SDSC Summer Institute 2020

ML3 - Data Preparation
Paul Rodriguez
08/05/20
Location: Breakout Room
Gitter (session support): Breakout Room
Gitter (general system support): Help Desk

Overview

- Highlights of data prep
- Variable Selection and Reduction

The Importance of Data Prep

- **“Garbage in, garbage out”**
- **Sometimes takes 60-80% of the whole data mining effort**
- **Preparing data is based on statistical principles**

The Importance of Data Prep

- **“Garbage in, garbage out”**
- **Sometimes takes 60-80% of the whole data mining effort**
- **Preparing data is based on statistical principles**

But also heuristics

Working definition

- **Data Preparation:**
  - Cleaning – outliers, missing data

Working definition

- **Data Preparation:**
  - Cleaning – outliers, missing data
  - Filtering – select rows or columns

Working definition

- **Data Preparation:**
  - Cleaning – outliers, missing data
  - Filtering – select rows or columns
  - Transforming – normalize or combine

Working definition

- **Data Preparation:**
  - Cleaning – outliers, missing data
  - Filtering – select rows or columns
  - Transforming – normalize or combine
  - Organizing data - aka ‘data wrangling’ or ‘data munging’

Working definition

- **Data Preparation:**
  - Cleaning – outliers, missing data
  - Filtering – select rows or columns
  - Transforming – normalize or combine
  - Organizing data - aka ‘data wrangling’ or ‘data munging’
  - Variable Selection/Dimension Reduction

Working definition

- **Data Preparation:**
  - Cleaning – outliers, missing data
  - Filtering – select rows or columns
  - Transforming – normalize or combine
  - Organizing data - aka ‘data wrangling’ or ‘data munging’
  - Variable Selection/Dimension Reduction

In a nutshell, prepare data for modeling

Missing Data – explore them

- **Get frequency counts and indices of missing variables**

Are the missing entries missing-at-random?

Quick Approaches

- **Delete instances**

In R: `X_data = na.omit(X_data)`

Quick Approaches

- Delete instances

In R: `X_data = na.omit(X_data)`

and/or

- Delete attributes with high missing-ness

In R use the is.na() function, returns 0 or 1

Quick Approaches

- Delete instances

  and/or

- Delete attributes with high missing-ness

In R: `X_data = na.omit(X_data)`

In R use the is.na() function, returns 0 or 1

```
foo      = function(x){sum(is.na(x))}      #a count of 'na' in x
sapply(X_data,foo)                         #apply foo to each column
X_data = subset(X_data,select=-c(your_col_name))  #delete a column
```

Imputation

- **Simple: Replace missing values with the mean**

Imputation

- **Simple: Replace missing values with the mean**

- **Complicated but most accurate:**

  Use a model (based on other attributes) to infer missing values

R and imputation

- **Several packages, such as ‘mice’ , ‘amelia’**

- **Iteratively estimate missing data in one column using data in other columns**

  Mice uses Gibbs sampling (slower)

  Amelia uses Expectation Maximization (faster)

R and imputation

- **‘Amelia’ package example**

  300K+ rows and 50 attributes from UN voting data
  1K-100K entries missing per col for about 20 cols

  Not run on user’s laptop; took about 1 hour on a Comet compute node

R and imputation

- **'Amelia' package example**

  300K+ rows and 50 attributes from UN voting data
  1K-100K entries missing per col for about 20 cols

  Not run on user's laptop; took about 1 hour on a Comet compute node

Identify the 'id' variables

```
library('amelia')
 a.out <- amelia(data,
                 idvars = …c("country-id"),
                 m=10, parallel = "multicore")
```

Run 10 models in parallel

Variable Transformations

- Normalize or Scale data (if needed)
- Engineer new features (if it helps)
- Combine attributes (e.g. rates and ratios)
- Discretize data into bins (maybe more intuitive)

Variable Transformations

- Normalize or Scale data (if needed)
- Engineer new features (if it helps)
- Combine attributes (e.g. rates and ratios)
- Discretize data into bins (maybe more intuitive)

If variables are on different scales

Use prior knowledge

Normalizing or scaling

- **Mean center** $x_{new} = x - \text{mean}(x)$
- **z-score** $score = \frac{x - \text{mean}(x)}{\text{std}(x)}$
- **Scale to [0…1]** $x_{new} = \frac{x - \min(x)}{\max(x) - \min(x)}$
- **log scaling** $x_{new} = \log(x)$

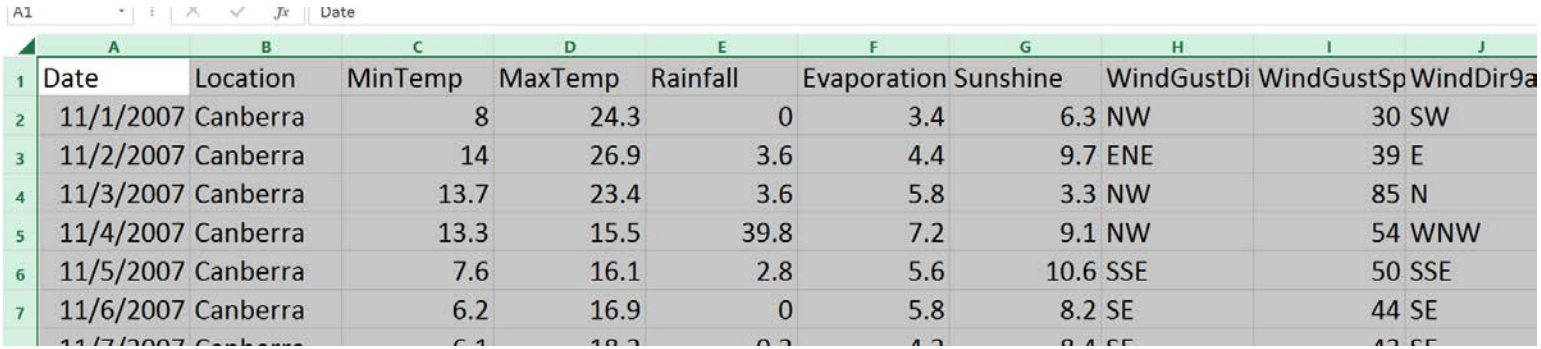
Data Wrangling

- **Organizing data for modeling**

lots of R packages and functions for date strings, matching, selecting, grouping, gathering, reading, etc…

We'll look at a couple of examples

Data Wrangling - Long to Wide transformation

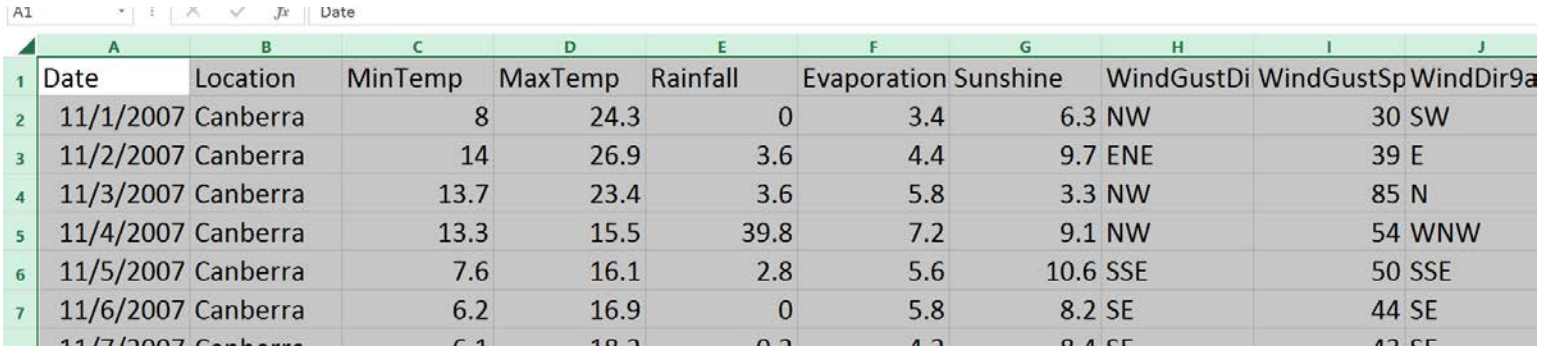


	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9a
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N
5	11/4/2007	Canberra	13.3	15.5	39.8	7.2	9.1	NW	54	WNW
6	11/5/2007	Canberra	7.6	16.1	2.8	5.6	10.6	SSE	50	SSE
7	11/6/2007	Canberra	6.2	16.9	0	5.8	8.2	SE	44	SE

date, location and the rest identify the row

WindGustDir and WindGustSpeed are repeatedly measured and measurements are on different rows

Data Wrangling - Long to Wide transformation



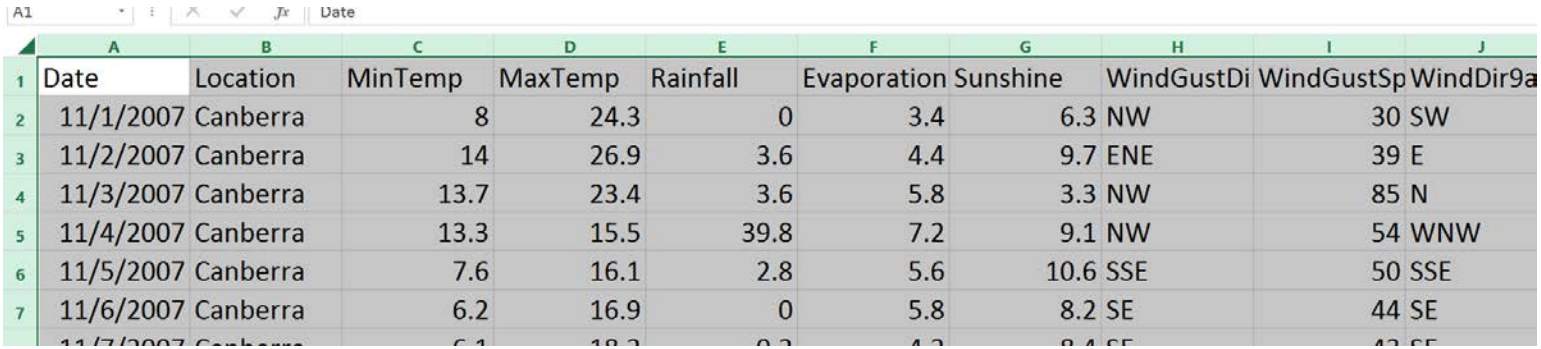
	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9a
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N
5	11/4/2007	Canberra	13.3	15.5	39.8	7.2	9.1	NW	54	WNW
6	11/5/2007	Canberra	7.6	16.1	2.8	5.6	10.6	SSE	50	SSE
7	11/6/2007	Canberra	6.2	16.9	0	5.8	8.2	SE	44	SE

date, location and the rest identify the row

WindGustDir and WindGustSpeed are repeatedly measured and measurements are on different rows

How to get all repeated measurements into 1 row?

Data Wrangling - Long to Wide transformation



	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9a
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N
5	11/4/2007	Canberra	13.3	15.5	39.8	7.2	9.1	NW	54	WNW
6	11/5/2007	Canberra	7.6	16.1	2.8	5.6	10.6	SSE	50	SSE
7	11/6/2007	Canberra	6.2	16.9	0	5.8	8.2	SE	44	SE

date, location and the rest identify the row

WindGustDir and WindGustSpeed are repeatedly measured and measurements are on different rows

How to get all repeated measurements into 1 row?

Let's try "reshape2" library

Data Wrangling - Long to Wide transformation

	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9am
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N

date, location and the rest identify the row

```
library(reshape2)
W_long   =dcast(W_df,
          formula=Date+Location+ ...~
```

Data Wrangling - Long to Wide transformation

	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9am
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N

date, location and the rest identify the row

Put variable that has labels for the repeated measures

```
library(reshape2)
W_wide   =dcast(W_df,
          formula=Date+Location+ ...~ <<<variable-name>>>,
```

Data Wrangling - Long to Wide transformation

	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9am
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N

date, location and the rest identify the row

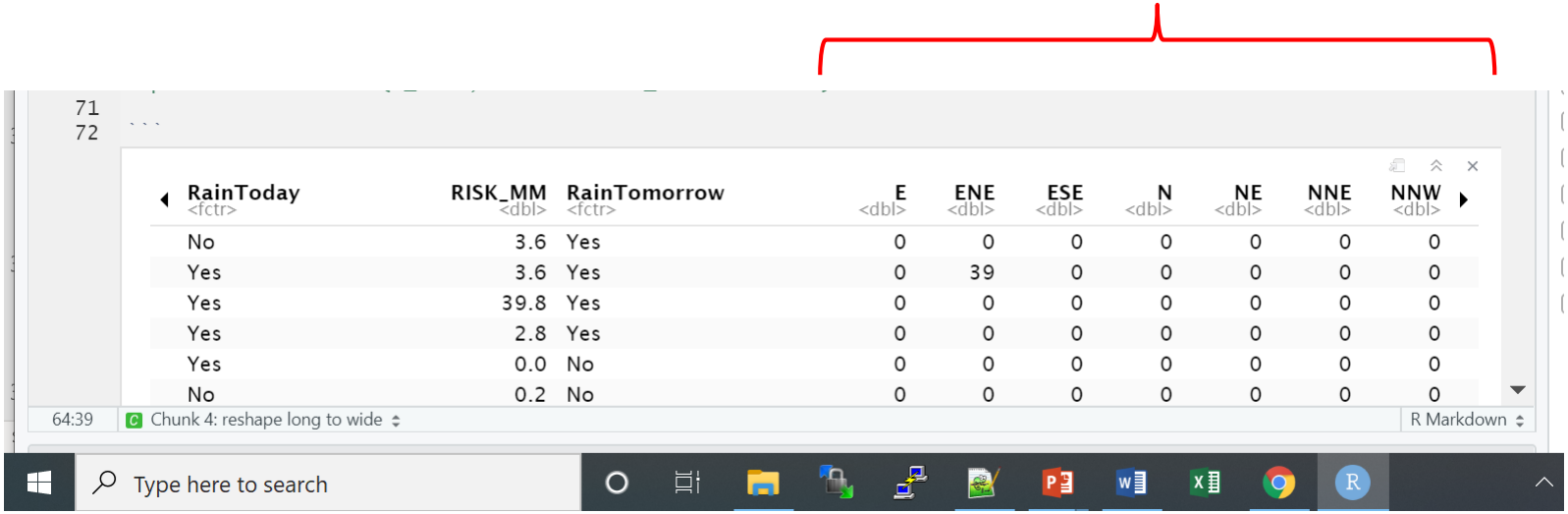
Put variable that has labels for the repeated measures

```
library(reshape2)
W_wide   =dcast(W_df,
          formula=Date+Location+ ...~ <<<variable-name>>>,
          fill=0,
          value.var="<<<variable-name>>>")
```

Indicate variable that has the repeated measurement values

Transformed Data Matrix

After running dcast: WindGustDir category labels are new columns



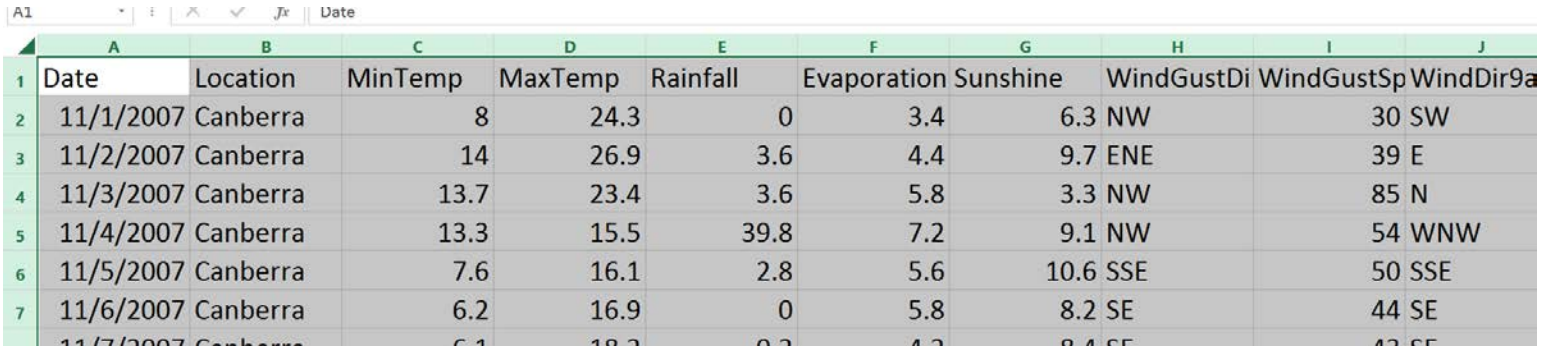
71
72 ```

RainToday <fctr>	RISK_MM <dbl>	RainTomorrow <fctr>	E <dbl>	ENE <dbl>	ESE <dbl>	N <dbl>	NE <dbl>	NNE <dbl>	NNW <dbl>
No	3.6	Yes	0	0	0	0	0	0	0
Yes	3.6	Yes	0	39	0	0	0	0	0
Yes	39.8	Yes	0	0	0	0	0	0	0
Yes	2.8	Yes	0	0	0	0	0	0	0
Yes	0.0	No	0	0	0	0	0	0	0
No	0.2	No	0	0	0	0	0	0	0

64:39 Chunk 4: reshape long to wide R Markdown

Type here to search

Data Wrangling – grouping



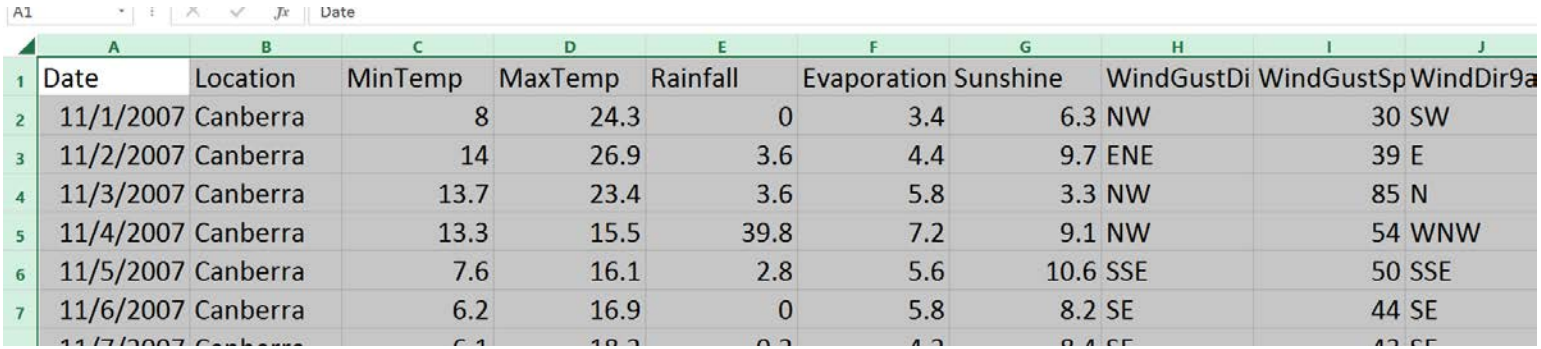
	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9a
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N
5	11/4/2007	Canberra	13.3	15.5	39.8	7.2	9.1	NW	54	WNW
6	11/5/2007	Canberra	7.6	16.1	2.8	5.6	10.6	SSE	50	SSE
7	11/6/2007	Canberra	6.2	16.9	0	5.8	8.2	SE	44	SE

date, location and the rest identify the row

WindGustDir and WindGustSpeed are repeatedly measured

How to get mean speed for each direction?

Data Wrangling – grouping



	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9a
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N
5	11/4/2007	Canberra	13.3	15.5	39.8	7.2	9.1	NW	54	WNW
6	11/5/2007	Canberra	7.6	16.1	2.8	5.6	10.6	SSE	50	SSE
7	11/6/2007	Canberra	6.2	16.9	0	5.8	8.2	SE	44	SE

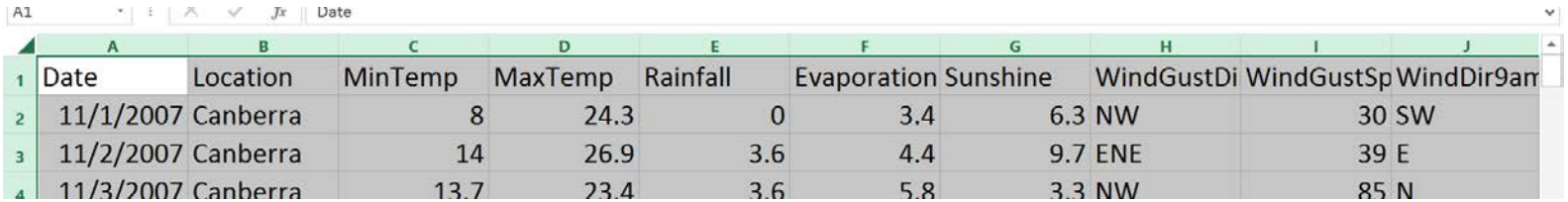
date, location and the rest identify the row

WindGustDir and WindGustSpeed are repeatedly measured

How to get mean speed for each direction?

Let's try "dplyr" library

Data Wrangling - grouping



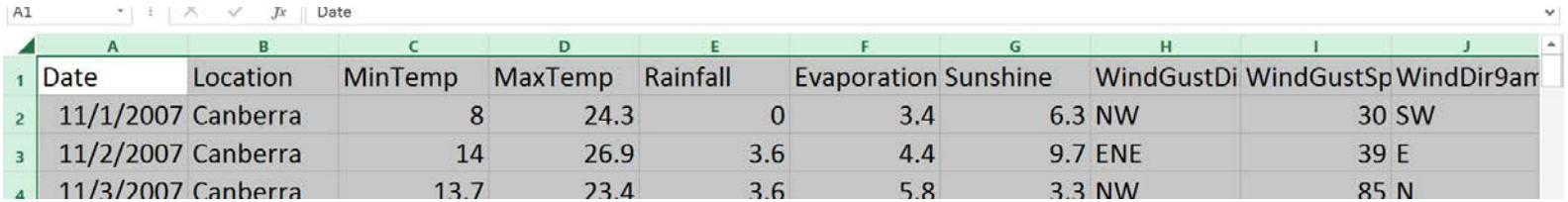
	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9am
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N

Identify groups of the values of WindGustDir

```
library(dplyr)

a1 <- group_by(na.omit(W_df), WindGustDir)
```

Data Wrangling - grouping



	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9am
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N

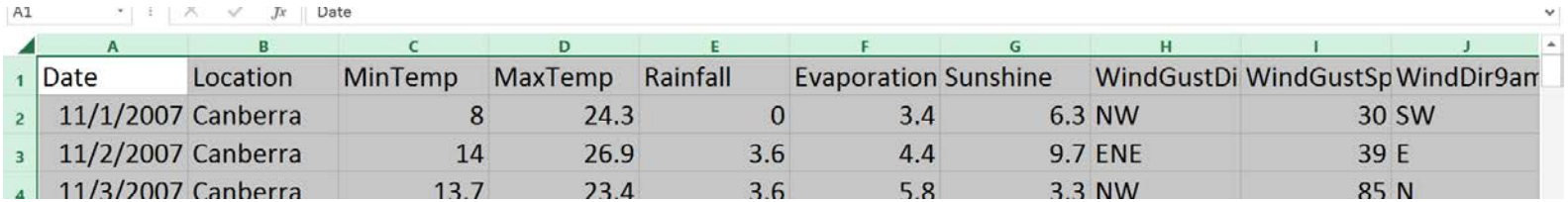
Identify groups of the values of WindGustDir

Select columns to aggregate

```
library(dplyr)

a1 <- group_by(na.omit(W_df), WindGustDir)
a2 <- select(a1,WindSpeed9am,Temp9am)
```

Data Wrangling - grouping



	A	B	C	D	E	F	G	H	I	J
1	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDi	WindGustSp	WindDir9am
2	11/1/2007	Canberra	8	24.3	0	3.4	6.3	NW	30	SW
3	11/2/2007	Canberra	14	26.9	3.6	4.4	9.7	ENE	39	E
4	11/3/2007	Canberra	13.7	23.4	3.6	5.8	3.3	NW	85	N

Identify groups of the values of WindGustDir

Select columns to aggregate

```
library(dplyr)

a1 <- group_by(na.omit(W_df), WindGustDir)
a2 <- select(a1,WindSpeed9am,Temp9am)
a3 <- summarise(a2,
  avg_speed = mean(WindSpeed9am, na.rm = TRUE),
  avg_temp  = mean(Temp9am, na.rm = TRUE)  )
```

Summarise

R exercise

Use “reshape2” library to

1. “cast” repeated measurements into one row (long to wide): <<< fill-in variable names >>>

[Extra: “melt” row back into repeated measurements]

2. Use “dplyr” library to perform grouping and aggregations: <<< fill-in variable names >>>

and compare that to ‘reshape’ with ‘sum’

pause

Reading Material

- **Data Preparation for Data Mining by Dorian Pyle**
  - http://www.ebook3000.com/Data-Preparation-for-Data-Mining_88909.html
- **Data mining – Practical Machine learning tools and techniques by Witten & Frank**
  - http://books.google.com

Many Variables

- **More variables => more information, but also more noise and more ways of interactions**

- **2 ways to handle many variables**
  - Variable Selection
  - Dimension reduction methods

Variable selection

- **Heuristically, pick off or put in 1 variable at a time (step wise)**

  based on some criteria, like correlation with outcomes

Matrix Factorization:

Given a numeric matrix, can we reduce the number of columns?

Matrix Factorization:

Given a numeric matrix, can we reduce the number of columns?

- Yes, if features are constant or redundant

Matrix Factorization:

Given a numeric matrix, can we reduce the number of columns?

- Yes, if features are constant or redundant
- Yes, if features only contribute noise

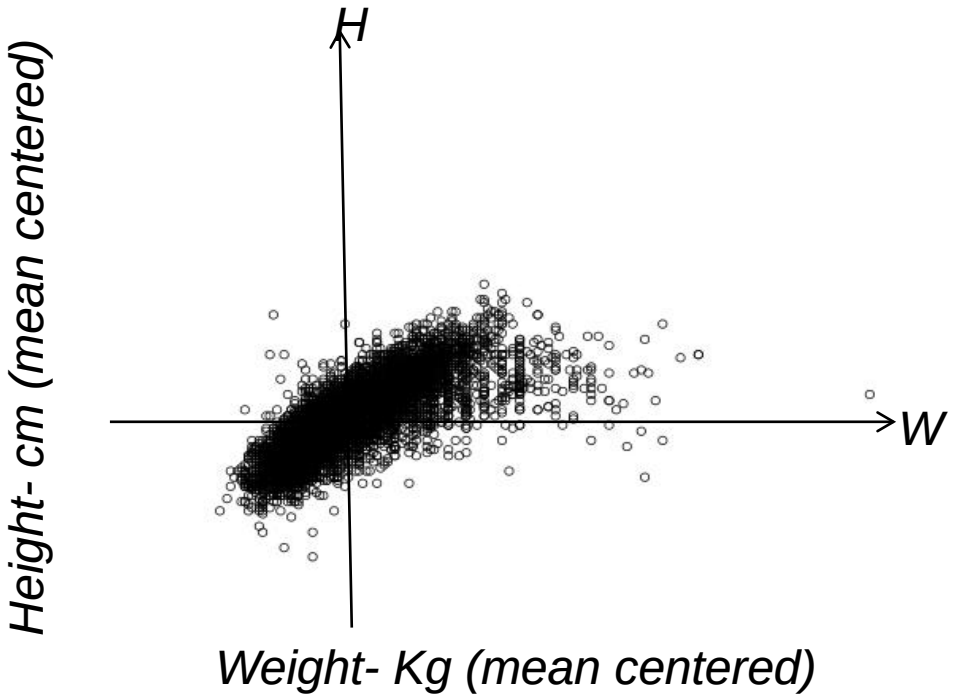
Matrix Factorization:

Given a numeric matrix, can we reduce the number of columns?

- Yes, if features are constant or redundant
- Yes, if features only contribute noise

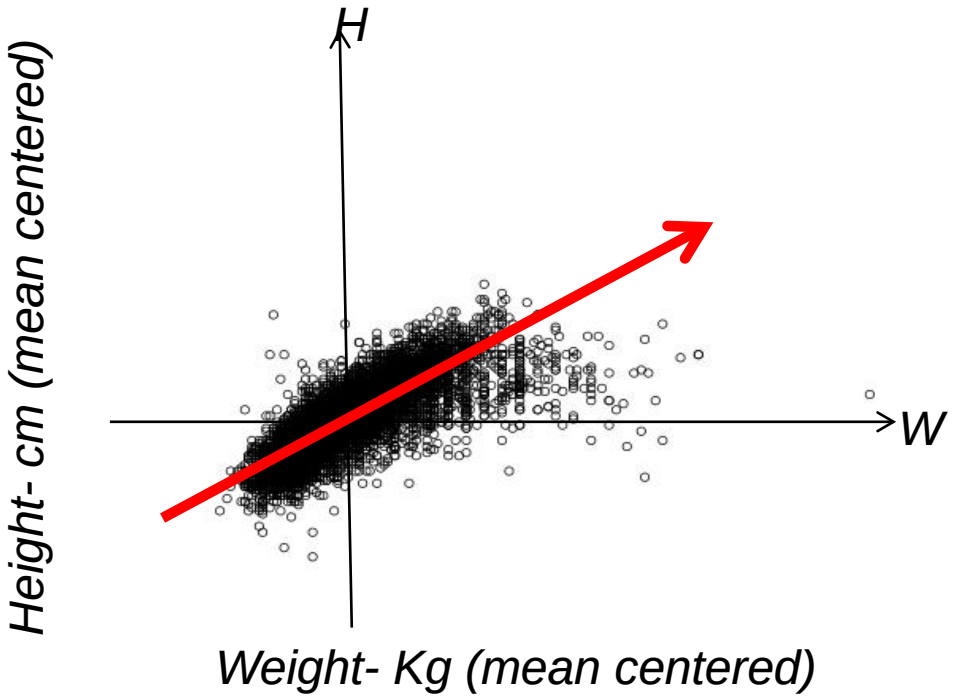
Conversely, want features that contribute to variations of the data

Example: Athletes' Height by Weight



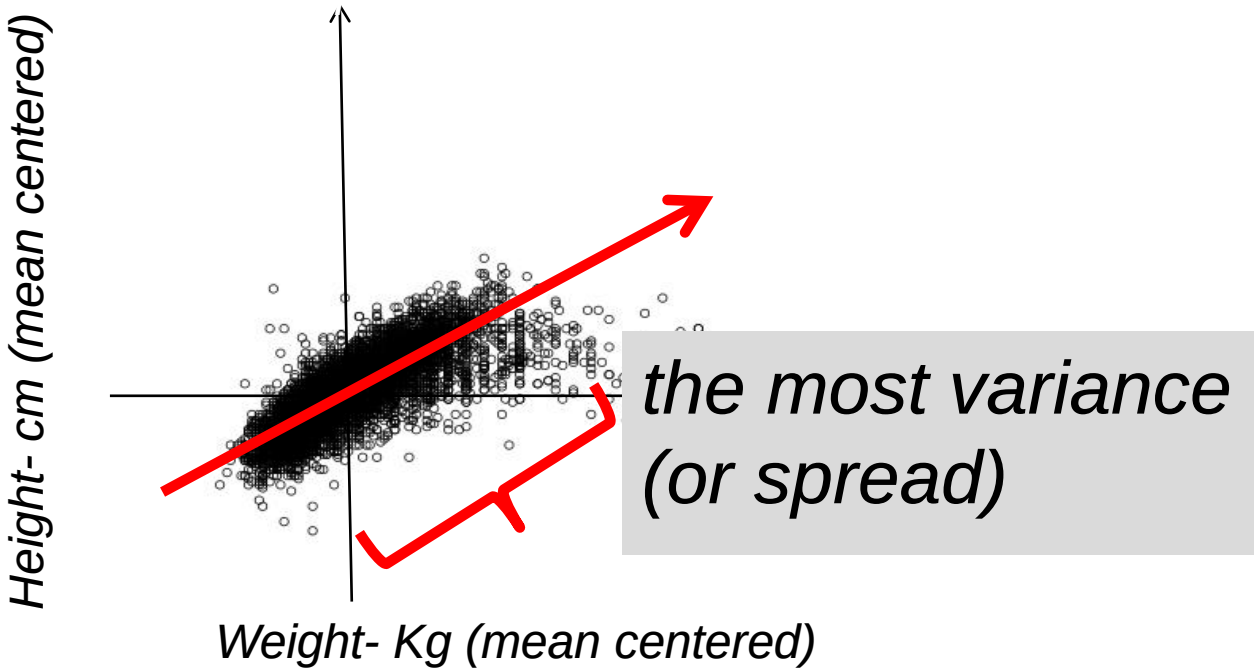
Find a line that aligns with the data.

Example: Athletes' Height by Weight



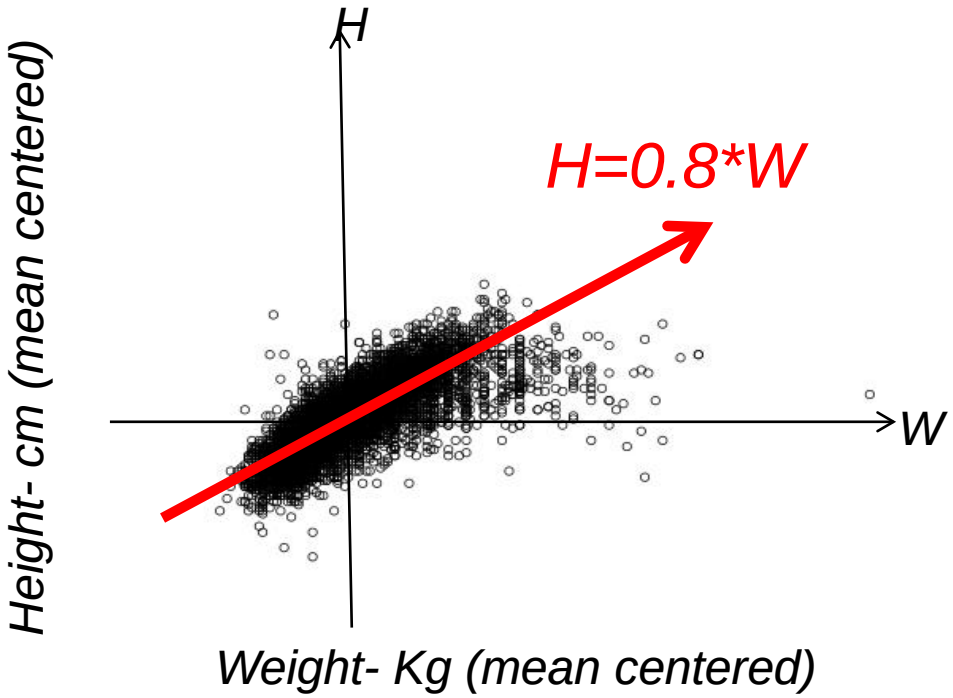
Find a line that aligns with the data.

Example: Athletes' Height by Weight



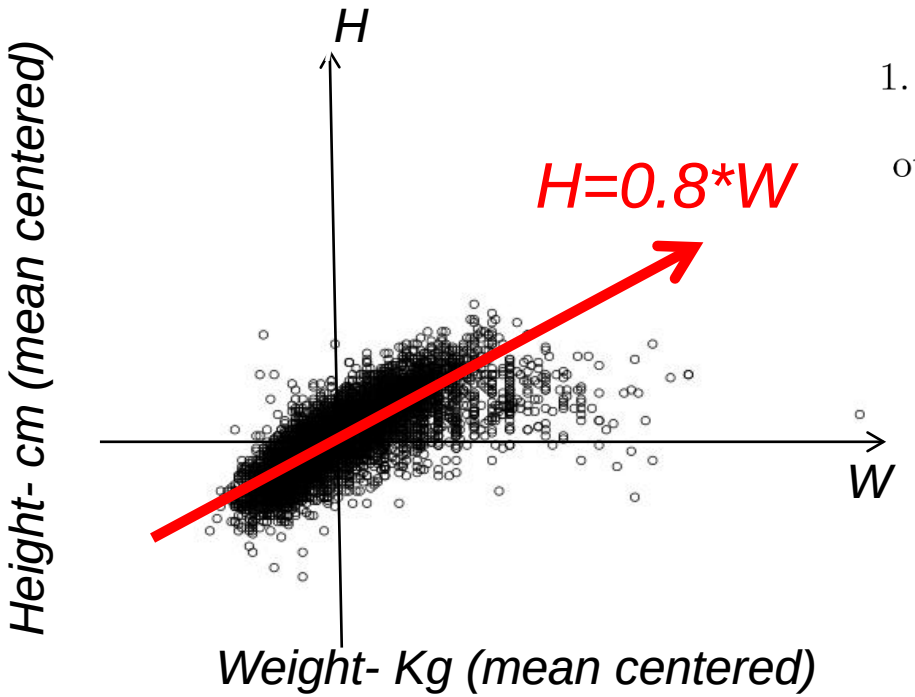
Find a line that aligns with the data.

Example: Athletes' Height by Weight



Find a line that aligns with the data.

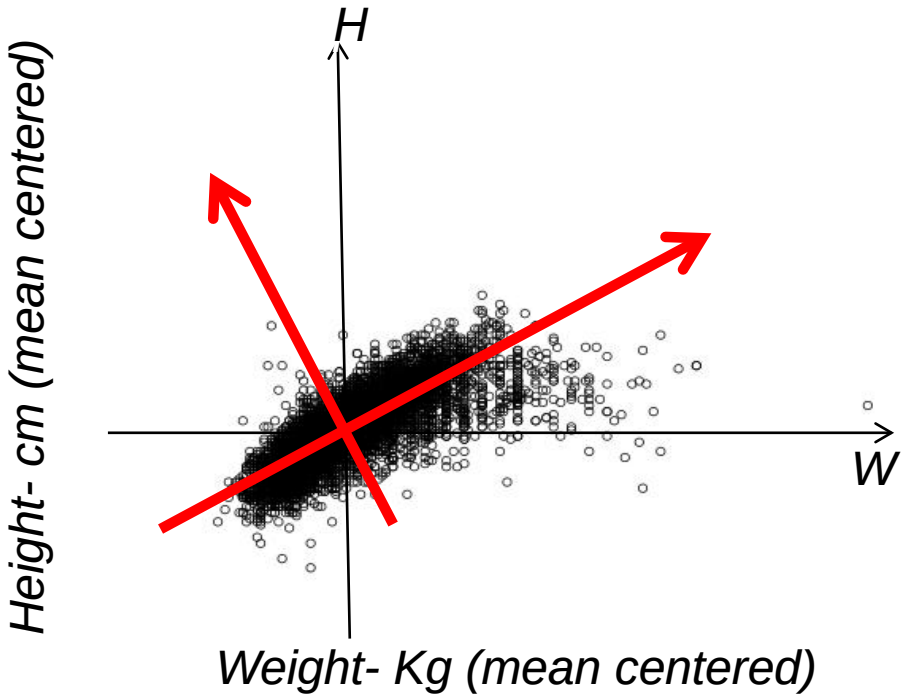
Note that ***(0,0) and (1,0.8) are points on the line that are combinations of H,W***



1. The vector in HxW space: $v = \begin{pmatrix} W = 1 \\ H = 0.8 \end{pmatrix}$

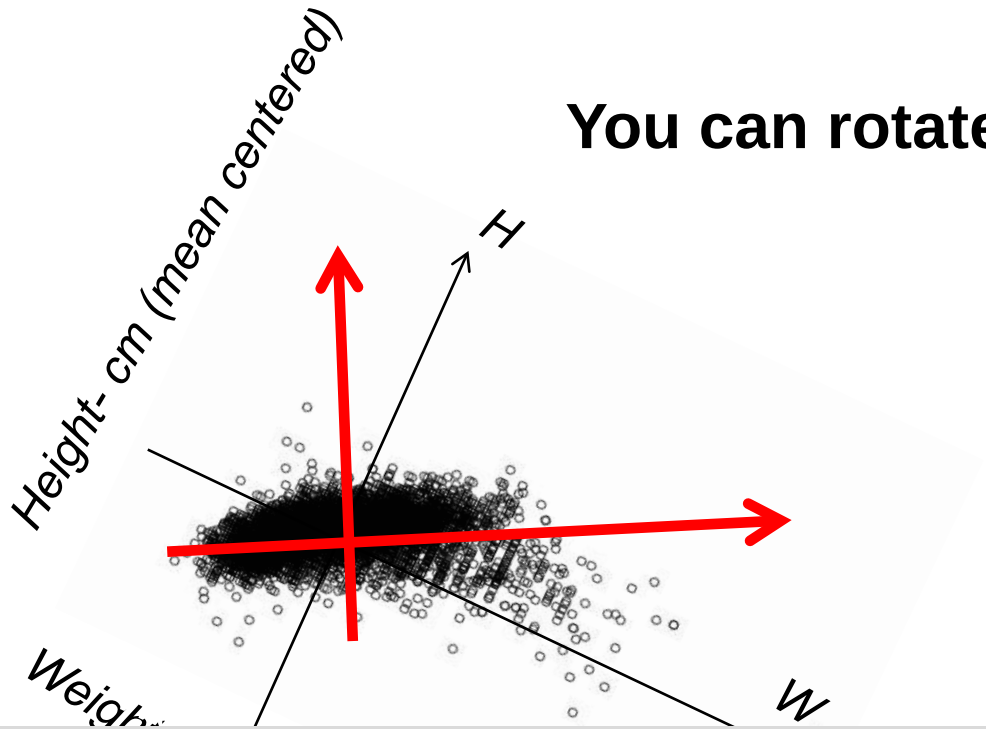
other points also satisfy H=0.8*W : $\alpha * v$

Find a line that aligns with the data.



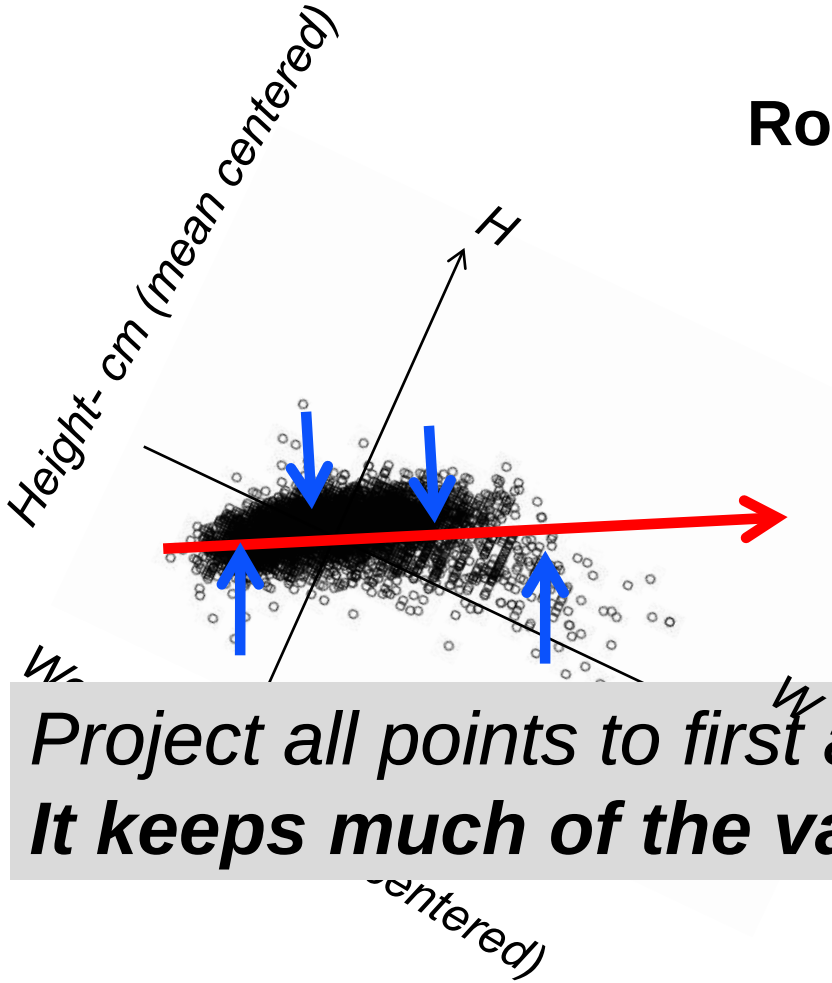
The next direction of most variance.

You can rotate the axes



New axes (i.e., new features or latent factors) are combinations of old axis

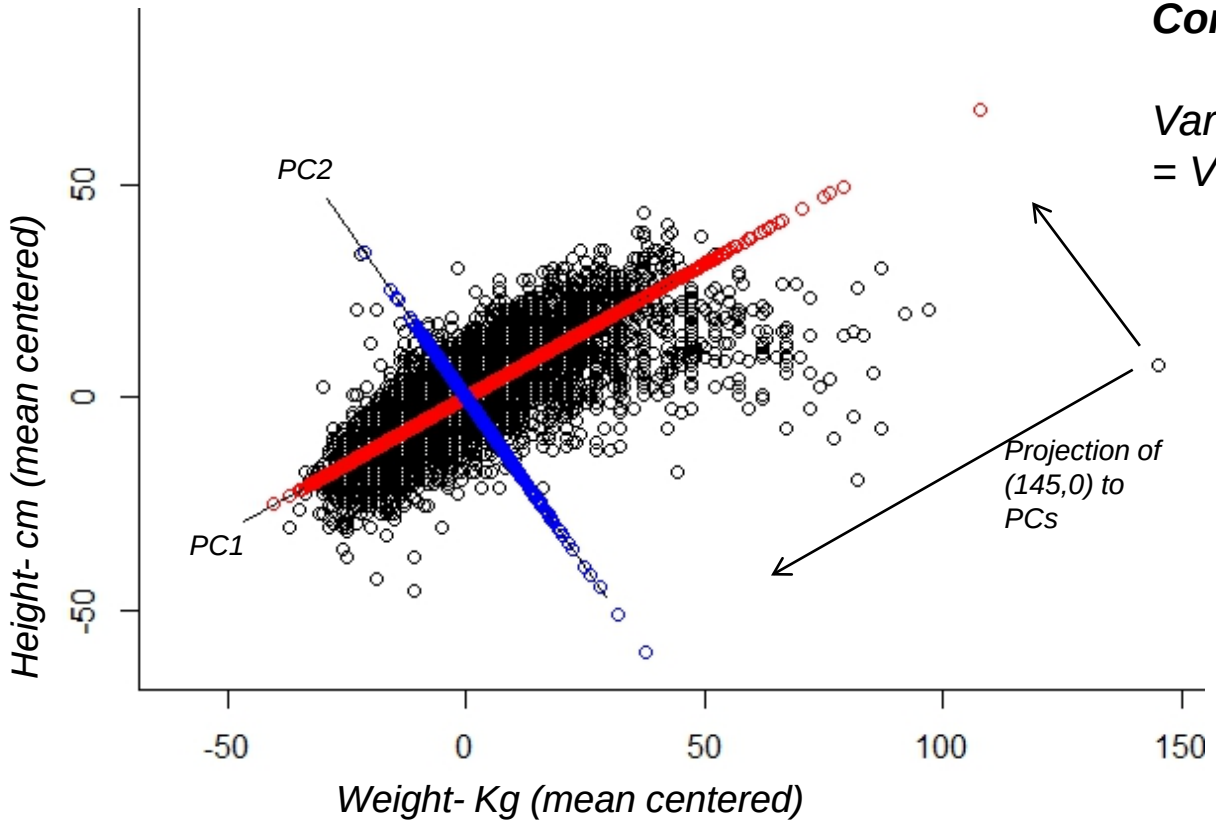
Rotate axis



Project all points to first axis

It keeps much of the variance

Note: this factorization conserves total variance



Total Variance Conserved:

Var in Weight + Var in Height
= Var in PC1 + Var in PC2

Best Known Algorithms

SVD (singular value decomposition)

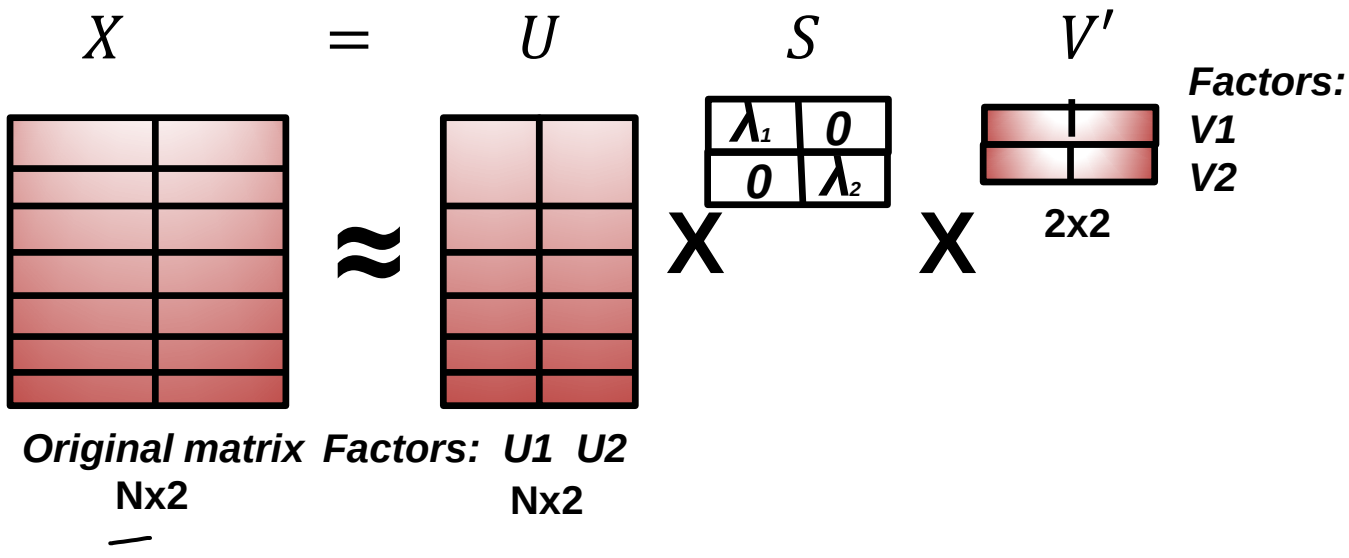
PCA (principle component analysis)

SVD more generally works on non square matrices

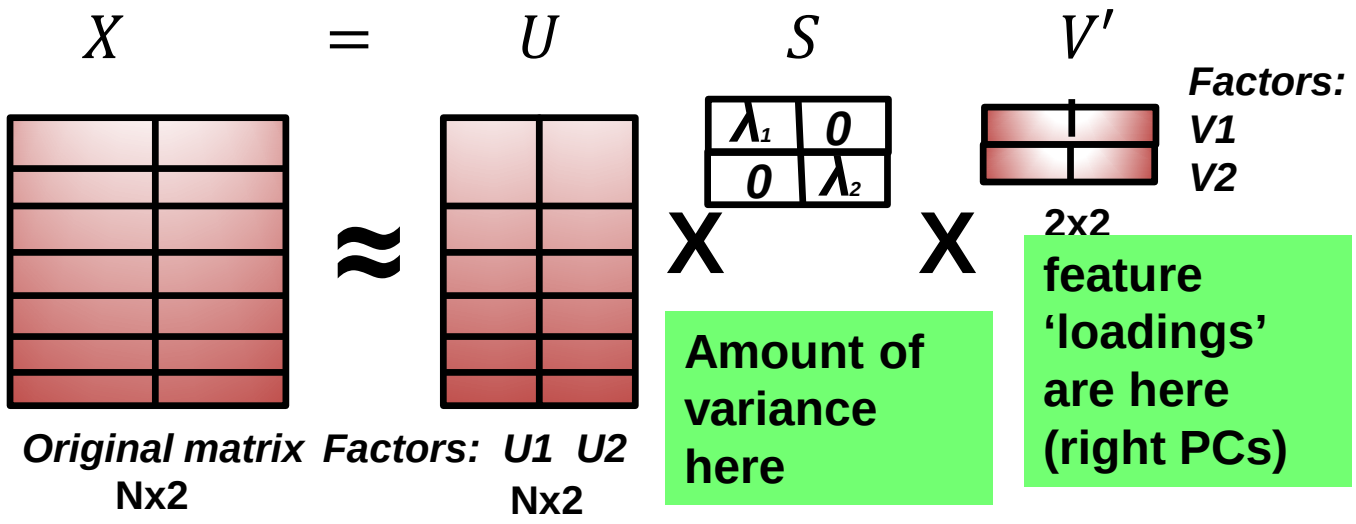
SVD decomposes X matrix into factors (ie column vectors) and 'singular' values

$$X = U S V'$$

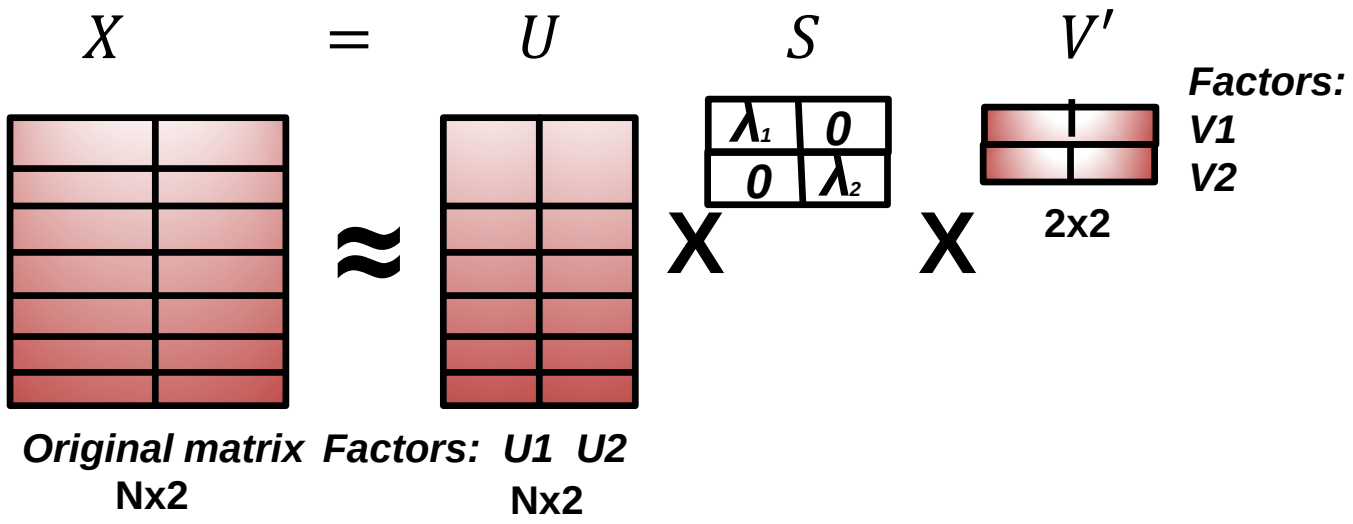
SVD decomposes X matrix into factors (ie column vectors) and 'singular' values



SVD decomposes X matrix into factors (ie column vectors) and 'singular' values



SVD decomposes X matrix into factors (ie column vectors) and 'singular' values



For U,S,V with less than P dimensions it is an approximation $X \sim U\ S\ V'$

The V here is same as in Principle Components (up to a sign change) of cov(X) matrix

Using PCs

- **SVC or PCA:**

  only use numeric columns, center and normalize

Use for dimension reduction, visualization, examine factor scores/loadings

Combine with clustering, regression, classification, etc…

Run the SVD exercise for practice and later it with K-means clustering

Note, the SVD command in R looks like this:

```
> Xsvd=svd(X)

> str(Xsvd)

List of 3
 $ d: num [1:9] 27442.7 231.2 96.4 68.2 44.5 ...
 $ u: num [1:363, 1:9] -0.0524 -0.0521 -0.052 -0.0519 -0.0525 ...
 $ v: num [1:9, 1:9] -0.005042 -0.014276 -0.000969 -0.00314 -0.005491 ...
```

- **end**