

[Home](#)[contact](#)

On the lambda

(<http://www.onthelambda.com/>)

another blog from a data analyst – by tony fischetti

Kickin' it with elastic net regression

👤 tony.fischetti@gmail.com (<http://www.onthelambda.com/author/tony-fischetti@gmail-com/>) 📅 August 19, 2015

(<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/>) 💬 [7 Comments](http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/#comments) (<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/#comments>)

With the kind of data that I usually work with, overfitting regression models can be a huge problem if I'm not careful. Ridge regression is a really effective technique for thwarting overfitting. It does this by penalizing the L2 norm (euclidean distance) of the coefficient vector which results in "shrinking" the beta coefficients. The aggressiveness of the penalty is controlled by a parameter λ .

Lasso regression is a related regularization method. Instead of using the L2 norm, though, it penalizes the L1 norm (manhattan distance) of the coefficient vector. Because it uses the L1 norm, some of the coefficients will shrink to zero while lambda increases. A similar effect would be achieved in Bayesian linear regression using a Laplacian prior (strongly peaked at zero) on each of the beta coefficients.

Because some of the coefficients shrink to zero, the lasso doubles as a crackerjack feature selection technique in addition to a solid shrinkage method. This property gives it a leg up on ridge regression. On the other hand, the lasso will occasionally achieve poor results when there's a high degree of collinearity in the features and ridge regression will perform better. Further, the L1 norm is underdetermined when the number of predictors exceeds the number of observations while ridge regression can handle this.

Elastic net regression is a hybrid approach that blends both penalization of the L2 *and* L1 norms. Specifically, elastic net regression minimizes the following...

$$\|y - X\beta\| + \lambda[(1 - \alpha)|\beta|_2^2 + \alpha|\beta|_1]$$

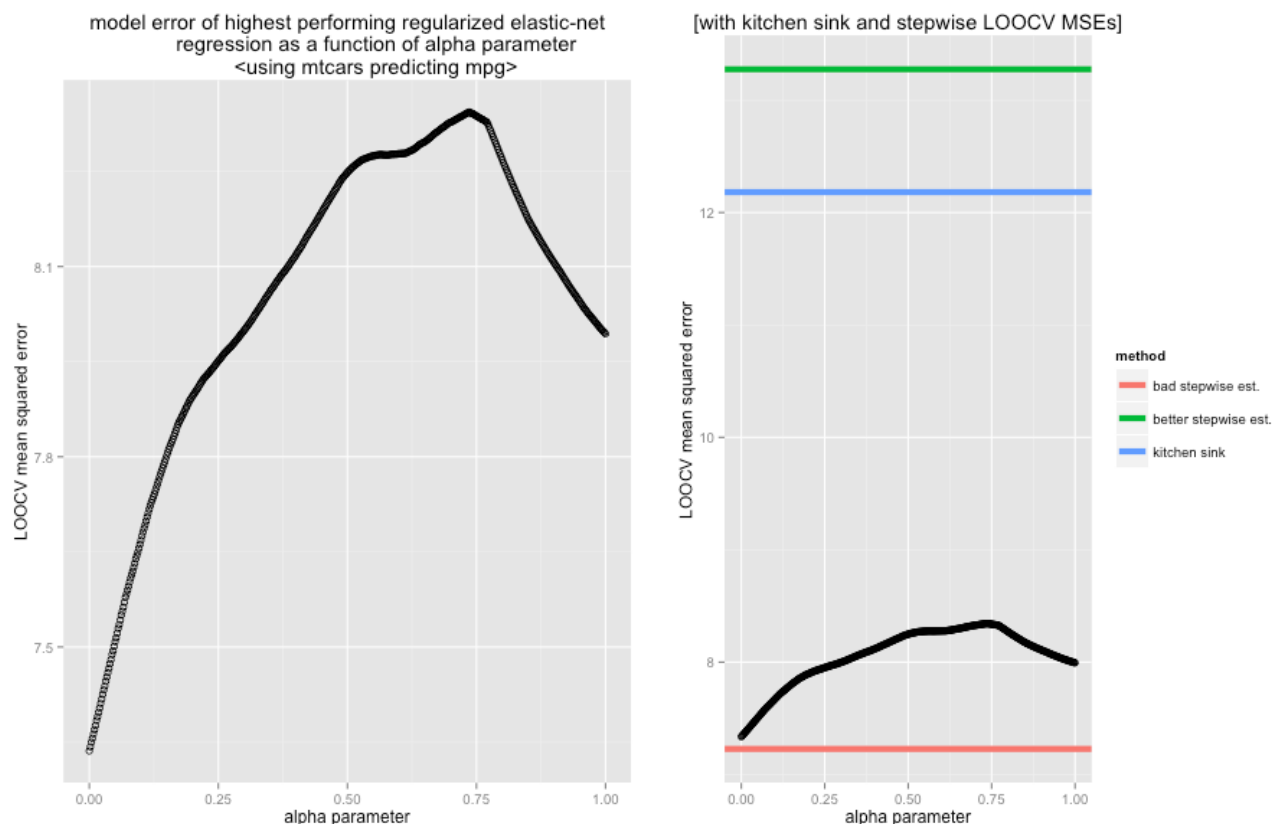
the α hyper-parameter is between 0 and 1 and controls how much L2 or L1 penalization is used (0 is ridge, 1 is lasso).

The usual approach to optimizing the lambda hyper-parameter is through cross-validation—by minimizing the cross-validated mean squared prediction error—but in elastic net regression, the optimal lambda hyper-parameter also depends upon and is heavily dependent on the alpha hyper-parameter (hyper-hyper-parameter?).

This blog post takes a cross-validated approach that uses grid search to find the optimal alpha hyper-parameter while also optimizing the lambda hyper-parameter for three different data sets. I also compare the performances against the stepwise regression and showcase some of the dangers of using stepwise feature selection.

mtcars

In this example, I try to predict "miles per gallon" from the other available attributes. The design matrix has 32 observations and 10 predictors and there is a high degree of collinearity (as measured by the [variance inflation factors](https://en.wikipedia.org/wiki/Variance_inflation_factor) (https://en.wikipedia.org/wiki/Variance_inflation_factor)).



(<http://www.onthelambda.com/wp-content/uploads/2015/08/mtcarsplot.png>)

The left panel above shows the leave-one-out cross validation (LOOCV) mean squared error of the model with the optimal lambda (as determined again by LOOCV) for each alpha parameter from 0 to 1. This panel indicates that if our objective is to purely minimize MSE (with no regard for model complexity) than pure ridge regression outperforms any blended elastic-net model. This is probably because of the substantial collinearity. Interestingly, the lasso outperforms blended elastic net models that weight the lasso heavily.

The right panel puts things in perspective by plotting the LOOCV MSEs along with the MSE of the "kitchen sink" regression (the blue line) that includes all features in the model. As you can see, any degree of regularization offers a substantial improvement in model generalizability.

It is also plotted with two estimates of the MSE for models that blindly use the coefficients from automated bi-directional stepwise regression. The first uses the features selected by performing the stepwise procedure on *the whole dataset* and then assesses the model performance (the red line). The second estimate uses the step procedure and resulting features on *only the training set for each fold of the cross validations*. This is the estimate without the subtle but treacherous "knowledge leaking" eloquently described in [this plot post](http://www.alfredo.motta.name/cross-validation-done-wrong/) (<http://www.alfredo.motta.name/cross-validation-done-wrong/>). This should be considered the more correct assessment of the model. As you can see, if we weren't careful about interpreting the stepwise regression, we would have gotten an incredibly inflated and inaccurate view of the model performance.

Forest Fires

The second example uses a very-difficult-to-model dataset from [University of California, Irvine machine learning repository](http://archive.ics.uci.edu/ml/datasets/Forest+Fires) (<http://archive.ics.uci.edu/ml/datasets/Forest+Fires>). The task is to predict the burnt area from a forest fire given 11 predictors. It has 517 observations. Further, there is a relatively low degree of collinearity between predictors.

(<http://www.onthelambda.com/wp-content/uploads/2015/08/fireplot.png>)

Again, highest performing model is the pure ridge regression. This time, the performance asymptotes as the alpha hyper-parameter increases. The variability in the MSE estimates is due to the fact that I didn't use LOOCV and used 400-k CV instead because I'm impatient.

As with the last example, the properly measured stepwise regression performance isn't so great, and the kitchen sink model outperforms it. However, in contrast to the previous example, there was a lot less variability in the selected features across folds—this is probably because of the significantly larger number of observations.

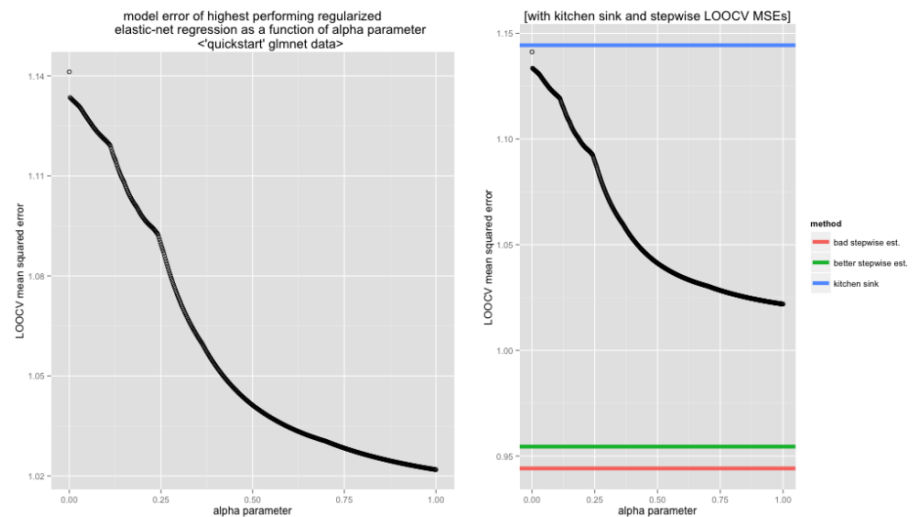
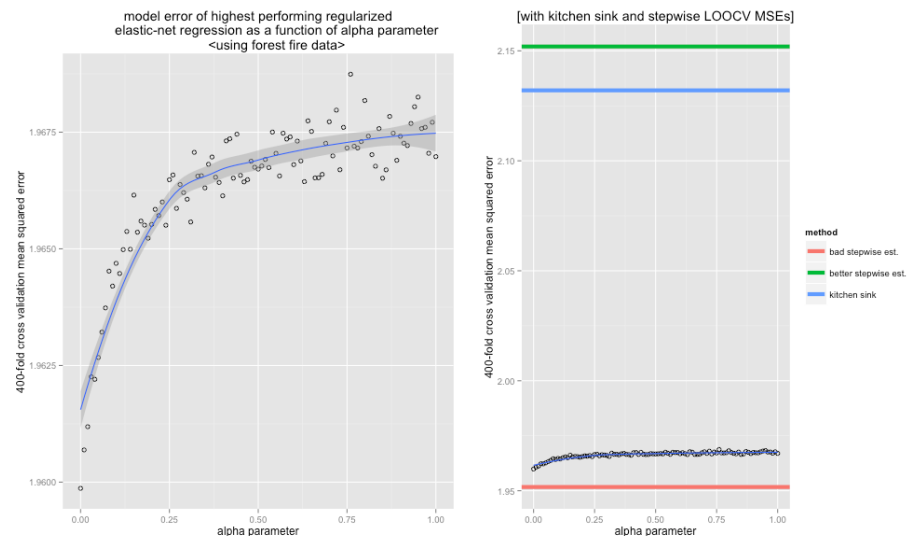
"QuickStartExample"

This dataset is a contrived one that is included with the excellent [glmnet package](http://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html) (http://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html) (the one I'm using for the elastic net regression). This dataset has a relatively low degree of collinearity, has 20 features and 100 observations. I have no idea how the package authors created this dataset.

(<http://www.onthelambda.com/wp-content/uploads/2015/08/quickstartplot.png>)

Finally, an example where the lasso outperforms ridge regression! I think this is because the dataset was specifically manufactured to have a small number of genuine predictors with large effects (as opposed to many weak predictors).

Interestingly, stepwise progression far outperforms both—probably for the very same reason. From fold to fold, there was virtually no variation in the features that the stepwise method automatically chose.



Conclusion

So, there you have it. Elastic net regression is awesome because it can perform *at worst as good* as the lasso or ridge and—though it didn't on these examples—can sometimes substantially outperform both.

Also, be careful with step-wise feature selection!

PS: If, for some reason, you are interested in the R code I used to run these simulations, you can find it [on this GitHub Gist](https://gist.github.com/tonyfischetti/b4fd17a94aa0a6e07fdb) (<https://gist.github.com/tonyfischetti/b4fd17a94aa0a6e07fdb>).

share this: [f](#) [t](#) [g+](#) [d](#) [p](#) [in](#) [t](#) [e](#)

machine learning (<http://www.onthelambda.com/category/machine-learning/>), R (<http://www.onthelambda.com/category/r/>) data mining (<http://www.onthelambda.com/tag/data-mining/>), ggplot2 (<http://www.onthelambda.com/tag/ggplot2/>), machine learning

(<http://www.onthelambda.com/tag/machine-learning/>), R (<http://www.onthelambda.com/tag/r/>), research (<http://www.onthelambda.com/tag/research/>), statistics (<http://www.onthelambda.com/tag/statistics/>) ■ Bookmark (<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/>)

◀ PREVIOUS ARTICLE

Lessons learned in high-performance R

(<http://www.onthelambda.com/2015/05/31/lessons-learned-in-high-performance-r/>)

7 RESPONSES

Pingback: [Kickin' it with elastic net regression | Mubashir Qasim](#) (<http://mqasim.me/?p=37058>)



KATYA VASILAKY ([HTTP://KATYAVASILAKY.COM](http://katyavasilaky.com))

August 20, 2015 / 1:24 pm

Great post. Thanks!

↩ Reply (<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/?replytocom=9664#respond>)



TONY.FISCHETTI@GMAIL.COM

August 20, 2015 / 1:54 pm

Aw, thanks!

↩ Reply (<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/?replytocom=9665#respond>)



MARIODENG

September 3, 2015 / 6:13 am

Can't say anything else, good job.

Thank you!

↩ Reply (<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/?replytocom=9800#respond>)

Pingback: [Distilled News | Data Analytics & R](#) (<http://advanceddataanalytics.net/2015/08/20/distilled-news-182/>)



THIBAUT P

August 21, 2015 / 9:31 am

Really interesting article!

But isn't the y-axis of the right graphs quite small? It seems like MSEs are not that different from each other.

It would be nice to plot errors for the different methods to see if they're really different :)

↩ Reply (<http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/?replytocom=9675#respond>)



ANDREAS

September 4, 2015 / 6:33 pm

Hi,

Thanks for the interesting read! I have done some research myself in a similar context. So it was very useful to see another angle to what I myself have looked at.

If I may do some shameless self promotion, if interested, my papers are in the context of gene regulatory networks. Specifically in trying to infer a

regulatory (dynamic) structure of genetic interactions.

In the first paper we looked at feature selection or sparsity estimation

<http://online.liebertpub.com/doi/abs/10.1089/cmb.2012.0268> (<http://online.liebertpub.com/doi/abs/10.1089/cmb.2012.0268>)

and in the second relevant paper we looked at something similar to co-linearity in relation to performance of finding the "correct" structure of the regulatory network.

<http://pubs.rsc.org/en/content/articlelanding/2014/mb/c4mb00419a/unauth#!divAbstract>

<http://pubs.rsc.org/en/content/articlelanding/2014/mb/c4mb00419a/unauth#!divAbstract>

I thought your post overlapped so well I had to share this.

Again, really interesting read. Especially your comment about elastic net being as good as either L1 or L2. Something we didn't consider for our publication was to do a grid search for the alpha parameter, for various reasons, which I should do in the future.

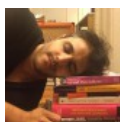
↩ [Reply \(http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/?replytocom=9819#respond\)](http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/?replytocom=9819#respond)

Leave a Reply

Your email address will not be published. Required fields are marked *

POST COMMENT

ABOUT ME



Data scientist at [CollegeFactual](http://www.collegefactual.com) (<http://www.collegefactual.com>). Vegetarian. Interests lie in R, python, cognition, game theory, machine learning, books, ethics, post-punk, animal rights, and oxford commas.

Follow @tonyfischetti

[f \(https://www.facebook.com/tony.fischetti\)](https://www.facebook.com/tony.fischetti) [github \(https://github.com/tonyfischetti\)](https://github.com/tonyfischetti) [t \(https://twitter.com/tonyfischetti\)](https://twitter.com/tonyfischetti)

TAG CLOUD

[awk \(http://www.onthelambda.com/tag/awk/\)](http://www.onthelambda.com/tag/awk/) [classification \(http://www.onthelambda.com/tag/classification/\)](http://www.onthelambda.com/tag/classification/) [data mining \(http://www.onthelambda.com/tag/data-mining/\)](http://www.onthelambda.com/tag/data-mining/) [datavis \(http://www.onthelambda.com/tag/datavis/\)](http://www.onthelambda.com/tag/datavis/) [debian \(http://www.onthelambda.com/tag/debian/\)](http://www.onthelambda.com/tag/debian/) [dplyr \(http://www.onthelambda.com/tag/dplyr/\)](http://www.onthelambda.com/tag/dplyr/) [evolutionary biology \(http://www.onthelambda.com/tag/evolutionary-biology/\)](http://www.onthelambda.com/tag/evolutionary-biology/) [functional programming \(http://www.onthelambda.com/tag/functional-programming/\)](http://www.onthelambda.com/tag/functional-programming/) [ggplot2 \(http://www.onthelambda.com/tag/ggplot2/\)](http://www.onthelambda.com/tag/ggplot2/) [graph theory \(http://www.onthelambda.com/tag/graph-theory/\)](http://www.onthelambda.com/tag/graph-theory/) [haskell \(http://www.onthelambda.com/tag/haskell/\)](http://www.onthelambda.com/tag/haskell/) [high performance computing \(http://www.onthelambda.com/tag/high-performance-computing/\)](http://www.onthelambda.com/tag/high-performance-computing/)

[performance-computing/](http://www.onthelambda.com/tag/performance-computing/) [hypothesis testing](http://www.onthelambda.com/tag/hypothesis-testing/) [linguistics](http://www.onthelambda.com/tag/linguistics/)
[\(http://www.onthelambda.com/tag/linguistics/\)](http://www.onthelambda.com/tag/linguistics/) [lisp](http://www.onthelambda.com/tag/lisp/) [\(http://www.onthelambda.com/tag/lisp/\)](http://www.onthelambda.com/tag/lisp/) [mac](http://www.onthelambda.com/tag/mac/) [\(http://www.onthelambda.com/tag/mac/\)](http://www.onthelambda.com/tag/mac/)
[machine learning](http://www.onthelambda.com/tag/machine-learning/) [\(http://www.onthelambda.com/tag/machine-learning/\)](http://www.onthelambda.com/tag/machine-learning/) [magrittr](http://www.onthelambda.com/tag/magrittr/) [\(http://www.onthelambda.com/tag/magrittr/\)](http://www.onthelambda.com/tag/magrittr/)
[markov chains](http://www.onthelambda.com/tag/markov-chains/) [\(http://www.onthelambda.com/tag/markov-chains/\)](http://www.onthelambda.com/tag/markov-chains/) [monads](http://www.onthelambda.com/tag/monads/) [\(http://www.onthelambda.com/tag/monads/\)](http://www.onthelambda.com/tag/monads/) [music](http://www.onthelambda.com/tag/music/)
[\(http://www.onthelambda.com/tag/music/\)](http://www.onthelambda.com/tag/music/) [natural language processing](http://www.onthelambda.com/tag/natural-language-processing/) [\(http://www.onthelambda.com/tag/natural-language-processing/\)](http://www.onthelambda.com/tag/natural-language-processing/) [nhst](http://www.onthelambda.com/tag/nhst/) [\(http://www.onthelambda.com/tag/nhst/\)](http://www.onthelambda.com/tag/nhst/) [optical character recognition](http://www.onthelambda.com/tag/optical-character-recognition/) [\(http://www.onthelambda.com/tag/optical-character-recognition/\)](http://www.onthelambda.com/tag/optical-character-recognition/) [os x](http://www.onthelambda.com/tag/os-x/) [\(http://www.onthelambda.com/tag/os-x/\)](http://www.onthelambda.com/tag/os-x/) [p-values](http://www.onthelambda.com/tag/p-values/) [\(http://www.onthelambda.com/tag/p-values/\)](http://www.onthelambda.com/tag/p-values/) [pedagogy](http://www.onthelambda.com/tag/pedagogy/)
[\(http://www.onthelambda.com/tag/pedagogy/\)](http://www.onthelambda.com/tag/pedagogy/) [perl](http://www.onthelambda.com/tag/perl/) [\(http://www.onthelambda.com/tag/perl/\)](http://www.onthelambda.com/tag/perl/) [python](http://www.onthelambda.com/tag/python/)
[\(http://www.onthelambda.com/tag/python/\)](http://www.onthelambda.com/tag/python/) [R](http://www.onthelambda.com/tag/r/)
[\(http://www.onthelambda.com/tag/r/\)](http://www.onthelambda.com/tag/r/) [reading](http://www.onthelambda.com/tag/reading/) [\(http://www.onthelambda.com/tag/reading/\)](http://www.onthelambda.com/tag/reading/) [research](http://www.onthelambda.com/tag/research/)
[\(http://www.onthelambda.com/tag/research/\)](http://www.onthelambda.com/tag/research/) [sake](http://www.onthelambda.com/tag/sake/) [\(http://www.onthelambda.com/tag/sake/\)](http://www.onthelambda.com/tag/sake/) [science](http://www.onthelambda.com/tag/science/)
[\(http://www.onthelambda.com/tag/science/\)](http://www.onthelambda.com/tag/science/) [scientific workflow systems](http://www.onthelambda.com/tag/scientific-workflow-systems/) [\(http://www.onthelambda.com/tag/scientific-workflow-systems/\)](http://www.onthelambda.com/tag/scientific-workflow-systems/) [self-tracking](http://www.onthelambda.com/tag/self-tracking/)
[\(http://www.onthelambda.com/tag/self-tracking/\)](http://www.onthelambda.com/tag/self-tracking/) [shell](http://www.onthelambda.com/tag/shell/) [\(http://www.onthelambda.com/tag/shell/\)](http://www.onthelambda.com/tag/shell/) [shiny](http://www.onthelambda.com/tag/shiny/) [\(http://www.onthelambda.com/tag/shiny/\)](http://www.onthelambda.com/tag/shiny/)
[software](http://www.onthelambda.com/tag/software/) [\(http://www.onthelambda.com/tag/software/\)](http://www.onthelambda.com/tag/software/) [statistics](http://www.onthelambda.com/tag/statistics/)
[\(http://www.onthelambda.com/tag/statistics/\)](http://www.onthelambda.com/tag/statistics/) [twitter](http://www.onthelambda.com/tag/twitter/)
[\(http://www.onthelambda.com/tag/twitter/\)](http://www.onthelambda.com/tag/twitter/) [unix](http://www.onthelambda.com/tag/unix/) [\(http://www.onthelambda.com/tag/unix/\)](http://www.onthelambda.com/tag/unix/)

RECENT POSTS

- [Kickin' it with elastic net regression](http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/) [\(http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/\)](http://www.onthelambda.com/2015/08/19/kickin-it-with-elastic-net-regression/)
- [Lessons learned in high-performance R](http://www.onthelambda.com/2015/05/31/lessons-learned-in-high-performance-r/) [\(http://www.onthelambda.com/2015/05/31/lessons-learned-in-high-performance-r/\)](http://www.onthelambda.com/2015/05/31/lessons-learned-in-high-performance-r/)
- [The hardest thing about teaching statistics](http://www.onthelambda.com/2015/04/30/the-hardest-thing-about-teaching-statistics/) [\(http://www.onthelambda.com/2015/04/30/the-hardest-thing-about-teaching-statistics/\)](http://www.onthelambda.com/2015/04/30/the-hardest-thing-about-teaching-statistics/)
- [I'm all about that bootstrap 'bout that bootstrap](http://www.onthelambda.com/2015/03/21/im-all-about-that-bootstrap-bout-that-bootstrap/) [\(http://www.onthelambda.com/2015/03/21/im-all-about-that-bootstrap-bout-that-bootstrap/\)](http://www.onthelambda.com/2015/03/21/im-all-about-that-bootstrap-bout-that-bootstrap/)
- [Playing around with #rstats twitter data](http://www.onthelambda.com/2015/02/28/playing-around-with-rstats-twitter-data/) [\(http://www.onthelambda.com/2015/02/28/playing-around-with-rstats-twitter-data/\)](http://www.onthelambda.com/2015/02/28/playing-around-with-rstats-twitter-data/)
- [Assertive R programming in dplyr/magrittr pipelines](http://www.onthelambda.com/2015/01/23/assertive-r-programming-in-dplyrmagrittr-pipelines/) [\(http://www.onthelambda.com/2015/01/23/assertive-r-programming-in-dplyrmagrittr-pipelines/\)](http://www.onthelambda.com/2015/01/23/assertive-r-programming-in-dplyrmagrittr-pipelines/)
- [What does Flatland have to do with Haskell?](http://www.onthelambda.com/2014/12/22/what-does-flatland-have-to-do-with-haskell/) [\(http://www.onthelambda.com/2014/12/22/what-does-flatland-have-to-do-with-haskell/\)](http://www.onthelambda.com/2014/12/22/what-does-flatland-have-to-do-with-haskell/)
- [Sending text messages at random times using python](http://www.onthelambda.com/2014/11/15/sending-text-messages-at-random-times-using-python/) [\(http://www.onthelambda.com/2014/11/15/sending-text-messages-at-random-times-using-python/\)](http://www.onthelambda.com/2014/11/15/sending-text-messages-at-random-times-using-python/)
- [Why is my OS X Yosemite install taking so long?: an analysis](http://www.onthelambda.com/2014/10/23/why-is-my-os-x-yosemite-install-taking-so-long-an-analysis/) [\(http://www.onthelambda.com/2014/10/23/why-is-my-os-x-yosemite-install-taking-so-long-an-analysis/\)](http://www.onthelambda.com/2014/10/23/why-is-my-os-x-yosemite-install-taking-so-long-an-analysis/)
- [Fun with .Rprofile and customizing R startup](http://www.onthelambda.com/2014/09/17/fun-with-rprofile-and-customizing-r-startup/) [\(http://www.onthelambda.com/2014/09/17/fun-with-rprofile-and-customizing-r-startup/\)](http://www.onthelambda.com/2014/09/17/fun-with-rprofile-and-customizing-r-startup/)
- [Interactive visualization of non-linear logistic regression decision boundaries with Shiny](http://www.onthelambda.com/2014/07/24/interactive-visualization-of-non-linear-logistic-regression-decision-boundaries-with-shiny/) [\(http://www.onthelambda.com/2014/07/24/interactive-visualization-of-non-linear-logistic-regression-decision-boundaries-with-shiny/\)](http://www.onthelambda.com/2014/07/24/interactive-visualization-of-non-linear-logistic-regression-decision-boundaries-with-shiny/)
- [Squeezing more speed from R for nothing. Rcpp style](http://www.onthelambda.com/2014/06/27/squeezing-more-speed-from-r-for-nothing-rcpp-style/) [\(http://www.onthelambda.com/2014/06/27/squeezing-more-speed-from-r-for-nothing-rcpp-style/\)](http://www.onthelambda.com/2014/06/27/squeezing-more-speed-from-r-for-nothing-rcpp-style/)
- [Damn the torpedoes, full speed ahead: making the switch to Python 3](http://www.onthelambda.com/2014/05/13/damn-the-torpedoes-full-speed-ahead-making-the-switch-to-python-3/) [\(http://www.onthelambda.com/2014/05/13/damn-the-torpedoes-full-speed-ahead-making-the-switch-to-python-3/\)](http://www.onthelambda.com/2014/05/13/damn-the-torpedoes-full-speed-ahead-making-the-switch-to-python-3/)
- [Take a look, it's in a book: distribution of kindle e-book highlights](http://www.onthelambda.com/2014/04/10/take-a-look-its-in-a-book-distribution-of-kindle-e-book-highlights/) [\(http://www.onthelambda.com/2014/04/10/take-a-look-its-in-a-book-distribution-of-kindle-e-book-highlights/\)](http://www.onthelambda.com/2014/04/10/take-a-look-its-in-a-book-distribution-of-kindle-e-book-highlights/)
- [How to make an absurd twitter bot in python](http://www.onthelambda.com/2014/03/20/how-to-make-an-absurd-twitter-bot-in-python/) [\(http://www.onthelambda.com/2014/03/20/how-to-make-an-absurd-twitter-bot-in-python/\)](http://www.onthelambda.com/2014/03/20/how-to-make-an-absurd-twitter-bot-in-python/)
- [How to fake a sophisticated knowledge of wine with Markov Chains](http://www.onthelambda.com/2014/02/20/how-to-fake-a-sophisticated-knowledge-of-wine-with-markov-chains/) [\(http://www.onthelambda.com/2014/02/20/how-to-fake-a-sophisticated-knowledge-of-wine-with-markov-chains/\)](http://www.onthelambda.com/2014/02/20/how-to-fake-a-sophisticated-knowledge-of-wine-with-markov-chains/)

- ⊙ [How dplyr replaced my most common R idioms \(http://www.onthelambda.com/2014/02/10/how-dplyr-replaced-my-most-common-r-idioms/\)](http://www.onthelambda.com/2014/02/10/how-dplyr-replaced-my-most-common-r-idioms/)
- ⊙ [Using Last.fm to data mine my music listening history \(http://www.onthelambda.com/2014/01/27/using-last-fm-to-data-mine-my-music-listening-history/\)](http://www.onthelambda.com/2014/01/27/using-last-fm-to-data-mine-my-music-listening-history/)
- ⊙ [The performance gains from switching R's linear algebra libraries \(http://www.onthelambda.com/2013/12/26/the-performance-gains-from-switching-rs-linear-algebra-libraries/\)](http://www.onthelambda.com/2013/12/26/the-performance-gains-from-switching-rs-linear-algebra-libraries/)
- ⊙ [Visualizing data analysis pipelines using NetworkX \(http://www.onthelambda.com/2013/12/08/visualizing-data-analysis-pipelines-using-networkx/\)](http://www.onthelambda.com/2013/12/08/visualizing-data-analysis-pipelines-using-networkx/)
- ⊙ [Compiling R from source and why you shouldn't do it \(http://www.onthelambda.com/2013/11/22/compiling-r-from-source-and-why-you-shouldnt-do-it/\)](http://www.onthelambda.com/2013/11/22/compiling-r-from-source-and-why-you-shouldnt-do-it/)
- ⊙ [Parallel R \(and air travel\) \(http://www.onthelambda.com/2013/11/13/parallel-r-and-air-travel/\)](http://www.onthelambda.com/2013/11/13/parallel-r-and-air-travel/)
- ⊙ [qstats - quick and dirty statistics tool for the Unix pipeline \(http://www.onthelambda.com/2013/11/05/qstats-quick-and-dirty-statistics-tool-for-the-unix-pipeline/\)](http://www.onthelambda.com/2013/11/05/qstats-quick-and-dirty-statistics-tool-for-the-unix-pipeline/)
- ⊙ [Linguistics, meet Evolutionary Biology \(http://www.onthelambda.com/2013/10/29/linguistics-meet-evolutionary-biology/\)](http://www.onthelambda.com/2013/10/29/linguistics-meet-evolutionary-biology/)
- ⊙ [Unsupervised correction of optical character misrecognition \(http://www.onthelambda.com/2013/10/22/unsupervised-correction-of-optical-character-misrecognition/\)](http://www.onthelambda.com/2013/10/22/unsupervised-correction-of-optical-character-misrecognition/)
- ⊙ [The state of package management on Mac OS X \(http://www.onthelambda.com/2013/10/14/the-state-of-package-management-on-mac-os-x/\)](http://www.onthelambda.com/2013/10/14/the-state-of-package-management-on-mac-os-x/)
- ⊙ [On the treachery of point-and-click "black-box" data analysis \(http://www.onthelambda.com/2013/10/07/on-the-treachery-of-point-and-click-black-box-data-analysis/\)](http://www.onthelambda.com/2013/10/07/on-the-treachery-of-point-and-click-black-box-data-analysis/)
- ⊙ [On the misinterpretation of p-values: \(http://www.onthelambda.com/2013/10/02/on-the-misinterpretation-of-p-values/\)](http://www.onthelambda.com/2013/10/02/on-the-misinterpretation-of-p-values/)

© 2015 On the lambda. All rights reserved.

[Flato \(http://www.themememe.com/flato\)](http://www.themememe.com/flato) by ThemeMeme