**Generalized Linear Mixed Models in SPSS**

The *general* in *general linear (mixed) models* refers to the usual continuous numeric dependent variables we've been analyzing so far, whereas *generalized* refers to these and any other kind of dependent variable. The other kinds we need worry about are binary variables, counts, and proportions. Binary variables are usually coded as 0 or 1 to indicate whether something occurred or was present; counts are non-negative integers representing how many things occurred or were present; proportions have values between 0 and 1, indicating the count of occurrences or presences expressed as a fraction of the total possible. If you apply the usual linear models to these variables, you can get into serious trouble: non-uniformity of errors and effects, and impossible predicted values and/or confidence limits thereof (negative counts, proportions outside the range 0 to 1). The solution is to model a transformation of the dependent variable that can go from -∞ to ∞, and you trust that the resulting effects on the transformed variable are realistic; they are likely to be more realistic than effects on the untransformed variable. Binary variables can be submitted for analysis as 0s and 1s, but they are analyzed either as counts or proportions, so in the first instance we need only two transformations and therefore two kinds of model. There are also two extensions of one of the models.

So, first an explanation of the models and transformations, then some remarks about procedures for dealing with repeated measurements with generalized models, then step-by-step instructions for two analyses with a dataset aimed at investigating the relationship between fitness tests and subsequent values of performance indicators in a short season of rugby union. The data were kindly provided by Josh Darrall-Jones from one of his PhD studies at Leeds Beckett University. The performance indicator is the count of effective rucks of each player in each of up to six games, which we will analyze first as a count using *Poisson regression*, then as a proportion of the rucks attempted by the player using *logistic regression*.

*Poisson Regression*

For counts the transformation is simply the (natural) log: a count goes from 0 to ∞, so log(count) goes from -∞ to ∞. You don't take logs of the counts, though, because if there are any zeros, you would get -∞, which becomes a missing value. What actually happens is that the model predicts the log of the count, using the observed counts, which can include zeros. The model also takes into account the fact that the sampling variation of a count depends on the expected (predicted or true) value of the count. For example, if you expect a count of 25, you will observe counts like 19, 27, 24, 18, 31,… The scatter of the counts is called the sampling distribution, and for counts the distribution is known as a Poisson distribution. The variance of a Poisson distribution is the same as the mean, so the standard deviation in this example is √(variance) = √25 = 5. Hence those typical numbers shown.

A linear model for a count is called Poisson regression, but what's being predicted by the linear model is the log of the count. Hence effects, which are differences or changes in a linear model, become ratios (factors) after back transformation. For example, the effect of an intervention could be an increase in the count of effective rucks in games of football; the counts after the intervention divided by the counts before might be a ratio or factor of 1.18, or an 18% increase.

Often the variance of a count is greater than the mean count, usually because the events making up the count occur non-randomly, i.e., in clusters within or between subjects. (You have to really think hard to figure out why the variance is greater than the mean, when the counts come in clusters.) The distribution of counts is then said to be an over-dispersed Poisson. Occasionally counts can even be under-dispersed. It's important to allow for the dispersion to be different from that of a pure Poisson, because it makes a difference to the estimates and their confidence limits.

*Logistic Regression*

For the analysis of the proportion of something, the transformation has to turn the lowest possible proportion (0) into -∞ and the highest possible proportion (1) into ∞. The logistic or log-odds transformation does the job: the odds of a proportion p is p/(1-p), which goes from 0 to ∞, so the log of the odds goes from -∞ to ∞.

A linear model predicting the log of the odds is called logistic regression. As with Poisson regression, the analytic procedure does the transformation, so you can input observed proportions of zero or 1. You can also input occurrences coded as 0 or 1, and the procedure aggregates these into proportions. Effects in the linear model are differences or changes in the log of the odds, so these turn into odds ratios after back-transformation. Odds ratios are the same as proportion ratios, when proportions are small (<0.1 or <10%): p1/(1-p1) is approximately p1, p2/(1-p2) is approximately p2, so the odds ratio p2/(1-p2)/[p1/(1-p1)] is approximately p2/p1. Fine, when proportion ratios are what matter and what need to be evaluated for magnitude and uncertainty in the magnitude. However, when the proportions are not small, you have to use a reference proportion to convert the odds ratios back into proportion ratios so that you can make sense of the effect. For example, if the odds ratio for a treatment effect is 2.0, and the reference (control-group) proportion is 0.30 (30%), the reference odds is 0.3/0.7, so the odds in the experimental group is 2.0\*0.3/0.7 = 0.6/0.7 = 0.857, so the experimental proportion is given by p/(1-p) = 0.857, from which it follows that p = 1/(1+0.857) = 0.539, so (finally!) the effect expressed as a proportion ratio is 0.539/0.30 = 1.80, which is a bit less than the original odds ratio of 2.0. If the reference proportion had been 0.70, say, the resulting proportion ratio would be only 1.18, which is now considerably less than the odds ratio. Moral: don't interpret odds ratios as proportion ratios unless the proportions are small.

A special application of logistic regression is the analysis of win-lose outcomes in match-play sports. Here it's best to convert the odds ratio to a proportion difference rather than a proportion ratio, and you center the proportion difference on 0.5, or 50%. For example, if the odds ratio is 2.0, the two proportions are 0.41 and 0.59 (via high-school algebra), or 41% and 59%, so the proportion difference is 0.18 or 18%. In other words, the effect will take the athlete or the team from winning 4.1 matches in every 10 matches up to 5.9 matches in every 10 matches, an increase of 1.8 in every 10 matches. This way of representing the outcome–the change in the number of matches won against an otherwise nearly equal opponent–allows the magnitude to be interpreted directly using my scale for competitive athletes: one competition in every 10 is small, and 3, 5, 7 and 9 in every 10 are moderate, large, very large and extremely large, respectively. In this example the effect would be small, but of course you would have to interpret the confidence limits to decide whether the effect was clear.

Proportions are made up of counts, so they too can be over- or under-dispersed. Substantial over- and under-dispersion occurs when analyzing time spent in an activity as a proportion of the whole time, where the time in seconds or minutes is treated as a count. Obviously each second or minute of the activity is not an independent event! So again, it's important to allow dispersion to be different from that expected for the odds of independent events (which is given by the binomial distribution). The stats procedure takes care of the details.

*Log-hazards Regression*

When modeling the presence or absence of something with a value of 0 or 1, logistic regression works fine, but if the presence or absence is a time-dependent event, such as an injury, it's not appropriate to model the log of the odds. Instead you model the log of the probability that the event will occur in subjects who are as yet unaffected. To understand why, think: yeah, if the chance of getting injured in one day is whatever really low value in a reference group or condition, the chance of getting injured in another group will also be really small, but it could be higher or lower by some reasonable factor, such as 1.53 (a 53% greater chance of injury). This short-term or instantaneous risk ratio is called the incidence-rate ratio or the hazard ratio. Over a period of two days, the chance of getting injured will be twice as high in both groups, but the risk ratio will still be 1.53. But if you consider a much longer period, you end up with substantial proportions injured in the two groups, and eventually almost everyone would be injured in both groups. By this stage the risk ratio drops to nearly 1.0 and the odds ratio becomes huge, but the incidence-rate ratio for those who still aren't injured could still be 1.53. It would therefore be wrong to model the log of odds, because the odds ratio changes with the proportion, and therefore effects change with time.

The solution is to use a transformation that converts probabilities to hazards rather than odds. The transformation is called the complementary log-log (=log(‑log(1-p))), and the sampling distribution is still the binomial. The resulting linear model does not have an official name that I can find, so I've called it log-hazard regression. It's based on the assumption that the hazard ratio doesn't change with changing proportions, but of course the same kind of assumption applies to logistic regression (the odds ratio doesn't change with increasing proportions) and Poisson regression (the count ratio doesn't change with increasing counts). A more sophisticated version of this kind of analysis is available as something called proportional-hazards or Cox regression, in which the time to the event rather than the occurrence of the event is modeled.

*Cumulative Logistic Regression*

If you are modeling outcomes in a sport where draws (ties) are common, cumulative logistic regression is the way to go. In this model, the odds ratio for drawing vs losing is assumed to be the same as for winning vs drawing. The transformation is called the cumulative logit, and the sampling distribution is the multinomial. The odds ratio needs to be squared (something I may have forgotten to do in a recent publications) and then converted to a proportion difference for interpretation.

Likert scales with only a few levels are supposed to be analyzed with cumulative logistic regression, but interpreting the magnitude of the effects is a nightmare. I usually invoke the central-limit theorem and analyze the Likert scores with the usual mixed linear model. I have also justified this approach by comparing the t statistics provided by both methods, to show that they give the same answer. It is much easier to do magnitude-based inference with differences or changes in Likert scores than with cumulative odds ratios.

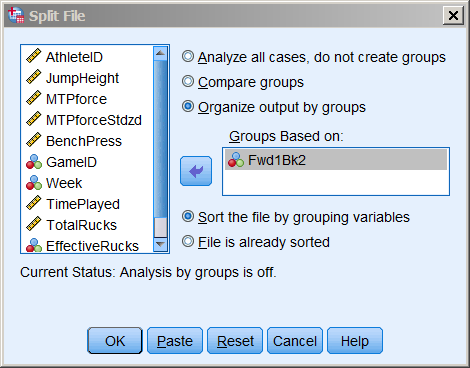
*Procedures for Generalized Linear Mixed Modeling*

There are two approaches to generalized linear mixed models in SPSS: generalized estimating equations (GEE) (via Analyze/Generalized Linear Models/Generalized Estimating Equations) and generalized linear models (via Analyze/Mixed Models/Generalized Linear…). GEEs were the first extension of mixed models to repeated measurements of counts and so on, and the approach is limited to random effects representing repeated measurements on subjects. The other approach is more advanced and allows other random effects (e.g., treating games as a randomly sampled variable as well subjects). I was hoping to introduce you to the other approach, because all my experience with generalized linear mixed models has been obtained with its equivalent in SAS, Proc Glimmix, which SAS introduced to supersede its GEE model, Proc Genmod. Unfortunately, this approach in SPSS Version 23 is practically unusable: it cannot work with split files, it cannot allow for over- or under-dispersion, and it doesn't allow use of /TEST. Even the GEE approach in SPSS does not support /TEST, so you will have to specify your models in ways that give the effects of interest directly from the fixed-effect coefficients. Another limitation of GEEs in SPSS and in SAS is that they don't provide a random-effect solution, not that I can find, anyway, so you can't use the analysis to rank subjects on their mean value of the dependent variable and thereby identify talent.

So, although I am providing this resource for doing generalized linear mixed modeling in SPSS, I am not that experienced with the GEE approach, so I am not completely confident with advice specific to this procedure. In any case, you should use it only to understand this kind of modeling and the kind of post-processing you have to do, but not to analyze data for a project. For that you should use Proc Glimmix SAS or the equivalent in some other package. I have yet to investigate the R package for such modeling, but I am not impressed with its functionality for the usual linear mixed models, so it's probably not there yet for the generalized models.

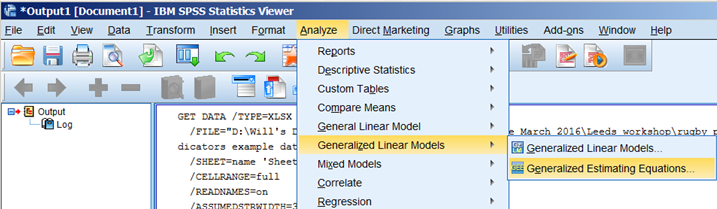
**Poisson Regression**

1. We're going to investigate the linear relationship between a fitness test (mid-thigh pull force) and a performance indictor in games (effective rucks) across six games. We'll also see if there's a linear trend in performance over the six games.
2. Import the Excel file "rugby performance indicators example data.xlsx". Inspect the data. I've deleted all but one fitness test, the isometric force in a mid-thigh pull (MTPforce), because it shows interesting relationships with the only two game performance indicators included, each player's effective rucks and total rucks. The variable MTPforceStdzd is MTPforce rescaled to give a mean of zero and a standard deviation of 1 for each of the two player positions, forwards and backs (the variable Fwd1Bk2). Note that the variable Week is identical to the variable Game, and WeekRescaled goes from -0.5 to 0.5. Note also TimePlayed, the time in minutes that the player was on the field in each game, and LnTimePlayed, the natural log of the fraction of the time expressed as a fraction of the usual game duration, 80 min. The exact duration of each game was sometimes >80 min, hence time played is >80 minutes sometimes, and LnTimePlayed is then >1.0. Check the Excel spreadsheet for the formulae for these derived variables.
3. We'll analyze forwards and backs separately, so choose **Data/Split File…**, click **Organize output by groups** and choose Fwd1Bk2:

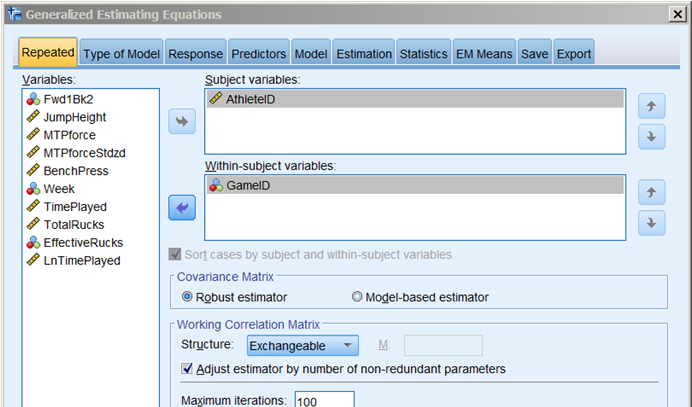


If we want to compare forwards and backs, we can use the spreadsheet "Combine/compare effects" at Sportscience, because these groups consist of different athletes, so they are effectively independent. They aren't *strictly* independent, but analyzing them in one analysis in SPSS would too difficult at this stage of your development, and keeping them separate requires less assumptions and less chance of making mistakes for effects within the groups. In any case, it is the effects within the groups that we're interested in. The comparison of the groups is more of an academic question. Forwards are obviously different from backs.

1. Choose **Analyze/Generalized Linear Models/Generalized Estimating Equations:**

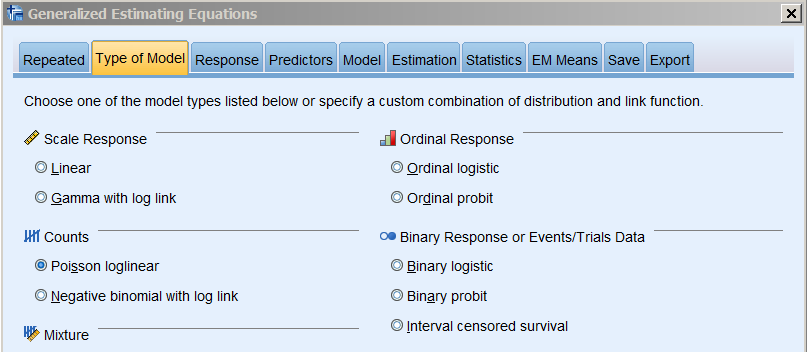


1. Choose AthleteID as the subject variable, GameID as the within-subject variable, and Exchangeable as the Working Correlation Matrix:

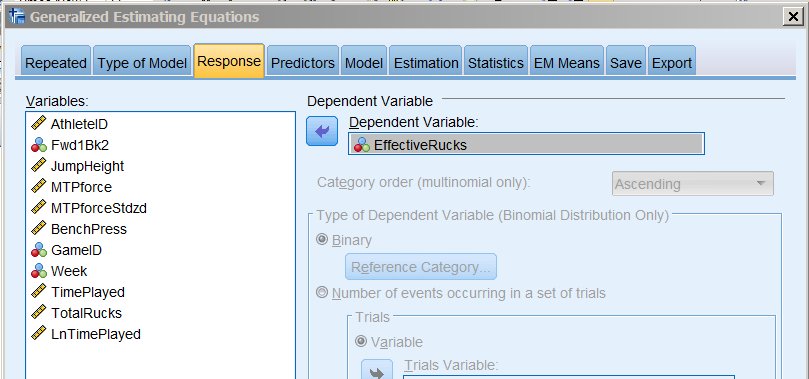


"Exchangeable" means that any value of the six GameIDs can be exchanged with any other value in terms of the correlation of the values of the dependent variable between pairs of games. Another even more mysterious name is compound symmetry. In plain language, we are assuming that the correlations between any games are all the same, apart from sampling variation. We won't bother to check this assumption, because I have found it to be reasonably accurate across many sports within a season. Basically athletes don't change much individually from one game to the next, although we can develop more complex models to allow for individual trends over a season and for individual changes between seasons.

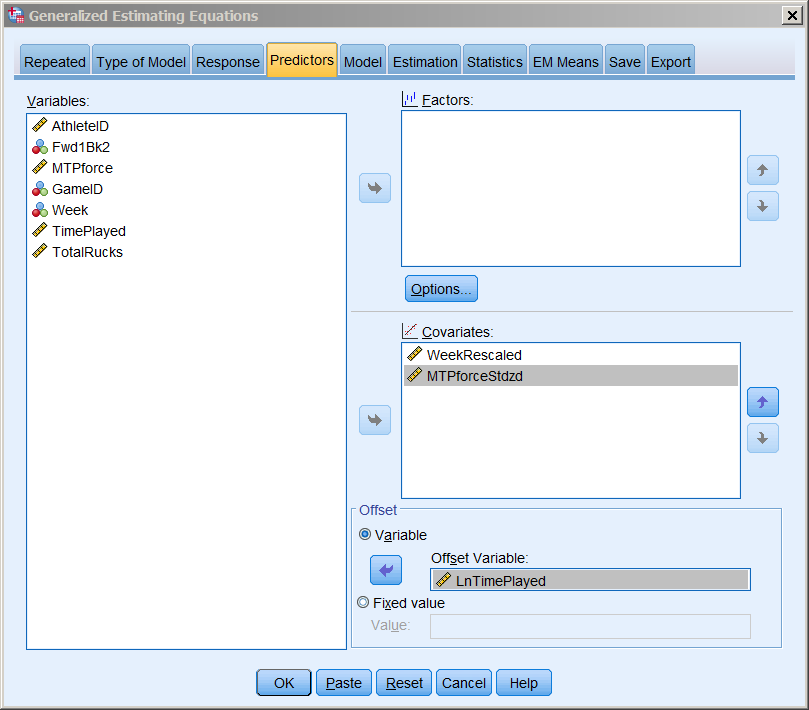
1. Click the **Type of Model** tab and select Poisson loglinear:



1. Click the **Response** tab and make EffectiveRucks the Dependent Variable:



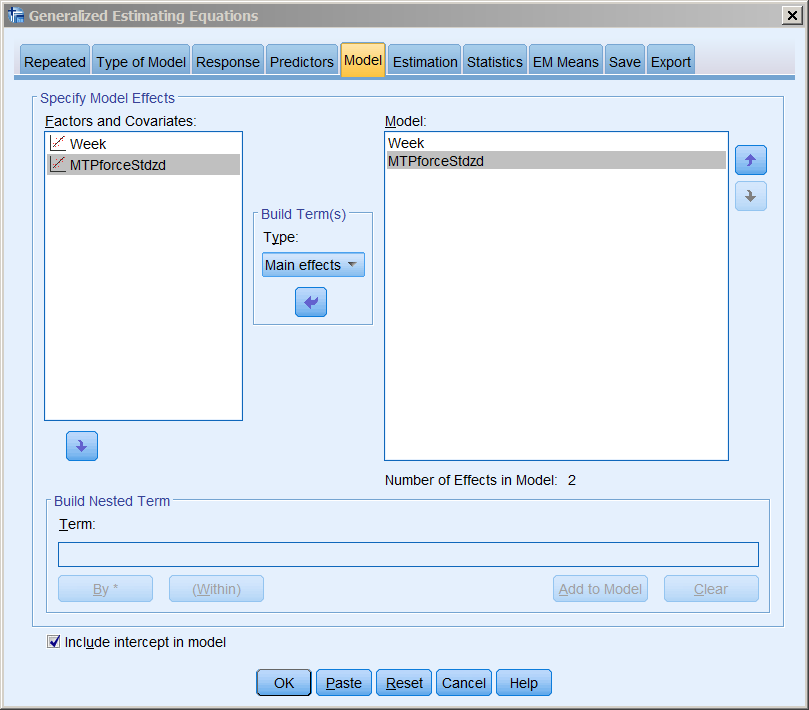
1. Click the **Predictors** tab and select Week and MTPforceStdzd as Covariates. We are interested in the extent to which the dependent variable changes in a simple linear fashion over the season, and of course the extent to which greater thigh-pull force is associated linearly with a player's count of effective rucks.



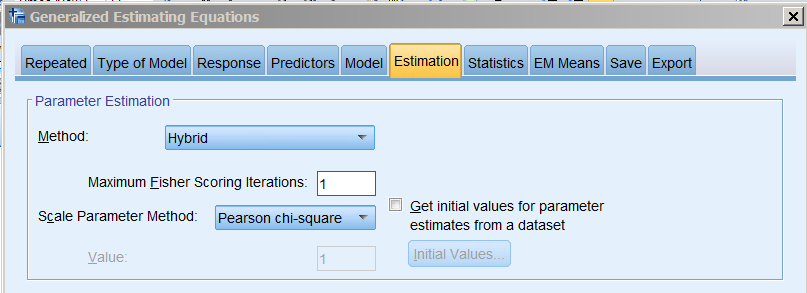
Here's where we would have selected Fwd2Bk2 as a factor, if we had not split the file. You can also use GameID as a factor instead of Week as a covariate.

Note also that you have to select LnTimePlayed (not TimePlayed) as the Offset Variable. What's this all about? The longer the player is on the field, the greater the count, so we need to estimate the count per unit of time to reduce the variability in the count arising from time on the field. The best unit of time is the normal duration of a game. You can't do it yourself by dividing the counts by the time on the field, because the resulting variable would no longer have a Poisson distribution. The stats program does the dividing for you, in the model predicting the log of the count, hence it has to be the log of the time on the field. If you're really sharp, you will understand that the stats package puts this variable into the right-hand side of the linear model with a forced coefficient of -1, because that amounts to dividing the predicted count by the time on the field.

1. Click the **Model** tab, select Week and MTPforceStdzd, and check Include intercept in model is ticked:

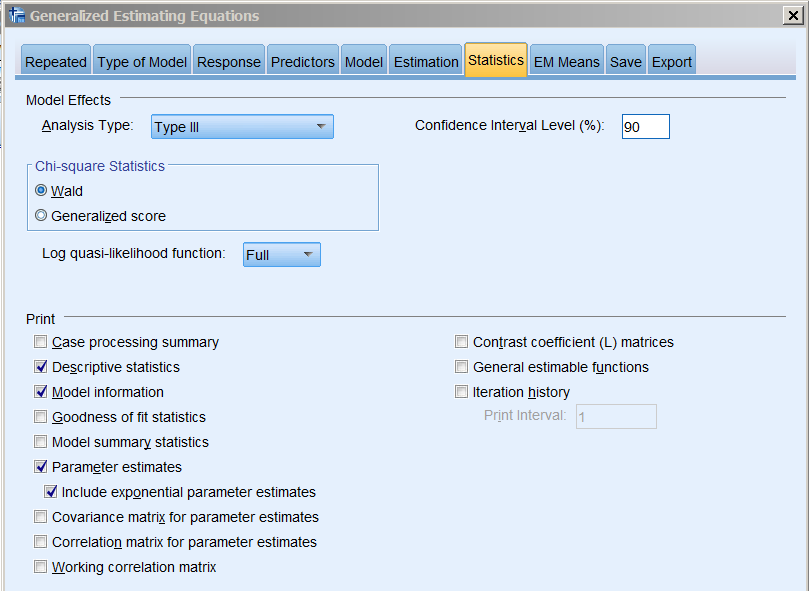


1. Click the **Estimation** tab and select Pearson chi-square as the Scale Parameter Method:

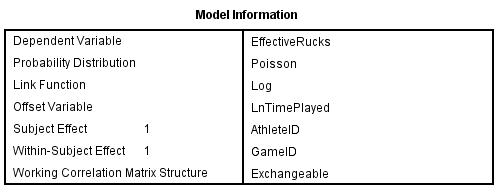


This is the option that allows for overdispersion. The value of scale is the over-dispersion factor, the amount by which the observed variance of the counts is greater or less than the mean predicted count (if the factor is >1 or <1, respectively).

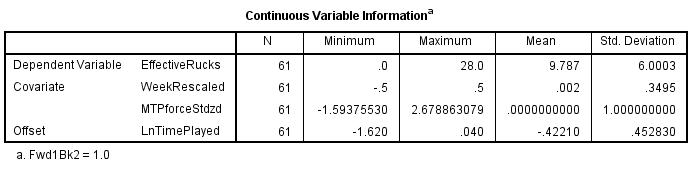
1. Click the **Statistics** tab, change the confidence level to 90%, and select/deselect options as follows:



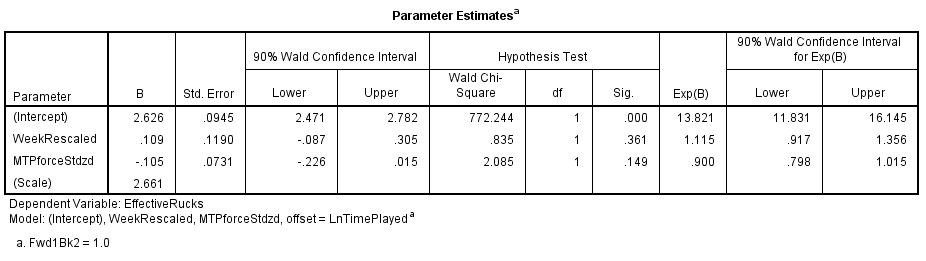
1. The **EM Means** (estimated marginal means) tab is irrelevant here, because we don't have any factor effects. The **Save** tab generates residuals and predicteds plus other statistics relevant to each observation as new variables in the dataset. The **Export** tab outputs model statistics available in the output as new datasets for further processing. We won't bother with either here.
2. Click **OK**, then check the model information to make sure you have dialed up the right options:



1. Here's the simple statistics for the all the variables for the forwards. Note the mean number of rucks (9.8). Check that the two covariates have values consistent with what I have said above. The values of the offset variable won't make much sense to anyone except the real experts. (For example, the maximum value of .04 implies that at least one subject had TimePlayed that was ~4% more than 80 min. Hint: if the transformation had been 100× the natural log, this value would have been 4.0.)



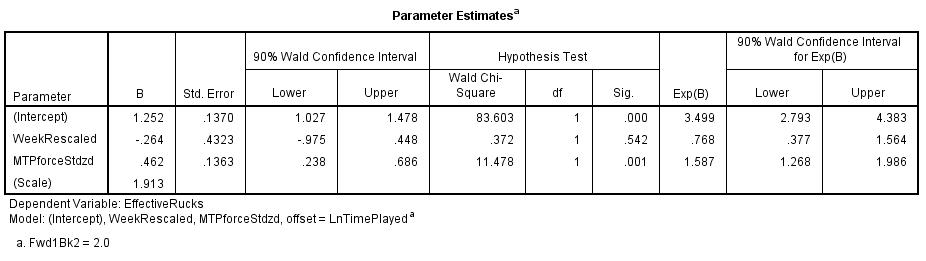
1. Here's the output with all the effects for the forwards:



The B is the coefficient of the effect in the log model. Exponential e raised to the power of the coefficient is the back-transformed value. Thus, for the intercept, Exp(B) = e2.626 = 13.821, which is the predicted mean count of effective rucks when the other covariates are zero. Now the wisdom of the rescaling and standardizing will start to become apparent. Their means are zero in the middle of the season (Week=3.5, WeekRescaled=0) and for the mean MTPforce (mean MTPforce=whatever, mean MTPforceStdzd = 0). This estimate for mean count of effective rucks correspond to the estimated marginal mean when you don't rescale or standardize covariates.

The Exp(B) value for WeekRescaled is the count ratio per unit of WeekRescaled, and since it has been rescaled to have a difference of one unit between Week 1 and Week 6, its value represents the overall linear change in effective rucks over the season. The value of 1.115 is right on the trivial-small threshold, if my default thresholds for count, hazard and proportion ratios apply here. The thresholds are 1.11, 1.43, 2.0. 3.3 and 10 for small, moderate, large, very large and extremely large, respectively, and their inverses 0.9, 0.7, 0.5, 0.3 and 0.1. Ideally we would have thresholds for efffective rucks that were tied to thresholds for differences in chances of winning a match, but until then, the defaults will have to do. Given these thresholds, you can see immediately from the confidence limits that the increase in effective rucks for this team over the season is clear, and you shouldn't have to use the Confidence limits and clinical chances spreadsheet to work out that it is possibly substantial.

The Exp(B) value for MTPforceStdzd is the ratio of effective rucks for players who differ by 1SD of thigh-pull force. To evaluate the effect of a subject characteristic, you have to consider the effect of 2SD, not 1SD. You can either double the B (and its confidence limits), then back-transform, or equivalently square the Exp(B) value, and its confidence limits. I could also have standardized MTPforce to have an SD of 0.5 rather than 1, so that one unit would correspond to 2SD, and you could read the value directly from the output. I usually do that, but here I thought it would be easier for you to understand doubling an SD of 1 unit, and it makes estimating the effect on proportions in logistic regression easier to understand. So, squaring the 0.900 and confidence limits 0.798 and 1.015 gives a reduction in effective rucks of 0.81, a small reduction, with confidence limits 0.64 and 1.03, which makes it a clear effect. The Confidence limits and clinical chances spreadsheet (Panel 3. Rate Ratio and other Log-Normally Distributed Effect Statistics) shows a likely reduction. Note that it is a counterintuitive effect: more mid-thigh pull force is likely associated with less effective rucks. Check out the backs and you see a clear large increase in effective rucks with higher mid-thigh pull force:

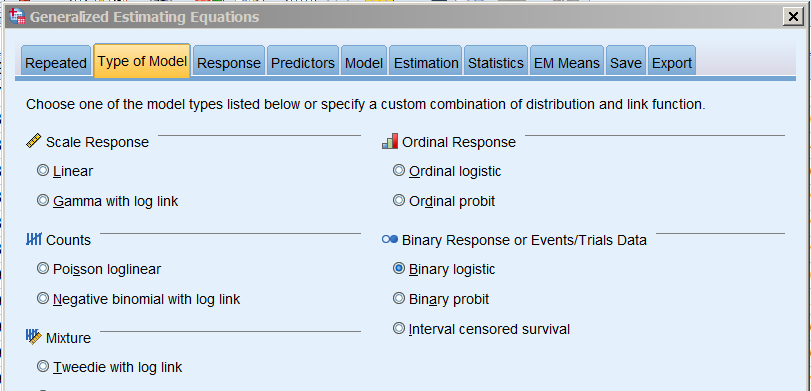


The backs get into only one quarter as many effective rucks, though. There's probably a good explanation for the counter-intuitive effect with the forwards.

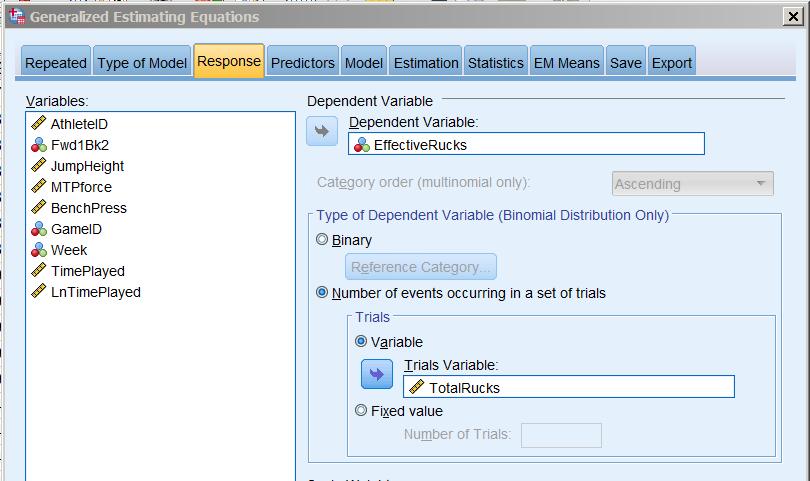
The scale value of 2.661 for the forwards indicates considerable overdispersion. The square root of this value, 1.63, is the factor increase in the players' SD above what you would expect for a mean count of 13.8. And by the way, how come the mean here is so much greater than the raw mean of 9.8? Think about it. Clue: time played.

**Logistic Regression**

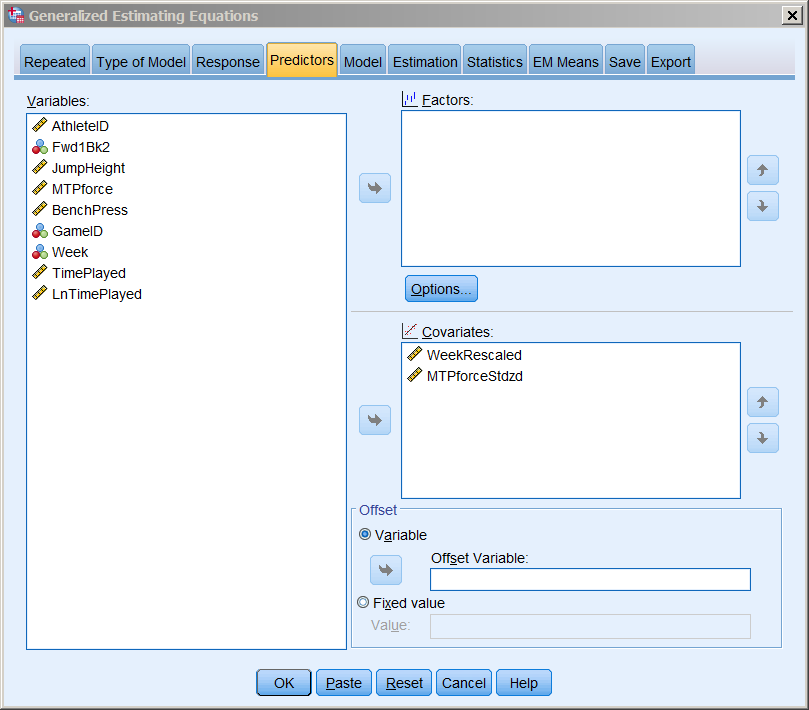
1. Now lets' modify the model to investigate the linear relationship between the fitness test and the proportion of effective rucks.
2. Click the **Type of Model** tab and select Binary logistic:



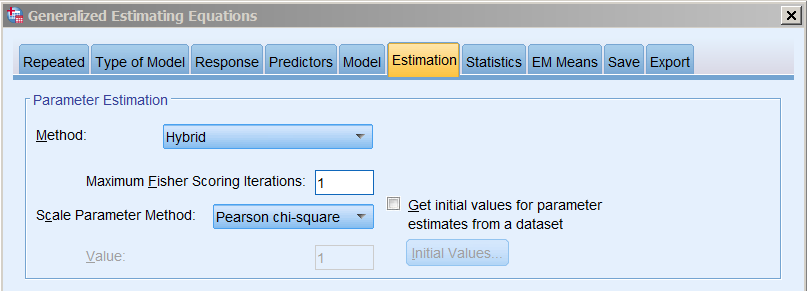
1. Click the **Response** tab, click Number of events occurring in a set of trials, and select TotalRucks.



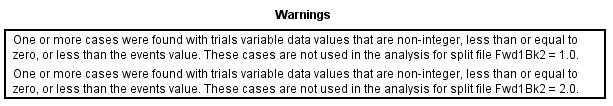
1. Click the Predictors tab and remove LnTimePlayed from the Offset Variable:



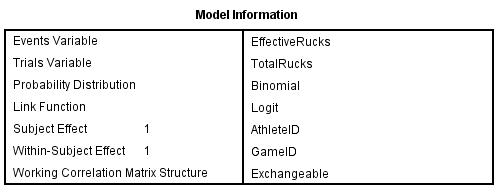
1. The **Model** window stays the same, but in the **Estimation** window you will need to reselect Pearson chi-square as the Scale Parameter Method:



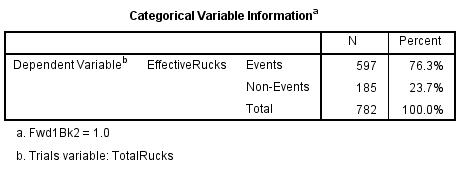
1. Click OK. First thing you will see is a warning, which here relates to several players who had zero TotalRucks:



1. Check the model information to make sure you have dialed up the right options:

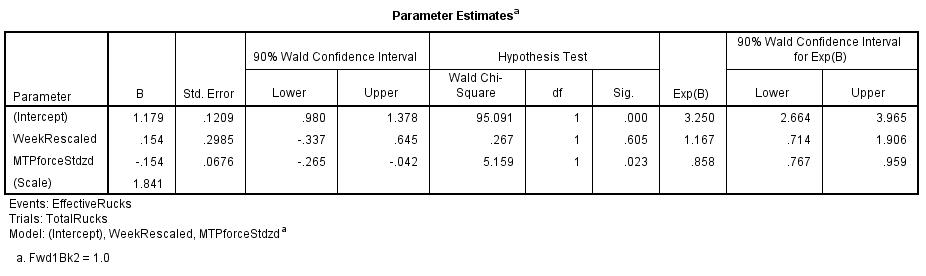


1. We've now got a panel of information about the total counts:



This shows that, if you add up all the effective rucks and all the TotalRucks for all the forwards, 76.3% of the total are effective.

1. Here are the effects for the forwards:



The Exp(B) are now odds (for the Intercept) and odds ratios (for WeekRescaled and MTPforceStdzd). These have to be turned into proportions and proportion ratios to evaluate the magnitudes of the effects. It turns out to be tricky I have done it on the accompanying spreadsheet, "process odds and odds ratios.xlsx". I started by copying and pasting the above panel of parameter estimates into the spreadsheet.

The first challenge is to get the right reference values for evaluating the effect of WeekRescaled. The odds for the Intercept represent the odds in the middle of the season. We want the odds at Week 1, which we will convert to a proportion at Week 1. Check that out first. You will notice than I have generated the odds at Week 1 by dividing the Intercept by the square root of the odds ratio for WeekRescaled. Why? Because Week 1 is -0.5 units of WeekRescaled below the middle of the season, and 0.5 in log units is a square root in factor units, and the minus means divide. Converting the odds to a proportion is much easier: odds/(1+odds). Now we have our reference proportion, and it's 0.75 or 75%; that is the effective rucks for the forwards are 75% of the total at the start of the season.

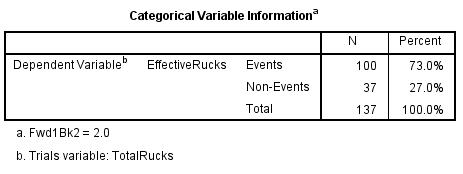
Note that if the GEE procedure had allowed us to use the /TEST command, we could have generated the odds at Week 1 that way without having to contend with square roots of odds ratios. However, we'd still have to do the remaining steps in a spreadsheet.

Next we want the odds at Week 6, to convert to a proportion at Week 6. To do that, we multiply the odds at the start of the season by an appropriate amount of the WeekRescaled effect, which in this is simply the odds ratio for WeekRescaled, because it represents the odds ratio between Week 1 and Week 6. Check that out.

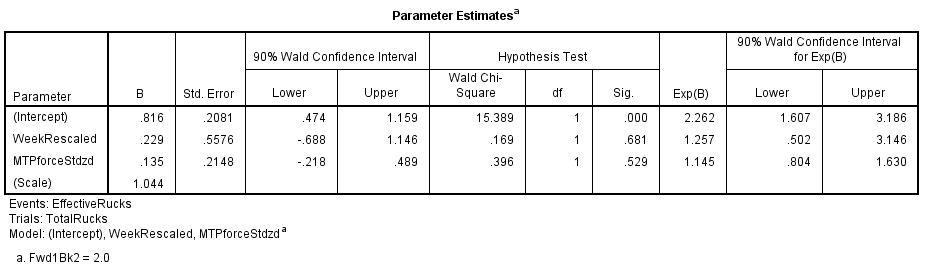
Next we get the ratio of the proportions, which is really simple. It's 1.04, so the observed effect is trivial, but is it clear? For that we have to generate the confidence limits, which we get by applying the p values for WeekRescaled (0.605) to the newly derived proportion ratio (1.04) using the "Confidence limits and clinical chances" spreadsheet. I have copied the relevant cells into this spreadsheet to show you what you should get. The confidence limits are 0.92 and 1.18. For smallest effects of 1.11 and 0.90, it's a clear effect, and likely trivial.

Too difficult? But what's the alternative? Just deciding whether an effect is significant or not is no longer tenable. There are no magnitude thresholds for the odds ratio, so you have to convert it to a proportion ratio. Fortunately you can apply the p value for the odds ratio to the proportion ratio: I've checked it with simulations for a wide range of sample sizes and effect magnitudes.

1. Here's the same analysis for the backs. First the raw frequencies:



1. And the parameter estimates:



1. I duplicated the spreadsheet for the forwards in the same workbook to analyze the odds and odds ratios for the backs. See the tab at the bottom of the window. With careful pasting, you just overwrite the panel for the forwards with the panel for the backs, then evaluate the proportion ratios. (I had to paste in the inference cells again.) The effects are obviously unclear.