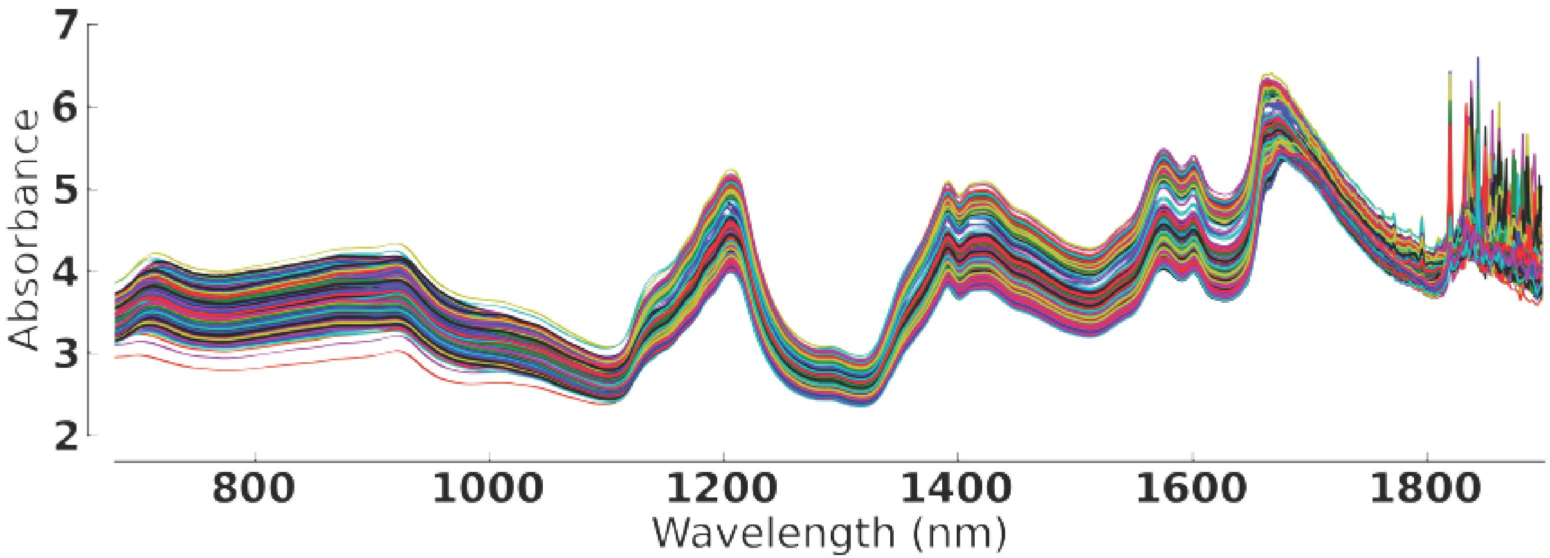
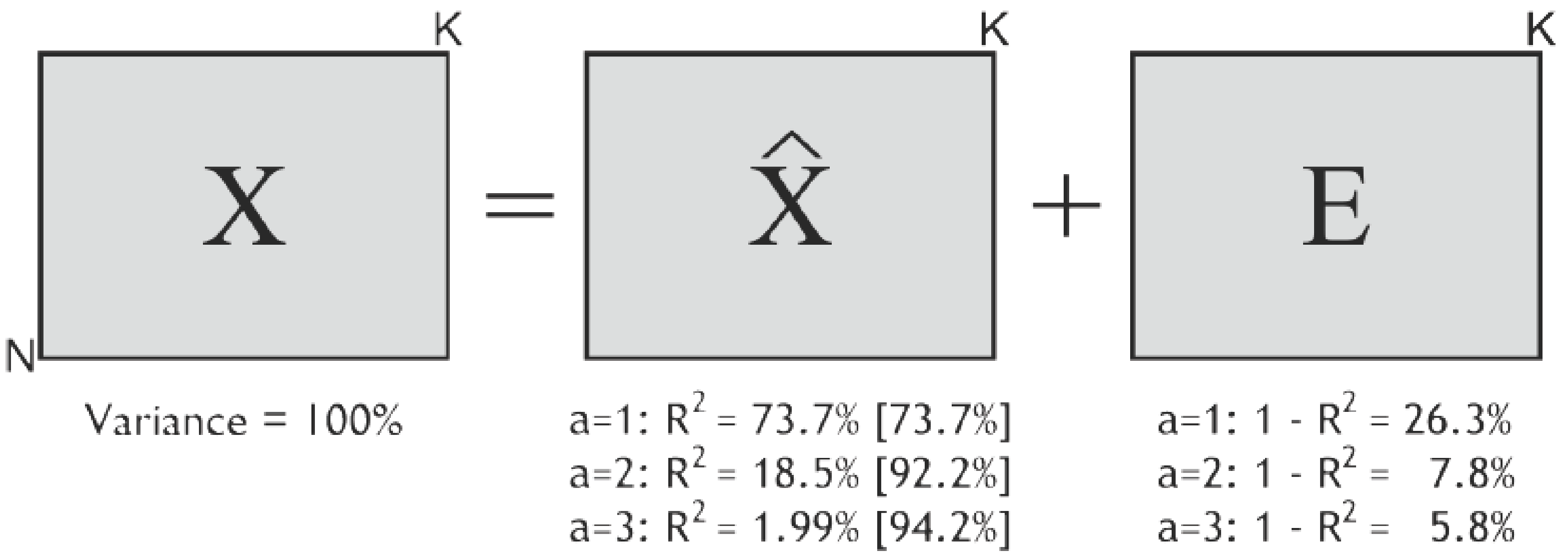
Tutorial 2

- Tutorial 2: spectral example (tablet-spectra.csv)
  - 460 observations (tablets), NIR absorbance measured at 650 different wavelengths.
  - i.e., **X** : (460x650)



Tutorial 2

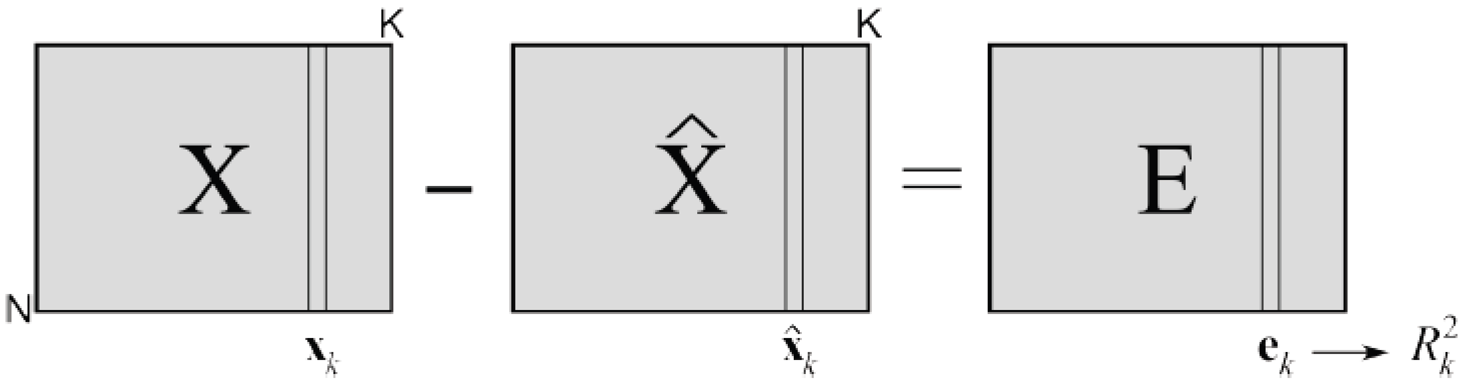
- Matrix residuals



- $R^2_{a=1} = 73.7\%$
- $R^2_{a=2} = 92.2\%$ (an additional 18.5%)
- $R^2_{a=3} = 94.2\%$ (an additional 2.00%)

Tutorial 2

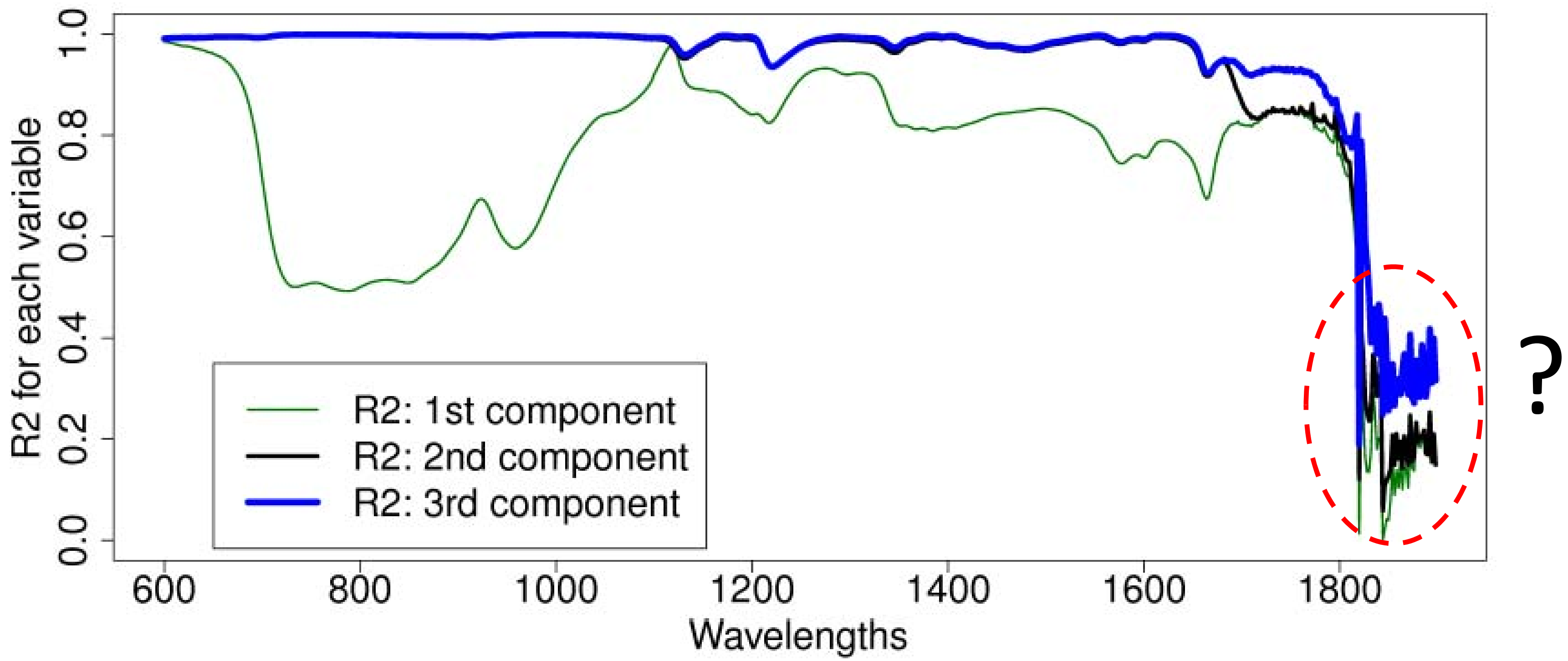
- Column residuals (from last lecture)
  - SPE is the row residual for **X**
  - Residuals also calculated for each column



  - How well each column is explained by the model

Tutorial 2; Column residuals

- Spectral example

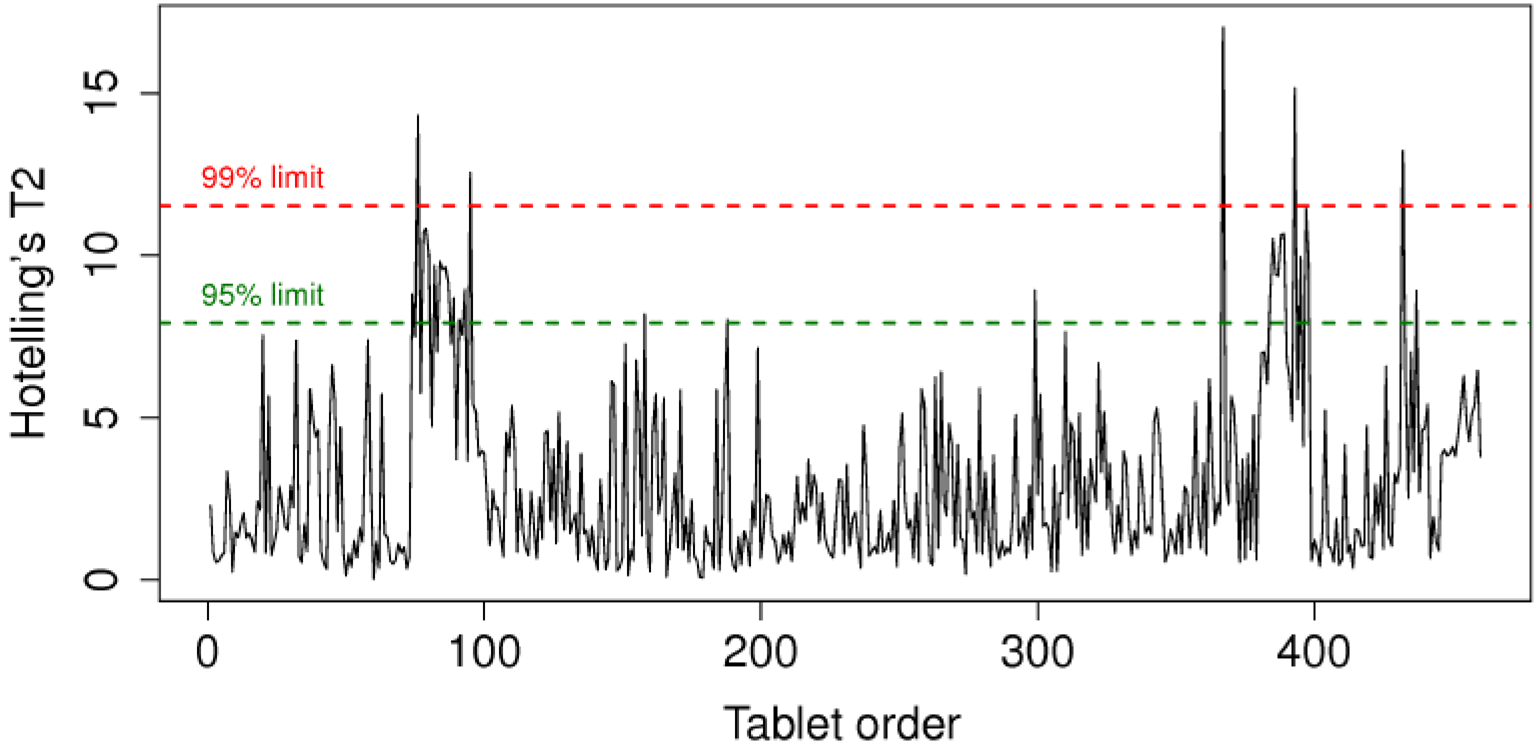


Hotelling's T^2

- Hotelling's T^2
  - SPE: summarizes error for all K variables for a row
  - T^2: summarizes all A components for a row
  - $T^2 = \sum_{a=1}^{a=A} \left(\frac{t_{i,a}}{s_a} \right)^2$
    - s_a^2 = variance of component a
    - $T^2 \geq 0$
    - Distance from the center to the projection on the plane
    - T^2 has an F-distribution; we usually show the 95% confidence limit.

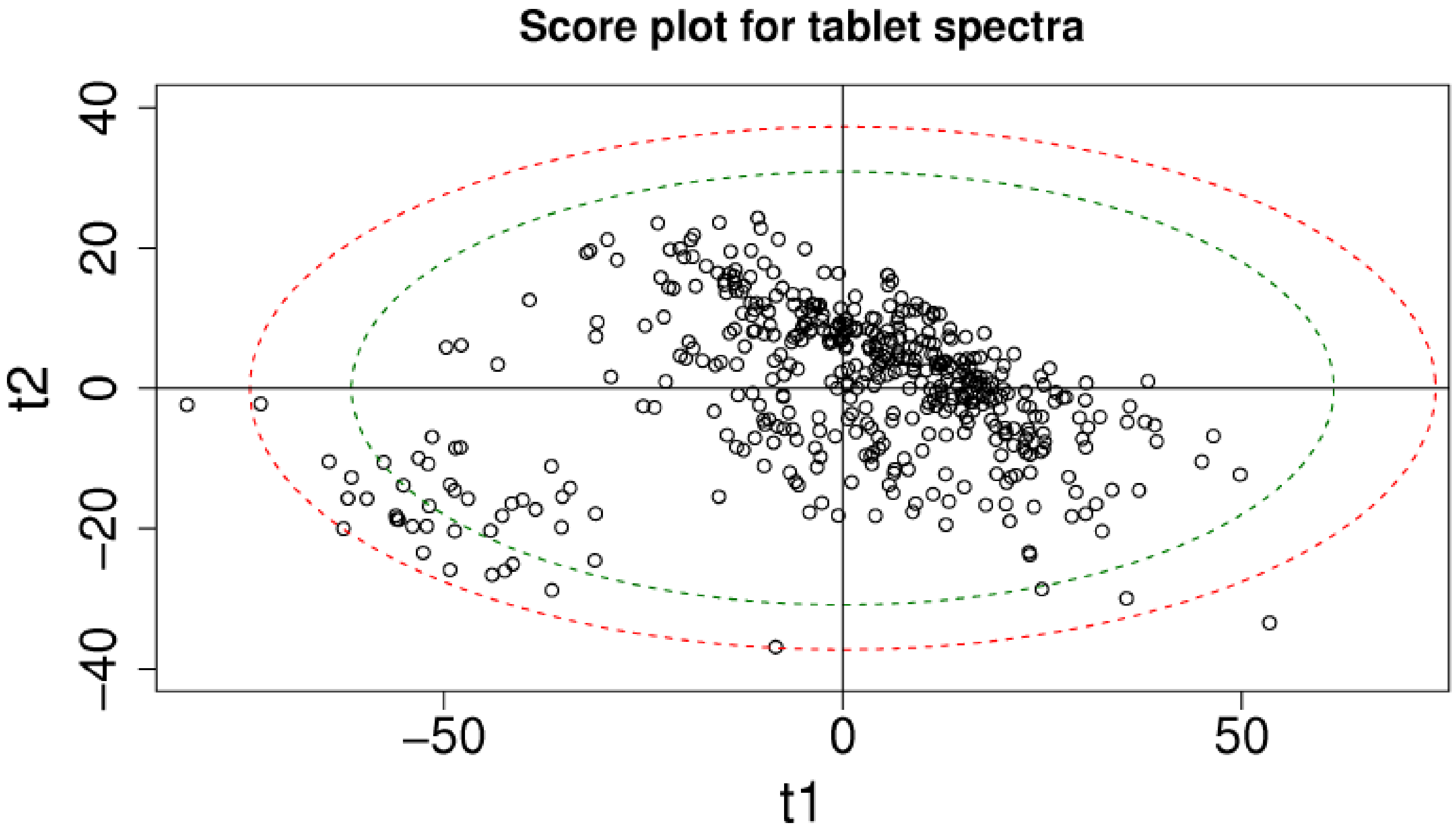
Tutorial 2: Hotelling's T^2

- Spectral example



Tutorial 2: Hotelling's T^2

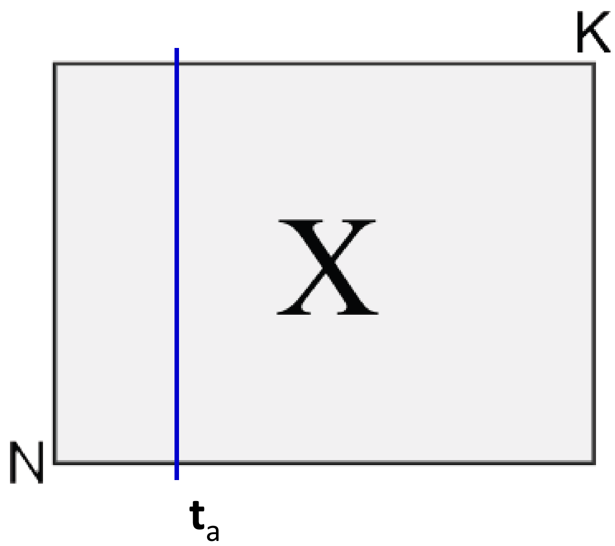
- If $A = 2$, at the 95% limit: $T^2_{A=2,\alpha=0.95} = \frac{t_1^2}{s_1^2} + \frac{t_2^2}{s_2^2}$
- An equation for an ellipse



NIPALS algorithm

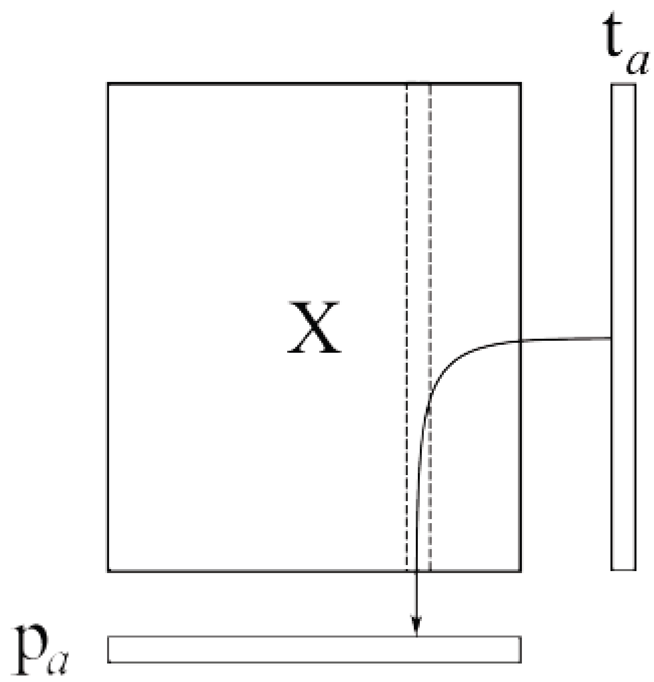
- Non-linear iterative partial least squares (NIPALS) algorithm
  - ▶ Start with **X** (preprocessed matrix of raw data)
  - ▶ We will show the algorithm for the a^{th} component

1. Select an arbitrary initial column for $\mathbf{t}_a$



NIPALS algorithm

2. Regression: every column from $\mathbf{X}$ (called $\mathbf{x}_k$) onto $\mathbf{t}_a$
   - regress $\mathbf{x}_k$ onto $\mathbf{t}_a$
   - store regression coefficient as entry in $p_{k,a}$
   - OLS: $\mathbf{y} = \beta \mathbf{x}$, and $\widehat{\beta} = \dfrac{\mathbf{x}'\mathbf{y}}{\mathbf{x}'\mathbf{x}}$
   - here: $\mathbf{x}_k = p_{k,a}\mathbf{t}_a$, so $p_{k,a} = \dfrac{\mathbf{t}'_a\mathbf{x}_k}{\mathbf{t}'_a\mathbf{t}_a}$

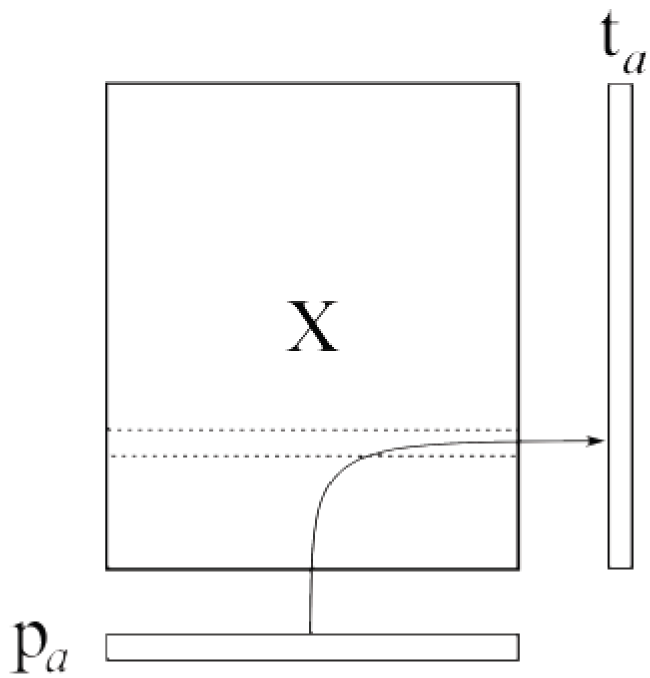


NIPALS algorithm

2. Regression (continues)
   - Repeat regression for every column in $\mathbf{X}$
   - In practice: $\mathbf{p}'_a = \frac{1}{\mathbf{t}'_a \mathbf{t}_a} \cdot \mathbf{t}'_a \mathbf{X}_a$
     - $\mathbf{t}_a$ is an $N \times 1$ column vector
     - $\mathbf{X}_a$ is an $N \times K$ matrix
     - Subscript a? Explained later ...
3. Normalize the loadings
   - $\mathbf{p}'_a$ won't have unit length (magnitude)
   - Rescale it to magnitude 1.0
   - $\mathbf{p}'_a = \frac{1}{\sqrt{\mathbf{p}'_a \mathbf{p}_a}} \cdot \mathbf{p}'_a$

NIPALS algorithm

4. Regression again: every row in $\mathbf{X}$ onto $\mathbf{p}'_a$
   - regress $\mathbf{x}_i$ onto $\mathbf{p}'_a$
   - store regression coefficient as entry in $t_{i,a}$
   - OLS: $\mathbf{y} = \beta\mathbf{x}$, and $\widehat{\beta} = \dfrac{\mathbf{x}'\mathbf{y}}{\mathbf{x}'\mathbf{x}}$
   - here: $\mathbf{x}_i = t_{i,a}\mathbf{p}'_a$, so $t_{i,a} = \dfrac{\mathbf{p}'_a\mathbf{x}'_i}{\mathbf{p}'_a\mathbf{p}_a}$



NIPALS algorithm

4. Regression (continues)
   - Repeat regression for every row in $\mathbf{X}$
   - In practice: $\mathbf{t}_a = \dfrac{1}{\mathbf{p}'_a \mathbf{p}_a} \cdot \mathbf{X}_a \mathbf{p}_a$
     - $\mathbf{t}_a$ is an $N \times 1$ column vector
     - $\mathbf{p}_a$ is an $K \times 1$ column vector

5. Converged?
   - $\mathbf{t}_a$ compared to $\mathbf{t}_a$ from previous iteration
   - change less than 1×10^{-6} to 1×10^{-9}? then stop

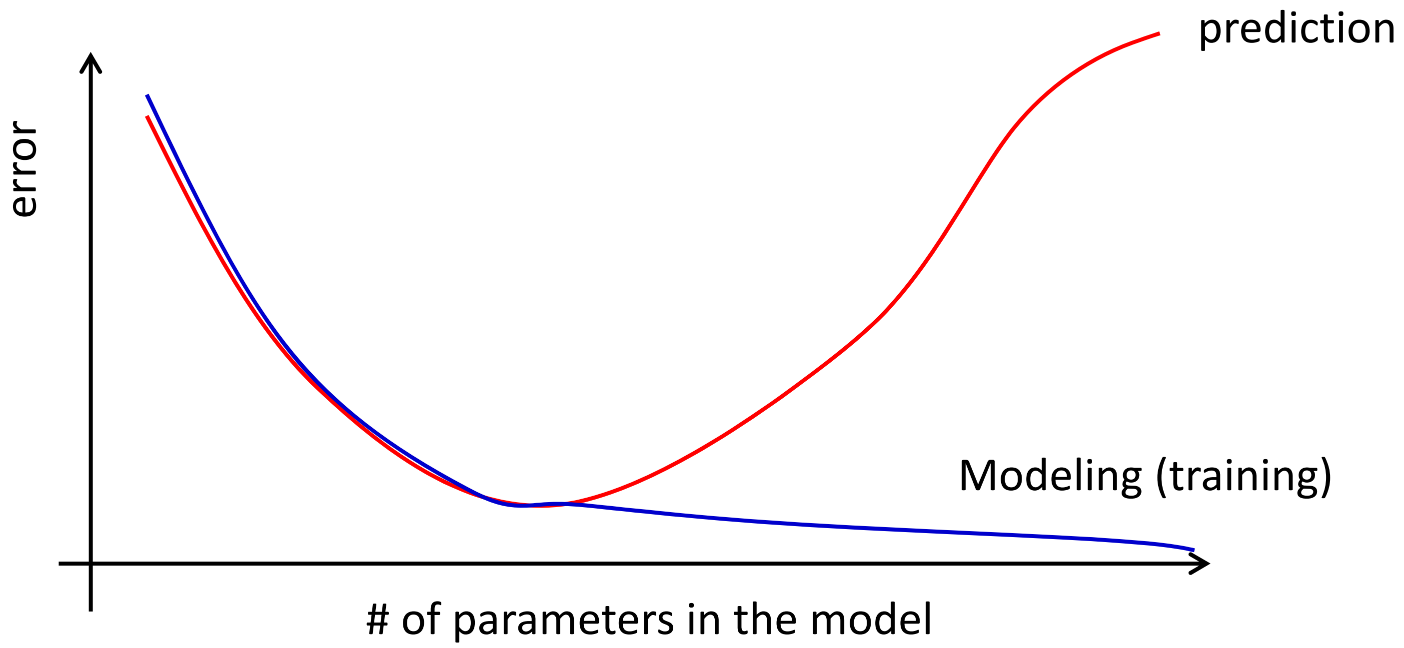
NIPALS algorithm

6. Store the score $\mathbf{t}_a$ and loading $\mathbf{p}_a$ vector
   - 200 or fewer iterations for convergence
   - Deflate:
     - removes variability captured
     - $\mathbf{E}_a = \mathbf{X}_a - \mathbf{t}_a \mathbf{p}'_a$
     - $\mathbf{X}_{a+1} = \mathbf{E}_a$
     - $a = 1$: $\mathbf{X}_a$ = preprocessed raw data
     - $a = 2$: calculated on residuals $\mathbf{E}_1$

Repeat steps 1 to 6 for every component

Cross-validation

- Cross-validation
  - A general tool for avoiding over-fitting
  - Can be applied to any model



Cross-validation

1. Rows of data (**X**) divided into G groups
2. PCA model estimated for data minus one group
3. Calculate residual $\mathbf{E}_{g,CV}$ for deleted group using the PCA model
4. Repeat 2 ~ 3 and get $\mathbf{E}_{G,CV}$

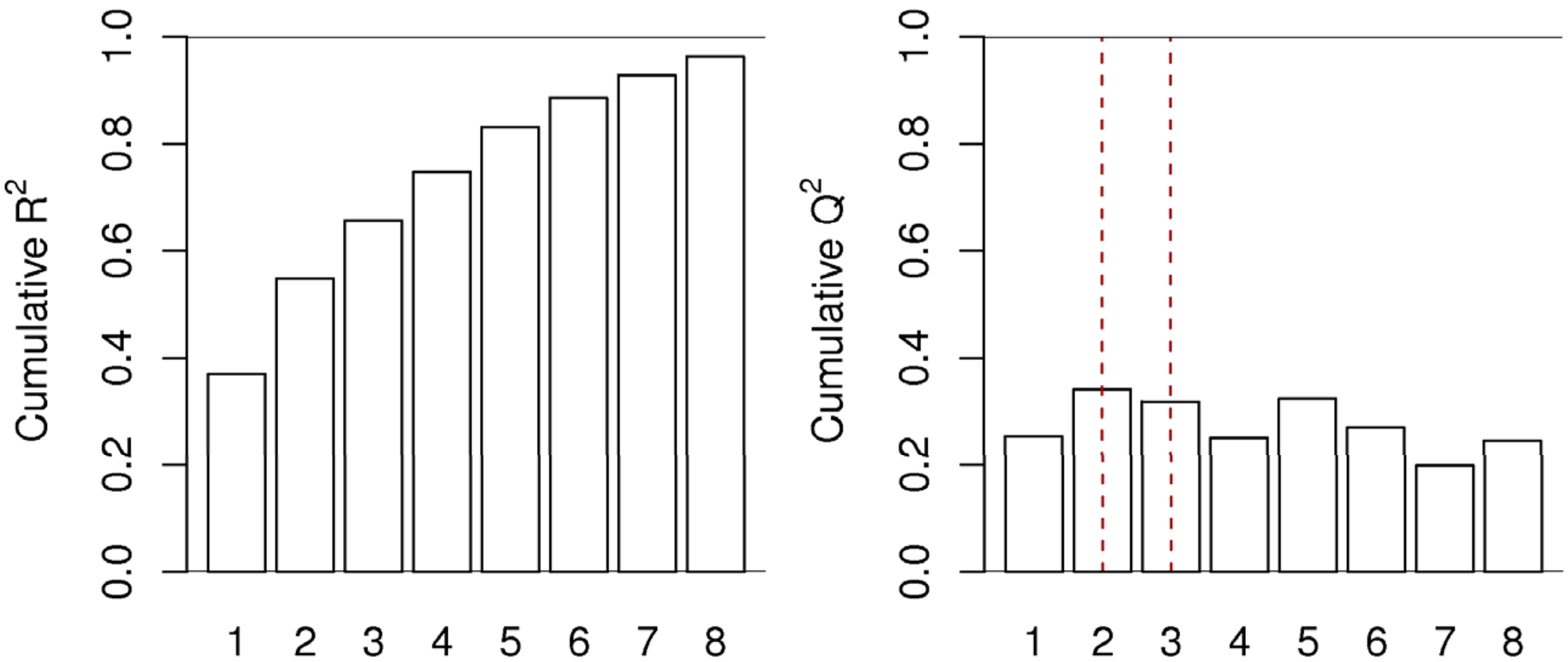
$$Q^2 = 1 - \frac{\mathrm{var}(\mathbf{E}_{G,CV})}{\mathrm{var}(\mathbf{X})} = 1 - \frac{PRESS}{SS_{\mathbf{X}}}$$

※ PRESS (prediction error sum of squares), $SS_{\mathbf{X}}$ (sum of squares of **X**)

- R^2: how well training data explained by the model
- Q^2: how well test data explained by the model

Cross-validation

- How many components are necessary?



Cross-validation

- True number of principal components?
  - No one knows.
  - Recommendation
    - Use cross-validation as guide, and always look at a few extra components and step back a few components
    - then make a judgement that is relevant to your intended use of the model.
  - Models where we intend to learn from, or optimize, or monitor a process may well benefit from fewer or more components than suggested by cross-validation.

Tutorial 3

- Food data (Foods.csv)
  - Food consumption data from 16 EU contries
  - % households consuming different types of foods
  - Objectives: find any similarities / differences among countries using ProMV

Tutorial 3

- Food data
  - % households consuming different types of foods

		1	2	3	4	5	6	7	8	9	10
		Grain_Coffee	**Inst_Coffee**	**Tea**	**Sweet**	**Bisc**	**Pa_Soup**	**Ti_Soup**	**In_Potat**	**Fro_Fish**	**Fro_Veg**
1	**Germany**	90	49	88	19	57	51	19	21	27	21
2	**Italy**	82	10	60	2	55	41	3	2	4	2
3	**France**	88	42	63	4	76	53	11	23	11	5
4	**Holland**	96	62	98	32	62	67	43	7	14	14
5	**Belgium**	94	38	48	11	74	37	23	9	13	12
6	**Luxembourg**	97	61	86	28	79	73	12	7	26	23
7	**England**	27	86	99	22	91	55	76	17	20	24
8	**Portugal**	72	26	77	2	22	34	1	5	20	3
9	**Austria**	55	31	61	15	29	33	1	5	15	11
10	**Switzerland**	73	72	85	25	31	69	10	17	19	15
11	**Sweden**	97	13	93	31		43	43	39	54	45
12	**Denmark**	96	17	92	35	66	32	17	11	51	42
13	**Norway**	92	17	83	13	62	51	4	17	30	15
14	**Finland**	98	12	84	20	64	27	10	8	18	12
15	**Spain**	70	40	40		62	43	2	14	23	7
16	**Ireland**	30	52	99	11	80	75	18	2	5	3

Tutorial 3

- Food data

		11	12	13	14	15	16	17	18	19	20
		Apples	**Orang**	**Ti_Fruit**	**Jam**	**Garlic**	**Butter**	**Margarine**	**Olive_Oil**	**Youg**	**Crisp_Bread**
1	**Germany**	81	75	44	71	22	91	85	74	30	26
2	**Italy**	67	71	9	46	80	66	24	94	5	18
3	**France**	87	84	40	45	88	94	47	36	57	3
4	**Holland**	83	89	61	81	15	31	97	13	53	15
5	**Belgium**	76	76	42	57	29	84	80	83	20	5
6	**Luxembourg**	85	94	83	20	91	94	94	84	31	24
7	**England**	76	68	89	91	11	95	94	57	11	28
8	**Portugal**	22	51	8	16	89	65	78	92	6	9
9	**Austria**	49	42	14	41	51	51	72	28	13	11
10	**Switzerland**	79	70	46	61	64	82	48	61	48	30
11	**Sweden**	56	78	53	75	9	68	32	48	2	93
12	**Denmark**	81	72	50	64	11	92	91	30	11	34
13	**Norway**	61	72	34	51	11	63	94	28	2	62
14	**Finland**	50	57	22	37	15	96	94	17		64
15	**Spain**	59	77	30	38	86	44	51	91	16	13
16	**Ireland**	57	52	46	89	5	97	25	31	3	9

Tutorial 3

- In ProMV,

Some properties of PCA models

- The model is defined by the loadings vectors, $\mathbf{p}_1$, $\mathbf{p}_2$, ... , $\mathbf{p}_A$; each are a (K×1) vector, and can be collected into a single matrix, **P**, a (K×A) loadings matrix.
- These vectors form a line for one component, a plane for 2 components, and a hyperplane for 3 or more components. This line, plane or hyperplane define the latent variable model.
- An equivalent interpretation of the model plane is that these direction vectors are oriented in such a way that the scores have maximal variance for that component. No other directions of the loading vector (i.e. no other hyperplane) will give a greater variance.

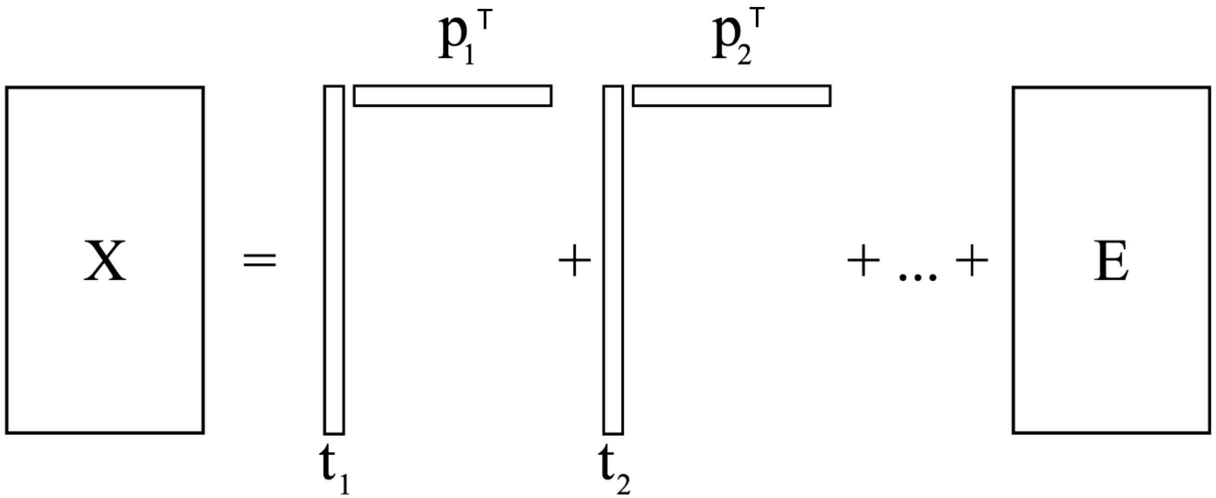
Some properties of PCA models

- This plane is calculated with respect to a given data set, **X**, an (N×K) matrix, so that the direction vectors best-fit the data. We can say then that with one component, the best estimate of the original matrix **X** is:

$$\hat{\mathbf{X}}_1 = \mathbf{t}_1\mathbf{p}_1^T \quad \text{or equivalently} \quad \mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{E}_1$$

- If we fit a second component:

$$\hat{\mathbf{X}}_2 = \mathbf{t}_2\mathbf{p}_2^T \quad \text{or equivalently} \quad \mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_1^T + \mathbf{E}_2$$



Some properties of PCA models

- The loadings vectors are of unit length: $\|\mathbf{p}_a\| = 1.0$
- The loading vectors are orthogonal to one another: $\mathbf{p}_i \perp \mathbf{p}_j$
- The variance of the $\mathbf{t}_1$ vector must be greater than the variance of the $\mathbf{t}_2$ vector, and so on.
- Each loading direction, $\mathbf{p}_a$, must point in the direction that best explains the data; but this direction is not unique, since $-\mathbf{p}_a$ also meets this criterion. If we did select $-\mathbf{p}_a$ as the direction, then the scores would just be $-\mathbf{t}_a$ instead. This does not matter too much, because $\mathbf{t}_a\mathbf{p}_a^T = (-\mathbf{t}_a)(-\mathbf{p}_a^T)$

Readings

- History
  - K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, **2**(6), 559–572. (1901)
  - H. Hotelling,"Analysis of a Complex of Statistical Variables with Principal Components," *Journal of Educational Psychology*, **24**, 417-441, 498-520, (1933)
  - Papers by K. Karhunen, (1947) in Russian & M. Loeve, (1948) in French
- NIPALS algorithm
  - H. Wold, "Estimation of principal components and related models by iterative least squares," in *Multivariate Analysis* (Ed., Krishnaiah, P. R.), Academic Press, NY, pp. 391-420 (1966).
- Cross-validation
  - S. Wold, "Cross-validatory estimation of the number of components in factor and principal components models," *Technometrics*, **20**, 397-405, (1978).

Readings

- General
  - S. Wold, K. Esbensen, and P. Geladi, "Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, **2**, 37-52, (1987).
  - T. Kourti and J. MacGregor, "Process analysis, monitoring and diagnosis using multivariate projection methods – a tutorial, *Chemometrics and Intelligent Laboratory Systems,* **28**, 3-21, (1995).
  - J. MacGregor, H. Yu, S. García-Muñoz, and J. Flores-Cerrillo, "Data-Based Latent Variable Methods for Process Analysis, Monitoring and Control". Computers and Chemical Engineering, **29**, 1217-1223, (2005).