Tecnológico de Monterrey

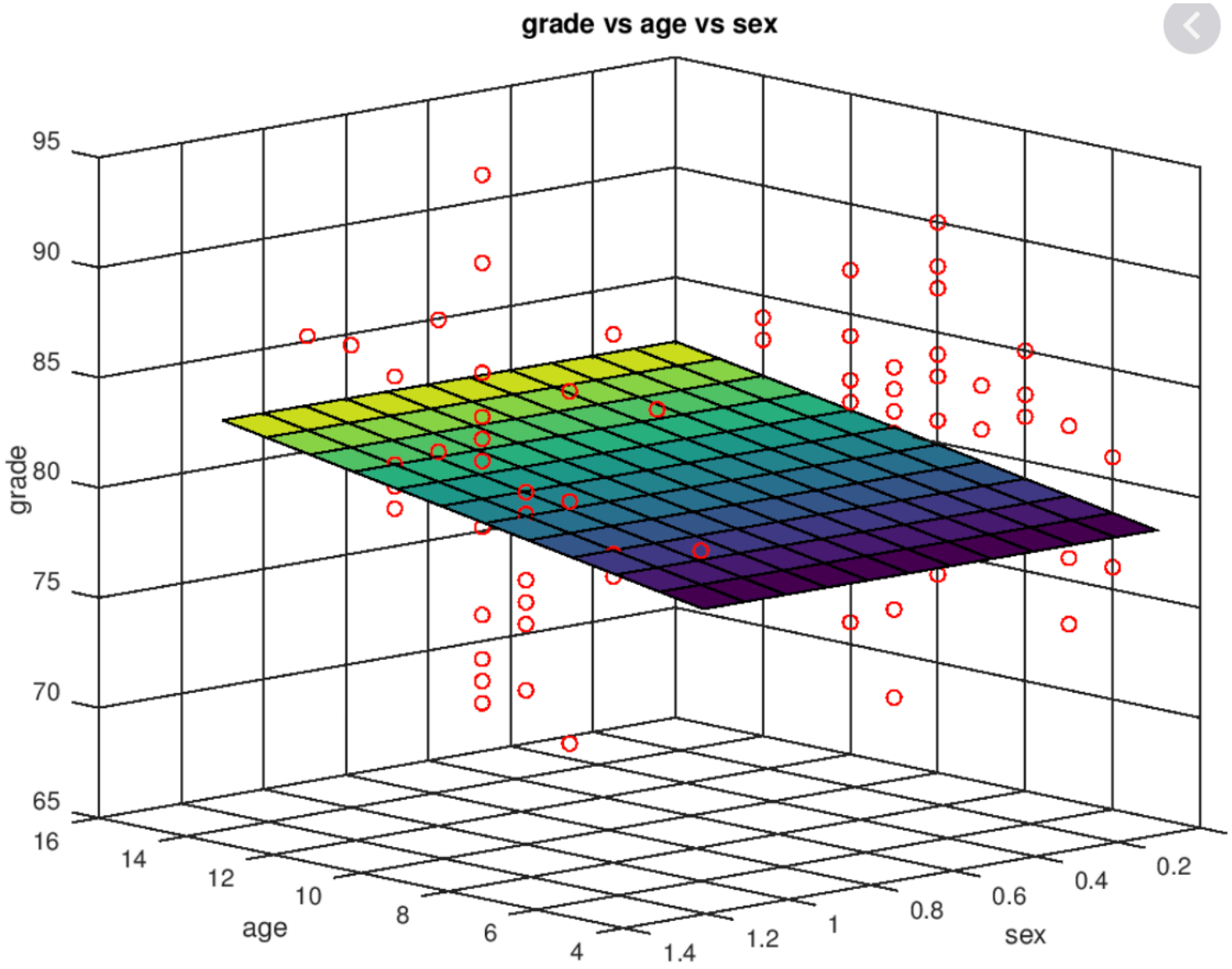
Multiple Regression Analysis

CS5056 Data Analytics

Francisco J. Cantú, Héctor Ceballos
Tecnológico de Monterrey
March 3, 2021

Februrary-June, 2021

Multiple Regression Analysis



Multiple Regression Analysis

- It is used to predict an outcome variable (y) on the basis of multiple distinct predictor variables (x_i)
- With three predictor variables (x_i), the prediction of y is expressed by the following multiple linear regression equation:
- $y = b_0 + b_1*x_1 + b_2*x_2 + b_3*x_3$
- The “b” values are called the regression weights (or *beta coefficients*)
- They measure the association between the predictor variable and the outcome. “b_j” can be interpreted as the average effect on y of a one unit increase in “x_j”, holding all other predictors fixed.

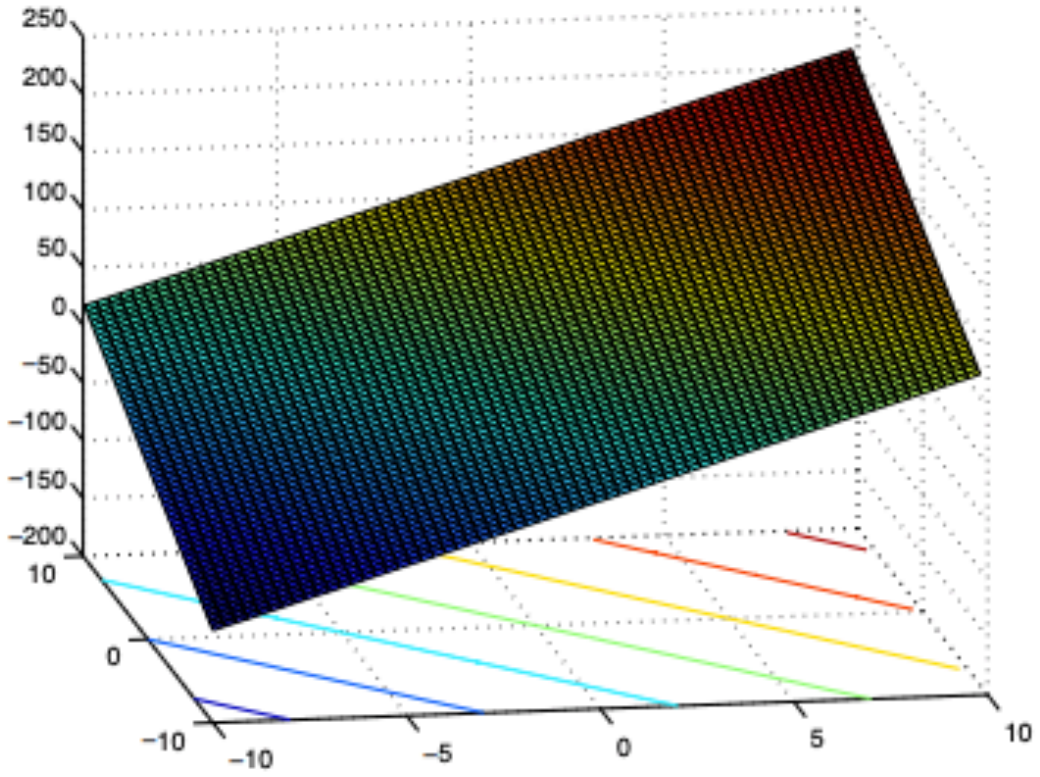
Example: The simplest multiple regression model for two predictor variables

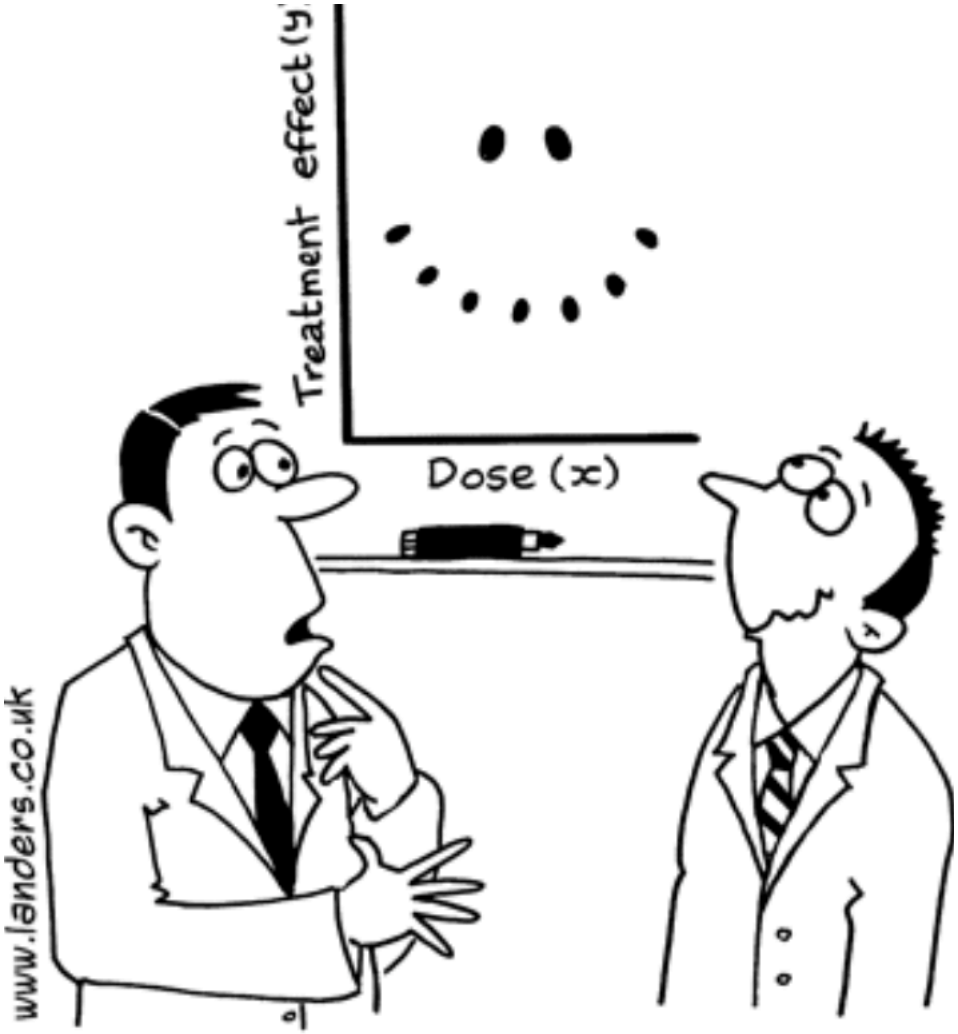
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

The surface that corresponds to the model

$$y = 50 + 10x_1 + 7x_2$$

looks like this. It is a plane in $\mathbb{R}^3$ with different slopes in x_1 and x_2 direction.

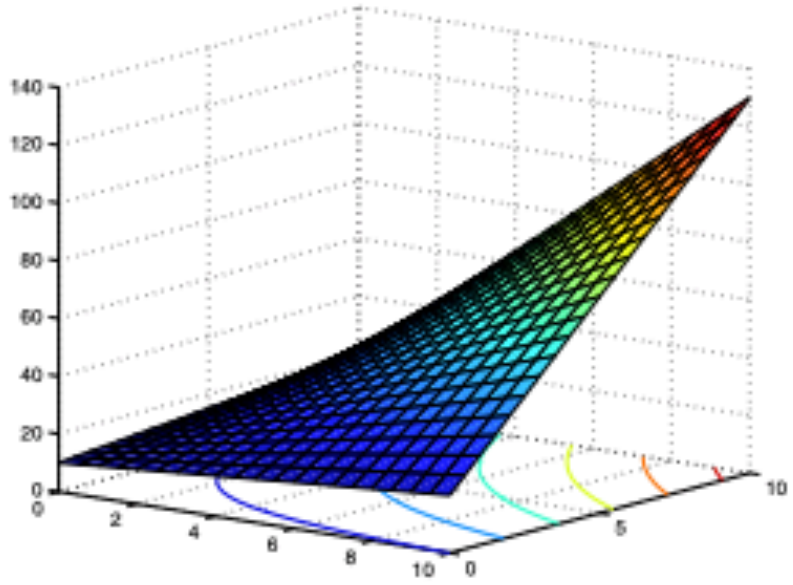




"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

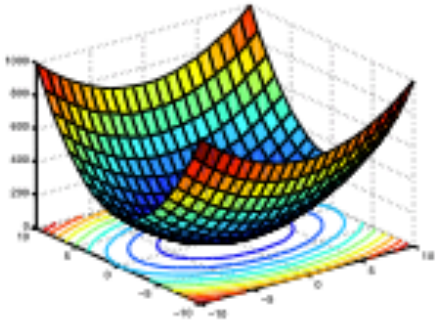
Example: For a simple linear model with two predictor variables and an interaction term, the surface is no longer flat but curved.

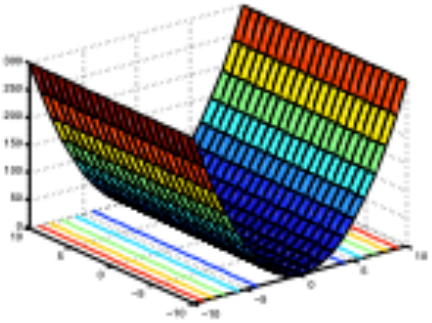
$$y = 10 + x_1 + x_2 + x_1 x_2$$

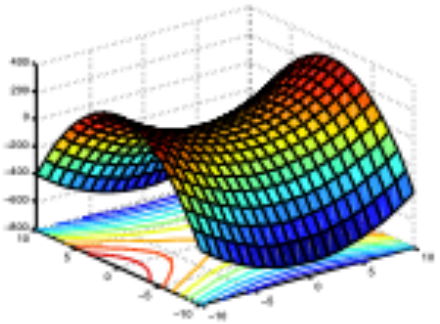


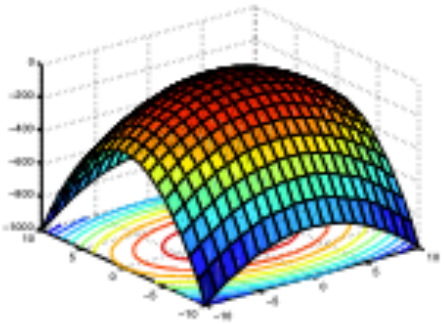
Example: Polynomial regression models with two predictor variables and interaction terms are quadratic forms. Their surfaces can have many different shapes depending on the values of the model parameters with the contour lines being either parallel lines, parabolas or ellipses.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon$$



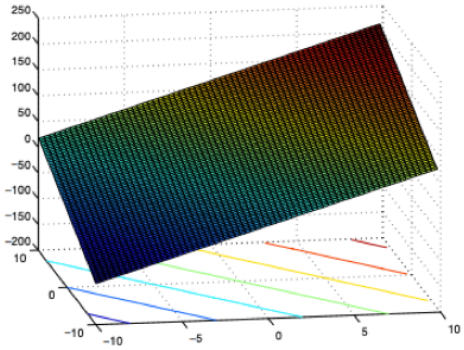






Polynomial regression

Is a polynomial equation a linear model?

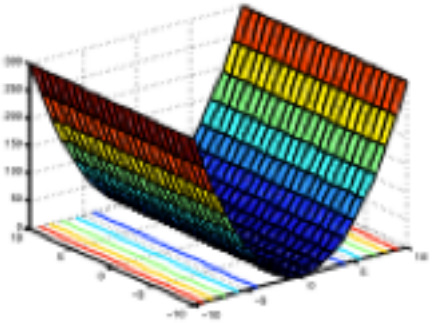


- First degree, 2-variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

- Quadratic, 2-variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2$$

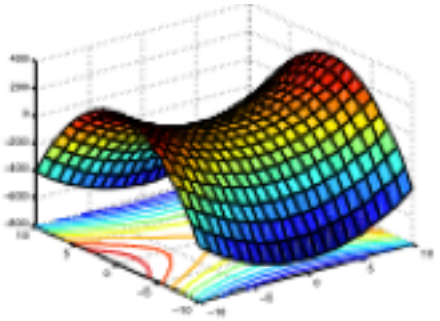


- n-degree, 1-variable

$$y = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n$$

- n-degree, m-variable

$$y = b_0 + b_{11} x_1 + b_{21} x_2^2 + \ldots + b_{nm} x^m$$



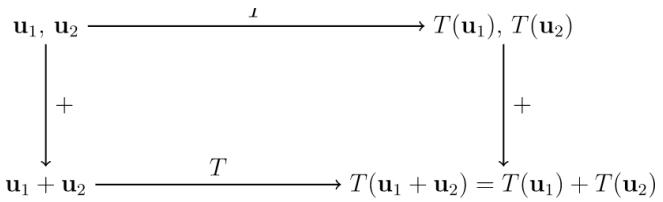
- Interacting variables

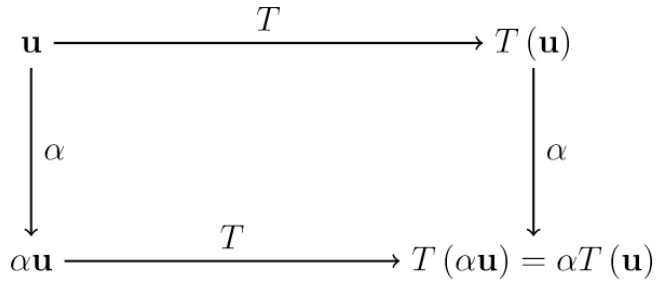
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon$$

Linear Combination

- A ***linear combination (transformation)***, T:U→V, is a function that converts elements of the vector space U (called the ***domain***) to the vector space V (called the ***codomain***), with two properties:

  - T(x+y)=T(x)+T(y) for all x, y ∈ U
  - T(αx)=αT(x) for all x ∈ U and all α ∈ C



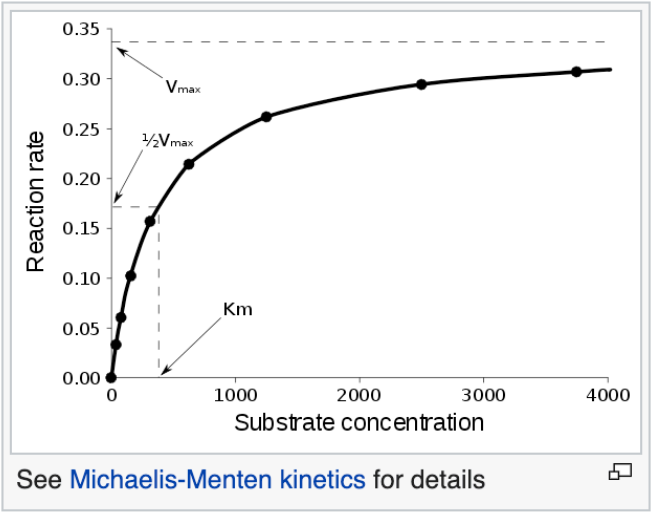


http://linear.ups.edu/html/section-LT.html

Non-Linear Regression

- It is a form of regression analysis in which observational data are modeled by a function which is a nonlinear combination of the model parameters and depends on one or more independent variables. The data are fitted by a method of successive approximations
- For example, the Michaelis–Menten model for enzyme kinetics has two parameters and one independent variable, related by f
- This function is nonlinear because it cannot be expressed as a linear combination of the two betas.

$$f(x, \boldsymbol{\beta}) = \frac{\beta_1 x}{\beta_2 + x}$$

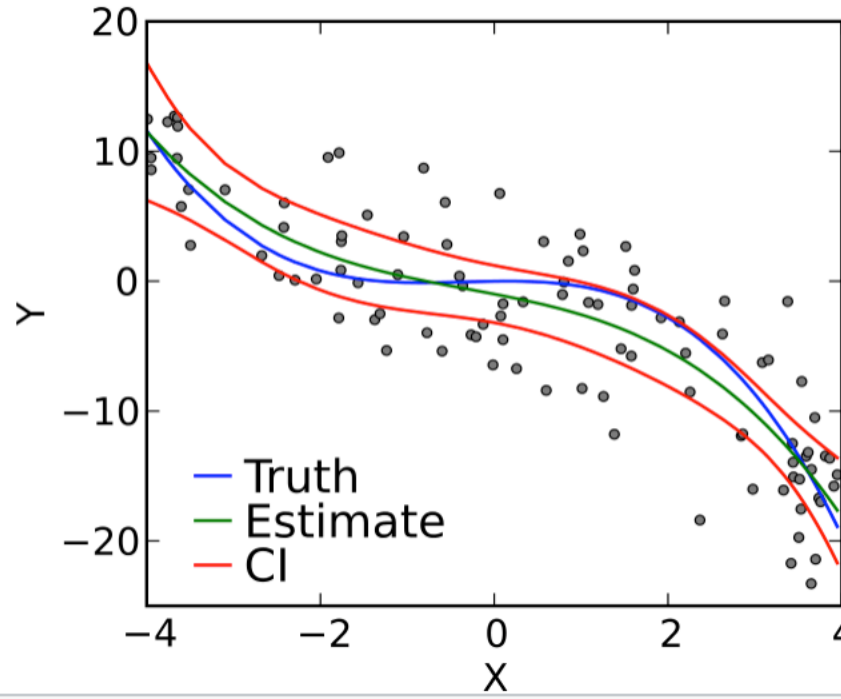


See Michaelis-Menten kinetics for details

https://en.wikipedia.org/wiki/Nonlinear_regression

Polynomial Linear Regression

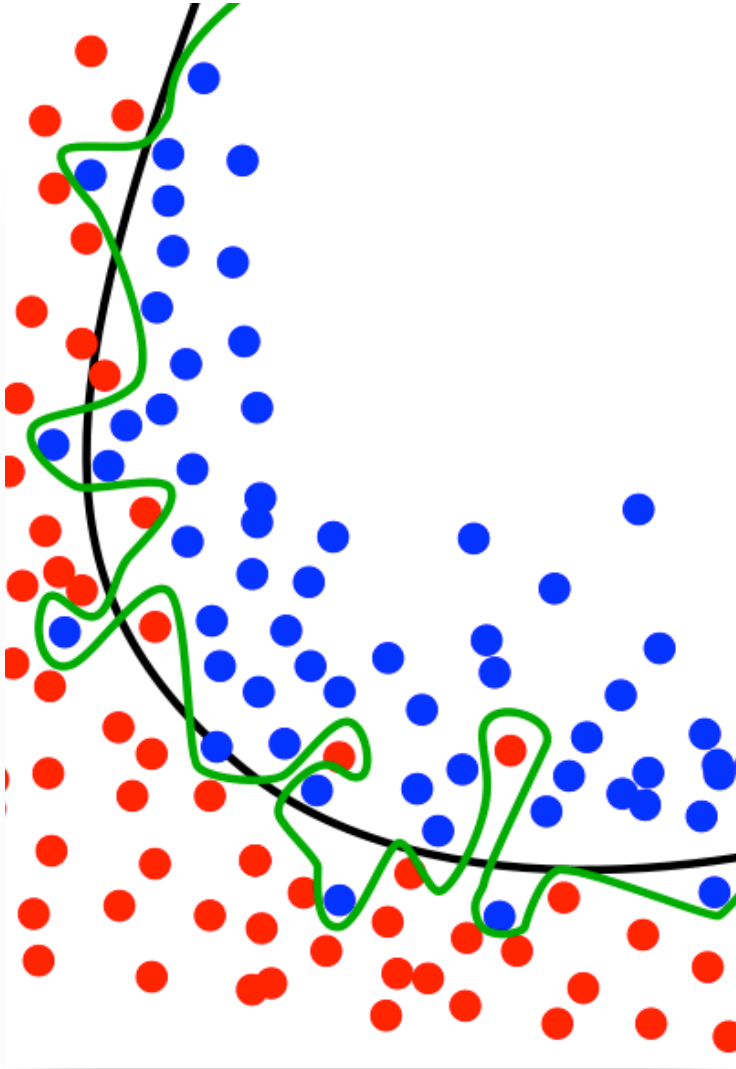
- Since the predictor variables are treated as fixed values, linearity is only a restriction on the parameters
- The predictor variables themselves can be arbitrarily transformed, and in fact multiple copies of the same underlying predictor variable can be added, each one transformed differently
- This technique is used, for example, in polynomial regression, which uses linear regression to fit the response variable as an arbitrary polynomial function (up to a given rank) of a predictor variable
- This makes linear regression an extremely powerful inference method
- In fact, models such as polynomial regression are often "too powerful", in that they tend to overfit the data.



Example of a cubic polynomial regression, which is a type of linear regression.

https://en.wikipedia.org/wiki/Linear_regression

Overfitting



https://en.wikipedia.org/wiki/Overfitting

Business Problem

- A company needs to build a predictive model for estimating sales of one of its main line of products
- The company holds an annual sales budget and needs to know how best invest this money in order to maximize product sales
- Options for advertising budget include purchasing ads in either Youtube, Facebook, or newspaper
- The company holds historical records of 200 campaigns

Business Problem

Best Media for Marketing Campaign

Period	Youtube	Facebook	Newspaper	Sales
1	276.1	45.4	83.0	26.5
2	53.4	47.2	54.1	12.5
3	20.6	55.1	83.2	11.2
4	181.8	49.6	70.2	22.2

Business Question

- How much should the company inveast in each media?

Hands-on Exercise

- data("marketing", package = "datarium")
- **head**(marketing, 5)
- sales = b0 + b1*youtube + b2*facebook + b3*newspaper
- model <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
- summary(model)

http://www.sthda.com/english/articles/40-regression-analysis/168-multiple-linear-regression-in-r/

Hands-on Exercise

```
Call:
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
    Min      1Q  Median      3Q     Max
-10.59  -1.07    0.29    1.43    3.40

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.52667    0.37429    9.42   <2e-16 ***
youtube      0.04576    0.00139   32.81   <2e-16 ***
facebook     0.18853    0.00861   21.89   <2e-16 ***
newspaper   -0.00104    0.00587   -0.18     0.86
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.02 on 196 degrees of freedom
Multiple R-squared:  0.897,  Adjusted R-squared:  0.896
F-statistic:  570 on 3 and 196 DF,  p-value: <2e-16
```

Interpretation

- The first step in interpreting the multiple regression analysis is to examine the F-statistic and the associated p-value
- In our example, it can be seen that p-value of the F-statistic is < 2.2e-16, which is highly significant
- This means that, at least, one of the predictor variables is significantly related to the outcome variable
- To see which predictor variables are significant, you can examine the coefficients table, which shows the estimate of regression beta coefficients and the associated t-statitic and p-values

Hands-on Exercise

- summary(model)$coefficient

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.52667	0.37429	9.422	1.27e-17
youtube	0.04576	0.00139	32.809	1.51e-81
facebook	0.18853	0.00861	21.893	1.51e-54
newspaper	-0.00104	0.00587	-0.177	8.60e-01

Interpretation

- For a given the predictor, the t-statistic evaluates whether or not there is significant association between the predictor and the outcome variable, that is whether the beta coefficient of the predictor is significantly different from zero.

- Changing in youtube and facebook advertising budget are significantly associated to changes in sales while changes in newspaper budget is not significantly associated with sales.

- For a given predictor variable, the coefficient (b) can be interpreted as the average effect on y of a one unit increase in predictor, holding all other predictors fixed.

Interpretation

- For example, for a fixed amount of youtube and newspaper advertising budget, spending an additional 1 000 dollars on facebook advertising leads to an increase in sales by approximately 0.1885*1000 = 189 sale units, on average.

- The youtube coefficient suggests that for every 1 000 dollars increase in youtube advertising budget, holding all other predictors constant, we can expect an increase of 0.045*1000 = 45 sales units, on average.

- We found that newspaper is not significant in the multiple regression model. This means that, for a fixed amount of youtube and newspaper advertising budget, changes in the newspaper advertising budget will not significantly affect sales units.

- As the newspaper variable is not significant, it is possible to remove it from the model:

Model Validation

- YouTube alone (Simple Linear Regression)
- YouTube, FaceBook, and Newspaper
- YouTube and FaceBook
- YouTube, FaceBook, and a combination of both (youtube * facebook)

Hands-on Exercise

```
:
ormula = sales ~ youtube + facebook, data = r

duals:
Min       1Q   Median       3Q      Max
557  -1.050    0.291    1.405    3.399

ficients:
            Estimate Std. Error t value Pr(>|t|)
ercept)   3.50532    0.35339    9.92   <2e-16
ube       0.04575    0.00139   32.91   <2e-16
book      0.18799    0.00804   23.38   <2e-16

if. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05

dual standard error: 2.02 on 197 degrees of
iple R-squared:  0.897,  Adjusted R-squared:
atistic:  860 on 2 and 197 DF,  p-value: <2e-
```

- model <- lm(sales ~ youtube + facebook, data = marketing)
- summary(model)

Interpretation

```
Call:
lm(formula = sales ~ youtube + facebook, data = marketing)

Residuals:
    Min      1Q  Median      3Q     Max
-10.557  -1.050   0.291   1.405   3.399

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.50532    0.35339    9.92   <2e-16 ***
youtube      0.04575    0.00139   32.91   <2e-16 ***
facebook     0.18799    0.00804   23.38   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.02 on 197 degrees of freedom
Multiple R-squared:  0.897,  Adjusted R-squared:  0.896
F-statistic:  860 on 2 and 197 DF,  p-value: <2e-16
```

Interpretation

- Finally, our model equation can be written as follow:

- sales = 3.5 + 0.045*youtube + 0.187*facebook

- The confidence interval of the model coefficient can be extracted as follow:

- confint(model)

	2.5 %	97.5 %
(Intercept)	2.808	4.2022
youtube	0.043	0.0485
facebook	0.172	0.2038

Model Accuracy Assessment

- R2 represents the correlation coefficient between the observed values of the outcome variable (y) and the fitted (i.e., predicted) values of y
- The value of R will always be positive and will range from zero to one
- R2 represents the proportion of variance, in the outcome variable y, that may be predicted by knowing the value of the x variables
- An R2 value close to 1 indicates that the model explains a large portion of the variance in the outcome variable.

Model Accuracy Assessment

- A problem with the **R2**, is that it will always increase when more variables are added to the model, even if those variables are only weakly associated with the response
- A solution is to adjust the R2 by taking into account the number of predictor variables.
- The adjustment in the “Adjusted R Square” value in the summary output is a correction for the number of x variables included in the prediction model: From 0.897 to 0.896

Model Accuracy Assessment

- In our example, with youtube and facebook predictor variables, the adjusted R2 = 0.89, meaning that “89% of the variance in the measure of sales can be predicted by youtube and facebook advertising budgets

- Thi model is better than the simple linear model with only youtube (Chapter simple-linear-regression), which had an adjusted R2 of 0.61.

Model Accuracy Assessment

- Residual Standard Error (RSE), or sigma estimate gives a measure of error of prediction
- The lower the RSE, the more accurate the model

- The error rate can be estimated by dividing the RSE by the mean outcome variable:

- sigma(model)/mean(marketing$sales)
- [1] 0.12

Model Accuracy Assessment

- In our multiple regression example, the RSE is 2.023 corresponding to **12%** error rate.

- This is better than the simple model, with only youtube variable, where the RSE was 3.9 (~23% error rate) (Chapter simple-linear-regression).

Model Comparison using Analysis of Variance (ANOVA)

- model1 <- lm(sales ~ youtube, data = marketing)
- model2 <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
- model3 <- lm(sales ~ youtube + facebook, data = marketing)
- model4 <- lm(sales ~ youtube + facebook + youtube*facebook, data = marketing)

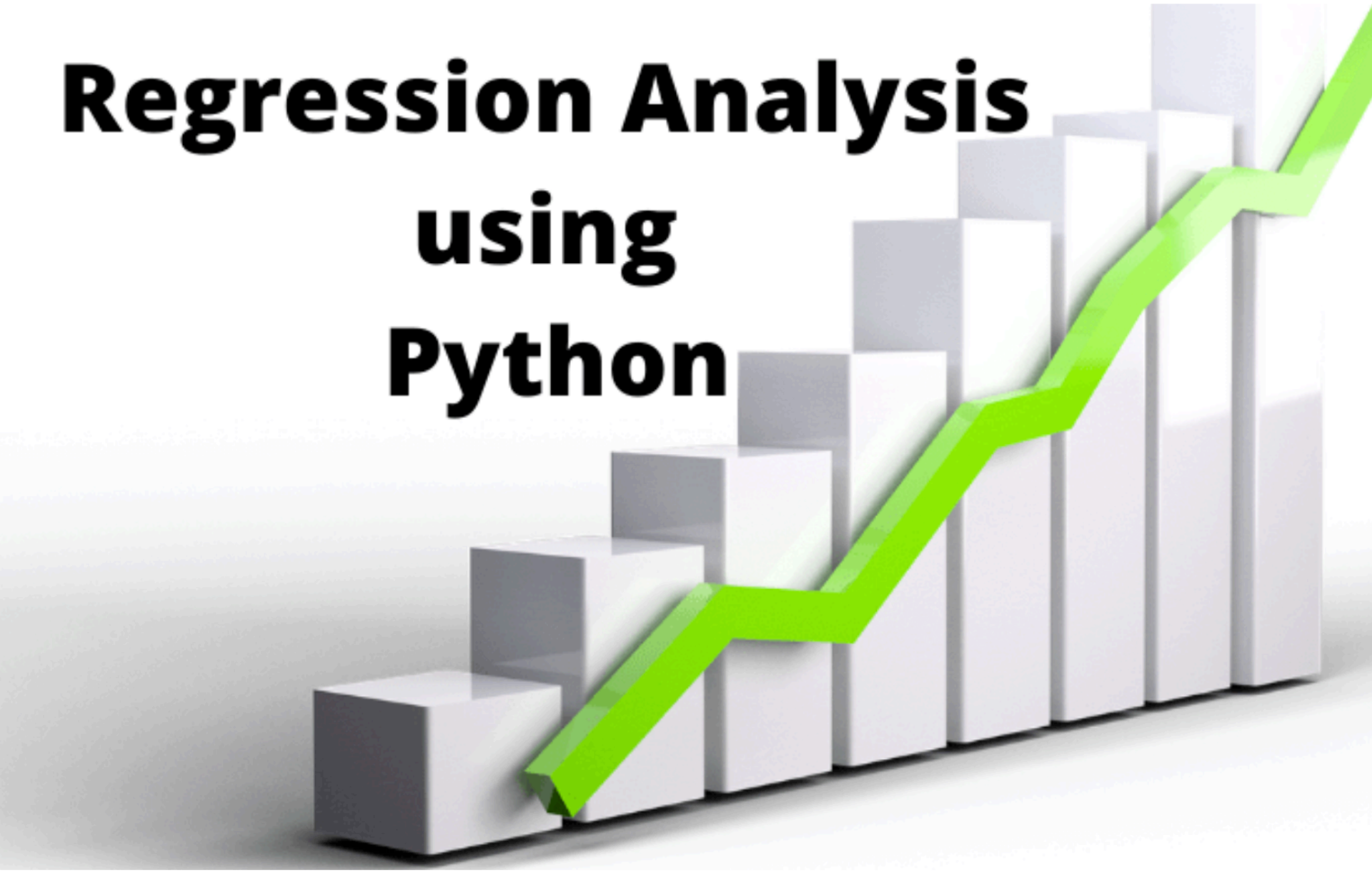
Hands-on Exercise

Using ANOVA to Compare Models

- model1 <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
- model2 <- lm(sales ~ youtube + facebook, data = marketing)
- model3 <- lm(sales ~ youtube, data = marketing)
- model4 <- lm(sales ~ youtube + facebook + youtube*facebook, data = marketing)

- anova(model1, model2)
- anova(model1, model3)
- anova(model1, model4)
- anova(model2, model3)
- anova(model2, model4)
- anova(model3, model4)

Regression Analysis using Python



Regression Analysis using Python

- https://realpython.com/linear-regression-in-python/#multiple-linear-regression

See also:

- https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9
- https://towardsdatascience.com/introduction-to-linear-regression-in-python-c12a072bedf0
- https://www.geeksforgeeks.org/linear-regression-python-implementation/

Non-linear regression using R

- Polynomial
  - Non-linear Least Squares
  - 2-degree simple polynomial
  - N-degree multiple polynomial
  - N-degree simple polynomial
- Logarithmic
- Splines
- Generalized Additive Regression
- Local Loess
- Logistic
- Other

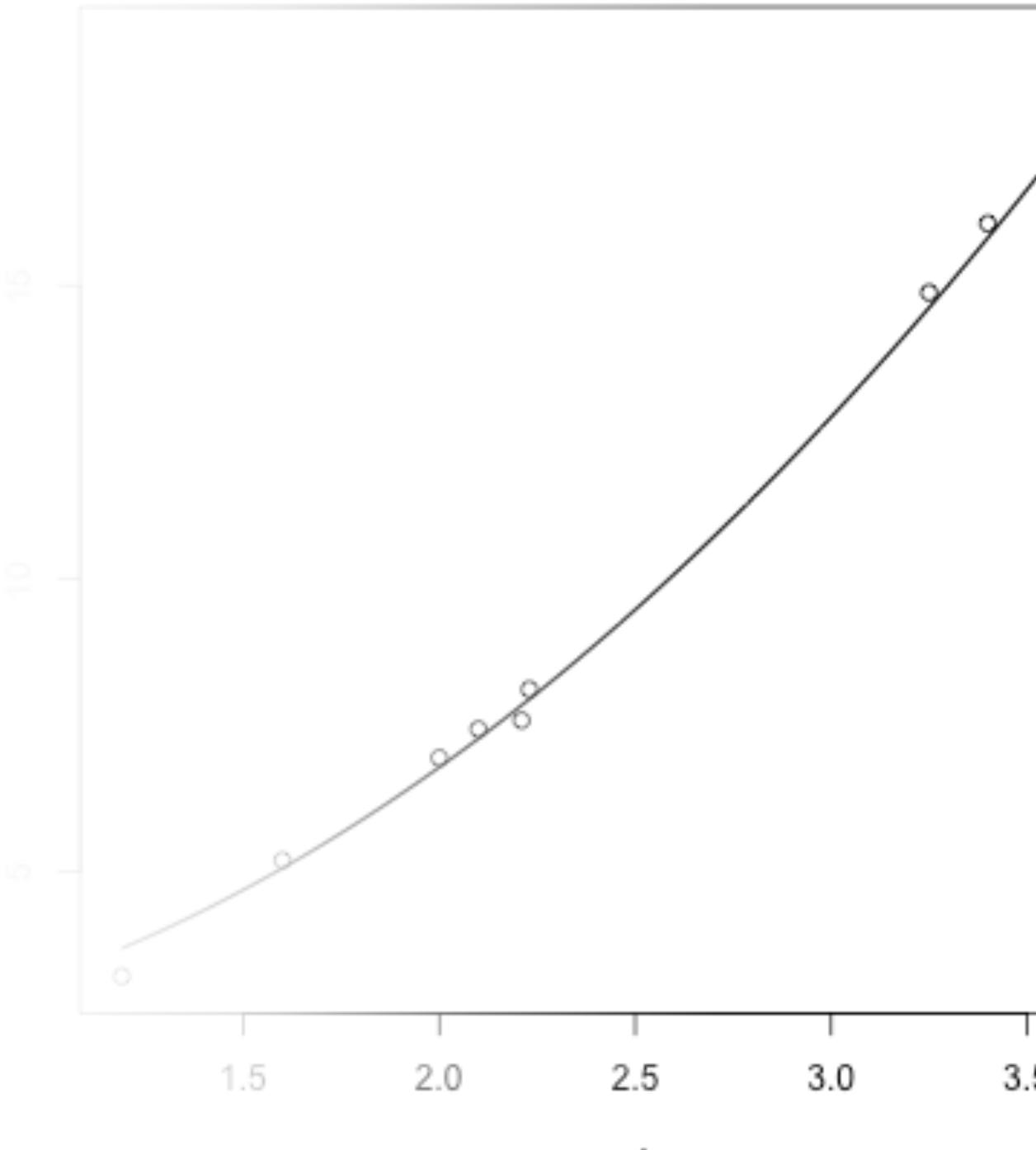
Hands-on Exercise Non-linear Least Square Model

- When modeling real world data for regression analysis, we observe that it is rarely the case that the equation of the model is a linear equation giving a linear graph
- Most of the time, the equation of the model of real-world data involves mathematical functions of higher degree like an exponent of 3 or a sin or exp function. In such a scenario, the plot of the model gives a curve rather than a line
- The goal of both linear and non-linear regression is to adjust the values of the model's parameters to find the line or curve that comes closest to your data
- On finding these values we will be able to estimate the response variable with good accuracy.
- In Non-linear Least Square regression, we establish a regression model in which the sum of the squares of the vertical distances of different points from the regression curve is minimized
- We generally start with a defined model and assume some values for the coefficients
- We then apply the **nls()** function of R to get the more accurate values along with the confidence intervals.

https://www.tutorialspoint.com/r/r_nonlinear_least_square.htm

Hands-on Exercise

- xvalues <- c(1.6,2.1,2,2.23,3.71,3.25,3.4,3.86,1.19,2.21)
- Yvalues<- c(5.19,7.43,6.94,8.11,18.75,14.88,16.06,19.12,3.21,7.58)
- png(file = "nls.png")
- model <- nls(yvalues ~ b1*xvalues^2+b2,start = list(b1 = 1,b2 = 3))
- new.data <- data.frame(xvalues = seq(min(xvalues),max(xvalues),len = 100))
  lines(new.data$xvalues,predict(model,newdata = new.data))
- dev.off()
- print(sum(resid(model)^2))
- print(confint(model))



Hands-on Exercise

The ”Boston” dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Massachusetts from the “MASS” package. The dataset is small in size with only 506 cases

There are **14** attributes in each case of the dataset:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in $1000's

https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html

https://cran.r-project.org/web/packages/MASS/MASS.pdf

https://rpubs.com/hoangh/boston

https://howilearnstatistics.com/post/data-analysis-with-r-boston-housing-dataset/

Boston Dataset Set dataset & Exploratory Data Analysis

https://howilearnstatistics.com/post/data-analysis-with-r-boston-housing-dataset/

```
Install.packages("MASS")
install.packages("corrplot")
library(MASS)

Boston
attach(Boston)
summary(Boston)
library(corrplot)
corr_matrix<-cor(Boston)
corrplot(corr_matrix, type="upper")
summary(crim)
require(ggplot2)
require(plotly)
plot_ly(data = Boston, x = ~lstat, y = ~crim)
plot_ly(data = Boston, x = ~tax, y = ~crim)
plot_ly(data=Boston, x = ~crim, type = "histogram")
```

Prediction with Boston Dataset Linear Regression

- any(**is**.na(Boston))
- data(Boston)
- smp_size<-floor(0.75*nrow(Boston))
- set.seed(12)
- smp_size<-floor(0.75*nrow(Boston))
- train_ind<-sample(seq_len(nrow(Boston)), size=smp_size)
- train<-Boston[train_ind,]
- test<-Boston[-train_ind,]
- lm.fit=lm(medv~lstat,data=train)
- summary(lm.fit)
- require(Metrics)
- evaluate<-predict(lm.fit, test)
- rmse(evaluate,test[,14])
- dat <- data.frame(**lstat** = (1:35), medv = predict(lm.fit, data.frame(**lstat** = (1:35))))
- plot_ly() %>% add_trace(**x**=~**lstat**, **y**=~medv, type="scatter",
- mode="lines", data = dat, name = "Predicted Value") %>%
- add_trace(**x**=~**lstat**, **y**=~medv, type="scatter", data = test, name = "Actual Value")

Prediction with Boston Dataset Non-linear Regression

- lm.fit=lm(medv~**lstat**+I(**lstat**^2),data=train) dat <- data.frame(**lstat** = (1:40),
- medv = predict(lm.fit, data.frame(**lstat** = (1:40))))

- plot_ly() %>% add_trace(**x**=~**lstat**, **y**=~medv, type="scatter", mode="lines",
- data = dat, name = "Predicted Value") %>%
- add_trace(**x**=~**lstat**, **y**=~medv, type="scatter", data = test, name = "Actual Value")

- **summary**(**lm**.fit)

- evaluate<-predict(lm.fit, test) rmse(evaluate,test[,14])
- rmse(evaluate,test[,14])

Boston Dataset
Multiple Non-linear Regression

- lm.fit=lm(medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+ black+
- **lstat**+I(**lstat**^2),data=train)

- summary(lm.fit)

- evaluate<-predict(lm.fit, test) rmse(evaluate,test[,14])

Second Exercise with Boston Dataset Multiple Non-linear Regression II

- library(dplyr)
- library(tidyverse)
- set.seed(123)
- **library**(caret)
- theme_set(theme_classic())
- set.seed(123)

- training.samples <- Boston$medv %>%
- createDataPartition(p = 0.8, list = FALSE)
- train.data <- Boston[training.samples,]
- test.data <- Boston[-training.samples,]

http://www.sthda.com/english/articles/40-regression-analysis/162-nonlinear-regression-essentials-in-r-polynomial-and-spline-regression-models/

Boston Dataset
Multiple Non-linear Regression II

- ggplot(train.data, aes(lstat, medv)) + geom_point() + stat_smooth()
- lm(medv ~ poly(lstat, 2, raw = TRUE), data = train.data)

- lm(medv ~ poly(lstat, 6, raw = TRUE), data = train.data) %>% summary()
- model <- lm(medv ~ poly(lstat, 5, raw = TRUE), data = train.data)
- predictions <- model %>% predict(test.data)
- data.frame(RMSE = RMSE(predictions, test.data$medv), R2 = R2(predictions, test.data$medv))
- ggplot(train.data, aes(lstat, medv)) + geom_point() +
- stat_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))

Boston Dataset
Logarithmic Regression

- model <- lm(medv ~ log(lstat), data = train.data)
- predictions <- model %>% predict(test.data)

- data.frame(RMSE = RMSE(predictions, test.data$medv), R2 = R2(predictions, test.data$medv))

- ggplot(train.data, aes(lstat, medv)) + geom_point() +
- stat_smooth(method = lm, formula = y ~ log(x))

Boston Dataset
Splines Regression

- Polynomial regression only captures a certain amount of curvature in a nonlinear relationship
- An alternative, and often superior, approach to modeling nonlinear relationships is to use splines
- Splines provide a way to smoothly interpolate between fixed points, called knots. Polynomial regression is computed between knots. In other words, splines are series of polynomial segments strung together, joining at knots
- The R package splines includes the function bs for creating a b-spline term in a regression model.
- You need to specify two parameters: the degree of the polynomial and the location of the knots
- In our example, we'll place the knots at the lower quartile, the median quartile, and the upper quartile:

Boston Dataset
Splines Regression

- **library**(splines)
- knots <- quantile(train.data$lstat, p = c(0.25, 0.5, 0.75))
- model <- lm (medv ~ bs(lstat, knots = knots), data = train.data)
- predictions <- model %>% predict(test.data)

- data.frame(RMSE = RMSE(predictions, test.data$medv), R2 = R2(predictions, test.data$medv))

- ggplot(train.data, aes(lstat, medv)) + geom_point() + stat_smooth(method = lm, formula = y ~ splines::bs(x, df = 3))

Boston Dataset
Generalized Additive Models

- Once you have detected a non-linear relationship in your data, the polynomial terms may not be flexible enough to capture the relationship, and spline terms require specifying the knots

- Generalized additive models, or GAM, are a technique to automatically fit a spline regression

- This can be done using the mgcv R package:

Boston Dataset
Generalized Additive Models

- **library**(mgcv)
- model <- gam(medv ~ s(lstat), data = train.data)

- data.frame(RMSE = RMSE(predictions, test.data$medv), R2 = R2(predictions, test.data$medv))

- ggplot(train.data, aes(lstat, medv)) + geom_point() + stat_smooth(method = gam, formula = y ~ s(x))

Boston Dataset
Comparing Models

- From analyzing the RMSE and the R2 metrics of the different models, we can see that:
  - the polynomial regression
  - the spline regression and
  - the generalized additive models
- outperform the linear regression model and the log transformation approaches.

Loess Regression

- *Loess* short for Local Regression is a non-parametric approach that fits multiple regressions in local neighborhood. This can be particularly resourceful, if you know that your *X* variables are bound within a range

- Loess regression can be applied using the loess() on a numerical vector to smoothen it and to predict the Y locally (i.e, within the trained values of *Xs*)

- The size of the neighborhood can be controlled using the span argument, which ranges between 0 to 1. It controls the degree of smoothing. So, the greater the value of span, more smooth is the fitted curve

- The predictor variable can just be indices from 1 to number of observations in the absence of explanatory variables. If other explanatory variables are available, they can be used as well (maximum of 4)

Hands-on Exercise

http://r-statistics.co/Loess-Regression-With-R.html

- data(economics, package="ggplot2")
- economics$index <- 1:nrow(economics)
- loessMod10 <- loess(uempmed ~ index, data=economics, span=0.10) # 10% smoothing span
- loessMod25 <- loess(uempmed ~ index, data=economics, span=0.25)
- loessMod50 <- loess(uempmed ~ index, data=economics, span=0.50)
- smoothed10 <- predict(loessMod10)
- smoothed25 <- predict(loessMod25)
- smoothed50 <- predict(loessMod50)
- plot(economics$uempmed, x=economics$date, type="l", main="Loess Smoothing and Prediction", xlab="Date", ylab="Unemployment (Median)")
- lines(smoothed10, x=economics$date, col="red")
- lines(smoothed25, x=economics$date, col="green")
- lines(smoothed50, x=economics$date, col="blue")

Hands-on Exercise

- calcSSE <- function(x){ loessMod <- **try**(**loess**(uempmed ~ index, data=economics, span=x), silent=T) res <- **try**(loessMod$residuals, silent=T) if(**class**(res)!="try-error"){ if((**sum**(res, na.rm=T) > 0)){ sse <- **sum**(res^2) } }else{ sse <- 99999 } **return**(sse) }

- **optim**(par=**c**(0.5), calcSSE, method="SANN")

An Example

Winter Olympics Medal Tally

Rank	Country	Gold ▼	Silver	Bronze	Total
1	Russian Fed.	13	11	9	**33**
2	Norway	11	5	10	**26**
3	Canada	10	10	5	**25**
4	United States	9	7	12	**28**
5	Netherlands	8	7	9	**24**
6	Germany	8	6	5	**19**
7	Switzerland	6	3	2	**11**
8	Belarus	5	0	1	**6**
9	Austria	4	8	5	**17**
10	France	4	4	7	**15**
11	Poland	4	1	1	**6**
12	China	3	4	2	**9**
13	Korea	3	3	2	**8**

Y variable: Number of medals

X variables: Latitude
Average elevation
Log population

An Example

The Winter Olympics!

Can we infer a relationship between

Number of medals won by a country → AND

1. The country's latitude
2. The country's average elevation
3. The country's population

$$number\ of\ medals_i = \beta_0 + \beta_1(latitude_i) + \beta_2(elevation_i) + \beta_3(\log population_i)$$

https://www.youtube.com/watch?v=VvlqA-iO2HA

An Example

What is "significance"?

- Start with hypothesis that the gradient is 0 (ie. there is no relationship)

$$H0: \beta = 0$$

- Use a sample to see if there is enough evidence to reject this null hypothesis.

If so, we can infer:

$$H1: \beta \neq 0$$

(ie. we infer that the variable is significant!)

The ANOVA section

$$\widehat{number\ of\ medals}_i = b_0 + b_1(latitude_i) + b_2(elevation_i) + b_3(\log population_i)$$

Source	SS	df	MS
Model	439.274821	3	146.42494
Residual	954.485179	21	45.4516752
Total	1393.76	24	58.0733333

Number of obs	= 25
F(3, 21)	= 3.22
Prob > F	= 0.0434
R-squared	= 0.3152
Adj R-squared	= 0.2173
Root MSE	= 6.7418

totalmedal	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
cen_lat	.522752	.1889091	2.77	0.012	.129894	.9156099
elev	.003171	.0038126	0.83	0.415	-.0047577	.0110996
logpop	2.146452	.9968635	2.15	0.043	.0733606	4.219543
_cons	-54.52767	21.97521	-2.48	0.022	-100.2276	-8.827715

The Variables section

totalmedal	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
cen_lat	.522752	.1889091	2.77	0.012	.129894	.9156099
elev	.003171	.0038126	0.83	0.415	-.0047577	.0110996
logpop	2.146452	.9968635	2.15	0.043	.0733606	4.219543
_cons	-54.52767	21.97521	-2.48	0.022	-100.2276	-8.827715

$$\widehat{number\ of\ medals}_i = -54.528 + 0.523(latitude_i) + 0.003(elevation_i) + 2.146(\log population_i)$$

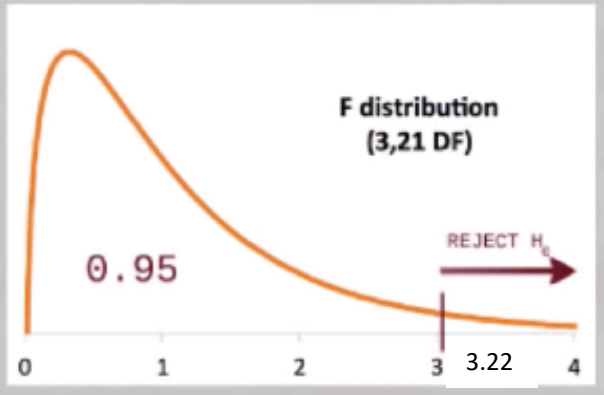
Interpretation

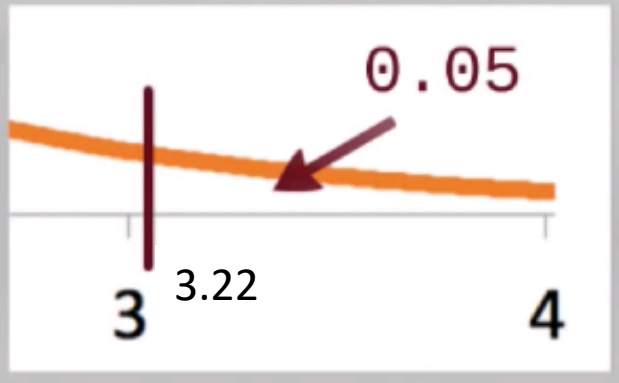
For every additional degree of latitude, the expected number of medals increases by 0.523 on average, holding all other variables constant.

For every additional metre of average elevation, the expected number of medals increases by 0.003 on average, holding all other variables constant.

The ANOVA section

Number of obs	=	25
F(3, 21)	=	3.22
Prob > F	=	0.0434
R-squared	=	0.3152
Adj R-squared	=	0.2173
Root MSE	=	6.7418



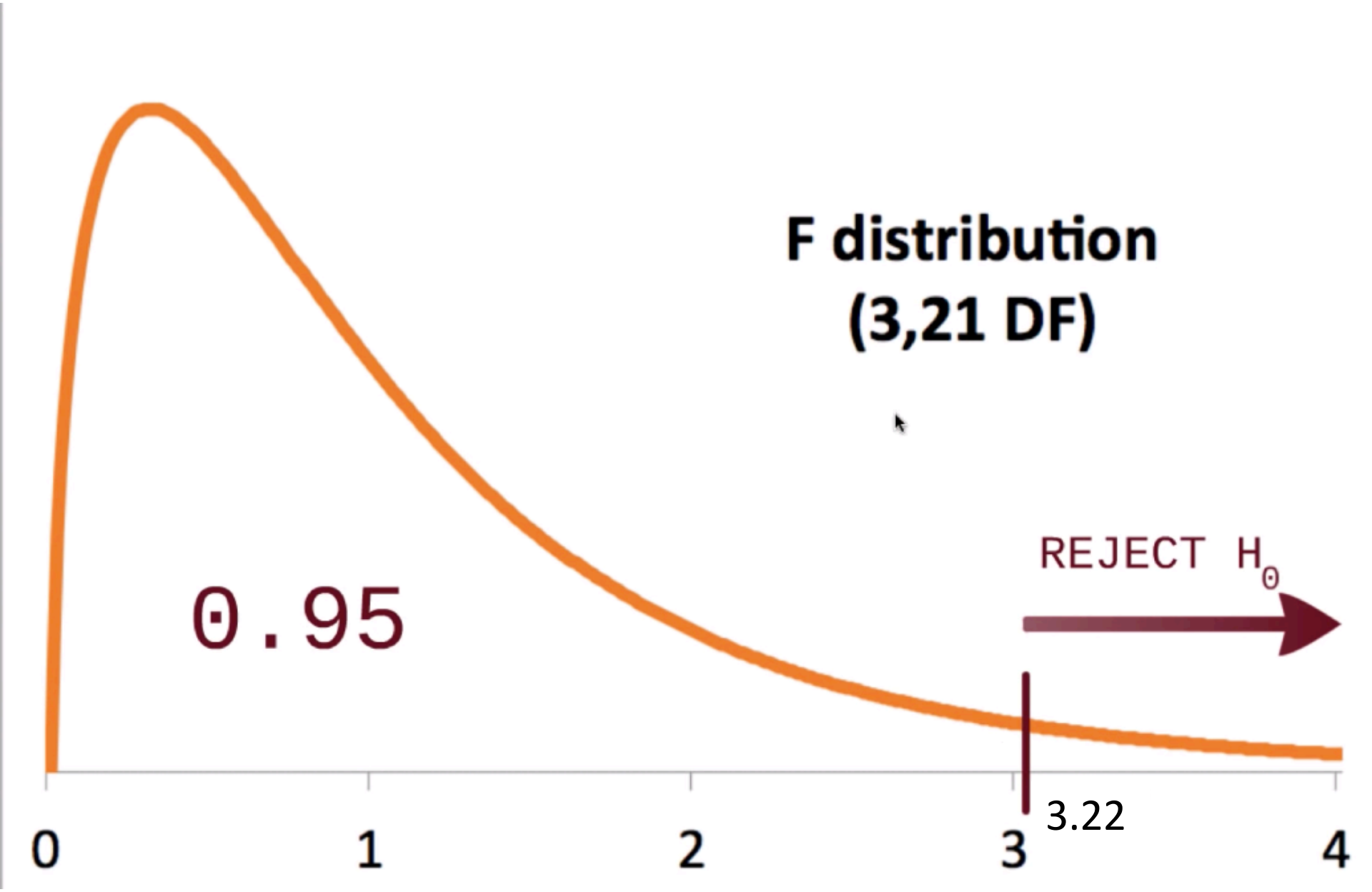


Is this model with 3 explanatory variables better than a model with 0 explanatory variables?

Prob > F = 0.0434

At 10% -> YES!
At 5% -> YES!
At 1% -> NO!

There is a 4.34% probability that the improvements we are seeing with our 3 variable model is due to random chance alone.


F distribution
(3,21 DF)
0.95
REJECT H_0
3.22
0
1
2
3
4

The Variables section

totalmedal	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
cen_lat	.522752	.1889091	2.77	0.012	.129894	.9156099
elev	.003171	.0038126	0.83	0.415	-.0047577	.0110996
logpop	2.146452	.9968635	2.15	0.043	.0733606	4.219543
_cons	-54.52767	21.97521	-2.48	0.022	-100.2276	-8.827715

$$\widehat{number\ of\ medals}_i = -54.528 + 0.523(latitude_i) + 0.003(elevation_i) + 2.146(\log population_i)$$

Estimate for Netherlands:

Latitude: 52.2 Elevation: 30.1m Pop: 16,500,000

Log pop: 16.62

$$\widehat{number\ of\ medals}_{NED} = -54.528 + 0.523(52.2) + 0.003(30.1) + 2.146(16.6) = 8.557$$

Error(NED) = 24 - 8.6 = +15.4

The Variables section

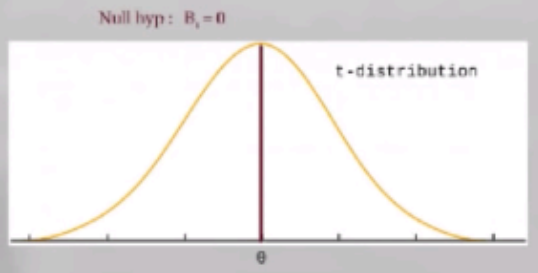
totalmedal	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
cen_lat	.522752	.1889091	2.77	0.012	.129894	.9156099
elev	.003171	.0038126	0.83	0.415	-.0047577	.0110996
logpop	2.146452	.9968635	2.15	0.043	.0733606	4.219543
_cons	-54.52767	21.97521	-2.48	0.022	-100.2276	-8.827715

Latitude -> $t_1 = b_1/SE_1$

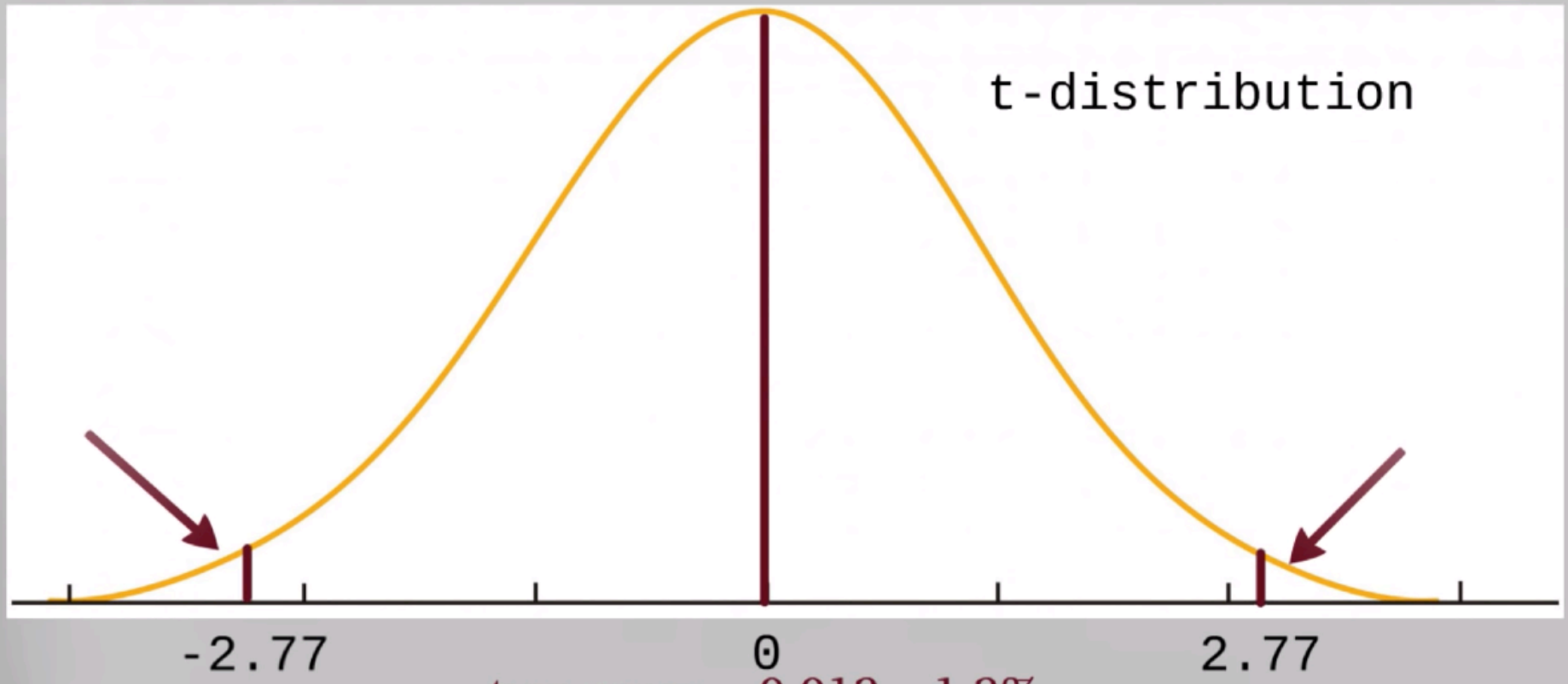
$= 0.522/0.189$

$= 2.77$

$p_1 =$



Null hyp : $B_1 = 0$ $b_1 = 0.522$ $t_1 = 2.77$



two areas = 0.012 = 1.2%

If the null hypothesis is true ($B_1 = 0$), the chance of us getting a sample AS extreme as we did ($b_1 = 0.522$), is 1.2%

Prezi

The Variables section

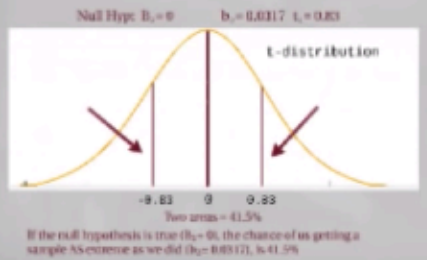
totalmedal	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
cen_lat	.522752	.1889091	2.77	0.012	.129894	.9156099
elev	.003171	.0038126	0.83	0.415	-.0047577	.0110996
logpop	2.146452	.9968635	2.15	0.043	.0733606	4.219543
_cons	-54.52767	21.97521	-2.48	0.022	-100.2276	-8.827715

Elevation -> $t_2 = b_2/SE_2$

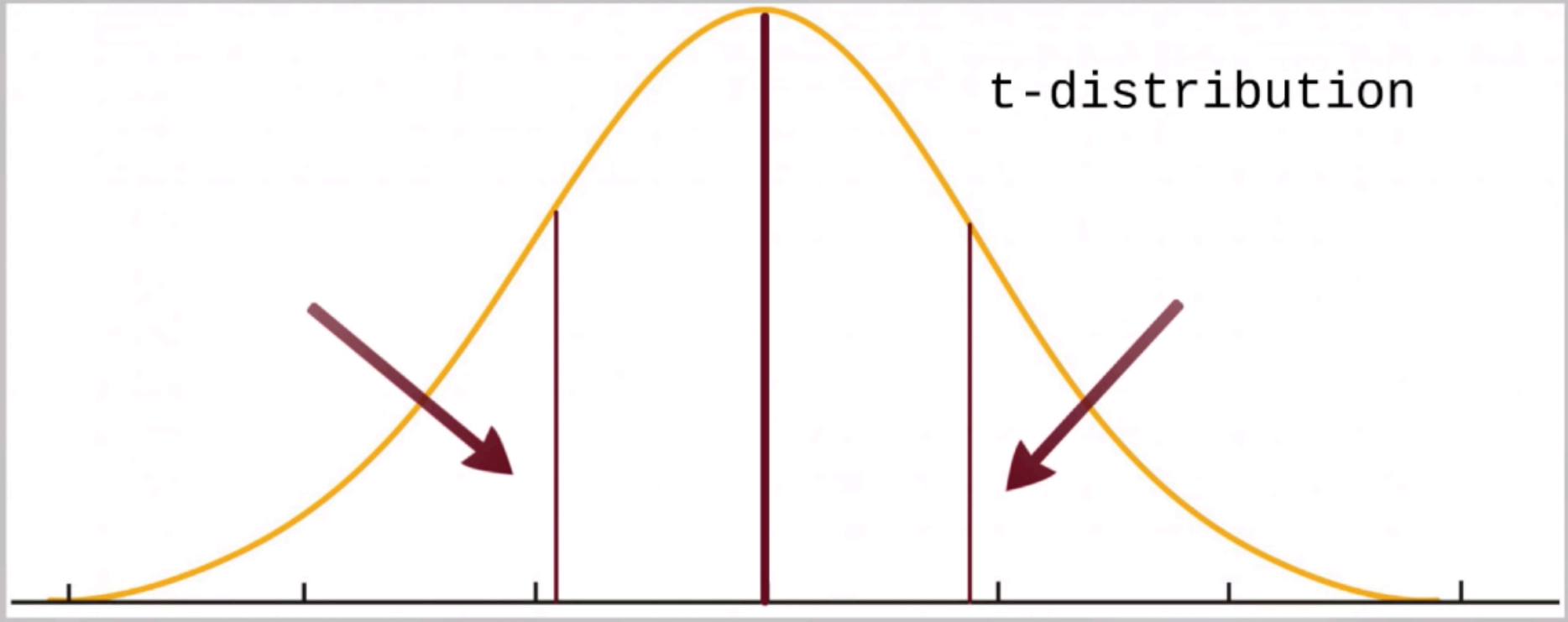
$= 0.0317/0.0038$

$= 0.83$

$p_2 =$



Null Hyp: $B_2 = 0$ $b_2 = 0.0317$ $t_2 = 0.83$



Two areas = 41.5%

If the null hypothesis is true ($B_2 = 0$), the chance of us getting a sample AS extreme as we did ($b_2 = 0.0317$), is 41.5%

Francisco J. Cantú-Ortiz, PhD

Professor of Computer Science and Artificial Intelligence
Tecnológico de Monterrey
Enago-Academy Advisor for Strategic Alliances

E-mail: fcantu@tec.mx, fjcantor@gmail.com
Cel: +52 81 1050 8294, SNI-2 CVU: 9804
Personal Page: http://semtech.mty.itesm.mx/fcantu/
Facebook: fcantu; Twitter: @fjcantor; Skype: fjcantor
Orcid: 0000-0002-2015-0562
Scopus ID:6701563520
Researcher ID: B-8457-2009
https://www.researchgate.net/profile/Francisco_Cantu-Ortiz
https://scholar.google.com.mx/citations?hl=es&user=45-uuK4AAAAJ
https://itesm.academia.edu/FranciscoJavierCantuOrtiz
Ave. Eugenio Garza Sada No. 2501, Monterrey N.L., C.P. 64849, México