

# Course 02402 Introduction to Statistics

## Lecture 9: Multiple linear regression

DTU Compute  
Technical University of Denmark  
2800 Lyngby – Denmark

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

# Agenda

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

## Example: Ozone concentration

We have a set of observations of: logarithm to ozone concentration ( $\log(\text{ppm})$ ), temperature, radiation and wind speed:

ozone	radiation	wind	temperature	month	day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
18	131	8.0	76	9	29
20	223	11.5	68	9	30

# Example: Ozone concentration

```

## Se info about data
?airquality
## Copy the data
Air <- airquality
## Remove rows with at least one NA value
Air <- na.omit(Air)

## Remove one outlier
Air <- Air[-which(Air$Ozone == 1), ]

## Check the empirical density
hist(Air$Ozone, probability=TRUE, xlab="Ozon", main="")

## Concentrations are positive and very skewed, let's
## log-transform right away:
## (although really one could wait and check residuals from models)
Air$logOzone <- log(Air$Ozone)
## Bedre epåf?
hist(Air$logOzone, probability=TRUE, xlab="log Ozone", main="")

## Make a time variable (R timeclass, se ?POSIXct)
Air$t <- ISOdate(1973, Air$Month, Air$Day)
## Keep only some of the columns
Air <- Air[,c(7,4,3,2,8)]
## New names of the columns
names(Air) <- c("logOzone", "temperature", "wind", "radiation", "t")

## What's in Air?
str(Air)
Air
head(Air)
tail(Air)

## Typically one would begin with a pairs plot
pairs(Air, panel = panel.smooth, main = "airquality data")

```

## Example: Ozone concentration

- Let us first analyse the relation between ozone and temperature
- Apply a *simple linear regressions model*

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad , \quad \varepsilon_i \sim N(0, \sigma^2) \text{ og i.i.d.}$$

where

- $Y_i$  is the (logarithm of) ozone concentration of observation  $i$
- $x_i$  is the temperature at observation  $i$

# Fit the model in R

```
#####  
  
## See the relation between ozone and temperature  
plot(Air$temperature, Air$logOzone, xlab="Temperature", ylab="Ozon")  
  
## Correlation  
cor(Air$logOzone, Air$temperature)  
  
## Fit a simple linear regression model  
summary(lm(logOzone ~ temperature, data=Air))  
  
## Add a vector with random values, is there a significant linear relation?  
## ONLY for ILLUSTRATION purposes  
Air$noise <- rnorm(nrow(Air))  
plot(Air$logOzone, Air$noise, xlab="Noise", ylab="Ozon")  
cor(Air$logOzone, Air$noise)  
summary(lm(logOzone ~ noise, data=Air))
```



## Simple linear regression model for the other two

We can also make a simple linear regression model with each of the other two independent variables

```
#####
## With each of the other two independent variables

## Simple linear regression model with the wind speed
plot(Air$logOzone, Air$wind, xlab="logOzone", ylab="Wind speed")
cor(Air$logOzone, Air$wind)
summary(lm(logOzone ~ wind, data=Air))

## Simple linear regression model with the radiation
plot(Air$logOzone, Air$radiation, xlab="logOzone", ylab="Radiation")
cor(Air$logOzone, Air$radiation)
summary(lm(logOzone ~ radiation, data=Air))
```

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression**
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

# Multiple linear regression

- $Y$  is the *dependent variable*
- We are interested in modelling the  $Y$ 's dependency of the *independent* or *explanatory* variables  $x_1, x_2, \dots, x_p$
- We are modelling a *linear relation* between  $Y$  and  $x_1, x_2, \dots, x_p$ , described with the regression model

$$Y_i = \beta_0 + \beta_1 x_{1,i} + \dots + \beta_p x_{p,i} + \varepsilon_i \quad , \quad \varepsilon_i \sim N(0, \sigma^2) \text{ and i.i.d.}$$

- $Y_i$  og  $\varepsilon_i$  are random variables and  $x_{j,i}$  are variables

## Least squares estimates

- The coefficient estimates are found by minimizing:

$$RSS(\beta_0, \beta_1, \dots, \beta_p) = \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_{i,1} + \dots + \beta_p x_{i,p})]^2$$

- The "predicted" (= "fitted") are found as

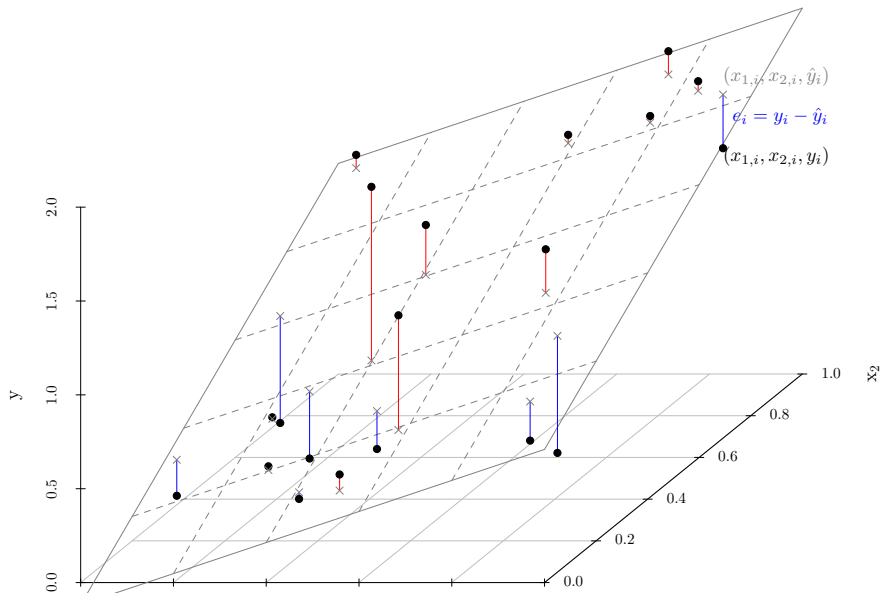
$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \dots + \hat{\beta}_p x_{i,p}$$

- And then the residuals are found as

$$e_i = y_i - \hat{y}_i$$

residual = observation – prediction

# Least squares estimates - The concept!



## Computations for MLR - no explicit formulas given!

- Remark 6.6: Extract  $\hat{\beta}_i$  and  $\hat{\sigma}_{\beta_i}$  from R-output (`summary(myfit)`)

## Computations for MLR - no explicit formulas given!

- Remark 6.6: Extract  $\hat{\beta}_i$  and  $\hat{\sigma}_{\beta_i}$  from R-output (`summary(myfit)`)
- Theorem 6.2: The t-distribution can be used for inference for parameters

## Computations for MLR - no explicit formulas given!

- Remark 6.6: Extract  $\hat{\beta}_i$  and  $\hat{\sigma}_{\beta_i}$  from R-output (`summary(myfit)`)
- Theorem 6.2: The t-distribution can be used for inference for parameters
- Methods 6.4 and 6.5: Hypothesis tests and Confidence intervals for parameters based on R-output.



## Computations for MLR - no explicit formulas given!

- Remark 6.6: Extract  $\hat{\beta}_i$  and  $\hat{\sigma}_{\beta_i}$  from R-output (`summary(myfit)`)
- Theorem 6.2: The t-distribution can be used for inference for parameters
- Methods 6.4 and 6.5: Hypothesis tests and Confidence intervals for parameters based on R-output.
- Everything: **THE SAME** as for **SIMPLE linear regression!**

## Computations for MLR - no explicit formulas given!

- Remark 6.6: Extract  $\hat{\beta}_i$  and  $\hat{\sigma}_{\beta_i}$  from R-output (`summary(myfit)`)
- Theorem 6.2: The t-distribution can be used for inference for parameters
- Methods 6.4 and 6.5: Hypothesis tests and Confidence intervals for parameters based on R-output.
- Everything: **THE SAME** as for **SIMPLE linear regression!**
- (In Section 6.6: Mathematical matrix based expressions including explicit formulas. Not syllabus in course 02402)

## Parameter interpretation in MLR (Remark 6.14)

What does  $\hat{\beta}_i$  express?

- The expected  $y$ -change with 1 unit  $x_i$ -change

## Parameter interpretation in MLR (Remark 6.14)

What does  $\hat{\beta}_i$  express?

- The expected  $y$ -change with 1 unit  $x_i$ -change
- The effect of  $x_i$  given the other variables
- The effect of  $x_i$  corrected for the other variables
- The effect of  $x_i$  "other variables being equal"
- The unique effect of  $x_i$

## Parameter interpretation in MLR (Remark 6.14)

What does  $\hat{\beta}_i$  express?

- The expected  $y$ -change with 1 unit  $x_i$ -change
- The effect of  $x_i$  given the other variables
- The effect of  $x_i$  corrected for the other variables
- The effect of  $x_i$  "other variables being equal"
- The unique effect of  $x_i$
- Depends on what else is in the model!!

## Parameter interpretation in MLR (Remark 6.14)

What does  $\hat{\beta}_i$  express?

- The expected  $y$ -change with 1 unit  $x_i$ -change
- The effect of  $x_i$  given the other variables
- The effect of  $x_i$  corrected for the other variables
- The effect of  $x_i$  "other variables being equal"
- The unique effect of  $x_i$
- Depends on what else is in the model!!
- Generally: NOT a causal/intervention effect!!

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection**
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

## Extend the model (forward selection)

- *Not included in the eNote*
- Start with the *linear regression model* with the most significant independent variable
- *Extend the model* with the remaining independent variables (inputs) one at a time
- *Stop* when there is not any significant extensions possible

```
#####  
## Extend the model  
  
## Forward selection:  
## Add wind to the model  
summary(lm(logOzone ~ temperature + wind, data=Air))  
## Add radiation to the model  
summary(lm(logOzone ~ temperature + wind + radiation, data=Air))
```



## Reduce the model (model reduction or backward selection)

- *Described in the eNote, section 6.5*
- Start with the full model
- Remove the most insignificant independent variable
- Stop when all prm. estimates are significant

```
#####  
## Backward selection  
  
## Fit the full model  
summary(lm(logOzone ~ temperature + wind + radiation + noise, data=Air))  
## Remove the most non-significant input, are all now significant?  
summary(lm(logOzone ~ temperature + wind + radiation, data=Air))
```

# Model selection

- There is no fully certain method for finding the best model!
- It will require subjective decisions to select a model
- Different procedures: either forward or backward selection (or both), depends on the circumstances
- Statistical measures and tests to compare model fits
- In this course only backward selection is described

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)**
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

# Residual analysis (model validation)

- Model validation: Analyze the residuals to check that the assumptions is met
- $e_i \sim N(0, \sigma^2)$  is independent and identically distributed (i.i.d.)
- Same as for the simple linear regression model

# Assumption of normal distributed residuals

- Make a qq-normalplot (normal score plot) to see if they seem normal distributed

```
#####  
## Assumption of normal distributed residuals  
  
## Save the selected fit  
fitSel <- lm(logOzone ~ temperature + wind + radiation, data=Air)  
  
## qq-normalplot  
qqnorm(fitSel$residuals)  
qqline(fitSel$residuals)
```

# Assumption of identical distribution of residuals

- Plot the residuals ( $e_i$ ) versus the predicted (fitted) values ( $\hat{y}_i$ )

```
#####
## Plot the residuals vs. predicted values

plot(fitSel$fitted.values, fitSel$residuals, xlab="Predicted values",
      ylab="Residuals")
```

- Seems like the model can be improved!
- Plot the residuals vs. the independent variables

```
#####
## Plot the residuals vs. the independent variables

par(mfrow=c(1,3))
plot(Air$temperature, fitSel$residuals, xlab="Temperature")
plot(Air$wind, fitSel$residuals, xlab="Wind speed")
plot(Air$radiation, fitSel$residuals, xlab="Radiation")
```

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity**
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

## Curvilinear model

If we want to estimate a model of the type

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$$

we can use a multiple linear regression model

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \varepsilon_i$$

where

- $x_{i,1} = x_i$
- $x_{i,2} = x_i^2$

and apply the same methods as for multiple linear regression.



# Extend the ozone model with appropriate curvilinear regression

```
#####
## Extend the ozone model with appropriate curvilinear regression

## Make the squared wind speed
Air$windSq <- Air$wind^2
## Add it to the model
fitWindSq <- lm(logOzone ~ temperature + wind + windSq + radiation, data=Air)
summary(fitWindSq)

## Equivalently for the temperature
Air$temperature2 <- Air$temperature^2
## Add it
fitTemperatureSq <- lm(logOzone ~ temperature + temperature2 + wind + radiation, data=Air)
summary(fitTemperatureSq)

## Equivalently for the radiation
Air$radiation2 <- Air$radiation^2
## Add it
fitRadiationSq <- lm(logOzone ~ temperature + wind + radiation + radiation2, data=Air)
summary(fitRadiationSq)

## Which one was best?
## One could try to extend the model further
fitWindSqTemperaturSq <- lm(logOzone ~ temperature + temperature2 + wind + windSq + radiation, data=Air)
summary(fitWindSqTemperaturSq)

## Model validation
qqnorm(fitWindSq$residuals)
qqline(fitWindSq$residuals)
plot(fitWindSq$residuals, fitWindSq$fitted.values, pch=19)
```

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals**
- 7 Colinearity
- 8 The overall regression method

# Confidence and prediction intervals for the plane, Method 6.9:

Extract Confidence and prediction intervals for the plane by R-function predict. Options for confidence og prediction exist.

```
#####
## Confidence and prediction intervals for the curvilinear model

## Generate a new data.frame with constant temperature and radiation, but with varying wind speed
wind<-seq(1,20.3,by=0.1)
AirForPred <- data.frame(temperature=mean(Air$temperature), wind=wind,
                          windSq=wind^2, radiation=mean(Air$radiation))

## Calculate confidence and prediction intervals (actually bands)
CI <- predict(fitWindSq, newdata=AirForPred, interval="confidence", level=0.95)
PI <- predict(fitWindSq, newdata=AirForPred, interval="prediction", level=0.95)

## Plot them
plot(wind, CI[, "fit"], ylim=range(CI,PI), type="l",
     main=paste("At temperature =",format(mean(Air$temperature),digits=3),
               "and radiation =", format(mean(Air$radiation),digits=3)))
lines(wind, CI[, "lwr"], lty=2, col=2)
lines(wind, CI[, "upr"], lty=2, col=2)
lines(wind, PI[, "lwr"], lty=2, col=3)
lines(wind, PI[, "upr"], lty=2, col=3)
## legend
legend("topright", c("Prediction","95% confidence band","95% prediction band"), lty=c(1,2,2), col=1:3)
```

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity**
- 8 The overall regression method

# Colinearity

- MLR breaks down if X-data has "exact linear redundancy"
  - Example: Both height in cm and height in m is in the data.

# Colinearity

- MLR breaks down if X-data has "exact linear redundancy"
  - Example: Both height in cm and height in m is in the data.
- Interpretation and model stability is challenged if X-data has "near redundancy" patterns
  - Example: Both weight and BMI are in the X-data (highly correlated)

# Colinearity

- MLR breaks down if X-data has "exact linear redundancy"
  - Example: Both height in cm and height in m is in the data.
- Interpretation and model stability is challenged if X-data has "near redundancy" patterns
  - Example: Both weight and BMI are in the X-data (highly correlated)
- With e.g. two highly correlated  $x$ -variables:
  - Together in the model for  $y$  none of them may have a unique effect
  - Separately they may have a strong effect each of them

# Colinearity - an illustration in R

```
#####
## See problems with highly correlated inputs

## Generate values for MLR
n <- 100
## First variable
x1 <- sin(0:(n-1)/(n-1)*2*2*pi) + rnorm(n, 0, 0.1)
plot(x1, type="b")
## The second variable is the first plus a little noise
x2 <- x1 + rnorm(n, 0, 0.1)
## x1 and x2 are highly correlated
plot(x1,x2)
cor(x1,x2)
## Simulate an MLR
beta0=20; beta1=1; beta2=1; sigma=1
y <- beta0 + beta1 * x1 + beta2 * x2 + rnorm(n,0,sigma)
## See scatter plots for y vs. x1, and y vs. x2
par(mfrow=c(1,2))
plot(x1,y)
plot(x2,y)
## Fit an MLR
summary(lm(y ~ x1 + x2))

## If it was an experiment and the effects could be separated in the design
x1[1:(n/2)] <- 0
x2[(n/2):n] <- 0
## Plot them
plot(x1, type="b")
lines(x2, type="b", col="red")
## Now very low correlation
cor(x1,x2)
## Simulate MLR again
y <- beta0 + beta1 * x1 + beta2 * x2 + rnorm(n,0,sigma)
## and fit MLR
summary(lm(y ~ x1 + x2))
```

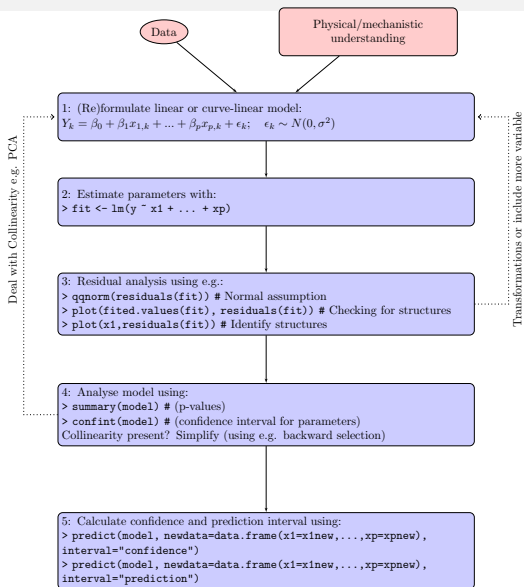


It is important how experiments are designed!

# Overview

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method

# The overall regression method box 6.16



# Agenda

- 1 Warm up with some simple linear regression
- 2 Multiple linear regression
- 3 Model selection
- 4 Residual analysis (model validation)
- 5 Curvilinearity
- 6 Confidence and prediction intervals
- 7 Colinearity
- 8 The overall regression method