```
  Course: Regression_Models
  Lesson: Count_Outcomes

- Class: text
  Output: "Count Outcomes. (Slides for this and other Data Science courses may be found at github https://github.com/DataScienceSpecialization/courses. If you care to use them, they must be downloaded as a zip file and viewed locally. This lesson corresponds to Regression_Models/03_03_countOutcomes.)"

- Class: text
  Output: "Many data take the form of counts. These might be calls to a call center, number of flu cases in an area, or number of cars that cross a bridge. Data may also be in the form of rates, e.g., percent of children passing a test. In this lesson we will use Poisson regression to analyze daily visits to a web site as the web site's popularity grows, and to analyze the percent of visits which are due to references from a different site."

- Class: text
  Output: "Visits to a web site tend to occur independently, one at a time, at a certain average rate. The Poisson distribution describes random processes of this type. A Poisson process is characterized by a single parameter, the expected rate of occurrence, which is usually called lambda. In our case, lambda will be expected visits per day. Of course, as the web site becomes more popular, lambda will grow. In other words, our lambda will depend on time. We will use Poisson regression to model this dependence."

- Class: cmd_question
  Output: "Somwhat remarkably, the variance of a Poisson process has the same value as its mean, lambda. You can quickly illustrate this by generating, say, n=1000 samples from a Poisson process using R's rpois(n, lambda) and calculating the sample variance. For example, type var(rpois(1000, 50)). The sample variance won't be exactly equal to the theoretical value, of course, but it will be fairly close."
  CorrectAnswer: 'var(rpois(1000, 50))'
  AnswerTests: expr_uses_func('var');expr_uses_func('rpois')
  Hint: Try typing var(rpois(1000, 50)).

- Class: mult_question
  Output: "A famous theorem implies that properly normalized sums of independent, identically distributed random variables will tend to become normally distributed as the number of samples grows large. What is that theorem?"
  AnswerChoices: The Central Limit Theorem;The Gauss-Markov BLUE Theorem;The Pythagorean Theorem
  CorrectAnswer: The Central Limit Theorem
  AnswerTests: omnitest(correctVal= 'The Central Limit Theorem')
  Hint: It deals with limits.

- Class: figure
  Output: "The counts generated by a Poisson process are, strictly speaking, slightly different than the normalized sums of the Central Limit Theorem. However, the counts in a given period of time will represent sums of larger numbers of terms as lambda increases. In fact, it can be formally shown that for large lambda a Poisson distribution is well approximated by a normal. The figure illustrates this effect. It shows progression from a sparse, asymetric, Poisson probability mass function on the left, to a dense, bell-shaped curve on the right as lambda varies from 2 to 100."
  Figure: poisson_2_normal.R
  FigureType: new

- Class: figure
  Output: "In a Poisson regression, the log of lambda is assumed to be a linear function of the predictors. Since we will try to model the growth of visits to a web site, the log of lambda will be a linear function of the date: log(lambda) = b0 + b1*date. This implies that the average number of hits per day, lambda, is exponential in the date: lambda = exp(b0)*exp(b1)^date. Exponential growth is also suggested by the smooth, black curve drawn though the data. Thus exp(b1) would represent the percentage by which visits grow per day."
  Figure: hits.R
  FigureType: new

- Class: video
```

```
  Output: "If you are connected to the internet right now, would you care to visit the
  Leek Group website?"
  VideoLink: 'http://biostat.jhsph.edu/~jleek/'

- Class: cmd_question
  Output: "Our data is in a data frame named hits. Use View(hits), head(hits), or
  tail(hits) to examine the data now."
  CorrectAnswer: 'View(hits)'
  AnswerTests: ANY_of_exprs('View(hits)', 'head(hits)', 'tail(hits)')
  Hint: "Type View(hits), head(hits), or tail(hits). Any of these will do."

- Class: text
  Output: "There are three columns of data labeled date, visits, and simplystats
  respectively. The simplystats column records the number of visits which are due to
  references from another site, the Simply Statistics blog. We'll come back to that
  column later. For now, we are interested in the date and visits columns. The date
  will be our predictor."

- Class: cmd_question
  Output: "Our dates are represented in terms of R's class, Date. Verify this by typing
  class(hits[,'date']), or something equivalent."
  CorrectAnswer: class(hits[,'date'])
  AnswerTests: ANY_of_exprs("class(hits[,'date'])", 'class(hits[,"date"])',
  'class(hits[,1])', 'class(hits$date)')
  Hint: Type class(hits[,'date']), or something equivalent.

- Class: cmd_question
  Output: "R's Date class represents dates as days since or prior to January 1, 1970.
  They are essentially numbers, and to some extent can be treated as such. Dates can,
  for example, be added or subtracted, or easily coverted to numbers. Type
  as.integer(head(hits[,'date'])) to see what I mean."
  CorrectAnswer: class(hits[,'date'])
  AnswerTests: ANY_of_exprs("as.integer(head(hits[,'date']))",
  'as.integer(head(hits[,"date"]))', 'as.integer(head(hits[,1]))',
  'as.integer(head(hits$date))')
  Hint: Type as.integer(head(hits[,'date']), or something equivalent.

- Class: cmd_question
  Output: "The arithmetic properties of Dates allow us to use them as predictors. We'll
  use Poisson regression to predict log(lambda) as a linear function of date in a way
  which maximizes the likelihood of the counts we actually see. Our formula will be
  visits ~ date. Since our outcomes (visits) are counts, our family will be 'poisson',
  and our third argument will be the data, hits. Create such a model and store it in a
  variable called mdl using the following expression or something equivalent, mdl <-
  glm(visits ~ date, poisson, hits)."
  CorrectAnswer: mdl <- glm(visits ~ date, poisson, hits)
  AnswerTests: creates_glm_model('mdl <- glm(visits ~ date, poisson, hits)')
  Hint: Type mdl <- glm(visits ~ date, poisson, hits) or something equivalent.

- Class: figure
  Output: "The figure suggests that our Poisson regression fits the data very well. The
  black line is the estimated lambda, or mean number of visits per day. We see that
  mean visits per day increased from around 5 in early 2011 to around 10 by 2012, and
  to around 20 by late 2013. It is approximately doubling every year."
  Figure: model_1.R
  FigureType: new

- Class: cmd_question
  Output: "Type summary(mdl) to examine the estimated coefficients and their
  significance."
  CorrectAnswer: summary(mdl)
  AnswerTests: omnitest('summary(mdl)')
  Hint: Just type summary(mdl)

- Class: text
  Output: "Both coefficients are significant, being far more than two standard errors
  from zero. The Residual deviance is also very significantly less than the Null,
  indicating a strong effect. (Recall that the difference between Null and Residual
  deviance is approximately chi-square with 1 degree of freedom.) The Intercept
```

```
    coefficient, b0, just represents log average hits on R's Date 0, namely January 1,
    1970. We will ignore it and focus on the coefficient of date, b1, since exp(b1) will
    estimate the percentage at which average visits increase per day of the site's life."

  - Class: cmd_question
    Output: "Get the 95% confidence interval for exp(b1) by exponentiating confint(mdl,
    'date')"
    CorrectAnswer: exp(confint(mdl, 'date'))
    AnswerTests: ANY_of_exprs("exp(confint(mdl, 'date'))", 'exp(confint(mdl, "date"))',
    "exp(confint(mdl,2))")
    Hint: Just type exp(confint(mdl, 'date')) or exp(confint(mdl, 2)).

  - Class: text
    Output: "Visits are estimated to increase by a factor of between 1.002192 and
    1.002399 per day. That is, between 0.2192% and 0.2399% per day. This actually
    represents more than a doubling every year."

  - Class: figure
    Output: "Our model looks like a pretty good description of the data, but no model is
    perfect and we can often learn about a data generation process by looking for a
    model's shortcomings. As shown in the figure, one thing about our model is 'zero
    inflation' in the first two weeks of January 2011, before the site had any visits.
    The model systematically overestimates the number of visits during this time. A less
    obvious thing is that the standard deviation of the data may be increasing with
    lambda faster than a Poisson model allows. This possibility can be seen in the
    rightmost plot by visually comparing the spread of green dots with the standard
    deviation predicted by the model (black dashes.) Also, there are four or five bursts
    of popularity during which the number of visits far exceeds two standard deviations
    over average. Perhaps these are due to mentions on another site."
    Figure: shortcomings.R
    FigureType: new

  - Class: figure
    Output: "It seems that at least some of them are. The simplystats column of our data
    records the number of visits to the Leek Group site which come from the related site,
    Simply Statistics. (I.e., visits due to clicks on a link to the Leek Group which
    appeared in a Simply Statisics post.) "
    Figure: bursts.R
    FigureType: new

  - Class: cmd_question
    Output: "In the figure, the maximum number of visits occurred in late 2012. Visits
    from the Simply Statistics blog were also at their maximum that day. To find the
    exact date we can use which.max(hits[,'visits']). Do this now."
    CorrectAnswer: which.max(hits[,'visits'])
    AnswerTests: omnitest("which.max(hits[,'visits'])", 704)
    Hint: Type which.max(hits[,'visits']) or something equivalent.

  - Class: cmd_question
    Output: "The maximum number of visits is recorded in row 704 of our data frame. Print
    that row by typing hits[704,]."
    CorrectAnswer: hits[704,]
    AnswerTests: omnitest(correctExpr='hits[704,]')
    Hint: Just type hits[704,].

  - Class: cmd_question
    Output: "The maximum number of visits, 94, occurred on December 4, 2012, of which 64
    came from the Simply Statistics blog. We might consider the 64 visits to be a special
    event, over and above normal. Can the difference, 94-64=30 visits, be attributed to
    normal traffic as estimated by our model? To check, we will need the value of lambda
    on December 4, 2012. This will be entry 704 of the fitted.values element of our
    model. Extract mdl$fitted.values[704] and store it in a variable named lambda."
    CorrectAnswer: lambda <- mdl$fitted.values[704]
    AnswerTests: omnitest(correctExpr='lambda <-
    mdl$fitted.values[704]');expr_creates_var('lambda')
    Hint: Just type lambda <- mdl$fitted.values[704].

  - Class: cmd_question
    Output: "The number of visits explained by our model on December 4, 2012 are those of
```

```
    a Poisson random variable with mean lambda. We can find the 95th percentile of this
    distribution using qpois(.95, lambda). Try this now."
    CorrectAnswer: qpois(.95, lambda)
    AnswerTests: ANY_of_exprs('qpois(.95, lambda)', 'qpois(0.95, lambda)')
    Hint: Type qpois(.95, lambda) or qpois(0.95, lambda).

- Class: text
    Output: "So, 95% of the time we would see 33 or fewer visits, hence 30 visits would
    not be rare according to our model. It would seem that on December 4, 2012, the very
    high number of visits was due to references from Simply Statistics. To gauge the
    importance of references from Simply Statistics we may wish to model the proportion
    of traffic such references represent. Doing so will also illustrate the use of glm's
    parameter, offset, to model frequencies and proportions."

- Class: text
    Output: "A Poisson process generates counts, and counts are whole numbers, 0, 1, 2,
    3, etc. A proportion is a fraction. So how can a Poisson process model a proportion?
    The trick is to include the denominator of the fraction, or more precisely its log,
    as an offset. Recall that in our data set, 'simplystats' is the visits from Simply
    Statistics, and 'visits' is the total number of visits. We would like to model the
    fraction simplystats/visits, but to avoid division by zero we'll actually use
    simplystats/(visits+1). A Poisson model assumes that log(lambda) is a linear
    combination of predictors. Suppose we assume that log(lambda) = log(visits+1) + b0 +
    b1*date. In other words, if we insist that the coefficient of log(visits+1) be equal
    to 1, we are predicting the log of mean visits from Simply Statistics as a proportion
    of total visits: log(lambda/(visits+1)) = b0 + b1*date."

- Class: cmd_question
    Output: "glm's parameter, offset, has precisely this effect. It fixes the coefficient
    of the offset to 1. To create a model for the proportion of visits from Simply
    Statistics, we let offset=log(visits+1). Create such a Poisson model now and store it
    as a variable called mdl2."
    CorrectAnswer: mdl2 <- glm(simplystats ~ date, poisson, hits, offset=log(visits+1))
    AnswerTests: creates_glm_model('mdl2 <- glm(simplystats ~ date, poisson, hits,
    offset=log(visits+1))')
    Hint: "Enter mdl2 <- glm(formula = simplystats ~ date, family = poisson, data = hits,
    offset = log(visits + 1)), or something equivalent."

- Class: cmd_question
    Output: "Although summary(mdl2) will show that the estimated coefficients are
    significantly different than zero, the model is actually not impressive. We can
    illustrate why by looking at December 4, 2012, once again. On that day there were 64
    actual visits from Simply Statistics. However, according to mdl2, 64 visits would be
    extremely unlikely. You can verify this weakness in the model by finding mdl2's 95th
    percentile for that day. Recalling that December 4, 2012 was sample 704, find
    qpois(.95, mdl2$fitted.values[704])."
    CorrectAnswer: qpois(.95, mdl2$fitted.values[704])
    AnswerTests: ANY_of_exprs('qpois(.95, mdl2$fitted.values[704])', 'qpois(0.95,
    mdl2$fitted.values[704])')
    Hint: Just type qpois(.95, mdl2$fitted.values[704]).

- Class: mult_question
    Output: "A Poisson distribution with lambda=1000 will be well approximated by a
    normal distribution. What will be the variance of that normal distribution?"
    AnswerChoices: lambda;lambda squared;the square root of lambda.
    CorrectAnswer: lambda
    AnswerTests: omnitest(correctVal= 'lambda')
    Hint: The mean and the variance of a Poisson distribution are equal.

- Class: mult_question
    Output: "When modeling count outcomes as a Poisson process, what is modeled as a
    linear combination of the predictors?"
    AnswerChoices: The log of the mean;The mean;The counts
    CorrectAnswer: The log of the mean
    AnswerTests: omnitest(correctVal= 'The log of the mean')
    Hint: Count outcomes and their means are never negative, but linear combinations of
    predictors may be.

- Class: mult_question
```

```
    Output: "What parameter of the glm function allows you to include a predictor whose
    coefficient is fixed to the value 1?"
    AnswerChoices: offset;data;b0;family;formula
    CorrectAnswer: offset
    AnswerTests: omnitest(correctVal= 'offset')
    Hint: We used this parameter to include log(visits+1) when computing mdl2.

  - Class: text
    Output: "That completes the Poisson GLM example. Thanks for sticking with it. I hope
    we've made it count."

  - Class: mult_question
    Output: "Would you like to receive credit for completing this course on
      Coursera.org?"
    CorrectAnswer: NULL
    AnswerChoices: Yes;No
    AnswerTests: coursera_on_demand()
    Hint: ""

```