STAT 113
Analytic Inference for Regression

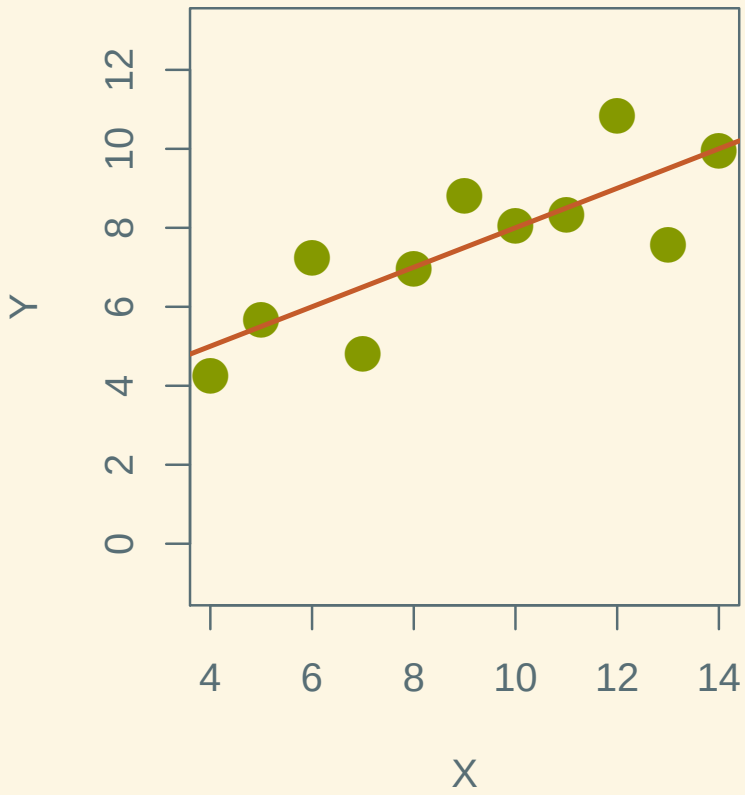
Colin Reimer Dawson

Oberlin College

November 17, 2017

Outline

What's a Good Prediction?



- Pretty much the simplest model we can have is a straight line.
- Two things determine what line we have:
  - The intercept
  - The slope

Review: Intercept Slope Form

- The intercept and slope are the **parameters** of our regression model.
- The general equation for a line is:

$$f(x) = a + bx$$

- In statistics notation, we write $\hat{y}$ ("y hat") to represent a *predicted* (or fitted) value.
- Given a value x_i, we predict using:

$$\hat{y}_i = a + bx_i$$

The Simple Linear Model

Prediction Function

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

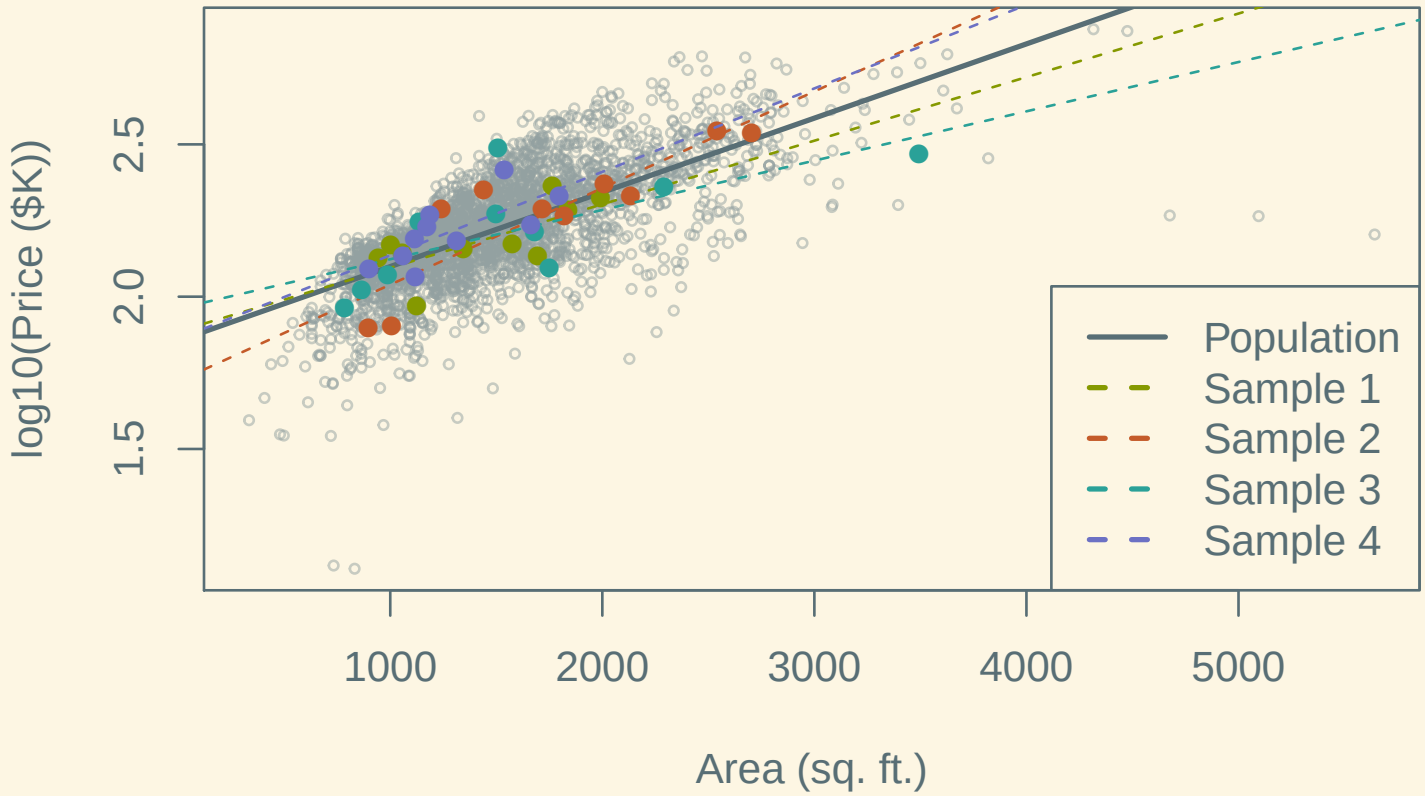
The Population Model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where ε is a *residual*, specific to each case.

Sample vs Population "Best-Fit" Line

- For a sample: choose intercept and slope to minimize sum of squared errors.
- But this does not yield the "correct" (or even "best") model for the population, due to sampling error.

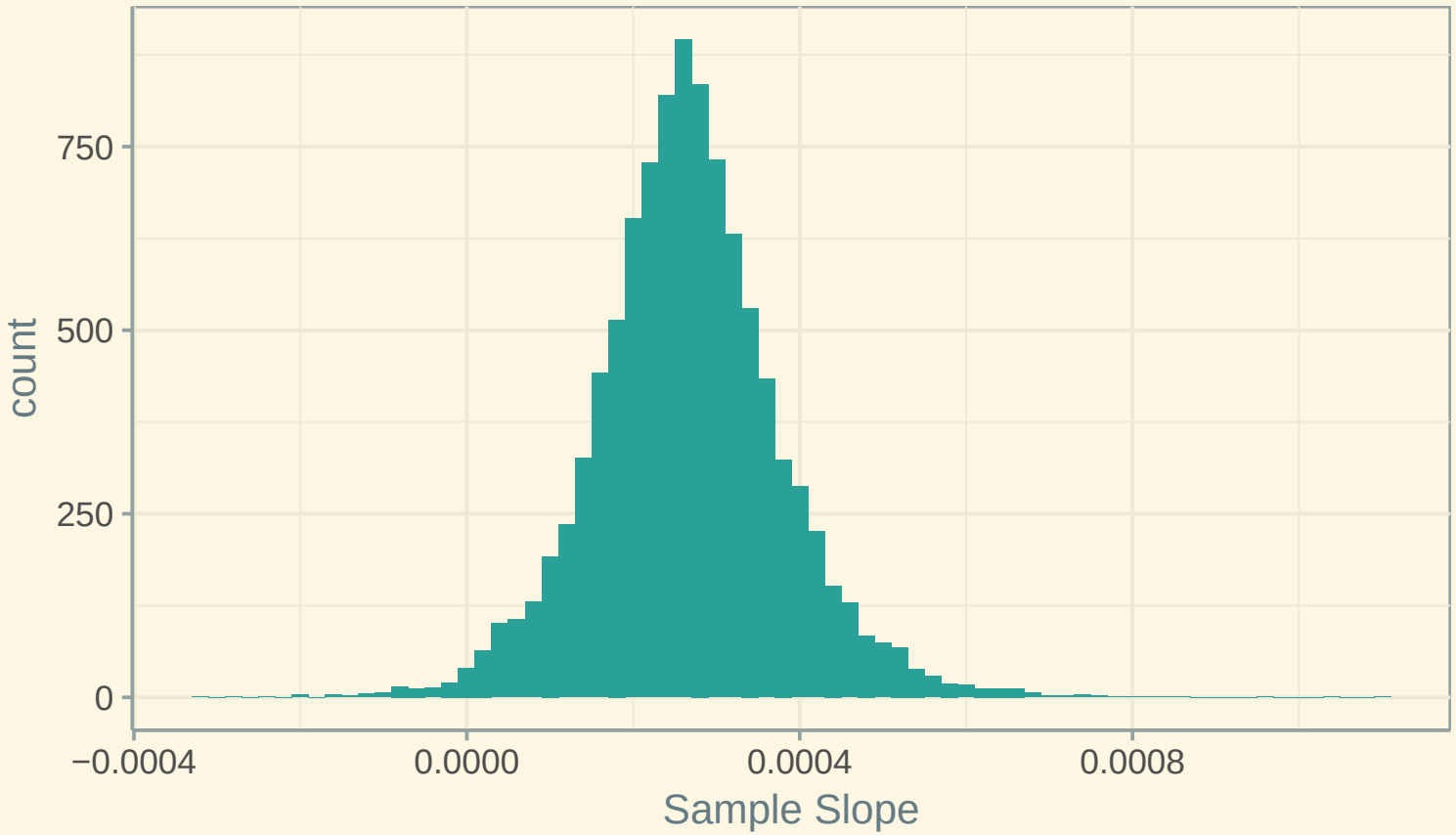


Reminder: Sampling Distributions

Sampling Distribution

The **sampling distribution** of a sample statistic (e.g., $\hat{\beta}_1$) for β_1, or $\bar{Y}$ for μ_Y) is the distribution that statistic has across all possible samples from the population.

Predicting Home Prices in Ames, Iowa



Two Methods for Estimating Sampling Distribution

1. Bootstrap distribution
2. t-distribution: assumes Normal residuals (along with other regression conditions)

Analytic Tests and Intervals for the Slope

The standardized sampling distributions are well-modeled using a t-distribution, provided:

1. The linear model is appropriate
2. The residuals have constant variance across all x
3. The residuals are Normally distributed, or the sample size is large enough

Refining the Linear Model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where the ε are distributed as $\mathcal{N}(0, \sigma_\varepsilon)$ for a σ_ε that does not depend on x.

Ways The Conditions Can Be Violated

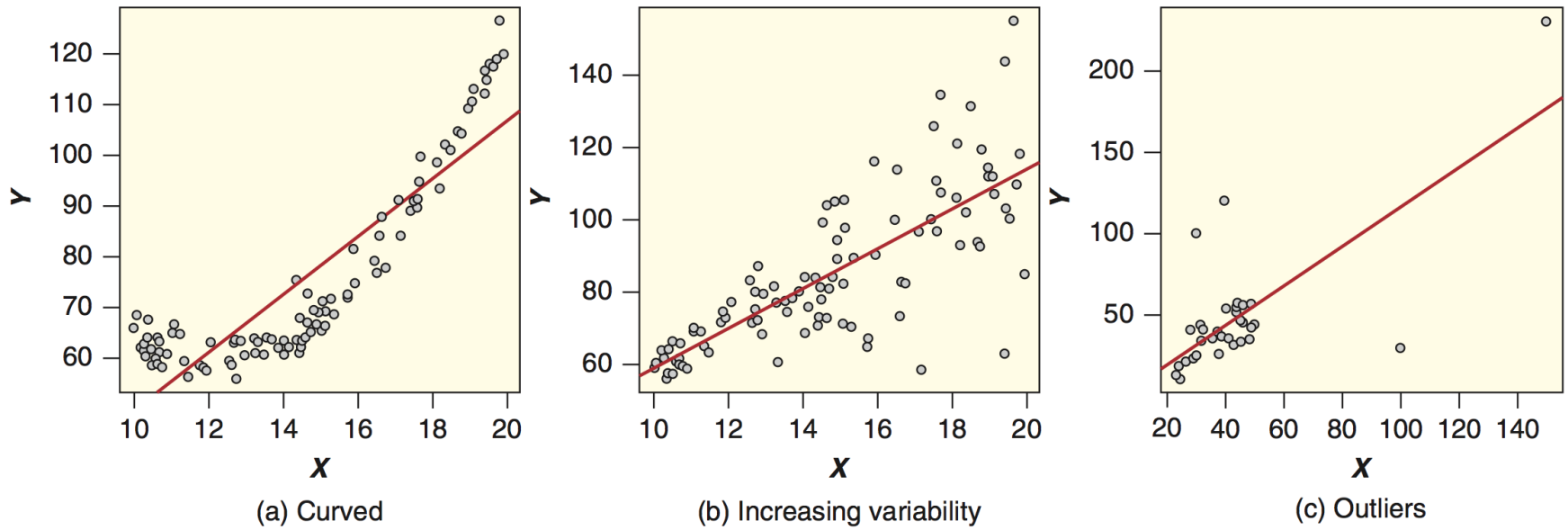
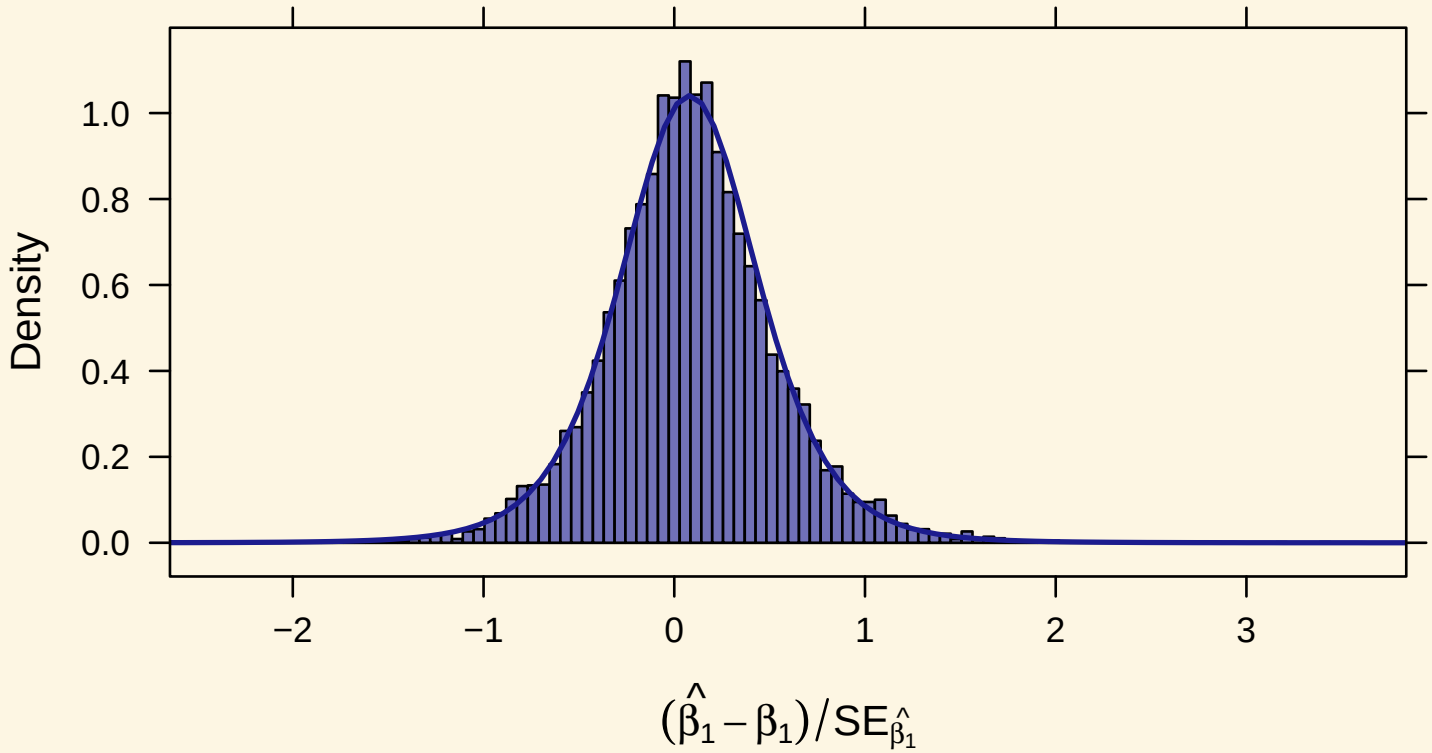


Figure 9.5 *Scatterplots for least squares fits with problems*

Figure: From the textbook (Fig. 9.5)

Normal Residuals

If $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$, then $\dfrac{\hat{\beta}_i - \beta_i}{\widehat{SE}_{\hat{\beta}_i}} \sim t_{n-2}$



t-based Confidence Interval and Test

$$CI : \hat{\beta}_1 \pm t^*_{n-2} \cdot \widehat{SE}_{\beta_1}$$

where t^*_{n-2} is the "standardized endpoint" or "critical value": that is, the $1 - \alpha/2$ quantile of the t_{n-2} distribution.

The standardized test statistic is

$$t_{obs} = \frac{\hat{\beta}_1 - 0}{\widehat{SE}_{\beta_1}}$$

and the P-value is the area beyond t_{obs} in a t_{n-2} distribution

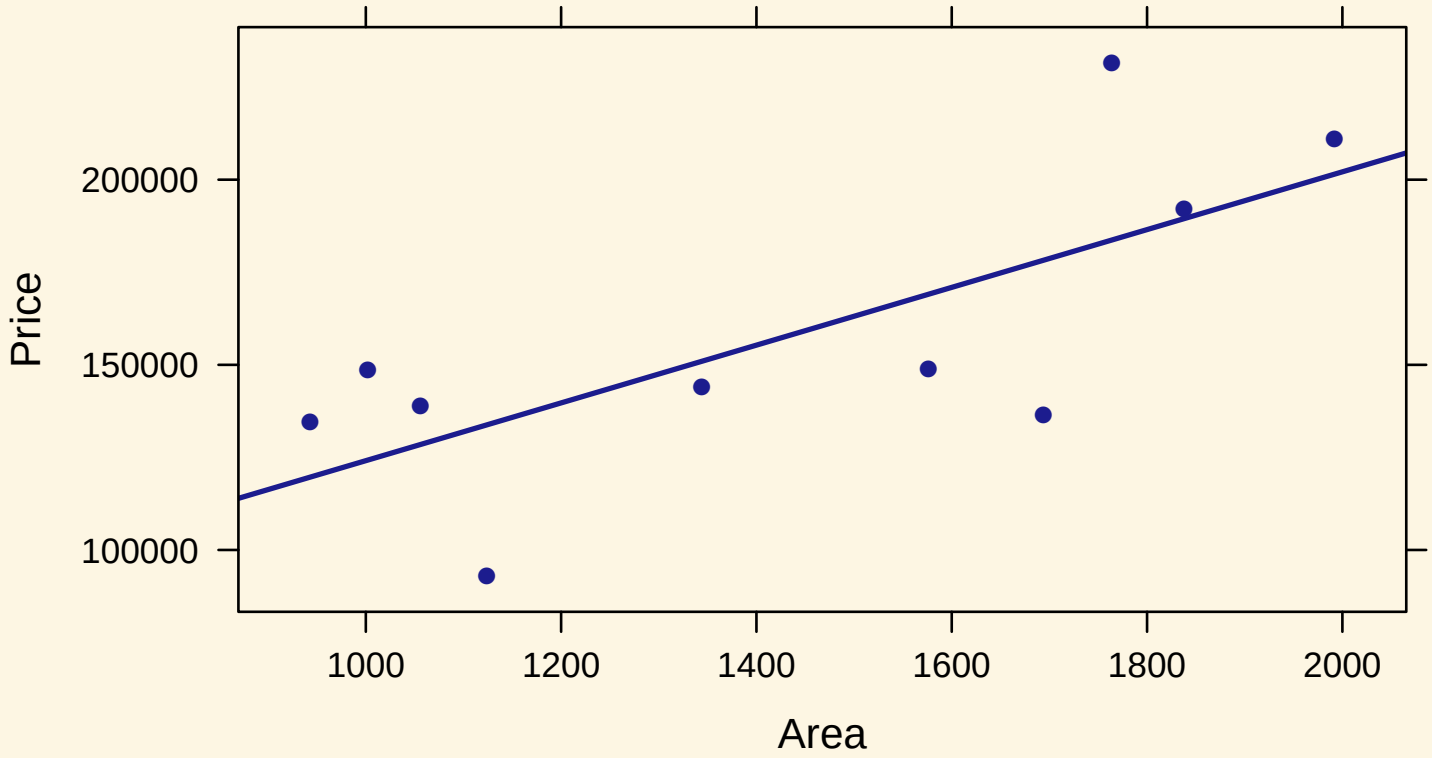
Analytic Tests and Intervals for the Slope

- The SE expression is not given in the book, but when the conditions are met, it is:

$$SE = \sqrt{\frac{s_\varepsilon^2 / s_x^2}{n-2}}$$

  where s_ε^2 is the variance of the *residuals*, and s_x is the variance of the X variable.
- We will not need to use this by hand

```
xyplot(Price ~ Area, data = AmesData, type = c("p","r"))
```



```
sample.model <- lm(Price ~ Area, data = AmesData)
summary(sample.model)$coefficients %>% round(digits = 3)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 46119.511  38139.131   1.209    0.261
Area           77.998     25.778   3.026    0.016
```

```
MoE.95 <- qt(0.975, df = 10 - 2) * 25.778
CI.95 <- c(77.998- MoE.95, 77.998 + MoE.95)
CI.95
```

```
[1]  18.55383 137.44217
```

```
confint(sample.model, level = 0.95)
```

```
                  2.5 %      97.5 %
(Intercept) -41829.48299 134068.5051
Area           18.55332    137.4426
```

Correlation Test and Interval

Can also estimate dist. for correlation r using t_{n-2}, where

$$\widehat{SE}_r = \sqrt{\frac{1-r^2}{n-2}} \tag{1}$$

$$CI : r \pm t^*_{n-2} \cdot \widehat{SE}_r \tag{2}$$

$$t_{obs} = \frac{r-0}{\widehat{SE}_r} \tag{3}$$

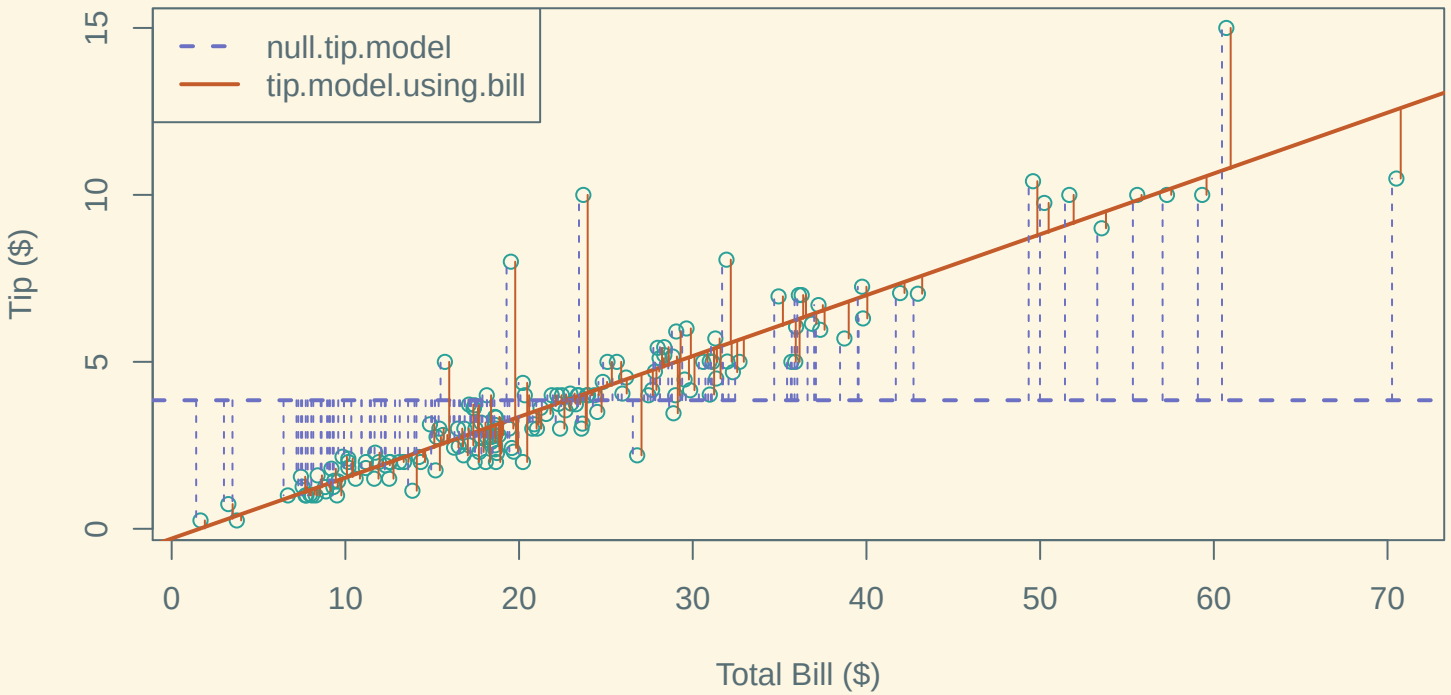
Proportion of Variability Explained

The Coefficient of Determination (R^2)

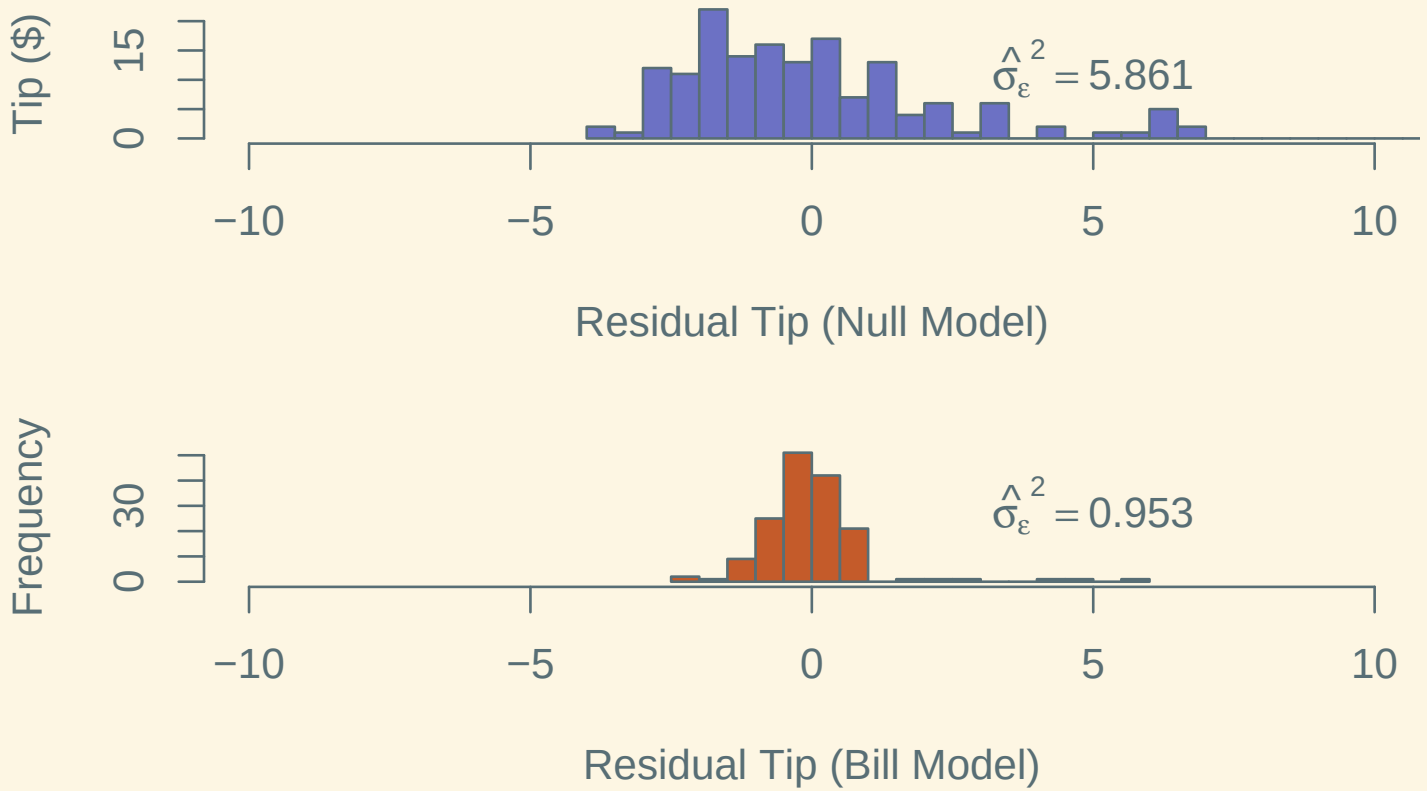
The **coefficient of determination**, or R^2 value, associated with a linear model, is the percent reduction in prediction uncertainty achieved by using the model.

Example: Restaurant Tips

```
library("Lock5Data"); library("mosaic")
data("RestaurantTips")
null.tip.model <- lm(Tip ~ 1, data = RestaurantTips)
tip.model.using.bill <- lm(Tip ~ Bill, data = RestaurantTips)
```



Example: Restaurant Tips



Regression Summary

```
summary(tip.model.using.bill)



    Call:
    lm(formula = Tip ~ Bill, data = RestaurantTips)

    Residuals:
        Min      1Q  Median      3Q     Max
    -2.3911 -0.4891 -0.1108  0.2839  5.9738

    Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
    (Intercept) -0.292267   0.166160  -1.759   0.0806 .
    Bill         0.182215   0.006451  28.247   <2e-16 ***
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

    Residual standard error: 0.9795 on 155 degrees of freedom
    Multiple R-squared:  0.8373,Adjusted R-squared:  0.8363
    F-statistic: 797.9 on 1 and 155 DF,  p-value: < 2.2e-16
```

Intervals at a particular X

- A confidence interval for the slope is useful, but if our goal is a predictive model, we want to be able to make statements about Y values at particular X values.
- I should be able to estimate
  1. What the mean Y value is at that X *in the population*
  2. Where the particular Y is likely to be for *this one new observation*
- Note: These are different things, in the same way that a 95% confidence interval does *not* tell us where 95% of the *individual cases* are.

Confidence and Prediction Intervals for a Linear Model

(Population) linear model:

$$\begin{aligned} Y &= \beta_0 + \beta_1 X + \varepsilon \\ &= f(X) + \varepsilon \end{aligned}$$

1. A **confidence interval** (for a particular X) is an estimate (with a margin of error) of $f(X)$.
2. A **prediction interval** (for a particular X) is an estimate about Y

Confidence vs. Prediction Intervals

Which is wider? The prediction interval is wider, b/c it has uncertainty about ε *plus* the uncertainty about $f(X)$

A Subtlety Re: Prediction Intervals

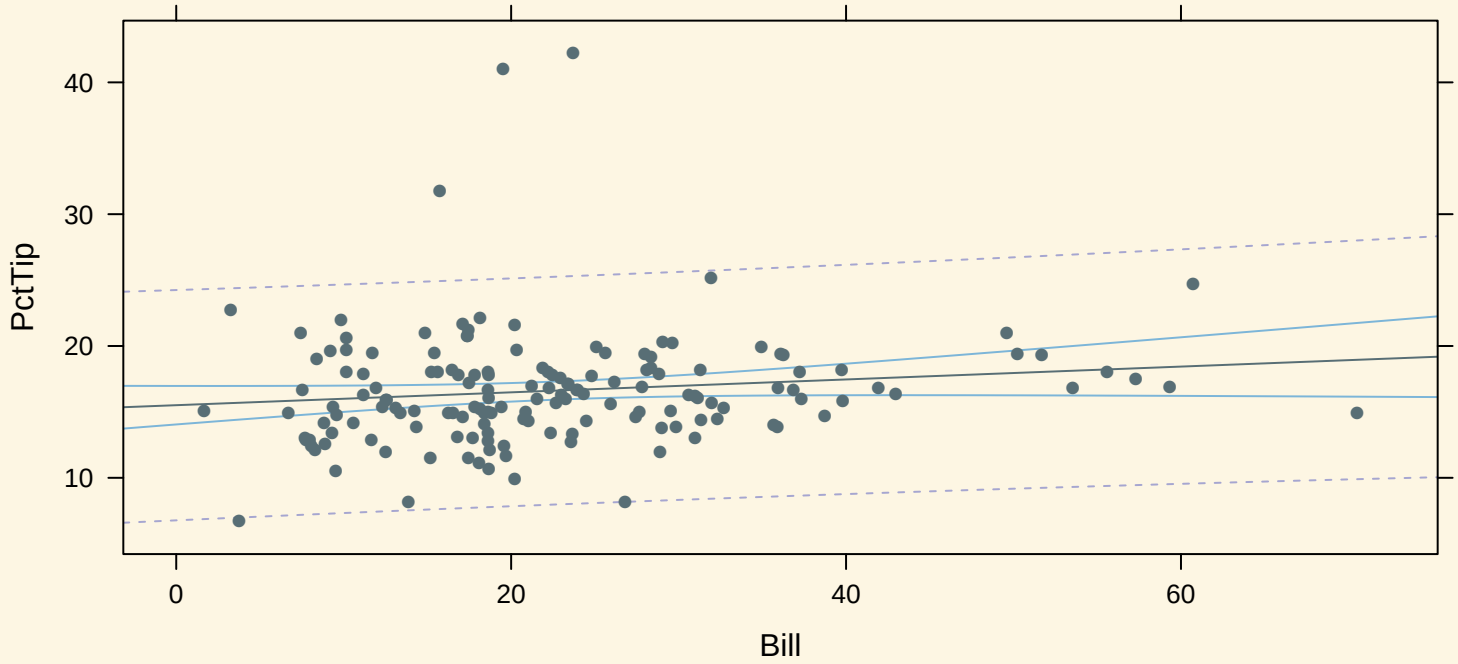
Interpreting Prediction Intervals

A coverage level of 95% for a prediction interval does *not* mean that, having fit a model from a *particular* sample, we will make successful predictions 95% of the time going forward. The worse our line, the lower the %.

What we can say is that the *average* success rate across *all possible samples* is 95%

Confidence and Prediction Bands

Intervals for all x in the range are called "confidence / prediction bands".



Why the hourglass shape? More leverage at extreme X^*: bigger change in line from one sample to the next

Calculating Confidence and Prediction Intervals

Both types of intervals are of the form

$$(1-\alpha) \text{ interval } = \text{Point Estimate} \pm t_{n-2}^{*(1-\alpha/2)} \cdot SE$$

Confidence Interval:

$$\hat{f}(X^*) \pm t_{n-2}^{*(1-\alpha/2)} \cdot \sqrt{\hat{\sigma}^2_{\hat{f}(X^*)}}$$

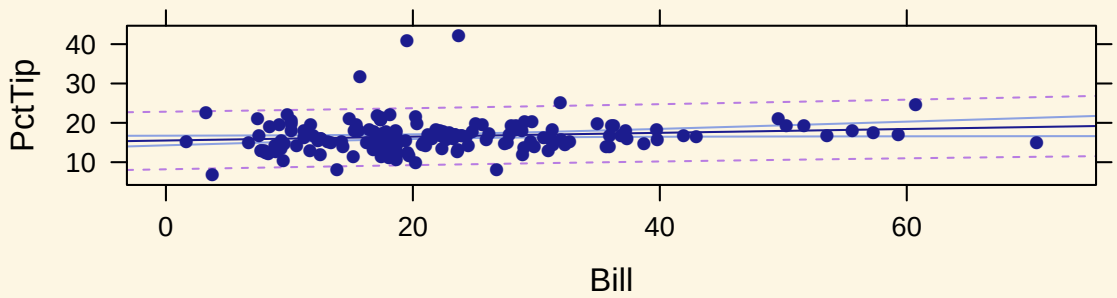
where $\hat{\sigma}^2_{\hat{f}(X^*)} = \hat{\sigma}^2_\varepsilon h(X^*)$ and $h(X^*) = \frac{1}{n} + \frac{(x*-\bar{x})^2}{\sum(x_i-\bar{x})^2}$ is the leverage at X^*.

Prediction Interval:

$$\hat{Y}* \pm t_{n-2}^{*(1-\alpha/2)} \cdot \sqrt{\hat{\sigma}^2_{\hat{f}(X^*)} + \hat{\sigma}^2_\varepsilon}$$

R code for a confidence/prediction bands plot:

```
library("mosaic"); library("Lock5Data")
data("RestaurantTips")
xyplot(PctTip ~ Bill, data = RestaurantTips,
       panel = panel.lmbands,  # Note, no quotes
       level = 0.90,           # The confidence level
       ## OPTIONAL: band.lty= what kind of lines to use
       ##    format: c(conf.linetype, pred.linetype), where
       ##    1 = solid, 2 = dashed, 3 = dotted
       band.lty = c(1,2),
       ## OPTIONAL: band.col: what color lines to use
       ##    format: c(conf.color, pred.color)
       band.col = c("royalblue", "blueviolet")
       )
```



We can get intervals for specific X values as follows:

```
tip.model.using.bill <- lm(PctTip ~ Bill, data = RestaurantTips)
## Creates a new function with the given name
f.hat <- makeFun(tip.model.using.bill)
## Use it like a regular function
##   First arg name: name of predictor variable
##      (= the desired x value to get the interval for)
##   interval="confidence" or interval="prediction"
##      controls which interval type to return
##      (or leave this out to just get the pt estimate)
##   level=confidence.level controls the confidence level
f.hat(Bill = 40, interval = "confidence", level = 0.90)

            fit      lwr      upr
    1 17.46215 16.45974 18.46455

f.hat(Bill = 40, interval = "prediction", level = 0.90)

            fit     lwr      upr
    1 17.46215 10.1786 24.74569
```