Graphics and Visualization

Wolfgang Huber
See also

www.huber.embl.de/msmb Chapter 3
Horror Picture Show
Hematocrit was not validated as a surrogate end point for survival among epoetin-treated hemodialysis patients

Dennis J. Cotter*, Kevin Stefanik, Yi Zhang, Mae Thamer, Daniel Scharfstein, James Kaufman

Medical Technology and Practice Patterns Institute, Inc., 4733 Bethesda Avenue, Suite 510, Bethesda, MD 20814

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD, 21205-2179

VA Boston Healthcare System, Jamaica Plain, MA 02130

Accepted 30 April 2004

Abstract

Objective: To evaluate the use of hematocrit as a surrogate end point for survival among end-stage renal disease (ESRD) patients treated with epoetin.

Study design and setting: Using United States Renal Data System (USRDS) data, we conducted an observational prospective study to analyze the relationships among epoetin dose, hematocrit, and survival for 31,301 facility-based hemodialysis patients incident to ESRD therapy in 1998. To address our objective, we used criteria developed by Prentice based on results from a Cox regression model.

Results: Results indicate that hematocrit is inversely associated with epoetin dose. For the same epoetin treatment-related achieved hematocrit levels, there were widely varying treatment-related survival outcomes, thereby challenging a central criterion required to empirically validate a surrogate end point.

Conclusion: Our results support earlier clinical trial and epidemiological data suggesting that hematocrit may not be a valid surrogate for survival among the epoetin-treated renal failure population. We hypothesize that hematocrit may not be in the causal pathway or that epoetin may have important mechanisms of action apart from increasing hematocrit. Effective treatment for anemia may therefore not be simply a matter of increasing hematocrit. This study has potential implications for revising the existing treatment guidelines for anemia management and selecting an appropriate treatment regimen.

1. Introduction

Anemia is a chronic comorbidity affecting nearly all end-stage renal disease (ESRD) patients and resulting in reduced quality of life and decreased survival rates. In 1987, investigators reported successful use of recombinant human erythropoietin (epoetin, or EPO) in treating the anemia of ESRD patients. By 2000, more than 90% of center hemodialysis patients received epoetin treatment for their anemia. In their review of progress made in the management of anemia, the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) cited a relationship between improved hematocrit levels and mortality, stating that "subsequent studies have shown that increasing patients' hematocrits with EPO decreases mortality and improves quality of life." Given this perceived relationship, the use of hematocrit as a surrogate for guiding epoetin treatment among chronic renal failure patients receiving epoetin is widely accepted among nephrologists. Furthermore, given the difficulty of studying the true clinical outcome (i.e., survival), it is not surprising that the pivotal Phase III epoetin trials to obtain U.S. Food and Drug Administration (FDA) approval implicitly used hematocrit as a surrogate end point.

Although there is a paucity of clinical trials examining the causal effect of epoetin on mortality, the existing data support our contention that hematocrit may not be a valid surrogate for survival among epoetin-treated renal failure patients. In one narrowly focused clinical trial among ESRD patients with cardiovascular complications, the normalized
Hematocrit was not validated as a surrogate end point for survival among epoetin-treated hemodialysis patients

Dennis J. Cotter, Kevin Stefanik, Yi Zhang, Mae Thamer, Daniel Scharfstein, James Kaufman

Medical Technology and Practice Patterns Institute, Inc., 4733 Bethesda Avenue, Suite 510, Bethesda, MD 20814
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD, 21205-2179
VA Boston Healthcare System, Jamaica Plain, MA 02130

Accepted 30 April 2004

Abstract

Objective: To evaluate the use of hematocrit as a surrogate end point for survival among end-stage renal disease (ESRD) patients treated with epoetin.

Study design and setting: Using United States Renal Data System (USRDS) data, we conducted an observational prospective study to analyze the relationships among epoetin dose, hematocrit, and survival for 31,301 facility-based hemodialysis patients incident to ESRD therapy in 1998. To address our objective, we used criteria developed by Prentice based on results from a Cox regression model.

Results: Results indicate that hematocrit is inversely associated with epoetin dose. For the same epoetin treatment-related achieved hematocrit levels, there were widely varying treatment-related survival outcomes, thereby challenging a central criterion required to empirically validate a surrogate end point.

Conclusion: Our results support earlier clinical trial and epidemiological data suggesting that hematocrit may not be a valid surrogate for survival among the epoetin-treated renal failure population. We hypothesize that hematocrit may not be in the causal pathway or that epoetin may have important mechanisms of action apart from increasing hematocrit. Effective treatment for anemia may therefore not be simply a matter of increasing hematocrit. This study has potential implications for revising the existing treatment guidelines for anemia management and selecting an appropriate treatment regimen.

Keywords: End point, surrogate; Outcome assessment; Causal effect; Anemia, management of; Epoetin; Survival; Hematocrit, target

1. Introduction

Anemia is a chronic comorbidity affecting nearly all end-stage renal disease (ESRD) patients and resulting in reduced quality of life and decreased survival rates.

In 1987, investigators reported successful use of recombinant human erythropoietin (epoetin, or EPO) in treating the anemia of ESRD patients. By 2000, more than 90% of in-center hemodialysis patients received epoetin treatment for their anemia (Fig. 40 in [4]). In their review of progress made in the management of anemia, the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) cited a relationship between improved hematocrit levels and mortality, stating that “subsequent studies have shown that increasing patients’ hematocrits with EPO decreases mortality and improves quality of life” [5]. Given this perceived relationship, the use of hematocrit as a surrogate for guiding epoetin treatment among chronic renal failure patients receiving epoetin is widely accepted among nephrologists [3,6–8]. Furthermore, given the difficulty of studying the true clinical outcome (i.e., survival), it is not surprising that the pivotal Phase III epoetin trials to obtain U.S. Food and Drug Administration (FDA) approval implicitly used hematocrit as a surrogate end point [9].

Although there is a paucity of clinical trials examining the causal effect of epoetin on mortality, the existing data support our contention that hematocrit may not be a valid surrogate for survival among epoetin-treated renal failure patients. In one narrowly focused clinical trial among ESRD patients with cardiovascular complications, the normalized
Why graphics?

1. To explore data (interactively)
2. To communicate data & preliminary insights with collaborators
3. To publish results
Goals for this lecture

• Review base R plotting
• Understand the grammar of graphics concept
• Introduce ggplot2's ggplot function
• See how to plot 1D, 2D, 3-5D data and understand faceting
• Visualisation for quickling viewing large datasets and discover large-scale trends (e.g. batch effects)
• Use colours like a pro
Canvas model: a series of instructions that sequentially fill the plotting canvas

```r
head(DNase)
##   Run conc density
## 1  1    0.0488 0.017
## 2  1    0.0488 0.018
## 3  1    0.1953 0.121
## 4  1    0.1953 0.124
## 5  1    0.3906 0.206
## 6  1    0.3906 0.215
```

```r
plot(DNase$conc, DNase$density, ylab = attr(DNase, "labels")$y, xlab = paste(attr(DNase, "labels")$x, attr(DNase, "units")$x), pch = 3, col = "blue")
```
In Figure 4.3: Same data as in Figure 4.2, consider the following code snippet:

```r
head(DNase)
##  Run  conc  density
## 1     1  0.04  0.0488
## 2     1  0.04  0.0488
## 3     1  0.19  0.1953
## 4     1  0.19  0.1953
## 5     1  0.39  0.3906
## 6     1  0.39  0.3906
```

And the plot function:

```r
plot(DNase$conc, DNase$density, xlab = "DNase concentration (ng/ml)", ylab = "Optical density", pch = 3, col = "blue")
```

This code creates a plot showing the relationship between DNase concentration and optical density. The plot is used to visualize data from an enzyme-linked immunosorbent assay (ELISA) assay of DNase. For an ELISA assay of DNase, the measured optical density is related to the concentration of the enzyme, and such plots are crucial in identifying enzymes and quantifying their activities.
Drawbacks:

- Layout choices have to be made at the beginning with no overview over what may still be coming
- Different functions for different plot types, with different interfaces
- Many routine tasks require a lot of ‘boilerplate’ code
- No concept of facets / lattices
- No concept of viewports, only a single global coordinate system
- Default colours are poor
- Resizing often leads to unsatisfactory results
The grammar of graphics

The components of ggplot2's grammar of graphics are

- one or more datasets ("noun"),
- one or more geometric objects that serve as the visual representations of the data, for instance, points, lines, rectangles, contours ("verb"),
- descriptions of how the variables in the data are mapped to visual properties (aesthetics) of the geometric objects, and an associated scale (e.g. linear, logarithmic, rank),
- one or more coordinate systems,
- statistical summarization rules (e.g. line fit, binning),
- a facet specification, i.e. multiple similar subplots to look at subsets of the same data,
- optional parameters for layout and rendering, e.g., text size, font, alignment; legend positions

```r
ggplot(groups, aes(x = sampleGroup, y = n, fill = sampleGroup)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = groupColour, name = "Groups") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```
A Layered Grammar of Graphics
Hadley Wickham
Journal of Computational and Graphical Statistics, 2010
Volume 19, Number 1, Pages 3–28
DOI: 10.1198/jcgs.2009.07098
Layers

```r
ggplot( dftx, aes( x = X1426642_at, y = X1418765_at ) ) +
  geom_point( aes( colour = sampleColour), shape = 19 ) +
  geom_smooth( method = "loess" ) +
  scale_colour_discrete( guide = FALSE )
```
A more complex example: themes

```r
pb <- ggplot(data.frame(
    name = names(groupSize),
    size = as.vector(groupSize)),
    aes(x = name, y = size))
```

Let's come back to the barplot example from above and see how it is done in the `ggplot` way.

```r
pb <- ggplot(data.frame(name = names(groupSize), size = as.vector(groupSize)),
             aes(x = name, y = size))
```

For now we have simply created a plot object `pb` and have not generated a plot yet. In fact we cannot make a plot yet, because we haven't specified what geometric object we want to use for our plot. All that we have in our `pb` object so far are the data and the aesthetics.

Now we can literally add on the other components of our plot through using the `+` operator:

```r
pb <- pb + geom_bar(stat = "identity") +
     aes(fill = name) +
     scale_fill_manual(values = group Colour, name = "Colour code") +
     theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
     xlab("Groups") +
     ylab("Number of Samples")
```

Thus we recreate our previous `qplot` result using the `ggplot` approach. This modular approach allows us a lot of freedom in creating figures and setting parameters. For example we can switch our plot to polar coordinates to create a popular alternative visualization of the barplot.

```r
pb.polar <- pb + coord_polar() +
            theme(axis.text.x = element_text(angle = 0, hjust = 1),
                  axis.text.y = element_blank(),
                  axis.ticks = element_blank()) +
            xlab("") +
            ylab("")
```

Note above that we can override previously set `theme` parameters by simply resetting them – no need to go back to recreating `pb`, where we
A more complex example: themes

```r
pb <- ggplot(data.frame(
    name = names(groupSize),
    size = as.vector(groupSize)),
    aes(x = name, y = size))
```

No geom defined yet!
A more complex example: themes

```r
pb <- ggplot(data.frame(
    name = names(groupSize),
    size = as.vector(groupSize)),
    aes(x = name, y = size))

No geom defined yet!

pb <- pb + geom_bar(stat = "identity") +
    aes(fill = name) +
    scale_fill_manual(values = groupColour, name = "Colour code") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    xlab("Groups") + ylab("Number of Samples")
```

Now we can literally add on the other components of our plot through using `grid.arrange` because we haven't specified what geometric object we want to use for our barplot. Let's come back to the barplot example from above and see how it is done. Note above that we can override previously set parameters by simply resetting them – no need to go back to recreating the plot yet. In fact we cannot make a plot yet, but we cannot make a plot yet, in the `grid.arrange` library we have not generated a result using the `angle` to create our `name` and `pb` so far are the data and the aesthetics.
A more complex example: themes

```
pb <- ggplot(data.frame(
    name = names(groupSize),
    size = as.vector(groupSize)),
    aes(x = name, y = size))

No geom defined yet!

pb <- pb + geom_bar(stat = "identity") +
    aes(fill = name) +
    scale_fill_manual(values = groupColour, name = "Colour code") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    xlab("Groups") + ylab("Number of Samples")

pb.polar <- pb + coord_polar() +
    theme(axis.text.x = element_text(angle = 0, hjust = 1),
    axis.text.y = element_blank(),
    axis.ticks = element_blank()) +
    xlab("") + ylab("")

pb.polar
```
Visualizing distributions in 1D

- **Barplots**: Used for displaying the mean value of genes. For example, `ggplot(data,...)`.
- **Boxplots**: Display the distribution of values for genes, such as `ggplot(data,...)`.
- **Violin plots**: Show the distribution density without overlap, using `ggplot(data,...)`.
- **Scatter plots with error bars**: Display individual values with error bars, using `ggplot(data,...)`.
- **Dot plots and beeswarm plots**: Used for high-quality graphics, with `ggplot(data,...)`.

**Key Points**:
- The need for choosing a smoothing window.
- Better contrast in plots.
- Density lines convey less information on data amount.
- The plot is shown in the right panel of Figure 4.19.
1D plot types

**Boxplot** makes sense for unimodal distributions

**Histogram** requires definition of bins (width, positions) and can create visual artifacts esp. if the number of data points is not large

**Density** requires the choice of bandwidth; obscures the sample size (i.e. the uncertainty of the estimate)

**ecdf** does not have these problems; but is more abstract and interpretation requires more training. Good for reading off quantiles and shifts in location in comparative plots; OK for detecting differences in scale; less good for detecting multimodality.

Up to a few dozens of points - just show the data! ([beeswarm](#))
The empirical cumulative distribution function

\[ F_n(x) = \frac{\text{number of } i \text{ for which } x_i \leq x}{n} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x \leq x_i). \]

`simdata = rnorm(70)`
`tibble(index = seq(along = simdata),
       sx = sort(simdata)) %>%
ggplot(aes(x = sx, y = index)) + geom_step()"
LETTER

doi:10.1038/nature12213

Mutational heterogeneity in cancer and the search for new cancer-associated genes

Michael S. Lawrence1,2, Petar Stojanov1,2, Paz Palou3,4, Gregory V. Kryukov1,3, Kristian Olubolde1, Andrey Stvachenko1,2, Scott L. Carter1, Chip Stewart1, Craig H. Mermel2,3, Steven A. Roberts1, Adam Kiezun1, Peter S. Hammerman1, Aaron McKenna1, Yoram Drier2,3,4, Lihua Zou1, Alex H. Ramos1, Trevor J. Pugh1,2, Nicolas Strayansky1,2, Elena Hiehlman3,4, Jeege Kim1, Carrie Sougnez1, Lauren Ambrogio1, Elizabeth Nickerson1, Erin Sheffer1, Maria L. Cortes1, Daniel Aucouturier1, Gordon Sakosits1, Douglas Voel1, Michael Noble1, Daniel DeCar1, Pei Lin1, Lee Lichtenstein1, David I. Heiman1, Timothy Fennell1, Marcia Irwin1,2,3, Brian Hernandez1, Eran Hodith1,2, Sylvestre Bacc1,2, Austin M. Dukic1,2,4,4, Jens Lohr1,2, Dan-Avi Landa2,3,4, Catherine E. Wu2,3,4, Jorge Melendez-Zajil1,2, Alfredo Hidalgo-Miranda1,2, Amanon Koren1,2, Steven A. McCarroll3,4, Faustine Mora1, Ryan S. Lee1,2,3,4, Brian Crompton1,3, Robert Onofrio1, Melissa Parkin1, Wendy Winckler1, Kristin Ardlie1, Stacey R. Gabrielson1, Charles W. M. Roberts1,2,3,4, Jaclyn A. Ringlef1, Kimberly Stegmaier1,2,2, Adam J. Rust1,2,4,4, Levi A. Garraway1,2,2, Michael Meyerson1,2,3,4, Todd R. Golub1,2,3,4, Dmitry A. Gordenin1,2,3,4, Shamil Sunyaev1,2,4,4, Eric S. Lander1,2,4,5, & Gad Getz1,5
Impact of non-linear transformation on the shape of a density

- The mode of a distribution is an infinitesimal concept.
- Need either an infinite amount of data or choose smoothing / binning bandwidth
- Number of modes (let alone their positions) can change under non-linear data transformations (Question 3.5 in the book)
Let us take a look at differential expression between a wildtype and an FGF4-KO mutant. Two different datasets are now available: one containing 59 E4.5 (PE) measurements for one of the samples, and the other containing 92 E4.5 (FGF4-KO) measurements for the other sample.

```r
scp <- ggplot(dfx, aes(x = '59 E4.5 (PE)', y = '92 E4.5 (FGF4-KO)'))
scp + geom_point()
```

This is already better than Figure 4.21: Scatterplot of 45101 expression values.

```r
scp + geom_point(alpha = 0.1)
```

For even better results, you can use a contour plot of the 2D density estimate.

```r
scp + geom_density2d(h = 0.5, bins = 60)
```

A potential drawback is efficiency: even though there are only 4 probe–gene symbol pairs, we now have some columns that represent different measurements for two of the samples. The plot is shown in Figure 4.22: As Figure 4.7 2D visualisation: Scatter Plots in Bioconductor, such as the hexbin package.

In Figure 4.23, parameters of density2d are more visually appealing – the argument `fixed` is used to fix the aspect ratio, and the `coord` parameter of density2d is set to `gradientn`. The plot is shown in Figure 4.24: Scatter plots are useful for visualizing treatment–response comparisons (as in Figure 4.25).

```r
scp <- ggplot(dfx, aes(x = '59 E4.5 (PE)', y = '92 E4.5 (FGF4-KO)'))
scp + geom_point()
```

This is already better than Figure 4.21: Scatterplot of 45101 expression values.

```r
scp + geom_point(alpha = 0.1)
```

For even better results, you can use a contour plot of the 2D density estimate.

```r
scp + geom_density2d(h = 0.5, bins = 60)
```

A potential drawback is efficiency: even though there are only 4 probe–gene symbol pairs, we now have some columns that represent different measurements for two of the samples. The plot is shown in Figure 4.22: As Figure 4.7 2D visualisation: Scatter Plots in Bioconductor, such as the hexbin package.

In Figure 4.23, parameters of density2d are more visually appealing – the argument `fixed` is used to fix the aspect ratio, and the `coord` parameter of density2d is set to `gradientn`. The plot is shown in Figure 4.24: Scatter plots are useful for visualizing treatment–response comparisons (as in Figure 4.25).

```r
scp <- ggplot(dfx, aes(x = '59 E4.5 (PE)', y = '92 E4.5 (FGF4-KO)'))
scp + geom_point()
```

This is already better than Figure 4.21: Scatterplot of 45101 expression values.

```r
scp + geom_point(alpha = 0.1)
```

For even better results, you can use a contour plot of the 2D density estimate.

```r
scp + geom_density2d(h = 0.5, bins = 60)
```
Choosing the proper shape for your plot is important to make sure the information is conveyed well. By default, the shape parameter, that is, the ratio, between the height of the graph and its width, is chosen by `ggplot2` based on the available space in the current plotting device. The width and height of the device are specified when it is opened in R, either explicitly by you or through default parameters. Moreover, the graph dimensions also depend on the presence or absence of additional decorations, like the colour scale bars in Figure ??.

```
scp + stat_binhex(binwidth = c(0.2, 0.2)) + colourscale + coord_fixed()
```
Plot shape, banking

Yearly sunspot numbers
1849-1924

Changes in amplitude

Banking to 45 degrees:

Choose aspect ratio so that center of absolute values of slopes is 45 degrees

Sawtooth: Sunspot cycles typically rise more rapidly than they fall (pronounced for high peaks, less for medium and not for lowest)
Plot shape, banking

Yearly sunspot numbers 1849-1924
Changes in amplitude
Banking to 45 degrees:
Choose aspect ratio so that center of absolute values of slopes is 45 degrees

Sawtooth: Sunspot cycles typically rise more rapidly than they fall (pronounced for high peaks, less for medium and not for lowest)

For plots where x- and y-axis have same units: use 1:1 aspect ratio (PCA!)
3-5 D and faceting

geom_point offers these aesthetics (beyond x and y):

- fill
- colour
- shape
- size
- alpha

```r
ggplot(dftx, aes( x = X1426642_at, y = X1418765_at)) + geom_point() + facet_grid( . ~ lineage )
```

```r
ggplot( dftx,
aes( x = X1426642_at, y = X1418765_at)) + geom_point() + facet_grid( Embryonic.day ~ lineage )
```
Data from an agricultural field trial to study the crop barley.

At 6 sites in Minnesota, 10 varieties of barley were grown in each of two years.

Data: yield, for all combinations of site, variety, and year (6 x 10 x 2 = 120 observations)

Note the data for Morris - reanalysis in the 1990s using Trellis revealed that the years had been flipped!
EDA for finding batch effects
TopGene <- order(rowVars(exprs(x)), decreasing = TRUE)[seq_len(500)]

rowCenter <- function(x) {x - rowMeans(x) }

pheatmap(rowCenter(exprs(x)[TopGene,]), show_rownames = FALSE, show_colnames = FALSE, 
  breaks = seq(-5, +5, length = 101),
  annotation_col = pData(x)[,c("sampleGroup", "Embryonic.day", "ScanDate")],
  annotation_colors = list(sampleGroup = groupColour, genotype = c("FGF4-KO" = "chocolate1", "WT" = "azure2"), Embryonic.day = setNames(brewer.pal(9, "Blues")[[c(3, 6, 9)]], c("E3.25", "E3.5", "E4.5")), ScanDate = setNames(brewer.pal(nlevels(x$ScanDate), "YlGn"), levels(x$ScanDate)), cutree_rows = 4)

Figure 4.33: A heatmap of relative expression values, i.e., $\log_2$ fold change compared to the average expression of that gene (row) across all samples (columns). The colour scale uses a diverging palette, whose neutral midpoint is at 0.

Let us take a minute to deconstruct the rather massive-looking call to pheatmap. The options show_rownames and show_colnames control whether the row and column names are printed at the sides. Because our matrix is large in relation to the available plotting space, the labels would anyway not be readable, and we suppress them. The annotation_col allows easy addition of column and row metadata. See also ComplexHeatmaps package.
pheatmap

Since we have many reasonable defaults, it is easy to add column and row ‘metadata’ at the sides.

See also ComplexHeatmaps package.
Interactivity

Use shiny or plotly
https://shiny.rstudio.com/gallery/genome-browser.html

Animations (time-dependent plots):
https://gganimate.com

Linked Charts
https://anders-biostat.github.io/linked-charts/

NB: ggvis is senescent
pie(rep(1, 8), col=1:8)

display.brewer.all()
Consider these:
Different requirements for line & area colours
Many people are red-green colour blind
Lighter colours tend to make areas look larger than darker colours → use colors of equal luminance for filled areas.
RGB color space
Motivated by computer screen hardware
HSV color space

Hue-Saturation-Value (Smith 1978)

$V_{\text{max}}$: a planar area of fully saturated colours, with white in the centre

hue: similarity to a primary color

saturation: width of the spectral distribution

$V_{\text{min}}$: black (one point)
HSV color space

GIMP colour selector

linear or circular hue chooser
and a two-dimensional area (usually a square or a triangle) to choose saturation and value/lightness for the selected hue
Conversion from RGB to HSL or HSV

Let \( r, g, b \in [0, 1] \) be the red, green, and blue coordinates, respectively, of a color in RGB space.

Let max be the greatest of \( r, g, \) and \( b, \) and min the least.

To find the hue angle \( h \in [0, 360] \) for either HSL or HSV space, compute:

\[
\begin{align*}
    h &= \begin{cases} 
        0 & \text{ if max = min} \\
        (60^\circ \times \frac{g-b}{\text{max} - \text{min}} + 360^\circ \text{ mod } 360^\circ) & \text{ if max = } r \\
        60^\circ \times \frac{r-g}{\text{max} - \text{min}} + 120^\circ & \text{ if max = } g \\
        60^\circ \times \frac{r-b}{\text{max} - \text{min}} + 240^\circ & \text{ if max = } b 
    \end{cases}
\end{align*}
\]

To find saturation and lightness \( s, l \in [0, 1] \) for HSL space, compute:

\[
\begin{align*}
    s &= \begin{cases} 
        0 & \text{ if max = min} \\
        \frac{\text{max} - \text{min}}{\text{max} + \text{min}} & \text{ if } l \leq \frac{1}{2} \\
        \frac{\text{max} - \text{min}}{2(\text{max} + \text{min})} & \text{ if } l > \frac{1}{2}
    \end{cases}
\end{align*}
\]

\[
l = \frac{1}{2}(\text{max} + \text{min})
\]

The value of \( h \) is generally normalized to lie between 0 and 360°, and \( h = 0 \) is used when \( \text{max} = \text{min} \) (that is, for grays) though the hue has no geometric meaning there, where the saturation \( s \) is zero. Similarly, the choice of 0 as the value for \( s \) when \( l \) is equal to 0 or 1 is arbitrary.

HSL and HSV have the same definition of hue, but the other components differ. The values for \( s \) and \( v \) of an HSV color are defined as follows:

\[
\begin{align*}
    s &= \begin{cases} 
        0, & \text{ if max = 0} \\
        \min - \min \frac{\text{max}}{\text{max}} & \text{ otherwise}
    \end{cases}
\end{align*}
\]

\[
v = \text{max}
\]

The range of HSV and HSL vectors is a cube in the cartesian coordinate system; but since hue is really a cyclic property, with a cut at red, visualizations of these spaces invariably involve hue circles;[4] cylindrical and conical (bi-conical for HSL) depictions are most popular; Spherical depictions are also possible.
Perceptual colour spaces

Human perception of colour corresponds neither to RGB nor HSV coordinates, and neither to the physiological axes light-dark, yellow-blue, red-green
Perceptually based coordinates of colour space: CIELUV, CIELAB

Commission Internationale de l’Éclairage (CIE) in 1931, on the basis of extensive colour matching experiments with people, defined a “standard observer” who represents a typical human colour response (response of the three light cones + their processing in the brain) to a triplet (x,y,z) of primary light sources.

https://en.wikipedia.org/wiki/CIE_1931_color_space

1976: CIELUV (L, u, v)-coordinates are preferred by those who work with emissive colour technologies (e.g. computer displays); CIELAB by those working with dyes and pigments (such as in the printing and textile industries)

Ihaka 2003
HCL colours

(u,v) = C * (cos H, sin H)

H: hue (dominant wavelength)
C: chroma (colorfulness, intensity of color as compared to gray)
L: luminance (brightness, amount of gray), same as in CIELUV

(C, H) are simply polar coordinates for (u,v)
Figure 2: Circles in HCL colorspace.  

a: circles in HCL space at constant $L = 75$, with the angular coordinate $H$ varying from 0 to 360 and the radial coordinate $C = 0, 10, \ldots, 60$.  
b: constant $C = 50$, and $L = 10, 20, \ldots, 90$.  

Pick your favourite

From A. Zeileis, Reisensburg 2007
Balance

The intensity of colour that should be used is dependent on the area that that colour is to occupy. Small areas need to be more colourful than larger ones.

Choose colours centred on a mid-range or neutral value, or;
Choose colours at equally spaced points along smooth paths through (perceptually uniform) colour space: equal luminance and chroma and correspond to set of evenly spaced hues.
Acknowledgements

Susan Holmes
Robert Gentleman
Florian Hahne

Hadley Wickham
Ross Ihaka
Achim Zeileis
Kurt Hornik