

Third Fitting of Thermal Models

Michael Gilchrist

date: 2022-10-20

Goal

- Fit series of thermal models, including, ultimately, those found in `rTPC` to data collected in the Derryberry lab.

Recap

First and Second Fitting

- Temp is hard to control in chambers, so we should work with `temp_mean` (mean value during trial), not `temp_target`.
- `humidity_mean` and `temp_mean` strongly co-vary so consider using aggregate index as explanatory variable
 - Vapor Pressure Deficit (provided by `humidity` package)
 - Heat Index (formulated for humans, and provided by `weathermetrics`)
- Round 3 data only for curve fitting
 - Only round where `temp_mean` and `humidity_mean` exist.
- `count_total_round` are consistent between Rounds 1 and 3, so could use that info to classify birds, estimate variance function in response, etc.
- One bird in `round = 3` collapsed during the trial so it was terminated. We've set the `song_count` from NA to 0 and should consider making at `temp_working` column using `temp_target` in this instance and `temp_mean` in all other instances. Would need to do something similar for relative humidity, i.e. use `mean(humidity_mean)` for the `temp_target` value.

Third Fitting

- Created `temp` and `humidity` variables from `temp_mean` and `humidity_mean` for males who didn't collapse, used `mean(temp_mean)` and `mean(humidity_mean)` values for one male that did.
- Copied `data_full` and set `male = "combined"` so we could look at all of the data at once.
- Learned about using `model` argument for glm models
- Can't fit ME models
 - I don't understand why the ME model with `song_count` where we use a RE for the intercept doesn't have a similar effect as using `song_prop`. This intercept value is essentially multiplying by a constant, so we could try and specify this value using `offset(log(count_total_round))`. So why doesn't this work?

Fourth Fitting

Preliminary Notes

Which Explanatory Variable: temp, humidity, vpd?

- Liz will look into operative temp
- Use of `vpd_mean` is somewhat arbitrary since it depends on the experimental design.
 - Use of 0 reference in `vpd` seems less arbitrary than using 0 C.
- Note that we could scale `temp_mean` relative to $C = 45$, which is the understood thermal maximum.
 - This is what Liz wants to do since it's easier to interpret

Type of Model Fitted

- Use `temp_mean - temp_ref` where Don't worry about random effects for now
- Try the quadratic fit with just temperature and the 0 set at 45C.
 - Quadratic function – pull out the peak, the curvature at the peak, calculate the intercept at some temp – what is the intercept
- Begin fitting rTPC models to combined dataset.

Summary of Results

- Tried fitting lots of variations of `glm` and `glmer`
- Optimizer matters in terms of convergence.
- Adding fixed `male` term rather than treating it as RE via `(1|male)` was very important since the variation in overall male singing varies so greatly.
- poisson fits indicate data is greatly overdispersed.
- Got one particular form of GLMM (`glmer`) working well (i.e. when there's no covariance matrix). I think this is a valid approach, but am unsure and a bit concerned since the `|` fitting shows the RE for x_1 and x_1^2 being close to perfectly correlated.
- Overall results indicate that this approach is problematic
 - Data is extremely overdispersed in these fits.
 - As a result, no signal beyond a `male` effect on overall activity is reliably detected.

Future Steps

Revisit `lmList`

- are the estimates from individual models consistent with the parameters estimated with the `|` and `/or` `||` models? This is really only worth doing if
 - We think we want to ultimately use the quadratic `link = "log"` models
 - We want to make sure we correctly understand GLMM

Analyze with rTPC

According to Padfield et al. (2021)

However, [the rTPC] pipeline does not accommodate non-independent (related) replicates, and clustered or stratified sampling (possibly with missing values). In such situations, nonlinear mixed effects model fitting (e.g. using the `nlme` R package; Oddi et al., 2019) or Bayesian approaches (e.g. using the `brms` R package; Bürkner, 2017) would be more appropriate. Nevertheless, for fitting massive TPC datasets to multiple mathematical models, rTPC offers a simple, reliable and reproducible computational pipeline with robust methods for calculation of model uncertainty, requiring minimal statistical and computational expertise, and suitable for a wide range of applications.

Thus, we can't incorporate individual or random effects. We could try and use all of the observations in `song_prop` to deal with the `male` term.

Analyze with other packages

- i.e. use `stan` and/or `brms` which uses `stan`.

Including beak and mass data

- Include beak size (surface area) as additional explanatory variable. We do have bird mass as well
- Notes from Liz
 - Do birds with larger beaks maintain singing at higher rates at higher temps?
 - We calculated bill surface area approximately as the surface area of a cone: $\text{length} * \pi * (\text{width} + \text{depth}) / 4$.
 - Because larger animals produce more heat (Kleiber 1932), we scaled bill size relative to heat production by dividing bill surface area by expected daily energy consumption (`mass0.658`; Speakman and Król 2010, Hudson et al. 2013).
 - So, looks like we need to calculate bill surface area and then scale relative to heat production (`bill surface area/mass0.658`)

Using Data from Rounds 1 and 2

- Liz: Ask Kayci about `temp_mean` and `humidity_mean` data for rounds 1 and 2.
- Look for order effects in round 1 and 2? Can we use any of this data? Seems like there will be an issue if we throw out, say, first two trials, when using `song_prop`. Note that since `total_count` is consistent for a bird between rounds, this may not be an issue.
- Begin fitting rTPC models to combined dataset.
- We could include `round = 2`
 - Would need to down weight `song_count` values when combining across `count_total_round` values.

Additional Liz

- Follow up with Ray/Juan about ground versus surface temp for operative temperature.
- What is the 0 for operative temperature?
- Ponder utility of other ZF data on panting.

Set up

Install libraries

```
## install packages user might not have by replacing FALSE with TRUE
if(FALSE) {
  BiocManager::install("mixOmics") ## needed by RVAideMemoire
  install.packages(c("RSQLite", "nls.multstart", "lme4", "RVAideMemoire"))
  ## Install the thermal curve package from git_hub, not cran
  remotes::install_github("padpadpadpad/rTPC")
}
```

```
## load libraries
library(stats)
require(MASS) # provides negative binomial fitting: glm.nb
```

Loading required package: MASS

```
library(RSQLite) # Don't think we need this.
library(rTPC) ##
library(nls.multstart)
library(broom)
library(tidyverse)
```

-- Attaching packages

tidyverse 1.3.2 --

```
## v ggplot2 3.3.6          v purrr 0.3.4
## v tibble 3.1.8           v dplyr 1.0.99.9000
## v tidyr 1.2.0.9000       v stringr 1.4.1
## v readr 2.1.3            v forcats 0.5.2
```

-- Conflicts -----
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()
x dplyr::select() masks MASS::select()

```
require(ggplot2)
require(ggpubr)
```

Loading required package: ggpubr

```
require(viridisLite)
```

Loading required package: viridisLite

```
#options(ggplot2.continuous.colour="viridis",
#         ggplot2.discrete.colour="viridis",
#         ggplot2.scale_fill_discrete = scale_fill_viridis_d,
#         ggplot2.scale_fill_continuous = scale_fill_viridis_c)
```

```
require(GGally)
```

```
## Loading required package: GGally
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2
```

```
require(reshape2)
```

```
## Loading required package: reshape2
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
require(lme4)
```

```
## Loading required package: lme4
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
require(RVAideMemoire) # provides overdisp.glmr()
```

```
## Loading required package: RVAideMemoire
## *** Package RVAideMemoire v 0.9-81-2 ***
##
## Attaching package: 'RVAideMemoire'
##
## The following object is masked from 'package:lme4':
##
##     dummy
##
## The following object is masked from 'package:broom':
##
##     bootstrap
```

```
require(humidity) ## provides VPD
```

```
## Loading required package: humidity
```

```
require(weathermetrics)
```

```
## Loading required package: weathermetrics
```

```
require(latex2exp)
```

```
## Loading required package: latex2exp
```

Load Data

```
## Read in ZEFI Data sets
```

```
## Treat 'repeatability' as round = 0
```

```
## Add round info
```

```
## Repeatability was done between round 1 and 2, female was present, but only one temp. so treating as
```

```
git_root <- system("git rev-parse --show-toplevel", intern = TRUE)
```

```
data_raw = list()
```

```
data_raw[[1]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Round-1-Heat-Trials.csv")) %>%  
  ## Note T237 and T230 are missing numbers in the song_count column  
  ## so we are filtering these observations out until they are found  
  filter(!is.na(song_count))
```

```
data_raw[[2]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Repeatability-Song-Count.csv"))  
  mutate(round = 2) %>%  
  group_by(male) %>%  
  mutate(test_order = rank(date)) %>%  
  ungroup()
```

```
data_raw[[3]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Round-2-Heat-Trials.csv")) %>%  
  mutate(round = 3) %>%  
  ## Deal with missing temp_mean and humidity_mean values  
  ## in round == 3  
  ## 2022/10/19 - code no longer needed  
  ## group_by(temp_target) %>%  
  ##mutate(temp = if_else((round == 3 & is.na(temp_mean)),  
  ##                      mean(temp_mean, na.rm = TRUE),  
  ##                      temp_mean)) %>%  
  ##mutate(humidity = if_else((round == 3 & is.na(humidity_mean)),  
  ##                      mean(humidity_mean, na.rm = TRUE),  
  ##                      humidity_mean)) %>%  
  ungroup()
```

```
## Join data and discard empty columns
```

```
data_full <- full_join(data_raw[[1]], data_raw[[2]]) %>%  
  full_join(data_raw[[3]]) %>%  
  discard(~all(is.na(.) | . == "")) %>% ## get rid of columns of only NA
```

```

mutate(trial_completed = !(is.na(song_count)) ) %>%
mutate(song_count = ifelse(is.na(song_count), 0, song_count)) %>%
mutate(song_count = song_count*1.0) %>% ## convert to a double so it's not treated as an integer
mutate(chamber = as.factor(chamber), male = as.factor(male)) %>%
## create a global variable trial_order based on individual rounds
mutate(trial_index = as.integer(round*10+test_order)) %>%
mutate(song_count_plus_1 = (song_count + 1)) %>%
mutate(log_song_count_plus_1 = log(song_count + 1)) %>%
mutate(temp_target = as.numeric(temp_target)) %>%
## Add column with total song_count for a given round
group_by(male, round ) %>%
mutate(count_total_round = sum(song_count) ) %>%
ungroup() %>%
mutate(song_prop = song_count/count_total_round) %>%
## assuming poisson error
## From glm man page
## > Non-'NULL' 'weights' can be used to indicate that different
## > observations have different dispersions (with the values in
## > 'weights' being inversely proportional to the dispersions);
## add +1 to deal with single 0
mutate(count_wt = 1/(song_count + 1)) %>%
## need to rescale wts for song_prop data
mutate(prop_wt = count_wt * count_total_round^2) %>%
## Add vpd
mutate(svp = SVP(t = temp_mean + 273.15, isK = TRUE), vpd = svp*(1-humidity_mean/100) ) %>%
group_by(round) %>%
mutate(vpd_offset = vpd - mean(vpd)) %>%
ungroup() %>%
relocate(song_count, song_prop, vpd, temp_mean, humidity_mean, .after = male) %>%
mutate() ## Dummy function so we can comment out lines above it w/o any issues

```

```
## Joining, by = c("male", "chamber", "date", "song_count", "counter", "test_order", "temp_target", "round")
```

```
## Joining, by = c("male", "chamber", "date", "song_count", "counter", "test_order", "temp_target", "round")
```

Fourth Analysis

Examine Data

```

data_count_total <- data_full %>% group_by(round) %>%
  select(male, round, count_total_round) %>%
  distinct()

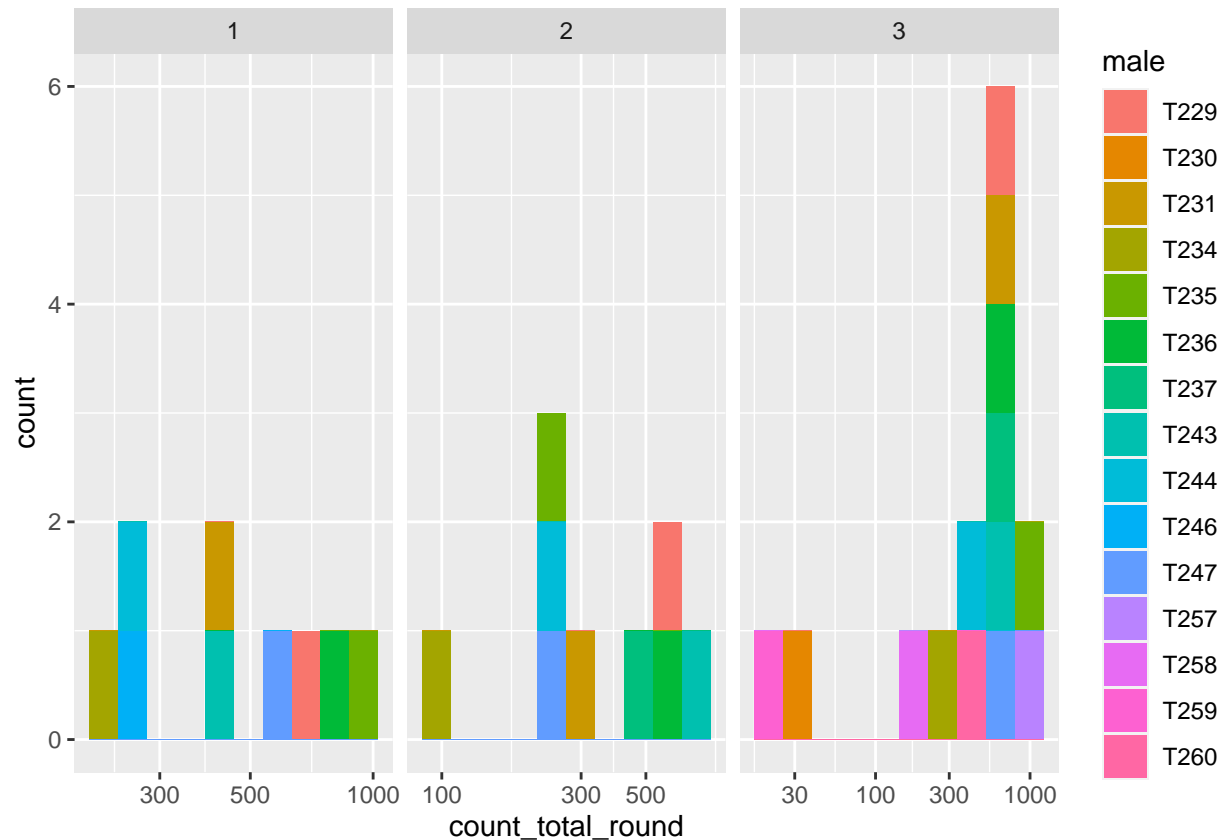
t <- ggplot(data_count_total, aes(count_total_round, fill = male)) +
  geom_histogram(bins = 10) +
  scale_x_log10()

hist_count_total <- t +
  facet_grid(cols = vars(round), scales = "free_x")
hist_count_total

```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

Warning: Removed 1 rows containing non-finite values (stat_bin).



Compare count_total_round between round 1 and 3

- See third.fitting.Rmd

Result

- As before, we see strong consistency between round 1 and 3.
- Consistency with round 2 is weaker, but sample sizes are smaller: 3 trials/male in round 2 vs 6 trials/male in round 3.

Create & Plot Filtered Data

```
data_ind <- data_full %>%
  filter(round==3) %>%
  filter(count_total_round >=150)
## copy data frame and assign `male = "combined"`
data_comb <- data_ind %>% mutate(male = "combined")

data <- bind_rows(data_ind, data_comb)

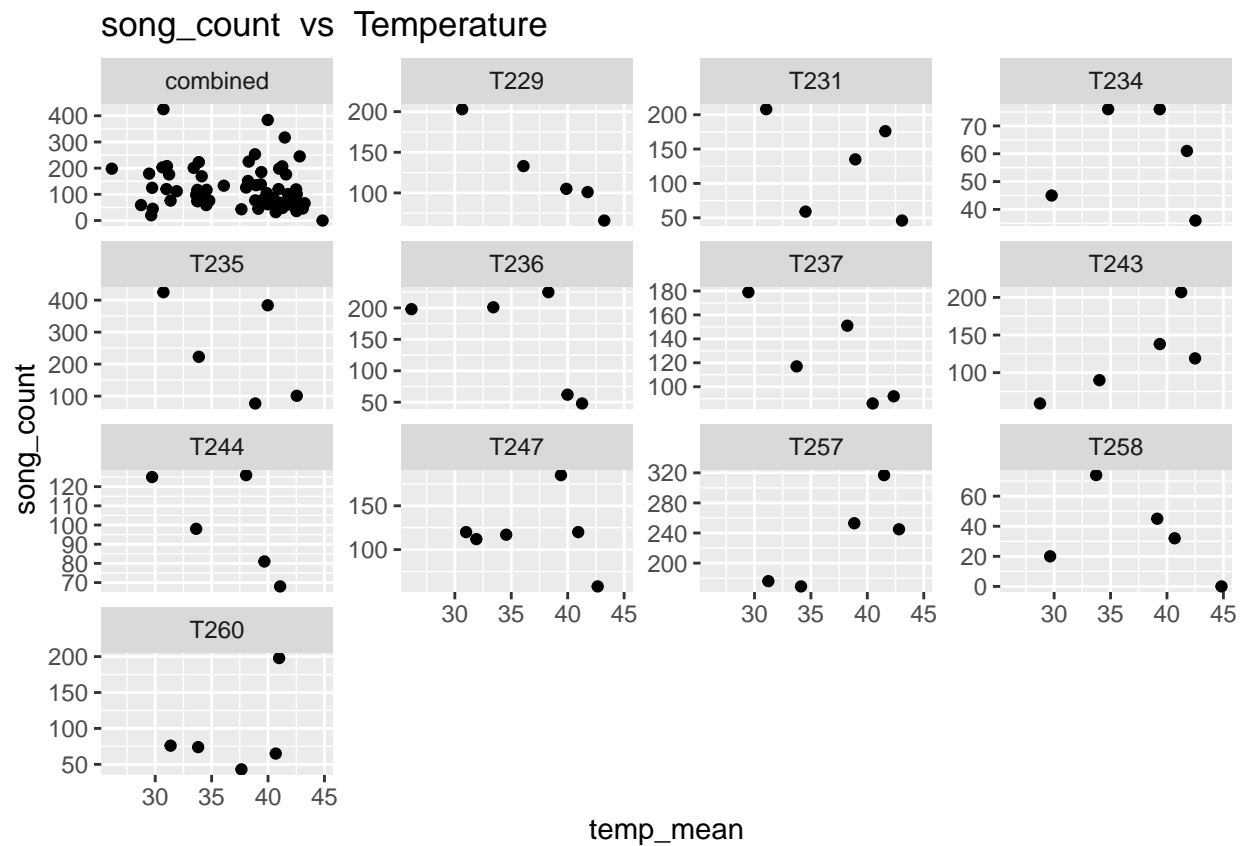
xlab <- "Temperature"
ylab <- "song_count"
```



```

plot_temp_data <-
  ggplot(data) +
    aes(x = temp_mean,
        y = song_count) +
    facet_wrap("male", scales = "free_y") +
    geom_point() +
    labs( title = paste( ylab, " vs ", xlab))
last_plot()

```



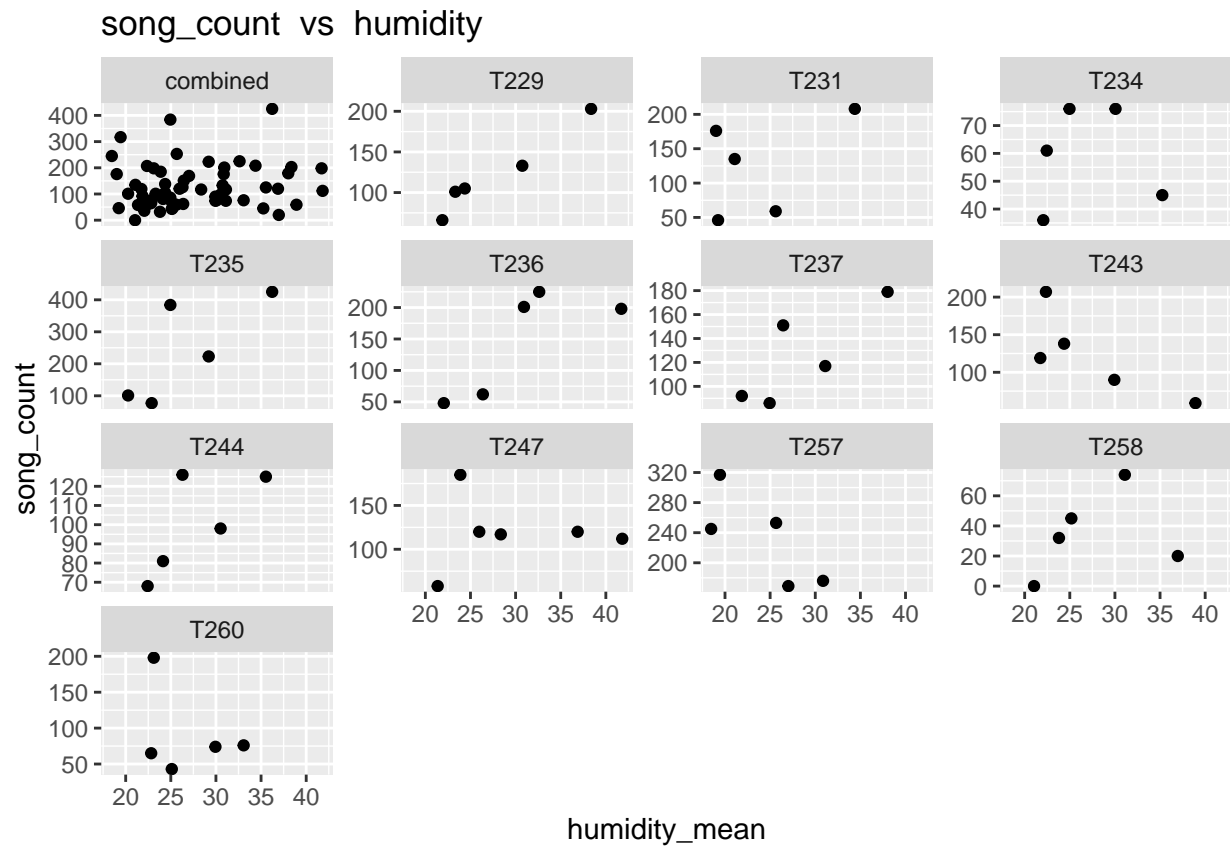
```

## Create again for humidity

xlab <- "humidity"

plot_humidity_data <-
  ggplot(data) +
    aes(x = humidity_mean,
        y = song_count) +
    facet_wrap("male", scales = "free_y") +
    geom_point() +
    labs( title = paste( ylab, " vs ", xlab))
last_plot()

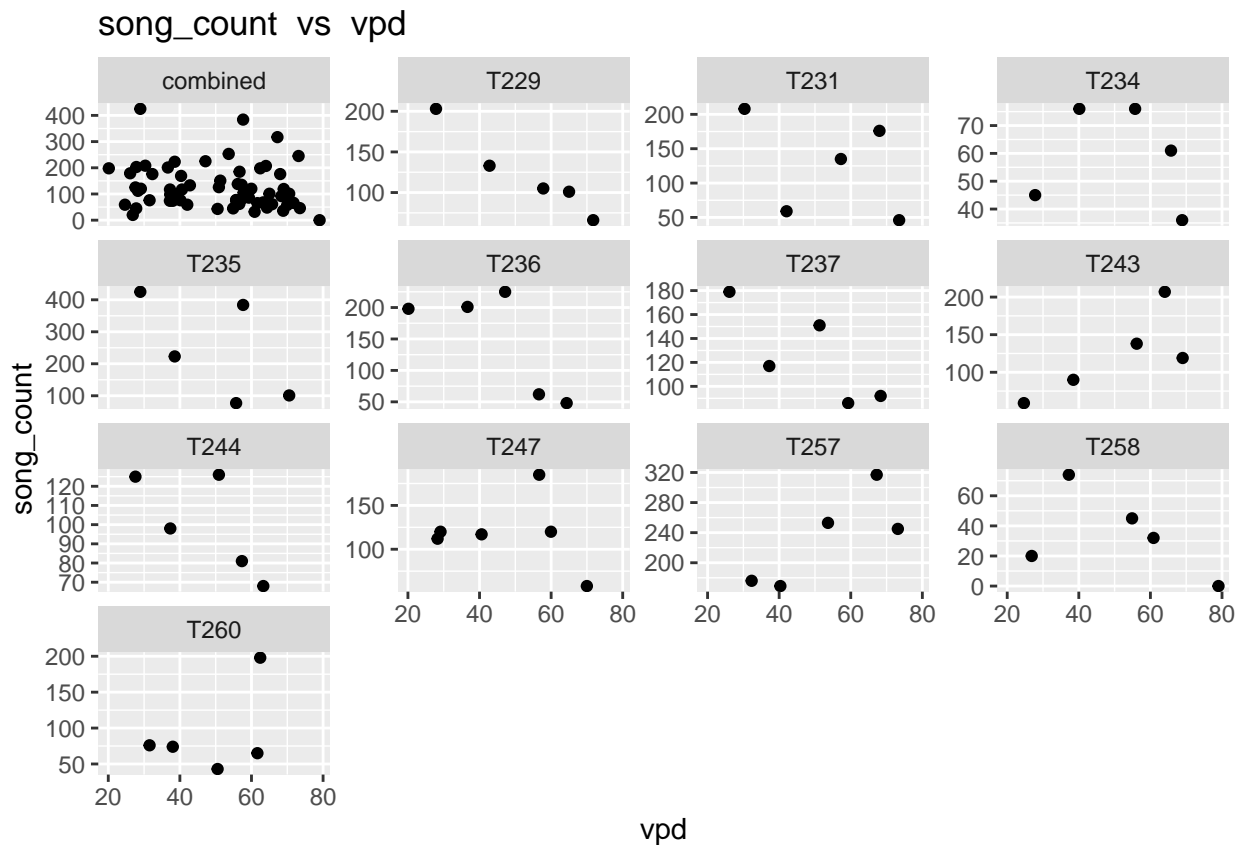
```



```
## Create for vpd

xlab <- "vpd"

plot_vpd_data <-
  ggplot(data) +
    aes(x = vpd,
        y = song_count) +
    facet_wrap("male", scales = "free_y") +
    geom_point() +
    labs(title = paste(ylab, " vs ", xlab))
last_plot()
```



Work with mean and song_prop

- This approach won't work because `temp_target` varies greatly from `temp_mean`.

```
plot_combine <- ggplot(data_ind) +
  aes(x = temp_mean, y = song_prop, color = male) +
  geom_point()

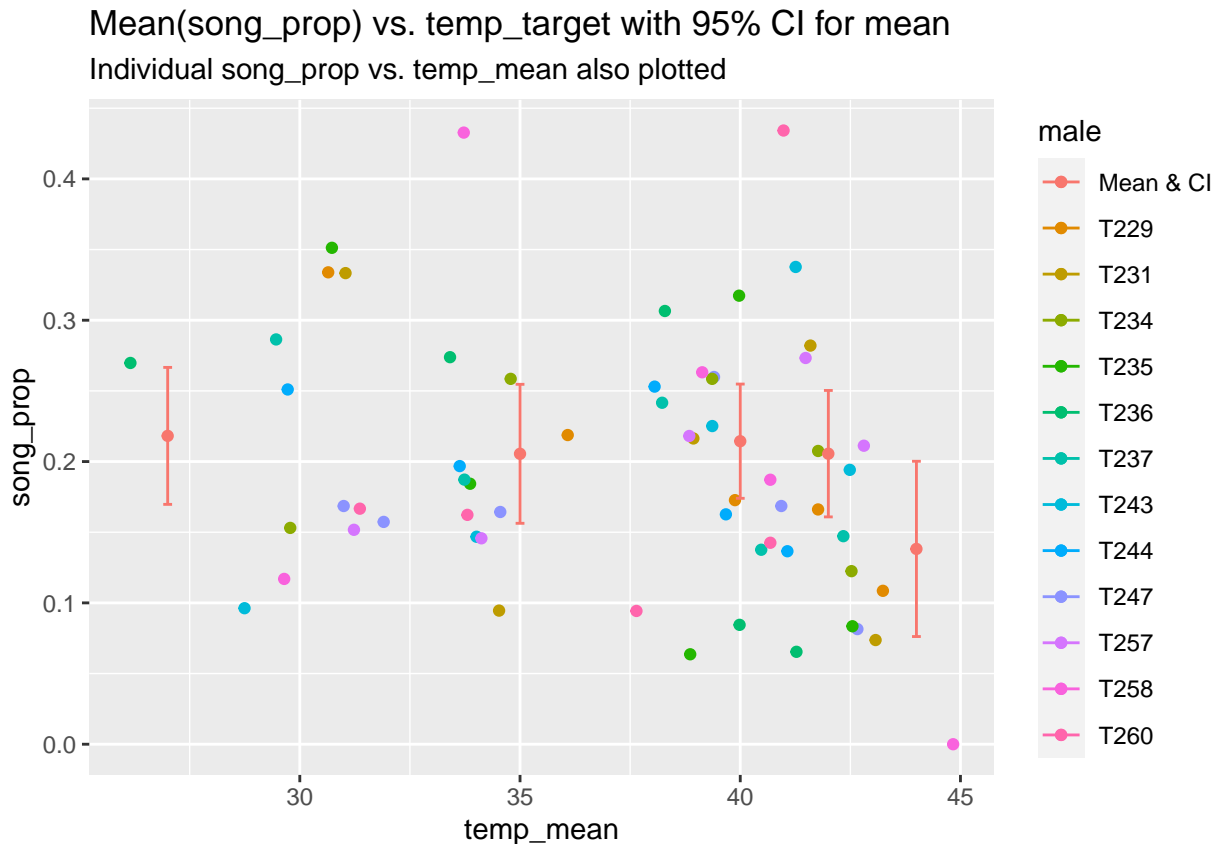
# last_plot()

data_summarize <-
  data_ind %>% group_by(temp_target) %>%
  summarize(song_mean = mean(song_prop),
            song_sd = sd(song_prop),
            song_n = dplyr::n(),
            song_se = song_sd/sqrt(song_n),
            song_ci = song_se*1.96
  )

plot_summarize <- ggplot(data_summarize) +
  aes(x = temp_target, y = song_mean) +
  geom_point()+
  geom_errorbar(aes(ymin=song_mean - song_ci, ymax=song_mean+song_ci), width=.2)

# last_plot()
```

```
plot_combine +
  geom_point(data = data_summarize, aes(x = temp_target, y = song_mean, colour = "Mean & CI"))+
  geom_errorbar(data = data_summarize, aes(x = temp_target,
    y = song_mean,
    ymin=song_mean - song_ci,
    ymax=song_mean+song_ci,
    colour= "Mean & CI"), width=.2) +
  labs(title = "Mean(song_prop) vs. temp_target with 95% CI for mean",
    subtitle = "Individual song_prop vs. temp_mean also plotted")
```



Result

- Data is noisy.
- Using `song_prop` reduces impact of male T225 on lower temps.
- Note `temp_target` is not necessarily accurate.

Humidity, Temp, and VPD

Previous Results from `second.fitting.pdf`

- Values clearly co-vary.
- Should consider using
 - Vapor Pressure Deficit
 - * Uses temp to calculate max humidity and then looks at difference with relative humidity.

- Heat Index
 - * Adjusted for birds if available
 - * Use just first terms ($c_1 + c_2 T + c_3 R + c_4 T R + \dots$)

Using VPD as Predictor

- In third.fitting we used VPD, but Liz suggested moving back to temp.

Compare Temp, Humidity, and VPD as predictors

- see third.fitting.Rmd

Result

- VPD and temp are *highly* correlated

Model Fitting

Plots

Formal Model Fits to song_count

Initial

Using

- `x = temp_mean - 45C`
- male as a factor
- Filter data

```
tem_ref <- 45
verbose <- 0
trace <- FALSE

## Try filtering the data a bit more
## Goal is to get good starting values

data <- data_ind %>%
  mutate(x1 = (temp_mean - temp_ref)) %>%
  filter( !(male %in% c("T231", "T260")))
```

```
## Error in 'mutate()':
## ! Problem while computing 'x1 = (temp_mean - temp_ref)'.
## Caused by error:
## ! object 'temp_ref' not found
```

```
make_plot = FALSE;

glm_poisson <- glm(song_count ~
  1 + male + (x1) + I(x1^2), # + offset(log(count_total_round)),
  data = data,
  family = poisson(link = "log")
)
```

```
## Error in eval(predvars, data, env): object 'x1' not found
```

```
summary(glm_poisson)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method for function 'summary'
```

```
if(make_plot) plot(glm_poisson, ask = FALSE)

glm_qpoisson <- update(glm_poisson,
  family = quasipoisson(link = "log")
)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method for function 'summary'
```

```
summary(glm_qpoisson)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method for function 'summary'
```

```
if(make_plot) {
  plot(glm_qpoisson, ask = FALSE)
}

## Clearly the data is over dispersed
## quasipoisson() doesn't seem to exist for glmer (but likely exists in nlme)

## Try using negative binomial

## This model formulation seems to be correct and converges!!
## However, it ignores the correlation between x1 and I(x1^2)
formula_quad_RE <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2)||male)

## This model doesn't converge due to high correlation mentioned above.
formula_quad_RE <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2)||male)
## Remove previous results to avoid confusion if model fails to fit
rm(glmer_poisson)
```

```
## Warning in rm(glmer_poisson): object 'glmer_poisson' not found
```

```
glmer_poisson <- glmer(formula_quad_RE,
  data = data,
  control = glmerControl(
    optCtrl = list(maxiter = 1E5,
                   maxfun = 2E6,
                   trace = trace),
    optimizer="nloptwrap"),
  family = poisson(link = "log"),
  verbose = verbose
)
```

```
## Error in eval(predvars, data, env): object 'x1' not found
```

```
summary(glmer_poisson)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method for function 'h'
```

```
if(make_plot) plot(glmer_poisson, ask = FALSE)

## Results indicate that var in x1^2 term is very small (1E-4)
## So we should drop that.
## Except the model complains about convergence

## Fit using `| male` instead of `|| male` to see what happens
## An extra term, the correlation between the RE terms is estimated.
## Random effects:
##   Groups Name      Variance Std.Dev. Corr
##   male   x1         0.0472762 0.21743
##           I(x1^2) 0.0001374 0.01172  0.96
## Fit improves, but I get convergence complaints.
## This is likely due to the high correlation between the two RE.

## WOuld like to post a question about this to SE
## data_example <- data %>% mutate(f = as.factor(as.integer(male))) %>% select(f, x = temp_mean, y = song)

#tmp <- update(glmer_poisson, formula = formula_quad_RE <- song)

glmer_nb <-
  glmer.nb(formula_quad_RE,
    data = data,
    ## control values are used by the initial optimization
    ## using a poisson glmer model, which doesn't converge
    control = glmerControl(
      boundary.tol = 0,
      tolPwrss=1e-1,
      optCtrl = list(maxiter = 1E5,
                     maxfun = 2E6,
                     trace = trace),
      optimizer="nloptwrap"),
    ## nb.control values are used by the second optimizer
    ## Note need to set own optCtrl values
```

```

      nb.control = list(
        optCtrl = list(maxit = 1000,
                      maxfun = 2E5)),
      verbose = verbose
    )

```

```
## Error in eval(predvars, data, env): object 'x1' not found
```

```

## Try dropping the quadratic RE
formula_linear_RE <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1| male)
rm(glmer_poisson_linear)

```

```
## Warning in rm(glmer_poisson_linear): object 'glmer_poisson_linear' not found
```

```

try(glmer_poisson_linear <- update(glmer_poisson, formula = formula_linear_RE,
                                verbose = verbose)
);

```

```
## Error in h(simpleError(msg, call)) :
```

```
## error in evaluating the argument 'object' in selecting a method for function 'update': object 'glm
```

```

if(exists("glmer_poisson_linear")){
  summary(glmer_poisson_linear);
  if(make_plot) plot(glmer_poisson_linear, ask = FALSE)
}

```

```
## This model fails to converge, strangely.
```

```

## NB model fails to converge
## This appears to be because the variances of the RE become very small (on the order of E-11!)
## See B. Bolker's notes on GLMM fitting and comparisons
## https://bbolker.github.io/mixedmodels-misc/ecostats_chap.html#diagnostics-and-model-summaries
glmer_nb_linear <-
  glmer.nb(formula_linear_RE,
    data = data,
    ## control values are used by the initial optimization
    ## using a poisson glmer model, which doesn't converge
    control = glmerControl(
      boundary.tol = 0,
      tolPwrss=1e-1,
      optCtrl = list(maxiter = 1E5,
                    maxfun = 2E6,
                    trace = trace),
      optimizer="nloptwrap"),
    ## nb.control values are used by the second optimizer
    ## Note need to set own optCtrl values
    nb.control = list(
      optCtrl = list(maxit = 1000,
                    maxfun = 2E5)),
    verbose = verbose
  )

```



```
## Error in eval(predvars, data, env): object 'x1' not found
```

```
rm(make_plot, formula_quad_RE, formula_linear_RE)
```

Result

- Model doesn't converge.
- Output includes > Model is nearly unidentifiable: very large eigenvalue > - Rescale variables?; Model is nearly unidentifiable: large eigenvalue ratio > - Rescale variables? To me this initially suggested we should use `song_prop` and the `glmer` weights function. However, I've made some progress in my understanding of ME models.

Use All Round 2 Data

Create `fit_db`

```
vec_temp_ref <- c(27, 35, 44, 45)

## Create a tibble for holding model result
## filter out rows that won't get defined.

fit_db <- expand_grid(temp_ref = vec_temp_ref, model = c("glm", "glmer"), re = c("none", "linear", "quad"),
  filter( !(model == "glm" & est_cov == TRUE)) %>%
  filter( !(model == "glmer" & dist == "qpoisson")) %>%
  add_column(fit = list(list(NA)))

## Try to understand how to work with fit_db

## Filter fit's that are NA, but doesn't work if I assign glm objects to fit[[i]]
subset(fit_db, is.na(fit[[1]]))
```

```
## # A tibble: 112 x 6
##   temp_ref model re     est_cov dist      fit
##   <dbl> <chr> <chr> <lgl> <chr> <list>
## 1      27 glm   none  FALSE poisson <list [1]>
## 2      27 glm   none  FALSE qpoisson <list [1]>
## 3      27 glm   none  FALSE nb      <list [1]>
## 4      27 glm   linear FALSE poisson <list [1]>
## 5      27 glm   linear FALSE qpoisson <list [1]>
## 6      27 glm   linear FALSE nb      <list [1]>
## 7      27 glm   quad  FALSE poisson <list [1]>
## 8      27 glm   quad  FALSE qpoisson <list [1]>
## 9      27 glm   quad  FALSE nb      <list [1]>
## 10     27 glm   both  FALSE poisson <list [1]>
## # ... with 102 more rows
```

```
## Doesn't know what fit is
select(fit_db, is.na(fit[[1]]))
```

```
## Error in 'select()':
## ! Problem while evaluating 'is.na(fit[[1]])'.
## Caused by error:
## ! object 'fit' not found
```

GLM Fits

- Just FE

Fit Fixed Effects models

Fit Fixed Effects models

```
est_cov <- FALSE
verbose <- 0
trace <- FALSE

for(temp_ref in vec_temp_ref ){

  data <- data_ind %>%
    mutate(x1 = (temp_mean - temp_ref))
  print(paste("temp_ref:", temp_ref))

  re <- "none"

  glm_poisson <-
    glm(song_count ~ 1 + male + (x1) + I(x1^2),
        data = data,
        family = poisson(link = "log")
    )

  summary(glm_poisson)
  plot(glm_poisson, ask = FALSE)

  fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glm" & fit_db$dist == "poisson" & fit_db$re == re] = glm_poisson

  glm_qpoisson <- update(glm_poisson,
                        family = quasipoisson(link = "log")
    )
  summary(glm_qpoisson)

  plot(glm_qpoisson, ask = FALSE)

  fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glm" & fit_db$dist == "qpoisson" & fit_db$re == re] = glm_qpoisson

  glm_qpoisson <- update(glm_poisson,
                        family = quasipoisson(link = "log")
    )
  summary(glm_qpoisson)
  ## Note the dispersion parameter is 29.7 rather than 1!!
  ## Clearly the data is over dispersed
```

```

## quasipoisson() doesn't seem to exist for glmer (but likely exists in nlme)
plot(glm_qpoisson, ask = FALSE)

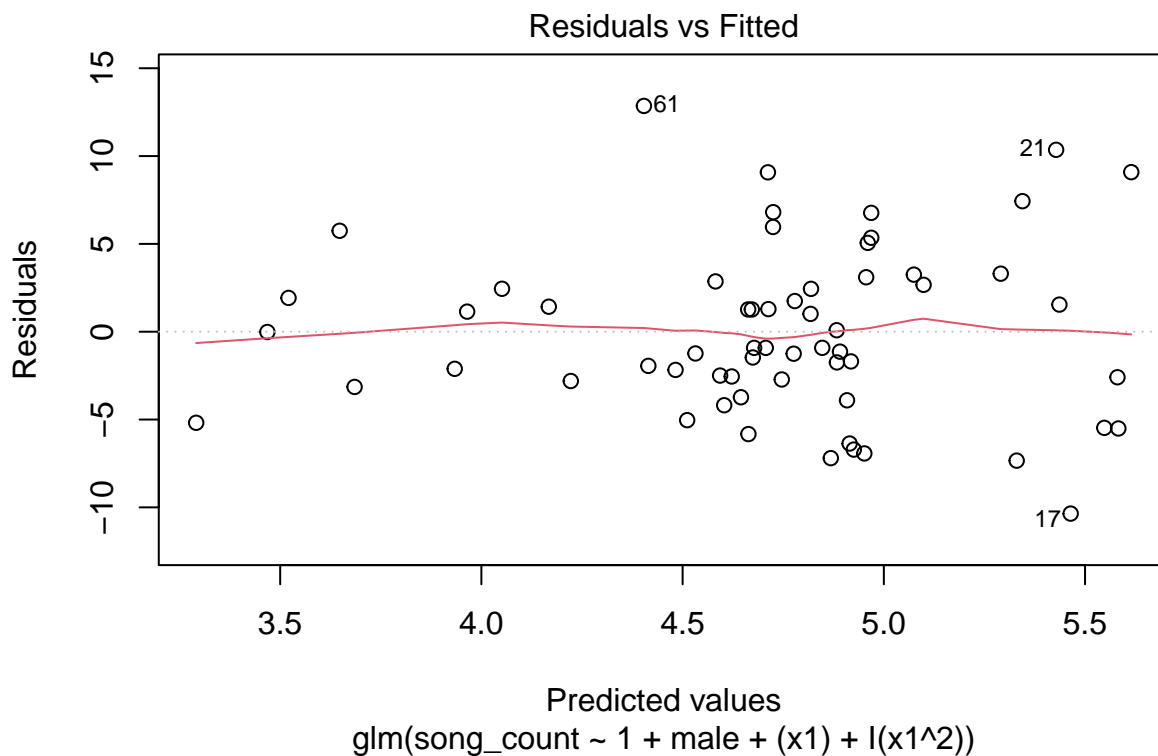
fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glm" & fit_db$dist == "qpoisson" & fit_db$re == re]

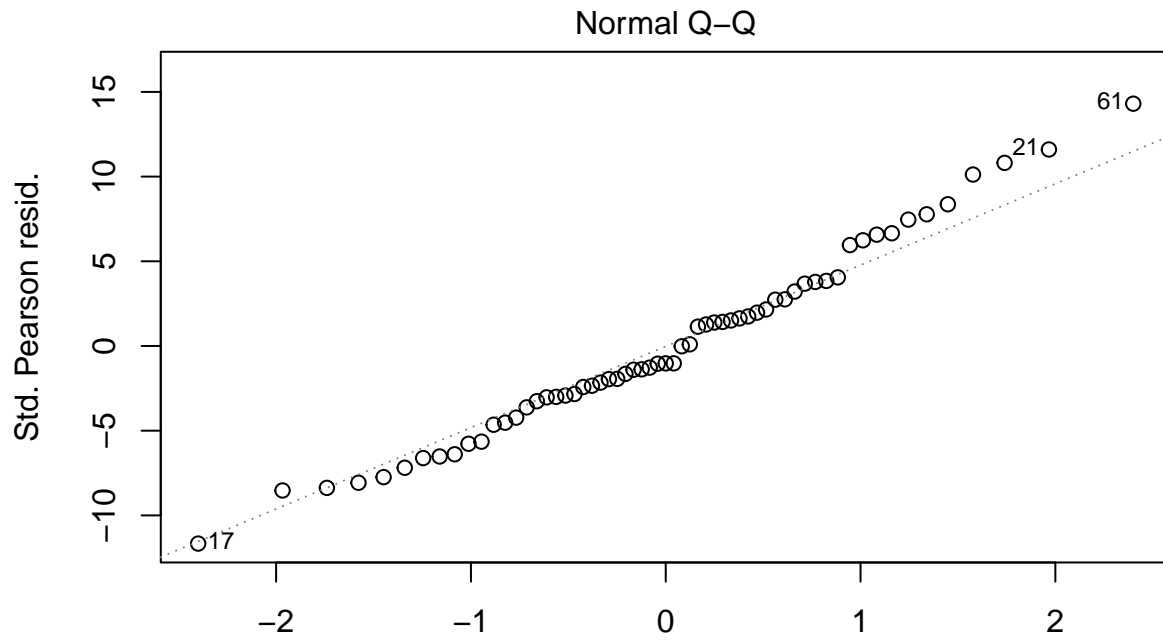
glm_nb <- glm.nb <-
  glm(song_count ~ 1 + male + (x1) + I(x1^2),
      data = data
    )
summary(glm_nb)
plot(glm_nb, ask = FALSE)

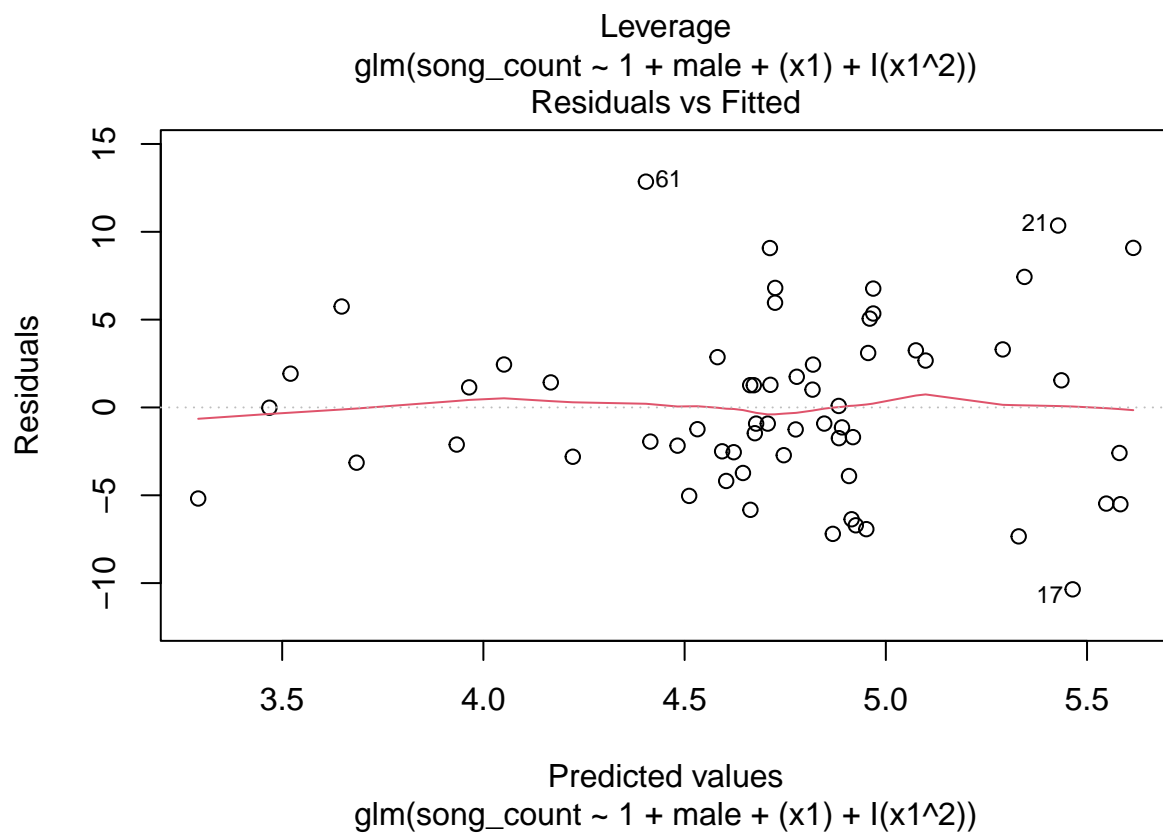
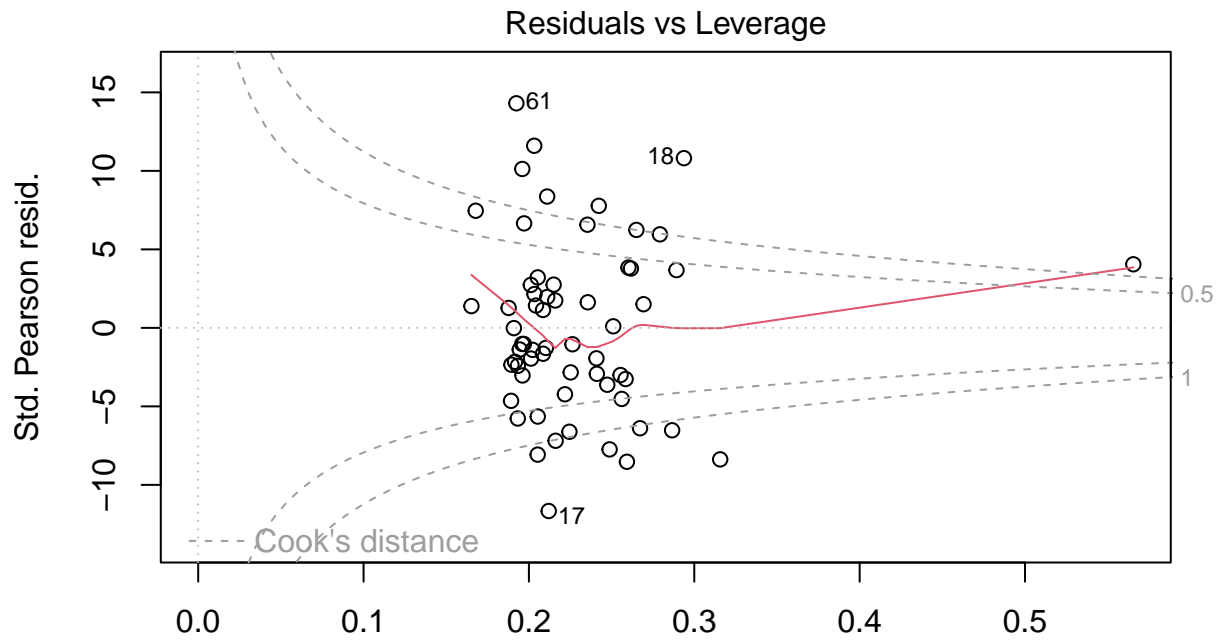
fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glm" & fit_db$dist == "nb" & fit_db$re == re]
## Dispersion parameter is 4547!
}

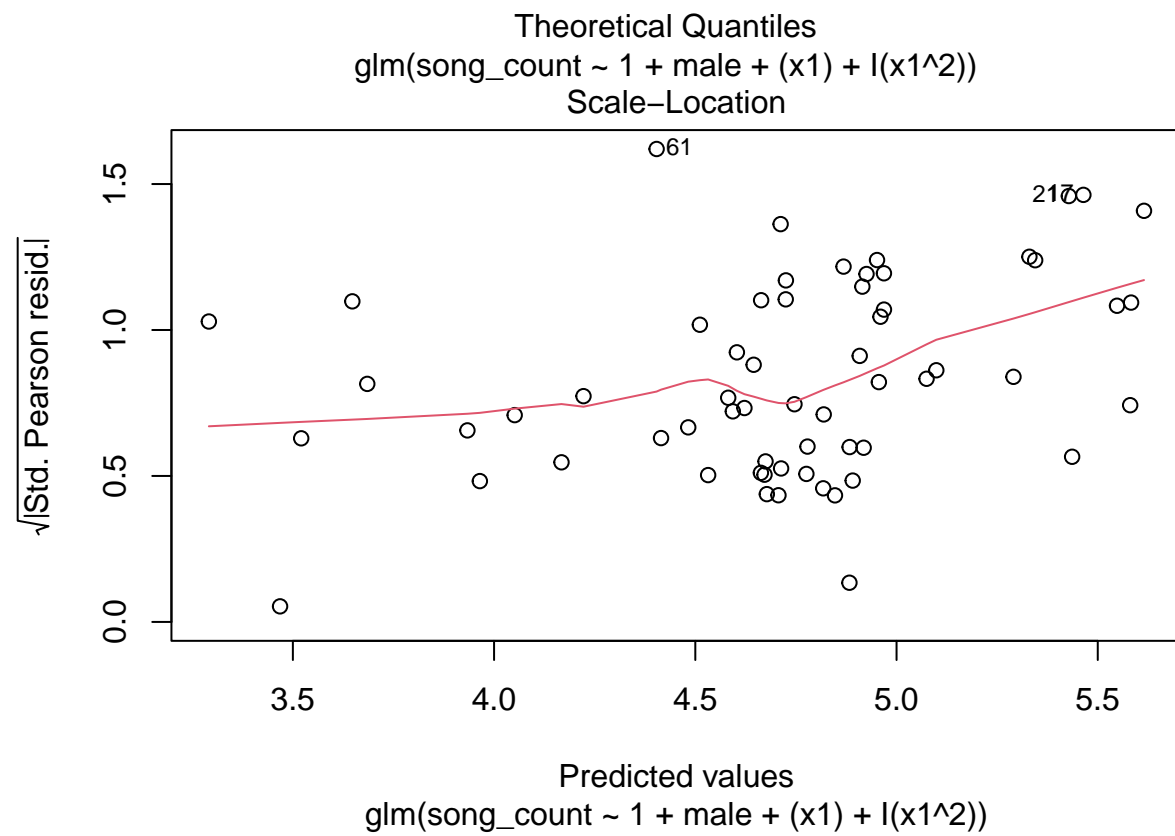
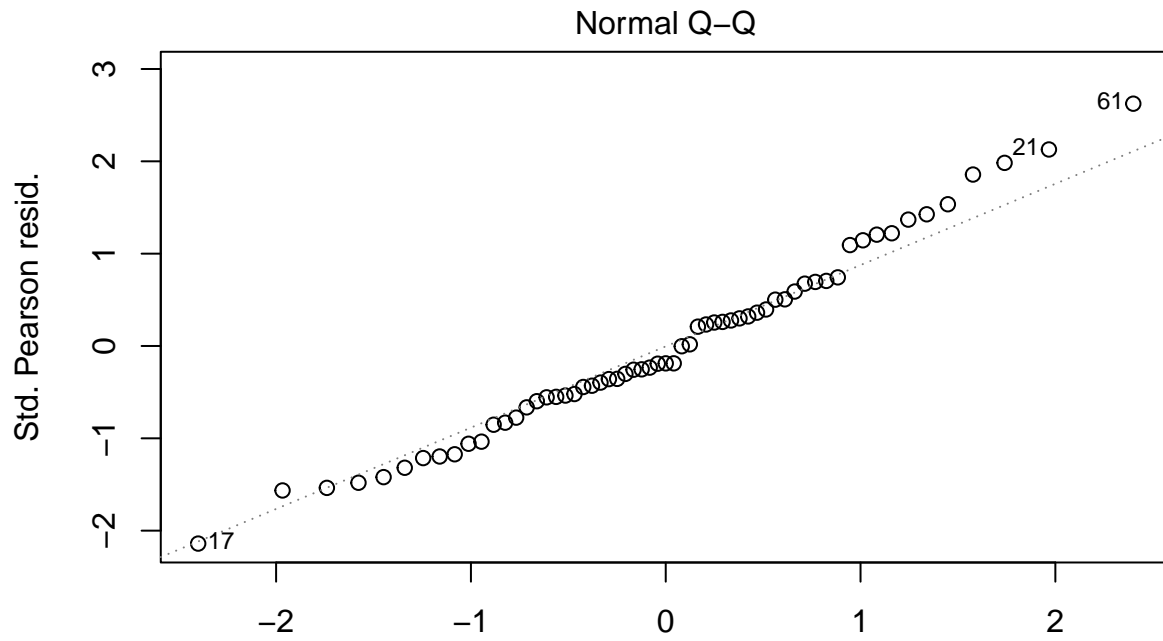
```

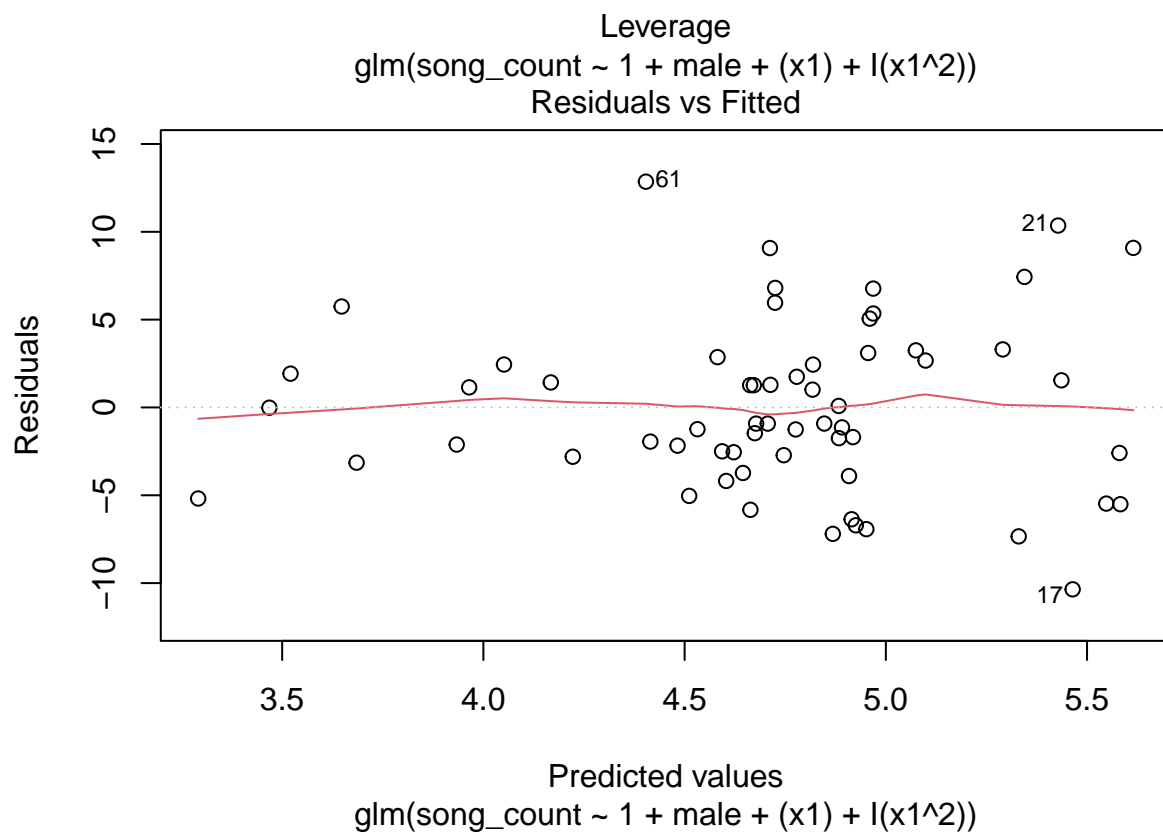
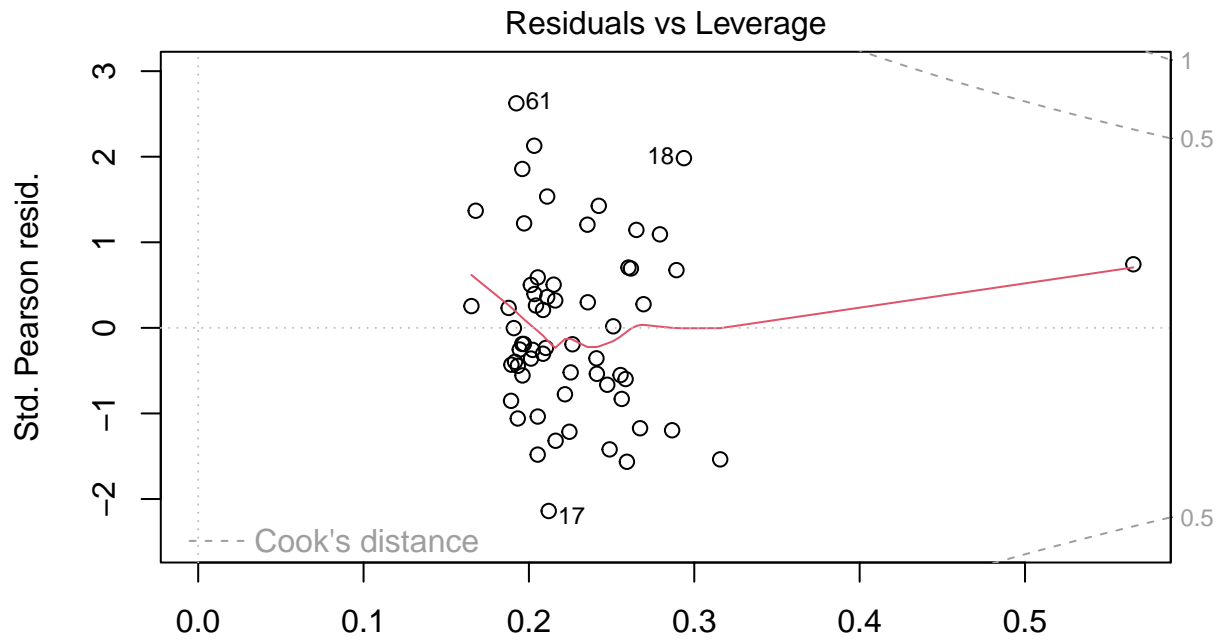
```
## [1] "temp_ref: 27"
```

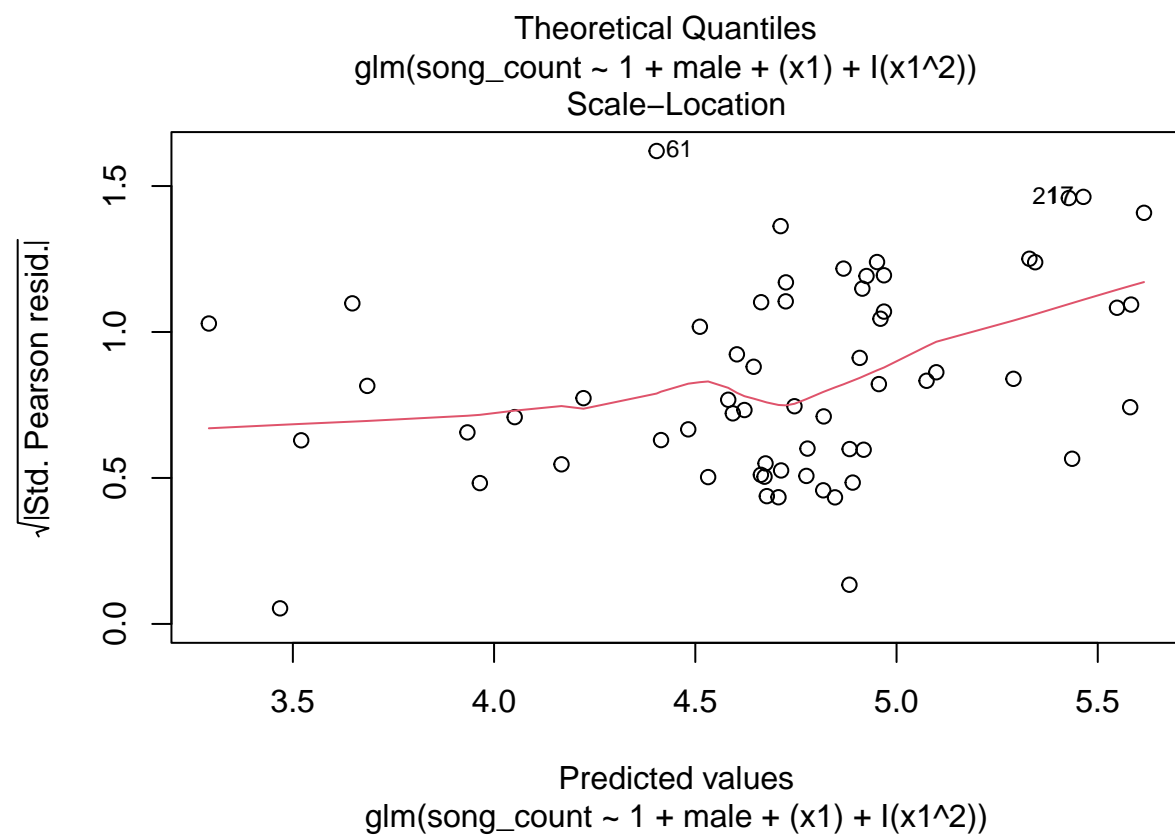
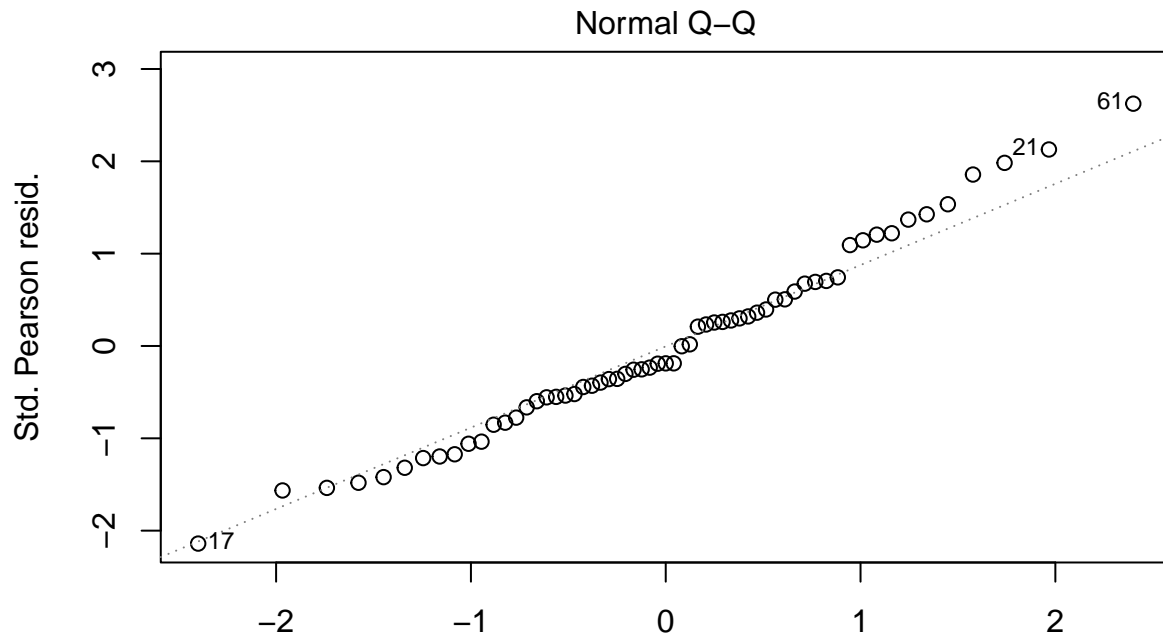


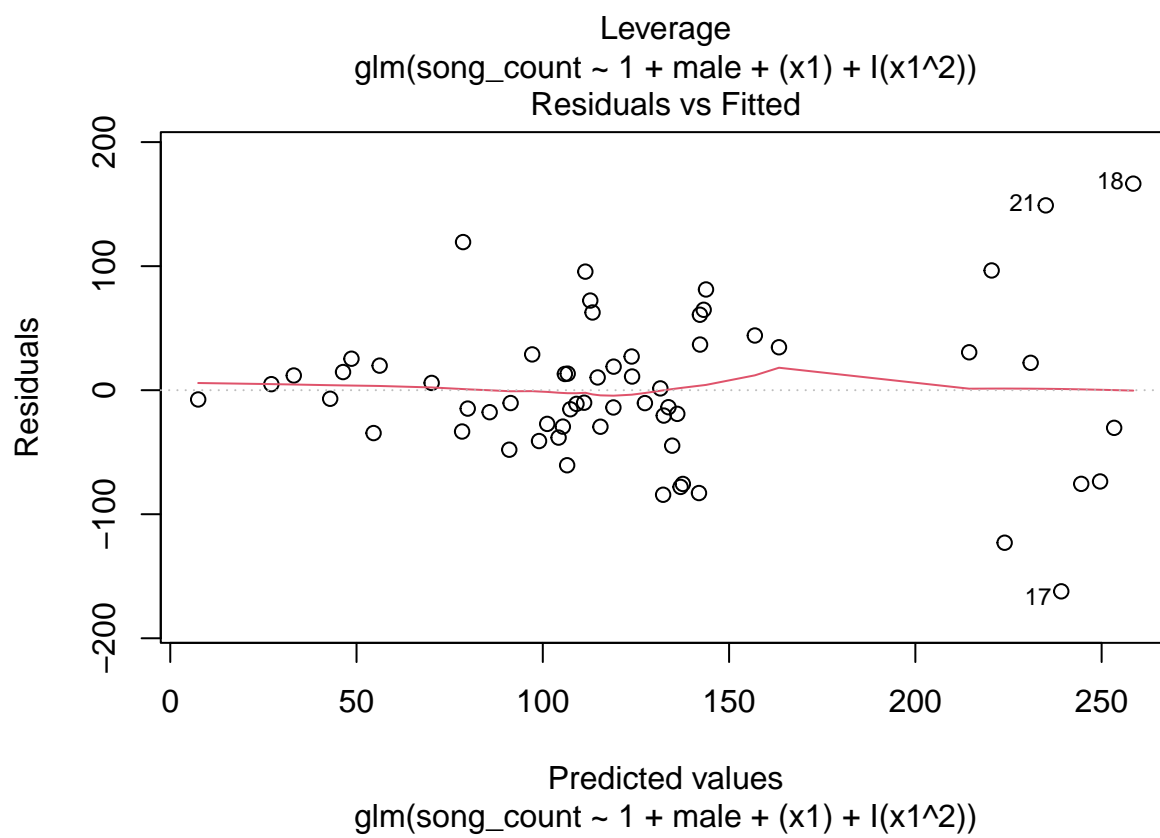
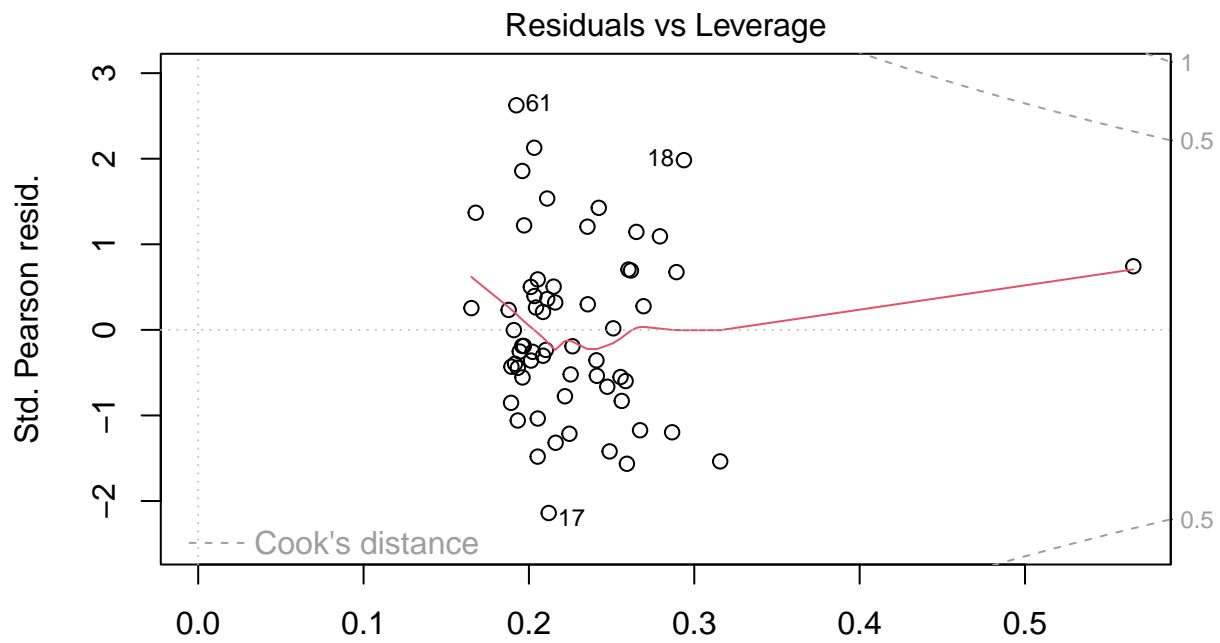


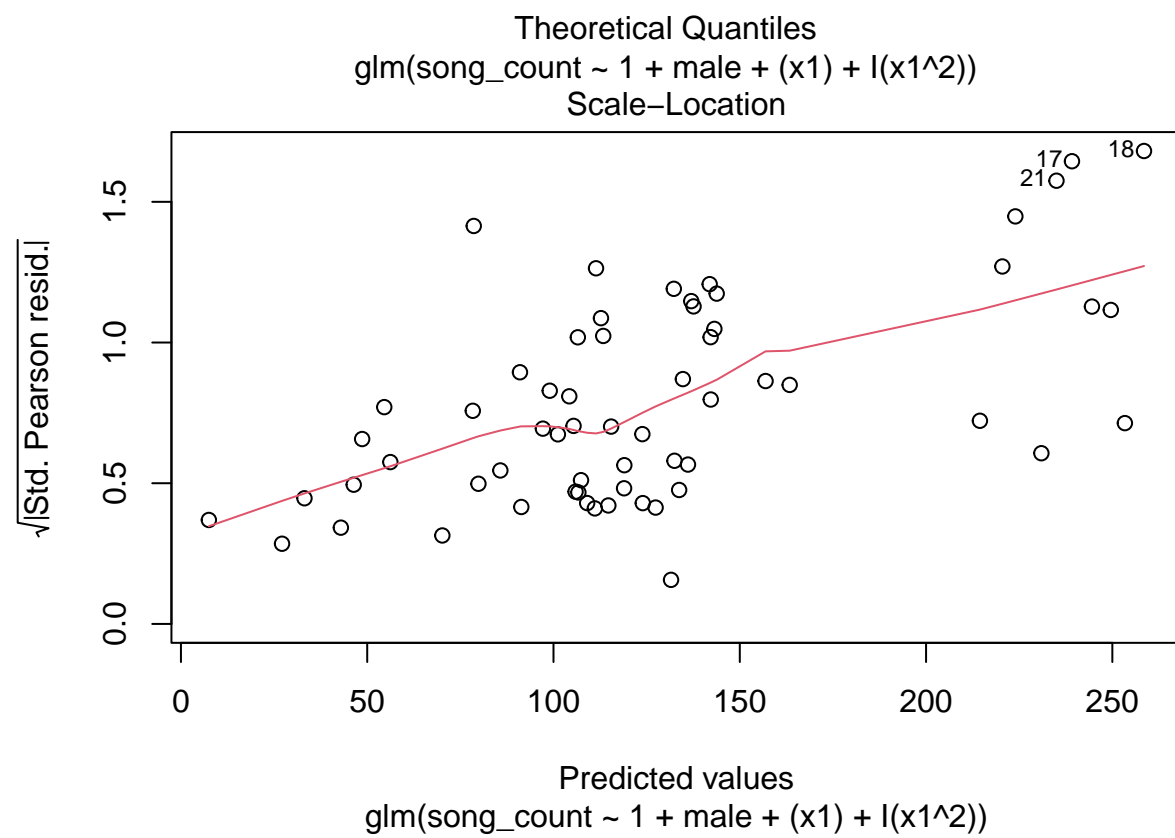
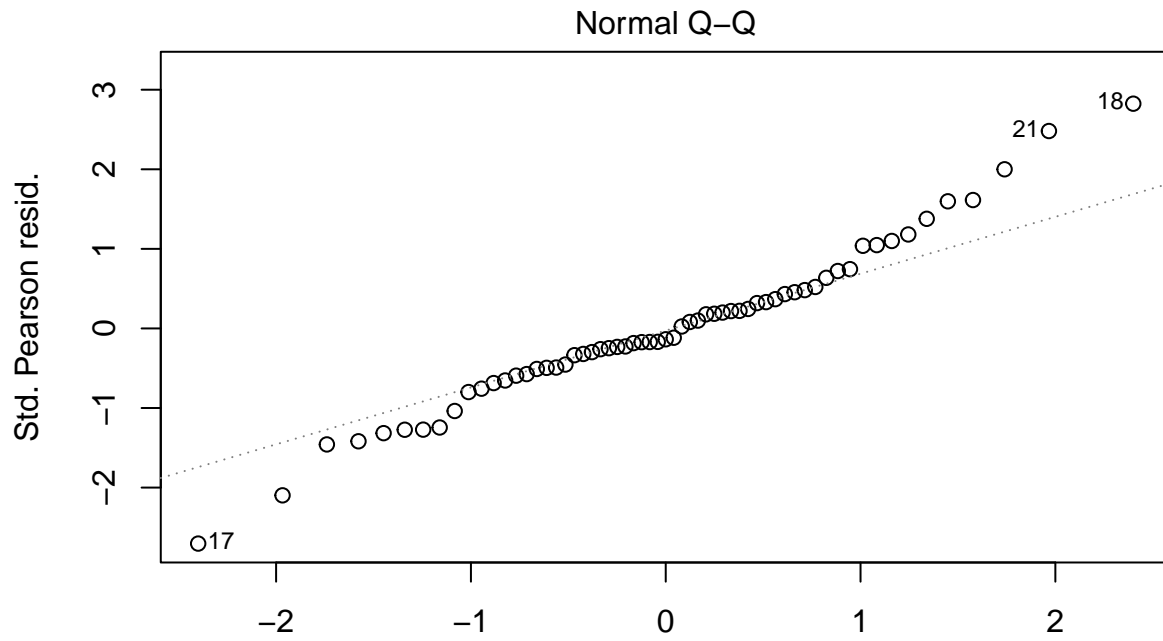


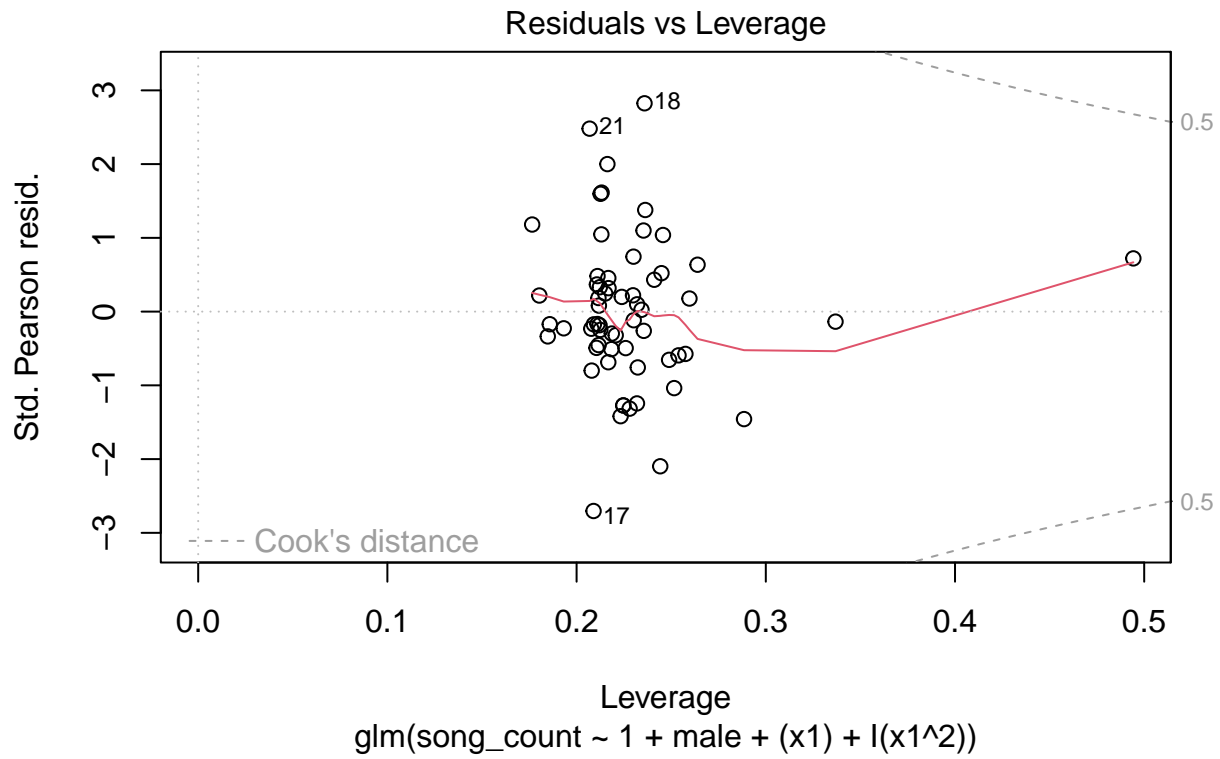




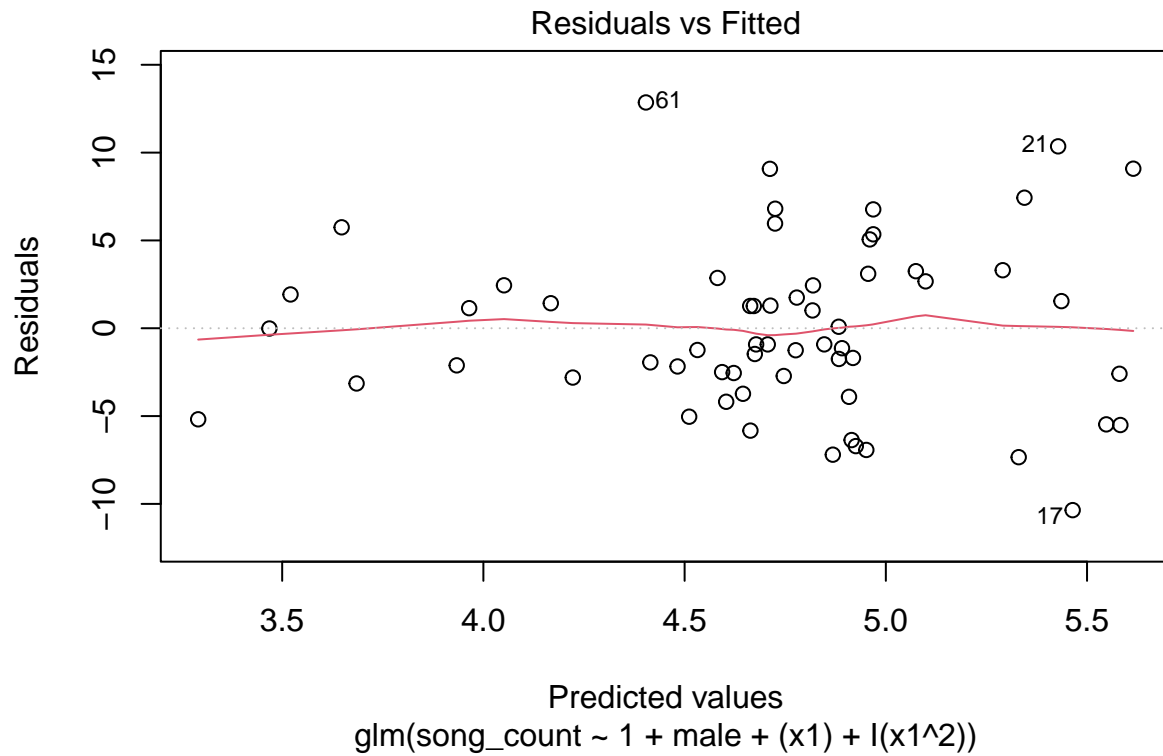


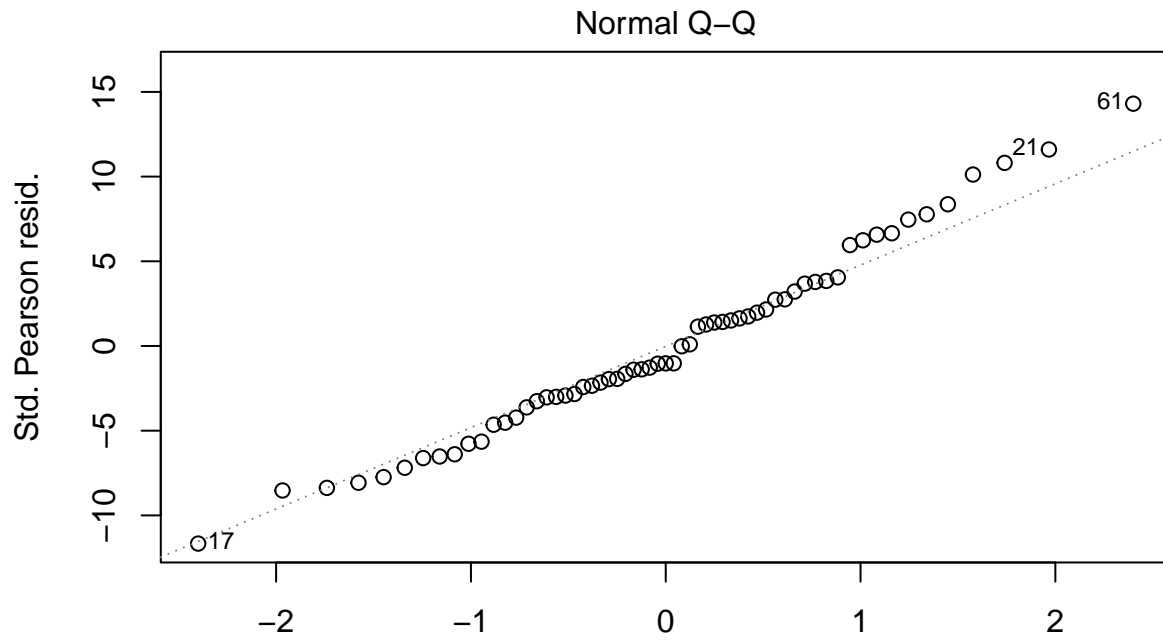


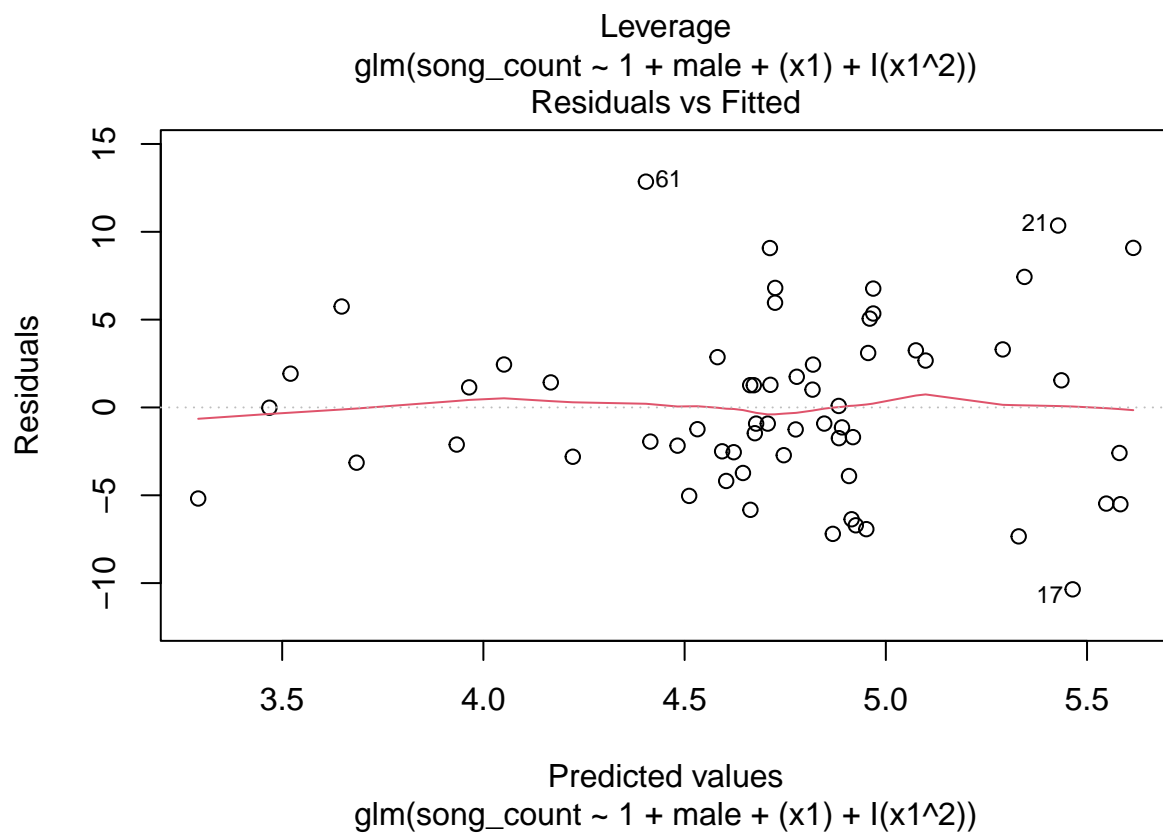
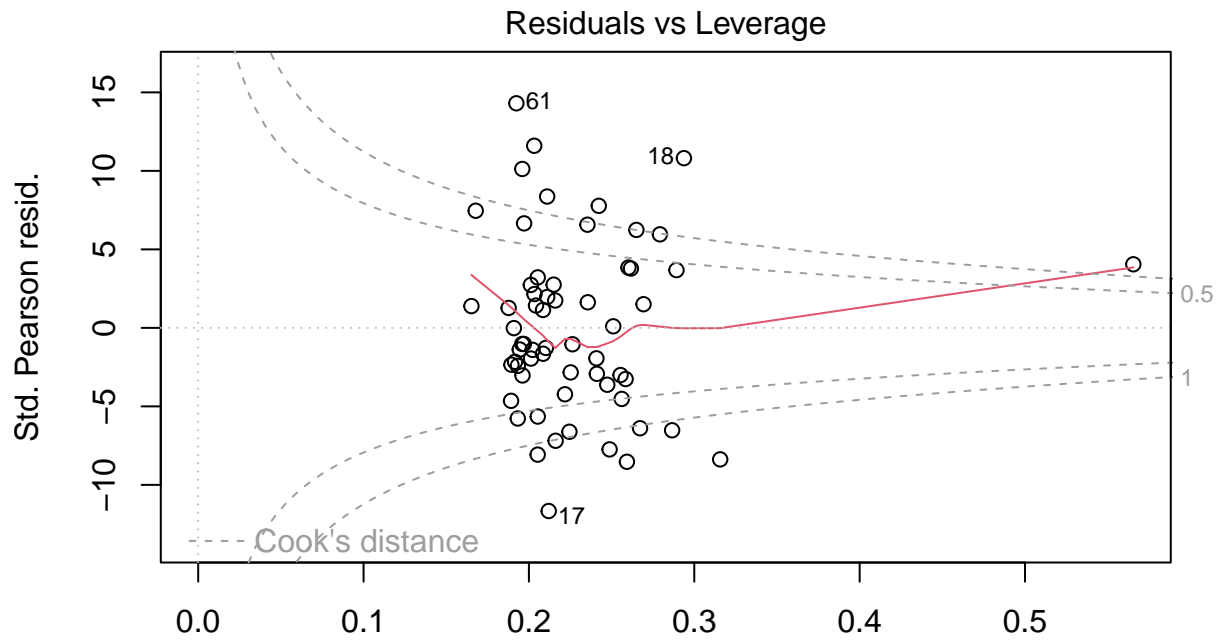


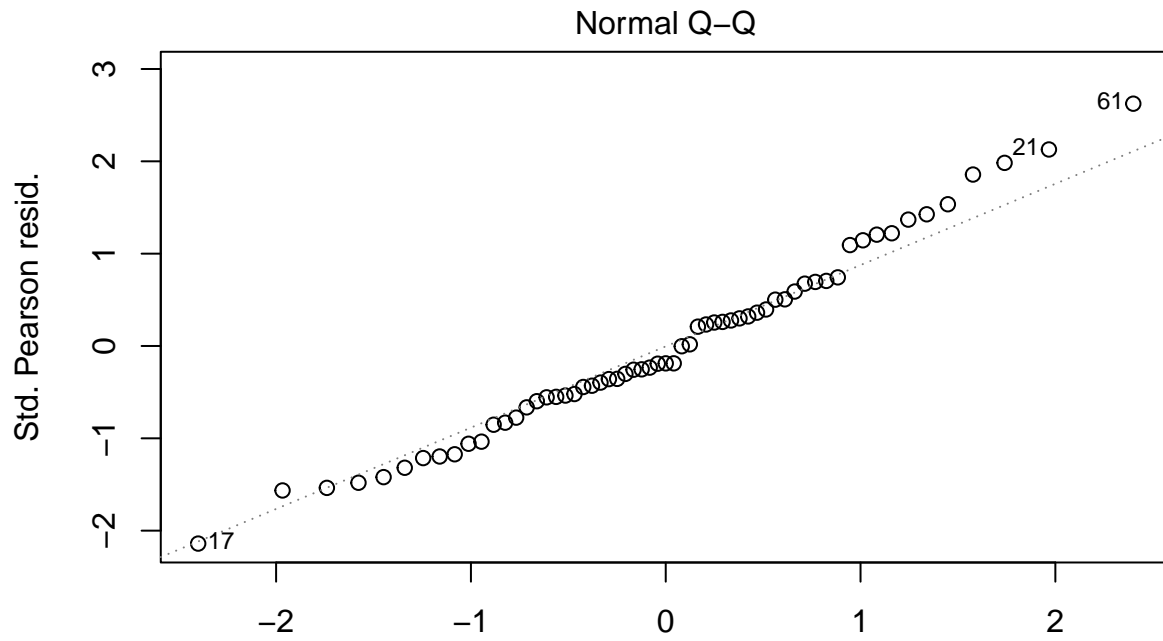


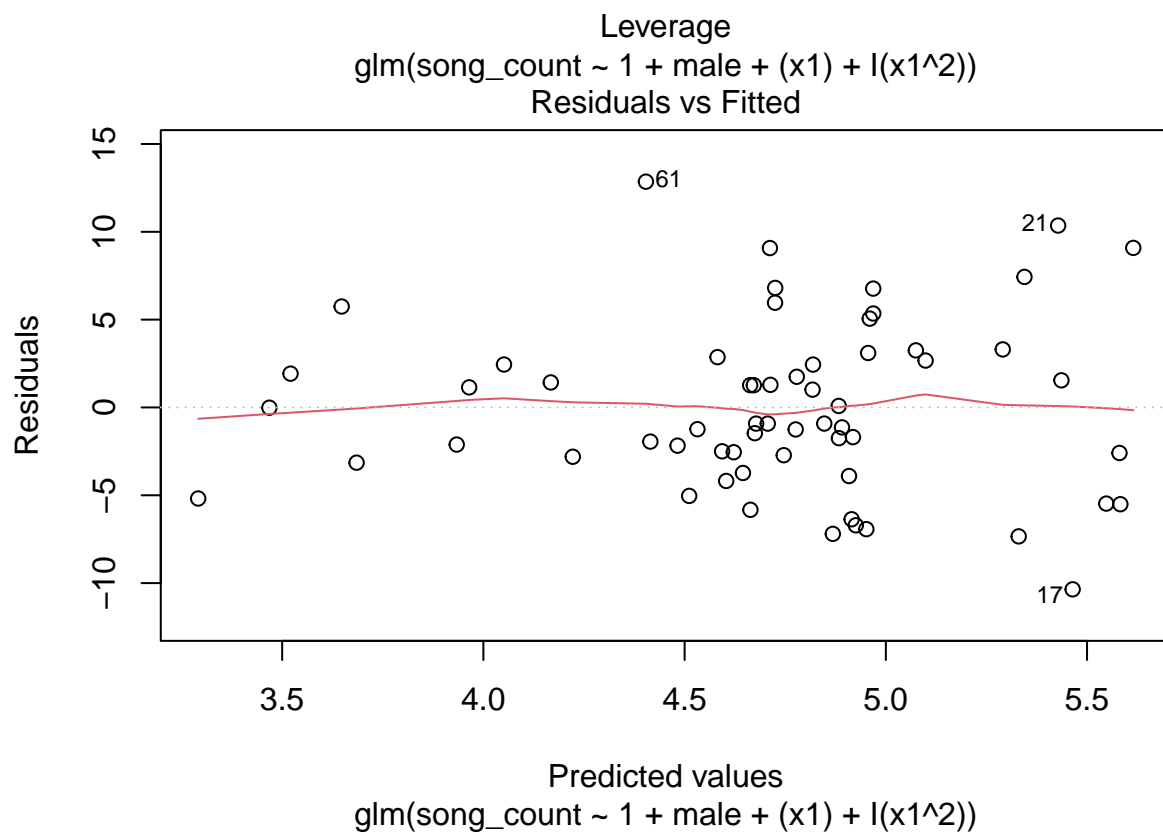
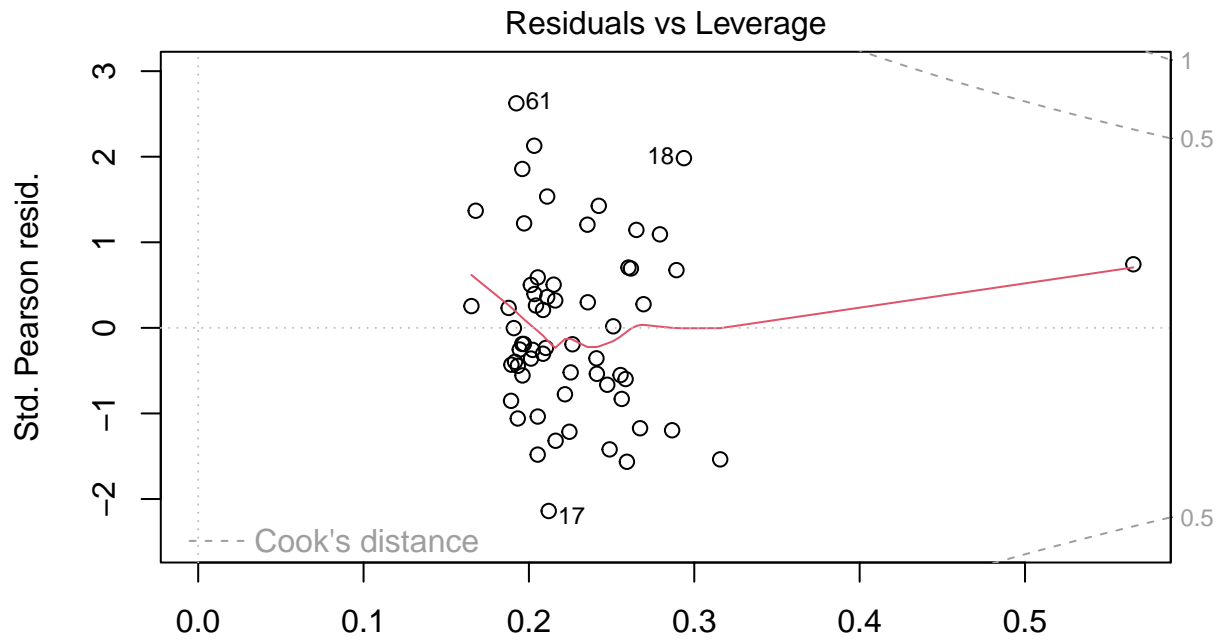
```
## [1] "temp_ref: 35"
```

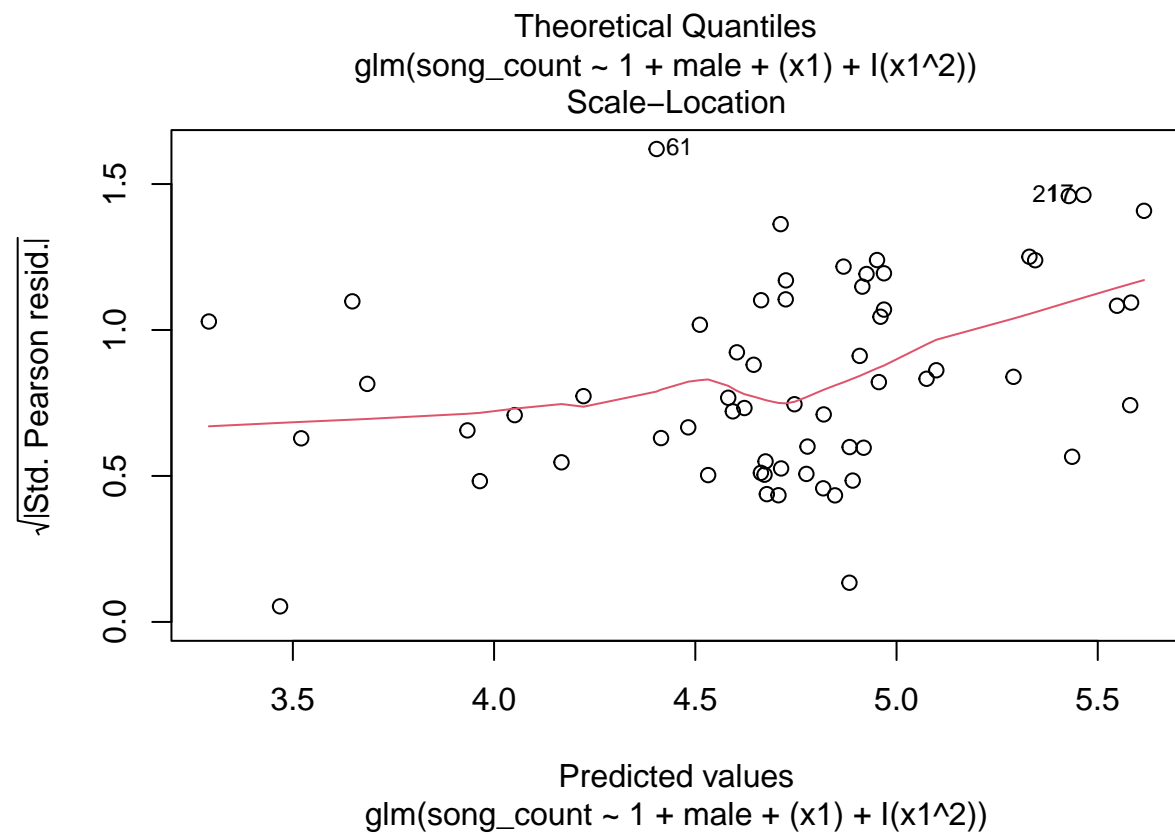
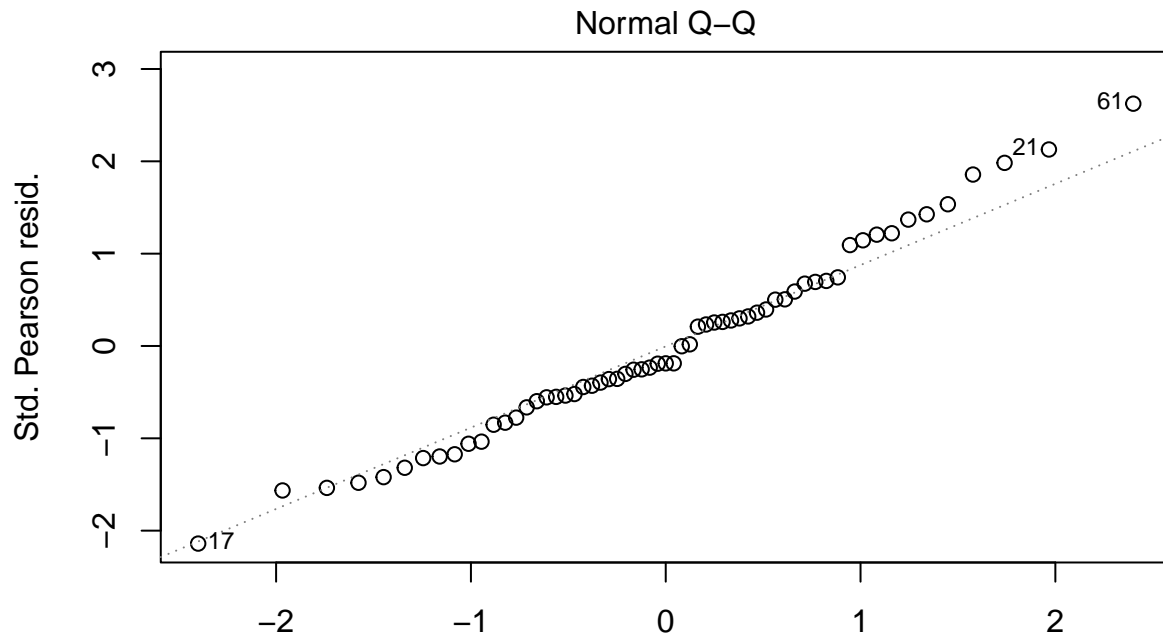


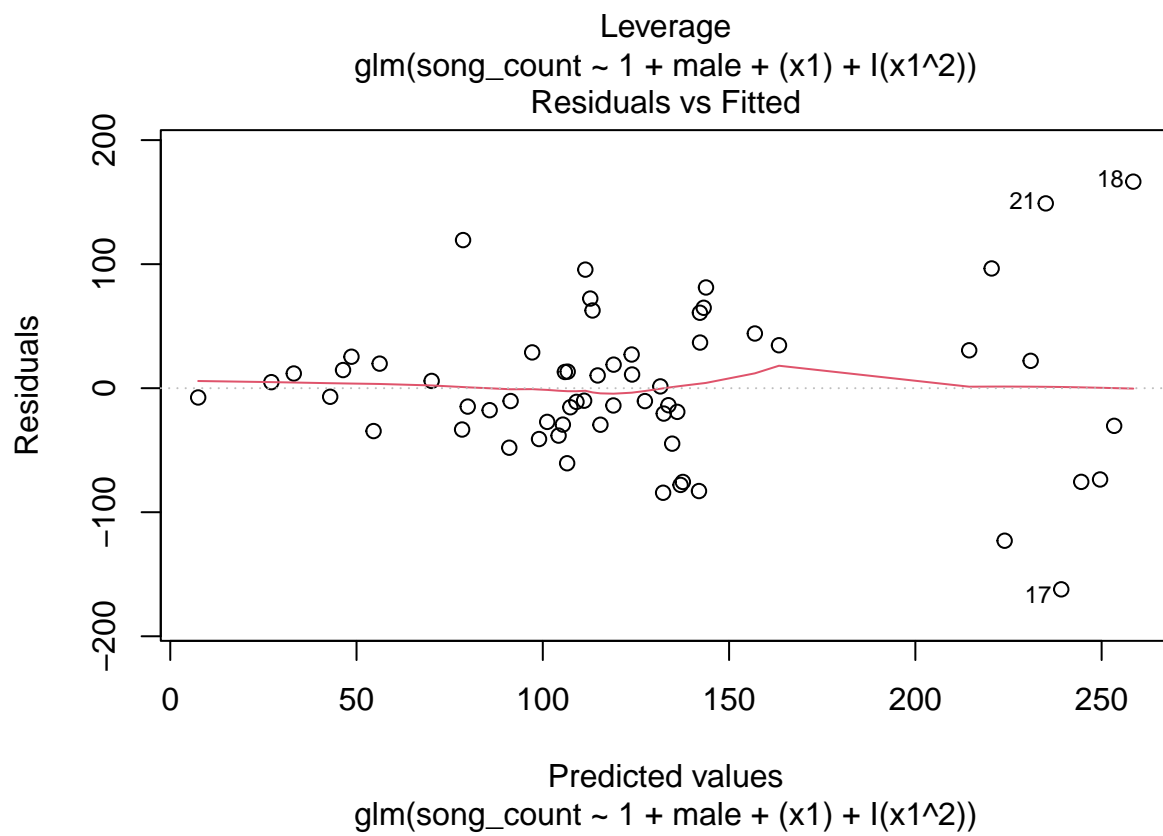
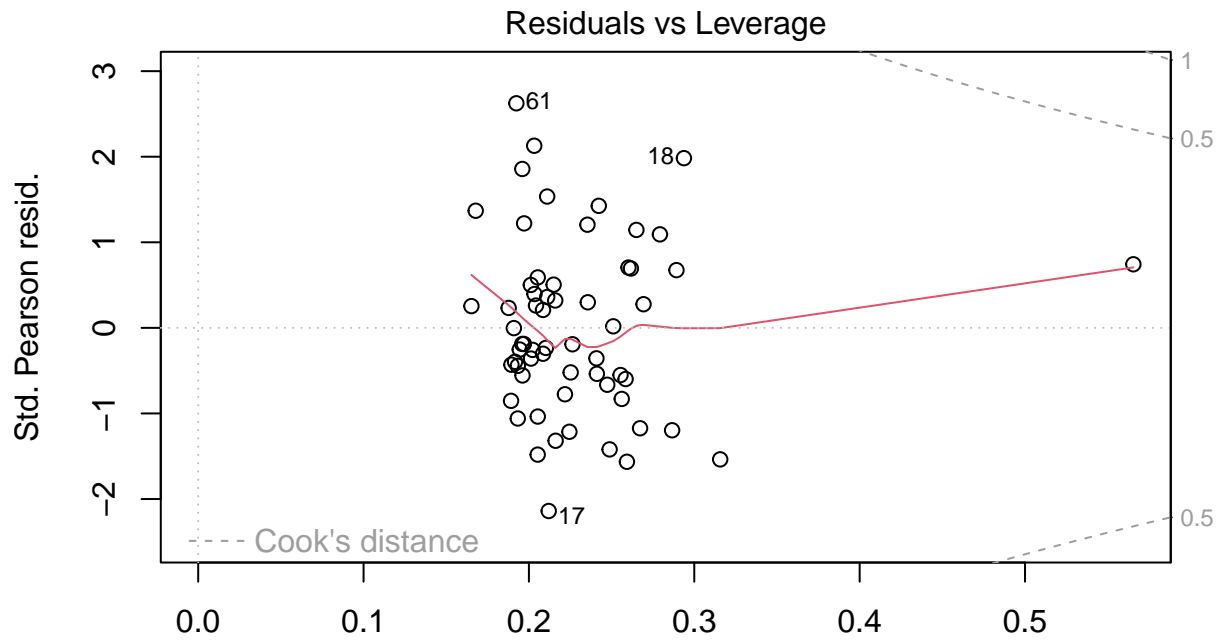


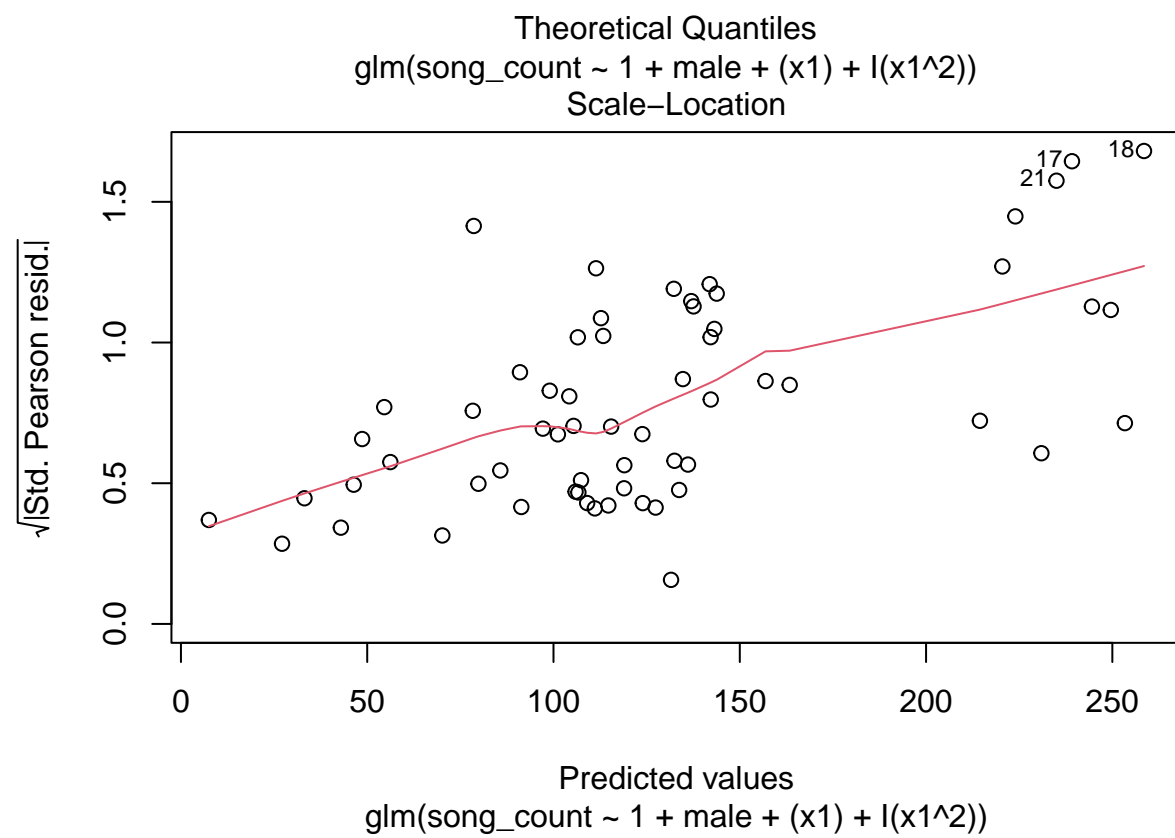
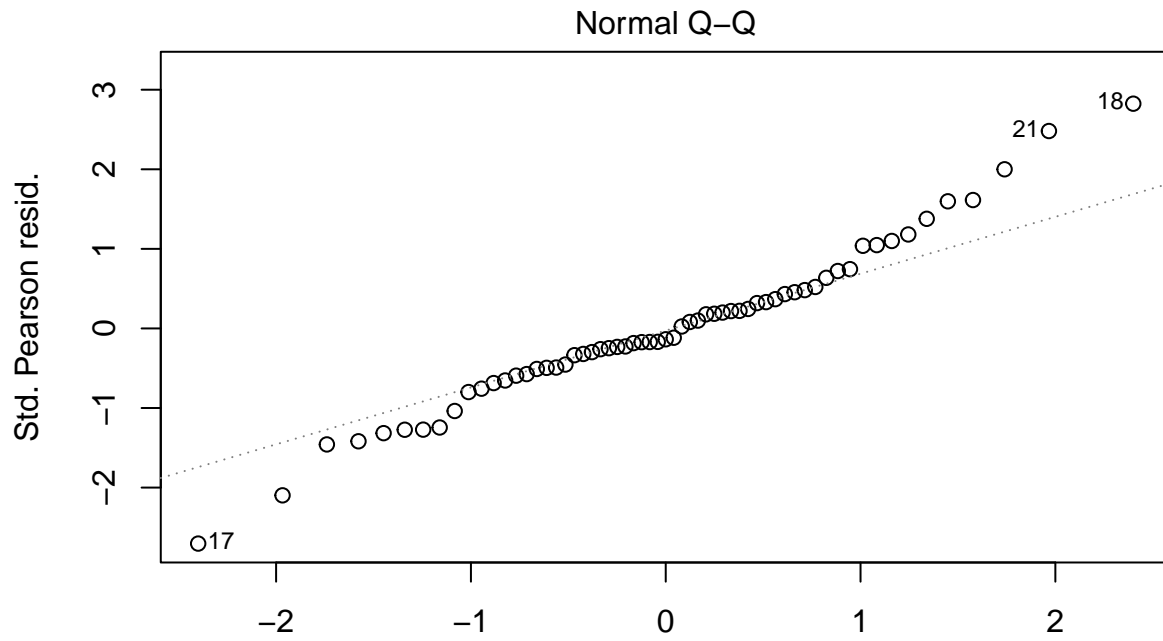


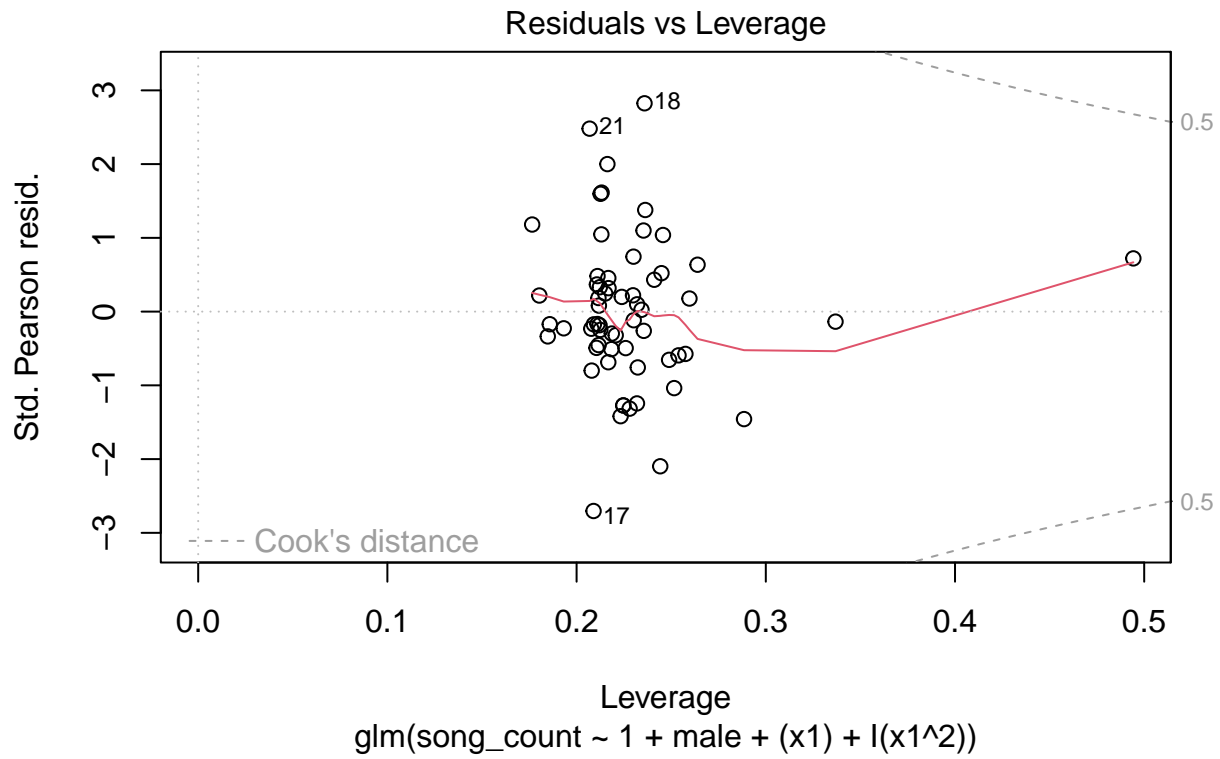




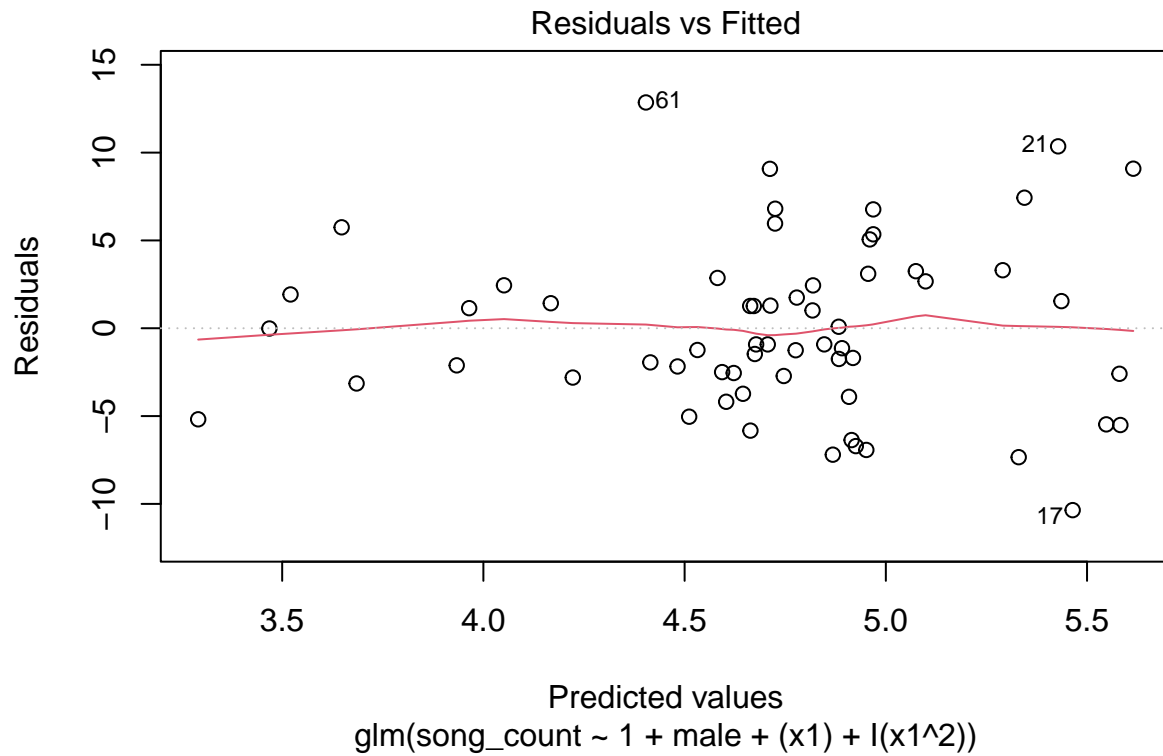


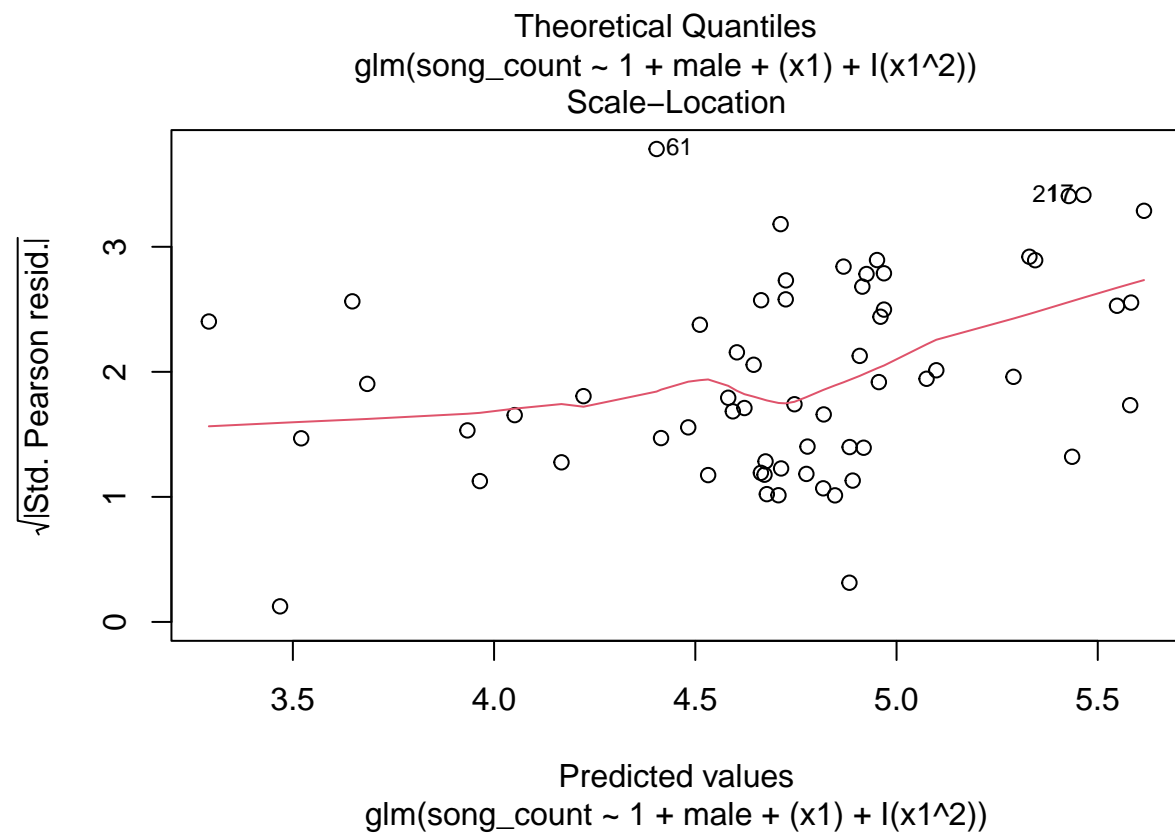
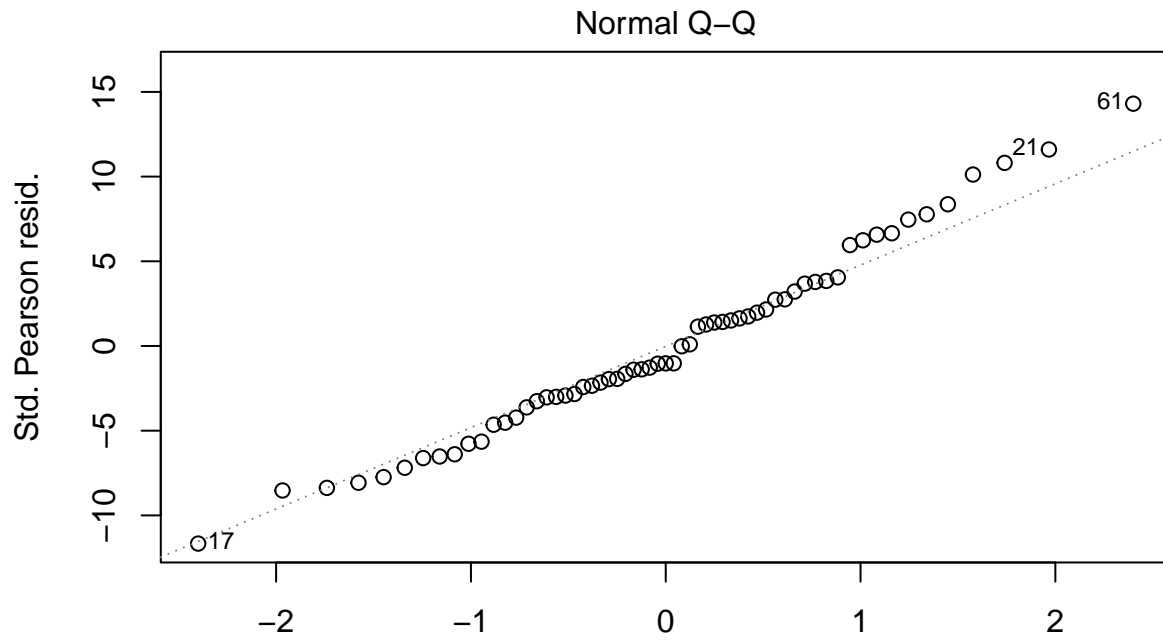


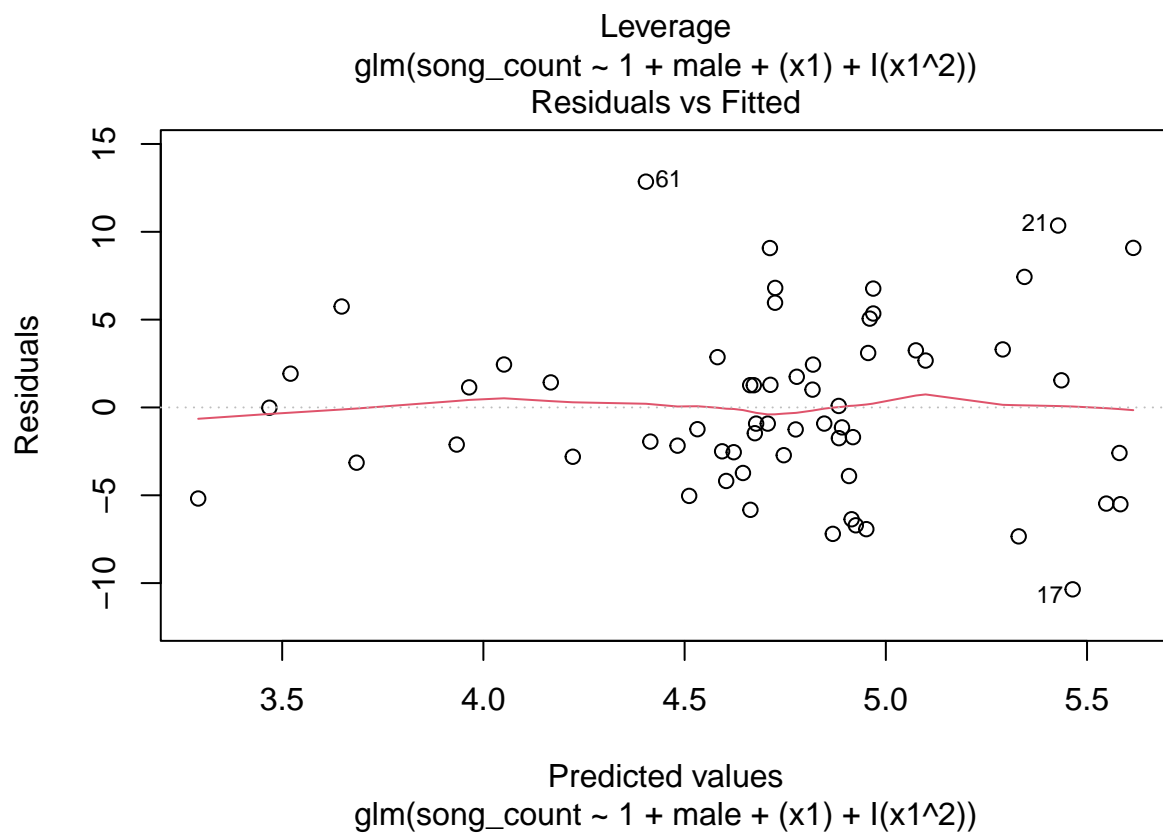
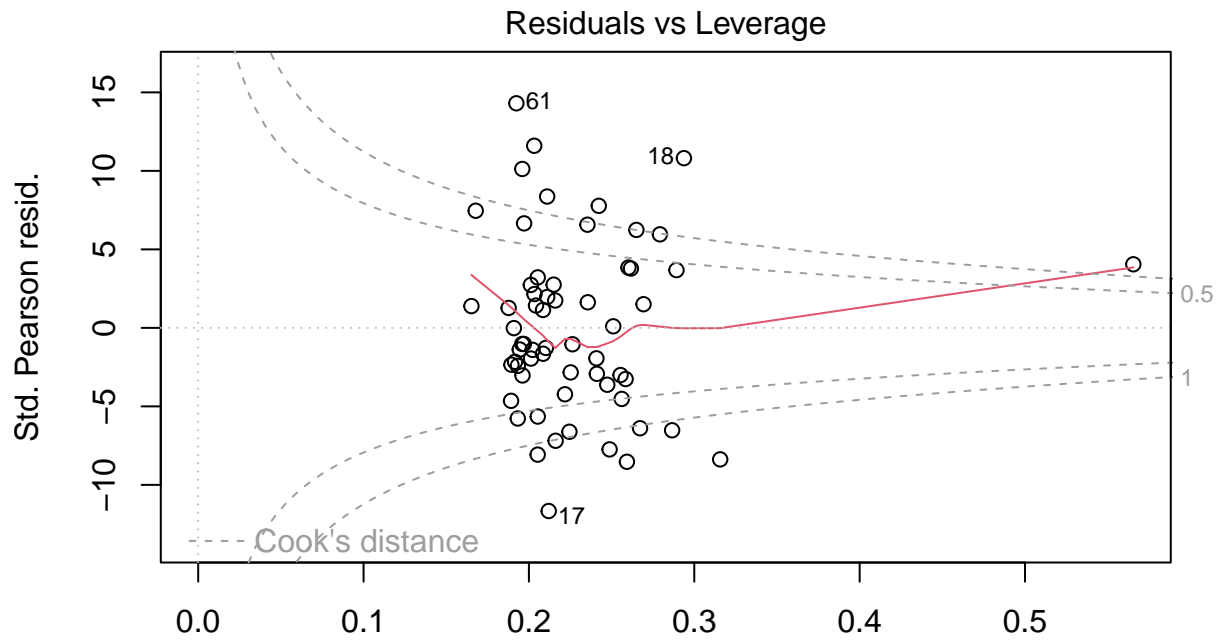


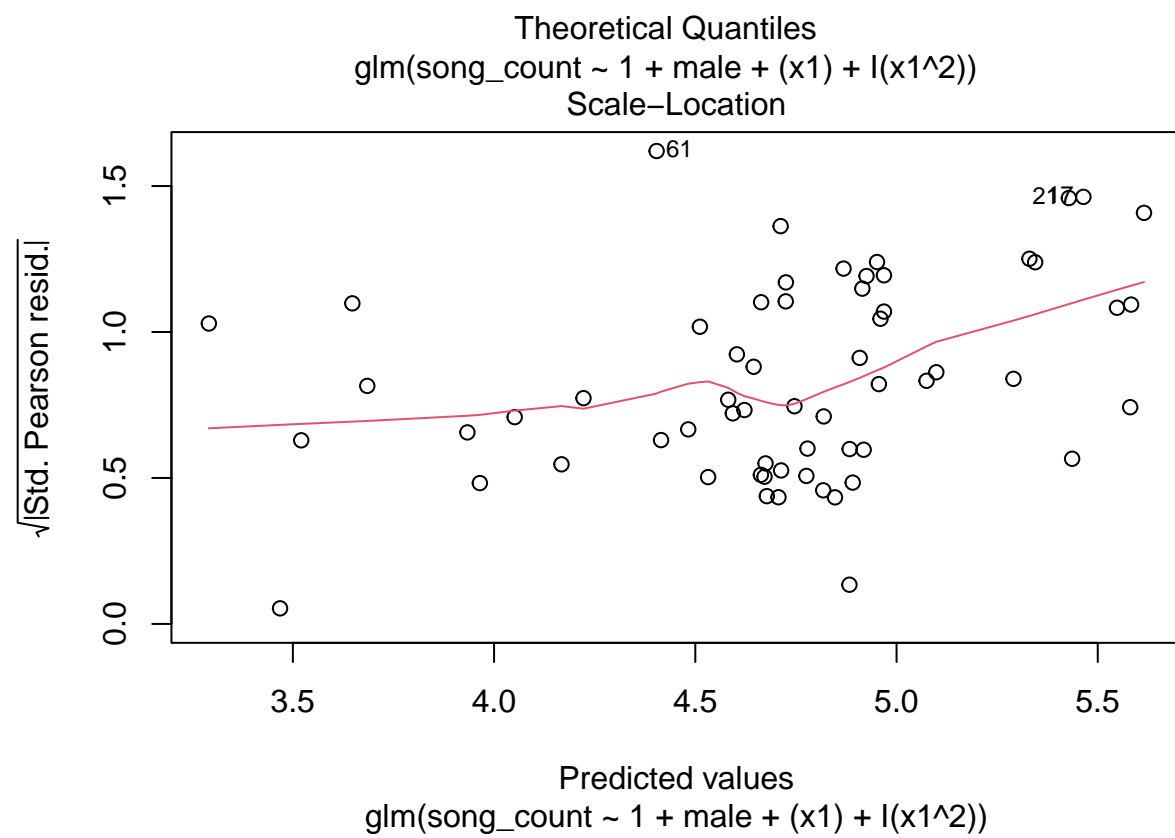
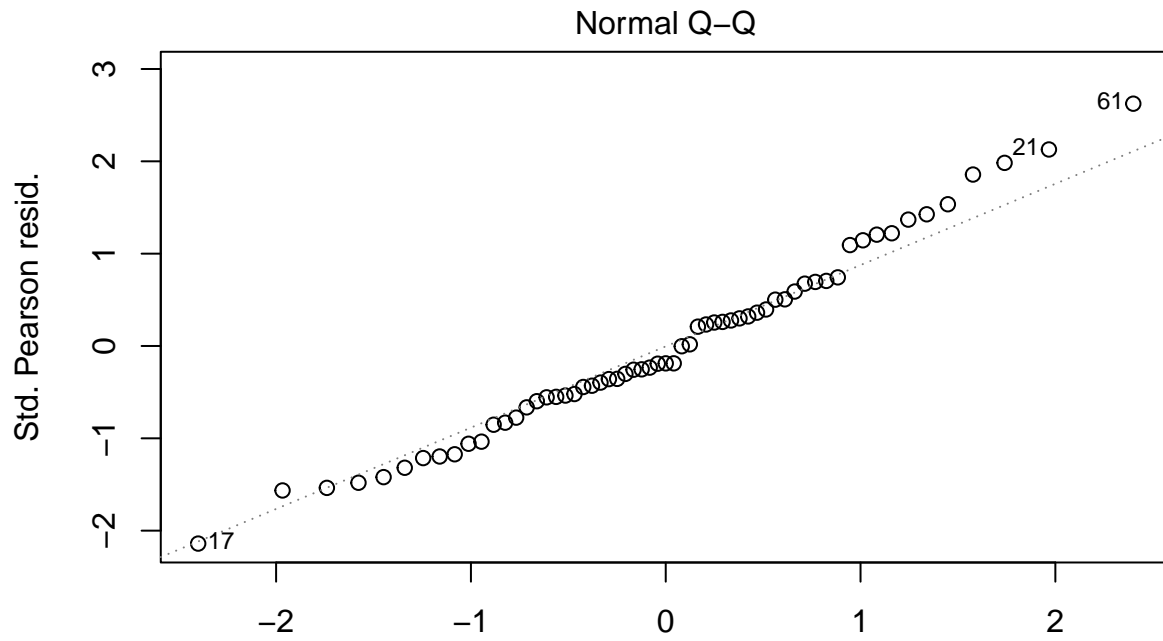


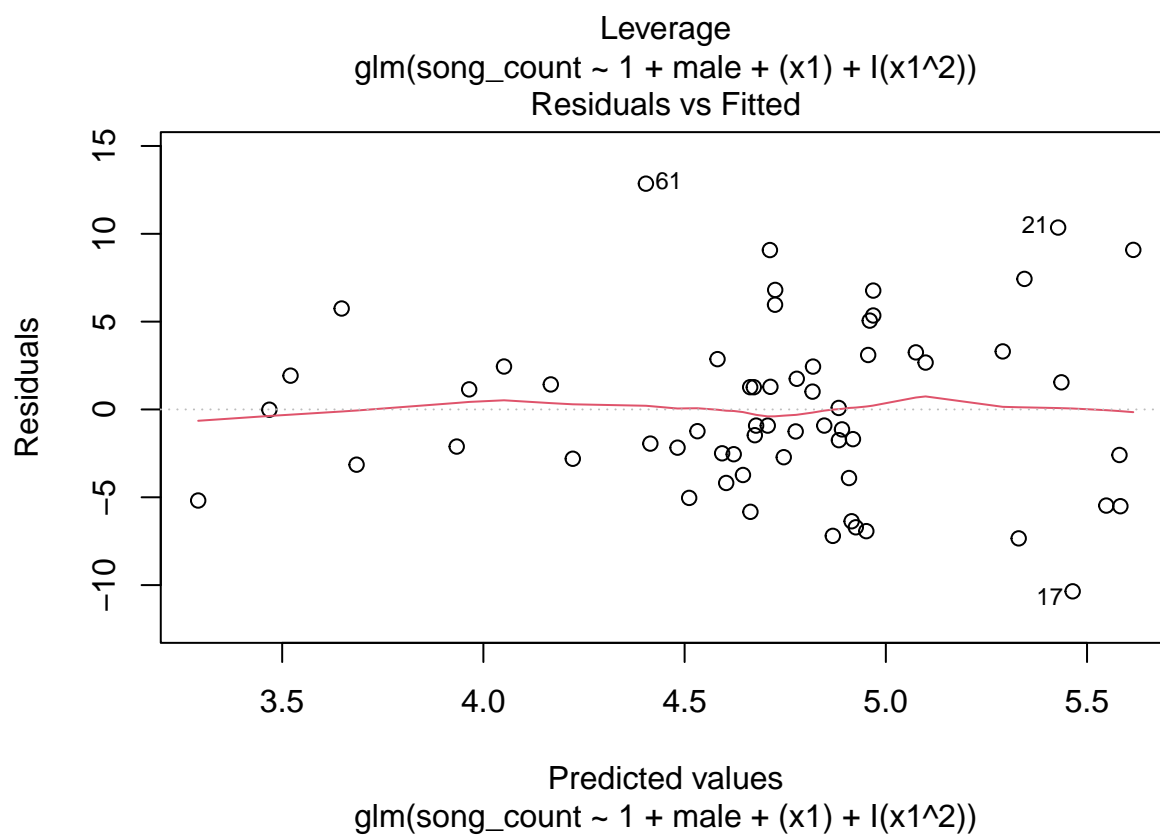
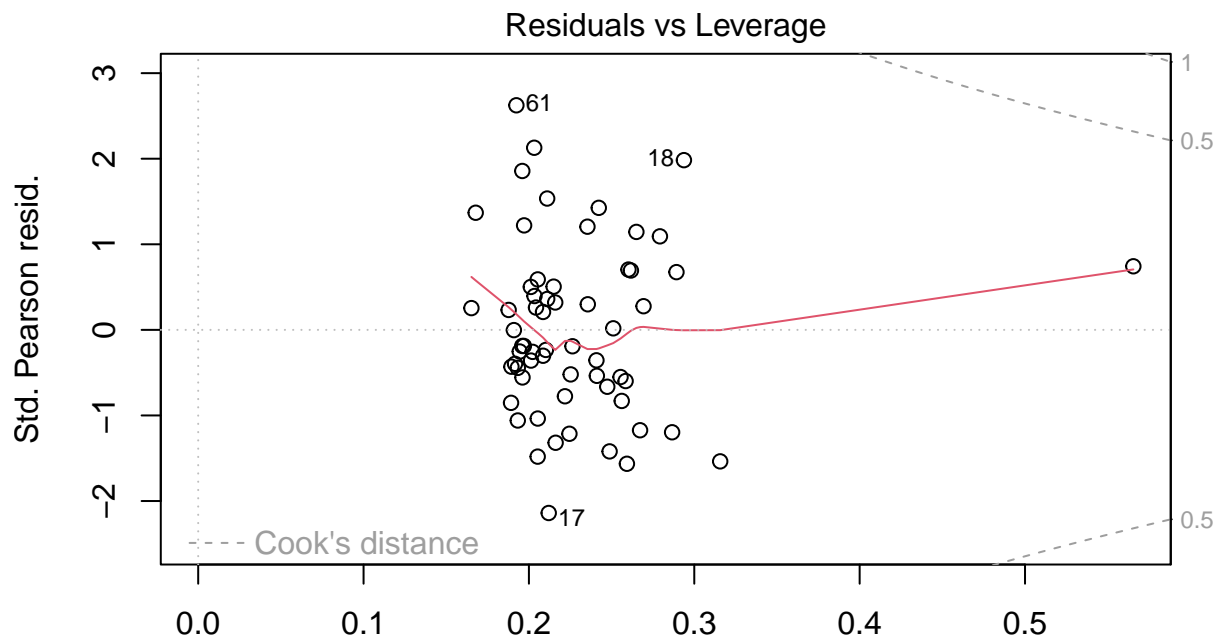
```
## [1] "temp_ref: 44"
```

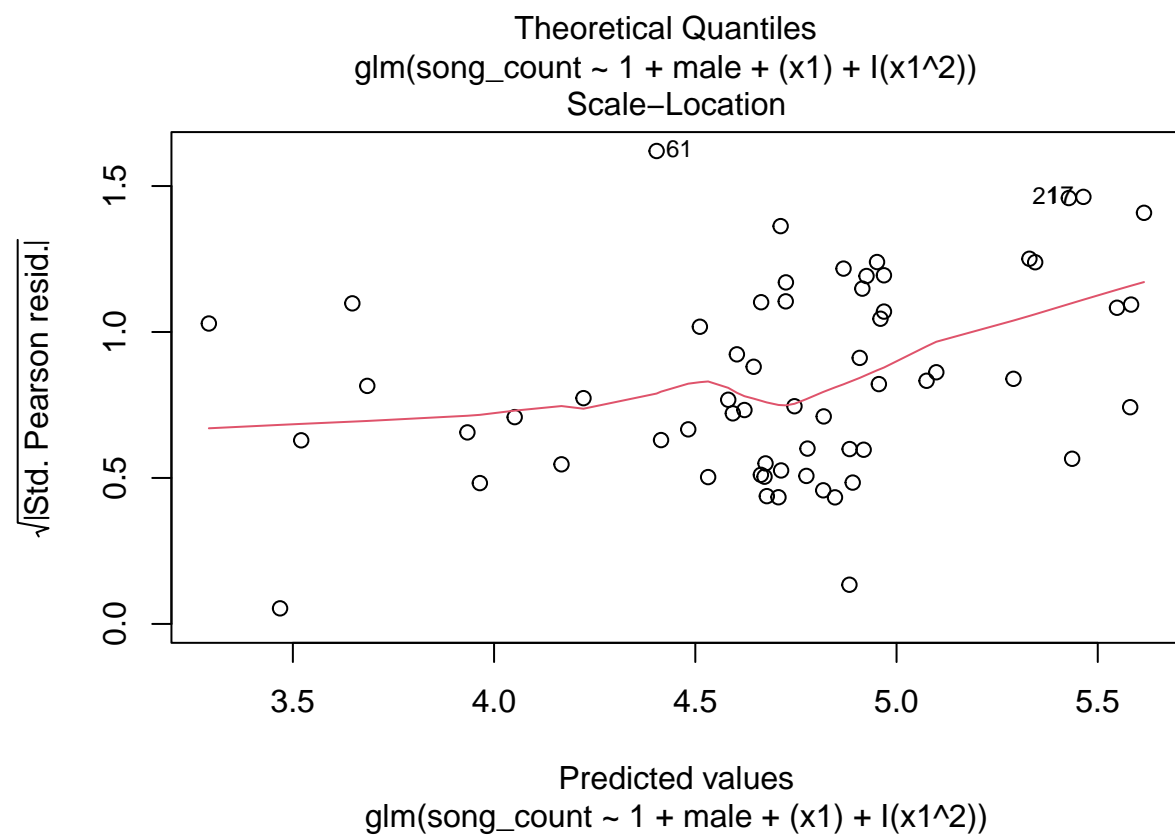
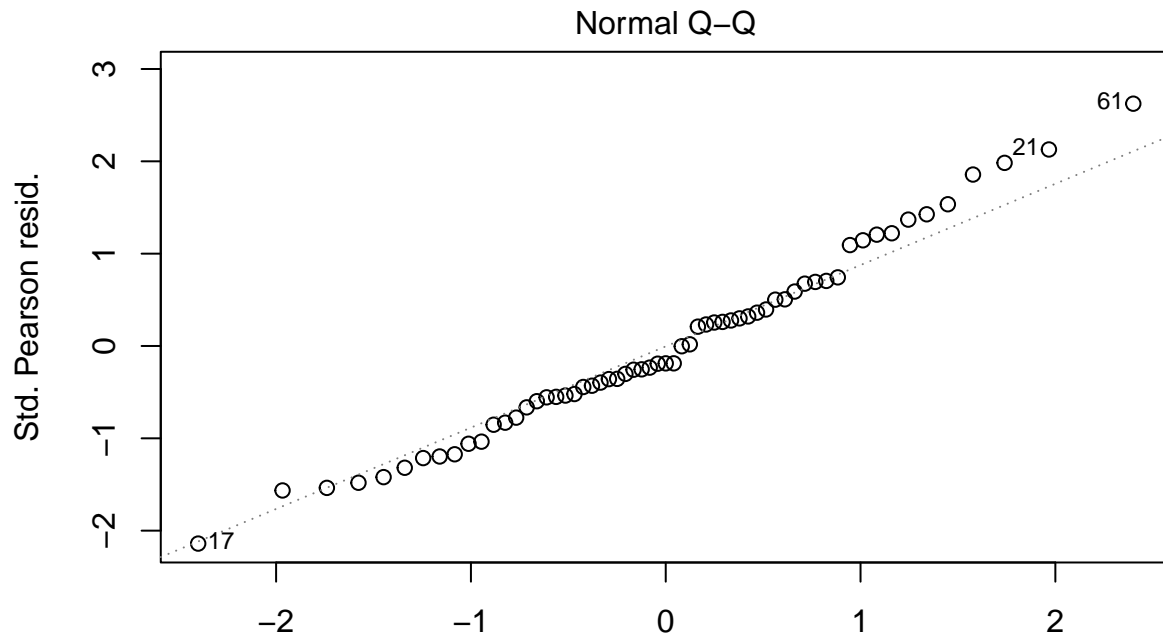


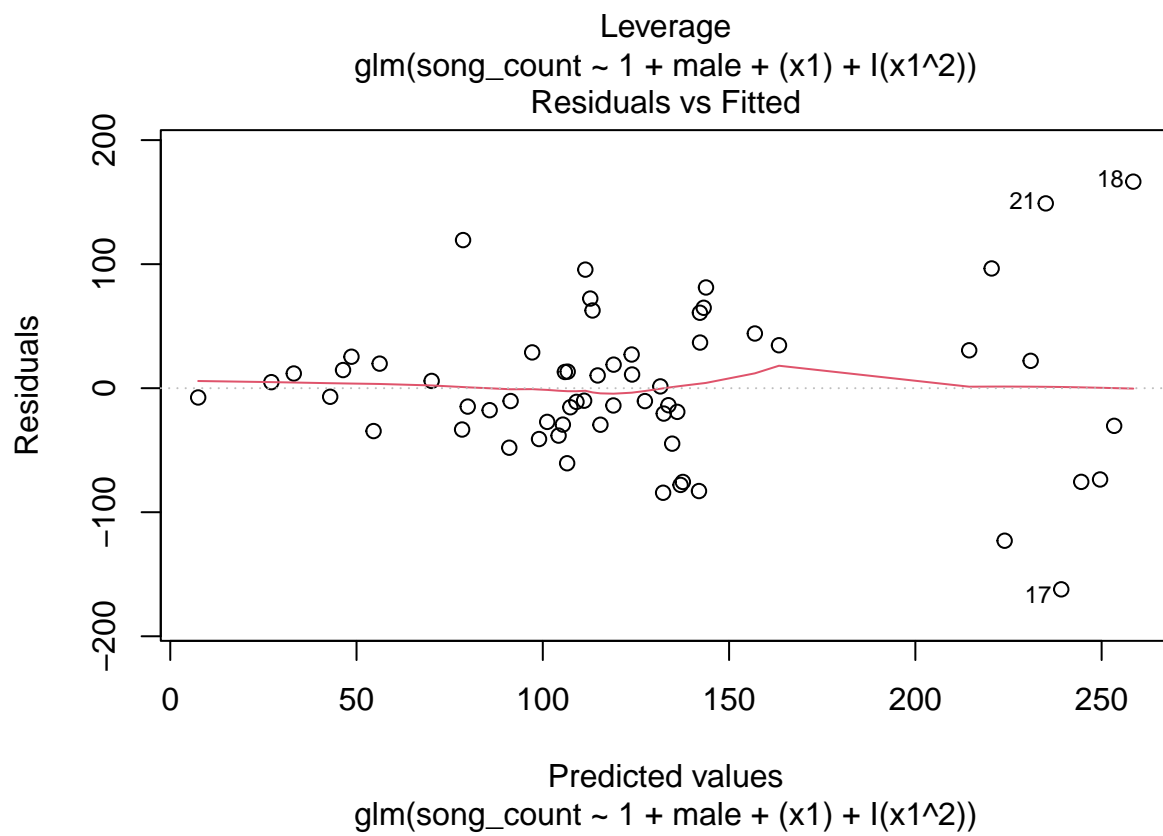
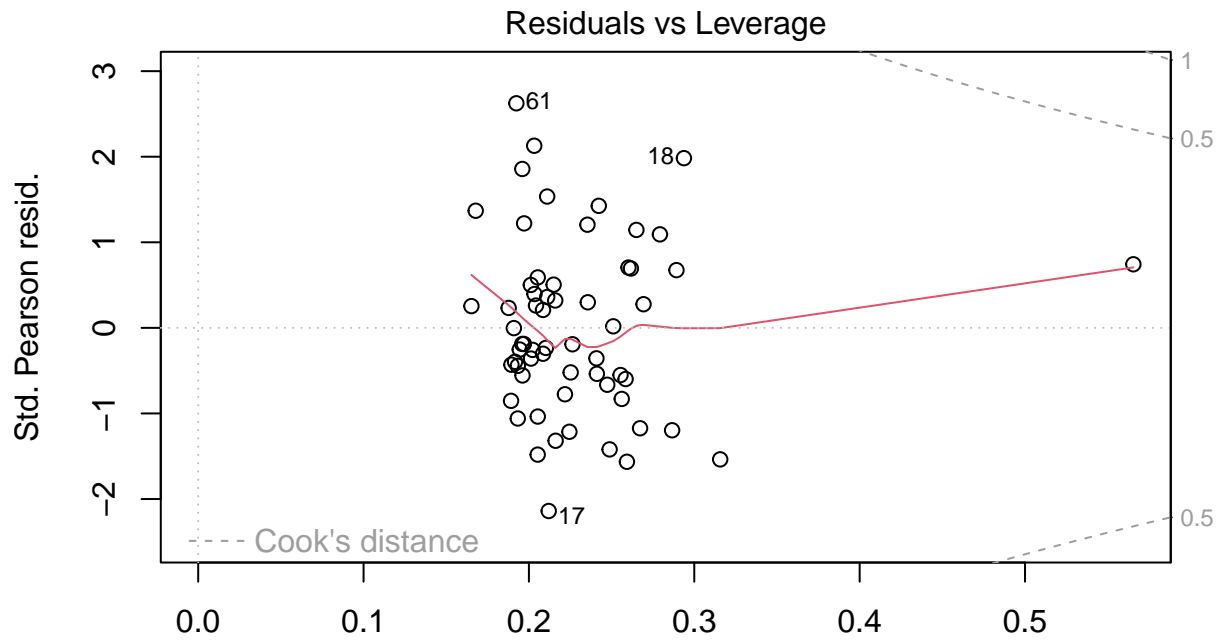


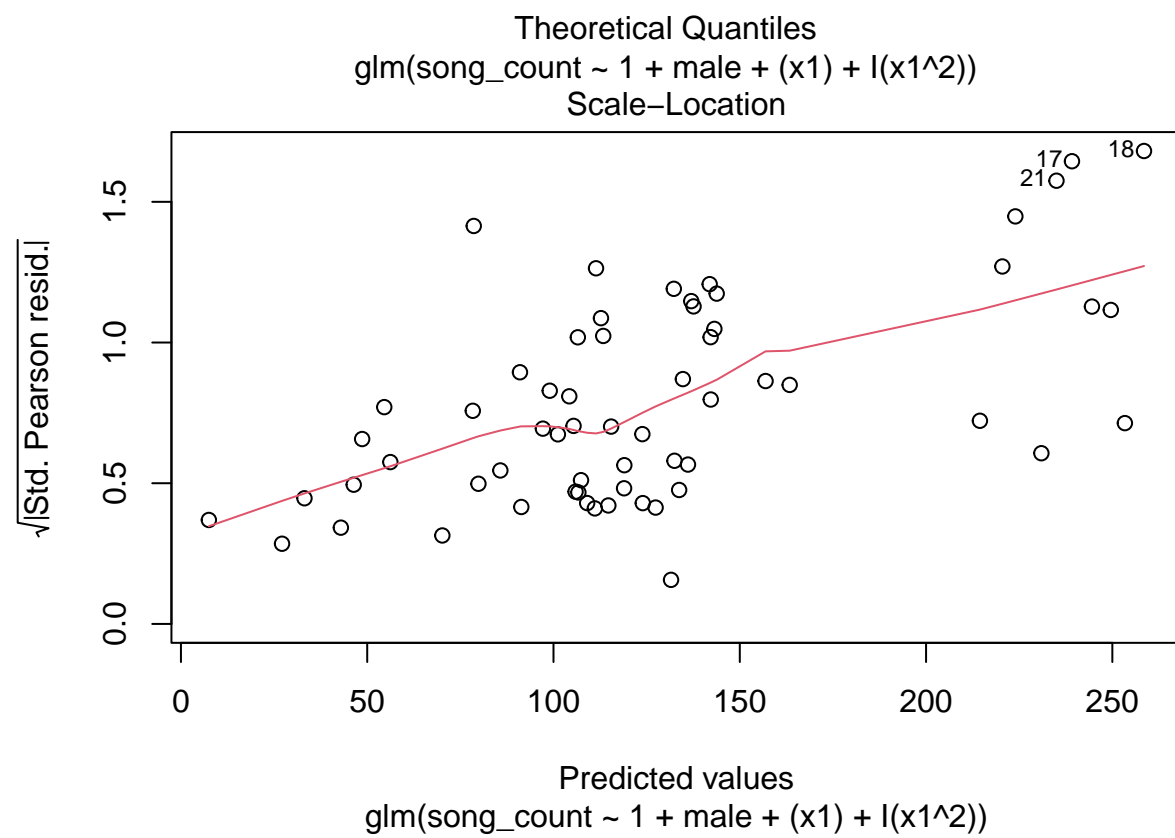
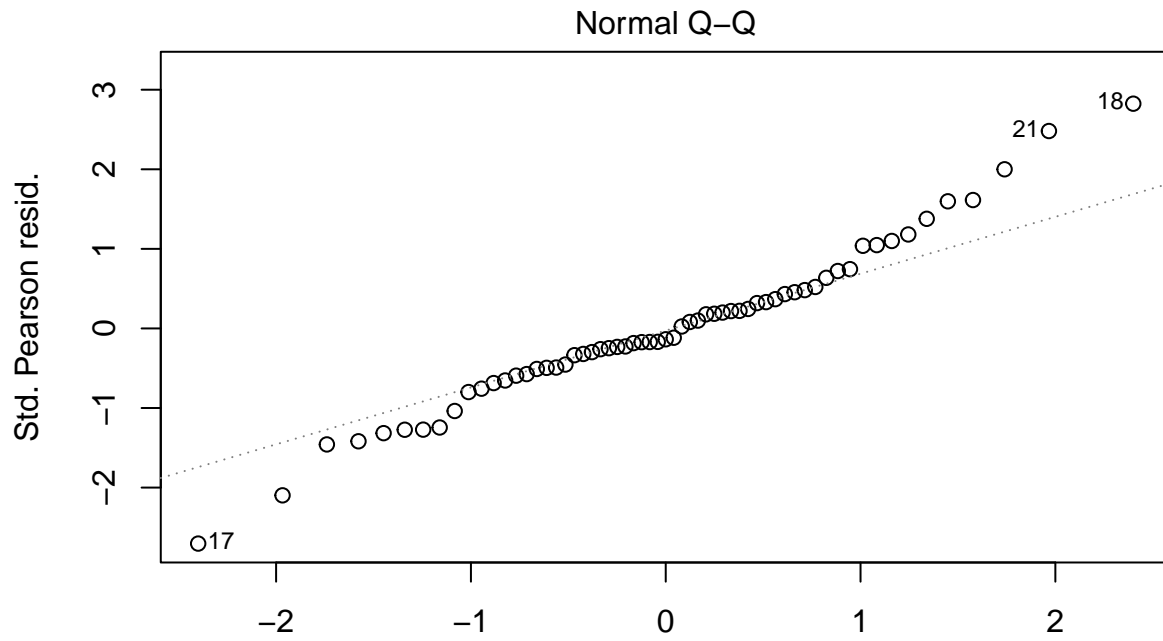


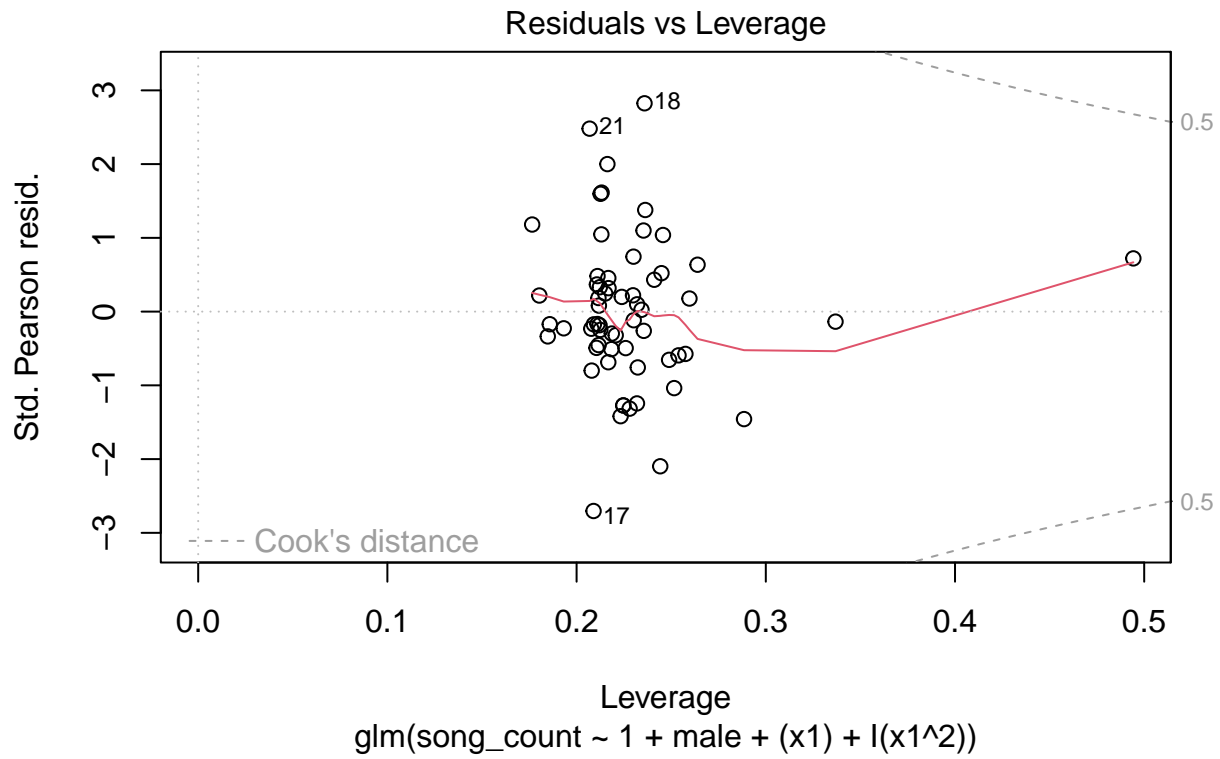




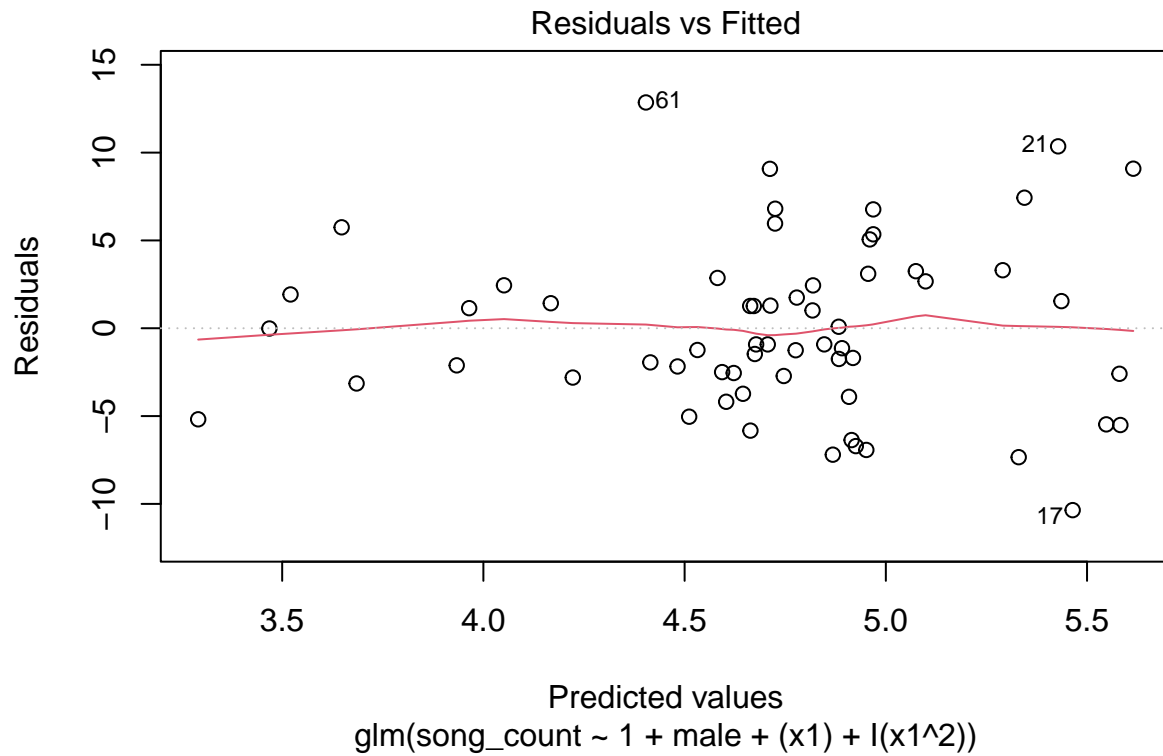


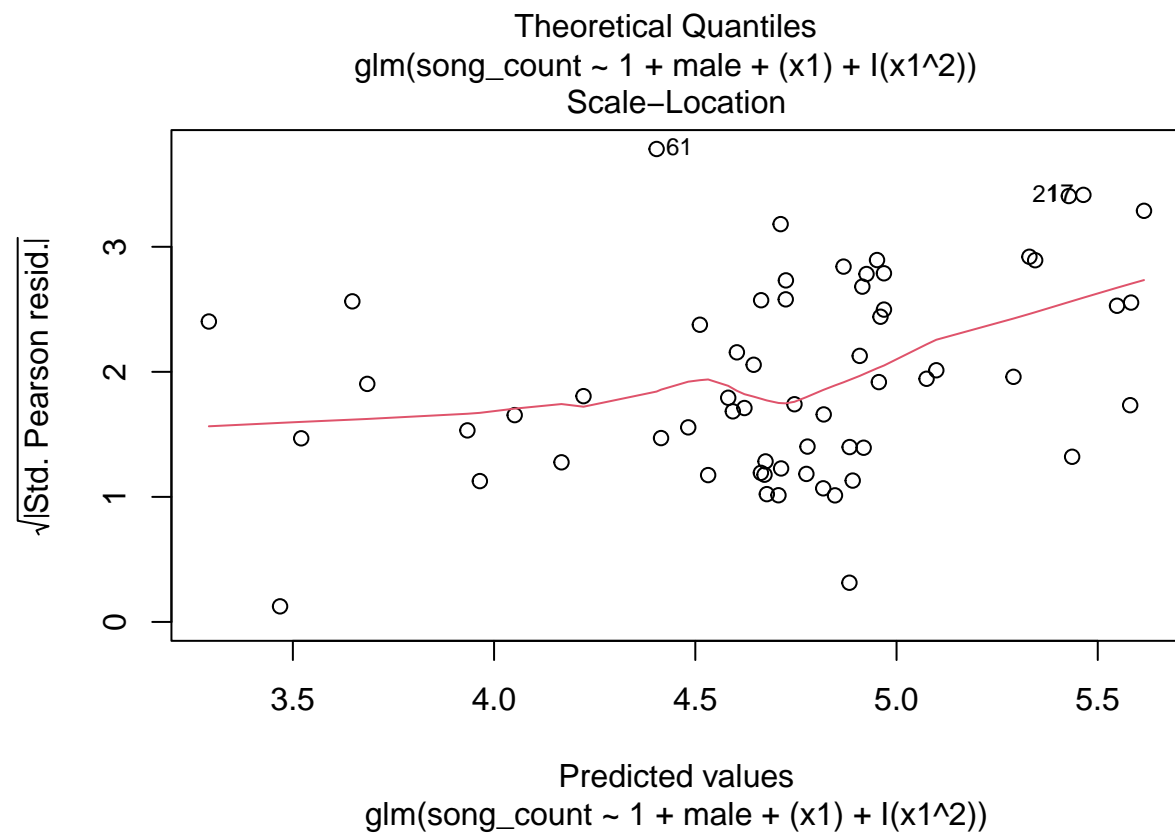
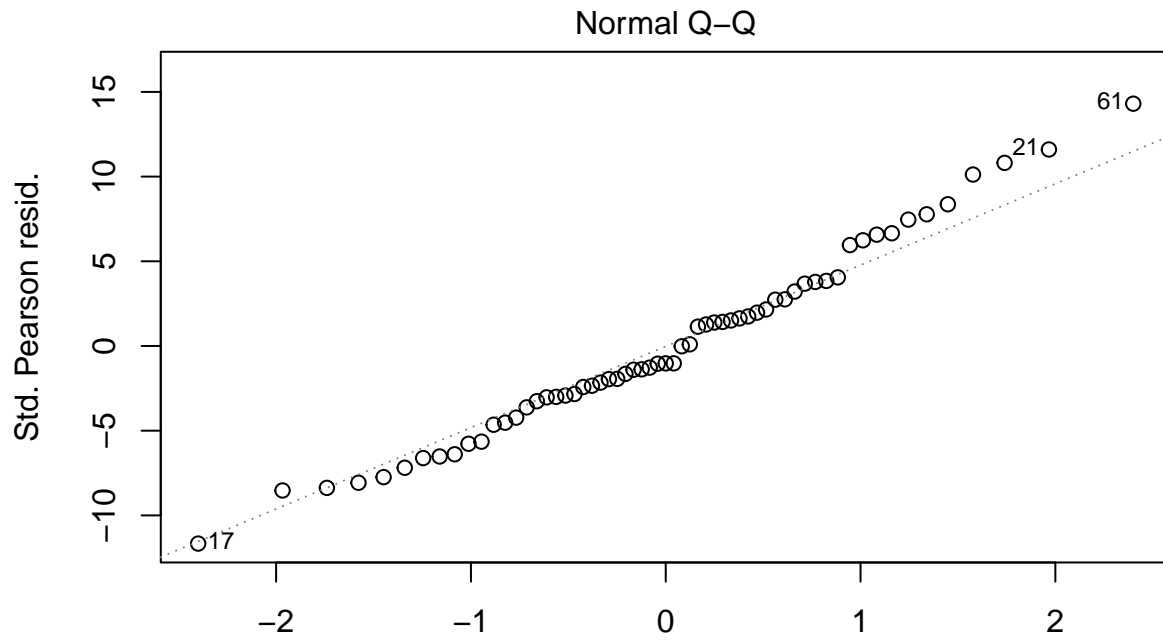


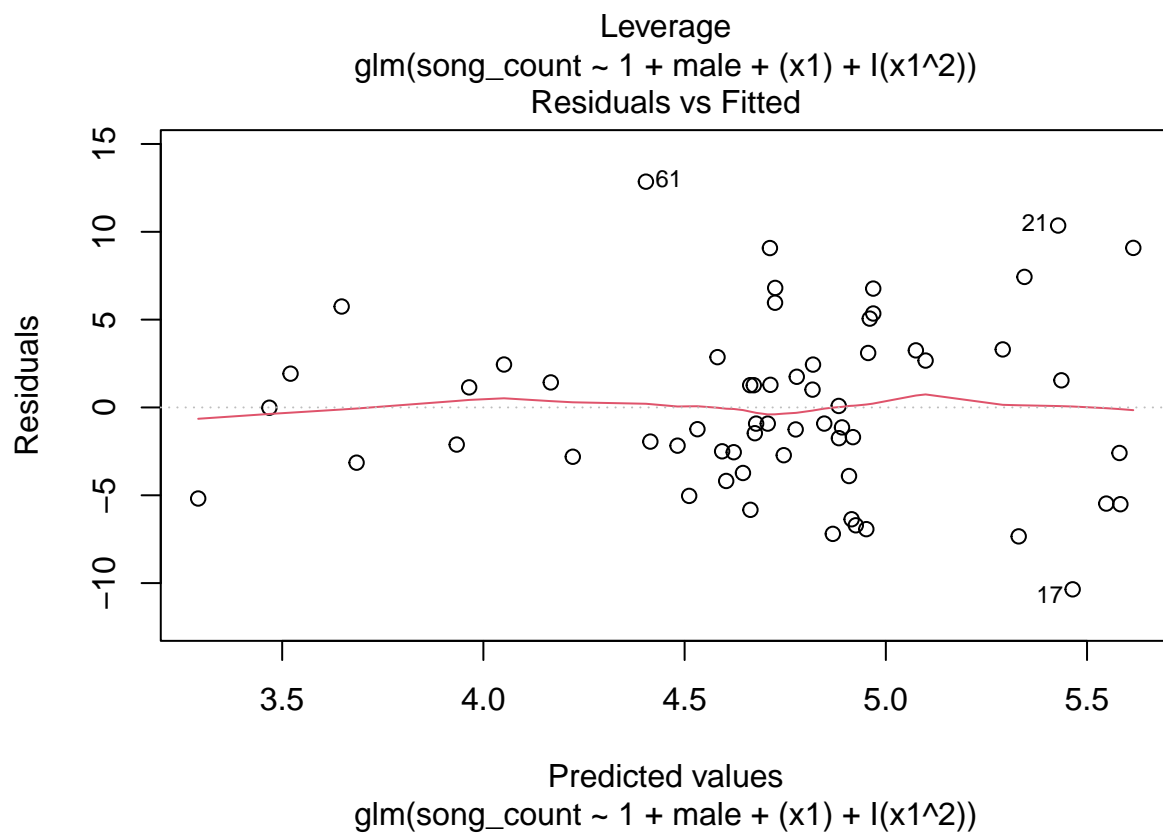
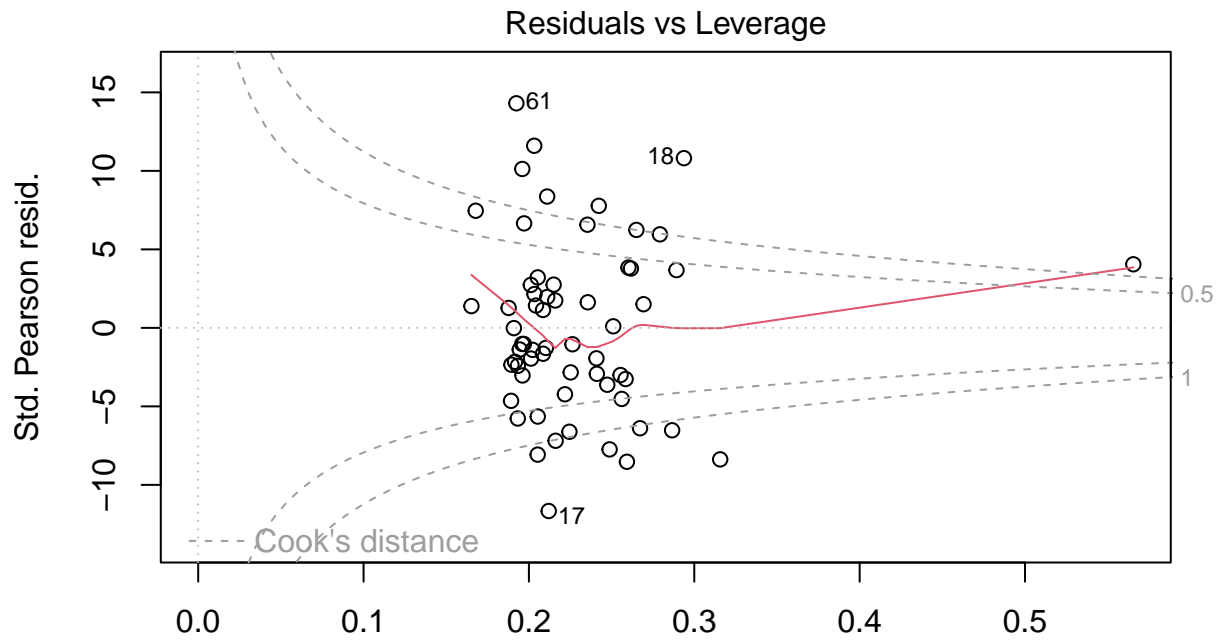


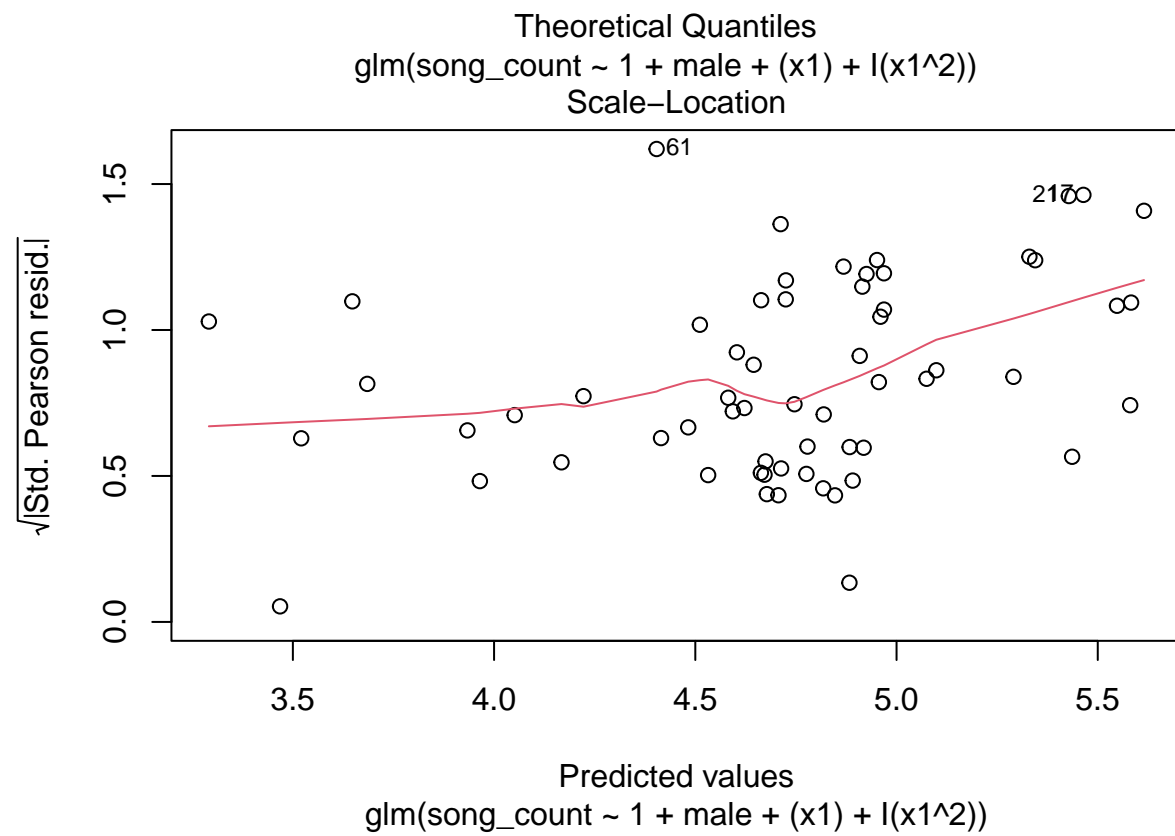
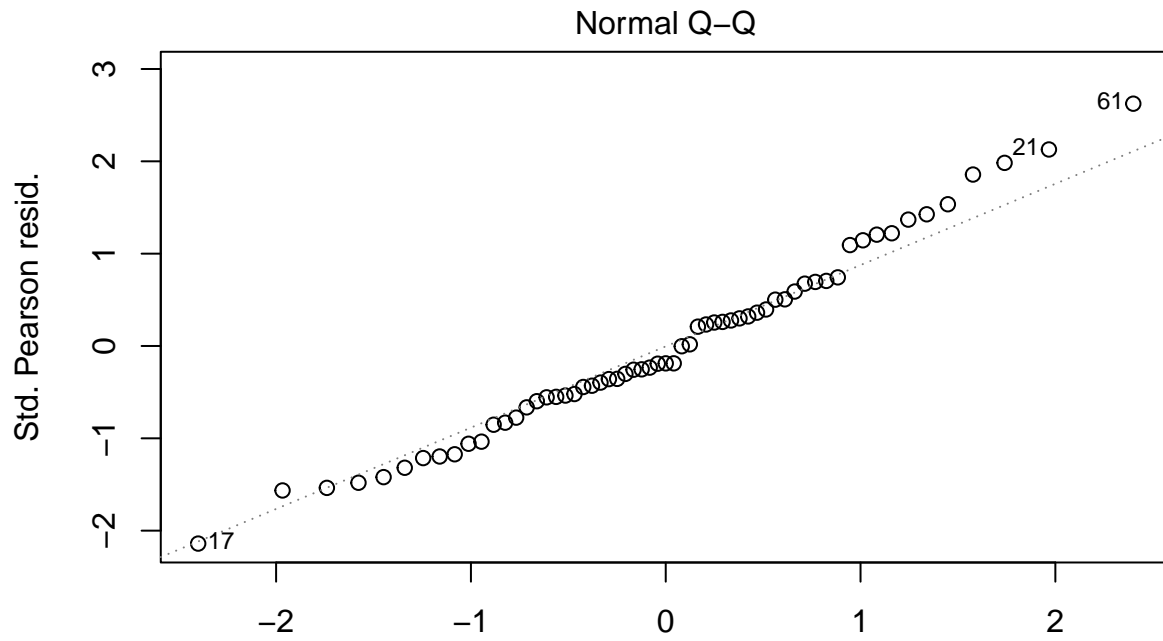


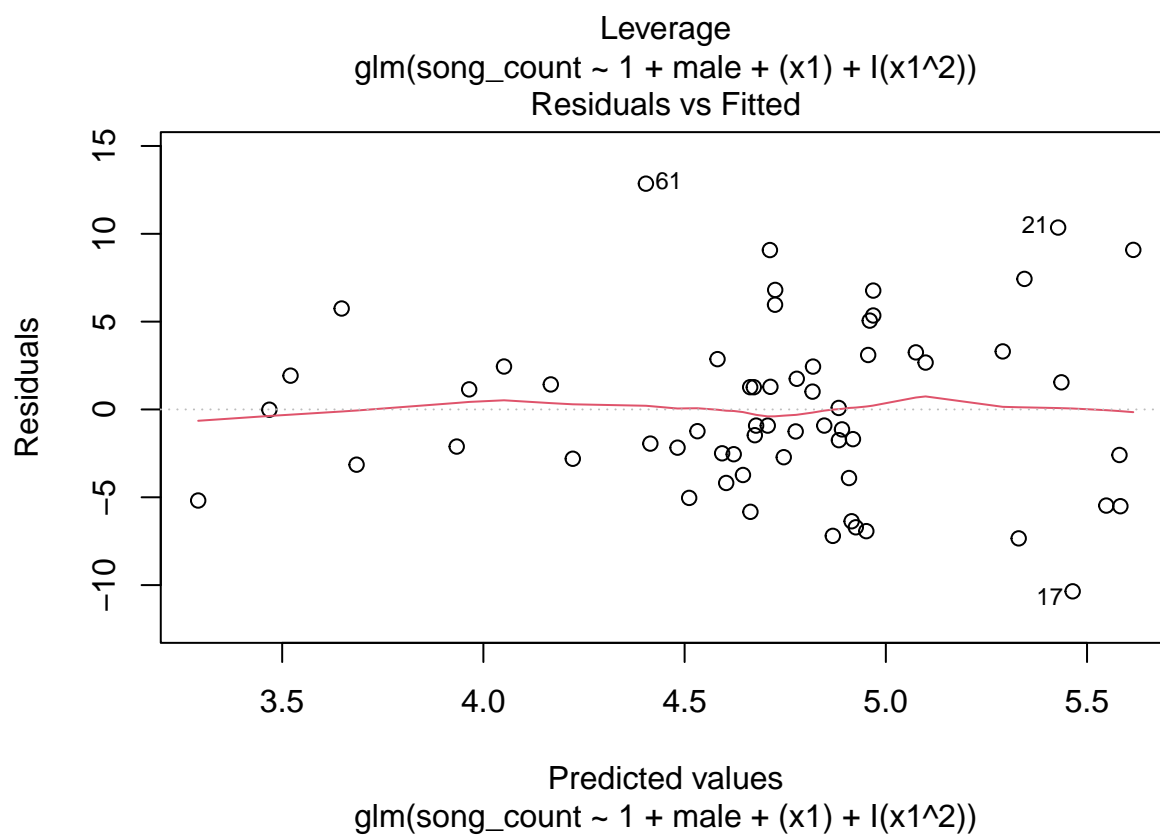
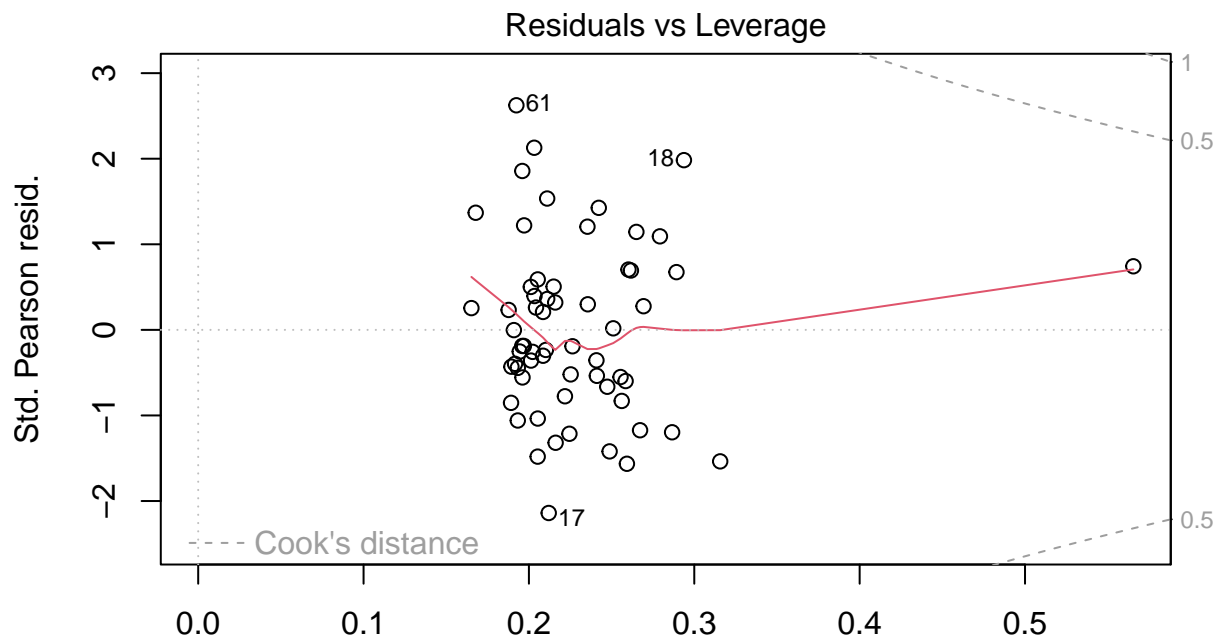
```
## [1] "temp_ref: 45"
```

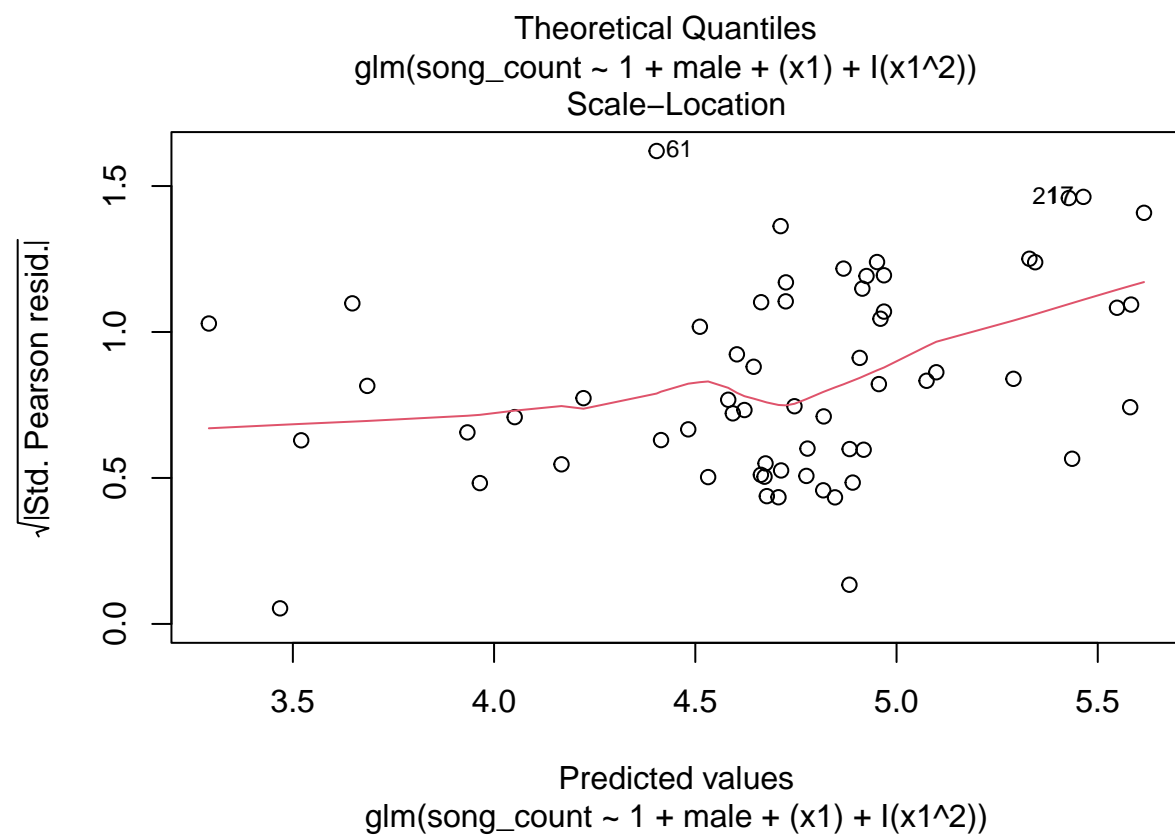
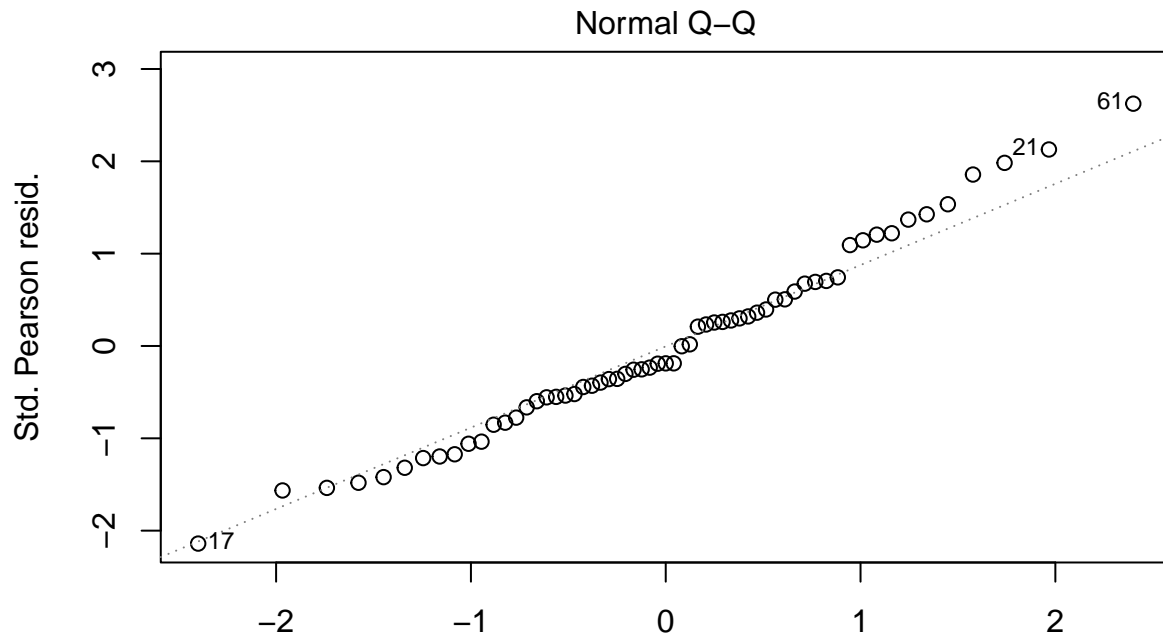


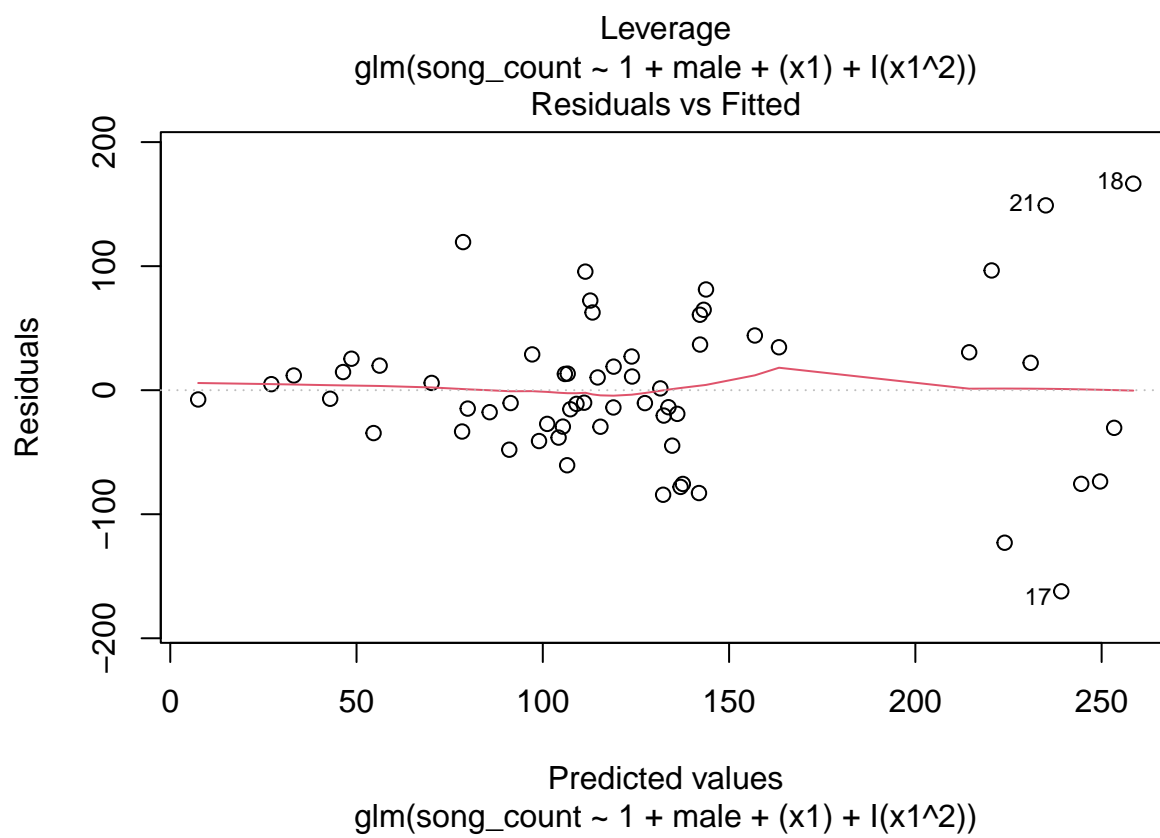
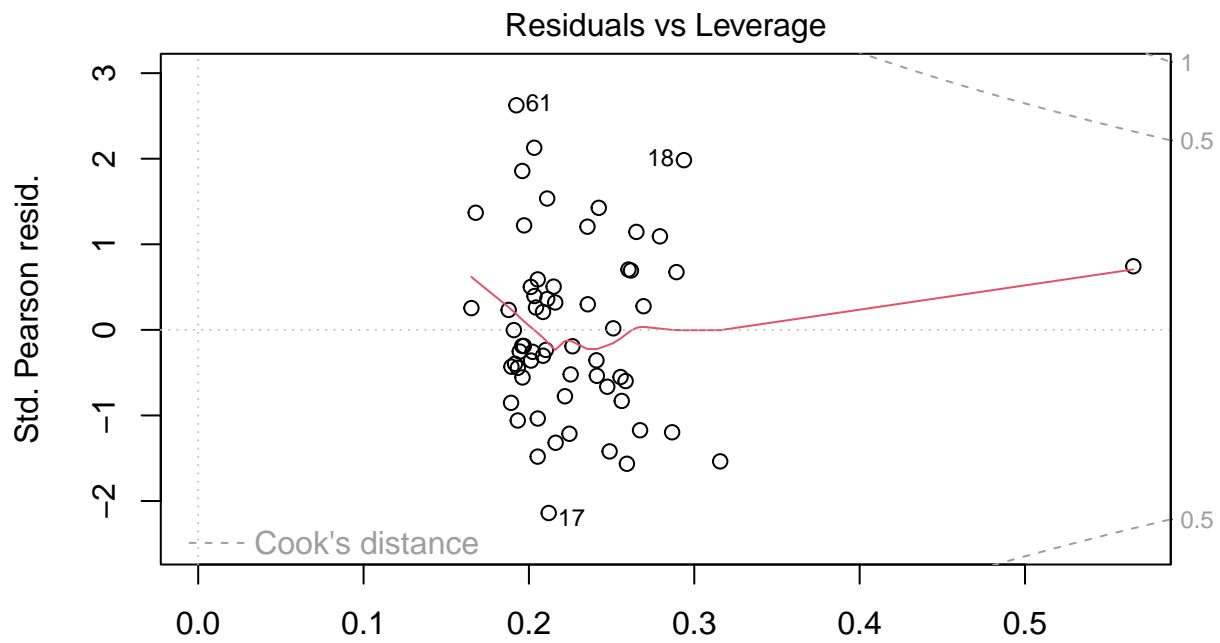


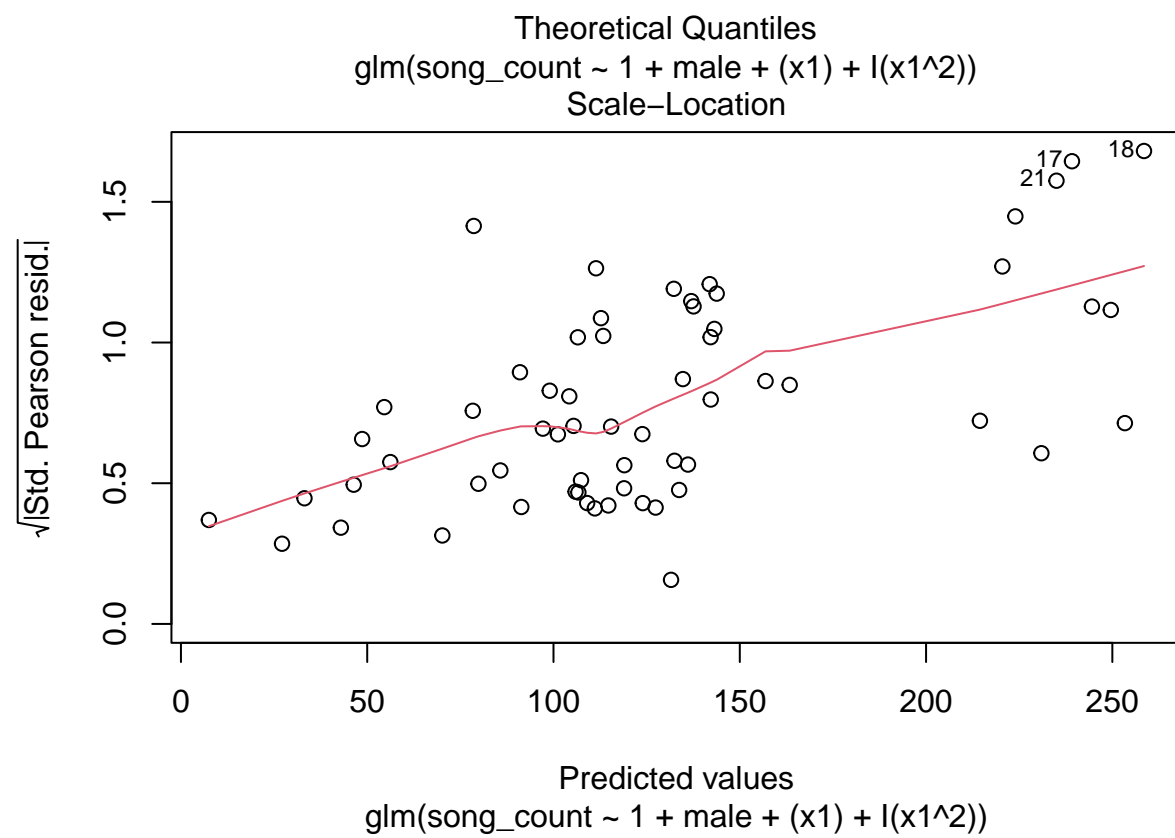
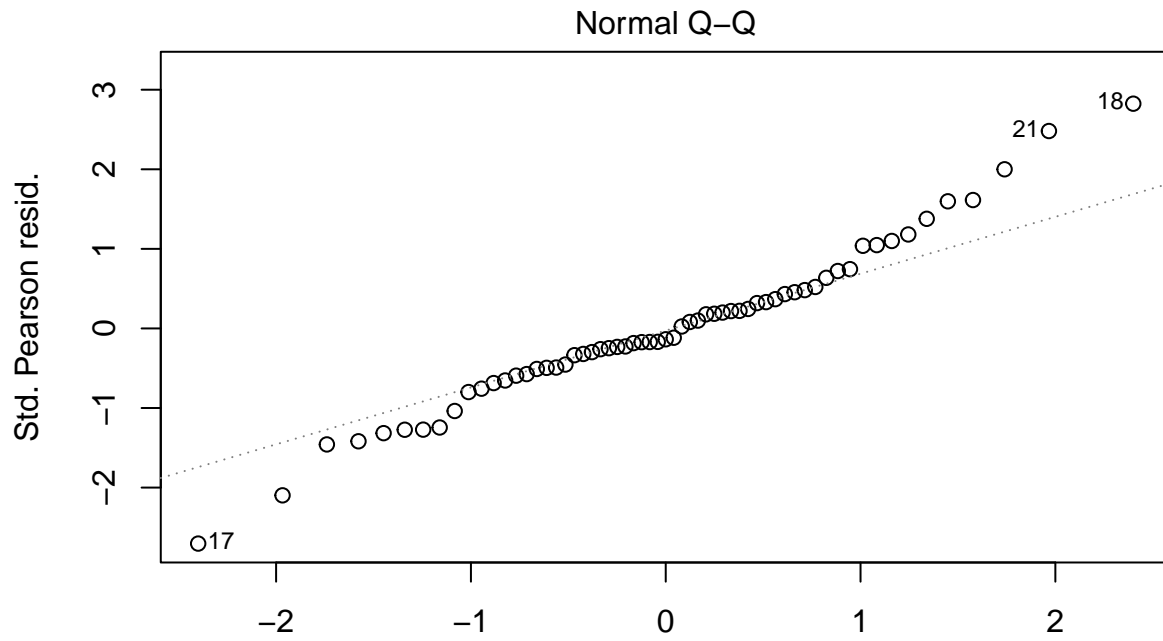


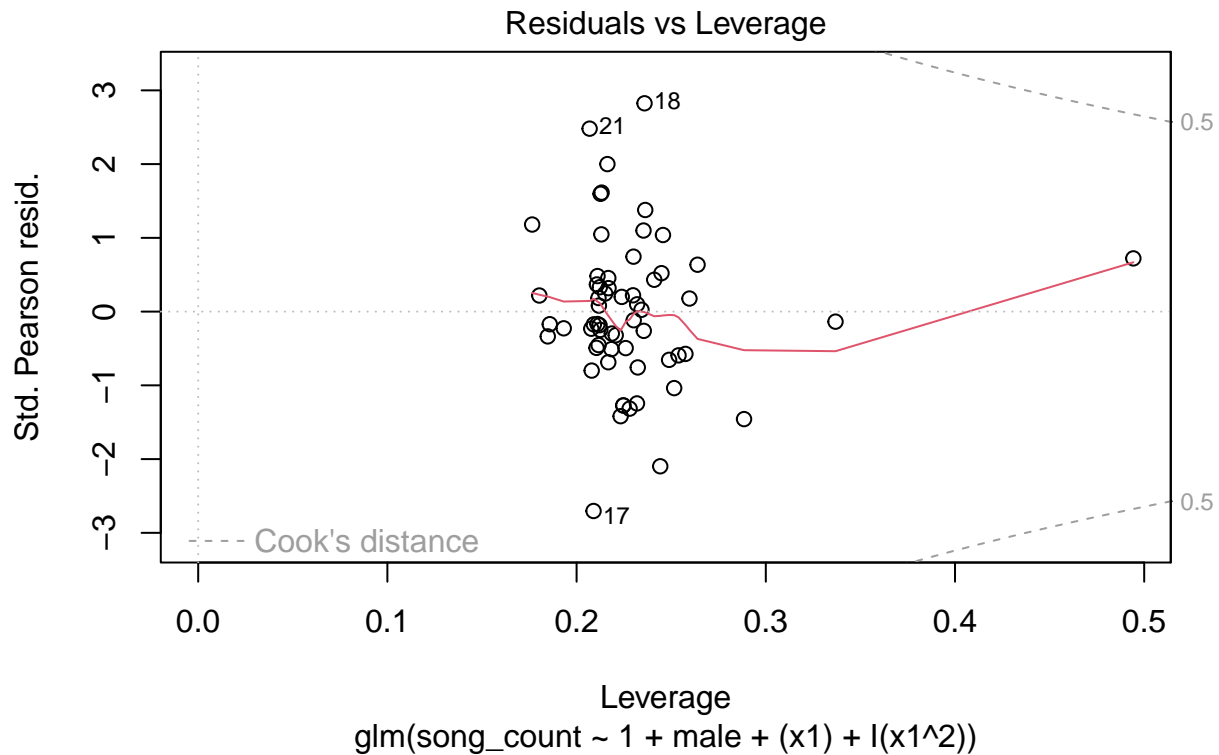












Results

- glm of poisson gives significant x1 and x1^2 terms, but if I adjust for overdispersion using qpoisson they become insignificant.
- I see the same result, enormous CI for x1 and x1^2 terms, with glm_nb.
- These results are the motivation for using RE. That is, to deal with the heterogeneity in a more controlled manner. However, none of the glmer_nb model fittings work well
 - There appears to be a negative correlation between the RE for x1 and x1^2.
 - The variance in the RE are pretty small which makes me wonder why don't the fixed effect models work well enough.

GLMM Fits

No RE co-var: || Formulation

- On reflection the “both” is the only RE setting where || should differ from |

```
est_cov <- FALSE
verbose <- 0
trace <- FALSE
optimizer <- "bobyqa" #"nloptwrap" # "bobyqa"

## Fit random effects models
for(dist in c("poisson", "nb")) {
  for(temp_ref in c(35)) { # vec_temp_ref
    for(re in c("linear", "quad", "both")) {
```

```

switch(re,
  both = (formula_re <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2) |
  linear = (formula_re <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 || male))
  quad = (formula_re <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + I(x1^2) || mal
)
## This model formulation seems to be correct and converges with optimizer = "bobyqa"!!
## H0wever, it ignores the correlation between x1 and I(x1^2)
## formula_quad_RE <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2)||male)
## I'm not sure what that means
if(dist == "poisson"){
  print(paste("temp_ref =", temp_ref, "glmer_poisson", "re = ", re))
  glmer_poisson <- glmer(formula = formula_re,
    data = data,
    control = glmerControl(
      optCtrl = list(maxiter = 1E5,
        maxfun = 2E6,
        trace = trace),
      optimizer=optimizer),
    family = poisson(link = "log"),
    verbose = verbose
  )

  print(summary(glmer_poisson))
  plot(glmer_poisson, ask=FALSE)

  fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glmer" & fit_db$dist == "poisson"

  print("evaluate overdispersion")

  glmer_qpoisson <- overdisp.glmer(glmer_poisson)
}

if(dist == "nb"){
  print(paste("temp_ref =", temp_ref, "glmer_nb", "re = ", re))
  try(glmer_nb <-
    glmer.nb(formula = formula_re,
      data = data,
      ## control values are used by the initial optimization
      ## using a poisson glmer model, which doesn't converge
      control = glmerControl(
        boundary.tol = 0,
        tolPwrss=1e-1,
        optCtrl = list(maxiter = 1E5,
          maxfun = 2E6,
          trace = trace),
        optimizer = optimizer),
      ## nb.control values are used by the second optimizer
      ## Note need to set own optCtrl values
      nb.control = list(
        optCtrl = list(maxit = 1000,
          maxfun = 2E5)),
      verbose = verbose
    )
  )
}

```

```

    )
  )

  plot(glmer_nb, ask = FALSE)
  print(summary(glmer_nb))

  fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glmer" & fit_db$dist == "nb" & fi
}
}
}
}
}

```

```
## [1] "temp_ref = 35 glmer_poisson re = linear"
```

```
## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored
```

```
## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.015591 (tol = 0.002, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly uniden
## - Rescale variables?
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 || male)
## Data: data
## Control: glmerControl(optCtrl = list(maxiter = 1e+05, maxfun = 2e+06,
## trace = trace), optimizer = optimizer)
##
```

```
##      AIC      BIC    logLik deviance df.resid
##  1431.7   1463.3   -700.8   1401.7      46
##
```

```
## Scaled residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -9.4099 -2.3601 -0.3221  1.7150 14.1120
##
```

```
## Random effects:
```

```
## Groups Name Variance Std.Dev.
```

```
## male x1 0.002995 0.05473
```

```
## Number of obs: 61, groups: male, 12
```

```
##
```

```
## Fixed effects:
```

```
##      Estimate Std. Error z value Pr(>|z|)
## maleT229  3.958900  0.091330  43.347 < 2e-16 ***
## maleT231  4.240132  0.089705  47.267 < 2e-16 ***
## maleT234  3.758839  0.115413  32.569 < 2e-16 ***
## maleT235  4.504916  0.082918  54.330 < 2e-16 ***
```

```

## maleT236 3.876809 0.112270 34.531 < 2e-16 ***
## maleT237 4.058857 0.101564 39.964 < 2e-16 ***
## maleT243 4.994830 0.081779 61.077 < 2e-16 ***
## maleT244 3.852043 0.121522 31.698 < 2e-16 ***
## maleT247 4.406083 0.089316 49.331 < 2e-16 ***
## maleT257 5.526669 0.063079 87.615 < 2e-16 ***
## maleT258 2.833279 0.156104 18.150 < 2e-16 ***
## maleT260 4.695259 0.113902 41.222 < 2e-16 ***
## x1 -0.118092 0.021028 -5.616 1.95e-08 ***
## I(x1^2) -0.005618 0.000760 -7.393 1.44e-13 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_poisson), correlation=TRUE) or
## vcov(summary(glmer_poisson)) if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.015591 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## unused control arguments ignored
##
## [1] "evaluate overdispersion"
## Residual deviance: 939.335 on 46 degrees of freedom (ratio: 20.42)
## [1] "temp_ref = 35 glmer_poisson re = quad"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00649726 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + I(x1^2) || male)
## Data: data
## Control: glmerControl(optCtrl = list(maxiter = 1e+05, maxfun = 2e+06,
## trace = trace), optimizer = optimizer)
##
## AIC BIC logLik deviance df.resid
## 1451.5 1483.1 -710.7 1421.5 46
##
## Scaled residuals:

```

```

##      Min      1Q  Median      3Q      Max
## -8.9198 -2.4339 -0.2126  1.5026 14.6714
##
## Random effects:
##   Groups Name   Variance Std.Dev.
##   male     I(x1^2) 8.531e-06 0.002921
## Number of obs: 61, groups:  male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229    4.124778    0.074827  55.124 < 2e-16 ***
## maleT231    4.266849    0.074674  57.140 < 2e-16 ***
## maleT234    3.743742    0.090030  41.583 < 2e-16 ***
## maleT235    4.692110    0.069463  67.549 < 2e-16 ***
## maleT236    4.230641    0.083474  50.682 < 2e-16 ***
## maleT237    4.179025    0.080902  51.656 < 2e-16 ***
## maleT243    4.726213    0.067627  69.886 < 2e-16 ***
## maleT244    3.976604    0.090476  43.952 < 2e-16 ***
## maleT247    4.401292    0.073165  60.155 < 2e-16 ***
## maleT257    5.311227    0.056622  93.802 < 2e-16 ***
## maleT258    3.084881    0.117646  26.222 < 2e-16 ***
## maleT260    4.344404    0.084856  51.197 < 2e-16 ***
## x1          -0.114833    0.014031  -8.184 2.74e-16 ***
## I(x1^2)     -0.005631    0.001153  -4.884 1.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_poisson), correlation=TRUE) or
##      vcov(summary(glmer_poisson))      if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00649726 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## unused control arguments ignored
##
## [1] "evaluate overdispersion"
## Residual deviance: 968.391 on 46 degrees of freedom (ratio: 21.052)
## [1] "temp_ref = 35 glmer_poisson re = both"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##   Family: poisson ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2) ||
##         male)

```

```

## Data: data
## Control: glmerControl(optCtrl = list(maxiter = 1e+05, maxfun = 2e+06,
##   trace = trace), optimizer = optimizer)
##
##      AIC      BIC   logLik deviance df.resid
## 1331.2   1365.0   -649.6   1299.2      45
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -8.240 -2.028 -0.278  1.208 15.096
##
## Random effects:
##   Groups Name      Variance Std.Dev.
## male    x1         0.0782063 0.27965
## male.1 I(x1^2) 0.0002153 0.01467
## Number of obs: 61, groups: male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229    4.153562    0.150360  27.624 <2e-16 ***
## maleT231    4.728800    0.146638  32.248 <2e-16 ***
## maleT234    3.281943    0.216555  15.155 <2e-16 ***
## maleT235    5.094337    0.142709  35.697 <2e-16 ***
## maleT236    3.062421    0.189852  16.131 <2e-16 ***
## maleT237    4.375082    0.178590  24.498 <2e-16 ***
## maleT243    5.027424    0.147155  34.164 <2e-16 ***
## maleT244    3.849734    0.286147  13.454 <2e-16 ***
## maleT247    3.787706    0.186288  20.333 <2e-16 ***
## maleT257    5.654247    0.111432  50.742 <2e-16 ***
## maleT258    1.010165    0.398468   2.535 0.0112 *
## maleT260    6.573144    0.351424  18.704 <2e-16 ***
## x1          -0.114864    0.082535  -1.392 0.1640
## I(x1^2)     -0.005112    0.004342  -1.177 0.2391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_poisson), correlation=TRUE) or
##   vcov(summary(glmer_poisson)) if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## unused control arguments ignored
##
## [1] "evaluate overdispersion"
## Residual deviance: 728.431 on 45 degrees of freedom (ratio: 16.187)
## [1] "temp_ref = 35 glmer_nb re = linear"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

```



```

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.063488 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## unused control arguments ignored

## boundary (singular) fit: see help('isSingular')

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(4.895) ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 || male)
## Data: data
## Control:
## glmerControl(boundary.tol = 0, tolPwrss = 0.1, optCtrl = list(maxiter = 1e+05,
## maxfun = 2e+06, trace = trace), optimizer = optimizer)
##
##      AIC      BIC   logLik deviance df.resid
##    680.2    714.0   -324.1    648.2      45
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.9753 -0.6912 -0.1580  0.5586  3.0699
##
## Random effects:
##   Groups Name Variance Std.Dev.
##   male   x1  3.432e-11 5.859e-06
## Number of obs: 61, groups:  male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229    4.275715    0.301350  14.189  <2e-16 ***
## maleT231    4.301052    0.302327  14.227  <2e-16 ***
## maleT234    3.557692    0.305281  11.654  <2e-16 ***
## maleT235    4.907960    0.316681  15.498  <2e-16 ***
## maleT236    4.432163    0.300701  14.739  <2e-16 ***
## maleT237    4.262585    0.309520  13.772  <2e-16 ***
## maleT243    4.355303    0.289683  15.035  <2e-16 ***
## maleT244    4.002712    0.324817  12.323  <2e-16 ***
## maleT247    4.216963    0.305795  13.790  <2e-16 ***
## maleT257    4.992306    0.291354  17.135  <2e-16 ***
## maleT258    2.935247    0.330662   8.877  <2e-16 ***
## maleT260    3.979770    0.311631  12.771  <2e-16 ***

```

```

## x1      -0.127369   0.062492  -2.038   0.0415 *
## I(x1^2) -0.005606   0.003320  -1.689   0.0913 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_nb), correlation=TRUE) or
##      vcov(summary(glmer_nb))      if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
## unused control arguments ignored
##
## [1] "temp_ref = 35 glmer_nb re = quad"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0064789 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00834887 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?; Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## unused control arguments ignored

## boundary (singular) fit: see help('isSingular')

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(4.9599) ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + I(x1^2) || male)
## Data: data

```

```

## Control:
## glmerControl(boundary.tol = 0, tolPwrss = 0.1, optCtrl = list(maxiter = 1e+05,
##     maxfun = 2e+06, trace = trace), optimizer = optimizer)
##
##      AIC      BIC   logLik deviance df.resid
##    680.2    714.0   -324.1    648.2      45
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.9724 -0.6915 -0.1994  0.5434  3.1766
##
## Random effects:
##   Groups Name      Variance Std.Dev.
##   male    I(x1^2) 0          0
## Number of obs: 61, groups:  male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229    4.216007    0.297743  14.160 <2e-16 ***
## maleT231    4.242947    0.299006  14.190 <2e-16 ***
## maleT234    3.499089    0.301998  11.586 <2e-16 ***
## maleT235    4.844355    0.313310  15.462 <2e-16 ***
## maleT236    4.382738    0.297332  14.740 <2e-16 ***
## maleT237    4.209983    0.306672  13.728 <2e-16 ***
## maleT243    4.289398    0.285830  15.007 <2e-16 ***
## maleT244    3.941226    0.321803  12.247 <2e-16 ***
## maleT247    4.144315    0.301920  13.727 <2e-16 ***
## maleT257    4.914289    0.286369  17.161 <2e-16 ***
## maleT258    2.869099    0.327170   8.769 <2e-16 ***
## maleT260    3.910894    0.308047  12.696 <2e-16 ***
## x1          -0.143609    0.061788  -2.324  0.0201 *
## I(x1^2)     -0.006424    0.003283  -1.957  0.0504 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_nb), correlation=TRUE) or
##     vcov(summary(glmer_nb))          if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
## unused control arguments ignored
##
## [1] "temp_ref = 35 glmer_nb re = both"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

```

```

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## unused control arguments ignored

## boundary (singular) fit: see help('isSingular')

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(5.2598) ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2) ||
## male)
## Data: data
## Control:
## glmerControl(boundary.tol = 0, tolPwrss = 0.1, optCtrl = list(maxiter = 1e+05,
## maxfun = 2e+06, trace = trace), optimizer = optimizer)
##
##      AIC      BIC   logLik deviance df.resid
##  696.2    732.1   -331.1    662.2      44
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.0414 -0.7625 -0.2002  0.4121  2.2875
##
## Random effects:
## Groups Name      Variance Std.Dev.
## male  x1         1.045e-03 3.232e-02
## male.1 I(x1^2) 2.891e-11 5.377e-06
## Number of obs: 61, groups:  male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229  4.4518665   0.2984930  14.914 < 2e-16 ***
## maleT231  4.1955448   0.2744380  15.288 < 2e-16 ***
## maleT234  3.8829865   0.2980811  13.027 < 2e-16 ***
## maleT235  4.7403106   0.2999752  15.802 < 2e-16 ***
## maleT236  4.0008904   0.3171432  12.615 < 2e-16 ***
## maleT237  4.8246073   0.3442351  14.015 < 2e-16 ***
## maleT243  4.7482484   0.2767484  17.157 < 2e-16 ***
## maleT244  3.8724934   0.3140062  12.333 < 2e-16 ***
## maleT247  4.5707888   0.3018051  15.145 < 2e-16 ***
## maleT257  5.3796854   0.2805020  19.179 < 2e-16 ***
## maleT258  2.9799895   0.3276061   9.096 < 2e-16 ***
## maleT260  4.3916282   0.2994277  14.667 < 2e-16 ***
## x1        -0.0958858   0.0161136  -5.951 2.67e-09 ***
## I(x1^2)   -0.0044420   0.0001864 -23.834 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_nb), correlation=TRUE) or
##      vcov(summary(glmer_nb))      if you need it

```

```
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
## unused control arguments ignored
```

Results

- `glmer_poisson` and `glmer_nb` only converge under certain `re` and only when `optimizer = "bobyqa"`.
- `glmer_poisson` summary
 - `re = "linear"`: Singular/boundary fit
 - `re = "quad"`: Fails to converge
 - `re = "both"`: Converges, `overdisp.glmer` indicates data is greatly overdispersed.
- `glmer_nb` summary
 - `re = "linear"`: Singular/boundary fit
 - `re = "quad"`: Singular/boundary fit
 - `re = "both"`: Singular/boundary fit
- It's note worthy that `glmer_nb` gives a singular value for the variance (i.e. effectively 0) for `x1` using `optimizer = "bobyqa"`, but gives the variance (i.e. effectively 0) for `x1^2` using `optimizer = "nloptwrap"`. This seems consistent with them being highly correlated as suggested in the || fittings.
- The `x1^2` term is often (always?) non-significant in the `nb` models

| Formulation

```
est_cov <- TRUE
verbose <- 0
trace <- FALSE
optimizer <- "bobyqa" #"nloptwrap" # "bobyqa"

## Fit random effects models
for(dist in c("poisson", "nb")) {
  for(temp_ref in c(35)) { # vec_temp_ref ){
    for(re in c("linear", "quad", "both")) {

      switch(re,
        both = (formula_re <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2) |
        linear = (formula_re <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 | male)),
        quad = (formula_re <- song_count ~ -1 + male + x1 + I(x1^2) + (-1 + I(x1^2) | male
        )

      if(dist == "poisson"){
        print(paste("temp_ref =", temp_ref, "glmer_poisson", "re = ", re))
        glmer_poisson <- glmer(formula = formula_re,
                              data = data,
                              control = glmerControl(
                                optCtrl = list(maxiter = 1E5,
                                                maxfun = 2E6,
                                                trace = trace),
                                optimizer=optimizer),
                              family = poisson(link = "log"),
```

```

        verbose = verbose
    )

    print(summary(glmer_poisson))
    plot(glmer_poisson, ask=FALSE)

    fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glmer" & fit_db$dist == "poisson"]

    print("evaluate overdispersion")

    glmer_qpoisson <- overdisp.glmer(glmer_poisson)

}

if(dist == "nb"){
    print(paste("temp_ref =", temp_ref, "glmer_nb", "re = ", re))
    try(glmer_nb <-
        glmer.nb(formula = formula_re,
            data = data,
            ## control values are used by the initial optimization
            ## using a poisson glmer model, which doesn't converge
            control = glmerControl(
                boundary.tol = 0,
                tolPwrss=1e-1,
                optCtrl = list(maxiter = 1E5,
                    maxfun = 2E6,
                    trace = trace),
                optimizer = optimizer),
            ## nb.control values are used by the second optimizer
            ## Note need to set own optCtrl values
            nb.control = list(
                optCtrl = list(maxit = 1000,
                    maxfun = 2E5)),
            verbose = verbose
        )

    plot(glmer_nb, ask = FALSE)
    print(summary(glmer_nb))

    fit_db[fit_db$temp_ref==temp_ref & fit_db$model == "glmer" & fit_db$dist == "nb" & fi

}

}

}

```

```
## [1] "temp_ref = 35 glmer_poisson re = linear"
```

```
## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored
```

```
## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored
```

```

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.015591 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 | male)
## Data: data
## Control: glmerControl(optCtrl = list(maxiter = 1e+05, maxfun = 2e+06,
## trace = trace), optimizer = optimizer)
##
##      AIC      BIC   logLik deviance df.resid
## 1431.7   1463.3   -700.8   1401.7       46
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -9.4099 -2.3601 -0.3221  1.7150 14.1120
##
## Random effects:
## Groups Name Variance Std.Dev.
## male x1 0.002995 0.05473
## Number of obs: 61, groups: male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229 3.958900    0.091330 43.347 < 2e-16 ***
## maleT231 4.240132    0.089705 47.267 < 2e-16 ***
## maleT234 3.758839    0.115413 32.569 < 2e-16 ***
## maleT235 4.504916    0.082918 54.330 < 2e-16 ***
## maleT236 3.876809    0.112270 34.531 < 2e-16 ***
## maleT237 4.058857    0.101564 39.964 < 2e-16 ***
## maleT243 4.994830    0.081779 61.077 < 2e-16 ***
## maleT244 3.852043    0.121522 31.698 < 2e-16 ***
## maleT247 4.406083    0.089316 49.331 < 2e-16 ***
## maleT257 5.526669    0.063079 87.615 < 2e-16 ***
## maleT258 2.833279    0.156104 18.150 < 2e-16 ***
## maleT260 4.695259    0.113902 41.222 < 2e-16 ***
## x1      -0.118092    0.021028 -5.616 1.95e-08 ***
## I(x1^2) -0.005618    0.000760 -7.393 1.44e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_poisson), correlation=TRUE) or
##      vcov(summary(glmer_poisson))      if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.015591 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue

```

```

## - Rescale variables?
## unused control arguments ignored
##
## [1] "evaluate overdispersion"
## Residual deviance: 939.335 on 46 degrees of freedom (ratio: 20.42)
## [1] "temp_ref = 35 glmer_poisson re = quad"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00649726 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable
## - Rescale variables?

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + I(x1^2) | male)
## Data: data
## Control: glmerControl(optCtrl = list(maxiter = 1e+05, maxfun = 2e+06,
## trace = trace), optimizer = optimizer)
##
##      AIC      BIC    logLik deviance df.resid
##  1451.5   1483.1   -710.7   1421.5      46
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -8.9198 -2.4339 -0.2126  1.5026 14.6714
##
## Random effects:
## Groups Name      Variance Std.Dev.
## male    I(x1^2) 8.531e-06 0.002921
## Number of obs: 61, groups: male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229    4.124778    0.074827  55.124 < 2e-16 ***
## maleT231    4.266849    0.074674  57.140 < 2e-16 ***
## maleT234    3.743742    0.090030  41.583 < 2e-16 ***
## maleT235    4.692110    0.069463  67.549 < 2e-16 ***
## maleT236    4.230641    0.083474  50.682 < 2e-16 ***
## maleT237    4.179025    0.080902  51.656 < 2e-16 ***
## maleT243    4.726213    0.067627  69.886 < 2e-16 ***
## maleT244    3.976604    0.090476  43.952 < 2e-16 ***
## maleT247    4.401292    0.073165  60.155 < 2e-16 ***
## maleT257    5.311227    0.056622  93.802 < 2e-16 ***
## maleT258    3.084881    0.117646  26.222 < 2e-16 ***
## maleT260    4.344404    0.084856  51.197 < 2e-16 ***

```



```

## x1          -0.114833    0.014031   -8.184 2.74e-16 ***
## I(x1^2)     -0.005631    0.001153   -4.884 1.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_poisson), correlation=TRUE) or
##      vcov(summary(glmer_poisson))      if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00649726 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## unused control arguments ignored
##
## [1] "evaluate overdispersion"
## Residual deviance: 968.391 on 46 degrees of freedom (ratio: 21.052)
## [1] "temp_ref = 35 glmer_poisson re = both"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0254573 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2) |
##      male)
## Data: data
## Control: glmerControl(optCtrl = list(maxiter = 1e+05, maxfun = 2e+06,
##      trace = trace), optimizer = optimizer)
##
##      AIC      BIC    logLik deviance df.resid
## 1293.6   1329.5   -629.8   1259.6      44
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -8.2553 -1.9791 -0.2244  1.2558 15.0288
##
## Random effects:
## Groups Name      Variance Std.Dev. Corr
## male  x1         0.099106 0.31481
##      I(x1^2) 0.000276 0.01661  0.98

```

```

## Number of obs: 61, groups:  male, 12
##
## Fixed effects:
##      Estimate Std. Error z value Pr(>|z|)
## maleT229  4.157936   0.151855  27.381  <2e-16 ***
## maleT231  4.743229   0.147972  32.055  <2e-16 ***
## maleT234  3.251286   0.221754  14.662  <2e-16 ***
## maleT235  5.107784   0.143705  35.544  <2e-16 ***
## maleT236  3.095568   0.189778  16.311  <2e-16 ***
## maleT237  4.389249   0.180169  24.362  <2e-16 ***
## maleT243  5.030252   0.149097  33.738  <2e-16 ***
## maleT244  3.857537   0.293272  13.153  <2e-16 ***
## maleT247  3.763094   0.189936  19.812  <2e-16 ***
## maleT257  5.671076   0.112336  50.483  <2e-16 ***
## maleT258  0.695339   0.424577   1.638    0.101
## maleT260  6.912868   0.354188  19.518  <2e-16 ***
## x1        -0.111849   0.092585  -1.208    0.227
## I(x1^2)   -0.004894   0.004895  -1.000    0.317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_poisson), correlation=TRUE) or
##      vcov(summary(glmer_poisson))      if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0254573 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## unused control arguments ignored
##
## [1] "evaluate overdispersion"
## Residual deviance: 724.424 on 44 degrees of freedom (ratio: 16.464)
## [1] "temp_ref = 35 glmer_nb re = linear"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.063488 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

```

```

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## unused control arguments ignored

## boundary (singular) fit: see help('isSingular')

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(4.895) ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 | male)
## Data: data
## Control:
## glmerControl(boundary.tol = 0, tolPwrss = 0.1, optCtrl = list(maxiter = 1e+05,
## maxfun = 2e+06, trace = trace), optimizer = optimizer)
##
##      AIC      BIC    logLik deviance df.resid
##    680.2    714.0   -324.1    648.2      45
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.9753 -0.6912 -0.1580  0.5586  3.0699
##
## Random effects:
## Groups Name Variance Std.Dev.
## male  x1   3.432e-11 5.859e-06
## Number of obs: 61, groups: male, 12
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## maleT229    4.275715    0.301350  14.189  <2e-16 ***
## maleT231    4.301052    0.302327  14.227  <2e-16 ***
## maleT234    3.557692    0.305281  11.654  <2e-16 ***
## maleT235    4.907960    0.316681  15.498  <2e-16 ***
## maleT236    4.432163    0.300701  14.739  <2e-16 ***
## maleT237    4.262585    0.309520  13.772  <2e-16 ***
## maleT243    4.355303    0.289683  15.035  <2e-16 ***
## maleT244    4.002712    0.324817  12.323  <2e-16 ***
## maleT247    4.216963    0.305795  13.790  <2e-16 ***
## maleT257    4.992306    0.291354  17.135  <2e-16 ***
## maleT258    2.935247    0.330662   8.877  <2e-16 ***
## maleT260    3.979770    0.311631  12.771  <2e-16 ***
## x1          -0.127369    0.062492  -2.038   0.0415 *
## I(x1^2)     -0.005606    0.003320  -1.689   0.0913 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_nb), correlation=TRUE) or
##      vcov(summary(glmer_nb))      if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
## unused control arguments ignored

```

```

##
## [1] "temp_ref = 35 glmer_nb re = quad"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0064789 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00834887 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?;Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## unused control arguments ignored

## boundary (singular) fit: see help('isSingular')

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(4.9599) ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + I(x1^2) | male)
## Data: data
## Control:
## glmerControl(boundary.tol = 0, tolPwrss = 0.1, optCtrl = list(maxiter = 1e+05,
## maxfun = 2e+06, trace = trace), optimizer = optimizer)
##
##      AIC      BIC    logLik deviance df.resid
##    680.2    714.0   -324.1    648.2      45
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.9724 -0.6915 -0.1994  0.5434  3.1766
##
## Random effects:
##  Groups Name      Variance Std.Dev.

```

```

## male I(x1^2) 0 0
## Number of obs: 61, groups: male, 12
##
## Fixed effects:
## Estimate Std. Error z value Pr(>|z|)
## maleT229 4.216007 0.297743 14.160 <2e-16 ***
## maleT231 4.242947 0.299006 14.190 <2e-16 ***
## maleT234 3.499089 0.301998 11.586 <2e-16 ***
## maleT235 4.844355 0.313310 15.462 <2e-16 ***
## maleT236 4.382738 0.297332 14.740 <2e-16 ***
## maleT237 4.209983 0.306672 13.728 <2e-16 ***
## maleT243 4.289398 0.285830 15.007 <2e-16 ***
## maleT244 3.941226 0.321803 12.247 <2e-16 ***
## maleT247 4.144315 0.301920 13.727 <2e-16 ***
## maleT257 4.914289 0.286369 17.161 <2e-16 ***
## maleT258 2.869099 0.327170 8.769 <2e-16 ***
## maleT260 3.910894 0.308047 12.696 <2e-16 ***
## x1 -0.143609 0.061788 -2.324 0.0201 *
## I(x1^2) -0.006424 0.003283 -1.957 0.0504 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_nb), correlation=TRUE) or
## vcov(summary(glmer_nb)) if you need it

## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
## unused control arguments ignored
##
## [1] "temp_ref = 35 glmer_nb re = both"

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0262617 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in (function (npt = min(n + 2L, 2L * n), rhobeg = NA, rhoend = NA, :
## unused control arguments ignored

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## unused control arguments ignored

```

```

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, :
## Model failed to converge with max|grad| = 1.11108 (tol = 0.002, component 1)

## Warning in optTheta(g1, interval = interval, tol = tol, verbose = verbose, : Model is nearly unident.
## - Rescale variables?;Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(25.2301) ( log )
## Formula: song_count ~ -1 + male + x1 + I(x1^2) + (-1 + x1 + I(x1^2) |
## male)
## Data: data
## Control:
## glmerControl(boundary.tol = 0, tolPwrss = 0.1, optCtrl = list(maxiter = 1e+05,
## maxfun = 2e+06, trace = trace), optimizer = optimizer)
##
## AIC BIC logLik deviance df.resid
## 743.7 781.7 -353.8 707.7 43
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -3.0202 -0.8896 -0.2093 0.5991 4.6854
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## male x1 0.0806149 0.28393
## I(x1^2) 0.0002113 0.01454 0.98
## Number of obs: 61, groups: male, 12
##
## Fixed effects:
## Estimate Std. Error z value Pr(>|z|)
## maleT229 4.080355 0.324722 12.566 < 2e-16 ***
## maleT231 4.531626 0.384781 11.777 < 2e-16 ***
## maleT234 3.264457 0.388690 8.399 < 2e-16 ***
## maleT235 4.761868 0.462525 10.295 < 2e-16 ***
## maleT236 2.989138 0.500802 5.969 2.39e-09 ***
## maleT237 4.288885 0.401619 10.679 < 2e-16 ***
## maleT243 5.012198 0.373985 13.402 < 2e-16 ***
## maleT244 3.725739 0.598097 6.229 4.68e-10 ***
## maleT247 3.747123 0.432882 8.656 < 2e-16 ***
## maleT257 5.722484 0.356312 16.060 < 2e-16 ***
## maleT258 0.671660 0.589491 1.139 0.255
## maleT260 6.428693 0.719856 8.931 < 2e-16 ***
## x1 -0.143488 0.090652 -1.583 0.113
## I(x1^2) -0.006533 0.004714 -1.386 0.166
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(summary(glmer_nb), correlation=TRUE) or
## vcov(summary(glmer_nb)) if you need it

```

```
## optimizer (bobyqa) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 1.11108 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?
## unused control arguments ignored
```

Results

- Changing `temp_ref` alters the linear term, as expected, but has little impact on the overall model fit.
- fixed effects qpoisson and nb indicates data is highly over dispersed
- poisson and nb RE models do not converge
 - With RE for both `x1` and `x1^2` we get failure to converge and the correlation between these two RE is 1 (this means there's really, at best, one RE|male rather than two).
 - With single RE we get return `boundary(singular)` error because the var of teh RE is essentially zero. This indicates the model is too complex given the data

Basically, I don't think this avenue is fruitful, but I do need to understand how the covariance matrix is used in these models - What is the difference between the | and || models? - I believe the | is a correlated random effects approach - Houseman test is used for comparing FE and RE - Does one of these assume the random effects themselves vary between trials for the same male as some references suggest, i.e. the RE for `x1` for male `z` varies between replicates? Trials? - Seems like we could compare a `lmList` result with the `glmer`

Additional Fixed Effects fits

Fit Fixed Effects models

```
est_cov <- FALSE
verbose <- 0
trace <- FALSE

#for(temp_ref in vec_temp_ref){
temp_ref <- 35

  data <- data_ind %>%
    mutate(x1 = (temp_mean - temp_ref))
  print(paste("temp_ref:", temp_ref))

  re <- "none"

  glm_poisson <-
    glm(song_count ~ 1 + male + x1/male + I(x1^2)/male,
        data = data,
        family = poisson(link = "log")
    )

  summary(glm_poisson)
  plot(glm_poisson, ask = FALSE)
```

```

glm_qpoisson <- update(glm_poisson,
                      family = quasipoisson(link = "log")
                      )
summary(glm_qpoisson)

## Note the dispersion parameter is 29.7 rather than 1!!
## Clearly the data is over dispersed
## quasipoisson() doesn't seem to exist for glmer (but likely exists in nlme)
plot(glm_qpoisson, ask = FALSE)

glm_nb <- glm.nb <-
  glm(song_count ~ 1 + male + (x1/male) + I(x1^2)/male,
      data = data
      )
summary(glm_nb)
plot(glm_nb, ask = FALSE)

## Dispersion parameter is 4547!
}

## Error: <text>:43:1: unexpected '}'
## 42:      ## Dispersion parameter is 4547!
## 43: }
##      ^

```

Results

- `poisson` model does give significant effects, but `qpoisson` indicates data is overdispersed. When overdispersion is taken into account, significance disappears.
 - This occurs even if you're only using `/male` on one of the `x1` terms.

End

```
knitr::knit_exit()
```