Introduction to Statistical Learning

Eltecon Data Science Course by Emarsys

Péter Lukács

November 6, 2019

About me

- Eltecon BSc
- University of Amsterdam MSc in Economics
- Last 6+ years working with data
  - 2.5 year @ Emarsys as a Data Scientist
- lukacs.peter.andras@gmail.com

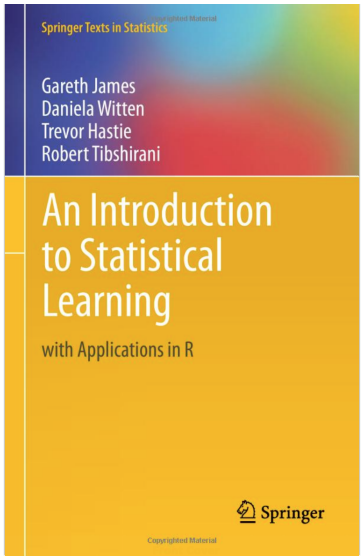
Goal of the lesson

- touch on stat learning basic theory (with examples!)
- see/try basic `R` linreg and classification syntax
- give you resources to reach for when you'll need it

Section 1

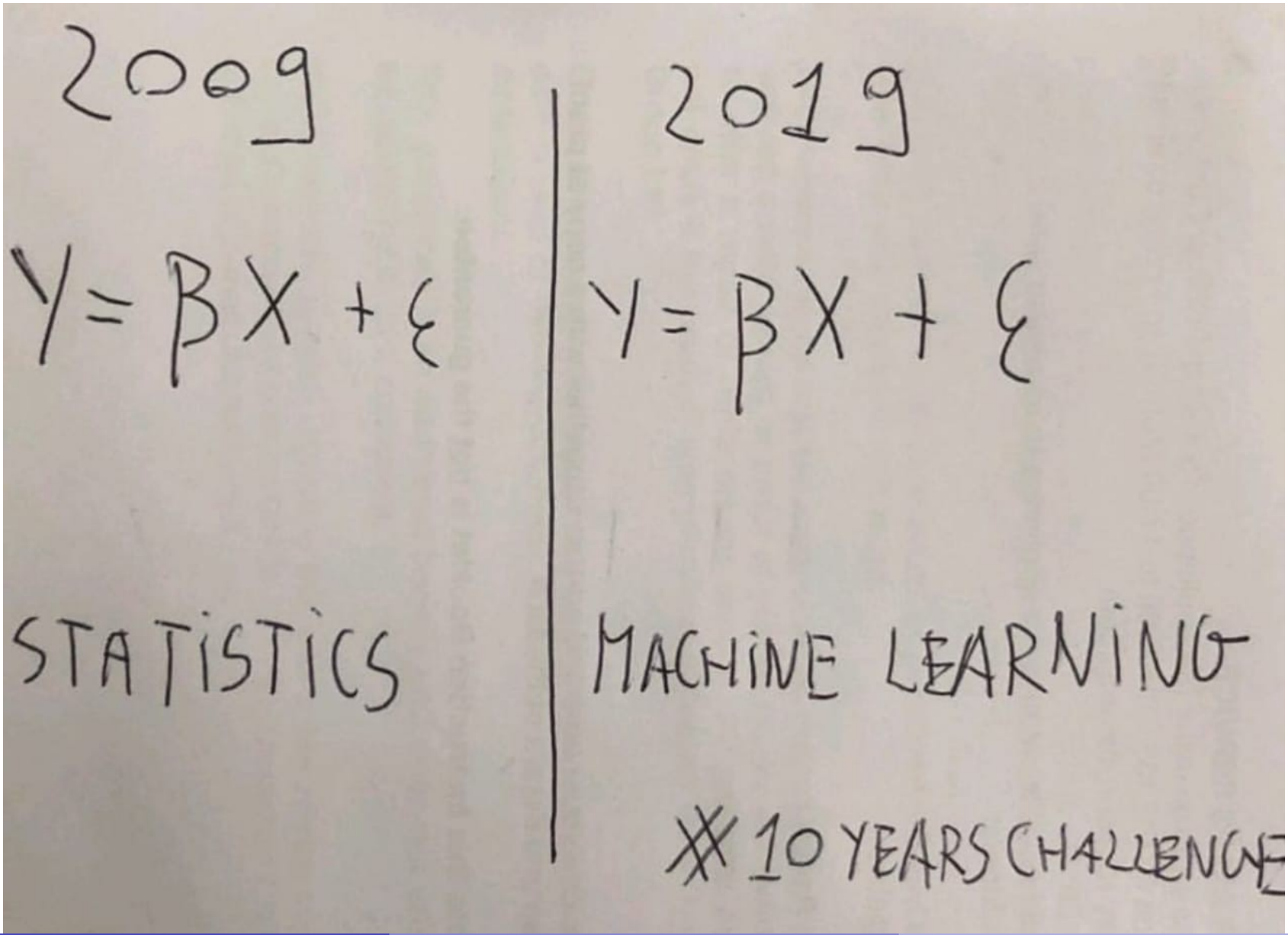
Statistical Learning in General

Introduction to Statistical Learning



- What is Statistical Learning? [pp. 15 - 28]
- Simple Linear Regression [pp. 61 - 70]
- Multiple Linear Regression [pp. 71 - 81]
- Classification [pp. 128 - 137]
- Plus `Lab` from the end of the chapters

What is Statistical Learning?



What is Statistical Learning?

*"**Machine learning** is all about results, it is likely working in a company where your worth is characterized solely by your performance. Whereas, **statistical modeling** is more about finding relationships between variables and the significance of those relationships, whilst also catering for prediction"*

source

What is Statistical Learning?

Assumption:

$$Y = f(X) + \epsilon$$

- We **assume** a systematic relationship between X and Y
- f represents the systematic information that X provides about Y and is generally unknown
- **Statistical Learning refers to a set of approaches for estimating f based on the available observations (X)**

What is Statistical Learning?

Assumption:

$$Y = f(X) + \epsilon$$

- ϵ is assumed to have mean 0
- ϵ is assumed to be independent of X

$\Rightarrow$ **otherwise** could be modeled through f

What is Statistical Learning?: Reducible error/Irreducible error

$$Y = f(X) + \epsilon$$

$$\begin{aligned} E(Y - \hat{Y}) &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{reducible error}} + \underbrace{Var(\epsilon)}_{\text{irreducible error}} \end{aligned}$$

- the aim is to estimate f by reducing the reducible error
- What about the irreducible error? Can't do anything about that.
  - Didn't measure :(
  - Can't measure: e.g. mood of a buyer on the day she's buying the house

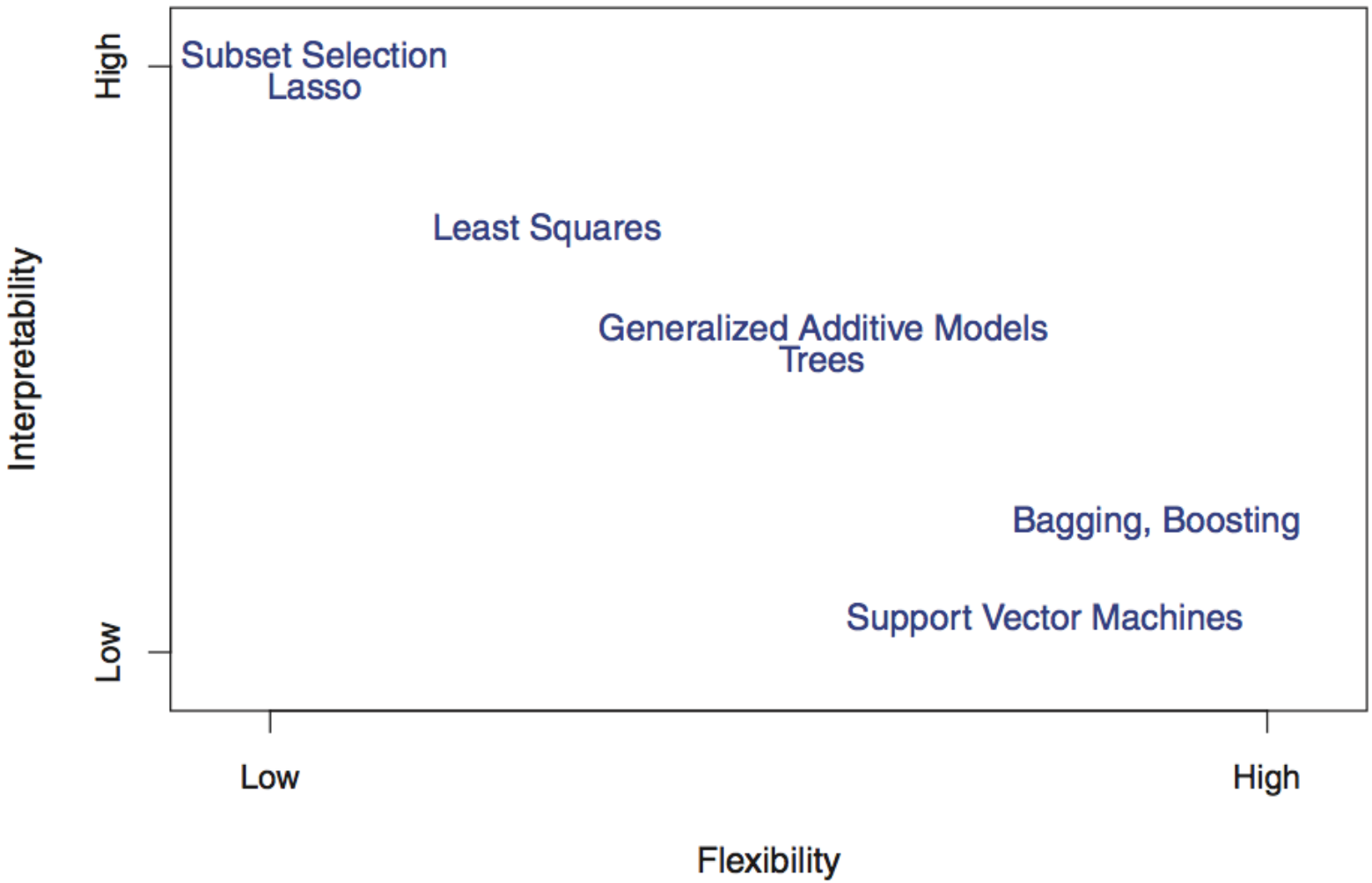
Why estimate f?

- Causality/Inference (more in Econ, e.g. What drives unemployment?)
- Prediction (more in Business, e.g. How much Happy Socks are we selling next month?)

How to estimate f?

- parametric models
  + less parameters to learn (needs less training data)
  - can erroneously assume f
- non-parametric models
  + more flexible
  - more parameters to learn (needs more training data)
  - can overfit the data

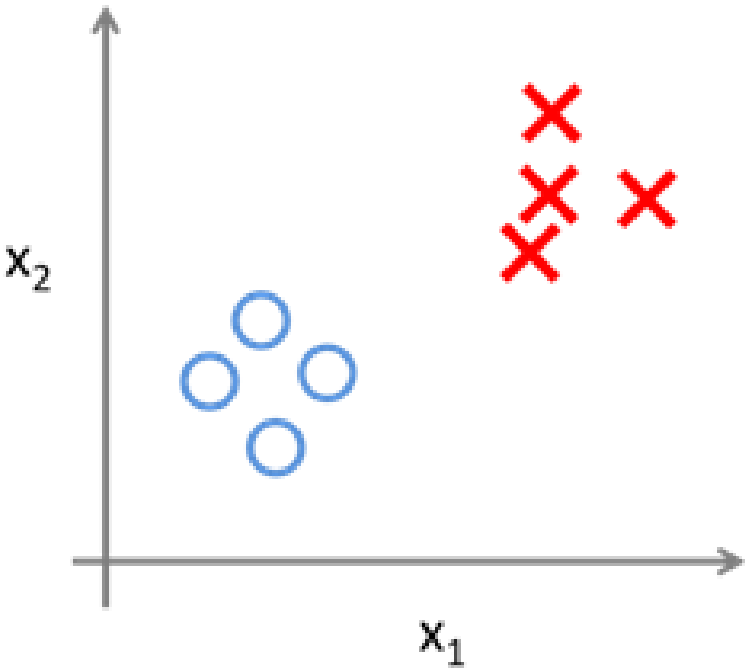
Prediction Accuracy vs. Model Interpretability



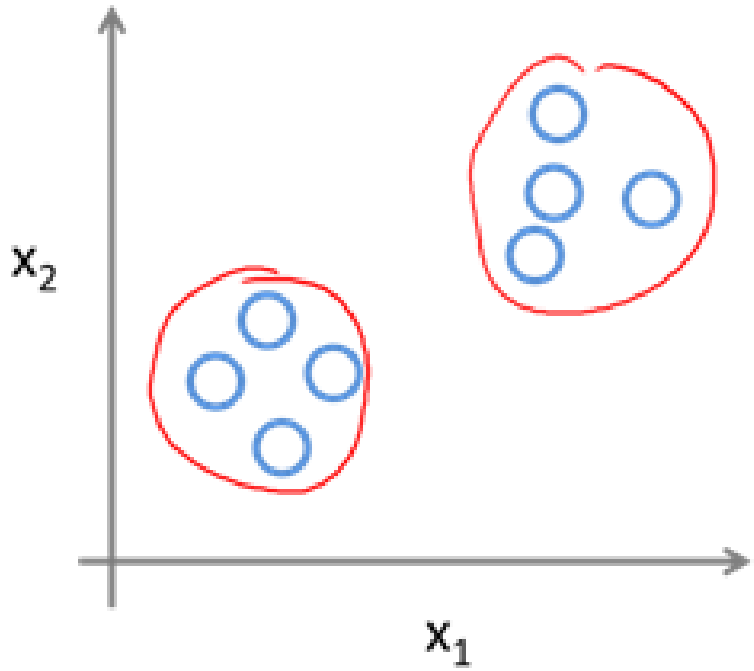
source: ISLR, p.25.

Supervised vs. Unsupervised Learning

- Supervised: has response variable (Y)
  - linear reg., logistic reg., GAM, SVC
- Unsupervised: no supervisor response variable
  - cluster analysis



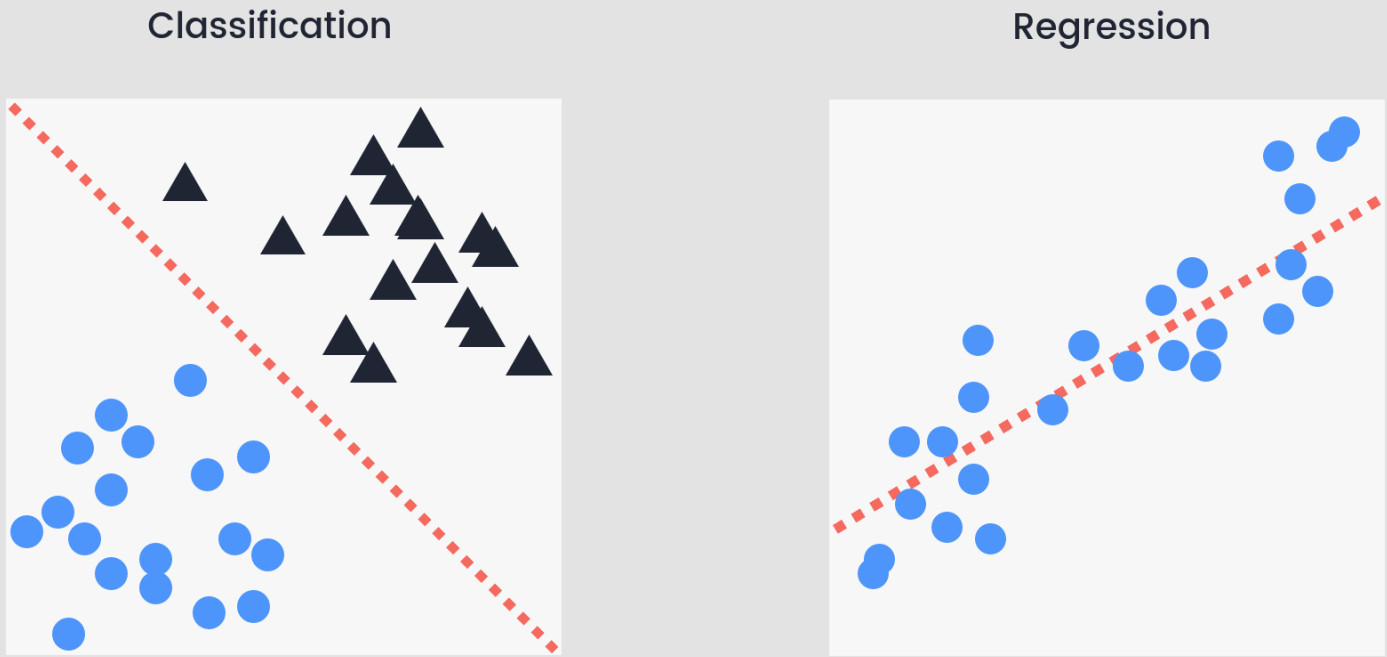
Supervised Learning



Unsupervised Learning

Regression vs. Classification

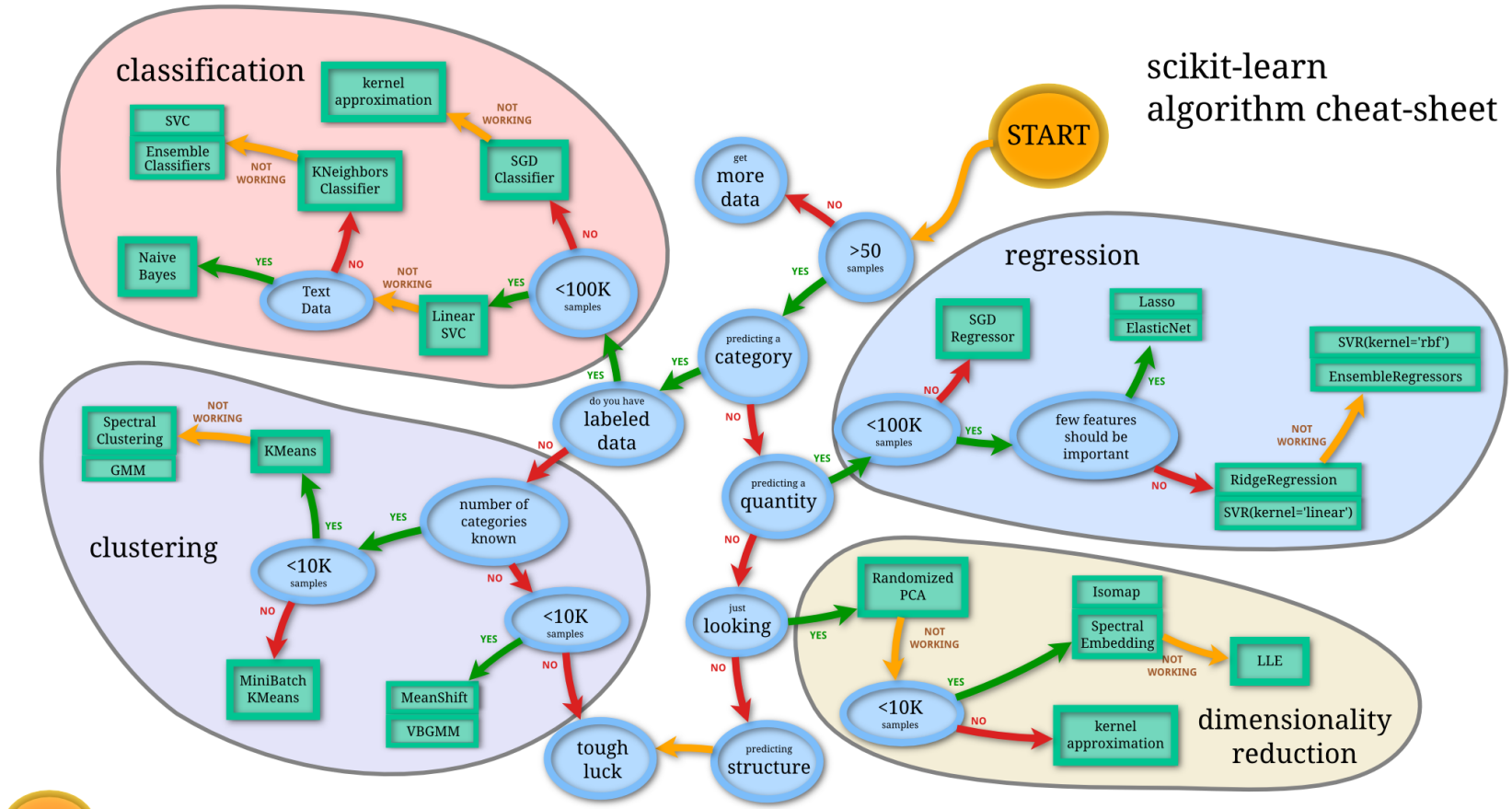
- Regression: quantitative response (e.g. market price prediction)
- Classification: qualitative response (e.g. male/female based on purchase patterns)



Statistical Learning Dimensions Summarized

- Goal: inference vs. prediction
- Model interpretability vs. Prediction Accuracy
- Supervised vs. Unsupervised
- Regression vs. Classification

Other model selection decision points



Back

source

Section 2

Linear Regression

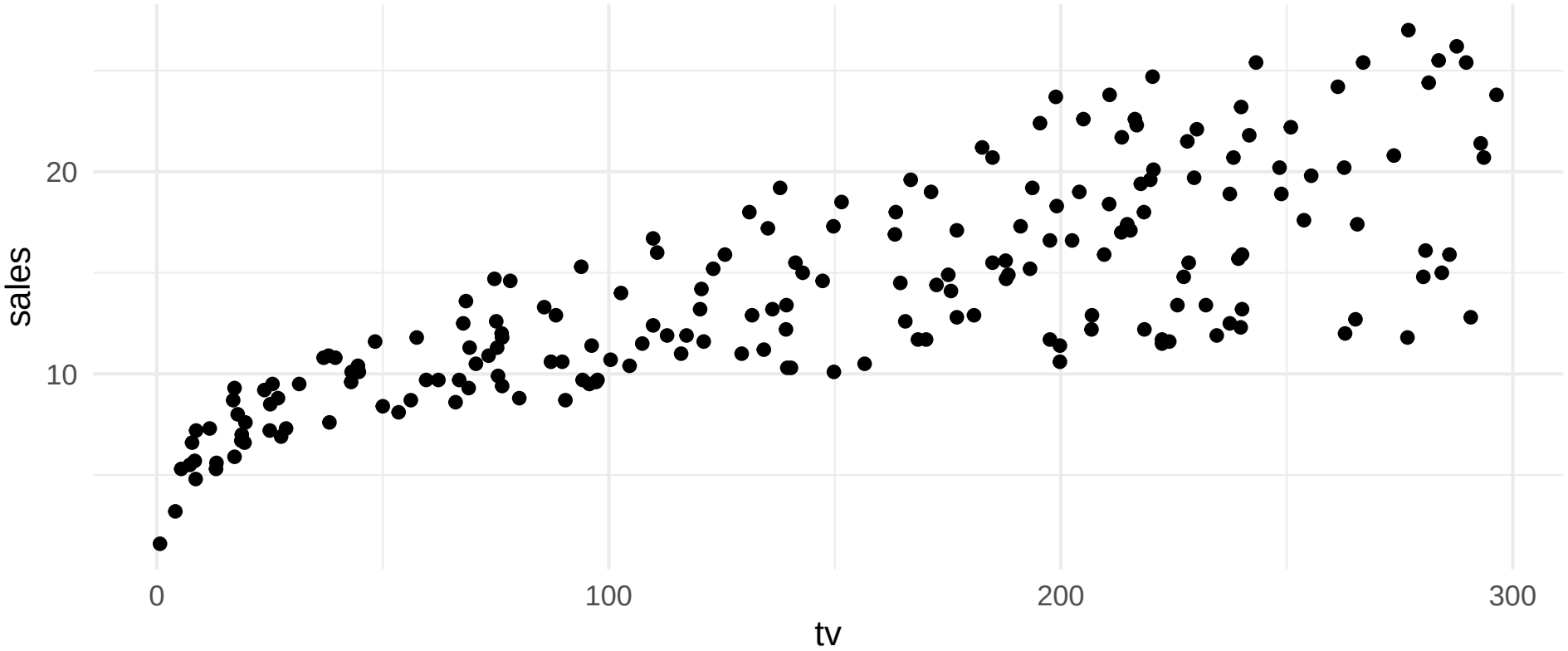
Simple Linear Regression Formula

Assumes an approximate linear relationship between X and Y:

$$Y \approx \beta_0 + \beta_1 X$$

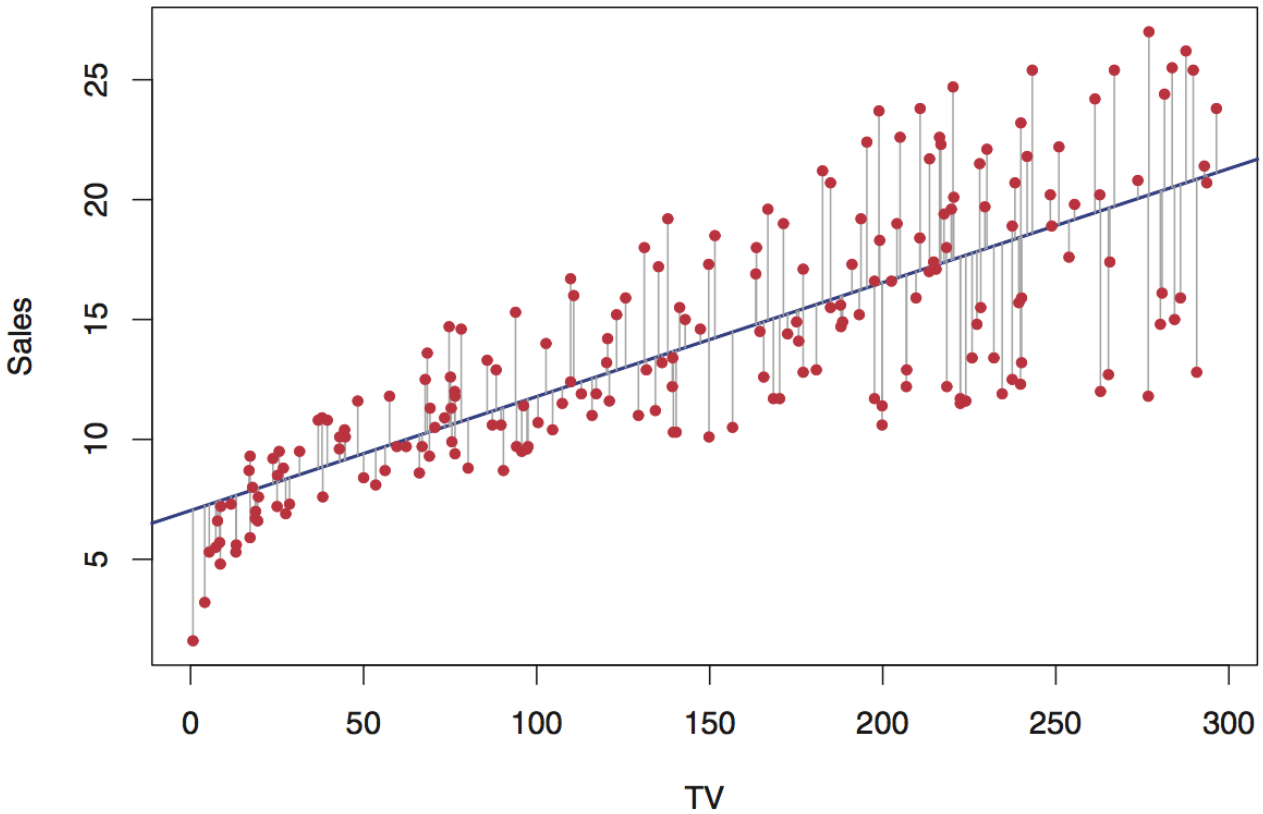
Simple Linear Regression: Advertising Data

```
adv <- fread("./data/advertising_data.csv")
ggplot(adv, aes(tv, sales)) + geom_point() + theme_minimal()
```



Estimating Coefficients

We want to find the coefficients so that the resulting line is as "close" to the observations as possible.



source: ISLR, p.62.

Estimating Coefficients: Least Squares

Minimize the *Residual Sum of Squares* (*RSS*):

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_2 x_2)^2 + ... + (y_n - \hat{\beta}_0 - \hat{\beta}_n x_n)^2$$

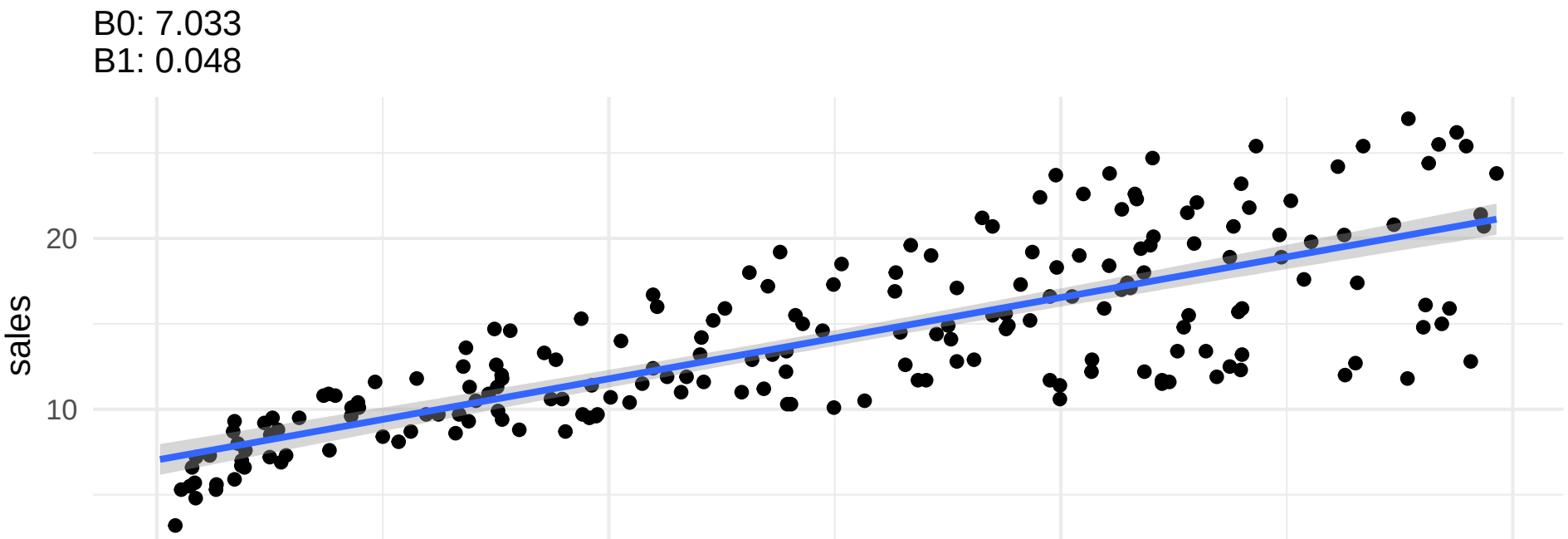
```
slm <- lm(formula = sales ~ tv, data = adv)
slm$coefficients
```

```
## (Intercept)          tv
##  7.03259355  0.04753664
```

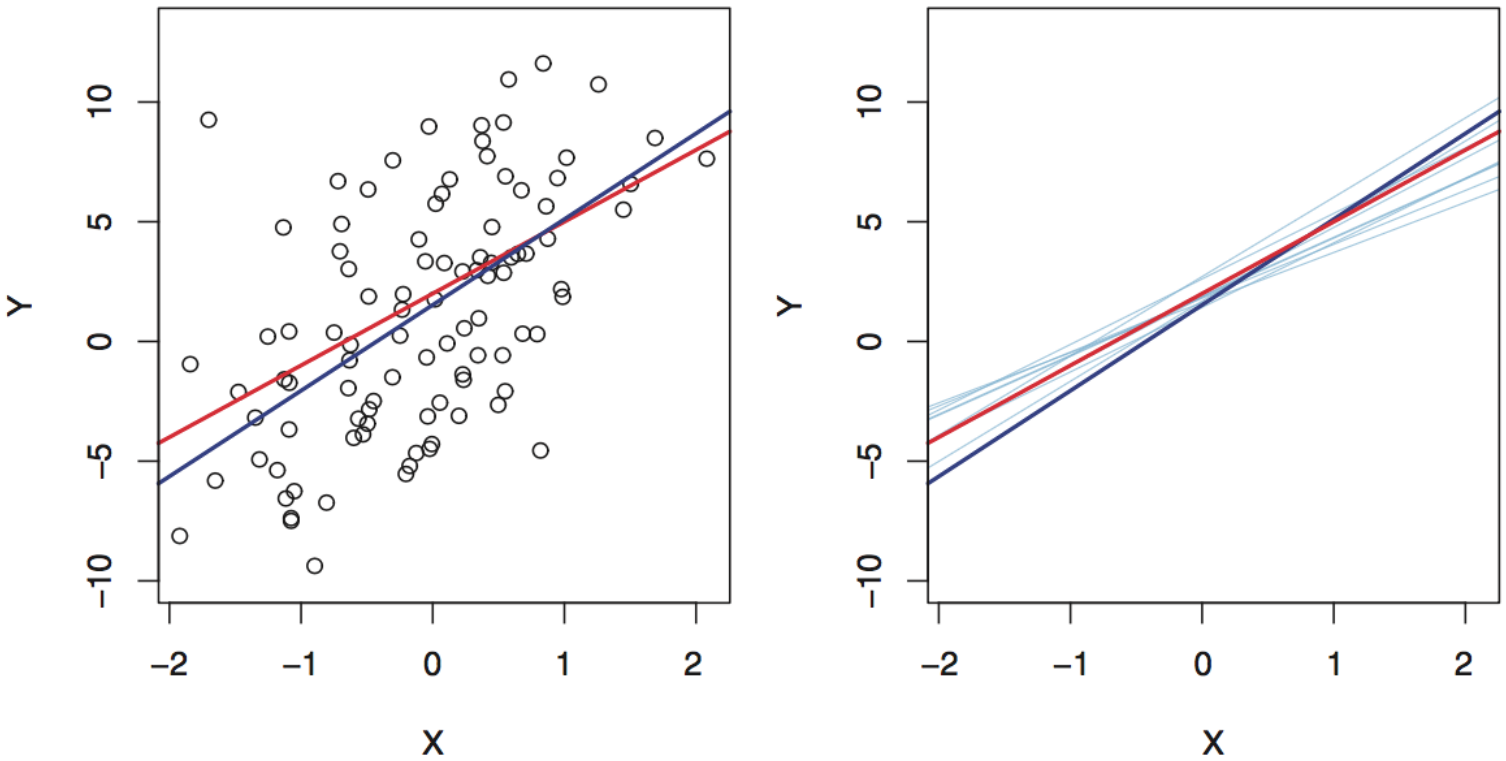
Type `names(slm)` to the console to see `slm`'s other attributes

Estimating Coefficients: Least Squares

```
int <- slm$coefficients["(Intercept)"]
b1  <- slm$coefficients["tv"]
ggplot(adv, aes(tv, sales)) + geom_point() +
geom_smooth(method = "lm") + labs(subtitle = glue(
        "B0: {round(int, digits = 3)}\n",
        "B1: {round(b1, digits=3)}"
    )) + theme_minimal()
```



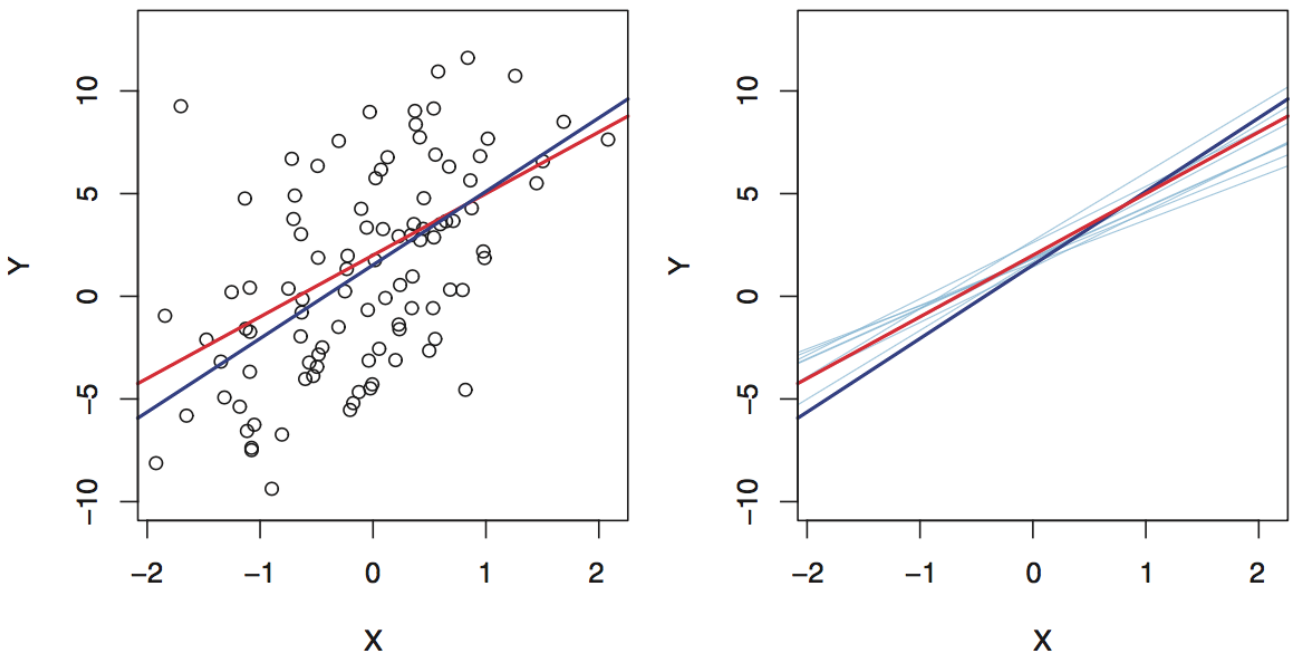
Assessing the Coefficient Estimation Accuracy



We only have one data set, and so what does it mean that two different lines describe the relationship between the predictor and the response?

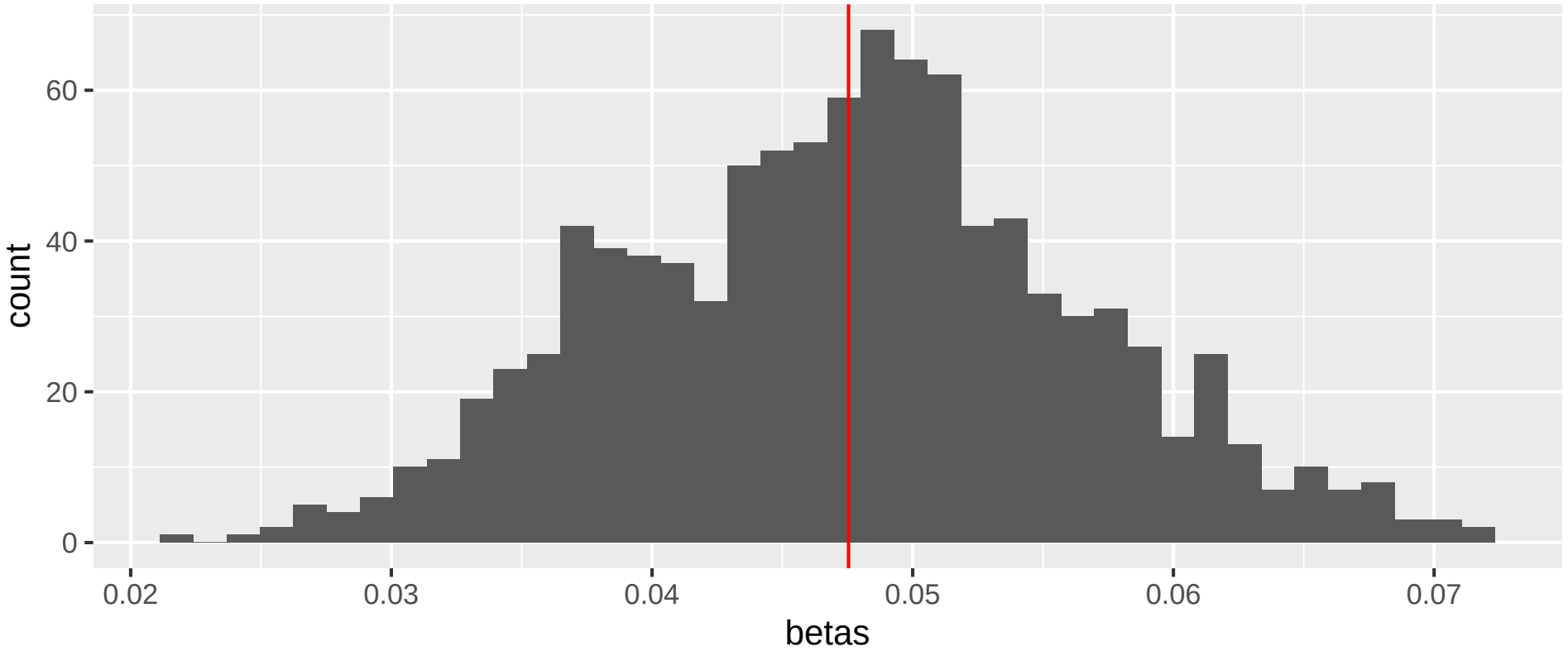
source: ISLR, p.64.

Assessing the Coefficient Estimation Accuracy



- Data Generated: $f(X) = 2 + 3X + \epsilon$
- Population regression line (red): $f(X) = 2 + 3X$
- Least Squares regression line (blue)
- Unbiased estimation

Assessing the Coefficient Estimation Accuracy



Assessing the Coefficient Estimation Accuracy: Standard Error

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right)$$

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

$$\sigma^2 = Var(\epsilon)$$

$$\hat{\sigma} = RSE = \sqrt{RSS/(n-2)}$$

Assessing the Coefficient Estimation Accuracy: Confidence Intervals

- A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter
- For linear regression, the 95% confidence interval for β_1 approximately takes the form

$$\hat{\beta}_1 \pm 2 \cdot SE(\hat{\beta}_1)$$

Assessing the Coefficient Estimation Accuracy: Hypothesis test

$$H_0 : \hat{\beta}_1 = 0$$

$$H_a : \hat{\beta}_1 \neq 0$$

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

- t measures the number of standard deviations that β_1 is away from 0
- the p value tells you how likely it is to observe such t value given $\hat{\beta}_1 = 0$

Assessing the Coefficient Estimation Accuracy: Hypothesis test

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	7.0325	0.4578	15.36	< 0.0001
`TV`	0.0475	0.0027	17.67	< 0.0001

source: ISLR, p.68.

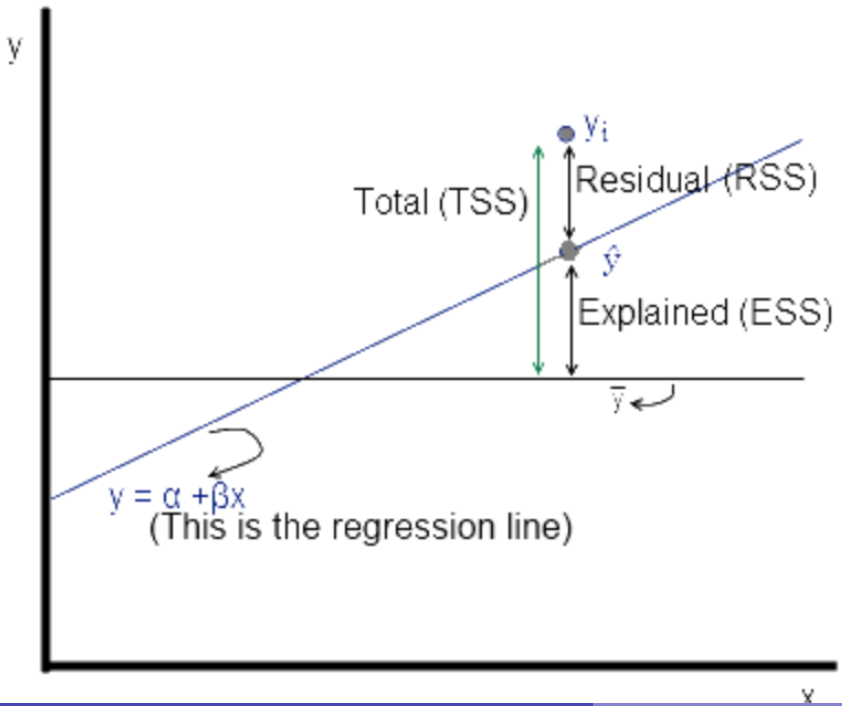
Assessing the Accuracy of the Model: Residual Standard Error (RSE)

- $RSE = \hat{\sigma} = \sqrt{RSS/(n-2)}$
- Roughly speaking, it is the average amount that the response will deviate from the true regression line
- Sales in each market deviate from the true regression line by approximately 3,260 units, on average
- The RSE is considered a measure of the lack of fit of the model to the data

Assessing the Accuracy of the Model: R^2

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

- where $TSS = \sum(y_i - \bar{y})^2$ is the *total sum of squares*
- R^2 measures the proportion of variability in Y that can be explained using X



R syntax: Multiple Linear Regression

Example: median Boston house prices

```
?MASS::Boston
boston <- MASS::Boston
names(boston)
mlm <- lm(medv ~ lstat, data = boston)
mlm <- lm(medv ~ lstat + age, data = boston)
summary(mlm)
mlm <- lm(medv ~ ., data = boston)
mlm <- lm(medv ~ . - indus -age, data = boston)
mlm <- lm(medv ~ . + zn*chas - indus -age, data = boston)
mlm <- lm(medv ~ . + I(lstat^2) - indus -age, data = boston)
```

R syntax: Regression w/ quantitative var.

```
car <- ISLR::Carseats
lm(Sales ~ ShelveLoc, data = car)
```

```
##
## Call:
## lm(formula = Sales ~ ShelveLoc, data = car)
##
## Coefficients:
##       (Intercept)     ShelveLocGood  ShelveLocMedium
##             5.523             4.691             1.784
```

```
contrasts(car$ShelveLoc)
```

```
##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```

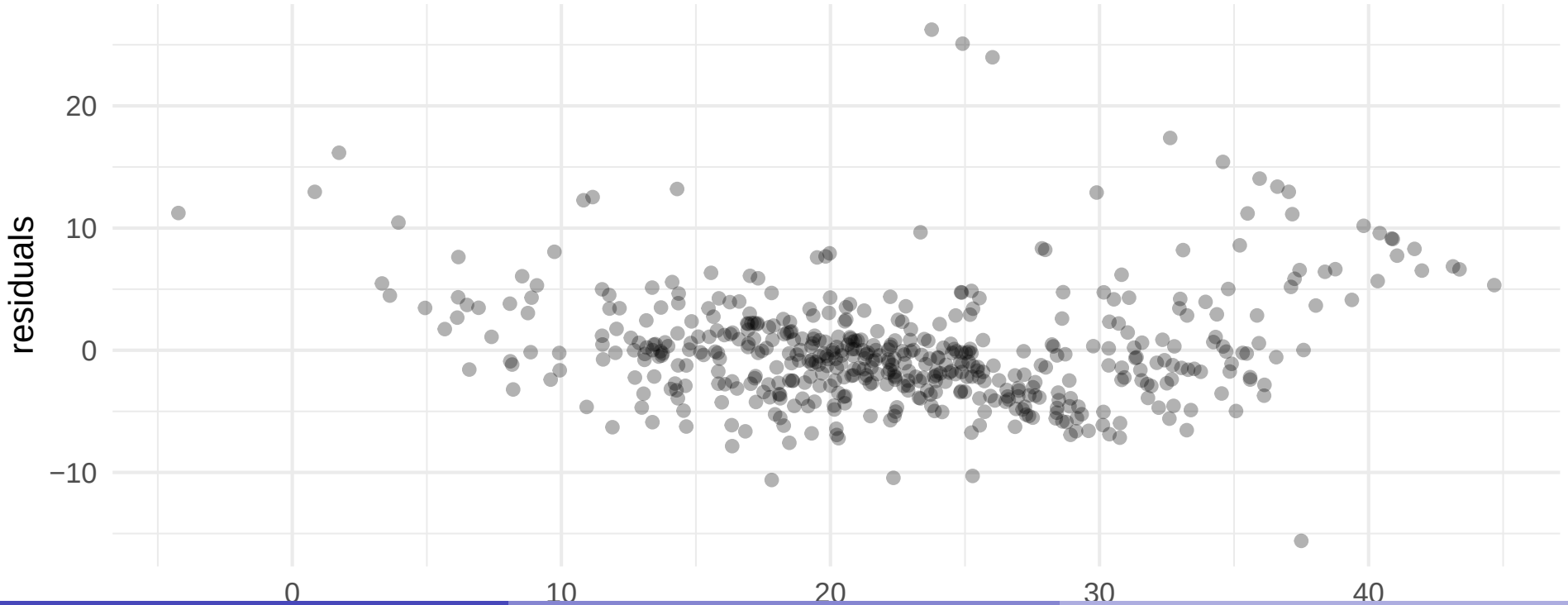
R syntax: Correlation

```
boston <- MASS::Boston
cor(boston)
```

```
##                crim          zn       indus          chas
## crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0
## zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0
## indus    0.40658341 -0.53382819  1.00000000  0.062938027  0
## chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0
## nox      0.42097171 -0.51660371  0.76365145  0.091202807  1
## rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0
## age      0.35273425 -0.56953734  0.64477851  0.086517774  0
## dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0
## rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0
## tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0
## ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0
## black   -0.38506394  0.17552032 -0.35697654  0.048788485 -0
## lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0
```

R syntax: Residuals

```
boston <- MASS::Boston
mlm <- lm(medv ~ . - indus -age, data = boston)
residuals <- data.table(residuals = mlm$residuals, fitted = ml
ggplot(residuals, aes(fitted, residuals)) +
    geom_point(alpha = .3) +
    theme_minimal()
```

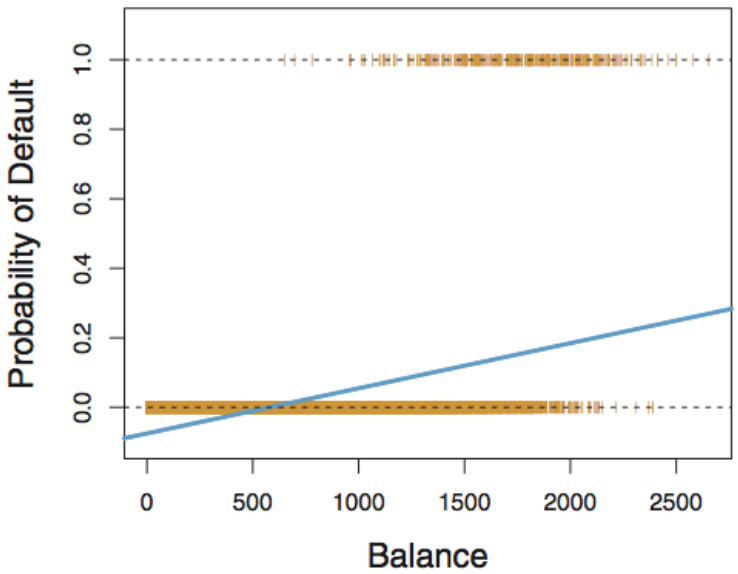


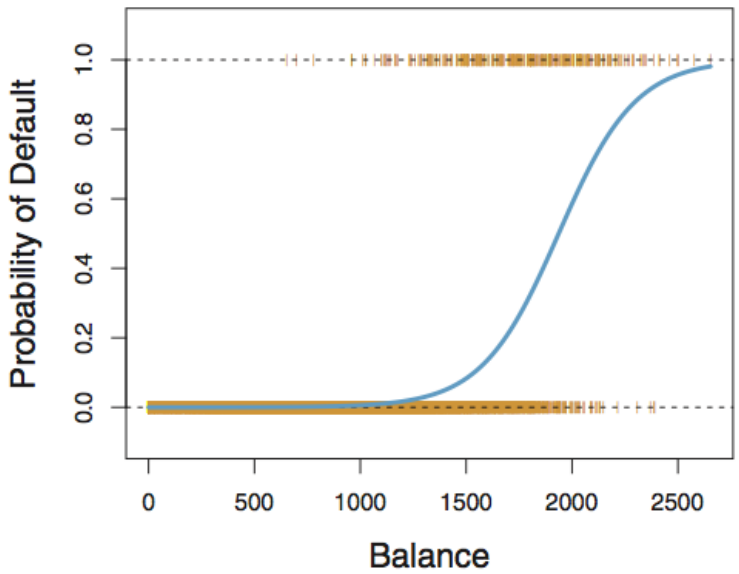
Section 3

Binary Classification

Why not Linear Regression?

- Encoding output with numbers would suggest ordering and distance (e.g. 0 - epileptic seizure, 1 - stroke, 2 - drug overdose)
- Shuffling the encoding would result in different predictions
- Logistic regression models the probability if Y belongs to group, instead of modeling it's value directly. $\Rightarrow p(X) = Pr(Y = 1|X)$





source: ISLR, p.131.

Logistic Regression

Instead of the linear approach we use the *logistic function* so that all our predictions will be between 0 and 1. This is useful if we are talking about probabilities.

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Estimate the coefficients with the *log likelihood* method

R's glm function

```
default <- data.table(ISLR::Default)
glm(default ~ balance, data = default, family = "binomial")
```

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

source: ISLR, p.134.

How to interpret the model: Log-likelihood (Logit)

The problem with the logistic regression is that it's hard to interpret its coefficients. (See the plot above.) For easy interpretation we need a linear formula.

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \qquad (logistic)$$

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X} \qquad (odds)$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0+\beta_1 X \qquad (log-odds)$$

- What does it mean to have odds of 4 to win the horse race?
- How do you calculate your chance of winning the horse race when $X = 8$?
- How do you interpret if $\beta_1 = 2$?

R syntax: Classification

```
default <- data.table(ISLR::Default)
# simple binary model
sbm <- glm(default ~ balance, data = default,
    family = "binomial")
pred_prob <- predict(sbm, type = "response")
# type = "response" -> P(Y = 1 | X) otherwise logit
contrasts(default$default)
pred_response <- rep("No", length(pred_prob))
pred_response[pred_prob > .5] <- "Yes"
```

R syntax: Evaluation

```
table(pred_response, default$default)
```

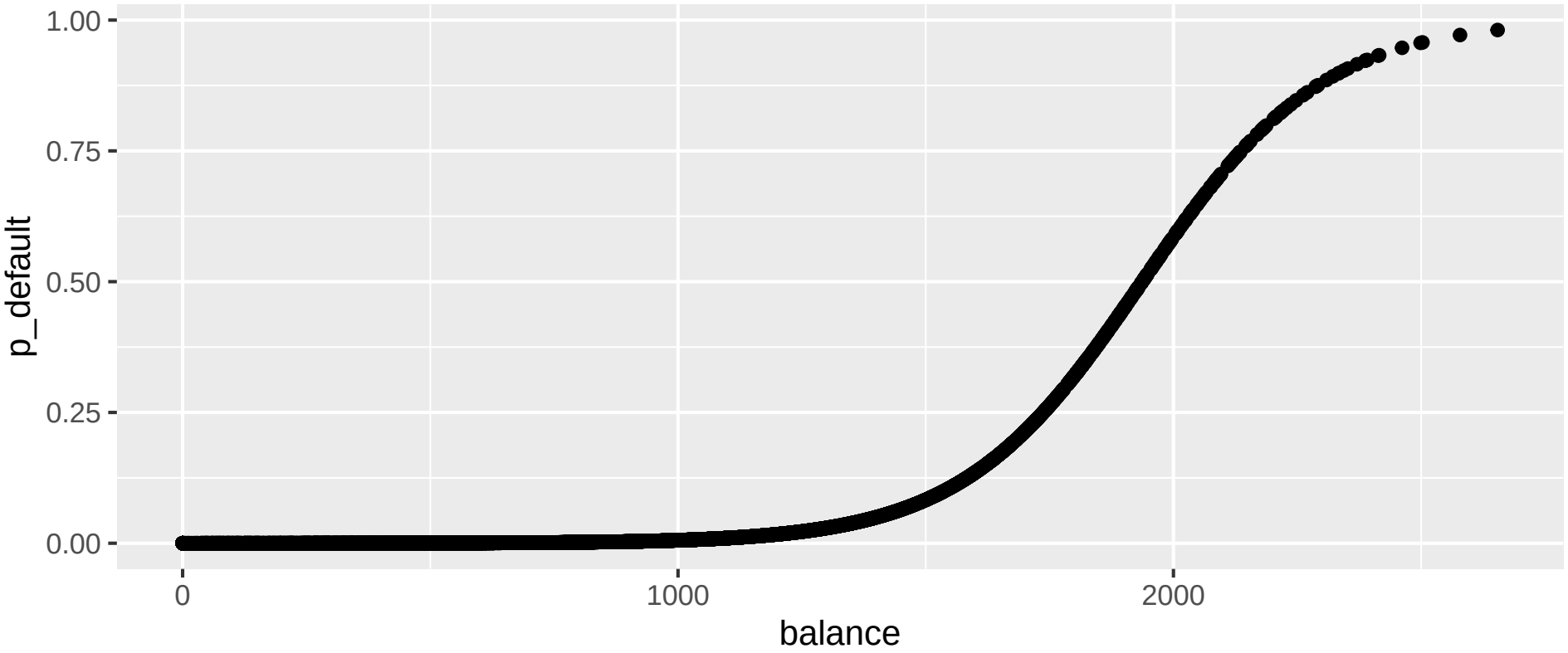
```
##
## pred_response   No  Yes
##           No  9625  233
##           Yes   42  100
```

```
mean(pred_response == default$default)
```

```
## [1] 0.9725
```

R syntax: Classification

```
pred <- data.table(
    p_default = pred_prob, balance = default$balance)
ggplot(pred) +
    geom_point(aes(balance, p_default))
```



Section 4

Hands on Exercises

Now your turn!

1. Either use your project data or find Something on Kaggle
2. Find a good research/business question that involves prediction (either) and write it down
3. Answer your question using what we've learned previously and today
   a. Use at least one graph to see variable interdependence (use `geom_smooth`)
   b. Look at variable correlations
   c. See what do residuals look like. Is it random? Why? Why not?
   d. Run a regression
   e. Add/Remove variables based on the result, rerun the regression
   f. Try adding an interaction
   g. Describe how well the model fits your data. Try to reason why is it week if so?
   h. Try answering your question with your words
   i. finis @ Home!

Thanks!

Thanks for being here!

Great resources (click the links!)

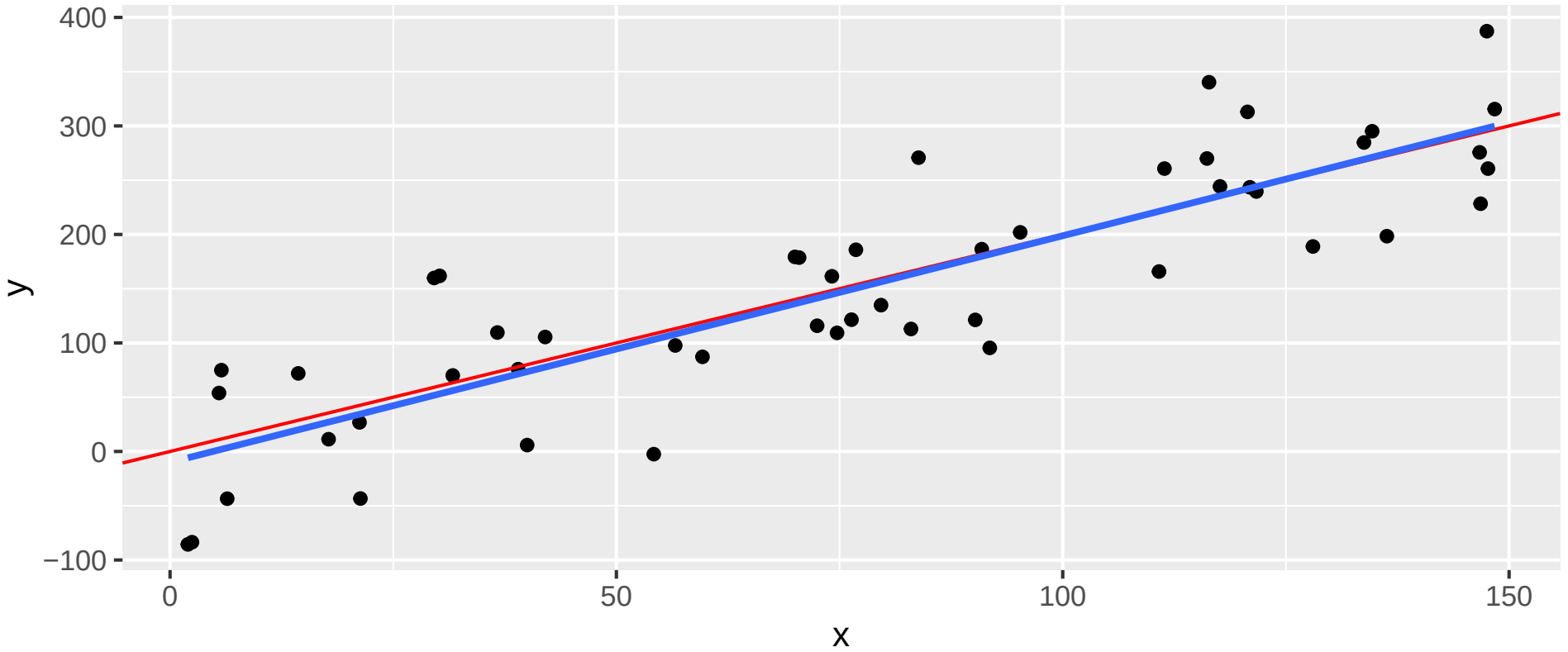
- Free Intro to Stat Learning textbook
- Causality: http://nickchk.com/causalgraphs.html
- Two Stat blogs: 1 and 2
- Online Stat
- MOOCS:
  - Machine Learning @ Coursera
  - Deep Learning @ Coursera
  - Statistics @ Edx

Section 5

Interesting Stuff

Assessing the Coefficient Estimation Accuracy: hands on (1)

```
p <- plotLinData(50, i = 0, b = 2)
p + geom_smooth(method = lm, se = FALSE)
```



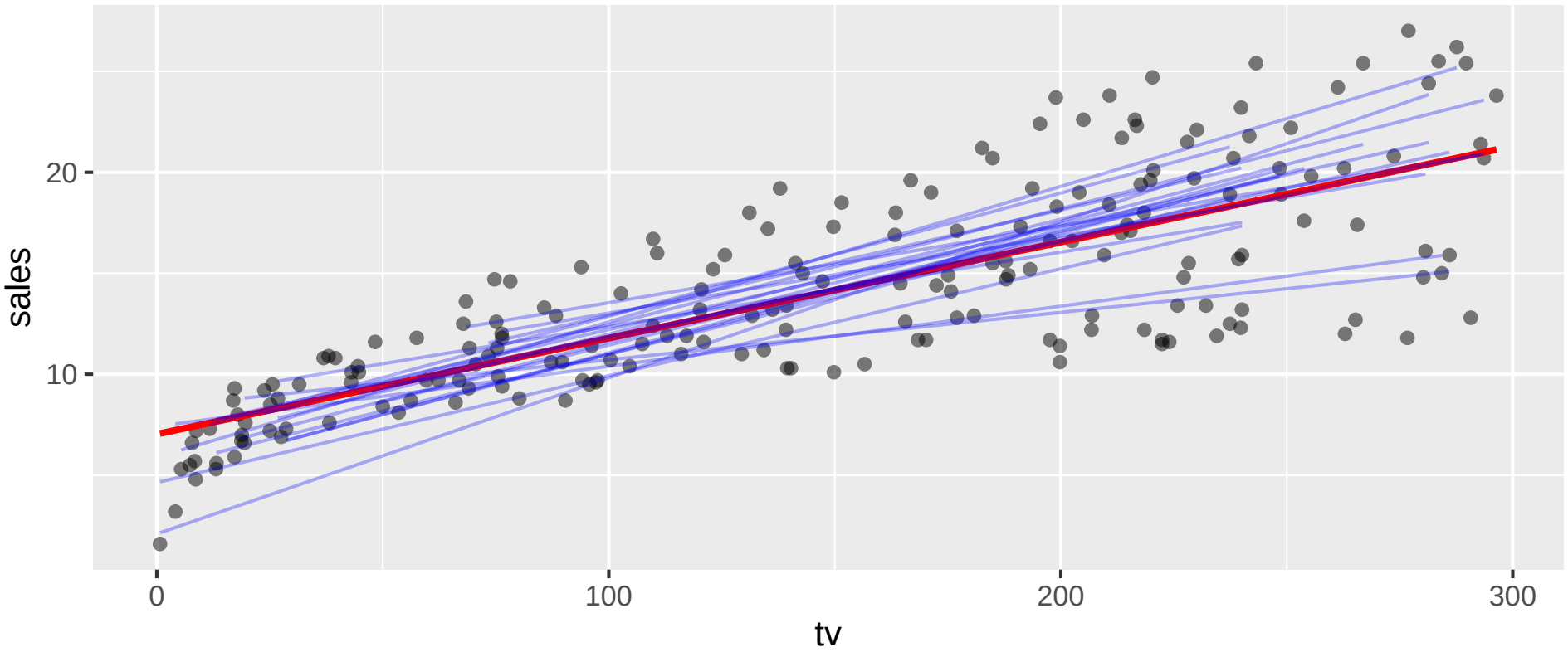
Assessing the Coefficient Estimation Accuracy: hands on (2)

```
ls_lines <- map(1:20, ~{
    adv_s <- adv[sample(.N, 10)]
    stat_smooth(data = adv_s, mapping = aes(tv, sales), metho
        se = FALSE, alpha = .3, geom = 'line', color = "blue"
})
```

Plot on next slide

Least Square estimates on multiple samples

```
ggplot(adv, aes(tv, sales)) +
    geom_point(alpha = .5) +
    geom_smooth(method = lm, color = "red", se = FALSE) +
    ls_lines
```



Assessing the Coefficient Estimation Accuracy: hands on (3)

```
ls_betas <- map(1:1000, ~{
    adv_s <- adv[sample(.N, 20)]
    lm <- lm(sales ~ tv, data = adv_s)
    lm$coefficients["tv"]
}) %>% unlist()

true_beta <- lm(sales ~ tv, data = adv)$coefficients["tv"]
```

Plot on next slide

Output

```
ggplot(data.table(betas = ls_betas), aes(betas)) +
    geom_histogram(bins = 40) +
    geom_vline(xintercept = true_beta, color = "red")
```