HW2 Yuanrong Liu

2024-10-11

Packages Loaded

```
library(ggplot2)
library(performance)
library(DHARMa)
library(dotwhisker)
library(ggeffects)
library(sjPlot)
library(bbmle)
library(brglm2)
library(arm)
library(logistf)
library(pscl)
library(MASS)
```

Question 1

```
# import the data without loading the package
data("Contraception", package = "mlmRev")

# Convert 'use' factor variable to binary numeric (0/1)
# `as.numeric(use)` would convert "N" to `1` and "Y" to `2`
Contraception$use_num = as.numeric(Contraception$use) - 1

# Ensure urban is a factor with levels "N" and "Y"
Contraception$urban <- factor(Contraception$urban, levels = c("N", "Y"))
```

(a) Analysis Strategy

Family / Link Function

- Since the response variable, `use_num`, converted from `use`, is a binary outcome, we are going to use a **binomial family** and a **logit link function**.

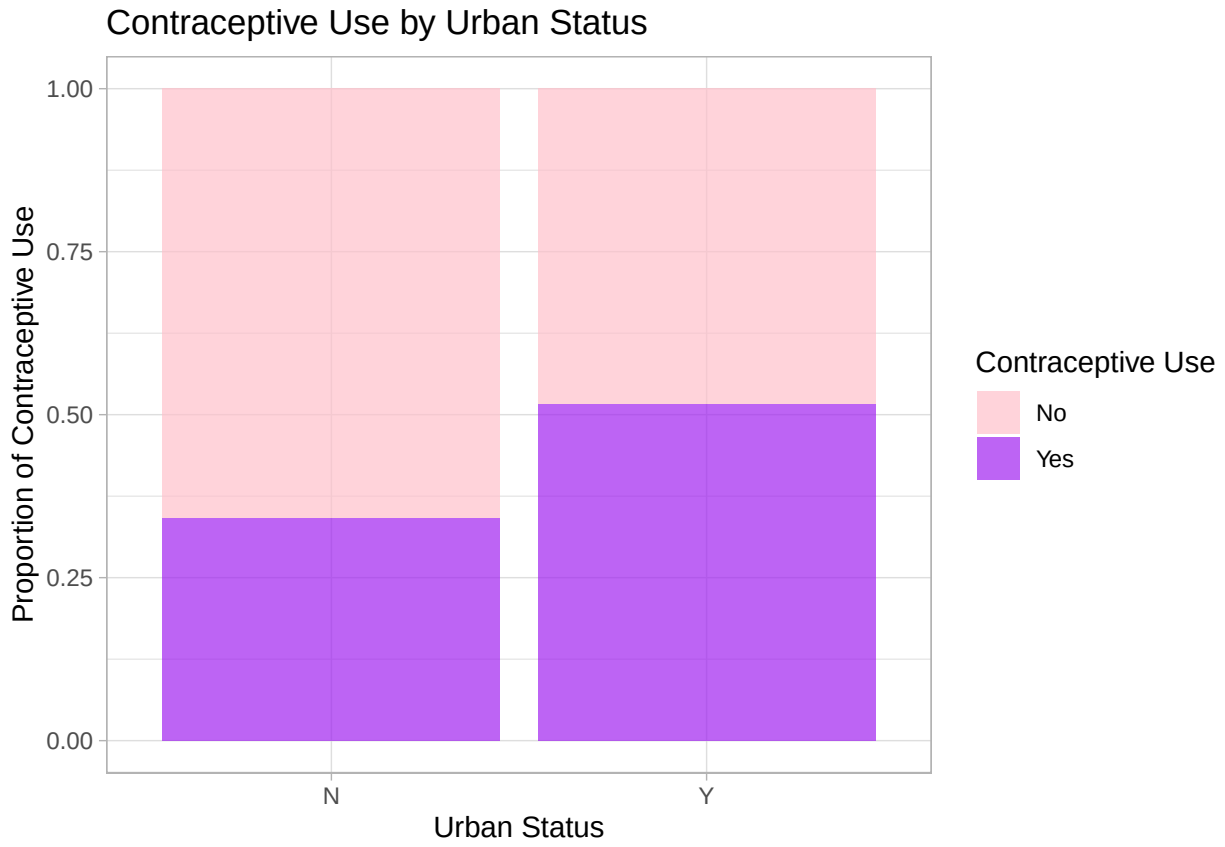
Predictors to include

- **urban**: a categorical predictor, indicating whether the woman resides in an urban or rural area.
- **age**: a continuous predictor, centered around the mean, included as a linear predictor.
- **livch**: an ordinal predictor with four values (0, 1, 2, 3+), to assess how the number of living children impacts contraceptive use.

(b) Plot the Data

Plot Contraceptive Use by Urban Status

```
ggplot(Contraception, aes(x = urban, fill = factor(use_num))) +
  geom_bar(position = "fill", alpha = 0.7) +
  scale_fill_manual(values = c("pink", "purple"), labels = c("No", "Yes")) +
  labs(x = "Urban Status", y = "Proportion of Contraceptive Use",
       fill = "Contraceptive Use",
       title = "Contraceptive Use by Urban Status") +
  theme_light()
```



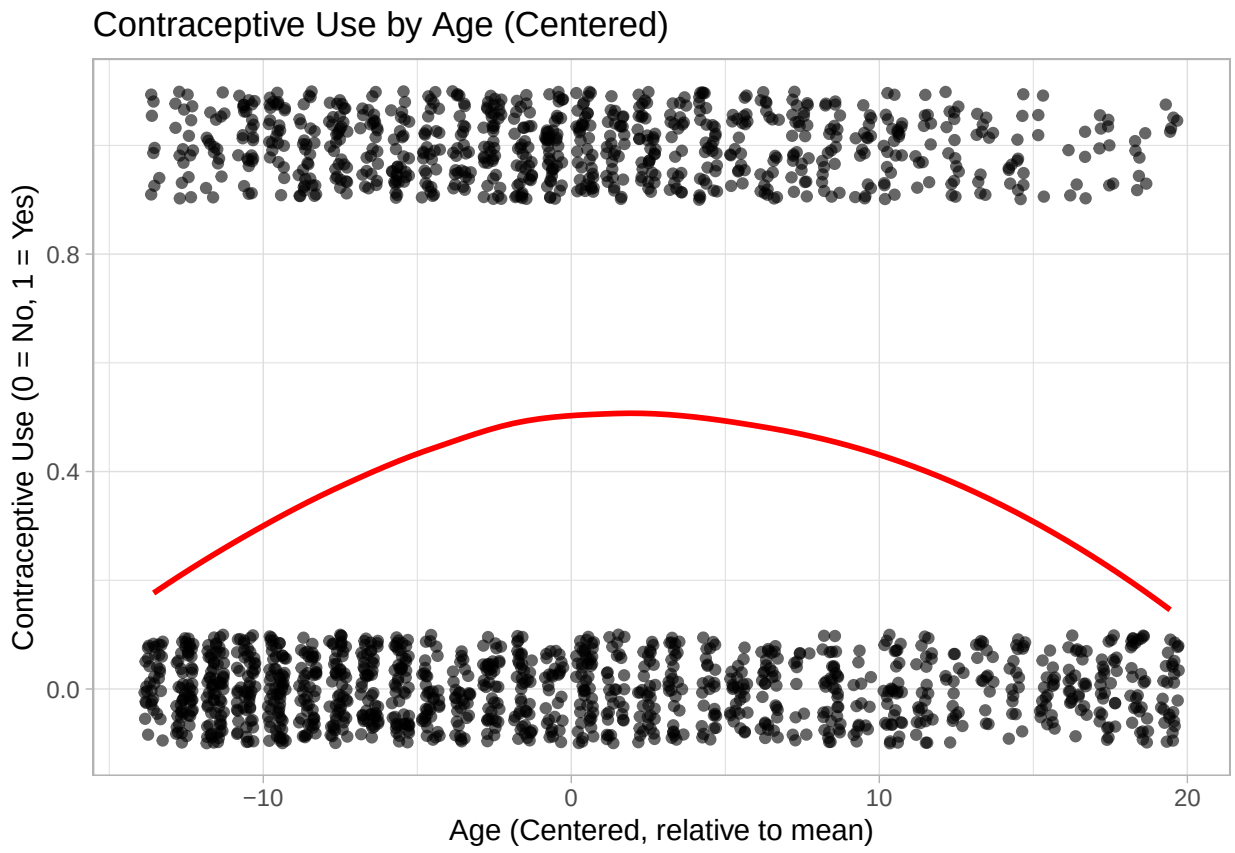
- The **purple** bars indicate the proportion of women using contraceptives, while the **pink** bars indicate the proportion of women not using contraceptives.
- In **urban** areas, a slightly greater proportion of women use contraceptives compared to those who do not, indicated by the larger purple section compared to the pink section. In **rural** areas, the majority of women do not use contraceptives, as evidenced by the larger pink section compared to the purple section.
- Contraceptive use is **more common in urban areas** than in rural areas.

Plot Contraceptive Use by Age

```
ggplot(Contraception, aes(x = age, y = use_num)) +
  geom_point(position = position_jitter(width = 0.3, height = 0.1), alpha = 0.6) + # plot each point
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 1) + # plot the general trend
```

```
  labs(x = "Age (Centered, relative to mean)", y = "Contraceptive Use (0 = No, 1 = Yes)",
       title = "Contraceptive Use by Age (Centered)") +
  theme_light()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

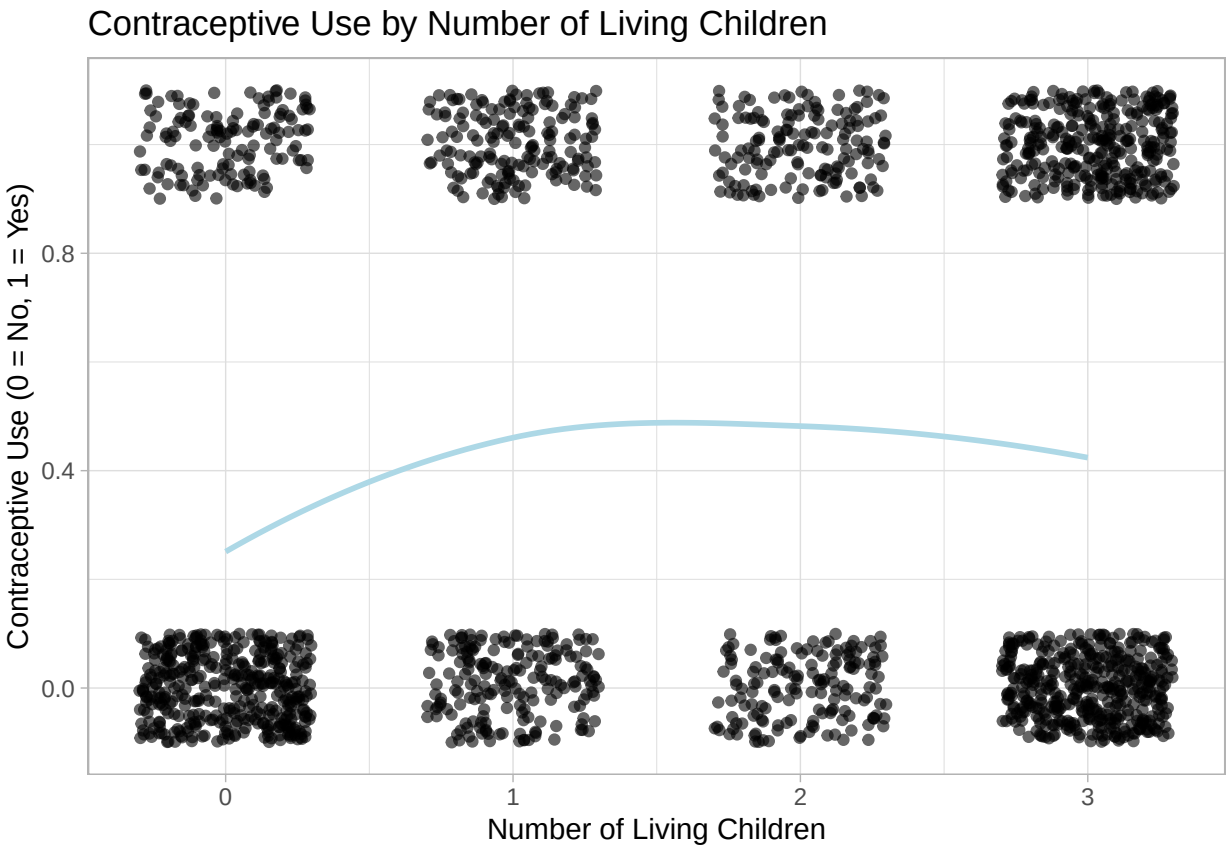


- The **black points** represent individual data points (use of contraceptives at different ages), with jitter added for clarity.
- The **red line** represents the general trend of contraceptive use by age, using LOESS smoothing.
- From the LOESS curve, we can see that the contraceptive use increases with age, reaching a peak when the age is around 2 years older than average and then decreasing with age. It might correspond to a particular life stage where contraception is more common.

Plot Contraceptive Use by Number of Living Children (livch)

```
ggplot(Contraception, aes(x = as.numeric(livch) - 1, y = use_num)) +
  geom_point(position = position_jitter(width = 0.3, height = 0.1), alpha = 0.6) +
  geom_smooth(method = "loess", se = FALSE, color = "lightblue", linewidth = 1) +
  labs(x = "Number of Living Children", y = "Contraceptive Use (0 = No, 1 = Yes)",
       title = "Contraceptive Use by Number of Living Children") +
  theme_light()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



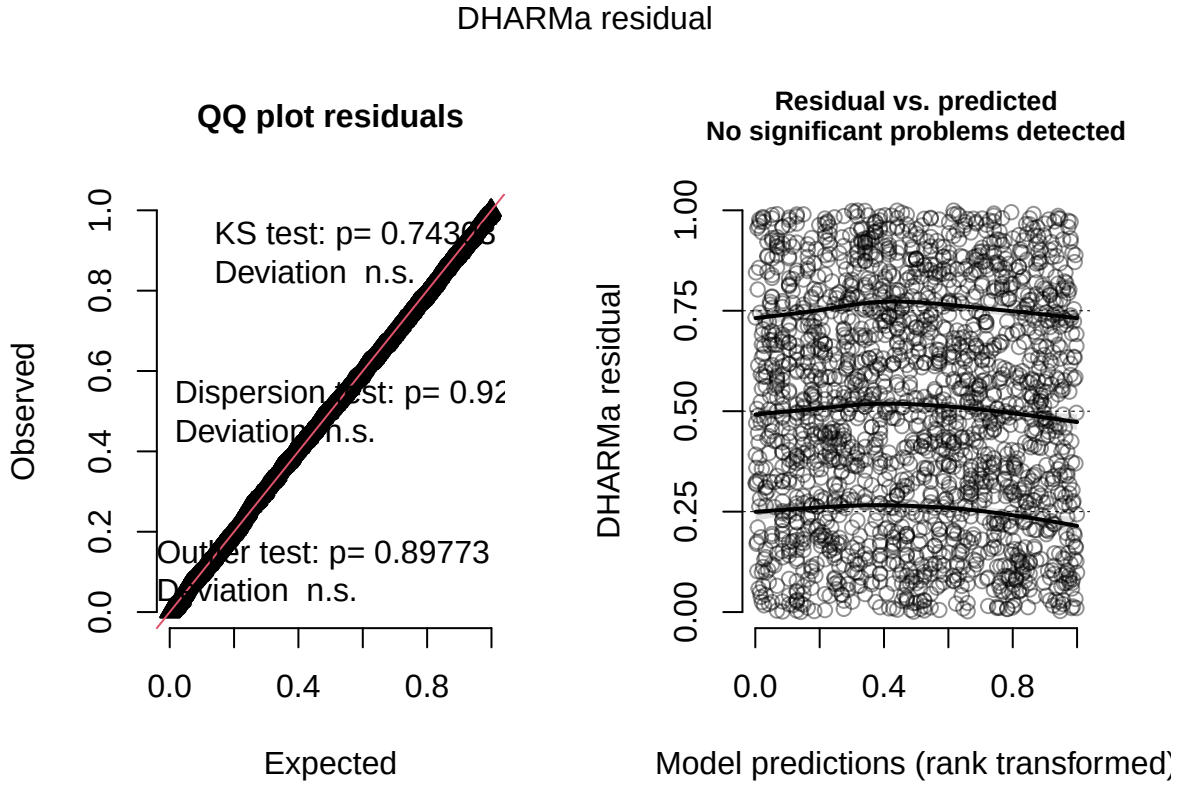
- **Black points** indicate individual data points representing whether a woman is using contraceptives given the number of living children.
- The **blue line** represents a LOESS (locally weighted scatterplot smoothing) trend showing the general relationship between the number of living children and contraceptive use.
- Contraceptive use appears to **increase** initially as the number of living children increases **from 0 to 1**.
- It seems to **peak around 1 or 2** children, suggesting that women are more likely to use contraceptives when they have one or two children. Beyond two children, the trend slightly decreases.

(c) Fit the Model

```
model <- glm(formula =  use_num ~ urban + age + livch,
             family = binomial(link = "logit"),
             data = Contraception)
```

(d) Diagnostic Plots

```
plot(simulateResiduals(fittedModel = model))
```

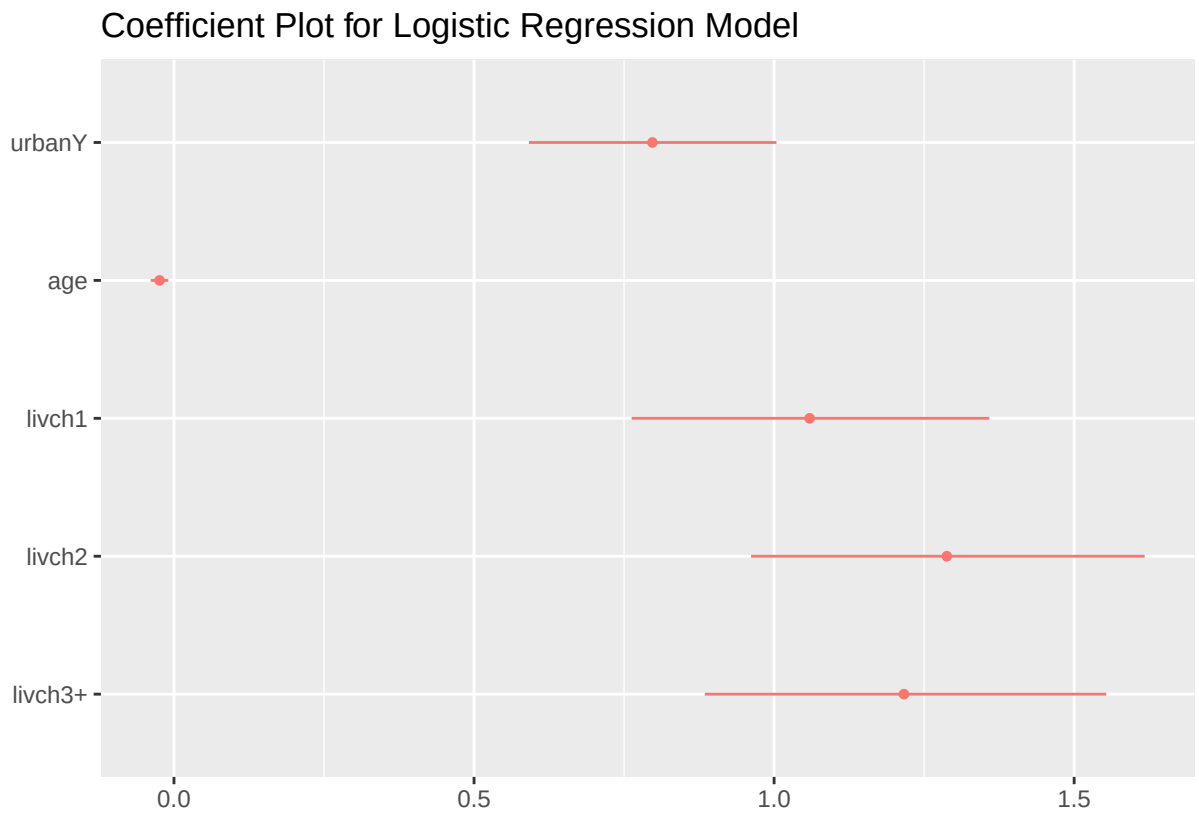


- I decided to use **DHARMa residual** diagnostics because they provide easily interpretable residual distributions and identify model misspecifications visually. The DHARMa residuals help assess uniformity and homoscedasticity, which are key to determining the validity of the logistic regression model.
- **QQ Plot Residuals**: the residuals from the model (represented as the red line) lie mostly along the ideal line (the black line), suggesting that the residuals are approximately **uniformly distributed**, which indicates a good model fit. The p-values are `0.7430` for KS test, `0.928` for dispersion test, and `0.8977` for outlier test respectively, showing that the model residuals do not significantly deviate from expectations.
- **Residual vs. Predicted**: the residuals appear fairly **uniformly distributed** across the predicted values (represented by the horizontal lines), with no clear trends or systematic deviations, which suggests no significant problems with the model.
- Both plots and the statistical tests suggest that the model fits the data well.

(e) Interpretation with Coefficient and Effect plots

Coefficient plot using `dotwhisker`

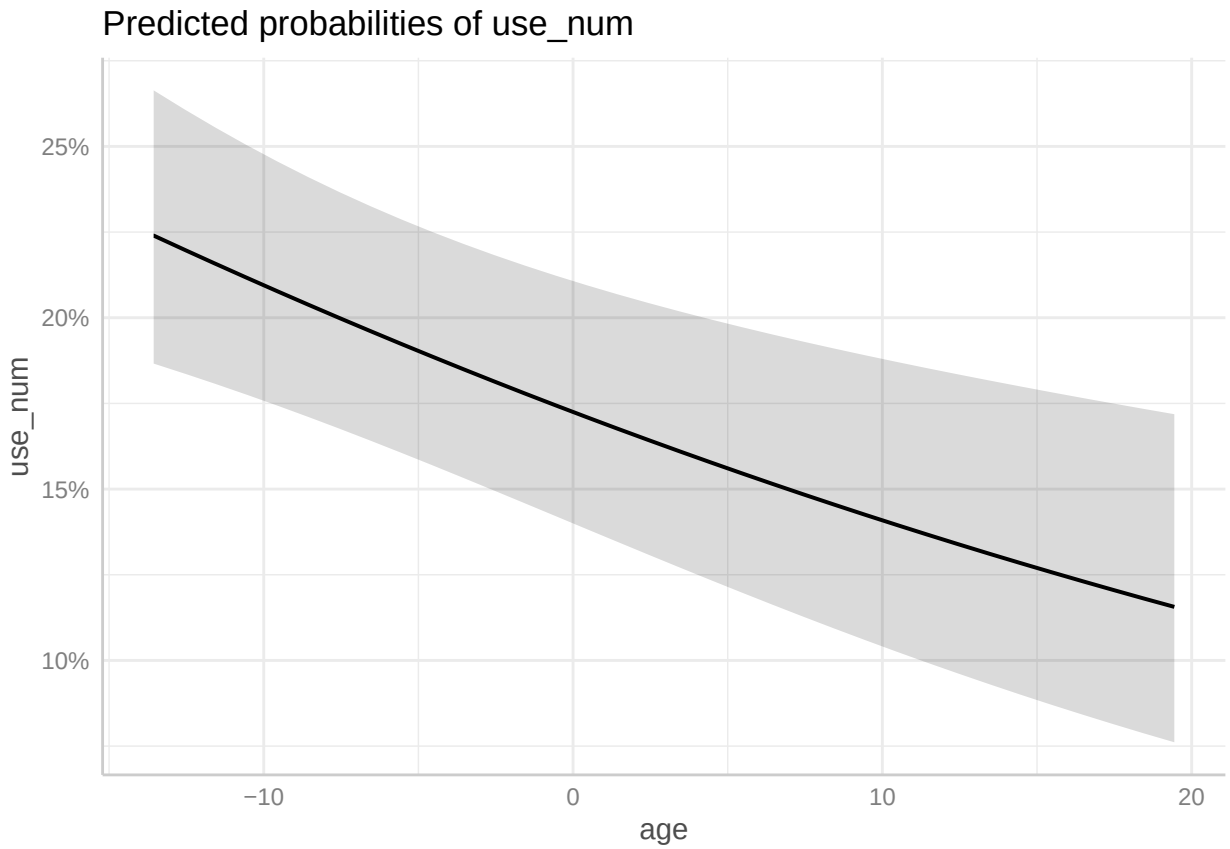
```
# Plotting model coefficients with their confidence intervals
dwplot(model) + ggtitle("Coefficient Plot for Logistic Regression Model")
```



- The **dots** represent the estimated coefficients for each predictor: `urbanY` (difference between urban and rural areas), `age` (centred around mean), `livch1`, `livch2`, `livch3+` (where the reference category is `livch0`).
- The horizontal lines around each point represent the 95% **confidence intervals** (CI) for these estimates.
- For `urbanY`, the CI is far from 0, suggesting that women from urban areas have a significantly larger likelihood of using contraceptives, compared to those from rural areas.
- For `age`, the estimate is negative, and the CI is very close to 0, indicating that age may not have a strong impact on contraceptive use, for this model.
- For `livch1`, `livch2`, `livch3+`, the CIs are all far away from 0, indicating that women with living children tend to have higher likelihood of using contraceptives, compared to those without living children.

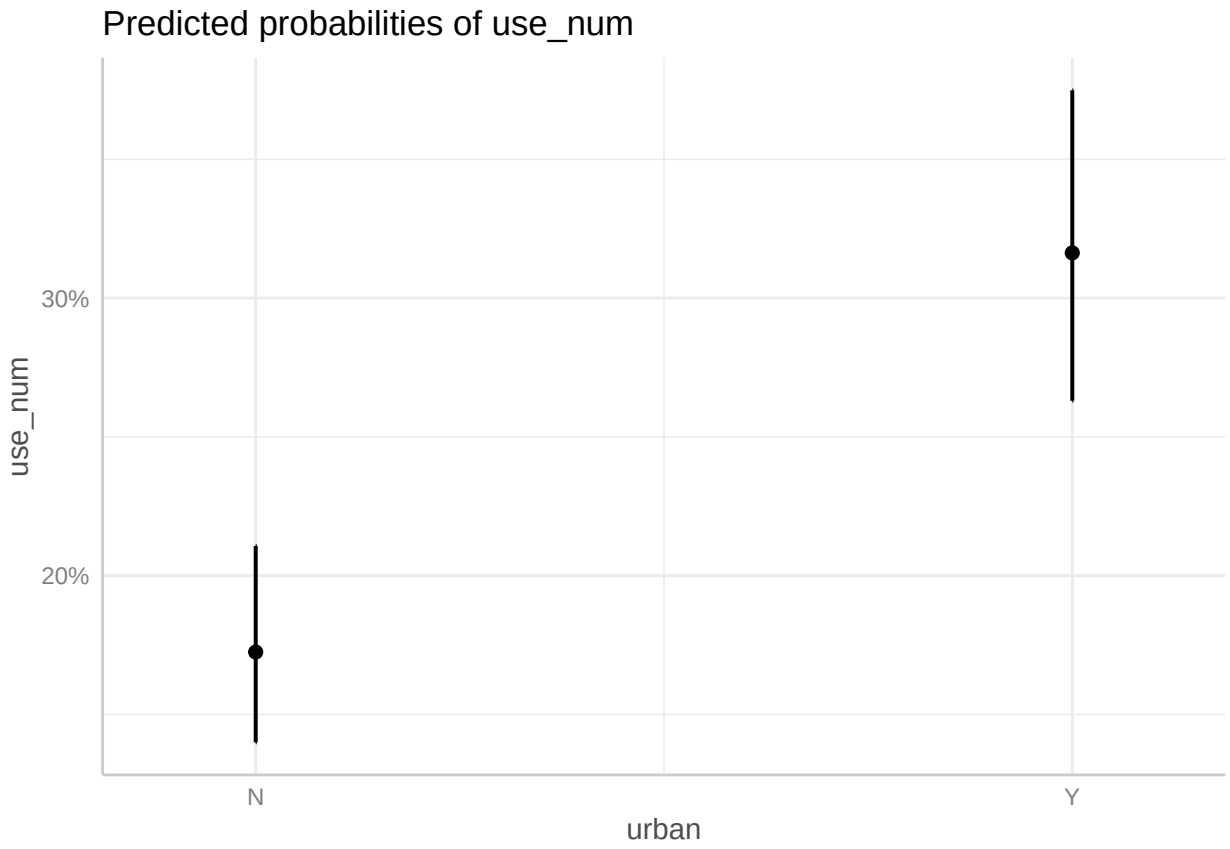
Effect plot using `ggeffects`

```
# Plotting the predicted effect of age on contraceptive use
effect_plot_age <- ggpredict(model, terms = c("age [all]"))
plot(effect_plot_age)
```



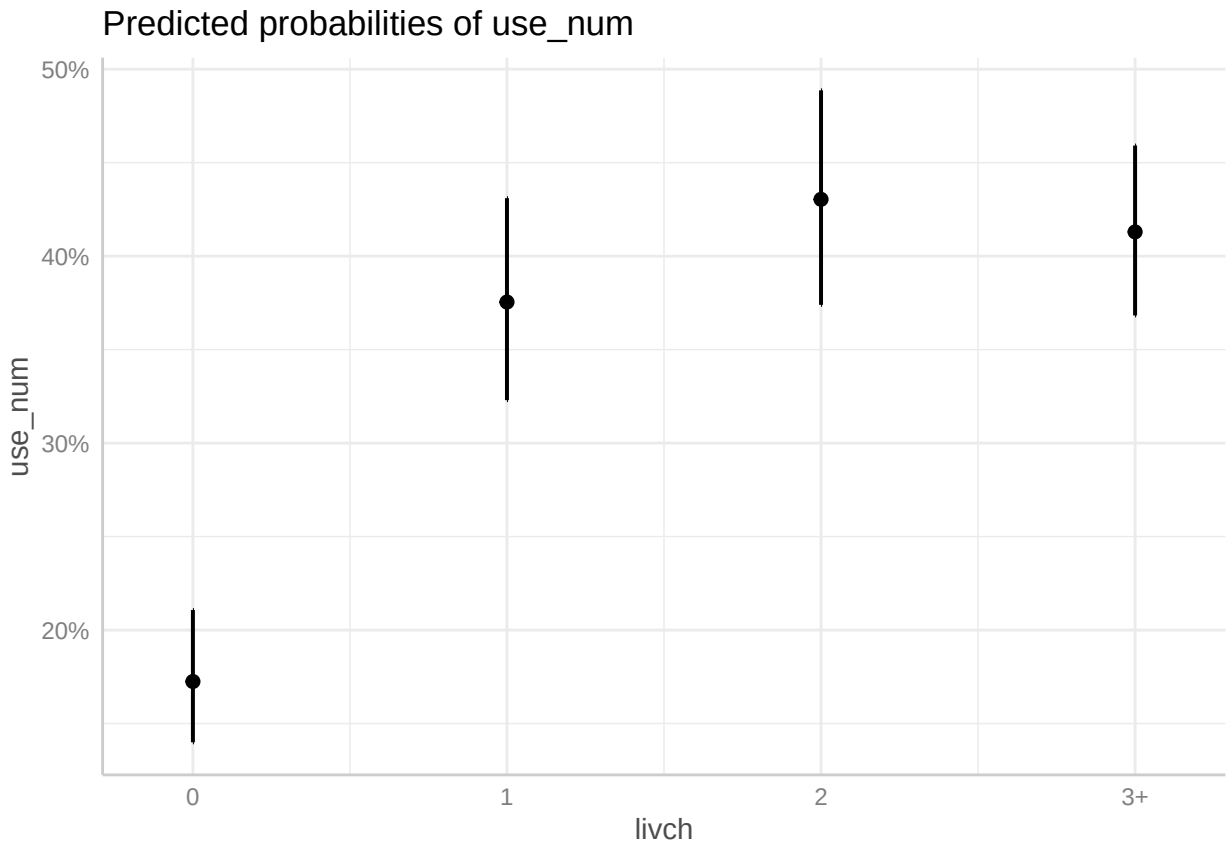
- The plot shows a downward trend in contraceptive use probability as age increases. Younger women are more likely to use contraceptives than older women.
- The confidence interval (represented as the gray shaded area) widens as age moves further from mean, indicating increased uncertainty about the predicted probabilities for older women.

```r
# Plotting the predicted effect of urban status on contraceptive use
effect_plot_urban <- ggpredict(model, terms = c("urban"))
plot(effect_plot_urban)
```



- The predicted probability of contraceptive use is higher for women living in **urban** areas (`Y`, around `32%`) than those in **rural** areas (`N`, about `17%`).
- The confidence intervals for urban and rural do not overlap, suggesting that the difference in contraceptive use between urban and rural women is statistically significant.
- The confidence interval for urban is wider than that for rural, showing **more uncertainty** of the predicted probabilities of contraceptive use for women in urban areas.

```
# Plotting the predicted effect of number of living children (livch) on contraceptive use
effect_plot_livch <- ggpredict(model, terms = c("livch"))
plot(effect_plot_livch)
```



- The probability of contraceptive use is **lowest** for women with `0` living children (around `17%`).
- The probability of contraceptive use **increases** significantly for women with `1`, `2`, or `3+` children, reaching around `40%`, as there is **no overlap** between their confidence intervals and the confidence interval for `livch0`, suggesting that women are more likely to use contraceptives as they have more children.
- The **confidence interval** become wider as the number of living children increases, particularly for `livch3+`, indicating greater uncertainty about the predicted probability of contraceptive use.

Question 2

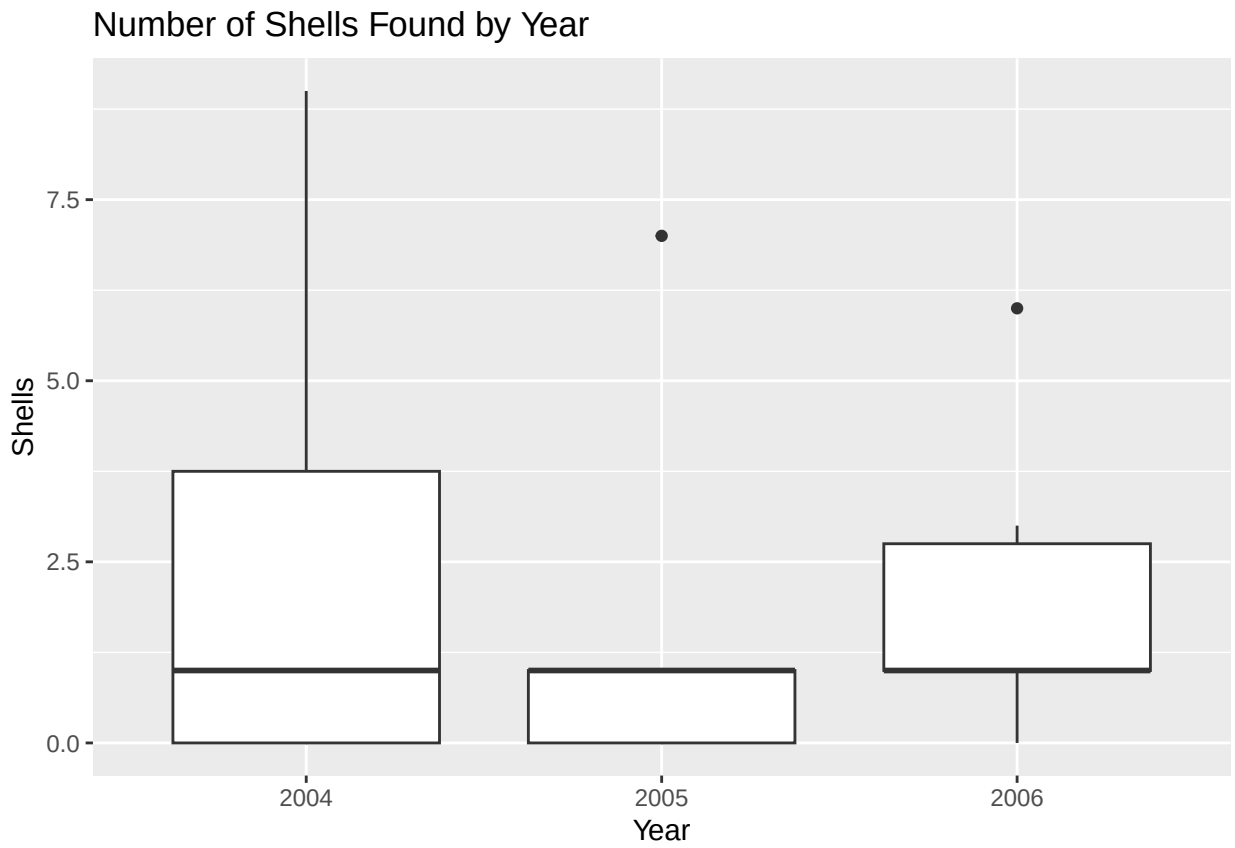
```r
# Load the data
g_url <- "https://raw.githubusercontent.com/bbolker/mm_workshops/master/data/gopherdat2.csv"
g_data <- read.csv(g_url)

# Center the year variable to make the intercept more interpretable
g_data$year_centered <- g_data$year - mean(g_data$year)
```

(a) Plot the Data

```r
# Plot shells vs year and prevalence
ggplot(g_data, aes(x = factor(year), y = shells)) +
  geom_boxplot() +
```

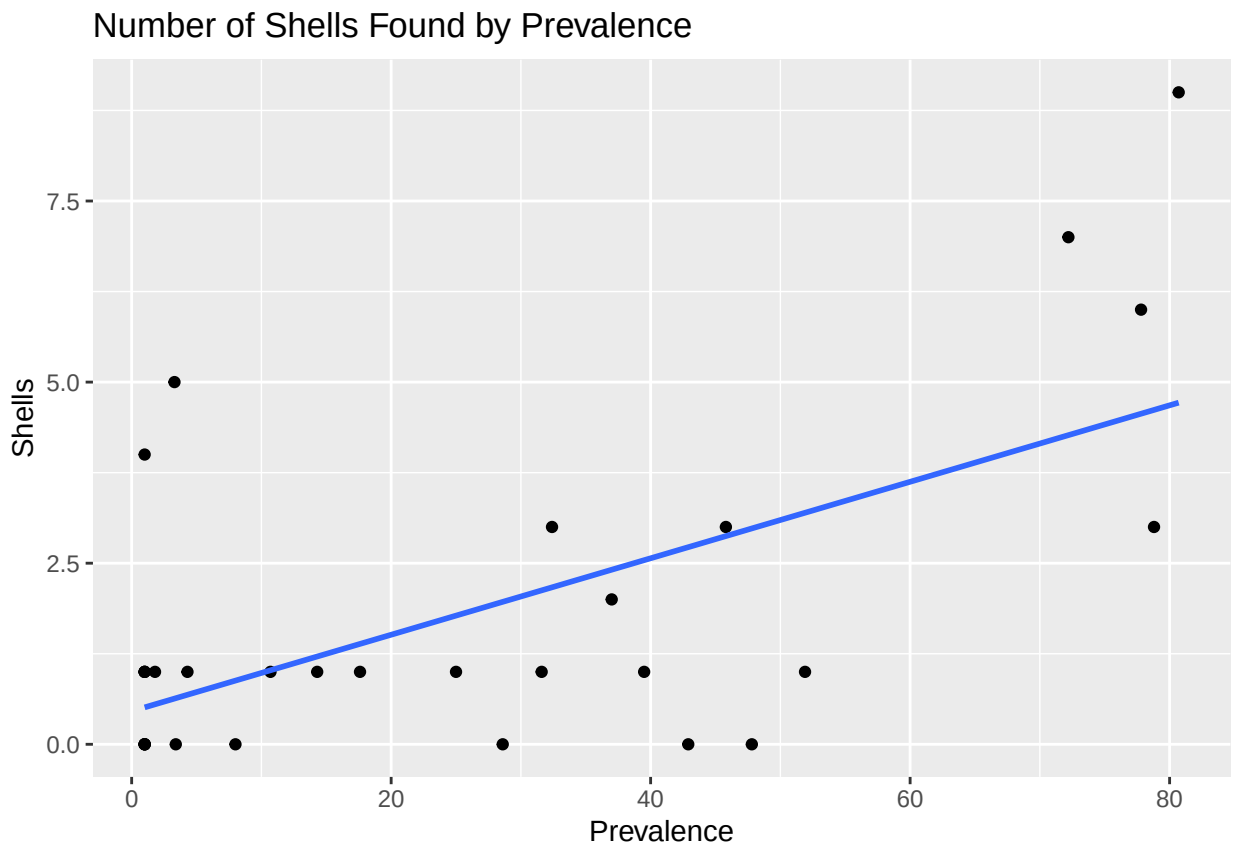
```
  labs(title = "Number of Shells Found by Year",
       x = "Year", y = "Shells")
```



- The **boxplot** shows that there is large variability with shell counts ranging from 0 to 9, in 2004. 2005 and 2006 show less variability and consistently low counts, with all values at or close to 0.

```
# Plot shells vs prevalence
ggplot(g_data, aes(x = prev, y = shells)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Number of Shells Found by Prevalence",
       x = "Prevalence", y = "Shells")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



- The **scatter plot** shows an increasing trend between prevalence and the number of shells found.

(b) Fit a GLM

- The response (`shells`) is count data, so we use a `Poisson` GLM. Since the number of shells found depends on the area, we include `log(Area)` as an offset.

```
fit_glm <- glm(shells ~ year_centered + prev + offset(log(Area)),
               data = g_data,
               family = poisson)
```

```
# Check for Overdispersion
deviance_ratio <- fit_glm$deviance / fit_glm$df.residual
deviance_ratio
```

```
## [1] 0.9006464
```

- The value of deviance_ratio is `0.9006464`, which is **close to 1** and slightly less than 1, suggesting that the variance is roughly equal to the mean, which aligns with the assumptions of a **Poisson distribution**, where overdispersion is not an issue for this model.

(c) Fit the Same Model Using bbmle

```
fit_mle2 <- mle2(shells ~ dpois(lambda = exp(a + b * year_centered + c * prev
                                             + log(Area))),
                 start = list(a = 0, b = 0, c = 0),
                 data = g_data)
summary(fit_mle2)
```

```
## Maximum likelihood estimation
##
## Call:
## mle2(minuslogl = shells ~ dpois(lambda = exp(a + b * year_centered +
##     c * prev + log(Area))), start = list(a = 0, b = 0, c = 0),
##     data = g_data)
##
## Coefficients:
##     Estimate Std. Error  z value     Pr(z)
## a -3.5563450  0.2412167 -14.7434 < 2.2e-16 ***
## b -0.2283314  0.1649610  -1.3842    0.1663
## c  0.0218136  0.0043262   5.0422 4.601e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: 78.17166
```

- `a` (**Intercept**): The estimate for the intercept is `-3.5563`. This value represents the expected log count of shells at the average year (2005) and average values of the other predictors (after adjusting for Area). The **p-value is very small** (`< 2.2e-16`), indicating that this estimate is **statistically significant**.
- `b` (**Year Centered**): The estimate for b is `-0.2283`, indicating a negative effect of year on the shell count. The **p-value (0.1663)**, means that the effect of year is **not statistically significant** (at the 5% significance level). There is no strong evidence that the number of shells changes significantly over time within the years analyzed.
- `c` (**Prevalence**): The estimate for c is `0.0218`, showing a **positive relationship** between prevalence and shell counts. The **p-value (4.601e-07)** indicates that this effect is highly statistically significant.

(d) Write Negative Log-Likelihood

```
neg_log_likelihood <- function(a, b, c) {
  lambda <- exp(a + b * g_data$year_centered + c * g_data$prev + log(g_data$Area))
  -sum(dpois(g_data$shells, lambda = lambda, log = TRUE)) # the Negative Log-Likelihood
}

# Using optim to Minimize the Negative Log-Likelihood
optim_fit <- optim(par = c(a = 0, b = 0, c = 0),
                   fn = function(params)
                     neg_log_likelihood(params[1], params[2], params[3]))
```

(e) Compare the Parameters from Different Models

```
compare_parameters <- function() {
  # Extract parameter estimates from GLM
  glm_params <- coef(fit_glm)

  # Extract parameter estimates from MLE2
  mle2_params <- coef(fit_mle2)

  # Extract parameter estimates from Optim
  optim_params <- optim_fit$par

  # Create a data frame to compare parameters
  comparison <- data.frame(
    GLM = glm_params,
    MLE2 = mle2_params,
    Optim = optim_params
  )

  return(comparison)
}

# Call the function to compare parameters
compare_parameters()
```

```
##                       GLM        MLE2       Optim
## (Intercept)   -3.55710147 -3.55634500 -3.55661473
## year_centered -0.22909974 -0.22833145 -0.22926177
## prev           0.02182806  0.02181357  0.02182335
```

- From the dataframe above, we can see that the parameters are nearly identical, with the values around `-3.557` for `intercept`, `-0.229` for `year_centered`, and `0.0218` for `prev`.

(f) Compare Confidence Intervals

```
# Wald Confidence Intervals
confint.default(fit_glm)
```

```
##                     2.5 %      97.5 %
## (Intercept)   -4.02999442 -3.08420851
## year_centered -0.55242639  0.09422691
## prev           0.01334873  0.03030739
```

```
# Profile Confidence Intervals
confint(fit_glm)
```

```
## Waiting for profiling to be done...
```

```
##                     2.5 %      97.5 %
## (Intercept)   -4.06178997 -3.11252874
## year_centered -0.55540483  0.09425315
## prev           0.01337972  0.03041908
```

```
confint(fit_mle2)
```

```
##         2.5 %      97.5 %
## a -4.06181471 -3.11256088
## b -0.55539566  0.09427119
## c  0.01337938  0.03041905
```

- The **Wald confidence interval** is calculated for the **GLM model** (`fit_glm`), while the **Profile confidence interval** is calculated for both the **GLM model** and the **MLE model** (`fit_mle2`).
- The **Wald Confidence Intervals** and **Profile Confidence Intervals** for the model parameters are **nearly identical**, with the value of around `[-4.05, -3.1]` for `Intercept`, `[-0.55, 0.094]` for `year_centered`, and `[0.0133, 0.0303]` for `prev`.
- Since the intervals are similar, it suggests that the model fits the data well, and there is no significant skewness or non-linearity in the parameter estimates.
- The function `confint.default()` expects an **S3** object, which is not working for `mle2` from the `bbmle` package produces an **S4** class object.

Question 3

```
# Load the Endometrial data
data("endometrial", package = "brglm2")
data_e <- endometrial  # Assign the data to the variable data_e
```

Introduction

- The goal of your analysis is to predict the **response variable** `HG` (a binary varible, `0` for **benign** and `1` for **malignancy**) using the **predictor variables** (`NV`, `PI`, `EH`).

Fit the model using three methods

```
# 1. Regular Generalized Linear Model (GLM)
model_glm <- glm(HG ~ NV + PI + EH, family = binomial, data = data_e)

# 2. Bayesian GLM with regularizing priors
model_bayesglm <- bayesglm(HG ~ NV + PI + EH, family = binomial, data = data_e)

# 3. Logistic Firth Regression
model_logistf <- logistf(HG ~ NV + PI + EH, data = data_e)
```

Compare models

1. Coefficient Estimates

```r
coef_glm <- coef(model_glm)
coef_bayesglm <- coef(model_bayesglm)
coef_logistf <- coef(model_logistf)

df_coef_comparison <- data.frame(
  Model = c("GLM", "Bayesian GLM", "Logistic Firth Regression"),
  Intercept = c(coef_glm["(Intercept)"], coef_bayesglm["(Intercept)"], coef_logistf["(Intercept)"]),
  NV = c(coef_glm["NV"], coef_bayesglm["NV"], coef_logistf["NV"]),
  PI = c(coef_glm["PI"], coef_bayesglm["PI"], coef_logistf["PI"]),
  EH = c(coef_glm["EH"], coef_bayesglm["EH"], coef_logistf["EH"])
)
print(df_coef_comparison)
```

```
##                        Model Intercept        NV          PI        EH
## 1                        GLM  4.304518 18.185556 -0.04218340 -2.902606
## 2               Bayesian GLM  3.711594  3.298156 -0.02903616 -2.628647
## 3 Logistic Firth Regression  3.774560  2.929273 -0.03475175 -2.604164
```

- **Intercept**: The intercept represents the log odds of `HG = 1` when all predictor variables (`NV`, `PI`, `EH`) are zero. The intercepts for the three models are similar, indicating a **similar baseline** log odds for malignancy when other predictors are not present.
- **NV** (Nuclear Volume): The **GLM** model has a significantly higher coefficient (`18.1856`) compared to **Bayesian GLM** (`3.2982`) and **Logistic Firth Regression** (`2.9293`), showing that the GLM model suggests a much stronger positive association between `NV` and `HG`.
- **PI** (Polymorphism Index): All three models indicate a **negative** relationship between `PI` and `HG`, suggesting that higher values of PI are associated with a lower probability of malignancy.
- **EH** (Endometrial Height): The coefficient for `EH` is **negative** in all models, indicating an **inverse** relationship with `HG`.

2. Confidence Intervals

```r
# 1. GLM - Using Wald Confidence Intervals
confint_wald_glm <- confint.default(model_glm)

# 2. Bayesian GLM - Using Wald Confidence Intervals
confint_wald_bayesglm <- confint.default(model_bayesglm)

# 3. Logistic Firth Regression - Wald and Profile Confidence Intervals
confint_wald_logistf <- confint.default(model_logistf)
confint_profile_logistf <- confint(model_logistf)

df_confint_summary <- data.frame(
  Model = c("GLM (Wald)", "Bayesian GLM (Wald)",
            "Logistic Firth Regression (Wald)", "Logistic Firth Regression (Profile)"),
  Intercept = c(paste0("[", round(confint_wald_glm["(Intercept)", 1], 3), ", ",
                       round(confint_wald_glm["(Intercept)", 2], 3), "]"),
                paste0("[", round(confint_wald_bayesglm["(Intercept)", 1], 3), ", ",
                       round(confint_wald_bayesglm["(Intercept)", 2], 3), "]"),
                paste0("[", round(confint_wald_logistf["(Intercept)", 1], 3), ", ",
                       round(confint_wald_logistf["(Intercept)", 2], 3), "]"),
```

```
                paste0("[", round(confint_profile_logistf["(Intercept)", 1], 3), ", ",
                       round(confint_profile_logistf["(Intercept)", 2], 3), "]")),
  NV = c(paste0("[", round(confint_wald_glm["NV", 1], 3), ", ",
                round(confint_wald_glm["NV", 2], 3), "]"),
         paste0("[", round(confint_wald_bayesglm["NV", 1], 3), ", ",
                round(confint_wald_bayesglm["NV", 2], 3), "]"),
         paste0("[", round(confint_wald_logistf["NV", 1], 3), ", ",
                round(confint_wald_logistf["NV", 2], 3), "]"),
         paste0("[", round(confint_profile_logistf["NV", 1], 3), ", ",
                round(confint_profile_logistf["NV", 2], 3), "]")),
  PI = c(paste0("[", round(confint_wald_glm["PI", 1], 3), ", ",
                round(confint_wald_glm["PI", 2], 3), "]"),
         paste0("[", round(confint_wald_bayesglm["PI", 1], 3), ", ",
                round(confint_wald_bayesglm["PI", 2], 3), "]"),
         paste0("[", round(confint_wald_logistf["PI", 1], 3), ", ",
                round(confint_wald_logistf["PI", 2], 3), "]"),
         paste0("[", round(confint_profile_logistf["PI", 1], 3), ", ",
                round(confint_profile_logistf["PI", 2], 3), "]")),
  EH = c(paste0("[", round(confint_wald_glm["EH", 1], 3), ", ",
                round(confint_wald_glm["EH", 2], 3), "]"),
         paste0("[", round(confint_wald_bayesglm["EH", 1], 3), ", ",
                round(confint_wald_bayesglm["EH", 2], 3), "]"),
         paste0("[", round(confint_wald_logistf["EH", 1], 3), ", ",
                round(confint_wald_logistf["EH", 2], 3), "]"),
         paste0("[", round(confint_profile_logistf["EH", 1], 3), ", ",
                round(confint_profile_logistf["EH", 2], 3), "]")))
print(df_confint_summary)
```

```
##                                Model      Intercept                    NV
## 1                        GLM (Wald) [1.095, 7.514] [-3344.624, 3380.996]
## 2                Bayesian GLM (Wald) [0.988, 6.435]          [0.2, 6.396]
## 3   Logistic Firth Regression (Wald) [0.954, 6.595]        [0.058, 5.801]
## 4 Logistic Firth Regression (Profile) [1.083, 7.209]         [0.61, 7.855]
##                PI               EH
## 1 [-0.129, 0.045]  [-4.56, -1.245]
## 2 [-0.101, 0.043] [-4.076, -1.181]
## 3  [-0.109, 0.04] [-4.081, -1.127]
## 4  [-0.124, 0.04] [-4.365, -1.233]
```

- Each cell contains a **95% confidence interval** (CI) for the corresponding coefficient, presented in the form `[lower bound, upper bound]`.
- **GLM (Wald)**: The CI for `NV` is very wide (`[-3344.624, 3380.996]`), indicating **high uncertainty** in the estimation. The inclusion of both negative and positive values implies that the effect of `NV` could be negative, positive, or zero, making it **non-significant**. `PI` has a CI of `[-0.129, 0.045]`, which includes 0, suggesting that `PI` might not be statistically significant. The `EH` CI (`[-4.560, -1.245]`) does not include 0, suggesting a **statistically significant negative effect** of `EH` on HG.
- **Bayesian GLM (Wald)**: The Bayesian GLM has **narrower CIs** compared to the GLM, indicating **more stable and regularized estimates** due to the use of priors. The CI for `NV` (`[0.200, 6.396]`) suggests a **significant positive effect** since it does not include 0. The CI for `PI` (`[-0.101, 0.043]`) includes 0, indicating it might **not be significant**. The `EH` CI (`[-4.076, -1.181]`) suggests a **significant negative effect** on `HG`.

- **Logistic Firth Regression (Wald)**: The CIs for `NV`, `PI`, and `EH` are somewhat similar to those from **Bayesian GLM**. The `NV` CI (`[0.058, 5.801]`) does not include 0, suggesting a significant **positive effect**. The `PI` CI (`[-0.109, 0.040]`) includes 0, indicating **non-significance**. The `EH` CI (`[-4.081, -1.127]`) does not including 0, suggesting a **significant negative effect**.
- **Logistic Firth Regression (Profile)**: The `NV` CI (`[0.610, 7.855]`) is slightly **wider** compared to the Wald CI, but still does not include 0, indicating **positive significance**. The `PI` CI (`[-0.124, 0.040]`) includes 0, suggesting it is not significant. The `EH` CI (`[-4.365, -1.233]`) is significant as it does not include 0, indicating a negative relationship.
- When calculating **profile likelihood confidence intervals** for GLM, multiple warnings were encountered indicating that the fitted probabilities were numerically close to 0 or 1. This suggests issues with convergence, possibly due to **complete or quasi-complete separation** in the data. To avoid misleading results, **Wald confidence** intervals were used instead
- I am unable to get the **profile confidence interval** for **Bayesian GLM**, since the original model fit had not converged properly, which led to errors.

3. p-values for Non-Intercept Coefficients

```r
# 1. Wald p-values for GLM
p_values_wald_glm <- summary(model_glm)$coefficients[-1, 4]

# Likelihood Ratio Test for GLM
lrt_glm <- anova(model_glm, test = "LRT")
lrt_p_values_glm <- lrt_glm$`Pr(>Chi)`[-1]
names(lrt_p_values_glm) <- c("NV", "PI", "EH")

# 2. Wald p-values for Bayesian GLM
p_values_wald_bayesglm <- summary(model_bayesglm)$coefficients[-1, 4]

# Likelihood Ratio Test for Bayesian GLM
lrt_bayesglm <- anova(model_bayesglm, test = "LRT")
lrt_p_values_bayesglm <- lrt_bayesglm$`Pr(>Chi)`[-1]
names(lrt_p_values_bayesglm) <- c("NV", "PI", "EH")
```

```r
# 3. Wald p-values for Logistic Firth Regression
summary_logistf <- summary(model_logistf)
```

```
## logistf(formula = HG ~ NV + PI + EH, data = data_e)
##
## Model fitted by Penalized ML
## Coefficients:
##                    coef   se(coef) lower 0.95  upper 0.95      Chisq
## (Intercept)  3.77455951 1.43900672  1.0825371  7.20928050  8.1980136
## NV           2.92927330 1.46497415  0.6097244  7.85463171  6.7984572
## PI          -0.03475175 0.03789237 -0.1244587  0.04045547  0.7468285
## EH          -2.60416387 0.75362838 -4.3651832 -1.23272106 17.7593175
##                         p method
## (Intercept) 4.193628e-03      2
## NV          9.123668e-03      2
## PI          3.874822e-01      2
## EH          2.506867e-05      2
##
```

```
## Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None
##
## Likelihood ratio test=43.65582 on 3 df, p=1.78586e-09, n=79
## Wald test = 21.66965 on 3 df, p = 7.641345e-05
```

```
p_values_wald_logistf <- summary_logistf$prob[-1]
names(p_values_wald_logistf) <- c("NV", "PI", "EH")
```

```
# Likelihood Ratio Test for Logistic Firth Regression
lrt_logistf <- logistftest(model_logistf)
lrt_p_value_logistf <- lrt_logistf$`p`
```

```
# Create data frames to compare p-values
# 1. Wald p-values comparison
df_wald_p_values <- data.frame(
  Model = c("GLM", "Bayesian GLM", "Logistic Firth Regression"),
  NV = c(p_values_wald_glm["NV"], p_values_wald_bayesglm["NV"], p_values_wald_logistf["NV"]),
  PI = c(p_values_wald_glm["PI"], p_values_wald_bayesglm["PI"], p_values_wald_logistf["PI"]),
  EH = c(p_values_wald_glm["EH"], p_values_wald_bayesglm["EH"], p_values_wald_logistf["EH"])
)
print(df_wald_p_values)
```

```
##                        Model          NV        PI           EH
## 1                        GLM 0.991543235 0.3413330 5.973924e-04
## 2              Bayesian GLM 0.036929294 0.4272606 3.726778e-04
## 3 Logistic Firth Regression 0.009123668 0.3874822 2.506867e-05
```

- **GLM**: `NV` has a p-value of `0.9915`, suggesting it is not significant in this model. `PI` has a p-value of `0.3413`, which is also not significant. `EH` has a p-value of `5.9739e-04` (< 0.001), which is highly significant.
- **Bayesian GLM**: `NV` has a p-value of `0.0369`, suggesting it is significant at the 5% level. `PI` has a p-value of `0.4273`, which is not significant. `EH` has a p-value of `3.7268e-04` (< 0.001), which is highly significant.
- **Logistic Firth Regression**: `NV` has a p-value of `0.0091`, suggesting it is significant at the 1% level. `PI` has a p-value of `0.3875`, which is not significant. `EH` has a p-value of `2.5069e-05` (< 0.001), indicating strong significance.
- For all three models, `EH` is **consistently significant**, which suggests that it has a meaningful effect on the response variable `HG`. However, `PI` is not significant across any of the models. `NV` is significant in the **Bayesian GLM** and **Logistic Firth Regression**, but not in the regular GLM.

```
# 2. Likelihood Ratio Test p-values comparison
df_lrt_p_values <- data.frame(
  Model = c("GLM", "Bayesian GLM", "Logistic Firth Regression"),
  # NV p-values for GLM and Bayesian GLM, NA for Firth
  NV = c(lrt_p_values_glm["NV"], lrt_p_values_bayesglm["NV"], NA),
  # PI p-values for GLM and Bayesian GLM, NA for Firth
  PI = c(lrt_p_values_glm["PI"], lrt_p_values_bayesglm["PI"], NA),
  # EH p-values for GLM and Bayesian GLM, NA for Firth
  EH = c(lrt_p_values_glm["EH"], lrt_p_values_bayesglm["EH"], NA),
  # Global p-value for the Logistic Firth Regression
  Global = c(NA, NA, lrt_p_value_logistf))
print(df_lrt_p_values)
```

```
##                        Model           NV        PI           EH      Global
## 1                        GLM 5.322163e-08 0.6957899 8.777197e-06          NA
## 2              Bayesian GLM 5.322163e-08 0.6957899 1.377741e-05          NA
## 3 Logistic Firth Regression           NA        NA           NA 1.78586e-09
```

- Note: The Logistic Firth Regression model does not provide individual likelihood ratio test p-values for each coefficient. Instead, it provides a global p-value that tests the overall significance of all predictors together.
- **GLM and Bayesian GLM**: The p-values for the likelihood ratio tests of `NV`, `PI`, and `EH` are similar for both models. The p-value for `NV` (`5.3222e-08`) is very small, suggesting that `NV` is **highly significant** in explaining `HG`. `PI` has a p-value of `0.6958`, which indicates that it is not significant. `EH` has a p-value ranging from `8.7772e-06` (GLM) to `1.3777e-05` (Bayesian GLM), indicating that `EH` is significant.
- **Logistic Firth Regression**: The **global p-value** (`1.7859e-09`) indicates that, collectively, the predictors (`NV`, `PI`, `EH`) significantly improve the model fit compared to the null model. However, individual likelihood ratio test p-values for `NV`, `PI`, and `EH` are not directly provided by logistftest().

Question 4

```r
# load the data
data("bioChemists", package="pscl")
data_bc <- bioChemists
```

Fit a Negative Binomial Model

```r
model_nb <- glm.nb(art ~ fem + mar + kid5 + phd + ment,
                   data = data_bc)
```

Simulate 1000 New Responses

```r
set.seed(123)  # For reproducibility
sim_vals <- simulate(model_nb, nsim = 1000)

# Compute the total number of zero observations for each simulation
zero_counts_sim <- colSums(sim_vals == 0)
summary(zero_counts_sim)
```

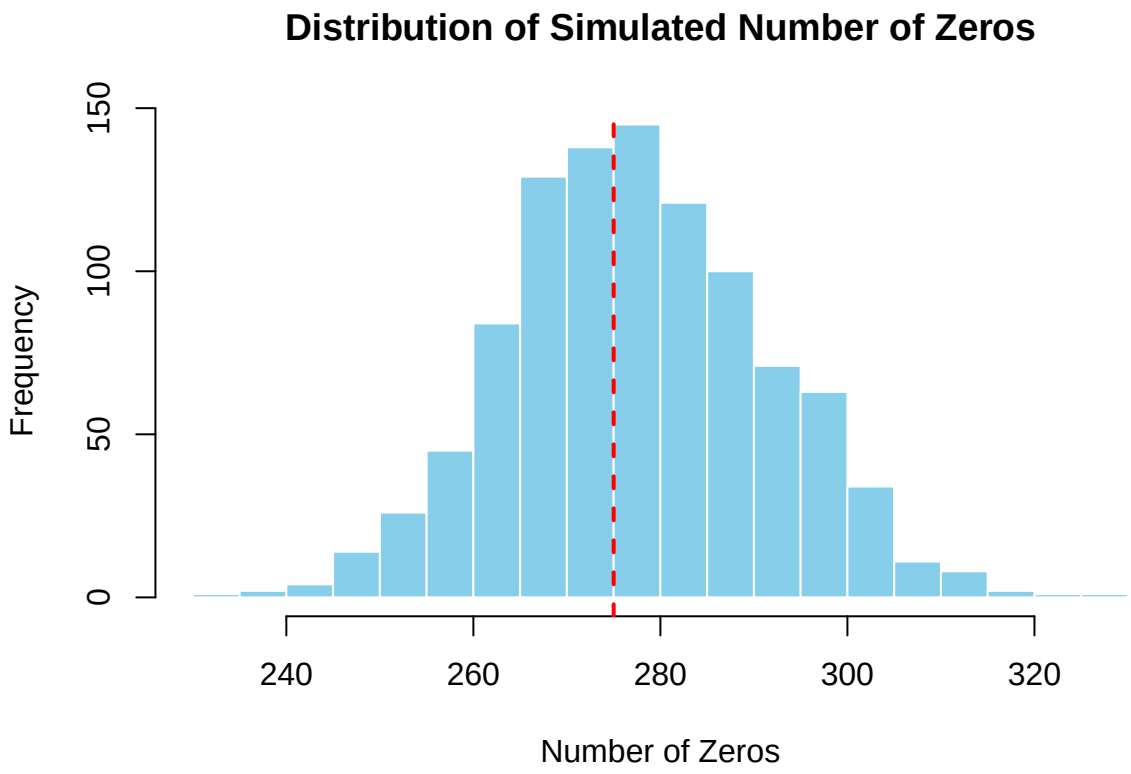
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   234.0   268.0   277.0   278.1   288.0   326.0
```

```r
# Number of zeroes in the observed data
obs_zero_count <- sum(data_bc$art == 0)
obs_zero_count
```

```
## [1] 275
```

Draw a Histogram of the Simulation

```
hist(zero_counts_sim, breaks = 30,
     main = "Distribution of Simulated Number of Zeros",
     xlab = "Number of Zeros",
     ylab = "Frequency", col = "skyblue",
     border = "white")
abline(v = obs_zero_count,
       col = "red",
       lwd = 2, lty = 2) # indicate the 0s in the observed data
```



Compute p-value

```
p_value_one_sided <- mean(
  zero_counts_sim >= obs_zero_count)
cat("One-side p-value for zero inflation test based on simulation:",
    p_value_one_sided, "\n")
```

```
## One-side p-value for zero inflation test based on simulation: 0.6
```

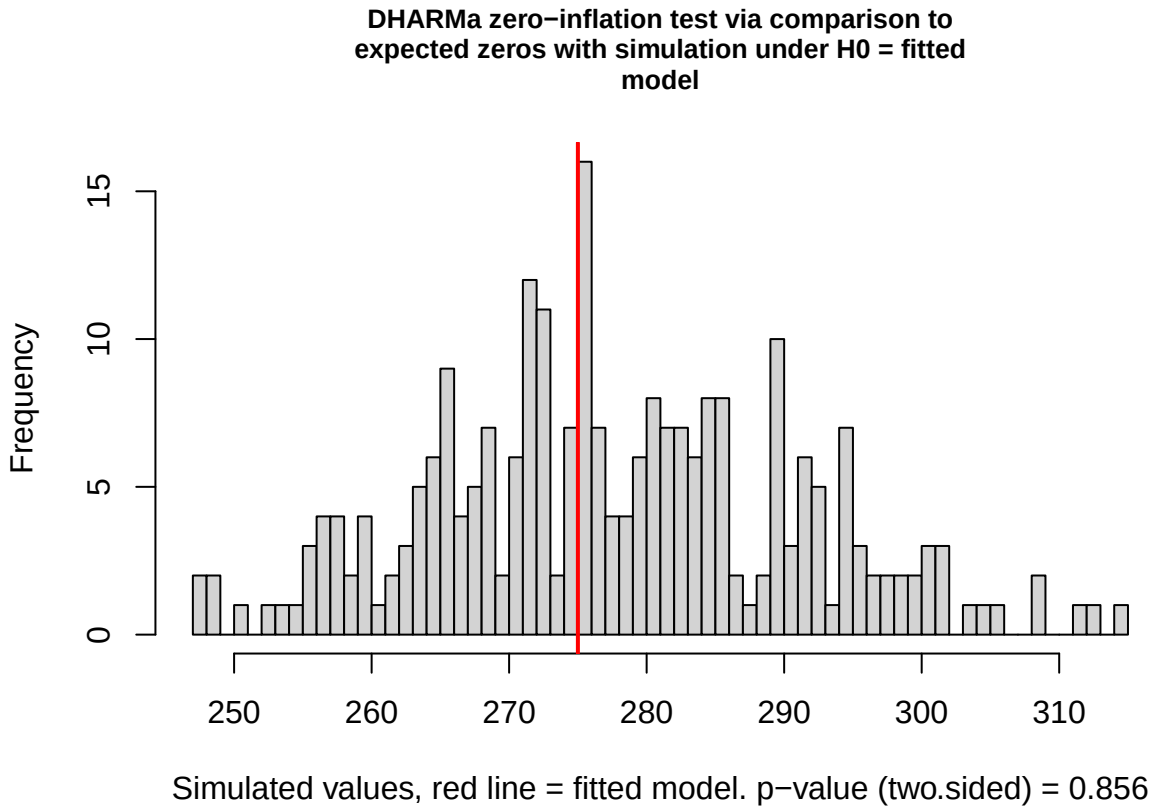
```
p_value_two_sided <- mean(
  abs(zero_counts_sim - mean(zero_counts_sim)) >= abs(obs_zero_count - mean(zero_counts_sim)))
cat("Two-sided p-value for zero inflation test based on simulation:",
    p_value_two_sided, "\n")
```

```
## Two-sided p-value for zero inflation test based on simulation: 0.826
```

- The two-sided p-value for zero-inflation test based on simulation is `0.826`, which is a **high** value, indicating no significant evidence of zero inflation. The **negative binomial model** appears to adequately explain the zero counts observed in the data.

Compare with DHARMa zero-inflation test

```
residuals_dharma <- simulateResiduals(model_nb)
zi_test_result <- testZeroInflation(residuals_dharma)
```




```
# Print DHARMa test result
print(zi_test_result)
```

```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 0.98847, p-value = 0.856
## alternative hypothesis: two.sided
```

- The two-sided p-value for DHARMa zero-inflation test is `0.856`, which is a high value and close to `0.826` (the two-sided p-value we calculated before), suggesting that the observed number of zeros is

very much in line with the distribution of zeros in the simulated data. The **negative binomial model** sufficiently explains the number of zeros.

Reference

- OpenAI, ChatGPT (2024). Assistance with R programming and output analysis. Accessed on October 2, 2024.
- Dr. Ben Bolker, Lecture Notes for "Statistical Modeling", McMaster University. Accessed on September 10, 2024.

Appendix

- Here is a list of prompts I used in **ChatGPT 4o with canvas**:

1. In R programming, how to plot a bar chart with a binary predictor on the x-axis and a binary response on the y-axis?
2. How to use `DHARMa` for diagnostic checking of (generalized) linear mixed models?
3. How to create efficient plots with `dotwhisker`?
4. How to create effect plots with `ggeffects`?
5. How to fit a GLM, with an offset?
6. How to check for overdispersion in a GLM?
7. How is `mle2` different from `glm`? How to fit the same model as `glm` with the formula interface of `bbmle`?
8. How to write a negative log-likelihood function and use `optim` to fit the GLM?
9. How to use `DHARMa::testZeroInflation()` in R?