Introduction to dimensionality reduction

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Dimensions

- Dimensions are the vertical components of a tidy table
- Dimensions = Columns = Features
- # of dimensions = # of columns

```
df %>% ncol()
```

```
3
```

Name	Age	Eye Color
Rob Pitcher	46	Brown
Leslie Sanchez	57	Black
Rafael Linguini	23	Hazel
Petra Sloboda	33	Blue
...	...	...

What is dimensionality reduction?

Eliminating or combining features with little or no new information

Example

Name	Age	Eye Color	Weight (kg)	Weight (lb)	Role
Rob Pitcher	46	Brown	87	194	Developer
Leslie Sanchez	57	Black	114	253	Developer
Rafael Linguini	23	Hazel	68	151	Developer
Petra Sloboda	33	Blue	85	188	Developer
...	...	...	...	...	...

What is dimensionality reduction?

Eliminating or combining features with little or no new information

Example

Name	Age	Eye Color	Weight (kg)	Weight (lb)	Role
Rob Pitcher	46	Brown	87	194	Developer
Leslie Sanchez	57	Black	114	253	Developer
Rafael Linguini	23	Hazel	68	151	Developer
Petra Sloboda	33	Blue	85	188	Developer
...	...	...	...	...	...

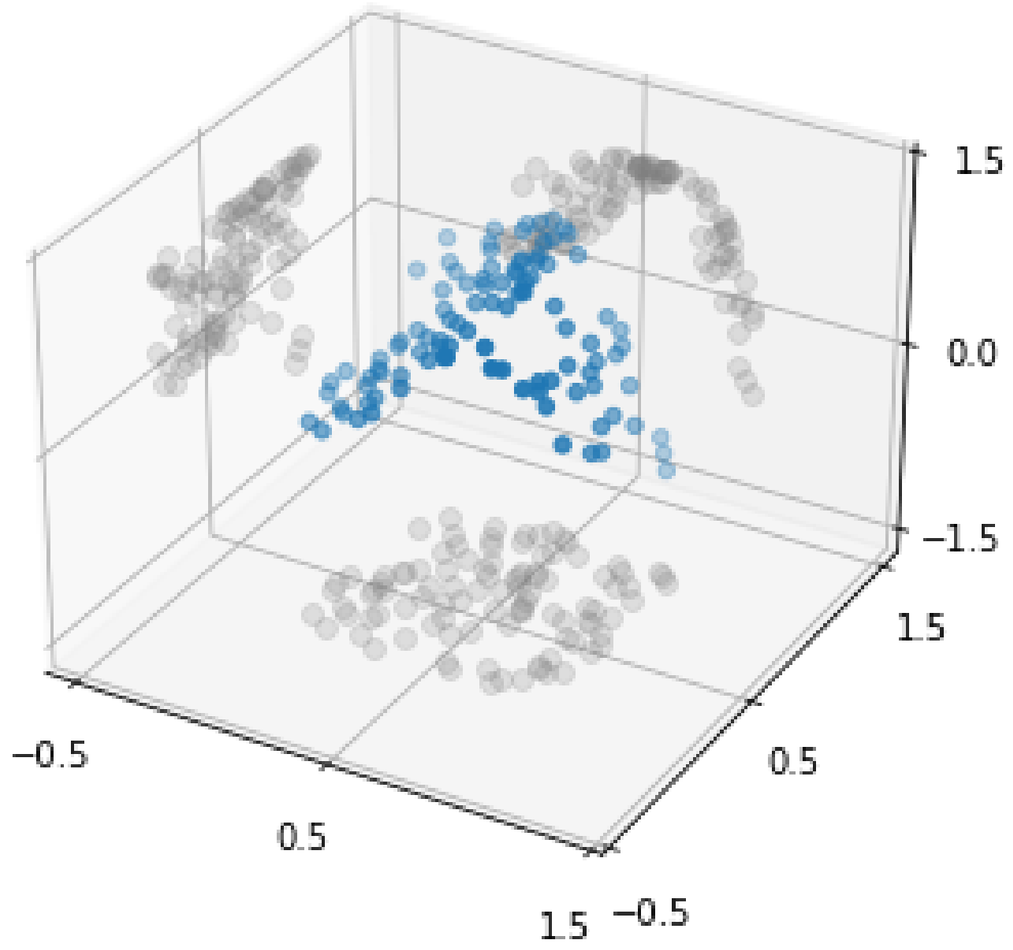
What is dimensionality reduction?

Eliminating or combining features with little or no new information

Example

Name	Age	Eye Color	Weight (kg)	Weight (lb)	Role
Rob Pitcher	46	Brown	87	194	Developer
Leslie Sanchez	57	Black	114	253	Developer
Rafael Linguini	23	Hazel	68	151	Developer
Petra Sloboda	33	Blue	85	188	Developer
...	...	...	...	...	...

Dimensionality reduction visually

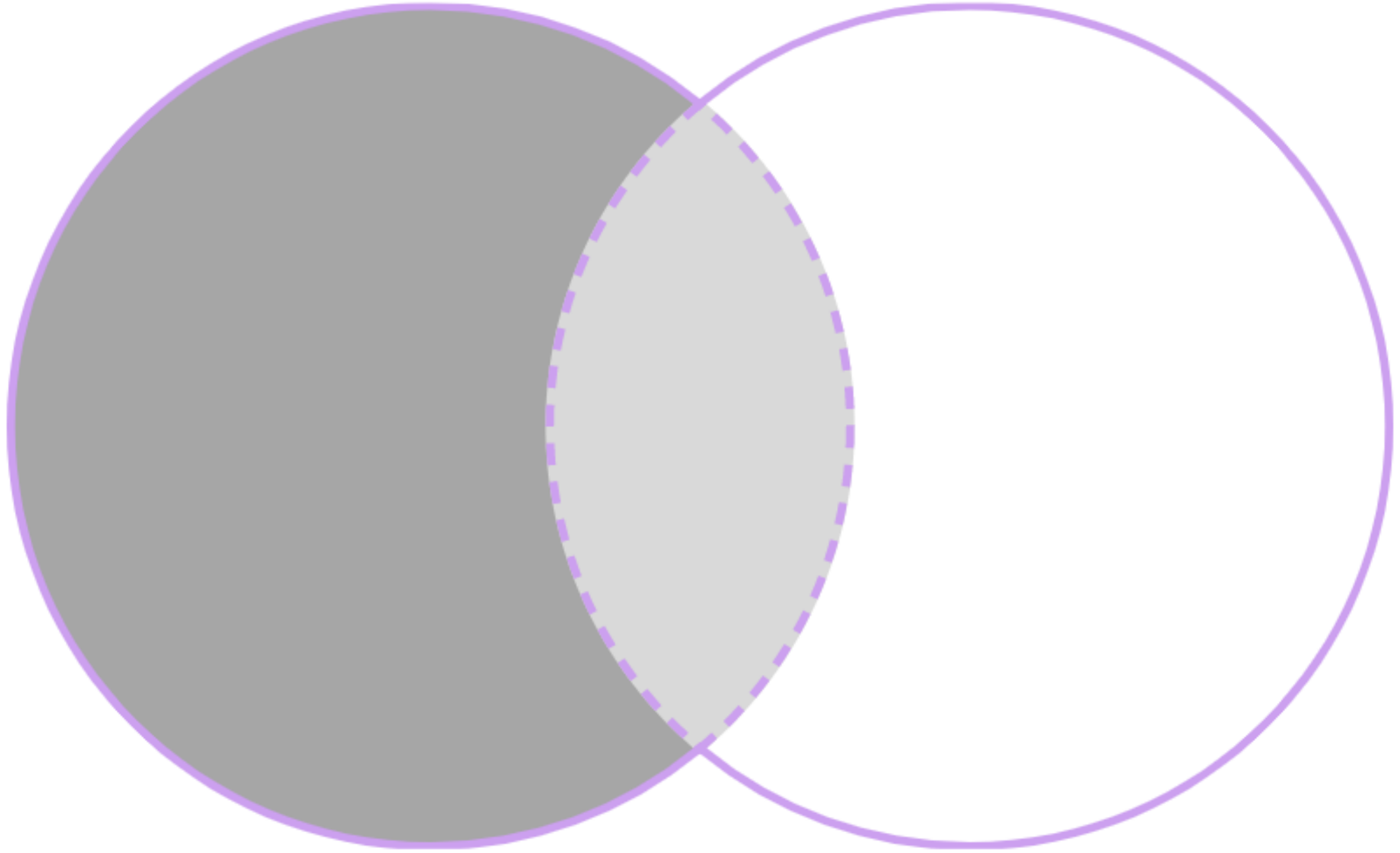


Finding numeric columns with no variance

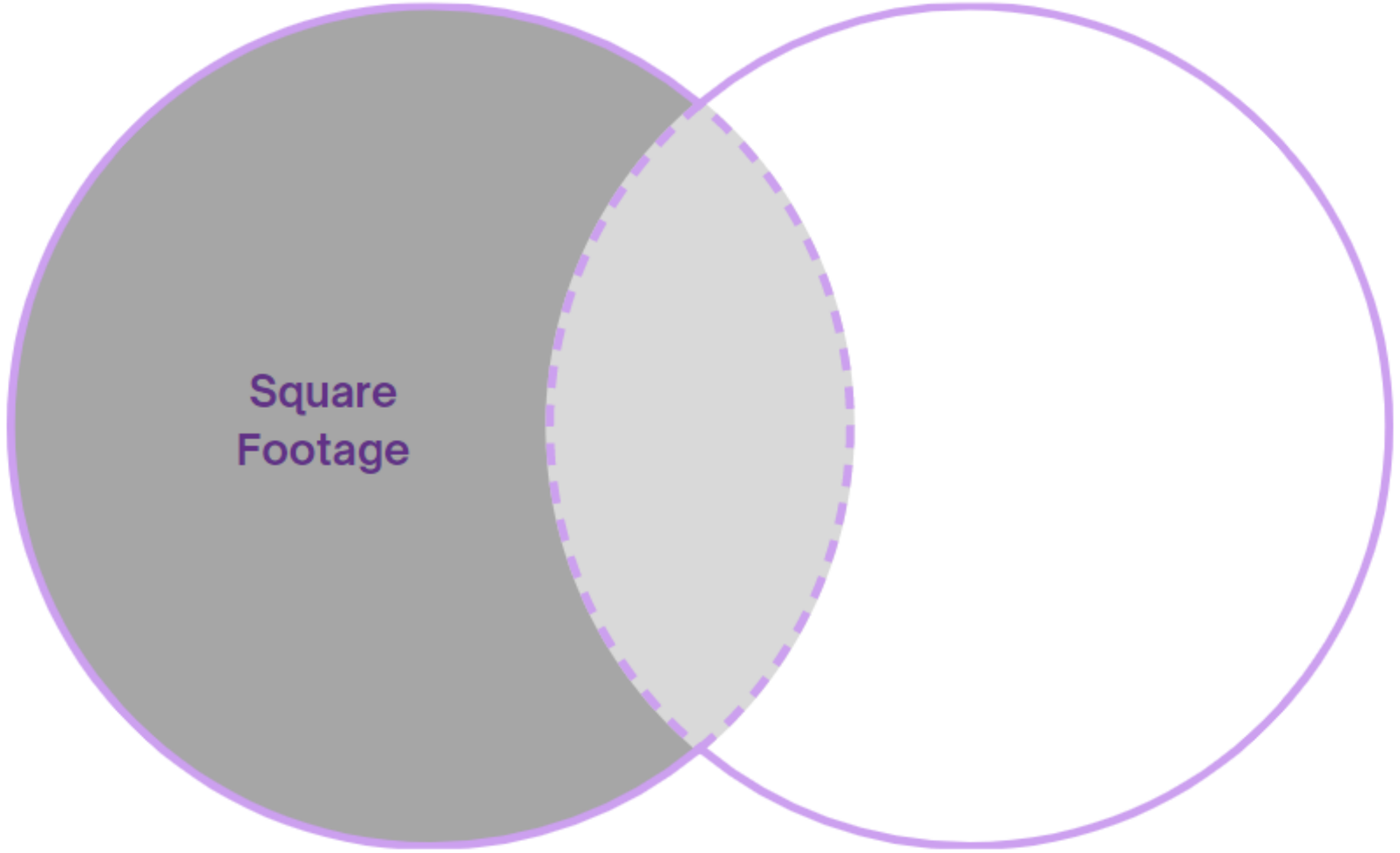
```
df %>%
  summarize(
    across(
      everything(),
      ~ var(., na.rm = TRUE))) %>%
  pivot_longer(
    everything(),
    "feature",
    "variance")
```

```
# A tibble: 7 × 2
  feature                variance
  <chr>                     <dbl>
1 sqft_living             843534.
2 sqft_above              685735.
3 sqft_basement           195873.
4 sqft_living_near15      475480.
5 sqft_lot_near15    863386815.
6 num_garages                  0
7 num_hvac_units               0
```

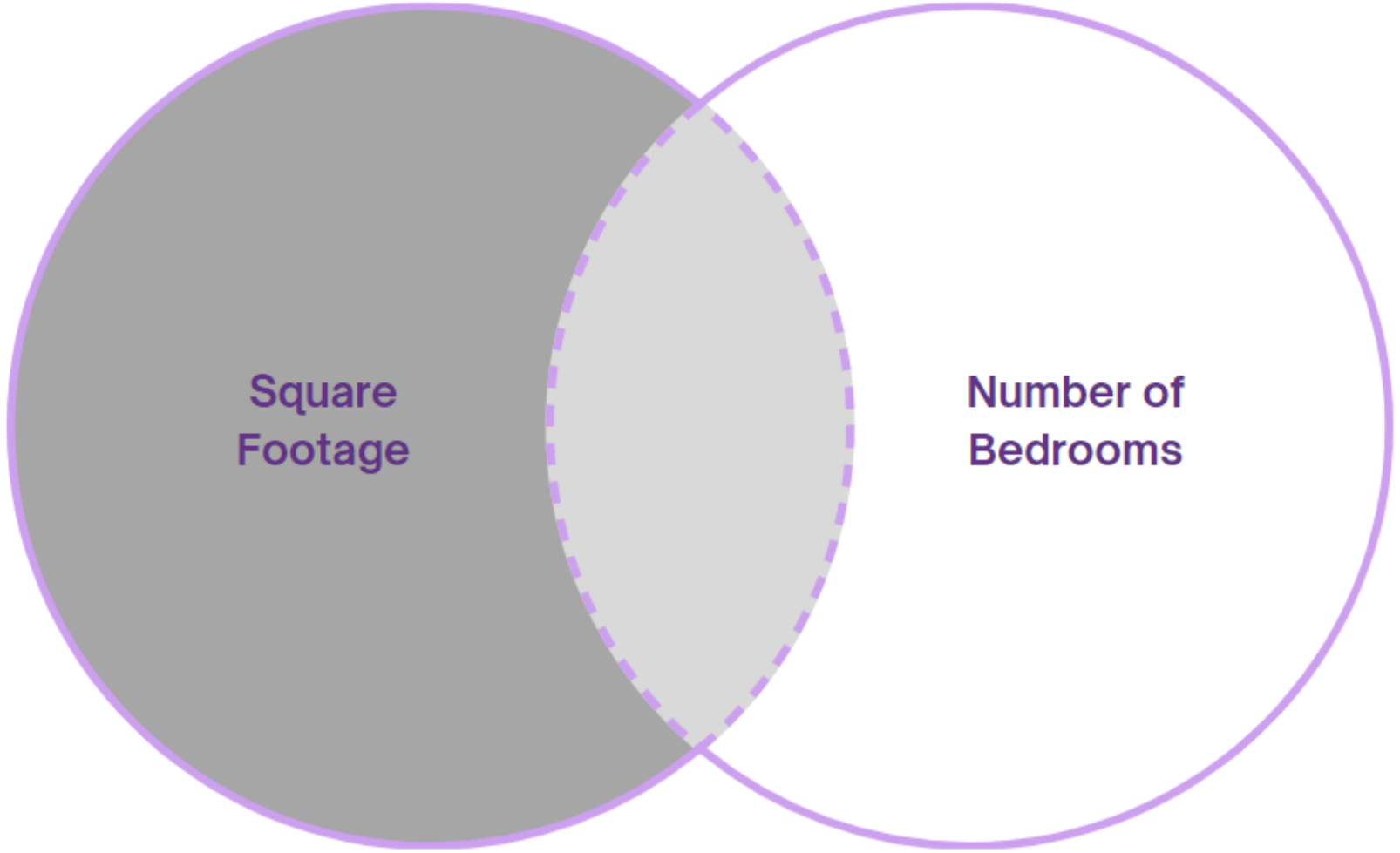
Mutual information



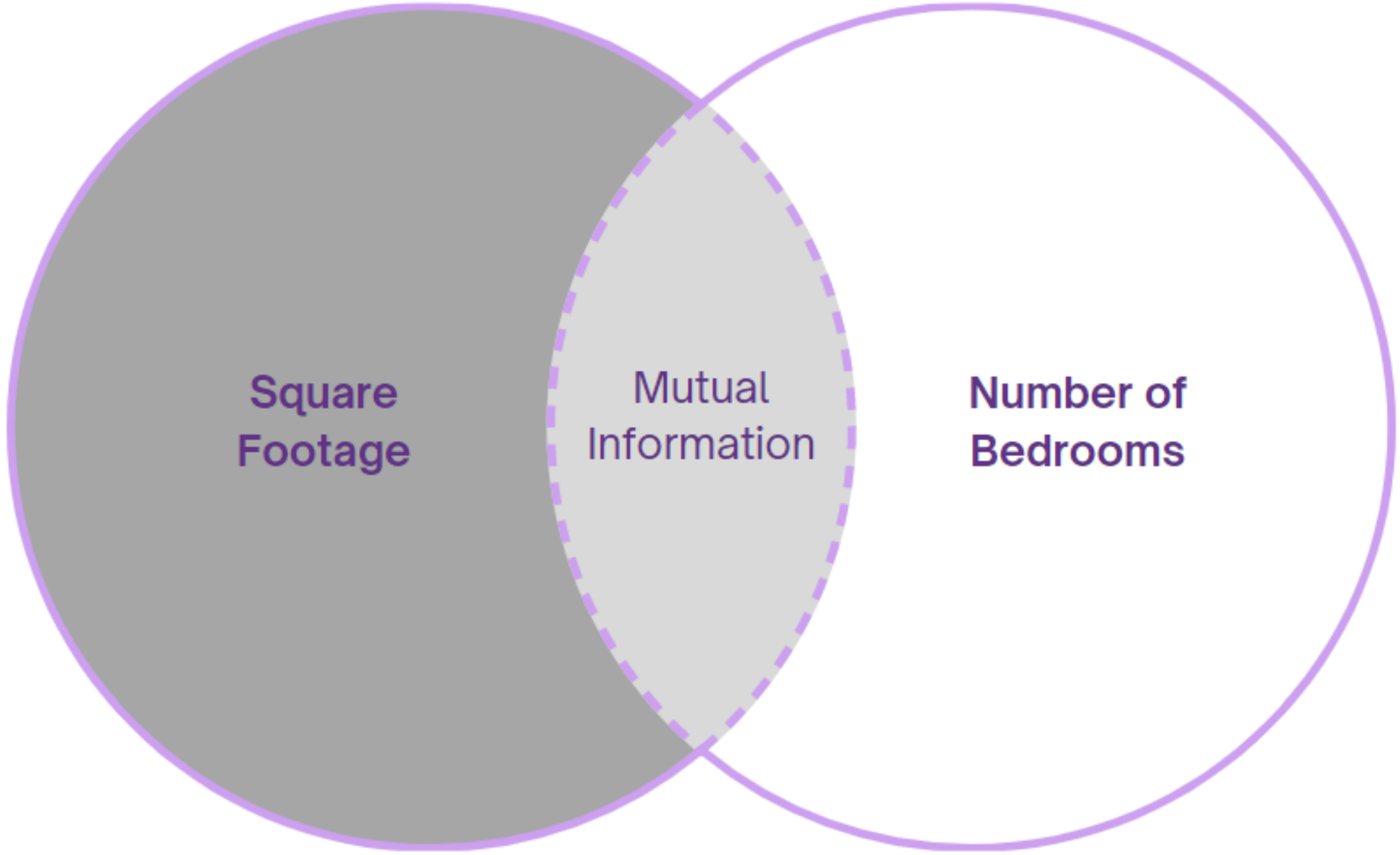
Mutual information



Mutual information



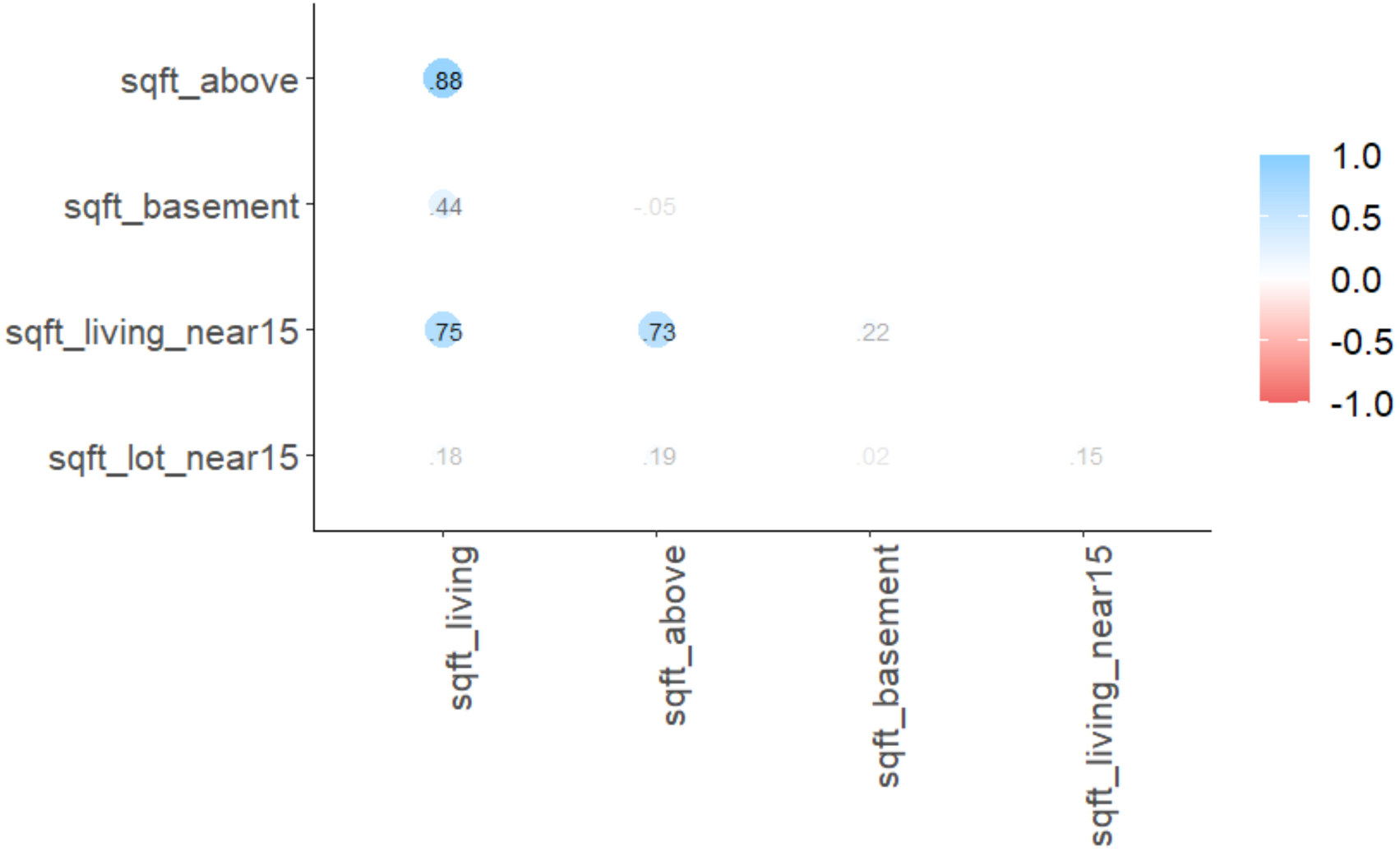
Mutual information



Create a correlation plot

```
library(corrr)
house_sales_df %>% select(where(is.numeric)) %>%
  correlate() %>%
  shave() %>%
  rplot(print_cor = TRUE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

Correlation plot



Let's practice!

DIMENSIONALITY REDUCTION IN R

Information and feature importance

DIMENSIONALITY REDUCTION IN R

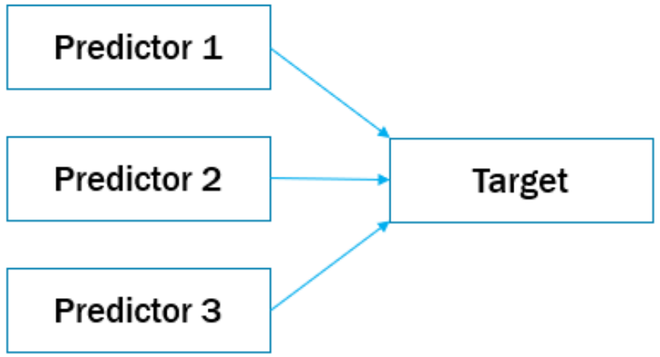
Matt Pickard
Owner, Pickard Predictives, LLC

When the world presents you with very large sets of features, it may be (extremely) useful to hearken back to...the idea of information gain and to select a subset of informative attributes. Doing so can substantially reduce the size of an unwieldy dataset, and...often will improve the accuracy of the resultant model.[1]

[1] Provost, Foster; Fawcett, Tom (2013-07-27). Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media. Kindle Edition.

Feature importance

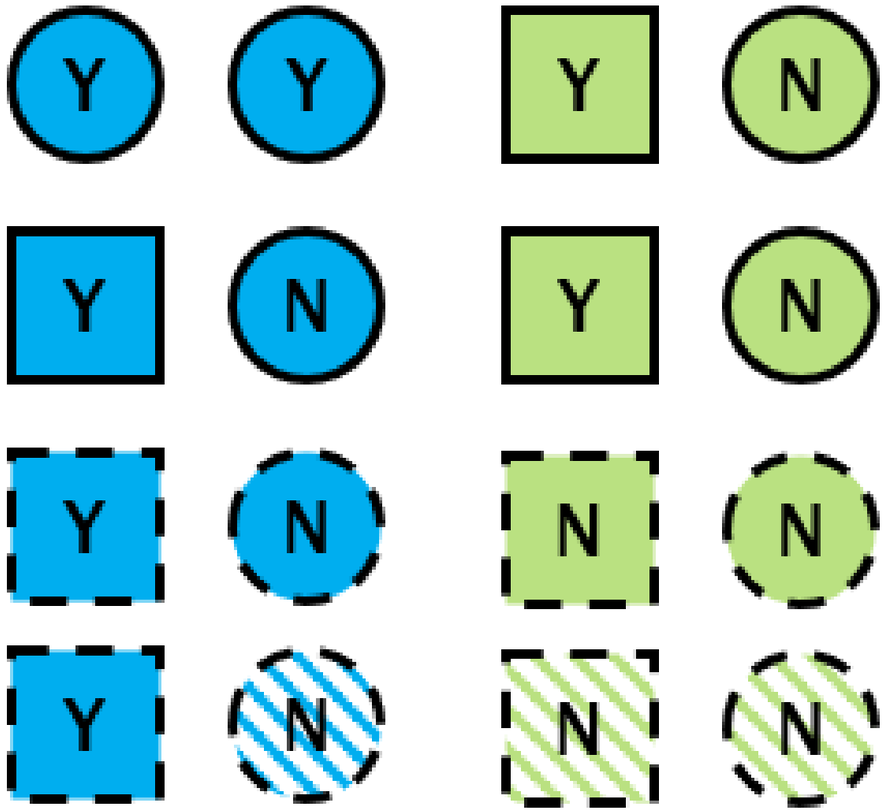
Feature importance: a measure of information in model building



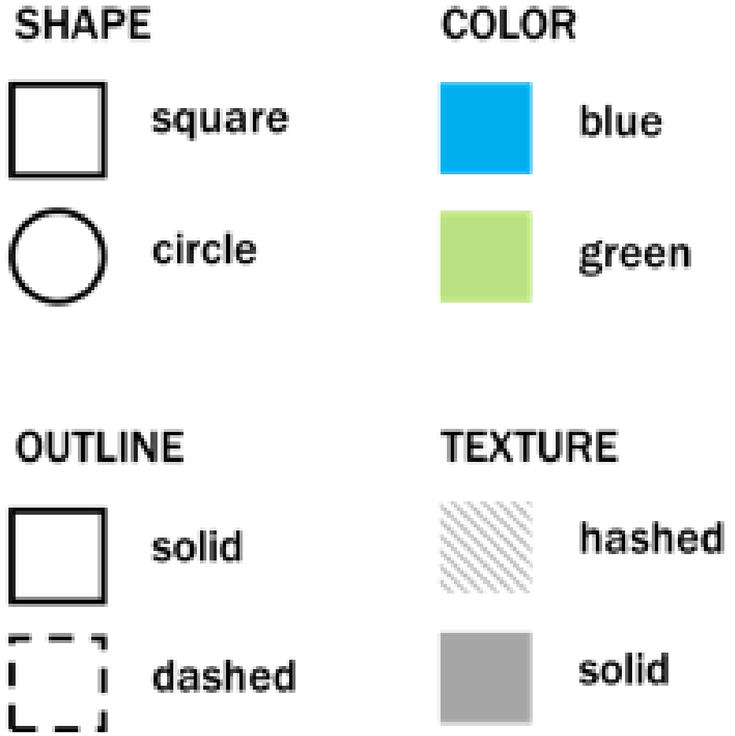
Many ways to measure feature importance

- Correlation (with target variable)
- Standardize regression coefficients
- Information gain

Decision tree example



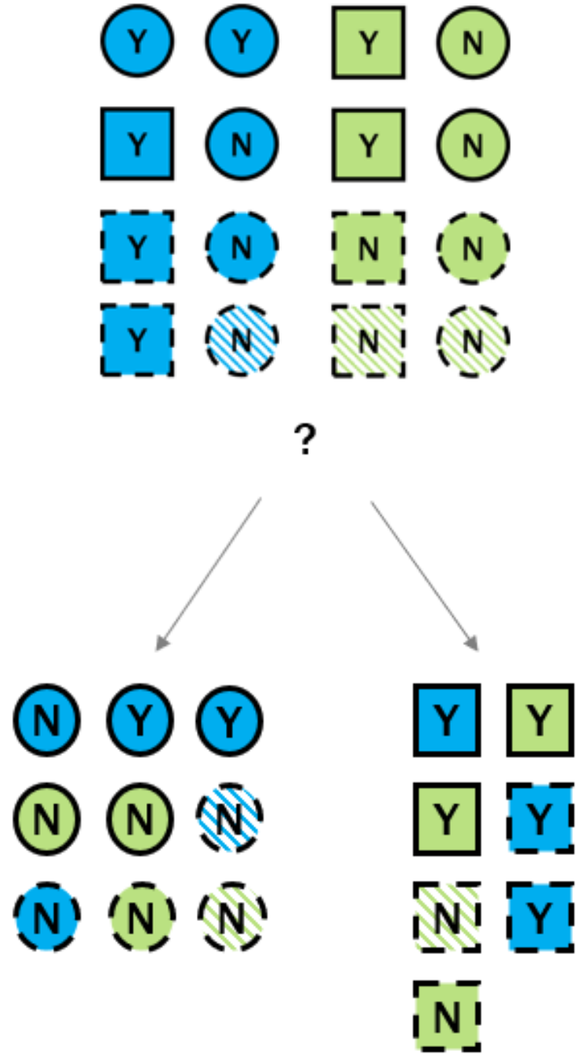
Y = Default
N = No default



Decision tree and information gain

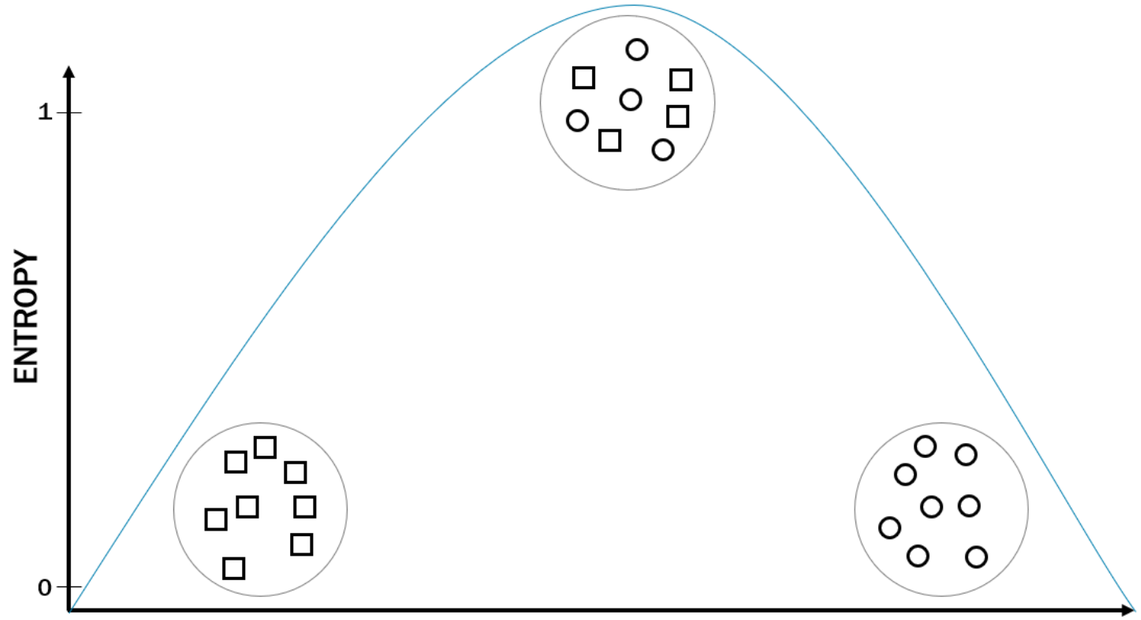
Information gain - the amount of information we know about one variable by observing another variable

$$IG = entropy(parent) - entropy(children)$$



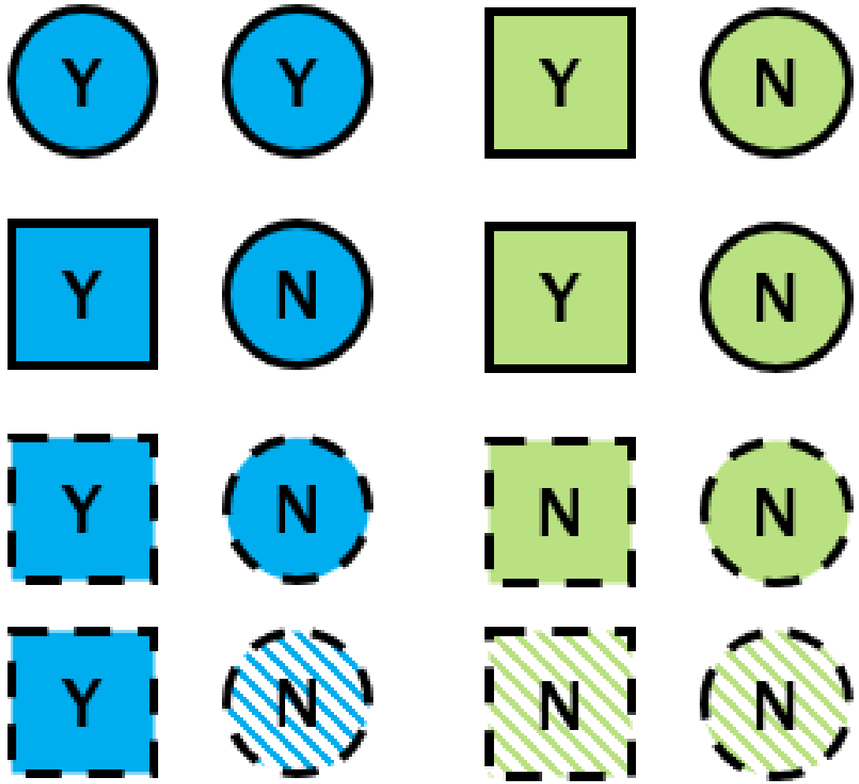
Entropy

- A measure of disorder
- As purity goes up, entropy goes down
- Entropy values range from 0 (perfect purity) to 1 (perfect entropy)



Entropy: root node

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

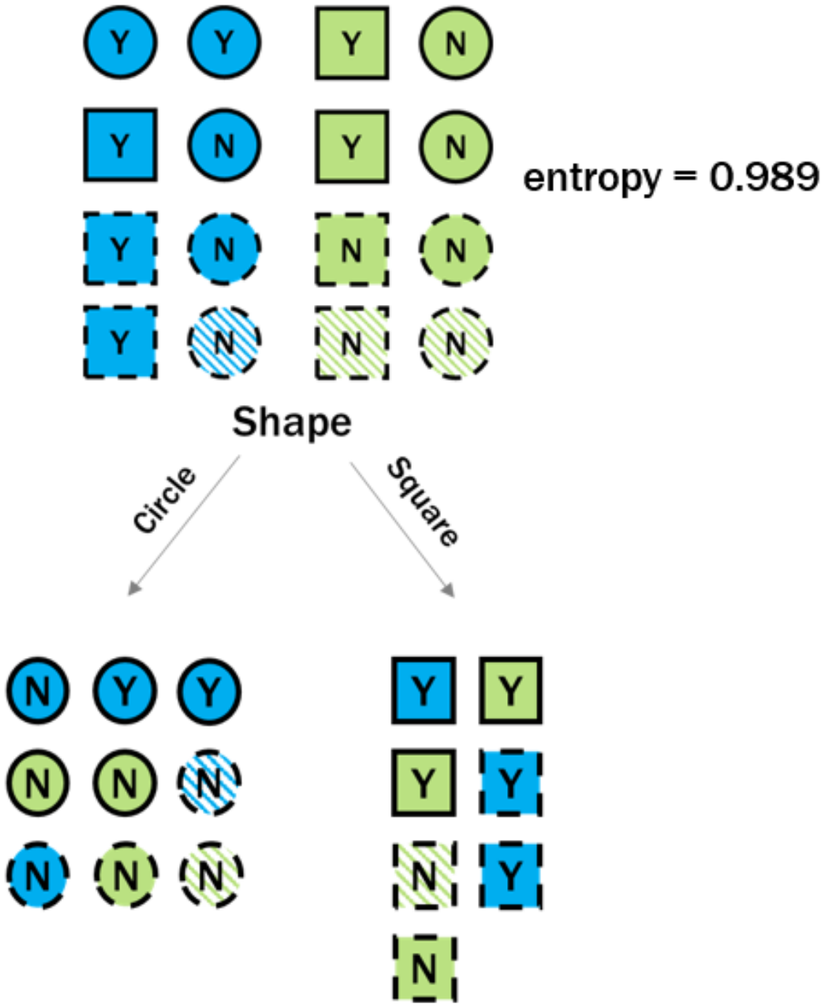



```
p_yes <- 7/16
p_no <- 9/16
entropy_root <-
  -(p_yes * log2(p_yes)) +
  -(p_no * log2(p_no))
entropy_root
```

```
0.989
```

Entropy: children nodes

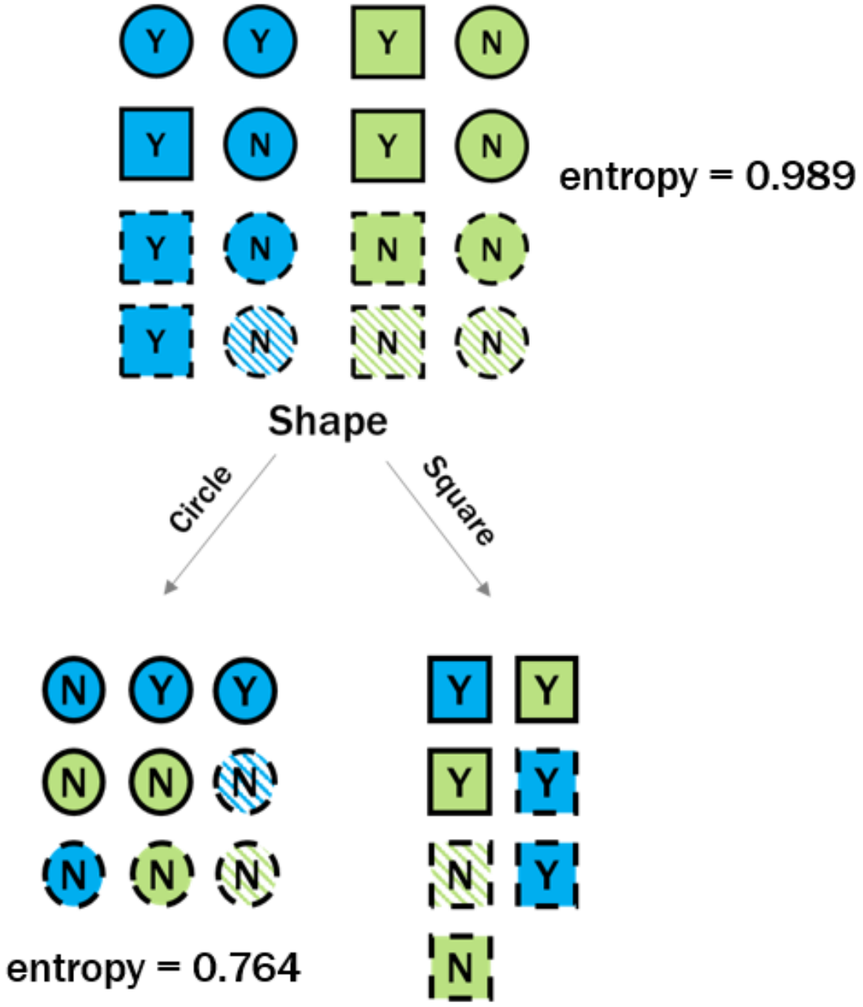
```
p_left_yes <- 2/9
p_left_no <- 7/9

entropy_left <-
  -(p_left_yes * log2(p_left_yes)) +
  -(p_left_no * log2(p_left_no))
```



Entropy: children nodes

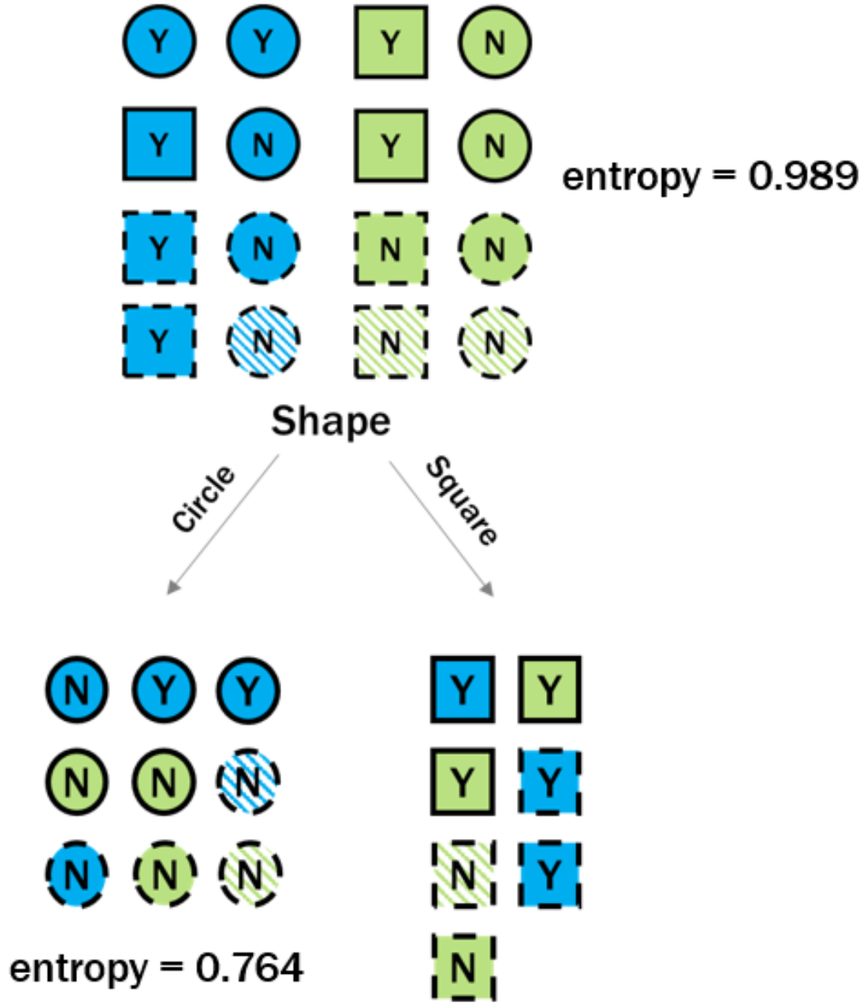
```
p_left_yes <- 2/9
p_left_no <- 7/9

entropy_left <-
  -(p_left_yes * log2(p_left_yes)) +
  -(p_left_no * log2(p_left_no))

entropy_left
```

```
0.764
```



Entropy: children nodes

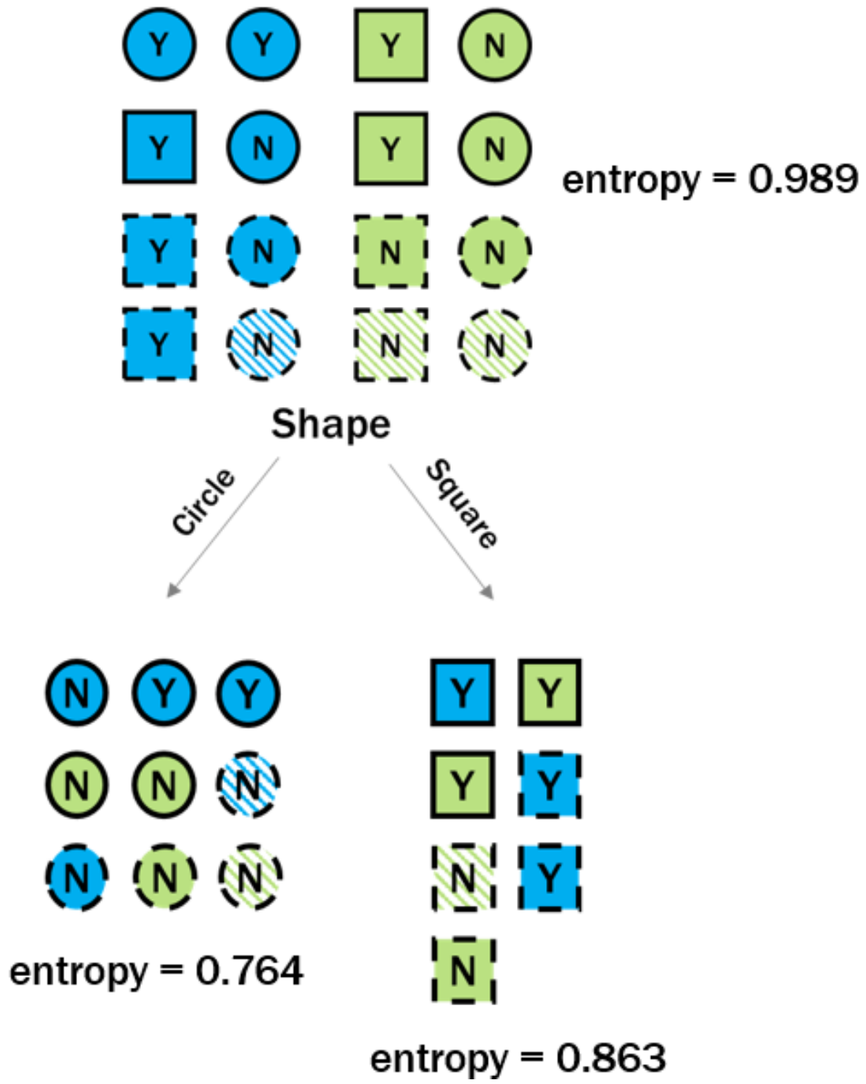
```
p_right_yes <- 5/7
p_right_no <- 2/7

entropy_right <-
  -(p_right_yes * log2(p_right_yes)) +
  -(p_right_no * log2(p_right_no))
```



Entropy: children nodes

```
p_right_yes <- 5/7
p_right_no <- 2/7

entropy_right <-
  -(p_right_yes * log2(p_right_yes)) +
  -(p_right_no * log2(p_right_no))

entropy_right
```

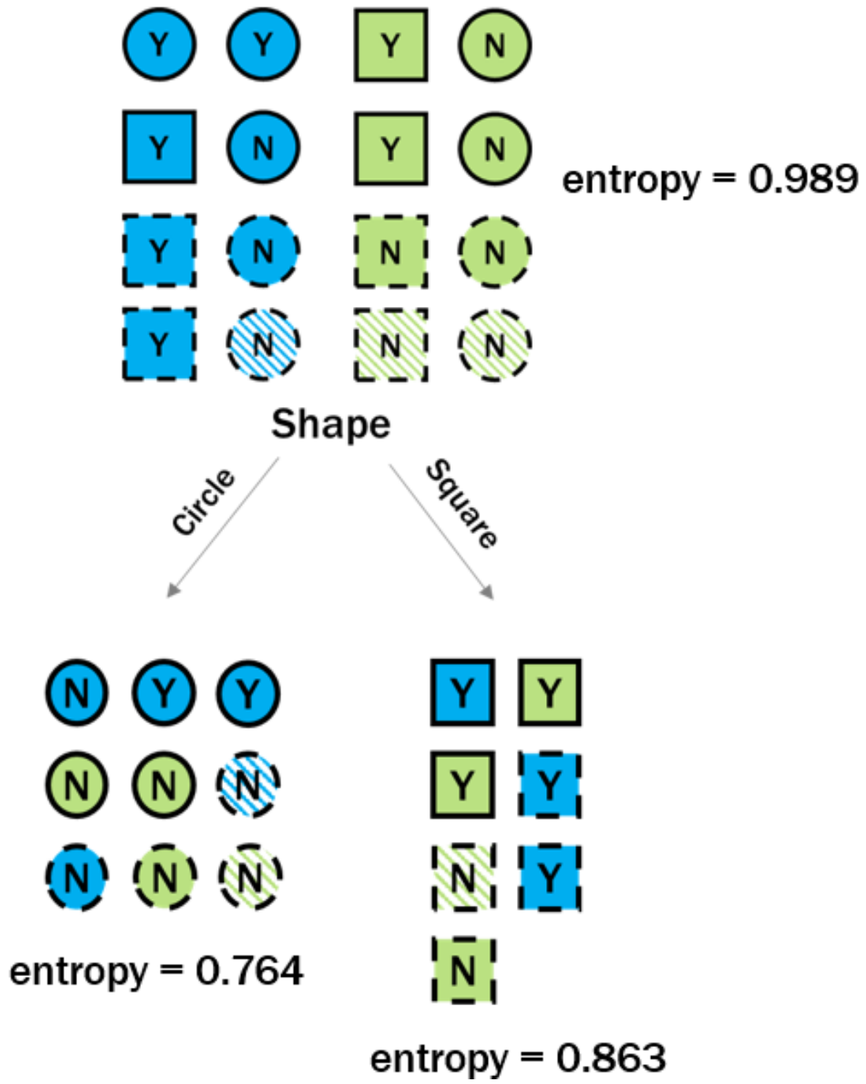
```
0.863
```



Information gain: root to children

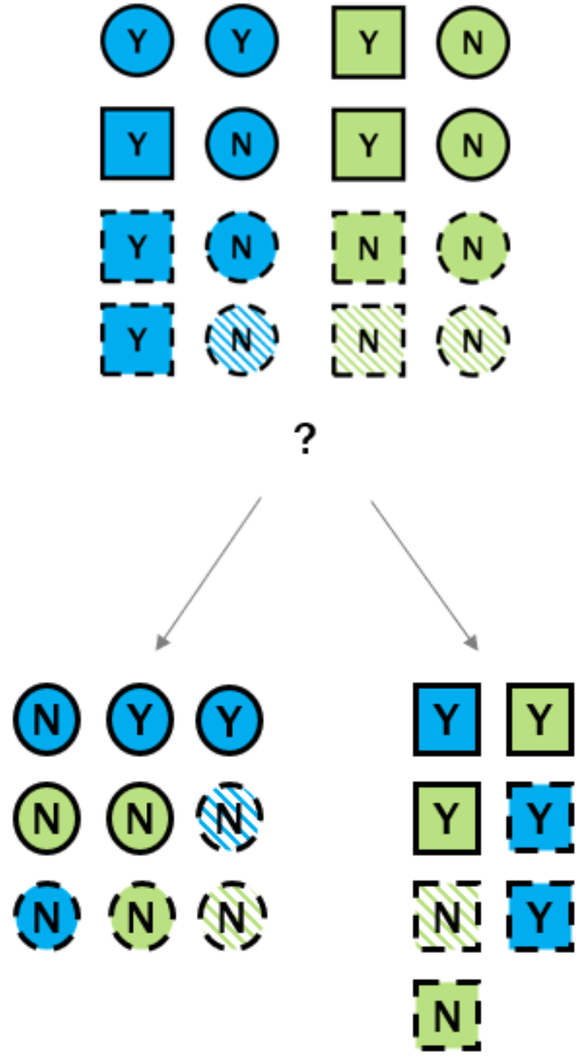
```
p_left <- 9/16
p_right <- 7/16

info_gain <- entropy_root -
  (p_left * entropy_left +
  p_right * entropy_right)

info_gain
```

```
0.181
```



Compare information gain across features

Feature	Information Gain
shape	0.181
texture	0.180
outline	0.106
color	0.106



Let's practice!

DIMENSIONALITY REDUCTION IN R

The Importance of Dimensionality Reduction in Data and Model Building

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

The curse of dimensionality

- a marginal increase in dimensionality requires an exponential increase in data volume
  - data sparsity → bias and overfitting

Gender	Veteran
Female	Yes
Female	No
Male	Yes
Male	No

The curse of dimensionality

- problems dealing with high-dimensional data
- a marginal increase in dimensionality requires an exponential increase in data volume
  - data sparsity → bias and overfitting

Gender	Veteran
Female	Yes
Female	No
Male	Yes
Male	No

2 (Female, Male)
x 2 (Yes, No)
4

The curse of dimensionality

Gender	Veteran	Blood Type
Female	Yes	A
Female	Yes	B
Female	Yes	AB
Female	Yes	O
Female	No	A
Female	No	B
Female	No	AB
Female	No	O
Male	Yes	A
Male	Yes	B
Male	Yes	AB
Male	Yes	O
Male	No	A
Male	No	B
Male	No	AB
Male	No	O

The curse of dimensionality

Gender	Veteran	Blood Type
Female	Yes	A
Female	Yes	B
Female	Yes	AB
Female	Yes	O
Female	No	A
Female	No	B
Female	No	AB
Female	No	O
Male	Yes	A
Male	Yes	B
Male	Yes	AB
Male	Yes	O
Male	No	A
Male	No	B
Male	No	AB
Male	No	O

2 (Female, Male)
2 (Yes, No)
x 4 (A, B, AB, O)
16

Sparsity

All combinations

Gender	Veteran	Blood Type
Female	Yes	A
Female	Yes	B
Female	Yes	AB
Female	Yes	O
Female	No	A
Female	No	B
Female	No	AB
Female	No	O
Male	Yes	A
Male	Yes	B
Male	Yes	AB
Male	Yes	O
Male	No	A
Male	No	B
Male	No	AB
Male	No	O

Sparsity

All combinations

Gender	Veteran	Blood Type
Female	Yes	A
Female	Yes	B
Female	Yes	AB
Female	Yes	O
Female	No	A
Female	No	B
Female	No	AB
Female	No	O
Male	Yes	A
Male	Yes	B
Male	Yes	AB
Male	Yes	O
Male	No	A
Male	No	B
Male	No	AB
Male	No	O

Actual data collected

Gender	Veteran	Blood Type
Male	No	O
Female	No	B
Male	No	O
Female	Yes	O
Female	No	A
Female	No	B
Female	No	AB
Female	No	O
Male	No	A
Male	No	B
Male	Yes	O
Male	Yes	O
Female	Yes	O
Male	No	B
Male	No	O
Male	No	O

Sparsity

All combinations

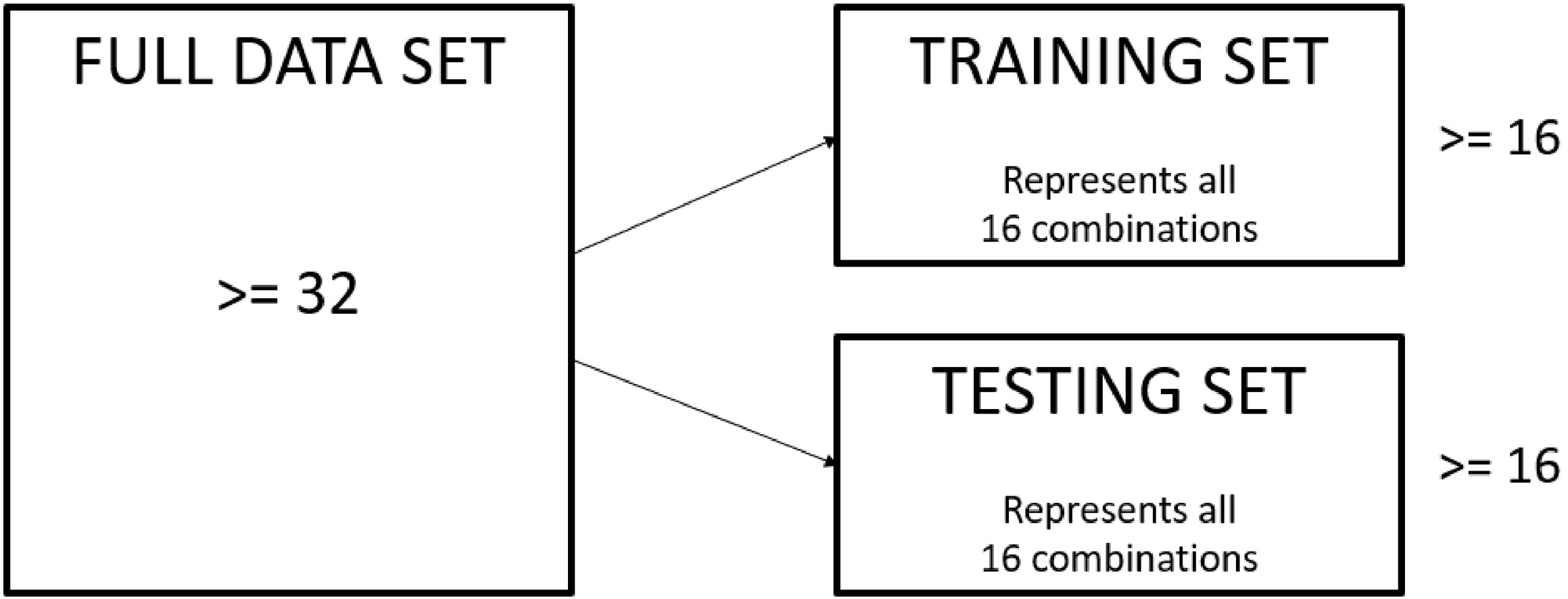
Gender	Veteran	Blood Type
Female	Yes	A
Female	Yes	B
Female	Yes	AB
Female	**Yes**	**O**
Female	**No**	**A**
Female	**No**	**B**
Female	**No**	**AB**
Female	**No**	**O**
Male	Yes	A
Male	Yes	B
Male	Yes	AB
Male	**Yes**	**O**
Male	**No**	**A**
Male	**No**	**B**
Male	No	AB
Male	**No**	**O**

Actual data collected

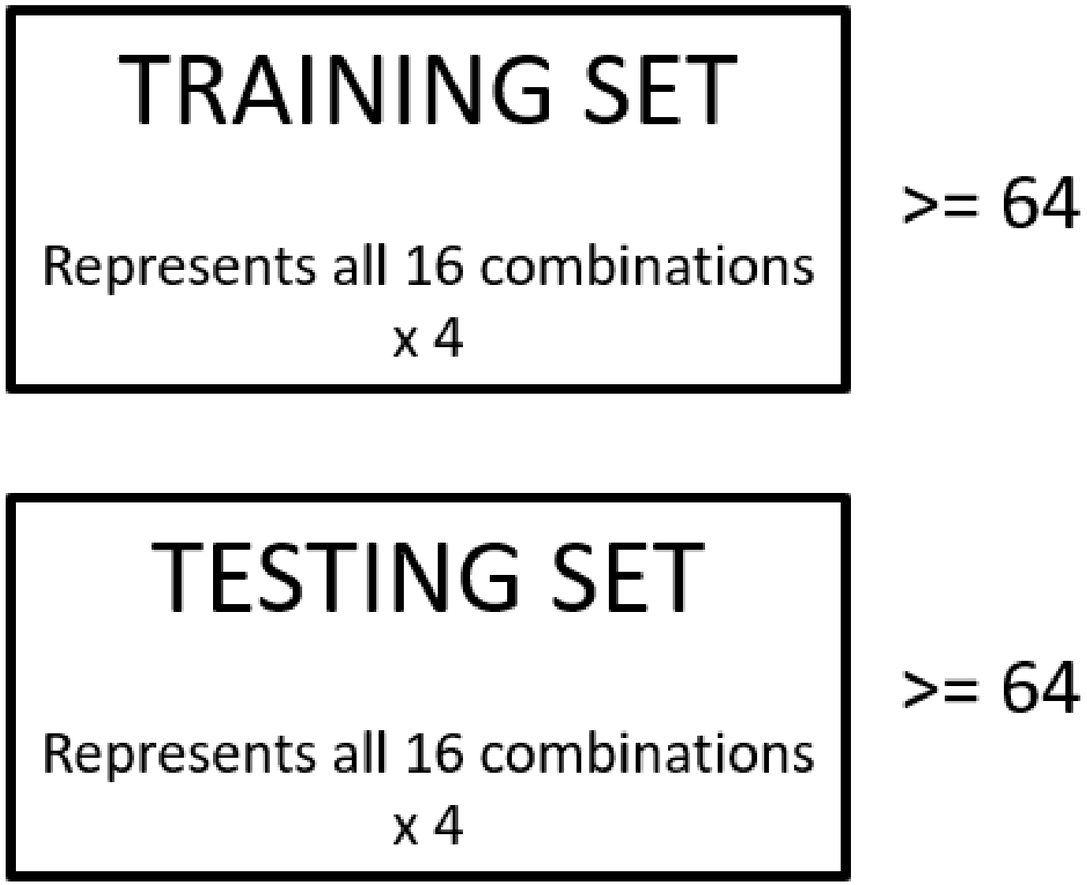
Gender	Veteran	Blood Type
Male	No	O
Female	No	B
Male	No	O
Female	Yes	O
Female	No	A
Female	No	B
Female	No	AB
Female	No	O
Male	No	A
Male	No	B
Male	Yes	O
Male	Yes	O
Female	Yes	O
Male	No	B
Male	No	O
Male	No	O

Sparsity: training and test sets

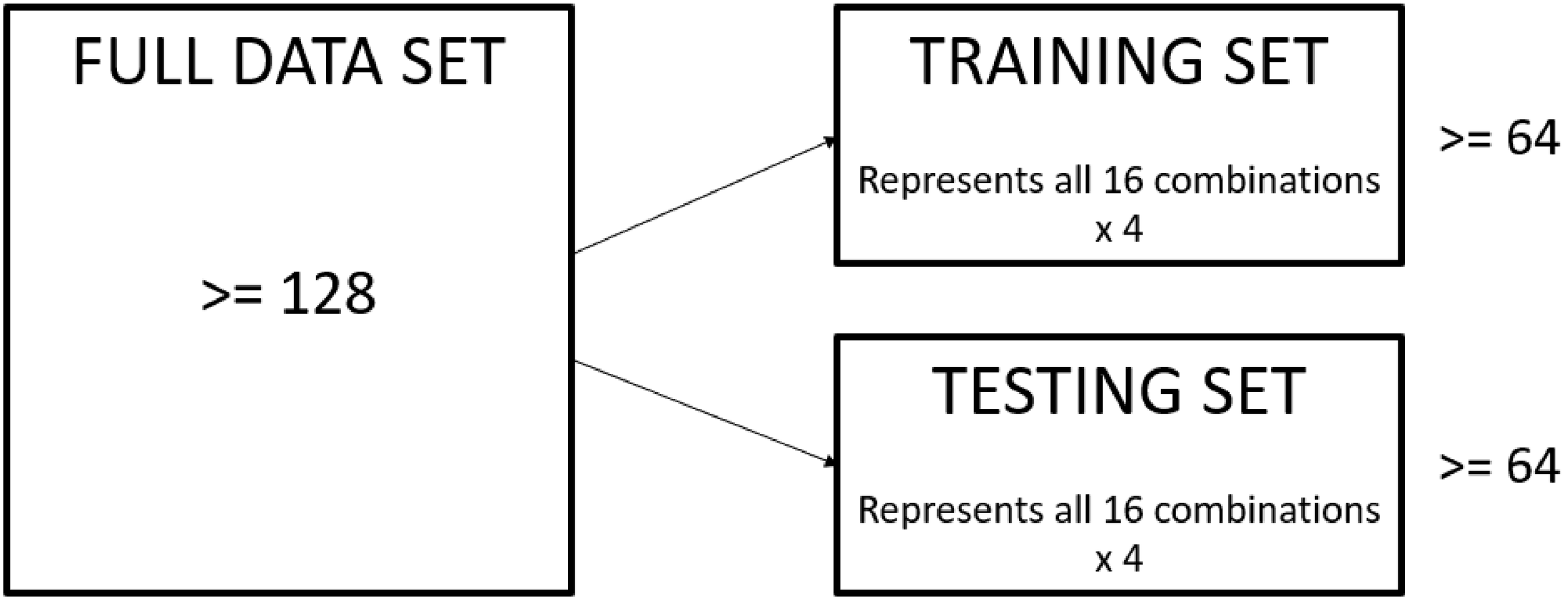
Sparsity: training and test sets



Sparsity: training and test sets



Sparsity: training and test sets



Calculate minimum number of observations

```
blood_type_df <-
  expand_grid(
    gender = c("Female", "Male"),
    veteran = c("Yes", "No"),
    bloodtype = c("A", "B", "AB", "O")
)
```

```
# A tibble: 16 × 3
   gender veteran bloodtype
   <chr>  <chr>   <chr>
 1 Female Yes     A
 2 Female Yes     B
 3 Female Yes     AB
 4 Female Yes     O
 5 Female No      A
 6 Female No      B
 7 Female No      AB
 8 Female No      O
 9 Male   Yes     A
   ...    ...     ...
```

Calculate minimum number of observations

```
blood_type_df %>%
  summarize(across(everything(), ~ length(unique(.)))) %>%
  prod()
```

```
16
```

NOTE: That's the number to represent each combination only once!

Multiple representations of each combination

```
blood_type_df %>%
  summarize(across(everything(), ~ length(unique(.)))) %>%
  prod() * 4
```

```
128
```

Let's practice!

DIMENSIONALITY REDUCTION IN R