Linear Regression

Gordon Erlebacher

Where we stand

- Vectors, data.frames, logical expressions
- Extraction, reading data
- Conversions (as.factor, as.matrix)
- Plotting (plot, hist)
- Correlation between variables
- Creation of scripts, functions

What next

- Linear regression (two lesson)
- Working with imaging software
- Working with sound software

All concepts are illustrated through R code

Models

- Consider a population of adults
- We perform an experiment and measure **M**uscle **S**trength **I**ncrease (MSI) as a function of drug dosage (DD)
- The question asked is:
  - what is the relationship between MSI and drug DD?
  - with such a relationship, one might predict MSI for any DD. One might establish confidence intervals for MSI.

Height

- Given a population of adults, we are interested in explaining height: what does height depend on?
  - age? diet? exercise? where one lives?
- Different effects affect the height distribution differently
- We would like to establish the dominant effects and make a variety of predictions

An experiment

- Consider all students taking research psychology in all schools
  - the independent variable X is the grade [0,100] given at the midterm
  - the dependent variable Y is the grade given at the final
- Is there a relationship between X and Y?
- Does a higher midterm grade imply a higher final grade?
- Given some value of X, can anything be said about Y?

Why Regression?

- Often, experiments provide us with variables (called regressive, or quantitative)
- Some questions are:
  - are these variables correlated?
  - does a larger value of one variable affect the value of the other variable?
  - if a value of the independent variable (IV) takes a value different from a value in the input data, what can one say about the dependent variable (DV)?
  - if the IV takes a value *outside* the input data range, can one infer a probable value for the DV?
  - since all data is inexact, can one derivate confidence intervals for the regression (model) variables?

Income v. Happiness

Happiness economics is the quantitative study of happiness, positive and negative affect, well-being, quality of life, life satisfaction and related concepts, typically combining economics with other fields such as psychology and sociology. It typically treats such happiness-related measures, rather than wealth, income or profit, as something to be maximized. The field has grown substantially since the late 20th century, for example by the development of methods, surveys and indices to measure happiness and related concepts.

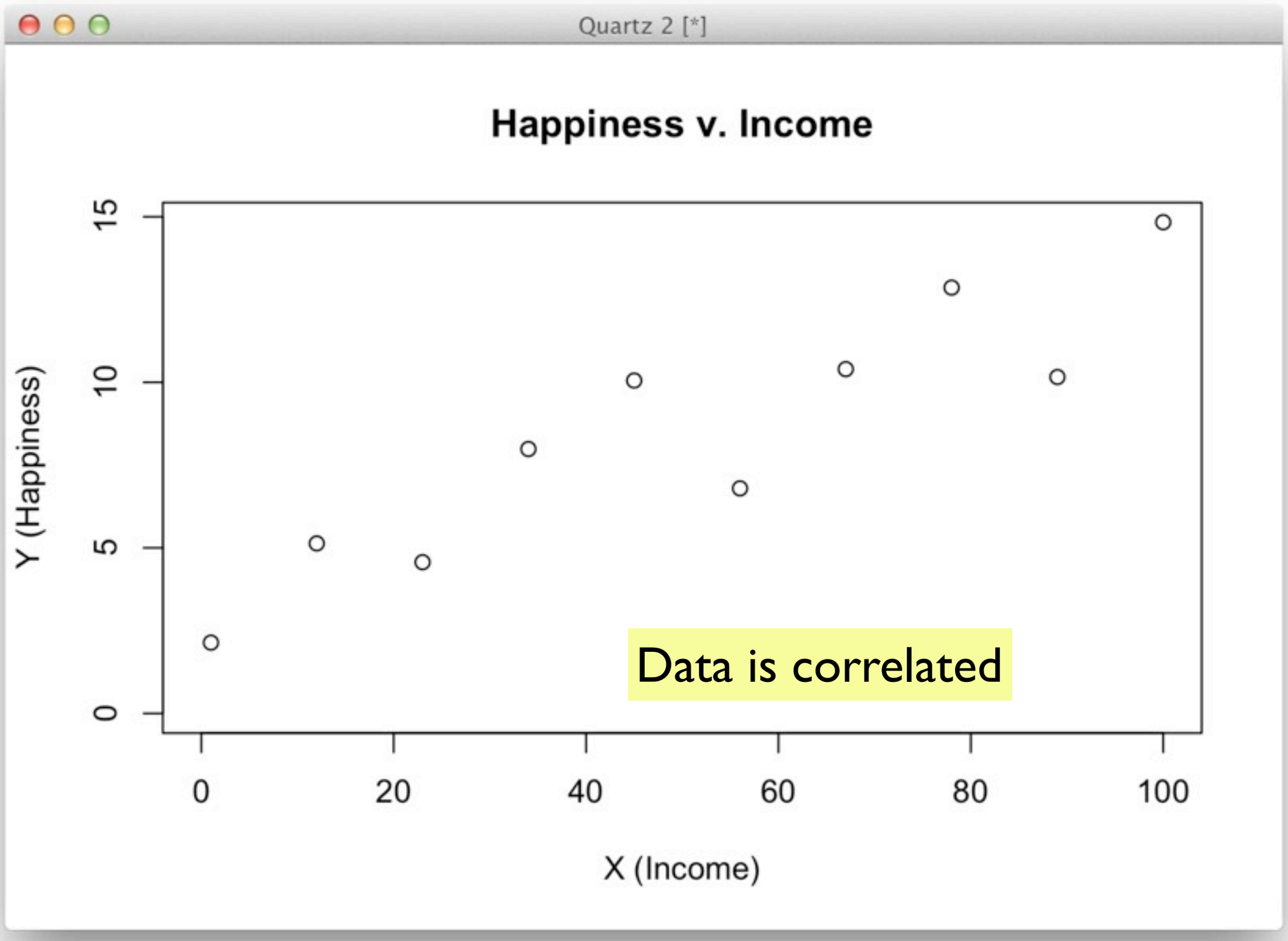
Functional Relationships

Direct causal relationship

$Y = f(X)$

E.g., $Y = b0 + b1*X$

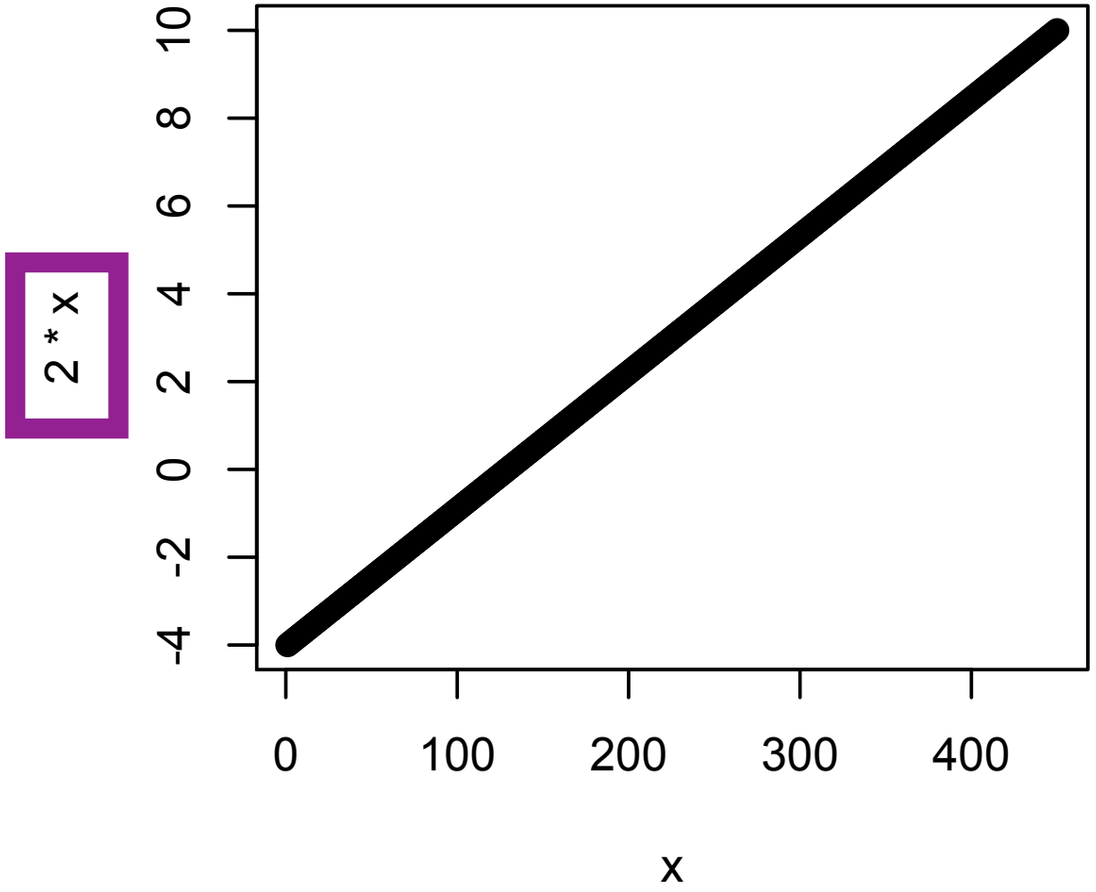
Model: $Y = b0 + b1*X + error$


Quartz 2 [*]
Happiness v. Income
Y (Happiness)
15
10
5
0
0
20
40
60
80
100
X (Income)
Data is correlated

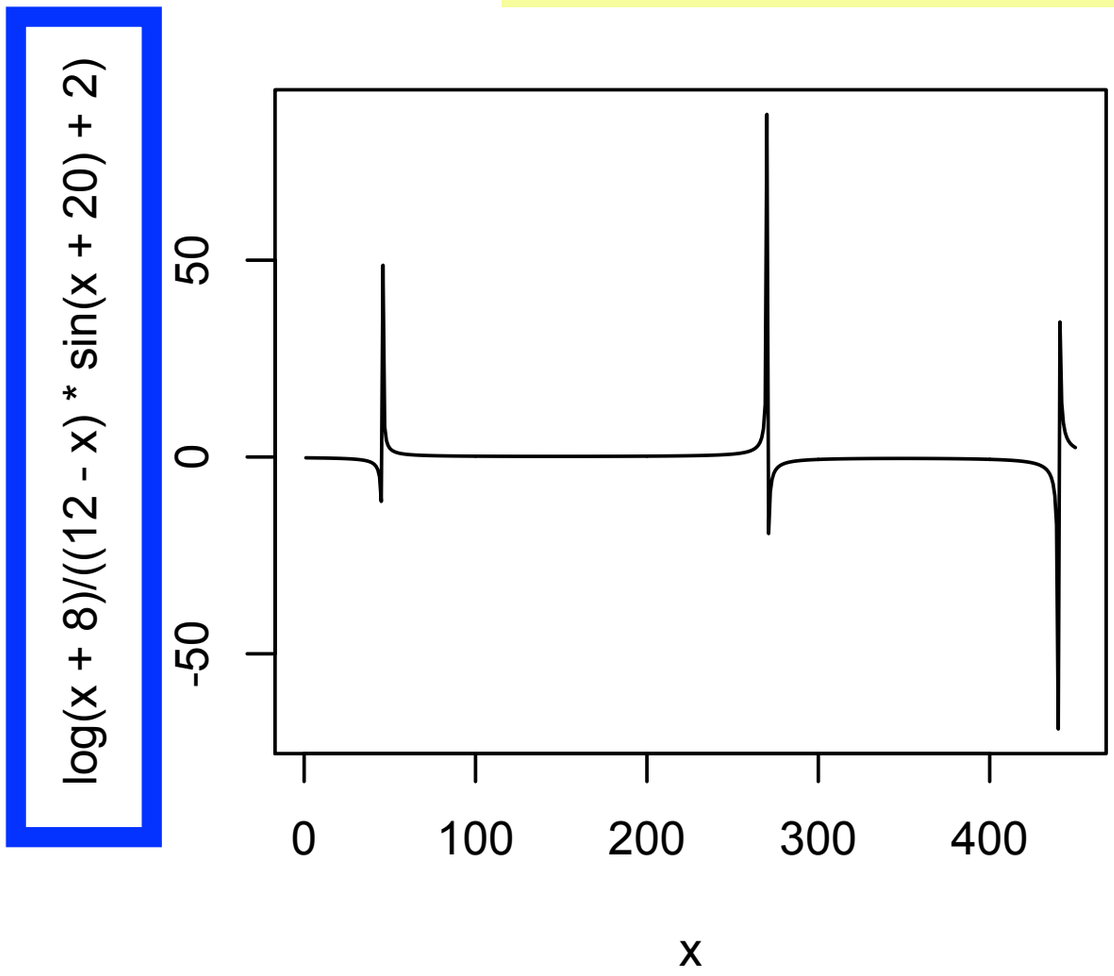
Functional Relation

- y = f(x)
- A single value of x corresponds to a single value of y
- Given a level of income, estimate the level of happiness
  - happiness = 3 * income
  - This is a linear function
- f(x) need not be linear, for example
  - happiness = 3*income + 0.02 * income*income
- A specified **x** always leads to the same **y**

Linear function



Nonlinear function



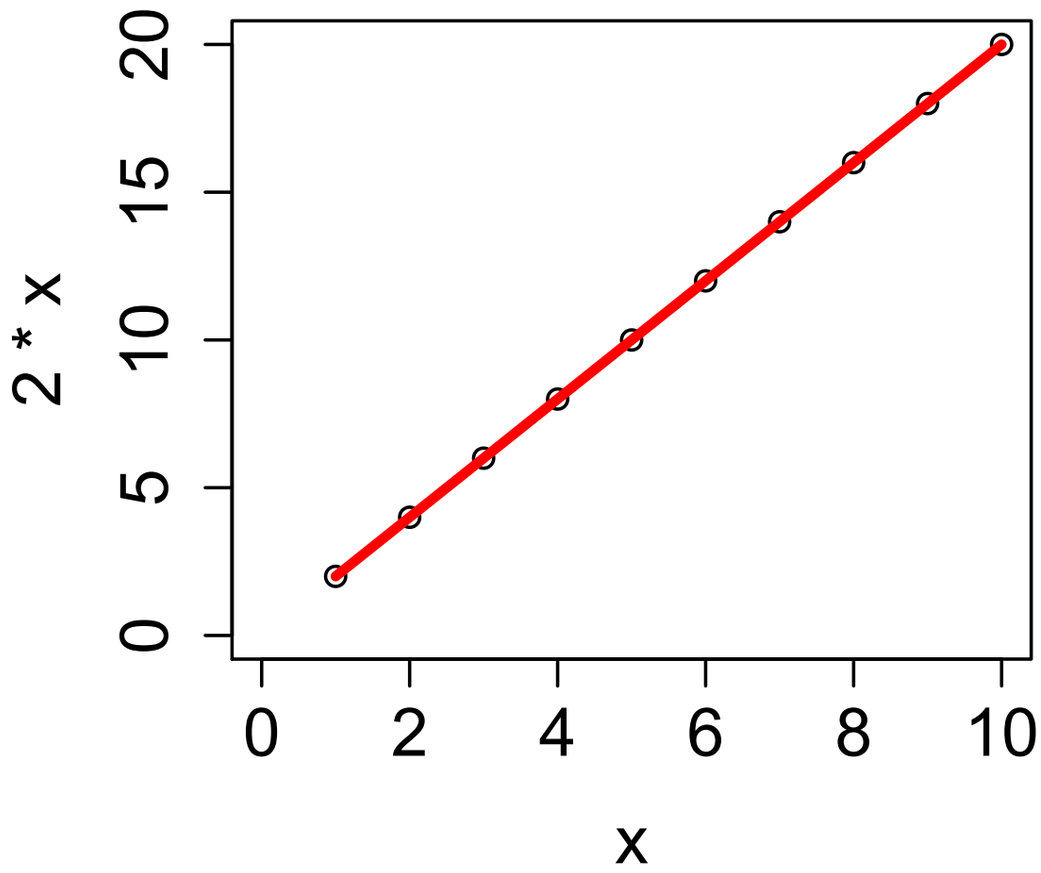

```
par(ps=12, cex=0.5)
par(mfrow=c(2,2))
x = seq(-2,5,length=450)
plot(2*x, xlab="x")
plot(log(x+8)/((12-x)*sin(x+20)+2), xlab="x", type='l')
```

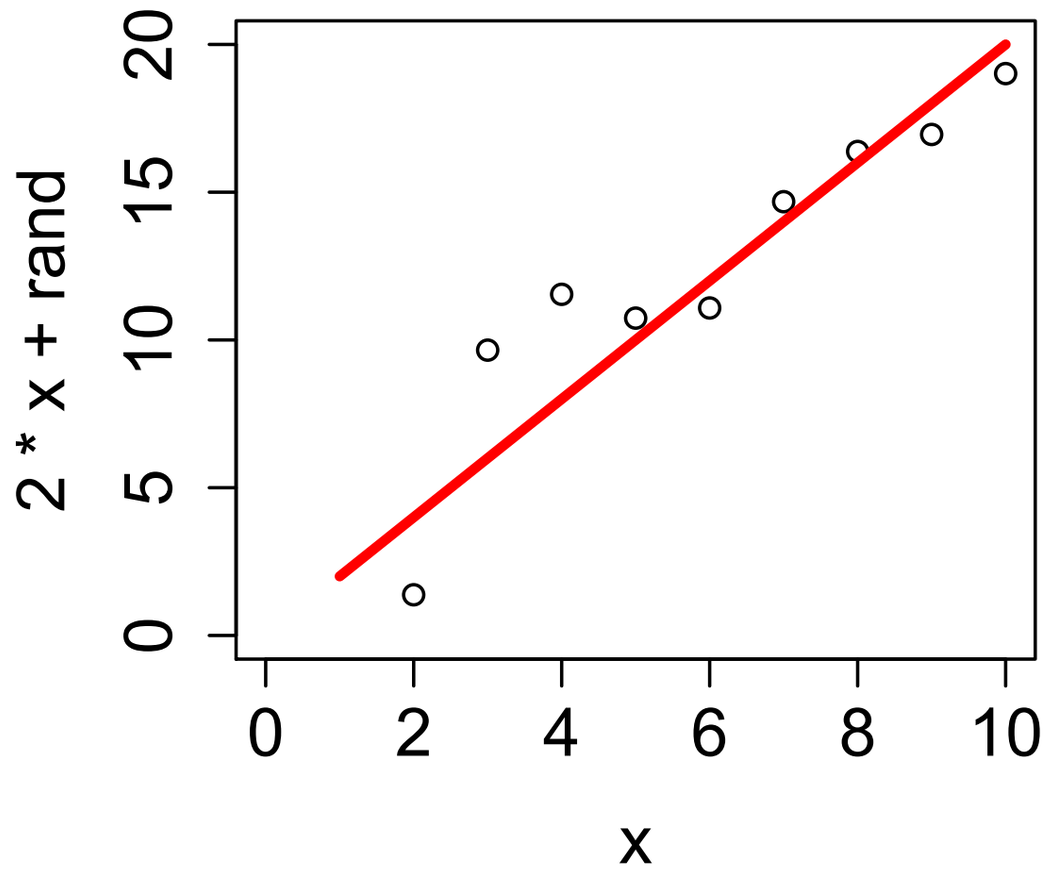
Statistical Relationship

- In reality, an e*xact* relation between happiness and income is not known
- Happiness surely depends on more than income
- happiness = f(income) + error
- error accounts for all unknown influences

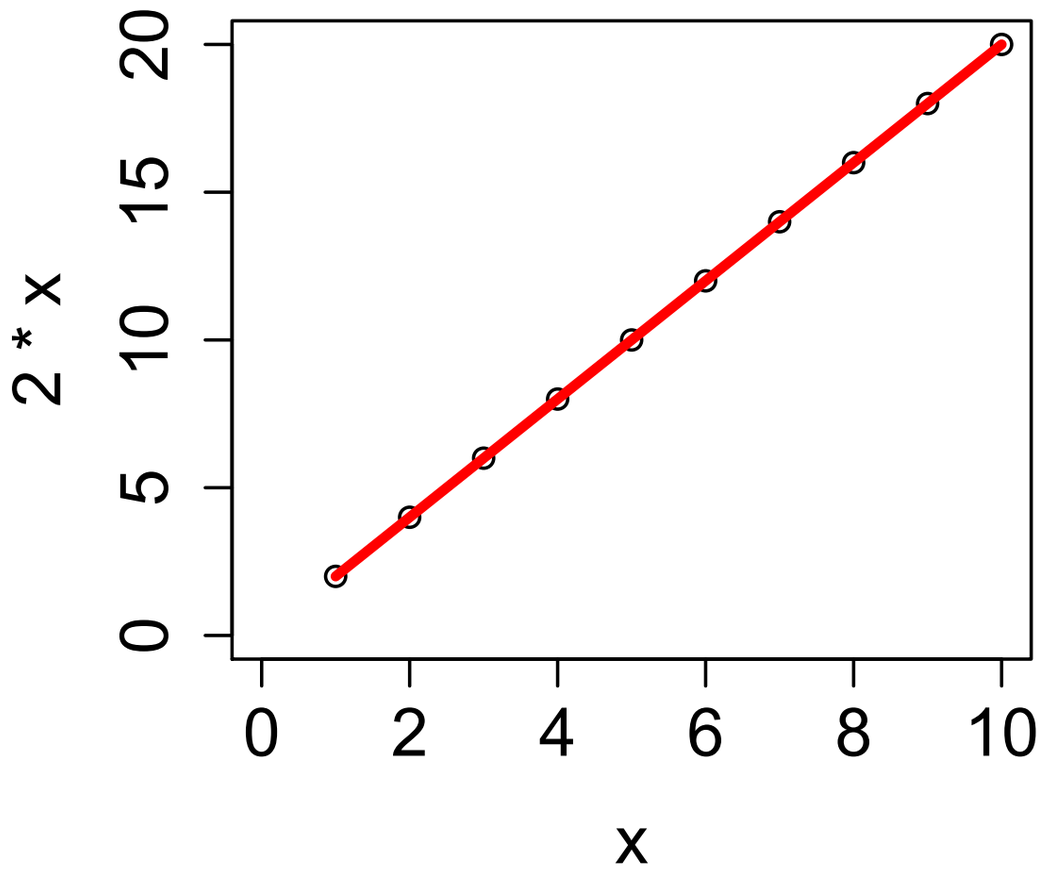
More Generally

- One performs an experiment on "a" sample of a population
- One measures some dependent variable (i.e., a score) and wishes to identify the experimental/observational variables that can explain this score
- In the simplest case, **x** corresponds to a single one of these experimental/observational variables, referred to as an independent/explanatory/predictor variable

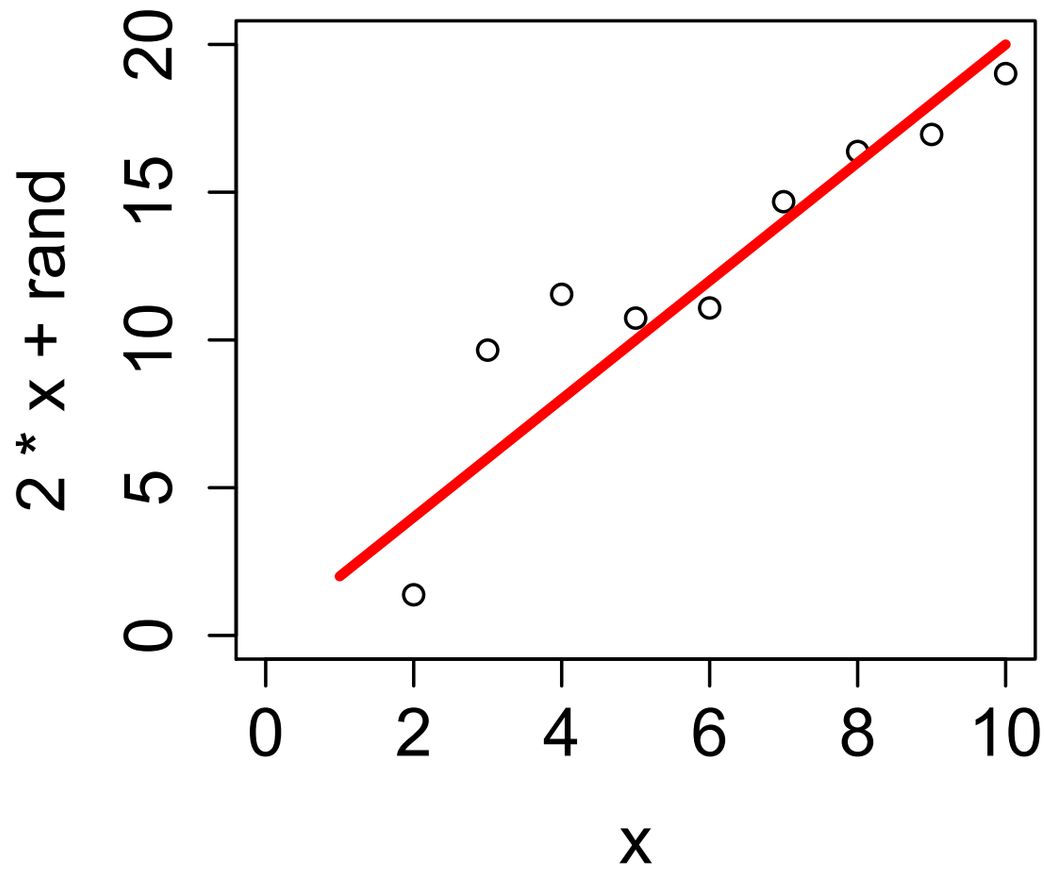





```
par(ps=18, cex=2, mfrow=c(2,2))
x = seq(1,10,length=10)
rand = rnorm(length(x), mean=0, sd=3)
plot(x,2*x,xlim=c(0,10),ylim=c(0,20))          # force given x and y limits
lines(x,2*x,col='red',lwd=3)          # superimpose line over previous plot
plot(x,2*x+rand,xlim=c(0,10),ylim=c(0,20))  # force given x and y limits
lines(x,2*x,col='red',lwd=3)
```



Values of y follow the relation y=2*x **exactly**



Values of y follow the relation y=2*x **inexactly**. In general, the collection of (x,y) form a sample from some population

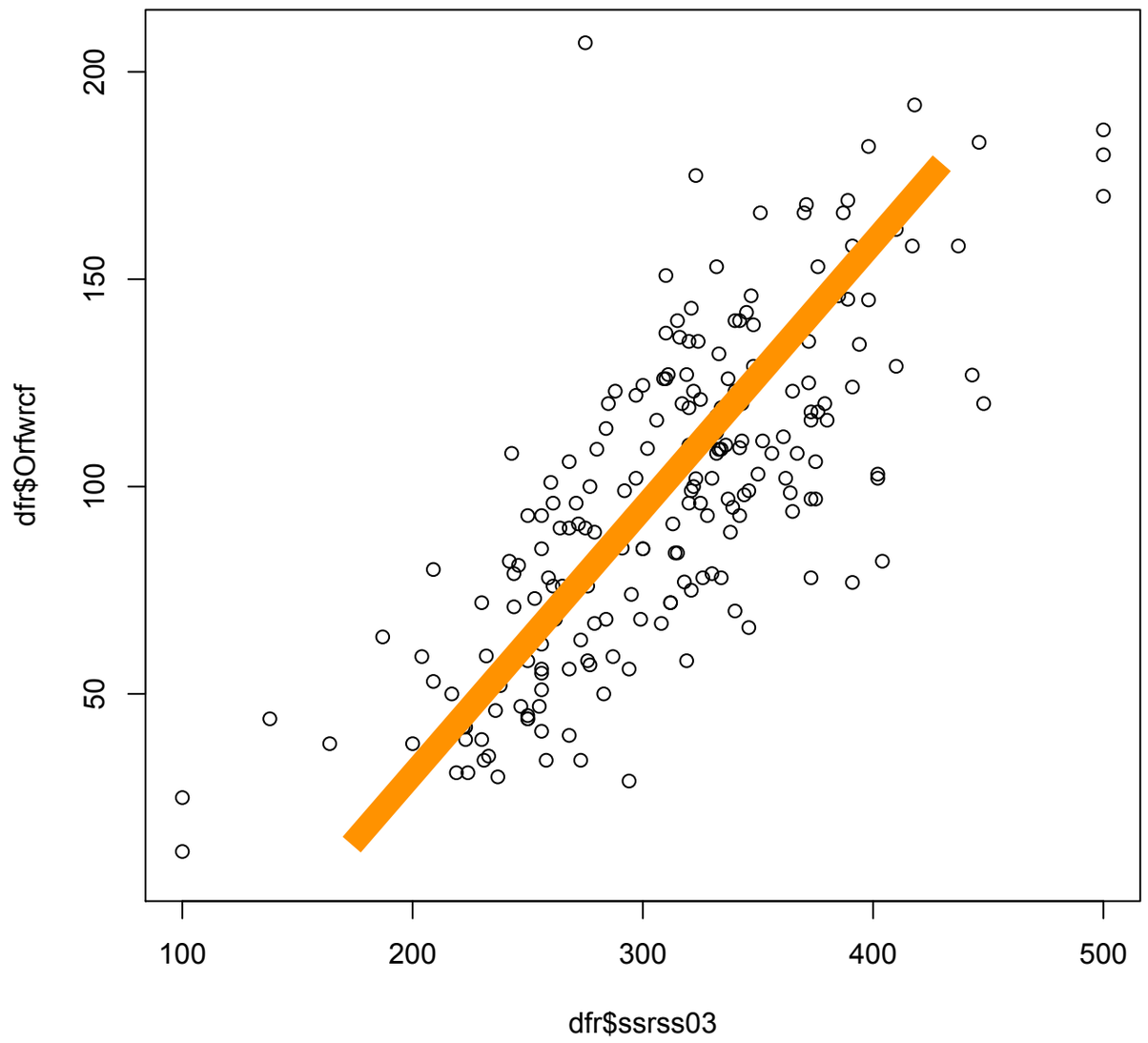
Example

- Consider FCAT datafile
- Let us plot FCAT scores against Reading Fluency of an FCAT passage (there are passages of different types for other tests)

First 3 lines fo FCAT_Mult_grade3.csv

```
ssrss03,iiid___,iiage__,iige___,gortcss,gortfss,orfwrcg,Orfwrcf,orfwrct,tswessa
,tpdessa,tsum_ss,rspatc,lspatc,totalmq,wavoto2,wabdto2,wasito2,wamrto2,wafu
lIQ_,wapeIQ_,waveIQ_,lc1sum,lc2sum,lc3sum

 ,492,8.58904109589041,M,5,6,70,48,74,92,108,100,7,7, ,
20,49,32,51,80,99,67,4,2,1

 ,530,9.39178082191781,M,
8,4,28,33,27,77,85,77,17,6,70,42,40,52,51,93,93,95,3,1,2
```

```
> dfr = read.csv("FCAT_Mult_grade3.csv")

head(a,2)

  ssrss03 iiid___  iiage__ iige___ gortcss gortfss orfwrcg Orfwrcf orfwrct
1      NA     492 8.589041       M       5       6      70      48      74
2      NA     530 9.391781       M       8       4      28      33      27
  tswessa tpdessa tsum_ss rspatc lspatc totalmq wavoto2 wabdto2 wasito2 wamrto2
1      92     108     100      7      7      NA      20      49      32      51
2      77      85      77     17      6      70      42      40      52      51
  wafulIQ_ wapeIQ_ waveIQ_ lc1sum lc2sum lc3sum
1       80      99      67      4      2      1
2       93      93      95      3      1      2
```



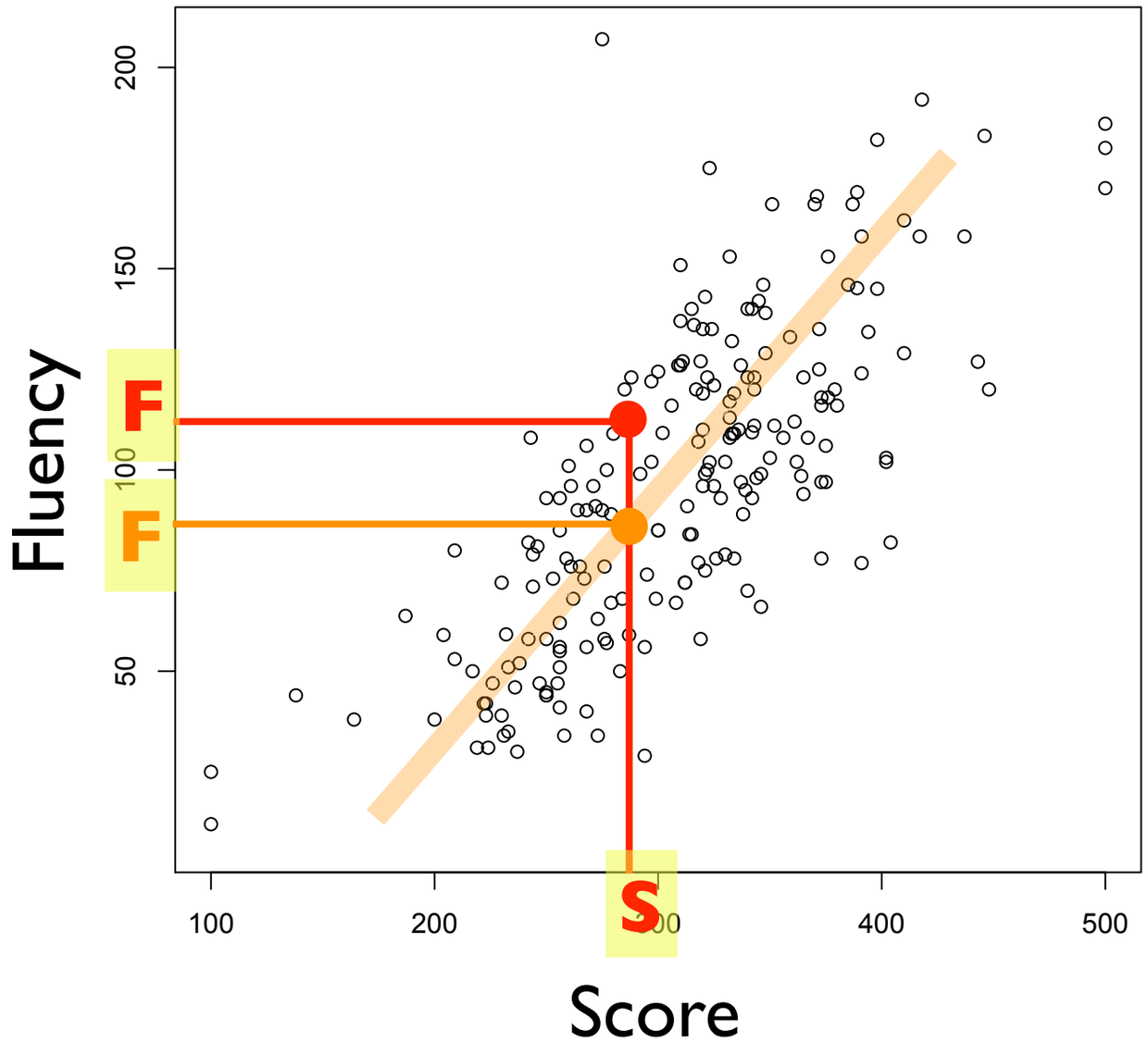
Orange line was added by hand.

The line shows a trend, which is not exact.

Higher degrees of fluency correlates with higher scores .

However, this does NOT imply that higher fluency is the ***cause*** of higher scores.

```
dfr <- read.csv(file="FCAT_Mult_grade3.csv",head=TRUE,sep=",")

plot(dfr$ssrss03, dfr$Orfwrcf)
```



The difference

F - F = ε

is the model error,

called **residual**

Pick a random student. His score **S** and fluency **F** correspond to one point on this graph.

The Fluency **F** would be obtained if all the points lie on the straight line, which serves to model the data

Population Model

$$Y = \alpha + \beta X + \varepsilon$$

α : intercept

β : slope

ε : model error = residual

In R, use a **formula**: Y ~ X

Population Model

$$Y = \alpha + \beta X + \varepsilon$$

- α and β are unknown model parameters
- ε : **model error** which follows some statistical distribution, **assumed to be N(0,σ)**

a and **b** are computed based on sample data

Some Objectives:

- estimate the parameters **α** and **β** by **a** and **b** from sample data
- establish confidence intervals for **a** and **b**
- determine the "goodness" of the fit that models the population

Estimation of α and β

- In practice, we work with a sample of *n* pairs (x_i, y_i)
- Given the line $y = a + b\,x$, compute the residual (error)

$$r_i = y_i - (a + b\,x_i) = \varepsilon_i$$

- Sum the square of the residuals for all *n* points of the sample:

$$\mathrm{SSE} = \sum_{i=1}^{n} r_i^2$$

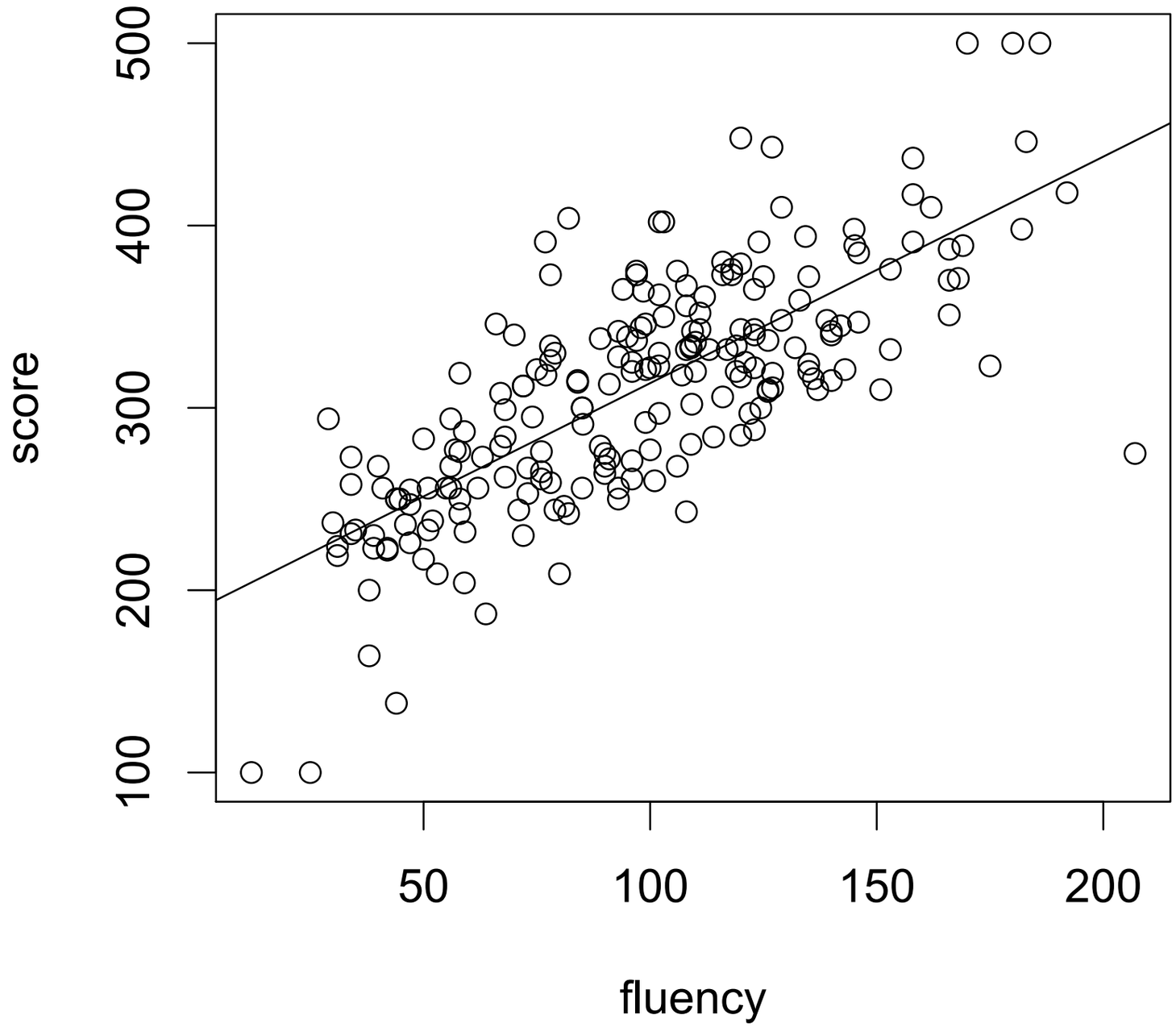
SSE used in ANOVA

- SSE depends on ***a*** and ***b***
- Choose ***a*** and ***b*** to minimize SSE

Sum of Squares

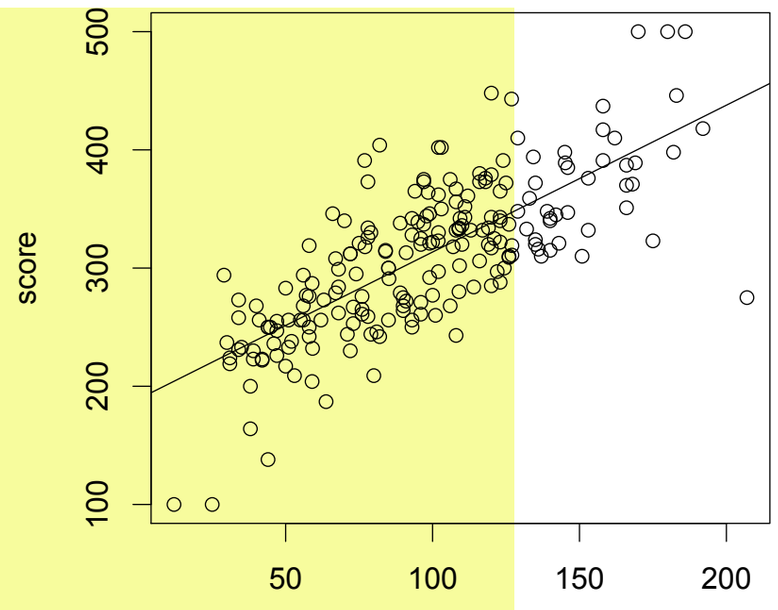
- In regression, one minimizes the sum of squares of residuals to estimate model parameters
- In ANOVA (which operates with factors), sum of squares of the errors (all that is not accounted for by the averages of "groups") is fundamental to estimate the strength of dependencies and factor interactions

Code in first_plot.r



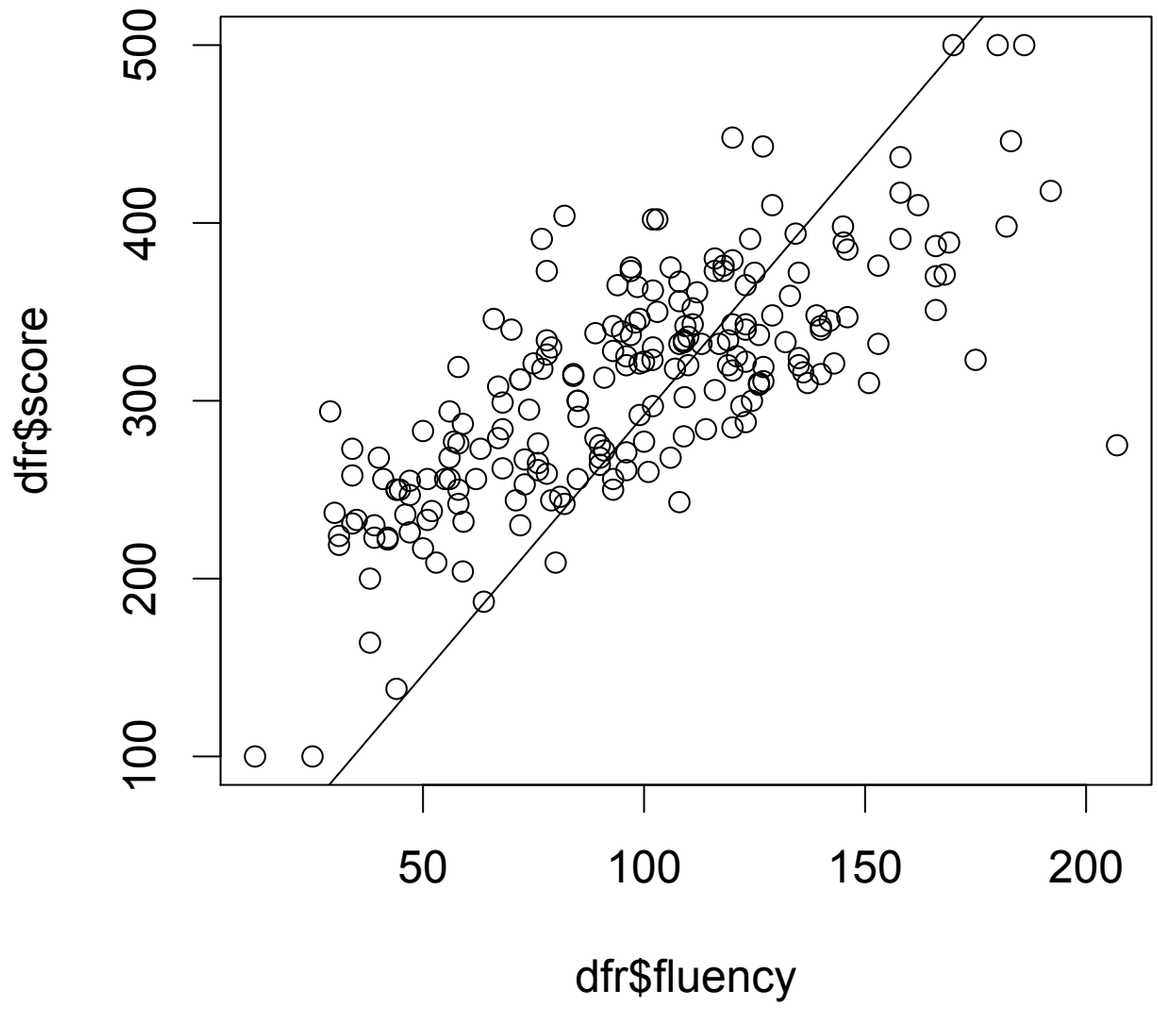
Code for previous slide

```
rm(list=ls())
dfr <- read.csv(file="FCAT_Mult_grade3.csv",
                                head=TRUE,sep=",")
score = dfr$ssrss03
fluency = dfr$Orfwrcf
dfr = data.frame(score, fluency)[!is.na(score),]
llm = lm(score ~ fluency, data=dfr)

par(ps=12, cex=1.5)
plot(dfr$fluency, dfr$score,        # X axis,  Y axis
    xlab="fluency", ylab="score")
abline(llm)                                  # draw regression line
```



Repeat experiment with formula:

score ~ fluency + 0



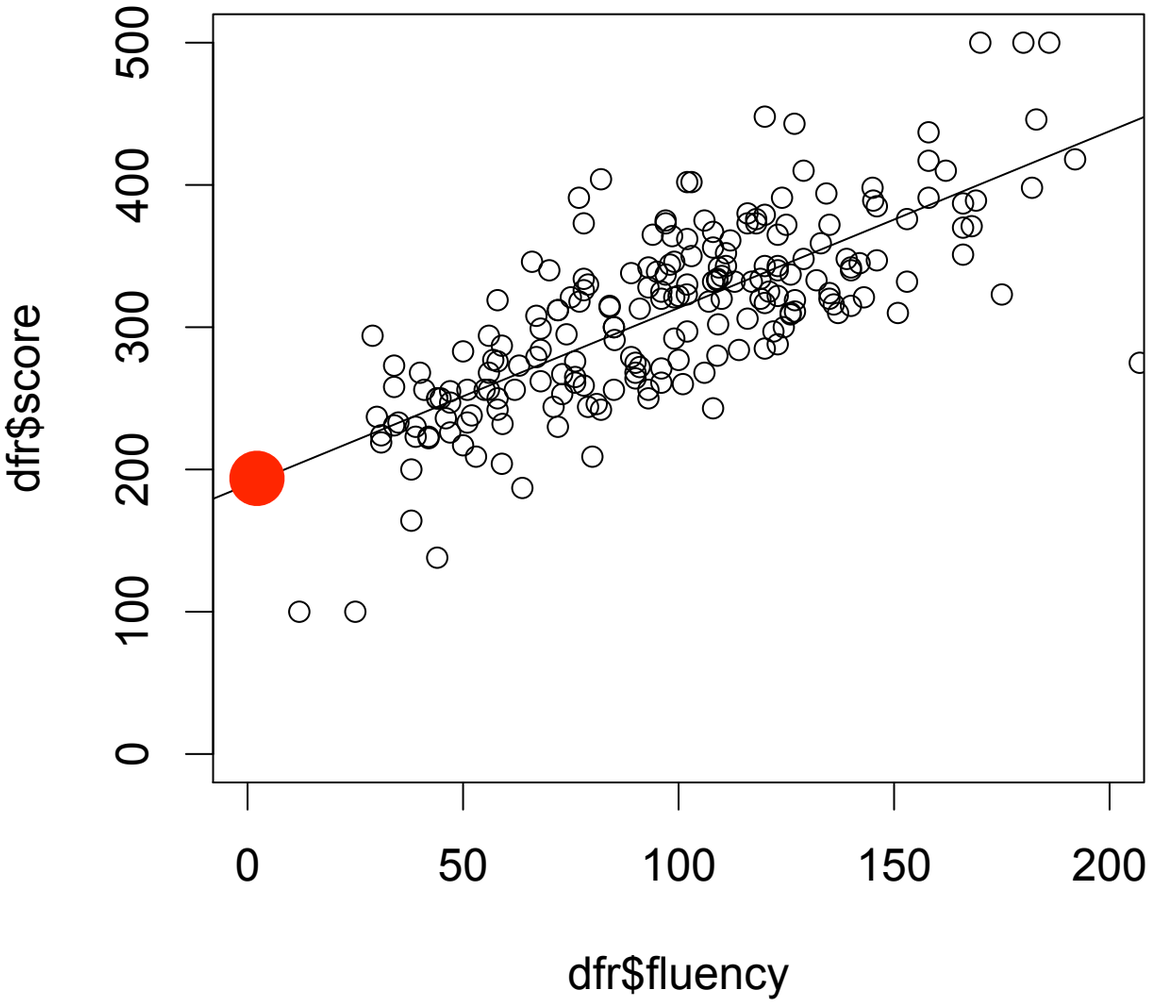
score ~ fluency + 0

The regression line now goes through the origin (0,0)

```
dfr = data.frame(score, fluency)[!is.na(score),]
llm = lm(score ~ fluency + 0, data=dfr)
…
plot(dfr$fluency, dfr$score)
abline(llm)
```

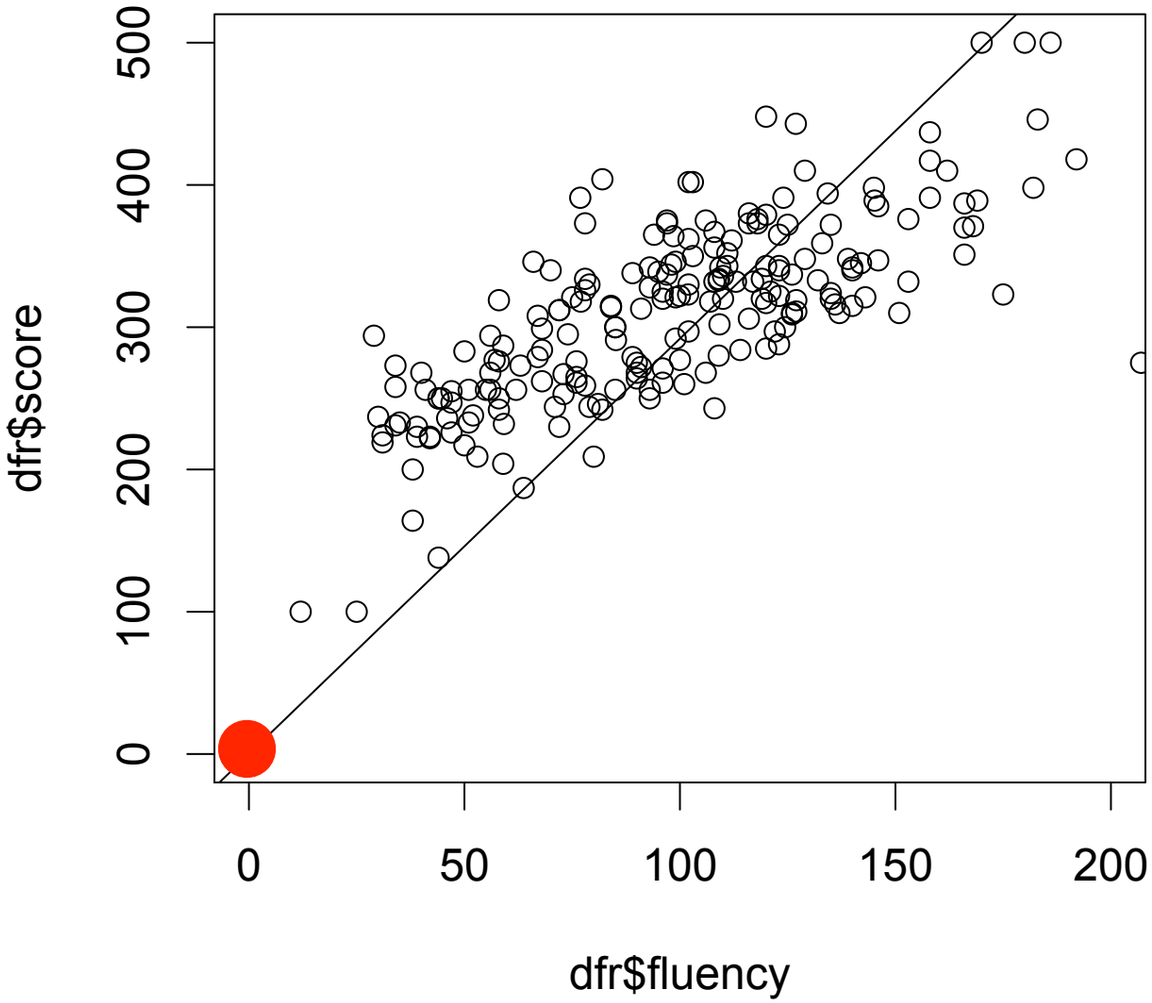
```
# remove NAs
# + 0 : set α = 0
```

Intercept



lm(score ~ fluency, data=dfr)

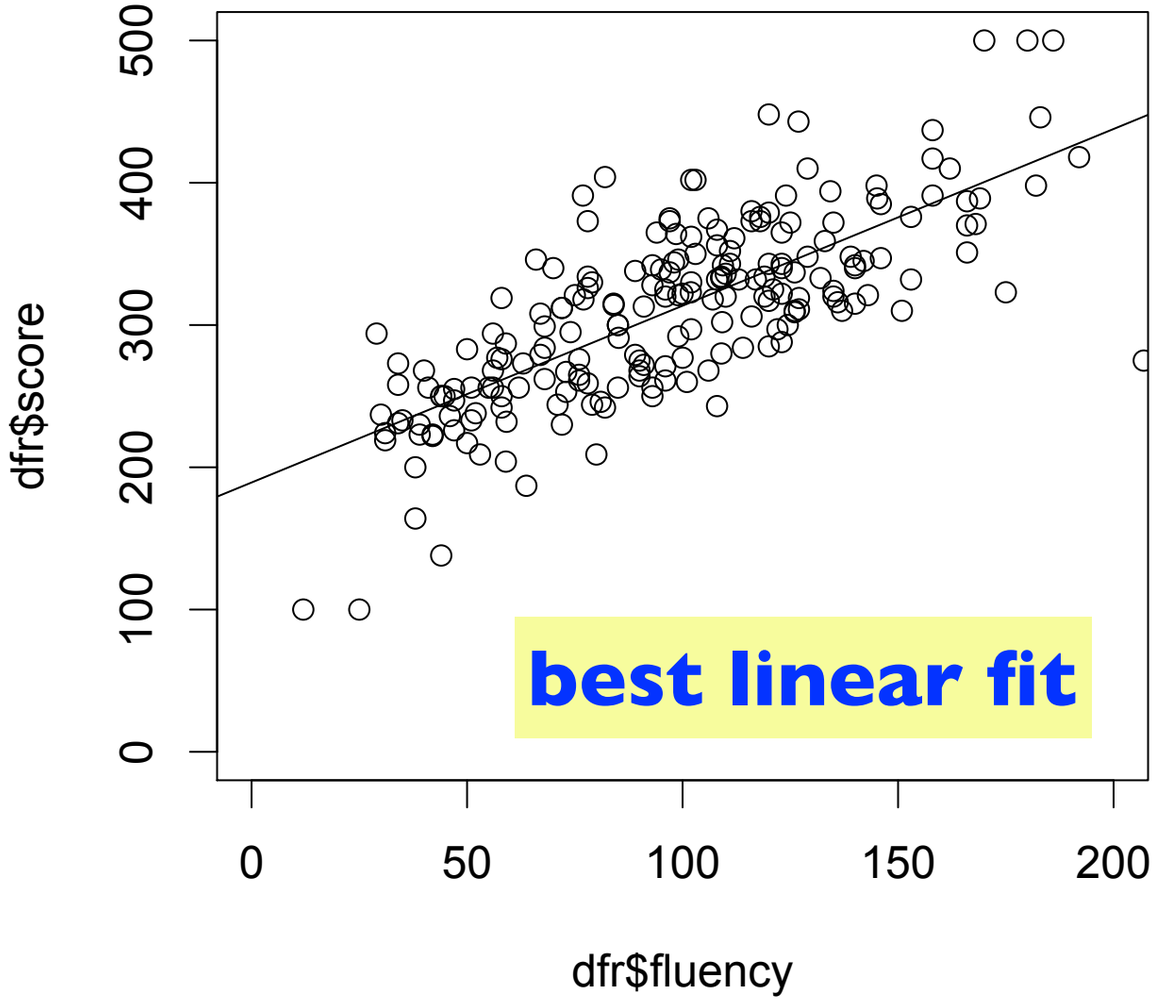
Zero Intercept



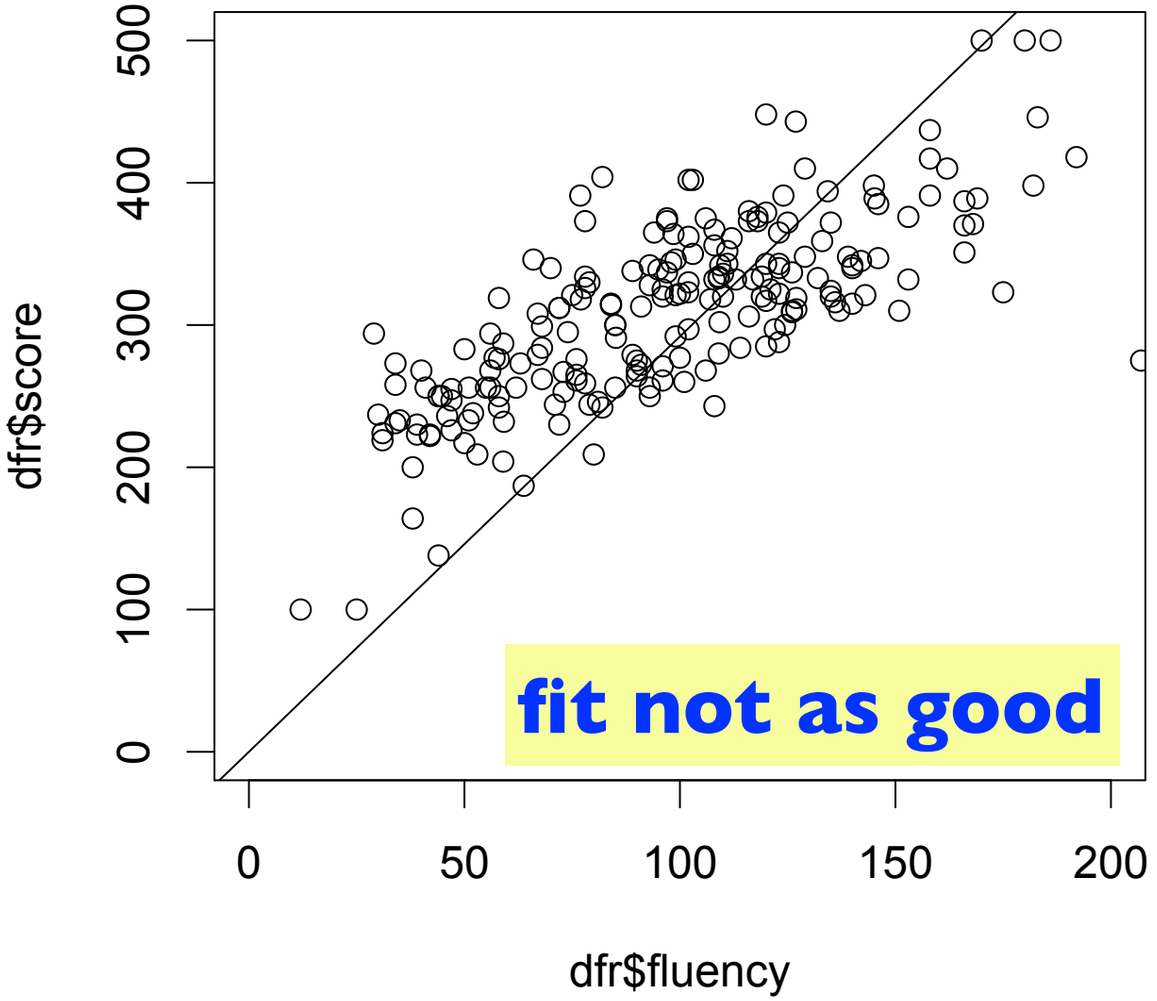
lm(score ~ fluency + 0, data=dfr)

Which line better fits the data?

lm(score ~ fluency, data=dfr)



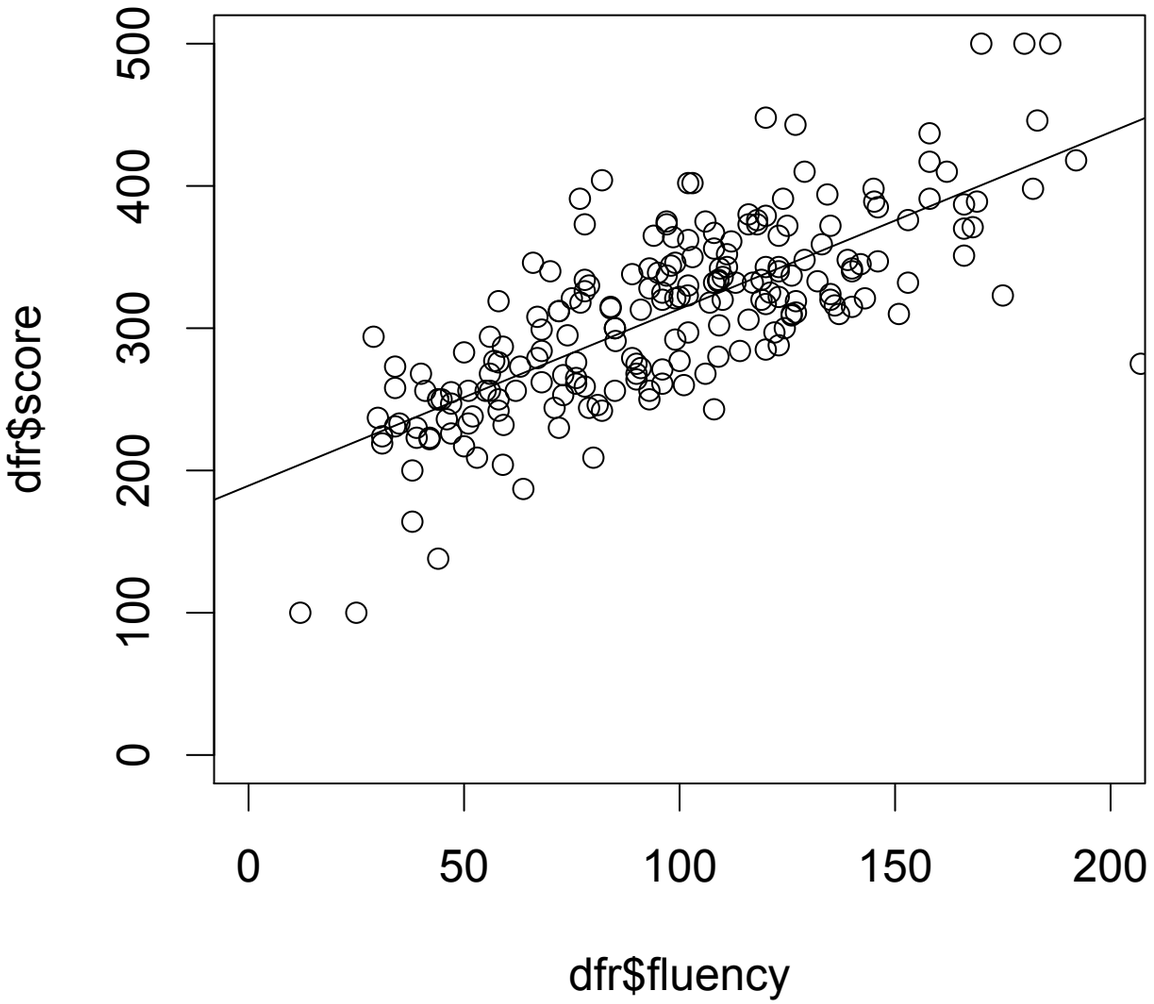
lm(score ~ fluency + 0, data=dfr)



Correlation/Causation

- Higher scores are *correlated* with higher fluency of FCAT passages
- Higher fluency is **not necessarily the *cause*** of higher scores, although it is certainly possible
- Can we quantify this correlation?

Correlation/Causation




```
> cor(dfr$score, dfr$fluency)

0.7484397
```

$$cor(x, y) = \frac{cov(x, y)}{sd(x)sd(y)}$$

R: ?lm(. . .)

Description:

'lm' is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although 'aov' may provide a more convenient interface for these).

Usage:

```
lm(formula, data, subset, weights, na.action,
    method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
    singular.ok = TRUE, contrasts = NULL, offset, ...)
```

Population vs. sample

- Consider sample S taken from population P
- Each element in sample is (x_i,y_i), $i=1..n$
- We wish to fit a line to the data:
  - $y = ax + b$ + residual
- Use lm(…) to estimate a and b
- Choose another sample, and fit another line. One gets a new a and b
- So a and b are random variables, and they have a mean value and a **s.d.**

Confidence Intervals

- From previous slide:
- ***a*** and ***b*** are random variables, and they have a mean value and a **s.d.**

- *a* and *b* have their own distributions with mean and standard deviation
- **Establish 95% confidence intervals for α and β for linear population model**

Best Estimate

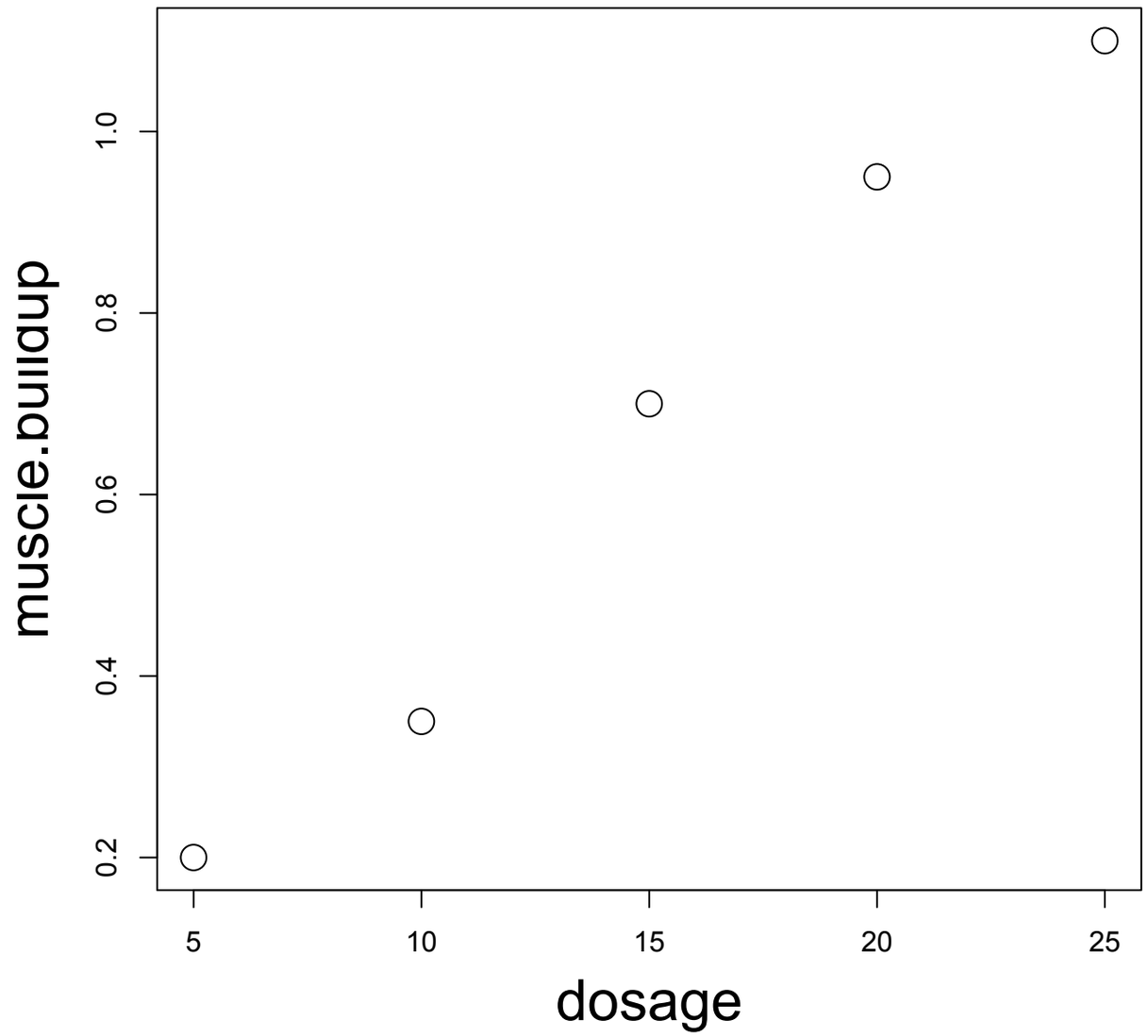
- Assume that we have a pair of variables
  - (Muscle Strength Index (MSI) versus Drug Dosage)
  - using a single number, what is the best approximation of the MSI?
  - ANSWER: **mean(MSI)**
- A linear relationship suggests additional assumptions: that of a linear dependence of MSI as a function of drug dosage

Experiment

- Three steroid dosages: 5, 10, 15, 20, 25 mg
- Measure muscle strength increase
- Objective: model muscle strength increase in the population of people between ages of 15 and 30, as a function of steroid dosage
- Hypothesis: as dosage increases, muscle strength also increases

Sample

- Take a sample of 5 adults
- Of course, zero drug dosage increase leads to zero muscle tone increase so that intercept should be zero



It looks like these points could be "almost" placed on a straight line.

Use lm() to compute this line

```
dosage = c(5,10,15,20,25)
muscle.buildup = c(.2,.35,.7,.95,1.1)
plot(dosage,muscle.buildup,type='p',cex=2, cex.lab=2)
```

```
> lm(muscle.buildup ~ dosage)

Call:
lm(formula = muscle.buildup ~ dosage)

Coefficients:
(Intercept)       dosage
     -0.060        0.048
```

muscle.buildup = intercept + 0.048 * dosage

+ residual error

Consider a dosage of 10

0.35 = -0.06 + 0.048 * 10 + **residual** # **error**

residual = 0.35 + 0.06 - 0.48 =

= -0.07

Also try:

lm (muscle.buildup ~ dosage + 0)

So far ...

The line

Y = intercept + 0.048 * dosage

provides an approximation to the data collected.

We have created a best fit to the sample data.

Another sample

- Let us choose 5 more people from our population and administer the test.
- The data now looks like
  - see next slide

Dosage	Sample 1	Sample 2
5	0.2	0.15
10	0.35	0.38
15	0.7	0.6
20	0.95	1.0
25	1.1	1.2

```
> llm = lm(muscle.buildup ~ dosage)
> names(llm)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

These different names refer to functions, as we will see.

For example:

llm$coefficients

llm$residuals

llm$effects …

```
> class(lm)
lm
>> class(llm[1])
[1] "list"
```

What do these names mean?

str(lm)

What is the model?

```
> llm$model
```

	muscle.buildup	dosage
1	0.20	5
2	0.35	10
3	0.70	15
4	0.95	20
5	1.10	25

Original data on which the regression is based

Residuals

```
> llm$residuals

    1     2     3     4     5

 0.02 -0.07  0.04  0.05 -0.04
```

Error made in the linear approximation. One can check that the mean of the residuals is zero.

```
> d = data.frame(llm$model, Y=-0.06+0.048*llm$model[[2]],
      error=llm$residuals)
> d
```

	muscle.buildup	dosage	Y	error
1	0.20	5	0.18	0.02
2	0.35	10	0.42	-0.07
3	0.70	15	0.66	0.04
4	0.95	20	0.90	0.05
5	1.10	25	1.14	-0.04

```
> sum(d$error)
[1] 3.469447e-18
```

General property of linear regression: errors average to zero

Note: the error is the difference between the original data (muscle.buildup), and the result predicted from the linear model (Y)

error =muscle.buildup - (-.06+0.048*dosage)

```
> llm$coefficients
(Intercept)      dosage
    -0.060       0.048
```

Intercept and **slope**

```
> llm$fitted.values
    1     2     3     4     5
0.18  0.42  0.66  0.90  1.14
```

values predicted by the linear model

```
> llm$residuals
     1      2     3     4      5
 0.02  -0.07  0.04  0.05  -0.04
```

```
> muscle.buildup - llm$fitted.values
    1     2     3     4     5
 0.02  -0.07  0.04  0.05  -0.04
```

 # named vector

Result is a named vector

(see next slide)

```
> str(muscle.buildup)
 num [1:5] 0.2 0.35 0.7 0.95 1.1

> str(llm$fitted.values)
 Named num [1:5] 0.18 0.42 0.66 0.9 1.14
-  attr(*, "names")= chr [1:5] "1" "2" "3" "4" …
```

```
#Get rid of names

names(llm$fitted.values) <- NULL

> str(llm$fitted.values)

(NOT POSSIBLE. llm$fitted.value
    NO LONGER EXISTS
```

-> is an alternative to =

```
>     a = 3
       a <- 3
```

Can also use ->

```
>     3  ->  a
```

Return to sample 2

- How to compute the fitted value based on sample 2?
- muscle.strength from sample 2 =
  - 0.06 + 0.048 * sample2

Property of Error

```
> e = muscle.buildup - (-.06+.048*dosage)
> sum(e)
[1] 2.775558e-16
```

Zero sum when using the sample on which the line is based

```
> e = sample2 - (-.06+0.048*dosage)
> sum(e)
[1] 0.03
```

Non-zero sum when using any other sample

dosage	muscle.strength
5	0.2
10	0.35
15	0.7
20	0.95
25	1.1

Choose some arbitrary intercept a and intercept b

muscle.strength[i] = a + b * dosage[i] + error[i] (= sample value)

Calculate the variance of the error:

var(error) = error[1]^2 + error[2]^2 + error[3]^2
+ error[4]^2 + error[5]^2

Calculate the sum of squares of the error:

total.error = error[1]^2 + error[2]^2 + error[3]^2
+ error[4]^2 + error[5]^2
= error^2

Change the intercept **a** and slope **b,** which leads to a different var(error). The values chosen by the **lm(...)** function lead to the minimum possible value of **total.error**

This calculation is performed by lm(….)

Population

- Given a best linear fit for a single sample
  - we wish to derive information for a population
  - since the calculated line statistics (slope and intercept) change from sample to sample, we would like to establish confidence intervals
- Choose 100 samples; compute **a** and **b** for each sample. The question is: compute intervals for **α** and **β** be such that they are contained within the computed intervals 95% of the time on average (with rejection of 5%)

Confidence Intervals

- Use the **summary()** function on the output to lm(…)

> **summary(llm)**

Call:

lm(formula = muscle.buildup ~ dosage)

Residuals:

1	2	3	4	5
0.02	-0.07	0.04	0.05	-0.04

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-0.06000	0.06351	-0.945	0.41448	
dosage	0.04800	0.00383	12.534	0.00109	**

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06055 on 3 degrees of freedom

Multiple R-squared: 0.9813, **Adjusted R-squared: 0.975**

F-statistic: 157.1 on 1 and **3 DF**, **p-value: 0.001095**

Confidence Intervals

- Use the command **predict()**
- **?predict**

predict()

- Given the original X-values, calculate predicted Y along with confidence intervals

```
llm = lm(muscle.buildup ~ dosage)

p = predict(llm, interval="confidence")
```

```
> p
      fit        lwr        upr
1 0.18   0.03073004 0.3292700
2 0.42   0.31445020 0.5255498
3 0.66   0.57381895 0.7461811
4 0.90   0.79445020 1.0055498
```

0.031 < 0.18 < 0.33

with 95% confidence

```
> predict(<tab>
...=                interval=          newdata=          prediction.interval=  terms=
deriv=               level=            newxreg=           scale=               type=
df=                 n.ahead=           object=           se.fit=              weights=
dispersion=          na.action=         pred.var=          se=                  x=
> predict(
```

predict

- Read help file
  - ?predict
  - ?predict.lm
    - (arguments specific to lm() objects)

?predict

?predict ?predict.arima0 ?predict.nls

?predict.Arima ?predict.glm ?predict.poly

?predict.HoltWinters **?predict.lm** ?predict.prcomp

?predict.StructTS ?predict.loess ?predict.princomp

?predict.ar ?predict.mlm ?predict.smooth.spline

Predict offers an easy way to apply a model to new predictor values.

example(predict.lm) for some test cases

Possible Commands for next lab

- **transform(...)** : transform one data.frame into another
- **cat(...)** : print out a list of characters, integers, floats
- **length(...)** : length of a vector/list
- **subset(...)** : select specific data.frame rows
- **with(...)** : avoid typing name of data.frame
- **points(...), lines(...)**
- **lm(...)** : linear model
- **fligner.test(...), var.test(...)** : test of variance
- **confint(lm.object)** : confidence intervals

Next lesson/lab

- Another lesson on regression with example problems
- We will start work with a image manipulation program
  - JImage in class (week of Feb. 13)
- Next lab (week of Feb. 20), will use JImage.