It is often the small steps, not the giant leaps, that bring about the most lasting change. – Queen Elizabeth II

Data Mining

DR. SUSAN SIMMONS

INSTITUTE FOR ADVANCED ANALYTICS

Overview

Splitting Data

Bootstrap method

Adjusting p-values

Dealing with transactional data

Dealing with missing values

Transformations/Standardizations

Association Analysis

Splitting data

Training/Validation/Test

Want to make sure your models are generalizable

- Not just good models of training sample
- Can predict equally well on out-of-sample data

Split into Training/Validation/Test sets (no set amount for each)

- Some common splits observed are:
  - Lots of data? 50-40-10 split
  - Not so much data? 70-20-10 split
  - Not enough data? Use cross-validation (have a training/test set)

How do you know when you do NOT have enough data? NO hard rule, but good to have at least 10 observations per variable.

Training/Validation

Use the training data to build your model

Evaluate and tune the model based on how it performs on the validation data (careful to NOT “train” on validation data!!)

NEVER report accuracy measures from the training data!!! Best to state on the TEST data!

Model creation

Continually adapting a model to perform better on the validation data is essentially training your model to the validation data (DON'T do this). Model creation should be on the training data and then applied to validation to see if you might need to enhance it.

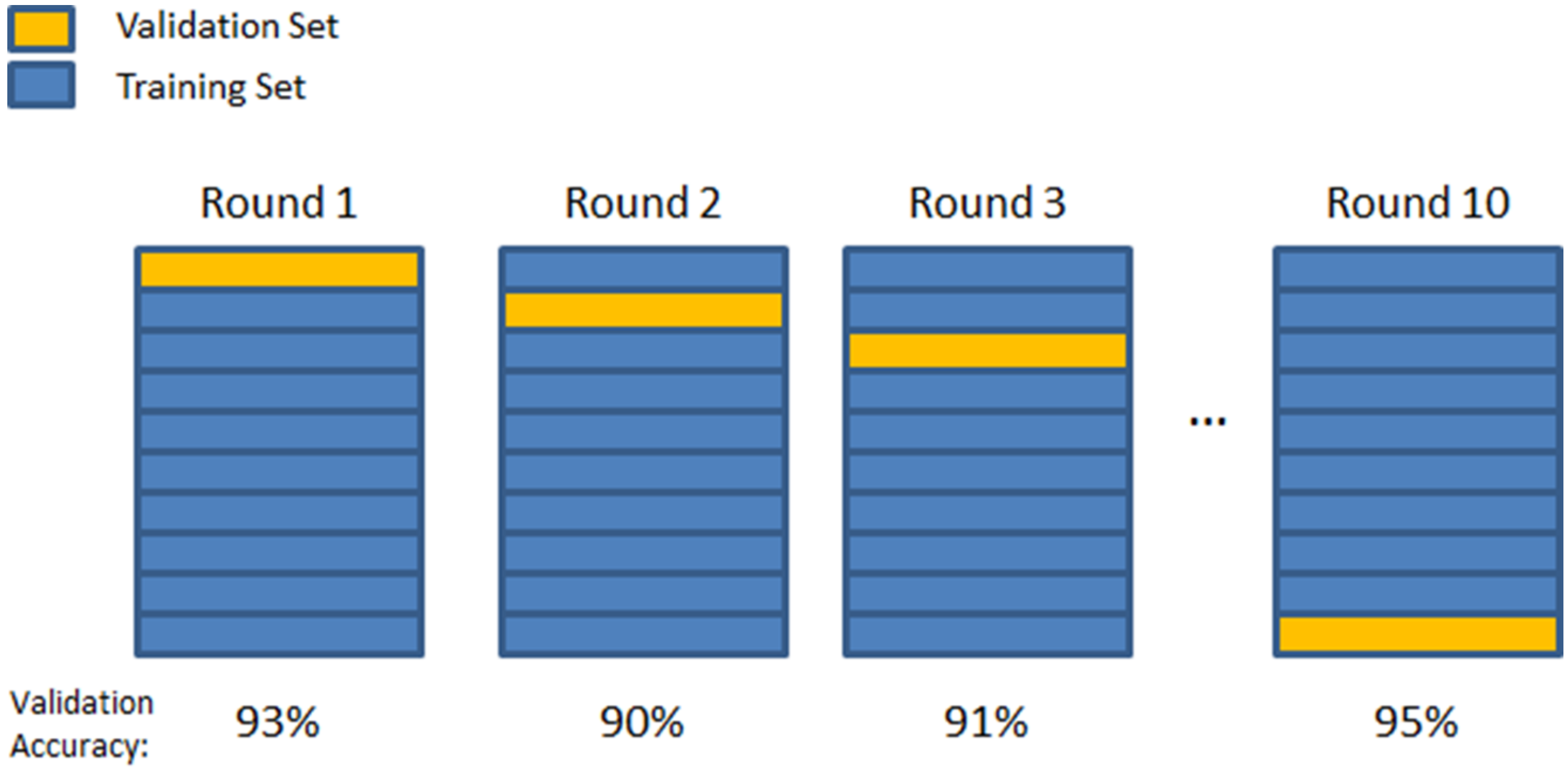
Once a final model is chosen, re-run this model on training + validation data to finalize your parameters (at this stage, your model is set…you are just updating the parameters). Use this model to run on the test data set (this is the accuracy that should be reported).

Before deploying final model, you can use ALL data to update parameters!

K-fold Cross-Validation

- Divide your data into k equally-sized samples (*folds*)
  - k=10 or k=100 are common.
  - Depends on time complexity of model and size of the dataset!
- For each fold, train the model on all other data, using that fold as a validation set
- Record measures of error/accuracy
- In the end, report summary of error/accuracy (average, std. deviation etc)
- Use that report summary to choose a model

10-fold Cross-Validation



Final Accuracy = Average(Round 1, Round 2, ...)

Cross-Validation

- Can use cross-validation in any situation.

- Will be necessary if you do not have **sufficient** observations to split into training/validation/test

- What is **sufficient**? It depends!
  - **Rule of thumb:** AT LEAST 10 observations per input variable in training set
  - Don’t Forget: For **categorical variables – each level counts!**

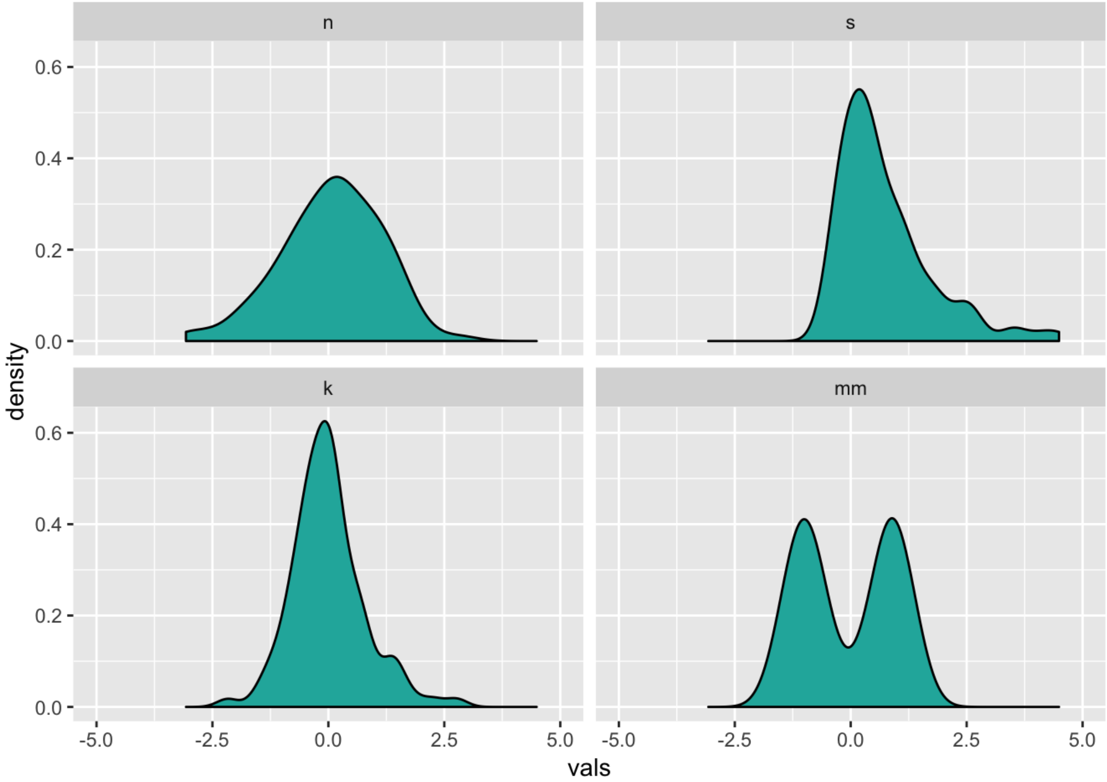
Leave-One-Out Cross-Validation (Jackknife)

- *n*-fold cross validation where *n* is number of obs.
- Use only one observation as the validation-set
- Repeat for every observation in the dataset

Can be extremely time consuming! Only use when necessary (very small sample sizes)

Bootstrapping

- Developed by Bradley Efron back in the late 1970’s
- Nonparametric procedure that can estimate standard error of a statistic, compute confidence intervals for a statistic or perform hypothesis test
- Use data as the population and resample (with replacement)
- This resampling is used to estimate the distribution of the quantity of interest



Original Data	Bootstrap Sample
1	18
2	2
3	9
4	4
5	7
6	6
7	8
8	3
9	8
10	16
11	17
12	5
13	3
14	10
15	12
16	16
17	1
18	11

Sample with replacement

Same sample size

Calculate statistic

$\bar{x} = 8.6667$

Repeat this process again and again and again

Bootstrap sample 1…..$\bar{x} = 8.6667$

Bootstrap sample 2…..$\bar{x} = 7.69$

Bootstrap sample 3…..$\bar{x} = 9.24$

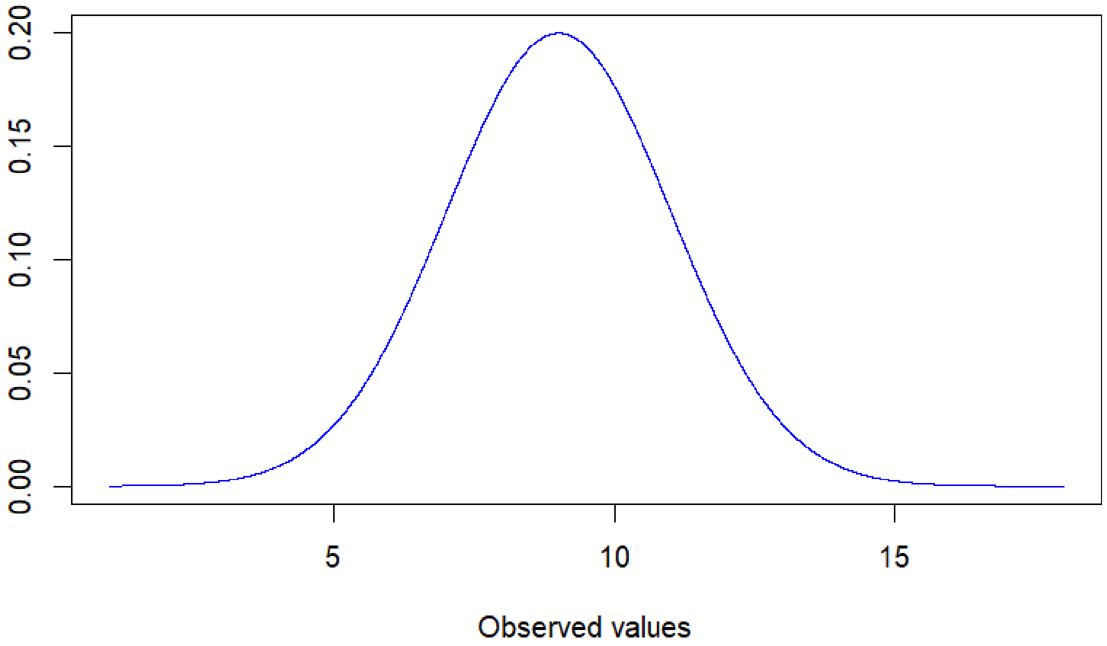
Bootstrap sample 4…..$\bar{x} = 7.11$

▪

▪

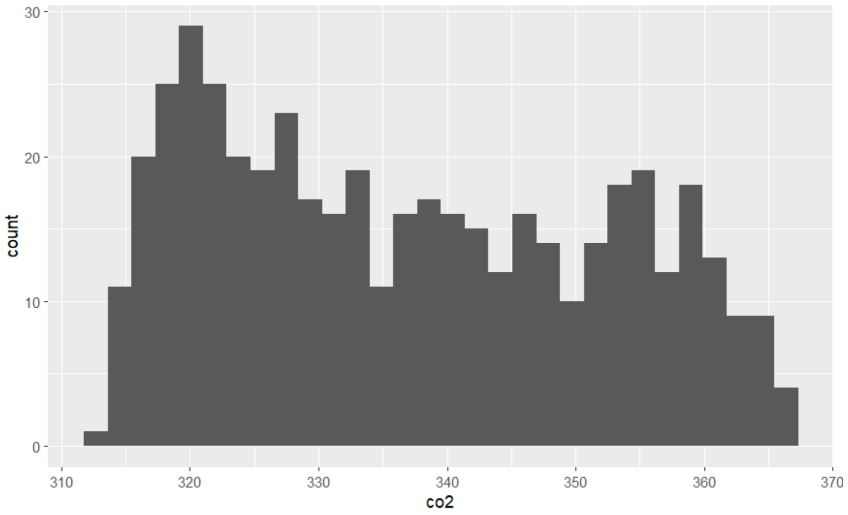
▪

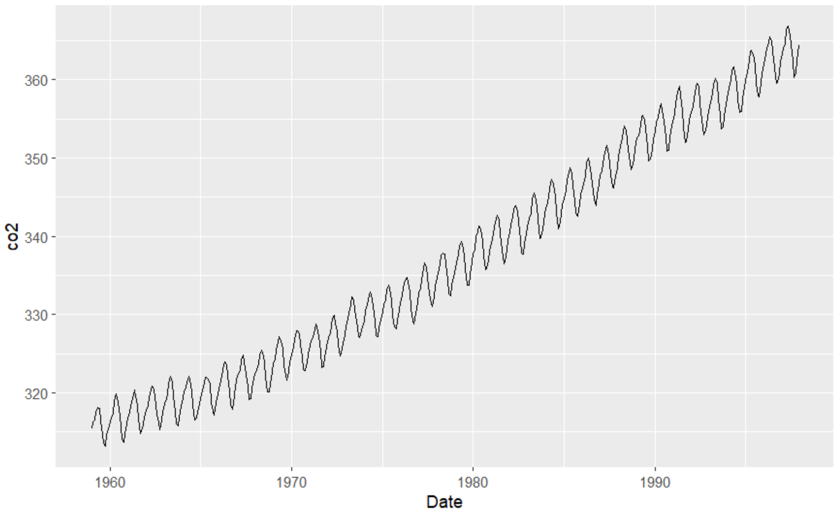
From sample statistics, we get sampling distribution for the statistic!!!



Examples

In the datasets package in R, the data set co2 contains information on the atmospheric concentrations of CO2 in ppm from 1959 to 1997 (data is monthly).





Variability of the median

Want to estimate the variability (standard error) of the median

Can basically do this for any statistic, too!!

Algorithm

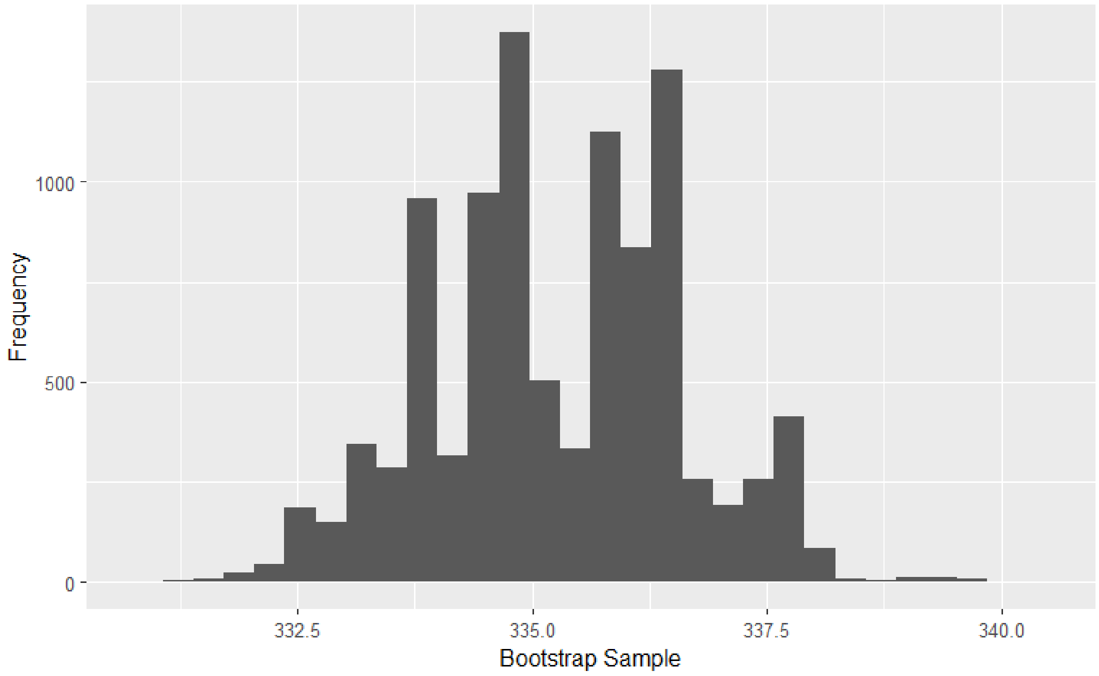
1. Get a bootstrap sample of the data (sample with replacement)

2. Calculate statistic of interest

Do steps 1 and 2 over and over and over again to get the distribution of the Median. Then get quantiles (for 95% confidence interval, use 2.5% and 97.5%).

```
act.med=median(co2)
boot.med=vector(length=10000)
for (i in 1:length(boot.med)){
  boot.samp=sample(co2,replace=T)
  boot.med[i]=median(boot.samp)
}
sd(boot.med)
act.med
quantile(boot.med,probs = c(0.025,0.975))
ggplot(data=data.frame(boot.med),aes(x=boot.med))+geom_histogram() + labs(x="Bootstrap Sample",y="Frequency")
```

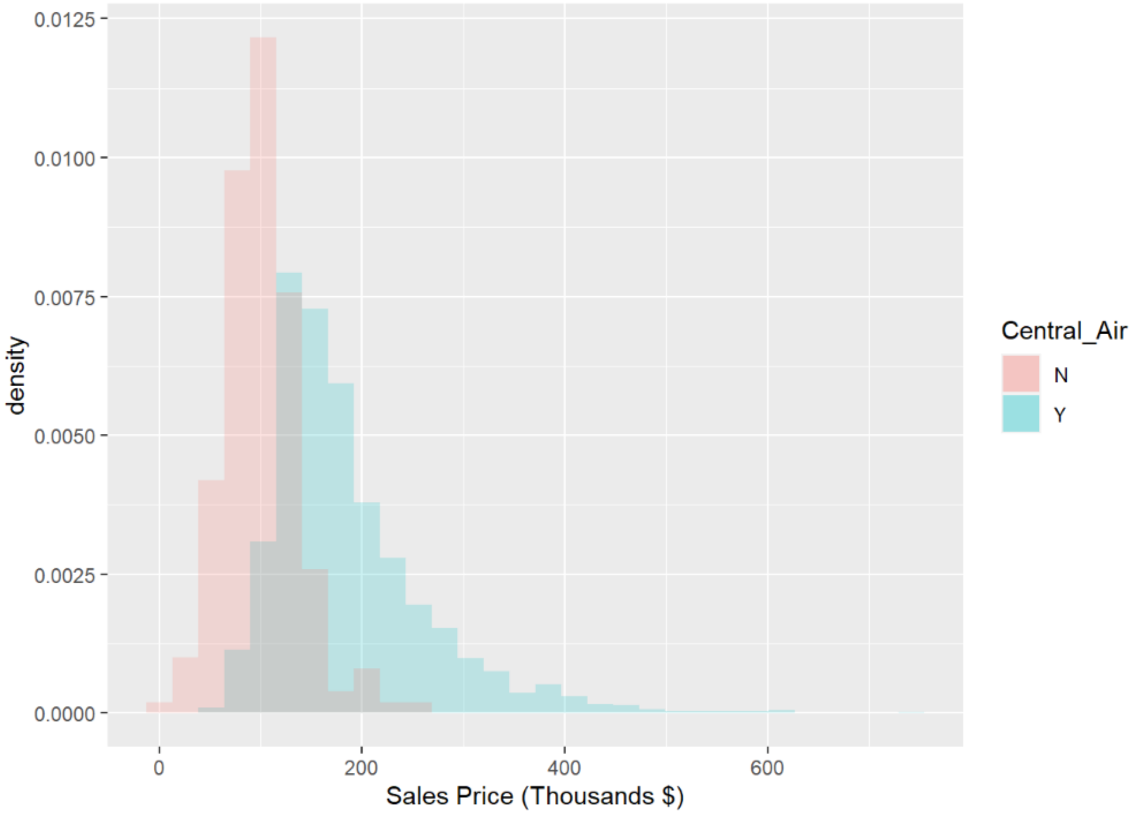
Actual Median is 335.17
Std Dev of Median is 1.34
95% confidence interval is:
332.675, 337.800



Using bootstrap for testing if two medians are significantly different

Can use bootstrap to get the distribution of differences in medians (we can see if confidence interval includes 0!).

Recall Sales Price from the Ames housing data set. We wanted to know if the median sales price for those homes with air conditioning is significantly different than the median sales price of those homes without air conditioning. We will create confidence interval for Median$_{YES}$ – Median$_{NO}$.

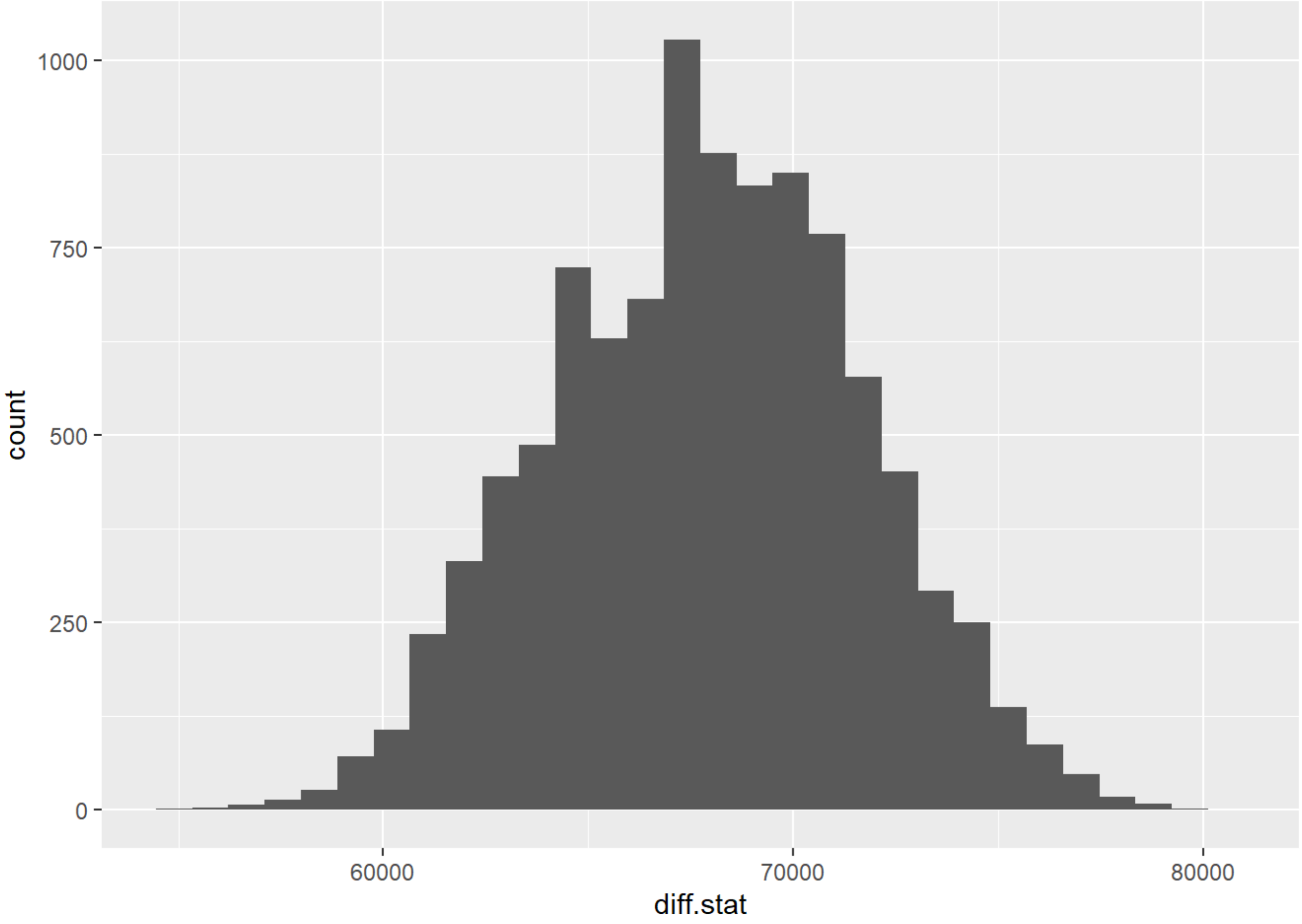


Distribution

```
diff.stat=vector(length=10000)
yes<-ames %>% filter(Central_Air=="Y")
%>% pull(Sale_Price)
no<-ames %>% filter(Central_Air=="N")
%>% pull(Sale_Price)

for (i in 1:10000){
yes.vec <-
median(sample(yes,length(yes),replace = T))
no.vec <-
median(sample(no,length(no),replace = T))

diff.stat[i]<- yes.vec-no.vec}
```



Adjusting p-values

Adjusting p-values

We’ve already talked about adjusting significance levels when your sample is large (due to *n* inherently making very small p-values)

IF you are doing A LOT of hypothesis testing, then you need to be aware of inflating your Type I error (remember, we learned about controlling the “experiment-wise error”)

Family-wise error rate is the same idea…we are controlling the overall probability of making a Type 1 error. Bonferroni is an example of a technique that controls the FWER.

The Bonferroni adjustment simply multiplies p-values by the number of tests you are doing….these are adjusted p-values

Example:

Say we are conducting 4 hypothesis tests and got the following p-values:

0.001, 0.03, 0.2, 0.4

To adjust these p-values for the number of tests, we need to multiply each of these by the number 4. The adjusted p-values are now:

0.004, 0.12, 0.8, 1 (notice that you do NOT go greater than 1)

As you can see, this can be very stringent!

FDR (False Discovery Rate)

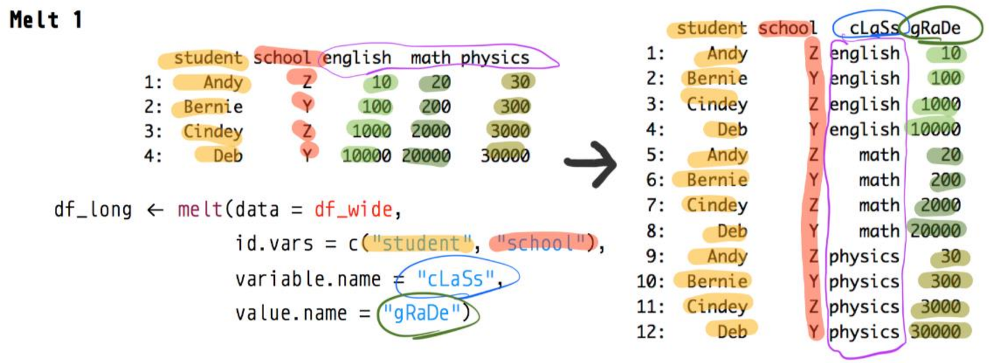
Significance level (α) – Controls the Type 1 error rate for an individual hypothesis (recall that Type 1 is when you reject a null hypothesis when it really should NOT be rejected).

False Discovery Rate (FDR) – controls ***rate*** of Type 1 errors. This is the expected proportion of “false discoveries” (does NOT control FWER, but does prevent us from finding too many significant tests).

In R

```
temp=c(0.001,0.03,0.2,0.4)
#Bonferoni
p.adjust(temp,method="bonferroni")
#Benjamini & Hochberg
p.adjust(temp,method="BH")
```

```
[1] 0.004       0.120       0.800       1.000       ##Bonferoni
[1] 0.0040000   0.0600000   0.2666667   0.4000000 # BH
```



Melt 1

```
   student school english  math physics
1:    Andy      Z      10    20      30
2:  Bernie      Y     100   200     300
3:  Cindey      Z    1000  2000    3000
4:     Deb      Y   10000 20000   30000
```

```
df_long ← melt(data = df_wide,
               id.vars = c("student", "school"),
               variable.name = "cLaSs",
               value.name = "gRaDe")
```

```
    student school   cLaSs gRaDe
 1:    Andy      Z english    10
 2:  Bernie      Y english   100
 3:  Cindey      Z english  1000
 4:     Deb      Y english 10000
 5:    Andy      Z    math    20
 6:  Bernie      Y    math   200
 7:  Cindey      Z    math  2000
 8:     Deb      Y    math 20000
 9:    Andy      Z physics    30
10:  Bernie      Y physics   300
11:  Cindey      Z physics  3000
12:     Deb      Y physics 30000
```

Dealing with Transactional Data

MOVING FROM LONG TO WIDE

Transactional Data

Person 1

1	2000
1	500
1	2500
1	5000

Person 2

2	18000
2	10000
2	300
2	NA

Person 3

3	600
3	200
3	100

Transactional data is LONG and has many rows per modeling observation!

Transaction Data

- Typically, the solution for modeling with transactional data is to “roll it up” so it has one row per observation modeled.

- It is transformed from long to **wide**

- Can use “group_by” in dplyr…for example…(see R code)

```
new.check = check %>% group_by(ID) %>%
summarise(mean.check=mean(Checking,na.rm=T),std.check=
sd(Checking,na.rm=T))
```

Transaction Data

A **subset** of columns we might consider in the process:

1. ID
2. Date of first transaction
3. Date of last transaction
4. Total number of transactions
5. Average time between transactions
6. Maximum number of items purchased
7. Average number of items purchased
8. Minimum number of items purchased
9. Std Deviation of number of items purchased
10. Maximum cost of items purchased
11. Average cost of items purchased
12. Minimum cost of items purchased
13. Stand. Deviation of cost of items purchased
14. Slope of regression line of cost over time

Data Cleaning:
Handling missing values

If you have missing values:

HIGHLY RECOMMEND:

Create a flag to indicate which values are missing and which ones are not (sometimes, missingness is informative!!)

NUMERIC:

Consider how much of the variable is missing (if over 50% need to consider how much information this variable is giving)

If you want to keep the variable, you can either 1. Impute values (you will talk about this later) or 2. bin the variables and create a separate bin for missing values

CATEGORICAL/ORDINAL

You can consider creating a “bin” for missing values (again, if too much is missing, this will be a HUGE bin…how much information is this providing?)

Missing Value Imputation

Imputation: Replacing missing values with a substitute value, typically a guess at what you think the value should have been (can be mean or median or mode of the variable; more sophisticated packages such as MICE and RF imputation)

* Keep in mind that you are “falsifying records”…i.e. making up data

Imputing Missing Values

Obs.	Gender	Q1 Response
1	M	5
2	M	4
3	F	NA
4	M	1
5	F	NA

Obs.	Gender	Q1 Response	Q1 Flag
1	M	5	0
2	M	4	0
3	F	3	1
4	M	1	0
5	F	3	1

Always! Always! Always create a binary flag = 1 indicating that the value has been imputed and include the flag in your model. Nonresponse might be an important indicator of target

PAY ATTENTION

- Blind imputation can potentially generate impossible or highly unlikely data
- For Example:
  - A 16 year old who makes $80,000 a year
  - A male patient who is menopausal

Soooo, what do you do?

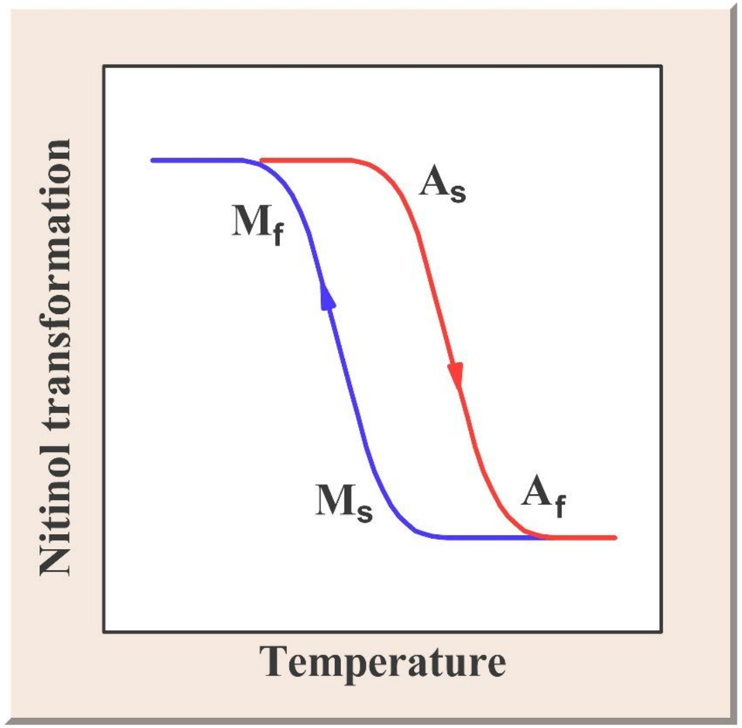
IT DEPENDS!!!

- Only the person closest to the data and to the problem can make these judgment calls!

- Can try several methods to see what works best.

- The binary flag indicating imputed value will show you if there is something special about missing values.

Transformations/Standardization

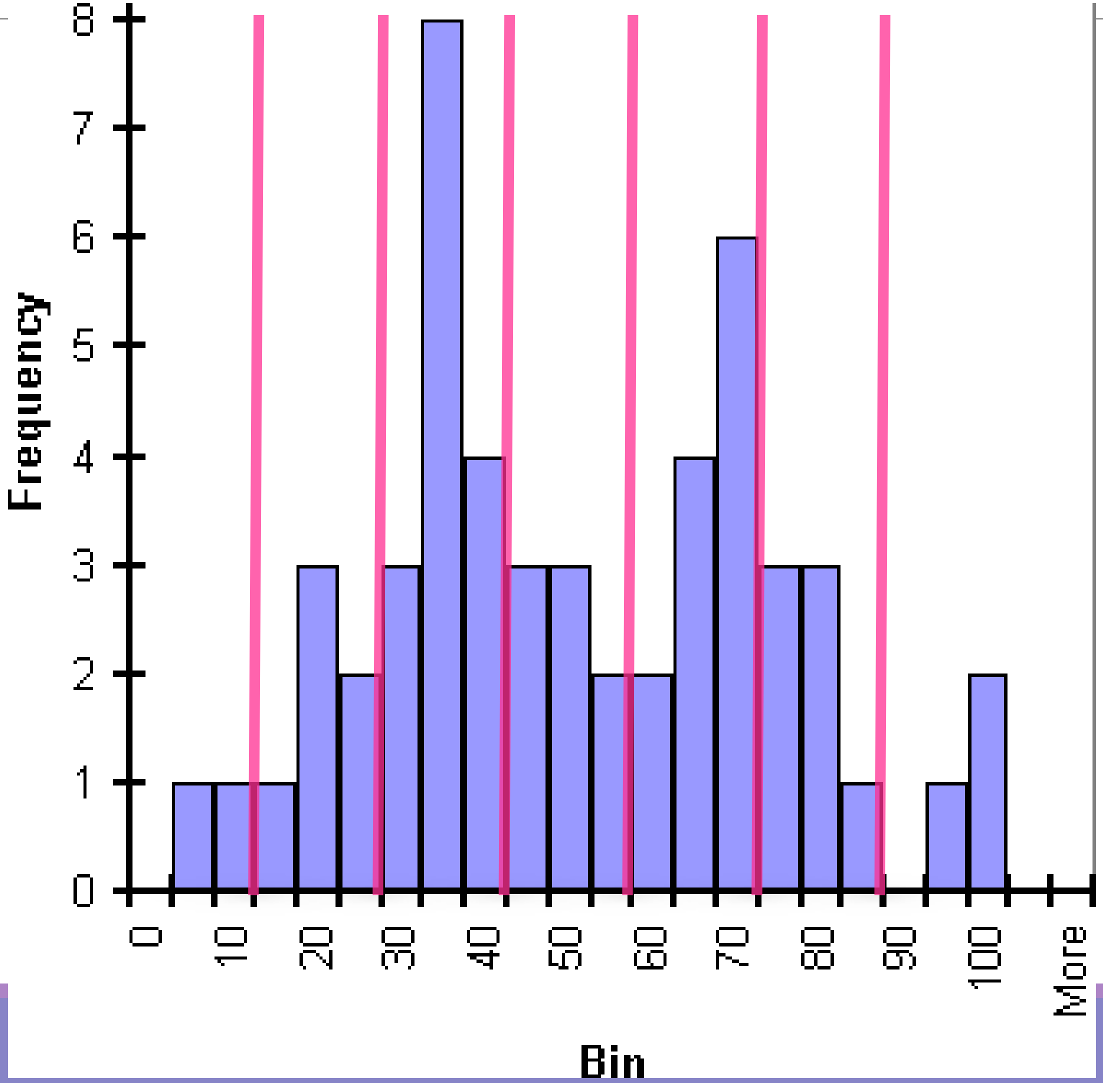


Variable Transformations

- Discretizing (Binning) Numeric Variables
  - Equal Width
  - Equal Depth
  - Supervised Binning
- Standardization and Normalization
  - Statistical Standardization
  - Range, MinMax Standardization

Binning Numeric Variables

Unsupervised Approach 1: *Equal Width*

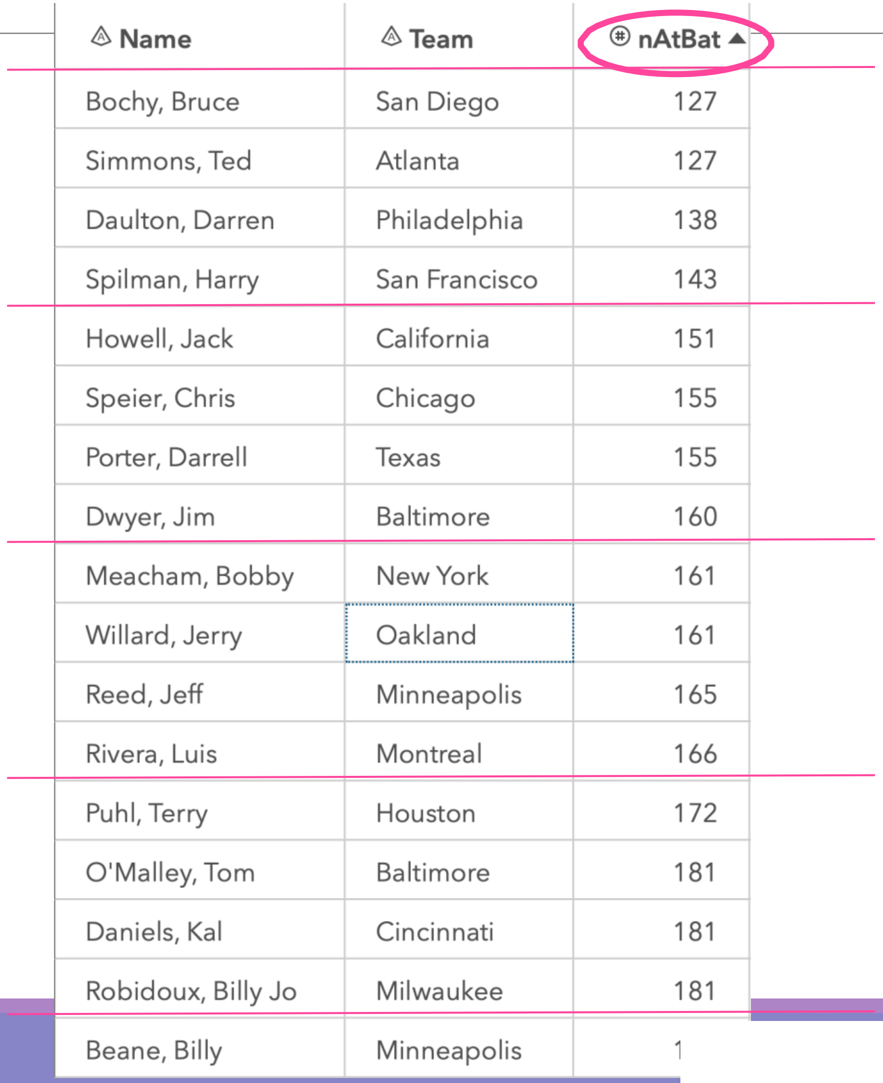


Each bin has the same width in variable values

Each bin has different number of observations

Binning Numeric Variables

Unsupervised Approach 2: *Equal Depth*



Name	Team	nAtBat ▲
Bochy, Bruce	San Diego	127
Simmons, Ted	Atlanta	127
Daulton, Darren	Philadelphia	138
Spilman, Harry	San Francisco	143
Howell, Jack	California	151
Speier, Chris	Chicago	155
Porter, Darrell	Texas	155
Dwyer, Jim	Baltimore	160
Meacham, Bobby	New York	161
Willard, Jerry	Oakland	161
Reed, Jeff	Minneapolis	165
Rivera, Luis	Montreal	166
Puhl, Terry	Houston	172
O'Malley, Tom	Baltimore	181
Daniels, Kal	Cincinnati	181
Robidoux, Billy Jo	Milwaukee	181
Beane, Billy	Minneapolis	1

Take percentiles of the population.

Each bin has the same number of observations.

Binning Numeric Variables

Supervised Approach

- Use target variable info to ‘optimally’ bin numeric variables for prediction.

- *Typically* used in classification problems.

- Want bins that result in the most *pure* set of target classes.

Binning Numeric Variables

Supervised Approach

- Decision tree methods can be helpful to create these bins (conditional trees).

- Also, weight of evidence (WOE)

- More on these techniques later.

Standardization and Normalization

- Standardization in statistics (Z-score standardization) transform units to “number of standard deviations away from the mean”:

$$\frac{x - \bar{x}}{\sigma_x}$$

- Avoid having variable with large values (e.g. income) dominate a calculation.
- Many other ways to standardize/normalize
  - Range Standardization: Divide by the range of the variable
  - MinMax Standardization: Subtract min. and divide by (max-min.)
    - Puts variable on a scale from 0 to 1
  - Divide by 2-norm, Divide by 1-norm, Divide by sum

Transformation Considerations

- Transformations change the nature of the data.
  - Ex: x={1,2,3} transform to 1/x = $\{1, \frac{1}{2}, \frac{1}{3}\}$
    - The sorting order of the observations reverses
    - Observations close to 0 will get **very** large
- Always consider the following questions:
  - Does the order of the data need to be maintained? (other code/documentation)
  - Does the transformation apply to all values, especially negative values and 0? (Think log(x) and 1/x)
  - What is the effect on values between 0 and 1?

Association Analysis

Famous Example

June 1992 study by NCR (now TeraData) for Osco Drug found some interesting associations of products that are frequently bought together:

- Beer and diapers
- Fruit juice and cough syrup…and numerous others!!

Association Analysis

Unsupervised approach (no target or outcome variable for training!)….searching for patterns in the data

Association analysis gives us sets of products that are likely to be purchased together

Can be used in retail (examples: coupon marketing, targeted upselling and product placement), medical (example: diagnoses that appear together), repairs (example: what type of repairs are seen together)

And many, many other situations!!

Small grocery data set

Transaction data…will need it to be wide!!!

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Small grocery data set

Transaction data…will need it to be wide!!!

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

For those who buy butter, do they tend to also buy bread?

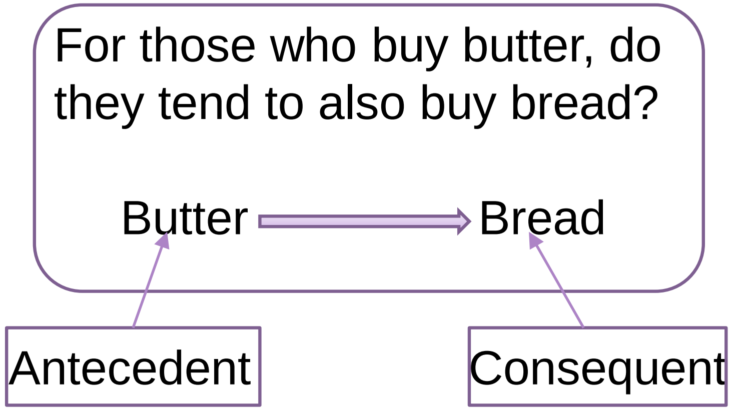
Butter ⇒ Bread

Small grocery data set

Transaction data…will need it to be wide!!!

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Interpretation: Someone who buys butter is also likely to (simultaneously) buy bread



For those who buy butter, do they tend to also buy bread?

Quantifying Association Rules

The STRENGH of an association rule A⟹B is quantified using three statistics:

1. Support: P(A∩B) = P(A and B)
   - Measures how often we find instances of this rule in the data
2. Confidence: P(B|A) = $\frac{P(A \cap B)}{P(A)}$
   - Measures what percent of transactions containing A also contain B
3. Lift: $\frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A)P(B)}$
   - Measures how much more likely we are to buy B given that we also buy A than we are to buy B at random.
   - Want lift values greater than 1!!

Small grocery data set

Calculate the support of Butter

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Small grocery data set

Calculate the support of Butter

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Answer: 3/11 = 0.2727

Small grocery data set

Calculate the ***support*** of Butter ⇒ Bread

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Small grocery data set

Calculate the ***support*** of Butter and Bread

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Answer: 3/11=0.2727

We see Butter and Bread in 27.25% of the transactions.

Small grocery data set

Calculate the ***confidence*** of Butter ⇒ Bread

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Small grocery data set

Calculate the ***confidence*** of Butter ⇒ Bread

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Answer: (3/11)/(3/11)=1

Of those purchases containing Butter, 100% of them also purchased Bread.

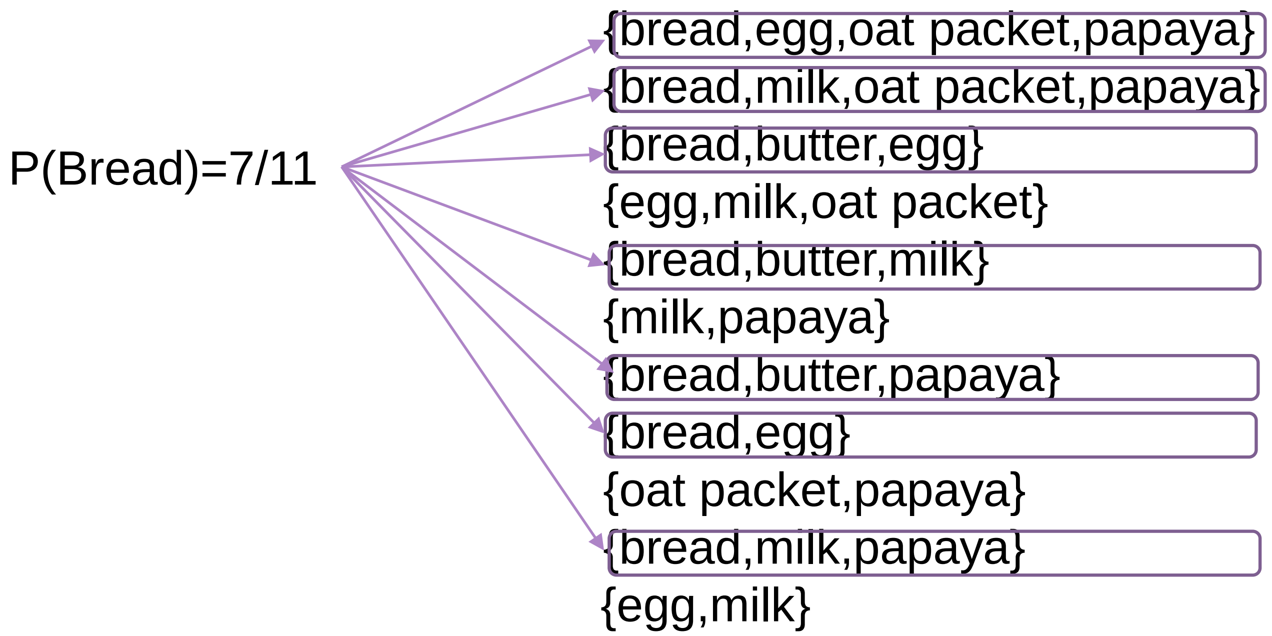
Small grocery data set

Calculate the ***lift*** of Butter ⇒ Bread

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Small grocery data set

Calculate the ***lift*** of Butter ⟹ Bread



P(Bread)=7/11

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Answer: 1/(7/11)=1.57

We are 1.5 times more likely to see Bread bought with Butter than Bread bought randomly.

Some Post-Hoc Take-aways

Product A → Product B

Product B as a consequent: Determine what can be done to boost its sales.

- Product placement
- Optimized upselling
- Coupons for related products

Product A as antecedent: Determine what other products would be affected by changes to Product A

- If we discontinued A, what other products might be affected
- If price changes on A, what other products might be affected
- Potential Cannibalization

Direction of Association

A → B versus B → A

Same Support

Same Lift

Different Confidence

(In this analysis, there is NO time component!!)

We do NOT say “those who buy A will THEN buy B.”

Finding Association Rules

Most algorithms have two parts:

- Itemset generation: find all sets of items that satisfy some minimum support
- Rule generation: determine which sets generated in step 1 satisfy some minimum confidence
- For more details of each part, see text by Tan, Steinback and Kumar

In R (arules), you need to create a “transaction data set”

```
trans.dat <- as(split(temp.dat$Grocery, temp.dat$ID), "transactions")
inspect(trans.dat)
```

The code above will take a long data set and turn it into a transaction data set. You need two columns to do this (a column with the transactions and a column identifying the ID). The inspect statement just prints the transaction data (if data is really long, you might want to just pull of a couple of rows to ensure it is doing what you want it to do).

Data

{bread,egg,oat packet,papaya}	1
{bread,milk,oat packet,papaya}	2
{bread,butter,egg}	3
{egg,milk,oat packet}	4
{bread,butter,milk}	5
{milk,papaya}	6
{bread,butter,papaya}	7
{bread,egg}	8
{oat packet,papaya}	9
{bread,milk,papaya}	10
{egg,milk}	11

Some ways to view items:

We can view the individual items by:

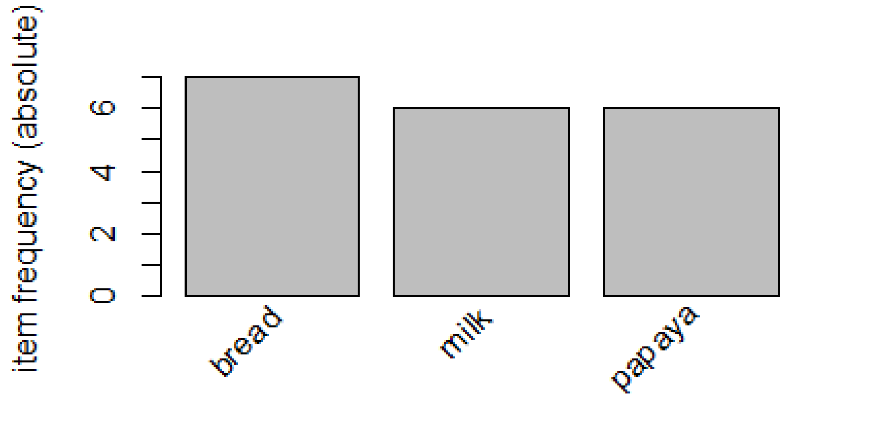
```
trans.dat@itemInfo$labels

[1] "bread"      "butter"     "egg"
[4] "milk"       "oat packet" "papaya"
```

absolute or relative for the height of the bars

Or create a plot of the 3 most common items:

```
itemFrequencyPlot(trans.dat,topN=3,type="absolute")
```



To actually run the analysis:

```
rules <- apriori(trans.dat, parameter = list(supp = 0.1, conf = 0.001, target="rules"))
rules<-sort(rules, by="confidence", decreasing=TRUE)
```

There are a number of parameters that you can control. Here are just a couple (setting the minimum support and confidence to use in creating rules….too low and you will get A LOT of rules (may take A LONG time)…too high and you might not get anything!!

A couple of association rules:

```
inspect(rules[1:4])

    lhs                       rhs         support     confidence
[1] {butter}               => {bread}     0.2727273   1.0000000
[2] {bread,oat packet}     => {papaya}    0.1818182   1.0000000
[3] {oat packet}           => {papaya}    0.2727273   0.7500000
[4] {papaya}               => {bread}     0.3636364   0.6666667

    coverage     lift        count
[1] 0.2727273    1.571429    3
[2] 0.1818182    1.833333    2
[3] 0.3636364    1.375000    3
[4] 0.5454545    1.047619    4
```

Some other ways of getting rules:

```
oat.rules = apriori(trans.dat, parameter = list(supp=0.001, conf=0.8),appearance =
list(default="lhs",rhs="oat packet"))
```

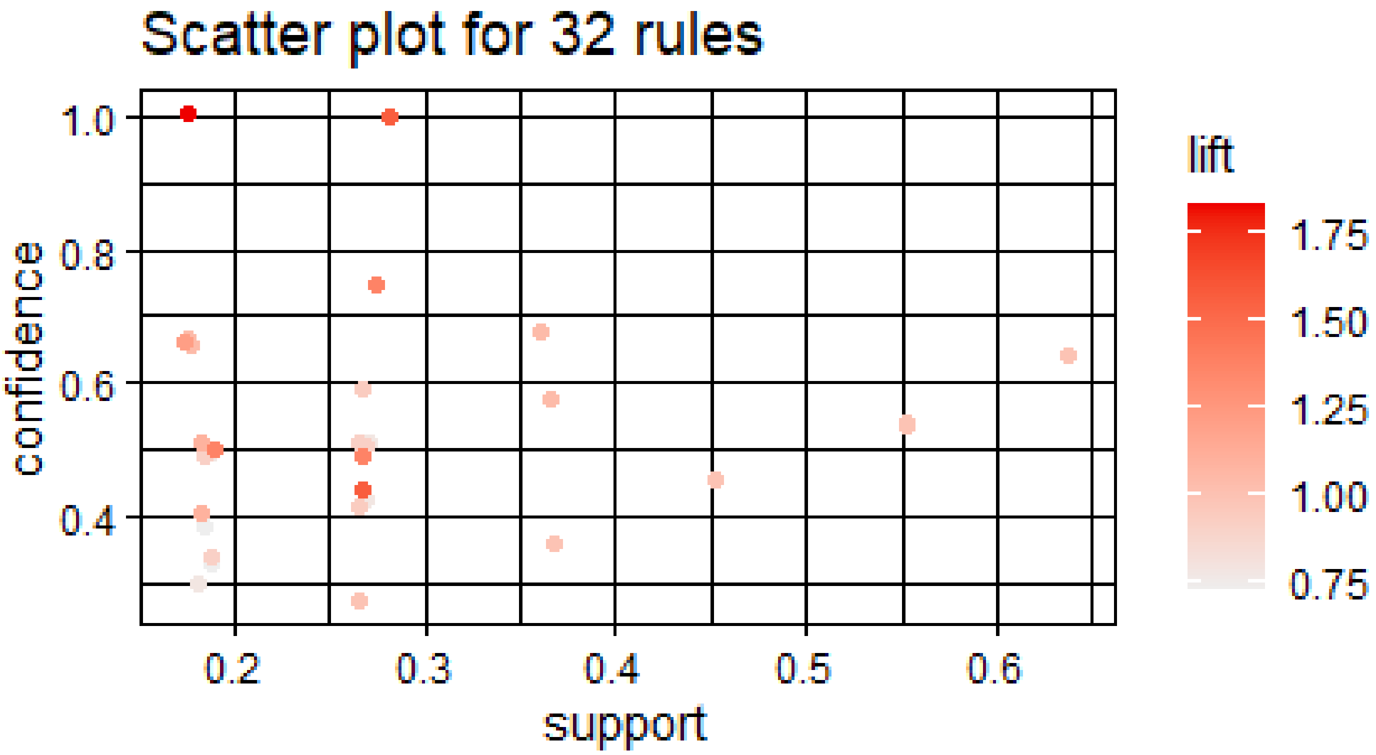
```
inspect(oat.rules)
         lhs                                 rhs          support
[1] {egg,papaya}           => {oat packet} 0.09090909
[2] {bread,egg,papaya} => {oat packet} 0.09090909
    confidence coverage       lift    count
[1] 1                 0.09090909 2.75   1
[2] 1                 0.09090909 2.75   1
```

```
oat.rules2 = apriori(trans.dat, parameter = list(supp=0.001, conf=0.8),appearance =
list(lhs="oat packet", default="rhs"))
```

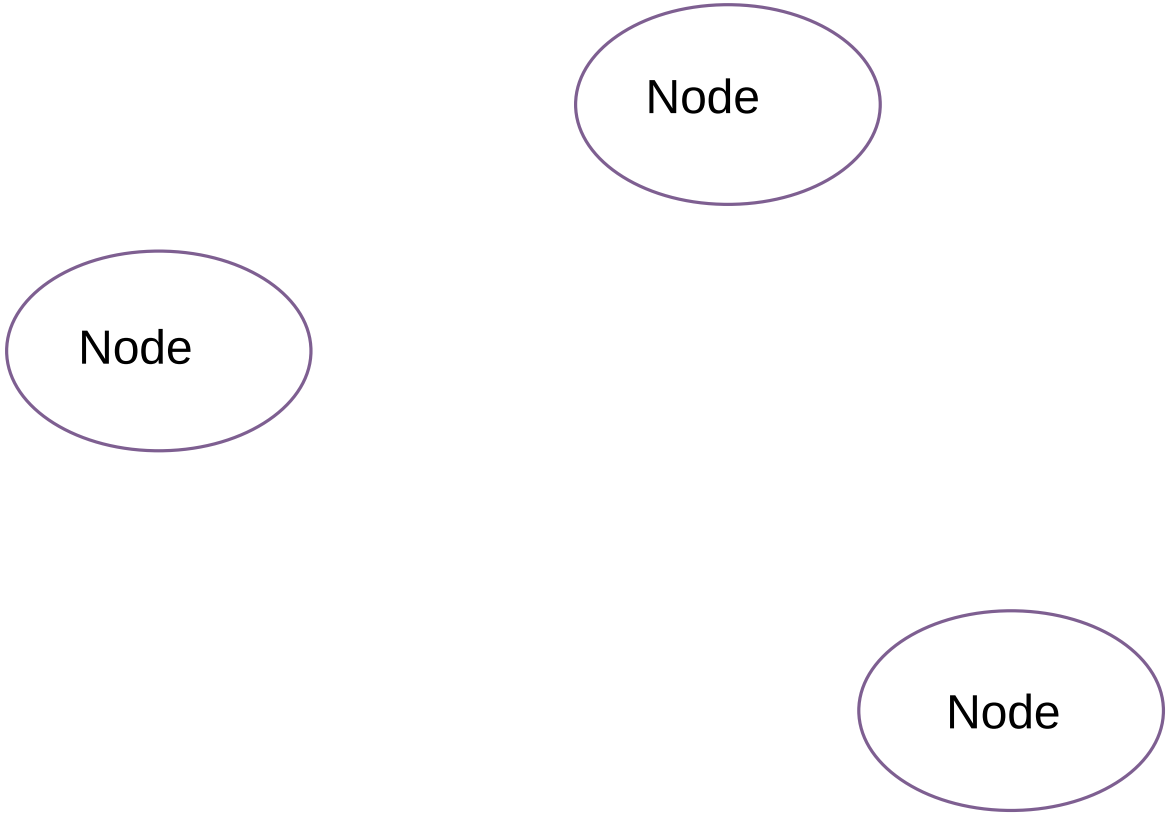
```
inspect(oat.rules2)
 NO RULES!!!!!
```

Visualizing rules:

plot(rules)



Small background on graphs…

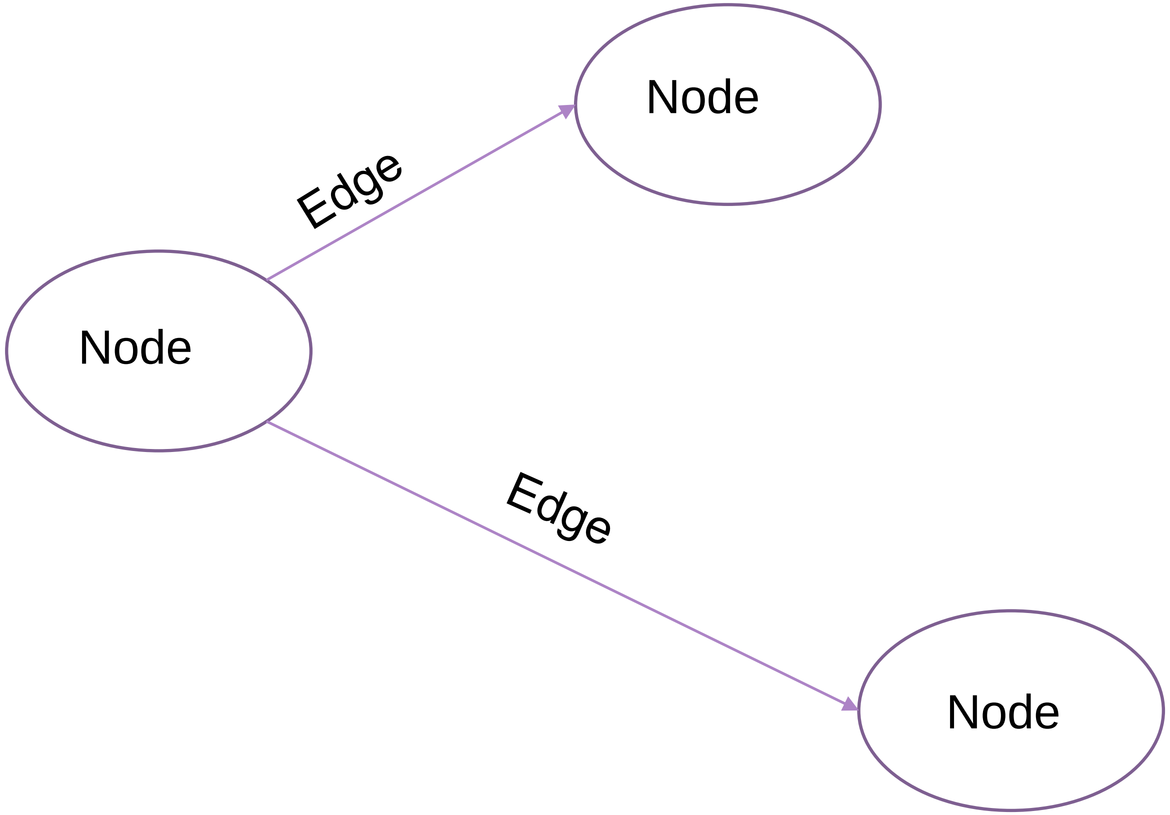


Node

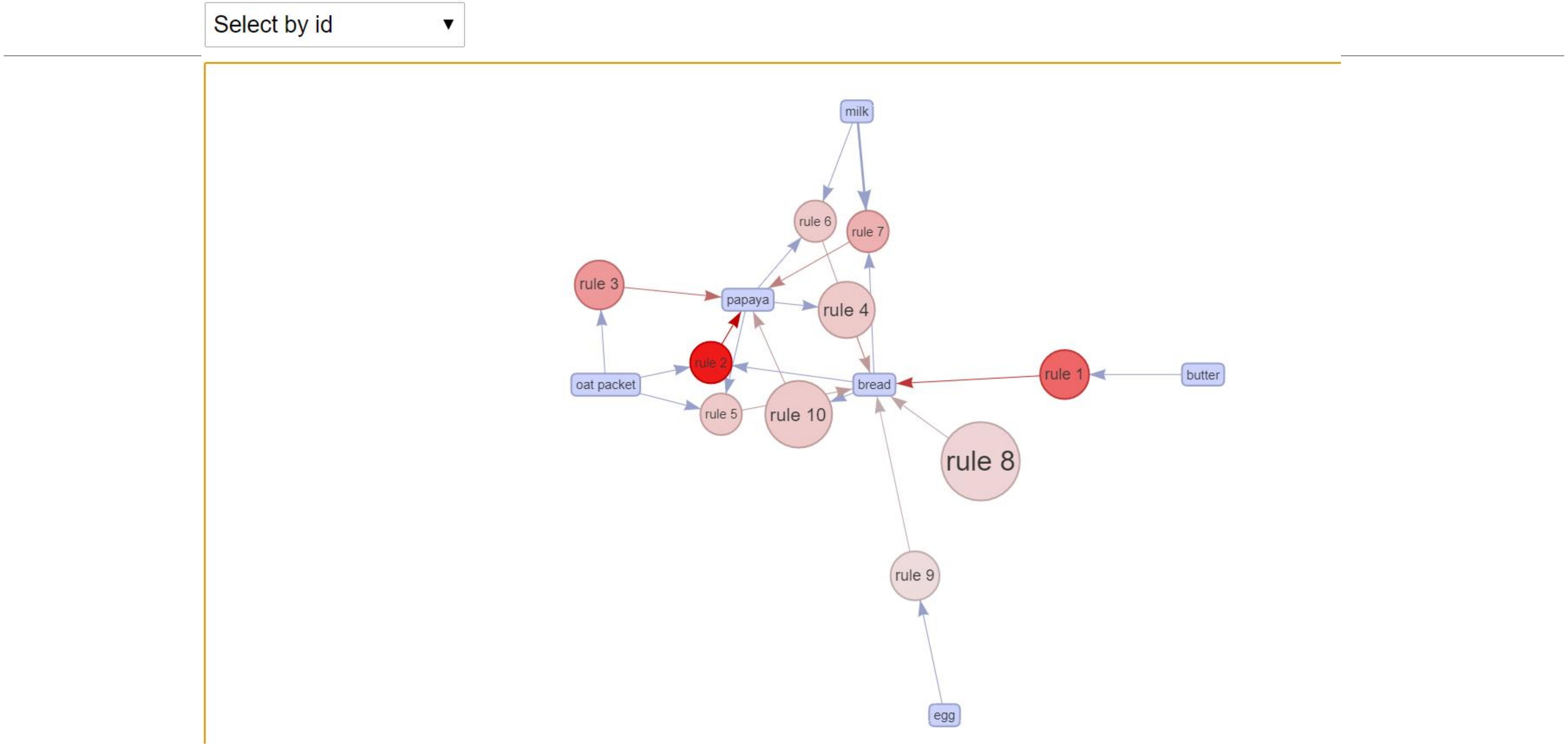
Node

Node

Small background on graphs…



```
top10rules = head(rules, n = 10, by = "confidence")
plot(top10rules, method = "graph",  engine = "htmlwidget")
```



Select by id ▾
milk
rule 6
rule 7
rule 3
papaya
rule 4
rule 2
oat packet
bread
rule 1
butter
rule 5
rule 10
rule 8
rule 9
egg