

STAT0023 ICA2 (2019–20 SESSION) — GENERAL FEEDBACK

This was a challenging assessment, and many of you put a lot of work into it. We hope that you learned something from the experience! It seems that many of you looked at the feedback from last year's assignment: this year, there were fewer mistakes of the type highlighted there. There were also some really impressive submissions — including one that obtained full marks both for the coding and for the report, which is a phenomenal achievement.

On the other hand, there quite a lot of submissions where scripts failed to run — and 15 submissions out of 94 where the script didn't produce the predictions that were submitted: those submissions scored zero marks out of 20 for their predictions. There were even more submissions where the prediction file had the wrong format (e.g. including a header, or row names, or with the columns separated by commas instead of spaces): we tried hard to read all of these where possible, but sometimes the format was so far from what was required that we had to give up.

In this general feedback we'll comment on the coding, on your reports and then say something about the prediction performance. Please look at the specimen report to understand some of the points that we're making, both here and in your individual feedback. E.g. if your personal feedback says "Your graphs are clear and well labelled, although some more creative choices would have revealed more structure" then you might not immediately know what "more creative choices" might mean: look at the specimen report (look at the graphs, and also read the text carefully), and think about how much more information can be packed into a small space if you're creative about it.

Your individual feedback and provisional grades can be found on the ICA2 Moodle page: for each of you, the feedback is provided in a file called `Feedback.txt` which should be visible alongside the script that you submitted. If you can't see it, the reason is that it has gone off the right-hand edge of the Moodle tab: in this case, you may be able to reveal it by hiding the blocks — there should be an option to do this at the top-right of your Moodle page. Otherwise, try making the text smaller by pressing `<Ctrl>` and `-` (that's "Control-minus"). The feedback consists partly of a set of automated checks, and partly of some individual comments that are intended to help you see what you did well and what you did less well.

Coding

All students used R for this assignment. We were pleased to see that *almost* everyone is comfortable using R to read and manipulate data, carry out basic — and in some cases, quite advanced — statistical analyses, produce graphs and so on. If you got a 'B' grade or above then you have the computing skills that will enable you to survive if you get a job involving analytics in some way. Conversely, if you got a 'C' or below then you need to work on your computing skills if you're aiming for this kind of career.

Our other general comments on R coding are as follows:

- Most of your scripts were commented reasonably — and, in some cases, exceptionally —

well. Keep it up! We can't overemphasize the importance of commenting code: it might seem obvious while you're writing it, but if you have to come back to it six months later then you'll realise that it wasn't obvious at all. It is also important to explain *why* you're doing what you're doing, not just what. For example, a comment like

```
# compute the mean of fuel poverty
print(mean(obesity$FuelPoverty))
```

is not particularly helpful, since it's pretty much repeating what the code says. Instead, you might want to explain *why* you have decided to compute the mean of a particular covariate — for example, the reason might depend on some of the preceding output.

- Many of you clearly spent a lot of time commenting your code, but it was still not so easy to read because you did not use clear structure and blank space constructively. This is easy to do and makes a big difference for the user — or the marker looking at about a hundred of these 😊 Look at examples you have been given.
- Quite a few of you used additional libraries, even though you had been explicitly told not to (except for the few that were allowed). You were penalised for this. There is good reason why we did not allow additional libraries: unless you are doing something particularly advanced, which was not necessary for this assessment, base R is usually the cleanest and fastest way to achieve something, as well as the easiest for another user to follow.
- Many of the scripts showed little evidence of the programming principles learned in the first half of the course. A script is *more* than a sequence of commands that you've managed to get to work by pasting them consecutively into the command prompt. A good script shows structure and design, and (typically) uses the kinds of programming constructions that you have been taught. A good script has a clear structure; it is well commented to create a 'narrative'; and it uses functions to implement tasks that are needed repeatedly. It *certainly* does not contain any `View()` commands within it, as several of your scripts did: if we want to inspect a data frame then we can do it in interactive mode, but we don't need to do it every time we run a script.
- There were surprisingly many scripts which did not run without errors. This was usually because (a) you used a path to your own working directory, which doesn't exist on our machines, or (b) because you re-arranged your code or renamed your variables inconsistently. If you want to check that an R script works, ideally you should do it from a 'clean' start: restart your R session (via the Session → Restart R menu in Rstudio), then `source()` your script. This will enable you to check that you have created all the objects you need, by the time you need them.
- If you use `ggplot`, you need to enclose your plotting commands with `plot()` if you want to guarantee that your plots will appear when your script is `source()`d. We penalised people for not doing this, because you have been taught explicitly to do it.
- In the dataset that was provided to you, missing obesity values were denoted by `-1`. When you read data with 'dummy' missing value codes like this, the first thing you should do is

to set them to NA. If you don't deal with them *immediately*, there's a risk that you will accidentally forget about them later and treat them as genuine numbers. The efficient, no-nonsense way to do it is to go

```
ObesityData <- read.csv("obesity.csv", header = TRUE, na.strings = "-1")
```

If you tried anything more complicated than this, you should probably go back and do the Moodle quizzes for Week 1 a few more times.

Another benefit of setting the missing values to NA is that they will be handled automatically by R in any subsequent analysis, without your needing to split the dataset into two parts (as many students did). Aside from avoiding unnecessary copies that occupy memory, this allows you to use the entire dataset for principal component analysis or hierarchical clustering, as opposed to just the portion of the data with observed responses.

- In applied statistical work, you will often want to rename covariates into something meaningful, or create new variables from existing ones — for example, the obesity rate. However, it is not a good idea to make copies of the original covariates into separate vectors with meaningful names. Instead, you should rename the columns of the original data frame, and append any new columns as needed.
- Many of you used `attach()` and `detach()` in your code. This is not incorrect as such, but it can cause all sorts of confusion. It is much cleaner to either reference variables directly (for example, `obesity$counts`), specify the dataframe for commands which support this (such as `lm()`), or use the `with()` and `within()` functions.
- If you want to ensure that your saved graphics files will match what you see on the screen, it's helpful to ensure that your graphics devices (screen windows and saved files) are all the same size — otherwise you might find that text disappears when you save the file because it won't fit in the space available, or that the plots get squashed up unexpectedly so that you can't really see the patterns. This can be done by using the `x11()` command to open a graphics window, and then `dev.copy()` ... `dev.off()` to copy the contents of the window to a graphics file of the same size. There are several examples of this in the workshop scripts that you have been given. An alternative approach, which several of you used effectively, is to open the required graphics file directly using the `pdf()`, `png()` or `jpeg()` commands.
- If you're going to produce many similar plots, it is often helpful to use `par(mfrow=...)` to create an array of plots directly on your graphics device — many of you saved each plot individually and then pasted them into your reports, which can be messy. One potential problem with arrays of plots is that the default plot margins in R can be very wide if you have several plots on the screen: you can use `par(mar=...)` to change these defaults, and `par(mgp=...)` to move the axis labels closer to the plots so that you can use the space more efficiently. See the scripts that you've been given in workshops for examples.
- Many of you referenced the rows and columns of the dataset by number — for example, you selected the rows with observed obesity counts using `1:1785` or you manually picked

out a few columns by index. This didn't create any problems in this particular situation, but in general it is bad practice. It is not uncommon to start analysing datasets in which you discover there's some kind of problem, so you go back to the data provider who gives you some additional observations, or you decide to remove some problem cases, or whatever. If this happens, then the row numbers in your code will no longer be correct. You can, of course, go and change them but there's always a risk that you won't find all of the places that need changing — and then your results will be nonsensical (and you might not notice!). It's also very tedious and annoying to have to change your code if the dataset changes. It's much better to use logical conditions to select what you want.

Reports

If you haven't written an extended report like this before, then it can be quite difficult: you have to decide what to exclude, and then organise everything into a coherent narrative that is accessible to the reader. Many of you were probably doing this for the first time: hopefully the experience will be useful for you in the future.

It was clear from the reports that many of you had a reasonable operational understanding of the models that you were using. By and large, you seem to be comfortable using GLMs (and even GAMs in a few instances!). Many submissions used a quasipoisson model to account for overdispersion (well done for spotting this), and also understood how to use an offset term in a GLM — both of these tricks are worth remembering. Many of you also used a wide variety of criteria to help choose a model (e.g. p -values, nested model comparisons, diagnostics, predictive performance etc.) — this shows a good level of basic technical competence with the statistical methodology.

In a task like this however, there is *always* room for improvement: it really takes a lot of experience to juggle the subject-matter context, the mathematics and the statistical options available to you. The difficulty of doing good applied statistics is often underestimated. Here are some general comments that may be useful if you ever have to carry out a similar task in the future:

- There were few really good graphs in the reports. By 'really good', we mean that they are carefully chosen to highlight the most important messages, and that they are carefully presented so that those messages stand out very clearly and immediately.
- One common example of a 'not particularly useful' graph is a histogram or density plot of the response variable. Several submissions included such a plot, ostensibly as a way of checking whether the response can be assumed normally distributed or (in some cases) to look for evidence of overdispersion. It was quite disappointing to see so many of these, because the issue was discussed on the Moodle forum. For those who produced such a plot to check normality, please note: the assumption in linear regression models is that the *errors* are normally distributed, not the original observations.

Similarly, if you're going to check for overdispersion in a Poisson GLM then you can't just compare the mean and the variance of the response variable: you need to fit the model first and look at the variance of the Pearson residuals. To see why: suppose that the data

were indeed generated by a Poisson GLM so that, conditional on a covariate vector \mathbf{X} , the response variable Y has a Poisson distribution with mean μ (which is a function of \mathbf{X}). Thus $E(Y|\mathbf{X}) = \text{Var}(Y|\mathbf{X}) = \mu$. In this case, the iterated expectation formula gives us $E(Y) = E[E(Y|\mathbf{X})] = E(\mu)$ as you might expect (this latter expectation is over the distribution of the covariate vector \mathbf{X}). Likewise, the variance of the response is

$$\text{Var}(Y) = E[\text{Var}(Y|\mathbf{X})] + \text{Var}[E(Y|\mathbf{X})] = E(\mu) + \text{Var}(\mu) \geq E(\mu) = E(Y) ,$$

with equality only if $\text{Var}(\mu) = 0$ i.e. if the responses all come from Poisson distributions with the same mean. So: if the data are generated from a Poisson GLM then the overall variance of the response will be bigger than the overall mean, unless the GLM produces the same conditional mean for each observation (e.g. if the model contains only an intercept).

- Finally on the subject of graphs: we can't emphasize strongly enough the need to plot the response variable against potential covariates. We've said it many times during this course, and you've probably had it in previous statistics courses as well — despite this, several submissions contained no such plots. Some just reported correlations between the response and the covariates: this isn't enough because, as you all know, correlation is just a measure of linear association and is therefore very limited in what it can tell you.
- Some submissions showed a really good awareness of the background to the problem: people had taken care to find relevant literature and use it to inform their analysis. It wasn't always possible to find the evidence for some of your assertions, however. For example, many of you provided a list of references at the end of your report, but the report itself didn't cite them anywhere. You should always cite a reference at the point where you use it (see the specimen report for examples): this allows the reader to go and find out more if they want.
- There were also some submissions where people made speculative assertions about factors associated with variations in obesity levels, without providing any supporting evidence. This is not appropriate: in scientific work, *everything* must be supported by evidence — either from the data, or from previous published work, appropriately acknowledged.
- Although we should always consider previous work when carrying out a statistical analysis, we shouldn't allow it to completely dictate our conclusions — for example, by completely discarding variables that have not been considered by previous studies (several of you did this). Similarly, you shouldn't discard variables just because they don't seem important according to your exploratory analysis. The purpose of looking at previous people's work, and of carrying out an exploratory analysis, is to get you to a good starting point for your own analysis. After you get there and have made some progress, you can look to see whether anything else has emerged that you weren't expecting. After all, the purpose of a scientific investigation is to learn things that you didn't know already — if you don't look at anything that might surprise you, you won't discover anything interesting!
- On a related note: some of you ruled out the possibility of a relationship between a particular covariate and the obesity count (or rate) because you didn't expect any direct connection between them. This can be a bit naïve: some potential covariates may be proxies for other information that isn't available to you. Example: air pollution is likely to be higher in urban

than in rural areas, so pollution could be a proxy for degree of urbanisation (it's known that obesity prevalence tends to be higher in urban areas). Another example: some of you considered '% of population over 64' to be irrelevant because we're studying obesity prevalence in children — but the population age structure of a district will probably be related to the kinds of facilities available there e.g. you might expect more fast food outlets in UAs with larger young populations, and this in turn could influence obesity rates (indeed, the models with the best predictions all included this covariate — see below). Sometimes you have to use your imagination (and knowledge obtained by reading around the background to a subject) to think about the mechanisms that may be operating.

- Most people seemed to understand the concept of interactions, which is good. However, a substantial minority still seem to think, wrongly, that interactions represent relationships between covariates. This issue was also covered on the Moodle discussion forum.
- On the subject of relationships between covariates: there was less obsession with collinearity this year than in previous years, which is good. There *are* still some people in the class, however, who think that if you have two correlated covariates then you need to drop one of them. This is over-simplistic and, again, naïve. The effect of collinearity on regression models, GLMs etc. is that the coefficients will be estimated less precisely than if the covariates were uncorrelated: this only matters if the estimates aren't precise enough for their intended purpose and, in this case, dropping one of the covariates is only one possible option. In our specimen report, we *did* eventually decide that the two 'offence' variables were too highly correlated for the coefficient estimates to be interpretable: we didn't arbitrarily drop one of them though, choosing instead to work with the sum of the two variables as a measure of overall crime rate.
- There were a few reports that appeared to suggest, either directly or indirectly, that the response variable is transformed in a GLM. This is not true (remember our introduction to GLMs in Workshop 6, where the motivation for using a Poisson GLM in the Galapagos 'endemics' example was to avoid transforming the response and hence taking the log of zero). When marking your reports, we tried to give the benefit of the doubt where possible — for example, if you were trying to justify the use of a log link function by plotting the log of the response against a covariate. You just need to take care when describing the underlying model: the transformation applies to the *expected* response, not to the response itself.

Predictions

In your individual feedback, you will all find your prediction score and your rank in the class. We also calculated your root mean squared prediction error, defined as

$$\text{RMSE} = \sqrt{\frac{1}{447} \sum_{i=1}^{447} (Y_i - \hat{\mu}_i)^2}$$

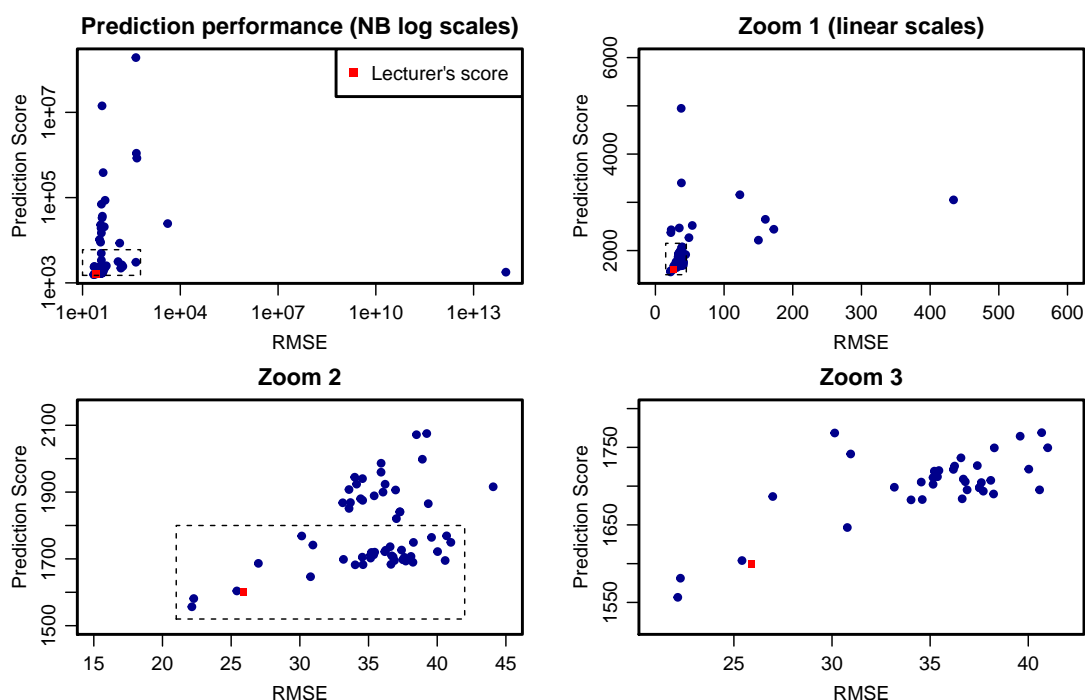


Figure 1: Performance of the predictions from the submitted scripts. The horizontal axis in each plot represents the root mean squared error of the predictions, and the vertical axis is the score S that was used to assign marks for your predictions. The top left panel shows the performance for all submitted scripts: the remaining plots zoom in on successively smaller regions corresponding to better (i.e. lower) scores. In the second and third plots, the ‘zoom’ region is indicated by a dashed rectangle.

using the same notation as in the question sheet. This is a more common measure of prediction performance than the score S that we used to assign your marks: the reason for using S is that it takes account of your prediction standard errors as well, and is able to reward those of you who gave an honest assessment of how accurate your predictions would be.

If you’re interested in knowing the real values for your predictions, we have uploaded a file `ObesityAgg.rda` to the Moodle page. If you load this into R using `load("ObesityAgg.rda")`, you will find a data frame in your workspace called `obesity_agg`: this is the complete data set, in which the `-1`s in the `obesity` column are replaced with the actual values and in which you can see the identities of the individual UAs.

There was substantial variation in prediction performance. Two submissions achieved better S scores than our specimen, and five achieved better RMSEs — very good! There were also some much less accurate predictions. Figure 1 shows your scores and RMSEs for all scripts, also showing the performance for the specimen answer. There is a massive range of scores, so the successive plots in the figure ‘zoom in’ to enable you to see more relevant detail.

It’s also worth pointing out that the overall standard deviation of obesity counts is just over 300:

so anyone with a RMSE greater than this is doing worse than somebody who didn't build a model at all. 11 of 94 groups achieved this: if you're a member of one of these groups then you should be quite embarrassed — and be very careful in the future when building models, particularly if you're going to use them to do things like predict investment returns!

Some of you may be interested in what models achieved the best prediction performance. Here is a summary of the five models that beat our specimen, in terms of either the S score or of RMSE.

Submission 1 (score 1 556, RMSE 22.14). The model here is a quasipoisson GLM with a log link function. The included covariates are $\log(\text{popCount})$ (as a covariate, rather than an offset), $\log(\text{percentage of population aged over 64})$, $\log(\text{home affordability})$, year and UA (both treated as factors), together with an interaction between year and $\log(\text{home affordability})$. The model has 338 coefficients in total.

Submission 2 (score 1 581, RMSE 22.28). This model is also a quasipoisson GLM with a log link. The covariates are $\log(\text{popCount})$ (again as a covariate rather than an offset), Year (treated as a factor), UA¹, pupil absence rate, percentage of population aged over 64, violent offence rate, air pollution, and a 'wealth score' obtained via principal component analysis of fuel poverty, excess winter deaths, home affordability and average weekly earnings. The model also contains an interaction between Year and violent offence rate. The model has 340 coefficients in total.

Submission 3 (score 2 372, RMSE 22.57). This uses a Poisson GAM with a log link. The covariates are $\log(\text{popCount})$, economic inactivity rate, a smooth function of pupil absence rate, bivariate smooth functions of violent and sexual offences and of the percentage of population aged under 18 and over 64, Year (treated as a factor) and UA. The model has 363.2 effective degrees of freedom.

Submission 4 (score 2431, RMSE 23.03). This is a Poisson GLM with a log link, and with $\log(\text{popCount})$ as an offset. The covariates are year (treated as a continuous variable), violent offence rate, percentage of population aged over 64, an interaction between average weekly earnings and home affordability, another interaction between fuel poverty and excess winter deaths, and UA (treated as a factor). The model has 327 coefficients in total.

Submission 5 (score 1 604, RMSE 25.41). This is a negative binomial GLM with a log link and with $\log(\text{popCount})$ as an offset. The covariates are year (as a continuous variable), air pollution, $\log(\text{weekly earnings})$, $\log(\text{home affordability})$, percentages of the population aged under 18 and over 64, $\log(\text{economic inactivity rate})$, violent crime rate, pupil absence rate, excess winter deaths and UA group (defined via hierarchical clustering on the means of the numerical variables in the supplied data — and with 105 groups in total). Interactions are included between air pollution and $\log(\text{weekly earnings})$, and between the percentage of population aged over 64 and $\log(\text{home affordability})$. The model has 116 coefficients in total.

¹The report for this submission suggests that the intention was to carry out a hierarchical clustering of the UAs, but there was an error in the code which meant that the 'groups' as calculated were in fact just the original UAs.

The score and RMSE for the specimen solution were 1 599 and 25.92 respectively. The model used here was a quasipoisson GLM with a log link, and with $\log(\text{popCount})$ as an offset. The included covariates are UA group (defined via hierarchical clustering on the coefficients of a simpler model in which UA was itself considered as a factor with 323 levels) and Year (treated as a factor) together with school absence rate, economic activity rate, percentages of population aged under 18 and over 64, home affordability, weekly earnings, elderly fall rate, gender pay gap and crime rate (sexual and violent offences combined). The model also contains interactions between UA group and pupil absence rate, elderly fall rate and gender pay gap. There are seven UA groups in this model, which has 40 coefficients in total.

All of these models used obesity count as a response, rather than obesity rate. Notice that none of them uses Region as a covariate (actually, one of them did — but this was redundant given that UA was already in the model) and they all relied to some extent on the variation of obesity rates between UAs. This was achieved either by including UA directly as a factor in the model (submissions 1–4 above), by clustering UAs into a large number of groups defined in terms of the covariates (submission 5) or by clustering into a much smaller number of groups defined in such a way as to be directly relevant to the aim of the analysis (specimen solution). One of the disadvantages of including a large number of UA coefficients is that most of the variation is explained by the UA so that it becomes difficult to identify the covariate effects: this is one reason for putting the UAs into a *much* smaller number of groups in the specimen solution. Another reason is that the resulting model is more parsimonious (i.e. simpler).

Notice that the S scores for submissions 3 and 4 above are much higher than those for the other three submissions and the specimen solution. This is because submissions 3 and 4 used Poisson models that failed to account for the overdispersion in the data, whereas all of the other submissions accounted for it (in most cases by using a quasipoisson model, but in one case using a negative binomial). The quoted prediction error standard deviations for submissions 3 and 4 were too small therefore, and their actual prediction errors were larger than they expected. The S score detects this overconfidence, whereas the RMSE doesn't. As a result, these submissions did not achieve such high prediction marks. This is not unreasonable: if you are overconfident in your predictions, sooner or later you will pay the price!

We can't go through every single model in detail here, but you may be interested in the prediction performance of the different types of models that were used. Figure 2 shows this. The top panel shows all of the S scores below 10 000: you can see from this that several different model types were able to produce both good (i.e. low) and bad (i.e. high) scores (there are some model types with no points on the plot: for these model types, there were no submissions with scores below 10 000). The second panel zooms in on the left-hand end of the distribution: here, you can see clearly that the best scores were obtained from quasipoisson and negative binomial GLMs. Note that the simpler linear models for obesity rate did not perform so well, although this was perhaps associated with the fact that there were no linear models with UA as a covariate. The bottom panel of Figure 2 shows the RMSEs for each model type: here, some of the Poisson models are also highlighted as performing well.

Richard Chandler and Ioanna Manolopoulou
21 May 2020

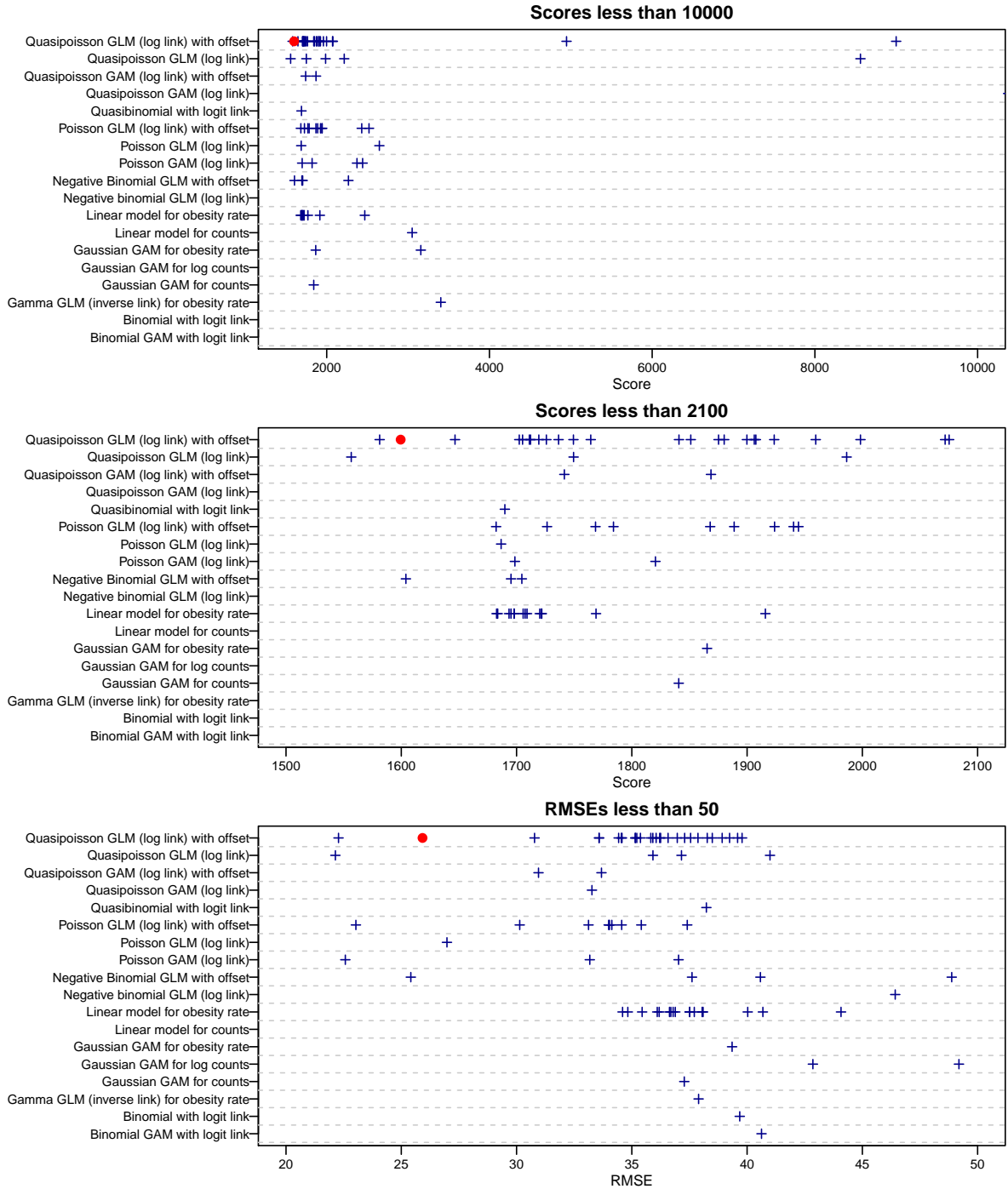


Figure 2: Prediction performance by model type. Each cross represents one group's submission: the red dots are from the specimen answer. The top two plots show the distributions of the score S that was the basis for the marking scheme: the top plot shows submissions with scores below 10 000, and the second plot shows submissions with scores below 2 100. The bottom plot shows the root mean squared error (RMSE) by model type, for submissions with a RMSE below 50.