Fitting Nonlinear Mixed Effects Models

Richard A. Bilonick, PhD

R. A. Bilonick Statistical Consultancy

237th Meeting of the American Astronomical Society

rabilonick@gmail.com

January 7, 2021

Fitting Nonlinear Mixed Effects Models

Richard A. Bilonick, PhD

R. A. Bilonick Statistical Consultancy

237th Meeting of the American Astronomical Society

rabilonick@gmail.com

January 7, 2021

Empirical Modeling

Fitting straight lines or other polynomial regressions are usually examples of empirical modeling.

Many relationships are approximately linear only over a narrow range.

Relationships over broader ranges typically are nonlinear.

Finally, these types of regression models do not explain the process and the coefficients are often not particularly meaningful.

Empirical Modeling

Fitting straight lines or other polynomial regressions are usually examples of empirical modeling.

Many relationships are approximately linear only over a narrow range.

Relationships over broader ranges typically are nonlinear.

Finally, these types of regression models do not explain the process and the coefficients are often not particularly meaningful.

1. Some nonlinear models can model the process that generates the data, rather than just describin the pattern.
2. For the data we will be examining, we are still just describing the patterns.

Nonlinear Regression

Complex patterns can often be accommodated by using various types of spline functions. Spline functions have a set of basis functions whose corresponding coefficients describe the curve.

But, as with polynomials, these coefficients are purely descriptive.

If you know the functional nonlinear relationship, you can use nonlinear regression to fit a mechanistic model to capture the process.

Nonlinear regression requires an iterative approach to estimating the model parameters. Unlike ordinary regression which has a closed form solution, starting values are required.

It's important to ensure that the iterative process has converged.

Nonlinear Regression

Complex patterns can often be accommodated by using various types of spline functions. Spline functions have a set of basis functions whose corresponding coefficients describe the curve.

But, as with polynomials, these coefficients are purely descriptive.

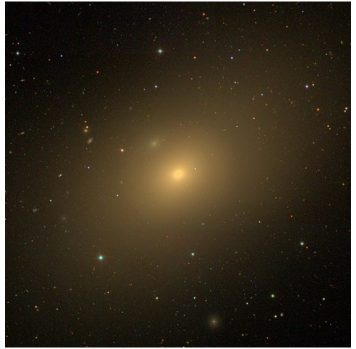
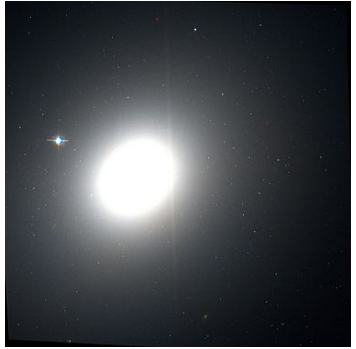
If you know the functional nonlinear relationship, you can use nonlinear regression to fit a mechanistic model to capture the process.

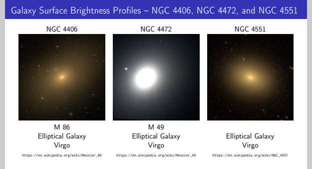
Nonlinear regression requires an iterative approach to estimating the model parameters. Unlike ordinary regression which has a closed form solution, starting values are required.

It's important to ensure that the iterative process has converged.

1.
2.
3.

Galaxy Surface Brightness Profiles – NGC 4406, NGC 4472, and NGC 4551

NGC 4406	NGC 4472	NGC 4551
		
M 86 Elliptical Galaxy Virgo	M 49 Elliptical Galaxy Virgo	Elliptical Galaxy Virgo
https://en.wikipedia.org/wiki/Messier_86	https://en.wikipedia.org/wiki/Messier_49	https://en.wikipedia.org/wiki/NGC_4551



1. Star surface brightness was measured for each of these three galaxies.
2. Two of the galaxies are in both the Messier catalog and the NGC.

Galaxy Surface Brightness Profiles

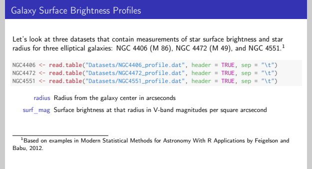
Let's look at three datasets that contain measurements of star surface brightness and star radius for three elliptical galaxies: NGC 4406 (M 86), NGC 4472 (M 49), and NGC 4551.[1]

```
NGC4406 <- read.table("Datasets/NGC4406_profile.dat", header = TRUE, sep = "\t")
NGC4472 <- read.table("Datasets/NGC4472_profile.dat", header = TRUE, sep = "\t")
NGC4551 <- read.table("Datasets/NGC4551_profile.dat", header = TRUE, sep = "\t")
```

radius Radius from the galaxy center in arcseconds

surf_mag Surface brightness at that radius in V-band magnitudes per square arcsecond

[1] Based on examples in Modern Statistical Methods for Astronomy With R Applications by Feigelson and Babu, 2012.



1. All three datasets have the same variables.
2. These three datasets will be merged later into one dataset.

Galaxy Surface Brightness Profiles

```r
head(NGC4406)
```

```
##   radius surf_mag
## 1  1.384   16.323
## 2  1.501   16.401
## 3  1.656   16.494
## 4  1.793   16.578
## 5  2.030   16.715
## 6  2.178   16.796
```

```r
head(NGC4472)
```

```
##   radius surf_mag
## 1  3.526   16.917
## 2  4.015   17.032
## 3  4.557   17.150
## 4  4.966   17.239
## 5  5.491   17.339
## 6  6.039   17.439
```

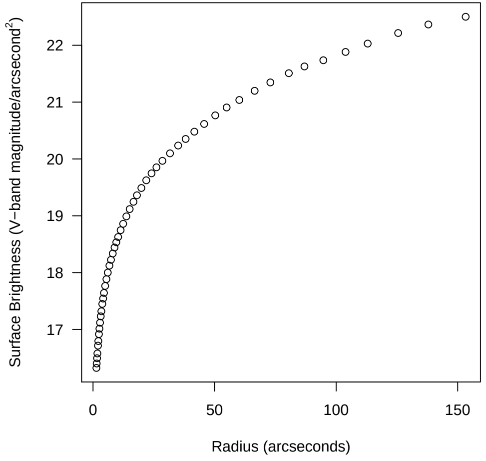
```r
head(NGC4551)
```

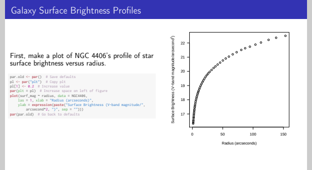
```
##   radius surf_mag
## 1  2.780   18.428
## 2  3.208   18.566
## 3  3.443   18.643
## 4  3.801   18.733
## 5  4.276   18.881
## 6  4.595   18.954
```


1. The first six records of each `data.frame` are displayed.

First, make a plot of NGC 4406's profile of star surface brightness versus radius.

```
par.old <- par()  # Save defaults
pl <- par("plt")  # Copy plt
pl[1] <- 0.2  # Increase value
par(plt = pl)  # Increase space on left of figure
plot(surf_mag ~ radius, data = NGC4406,
    las = 1, xlab = "Radius (arcseconds)",
    ylab = expression(paste("Surface Brightness (V-band magnitude/",
        arcsecond^2, ")", sep = "")))
par(par.old)  # Go back to defaults
```





1. Here is a plot of star surface brightness as a function of radius for NGC 4406.

Galaxy Surface Brightness Profiles – Linear Model (Quadratic)

```r
fit.1 <- lm(surf_mag ~ poly(radius, 2), data = NGC4406)
summary(fit.1)
```

```r
par.old <- par()  # Save defaults
pl <- par("plt")  # Copy plt
pl[1] <- 0.2  # Increase value
par(plt = pl)  # Increase space on left of figure
plot(surf_mag ~ radius, data = NGC4406, las = 1, xlab = "Radius (arcseconds)",
    ylab = expression(paste("Surface Brightness (V-band magnitude/",
        arcsecond^2, ")", sep = "")))
pred <- predict(fit.1, newdata = data.frame(radius = seq(0, 150,
    1)))
with(NGC4406, lines(seq(0, 150, 1), pred, col = "red"))
par(par.old)  # Go back to defaults\t
```


Galaxy Surface Brightness Profiles – Linear Model (Quadratic)

1.
2.
3.

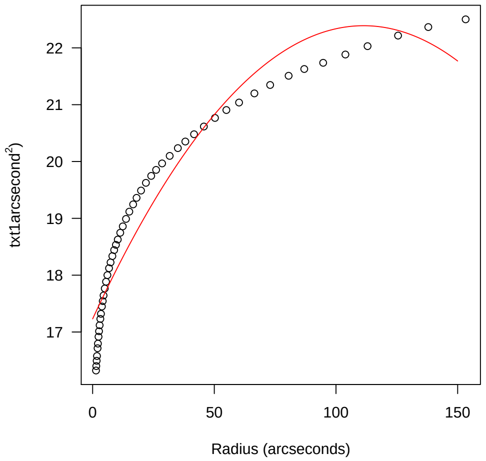
Galaxy Surface Brightness Profiles – Linear Model (Quadratic)

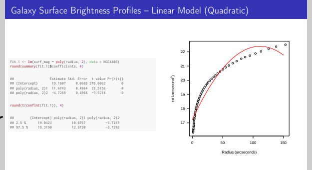
```r
fit.1 <- lm(surf_mag ~ poly(radius, 2), data = NGC4406)
round(summary(fit.1)$coefficients, 4)
```

```
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       19.1807     0.0688 278.6062        0
## poly(radius, 2)1  11.6743     0.4964  23.5156        0
## poly(radius, 2)2  -4.7269     0.4964  -9.5214        0
```

```r
round(t(confint(fit.1)), 4)
```

```
##       (Intercept) poly(radius, 2)1 poly(radius, 2)2
## 2.5 %     19.0423          10.6767          -5.7245
## 97.5 %    19.3190          12.6720          -3.7292
```





1.
2.
3.

Galaxy Surface Brightness Profiles – Transformation

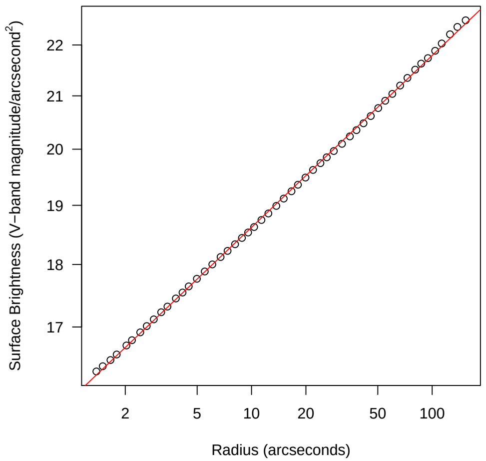
```
par.old <- par()  # Save defaults
pl <- par("plt")  # Copy plt
pl[1] <- 0.2  # Increase value
par(plt = pl)  # Increase space on left of figure
laby <- "Surface Brightness (V-band magnitude/"
plot(surf_mag ~ radius, data = NGC4406, log = "xy",
    las = 1, xlab = "Radius (arcseconds)", ylab = expression(paste(laby,
      arcsecond^2, ")", sep = "")))
fit.trans <- lm(log(surf_mag) ~ log(radius), data = NGC4406)
round(summary(fit.trans)$coefficients, 3)
```

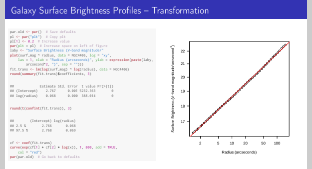
```
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    2.767      0.001 5232.363        0
## log(radius)    0.068      0.000  388.014        0
```

```
round(t(confint(fit.trans)), 3)
```

```
##        (Intercept) log(radius)
## 2.5 %        2.766       0.068
## 97.5 %       2.768       0.069
```

```
cf <- coef(fit.trans)
curve(exp(cf[1] + cf[2] * log(x)), 1, 800, add = TRUE,
    col = "red")
par(par.old)  # Go back to defaults
```





1.
2.
3.

Galaxy Surface Brightness Profiles – Transform Back to Original Scale

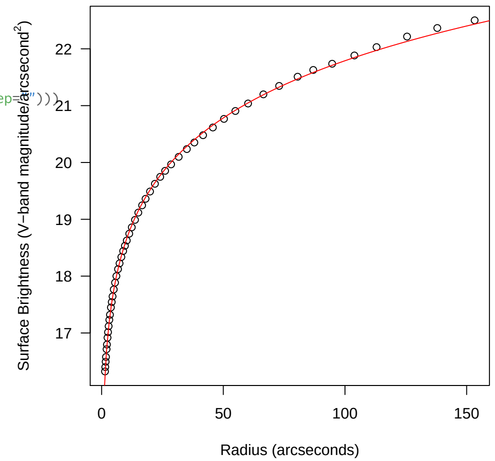
```
par.old <- par() # Save defaults
pl <- par("plt") # Copy plt
pl[1] <- 0.2 # Increase value
par(plt=pl) # Increase space on left of figure
plot(surf_mag~radius,data=NGC4406,las=1,
xlab="Radius (arcseconds)",

ylab=expression(paste("Surface Brightness (V-band magnitude/",arcsecond^2,")",sep="")))
fit.trans <- lm(log(surf_mag)~log(radius),data=NGC4406)
round(summary(fit.trans)$coefficients,3)
```

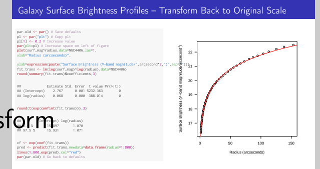
```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)     2.767      0.001 5232.363        0
## log(radius)     0.068      0.000  388.014        0
```

```
round(t(exp(confint(fit.trans))),3)
```

```
##        (Intercept) log(radius)
## 2.5 %       15.897       1.070
## 97.5 %      15.931       1.071
```

```
cf <- exp(coef(fit.trans))
pred <- predict(fit.trans,newdata=data.frame(radius=1:800))
lines(1:800,exp(pred),col="red")
par(par.old) # Go back to defaults
```





1.
2.
3.

Galaxy Surface Brightness Profiles – Nonlinear Model

Transformations are a powerful tool for fitting models.

We can transform the parameter estimates to the original scale, but we can't do that for the corresponding confidence intervals. Also, the model is optimized on the transformed scale, not the original scale.

We have fit:

$$log(y) = a + b\ log(x)$$

where y is the surface brightness and x is the radius. Transforming back to y:

$$y = e^{a+b\ log(x)} = kx^b$$

where $k = e^a$.

1.
2.
3.

Galaxy Surface Brightness Profiles – Nonlinear Model

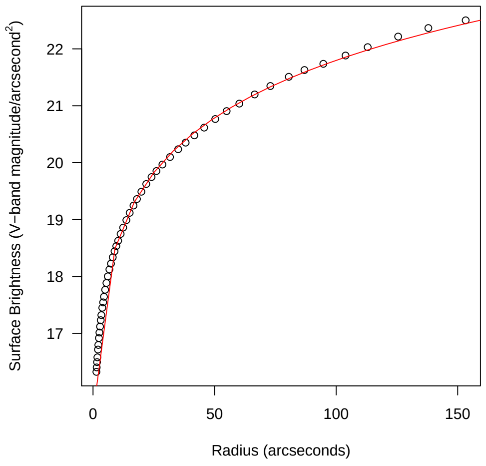
```r
par.old <- par()  # Save defaults
pl <- par("plt")  # Copy plt
pl[1] <- 0.2  # Increase value
par(plt = pl)  # Increase space on left of figure
plot(surf_mag ~ radius, data = NGC4406, las = 1, xlab = "Radius (arcseconds)",
    ylab = expression(paste("Surface Brightness (V-band magnitude/",
        arcsecond^2, ")", sep = "")))

fit.orig <- nls(surf_mag ~ k * radius^b, data = NGC4406,
    start = list(k = exp(cf[1]), b = cf[2]))
round(summary(fit.orig)$coefficients, 3)
```

```
##   Estimate Std. Error  t value Pr(>|t|)
## k   15.906      0.009 1676.211        0
## b    0.068      0.000  370.933        0
```

```r
round(t(confint(fit.orig)), 3)
```

```
##           k     b
## 2.5%  15.887 0.068
## 97.5% 15.925 0.069
```

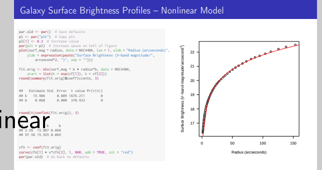
```r
cfn <- coef(fit.orig)
curve(cfn[1] * x^cfn[2], 1, 800, add = TRUE, col = "red")
par(par.old)  # Go back to defaults
```



2021-02-02

Nonlinear Fixed Effects Models

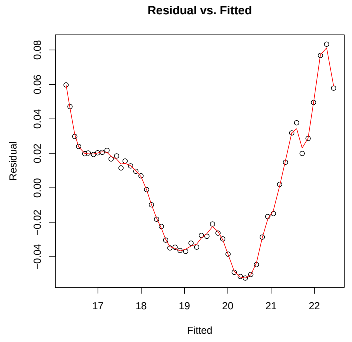
└Galaxy Surface Brightness Profiles – Nonlinear Model



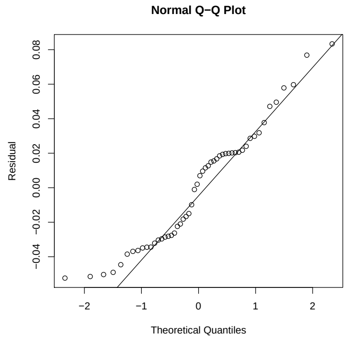
1.
2.
3.

Galaxy Surface Brightness Profiles – Diagnostic Plots

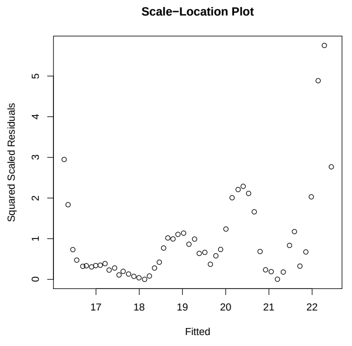
```r
plot(fitted(fit.orig), resid(fit.orig),
    xlab = "Fitted", ylab = "Residual")
lines(smooth.spline(fitted(fit.orig),
    resid(fit.orig)), col = "red")
title(main = "Residual vs. Fitted")
```



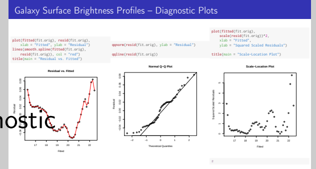

```r
qqnorm(resid(fit.orig), ylab = "Residual")

qqline(resid(fit.orig))
```




```r
plot(fitted(fit.orig),
    scale(resid(fit.orig))^2,
    xlab = "Fitted",
    ylab = "Squared Scaled Residuals")

title(main = "Scale-Location Plot")
```




```
#
```

1.
2.
3.

Galaxy Surface Brightness Profiles – Using lattice Functions

The lattice R package contains many functions to produce high quality informative graphics.

We will use it to display all three profiles for NGC 4406, NGC 4472, and NGC 4551

First, we need to combine and label the three profile datasets.

```
NGC <- rbind(NGC4406, NGC4472, NGC4551)
NGC$ngc <- c(rep("NGC 4406", nrow(NGC4406)), rep("NGC 4472",
    nrow(NGC4472)), rep("NGC 4551", nrow(NGC4551)))

library(lattice)
xyplot(surf_mag ~ radius | ngc, data = NGC, layout = c(3,
    1), xlab = "Radius (arcseconds)", ylab = expression(paste("Surface Brightness (V-band magnitude/",
    arcsecond^2, ")", sep = "")), type = "b")
```


Galaxy Surface Brightness Profiles – Using lattice Functions

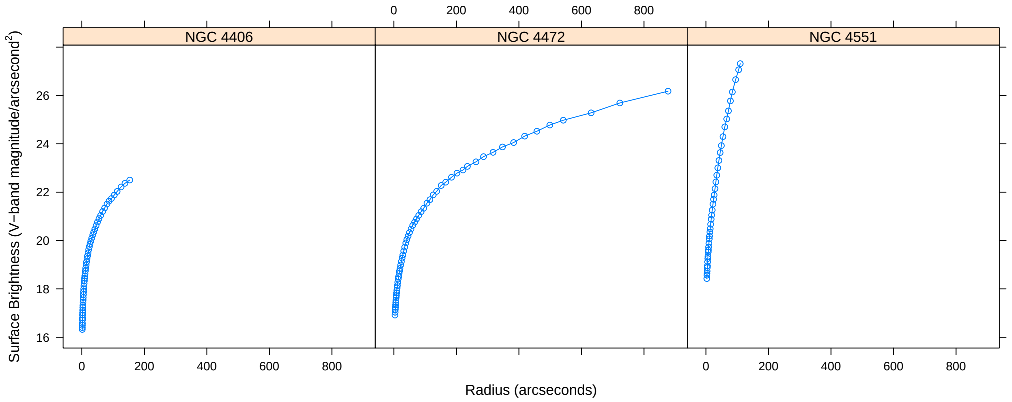
The lattice R package contains many functions to produce high quality informative graphics.

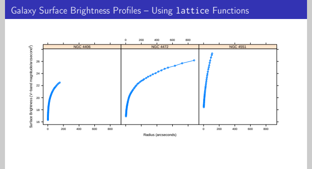
We will use it to display all three profiles for NGC 4406, NGC 4472, and NGC 4551

First, we need to combine and label the three profile datasets.

1.
2.
3.

Galaxy Surface Brightness Profiles – Using lattice Functions





1. NGC 4406 and NGC 4472 have similar curves and differ from NGC 4551.
2. The curve for NGC 4551 rises much more quickly than the others.

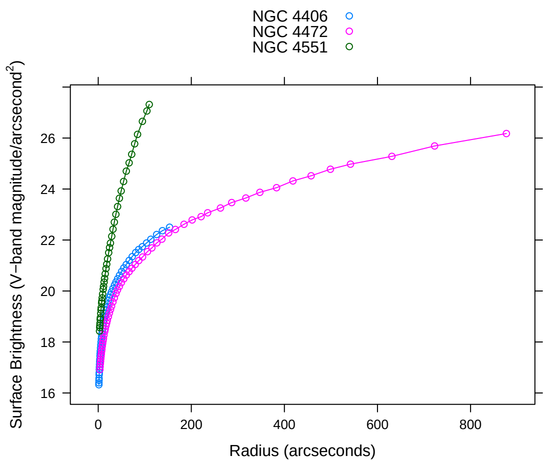
Galaxy Surface Brightness Profiles – All in One `lattice` Plot

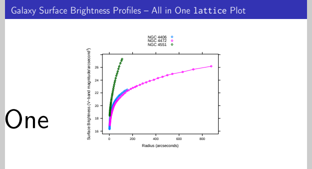
```
xyplot(surf_mag~radius,group=ngc,data=NGC,xlab="Radius (arcseconds)",
ylab=expression(paste("Surface Brightness (V-band magnitude/",arcsecond^2,")",sep="")),type="b",auto.key=TRUE)
))
```


Galaxy Surface Brightness Profiles – All in One lattice Plot

1.
2.
3.

Galaxy Surface Brightness Profiles – All in One `lattice` Plot





1. NGC 4406 and NGC 4472 have similar curves and differ from NGC 4551.
2. The curve for NGC 4551 rises much more quickly than the others.

Galaxy Surface Brightness Profiles – Nonlinear Model for All Three Galaxies

```r
# Create 2 dummy variables (NGC 4406 as reference)
NGC$g1 <- ifelse(NGC$ngc == "NGC 4472", 1, 0)
NGC$g2 <- ifelse(NGC$ngc == "NGC 4551", 1, 0)

# Determine starting values from log-log model for
# NGC 4406
fit.trans <- lm(log(surf_mag) ~ log(radius), data = NGC4406)
cf <- exp(coef(fit.trans))

fit.all <- nls(surf_mag ~ (k + kd_NGC4472 * g1 + kd_NGC4551 *
    g2) * radius^(b + bd_NGC4472 * g1 + bd_NGC4551 *
    g2), data = NGC, start = list(k = cf[1], b = cf[2],
    kd_NGC4472 = 0, kd_NGC4551 = 0, bd_NGC4472 = 0,
    bd_NGC4551 = 0))
round(summary(fit.all)$coefficients, 4)
round(t(confint(fit.all)), 4)
```


1.
2.
3.

Galaxy Surface Brightness Profiles – Nonlinear Model for All Three Galaxies

```
##              Estimate Std. Error  t value Pr(>|t|)
## k             15.9056     0.0751 211.8778   0.0000
## b              0.0684     0.0015  46.8870   0.0000
## kd_NGC4472   -0.9667     0.1086  -8.8991   0.0000
## kd_NGC4551   -0.2578     0.1244  -2.0733   0.0399
## bd_NGC4472    0.0117     0.0019   6.3189   0.0000
## bd_NGC4551    0.0424     0.0024  17.7586   0.0000

## Waiting for profiling to be done...

##              k      b kd_NGC4472 kd_NGC4551 bd_NGC4472 bd_NGC4551
## 2.5%   15.7575 0.0655    -1.1817    -0.5048     0.0080     0.0376
## 97.5% 16.0544 0.0713    -0.7514    -0.0101     0.0154     0.0471
```


1.
2.
3.

Fixed Effects vs Random Effects

The galaxy dataset NGC provided information on three galaxies. For each galaxy we had repeated measurements on surface brightness and radius. Thus we have a clustered dataset.[2]

In the previous model for the three galaxies, the differences among galaxies were captured by coefficients that measured the differences (also known as effects) among the model parameters. To measure these effects, we needed to add four coefficients (2 parameters $\times$ (3 galaxies minus 1).

Because there were only three galaxies, it makes sense to use fixed effects. This is reasonable when there are only a small number of objects. These coefficients/effects tell us something specifically about these particular galaxies.

[2] Particular examples of clustered datasets include nested (also called hierarchical) datasets and crossed datasets. The galaxy dataset is an example of a nested or hierarchical dataset.

Fixed Effects vs Random Effects

The galaxy dataset NGC provided information on three galaxies. For each galaxy we had repeated measurements on surface brightness and radius. Thus we have a clustered dataset.[2]

In the previous model for the three galaxies, the differences among galaxies were captured by coefficients that measured the differences (also known as effects) among the model parameters. To measure these effects, we needed to add four coefficients (2 parameters $\times$ (3 galaxies minus 1).

Because there were only three galaxies, it makes sense to use fixed effects. This is reasonable when there are only a small number of objects. These coefficients/effects tell us something specifically about these particular galaxies.

[2] Particular examples of clustered datasets include nested (also called hierarchical) datasets and crossed datasets. The galaxy dataset is an example of a nested or hierarchical dataset.

1.
2.
3.

Fixed Effects vs Random Effects (cont.)

Now, imagine we had similar information, not on three galaxies, but on 1000 galaxies. There would be an enormous number of additional parameters needed if we wanted to use fixed effects (1998 additional parameters). Given that, in many cases, the individual effects are not of particular interest, we can instead add just one new parameter – the variance of the effect – for each corresponding fixed parameter in the model. Thus, using the previous model, we would only add two random effects even though we had 1000 galaxies.

Just as the fixed effects parameters handled the variation among 3 galaxies, the random effects (computed as variances) would account for the variation among the 1000 galaxies. This is a very efficient way to construct a model – we always want to use as few parameters as possible.[3]

Statistical models with such random effects are referred to as mixed effects models. In R, such models can be fitted using the `nlme` and `lme4` packages (among many others).

[3]With only three galaxies, it would be difficult to estimate random effects.

Fixed Effects vs Random Effects (cont.)

Now, imagine we had similar information, not on three galaxies, but on 1000 galaxies. There would be an enormous number of additional parameters needed if we wanted to use fixed effects (1998 additional parameters). Given that, in many cases, the individual effects are not of particular interest, we can instead add just one new parameter – the variance of the effect – for each corresponding fixed parameter in the model. Thus, using the previous model, we would only add two random effects even though we had 1000 galaxies.

Just as the fixed effects parameters handled the variation among 3 galaxies, the random effects (computed as variances) would account for the variation among the 1000 galaxies. This is a very efficient way to construct a model – we always want to use as few parameters as possible.[3]

Statistical models with such random effects are referred to as mixed effects models. In R, such models can be fitted using the nlme and lme4 packages (among many others).

[3]With only three galaxies, it would be difficult to estimate random effects.

1.
2.
3.