L. Scott-Hayward and *V.Popov*

MT5761: Statistical Modelling with GLMs

University of
St Andrews

Contents

Welcome

Welcome to MT5761 Applied Statistical Modelling Using GLMs

This applied statistics module covers the main aspects of linear models (LMs) and generalized linear models (GLMs). In each case the course describes model specification, various options for model selection, model assessment and tools for diagnosing model faults. Common modelling issues such as collinearity and residual correlation are also addressed, and as a consequence of the latter the Generalized Least squares (GLS) method is described. The GLM component has emphasis on models for count data and presence/absence data while GLMs for multinomial (sometimes called choice-based models) are also covered for nominal and ordinal response outcomes. The largest part of the course material is taught inside an environmental impact assessment case study with reality-based research objectives. Political and medical examples are used to illustrate the multinomial models.

Prerequisites

- Programming basics in R

Learning outcomes

- Understand the key concepts and terminology used in statistical modelling
- Use R to fit linear and generalised linear models in R
- Recognise practical issues with fitting these models
- Checking model fit
- Perform model comparisons

Recommended reading

- Generalized Linear Models With Examples in R
- Modelling Count Data
- Applied Regression Analysis and Generalized Linear Models
- Chance Encounters

Data files

All data files can be found on Moodle.

Assessment

50% written exam and 50% **individual** coursework

Lateness policy

The School has a lateness policy. The standard policy is an initial penalty of 15% of the maximum available mark, then a further 5% per 8-hour period, or part thereof for work submitted late without good reason.

Work submitted late for good reason

If students have a justified reason for submitting work late, then the various University's policies relating to extenuating circumstances apply. In these circumstances, students must as soon as possible submit a self-certificate of absence and contact the relevant member of School (usually the module coordinator). You will

then be advised whether further documentation is required and what format this documentation will take.

Acknowledgements

We are indebted to all the statisticians who made some stats possible.

1

Introduction

We are going to use some environmental impact assessment (EIA) data to learn about Generalised Least Squares (GLS) models (a type of linear model) and Generalized Linear Models (GLMs).

The EIA data in focus is collected from a site which contains two off-shore wind farms of the coast of Denmark. These wind farms are among the largest in the world (Figure 1.1) and generate large amounts of renewable energy[1].

```
knitr::include_graphics('figures/nysted-farm.jpg')
```

FIGURE 1.1 Nysted I wind farm

Here are some details about the data:

[1] http://en.wikipedia.org/wiki/Nysted_Wind_Farm

- The data are collected from aircraft travelling along transects (pre-determined tracks) across the water. The data collected were counts of birds seen from the aircraft, within some distance of the transects (Figure 1.2).

```
df <- read.csv("data/NystedFarms.csv")
library(ggplot2)
library(dplyr)
```

```
# Figure 2
ggplot() +
  geom_point(aes(x=XPos, y=YPos),
             colour='lightgrey', data=df) +
  geom_point(aes(x=XPos, y=YPos, size=log(Count)),
             colour="#377eb8", data=subset(df, Count>0)) +
  theme_bw() +
  theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank())+
  coord_equal()
```

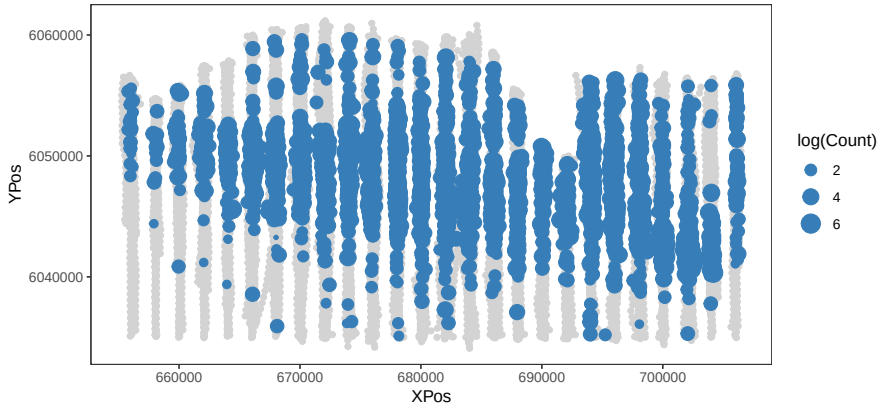


FIGURE 1.2 Transect locations (grey lines) and locations with non-zero counts (blue dots). The size of the blue dots are scaled based on the associated count at each location and are shown along the transects.

- Environmental data (e.g. Depth) is also available for each spatial location visited along the transects.
- The count data we will use as input for the models in this course are already corrected for the animals that were missed due to the imperfect detection process (animals farthest from the plane are more difficult to see).
- This inflation for the numbers overlooked was undertaken using a survey method called **distance sampling**[2]; these methods are routinely used for this purpose but are outside the scope of this course.

The objectives of the monitoring (and impact assessment) programme was to:

1. characterise the 'baseline' abundance and distribution of a number of bird populations in the area, and
2. assess any changes in the abundance and/or distribution of these populations post-installation/operation of two large wind farms.

For this reason we can partition the data into three phases:

- A: before any farms were installed in the area (the 'baseline' phase)
- B: after one wind farm (Rodsand I) was installed
- C: after two wind farms were installed (Rodsand I and II)

There were multiple transects surveyed, across many survey days, within the three phases (A, B and C), and the survey effort was highly uneven. There were:

TABLE 1.1: Table showing the survey effort in the three construction phases.

Phase	Survey Effort (days)
A	11
B	13
C	5

There were different areas associated with each observed count (between 0.00157 and 0.956 square kilometres).

We will be considering the (estimated) abundance per unit area (referred to from now on as 'density') as the response variable in all analyses to account for this

[2] see Thomas, Buckland, Burnham, Anderson, Laake, Borchers & Strindberg. 2006. Encyclopedia of Environmetrics; http://tinyurl.com/or3rkba

variable effort and we will also be ignoring the uncertainty in this response due to the correction for imperfect detection[3].

Note, we know from the outset that the densities might well be correlated in time/space: densities along transects are likely to be similar to each other (e.g. there are likely to be clusters of high values (and low values) for consecutive sites along transects).

1.1 Research Questions

In line with the programme objectives, we are going to ask some questions of the data:

- Does density appear to have changed across phases?
- What are the best predictors of density and what do these relationships look like?
- Do the animals appear to have redistributed across phases? If so, what does that redistribution look like?

1.2 Exploratory Data Analysis

We will begin to address these questions using some exploratory work.

```
# Figure 3
# Compute CIs for each construction phase using normal approx.
alpha <- 0.05
z_alpha_2 <- abs(qnorm(alpha/2)) # z_{alpha/2}
stats <- df %>%
  group_by(Phase) %>%
  summarise(Mean=mean(Count),
            Var=var(Count),
            N=n(),
            CI=z_alpha_2*sqrt(Var/N))

# Plot mean and 95% CIs
```

[3]The estimated abundances per unit area will have some uncertainty associated with these since they are themselves a result of a model.

```
ggplot(stats, aes(x=Phase, y=Mean, colour=Phase)) +
  geom_errorbar(aes(ymin=Mean-CI, ymax=Mean+CI), size=1) +
  geom_point(size=4) +
  geom_text(aes(x=Phase, y=Mean+1.2*CI, label=paste0('N=', N))) +
  ylab("Counts")
```

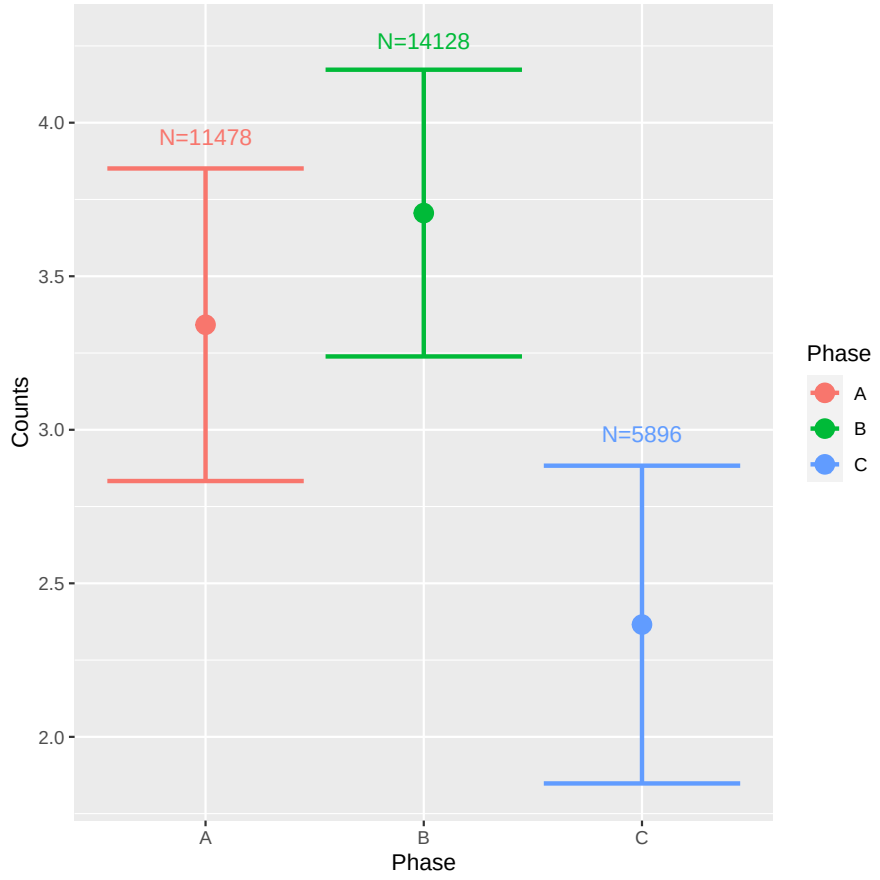


FIGURE 1.3 Mean counts per unit area (with 95% CIs) across phases.

```
# Compute CIs for density using normal approx.
alpha <- 0.05
z_alpha_2 <- abs(qnorm(alpha/2)) # z_{alpha/2}
stats <- df %>%
  group_by(YearMonth, Phase) %>%
  summarise(Mean=mean(Count/Area),
```

```
            Var=var(Count/Area),
            N=n(),
            CI=z_alpha_2*sqrt(Var/N))
```

```
# Plot mean and 95% CIs
ggplot(stats, aes(x=YearMonth, y=Mean, colour=Phase)) +
  geom_errorbar(aes(ymin=Mean-CI, ymax=Mean+CI), size=1) +
  geom_point(size=2) +
  theme(axis.text.x=element_text(angle=90)) +
  ylab("Abundance per unit area") +
  xlab("Year/Month")
```

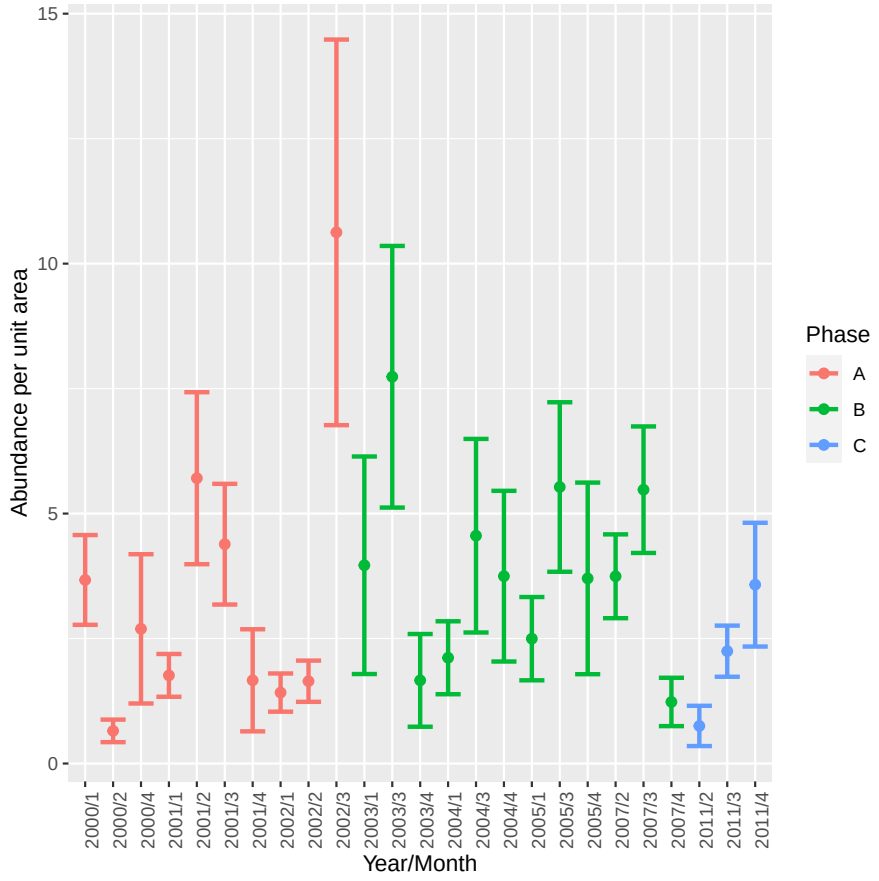


FIGURE 1.4 Average counts per unit area (with 95% CIs) across years and months (with phases indicated).

Simple 95% confidence intervals[4] for each phase (Figure 1.3) suggest there may be fewer animals on average in phase C (compared to phases A and B)

The confidence intervals in Figure 1.3 should be treated with caution; we suspect the densities are correlated along transects (and thus are unlikely to be independent of each other).

However, there don't appear to be any compelling differences in average numbers between phases A and B, since the confidence intervals share values (Figure 1.3). Even if we had seen significant differences across phases, any apparent changes in density could be due to shorter temporal coverage in phase C compared with earlier phases (5 days compared with 11 and 13 days; Table 1.1 , page 9).

For instance, density might be lower in those particular months, on average, anyway – regardless of wind farm development (Figure 1.4).

We would also need to be sure that any differences in densities across phases (and thus stages in wind farm development) are not due to other predictors.

For these (and other) reasons, it makes sense to consider several covariates simultaneously even if the primary interest solely lies in wind-farm related differences in density across phases.

In order to find good predictors of (estimated) densities in this area and to examine what these relationships might look like, scatterplots and boxplots can also be useful (Figure 1.5:

```
# Libraries
library(gridExtra) # to arrange ggplots in a grid

# Create list of plots of density against other covariates
p <- list()
for (covariate in c("XPos", "YPos", "Depth", "DistCoast"))
{
  p[[covariate]] <- ggplot(df, aes_string(x=covariate,
                                          y="Count/Area")) +
    geom_point(alpha=1/2) +
    ylab("Abundance per unit area")
}

# Arrange ggplots in a grid
grid.arrange(grobs=p)
```

[4] 85% intervals may be better for this kind of informal comparison of means

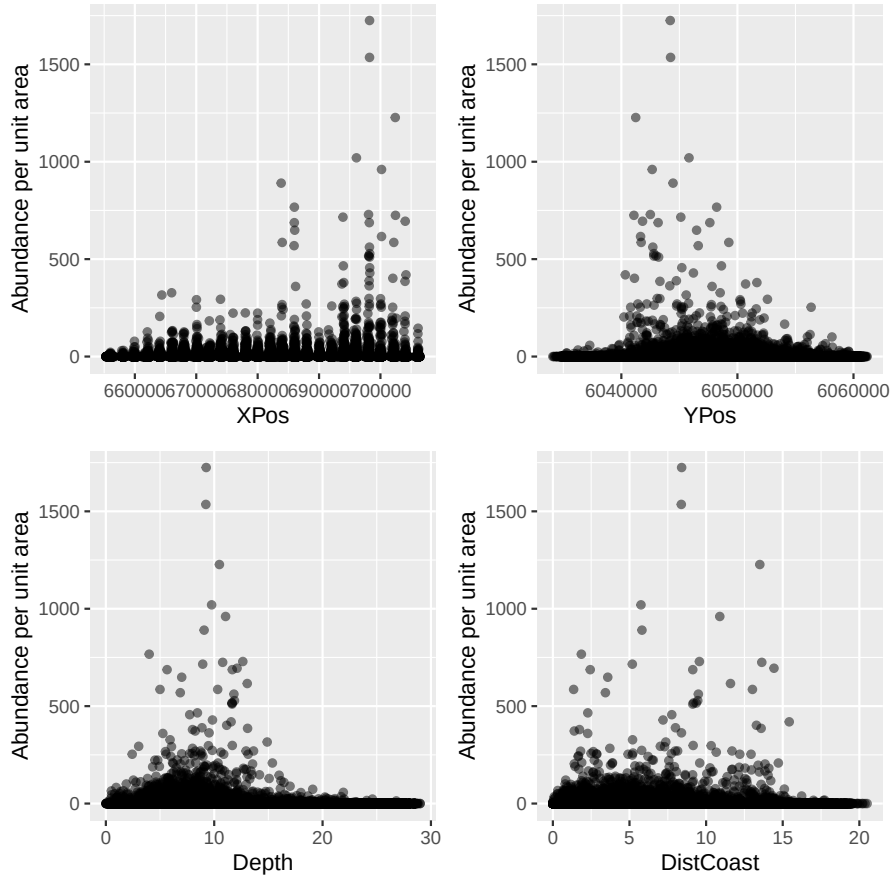


FIGURE 1.5 Scatterplots for potential model covariates and estimated densities.

The highest densities (Figure 1.5) seem to be associated with:

- moderate depths
- central values of the Y-coordinate
- large values of the X-coordinate
- locations near the coast

These plots tell us that some covariate relationships might be worth pursuing and also help provide a quick check for outliers in x and/or y.

The `distance from coast' and`depth' variables can also be viewed spatially for the surveyed area (Figures 1.6 and 1.7.

The following are `quilt plots' from the`fields' library.

```
library(fields)
quilt.plot(df$XPos, df$YPos, df$DistCoast, nx=35, ny=35,
           xlab="XPos", ylab="YPos", asp=1)
```

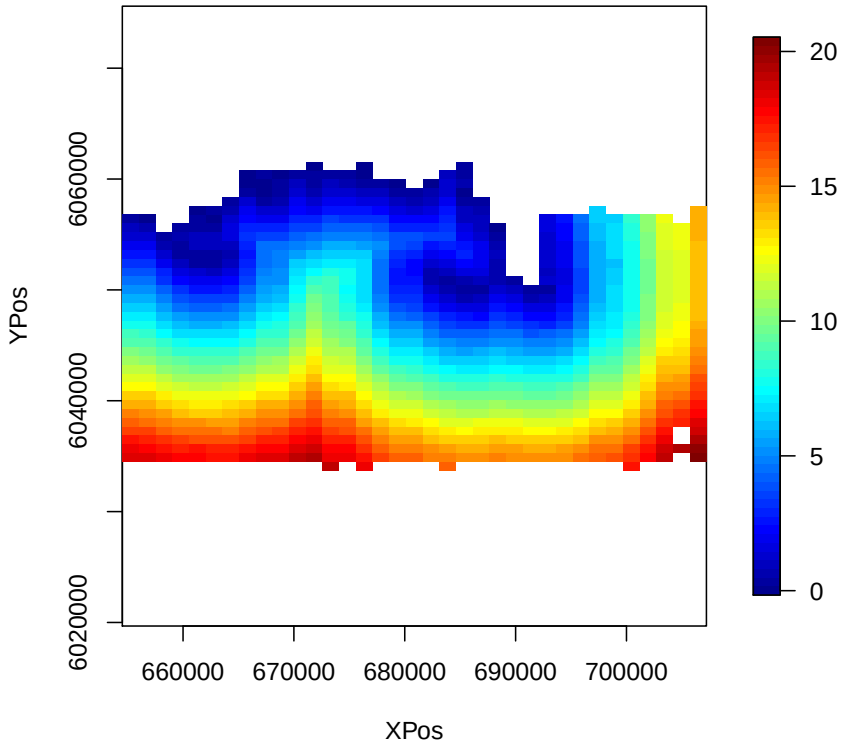


FIGURE 1.6 Distance from coast across the surveyed area. The colours shown in each grid cell represent the average distance from coast in that cell.

```
quilt.plot(df$XPos, df$YPos, df$Depth, nx=35, ny=35,
           xlab="XPos", ylab="YPos", asp=1)
```

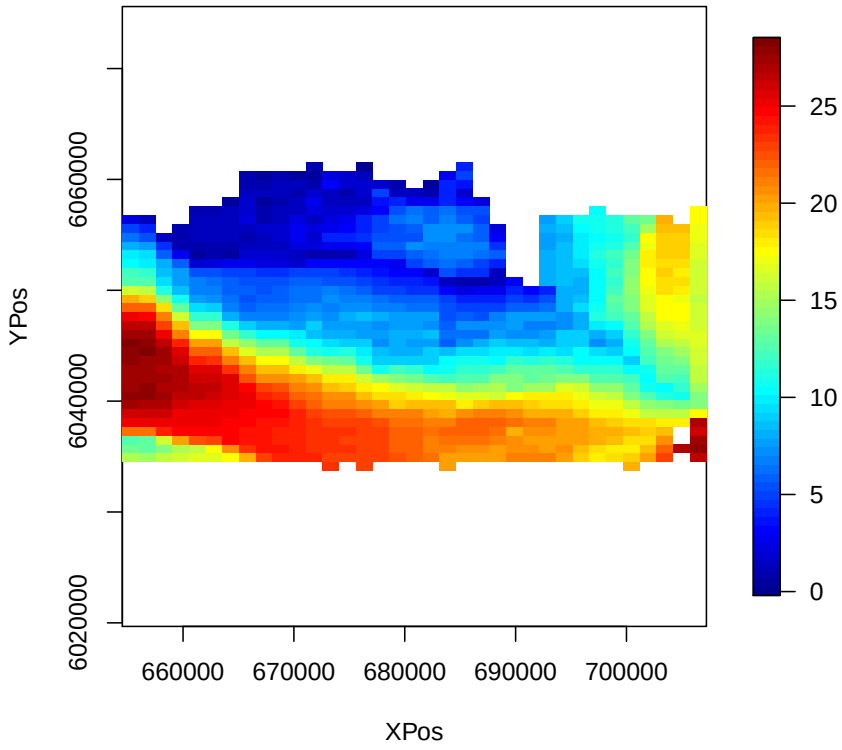
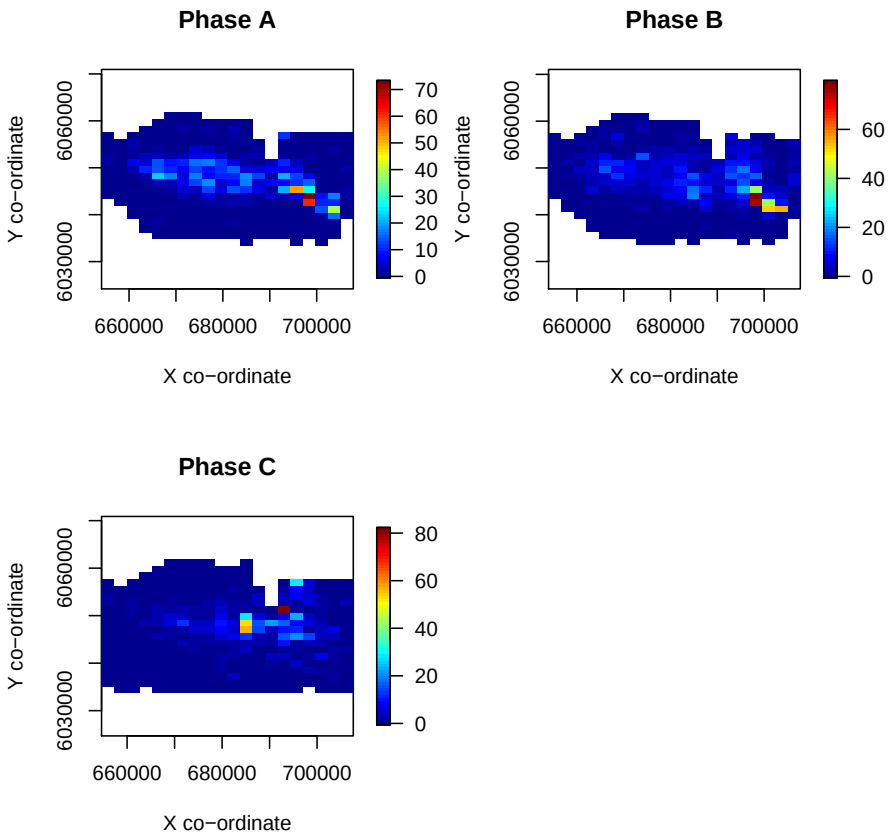


FIGURE 1.7 Depth across the surveyed area. The colours shown in each grid cell represent the average depth in that cell.

We can also examine whether density appears to have changed over space and phase by using point plots like Figure 1.2 or quilt plots like Figure 1.6.

```
par(mfrow=c(2,2))
quilt.plot(df$XPos[df$Phase=="A"], df$YPos[df$Phase=="A"], df$Count[df$Phase=='A']/df$Area
           main="Phase A", xlab="X co-ordinate", ylab="Y co-ordinate",
           nrow=20, ncol=20, asp=1)
quilt.plot(df$XPos[df$Phase=="B"], df$YPos[df$Phase=="B"], df$Count[df$Phase=='B']/df$Area
           main="Phase B", xlab="X co-ordinate", ylab="Y co-ordinate",
           nrow=20, ncol=20, asp=1)
quilt.plot(df$XPos[df$Phase=="C"], df$YPos[df$Phase=="C"], df$Count[df$Phase=='C']/df$Area
           main="Phase C", xlab="X co-ordinate", ylab="Y co-ordinate",
```

```
            nrow=20, ncol=20, asp=1)
par(mfrow=c(1,1))
```




```
ggplot() +
  geom_point(aes(x=XPos, y=YPos), colour='lightgrey', data=df) +
  geom_point(aes(x=XPos, y=YPos, size=log(Count/Area)),
             colour="#377eb8", data=subset(df, Count>0)) +
  scale_size(range=c(1,3)) +
  theme_bw() +
  theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank()) +
  coord_equal() +
  facet_wrap(~Phase,nrow = 3)
```

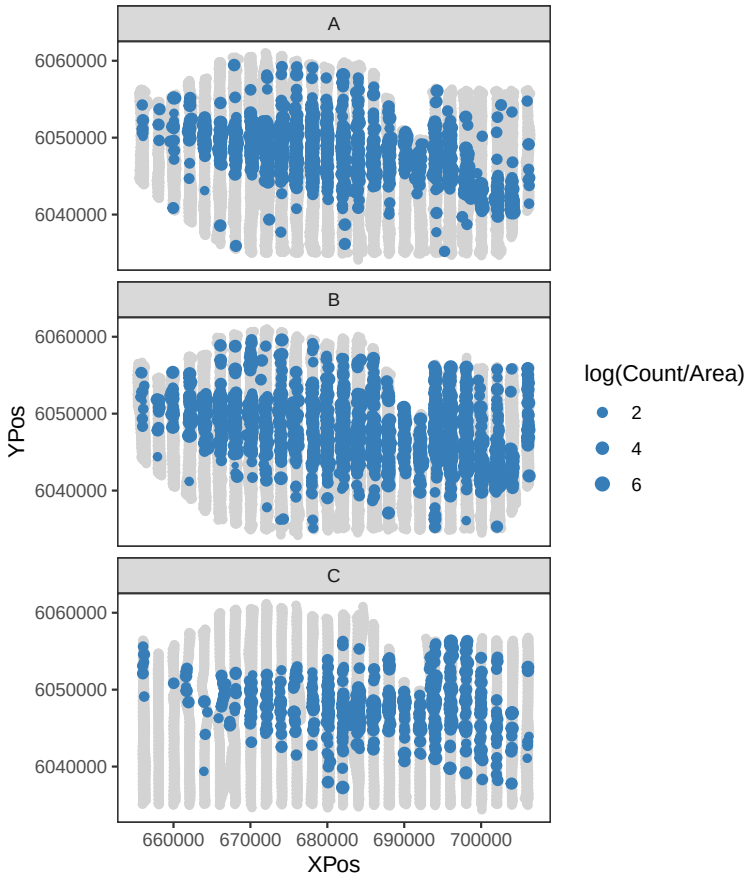


FIGURE 1.8 Geo-referenced densities of animals in each phase. The colours represent the average density in the 20 $imes$ 20 grid cell for that phase.

There seems to be some evidence of redistribution across phases (Figure 1.8, particularly in phase C compared with phases A and B.

This tells us that we might want to consider models that permit redistribution in the fitted surfaces across phases (e.g. via 'interaction' terms in our model).

2

Multiple Linear Regression

We are going to try and use variables in the EIA data to predict density in this area using linear regression. **Regression** is a way to study relationships between variables. There are two main reasons why we may want to do this:

Description: We may be genuinely interested in finding the relationship between such variables (e.g. what, if any, is the relationship between density and depth?)

Prediction: If there is a relationship between the variables under study, then knowledge of some variables will help us predict others (e.g., if we know that density changes with depth on the transects, then knowing the depth of a site will help us predict density off the transects).

Linear regression models contain **explanatory** variable(s) that help us explain or predict the behaviour of the **response** variable (whose behaviour we want to predict).

Linear models assume constantly increasing or decreasing relationships between each explanatory variable and the response.

Note, there is some overlap between year and phase information, since the phases move from A, B to C over the years. Specifically, phase C only occurs in 2011 while phase B occurs 2003-2007 and phase A in 2000-2002.

```
knitr::kable(table(df$Phase, df$Year))
```

	2000	2001	2002	2003	2004	2005	2007	2011
A	3901	4273	3304	0	0	0	0	0
B	0	0	0	3109	4410	3314	3295	0
C	0	0	0	0	0	0	0	5896

This will mean that both phase and year will be unable to be fitted together in a model (for *collinearity* reasons, see the notes to the introductory statistics course)

2.1 Model specification

Multiple regression models have $p \geq 1$ explanatory variables which can be written as:

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} +, ..., +\beta_p x_{pit} + \varepsilon_{it}$$

where

$$\varepsilon_{it} \sim N(0, \sigma^2) \qquad \text{for} \quad \forall i, t$$

where y_{it} is the response (density for transect i at time t), β_0, is the intercept parameter, β_1, β_2,...,β_p are slope coefficients and $x_{1it}, x_{2it}, ..., x_{pit}$ are the explanatory variables.

To start with we will use 6 predictors/covariates (i.e. $p = 6$):

- x_{1it} represents the X coordinate (for the i-transect at time t)
- x_{2it} represents the Y-coordinate (for the i-transect at time t)
- x_{3it} represents distance from coast (for the i-transect at time t)
- x_{4it} represents depth (for the i-transect at time t)
- x_{5it} represents month=2,
- x_{6it} represents month=3,
- x_{7it} represents month=4,
- x_{8it} represents phase B (for the i-transect at time t).
- x_{9it} represents phase C (for the i-transect at time t).

At this point we are going to assume the relationship between each continuous covariate and density is linear.

In order to ask if there has been some redistribution across phases, we can quantify the evidence that (a particular sort of) density pattern in the `X` or `Y` direction differs across phases.

We can ask this by introducing *interaction* effect(s) which permit a different slope coefficient (in this case based around `X` or `Y`) for different levels of the phase variable (`A`, `B` or `C`).

We can implement this in a model using `phase:X` and `phase:Y` terms.

In our interaction-based model we have:

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} + \beta_3 x_{3it} + ... + \beta_{13} x_{13it} + \varepsilon_{it}$$

where $\beta_0 - \beta_9$ and $x_{1it} - -x_{9it}$ are as described before and relate to `X`, `Y`, `DistCoast`, `depth`, `month` and `phase`. The new aspects of the output are as follows:

- X:phaseB: β_{10} is the expected change in the slope coefficient for the X relationship in phase B compared with the X relationship in phase A
- X:phaseC: β_{11} is the expected change in the slope coefficient for the X relationship in phase C compared with the X relationship in phase A
- Y:phaseB: β_{12} is the expected change in the slope coefficient for the Y relationship in phase B compared with the Y relationship in phase A
- Y:phaseC: β_{13} is the expected change in the slope coefficient for the Y relationship in phase C compared with the Y relationship in phase A

2.2 Model fitting

Estimating two or more slope coefficients is straightforward using least-squares. We find estimates of $\beta_0, \beta_1, ..., \beta_p$ that *best* fit the data.

We can do this by finding the minimum of:

$$\sum_{i=1}^{s}\sum_{t=1}^{n_i}(y_{it} - (\hat{\beta}_0 + \hat{\beta}_1 x_{1it} + ... + \hat{\beta}_p x_{pit}))^2$$

The estimates can be found using:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$$

where $\mathbf{y}$ is the $N \times 1 = 31502 \times 1$ response vector and $\mathbf{X}$ is a $N \times (p+1)$ covariate/design matrix.

Least-squares have some nice properties and are what's called 'Best Linear Unbiased Predictors (BLUE)' :

- The estimates are unbiased, i.e., the distribution of estimates is centred around the true parameter value (which means they are neither systematically too large or too small)
- The estimates are consistent (i.e., if we increase the sample size we get closer, on average, to the true parameter values)
- The estimates are 'best' since there are no other unbiased estimators that are more efficient – efficiency means the estimates get closer on average to the true parameter more often compared with another estimator.

For a 7 minute clip about least squares estimation, see here

The estimates $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ are produced by the `lm` function in `R` in the `Estimate` column. For example,

```
df$FMonth<-as.factor(df$Month)
workingModel_Int<- lm(Count/Area ~ XPos + YPos + DistCoast + Depth + FMonth +
                        Phase + Phase:XPos + Phase:YPos, data=df)
summary(workingModel_Int)
```

```
Call:
lm(formula = Count/Area ~ XPos + YPos + DistCoast + Depth + FMonth +
    Phase + Phase:XPos + Phase:YPos, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
 -12.27   -5.28   -2.96   -0.16 1715.13

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.279e+03  3.308e+02   9.913  < 2e-16 ***
XPos         8.446e-05  1.958e-05   4.314 1.61e-05 ***
YPos        -5.501e-04  5.445e-05 -10.102  < 2e-16 ***
DistCoast   -3.149e-01  6.937e-02  -4.539 5.68e-06 ***
Depth       -4.548e-01  4.077e-02 -11.154  < 2e-16 ***
FMonth2      5.253e-01  5.363e-01   0.979  0.32737
FMonth3      3.153e+00  4.624e-01   6.819 9.34e-12 ***
FMonth4      6.542e-01  4.587e-01   1.426  0.15379
PhaseB       1.041e+02  3.253e+02   0.320  0.74901
PhaseC      -2.237e+02  4.045e+02  -0.553  0.58025
XPos:PhaseB  7.107e-05  2.572e-05   2.764  0.00572 **
XPos:PhaseC  6.310e-06  3.135e-05   0.201  0.84048
YPos:PhaseB -2.525e-05  5.329e-05  -0.474  0.63564
YPos:PhaseC  3.604e-05  6.646e-05   0.542  0.58770
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27.84 on 31488 degrees of freedom
Multiple R-squared:  0.01512,   Adjusted R-squared:  0.01472
F-statistic: 37.19 on 13 and 31488 DF,  p-value: < 2.2e-16
```

2.3 Parameter interpretation

Typically we wouldn't proceed with interpreting model output until we had assessed model assumptions and had confidence in this revised model.

- The parameter for each continuous covariate is defined as the change in the expected density for a unit increase in a given covariate.
- The parameter for each discrete covariate (month and phase) is defined as the change in expected density for a given month or phase compared with the baseline (month=1 and phase=A),

2.4 Parameter inference

So far, we have computed the parameter estimates for each coefficient (each β_j) but we know that each time we take a sample and find the estimates they are going to be slightly different (because the data going into the recipes/estimators will be different).

In order to be able to make general statements about the model parameters we need to be able to construct CIs and test hypotheses for these parameters.

The variance-covariance matrix of the parameter estimates can be obtained using the error variance and the design matrix:

$$Var(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$$

and the square-root of the diagonal of this $(p+1) \times (p+1)$ matrix returns the standard errors:

```
sqrt(diag(vcov(workingModel_Int)))
```

```
 (Intercept)         XPos         YPos    DistCoast        Depth
3.307893e+02 1.957593e-05 5.445387e-05 6.936881e-02 4.077389e-02
     FMonth2      FMonth3      FMonth4       PhaseB       PhaseC
5.362824e-01 4.624210e-01 4.586775e-01 3.253239e+02 4.045341e+02
 XPos:PhaseB  XPos:PhaseC  YPos:PhaseB  YPos:PhaseC
2.571680e-05 3.134823e-05 5.328567e-05 6.646321e-05
```

Building CIs for model parameters is very similar to building a CI for a mean estimate. We need the estimate, the standard error, a confidence level and we will use the t-distribution (with $df = N - p - 1$) to give us our multiplier.

2.4.1 Hypothesis testing

Hypothesis test results for each covariate are shown in the `t` value and `Pr(>|t|)` columns in the `R` output.

Specifically, the two-sided hypothesis test of **no relationship** for each covariate (ie. $H_0 : \beta_j = 0$, $H_1 : \beta_j \neq 0$) is performed in the familiar way:

$$\begin{aligned}\text{test statistic} =& \frac{\text{data-estimate - hypothesised value}}{\text{standard error}} \\ =& \frac{\hat{\beta}_j - 0}{SE(\hat{\beta}_j)}\end{aligned}$$

As an approximation, data-estimates that are more than about 2 standard errors from the hypothesized value ($\beta_j = 0$ in this case, no real underlying relationship) provide compelling evidence against H_0.}

An 8 minute clip about hypothesis testing for multiple linear regression coefficients can be found here

So, based on the p-values in the `workingModel_Int` output, which variables should be retained in this model?

It is difficult to know. Factor variables have multiple coefficients so we are interested in assessing a group of coefficients simultaneously.

For example, trying to choose between models with and without `phase` means comparing models that differ by 2 parameters. Here, we'll compare a reduced model (without any phase parameters), and the full model (with the phase coefficients)

We can formally test the hypothesis that the reduced model (with q parameters) is as good as the full model (with p parameters) and hence the reduced model is preferred.

If H_0 is true, and a reduced model is as good as the full model, the F-statistic will be small:

$$F = \frac{(ESS_{ReducedModel} - ESS_{FullModel})/(p-q)}{ESS_{FullModel}/(N-p-1)} \sim F_{(p-q,N-p-1)}$$

This procedure to evaluate the F-test is also called the **Analysis of Variance**

(ANOVA). If we fit a reduced model and compare it with a full model, then R can do the calculations for each covariate:

```
library(car)
Anova(workingModel_Int)
```

```
Anova Table (Type II tests)

Response: Count/Area
             Sum Sq    Df  F value    Pr(>F)
XPos          74064     1  95.5494 < 2.2e-16 ***
YPos         120982     1 156.0783 < 2.2e-16 ***
DistCoast     15969     1  20.6017 5.675e-06 ***
Depth         96433     1 124.4073 < 2.2e-16 ***
FMonth        49766     3  21.4011 7.666e-14 ***
Phase          9425     2   6.0797  0.002291 **
XPos:Phase     7029     2   4.5340  0.010745 *
YPos:Phase      717     2   0.4624  0.629763
Residuals  24407454 31488
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A 7 minute clip about the F-test can be found online here

2.5 Model selection

Now that we have multiple covariates and we know that the coefficients and associated p-values for each depend, to some extent, on what else is in the model, how do we decide which variables to include?

We want to include variables that:

1. have a genuine relationship with the response
2. offer a sufficient amount of new information about the response (after considering those variables already in the model)

We want to exclude variables that offer essentially the same information about response; i.e., we want to avoid **collinearity**.

A 5 minute clip about this issue can be found here

2.5.1 Automated variable selection

We want to have a good set of covariates in our model.

If we include too few variables in a model we throw away valuable information.

If we include both essential and non-essential variables in a model, the standard errors, confidence intervals and p-values tend to be too large.

Methods of variable selection:

- backward elimination (e.g. stepwise selection using the `step` function or p-values)
- forwards selection
- all possible subsets (e.g. using the `dredge` function).

Options for assessing "best" fit for each of the methods above:

- Information criteria (e.g. AIC, BIC, etc.),
- p-values (Wald tests or F-tests),
- other criteria such as cross-validation

Using the F-test results, if we remove the Y-phase interaction from the model then all terms are now significant in the model.

```
workingModel_Int<- update(workingModel_Int, .~. -YPos:Phase)
Anova(workingModel_Int)
```

```
Anova Table (Type II tests)

Response: Count/Area
             Sum Sq    Df  F value    Pr(>F)
XPos          74801     1  96.5043 < 2.2e-16 ***
YPos         120982     1 156.0837 < 2.2e-16 ***
DistCoast     16217     1  20.9219 4.802e-06 ***
Depth         96169     1 124.0715 < 2.2e-16 ***
FMonth        49741     3  21.3908 7.783e-14 ***
Phase          9425     2   6.0799  0.002291 **
XPos:Phase     7630     2   4.9217  0.007292 **
Residuals  24408170 31490
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.6 Checking model assumptions

Before we interpret model coefficients and/or make predictions based on this model, we should assess if the assumptions on which the model is based are reasonable.

In setting up the model, we have assumed that the relationship between each covariate and the response is linear, but we have also assumed the errors are Normally distributed, independent (i.e., uncorrelated with each other) and have constant variance.

If all model assumptions are satisfied, the residuals should behave approximately like a random sample from a Normal distribution centered at 0. If some of the assumptions are violated we should see a systematic pattern in the residuals.

2.6.1 Assessing Linearity

To check that linearity for each term in a working model (with or without interactions) is appropriate, it is best to produce **partial residual plots**. Let's first consider these without interactions.

2.6.2 Partial residual plots

Recall that we want to include variables that improve model fit. This means including variables with strong relationships between x and y which offer new information about y after considering those variables already in the model.

We can view the contribution of each covariate to the model using partial residual plots.

The partial residuals (for the p covariate/predictor) are found by adding the estimated relationship (for the p-th predictor; $\hat{\beta}_p x_{pit}$) to the residuals for the working model (r_{it}):

$$r_{pit} = r_{it} + \hat{\beta}_p x_{pit}$$

and when the x-variable (x_{pit}) is plotted with the partial residuals (r_{pit}) we have a **partial residual plot**; they have several useful properties:

1. The slope of the line is the regression coefficient
2. The amount of scatter about the line reflects how important x_p is as a predictor: large scatter=less important

3. Large residuals can be identified
4. Curved plots signal non-linear relationships

```
workingModel<-update(workingModel_Int, .~. - XPos:Phase)
par(mfrow=c(3,2))
termplot(workingModel,se=T)
```

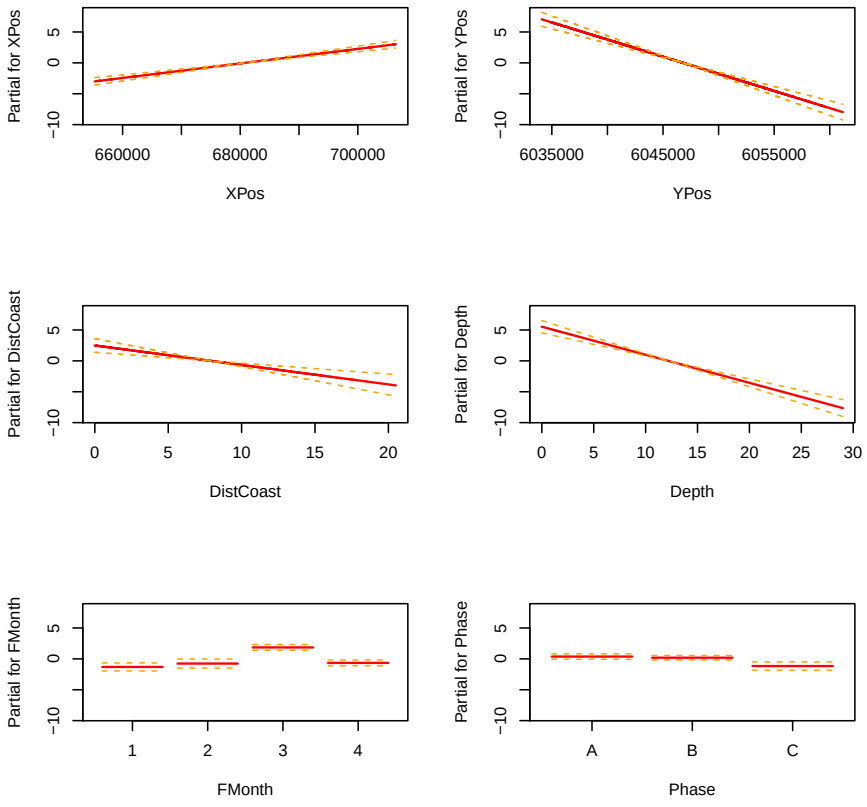


FIGURE 2.1 Partial residual plots for the 'workingModel' model without partial residuals.

```
par(mfrow=c(3,2))
termplot(workingModel,se=T,partial.resid=T)
```

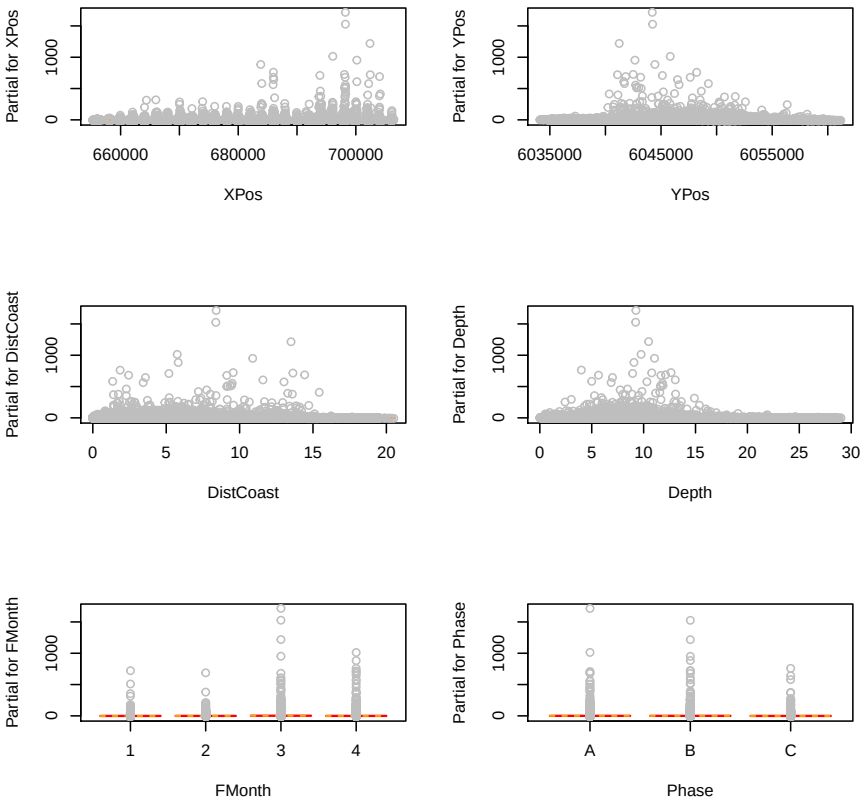


FIGURE 2.2 Partial residual plots for the 'workingModel' model with partial residuals.

Partial residual plots (for models without interactions) are easily obtained in `R` using the `termplot` function `workingModel` is a fit without interaction terms).

To view partial relationships when interaction terms are present, we need to use the `effects` library.

This can be used to generate the partial plots for either the terms separately (e.g. for the `Y`-coordinate relationship, see below) or the interactions (e.g. for `XPos:Phase`, see below):

```
library(effects)
plot(effect(c("XPos:Phase"), workingModel_Int, ylab="XPos"))
```

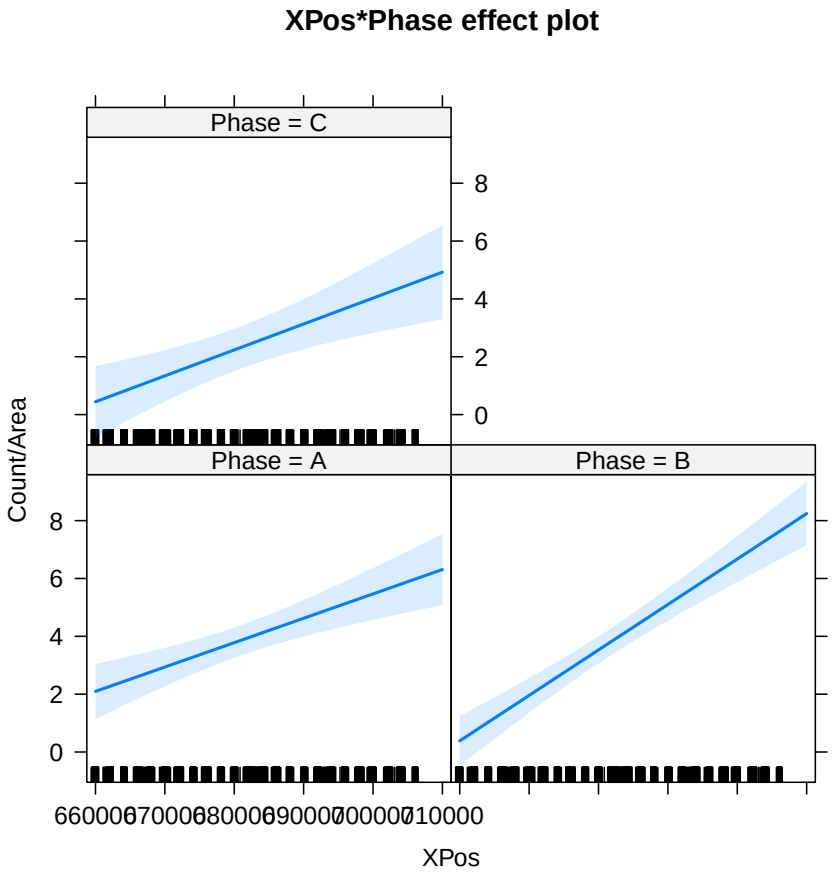


FIGURE 2.3 Partial plots for the interaction term in the interaction working model.

In this case, it is hard to determine if linearity is reasonable for the continuous covariates due to the size of the partial residuals (Figure 2.2).

2.6.2.1 Influential points

These partial plots may also show us:

Outliers: observations that are not well fitted by the model. If a standardized residual is **greater** than 2.5, this observation deserves further attention.

Influential observations: are observations which are very influential in the

fitting process (ie. if removed, results change substantially). Often this happens if an observation is an outlier or if it is well separated in terms of values of regressors (outlying x-values).

We could use something like Cooks distance to measure influence.

The Cook's distance can be obtained for each residual and plotted using:

```
plot(workingModel_Int, which=4)
```

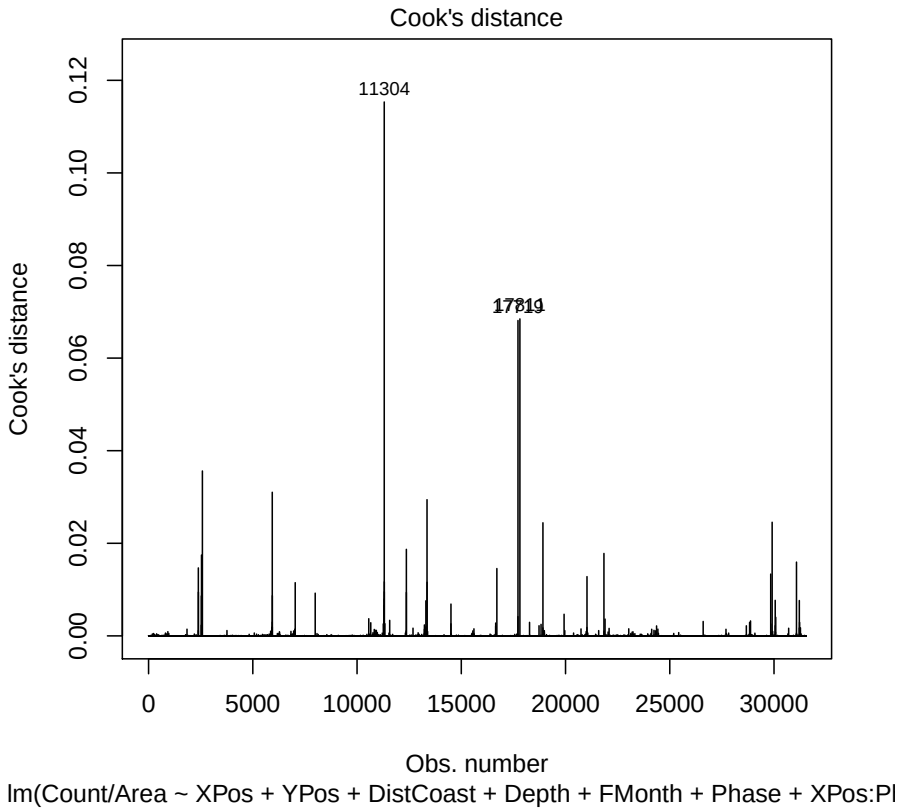


FIGURE 2.4 Cooks distance values for the interaction model.

2.6.3 Assessing constant variance

Constant error variance can be checked visually using residual plots.

The residuals should exhibit roughly equal spread across the range of the fitted values if constant error variance holds and thus, changes in the spread of residuals across the fitted value range indicate this assumption is violated.

We can also formally test for non-constant error variance using the **Breusch-Pagan** test (H_0: constant error variance).

The idea behind the test is that if we have constant error variance then the variation in the residuals (the squared residuals from our working model, r_{it}^2) should be unrelated to any of the covariates. For more, this online tutorial is excellent.

The variance of the residuals appears to increase with the fitted values Figure 2.5).

The Breusch-Pagan test also suggests strong evidence of non-constant error variance (see below).

```
ggplot(workingModel_Int, aes(.fitted, .stdresid)) + geom_point()
```

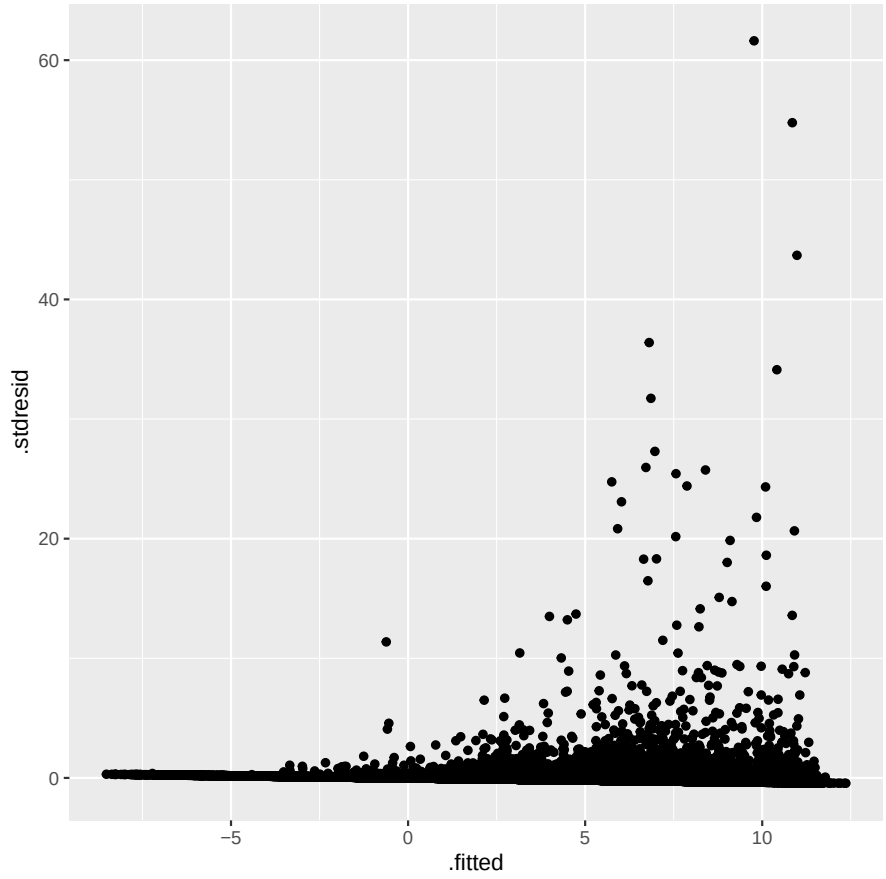


FIGURE 2.5 Fitted values vs the residuals for the interaction based model.

```
ncvTest(workingModel_Int)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 30794.31, Df = 1, p = < 2.22e-16
```

In response to this issue, the response variable could be transformed and rather than arbitrarily choose a transformation (such as `log` or `sqrt`) for the response, we can use `R` to estimate a transformation needed to return constant error variance.

This can be done in `R` using the `spreadLevelPlot` function in the `car` library:

```
sp<-spreadLevelPlot(workingModel_Int)
```

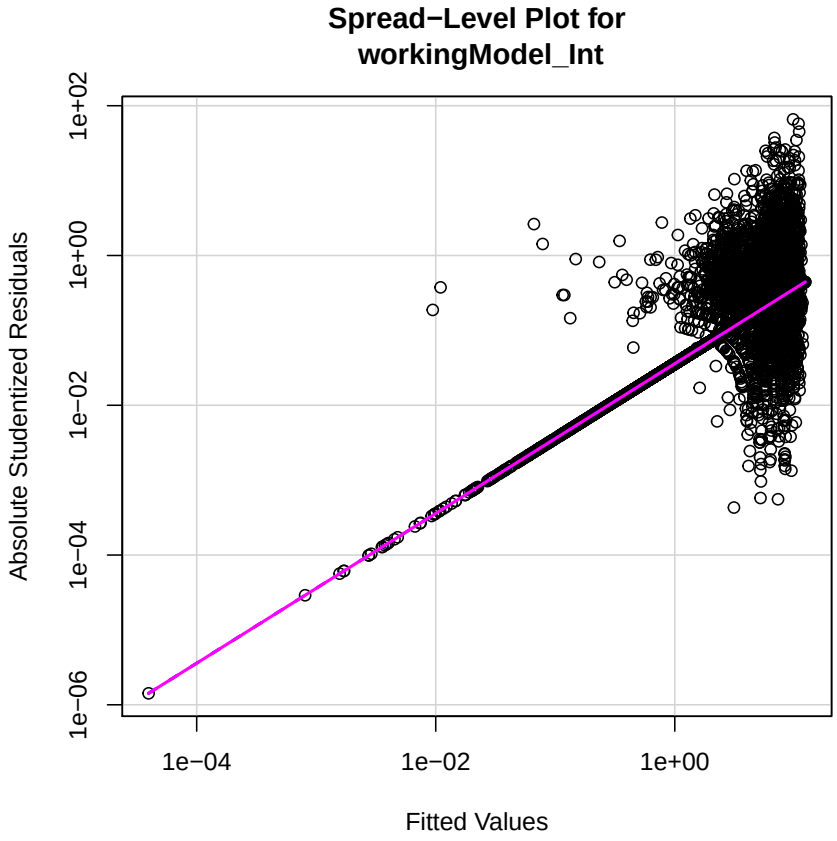


FIGURE 2.6 Spread level plot results for the interaction based model.

In this case the suggested power transformation is 1.6×10^{-5} and so $(count/area)^{1.6\times10^{-5}}$ is the suggested response.

The suggested transformation looks to have slightly stabilised the variance (Figure 2.6), though it is now difficult to interpret the associated model coefficients.

Looking for constant error variance in detail

We can investigate the way the spread of the residuals varies with the fitted values in more detail by plotting the fitted values vs the variance of the residuals (Figure 2.7):

```
cut.fit<- cut(fitted(workingModel_Int),
              breaks=quantile(fitted(workingModel_Int),
                              probs=c(seq(0,1,length=20))))
table(cut.fit)
```

```
cut.fit
 (-8.52,-2.73]  (-2.73,-1.17] (-1.17,0.0749]  (0.0749,1.06]     (1.06,1.7]
          1657           1658           1658           1658           1658
    (1.7,2.18]    (2.18,2.61]    (2.61,2.99]    (2.99,3.43]    (3.43,3.89]
          1658           1658           1658           1658           1658
   (3.89,4.35]     (4.35,4.8]     (4.8,5.25]    (5.25,5.72]    (5.72,6.24]
          1658           1658           1658           1658           1658
   (6.24,6.89]    (6.89,7.65]    (7.65,8.68]    (8.68,12.4]
          1658           1658           1658           1658
```

A model which satisfies the constant error variance assumption would show no mean-variance relationship and produce a patternless plot much like the plot based on the estimated transformation. The variance still appears not to be constant however, the size of the variance is much reduced compared with the raw response.

```
means1<- tapply(fitted(workingModel_Int),cut.fit,mean)
vars1<- tapply(residuals(workingModel_Int),cut.fit,var)
plot(means1,vars1, xlab="Fitted Values",
     ylab="Variance of the residuals",main="Raw response",
     pch=16)
abline(h=summary(workingModel_Int)$sigma**2,lwd=2)
```

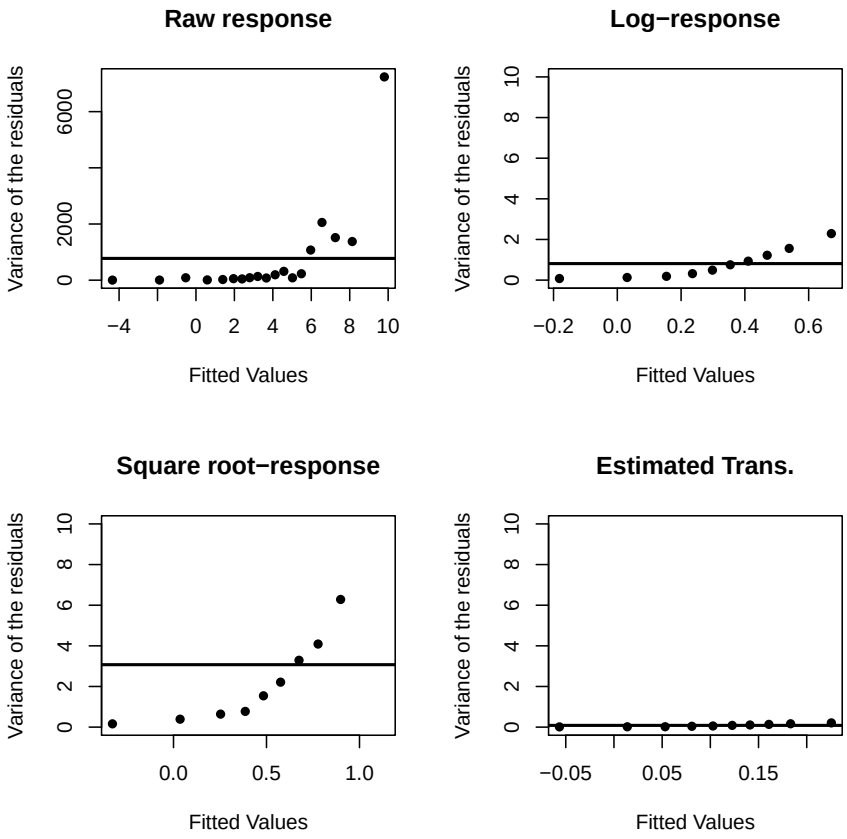


FIGURE 2.7 Fitted values vs variance of the residuals for the range of models. The variance assumed under each model is represented by the solid line. Note that the scales for the three transfom plots are all the same but different to the raw plot.

An important thing to note is that the model(s) fitted to date produce negative fitted values, while the input data (counts/area) are never negative.

This is a common problem when fitted normal-errors based models to continuous data that is bounded by zero. We will address this problem in later sections and restrict the fitted values to be non-negative.

2.6.4 Assessing Independence:

When the errors are independent the residuals should resemble a random scatter of points about the horizontal axis. Clusters of successive positive or negative residuals suggest serial correlation (a relationship between successive residuals).

```
sres200<-rstandard(workingModel_Int)[1:200]
dr<-data.frame(1:200,sres200)
colnames(dr)<-c("Index","StRes")

ggplot(dr, aes(x=Index, y=StRes)) +
  geom_line() +
  geom_point() +
  theme_bw() +
  geom_hline(aes(yintercept=0))+
  xlab("Observation Order") + ylab("Standardised Residuals")
```

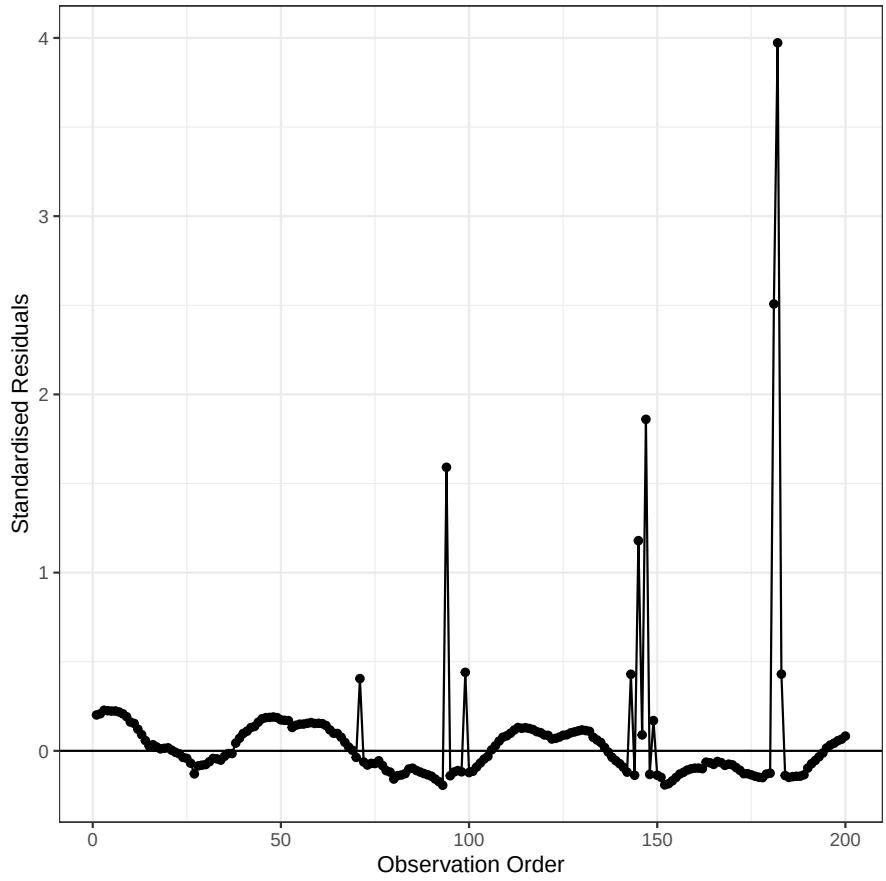


FIGURE 2.8 Standardized residuals for the preliminary model in observation order. These are the first 200 residuals only.

In this case, a pattern is evident (Figure 2.8); there is clear oscillation between negative and positive residuals suggesting some positive temporal correlation.

This is not surprising when you consider we have density data collected along transects over time.

Independence is also a **critical** model assumption and violation of this assumption can lead to unrealistic standard errors and misleading significance tests.

In short, positively correlated data:

- vary less than independent data (data along transects is typically more similar than data from different transects) and

- offer less information than independent pieces of data (i.e., our effective sample size is less than the apparent sample size)}.

These features combined lead us to underestimate the error variance (compared with independent data) and assume we have a sample size which is larger than it is. This can lead us to falsely conclude that one or more irrelevant variables are important.

Correlated errors in the model also means that the least-squares method of obtaining the parameter estimates is no longer ideal – that is they don't always return parameter estimates which are closest to the true parameter value. There are better estimators for data of this type, which get closer to the parameter more of the time.

A 6-minute clip about this and some causes of autocorrelation can be found here

In these cases, methods which do not require independent errors can be used instead; a generalized least squares (GLS) model is a linear model alternative which allows correlated errors (see Chapter 3).

For a short description see here

2.6.5 Normality

Normality can be visually checked using histograms and/or quantile-quantile (QQ) plots. The idea is the following: if we have two samples from the Normal distribution and we order these samples and plot these ordered samples should result in roughly a **straight line**.

In this case, we have the residuals which we can plot against a hypothetical sample (i.e., quantiles) from a N(0,1) distribution. If the normality assumption holds, we should obtain a straight line (with scatter). We could also plot a histogram of the residuals, to see whether we obtain something "Normal-looking" (e.g., it should be symmetric). We will return to this assumption after dealing with the constant error variance and independence assumptions using GLS models.

3

Generalized Least Squares

One mechanism for dealing with non-constant error variance is to replace the traditional *no mean-variance* assumption with a different mean-variance relationship.

Very often data exhibit a positive mean-variance relationship, (i.e. the residual variance increases with the fitted values) and this is what we have in this case (Figure 2.7).

An increasing mean-variance relationship can be modelled explicitly using **generalized least squares**, and if we select the type of this relationship it can be implemented using the `gls` function in the `nlme` library in `R`.

3.1 Model Specification

In this case, it seems reasonable to allow the variance to increase with the fitted values; there are (at least) two ways we can do this in `R`.

The power-related non-constant variance relationship (Equation (3.2)) and the exponential-based relationship (Equation (3.3)) are two ways to do this and each only require one additional parameter (m_1 or m_2) to be estimated under the model:

$$y_{it} = \beta_0 + \beta_1 x_{1it}, ..., \beta_p x_{pit} + N(0, \sigma^2) \tag{3.1}$$

$$y_{it} = \beta_0 + \beta_1 x_{1it}, ..., \beta_p x_{pit} + N(0, \sigma^2 |\hat{y}_{it}|^{2m_1}) \tag{3.2}$$

$$y_{it} = \beta_0 + \beta_1 x_{1it}, ..., \beta_p x_{pit} + N(0, \sigma^2 \exp(2m_2 \hat{y}_{it})) \tag{3.3}$$

We will consider a GLS-based model for the square-root transformed response, since if we "undo" this transformation after model fitting and square the fitted values (to obtain predictions on the response scale), these density predictions can never be negative:

```
df$FMonth<-as.factor(df$Month)

sqrtModel<-lm(sqrt(Count/Area) ~ XPos + YPos + DistCoast + Depth + FMonth +
                Phase+Phase:XPos, data = df)

summary(fitted(sqrtModel))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.9717  0.2589  0.5279  0.4889  0.7767  1.5594
```

```
summary(fitted(sqrtModel)^2)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.1027  0.2912  0.4101  0.6062  2.4317
```

We can use the `varPower` or `varExp` options in the `gls` function to implement Equations 3.2 and 3.3 respectively, e.g.:

```
require(nlme)
```

```
workingModel_GLS<- gls(sqrt(Count/Area) ~ XPos + YPos + DistCoast +
                         Depth + FMonth +Phase*XPos,
                       data = df, weights=varExp(), method="ML")
```

We are using ML-based estimation here (which we'll cover later) to ensure we can compare AIC scores across models. The attempt to fit the power-related variance function to this mean-variance relationship was unsuccessful, and this might be due to the almost zero variance values for the smallest fitted values (see Figure 2.7).

The exponential-based variance function (which could be fitted) is less than best – it only approximates the function until the fitted values reach 0.6; after this point the residual variance is estimated to be much larger than we see in the residuals (Figure 3.1)

```
# Mean-Variance relationship Figure
cut.fit<- cut(fitted(workingModel_GLS),
              breaks=quantile(fitted(workingModel_GLS),
                              probs=c(seq(0,1,length=100))))

means1<- tapply(fitted(workingModel_GLS),cut.fit,mean)
vars1<- tapply(residuals(workingModel_GLS),cut.fit,var)
```

```
fitted1<- (summary(workingModel_GLS)$sigma^2)*exp(2*coef(workingModel_GLS$model)*means1)
df1<-data.frame(means1,vars1,fitted1)
colnames(df1)=c("Means","Vars","Fitted")

ggplot(df1, aes(x=Means, y=Vars))+
   geom_point()+geom_line(aes(Means,Fitted),color="red") +
   xlab("Fitted Values") + ylab("Variance of the residuals")
```

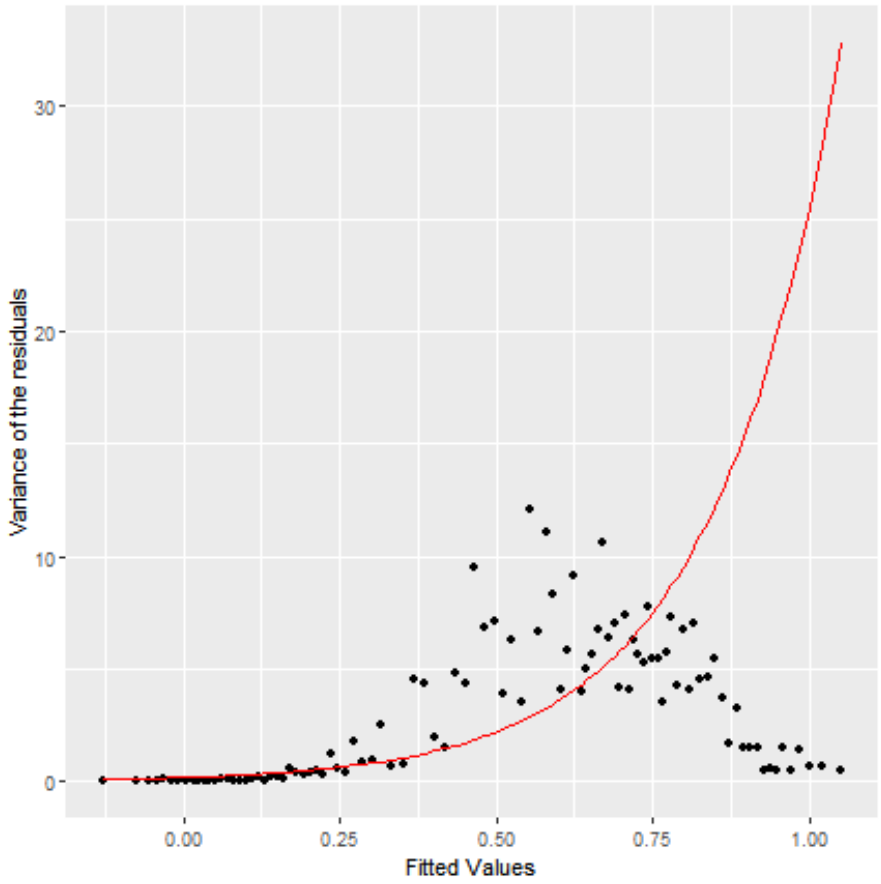


FIGURE 3.1 Fitted values vs variance of the residuals for the GLS model using the exponential-based function

Due to the extreme nature of this non-constant error variance, the GLS-based model conclusions (based on p-values) are quite different to the constant-variance

based conclusions. In particular, the Y-coordinate and distance from coast relationships are no longer statistically significant in the new model.

Further, the AIC favours the GLS-based model (1.11891×10^5 compared with 1.2477×10^5).

ANOVA on the initial fit

```
require(car)
Anova(sqrtModel)
```

```
Anova Table (Type II tests)

Response: sqrt(Count/Area)
           Sum Sq    Df F value    Pr(>F)
XPos          503     1 163.706 < 2.2e-16 ***
YPos         1592     1 518.360 < 2.2e-16 ***
DistCoast     242     1  78.935 < 2.2e-16 ***
Depth        1593     1 518.670 < 2.2e-16 ***
FMonth        781     3  84.790 < 2.2e-16 ***
Phase         124     2  20.170 1.762e-09 ***
XPos:Phase     99     2  16.133 9.937e-08 ***
Residuals   96741 31490
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(workingModel_GLS, type = "marginal")
```

```
Denom. DF: 31490
            numDF  F-value p-value
(Intercept)     1  13.3208  0.0003
XPos            1  59.1109  <.0001
YPos            1   9.8356  0.0017
DistCoast       1  17.2568  <.0001
Depth           1 763.8130  <.0001
FMonth          3  16.4270  <.0001
Phase           2  15.2653  <.0001
XPos:Phase      2  15.8289  <.0001
```

```
AIC(sqrtModel,workingModel_GLS)
```

```
                 df      AIC
sqrtModel        13 124769.7
workingModel_GLS 14 111891.0
```

```
summary(workingModel_GLS)
```

```
Generalized least squares fit by maximum likelihood
  Model: sqrt(Count/Area) ~ XPos + YPos + DistCoast + Depth + FMonth +      Phase * XPos
  Data: df
     AIC    BIC    logLik
  111891 112008 -55931.51

Variance function:
 Structure: Exponential of variance covariate
 Formula: ~fitted(.)
 Parameter estimates:
   expon
2.444357

Coefficients:
               Value Std.Error    t-value p-value
(Intercept) 29.408303  8.057597   3.649761  0.0003
XPos        -0.000004  0.000001  -7.688360  0.0000
YPos        -0.000004  0.000001  -3.136174  0.0017
DistCoast   -0.009329  0.002246  -4.154127  0.0000
Depth       -0.040233  0.001456 -27.637167  0.0000
FMonth2      0.022777  0.015671   1.453404  0.1461
FMonth3      0.067722  0.014282   4.741703  0.0000
FMonth4     -0.014766  0.013034  -1.132868  0.2573
PhaseB      -2.479799  0.461548  -5.372790  0.0000
PhaseC      -1.760770  0.509436  -3.456315  0.0005
XPos:PhaseB  0.000004  0.000001   5.472763  0.0000
XPos:PhaseC  0.000003  0.000001   3.500268  0.0005

 Correlation:
            (Intr) XPos   YPos   DstCst Depth  FMnth2 FMnth3 FMnth4 PhaseB
XPos         0.159
YPos        -0.999 -0.206
DistCoast   -0.633 -0.407  0.646
Depth       -0.039  0.525  0.012 -0.458
FMonth2      0.020  0.010 -0.021 -0.021  0.006
FMonth3      0.029 -0.015 -0.028 -0.010 -0.012  0.489
FMonth4      0.030 -0.017 -0.030 -0.010 -0.009  0.541  0.612
PhaseB       0.023  0.552 -0.049  0.005 -0.041  0.010 -0.006 -0.007
PhaseC      -0.001  0.551 -0.025 -0.023  0.068  0.000 -0.016 -0.021  0.426
XPos:PhaseB -0.023 -0.550  0.049 -0.005  0.041 -0.008  0.003  0.005 -1.000
XPos:PhaseC  0.002 -0.549  0.024  0.021 -0.069 -0.004  0.009  0.015 -0.425
            PhaseC XPs:PB
```

```
XPos
YPos
DistCoast
Depth
FMonth2
FMonth3
FMonth4
PhaseB
PhaseC
XPos:PhaseB -0.425
XPos:PhaseC -1.000  0.423

Standardized residuals:
       Min         Q1        Med         Q3        Max
-0.3428873 -0.3054422 -0.2482559 -0.1394397 27.3195667

Residual standard error: 0.4389241
Degrees of freedom: 31502 total; 31490 residual
```

3.2 Assessing residual independence

More here.

Despite including extra covariates in our model, the residuals still appear to be correlated when plotted in observation order (Figure 3.2).

```
par(mfrow=c(1,2))
plot(residuals(workingModel_GLS)[1:500], type="l", ylab="Residuals GLS")
acf(residuals(workingModel_GLS), main="ACF of the GLS Residuals")
```

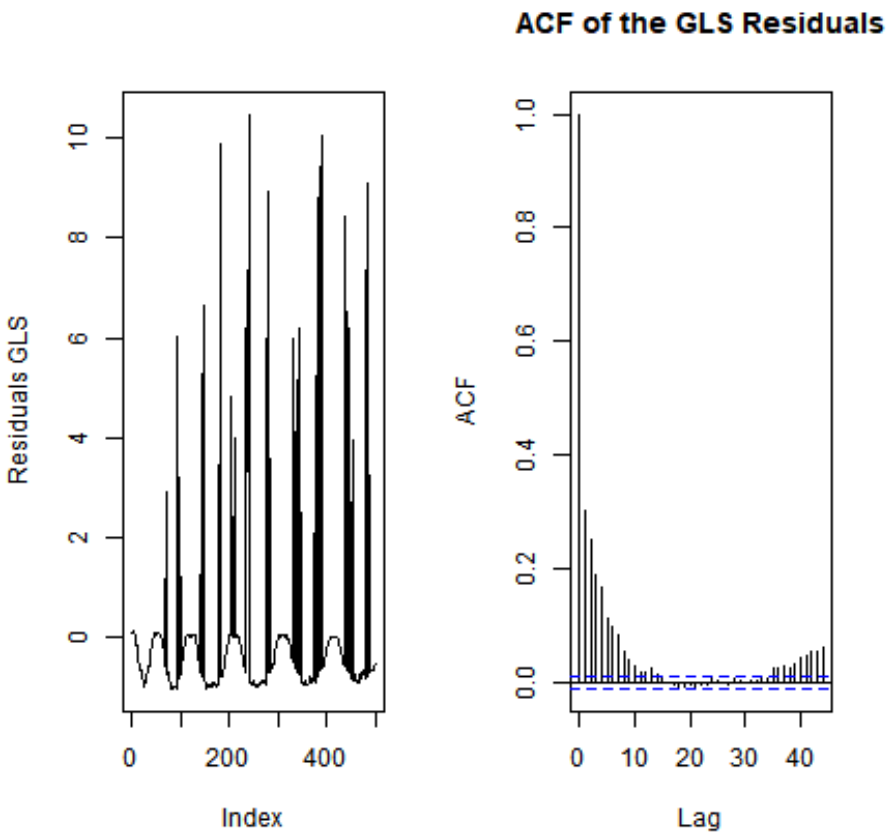


FIGURE 3.2 Residuals in observation order for the GLS model and empirical autocorrelation function (acf)

```
par(mfrow=c(1,1))
```

The Durbin-Watson test confirms this postive residual correlation:

```
durbinWatsonTest(sqrtModel)
```

```
 lag Autocorrelation D-W Statistic p-value
   1       0.2783551      1.443287       0
 Alternative hypothesis: rho != 0
```

However technically this result is approximate since this function only works on `lm`-based models (and we are currently using a GLS-based model).

The practical consequences of falsely assuming independence are unknown to us at this point; we could be concluding that one or more (unrelated) variables are genuinely related to the response.

We have 3 options in this case. We can

1. **ignore** the correlation in the residuals; easy but unwise: current model conclusions may be quite misleading
2. **remove** the correlation in model residuals by sub-setting the data (e.g. re-run analysis using every 20th observation). This is also easy, but a waste of information and may require many sub-setting attempts to remove the correlation in full}
3. **account** for the type of correlation seen in the residuals and fit a more appropriate model which does not assume independence.This takes extra time but data are not wasted and a defensible comparison with original results is available.

The current model ignores the correlation and why discard data? We'll replace the traditional "independent-errors" model with a model which assumes a correlation structure

3.3 Modelling residual correlation using GLS

A model for the correlation structure (i.e. across time or across spatial coordinates) must be chosen or a flexible (parameter hungry) structure used. The structure is typically chosen based on the sampling design. In this case, these data were collected along transects over time and so a decaying function of time might be reasonable.

Correlation structures can be chosen using autocorrelation functions (`acf`).

For a time series $(x_t)_{t=1,\ldots,T}$ the lag-l autocorrelation is:

$$\hat{\rho}_l = \frac{\sum_{t=1}^{T-l}(x_t - \bar{x})(x_{t+l} - \bar{x})}{\sum_{t=1}^{T}(x_t - \bar{x})^2}$$

$\hat{\rho}_l$ for the residuals can be calculated, and plotted, for many lags using the `acf` function (Figure 3.2). The `acf` function also displays 95% confidence intervals around 0 for comparison with the estimated correlation for each lag.

The Durbin-Watson statistic can also be obtained for a variety of lags and associated *p*-values obtained (with H_0 : correlation is 0).

```
durbinWatsonTest(sqrtModel, max.lag=15)
```

```
 lag Autocorrelation D-W Statistic p-value
   1     0.278355113      1.443287   0.000
   2     0.227947936      1.544098   0.000
   3     0.163430646      1.673129   0.000
   4     0.143367659      1.713251   0.000
   5     0.087554239      1.824875   0.000
   6     0.072331774      1.855315   0.000
   7     0.058963349      1.882048   0.000
   8     0.031767347      1.936436   0.000
   9     0.020515636      1.958936   0.002
  10     0.008455949      1.983052   0.150
  11     0.001098047      1.997765   0.802
  12     0.002715429      1.994529   0.596
  13     0.011285148      1.977389   0.068
  14     0.003336796      1.993285   0.542
  15    -0.002752112      2.005462   0.598
 Alternative hypothesis: rho[lag] != 0
```

There is a great deal of auto-correlation in the ordered model residuals pooled across transects. In particular, the correlation generally decays with time and is significantly non-zero until the observations are 10 measurements apart:

- If we consider our data as 'blocks' of 10 observations, we can permit the residuals within blocks to be correlated but assume observations between blocks are independent.
- This might be realistic for observations 10 measurements apart, but will be unreasonable for measurements which adjoin blocks.

Typically, the survey design will help us choose a blocking structure. In this case we have multiple observations within transects which were visited over time. For this reason, transect-day (indicated by `TransectID`) could be a sensible starting point for the blocking structure. Under this system we would be assuming that residuals associated with repeat visits to similar locations over time are uncorrelated. However this might well be reasonable – this is the marine environment and very often the reasons for unexplained patterns in model residuals (e.g. prey density) move day to day.

The AR(1) process is most commonly used to describe time-based within block correlation and the ACF of this process ($h(l, \rho)$) decays as the distance between measurements increases:

$$h(l, \rho) = \rho^l$$

l represents the lag between measurements and ρ is the correlation parameter (and $-1 \leq \rho \leq 1$) which is estimated.

```
x<-0:15
ar1<- 0.2705746
ACF<-ARMAacf(ar = c(ar1), ma = 0,
             lag.max = 15, pacf = FALSE)
dacf<-data.frame(x,ACF)
ggplot(dacf,aes(x=x,y=ACF))+geom_point(size=3) + geom_line() +xlab("Lag")
```

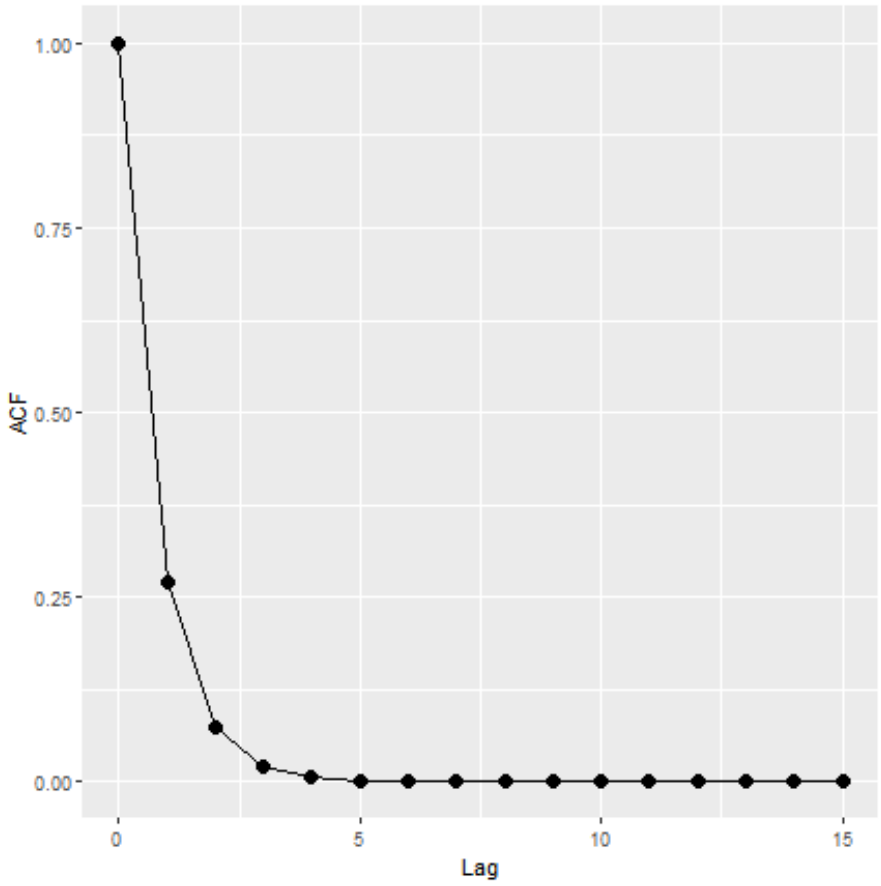


FIGURE 3.3 Decay in the correlation using an AR(1) error structure.

Here $\rho = 0.2705746$. Observations zero units apart are perfectly correlated ($\rho = 1$) and observations one measurement apart have a correlation of $\rho = 0.2705746$.

This decays to $\rho^2 = 0.2705746^2 = 0.0732106$ for measurements two units apart and so on.

AR(2) (Equation (3.4)) and AR(3) (Equation (3.5)) models are higher order correlation functions (with more parameters) that aren't as convenient to define:

$$h(l, \rho_1, \rho_2) = \rho_1 h(|l-1|, \rho_1, \rho_2) + \rho_2 h(|l-2|, \rho_1, \rho_2) \tag{3.4}$$

$$h(l, \rho_1, \rho_2, \rho_3) = \rho_1 h(|l-1|, \rho_1, \rho_2, \rho_3) + \rho_2 h(|l-2|, \rho_1, \rho_2, \rho_3) + -\rho_3 h(|l-3|, \rho_1, \rho_2, \rho_3) \tag{3.5}$$

$$\tag{3.6}$$

but are more flexible in their time-based decay.

We are going to compare AR(1), AR(2) and AR(3) auto-regressive correlation structures for the residuals within blocks using objective fit criteria (such as the AIC statistic).

```
# AR(1)
workingModel_GLScorr<- gls(sqrt(Count/Area) ~ YPos + DistCoast + Depth +
                             FMonth + Phase * XPos, data = df, weights=varExp(),
                           correlation=corAR1(form =~1|TransectID),method="ML")
summary(workingModel_GLScorr)
```

```
Generalized least squares fit by maximum likelihood
  Model: sqrt(Count/Area) ~ YPos + DistCoast + Depth + FMonth + Phase *      XPos
  Data: df
     AIC      BIC   logLik
  109317 109442.4 -54643.5

Correlation Structure: AR(1)
 Formula: ~1 | TransectID
 Parameter estimate(s):
      Phi
0.2707919
Variance function:
 Structure: Exponential of variance covariate
 Formula: ~fitted(.)
 Parameter estimates:
   expon
2.457128

Coefficients:
                Value Std.Error    t-value p-value
```

```
(Intercept) 30.013032 10.515721   2.854111  0.0043
YPos        -0.000004  0.000002  -2.471852  0.0134
DistCoast   -0.009782  0.002920  -3.350214  0.0008
Depth       -0.039188  0.001872 -20.936127  0.0000
FMonth2      0.020571  0.020551   1.001003  0.3168
FMonth3      0.066061  0.018750   3.523269  0.0004
FMonth4     -0.015546  0.017122  -0.907972  0.3639
PhaseB      -2.491163  0.605707  -4.112820  0.0000
PhaseC      -1.847877  0.670755  -2.754921  0.0059
XPos        -0.000004  0.000001  -5.675043  0.0000
PhaseB:XPos  0.000004  0.000001   4.187601  0.0000
PhaseC:XPos  0.000003  0.000001   2.788381  0.0053

 Correlation:
            (Intr) YPos   DstCst Depth  FMnth2 FMnth3 FMnth4 PhaseB PhaseC
YPos        -0.999
DistCoast   -0.640  0.653
Depth       -0.039  0.013 -0.453
FMonth2      0.020 -0.021 -0.021  0.006
FMonth3      0.028 -0.028 -0.010 -0.012  0.490
FMonth4      0.029 -0.029 -0.010 -0.010  0.543  0.613
PhaseB       0.023 -0.049  0.004 -0.041  0.010 -0.006 -0.007
PhaseC      -0.001 -0.026 -0.025  0.073  0.000 -0.016 -0.021  0.425
XPos         0.161 -0.208 -0.402  0.515  0.010 -0.015 -0.017  0.556  0.555
PhaseB:XPos -0.023  0.049 -0.004  0.041 -0.008  0.003  0.005 -1.000 -0.423
PhaseC:XPos  0.001  0.025  0.024 -0.074 -0.004  0.010  0.015 -0.423 -1.000
            XPos   PhB:XP
YPos
DistCoast
Depth
FMonth2
FMonth3
FMonth4
PhaseB
PhaseC
XPos
PhaseB:XPos -0.555
PhaseC:XPos -0.554  0.422

Standardized residuals:
       Min         Q1        Med         Q3        Max
-0.3386376 -0.3028303 -0.2476857 -0.1404629 27.4440013

Residual standard error: 0.4421224
Degrees of freedom: 31502 total; 31490 residual
```

```
# AR(2)
workingModel_GLScorr2<-update(workingModel_GLScorr,
                              corr = corARMA(p = 2, q = 0, form = ~ 1 | TransectID))
summary(workingModel_GLScorr2)
```

```
Generalized least squares fit by maximum likelihood
  Model: sqrt(Count/Area) ~ YPos + DistCoast + Depth + FMonth + Phase *      XPos
  Data: df
       AIC    BIC    logLik
  108630.2 108764 -54299.12

Correlation Structure: ARMA(2,0)
 Formula: ~1 | TransectID
 Parameter estimate(s):
     Phi1      Phi2
0.2270812 0.1596690
Variance function:
 Structure: Exponential of variance covariate
 Formula: ~fitted(.)
 Parameter estimates:
  expon
2.48656

Coefficients:
                Value Std.Error    t-value p-value
(Intercept) 29.591106 12.077812   2.450039  0.0143
YPos        -0.000004  0.000002  -2.121022  0.0339
DistCoast   -0.010206  0.003336  -3.059073  0.0022
Depth       -0.038051  0.002119 -17.956384  0.0000
FMonth2      0.018941  0.023793   0.796083  0.4260
FMonth3      0.063348  0.021691   2.920436  0.0035
FMonth4     -0.015135  0.019848  -0.762526  0.4458
PhaseB      -2.477127  0.700824  -3.534594  0.0004
PhaseC      -2.004761  0.782092  -2.563333  0.0104
XPos        -0.000004  0.000001  -4.883074  0.0000
PhaseB:XPos  0.000004  0.000001   3.596967  0.0003
PhaseC:XPos  0.000003  0.000001   2.594131  0.0095

 Correlation:
            (Intr) YPos   DstCst Depth  FMnth2 FMnth3 FMnth4 PhaseB PhaseC
YPos        -0.999
DistCoast   -0.649  0.662
Depth       -0.040  0.014 -0.442
FMonth2      0.021 -0.022 -0.021  0.005
```

```
FMonth3      0.028 -0.028 -0.010 -0.011  0.491
FMonth4      0.029 -0.029 -0.011 -0.010  0.542  0.613
PhaseB       0.023 -0.049  0.002 -0.040  0.011 -0.006 -0.007
PhaseC       0.001 -0.027 -0.027  0.079  0.000 -0.016 -0.021  0.423
XPos         0.162 -0.209 -0.393  0.504  0.010 -0.015 -0.018  0.563  0.559
PhaseB:XPos -0.023  0.049 -0.002  0.041 -0.009  0.003  0.005 -1.000 -0.422
PhaseC:XPos  0.000  0.027  0.026 -0.080 -0.004  0.010  0.015 -0.422 -1.000
            XPos   PhB:XP
YPos
DistCoast
Depth
FMonth2
FMonth3
FMonth4
PhaseB
PhaseC
XPos
PhaseB:XPos -0.561
PhaseC:XPos -0.557  0.421

Standardized residuals:
       Min         Q1        Med         Q3        Max
-0.3325404 -0.2977950 -0.2446092 -0.1394263 27.7086981

Residual standard error: 0.4448997
Degrees of freedom: 31502 total; 31490 residual
```

```
#AR(3)
workingModel_GLScorr3<-update(workingModel_GLScorr,
                              corr = corARMA(p = 3, q = 0, form = ~ 1 | TransectID))
summary(workingModel_GLScorr3)
```

```
Generalized least squares fit by maximum likelihood
  Model: sqrt(Count/Area) ~ YPos + DistCoast + Depth + FMonth + Phase *      XPos
  Data: df
       AIC      BIC    logLik
  108655.8 108797.8 -54310.88

Correlation Structure: ARMA(3,0)
 Formula: ~1 | TransectID
 Parameter estimate(s):
      Phi1       Phi2       Phi3
0.21731241 0.14564488 0.06111608
Variance function:
 Structure: Exponential of variance covariate
```

```
 Formula: ~fitted(.)
 Parameter estimates:
   expon
2.490052

Coefficients:
               Value Std.Error    t-value p-value
(Intercept) 29.543848 12.682588   2.329481  0.0198
YPos        -0.000004  0.000002  -2.014256  0.0440
DistCoast   -0.010281  0.003499  -2.938663  0.0033
Depth       -0.037667  0.002209 -17.049776  0.0000
FMonth2      0.018237  0.025151   0.725076  0.4684
FMonth3      0.061647  0.022905   2.691415  0.0071
FMonth4     -0.014511  0.021002  -0.690909  0.4896
PhaseB      -2.467865  0.739903  -3.335388  0.0009
PhaseC      -2.084628  0.829551  -2.512958  0.0120
XPos        -0.000004  0.000001  -4.685010  0.0000
PhaseB:XPos  0.000004  0.000001   3.393918  0.0007
PhaseC:XPos  0.000003  0.000001   2.543371  0.0110

 Correlation:
            (Intr) YPos   DstCst Depth  FMnth2 FMnth3 FMnth4 PhaseB PhaseC
YPos        -0.999
DistCoast   -0.654  0.666
Depth       -0.040  0.014 -0.437
FMonth2      0.021 -0.022 -0.021  0.005
FMonth3      0.028 -0.028 -0.010 -0.011  0.491
FMonth4      0.029 -0.029 -0.011 -0.010  0.542  0.613
PhaseB       0.022 -0.049  0.001 -0.040  0.011 -0.006 -0.007
PhaseC       0.001 -0.028 -0.028  0.082  0.001 -0.016 -0.021  0.423
XPos         0.164 -0.210 -0.389  0.499  0.010 -0.015 -0.018  0.567  0.560
PhaseB:XPos -0.022  0.049 -0.001  0.040 -0.009  0.003  0.005 -1.000 -0.421
PhaseC:XPos -0.001  0.027  0.027 -0.083 -0.004  0.010  0.015 -0.421 -1.000
            XPos   PhB:XP
YPos
DistCoast
Depth
FMonth2
FMonth3
FMonth4
PhaseB
PhaseC
XPos
PhaseB:XPos -0.565
PhaseC:XPos -0.559  0.420
```

```
Standardized residuals:
       Min         Q1        Med         Q3        Max
-0.3291386 -0.2950674 -0.2428559 -0.1370301 27.7884437

Residual standard error: 0.4488677
Degrees of freedom: 31502 total; 31490 residual
```

3.3.1 Model selection

```
AIC(workingModel_GLS,
    workingModel_GLScorr,
    workingModel_GLScorr2,
    workingModel_GLScorr3)
```

```
                     df      AIC
workingModel_GLS     14 111891.0
workingModel_GLScorr  15 109317.0
workingModel_GLScorr2 16 108630.2
workingModel_GLScorr3 17 108655.8
```

```
BIC(workingModel_GLS,
    workingModel_GLScorr,
    workingModel_GLScorr2,
    workingModel_GLScorr3)
```

```
                     df      BIC
workingModel_GLS     14 112008.0
workingModel_GLScorr  15 109442.4
workingModel_GLScorr2 16 108764.0
workingModel_GLScorr3 17 108797.8
```

Notice, the AR models add as many parameters as the order: AR(1) needs 1, AR(2) needs 2 etc. and the AR(2) model appears to fit the best based on the AIC (and BIC) scores.

```
Lag<-0:15
Empirical<-acf(residuals(sqrtModel),lag.max=15, plot=FALSE)[[1]]
AR1<-ARMAacf(ar = c(0.2707919), ma = 0,
             lag.max = 15, pacf = FALSE)
AR2<-ARMAacf(ar = c(0.2270812, 0.1596690),
```

```
                ma = 0, lag.max = 15, pacf = FALSE)
AR3<-ARMAacf(ar = c(0.21731241, 0.14564488, 0.06111608  ),
                ma = 0, lag.max = 15, pacf = FALSE)


dacf<-data.frame(Lag,Empirical,AR1,AR2,AR3 )

require(tidyr)
dacf<-dacf %>%  gather(key = "Series", value = "ACF",- Lag )
ggplot(dacf,aes(x=Lag,y=ACF)) +
  geom_point(aes(color=Series), size=3) +
  geom_line(aes(color=Series, linetype=Series))
```

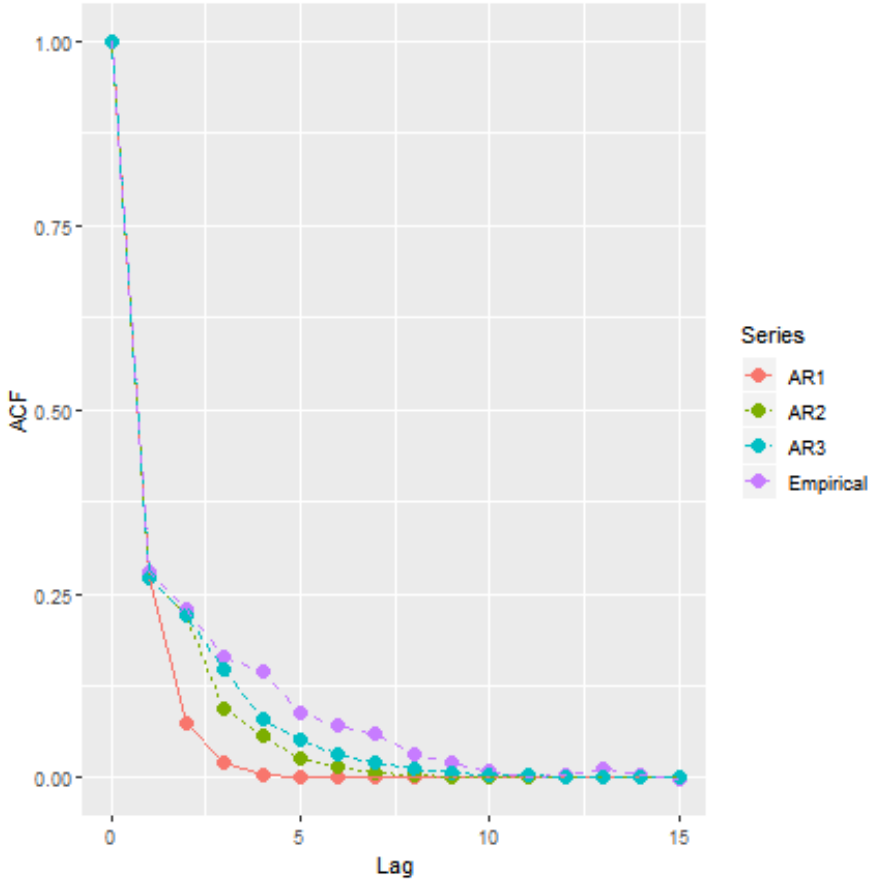


FIGURE 3.4 ACF for the sqrtModel residuals with different correlation structures overlaid

```
# No correlation
anova(workingModel_GLS, type="marginal")
```

```
Denom. DF: 31490
            numDF  F-value p-value
(Intercept)     1  13.3208  0.0003
XPos            1  59.1109  <.0001
YPos            1   9.8356  0.0017
DistCoast       1  17.2568  <.0001
Depth           1 763.8130  <.0001
FMonth          3  16.4270  <.0001
Phase           2  15.2653  <.0001
XPos:Phase      2  15.8289  <.0001
```

```
# AR(1)
anova(workingModel_GLScorr, type="marginal")
```

```
Denom. DF: 31490
            numDF  F-value p-value
(Intercept)     1   8.1459  0.0043
YPos            1   6.1101  0.0134
DistCoast       1  11.2239  0.0008
Depth           1 438.3214  <.0001
FMonth          3   9.2720  <.0001
Phase           2   9.0767  0.0001
XPos            1  32.2061  <.0001
Phase:XPos      2   9.4019  0.0001
```

```
# AR(2)
anova(workingModel_GLScorr2, type="marginal")
```

```
Denom. DF: 31490
            numDF  F-value p-value
(Intercept)     1   6.0027  0.0143
YPos            1   4.4987  0.0339
DistCoast       1   9.3579  0.0022
Depth           1 322.4317  <.0001
FMonth          3   6.3860  0.0003
Phase           2   6.9400  0.0010
XPos            1  23.8444  <.0001
Phase:XPos      2   7.1794  0.0008
```

```
# AR(3)
anova(workingModel_GLScorr3, type="marginal")
```

```
Denom. DF: 31490
            numDF   F-value p-value
(Intercept)     1   5.42648  0.0198
YPos            1   4.05723  0.0440
DistCoast       1   8.63574  0.0033
Depth           1 290.69487  <.0001
FMonth          3   5.39041  0.0010
Phase           2   6.30333  0.0018
XPos            1  21.94932  <.0001
Phase:XPos      2   6.51873  0.0015
```

```
# Confidence Intervals
intervals(workingModel_GLScorr2)
```

```
Approximate 95% confidence intervals

 Coefficients:
                     lower          est.         upper
(Intercept)  5.918119e+00  2.959111e+01  5.326409e+01
YPos        -8.212121e-06 -4.268029e-06 -3.239372e-07
DistCoast   -1.674558e-02 -1.020617e-02 -3.666774e-03
Depth       -4.220453e-02 -3.805104e-02 -3.389756e-02
FMonth2     -2.769420e-02  1.894133e-02  6.557687e-02
FMonth3      2.083229e-02  6.334818e-02  1.058641e-01
FMonth4     -5.403826e-02 -1.513484e-02  2.376859e-02
PhaseB      -3.850769e+00 -2.477127e+00 -1.103485e+00
PhaseC      -3.537691e+00 -2.004761e+00 -4.718309e-01
XPos        -5.801018e-06 -4.139461e-06 -2.477904e-06
PhaseB:XPos  1.689606e-06  3.712722e-06  5.735839e-06
PhaseC:XPos  7.305111e-07  2.988592e-06  5.246673e-06
attr(,"label")
[1] "Coefficients:"

 Correlation structure:
        lower      est.     upper
Phi1 0.2195213 0.2270812 0.2343305
Phi2 0.1494044 0.1596690 0.1698992
attr(,"label")
[1] "Correlation structure:"

 Variance function:
```

```
         lower    est.    upper
expon 2.446404 2.48656 2.526717
attr(,"label")
[1] "Variance function:"

 Residual standard error:
    lower      est.     upper
0.4357299 0.4448997 0.4542625
```

In this case, there don't appear to be any major practical consequence of acknowledging non-independence in the residuals. The p-values don't change in any substantial way across models but we could not have known this without carrying out this work.

We can measure the adequacy of a correlation method using normalised residuals. These are the raw residuals adjusted for the variance covariance estimated to be present within the errors for the blocks/panels/subjects (i.e. transect-days). If the correlation and variance structures used in the model are correct, the normalised residuals should be independent and approximately Normally distributed with mean zero and constant variance. In this case, we can see that the correlation-based models substantially reduce the residual auto-correlation (Figure 3.5):

```
# Compare ACF plots
par(mfrow=c(2,2))
acf(residuals(workingModel_GLS), main="Residual Indep.")
acf(residuals(workingModel_GLScorr, type="normalized"),
    main="Residuals for AR(1) model")
acf(residuals(workingModel_GLScorr2, type="normalized"),
    main="Residuals for AR(2) model")
acf(residuals(workingModel_GLScorr3, type="normalized"),
    main="Residuals for AR(3) model")
```

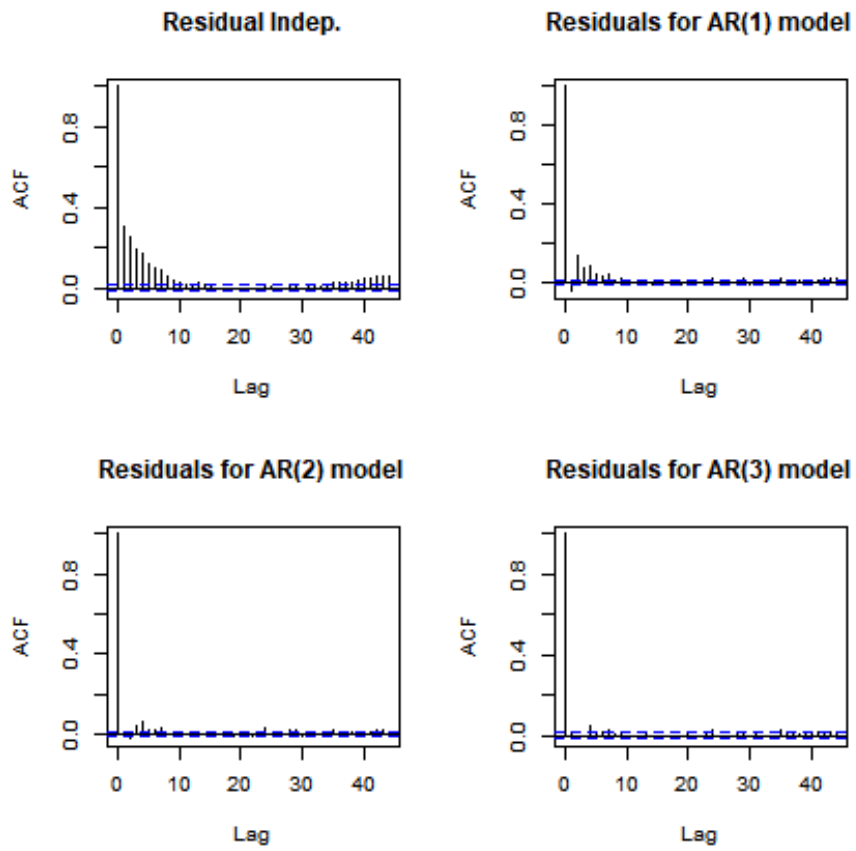


FIGURE 3.5 Normalised residuals for the independence and correlation-based models

3.4 Assessing normality

The normalised residuals appear to be right skewed compared with what we would expect from a Normal distribution (Figure 3.6) and could be of some practical concern.

```
hist(residuals(workingModel_GLScorr2, type="normalized"), main='')
```

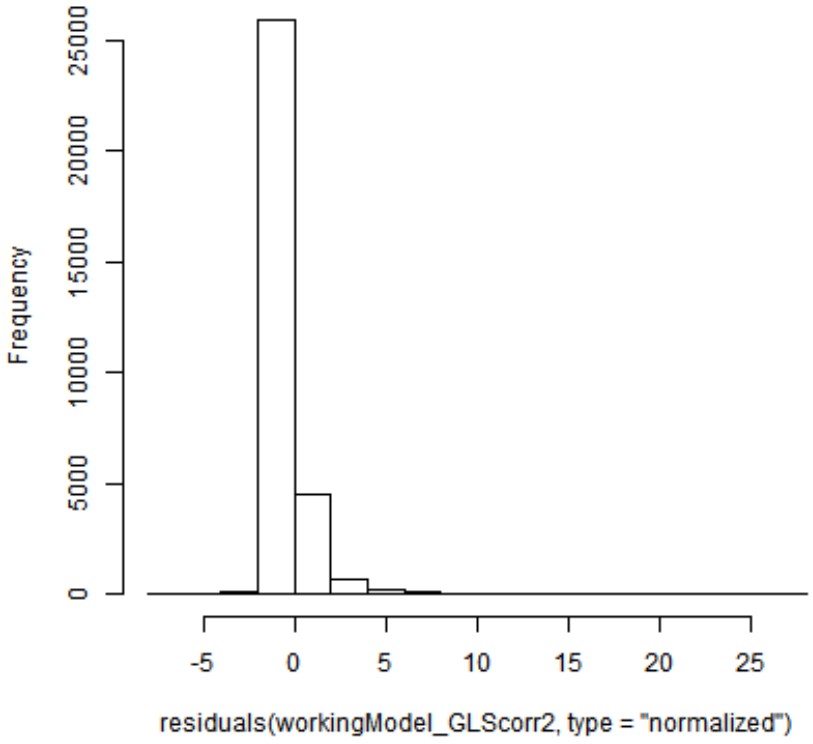


FIGURE 3.6 Normalised residuals for the AR(2)-based model

3.5 Conclusions to date

We need to do more to be able to answer our research questions - while there is little/no evidence of any density changes across phase this model was both a poor fit to the data and gave impossible predictions (negative values). For this reason we continued to fit models to a square root transformed response – in this way model predictions were guaranteed to be positive by back transforming (i.e. 'squaring') the fitted values. While this model type ensures the back transformed predictions are non-negative, working with a transformed response

rendered parameter interpretation difficult/impossible. Based on models fitted to the transformed response, the following was noted:

- There was evidence for substantial non-constant error variance and this proved difficult to approximate using GLS-based models (and the available variance structures).

- There was also significant levels of temporal correlation evident in model residuals and of those available/tried, the AR(2) model appears to approximate this best.
- Despite the transformed response, we still have some right-skewness in model residuals. Based on the final model to date, there does seem to be some redistribution of (square root) density in the X and Y co-ordinate direction. However, the evidence for the relationship with the Y co-ordinate is not compelling.
- There also seems to be genuine relationships for depth and month, with lower densities in deeper waters and differences in densities across months.

4

Modelling abundance using Count Models

In this chapter, we are going to revisit the research questions posed in the earlier sections using models designed explicitly for count (per unit area) data. We are going to start by assuming the data come from a Poisson distribution with mean λ. Data of this sort are bounded by zero and typically heavily right skewed. This distribution assumes that the mean is also the variance and thus as the mean increases so does the variance (there is a positive mean-variance relationship assumed under the model).

4.1 Maximum likelihood for Poisson data

4.1.1 Setting the scene:

We can estimate abundance (per unit area) across all transects and all time points (i.e. the pooled data) by assuming something about the nature of the process generating the data (in this case that the data come from a Poisson distribution) and estimating the mean (λ). We are going to start by assuming our pooled data come from a Poisson distribution, and that we have an independent set of discrete values with a constant average rate (λ). We are also going to start by assuming we have the same area associated with each count - we will include this information at the modelling stage.

4.1.2 Constructing the likelihood:

Recall that the Poisson distribution for one observation can be written as:

$$f(y_{it}|\lambda) = \frac{e^{-\lambda}\lambda^{y_{it}}}{y_{it}!}$$

If we assume that the collection of observations come from a distribution with the same mean, then we can consider all 31502 observations together using:

$$f(y_{(i=1,t=1)}, ..., y_{(i=731,t=43)}|\lambda) = \prod_{i=1}^{s}\prod_{t=1}^{n_i} \frac{e^{-\lambda}\lambda^{y_{it}}}{y_{it}!} = L(\lambda) \tag{4.1}$$

We can call this the ''Likelihood function'' ($L(\lambda)$) for these data. While this distribution might describe the data well, we don't know the mean of this distribution and instead need to estimate it.

A sensible way to approach this would be to try a bunch of values for the mean and see which one of these looks most like the mean that generated our data (from this Poisson distribution). We want to find the value that maximises the likelihood function for our data (Equation (4.1)), i.e. the maximum likelihood estimate of μ is the value of μ that maximises $L(\mu)$.

The likelihood function for a random sample of 100 from a Poisson distribution with $\lambda = 5$, evaluated for a range of values for the mean (λ) is shown in Figure 4.1.

```
set.seed(4)
y<- rpois(100,5)
# Likelihood Function
lik<- c()
for(lambda in seq(1,10, length=100)){
  val<-(exp(-lambda)*lambda**y)/factorial(y)
  lik<- rbind(lik,cbind(lambda, likelihood=prod(val)))

}
lik<- as.data.frame(lik)
```

One of the issues with the likelihood function is that a product of lots of very small numbers is numerically difficult/impossible to calclulate and so, we often work instead with the log of the likelihood function. That process turns products into sums. It turns out the value of μ that maximises the likelihood, $L(\mu)$, is the same value that maximises the log-likelihood function (ie. $l(\mu) = log[L(\mu)]$).

```
#Log Likelihood

loglik<- c()
for(lambda in seq(1,10, length=100)){
  val<-(exp(-lambda)*lambda**y)/factorial(y)
  loglik<- rbind(loglik,cbind(lambda, loglikelihood=sum(log(val))))
}
loglik<- as.data.frame(loglik)
```

Plotting the two functions:

```
par(mfrow=c(1,2))
plot(lik$lambda, lik$likelihood, type="b", main="Likelihood function")
plot(loglik$lambda, loglik$loglikelihood, type="b", main="Log-Likelihood function")
```

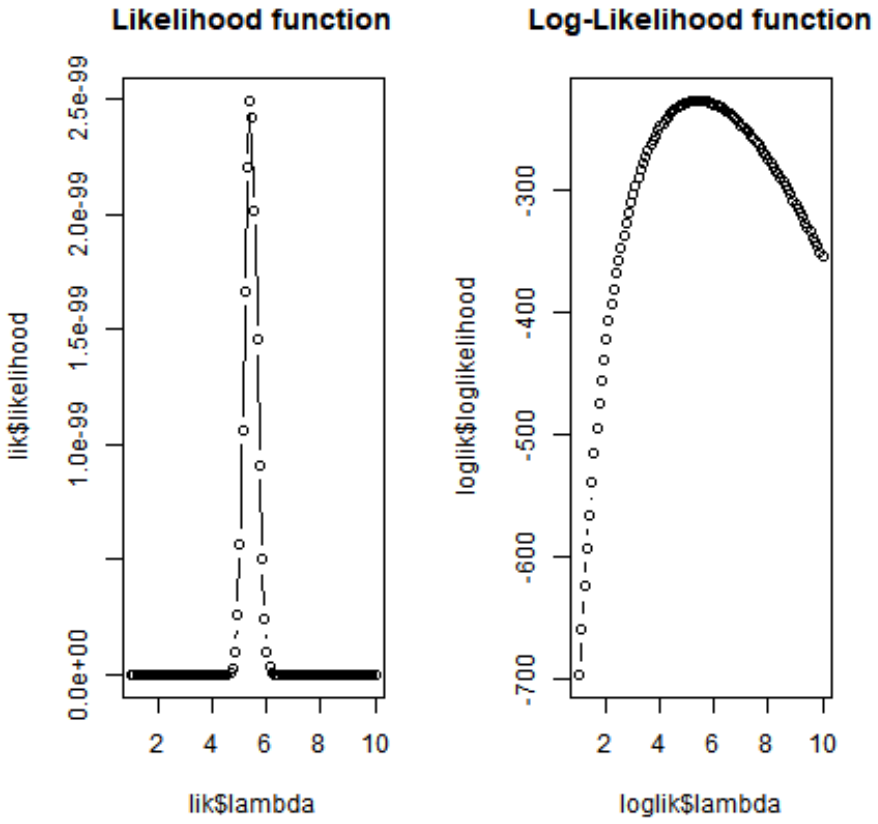


FIGURE 4.1 The likelihood function (left) and log-likelihood function (right) evaluated at a range of values for the mean for a random sample of 100 values sampled from a Poisson distribution with lambda=5

4.1.3 Maximising the log-likelihood

We can visually (and numerically determine) the value for λ that gives us the highest value for the likelihood and log likelihood function (e.g. Figure 4.1) for a sequence of candidate values, but there is a quicker way.

We can find the 'best' value for λ by evaluating the slope of the (log)likelihood

function for each candidate value and locating the value that returns a slope of zero. We can do this easily by differentiating the log likelihood function with respect to λ and solving for zero.

For the Poisson distribution, the recipe/estimator that results from this process is the following:

$$\hat{\lambda} = \frac{\sum_{i=1}^{s}\sum_{t=1}^{n_i} y_{it}}{31502}$$

Why are we so keen on maximum likelihood?

Ideally we would like the distribution of the estimator[1] to give values which are:

- concentrated around the true parameter value (θ); we want an estimator which is **unibiased** with a variance as **small** as possible.
- more precise as the number of observations, n, increases. As we gather more data (ie. as n gets larger) we should have more information about the -unknown parameter and the estimator become "better" as n increases.

Further, as $n \to \infty$ (ie. we have as many observations as we want) it should give θ exactly (with no error): this is the concept of a **consistent estimator**.

For the Poisson distribution the expected value of the estimator is λ and the standard error is $\sqrt{\frac{\hat{\lambda}}{n}}$ and so the estimator is both unbiased and consistent — attractive properties.

For the pooled data and for each of the three phases, we find:

```
mean(df$Count)
```

```
[1] 3.322392
```

```
df %>% group_by(Phase) %>% summarise(Mean=mean(Count))
```

```
# A tibble: 3 x 2
  Phase  Mean
  <fct> <dbl>
1 A      3.34
2 B      3.71
3 C      2.37
```

[1] (the recipe for the estimates of the Normal, Poisson and Binomial distributions)

These values appear to be similar across phases A and B but slightly lower in phase C. However since these are *estimates* it is very difficult to tell if the *underlying* means differ between years or any differences are due to sampling variation alone. To help us discriminate between real differences across time and sampling variability we need to build confidence intervals.

4.2 Confidence intervals (CIs) for Poisson data

Mean estimates for the Poisson distribution are Normally distributed about the true (and underlying) mean λ (in large samples) and so these estimates behave in a similar way to large samples of Normal data.

The distribution of these estimates can be written as:

$$\hat{\lambda} \sim Normal\left(\lambda, se(\hat{\lambda}) = \sqrt{\frac{\hat{\lambda}}{n}}\right)$$

This means that (for large samples) an estimate should lie within 2 standard errors of λ most of the time and so, building a CI which stretches about 2 standard errors either side of the estimate should contain λ.

A 95% CI can be found using:

$$\text{estimate} \quad \pm z_{0.025} \times \quad \text{standard error}$$

.

So, for the pooled data:

$$95\%\, CI = \hat{\lambda} \pm z_{0.025} \times se(\hat{\lambda}) = 3.322392 \pm 1.959964 \times 0.01026967 = (3.302264, 3.34252)$$

Interpret the interval: We can be 95% confident that the number of birds for the pooled data are somewhere between 3.30 and 3.34 for a randomly chosen site on average.

While this process can be useful it doesn't permit us to consider multiple covariates simultaneously and is based on assuming the mean is equal to the variance. This equivalence is a rather strict Poisson-based assumption which may be drastically unrealistic in practice.

5

GLMs for Poisson data

We are going to model counts (per unit area) using a statistical model. We could approach this using linear regression, but this is inappropriate for many reasons:

- the counts per unit area is not guaranteed to change linearly with model covariate (e.g. depth),
- the response (counts per unit area) is naturally bounded by zero and linear model predictions can give negative values.
- the errors are unlikely to be normal with constant variance.

While we can find ways to relax the linearity, normality and constant variance assumptions, predictions may still need truncating if linear models are used.

5.1 Generalized Linear Models (GLMs)

Generalized Linear Models (GLMs) are an extension of standard linear models in that they allow the response data (given a model) to follow any distribution inside the 'exponential family': this family of distributions includes the Normal, Poisson, Binomial and Gamma.

Y_{it} has a distribution within the exponential family if its probability density function can be written in the following way:

$$f(y_{it}|\theta_{it},\phi) = \exp\left\{\frac{y_{it}\theta_{it} - b(\theta_{it})}{a(\phi)} + c(y_{it},\phi)\right\}$$

where θ_{it} is the canonical (or natural) parameter, and ϕ is the scale parameter required to produce sensible standard errors for a distribution from the exponential family. The functions $a(.)$, $b(.)$ and $c(.)$ are specific for each distribution.

The exponential family provides a notation that allows us to model continuous, discrete, proportional, count and binary outcomes.

GLMs:

- allow the mean of the response to be a function of the **linear predictor** via a **link function**.
- can be appropriate for continous, discrete and proportional data.

Poisson-based GLMs will be used to model the counts per unit area across phases and with multiple covariates. We are going to model bird counts (per unit area) across phases using a GLM with a **log link** and **Poisson errors**.

5.2 Model specification

We are going to continue to model counts using a Poisson distribution but instead of estimating a single mean for the data pooled across transects and time points, we are going to allow λ_{it} to vary across phases using a Generalised Linear Model (GLM) with a log link.

This model can be written in terms of the response as:

$$E(y_{it}) := \lambda_{it} = e^{\eta_{it}} = e^{\beta_0+\beta_1 x_{1it}+...+\beta_p x_{pit}}$$

or on the scale of the log link function:

$$g(\lambda_{it}) = \log(\lambda_{it}) = \eta_{it} = \beta_0 + \beta_1 x_{1it} + ... + \beta_p x_{pit}$$

Note, the relationship between the response and the explanatory variables is non-linear while the relationship between the response on the link scale is linear.

The log link function is most commonly used for Poisson data, but the square-root link function is sometimes used instead.

This link function has:

$$\lambda_{it} = \eta_{it}^2 = (\beta_0 + \beta_1 x_{1it} + ... + \beta_p x_{pit})^2$$

or in a different way:

$$g(\lambda_{it}) = \sqrt{\lambda_{it}} = \eta_{it} = \beta_0 + \beta_1 x_{1it} + ... + \beta_p x_{pit}$$

For these data, modelling bird numbers at each spatial location may not be ideal since the area associated with each (corrected) count differs along the transects. Unless this uneven effort is recognised by the model some areas may look to be more popular with birds simply because more effort was spent in those areas. To avoid this occurring, the amount of search time (or sampling effort) can be

explicitly included in the model using an **offset** term. For these data, it is more sensible to model bird counts per unit effort (e.g. area associated with each spatial location) and this naturally results in some changes to the Poisson-based model.

For instance, the new linear predictor still has a log link but counts per unit area are being explicitly modelled instead (Equation (5.1)). Equation (5.1) can be re-arranged to give Equation (5.2) which can itself be re-organised to leave λ_i sitting on its own (Equations (5.3) and (5.4):

$$\log\left(\frac{\lambda_{it}}{\text{area}_{it}}\right) = \beta_0 + \beta_1 x_{1it} \tag{5.1}$$

$$\log(\lambda_{it}) - \log(\text{area}_{it}) = \beta_0 + \beta_1 x_{1it} \tag{5.2}$$

$$\log(\lambda_{it}) = \log(\text{area}_{it}) + \beta_0 + \beta_1 x_{1it} \tag{5.3}$$

$$\lambda_{it} = \text{area}_{it} \times e^{\beta_0 + \beta_1 x_{1it}} \tag{5.4}$$

These equations tell us we need to add the equivalent of `log(area)` as an *offset* term to the linear predictor (Equation (5.3)) when specifying this type of model in R. Note,the offset term does not have an associated coefficient.

5.3 Model Fitting

Poisson based GLMs can be fitted using ML and the log-likelihood function for a log link function (in the absence of an offset) is the following:

$$log(L) = \sum_{i=1}^{s} \sum_{t=1}^{n_i} [y_{it} \log(\lambda_{it}) - \lambda_{it} - \log \Gamma(y_{it} + 1)]$$

Note: under this model, the mean and variance are assumed to be the **same** and equal to λ; i.e. $\mu = V(\mu) = \lambda$.

GLMs fitted to bird counts using `Phase` with and without an offset and either the log or square root link function is as follows:

```
pois.phase_log<- glm(Count ~ Phase, data=df,family=poisson)
pois.phase_sqrt<- glm(Count ~ Phase, data=df,family=poisson(link="sqrt"))
pois.phase_log_offset<- glm(Count ~ Phase, data=df,family=poisson,
                            offset=log(Area))
```

5.3.1 Model Selection {mselpois}

The AIC and/or log-liklihood values can be used to discriminate between models. However, since we are modelling a different response when an offset is used (e.g. counts per unit area), AIC scores should not be compared between models fitted with and without an offset.

```
AIC(pois.phase_log)
```

```
[1] 678995.5
```

```
AIC(pois.phase_sqrt)
```

```
[1] 678995.5
```

```
AIC(pois.phase_log_offset)
```

```
[1] 676821.8
```

5.3.2 Fitted values

Fitted values (on the response scale) for a GLM fitted without an offset, Poisson errors and log link can be found using:

$$\hat{y}_{it} = e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1it} + ,...,+\hat{\beta}_p x_{pit}}$$

while the fitted values for an equivalent model fitted with an offset are:

$$\hat{y}_{it} = \text{area}_{it} \times e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1it} + ,...,+\hat{\beta}_p x_{pit}}$$

These fitted value types depending on the model fitted can be obtained using the `predict` function in `R` with `type="response"`.

5.3.3 Pearson Residuals

Assessing a GLM (with a mean variance relationship) can be carried out using **Pearson** residuals. Pearson residuals are the usual residuals divided by the estimated standard deviation ($\sqrt{\text{Variance function}}$) under the fitted model. This makes the magnitude of residuals comparable across observations and so should

exhibit no patterns when plotted against the fitted values. Pearson residuals for Poisson-based models can be defined as:

$$r_{it}^{p} = \frac{y_{it} - \hat{y}_{it}}{\sqrt{\hat{y}_{it}}}$$

since $\hat{y}_{it} = \hat{\lambda}_{it}$.

Pearson residuals can be obtained using `type="pearson"` while the raw residuals, $r_{it} = y_{it} - \hat{y}_{it}$, can be found using `type="response"` (see below). Pearson residuals should not show systematic patterns or trends in magnitude when plotted against fitted values or (candidate) explanatory variables. If they show some kind of pattern, then we have to question the validity of the mean-variance relationship assumed under the model.

5.4 Overdispersion

Overdispersion occurs in discrete response models when the variance of the response is greater than the variance specified by the model. Overdispersion causes the standard errors of the estimated coefficients to be underestimated and may provide unrealistic significance test results. For instance, a variable may appear to be a significant predictor when it is not. There are two general types of overdispersion: true and apparent. Apparent overdispersion may occur because important covariates (or interaction terms) are omitted from the model, there may be outliers in the data, a covariate may need to be transformed or the linear relationship assumed by the model is not reasonable.

If one or more of these causes are identified these can often be remedied and the fit of the model improved. However, sometimes the overdispersion is inherent in the data (true overdispersion) and we must use methods which address this true overdispersion. While it is helpful to find the source of the dispersion this is not always possible.

Poisson (and Binomial) GLMs should be checked for overdispersion and if present, the standard errors re-scaled. The dispersion parameter ϕ for the overdispersed Poisson model can be found using:

$$\widehat{\phi} = \frac{1}{N - p - 1} \sum_{i=1}^{s} \sum_{t=1}^{n_i} \frac{(y_{it} - \hat{\mu}_{it})^2}{\hat{\mu}_{it}}$$

Overdispersion should be estimated for Poisson models via ϕ; in this case the variance is then assumed to be proportional (rather than equal) to the mean:

$Var(Y) = \phi\lambda$. Estimating overdispersion results in re-scaled standard errors and the use of the t-distribution (rather than the z-distribution) when constructing confidence intervals and carrying out hypothesis tests. The standard errors for an overdispersed model ($se(\hat{\beta}_{1,\,\hat{\phi}})$) can be found using the standard errors from the original model ($se(\hat{\beta}_1)$) and the dispersion parameter estimate ($\hat{\phi}$):

$$se(\hat{\beta}_{1,\,\hat{\phi}}) = \sqrt{\hat{\phi}} \cdot se(\hat{\beta}_1)$$

The dispersion parameter can be estimated by specifying `family=quasipoisson` inside the `glm` function:

```
pois.phase_log_offsetOD<- glm(Count ~ Phase, data=df, family=quasipoisson, offset=log(Area
summary(pois.phase_log_offsetOD)
```

```
Call:
glm(formula = Count ~ Phase, family = quasipoisson, data = df,
    offset = log(Area))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
 -2.737   -2.737   -2.601   -2.187  130.852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.26326    0.07425  17.013   <2e-16 ***
PhaseB       0.10234    0.09774   1.047   0.2951
PhaseC      -0.34594    0.14379  -2.406   0.0161 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 211.4993)

    Null deviance: 664249  on 31501  degrees of freedom
Residual deviance: 661844  on 31499  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 8
```

In this case, we have massive overdispersion; the dispersion parameter is estimated to be more than 1 ($\hat{\phi} = 211.4993$). This has practical consequences for the standard errors in the model and the conclusions drawn.

```
summary(pois.phase_log_offset)
```

```
Call:
glm(formula = Count ~ Phase, family = poisson, data = df, offset = log(Area))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
 -2.737   -2.737   -2.601   -2.187  130.852

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.263262   0.005106  247.42   <2e-16 ***
PhaseB       0.102338   0.006721   15.23   <2e-16 ***
PhaseC      -0.345937   0.009887  -34.99   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 664249  on 31501  degrees of freedom
Residual deviance: 661844  on 31499  degrees of freedom
AIC: 676822

Number of Fisher Scoring iterations: 8
```

```
summary(pois.phase_log_offsetOD)
```

```
Call:
glm(formula = Count ~ Phase, family = quasipoisson, data = df,
    offset = log(Area))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
 -2.737   -2.737   -2.601   -2.187  130.852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.26326    0.07425  17.013   <2e-16 ***
PhaseB       0.10234    0.09774   1.047   0.2951
PhaseC      -0.34594    0.14379  -2.406   0.0161 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for quasipoisson family taken to be 211.4993)

    Null deviance: 664249  on 31501  degrees of freedom
Residual deviance: 661844  on 31499  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 8
```

We can confirm the estimate for the dispersion parameter:

```
(1/(31502-2-1))*sum(((df$Count)- fitted(pois.phase_log_offsetOD))**2/
                      fitted(pois.phase_log_offsetOD))
```

```
[1] 211.4965
```

and the adjusted standard error for `PhaseB` using the estimated dispersion parameter (211.50) and the unadjusted standard error:

```
sqrt(211.4993)*0.006721
```

```
[1] 0.09774361
```

Rather than assess this rather simple model, we will continue by including more covariates and assess the resulting model. We anticipate this unexplained variation (overdispersion) to decrease as we introduce more covariates into the model since valuable predictors will explain additional patterns in the data.

5.5 Parameter interpretation

The coefficients reflect a set of multiplicative effects. The fitted values for phases A, B and C (Equations (5.5), (5.6) and (5.7) respectively) help illustrate this:

$$\hat{y}_{it} = \exp(\hat{\eta}_{it}) = \exp\left(\hat{\beta}_0\right) = \exp\left(1.2633\right) = 3.537075 \tag{5.5}$$

$$\hat{y}_{it} = \exp(\hat{\eta}_{it}) = \exp\left(\hat{\beta}_0 + \hat{\beta}_1\right) = \exp\left(1.2633 + 0.1023\right) = 3.918073 \tag{5.6}$$

$$\hat{y}_{it} = \exp(\hat{\eta}_{it}) = \exp\left(\hat{\beta}_0 + \hat{\beta}_2\right) = \exp\left(1.2633 - 0.3459\right) = 2.502775 \tag{5.7}$$

The effect of being in phase B (rather than phase A) means we multiply the fitted

value in phase A by the exponentiated coefficient (owing to the log-link) for phase B: $3.537075 \times \exp(0.1023) = 3.918074$.

The effect of being in phase C (rather than phase A) means we multiply the fitted value in phase A by the exponentiated coefficient (owing to the log-link) for phase C: $3.537075 \times \exp(-0.3459) = 2.502775$

5.6 Parameter inference

Using ML we can say that the parameter estimates will be Normally distributed (for large samples) with mean equal to the true parameter value and some estimated variance, i.e. if n is not too small: $\hat{\beta}_j \sim N(\beta_j, \phi\widehat{V}_j)$ approximately. $\widehat{V}_j$ is an estimate of the variance and ϕ is the dispersion parameter. Since we are assuming large samples, and if we assume we have mean=λ and variance=λ we can use a z-multiplier (a value from the Normal distribution) to construct confidence intervals:

$$\text{estimate} \quad \pm z - \text{multiplier} \times \text{standard error}$$

.

If instead we estimate overdispersion (i.e. variance=$\phi\lambda$ we instead need to use a t-multiplier) to construct confidence intervals.

e.g. for the intercept parameter under the standard model:

```
exp(1.263262 + qnorm(0.025)*  0.005106)
```

```
[1] 3.501721
```

```
exp(1.263262 - qnorm(0.025)*  0.005106)
```

```
[1] 3.572514
```

and for the intercept parameter under the overdispersed model:

```
exp(1.263262 + qt(0.025, df=31499)*  0.07425)
```

```
[1] 3.057902
```

```
exp(1.263262 - qt(0.025, df=31499)*  0.07425)
```

```
[1] 4.091022
```

Testing for **no relationship** between each covariate and the response proceeds as before:

$$H_0 : \beta_j = 0, \quad H_1 : \beta_j \neq 0, \quad \text{test-statistic} = \frac{\text{estimate}}{\text{SE}}$$

If H_0 is true, the test statistic should look like it has come from a Normal distribution ($N(0, 1)$) and large test statistics give compelling evidence against H_0. The *p*-value is found in the standard way.

5.7 Identifying redistribution across phases

We are going to fit a Poisson-based GLM to these data using an offset term and many predictors. We will fit some models, estimate overdispersion, run model selection and make predictions. We will also interpret the model output, speculate about why we have obtained these results and discuss possible model improvements.

5.7.1 Model specification

```
fullModel<- glm(Count ~ XPos+YPos+DistCoast+Depth + FMonth + XPos*Phase+YPos*Phase, data=d

require(car)
Anova(fullModel)
```

```
Analysis of Deviance Table (Type II tests)

Response: Count
           LR Chisq Df Pr(>Chisq)
XPos          27680  1    < 2e-16 ***
YPos          39413  1    < 2e-16 ***
DistCoast         3  1    0.07925 .
Depth         61038  1    < 2e-16 ***
```

```
FMonth        12271  3    < 2e-16 ***
Phase          3283  2    < 2e-16 ***
XPos:Phase     2066  2    < 2e-16 ***
YPos:Phase     1457  2    < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is there any evidence of overdispersion?

```
fullModel_OD<- glm(Count ~ XPos+YPos+DistCoast+Depth + FMonth + XPos*Phase+YPos*Phase, dat
summary(fullModel_OD)
```

```
Call:
glm(formula = Count ~ XPos + YPos + DistCoast + Depth + FMonth +
    XPos * Phase + YPos * Phase, family = quasipoisson, data = df,
    offset = log(Area))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-10.807   -2.574   -1.610   -0.684  105.576

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.346e+03  9.891e+01  13.604  < 2e-16 ***
XPos         3.468e-05  5.533e-06   6.269 3.69e-10 ***
YPos        -2.258e-04  1.607e-05 -14.051  < 2e-16 ***
DistCoast   -2.241e-03  1.310e-02  -0.171 0.864147
Depth       -3.003e-01  1.353e-02 -22.200  < 2e-16 ***
FMonth2      1.348e-01  1.238e-01   1.089 0.276328
FMonth3      8.197e-01  9.697e-02   8.453  < 2e-16 ***
FMonth4      1.726e-01  1.054e-01   1.637 0.101538
PhaseB      -1.027e+02  9.977e+01  -1.029 0.303532
PhaseC      -5.455e+02  1.473e+02  -3.704 0.000213 ***
XPos:PhaseB  2.794e-05  7.482e-06   3.734 0.000188 ***
XPos:PhaseC  3.977e-05  1.111e-05   3.581 0.000343 ***
YPos:PhaseB  1.379e-05  1.600e-05   0.862 0.388640
YPos:PhaseC  8.560e-05  2.359e-05   3.629 0.000285 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 105.101)

    Null deviance: 664249  on 31501  degrees of freedom
Residual deviance: 526732  on 31488  degrees of freedom
```

```
AIC: NA

Number of Fisher Scoring iterations: 8
```

Is each covariate now important?

```
Anova(fullModel_OD, test="F")
```

```
Analysis of Deviance Table (Type II tests)

Response: Count
Error estimate based on Pearson residuals

            Sum Sq    Df  F value    Pr(>F)
XPos         27680     1 263.3675 < 2.2e-16 ***
YPos         39413     1 375.0104 < 2.2e-16 ***
DistCoast        3     1   0.0293 0.8640728
Depth        61038     1 580.7690 < 2.2e-16 ***
FMonth       12271     3  38.9196 < 2.2e-16 ***
Phase         3283     2  15.6191 1.660e-07 ***
XPos:Phase    2066     2   9.8283 5.407e-05 ***
YPos:Phase    1457     2   6.9337 0.0009759 ***
Residuals  3309366 31488
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What happens if we ignore the offset?

```
fullModel_OD_nooffset<- glm(Count ~ XPos+YPos+DistCoast+Depth + FMonth + XPos*Phase+YPos*P
summary(fullModel_OD_nooffset)
```

```
Call:
glm(formula = Count ~ XPos + YPos + DistCoast + Depth + FMonth +
    XPos * Phase + YPos * Phase, family = quasipoisson, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-10.755   -2.575   -1.618   -0.691  105.709

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.346e+03  9.848e+01  13.663  < 2e-16 ***
XPos         3.472e-05  5.551e-06   6.254 4.04e-10 ***
```

```
YPos         -2.258e-04  1.600e-05 -14.113  < 2e-16 ***
DistCoast    -2.066e-03  1.318e-02  -0.157 0.875481
Depth        -3.009e-01  1.360e-02 -22.124  < 2e-16 ***
FMonth2       1.351e-01  1.244e-01   1.086 0.277271
FMonth3       8.190e-01  9.740e-02   8.409  < 2e-16 ***
FMonth4       1.732e-01  1.058e-01   1.636 0.101812
PhaseB       -9.994e+01  9.901e+01  -1.009 0.312766
PhaseC       -5.348e+02  1.464e+02  -3.654 0.000259 ***
XPos:PhaseB   2.776e-05  7.505e-06   3.699 0.000217 ***
XPos:PhaseC   3.956e-05  1.113e-05   3.555 0.000379 ***
YPos:PhaseB   1.336e-05  1.587e-05   0.842 0.399802
YPos:PhaseC   8.386e-05  2.344e-05   3.577 0.000348 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 106.0454)

    Null deviance: 666428  on 31501  degrees of freedom
Residual deviance: 528463  on 31488  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 8
```

```
Anova(fullModel_OD_nooffset, test="F")
```

```
Analysis of Deviance Table (Type II tests)

Response: Count
Error estimate based on Pearson residuals

            Sum Sq    Df  F value    Pr(>F)
XPos         27694     1 261.1550 < 2.2e-16 ***
YPos         40125     1 378.3833 < 2.2e-16 ***
DistCoast        3     1   0.0246  0.875418
Depth        61083     1 576.0216 < 2.2e-16 ***
FMonth       12238     3  38.4678 < 2.2e-16 ***
Phase         3273     2  15.4305 2.004e-07 ***
XPos:Phase    2049     2   9.6621 6.384e-05 ***
YPos:Phase    1431     2   6.7451  0.001178 **
Residuals  3339099 31488
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.7.2 Model Selection

Attempt stepwise selection

```
try(step(fullModel_OD))
```

```
Error in step(fullModel_OD) :
  AIC is not defined for this model, so 'step' cannot proceed
```

We can use p-value based backwards selection and drop `DistCoast`:

```
fullModel_OD_revised<-glm(Count ~ XPos+YPos+Depth + FMonth + XPos*Phase+YPos*Phase, data=d
Anova(fullModel_OD_revised, test="F")
```

```
Analysis of Deviance Table (Type II tests)

Response: Count
Error estimate based on Pearson residuals

            Sum Sq    Df  F value    Pr(>F)
XPos         27720     1 263.3418 < 2.2e-16 ***
YPos         44816     1 425.7565 < 2.2e-16 ***
Depth        97905     1 930.1043 < 2.2e-16 ***
FMonth       12268     3  38.8495 < 2.2e-16 ***
Phase         3286     2  15.6096 1.676e-07 ***
XPos:Phase    2063     2   9.8002 5.561e-05 ***
YPos:Phase    1464     2   6.9531 0.0009571 ***
Residuals  3314619 31489
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The fit of the revised model

```
summary(fullModel_OD_revised)
```

```
Call:
glm(formula = Count ~ XPos + YPos + Depth + FMonth + XPos * Phase +
    YPos * Phase, family = quasipoisson, data = df, offset = log(Area))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-10.876   -2.576   -1.609   -0.684  105.547
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.342e+03  9.661e+01  13.890  < 2e-16 ***
XPos         3.470e-05  5.534e-06   6.272 3.62e-10 ***
YPos        -2.252e-04  1.569e-05 -14.356  < 2e-16 ***
Depth       -3.014e-01  1.186e-02 -25.408  < 2e-16 ***
FMonth2      1.344e-01  1.239e-01   1.085 0.277983
FMonth3      8.193e-01  9.702e-02   8.445  < 2e-16 ***
FMonth4      1.722e-01  1.054e-01   1.633 0.102497
PhaseB      -1.031e+02  9.985e+01  -1.032 0.301976
PhaseC      -5.464e+02  1.473e+02  -3.709 0.000209 ***
XPos:PhaseB  2.787e-05  7.472e-06   3.730 0.000192 ***
XPos:PhaseC  3.975e-05  1.111e-05   3.579 0.000345 ***
YPos:PhaseB  1.387e-05  1.601e-05   0.866 0.386341
YPos:PhaseC  8.574e-05  2.359e-05   3.634 0.000279 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 105.2642)

    Null deviance: 664249  on 31501  degrees of freedom
Residual deviance: 526735  on 31489  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 8
```

We can see the overdispersion has been significantly reduced (but is still high).

5.7.3 Model Assessment

We will plot the observed versus the fitted values, and assess both the mean variance relationship and the residual independence assumption.

```
phi_hat <- summary(fullModel_OD_revised)$dispersion
fitted_values <- fitted(fullModel_OD_revised)
res_raw <- residuals(fullModel_OD_revised)
scaled_resid <- (df$Count-fitted_values)/sqrt(phi_hat*fitted_values)

p<-list()
p[[1]]<-qplot(df$Count,fitted(fullModel_OD_revised)) + geom_abline(intercept=0,slope=1,col
                  xlab("Count") + ylab("Fitted values")
p[[2]]<-qplot(fitted(fullModel_OD_revised), scaled_resid) + xlab("Fitted values") + ylab("

library(gridExtra)
```

```
library(grid)
grid.arrange(grobs=p, nrow = 1)
```

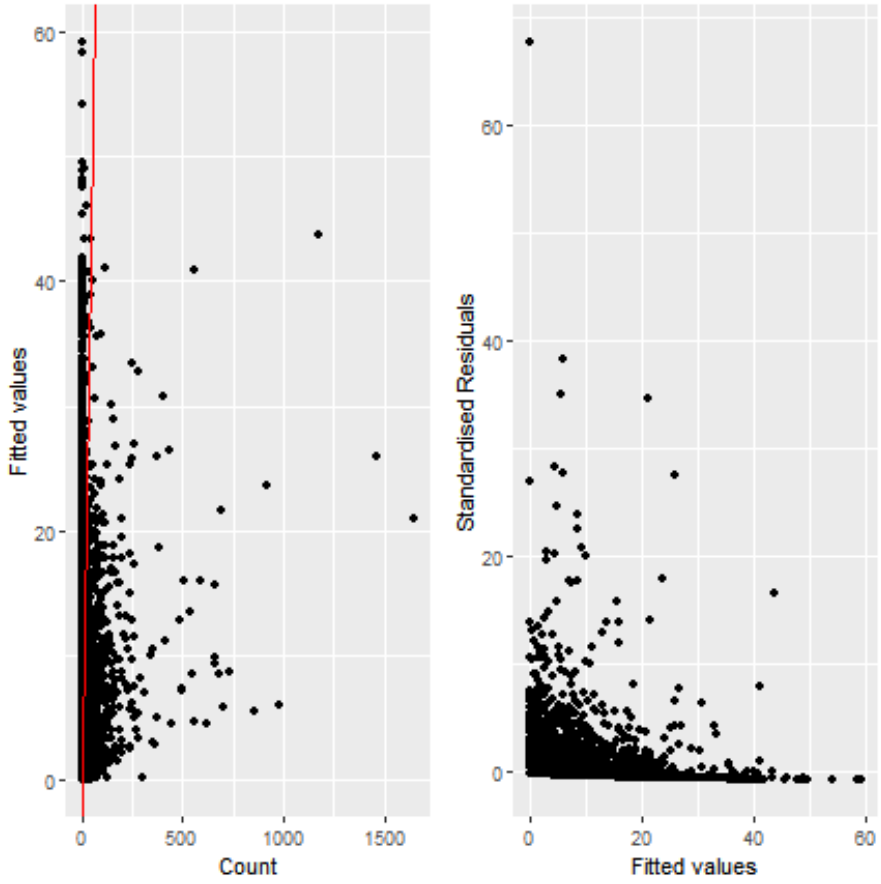


FIGURE 5.1 Model assessment for the best model

Checking autocorrelation

```
par(mfrow =c(1,2))
set.seed(5)
acf(rnorm(length(scaled_resid)), lag.max = 40, main = "Ideal correlation")
acf(scaled_resid, main = "Actual correlation")
```

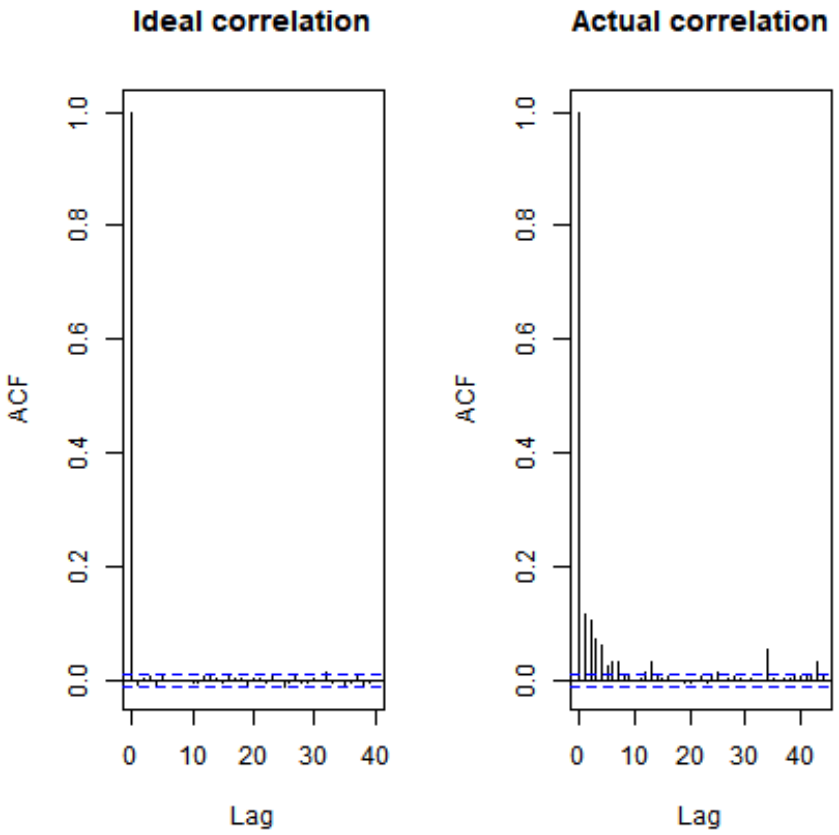


FIGURE 5.2 ACF plot for the scaled residuals from the best model (right plot) compared to an ACF of iid normal random variables (left plot)

We can see that we have some severe underprediction of the very large values (left-hand plot, Figure 5.1) and some evidence that the spread of the residuals is a bit larger for the smaller fitted values which renders the mean-variance relationship a bit suspect. There is also some evidence for correlation in the data after the model has been fitted (Figure 5.2) , suggesting our residuals along transects are not independent. A 'runs-test' could be used to confirm this correlation (see MT5764 for more).

5.7.4 Model Interpretation

The final model chosen has 12 parameters:

$$\hat{y}_{it} = \hat{\lambda}_{it} = \text{area}_{it} \times e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1it} + , ..., + \hat{\beta}_{12} x_{12it}}$$

and on the log-link scale as:

$$g(\hat{\lambda}_{it}) = \log(\hat{\lambda}_{it}) = \log(\text{area}_{it}) + \hat{\beta}_0 + \hat{\beta}_1 x_{1it} + , ..., + \hat{\beta}_{12} x_{12it}$$

where y_{it} represents (corrected) bird counts at transect i at time t, area_{it} represents the area associated with each location along the transects, x_{1it} represents the X coordinate and x_{12it} is a dummy variable, which is "switched on" in Phace C.

Under the model, bird numbers are predicted to:

- increase with increases in the X-coordinate direction
- decrease with increases in the Y-coordinate direction and depth.
- increase with the X-coordinate more rapidly for phases B and C compared to A. Similarly for the Y-coordinate.
- be significantly lower in January compared with March (but numbers were statistically indistinct between January and February and January and April)

There is still a large amount of unexplained variability in the model residuals. The residual variability observed is just over 100 times what we would expect under a Poisson model. This extra variation may be due in part, to:

- the large number of zeros in the data
- the sometimes very large bird counts (e.g. > 1000) and/or
- non-independence (i.e temporal autocorrelation) in model residuals.

Conisidering this extra-Poisson variability is important for model conclusions. In our case, however, it didn't make much difference.

This extra-poisson variation could also be handled using *zero inflated poisson models* (to allow for the large numbers of zeros) and/or *negative binomial* models.

Generalized estimating equation based-models (GEEs) can also be considered for these data. GEEs work much like GLMs but allow for any non-independence to be detected and incorporated into the model.

Note, the AIC statistic was unavailable for the quasi-poisson (overdispersed) models. This is because by fitting such a model we are admitting a Poisson GLM is inappropriate and so, a true likelihood for this model is unavailable. Instead, we used a backwards model selection approach based on model p-values. We could use a QAIC (quasi-likelihood based information criterion) and this is now implemented using the QAIC function in the `MuMIn` library.

The highest (corrected) counts were difficult to predict. Predictions -were only as large as 60 when counts as high as about 1600 were observed. However, only about 1% of bird counts were larger than 60.

6

Modelling changes in presence pre and post impact

6.1 Estimating proportions

In this chapter we are going to quantify the probability of seeing one or more birds in this area during the survey period using presence/absence.

Research questions

1. How can we quantify the probability of seeing birds in this area for the pooled data?
2. Is there any evidence that bird presence/absence differs across time and/or phases? If so, how so?
3. Can we predict the probability of seeing birds in this area using the environmental data available? If so, which covariates are best at predicting bird sightings?

We will address these questions using:

- graphical summaries,
- finding sample proportions using Maximum Likelihood,
- confidence intervals for proportions,
- hypothesis tests for proportions (the z-test) and,
- Generalized Linear Models with Binomial errors.

Exploratory Data Analysis Trained observers recorded the presence or absence of birds at each spatial location; when birds were seen from the plane the variable `pres=1` and when birds were absent `pres=0`. Summary statistics for this variable for the entire survey period and for each year can be found using the code below and are illustrated in Figure 6.1.

```
# Creating a presence variable
df$Pres<-ifelse(df$Count>0,1,0)
# Frequency
table(df$Pres)
```

```
    0     1
28266  3236
```

```
#sighting data for each year
table(df$Pres, df$YearMonth)
```

```
   2000/1 2000/2 2000/4 2001/1 2001/2 2001/3 2001/4 2002/1 2002/2 2002/3
0     997    555   2131    865    927    939   1020    996    994    911
1     113     36     69    125    175    165     57    106    105    192

   2003/1 2003/3 2003/4 2004/1 2004/3 2004/4 2005/1 2005/3 2005/4 2007/2
0     807    940   1042   1009   1911   1003    988    927   1007    929
1      95    164     61     94    292    101    121    175     96    168

   2007/3 2007/4 2011/2 2011/3 2011/4
0     894   1038   1130   2104   2202
1     207     59     62    247    151
```

```
fr<-table(df$Pres, df$YearMonth)
p<-rep(c("No","Yes"),each=ncol(fr))
fr<-data.frame(rep(colnames(fr),2), as.vector(t(fr)),p)
colnames(fr)<-c("YearMonth","Fr", "Pres")

ggplot(fr, aes(x= YearMonth, y = Fr, fill=Pres))+
  geom_bar(stat="identity",position = position_stack(reverse = TRUE)) +
  geom_text(aes(y=Fr, label=Fr), vjust=1.2, color="white", size=2.2, position = position_s
  theme(axis.text.x=element_text(angle=90)) + ylab("Frequency")
```

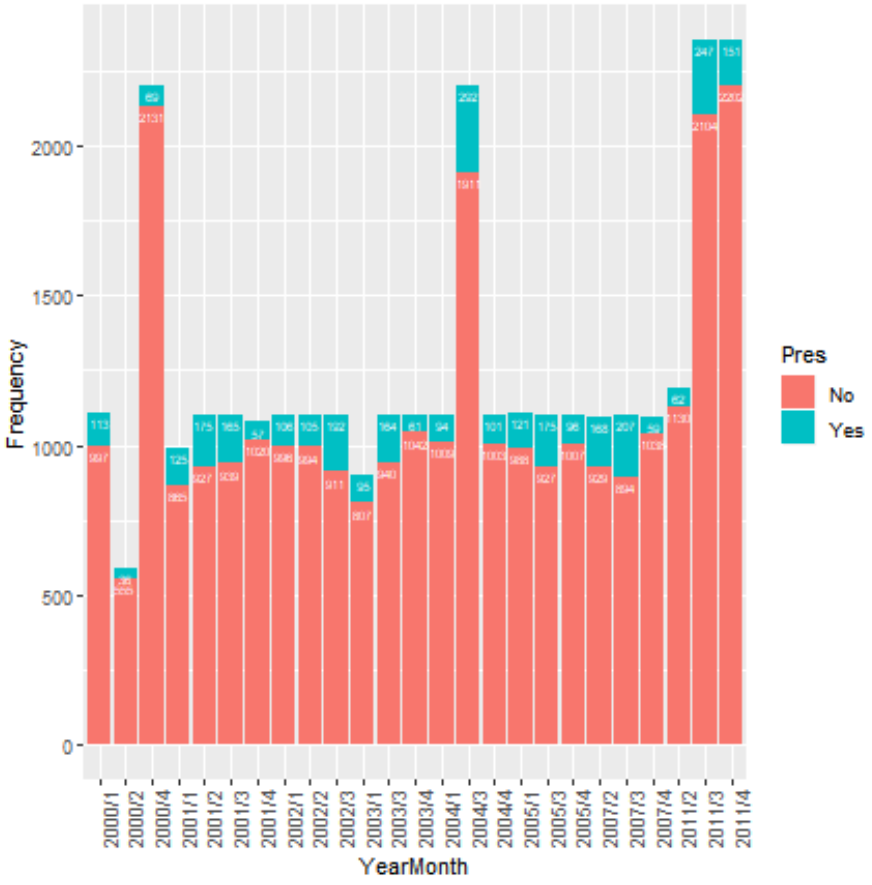


FIGURE 6.1 Barplot showing the presence/absence data across the survey period.

What can we conclude?

- There were many more absence than presence observations.
- Survey effort was highest in April 2000, March 2004 and March-April 2011
- From this plot alone it is very difficult to determine the proportion of observations where one or more birds were seen.
- It is also difficult to see how this proportion may have changed across time.

To estimate the probability of sighting birds in this area we are going to quantify the proportion of observations with birds present at the surface (during some specified period) using a summary statistic. We are going to estimate the *underlying* and *unknown* probability of sighting one or more birds using maximum likelihood.

6.2 Estimating a proportion: The theory

To estimate the probability of sighting birds in this area for the entire survey period we are going to assume the data have come from a Binomial distribution with known n_i (fixed number of trials) and unknown p (probability of success). We will estimate the proportion of ''successful'' observations, p, using the number of observations with sightings ("successes") and the total number of observations ("trials").

Recall, there are 4 conditions of the Binomial distribution: two outcomes, a fixed number of trials, independence of observations, constant probability of success.

Task 2

How realistic are these 4 conditions in this case?

Show Solution on P??

We have 25 year-month combinations with sighting information (ie. $n = 25$) and for each we have the number of successful observations and the number of observations in total.

TABLE 6.1: Table showing the survey effort in the three construction phases.

No. successes	No. of visits
$y_1 = 113$	$n_1 = 1110$
y_2	$n_2 = 591$
...	...
$y_{25} = 151$	$n_{25} = 2353$

Y_i = no. of visits with sightings out of the n_i visits $\sim Binomial(n_i, p), i = 1, ..., 25$

The binomial distribution can be written as:

$$f(y_i; p) = \begin{pmatrix} n_i \\ y_i \end{pmatrix} p^{y_i}(1-p)^{n_i - y_i}$$

and so the likelihood function is,

$$f(y_1, ..., y_{25}; p) \propto \prod_{i=1}^{25} p^{y_i}(1-p)^{n_i - y_i} \propto L(p)$$

As before, the log of the likelihood function, $\log(L(p)) = l(p)$, is what is maximised. This can be written as:

$$Log(L(p)) = l(p) \propto \sum_{i=1}^{25} [y_i \log(p) + (n_i - y_i) \log(1-p)]$$

where $y_i = y_1, ..., y_{25}$ is the number of successes, $n_i = n_1, ..., n_{25}$ is the number of trials, and p is the probability of success.

The likelihood and log-likelihood functions for the probability of sighting birds in the first year-month combination can be plotted:

```
sighting_rates<- df %>%
                   group_by(YearMonth) %>%
                   summarise(successes=sum(Pres), trials=length(Pres))
sighting_rates
```

```
# A tibble: 25 x 3
   YearMonth successes trials
   <fct>         <dbl>  <int>
 1 2000/1          113   1110
 2 2000/2           36    591
 3 2000/4           69   2200
 4 2001/1          125    990
 5 2001/2          175   1102
 6 2001/3          165   1104
 7 2001/4           57   1077
 8 2002/1          106   1102
 9 2002/2          105   1099
10 2002/3          192   1103
# ... with 15 more rows
```

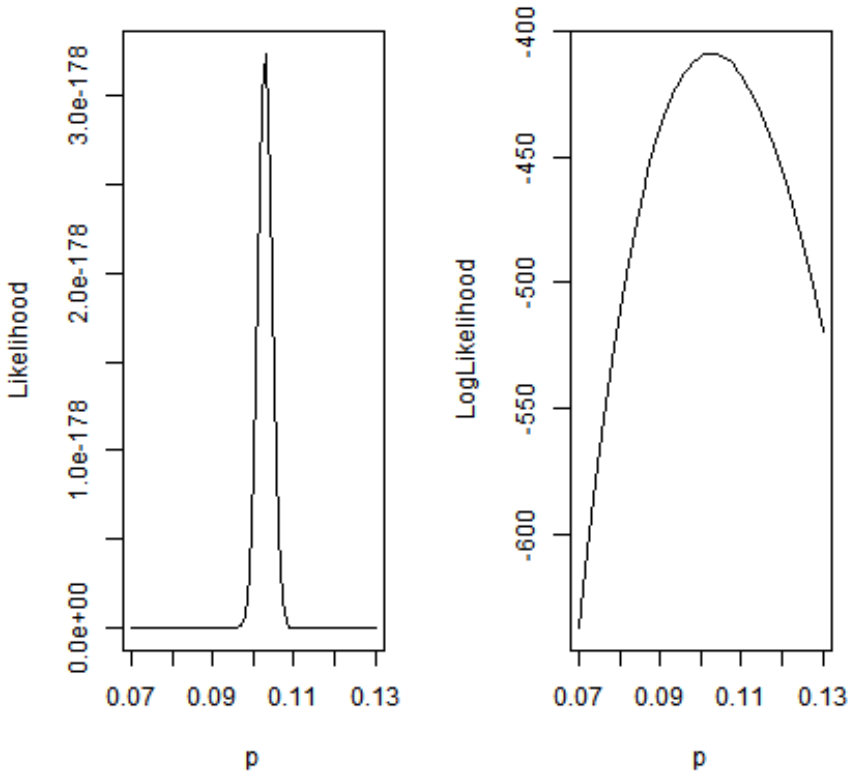


FIGURE 6.2 Likelihood and Loglikelihhod functions of the data

From the likelihood function and log-likelihood functions in Figure 6.2, $\hat{p} \approx 0.1$.

The 'best guess' in light of the data: estimating the probability –of sighting birds (p).

We estimate p by maximising the log of the likelihood function.} This is done by differentiating the log likelihood with respect to p and setting this derivative to zero.

The estimate of p using the sightings data (for the entire survey period) can be found using:

$$\hat{p} = \frac{\sum_{i=1}^{25} y_i}{\sum_{i=1}^{25} n_i} \qquad = \frac{\text{total number of successes}}{\text{total number of trials}} = \frac{3236}{31502}$$

and this is a general result:

If

$$Y_1, ..., Y_n$$

are indep. with

$$Y_i \sim Binomial(n_i, p)$$

, and n_i is known and *p* unknown, then **the MLE of** p is:

$$\hat{p} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{N} n_i} = \frac{\text{total number of successes}}{\text{total number of trials}}$$

To find the sample proportion for bird prevalence for the entire study period and for each year, we type:

```
# Sample proportion
phat<-sum(sighting_rates$successes)/sum(sighting_rates$trials)
phat
```

```
[1] 0.1027236
```

```
# Estimated probabilities
mean(df$Pres)
```

```
[1] 0.1027236
```

```
# Estimated probabilities per YearMonth
phats<-df %>% group_by(YearMonth) %>% summarise(phat=mean(Pres))
phats %>% print(n=25,width=Inf)
```

```
# A tibble: 25 x 2
   YearMonth   phat
   <fct>      <dbl>
 1 2000/1    0.102
 2 2000/2    0.0609
 3 2000/4    0.0314
 4 2001/1    0.126
 5 2001/2    0.159
 6 2001/3    0.149
 7 2001/4    0.0529
 8 2002/1    0.0962
 9 2002/2    0.0955
10 2002/3    0.174
```

```
11 2003/1     0.105
12 2003/3     0.149
13 2003/4     0.0553
14 2004/1     0.0852
15 2004/3     0.133
16 2004/4     0.0915
17 2005/1     0.109
18 2005/3     0.159
19 2005/4     0.0870
20 2007/2     0.153
21 2007/3     0.188
22 2007/4     0.0538
23 2011/2     0.0520
24 2011/3     0.105
25 2011/4     0.0642
```

6.3 Quoting a range of plausible probabilities for bird prevalence: Confidence intervals for Binomial data

It turns out that sample proportions (estimates for p; $\hat{p}$) are Normally distributed about the true (underlying) population proportion (in **large** samples). So CIs for Binomial data are constructed in a similar way to CIs for large samples of Normal data. The standard deviation of the sample proportion (the standard error) can be found using:

$$se(\hat{p}) = \sqrt{\frac{p(1-p)}{\sum_{i=1}^{N} n_i}}$$

and

$$\hat{p} \sim Normal\left(p, se(\hat{p}) = \sqrt{\frac{p(1-p)}{\sum_{i=1}^{N} n_i}}\right)$$

This means that (for large samples):

- n estimate will mostly likely lie within 2 standard errors from *p* and so,
- building a CI with limits that stretch about 2 standard errors either side of the estimate, should capture the true parameter, *p*, most of the time.

Therefore, to find a 95% confidence interval for *p* we use:

$$\text{estimate} \quad \pm z_{0.025} \times \quad \text{standard error}$$
$$\hat{p} \pm 1.959964 \times \sqrt{\frac{\hat{p}(1-\hat{p})}{\sum_{i=1}^{N} n_i}}$$

How large is large enough?

To assume these estimates are approximately normally distributed about p, we require large samples. A rule of thumb to check whether the sample size is large enough

$$n\hat{p} > 5$$
$$n(1-\hat{p}) > 5$$

In this case our estimate for the entire survey period is about 0.1027236 and the total number of site visits is 31502, so results based on these large-sample properties should be sound.

6.3.1 Binomial confidence intervals: large sample sizes

We are going to build a 99% CI for the probability of sighting birds during the survey period. We can do this as follows:

$$\widehat{se}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = \sqrt{\frac{0.1027236(1-0.1027236)}{31502}} = 0.001710524$$

$$99\%\, CI = \hat{p} \pm z_{0.005} \times \widehat{se}(\hat{p}) = 0.1027236 \pm 2.575829 \times 0.001710524 = (0.09831758, 0.1071296)$$

Interpret the interval: We can be 99% confident that the probability of sighting a bird in this area is somewhere between 0.09 (9.0%) and 0.107 (10.7%).

6.3.2 Binomial confidence intervals: small sample sizes

For 'small' samples (less than those in the table) when p is close to zero or one, this machinery doesn't work (see the Asymptotic results below for $\hat{p} = 0.01, n = 100$). When sample size is low in these cases, CIs can easily include impossible values (less than zero, greater than 1). The literature suggests that "Wilson"" intervals

are preferred in these cases (Agresti and Coull, 1998)[1] which are the default intervals provided using the `binconf` function in `R`.

```
require(Hmisc)
binconf(1,100,method="all")
```

```
           PointEst          Lower      Upper
Exact          0.01  0.0002531460 0.05445939
Wilson         0.01  0.0005129329 0.05448620
Asymptotic     0.01 -0.0095013954 0.02950140
```

```
binconf(1,100)
```

```
 PointEst        Lower     Upper
     0.01 0.0005129329 0.0544862
```

For the bird data, the total sample size is large but the monthly samples ($n_i \geq 591$) may not be large enough given the small $\hat{p}$'s.

A comparison is prudent:

```
binconf(3236,31502,method="all")
```

```
            PointEst      Lower     Upper
Exact      0.1027236 0.09939194 0.1061288
Wilson     0.1027236 0.09941936 0.1061248
Asymptotic 0.1027236 0.09937107 0.1060762
```

For a wider variety of confidence interval methods, use `binom.confint(3236,31502)` in the `Hmisc` package.

And for each annual sighting proportion:

```
binconf(sighting_rates$successes,sighting_rates$trials)
```

```
   PointEst      Lower      Upper
 0.10180180 0.08536385 0.12098640
 0.06091371 0.04432107 0.08317754
 0.03136364 0.02485781 0.03950319
 0.12626263 0.10700688 0.14840756
```

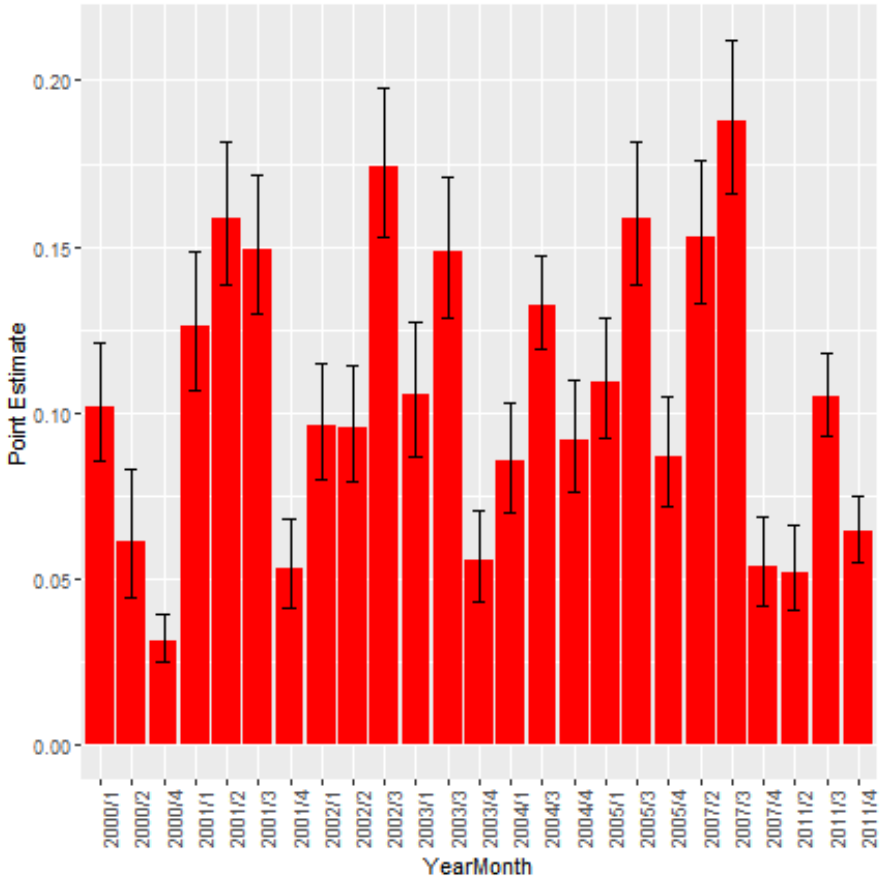
[1] Agresti and Coull, *Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions*, The American Statistician, Vol.52, 1998, pp.119-126

```
0.15880218 0.13841320 0.18156166
0.14945652 0.12964192 0.17170215
0.05292479 0.04107235 0.06795517
0.09618875 0.08015686 0.11502614
0.09554140 0.07954383 0.11435662
0.17407072 0.15283552 0.19756829
0.10532151 0.08693524 0.12705525
0.14855072 0.12879157 0.17074719
0.05530372 0.04329325 0.07040095
0.08522212 0.07014980 0.10317354
0.13254653 0.11902453 0.14734778
0.09148551 0.07586642 0.10993766
0.10910730 0.09208931 0.12882398
0.15880218 0.13841320 0.18156166
0.08703536 0.07180030 0.10513693
0.15314494 0.13304728 0.17566335
0.18801090 0.16603108 0.21216025
0.05378304 0.04192336 0.06875693
0.05201342 0.04078501 0.06612001
0.10506168 0.09330448 0.11810740
0.06417340 0.05496460 0.07480291
```

Do sighting probabilities appear to differ across time?

```
# Confidence intervals
phatci<-data.frame(fr[1:25,1],binconf(sighting_rates$successes,sighting_rates$trials))
colnames(phatci)[1]<-"YearMonth"

ggplot(phatci, aes(x=YearMonth, y=PointEst)) +
  geom_bar(stat="identity",fill="red") +
  geom_errorbar(aes(ymin=Lower, ymax=Upper), width=0.4) +
     theme(axis.text.x=element_text(angle=90)) + ylab("Point Estimate")
```



Does bird prevalence appear to differ during the survey period (i.e. for the year month combinations)?

- It appears so; the 95% CIs for the proportions share values for some year month combinations and not for others.
- It is possible that there are some year-month differences, but we cannot tell from these results alone.
- The various methods for calculating the intervals give virtually identical results.

- We need to formally test for differences between the time periods.

6.4 Comparing proportions using the z-test

Comparing sighting probabilities in January 2000 and April 2011: the z-test:

We are going to **formally** compare sighting probabilities in the survey area in the beginning and end of the survey period using a hypothesis z-test. Rather than use confidence intervals which give a range of likely values for each parameter we are going to test for **no difference** between these parameter values and evaluate the strength of evidence against this null hypothesis.

What do we want to test?

We want to test the research hypothesis that bird prevalence was different at the start and the end of the survey period. We can test this research hypothesis using the null hypothesis of no difference:

$$H_0 : p_{Jan,2000} - p_{April,2011} = 0$$

,

$$H_1 : p_{Jan,2000} - p_{April,2011} \neq 0$$

What do we expect to see if the null hypothesis is true?

We expect to see small differences between sighting rates in these months.

How does our data-estimate compare with the hypothesized value?

To quantify the uncertainty in our estimate of the difference between sighting rates in each time period we can use the following:

$$\widehat{se}(\hat{p}_1 - \hat{p}_2) = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}} = \sqrt{\frac{0.10180180(1-0.10180180)}{1110} + \frac{0.06417340(1-0.06417340)}{2353}} = 0$$

Note: this uncertainty measure requires these sample proportions are independent.
Do you think this is realistic in this case?

Assuming independence between sample proportions, the estimate is:

$$frac\text{estimate-hypothesised value}\text{standard error} = \frac{0.0376284}{0.01038747} = 3.62248$$

standard errors from zero.

What is the chance of getting a value at least as extreme as 3.62 when H_0 is

true and there was **no difference** in sighting probabilities at each end of the survey period? Our data-estimates are $\hat{p}_{Jan,2000} - \hat{p}_{April,2011} = 0.10180180 - 0.06417340 = 0.0376284$ and the hypothesised value is 0, therefore our estimate is 0.0376284 units from zero .

```
2*pnorm(-3.62248)
```

```
[1] 0.000291792
```

What can we conclude?

- We have **very strong evidence** for a difference in bird prevalence between January 2000 and April 2011.
- We can also reject the null hypothesis of no difference in bird prevalence at each end of the survey period at both the 5% and 1% levels.
- Specifically, the probability of sighting a bird at the end of the survey period appears to be significantly lower than that at the beginning.

7

GLMs for Proportions

We are going to model the proportion of successful observations (locations with birds sighted) during the survey period using a statistical model. We could approach this using linear regression, but this is inappropriate for many reasons:

- the probability of sighting a bird is not guaranteed to change linearly with time,
- the response (the proportion of successful visits) is naturally bounded by zero and one (i.e. $0 \leq p \leq 1$) and
- linear model predictions can give values outside this range
- the errors are unlikely to be normal with constant variance.

While we can find ways to relax the linearity, normality and constant variance assumptions, predictions may still need truncating if linear models are used.

Recall: **Generalized Linear Models (GLMs)** are an extension of standard linear models in that they allow the response data (given a model) to follow any distribution inside the "exponential family". This family of distributions includes the Normal, Binomial, Poisson and Gamma distributions. GLMs allow the mean of the response to be a function of the "linear predictor" via a "link function".

GLMs can be appropriate for discrete, proportional data and GLMs with Binomial errors will be used to model the probability of bird presence over time.

The proportion of observations with birds sighted over time, can be viewed using:

```
sighting_rates<- df %>%
  group_by(Year) %>%
  summarise(successes=sum(Pres), trials=length(Pres)) %>% mutate(Props = successes/trials)
# Plot
qplot(x = Year, y= Props, data= sighting_rates) +
  scale_x_continuous(breaks=seq(2000,2010,2), labels=seq(2000,2010,2))
```

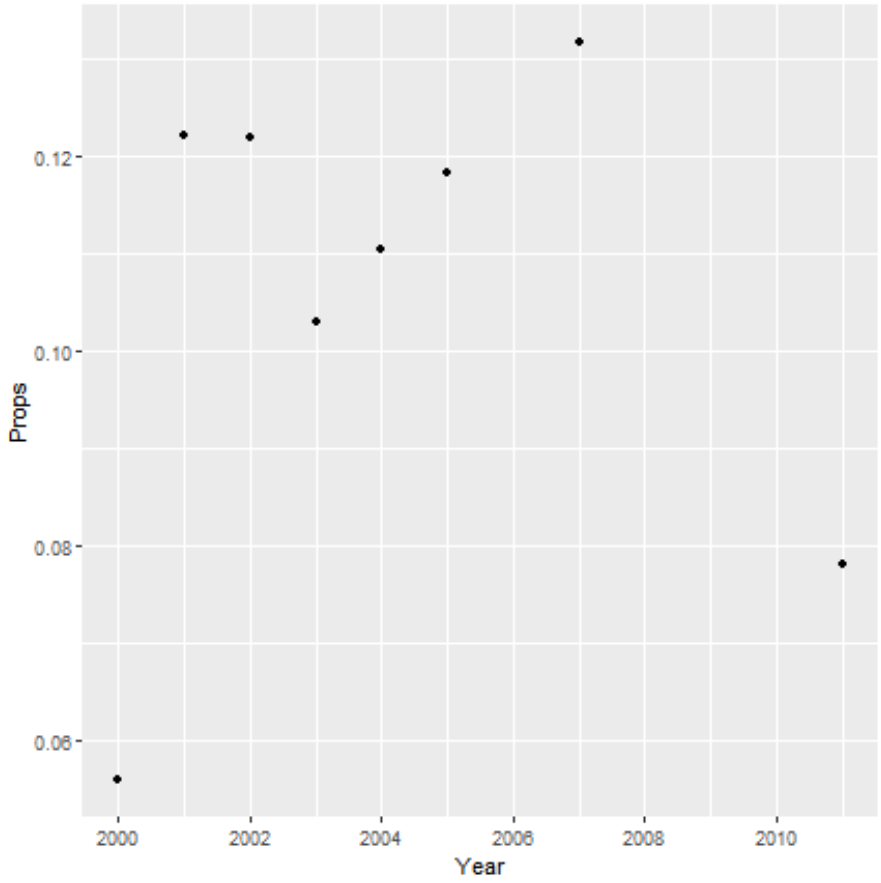


FIGURE 7.1 Bird sighting prevalence over time

Comments on the data:

- There seems to be an unclear pattern in bird presence over time and
- it is difficult to tell if any differences are due to sampling variation alone or due to underlying differences in bird prevalence.

7.1 Model Specification

We are going to proceed as before and use a Binomial distribution to model bird prevalence using the proportion of successful observations in the survey area. Recall:

$$Y_i = \text{no. of observations with bird sightings out of the}$$

$$n_i \text{ observations } \sim \text{Binomial}(n_i, p)$$

and we are going to allow *p* to vary across time using a GLM with a "logit link function" (a "logistic regression model").

The *logistic regression model* (with one predictor) can be written in terms of the response:

$$p_i = \frac{e^{\eta_i = \beta_0 + \beta_1 x_i}}{1 + e^{\eta_i = \beta_0 + \beta_1 x_i}} \tag{7.1}$$

or in terms of the link function $g(p_i)$:

$$g(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \eta_i = \beta_0 + \beta_1 x_i \tag{7.2}$$

The logit link function maps the response data onto the real line $(-\infty, +\infty)$ and ensures the model predictions are bounded by zero and one. The *so-called* linear predictor (η_i) represents the linear component $\beta_0 + \beta_1 x_{1i}$ on the scale of the link function.

So under the model, the relationship between the response (p_i) and the predictor (x_i) is nonlinear, and the relationship between the response on the link scale $g(p_i)$ and x is linear. For example, if we choose $\beta_0 = 1.5$, $\beta_1 = -1$ and $x = 1, ..., 7$ in Equations (7.1) and (7.2) the following figure results:

```
X=1:7
Y=1.5-X
df_exmp<-data.frame(X,Y) %>%
  mutate(Logit=exp(Y)/(1+exp(Y)))
p<-list()
p[[1]]<-qplot(x=X,y=Logit,data = df_exmp, geom=c("point", "line"))
p[[2]]<- qplot(x=X,y=Y, data= df_exmp, geom=c("point", "line"))
grid.arrange(grobs=p, nrow=1)
```

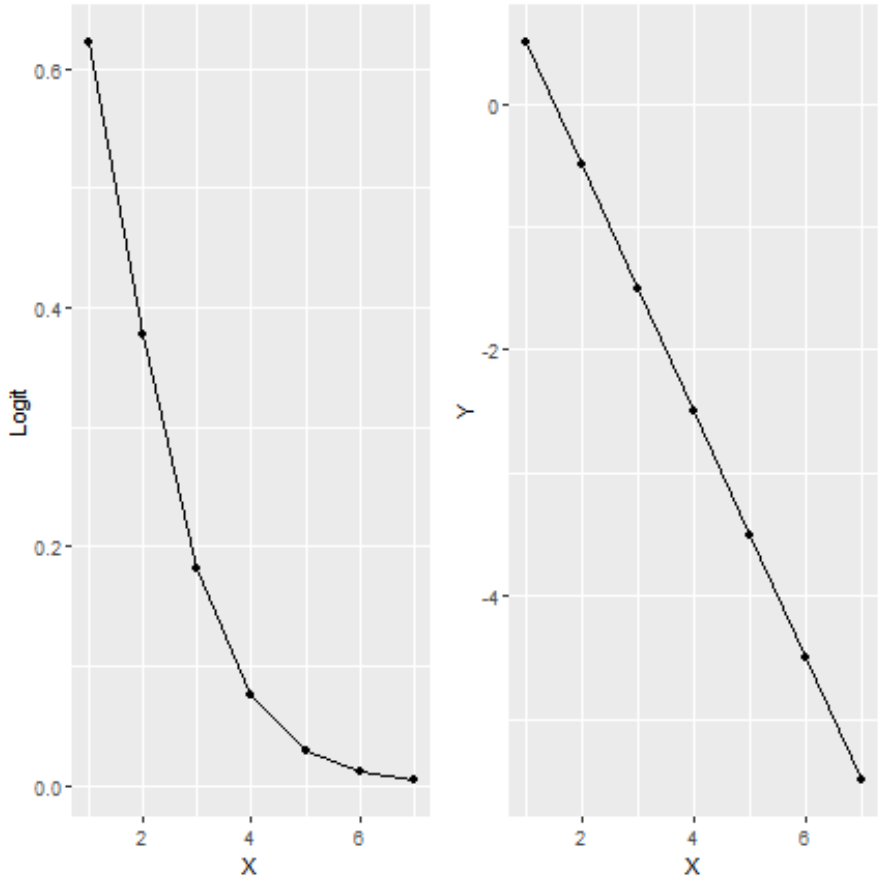


FIGURE 7.2 Illustration of x vs y and x vs y on the scale of the link function.

While the logit link $\left(\log(\frac{p}{1-p})\right)$ is most common for binomial GLMs, there are two other link functions which serve this purpose:

$$\begin{aligned} g(p_i) =& \Phi^{-1}(p_i) \quad \text{probit model} \\ g(p_i) =& \log(-\log(1-p_i)) \quad \text{Complementary log-log (cloglog)} \end{aligned} \tag{7.3}$$

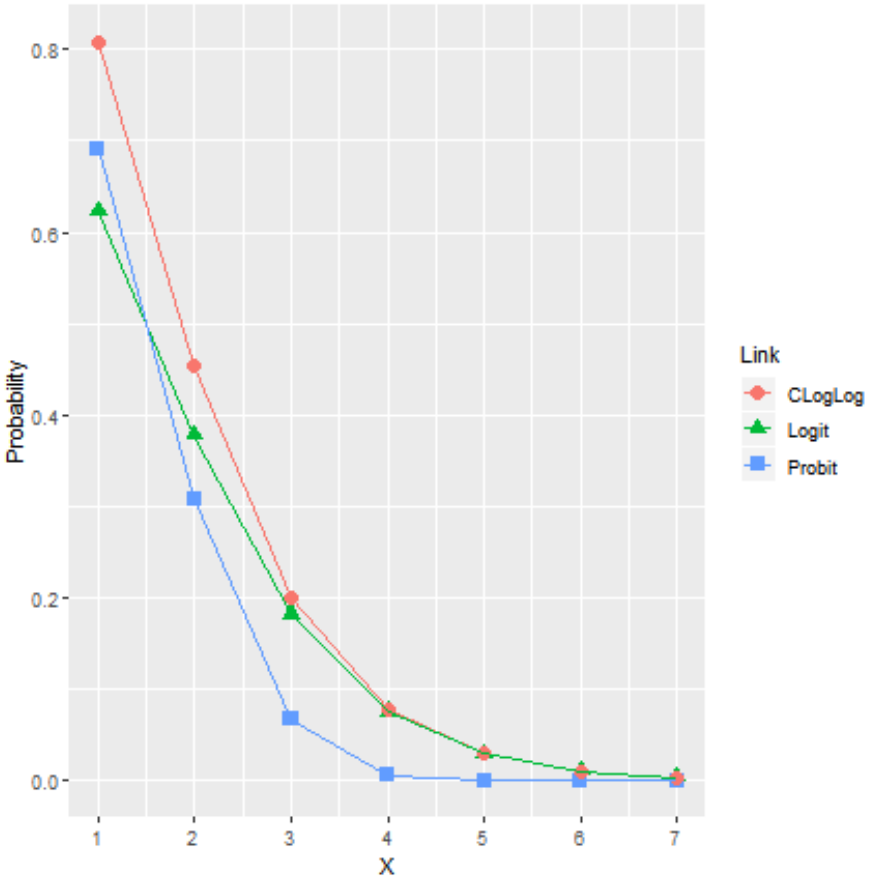
where $\Phi(.)$ is the cumulative distribution function of $N(0,1)$ which can be obtained using the `pnorm` function in `R`. These models can also be written in terms of p_i (using the "inverse link function").

$$p_i = \Phi(\beta_0 + \beta_1 x_i) \quad \text{probit model}$$
$$p_i = 1 - \exp(-\exp(\beta_0 + \beta_1 x_i)) \quad \text{Complementary log-log (cloglog)}$$

For a visual comparison of these link functions, the probability of "success", p is plotted using $x = 1, ..., 7$, $\beta_0 = 1.5$ and $\beta_1 = -1$ for the 3 link functions described.

```
df_exmp<-df_exmp %>%
  mutate(Probit=pnorm(Y), CLogLog=1-exp(-exp(Y)))
df_plot<-select(df_exmp,-Y) %>%
  gather(key="Link", value = "Probability", - X)
```

```
ggplot(df_plot,aes(x=X,y=Probability)) +geom_point(aes(color=Link, shape=Link), size=3) +
  geom_line(aes(color=Link)) + scale_x_continuous(breaks=1:7)
```



We can fit Binomial GLMs using logit, probit and complementary log-log link functions using ML. For a logit link function the following log-likelihood is maximised:

$$\log(L) = \sum_{i=1}^{n} \left(y_i \log\left(\frac{p_i}{1-p_i}\right) + n_i \log(1-p_i) + \log\left(\binom{n_i}{y_i}\right) \right)$$

y_i is the number of successful observations, p_i is the ''success" probability and n_i is the total number of observations (the trials).

The mean and variance assumed under this Binomial model (for all link functions) are:

$$E(Y_i) = n_i p_i \qquad \text{and} \qquad V(Y_i) = \phi n_i p_i (1-p_i) = n_i p_i (1-p_i) \quad (\phi = 1)$$

A GLM using each of the 3 link functions can easily be fitted using R.

```
binLogit<- glm(cbind(successes,trials-successes) ~
                 Year, data=sighting_rates,
               family=binomial)
binProbit<- glm(cbind(successes,trials-successes) ~
                  Year, data=sighting_rates,
                family=binomial(link="probit"))
binCloglog<- glm(cbind(successes,trials-successes) ~
                   Year, data=sighting_rates,
                 family=binomial(link="cloglog"))
```

7.2 Model Selection

As for linear models with normal errors, log-likelihood values and AIC statistics can help choose between models with different link functions and/or model covariates; large log-likelihood values and small AIC values signal well fitting models.

```
AIC(binLogit,binProbit,binCloglog)
```

```
           df      AIC
binLogit    2 282.6905
binProbit   2 282.6195
binCloglog  2 282.7147
```

For these data there is no appreciable difference in model results when different link functions are used; the differences in the AIC scores between models are very small. In this case, the logistic model would typically be chosen; model interpretation is often easier when the logit link is used.

```
summary(binLogit)
```

```
Call:
glm(formula = cbind(successes, trials - successes) ~ Year, family = binomial,
    data = sighting_rates)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-11.113   -1.450    2.277    3.258    5.657

Coefficients:
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 14.513016  10.145223   1.431    0.153
Year        -0.008322   0.005061  -1.644    0.100

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 219.82  on 7  degrees of freedom
Residual deviance: 217.10  on 6  degrees of freedom
AIC: 282.69

Number of Fisher Scoring iterations: 4
```

7.3 Parameter interpretation

Estimates: 14.5130156 and -0.0083216. The slope coefficient is negative and so the probability of bird presence is decreasing over time in the model. Parameter estimates for logistic models can also be interpreted as the odds of success.

The odds/odds of success

In this case a "success"" is a visit where one or more birds were seen; we can calculate the odds of presence vs absence using:

$$\text{odds} = \frac{\text{probability of success}}{\text{probability of failure}} = \frac{p_{it}}{1-p_{it}} = e^{\beta_0} e^{\beta_1 x_{1it}} = e^{\beta_0 + \beta_1 x_{1it}}$$

So, if the probability of success=0.75, the probability of failure is 0.25. When the odds ≥ 1, success is more likely than a failure. When the odds $= 3$ a success is 3 times as likely as a failure; we expect about 3 successes for every failure. Conversely when the odds $= 1/3$ a failure is 3 times as likely as a success; we expect about one success for every 3 failures.

Odds of success across years:

$$\widehat{\left(\frac{p_{it}}{1-p_{it}}\right)} = e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1it}}$$

The estimated odds of presence vs absence in:

- year 0 is $e^{14.513} = 2008703$ (nonsense!)
- year 2000 is $e^{14.513+(-0.008322*2000)} = 0.1187185$ (just over $\frac{1}{10}$)
- year 2001 is $e^{14.513+(-0.008322*2001)} = 0.1177346$

Using the odds we can say that every 1 unit increase in x has a multiplicative effect of $e^{\hat{\beta}}$ on the odds. $e^{\hat{\beta}_0}$ is the odds of bird presence vs bird absence at `year = 0`. Thus, each time year increases by one unit, the odds of bird presence are multiplied by $e^{\hat{\beta}_1}$. So, as we move from year 2000 to year 2001 (a 1 unit increase), the odds of presence vs absence are estimated to change by a factor $e^{\hat{\beta}_1} = e^{-0.008322} = 0.9917125$ and $0.1187185 \times 0.9917125 = 0.1177346$.

7.4 Parameter Inference

Confidence interval for parameters

As for the Poisson-based GLM, we use a z-multiplier to construct confidence intervals.

$$\text{estimate} \quad \pm z - \text{multiplier} \times \text{standard error}$$

Testing for non-zero covariate relationships

We also test for no relationship between each covariate and the response in the standard way using a χ^2 test. This is a likelihood ratio test between a model with the covariate and one without.

$$H_0 : \beta_p = 0, \quad H_1 : \beta_p \neq 0$$

(Note: for factor covariates, this is a collection of β's.)

Thus, large test statistics (and small p-values) give compelling evidence for a non-zero relationship with the response.

What do we see here?

- the p-value for year is 0.1, which means that we have weak evidence against H_0 (H_0: β_1=0).
- We can use the `Anova` function in the `car` library to find the χ^2 statistic and corresponding p-value. The p value is ~ 0.1, which means we have weak evidence against H_0 (H_0: β_1=0).
- Both tests indicate that year is not a significant predictor of bird presence.

```
require(car)
Anova(binLogit)
```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: cbind(successes, trials - successes)
     LR Chisq Df Pr(>Chisq)
Year   2.7184  1     0.0992 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.5 Model Assessment

7.5.1 Graphical Assessment

As before, we can assess the fit of a model by plotting the observed proportions (y_i) vs the fitted values. The fitted values ($\hat{y}_i$) are obtained using the inverse link function and the estimated coefficients:

$$\hat{y}_i = \frac{e^{\hat{\eta}_i}}{1 + e^{\hat{\eta}_i}} = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1i}}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1i}}}$$

These are provided by `R` using the `fitted` function:

```
plot(sighting_rates$Props, fitted(binLogit),pch=16)
abline(0,1)
```

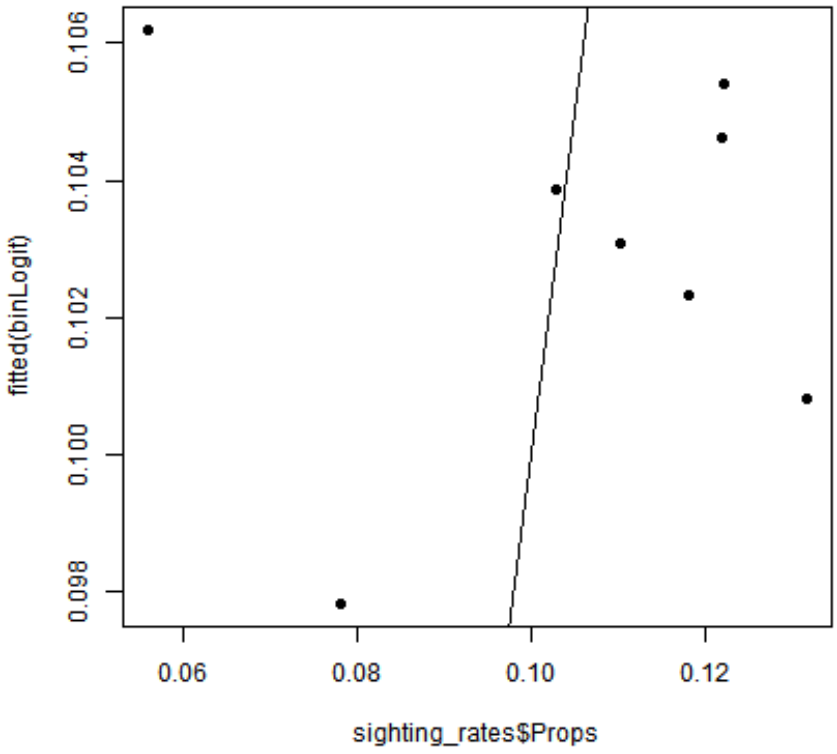


FIGURE 7.3 Fit plot for the logistic regression model for Year

In this case there is fairly poor agreement between the observed data and the fitted values.

7.5.2 Numerical assessment

For GLMs, there is no real analogue of the R^2 but we can use the value of the maximised log likelihood function (using the `logLik` function) instead; the higher this value, the better the fit of the model. To assess if our model is adequate using this value, we compare the maximised log-likelihood of our model with that of the best possible fitting model (using a GLM with same random component and link function). We define the best possible fitting model as a **saturated model** which has 1 parameter per observation, e. a saturated model has perfect fit and therefore the largest possible value of the maximised log-likelihood.

The deviance

The deviance, D, provides a measure of discrepancy between the fitted model and the saturated model (the model and the data).

$$\begin{aligned} D &= 2\left[l(\hat{\beta}_{sat},\phi) - l(\hat{\beta},\phi)\right] \\ &= 2[\text{maximised log-lik. in saturated model} - \text{maximised log-lik. in our model}] \end{aligned} \tag{7.4}$$

For any model, $D \geq 0$ and the smaller the D the better the fit of the model. For the saturated model, $D = 0$.

R gives us the deviance as `Residual Deviance` in the model output. We could fit a saturated model to these data by fitting `Year` as a factor (i.e. a coefficient is estimated for each level of `Year`), however this is no longer a summary of the data.

How do I know whether the deviance is small enough to be satisfied that my model is reasonable?

If a GLM with parameters, $\beta_0, \beta_1, ..., \beta_p$ is "correct" then: $D \sim \chi^2_{n-p-1}$ approximately if n is large.

The deviance is exactly chi-squared when Normal errors are assumed. However, this is only an approximation when fitting non-normal errors models and it should only be used as a **rough indication**. Thus for the test:

$$H_0 : \text{my model is correct}, \quad H_1 : \text{my model is NOT correct}$$

The test statistic, T, is $D \sim \chi^2_{n-p-1}$ under H_0. So the p-value for the test can be found using: `1-pchisq(d,n-p-1)`.

Note that computing D involves ϕ. So if ϕ is unknown I cannot use this result R provides as `Residual Deviance`. For the logistic model we have deviance=217.1026 (recall, we assume $\phi = 1$ under this model), $n = 8$ observations and 1 predictor (`Year`). This means that if our model is adequate, the deviance should look like it has come from a Chi-squared distribution with $n - p - 1 = 8 - 1 - 1 = 6$ degrees of freedom.

```
deviance(binLogit)
```

```
[1] 217.1026
```

```
1-pchisq(217.1026,6)
```

```
[1] 0
```

The p-value for the logit model is very small $p = 0$ and so we have compelling evidence against the hypothesis that the model is adequate; we are unhappy with our model.

```
ggplot(data.frame(x=0:25),aes(x)) + stat_function(fun= function(x) dchisq(x, df=6)) +
                                    ylab("Density") +
                                    ggtitle("Deviance distribution under H0")
```

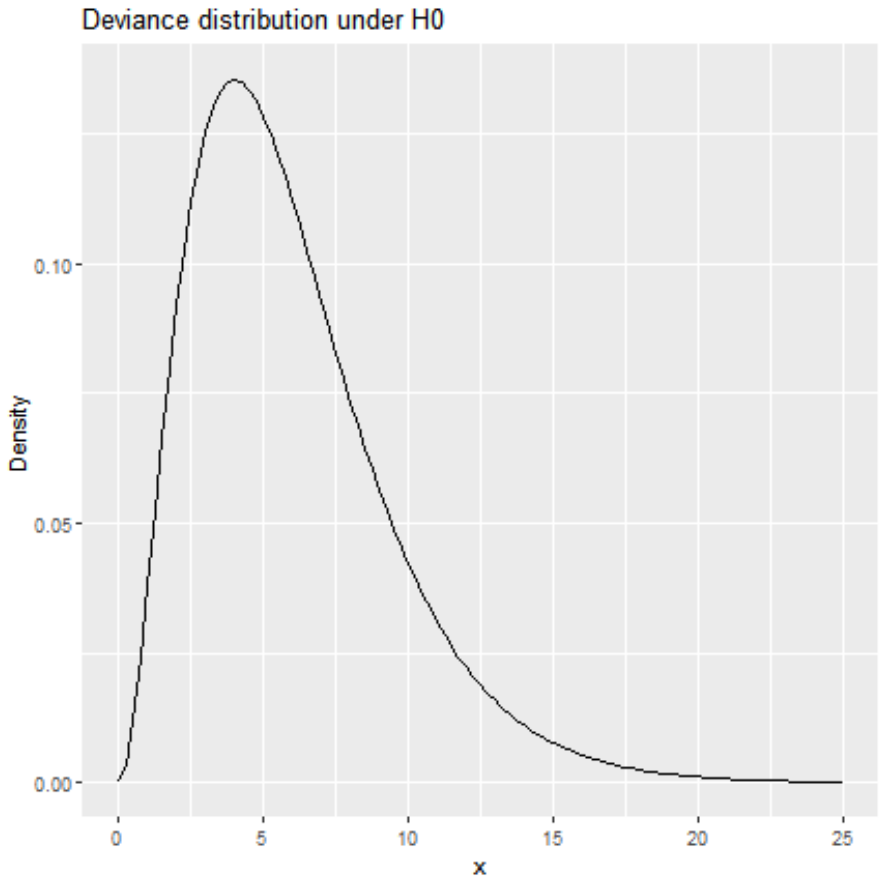


FIGURE 7.4 Distribution of the deviance under a correct model.

7.6 Model Assumptions

These are the same as for the Poisson model fitted earlier:

- Linearity on the link scale
- Assessing the mean-variance relationship
- Independence in model residuals

Linearity

Under the model we are assuming the relationships are linear on the link scale. We can use partial residual plots (in the `effects` library) to create these.

Mean-Variance relationship

Under the model we are assuming that the mean and the variance are np and $np(1-p)$ respectively. We can assess the mean variance relationship by plotting our fitted values against the pearsons residuals. We expect there to be no pattern if the mean-variance relationship is appropriate.

Independence

Under the model we are assuming that the model residuals are independent (i.e. once the patterns in the data have been accounted for, what is left over (the noise) is random). We can use an `acf` plot or runs test to assess this.

8

GLMs for binary data

Sometimes instead of wanting to model counts (per unit area) we are interested in modelling 'presence or absence' data. In this case it is presence/absence of birds but it might be a disease mapping study or something else. In these cases, we are best to use a different GLM-based approach in order to deal with the different nature of the data; the response data are now either binary (0's or 1's) or proportions (and are bounded by 0 and 1).

In this chapter we are going to model the probability of seeing one or more animals in this area as a function of the available covariates using Generalized Linear Models (GLMs)

8.1 Research Questions

We will look at three main research questions:

1. Is there any evidence that animal presence/absence differs across phases? If so, how so?
2. Can we predict the probability of seeing animals in this area using the covariates available? If so, which covariates are best at predicting sightings?
3. Does the presence/absence distribution appear to have changed across phases?

8.2 Exploratory data analysis

Trained observers recorded the presence or absence of birds at each spatial location; when birds were seen from the plane the variable `pres=1` and when birds

were absent pres=0. Summary statistics for this variable across phase can be found using the code below and are illustrated in Figure @ref{fig:barplotRes}

```
# for the entire period
table(df$Pres)
```

```
    0     1
28266  3236
```

```
# sighting data per phase
table(df$Pres, df$Phase)
```

```
        A     B     C
  0 10335 12495  5436
  1  1143  1633   460
```

```
fr<-table(df$Pres, df$Phase)
p<-rep(c("Absence", "Presence"),each=ncol(fr))
fr<-data.frame(rep(colnames(fr),2), as.vector(t(fr)),p)
colnames(fr)<-c("Phase","Fr", "Pres")

ggplot(fr, aes(x= Phase, y = Fr, fill=Pres))+
  geom_bar(stat="identity",position = position_stack(reverse = TRUE)) +
  geom_text(aes(y=Fr, label=Fr), vjust=1.2, color="white", size=3, position = position_sta
  ylab("Frequency")
```

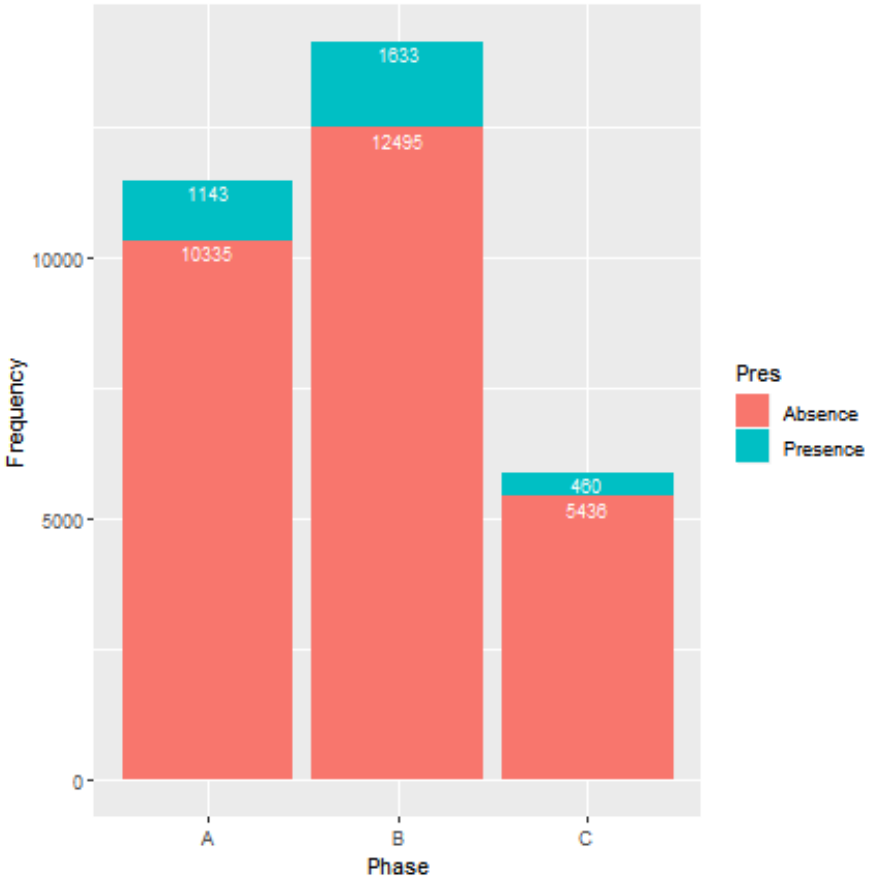


FIGURE 8.1 Barplot showing the presence absence data across the survey period

What can we conclude?

- There were many more absence than presence observations and survey effort was highest in phase B,
- however it is difficult to determine from this graphic if the sighting probability has genuinely changed across phases.
- Even if there were no genuine changes across phase, the observed values will never be exactly the same and so we need to consider the uncertainty in these estimates for each phase.

8.3 Model Specification

For illustration, we will model the probability of seeing one or more animals (i.e. successes) as a function of phase alone at first, and followed by a Binomial-based Generalized Linear Model using all of the available covariates. These models:

- allow the probability of sighting a bird to change nonlinearly with covariate values
- respect the fact that probabilities/proportions are naturally bounded by zero and one (i.e. $0 \leq p \leq 1$) and so only return predictions inside this range.

We are going to use a Bernoulli distribution with probability of "success"" p_{it}. We are going to allow p_{it} to vary across one or more covariates using a GLM with a "logit link function" (a "logistic regression model"). This model with `Phase` as a covariate can be written in terms of the response:

$$p_{it} = \frac{e^{\eta_{it}=\beta_0+\beta_1 x_{it}+\beta_2 x_{2it}}}{1+e^{\eta_i=\beta_0+\beta_1 x_{1it}+\beta_2 x_{2it}}}$$

or in terms of the link function $g(p_{it})$:

$$g(p_{it}) = \log\left(\frac{p_{it}}{1-p_{it}}\right) = \eta_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it}$$

8.4 Model fitting

We can fit Binomial/Bernoulli-based GLMs for proportional/binary data using either a logit, probit or complementary log-log link using the Maximum Likelihood method (The mean and variance assumed under this model (for all link functions) are $\mu = Np$ and $V(\mu) = \phi Np(1-p) = Np(1-p)$ respectively (i.e. $\phi = 1$)). For example we can fit these GLMs in `R` using each of the 3 link functions fitted with `Phase` alone and also with other covariates.

```
# Phase
binLogitPhase<- glm(Pres ~  Phase , data=df,
                    family=binomial)

binProbitPhase<- glm(Pres ~  Phase , data=df,
                     family=binomial(link="probit"))
```

```
binCloglogPhase<- glm(Pres ~  Phase , data=df,
                      family=binomial(link="cloglog"))

# Full
binLogit<- glm(Pres ~ XPos + YPos + DistCoast + Depth + FMonth +
                 Phase + XPos:Phase + YPos:Phase, data=df,
               family=binomial)

binProbit<- glm(Pres ~ XPos + YPos + DistCoast + Depth + FMonth +
                  Phase + XPos:Phase + YPos:Phase, data=df,
                family=binomial(link="probit"))

binCloglog<- glm(Pres ~ XPos + YPos + DistCoast + Depth + FMonth +
                   Phase + XPos:Phase + YPos:Phase, data=df,
                 family=binomial(link="cloglog"))
```

Let's have a look at the fit of the logit model with Phase as a covariate:

```
summary(binLogitPhase)
```

```
Call:
glm(formula = Pres ~ Phase, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4956  -0.4956  -0.4580  -0.4031   2.2587

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.20188    0.03117 -70.638  < 2e-16 ***
PhaseB       0.16697    0.04079   4.093 4.26e-05 ***
PhaseC      -0.26769    0.05770  -4.639 3.50e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 20856  on 31501  degrees of freedom
Residual deviance: 20788  on 31499  degrees of freedom
AIC: 20794

Number of Fisher Scoring iterations: 5
```

8.5 Parameter interpretation

As before, parameter estimates for logistic models (with a logit link) can be interpreted in terms of the odds of success. In this case a "success" is a visit where one or more birds were seen. Using the model with phase alone, $\hat{\beta}_0 = -2.20188$, $\hat{\beta}_1 = 0.16697$ and $\hat{\beta}_2 = -0.26769$ and we'll use the following to find the odds of presence versus absence in phases A, B and C:

$$\widehat{\left(\frac{p_{it}}{1-p_{it}}\right)} = e^{\hat{\beta}_0+\hat{\beta}_1 x_{1it}+\hat{\beta}_2 x_{2it}}$$

The estimated odds of presence vs absence in:

- phase A is $\exp(-2.20188) = 0.110595$
- phase B is $\exp(-2.20188 + 0.16697) = 0.13069$
- hase C is $\exp(-2.20188 - 0.26769) = 0.08462$

So, as we move from phase A to phase B, the odds of presence vs absence are estimated to increase by a factor $e^{\hat{\beta}_1} = \exp(0.16697) = 1.181719$ and $0.110595 \times 1.181719 = 0.1306922$.

8.6 Parameter inference

Confidence interval for parameters

As for the proportional Binomial GLM, we use a z-multiplier to construct confidence intervals.

$$\text{estimate} \quad \pm z - \text{multiplier} \times \text{standard error}$$

Testing for non-zero covariate relationships

We also test for no relationship between each covariate and the response in the standard way using a χ^2 test. Thus, large test statistics (and small p-values) give compelling evidence for a non-zero relationship with the response.

8.7 Model Selection

A comparison between models with and without Phase (when Phase is the only covariate) tells us the fit is significantly improved by retaining Phase in the model. This metric also tells us all covariates are significant in the model. This suggests some sort of redistribution in the X-covariate and Y-covariate direction.

```
require(car)
Anova(binLogitPhase)
```

```
Analysis of Deviance Table (Type II tests)

Response: Pres
      LR Chisq Df Pr(>Chisq)
Phase   67.926  2  1.778e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(binLogit)
```

```
Analysis of Deviance Table (Type II tests)

Response: Pres
           LR Chisq Df Pr(>Chisq)
XPos         170.73  1  < 2.2e-16 ***
YPos         804.81  1  < 2.2e-16 ***
DistCoast     15.95  1  6.490e-05 ***
Depth       1221.18  1  < 2.2e-16 ***
FMonth       425.47  3  < 2.2e-16 ***
Phase         56.84  2  4.549e-13 ***
XPos:Phase    67.18  2  2.576e-15 ***
YPos:Phase    10.69  2   0.004767 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Information criteria based selection

AIC/AICc/BIC scores can help choose between models with different link functions and/or model covariates. We see that for a model with phase alone there is no measurable difference between the link functions (phase is a factor covariate and so does not rely on the linear relationship on the link scale), but a clear difference emerges for the larger model and the logit link is clearly preferred.

```
AIC(binLogitPhase,binProbitPhase,binCloglogPhase)
```

```
                df      AIC
binLogitPhase    3 20794.07
binProbitPhase   3 20794.07
binCloglogPhase  3 20794.07
```

```
AIC(binLogit,binProbit,binCloglog)
```

```
           df      AIC
binLogit   14 18036.73
binProbit  14 18045.38
binCloglog 14 18068.29
```

AICc from the dredge function also confirms that the full model should be selected.

```
require(MuMIn)
options(na.action="na.fail")
head(dredge(binLogit))
```

```
Global model call: glm(formula = Pres ~ XPos + YPos + DistCoast + Depth + FMonth +
    Phase + XPos:Phase + YPos:Phase, family = binomial, data = df)
---
Model selection table
    (Int)     Dpt      DsC FMn Phs       XPs        YPs Phs:XPs Phs:YPs df
256  1154 -0.2408 -0.03429   +   + 5.432e-06 -0.0001913       +       + 14
128  1074 -0.2388 -0.03546   +   + 8.286e-06 -0.0001784       +         12
254  1097 -0.2553            +   + 5.868e-06 -0.0001821       +       + 13
126  1012 -0.2537            +   + 8.894e-06 -0.0001683       +         11
64   1051 -0.2383 -0.03354   +   + 2.376e-05 -0.0001764                 10
192  1014 -0.2383 -0.03358   +   + 2.376e-05 -0.0001703               + 12
       logLik    AICc delta weight
256 -9004.365 18036.7  0.00  0.965
128 -9009.711 18043.4  6.69  0.034
254 -9012.342 18050.7 13.95  0.001
126 -9018.265 18058.5 21.79  0.000
64  -9039.104 18098.2 61.47  0.000
192 -9037.958 18099.9 63.18  0.000
Models ranked by AICc(x)
```

8.8 Predictive power

We could assess the fit of a model by plotting the observed values (y_{it}) vs the fitted values ($\hat{y}_{it} = n_{it}\hat{p}_{it} = \hat{p}_{it}$); but this is relatively uninformative since the response values are binary and the fitted values are probabilities. When the response values are proportions, this comparison is of more use.

8.8.1 An R-squared metric

An R^2 measure for binary data can be used to assess model fit[1]

$$R^2 = \frac{1 - \exp((D - D_{null})/N)}{1 - \exp(-D_{null}/N)}$$

where N is the total number of binary observations and D is the deviance of the fitted model; the deviance is a measure of fit which compares our model with a "saturated" model. D_{null} is the deviance of the null (i.e. intercept only) model. This value is constrained to lie within 0 and 1.

```
# r2 of the Fitted model
(1-exp((binLogit$dev-binLogit$null)/31502))/
  (1-exp(-binLogit$null/31502))
```

```
[1] 0.1784754
```

Is this too low? To address this question we simulate from the fitted model and calculate R^2 for the simulated data.

```
set.seed(196)
df$Presfake<- rbinom(n=nrow(df), size=1,
                     prob=fitted(binLogit))
fakelogitfit<- update(binLogit, Presfake ~ . , data=df)

# r2 of the simulated model
(1-exp((fakelogitfit$dev-fakelogitfit$null)/31502))/
  (1-exp(-fakelogitfit$null/31502))
```

```
[1] 0.1822362
```

[1] see Naglekerke, N. 1991.A note on the general definition of the coefficient of determination. *Biometrika*, 78, 691–692.

8.9 The Confusion Matrix

Assessing model performance, in absolute terms, is often difficult for binary data because the values returned by a model are probabilities between zero and one, while the input data are binary. This process generates a "confusion matrix"" which also lets us examine the false positive rate and false negative rate under the method/model combination.

Table Confusion matrix

Observed values	0	1
Predicted 0	True Negative (A)	False Negative (B)
Values 1	False Positive (C)	True Positive (C)

A **threshold value** must be chosen to return predictions to binary (0/1) data. This more easily enables comparison with the observed data. A common choice is the mean of the fitted values[2].

```
# Confusion Matrix
val<-mean(fitted(binLogit))
resp<-ifelse(fitted(binLogit)>val,1,0)
table(resp,df$Pres)
```

```
resp     0     1
   0 18233   694
   1 10033  2542
```

```
(18233+2542)/31502
```

```
[1] 0.6594819
```

65.9% of the response values are correctly classified (using the mean of the fitted values as the 0/1 threshold).

- The false negative rate (the response is a success but the model predicts a failure) is $694/31502 = 2.2\%$.
- The false positive rate (the response is a failure but the model predicts a success) is $10033/31502 = 31.8\%$

[2] C. Liu, P. Berry, T. Dawson, and R. Pearson. Selecting thresholds of occurrence in the prediction of species distributions. Ecography, 28:385-393, 2005

Receiver Operating Characteristic (ROC) Curves

ROC curves may be used to assess the predictive nature of the model at all threshold values.

```
require(pROC)
df$Prob<- predict(binLogit,type=c("response"))
g <- roc(Pres ~ Prob, data = df)
(best<-coords(g, "best",transpose = FALSE))
```

```
     threshold specificity sensitivity
best 0.1077926   0.6635534   0.7744129
```

```
df_ROC<-data.frame(coords(g, seq(0, 1, 0.01),transpose = FALSE))
p <- ggplot(df_ROC)
p <- p + geom_line(aes(1-specificity, sensitivity, colour=threshold), size=3) + theme_bw()
p + geom_abline(intercept=0, slope=1) + xlab('1-Specificity (False Positive Rate)') +
    ylab('Sensitivity (True Positive Rate)') + geom_hline(yintercept=as.numeric(best[3]),
```

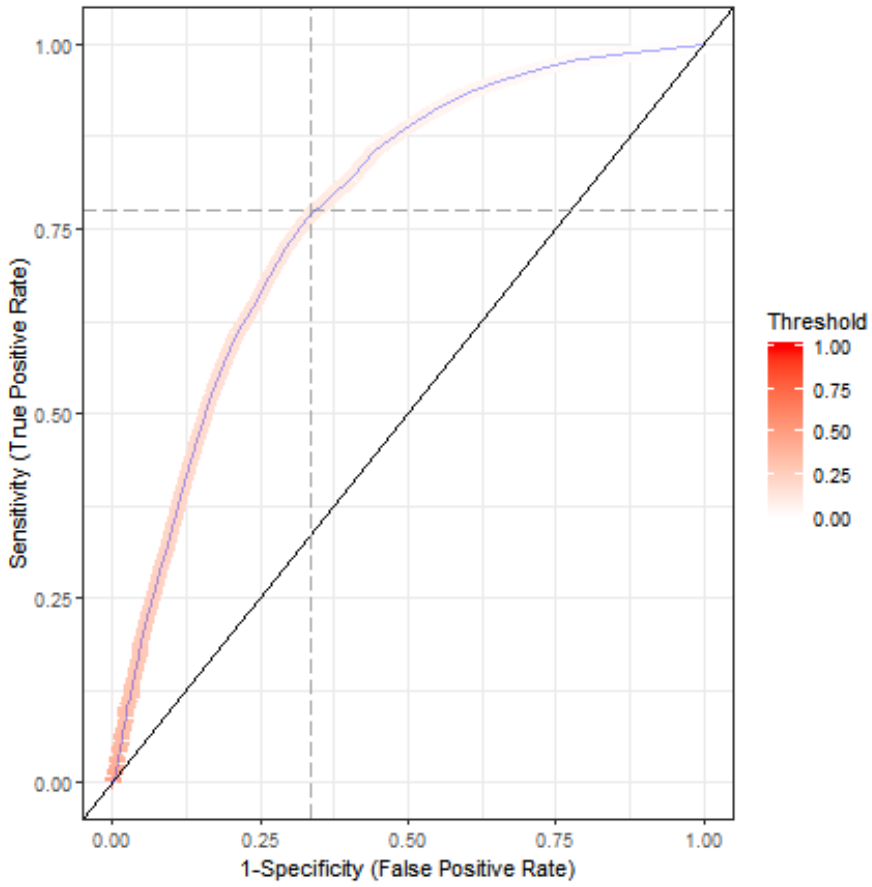


FIGURE 8.2 ROC curve for the binLogit model

Sensitivity is the proportion of correctly classified ones (D/(B+D) and *Specificity* is the proportion of correctly classified zeros (A/(A+C) (see Table 8.9). These are also known as the true positive rate and (1-false positive rate). The "best" threshold is considered to be that which maximises the true positives and minimises the false positives. This can be found by the location on the curve closest to the top left corner of the ROC plot. In this case, it is at a threshold value of ~ 0.1.

Create a confusion matrix based on the "best"" threshold. Here we use the `caret` package:

```
df$Prediction=ifelse(df$Prob>=best[[1]],1,0)
require(caret)
confusionMatrix(as.factor(df$Prediction),as.factor(df$Pres))
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 18756   730
         1  9510  2506

               Accuracy : 0.6749
                 95% CI : (0.6697, 0.6801)
    No Information Rate : 0.8973
    P-Value [Acc > NIR] : 1

                  Kappa : 0.199

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6636
            Specificity : 0.7744
         Pos Pred Value : 0.9625
         Neg Pred Value : 0.2086
             Prevalence : 0.8973
         Detection Rate : 0.5954
   Detection Prevalence : 0.6186
      Balanced Accuracy : 0.7190

       'Positive' Class : 0
```

Sensitivity and Specificity are described above. Accuracy = (A+D)/(A+B+C+D). 67.4% of the response values are correctly classified (this was 65.9% when using the mean of the fitted values as the 0/1 threshold).

8.10 Model diagnostics

8.10.1 Linearity on the link scale

Under the model we are assuming the relationships are linear on the link scale (and thus nonlinear on the response scale), however this is almost impossible to check in practice. Partial residual plots are of little use for binary data to assess the validity of this assumption since they necessarily have a pattern. This pattern occurs because we have binary response data and the fitted probabilities

lie between zero and one. Partial plots (using the `effects` library for example) are of more use when the input data are proportions and thus we are better able to map the response values against the fitted values from the model (which then both lie between zero and one).

8.10.2 Assessing the mean variance relationship

For binary data, it is not appropriate to model overdispersion since the number of trials in each case is equal to one[3]. For this reason, there is no reason to use `family=quasibinomial` during GLM fitting. In contrast this would be a necessary check/comparison when the response data are proportions, rather than binary values (and the number of trials in each case > 1). When we compare the fitted values from our model with the raw residuals (as for previous models) we see that there is strong pattern in the residuals (Figure 8.3). The pattern stems from the binary nature of the response; the values returned by a model are probabilities between zero and one, while the input data are binary. This renders this diagnostic ineffective.

```
par(mfrow=c(1,2))
plot(fitted(binLogit), residuals(binLogit,type="response"), xlab='Fitted Values', ylab='Ra
plot(fitted(binLogit), residuals(binLogit,type="deviance"), xlab='Fitted Values', ylab='De
```

[3] Skrondal & Rabe-Hesketh (2007) Redundant Overdispersion Parameters in Multilevel Models for Categorical Responses. *Journal of Educational and Behavioral Statistics*

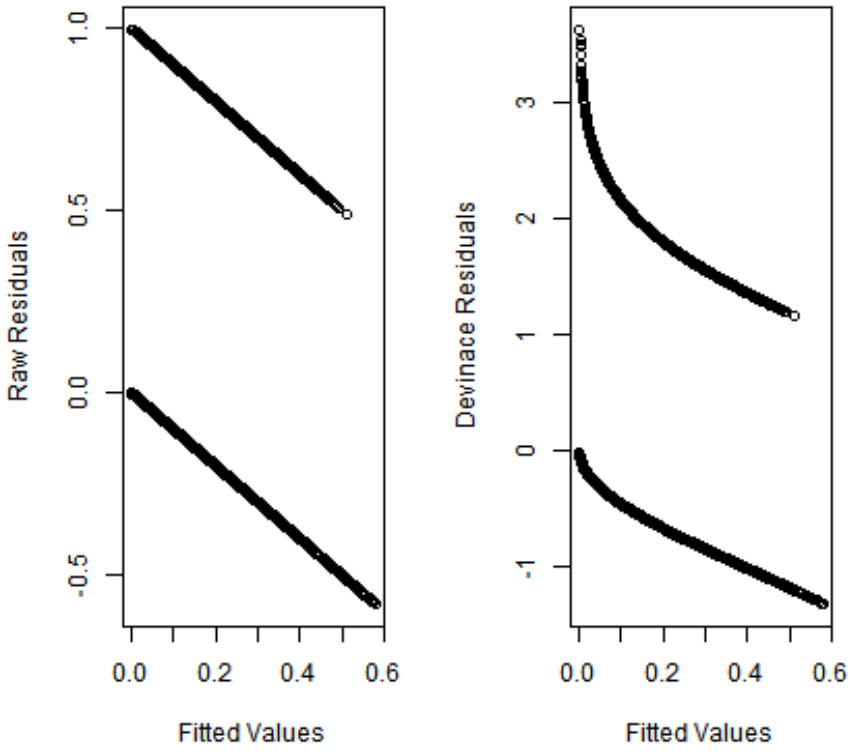


FIGURE 8.3 Fitted values vs deviance residuals for the binary logit model.

You would expect the raw residuals to give some pattern (owing to the mean-variance relationship) but the Pearsons residuals to show random scatter.The next figures give an example of what you might expect when the input data is binomial.

```
knitr::include_graphics('figures/Binomial_RawResid.png')
```

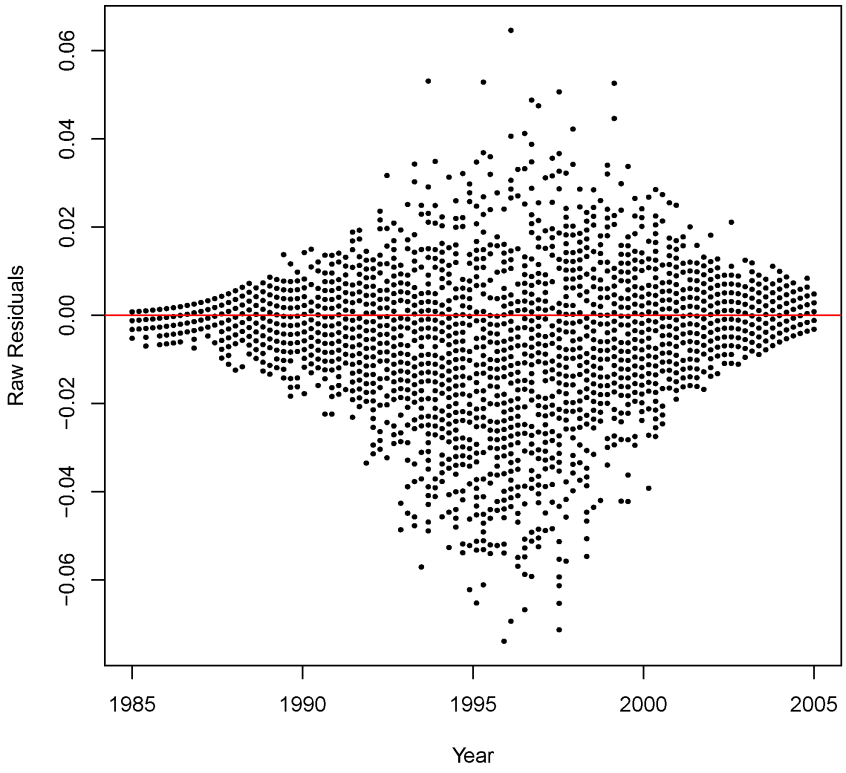


FIGURE 8.4 Fitted values vs raw residuals (upper plot) and vs Pearsons Residuals (lower plot) for a binomial logistic model.

```
knitr::include_graphics('figures/Binomial_PearsonsResid.png')
```

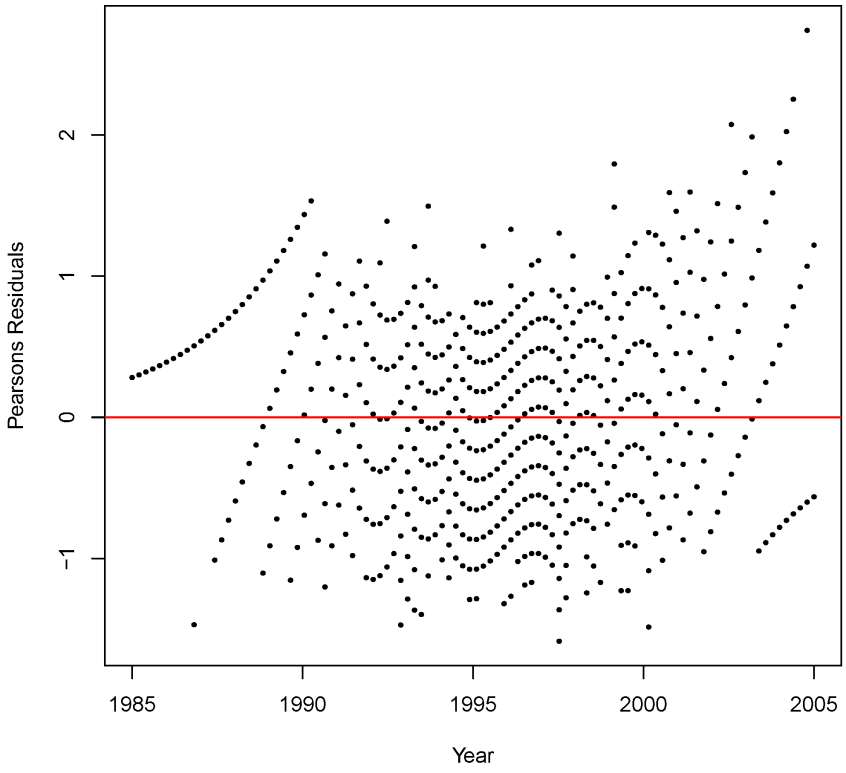


FIGURE 8.5 Fitted values vs raw residuals (upper plot) and vs Pearsons Residuals (lower plot) for a binomial logistic model.

8.10.3 Non-independence in model residuals

There is compelling evidence for some space/time-based correlation in model residuals (left-hand plot, Figure 8.6), and this is in stark contrast to data generated from a uncorrelated Binomial process with the same means (right-hand plot, Figure 8.6). This means we should treat all *p*-values based on our current model with caution and due to this positive correlation, these might well be too small.

```
par(mfrow=c(1,2))
acf(residuals(binLogit, type="pearson"), main="Actual model")
acf(residuals(fakelogitfit, type="pearson"), main="Correct model")
```

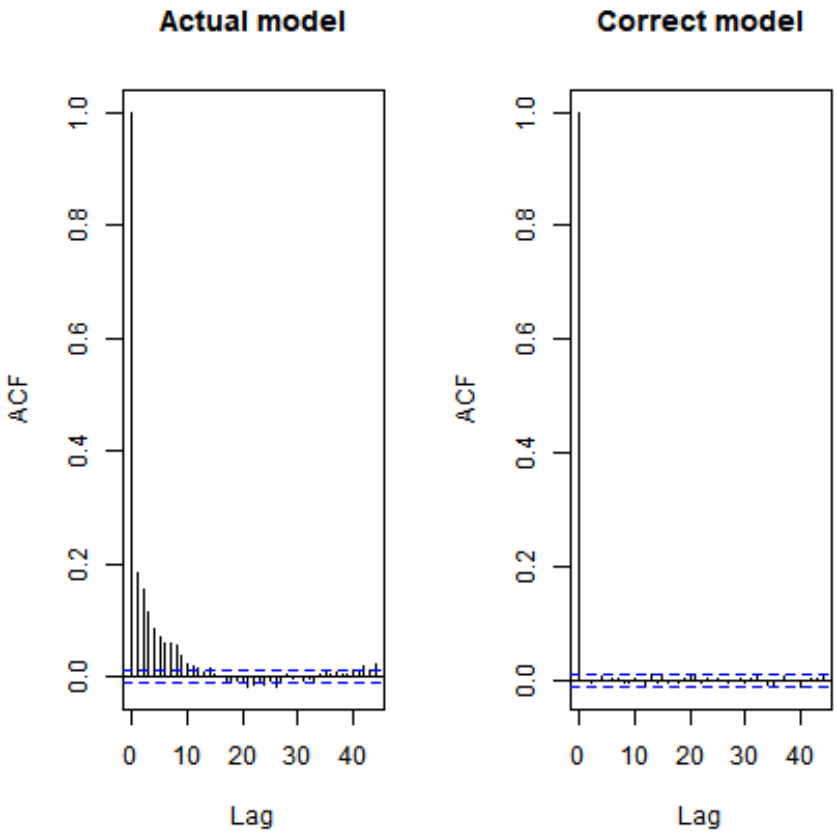


FIGURE 8.6 acf

8.10.4 Modelling the spatial element

There seems to be some minor changes in the sighting probabilities across phases (Figures 8.7–8.9 and while the overall fit appears to be good, there are some surface features which are not well captured by the model.

```
require(fields)
quilt.plot(predictionData$XPos[predictionData$Phase=="A"],
           predictionData$YPos[predictionData$Phase=="A"],
           predBinLogit[predictionData$Phase=="A"], ncol=50, nrow=60, main="Phase A")
```

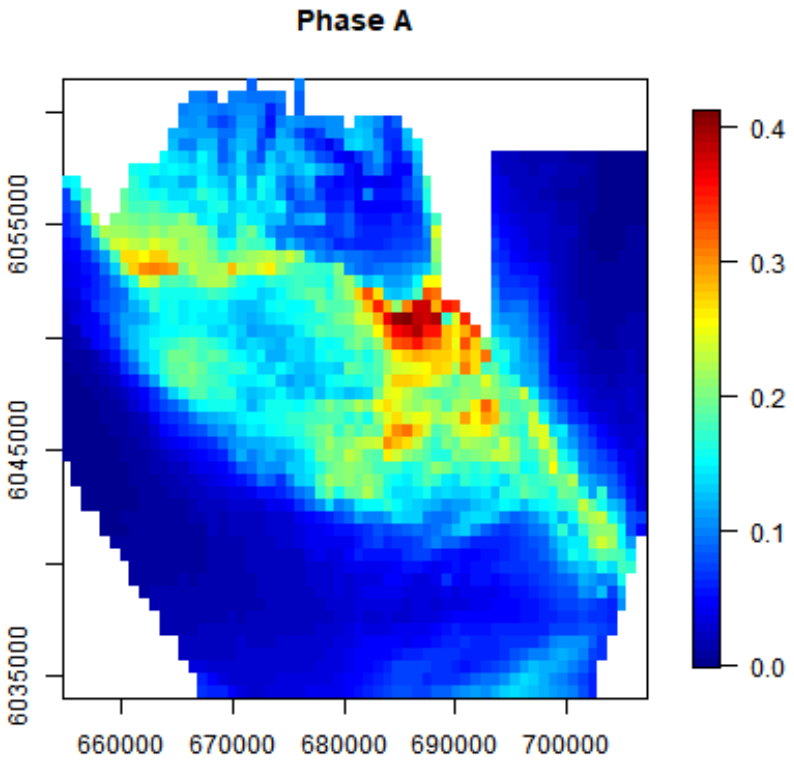


FIGURE 8.7 Fitted surface for the binLogit model - Phase A.

```
quilt.plot(predictionData$XPos[predictionData$Phase=="B"],
           predictionData$YPos[predictionData$Phase=="B"],
           predBinLogit[predictionData$Phase=="B"], ncol=50, nrow=60, main="Phase B")
```

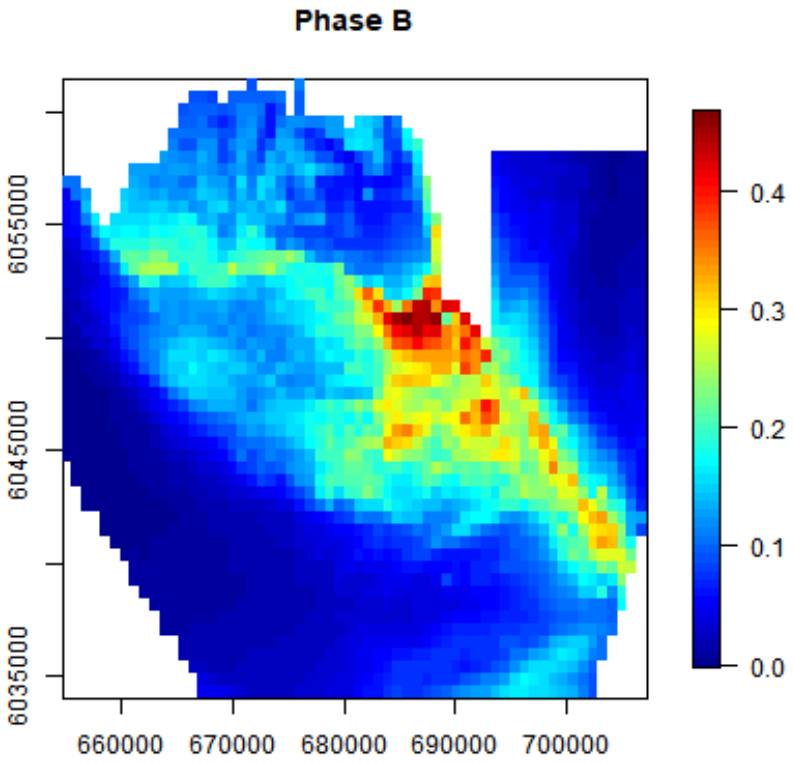


FIGURE 8.8 Fitted surface for the binLogit model - Phase B.

```
quilt.plot(predictionData$XPos[predictionData$Phase=="C"],
           predictionData$YPos[predictionData$Phase=="C"],
           predBinLogit[predictionData$Phase=="C"], ncol=50, nrow=60, main="Phase C")
```

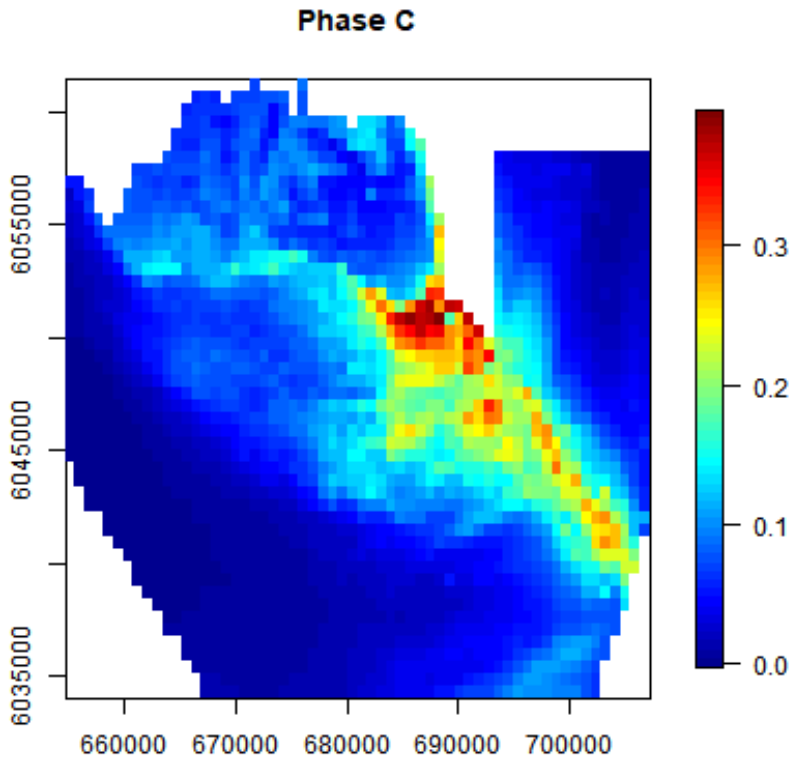


FIGURE 8.9 Fitted surface for the binLogit model - Phase A.

The Pearson's residuals show systematic patterns which are clearly absent when the fitted model is assumed to be correct and data are generated from this process (Figures 8.10–8.12). This gives us reason for concern since we are interested in phase-based surface changes in particular, as one of the analysis objectives.

```
par(mfrow=c(1,2),mar = c(5, 4, 4, 1) + 0.1)

# Residuals Fitted Model Phase A
zr<-range(-1,2)
quilt.plot(df$XPos[df$Phase=="A"], df$YPos[df$Phase=="A"],
           residuals(binLogit, type="pearson")[df$Phase=="A"], ncol=25, nrow=25, main="Pha
           zlim=zr)
# Residuals simulated model Phase A
```

```
quilt.plot(df$XPos[df$Phase=="A"], df$YPos[df$Phase=="A"],
           residuals(fakelogitfit, type="pearson")[df$Phase=="A"], ncol=25, nrow=25, main=
           zlim=zr)
```

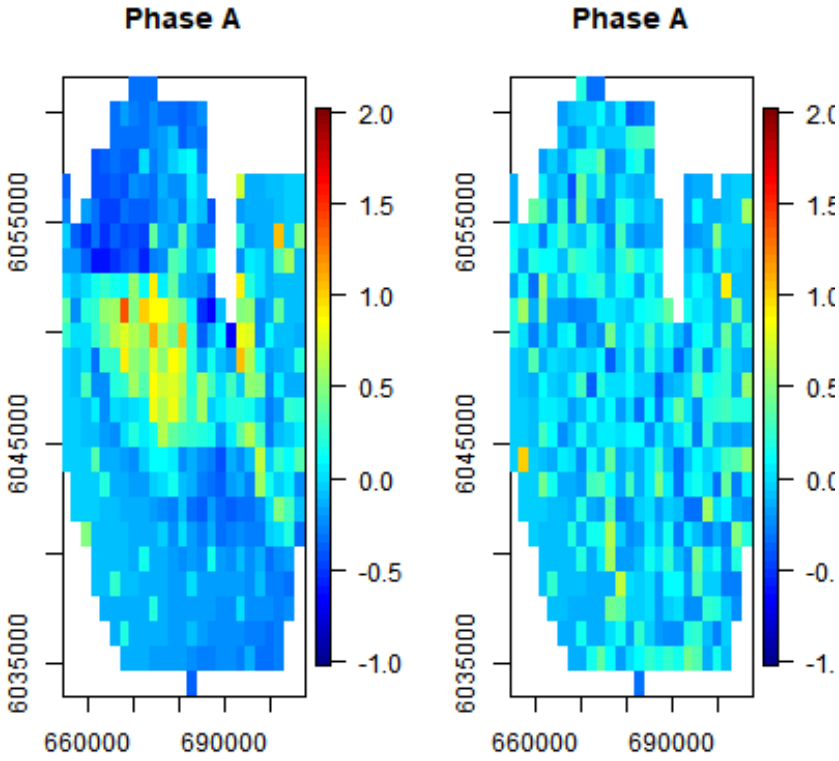


FIGURE 8.10 Residuals for the current model (left) and randomly sampled correct model (right) - Phase A.

```
par(mfrow=c(1,2),mar = c(5, 4, 4, 1) + 0.1)

quilt.plot(df$XPos[df$Phase=="B"], df$YPos[df$Phase=="B"], residuals(binLogit, type="pears

quilt.plot(df$XPos[df$Phase=="B"], df$YPos[df$Phase=="B"], residuals(fakelogitfit, type="p
```

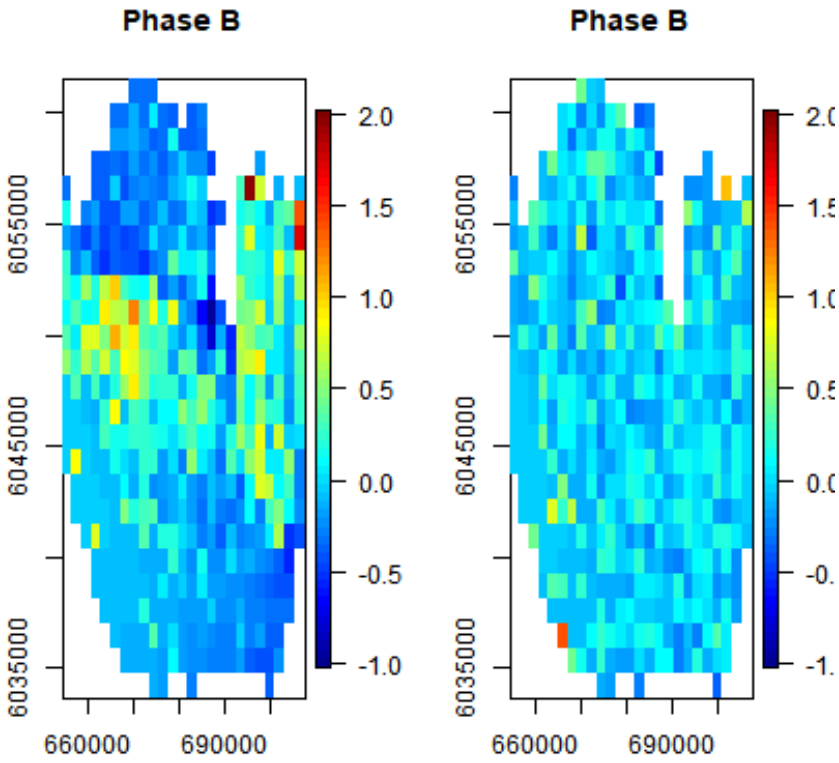


FIGURE 8.11 Residuals for the current model (left) and randomly sampled correct model (right) - Phase B.

```
par(mfrow=c(1,2),mar = c(5, 4, 4, 1) + 0.1)

quilt.plot(df$XPos[df$Phase=="C"], df$YPos[df$Phase=="C"], residuals(binLogit, type="pears

quilt.plot(df$XPos[df$Phase=="C"], df$YPos[df$Phase=="C"], residuals(fakelogitfit, type="p
```

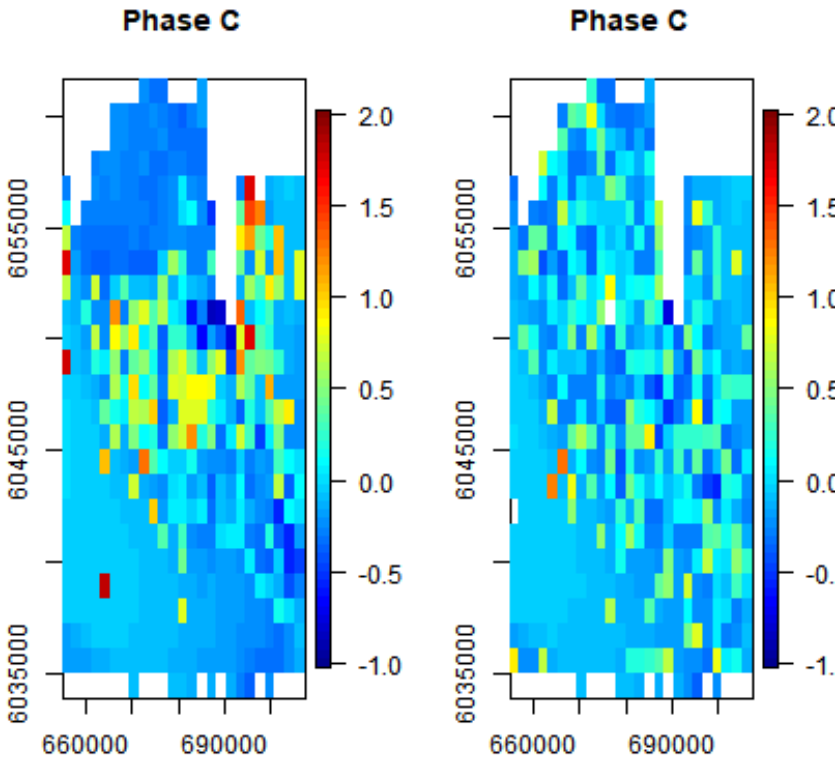


FIGURE 8.12 Residuals for the current model (left) and randomly sampled correct model (right) - Phase C.

8.11 Revisiting the research questions

- Based on model results to date, there appears to be differences across phases in average sighting probabilities and also a distributional shift in sighting probabilities across phases.
- There is also evidence of month-to-month and depth-related changes in sighting probabilities.
- Overall, the model appears to fit well. The mean-variance relationship is characteristic of what we would expect under the model and the quality of model fit

(using the R^2 and the correct classification rate) is also what we would expect under a correct model of this type.

- While overall model fit appears to be good, we can see the model for the spatial element may be missing some local surface features which are tricky/impossible to capture using GLMs - even using interaction effects.
- We also have some concerns about model inference; in particular the violation of residual independence. At this stage we might be falsely concluding some of these effects are important and any confidence intervals based on model predictions might well be too small.
- This can be a major issue if there is interest in quantifying the practical consequences of any phase-based differences, using model p-values and/or the sets of plausible values (via confidence intervals).
- For this reason, models which permit residual correlation (e.g **GEEs**) would be a suitable alternative in this case to adjust model results for residual correlation within transects.

Note

Fitting Binomial GLMs and Bernoulli GLMs would give the same results if the response and the covariates are the same. Compare

with

```
Fit1<-glm(cbind(successes, trials - successes)~ Year, family = binomial, data = sighting_r
summary(Fit1)
```

```
Call:
glm(formula = cbind(successes, trials - successes) ~ Year, family = binomial,
    data = sighting_rates)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-11.113   -1.450    2.277    3.258    5.657

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 14.513016  10.145223   1.431    0.153
Year        -0.008322   0.005061  -1.644    0.100

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 219.82  on 7  degrees of freedom
Residual deviance: 217.10  on 6  degrees of freedom
AIC: 282.69

Number of Fisher Scoring iterations: 4
```

with

```
Fit2<-glm(Pres~Year,family = binomial, data=df)
summary(Fit2)
```

```
Call:
glm(formula = Pres ~ Year, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4739  -0.4702  -0.4665  -0.4537   2.1562

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 14.513012  10.144431   1.431    0.153
Year        -0.008322   0.005061  -1.644    0.100

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 20856  on 31501  degrees of freedom
Residual deviance: 20853  on 31500  degrees of freedom
AIC: 20857

Number of Fisher Scoring iterations: 4
```

9

Modelling multiple outcome data: Multinomial models for nominal categorical data

Multinomial models (also called *discrete choice models* or *multicategory models*) are extensions of binomial models to accommodate categorical response types with more than two levels. For example, a food preference survey might ask a sample of people which is their favourite delicacy out of haggis, muktuk or smalahove.

9.1 Background

9.1.1 Multinomial distribution

Let Y_i, the response of the ith subject, be a random variable that takes one of a finite number of values, 1, 2, ..., J. In the above example, it could be 1 for haggis, 2 for muktuk and 3 for smalahove. Assume each response is independent and identically distributed, with $\Pr(Y_i = j) = p_j$. (Note that $\sum_{j=1}^{J} p_{ij} = 1$.) Let $n_1, \ldots, n_J$ be the number of responses in each category $1, \ldots, J$ respectively, and let $n = \sum_{j=1}^{J} n_j$.

$$(n_1, \ldots, n_J) \sim \mathrm{Mn}\,(n, p_1, \ldots, p_J)$$

Note when $J = 2$ the above reduces to a binomial distribution.

9.1.2 Aggregated vs. disaggregated categorical data

There are two ways to present categorical (e.g. multinomial and binomial) data for modelling: aggregated and disaggregated. These are also called grouped and ungrouped. With **aggregated data**, all responses with the same covariate are grouped together and the data are the number of responses in each category.

For example, if the above food preference survey took place in multiple towns, the data could be in the form:

haggis	muktuk	smalahove	town
27	3	0	St Andrews
2	37	1	Barrow
17	5	18	Voss
...	...	...	...

Note, for binomial data, aggregated data are often given as count in one category and total count - e.g., collapsing the above into 2 categories:

haggis	total	town
27	30	St Andrews
2	40	Barrow
17	40	Voss
...	...	...

With **disaggregated data**, there is one record for each individual sample, and the data are the value of the response (either as a number of a class).For example:

food	town
haggis	St Andrews
haggis	St Andrews
muktuk	St Andrews
haggis	Barrow
smalahove	Barrow
haggis	St Andrews
muktuk	Voss
smalahove	Voss

Results from multinomial (and binomial) analyses are the same no matter which form you use (in effect they are just different ways of presenting the data). However if you have one or more explanatory variables that are continuous (e.g., respondent's weight in the above example), the disaggregated form must be used.

9.1.3 Nominal vs Ordinal responses

In this chapter we deal with **nominal** multinomial data, where the response value categories have no natural ordering. For example, in the food preference survey, the ordering of the three options is arbitrary.

Contrast this with **ordinal** multinomial data, where the order matters – e.g., a

food survey might ask you to express your fondness for haggis suppers by ticking one of: detest, dislike, don't mind, like, love it. Here, the order matters – each category is, in some sense, "more" than the previous one.

We deal with ordinal data in the next chapter.

9.2 Scottish Independence Referendum data

For this chapter, we'll use data from the 2014 Scottish independence referendum[1]. Our goal is to explain the regional patterns of voting. The response variable is the number of registered voters who voted yes, no or neither (i.e., who didn't vote or whose vote was spoiled) in each of the 32 Scottish council areas.

Question 3

Is this aggregated or disaggregated data?

Show Answer on P??

There are many potential explanatory variables, but here we'll focus on two: + `scottish` – proportion of voters who were born in Scotland + `income` – median weekly income (in pounds)

Let's look at the data first:

```
head(indyref)
```

```
                council   yes     no neither income scottish  votes
1              Aberdeen 59390  84094   32032  547.8    75.04 175516
2         Aberdeenshire 71337 108606   26437  572.3    80.51 206380
3                Angus 35044  45192   13236  478.7    85.87  93472
4       Argyll And Bute 26324  37143    8483  463.0    76.06  71950
5      Clackmannanshire 16350  19036    4558  474.8    86.37  39944
6 Dumfries And Galloway 36614  70039   15250  443.0    77.11 121903
       pyes       pno  pneither
1 0.3383737 0.4791244 0.1825019
2 0.3456585 0.5262429 0.1280987
3 0.3749144 0.4834817 0.1416039
4 0.3658652 0.5162335 0.1179013
```

[1]Data from [here](http://blogs.ft.com/ftdata/2014/09/19/scottish-referendum-who-voted-which-way/), supplemented by count data from the BBC web site

```
5 0.4093231 0.4765672 0.1141098
6 0.3003536 0.5745470 0.1250995
```

Income:

```
p<-list()
p[[1]]<-qplot(income,pyes,data=indyref)
p[[2]]<-qplot(income,pno, data=indyref)
p[[3]]<-qplot(income,pneither, data=indyref)

grid.arrange(grobs=p,nrow=1)
```

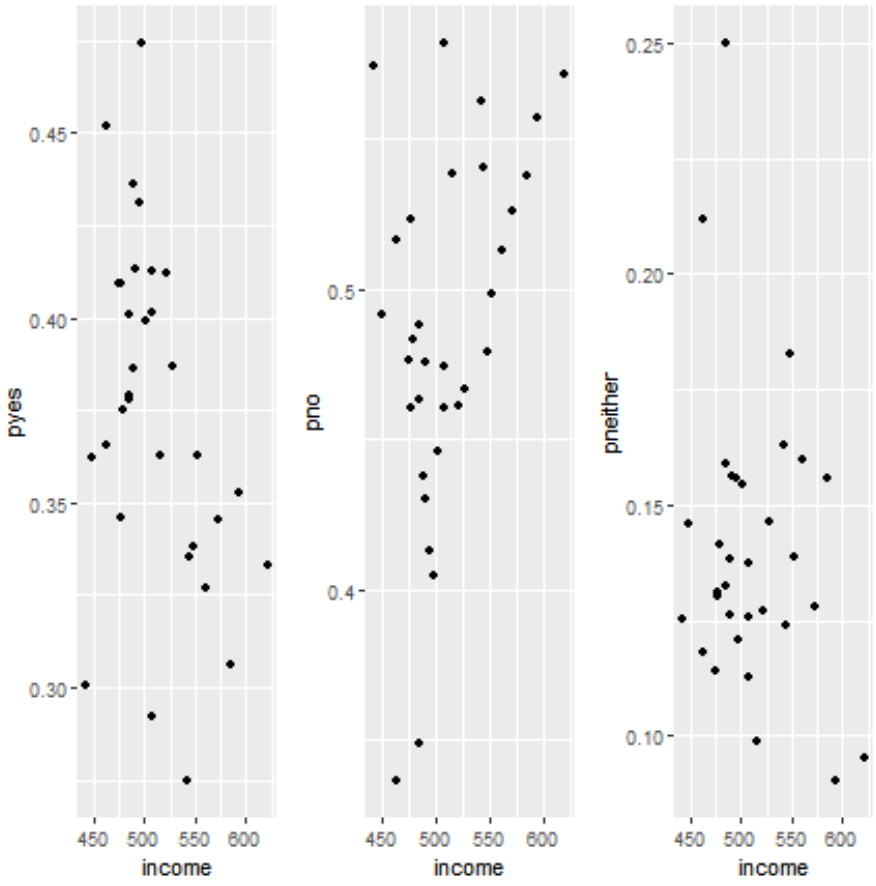


FIGURE 9.1 Proportion of votes in each category vs income

Scottish:

```
p<-list()
p[[1]]<-qplot(scottish,pyes,data=indyref)
p[[2]]<-qplot(scottish,pno, data=indyref)
p[[3]]<-qplot(scottish,pneither, data=indyref)

grid.arrange(grobs=p,nrow=1)
```

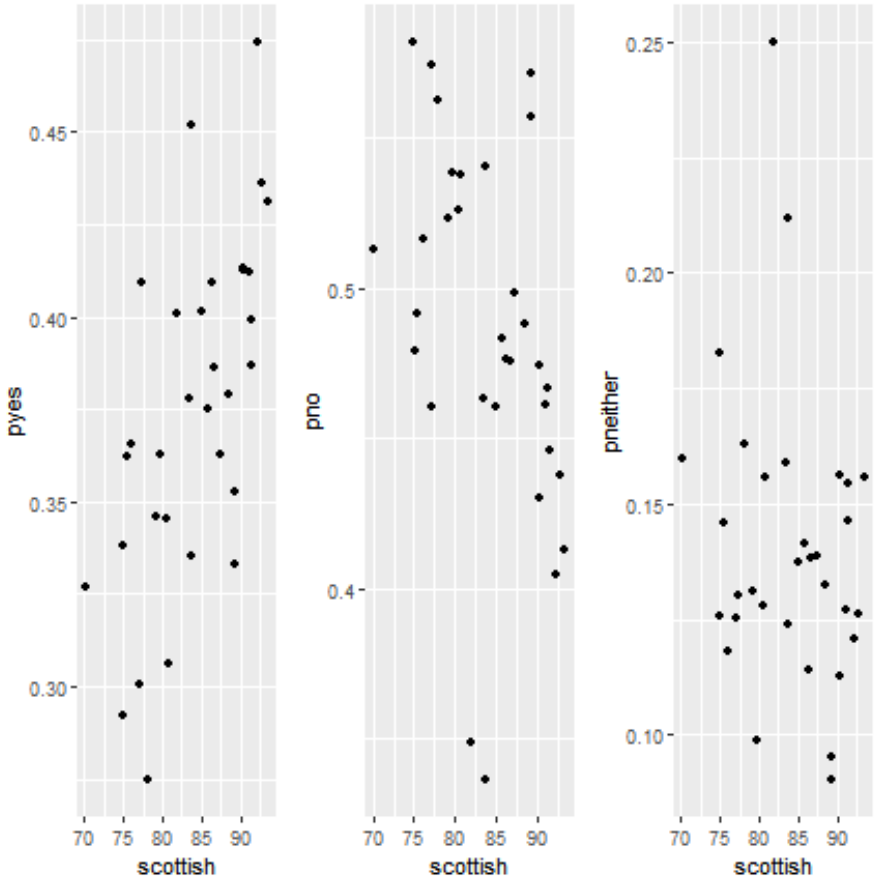


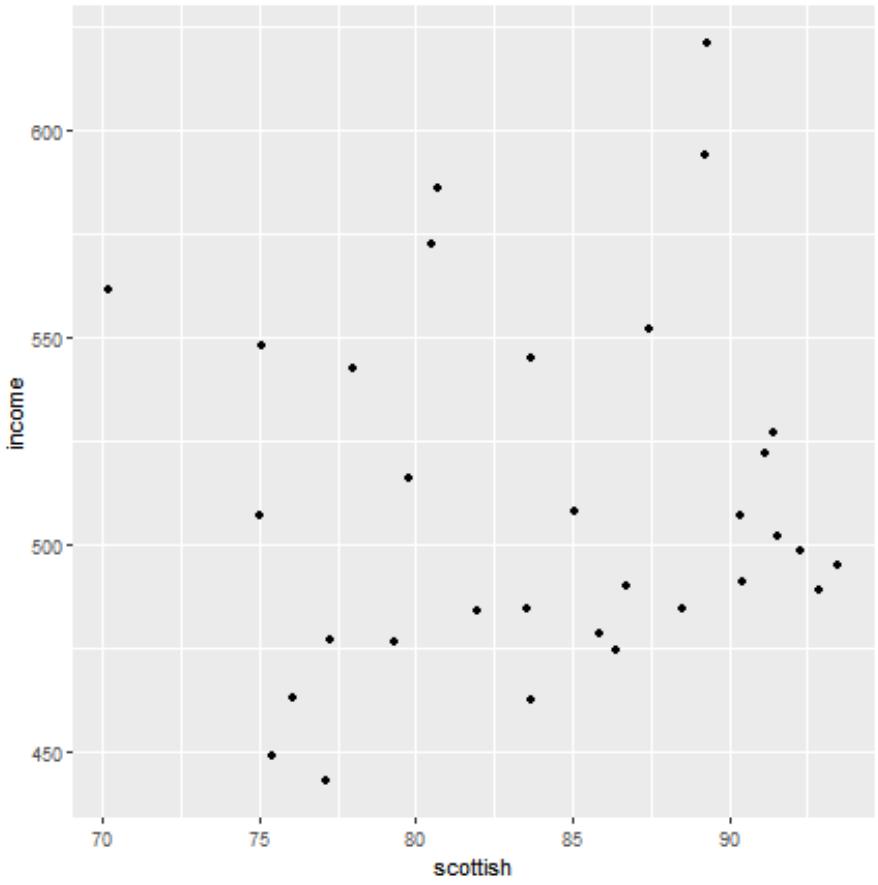
FIGURE 9.2 Proportion of votes in each category vs percentage Scottish born.

Question 4

What other plot/check should we make before starting to model with these two covariates?

Show Answer on P??

```
qplot(scottish,income,data=indyref)
```



9.3 Model specification

Here, we cover the **multinomial logit model**. This can be thought of as an extension of the logistic regression model for binomial data.

For the binomial model, we have (assuming disaggregated data, and changing the notation from the previous chapter a little):

$$
\begin{aligned}
Y_i &\overset{indep}{\sim} \text{Bin}\,(1, p_{i1}) \quad [\text{Random part}] \\
\text{logit}\,(p_{i1}) &= \eta_{i1} \quad [\text{Link function}] \\
\eta_{i1} &= \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta} \quad [\text{Structural part}]
\end{aligned}
$$

where Y_i is the response of the ith subject, $p_{i1} = \text{Pr}(Y_i = 1)$, $\mathbf{x}_i^T$ is the vector of covariate values, β_0 is an intercept parameter and β is a vector of length $p-1$ of model parameters associated with the covariates.

The logit link function for binomial models is

$$
\begin{aligned}
\text{logit}\,(p_{i1}) &= \log\left(\frac{p_{i1}}{1-p_{i1}}\right) \\
&= \log\left(\frac{p_{i1}}{p_{i2}}\right) \qquad (9.1)
\end{aligned}
$$

where $p_{i2} = 1 - p_{i1}$.

So, the logit function can be thought of as giving the *log odds* of two alternative outcomes (outcome 1 vs outcome 2). The odds (i.e., the ratio of two probabilities) are also sometimes called the *relative risk* – so another way to say this is that the logit link gives the *log relative risk* of two alternative outcomes.

The multinomial logit model generalizes this idea, modelling the log odds of each outcome relative to a *baseline* (or *reference value*). For example, if a response value of 1 is taken as the baseline, then

$$
\begin{aligned}
Y_i &\overset{indep}{\sim} \text{Mn}\,(1, p_{i1}, \ldots, p_{iJ}) && (9.2) \\
\log\left(\frac{p_{ij}}{p_{i1}}\right) &= \eta_{ij} && (9.3) \\
\eta_{ij} &= \beta_{0,j} + \mathbf{x}_i^T \boldsymbol{\beta}_j j = 2, ..., J && (9.4)
\end{aligned}
$$

or, if a response value of J is taken as the baseline, then

$$
\begin{aligned}
Y_i &\overset{indep}{\sim} \text{Mn}\,(1, p_{i1}, \ldots, p_{iJ}) && (9.5) \\
& && (9.6) \\
\log\left(\frac{p_{ij}}{p_{iJ}}\right) &= \eta_{ij} && (9.7) \\
& && (9.8) \\
\eta_{ij} &= \beta_{0,j} + \mathbf{x}_i^T \boldsymbol{\beta}_j \qquad j = 1, \ldots, (J-1) && (9.9)
\end{aligned}
$$

9.4 Model fitting

We will use maximum liklihood to fit multinomial logit models to data. Several R functions are available.

One option is the multinom function from the nnet package. This uses the first level of the categorical response variable as the baseline.

A second option is the vglm function from the (very powerful) VGAM package. In this case, it is the last level of the response variable that is used as the baselineby (default – can be changed).

```
# Fitting models
require(nnet)
mn.is<-multinom(cbind(yes,no,neither)~income+scottish,dat=indyref)
```

```
# weights:  12 (6 variable)
initial  value 4703291.041263
iter  10 value 4317775.208185
final  value 4317769.922312
converged
```

```
summary(mn.is)
```

```
Call:
multinom(formula = cbind(yes, no, neither) ~ income + scottish,
    data = indyref)

Coefficients:
        (Intercept)        income    scottish
no        0.1768074  0.0025330743 -0.01505325
neither   1.0518547 -0.0009624545 -0.01748487

Std. Errors:
         (Intercept)       income     scottish
no      5.822145e-07 1.763487e-05 0.0001076103
neither 8.039257e-07 2.468112e-05 0.0001502793

Residual Deviance: 8635540
AIC: 8635552
```

```
require(VGAM)
vglm.is<-vglm(cbind(no,neither,yes)~income+scottish,family=multinomial,dat=indyref)
summary(vglm.is)
```

```
Call:
vglm(formula = cbind(no, neither, yes) ~ income + scottish, family = multinomial,
    data = indyref)

Pearson residuals:
                         Min      1Q Median     3Q    Max
log(mu[,1]/mu[,3]) -124.39  -2.248  11.15 23.328  93.24
log(mu[,2]/mu[,3])  -52.97 -23.792 -12.13  6.477 136.37

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept):1  1.768e-01  1.999e-02   8.843   <2e-16 ***
(Intercept):2  1.052e+00  2.777e-02  37.872   <2e-16 ***
income:1       2.533e-03  2.735e-05  92.629   <2e-16 ***
income:2      -9.624e-04  3.838e-05 -25.072   <2e-16 ***
scottish:1    -1.505e-02  1.548e-04 -97.261   <2e-16 ***
scottish:2    -1.748e-02  2.149e-04 -81.366   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Residual deviance: 73969.08 on 58 degrees of freedom

Log-likelihood: -37353.06 on 58 degrees of freedom

Number of Fisher scoring iterations: 4

No Hauck-Donner effect found in any of the estimates


Reference group is level  3  of the response
```

Note that the order of the categories is different in the two calls.

Question 5

Question Which will be the reference level in each?

Show Answer on P??

The model fit

```
summary(mn.is)
```

```
Call:
multinom(formula = cbind(yes, no, neither) ~ income + scottish,
    data = indyref)

Coefficients:
        (Intercept)        income    scottish
no        0.1768074  0.0025330743 -0.01505325
neither   1.0518547 -0.0009624545 -0.01748487

Std. Errors:
         (Intercept)       income     scottish
no      5.822145e-07 1.763487e-05 0.0001076103
neither 8.039257e-07 2.468112e-05 0.0001502793

Residual Deviance: 8635540
AIC: 8635552
```

```
summary(vglm.is)
```

```
Call:
vglm(formula = cbind(no, neither, yes) ~ income + scottish, family = multinomial,
    data = indyref)

Pearson residuals:
                       Min      1Q Median     3Q    Max
log(mu[,1]/mu[,3]) -124.39  -2.248  11.15 23.328  93.24
log(mu[,2]/mu[,3])  -52.97 -23.792 -12.13  6.477 136.37

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept):1  1.768e-01  1.999e-02   8.843   <2e-16 ***
(Intercept):2  1.052e+00  2.777e-02  37.872   <2e-16 ***
income:1       2.533e-03  2.735e-05  92.629   <2e-16 ***
income:2      -9.624e-04  3.838e-05 -25.072   <2e-16 ***
scottish:1    -1.505e-02  1.548e-04 -97.261   <2e-16 ***
scottish:2    -1.748e-02  2.149e-04 -81.366   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
```

```
Residual deviance: 73969.08 on 58 degrees of freedom

Log-likelihood: -37353.06 on 58 degrees of freedom

Number of Fisher scoring iterations: 4

No Hauck-Donner effect found in any of the estimates


Reference group is level  3  of the response
```

9.5 Parameter interpretation

The parameters of a multinomial logit model can be hard to interpret!

$\beta_{0,j}$ is the log odds (or log relative risk) of outcome j relative to the baseline outcome when all covariates are zero.

$$\log\left(\frac{p_{ij}}{p_{ik}}\right) = \beta_{0,j} + x_i^T \beta_j$$

where k indicates the baseline outcome.

When $\mathbf{x}_i = 0$ then

$$\log\left(\frac{p_{ij}}{p_{ik}}\right) = \beta_{0,j}$$

```
coef(mn.is)
```

```
        (Intercept)        income    scottish
no        0.1768074  0.0025330743 -0.01505325
neither   1.0518547 -0.0009624545 -0.01748487
```

Note the intercept (i.e., β_0) for outcome no is 0.1768074.

As a check of the above interpretation, calculate the log odds of no vs yes when `income` and `scottish` are 0:

```
predict(mn.is,newdata=data.frame(scottish=c(0),income=0),type="probs")
```

```
      yes        no   neither
0.1977708 0.2360200 0.5662092
```

```
log(0.2360200/0.1977708)
```

```
[1] 0.1768078
```

If the log odds of no vs yes is 0.1768, this means the odds of no vs yes is $\exp(0.1768) = 1.193$. So, the model estimates that voters in a council area with income of 0 and 0% Scottish born residents are 1.193 times more likely to vote no rather than vote yes. Each element of β_j is the change in the log odds of outcome j relative to the baseline that is caused by a 1-unit increase in the corresponding covariate.

For simplicity of notation, we'll assume just one covariate. Then, comparing two observations, 1 and 2, with covariate values x_1 and x_2,

$$\log\left(\frac{p_{1j}}{p_{1k}}\right) - \log\left(\frac{p_{2j}}{p_{2k}}\right) = (\beta_{0,j} + x_1\beta_j) - (\beta_{0,j} + x_2\beta_j) = (x_1 - x_2)\beta_j \quad (9.10)$$

Note the slope for the `scottish` covariate (i.e., β_{scottish}) for outcome no is -0.0150533. As a check of the above interpretation, calculate how much the log odds of no vs yes changes with a 1-point increase in `scottish`

```
predict(mn.is,newdata=data.frame(scottish=c(0,1),income=0),type="probs")
log(0.2356372/0.2004448)-log(0.2360200/0.1977708)
```

Note, you'd get the same answer as above if you used

```
predict(mn.is,newdata=data.frame(scottish=c(0,1),income=0),type="probs")
```

```
        yes        no   neither
1 0.1977708 0.2360200 0.5662092
2 0.2004448 0.2356372 0.5639180
```

or

```
predict(mn.is,newdata=data.frame(scottish=c(50,51),income=0),type="probs")
```

```
        yes        no   neither
1 0.3627678 0.2039567 0.4332755
2 0.3666384 0.2030531 0.4303084
```

```
log(0.2030531/0.3666384)-log(0.2039567/0.3627678)
```

```
[1] -0.01505331
```

It is the *difference* in the covariate value that's important, not its absolute value. Also, it does not matter what values the other covariates have, so long as they don't change.

If a 1-point increase in `scottish` increases the log odds of no vs yes by -0.01505, this means the odds of no vs yes changes by a factor of $\exp(-0.01505) = 0.9851$ (i.e., decreases by 1.49%). This means the odds are multiplied by 0.9851 – in equation (9.10) we have the difference between log odds. A positive β coefficient means the odds will increase, while a negative β, like the one we have here, means that the odds will decrease. All of this is analogous to the multiple linear regression we covered early in the course – just the response and the nonlinear link are different. Nevertheless, interpretation of coefficients in these models can be hard an easier approach in many cases is to predict probabilities of each outcome at a range of covariate values ... which brings us to...

9.6 Obtaining predictions

Since (assuming $j = 1$ is the baseline)

$$\log\left(\frac{p_{ij}}{p_{i1}}\right) = \log\left(\frac{p_{ij}}{1 - \sum_{k=2}^{J} p_{ik}}\right) = \eta_{ij}$$

where $\eta_{ij} = \beta_{0,j} + \mathbf{x}_i^T \boldsymbol{\beta_j}$, then (with some algebra) we can show that

$$p_{ij} = \frac{\exp(\eta_{ij})}{1 + \sum_{k=2}^{J} \exp(\eta_{ik})}$$

Hence we can make predictions of probabilities of the outcomes, given covariate values.

For example, predict the probability of obtaining yes, no or neither at the median levels of income and scottish.

```
# Obtaining predictions
#Demonstration of prediction
x<-c(median(indyref$income),median(indyref$scottish))
x
```

```
[1] 500.150  84.395
```

```
alpha<-coef(mn.is)[,1]
alpha
```

```
       no   neither
0.1768074 1.0518547
```

```
beta<-coef(mn.is)[,-1]
beta
```

```
               income    scottish
no       0.0025330743 -0.01505325
neither -0.0009624545 -0.01748487
```

```
nu.1<-0
nu.2<-alpha[1]+x[1]*beta[1,1]+x[2]*beta[1,2]
nu.3<-alpha[2]+x[1]*beta[2,1]+x[2]*beta[2,2]
#make predictions for yes, no, neither
exp(nu.1)/(1+sum(exp(nu.2),exp(nu.3)))
```

```
[1] 0.3855482
```

```
exp(nu.2)/(1+sum(exp(nu.2),exp(nu.3)))
```

```
       no
0.4585052
```

```
exp(nu.3)/(1+sum(exp(nu.2),exp(nu.3)))
```

```
  neither
0.1559466
```

In practice, both multinom and vgam in R have predict functions.

```
predict(mn.is,newdata=data.frame(income=x[1],scottish=x[2]),type="probs")
```

```
      yes        no   neither
0.3855482 0.4585052 0.1559466
```

```
predict(vglm.is,newdata=data.frame(income=x[1],scottish=x[2]),type="response")
```

```
         no   neither       yes
1 0.4585042 0.1559466 0.3855491
```

We can use these to produce useful effects plots – for example, showing the predicted probability of each at median income, but over the range of `scottish` from 0 to 100:

```
scottish<-0:100
pred<-predict(mn.is,newdata=data.frame(income=x[1],scottish=scottish),type="probs")
plot(scottish,pred[,1],type="l",lty=1,ylim=range(pred),ylab="p")
lines(scottish,pred[,2],lty=2)
lines(scottish,pred[,3],lty=3)
```

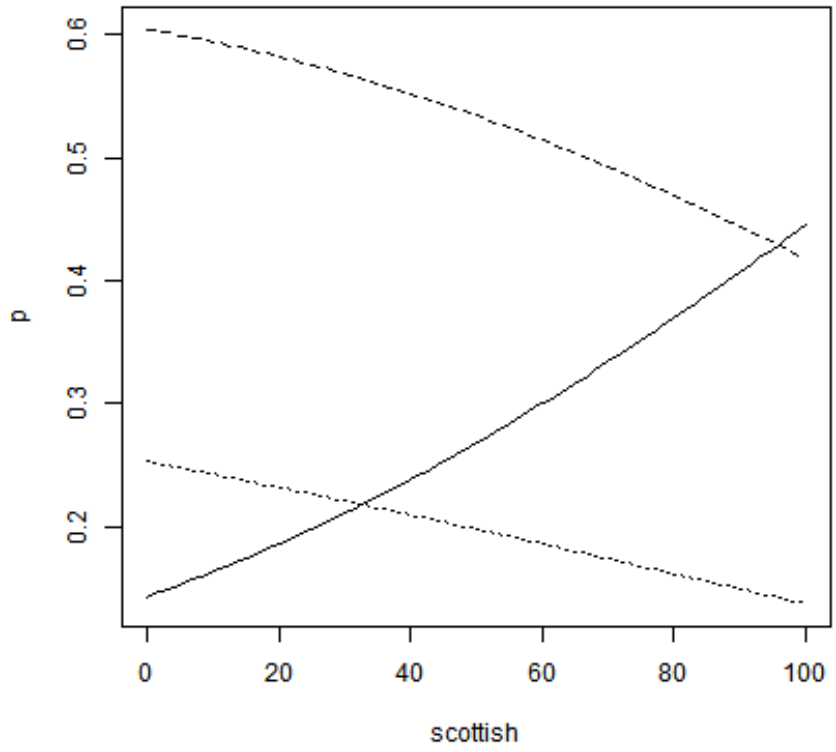


FIGURE 9.3 Predicted proportion of votes at median income. The solid line is yes, dashed is no, and dotted is neither.

The `effects` package can produce very nice effects plots, including standard errors. For more details, see

Fox, J. and J. Hong. 2009. Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package. *Journal of Statistical Software* 32(1):1-24

and the next chapter.

As well as predicting probabilities, it is straightforward to predict counts, simply by multiplying by the number of subjects.

Predict a count

```
totcount<-indyref$yes+indyref$no+indyref$neither
```

The hard way

```
indyref[1,]
```

```
   council   yes    no neither income scottish  votes      pyes       pno
1 Aberdeen 59390 84094   32032  547.8    75.04 175516 0.3383737 0.4791244
   pneither
1 0.1825019
```

```
x<-c(547.8,75.04)
nu.2<-alpha[1]+x[1]*beta[1,1]+x[2]*beta[1,2]
nu.3<-alpha[2]+x[1]*beta[2,1]+x[2]*beta[2,2]
exp(nu.1)/(1+sum(exp(nu.2),exp(nu.3)))*totcount[1]
```

```
[1] 58511.06
```

```
exp(nu.2)/(1+sum(exp(nu.2),exp(nu.3)))*totcount[1]
```

```
      no
90382.01
```

```
exp(nu.3)/(1+sum(exp(nu.2),exp(nu.3)))*totcount[1]
```

```
 neither
26622.93
```

The easy way

```
predict(mn.is,type="probs")[1,]*totcount[1]
```

```
     yes       no  neither
58511.06 90382.01 26622.93
```

9.7 Parameter inference and model selection

We can use the same methods as for Generalized Linear Models – e.g.,

- z-tests for the "significance" of parameters
- Likelihood ratio tests on parameters, and for model selection
- AIC or BIC-based model selection

```
mn.is<-multinom(cbind(yes,no,neither)~income+scottish,dat=indyref)
```

```
# weights:  12 (6 variable)
initial  value 4703291.041263
iter  10 value 4317775.208185
final  value 4317769.922312
converged
```

```
mn.s<-multinom(cbind(yes,no,neither)~scottish,dat=indyref)
```

```
# weights:  9 (4 variable)
initial  value 4703291.041263
final  value 4324465.770899
converged
```

```
mn.i<-multinom(cbind(yes,no,neither)~income,dat=indyref)
```

```
# weights:  9 (4 variable)
initial  value 4703291.041263
final  value 4323572.853127
converged
```

```
mn<-multinom(cbind(yes,no,neither)~1,dat=indyref)
```

```
# weights:  6 (2 variable)
initial  value 4703291.041263
final  value 4331095.509636
converged
```

```
aic<-AIC(mn.is,mn.s,mn.i,mn)
aic$DeltaAIC<-with(aic,AIC-min(AIC))
aic
```

```
      df     AIC DeltaAIC
mn.is  6 8635552     0.00
mn.s   4 8648940 13387.70
mn.i   4 8647154 11601.86
mn     2 8662195 26643.17
```

9.8 Model assessment

9.8.1 Model assumptions

First, we state the model assumptions.

- Observations come from a multinomial distribution
- Observations are independent, given the covariates
- Log odds of each outcome, relative to a baseline, have a linear relationship with covariates
- Log odds are not affected by other outcomes (assumption of Independence from Irrelevant Alternatives (IIA))

Explicit checking of these assumptions is not well dealt with either in standard texts or standard software. Here, we give an overview of some possible approaches.

Multinomial observations

This is hard to check. One option is to fit a series of binomial models (e.g., no vs (yes+neither), yes vs (no+neither), neither vs (no+yes)), and assess the binomial residuals.

Independent observations

When the data have a natural ordering, a plot of residuals against data order can be useful, just as with previous models. Note that the `residuals` function associated with `multinom` produces raw residuals for the probabilities of each category, while `residuals` for `vglm` produces residuals on the log odds.

Linear relationship with covariates, on log odds scale

Just as with previous models, one can plot the response against covariates and assess whether the pattern appears linear. In this case, we would calculate the log odds of no vs yes and neither vs yes, and plot these against the explanatory variables.

Independence from Irrelevant Alternatives (IIA)

We cover this "new"" assumption in more detail. In the multinomial logit model we are, modelling the (logged) odds of one outcome vs another – i.e., the preference for one outcome over another. The model assumes that the odds of choosing one outcome over another does not depend on the what alternative outcomes are available. This has been called "independence from irrelevant alternatives".

In the Scottish Independence example, we can argue that odds of voting no vs yes is not affected by whether there is a third option not to vote. However, if a fourth option had been available, "devo-max" (i.e., an option to vote for more devolved

powers instead of full independence), then it seems likely that this would affect the odds of voting no vs yes (e.g., by decreasing people's chance of supporting a straight yes, so increasing the odds of no vs yes). There are tests that test for this (basically by deleting each alternative in turn, refitting the rest and checking the coefficients don't change), but they are not very reliable, so its probably best to rely on your knowledge of the study (or that of your client).

More information on the IIA can be found here

9.8.2 Graphical assessment

There are many options here – for example a plot of observed vs predicted proportions:

```
mn.is.resid<-residuals(mn.is)
mn.is.fitted<-fitted(mn.is)
mn.is.observed<-mn.is.fitted+mn.is.resid
par(mfrow=c(1,3))
for(i in 1:3){
  plot(mn.is.fitted[,i],mn.is.observed[,i],main=colnames(mn.is.fitted)[i],xlab="fitted p",
  abline(a=0,b=1)
  lines(smooth.spline(mn.is.fitted[,i],mn.is.observed[,i],df=4),col="red",lty=2)
}
```

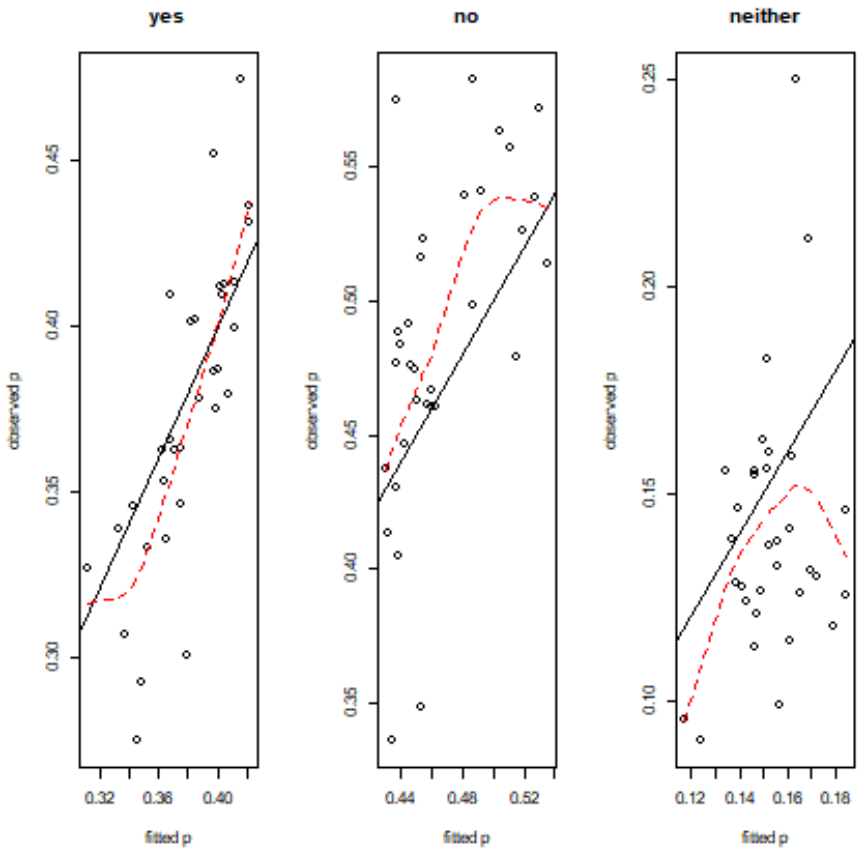


FIGURE 9.4 Observed vs fitted proportions. Black line shows 1:1; red is a smooth fit

```
par(mfrow=c(1,1))
```

9.8.3 Numerical assessment

As with the binomial GLM, a deviance-based Chi-sq statistic can be used to assess overall goodness-of-fit. First calculate the residual deviance (requires subtracting off the deviance from the saturated model when `multinom` is used – this is not necessary with `vglm`.

```
mn.sat<-multinom(cbind(yes,no,neither)~council,dat=indyref)
```

```
# weights:  99 (64 variable)
initial  value 4703291.041263
iter  10 value 4335384.840197
iter  20 value 4334454.497534
iter  30 value 4331941.740969
iter  40 value 4289864.294626
iter  50 value 4286075.557485
iter  60 value 4282712.233794
iter  70 value 4280922.665474
final  value 4280785.380183
converged
```

```
dev<-deviance(mn.is)-deviance(mn.sat)
dev
```

```
[1] 73969.08
```

```
n<-dim(indyref)[1]
n
```

```
[1] 32
```

```
p<-length(coef(mn.is))
p
```

```
[1] 6
```

```
1-pchisq(dev,(3-1)*n-p)
```

```
[1] 0
```

(Because the data are aggregated the number of observations is $(J-1)*n$).

Looks like a highly significantly bad fit! Note: $(n-p)$ is used here for degrees of freedom because the "intercept" term is included as a parameter - unlike in the GLM section, where this was not counted as a model parameter, so df was $(n-p-1)$.

Another measure of fit is a type of adjusted R^2 – McFadden's R^2, which is given by

$$R^2_{\text{McFadden}} = 1 - \frac{lnL(L_c)}{lnL(L_{null})} \tag{9.11}$$

$lnL(L_c)$ is the log-likelihood of the model under consideration, and $LnL(L_{null})$ is the log-likelihood of the null model (i.e., model with just an intercept term). Note: Since the numerator and denominator are both log-likelihoods, then the deviance can also be used – the -2 just cancels out.

Calculate the McFadden R^2 using the `vglm` function:

```
vglm.is<-vglm(cbind(no,neither,yes)~income+scottish,family=multinomial,dat=indyref)
vglm.null<-vglm(cbind(no,neither,yes)~1,family=multinomial,dat=indyref)
1-logLik(vglm.is)/logLik(vglm.null)
```

```
[1] 0.2629427
```

9.9 Other models for nominal multnomial data

We have covered one model class (multinomial logit) for nominal categorical data, and two functions in `R`. Multinomial logit analysis can be performed in `R` using other functions, for example:

- Poisson `glm` (see Faraway, J., Extending the Linear Model with R, *Chapman & Hall/CRC*, 2006)
- `mlogit`function in `mlogit` package – this also allows many other categorical data models

Other link functions are possible, such as probit (e.g., in the `mlogit` function). Nested (hierarchical) models can be specified (E.g., Scottish referrendum: Stage 1: do I vote; Stage 2: yes or no). If the covariates are all discrete-valued nominal, then contingency table models are possible. These don't distinguish which are the response variables and which the explanatory variables. The regression coefficients (β parameters) can be constrained so that all log odds share the same parameters. This is similar in spirit to the proportional odds model we cover in the next chapter.

10

Multinomial Models for ordinal data

Recall, multinomial models are extensions of binomial models and accommodate response types with three or more levels. In this section we are going to talk about **ordinal** multinomial data where the response values have a natural order (e.g. low ($y_{it} = 1$), medium ($y_{it} = 2$), high ($y_{it} = 3$)). }

10.1 Introducing the data

In this section we will focus on some schizophrenia data and examine how this illness responds to medication over time. Schizophrenia is a mental disorder which typically presents as abnormal social behaviour and failure to recognize what is real. The study was a randomised controlled trial of 437 patients randomly allocated one of four medications (one placebo, or one of three active drugs) and the subjects were followed for a number of weeks.Due to previous work (which showed that the active drugs performed similarly) the focus here is on comparing changes in illness severity with either the placebo or an active drug and so the "treatment"" covariate will have just two levels.

The response data has been classified into 4 response classes (these were originally formed from 7 classes):

1. normal or borderline mentally ill ($y_{it} = 1$)
2. mildly or moderately ill ($y_{it} = 2$)
3. markedly ill ($y_{it} = 3$)
4. severely or among the most extremely ill ($y_{it} = 4$)

where y_{it} is the response for the i-th subject ($i = 1, ..., 437$) at time t ($t = 1, ..., 6$).

The covariate data of interest is:

- ID: subject identifier
- weeks: number of weeks in the study (0-6 weeks; weeks=0 for baseline data)

- treatment: this is a dummy variable which is assigned to be a `1' if an active drug is used and a0'` if a placebo is administered. Note each subject is assigned a treatment for the duration of the study (it is an "across-patient" variable)

10.2 Exploratory Data Analysis

We can look at the overall percentage of observations in each response class (across people), by treatment group (Figure 10.1) and/or look at how the response changes within people over time.

```
require(lattice)
# Pooled data
histogram(week ~ response|as.factor(treatment))
```

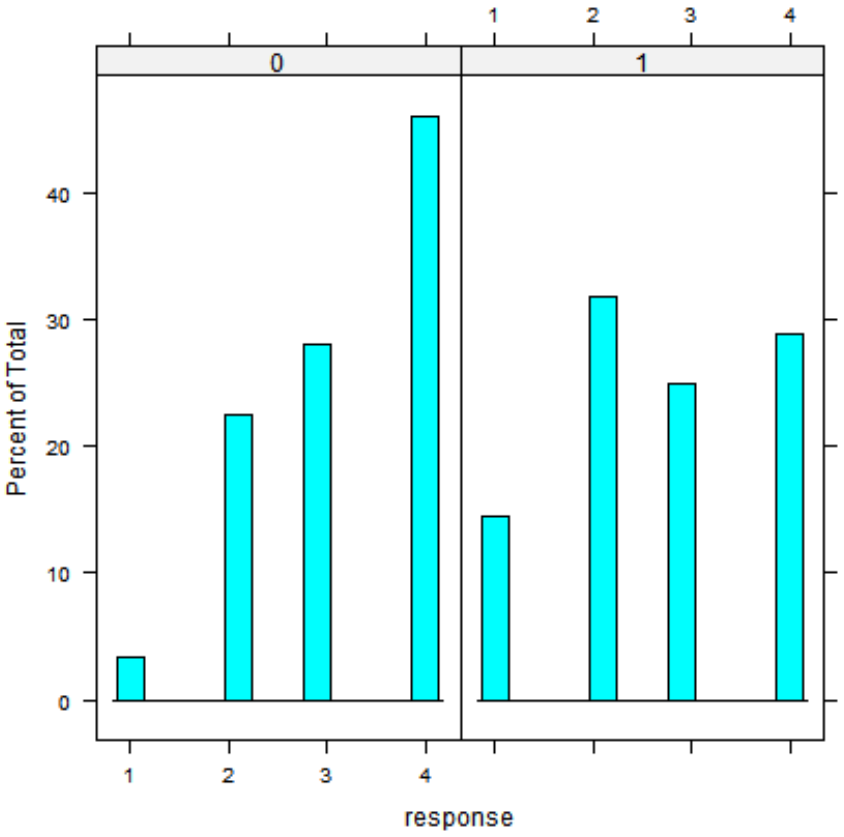


FIGURE 10.1 Barplots for each response class by treatment group for all weeks

For comparison, the two distributions at week 0

```
ind0<-which(week==0)
datab<-data[ind0,]
histogram(week~response|as.factor(treatment),data=datab)
```

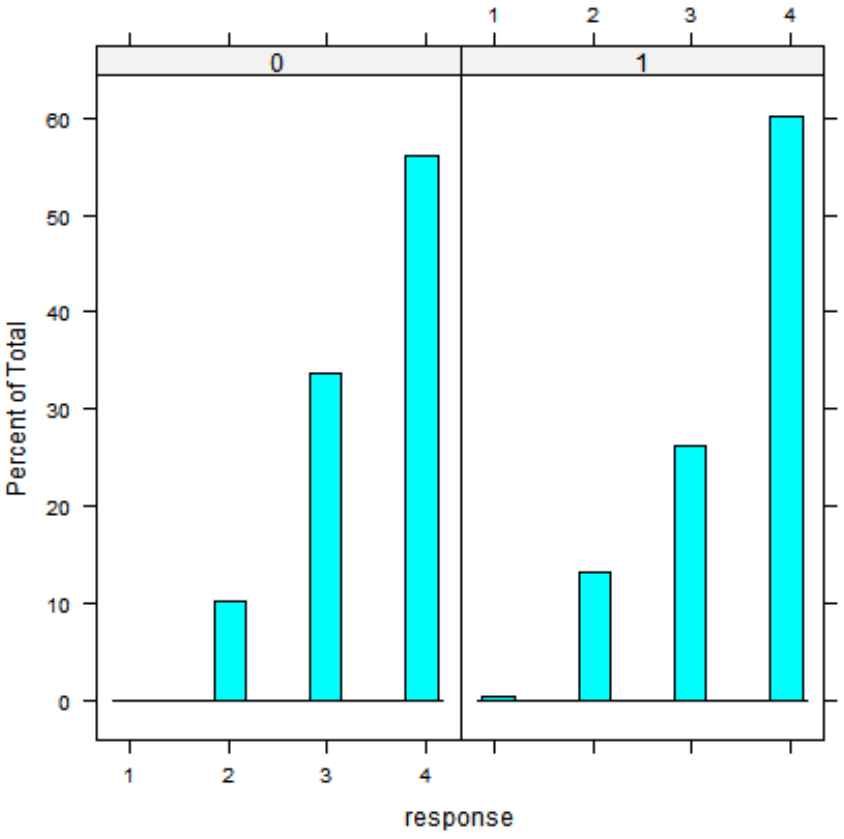


FIGURE 10.2 Barplots for each response class by treatment group for week 0

Due to the large number of subjects in this study, changes in response values over time for a random subset of people from each treatment group gives us an idea without the clutter (Figure 10.3).

```
set.seed(100)
samPlacebo<-sample(unique(data$ID[data$treatment==0]), 10)
samTrt<-sample(unique(data$ID[data$treatment==1]), 10)

p<-list()
p[[1]]<-ggplot() +
  geom_jitter(aes(x=week, y=response, colour=ID),
              data=subset(data, ID %in% samPlacebo), width = 0, height = 0.2) +
  geom_line(aes(x=week, y=response, colour=ID),
```

```
            data=subset(data, ID %in% samPlacebo))+
             xlab("Week") + ylab("Response") +  ggtitle("Placebo")

p[[2]]<-ggplot() +
  geom_jitter(aes(x=week, y=response, colour=ID),
              data=subset(data, ID %in% samTrt), width = 0, height = 0.2) +
  geom_line(aes(x=week, y=response, colour=ID),
            data=subset(data, ID %in% samTrt))+
  xlab("Week") + ylab("Response") + ggtitle("Treatment")

require(gridExtra)
grid.arrange(grobs=p, nrow=1)
```

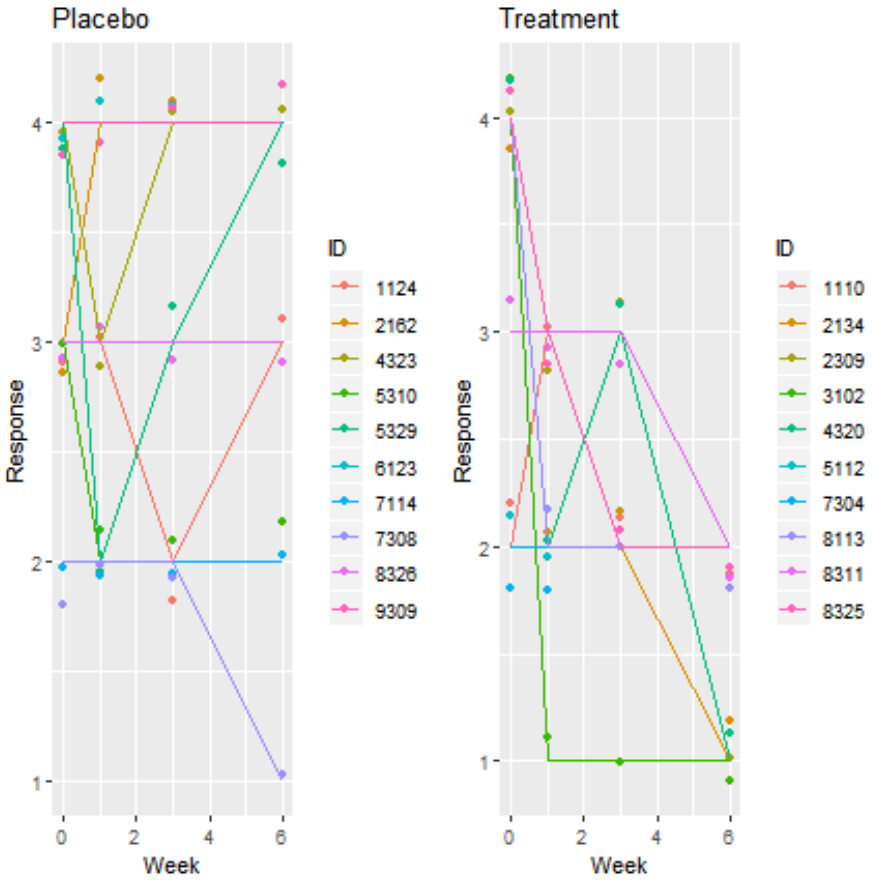


FIGURE 10.3 Exploratory data analysis for 10 randomly chosen patients (note that response values have been jittered so that multiple patients do not overlap).

10.3 Motivating the model

The idea behind an ordinal model is that there is some "latent" continuous random variable (e.g. y_{it}^*) that directly corresponds to the levels of the ordered categorical variable, but the latent variable is never actually observed. In this situation we work instead with a set of J ordered categorical/interval data which is defined using a set of 'cutpoints' $(\alpha_1, \alpha_2, ..., \alpha_{J-1})$. In this case, the response of interest, y_{it}, is equal to some interval value j when

$$\alpha_{j-1} < y_{it}^* \leq \alpha_j$$

To make this more concrete, consider an example using annual income. People tend not to report their annual income exactly (because they can't easily recall it or they are uncomfortable sharing it) and so income tends to be reported in categories, e.g:

- Less than £10,000
- at least £10,000 but less than £20,000
- at least £20,000 but less than £50,000
- £50,000 or more

In this case, we would have 4 categories with cutpoints $\alpha_1 = £10,000$, $\alpha_2 = £20,000$, $\alpha_3 = £50,000$. We also technically have the end points for this set of categories: $\alpha_0 = -\infty$ (in practice this is 0), $\alpha_4 = +\infty$ (in practice this means no upper limit). So a £15,000 annual income (for person i in year t, say) would correspond to the second response class ($y_{it} = 2$), since $£10,000 < £15,000 \leq £20,000$.

10.4 Schizophrenia data

In the schizophrenia data described in section 10.1, patients are classified into one of $J = 4$ ordered response categories. The objective of the analysis is to try to explain or predict the response class of patients given the number of weeks taking a placebo or active treatment. In particular we aim to identify any differences in illness levels with an active versus an inactive treatment.

We could start to do this by simply looking at the data (e.g. Figures 10.1 and 10.3, but this will typically be inconclusive, and further analysis is required. Specifically, if we let $y_{it} = j$ $(j = 1, ..., J)$ be the ordered categorical response variable

for subject i at time t, then we can use a model that returns a set of probabilities $(p_1, \ldots, p_J)$ for each response level given some covariates ($\mathbf{x}_{it}$: week and treatment in this case):

$$p_j(\mathbf{x}_{it}) = Pr(y_{it} = j|\mathbf{x}_{it})$$

Note, the probabilities $(p_1, \ldots, p_J)$ sum to one.

Since the response levels have a natural order, we can work with a set of J **cumulative response probabilities** (which we will call $\gamma_1, ..., \gamma_J$). The first cumulative probability contains just p_1, while the second (γ_2) contains both p_1 and p_2 and so on. So, when $J = 4$:

$$\gamma_1 = p_1, \quad \gamma_2 = p_1 + p_2, \quad \gamma_3 = p_1 + p_2 + p_3, \quad \gamma_4 = 1$$

In this model, each cumulative probability can be based on a set of covariate values:

$$\gamma_j(\mathbf{x}_{it}) = Pr(y_{it} \leq j|\mathbf{x}_{it})$$

Model specification {#modspecord}

We can model multiple ordinal response outcomes using a set of cumulative logits using the familiar link function:

$$\text{logit}[Pr(Y_{it} \leq j)] = \beta_{0j} + \beta_1 x_{1it} + ... + \beta_p x_{pit} = \beta_{0j} + \mathbf{x}_{it}^T \boldsymbol{\beta}$$

for response categories $j = 1, ..., J-1$ (since they add to one). In particular, each category attracts its own intercept parameter (β_{0j}). This model can also be written on the probability scale using the inverse link function:

$$Pr(Y \leq j) = \frac{\exp(\beta_{0j} + \mathbf{x}^T \boldsymbol{\beta})}{1 + \exp(\beta_{0j} + \mathbf{x}^T \boldsymbol{\beta})}$$

for $j = 1, ..., J-1$ (since the probabilities add to one across categories).

The covariate effects ($\boldsymbol{\beta}$) are assumed to hold for each cumulative logit ($j = 1, ..., J-1$) and do not vary across logits like the nominal multinomial models you've encountered so far. This is why this model is often called the **proportional odds** model.

An alternative is to construct the linear predictors in terms of $Pr(Y_{it} \geq j)$:

$$\text{logit}[Pr(Y_{it} \geq j)] = \beta_{0j} + \mathbf{x}_{it}^T \boldsymbol{\beta}$$

for response categories $j = 2, \ldots, J$. This can be written on the probability scale as

$$Pr(Y \geq j) = \frac{\exp(\beta_{0j} + \mathbf{x}^T\boldsymbol{\beta})}{1 + \exp(\beta_{0j} + \mathbf{x}^T\boldsymbol{\beta})}$$

Question 6

How are the β values related between the first and second specifications?

Show Answer on P??

10.5 Model fitting

We will use maximum likelihood to fit proportional odds models to data. Several R functions are available, but we will focus on the `vglm` function from the `VGAm` package. This package uses the $Pr(Y \geq j)$ specification.

An alternative is the `polr` function from the `MASS` library – we will briefly mention this later in the context of effects plotting. This package uses the $Pr(Y \leq j)$ specification.

We will begin by looking at illness levels over time (as a continuous covariate) and fit the treatment effect as a factor variable. Recall we have data across 7 weeks (including baseline information) and 4 response categories and two treatment levels.

Question 7

How many parameters will be in our model?

Show Answer on P??

Later on we will extend this model to include the interaction between time and treatment and compare this more complicated model with the relatively simplistic model using AIC scores.

We will begin to address these questions using some exploratory work.

```
require(VGAM)
fitPropOdds<- vglm(response ~ treatment+week, family=propodds, data = data)

summary(fitPropOdds)
```

```
Call:
vglm(formula = response ~ treatment + week, family = propodds,
    data = data)

Pearson residuals:
                       Min      1Q  Median     3Q    Max
logitlink(P[Y>=2]) -6.070  0.1252  0.1770 0.4632 0.8952
logitlink(P[Y>=3]) -2.905 -0.6162  0.3050 0.7618 2.4026
logitlink(P[Y>=4]) -1.804 -0.7904 -0.1906 1.0257 3.7638

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept):1  4.04498    0.15331  26.385  < 2e-16 ***
(Intercept):2  2.02544    0.12154  16.665  < 2e-16 ***
(Intercept):3  0.74112    0.11201   6.617 3.67e-11 ***
treatment     -0.89030    0.11358  -7.839 4.55e-15 ***
week          -0.42461    0.02318 -18.318  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: logitlink(P[Y>=2]), logitlink(P[Y>=3]),
logitlink(P[Y>=4])

Residual deviance: 3826.429 on 4804 degrees of freedom

Log-likelihood: -1913.214 on 4804 degrees of freedom

Number of Fisher scoring iterations: 5

No Hauck-Donner effect found in any of the estimates


Exponentiated coefficients:
treatment      week
 0.410533  0.654027
```

10.6 Inference about model parameters

Wald confidence intervals and hypothesis testing for these models proceeds as for other GLMs using the estimate (e.g. $\hat{\beta}_p$) and the standard error reported for that estimate (SE):

$$\hat{\beta}_p \pm 1.96 \times SE$$

The test statistic used is typically based on:

$$z = \frac{\hat{\beta}_p}{SE}$$

and assumed to follow a Normal distribution under the null hypothesis or the square of this test statistic is assumed to follow a Chi-squared distribution with $df = 1$. In this case we can see the large test statistics (> 7.8388) will lead us to conclude that the effects of either treatment or week are non-zero and they should be retained in the model.

In cases where a large percentage of observations fall into the highest or lowest category however, this approach to confidence interval and hypothesis testing may not yield reasonable results (since the normal assumption is no longer reliable). In these cases it is better to base hypothesis testing on likelihood ratio tests:

$$-2(L_0 - L_1)$$

where L_0 is the maximised log-likelihood function under the null hypothesis that a particular parameter is equal to zero ($\beta_p = 0$) and L_1 is the maximised log-likelihood function evaluated at $\hat{\beta}_p$.

10.7 Parameter Interpretation

Parameter interpretation for these models is not straightforward, but similar in spirit to that of the multinomial logit models of the last chapter. In the following, we will assume the $Pr(Y \geq j)$ specification used by `vglm`. β_{0j} is the cumulative log odds of outcomes $\geq j$ vs $< j$ when all covariates are zero. β_p (for covariate p) is the change in cumulative log odds of each outcome with a one unit change in the corresponding covariate.

```
beta<-coef(fitPropOdds)
beta
```

```
(Intercept):1 (Intercept):2 (Intercept):3     treatment          week
    4.0449807     2.0254376     0.7411173    -0.8902989    -0.4246066
```

Create the intercept parameters by hand:

```
pred<-predict(fitPropOdds,newdata=data.frame(treatment=0,week=0),type="response")
#beta01
log(sum(pred[2:4])/sum(pred[1]))
```

```
[1] 4.044981
```

```
#beta02
log(sum(pred[3:4])/sum(pred[1:2]))
```

```
[1] 2.025438
```

```
#beta03
log(sum(pred[4])/sum(pred[1:3]))
```

```
[1] 0.7411173
```

We can exponentiate to get back to the odds: for example, comparing the response level 3 or more vs the 2 or less, the log odds are 2.0254376, so the odds are $\exp(2.20254) = 7.579$. In other words, for patients who are in the placebo group in week 0, the model estimates they are 7.6 times more likely to report an illness severity of 3 or 4 than 1 or 2.

Confirming the interpretation for the $\boldsymbol{\beta}$ – e.g., β_1, the treatment covariate:

```
pred<-predict(fitPropOdds,newdata=data.frame(treatment=c(0,1),week=0),type="response")
pred
```

```
           1          2         3         4
1 0.01720872 0.09934919 0.2062020 0.6772401
2 0.04090720 0.20230683 0.2940124 0.4627736
```

```
#Use any arbitrary j
log(sum(pred[2,2:4])/sum(pred[2,1]))-log(sum(pred[1,2:4])/sum(pred[1,1]))
```

```
[1] -0.8902989
```

Again, we can exponentiate the difference in log odds to get back to ratio of odds: the change in log odds associated with going from placebo to active drug are 0.7411173, so the ratio of odds is $\exp(-0.8902989) = 0.410533$. In other words, the model estimates that patients given the active drug have an odds of having a particular illness severity or greater versus having less than that severity

that is 0.41 times that of patients given the placebo. We can possibly do even better in our explanation (using $1/0.41$): The model estimates that patients who received the active drug are 2.4 times *less* likely to have any given illness severity or worse than those who received the placebo. In practice, it is often much easier to interpret the model results on the probability scale using the fitted values and assisted by plots based on the fitted relationships.

10.8 Model predictions

We can obtain predictions by substituting the coefficients with their estimates based on the model. E.g. for observation 1 and the model fitted thus far:

```
head(data)
```

```
    ID response treatment week  sweek
1 1103        4         1    0 0.0000
2 1103        2         1    1 1.0000
3 1103        2         1    3 1.7321
4 1103        2         1    6 2.4495
5 1104        4         1    0 0.0000
6 1104        2         1    1 1.0000
```

Taking ID 1103 as an example, for treatment 1 and week 0:

$$\begin{aligned}
\hat{Pr}(Y_{10} \geq 1) =& 1 && (10.1)\\
\text{logit}[Pr(Y_{10} \geq 2)] =& \beta_{01} + \beta_1 x_{1it} + \beta_2 x_{2it} && (10.2)\\
=& 4.04498 - 0.89030 \times 1 - 0.42461 \times 0 && (10.3)\\
=& 3.15468 && (10.4)\\
\hat{Pr}(Y_{10} \geq 2) =& \exp(3.15468)/(1 + \exp(3.15468)) && (10.5)\\
=& 0.9590927 && (10.6)\\
\text{logit}[Pr(Y_{10} \geq 3)] =& \beta_{02} + \beta_1 x_{1it} + \beta_2 x_{2it} && (10.7)\\
=& 2.02544 - 0.89030 \times 1 - 0.42461 \times 0 && (10.8)\\
=& 1.13514 && (10.9)\\
\hat{Pr}(Y_{10} \geq 3) =& \exp(1.13514)/(1 + \exp(1.13514)) && (10.10)\\
=& 0.7567862 && (10.11)\\
\text{logit}[Pr(Y_{10} \geq 4)] =& \beta_{03} + \beta_1 x_{1it} + \beta_2 x_{2it} && (10.12)\\
=& 0.74112 - 0.89030 \times 1 - 0.42461 \times 0 && (10.13)\\
=& -0.14918 && (10.14)\\
\hat{Pr}(Y_{10} \geq 4) =& \exp(-0.14918)/(1 + \exp(-0.14918)) && (10.15)\\
=& 0.462774 && (10.16)
\end{aligned}$$

and

- $Pr(Y_{10} = 1) = 1 - 0.9590927 = 0.04090727$
- $Pr(Y_{10} = 2) = 0.9590927 - 0.7567862 = 0.2023065$
- $Pr(Y_{10} = 3) = 0.7567862 - 0.462774 = 0.2940122$
- $Pr(Y_{10} = 4) = 0.462774$

which is what we see if use the {fitted} command to get the predictions from the model:

```
fitted(fitPropOdds)[1,]
```

```
        1         2         3         4
0.0409072 0.2023068 0.2940124 0.4627736
```

10.9 Visual Interpretation

To obtain plots for model relationships (``effects plots'') we will refit the model using the polr function (instead of the

vglm function) to exploit the plotting functions in the effects package. Note that the polr function is called directly from the MASS package in the code below. This is because the MASS package contains other functions that will overwrite those in the dplyr package.

```
#require(MASS)
require(effects)

data$response2<- as.factor(data$response)
data$trt<- as.factor(treatment)

polrSimple<-MASS::polr(response2 ~ trt+week, data=data)
```

```
plot(effect("week", polrSimple))
```

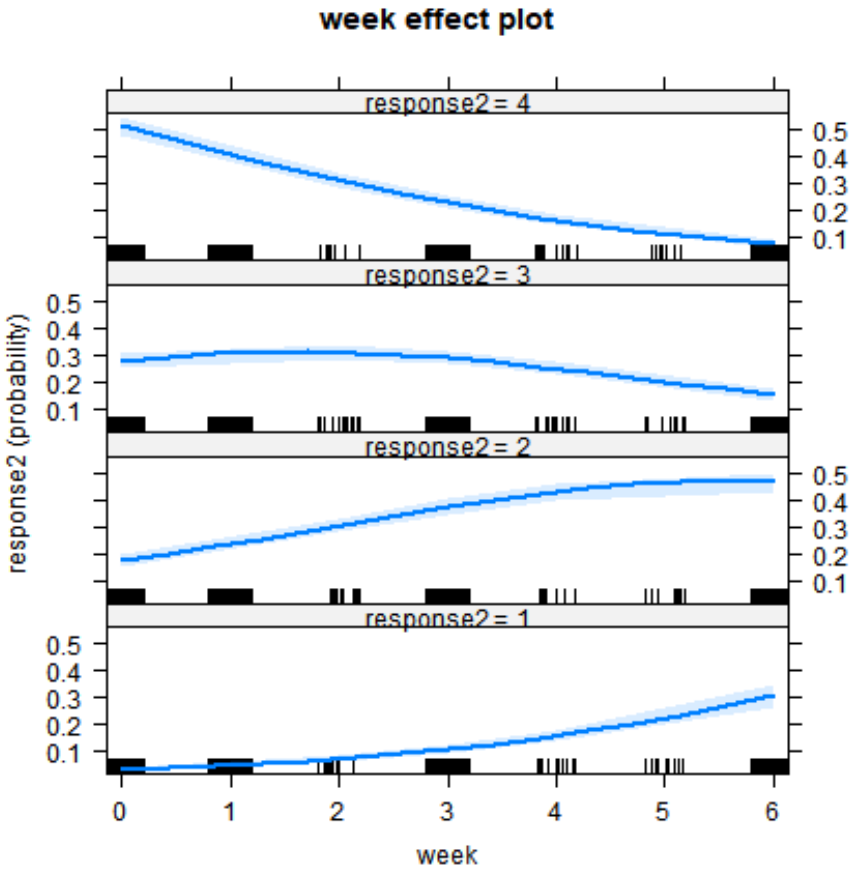


FIGURE 10.4 Fitted week relationship under the polrSimple model.

```
plot(effect("trt", polrSimple))
```

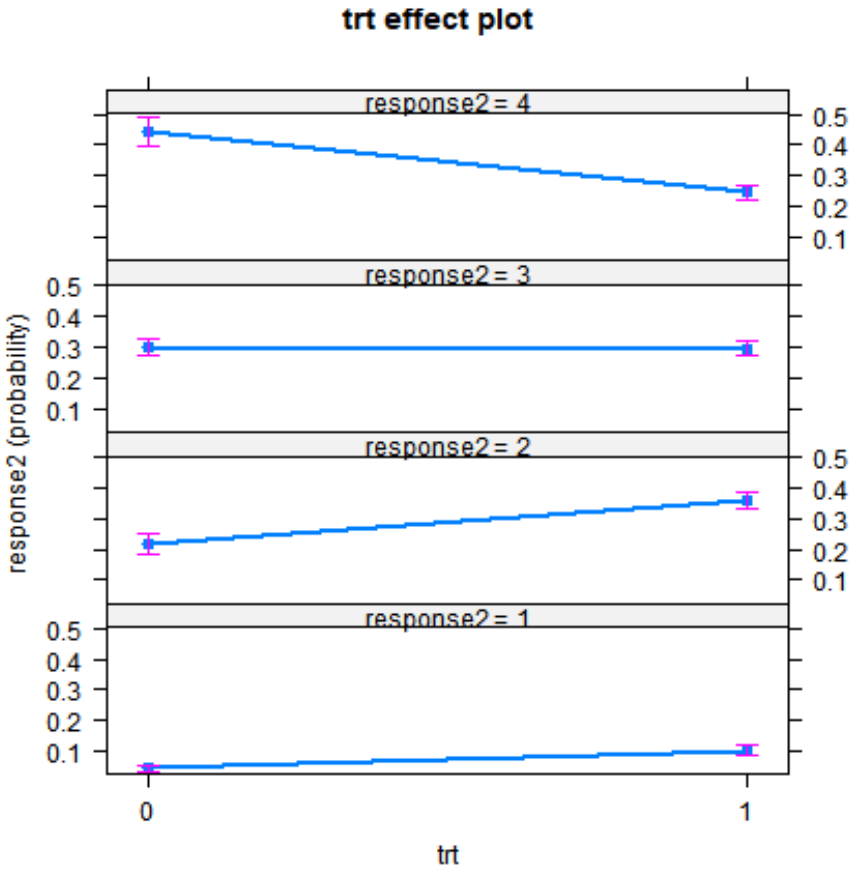


FIGURE 10.5 Fitted treatment relationship under the polrSimple model.

For example based on the relatively simple model (Figures 10.4 and 10.5) we can see,

- the probability of falling into either the normal or mild group increases over time (though in the latter case this probability plateaus after about week 5).
- the probability of being in the markedly ill or severely ill groups decreases over time (though in the latter case this probability falls much quicker).
- the probability of being in the normal or markedly ill groups are similar across treatments
- the probability of being in the severely ill group is lower for the treatment group (compared to the control) though the opposite is true for the mildly ill group.

Warning: the patterns we observe are dependent on the model assumptions (e.g., covariate effect is linear on the cumulative log odds scale; proportional odds; etc.), so this will need to be checked before we are confident in our interpretation. We could also examine the model output using the fitted logits, although they are harder to interpret. We can see if these relationships are statistically significant in the model based on z-tests or likelihood ratio χ^2 tests:

```
require(car)
Anova(polrSimple)
```

```
Analysis of Deviance Table (Type II tests)

Response: response2
     LR Chisq Df Pr(>Chisq)
trt     63.02  1  2.047e-15 ***
week   368.24  1  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.10 Model selection and inference

One of the main objectives of this study was to identify if the active treatment affects the illness profile for each subject across time, compared with the placebo group. In this case, this amounts to asking if the `week' relationship changes with treatment. We can assess this by including an interaction effect.

```
t.plus.w<- vglm(response ~ treatment+week, family=propodds, data = data)
BIC(t.plus.w)
```

```
[1] 3863.327
```

```
t.times.w<- vglm(response ~ treatment*week, family=propodds, data = data)
BIC(t.times.w)
```

```
[1] 3843.764
```

The interaction model is strongly selected by BIC.

What inference can we make about the interaction?

```
summary(t.times.w)
```

```
Call:
vglm(formula = response ~ treatment * week, family = propodds,
    data = data)

Pearson residuals:
                       Min      1Q  Median     3Q    Max
logitlink(P[Y>=2]) -5.535  0.1167  0.1603 0.4235 0.9576
logitlink(P[Y>=3]) -2.315 -0.5562  0.2900 0.8046 2.7010
logitlink(P[Y>=4]) -1.485 -0.8047 -0.1979 0.9603 4.2867

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1    3.6553     0.1704  21.453  < 2e-16 ***
(Intercept):2    1.5743     0.1462  10.771  < 2e-16 ***
(Intercept):3    0.2819     0.1397   2.017   0.0436 *
treatment       -0.3015     0.1592  -1.894   0.0582 .
week            -0.2200     0.0445  -4.944 7.64e-07 ***
treatment:week  -0.2691     0.0512  -5.255 1.48e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: logitlink(P[Y>=2]), logitlink(P[Y>=3]),
logitlink(P[Y>=4])

Residual deviance: 3799.486 on 4803 degrees of freedom

Log-likelihood: -1899.743 on 4803 degrees of freedom

Number of Fisher scoring iterations: 5

No Hauck-Donner effect found in any of the estimates


Exponentiated coefficients:
     treatment           week treatment:week
     0.7397299      0.8024826      0.7640955
```

The interaction term is negative, and the z-value is large --
what does this mean in terms of its effect on the response?

Roughly: This tells us that the active treatment tends to result in reduced illness levels over time (compared with the placebo group) - a desirable outcome.

Another way to interpret the results is through an effect plot, again using polr:

```
polrInteraction<-MASS::polr(response2 ~ trt*week, data=data)
plot(effect("trt:week", polrInteraction, style="stacked"))
```

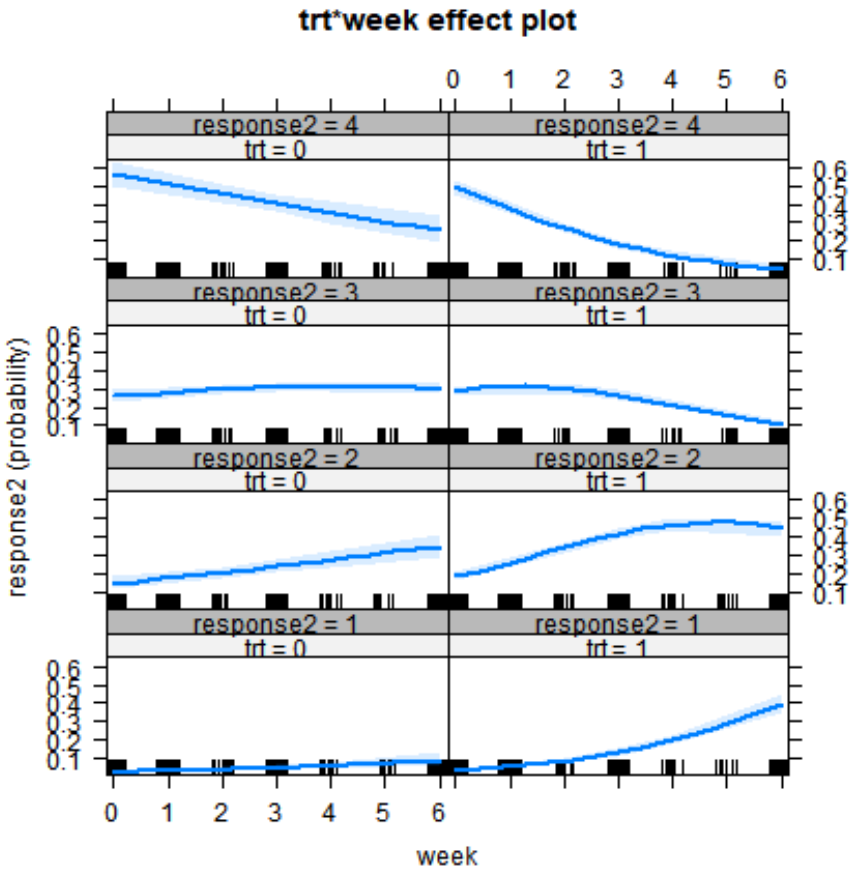


FIGURE 10.6 Fitted treatment:week relationship under the polrInteraction model.

Interpreting the plot:

- the probability of falling into either the normal or mild

group increases over time, and these increases are more rapid for those receiving active treatment.
- the probability of being in the markedly ill group looks relatively unchanged over time for the placebo group but decreases over time for those receiving active treatment.
- the probability of being in the severely ill group decreases over time regardless of treatment, however this decreases more quickly for those receiving active treatment.

10.11 Model assessment

Model assumptions

- Observations come from a multinomial distribution
- Observations are independent, given the covariates
- Cumulative log odds of each outcome have a linear relationship with covariates
- The slope of this relationship is the same for each outcome level (the ``proportional odds''" assumption)

Multinomial observations

As for the multinomial logit models, one option is to fit a series of binomial models, and assess the binomial residuals.

Independent observations

Again, as for the multinomial logit models, plots of residuals against data order can be useful where the data have a natural order. In the Schizophrenia example, the data come from a time series, so it might be possible to plot residuals against time. However, there are few time points, and results likely vary strongly by individual, masking any possible pattern. Nevertheless, because the data are multiple replicated time series, the independence assumption is unlikely to be met. In this case, Generalized Estimating Equations GEEs; covered in MT5764) would be a good alternative for comparison with model results.

Linear relationship with covariates, on cumulative log odds scale

If the data are aggregated (i.e., grouped), then one could

calculate empirical cumulative log odds for each group, and plot these empirical log odds against covariate values to assess linearity. The schizophrenia data are disaggregated (i.e., ungrouped) but could be grouped by individuals. For other datasets with continuous covariates, the covariates could be binned for the purpose of addressing this assumption.

Proportional odds

Possibly the easiest way to test this assumption is by fitting a model that does not assume it, and using a model selection statistic to determine if it is better supported by the data.

```
prop.odds<- vglm(response ~ treatment*week, family=cumulative(parallel=TRUE), data = data)
BIC(prop.odds)
```

```
[1] 3843.764
```

```
cum.odds<- vglm(response ~ treatment*week, family=cumulative, data = data)
BIC(cum.odds)
```

```
[1] 3878.096
```

10.12 Other models for ordinal categorical data

Other functions are available to analyze proportional odds models. For example lrm (library Desing), lcr (library ordinal) and nordr (library gnlm). We have no first-hand experience of any of these, but the ordinal package in general seems to offer plenty of flexibility for fitting proportional odds and related models.

Proportional odds models are a special case of cumulative link models -- see, for example, Agresti (2010) Analysis of Ordinal Categorical Data. As well as models that don't assume proportional odds, there are half-way-house versions where some of the β parameters are shared among cumulative response levels and some are not. These models are possible to fit using vglm (and likely other software).

Other link functions may be used. Two common options are probit and complementary log-log ; there are many others.

- The probit link has the attractive interpretation that the underlying latent variable is normally distributed; in practice results are often very similar to the logit link.
- Complementary log-log is skewed, in that it approaches 1.0 faster than 0.0, so is better for applications where higher categories are more probable, such as human life-table analysis.