gamma(index+1) *g
amma(curN-ind@
ex+1))*exp(index*ln(curP))*exp((curN-index)*ln(1-curP))
                        endif
                else
                        calc    pmfa(index):= 0
                endif
        endloop
        calc    numa:= numa+1
        do      round_pmf(1)
        calc    numa:= numa-1
        do      update_pmf(1)
else
        loop    index:= 0, numb-1
                if      index<=curN
                        if      curP=0 | curP=1
                                calc    pmfb(index):= 0
                                if      index=0 & curP = 0
                                        calc    pmfb(index):= gamma(curN+1) / (gamma(ind
ex+1) * gamma(curN-i@
ndex+1))*exp((curN-index)*ln(1-curP))
                                elseif  curN-index=0 & curP = 1
                                        calc    pmfb(index):= gamma(curN+1)/ (gamma(inde
x+1) * gamma(curN-in@
dex+1))*exp(index*ln(curP))
                                endif

                        else
                                calc    pmfb(index):= gamma(curN+1) / (gamma(index+1) *g
amma(curN-ind@
ex+1))*exp(index*ln(curP))*exp((curN-index)*ln(1-curP))
                        endif
                else
                        calc    pmfb(index):= 0
                endif
        endloop
        calc    numb:= numb+1
        do      round_pmf(2)
        calc    numb:= numb-1
        do      update_pmf(2)
endif

do      convolve(1,1)
do      update_pmf(3)

@1ff   1@1pl4006unit    set_pascal(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        i: index

slider  reset, Pslider; value, round(zvalue(Pslider)*100)/100
slider  reset, Rslider; value, round(zvalue(Rslider)*100)/100
if      which>3
        calc    which:= which-5
        slider  reset, Pslider; value, curP
        slider  reset, Rslider; value, curR
else
        calc    curP:= zvalue(Pslider)
        calc    curR:= zvalue(Rslider)
endif

pattern zpatterns, 16
erase   340,260; 363,290
at      340,260
show    curP
erase   390,260;420,290
at      390,260
show    curR

if      which=1
        loop    index:= 0, curR-1
                if      index<numa
                        calc    pmfa(index):= 0
                endif
        endloop
        loop    index:= curR, numa-1
                if      curP=0 | curP=1
                        calc    pmfa(index):= 0
                        if      index=curR & curP = 1
                                calc    pmfa(index):= gamma(index) / (gamma(curR) * gamma
(index-curR+@
1))*exp((curR)*ln(curP))
                        endif

                else
                        calc    pmfa(index):= gamma(index) / (gamma(curR) *gamma(index-cu
rR+1)@
)*exp(curR*ln(curP))*exp((index-curR)*ln(1-curP))
                endif
        endloop
        calc    numa:= numa+1
        do      round_pmf(1)
        calc    numa:= numa-1
        do      update_pmf(1)
else
        loop    index:= 0, curR-1
                if      index<numb
                        calc    pmfb(index):= 0
                endif
```

```
        endloop
        loop    index:= curR, numb-1
                if      curP=0 | curP=1
                        calc    pmfb(index):= 0
                        if      index=curR & curP = 1
                                calc    pmfb(index):= gamma(index) / (gamma(curR) * gamm
a(index-curR+@
1))*exp((curR)*ln(curP))
                        endif

                else
                        calc    pmfb(index):= gamma(index) / (gamma(curR) *gamma(index-c
urR+1)@
)*exp(curR*ln(curP))*exp((index-curR)*ln(1-curP))
                endif
        endloop
        calc    numb:= numb+1
        do      round_pmf(1)
        calc    numb:= numb-1
        do      update_pmf(2)

endif

do      convolve(1,1)
do      update_pmf(3)


@1ff   1@1pl4006unit    main_random(which)@1ff8000
@1pl4007        i: which
        merge, screenstatus:

do      generate_random(which)
do      jump_to_page(nextscreen-1)

@1ff   1@1pl4006unit    generate_random(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        merge, graphs:
        merge, screenstatus:
        merge, slidervars:
        i: index, sum

calc    sum:= 0

if      which=1
        loop    index:= 0, numa-1
                randu   pmfa(index), 100
                calc    sum:= sum + pmfa(index)
        endloop
        loop    index:= 0, numa-1
                calc    pmfa(index):= pmfa(index) / sum
        endloop
else
        loop    index:= 0, numb-1
                randu   pmfb(index), 100
                calc    sum:= sum + pmfb(index)
        endloop
        loop    index:= 0, numb-1
                calc    pmfb(index):= pmfb(index) / sum
        endloop
endif

do      round_pmf(which)
@1pl4006
@1ff   1unit    equation_page@1ff8000

do      set_screen(2)
font    zserif, 128
text    2,2; 350,350;
@1fs  14@1pl8000The convolution operation...@1fs8000
@1pl4006\
get     scr1;"conteqtn.gif"
put     scr1; 80,105
get     scr1;"disceqtn.gif"
put     scr1; 80, 215
$$text  80,105; 270, 300;

at      270, 350
write   ...for w@1ff   2 = x  + y @1ff8000and x@1ff   2, y@1ff8000 indep@
endent
do      next_button
do      prev_button

font    zserif, 24

$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Toggle whether pmfa or pmfb is flipped and slid
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    toggle_slide(which)@1ff8000
        merge, slidervars:
        merge, global:
        i: which

pattern zpatterns, 16
if      which=2
        button  reset, slidebuttona; value, FALSE
        button  reset, slidebuttonb; value, TRUE
        calc    slidewhich:= 1

        erase   0,420; 408, 440

        do      update_pmfslide(slidewhich, curvalue)
        do      slide_graphs(TRUE)
        do      draw_strip(2)
else
        button  reset, slidebuttona; value, TRUE
        button  reset, slidebuttonb; value, FALSE
        calc    slidewhich:= 2
        erase   0,420; 450, 440
        do      update_pmfslide(slidewhich, curvalue)
        do      slide_graphs(TRUE)
        do      draw_strip(1)
endif

$$if    slidewhich= 1
$$else
$$      button  slidewhichbutton; 100,300; 160,300; toggle_slide(2); "Slide @
PMF $$B"; erase
$$endif

$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Screen for showing the sample-space method of adding RV's.
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    stat_page@1ff8000
        merge, pmfs:
        merge, graphs:
        merge, screenstatus:
```

```
        i: indexa, indexb, indexc
        f: scaler
        button: showstats

do      next_button
do      prev_button

if      showcalcs=TRUE
        jump    stat_page_stats2
endif
font    zserif, 16
do      update_pmf(1)
do      update_pmf(2)
do      make_edit_buttons
$$do    edit_pmf2(1)
do      convolve(1,1)
do      update_pmf(3)

font    zserif, 66
at      xorigin(1)+20, yorigin(1)-ypos(1)-50
write   @1ff   1x@1ff8000
at      xorigin(2)-20, yorigin(2)-ypos(2)-50
write   @1fs  3c@1ff   1+@1fs8000@1ff8000
at      xorigin(2)+20,yorigin(2)-ypos(2)-50
write   @1ff   1y@1ff8000
at      xorigin(3)-20, yorigin(3)-ypos(3)-50
write   @1fs  3c@1ff   1=@1fs8000@1ff8000
at      xorigin(3)+20, yorigin(3)-ypos(3)-50
write   @1ff   1w@1ff8000

font    zserif, 16
do      display_pmfvalue(1)
do      display_pmfvalue(2)
do      display_pmfvalue(3)
button  showstats; 190, 440; 310, 499; jump_stat_page; "Show Calculatio@
ns"

@1ff   1unit    jump_stat_page@1ff8000
        merge, screenstatus:

calc    showcalcs:= TRUE
jump    stat_page_stats2

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Called when "Show Calculations" is clicked, displays the tree/table@
 of sample space calcs
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    stat_page_stats@1ff8000
        merge, pmfs:
        merge, graphs:
        i: indexa, indexb, indexc
        f: scaler

erase
font    zserif, 48
at      xorigin(1)+20, yorigin(1)-ypos(1)-50
write   @1ff   1x@1ff8000
at      xorigin(2)-20, yorigin(2)-ypos(2)-50
write   @1fs  3c@1ff   1+@1fs8000@1ff8000
at      xorigin(2)+20,yorigin(2)-ypos(2)-50
write   @1ff   1y@1ff8000
at      xorigin(3)-20, yorigin(3)-ypos(3)-50
write   @1fs  3c@1ff   1=@1fs8000@1ff8000
at      xorigin(3)+20, yorigin(3)-ypos(3)-50
write   @1ff   1z@1ff8000
do      next_button
do      prev_button

font    zserif, 16

do      update_pmf(1)
do      update_pmf(2)
do      display_pmfvalue(1)
do      display_pmfvalue(2)
do      display_pmfvalue(3)
do      make_edit_buttons
$$do    main_edit_pmf(1)
do      convolve(1,1)
do      update_pmf(3)

draw    0,197; 320, 197;320, 440; 0, 440; 0, 197
draw    0,233; 320, 233; 320, 234; 0, 234
draw    85, 197; 85, 440; 170, 440; 170, 197

at      18,210
write   [ x , y ]
at      7,195
write   Sample Points
at      129, 195
write   w
at      115, 210
write   = x + y
at      190, 195
write   Probability of Event [x,y]
at      215, 210
write   P(x) x P(y)
at      360, 195
write   Probability of Events [w]
at      360, 210

calc    scaler:= 200/(numa*numb)

loop    indexa:= 0, numa-1
        loop    indexb:= 0, numb-1
                draw    0,235+(indexa*numb+indexb)*scaler-2; 320, 235+(indexa*numb+inde@
xb)*scaler-2
                at      20, 235+(indexa*(numb)+indexb)*scaler
                write   [
                show    indexa
                write    ,
                show    indexb
                write    ]

                at      130, 235+(indexa*(numb)+indexb)*scaler
                show    indexa+indexb

                at      200,235+(indexa*(numb)+indexb) * scaler
                showt   pmfa(indexa),0,2
                write    x
                showt   pmfb(indexb),0,2
                write    @1fs  14= @1fs8000
                showt   pmfa(indexa)*pmfb(indexb),0,2
        endloop
endloop

loop    indexc:= 0, numc-1
        at      340, 225+indexc * 30
        write   P(C=
```

```
        show    indexc
        write   ) =
        loop    indexa:= 0, numa-1
                loop    indexb:= 0, numb-1
                        if      (indexa + indexb = indexc) & (pmfa(indexa)*pmfb(indexb)~
=0)
                                show    round(pmfa(indexa)*pmfb(indexb)*100)/100
                                if      indexa=indexc | indexa=numa-1
                                        write    =
                                else
                                        write    +
                                endif
                        endif
                endloop
        endloop

        show    pmfc(indexc)
endloop

@1ff   1unit    stat_page_stats2@1ff8000
        merge, pmfs:
        merge, graphs:
        merge, statvars:
        i: indexa, indexb, indexc

erase
font    zserif, 48
at      xorigin(1)+20, yorigin(1)-ypos(1)-50
write   @1ff   1x@1ff8000
at      xorigin(2)-20, yorigin(2)-ypos(2)-50
write   @1fs  3c@1ff   1+@1fs8000@1ff8000
at      xorigin(2)+20,yorigin(2)-ypos(2)-50
write   @1ff   1y@1ff8000
at      xorigin(3)-20, yorigin(3)-ypos(3)-50
write   @1fs  3c@1ff   1=@1fs8000@1ff8000
at      xorigin(3)+20, yorigin(3)-ypos(3)-50
write   @1ff   1w@1ff8000
do      next_button
do      prev_button

font    zserif, 16
do      update_pmf(1)
do      update_pmf(2)
do      display_pmfvalue(1)
do      display_pmfvalue(2)
do      display_pmfvalue(3)
do      make_edit_buttons
$$do    main_edit_pmf(1)
do      convolve(1,1)
do      update_pmf(3)

draw    60, 215; 60, 435; 305, 435; 305, 215; 60, 215

calc    scalerx:= 245/(numa)
calc    scalery:= 220/(numb)

draw    329, 225+highlightc*30; 499, 225+highlightc*30; 499, 225+highligh@
tc*30+20; 329, 225+highlightc*30+20; 329, 225+highlightc*30

loop    indexb:= 0, numb-1
        $$at    0,215+(indexb+.5)*scalery-20
        $$write y =
        $$show  indexb
        at      0, 215+(indexb+.5)*scalery
        write   P(y=
        show    indexb
        write   ) =
        show    pmfb(indexb)
endloop

loop    indexa:= 0, numa-1
        $$at    60+scalerx*(indexa)+10, 175
        $$write x =
        $$show  indexa
        at      60+scalerx*(indexa)+10, 195
        write   P(x=
        show    indexa
        write   ) =
        show    pmfa(indexa)
        loop    indexb:= 0, numb-1
                draw    60+scalerx*indexa, 215; 60+scalerx*indexa, 435
                draw    60, 215+scalery*indexb; 305, 215+scalery*indexb
                do      fill_box(indexa, indexb)
        endloop
endloop

loop    indexc:= 0, numc-1
        at      330, 225+indexc * 30
        write   P(w=
        show    indexc
        write   ) =
        loop    indexa:= 0, numa-1
                loop    indexb:= 0, numb-1
                        if      (indexa + indexb = indexc)
                                show    round(pmfa(indexa)*pmfb(indexb)*100)/100
                                if      indexa=indexc | indexa=numa-1
                                        write    =
                                else
                                        write    +
                                endif
                        endif
                endloop
        endloop

        show    pmfc(indexc)
endloop

at      400, 430
write   Sum =
at      435, 430
do      show_highlights(highlightc)
$$show  highlightc
slider  slidec; 380, 450; 480, 470; highlight_boxes(FALSE); value, high@
lightc; horizontal; range, 0, numc-1; line, 1; page, 1

@1ff   1unit    highlight_boxes(init)@1ff8000
        i: init
        merge, statvars:

if      round(zvalue(slidec))~=highlightc
        do      show_highlights(round(zvalue(slidec)))
        do      set_graph(3)
        mode    rewrite
$$      if      abs(curvalue-lastvalue)>.5
$$              if      lastvalue>=0
$$                      if      pmfc(lastvalue)<=1
$$                              if      pmfc(lastvalue)>=0
$$                                      vbar    $$round(lastvalue),pmfc(lastvalue)
```

```
$$                              else
$$                                      vbar    round(lastvalue),0
$$                              endif
$$                      else
$$                              vbar    round(lastvalue),1
$$                      endif
$$
$$                      mode    write
$$              endif
$$              calc    lastvalue:= round(curvalue)
$$              if      curvalue>=0
$$                      if      pmfc(curvalue)<=1
$$                      if      pmfc(curvalue)>=0
$$                              vbar    $$round(curvalue),pmfc(curvalue)
$$                      else
$$                              vbar    round(curvalue),0
$$                      endif
$$              else
$$                      vbar    round(curvalue),1
$$              endif
$$      endif
        mode    write

endif

@1ff   1unit    fill_box(a,b)@1ff8000
        i: a, b
        merge, pmfs:
        merge, statvars:

at      60+scalerx*a+4,215+scalery*(.5+b)-20
write   w =
show    a
write    +
show    b
write    =
show    a+b

at      60+scalerx*a+4, 215+scalery*(.5+b)
show    pmfa(a)
write    x
show    pmfb(b)
write    =
show    pmfa(a)*pmfb(b)

@1ff   1unit    show_highlights(c)@1ff8000
        i: c
        merge, statvars:
        merge, pmfs:
        i: indexa, indexb

pattern zpatterns, 16
erase   435, 430; 460, 448
at      435, 430
show    c
loop    indexa:= 0, numa-1
        loop    indexb:= 0, numb-1
                if      indexa+indexb=c | indexa+indexb=highlightc
                        do      toggle_highlight_box(indexa, indexb)
                endif
        endloop
endloop

mode    xor
draw    329, 225+highlightc*30; 499, 225+highlightc*30; 499, 225+highligh@
tc*30+20; 329, 225+highlightc*30+20; 329, 225+highlightc*30
draw    329, 225+c*30; 499, 225+c*30; 499, 225+c*30+20; 329, 225+c*30+20;@
 329, 225+c*30
mode    write

$$loop  indexa:= 0, numa-1
$$      loop    indexb:= 0, numb-1
$$              if      (indexa + indexb = c) & (pmfa(indexa)*pmfb(indexb)~=0)
$$                      at      340, 225+c*30
$$                      show    round(pmfa(indexa)*pmfb(indexb)*100)/100
$$                              write    =
$$                      else
$$                              write    +
$$                      endif
$$              endif
$$      endloop
$$endloop
$$show  pmfc(c)
$$at    340, 225+c * 30
$$write P(C=
$$show  c
$$write ) =

calc    highlightc:= c

@1ff   1unit    toggle_highlight_box(a,b)@1ff8000
        i: a,b
        merge, statvars:

mode    xor
draw    60+scalerx*a+1, 215+scalery*b+1; 60+scalerx*(a+1)-1, 215+scalery*@
b+1;
        60+scalerx*(a+1)-1,215+scalery*(b+1)-1; 60+scalerx*a+1, 215+scalery*(@
b+1)-1;
        60+scalerx*a+1, 215+scalery*b+1
draw    60+scalerx*a+2, 215+scalery*b+2; 60+scalerx*(a+1)-2, 215+scalery*@
b+2;
        60+scalerx*(a+1)-2, 215+scalery*(b+1)-2; 60+scalerx*a+2, 215+scalery*@
(b+1)-2;
        60+scalerx*a+2, 215+scalery*b+2
draw    60+scalerx*a+3, 215+scalery*b+3; 60+scalerx*(a+1)-3, 215+scalery*@
b+3;
        60+scalerx*(a+1)-3, 215+scalery*(b+1)-3; 60+scalerx*a+3, 215+scalery*@
(b+1)-3;
        60+scalerx*a+3, 215+scalery*b+3
mode    write

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Screen for the demonstration of convolution sliding
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    slider_page@1ff8000
        merge, slidervars:
        merge, global:
        merge, graphs:
$$do    next_button
do      prev_button

font    zserif, 48
at      xorigin(1)+20, yorigin(1)-ypos(1)-50
write   @1ff   1x@1ff8000
at      xorigin(2)-20, yorigin(2)-ypos(2)-50
write   @1fs  50@1ff   1+@1fs8000@1ff8000
at      xorigin(2)+20,yorigin(2)-ypos(2)-50
```

```
write   @1ff   1y@1ff8000
at      xorigin(3)-20, yorigin(3)-ypos(3)-50
write   @1fs  3c@1ff   1=@1fs8000@1ff8000
at      xorigin(3)+20, yorigin(3)-ypos(3)-50
write   @1ff   1w@1ff8000
font    zserif, 16
do      set_graph(3)
do      set_graph(4)
do      convolve(1,1)
do      update_pmf(1)
do      update_pmf(2)
$$do    update_pmf(3)
do      set_graph(3)
do      display_pmfvalue(1)
do      display_pmfvalue(2)
do      display_pmfvalue(3)
do      make_edit_buttons
$$do    main_edit_pmf(1)

if      slidewhich=2
        button  slidebuttona; 420, 220; 499, 240; toggle_slide(1); "Flip & Sli@
de B"; erase;radio; value, TRUE
        button  slidebuttonb; 420,200; 499, 220; toggle_slide(2); "Flip & Slid@
e A"; erase; radio; value, FALSE
else
        button  slidebuttona; 420, 220; 499, 240; toggle_slide(slidewhich); "F@
lip & Slide B"; erase;radio; value, FALSE
        button  slidebuttonb; 420,200; 499, 220; toggle_slide(slidewhich); "Fl@
ip & Slide A"; erase; radio; value, TRUE
endif


calc    lastvalue:= curvalue

do      slide_graphs(TRUE)

$$   value == initial value;   line == increment for pressing on arrow@
;
$$   page == increment for pressing in slider
slider  graphslider; 120, 300; 380, 330; slide_graphs(FALSE); horizonta@
l; value, curvalue; range, -1, 5; line, .1; page, 1
at      400,417
write   =

if      slidewhich=2
        do      draw_strip(1)
else
        do      draw_strip(2)
endif
do      update_pmfslide(slidewhich,curvalue)

$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Callback function for the graphs being slid
$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    slide_graphs(init)@1ff8000
        i: init
        merge, slidervars:
        merge, graphs:
        merge,pmfs:
        f: temp
        i: index

if      zleftdown=FALSE
        slider  reset, graphslider; value, round(zvalue(graphslider))
endif

if      init=FALSE
        calc    curvalue:= zvalue(graphslider)  $$ save the state of the slider
endif

if      abs(curvalue-lastvalue)>.5
        if      lastvalue>=0
                do      set_graph(3)
                mode    rewrite
                if      pmfc(lastvalue)<=1
                        if      pmfc(lastvalue)>=0
                                vbar    round(lastvalue),pmfc(lastvalue)
                        else
                                vbar    round(lastvalue),0
                        endif
                else
                        vbar    round(lastvalue),1
                endif
                mode    write
                pattern zpatterns, 16
                erase   xorigin(4),420; 408, 440
        endif
endif

pattern zpatterns, 16
erase   408,260; 499, 499
at      412,417-(curvalue)*20-20
font    zserif, 48
write   @1ff   1w@1ff8000
font    zserif, 16
box     409,417-(curvalue)*20; 441, 417+(numc-curvalue)*20

calc    temp:= xorigin(4)-xneg(4)-35
loop    index:= 0, numc-1
        at      415,417-(curvalue-index)*20
        showt   pmfc(index),0,2
endloop

if      abs(curvalue-round(curvalue))<.1
        do      set_graph(3)
        mode    rewrite

        calc    lastvalue:= round(curvalue)
        pattern zpatterns,16
        if      curvalue>=0
                if      pmfc(curvalue)<=1
                        if      pmfc(curvalue)>=0
                                vbar    round(curvalue),pmfc(curvalue)
                        else
                                vbar    round(curvalue),0
                        endif
                else
                        vbar    round(curvalue),1
                endif
        endif
        mode    write
        draw    410, 417; 440,417; 440, 437; 410, 437; 410, 417
        at      445, 417
        write   =P(C=
        show    round(curvalue)
        write   )
        if      slidewhich=2
                loop    index:= 0, numa-1
```

```
                        if      lastvalue-index>=0
                                at      xorigin(4)+stripscale*(-xmin(4)+index),420
                                showt   round(100*pmfa(index)*pmfb(round(curvalue)-index
))/100,0,2
                                if      lastvalue-index>0 & index < numa-1
                                        at      xorigin(4)+stripscale*(-xmin(4)+index+.6
),420
                                        write   +
                                endif
                        endif
                endloop
        else
                loop    index:= 0, numb-1
                        if      lastvalue-index>= 0
                                at      xorigin(4)+stripscale * (-xmin(4) + index), 420
                                showt   round(100*pmfb(index)*pmfa(round(curvalue)-index
))/100,0,2
                                if      lastvalue-index>0 & index < numb-1
                                        at      xorigin(4)+stripscale* (-xmin(4)+index+.
6),420
                                        write   +
                                endif
                        endif
                endloop
        endif

elseif  abs(curvalue-round(curvalue))>.2
        do      set_graph(3)
        mode    rewrite
        if      lastvalue>=0
                if      pmfc(lastvalue)<=1
                        if      pmfc(lastvalue)>=0
                                vbar    round(lastvalue),pmfc(lastvalue)
                        else
                                vbar    round(lastvalue),0
                        endif
                else
                        vbar    round(lastvalue),1
                endif
        endif
        mode    write
        pattern zpatterns, 16
$$      erase   xorigin(4),410;@1pl800098,484
@1pl4006endif
do      update_pmfslide(slidewhich, curvalue)

$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Display the values of the pmfs under the appropriate graphs
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    display_pmfvalue(which)@1ff8000
        i: which
        merge, pmfs:
        merge, graphs:
        f: scaler, sum
        i: index

calc    scaler:= (xpos(which)-xneg(which))/(xmax(which)-xmin(which))

calc    sum:= 0
case    which
1
        loop    index:= 0, numa-1
                $$calc  sum:= sum + pmfa(index)
                at      xorigin(1)+(index-xmin(1)-.25)*scaler, 40
                show    pmfa(index)
                at      xorigin(1)+(index-xmin(1)+.5)*scaler, 40
                $$if    index<numa-1
                $$      write   +
                $$else
                $$      write   =
                $$endif
        endloop
        $$at    xorigin(1)+(numa-xmin(1)-.25)*scaler,40
        $$show  sum
2
        loop    index:= 0, numb-1
                $$calc  sum:= sum + pmfb(index)
                at      xorigin(2)+(index-xmin(2)-.25)*scaler, 40
                show    pmfb(index)
                at      xorigin(2)+(index-xmin(2)+.5)*scaler, 40
                $$if    index<numb-1
                $$      write   +
                $$else
                $$      write   =
                $$endif
        endloop
        $$at    xorigin(2)+(numb-xmin(2)-.25)*scaler,40
        $$show  sum
3
        loop    index:= 0, numc-1
                calc    sum:= sum + pmfc(index)
                at      xorigin(3)+(index-xmin(3)-.25)*scaler, 40
                show    pmfc(index)
                at      xorigin(3)+(index-xmin(3)+.5)*scaler, 40
                $$if    index<numc-1 & numc<5
                $$      write   +
                $$endif
        endloop
        $$if    numc<5
        $$      at      xorigin(3)+(numc-xmin(3))*scaler,40
        $$      write   =
        $$      show    sum
        $$endif
endcase

****************************
* Update the pmf on the screen
****************************
$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Update the display of the pmf on the graph
$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    update_pmf(which)@1ff8000
        i: which
        merge, graphs:

pattern zpatterns,16
erase   xorigin(which) - xneg(which)-15, yorigin(which)-ypos(which); xor@
igin(which)+xpos(which)+8, yorigin(which)-yneg(which)+20
do      set_graph(which)
do      bar_graph(which)
at      xorigin(which), yorigin(which)+20
write   possible values for random variable
case    which
1
        write   x
2
        write   y
3
```

```
        write   w
endcase

at      xorigin(which)-10, yorigin(which)-ypos(which)-20
write   P(
case    which
1
        write   x
2
        write   y
3
        write   w
endcase
write   )

if      which=1
$$      button  editA; 210, 10; 248, 40; edit_pmf(1); "Edit A"
elseif  which=2
$$      button  editB; 460, 10; 499, 40; edit_pmf(2); "Edit B"
endif

@1ff   1unit    make_edit_buttons@1ff8000
        merge, graphs:

button  editA; 330, 459; 360, 499; edit_pmf(1); "Edit"

@1ff   1unit    convolution_page@1ff8000
        merge, graphs:
        merge, pmfs:
        merge, convolutionvars:

erase
font    zserif, 48
at      xorigin(1)+20, yorigin(1)-ypos(1)-50
show    multipliera
write    x PMF A
at      xorigin(2)-20, yorigin(2)-ypos(2)-55
write   @1fs  3c@1ff   1+@1fs8000@1ff8000
at      xorigin(2)+20,yorigin(2)-ypos(2)-50
show    multiplierb
write    x PMF B
at      xorigin(3)-20, yorigin(3)-ypos(3)-55
write   @1fs  3c@1ff   1=@1fs8000@1ff8000
at      xorigin(3)+20, yorigin(3)-ypos(3)-50
write   PMF C
font    zserif, 16
do      update_pmf(1)
do      update_pmf(2)
do      update_pmf(3)
do      next_button
do      prev_button

at      xorigin(1)+xpos(1)/2, 180
show    numa
slider  slidenuma; xorigin(1), 200; xorigin(1)+xpos(1), 220; set_numsli@
de(1); horizontal; value, numa; range, 1,10; line, 1
at      xorigin(2)+xpos(2)/2, 180
show    numb
slider  slidenumb; xorigin(2), 200; xorigin(2)+xpos(2), 220; set_numsli@
de(2); horizontal; value, numb; range, 1,10; line, 1
at      xorigin(3), 202
write   Number of Values

$$slider        slidemulta; xorigin(1), 230; xorigin(1)+xpos(1), 235; set_mul@
ta(1); horizontal; value, multipliera; range, 1, 10; line, 1

@1ff   1unit    credits_page@1ff8000

erase
do      prev_button
at      150, 150

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$ Callback function for the sliders to set # of values for the pmfs
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
@1ff   1unit    set_numslide(which)@1ff8000
@1pl4007        i: which
        merge, pmfs:
        merge, graphs:
        merge, convolutionvars:

if      which=1
        if      numa>zvalue(slidenuma)
                do      round_pmf(1)
        endif
        do      set_num(1,zvalue(slidenuma))
        pattern zpatterns, 16
        erase   xorigin(1)+xpos(1)/2, 180; xorigin(1)+xpos(1)/2+20, 198
        at      xorigin(1)+xpos(1)/2, 180
        show    numa
        do      update_pmf(1)
else
        if      numb>zvalue(slidenumb)
                do      round_pmf(2)
        endif
        do      set_num(2,zvalue(slidenumb))
        pattern zpatterns, 16
        erase   xorigin(2)+xpos(2)/2, 180; xorigin(2)+xpos(2)/2+20, 198
        at      xorigin(2)+xpos(2)/2, 180
        show    numb
        do      update_pmf(2)
endif

do      update_pmf(3)

****************************
* Set the current graph to 1, 2, 3, or 4
****************************
@1ff   1@1pl4006unit    set_graph(curgraph)@1ff8000
        i: curgraph@1ff   1
        @1ff8000merge, graphs:

gorigin xorigin(curgraph), yorigin(curgraph)
axes    xneg(curgraph),yneg(curgraph); xpos(curgraph), ypos(curgraph)
bounds  xneg(curgraph),yneg(curgraph); xpos(curgraph), ypos(curgraph)
scalex  xmax(curgraph), xmin(curgraph)
scaley  ymax(curgraph)
delta   barwidth(curgraph)
pattern zpatterns,barcolor(curgraph)
if      xmax(curgraph)<=12
        labelx  1
else
        labelx  5,1
endif

labely  .5,.1
axes
```

```
@1ff   1unit    set_screen(curscreen)@1ff8000
@1pl4007        i: curscreen
        merge,screenstatus:

calc    nextscreen:= curscreen+1

calc    prevscreen:= curscreen-1

@1ff   1@1pl4006unit    next_button
@1pl4007        @1ff8000merge, global:
        merge, screenstatus:

font    zserif, 36
button  nextbutton; 68,440; 133,499; jump_to_page(nextscreen); "Next"
font    zserif, 16

@1ff   1@1pl4006unit    prev_button@1ff8000
@1pl4007        merge, global:
        merge, screenstatus:

if      prevscreen>0
        font    zserif,36
        button  prevbutton; 0, 440; 65,499; jump_to_page(prevscreen); "Prev"
        font    zserif, 16
endif

@1ff   1@1pl4006unit    jump_to_page(new_unit)
        @1ff8000i: new_unit
        merge, pmfs:

do      set_screen(new_unit)

case    new_unit
1
        jump    title_page
2
        jump    equation_page
3
        if      numa>3
                do      set_num(1,3)
                do      round_pmf(1)
        endif
        if      numb>3
                do      set_num(2,3)
                do      round_pmf(2)
        endif
        jump    stat_page
4
        if      numa>3
                do      set_num(1,3)
                do      round_pmf(1)
        endif
        if      numb>4
                do      set_num(2,4)
                do      round_pmf(2)
        endif

        jump    slider_page
5
$$      do      set_num(1,10)
$$      do      generate_random(1)
$$      do      set_num(2,10)
$$      do      generate_random(2)
        jump    convolution_page
6
        do      credits_page
endcase

@1ff   1unit    help_page@1ff8000
        merge, screenstatus:
        button: okbutton

erase
button  okbutton; 230, 470; 270, 499; jump_to_page(nextscreen-1); "OK"
pause@cT end
```