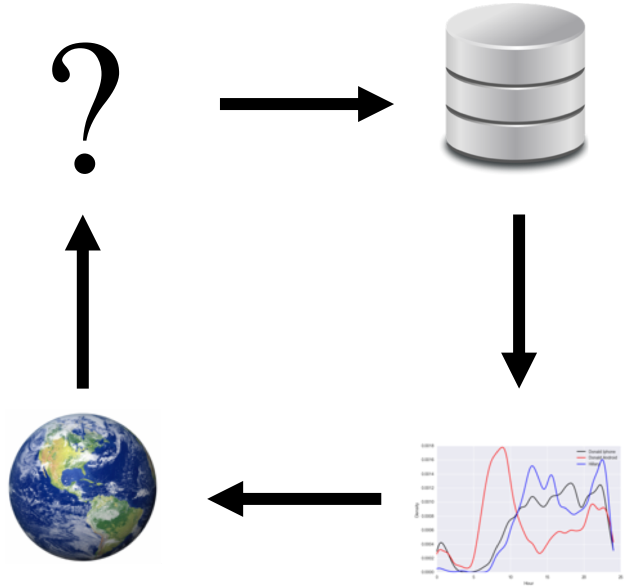
Data Science 100

Lecture 16:
Probability
Prediction
Dummy Variables



Probability Model & Expected Loss

Simple Linear Probability Model

Tilde denotes the true parameter values

Epsilon is random noise

$$Y = \tilde{\beta}_0 + \tilde{\beta}_1 x + \epsilon$$

Capital Y denotes a random variable

Treat x as given (conditional)

Simple Linear Probability Model

Epsilon is
random noise

$$Y = \tilde{\beta}_0 + \tilde{\beta}_1 x + \epsilon$$

$$\mathbb{E}(\epsilon) = 0$$

Errors have no trend
They do not depend on x or beta

$$\mathbb{V}ar(\epsilon) = \sigma^2$$

The size of the errors have no trend
They do not depend on x or beta

Simple Linear Probability Model

$$Y_i = \underbrace{\tilde{\beta}_0 + \tilde{\beta}_1 x_i}_{\text{Constant}} + \underbrace{\epsilon_i}_{\text{Random Variable}} \qquad i = 1, 2, \ldots, n$$

$$\mathbb{E}(Y_i) = \mathbb{E}(\tilde{\beta}_0 + \tilde{\beta}_1 x_i + \epsilon_i)$$

Expectation is Conditional on x

$$= \tilde{\beta}_0 + \tilde{\beta}_1 x_i + \mathbb{E}(\epsilon_i)$$

$$= \tilde{\beta}_0 + \tilde{\beta}_1 x_i$$

Property of expectation

$$E(c + dZ) = c + dE(Z)$$

Simple Linear Probability Model

$$Y_i = \overbrace{\tilde{\beta}_0 + \tilde{\beta}_1 x_i}^{\text{Constant}} + \overbrace{\epsilon_i}^{\text{Random Variable}} \quad i = 1, 2, \ldots, n$$

$$\mathbb{V}ar(Y_i) = \mathbb{V}ar(\tilde{\beta}_0 + \tilde{\beta}_1 x_i + \epsilon_i)$$

Expectation is Conditional on x

$$= \mathbb{V}ar(\epsilon_i)$$

$$= \sigma^2$$

Property of variance

$$Var(c + dZ) = d^2 Var(Z)$$

L_2 Risk Minimization

If our goal is to predict Y, we can choose a prediction based on minimization of risk (expected loss)

$$\min_{\beta_0,\beta_1} \mathbb{E}[Y - (\beta_0 + \beta_1 x)]^2$$

Minimize Expected Square Error

Conditional on x

L_2 Risk

Conditional on x

$$\mathbb{E}[Y - (\beta_0 + \beta_1 x)]^2 = \mathbb{E}[\tilde{\beta}_0 + \tilde{\beta}_1 x + \epsilon - (\beta_0 + \beta_1 x)]^2$$

$$= \mathbb{E}[\epsilon]^2 + [\tilde{\beta}_0 - \beta_0 + \tilde{\beta}_1 x - \beta_1 x)^2$$

Since $(\tilde{\beta}_0 - \beta_0 + \tilde{\beta}_1 x - \beta_1 x)\mathbb{E}(\epsilon) = 0$

Minimized at $\tilde{\beta}_0, \tilde{\beta}_1$ the true parameters

Empirical Risk Minimization

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(Y_i - \bar{Y})}{\sum (x_i - \bar{x})^2}$$

How well do the parameters estimated from the data estimate the true parameter values?

Since

$$\mathbb{E}(Y_i) = \tilde{\beta}_0 + \tilde{\beta}_1 x_i$$

$$\mathbb{E}(\bar{Y}) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(Y_i)$$

$$= \tilde{\beta}_0 + \tilde{\beta}_1 \bar{x}$$

$$\mathbb{E}(Y_i - \bar{Y}) = \mathbb{E}(Y_i) - \mathbb{E}(\bar{Y})$$

$$= \tilde{\beta}_1 (x_i - \bar{x})$$

First we derive some useful expectations

$$\hat{\beta}_1 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2}$$

$$\mathbb{E}(\hat{\beta}_1) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})\mathbb{E}(Y_i - \bar{Y})}{\sum(x_i - \bar{x})^2}$$

$$= \tilde{\beta}_1 \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{\sum(x_i - \bar{x})^2}$$

$$= \tilde{\beta}_1$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$$

$$\mathbb{E}(\hat{\beta}_0) = \mathbb{E}(\bar{Y}) - \mathbb{E}(\hat{\beta}_1)\bar{x}$$

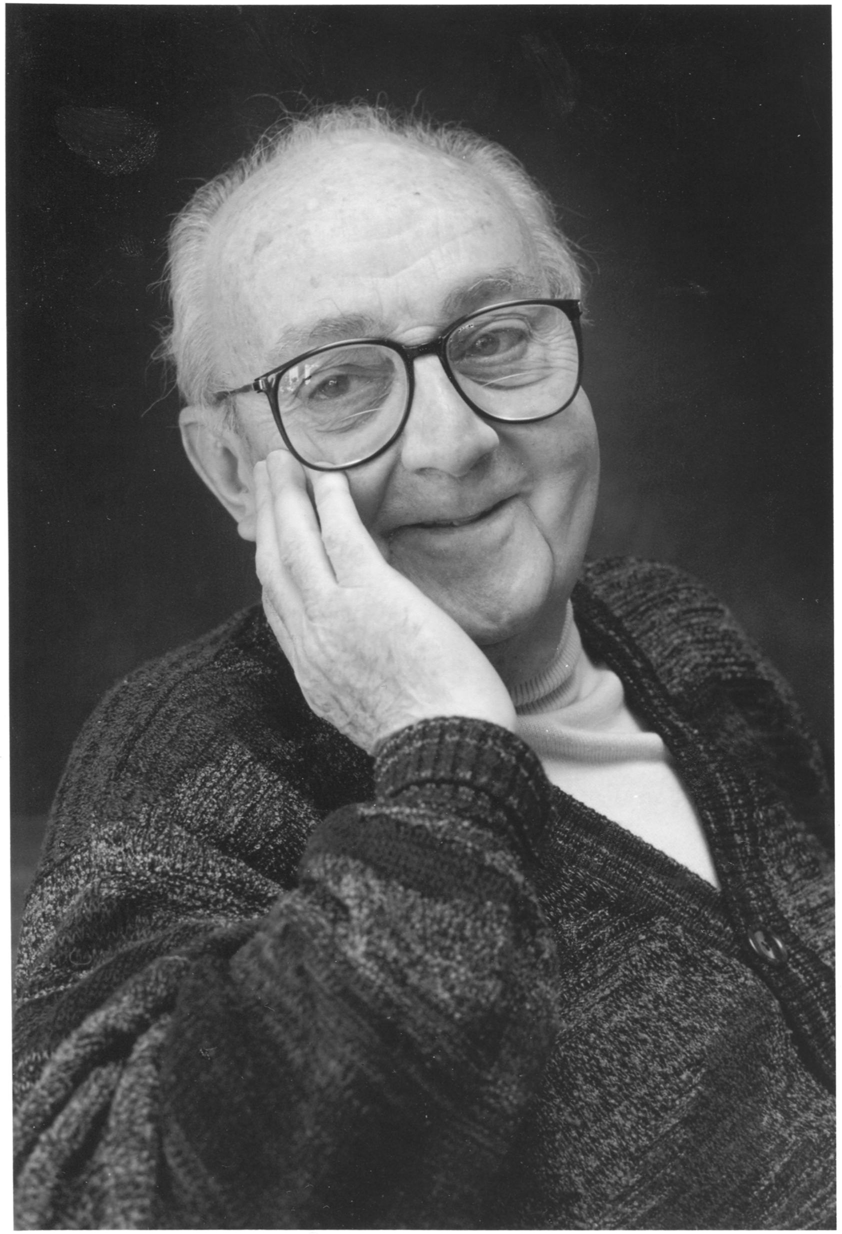
$$= \tilde{\beta}_0 + \tilde{\beta}_1 \bar{x} - \tilde{\beta}_1 \bar{x}$$

$$= \tilde{\beta}_0$$

If the linear model holds, then the least squares regression l parameters are unbiased.

Essentially,
all models are wrong,
but some are useful.

George Box



What happens when they are wrong? To Be Continued on Thursday

Data Science Life Cycle

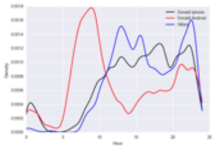
Context
Question
Refine Question to an one answerable with data

?

Design
Data Collection
Data Cleaning

Modeling
Test-train split
Loss function choice
Feature engineering
- Transformations,
- Dummy Variables

Model selection
- Best subset regression
- Cross-Validation



Model evaluation
Prediction error

Context

How to weigh a donkey in the Kenyan countryside, *Significance*, 2014, Milner and Rougier

Context

- Rural Kenya
- Donkeys very important for transport - crops, water, people, ploughing
- When donkeys fall sick, vets need to prescribe medicine
- Dosage depends on weight, but no scale in the countryside

1.8 million donkeys in Kenya

Question

How can a vet prescribe medication without knowing the weight of the donkey?

Refined Question

Can we accurately estimate the weight of a donkey from other more easily obtained measurements?

Sampling Frame

Kate Milner received a grant from The Donkey Sanctuary to Design a Study to Answer this question

Sampling Frame

Donkeys are routinely brought to The Donkey Sanctuary for de-worming

At the sanctuary, they can be weighed and additional measurements taken, such as girth and height.

Measuring girth (cm)

Measuring height (cm)

Other Design Considerations

- Donkeys were randomly selected at the de-worming site

  Why random selection?

- Donkeys were marked after being measured

  Why marked?

- Thirty donkeys were weighed twice, with other donkeys weighed between the 2 measurements

  Why weigh other
  donkeys in between?

Data Collection

	BCS	Age	Sex	Length	Girth	Height	Weight	WeightAlt
0	3.0	<2	stallion	78	90	90	77	NaN
1	2.5	<2	stallion	91	97	94	100	NaN
2	1.5	<2	stallion	74	93	95	74	NaN
3	3.0	<2	female	87	109	96	116	NaN
4	2.5	<2	female	79	98	91	91	NaN
5	1.5	<2	female	86	102	98	105	NaN
6	2.5	<2	stallion	83	106	96	108	NaN
7	2.0	<2	stallion	77	95	89	86	NaN
8	3.0	<2	stallion	46	66	71	27	NaN
9	3.0	<2	stallion	92	110	99	141	NaN

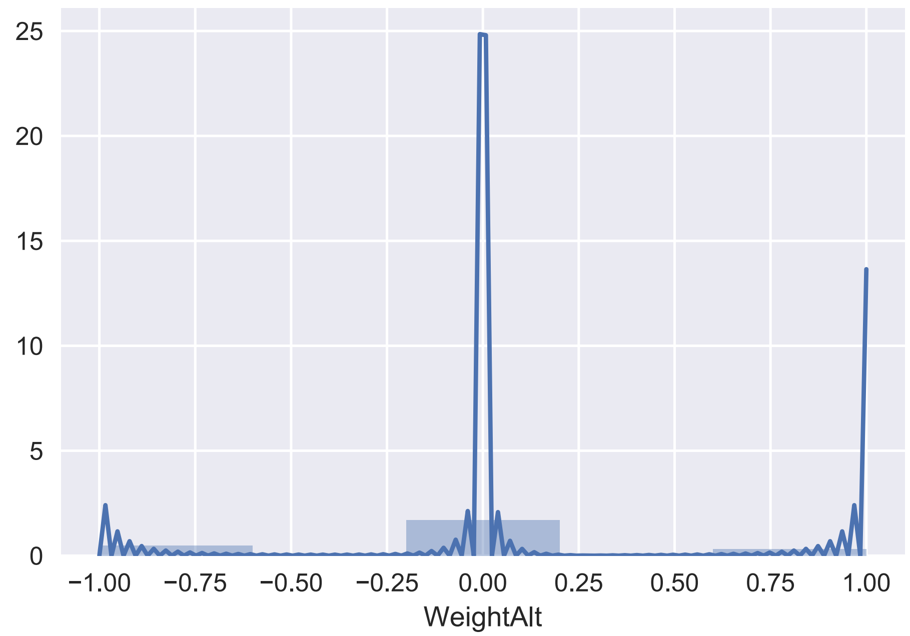
- **BCS** – Body Condition Score 1=emaciated, 3=healthy, 5=obese, with ½ scales
- **Age** - <2, 2-5, 5-10, 10-15, 15-20, >20 years
- **Sex** – stallion, gelding, female
- **Length** (cm)
- **Girth** (cm)
- **Height** (cm)
- **Weight** (kg) - RESPONSE

Data Cleaning

Data Cleaning

Compare the second weighing to the first weighing for the 30 donkeys

Conclusion:



Data Cleaning

Further investigation reveals

- 1 donkey has a BCS 1
- 1 donkey has a BCS 4.5
- 1 donkey weighs 27 kg and is determined to be a baby

```
donkeys.describe()
```

	BCS	Length	Girth	Height	Weight	WeightAlt
count	544.000000	544.000000	544.000000	544.000000	544.000000	31.000000
mean	2.889706	95.674632	115.946691	101.349265	152.104779	150.258065
std	0.425656	7.348897	7.438570	4.256430	26.506715	22.711183
min	1.000000	46.000000	66.000000	71.000000	27.000000	98.000000
25%	2.500000	92.000000	112.750000	99.000000	139.000000	141.500000
50%	3.000000	97.000000	117.000000	102.000000	155.000000	151.000000
75%	3.000000	101.000000	121.000000	104.000000	170.000000	165.500000
max	4.500000	112.000000	134.000000	116.000000	230.000000	194.000000

What to do with these 3 donkeys?

Modeling

Modeling

- We want to build a model for ***predicting*** weight of a donkey when we don't have the donkey's weight
- The model needs to perform well enough to be used in the field
- The model needs to be ***simple*** enough for implementation in the field

The Variables in Our Model:

$$\min_{\vec{\beta}} ||\vec{y} - \mathbb{X}\vec{\beta}||^2$$

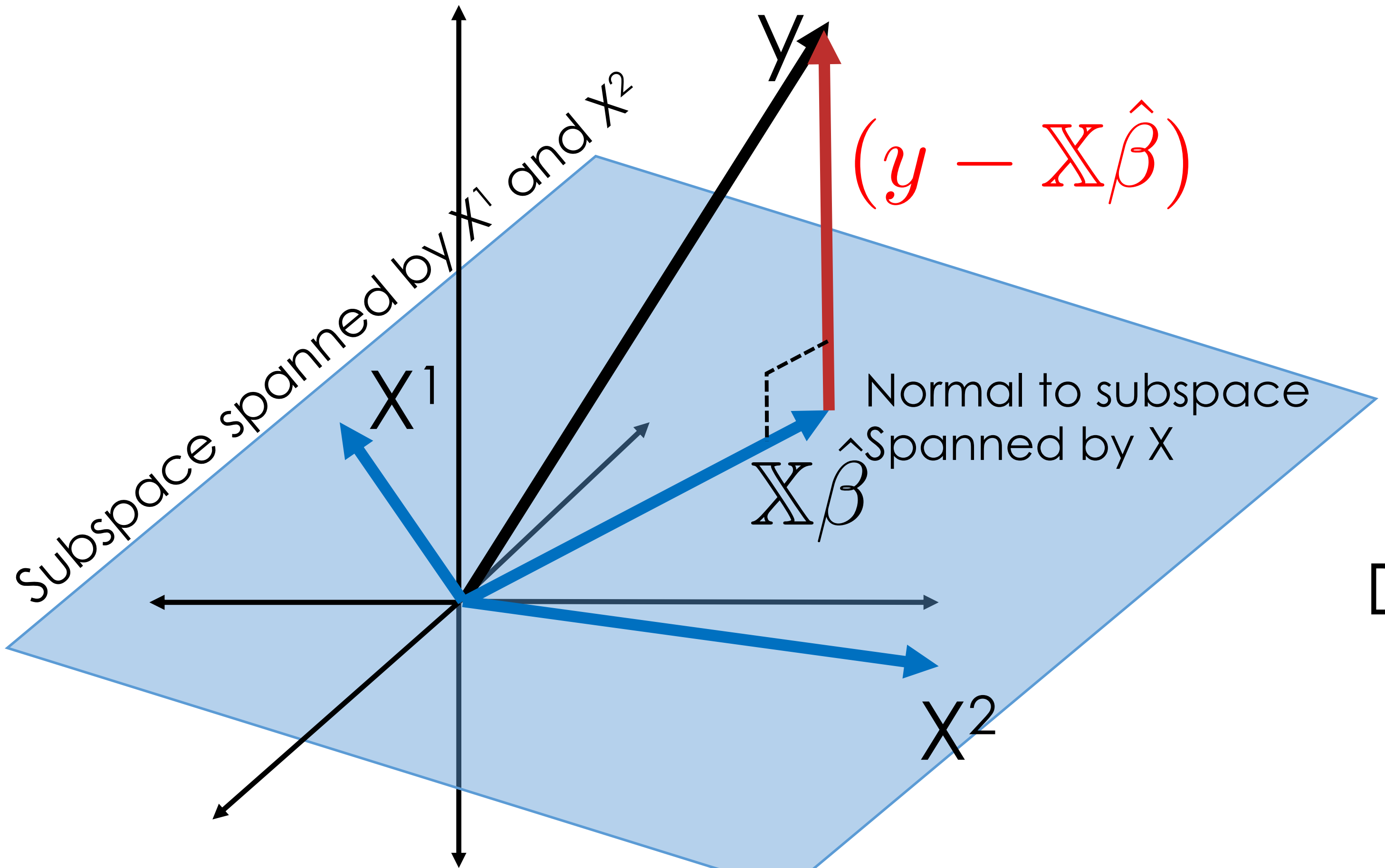
Column/feature space

$$\mathbb{X} = n\left\{\underbrace{\begin{bmatrix} | & | & | & \cdots & | \\ \vec{1} & \vec{x}_1 & \vec{x}_2 & \cdots & \vec{x}_p \\ | & | & | & \cdots & | \end{bmatrix}}_{p+1}\right. \qquad \vec{y} = n\left\{\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{1}\right. \in \mathbb{R}^n$$

n records in ***p+1*** dimensions (columns or features)

Ŷ minimizes the L_2 empirical risk

$$\min_{\vec{\beta}} \|\vec{y} - \mathbb{X}\vec{\beta}\|^2$$



Ŷ is the PROJECTION of Y into the subspace spanned by the columns of X

Definition of orthogonal

$$0 = \mathbb{X}^t(\vec{y} - \mathbb{X}\vec{\hat{\beta}})$$

Solve for $\vec{\hat{\beta}}$

$$0 = \mathbb{X}^t(\vec{y} - \mathbb{X}\vec{\hat{\beta}})$$ Definition of orthogonal

$$0 = \mathbb{X}^t\vec{y} - \mathbb{X}^t\mathbb{X}\vec{\hat{\beta}}$$

$$\mathbb{X}^t\mathbb{X}\vec{\hat{\beta}} = \mathbb{X}^t\vec{y}$$ Normal Equations

$$\vec{\hat{\beta}} = (\mathbb{X}^t\mathbb{X})^{-1}\mathbb{X}^t\vec{y}$$

$$\vec{\hat{y}} = \mathbb{X}\vec{\hat{\beta}} = \mathbb{X}(\mathbb{X}^t\mathbb{X})^{-1}\mathbb{X}^t\vec{y}$$

How can we assess our model?

- How well does our model predict the weight of a new donkey?
- The risk: For a new donkey with p features: x_0

$$\mathbb{E}(Y_0 - \hat{Y}_0)^2 = \mathbb{E}(Y_0 - \underbrace{x_0^t}_{1 \text{ x } (p+1)} \overbrace{\hat{\beta}}^{(p+1) \text{ x } 1})^2$$

Only problem is that we can't take this expectation

E.g., a row in the design X

Train – Test Paradigm

Set aside some data before we begin our EDA and model fitting

How can we assess our model?

- If we use the same data to fit and assess the model, then we overestimate how well our model does at prediction.
- Instead, use a test set: $(x_j, Y_j) \quad \text{for } j = 1, \ldots, m$

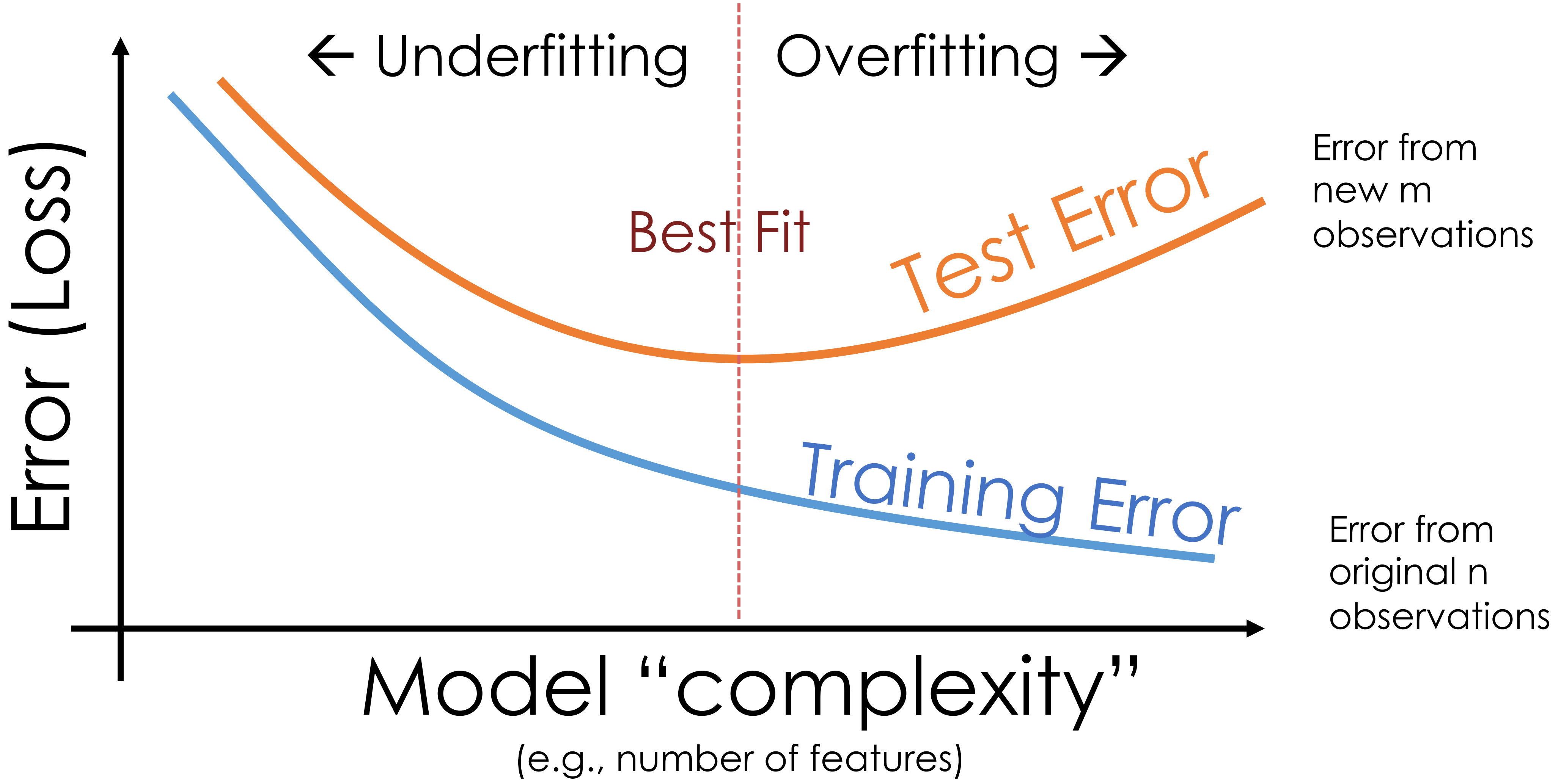
$$\mathbb{E}(Y_0 - \hat{Y}_0)^2 \; = \mathbb{E}(Y_0 - x_0^t \hat{\beta})^2$$

$$\approx \frac{1}{m} \sum_{j=1}^{m} (Y_j - x_j^t \hat{\beta})^2$$

Assessed (AKA tested) on m independent observations

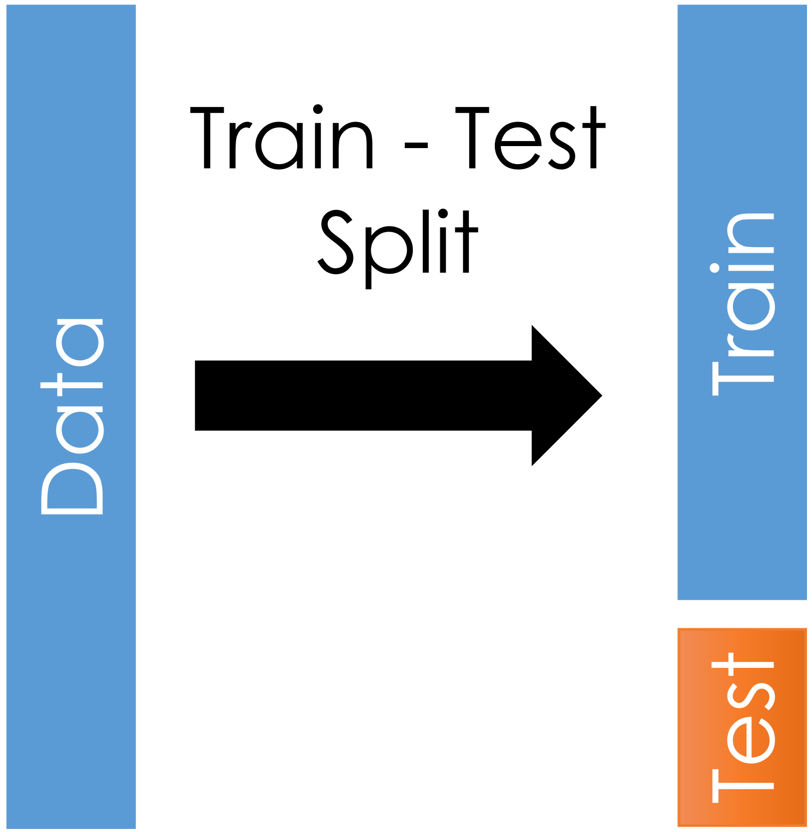
Fitted (AKA trained) on n observations

Training vs **Test** Error



Train-Test Split – With one set of data

- **Training Data:** used to fit model
- **Test Data:** check generalization error
- How to split?
  - Randomly, Temporally, Geo…
  - Depends on application (usually randomly)
- What size? (90%-10%)
  - Larger training set → more complex models
  - Larger test set → better estimate of generalization error
  - Typically between 75%-25% and 90%-10%



You only use the test dataset ***once*** after deciding on the model.

Split our data before we begin EDA

Set aside 20% of the records

We will use these to assess the accuracy of our model

```
indices = np.arange(len(donkeys2))
np.random.shuffle(indices)
n_train = int(np.round((len(donkeys2)*0.8)))
n_test = len(donkeys2) - n_train
```

```
indices[:n_train]
```

```
array([454, 108, 271, 453, 339, 142, 518, 513, 151, 443, 194, 523, 470,
       342, 287,  34, 514, 314, 220, 100, 185,   5, 512, 331, 224, 153,
       386, 463,  74, 164, 458, 270, 102,  92,   3, 393, 278, 189,  31,
        21, 344, 304, 155, 492, 318, 133,  69, 343, 242,  61, 363, 262,
        91, 407, 491, 481, 120, 276,  42, 404, 460, 255, 418, 234, 149,
```

```
train = donkeys2.iloc[indices[ :n_train], ]
```

Train Model then Test Model

- Optimize on Train set

$$\min_{\beta} ||\vec{y}_{train} - \mathbb{X}_{train}\vec{\beta}||^2$$

0.8n x 1 0.8n x p p x 1

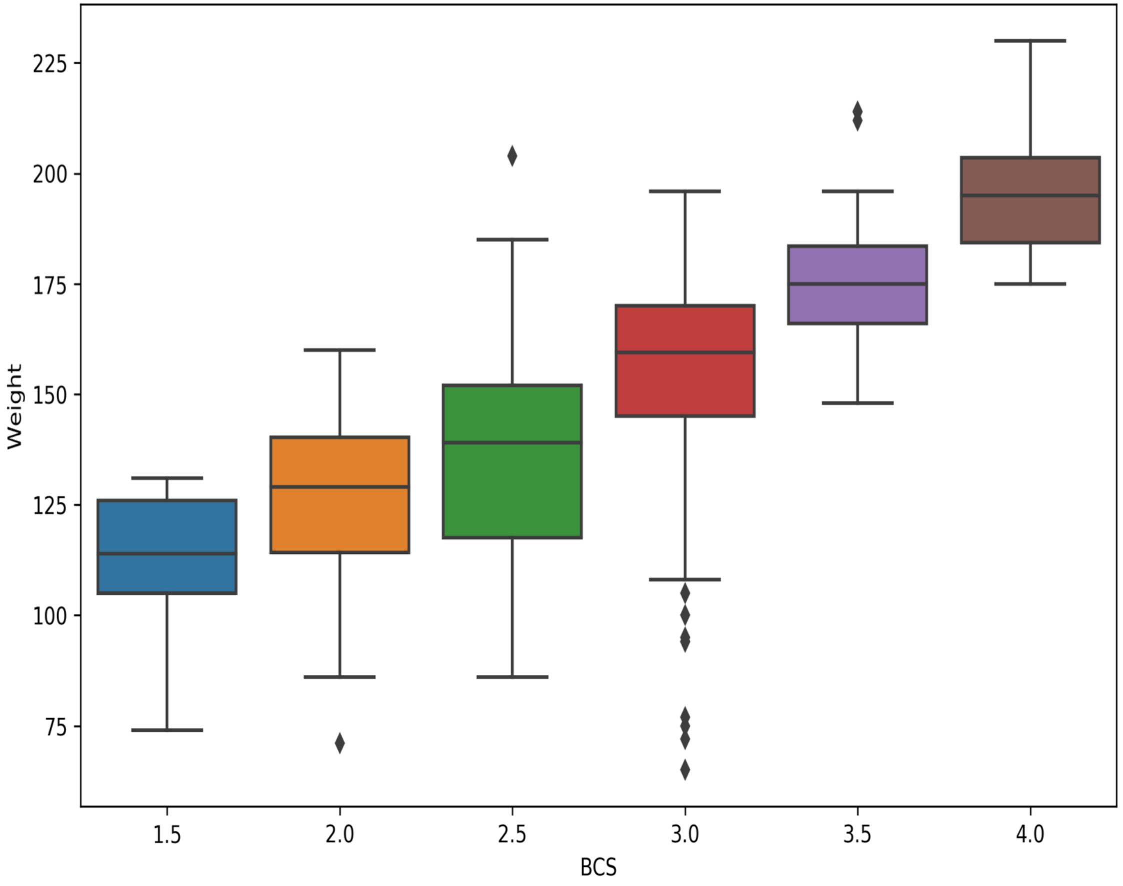
- Minimizer: $\vec{\hat{\beta}}_{train}$

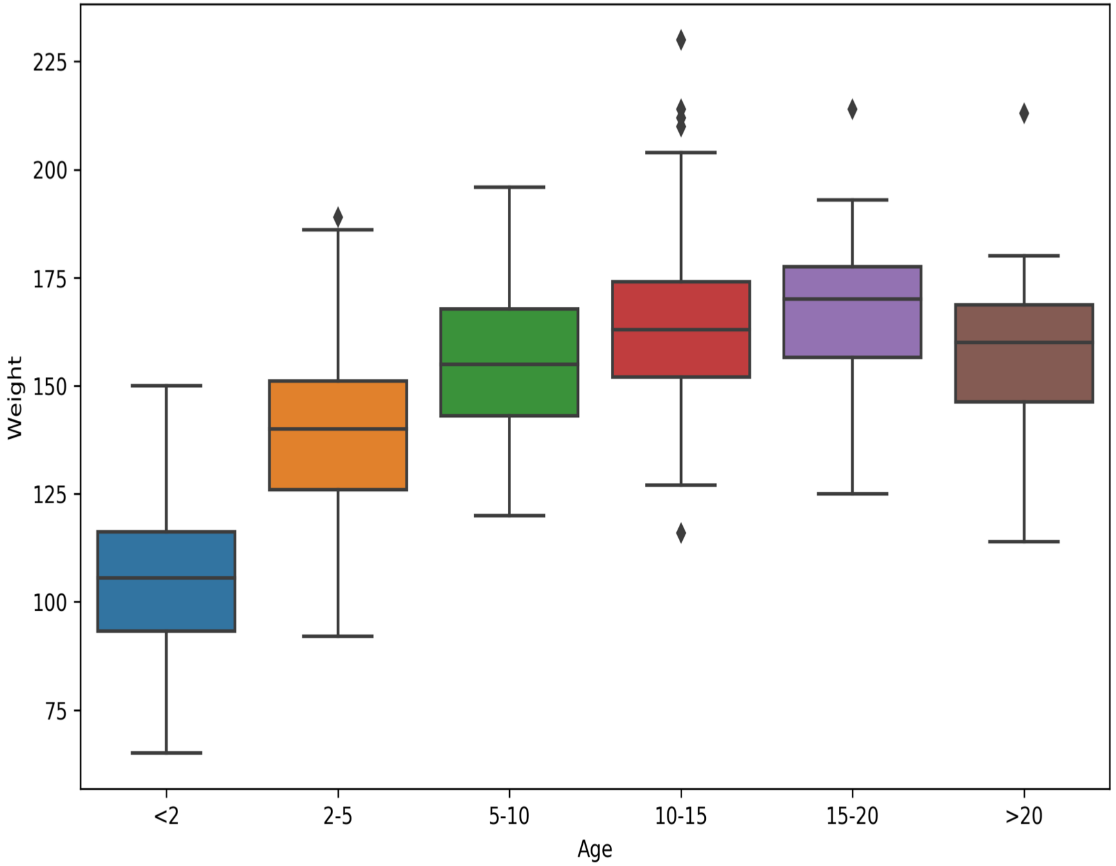
- Evaluation on Test set

$$||\vec{y}_{test} - \mathbb{X}_{test}\vec{\hat{\beta}}_{train}||^2$$

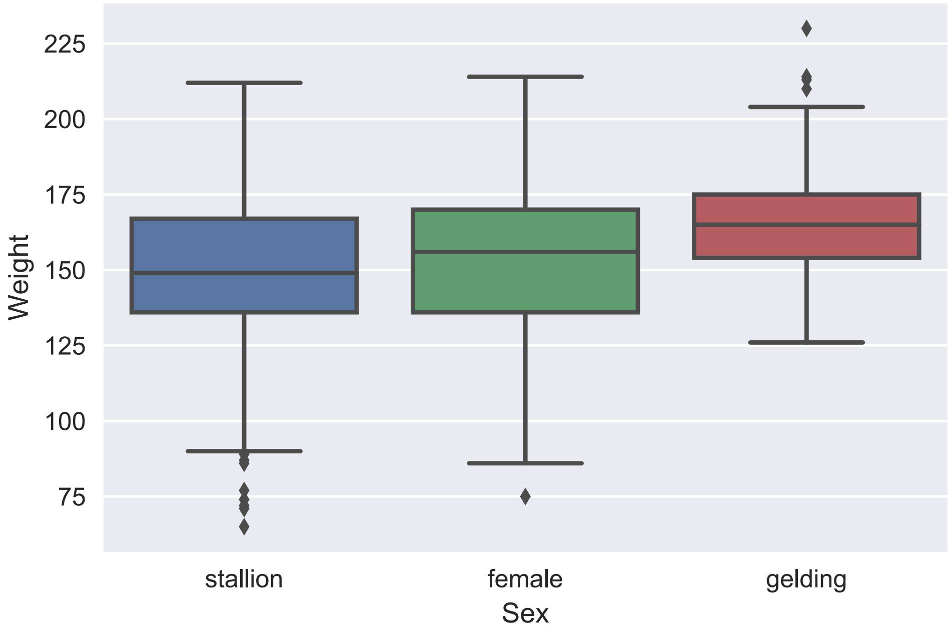
0.2n x 1 0.2n x p p x 1

EDA

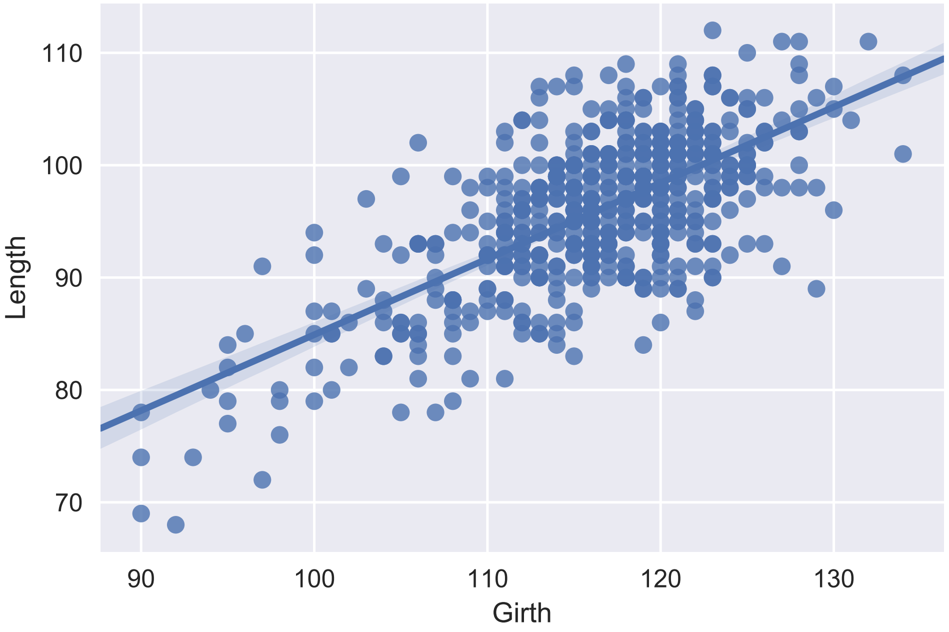




Those over 5 seem to have the same weight distribution



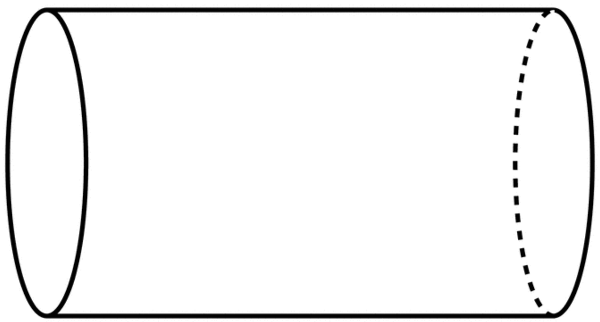
Not a big difference between stallions and females



Girth and length are correlated

Starting Point for Model

Physical Model



The donkey as a cylinder with appendages

Suggests Model:

$$h(weight) = \alpha + \beta log(girth) + \gamma log(length)$$

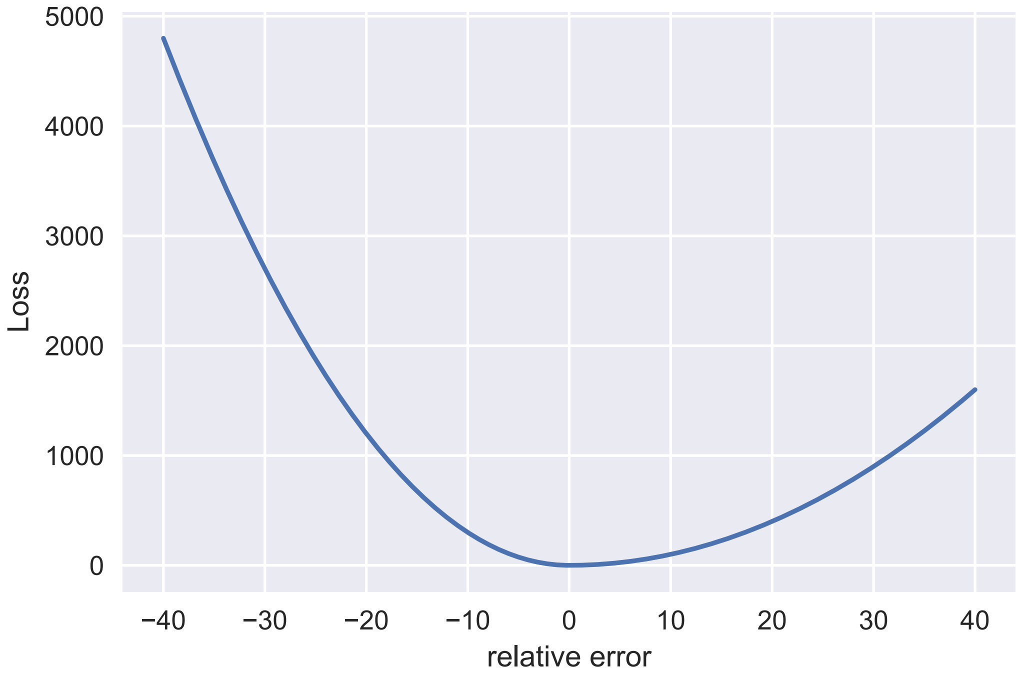
Statistically, consider other variables and various transformations of weight

Loss Function

Two Scenarios

- Loss function should reflect the cost to the donkey's health of prescribing the wrong dose
- Antibiotics:
  - Effect is less sensitive to the weight of the donkey
  - Better to overdose: otherwise infection might not be treated
  - An under-dose could lead to drug resistance
- Anesthetics:
  - Effect is more sensitive to the weight of the donkey
  - Better to under-dose: the effect can be observed and adjusted

Anesthetics Scenario



QUESTION: Does a negative value correspond to an overdose or an under-dose?

entire

The x-axis is relative error as a percentage

A value of -10% corresponds to the situation where the actual weight is 10% smaller than the predicted weight

$$\frac{\text{actual} - \text{predicted}}{\text{predicted}} * 100\%$$

Minimization

- Geometric perspective useful for L_2 loss, but not here
- We can use calculus to derive the normal equations for this loss and easily solve for the optimizing parameters
- In lab, we saw techniques for minimizing general loss functions, we will cover this in more detail next week

```
In [143]: from scipy.optimize import minimize

          res = minimize(lambda theta: new_loss(theta, X, y), np.ones(3))
          # estimates for theta
          theta_hat = res['x']
```

Feature Engineering

Keeping it *Real*

Feature Engineering

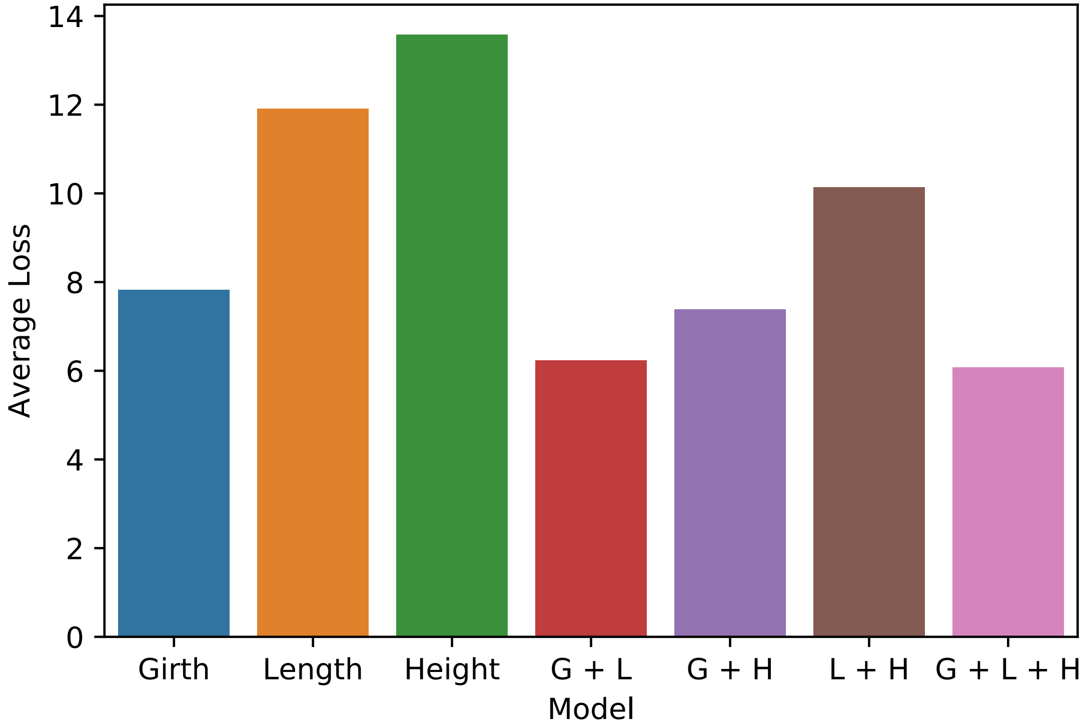
- The process of transforming the inputs to a model to improve prediction accuracy.
  - A key focus in many applications of data science
  - An art …
- Feature Engineering enables you to:
  - **encode** non-numeric features to be used as inputs to models
  - capture **domain knowledge** (e.g., periodicity or relationships between features)
  - **transform complex relationships** into simple linear relationships

Basic Transformations

- Uninformative features: (e.g., UID)
  - Is this informative (probably not?)
  - **Transformation:** remove uninformative features (why?)
- Quantitative Features (e.g., Length)
  - **Transformation:** May apply non-linear transformations (e.g., log)
  - **Transformation:** Normalize/standardize (more on this later …)
    - Example: (x – mean)/stdev
- Categorical Features (e.g., sex)
  - How do we convert sex into meaningful numbers?
    - female =1 , gelding = 2, stallion = 3?
    - Implies order/magnitude means something … we don't want that ...
  - **Transformation:** *One-hot-Encode*

We have 3 numeric variables

Use 1, 2, or 3 variables in the model?



There are only 7 combinations of variables, so we try all of them.

What would you do?

deal

Take Stock

- Dropped 3 records
- Divided the data into 20%-80% split and set 20% aside
- Selected a loss function that erred on the side of under-dosing
- Examined models for weight based on the numeric variables and selected girth and length to model weight
- EDA showed that the qualitative variables may be useful

Qualitative Variables

Recall

Recall our original optimization problem when we had no additional information and wanted to find the closest constant to **y**

$$\frac{1}{n}\sum_{i=1}^{n} loss(y_i, \beta)$$

We saw that for L_2 loss the minimizer was the mean:

$$\hat{\beta} = \bar{y}$$

We have information about which group each observation belongs to

We are interested in finding the closest constant to each group.

Call them
$\beta_g, \beta_s, \beta_f$

$$\sum_{i \in gelding} loss(y_i, \beta_g)$$

$$\sum_{i \in stallion} loss(y_i, \beta_s)$$

$$\sum_{i \in female} loss(y_i, \beta_s)$$

Use the information about which group each observation belongs to

Minimize with respect to β_g

The minimum is the average for the group, $\hat{\beta}_g = \bar{y}_g$

$$\sum_{i \in gelding} loss(y_i, \beta_g)$$

$$\cancel{\sum_{i \in stallion} loss(y_i, \beta_s)}$$

$$\cancel{\sum_{i \in female} loss(y_i, \beta_s)}$$

Introduce 0-1 Variables

$\vec{x}_g$ Vector (n by 1) of 0s and 1s:
1 for the observations that correspond to geldings

$$\begin{aligned} x_{g,i} &= 1 \;\; \text{if the } i^{th} \text{ observation is a gelding} \\ &= 0 \;\; \text{if the } i^{th} \text{ observation is not a gelding} \end{aligned}$$

$\vec{x}_s, \; \vec{x}_f$ Vector of 0s and 1s to indicate stallion (or female)

$\vec{y}$ Weight measurements

Transforming a Qualitative Variable

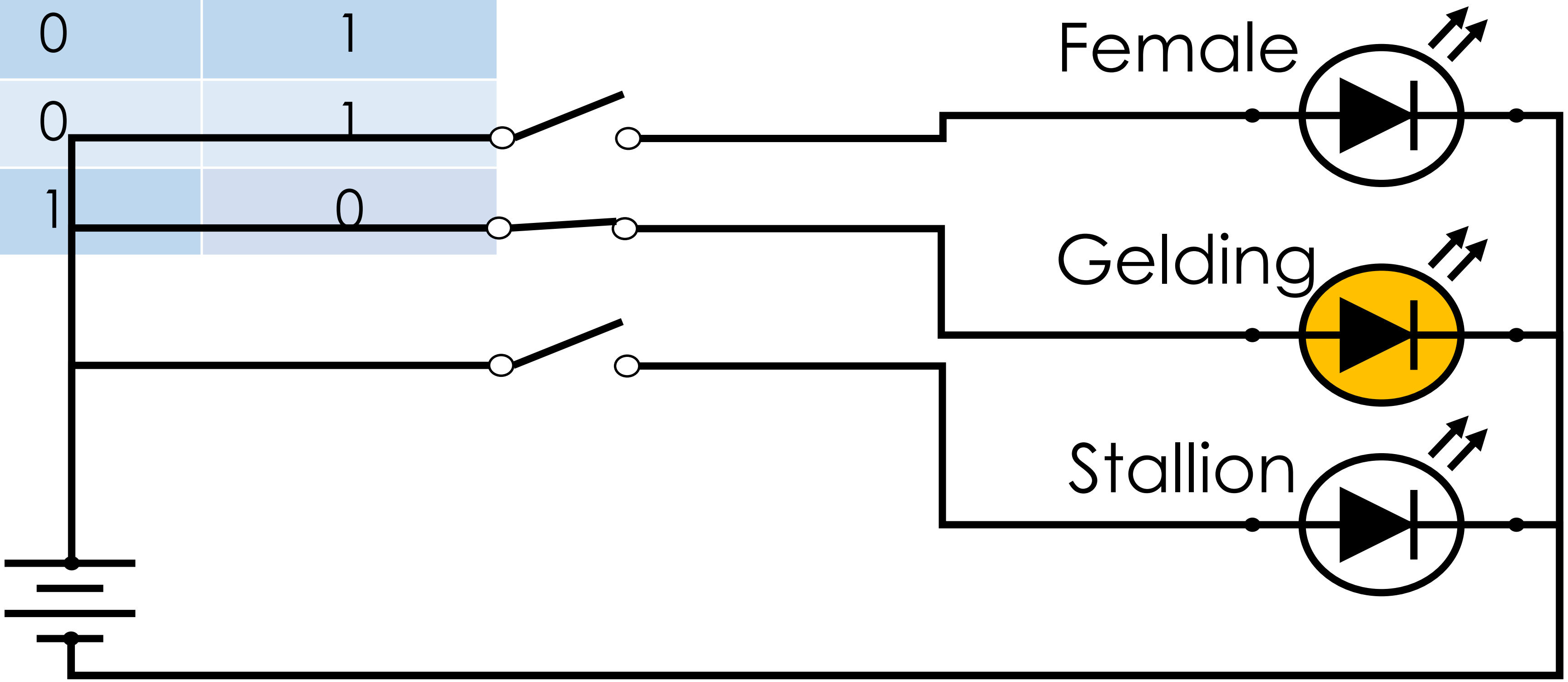
- Transform categorical feature into binary features:

Sex
gelding
stallion
female
female
stallion

gelding	stallion	female
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0

AKA One-hot encoding

gelding	stallion	female
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0



Re-express Loss with 0-1 Variables

$$\sum_{i=1}^{n}[y_i - (x_{g,i}\beta_g + x_{s,i}\beta_s + x_{f,i}\beta_f)]^2$$

$$= \|\vec{y} - (\vec{x}_g\beta_g + \vec{x}_s\beta_s + \vec{x}_f\beta_f)\|^2$$

$$= \|\vec{y} - \mathbb{X}\vec{\beta}\|^2$$

Model with girth and sex dummies

$\vec{x}_g \ \vec{x}_s, \ \vec{x}_f$ Vectors (n by 1) of 0s and 1s for geldings, stallions, and females respectively

$\vec{x}_r$ Girth measurements

$\vec{y}$ Weight measurements

$$\|\vec{y} - (\vec{x}_r\beta_r + \vec{x}_g\beta_g + \vec{x}_s\beta_s + \vec{x}_d\beta_f)\|^2$$

Model with girth and sex dummies

$$\underbrace{\vec{x}_r\beta_r + \vec{x}_g\beta_g + \vec{x}_s\beta_s + \vec{x}_f\beta_f}$$

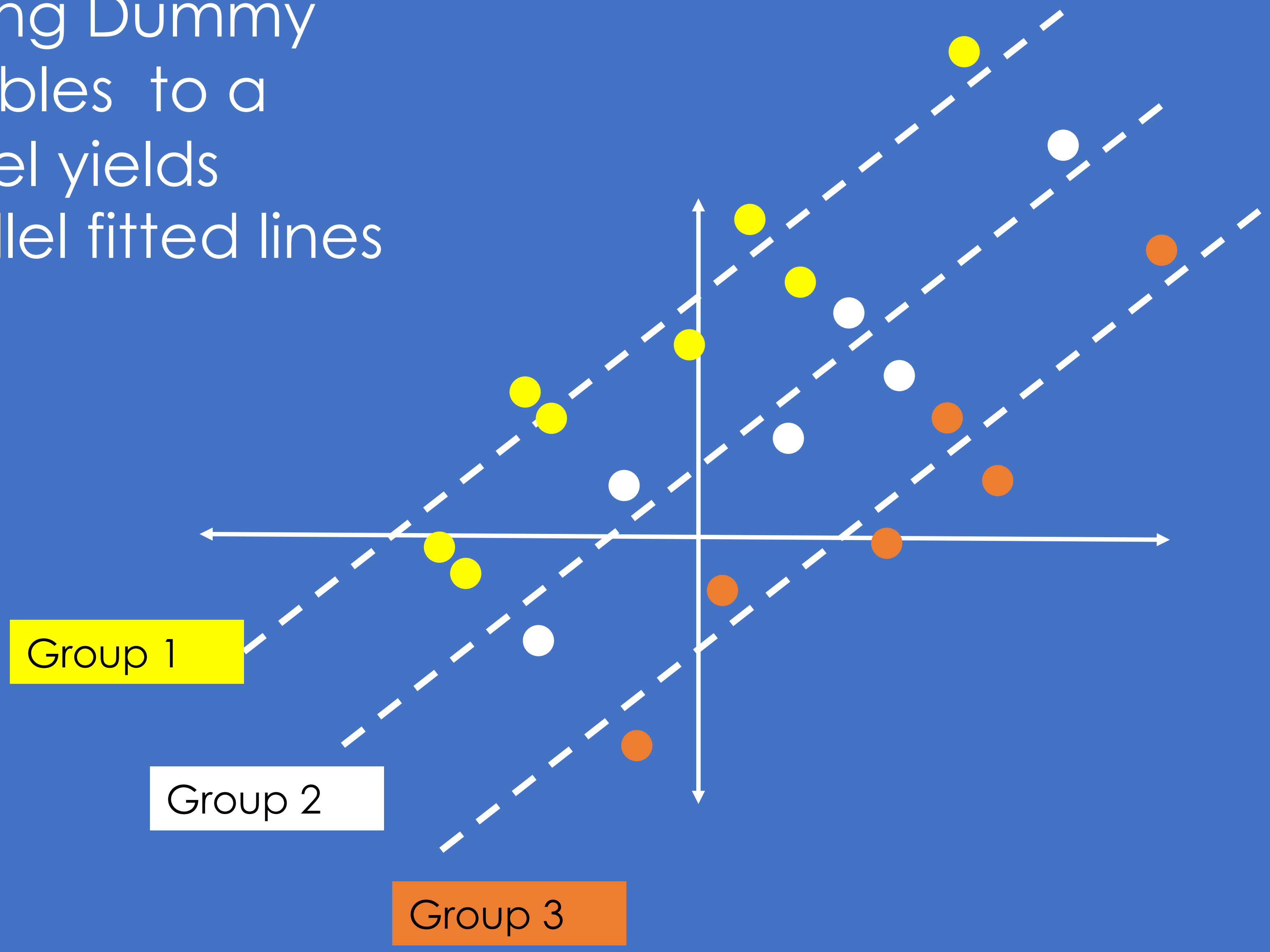
For a gelding, what does this linear model reduce to?

The stallion and female dummies are both 0 $x_{r,i}\beta_r + \beta_g$

The stallion model is $x_{r,i}\beta_r + \beta_s$

The female model is $x_{r,i}\beta_r + \beta_f$

Adding Dummy variables to a model yields parallel fitted lines



Sex and Girth

When our model has dummies and quantitative variables, we often include an intercept term.

Our design has collinearity problems

1	girth
1	100
1	110
1	121
1	92
1	100

gelding	stallion	female
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0

Sex and Girth Design

We often remove one of the dummy variables.

gelding	stallion	female	1	Girth
1	0	0	1	..
0	1	0	1	..
0	0	1	1	..
0	0	1	1	..
0	1	0	1	..

How can we express in terms of the remaining variables?

gelding	stallion	1	Girth
1	0	1	..
0	1	1	..
0	0	1	..
0	0	1	..
0	1	1	..

In this case, the female donkey average is the intercept, and the gelding and stallion coefficients represent the amount to be added or removed from the female average

Sex and BCS

If we include both Sex and BCS we run into the same problem, i.e., the sum of the sex dummies = sum of the BCS dummies so we have collinearity again

gelding	stallion	female
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0

BCS_1.5	BCS_2.0	BCS_2.5	…	BCS_4.0
1	0	0		0
0	1	0		0
0	0	1		0
0	0	1		0
0	1	0		0

Sex and BCS

What is the rank of this design matrix?

How do you suggest fixing it?

gelding	stallion	female
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0

BCS_1.5	BCS_2.0	BCS_2.5	…	BCS_4.0
1	0	0		0
0	1	0		0
0	0	1		0
0	0	1		0
0	1	0		0

Choice of dummy encodings

Inference

- We are interested in the form of the model and the fitted values of the parameters
- These are not uniquely defined if the model is over parameterized

Prediction

- It doesn’t matter that the model is over parameterized
- We are not interested in the fitted coefficients

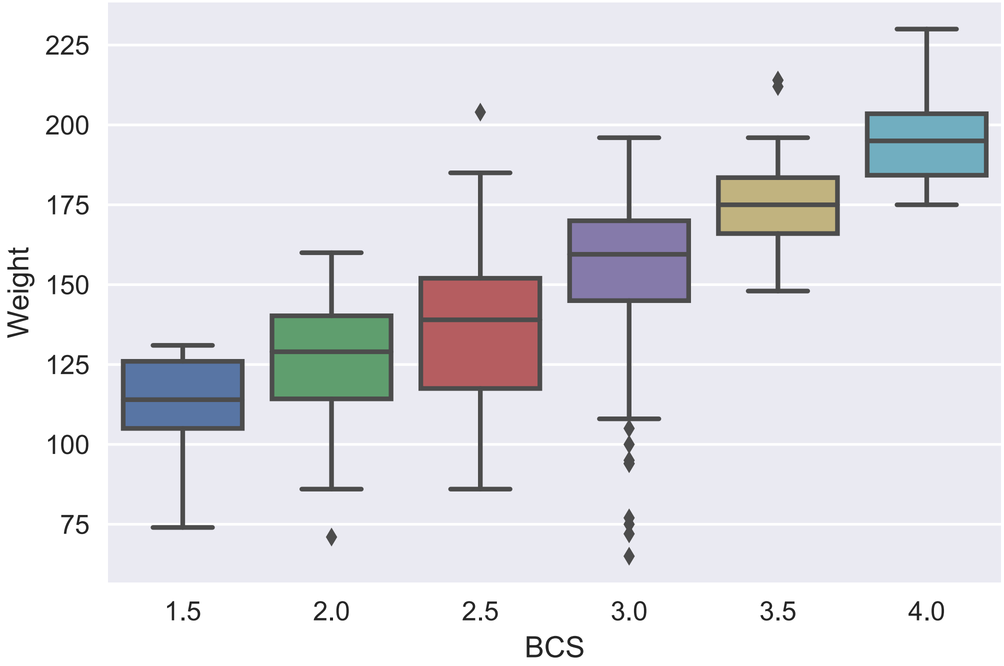
Choice of dummy encodings

- Typically we include the 1 vector
- Select one of the categories for the qualitative variable to be the base/comparison group
- Drop the dummy variable corresponding to that category
- Interpret the other coefficients as the change from the base
- BCS – drop 3, the healthy category
- Sex – drop female because we are interested in collapsing the other two categories or possibly dropping all together

Why not treat BCS as numeric?

BCS
15
2.0
2.5
2.5
2.0

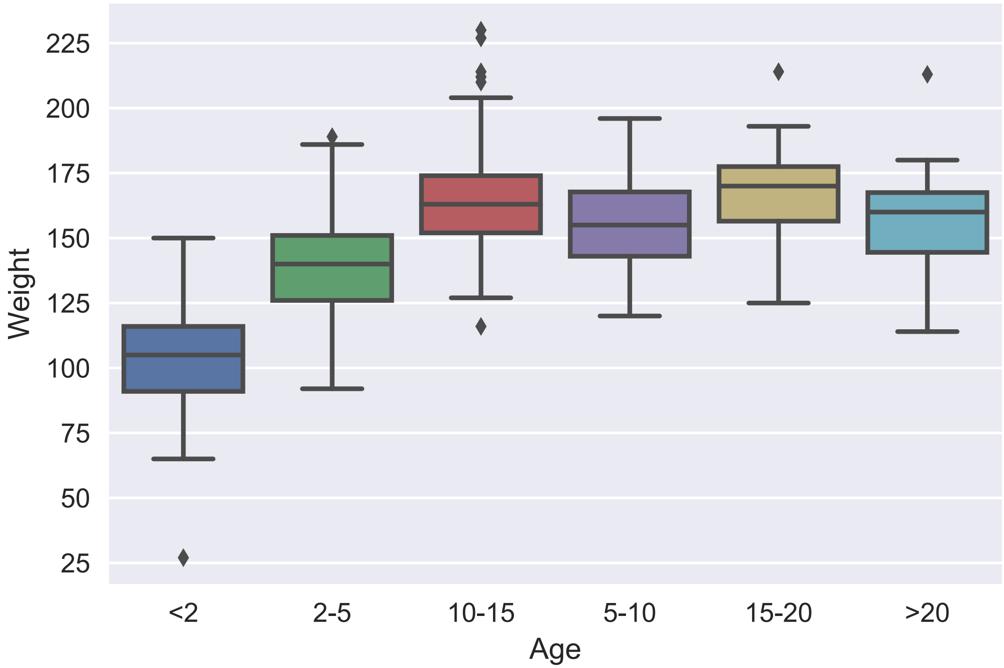
BCS_1.5	BCS_2.0	BCS_2.5	...	BCS_4.0
1	0	0		0
0	1	0		
0	0	1		
0	0	1		
0	1	0		



The relationship need not be linear in the numeric values.

This coding is more flexible

We collapse categories?



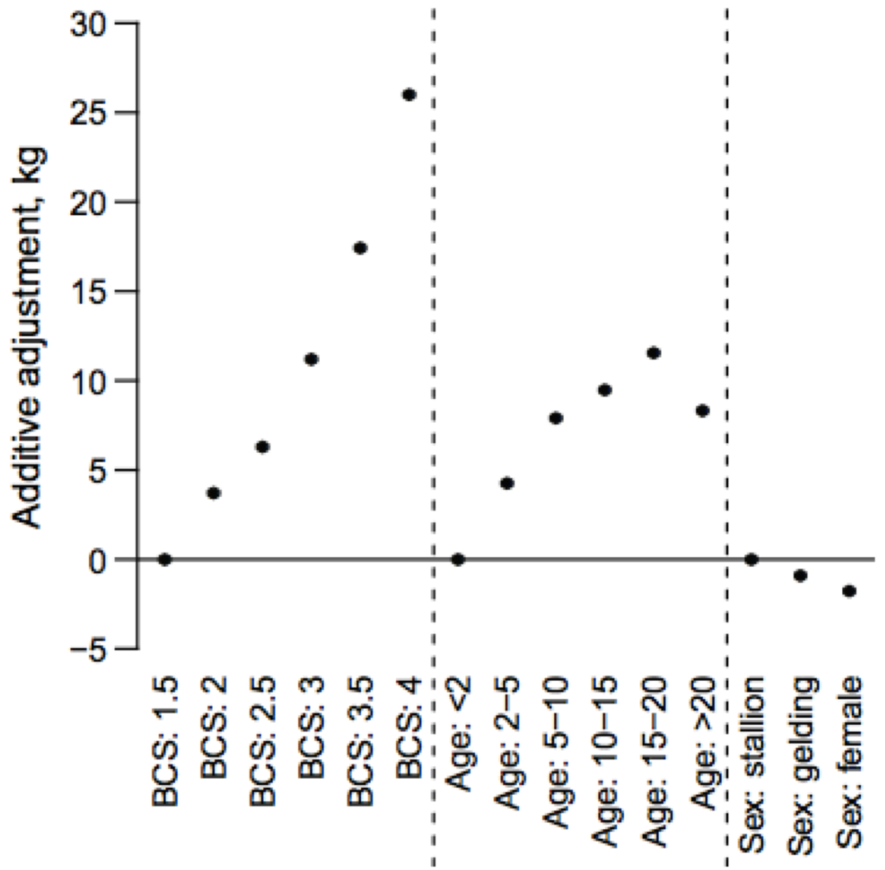
It appears that for donkeys over 5, the groups have similar averages.

Model Selection

Count the variables

- 3 numeric + 2 Sex dummies + 5 BCS dummies + 6 Age dummies = 16 variables
- With dummy variables we are careful when we add and drop variables as that implicitly collapses categories into the base category

Final Model



Keep all levels of BCS

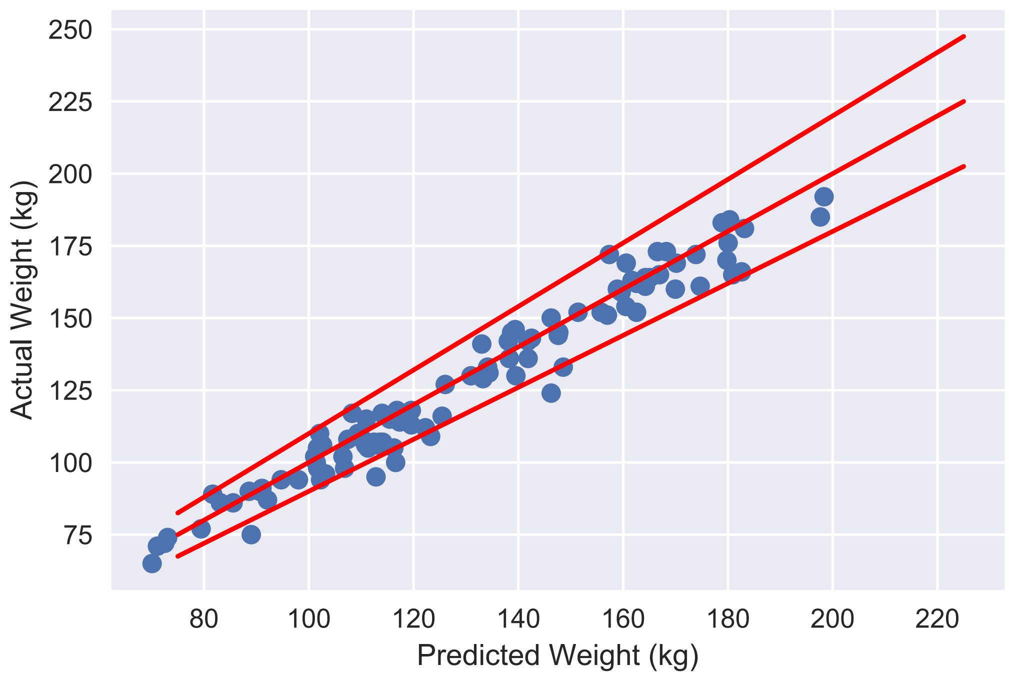
Collapse Age levels over 5 into one

Drop Sex all together

Plus Girth and Length

Model Assessment

Test Data Returns!



Nearly all (95%) of the actual weights are within 10% of the predicted weights

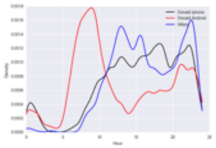
Data Science Life Cycle

Context
Question
Refine Question to an one answerable with data

?

Design
Data Collection
Data Cleaning

Modeling
Test-train split
Loss function choice
Feature engineering
Transformations,
Dummy Variables



Model evaluation
Prediction error

Model selection
Best subset regression
Cross-Validation
Regularization