STAT 245 Course Notes

Stacy DeRuiter, Calvin University

2019-09-26

Contents

Chapter 1

Description

This is a set of course notes distributed in STAT 245 at Calvin University in Fall 2019. Contact sld33 at calvin.edu with comments, corrections or suggestions.

Chapter 2

Linear Regression

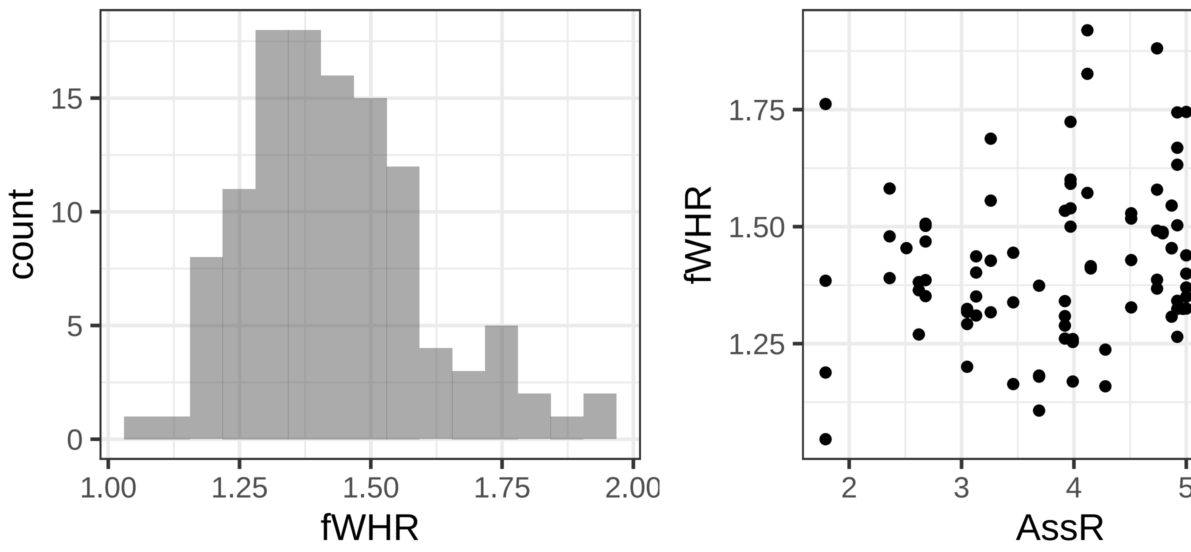
You probably learned something about linear regression in a previous course. Here, we briefly review the main concepts of simple linear regression and quickly expand our tool box to multiple regression (with both quantitative and categorical predictors).

2.1 Data

We will consider a small dataset from an article by J.S. Martin and colleagues, titled *Facial width-to-height ratio is associated with agonistic and affiliative dominance in bonobos (**Pan paniscus**)*

Notes: variable `fWHR` is the facial width-height ratio and `AssR` is the Assertiveness score of affiliative dominance. `normDS` is another dominance score. A few figures of the data are below - we will do some more exploration together.

```
## Observations: 117
## Variables: 8
## $ Name   <fct> Zuani, Zuani, Zorba, Zorba, Zorba, Zomi, Zomi, Zamba, Z...
## $ Group  <fct> Apenheul, Apenheul, Wilhelma, Wilhelma, Wilhelma, Frank...
## $ Sex    <fct> Female, Female, Male, Male, Male, Female, Female, Male,...
## $ Age    <int> 22, 22, 34, 34, 34, 15, 15, 14, 14, 14, 18, 18, 18, 18,...
## $ fWHR   <dbl> 1.475052, 1.321814, 1.581446, 1.479237, 1.390086, 1.340...
## $ AssR   <dbl> 5.36, 5.36, 2.36, 2.36, 2.36, 3.92, 3.92, 4.74, 4.74, 4...
## $ normDS <dbl> 1.430, 1.430, 2.341, 2.341, 2.341, 3.087, 3.087, 3.035,...
## $ weight <dbl> 24.0, 24.0, NA, NA, NA, NA, NA, 41.6, 41.6, 41.6, 38.0,...
```



2.2 Simple linear regression, Residuals & Least squares

First, let's review and consider a simple (one-predictor) linear regression model. Fit the model

```
slr <- lm(fWHR ~ AssR, data=bonobos)
```

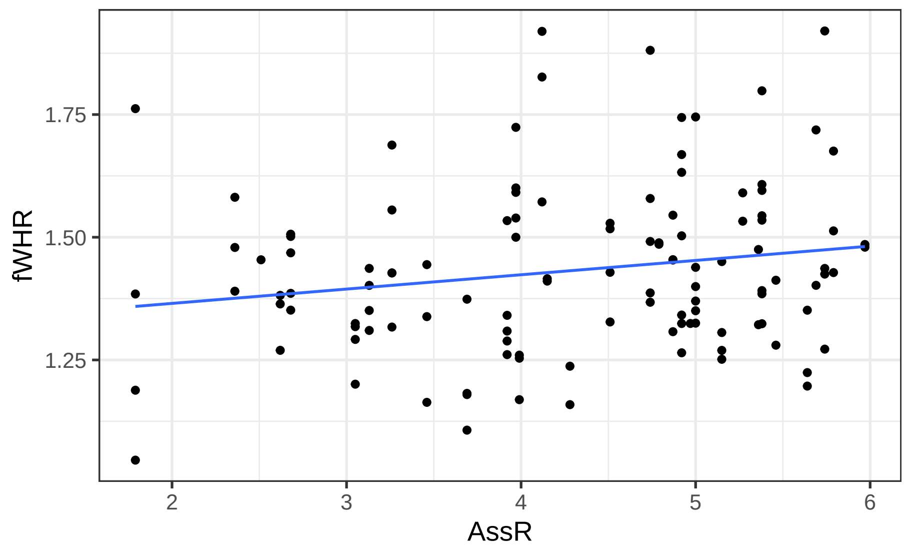
Extract the slope and intercept values:

```
coef(slr)
```

```
## (Intercept)        AssR
##  1.30685287  0.02918242
```

Add the regression line to the plot:

```
gf_point(fWHR ~ AssR, data=bonobos) %>%
  gf_lm()
```



```r
summary(slr)
```

```
##
## Call:
## lm(formula = fWHR ~ AssR, data = bonobos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31320 -0.11369 -0.01242  0.09008  0.49241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.30685    0.06283  20.801   <2e-16 ***
## AssR         0.02918    0.01420   2.055   0.0421 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1689 on 115 degrees of freedom
## Multiple R-squared:  0.03542,    Adjusted R-squared:  0.02704
## F-statistic: 4.223 on 1 and 115 DF,  p-value: 0.04213
```

2.2.1 Using `lm()` to fit a linear regression in R

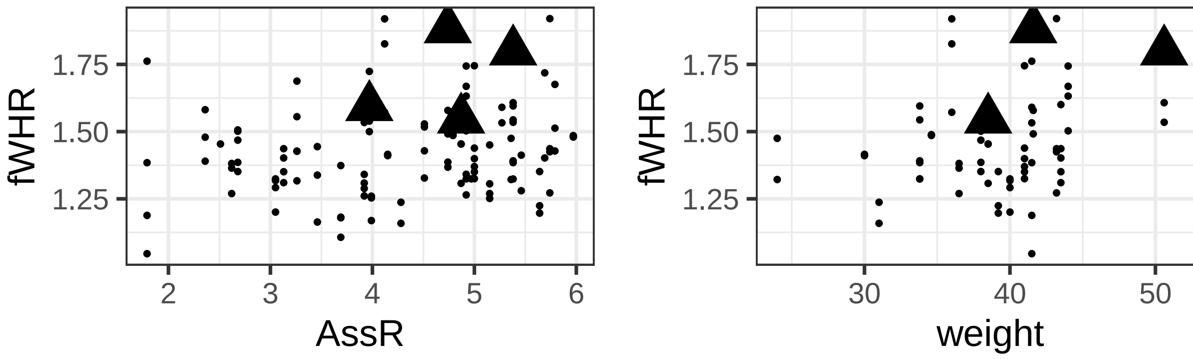
2.2.2 Equation of the fitted regression line

2.3 Multiple regression

Rarely does our response variable **really** depend on only one predictor. Can we improve the model by adding more predictors?

```r
mlr <- lm(fWHR ~ AssR + weight, data=bonobos)
coef(mlr)
```

```
## (Intercept)        AssR      weight
## 0.944790930 0.039888045 0.008644299
```



2.3.1 Is it really better?

How do we know if the model with more predictors is "better"? (For a more detailed answer, wait about a week...) But before we can define a "beter" model: how did R find the "best" intercept and slopes?

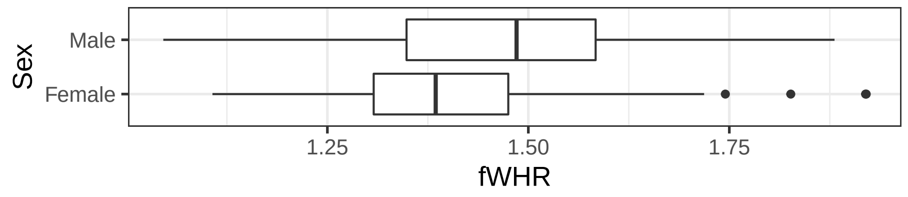
2.3.2 Regression residuals = "errors"

2.3.3 Computing Predictions

Use the regression equation to compute **predicted values** for the three data points below:

```
##        fWHR AssR weight
## 8  1.880866 4.74   41.6
## 25 1.798387 5.38   50.6
## 41 1.591440 3.97     NA
## 65 1.545019 4.87   38.5
```

2.4 Predictors with two categories



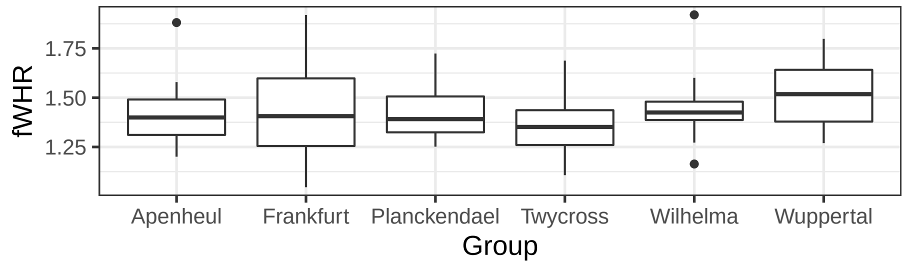
```
mlr2 <- lm(fWHR ~ AssR + weight + Sex, data = bonobos)
coef(mlr2)
```

```
## (Intercept)        AssR      weight     SexMale
## 1.065420976 0.058435841 0.002257142 0.128484275
```

How does the model incorporate this covariate mathematically?

2.4.1 Predictors with more categories

```
gf_boxplot(fWHR ~ Group, data = bonobos)
```



```
mlr3 <- lm(fWHR ~ AssR + weight + Sex + Group, data = bonobos)
coef(mlr3)
```

```
##       (Intercept)              AssR            weight           SexMale 
##       1.007734691       0.064361973       0.003458979       0.124854271 
##    GroupFrankfurt GroupPlanckendael     GroupTwycross     GroupWilhelma 
##       0.037426358      -0.008464572      -0.112907589       0.011186724 
##    GroupWuppertal 
##      -0.004364826
```

How does the model incorporate **this** covariate mathematically?

2.5 Returning to the R Model Summary

There are several bits of information you should be able to extract from the `summary()` output R produces on a fitted linear regression model:

- βs, Coefficient Estimates
- σ, labeled "residual standard error"
- R^2 (adjusted)

```
mlr3 <- lm(fWHR ~ AssR + weight + Sex + Group, data = bonobos)
summary(mlr3)
```

```
## 
## Call:
## lm(formula = fWHR ~ AssR + weight + Sex + Group, data = bonobos)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.38288 -0.09488 -0.02642  0.07196  0.48464 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        1.007735   0.217585   4.631 2.05e-05 ***
## AssR               0.064362   0.021158   3.042   0.0035 ** 
## weight             0.003459   0.005547   0.624   0.5353    
## SexMale            0.124854   0.059278   2.106   0.0394 *  
## GroupFrankfurt     0.037426   0.074892   0.500   0.6191    
## GroupPlanckendael -0.008465   0.075407  -0.112   0.9110    
## GroupTwycross     -0.112908   0.074779  -1.510   0.1364    
## GroupWilhelma      0.011187   0.085538   0.131   0.8964    
## GroupWuppertal    -0.004365   0.071292  -0.061   0.9514    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1691 on 59 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:  0.2517, Adjusted R-squared:  0.1502
## F-statistic:  2.48 on 8 and 59 DF,  p-value: 0.02167
```

2.6 Predictions from the model

2.6.1 By Hand

The equation for the fitted model above is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 I_{Male} + \beta_4 I_{Frankfurt} + \beta_5 I_{Planckendael} + \beta_6 I_{Twycross} + \beta_7 I_{Wilhelma} + \beta_7 I_{Wuppertal} + \epsilon$$

where

- $y =$
- $\beta_0 =$

- $x_1 =$
- $x_2 =$
- $\beta_1, \beta_2, \beta_3 ...$ are:
- $I_{Male} =$
- $I_{Frankfurt} =$
- $I_{Planckendael} =$, etc.
- $\epsilon =$

2.6.1.1 Comprehension check:

What is the expected fWHR (according to this model) for a 30 kg female bonobo at the Wilhelma zoo?

2.6.2 Prediction Plots in R

We can ask R to compute predictions for **all** the data points in the real dataset.

```
bonobos <- bonobos %>%
  mutate(preds = predict(mlr3))
```

```
## Error: Column `preds` must be length 117 (the number of rows) or one, not 68
```

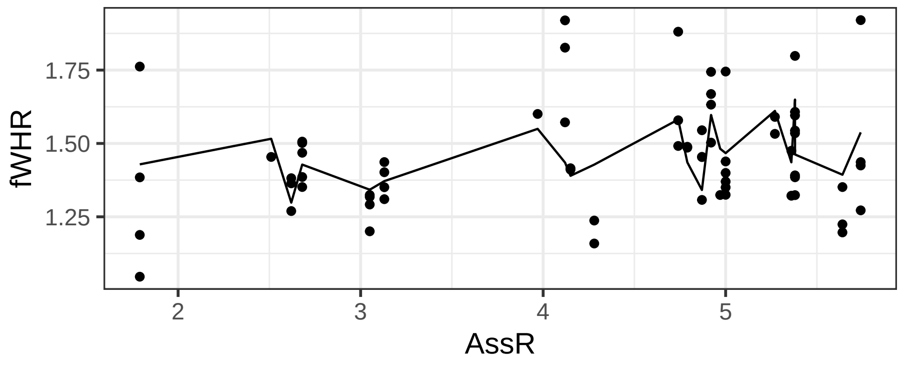
Wait, what? This error is because the `lm()` function removes rows containing missing values from the dataset, so it computes only 68 residuals (for the complete cases in the data). This doesn't match the 117 rows in the original data. We can solve the problem by omitting rows with missing values first. To be safe, we first select only the variables we need, so we don't omit rows based on missing values in unused variables.

```
b2 <- bonobos %>%
  dplyr::select(fWHR, weight, AssR, Sex, Group) %>%
  na.omit() %>%
  mutate(preds = predict(mlr3))
```

We have a full set of predictions!

But if we plot these predictions on a scatter plot of `fWHR` as a function of `AssR`, we *do not* get a straight line, because the predictions are also impacted by varying values of `weight`, `Sex`, and `Group`:

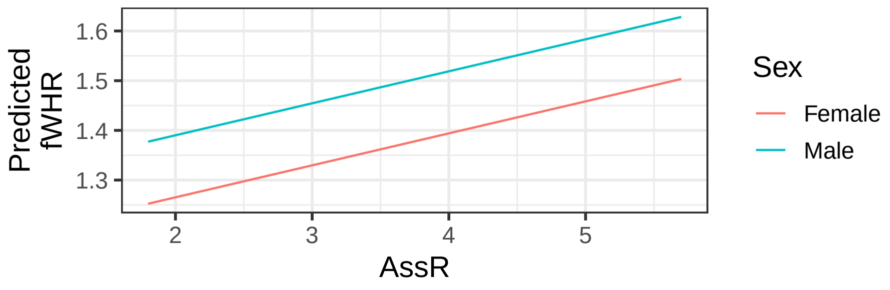
```
gf_point(fWHR ~ AssR, data = b2) %>%
  gf_line(preds ~ AssR, data=b2)
```



But...we would really like a straight line that helps us visualize the meaning of the β (slope coefficient) for `AssR`*.* We can make predictions for a **hypothetical** dataset, in which `AssR` varies over a reasonable range, but the other predictors stay constant. This lets us see how `AssR` (and only `AssR`) affects the response, without contributions from other predictors. In choosing the values to include in hypothetical dataset, we often choose to hold variables constant at their most common or median values, but not blindly: also, avoid impossible or implausible variable combinations (for example, specifying that a person lives in the state of Michigan but the city of Chicago, or that they are a 5-year-old person with 4 children). *In this case, to match the figures in the published paper, we are also going to vary the* `Sex` *- but generally you'd only allow one predictor to vary.*

```r
fake_data <- expand.grid(AssR = seq(from=1.8, to=5.7, by=0.05),
                         weight = 38.5,
                         Sex = c('Female', 'Male'),
                         Group = 'Wuppertal')

fake_data <- fake_data %>%
  mutate(preds = predict(mlr3, newdata = fake_data))
gf_line(preds ~ AssR, color = ~Sex, data=fake_data) %>% gf_labs(y='Predicted\nfWHR')
```

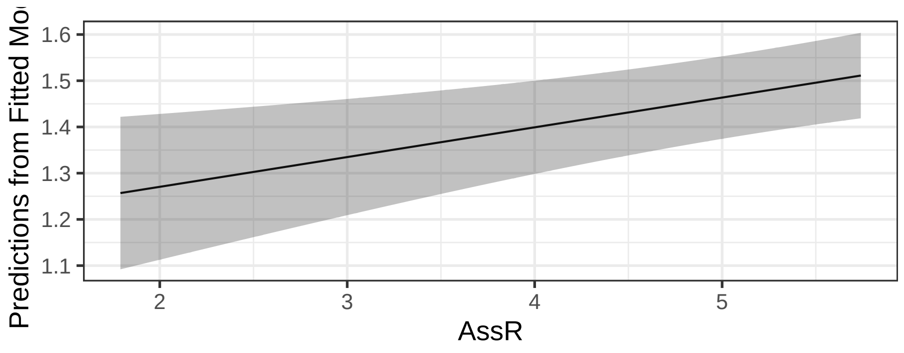


2.6.2.1 Comprehension checks:

- Should we overlay prediction-plot line(s) on the data scatter plot?
- How do you think the plot would look if we changed the constant predictor values?
- What is missing from this picture?

2.6.2.2 Shortcut

```r
require(s245)
pred_plot(mlr3, 'AssR')
```



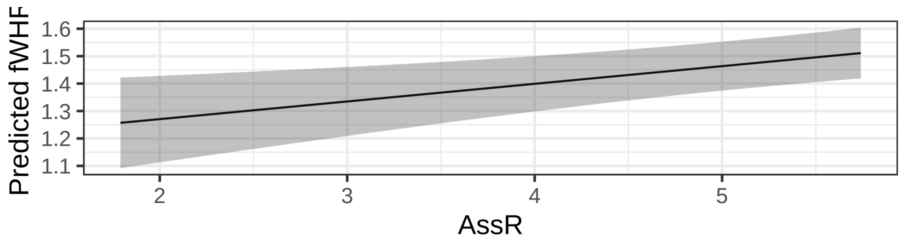
2.7 Why are we doing this again?

Why make prediction plots?

2.8 Shortcut Method - With Uncertainty

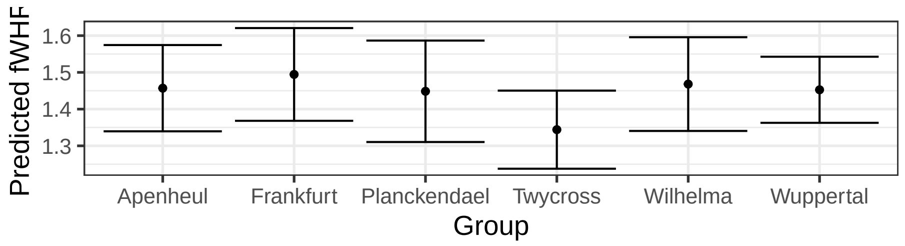
We saw before that `pred_plot()` makes it very easy for us to generate prediction plots showing what a (multiple regression) model says about the relationship between the response and *one* of the predictors:

```r
require(s245)
pred_plot(mlr3, 'AssR') %>%
  gf_labs(y = 'Predicted fWHR')
```



Note the custom axis label - otherwise you get a long, unwieldy default "Predictions from fitted model"

```r
require(s245)
pred_plot(mlr3, 'Group') %>%
  gf_labs(y = 'Predicted fWHR')
```



They look nice! But they should raise two questions:

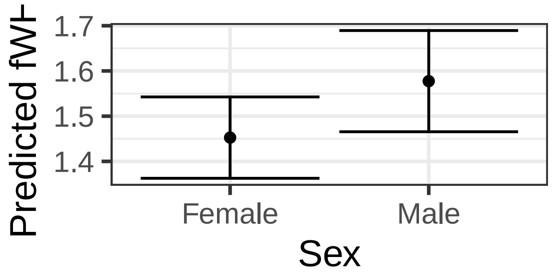
- Uncertainty:
- Fixed values:

```r
get_fixed(bonobos)  %>%
  pander::pander()
```

Name	Group	Sex	Age	fWHR	AssR	normDS	weight	three	pt_size
Eja	Twycross	Female	21	1.412	4.51	2.368	40	no	1

2.8.1 Anatomy of a Confidence Interval

```r
pred_plot(mlr3, 'Sex') %>%
  gf_labs(y = 'Predicted fWHR')
```



2.9 DIY Method

2.9.1 Creating a hypothetical dataset

We would like to create a hypothetical dataset where one predictor variable varies, and all the rest stay fixed. Let's choose `AssR`. We use `expand.grid()`:

```
fake_data <- expand.grid(AssR = seq(from=1.8, to=5.7, by=0.05),
                         weight = 40,
                         Sex = 'Female',
                         Group = 'Twycross')
glimpse(fake_data)
```

```
## Observations: 79
## Variables: 4
## $ AssR   <dbl> 1.80, 1.85, 1.90, 1.95, 2.00, 2.05, 2.10, 2.15, 2.20, 2...
## $ weight <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,...
## $ Sex    <fct> Female, Female, Female, Female, Female, Female, Female,...
## $ Group  <fct> Twycross, Twycross, Twycross, Twycross, Twycross, Twycr...
```

Now, make predictions for our fake data.

```
preds <- predict(mlr3, newdata = fake_data, se.fit = TRUE)
fake_data <- fake_data %>%
  mutate(fitted = preds$fit,
         se.fit = preds$se.fit)
glimpse(fake_data)
```

```
## Observations: 79
## Variables: 6
## $ AssR   <dbl> 1.80, 1.85, 1.90, 1.95, 2.00, 2.05, 2.10, 2.15, 2.20, 2...
## $ weight <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,...
## $ Sex    <fct> Female, Female, Female, Female, Female, Female, Female,...
## $ Group  <fct> Twycross, Twycross, Twycross, Twycross, Twycross, Twycr...
## $ fitted <dbl> 1.149038, 1.152256, 1.155474, 1.158692, 1.161910, 1.165...
## $ se.fit <dbl> 0.08347207, 0.08267088, 0.08187552, 0.08108616, 0.08030...
```

How do we go from *standard errors* to *confidence intervals*? We can either do this before plotting, or while plotting. To do it before and add the results to the hypothetical dataset:

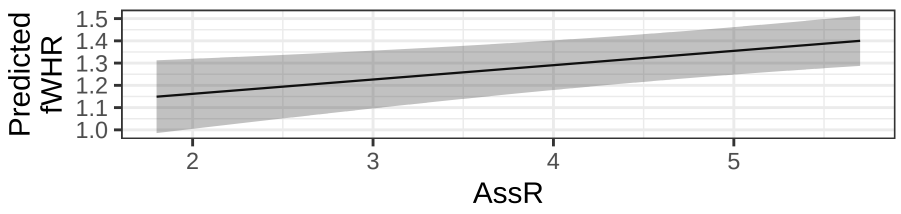
```
fake_data <- fake_data %>%
  mutate(CI_lower = fitted - 1.96*se.fit,
         CI_upper = fitted + 1.96*se.fit)
glimpse(fake_data)
```

```
## Observations: 79
## Variables: 8
## $ AssR     <dbl> 1.80, 1.85, 1.90, 1.95, 2.00, 2.05, 2.10, 2.15, 2.20,...
## $ weight   <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 4...
## $ Sex      <fct> Female, Female, Female, Female, Female, Female, Femal...
## $ Group    <fct> Twycross, Twycross, Twycross, Twycross, Twycross, Twy...
## $ fitted   <dbl> 1.149038, 1.152256, 1.155474, 1.158692, 1.161910, 1.1...
## $ se.fit   <dbl> 0.08347207, 0.08267088, 0.08187552, 0.08108616, 0.080...
## $ CI_lower <dbl> 0.9854326, 0.9902210, 0.9949980, 0.9997632, 1.0045164...
## $ CI_upper <dbl> 1.312643, 1.314291, 1.315950, 1.317621, 1.319304, 1.3...
```

2.9.2 Making the plot

Now, we just need to plot!

```
gf_line(fitted ~ AssR, data=fake_data) %>%
  gf_labs(y='Predicted\nfWHR') %>%
  gf_ribbon(CI_lower + CI_upper ~ AssR, data = fake_data)
```



If we wanted to figure out the CI bounds *while* plotting, we could calculate them on the fly like this:

```
gf_line(fitted ~ AssR, data=fake_data) %>%
  gf_labs(y='Predicted\nfWHR') %>%
  gf_ribbon((fitted - 1.96*se.fit ) + (fitted + 1.96*se.fit) ~ AssR,
            data = fake_data)
```

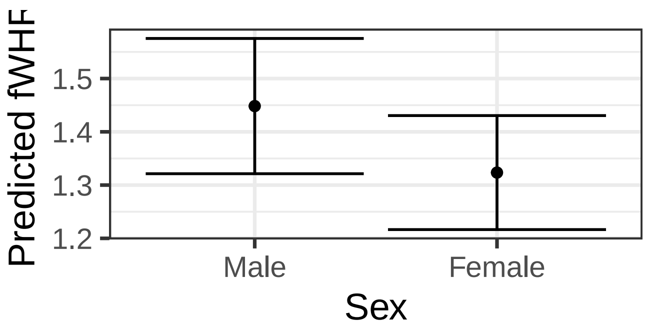
(which will look just the same).

2.9.3 Categorical predictors

What will be different if the predictor of interest is *categorical*?

- hypothetical data:
- plot:

```
fake_sex_data <- expand.grid(AssR = 4.51,
                             weight = 40,
                             Sex = c('Male', 'Female'),
                             Group = 'Twycross')
preds <- predict(mlr3, newdata = fake_sex_data, se.fit = TRUE)
fake_sex_data <- fake_sex_data %>%
  mutate(fitted = preds$fit,
         se.fit = preds$se.fit)
gf_point(fitted ~ Sex, data=fake_sex_data) %>%
  gf_labs(y='Predicted fWHR') %>%
  gf_errorbar((fitted - 1.96*se.fit ) + (fitted + 1.96*se.fit) ~ Sex,
            data = fake_sex_data)
```



Chapter 3

Model Selection Using Information Criteria

So far, we have learned to fit models with multiple predictors, both quantitative and categorical, and to assess whether required conditions are met for linear regression to be an appropriate model for a dataset.

One missing piece is: If I have an appropriate model with a set of multiple predictors, how can I choose which predictors are worth retaining in a "best" model for the data (and which ones have no relationship, or a weak relationship, with the response, so should be discarded)?

3.1 Data and Model

Today we will recreate part of the analysis from *Vertebrate community composition and diversity declines along a defaunation gradient radiating from rural villages in Gabon*, by Sally Koerner and colleagues. They investigated the relationship between rural villages, hunting, and wildlife in Gabon. They asked how monkey abundance depends on distance from villages, village size, and vegetation characteristics. They shared their data at Dryad.org and we can read it in and fit a regression model like this:

```r
defaun <- read.csv('http://sldr.netlify.com/data/koerner_gabon_defaunation.csv')
```

```r
ape_mod <- lm(RA_Apes ~ Veg_DBH + Veg_Canopy + Veg_Understory +
                     Veg_Rich + Veg_Stems + Veg_liana +
                     LandUse + Distance + NumHouseholds, data = defaun)
summary(ape_mod)
```

```
##
## Call:
## lm(formula = RA_Apes ~ Veg_DBH + Veg_Canopy + Veg_Understory +
##     Veg_Rich + Veg_Stems + Veg_liana + LandUse + Distance + NumHouseholds,
##     data = defaun)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9857 -0.9419 -0.0360  0.8239  6.3832
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.752517  13.372210   0.430   0.6741
## Veg_DBH        -0.093171   0.073114  -1.274   0.2249
## Veg_Canopy      0.670094   2.062545   0.325   0.7504
## Veg_Understory -1.691235   2.071299  -0.817   0.4289
## Veg_Rich        0.361960   0.480362   0.754   0.4646
## Veg_Stems      -0.097211   0.169073  -0.575   0.5751
## Veg_liana      -0.158505   0.253031  -0.626   0.5419
## LandUseNeither  1.696755   2.058937   0.824   0.4247
## LandUsePark    -2.947189   2.222710  -1.326   0.2077
## Distance        0.302626   0.119865   2.525   0.0254 *
## NumHouseholds  -0.002107   0.043458  -0.048   0.9621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.725 on 13 degrees of freedom
## Multiple R-squared:  0.5439, Adjusted R-squared:  0.1931
## F-statistic: 1.551 on 10 and 13 DF,  p-value: 0.2262
```

```r
as.numeric(logLik(ape_mod))
```

```
## [1] -50.75799
```

3.2 Calculations

- Information criteria allow us to **balance the conflicting goals** of having a model that *fits the data as well as possible* (which pushes us toward models with more predictors) and *parsimony* (choosing the simplest model, with the fewest predictors, that works for the data and research question). The basic idea is that we **minimize** the quantity $-(2LogLikelihood - penalty) = -2LogLikelihood + penalty$

- AIC is computed according to $-2LogLikelihood + 2k$, where k is the number of coefficients being estimated (don't forget σ!) **Smaller AIC is better.**
- BIC is computed according to $-2LogLikelihood + ln(n)k$, where n is the number of observations (rows) in the dataset and k is the number of coefficients being estimated. **Smaller BIC is better.**
- Verify that the BIC for this model is 139.65.

3.3 Decisions with ICs

The following rules of thumb (**not** laws, just common rules of thumb) may help you make decisions with ICs:

- A model with lower IC *by at least 3 units* is notably better.
- If two or more models have ICs *within* 3 IC units of each other, there is not a lot of difference between them. Here, we usually choose the model with fewest predictors.
- In some cases, if the research question is to measure the influence of some particular predictor on the response, but *the IC does not strongly support including that predictor* in the best model (IC difference less than 3), you might want to keep it in anyway and then discuss the situation honestly, for example, "AIC does not provide strong support for including predictor x in the best model, but the model including predictor x indicates that as x increases the response decreases slightly. More research would be needed…"

3.4 All-possible-subsets Selection

The model we just fitted is our *full model*, with all predictors of potential interest included. How can we use information criteria to choose the best model from possible models with subsets of the predictors?

We can use the `dredge()` function from the `MuMIn` package to get and display ICs for all these models.

Before using dredge, we need to make sure our dataset has no missing values, and also set the "na.action" input for our model (can be done in call to `lm(..., na.action = 'na.fail')` also).

```r
require(MuMIn)
ape_mod <- ape_mod %>% update(na.action = 'na.fail')
ape_dredge <- dredge(ape_mod, rank='BIC')
```

```
## Fixed term is "(Intercept)"
```

```
pander::pander(head(ape_dredge, 7))
```

Table 3.1: Table continues below

	(Intercept)	Distance	LandUse	NumHouseholds	Veg_Canopy
258	8.753	0.195	NA	NA	NA
2	-0.6912	0.2303	NA	NA	NA
274	11.44	0.1848	NA	NA	NA
322	11.9	0.2033	NA	NA	NA
290	9.805	0.1884	NA	NA	NA
386	9.49	0.1976	NA	NA	NA
266	7.783	0.1896	NA	NA	0.2771

Table 3.2: Table continues below

	Veg_DBH	Veg_liana	Veg_Rich	Veg_Stems	Veg_Understory	df
258	NA	NA	NA	NA	-2.988	4
2	NA	NA	NA	NA	NA	3
274	-0.04551	NA	NA	NA	-3.144	5
322	NA	NA	-0.1939	NA	-3.11	5
290	NA	-0.09802	NA	NA	-2.952	5
386	NA	NA	NA	-0.03113	-2.904	5
266	NA	NA	NA	NA	-2.964	5

	logLik	BIC	delta	weight
258	-53.9	120.5	0	0.3284
2	-55.8	121.1	0.6241	0.2404
274	-53.38	122.7	2.146	0.1123
322	-53.55	123	2.491	0.09449
290	-53.67	123.2	2.727	0.08399
386	-53.82	123.5	3.03	0.0722
266	-53.88	123.7	3.144	0.0682

- What is the best model according to BIC, for this dataset?

3.5 Which IC should I use?

AIC and BIC may give different best models, especially if the dataset is large. You may want to just choose one to use *a priori* (before making calculations). You might prefer BIC if you want to err on the "conservative" side, as it is more likely to select a "smaller" model with fewer predictors. This is because of its larger penalty.

3.6 Quantities derived from AIC

- ΔAIC is the AIC for a given model, minus the AIC of the best one in the dataset. (Same for ΔBIC)
- *Akaike weights* are values (ranging from 0-1) that measure the weight of evidence suggesting that a model is the best one (given that there is one best one in the set)

3.7 Important Caution

Very important: IC can **ONLY** be compared for models with the same response variable, and the exact same rows of data.

Chapter 4

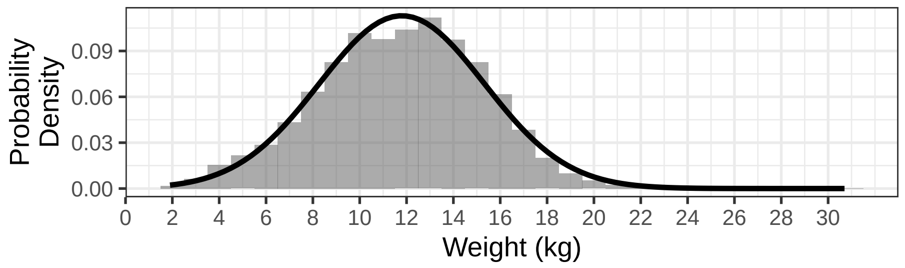
Likelihood

In the last section, we said that "likelihood" is a measure of goodness-of-fit of a model to a dataset. But what is it *exactly* and just how do we compute it?

4.1 Data

Today's dataset was collected in Senegal in 2015-2016 in a survey carried out by UNICEF, of 5440 households in the urban area of Dakar, Senegal. Among these households, information was collected about 4453 children under 5 years old, including their **weights in kilograms.**

```
gf_dhistogram(~AN3, data=wt, binwidth=1) %>%
  gf_labs(x='Weight (kg)', y='Probability\nDensity') %>%
  gf_fitdistr(dist='dnorm', size=1.3) %>%
  gf_refine(scale_x_continuous(breaks=seq(from=0, to=30, by=2)))
```



4.2 Review - the Normal probability density function (PDF)

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

4.3 A simple model

The distribution of weights looks quite unimodal and symmetric, so we will model it with a normal distribution with mean 11.8 and standard deviation 3.53 (N($\mu = 11.8$, $\sigma = 3.53$), black line).

4.4 Using the Model to Make Predictions

If you had to predict the weight of one child from this population, what weight would you guess?

Is it more likely for a child in Dakar to weigh 10kg, or 20kg? How much more likely?

What is the *probability* of a child in Dakar weighing 11.5 kg?

4.5 Likelihood to the Rescue!

Which is more likely: three children who weigh 11, 8.2, and 13kg, or three who weigh 10, 12.5 and 15 kg?

How did you:

- Find the likelihood of each observation?
- Combine the likelihoods of a set of three observations?

What did you have to assume about the set of observations?

4.6 How does this relate to linear regression?

What if we think of this situation as a linear regression problem (with no predictors)?

```r
lm_version <- lm(AN3 ~ 1, data = wt)
summary(lm_version)
```

```
##
## Call:
## lm(formula = AN3 ~ 1, data = wt)
##
## Residuals:
## <Labelled double>: Poids de l'enfant (kilogrammes)
##     Min      1Q  Median      3Q     Max
## -9.8964 -2.3964  0.1036  2.4036 18.9036
##
## Labels:
##  value            label
##   99.9 poids non mesuré
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79644    0.05435   217.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.529 on 4216 degrees of freedom
##   (235 observations deleted due to missingness)
```

4.6.1 Model Equation:

4.7 Likelihood of a dataset, given a model

Finally, now, we can understand what we were computing when we did

```r
logLik(lm_version)
```

```
## 'log Lik.' -11301.19 (df=2)
```

For our chosen regression model, we know that the residuals should have a normal distribution with mean 0 and standard deviation σ (estimated Residual Standard Error from R `summary()` output).

For each data point in the dataset, for a given regression model, we can compute a model prediction.

We can subtract the prediction from the observed response-variable values to get the residuals.

We can compute the **likelihood** (L) of this set of residuals by finding the likelihood of each individual residual e_i in a $N(0, \sigma)$ distribution.

To get the likelihood of the full dataset given the model, we use the fact that the residuals are independent (they better be, because that was one of the conditions of of linear regression model) – we can multiply the likelihoods of all the individual residuals together to get the joint likelihood of the full set.

That is the “likelihood” that is used in the AIC and BIC calculations we considered earlier.

Chapter 5

PDFs and PMFs

5.1 Beyond Normal

In our exploration of likelihoods, we did a little bit of work with the Normal probability density function. Here, we will state some characteristics of the normal distribution slightly more formally, and then we will get familiar with some other probability distributions (a.k.a. "the normal distribution's wierd friends").

5.2 Types of probability distributions

5.2.1 Continuous distributions

The probability distribution of a continuous variable (one that can take on continuous real values, at least within a certain range) is called a *probability density function* or **PDF** for short.

PDFs are functions of one continuous variable (we'll call it x) that have two properties in common:

- The total area under the curve is 1 ($\int_{-\infty}^{\infty} f(x)dx = 1$)
- The values of the function are non-negative (0, or a positive real number) for all real x ($f(x) \geq 0 \forall x \in \mathbb{R}$)

For PDFs, the function values (y-axis values) are *probability densities*, useful for computing likelihoods; probabilities are given by finding *areas* under the curve.

5.2.2 Discrete Distributions

We use categorical as well as quantitative variables in our regression models, so it will prove useful to have some discrete probability distributions as well as continuous ones. Discrete distributions associate each possible value of a discrete variable with its probability of occurence.

A discrete probability distribution is characterized by a *probability mass function* or PMF for short (this is the discrete equivalent of a PDF).

A PMF is a function of one *discrete* variable (we'll call it x) that have two properties in common:

- The sum of all the function's values is 1 ($\sum_{x \in S} f(x) = 1$, where S is the set of all possible values x can have)
- All values of the function are between 0 and 1 inclusive (they are probabilities) ($f(x) \in [0,1] \forall x \in S$)

5.3 Relevant Features of Distributions

For this course, the most important features to note about each probability distribuiton will be:

- The **type** of distribution: discrete or continuous?
- The **support** of the distribution: what range of possible values can the random variable X take on?
- The **parameters** of the distribution. Changing the parameters tunes the center and shape of the distribution, so these are what we need to estimate to fit a particular kind of distribution to a specific dataset. (For example, the parameters of the normal distribution are the mean μ and the standard deviation σ.)
- The **shape** of the distribution: what shapes can the function take one? (For example, the normal distribution is always unimodal and symmetric.)
- The PDF or PMF of the distribution. What mathematical expression controls the shape of the distribution?
- We will also note a few examples of variables that might be well modelled by each distribution.

5.4 Examples of Continuous Distributions

5.4.1 Normal

5.4.1.0.1 Type

Continuous

5.4.1.0.2 Support

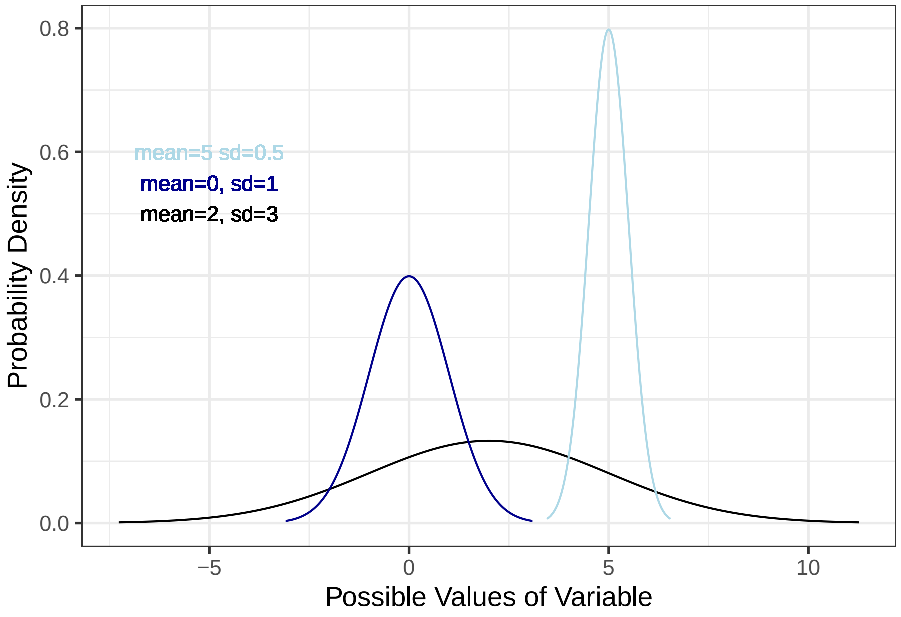
All real numbers

5.4.1.0.3 Parameters

- μ, the mean, which can take on any real value
- σ, the standard deviation, which can take on any positive real value

5.4.1.0.4 Shapes

The shape is always unimodal and symmetric.



5.4.1.0.5 PDF or PMF

The normal distribution has PDF:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

5.4.1.0.6 Examples

A normal distribution might be a good fit for data on childrens' weights in kg, or for the duration of visits at a zoo, or…

5.4.2 Gamma

5.4.2.0.1 Type

Continuous

5.4.2.0.2 Support

positive real numbers

5.4.2.0.3 Parameters

There are two alternate but equivalent parameterizations for the gamma distribution.

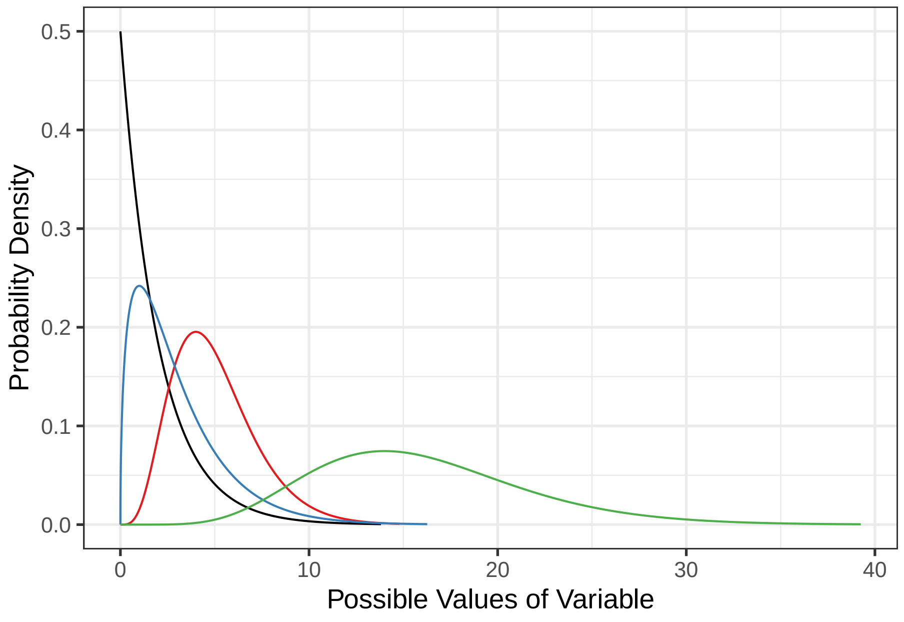
The first option: α (shape) and β (rate), where $\alpha > 0$ and $\beta > 0$

The second option: k (shape) and θ (scale), where $k > 0$ and $\theta > 0$.

Converting between the two parameterizations: $\alpha = k$ and $\beta = \frac{1}{\theta}$.

5.4.2.0.4 Shapes

The gamma distribution can take on a unimodal, symmetric shape or a unimodal shape with any amount of right skew (up to an exponential distribution shape).



Note: you don't need to be familiar with exactly how the different values of parameters influence the shape of the gamma distribution PDF, so the curves here are not labelled with parameter values.

5.4.2.0.5 PDF or PMF

The gamma distribution has PDF:

$$f(x) = \frac{1}{\Gamma(k)\theta^k} x^{(k-1)} e^{\frac{-x}{\theta}}$$

Where Γ is the Gamma function (look up the definition if you choose), or equivalently,

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{(\alpha-1)} e^{-\beta x}$$

5.4.2.0.6 Examples

Gamma distributions are often used to model things like wind speed or duration of an event (any quantity that might have right skew and is never negative).

5.4.3 Beta

5.4.3.0.1 Type

Continuous

5.4.3.0.2 Support

Real numbers between 0 and 1 ([0,1])

5.4.3.0.3 Parameters

α (shape 1) and β (shape 2), both of which must be > 1.

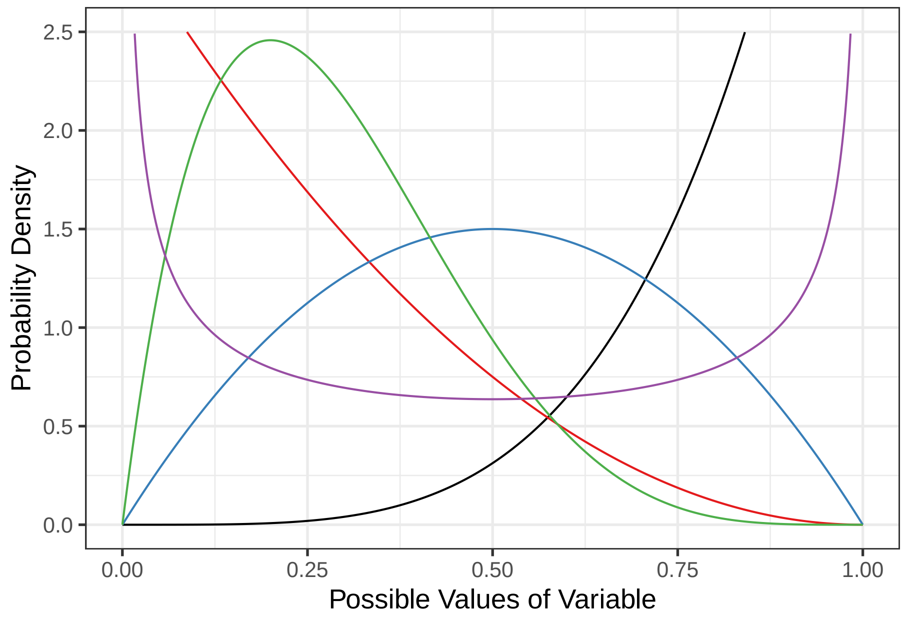
5.4.3.0.4 Shapes

This distribution can take on almost any shape, for example:

```
gf_dist('beta', params = c(shape1=5, shape2=1)) %>%
  gf_dist('beta', params = c(shape1=1, shape2=3), color=colrs[1]) %>%
  gf_dist('beta', params = c(shape1=2, shape2=2), color=colrs[2]) %>%
  gf_dist('beta', params = c(shape1=2, shape2=5), color=colrs[3]) %>%
  gf_dist('beta', params = c(shape1=0.5, shape2=0.5), color=colrs[4]) %>%
  gf_labs(x='Possible Values of Variable', y='Probability Density') %>%
  gf_lims(y=c(0,2.5))
```

```
## Warning: Removed 796 rows containing missing values (geom_path).
```

```
## Warning: Removed 436 rows containing missing values (geom_path).
```



5.4.3.0.5 PDF or PMF

The PDF is:

$$f(x) = \frac{x^{(\alpha-1)}(1-x)^{(\beta-1)}}{B(\alpha,\beta)}$$

Where B is the Beta function (again, feel free to look up the definition if you are interested).

5.4.3.0.6 Examples

Beta distributions could be used for any variable that takes on values between 0-1, for example, baseball players' batting averages, or test scores (as proportions).

5.5 Examples of Discrete Distributions

5.5.1 Binomial

5.5.1.0.1 Type

Discrete

5.5.1.0.2 Support

You can think about the support of this distribution two ways. Technically, the support is $k = 0, 1, 2, 3, \ldots$: 0 and positive integers, interpreted as the number of "successes" in n binomial trials. Binomial trials are independent observations of a process that has two possible outcomes, "success" or "failure", with set probabilities of each occurring. (And probabilities of success and failure must sum to 1.)

So, for each individual binomial trial, the possible outcomes are 0 and 1, often interpreted as TRUE and FALSE or "success" and "failure".

For our purposes, this distribution will be useful for modelling response variables that are categorical with two possible values (and the n trials will be the n rows in our dataset).

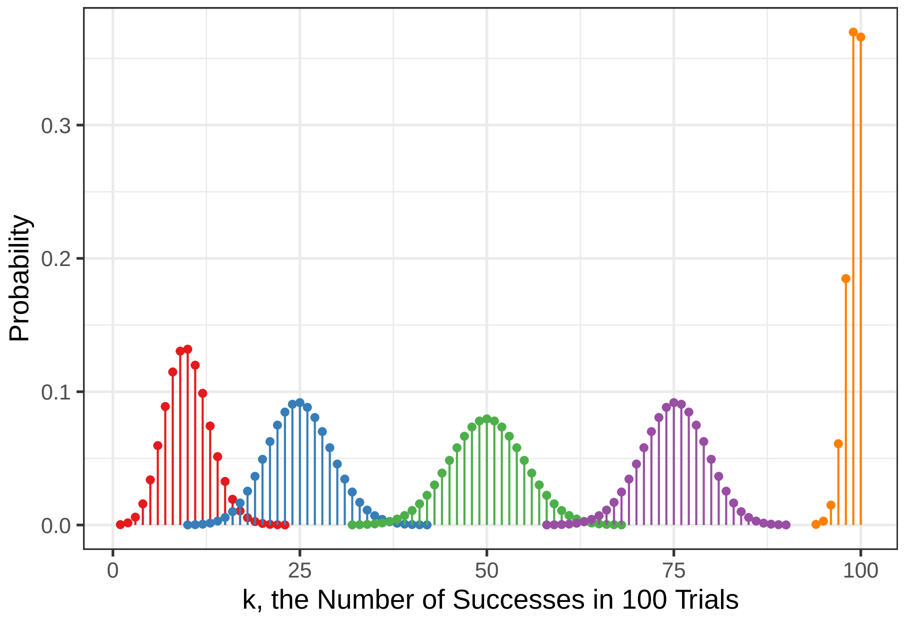
5.5.1.0.3 Parameters

Parameters are n, the number of independent trials, and p, the probability of success in each trial.

5.5.1.0.4 Shapes

The figure below shows the shape of binomial distributions with fixed $n = 100$ and varying p:

```r
gf_dist('binom', params = c(size=100, prob=0.1), color=colrs[1]) %>%
  gf_dist('binom', params = c(size=100, prob=0.25), color=colrs[2]) %>%
  gf_dist('binom', params = c(size=100, prob=0.5), color=colrs[3]) %>%
  gf_dist('binom', params = c(size=100, prob=0.75), color=colrs[4]) %>%
  gf_dist('binom', params = c(size=100, prob=0.99), color=colrs[5]) %>%
  gf_labs(x='k, the Number of Successes in 100 Trials', y='Probability')
```



The p used were 0.1, 0.25, 0.5, 0.75, and 0.99. Can you tell which is which?

5.5.1.0.5 PDF or PMF

The PMF for the binomial distribution is:

$$P(X = k|n,p) = \binom{n}{k} p^k (1-p)^{n-k}$$

Where k is the number of successes observed in n trials (you can think of k as our "x-axis variable" for this PMF).

5.5.1.0.6 Examples

We might use this distribution to model any categorical variable with two possible values, like Age (if possible values are "adult" and "child") or health status ("has disease" or "does not have disease"). We'll think of each observation in the dataset as one of the n indpendent trials, with one of two possible outcomes for each trial.

5.5.2 Poisson

5.5.2.0.1 Type

Discrete

5.5.2.0.2 Support

The support is 0 and positive integers (i.e., this distribution works well for count data).

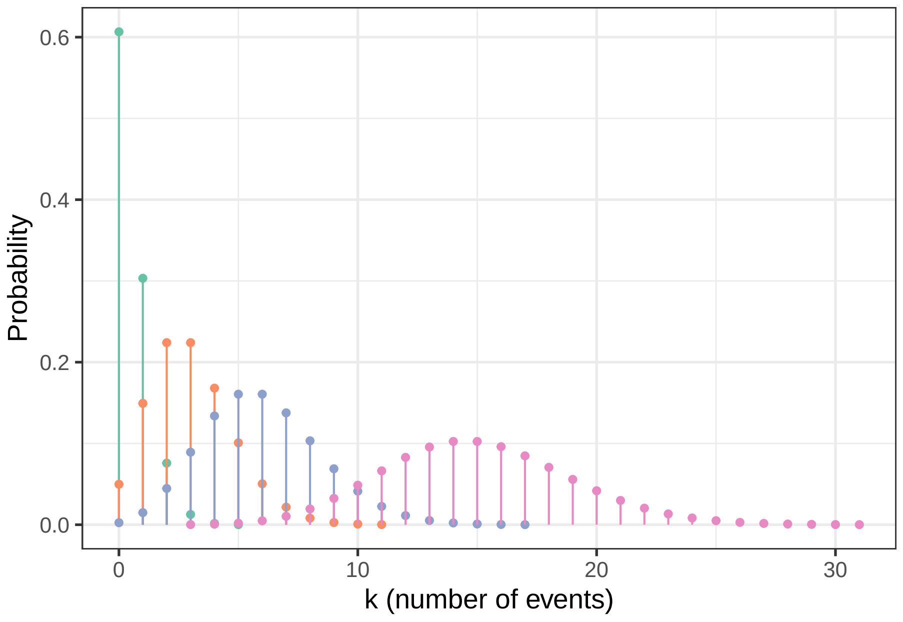
5.5.2.0.3 Parameters

The Poisson distribution has one parameter, λ (the event rate per unit time) which must be greater than 0.

5.5.2.0.4 Shapes

The distribution can take on unimodal shapes with varying amounts of right skew (from none, to an exponential shape).

```
colrs <- RColorBrewer::brewer.pal(8, 'Set2')
gf_dist('pois', params=c(lambda=0.5), color=colrs[1]) %>%
  gf_dist('pois', params=c(lambda=3), color=colrs[2]) %>%
  gf_dist('pois', params=c(lambda=6), color=colrs[3]) %>%
  gf_dist('pois', params=c(lambda=15), color=colrs[4]) %>%
  gf_labs(x='k (number of events)', y='Probability')
```



5.5.2.0.5 PDF or PMF

The Poisson PMF is:

$$P(X = k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

5.5.2.0.6 Examples

The Poisson distribution might be used to model any response variable that is comprised of counts, for example, the number of birds sighted in a bird survey, or the number of people admitted to an emergency room each hour.

5.5.3 Negative Binomial

There are two versions or "types" of this distribution, cleverly known as NB1 (type 1) and NB2 (type 2). NB1 has "constant overdispersion" – the variance of the distribution is greater than the mean according to a constant ratio. NB2 has "variable overdispersion" – the variance is a quadratic function of the mean. The NB2 is the one that corresponds directly to conceptualization in terms of binomial trials (with the PMF giving the probability of observing y failures before the rth success). Hardin and Hilbe 2007 describe the negative binomial this way: "Instead of counts entering uniformly, we see counts entering with a specific gamma-distributed shape."

5.5.3.0.1 Type

Discrete

5.5.3.0.2 Support

The support is 0 and positive integers (i.e., this distribution works well for count data). It also has a derivation in terms of binomial trials, but in our regression models, we will only use it with count data.

5.5.3.0.3 Parameters

A common parameterization of the negative binomial (online and in actuarial science) has parameters p, the probability of success on each binomial trial, and r, the number of failures observed. The PMF then gives the probability of observing k failures before the rth success in a series of Bernoulli trials.

Here, we will use an alternate parameterization. The other common way to parameterize and derive the NB is as a Poisson-gamma mixture – a modified version of a Poisson distribution. In this scheme, the parameters of the distribution are μ and α.

5.5.3.0.4 Shapes

These distributions can take on unimodal shapes with varying amounts of right skew. In NB2 (type 2) distributions the variance (spread) is larger relative to the mean.

NB1:

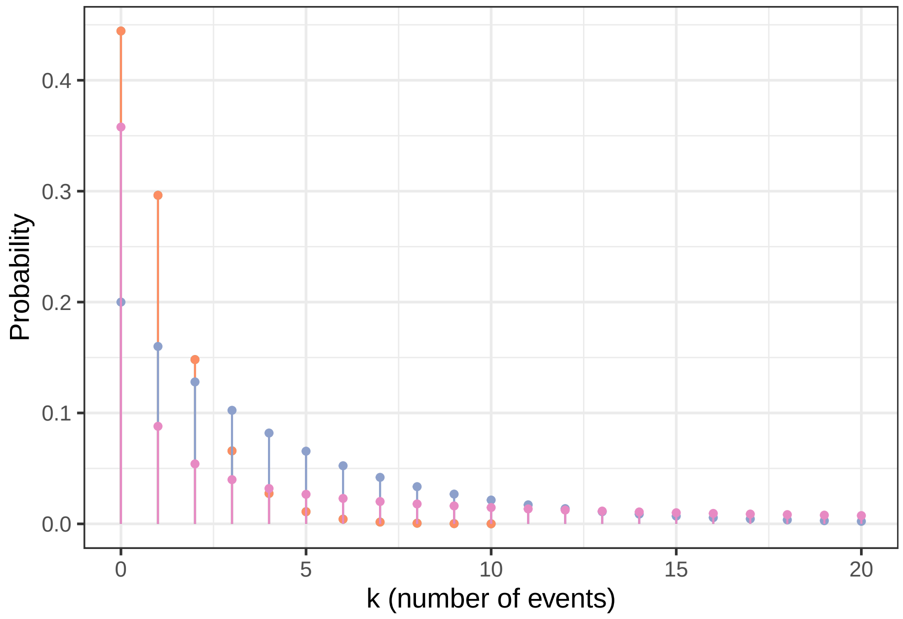
```
require(gamlss.dist)
colrs <- RColorBrewer::brewer.pal(8, 'Set2')
gf_dist('NBI', params=c(mu=1, sigma=0.5), color=colrs[1]) %>%
  gf_dist('NBI', params=c(mu=1, sigma=0.5), color=colrs[2]) %>%
  gf_dist('NBI', params=c(mu=4, sigma=1), color=colrs[3]) %>%
  gf_dist('NBI', params=c(mu=15, sigma=4), color=colrs[4]) %>%
  gf_lims(x=c(0,20)) %>%
  gf_labs(x='k (number of events)', y='Probability')
```

```
## Warning: Removed 15 rows containing missing values (geom_segment).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 177 rows containing missing values (geom_segment).

## Warning: Removed 177 rows containing missing values (geom_point).
```



NB2:

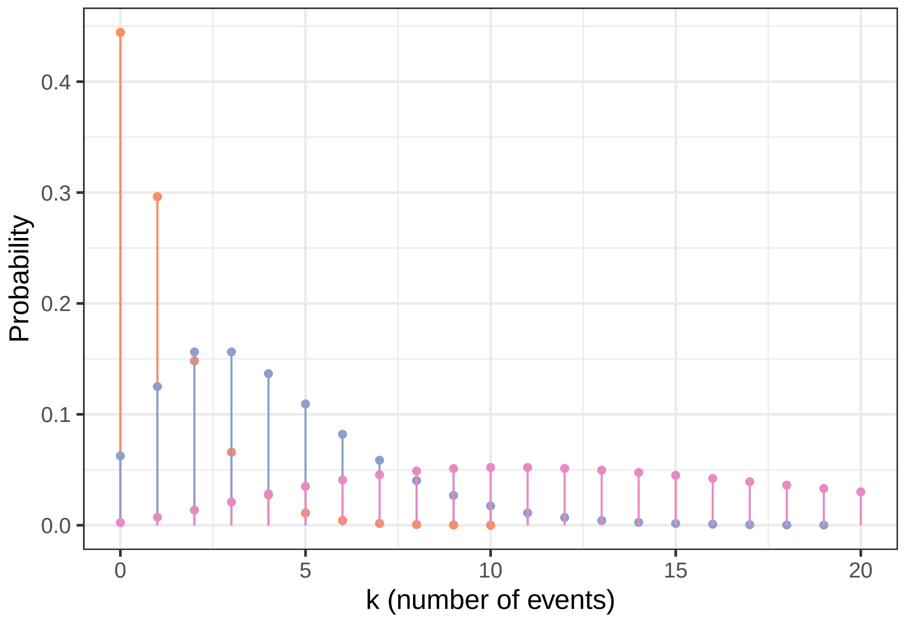
```
colrs <- RColorBrewer::brewer.pal(8, 'Set2')
gf_dist('NBII', params=c(mu=1, sigma=0.5), color=colrs[1]) %>%
  gf_dist('NBII', params=c(mu=1, sigma=0.5), color=colrs[2]) %>%
  gf_dist('NBII', params=c(mu=4, sigma=1), color=colrs[3]) %>%
  gf_dist('NBII', params=c(mu=15, sigma=4), color=colrs[4]) %>%
  gf_lims(x=c(0,20)) %>%
  gf_labs(x='k (number of events)', y='Probability')
```

```
## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 39 rows containing missing values (geom_segment).

## Warning: Removed 39 rows containing missing values (geom_point).
```



5.5.3.0.5 PDF or PMF

Details of the parameterizations and likelihood and fitting of NB1 and NB2 distributions can be found in Hardin and Hilbe 2007, if you are interested.

The PMF for the NB1, where the variance is a constant multiple of the mean, is:

$$f(x|\mu,\alpha) = \frac{\Gamma(x+\mu)}{\Gamma(\mu)\Gamma(x+1)}(\frac{1}{1+\alpha})^{\mu}(\frac{\alpha}{1+\alpha})^{x}$$

Where Γ is a Gamma function. Note that if $\alpha = 0$ this becomes a Poisson distribution, so the Poisson is a special case of the NB1.

The PMF for the NB2, where the variance is a quadratic function of the mean, is:

$$f(x|\mu,\alpha) = \frac{\Gamma(x+\frac{1}{\alpha})}{\Gamma(\frac{1}{\alpha})\Gamma(x+1)}(\frac{1}{1+\alpha\mu})^{\frac{1}{\alpha}}(1-\frac{1}{1+\alpha\mu})^{x}$$

5.5.3.0.6 Examples

NB distributions are good models for overdispersed count data, where (in the regression context) the residual variance is not equal to the expected (predicted) value. (Note that if you are reading this before learning about regression models for count data, you may not understand this sentence yet…don't worry, it will make sense when you return later!) Some examples might include sightings data on numbers of animals seen on wildlife surveys, or the number of items bought per order at an online retailer.

5.5.4 A Mixture Distribution: Tweedie

The Tweedie family of distributions is a very large one - depending on the values of the different parameters, the PMF/PDF can be written in many different ways, and it can take on many different shapes. The description below is a simplified one, geared toward the types of Tweedie distributions we are likely to try to use in regression models in this course – mainly the "compound Poisson-gamma" type.

Some extra resources for which you will not be held responsible in this course:

- You can find an accessible description and example of this kind of distribution at: http://www.notenoughthoughts.net/posts/modeling-activity.html.
- The following site may also be useful in regard to using the Tweedie in regression models: http://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_genmod_details28.htm

5.5.4.0.1 Type

These distributions are *both* continuous and discrete - a kind of mix of a Poisson distribution and gamma distribution(s).

5.5.4.0.2 Support

The support is non-negative real numbers (greater than or equal to 0).

5.5.4.0.3 Parameters

(Note: there are multiple different ways to parameterize these distributions.) We will use:

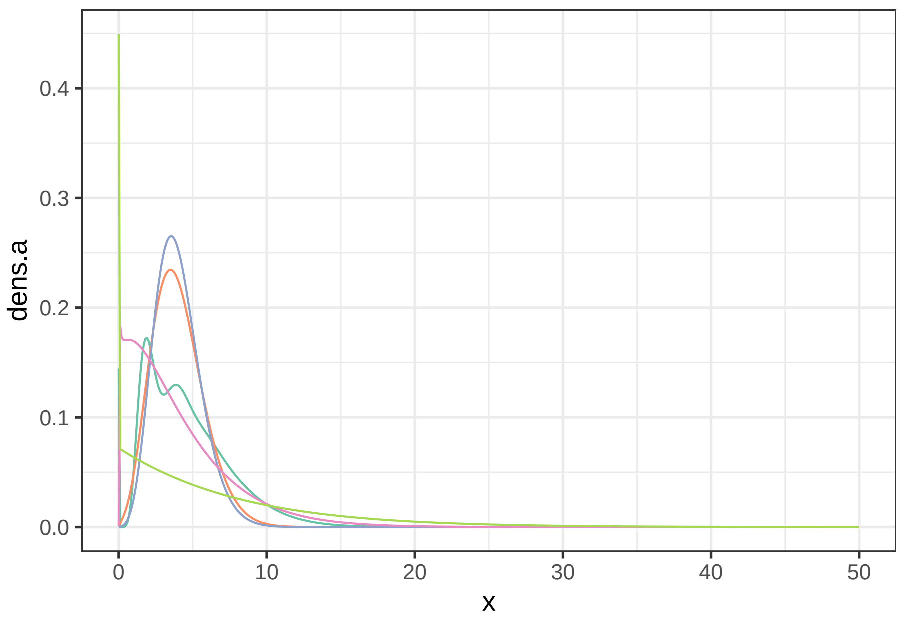
- p, the index or power parameter, which can be 0 (resulting in a normal distribution), 1 (resulting in a Poisson distribution), $1 < p < 2$ **(a compound Poisson-gamma distribution – what we will mainly use)**, 2 (a gamma distribution), 3 (an inverse Gaussian distribution), < 3 (a positive stable distribution), or ∞ (an extreme positive stable distribuiton). For the case where $1 < p < 2$, and the distribution is a compound of a Poisson and a gamma, then $p = \frac{k+2}{k+1}$ where k is the parameter of the gamma distribution. When $1 < p < 2$, p closer to 1 means thant the Poisson distribution (the mass at 0) gets more "weight" in the compound distribution, and values of p closer to 2 mean that the gamma distribution gets more "weight."
- μ. For the case where $1 < p < 2$, and the distribution is a compound of a Poisson and a gamma, then $\mu = \lambda k \theta$ where λ is the parameter of the Poisson distribution and k and θ are the parameters of the gamma.
- ϕ For the case where $1 < p < 2$, and the distribution is a compound of a Poisson and a gamma, then $\phi = \frac{\lambda^{(1-p)}(k\theta)^{(2-p)}}{2-p}$ where λ is the parameter of the Poisson distribution and k and θ are the parameters of the gamma.

5.5.4.0.4 Shapes

A compound Poisson-gamma Tweedie distribution can take on varying shapes; the main characteristic of interest for us is that it can have a mass at 0, then a unimodal or multimodal distribution with a long right tail (lots of right skew).

```
require(tweedie)
tex <- data.frame(x=seq(from=0, by=0.1, to=50)) %>%
  mutate(dens.a = dtweedie(x, xi = 1.1, mu=4, phi=2)) %>%
  mutate(dens.b = dtweedie(x, xi = 1.3, mu=4, phi=0.5)) %>%
  mutate(dens.c = dtweedie(x, xi = 1.5, mu=4, phi=0.3)) %>%
  mutate(dens.d = dtweedie(x, xi = 1.8, mu=4, phi=1)) %>%
  mutate(dens.e = dtweedie(x, xi = 1.5, mu=4, phi=5))


gf_line(dens.a ~ x, data=tex, color=colrs[1]) %>%
  gf_line(dens.b ~x, data=tex, color=colrs[2]) %>%
  gf_line(dens.c ~x, data=tex, color=colrs[3]) %>%
  gf_line(dens.d ~x, data=tex, color=colrs[4]) %>%
  gf_line(dens.e ~x, data=tex, color=colrs[5])
```



5.5.4.0.5 PDF or PMF

A Tweedie distribution with $p > 1$ has the form:

$$f(X|\mu, \phi, p) = a(x, \phi)e^{\frac{1}{\phi}(\frac{x\mu^{(1-p)}}{(1-p)} - \kappa(\mu,p))}$$

where $\kappa(\mu, p) = \frac{\mu^{(2-p)}}{(2-p)}$ if $p \neq 2$, and if $p = 2$, $\kappa(\mu, p) = log(\mu)$; but $a(x, \phi)$ is a function that does not have an analytical expression. This expression is from SAS documentation at https://support.sas.com/rnd/app/stat/examples/tweedie/tweedie.pdf.

Alternately (and more simply(?)), a Tweedie distribution with $1 < p < 2$ is a compound of a Poisson distribution with parameter λ and a gamma distribution with parameters k and θ. For example:

“Suppose that airplanes arrive at an airport following a Poisson process, and the number of passengers in each airplane follows a certain gamma distribution. Then, the number of passengers arriving at the airport follows a compound Poisson gamma process

$$Y = \sum_{i=1}^{N} D_i$$

where N is the Poisson process that the airplanes follow, and D_i is the gamma distribution that the passengers follow." (Thanks to D. Mao, http://math.uchicago.edu/~may/REU2013/REUPapers/Mao.pdf for this example.)

5.5.4.0.6 Examples

The Tweedie distributions may be useful for "zero-inflated" data, where there is a class of observations for which the observed value of the variable is always zero, and another class for which the variable takes on positive continuous values. For example, this might model the number of birds present per unit area (when the study area includes places of unsuitable habitat where none are ever found), or perhaps the quantity of alcohol consumed per week by different people (some of whom may drink varying amounts, and others of whom may never drink at all).