Supplementary Materials : Regression with archaeological count data

Brian F. Codding, Department of Anthropology, University of Utah
Simon C. Brewer, Department of Geography, University of Utah

2024-02-27

Contents

This document provides a companion to the paper "Regression with archaeological count data" in *Advances in Archaeological Practice*. All data and code required to reproduce the examples in the paper are available below.

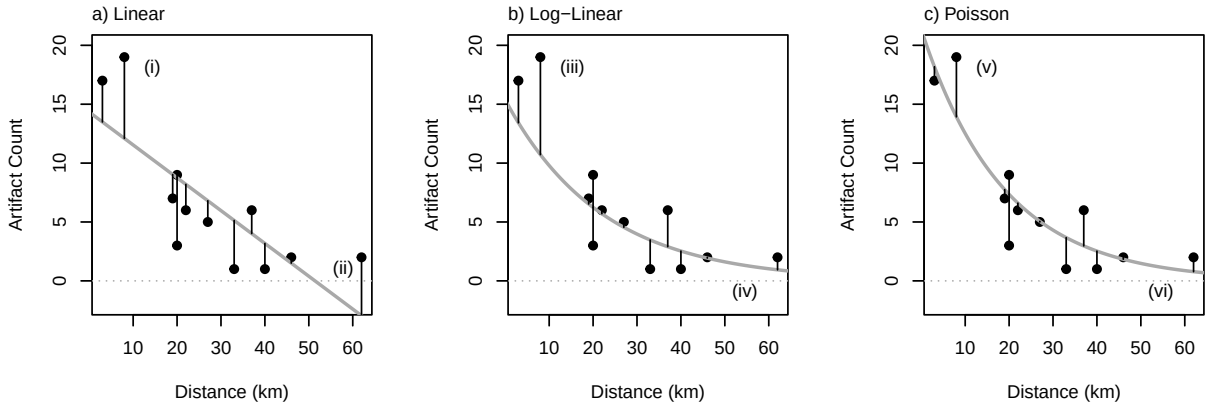
As discussed in the introduction of the paper, regression in archaeology often relies on ordinary least squares regression (see Table 1 in the main manuscript). However, there are several reasons why this may not be the best approach when working with counts. After a brief review comparing count regression with least squares regression, we introduce a suggested workflow (Figure 1) for fitting and checking count regression models. We then illustrate different paths through this workflow with five case studies. These include:

1. Counts of Yurok houses by village area (where Poisson regression works OK)
2. Counts of household members by household size in Guatemala (where we fail to reject the null hypothesis)
3. Counts of active ceramic vessels per household by the number of adult residents (where overdispersion leads to a negative binomial)
4. Counts of cattle bones across time periods (where sampling windows vary with categorical predictors)
5. Counts of projectile point types per time period vary with proxies of precipitation and temperature (where multivariate analysis is needed)

Count regression vs. Least Squares regression

Here we draw on a hypothetical example of how counts of obsidian artifacts vary with distance from the volcanic source to illustrate the limitations of OLS and the benefits of count regression.

```
##    dist.km count
## 1        3    17
## 2        8    19
## 3       19     7
## 4       20     9
## 5       20     3
## 6       22     6
## 7       27     5
## 8       33     1
## 9       37     6
## 10      40     1
## 11      46     2
## 12      62     2
```



Results of a) linear (ordinary least squares [OLS]), b) log-linear (OLS with a logged response variable), and c) Poisson regression predicting counts of obsidian artifacts across hypothetical archaeological sites as a function of the distance from the volcanic source. Black dots show the observed values at each site. Grey solid lines show the predicted model fit. Black vertical lines show the distance between the predicted and observed value for each site (the residuals), which the model is trying to minimize should. Grey horizontal dashed lines indicate zero.

As discussed in the text, each model does progressively better at describing the data (Table 1 in the main text for definitions of terms):

a) The linear model has several issues: it under-predicts both high values (see i) and low values (see ii), which is a sign of a poor fit and patterning in the residuals; moreover, it predicts counts below zero (see ii), which is impossible. Overall it has a reasonable goodness-of-fit, accounting for 60% of the deviance in model fit (likelihood r-squared $[r_l^2] = 0.60$).
b) The log-linear model does a bit better: while it also under-estimates high values (see iii), it does not predict counts below zero (see iv). It has a slightly lower goodness-of-fit than the linear model ($r_l^2 = 0.57$).
c) The Poisson model does the best job: it more closely predicts the high values (see v) and does not predict values below zero (see vi). It has the highest goodness-of-fit ($r_l^2 = 0.57$). In this case, count regression is a better choice than a linear or log-linear model for describing the relationship and making predictions about how obsidian tool count declines as a function of distance to the volcanic source.

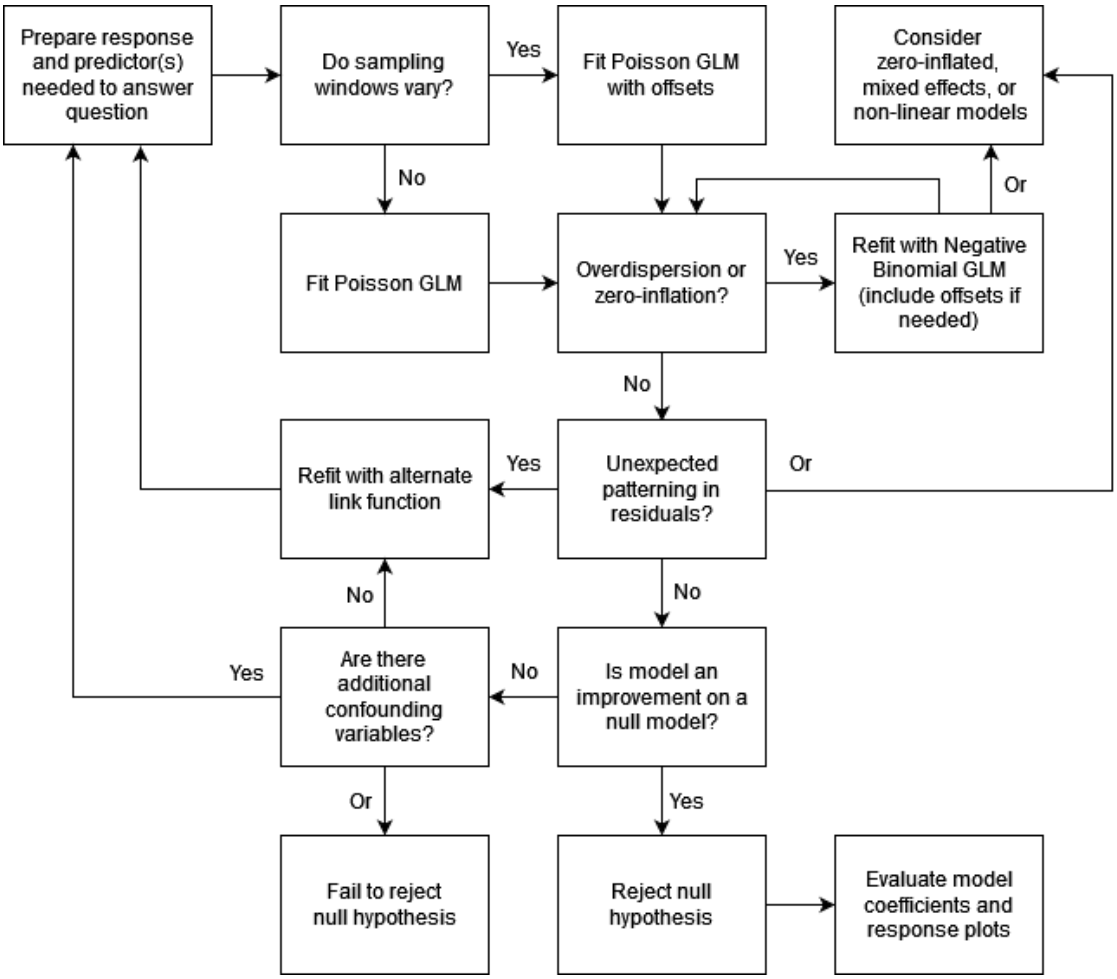


Figure 1: Flowchart outlining recommended procedures for fitting and evaluating count models. This is not exhaustive, but provides guidance on model fitting and diagnostics that archaeologists are likely to encounter. An example flowchart for multiple regression is available below

1 Poisson regression: Yurok villages

Question: How do house counts vary with village (or archaeological site) size?

To answer this question, we draw on data from Cook and Treganza (1950) who report investigations from Waterman (1920) on how the counts of houses (and inferred people) vary with village size across historic Yurok villages in northwestern California. These data are commonly used with other data sets examining how hunter-gatherer population size may vary with village size to make predictions about residential population size from archaeological site size (e.g., Yellen 1977; Codding et al. 2016). These data can also help estimate how many structures may be expected in a site of a specific size.

Spoiler alert: this first example illustrates when a simple Poisson regression works OK.

```
Waterman
```

```
##      Village  Area Houses Population
## 1      Omen     766      4         24
## 2    Rekwoi   19314     22        132
## 3    Woxero    2251      6         36
## 4    Woxtek    3942      7         42
## 5     Qootep   4896     19        114
## 6    Pekwan    5690     16         96
## 7      Meta    2223      5         30
## 8     Murek   16002     18        108
## 9       Saa    4719      9         54
## 10    Kepel    3068      9         54
## 11    Qenek    1802      6         36
## 12   Wahsek    2132      9         54
## 13 Weitspus   11589     21        126
## 14    RLrgr    1338      5         30
## 15 Pekwutul    2174      6         36
## 16   Tsurai    6349     11         66
```

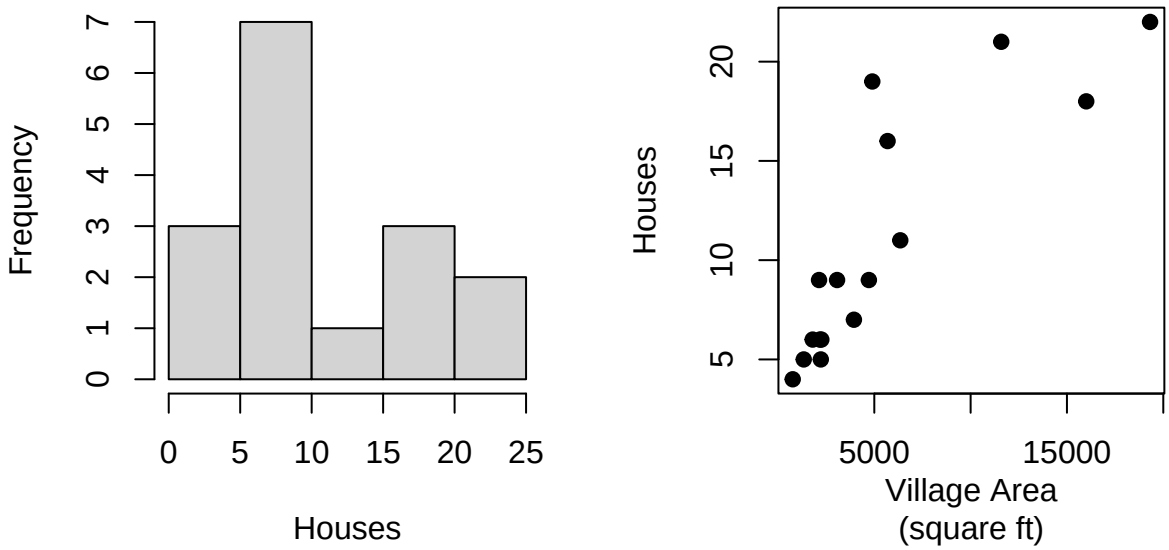
Variables include:

- Village = Yurok village name
- Area = area in square feet (note: their table says "Area (in sq. mi.)" but this cannot be the case and Waterman's original maps indicate feet as the unit).
- Houses = the number of houses per village
- Population = the village population estimated by multiplying house count by 6, which they say is "the most likely value for the mean number of inhabitants per house" (Cook and Treganza 1950: 232).

1.1 Exploratory data analysis

```
#pdf("Figure3.pdf", height = 4, width = 7)

par(pty = "s", mfrow = c(1, 2))

with(Waterman,
     hist(Houses,
          main = NA,
          xlab = "Houses"
          )
     )

with(Waterman,
     plot(Houses ~ Area,
```

```
        ylab = "Houses",
        xlab = "Village Area\n(square ft)",
        pch = 19
        )
    )
```



1.2 Fit a Poisson GLM

Note: here we follow an arbitrary convention in naming model objects where the first terms before the period indicate the model and family, and the second set after the period indicate the response and predictor(s). Subsequent objects derived from the model object will have an additional period followed by the next object type.

```
glm_Pois.House_Area <- glm(Houses ~ Area,
                           family = poisson(link = "log"),
                           data = Waterman
                           )
```

1.2.1 Model diagnostics

Before checking the model results, we should run some diagnostics.

1.2.1.1 Overdispersion First, check for overdispersion: if the variance is greater than the mean. Following Bolker et al. (2022) we do this by dividing the sum of Pearson's squared residuals over the residual degrees of freedom. The value should be close to one.

This can be done as such:

```
sum(residuals(model.object, type="pearson")^2)/df.residual(model.object)
```

Or, we can use the function written by Bolker et al. (2022) that includes the estimation of a p-value as well:

```
overdisp_fun <- function(model) {         #create a new function
    rdf <- df.residual(model)             #object for residual degrees of freedom (rdf)
    rp <- residuals(model,type="pearson")#object for Pearson's residuals
    Pearson.chisq <- sum(rp^2)            #sum of squared residuals (ssr)
    prat <- Pearson.chisq/rdf             #ssr divided by rdf
    pval <- pchisq(Pearson.chisq,         #chi-square test on ssr
                   df=rdf,
                   lower.tail=FALSE)
    c(chisq = Pearson.chisq,              #data for output
      ratio = prat,
      rdf = rdf,
      p = pval)
}
```

Apply the function to our model:

```
overdisp_fun(glm_Pois.House_Area)
```

```
##       chisq       ratio         rdf           p
## 21.21964206  1.51568872 14.00000000  0.09613494
```

The over-dispersion parameter (ratio) is close enough to 1, suggesting a Poisson distribution that assumes equal mean and variance is sufficient.

1.2.1.2 Zero-inflation Zero-inflation is when the model over- or under-fits zeros. We can check this by calculating the ratio of observed zeros to predicted zeros. This should also be around 1 ($\pm$ 1). In this particular case, we know this cannot be a problem as there are no zeros in the data. Nonetheless, we can calculate this by dividing the sum of observed zero cases by the sum of predicted zero cases:

```
sum(Waterman$Houses == 0)/sum(glm_Pois.House_Area$fitted.values == 0)
```

We can also make this into a function for future use:

```
zeroinfl_fun <- function(model, response) {
    obs_zero <- sum(response == 0)                  #number of zeros in observations
    mod_zero <- sum(round(model$fitted.values) == 0)#number of predicted zeros rounded
    c(obs_0 = obs_zero,
      mod_0 = mod_zero,
      ratio = obs_zero/mod_zero)
}
```

To apply this function, pass the model object and the response variables:

```
zeroinfl_fun(model = glm_Pois.House_Area, response = Waterman$Houses)
```

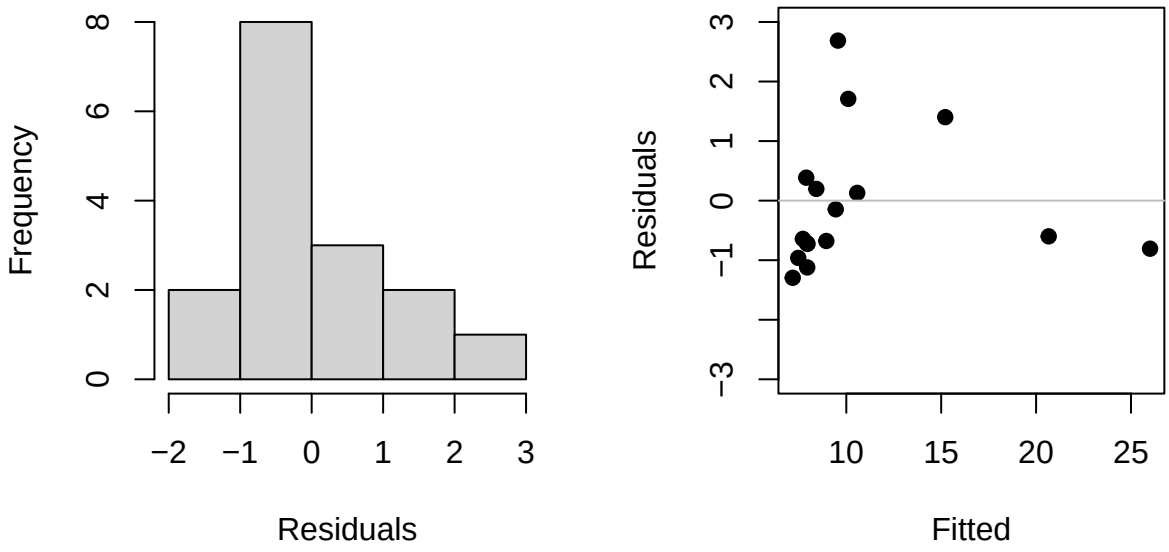
```
## obs_0 mod_0 ratio
##     0     0   NaN
```

There are no zero observations (i.e., no villages without houses), so the model does not predict any zeros. The ratio returned is "not a number" or ''NaN"'.

1.2.1.3 Residuals Now let's look at the model residuals, specifically, the deviance residuals.

```
glm_Pois.House_Area.resid <- residuals(glm_Pois.House_Area, "deviance")
```

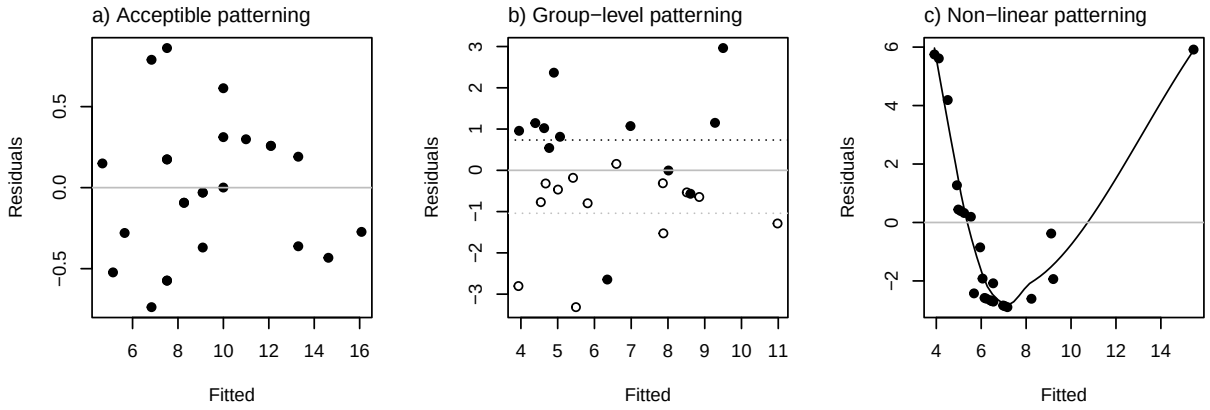
```r
par(pty = "s", mfrow = c(1,2))

hist(glm_Pois.House_Area.resid,
     xlab = "Residuals",
     main = "")

plot(glm_Pois.House_Area.resid ~ glm_Pois.House_Area$fitted.values,
     ylim = c(-3,3),
     ylab = "Residuals",
     xlab = "Fitted",
     main = "",
     pch = 19
     )
abline(h = 0, #draw a horizontal line at zero
       col = "grey")
```



Two or three villages have more houses than expected given their size. This suggests that they are densely occupied villages, which may be due to a number of factors such as physical restraints on areas suitable for houses, or social benefits to aggregation in some locations. Examining outliers such as this can lead to further research questions and bring the investigator back to the data in order to better explain such trends. For more on these villages, see Waterman (1920).

Though overall, this is not terribly bad, especially given the small sample size.

Sidebar: What would "bad" residuals look like?



Examples of residual by fitted plots to examine a) acceptable pattering in Poisson residuals, b) patterning structured by between-group variation (dashed lines show group-level mean residuals), and c) patterning structured by a non-linear relationship between y and x not accounted for in the model.

1.2.2 Results

1.2.2.1 Goodness of fit

Is our model an improvement on a null model?

Fit the null model:

```
glm_Pois.House_null <- glm(Houses ~ 1,
                           family = poisson(link = "log"),
                           data = Waterman
                           )
```

Compare to the proposed model using a likelihood ratio test with the `anova` function:

```
anova(glm_Pois.House_null, #note: the models must be nested from less to more complex
      glm_Pois.House_Area,
      test = "LRT"         #specify likelihood ratio test (lrt)
)
```

```
## Analysis of Deviance Table
##
## Model 1: Houses ~ 1
## Model 2: Houses ~ Area
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        15     51.247
## 2        14     19.103  1   32.144 1.432e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yes, the model is a significant improvement on a null model with only the y-intercept. If the proposed model were not an improvement, the `Pr(>Chi)` value would be blank.

Now calculate the likelihood r-squared (r_l^2) value. This can be done with the model object by either calculating one minus the residual deviance over the null deviance, or by the model deviance over the null deviance.

```r
with(glm_Pois.House_Area, 1 - (deviance/null.deviance))
```

```
## [1] 0.6272335
```

```r
with(glm_Pois.House_Area, (null.deviance-deviance)/null.deviance)
```

```
## [1] 0.6272335
```

This shows that village area accounts for about 62% of the deviance, or variation, in house count.

1.2.2.2 Coefficients Now that we know the model is doing a reasonable job, we can examine what it tells us about the relationship between the number of houses and village size. To do this we pass the model object to the `summary` function:

```r
summary(glm_Pois.House_Area)
```

```
##
## Call:
## glm(formula = Houses ~ Area, family = poisson(link = "log"),
##     data = Waterman)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.918e+00  1.195e-01  16.052  < 2e-16 ***
## Area        6.940e-05  1.151e-05   6.029 1.65e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 51.247  on 15  degrees of freedom
## Residual deviance: 19.103  on 14  degrees of freedom
## AIC: 88.498
##
## Number of Fisher Scoring iterations: 4
```

The summary function returns a lot of information. Let's focus on a few important points:

- Call: this repeats the model call.
- Deviance Residuals: this shows the quantiles (minimum, first quartile, median, third quartile, and maximum value) of deviance residuals. These are the same values we plotted above in a histogram.
- Coefficients: these are very important, they show all the model coefficients for the intercept and the predictor variables. It is good practice to produce this table in your manuscript or supplementary materials when publishing results.
  - Estimate: this is the coefficient for each term (see main text). Note: as the mode link is logarithm, take the exponent of the coefficient for it to be on the scale of the response (counts).
  - Std. Error: this is the standard error of the coefficient.
  - z value: this is the test statistic, for a Poisson it will be a "z" value.
  - Pr(>|z|): this is the p-value which reports the probability of a Type-I error, or a false positive, specifically, falsely rejecting a true null hypothesis. While researchers often pay a lot of attention to p-values, they are prone to several issues and misinterpretations. Nonetheless, it is good to carefully consider what they imply, and to report them.
- Signif. codes: this is just a key to the symbols that correspond to each p-value at standard thresholds or alpha values.
- (Dispersion parameter...): this reports the dispersion parameter of the model, note it will not be 1 with all models such as negative binomial models (see below).

- Null deviance: this is the total deviance from the null model, calculated as the sum of log-likelihoods of each y observation given only information about the y-intercept.
- Residual deviance: this is the deviance remaining after accounting for the predictor variables, calculated as the sum of log-likelihoods of each observation given information about the y-intercept and predictor variable(s).
- . . . degrees of freedom: the number of independent values, should be n-2 for a bivariate model accounting for the slope and intercept.
- AIC: this is the Aikake's information criteria value, which helps approximate how well a model fits the data. Lower values indicate a better fit.
- Number of Fisher. . . : This reports how many rounds were needed to fit the model.

We can extract each coefficient from the object. For example, let's pull the coefficient for the predictor variable and take it's exponent:

```
exp(glm_Pois.House_Area$coefficients[2])
```

```
##     Area
## 1.000069
```

As discussed in the text, this means that the number of houses increases by 0.0007% with each unit increase in the predictor, or by each square foot. This may not seem like a lot, but remember the range of square feet in village size is about 800 - 19000.

1.2.2.3 Prediction Now we can use prediction to estimate counts from other values. For example, if working on an archaeological site that is 4,000 square feet in area, how many house features should you expect to find?

To answer this, we again use the `predict` function. Note that we have to supply a `data.frame` with the new data we'd like to predict from. The variable(s) in this data frame must have exactly the same names as the variables in the model, and must include all variables in the model.

```
predict(glm_Pois.House_Area,
        newdata = data.frame(Area = 4000),
        type = "response" #*
        )
```

```
##        1
## 8.984357
```

```
#* for simplicity sake, we are asking for the prediction to be in the response scale
#* (i.e., counts) but this is not the best practice when also taking the standard
#* error in order to plot confidence intervals; this is discussed more below
#* and in Simpson (2018a)
```

If similar processes are structuring the relationship between house number and village size, we should expect to find about 9 houses at a village site of 4,000 square feet.

To examine how the predicted number of houses varies across the full range of village areas, we can predict along a sequence (note: if your values have missing or NA values, include `na.rm = TRUE` in the `min` and `max` call used to identify the range of values; this is true for `median` or `mean` as well if used to hold additional variables constant at the central tendency in multivariate models; see Section 5 below).

```
#create a vector across the range of x values
area_seq <- seq(from = min(Waterman$Area, na.rm = TRUE) - 1000, #from the min -1000 -NAs
                to = max(Waterman$Area, na.rm = TRUE) + 1000,   #to the max +1000 -NAs
                by = 1000 #for each 1000 square feet
                  )

#predict ceramic counts a cross the range of x
```

```
house_pred <- predict(glm_Pois.House_Area, #the model object
                      newdata = data.frame(Area = area_seq), #*
                       se = TRUE,       #include the standard error of the prediction
                       type = "link"    #be sure to predict on the link scale
                       )

#* note, the new data must be a data frame with columns corresponding to
#* each variable in the model, each of which must have the exact same name
#* as the model variable.
```

Check out the resulting object. For each value in the prediction sequence (`area_seq`), the object has a predicted fit and standard error around the fit. Note these are on the link scale, so they are the logged values:

```
house_pred
```

```
## $fit
##        1        2        3        4        5        6        7        8
## 1.901654 1.971052 2.040450 2.109848 2.179246 2.248644 2.318042 2.387440
##        9       10       11       12       13       14       15       16
## 2.456838 2.526236 2.595634 2.665032 2.734430 2.803828 2.873226 2.942624
##       17       18       19       20       21
## 3.012022 3.081420 3.150818 3.220216 3.289614
##
## $se.fit
##          1          2          3          4          5          6          7
## 0.12156780 0.11281644 0.10460079 0.09705698 0.09035345 0.08468997 0.08028694
##          8          9         10         11         12         13         14
## 0.07735988 0.07607934 0.07652802 0.07867634 0.08239145 0.08747395 0.09370160
##         15         16         17         18         19         20         21
## 0.10086250 0.10877250 0.11728011 0.12626459 0.13563122 0.14530611 0.15523163
##
## $residual.scale
## [1] 1
```

We can use these to make a plot of the model fit, but first, we need one more thing. We have to save the inverse link function from the model object (note, this will be the same for all Poisson and negative binomial GLMs that have a log link). We use this to convert the predicted value, which is logged, back to the response scale.

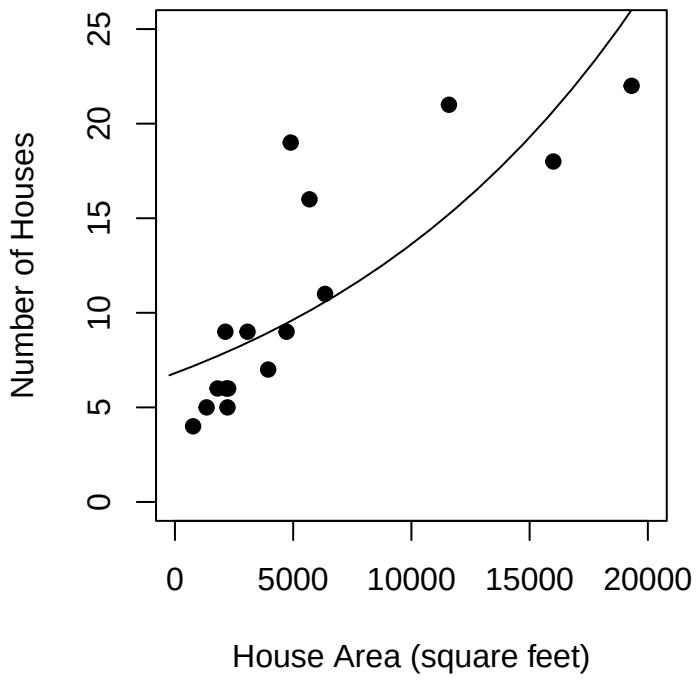
```
family(glm_Pois.House_Area)
```

```
##
## Family: poisson
## Link function: log
```

```
#this function tells you the model family and link, it also
#includes an inverse link function

inv_link.glm_Pois <- family(glm_Pois.House_Area)$linkinv #save model inverse link
```

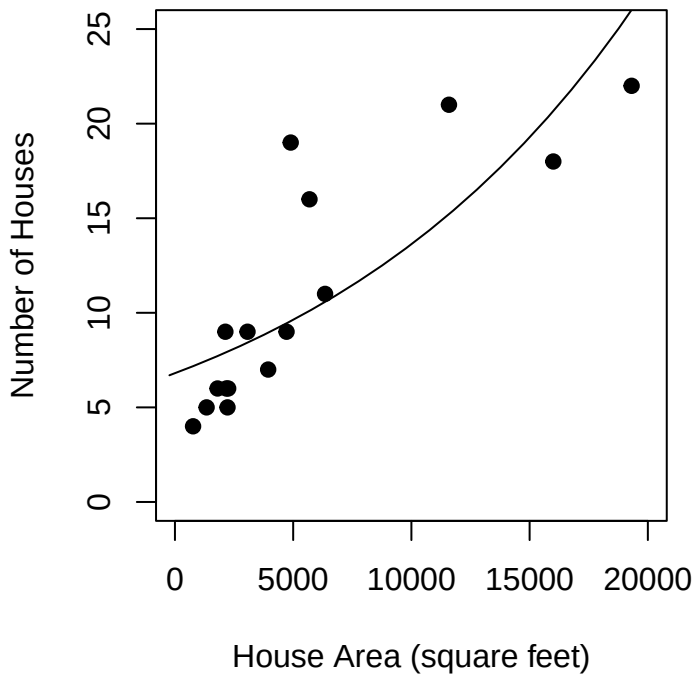
Plot the predicted model fit:

```
par(pty = "s")

with(Waterman,
     plot(Houses ~ Area,
          pch = 19,
          xlim = c(0, 20000),
          ylim = c(0, 25),
          xlab = "House Area (square feet)",
          ylab = "Number of Houses",
          type = "p"
          )
     )

#this plots the predicted model fit
lines(inv_link.glm_Pois(house_pred$fit) ~ area_seq)
```



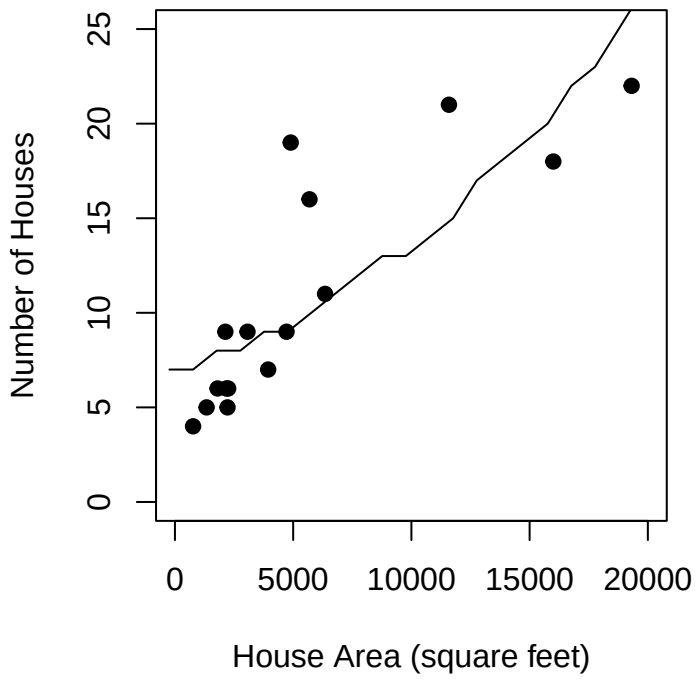
In this case, we would get the same thing if we plotted the exponent of the predicted fit:

```
par(pty = "s")

with(Waterman,
     plot(Houses ~ Area,
          pch = 19,
          xlim = c(0, 20000),
          ylim = c(0, 25),
          xlab = "House Area (square feet)",
          ylab = "Number of Houses",
          type = "p"
          )
     )

#this plots the predicted model fit
lines(exp(house_pred$fit) ~ area_seq) #using exp instead of inv_link
```



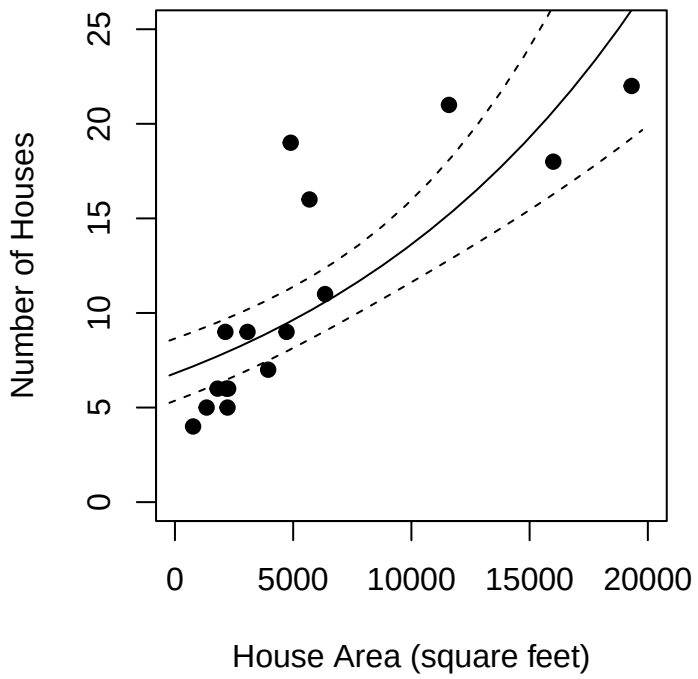
Note, we could also round the prediction so that we display the response as counts:

```
par(pty = "s")

with(Waterman,
     plot(Houses ~ Area,
          pch = 19,
          xlim = c(0, 20000),
          ylim = c(0, 25),
          xlab = "House Area (square feet)",
          ylab = "Number of Houses",
          type = "p"
          )
     )

#this plots the predicted model fit
lines(round(inv_link.glm_Pois(house_pred$fit), 0) ~ area_seq)
```



We can add the 95% confidence intervals by adding and subtracting two times the standard error (or, approximately 95%) to the predicted response:

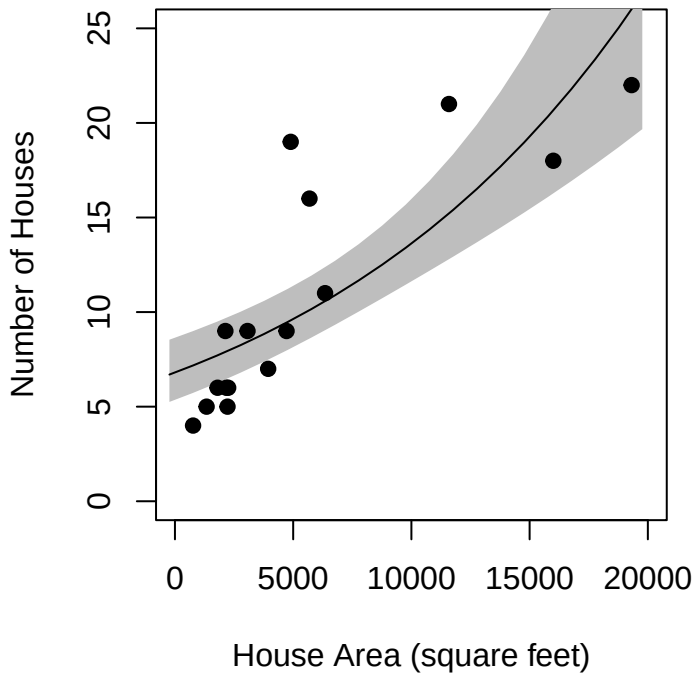
```
par(pty = "s")

with(Waterman,
     plot(Houses ~ Area,
          pch = 19,
          xlim = c(0, 20000),
          ylim = c(0, 25),
          xlab = "House Area (square feet)",
          ylab = "Number of Houses",
          type = "p"
          )
     )

#this plots the predicted model fit
lines(inv_link.glm_Pois(house_pred$fit) ~ area_seq)

#this plots the predicted model fit plus two times the standard error, approx. 95%
lines(inv_link.glm_Pois(house_pred$fit + 2 * house_pred$se.fit) ~ area_seq, lty = 2)

#this plots the predicted model fit minus two times the standard error, approx. 95%
lines(inv_link.glm_Pois(house_pred$fit - 2 * house_pred$se.fit) ~ area_seq, lty = 2)
```



Note: this is where we need to use the inverse link as opposed to taking the exponent. For an excellent post on why, see Simpson (2018a).

We can also plot the confidence intervals as a polygon:

```
par(pty = "s")

plot(NA,  #start with a blank plot
     pch = 19,
     xlim = c(0, 20000),
     ylim = c(0, 25),
     xlab = "House Area (square feet)",
     ylab = "Number of Houses",
     type = "p"
     )

polygon(y = c(inv_link.glm_Pois(house_pred$fit + 2 * house_pred$se.fit),
              rev(inv_link.glm_Pois(house_pred$fit - 2 * house_pred$se.fit))),
        x = c(area_seq, rev(area_seq)),
        col = "grey",
        border = F
        )
#note the use of c() to stitch together the back and forth, and the use of rev to flip
#the vector. Imagine you are specifying all the points to draw outlines of the polygon.

lines(inv_link.glm_Pois(house_pred$fit) ~ area_seq)

#add the data points
with(Waterman,
     points(Houses ~ Area, pch = 19)
)
```



There are clearly some outliers here, specifically Qo'-o-tep and Pe'kwan villages which appear to have more densely clustered houses than other villages (see Waterman 1920: Maps 14-15). This residual variation may inspire the researcher to look more into this to understand what might be behind this behavioral difference.

---

2 Poisson regression: Guatemalan households

Question: How do the number of residents vary with house size?

To answer this, we draw on Loucky's data reporting occupants and size from two maize-growing communities in Guatemala (San Juan la Laguna and Santa Catarina Palopó) taken from a random samples of households in each village (Kolb 1985, first presented by Kolb and Loucky in 1974 at the Society for American Archaeology meeting). Data are from Kolb (1985: Tables 9 and 10; https://www.jstor.org/stable/2743081). Spoiler alert: this is an example where we fail to reject the null hypothesis.

```
Loucky
```

```
##                COMMUNITY HOUSEHOLD.ID FAMILY.TYPE NUM.PERSONS HOUSE.SIZE.M2
## 1     SAN JUAN LA LAGUNA            1           N           4         52.67
## 2     SAN JUAN LA LAGUNA            2           N           2         16.85
## 3     SAN JUAN LA LAGUNA            3           N           4         28.09
## 4     SAN JUAN LA LAGUNA            4           N           6         16.85
## 5     SAN JUAN LA LAGUNA            5           N           6         35.29
## 6     SAN JUAN LA LAGUNA            6           N           7         35.82
## 7     SAN JUAN LA LAGUNA            7           E           7         77.95
## 8     SAN JUAN LA LAGUNA            8           E           6         35.11
## 9     SAN JUAN LA LAGUNA            9           E           9         30.90
## 10    SAN JUAN LA LAGUNA           10           N           6         24.58
## 11    SAN JUAN LA LAGUNA           11           E           5         21.07
## 12    SAN JUAN LA LAGUNA           12           N           2         47.05
## 13    SAN JUAN LA LAGUNA           13           E           5         33.36
## 14    SAN JUAN LA LAGUNA           14           N           6         55.48
## 15    SAN JUAN LA LAGUNA           15           N           6         50.56
## 16    SAN JUAN LA LAGUNA           16           N           7         54.77
## 17    SAN JUAN LA LAGUNA           17           N           4         39.33
## 18    SAN JUAN LA LAGUNA           18           N           7         54.07
## 19    SAN JUAN LA LAGUNA           19           N           6         24.23
## 20    SAN JUAN LA LAGUNA           20           N           6         64.25
## 21    SAN JUAN LA LAGUNA           21           N           4         67.06
## 22    SAN JUAN LA LAGUNA           22           E           2         61.09
## 23    SAN JUAN LA LAGUNA           23           N           7         28.97
## 24    SAN JUAN LA LAGUNA           24           N           6         28.09
## 25 SANTA CATARINA PALOPO            1           N           5         44.25
## 26 SANTA CATARINA PALOPO            2           N           6         84.24
## 27 SANTA CATARINA PALOPO            3           N           7         28.07
## 28 SANTA CATARINA PALOPO            4           N           4         28.08
## 29 SANTA CATARINA PALOPO            5           N           7         16.87
## 30 SANTA CATARINA PALOPO            6           N           2         14.04
## 31 SANTA CATARINA PALOPO            7           N           2         42.14
## 32 SANTA CATARINA PALOPO            8           E           2         28.08
## 33 SANTA CATARINA PALOPO            9           N           5         33.70
## 34 SANTA CATARINA PALOPO           10           N           2         21.06
## 35 SANTA CATARINA PALOPO           11           N           8         14.08
```

Variables include:

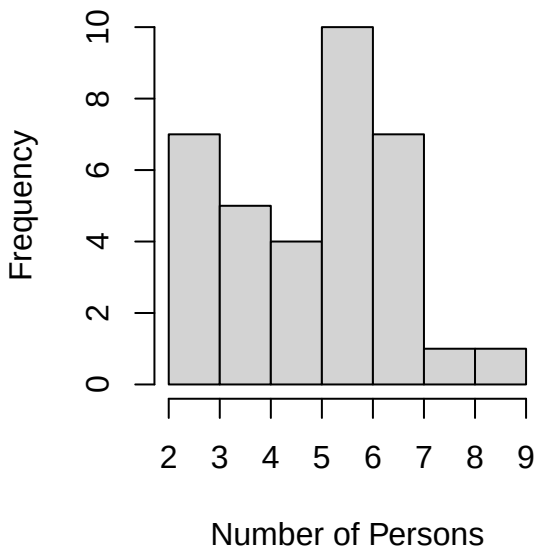
- COMMUNITY = San Juan la Laguna or Santa Catarina Palopó
- HOUSEHOLD.ID = arbitrary identifier
- FAMILY.TYPE = Nuclear or Extended
- NUM.PERSONS = Number of individuals residing in the house
- HOUSE.SIZE.M2 = Size of house in meters squared

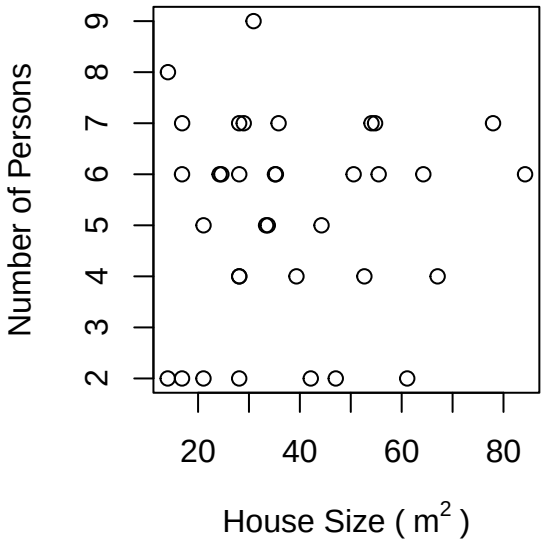
Other variables in Kolb (1985: Tables 9 and 10) not included here:

- AREA.PER.PERSON.M2 = Per person household area
- NUM.ROOMS = Number of rooms in each household
- WALL = Construction of walls: A, adobe; C, cania (cane, vertical strips); W, wattle-and-daub; Cm, cement, B, bajareque (cane, horizontal strips)
- WALL.HL = Construction of walls simplified to indicate high (adobe or cement) vs. low investment (cane) (or mixed; M); any one building with adobe or cement marked as permenant for the household
- ROOF = Construction of roof: C, tejada (ceramic tile); L, limina (sheet roofing); T, techumbre (thatch) or paja (straw).

2.1 Exploratory data analysis

```
par(pty = "s", mfrow = c(1,2))

with(Loucky,
     hist(NUM.PERSONS,
          xlab = "Number of Persons",
          main = NA
          )
     )

with(Loucky,
     plot(NUM.PERSONS ~ HOUSE.SIZE.M2,
          ylab = "Number of Persons",
          xlab = expression("House Size ("~m^2~")")
          )
     )
```





2.2 Fit a Poisson GLM

```
glm_Pois.Persons_Area <- glm(NUM.PERSONS ~  HOUSE.SIZE.M2,
                             family = poisson,
                             data = Loucky
                             )
```

2.2.1 Diagnostics

As above, we will check for overdispersion and zero-inflation.

```
overdisp_fun(glm_Pois.Persons_Area)
```

2.2.1.1 Overdispersion

```
##      chisq      ratio        rdf          p
## 25.3801189  0.7690945 33.0000000  0.8258485
```

No meaningful overdispersion.

```
zeroinfl_fun(glm_Pois.Persons_Area, response = Loucky$NUM.PERSONS)
```

2.2.1.2 Zero-inflation

```
## obs_0 mod_0 ratio
```

```
##     0     0   NaN
```

There are no zero observations.

2.2.2 Results

Next evaluate the model fit relative to a null model and evaluate the proportion deviance explained.

```
glm_Pois.Persons_null <- glm(NUM.PERSONS ~  1,
                             family = poisson,
                             data = Loucky
                             )

anova(glm_Pois.Persons_null,
      glm_Pois.Persons_Area,
      test = "LRT"
      )
```

2.2.2.1 Goodness of fit

```
## Analysis of Deviance Table
##
## Model 1: NUM.PERSONS ~ 1
## Model 2: NUM.PERSONS ~ HOUSE.SIZE.M2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        34     28.230
## 2        33     28.119  1  0.11039   0.7397
```

The model is not a significant improvement on the null model.

```
with(glm_Pois.Persons_Area, (null.deviance-deviance)/null.deviance)
```

```
## [1] 0.003910364
```

House size only accounts for 0.3% of the variation in the number of residents.

```
summary(glm_Pois.Persons_Area)
```

2.2.2.2 Coefficients

```
##
## Call:
## glm(formula = NUM.PERSONS ~ HOUSE.SIZE.M2, family = poisson,
##     data = Loucky)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.584871   0.175754   9.018   <2e-16 ***
## HOUSE.SIZE.M2 0.001371   0.004115   0.333    0.739
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 28.230  on 34  degrees of freedom
## Residual deviance: 28.119  on 33  degrees of freedom
```

```
## AIC: 151.89
##
## Number of Fisher Scoring iterations: 4
```

Given the poor model fit, there is no need to evaluate the model further. We can reject the null hypothesis that the number of household members varies positively with house area.

What other factors may confound this relationship? Consider looking at the data set further to evaluate this.

---

3 Negative binomial regression: Michoacán pottery

Question: How do counts of active ceramic vessels vary with the number of household adults?

Michael Shott's ethnoarchaeological work examines the distribution of surviving and failed pottery across 24 households in Michoacán (Shott 2018, 2022). These data are from Shott (2022) "Shott supplementary material 1" available at https://doi.org/10.1017/aaq.2022.57. The data table used here aggregates Shott's supplementary data by household.

Spoiler alert: this case illustrates overdisperion and the need to move to a negative binomial regression.

```
Shott
```

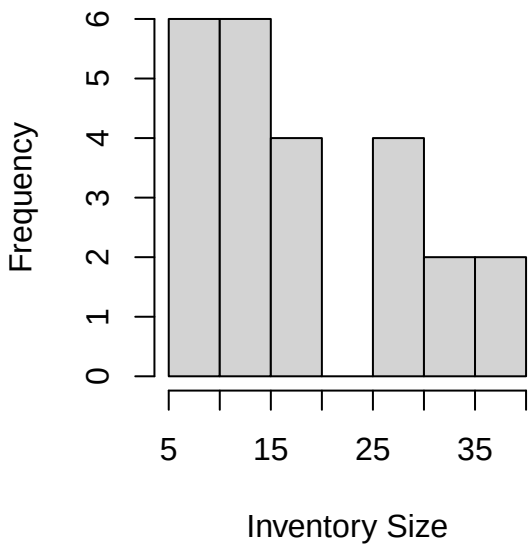
```
##    fam failed active famsiz adults volume
## 1    1     50     11      4      2     NA
## 2    2     69     26      5      4     NA
## 3    3     49     12      7      3     NA
## 4    4     10     19      3      2     NA
## 5    5     16     31     10      4    456
## 6    6     38     19      7      4     NA
## 7    7     10     16      5      2    246
## 8    8     44      7      2      2     NA
## 9    9     41     14      7      3     NA
## 10  10     35     14      1      1    341
## 11  11     33     38      7      5    275
## 12  12     15     37      8      4     NA
## 13  13     27      8      4      4     NA
## 14  14     27     28      4      4    243
## 15  15     90      5      1      1     NA
## 16  16     18     16      2      2     NA
## 17  17     20     14      3      2     NA
## 18  18     24     28      3      3     NA
## 19  19     15      6      7      2    754
## 20  20     33     32      7      3    420
## 21  21      5      7      5      4    541
## 22  22      0     11      2      2    261
## 23  23      0     27      6      5    285
## 24  24      2      9     NA     NA   1197
```

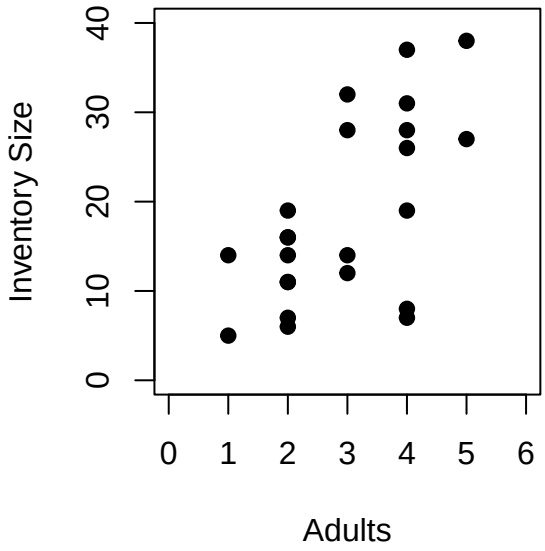
The data include:

- fam = number code for household (1-24)
- failed = failed pottery (summed from status column; 0 = failed; 1 = surviving)
- active = surviving pottery (summed from status column; 0 = failed; 1 = surviving)
- famsiz = number of co-residents
- adults = number of adults
- volume = median volume of all pottery in cc

3.1 Exploratory data analysis

```
par(pty = "s", mfrow = c(1,2))

with(Shott,
     hist(active,
          main = NA,
          xlab = "Inventory Size"
          )
     )

with(Shott,
     plot(active ~ adults,
          pch = 19,
          xlim = c(0, 6),
          ylim = c(0, 40),
          xlab = "Adults",
          ylab = "Inventory Size"
          )
     )
```





3.2 Fit a Poisson GLM

```
glm_pois.inv_adu <- glm(active ~ adults,
                        family = poisson(link = "log"),
                        dat = Shott)
```

3.2.1 Model diagnostics

Before checking the model results, we should run sum diagnostics. First, check for overdispersion. Apply the function to our model:

```
overdisp_fun(glm_pois.inv_adu)
```

```
##        chisq        ratio          rdf            p
## 7.134988e+01 3.397613e+00 2.100000e+01 2.131621e-07
```

Significant overdispersion, meaning the mean and variance are not equal. The ratio suggests the variance is 3.4 times the mean.

3.3 Refit with a negative binomial GLM

Let's retry the fit with a negative binomial model in the `MASS` library (Venables and Ripley 2002):

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

Fit the negative binomial model:

```
glm_nb.inv_adu <- glm.nb(active ~ adults,
                         dat = Shott
                         )
#note, we do not need to specify the family here, the default link is log
```

3.3.1 Model diagnostics

3.3.1.1 Overdispersion

Check if the negative binomial model fits better:

```
overdisp_fun(glm_nb.inv_adu)
```

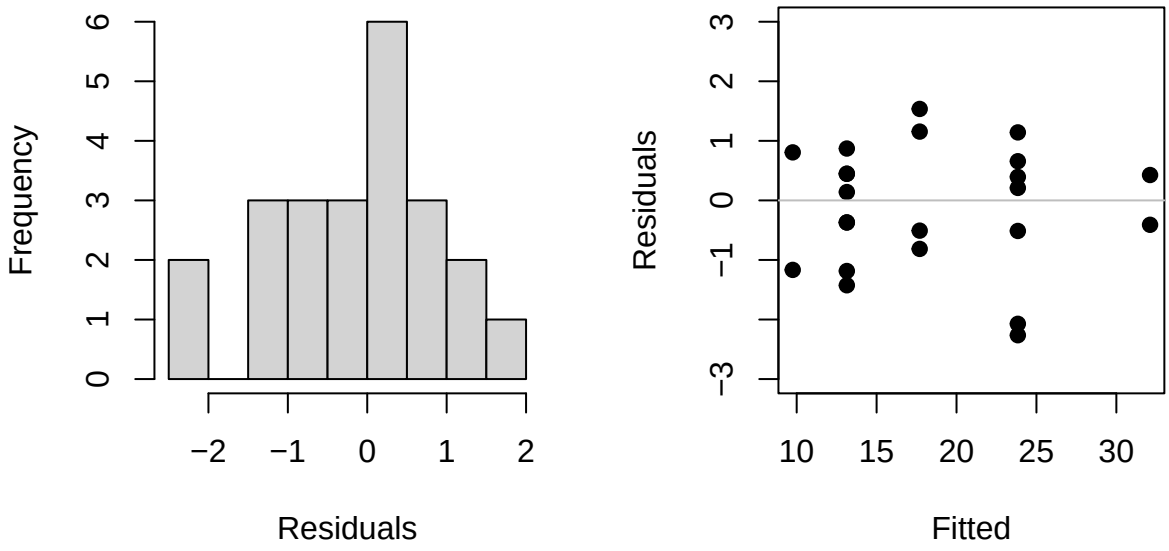
```
##      chisq      ratio        rdf          p
## 20.0899018  0.9566620 21.0000000  0.5155733
```

Looks good! This model adequately accounts for overdispersion.

3.3.1.2 Residuals

Just as with a Poisson model, we can plot the deviance residuals.

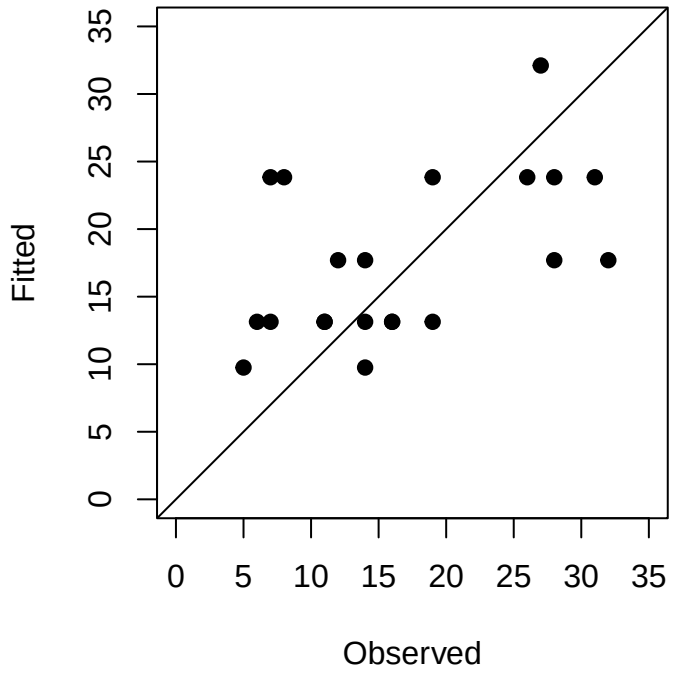
```
glm_nb.inv_adu.resid <- residuals(glm_nb.inv_adu, "deviance")

par(pty = "s", mfrow = c(1,2))

hist(glm_nb.inv_adu.resid,
     xlab = "Residuals",
     main = "")

plot(glm_nb.inv_adu.resid ~ glm_nb.inv_adu$fitted.values,
     ylim = c(-3,3),
     ylab = "Residuals",
     xlab = "Fitted",
     pch = 19
     )
abline(h = 0,
       col = "grey")
```



The residuals look as expected. Some linear patterning due to the nature of count data, but overall centered on zero and spread fairly evenly across the range of fitted values.

We can also examine how well the model performs by examining the fitted by observed values. If this were a perfect fit, the points should all vary along a 45 degree line.

```
par(pty = "s")

plot(glm_nb.inv_adu$fitted.values ~ Shott[-24,]$active,
     xlab = "Observed",
     ylab = "Fitted",
     pch = 19,
     xlim = c(0,35),
     ylim = c(0,35)
     )

abline(a = 0, b = 1) #45 degree line
```




```
#note: last case does (#24) not have co-resident estimate so this case is dropped
#automatically in the model and needs to be manually dropped here for plotting.
#This is why the code includes "Shott.inv[-24,]".
```

3.3.2 Results

First check goodness of fit.

3.3.2.1 Goodness of fit

Compare to a null model:

```
glm_nb.inv_null <- glm.nb(active ~ 1,
                          dat = Shott
                          )

anova(glm_nb.inv_null,
      glm_nb.inv_adu,
      test = "Chisq" #X^2 for negative binomial see ?MASS::anova.negbin
      )
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: active
##    Model    theta Resid. df    2 x log-lik.   Test    df LR stat.      Pr(Chi)
## 1      1 3.911032        23       -174.4211
## 2 adults 7.351390        21       -156.9579 1 vs 2     2 17.46318 0.0001614055
```

A model that includes the number of adults significantly improves the model fit.

Calculate the proportion of deviance explained:

```
with(glm_nb.inv_adu, (null.deviance-deviance)/null.deviance)
```

```
## [1] 0.373723
```

The number of co-residing adults accounts for about 37% of the deviance in ceramic inventory.

3.3.2.2 Coefficients

Now that we know the model is doing a reasonable job, we can examine what it tells us about the relationship between the number of co-residing adults and active pottery per household. To do this we pass the model object to the `summary` function:

```
summary(glm_nb.inv_adu)
```

```
##
## Call:
## glm.nb(formula = active ~ adults, data = Shott, init.theta = 7.351389966,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.97980    0.26147   7.572 3.68e-14 ***
## adults       0.29786    0.07961   3.741 0.000183 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(7.3514) family taken to be 1)
##
##     Null deviance: 37.497  on 22  degrees of freedom
## Residual deviance: 23.483  on 21  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 162.96
##
## Number of Fisher Scoring iterations: 1
```

```
##
##
##               Theta:  7.35
##           Std. Err.:  3.12
##
##  2 x log-likelihood:  -156.958
```

This output looks the same as for the Poisson GLM above, but note that it reports the dispersion parameter/Theta value, which is estimated by the model.

Extract the coefficient and take the exponent:

```
exp(glm_nb.inv_adu$coefficients[2])
```

```
##   adults
## 1.346978
```

With each additional adult, we expect there to be 35% more ceramic vessels in the house.

3.3.2.3 Prediction

How many pots would we expect to find in a household of 10 adults?

```
predict(glm_nb.inv_adu,                    #the model object
        newdata = data.frame(adults = 10), #for 10 adults
        type = "response"                  #for simplicity, predict on the response scale
        )
```

```
##        1
## 142.3713
```

The model predicts 142 active vessels for a house of ten adults, but note: this does predict outside the range of observation, and so should be treated with caution.

Predict across a range of x:

```
#create a vector across the range of x values
adults_seq <- seq(from = min(Shott$adults, na.rm = TRUE) - 1, #from the min -1 -NAs
                  to = max(Shott$adults, na.rm = TRUE) + 1,   #to the max +1 -NAs
                  by = 0.5 #for each 0.5 value
                  )

#predict ceramic counts a cross the range of x
pottery_pred <- predict(glm_nb.inv_adu, #the model object
                        newdata = data.frame(adults = adults_seq), #*
                        se = TRUE,      #include the standard error of the prediction
                        type = "link"   #be sure to predict on the link scale
                        )

#* note, the new data must be a data frame with columns corresponding to each variable
#* in the model, each of which must have the exact same name as the model variable.
```

Check out the resulting object. For each value in the prediction sequence (`adults_seq`), the object has a predicted fit and standard error around the fit. Note these are on the link scale, so they are the logged value:

```
pottery_pred
```

```
## $fit
##        1        2        3        4        5        6        7        8
## 1.979802 2.128734 2.277666 2.426597 2.575529 2.724461 2.873393 3.022325
##        9       10       11       12       13
```

```
## 3.171257 3.320188 3.469120 3.618052 3.766984
##
## $se.fit
##          1          2          3          4          5          6          7
## 0.26146708 0.22464346 0.18903261 0.15547013 0.12560886 0.10272878 0.09218357
##          8          9         10         11         12         13
## 0.09803780 0.11787282 0.14610129 0.17879083 0.21390589 0.25042821
##
## $residual.scale
## [1] 1
```

We can use these to make a plot of the model fit, but first, we need one more thing. We have to save the inverse link function from the model object (note, this will be the same for all negative binomial GLMs). We use this to convert the predicted value, which is logged, back to the response scale. For a more detailed explanation and tutorial on this, see Simpson (2018a).

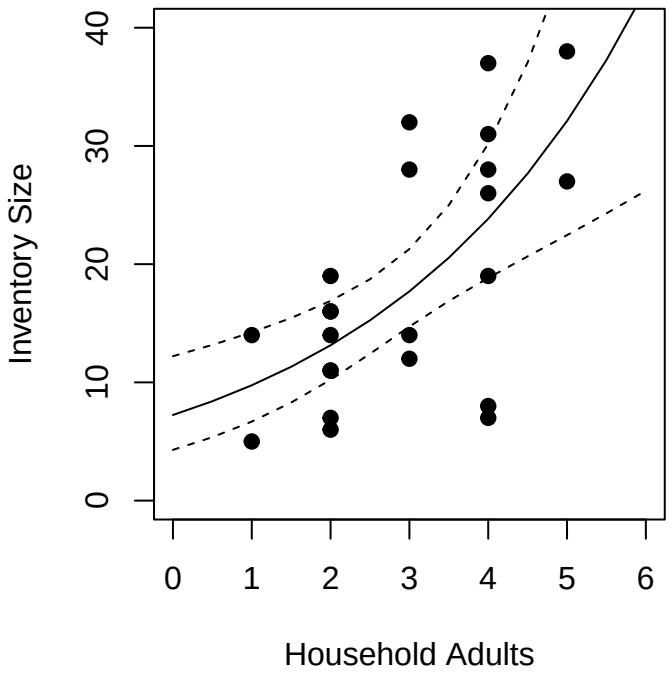
```
family(glm_nb.inv_adu)
```

```
##
## Family: Negative Binomial(7.3514)
## Link function: log
```

```
#this function tells you the model family and link,
#it also includes an inverse link function

inv_link.glm_nb <- family(glm_nb.inv_adu)$linkinv #inverse link of the model
```

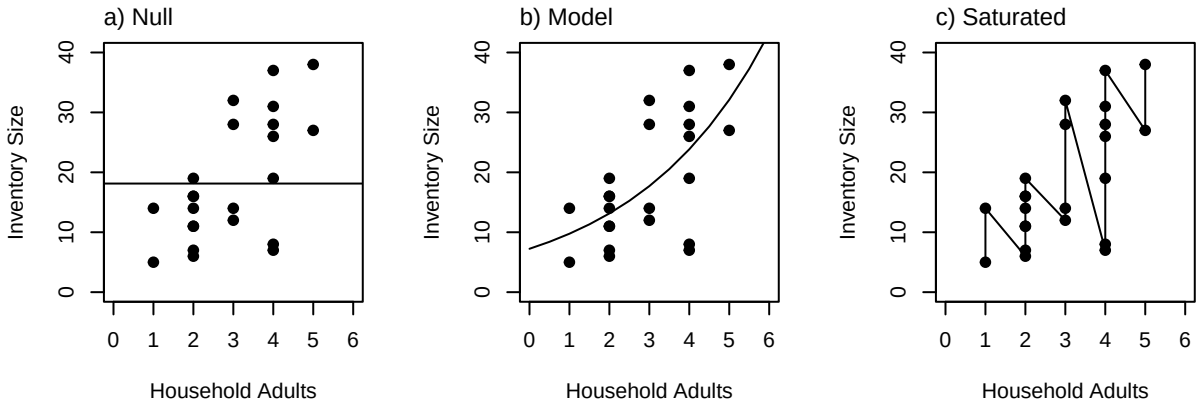
Plot the response:

```
par(pty = "s")

with(Shott,
     plot(active ~ adults,
          pch = 19,
          xlim = c(0, 6),
          ylim = c(0, 40),
          xlab = "Household Adults",
          ylab = "Inventory Size",
          type = "p"
          )
     )

lines(inv_link.glm_nb(pottery_pred$fit) ~ adults_seq)

lines(inv_link.glm_nb(pottery_pred$fit + 2 * pottery_pred$se.fit) ~ adults_seq, lty = 2)

lines(inv_link.glm_nb(pottery_pred$fit - 2 * pottery_pred$se.fit) ~ adults_seq, lty = 2)
```



Graphical representation of the predicted model fit compared to the null and saturated models:

```
#pdf("Figure5.pdf", height = 3, width = 7.5) #save pdf figure

par(pty = "s", mfrow = c(1,3))

with(Shott,
     plot(active ~ adults,
          pch = 19,
          xlim = c(0, 6),
          ylim = c(0, 40),
          xlab = "Household Adults",
          ylab = "Inventory Size",
          type = "p"
          )
     )
abline(h = mean(Shott$active)) #only the mean
mtext("a) Null", side = 3, line = 0.5, adj = 0, cex = 0.75)

with(Shott,
     plot(active ~ adults,
          pch = 19,
          xlim = c(0, 6),
          ylim = c(0, 40),
          xlab = "Household Adults",
          ylab = "Inventory Size",
          type = "p"
          )
     )
lines(inv_link.glm_nb(pottery_pred$fit) ~ adults_seq)
mtext("b) Model", side = 3, line = 0.5, adj = 0, cex = 0.75)

with(Shott[with(Shott, order(adults, active)),], #order by x axis values
     plot(active ~ adults,
          pch = 19,
          xlim = c(0, 6),
          ylim = c(0, 40),
          xlab = "Household Adults",
          ylab = "Inventory Size",
          type = "o"                        #over plot - a fit for every point
          )
     )
mtext("c) Saturated", side = 3, line = 0.5, adj = 0, cex = 0.75)
```


a) Null
b) Model
c) Saturated
Inventory Size
Household Adults
0
10
20
30
40
0
1
2
3
4
5
6

4 Count regression with variable sampling windows: Neolithic cattle

Question: Do counts of auroch and cattle (*Bos* spp.) bones increase during the Neolithic compared to the Mesolithic?

To address this question, we draw on the EUROEVOL data set that examines the Cultural Evolution of Neolithic Europe (Manning 2016; Manning et al. 2015; Timpson et al. 2016). These data are from Manning and colleague's (2015) "FaunalBones" – two files providing the data (EUROEVOL09-07-201516-34_FaunalBones.csv) and field type definitions (FaunalBones_fields.csv) for each bone assigned by PhaseCode and species with associated measurements.

To answer our question, we are interested in two species: aurochs (*Bos primigenius*) and cattle (*Bos taurus*). However, distinguishing between the two based on their bones alone is problematic given that it is done by size and there is overlap between them (Wright 2016). Here we include all *Bos* counts for this analysis given that we are interest to determine if there is an increase reliance on *Bos* with the transition to agriculture. Here we model variation over categorical time periods.

```
Manning
```

```
##      Phase Period bos NISP
## 1    AARTS     LN  24   63
## 2      ABL     LN 753 3249
## 3      ADP     EN  12   26
## 4     AGER     LM   0   43
## 5      AGR     LN   1    3
## 6       AH     LN   0   15
## 7      ALS     MN  10   21
## 8       AP     EN  11   23
## 9      AUE     LN  27   58
## 10    AULS     LN 187  910
## 11     AUW     EN  11   29
## 12    BAL1     EN 138  312
## 13    BAL2     MN  96  206
## 14     BR1     EN  16   40
## 15    BRNK     MN   6   13
## 16     BUC    EBA 147  482
## 17     BUH    EBA  51  118
## 18    CALB     LN  99  223
## 19     CCZ    EBA 250  602
## 20      CH     LN  21   42
## 21    CHAT     MN  20   40
## 22    CHL2     LN  75  273
## 23    CHL4     LN  25  113
## 24    CHL6     LN 117  395
## 25    CHL8     LN  91  468
## 26     CLM     MN  16   39
## 27     CML     MN  14   36
## 28    COUR     MN   9   18
## 29    CP-C     LN   6   13
## 30   CRN12     MN   6   13
## 31     CSO     EN  30   75
## 32     CYA     EN  18   36
## 33     DBK   LMEN   7   31
## 34      DH     LN  11   36
```

```
## 35     DM1     EN  12   31
## 36     DRC  LNEBA   2    5
## 37     DRZ     MN  13   33
## 38     DUL     LN  11   30
## 39      DW     LN   0 1484
## 40      E1     EN 146  341
## 41    ERGO     LN  64  226
## 42    FALK     MN  63  163
## 43      FF    EBA   6   12
## 44      FN     MN   4   17
## 45    FTF2   LMEN   3    6
## 46    FTF4     LN   5   12
## 47    FTF5    EBA   7   14
## 48    GDEC     LN 311  622
## 49     GDF     LN  25  108
## 50     GGB     MN   0   13
## 51    GGBZ    EBA   9   39
## 52     GGC     MN   0   20
## 53   GGCAM     LN  19   67
## 54     GNA     EN  28   82
## 55    GNA2     MN   4   18
## 56    GRAV     MN  40   81
## 57   GRYS1     LN  30   70
## 58      GW     EN  22   48
## 59     GW2     MN  23   47
## 60     GW3     MN   8   16
## 61     GW4     MN  19   39
## 62     HAL     MN   9   20
## 63     HCE     EN  12   24
## 64   HENE1     LN 188  587
## 65      HH     EN  13   31
## 66   HH-HS     EN   2    5
## 67   HH-SS     EN  10   41
## 68      HK     EN  12   39
## 69     HK2     MN 908 2608
## 70     HLB     EN   2    4
## 71      K1     EN   0    1
## 72      K2     LN   2    4
## 73     KBR     LN   1    3
## 74    KRP2     MN  26   52
## 75    KWII     LN   1    3
## 76     LAG     EN  10   20
## 77    LAG5     MN  58  129
## 78      LD     MN  36   82
## 79     LGX     LN   6   12
## 80    LROB     MN 105  241
## 81   LSMII     EN  11   44
## 82  LSMIII     EN  25   56
## 83    MAIX     LN 494 1154
## 84    MAIZ     MN  29   88
## 85     MC1     EN   4   24
## 86   MESP2     LN  16   52
## 87     MIC     MN 164  375
## 88     ML1     LM   0    8
```

```
## 89   ML234    EBA   9   26
## 90     MLB     EN   4   12
## 91     MRO     MN  23   60
## 92    MSCH     LN   2    4
## 93    MSEE     LN  31   73
## 94      MW     MN  99  214
## 95     NBO     MN  32   82
## 96    NIED     LN   3    6
## 97    NOY1     MN 101  255
## 98     ODE     LN  18   53
## 99      P1     EN   7   14
## 100     P2     LN   2    4
## 101   PBF1     LN   6   14
## 102    PFH     EN   1    6
## 103  PLARB     MN  21   56
## 104    PLF     EN   7   14
## 105   POGJ     MN  13   27
## 106   PORH     EN   0   41
## 107   RADW    EBA   3   30
## 108    RAT     EN  29   70
## 109     RB     MN  31  104
## 110   REBA     MN  30   74
## 111    RES     LN  77  238
## 112     RG     EN   3    9
## 113    RI2  LNEBA  13   36
## 114   ROTF     EN   3   28
## 115   SCB3    EBA 147  362
## 116   SCBG     LN 371  882
## 117    SCE     EN  37   74
## 118    SCH     MN  13   38
## 119   SCHI     MN  60  165
## 120   SEEA     LN  61  138
## 121    SHN    EBA   3    9
## 122    SPH     LM   2    9
## 123    SR8     MN  52  114
## 124    SRF     LN   3   12
## 125    STF    EBA 191  465
## 126  STRL1     EN  75  182
## 127  STRL2     MN   4    8
## 128    STW     MN  21   42
## 129   TEM1     LN  23   46
## 130   TEM2    EBA  36   72
## 131   THIE     MN  12   24
## 132   TRM2     LN   9   18
## 133    VAI     EN 925 2447
## 134   VTOL     MN 193  463
## 135  WEG12     MN  52  104
## 136    WH1     EN 145  307
## 137    WH3     EN  46  100
## 138    WHX     EN   1    2
## 139   WHX2     LN  12   25
## 140    WIT     LN   1    3
## 141    WKP     LN   6   32
## 142     WM     LN   4   22
```

```
## 143    ZWJ     EN   3    6
```

- Phase = "The use of the term 'Phase' in these datasets refers to data aggregated at the level of the cultural unit... the most common level of aggregation in the faunal and archaeobotanical reports, and therefore offered maximum comparative potential between the different datasets" (Timpson et al. 2016).
- Period = time period
  - "LM" = late Mesolithic
  - "LMEN" = late Mesolithic/early Neolithic
  - "EN" = early Neolithic
  - "MN" = middle Neolithic
  - "LN" = late Neolithic
  - "LNEBA" = late Neolithic/early Bronze Age
  - "EBA" = early Bronze Age
- bos = counts of identifiable specimens (NISP) to *Bos primigenius* or *Bos taurus*.
- NISP = total number of all identifiable specimens (NISp)

First we need to prepare the data to answer the question: how do counts of *Bos* bones vary over time? We need to order the factors for each time period.

```
#set as factor
Manning$Period <- factor(Manning$Period, levels = c("LM",
                                                    "LMEN",
                                                    "EN",
                                                    "MN",
                                                    "LN",
                                                    "LNEBA",
                                                    "EBA")
                         )
```

The unit of analysis is each "phase", which we can examine by period to see how counts of *Bos* bones vary. However, by modeling time period as a factor, the result will examine variation relative to the reference class, in this case, how each subsequent time period compares to the late Mesolithic. This is appropriate given our question, however, the transitional time period "LMEN" complicates this hypothesis test. As such, we remove it.

```
Manning.sub <- subset(Manning, Period != "LMEN") #all except transitional M-N

Manning.sub$Period <- droplevels(Manning.sub$Period) #drop unused factor level
```

How many levels do we have?

```
levels(Manning.sub$Period)
```

```
## [1] "LM"    "EN"    "MN"    "LN"    "LNEBA" "EBA"
```

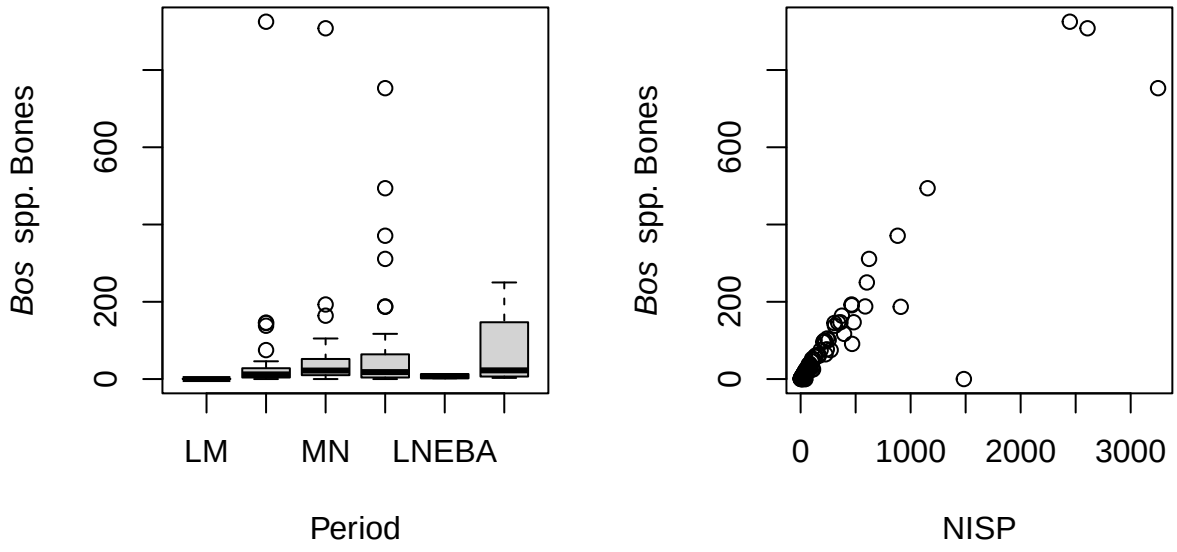
Note: the first level value is the reference class to which all others will be compared. For us, this is perfect as we want to compare agricultural periods with the last non-agricultural period. However, you can re-order factor levels however you need to answer the question. See `?levels`.

4.1 Exploratory data analysis

Plot the distribution of *Bos* bones for each time period using box plots and see how the number of *Bos* bones varies with the total number of identifiable specimens (NISP).

```
par(pty = "s", mfrow = c(1,2))

with(Manning.sub,
     plot(bos ~ Period/NISP, #plot Bos bones by period and NISP
```

```
          ylab = expression(italic("Bos ")~"spp. Bones")
          )
    )
```



Clearly the number of cattle bones increases as a function of the total number of bones per phase, so we need to include an offset for NISP.

4.2 Fit a Poisson GLM with offsets

```
glm_Pois.bos_period <- glm(bos ~ Period,
                           offset = log(NISP), #include a log offset
                           family = poisson(link = "log"),
                           data = Manning.sub
                           )
```

4.2.1 Model diagnostics

```
overdisp_fun(glm_Pois.bos_period)
```

4.2.1.1 Overdispersion

```
##         chisq         ratio           rdf             p
##  9.973755e+02  7.387967e+00  1.350000e+02 1.746483e-131
```

Significant overdispersion detected.

4.3 Refit with a negative binomial GLM

Note, the syntax for including an offset differs with `MASS::glm.nb` so that we include it as an additive term specified as an offset.

```
glm_negb.bos_period <- glm.nb(bos ~ Period +
                              offset(log(NISP)),
                              data = Manning.sub
                              )
```

4.3.1 Model diagnostics

```
overdisp_fun(glm_negb.bos_period)
```

4.3.1.1 Overdispersion

```
##       chisq       ratio         rdf           p
##  71.1213609   0.5268249 135.0000000   0.9999988
```

The negative binomial model accounts for overdispersion.

```
zeroinfl_fun(model = glm_negb.bos_period, response = Manning.sub$bos)
```

4.3.1.2 Zero-inflation

```
##    obs_0    mod_0    ratio
## 8.000000 3.000000 2.666667
```

There are 8 observed zeros and 3 predicted zeros, which suggests some zero-inflation, but nothing too concerning. Just make sure to pay attention when interpreting predictions of low counts as these could be zero.

4.3.2 Results

Now assess goodness of fit and model coefficients.

4.3.2.1 Goodness of fit First compare to a null model:

```
glm_negb.bos_null <- glm.nb(bos ~ 1 +
                            offset(log(NISP)),
                            data = Manning.sub
                            )

anova(glm_negb.bos_null,
      glm_negb.bos_period,
      test = "Chisq" #X^2 for negative binomial see ?MASS::anova.negbin
      )
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: bos
##                         Model    theta Resid. df    2 x log-lik.   Test    df
## 1      1 + offset(log(NISP)) 6.087784       140      -1009.5073
## 2 Period + offset(log(NISP)) 7.195535       135       -990.7728 1 vs 2     5
##   LR stat.     Pr(Chi)
## 1
## 2 18.73456 0.002153641
```

This model is a significant improvement on the null.

Likelihood r_l^2:

```
with(glm_negb.bos_period, (null.deviance-deviance)/null.deviance)
```

```
## [1] 0.1122308
```

Period accounts for about 11% of the deviance in *Bos* bone counts.

```
summary(glm_negb.bos_period)
```

4.3.2.2 Coefficients

```
##
## Call:
## glm.nb(formula = bos ~ Period + offset(log(NISP)), data = Manning.sub,
##     init.theta = 7.19553472, link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.2933     0.7233  -4.553 5.29e-06 ***
## PeriodEN      2.3420     0.7277   3.219 0.001288 **
## PeriodMN      2.4023     0.7265   3.307 0.000944 ***
## PeriodLN      2.2028     0.7267   3.031 0.002437 **
## PeriodLNEBA   2.3003     0.8282   2.777 0.005479 **
## PeriodEBA     2.3094     0.7346   3.144 0.001668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(7.1955) family taken to be 1)
##
##     Null deviance: 175.61  on 140  degrees of freedom
## Residual deviance: 155.90  on 135  degrees of freedom
## AIC: 1004.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  7.20
##           Std. Err.:  1.24
##
##  2 x log-likelihood:  -990.773
```

Bos bone counts are significantly higher during all Neolithic and Bronze age periods compared to the Mesolithic.

To get the rate ratio for any period, just take the exponent of that coefficient. For example, for the Early Neolithic (EN) period, the ratio is `exp(2.34)`, or ≈ 10.4 times the number of *Bos* spp. bones when compared to the late Mesolithic.

Note: when the predictor variable is a factor, the first factor is the "reference class" to which all others are compared, as such, the model does not return coefficients for the first class. Here this is useful as it allows us to assess how *Bos* bone counts vary across Neolithic (i.e., farmer) periods relative to the Mesolithic (i.e., hunter-gatherer) period.

4.3.2.3 Prediction Plot the predicted responses per time period holding the sample size (NISP) at the median value (n = 40):

```
bos_pred <- predict(glm_negb.bos_period,
                    newdata = data.frame(
                      Period = levels(Manning.sub$Period), #for each level of period
                      NISP = median(Manning.sub$NISP) #for the median NISP
                                    ),
                    type = "link",
                    se = TRUE
                    )
```

As we predicted on the link scale, this will show the predicted log counts of bones:

```
bos_pred$fit
```

```
##         1         2         3         4         5         6
## 0.3955356 2.7375444 2.7978110 2.5983163 2.6957899 2.7049125
```

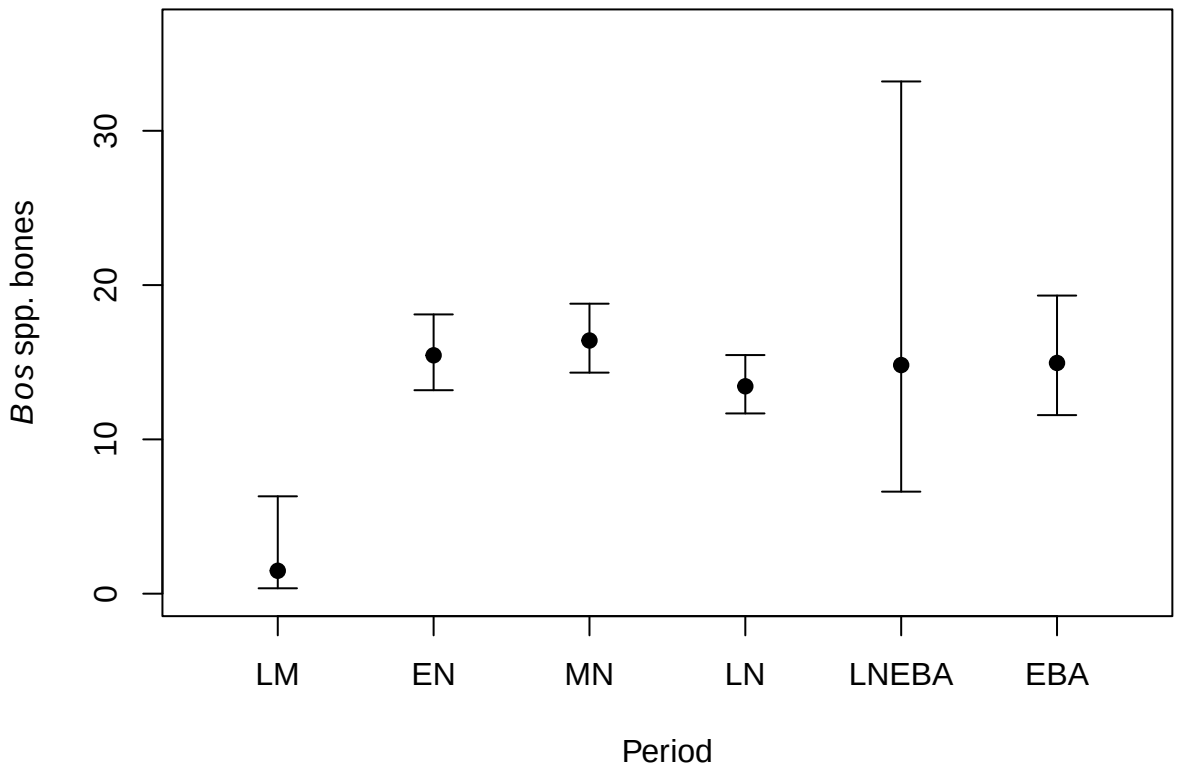
As above, take the exponent to see the predicted counts:

```
exp(bos_pred$fit)
```

```
##         1         2         3         4         5         6
##  1.485179 15.449002 16.408690 13.441088 14.817218 14.953009
```

Show the predicted number of *Bos* bones for each time period with confidence intervals. This can be done as a box-and-whisker plot using the `points` and `arrows` functions.

```
plot(NA,
     xlab = "Period",
     ylab = expression(italic(Bos) ~ "spp. bones"),
     xlim = c(0.5,6.5), #number of categories with buffer
     ylim = c(0, max(inv_link.glm_nb(bos_pred$fit))+20), #max value plus buffer
     xaxt = "n" #turn off the axis to plot periods later
     )

axis(side = 1, at = 1:6, levels(Manning.sub$Period)) #add time period labels

points(1:6, inv_link.glm_nb(bos_pred$fit), pch = 19)

#use arrows to make the whiskers as 95% confidence intervals
arrows(x0 = 1:6,
       x1 = 1:6,
       y0 = inv_link.glm_nb(bos_pred$fit + (2*bos_pred$se.fit)),
       y1 = inv_link.glm_nb(bos_pred$fit - (2*bos_pred$se.fit)),
       angle = 90, #90 degree angle for flat arrowhead
       code = 3, #heads on y0 (1), y1 (2), or or both (3)
       length = 0.1 #length of arrowhead in in inches
       )
```



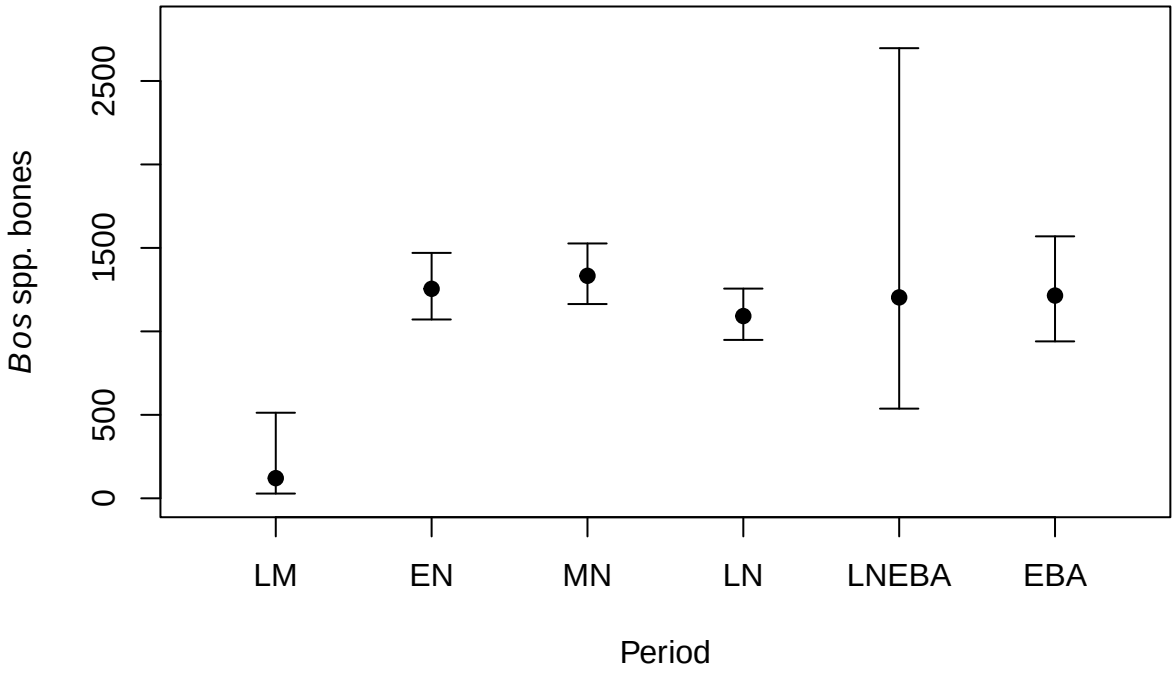
This shows the predicted number of *Bos* bones for each time period from the Late Mesolithic to Early Bronze Age with 95% confidence intervals as if each sample size (NISP) was the median (n = 40).

For an example of how predictions would vary with different sample sizes, let's plot this again but hold NISP at the maximum value (n = 3249):

```
bos_pred <- predict(glm_negb.bos_period,
                    newdata = data.frame(
                      Period = levels(Manning.sub$Period), #for each level of period
                      NISP = max(Manning.sub$NISP) #for the median NISP
                                        ),
                    type = "link",
                    se = TRUE
                    )
```

```
plot(NA,
     xlab = "Period",
     ylab = expression(italic(Bos) ~ "spp. bones"),
     xlim = c(0.5,6.5), #number of categories with buffer
     ylim = c(0, max(inv_link.glm_nb(bos_pred$fit))+1500), #max value plus buffer
     xaxt = "n" #turn off the axis to plot periods later
     )

axis(side = 1, at = 1:6, levels(Manning.sub$Period)) #add time period labels
```

```r
points(1:6, inv_link.glm_nb(bos_pred$fit), pch = 19)

#use arrows to make the whiskers as 95% confidence intervals
arrows(x0 = 1:6,
       x1 = 1:6,
       y0 = inv_link.glm_nb(bos_pred$fit + (2*bos_pred$se.fit)),
       y1 = inv_link.glm_nb(bos_pred$fit - (2*bos_pred$se.fit)),
       angle = 90, #90 degree angle for flat arrowhead
       code = 3, #heads on y0 (1), y1 (2), or or both (3)
       length = 0.1 #length of arrowhead in in inches
       )
```



Note that their relative position remains the same, but the absolute predicted count is inflated.

For another example, consider repeating the above using the Snodgrass data (Price and Griffin 1979; Cogswell et al. 2001) in the `archdata` package (Carlson 2017) to see how "prestige" items such as earplugs and effigies vary by excavation section while controlling for household size.

Multiple Regression

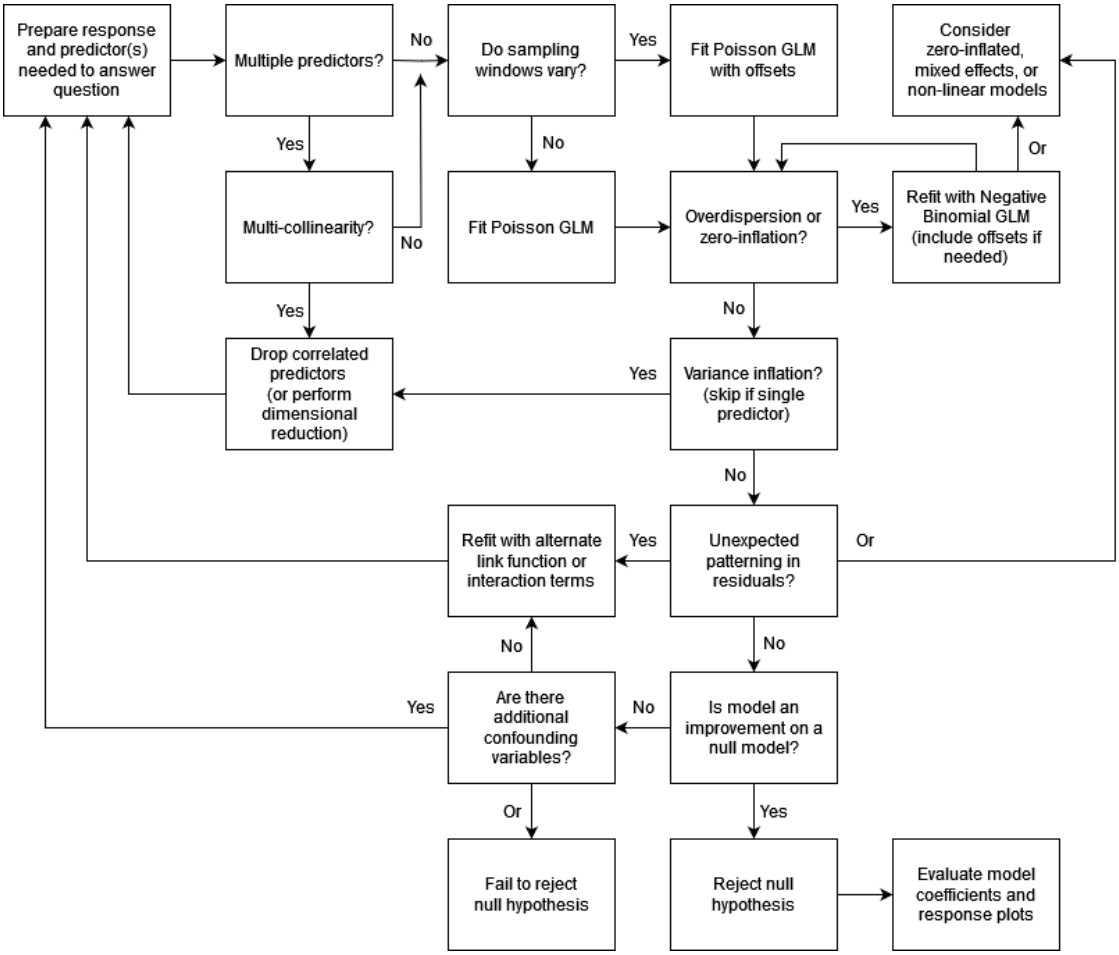


Figure 2: Flowchart outlining recommended procedures for fitting and evaluating count models with multiple predictors. For other examples with mixed effects models, see Bolker et al. (2009).

5 Count regression with multiple predictors: Texas point types

Question: Does environmental stress and risk of resource shortfall influence technological intensification and innovation?

In order to assess the "risk hypothesis" for technological complexity – that populations under risk of resource stress or shortfall will invest more in specialized tools leading to greater technological types (see Torrence 1983) – Buchannan et al. (2016: Table 1) compile data on the number of projectile point types by time period in Texas relative to proxies of environmental "risk".

```
Buchannan
```

```
##            Time.period Number.of.point.types Start   End Duration
## 1    Early Paleoindian                     2 13060 11910     1150
## 2     Late Paleoindian                    19 11910  8854     3056
## 3        Early Archaic                    27  8854  5142     3712
## 4       Middle Archaic                    30  5142  3185     1957
## 5         Late Archaic                    32  3185  1523     1662
## 6 Transitional Archaic                    16  1523  1174      349
## 7     Late Prehistoric                    40  1174   404      770
##   Regional_risk_C13 Global_risk_O18
## 1            -4.035          -3.486
## 2            -7.985          -1.540
## 3            -6.360           0.347
## 4            -5.960           0.116
## 5            -8.190          -0.043
## 6            -6.770          -0.184
## 7            -3.730          -0.163
```

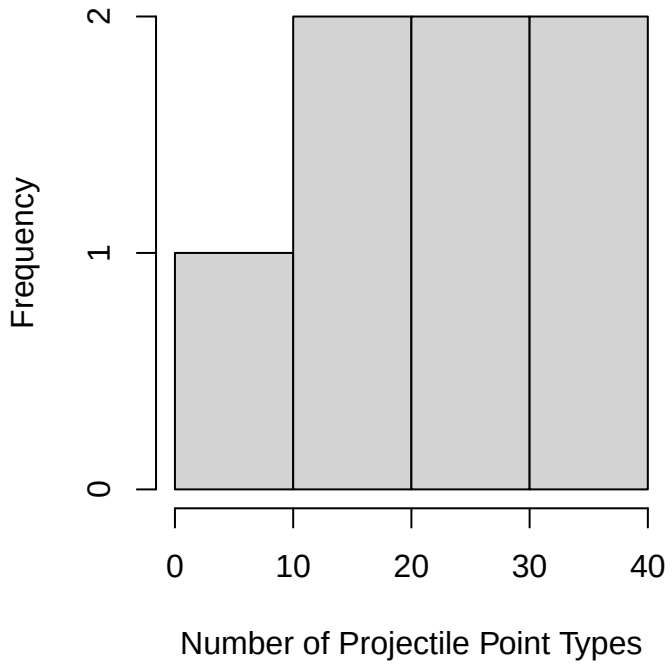
The data include:

- Time.period = Categorical name of each time period: "Early Paleoindian", "Late Paleoindian", "Early Archaic", "Middle Archaic", "Late Archaic", "Transitional Archaic", "Late Prehistoric".
- Number.of.point.types = The number of projectile point types in Texas per period.
- Start = Beginning year of the time period (calibrated years BP).
- End = Ending year of the time period (calibrated years BP).
- Duration = Total duration of the time period.
- Regional_risk_C13 = Stable carbon-13 ratio from a sediment cores in Denton County (north-central Texas) as a proxy for precipitation. Higher values equate to drier conditions.
- Global_risk_O18 = Average ratio of oxygen-18 to oxygen-16 stable isotopes in the atmosphere derived from the Vostok (Antarctica) ice core as a measure of global temperature. Higher values equate to higher global temperatures.

They note that, "[f]or the purposes of testing the risk hypothesis, we assumed that decreasing precipitation and increasing temperature coincided with increasing risk because drier and hotter environments in Texas, such as large areas of southwestern Texas, have less biomass available, which translates into less subsistence resources available for human foragers." One question they ask is whether regional precipitation or global temperature has a stronger influence on point types.

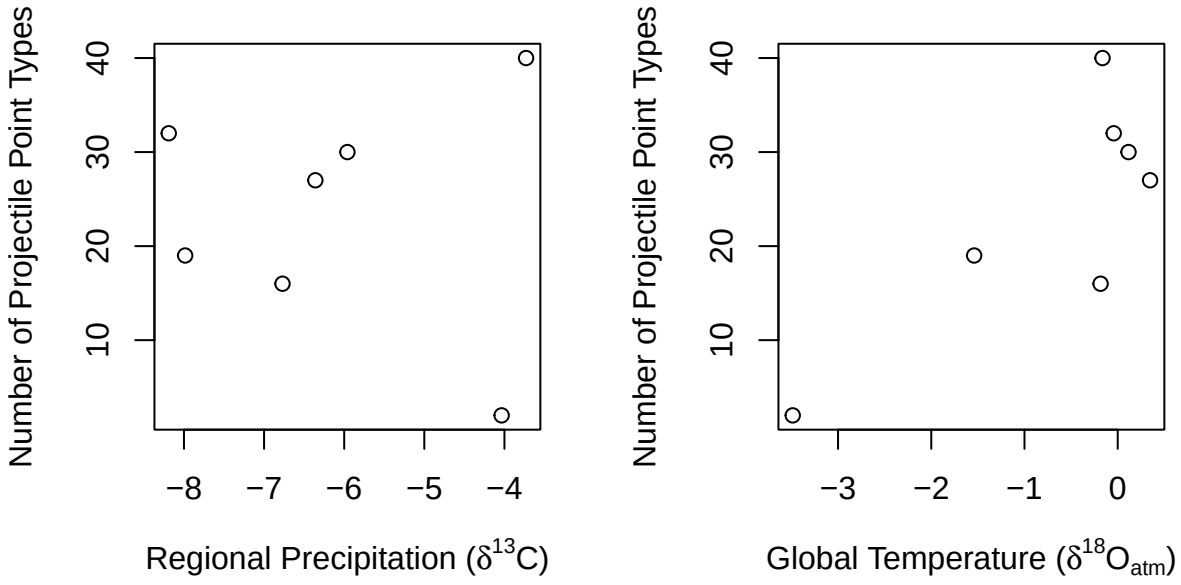
5.1 Exploratory data analysis

```
par(pty = "s")

with(Buchannan,
     hist(Number.of.point.types,
          xlab = "Number of Projectile Point Types",
          main = NA)
     )
```



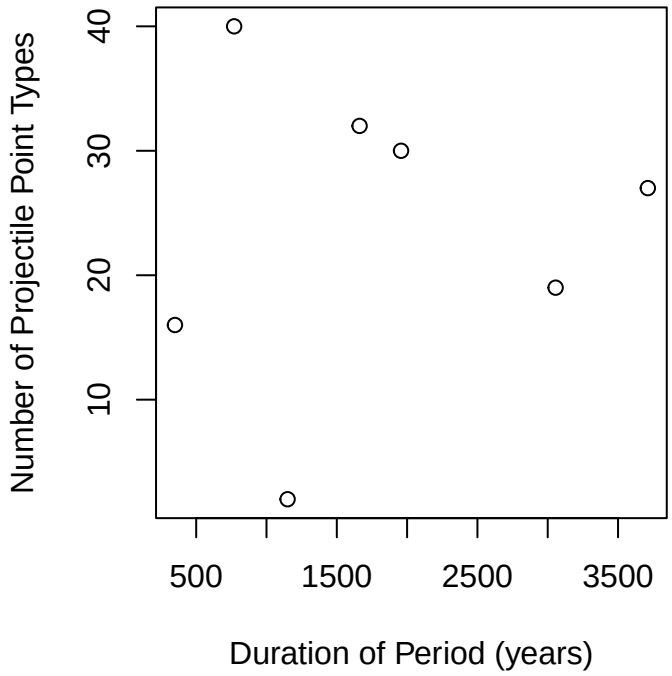
Create bivariate plots of the response and predictors (note, the `expression` and `paste` functions are used to include Greek symbols, subscript, and superscript type).

```
par(pty = "s",  mfrow = c(1,2))

with(Buchannan,
     plot(Number.of.point.types ~ Regional_risk_C13,
          ylab = "Number of Projectile Point Types",
          xlab = expression(paste("Regional Precipitation (", delta ^{13},"C)")),
          )
     )

with(Buchannan,
     plot(Number.of.point.types ~ Global_risk_O18,
          ylab = "Number of Projectile Point Types",
          xlab = expression(paste("Global Temperature (", delta ^{18},"O",''[atm],")")),
          )
     )
```



As the duration of time periods varies, our model will need to account for this:

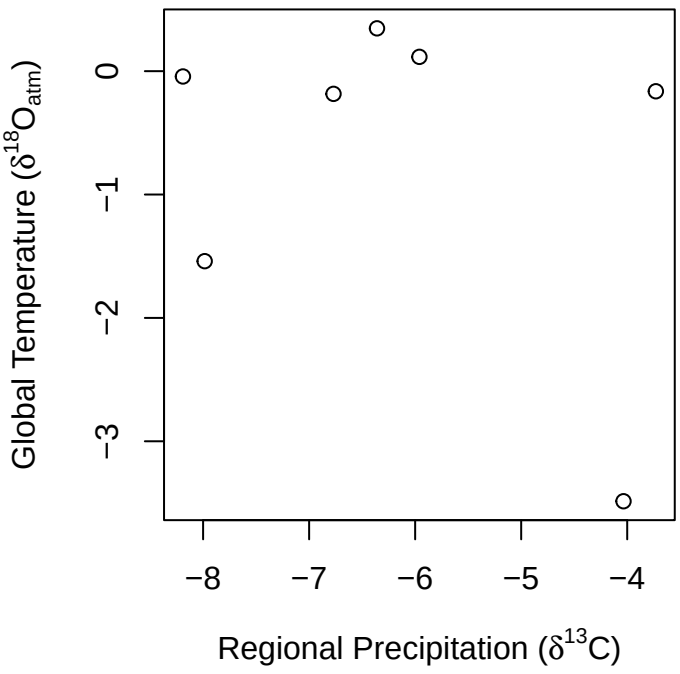
```
par(pty = "s")

with(Buchannan,
     plot(Number.of.point.types ~ Duration,
          ylab = "Number of Projectile Point Types",
          xlab = "Duration of Period (years)")
     )
```



5.1.1 Multicollinearity

Importantly, as we want to assess multivariate relationships, we need to check for multicollinearity. This can be done with a bi-variate plot and a simple correlation test.

```
par(pty = "s")

with(Buchannan,
     plot(Global_risk_O18 ~ Regional_risk_C13,
          ylab = expression(paste("Global Temperature (", delta ^{18},"O",''[atm],")")),
          xlab = expression(paste("Regional Precipitation (", delta ^{13},"C)"))
          )
     )
```




```
#correlation test
with(Buchannan,
     cor.test(Global_risk_O18, Regional_risk_C13)
     )
```

```
##
##  Pearson's product-moment correlation
##
## data:  Global_risk_O18 and Regional_risk_C13
## t = -0.79018, df = 5, p-value = 0.4652
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8683649  0.5605065
## sample estimates:
##        cor
## -0.3331884
```

While the two predictors have a correlation coefficient of -0.33, that should not be strong enough to inflate model coefficients – though we will double-check below. We should feel comfortable using both predictors.

5.2 Fit a Poisson GLM

```
glm_pois.point_multi <- glm(Number.of.point.types ~ Global_risk_018 +
                                                    Regional_risk_C13,
                            offset = log(Duration),
                            data = Buchannan,
                            family = poisson
                            )
```

5.2.1 Model diagnostics

```
overdisp_fun(glm_pois.point_multi)
```

5.2.1.1 Overdispersion

```
##        chisq        ratio          rdf            p
## 9.023024e+01 2.255756e+01 4.000000e+00 1.176511e-18
```

Meaningful overdispersion. Refit with negative binomial GLM.

5.3 Fit negative binomial GLM

```
glm_nb.point_multi <- glm.nb(Number.of.point.types ~ Global_risk_018 +
                                                     Regional_risk_C13 +
                             offset(log(Duration)),
                             data = Buchannan
                             )
```

5.3.1 Model diagnostics

```
overdisp_fun(glm_nb.point_multi)
```

5.3.1.1 Overdispersion

```
##     chisq     ratio       rdf         p
## 5.6935509 1.4233877 4.0000000 0.2232328
```

The negative binomial model adequately handles overdispersion.

```
zeroinfl_fun(glm_nb.point_multi, response = Buchannan$Number.of.point.types)
```

5.3.1.2 Zero-inflation

```
## obs_0 mod_0 ratio
##     0     0   NaN
```

No zero values.

5.3.1.3 Variance inflation factor While it was not apparent *a priori*, multicollinearity may still influence the model result. As such, we need to assess the influence of multicollinearity on the model result. We can rely on the `vif` function in the `car` package (Fox and Weisberg 2019). For more, see `?car::vif`.

```
library(car)
```

```
car::vif(glm_nb.point_multi)
```

```
##   Global_risk_O18 Regional_risk_C13
##           1.04068           1.04068
```

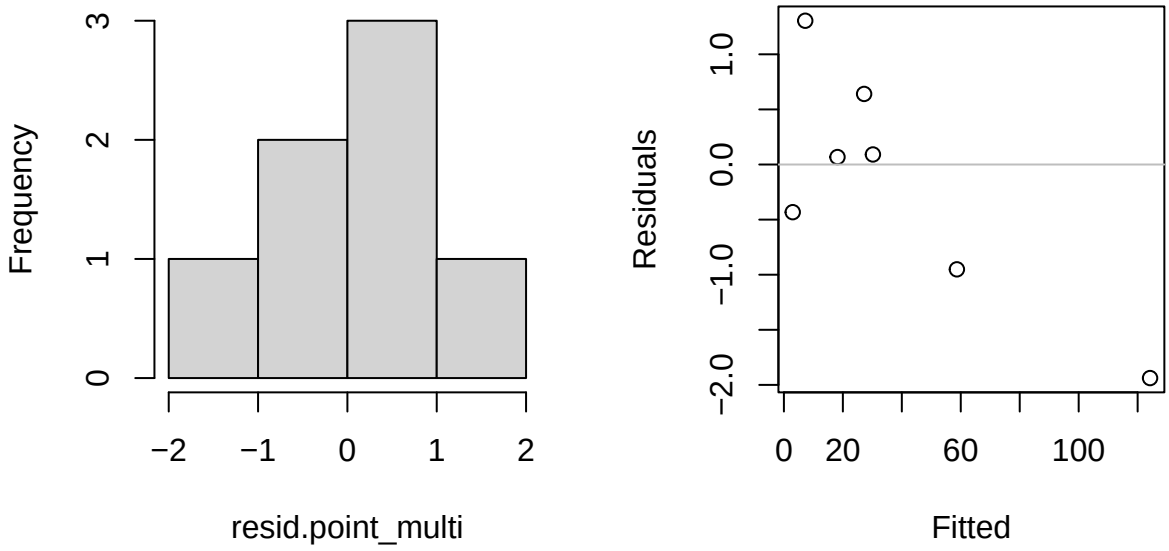
If the VIF value is near 1, then there is no concern. If the value is close to 5, then there is moderate correlation. If the value is well above 5, then the predictors are highly correlated.

5.3.1.4 Residuals

Plot the residuals.

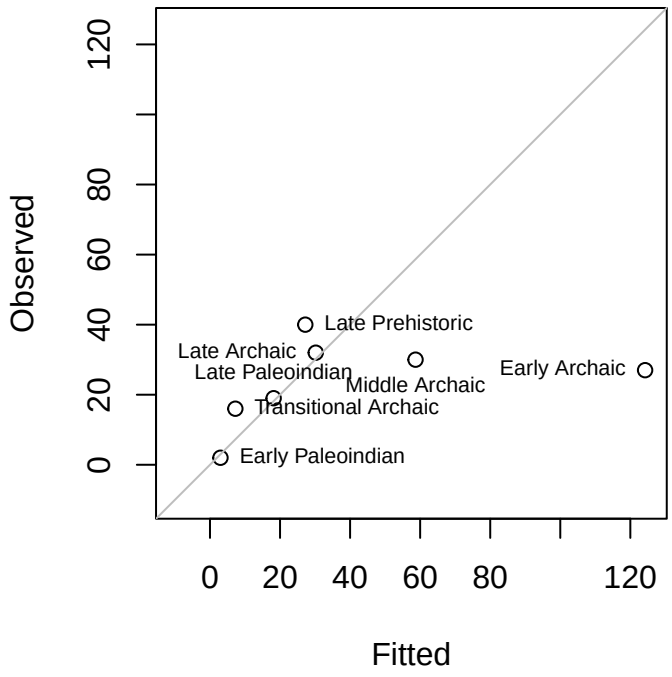
```
resid.point_multi <- residuals(glm_nb.point_multi, type = "deviance")
```

```
par(pty = "s", mfrow = c(1,2))

hist(resid.point_multi, main = NA)

plot(resid.point_multi ~ glm_nb.point_multi$fitted.values,
     xlab = "Fitted",
     ylab = "Residuals")
abline(h=0, col="gray")
```



The model does a fairly good job, but note the high outlier. We can examine this further by plotting the observed by fitted values and labeling them to see which time period is not predicted well by the model.

```
par(pty = "s")

plot(Buchannan$Number.of.point.types ~ glm_nb.point_multi$fitted.values,
     ylab = "Observed",
     xlab = "Fitted",
     xlim = c(-10,125),
     ylim = c(-10,125)
     )

abline(a=0, b = 1, col="gray")

#label by time period
text(x = glm_nb.point_multi$fitted.values,
     y = Buchannan$Number.of.point.types,
     Buchannan$Time.period,
     pos = c(4,3,2,1,2,4,4), #custom placement
     cex = 0.65 #smaller labels
     )
```



The model predicts way more point types than observed for the "Early Archaic" period. Why might this be?

Technically these data are a time series, so it would be good to assess the residuals for temporal autocorrelation. This is beyond our scope here, but see Simpson (2018b)

5.3.2 Results

Examining the results of multivariate models follow bivariate models, with a few additions.

5.3.2.1 Goodness of fit

Compare to a null model:

```
glm_nb.point_null <- glm.nb(Number.of.point.types ~ 1 +
                            offset(log(Duration)),
                            data = Buchannan
                            )


anova(glm_nb.point_null,
      glm_nb.point_multi,
      test = "Chisq"
      )
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: Number.of.point.types
##                                                                 Model    theta
## 1                                     1 + offset(log(Duration)) 1.235997
## 2 Global_risk_O18 + Regional_risk_C13 + offset(log(Duration)) 2.626179
##   Resid. df    2 x log-lik.   Test    df LR stat.    Pr(Chi)
## 1         6       -61.52922
## 2         4       -55.53585 1 vs 2     2 5.993371 0.04995236
```

A slight improvement on a null model with only the offsets.

How much deviance in projectile point type count is explained by both predictors?

```
with(glm_nb.point_multi, (null.deviance-deviance)/null.deviance)
```

```
## [1] 0.5450301
```

About 55% of the variation in the number of projectile point types is accounted for by the combination of local precipitation and global temperature.

5.3.2.2 Coefficients

Let's evaluate the model coefficients.

```
summary(glm_nb.point_multi)
```

```
##
## Call:
## glm.nb(formula = Number.of.point.types ~ Global_risk_O18 + Regional_risk_C13 +
##     offset(log(Duration)), data = Buchannan, init.theta = 2.62617933,
##     link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.5839     1.0853  -2.381  0.01727 *
## Global_risk_O18     0.7692     0.2358   3.262  0.00111 **
## Regional_risk_C13   0.1699     0.1637   1.038  0.29944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(2.6262) family taken to be 1)
##
```

```
##     Null deviance: 15.3273  on 6  degrees of freedom
## Residual deviance:  6.9735  on 4  degrees of freedom
## AIC: 63.536
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.63
##           Std. Err.:  1.47
##
##  2 x log-likelihood:  -55.536
```

The global measure of temperature seems to have more influence over the number of projectile point types, but how much more important is it than regional precipitation? Determining this requires that each variable be scaled so they vary by the order of magnitude and are centered on the mean, which also has the added benefit of making the intercept parameter represent the expected count under average conditions.

5.3.2.3 Variable importance Refit the model with scaled variables so that each vary by the same order of magnitude and are centered on their mean. This can be done using the `scale` function:

```
glm_nb.point_multi_s <- glm.nb(Number.of.point.types ~ scale(Global_risk_018) +
                                                       scale(Regional_risk_C13) +
                               offset(log(Duration)),
                               data = Buchannan
                               )
```

Note how the coefficients are different than above:

```
coeff.point_multi_s <- summary(glm_nb.point_multi_s)$coefficients

coeff.point_multi_s
```

```
##                            Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)              -4.1723183  0.2576743 -16.192220 5.722598e-59
## scale(Global_risk_018)    1.0516325  0.3223692   3.262199 1.105515e-03
## scale(Regional_risk_C13)  0.2968273  0.2860595   1.037642 2.994368e-01
```

Note, to plot these below, we will use some standard indexing to select specific rows and columns. This is done with a square bracket following the object. For example, if we wanted only the estimates corresponding to the predictor variables, we'd want to drop the first row and select only the first column:

```
coeff.point_multi_s[-1,1]
```

```
##   scale(Global_risk_018) scale(Regional_risk_C13)
##                1.0516325                0.2968273
```

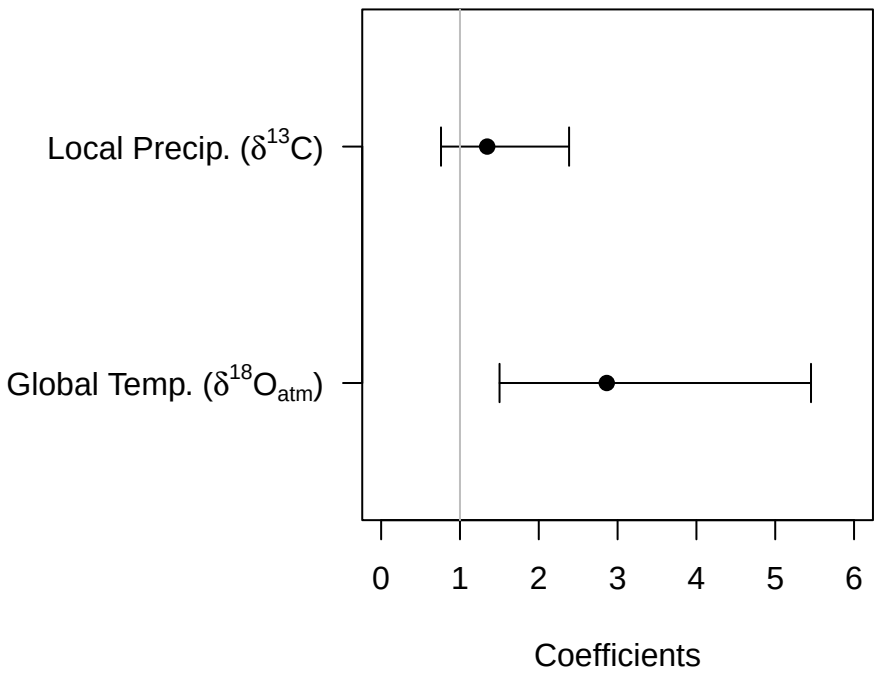
Note that when one coefficient is negative and one is positive, indicating a different directional response, it may be useful to aid comparison of the magnitude of effect by taking the absolute value (`abs`) of the scaled coefficients. If this is done, be sure to note so.

```
abs(coeff.point_multi_s[-1,1])
```

```
##   scale(Global_risk_018) scale(Regional_risk_C13)
##                1.0516325                0.2968273
```

Plot the value of the scaled coefficients and their standard error:

```
par(pty = "s", oma = c(0,1,0,0)) #square plot, add outer margin to left side

plot(NA,
     xlab = "Coefficients",
     ylab = NA,
     xlim = c(0, 6),
     ylim = c(0.5, 2.5),
     yaxt = "n"
     )

#points can plot the scaled coefficients
points(x = exp(coeff.point_multi_s[-1,1]), #select predictor coeffs
       y = 1:2,
       pch = 19)

#the arrows function can draw "whiskers" representing the 95% confidence intervals
arrows(x0 = exp(coeff.point_multi_s[-1,1] + (2*coeff.point_multi_s[-1,2])),
       x1 = exp(coeff.point_multi_s[-1,1] - (2*coeff.point_multi_s[-1,2])),
       y0 = 1:2,
       y1 = 1:2,
       length = 0.1,
       code = 3,
       angle = 90
       )

axis(side = 2,
     at = 1:2,
     rev(c(expression(paste("Local Precip. (", delta ^{13},"C)")),
           expression(paste("Global Temp. (", delta ^{18},"O",''[atm],")")))),
     las = 1
     )

abline(v = 1, col = "grey")
```



Both variables have positive effects, as predicted, but the local precipitation proxy has confidence intervals that overlap with 1 (meaning no response) while global temperature has only values above 1.

5.3.2.4 Prediction We can examine the effect of each variable on the response while holding the other constant. These are called partial or marginal response plots.

For interpretive purposes, we are going to return back to the un-scaled model and make predictions for each variable. As above, we will make predictions across the range of each focal variable, though here we need to specify the value at which we want to hold the other variable constant. Often it is recommended to go with the `mean` or `median` value, though other values can allow the investigator to explore different scenarios.

In this case, let's explore how the number of projectile point types varies with each predictor while holding the other constant at their *minimum* value so to evaluate the scenario where "environmental risk" is lowest for one predictor across the range of the other. In other words, this will allow us to assess how investment in specialized technology varies with temperature when precipitation is high (i.e., when precipitation induced risk is low), and with precipitation when temperature is low (i.e., when temperature induced risk is low).

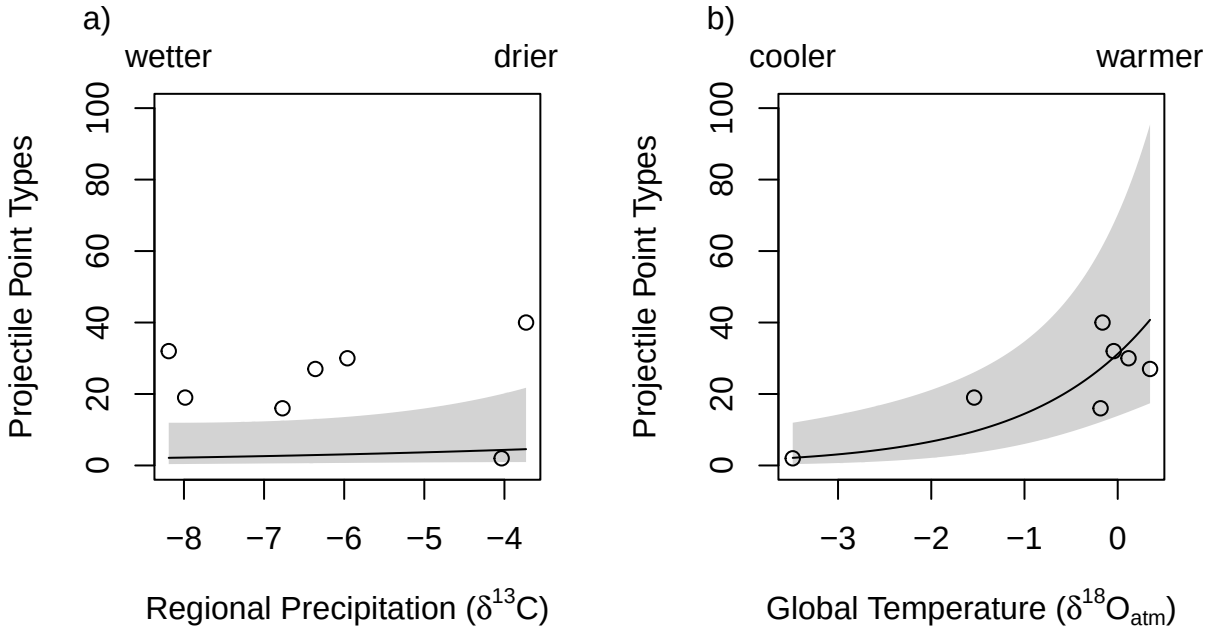
```
#a sequence across the range regional risk
reg.seq <- seq(min(Buchannan$Regional_risk_C13),
               max(Buchannan$Regional_risk_C13),
                length.out = 100) #arbitrary 100 values

con_pred.reg <- predict(glm_nb.point_multi,
                        newdata = data.frame(
                          #predict across the range of precip
                          Regional_risk_C13 = reg.seq,
                          #hold temp at the min value
```

```
                          Global_risk_O18 = min(Buchannan$Global_risk_O18),
                          #hold at the median duration
                          Duration = median(Buchannan$Duration)
                          ),
                        type = "link",
                        se = TRUE
                        )

#a sequence across the range global risk
glo.seq <- seq(min(Buchannan$Global_risk_O18),
               max(Buchannan$Global_risk_O18),
                length.out = 100) #arbitrary 100 values

con_pred.glo <- predict(glm_nb.point_multi,
                        newdata = data.frame(
                          #hold precip at the min value
                          Regional_risk_C13 = min(Buchannan$Regional_risk_C13),
                          #predict across the range of temp
                          Global_risk_O18 = glo.seq,
                          #hold at the median duration
                          Duration = median(Buchannan$Duration)
                          ),
                        type = "link",
                        se = TRUE
                        )
```

Plot the predicted values side-by-side with 95% confidence intervals.

```
#pdf("Figure6.pdf", height = 4, width = 7)

par(mfrow = c(1,2), pty = "s")

#blank plot
with(Buchannan,
     plot(NA,
          ylim = c(0,100),
          xlim = c(min(reg.seq), max(reg.seq)),
          xlab = expression(paste("Regional Precipitation (", delta ^{13},"C)")),
          ylab = "Projectile Point Types"
          )
     )

#add 95% CI
polygon(x = c(reg.seq, rev(reg.seq)),
        y = c(inv_link.glm_Pois(con_pred.reg$fit + (2*con_pred.reg$se)),
              rev(inv_link.glm_Pois(con_pred.reg$fit - (2*con_pred.reg$se)))
              ),
        border = NA,
        col = "lightgrey"
        )

#add model fit
lines(exp(con_pred.reg$fit) ~ reg.seq)
```

```r
#add the data points
with(Buchannan,
     points(Number.of.point.types ~ Regional_risk_C13)
     )

#add a panel label for publication
mtext("a)", side = 3, adj = -0.2, line = 1.5)

#add text in the margin of the plot to aid interpretation
mtext(text = c("wetter", "drier"),
      at = c(min(reg.seq), max(reg.seq)),
      side = 3,
      line = 0.5
      )



#blank plot
with(Buchannan,
     plot(NA,
          ylim = c(0,100),
          xlim = c(min(glo.seq), max(glo.seq)),
          xlab = expression(paste("Global Temperature (", delta ^{18},"O",''[atm],")")),
          ylab = "Projectile Point Types"
          )
     )

#add 95% CI
polygon(x = c(glo.seq, rev(glo.seq)),
        y = c(inv_link.glm_Pois(con_pred.glo$fit + (2*con_pred.glo$se)),
              rev(inv_link.glm_Pois(con_pred.glo$fit - (2*con_pred.glo$se)))
              ),
        border = NA,
        col = "lightgrey"
        )

#add model fit
lines(exp(con_pred.glo$fit) ~ glo.seq)

#add points
with(Buchannan,
     points(Number.of.point.types ~ Global_risk_O18)
     )

#add a panel label for publication
mtext("b)", side = 3, adj = -0.2, line = 1.5)

#add text in the margin of the plot to aid interpretation
mtext(text = c("cooler", "warmer"),
      at = c(min(glo.seq), max(glo.seq)),
      side = 3,
      line = 0.5
      )
```



The results illustrate that even holding global temperature to cooler conditions, an increase in regional precipitation does not lead to a increase in projectile point types, while an increase in global temperature even holding regional precipitaiton to wetter conditions does lead to an increase in projectile point types.

---

6 References

Bolker, Ben and others (2022). GLMM FAQ. http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html. Updated 05 Oct 2022. Accessed 06 May 2023.

Buchanan, Briggs, Michael J. O'Brien, and Mark Collard (2016). Drivers of technological richness in prehistoric Texas: An archaeological test of the population size and environmental risk hypotheses. *Archaeological and Anthropological Sciences* 8: 625-634.

Carlson, David L. (2017). *Quantitative Methods in Archaeology Using R.* Cambridge University Press, pp 171-183, 232-242.

Codding, Brian F., Zeanah, Bleige Bird, Rebecca, Parker, Christopher Hugh Liam, and Bird, Douglas W. (2016). Martu ethnoarchaeology: Foraging ecology and the marginal value of site structure. *Journal of Anthropological Archaeology*, 44, 166-176.

Cogswell, J. W., M. J. O'Brien, and D. S. Glover. (2001). The Artifactual Content of Selected House Floors at Turner and Snodgrass. In *Mississippian Community Organization: The Powers Phase in Southeastern Missouri*, edited by M. J. O'Brien, pp 181–229. Kluwer Academic/Plenum.

Drennan, Robert D. (2009). *Statistics for archaeologists.* New York: Springer.

Fox, John and Sanford Weisberg (2019). *An R Companion to Applied Regression*, Third Edition. Thousand Oaks CA: Sage. URL: https://socialsciences.mcmaster.ca/jfox/Books/Companion/

Kolb, Charles. C. (1985). Demographic estimates in archaeology: contributions from ethnoarchaeology on Mesoamerican peasants. *Current Anthropology*, 26(5), 581-599.

Manning, Katie (2016). The cultural evolution of Neolithic Europe. EUROEVOL dataset 2: zooarchaeological data. *Journal of Open Archaeology Data*, 5.

Manning, Katie, Timpson, Adrian, Colledge, Sue, Crema, Enrico, and Shennan, Stephen (2015). The Cultural Evolution of Neolithic Europe. EUROEVOL Dataset. [Dataset]. discovery.ucl.ac.uk/id/eprint/1469811/

Price, J. E. and J. B. Griffin. (1979). *The Snodgrass Site of the Powers Phase of Southeast Missouri.* Anthropological Papers. Museum of Anthropology, University of Michigan, No. 66.

Shott, Michael J. (2018). *Pottery ethnoarchaeology in the Michoac'an Sierra.* University of Utah Press.

Shott, Michael J. (2022). Inferring Use-Life Mean and Distribution: A Pottery Ethnoarchaeological Case Study from Michoac'an. *American Antiquity*, 87(4), 794-815.

Simpson, Gavin L. (2018a). Confidence intervals for GLMs. *From the bottom of the heap* https://fromthebot tomoftheheap.net/2018/12/10/confidence-intervals-for-glms.

Simpson, Gavin L. (2018b) Modelling Palaeoecological Time Series Using Generalised Additive Models. *Frontiers in Ecology and Evolution.* 6:149. DOI: 10.3389/fevo.2018.00149.

Timpson, Adrian (2016). The Cultural Evolution of Neolithic Europe. EUROEVOL Dataset 1: Sites, Phases and Radiocarbon Data. https://openarchaeologydata.metajnl.com/articles/10.5334/joad.40

Torrence Robin (1983) Time budgeting and hunter-gatherer technology. In: Bailey G (ed), *Hunter-gatherer Economy in Prehistory*, pp 11–22. Cambridge University Press, Cambridge.

Wright, Elizabeth (2016). *The Morphological Variability of the European Aurochs (Bos primigenius) from the Middle Pleistocene to its Extinction: A zooarchaeological study.* BAR International Series 2815. DOI: 10.30861/9781407314839.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S.* Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Yellen, John (1977). *Archaeological Approaches to the Present.* Academic Press, New York.

7 Session Information

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## crt            ucrt
## system         x86_64, mingw32
## status
## major          4
## minor          3.0
## year           2023
## month          04
## day            21
## svn rev        84292
## language       R
## version.string R version 4.3.0 (2023-04-21 ucrt)
## nickname       Already Tomorrow
```

```
names(sessionInfo()$otherPkgs)
```

```
## [1] "car"     "carData" "MASS"    "mgcv"    "nlme"
```