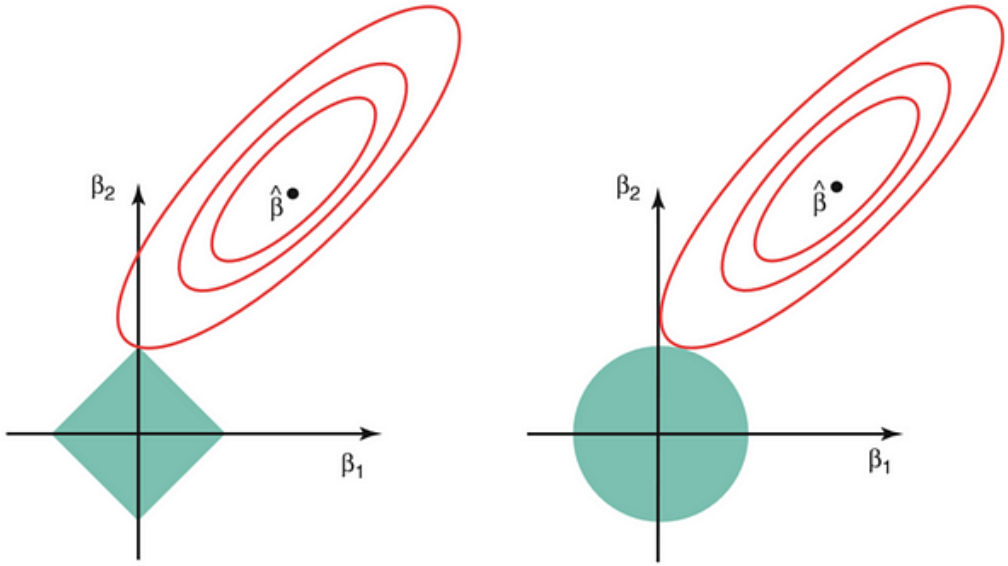
On {glmnet}

PSY-GS 8875 Behavioral Data Science



On {glmnet}

Set up Data

```r
# Load packages
library(glmnet)

# Set seed for reproduciblity
set.seed(42)

# Load data
load("../../data/ncds/ncds_sample.RData")

# Set up our X and Ys
X <- as.matrix(ncds_sample[,2:51]) # 50 personality items
Y <- as.matrix(ncds_sample["wem_well_being"]) # well-being

# Update X and Y
keep <- complete.cases(X) & !is.na(Y)
X <- X[keep,]; Y <- Y[keep,, drop = FALSE]
```

Unstandardized Coefficients

Unstandardized Analytic Ridge Coefficients

```
# Add intercept
X_intercept <- cbind(1, X)

# Set up penalization matrix
lambda <- 10
ridge_matrix <- lambda * diag(ncol(X_intercept))
ridge_matrix[1, 1] <- 0 # don't regularize the intercept

# Matrix algebra computation
analytical <- solve(
  crossprod(X_intercept) + ridge_matrix
) %*% crossprod(X_intercept, Y)
```

Unstandardized Analytic Ridge Coefficients

	B
I spend time reflecting on things	0.0385498
I am quiet around strangers	0.0192946
I make people feel at ease	0.2232042
I am exacting in my work	-0.1417021
I often feel blue	-1.3504913
I am full of ideas	1.3169125

Unstandardized {glmnet} Ridge Coefficients

```
# Compute ridge regression
glmnet_unstandardized <- glmnet(
  x = X, y = Y, family = "gaussian",
  alpha = 0, # 0 = ridge; 1 = lasso
  lambda = 10, # penalty parameter
  thresh = 1e-20, # high precision accuracy
  standardize = FALSE
)
```

Unstandardized {glmnet} Ridge Coefficients

	$B_{analytical}$	$B_{\{glmnet\}}$
I spend time reflecting on things	0.0385498	0.0394195
I am quiet around strangers	0.0192946	-0.0364021
I make people feel at ease	0.2232042	0.1911952
I am exacting in my work	-0.1417021	0.0838487
I often feel blue	-1.3504913	-0.7813674
I am full of ideas	1.3169125	0.6209150

Scaling {glmnet} Ridge Coefficients

```
# Get {glmnet} unstandardized coefficients
scaled_glmnet_unstandardized <- coef(
  glmnet_unstandardized, x = X, y = Y,
  s = sd(Y) * 10 / nrow(Y), # use lambda
  exact = TRUE
)
```

Unstandardized {glmnet} Ridge Coefficients

	$B_{analytical}$	$B_{\{glmnet\}}$
I spend time reflecting on things	0.0385498	0.0385523
I am quiet around strangers	0.0192946	0.0192970
I make people feel at ease	0.2232042	0.2232031
I am exacting in my work	-0.1417021	-0.1416859
I often feel blue	-1.3504913	-1.3504425
I am full of ideas	1.3169125	1.3168534

Close! But still not quite...

Unstandardized {glmnet} Ridge Coefficients

$$\sqrt{\frac{\sum_{i=1}^{N}(y_i - \bar{y})^2}{N}}$$

```
# Use {glmnet}'s standard deviation
glmnet_sd <- sqrt(sum((Y - mean(Y))^2) / nrow(Y))

# Get scaled coefficients
scaled_glmnet_unstandardized <- coef(
  glmnet_unstandardized,
  x = X, y = Y,
  s = glmnet_sd * 10 / nrow(Y), # use lambda
  exact = TRUE
)
```

Unstandardized {glmnet} Ridge Coefficients

	$B_{analytical}$	$B_{\{glmnet\}}$
I spend time reflecting on things	0.0385498	0.0385498
I am quiet around strangers	0.0192946	0.0192946
I make people feel at ease	0.2232042	0.2232042
I am exacting in my work	-0.1417021	-0.1417021
I often feel blue	-1.3504913	-1.3504913
I am full of ideas	1.3169125	1.3169125

Standardized Coefficients

Z-score X and Y

```
# Scale X
X_scaled <- scale(X)
Y_scaled <- scale(Y)

# Add intercept
X_scaled_intercept <- cbind(1, X_scaled)

# Matrix algebra computation
analytical_scaled <- solve(
  crossprod(X_scaled_intercept) + ridge_matrix # doesn't change
) %*% crossprod(X_scaled_intercept, Y_scaled)
```

Standardized Analytic Ridge Coefficients

	β
I spend time reflecting on things	0.0040782
I am quiet around strangers	0.0026509
I make people feel at ease	0.0237298
I am exacting in my work	-0.0169081
I often feel blue	-0.1923352
I am full of ideas	0.1529323

Standardized {glmnet} Ridge Coefficients

```
# Compute ridge regression
glmnet_standardized <- glmnet(
  x = X, y = Y, family = "gaussian",
  alpha = 0, # 0 = ridge; 1 = lasso
  lambda = 10, # penalty parameter
  thresh = 1e-20, # high precision accuracy
  standardize = TRUE # default
)
```

Standardized {glmnet} Ridge Coefficients

	$\beta_{analytical}$	$\beta_{\{glmnet\}}$
I spend time reflecting on things	0.0040782	0.0063054
I am quiet around strangers	0.0026509	-0.0489420
I make people feel at ease	0.0237298	0.2109023
I am exacting in my work	-0.0169081	0.0803031
I often feel blue	-0.1923352	-0.6860122
I am full of ideas	0.1529323	0.6347169

Here we go again. . .

Scaling {glmnet} Ridge Coefficients

```
# Get {glmnet} standardized coefficients
scaled_glmnet_standardized <- coef(
  glmnet_standardized,
  x = X_scaled, y = Y_scaled,
  s = 10 / nrow(Y), # use lambda
  exact = TRUE
)
```

Standardized {glmnet} Ridge Coefficients

	$\beta_{analytical}$	$\beta_{\{glmnet\}}$
I spend time reflecting on things	0.0040782	0.0040783
I am quiet around strangers	0.0026509	0.0026508
I make people feel at ease	0.0237298	0.0237302
I am exacting in my work	-0.0169081	-0.0169089
I often feel blue	-0.1923352	-0.1923447
I am full of ideas	0.1529323	0.1529392

Close! But still not quite...

Standardized {glmnet} Ridge Coefficients

We need to change the standardization of **Y** to get them to match. . .

```
# Scale Y the {glmnet} way
Y_scaled_glmnet <- (Y - mean(Y)) / glmnet_sd

# Matrix algebra computation
analytical_scaled <- solve(
  crossprod(X_scaled_intercept) + ridge_matrix # doesn't change
) %*% crossprod(X_scaled_intercept, Y_scaled_glmnet)

# Compute ridge regression
glmnet_standardized <- glmnet(
  x = X_scaled, y = Y_scaled_glmnet, family = "gaussian",
  alpha = 0, # 0 = ridge; 1 = lasso
  lambda = 10, # penalty parameter
  thresh = 1e-20, # high precision accuracy
  standardize = FALSE # switch to `FALSE` !!
)
```

Standardized {glmnet} Ridge Coefficients

	$\beta_{analytical}$	$\beta_{\{glmnet\}}$
I spend time reflecting on things	0.0040835	0.0040835
I am quiet around strangers	0.0026544	0.0026544
I make people feel at ease	0.0237608	0.0237608
I am exacting in my work	-0.0169302	-0.0169302
I often feel blue	-0.1925868	-0.1925868
I am full of ideas	0.1531324	0.1531324

On {glmnet}

What on Earth is {glmnet} doing?!

- Standardizes **Y** by default using the population standard deviation
- Scales coefficients based on standard deviation of **Y**, λ, and number of cases
- By default, {glmnet} will...
  - standardize **X** and **Y** using the population standard deviation
  - convert coefficients to *unstandardized* values for interpretability