CS 696 Intro to Big Data: Tools and Methods
Fall Semester, 2016
Doc 11 Regression
Oct 4, 2016

Machine Learning

Supervised

Unsupervised

Reinforcement learning

Classification

Regression

Clustering

Density Estimation

Dimensionality Reduction

Supervised learning

Artificial neural network
Bayesian statistics
Bayesian network
Gaussian process regression
Inductive logic programming
Learning Vector Quantization
Logistic Model Tree
Nearest Neighbor Algorithm
Random Forests
Ordinal classification
ANOVA
Linear classifiers
Fisher's linear discriminant
Linear regression
Logistic regression
Multinomial logistic regression
Naive Bayes classifier

Quadratic classifiers
k-nearest neighbor
Boosting
Decision trees
Random forests
Bayesian networks
Naive Bayes
Hidden Markov models

Unsupervised learning

Expectation-maximization algorithm
Vector Quantization
Generative topographic map
Information bottleneck method
Artificial neural networks

Hierarchical clustering
- Single-linkage clustering
- Conceptual clustering
- Cluster analysis[edit]
- K-means algorithm
- Fuzzy clustering
- DBSCAN
- OPTICS algorithm

Outlier Detection
- Local Outlier Factor

Other

Reinforcement learning
- Temporal difference learning
- Q-learning
- Learning Automata
- SARSA

Deep learning
- Deep belief networks
- Deep Boltzmann machines
- Deep Convolutional neural networks
- Deep Recurrent neural networks
- Hierarchical temporal memory

Machine Learning & Patterns

Machine learning algorithms

- Detect patterns
- Generate models based on those patterns

Feed a neural network pictures of cats

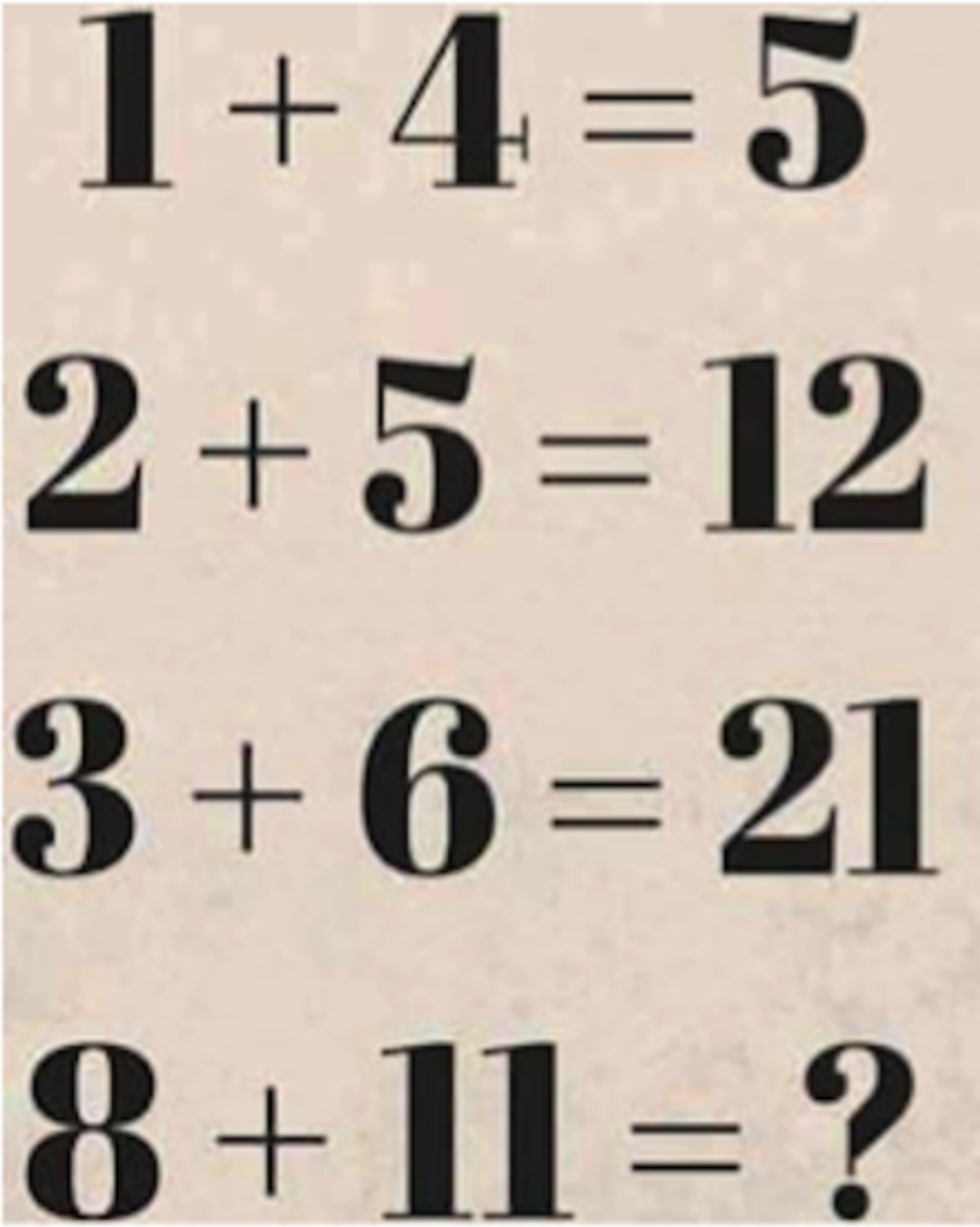
- Neural net can identify cats
- Can automate finding cat photo on internet

Drive a car with neural network “watching”

- You actions
- Videos of surroundings

Neural net can identify patterns & start to drive

Limits of Pattern Matching



1 * (4 + 1) = 5

2 * (5 + 1) = 12

3 * (6 + 1) = 21

8 * (11 + 1) = 96

0 + 1 + 4 = 5

5 + 2 + 5 = 12

12 + 3 + 6 = 21

21 + 8 + 11 = 40

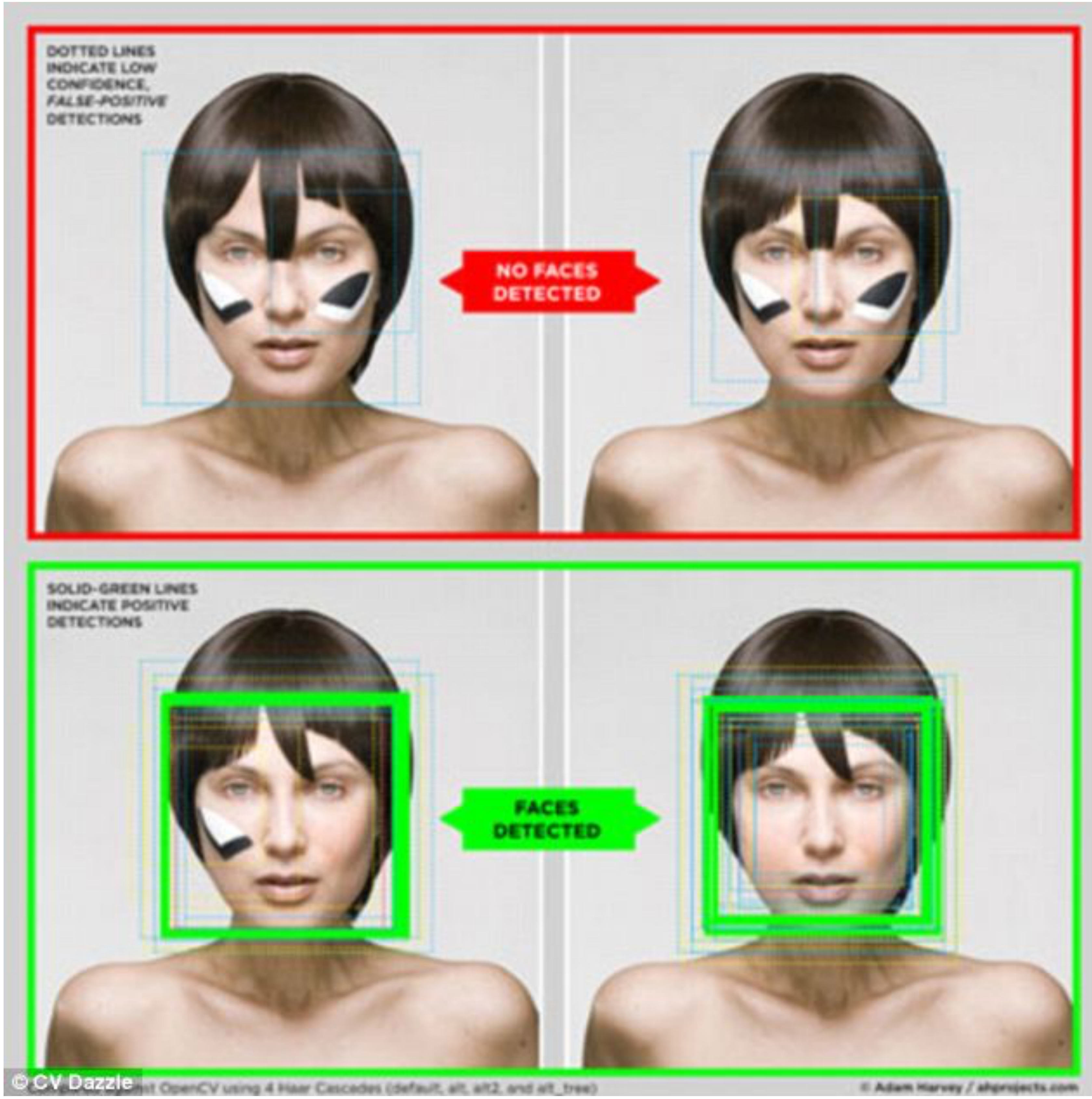
http://nautil.us/blog/the-fundamental-limits-of-machine-learning

No Free Lunch Theorems

David Wolpert

For every pattern a machine learning algorithm is good at learning, there's another pattern that same learner would be terrible at picking up

No Free Lunch



Models

Machine Learning algorithms produce models

Models allow predictions or offer insights

Examples

Decreasing latency by X increases Amazon's daily revenue by Y

White males without college degrees favor Trump by X%
Females favor Clinton by Y%
...

Models Approximate Reality

World is flat

World is a sphere

World is an oblate ellipsoid

Does the model provide useful predictions/insights

Under what condidtions is the model useful

What are the estimates of the model’s error

Multiple Factors in Model

Amazon's daily revenue depends on

- Latency
- Price
- Steps needed to order
- Page layout
- Relevant suggestions
- Search results
- Font sizes
- Color
- Shipping costs

Some factors will be more important

Stochastic in nature

Independent variables

Regression

Regression

Measure of relation between mean of one variable (dependent) on

one or more other variables (independent)

In chapter 11 of Julia for Data Science

Download the Jupyter notebook before reading

https://technicspub.com/analytics/
https://app.box.com/v/codefiles

Overview

Linear regression

Multiple linear regression

Generalized linear regression (model)

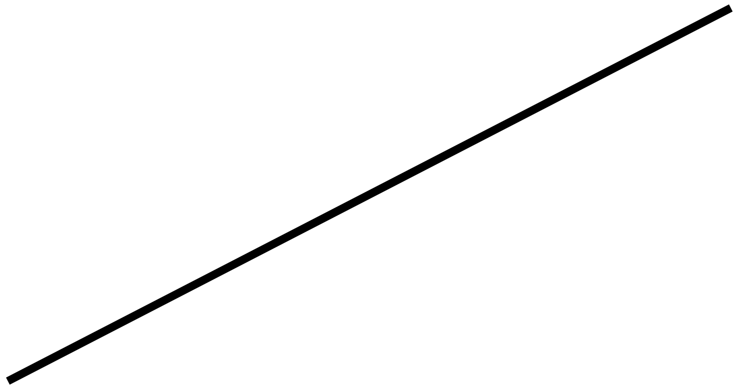
Is the dependent variable related to the independent variable

Generating the model

Error in the model

Effect of independent variables

Linear Regression



Model

$f(x) = 2x + 3$

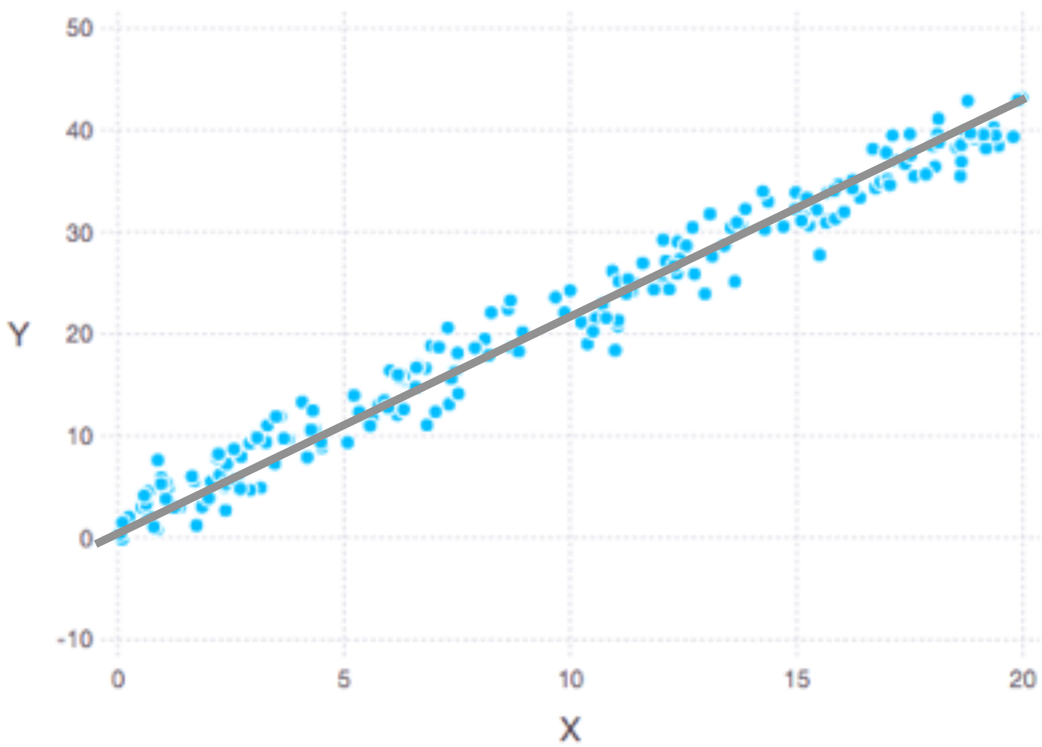
$y = 2x + 3$

$y = 2x + 3$

Dependent Variable

Independent Variable

Linear Regression



Actual relation (assumed)

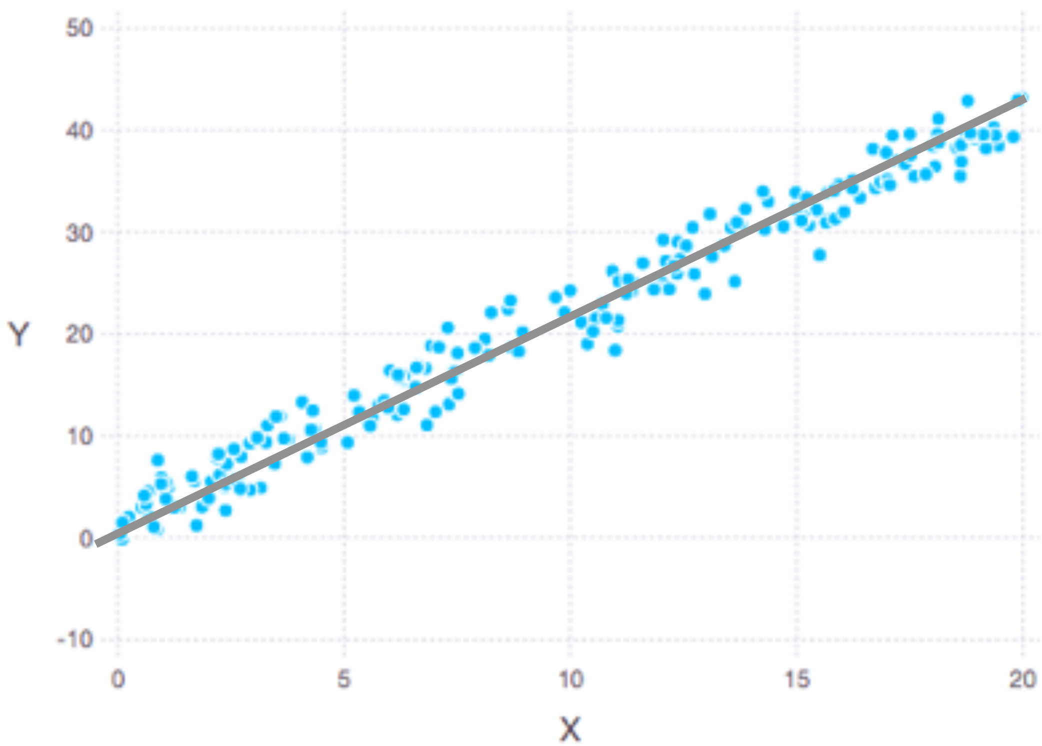
$y = a + bx$

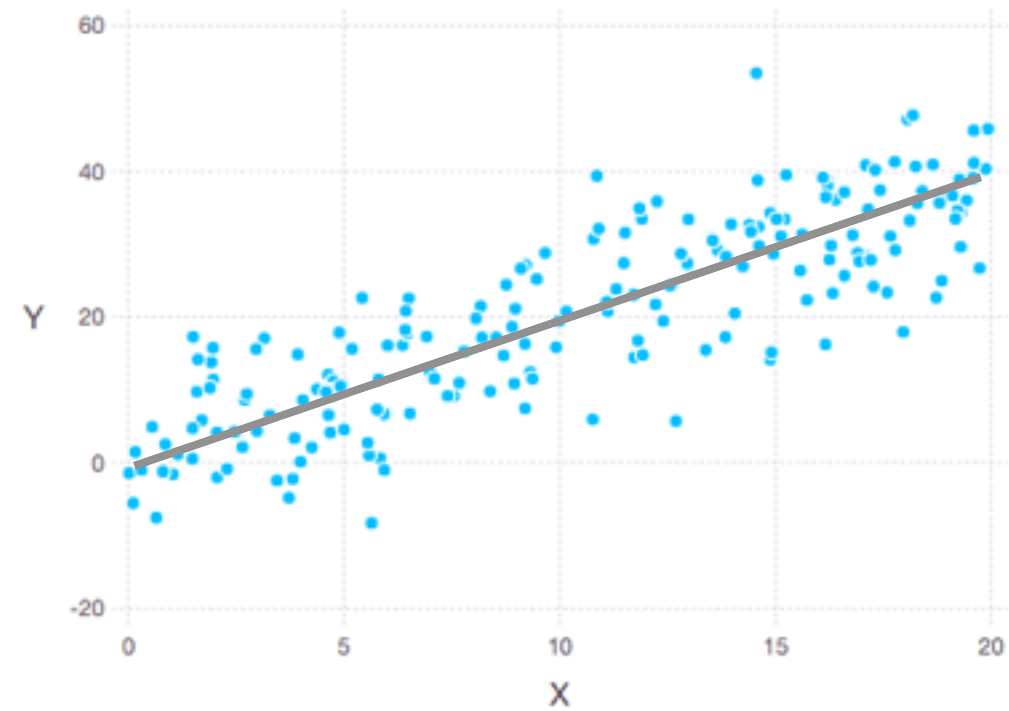
Compute linear line that fits the data best

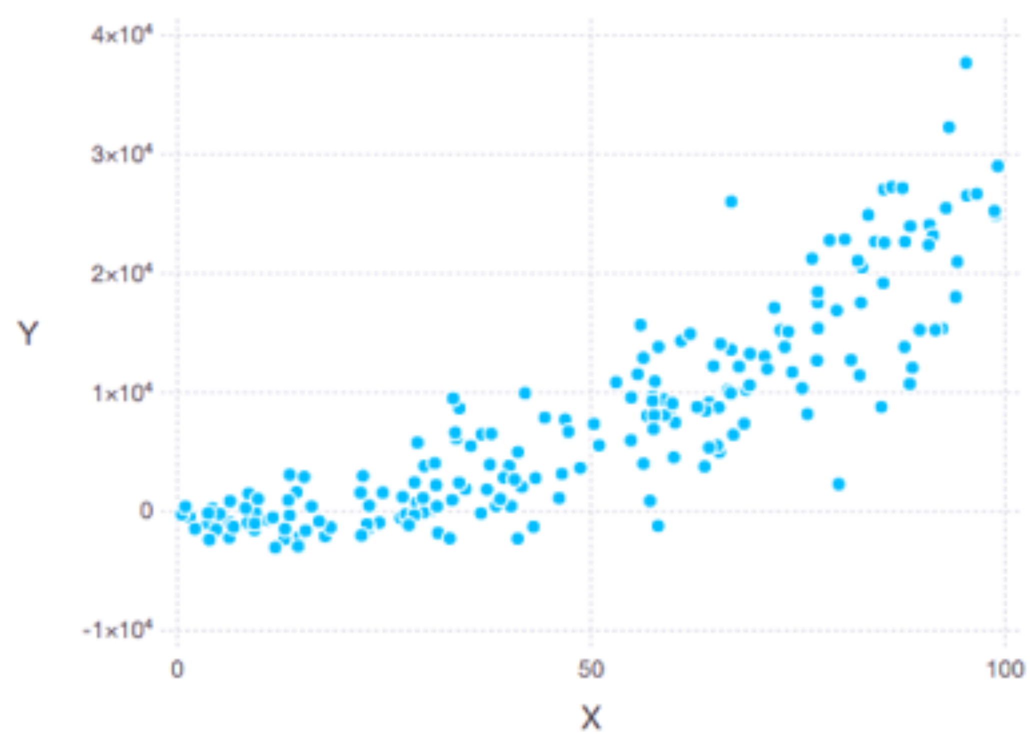
$\hat{y} = a + bx + e$

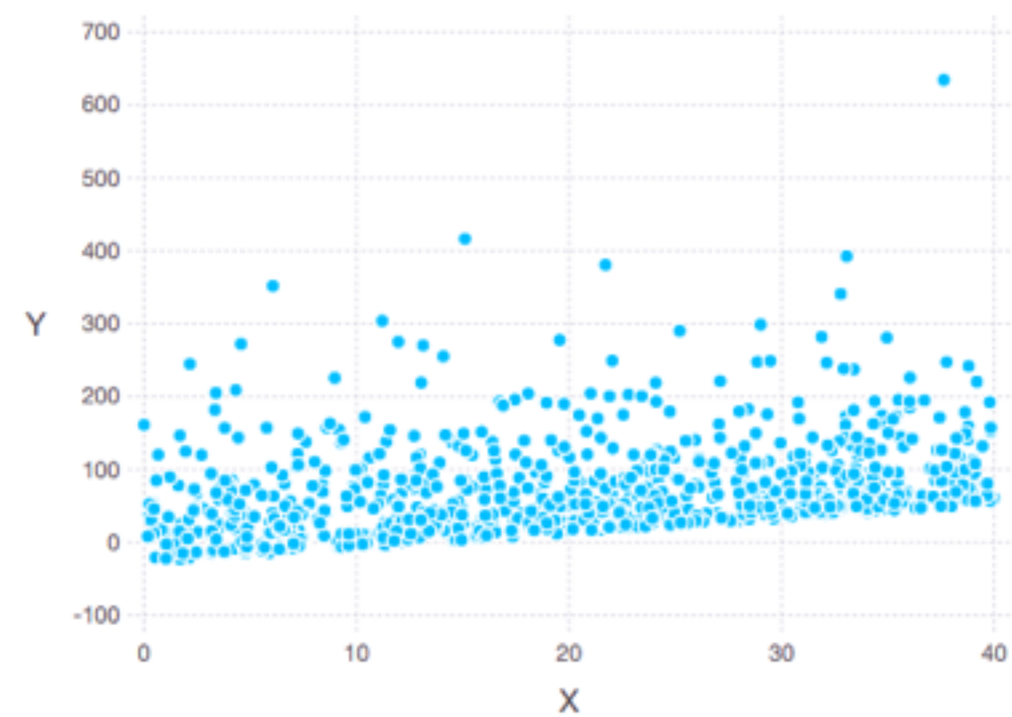
e - error or residual

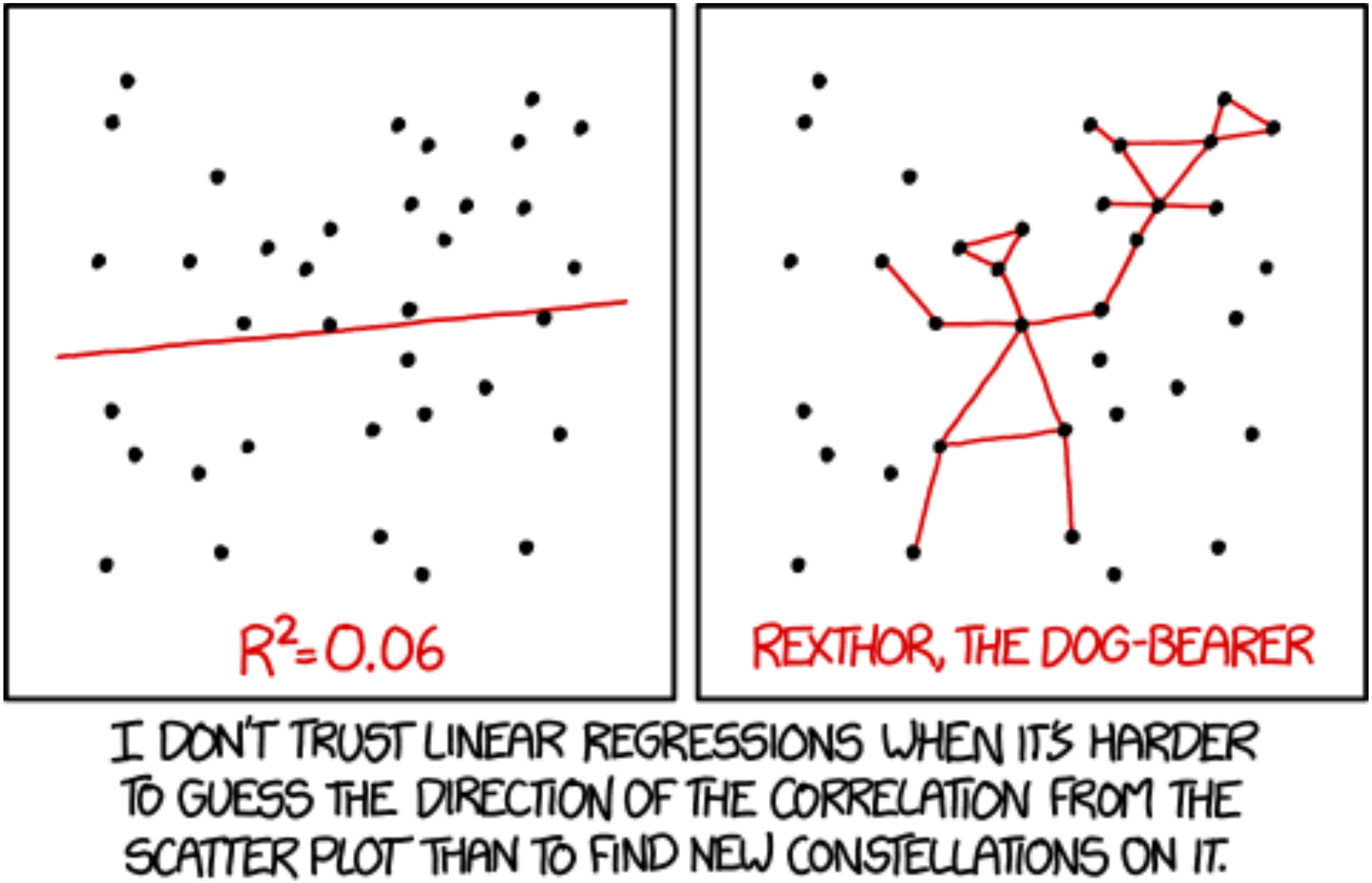
Goal is to minimize residual overall


Y
X
50
40
30
20
10
0
-10
0
5
10
15
20


Y
X
60
40
20
0
-20
0
5
10
15
20


Y
X
4×10⁴
3×10⁴
2×10⁴
1×10⁴
0
-1×10⁴
0
50
100

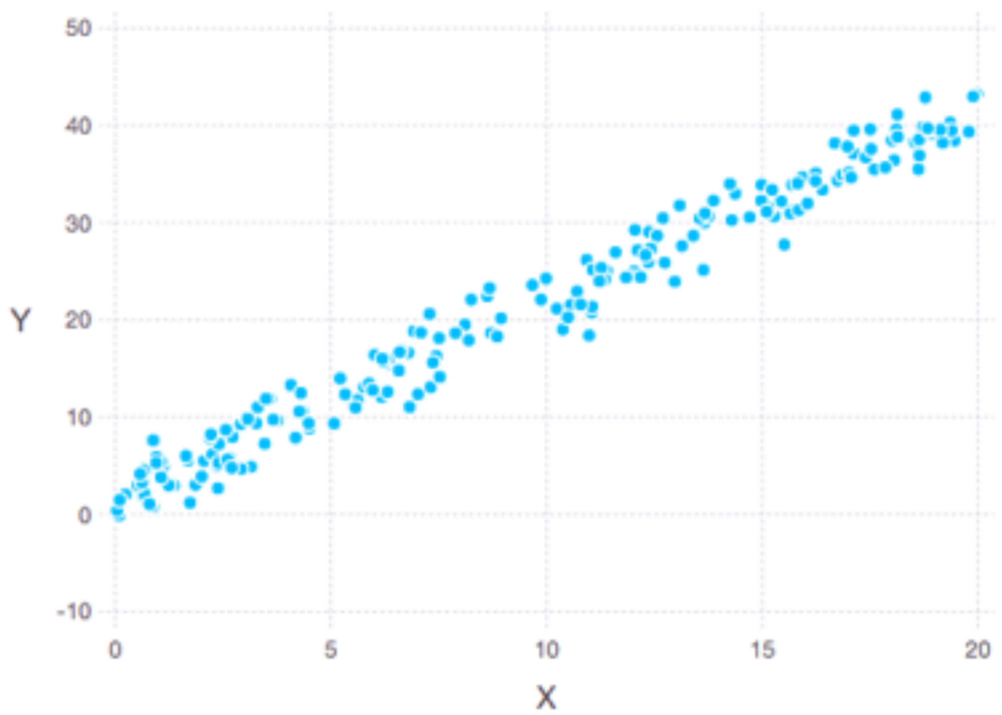

Y
X
700
600
500
400
300
200
100
0
-100
0
10
20
30
40

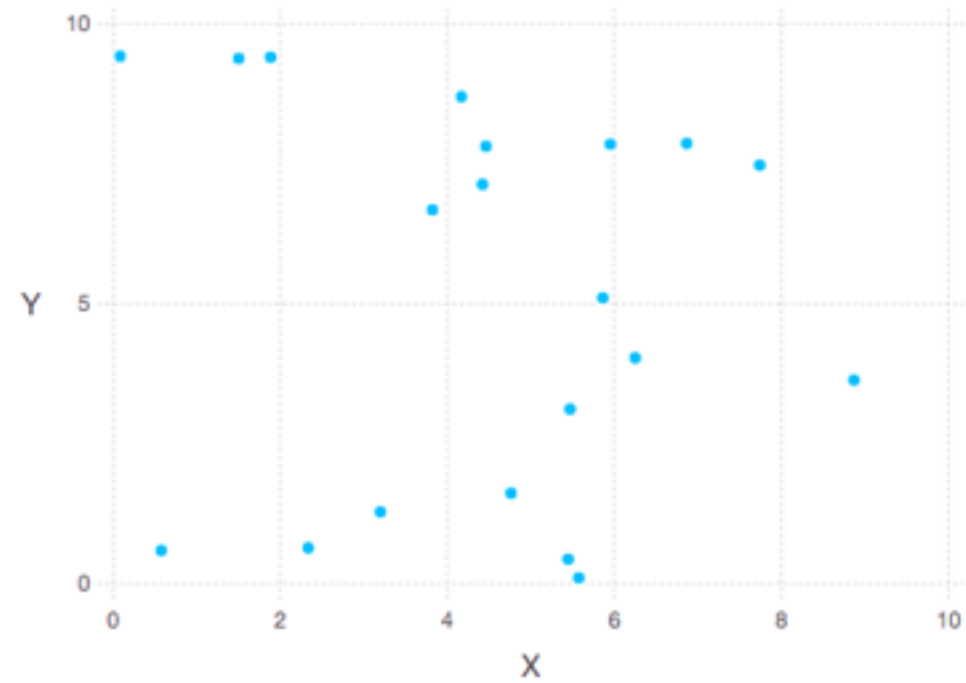


Tuesday, October 4, 16

http://xkcd.com/1725/

Are They Related?





Covariance

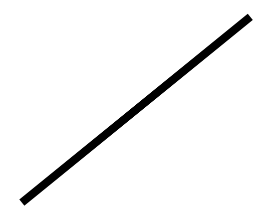
If x & y are related then they should vary from their means in a similar way

$$dx_i = x_i - \overline{x}$$

Values near zero indicate no relation

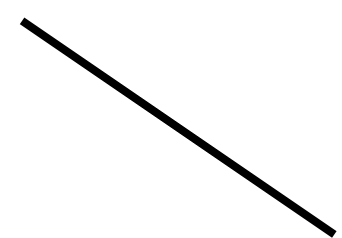
$$dy_i = y_i - \overline{y}$$

positive values - positive relation



$$\mathrm{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n} dx_i dy_i$$

negative values - negative relation



In Julia use function

cov

Effects of Scale

Cost USD	Pounds	Grams
9	3	1357.8
24	7	3168.2
38	10	4526.0

1 Pound = 452.6 grams

Changing the scale of units
- Does not change the relationship
- Does change magnitude of Covariance

- Makes covariance hard to evaluate

cov(pounds,Cost USD) == 50.8

cov(grams, Cost USD) == 23007

cov(grams, Cost INR) == 1,528,308.996

Units

$$dx_i = x_i - \overline{x}$$ Lbs

$$dy_i = y_i - \overline{y}$$ USD

$$\mathrm{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n} dx_i dy_i$$

cov(pounds,Cost USD) == 50.8 lbs*USD

cov(grams, Cost USD) == 23007 grams*USD

Cost USD	Pounds	Grams
9	3	1357.8
24	7	3168.2
38	10	4526.0

Normalizing Data

Convert data to a common scale

Example - divide by maximum value

Cost USD	Pounds	Grams
9	3	1357.8
24	7	3168.2
38	10	4526.0

Cost	Amount
0.237	0.3
0.632	0.7
1.00	1

cov(Cost,Amount) == 0.134 (unitless)

Pearson's Correlation - r

$$r = \frac{\text{cov}(X,Y)}{\sigma_x \sigma_y}$$

Julia function

cor

Normalized Covariance

Unitless

Range -1 to 1

1 = maximumly related

-1 - maximumly inversely related

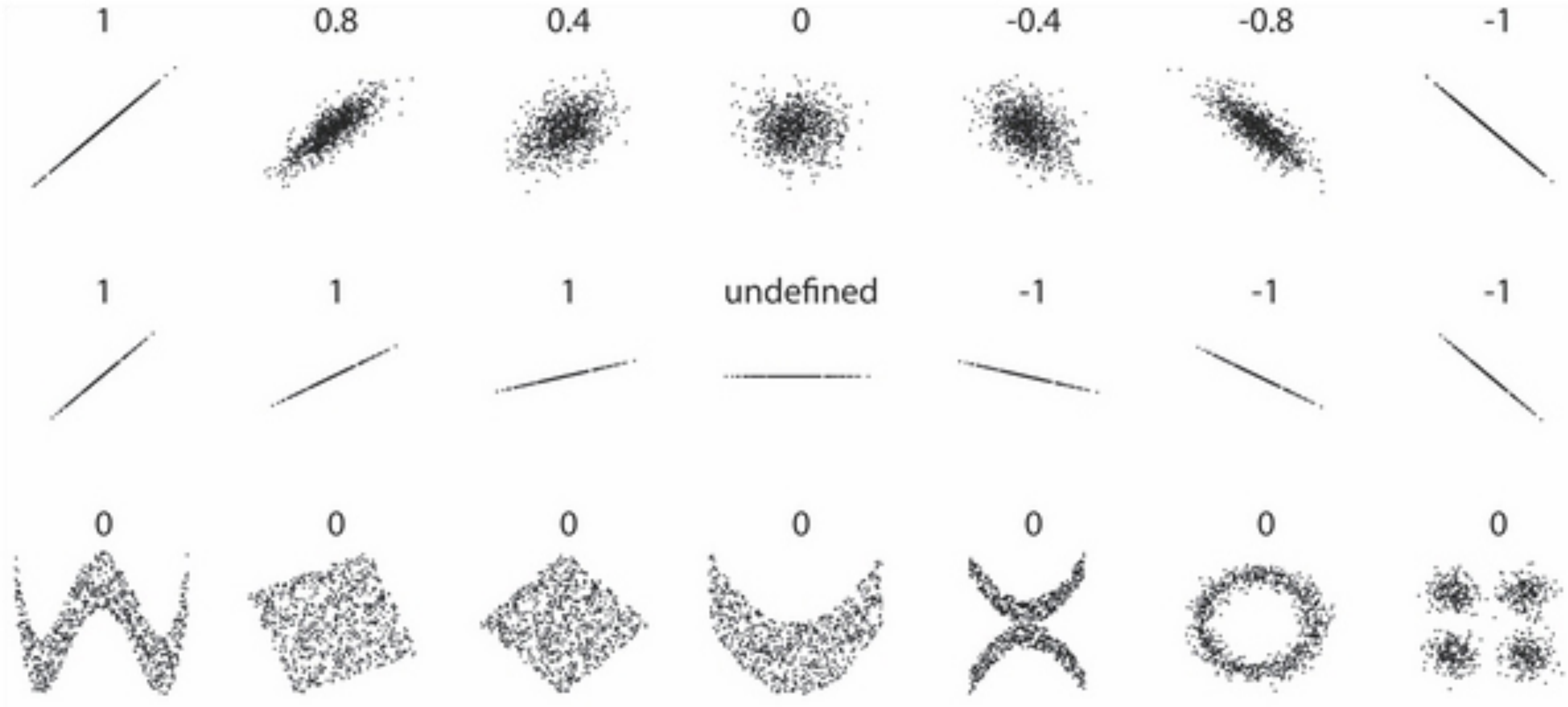
0 - not related

Pearson's Correlation - r

Cost USD	Pounds	Grams
9	3	1357.8
24	7	3168.2
38	10	4526.0

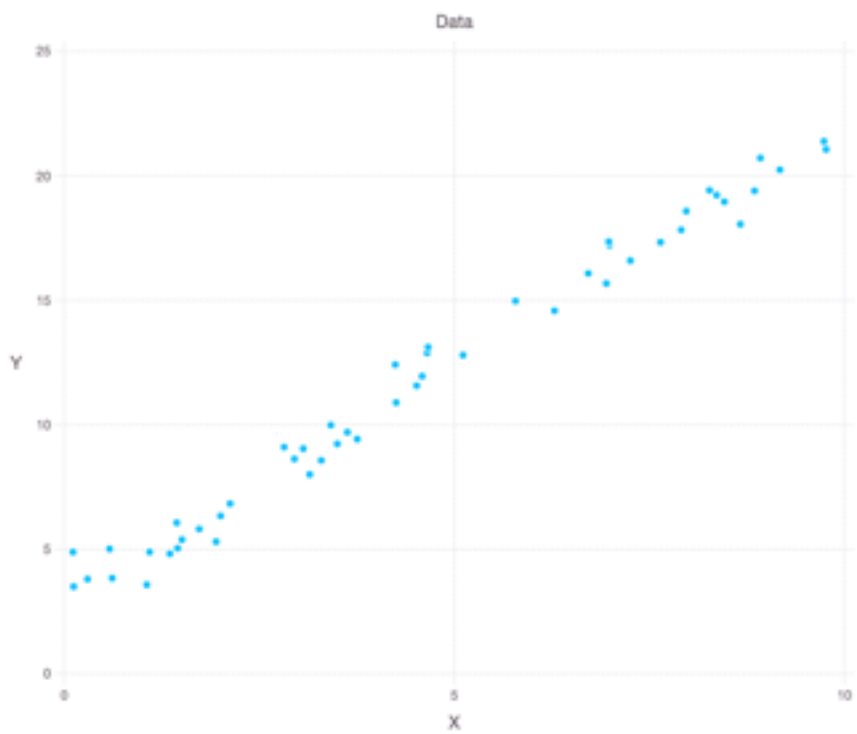
cor(Cost USD,pounds) == 0.998
cor(Cost USD,grams) == 0.998

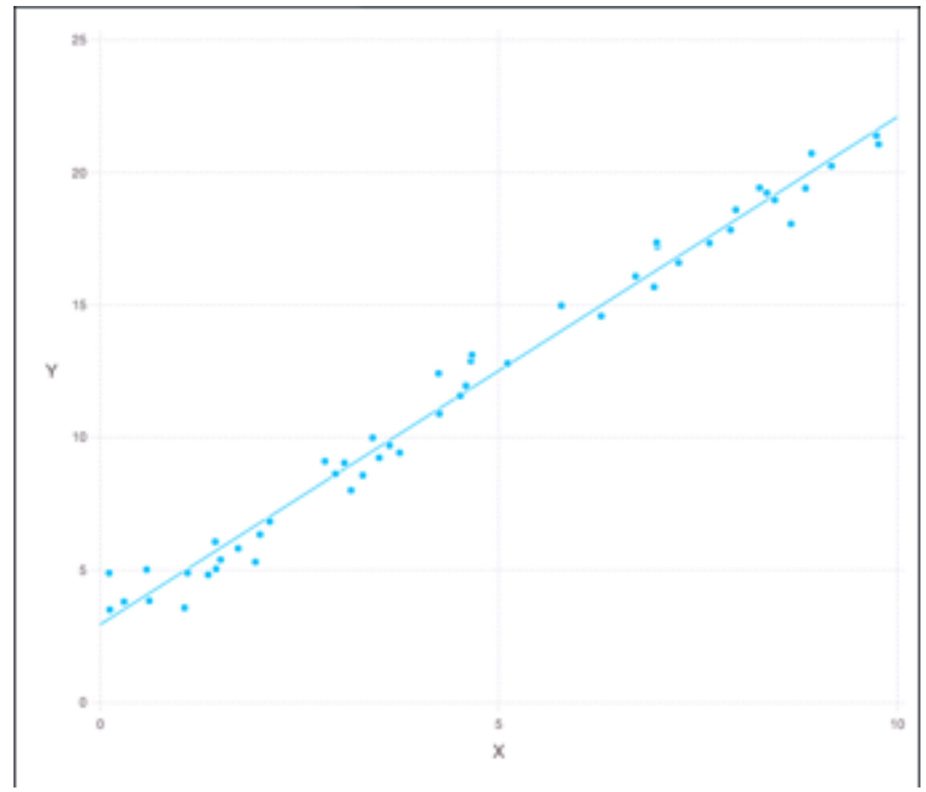
Pearson's Correlation r Value Examples



https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

Regression Line





Pearson’s Co
cor(x,y) == 0.992

What the line that minimizes the amount of residuals

Ordinary least squares

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Standard way to fit line to data

$$\mathrm{b} = \frac{\mathrm{cov}(\mathrm{X},\mathrm{Y})}{\mathrm{var}(\mathrm{X})}$$

$$a = \bar{y} - b\bar{x}$$

GLM.jl Package

Linear models (lm) & Generalized linear models (glm)

Pkg.add("GLM")
using GLM

lm(independentVars,dataframe) returns linear model fitting the data

glm(independentVars,dataframe,distribution, link)

fit() called by glm and lm to produce model

residuals(model)	
coef(model)	returns coefficients of fitted line
deviance(model)	
stderr(model)	
predict(model)	returns predicted values of dependent variable
r2(model)	

Example - Some Fake Data

```
using DataFrames
using Gadfly
using GLM
using Distributions
```

```
#Adds random amount to value from distribution “dist”
#Amount added is less than limit

function jitter(dist,value,limit)
  value + (rand(dist,1)[1] * 2 * limit ) - limit
end
```

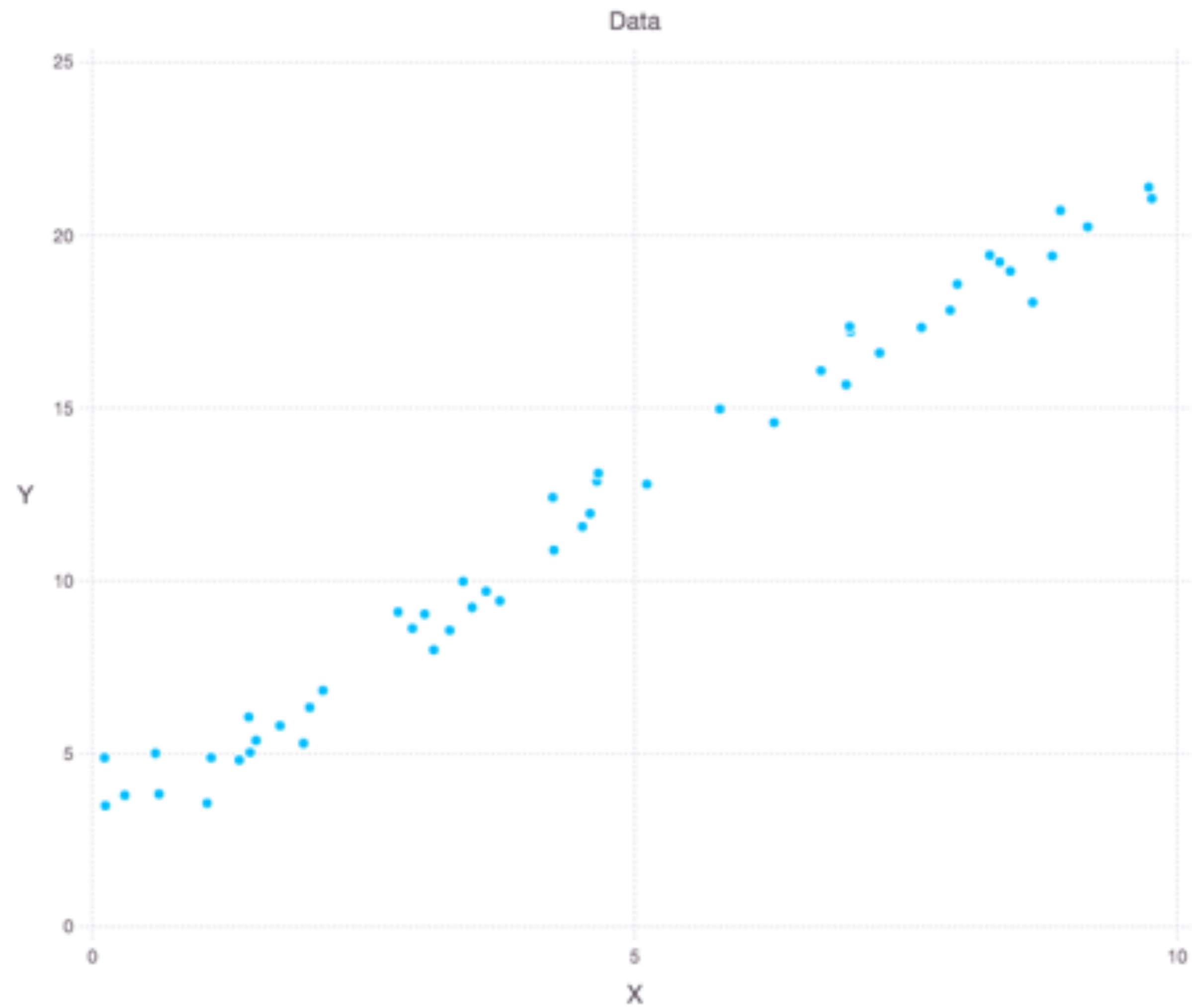
```
f(x) = 2*x + 3

x = rand(50) * 10

y = map(z -> jitter(Normal(),f(z), 0.4), x)
```

Example - Are X & Y related linearly?

Pearson's Co
cor(x,y) == 0.992




```
near_exact_data = DataFrame(X=x,Y=y)
plot(near_exact_data,x="X",y="Y",Geom.point,
        Guide.XLabel("X"),Guide.YLabel("Y"),Guide.Title("Data"))
```

Fitting the Data

```
near_exact_model = lm(Y~X, near_exact_data)
show(near_exact_model)
```

```
Formula: Y ~ 1 + X

Coefficients:
             Estimate Std.Error t value Pr(>|t|)
(Intercept)   2.94384  0.188246 15.6382   <1e-19
X             1.91493 0.0344778 55.5411   <1e-44
```

Source

f(x) = 2*x + 3

Model

fitted_f(x) = 1.91493*x + 2.94384

What is t?

	Estimate	Std.Error	t value	Pr(>|t|)
(Intercept)	2.94384	0.188246	15.6382	<1e-19
X	1.91493	0.0344778	55.5411	<1e-44

From Student's T-test

Used when do not know the population parmeters

When population in know use z value

Used to determine if should accept the regression line

Use Pr(>|t|)

Examples

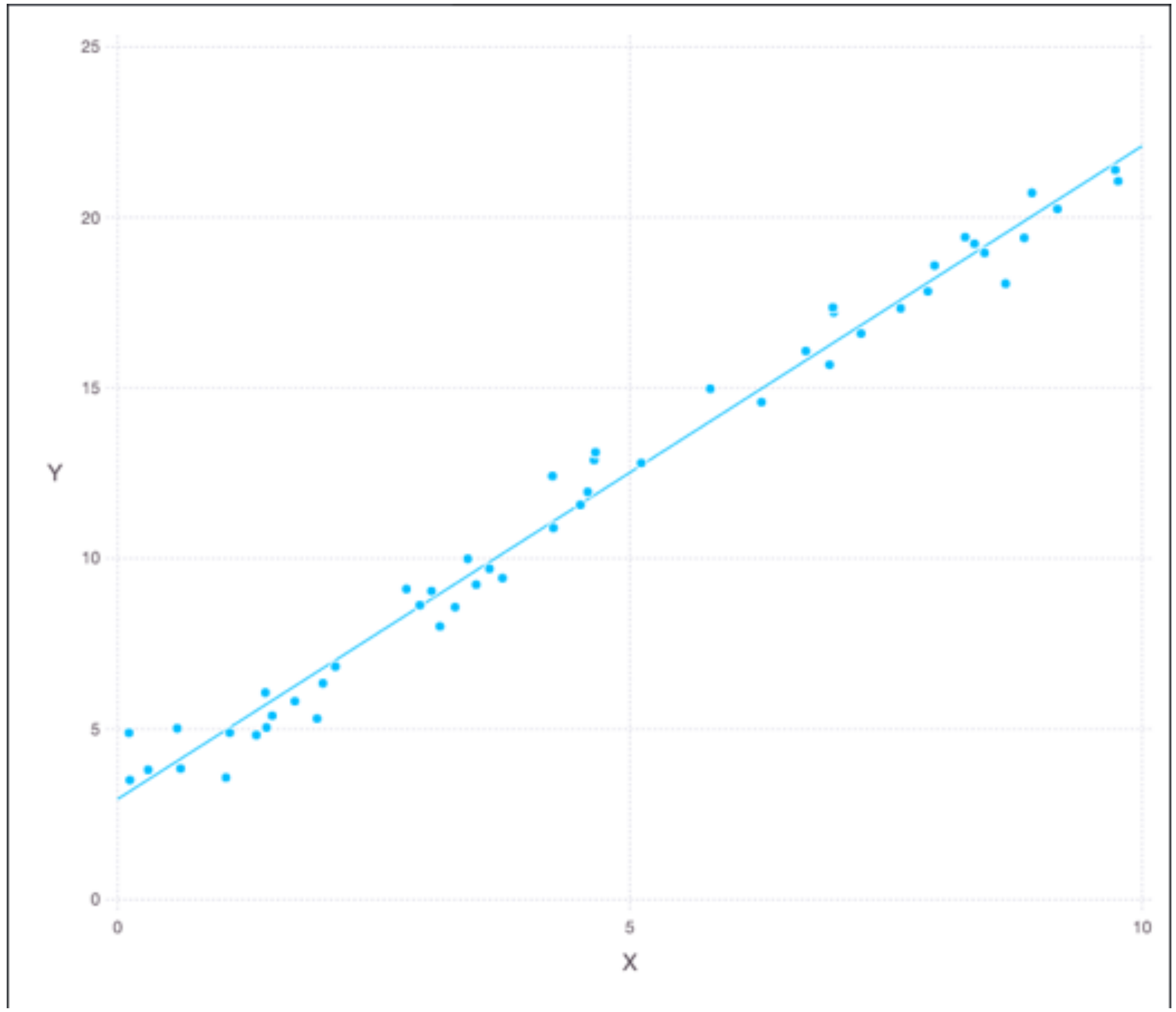
X & Y both random, no relation   cor(x,y) == 0.0254

	Estimate	Std.Error	t value	Pr(>|t|)
(Intercept)	10.8038	0.942533	11.4625	<1e-22
X	0.0270376	0.0756465	0.35742	0.7212

Y = X   cor(x,y) == 1.0

	Estimate	Std.Error	t value	Pr(>|t|)
(Intercept)	2.00972e-15	1.67129e-16	12.025	<1e-24
X	1.0	1.34135e-17	7.45515e16	<1e-99

Regression Line

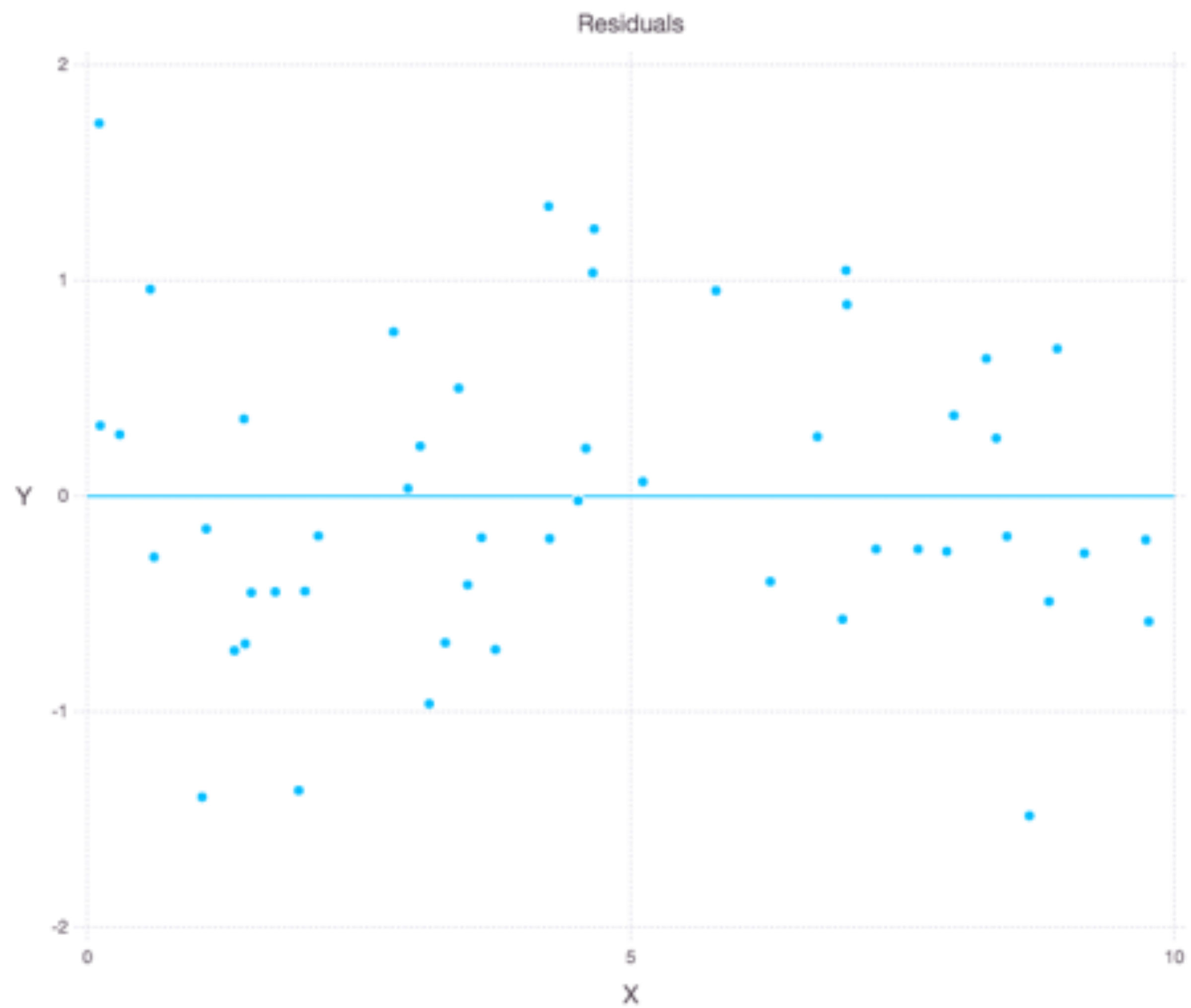



```
fitted_f(x) = 1.91493*x + 2.94384
plot(layer(near_exact_data,x="X",y="Y",Geom.point),
     layer(fitted_f,0,10),
     Guide.XLabel("X"),Guide.YLabel("Y"))
```

Regression Equation

```
fitted_coef = coef(near_exact_model)
fitted_f(x) = fitted_coef[2]*x + fitted_coef[1]
```

Residuals




```
near_exact_data[:Residual] = residuals(near_exact_model)

plot(layer(near_exact_data,x="X",y="Residual",Geom.point),
     layer(x-> 0, 0,10),
    Guide.XLabel("X"),Guide.YLabel("Y"),Guide.Title("Residuals"))
```

Coefficient of Determination R^2

$$R^2 = 1 - \frac{\text{var}(\varepsilon)}{\text{var}(Y)}$$

e = residuals

Y = observed data

Measure of how much the independent variable explains the variance of the data

r2(near_exact_model) == 0.985

So one independent variable x contributes 98.5% of the variation in the data

Simple Regression and R^2

If only one independent variable

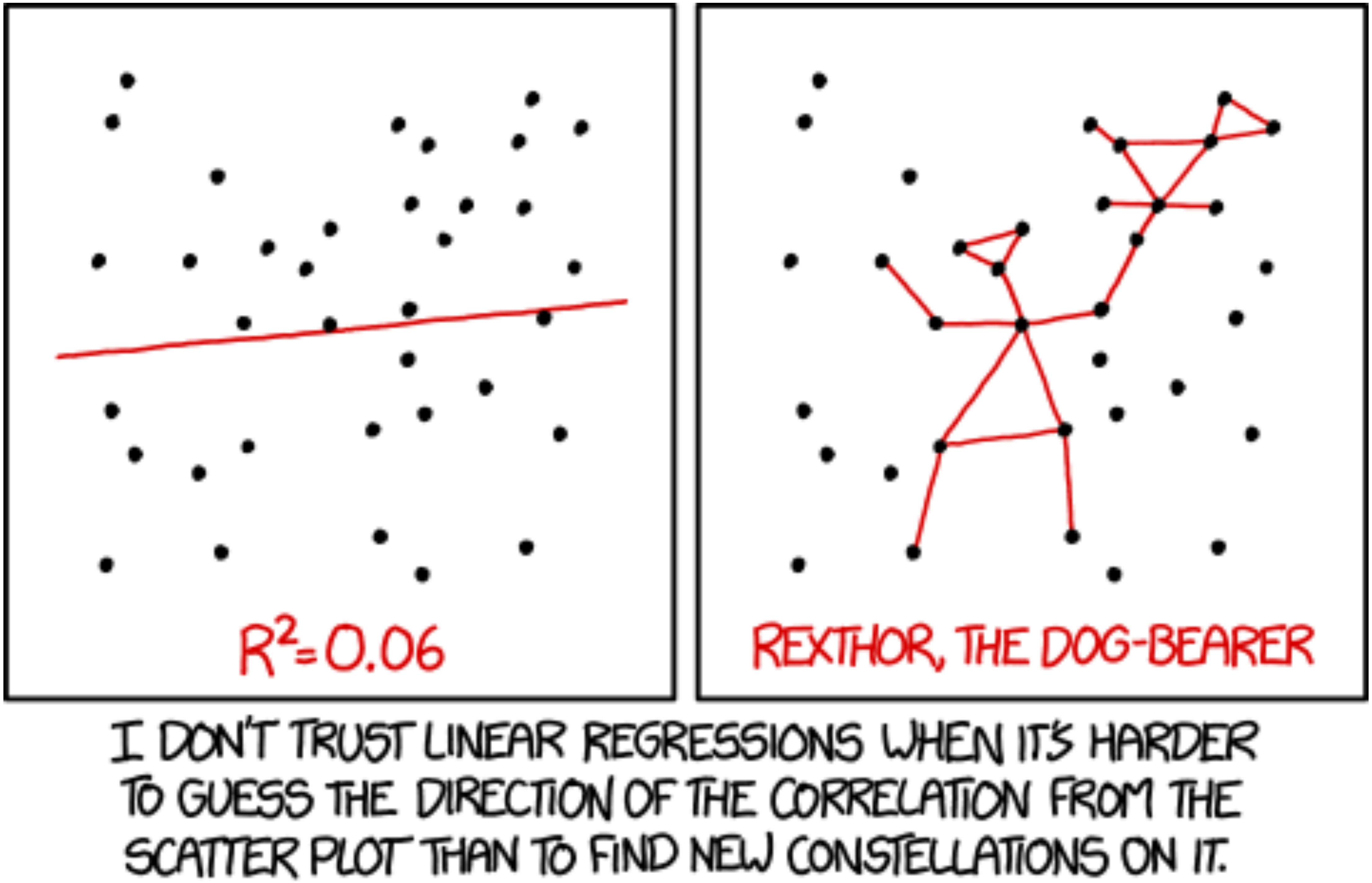
$R^2 = r^2$ (Pearson's Correlation squared)

In example

Pearson's Co
cor(x,y) == 0.992

r2(near_exact_model) == 0.985

0.992^2 == 0.984

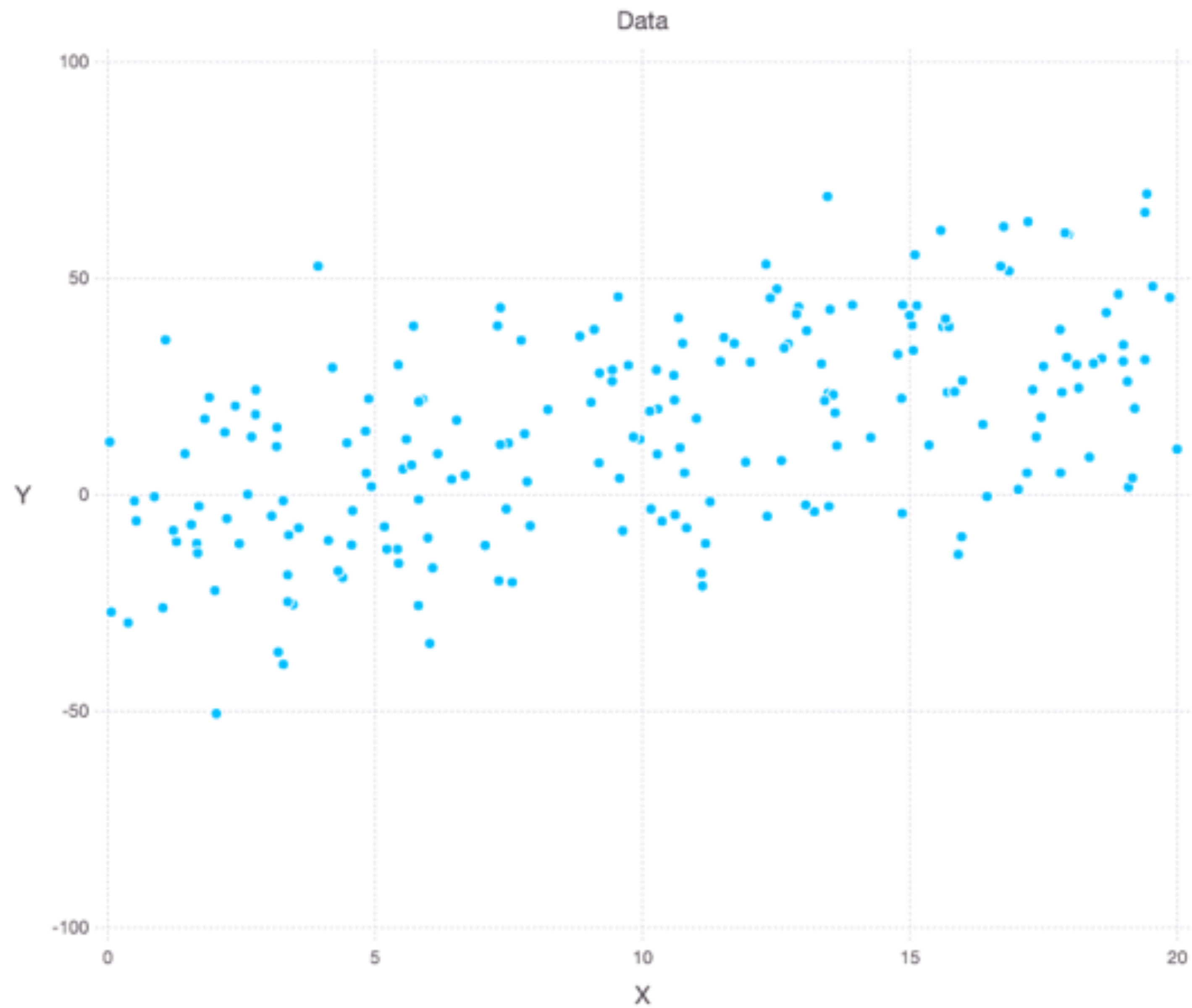


Tuesday, October 4, 16

http://xkcd.com/1725/

Second Example

cor(x,y) == 0.552

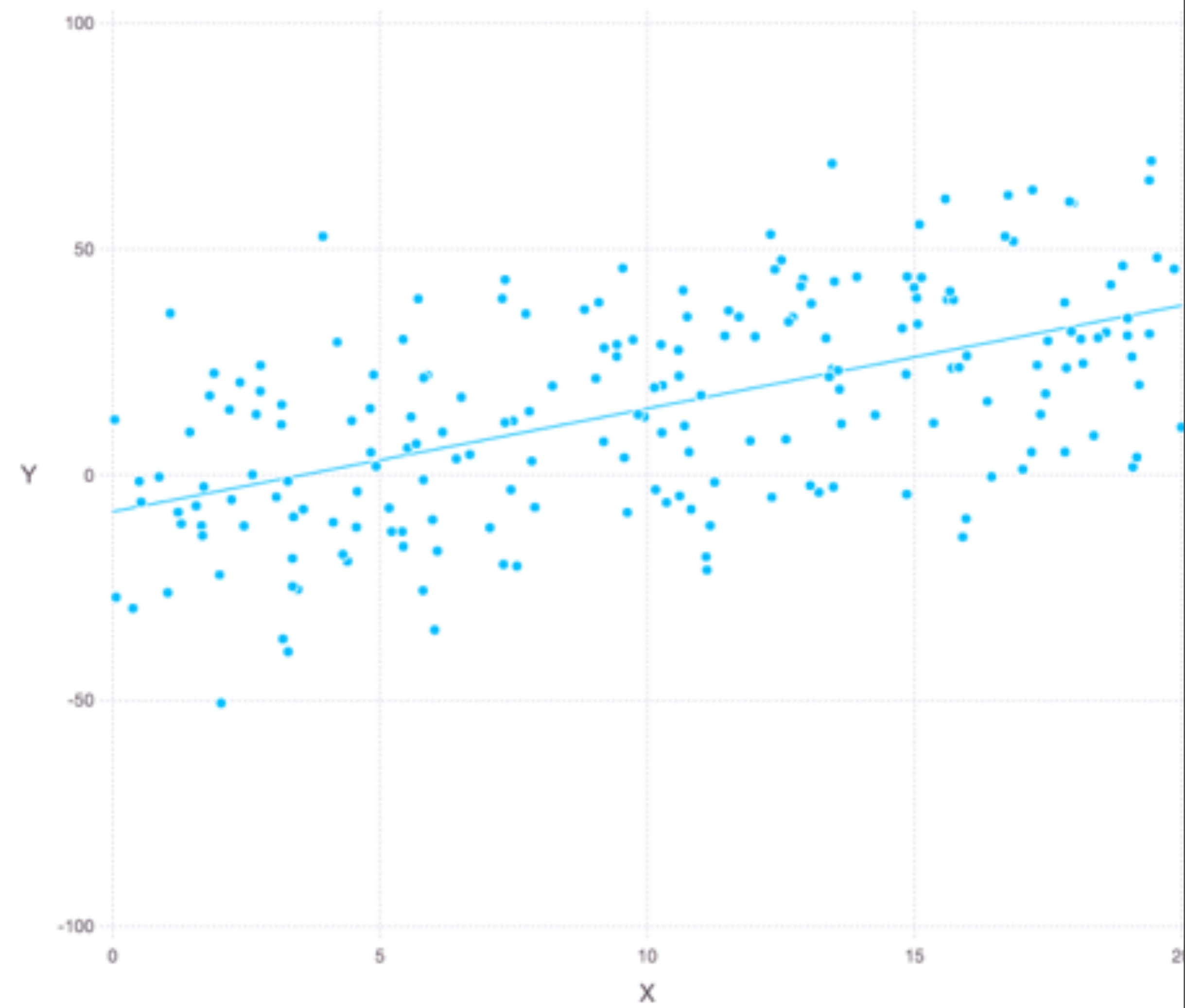


f(x) = 2*x + 3

x = rand(200) * 20

y = map(z -> jitter(Normal(),f(z), 10),x)

Regression line



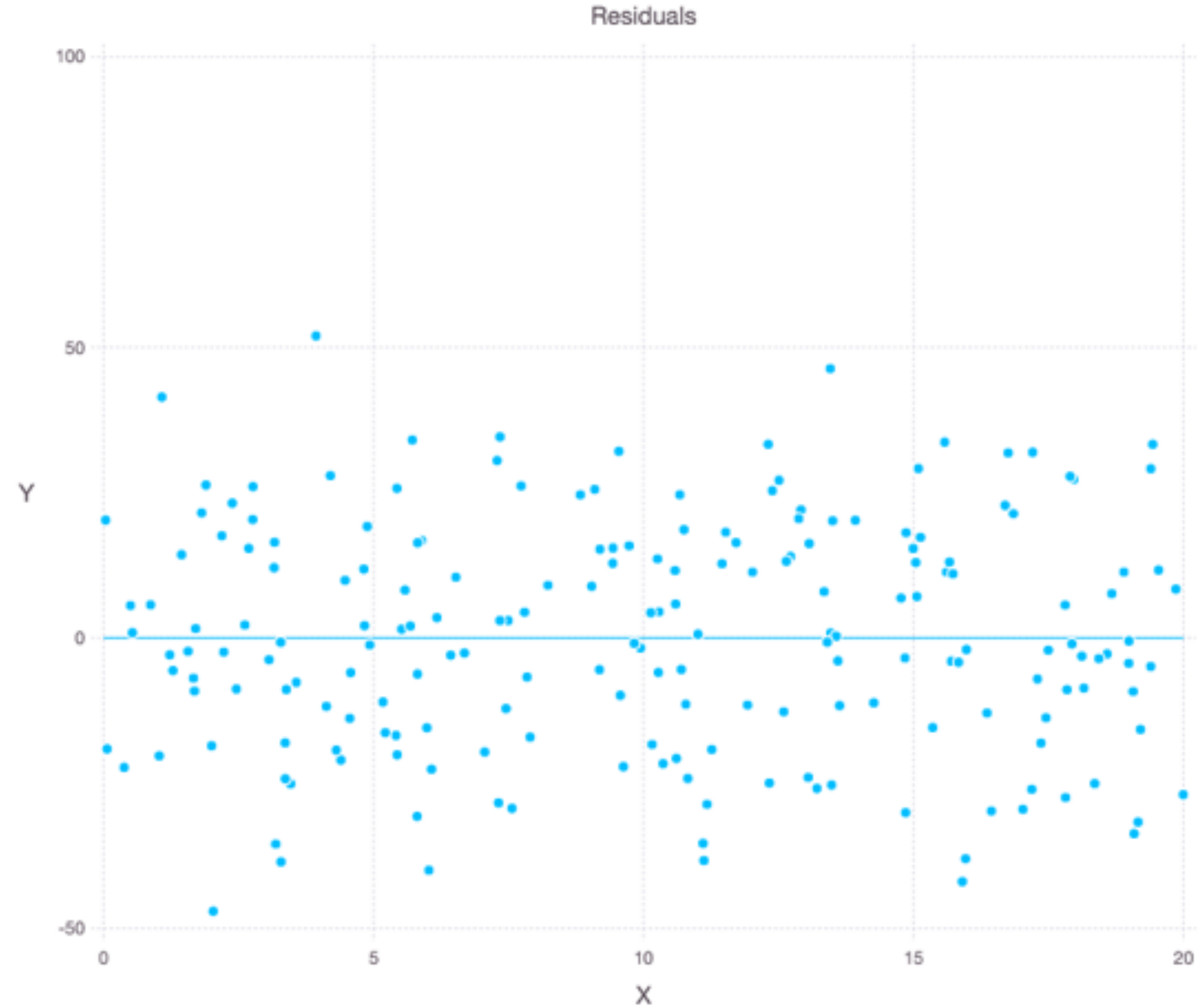

```
Coefficients:
              Estimate Std.Error  t value Pr(>|t|)
(Intercept)   -8.12406   2.83688 -2.86373   0.0046
X              2.28285   0.24535  9.30447   <1e-16
```

fitted_f(x) = 2.28*x - 8.12

f(x) = 2*x + 3

Residuals

$R^2 == 0.304$



Why Intercept So Off?

fitted_f(x) = 2.28*x - 8.12

f(x) = 2*x + 3

```
Coefficients:
             Estimate Std.Error  t value Pr(>|t|)
(Intercept)  -8.12406   2.83688 -2.86373   0.0046
X             2.28285   0.24535  9.30447   <1e-16
```

Multiple Linear Regression

Using multiple independent varibles

Amazon's daily revenue depends on

- Latency
- Price
- Steps needed to order
- Page layout
- Relevant suggestions
- Search results
- Font sizes
- Color
- Shipping costs

Two Independent Variable Example

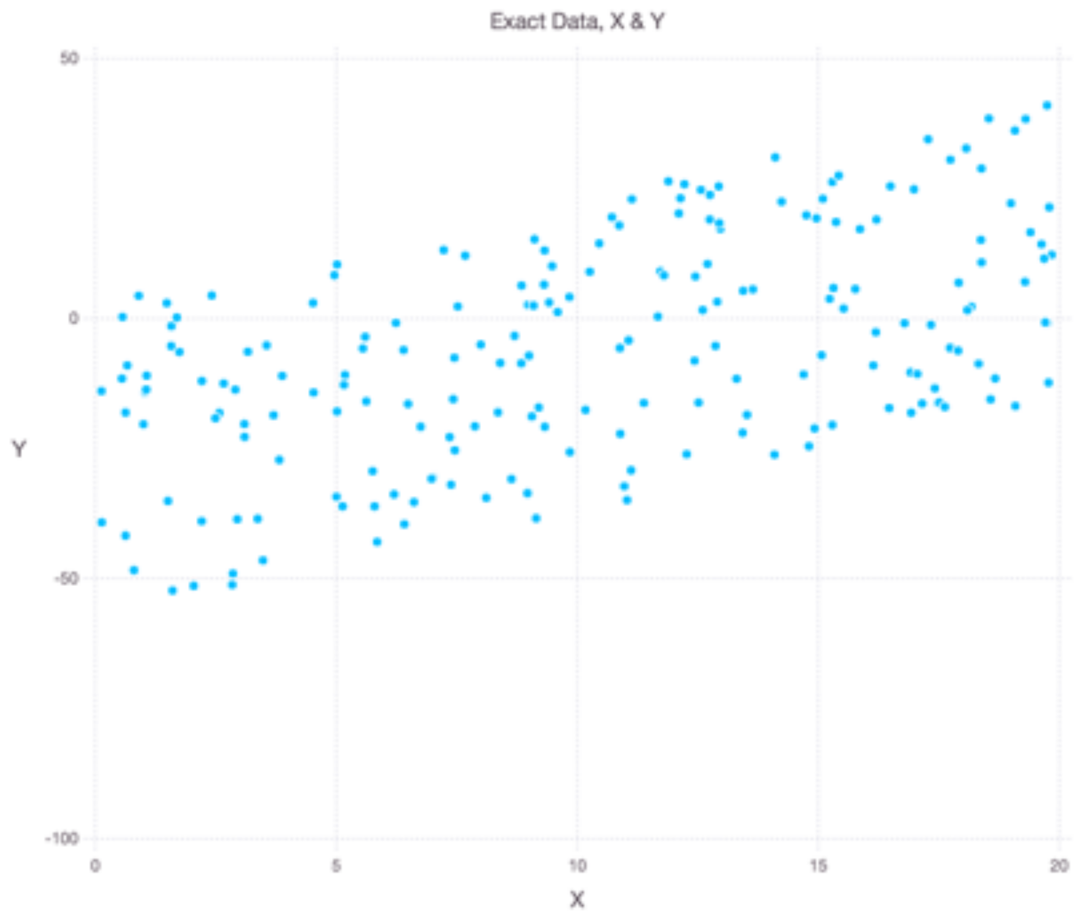
f(x, z) = 2*x - 3*z + 3

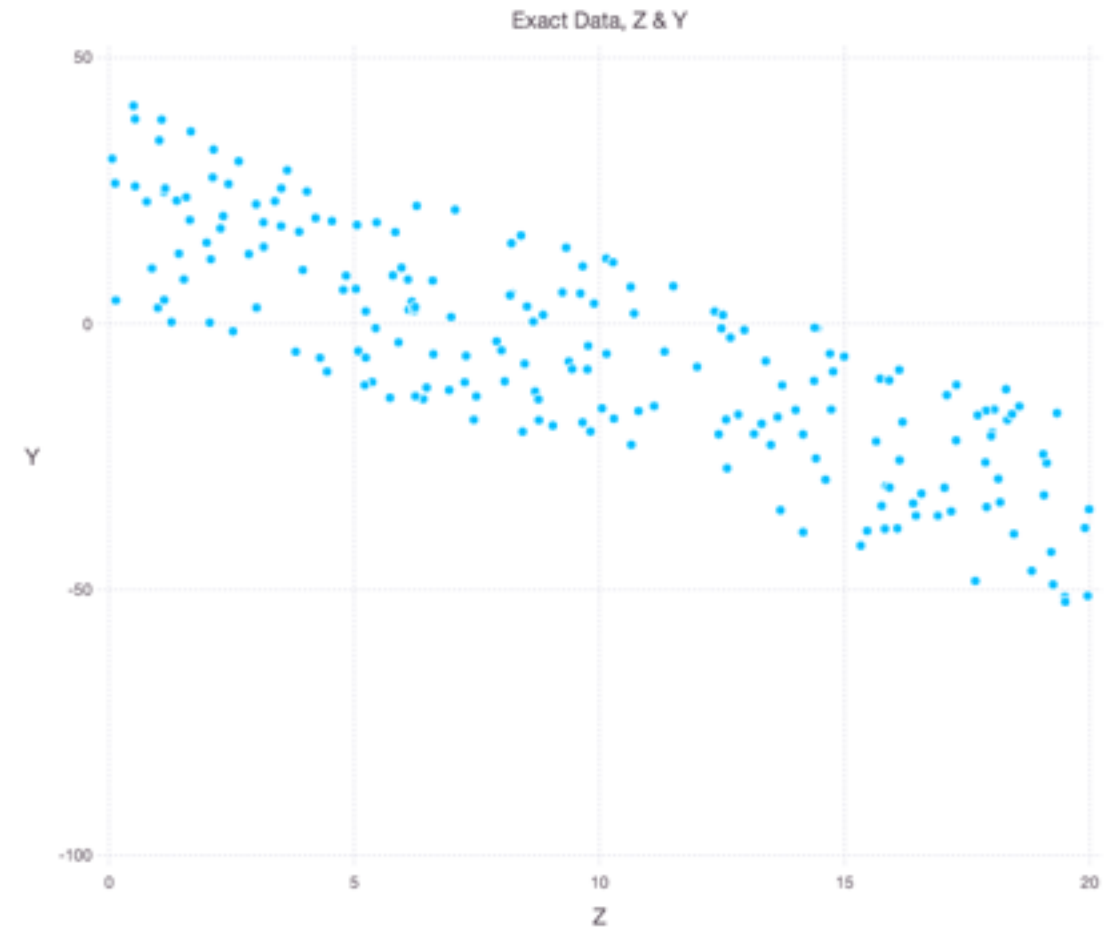
x = rand(200) * 20

z = rand(200) * 20

randomized_f(x,z) = jitter(Normal(),2*x, 1) - jitter(Normal(),3*z,0.5) + 3

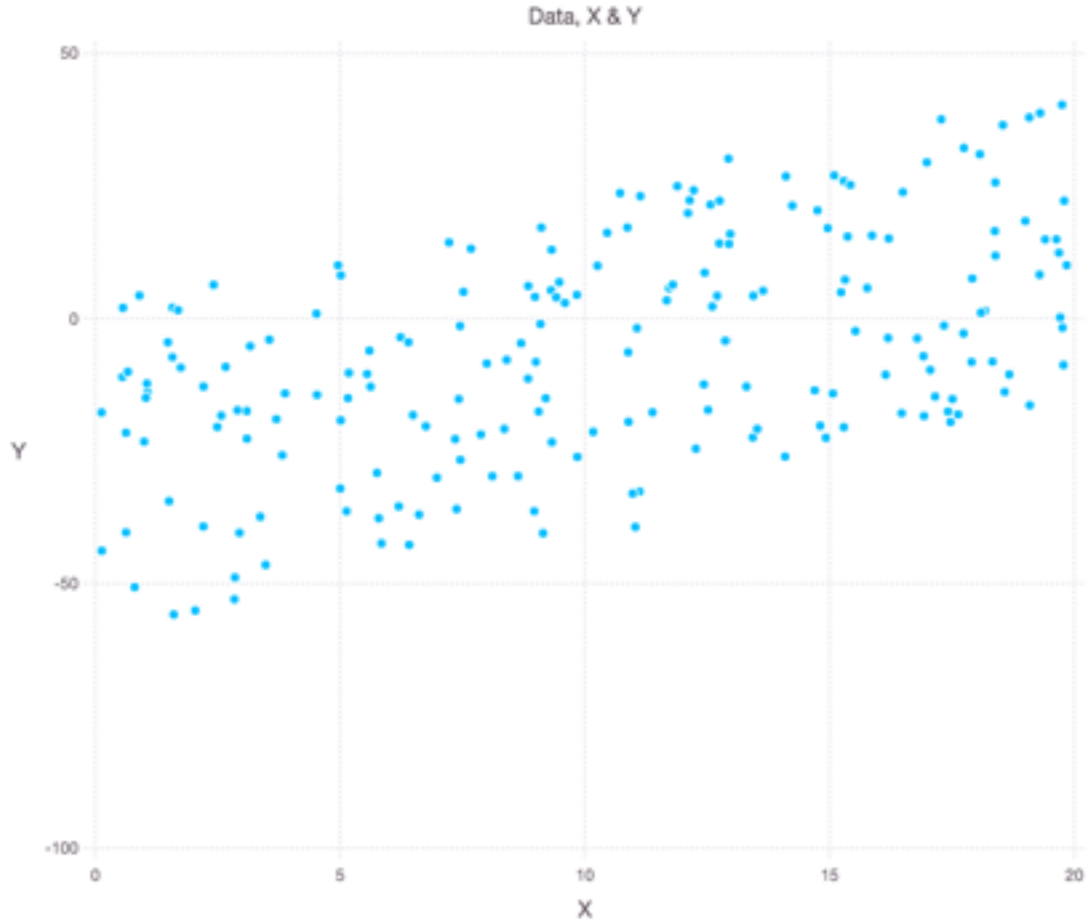
Exact Data

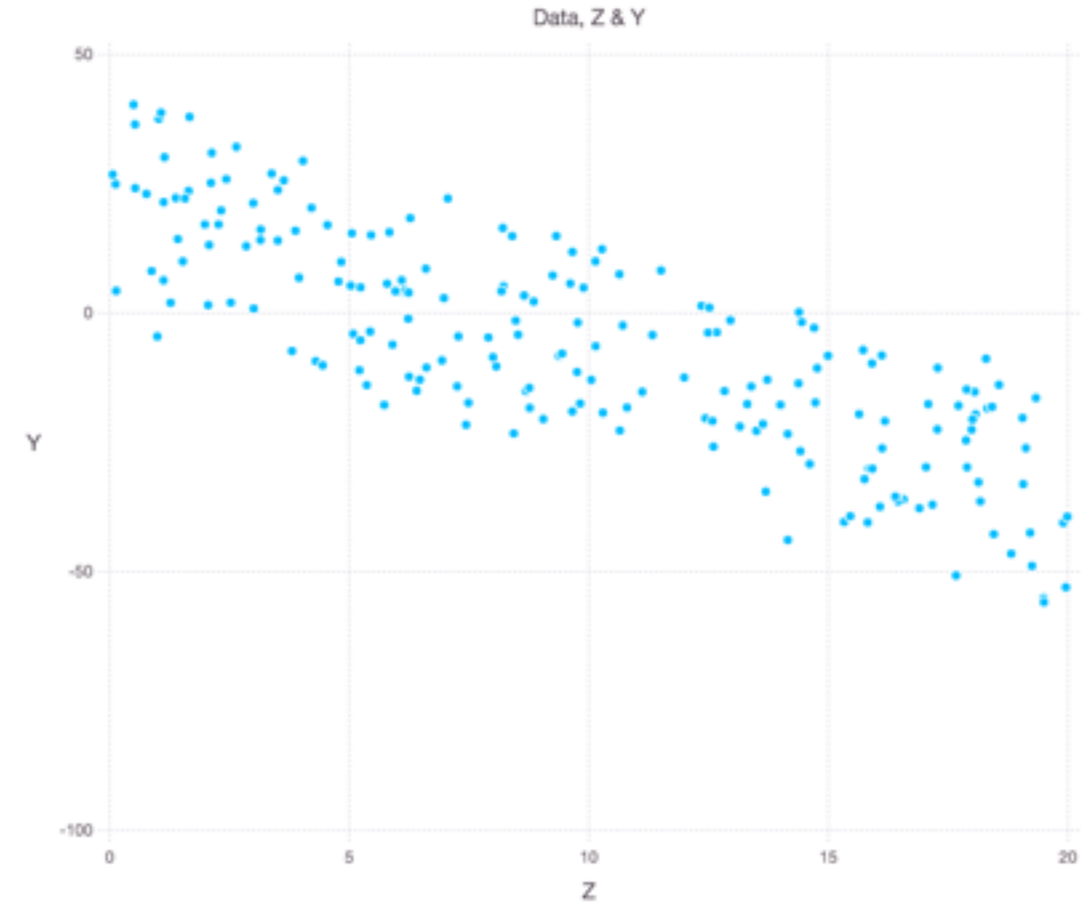





```
exact_y = map((x,z) -> f(x,z),x,z)
exact_data = DataFrame(X=x,Z=z,Y=exact_y)
plot(exact_data,x="X",y="Y",Geom.point,
                    Guide.XLabel("X"),Guide.YLabel("Y"),Guide.Title("Exact Data, X & Y"
plot(exact_data,x="Z",y="Y",Geom.point,
                    Guide.XLabel("Z"),Guide.YLabel("Y"),Guide.Title("Exact Data, Z & Y")
```

Fake Data






```
y = map((x,z) -> randomized_f(x,z),x,z)

two_data = DataFrame(X=x,Z=z,Y=y)
plot(two_data,x="X",y="Y",Geom.point,
        Guide.XLabel("X"),Guide.YLabel("Y"),Guide.Title("Data, X & Y"))
plot(two_data,x="Z",y="Y",Geom.point,
        Guide.XLabel("Z"),Guide.YLabel("Y"),Guide.Title("Data, Z & Y"))
```

cor(x,exact_y) == 0.519
cor(z,exact_y) == -0.825

cor(x,y) = 0.519
cor(z,y) = -0.819

The Model

two_model = lm(Y~X + Z,two_data)
show(two_model)

```
Formula: Y ~ 1 + X + Z

Coefficients:
             Estimate Std.Error  t value Pr(>|t|)
(Intercept)    2.1751  0.431312  5.04299    <1e-5
X             2.02513 0.0288004   70.316   <1e-99
Z            -3.00437 0.0285496 -105.233   <1e-99
```

fitted_coef = coef(two_model)
fitted_f(x,z) = fitted_coef[3]*z + fitted_coef[2]*x + fitted_coef[1]
= -3.004*z + 2.025*x + 2.1751

f(x, z) = 2*x - 3*z + 3

R^2 - Coefficient of Multiple Determination

When have multiple independent variables R^2 has new name

Adding an other independent variable

- Contributes to explain dependent variable
  - R^2 increases
- Has nothing to do with dependent variable
  - R^2 increases

Adjusted R^2

Modified version of R^2

Adding new independent variable only increases R^2 more that expected by chance

```
adjr2(two_model)
```