Statistical Theory and Modeling (ST2601)
Lecture 12 - Nonlinear regression, Regularization and Time Series

Mattias Villani

Department of Statistics
Stockholm University

Overview

- **Non-linear regression**
- **Regularization**
- **Time series, dependence and autocorrelation**

Polynomial regression

- **Polynomial regression** of degree/order p

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_p x^p + \varepsilon, \quad \varepsilon \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$$

- **Nonlinear in x**
- **Linear in $\beta_0, \beta_1, \ldots, \beta_p$**
- Polynomial regression is just a linear regression with features:
  - $x_1 = x$
  - $x_2 = x^2$
  - $\vdots$
  - $x_p = x^p$
- Can use **least squares estimate** for the model

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$$

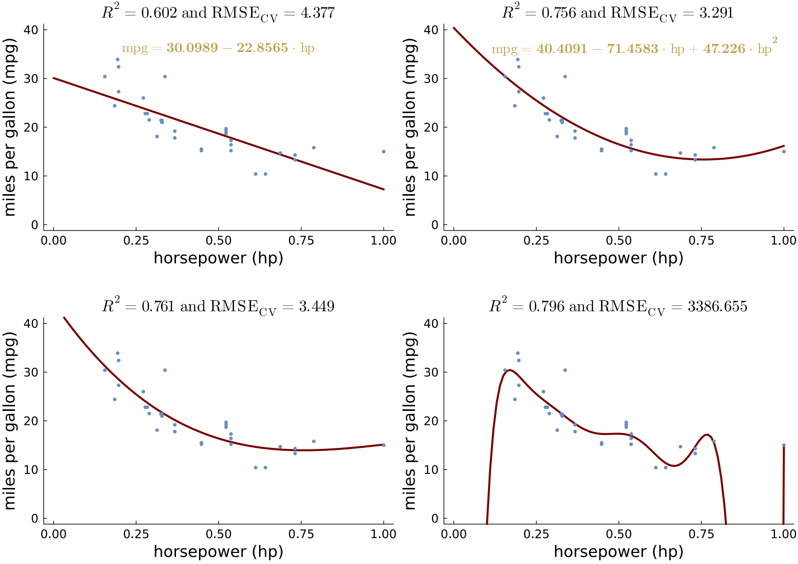
where the covariate/feature vector has $p + 1$ elements

$$\boldsymbol{x}_i = (1, x_i, x_i^2, \ldots, x_i^p)^\top$$

Polynomial regression data setup

	A	B	C	D	E	F
1		mpg (y)	hp (x)	x^2	x^3	x^4
2	Mazda RX4	21.000	0.328	0.108	0.035	0.012
3	Mazda RX4 Wag	21.000	0.328	0.108	0.035	0.012
4	Datsun 710	22.800	0.278	0.077	0.021	0.006
5	Hornet 4 Drive	21.400	0.328	0.108	0.035	0.012
6	Hornet Sportabo	18.700	0.522	0.273	0.143	0.074
7	Valiant	18.100	0.313	0.098	0.031	0.010
8	Duster 360	14.300	0.731	0.535	0.391	0.286
9	Merc 240D	24.400	0.185	0.034	0.006	0.001
10	Merc 230	22.800	0.284	0.080	0.023	0.006
11	Merc 280	19.200	0.367	0.135	0.049	0.018
12	Merc 280C	17.800	0.367	0.135	0.049	0.018
13	Merc 450SE	16.400	0.537	0.289	0.155	0.083
14	Merc 450SL	17.300	0.537	0.289	0.155	0.083
15	Merc 450SLC	15.200	0.537	0.289	0.155	0.083
16	Cadillac Fleetwo	10.400	0.612	0.374	0.229	0.140
17	Lincoln Contine	10.400	0.642	0.412	0.264	0.170
18	Chrysler Imperia	14.700	0.687	0.471	0.324	0.222
19	Fiat 128	32.400	0.197	0.039	0.008	0.002
20	Honda Civic	30.400	0.155	0.024	0.004	0.001
21	Toyota Corolla	33.900	0.194	0.038	0.007	0.001
22	Toyota Corona	21.500	0.290	0.084	0.024	0.007
23	Dodge Challeng	15.500	0.448	0.200	0.090	0.040
24	AMC Javelin	15.200	0.448	0.200	0.090	0.040
25	Camaro Z28	13.300	0.731	0.535	0.391	0.286
26	Pontiac Firebird	19.200	0.522	0.273	0.143	0.074
27	Fiat X1-9	27.300	0.197	0.039	0.008	0.002
28	Porsche 914-2	26.000	0.272	0.074	0.020	0.005
29	Lotus Europa	30.400	0.337	0.114	0.038	0.013
30	Ford Pantera L	15.800	0.788	0.621	0.489	0.386
31	Ferrari Dino	19.700	0.522	0.273	0.143	0.074

Polynomial regression for mtcars data



K-fold cross-validation

Fold 1

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

Fold 2

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

Fold 3

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

Fold 4

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

- Fold k:
  - Index for **test observations** in fold k: $\mathcal{T}_k$.
  - Model is fitted to **training data** in fold k
  - Predictions $\hat{y}_i^{(k)}$ for test data $i \in \mathcal{T}_k$.

K-fold cross-validation

Fold 1

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

Fold 2

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

Fold 3

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

Fold 4

biltyp	mpg	hp
Hornet Sportabout	18.7	0.52
Fiat X1-9	27.3	0.20
Merc 450SL	17.3	0.54
Merc 450SLC	15.2	0.54
Merc 240D	24.4	0.19
Duster 360	14.3	0.73
Datsun 710	22.8	0.28
Ferrari Dino	19.7	0.52
Ford Pantera L	15.8	0.79
Pontiac Firebird	19.2	0.52
Toyota Corona	21.5	0.29
AMC Javelin	15.2	0.45
Camaro Z28	13.3	0.73
Fiat 128	32.4	0.20
Merc 280C	17.8	0.37
Lotus Europa	30.4	0.34
Cadillac Fleetwood	10.4	0.61
Chrysler Imperial	14.7	0.69
Mazda RX4	21	0.33
Volvo 142E	21.4	0.33
Mazda RX4 Wag	21	0.33
Merc 230	22.8	0.28
Toyota Corolla	33.9	0.19
Merc 280	19.2	0.37
Dodge Challenger	15.5	0.45
Lincoln Continental	10.4	0.64
Valiant	18.1	0.31
Honda Civic	30.4	0.16
Hornet 4 Drive	21.4	0.33
Merc 450SE	16.4	0.54
Maserati Bora	15	1.00
Porsche 914-2	26	0.27

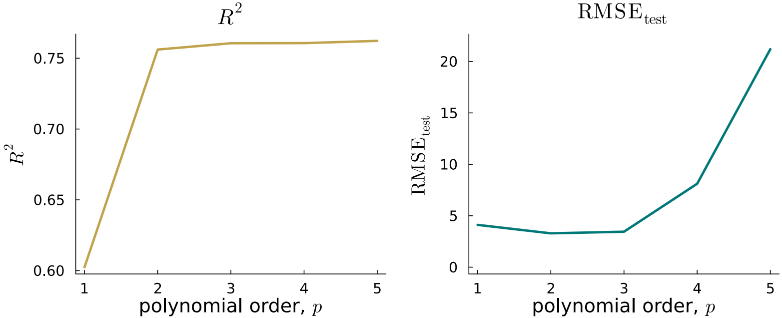
- **K-fold cross-validated prediction error**

$$\mathrm{SSE}_{CV} = \sum_{i \in \mathcal{T}_1} \left(y_i - \hat{y}_i^{(1)} \right)^2 + \ldots + \sum_{i \in \mathcal{T}_K} \left(y_i - \hat{y}_i^{(K)} \right)^2$$

$$\mathrm{RMSE}_{CV} = \sqrt{\frac{\mathrm{SSE}_{CV}}{n}}$$

- Can be used for **model choice**, for example polynomial order.

mtcars data - R^2 and RMSE-CV ($K = 4$)



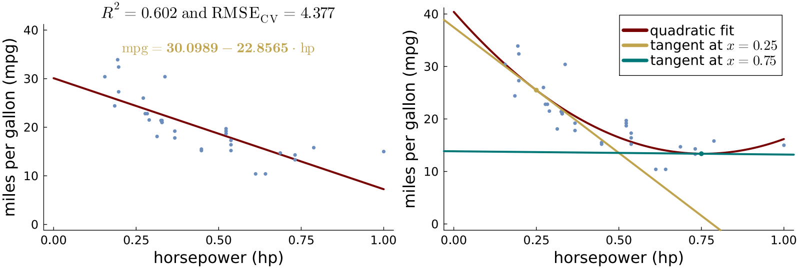
Interpretation in nonlinear model is more tricky

- **Derivative**: how much does y change when x changes?
- **Linear** model - derivative does not depend on x

$$\frac{\mathrm{d}}{\mathrm{d}x}(\beta_0 + \beta_1 x) = \beta_1$$

- **Quadratic** model - derivative depends on x

$$\frac{\mathrm{d}}{\mathrm{d}x}(\beta_0 + \beta_1 x + \beta_2 x^2) = \beta_1 + 2\beta_2 x$$



L2-regularization (Ridge regression)

- Least squares **minimizes residual sum of squares**

$$\mathrm{RSS}(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

- Same estimator as from **maximum likelihood**

$$\ell(\beta_0, \beta_1) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

- Flexible models with many parameters can **overfit**.
- **Regularization** penalizes large values of the parameters.
- **L2-regularization**

$$\mathrm{RSS}_{\mathrm{P}}(\beta_0, \beta_1) = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \quad + \quad \underbrace{\lambda \cdot (\beta_0^2 + \beta_1^2)}_{\text{L2}-\text{penalty}}$$

L2-regularization (Ridge regression)

- Multiple regression: least squares $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ minimizes

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$$

- **L2-regularization**

$$\mathrm{RSS}_{\mathrm{P}}(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \quad + \quad \underbrace{\lambda \cdot \boldsymbol{\beta}^\top \boldsymbol{\beta}}_{\mathrm{L2-penalty}}$$

- Solving for $\boldsymbol{\beta}$ (Linear Algebra section in the Prequel book)

$$\frac{\partial}{\partial \boldsymbol{\beta}} \mathrm{RSS}_{\mathrm{P}}(\boldsymbol{\beta}) = -2\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0}$$

$$\text{solution:} \quad \hat{\boldsymbol{\beta}}_{L_2} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

- **Shrinkage** of least squares $\hat{\boldsymbol{\beta}}$ toward zero. When $\boldsymbol{X}^\top \boldsymbol{X} = I_p$,

$$\hat{\boldsymbol{\beta}}_{L_2} = \frac{1}{1+\lambda} \hat{\boldsymbol{\beta}}$$

L1-regularization (Lasso regression)

- **L1-regularization** (**Lasso**)

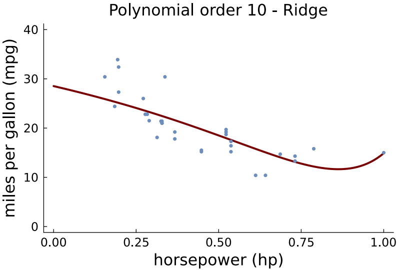
$$\mathrm{RSS}_{\mathrm{P}}(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \quad + \quad \underbrace{\lambda \cdot \sum_{j=1}^{p} |\, \beta_j \,|}_{\mathrm{L1-penalty}}$$

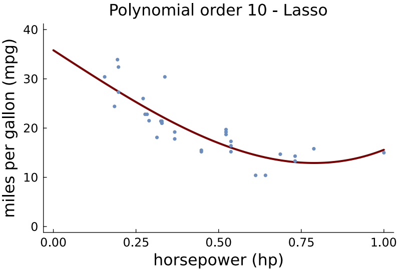
- No explicit formula, but very efficient algorithm (LARS).
- Lasso does both:
    - **shrinkage** and
    - **selection** - sets some $\hat{\beta}_j$ exactly to zero.
- $L1$ and $L2$ regularization can be seen a **Bayesian priors**. 😍

Regularization mtcars data

- Shrinkage parameter λ selected by cross-validation.
- Lasso:

$$y = 35.81 - 43.54 \cdot \mathrm{hp} + 23.32 \cdot \mathrm{hp}^3$$

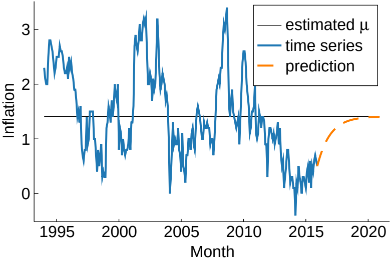


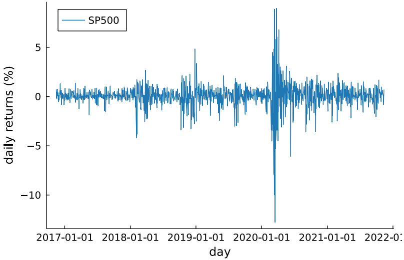


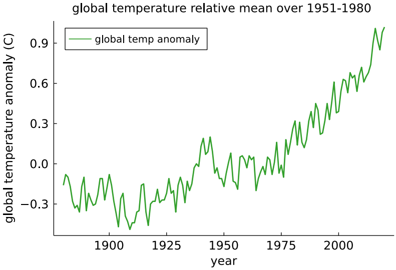
Time series data are special

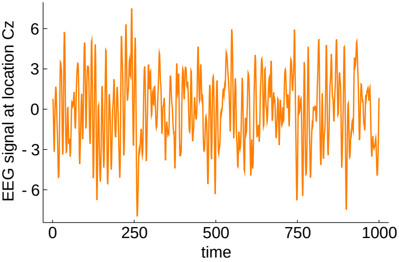
- **Time series**: data measured over time y_t, $t = 1, 2, \ldots$
- **Cross-sectional** data measured over time. **Time series regression**.
- Time series are special:
  - **Trend**, **seasonality**.
  - **Dependent observations** over time. Yesterday's value y_{t-1} can predict today's value y_t. **Autocorrelation**.
  - Sometimes the observations are **not equi-distant in time**.
- **Monte Carlo methods** like **MCMC** and **HMC** (see Bayes course!) give dependent simulated draws. Time series methods useful for measuring efficiency and diagnosing convergence problems.

Example time series

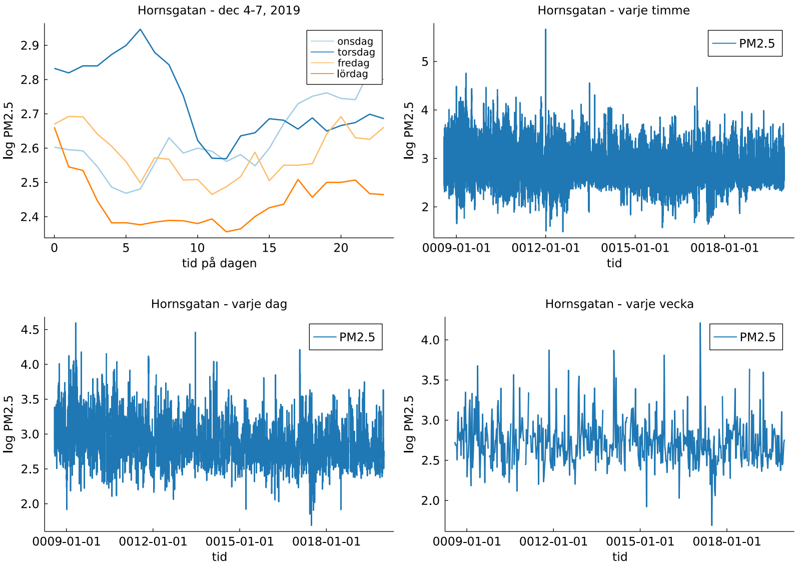








Particle matter (PM2.5) at the street Hornsgatan



Repetition - sample correlation

- **Covariance** between two variables

$$s_{xy} = \mathrm{cov}(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

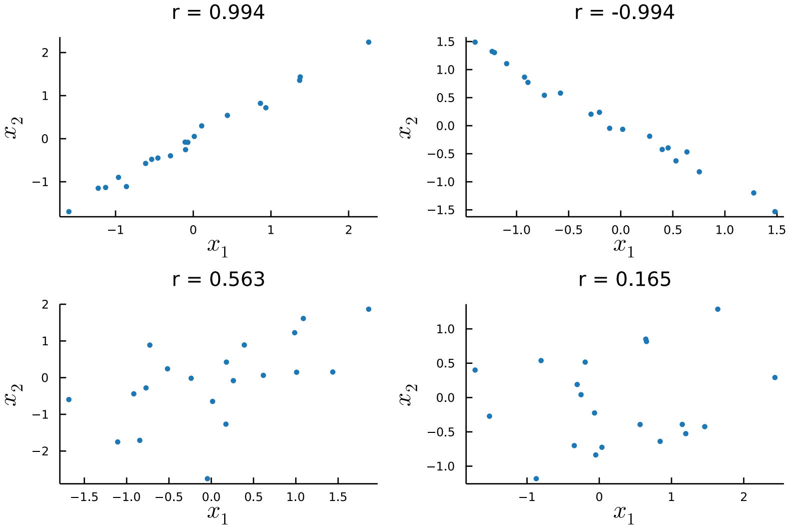
- **Correlation** between two variables:

$$r_{xy} = \mathrm{corr}(x, y) = \frac{s_{xy}}{s_x s_y}$$

where

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

Repetition - sample correlation



Autocorrelation of order 1

- Observations in a **time series** y_t are often dependent/**correlated**.

- **Autocorrelation** of **order 1**:

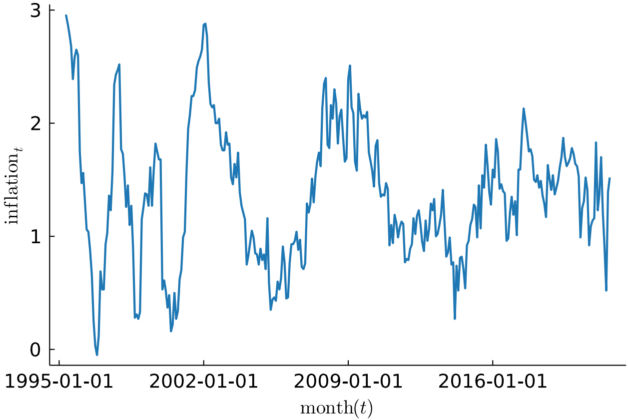
$$r_1 = \mathrm{corr}(y_t, y_{t-1})$$

- "Correlation between today's and yesterday's value."

- "Correlation between this month and the previous month."

- "First lag": y_{t-1}.

Inflation

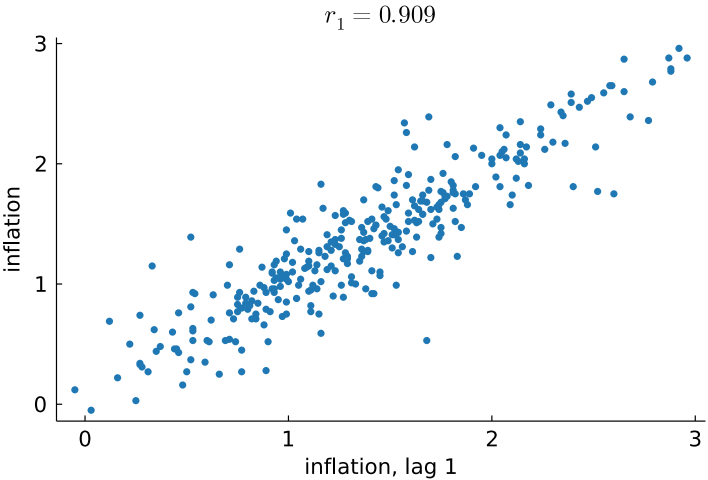


3
2
1
0
inflation$_t$
1995-01-01
2002-01-01
2009-01-01
2016-01-01
month(t)

Lagged variables - inflation

	A	B	C	D	E	F
1	Månad	Inflation(t)	Inflation(t-1)	Inflation(t-2)	Inflation(t-3)	Inflation(t-4)
2	1995-05-01	2.96				
3	1995-06-01	2.88	2.96			
4	1995-07-01	2.79	2.88	2.96		
5	1995-08-01	2.68	2.79	2.88	2.96	
6	1995-09-01	2.39	2.68	2.79	2.88	2.96
7	1995-10-01	2.58	2.39	2.68	2.79	2.88
8	1995-11-01	2.65	2.58	2.39	2.68	2.79
9	1995-12-01	2.6	2.65	2.58	2.39	2.68
10	1996-01-01	1.75	2.6	2.65	2.58	2.39
11	1996-02-01	1.47	1.75	2.6	2.65	2.58
12	1996-03-01	1.56	1.47	1.75	2.6	2.65
13	1996-04-01	1.31	1.56	1.47	1.75	2.6
14	1996-05-01	1.06	1.31	1.56	1.47	1.75
15	1996-06-01	1.04	1.06	1.31	1.56	1.47
16	1996-07-01	0.88	1.04	1.06	1.31	1.56
17	1996-08-01	0.66	0.88	1.04	1.06	1.31
18	1996-09-01	0.25	0.66	0.88	1.04	1.06
19	1996-10-01	0.03	0.25	0.66	0.88	1.04
20	1996-11-01	-0.05	0.03	0.25	0.66	0.88
21	1996-12-01	0.12	-0.05	0.03	0.25	0.66
22	1997-01-01	0.69	0.12	-0.05	0.03	0.25

Inflation - autocorrelation lag 1



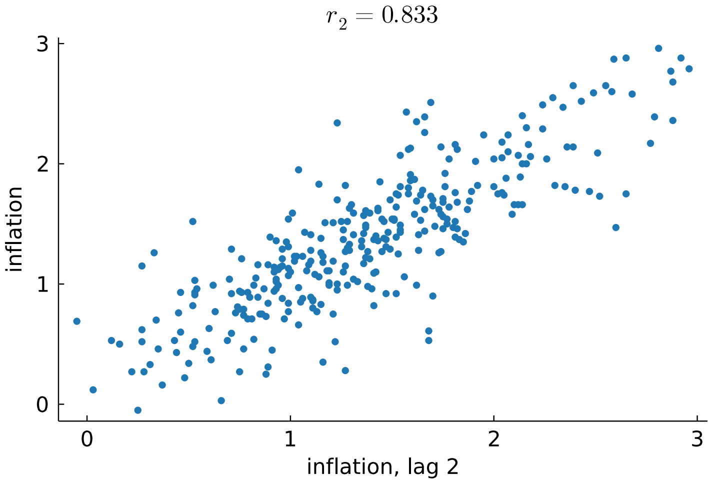
Autocorrelation of order 2

- **Autocorrelation** of **order 2**:

$$r_2 = \mathrm{corr}(y_t, y_{t-2})$$

- "Correlation between today's value and the value two days back."
- "Correlation between this month's value and the value two months back."
- "Second lag": y_{t-2}.

Autocorrelation lag 2



Autocorrelation function

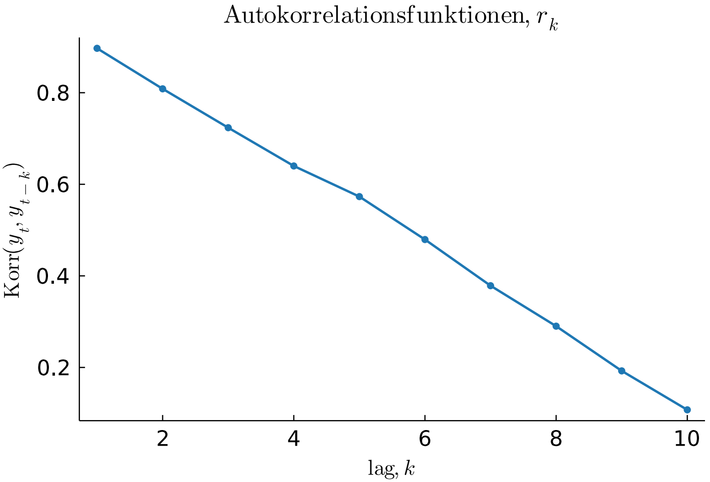
- **Autocorrelation** of **order** k

$$r_k = \mathrm{corr}(y_t, y_{t-k})$$

- "Correlation between this month's value and k months back in time.".

- **Autocorrelation function** (**ACF**) is r_k as a function of the time delay, k.

Inflation - autokorrelationsfunktion



Autoregressive models

- Autoregressive model of order 1 ($\mathrm{AR}(1)$)

$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim N(0, \sigma_\varepsilon^2)$$

- $\mathrm{AR}(1)$ is a **regression with y_{t-1} as explanatory variable**!
- Fit with the **least squares** method

$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t$$

- Autoregressiv modell av ordning p ($\mathrm{AR}(p)$)

$$y_t = \beta_0 + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} + \varepsilon_t, \qquad \varepsilon_t \sim N(0, \sigma_\varepsilon^2)$$

- $\mathrm{AR}(p)$ is a **multiple regression** with the p explanatory variables $y_{t-1}, ..., y_{t-p}$.

AR(1) for inflation - R

```
> library(SUdatasets)
> arimafit = arima(swedinfl$KPIF, order = c(1,0,0))
> arima_coef_summary(arimafit)

Parameter estimates
--------------------------------------------------------------
     Estimate Std. Error z-ratio Pr(>|z|)   2.5 %  97.5 %
ar1   0.91801   0.022383 41.0135        0 0.87414 0.96188
mean  1.43624   0.165006  8.7042        0 1.11282 1.75965
>
> |
```

AR(4) for inflation - R

```
> library(SUdatasets)
> arimafit = arima(swedinfl$KPIF, order = c(4,0,0))
> arima_coef_summary(arimafit)

Parameter estimates
-------------------------------------------------------------
      Estimate Std. Error   z-ratio Pr(>|z|)     2.5 %   97.5 %
ar1   0.8900015   0.055640 15.995742  0.00000  0.780947 0.999056
ar2   0.0586250   0.075101  0.780619  0.43503 -0.088572 0.205822
ar3   0.0062025   0.076370  0.081216  0.93527 -0.143483 0.155888
ar4  -0.0405666   0.057249 -0.708605  0.47857 -0.152774 0.071641
mean  1.4334525   0.158225  9.059583  0.00000  1.123331 1.743573
```

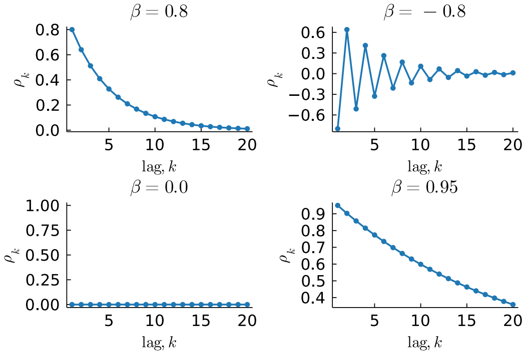
Autocorrelation function AR(1)

- AR(1)

$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim N(0, \sigma_\varepsilon^2)$$

- Population **autocorrelation function** (**ACF**) for AR(1)

$$\rho_k = \beta^k, \text{ for } k = 1, 2, \ldots$$



Simulate AR(p) and estimate ACF