Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

# Business Analytics using Data Mining

BU7143

Dr. Nicholas P. Danks
Business Analytics
nicholas.danks@tcd.ie

# Tools we *will* use

## Coding language

Install R:
http://www.r-project.org/

## Integrated Development Environment

Install RStudio:
http://www.rstudio.com/

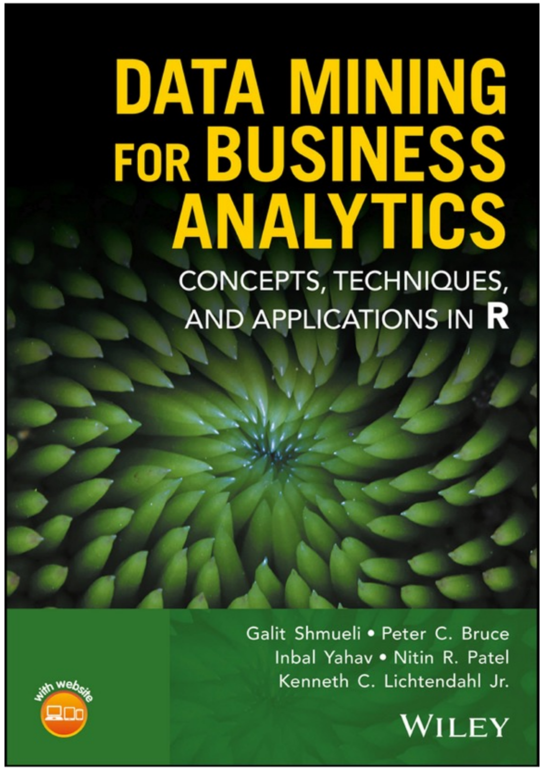
## Version control

Join GitHub:
https://github.com/

# Textbook

Data Mining for Business Analytics in R

Shmueli, Bruce, Yahav, Patel & Lichtendahl





(any version is OK – but for **R** is better)

# Overview of Today's Session

1. Translating Business Problems to Statistical Problems
2. Core tasks/goals of Data Mining
3. The process of Data Mining
4. Sampling
5. Variable types
6. Outliers, missing data, normal data
7. Dimension Reduction
8. Performance

# Business Problem -> Statistical Problem

1. Understand & Define the problem
- *Frame the business problem*
- *Prepare for a decision*
2. Set analytic goals and scope your solution
- *Set objectives and define milestones*
- *Design minimum viable product*
- *Identify target metrics*
3. Plan the analysis
- *Plan your datasets*
- *Plan your methods*

https://ayotomiwasalau.medium.com/translate-business-problems-to-data-science-problems-b1693d8f313a

# Pandemic Example

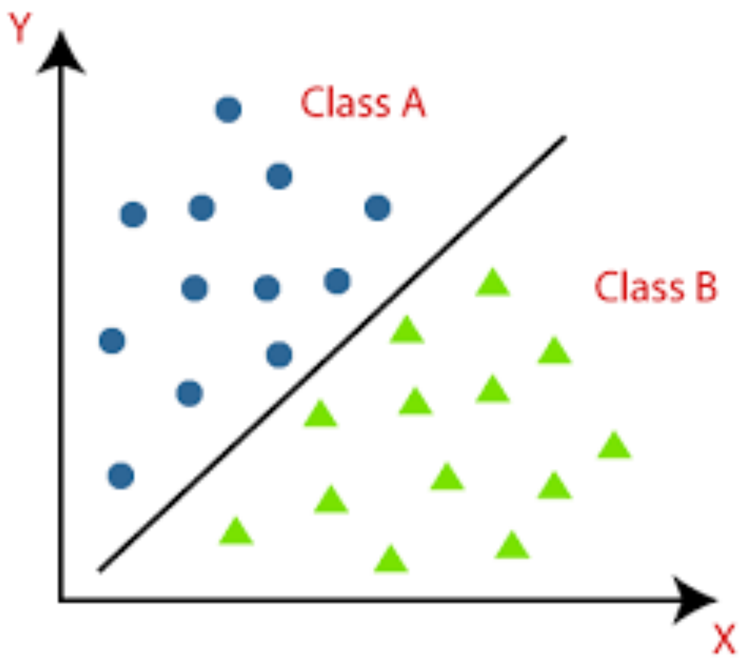
What **data** do we have?
How can it be **converted**?
What can be **predicted**?
What is the **business value**?


https://youtu.be/TGahNuPH9LY

| Vendor serial number | User phone number | Timestamp |
|---|---|---|
| 111-111-111-111 | 0851991999 | 10:45:22-21:05:2021 |

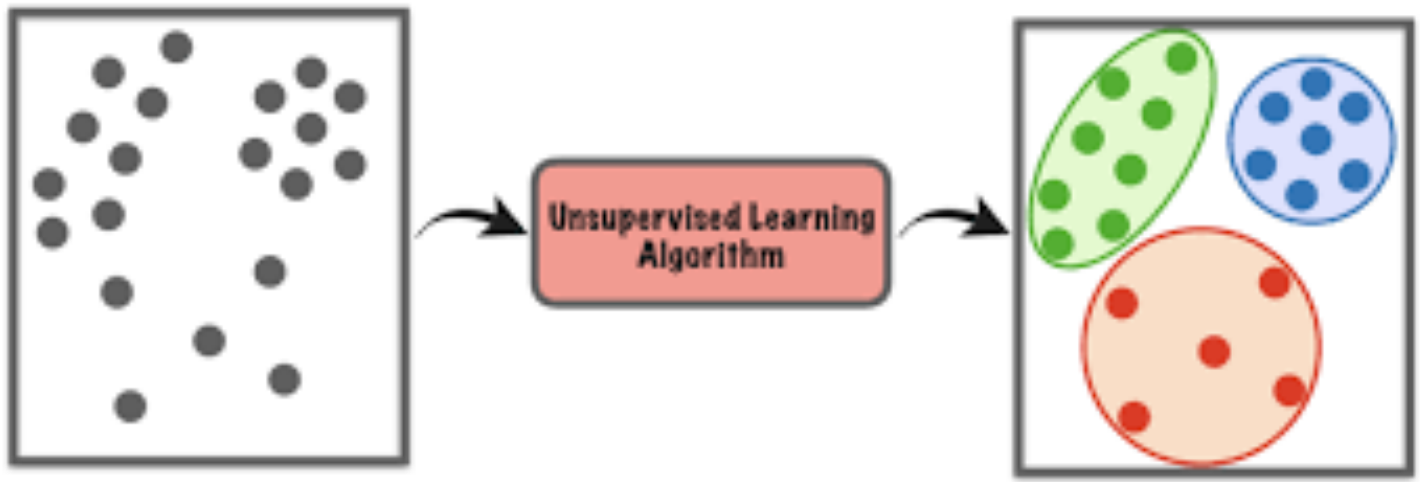
# Supervised: Classification

- Goal: Predict categorical **target** (outcome) variable
- Examples: Purchase/no purchase, fraud/no fraud, creditworthy/not creditworthy…
- Each row is a case (customer, tax return, applicant)
- Each column is a variable
- **Target variable** is often binary (yes/no)



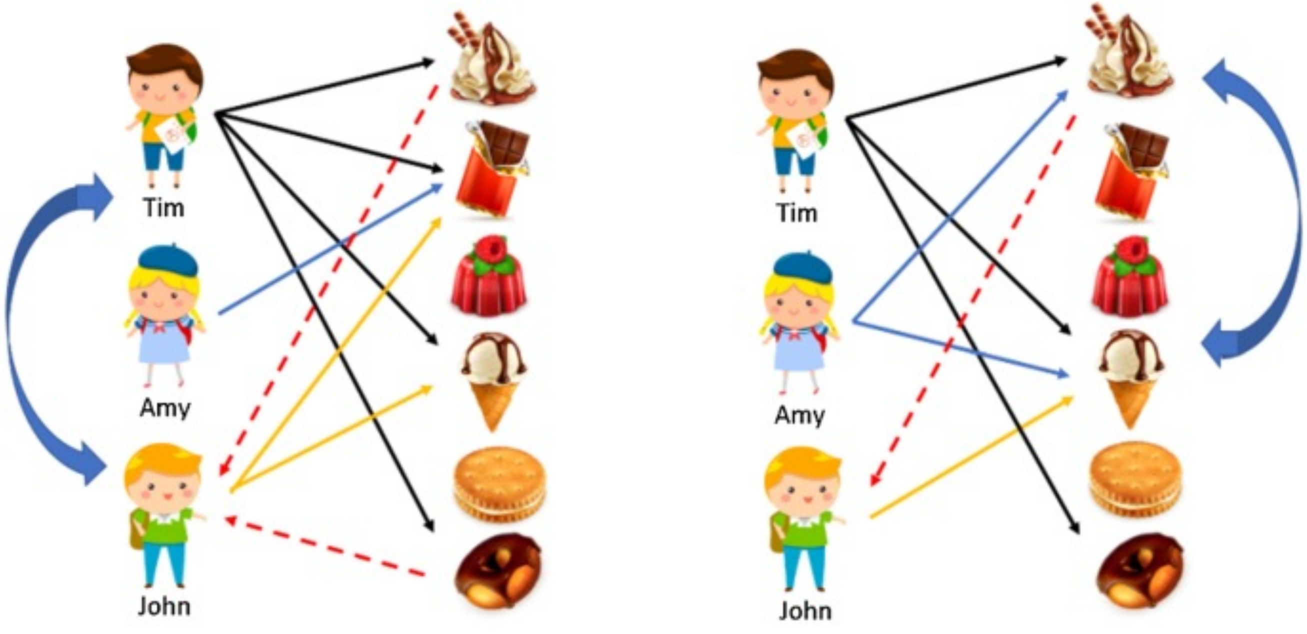
# Unsupervised Learning

- **Task:** Segment data into meaningful segments; detect patterns
- **Data**: There is no target (outcome) variable to predict or classify
- **Goal:** Identify which group an obs belongs to
- **Methods:** Association rules, collaborative filters, data reduction & exploration, visualization



# Unsupervised: Collaborative Filtering

- **Task:** Recommend products to purchase
- **Data:** Based on products that customer rates, selects, views, or purchases
- **Goal:** Recommend products that "customers like you" purchase (user-based); or
- **Goal:** recommend products that share a "product purchaser profile" with your purchases (item-based)

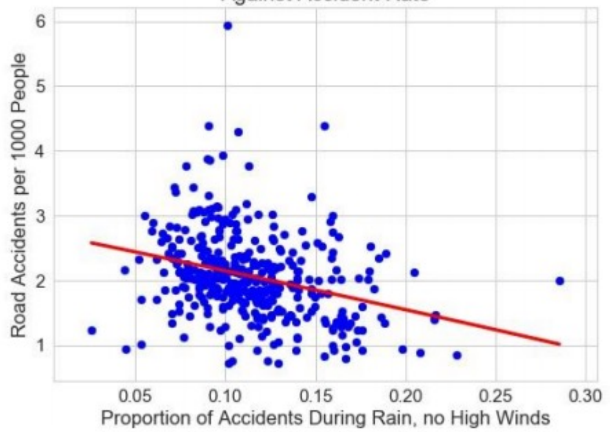



(a) User-based filtering (b) Item-based filtering

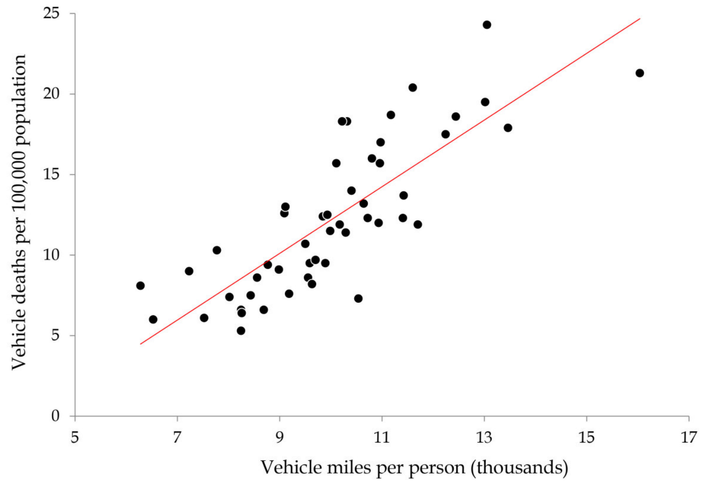
# Cross-sectional (Stationary)

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | RushHour | WRK_ZONE | WKDY | INT_HWY | LGTCON_day | LEVEL | SPD_LIM | SUR_COND_ | TRAF_two_w | WEATHER_a | MAX_SEV |
| 2 | 1 | 0 | 1 | 1 | 0 | 1 | 70 | 0 | 0 | 1 | no-injury |
| 3 | 1 | 0 | 1 | 0 | 0 | 0 | 55 | 0 | 1 | 0 | non-fatal |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 35 | 0 | 0 | 1 | no-injury |
| 5 | 1 | 0 | 1 | 0 | 0 | 1 | 35 | 0 | 0 | 1 | no-injury |
| 6 | 1 | 0 | 1 | 0 | 0 | 0 | 25 | 0 | 0 | 1 | non-fatal |
| 7 | 1 | 0 | 1 | 0 | 0 | 0 | 35 | 0 | 0 | 1 | non-fatal |
| 8 | 1 | 0 | 1 | 0 | 0 | 0 | 60 | 0 | 0 | 0 | no-injury |
| 9 | 1 | 0 | 1 | 0 | 0 | 1 | 45 | 1 | 1 | 0 | non-fatal |
| 10 | 0 | 0 | 1 | 1 | 0 | 0 | 55 | 1 | 0 | 0 | no-injury |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 70 | 1 | 0 | 0 | non-fatal |
| 12 | 0 | 0 | 1 | 1 | 0 | 0 | 65 | 1 | 0 | 0 | no-injury |
| 13 | 1 | 0 | 1 | 0 | 0 | 0 | 40 | 1 | 0 | 0 | non-fatal |
| 14 | 1 | 0 | 1 | 0 | 0 | 0 | 45 | 1 | 0 | 0 | non-fatal |
| 15 | 1 | 0 | 0 | 0 | 0 | 0 | 45 | 1 | 1 | 0 | non-fatal |
| 16 | 1 | 0 | 1 | 0 | 0 | 0 | 45 | 1 | 1 | 0 | no-injury |
| 17 | 1 | 0 | 1 | 0 | 0 | 0 | 30 | 1 | 1 | 0 | non-fatal |
| 18 | 1 | 0 | 1 | 0 | 0 | 0 | 55 | 1 | 1 | 0 | non-fatal |
| 19 | 1 | 0 | 1 | 0 | 0 | 0 | 55 | 1 | 1 | 0 | no-injury |
| 20 | 1 | 0 | 1 | 0 | 0 | 0 | 25 | 1 | 1 | 0 | no-injury |
| 21 | 0 | 0 | 1 | 0 | 0 | 1 | 35 | 0 | 0 | 1 | no-injury |
| 22 | 0 | 0 | 1 | 0 | 0 | 1 | 35 | 0 | 1 | 1 | no-injury |
| 23 | 0 | 0 | 1 | 0 | 0 | 0 | 25 | 0 | 1 | 1 | no-injury |
| 24 | 0 | 0 | 1 | 0 | 0 | 1 | 45 | 0 | 0 | 1 | no-injury |
| 25 | 1 | 0 | 1 | 0 | 0 | 0 | 35 | 0 | 1 | 1 | no-injury |
| 26 | 0 | 0 | 1 | 0 | 0 | 1 | 55 | 0 | 0 | 1 | non-fatal |
| 27 | 1 | 0 | 1 | 0 | 0 | 1 | 40 | 0 | 0 | 1 | no-injury |
| 28 | 1 | 0 | 0 | 0 | 0 | 1 | 35 | 0 | 1 | 1 | non-fatal |
| 29 | 0 | 0 | 0 | 0 | 0 | 1 | 25 | 0 | 1 | 0 | non-fatal |
| 30 | 0 | 0 | 1 | 0 | 0 | 1 | 25 | 0 | 1 | 1 | no-injury |

Weather Condition: Raining, no high winds
Against Accident Rate




**Does driving cause traffic fatalities?**
*Miles driven and fatality rate: U.S. states, 2012*




single value of dependent variable

slope

single value of independent variable

y-intercept

$$y = mx + b$$

$$Y = \beta_0 + \beta_1 X + \epsilon$$

all observed values for dependent variable

y-intercept a.k.a "bias"

slope a.k.a. "coefficient"

all observed values of independent variable

error*

* additional term

$\alpha$

# Quantitative Measurement Scales

## Nominal Scale
- grouping / categorization

## Ordinal Scale
- greater-than / less-than comparisons

## Interval Scale
- greater-than / less-than comparisons
- meaningful units
- meaningful distance within scale

## Ratio Scale
- greater-than / less-than comparisons
- meaningful units
- meaningful distance within scale
- absolute zero
- meaningful multiples

Nominal

Interval

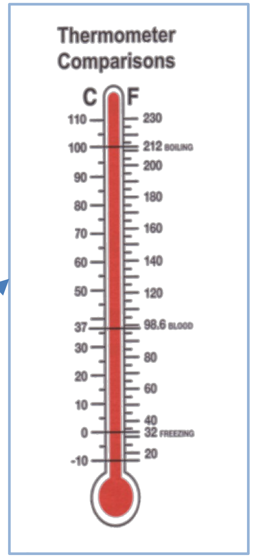
Ordinal

source: ESPN.com; thetipsbank.com

Men's Singles Rankings

| RK | PLAYER | COUNTRY | MOVEMENT | POINTS |
|---|---|---|---|---|
| 1 | Roger Federer | | 0 | 10105 |
| 2 | Rafael Nadal | | 0 | 9760 |
| 3 | Marin Cilic | | 0 | 4960 |
| 4 | Grigor Dimitrov | | 0 | 4635 |
| 5 | Alexander Zverev | | 0 | 4450 |
| 6 | Dominic Thiem | | 0 | 3810 |
| 7 | David Goffin | | 0 | 3280 |
| 8 | Kevin Anderson | | 1 | 2825 |
| 9 | Juan Martin del Potro | | 1 | 2745 |
| 10 | Jack Sock | | 2 | 2650 |

Ratio




Scribbr

The four levels of measurement

| | Nominal | Ordinal | Interval | Ratio |
|---|---|---|---|---|
| Categories | ✓ | ✓ | ✓ | ✓ |
| Rank order | | ✓ | ✓ | ✓ |
| Equal spacing | | | ✓ | ✓ |
| True zero | | | | ✓ |

# Creating Binary Dummies

```
> head(cbind(housing.df$REMODEL, xtotal))
  housing.df$REMODEL REMODELNone REMODELOld REMODELRecent
1               None           1          0             0
2             Recent           0          0             1
3               None           1          0             0
4               None           1          0             0
5               None           1          0             0
6                Old           0          1             0
```

- Most algos require data to be ordered
- The REMODEL variable is categorical
- Is it ordered?
- Unordered?
- Unordered variables usually have to be coded as dummies

Note: R's lm() function automatically creates dummies, so you can skip dummy creation when using lm()

What will happen if you do not recode it?

# Handling Missing Data

WE can use statistical tests for missing evaluation:
e.g. Little's test of MCAR

- Most algorithms will not process records with missing values.
  - Default is to drop those records.
- Solution 1: Omission
  - If a small number of records have missing values, can omit them
  - If many records are missing values on a small set of variables, can drop those variables (or use proxies)
  - If many records have missing values, omission is not practical
- Solution 2: Imputation
  - Replace missing values with reasonable substitutes
  - Let's you keep the record and use the rest of its (non-missing) information

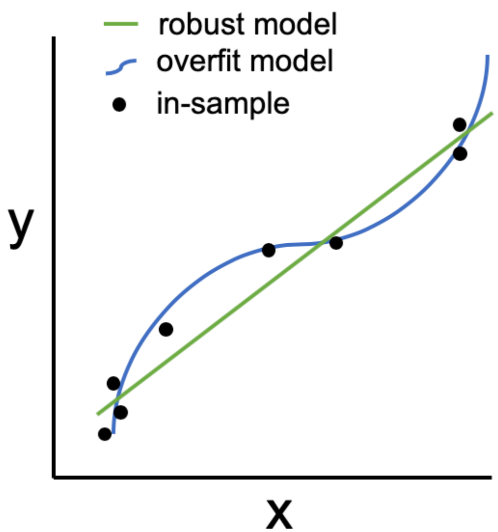
NB: Determine if "missingness" has value!!

# The Problem of Overfitting

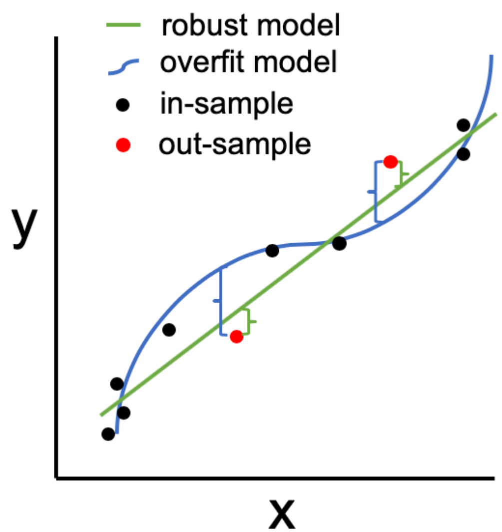
How can we balance fit and prediction?

- Statistical models can produce highly complex explanations of relationships between variables
- Causes:
  - Too many predictors (too many p, or too few n)
  - A model with too many parameters
  - Trying many different models
- (When p = n, we have perfect fit)
- Consequence: Deployed model will not work as well as expected with completely new data.

100% fit – Excellent!!

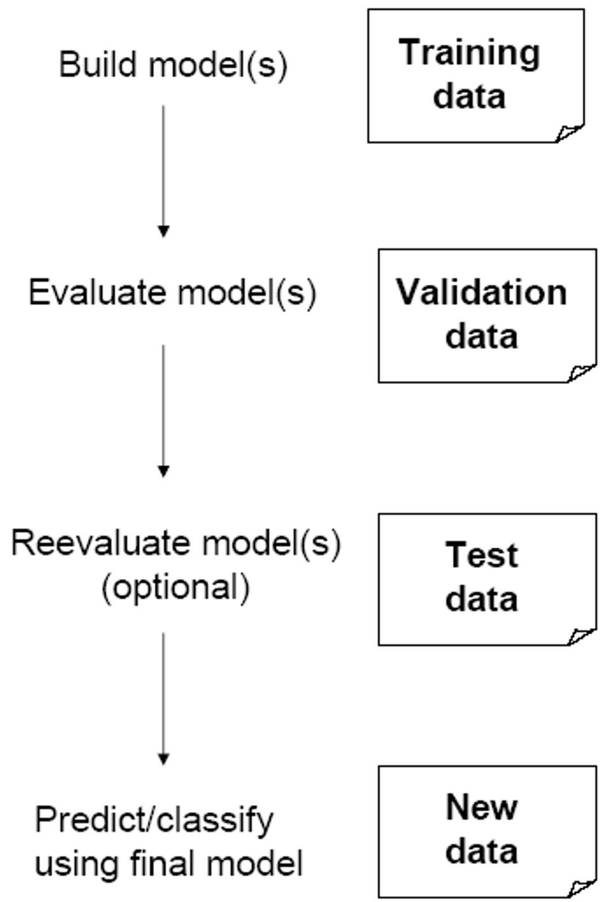



Or not!



# Test Partition

- When a model is developed on **training data**, it can overfit the training data (hence need to assess on validation)
- Assessing multiple models on same **validation data** can overfit validation data
- Some methods use the validation data to choose a parameter. This too can lead to overfitting the validation data
- Solution: final selected model is applied to a **<u>test</u> partition** to give unbiased estimate of its performance on new data



# Part 2
# Dimension Reduction

# Rotation: Change in Perspective




Video: https://www.youtube.com/watch?v=1VcWXwV0vW8

# Dimension Reduction: Decathlon

*One use of PCA is to reduce the dimensionality of data*

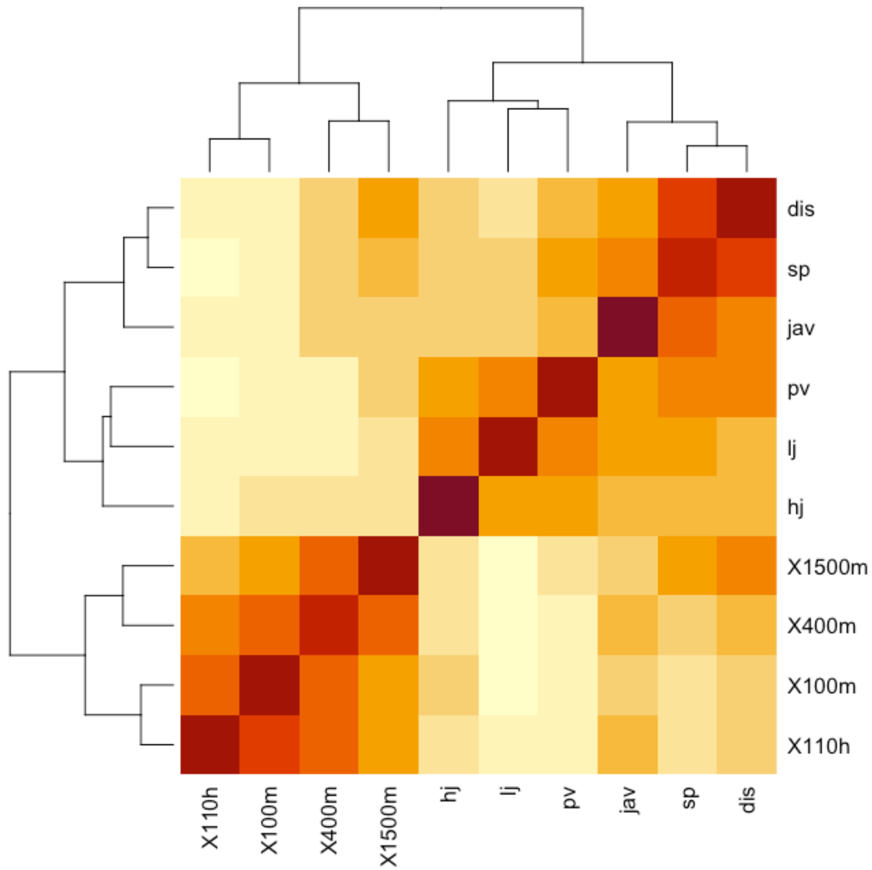
**Correlates:**

```
round(cor(decathlon),2)
```

Console   Terminal ×   Jobs ×

~/Google Drive/Teaching/Trinity/BU7143 BADM/Classes/Session 1-2/Session 1-2 RProj/

```
> round(cor(decathlon),2)
       X100m    lj    sp    hj X400m X110h   dis    pv   jav X1500m
X100m   1.00 -0.54 -0.21 -0.15  0.61  0.64 -0.05 -0.39 -0.06   0.26
lj     -0.54  1.00  0.14  0.27 -0.52 -0.48  0.04  0.35  0.18  -0.40
sp     -0.21  0.14  1.00  0.12  0.09 -0.30  0.81  0.48  0.60   0.27
hj     -0.15  0.27  0.12  1.00 -0.09 -0.31  0.15  0.21  0.12  -0.11
X400m   0.61 -0.52  0.09 -0.09  1.00  0.55  0.14 -0.32  0.12   0.59
X110h   0.64 -0.48 -0.30 -0.31  0.55  1.00 -0.11 -0.52 -0.06   0.14
dis    -0.05  0.04  0.81  0.15  0.14 -0.11  1.00  0.34  0.44   0.40
pv     -0.39  0.35  0.48  0.21 -0.32 -0.52  0.34  1.00  0.27  -0.03
jav    -0.06  0.18  0.60  0.12  0.12 -0.06  0.44  0.27  1.00   0.10
X1500m  0.26 -0.40  0.27 -0.11  0.59  0.14  0.40 -0.03  0.10   1.00
>
```




*These correlates capture WHAT?*

**Eigenvalues:**

```
decathlon_eigen <- eigen(cor(correlates))

decathlon_eigen$values
  [1] 2.43 0.96 0.35 0.26
```

***Eigenvalues:*** *Variance of each component*

```
sum(decathlon_eigen$values)
  [1] 4
```

*Eigenvalues sums up to number of dimensions*

```
decathlon_eigen$values / sum(decathlon_eigen$values)
[1] 0.61 0.24 0.09 0.07
```

*Ratio of eigenvalue/dimensions is variance captured!*

*PC1 captures >61% of original data's variance!*

# Interpreting Principal Components: Decathlon Example

***Examining the results of PCA***

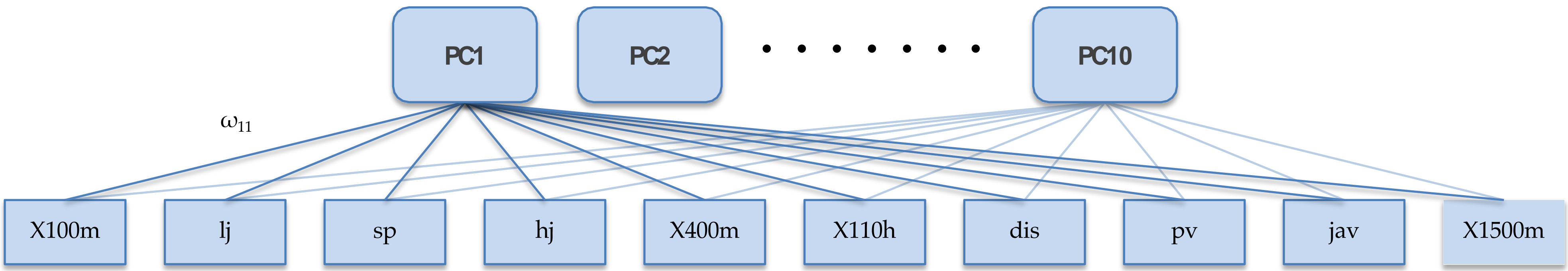
```
dec_pca <- prcomp(dec, scale. = TRUE)

dec_pca$rotation
        PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
X100m -0.42  0.15 -0.27  0.09 -0.44  0.03  0.25 -0.66  0.11 -0.11
lj      0.39 -0.15 -0.17  0.24  0.37 -0.09  0.75 -0.14 -0.05 -0.06
sp      0.27  0.48  0.10  0.11 -0.01  0.23 -0.11 -0.07 -0.42 -0.65
hj      0.21  0.03 -0.85 -0.39  0.00  0.07 -0.14  0.16  0.10 -0.12
X400m -0.36  0.35 -0.19 -0.08  0.15 -0.33  0.14  0.15 -0.65  0.34
X110h -0.43  0.07 -0.13  0.38 -0.09  0.21  0.27  0.64  0.21 -0.26
dis     0.18  0.50  0.05 -0.03  0.02  0.61  0.14 -0.01  0.17  0.53
pv      0.38  0.15  0.14 -0.14 -0.72 -0.35  0.27  0.28  0.02  0.07
jav     0.18  0.37 -0.19  0.60  0.10 -0.44 -0.34 -0.06  0.31  0.13
X1500m -0.17  0.42  0.22 -0.49  0.34 -0.30  0.19 -0.01  0.46 -0.24
```

**ω**: *“**weights**” are like regression coefficients between PC score and items (but they are still hard to interpret)*

***Confirming orthogonality of components***

```
round( cor(scores), 2)
     PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10
PC1    1   0   0   0   0   0   0   0   0    0
PC2    0   1   0   0   0   0   0   0   0    0
PC3    0   0   1   0   0   0   0   0   0    0
PC4    0   0   0   1   0   0   0   0   0    0
PC5    0   0   0   0   1   0   0   0   0    0
PC6    0   0   0   0   0   1   0   0   0    0
PC7    0   0   0   0   0   0   1   0   0    0
PC8    0   0   0   0   0   0   0   1   0    0
PC9    0   0   0   0   0   0   0   0   1    0
PC10   0   0   0   0   0   0   0   0   0    1
```



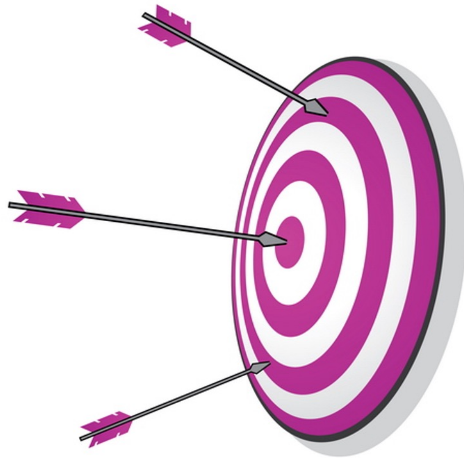

$$PC_i = w_{i1}\cdot X100m + w_{i2}\cdot lj + w_{i3}\cdot sp + w_{i4}\cdot hj + w_{i5}\cdot X400m + w_{i6}\cdot X110h + w_{i7}\cdot dis + w_{i8}\cdot pv + w_{i9}\cdot jav + w_{i10}\cdot X1500m$$

*The **scores** of each principal component is a **weighted sum** of our original dimensions*

# Summary

- **Data summarization** is an important for data exploration
- **Data summaries** include numerical metrics (average, median, etc.) and graphical summaries
- **Data reduction** is useful for compressing the information in the data into a smaller subset
    - Categorical variables can be reduced by combining similar categories
    - Principal components analysis transforms an original set of numerical data into a smaller set of weighted averages of the original data that contain most of the original information in less variables.

# Why Evaluate?



- Multiple methods are available to classify or predict
- For each method, multiple choices are available for settings
- To choose best model, need to assess each model's performance

# HW Suggestions

**CREATE well formatted reports**

Briefly summarize the question

Format it to distinguish:

*question / description / code / output / answers*

Show code and relevant text output

*use text, not screenshots*

Show relevant visualizations

*export graphics from Rstudio; not screenshots*

CREDIT peers who helped!!

Mention their ID at the top of your assignment!

Peers who help will get extra-credit at end-of-semester

**No screen shots of code, results, or visualizations!**