The Effect of the Aging Lens on Color Vision

Glenn Davis <gdavis@gluonics.com>

February 10, 2025

The goal of this **colorSpec** vignette is to simulate the effect of age on human color vision. The colored figures are best viewed by an observer with age of 32 years, and on a display calibrated for sRGB.

The Human Lens

It is well known that the lens in the human eye does not transmit all wavelengths equally; the transmission at short wavelengths is less, which means the lens is yellowish. In the UV there is very little transmission; which is a good thing since the lens protects the retina from UV damage. It is also well known that the lens gets yellower with age. When making these colored images it is appropriate to use the CIE 1931 color matching functions (CMFs). Unfortunately I could not find the average age of the observers used to establish the 1931 standard observer (there were 17 of them). But it *is* known that the average age of the observers used to create the CIE 1964 standard observer is 32 years, see [Pok87]. So we'll take 32 years for the CIE 1931 standard observer as well. Featured functions in this vignette are: `linearize()`, `lensAbsorbance()`, `extradata()`, `applyspec()`, and `calibrate()`.

Start the **R** session and load the **colorSpec** package,

```
library( colorSpec )
library( spacesXYZ )    # for function standardXYZ()
library( spacesRGB )    # for functions RGBfromXYZ() and plotPatchesRGB()
```

Compute and plot lens transmittance at 32 and 64 years using the model in [Pok87].

```
lens.trans = linearize( lensAbsorbance( c(32,64), wave=380:780 ) )
par( omi=c(0,0,0,0), mai=c(0.6,0.7,0.3,0.2) )
plot( lens.trans, color='black', lty=1:2, main=FALSE, legend='topleft' )
```

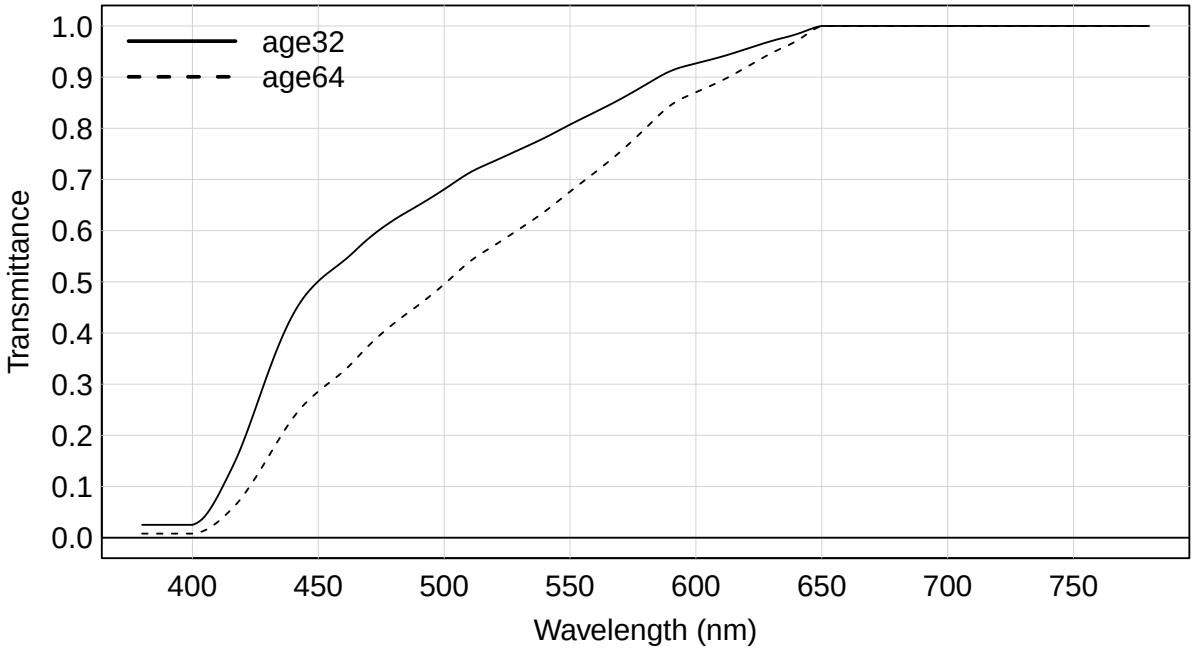

Figure 1: Human Lens Transmittance at age=32 and age=64

To compare the color appearance at age 64 to that at age 32, we need the transmittance at age 64 relative to that at age 32. We know that object `lens.trans` is a matrix, so use the standard **R** matrix subset operation to extract each spectrum. Then perform the division and plot the ratio.

```
lens.64 = lens.trans[ ,2] / lens.trans[ ,1]
lens.64 = colorSpec( lens.64, wavelength(lens.trans), 'transmittance' )
specnames(lens.64) = "trans.64 / trans.32"
par( omi=c(0,0,0,0), mai=c(0.6,0.7,0.3,0.2) )
plot( lens.64, main=TRUE, legend=FALSE, ylab='Relative Transmittance', col='black' )
```

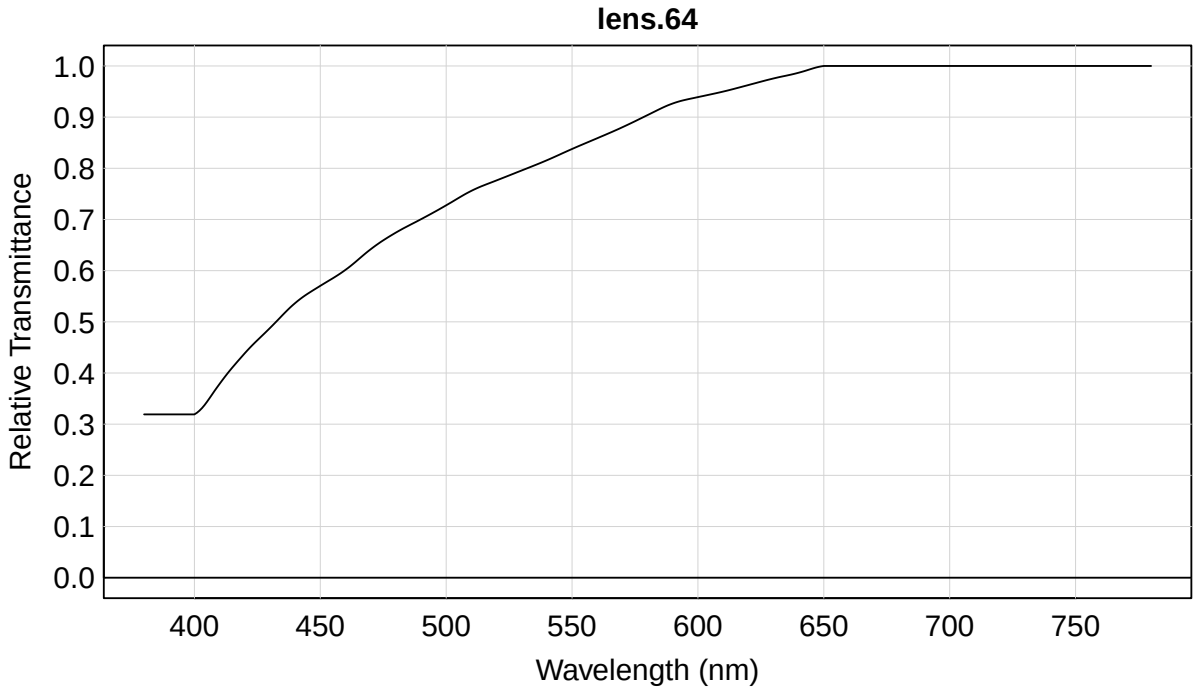

Figure 2: Human Lens Transmittance at age 64 relative to age 32

Think of this curve as defining a pair of glasses with a yellowish tint. In this vignette, going from an age of 32 years to 64 years is equivalent to putting on these tinted glasses.

The Macbeth ColorChecker with Observer Age 32

We first read the spectra of the *ColorChecker* target. This data has been kindly provided in CGATS format by [Pas]. *ColorChecker* is a Registered Trademark of X-Rite, and X-Rite is a Trademark.

```
path = system.file( 'extdata/targets/CC_Avg30_spectrum_CGATS.txt', package='colorSpec')
MacbethCC = readSpectra( path, wave=wavelength(lens.64) )
MacbethCC = MacbethCC[ order(MacbethCC$SAMPLE_ID), ]
print( extradata(MacbethCC), row.names=F )
```

```
 SAMPLE_ID   SAMPLE_NAME        Munsell          ISCC-NBS_Name LEFT TOP WIDTH HEIGHT
         1     dark skin    3YR 3.7/3.2         moderate brown    7   9    29     29
         2    light skin 2.2YR 6.47/4.1    light reddish brown   40   9    29     29
         3      blue sky 4.3PB 4.95/5.5         moderate blue   73   9    29     29
         4       foliage  6.7GY 4.2/4.1   moderate olive green  106   9    29     29
         5   blue flower 9.7PB 5.47/6.7          light violet  139   9    29     29
         6  bluish green      2.5BG 7/6     light bluish green  172   9    29     29
         7        orange       5YR 6/11          strong orange    7  42    29     29
         8 purplish blue   7.5PB 4/10.7   strong purplish blue   40  42    29     29
         9  moderate red      2.5R 5/10          moderate red   73  42    29     29
        10        purple         5P 3/7           deep purple  106  42    29     29
        11  yellow green    5GY 7.1/9.1    strong yellow green  139  42    29     29
        12 orange yellow    10YR 7/10.5   strong orange yellow  172  42    29     29
        13          Blue 7.5PB 2.9/12.7    vivid purplish blue    7  75    29     29
        14         Green 0.25G 5.4/8.65 strong yellowish green   40  75    29     29
        15           Red        5R 4/12             strong red   73  75    29     29
```

```
       16        Yellow      5Y 8/11.1          vivid yellow  106  75    29     29
       17       Magenta     2.5RP 5/12 strong reddish purple  139  75    29     29
       18          Cyan         5B 5/8  strong greenish blue  172  75    29     29
       19         white          N9.5/                 white    7 108    29     29
       20     neutral 8            N8/            light gray   40 108    29     29
       21   neutral 6.5          N6.5/      light medium gray   73 108    29     29
       22     neutral 5            N5/           medium gray  106 108    29     29
       23   neutral 3.5          N3.5/             dark gray  139 108    29     29
       24         black            N2/                 black  172 108    29     29
```

Note that `MacbethCC` is organized as `'df.row'` and contains extra data for each spectrum, notably the coordinates of the patch rectangle.

Now build the "material responder" from Illuminant D65 and the 1931 CMFs:

```
D65.eye = product( D65.1nm, "artwork", xyz1931.1nm, wave=wavelength(lens.64) )
# Calibrate so that when "artwork" is the perfect-reflecting-diffuser, then Y=1,
# and all 3 channels of D65.eye are scaled by the same factor.
# This is the same as the ASTM recommended method, except Y=100 is replaced by Y=1
prd = neutralMaterial( 1, wavelength(lens.64) )
D65.eye = calibrate( D65.eye, stimulus=prd, response=c(NA,1,NA), method='scaling' )
```

Calculate XYZ and then RGB:

```
XYZ = product( MacbethCC, D65.eye, wave=wavelength(lens.64) )
RGB = RGBfromXYZ( XYZ, space='sRGB', which='scene' )$RGB  # this is *signal* sRGB
# add the rectangle data to RGB, so they can be plotted in proper places
patches = extradata(MacbethCC)
patches$RGB = RGB
patches.first = patches   # save this reference object for later
#  display in proper location, and use the sRGB display transfer function
par( omi=c(0,0,0,0), mai=c(0.2,0.2,0.2,0.2) )
plotPatchesRGB( patches, space='sRGB', which='signal', back='gray20', labels=FALSE )
```

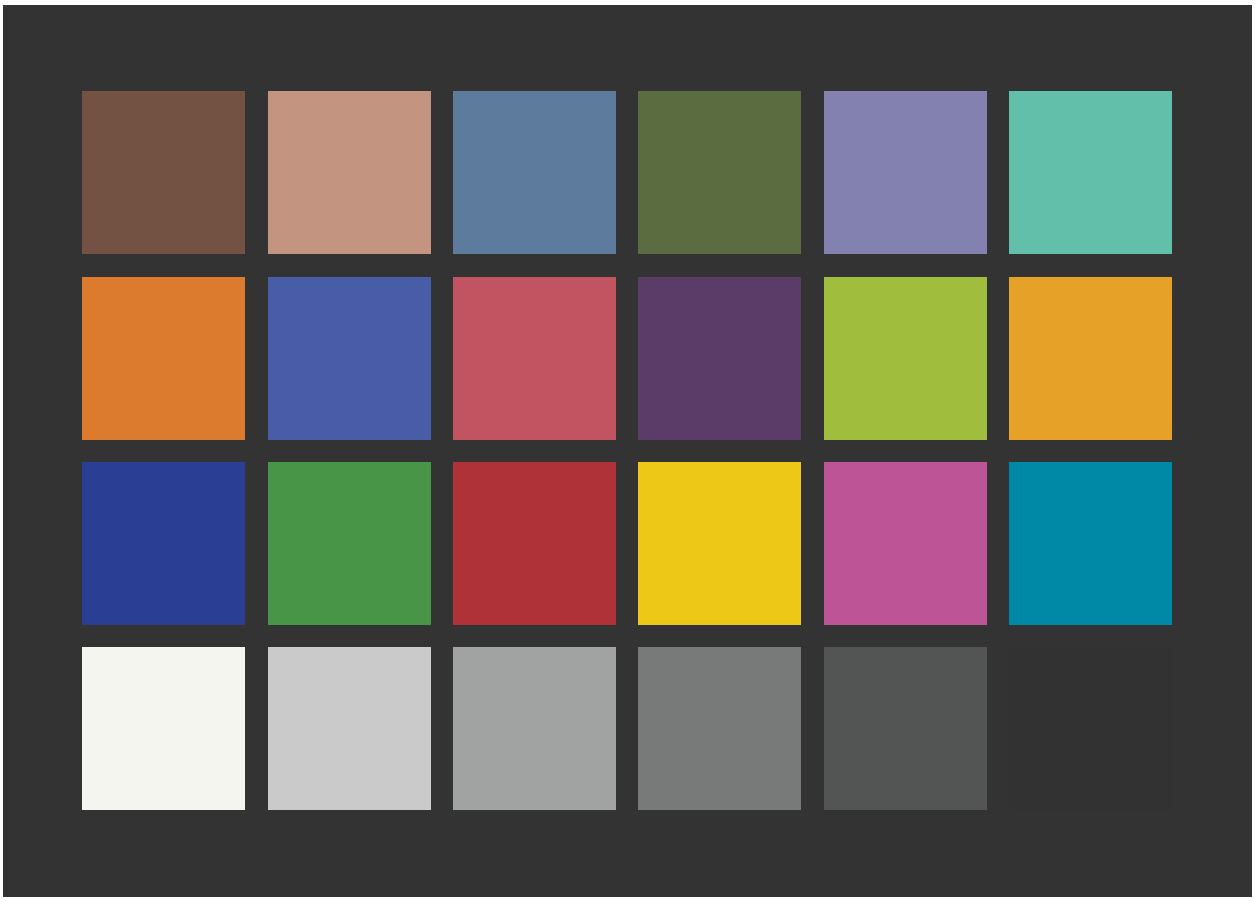

Figure 3: Rendering with Illuminant D65 and xyz1931.1nm, at age=32

This figure has the colors as perceived by the 1931 standard observer.

The Macbeth ColorChecker with Observer Age 64

Make new responder by inserting the hypothetical pair of tinted glasses (defined by `lens.64` in Figure 2) between target and the eye, and then recalculate RGBs.

```
D65.eye.64 = applyspec( D65.eye, function(y) {lens.64 * y} )
XYZ = product( MacbethCC, D65.eye.64, wave=wavelength(lens.64) )
patches = extradata(MacbethCC)
patches$RGB = RGBfromXYZ( XYZ, space='sRGB', which='scene' )$RGB  # this is *signal* sRGB
par( omi=c(0,0,0,0), mai=c(0.2,0.2,0.2,0.2) )
plotPatchesRGB( patches, space='sRGB', which='signal', back='gray20', labels=FALSE )
```

Figure 4: Rendering with Illuminant D65 and xyz1931.1nm, at age=64 without adaptation

As expected, the result has a yellow tint. Now make a plot that compares the effective responsivities.

```
#   the effective responsivities for age=32
par( omi=c(0,0,0,0), mai=c(0.6,0.7,0.3,0.2) )
specnames( D65.eye ) = sprintf( "%s.32", c('x','y','z') )
plot( D65.eye,  lty=1, legend='top' )
#   the effective responsivities for age=64
specnames( D65.eye.64 ) = sprintf( "%s.64", c('x','y','z') )
plot( D65.eye.64,  lty=2, add=TRUE, legend='topright' )
```

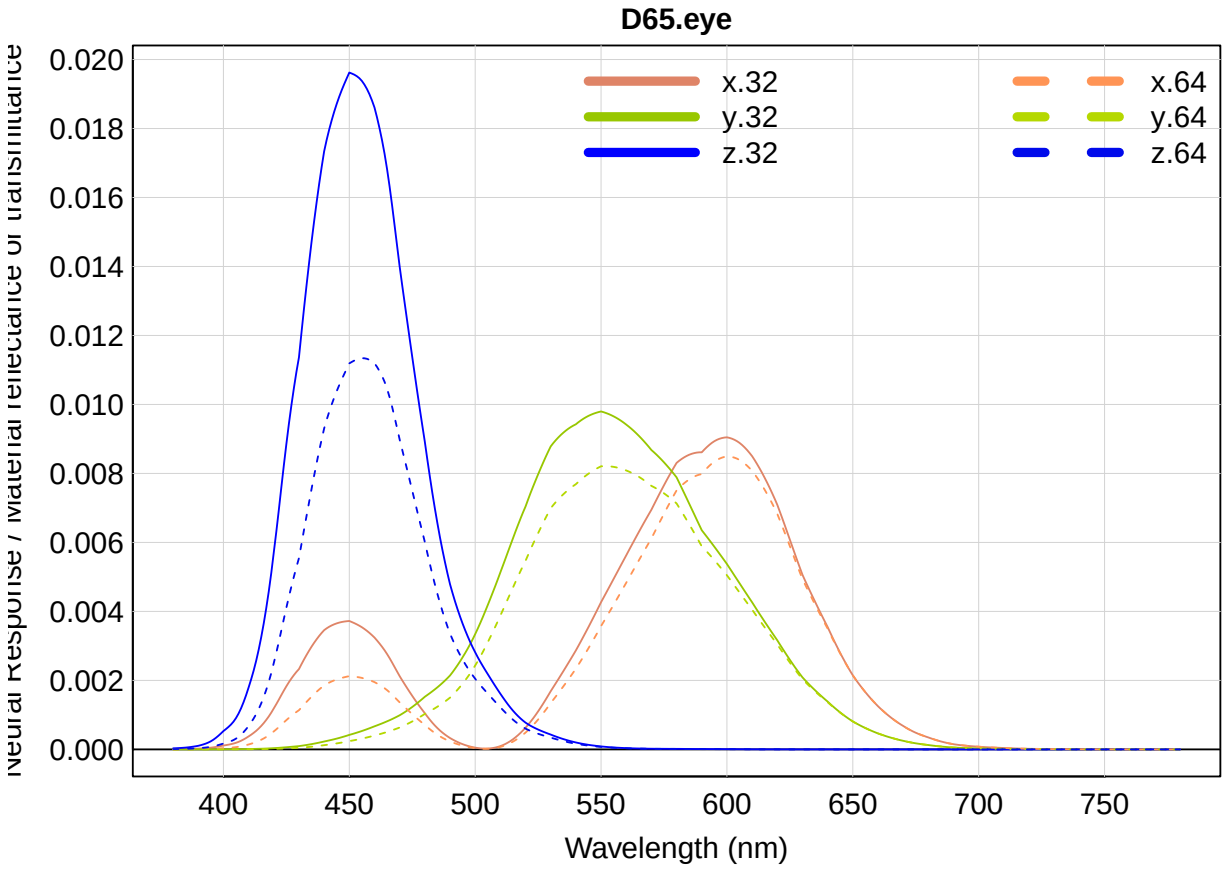

Figure 5: comparison of effective responsivities, at age=32 and age=64

But these figures are only appropriate for the instant in time that the change was made, and before the eye and brain have had the time to adapt. In electronic camera terms, there is no "white balance" yet.

So now calibrate and adapt to D65 using the *Bradford Method*. This method is regarded as being a good model for the way that the human eye and brain achieve *color constancy*, see [Lin].

```
prd = neutralMaterial( 1, wavelength(lens.64) )
XYZ.D65 = spacesXYZ::standardXYZ('D65')
D65.eye.64 = calibrate( D65.eye.64, stimulus=prd, response=XYZ.D65, method='Bradford' )
XYZ = product( MacbethCC, D65.eye.64, wave=wavelength(lens.64) )
patches = extradata(MacbethCC)
patches$RGB = RGBfromXYZ( XYZ, space='sRGB' )$RGB        # this is *signal* sRGB
par( omi=c(0,0,0,0), mai=c(0.2,0.2,0.2,0.2) )
plotPatchesRGB( patches, space='sRGB', which='signal', back='gray20', labels=FALSE )
```

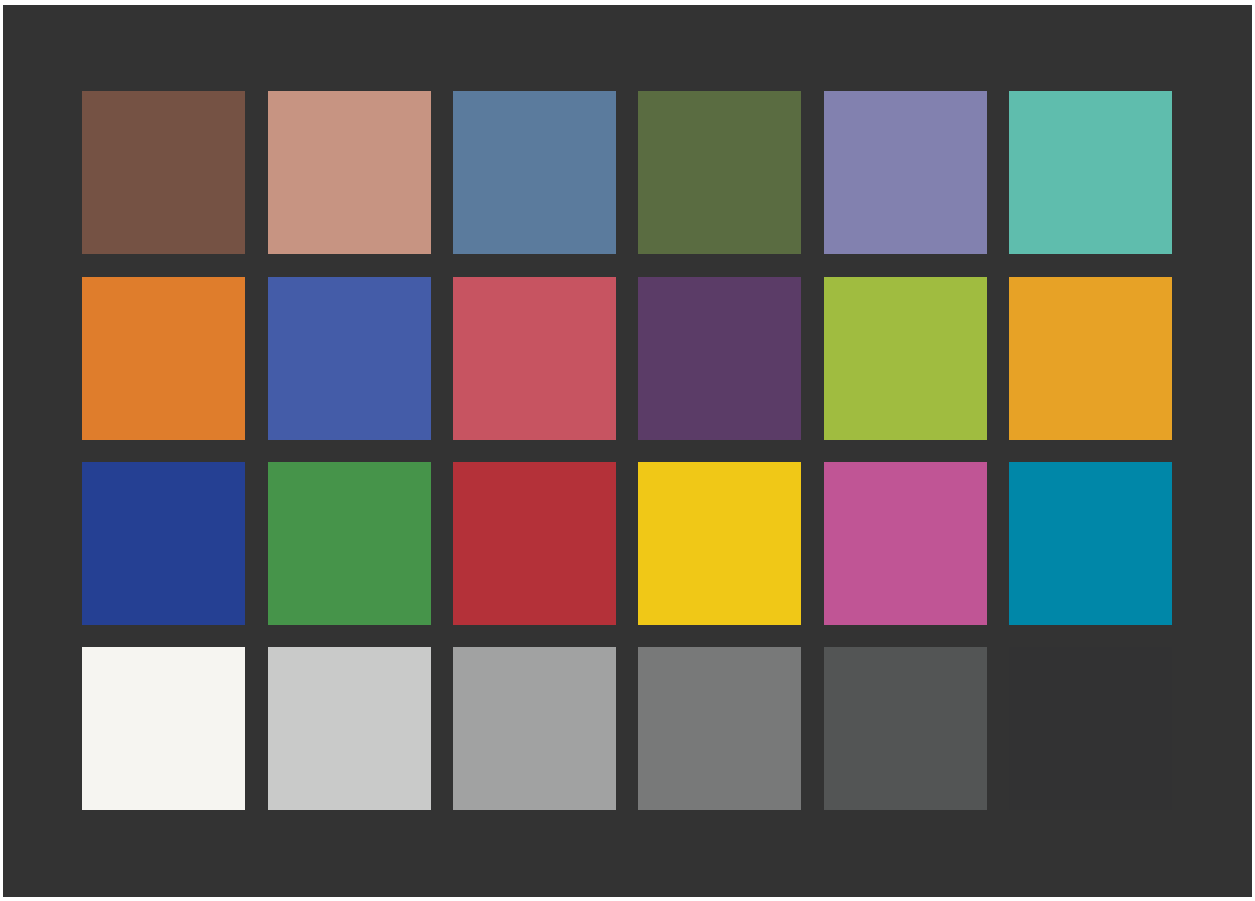

Figure 6: Rendering with Illuminant D65 and xyz1931.1nm, at age=64 after chromatic adaptation

The tint is now gone. But it hard to compare colors in this figure with the ones way back in Figure 3. So combine the original age=32 rendering with the age=64 rendering by splitting each square into 2 triangles.

```
par( omi=c(0,0,0,0), mai=c(0.2,0.2,0.2,0.2) )
#  draw full squares from Figure 3
plotPatchesRGB( patches.first, space='sRGB', back='gray20', labels=F )
#  overwrite the squares with triangles by setting shape= and add=
plotPatchesRGB( patches, space='sRGB', labels=F, shape='bottomright', add=T )
```

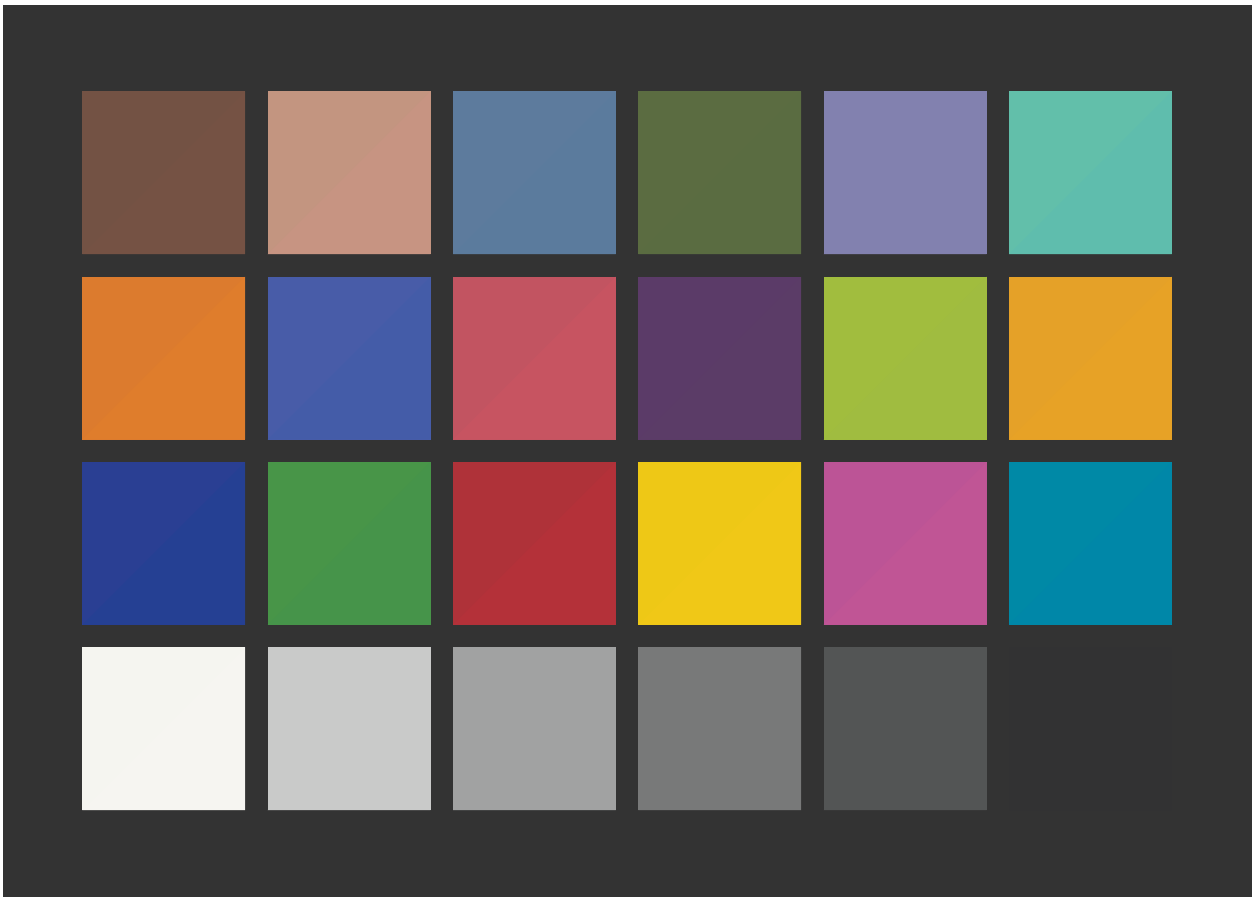

Figure 7: Rendering with both age=32 (Figure 3), and age=64 (Figure 6)

The top-left triangle has the color from Figure 3 and the bottom-right triangle has the color from Figure 6. There are minor differences in the **Red** and **Magenta** patches, and some smaller differences in a few others.

Here are the responsivity functions *after* adaptation:

```
par( omi=c(0,0,0,0), mai=c(0.6,0.7,0.3,0.2) )
plot( D65.eye,  lty=1, legend='top', main=FALSE )
plot( D65.eye.64, lty=2, add=TRUE, legend='topright' )
```

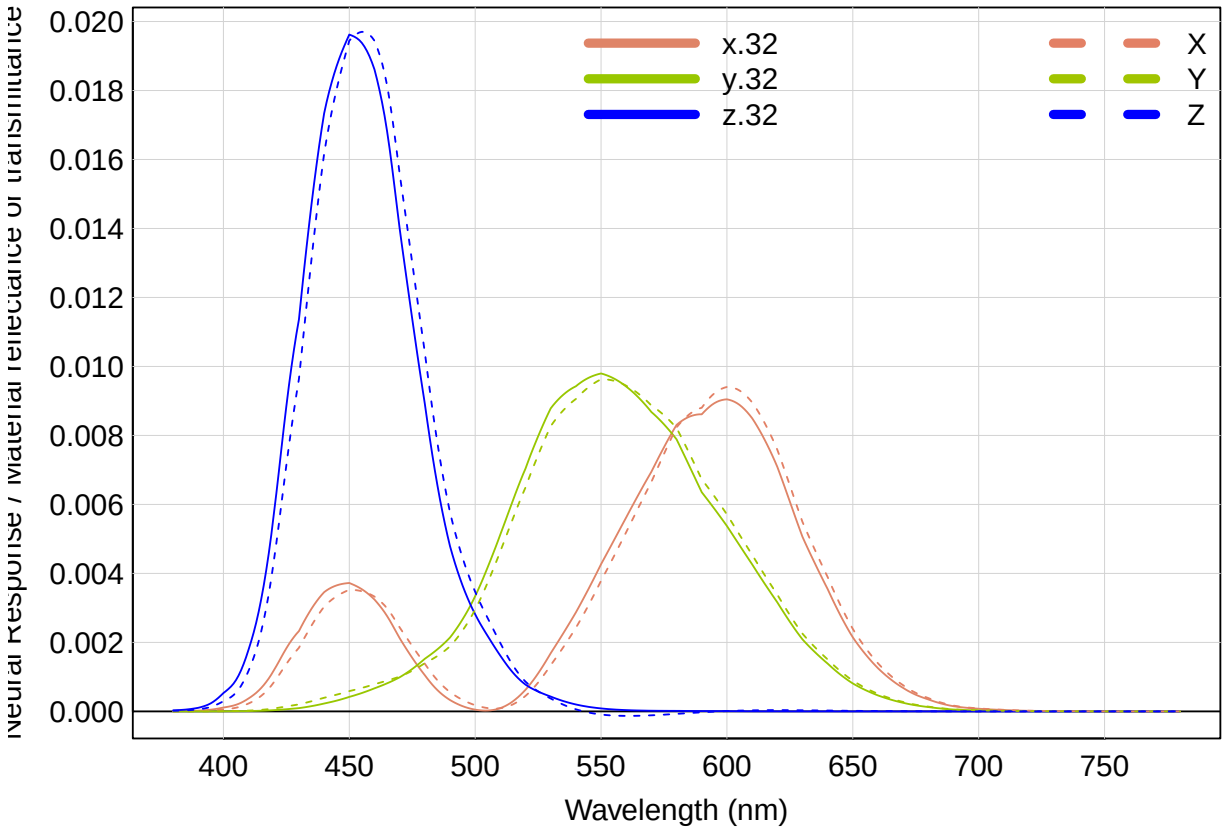

Figure 8: comparison of effective responsivities

So despite the fairly radical yellowing of the lens with age (in Figure 2), this adaptation model shows that the perceived colors are not all that different. Great !

References

[Lin] Lindbloom, Bruce. How the Chromatic Adaptation Calculator Works. `http://brucelindbloom.com/index.html?ChromAdaptCalcHelp.html`.

[Pas] Pascale, Danny. The ColorChecker, page 2. `http://www.babelcolor.com/colorchecker-2.htm`.

[Pok87] Pokorny, Joel, Vivianne C. Smith, and Margaret Lutze. Aging of the Human Lens. *Applied Optics*, Vol. 26, No. 8, 15 April 1987. Table I. Page 1439.

Appendix

This document was prepared February 10, 2025 with the following configuration:

- R version 4.4.2 (2024-10-31 ucrt), `x86_64-w64-mingw32`
- Running under: `Windows 11 x64 (build 26100)`
- Matrix products: default

- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: colorSpec 1.7-0, knitr 1.49, spacesRGB 1.7-0, spacesXYZ 1.5-1
- Loaded via a namespace (and not attached): MASS 7.3-61, R6 2.5.1, bslib 0.8.0, cachem 1.1.0, cli 3.6.3, compiler 4.4.2, digest 0.6.37, evaluate 1.0.1, fastmap 1.2.0, glue 1.8.0, highr 0.11, htmltools 0.5.8.1, jquerylib 0.1.4, jsonlite 1.8.9, lifecycle 1.0.4, logger 0.4.0, microbenchmark 1.5.0, rlang 1.1.4, rmarkdown 2.29, rootSolve 1.8.2.4, sass 0.4.9, tools 4.4.2, xfun 0.49, yaml 2.3.10, zonohedra 0.4-0