Climate Change Ecology

The Science, Economics, and Politics of Climate Change

Drawing a 95% confidence interval in R

Posted on **August 5, 2013** by **Nathan Lemoine**

I'm writing a post on how to draw a in 95% confidence interval in R **by hand**. I spent an hour or so trying to figure this out, and most message threads point someone to the ellipse() function. However, I wanted to know how it works.

The basic problem was this. Imagine two random variables with a bivariate normal distribution, called **y**, which is an *n* x 2 matrix with *n* rows and 2 columns. The random variables are described by a mean vector **mu** and covariance matrix **S**. The equation for an ellipse is:

(y – mu) S^1 (y – mu)' = c^2

The number **c^2** controls the radius of the ellipse, which we want to extend to the 95% confidence interval, which is given by a chi-square distribution with 2 degrees of freedom. The ellipse has two axes, one for each variable. The axes have half lengths equal to the square-root of the eigenvalues, with the largest eigenvalue denoting the largest axis. A further description of this can be found in any multivariate statistics book (or online).

To calculate the ellipse, we need to do a few things: 1) convert the variables to polar coordinates, 2) extend the new polar variables by the appropriate half lengths (using eigenvalues), 3) rotate the coordinates based on the variances and covariances, and 4) move the location of the new coordinates back to the original means. This will make more sense when we do it by hand.

First, generate some data, plot it, and use the ellipse() function to make the 95% confidence interval. This is the target interval (I use it to check myself. If my calculations match, hooray. If not, I screwed up).

```r
library(mvtnorm) # References rmvnorm()
library(ellipse) # References ellipse()
set.seed(17)

# Set the covariance matrix
sigma2 <- matrix(c(5, 2, 2, 5), ncol=2)

# Set the means
mu <- c(5,5)

# Get the correlation matrix
P <- cov2cor(sigma2)

# Generate the data
p <- rmvnorm(n=50, mean=mu, sigma=sqrt(sigma2))

# Plot the data
plot(p)

# Plot the ellipse
lines( ellipse( P, centre = c(5,5)) , col='red')
```

Second, get the eigenvalues and eigenvectors of the **correlation** matrix.

```r
evals <- eigen(P)$values
evecs <- eigen(P)$vectors
```

Third, make a vector of coordinates for a full circle, from 0 to 2*pi and get the critical value (**c^2**).

```r
# Angles of a circle
a <- seq(0, 2*pi, len=100)

# Get critical value
c2 <- qchisq(0.95, 2)
c <- sqrt(c2)
```

The vector A above are angles that describe a unit circle. The coordinates of a unit circle are found by x = cos(a) and y = sin(a) (use trigonometry of a triangle to get this, where the hypotenuse = 1). We need to extend the unit circle by the appropriate lengths based on the eigenvalues and then even more by the critical value.

```r
# Get the distances
xT <- c * sqrt(evals[1]) * cos(a)
```

```
yT <- c * sqrt(evals[2]) * sin(a)

M <- cbind(xT, yT)
```

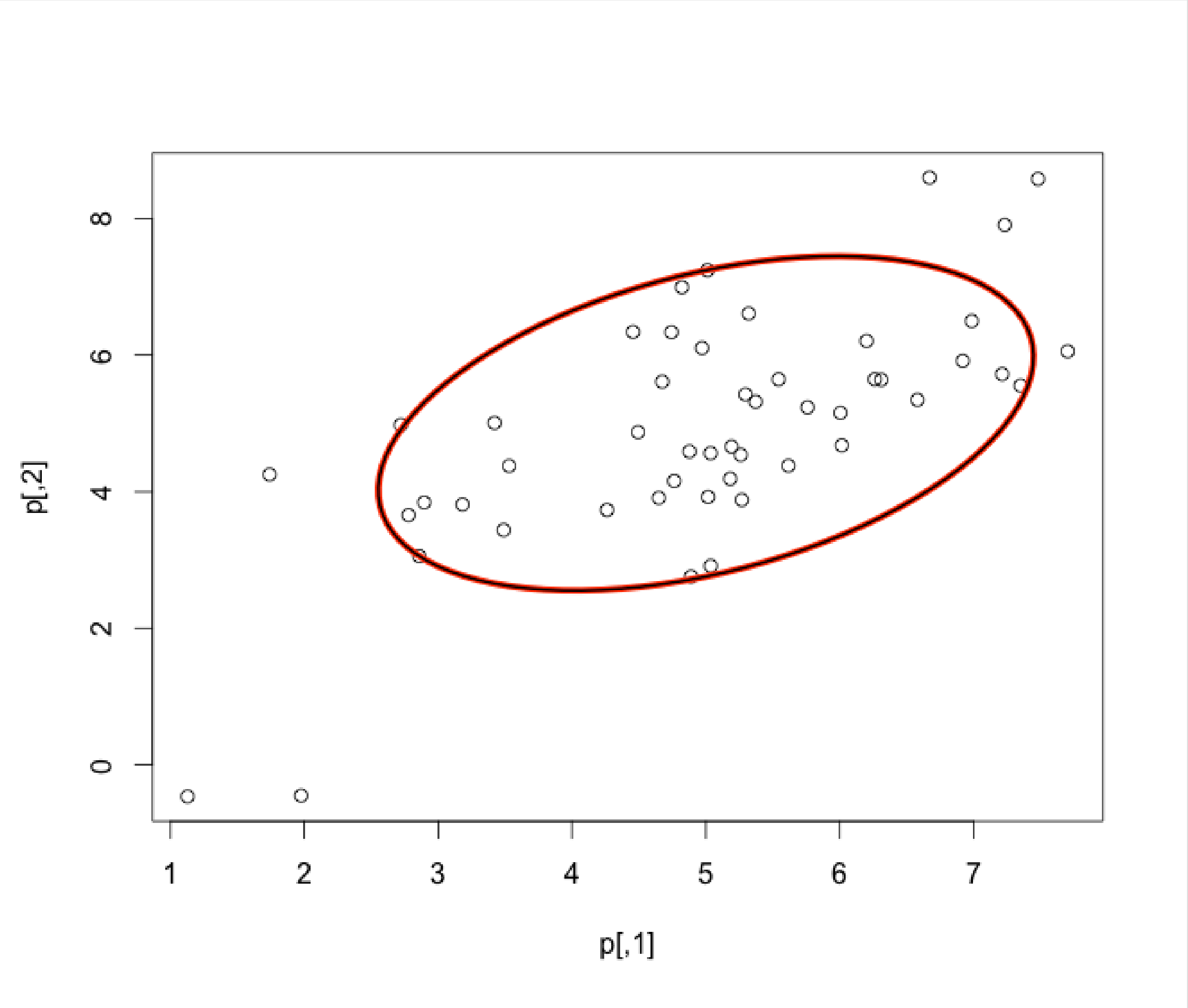
If you plot M, you'll get an ellipse of the appropriate axes lengths, but centered on 0 and unrotated. Rotate the ellipse using the eigenvectors, which describe the relationships between the variables (more appropriately, they give the directions for the vectors of the major axes of variation). Use the equation **u*M'** (write this out to see why this works).

```
# Covert the coordinates
transM <- evecs %*% t(M)
transM <- t(transM)
```

The final step is to move the rotated ellipse back to the original scale (centered around the original means) and plot the data.

```
lines(transM + mu)
```

This gives the following plot, with the red line being the output from the ellipse() function.



And that’s that! Hopefully this helps someone like me who spent hours looking but couldn’t find anything.

RELATED

Eigen-analysis of Linear Model Behavior in R
In "Blogs"

Python in Ecology - Stability Analysis of a Predator-Prey Model
In "Blogs"

R for Ecologists: Simulating Species-Area Curves (linear vs. nonlinear regression)
In "Blogs"

This entry was posted in **Blogs**, **R**, **Statistics** and tagged **R**, **statistics** by **Nathan Lemoine**. Bookmark the **permalink [https://climateecology.wordpress.com/2013/08/05/drawing-a-95-confidence-interval-in-r/]** .

About Nathan Lemoine

I am a PhD student studying the effects of climate change on plant-herbivore interactions and community ecology.
View all posts by Nathan Lemoine →

15 THOUGHTS ON "DRAWING A 95% CONFIDENCE INTERVAL IN R"

Luis A Cubillos
on **August 5, 2013 at 9:58 PM** said:

Excelent job !. The following function was obtained from Claude J (2008) Morphometrics with R, Springer, which could be used as a test for your approach:

```
ellipse<-function(x,y,conf=0.95,np=50)
{centroid<-apply(cbind(x,y),2,mean)
ang <- seq(0,2*pi,length=np)
z<-cbind(cos(ang),sin(ang))
radiuscoef<-qnorm((1-conf)/2, lower.tail=F)
vcvxy<-var(cbind(x,y))
r<-cor(x,y)
M1<-matrix(c(1,1,-1,1),2,2)
M2<-matrix(c(var(x), var(y)),2,2)
M3<-matrix(c(1+r, 1-r),2,2, byrow=T)
ellpar<-M1*sqrt(M2*M3/2)
t(centroid + radiuscoef * ellpar %*% t(z))}
```

Petr Keil
on **August 8, 2013 at 6:39 PM** said:

Hi Nathan,
thanks for bringing this out. I have some comments and questions:

1. Here is a little quarrel: You are not at all specific about what is the meaning of the 95% confidence intervals – you use a very general language. To me, it seems that you simply present a way to plot 95% quantile of a bivariate normal distribution, and I don't understand why to you call this a confidence interval – a term that is usually reserved for quantification of uncertainty of parameter estimates. I worry that some readers may think that you present a general approach that is applicable also to linear regression, or other models, yet is is not the case. Or is it?

2. Here is a question: I have always been confused by the very existence of the ellipse library. It is similarly vague about what "confidence regions" stand for. Mostly, it seem to draw the ellipsoid 95% quantiles of a bivariate normal distribution. Yet, it gives you an option to use the ellipse.lm() function that takes a linear model as an argument and pops out what seems like a bivariate normal distribution of uncertainty in coefficient estimates. Is there any reason to expect that coefficient estimates should follow a multivariate (bivariate) normal distribution? Shouldn't regression coefficients be independent on each other? What am I missing here?

3. Another question: What exactly did you use your confidence ellipse for? What kind of data/analysis did you applied it to? And why did you think it necessary?

Thanks a lot in advance!

Nathan Lemoine
on **August 8, 2013 at 7:13 PM** said:

I'll try to address your questions in order, if I can:

1) You're correct in that what I plot is not the 95% confidence interval of parameter estimates. It would be more accurately called the 95% quartile range, as you suggest. This is just mislabeling on my part. What I plot is, more or less, the ellipse which contains 95% of the data points (or should). This is not the same as a bivariate 95% confidence ellipse for parameter estimates, i.e. a slope and intercept in a linear regression model. That ellipse would come from the variance-covariance matrix of the linear regression model, not from the raw data (as is the case here).

2) Regression coefficients aren't always independent of each other. Think of a situation where the slope and intercept are inter-related. For example, imagine a positive line. If the line is shallow (small slope), the intercept is high. If the line is steep (large slope), the intercept is low. In extreme cases, the slope and intercept can be nearly perfectly correlated (which can cause MCMC Gibbs sampler chains for posterior distributions in Bayesian models to take their sweet time converging). This happens frequently when the predictor variable has very high values (a large mean). A way around this is to center the predictor variable, which usually yields uncorrelated estimates of slopes and intercepts, but this isn't commonly done with ecological analyses (that I see, anyway). Maybe it is behind the scenes and no one reports it. In most practical cases I've encountered, the slopes and intercepts are pretty well correlated, but I try to avoid having extreme correlations.

3) I was just playing around, really. I use a lot of multivariate statistics because ecology is a multivariate discipline (as I'm sure you know). Getting a solid understanding of eigenvectors and eigenvalues and how they relate to the major axes of variation is a necessity for understanding most multivariate statistical tests (i.e. MANOVA) and ordination (i.e. PCA) procedures. I wasn't doing anything in particular, but this was more of a pedagogical tool for myself that I thought others might be interested in as well.

I hope these answers help (and that I got them right)!

Petr Keil
on **August 8, 2013 at 9:18 PM** said:

Nathan thanks! You answered my question precisely and exactly! I did not know that about the regression coefficients, that they can be correlated – I was trying to google it but unsuccessfully, so that is why I was confused. I will need to read more about it so that I understand it fully. Have you got a nice reference for that?

Nathan Lemoine
on **August 8, 2013 at 10:06 PM** said:

A few books where I learned it:

Draper and Smith (Applied Linear Regression) cover it pretty well for linear regression, including the calculation of the variance-covariance matrix for the parameters.

Zuur et al. (Mixed Effects Models in R) also touch on it more as a diagnostic than anything for mixed-effects models.

Kruschke (Doing Bayesian Analysis) might be one of the best books that I've ever read for introductory applied Bayesian statistics, and he covers it pretty thoroughly with respect to MCMC sampling.

Gelman and Hill (Data Analysis Using Multilevel/Hierarchical Models) also cover it pretty thoroughly with respect to both mixed effects models and Bayesian models.

Honestly, I found that Kruschke and Gelman/Hill probably do the best job of explaining it. Draper and Smith is just an excellent reference to have on the shelf for anything related to linear models and regression.

Fr.
on **August 11, 2013 at 3:27 PM** said:

There's a stat_ellipse function to do the same in ggplot2.

Benjamin Blalock
on **July 27, 2016 at 4:32 PM** said:

So wait. Does stat_ellipse draw a 95% quartile range or a 95% confidence region?

Catherine Dion
on **August 29, 2013 at 2:02 PM** said:

Hi!

I use the package “candisc” to make a canonical discriminant analysis and the ellipse appear automatically. I can change the confidence interval (0.99, 0.95 or 0.90 …) but do you know if the function use the same technical that you explain here to build the ellipse ?

Thank you

> **Nathan Lemoine**
> on **August 30, 2013 at 1:41 PM** said:
>
> I don’t know off hand and I haven’t had time to really dig into it to figure it out. I would suspect that it does something similar, even if the programming is different.
>
> The ellipse() function is very similar, but programmed a bit differently.

Armen
on **August 9, 2014 at 3:28 PM** said:

Hi Nathan,

I was playing with some bivariate-normal concentration ellipses for bivariate distribution of
two variables, and I saw this post. It looks good but when I use your data set and draw the bivariate-normal concentration ellipses by dataEllipse() in car package it gives different orientation/size for the ellipse (see the last section of the script below).

```
library(mvtnorm) # References rmvnorm()
library(ellipse) # References ellipse()
set.seed(17)
# Set the covariance matrix
sigma2 <- matrix(c(5, 2, 2, 5), ncol=2)
# Set the means
mu <- c(5,5)
# Get the correlation matrix
P <- cov2cor(sigma2)
# Generate the data
p <- rmvnorm(n=50, mean=mu, sigma=sqrt(sigma2))
# Plot the data
plot(p, xlim=c(-1,10), ylim=c(-1,10))
# Plot the ellipse
lines( ellipse( P, centre = c(5,5)) , col='red')

#adding a bivariate-normal concentration ellipse to your original plot by using dataEllipse() in car package
library(car)
dataEllipse(p, levels=0.95, add=T, center.pch=F, robust=F, fill=F, fill.alpha=.1, plot.points=F, col='green', lty='solid')
```

Do you know why is this happening? or i am missing something…

Thanks,
Armen

Nathan Lemoine
on **August 9, 2014 at 4:34 PM** said:

I don’t have a good answer. Looking at the methods, the ‘ellipse’ package uses the method I used here (almost identically). The ‘dataEllipse’ function from car uses the Cholesky decomposition of the covariance matrix. As far as why that leads to different ellipses, I’m not sure.

I’ve played around with it some. If you use the ellipse() function from car (not data ellipse), like this:
ellipse(center = c(5,5), shape = sigma2, radius = 1)
you get an ellipse of the same orientation as the regular ellipse function.

dataEllipse is simply a wrapper around the car version of ellipse, and passes it a few arguments, like radius. Radius is what scales the ellipse, and dataEllipse uses a different calculation of the radius (based on the f-distribution, not chi-squared like mine/regular ellipse function). I suspect it’s this difference, using F and not X2, that’s the big difference.

Why they chose to do it like this is beyond me. You’d probably get a pretty quick answer on StackOverflow or Stats Exchange. If you find out, do let me know!

Armen
on **August 9, 2014 at 6:12 PM** said:

OK. Thanks a lot for you explanation! I will let you know when I figure it out.

Vincent
on **March 10, 2015 at 6:49 AM** said:

Nice post! I recently wrote a similar article on how to draw an error ellipse, with code samples in Matlab and C++ that might be complimentary to your solution: http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/

Johnny Cash
on **June 18, 2015 at 6:41 AM** said:

Reblogged this on In the process of ….

Kevin Mahoney
on **March 28, 2016 at 6:21 AM** said:

Thanks for a very informative and useful post, Nathan.
I actually used your code as the basis of an Excel routine to calculate the 95% confidence interval (using the excellent PopTools add-in for the eigenvalues and vectors).
I wasn't able to replicate your results exactly until I realised that you calculate the correlation matrix from the predetermined covariance matrix (sigma2 <- matrix(c(5, 2, 2, 5), ncol=2) rather from the simulated data.
Should anyone want an Excel version of your code, please e-mail me at Mahonk@aol.com.
Thanks again.
Kevin