University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAD29 (K. Butler), Midterm Exam
March 2, 2024

Aids allowed (on paper, no computers):

- My lecture overheads (slides)
- Any notes that you have taken in this course
- Your marked assignments
- My assignment solutions
- Non-programmable, non-communicating calculator

This exam has xx numbered pages of questions plus this cover page.
In addition, you have an additional booklet of Figures to refer to during the exam.
The maximum marks available for each part of each question are shown next to the question part.
If you need more space, use the last page of the exam. Anything written on the back of the page will not be graded.
You may assume throughout this exam that the code shown in Figure 1 of the booklet of Figures has already been run.

- regression (glowworm)
- logistic (kinase)
- ordinal logistic (faces)
- survival (larynx)
- two-way anova (wrinkle)

1. Female glow-worms attract males by glowing with part of their abdomen (called the "lantern"). Researchers believe the brightness of glow might be related positively to mating success. They measure the brightness of glow by the length of the lantern (in mm), and the mating success by the number of eggs laid by the female. Some of the data are shown in Figure 2, with some summary statistics below that.

   (a) [2] Based on Figure 3, do you think the researchers' belief is supported by the data? Explain briefly.

   My answer:

   There is a significant relationship between lantern length and the number of eggs, with a P-value of 0.0003. The slope is also positive, meaning that a longer lantern is associated with a larger number of eggs. So the researchers' belief is supported by the data. (A fairly gentle warmup.)

   (b) [2] Some predictions are shown in Figure 4, with the code that produced them at the top of the Figure. What precisely do the confidence limits in the first row of predictions tell you?

   My answer:

   With 95% confidence, the mean number of eggs for all female glowworms with a lantern length of 5mm is between 0.8 and 54.5.

   The key part here is the "all female glowworms with a lantern length of 5mm"; those are what you are predicting for.

   (c) [3] Which one of the intervals in Figure 4 is shortest? Why does that make sense?

   My answer:

   The second one, for a lantern length of 12. (Haul out your calculator for this, or eyeball the differences between `conf.low` and `conf.high` as about 50, 25, and 50 respectively, which is as accurate as you need to get.) One gimme point.

   A confidence interval for a mean response is shortest for the mean lantern length, which is 11.9 (look at the summary stats at the bottom of Figure 2). This is very close to the lantern length of 12 that we predicted for, and so we would expect this interval to be shortest. (The other two points.)

(d) [3] Why does it make sense that the first interval in Figure 4 is shorter than the first interval in Figure 5?

My answer:

The intervals in Figure 5 are prediction intervals, for the number of eggs of an individual glowworm whose lantern length is as shown (see the code above the prediction intervals). These are (much) more variable than the corresponding intervals for the mean response of many glowworms (in Figure 4), because an individual glowworm can happen to have a very large or small number of eggs just by chance, and the prediction intervals have to account for that. In finding a confidence interval for the mean response, these extreme individuals are averaged out.

Another way to say this is that the uncertainty in Figure 4 comes from the uncertainty in the intercept and slope (that come from our data being a sample rather than the whole population). The uncertainty in Figure 5 contains that, but *also* the fact that an individual observation can be a long way above or below the trend, which adds (considerably) to the length of the interval.

2. Over time, a hospital admitted 360 patients with suspected heart attacks. Some of the patients had actually had a heart attack, but some had not (they had had something that looked like a heart attack, but was actually something else). A doctor believes that whether a suspected heart attack actually is one might depend on the level of creatinine kinase (something that can be measured with a blood test). However, creatinine kinase level can only be measured to a certain degree of accuracy, and so in a sample like this one, you would expect several patients to have the same creatinine kinase level. The data are shown in Figure 6. The three columns are:

   - `mck`: the measured creatinine kinase level
   - `ha`: the number of patients with that creatinine kinase level whose suspected heart attack actually *was* a heart attack
   - `nha`: the number of patients with that creatinine kinase level whose suspected heart attack actually *was not* a heart attack.

   (a) [2] Some code and the output from the code is shown in Figure 7. Why is it necessary to do this in preparation for the analysis that follows?

   My answer:

   This is creating a response matrix for a logistic regression, with the number of "successes" (actual heart attacks) in one column and the number of "failures" (suspected heart attacks that were actually not) in the other.

   It is necessary because the data were laid out with *more than one* individual per row. (The points for saying this.)

   There are some clues that this is the case: the question says that there were 360 individuals but there are only 13 rows in the dataframe. Also, the last two columns are frequencies, rather than something like a column called `heart_attack` with values `yes` and `no` as you would expect if there were one row per individual patient.

   (b) [2] A model and its output are shown in Figure 8. Does creatinine kinase level predict whether or not a suspected heart attack actually is one, and if so, is it a higher or lower value that tends to go with an actual heart attack? Explain briefly.

   My answer:

   The estimate for `mck` is (strongly) significant, with a P-value less than 2.2×10^{-16}, so it most certainly does help to predict (one point). The model is predicting the probability of a suspected heart attack being real (the first column of `response`), and the estimate for `mck` is positive, so a *higher* value of `mck` goes with a higher probability of the suspected heart attack being a real one.

   Extra: if you look back at the data, at low levels of `mkc`, almost all the suspected heart attacks actually were not, but at high levels, almost all of them *were* actually heart attacks.

(c) [2] A plot of predictions from the model is shown in Figure 9, with the code used to produce the plot above it. What are two ways in which this plot supports your conclusions from the previous part (or contradicts them, if that's what you think it does)?

My answer:

The plot is a home-made version of what `plot_predictions` (from `marginaleffects`) would look like, if it worked for this model (it does not because the `response` we made is not part of the dataframe). The upper and lower points on the ribbon all the way across are the upper and lower 95% confidence limits for the predicted probability.

Above, we said that a higher value of `mck` goes with a higher probability of a suspected heart attack being a real one, and that is shown on this graph by the predicted probability increasing as `mck` increases. One rather obvious point.

We also said that there is a significant relationship between `mck` and the probability of a suspected heart attack being real. This shows up in the confidence interval being short at any value of `mck`: the probability is estimated accurately, and the probability is definitely higher for higher values of `mck` because the whole confidence interval is higher. Say something that talks about the accuracy of estimation for the second point.

(d) [1] Some more predictions are shown in Figure 10, with the code that produced these predictions shown at the top of the Figure. What are these predictions of?

My answer:

The *log-odds* of the probability that a suspected heart attack is a real one, as indicated by the `type = "link"` on the `predictions` line. This is what a logistic regression is actually predicting.

(e) [2] How are the predictions in Figure 10 consistent with one (or two) of the numbers in Figure 8? Explain briefly.

My answer:

My immediate reaction is to look at the changes from each prediction to the next: an increase of almost exactly 0.7 from one to the next. Since `mck` increases by 20 each time, this increase should be 20 times the "slope" estimate in Figure 8, which is about 0.035:

```
0.035 * 20
```

```
[1] 0.7
```

and so it is.

Another reasonable way to tackle this is to reproduce one of the predictions, using the intercept as well as the slope from Figure 8. The intercept is about -3, so the predicted log-odds for `mck` of 20 would be about

```
-3 + 20 * 0.035
```

```
[1] -2.3
```

and for 40 it would be

```
-3 + 40 * 0.035
```

```
[1] -1.6
```

which are close enough to the values shown in Figure 10. (Feel free to be more accurate than this, but this is the idea.)

3. What determines whether a man is judged attractive by female college students? 38 men were rated for attractiveness, on a scale from A (most attractive) to D (least attractive). Each of the men had two other measurements taken:

- `MaxGripStrength`: the man gripped a handheld dynamometer in their dominant hand and squeezed as hard as they could. The maximum of three grip strength measurements was taken (measured in kilograms).
- `SHR`: shoulder to hip ratio; the circumference of the shoulders divided by the circumference of the waist.

Some of the data are shown in Figure 11.

(a) [2] A model is fitted, as shown in Figure 12. Why did I use `polr` (from package `MASS`) rather than something else?

My answer:

Our response variable `Attractive` is categorical with four categories (more than two), and those categories have a natural order with A being most attractive and D being least. This means we need to fit an ordinal logistic regression, which is what `polr` does. Roughly speaking, one

point for "ordered" and one for "more than two categories". If there were only two categories, it wouldn't matter whether the categories are ordered or not; you model the probability of being in one of them, and then you can work out the probability of being in the other.

(b) [2] In the model shown in Figure 12, why do you think I added a squared term in `SHR`?

My answer:

I actually added this term because I expected there to be a shoulder-hip ratio that was rated most attractive: that is to say, I expected rated attractiveness to go up and then down again as `SHR` increased. If I had only a linear term in `SHR`, attractiveness would have to keep getting better (or worse) as `SHR` increased. This seemed unlikely to me, as it seemed that a man with a very large `SHR` (or, for that matter, a very small one) would look very oddly-shaped (and therefore unattractive), that is to say, having very large or very small shoulders.

One point for saying that I wanted to model a non-linear relationship; the second is for coming up with a plausible reason I might want to do so, given what the data represent. I was fairly picky about whether you had earned the second point here.

(c) [2] Some predictions are shown in Figure 14. In the code above the predictions, why did I decide to use `pivot_wider`?

My answer:

`predictions` gives us one long column of predictions, one predicted probability of each level of attractiveness for each combination of grip strength and `SHR`. This is difficult to read (because I want to see how overall attractiveness changes as `SHR` or grip strength change), so I re-format it to have the response categories go across the page. Some reasonable discussion along these lines.

(d) [3] Look at Figures 13 and 14. How are they telling a consistent story about how `MaxGripStrength` influences attractiveness? Explain briefly.

My answer:

The `drop1` table says that `MaxGripStrength` is very far from being significant (one point). In the predictions, we see that for any fixed value of `SHR` (say, comparing the first two rows), the predicted probabilities for each attractiveness category are very close to each other (one point). Thus, the effect of `MaxGripStrength` is very small or non-existent (the third point). You could get away without explicitly saying all three of those things if it seemed clear to me that you understood what was going on.

The obvious thing to do is to take out `MaxGripStrength`, but I wanted you to see what happens if I left it in the model.

(e) [3] According to Figure 14, how would you describe the effect of `SHR` on attractiveness? Explain briefly.

My answer:

Pick a value for `MaxGripStrength` (either one will do), and see how the probabilities of the attractiveness categories change as `SHR` changes. The probability of attractiveness A goes up and then down again, and correspondingly the probability of being rated D goes down and then up again, so that the overall rating is highest at `SHR` of 1.2 and lower either side.

Make a statement about what happens to the overall attractiveness rating as `SHR` changes: for example, look at the probabilities of both A and D and notice that they are both indicating the same thing. For this, you need to do more than look at (say) rows 1 and 3, because you need a picture of what is happening to attractiveness as SHR goes through all four values. Most people who said there was an increasing or a decreasing effect were only looking at two values of SHR, not at all four of them.

Saying something like "attractiveness increases up to an SHR of 1.2 and decreases again after that" is also good, and you could probably get away with only looking at the probability of A to see what happens here.

Extra: I didn't ask you about the significance of `SHR`; according to Figure 13, neither it nor the squared term seem to be especially close to significance. But I wanted to try a squared term, because of what the data are about, and I suspect that `SHR` and its square are pretty highly correlated, so that there is multicollinearity, and the P-values of both of them are higher than you might expect.

The obvious way to assess that is to try taking out the squared term:

```
Faces.2 <- update(Faces.1, . ~ . - I(SHR^2))
drop1(Faces.2, test = "Chisq")
```

	Df	AIC	LRT	Pr(>Chi)
	NA	102.9592	NA	NA

MaxGripStrength	1	101.0405	0.0813532	0.7754718
SHR	1	110.2420	9.2828969	0.0023130

and now you see the strong effect of `SHR` (that you would have expected because the predictions changed substantially as `SHR` changed).

How do predictions from this model look?

```
new <- datagrid(model = Faces.2, SHR = c(1, 1.2, 1.4, 1.6), MaxGripStrength = c(42, 54))
cbind(predictions(Faces.2, newdata = new)) %>%
  select(group, estimate, SHR, MaxGripStrength) %>%
  pivot_wider(names_from = group, values_from = estimate)
```

```
Re-fitting to get Hessian
```

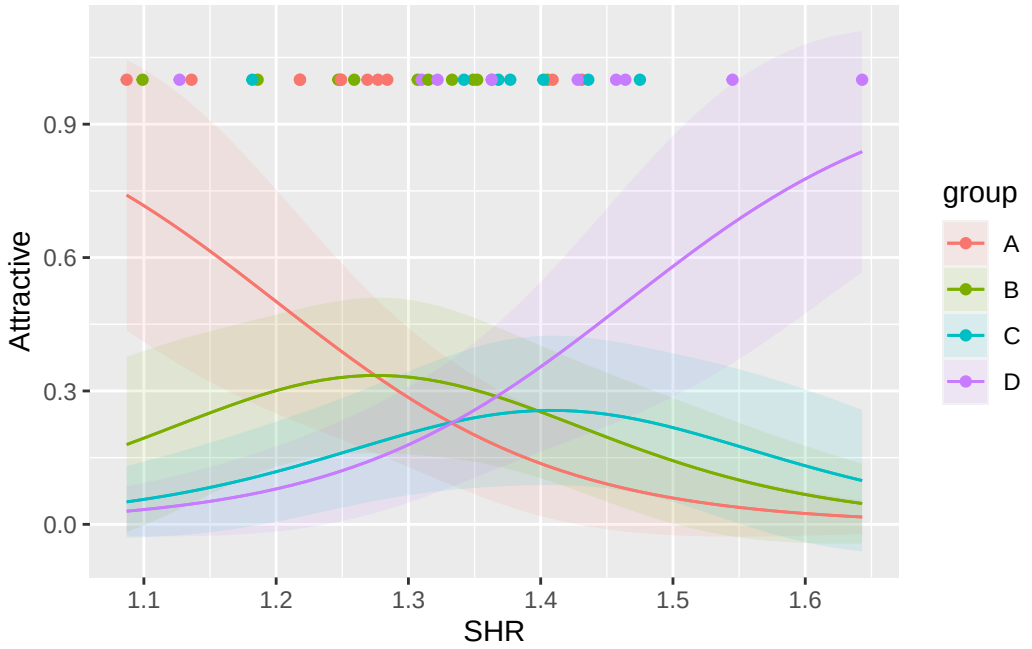
SHR	MaxGripStrength	A	B	C	D
1.0	42	0.8705688	0.0938513	0.0228202	0.0127597
1.0	54	0.8570508	0.1032059	0.0254510	0.0142924
1.2	42	0.5146358	0.2957179	0.1138712	0.0757752
1.2	54	0.4858963	0.3061526	0.1237196	0.0842315
1.4	42	0.1432104	0.2592732	0.2553708	0.3421456
1.4	54	0.1296714	0.2454946	0.2563573	0.3684766
1.6	42	0.0256728	0.0703199	0.1366069	0.7674004
1.6	54	0.0229481	0.0635191	0.1262407	0.7872921

As we guessed, attractiveness now proceeds in one direction according to these predictions: high attractiveness at low SHR, low attractiveness at high SHR. That is a much less interesting, and I think also less reasonable, story than we had before.

One way to think about this further is to plot the predictions, but with the data added. Next is the model we just fitted, without the squared term. Each man has a different SHR value, so I can't easily work out proportions as I did for the coal miners, so I'm plotting each man's attractiveness rating and their SHR across the top:

```
plot_predictions(Faces.2, condition = c("SHR", "group")) +
  geom_point(data = Faces, aes(x = SHR, y = 1, colour = Attractive))
```

```
Re-fitting to get Hessian
```

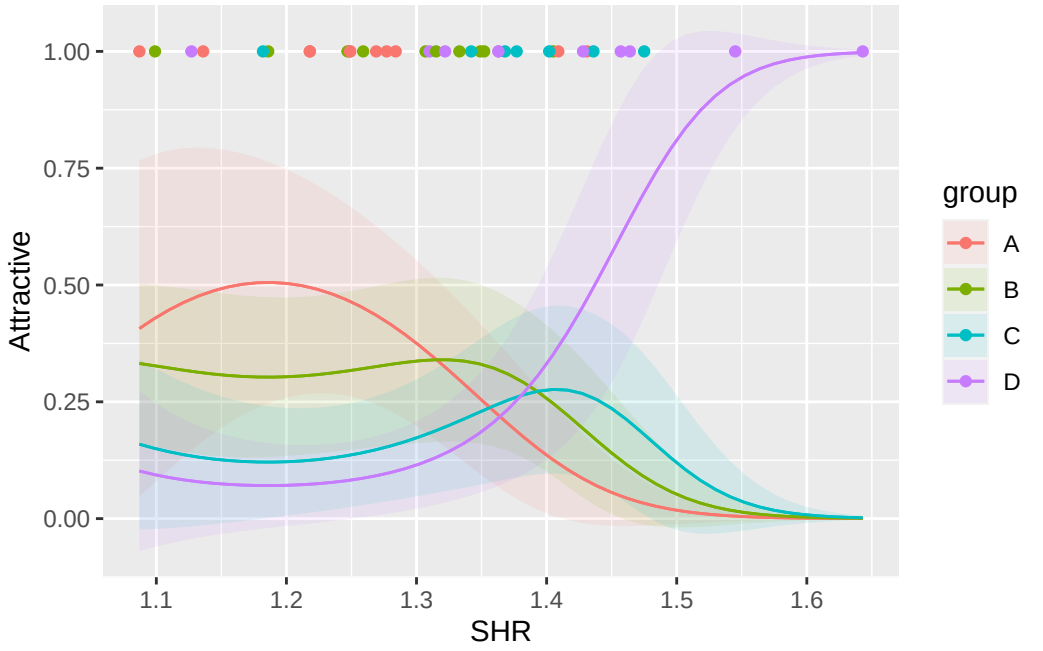


The purple dots (the men rated D) are mostly on the right, and the red dots (rated A) are sort of on the left. I don't think there's anything in the data to indicate that the red curve should keep going up as SHR decreases; that's just an implication of the model.

Here's the model we fitted first, with the squared term:

```
plot_predictions(Faces.1, condition = c("SHR", "group")) +
  geom_point(data = Faces, aes(x = SHR, y = 1, colour = Attractive))
```

```
Re-fitting to get Hessian
```



The predictions are not that different (which is probably why the squared term was not significant). Really, the principal difference is what happens on the left: now, the red curve goes back down again as `SHR` decreases (and the other curves level off), so that on this graph, the maximum probability of being rated an A comes at an `SHR` of about 1.2. Looking at the points along the top, I think it fits with the data to have the predicted probability of being rated A be highest up to an `SHR` of about 1.3.

In each case, there is a lot of uncertainty about the red curve, which is probably why statistically we couldn't distinguish between these two models.

4. 90 males diagnosed with cancer of the larynx (where the vocal cords are) at a Dutch hospital took part in a study. Cancer cases are classified into one of four stages, numbered 1 through 4, in the column `stage` of this dataset. Stage 1 is the least advanced stage of the cancer, and Stage 4 is the most advanced, which would be expected to be worst. The numbers serve only to identify the stages; the numbers 1 through 4 have no meaning as numbers. The main aim of this study was to investigate the effect of the stage of cancer on survival times. The researchers also recorded the `age` of each patient, and the year of diagnosis (in `diagyr`, as two digits, the year minus 1900). The `time` in months from diagnosis until death, or until the end of the study, was also recorded, along with an indication `delta` of whether the patient was alive or dead at that time.

   Some of the data are shown in Figure 15.

(a) [3] What code would I use to create a suitable response variable `y` for a Cox proportional-hazards model? This could be either a new column in the dataframe, or a separate variable outside the dataframe.

My answer:

In the dataframe:

```
larynx %>% mutate(y = Surv(time, delta == "dead")) -> larynx
```

or, as a separate variable,

```
y <- with(larynx, Surv(time, delta == "dead"))
```

or, if you must:

```
y <- Surv(larynx$time, larynx$delta == "dead")
```

(on the last one, you will probably get fed up with writing `larynx` and the dollar sign twice. This is why we have `with`.)

Use `Surv` (from the `survival` package); the first input is the time the patient was observed (the column called `time` here), and the second input is something that will be `TRUE` if the event (here death) happened for the patient. If you make a new column in the dataframe, you should probably save the result, but I have no problem if you don't, because the issue is to get the `Surv` thing right. Fast ways to lose points included:

- calling the response variable something other than `y` (calling it `mth` reveals that you were copying from your notes without thinking)
- using a variable name other than `delta` in the second part of `Surv`
- forgetting that you need a second = between `delta` and `"dead"` because you are making something that has to be true or false
- forgetting the quotes around `dead` because it is a literal value, not a variable name

If you want to do some pre-calculation, for example to make a variable `status` that is true or false, maybe using `ifelse`, and then feeding that `status` as is into `Surv`, be my guest. It isn't actually necessary here, but if it will work, it is good.

(b) [2] The values of `y` for the first twenty observations are shown in Figure 16. Why do some of them have a `+` next to them? How do you know?

My answer:

These are the patients who were still alive at the end of the study. These are the censored observations: that is to say, the patients for which the event (death) never happened before the end of the study; these are by convention labelled with a `+` in survival analysis.

One point for "still alive", one for some discussion of censoring in this context.

The values of `y` are given in the same order as the original data I showed you in Figure 15, so you can go back and check that you were right about them still being alive. Thus the first point was a bit of a giveaway.

Now that I have graded this, it was perhaps a bit too easy to get two points here. Some discussion along the lines of the analysis having to include both people who died and people that were still alive when last seen was enough to get the second point (it was nice if you included the idea that the censored observations actually mean "a lifetime of at least the value shown").

(c) [2] A proportional-hazards model is shown in Figure 17. Why did I include the `drop1` output in addition to the `summary` output?

My answer:

I am looking to see whether anything can be removed, and `stage` is categorical, so I need to see whether I can remove it as a whole, which the `drop1` output tells me (it tells me that I cannot remove it), and the `summary` output does not (it only tells me how the stages compare to the baseline stage 1).

There are some clues about the categoricalness of `stage`: a sentence in the question about how the stage numbers are labels rather than being meaningful as numbers, and the way `stage` appears in the `summary` output (with three coefficients, for all the stages except the baseline).

There was also a clue in the wording of the question: the way I phrased it, I tried to suggest that the best answer would say something about what `drop1` gives you that `summary` does not. There was one point for saying something about assessing significance of explanatory variables, which a lot of people got; this is sort-of relevant discussion without actually answering the question.

(d) [2] A second proportional hazards model is shown in Figure 18, and some further analysis is shown in Figure 19. Why did I need to do the further analysis, and what do you conclude from it?

My answer:

I needed to do the analysis in Figure 19 because I had removed two (more than one) explanatory variables from the first model. To check whether that was reasonable, I could not use the

`summary` or `drop1` output from the first model, because that is only applicable to removing *one* explanatory variable. So I had to do the `anova` to compare the fit of the two models.

The two models are not significantly different in fit (P-value 0.39), so we prefer the smaller, simpler one with just `stage` in it, the model `larynx.2`.

An appreciable fraction of that for the two points. I definitely wanted you to recognize what the two models were that you were comparing, and which one you preferred. The issue here was *not* the significance of `stage` (it was significant in both models) but in whether it was reasonable to remove both `age` and `diagyr` at once (yes, it was reasonable). The output of model `larynx.1` suggested that removing them both would be all right, but to prove that, you need the `anova` (because, especially if there is multicollinearity, removing one might make the other one significant).

Extra: the C32-style approach would have been to remove just the year of diagnosis first (largest P-value), and then re-fit with `stage` and `age`. When I was putting this question together, that's what I did (and I found that age was still not significant, so I removed that too). This seemed a bit involved for an exam question that wasn't really testing this, so I took them both out and produced the `anova` version for you. It's actually not an ANOVA but a likelihood-ratio test for a model like this, but the way it works is the same: if the bigger model had fit significantly better (in the sense here of having a much higher likelihood or much lower deviance), the P-value would have come out small. It didn't, so I was justified in tossing `age` and `diagyr` both.

Details, for those interested: the likelihood is the probability of observing exactly the data you did. This will depend on the parameters you are trying to estimate (the coefficients of the proportional-hazards model in this case). What you do is choose values for those parameters that maximize the probability of seeing what you saw (the so-called "maximum likelihood estimators" or MLEs). Cox worked out how to calculate them for this model; in other models you have seen, things are much simpler. For example, in estimating the mean of a single normally-distributed population, the MLE of the population mean is the sample mean, and the MLE of the population SD is the sample SD (more or less). In (simple) regression, the maximum likelihood estimates of intercept and slope are the same as the least squares estimates, under the assumption of normally-distributed errors with constant SD. (You probably saw those in your second course, B27 or similar.)

One really nice thing that comes out of the likelihood theory is that you can compare any two models where the first one is the second one plus extra stuff. Our models `larynx.1` and `larynx.2` are like that, where the extra stuff is age and year of diagnosis. What you do is to fit each model, estimating the coefficients in it by maximum likelihood, and then compare the two likelihoods by looking at their ratio. If the likelihood for `larynx.1` had been several times bigger than for `larynx.2` (it wasn't), then we would have known to keep `age` and `diagyr`. The trouble is that likelihoods tend to be very small probabilities (even at their maximum, it is very unlikely that you would have observed *exactly* what you did observe), so it is numerically a much better idea to look at the *log* of the likelihoods, which will be a very negative number. There are also good mathematical reasons to look at the log of the likelihood rather than the likelihood itself. So, looking at the *ratio* of two likelihoods is equivalent to looking at the *difference* of two log-likelihoods (because of rules of logarithms). When you do that, as Figure

19 shows, what you do is to take the difference in log-likelihoods (in the `loglik` column), which is about 0.95, and double it to get 1.90 (the 1.88 in the `Chisq` column is 1.90 more accurately). To get a P-value, you look in chi-squared tables; the df are the total df of the things that were in `larynx.1` but not `larynx.2` (two quantitative variables, so 2). If you look this one up in your tables, you'll get a P-value "bigger than 0.1", or something like that.

The likelihood ratio test is very general: whenever you are doing statistics you have a probability model and you can work out the likelihood (in principle at least) for any model you care to fit. The P-value from a chi-squared distribution is actually an "asymptotic approximation", meaning that it works better for larger sample sizes, but it is often very useful and accurate enough, and there isn't always any better theory. The proportional-hazards model is one of those cases where likelihood ratio is about the best you have.

(e) [4] Figure 20 shows some predicted survival curves. What do you conclude from the graph, and how is this conclusion consistent with the appropriate one of Figures 17 or 18? (Say which one of these last two Figures you are looking at.)

My answer:

The display of `new` above the graph shows that we are predicting survival probabilities for the different stages, and that the strata numbers are the same as the numbers of the stages (which makes your life easier). We are talking about an undesirable event (death), so the best stage to be in is the one where survival for longest is most likely (up and to the right). Stage 1 and Stage 2 are very close, with Stage 1 being slightly better. Stage 3 has worse survival, and Stage 4 the worst of all. Two points.

The predictions that made the graph in Figure 20 were made from the model `larynx.2` (see the `survfit` in the definition of `s` above the graph). So we need to look at Figure 18. In this Figure, look for the Coef numbers in the `summary` output. These show the effect of being in different stages on the hazard. The baseline (stage 1) is zero, and the others are relative to that. The coefficient for stage 2 is only slightly bigger than the baseline, so the hazard of death is only slightly higher than for stage 1. The coefficient for stage 3 is much bigger (more positive), and the coefficient for stage 4 is much bigger still, so survival on stage 3 is worse than for stages 1 and 2, and survival on stage 4 is worse than for stage 3. These are entirely consistent with the graph. (This is better than talking about P-values, because the P-value could be small if the hazard rate was significantly higher *or lower* than the baseline Stage 1.) You can undoubtedly say this more briefly than I just did, but make sure you make the connection between the numbers in the `summary` output in Figure 18 and the relative positions of the estimated survival curves.

It is not really insightful to talk about how all the survival curves are going down over time; this is something that *always* happens in survival analysis, because the chance of it taking a long time for the event to happen is never greater than the chance of it taking a short time.

I realized as I was marking this that it would have been better to split it into two parts: first, have you interpret Figure 20, and then, in the next part, have you interpret the appropriate

one of Figures 17 or 18. It is probably better to have each question part ask you one thing (as opposed to two things, which you then have to check that you answered both of).

5. After you come out of a swimming pool, your fingers are wrinkled because they are wet. If you have to do a precision task with a wet object, is it better if your fingers are wrinkled or dry? To find out, 80 participants were observed doing a "transfer task" under various conditions. The task was to pick up an item with the right thumb and index finger, pass the item through a small hole, grab it with the left thumb and index finger, then put the item into a box through a hole in the lid. Each participant was timed, and the time to complete the whole task was recorded. Some of the data are shown in Figure 21. The columns of interest are:

   - `Time`: total time to complete the task, in seconds.
   - `Fingers`: whether the participant's fingers were `wrinkled` or `non` (not wrinkled)
   - `Objects`: whether the object being handled was `wet` or `dry`.

   For this question, carry out all tests at $\alpha = 0.10$.

   (a) [2] A plot is shown in Figure 22. What is the main thing that you learn from this plot? Explain briefly.

   My answer:

   This is an interaction plot (see the code above the graph), so we look at the lines and decide whether they are parallel. It seems pretty clear that they are not, so we would expect to see an interaction between `Fingers` and `Objects`.

   That's all I needed. If you wanted to be more specific about the comparison of means as well as (or instead of) the above, that's fine too; just make sure you get at the likely interaction effect somehow. For example, "if the object is dry, there is not much difference between wrinkled or non-wrinkled fingers in the time needed to complete the task, but if the object is wet, wrinkled fingers complete the task much quicker". Saying something like this now will help you for part (d), provided, of course, that the analysis along the way supports this conclusion. (These are the kind of words you would use in describing simple effects.)

   Try to avoid using the word "significant" in this part, because we don't have P-values for anything yet, and that is what we need to assess significance in a statistical context.

   I am not showing you the grouped boxplot, but if you had that as well, you could judge whether the non-parallelism was beyond chance, given how much variability there was. But don't overthink this one. If you weren't sure, you could come back to this one after doing the next part, and see if you can make this part and the next one consistent.

   (b) [2] Some analysis is shown in Figure 23. Is the result of this analysis what you were expecting, given your conclusion from the previous part? Explain (very) briefly, keeping in mind the α we are using.

   My answer:

   The P-value for the interaction term is 0.084, which is less than $\alpha = 0.10$, so at this level, the interaction term is significant. This agrees with the interaction plot in the previous part, which also indicated that an interaction was present.

   About the briefest answer that will work is "both the `aov` and the interaction plot indicate that

there is an interaction, so they agree with each other".

If you forget and use the standard α of 0.05, you will lose something, but if you then say something like "I expected to see a significant interaction, so I was surprised that it was actually not significant", you will get something for being consistent with yourself.

(c) [2] Someone tells you that the P-value for `Objects` in Figure 23 is the smallest, so you should now do a Tukey analysis of the types of objects. Do you think they are correct, or not? Explain briefly.

My answer:

No, they are not: you have a significant interaction, so that is the first thing that you should assess. When you have a significant interaction, that *is* the finding, and any other significant terms, even ones with small P-values, do not enter into the interpretation. The next stage is to understand the interaction (which the next part rather gives away).

That was the answer I was expecting, but because you are smarter than I am, some of you also realized that you can refute the someone's analysis by saying that there are only two types of objects, wet and dry, and we already know they are different (the task is done more quickly with dry objects than with wet ones), and so there is no need for Tukey for that reason. I wasn't anticipating that as an answer, but it also answers the question, if perhaps in a less insightful way than the first answer, so it also gets two points. I could perhaps have quibbled that you should not be looking at the test for `objects` at all until you have looked at the interaction, but I didn't.

You *could* run a Tukey on the *interaction*: there are only four combinations ($\$ = 2 \times 2\$$) of `objects` and `fingers`, so it would not be that hard to interpret. But that's not what the person was saying to do.

(d) [3] Some more analysis is shown in Figure 24. What precisely does this enable you to conclude about the effects of `Fingers` or `Objects` or their combination? Looking back at Figure 22, summarize your conclusions clearly.

My answer:

This additional analysis is of simple effects, specifically the simple effects of `Fingers` when `Objects` is held fixed.

The first part of Figure 24 says that if the object is dry, there is no effect of `Fingers`: the P-value of 0.863 is nowhere near significant. That is, the mean time to complete the task is not significantly different when the fingers are wrinkled or not, if the object is dry.

The second part of the Figure says that if the object is wet, there *is* a significant effect of `Fingers`: the P-value of 0.057 is significant, at $\alpha = 0.10$. That is, when the object is wet, it makes a difference to the mean time to complete the task whether the fingers are wrinkled or not. To see what kind of difference it makes, go back and look at the interaction plot: when the object is wet, the task is completed more quickly when the fingers are wrinkled than when

they are not.

If you did some actual comparisons of means back in part (a), you'll probably find that this part confirms what you guessed from looking at the interaction plot. I wanted you to get all the way to "if the object is wet, the task is completed more quickly when the fingers are wrinkled than when they are not", which is why I directed you back to the interaction plot in the question. If you didn't say that here, I checked back to see whether you said it in (a); if you did, you have credit for it here.

Extras:

These data came from a paper entitled "Water-induced finger wrinkles improve handling of wet objects", and now you see where that title came from.

It might not be clear to you, in working with simple effects, which explanatory variable you hold fixed and which one you allow to vary. My thought process here was to imagine that the objects that come our way are either dry or wet, and this is (I imagine) something out of our control: the experimenters get to decide whether the object we handle is wet or dry, not us. With that in mind, we can think of what strategy we might use once we know whether we are dealing with a wet or a dry object: the one thing we can choose is whether we should wet our fingers first (so that they are wrinkled) or leave them dry (not wrinkled). The strategy we should adopt, revealed by our simple effects, is this: if the object is dry, it doesn't matter what we do, but if the object is wet, we should handle it with wrinkled fingers. (When we were working with the engine noise data in class, I said we didn't really care about engine size, so we held that fixed and let the filter vary, since the filter effect *was* of interest to us. This one is sort of similar.)

I had to stretch a bit to lead you to this conclusion of this question, by making $\alpha = 0.10$ throughout here. If we had stuck with 0.05, we would have come to a different conclusion: the interaction would not have been significant, so we would have removed it, and then we would have had no effect of `Fingers`, but a significant effect of `Objects`, and the latter would have led us to the much less interesting conclusion that, overall, wet objects are more difficult to handle than dry ones (in that it takes longer to complete the task on average if the object is wet). Having said that, though, the conclusions from the two simple effects were *very* different, which suggests that the interaction was worth treating as significant, even though its P-value was not smaller than 0.05.

Use this page if you need more space. Be sure to label any answers here with the question and part they belong to.

Numbered Figures begin here, in with caption and label

```
library(MASS)
library(tidyverse)
library(marginaleffects)
library(survival)
library(survminer)
```

Figure 1: Packages loaded

Glowworm

```
GlowWorms %>% slice_sample(n = 10)
```

Lantern	Eggs
16.5	159
13.5	68
8.1	62
12.3	142
14.4	27
13.3	113
9.9	63
10.2	41
14.9	99
11.8	114

```
GlowWorms %>% summarize(n = n(), mean_Lantern = mean(Lantern), mean_Eggs = mean(Eggs))
```

n	mean_Lantern	mean_Eggs
26	11.9	78.19231

Figure 2: Glowworms data (some randomly chosen rows), with summary statistics (for whole dataset)

```
glow.1 <- lm(Eggs ~ Lantern, data = GlowWorms)
summary(glow.1)
```

```
Call:
lm(formula = Eggs ~ Lantern, data = GlowWorms)

Residuals:
   Min     1Q Median     3Q    Max
-69.50 -23.59  -3.20  22.95  63.33

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.977     21.869  -0.410 0.685087
Lantern        7.325      1.757   4.169 0.000343 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 32.71 on 24 degrees of freedom
Multiple R-squared:  0.4201,    Adjusted R-squared:  0.3959
F-statistic: 17.38 on 1 and 24 DF,  p-value: 0.0003431
```

Figure 3: Glowworms regression

```
new <- datagrid(model = glow.1, Lantern = c(5, 12, 19))
cbind(predictions(glow.1, newdata = new)) %>%
  select(estimate, conf.low, conf.high, Lantern)
```

estimate	conf.low	conf.high	Lantern
27.64885	0.768477	54.52923	5
78.92482	66.348540	91.50110	12
130.20079	102.709804	157.69178	19

Figure 4: Glowworms predictions 1

```
p <- predict(glow.1, new, interval = "p")
cbind(new, p)
```

Eggs	Lantern	fit	lwr	upr
78	5	27.64885	-45.54768	100.8454
78	12	78.92482	10.13605	147.7136
78	19	130.20079	56.75322	203.6484

Figure 5: Glowworms predictions 2, using same `new` as in previous Figure

Kinase

```
heart
```

mck	ha	nha
20	2	88
60	13	26
100	30	8
140	30	5
180	21	0
220	19	1
260	18	1
300	13	1
340	19	0
380	15	0
420	7	0
460	8	0
500	35	0

Figure 6: Creatinine kinase data

```
heart %>% select(ends_with("ha")) %>%
  as.matrix() -> response
response
```

```
      ha nha
 [1,]  2  88
 [2,] 13  26
 [3,] 30   8
 [4,] 30   5
 [5,] 21   0
 [6,] 19   1
 [7,] 18   1
 [8,] 13   1
 [9,] 19   0
[10,] 15   0
[11,]  7   0
[12,]  8   0
[13,] 35   0
```

Figure 7: Creatinine kinase data: some code and its output

```r
heart.1 <- glm(response ~ mck, data = heart, family = "binomial")
summary(heart.1)
```

```
Call:
glm(formula = response ~ mck, family = "binomial", data = heart)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.028360   0.366977  -8.252   <2e-16 ***
mck          0.035104   0.004081   8.602   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 311.29  on 12  degrees of freedom
Residual deviance:  28.14  on 11  degrees of freedom
AIC: 51.596

Number of Fisher Scoring iterations: 6
```

Figure 8: Creatinine kinase logistic regression

```
new <- tibble(mck = seq(0, 300, 20))
cbind(predictions(heart.1, newdata = new)) %>%
  select(estimate, conf.low, conf.high, mck) %>%
  ggplot(aes(x = mck, y = estimate, ymin = conf.low, ymax = conf.high)) +
  geom_line()  + geom_ribbon(alpha = 0.2)
```

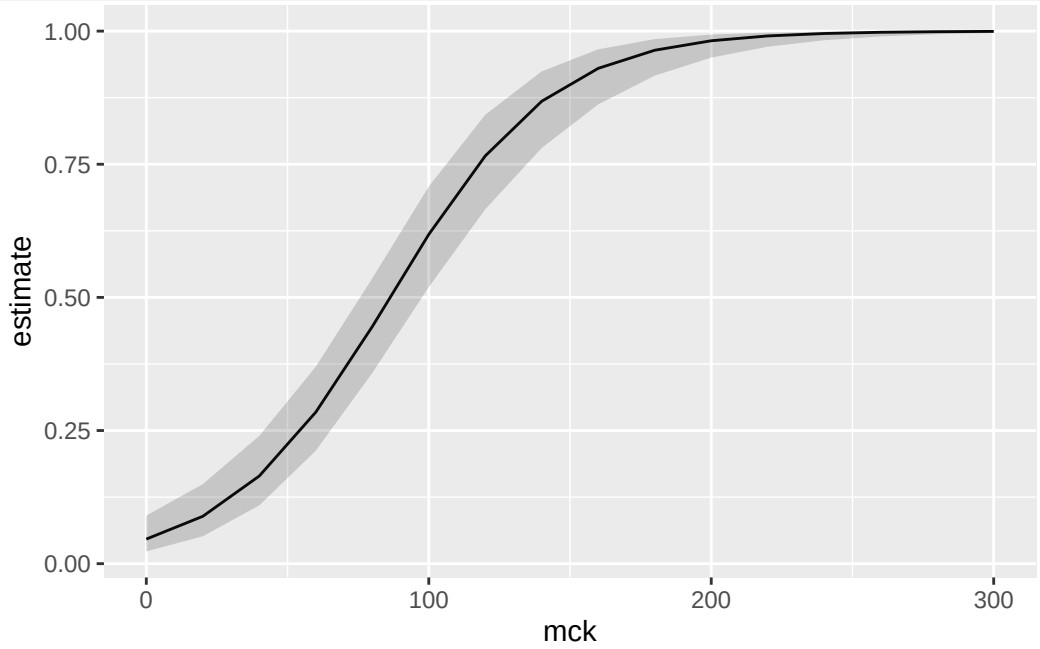


Figure 9: Creatinine kinase model predictions

```
new2 <- tibble(mck = c(20, 40, 60, 80))
cbind(predictions(heart.1, newdata = new2, type = "link")) %>%
  select(estimate, mck)
```

estimate	mck
-2.3262717	20
-1.6241839	40
-0.9220960	60
-0.2200082	80

Figure 10: Creatinine kinase data: more predictions

Faces

```
Faces %>% slice_sample(n = 20)
```

MaxGripStrength	SHR	Attractive
51.5	1.464	D
50.5	1.218	A
44.0	1.352	B
46.5	1.363	D
35.0	1.099	B
50.5	1.127	D
55.0	1.315	B
34.0	1.247	B
58.5	1.428	D
47.0	1.342	C
49.0	1.136	A
41.0	1.322	D
44.5	1.333	B
42.5	1.310	D
45.5	1.436	C
49.5	1.284	A
43.5	1.249	A
54.0	1.363	D
58.0	1.349	B
37.0	1.377	C

Figure 11: Attractiveness data (random sample of rows)

```
Faces.1 <- polr(Attractive ~ MaxGripStrength + SHR + I(SHR^2), data = Faces)
```

Figure 12: Model for attractiveness data

```
drop1(Faces.1, test = "Chisq")
```

	Df	AIC	LRT	Pr(>Chi)
	NA	102.8388	NA	NA
MaxGripStrength	1	100.9271	0.088272	0.7663855
SHR	1	102.4722	1.633377	0.2012366
I(SHR^2)	1	102.9592	2.120328	0.1453554

Figure 13: `drop1` output for attractiveness data model

```
new <- datagrid(model = Faces.1, SHR = c(1, 1.2, 1.4, 1.6), MaxGripStrength = c(42, 54))
cbind(predictions(Faces.1, newdata = new)) %>%
  select(group, estimate, SHR, MaxGripStrength) %>%
  pivot_wider(names_from = group, values_from = estimate)
```

```
Re-fitting to get Hessian
```

SHR	MaxGripStrength	A	B	C	D
1.0	42	0.2037797	0.3097473	0.2532528	0.2332202
1.0	54	0.1952400	0.3049208	0.2569195	0.2429197
1.2	42	0.5097292	0.3011724	0.1194417	0.0696567
1.2	54	0.4963609	0.3062037	0.1242323	0.0732031
1.4	42	0.1388698	0.2605816	0.2749903	0.3255583
1.4	54	0.1325973	0.2540969	0.2758976	0.3374082
1.6	42	0.0009542	0.0029696	0.0081966	0.9878797
1.6	54	0.0009045	0.0028157	0.0077763	0.9885035

Figure 14: Predictions of attractiveness for various values of shoulder-hip ratio and grip strength

larynx

```
larynx %>% slice(1:20)
```

stage	time	age	diagyr	delta
stage1	0.6	77	76	dead
stage1	1.3	53	71	dead
stage1	2.4	45	71	dead
stage1	2.5	57	78	alive
stage1	3.2	58	74	dead
stage1	3.2	51	77	alive
stage1	3.3	76	74	dead
stage1	3.3	63	77	alive
stage1	3.5	43	71	dead
stage1	3.5	60	73	dead
stage1	4.0	52	71	dead
stage1	4.0	63	76	dead
stage1	4.3	86	74	dead
stage1	4.5	48	76	alive
stage1	4.5	68	76	alive
stage1	5.3	81	72	dead
stage1	5.5	70	75	alive
stage1	5.9	58	75	alive
stage1	5.9	47	75	alive
stage1	6.0	75	73	dead

Figure 15: Larynx cancer data (some)

```
 [1] 0.6  1.3  2.4  2.5+ 3.2  3.2+ 3.3  3.3+ 3.5  3.5  4.0  4.0  4.3  4.5+ 4.5+
[16] 5.3  5.5+ 5.9+ 5.9+ 6.0
```

Figure 16: Larynx cancer: some values of y

```
larynx.1 <- coxph(y ~ stage + age + diagyr, data = larynx)
summary(larynx.1)
```

```
Call:
coxph(formula = y ~ stage + age + diagyr, data = larynx)

  n= 90, number of events= 50

               coef exp(coef) se(coef)      z Pr(>|z|)
stagestage2  0.15164   1.16375  0.46481  0.326   0.7442
stagestage3  0.64473   1.90546  0.35619  1.810   0.0703 .
stagestage4  1.73211   5.65255  0.43596  3.973 7.09e-05 ***
age          0.01869   1.01887  0.01433  1.304   0.1922
diagyr      -0.01819   0.98198  0.07646 -0.238   0.8120
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

            exp(coef) exp(-coef) lower .95 upper .95
stagestage2     1.164     0.8593    0.4680     2.894
stagestage3     1.905     0.5248    0.9480     3.830
stagestage4     5.653     0.1769    2.4052    13.284
age             1.019     0.9815    0.9906     1.048
diagyr          0.982     1.0184    0.8453     1.141

Concordance= 0.674  (se = 0.039 )
Likelihood ratio test= 18.37  on 5 df,   p=0.003
Wald test            = 21.2  on 5 df,   p=7e-04
Score (logrank) test = 24.84  on 5 df,   p=1e-04
```

```
drop1(larynx.1, test = "Chisq")
```

	Df	AIC	LRT	Pr(>Chi)
	NA	385.3583	NA	NA
stage	3	394.8735	15.515268	0.0014253
age	1	385.0995	1.741195	0.1869875
diagyr	1	383.4147	0.056456	0.8121876

Figure 17: Larynx cancer: Cox model 1

```
larynx.2 <- coxph(y ~ stage, data = larynx)
summary(larynx.2)
```

```
Call:
coxph(formula = y ~ stage, data = larynx)

  n= 90, number of events= 50

               coef exp(coef) se(coef)     z Pr(>|z|)
stagestage2 0.06481   1.06696  0.45843 0.141   0.8876
stagestage3 0.61481   1.84930  0.35519 1.731   0.0835 .
stagestage4 1.73490   5.66838  0.41939 4.137 3.52e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

            exp(coef) exp(-coef) lower .95 upper .95
stagestage2     1.067     0.9372    0.4344      2.62
stagestage3     1.849     0.5407    0.9219      3.71
stagestage4     5.668     0.1764    2.4916     12.90

Concordance= 0.668  (se = 0.037 )
Likelihood ratio test= 16.49  on 3 df,   p=9e-04
Wald test            = 19.24  on 3 df,   p=2e-04
Score (logrank) test = 22.88  on 3 df,   p=4e-05
```

```
drop1(larynx.2, test = "Chisq")
```

	Df	AIC	LRT	Pr(>Chi)
	NA	383.2416	NA	NA
stage	3	393.7270	16.48538	0.0009016

Figure 18: Larynx cancer: Cox model 2

```
anova(larynx.2, larynx.1)
```

loglik	Chisq	Df	Pr(>|Chi|)
-188.6208	NA	NA	NA
-187.6791	1.883308	2	0.3899823

Figure 19: Larynx cancer: further analysis

```
larynx %>% count(stage) -> new
new
```

stage	n
stage1	33
stage2	17
stage3	27
stage4	13

```
s <- survfit(larynx.2, new, data = larynx)
ggsurvplot(s, conf.int = FALSE)
```

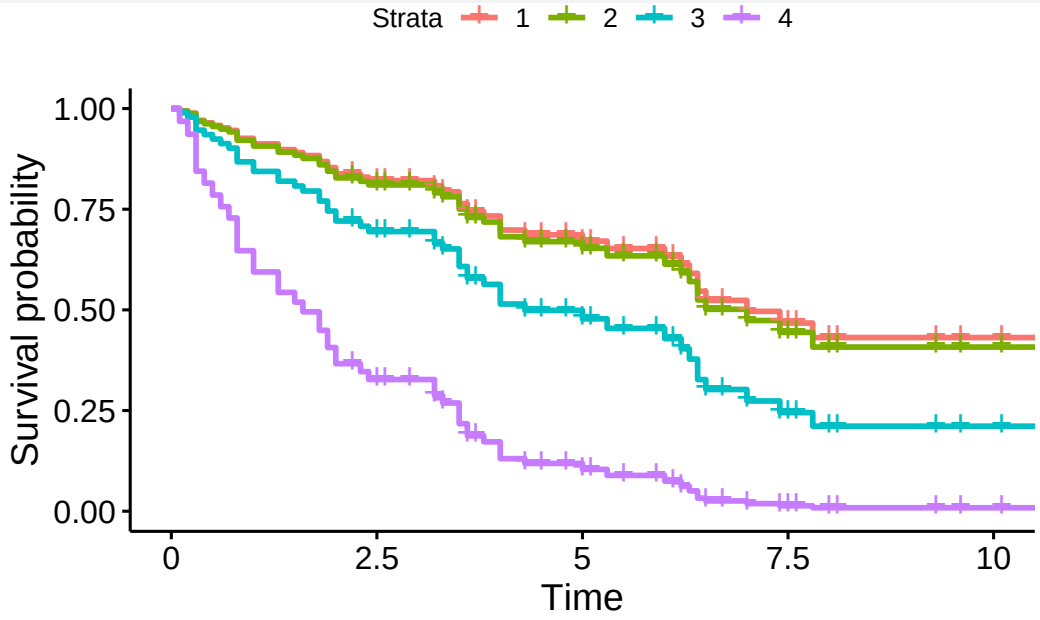


Figure 20: Larynx cancer: predicted survival curves

Wrinkle

```
Wrinkle
```

Time	Fingers	Objects
106	non	dry
113	non	dry
94	non	dry
96	non	dry
93	non	dry
123	non	dry
76	non	dry
92	non	dry
87	non	dry
79	non	dry
82	non	dry
104	non	dry
73	non	dry
95	non	dry
77	non	dry
92	non	dry
97	non	dry
79	non	dry
120	non	dry
88	non	dry
139	non	wet
138	non	wet
136	non	wet
101	non	wet
108	non	wet
143	non	wet
88	non	wet
109	non	wet
99	non	wet
93	non	wet
96	non	wet
156	non	wet
98	non	wet
116	non	wet
102	non	wet
117	non	wet
105	non	wet
93	non	wet
198	non	wet
123	non	wet
107	wrinkled	dry
97	wrinkled	dry
117	wrinkled	dry
95	wrinkled	dry
104	wrinkled	dry
108	wrinkled	dry
83	wrinkled	dry
83	wrinkled	dry

```
Wrinkle %>%
  group_by(Fingers, Objects) %>%
  summarize(mean_time = mean(Time)) -> Wrinkle_means
```

```
`summarise()` has grouped output by 'Fingers'. You can override using the
`.groups` argument.
```

```
Wrinkle_means
```

Fingers	Objects	mean_time
non	dry	93.30
non	wet	117.90
wrinkled	dry	94.15
wrinkled	wet	102.85

```
ggplot(Wrinkle_means, aes(x = Objects, y = mean_time, colour = Fingers, group = Fingers)) +
  geom_point() + geom_line()
```

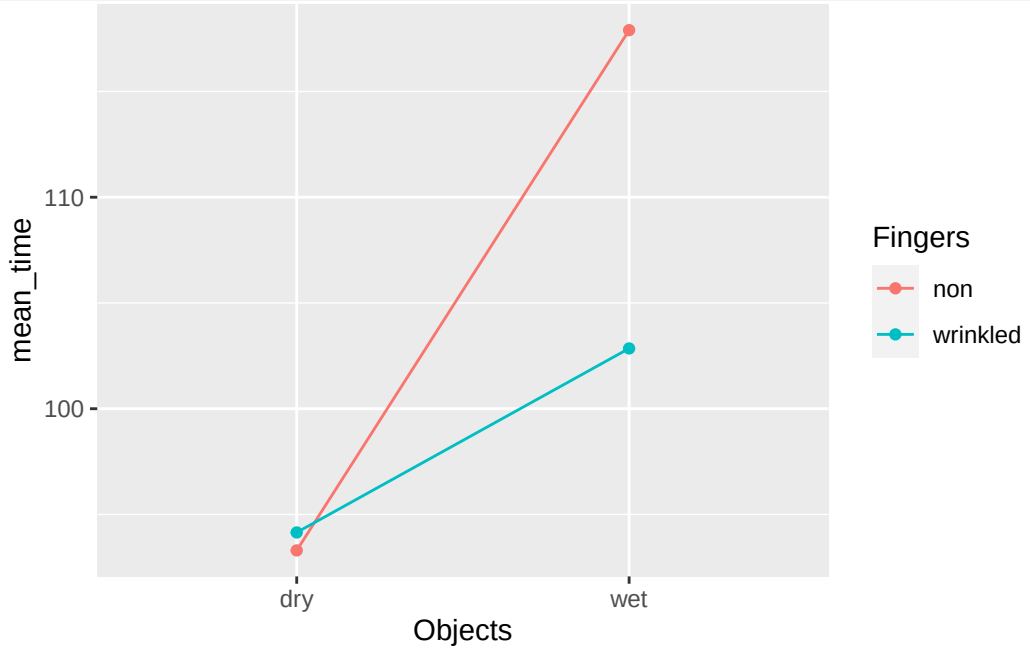


Figure 22: Wrinkled fingers plot

```
Wrinkle.1 <- aov(Time ~ Fingers * Objects, data = Wrinkle)
summary(Wrinkle.1)
```

```
               Df Sum Sq Mean Sq F value   Pr(>F)
Fingers         1   1008    1008   2.447 0.121939
Objects         1   5544    5544  13.454 0.000451 ***
Fingers:Objects 1   1264    1264   3.067 0.083912 .
Residuals      76  31319     412
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 23: Wrinkled fingers analysis

```
Wrinkle %>%
  filter(Objects == "dry") -> drys
drys.1 <- aov(Time ~ Fingers, data = drys)
summary(drys.1)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
Fingers      1      7    7.23    0.03  0.863
Residuals   38   9035  237.76
```

```
Wrinkle %>%
  filter(Objects == "wet") -> wets
wets.1 <- aov(Time ~ Fingers, data = wets)
summary(wets.1)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
Fingers      1   2265  2265.0   3.862 0.0567 .
Residuals   38  22284   586.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 24: Wrinkled fingers analysis continued