



## SUPERVISED LEARNING IN R: REGRESSION

# Categorical inputs

Nina Zumel and John Mount  
Win-Vector, LLC

# Example: Effect of Diet on Weight Loss

```
> WtLoss24 ~ Diet + Age + BMI
```

Diet	Age	BMI	WtLoss24
Med	59	30.67	-6.7
Low-Carb	48	29.59	8.4
Low-Fat	52	32.9	6.3
Med	53	28.92	8.3
Low-Fat	47	30.20	6.3



# model.matrix()

```
> model.matrix(WtLoss24 ~ Diet + Age + BMI, data = diet)
```

- All numerical values
- Converts categorical variable with N levels into N - 1 indicator variables

# Indicator Variables to Represent Categories

## Original Data

Diet	Age	...
Med	59	...
Low-Carb	48	...
Low-Fat	52	...
Med	53	...
Low-Fat	47	...

## Model Matrix

(Intercept)	DietLow-Fat	DietMed	...
1	0	1	...
1	0	0	...
1	1	0	...
1	0	1	...
1	1	0	...

- reference level: "Low-Carb"



# Interpreting the Indicator Variables

## Linear Model:

$$WtLoss24 = \beta_0 + \beta_{DietLowFat}x_{DietLowFat} + \beta_{DietMed}x_{DietMed} + \beta_{Age}x_{Age} + \beta_{BMI}x_{BMI}$$

```
> lm(WtLoss24 ~ Diet + Age + BMI, data = diet))  
  
## Coefficients:  
##      (Intercept)      DietLow-Fat      DietMed  
##      -1.37149      -2.32130      -0.97883  
##           Age           BMI  
##           0.12648           0.01262
```



# Issues with one-hot-encoding

- Too many levels can be a problem
  - Example: ZIP code (about 40,000 codes)
- Don't hash with geometric methods!



## SUPERVISED LEARNING IN R: REGRESSION

**Let's practice!**



## SUPERVISED LEARNING IN R: REGRESSION

# Interactions

Nina Zumel and John Mount  
Win-Vector, LLC





# Additive relationships

Example of an additive relationship:

```
> plant_height ~ bacteria + sun
```

- Change in height is the sum of the effects of bacteria and sunlight
  - Change in sunlight causes same change in height, independent of bacteria
  - Change in bacteria causes same change in height, independent of sunlight



# What is an Interaction?

*The simultaneous influence of two variables on the outcome is not additive.*

```
> plant_height ~ bacteria + sun + bacteria:sun
```

- Change in height is more (or less) than the sum of the effects due to sun/bacteria
- At higher levels of sunlight, 1 unit change in bacteria causes more change in height



# What is an Interaction?

*The simultaneous influence of two variables on the outcome is not additive.*

```
> plant_height ~ bacteria + sun + bacteria:sun
```

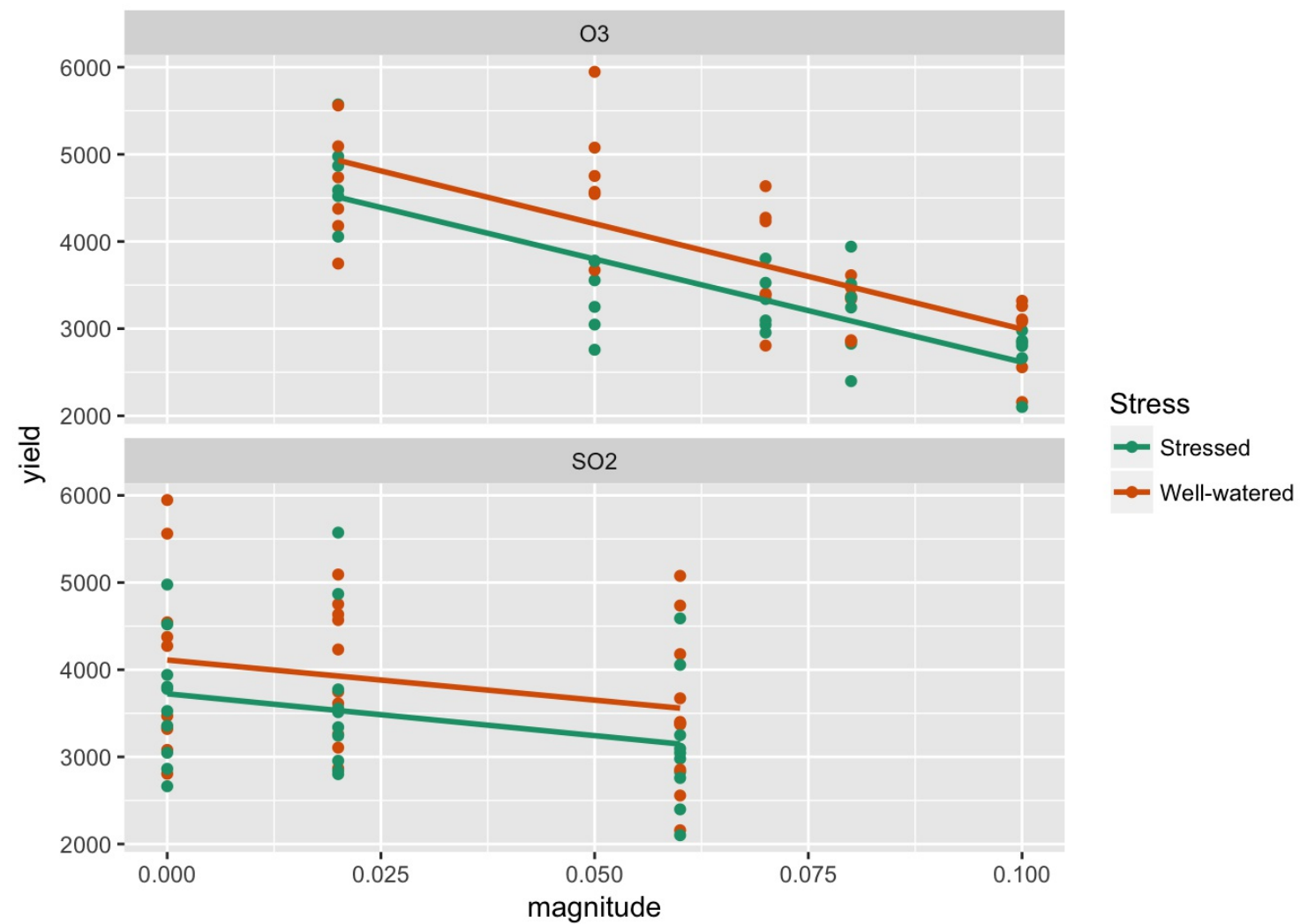
- sun: categorical {"sun", "shade"}
- In sun, 1 unit change in bacteria causes  $m$  units change in height
- In shade, 1 unit change in bacteria causes  $n$  units change in height

Like two separate models: one for sun, one for shade.



# Example of no Interaction: Soybean Yield

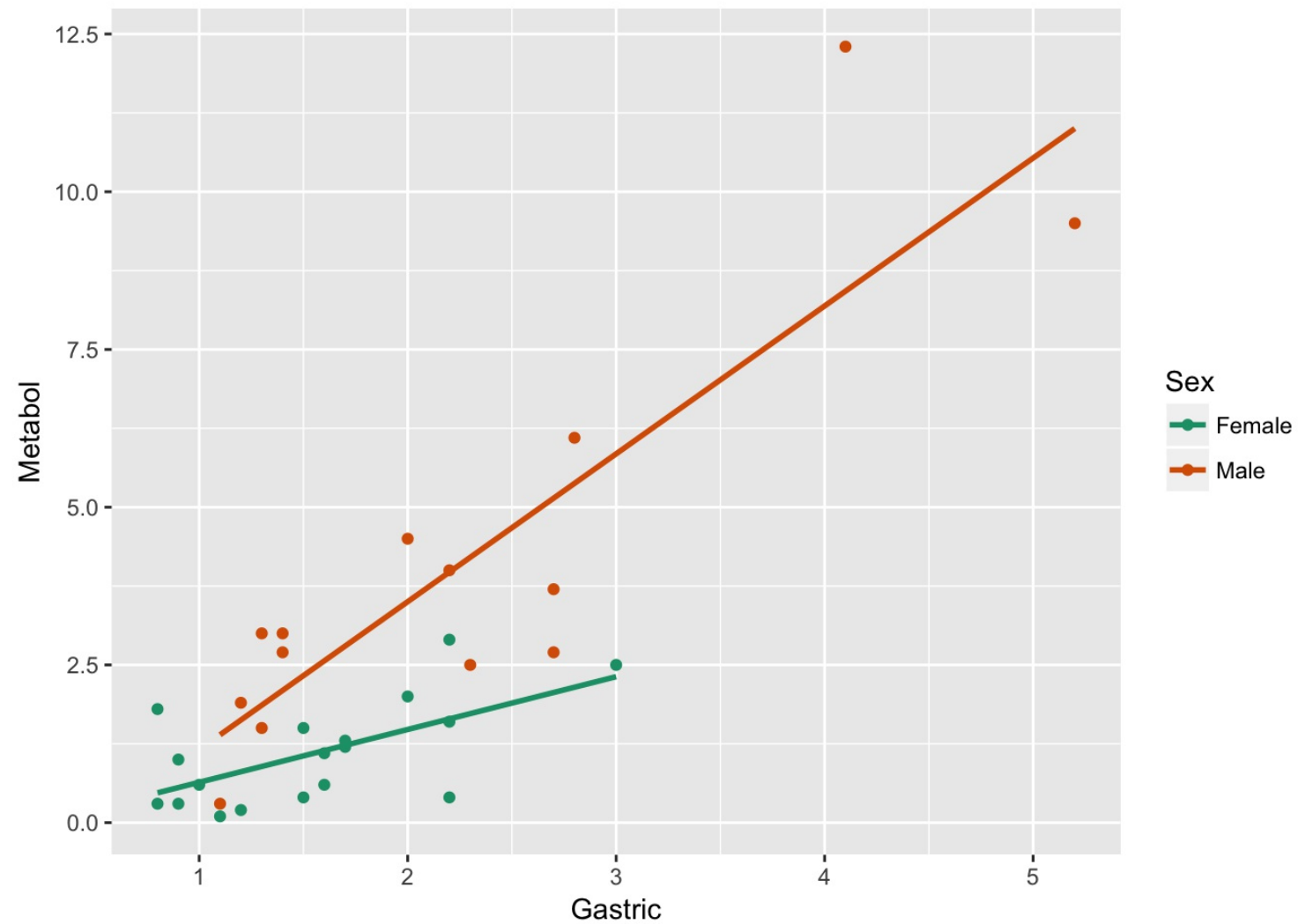
```
> yield ~ Stress + SO2 + O3
```





# Example of an Interaction: Alcohol Metabolism

```
> Metabol ~ Gastric + Sex
```



# Expressing Interactions in Formulae

- Interaction - Colon (:)

```
> y ~ a:b
```

- Main effects and interaction - Asterisk (\*)

```
> y ~ a*b  
# Both mean the same  
> y ~ a + b + a:b
```

- Expressing the product of two variables - **I**

```
> y ~ I(a*b)
```

same as  $y \propto ab$



# Finding the Correct Interaction Pattern

Formula	RMSE (cross validation)
Metabol ~ Gastric + Sex	1.46
Metabol ~ Gastric * Sex	1.48
Metabol ~ Gastric + Gastric:Sex	1.39



## SUPERVISED LEARNING IN R: REGRESSION

**Let's practice!**





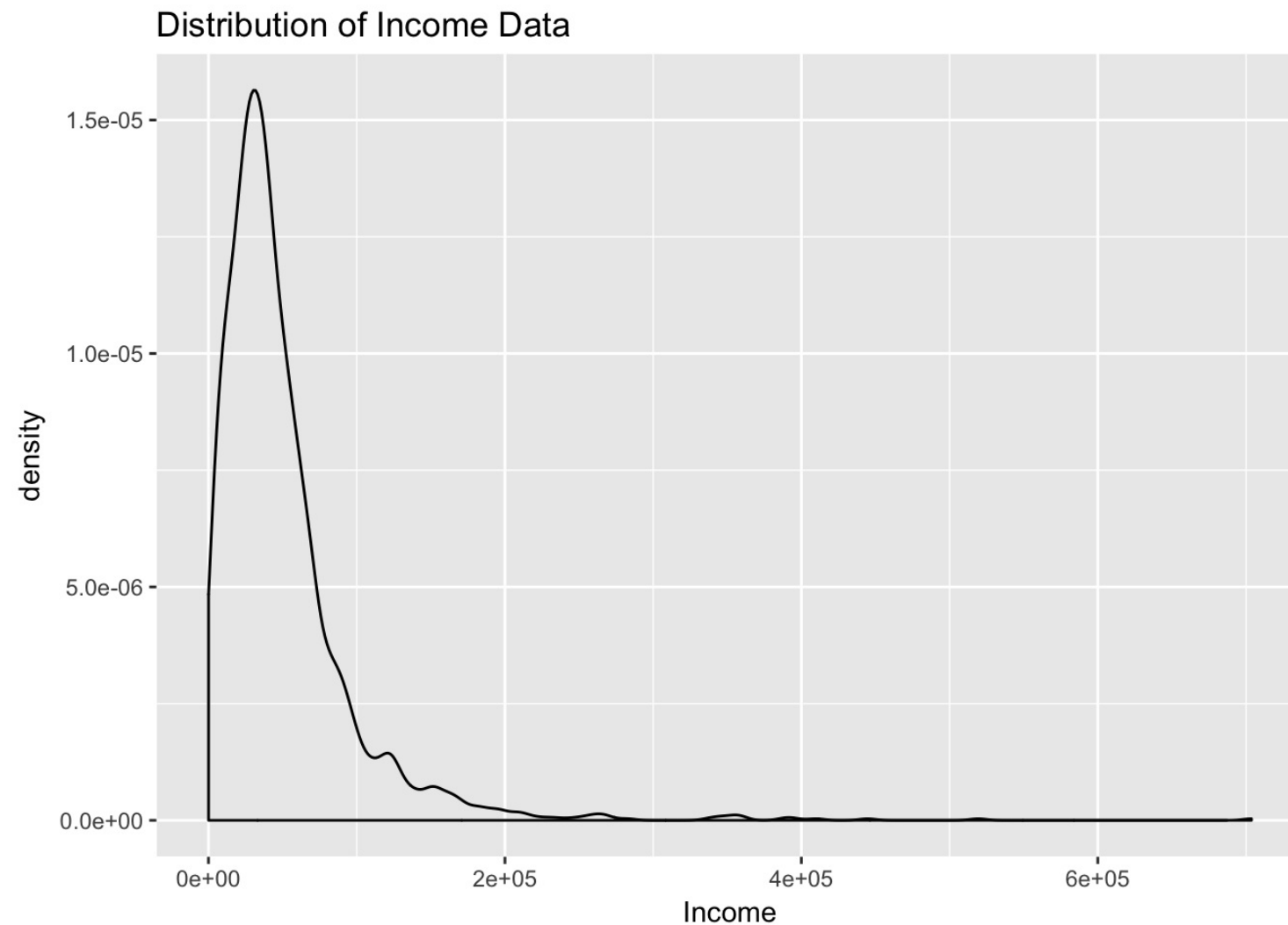
SUPERVISED LEARNING IN R: REGRESSION

# Transforming the response before modeling

Nina Zumel and John Mount  
Win-Vector, LLC



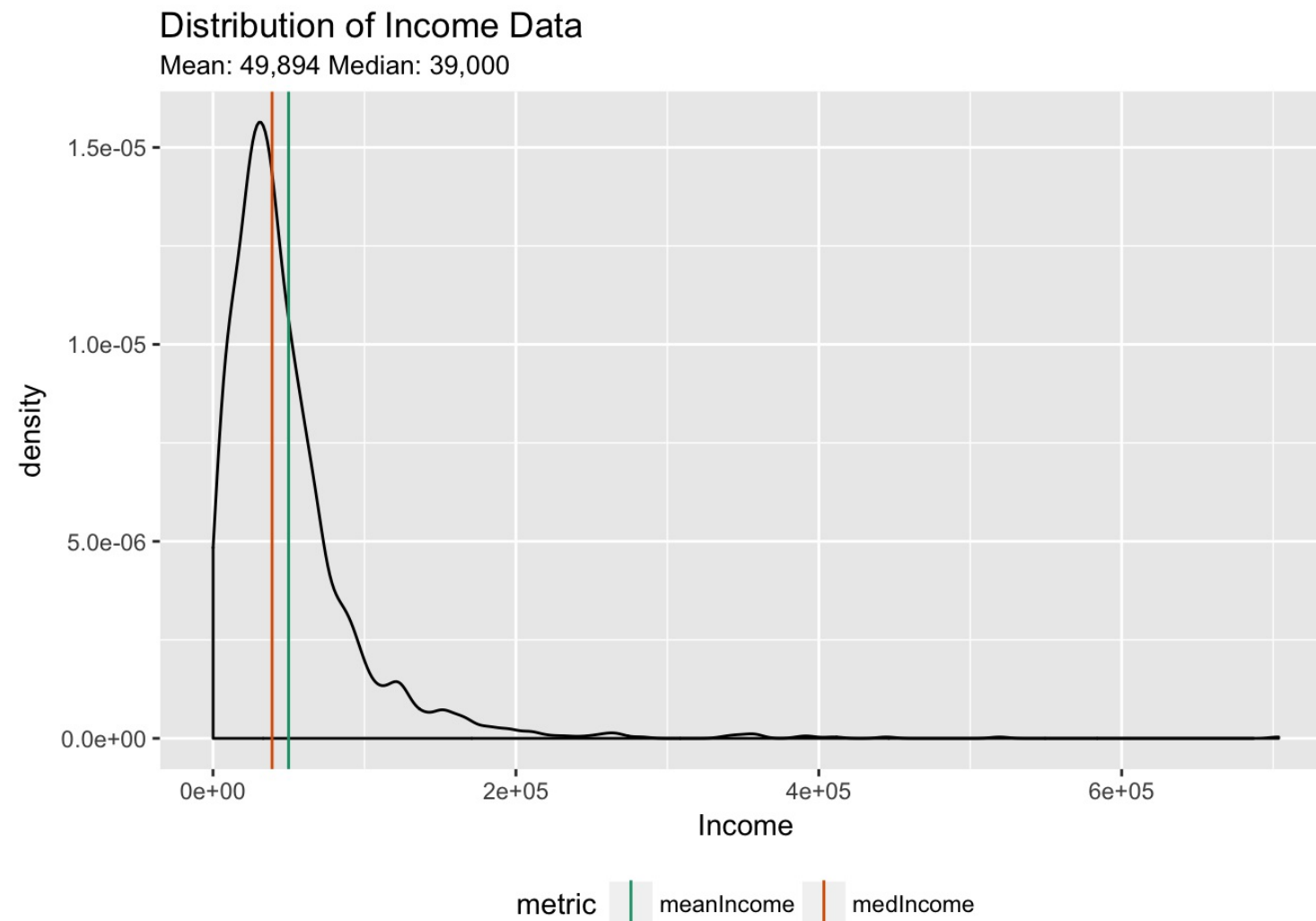
# The Log Transform for Monetary Data



- Monetary values: lognormally distributed
- Long tail, wide dynamic range (60-700K)



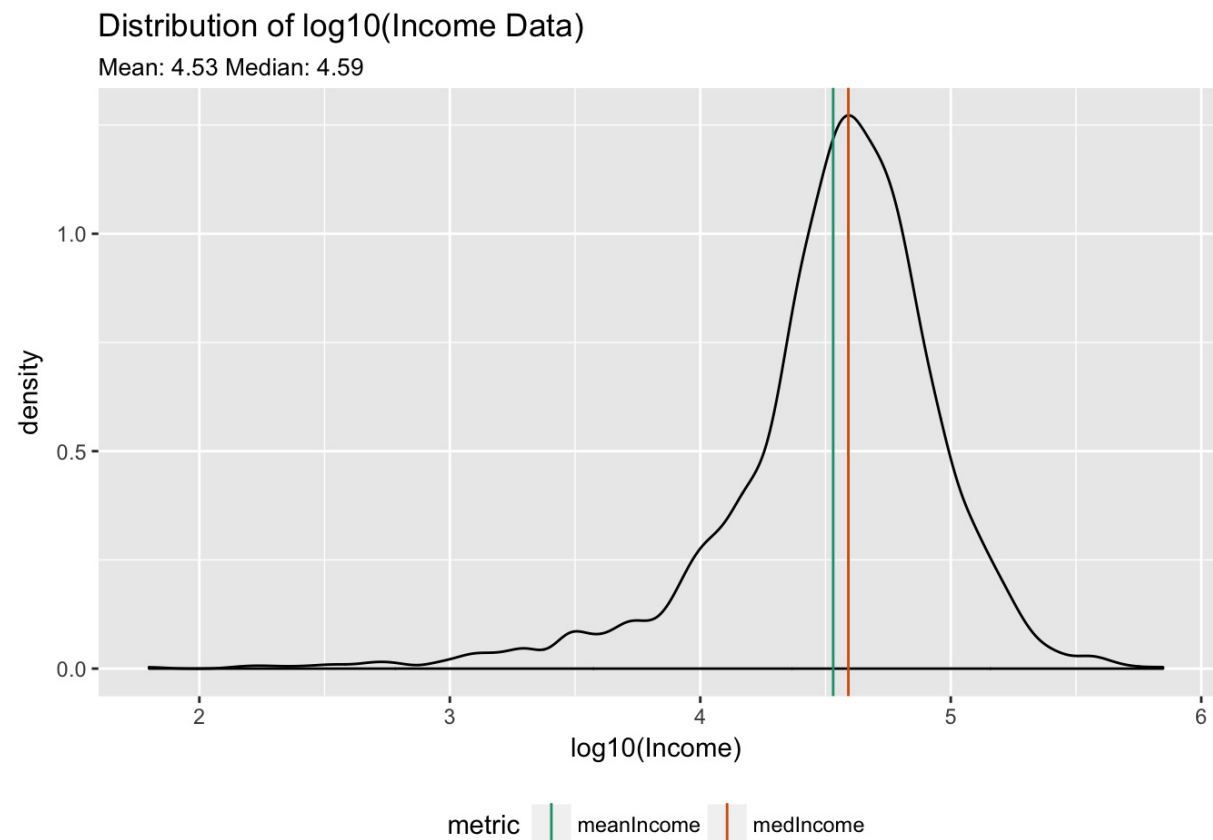
# Lognormal Distributions



- mean > median (~ 50K vs 39K)
- Predicting the mean will overpredict typical values



# Back to the Normal Distribution



For a Normal Distribution:

- mean = median (here: 4.53 vs 4.59)
- more reasonable dynamic range (1.8 - 5.8)



# The Procedure

1. Log the outcome and fit a model

```
> model <- lm(log(y) ~ x, data = train)
```



# The Procedure

1. Log the outcome and fit a model

```
> model <- lm(log(y) ~ x, data = train)
```

2. Make the predictions in log space

```
> logpred <- predict(model, data = test)
```

# The Procedure

1. Log the outcome and fit a model

```
> model <- lm(log(y) ~ x, data = train)
```

2. Make the predictions in log space

```
> logpred <- predict(model, data = test)
```

3. Transform the predictions to outcome space

```
> pred <- exp(logpred)
```



# Predicting Log-transformed Outcomes: Multiplicative Error

$$\log(a) + \log(b) = \log(ab)$$

$$\log(a) - \log(b) = \log(a/b)$$

- Multiplicative error:  $pred/y$
- Relative error:  $(pred - y)/y = \frac{pred}{y} - 1$

*Reducing multiplicative error reduces relative error.*





# Root Mean Squared Relative Error

$$\text{RMS-relative error} = \sqrt{\left(\frac{\text{pred} - y}{y}\right)^2}$$

- Predicting log-outcome reduces RMS-relative error
- But the model will often have larger RMSE



# Example: Model Income Directly

```
> modIncome <- lm(Income ~ AFQT + Educ, data = train)
```

- AFQT: Score on proficiency test 25 years before survey
- Educ: Years of education to time of survey
- Income: Income at time of survey



# Model Performance

```
> test %>%  
+   mutate(pred = predict(modIncome, newdata = test),  
+           err = pred - Income) %>%  
+   summarize(rmse = sqrt(mean(err^2)),  
+             rms.relerr = sqrt(mean((err/Income)^2)))
```

RMSE	RMS-relative error
36,819.39	3.295189



# Model log(Income)

```
> modLogIncome <- lm(log(Income) ~ AFQT + Educ, data = train)
```

# Model Performance

```
> test %>%  
+   mutate(predlog = predict(modLogIncome, newdata = test),  
+           pred = exp(predlog),  
+           err = pred - Income) %>%  
+   summarize(rmse = sqrt(mean(err^2)),  
+             rms.relerr = sqrt(mean((err/Income)^2)))
```

RMSE	RMS-relative error
38,906.61	2.276865



# Compare Errors

log(Income) model: smaller RMS-relative error, larger RMSE

Model	RMSE	RMS-relative error
On Income	36,819.39	3.295189
On log(Income)	38,906.61	2.276865



## SUPERVISED LEARNING IN R: REGRESSION

**Let's practice!**



SUPERVISED LEARNING IN R: REGRESSION

# Transforming inputs before modeling

Nina Zumel and John Mount  
Win-Vector LLC





# Why To Transform Input Variables

- Domain knowledge/synthetic variables
  - Intelligence  $\sim mass.brain/mass.body^{2/3}$



# Why To Transform Input Variables

- Domain knowledge/synthetic variables
  - Intelligence  $\sim mass.brain/mass.body^{2/3}$
- Pragmatic reasons
  - Log transform to reduce dynamic range
  - Log transform because meaningful changes in variable are multiplicative

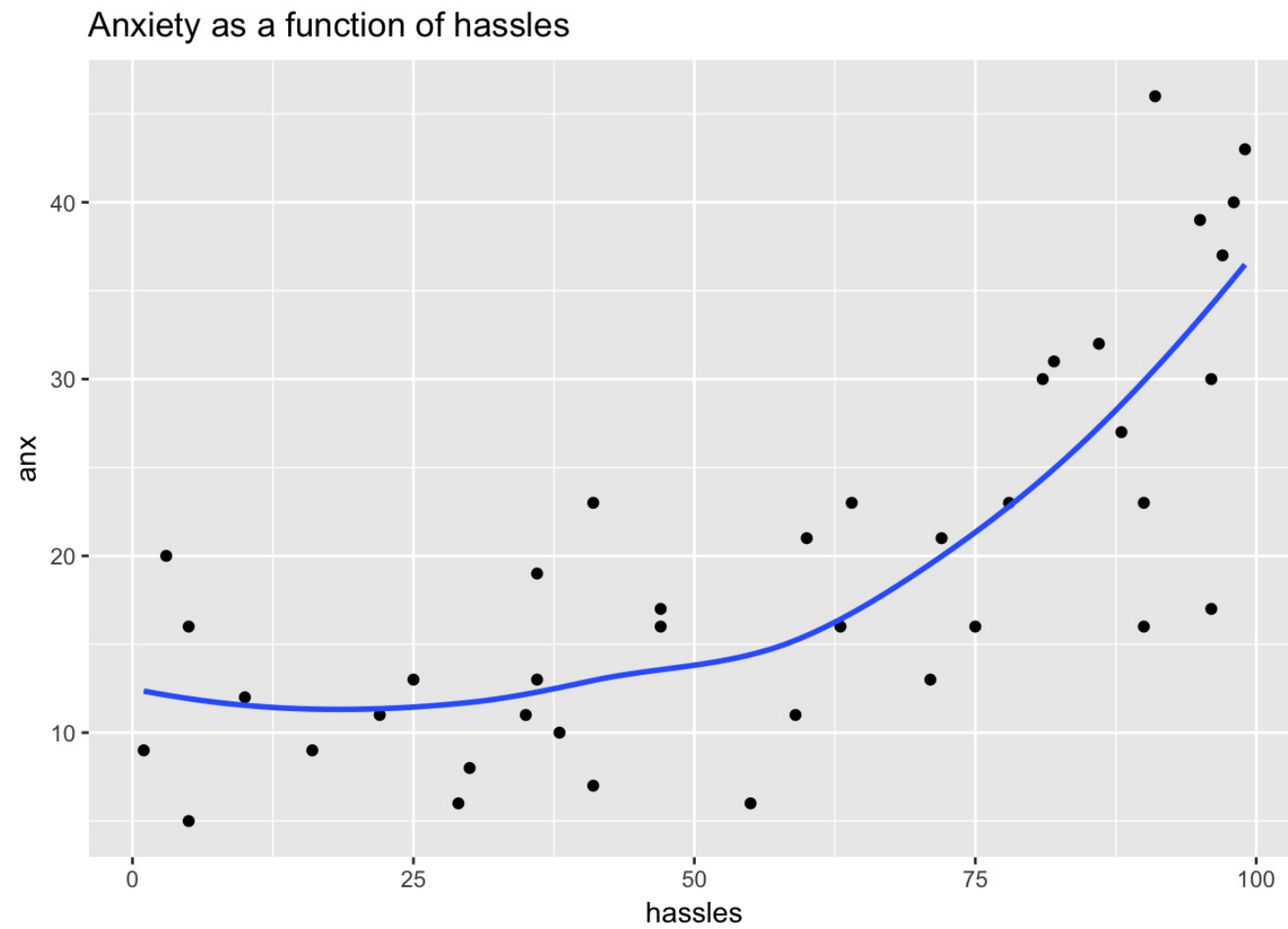


# Why To Transform Input Variables

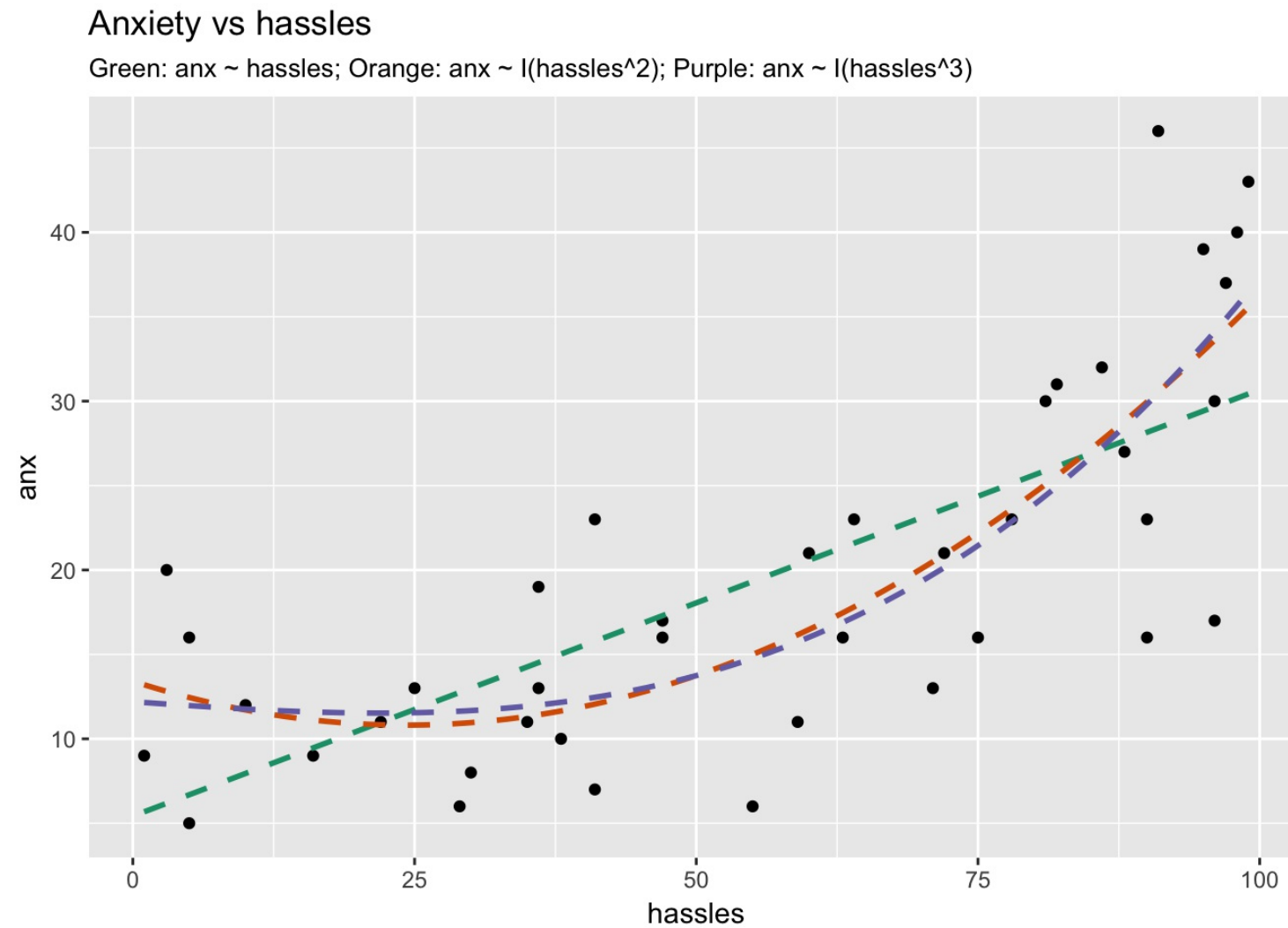
- Domain knowledge/synthetic variables
  - Intelligence  $\sim mass.brain/mass.body^{2/3}$
- Pragmatic reasons
  - Log transform to reduce dynamic range
  - Log transform because meaningful changes in variable are multiplicative
  - $y$  approximately linear in  $f(x)$  rather than in  $x$



# Example: Predicting Anxiety



# Transforming the hassles variable





# Different possible fits

## Which is best?

- $\text{anx} \sim \text{l(hassles}^2)$
- $\text{anx} \sim \text{l(hassles}^3)$
- $\text{anx} \sim \text{l(hassles}^2) + \text{l(hassles}^3)$
- $\text{anx} \sim \exp(\text{hassles})$
- ...

`l()`: treat an expression literally (not as an interaction)



# Compare different models

## Linear, Quadratic, and Cubic models

```
> mod_lin <- lm(anx ~ hassles, hassleframe)
> summary(mod_lin)$r.squared
[1] 0.5334847

> mod_quad <- lm(anx ~ I(hassles^2), hassleframe)
> summary(mod_quad)$r.squared
[1] 0.6241029

> mod_tritic <- lm(anx ~ I(hassles^3), hassleframe)
> summary(mod_tritic)$r.squared
[1] 0.6474421
```



# Compare different models

Use cross-validation to evaluate the models

Model	RMSE
Linear ( <i>hassles</i> )	7.69
Quadratic ( <i>hassles</i> <sup>2</sup> )	6.89
Cubic ( <i>hassles</i> <sup>3</sup> )	6.70





## SUPERVISED LEARNING IN R: REGRESSION

**Let's practice!**