Data Analytics using Python (Part 2)

Stephen Goldrick & Cheng Zhang
Lecturer in Bioprocess Digitalisation

Contents

- Review Jupyter-Notebook
- Live Demo of data analysis
- Introduction to PCA
- Calculation of PCA on data set

Contents

- Review Jupyter-Notebook

Please raise your hand if you not can access it?
Please raise your hand if you can accessed through Anaconda?

Please raise your hand if you can accessed through Google Co-Lab?

Overview of lecture series outline (Part 1)

- Introduction to Multivariate Data Analysis (MVDA)
- Understanding the Covariance and Correlation Matrix
  - Calculation of Covariance and Correlation matrix
- Overview of Python and Jupyter-notebook
  - Installation
  - Simple importation and plotting
  - Basic statistics (Covariance/Correlation)
  - Sample Jupyter notebooks
  - Demonstration of advanced Jupyter-Notebook

Learning outcome from Python Data analytics lecture series Day 1

- To introduce the concepts and principles of effective and efficient multivariate data analysis
  - Calculation of the Covariance matrix
- How to create and run a Jupyter-Notebook
  - Sample code for Plotting, correlation development

Overview of lecture series outline (Part 2)

- Calculation of simple statistics and
- Introduction to Principal Component Analysis - Understanding the Covariance and Correlation Matrix
  - Calculation of Covariance and Correlation matrix
- Overview of Python and Jupyter-notebook
  - Installation
  - Simple importation and plotting
  - Basic statistics (Covariance/Correlation)
  - Sample Jupyter notebooks
  - Demonstration of advanced Jupyter-Notebook

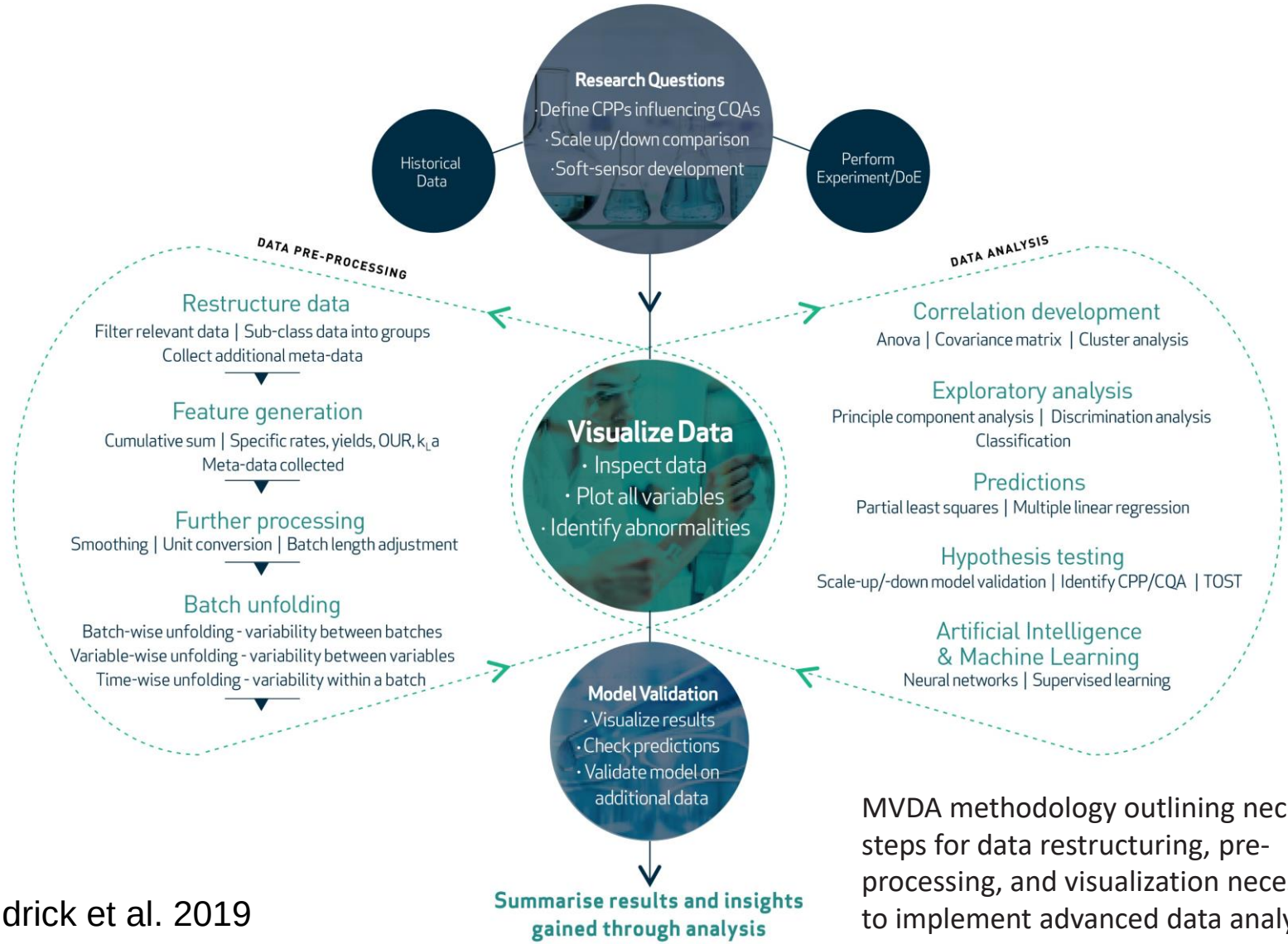
Overview of lecture series outline (Part 2)

- Demonstration of Jupyter-notebook
  - Simple plotting exercise's
- Concepts of Principal Component Analysis (PCA)
- Understanding scores and loadings plots
  - Test examples of PCA and interpretation of results
- Introduction in Big Data
  - What is it and how do we handle it
  - Jupyter notebook examples handling Big data
- Advanced demonstration of MVDA in an industrial setting

Learning outcome from Python Data analytics lecture series

- To introduce the concepts and principles of effective and efficient multivariate data analysis
  - Calculation of the Covariance matrix
  - How to perform Principal Component Analysis (PCA) on complex data sets
    - How to interpret Scores and Loading plots
- How to create and run a Jupyter-Notebook
  - Sample code for Plotting, correlation development and PCA calculation

Data analysis methodology



Goldrick et al. 2019

MVDA methodology outlining necessary steps for data restructuring, pre-processing, and visualization necessary to implement advanced data analytics on complex biopharmaceutical data sets.

Different MVDA methods (there are 100's)

An example of some common MVDA techniques:

- **Linear regression (LR):** $y = \beta_1 X_1$
- **Multiple linear regression (MLR):** $y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- **Covariance:** Analysis of relationships between variables $\mathrm{cov}(X_1, X_2)$
- **Discrimination/factor analysis (DA/FA):** Defines discrimination features related to associating variables different groups or clusters $DA(X)$
- **Principal Component Analysis (PCA):** Determines correlation between variables in the X-data structure $PCA(X)$
- **Partial Least Squares (PLS):** Determines relationships between the X variables to enable predictions of the Y variables $PLS(X,Y)$

Live Demo of Data analysis for a Bioreactor problem

Multivariate data analysis: 48 batch records

- **Big data (Available on Moodle)**
  - Download and unzip the folder on desktop
  - Open text file called “Bioreactor_data_headers_v1.csv ”
- **Overview of data set**
  - 48 mammalian cell culture batches were operated at different temperatures and pH set points and initial RPMS
  - Final Titre and Aggregation was recorded for each batch and classed as
    - 1: Good Batches for Aggregation below 5 g/L (Classification = 1)
    - 2: Average Batches Aggregation between 3-5 g/L g/L (Classification = 2 in Column 6)
    - 3: Bad Batches for aggregation over 8 g/L (Classification = 3)

pH	Temp	RPM	Titre	Aggregation	Classification
6.2309	23.781	2433.8	0.39391	12.133	3
6.319	26.805	2614.2	0.59547	5.614	2
6.1867	13.402	2396.9	-1.1285	10.902	3
6.2093	37.635	2202.6	0.96203	10.18	3
5.9788	27.495	2442.6	-0.19477	8.5016	3
6.0208	32.504	2094.8	2.9491	10.453	3
5.9951	41.044	2018.5	3.0823	10.63	3
6.0619	34.635	2244.5	2.3326	10.755	3
6.2883	31.368	2047.3	1.8792	9.3396	3
6.0482	28.663	2010.7	2.6717	9.5146	3
6.7819	32.798	2063	4.5927	6.777	2
6.5332	38.517	1980.3	4.4523	5.7471	2
6.0466	24.293	1972	2.4619	7.2148	2
6.2408	34.082	1992.5	3.3688	7.9356	3
6.4596	39.85	1770	1.9887	9.2769	3
6.1928	25.061	1721.4	3.9809	5.6065	2
6.4492	39.173	1569.8	4.5535	6.021	2

- Data set = [48x6] i.e 24 batches and 5 variables

Different MVDA methods (there are 100's)

An example of some common MVDA techniques:

- **Linear regression (LR):** $y = \beta_1 X_1$
- **Multiple linear regression (MLR):** $y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- **Covariance:** Analysis of relationships between variables $\mathrm{cov}(X_1, X_2)$
- **Discrimination/factor analysis (DA/FA):** Defines discrimination features related to associating variables different groups or clusters $DA(X)$
- **Principal Component Analysis (PCA):** Determines correlation between variables in the X-data structure $PCA(X)$
- **Partial Least Squares (PLS):** Determines relationships between the X variables to enable predictions of the Y variables $PLS(X, Y)$

Covariance – Basic statistics required

- **Mean($\bar{X}$)**: average point within dataset

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

- **Standard deviation (std)**: A measure of the spread of data points in a dataset: i.e it is the average distance from mean of the data set to a point.

$$std = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n-1}}$$

- **Variance (std^2)**: squared standard deviation

$$\text{var} = std^2 = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(X_i - \overline{X})}{n-1}$$

- **Covariance (cov)**: measures the strength of correlation between two variables (X,Y) or more sets of variables

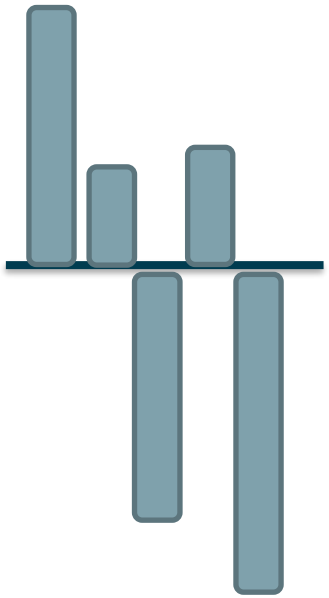
$$\text{cov}(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)}$$

The importance of scaling

- Scaling can help transform the data to a suitable form for analysis.
  - Essential when analysing variables that are measured in different units. i.e Volume ranging from 10,000-20,000 Litres compared to pH ranging from 6.5 – 7.1
  - Allows variables with small variances to have equal weight during data analysis

Raw data

$$X$$

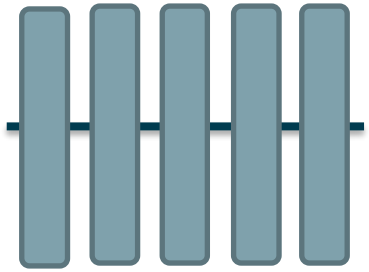


Mean centered data:

$$X - \bar{X}$$

Mean centered data and scaled data:

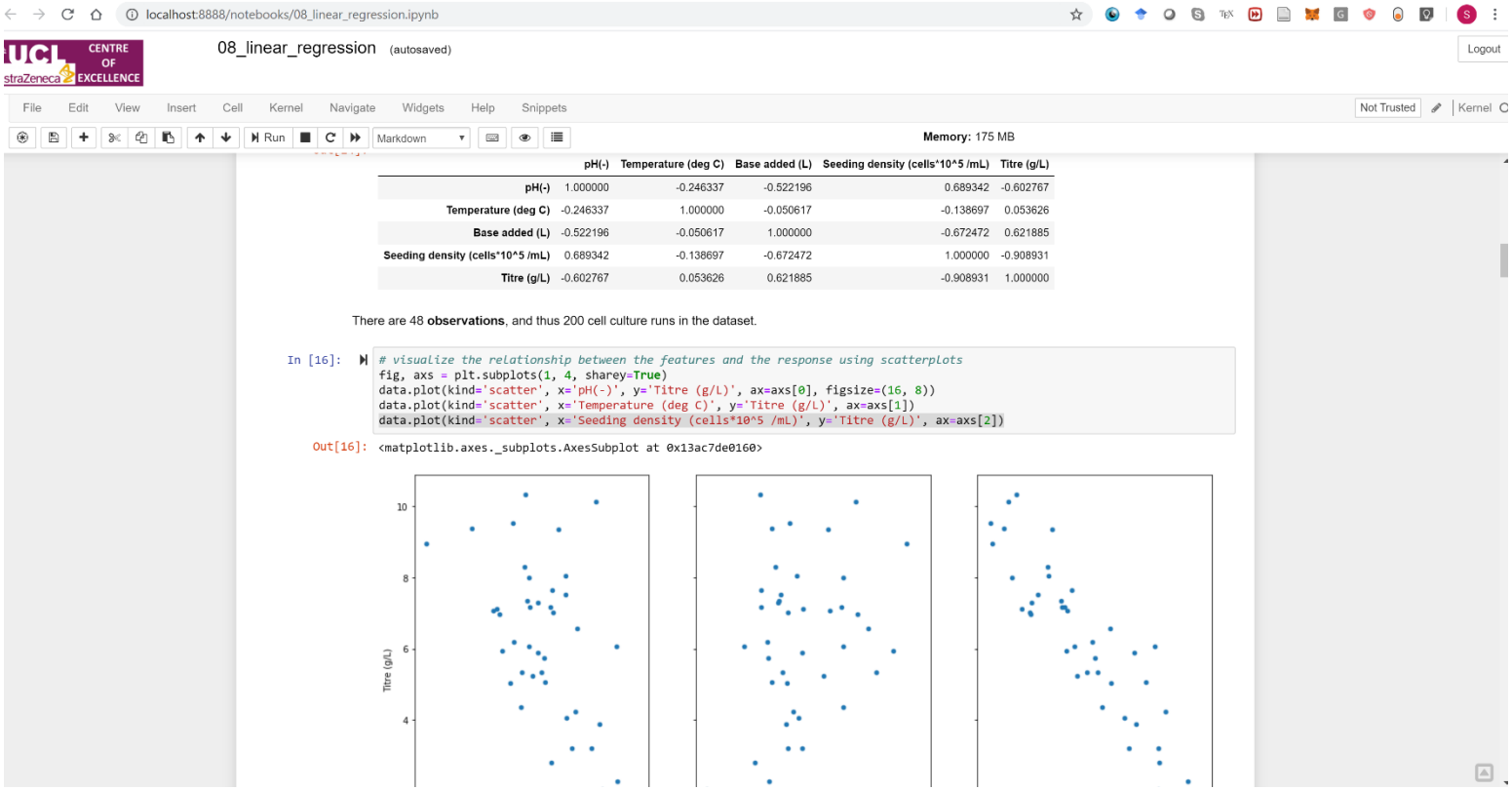
$$\frac{X - \bar{X}}{std(X)}$$



Also called normalising data or autoscaling

Jupyter- Notebook – front end for Python

- (Makes Python Look “Pretty”)
- Excellent tool to analysis and visualise your data
- Generate reports that are interactive



Jupyter-Notenote is essentially Excel on Steroids!!

Mostly likely you will an error

Modulenotfound: No Module named "plotly"

- Open anaconda prompt and type

  pip install plotly

  Click "Restart kernel and run all"

Python – Jupyter – Notebook

- Each cell can be run individually (ctrl Enter)
- Python packages need to imported before they are used

```
In [1]:  ## Import required Packages
         import pandas as pd

In [2]:  data = pd.read_csv('pH_Titre_Data.txt',sep=",")
```

Data will be saved as Dataframe called data

Name of file to import

Data is separated by comma

Pandas (pd) inbuilt function to read csv file
(File must be in the same folder as Jupyter-notebook!!)

Python – Jupyter – Notebook

- *data* saved as Dataframe
  - Rows are observations
  - Columns are variables

```
data
```

	pH set-point (-)	mAb titre
0	6.6	1.55
1	6.8	3.10
2	7.0	4.60
3	7.2	6.20

Python – Jupyter – Notebook

- *.describe() for* Dataframe
  - Prints out basic stats for data

```
data.describe()
```

6]:

	pH set-point (-)	mAb titre
count	4.000000	4.000000
mean	6.900000	3.862500
std	0.258199	1.994733
min	6.600000	1.550000
25%	6.750000	2.712500
50%	6.900000	3.850000
75%	7.050000	5.000000
max	7.200000	6.200000

Python – Jupyter – Notebook

- Dataframe enables simple calculations of useful statistics

```
data.mean()
```

```
pH set-point (-)    6.9000
mAb titre           3.8625
dtype: float64
```

```
data.cov()
```

	pH set-point (-)	mAb titre
pH set-point (-)	0.066667	0.515000
mAb titre	0.515000	3.978958

Python – Jupyter – Notebook

- How to normalise the data and then calculate correlation matrix

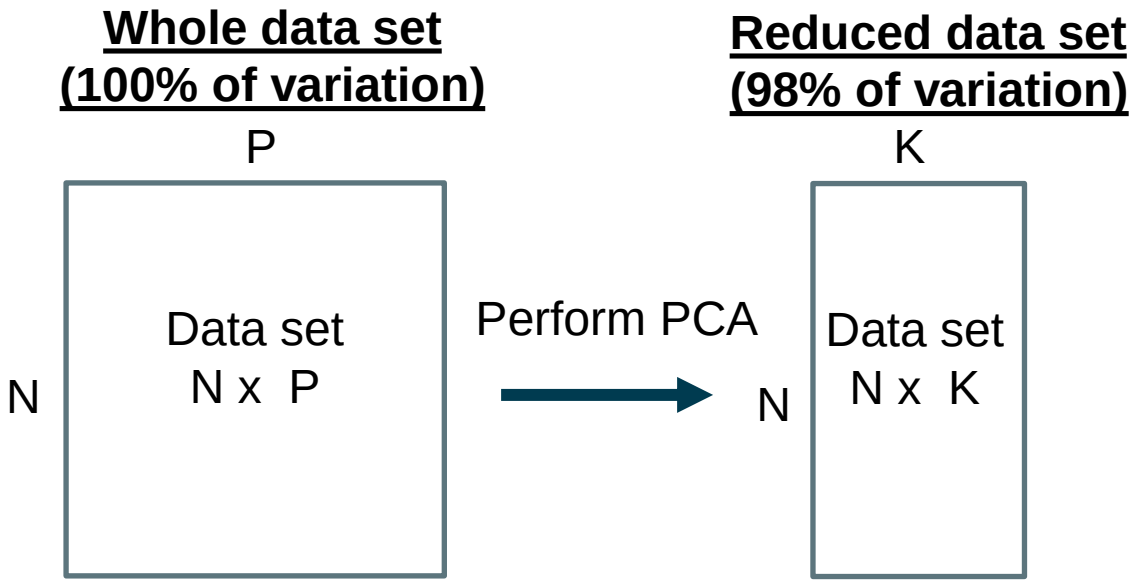
```
normalized_df=(data-data.mean())/data.std()
```

```
normalized_df.cov()
```

	pH set-point (-)	mAb titre
pH set-point (-)	1.000000	0.999927
mAb titre	0.999927	1.000000

MVDA – Muliple Linear Regression

- Principal component analysis (PCA) is the workhorse of the majority of MV
- PCA is technique that is suitable to analysis complex data sets by reducing the dimensionality of the data set and enabling patterns to be identified
  - Essentially summarising the main sources of variability through newly generated vectors (called components)

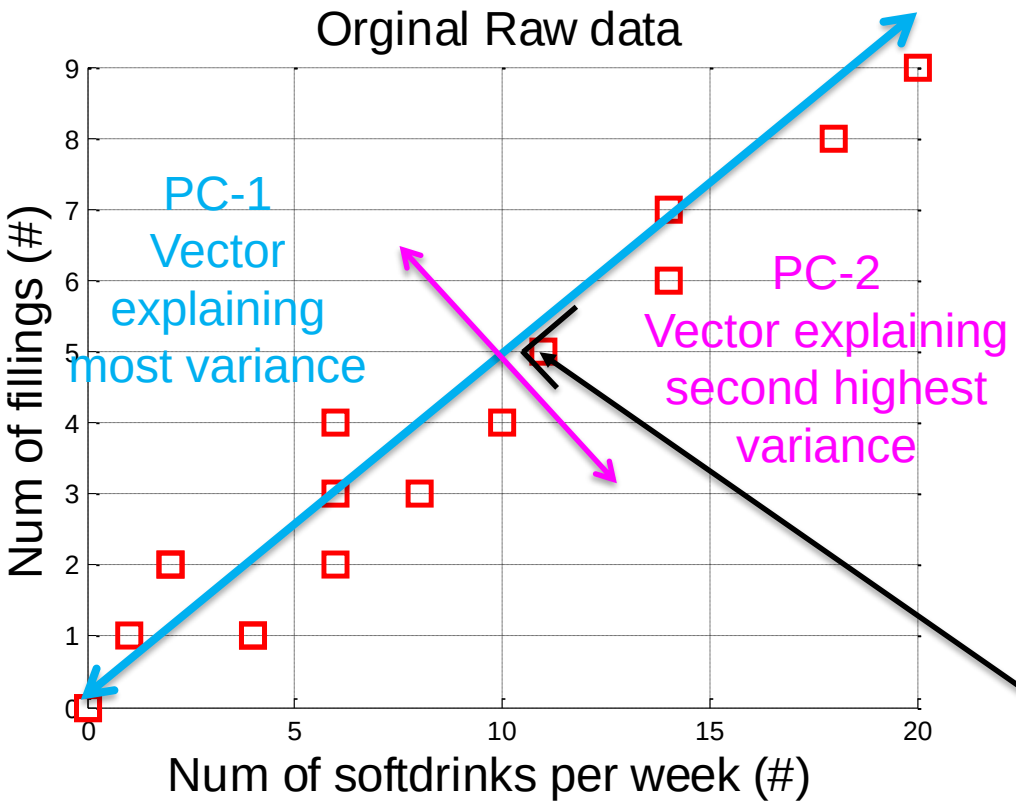


Principal Component Analysis (PCA) is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information in the large set.

PCA: How does it work?

- It essentially transforms the data to aid in identifying patterns
  - Expressing the data in such a way as to highlight their similarities and differences
- Principal Component Analysis decomposes the data into Scores and Loadings
  - **Scores Plot** represents each observation of the data set (e.g. batches or samples)
  - **Loadings Plot** represents the variables of the data set (e.g. temperature, pH)
  - **Analysing both plots side-by-side provides a richer interpretation of the outputs**

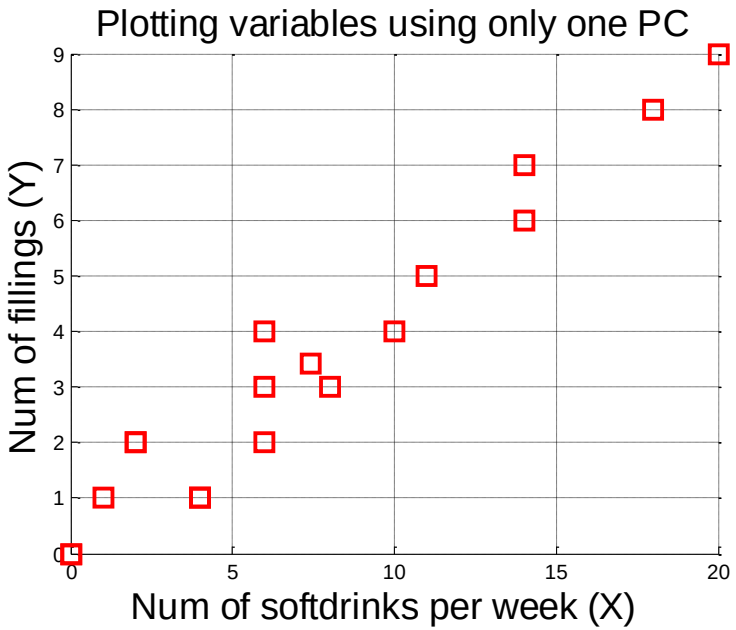
What are Principal Components (PC)?



- How to best describe this data with a single line?
  - Principal component 1 represents vector/line that explains highest variance
  - Principal component 2 represents vector/line that explains second variance
- Principal components are always perpendicular to each other

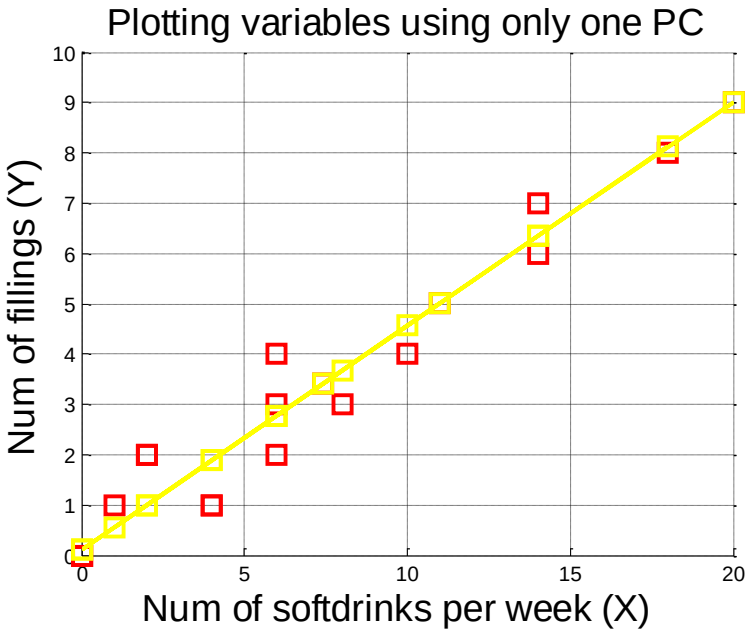
Principal Component Analysis (PCA) calculates the vectors that describe the highest variance

Reducing the data set using Principal Components



Raw data

(100% of variance)



Transformed the data using 1 PC

(98.5% of variance)

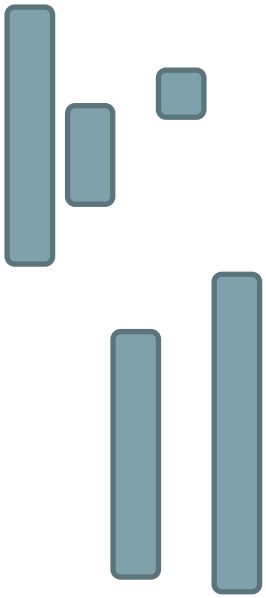
PCA can be used to reduce dimensionality of data set by identifying the key patterns within the data

The importance of scaling

- Scaling can help transform the data to a suitable form for analysis.
  - Essential when analysing variables that are measured in different units. i.e Volume ranging from 10,000-20,000 Litres compared to pH ranging from 6.5 – 7.1
  - Allows variables with small variances to have equal weight during data analysis

Raw data

$$X$$

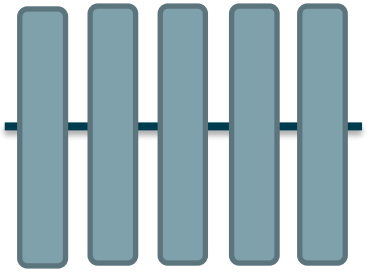


Mean centered data:

$$X - \bar{X}$$

Mean centered data and scaled data:

$$\frac{X - \bar{X}}{std(X)}$$

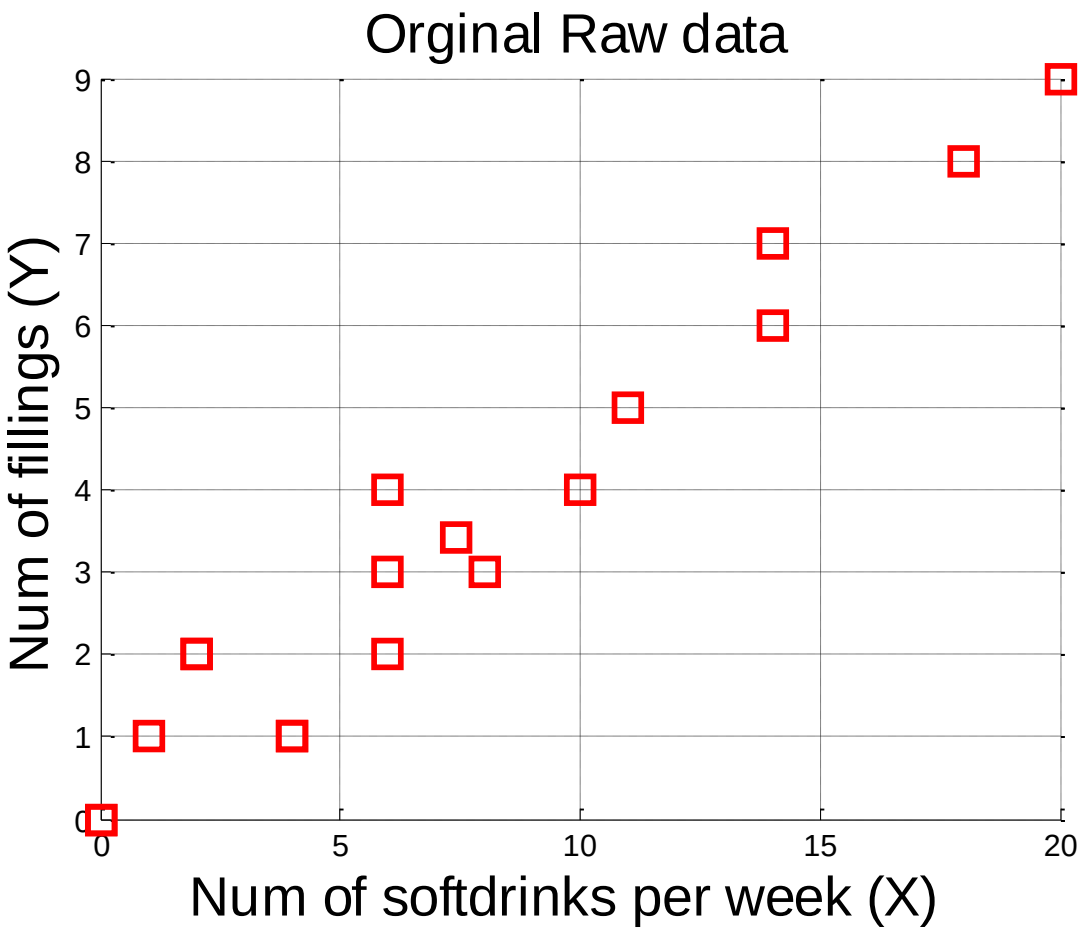


Also called normalising data or autoscaling

Performing PCA on basic data set

Num of soft drinks (X)	Num of fillings (Y)
6	2
14	6
20	9
10	4
4	1
0	0
11	5
6	3
8	3
14	7
18	8
2	2
6	4
0	0
2	2
1	1
4	1

Raw data has two Variables/Dimensions

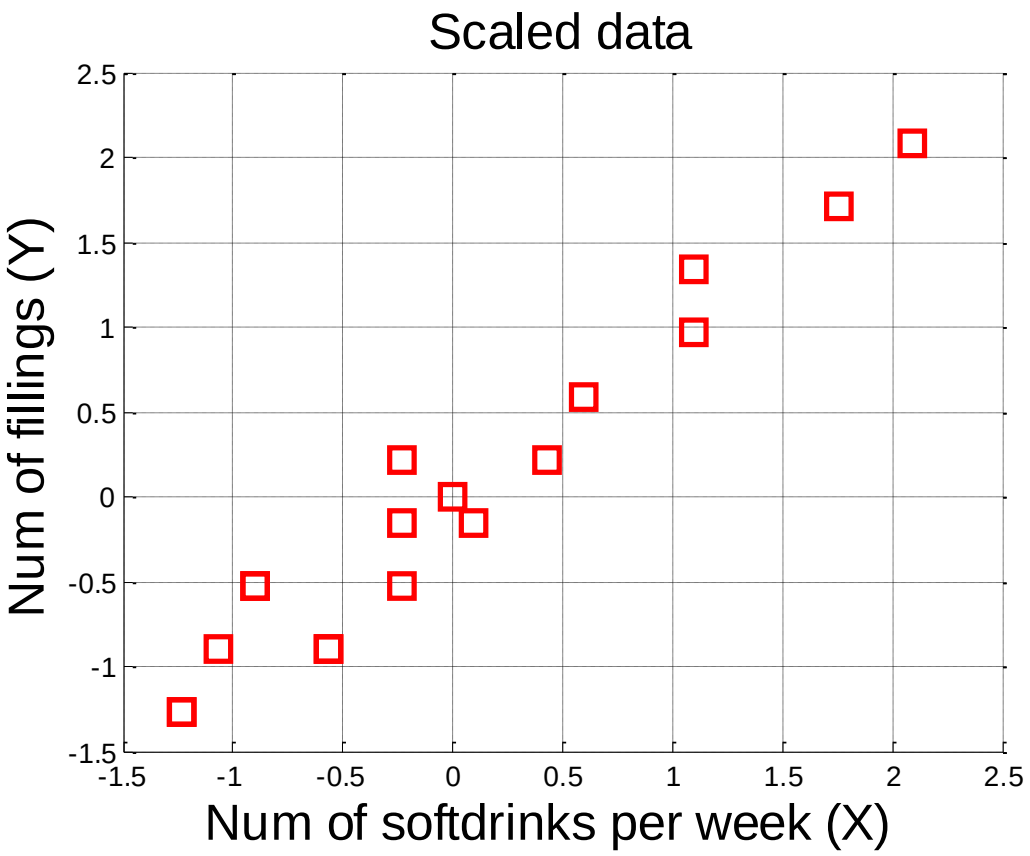


Is there a correlation/relationship between number of fillings a patient has compared to how many minutes they brush their teeth?

Step -1 Mean, center and scale the data

Mean, center and scale data

Num of soft drinks (X)	Num of fillings (Y)
-0.23	-0.51
1.06	0.94
2.03	2.02
0.42	0.21
-0.55	-0.87
-1.19	-1.23
0.58	0.57
-0.23	-0.15
0.09	-0.15
1.06	1.30
1.71	1.66
-0.87	-0.51
-0.23	0.21
-1.19	-1.23
-0.87	-0.51
-1.03	-0.87
-0.55	-0.87



Subtract mean and std from each column

X_{mean} = 7.41 *std(X)=6.20*

Y_{mean} = 3.41 *std(X)=2.76*

Formula to Mean, center and scale data

$$X_{scaled} = \frac{(X - X_{mean})}{std(X)} \qquad Y_{scaled} = \frac{(Y - Y_{mean})}{std(X)}$$

Step 2 – Calculate basic statistics for data

Num of softdrinks (X)	Num of Fillings (Y)	$(X-X_{mean})/std(X)$	$(Y-Y_{mean})/(std(X)$	$[(X-X_{mean})/std(x)]^2$	$[(Y-Y_{mean})/std(Y)]^2$	$[(X-X_{mean})/std(X)][(Y-Y_{mean})/std(Y)]$
6.00	2.00	-0.23	-0.51	0.05	0.26	0.12
14.00	6.00	1.06	0.94	1.13	0.88	0.99
20.00	9.00	2.03	2.02	4.12	4.09	4.10
10.00	4.00	0.42	0.21	0.17	0.05	0.09
4.00	1.00	-0.55	-0.87	0.30	0.76	0.48
0.00	0.00	-1.19	-1.23	1.43	1.53	1.48
11.00	5.00	0.58	0.57	0.33	0.33	0.33
6.00	3.00	-0.23	-0.15	0.05	0.02	0.03
8.00	3.00	0.09	-0.15	0.01	0.02	-0.01
14.00	7.00	1.06	1.30	1.13	1.69	1.38
18.00	8.00	1.71	1.66	2.91	2.76	2.83
2.00	2.00	-0.87	-0.51	0.76	0.26	0.45
6.00	4.00	-0.23	0.21	0.05	0.05	-0.05
0.00	0.00	-1.19	-1.23	1.43	1.53	1.48
2.00	2.00	-0.87	-0.51	0.76	0.26	0.45
1.00	1.00	-1.03	-0.87	1.07	0.76	0.90
4.00	1.00	-0.55	-0.87	0.30	0.76	0.48
				1.00	**1.00**	**0.97**

$$X_{scaled} = \frac{(X - X_{mean})}{std(X)}$$

$$Y_{scaled} = \frac{(Y - Y_{mean})}{std(Y)}$$

Cov(X,X) Cov(Y,Y) Cov(X,Y)

$$\begin{pmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(Y,X) & Cov(Y,Y) \end{pmatrix} = \begin{pmatrix} 1 & 0.97 \\ 0.97 & 1 \end{pmatrix}$$

Step 3- Calculate the covariance matrix of the scaled data (Correlation Matrix)

- Covariance of two variables = $\begin{pmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(Y,X) & Cov(Y,Y) \end{pmatrix}$
- Therefore covariance matrix of sample dataset: $\begin{pmatrix} 1 & 0.97 \\ 0.97 & 1 \end{pmatrix}$
- What does the correlation (covariance) matrix tell us?

Mean, centered and scaled data will always have a covariance of 1.

i.e Spread of X data is equal Y data

$\begin{pmatrix} 1 & 0.97 \\ 0.97 & 1 \end{pmatrix}$

Cov(X,Y) is **Positively** correlated i.e an increase in number of soft-drinks results in an increase in the number of fillings

Cov(X,Y) = Cov(Y,X)

N.B If data is scaled Covariance values vary between
-1: Strong negative correlation
-0.5: Moderation negative correlation
0 No Correlation
0.5: Moderation positive correlation
1 : Strong positive correlation

Eigenvalues and Eigenvectors from covariance matrix enable calculation of Principal components from PCA

Eigenvectors are calculated from Covariance matrix of dataset:

$$\begin{pmatrix} 1 & 0.97 \\ 0.97 & 1 \end{pmatrix}$$

Matlab inbuilt function calculates both Eigenvectors and Eigenvalues:

```
[V,D] =eigs(Covariance_matrix_cal,2, 'LA')
```

Eigenvectors (V) = $\begin{pmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{pmatrix}$ Eigenvalues (D) = $\begin{pmatrix} 1.97 & 0 \\ 0 & 0.029 \end{pmatrix}$

Note: Eigenvectors = Loadings

Eigenvalues = Explained variance

Eigenvectors are perpendicular to each other

Determining the % variance explained by each Principle Component

- Eigenvalues = $\begin{matrix} \text{PC-1} & \text{PC-2} \end{matrix}$ $\begin{pmatrix} 1.97 & 0 \\ 0 & 0.029 \end{pmatrix}$

- Variance for Principle Component 1:

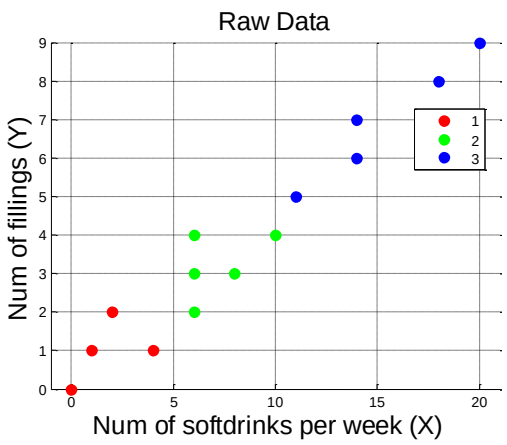
$$\frac{1.97}{1.97+0.029}\times 100 = 98.5\%$$

- Variance for Principle Component 2 =

$$\frac{0.029}{1.97+0.029}\times 100 = 1.5\%$$

Overview of PCA steps and outputs

Raw data



Scale data & Perform PCA

PCA results

Percent Variance Captured by PCA Model

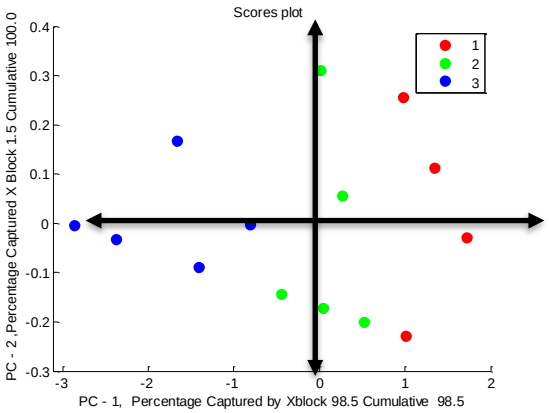
Principal Component Number	Eigenvalue of Cov(X)	% Variance Captured This PC	% Variance Captured Total
1	1.97e+00	98.51	98.51
2	2.98e-02	1.49	100.00

Max num of PC's = Num of variables (i.e 2)

1st PC accounts for 98.51% of variance in data
and 2nd PC accounts for 1.49% of variance

Scores Plot

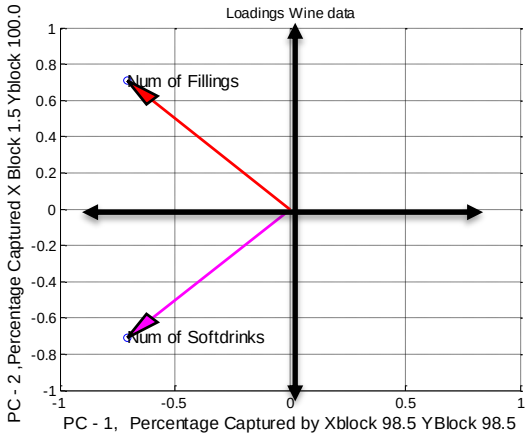
Shows samples represented in PC space



Scores = Loadings Vector x Scaled Data

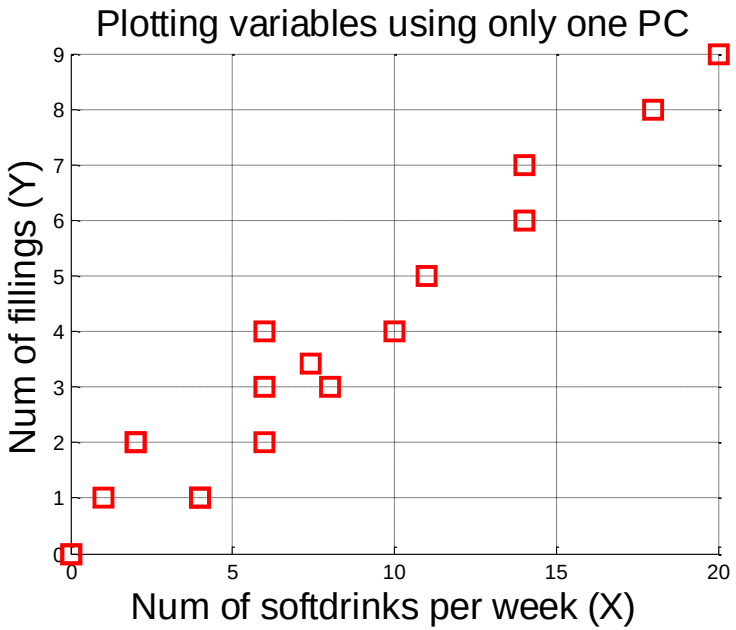
Loadings Plot

Shows variables represented in PC space



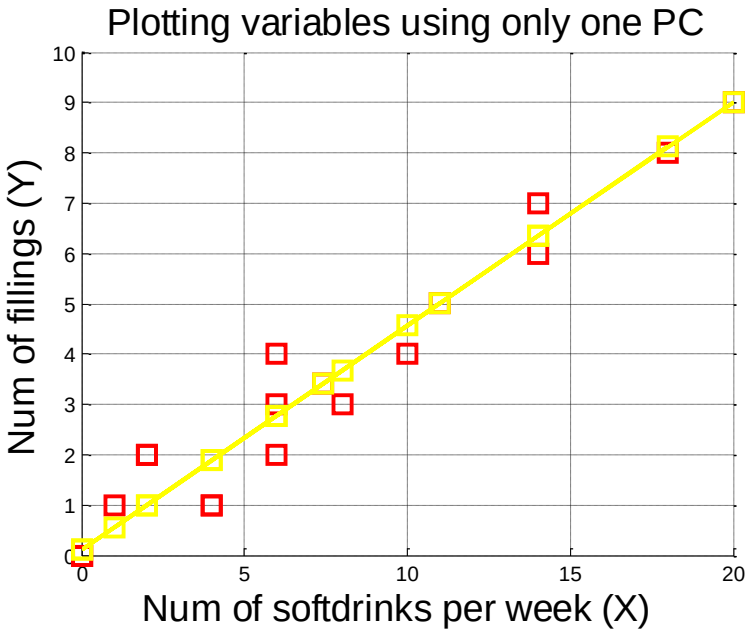
PC-1 =X-axis PC-2 = Y-axis

Reducing the data set using Principal Components



Raw data

(100% of variance)



Transformed the data using 1 PC

(98.5% of variance)

PCA can be used to reduce dimensionality of data set by identifying the key patterns within the data

Overview of PCA steps and outputs

Raw data

Batch ref	Temperature (oC)	pH Set-point	SD (cells x10^6/mL)	Titre (mg/L)	Classification
1	34.1	7.2	0.21	2199	3
2	34.2	7.15	0.25	2500	3
3	33.8	7.23	0.3	2233	3
4	33.95	7.17	0.4	2439	3
5	34.05	7.16	0.23	2350	3
6	33.87	6.72	0.75	4112	2
7	33.89	6.59	0.65	4239	2
8	34.12	6.63	0.78	4549	2
9	34.13	6.65	0.65	3678	2

Scale data & Perform PCA

PCA results

Percent Variance Captured by PCA Model

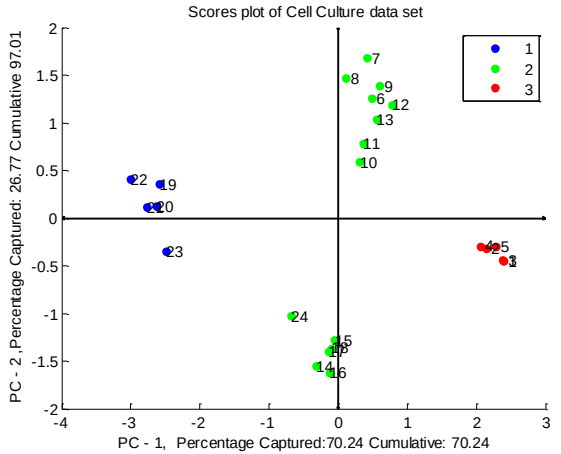
Principal Component Number	Eigenvalue of Cov(X)	% Variance Captured This PC	% Variance Captured Total
1	2.81e+00	70.24	70.24
2	1.07e+00	26.77	97.01
3	9.18e-02	2.30	99.31
4	2.77e-02	0.69	100.00

Max num of PC's = Num of variables (i.e 4)

1st PC accounts for 70.24% of variance in data and 2nd PC accounts for 26.77% of variance

Scores Plot

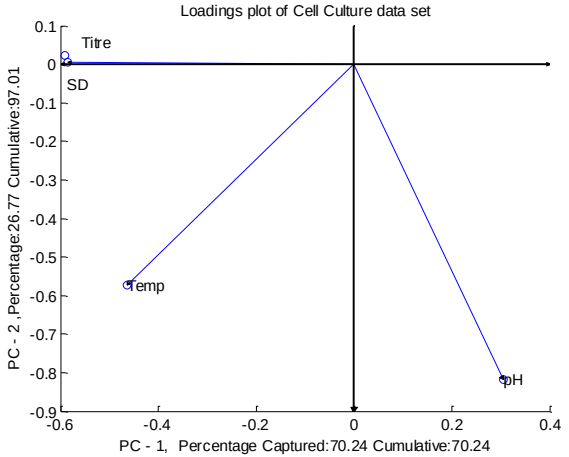
Shows samples represented in PC space



Scores = Loadings Vector x Scaled Data

Loadings Plot

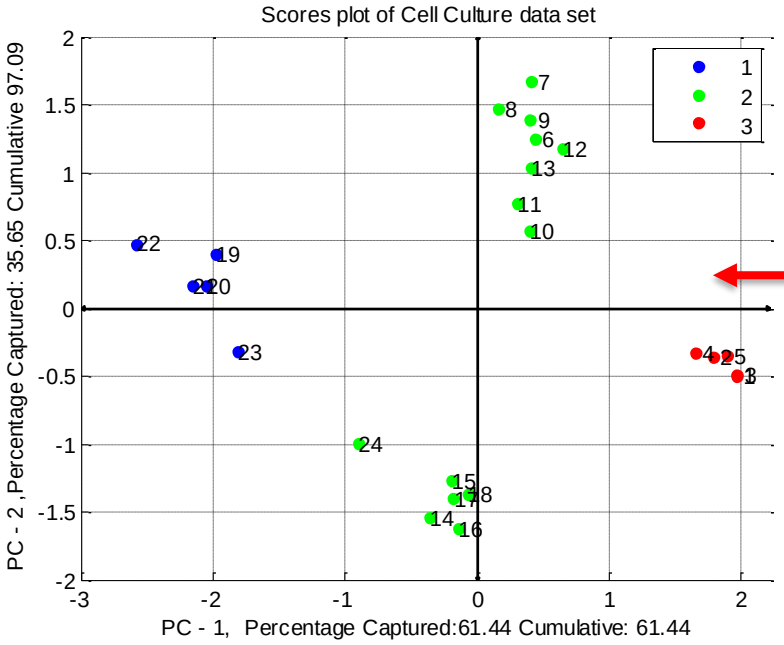
Shows variables represented in PC space



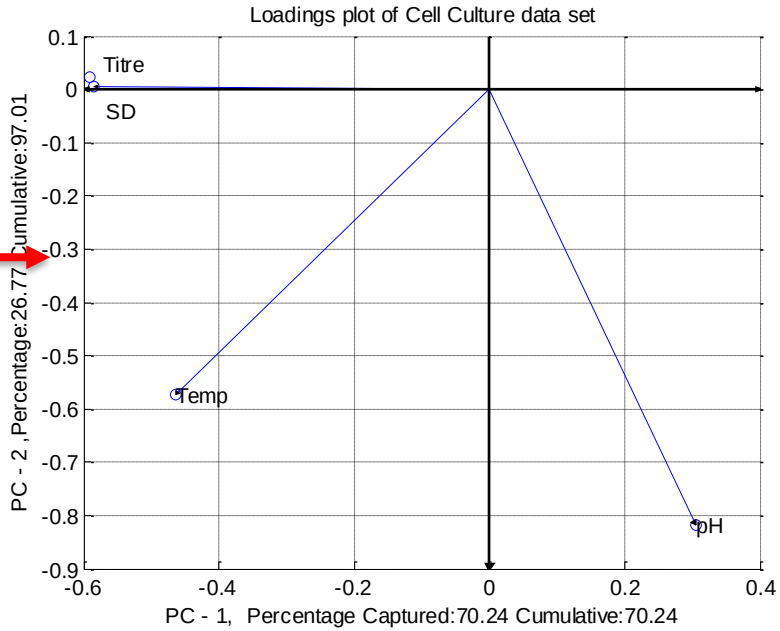
PC-1 =X-axis PC-2 = Y-axis

Cell culture MVDA example

Scores plot



Loadings plot



- Scores plot represents the batches or samples
  - 24 batches = 24 scores

- Loadings plot represents the variables
  - Four variables (Temperature, pH and SD, Titre)

Analysing the scores and loadings plots identifies correlations between samples and variables

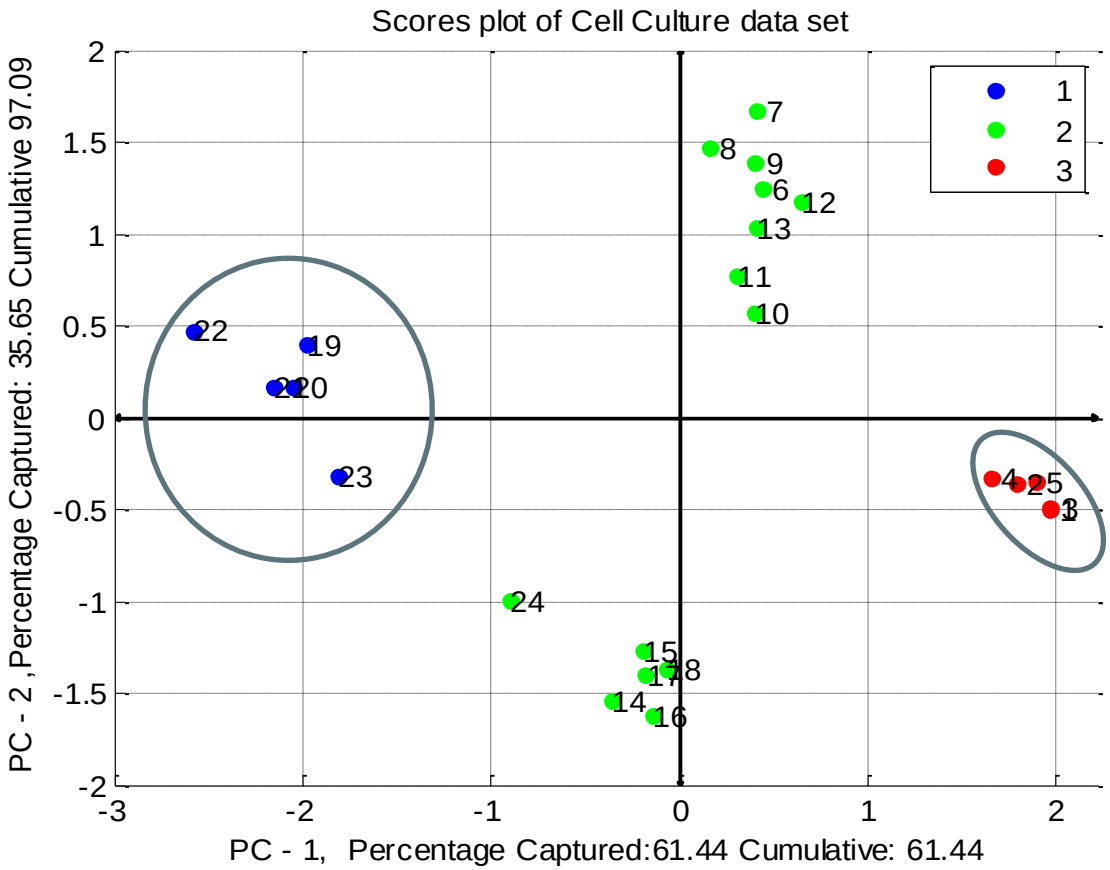
Scores plots

Batch classes:
1: Good Batches
2: Average batches
3: Bad Batches

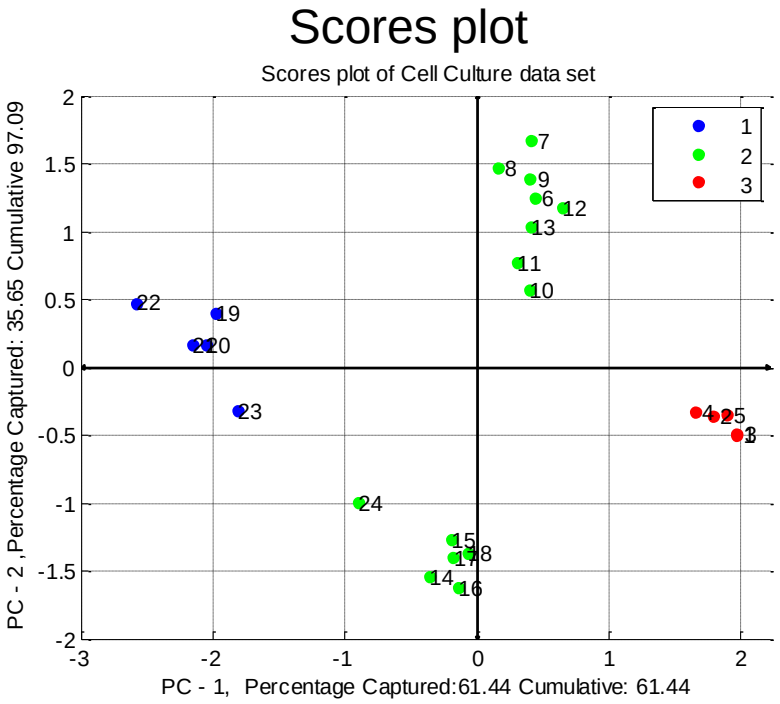
Batches clustered together on a scores plot represent batches with similar characteristics

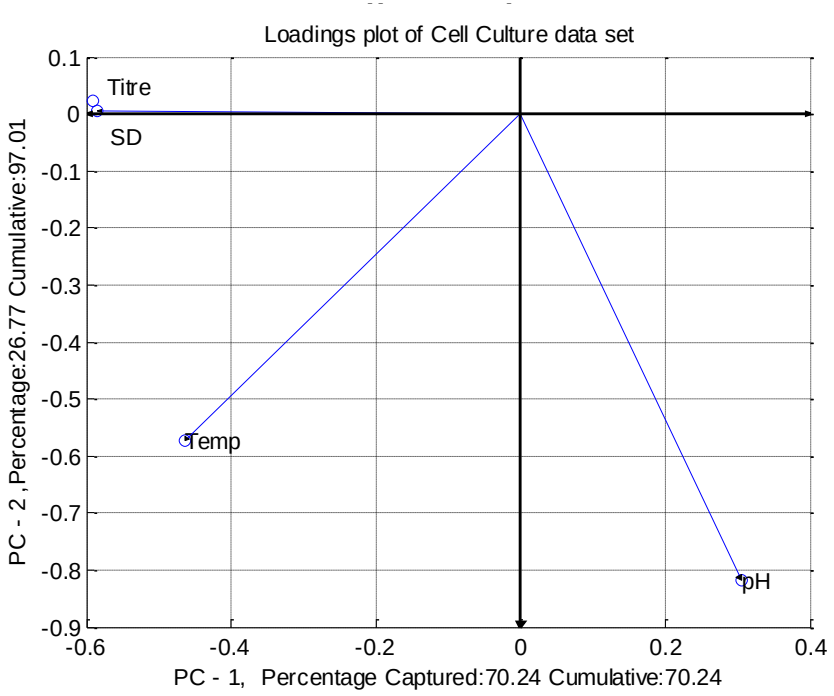
What is similar about batches 19, 20,21,22, 23?

What is similar about batches 1, 2, 3, 4, 5?



Cell culture MVDA example





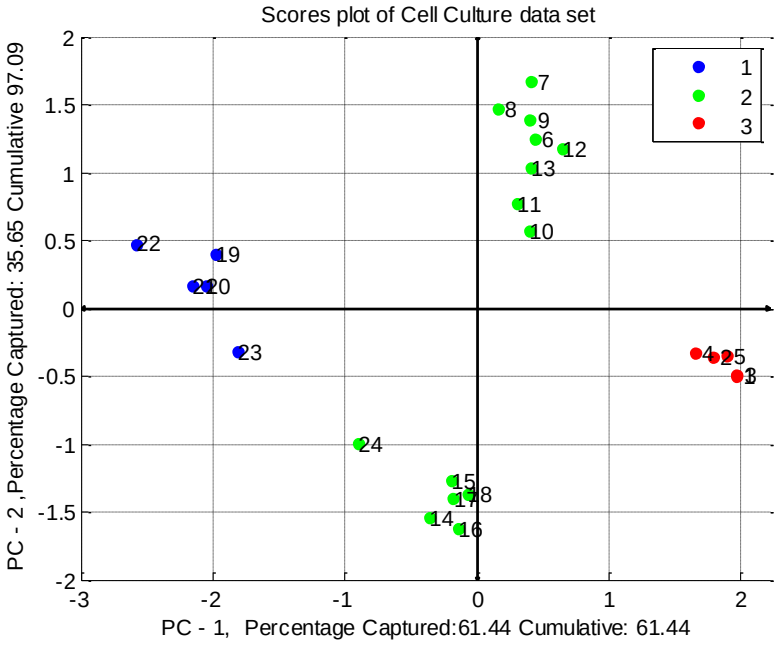
Position of batches on **Scores plot** compared to **Loadings plot** represents correlation

SD and Titre: Mainly represented by **PC-1** so batches to the left of the origin represent batches that are highly correlated with Seeding Density and Titre(**SD and Titre**)

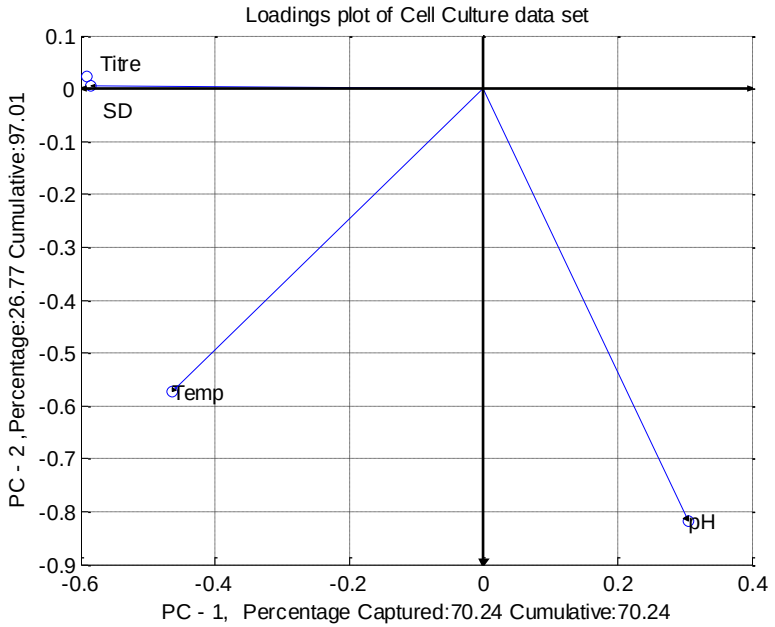
- Batches 19-23 all have high Seeding Density and high titre
- Batches 1-5 have low Seeding Density and low titre

Cell culture MVDA example

Scores plot



Loadings plot

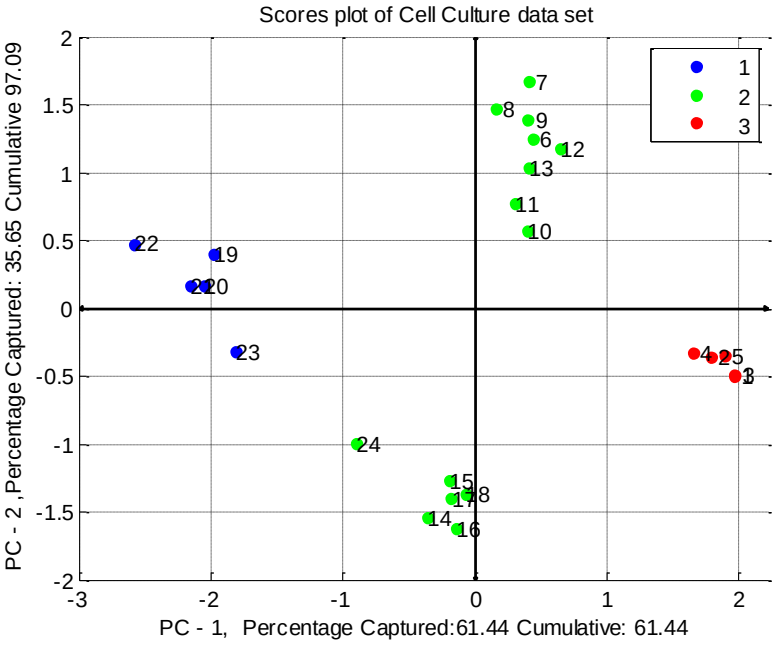


Temp: Equally represented by **PC-1** and **PC-2** so batches to the left and bottom quadrant are highly correlated with high temperature

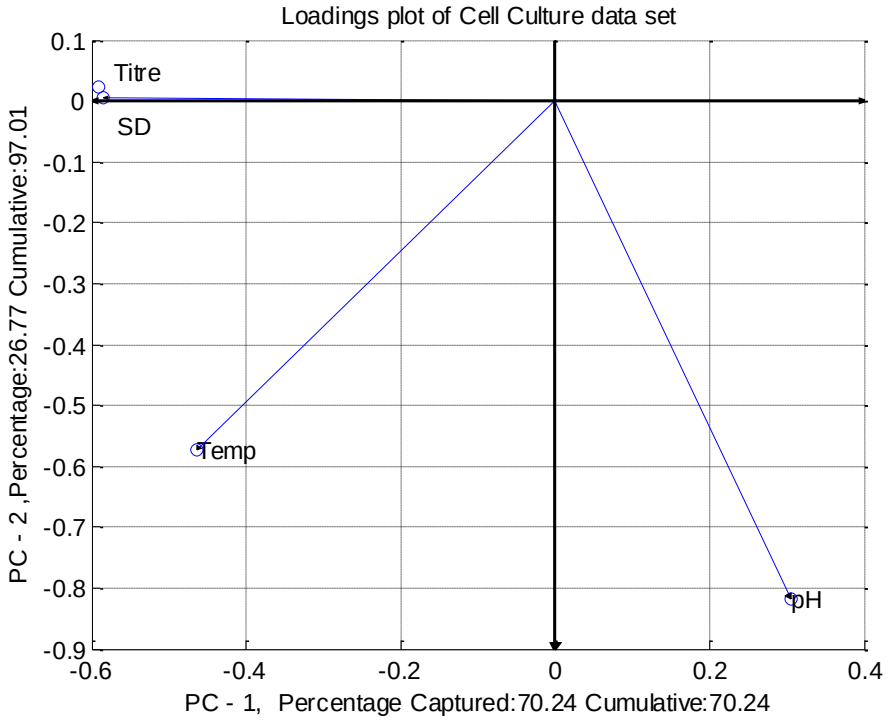
- Batches 19-23 all have high Temperature
- Batches 1-5 all have low Temperature
- Batches 6,7,8,9,10,11,12, have low Temperature
- Batches 15, 16, 17, 18, 24 have high Temperature

Cell culture MVDA example

Scores plot



Loadings plot

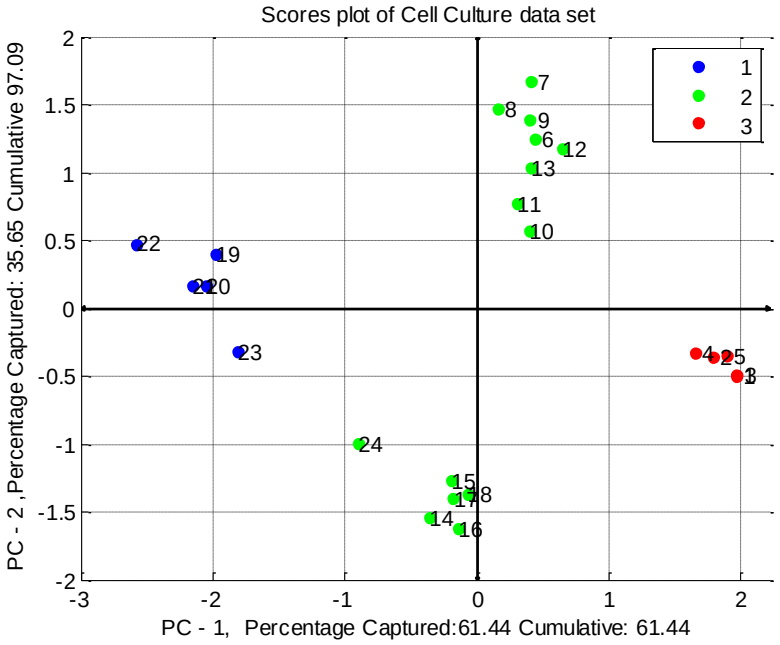


pH: Mainly represented by **PC-2** in addition to **PC-1** as batches to the right and bottom quadrant are highly correlated with high pH

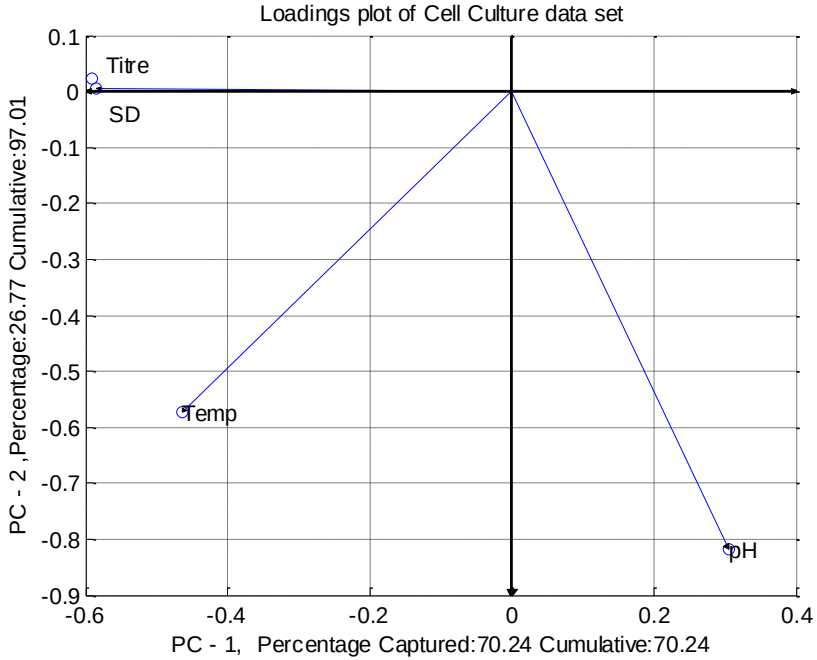
- Batches 19-23 all have low pH
- Batches 1-5 all have high pH
- Batches 6,7,8,9,10,11,12, have low pH
- Batches 24, 15, 16, 17, 18, 24 have high pH

Cell culture MVDA example

Scores plot



Loadings plot



Batches 1-5: Low Seeding density, Low Temperature, High pH and Low Titre

Batches 6-12: Medium Seeding density, Low Temperature, Low pH and Medium Titre

Batches 14-18, 24: Medium Seeding density, High Temperature, high pH and Medium Titre

Batches 19-23: High Seeding density, High Temperature, Low pH and High Titre

Quick overview of MVDA techniques covered

Raw Data (X)

Scale data

Covariance Matrix

Columns

$$X = \begin{bmatrix} 1.67 & 59.0 & 25 \\ 1.81 & 80.6 & 14 \\ 1.66 & 70.1 & 22 \\ 1.30 & 90.1 & 2 \end{bmatrix}$$ Rows

X= [4x3]

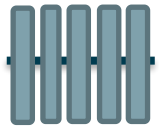
Rows -> Samples/Batches
Columns -> Variables

[X]= (Rows x Columns)

Raw data X

Subtract mean $X - \bar{X}$

Divide by Std $\frac{X - \bar{X}}{std(X)}$



Scaling is necessary to ensure variables with different units have equal weight during data analysis

Principal Component Analysis (PCA)

Scores plot

Highlights correlations between samples/batches

Loadings plot

Shows relationship between variables

PCA: Analysing the scores and loadings plot together, enables correlations to be developed within the data

Cov(X,Y)= $\begin{pmatrix} 1 & 0.97 \\ 0.97 & 1 \end{pmatrix}$

X and Y are highly positive correlated

Multivariate data analysis: 48 batch records

- **Big data (Available on Moodle)**
  - Download and unzip the folder on desktop
  - Open text file called "Bioreactor_data_headers_v1.csv "
- **Overview of data set**
  - 48 mammalian cell culture batches were operated at different temperatures and pH set points and initial RPMS
  - Final Titre and Aggregation was recorded for each batch and classed as
    - 1: Good Batches for Aggregation below 5 g/L (Classification = 1)
    - 2: Average Batches Aggregation between 3-5 g/L g/L (Classification = 2 in Column 6)
    - 3: Bad Batches for aggregation over 8 g/L (Classification = 3)

pH	Temp	RPM	Titre	Aggregation	Classification
6.2309	23.781	2433.8	0.39391	12.133	3
6.319	26.805	2614.2	0.59547	5.614	2
6.1867	13.402	2396.9	-1.1285	10.902	3
6.2093	37.635	2202.6	0.96203	10.18	3
5.9788	27.495	2442.6	-0.19477	8.5016	3
6.0208	32.504	2094.8	2.9491	10.453	3
5.9951	41.044	2018.5	3.0823	10.63	3
6.0619	34.635	2244.5	2.3326	10.755	3
6.2883	31.368	2047.3	1.8792	9.3396	3
6.0482	28.663	2010.7	2.6717	9.5146	3
6.7819	32.798	2063	4.5927	6.777	2
6.5332	38.517	1980.3	4.4523	5.7471	2
6.0466	24.293	1972	2.4619	7.2148	2
6.2408	34.082	1992.5	3.3688	7.9356	3
6.4596	39.85	1770	1.9887	9.2769	3
6.1928	25.061	1721.4	3.9809	5.6065	2
6.4492	39.173	1569.8	4.5535	6.021	2

- Data set = [48x6] i.e 24 batches and 5 variables

Cell culture MVDA example

Research question:

Can PCA identify the process conditions that influences final aggregation and Titre?

What are the main factors that result in high/low titres and Aggregation?

Are there any obvious groups in the data set?

PCA Example Application

	Wine ref	Alcohol %	Total Sulfur	Residual Sugar	Quality	Class
0	1	19.046	1123.50	20.8850	11	1
1	2	17.931	966.89	25.7460	8	1
2	3	19.047	1351.20	8.7366	6	1
3	4	16.395	1293.90	14.2490	3	1
4	5	17.792	1226.60	40.8580	0	1

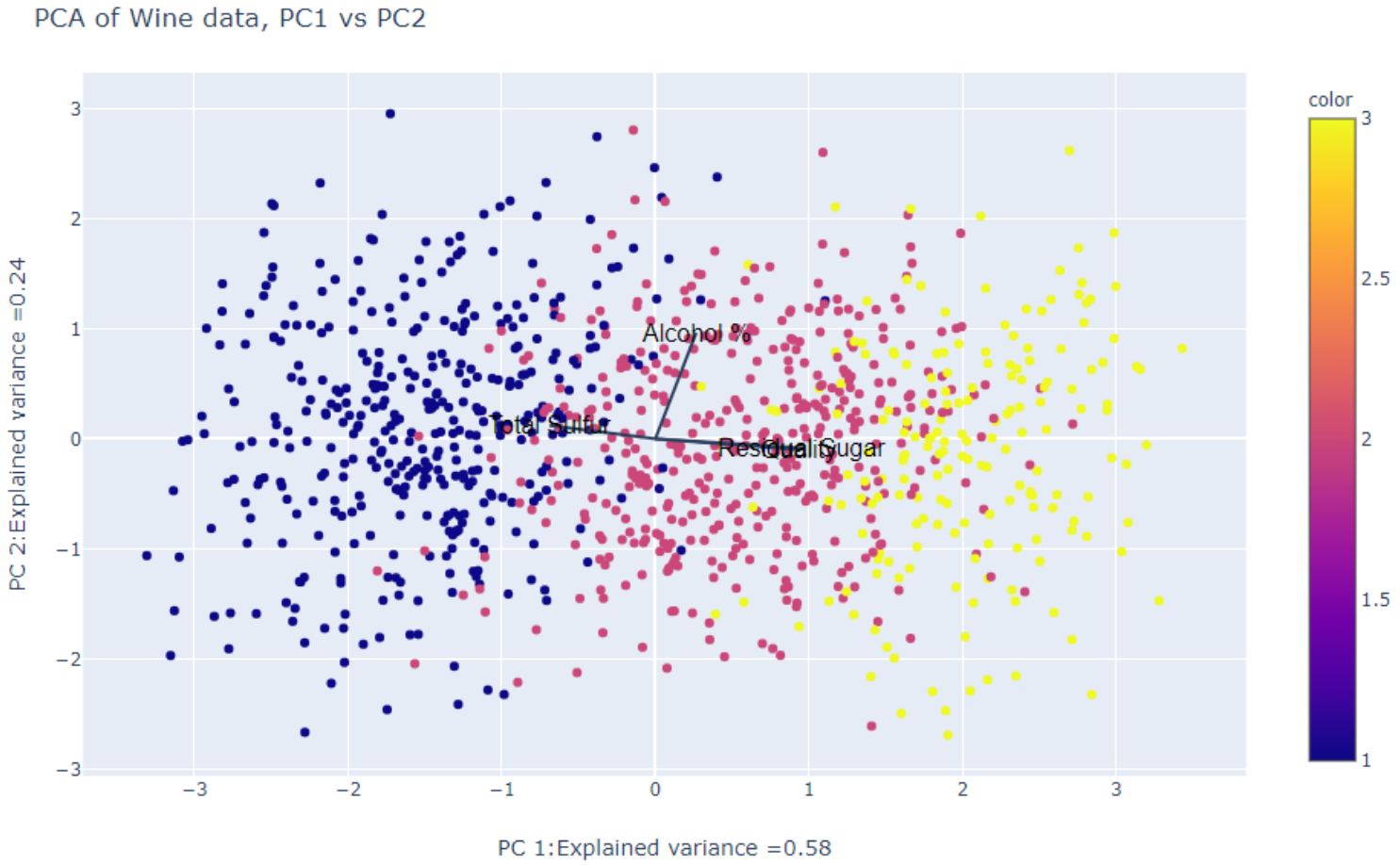
Data collected on 1000 wines, 5 different characteristics recorded and over all wine quality tested

1 Wine classification

- Yellow -> Great wine
- Purple -> Okay wine
- Blue -> Bad wine

Research Question: Can PCA determine which variables affect wine quality?

PCA Biplot of Wine data set



Based on PCA – Residual Sugar is highly correlated with Quality and Total Sulfur is negatively correlated with Quality, also Alcohol % is not shown to be correlated with quality (i.e less Sulfur more Sugar = statistically better wine ☺

“Big Data” - The Future Digital Revolution

Big data (**official)** definition (6 V’s):

- Volume
- Variety
- Veracity
- Velocity
- Variability
- Value

2.5 Gigabytes of data freely available to download:

- 111,189 Rows x 2789 Columns
- On-line variables
- Off-line variables
- Raman spectroscopy

Data for: Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process

Published: 1 Jul 2019 | **Version 1** | **DOI:** 10.17632/pdnjz7zz5x.1

Contributor(s): Stephen Goldrick

Description of this data

This data was generated using an advanced mathematical simulation of a 100,000 litre penicillin fermentation system referenced as IndPenSim. All details describing the simulation are available on the following website: www.industrialpenicillinsimulation.com. IndPenSim is the first simulation to include a realistic simulated Raman spectroscopy device for the purpose of developing, evaluating and implementation of advanced and innovative control solutions applicable to biotechnology facilities. This data set generated by IndPenSim represents the biggest data set available for advanced data analytics and contains 100 batches with all available process and Raman spectroscopy measurements (~2.5 GB). This data is highly suitable for the development of big data analytics, machine learning (ML) or artificial intelligence (AI) algorithms applicable to the biopharmaceutical industry. The 100 batches are controlled using different control strategies and different batch lengths representing a typical Biopharmaceutical manufacturing facility:

Batches 1-30: Controlled by recipe driven approach
Batches 31-60: Controlled by operators
Batches 61:90: Controlled by an Advanced Process Control (APC) solution using the Raman spectroscopy
Batches 91:100: Contain faults resulting in process deviations.

Please reference:

Goldrick S., Stefan, A., Lovett D., Montague G., Lennox B. (2015) The development of an industrial-scale fed-batch fermentation simulation Journal of Biotechnology, 193:70-82.
and
Goldrick S., Duran-Villalobos C., K. Jankauskas, Lovett D., Farid S. S, Lennox B., (2019) Modern day control challenges for industrial-scale fermentation processes. Computers and Chemical Engineering.

Latest version

Version 1 2019-07-01

Published: 2019-07-01

DOI: 10.17632/pdnjz7zz5x.1

Cite this dataset

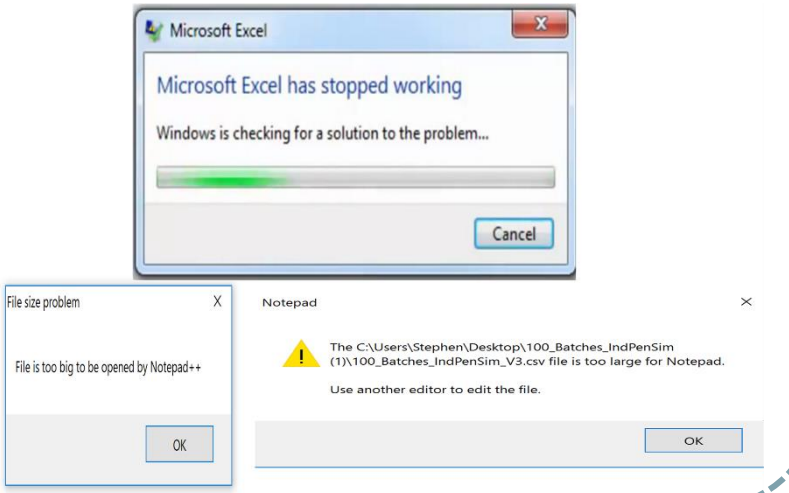
Goldrick, Stephen (2019), “Data for: Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process”, Mendeley Data, v1

http://dx.doi.org/10.17632/pdnjz7zz5x.1

Statistics

Views: **155** Downloads: **33**

Big data (**unofficial)** definition**:**



www.industrialpenicillinsimulation.com

Download Jupyter-notebook from website

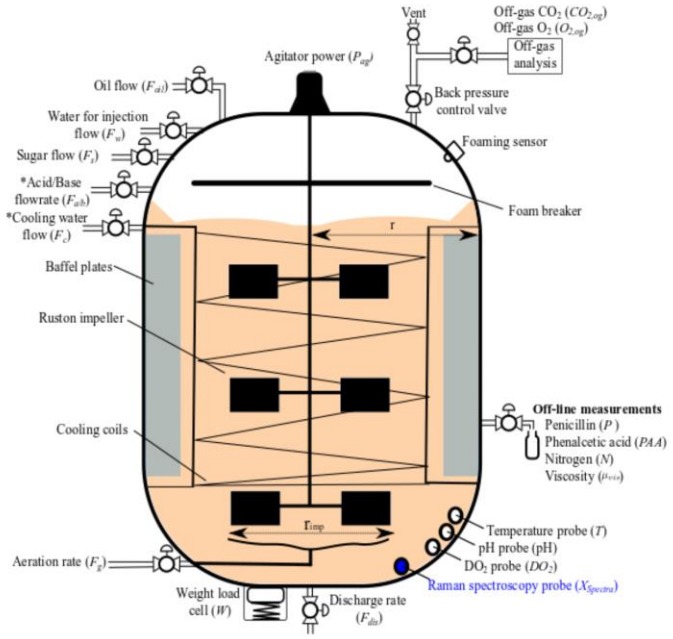
Industrial-Scale Penicillin Simulation (V2)

Realistic simulation of a 100,000 litre penicillin fermentation including Raman Spectroscopy Device

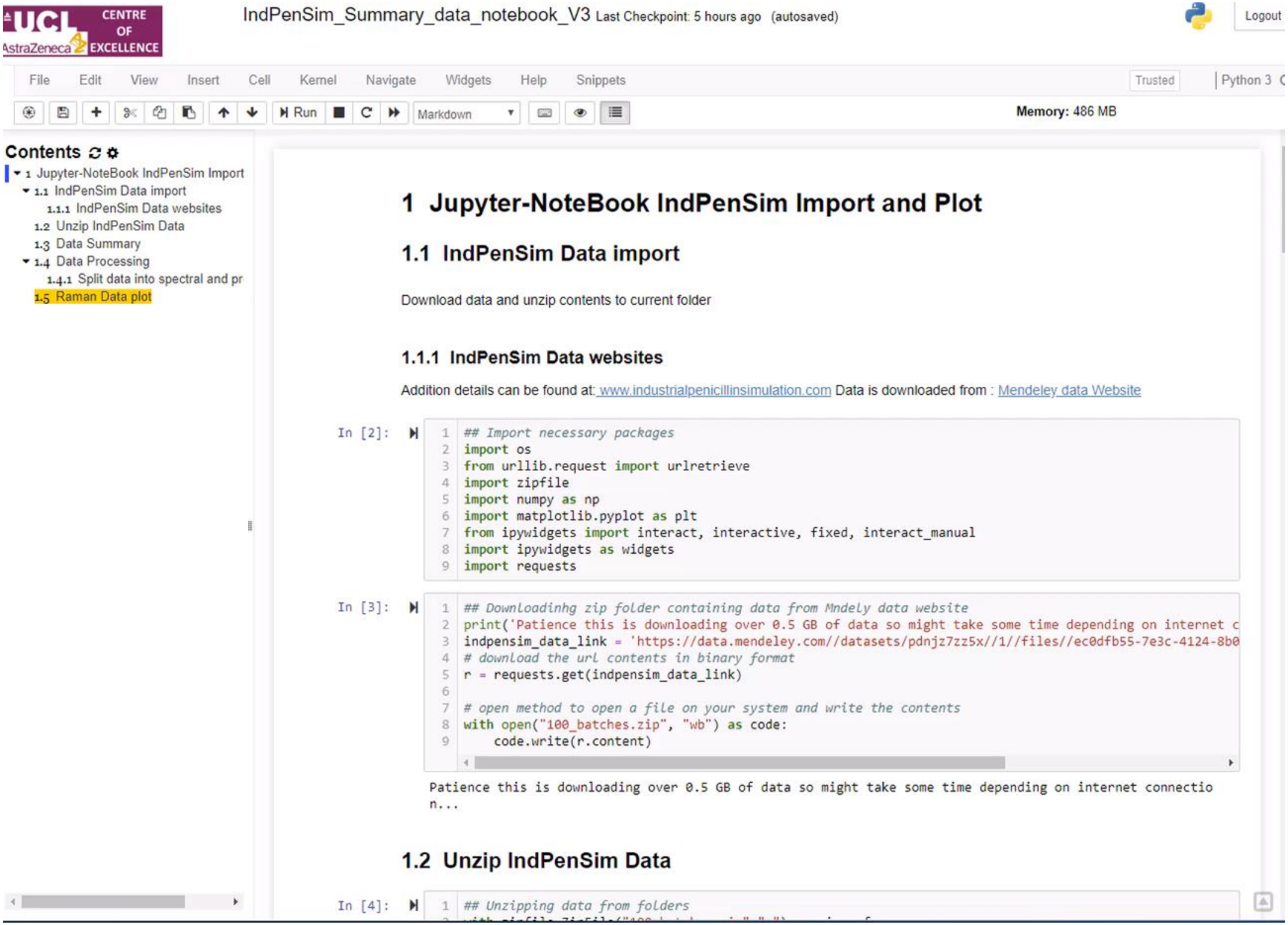
DOWNLOAD *INDPENSIM* SIMULATION | DOWNLOAD BIG DATA DOWNLOAD (100 BATCHES) | GITHUB: PYTHON - JUPYTER NOTEBOOK OF *INDPENSIM*

IndPenSim

Industrial-scale penicillin simulation (*IndPenSim*) is a first principles mathematical model of a *Penicillium chrysogenum* fermentation. The simulation was developed in Matlab and is freely available to download. The development of the simulation is discussed in the paper titled *"The development of an industrial-scale fed-batch fermentation simulation"* currently available to download: here (Journal of Biotechnology). The paper has recently been extended which include the addition of a simulated Raman spectroscopy device for the purpose of developing, evaluating and implementation of advanced and innovative control solutions applicable to biotechnology facilities. The capabilities of *IndPenSim* are demonstrated through the implementation of a QbD methodology utilising the three stages of the PAT framework. Additionally, *IndPenSim* evaluated a fault detection algorithm to detect process faults occurring on different batches recorded throughout a yearly campaign. Details of this work can be found in the paper titled: '*Modern day monitoring and control challenges outlined on an industrial-scale benchmark*



“Big Data” needs powerful software



Partial Least Squares (PLS)

- Partial least squares is a tool suitable whenever plant variables can be partitioned into cause (**X**) and effect (**Y**) values.The method may be used for regression or similarly, toPCA, reduction of the effective dimensionality of data. Theapproach works by selecting factors of cause variables in a sequence that successively maximizes the explained covariance between the cause and effect variables.

$$X = \sum T \cdot P^T + E \quad \text{Eq. 1}$$

$$Y = \sum U \cdot Q^T + F \quad \text{Eq. 2}$$

$$U = TB \quad \text{Eq. 3}$$

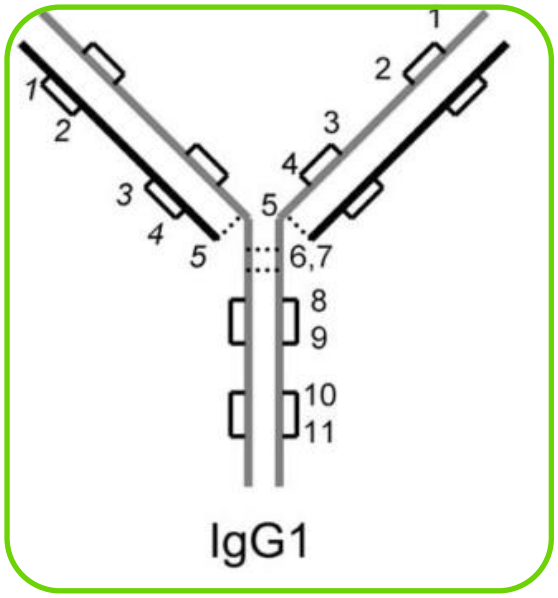
$$Y_{predicted} = X \times \beta$$

Case Study:
Multivariate Data Analysis (MVDA) to help determine product quality issues on mammalian cell culture

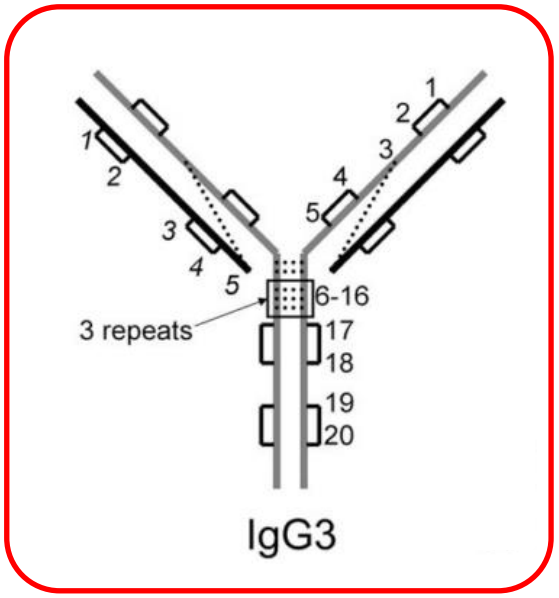
Application of MVDA for root cause determination of mAb heterogeneities on an industrial process

- A trisulfide bond (TSB) was detected on a novel recombinant antibody-peptide fusion expressed in mammalian cell culture during R&D for one of MedImmune's primary drug candidates

Correct folding



Incorrect folding - TSB



What process changes result in TSB formation?

Design of Experiment to investigate product heterogeneities

DoE Design: 3-Level Fractional Factorial (43 Cell culture runs)

Factors Manipulated:

Temperature: 34, 35.5, 37°C

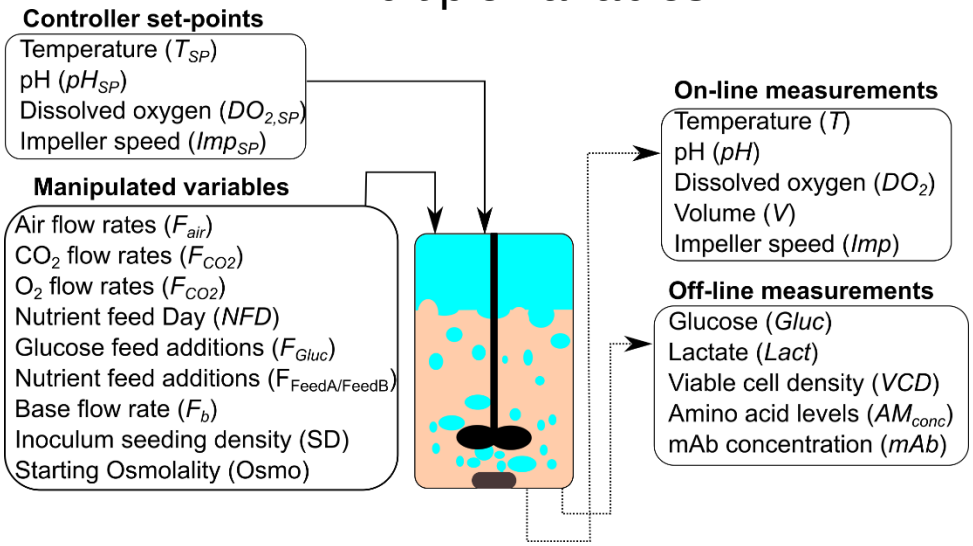
pH: 6.8-7.2

Initial Nutrient Feed Day: Day -1,2,3,4

Nutrient Feed Volume: 80-120%

Seeding Density: 50-150%

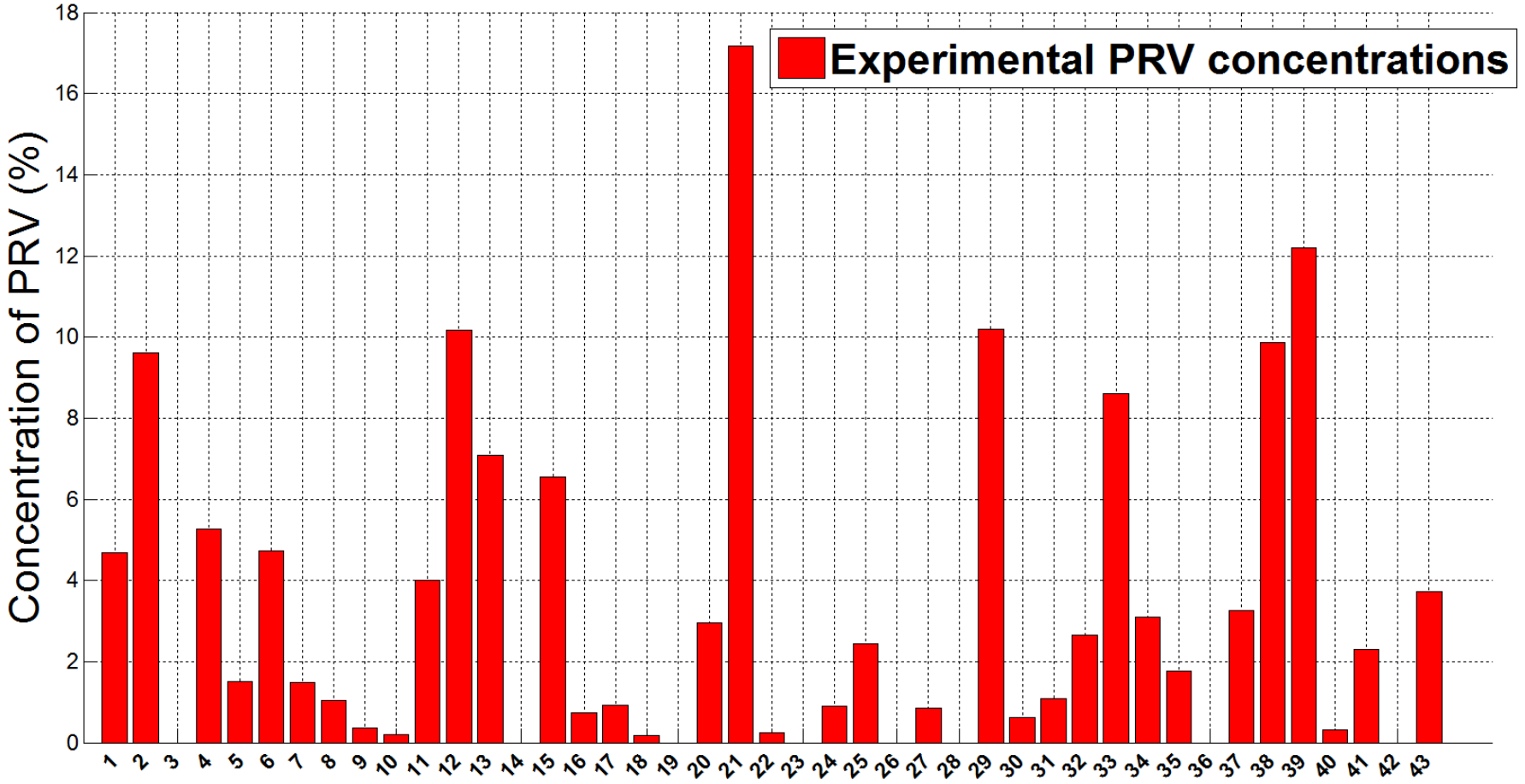
Complex process with multiple variables



- Difficult challenge
  - Consider a single variable (pH) recorded every 10 seconds for 14 days for each vessel
  - Data size equals 1 x 8640 x 14 = 1,209,600 data points
  - 25 variables therefore 25 x 1,209,600 = Massive Data set

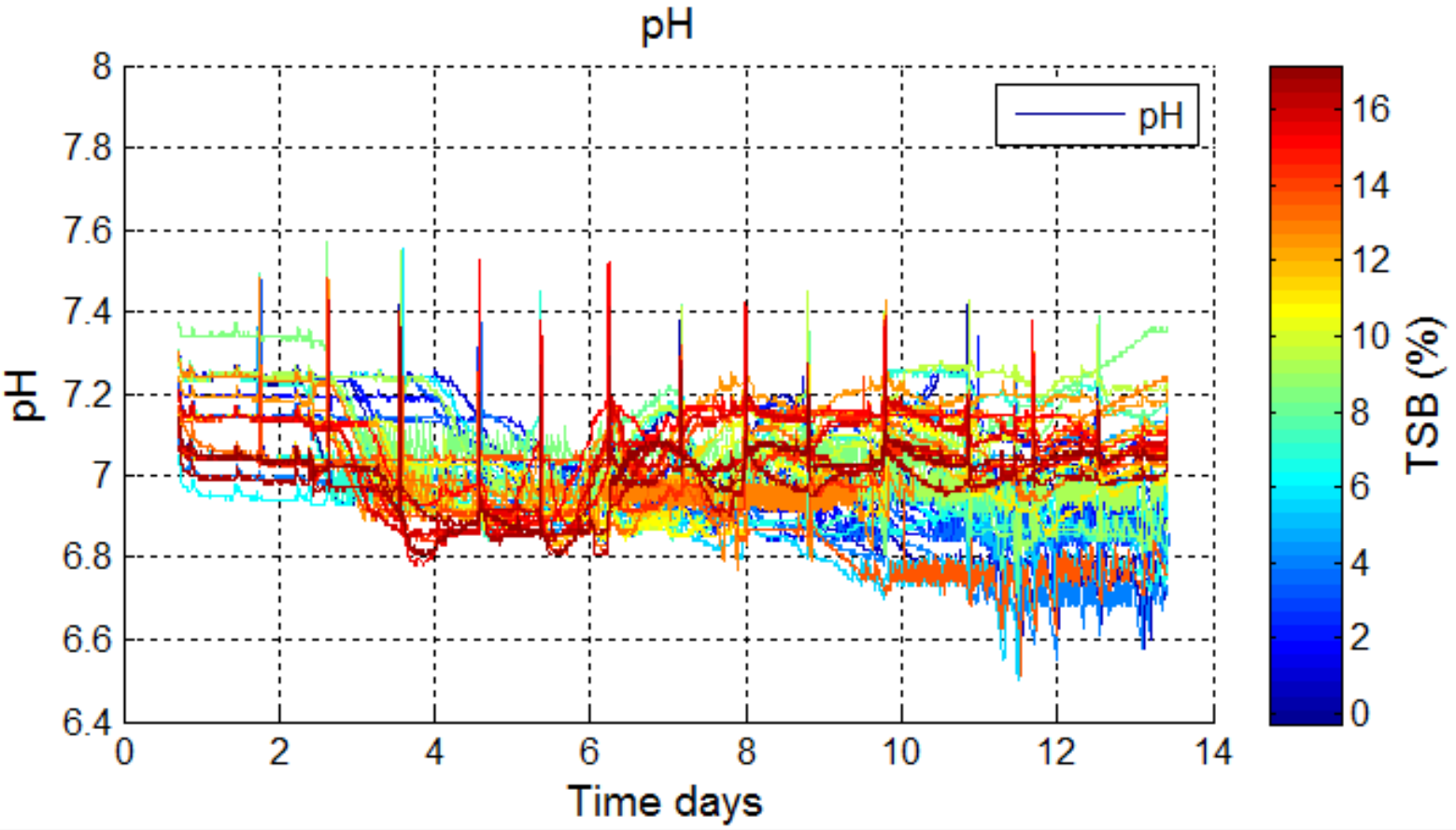
MVDA is necessary to analysis this complex data set

Concentration of TSB for 43 culture runs



What are the key process variables that are driving high TSB concentrations in these fermentations?

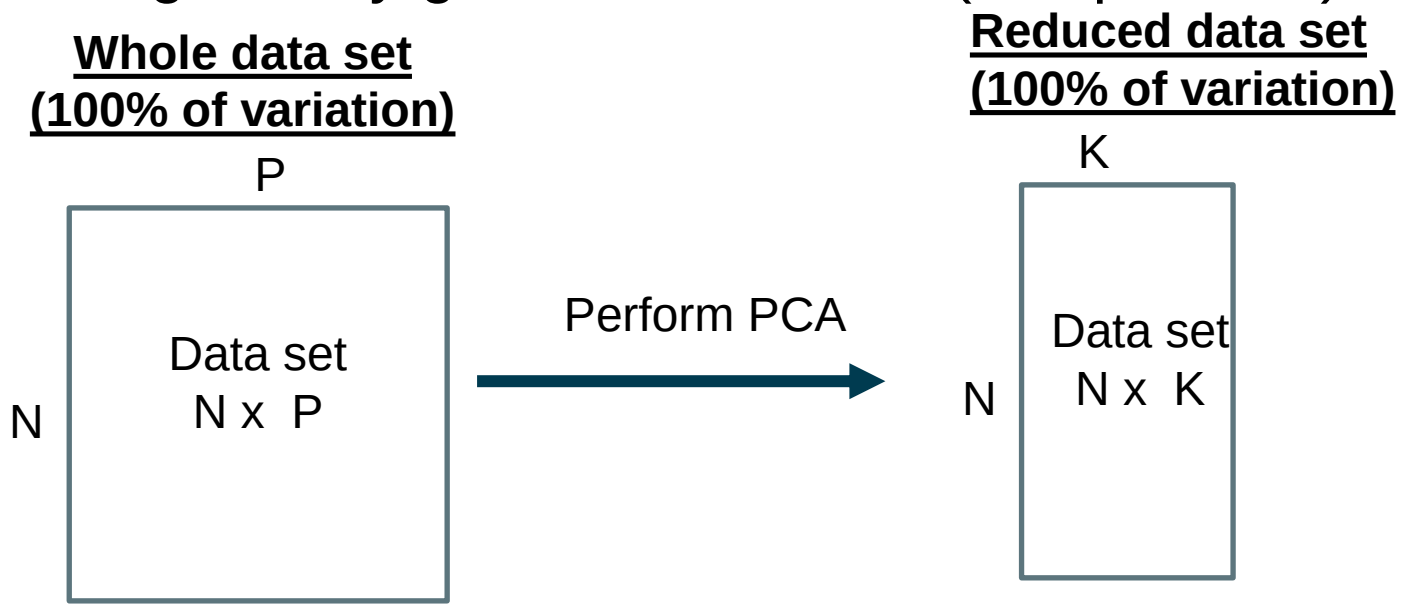
Do we need MVDA to analyse this data set?



Analysing one factor at a time is inefficient and can lead to misleading conclusions

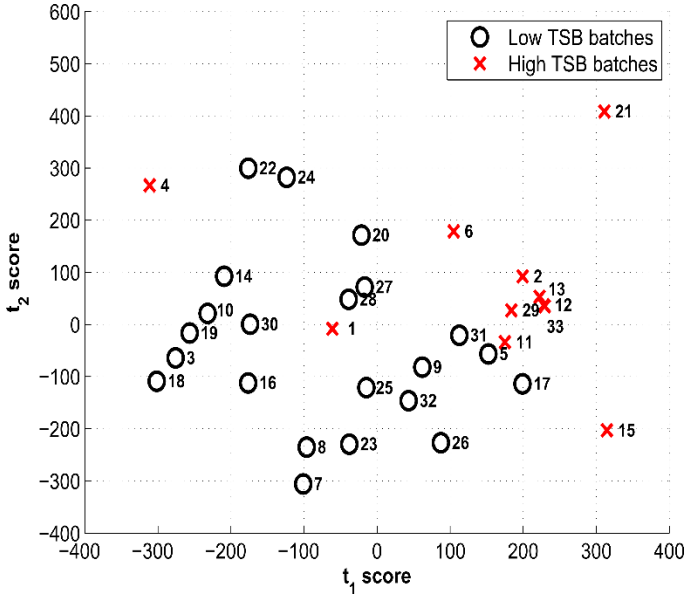
Complex data set analysed through Principal Component Analysis (PCA)

- PCA is technique that is suitable to analysis complex data sets by reducing the dimensionality of the data set
  - Essentially summarising the main sources of variability through newly generated vectors (components)

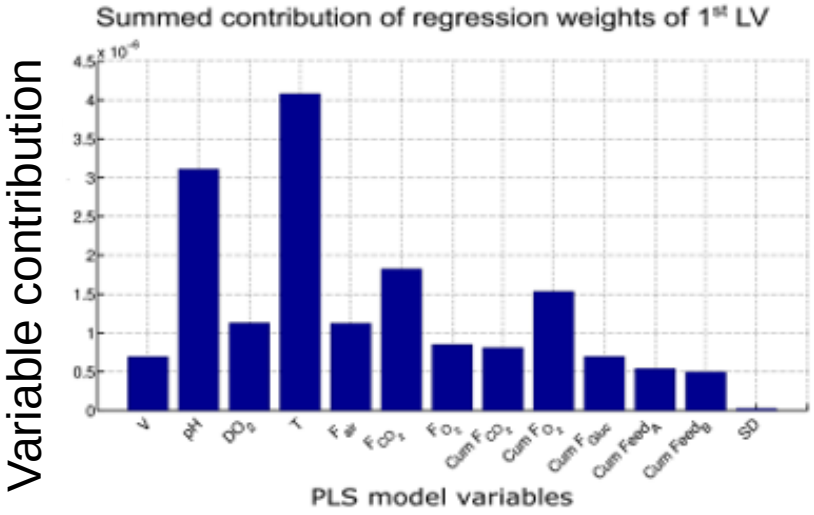


Principal Component Analysis of TSB problem

1st and 2nd principal components



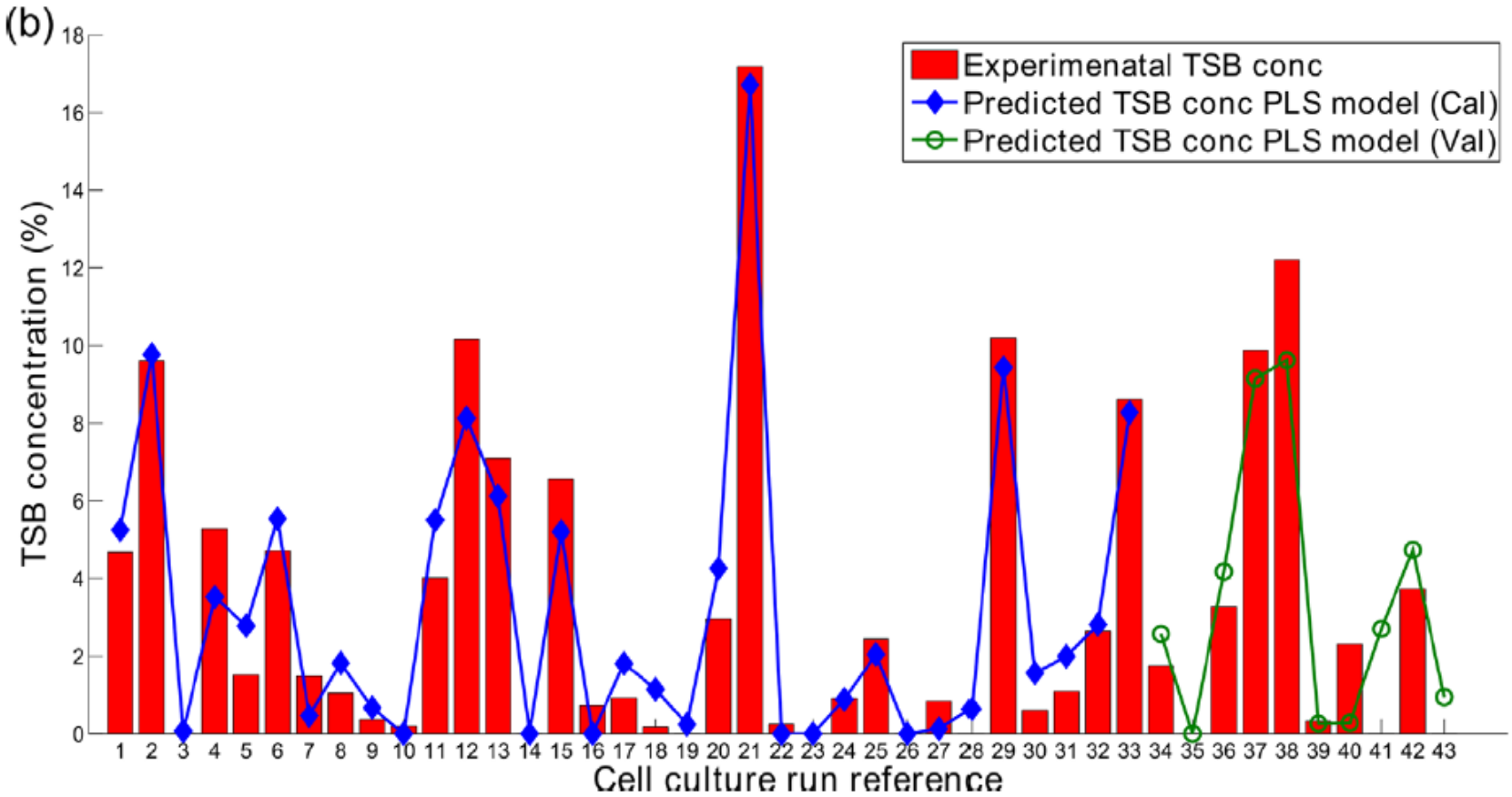
Summary of main variables contributing to 1st principle component



Why is MVDA important?

All on-line, off-line variables and initial conditions summarised into two graphs, allowing for easy interpretation of this complex data set

PLS enabled accurate predictions of end-point point deviations



Quick overview of MVDA techniques covered

Raw Data (X)

Scale data

Covariance Matrix

Columns

$$X = \begin{bmatrix} 1.67 & 59.0 & 25 \\ 1.81 & 80.6 & 14 \\ 1.66 & 70.1 & 22 \\ 1.30 & 90.1 & 2 \end{bmatrix}$$ Rows

X= [4x3]

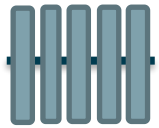
Rows -> Samples/Batches
Columns -> Variables

[X]= (Rows x Columns)

Raw data X

Subtract mean $X - \bar{X}$

Divide by Std $\frac{X - \bar{X}}{std(X)}$



Scaling is necessary to ensure variables with different units have equal weight during data analysis

Principal Component Analysis (PCA)

Scores plot

Highlights correlations between samples/batches

Loadings plot

Shows relationship between variables

PCA: Analysing the scores and loadings plot together, enables correlations to be developed within the data

Cov(X,Y)= $\begin{pmatrix} 1 & 0.97 \\ 0.97 & 1 \end{pmatrix}$

X and Y are highly positive correlated