

Stat 5421 Lecture Notes: Graphical Models

Charles J. Geyer

July 27, 2023

Contents

1 License	1
2 R	1
3 Introduction	2
4 Undirected Graphs	2
5 Undirected Graphs and Probability	3
6 R Package glmbb	4
7 Directed Graphs	11
8 Directed Acyclic Graphs and Probability	12
9 Directed Acyclic Graphs and Causality	13
10 Drawing Graphs	14
11 Drawing the Graph for a Formula	17
Bibliography	26

1 License

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (<http://creativecommons.org/licenses/by-sa/4.0/>).

2 R

- The version of R used to make this document is 4.3.1.
- The version of the `rmarkdown` package used to make this document is 2.23.
- The version of the `glmbb` package used to make this document is 0.5.1.
- The version of the `CatDataAnalysis` package used to make this document is 0.1.5.
- The version of the `network` package used to make this document is 1.18.1.
- The version of the `utils` package used to make this document is 4.3.1.

```
library(glmbb)
library(CatDataAnalysis)
library(network)

##
## 'network' 1.18.1 (2023-01-24), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information

library(utils)
```

3 Introduction

Graphical models come in many kinds. There are graphical models where all the variables are categorical (Lauritzen (1996); Chapter 4). There are graphical models where the variables are jointly multivariate normal (Lauritzen (1996); Chapter 5). There are graphical models where some of the variables are categorical and the rest are conditionally jointly multivariate normal given the categorical ones (Lauritzen (1996); Chapter 6).

Since this is a course in categorical data, we will only be interested in graphical models having only categorical variables.

4 Undirected Graphs

An *undirected graph* consists of a set whose elements are called *nodes* and a set whose elements are called *edges* and which are unordered pairs of nodes. We write $G = (N, E)$ where G is the graph, N is the node set, and E is the edge set.

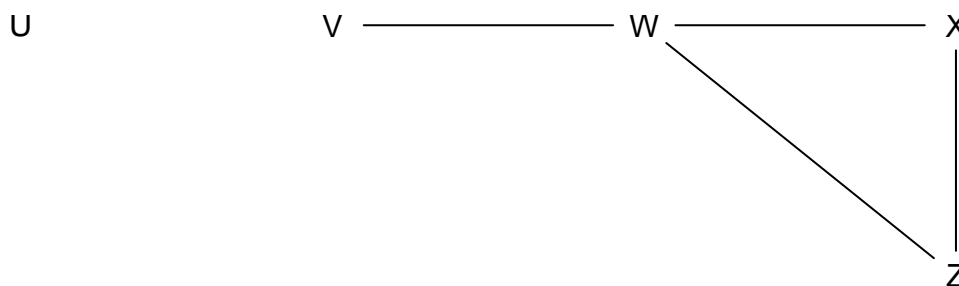


Figure 1: An Undirected Graph

Figure 1 is an example. The nodes are U , V , W , X , and Z and the edges are the line segments.

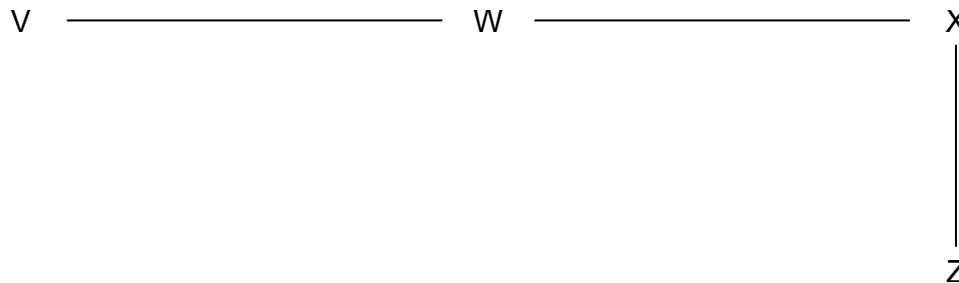
A graph is called *simple* if it has no repeats of its edges, which our insistence that the edges are a set (which cannot have repeats) rather than a multiset (which can) already rules out, and if it has no loops (edges of the form (n, n)). We are only interested in simple graphs. The graph in Figure 1 is simple.

A graph (N_1, E_1) is a *subgraph* of (N_2, E_2) if $N_1 \subset N_2$ and $E_1 \subset E_2$, where \subset means subset (we do not use \subseteq to mean subset, so $A \subset B$ includes the case $A = B$).

A graph is *complete* if every pair of nodes is connected by an edge (Lauritzen (1996); Section 2.1.1).

A *clique* in a graph is the node set of a maximal complete subgraph, one that is not a subgraph of another complete subgraph (Lauritzen (1996); Section 2.1.1). The cliques in Figure 1 are $\{U\}$, $\{V, W\}$, and $\{W, X, Z\}$.

A *path* in an undirected graph is a sequence of edges (n_i, n_{i+1}) , $i = 1, \dots, k$. For example,



is a path in the graph in Figure 1. We can say that this path goes from V to Z or from Z to V . Since the edges are undirected, there is no implied direction.

If A , B , and C are sets of nodes of a graph, then we say that C *separates* A and B if every path from a node in A to a node in B goes through C (has an edge, one of whose nodes is in C). In the graph in Figure 1, the node set $\{W\}$ separates the node sets $\{U, V\}$ and $\{X, Z\}$.

5 Undirected Graphs and Probability

A *log-linear model* for a contingency table is any model having one of the “sampling schemes” (Poisson, multinomial, product multinomial) described in the [Chapter 1 notes](#) and in much more detail in the [notes on sampling schemes](#).

Such a model is *hierarchical* if for every interaction term in the model all main effects and lower-order interactions involving variables in that term are also in the model.

The *interaction graph* for a hierarchical model is an undirected graph that has nodes that are the variables in the model and an edge for every pair of variables that appear in the same interaction term (Lauritzen (1996); Section 4.3.3).

A hierarchical model is *graphical* if its terms correspond to the cliques of its interaction graph (Lauritzen (1996); Section 4.3.3). Repeating what we said in the preceding section, the cliques the graph in Figure 1

are $\{U\}$, $\{V, W\}$, and $\{W, X, Z\}$. So the graphical model having this graph has formula

$$\sim U + V * W + W * X * Z.$$

Not every hierarchical model is graphical. For example, the hierarchical model with formula

$$\sim U + V * W + W * X + X * Z + Z * W$$

has the same interaction graph. But it is not graphical because it does not have the term $W * X * Z$.

A *Markov property* for a graph having random variables for nodes tells us something about conditional independence or factorization into marginals and conditionals. The relevant Markov property for log-linear models is the following, which is stated in Section 4.3.3 in Lauritzen (1996) and said to follow from Theorems 3.7 and 3.9 in Lauritzen (1996).

Theorem 5.1. *If C separates A and B in the interaction graph, then the variables in A are conditionally independent of those in B given those in C .*

The conclusion of the theorem is often written

$$A \perp\!\!\!\perp B \mid C.$$

It means, of course, that the conditional distribution factorizes

$$f(y_{A \cup B} \mid y_C) = f(y_A \mid y_C) f(y_B \mid y_C)$$

where we have the usual abuse of notation that the three f 's denote three different functions, the conditional PMF's of the indicated sets of variables, and where we are now using the notation y_S for subvectors of the response vector y of the probability model (described in the [section on subvectors in the notes on sampling](#)).

Repeating what we said in the preceding section, in the graph in Figure 1, $\{W\}$ separates $\{U, V\}$ and $\{X, Z\}$. So if Figure 1 is the interaction graph of a graphical model, we have

$$U, V \perp\!\!\!\perp X, Z \mid W.$$

Perhaps not quite so obvious, in the same graph the empty set separates U from all the other nodes. Since there are no paths from U to any other node. Every path from U to another node (there are none of them) goes through the empty set. Thus we can also say

$$U \perp\!\!\!\perp V, W, X, Z \mid \emptyset.$$

But, since conditioning on an empty set of variables is the same as not conditioning. We can also say

$$U \perp\!\!\!\perp V, W, X, Z$$

meaning U is (unconditionally) independent of the rest of the variables.

6 R Package glmbb

R package `glmbb` has some functions for dealing with graphical models.

R function `isGraphical` says whether a formula is for a graphical model.

```
isGraphical(~ U + V * W + W * X * Z)
```

```
## [1] TRUE
```

```
isGraphical(~ U + V * W + W * X + X * Z + Z * W)
```

```
## [1] FALSE
```

```
isHierarchical(~ U + V * W + W * X + X * Z + Z * W)
```

```
## [1] TRUE
```

These opinions of the computer agree with our analysis above.

There is also an option `graphical = TRUE` to R function `glm` to tell it to only consider graphical models.

```
data(exercise_6.28)
```

```
sapply(exercise_6.28, class)
```

```
## Occupational_aspirations      Socioeconomic_status      IQ      Residence
##                "factor"                "factor"                "factor"                "factor"
```

```
lapply(exercise_6.28, levels)
```

```
## $Occupational_aspirations
```

```
## [1] "High" "Low"
```

```
##
```

```
## $Socioeconomic_status
```

```
## [1] "High" "Low"
```

```
##
```

```
## $IQ
```

```
## [1] "High" "Low"
```

```
##
```

```
## $Residence
```

```
## [1] "Large urban" "Rural"      "Small urban"
```

```
##
```

```
## $Gender
```

```
## [1] "Female" "Male"
```

```
##
```

```
## $counts
```

```
## NULL
```

```
# all hierarchical models
```

```
gout <- glm(b ~ (Occupational_aspirations + Socioeconomic_status + IQ + Residence + Gender)^5, data = exercise_6.28)
```

```
summary(gout)
```

```
##
```

```
## Results of search for hierarchical models with lowest AIC.
```

```
## Search was for all models with AIC no larger than min(AIC) + 10
```

```
## These are shown below.
```

```
##
```

```
##   criterion  weight      formula
```

```
##   356.1      0.0831068 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
```

```
##   356.8      0.0602980 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
```

```
##   357.0      0.0536192 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
```

```
##   358.0      0.0323319 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
```

```
##   358.1      0.0313371 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
```

```
##   358.5      0.0249027 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
```

```
##   358.7      0.0231871 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Gender +
```

```
##   358.7      0.0230836 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
```

```
##   358.7      0.0224651 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
```

```
##   358.9      0.0202700 counts ~ Socioeconomic_status*IQ + IQ*Gender + Occupational_aspirations*Soci
```

```
##   359.1      0.0187457 counts ~ Occupational_aspirations*IQ + IQ*Residence + Occupational_aspiration
```

```
##   359.5      0.0153778 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
```

##	359.7	0.0136073	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomic
##	359.9	0.0126745	counts	~ Socioeconomic_status*IQ + IQ*Residence + Occupational_aspirations*S
##	359.9	0.0125381	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	359.9	0.0125226	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
##	360.0	0.0122413	counts	~ IQ*Residence + IQ*Gender + Occupational_aspirations*Socioeconomic_s
##	360.1	0.0111573	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
##	360.4	0.0099215	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
##	360.4	0.0097442	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
##	360.4	0.0095619	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
##	360.5	0.0095029	counts	~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
##	360.5	0.0094327	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*Residen
##	360.5	0.0090970	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
##	360.6	0.0090476	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	360.6	0.0089172	counts	~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
##	360.7	0.0084677	counts	~ Occupational_aspirations*IQ + IQ*Gender + Occupational_aspirations*
##	360.8	0.0081829	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
##	360.8	0.0080894	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*IQ*Residence + O
##	360.9	0.0076303	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	361.0	0.0073118	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
##	361.0	0.0071361	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	361.3	0.0063734	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	361.3	0.0063189	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
##	361.3	0.0062078	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
##	361.4	0.0059826	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	361.4	0.0057985	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	361.5	0.0055555	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	361.5	0.0055362	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Residence*G
##	361.7	0.0051214	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	361.7	0.0051202	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
##	361.8	0.0049367	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	361.8	0.0049230	counts	~ Socioeconomic_status*IQ + Residence*Gender + Occupational_aspiration
##	361.8	0.0048778	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	361.9	0.0047338	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	361.9	0.0046242	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
##	361.9	0.0046220	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	361.9	0.0046079	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.0	0.0044755	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.0	0.0044448	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
##	362.1	0.0042905	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Gender +
##	362.1	0.0042713	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.1	0.0041569	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
##	362.1	0.0041120	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
##	362.3	0.0038131	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.3	0.0037570	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.3	0.0037529	counts	~ Socioeconomic_status*IQ + IQ*Gender + Occupational_aspirations*Soci
##	362.3	0.0037368	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.4	0.0036142	counts	~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*Residence
##	362.4	0.0035616	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	362.4	0.0035294	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
##	362.5	0.0034982	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Gender +
##	362.5	0.0034826	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	362.5	0.0034770	counts	~ Occupational_aspirations*IQ + IQ*Residence + Occupational_aspiration
##	362.5	0.0034442	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
##	362.5	0.0034282	counts	~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup

##	362.5	0.0034229	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	362.5	0.0033892	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ*Residence + O
##	362.5	0.0033535	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
##	362.7	0.0030671	counts	~ Socioeconomic_status*IQ + IQ*Gender + Occupational_aspirations*IQ*R
##	362.7	0.0030389	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
##	362.8	0.0029820	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_sta
##	362.8	0.0029685	counts	~ IQ*Residence + Residence*Gender + Occupational_aspirations*Socioecon
##	362.8	0.0028873	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	362.9	0.0028482	counts	~ Occupational_aspirations*IQ + IQ*Residence + Socioeconomic_status*I
##	362.9	0.0028304	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	362.9	0.0027478	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.0	0.0027325	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.1	0.0025798	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.1	0.0025296	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.1	0.0025179	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
##	363.1	0.0024795	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	363.2	0.0024489	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	363.2	0.0024387	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.2	0.0024117	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_sta
##	363.2	0.0023744	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*Reside
##	363.3	0.0023455	counts	~ Socioeconomic_status*IQ + IQ*Residence + Occupational_aspirations*S
##	363.3	0.0023235	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.3	0.0022864	counts	~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*IQ
##	363.3	0.0022768	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
##	363.3	0.0022651	counts	~ IQ*Residence + IQ*Gender + Occupational_aspirations*Socioeconomic_s
##	363.3	0.0022476	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.5	0.0021289	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Gender +
##	363.5	0.0021194	counts	~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*IQ
##	363.5	0.0021085	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.5	0.0020626	counts	~ Occupational_aspirations*IQ + Residence*Gender + Occupational_aspir
##	363.5	0.0020529	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ*Gender + Occup
##	363.6	0.0020040	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.7	0.0019127	counts	~ Socioeconomic_status*IQ + IQ*Residence + Occupational_aspirations*I
##	363.7	0.0019098	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.7	0.0018554	counts	~ Socioeconomic_status*IQ + IQ*Gender + Residence*Gender + Occupation
##	363.7	0.0018494	counts	~ IQ*Residence + IQ*Gender + Occupational_aspirations*Socioeconomic_s
##	363.7	0.0018419	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.7	0.0018389	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.8	0.0018187	counts	~ Occupational_aspirations*IQ*Residence + Socioeconomic_status*IQ*Gen
##	363.8	0.0018030	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_sta
##	363.8	0.0017981	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
##	363.8	0.0017782	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Gender +
##	363.8	0.0017703	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.8	0.0017693	counts	~ Occupational_aspirations*Socioeconomic_status*Residence + Occupation
##	363.8	0.0017597	counts	~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
##	363.8	0.0017516	counts	~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*Reside
##	363.8	0.0017475	counts	~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
##	363.9	0.0017228	counts	~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
##	363.9	0.0017079	counts	~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
##	363.9	0.0016937	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	363.9	0.0016847	counts	~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_sta
##	363.9	0.0016829	counts	~ Occupational_aspirations*IQ + IQ*Residence + Residence*Gender + Occ
##	363.9	0.0016741	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
##	364.0	0.0016547	counts	~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp

```

## 364.0 0.0016500 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occu
## 364.0 0.0016226 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 364.0 0.0015941 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.1 0.0015722 counts ~ Socioeconomic_status*IQ + IQ*Gender + Occupational_aspirations*Soci
## 364.1 0.0015628 counts ~ Occupational_aspirations*IQ + IQ*Gender + Occupational_aspirations*
## 364.1 0.0015144 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 364.2 0.0014701 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*IQ*Gender + Occu
## 364.2 0.0014591 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.2 0.0014435 counts ~ Occupational_aspirations*IQ + IQ*Residence + Occupational_aspiration
## 364.2 0.0014426 counts ~ Occupational_aspirations*IQ*Residence + Socioeconomic_status*IQ*Res
## 364.2 0.0014385 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occu
## 364.2 0.0014328 counts ~ IQ*Residence + Occupational_aspirations*IQ*Gender + Socioeconomic_s
## 364.3 0.0013850 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.3 0.0013817 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 364.3 0.0013807 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.3 0.0013650 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Socioeconomic_st
## 364.4 0.0013529 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.4 0.0013453 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occu
## 364.4 0.0013233 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
## 364.4 0.0013037 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
## 364.5 0.0012896 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Gender +
## 364.5 0.0012838 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.5 0.0012765 counts ~ Occupational_aspirations*IQ + IQ*Gender + Socioeconomic_status*IQ*R
## 364.5 0.0012719 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.5 0.0012504 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
## 364.5 0.0012494 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 364.5 0.0012493 counts ~ Occupational_aspirations*IQ + Residence*Gender + Occupational_aspir
## 364.5 0.0012311 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*IQ*Residence + O
## 364.6 0.0012205 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.6 0.0012203 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.6 0.0012200 counts ~ Socioeconomic_status*IQ + IQ*Residence + Residence*Gender + Occupat
## 364.6 0.0011950 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.6 0.0011899 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 364.6 0.0011724 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.6 0.0011707 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 364.7 0.0011363 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.7 0.0011318 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.7 0.0011281 counts ~ IQ*Residence + IQ*Gender + Residence*Gender + Occupational_aspirati
## 364.7 0.0011228 counts ~ Socioeconomic_status*IQ + IQ*Gender + Occupational_aspirations*Soci
## 364.7 0.0011136 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.8 0.0011125 counts ~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*Re
## 364.8 0.0011121 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 364.8 0.0011031 counts ~ Occupational_aspirations*IQ*Gender + Socioeconomic_status*IQ*Gender
## 364.8 0.0010841 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + S
## 364.8 0.0010641 counts ~ Occupational_aspirations*IQ + IQ*Residence + Occupational_aspiration
## 364.9 0.0010435 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 364.9 0.0010304 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 364.9 0.0010297 counts ~ Socioeconomic_status*IQ + IQ*Residence + Occupational_aspirations*S
## 365.0 0.0010059 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupational
## 365.0 0.0010036 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.0 0.0009984 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.0 0.0009927 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + IQ*Residence
## 365.1 0.0009578 counts ~ IQ*Residence + IQ*Gender + Occupational_aspirations*Socioeconomic_s
## 365.1 0.0009541 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio

```



```

## 365.1 0.0009477 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.1 0.0009459 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Occupatio
## 365.1 0.0009450 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 365.1 0.0009449 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.1 0.0009202 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 365.1 0.0009136 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + U
## 365.2 0.0009109 counts ~ IQ*Residence + Residence*Gender + Occupational_aspirations*Socioecon
## 365.2 0.0008983 counts ~ Occupational_aspirations*IQ*Residence + Occupational_aspirations*IQ
## 365.2 0.0008947 counts ~ Socioeconomic_status*IQ + Residence*Gender + Occupational_aspiration
## 365.2 0.0008779 counts ~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*Re
## 365.2 0.0008689 counts ~ IQ*Gender + Residence*Gender + Occupational_aspirations*Socioeconom
## 365.2 0.0008672 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.3 0.0008613 counts ~ Occupational_aspirations*IQ + IQ*Residence + Occupational_aspiration
## 365.3 0.0008411 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 365.3 0.0008307 counts ~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*IQ
## 365.3 0.0008281 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.4 0.0008187 counts ~ IQ*Gender + Residence*Gender + Occupational_aspirations*Socioeconom
## 365.4 0.0008066 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 365.4 0.0008064 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.4 0.0007974 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.4 0.0007916 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 365.4 0.0007894 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.4 0.0007875 counts ~ Socioeconomic_status*IQ + Residence*Gender + Occupational_aspiration
## 365.5 0.0007796 counts ~ Occupational_aspirations*IQ + IQ*Gender + Residence*Gender + Occupa
## 365.5 0.0007785 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ*Residence + S
## 365.5 0.0007771 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*Residen
## 365.5 0.0007727 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Socioeconomic_st
## 365.5 0.0007568 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 365.5 0.0007473 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 365.6 0.0007451 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
## 365.6 0.0007377 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
## 365.6 0.0007333 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.6 0.0007324 counts ~ Socioeconomic_status*IQ + IQ*Residence + Occupational_aspirations*S
## 365.6 0.0007294 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ*Gender + IQ*R
## 365.7 0.0007058 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.7 0.0007039 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.7 0.0006985 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 365.7 0.0006939 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.7 0.0006914 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.7 0.0006853 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.7 0.0006839 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
## 365.7 0.0006828 counts ~ IQ*Residence + IQ*Gender + Occupational_aspirations*Socioeconomic_s
## 365.8 0.0006752 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.8 0.0006713 counts ~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*Re
## 365.8 0.0006678 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*Residence
## 365.8 0.0006661 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 365.8 0.0006648 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 365.8 0.0006633 counts ~ Socioeconomic_status*IQ + Occupational_aspirations*Socioeconomic_st
## 365.8 0.0006612 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*IQ + O
## 365.8 0.0006611 counts ~ Occupational_aspirations*IQ + IQ*Gender + Occupational_aspirations*
## 365.8 0.0006604 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.8 0.0006583 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.8 0.0006446 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 365.9 0.0006410 counts ~ IQ*Residence + Residence*Gender + Occupational_aspirations*Socioecon

```

```

## 365.9 0.0006343 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.9 0.0006325 counts ~ IQ*Gender + Occupational_aspirations*Socioeconomic_status*IQ + Occup
## 365.9 0.0006315 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.9 0.0006314 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.9 0.0006309 counts ~ Occupational_aspirations*Socioeconomic_status*Gender + Occupational
## 365.9 0.0006255 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 365.9 0.0006254 counts ~ Occupational_aspirations*IQ + Socioeconomic_status*IQ + Socioeconomi
## 365.9 0.0006128 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 366.0 0.0006107 counts ~ Residence*Gender + Occupational_aspirations*Socioeconomic_status*Re
## 366.0 0.0006000 counts ~ Occupational_aspirations*IQ + Occupational_aspirations*Socioeconomi
## 366.0 0.0005918 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 366.0 0.0005886 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
## 366.1 0.0005750 counts ~ IQ*Residence + Occupational_aspirations*Socioeconomic_status*Resider
## 366.1 0.0005741 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 366.1 0.0005638 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_asp
## 366.1 0.0005607 counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio

```

now just the graphical model

```

gout.gr <- glmbb(counts ~ (Occupational_aspirations + Socioeconomic_status +
  IQ + Residence + Gender)^5, data = exercise_6.28, graphical = TRUE)
summary(gout.gr)

```

```

##
## Results of search for graphical models with lowest AIC.
## Search was for all models with AIC no larger than min(AIC) + 10
## These are shown below.
##

```

##	criterion	weight	formula
##	362.5	0.554860	counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspi
##	363.8	0.282047	counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspi
##	367.0	0.057695	counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspi
##	367.5	0.044201	counts ~ Occupational_aspirations*Socioeconomic_status*IQ*Gender + Occupatio
##	368.9	0.022468	counts ~ Occupational_aspirations*Socioeconomic_status*Residence + Occupatio
##	369.7	0.015164	counts ~ Occupational_aspirations*Socioeconomic_status*IQ*Residence + Occupat
##	371.0	0.007967	counts ~ Occupational_aspirations*Gender + Occupational_aspirations*Socioecon
##	371.0	0.007890	counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspi
##	371.0	0.007708	counts ~ Occupational_aspirations*Socioeconomic_status*Gender + Occupational_

More hierarchical than graphical

```

results.hier <- summary(gout)$results
results.graph <- summary(gout.gr)$results
nrow(results.hier)

```

```
## [1] 244
```

```
nrow(results.graph)
```

```
## [1] 9
```

And

```
min(which(cumsum(results.hier$weight) >= 0.9))
```

```
## [1] 133
```

```
min(which(cumsum(results.graph$weight) >= 0.9))
```

```
## [1] 4
```

133 models account for 90% of the weight among all hierarchical models less than the cutoff. 4 models account for 90% of the weight among all graphical models less than the cutoff.

We will try to interpret some of these graphs [after we learn how to draw graphs](#).

7 Directed Graphs

A *directed graph* consists of a set whose elements are called *nodes* and a set whose elements are called *edges* and which are ordered pairs of nodes. When drawing a directed graph, the edges are drawn as arrows to indicate the direction. Figure 2 shows a directed graph.

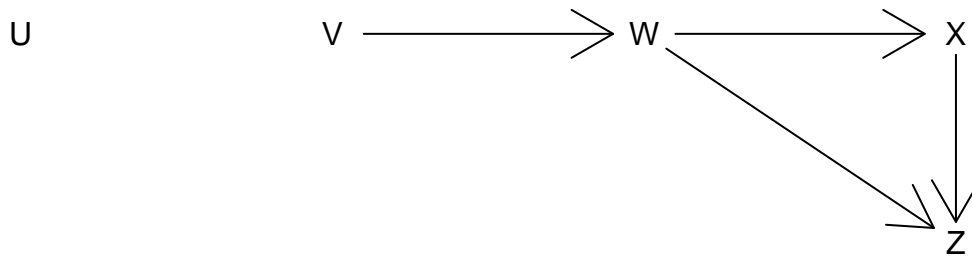
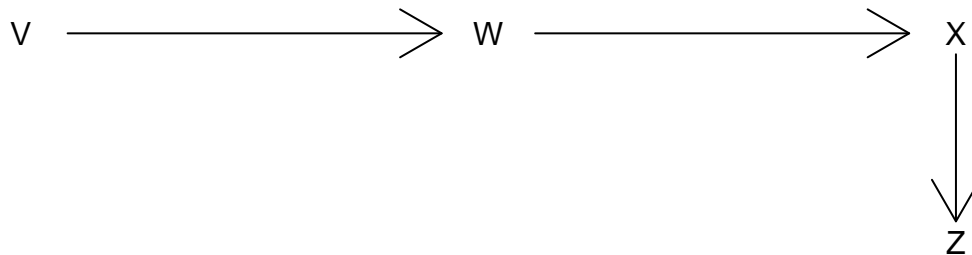


Figure 2: A Directed Acyclic Graph

A *path* in a directed graph is a sequence of edges $(n_i, n_{i+1}), i = 1, \dots, k$. Since the edges are directed, there is an implied direction. We say the path goes from n_1 to n_{k+1} . For example,



is a path from V to Z in the graph in Figure 2. (The path must go in the same direction as the arrows.)

A directed graph is called *acyclic* if it has no cycles, which are paths that return to their starting point. The graph in Figure 2 is acyclic. Directed acyclic graphs are so important that they have a well-known TLA (three-letter acronym) DAG.

In a DAG the *parents* of a node u are the nodes from which edges go to u . The set of parents of u is denoted $\text{pa}(u)$. In Figure 2 we have

$$\begin{aligned}
 \text{pa}(U) &= \emptyset \\
 \text{pa}(V) &= \emptyset \\
 \text{pa}(W) &= \{V\} \\
 \text{pa}(X) &= \{W\} \\
 \text{pa}(Z) &= \{W, X\}
 \end{aligned}$$

8 Directed Acyclic Graphs and Probability

Every DAG that has nodes that are the variables in a statistical model is associated with a factorization of the joint distribution for that model into a product of marginal and conditional distributions

$$f(y) = \prod_{i \in N} f(y_i \mid y_{\text{pa}(i)})$$

(Lauritzen (1996); Section 4.5.1). Using the parent sets we found in the preceding section, the graph in Figure 2 has the factorization

$$f(u, v, w, x, z) = f(z \mid w, x)f(x \mid w)f(w \mid v)f(v)f(u). \tag{1}$$

This agrees with the separation properties of the corresponding undirected graph discussed in Section 5 above. A general factorization, valid for any probability model would be

$$\begin{aligned}
 f(u, v, w, x, z) &= f(z \mid u, v, w, x)f(u, v, w, x) \\
 &= f(z \mid u, v, w, x)f(x \mid u, v, w)f(u, v, w) \\
 &= f(z \mid u, v, w, x)f(x \mid u, v, w)f(w \mid u, v)f(u, v) \\
 &= f(z \mid u, v, w, x)f(x \mid u, v, w)f(w \mid u, v)f(v \mid u)f(u)
 \end{aligned}$$

Comparing this general factorization with (1) we see that we have

$$\begin{aligned} f(z | u, v, w, x) &= f(z | w, x) \\ f(x | u, v, w) &= f(x | w) \\ f(w | u, v) &= f(w | v) \\ f(v | u) &= f(v) \end{aligned}$$

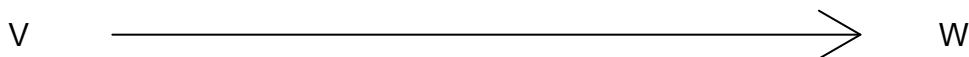
which imply

$$\begin{aligned} Z \perp\!\!\!\perp U, V | W, X \\ X \perp\!\!\!\perp U, V | W \\ W \perp\!\!\!\perp U | V \\ V \perp\!\!\!\perp U \end{aligned}$$

all of which are Markov properties that can be read off the undirected graph in Figure 1.

9 Directed Acyclic Graphs and Causality

DAG's are also widely used to indicate causal relationships (Pearl, 2009; Spirtes *et al.*, 2000). The idea is that a directed edge



indicates that changes in V cause changes in W .

Authorities on causal inference (Pearl, 2009; Spirtes *et al.*, 2000) understand that correlation is not causation. But they do insist that independence implies lack of causation.

If two variables are independent, then there cannot be a causal relationship between them (because that would induce dependence). If two variables are conditionally independent given a set S of other variables, then there cannot be a direct causal relationship. Any causal link must be indirect, the path from one to the other going through S .

Direction of causality cannot be inferred from independence, conditional or unconditional. If U and V are dependent, then changes in U may cause changes in V or changes in V may cause changes in U or changes in some other variable W may cause changes in both U and V .

There are two ways one can infer direction of causality. One is what is called “intervention” in the causal inference literature. It is the same thing that statisticians are talking about when they refer to controlled experiments. The other way is to simply assume that only one direction for edges makes sense (scientific sense, business sense, whatever).

In short, statistics can give you an undirected graph, but only controlled experiments or assumptions can give direction and causal interpretation of the edges.

10 Drawing Graphs

This about drawing graphs using R. By graphs we don't mean plots, but rather figures like Figures 1 and 2 above.

[TIMTODTWI](#). There is an [R task view on graphical models](#) that lists 8 CRAN packages and 2 Bioconductor packages for dealing with graphical models.

In addition to all of these packages one can just use base R graphics, which is how Figures 1 and 2 were done. But the packages are perhaps easier to use.

We will just illustrate using R package `network`. We will redraw the graphs that are Figures 1 and 2 above.

R function `network` in R package `network` makes objects of class `"network"` that can then be plotted. It wants the graph presented as an adjacency matrix which is a matrix whose row and column indices are nodes of the graph and the entries of the matrix are zero if there is no edge from one node to another and one otherwise. We can use the same matrix for both directed and undirected graphs. So we set up the matrix for Figure 2.

```
nodeset <- LETTERS["U" <= LETTERS & "Y" != LETTERS]
nodeset

## [1] "U" "V" "W" "X" "Z"

foo <- matrix(0, nrow = length(nodeset), ncol = length(nodeset))
rownames(foo) <- colnames(foo) <- nodeset
foo["V", "W"] <- foo["W", "X"] <- foo["W", "Z"] <- foo["X", "Z"] <- 1
foo

##   U V W X Z
## U 0 0 0 0 0
## V 0 0 1 0 0
## W 0 0 0 1 1
## X 0 0 0 0 1
## Z 0 0 0 0 0
```

Now we can plot it.

```
bar <- network(foo)
plot(bar, displaylabels = TRUE)
```

If we want an undirected graph, we have to remake the network object.

```
bar <- network(foo, directed = FALSE)
plot(bar, displaylabels = TRUE)
```

Every time you redo your plots, the graph nodes will be in different positions. But this doesn't matter. The nodes don't really have positions, only connections (edges).

On the other hand, one can just draw the graph in some drawing application and include it in the Rmarkdown

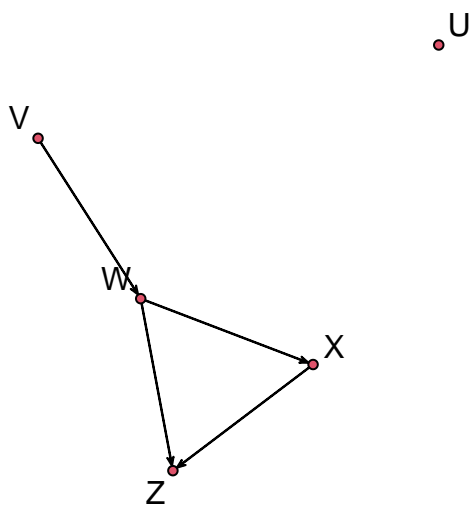


Figure 3: A Directed Graph

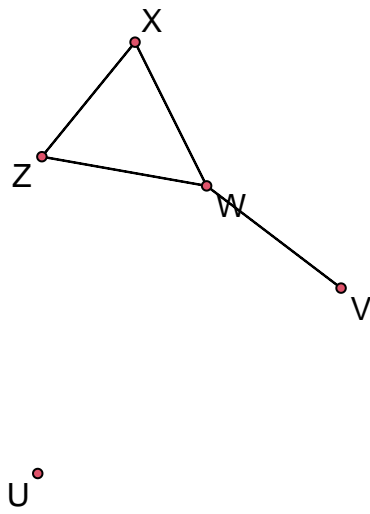
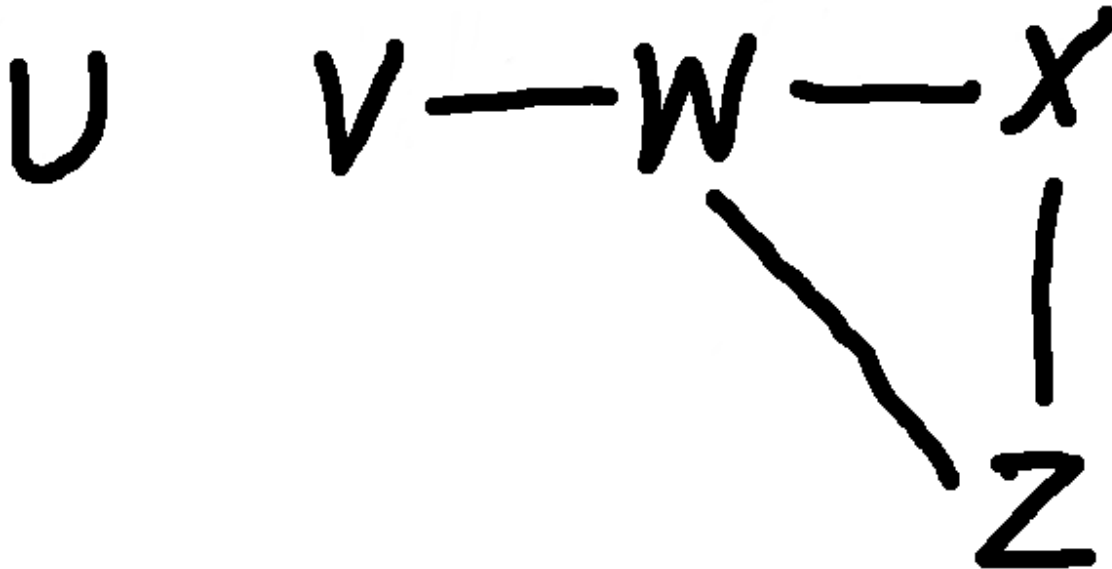


Figure 4: An Undirected Graph



11 Drawing the Graph for a Formula

```
idx <- min(which(cumsum(results.graph$weight) >= 0.9))
results.graph[1:idx, ]
```

```
## criterion weight
## 1 362.4692 0.55485965 counts ~ Occupational_aspirat
## 2 363.8225 0.28204722 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspi
## 3 366.9963 0.05769458 counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupa
## 4 367.5292 0.04420107 counts ~ Occupational_aspirations*So
```

```
foo <- results.graph$formula[1:idx]
```

So let's look at one of these formulas

```
foo[1]
```

```
## [1] "counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspirations*Socioecon
```

Now we are going to some real magic of R! R is not an ordinary computing language, in that it knows a lot about itself. Everything in R is an object, including R expressions. So R can take apart this formula. First we turn this character string into a formula, and then we call R function `terms` to interpret it.

```
foo[1]
```

```
## [1] "counts ~ Occupational_aspirations*Socioeconomic_status*IQ + Occupational_aspirations*Socioecon
```

```
ff <- as.formula(foo[1])
tt <- terms(ff)
names(attributes(tt))
```

```
## [1] "variables"      "factors"          "term.labels"     "order"           "intercept"
## [6] "response"        "class"            ".Environment"
```

Looks like we want `term.labels`

```
ll <- attr(tt, "term.labels")
ll
```

```
## [1] "Occupational_aspirations"
## [2] "Socioeconomic_status"
## [3] "IQ"
## [4] "Residence"
## [5] "Gender"
## [6] "Occupational_aspirations:Socioeconomic_status"
## [7] "Occupational_aspirations:IQ"
## [8] "Socioeconomic_status:IQ"
## [9] "Occupational_aspirations:Residence"
## [10] "Socioeconomic_status:Residence"
## [11] "Occupational_aspirations:Gender"
## [12] "Socioeconomic_status:Gender"
## [13] "Residence:Gender"
## [14] "Occupational_aspirations:Socioeconomic_status:IQ"
## [15] "Occupational_aspirations:Socioeconomic_status:Residence"
## [16] "Occupational_aspirations:Socioeconomic_status:Gender"
## [17] "Occupational_aspirations:Residence:Gender"
## [18] "Socioeconomic_status:Residence:Gender"
## [19] "Occupational_aspirations:Socioeconomic_status:Residence:Gender"
```

This is something of a mess because of hierarchicality. We really only need the highest order terms, but the others don't hurt. First use R function `strsplit` to get the variables

```
ss <- lapply(ll, strsplit, split = ":", fixed = TRUE)
head(ss)
```

```
## [[1]]
## [[1]][[1]]
## [1] "Occupational_aspirations"
##
##
## [[2]]
## [[2]][[1]]
## [1] "Socioeconomic_status"
##
##
## [[3]]
## [[3]][[1]]
## [1] "IQ"
##
##
## [[4]]
## [[4]][[1]]
## [1] "Residence"
##
##
## [[5]]
## [[5]][[1]]
## [1] "Gender"
```

```

##
##
## [[6]]
## [[6]][[1]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
ss <- lapply(ss, unlist)
ss

## [[1]]
## [1] "Occupational_aspirations"
##
## [[2]]
## [1] "Socioeconomic_status"
##
## [[3]]
## [1] "IQ"
##
## [[4]]
## [1] "Residence"
##
## [[5]]
## [1] "Gender"
##
## [[6]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
##
## [[7]]
## [1] "Occupational_aspirations" "IQ"
##
## [[8]]
## [1] "Socioeconomic_status" "IQ"
##
## [[9]]
## [1] "Occupational_aspirations" "Residence"
##
## [[10]]
## [1] "Socioeconomic_status" "Residence"
##
## [[11]]
## [1] "Occupational_aspirations" "Gender"
##
## [[12]]
## [1] "Socioeconomic_status" "Gender"
##
## [[13]]
## [1] "Residence" "Gender"
##
## [[14]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "IQ"
##
## [[15]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "Residence"

```

```
##
## [[16]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "Gender"
##
## [[17]]
## [1] "Occupational_aspirations" "Residence"
## [3] "Gender"
##
## [[18]]
## [1] "Socioeconomic_status" "Residence"          "Gender"
##
## [[19]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "Residence"              "Gender"
```

Now we want all pairs of components of the same interaction. There's an R function for that: `combn` in R package `utils`.

```
cc <- lapply(ss, combn, m = 2)
```

```
## Error in FUN(X[[i]], ...): n < m
```

But it doesn't like vector smaller than 2. So remove them and retry.

```
len.ss <- sapply(ss, length)
ss <- ss[len.ss >= 2]
ss
```

```
## [[1]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
##
## [[2]]
## [1] "Occupational_aspirations" "IQ"
##
## [[3]]
## [1] "Socioeconomic_status" "IQ"
##
## [[4]]
## [1] "Occupational_aspirations" "Residence"
##
## [[5]]
## [1] "Socioeconomic_status" "Residence"
##
## [[6]]
## [1] "Occupational_aspirations" "Gender"
##
## [[7]]
## [1] "Socioeconomic_status" "Gender"
##
## [[8]]
## [1] "Residence" "Gender"
##
## [[9]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "IQ"
##
```

```

## [[10]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "Residence"
##
## [[11]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "Gender"
##
## [[12]]
## [1] "Occupational_aspirations" "Residence"
## [3] "Gender"
##
## [[13]]
## [1] "Socioeconomic_status" "Residence"          "Gender"
##
## [[14]]
## [1] "Occupational_aspirations" "Socioeconomic_status"
## [3] "Residence"          "Gender"

```

```

cc <- lapply(ss, combn, m = 2)
cc

```

```

## [[1]]
##      [,1]
## [1,] "Occupational_aspirations"
## [2,] "Socioeconomic_status"
##
## [[2]]
##      [,1]
## [1,] "Occupational_aspirations"
## [2,] "IQ"
##
## [[3]]
##      [,1]
## [1,] "Socioeconomic_status"
## [2,] "IQ"
##
## [[4]]
##      [,1]
## [1,] "Occupational_aspirations"
## [2,] "Residence"
##
## [[5]]
##      [,1]
## [1,] "Socioeconomic_status"
## [2,] "Residence"
##
## [[6]]
##      [,1]
## [1,] "Occupational_aspirations"
## [2,] "Gender"
##
## [[7]]
##      [,1]
## [1,] "Socioeconomic_status"

```

```

## [2,] "Gender"
##
## [[8]]
##      [,1]
## [1,] "Residence"
## [2,] "Gender"
##
## [[9]]
##      [,1]                [,2]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "IQ"
##      [,3]
## [1,] "Socioeconomic_status"
## [2,] "IQ"
##
## [[10]]
##      [,1]                [,2]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "Residence"
##      [,3]
## [1,] "Socioeconomic_status"
## [2,] "Residence"
##
## [[11]]
##      [,1]                [,2]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "Gender"
##      [,3]
## [1,] "Socioeconomic_status"
## [2,] "Gender"
##
## [[12]]
##      [,1]                [,2]                [,3]
## [1,] "Occupational_aspirations" "Occupational_aspirations" "Residence"
## [2,] "Residence"                "Gender"                "Gender"
##
## [[13]]
##      [,1]                [,2]                [,3]
## [1,] "Socioeconomic_status" "Socioeconomic_status" "Residence"
## [2,] "Residence"          "Gender"                "Gender"
##
## [[14]]
##      [,1]                [,2]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "Residence"
##      [,3]                [,4]                [,5]
## [1,] "Occupational_aspirations" "Socioeconomic_status" "Socioeconomic_status"
## [2,] "Gender"                "Residence"          "Gender"
##      [,6]
## [1,] "Residence"
## [2,] "Gender"

```

Getting there. Now put this whole thing in one big two-row matrix,

```

rr <- Reduce(cbind, cc, init = NULL)
rr

##      [,1]                [,2]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "IQ"
##      [,3]                [,4]                [,5]
## [1,] "Socioeconomic_status" "Occupational_aspirations" "Socioeconomic_status"
## [2,] "IQ"                  "Residence"                "Residence"
##      [,6]                [,7]                [,8]
## [1,] "Occupational_aspirations" "Socioeconomic_status" "Residence"
## [2,] "Gender"              "Gender"                  "Gender"
##      [,9]                [,10]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "IQ"
##      [,11]               [,12]
## [1,] "Socioeconomic_status" "Occupational_aspirations"
## [2,] "IQ"                  "Socioeconomic_status"
##      [,13]               [,14]
## [1,] "Occupational_aspirations" "Socioeconomic_status"
## [2,] "Residence"            "Residence"
##      [,15]               [,16]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Socioeconomic_status"      "Gender"
##      [,17]               [,18]
## [1,] "Socioeconomic_status" "Occupational_aspirations"
## [2,] "Gender"              "Residence"
##      [,19]               [,20]                [,21]
## [1,] "Occupational_aspirations" "Residence" "Socioeconomic_status"
## [2,] "Gender"              "Gender"      "Residence"
##      [,22]               [,23]                [,24]
## [1,] "Socioeconomic_status" "Residence" "Occupational_aspirations"
## [2,] "Gender"              "Gender"      "Socioeconomic_status"
##      [,25]               [,26]
## [1,] "Occupational_aspirations" "Occupational_aspirations"
## [2,] "Residence"            "Gender"
##      [,27]               [,28]                [,29]
## [1,] "Socioeconomic_status" "Socioeconomic_status" "Residence"
## [2,] "Residence"          "Gender"          "Gender"

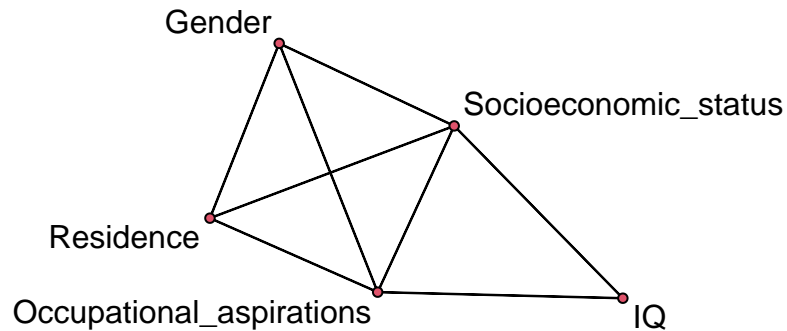
```

Now there is no honest way to do the last step without a loop AFAICS.

```

nn <- sort(unique(as.vector(rr)))
gr <- matrix(0, length(nn), length(nn))
rownames(gr) <- colnames(gr) <- nn
for (i in seq(1:ncol(rr))) gr[rr[1, i], rr[2, i]] <- 1
# symmetrize
gr <- gr + t(gr)
bar <- network(gr, directed = FALSE)
plot(bar, displaylabels = TRUE)

```



That's pretty easy to interpret. IQ is clearly separated from Residence and Gender by Occupational_aspirations and Socioeconomic_status. Using single letters for the variable names

$$I \perp\!\!\!\perp R, G \mid O, S$$

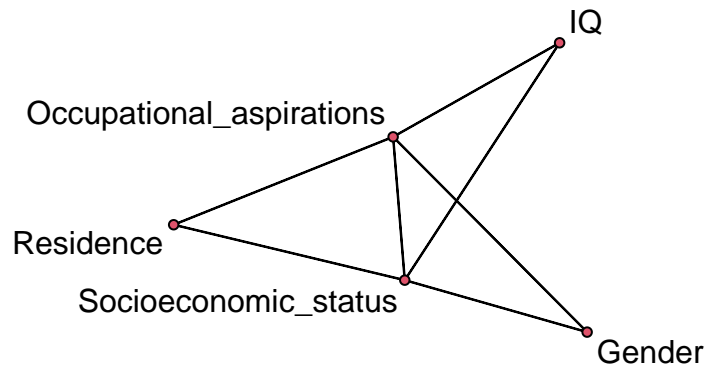
But that big clique of size four, nothing separates anything in it from anything else. That's the way cliques work.

Try it again. This time we systematize our calculations as a function.

```
formula.to.graph <- function(f) {
  stopifnot(inherits(f, "formula"))
  tt <- terms(f)
  ll <- attr(tt, "term.labels")
  ss <- lapply(ll, strsplit, split = ":", fixed = TRUE)
  ss <- lapply(ss, unlist)
  len.ss <- sapply(ss, length)
  ss <- ss[len.ss >= 2]
  cc <- lapply(ss, combn, m = 2)
  rr <- Reduce(cbind, cc, init = NULL)
  nn <- sort(unique(as.vector(rr)))
  gr <- matrix(0, length(nn), length(nn))
  rownames(gr) <- colnames(gr) <- nn
  for (i in seq(1:ncol(rr))) gr[rr[1, i], rr[2, i]] <- 1
  gr <- gr + t(gr)
  network(gr, directed = FALSE)
}
```

And give it a try-out.


```
plot(formula.to.graph(as.formula(foo[[2]])), displaylabels = TRUE)
```



This is a little more interesting. Now we have IQ, Residence, and Gender all conditionally independent given the other two variables. In math

$$\begin{aligned}
 I &\perp\!\!\!\perp R, G \mid O, S \\
 R &\perp\!\!\!\perp I, G \mid O, S \\
 G &\perp\!\!\!\perp I, R \mid O, S
 \end{aligned}$$

This is a bit annoying the $\perp\!\!\!\perp$ symbol isn't powerful enough to say what we want. Independence is not a pairwise property. So when three variables are conditionally independent given some others, it takes more than one use of symbols to get that.

Let's look at factorizations

$$\begin{aligned}
 f(i, r, g, o, s) &= f(i \mid r, g, o, s) f(r, g, o, s) \\
 &= f(i \mid o, s) f(r, g, o, s) \\
 &= f(i \mid o, s) f(r \mid g, o, s) f(g, o, s) \\
 &= f(i \mid o, s) f(r \mid o, s) f(g, o, s) \\
 &= f(i \mid o, s) f(r \mid o, s) f(g \mid o, s) f(o, s)
 \end{aligned}$$

where the equalities on even numbered lines are just joint equals conditional times marginal and the equalities on odd numbered lines are the conditional independence properties we read off the graph. Another way to write this is

$$f(i, r, g \mid o, s) = f(i \mid o, s) f(r \mid o, s) f(g \mid o, s)$$

which is what we said in words right at the beginning of our analysis of this graph I , R , and G are conditionally independent given O and S .

Bibliography

Lauritzen, S. L. (1996) *Graphical Models*. New York: Oxford University Press.

Pearl, J. (2009) *Causality: Models, Reasoning, and Inference*. Second. Cambridge: Cambridge University Press.

Spirtes, P., Glymour, C. and Scheines, R. (2000) *Causation, Prediction, and Search*. Second. Cambridge, MA: MIT Press.