Section 3: Linear Regression

Vadim Sokolov

Suggested Reading

OpenIntro Statistics, Chapters 4,5&6

Last Section

- Estimating Parameters and Fitting Distributions
- Confidence and Prediction Intervals
- Means, Proportions, Differences
- A/B Testing

This Section

- 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

Regression: Introduction

Probab.   Stats

Modeling   A/B testing

Binom, Pois, Normal → Analyzed Avg.

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

- Regression provides a conceptuall 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

$X \longrightarrow$ Rule / Model / Equation $\longrightarrow y$

input (dependent)   output (Response)

AIQ: Leavitt Stars Data

intercept

$y = a + bx$;

slope

Stats

intercept

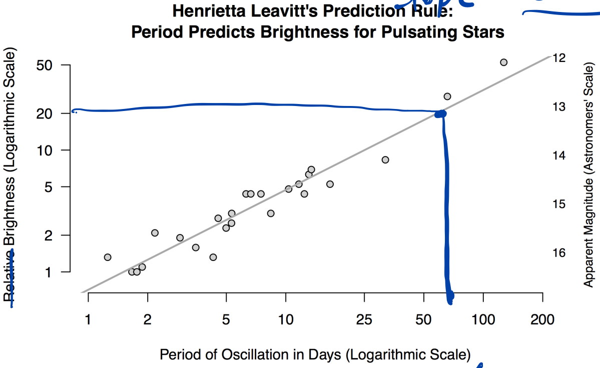
$y = \beta_0 + \beta_1 x$

slope

y

True Brightness

Intrinsic



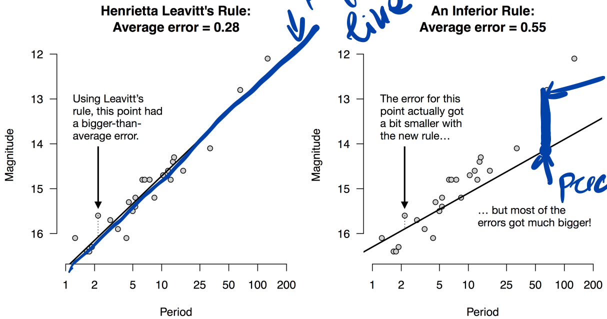
X

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.

Fitting Prediction Rules to Data

In AI, the criterion for evaluating prediction rules is simple: How big are the errors the rule makes, on average?

Regression line



True y

Predicted y

Gauss

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

$y = \beta_0 + \beta_1 X$

Values of B_0 & B_1 calculated from sample

Model

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

X: y:

Sqft Price

land

Bath

Beds

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

⋮

Solution:

Location

- 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

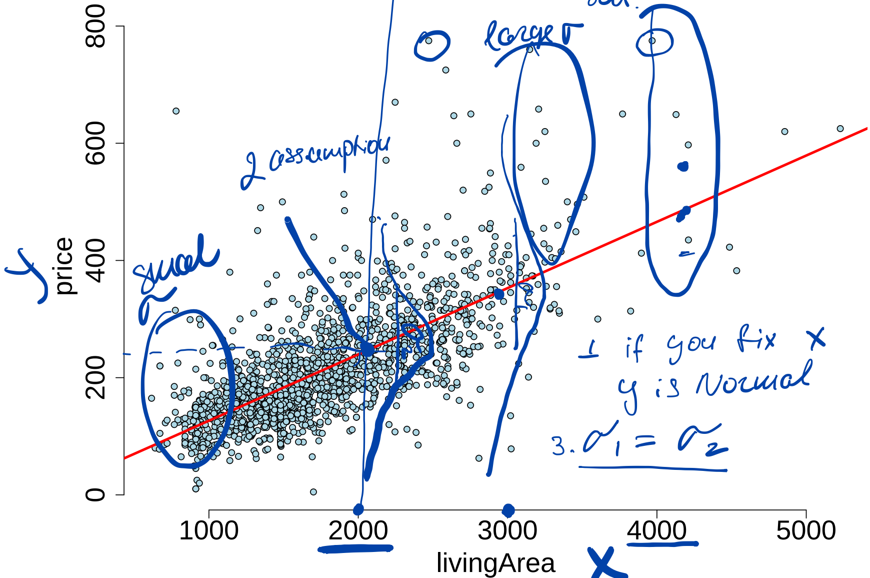
The value that we seek to predict is called the dependent (or output) variable, and we denote this by $Y =$ price of house (e.g. thousand of dollars)

The variable that we use to construct our prediction is the explanatory (or input) variable, and this is labeled $X =$ size of house (e.g. thousand of square feet)

Predicting House Prices

1. Unobs. variables loc age
2. Humans are random New kitchen
3. Season.

What's does the data look like?



y

small

larger σ

2 assumption

1 if you fix x
y is Normal

3. $\sigma_1 = \sigma_2$

x

Predicting House Prices

Price $\approx \beta_0 + \beta_1$ Sqft

approx

Simple Linear Regression (SLR) model

No house

$$\overset{y}{\text{price}} = \beta_0 + \beta_1 \overset{x}{\text{sqft}} + \epsilon \text{ where } \epsilon \sim N(0, \sigma^2)$$

sqft = 0

price = β_0 \$

where we add a random error term, ϵ.

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

price of land.

We find that $\boxed{\beta_1 = 0.11}$ $\boxed{\beta_0 = 13.44}$

Implication: every 1 sqft increase ups price by \$110K

\$ P = $\beta_0 + \beta_1$ SqSt

β_1 - Price per sqft (on average)

Units $\frac{\$}{\text{sqft}}$

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 $X = \boxed{2200}$

Predicted Price: $\hat{Y} = 13.44 + 0.11(2200) = \boxed{262}$

The intercept $\beta_0 = 13.44$ measures land value. In `R:` predict.lm(...)

$y = 13.44 + 0.11X$

Predicting House Prices

Now plot and run your regression ...

```
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)

When violated:
Linear model is not ideal
β_0 & β_1 are not correctly estimated

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*. Violated for time series. eg daily sales.

In practice: Allways violated

Simple Linear Regression (SLR)

if fix x

$$y = \underbrace{\beta_0 + \beta_1 x}_{\text{constant}} + \underbrace{\varepsilon}_{N(0,\sigma^2)}$$

The regression model looks like:

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

$$y \sim N(\underbrace{\beta_0 + \beta_1 x}, \sigma^2)$$

does not depend on x

β_1 measures the effect on Y of increasing X by one

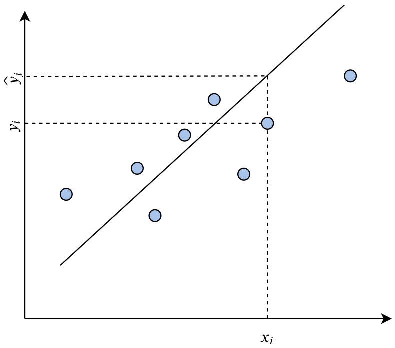
β_0 measures the effect on Y when $X = 0$.

X_f will denote a new/future value we wish to predict at

Fitted Values

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

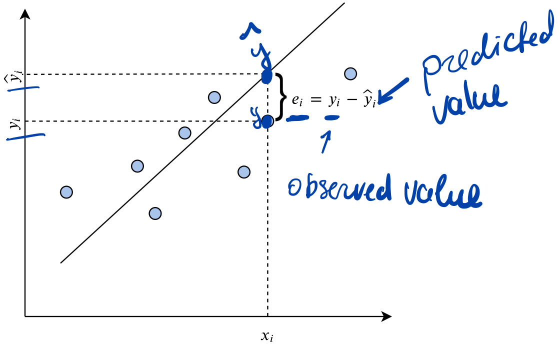
Let's look at the fitted values



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 ith 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

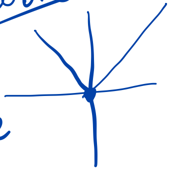
$$r_i = \frac{e_i}{s_{e_i}} = \frac{Y_i - \hat{Y}_i}{s_{e_i}}$$

Outliers are points that are extreme relative to our model predictions.

They simply have large residuals!

Residual Standard Error

$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$ Hard problem



How closely does the training dataset lie to our model?

easy to solve

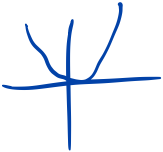
- s is the residual standard error
- s is our estimate of σ
- $s = \sqrt{s^2}$ where

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

estimation / fitting: find β_0 & β_1 that minimize s^2

Lower s values means tighter predictions!!

large^2 → larger
small^2 → smaller.



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

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

Define a vector for prediction

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

Confidence and Prediction Intervals

`lwr` lower limit, `upr` upper limit

```
       fit      lwr      upr
1 431.6129 397.0925 466.1333   # Prediction
       fit      lwr     upr
1 431.6129 416.7968 446.429    # Confidence
       fit      lwr      upr
1 474.1935 432.2873 516.0998   # Multiple Prediction
2 431.6129 397.0925 466.1333
3 389.0323 355.4325 422.6320
       fit      lwr      upr   # Multiple Confidence
1 474.1935 446.1938 502.1933
2 431.6129 416.7968 446.4290
3 389.0323 376.5104 401.5541
```

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 β_0 and β_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

least squares

$$\frac{1}{n}\sum(y_i-\hat{y}_i)^2 \rightarrow \text{minimize}$$

The formulas for β_0 and β_1 that minimize the least squares are:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_1 = r_{xy} \times \frac{s_y}{s_x}$$

$$\frac{\text{units of } y}{\text{units of } x}$$

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

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \ \text{ or } \ \bar{y} = \beta_0 + \beta_1 \bar{x}$$

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

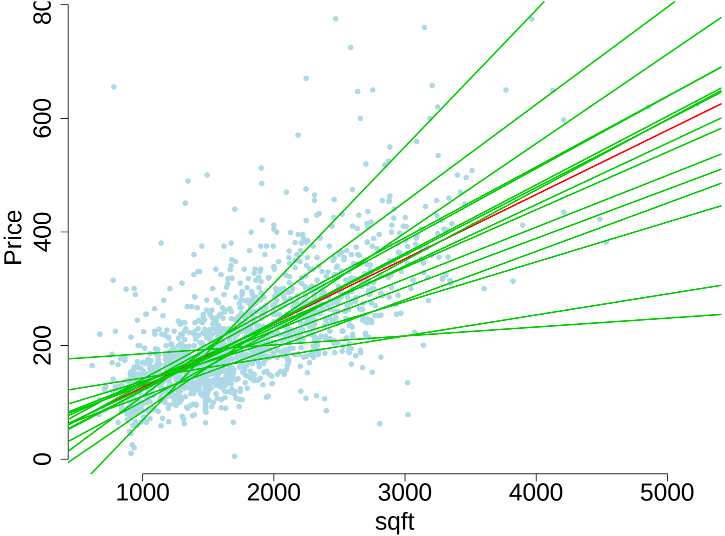
2. Slope

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

The estimate *b* is the correlation *r* times a scaling factor that ensures the proper units

Sampling Distribution for β_1 is uncertain follows Normal distrib.

Run linear regression several times using subsample of rows of the housing data set



Sampling Distribution for β_1

The sampling distribution of β_1 describes how it varies over different samples.

It allows us to calculate confidence and prediction intervals. Everything is uncertain!!

estimated from sample

It turns out that β_1 is normally distributed: $\beta_1 \sim N(\hat{\beta}_1, s^2_{\beta_1})$

- $\hat{\beta}_1$ is unbiased: $E(\beta_1) = \hat{\beta}_1$
- s_{β_1} is the standard error of β_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

$$s^2_{\beta_1} = \frac{s^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{s^2}{(n-1)s^2_x}$$

$H_0: \beta_1 = 0 \quad H_1: \beta_1 \neq 0$

Accept $H_1 \Rightarrow$ x & y are related

Prediction: revisited

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

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

CI

Var(y_f)

There's error in everything, β_0, β_1, e_f, ...

s^2 ↗ ↗

n ↑ ↘

s_x^2 ↑ ↘

$(x_f - \bar{x})^2$ ↑ ↗

After we account for all the uncertainty,

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

CI: uncert in e

PI: uncert in e + uncert in β_0 & β_1

In R: predict.lm(...)

Prediction errors

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

1. Large s (i.e. large errors, ϵ'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

```
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

x

y

$$\text{Google}_t = \alpha + \beta\,\text{sp500}_t + \epsilon_t$$

day

In finance (α, β) are used instead of (β_0, β_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? $H_0: \beta = 0 \quad H_1: \beta \neq 0$

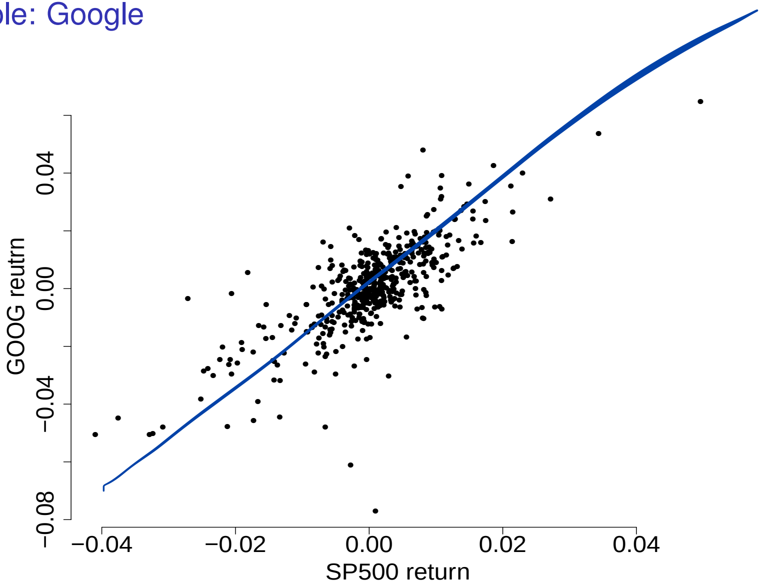
H_0 : does Google out-perform the market in a consistent fashion?

H_0 : is Google better than Nvidia?

$\text{sp500} = 0$

$H_0: \alpha = 0 \quad H_1: \alpha \neq 0$

Example: Google



Example: Google

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

```
> summary(model)
lm(formula = ret ~ SP500)
              Estimate Std. Error t value  P(>|t|)
(Intercept) 0.0004086  0.0002936   1.392   0.164
SP500       0.9232752  0.0232625  39.689   <2e-16 ***
Residual standard error: 0.01546
Multiple R-squared:  0.3622
```

α (Intercept)

β SP500

Cannot Reject

$H_0: \alpha = 0$

$H_0: \beta = 0$

Reject H_0

Accept $H_1: \beta \neq 0$

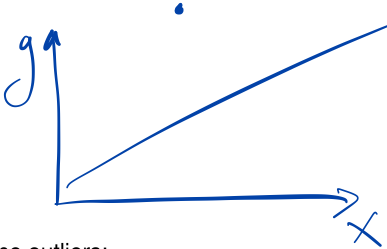
Our best estimates are: $\hat{\alpha} = 0.0004$, $\hat{\beta} = 0.9232$

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

What do the *t*-ratios show?

$$\boxed{goog = 0.0004 + 0.1 \cdot 0.9232}$$

Outliers



Residuals allow us to define outliers:

95%

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

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

Residuals will also help in assessing the validity of our model ...

Influential Points

infl. of point i is change in β_1 when point is removed

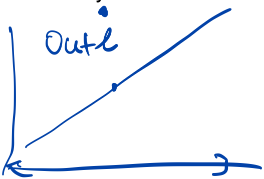
Influential points are observations that affect the magnitude of our estimates $\hat{\beta}_1$.

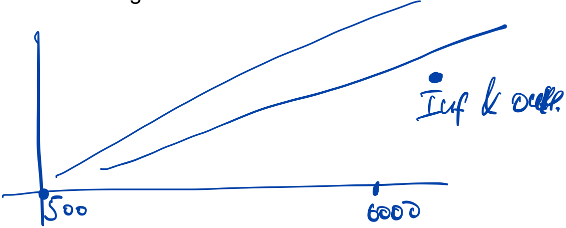
They are important to find as they typically have economic consequences.

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, $0 < h_i < 1$

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

where p is the number of variables

```
plot(model)

plot(cooks.distance(model))

datanew = data[-i,]  # Deletes ith row
```

Is $\beta_{1(-i)}$ is different from β_1?

Influential Points: Cook's D

They are three ranges: $0 < D_i < 0.5$, $0.1 < D_i < 1$ and $\boxed{D_i > 1}$

We will delete all observations with Cook's D > 1

To see how stable our β_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, before I use 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 β's
5. Prediction: `predict.lm`.

   A model is only as good as its predictions. Do some out-of-sample

   forecasting

R Regression Commands

Given input-output vectors x and y

`cor( ... )` computes correlation table

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

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

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

R provides diagnostics in

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

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

`rstudent(model)` outliers

Output and Prediction

`R` provides model output in

`summary(model)` provides a summary analysis of our model

`R` provides predictions in

`newdata = data.frame( ... )` constructs a new input variable

`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

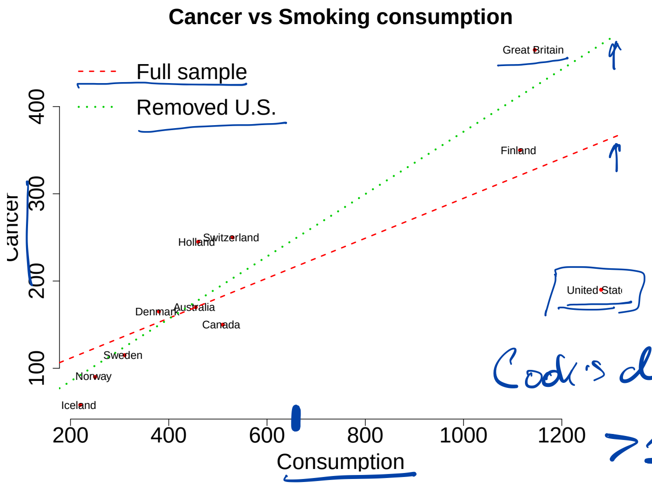
Famous dataset linking lung cancer and cigarette consumption.

$Y =$ lung cancer deaths/million in 1950

$X =$ cigarette consumption/capita in 1930

Country	Y	X	
1. Iceland	58	220	
2. Norway	90	250	
3. Sweden	115	310	
4. Canada	150	510	
5. Denmark	165	380	
6. Australia	170	455	
7. United States	190	1280	Outl & Lvg.
8. Holland	245	460	
9. Switzerland	250	530	
10. Finland	350	1115	
11. Great Britain	465	1145	Outl.

Cancer and Smoking Data



Cook's dist

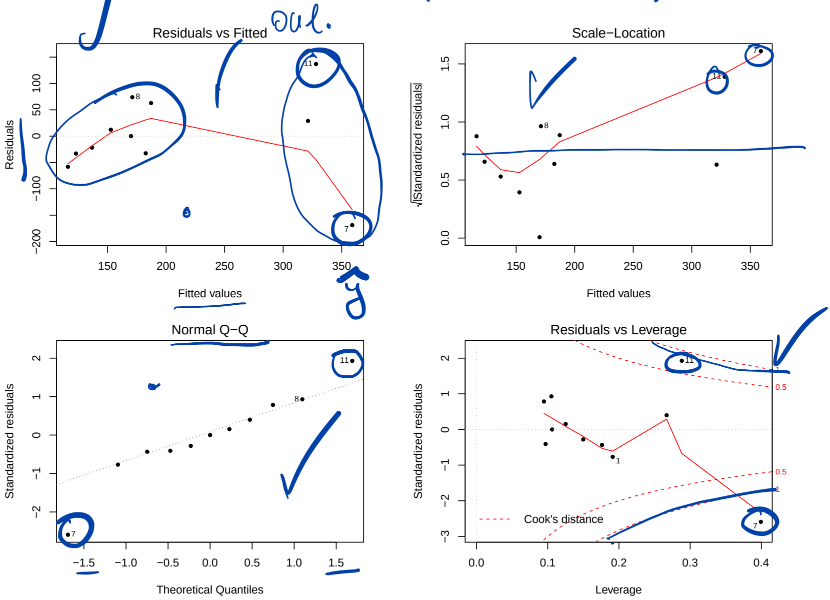
$>1 \Rightarrow$

infl. pt.

The US and UK are well off the regression line

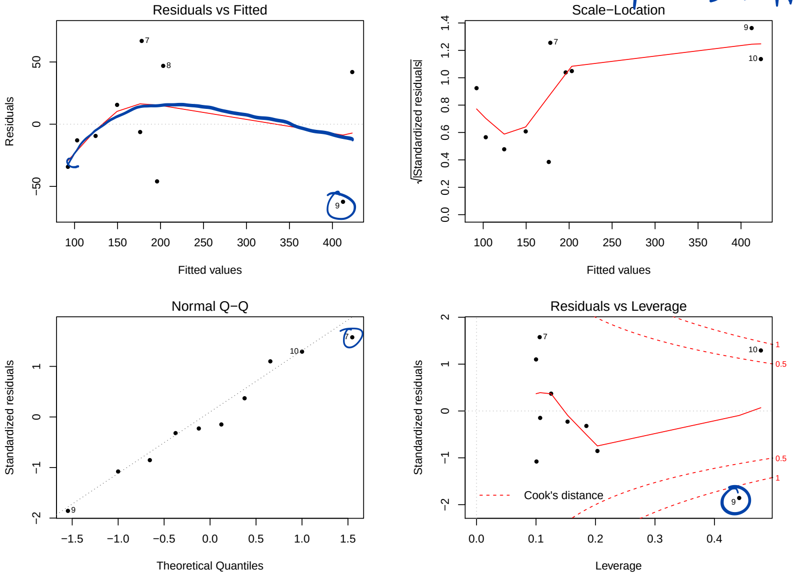
4 in 1 Residual Plots for Model 1

plot(model)



4 in 1 Residual Plots for Model 2

Model 2 satisf.
lm assumptions better.



Model Coefficients

There are two ways to get the model coefficients in `R`

1. coef(model)

   ```
   (Intercept) Consumption
    66.8434535   0.2286585
   ```

   β_0 β_1

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

The log-log Model

Y & X Change on % scale

Power/Multiplicative Model

$y = Ax^b$: $x \to kx$ e.g $k=1.5$

$Ax^b \to A k^b x^b$

Multiplicative Model: $Y = AX^b$ where $A = e^a$

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

$y \to k^b y$

1.1^b

Why? Taking logs of both sides gives

$\log(y) = \log(Ax^b) = \log(A) + \log(x^b) =$

$$\log Y = \log A + \log X^b = \beta_0 + \beta_1 \log X$$

$\log(A) + b\log(x)$

$\log(x^b) = b\log(x)$

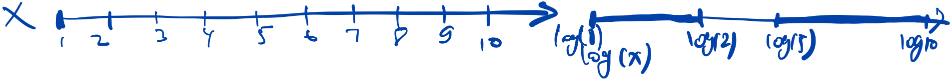
The slope, β_1, is an elasticity. % change in Y versus % change in X

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

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

In 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

X changes on linear scale
y changes on mult scale

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

This is equivalent to $\log(Y) = \beta_0 + \beta_1 X$

Taking logs of the original equation gives

$$\log Y = \log A + \beta_1 X$$

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

Compounding growth
X = time
y = value of acct.

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

$$\log(y) = \log(A) + \log(e^{bX})$$
$$= \log(A) + bX \underbrace{\log(e)}_{1}$$
$$= \log A + bX$$

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.

Example: World's Smartest Mammal

log transform

linear scale →

First of all, read in and attach our data ...

kg Multiplicative scale(%)

```
mammals = read.csv("data/mammals.csv")
attach(mammals)
head(mammals)
                    Mammal    Brain   Body
1         African_elephant 6654.000 5712.0
2 African_giant_pouched_rat    1.000    6.6
3               Arctic_Fox    3.385   44.5
4   Arctic_ground_squirrel    0.920    5.7
5           Asian_elephant 2547.000 4603.0
6                   Baboon   10.550  179.5
> tail(mammals)
                 Mammal        Brain Body
57               Tenrec        0.900  2.6
58           Tree_hyrax        2.000 12.3
59           Tree_shrew        0.104  2.5
60               Vervet        4.190 58.0
61        Water_opossum        3.500  3.9
62 Yellow-bellied_marmot       4.050 17.0
```

exp is opposite to log

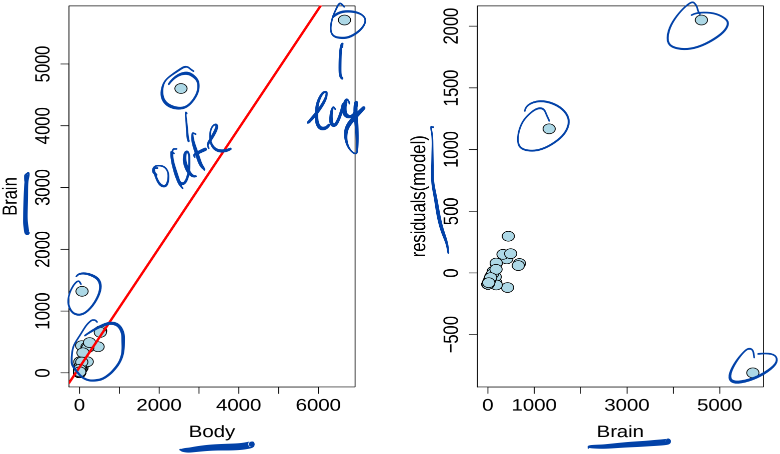
$\exp(x) = e^x$

Residual Diagnostics

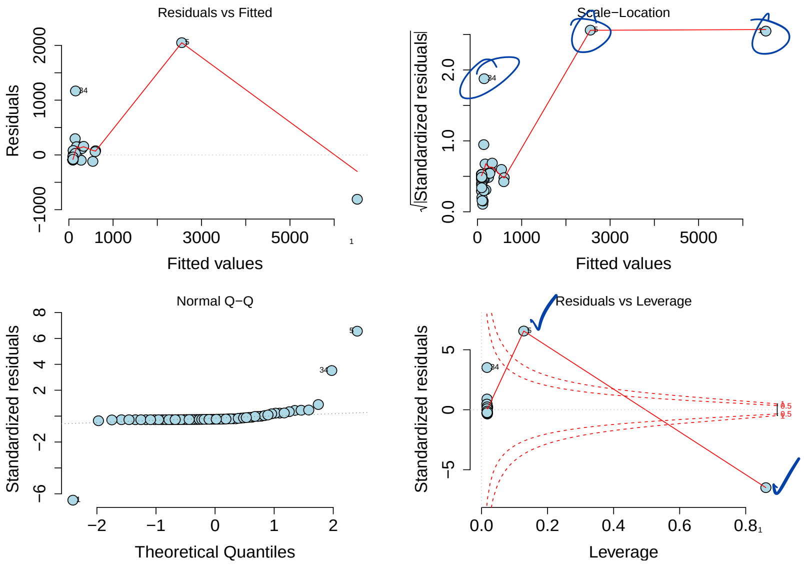
Brain = $\beta_0 + \beta_1$ Body

cannot interp

The residuals show that you need a transformation ...



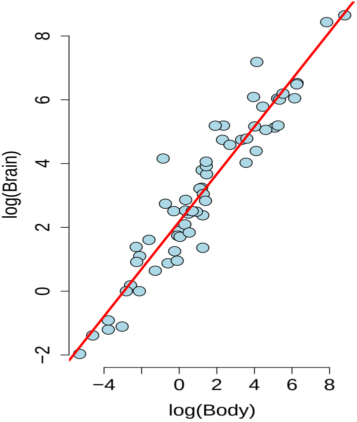
Residual Plots

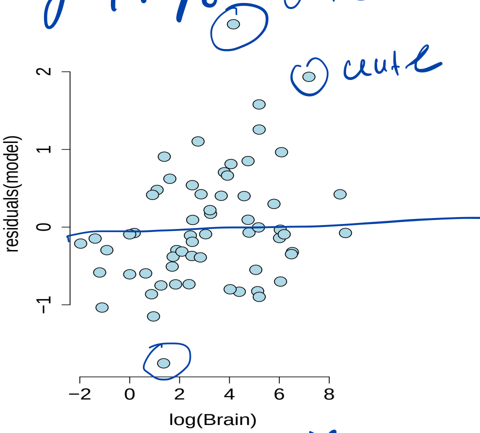


log-log model

if Body ↑ by 1%

Brain ↑ by β_1 %





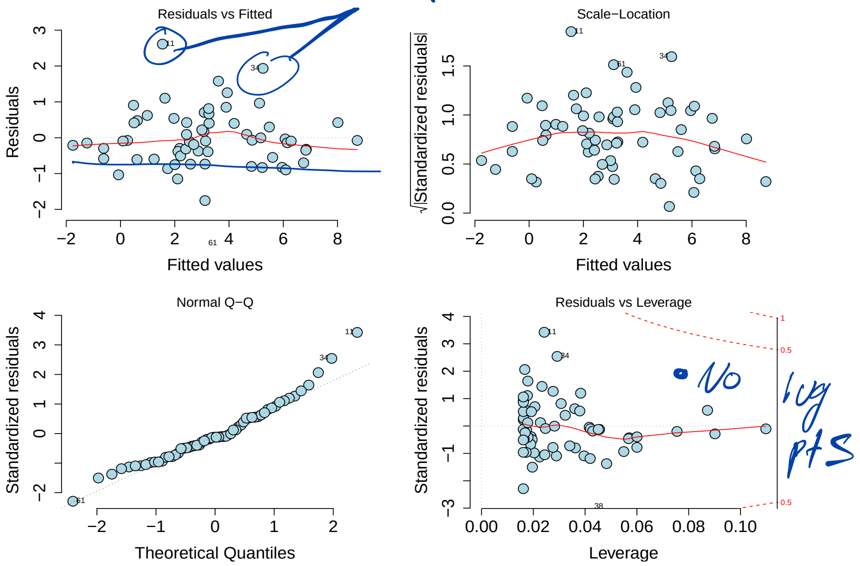
That's better!

$$\underbrace{\log(\text{Brain})}_{y} = \beta_0 + \beta_1 \log(\underbrace{\text{Body}}_{x})$$

- if $\log(\text{Body}) \uparrow$ by 1 unit then $\log(\text{Brain}) \uparrow$ by β_1 units

4 in 1 Residuals: log-log model

inspect



log-log Model

lm(formula = log(Brain) ~log(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 $R^2 = 90\%$

Brain = 1 g

log(Brain) = 0

log(Body) = 2.18

Body = exp(2.18) kg

Outliers

6.4

`rstudent(model)`

```
        Mammal  Brain    Body  Residual        Fit
11  Chinchilla   64.0   0.425 3.7848652   4.699002
34         Man 1320.0  62.000 2.6697886 190.672827
50 Rhesus_monkey 179.0  6.800 2.1221002  36.889735
6       Baboon  179.5  10.550 1.6651361  51.128826
42   Owl_monkey   15.5  0.480 1.4589815   5.143815
10  Chimpanzee  440.0  52.160 1.2734358 167.690600
```

$\hat{y}$ 9

There is a residual value of 3.78 extreme outlier.

if $|res| > 3/2 \Rightarrow$ plim uene of

It corresponds to the Chinchilla.

This suggests that the Chinchilla is a master race of supreme intelligence!

Outl: Large residual animal

Lvg pt: if removed β_1 changes signif.

Inference

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, β_0

Slope, β_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