Regular guys talk regularization

What we'll talk aboout

- Why regularization
- What is it?
- How do I do it?
- Tweedie
- Mathy stuff
- Conclusion

Why?

Why?

Let’s start with some data and a model

Data

- *dataOhlsson* from *insuranceData* R package
- Swedish motorcycle insurance from Wasa, 1994 to 1998
- We've renamed variables to English

```
library(insuranceData)
data("dataOhlsson")

# Drop the claim count variable
tbl_ohlsson <- dataOhlsson %>%
  select(
    age_number = agarald
    , territory = zon
    , motor_class = mcklass
    , vehicle_age = fordald
    , bonus_class = bonuskl
    , duration
    , losses = skadkost)
```

```r
str(tbl_ohlsson)
```

```
## 'data.frame':	64548 obs. of  7 variables:
##  $ age_number : int  0 4 5 5 6 9 9 9 10 10 ...
##  $ territory  : int  1 3 3 4 2 3 4 4 2 4 ...
##  $ motor_class: int  4 6 3 1 1 3 3 4 3 2 ...
##  $ vehicle_age: int  12 9 18 25 26 8 6 20 16 17 ...
##  $ bonus_class: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ duration   : num  0.175 0 0.455 0.173 0.181 ...
##  $ losses     : int  0 0 0 0 0 0 0 0 0 0 ...
```

Fit a model

```
fit_ols <- lm(
  losses ~ .
  , data = tbl_ohlsson
)
```

Which coefficient should we drop?

```r
fit_ols %>%
  summary()
```

```
##
## Call:
## lm(formula = losses ~ ., data = tbl_ohlsson)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3231   -475   -273    -61 365055
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1179.278     95.404  12.361  < 2e-16 ***
## age_number   -11.948      1.461  -8.176 2.98e-16 ***
## territory   -114.157     13.731  -8.314  < 2e-16 ***
## motor_class    3.827     12.714   0.301   0.7634
## vehicle_age  -17.788      1.968  -9.040  < 2e-16 ***
## bonus_class   17.914      8.140   2.201   0.0278 *
## duration      94.150     14.443   6.519 7.13e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4684 on 64541 degrees of freedom
## Multiple R-squared:  0.004616,	Adjusted R-squared:  0.004523
## F-statistic: 49.88 on 6 and 64541 DF,  p-value: < 2.2e-16
```

Options

- Manual selection based on standard error of coefficients
- Stepwise regression
- Feature engineering
- PCA
- Partial least squares
- Or ...

Regularization!

Benefits:

- "Curse of dimensionality" number of observations not much larger than p
- No p-hacking
- Let the model pick your variables!
- Reduce chance that the model will overfit
- Collinearity

What

What is regularization

Regluarization adjusts the cost function which creates the model

The *what* function?

- Models map data (predictors) to other data (target variable)
- The preferred model is one which optimizes some *cost* of model output
- OLS cost -> least squares
- GLM -> maximum likelihood/residual deviance
- Regularization augments OLS/GLM with a penalty based on the magnitude of the coefficients

OLS cost function

$$\sum_{i}^{n}\left(y_i - \hat{y}_i\right)^2 = \sum_{i}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j}^{p}\left(x_{ij} * \hat{\beta}_j\right)\right)^2 = RSS$$

Regularization cost function

$$\sum_{i}^{n}\left(y_i-\hat{\beta}_0-\sum_{j}^{p}\left(x_{ij}*\hat{\beta}_j\right)\right)^2+\lambda\sum_{j}^{p}\|\hat{\beta}_j\|_L=RSS+\lambda\sum_{j}^{p}\|\hat{\beta}_j\|_L$$

Two cost functions

OLS

$$\sum_{i}^{n}\left(y_i - \hat{y}_i\right)^2 = \sum_{i}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j}^{p}\left(x_{ij} * \hat{\beta}_j\right)\right)^2 = RSS$$

Regularization:

$$\sum_{i}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j}^{p}\left(x_{ij} * \hat{\beta}_j\right)\right)^2 + \lambda \sum_{j}^{p} \|\hat{\beta}_j\|_L = RSS + \lambda \sum_{j}^{p} \|\hat{\beta}_j\|_L$$

Overfitting and the role of λ

Analogue to credibility. λ applies a shrinkage to the parameters. The "complement" is the intercept.

Same idea: reduce variance on out of sample data.

Control weight given to predictors (i.e. $\hat{\beta}_j$), in favor of $\hat{\beta}_0$.

L?

$$\sum_{j}^{p} \|\hat{\beta}_j\|_L$$

$$L = 1 \implies \sum_{j}^{p} |\hat{\beta}_j|$$

$$L = 2 \implies \sum_{j}^{p} \hat{\beta}_j^{\,2}$$

L

- Can L be higher than 2?
- Must L be an integer?

L1 and L2 norms

L1 = **L**east **A**bsolute **S**hrinkage and **S**election **O**perator = LASSO

$$L = 1 \implies RSS + \lambda \sum_{j}^{p} |\beta_j|$$

L2 = Ridge regression

$$L = 2 \implies RSS + \lambda \sum_{j}^{p} {\beta_j}^2$$

How

Easy answer

Use `glmnet`

```
mat_ohlsson <- tbl_ohlsson %>%
  select(-losses) %>%
  as.matrix()

library(glmnet)
fit_ridge <- glmnet(
  x = mat_ohlsson
  , y = tbl_ohlsson$losses
  , family = 'gaussian'
  , alpha = 0
# , lambda = seq()
)
```

About `alpha`

Used to mix Ridge and Lasso

$(1-\alpha)/2\|\beta\|_2^2 + \alpha\|\beta\|_1$

$\alpha = 0 \implies Ridge$

$\alpha = 1 \implies Lasso$

What does `glmnet` do?

1. Standardize the predictor space (unless you tell it not to)
2. Form a set of candidate λ's (unless you provide your own)
3. Fit coefficients for each λ

We should:

1. Use cross validation to measure RMSE (or other metric) on out of sample (test) data
2. Pick the λ which optimizes out of sample predictions

Standardize predictors

- Why?
- OLS is scale-invariant, regularization isn't
- Extreme(ish) case: convert currency
- glmnet returns coefficients at the original scale.

```
fit_ols %>%
  summary()
```

```
##
## Call:
## lm(formula = losses ~ ., data = tbl_ohlsson)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3231   -475   -273    -61 365055
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1179.278     95.404  12.361  < 2e-16 ***
## age_number   -11.948      1.461  -8.176 2.98e-16 ***
## territory   -114.157     13.731  -8.314  < 2e-16 ***
## motor_class    3.827     12.714   0.301   0.7634
## vehicle_age  -17.788      1.968  -9.040  < 2e-16 ***
## bonus_class   17.914      8.140   2.201   0.0278 *
## duration      94.150     14.443   6.519 7.13e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4684 on 64541 degrees of freedom
## Multiple R-squared:  0.004616,   Adjusted R-squared:  0.004523
## F-statistic: 49.88 on 6 and 64541 DF,  p-value: < 2.2e-16
```

Fit using many different λ's

Log

Lambda

Coefficients

What does lasso look like?

```
fit_lasso <- glmnet(
    x = mat_ohlsson
  , y = tbl_ohlsson$losses
  , family = 'gaussian'
  , alpha = 1
)
```

What does lasso look like?

Log
Lambda

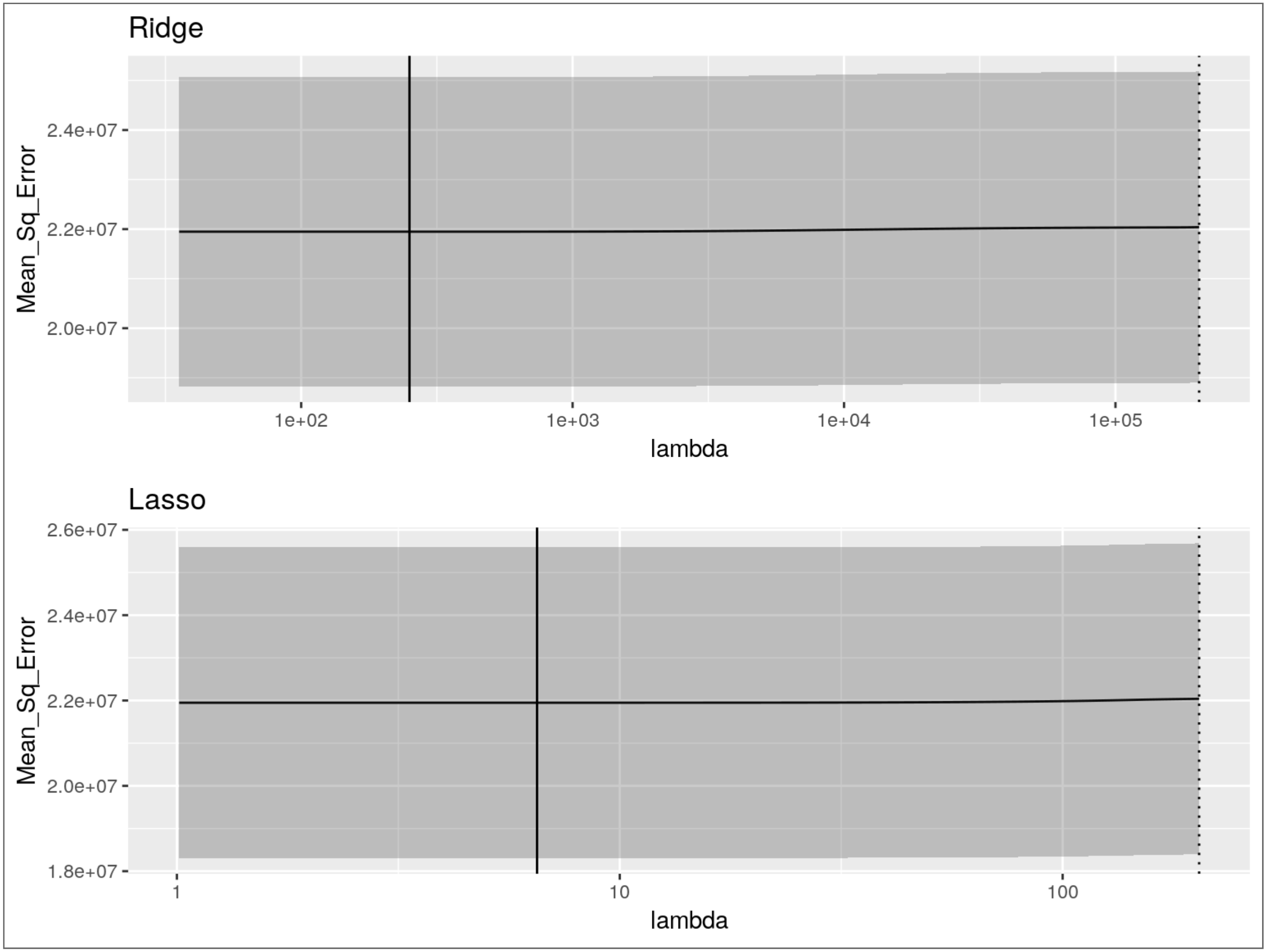
Coefficients

Use cross validation to measure RMSE (or other metric) on out of sample (test) data

```
fit_ridge_cv <- cv.glmnet(
    x = mat_ohlsson
  , y = tbl_ohlsson$losses
  , family = 'gaussian'
  , alpha = 0
  , nfolds = 10
# , foldid = NULL
)

fit_lasso_cv <- cv.glmnet(
    x = mat_ohlsson
  , y = tbl_ohlsson$losses
  , family = 'gaussian'
  , alpha = 1
  , nfolds = 10
)
```

What λ to pick?



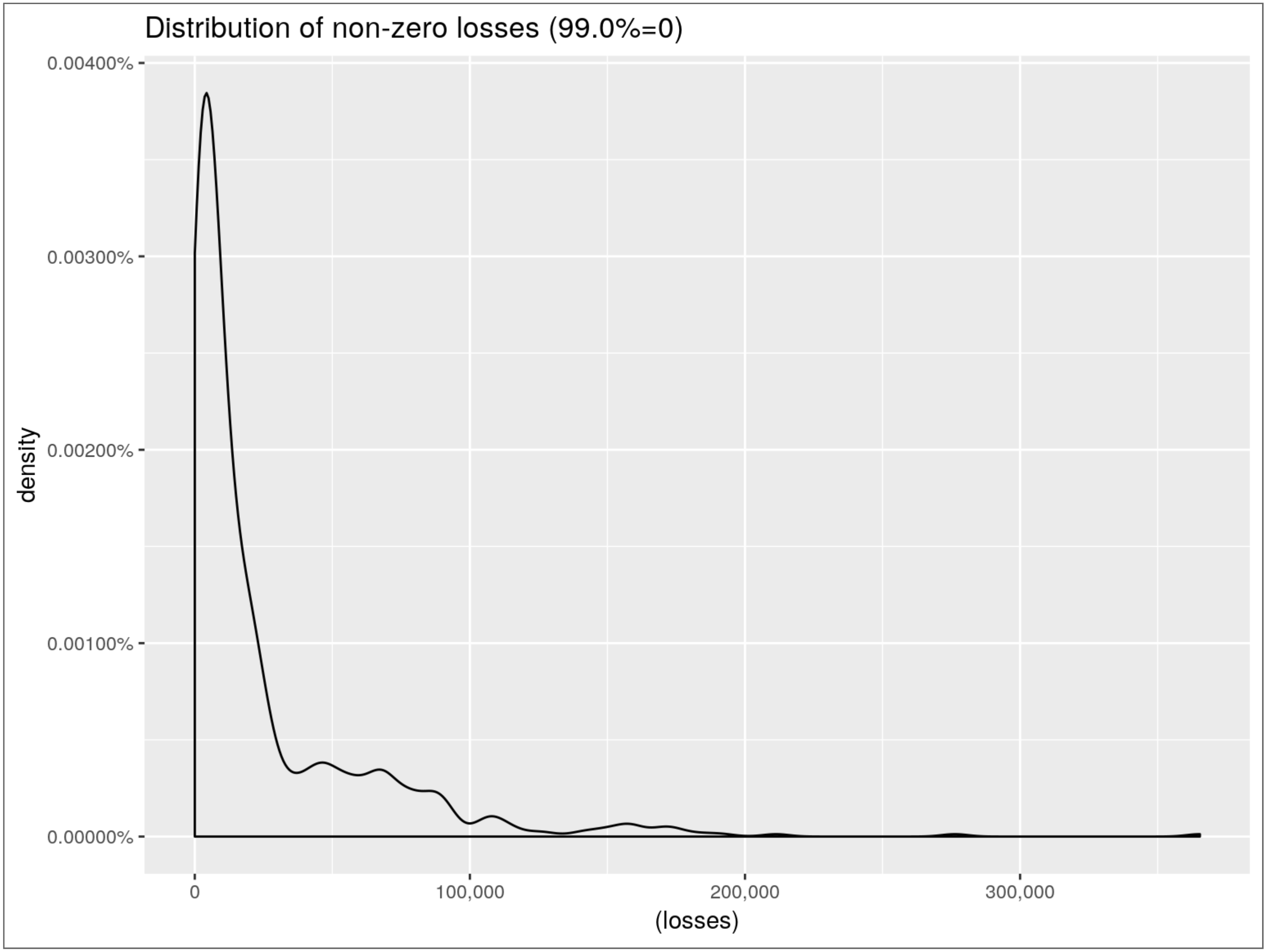
Pick λ to optimize OoS prediction

```r
ridge_lambda_select<-fit_ridge_cv$lambda.min
lasso_lambda_select<-fit_lasso_cv$lambda.min
selected_coef_gauss<-
    data.frame(as.matrix(coef(fit_ridge_cv,s=ridge_lambda_select))
           ,as.matrix(coef(fit_lasso_cv,s=lasso_lambda_select)))
names(selected_coef_gauss)<-c("Ridge","Lasso")
```

	Variable	Ridge	Lasso
1	(Intercept)	1127.43	1160.18
2	age_number	-11.32	-11.41
3	territory	-108.83	-109.49
4	motor_class	5.63	1.25
5	vehicle_age	-16.96	-17.38
6	bonus_class	17.11	15.59
7	duration	88.64	89.21

The Tweedie distribution

OLS but not so ordinary



One curve to rule them all

- Tweedie family contains any distribution that satisfies r: $Variance = \phi * \mu^p$
- This includes
- Normal: $p = 0$
- Poisson: $p = 1$
- Compound Gamma/Poisson: $1 < p < 2$
- Gamma: $p = 2$
- Inverse Gaussian: $p = 3$
- Generally no closed form.

GLM

```r
library(statmod)

fit_glm <- glm(
  losses ~ .
  , family = tweedie(var.power = 1.5, link.power = 0)
  , data = tbl_ohlsson
)

summary(fit_glm)
```

```
##
## Call:
## glm(formula = losses ~ ., family = tweedie(var.power = 1.5, link.power = 0),
##     data = tbl_ohlsson)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -24.37   -8.28   -6.64   -5.27  384.42
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.67935    0.75445  11.504  < 2e-16 ***
## age_number  -0.05368    0.01294  -4.149 3.35e-05 ***
## territory   -0.43013    0.11922  -3.608 0.000309 ***
## motor_class  0.06078    0.10692   0.569 0.569689
## vehicle_age -0.09108    0.02233  -4.078 4.55e-05 ***
## bonus_class  0.11721    0.07002   1.674 0.094134 .
## duration     0.14737    0.08492   1.735 0.082674 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for Tweedie family taken to be 20386.43)
##
##     Null deviance: 7708506  on 64547  degrees of freedom
```

HDTweedie

- Package is built on glmnet, with addition of the Tweedie family

```
library(HDtweedie)
fit_ridge_cv_tweedie <- cv.HDtweedie(
  x = mat_ohlsson
  , y = tbl_ohlsson$losses
  , p = 1.5
  , alpha = 0
  , lambda = seq(from=exp(0),to=exp(5),length.out = 100)
  , standardize=TRUE#
)

fit_lasso_cv_tweedie <- cv.HDtweedie(
  x = mat_ohlsson
  , y = tbl_ohlsson$losses
  , p = 1.5
  , alpha = 1
  , lambda = seq(from=exp(-2),to=exp(3),length.out = 100)
  , standardize=TRUE
)
```

Tweedie Ridge Path

```
## Warning: `as.tibble()` is deprecated, use `as_tibble()` (but mind the new semant
## This warning is displayed once per session.
```

Log Lambda

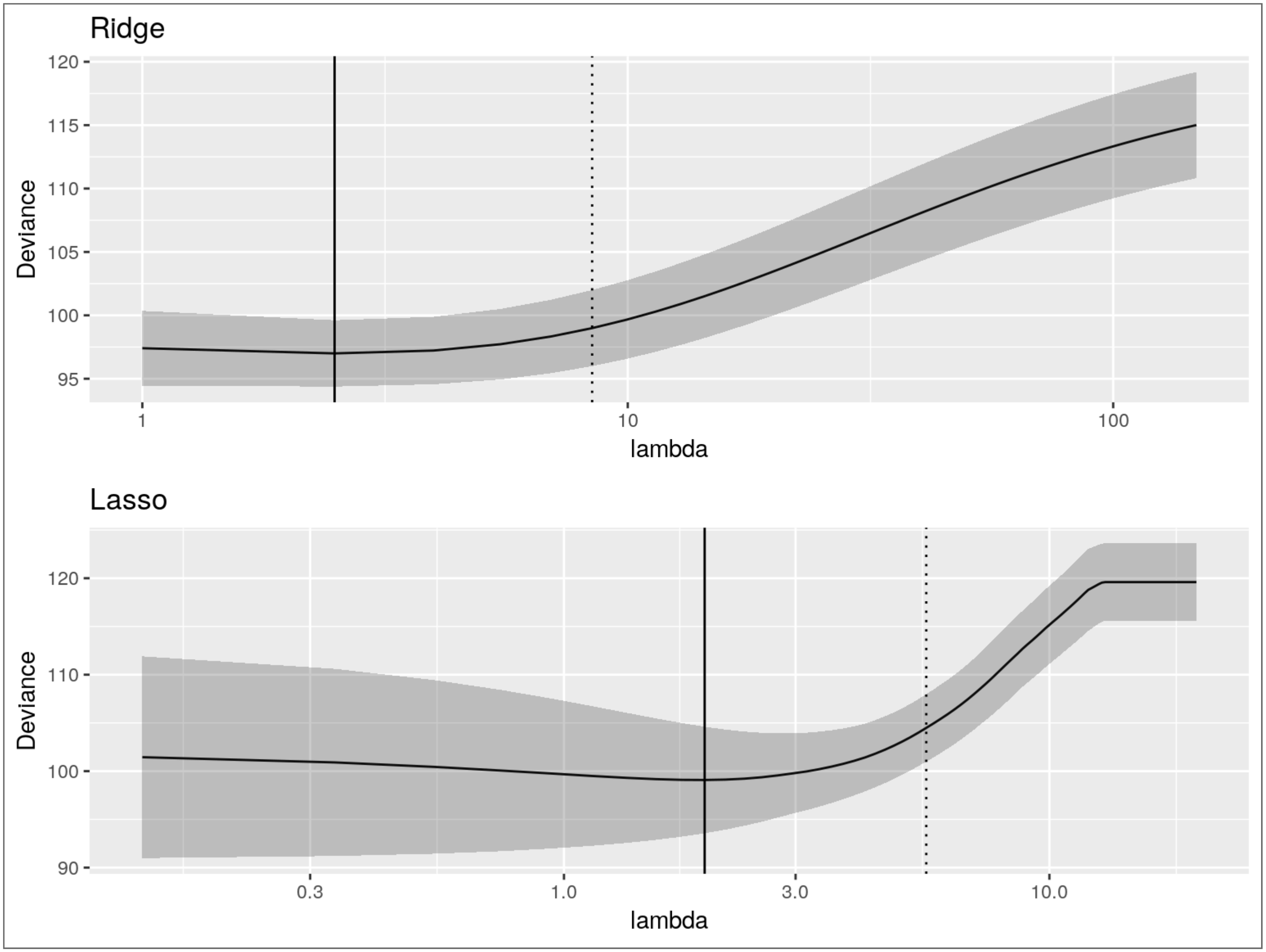
Coefficients

And Lasso...

Log Lambda

Coefficients

The Tweedie CV Plot



What λ to pick now?

```
ridge_lambda_select<-fit_ridge_cv_tweedie$lambda.min
lasso_lambda_select<-fit_lasso_cv_tweedie$lambda.min
selected_coef_tweed<-
    data.frame(as.matrix(coef(fit_ridge_cv_tweedie,s=ridge_lambda_select))
          ,as.matrix(coef(fit_lasso_cv_tweedie,s=lasso_lambda_select)))
names(selected_coef_tweed)<-c("Ridge","Lasso")
```

	Variable	Ridge	Lasso
1	(Intercept)	8.10	8.29
2	age_number	-0.04	-0.04
3	territory	-0.37	-0.32
4	motor_class	0.07	0.00
5	vehicle_age	-0.08	-0.08
6	bonus_class	0.09	0.05
7	duration	0.13	0.10

Let’s compare across families

```r
combind_coef<-cbind(selected_coef_gauss,selected_coef_tweed)
names(combind_coef)<-c("Ridge_Gauss","Lasso_Gauss","Ridge_Tweedie","Lasso_Tweedie")
```

	Variable	Ridge_Gauss	Ridge_Tweedie	Lasso_Gauss	Lasso_Twe
1	(Intercept)	1127.43	8.10	1160.18	8
2	age_number	-11.32	-0.04	-11.41	-0
3	territory	-108.83	-0.37	-109.49	-0
4	motor_class	5.63	0.07	1.25	0
5	vehicle_age	-16.96	-0.08	-17.38	-0
6	bonus_class	17.11	0.09	15.59	0
7	duration	88.64	0.13	89.21	0

Mathy stuff

The formula again

$$\sum_{i}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j}^{p}\left(x_{ij} * \hat{\beta}_j\right)\right)^2 + \lambda \sum_{j}^{p} \|\hat{\beta}_j\|_L = RSS + \lambda \sum_{j}^{p} \|\hat{\beta}_j\|_L$$

Equivalently

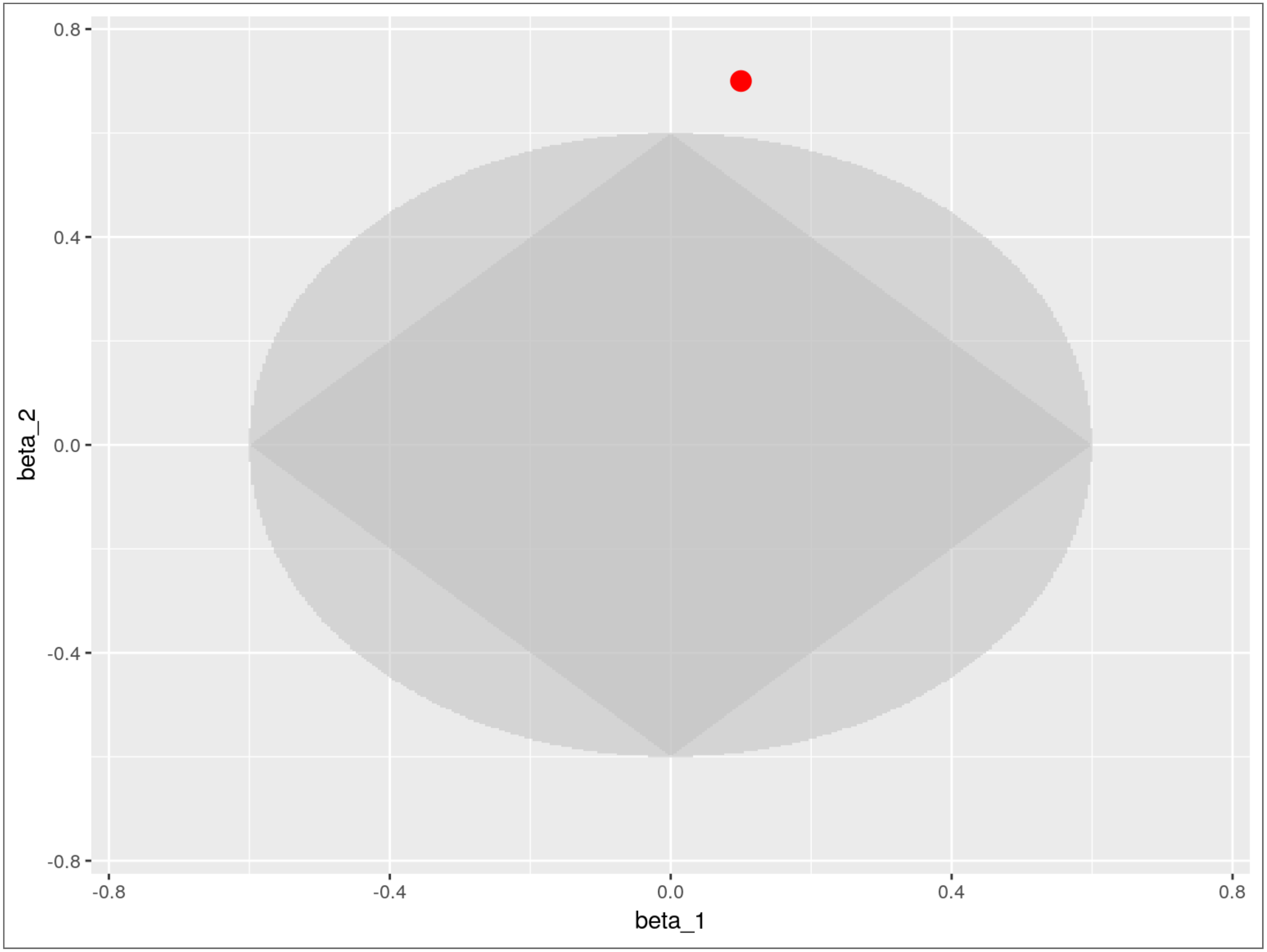
Maximize:

$$\sum_{i}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j}^{p}\left(x_{ij} * \hat{\beta}_j\right)\right)^2$$

Subject to:

$$\sum_{j}^{p} \|\hat{\beta}_j\|_L \leq t$$

Shrink or vanish



The formulaic way of saying that

$$\beta = \frac{2x_i y_i - \lambda}{2x_i^2}$$

$$\beta = \frac{2x_i y_i}{2x_i^2 + 2\lambda}$$

L0

Subject to:

$$\sum_{j}^{p} \|\hat{\beta}_j\|_0 = \sum_{j}^{p} I(\beta_j \neq 0) \leq t$$

No more than *t* coefficients are not zero -> best subset.

Collinearity

If we know both are important, we may not want to choose:

L1/LASSO pushes things to zero.

L2/Ridge restricts the size, but keeps both.

Bayesian link

L1 = Bayes with Laplace prior

L2 = Bayes with normal priors

$$\prod_{1}^{N} \Phi(y_n|\beta x_n, \sigma^2)\Phi(\beta|0, \lambda^{-1})$$

Conclusion

Conclusion

- Option to consider for high-dimension data
- Choice of hyperparameter needs a fair bit of data
- `glmnet` package or `HDtweedie`

Thank you!

References

- **http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf**
- **https://web.stanford.edu/~hastie/Papers/ESLII.pdf**
- **https://stats.stackexchange.com/questions/163388/l2-regularization-is-equivalent-to-gaussian-prior**
- **https://github.com/PirateGrunt/intro_regularization**