

Romania SEM

Marissa Dyck

2024-08-08

Contents

Before you begin	1
Notes	1
R and RStudio	2
R markdown	2
Install packages	2
Load libraries	2
Data	2
Import	2
Summary statistics	3
Correlations	4
Analysis	5

Before you begin

Notes

A few notes about this script.

If you are want to run through the full analysis with the published data make sure you download the whole (Romania_SEM repository)[https://github.com/marissadyck/Romania_SEM] from Marissa Dyck's GitHub. This will ensure you have all the files, data, and proper folder structure you will need to run this code and associated analyses.

Also make sure you open RStudio through the R project (Romania_SEM.Rproj) this will automatically set your working directory to the correct place (wherever you saved the repository) and ensure you don't have to change the file paths for the data.

If you have question please email the author,

Marissa A. Dyck
Postdoctoral research fellow
University of Victoria
School of Environmental Studies
Email: marissadyck17@gmail.com

R and RStudio

Before starting you should ensure you have the latest version of R and RStudio downloaded. This code was generated under R version 4.2.3 and with RStudio version 2024.04.2+764.

You can download R and RStudio [HERE](#)

R markdown

This script is written in R markdown and thus uses a mix of coding markup languages and R. If you are planning to run this script with new data or make any modifications you will want to be familiar with some basics of R markdown.

Below is an R markdown cheatsheet to help you get started,
R markdown cheatsheet

Install packages

If you don't already have the following packages installed, use the code below to install them.

```
install.packages('tidyverse')
install.packages('piecewiseSEM')
install.packages("PerformanceAnalytics")
install.packages("purrrlyr")
```

Load libraries

Then load the packages to your library for this current session.

```
library(piecewiseSEM) # used to run piecewise structural equation models
library(tidyverse) # data tidying, visualization, and much more; this will load all tidyverse packages
library(PerformanceAnalytics) # generates correlation chart
library(purrrlyr) # data manipulation
```

Data

Import

Let's read in the data that we cleaned and formatted with the earlier code '1_RO_data_formatting'

In this code chunk I will also add columns with the scaled the predictor variables fo analysis

```
ro_sem_dat <- read_csv('data/processed/ro_sem_dat_2018-2019.csv') %>%

  # scale predictor variables
  mutate(z.s = scale(z),
         denslocalr.s = scale(denslocalr),
         tri5.s = scale(tri5),
         clc_forest.s = scale(clc_forest))
```

```
## Rows: 138 Columns: 29
## -- Column specification -----
## Delimiter: ","
## chr (1): trapcode
## dbl (28): session, fox, wolf, badger, lynx, wild_boar, mustelid, bird, red_d...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Summary statistics

We will want to report some summary statistics for our data, the code chunks below will extract/calculate those details

Total occurrences

Let's calculate the total occurrences for 8 focal species by session/season

```
ro_sem_dat %>%

  # group by session
  group_by(session) %>%

  # calculate occurrences for species of interest (8)
  summarise(total_8 = sum(hare,
                          fox,
                          lynx,
                          wolf,
                          red_deer,
                          roe_deer,
                          wild_boar,
                          wild_cat))
```

```
## # A tibble: 2 x 2
##   session total_8
##   <dbl>   <dbl>
## 1     2    1626
## 2     3    1973
```

Species occurrences

We will also want some summary information for each species

```
# total species occurrence summaries

ro_sem_dat %>%

  # select species of interest
  select(badger:otter) %>%

  # use purrr map to calculate summary stats for all columns listed
```

```
dmap(~c(sum(.x),
        min(.x),
        max(.x),
        (sum(.x)/3599*100))) %>%

# create columns for new data to report
mutate(stat = c('sum',
                'min',
                'max',
                'percent')) %>%

relocate(stat)
```

```
## # A tibble: 4 x 18
##   stat    badger   lynx wild_boar mustelid  bird red_deer  bear roe_deer  hare
##   <chr>    <dbl>  <dbl>    <dbl>    <dbl> <dbl>    <dbl> <dbl>    <dbl> <dbl>
## 1 sum      208    332      694      69    46      772    604      386    67
## 2 min       0     0        0        0     0        0     0        0     0
## 3 max      83    20       79        9    12       81    49       24     9
## 4 percent  5.78  9.22    19.3     1.92  1.28    21.5   16.8    10.7   1.86
## # i 8 more variables: human <dbl>, wild_cat <dbl>, dog <dbl>, chamois <dbl>,
## #   squirrel <dbl>, livestock <dbl>, hedgehog <dbl>, otter <dbl>
```

Environmental variables

And we will want to report some summary info on the environmental variables we are using in the analysis

```
ro_sem_dat %>%

# select columns of raw data (not scaled)
select(z, denslocalr, tri5, clc_forest) %>%

# print summary info for each
summary()
```

```
##           z           denslocalr           tri5           clc_forest
## Min.      : 663      Min.      :0.2127      Min.      : 66.38      Min.      :0.08505
## 1st Qu.:1025      1st Qu.:0.2413      1st Qu.:178.43      1st Qu.:0.64883
## Median :1168      Median :0.2705      Median :221.61      Median :0.83402
## Mean      :1166      Mean      :0.2702      Mean      :220.27      Mean      :0.76541
## 3rd Qu.:1293      3rd Qu.:0.2939      3rd Qu.:260.65      3rd Qu.:0.91221
## Max.      :1617      Max.      :0.3434      Max.      :494.01      Max.      :1.00000
```

Correlations

Before we run any models we need to test an assumption of independence and see if any of our potential variables are correlated.

```
# subset data to just environmental variables for correlation tests

ro_sem_env <- ro_sem_dat %>%
  select(tri5.s,
```

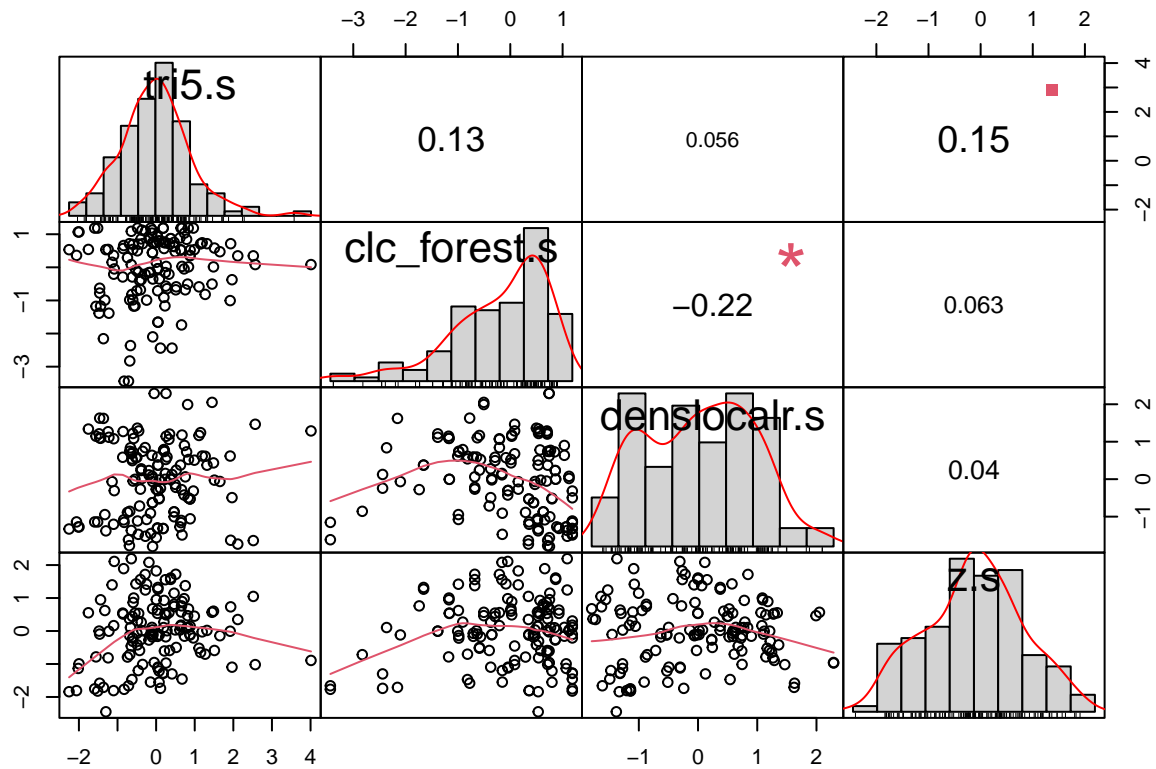
```

    clc_forest.s,
    denslocalr.s,
    z.s)

# get pairwise correlation matrix

chart.Correlation(ro_sem_env,
                  histogram = TRUE,
                  pch = 19)

```



all correlations for chosen environmental variables > 0.25

Analysis

The code chunks below define the models for our three hypotheses

- top-down
- bottom-up
- combined

Top-down

Hypothesis This first chunk is our initial model as we defined it a priori based on previous research and the literature review

```
sem.top_down_a <-  
  psem(  
    glm(lynx ~ denslocalr.s + clc_forest.s, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(wolf ~ denslocalr.s + clc_forest.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(fox ~ lynx + wolf, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(wild_cat ~ fox + lynx + z.s + tri5.s + clc_forest.s, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(hare ~ lynx + wild_cat + fox, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(red_deer ~ wolf, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(roe_deer ~ wolf + lynx, family = poisson(link = 'log'), data = ro_sem_dat),  
    glm(wild_boar ~ wolf, family = poisson(link = 'log'), data = ro_sem_dat),  
    (lynx %~~% wolf)  
  )  
  
summary(sem.top_down_a)
```

|

|

```

##
##      AIC
## 8475.855
##
## ---
## Tests of directed separation:
##
##      Independ.Claim Test.Type  DF Crit.Value P.Value
## wild_cat ~ denslocalr.s + ...   coef 131    -0.0094  0.9925
## fox ~ denslocalr.s + ...        coef 134    -2.8265  0.0047  **
## hare ~ denslocalr.s + ...        coef 133     1.3486  0.1775
## red_deer ~ denslocalr.s + ...    coef 135   -14.3898  0.0000  ***
## roe_deer ~ denslocalr.s + ...    coef 134    -1.7484  0.0804
## wild_boar ~ denslocalr.s + ...   coef 135    -6.6424  0.0000  ***
## fox ~ clc_forest.s + ...        coef 134    -3.3793  0.0007  ***
## hare ~ clc_forest.s + ...        coef 133    -0.7246  0.4687
## red_deer ~ clc_forest.s + ...    coef 135     7.3350  0.0000  ***
## roe_deer ~ clc_forest.s + ...    coef 134    -5.0679  0.0000  ***
## wild_boar ~ clc_forest.s + ...   coef 135    -6.6483  0.0000  ***
## lynx ~ tri5.s + ...             coef 134    -3.3142  0.0009  ***
## fox ~ tri5.s + ...             coef 134    -2.8160  0.0049  **
## hare ~ tri5.s + ...             coef 133    -2.6859  0.0072  **
## red_deer ~ tri5.s + ...          coef 135    -3.6073  0.0003  ***
## roe_deer ~ tri5.s + ...          coef 134    -9.7095  0.0000  ***
## wild_boar ~ tri5.s + ...         coef 135   -10.6837  0.0000  ***
## lynx ~ z.s + ...               coef 134     1.0908  0.2753
## wolf ~ z.s + ...               coef 133     2.6960  0.0070  **
## fox ~ z.s + ...               coef 134   -11.7849  0.0000  ***
## hare ~ z.s + ...               coef 133     3.2223  0.0013  **
## red_deer ~ z.s + ...           coef 135    -2.8774  0.0040  **
## roe_deer ~ z.s + ...           coef 134   -12.0251  0.0000  ***
## wild_boar ~ z.s + ...           coef 135   -11.6814  0.0000  ***
## red_deer ~ lynx + ...           coef 133     8.6889  0.0000  ***
## wild_boar ~ lynx + ...           coef 133     7.1423  0.0000  ***
## wild_cat ~ wolf + ...           coef 130    -0.3952  0.6927
## hare ~ wolf + ...              coef 130     1.9729  0.0485  *
## red_deer ~ wild_cat + ...        coef 130     7.5480  0.0000  ***
## roe_deer ~ wild_cat + ...        coef 130     1.9185  0.0551
## wild_boar ~ wild_cat + ...       coef 130    13.1172  0.0000  ***
## red_deer ~ fox + ...            coef 134     2.5796  0.0099  **
## roe_deer ~ fox + ...            coef 134    13.1233  0.0000  ***
## wild_boar ~ fox + ...            coef 134    12.8915  0.0000  ***
## red_deer ~ hare + ...           coef 132    11.4736  0.0000  ***
## roe_deer ~ hare + ...           coef 132     6.6932  0.0000  ***
## wild_boar ~ hare + ...           coef 132     6.6593  0.0000  ***
## roe_deer ~ red_deer + ...        coef 134     3.4164  0.0006  ***
## wild_boar ~ red_deer + ...       coef 135    15.4314  0.0000  ***
## wild_boar ~ roe_deer + ...       coef 134    18.7395  0.0000  ***
##
## --
## Global goodness-of-fit:
##
## Chi-Squared = 1553.174 with P-value = 0 and on 40 degrees of freedom
## Fisher's C = 2772.318 with P-value = 0 and on 80 degrees of freedom

```

```

##
## ---
## Coefficients:
##
##      Response      Predictor Estimate Std.Error  DF Crit.Value P.Value Std.Estimate
##      lynx denslocalr.s -0.0497   0.0565 135   -0.8804  0.3786   -0.0333
##      lynx clc_forest.s -0.0684   0.0536 135   -1.2770  0.2016   -0.0457
##      wolf denslocalr.s -0.6057   0.0729 134   -8.3043  0.0000   -0.3384
##      wolf clc_forest.s  0.0758   0.0646 134    1.1743  0.2403    0.0424
##      wolf      tri5.s -0.1310   0.0632 134   -2.0733  0.0381   -0.0731
##      fox      lynx  0.0273   0.0087 135    3.1259  0.0018    0.1078
##      fox      wolf  0.0465   0.0063 135    7.3987  0.0000    0.1961
##      wild_cat      fox  0.0458   0.0113 132    4.0610  0.0000    0.1732
##      wild_cat      lynx  0.0500   0.0258 132    1.9346  0.0530    0.0824
##      wild_cat      z.s -0.3170   0.1125 132   -2.8173  0.0048   -0.1621
##      wild_cat      tri5.s  0.0715   0.0894 132    0.8002  0.4236    0.0366
##      wild_cat clc_forest.s  0.2047   0.0974 132    2.1012  0.0356    0.1046
##      hare      lynx  0.0952   0.0259 134    3.6736  0.0002    0.1026
##      hare      wild_cat -0.2068   0.1184 134   -1.7461  0.0808   -0.1381
##      hare      fox  0.0035   0.017 134    0.2033  0.8389    0.0085
##      red_deer      wolf  0.0395   0.0075 136    5.2962  0.0000    0.0511
##      roe_deer      wolf  0.0251   0.0124 135    2.0290  0.0425    0.0477
##      roe_deer      lynx -0.0048   0.0162 135   -0.2983  0.7654   -0.0086
##      wild_boar      wolf  0.0461   0.0074 136    6.2464  0.0000    0.0639
##      ~~lynx      ~~wolf  0.3239      - 138    3.9777  0.0001    0.3239
##
##
##
## ***
##
## *
## **
## ***
## ***
##
## **
##
## *
## ***
##
##
## ***
## *
##
## ***
## ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
##
## ---
## Individual R-squared:
##
##      Response      method R.squared
##      lynx nagelkerke      0.01

```



```
##      wolf nagelkerke      0.52
##      fox  nagelkerke      0.37
##      wild_cat nagelkerke    0.32
##      hare  nagelkerke      0.12
##      red_deer nagelkerke    0.15
##      roe_deer nagelkerke    0.03
##      wild_boar nagelkerke   0.20
```

Optimized Now, based on the tests of directed separation, we will add significant pathways to model that make biological sense and fit hypotheses (top-down) and remove non-significant pathways and pathways with coefficient estimates that do not fit the hypothesis (e.g. positive estimate for a hypothesized negative causal pathway)

```
sem.top_down_b <-
  psem(
    glm(lynx ~ tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wolf ~ denslocalr.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(fox ~ lynx + wolf + z.s + clc_forest.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wild_cat ~ lynx + wolf + z.s + clc_forest.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(hare ~ wild_cat, family = poisson(link = 'log'), data = ro_sem_dat),
    #glm(red_deer ~ wolf, family = poisson(link = 'log'), data = ro_sem_dat),
    #glm(roe_deer ~ lynx, family = poisson(link = 'log'), data = ro_sem_dat),
    #glm(wild_boar ~ wolf, family = poisson(link = 'log'), data = ro_sem_dat),
    (lynx %~~% wolf),
    (wild_cat %~~% fox),
    (hare %~~% wolf),
    (hare %~~% lynx)
  )

summary(sem.top_down_b)
```

```
##      |
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
## Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
## Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
## Use c() or as.vector() instead.
##
## Structural Equation Model of sem.top_down_b
##
## Call:
##   lynx ~ tri5.s
##   wolf ~ denslocalr.s + z.s + tri5.s
##   fox ~ lynx + wolf + z.s + clc_forest.s
##   wild_cat ~ lynx + wolf + z.s + clc_forest.s
##   hare ~ wild_cat
##   lynx ~~ wolf
```

```

## wild_cat ~~ fox
## hare ~~ wolf
## hare ~~ lynx
##
## AIC
## 3369.631
##
## ---
## Tests of directed separation:
##
##      Independ.Claim Test.Type  DF Crit.Value P.Value
##      fox ~ tri5.s + ...      coef 132    0.3132  0.7541
##      wild_cat ~ tri5.s + ...   coef 132    0.6513  0.5149
##      hare ~ tri5.s + ...      coef 135   -3.1673  0.0015 **
##      lynx ~ denslocalr.s + ... coef 135   -0.4394  0.6604
##      fox ~ denslocalr.s + ...   coef 132   -1.3365  0.1814
##      wild_cat ~ denslocalr.s + ... coef 132   -0.2598  0.7950
##      hare ~ denslocalr.s + ...   coef 135    1.0493  0.2940
##      lynx ~ z.s + ...         coef 135    1.6253  0.1041
##      hare ~ z.s + ...         coef 135    3.0227  0.0025 **
##      lynx ~ clc_forest.s + ...  coef 135   -0.6614  0.5084
##      wolf ~ clc_forest.s + ...   coef 133    0.9113  0.3621
##      hare ~ clc_forest.s + ...   coef 135   -0.9177  0.3588
##      hare ~ fox + ...         coef 131    0.5774  0.5637
##
## --
## Global goodness-of-fit:
##
## Chi-Squared = 43.231 with P-value = 0 and on 15 degrees of freedom
## Fisher's C = 45.082 with P-value = 0.012 and on 26 degrees of freedom
##
## ---
## Coefficients:
##
##      Response      Predictor Estimate Std.Error  DF Crit.Value P.Value Std.Estimate
##      lynx      tri5.s    -0.2001   0.0575  136   -3.4776  0.0005   -0.1548
##      wolf denslocalr.s  -0.6493   0.0707  134   -9.1788  0.0000   -0.3573
##      wolf      z.s      0.1630   0.0581  134    2.8040  0.0050    0.0897
##      wolf      tri5.s   -0.1808   0.0679  134   -2.6612  0.0078   -0.0995
##      fox      lynx      0.0287   0.0084  133    3.4258  0.0006    0.0945
##      fox      wolf      0.0562   0.0065  133    8.6358  0.0000    0.1980
##      fox      z.s     -0.3768   0.0326  133  -11.5684  0.0000   -0.3854
##      fox clc_forest.s -0.0720   0.0288  133   -2.4976  0.0125   -0.0737
##      wild_cat      lynx   0.0535   0.0245  133    2.1881  0.0287    0.0744
##      wild_cat      wolf   0.0206   0.0255  133    0.8099  0.4180    0.0306
##      wild_cat      z.s   -0.4697   0.0985  133   -4.7686  0.0000   -0.2023
##      wild_cat clc_forest.s 0.2132   0.1021  133    2.0881  0.0368    0.0918
##      hare      wild_cat -0.1792   0.1153  136   -1.5541  0.1202   -0.0649
##      ~~lynx      ~~wolf   0.3159    - 138    3.8684  0.0001    0.3159
##      ~~wild_cat      ~~fox 0.1653    - 138    1.9470  0.0268    0.1653
##      ~~hare      ~~wolf 0.2029    - 138    2.4070  0.0087    0.2029
##      ~~hare      ~~lynx 0.2144    - 138    2.5505  0.0059    0.2144
##
## ***

```

```
## ***
## **
## **
## ***
## ***
## ***
## *
## *
##
## ***
## *
##
## ***
## *
## **
## **
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
##
## ---
## Individual R-squared:
##
## Response      method R.squared
##      lynx nagelkerke      0.09
##      wolf nagelkerke      0.54
##      fox  nagelkerke      0.78
## wild_cat nagelkerke      0.23
##      hare nagelkerke      0.03
```

Bottom-up

This section will define our bottom-up model

Hypothesis Again, this first chunk is our initial model as we defined it a priori based on previous research and the literature review

```
sem.bottom_up_a <-
  psem(
    glm(wild_boar ~ clc_forest.s + denslocalr.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(roe_deer ~ clc_forest.s + denslocalr.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(red_deer ~ clc_forest.s + z.s + denslocalr.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(hare ~ clc_forest.s + denslocalr.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wild_cat ~ hare, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(fox ~ hare, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wolf ~ red_deer + wild_boar + roe_deer, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(lynx ~ hare + roe_deer, family = poisson(link = 'log'), data = ro_sem_dat),
    (roe_deer %~~% wild_boar),
    (red_deer %~~% wild_boar),
    (hare %~~% wild_boar),
    (roe_deer %~~% red_deer),
    (roe_deer %~~% hare),
    (red_deer %~~% hare)
  )
```

```
summary(sem.bottom_up_a)
```

```
##      |
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
##
## Structural Equation Model of sem.bottom_up_a
##
## Call:
##   wild_boar ~ clc_forest.s + denslocalr.s + z.s + tri5.s
##   roe_deer ~ clc_forest.s + denslocalr.s + z.s + tri5.s
##   red_deer ~ clc_forest.s + z.s + denslocalr.s + tri5.s
##   hare ~ clc_forest.s + denslocalr.s
##   wild_cat ~ hare
##   fox ~ hare
##   wolf ~ red_deer + wild_boar + roe_deer
##   lynx ~ hare + roe_deer
##   roe_deer ~~ wild_boar
##   red_deer ~~ wild_boar
##   hare ~~ wild_boar
##   roe_deer ~~ red_deer
##   roe_deer ~~ hare
##   red_deer ~~ hare
##
##      AIC
## 8004.192
##
## ---
## Tests of directed separation:
##
##      Independ.Claim Test.Type  DF Crit.Value P.Value
## wild_cat ~ clc_forest.s + ...    coef 135      1.9473 0.0515
##      fox ~ clc_forest.s + ...    coef 135     -2.5304 0.0114 *
##      wolf ~ clc_forest.s + ...    coef 133      3.5417 0.0004 ***
##      lynx ~ clc_forest.s + ...    coef 134     -1.0418 0.2975
```

```

## wild_cat ~ denslocalr.s + ... coef 135 -1.6969 0.0897
## fox ~ denslocalr.s + ... coef 135 -4.9571 0.0000 ***
## wolf ~ denslocalr.s + ... coef 133 -8.8014 0.0000 ***
## lynx ~ denslocalr.s + ... coef 134 -0.8803 0.3787
## hare ~ z.s + ... coef 134 3.3629 0.0008 ***
## wild_cat ~ z.s + ... coef 135 -4.4477 0.0000 ***
## fox ~ z.s + ... coef 135 -11.3492 0.0000 ***
## wolf ~ z.s + ... coef 133 2.4445 0.0145 *
## lynx ~ z.s + ... coef 134 0.2185 0.8271
## hare ~ tri5.s + ... coef 134 -3.0844 0.0020 **
## wild_cat ~ tri5.s + ... coef 135 -1.0210 0.3073
## fox ~ tri5.s + ... coef 135 -3.7757 0.0002 ***
## wolf ~ tri5.s + ... coef 133 -1.9946 0.0461 *
## lynx ~ tri5.s + ... coef 134 -3.3464 0.0008 ***
## wild_cat ~ wild_boar + ... coef 131 8.2155 0.0000 ***
## fox ~ wild_boar + ... coef 131 4.5218 0.0000 ***
## lynx ~ wild_boar + ... coef 130 3.7369 0.0002 ***
## wild_cat ~ roe_deer + ... coef 131 4.0925 0.0000 ***
## fox ~ roe_deer + ... coef 131 7.3577 0.0000 ***
## wild_cat ~ red_deer + ... coef 131 3.2458 0.0012 **
## fox ~ red_deer + ... coef 131 0.5356 0.5923
## lynx ~ red_deer + ... coef 130 2.2286 0.0258 *
## wolf ~ hare + ... coef 131 4.2267 0.0000 ***
## fox ~ wild_cat + ... coef 135 7.7661 0.0000 ***
## wolf ~ wild_cat + ... coef 132 0.0040 0.9968
## lynx ~ wild_cat + ... coef 134 2.6527 0.0080 **
## wolf ~ fox + ... coef 132 7.6386 0.0000 ***
## lynx ~ fox + ... coef 134 4.2934 0.0000 ***
## lynx ~ wolf + ... coef 132 5.5700 0.0000 ***
##
## --
## Global goodness-of-fit:
##
## Chi-Squared = 531.355 with P-value = 0 and on 33 degrees of freedom
## Fisher's C = 813.142 with P-value = 0 and on 66 degrees of freedom
##
## ---
## Coefficients:
##
## Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
## wild_boar clc_forest.s -0.1592 0.0302 133 -5.2681 0.0000 -0.0925
## wild_boar denslocalr.s -0.2643 0.0398 133 -6.6402 0.0000 -0.1537
## wild_boar z.s -0.2510 0.039 133 -6.4367 0.0000 -0.1459
## wild_boar tri5.s -0.3137 0.0427 133 -7.3475 0.0000 -0.1824
## roe_deer clc_forest.s -0.1207 0.0411 133 -2.9347 0.0033 -0.0956
## roe_deer denslocalr.s -0.0133 0.0506 133 -0.2630 0.7925 -0.0105
## roe_deer z.s -0.4822 0.0564 133 -8.5516 0.0000 -0.3820
## roe_deer tri5.s -0.3375 0.0566 133 -5.9640 0.0000 -0.2673
## red_deer clc_forest.s 0.1632 0.0402 133 4.0639 0.0000 0.0788
## red_deer z.s -0.0213 0.0345 133 -0.6169 0.5373 -0.0103
## red_deer denslocalr.s -0.5283 0.0415 133 -12.7184 0.0000 -0.2551
## red_deer tri5.s -0.1115 0.0379 133 -2.9452 0.0032 -0.0538
## hare clc_forest.s -0.1014 0.1194 135 -0.8493 0.3957 -0.0270
## hare denslocalr.s 0.1089 0.1256 135 0.8670 0.3859 0.0290

```

```

##      wild_cat      hare -0.1993    0.1134 136    -1.7577  0.0788    -0.0866
##      fox      hare    0.0091     0.023 136     0.3951  0.6927     0.0109
##      wolf     red_deer  0.0077     0.0046 134     1.6799  0.0930     0.0656
##      wolf     wild_boar 0.0115     0.0053 134     2.1697  0.0300     0.0829
##      wolf     roe_deer  0.0071     0.0138 134     0.5158  0.6060     0.0239
##      lynx      hare    0.1171     0.0308 135     3.8078  0.0001     0.1152
##      lynx      roe_deer -0.0068     0.0124 135    -0.5462  0.5850    -0.0229
##  ~~roe_deer  ~~wild_boar 0.3162      - 138     3.8731  0.0001     0.3162
##  ~~red_deer  ~~wild_boar 0.4153      - 138     5.3049  0.0000     0.4153
##  ~~hare     ~~wild_boar 0.2145      - 138     2.5521  0.0059     0.2145
##  ~~roe_deer  ~~red_deer  0.1601      - 138     1.8846  0.0308     0.1601
##  ~~roe_deer  ~~hare     0.2132      - 138     2.5360  0.0062     0.2132
##  ~~red_deer  ~~hare     0.1287      - 138     1.5083  0.0669     0.1287
##
##  ***
##  ***
##  ***
##  ***
##  **
##
##  ***
##  ***
##  ***
##
##  ***
##  **
##
##
##
##
##  *
##
##  ***
##
##  ***
##  ***
##  **
##  *
##  **
##
##
##
##  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
##
##  ---
##  Individual R-squared:
##
##      Response      method R.squared
##  wild_boar nagelkerke    0.85
##  roe_deer  nagelkerke    0.76
##  red_deer  nagelkerke    0.86
##      hare  nagelkerke    0.02
##  wild_cat  nagelkerke    0.03
##      fox  nagelkerke    0.00

```

```
##      wolf nagelkerke      0.09
##      lynx nagelkerke      0.09
```

Optimized And again, based on the tests of directed separation, we will add significant pathways to model that make biological sense and fit hypotheses (top-down) and remove non-significant pathways and pathways with coefficient estimates that do not fit the hypothesis (e.g. positive estimate for a hypothesized negative causal pathway)

```
sem.bottom_up_b <-
  psem(
    glm(wild_boar ~ clc_forest.s + denslocalr.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(roe_deer ~ clc_forest.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(red_deer ~ clc_forest.s + z.s + denslocalr.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(hare ~ tri5.s + z.s, family = poisson(link = 'log'), data = ro_sem_dat),
    #glm(wild_cat ~ wild_boar, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(fox ~ roe_deer, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wolf ~ red_deer + wild_boar, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(lynx ~ hare, family = poisson(link = 'log'), data = ro_sem_dat),
    (roe_deer %~~% wild_boar),
    (red_deer %~~% wild_boar),
    (hare %~~% wild_boar),
    (roe_deer %~~% red_deer),
    (roe_deer %~~% hare),
    (red_deer %~~% hare),
    (lynx %~~% wolf),
    (wolf %~~% fox),
    (lynx %~~% fox),
    (wild_boar %~~% fox),
    (lynx %~~% wild_boar),
    (lynx %~~% red_deer)
  )

summary(sem.bottom_up_b)
```

```
##      |
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##      Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##      Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##      Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##      Use c() or as.vector() instead.
##
##
```

```

## Structural Equation Model of sem.bottom_up_b
##
## Call:
##   wild_boar ~ clc_forest.s + denslocalr.s + z.s + tri5.s
##   roe_deer ~ clc_forest.s + z.s + tri5.s
##   red_deer ~ clc_forest.s + z.s + denslocalr.s + tri5.s
##   hare ~ tri5.s + z.s
##   fox ~ roe_deer
##   wolf ~ red_deer + wild_boar
##   lynx ~ hare
##   roe_deer ~~ wild_boar
##   red_deer ~~ wild_boar
##   hare ~~ wild_boar
##   roe_deer ~~ red_deer
##   roe_deer ~~ hare
##   red_deer ~~ hare
##   lynx ~~ wolf
##   wolf ~~ fox
##   lynx ~~ fox
##   wild_boar ~~ fox
##   lynx ~~ wild_boar
##   lynx ~~ red_deer
##
##   AIC
##   7380.065
##
## ---
## Tests of directed separation:
##
##           Independ.Claim Test.Type  DF Crit.Value P.Value
##   hare ~ clc_forest.s + ...      coef 134    -0.7911  0.4289
##   fox ~ clc_forest.s + ...      coef 135    -0.1637  0.8700
##   wolf ~ clc_forest.s + ...      coef 134     3.5265  0.0004 ***
##   lynx ~ clc_forest.s + ...      coef 135    -0.9751  0.3295
##   roe_deer ~ denslocalr.s + ...   coef 133    -0.2630  0.7925
##   hare ~ denslocalr.s + ...      coef 134     1.0821  0.2792
##   fox ~ denslocalr.s + ...      coef 135    -4.1293  0.0000 ***
##   wolf ~ denslocalr.s + ...      coef 134    -8.8107  0.0000 ***
##   lynx ~ denslocalr.s + ...      coef 135    -0.8332  0.4047
##   fox ~ z.s + ...               coef 135    -6.7716  0.0000 ***
##   wolf ~ z.s + ...              coef 134     2.1504  0.0315  *
##   lynx ~ z.s + ...              coef 135     0.4238  0.6717
##   fox ~ tri5.s + ...            coef 135    -0.0964  0.9232
##   wolf ~ tri5.s + ...           coef 134    -2.0646  0.0390  *
##   lynx ~ tri5.s + ...           coef 135    -3.0540  0.0023  **
##   wolf ~ roe_deer + ...         coef 131     0.7179  0.4728
##   lynx ~ roe_deer + ...         coef 132    -1.2100  0.2263
##   fox ~ red_deer + ...          coef 131     0.2461  0.8056
##   fox ~ hare + ...              coef 133     0.5676  0.5703
##   wolf ~ hare + ...             coef 132     1.5017  0.1332
##
## --
## Global goodness-of-fit:
##

```



```

## Chi-Squared = 185.015 with P-value = 0 and on 20 degrees of freedom
## Fisher's C = 214.238 with P-value = 0 and on 40 degrees of freedom
##
## ---
## Coefficients:
##
##      Response      Predictor Estimate Std.Error  DF Crit.Value P.Value
##      wild_boar clc_forest.s  -0.1592   0.0302 133   -5.2681  0.0000
##      wild_boar denslocalr.s  -0.2643   0.0398 133   -6.6402  0.0000
##      wild_boar          z.s   -0.2510   0.039 133   -6.4367  0.0000
##      wild_boar      tri5.s   -0.3137   0.0427 133   -7.3475  0.0000
##      roe_deer  clc_forest.s  -0.1187   0.0406 134   -2.9263  0.0034
##      roe_deer          z.s   -0.4862   0.0544 134   -8.9396  0.0000
##      roe_deer      tri5.s   -0.3383   0.0565 134   -5.9920  0.0000
##      red_deer  clc_forest.s   0.1632   0.0402 133    4.0639  0.0000
##      red_deer          z.s   -0.0213   0.0345 133   -0.6169  0.5373
##      red_deer denslocalr.s  -0.5283   0.0415 133  -12.7184  0.0000
##      red_deer      tri5.s   -0.1115   0.0379 133   -2.9452  0.0032
##      hare      tri5.s   -0.5444   0.1488 135   -3.6591  0.0003
##      hare          z.s    0.4595   0.118 135    3.8932  0.0001
##      fox      roe_deer   0.0630   0.0048 136   13.0686  0.0000
##      wolf      red_deer   0.0077   0.0046 135    1.6722  0.0945
##      wolf      wild_boar   0.0128   0.0047 135    2.7296  0.0063
##      lynx      hare    0.1130   0.0298 136    3.7961  0.0001
##      ~~roe_deer ~~wild_boar 0.3163      - 138    3.8733  0.0001
##      ~~red_deer ~~wild_boar 0.4153      - 138    5.3049  0.0000
##      ~~hare     ~~wild_boar 0.2349      - 138    2.8084  0.0029
##      ~~roe_deer ~~red_deer  0.1604      - 138    1.8880  0.0306
##      ~~roe_deer  ~~hare    0.2040      - 138    2.4206  0.0084
##      ~~red_deer  ~~hare    0.1183      - 138    1.3841  0.0843
##      ~~lynx     ~~wolf    0.2759      - 138    3.3353  0.0006
##      ~~wolf     ~~fox     0.2912      - 138    3.5370  0.0003
##      ~~lynx     ~~fox     0.2227      - 138    2.6547  0.0044
##      ~~wild_boar  ~~fox    0.0902      - 138    1.0524  0.1472
##      ~~lynx     ~~wild_boar 0.1702      - 138    2.0068  0.0234
##      ~~lynx     ~~red_deer 0.1704      - 138    2.0089  0.0233
##      Std.Estimate
##      -0.0925 ***
##      -0.1537 ***
##      -0.1459 ***
##      -0.1824 ***
##      -0.0940 **
##      -0.3848 ***
##      -0.2677 ***
##      0.0788 ***
##      -0.0103
##      -0.2551 ***
##      -0.0538 **
##      -0.1642 ***
##      0.1386 ***
##      0.4273 ***
##      0.0642
##      0.0906 **
##      0.1121 ***

```

```
##          0.3163 ***
##          0.4153 ***
##          0.2349 **
##          0.1604 *
##          0.2040 **
##          0.1183
##          0.2759 ***
##          0.2912 ***
##          0.2227 **
##          0.0902
##          0.1702 *
##          0.1704 *
##
##   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
##
## ---
## Individual R-squared:
##
##      Response      method R.squared
##  wild_boar nagelkerke      0.85
##    roe_deer nagelkerke      0.76
##   red_deer nagelkerke      0.86
##        hare nagelkerke      0.20
##        fox nagelkerke      0.63
##       wolf nagelkerke      0.09
##       lynx nagelkerke      0.08
```

Combined

Lastly we will define a combined model that includes both bottom-up and top-down processes/interactions. This model will be based on the optimized models for both top-down and bottom-up

```
sem.combined <-
  psem(
    glm(lynx ~ tri5.s + hare, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wolf ~ z.s + tri5.s + denslocalr.s + wild_boar, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(fox ~ z.s + tri5.s + wolf + lynx + roe_deer, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wild_cat ~ clc_forest.s + z.s + lynx, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(hare ~ tri5.s + z.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(red_deer ~ clc_forest.s + denslocalr.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(roe_deer ~ clc_forest.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    glm(wild_boar ~ clc_forest.s + denslocalr.s + z.s + tri5.s, family = poisson(link = 'log'), data = ro_sem_dat),
    (lynx %~~% wolf),
    (wild_boar %~~% hare),
    (roe_deer %~~% hare),
    (red_deer %~~% hare),
    (wild_boar %~~% red_deer),
    (red_deer %~~% roe_deer),
    (wild_boar %~~% roe_deer),
    (wild_cat %~~% fox),
    (red_deer %~~% lynx),
    (wolf %~~% hare),
    (wild_cat %~~% wild_boar),
```

```

(wild_cat %~~% roe_deer),
(wild_cat %~~% red_deer),
(lynx %~~% wild_boar)
)

summary(sem.combined,
        conserve = TRUE)

```

```

##      |
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
## Warning in sd.x/sd.y: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
##
##
## Structural Equation Model of sem.combined
##
## Call:
##   lynx ~ tri5.s + hare
##   wolf ~ z.s + tri5.s + denslocalr.s + wild_boar
##   fox ~ z.s + tri5.s + wolf + lynx + roe_deer
##   wild_cat ~ clc_forest.s + z.s + lynx
##   hare ~ tri5.s + z.s
##   red_deer ~ clc_forest.s + denslocalr.s + tri5.s
##   roe_deer ~ clc_forest.s + z.s + tri5.s
##   wild_boar ~ clc_forest.s + denslocalr.s + z.s + tri5.s
##   lynx ~~ wolf
##   wild_boar ~~ hare
##   roe_deer ~~ hare
##   red_deer ~~ hare
##   wild_boar ~~ red_deer
##   red_deer ~~ roe_deer
##   wild_boar ~~ roe_deer

```

```

## wild_cat ~~ fox
## red_deer ~~ lynx
## wolf ~~ hare
## wild_cat ~~ wild_boar
## wild_cat ~~ roe_deer
## wild_cat ~~ red_deer
## lynx ~~ wild_boar
##
## AIC
## 7600.344
##
## ---
## Tests of directed separation:
##
##          Independ.Claim Test.Type  DF Crit.Value P.Value
##      wild_cat ~ tri5.s + ...      coef 133      0.5723 0.5671
##      lynx ~ z.s + ...             coef 134      1.1267 0.2599
##      red_deer ~ z.s + ...          coef 133     -0.6169 0.5373
##      lynx ~ denslocalr.s + ...      coef 134     -0.6754 0.4994
##      hare ~ denslocalr.s + ...      coef 134      1.0821 0.2792
##      fox ~ denslocalr.s + ...       coef 131     -1.0898 0.2758
##      roe_deer ~ denslocalr.s + ...   coef 133     -0.2630 0.7925
##      wild_cat ~ denslocalr.s + ...   coef 133     -0.5001 0.6170
##      lynx ~ clc_forest.s + ...       coef 134     -0.6531 0.5137
##      hare ~ clc_forest.s + ...       coef 134     -0.7911 0.4289
##      wolf ~ clc_forest.s + ...       coef 132      1.0614 0.2885
##      fox ~ clc_forest.s + ...       coef 131     -1.3297 0.1836
##      lynx ~ roe_deer + ...          coef 132     -1.2100 0.2263
##      fox ~ hare + ...              coef 131     -0.5015 0.6160
##      wild_cat ~ hare + ...          coef 132     -1.6533 0.0983
##      roe_deer ~ wolf + ...          coef 131      1.4863 0.1372
##      red_deer ~ wolf + ...          coef 131     -1.7058 0.0880
##      wild_cat ~ wolf + ...          coef 130     -0.6836 0.4942
##      fox ~ wild_boar + ...          coef 129      1.2506 0.2111
##      red_deer ~ fox + ...           coef 129     -1.5120 0.1305
##
## --
## Global goodness-of-fit:
##
## Chi-Squared = 0 with P-value = 1 and on 30 degrees of freedom
## Fisher's C = 47.927 with P-value = 0.182 and on 40 degrees of freedom
##
## ---
## Coefficients:
##
##      Response      Predictor Estimate Std.Error  DF Crit.Value P.Value
##      lynx      tri5.s    -0.1780   0.0583 135   -3.0540 0.0023
##      lynx      hare      0.0984   0.0303 135    3.2505 0.0012
##      wolf      z.s       0.1963   0.0605 133    3.2428 0.0012
##      wolf      tri5.s    -0.1540   0.0697 133   -2.2090 0.0272
##      wolf denslocalr.s  -0.6418   0.0717 133   -8.9552 0.0000
##      wolf wild_boar     0.0111   0.0048 133    2.2867 0.0222
##      fox      z.s      -0.2895   0.0366 132   -7.9154 0.0000
##      fox      tri5.s     0.0658   0.0326 132    2.0213 0.0432

```

##	fox	wolf	0.0508	0.0068	132	7.4469	0.0000
##	fox	lynx	0.0323	0.0089	132	3.6128	0.0003
##	fox	roe_deer	0.0434	0.0057	132	7.6274	0.0000
##	wild_cat	clc_forest.s	0.2226	0.1015	134	2.1933	0.0283
##	wild_cat	z.s	-0.4669	0.0988	134	-4.7271	0.0000
##	wild_cat	lynx	0.0595	0.0233	134	2.5512	0.0107
##	hare	tri5.s	-0.5444	0.1488	135	-3.6591	0.0003
##	hare	z.s	0.4595	0.118	135	3.8932	0.0001
##	red_deer	clc_forest.s	0.1621	0.0403	134	4.0218	0.0001
##	red_deer	denslocalr.s	-0.5309	0.0414	134	-12.8313	0.0000
##	red_deer	tri5.s	-0.1188	0.0361	134	-3.2891	0.0010
##	roe_deer	clc_forest.s	-0.1187	0.0406	134	-2.9263	0.0034
##	roe_deer	z.s	-0.4862	0.0544	134	-8.9396	0.0000
##	roe_deer	tri5.s	-0.3383	0.0565	134	-5.9920	0.0000
##	wild_boar	clc_forest.s	-0.1592	0.0302	133	-5.2681	0.0000
##	wild_boar	denslocalr.s	-0.2643	0.0398	133	-6.6402	0.0000
##	wild_boar	z.s	-0.2510	0.039	133	-6.4367	0.0000
##	wild_boar	tri5.s	-0.3137	0.0427	133	-7.3475	0.0000
##	~~lynx	~~wolf	0.2824	-	138	3.4205	0.0004
##	~~wild_boar	~~hare	0.2349	-	138	2.8084	0.0029
##	~~roe_deer	~~hare	0.2040	-	138	2.4206	0.0084
##	~~red_deer	~~hare	0.1180	-	138	1.3806	0.0848
##	~~wild_boar	~~red_deer	0.4154	-	138	5.3057	0.0000
##	~~red_deer	~~roe_deer	0.1588	-	138	1.8682	0.0319
##	~~wild_boar	~~roe_deer	0.3163	-	138	3.8733	0.0001
##	~~wild_cat	~~fox	0.1353	-	138	1.5862	0.0575
##	~~red_deer	~~lynx	0.1764	-	138	2.0820	0.0196
##	~~wolf	~~hare	0.1923	-	138	2.2765	0.0122
##	~~wild_cat	~~wild_boar	0.2596	-	138	3.1236	0.0011
##	~~wild_cat	~~roe_deer	0.1010	-	138	1.1799	0.1201
##	~~wild_cat	~~red_deer	0.1534	-	138	1.8039	0.0367
##	~~lynx	~~wild_boar	0.1741	-	138	2.0539	0.0210
##	Std.Estimate						
##	-0.1274	**					
##	0.0956	**					
##	0.1063	**					
##	-0.0834	*					
##	-0.3475	***					
##	0.0602	*					
##	-0.3581	***					
##	0.0814	*					
##	0.2166	***					
##	0.1288	***					
##	0.2512	***					
##	0.0923	*					
##	-0.1935	***					
##	0.0795	*					
##	-0.1642	***					
##	0.1386	***					
##	0.0764	***					
##	-0.2503	***					
##	-0.0560	**					
##	-0.0940	**					
##	-0.3848	***					

```

##      -0.2677 ***
##      -0.0925 ***
##      -0.1537 ***
##      -0.1459 ***
##      -0.1824 ***
##      0.2824 ***
##      0.2349 **
##      0.2040 **
##      0.1180
##      0.4154 ***
##      0.1588 *
##      0.3163 ***
##      0.1353
##      0.1764 *
##      0.1923 *
##      0.2596 **
##      0.1010
##      0.1534 *
##      0.1741 *
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
##
## ---
## Individual R-squared:
##
##      Response      method R.squared
##      lynx nagelkerke      0.15
##      wolf nagelkerke      0.56
##      fox nagelkerke      0.85
##      wild_cat nagelkerke    0.23
##      hare nagelkerke      0.20
##      red_deer nagelkerke    0.86
##      roe_deer nagelkerke    0.76
##      wild_boar nagelkerke    0.85

```

Based on the Chi-squared and fisher's C the combined model fits the data best