Business Statistics: 41000

Week 5: Linear Regression

Nick Polson
The University of Chicago Booth School of Business
http://faculty.chicagobooth.edu/nicholas.polson/teaching/41000/

Suggested Reading
OpenIntro Statistics, Chapters 4,5&6
Regression: Introduction

Regression analysis is the most widely used statistical tool for understanding relationships among variables

- Regression provides a conceptually approach for investigating relationships between one or more factors and an outcome of interest
- The relationship is expressed in the form of an equation or a model connecting the response or dependent variable and one or more explanatory or predictor variable
Henrietta Leavitt’s 1912 data on 25 pulsating stars. Pattern of period of oscillation with brightness allowed astronomers to measure cosmic distances over previously unimaginable scales.
In AI, the criterion for evaluating prediction rules is simple: How big are the errors the rule makes, on average?

Leavitt used “the principle of least squares" to fit a prediction rule to her data.
Prediction

Straight prediction questions:

- For how much will my house sell?
- Will the Chicago Cubs win the World Series?
- Will this person like that movie? (Netflix prize)

Explanation and understanding:

- What is the impact of an MBA on income?
- How does the returns on Google relate to the market?
Models, Parameters and Estimates

We’ll use probability to talk about uncertainty ... and build models

- Define the random variable, $Y$, of interest
- Construct a regression model from historical data on characteristics, $X$. This entails estimating parameters using their sample counterparts
- We are now ready to generate predictions, make decisions, evaluate risk, etc...
Predicting House Prices

Problem: Predict market price based on observed characteristics (Zillow)

Solution:

▶ Look at property sales data where we know the price and some observed characteristics

▶ Build a decision model that predicts price as a function of the observed characteristics.
Zillow: Zestimate

R and Zestimate

R and AWS for analytics are helping Zillow real estate data.

Zillow and Big Data

Database behind the Zestimate is 20TB in size.

Zillow employs various decision tree, random forest, and regression algorithms

By averaging models, margin of error in pricing improved from 14% to 5%

Zillow Prize

Predicting whether you have waterfront property ...
Predicting House Prices

What characteristics do we use?

There are many factors or variables that affect the price of a house (location, location, location, ...)

Some basics ones include

- size
- ZIP code
- location
- parking, ...

Let’s run a simple linear regression on size
Predicting House Prices

The value that we seek to predict is called the dependent (or output) variable, and we denote this by \( Y = \text{price of house (e.g. thousand of dollars)} \).

The variable that we use to construct our prediction is the explanatory (or input) variable, and this is labeled \( X = \text{size of house (e.g. thousand of square feet)} \).
Predicting House Prices

What's does the data look like?
Predicting House Prices

Simple Linear Regression (SLR) model

\[ \text{price} = \beta_0 + \beta_1 \text{ sqft} + \epsilon \quad \text{where} \quad \epsilon \sim N(0, \sigma^2) \]

where we add a random error term, \( \epsilon \).

The error term models the fact that not all prices will lie on our regression line.

We find that \( \beta_1 = 0.11 \)

**Implication:** every 1 sqft increase ups price by $110K
Predicting House Prices

We can now predict the price of a house when we only know that size: take the value off the regression line.

For example, given a house size of \( X = 2200 \)

\[ \hat{Y} = 13.44 + 0.11(2200) = 262 \]

The intercept \( \beta_0 = 13.44 \) measures land value. In R: `predict.lm( ... )`
Now plot and run your regression ...

```r
house = read.csv("data/SaratogaHouses.csv")
house$price = house$price/1000
plot(price~livingArea, data=house)
model=lm(price~livingArea, data=house)
coef(model)
abline(model, col="red", lwd=3)
coef(model)
```

The key command is `lm( ... )` which stands for linear model.

R: will calculate everything for you!!
Simple Linear Regression (SLR)

The underlying assumptions about the linear regression model are:

1. For each value $X$, the $Y$ values are *normally distributed*
2. The means of $Y$ all lie on the regression line
3. The *standard deviations* of these normal distributions are *equal*
4. The $Y$ values are statistically *independent*. 
Simple Linear Regression (SLR)

The regression model looks like:

\[ Y = \beta_0 + \beta_1 X + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2) \]

\( \beta_1 \) measures the effect on \( Y \) of increasing \( X \) by one

\( \beta_0 \) measures the effect on \( Y \) when \( X = 0 \).

\( X_f \) will denote a new/future value we wish to predict at
Fitted Values

The Fitted Values and Residuals have some special properties ...

Let’s look at the fitted values

Our predictions $\hat{Y}_i = \beta_0 + \beta_1 X_i$ are given by the line!!
Residuals

What is the “residual”, $e_i$, for the $i$th observation?

We can write $Y_i = \hat{Y}_i + (Y_i - \hat{Y}_i) = \hat{Y}_i + e_i$
Standardized Residuals

The residuals are $e_i = Y_i - \hat{Y}_i$. They estimate the errors from the line.

We re-scale the residuals by their standard errors. This lets us define standardized residuals

$$r_i = \frac{e_i}{s_{e_i}} = \frac{Y_i - \hat{Y}_i}{s_{e_i}}$$

Outliers are points that are extreme relative to our model predictions. They simply have large residuals!
Residual Standard Error

How closely does the training dataset lie to our model?

- $s$ is the residual standard error
- $s$ is our estimate of $\sigma$
- $s = \sqrt{s^2}$ where

$$s^2 = \frac{1}{n-2} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

Lower $s$ values means tighter predictions!!
Prediction

Suppose you have a regression of sales on price

\[ \text{sales} = \beta_0 + \beta_1 \text{price} \]

You have to predict for a given level of price. Then the two intervals correspond to

1. A sales forecast for the next store (or next week’s sales)
2. The average weekly sales (over many weeks)
Prediction

**Prediction** is the most important application of your model. Construct a new $X$ variable

```r
new = data.frame(price=5)
predict.lm(model,new,interval="prediction")
predict.lm(model,new,interval="confidence")
```

Define a vector for prediction

```r
new1 = data.frame(price=c(4,5,6))
predict.lm(model,new1,interval="prediction")
predict.lm(model,new1,interval="confidence")
```
Confidence and Prediction Intervals

<table>
<thead>
<tr>
<th>fit</th>
<th>lwr</th>
<th>upr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>431.6129</td>
<td>397.0925</td>
</tr>
<tr>
<td>1</td>
<td>431.6129</td>
<td>416.7968</td>
</tr>
<tr>
<td>1</td>
<td>474.1935</td>
<td>432.2873</td>
</tr>
<tr>
<td>2</td>
<td>431.6129</td>
<td>397.0925</td>
</tr>
<tr>
<td>3</td>
<td>389.0323</td>
<td>355.4325</td>
</tr>
<tr>
<td>3</td>
<td>431.6129</td>
<td>416.7968</td>
</tr>
<tr>
<td>3</td>
<td>389.0323</td>
<td>376.5104</td>
</tr>
</tbody>
</table>
Least Squares Principle

Ideally we want to minimize the size of all of the residuals:

- If they were all zero we would have a perfect line

We’ll use the least squares objective function to assess what constitutes a good “fit” to our empirical data. The line fitting process:

- Minimize the “total” sums of squares of the residuals to get the “best” fit

Least Squares chooses $\beta_0$ and $\beta_1$ to minimize $\sum_{i=1}^{n} e_i^2$

$$\sum_{i=1}^{n} e_i^2 = e_1^2 + \ldots + e_n^2 = (Y_1 - \hat{Y}_1)^2 + \ldots + (Y_n - \hat{Y}_n)^2$$
Least Squares Principle

The formulas for $\beta_0$ and $\beta_1$ that minimize the least squares are:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$
$$\beta_1 = r_{xy} \times \frac{s_y}{s_x}$$

where

- $\bar{x}$ and $\bar{y}$ are the sample means
- $s_x$ and $s_y$ are the sample standard deviations
- $r_{xy} = corr(x, y)$ is the sample correlation
Least Squares Principle

1. Intercept

\[ \beta_0 = \bar{y} - \beta_1 \bar{x} \quad \text{or} \quad \bar{y} = \beta_0 + \beta_1 \bar{x} \]

The point \((\bar{x}, \bar{y})\) is always on the regression line.

2. Slope

\[ \beta_1 = \text{corr}(x, y) \times \frac{s_Y}{s_X} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \]

\[ = \frac{\text{cov}(x, y)}{\text{Var}(x)} \]

The estimate \(b\) is the correlation \(r\) times a \textbf{scaling factor} that ensures the proper units.
Sampling Distribution for $\beta_1$

Run linear regression several times using subsample of rows of the housing data.
The sampling distribution of $\beta_1$ describes how it varies over different samples. It allows us to calculate confidence and prediction intervals. Everything is uncertain!!

It turns out that $\beta_1$ is normally distributed: $\beta_1 \sim N(\hat{\beta}_1, s_b^2)$

- $\hat{\beta}_1$ is unbiased: $E(\beta_1) = \hat{\beta}_1$
- $s_{\beta_1}$ is the standard error of $\beta_1$ The t-stat is $t_b = \beta_1 / s_{\beta_1}$
- The three factors: sample size ($n$), error variance ($s^2$), and $x$-spread, $s_x$

$$s_{\beta_1}^2 = \frac{s^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{s^2}{(n-1)s_x^2}$$
Prediction: revisited

How do we assess how much error that could be in our best prediction?

\[ \hat{Y}_f = \beta_0 + \beta_1 X_f + e_f \text{ where } e_f \sim N(0, s^2) \]

There’s error in everything, \( \beta_0, \beta_1, e_f, \ldots \)

After we account for all the uncertainty,

\[
\text{var}(y_f) = s^2 \left( 1 + \frac{1}{n} + \frac{(x_f - \bar{x})^2}{(n-1)s_x^2} \right)
\]

In R: `predict.lm( ... )`
Prediction errors

A large predictive error variance (high uncertainty) comes from four factors

1. Large $s$ (i.e. large errors, $\epsilon$’s)
2. Small $n$ (not enough data)
3. Small $s_x$ (not enough spread in the covariates)
4. Large difference between $x_f$ and $\bar{x}$ (predicting extremes)

As a practical matter, low $s$ values are more important for prediction than high $R^2$-values.
Example: Google Stock Returns

Let's use the quantmod package to read in the data

```r
library(quantmod)
Y = getSymbols("GOOG", from = "2005-01-01")
# Retrieve closing prices
y = GOOG$GOOG.Adj.Close
head(y)
[1] 101.25392 97.15301 96.65851 94.18098 96.82834 97.43274
tail(y)
[1] 796.42 794.56 791.26 789.91 791.55 785.05
```
Example: Google

Consider a CAPM regression for Google's stock

\[ \text{Google}_t = \alpha + \beta \text{sp500}_t + \epsilon_t \]

In finance \((\alpha, \beta)\) are used instead of \((\beta_0, \beta_1)\).

We’d like to know our estimates \((\hat{\alpha}, \hat{\beta})\).

Then formulate lots of hypothesis tests:

\(H_0:\) is Google related to the market?

\(H_0:\) does Google out-perform the market in a consistent fashion?

\(H_0:\) is Google better than Nvidia?
Example: Google
Example: Google

`summary(model)` command provides all of our estimates ...

```r
> summary(model)
Call:
  lm(formula = ret ~ SP500)

Residuals:
   Min     1Q   Median     3Q    Max
-0.0390 -0.0079  0.0001  0.0078  0.0490

Coefficients:  (5 not shown)
Estimate Std. Error t value  Pr(>|t|)
(Intercept) 0.0004086 0.0002936   1.392    0.164
SP500       0.9232752 0.0232625  39.689  < 2e-16 ***
```

Residual standard error: 0.01546 on 223 degrees of freedom
Multiple R-squared: 0.3622, Adjusted R-squared: 0.3500

Our best estimates are: $\hat{\alpha} = 0.0004$, $\hat{\beta} = 0.9232$

How much will Google move if the market goes up 10%?

What do the t-ratios show?
### Example: Keynes Investment Performance

Keynes’ investment returns for Cambridge Universities King’s College endowment

<table>
<thead>
<tr>
<th>Year</th>
<th>Keynes</th>
<th>Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>1928</td>
<td>-3.4</td>
<td>7.9</td>
</tr>
<tr>
<td>1929</td>
<td>0.8</td>
<td>6.6</td>
</tr>
<tr>
<td>1930</td>
<td>-32.4</td>
<td>-20.3</td>
</tr>
<tr>
<td>1931</td>
<td>-24.6</td>
<td>-25.0</td>
</tr>
<tr>
<td>1932</td>
<td>44.8</td>
<td>-5.8</td>
</tr>
<tr>
<td>1933</td>
<td>35.1</td>
<td>21.5</td>
</tr>
<tr>
<td>1934</td>
<td>33.1</td>
<td>-0.7</td>
</tr>
<tr>
<td>1935</td>
<td>44.3</td>
<td>5.3</td>
</tr>
<tr>
<td>1936</td>
<td>56.0</td>
<td>10.2</td>
</tr>
<tr>
<td>1937</td>
<td>8.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>1938</td>
<td>-40.1</td>
<td>-16.1</td>
</tr>
<tr>
<td>1939</td>
<td>12.9</td>
<td>-7.2</td>
</tr>
<tr>
<td>1940</td>
<td>-15.6</td>
<td>-12.9</td>
</tr>
<tr>
<td>1941</td>
<td>33.5</td>
<td>12.5</td>
</tr>
<tr>
<td>1942</td>
<td>-0.9</td>
<td>0.8</td>
</tr>
<tr>
<td>1943</td>
<td>53.9</td>
<td>15.6</td>
</tr>
<tr>
<td>1944</td>
<td>14.5</td>
<td>5.4</td>
</tr>
<tr>
<td>1945</td>
<td>14.6</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Example: Keynes Investment Performance

That’s crazy!! Many large losses ... but also big recoveries ... not for everyone!

Keynes vs Cash
Outliers

Residuals allow us to define outliers:

95% of the time we expect the standardized residuals to satisfy $-2 < r_i < 2$

Any observation with $|r_i| \geq 3$ is an extreme outlier

Residuals will also help in assessing the validity of our model ...
Influential points are observations that affect the magnitude of our estimates $\hat{\beta}_1$. They are important to find as they typically have economic consequences.

We will use **Cook's D** distance to assess the significance of an influential point. They are typically extreme in the characteristics, $X$-space.

We will delete observations with **Cook's D** greater than one and assess the sensitivity of our conclusions.
Influential Points: Cook’s D

Cook’s D depends on the standardized residual, $r_i$, and leverage, $0 < h_i < 1$

$$\text{CookD}_i = \frac{1}{p} \frac{r_i^2}{1 - h_i} h_i$$

where $p$ is the number of variables

```r
plot(model)
plot(cooks.distance(model))
datanew = data[-i,]  # Deletes ith row

Is $\beta_1(-i)$ is different from $\beta_1$?
Influential Points: Cook’s D

They are three ranges: \( 0 < D_i < 0.5, \ 0.1 < D_i < 1 \) and \( D_i > 1 \)

We will delete all observations with \textbf{Cook’s D} > 1

To see how stable our \( \beta_1 \)'s are to these data points

Quite often, I also delete the point with the largest Cook’s D just to check it doesn’t affect my conclusions

All this is done, \textbf{before} I use \texttt{summary(model)} and interpret my model.
Regression: Strategy

Five point basic strategy

1. Input and Plot Data: Use `plot` and `boxplot` commands

2. Build Regression Model: Use the `model = lm (y ~ x)` command

3. Diagnostics: `plot(model)` Fitted vs standardized residuals.
   - QQplot Residuals for Outliers and
   - Cook’s D for Influential

4. Interpretation: `summary(model)`. Regression $\beta$’s

5. Prediction: `predict.lm`.
   - A model is only as good as its predictions. Do some out-of-sample forecasting
Given input-output vectors $x$ and $y$

```r
cor( ... ) computes correlation table
```

```r
model = lm(y ~ x) for linear model (a.k.a regression)
```

```r
model = glm(y ~ x) for logistic regression
```

```r
model = lm(y ~ x1+ ... + xp) for linear multiple regression model
```

R provides diagnostics in

```r
plot(model) 4-in-1 diagnostics plot
```

```r
plot(cooks.distance(model)) influential points
```

```r
rstudent(model) outliers
```
Output and Prediction

R provides model output in

\texttt{summary(model)} provides a summary analysis of our model

R provides predictions in

\texttt{newdata = data.frame( \ldots )} constructs a new input variable
\texttt{predict.lm(model, newdata)} provides a prediction at a new input
Diagnostics: **plot(model) 4-in-1 plot**

Everything in `plot(model)` our 4-in-1 residual plot

1. **Residuals vs Fitted**: Straight line. Random looking pattern

2. **Scale-location**: Ought to be a straight line. Otherwise changing variance

3. **Normal Q-Q Plot**: Standardized residuals. This should be a straight line.
   You’re plotting quantiles of the standardized residuals vs what you’d expect if the assumptions are true, a standard normal

4. **Residuals vs Leverage**: Contours of Cook’s $D$. If $D \geq 1$ then influential.

   Remove and see what happens!!

**In R**: simply use `plot(model)`
Example: Lung Cancer Data

Famous dataset linking lung cancer and cigarette consumption.

\[ Y = \text{lung cancer deaths/million in 1950} \]
\[ X = \text{cigarette consumption/capita in 1930} \]

<table>
<thead>
<tr>
<th>Country</th>
<th>Y</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Iceland</td>
<td>58</td>
<td>220</td>
</tr>
<tr>
<td>2. Norway</td>
<td>90</td>
<td>250</td>
</tr>
<tr>
<td>3. Sweden</td>
<td>115</td>
<td>310</td>
</tr>
<tr>
<td>4. Canada</td>
<td>150</td>
<td>510</td>
</tr>
<tr>
<td>5. Denmark</td>
<td>165</td>
<td>380</td>
</tr>
<tr>
<td>6. Australia</td>
<td>170</td>
<td>455</td>
</tr>
<tr>
<td>7. United States</td>
<td>190</td>
<td>1280</td>
</tr>
<tr>
<td>8. Holland</td>
<td>245</td>
<td>460</td>
</tr>
<tr>
<td>9. Switzerland</td>
<td>250</td>
<td>530</td>
</tr>
<tr>
<td>10. Finland</td>
<td>350</td>
<td>1115</td>
</tr>
<tr>
<td>11. Great Britain</td>
<td>465</td>
<td>1145</td>
</tr>
</tbody>
</table>
Cancer and Smoking Data

The US and UK are well off the regression line
4 in 1 Residual Plots for Model 1

- **Residuals vs Fitted**: Shows the relationship between residuals and fitted values. The points are scattered along the horizontal line, indicating no apparent pattern.

- **Scale–Location**: Displays the square root of standardized residuals vs fitted values. The plot shows a linear trend, suggesting homoscedasticity.

- **Normal Q–Q**: Plots standardized residuals against theoretical quantiles. The points follow a straight line, indicating normal distribution of residuals.

- **Residuals vs Leverage**: Depicts Cook's distance against leverage values. Points are distributed fairly evenly, with no obvious outliers.

The plots collectively help in assessing the model's fit and identifying potential outliers or unusual patterns in the residuals.
4 in 1 Residual Plots for Model 2

Residuals vs Fitted

Scale–Location

Normal Q–Q

Residuals vs Leverage

Theoretical Quantiles

Standardized residuals

Leverage

Residuals

Cook's distance

Standardized residuals
Model Coefficients

There are two ways to get the model coefficients in R

1. `coef(model)`
   
   (Intercept) Consumption
   66.8434535    0.2286585

2. `lm(formula = Cancer ~ Consumption)`

   Coefficients:
   (Intercept) Consumption
   66.8435        0.2287
Transformations

Basic assumption is linearity What if this doesn’t hold?

1. A simple solution is to transform the variables.

2. Re-run the regression on the transformed

3. If all is fine then the model holds on the transformed scale.

   Then transform back to the original nonlinear scale.

The two most common models are

Power relationship

Exponential relationship
Orange Juice

- 83 Chicagoland Stores (Demographic info for each)
- Price, sales (log units moved), and whether advertised (feat)
Orange Juice: Price vs Sales
Orange Juice: Price vs log(Sales)
Orange Juice: Price vs log(Sales)
Orange Juice: \( \log(\text{Price}) \) vs \( \log(\text{Sales}) \)

Why? Multiplicative (rather than additive) change.
The log-log Model

Power/Multiplicative Model

**Multiplicative Model:** \( Y = AX^b \) where \( A = e^a \)

**Log-Log Transformation:** \( \log(Y) = \beta_0 + \beta_1 \log(X) \)

**Why?** Taking logs of both sides gives

\[
\log Y = \log A + \log X^b = \beta_0 + \beta_1 \log X
\]

The slope, \( \beta_1 \), is an elasticity. % change in \( Y \) versus % change in \( X \)

Variables are related on a multiplicative, or **percentage**, scale.

In \( \text{R} \): `model = lm(log(y) ~ log(x))`

Recall: \( \log \) is the natural \( \log_e \) with base \( e = 2.718 \ldots \) and that \( \log(ab) = \log \beta_0 + \log b \) and \( \log(a^b) = b \log a. \)
The Exponential Model

Suppose that we have an equation: \( Y = Ae^{bX} \) where \( A = e^a \).

This is equivalent to \( \log(Y) = \beta_0 + \beta_1X \)

Taking logs of the original equation gives

\[
\log Y = \log A + \beta_1X
\]

\[
\log Y = \beta_0 + \beta_1X
\]

Therefore, we can run a regression of \( \log Y \) on \( X \)!!

**Caveat:** not all variables can be logged!

\( Y > 0 \) needs to be positive.

**Dummy variables** \( X = 0 \) or 1 can’t be logged.

**Counting variables** are usually left alone as well.
First of all, read in and attach our data ...

```r
mammals = read.table("http://faculty.chicagobooth.edu/nicholas.polson/teaching/41000/mammals.txt")
attach(mammals)
head(mammals)
```

<table>
<thead>
<tr>
<th>Mammal</th>
<th>Brain</th>
<th>Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>African_elephant</td>
<td>6654.000</td>
<td>5712.0</td>
</tr>
<tr>
<td>African_giant_pouched_rat</td>
<td>1.000</td>
<td>6.6</td>
</tr>
<tr>
<td>Arctic_Fox</td>
<td>3.385</td>
<td>44.5</td>
</tr>
<tr>
<td>Arctic_ground_squirrel</td>
<td>0.920</td>
<td>5.7</td>
</tr>
<tr>
<td>Asian_elephant</td>
<td>2547.000</td>
<td>4603.0</td>
</tr>
<tr>
<td>Baboon</td>
<td>10.550</td>
<td>179.5</td>
</tr>
</tbody>
</table>

```r
> tail(mammals)
```

<table>
<thead>
<tr>
<th>Mammal</th>
<th>Brain</th>
<th>Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenrec</td>
<td>0.900</td>
<td>2.6</td>
</tr>
<tr>
<td>Tree_hyrax</td>
<td>2.000</td>
<td>12.3</td>
</tr>
<tr>
<td>Tree_shrew</td>
<td>0.104</td>
<td>2.5</td>
</tr>
<tr>
<td>Vervet</td>
<td>4.190</td>
<td>58.0</td>
</tr>
<tr>
<td>Water_opossum</td>
<td>3.500</td>
<td>3.9</td>
</tr>
<tr>
<td>Yellow-bellied_marmot</td>
<td>4.050</td>
<td>17.0</td>
</tr>
</tbody>
</table>
The residuals show that you need a transformation ....
Residual Plots

Residuals vs Fitted

Scale–Location

Normal Q–Q

Residuals vs Leverage
log-log model

That’s better!
4 in 1 Residuals: log-log model
log-log Model

\[ \text{lm(formula} = \log(\text{Brain}) \sim \log(\text{Body})) \]

Coefficients:

|             | Estimate | Std. Error | t value | Pr(>|t|) |
|-------------|----------|------------|---------|----------|
| (Intercept) | 2.18328  | 0.10682    | 20.44   | <2e-16   ***
| log(Body)   | 0.74320  | 0.03166    | 23.48   | <2e-16   ***

\[ \log(\text{Body}) = 2.18 + 0.74 \log(\text{Brain}). \]

The coefficients are highly significant \( R^2 = 90\% \).
### Outliers

The table below shows the results of the `rstudent(model)` function applied to the dataset:

<table>
<thead>
<tr>
<th>Mammal</th>
<th>Brain</th>
<th>Body</th>
<th>Residual</th>
<th>Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinchilla</td>
<td>64.0</td>
<td>0.425</td>
<td>3.7848652</td>
<td>4.699002</td>
</tr>
<tr>
<td>Man</td>
<td>1320.0</td>
<td>62.000</td>
<td>2.6697886</td>
<td>190.672827</td>
</tr>
<tr>
<td>Rhesus_monkey</td>
<td>179.0</td>
<td>6.800</td>
<td>2.1221002</td>
<td>36.889735</td>
</tr>
<tr>
<td>Baboon</td>
<td>179.5</td>
<td>10.550</td>
<td>1.6651361</td>
<td>51.128826</td>
</tr>
<tr>
<td>Owl_monkey</td>
<td>15.5</td>
<td>0.480</td>
<td>1.4589815</td>
<td>5.143815</td>
</tr>
<tr>
<td>Chimpanzee</td>
<td>440.0</td>
<td>52.160</td>
<td>1.2734358</td>
<td>167.690600</td>
</tr>
</tbody>
</table>

There is a residual value of **3.78 extreme outlier**.

It corresponds to the **Chinchilla**.

This suggests that the Chinchilla is a master race of supreme intelligence!
NO!!! I checked and there was a data entry error.

- The brain weight is given as 64 grams and should only be 6.4 grams.
- The next largest residual corresponds to mankind

In this example the log-log transformation used seems to achieve two important goals, namely linearity and constant variance.
Glossary of Symbols

Intercept, \( \beta_0 \)

Slope, \( \beta_1 \)

Error, \( e \)

Residual standard error, \( s \)

Standardised residual, \( r_i \)

Leverage, \( h_i \)
Summary

- Linear Patterns in Data (Leavitt, House Price)
- Simple Linear Regression
- Predictions (Confidence and Prediction Intervals)
- Least Squares Principle
- Hypothesis Testing (Google vs SP500)
- Model Diagnostics (Cancer and Smoking Data)
- Data transformations (World’s Smartest Mammal)