Chapter 15
Non-independent Data

15.1 Introduction

Many infectious disease experiments result in non-independent data because of spatial autocorrelation across fields (such as discussed in Chap. 13), repeated measures on experimental animals (such as the in-host *Plasmodium* data discussed in Sect. 7.7), or other sources of correlated experimental responses among experimental units (such as the possibility of correlated infection fates among the rabbit littermates discussed in Sect. 4.3). Statistical methods that assume independence of observations are not strictly valid and/or fully effective on such data (e.g., Legendre 1993; Keitt et al. 2002). "Mixed-effects models" and "Generalized linear mix-effects models" (GLMMs) have been/are being developed to optimize the analysis of such data (Pinheiro and Bates 2006).

While this full topic is outside the main scope of this text, it is very pertinent to analyses of disease data, so we will consider the three case studies.

```
require(nlme)
require(ncf)
require(lme4)
require(splines)
```

15.2 Spatial Dependence

We use the rust example introduced in Sect. 13.2 (Fig. 13.1) to illustrate two approaches to accounting for spatial dependence in disease data: (1) random blocks vs (2) spatial regression. This experiment looked at severity of a foliar rust infection

This chapter uses the following R-packages: `nlme`, `ncf`, `lme4`, and `splines`.

O. N. Bjørnstad, *Epidemics*, Use R!, https://doi.org/10.1007/978-3-319-97487-3_15

on three focal individuals of flat-top goldenrods in each of 120 plots across a field divided into four blocks. The experimental treatments were (1) watering or not and (2) whether surrounding non-focal host plants were conspecifics only, a mixture of conspecifics and an alternative host (the Canadian goldenrod) or the alternative host only.

15.2.1 Random Blocks

As in our spatial pattern analysis, we `jitter` the coordinates because some methods require unique coordinates for each data point.

```
data(gra)
gra$jx = jitter(gra$xloc)
gra$jy = jitter(gra$yloc)
```

We first use `lme` to fit two random effect models. The first considers individuals in blocks. The second considers plots nested in blocks.

```
fit=lme(score~comp+water, random = ~1 | block,
     data= gra, na.action=na.omit)
fit2=lme(score~comp+water, random = ~1 | block / plot,
     data= gra, na.action=na.omit)
```

We next do a likelihood ratio-test to check for the better fit. The likelihood ratio test (provided by `anova`) shows that the nested model provides the best fit.

```
anova(fit, fit2)

  ##      Model df      AIC      BIC    logLik   Test
  ## fit      1  6 1186.175 1209.424 -587.0874
  ## fit2     2  7 1077.579 1104.704 -531.7895 1 vs 2
  ##        L.Ratio p-value
  ## fit
  ## fit2 110.5959  <.0001
```

The `intervals`-call shows that the between-plot variance is about twice as large as the between-block variance, and watered plots have a significantly higher rust burden.

```
intervals(fit2)

  ## Approximate 95% confidence intervals
  ##  Fixed effects:
  ##                   lower      est.     upper
  ## (Intercept)  0.8678624 1.4180556 1.9682487
  ## compSOL     -0.2517755 0.2083333 0.6684422
  ## compSYM     -0.1726089 0.2875000 0.7476089
  ## watermesic   0.2548782 0.6305556 1.0062329
  ## attr(,"label")
  ## [1] "Fixed effects:"
  ##
  ##  Random Effects:
  ##   Level: block
  ##                     lower      est.   upper
  ## sd((Intercept)) 0.154977 0.4101308 1.08537
  ##   Level: plot
  ##                      lower      est.    upper
  ## sd((Intercept)) 0.7901556 0.9302044 1.095076
  ##
  ##  Within-group standard error:
  ##     lower      est.     upper
  ## 0.7317349 0.8001735 0.8750132
```

15.2.2 Spatial Regression

The above randomized block mixed-effects models are the classic solution to analyzing experiments with spatial structure. An alternative is to formulate a regression model that considers the spatial dependence among observations as a function of separating distance. To investigate how proximate observations on different experimental treatments may be spatially autocorrelated, we can explore the spatial dependence among the *residuals* from a simple linear analysis of the data. We use the nonparametric spatial covariance function (as implemented in the `spline.correlogram()`-function in the `ncf`-package) discussed in Chap. 13. We first fit the simple regression model that ignores space altogether.

```
fitlm = lm(score ~ comp + water, data = gra)
```

Next we calculate the spatial correlation function among the residuals of the fit (Fig. 15.1).

```
fitc = spline.correlog(gra$x, gra$y, resid(fitlm))
```

The nonparametric spatial correlation function reveals strong spatial autocorrelation that decays to zero around 38 m (with a CI of 31–43 m).

```
plot(fitc, ylim = c(-0.5, 1))
```

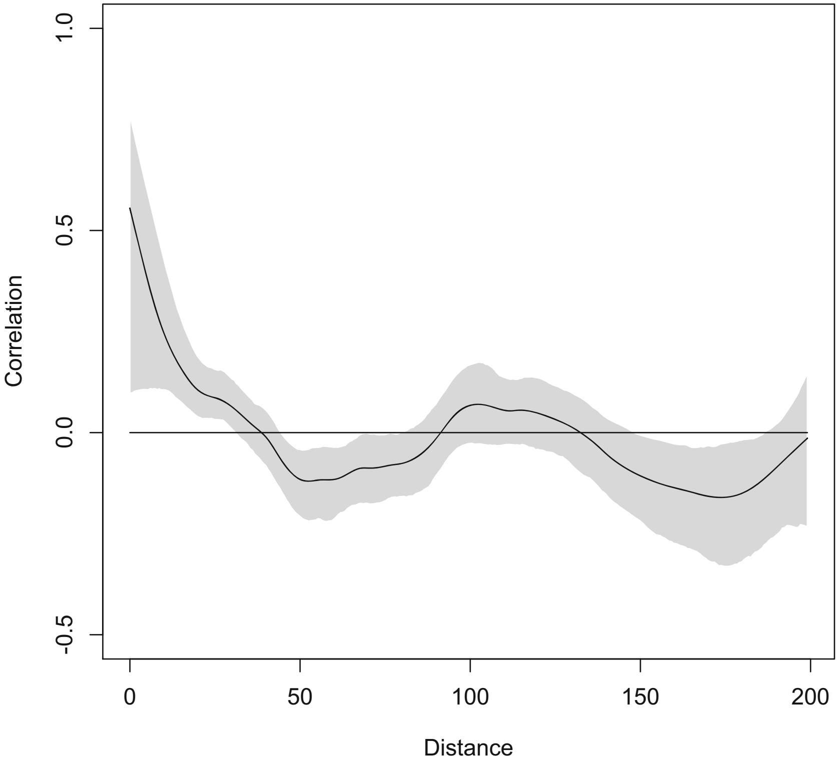

Fig. 15.1 The spline correlogram of the residuals from the regression model of Keslow's rust data

To fit the spatial regression model we use the gls-function from the nlme-package (Pinheiro and Bates 2006). This function fits mixed models from data that have a single dependence group, i.e., one spatial map, one time series, etc.; With multiple groups we use the lme-function discussed (see Sect. 15.3). There are many possible models for spatial dependence. We compare the exponential model (which assumes the correlation to decay with distance according to $exp(-d/a)$ where d is distance and a is the scale) and the Gaussian model ($exp(-(d/a)^2)$. [The nugget-flag means that the function is not anchored at one at distance zero]. We compare these to the nonspatial model (fitn) and the best random block model (fit2) using AIC.

```
fite=gls(score~comp+water, corr = corSpatial(form =
      ~jx + jy, type="exponential", nugget=TRUE),
     data=gra, na.action=na.omit)
fitg=gls(score~comp+water, corr = corSpatial(form =
     ~jx + jy, type="gaussian", nugget=TRUE), data=gra,
     na.action=na.omit)
fitn=gls(score~comp+water,  data=gra, na.action=na.omit)
AIC(fite, fitg, fitn, fit2)
```

```
## df AIC
## fite  7 1061.725
## fitg  7 1064.522
## fitn  5 1209.500
## fit2  7 1077.579
```

The AICs show that the exponential model provides the best fit. Moreover, the spatial regression model provides a better fit than the nested random effect model. This is presumably because of the gradual decay in correlation with distance (Fig. 15.1).

```
summary(fite, corr = FALSE)
```

```
## Generalized least squares fit by REML
##   Model: score ~ comp + water
##   Data: gra
##          AIC      BIC    logLik
##   1061.725 1088.849 -523.8623
##
## Correlation Structure: Exponential spatial
##  correlation
##  Formula: ~jx + jy
##  Parameter estimate(s):
##     range    nugget
## 9.9222621 0.3210873
##
## Coefficients:
##                 Value Std.Error  t-value p-value
## (Intercept) 1.4914991 0.2595408 5.746685  0.0000
## compSOL     0.1776521 0.2030045 0.875114  0.3821
## compSYM     0.2068005 0.2015687 1.025955  0.3056
## watermesic  0.4998769 0.1589941 3.143996  0.0018
##
##  Correlation:
##             (Intr) cmpSOL cmpSYM
## compSOL     -0.393
## compSYM    -0.397  0.547
## watermesic -0.291  0.041  0.022
##
## Standardized residuals:
##        Min         Q1        Med         Q3
## -1.3253792 -0.7737412 -0.1546712  0.6258009
##        Max
##  3.9090911
##
## Residual standard error: 1.281276
## Degrees of freedom: 360 total; 356 residual
```

The parametrically estimated range of 9.8 m is a bit longer (but within the confidence interval) of the e-folding scale (5.5 m) estimated by the spline correlogram; 1-nugget = 0.64 is comparable (but a little greater) than the 0.55 y-intercept. We can use the `Variogram`-function from the `nlme`-package to see if the spatial model adequately captures reflects the spatial dependence (Fig. 15.2). It looks like a plausible fit.

```
plot(Variogram(fite))
```

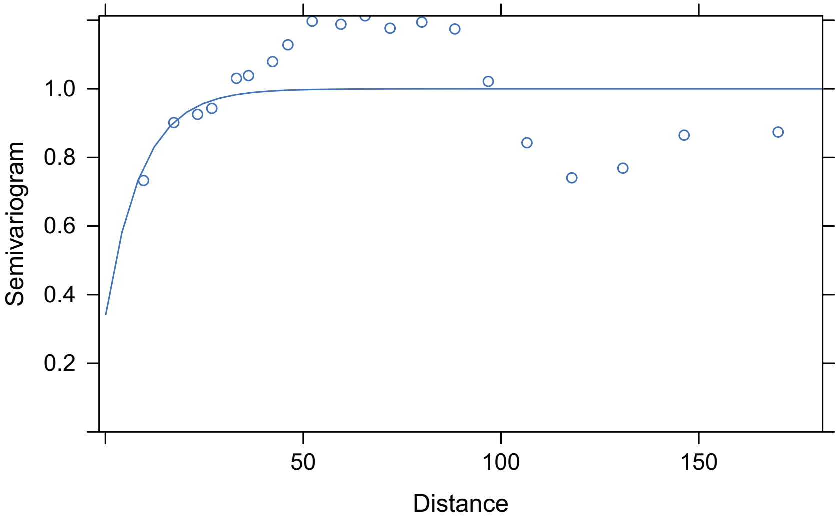

Fig. 15.2 A variogram plot of the fitted and observed spatial dependence for the spatial regression model

15.3 Repeated Measures of In-Host Mouse Malaria

Repeated measurements usually result in non-independent data because of the inherent serial dependence. Consider Huijben's data on anemia of mice infected by five different strains of *Plasmodium chaubodii* introduced in Sect. 7.7 with lots of measurements taken on days 3 through 21, 24, 26, 28, 31, 33, and 35. We will study the red blood cell counts (RBCs) of mice infected by one of five different clones as well as the control group. The sample sizes per treatment were 10 for AQ, BC, CB, and ER, 7 for AT and 5 for control. Eleven of the animals died. `SH9` has the data (in long format).[1] For the analysis we strip some unnecessary columns 1, 3, 4, 7, 8, and 11 that are extraneous to focus on the RBC count:

```
data(SH9)
SH9RBC = SH9[, -c(1, 3, 4, 7, 8, 10, 11)]
```

For the repeated measures analyses we create a `groupedData`-object from the data frame using the `nmle`-package. The below call declares how the RBC counts represent time series for each mouse. Note that mice that died are scored by zero RBC count in the data set and that these zeros end up dominating patterns, we therefore rescore these data as missing (`NA`), and plot the grouped data object to visualize the anemia by treatment (Fig. 15.3).

```
RBC = groupedData(RBC ~ Day | Ind2, data = SH9RBC)
RBC$RBC[RBC$RBC == 0] = NA
plot(RBC, outer = ~Treatment, key = FALSE)
```

The main difference is between control and treatments, but the maximum anemia varies somewhat among strains. To test for significant differences we use `lme` to build a repeated measures model. In the simplest case we follow standard convention and model the time series using day as an ordered factor and assume the treatment effect to be additive. The `random=` $\sim$ `1|Ind2`-call in the formula indicates that we assume there to be individual variation in the intercept (but not the slopes) among individuals. We then use the `ACF` function to look for evidence of serial dependence in the residuals from the fit. As is apparent from the ACF plot there is temporal autocorrelation in the residuals out to at least 4 days (Fig. 15.4).

```
mle.rbc=lme(RBC~Treatment+ordered(Day), random =
   ~1|Ind2, data=RBC, na.action=na.omit, method="ML")
plot(ACF(mle.rbc))
```

There are many models for serial dependence. We use a first order autoregressive process (AR1). This is specified by the `correlation=corAR1(form=` $\sim$

[1] With repeated measures data we often use both `long`-format with one line for each observation and `wide`-format with one line for each experimental unit.

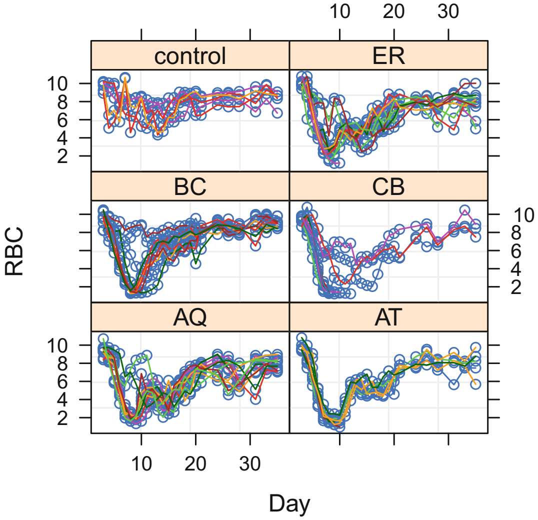

Fig. 15.3 RBC counts of control and *P. chaubodii*-infected mice. Each panel represents a different parasite strain

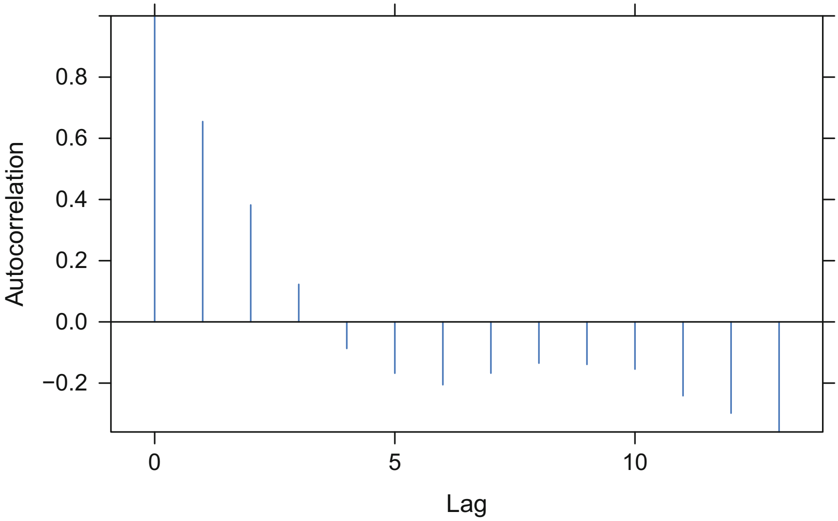

Fig. 15.4 Serial dependence as quantified using the `ACF`-function on the repeated measures mixed-effects model of the SH9RBC data

`Day | Ind2)` function call. Note that this is one of a variety of time-series models available in the `nlme`-package, the most general of which is the ARMA(p, q) model discussed in Sect. 6.2.1.

```
mle.rbc2=lme(RBC~Treatment+ordered(Day), random=
     ~1|Ind2, data=RBC, correlation=corAR1(form=~
     Day|Ind2), na.action=na.omit, method="ML")
mle.rbc2
```

```
## Linear mixed-effects model
##   Data: RBC
##   Log-likelihood: -1568.255
##   Fixed: RBC ~ Treatment + ordered(Day)
##       (Intercept)      TreatmentAT      TreatmentBC
##       5.860494309      0.024586193      0.947853117
##       TreatmentCB Treatmentcontrol      TreatmentER
##      -0.022048465      1.560872851      0.325308683
##    ordered(Day).L   ordered(Day).Q   ordered(Day).C
##       3.339300000      6.015597509     -5.057192257
##    ordered(Day)^4   ordered(Day)^5   ordered(Day)^6
##       1.498354649      0.067695099     -0.600409959
##    ordered(Day)^7   ordered(Day)^8   ordered(Day)^9
##       1.352000127     -1.122142721     -0.394162545
##   ordered(Day)^10  ordered(Day)^11  ordered(Day)^12
##       0.312998475     -0.673514349     -0.122937927
##   ordered(Day)^13  ordered(Day)^14  ordered(Day)^15
##       0.219014886      0.378460147      0.191963472
##   ordered(Day)^16  ordered(Day)^17  ordered(Day)^18
##       0.180627944     -0.024392052      0.032617128
##   ordered(Day)^19  ordered(Day)^20  ordered(Day)^21
##      -0.142080994     -0.046539002     -0.054854991
##   ordered(Day)^22  ordered(Day)^23  ordered(Day)^24
##      -0.039333282     -0.210031799      0.006591632
##
## Random effects:
##  Formula: ~1 | Ind2
##           (Intercept) Residual
## StdDev: 0.0002332905 1.327223
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~Day | Ind2
##  Parameter estimate(s):
##      Phi1
## 0.7088701
## Number of Observations: 1104
## Number of Groups: 52
```

The Phi1 parameter of 0.7088 represents the estimated day to day correlation, which is substantial. We can plot the predicted and observed correlation. The AR1-model seems to be a nice fit (Fig. 15.5).

```
tmp = ACF(mle.rbc2)
plot(ACF ~ lag, data = tmp)
lines(0:15, 0.7088^(0:15))
```

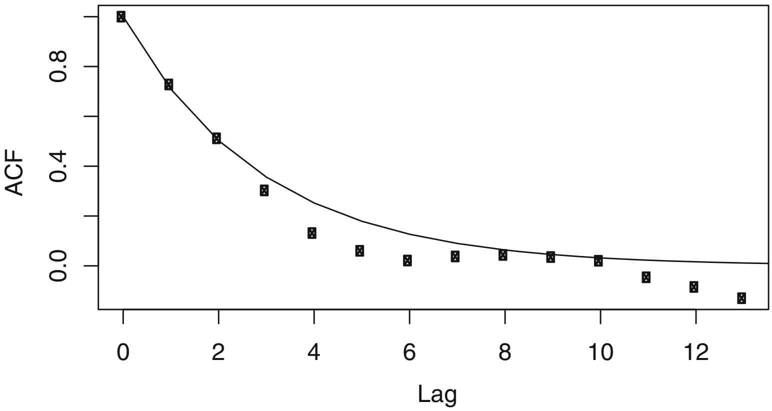

Fig. 15.5 An ACF plot of the fitted and observed serial dependence for the repeated measures regression model

Moreover, a formal likelihood-ratio test provided by the `anova` function reveals that the correlated error model provides a significantly better fit to the data:

```
anova(mle.rbc, mle.rbc2)

  ##           Model df      AIC      BIC    logLik
  ## mle.rbc       1 32 3834.369 3994.583 -1885.184
  ## mle.rbc2      2 33 3202.510 3367.731 -1568.255
  ##             Test  L.Ratio p-value
  ## mle.rbc
  ## mle.rbc2 1 vs 2 633.8586  <.0001
```

Statistically, the time-by-treatment interaction model, rather than the additive model, is better still:

```
options(width=50)
mle.rbc3=lme(RBC~Treatment*ordered(Day), random=
     ~1|Ind2, data=RBC, correlation=corAR1(form=
     ~Day|Ind2), na.action=na.omit, method="ML")
anova(mle.rbc2, mle.rbc3)
```

```
##          Model  df      AIC      BIC    logLik
## mle.rbc2     1  33 3202.510 3367.731 -1568.255
## mle.rbc3     2 153 3163.654 3929.679 -1428.827
##            Test  L.Ratio p-value
## mle.rbc2
## mle.rbc3 1 vs 2 278.8557  <.0001
```

Finally we can plot the predicted values against time (filtering out predictions for the missing values in the original data) (Fig. 15.6). There is a distinct ordering in the virulence of the strains:

```
pr=predict(mle.rbc3)
RBC$pr=NA
RBC$pr[!is.na(RBC$RBC)]=pr
plot(RBC$pr~RBC$Day, col=as.numeric(RBC$Treatment),
     pch=as.numeric(RBC$Treatment),xlab="Day",
     ylab="RBC count")
legend("bottomright",
     legend=c("AQ", "AT", "BC", "CB", "Control", "ER"),
     pch=unique(as.numeric(RBC$Treatment)), col=1:6)
```

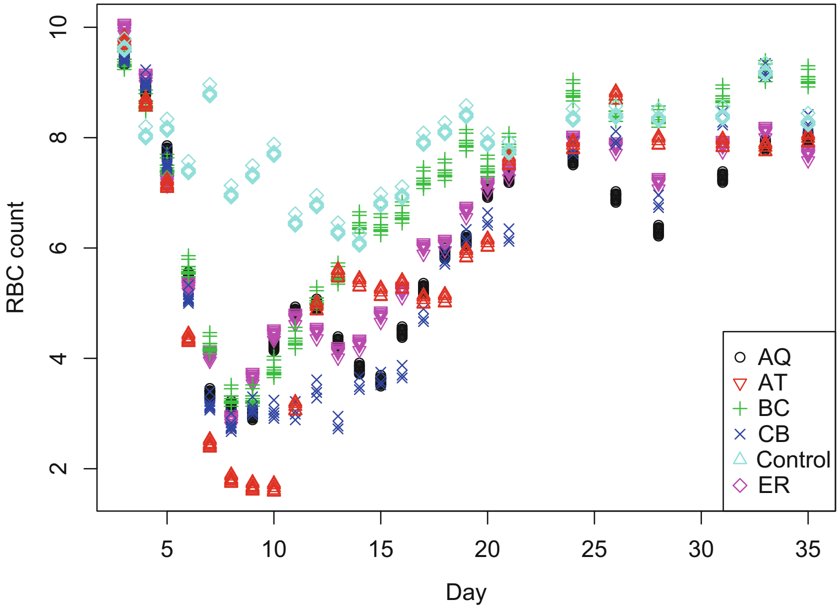

Fig. 15.6 Predicted and observed for the repeated measures RBC data

Modeling time as an ordered factor is quite parameter wasteful (the full interaction model has 153 parameters). A flexible yet more economic approach may be to model time using smoothing splines. The following example uses a B-spline with 5 degrees-of-freedom (Fig. 15.7). The qualitative features are similar to the more parameter rich model (Fig. 15.6)

```
require(splines)
mle.rbc4=lme(RBC~Treatment*bs(Day, df=5), random=
   ~1|Ind2, data=RBC, correlation=corAR1(form=
   ~Day|Ind2), na.action=na.omit, method="ML")
pr=predict(mle.rbc4)
RBC$pr=NA
RBC$pr[!is.na(RBC$RBC)]=pr
plot(RBC$pr~RBC$Day, col=as.numeric(RBC$Treatment),
   pch=as.numeric(RBC$Treatment),  xlab="Day",
   ylab="RBC count")
legend("bottomright",
legend=c("AQ", "AT", "BC", "CB", "Control", "ER"),
   pch=unique(as.numeric(RBC$Treatment)), col=1:6)
```

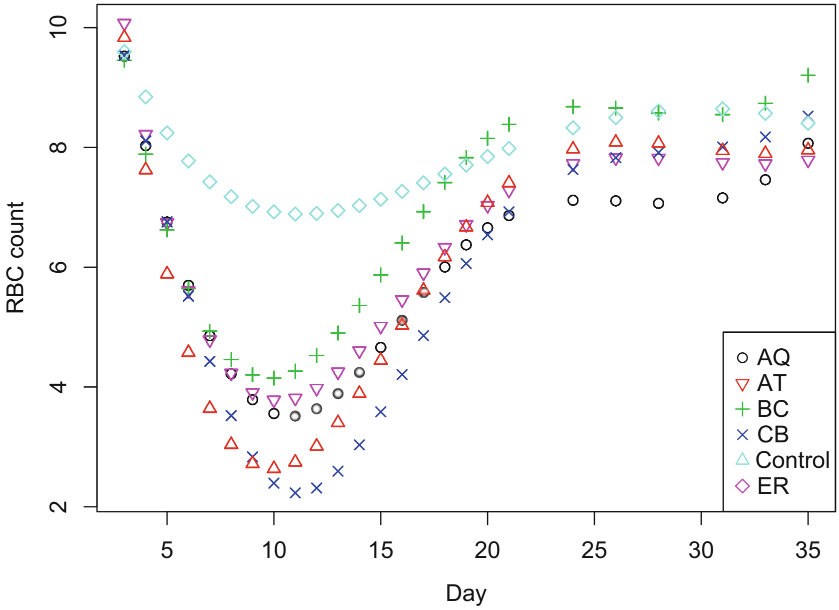

Fig. 15.7 Predicted and observed for the repeated measures RBC data using a spline model in time

15.4 *B. bronchiseptica* in Rabbits

Bordetella bronchiseptica is a respiratory infection of a range of mammals (e.g., Bjørnstad and Harvill 2005). Its congeners, *B. pertussis* and *B. parapertussis*, cause whooping cough in humans, but *B. bronchiseptica* is usually relatively asymptomatic (though it can cause snuffles in rabbits and kennel cough in dogs). The data comes from a commercial rabbitry which breeds NZW rabbits to study transmission paths in the colony. The data is from the same study as we used to study the age-specific force of infection in Sect. 4.3. Nasal swabs of female rabbits and their young were taken at weaning (∼4 weeks old). A total of 86 does and 408 kits were included in the study (Long et al. 2010).

```r
data(litter)
```

To investigate if (a) offspring of the infected mothers have an increased instantaneous risk of becoming infected and (b) if offspring of the same litter tended to have the same infection fate because of within-litter transmission, we use a random effect (generalized linear mixed model, GLMM) logistic regression, with litter as a random effect. We first do some data formatting.

```r
tdat=data.frame(lsize=as.vector(table(litter$Litter)),
  Litter=names(table(litter$Litter)),
  anysick=sapply(split(litter$sick,litter$Litter),sum))
ldat=merge(litter, tdat, by="Litter")
ldat$othersick=ldat$anysick-ldat$sick
ldat$anyothersick=ldat$othersick>0
ldat$X=1:408
```

Here, the concern is with whether littermates share correlated fates. Unlike for spatial or temporal autocorrelation, there are no canned functions to quantify this correlation. However, following our discussion of autocorrelation in Sect. 13.3, it is easy to customize our own calculations. In the below, the first double-loop makes a sibling-sibling "contact-matrix," `tmp`, that flags kits according to litter membership. After, `tmp2` rescales the binary `sick` vector that flags whether or not an animal was infected, and `tmp3` generates the correlation matrix. Finally `mean(tmp3*tmp)` provides the within-litter autocorrelation in infection status averaged across all litters.

```r
tmp=matrix(NA, ncol=length(ldat$Litter),
     nrow=length(ldat$Litter))
for(i in 1:length(ldat$Litter)){
     for(j in 1:length(ldat$Litter)){
        if(ldat$Litter[i]==ldat$Litter[j]){
          tmp[i,j]=1
        }
    }
}
diag(tmp)=NA
tmp2=scale(ldat$sick)[,1]
tmp3=outer(tmp2, tmp2, "*")
mean(tmp3*tmp, na.rm=TRUE)

  ## [1] 0.5302508
```

The within-litter correlation of 0.53 represents a substantial interdependence among littermates. Since the response variable is binary (infected vs noninfected) we cannot use `lme`. Instead we use the `lmer`-function from the `lme4`-package and specify using the "family" argument that the response is binomial. Using AICs we contrast the fit with within-litter correlation (`fitL`) with the fit that assumes independence (`fit0`); The appropriate independence fit is generated by declaring that each of the 408 individuals are in their own group (variable *X* in the data set).

```
require(lme4)
fitL=glmer(sick~msick+lsize+Facility+anyothersick+
     (1|Litter), family=binomial(), data=ldat)
fit0=glmer(sick~msick+lsize+Facility+anyothersick+
     (1|X), family=binomial(), data=ldat)
AIC(fitL, fit0)

  ##       df      AIC
  ## fitL   7 291.0263
  ## fit0   7 316.5853
```

The litter-dependent model is clearly best (no surprise given the strong empirical intra-litter correlation). The summary of the best model reveals that the key predictor of infection fate is whether or not a sibling was infected (`anyothersickTRUE`). The infection status of the mother was insignificant. The mixed-effect logistic regression thus reveals that the most important route of infection is likely to be sib-to-sib transmission (Long et al. 2010).

```
summary(fitL, corr = FALSE)

  ## Generalized linear mixed model fit by maximum
  ##   likelihood (Laplace Approximation) [glmerMod]
  ##  Family: binomial  ( logit )
  ## Formula:
  ## sick ~ msick + lsize + Facility + anyothersick +
  ##     (1 | Litter)   Data: ldat

  ##
```

```
##      AIC      BIC   logLik deviance df.resid
##    291.0    319.1   -138.5    277.0      400
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7277 -0.3199 -0.1333 -0.0386 13.2186
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Litter (Intercept) 2.077    1.441
## Number of obs: 407, groups:  Litter, 52
##
## Fixed effects:
##                  Estimate Std. Error z value
## (Intercept)      -3.43236    2.32298  -1.478
## msick             2.74171    1.65447   1.657
## lsize            -0.37908    0.19153  -1.979
## FacilityT3        1.15833    0.80626   1.437
## FacilityT9       -0.01773    0.68553  -0.026
## anyothersickTRUE  2.88387    0.71564   4.030
##                  Pr(>|z|)
## (Intercept)        0.1395
## msick              0.0975 .
## lsize              0.0478 *
## FacilityT3         0.1508
## FacilityT9         0.9794
## anyothersickTRUE 5.58e-05 ***
## ---
## Signif. codes:
## 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```