Heatmap for patterns of association in log-linear models

Below I provide code and step-by-step explanations to produce a heatmap for log odd-ratios from log-linear models. I exemplify the implementation in `R` using data on intergenerational class mobility for England, France and Sweden.

Steps:

1. Load the following packages for data manipulation (tidyverse,modelr,reshape2), log-linear model estimation (vcdExtra,logmult) and ploting (cowplot). Install previously if you do not have them.

```
library("tidyverse")
library("modelr")
library("reshape2")
library("vcdExtra")
library("logmult")
library("cowplot")
```

2. Input the contingency table an turn it into a data frame. I use the dataset `erikson` from the package `gnm`, a dependency of package `logmult`. This is a cross-classification of subject's occupational status (destination) and his father's occupational status (origin) across 3 countries.

```
# Inpute data and create contingency table as data.frame()
data(erikson)
table <- ftable(erikson)
mydata <- as.data.frame(table)
levels(mydata$country) <- c("England-Wales","France","Sweden")

# Save levels variables. To be used later.
levels.origin      <- levels(mydata$origin)
levels.destination <- levels(mydata$destination)
levels.country     <- levels(mydata$country)
```

This is what the data looks like:

```
## # A tibble: 243 x 4
##    origin destination country        Freq
##    <fct>  <fct>       <fct>         <dbl>
##  1 I      I           England-Wales   311
##  2 II     I           England-Wales   161
##  3 III    I           England-Wales   128
##  4 IVa    I           England-Wales    88
##  5 IVb    I           England-Wales    36
##  6 IVc    I           England-Wales    43
##  7 V/VI   I           England-Wales   356
##  8 VIIa   I           England-Wales   150
```

```
## 9 VIIb   I          England-Wales    12
## 10 I      II         England-Wales   130
## # … with 233 more rows
```

3. Next, I set the values to be used as reference categories.

```
    # Set reference categories
    mydata$origin      <- relevel(mydata$origin, ref = "V/VI")
    mydata$destination <- relevel(mydata$destination, ref = "V/VI")
    mydata$country     <- relevel(mydata$country, ref = "England-Wales")
    mydata$Freq        <- mydata$Freq + 1 # add small constant to avoid probl
ems with empty cells.
```

4. Fit different model specifications. Some of these modes are log-linear and other are log-multiplicative.

```
# Fit models

# independence
indep <- gnm(Freq ~ (origin + destination)*country, family = poisson, data =
mydata)

# quasi-perfect mobility
qpm  <- gnm(Freq ~ (origin + destination)*country + Diag(origin, destination)
*country, family = poisson, data = mydata)

# row-column association 1
rc1 <- gnm(Freq ~ (origin + destination)*country + Mult(origin, destination)
+ Diag(origin, destination)*country, family = poisson, data = mydata)

## Initialising
## Running start-up iterations..
## Running main iterations.....................................................
......
## ..............................................
## Done

# quasi-symmetry
qsymm <- gnm(Freq ~ (origin + destination)*country + Symm(origin, destination
)*country, family = poisson, data = mydata)

# unidiff or log-multiplicative layers
unidiff <- gnm(Freq ~ (origin + destination)*country + Mult(Exp(country), ori
gin:destination), family = poisson, data = mydata)

## Initialising
## Running start-up iterations..
```

```
## Running main iterations........
## Done
```

```r
# saturated
sat <- gnm(Freq ~ origin*destination*country, family = poisson, data = mydata
)


# Compare models via godness of fit statistics

models <- glmlist(indep,qpm,rc1,qsymm,unidiff,sat)
LRstats(models)
```

```
## Likelihood summary table:
##            AIC    BIC LR Chisq  Df Pr(>Chisq)
## indep   6498.4 6676.5   5152.6 192  < 2.2e-16 ***
## qpm     3130.9 3403.4   1731.1 165  < 2.2e-16 ***
## rc1     1973.4 2298.3    543.7 150  < 2.2e-16 ***
## qsymm   1689.1 2244.5    127.3  84   0.001605 **
## unidiff 1649.0 2057.7    171.2 126   0.004580 **
## sat     1729.8 2578.6      0.0   0   1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5. Goodness of fit statistics suggest that the `unidiff` models is the one that better fits the data. At this point I compute prediction from this model and from them, log odd ratios.

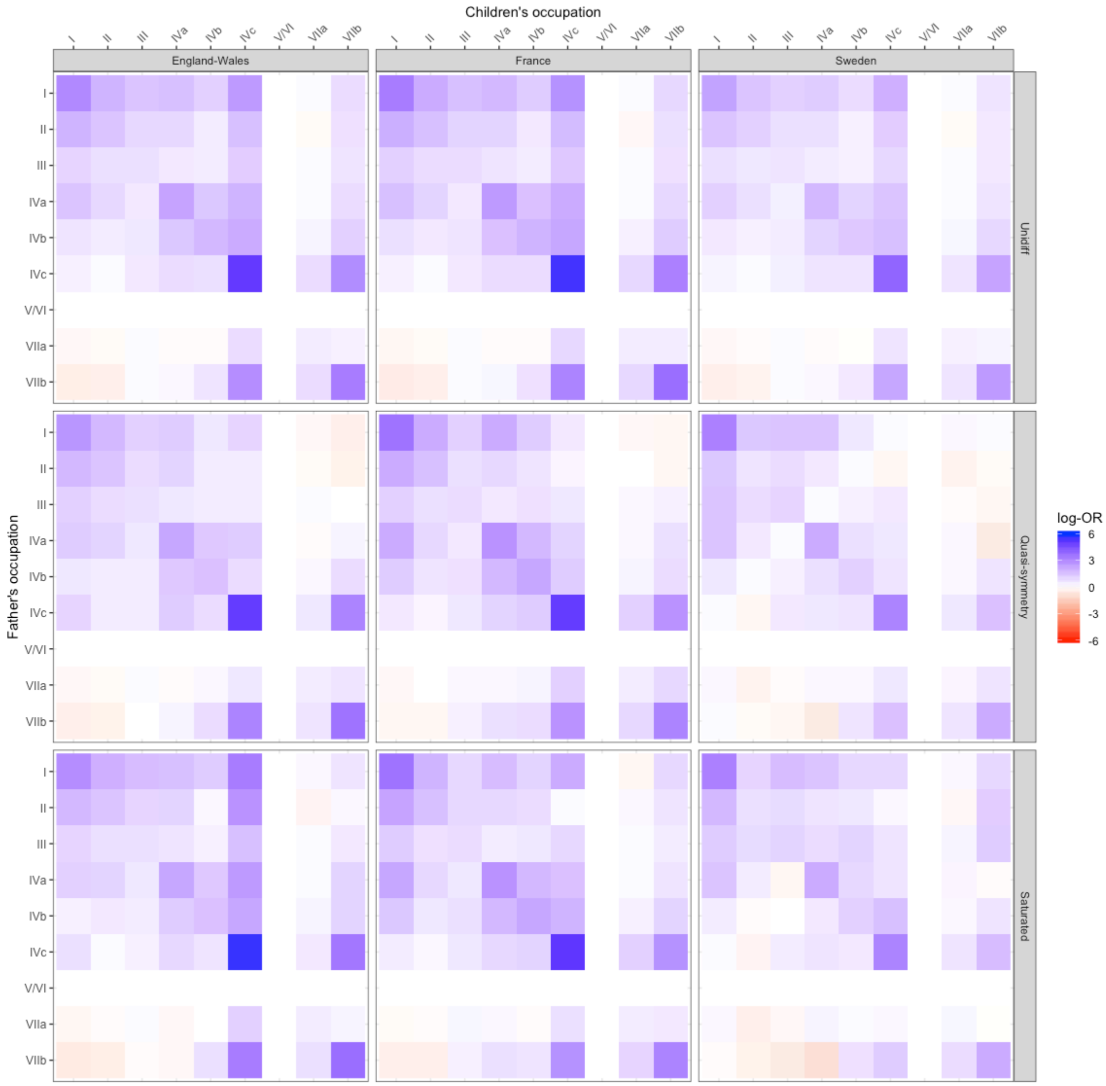
```r
# Create a synthetic dataset with all possible combinations of values

dummy.model <- lm(Freq ~ origin + destination + country, data=mydata)
new_x <- mydata %>% data_grid(origin,destination,country,.model=dummy.model)

# Compute predictions from different models. In this case: unidiff, quasi-sym
metry and saturated model.

for ( m in c("unidiff","qsymm","sat")) {

  chosen_model <- eval(parse(text = m ))

  # Predicted counts
  predictions <- cbind(mydata%>% data_grid(origin,destination,country,.model=
dummy.model), pred = predict(chosen_model, newdata=new_x)) %>%
    as_tibble()


  # Intercept
  intercept <- predictions %>% filter(origin=="V/VI", destination=="V/VI", co
untry=="England-Wales") %>% dplyr::summarise(pred) %>% as.numeric()
```

```r
  # Log odd ratios for marginal distributions
  predictions <- predictions %>% mutate(pred = pred - intercept) # remove int
ercept

  predictions_country     <- predictions %>% filter(origin=="V/VI", destinati
on=="V/VI")  %>% rename(margin_country=pred) %>% select(country,margin_countr
y)
  predictions_origin      <- predictions %>% filter(country=="England-Wales",
destination=="V/VI") %>% rename(margin_origin=pred) %>% select(origin,margin_
origin)
  predictions_destination <- predictions %>% filter(country=="England-Wales",
origin=="V/VI") %>% rename(margin_destination=pred) %>% select(destination,ma
rgin_destination)


  # match
  predictions <- predictions %>% left_join(predictions_country, by="country")
  predictions <- predictions %>% left_join(predictions_origin, by="origin")
  predictions <- predictions %>% left_join(predictions_destination, by="desti
nation")


  # Log odd ratios for marginal distributions origin and destination by count
ry

  predictions_country_origin <- predictions %>% filter(origin!="V/VI",country
!="England-Wales",destination=="V/VI") %>%
    rename(margin_country_origin=pred) %>% mutate(margin_country_origin = mar
gin_country_origin - (margin_country + margin_origin )) %>%
    select(country,origin,margin_country_origin)

  predictions_country_destination <- predictions %>% filter(origin=="V/VI",co
untry!="England-Wales",destination!="V/VI") %>%
    rename(margin_country_destination=pred) %>% mutate(margin_country_destina
tion = margin_country_destination - (margin_country + margin_destination )) %
>%
    select(country,destination,margin_country_destination)

  predictions <- predictions %>% left_join(predictions_country_origin, by=c("
country","origin")) %>%  replace_na(list(margin_country_origin = 0))
  predictions <- predictions %>% left_join(predictions_country_destination, b
y=c("country","destination")) %>%  replace_na(list(margin_country_destination
= 0))


  # Margin-free log-odd ratios (LORs)
  predictions <- predictions %>%
```

```
    mutate(`log-OR` = pred - (margin_country + margin_origin + margin_destina
tion + margin_country_origin + margin_country_destination) )


  # Save predictions
  assign(paste0("predictions_",m),predictions)

}
```

6. Finally, for each model I visualize the estimated log odd ratios capturing margin-free association between origin and destination across countries . Of course, other quantities can also be visualized in the same way.

```
# Combine models

predictions_unidiff <- predictions_unidiff %>% mutate(model = "Unidiff")
predictions_qsymm <- predictions_qsymm %>% mutate(model = "Quasi-symmetry")
predictions_sat <- predictions_sat %>% mutate(model = "Saturated")

predictions <- bind_rows(predictions_unidiff,predictions_qsymm,predictions_sa
t) %>%
  mutate(model = factor(model, levels=c("Unidiff","Quasi-symmetry","Saturated
")))


# Plot

plot <- predictions %>%
  ggplot(aes(y=factor(origin, levels = rev(levels.origin)),
             x=factor(destination, levels = levels.destination))) + facet_gri
d(model ~ country) + geom_raster(aes(fill= `log-OR`)) +
  scale_fill_gradientn(limits=c(-6,6), colours=c("red","white","blue")) +
  labs(y="Father's occupation", x= "Children's occupation", colour="") +
  theme_bw() + theme(axis.text.x = element_text(size=9, angle=45, vjust=-1, h
just=0),
                     axis.text.y = element_text(size=9, angle=0),
                     plot.title= element_text(size=11)) +
  scale_x_discrete(position="top")


# Add labels
plot <- plot %>% add_sub(.,"I+II: Service class, III: Routine non-manual empl
oyees, IVa+b:Petty bourgeoisie, IVc: Farmers, V/VI: Skilled working class, VI
Ia: Semi and unskilled working class, VIIb: Agricultural workers", size= 9) %
>%  ggdraw()

print(plot)
```



I+II: Service class, III: Routine non-manual employees, IVa+b:Petty bourgeoisie, IVc: Farmers, V/VI: Skilled working class, VIIa: Semi and unskilled working class, VIIb: Agricultural workers