Case study: null subjects

Gemma Hunter McCarley

September 2023

Contents

1 Introduction

The null subjects case study analyzed data from the (not yet published) CorDELES corpus. Samples pulled from the texts listed in "CorDELES Texts & Sources" were transcribed, parsed by the Stanford Parser (https://nlp.stanford.edu/software/spanish-faq.html), annotated by hand in xml, and then exported to csv. The datasets used in this study consist of the following columns:

- cordeles_2023.csv (each pronoun is a data point)
  - **docID**: the unique ID for each document made up of country + century + title, e.g. *dr16ent* = Dominican Republic, 16th century, *Entrémes*.

  - **sentenceID**: the sentence number automated by the Stanford Parser.

  - **sub_dep**: a marker for subjecthood, all are **nsubj** (a remnant of the universal dependencies used by the parser).

  - **subid**: the token ID of the subject.

  - **sub_word**: the lexical subject. In the case of multiple token subjects, a single token within the noun phrase was taken to represent the subject as a whole.

  - **sub_POS**: the part-of-speech of the subject. In this dataset, **sub_POS** can either be **NULL** or **OVERT**.

  - **Title**: full (or shortened, in the case of very long titles) text title.

  - **Region**: broader geographical grouping than country consisting of three levels (**Caribbean**, **South America**, **Peninsular (Spain)**).

  - **Country**: the country the text is from. For the most part this is also the country the author is from or was predominantly raised in; however, the earliest texts in the 16th century are naturally Spain-born authors who settled for a significant period in the specified country.

  - **Century**: the century the text was written in.

  - **Year**: the year the text was written in. Some texts only had a range for year, in which case the average was taken.

  - **Macro_Region**: the broadest geographical grouping consisting of two levels (**Spain** or **Non-Spain**).

  - **ORSCORE**: the degree of orality calculated for each text.

  - **Genre**: the initial binary genre split between literary (**LIT**) and non-literary (**DOC**) texts, plus a tag for supplemental texts (**SUPP**).

- orality.csv (each text is a data point)
  - **docID**, **ORSCORE**, and **Country** are the same as above.

  - **OVERT_RATE**: the proportion of overt subject pronouns per text.

  - **orality**: a categorical ranking of **ORSCOREs** split into **LOW** (0.00-0.25), **MID** (0.26-0.50), **HIGH** (0.51-0.75), and **HUGE** (1.00+). There isn't a tag for **ORSCOREs** between 0.76 and 1.00 as no text falls in that range.

The code in the following sections was run using R version 4.3.0.

2 Load dependencies and data

```
library(tidyverse)
library(lme4)
```

```r
library(DHARMa)
library(car)
library(ggsci)
library(gridExtra)

mydata <- read.csv("cordeles_2023.csv", header = TRUE, encoding = "UTF-8")
orality <- read.csv("orality.csv", header = TRUE, encoding = "UTF-8")

# get rid of duplicate data (translation of another text)
orality <- subset(orality, docID != "CPMTpoSP")
```

3 Plots and models

3.1 Orality plot and regression

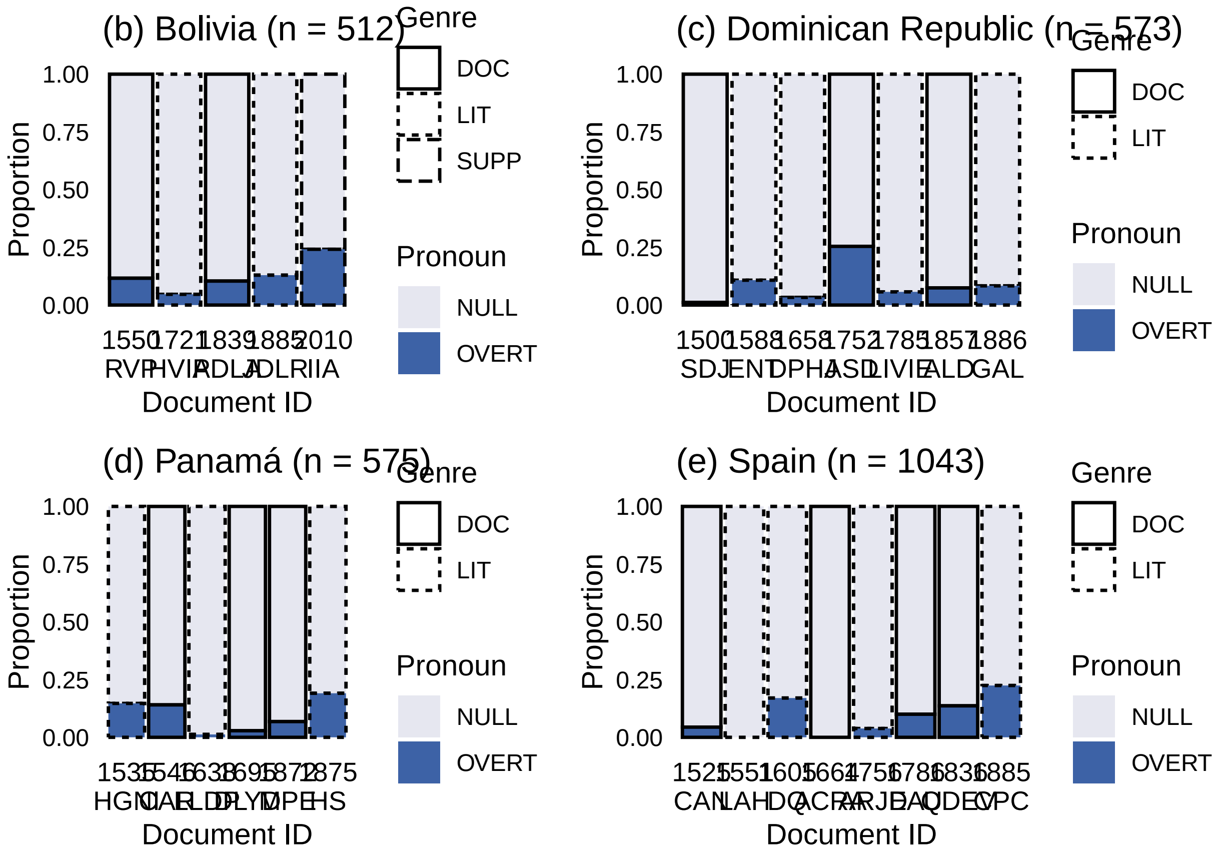
```r
# plot
orality$Country <- ifelse(orality$country == "DR", "Dominican Republic", orality$country)
orality$Country <- factor(orality$Country)
orality_plot_regression <- ggplot(data = orality, aes(x = ORSCORE, y = OVERT_RATE)) +
    geom_smooth(method = "lm", se = FALSE, color = "black") + labs(x = "ORSCORE",
    y = "Proportion of overt pronouns") + geom_point(size = 3, aes(pch = Country,
    color = Country)) + theme_bw() + scale_color_npg()

# save to external file
ggsave("Figure2.pdf", plot = orality_plot_regression, height = 3.5, width = 6)


# model
xmdl <- lm(OVERT_RATE ~ ORSCORE, data = orality)
summary(xmdl)
```

```
##
## Call:
## lm(formula = OVERT_RATE ~ ORSCORE, data = orality)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08527 -0.05271 -0.01067  0.03651  0.18698
```

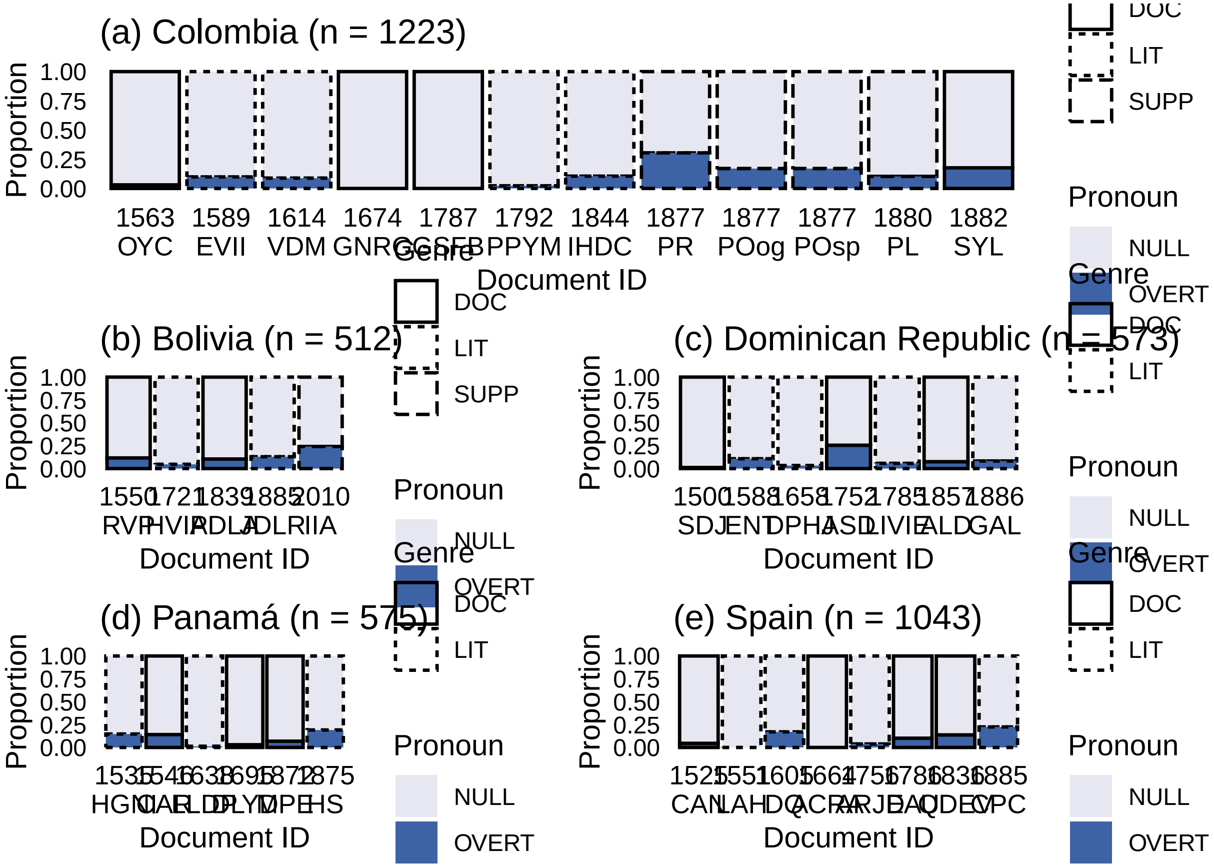
```
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.05016    0.01537   3.263 0.002465 ** 
## ORSCORE      0.10030    0.02307   4.348 0.000113 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.06403 on 35 degrees of freedom
## Multiple R-squared:  0.3507, Adjusted R-squared:  0.3322 
## F-statistic: 18.91 on 1 and 35 DF,  p-value: 0.0001128
```

3.2 Pronoun proportions plot

```
# Pronoun Plots####

pronoun_plot <- function(mydata, country, panelID, type = "regular") {
    # make country specific (country, re-leveling, and year labels):
    data <- filter(mydata, Country == country)

    if (country == "Colombia") {
        data$docID <- factor(data$docID, levels = c("co16oyc", "co16evii", "co17vdm",
            "co17gnrg", "co18gsfb", "co18ppym", "co19ihdc", "CPMTpr", "CPMTpoOG",
            "CPMTpoSP", "CPMTpl", "co19syl"))
        docs <- c("1563\nOYC", "1589\nEVII", "1614\nVDM", "1674\nGNRG", "1787\nGSFB",
            "1792\nPPYM", "1844\nIHDC", "1877\nPR", "1877\nPOog", "1877\nPOsp", "1880\nPL",
            "1882\nSYL")
    } else if (country == "Dominican Republic") {
        data$docID <- factor(data$docID, levels = c("dr16sdj", "dr16ent", "dr17dphj",
            "dr18asd", "dr18livie", "dr19ald", "dr19gal"))
        docs <- c("1500\nSDJ", "1588\nENT", "1658\nDPHJ", "1752\nASD", "1785\nLIVIE",
            "1857\nALD", "1886\nGAL")
    } else if (country == "Bolivia") {
        data$docID <- factor(data$docID, levels = c("bo16rvp", "bo18hvip", "bo19adla",
            "bo19jdlr", "bo21iia"))
        docs <- c("1550\nRVP", "1721\nHVIP", "1839\nADLA", "1885\nJDLR", "2010\nIIA")
    } else if (country == "Panamá") {
        data$docID <- factor(data$docID, levels = c("pa16hgni", "pa16car", "pa17lldp",
            "pa17dlyd", "pa19mpe", "pa19hs"))
        docs <- c("1535\nHGNI", "1546\nCAR", "1638\nLLDP", "1695\nDLYD", "1872\nMPE",
            "1875\nHS")
    } else if (country == "Spain") {
        data$docID <- factor(data$docID, levels = c("sp16can", "sp16lah", "sp17dq",
            "sp17acra", "sp18arjd", "sp18eau", "sp19qdev", "sp19cpc"))
        docs <- c("1525\nCAN", "1551\nLAH", "1605\nDQ", "1664\nACRA", "1756\nARJD",
            "1786\nEAU", "1836\nQDEV", "1885\nCPC")
    }

    # plot (repeat for each country)
    barchart <- data %>%
        ggplot(aes(x = docID, fill = sub_POS)) + scale_x_discrete(labels = docs) +
        geom_bar(position = "fill", aes(color = Genre, lty = Genre), linewidth = 0.6) +
        ylab("Proportion") + xlab("Document ID") + guides(fill = guide_legend(title = "Pronoun"),
        color = guide_legend(override.aes = list(fill = NA), title = "Genre")) +
        theme_minimal() + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

    barchart <- barchart + ggtitle(paste0("(", panelID, ") ", country, " (n = ",
        nrow(data), ")"))

    if (type == "supp") {
        # barchart <- barchart + scale_fill_manual(values = c('NULL' =
        # '#7294D4', 'OVERT' = '#C6CDF7')) + scale_color_manual(values =
        # c('DOC' = 'black', 'LIT' = 'white', 'SUPP' = 'gold')) +
        # theme(axis.text.x = element_text(size = 9, color = 'black')) barchart
        # <- barchart + scale_fill_manual(values = c('OVERT' = '#3d62a6',
        # 'NULL' = '#dfe2f7')) + scale_color_manual(values = c('DOC' = 'black',
        # 'LIT' = 'black', 'SUPP' = 'black')) + theme(axis.text.x =
        # element_text(size = 10, color = 'black'),
        # axis.text.y=element_text(color='black'))
        barchart <- barchart + scale_fill_manual(values = c(OVERT = "#3d62a6", `NULL` = "#e6e7f0")) +
            scale_color_manual(values = c(DOC = "black", LIT = "black", SUPP = "black")) +
            theme(axis.text.x = element_text(size = 10, color = "black"), axis.text.y = element_text(color = "black"))
        # supp genre ^^^
    } else if (type == "regular") {
        barchart <- barchart + scale_fill_manual(values = c(`NULL` = "#7294D4", OVERT = "#C6CDF7")) +
            scale_color_manual(values = c(DOC = "black", LIT = "white")) + theme(axis.text.x = element_text(size = 10,
            color = "black"), axis.text.y = element_text(color = "black"))
        # regular ^^^
    }

    barchart
}

mydata$Country <- ifelse(mydata$Country == "DR", "Dominican Republic", mydata$Country)
mydata$Country <- factor(mydata$Country)

pp1 <- pronoun_plot(mydata, country = "Colombia", type = "supp", panelID = "a")
pp2 <- pronoun_plot(mydata, country = "Bolivia", type = "supp", panelID = "b")
pp3 <- pronoun_plot(mydata, country = "Dominican Republic", type = "supp", panelID = "c")
pp4 <- pronoun_plot(mydata, country = "Panamá", type = "supp", panelID = "d")
pp5 <- pronoun_plot(mydata, country = "Spain", type = "supp", panelID = "e")

# combined plot
combined_plot <- grid.arrange(pp2, pp3, pp4, pp5, nrow = 2, ncol = 2, widths = c(0.85,
    1))
```



```
combined_plot <- grid.arrange(pp1, combined_plot, nrow = 2, ncol = 1, heights = c(1.1,
    2))
```



```
# save to external file
ggsave("Figure3.pdf", plot = combined_plot, height = 10, width = 9)
```

3.3 Main model

```
# Main Model#### Get data ready
binary_null <- within(mydata, Country <- relevel(factor(Country), ref = "Spain"))  #make Spain the reference level for Country
binary_null <- within(binary_null, Macro_Region <- relevel(factor(Macro_Region),
    ref = "Spain"))  #make Spain the reference level for Macro_Region

binary_null %>%
    select(Region, Genre, sub_POS, Country, Year, ORSCORE, docID, Century, Macro_Region) %>%
    na.omit() %>%
    mutate(Century = as.factor(Century)) -> binary_null

binary_null <- subset(binary_null, binary_null$docID != "CPMTpoSP")  #get rid of duplicate data (translation of another text)

# Model
binary_null_mdl_int <- glmer(factor(sub_POS) ~ scale(Year) * scale(ORSCORE) + Macro_Region +
    (1 | docID), data = binary_null, family = "binomial")
```

4 Tests

```
# tests
summary(binary_null_mdl_int)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: factor(sub_POS) ~ scale(Year) * scale(ORSCORE) + Macro_Region +
##     (1 | docID)
##    Data: binary_null
##
##      AIC      BIC   logLik deviance df.resid
##   2548.2   2585.6  -1268.1   2536.2     3767
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.5568 -0.4031 -0.2986 -0.2208  7.2692
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  docID  (Intercept) 0.3119   0.5585
## Number of obs: 3773, groups:  docID, 37
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.54816    0.27570  -9.242  < 2e-16 ***
## scale(Year)                0.05029    0.15364   0.327 0.743434
## scale(ORSCORE)             1.02970    0.26917   3.825 0.000131 ***
## Macro_RegionNon-Spain      0.62880    0.31997   1.965 0.049389 *
## scale(Year):scale(ORSCORE) -0.43702    0.20634  -2.118 0.034177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) scl(Y) s(ORSC M_RN-S
## scale(Year)  0.086
## sc(ORSCORE) -0.053 -0.577
## Mcr_RgnNn-S -0.831 -0.307  0.388
## s(Y):(ORSCO -0.180  0.356 -0.723 -0.170
```

```
drop1(binary_null_mdl_int, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## factor(sub_POS) ~ scale(Year) * scale(ORSCORE) + Macro_Region +
##     (1 | docID)
##                            npar    AIC    LRT Pr(Chi)
## <none>                          2548.2
## Macro_Region                  1 2550.0 3.8159 0.05077 .
## scale(Year):scale(ORSCORE)    1 2550.6 4.4147 0.03563 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

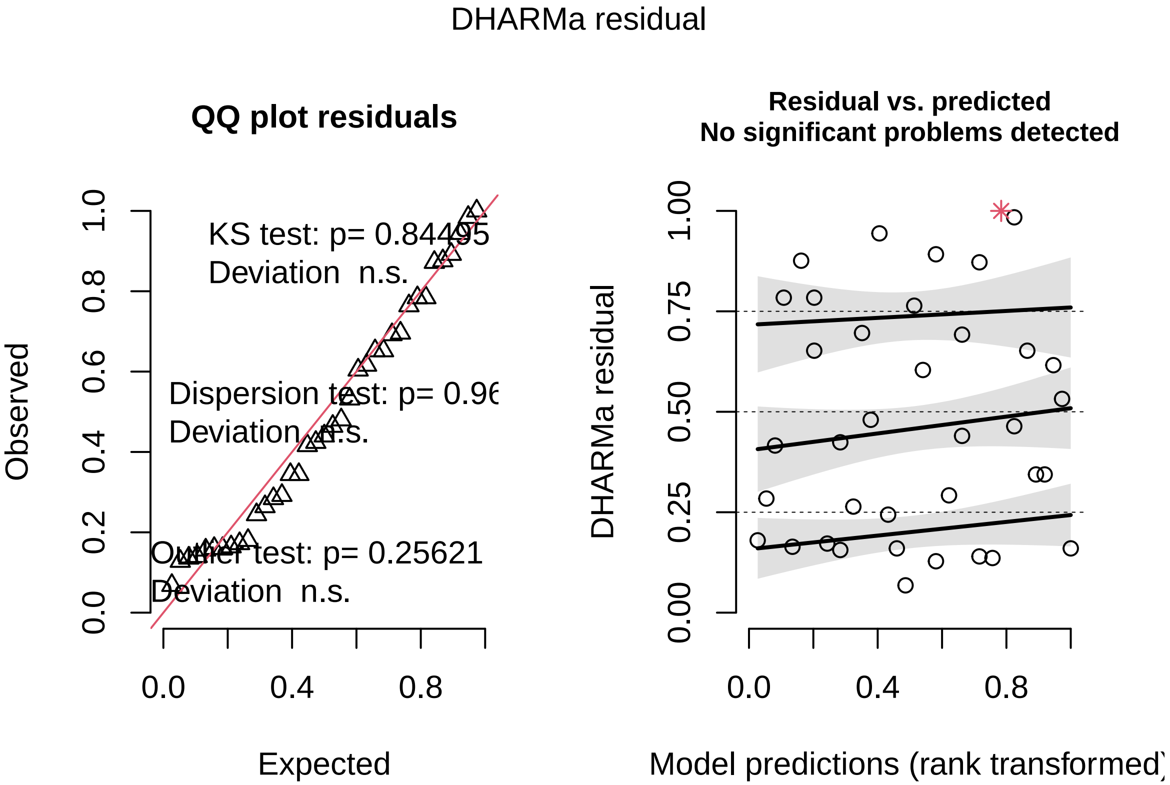
```
Anova(binary_null_mdl_int, type = "III")
```

```
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: factor(sub_POS)
##                              Chisq Df Pr(>Chisq)
## (Intercept)                85.4228  1  < 2.2e-16 ***
## scale(Year)                 0.1071  1  0.7434341
## scale(ORSCORE)             14.6342  1  0.0001305 ***
## Macro_Region                3.8621  1  0.0493893 *
## scale(Year):scale(ORSCORE)  4.4858  1  0.0341767 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

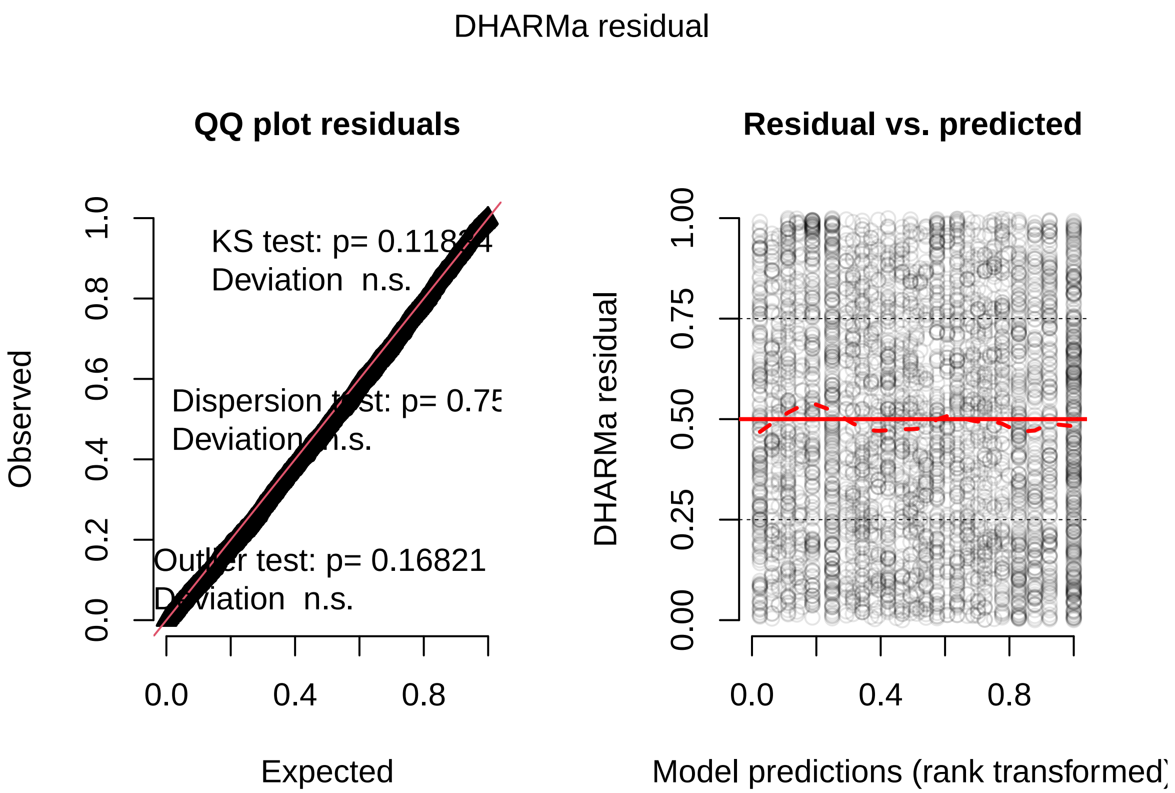
```
AIC(binary_null_mdl_int)
```

```
## [1] 2548.202
```

```
# checking residuals
res_orality <- simulateResiduals(xmdl, plot = T)
```



```
res <- simulateResiduals(binary_null_mdl_int, plot = T)
```


DHARMa residual
QQ plot residuals
KS test: p= 0.11824
Deviation n.s.
Dispersion test: p= 0.75
Deviation n.s.
Outlier test: p= 0.16821
Deviation n.s.
Observed
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.4
0.8
Expected
Residual vs. predicted
DHARMa residual
0.00
0.25
0.50
0.75
1.00
0.0
0.4
0.8
Model predictions (rank transformed)