ST523 – Statistical Modelling

Answers to the Take-home 2 Assignment, Winter 2025, Jan Ryszkiewicz

Task 1

subtask 1

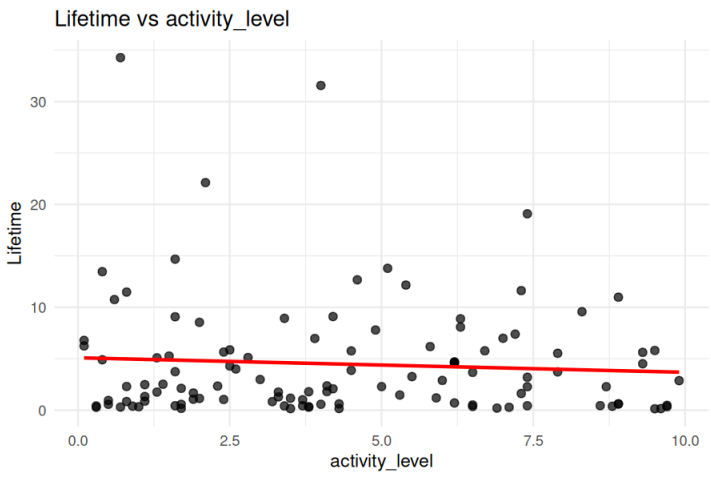
The dataset contains 113 observations across 9 variables. Summary statistics are presented below:

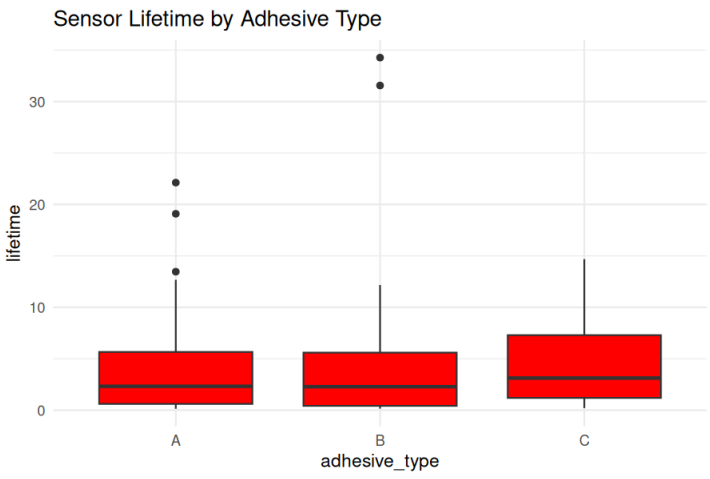
```
    lifetime         skin_temp        humidity     activity_level
 Min.   : 0.140   Min.   :31.10   Min.   :20.70   Min.   :0.100
 1st Qu.: 0.610   1st Qu.:32.70   1st Qu.:39.00   1st Qu.:1.700
 Median : 2.340   Median :34.30   Median :56.10   Median :4.000
 Mean   : 4.478   Mean   :34.18   Mean   :54.06   Mean   :4.345
 3rd Qu.: 5.800   3rd Qu.:35.70   3rd Qu.:71.80   3rd Qu.:6.500
 Max.   :34.270   Max.   :37.00   Max.   :83.00   Max.   :9.900
   sweat_rate    calibration_error  patient_bmi      experience
 Min.   : 2.3   Min.   : 0.30     Min.   :15.60   Min.   :1.000
 1st Qu.:46.8   1st Qu.: 5.80     1st Qu.:24.70   1st Qu.:3.000
 Median :58.4   Median :12.90     Median :27.60   Median :4.000
 Mean   :57.5   Mean   :19.27     Mean   :27.61   Mean   :3.965
 3rd Qu.:69.4   3rd Qu.:27.60     3rd Qu.:31.20   3rd Qu.:5.000
 Max.   :99.5   Max.   :78.30     Max.   :39.60   Max.   :9.000
```

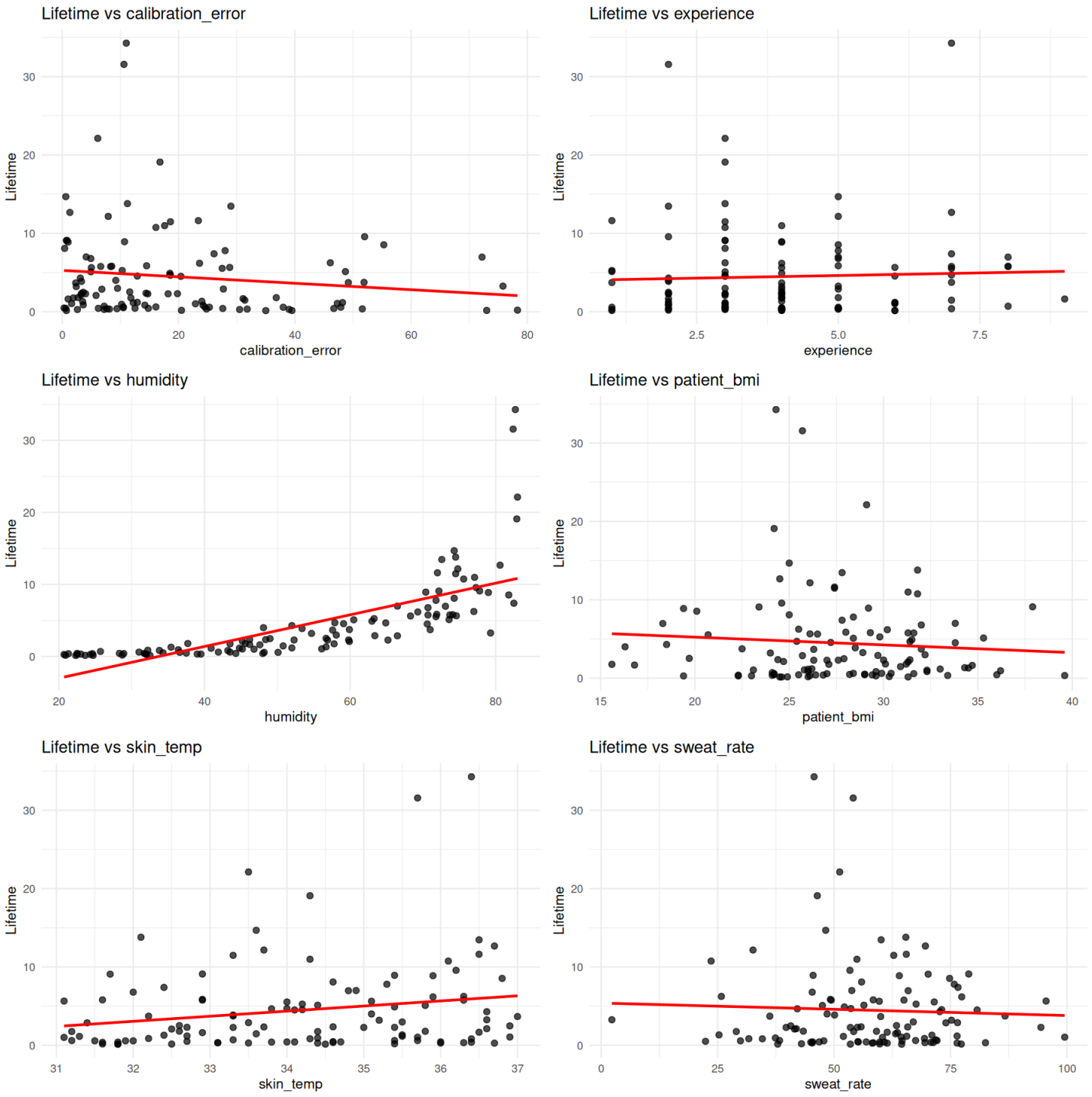
For the adhesive, as it is a factor variable, we only show the overall amount of data entries per type:

```
 A  B  C
52 35 26
```

Relationship with Lifetime plots:







It is worth noting that on the plots we can see 4 points which significantly higher lifetime stands out from the rest

---

---

subtask 2

Since lifetime is strictly positive and continuous, I considered two GLM families:

- Gamma distribution with log and canonical (inverse) links
- Inverse Gaussian distribution with log link

Log link function is chosen as the relationship between lifetime and humidity seems exponential - other continuous predictors appear linear on the plots. The canonical link for Inverse Gaussian caused convergence issues, so it was excluded.

I do not consider Poisson distribution as it is designed for count data, not continuous responses. Binomial on the other hand is mainly used for proportion data which is not present in our example.

We compare candidate models based on AIC:

```
       Gamma_Log Inv_Gaussian_Log  Gamma_Canonical       Normal_Log
        296.7400         376.2993         423.0153         468.5684
          Normal
        643.3063
```

and residual deviance:

```
Inv_Gaussian_Log         Gamma_Log  Gamma_Canonical       Normal_Log
        16.71100          16.83515          49.09622         344.27608
          Normal
      1616.14149
```

The Gamma distribution with log linking was selected based on lowest AIC. Even though Inverse Gaussian had slightly lower residual deviance, the substantial AIC difference ($\Delta \approx 80$) and the excessive cubic mean-variance relationship (μ^3) were not necessary given the visually observed variance structure on the plots. As a test I also tried fitting Normal (Gaussian) model with both log link and identity (where the second case is equal to fitting just *normal linear model*). Those two however perforemd significantly worse compared to Gamma or Inverse Gaussian fits.

Full fit model can be tested for interaction using `(...)^2` syntax, I performed sensible pairwise tests for the predictors including their interaction terms, ie. relating `activity_level` to, for instance, `patient_bmi`, `sweat_rate` and `humidity`, relating `experience:humidity` or `experience:calibration_error` etc. Those comparisons yielded no statistically significant interaction terms between the majority of predictors (interaction terms p-values were » 0.05). **However when testing individual interaction between each predictor and `adhesive_type` I noticed the existence of a significant interaction between `patient_bmi` and the aformentioned `adhesive_type`, what motivates the inclusion of this term in my model.**

We consider 2 different predictor selection methods:

- Stepwise, based on AIC using `step()`, on default model with included `patient_bmi` interaction:

```r
Model_1 <- step(glm(lifetime ~ skin_temp + humidity + activity_level +
                      sweat_rate + calibration_error +
                      experience + patient_bmi * adhesive_type,
                    data = df,
                    family = Gamma(link = "log")), trace=0)
```

- P-value based selection (threshold: $p < 0.05$):

```
                           Estimate Std. Error  t value Pr(>|t|)
(Intercept)                -3.96381    0.96503 -4.10746  0.00008
skin_temp                   0.06108    0.02342  2.60771  0.01050
humidity                    0.06645    0.00208 32.00586  0.00000
activity_level             -0.04027    0.01343 -2.99882  0.00341
sweat_rate                  0.00112    0.00236  0.47682  0.63452
calibration_error          -0.01492    0.00212 -7.04694  0.00000
experience                  0.01808    0.02260  0.79968  0.42577
patient_bmi                -0.02418    0.01216 -1.98889  0.04942
adhesive_typeB             -0.21996    0.55061 -0.39948  0.69038
adhesive_typeC             -1.00510    0.59047 -1.70220  0.09179
patient_bmi:adhesive_typeB  0.01197    0.01996  0.59984  0.54996
patient_bmi:adhesive_typeC  0.04333    0.02073  2.09055  0.03908
```

Initial full model coefficients were deemed:

- Significant ($p < 0.05$): `humidity`, `calibration_error`, `skin_temp`, `activity_level`, `patient_bmi`
- Non-significant : `sweat_rate` (p=0.635), `experience` (p=0.426),
- Adhesive type C showed significance by interaction with bmi during testing, therefore all cathegorical predictors of `adhesive_type` retained.

Comparison between two models:

- Model 1 - step() best on default model with included `patient_bmi` interaction.
- Model 2 - p-value based choice on default model with included `patient_bmi` interaction.

Is not needed as after the stepwise model selection based on Akaike Information Criterion and selection based on p values we end up with the same model, different in one predictor. After that I decided to try running `step()` command on a model with all pairwise interactions (`(...)^2` for all predictors). It returend a model with lower AIC (~19 lower), but it included all predictors and a lot of additional interactions (additional 18 (!) terms) that were less interpretable. In the following tasks I will be using the parsimonious Model 2 (equal to Model 1) for clarity and meaningful interpretation, balancing it with the quality of fit.

---

subtask 3

Final model is chosen to be Gamma GLM with log link with following predictors:

```
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                -3.5105     0.8666 -4.0509   0.0001
skin_temp                   0.0528     0.0222  2.3737   0.0195
humidity                    0.0666     0.0020 32.6798   0.0000
activity_level             -0.0373     0.0130 -2.8742   0.0049
calibration_error          -0.0147     0.0021 -7.0385   0.0000
patient_bmi                -0.0263     0.0119 -2.2086   0.0294
adhesive_typeB             -0.2300     0.5471 -0.4204   0.6751
adhesive_typeC             -0.9801     0.5860 -1.6727   0.0974
patient_bmi:adhesive_typeB  0.0126     0.0198  0.6360   0.5262
patient_bmi:adhesive_typeC  0.0421     0.0205  2.0481   0.0431
```

Since experience is not in the final model, it does not not affect the prediction. Using patient_bmi = 30, activity_level=4 and sample means for remaining predictors (`adhesive_type` = A as reference):

```
(Intercept)
   1.961085
```

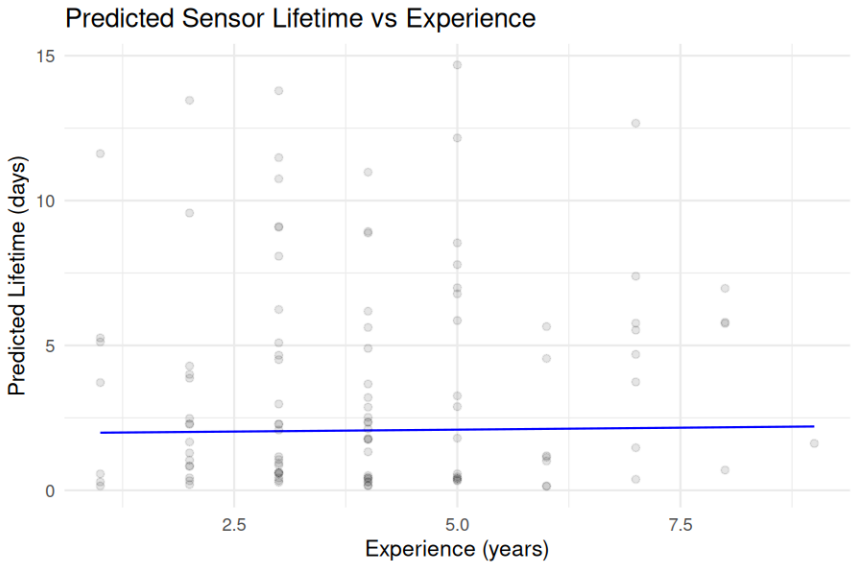
calculated by:

$$\hat{\mu} = \exp(\beta_0 + \beta_1 \cdot x_{\text{skin temp}} + \beta_2 \cdot x_{\text{humidity}} + \beta_3 \cdot x_{\text{activity level}} + \beta_4 \cdot x_{\text{calibration error}} + \beta_5 \cdot x_{\text{patient bmi}} + \ldots)$$

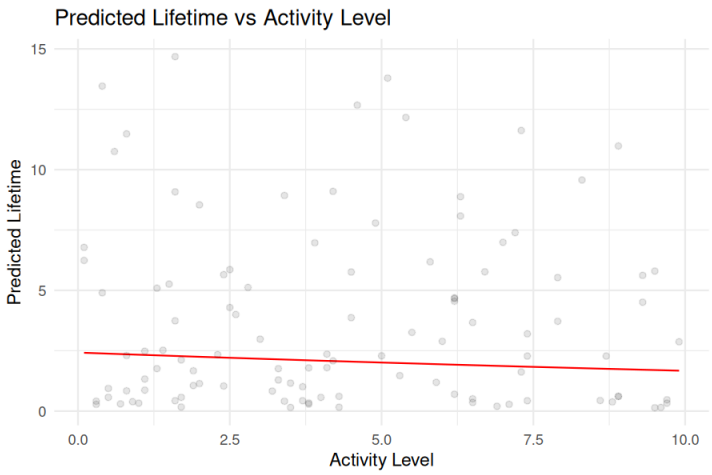
Where "..." normally represent the `adhesive_type` and interaction terms between `patien_bmi` and `adhesive_type`, however in our case they are all set to 0, because we use adhesive type A, which is the reference category in R (and also the most common type in the data).

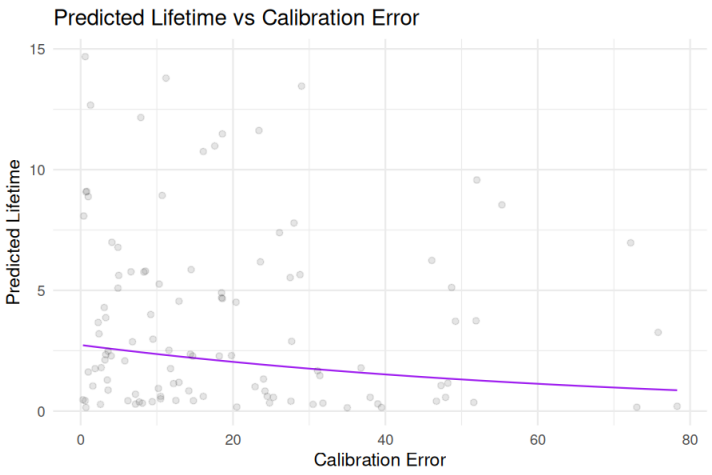
Since experience was removed from the final model, changing experience by 1 year produces no significant change in expected lifetime.

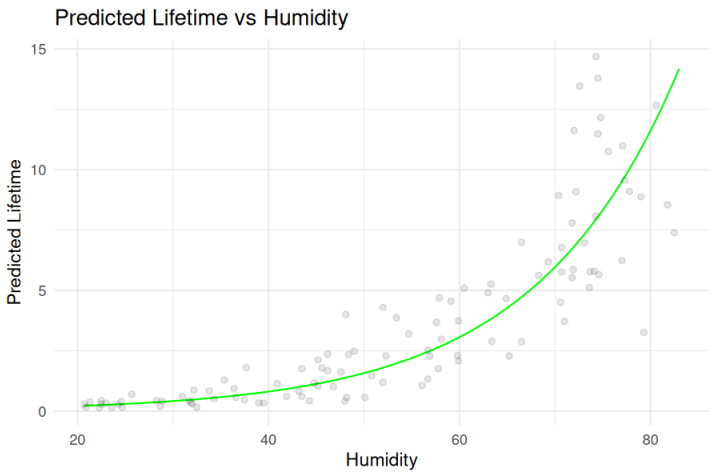
Plots presented here deliberatly cut off the lifetime >15 outlier points (only 4 of them) to increase visibility. As experience is not included as predictor in our final model following plot is made using default, full-fit model including all predictors.

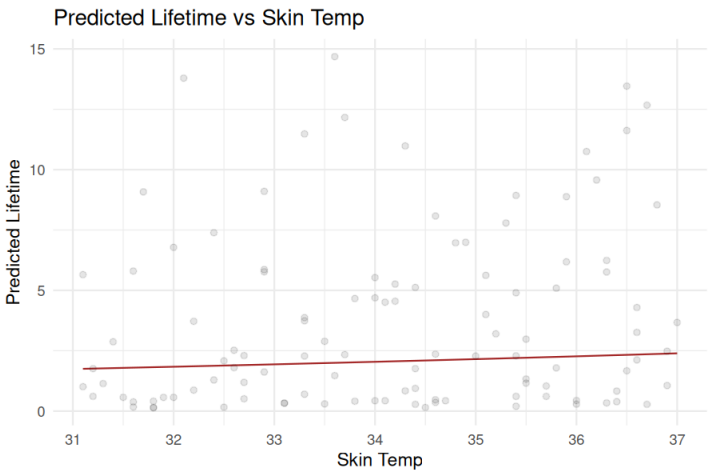


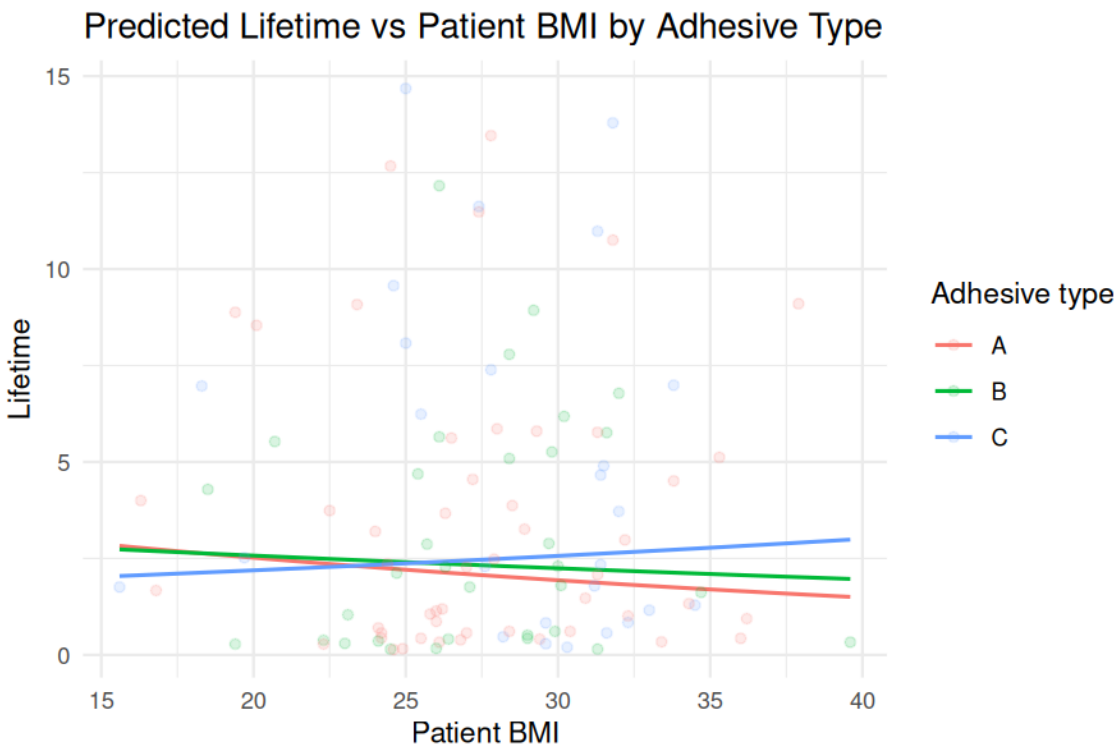
As we can see, user experience does not seem to impact the liftime in a meaningfull way compared to our chosen predictors, which effect is clearly visible on the following plots:











Here we can clearly see why the inclusion of the interaction term is necessary. The effect of adhesive type on sensor lifetime varies with patient BMI, as indicated by the differing slopes of the predicted curves.

Final Model Deviance:

```
[1] 16.29032
```

Final Model AIC:

```
[1] 292.9319
```

Final Model's Dispersion parameter is:

```
summary(final_model)$dispersion


[1] 0.151927
```

As Gamma belongs to *Exponential Dispersion Model* family it does not really make sense to talk about over- or underdispersion as its second parameter captures the additional changes in variation (well, that what EDMs are for). In models with only one parameter this value of $\sigma^2 \approx 0.15$ would signify underdispersion.

Sensor lifetime is significantly influenced by environmental and technical factors: higher `skin_temperature`and higher `humidity` increases lifetime, while higher `calibration_error` and `activity_level` decrease it. `Humidity` shows the strongest effect, followed by `calibration_error`, while `skin temperature` and `activity level` have more modest impacts. The data points strongly towards an interaction between `patien_bmi` and `adhesive_type` meaning that their specific combination influences the results in a different way. `User experience`, along with their `sweat rate`, do not significantly impact sensor lifetime and were excluded from the final model.

Task 2

Because our transformation g_α is monotonically increasing (trivial checks for $y \geq 0$ and $y < 0$)

$$P(Y_i < y_i) = P(g_\alpha(Y_i) < g_\alpha(y_i))$$

And since $g_\alpha(Y_i)$ is normally distributed, we can use the standard normal CDF Φ with normalized Y_i with its mean and standard error:

$$P(Y_i < y_i) = \Phi\left(\frac{g_\alpha(y_i) - \mathbf{x}_i^\top \beta}{\sigma}\right)$$

Taking the derivative with respect to y_i gives us:

$$\frac{d}{dy_i} P(Y_i < y_i) = \phi\left(\frac{g_\alpha(y_i) - \mathbf{x}_i^\top \beta}{\sigma}\right) \cdot \frac{\dot{g}_\alpha(y_i)}{\sigma}$$

where ϕ is the standard normal density and $\dot{g}_\alpha(y_i)$ being the derivative of $g_\alpha(y_i)$ defined as:

$$\dot{g}_\alpha(y_i) = \begin{cases} (y_i + 1)^{\alpha - 1}, & y \geq 0 \\ (1 - y_i)^{1-\alpha}, & y < 0 \end{cases}$$

The final density of Y_i is:

$$\begin{aligned} f_{Y_i}(y_i) =& \phi\left(\frac{g_\alpha(y_i) - \mathbf{x}_i^\top \beta}{\sigma}\right) \cdot \frac{\dot{g}_\alpha(y_i)}{\sigma} = \\ =& \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(\frac{g_\alpha(y_i) - \mathbf{x}_i^\top \beta}{\sigma}\right)^2 \right\} \cdot \frac{\dot{g}_\alpha(y_i)}{\sigma} \end{aligned}$$

$$\ell(\alpha, \beta, \sigma^2; y_1, \ldots, y_n) = \sum_{i=1}^{n} \log f_{Y_i}(y_i)$$

For independent Y_i with density f_{Y_i}, this becomes:

$$\ell(\alpha, \beta, \sigma^2; y_1, \ldots, y_n) = \log\left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(\frac{g_\alpha(y_i) - \mathbf{x}_i^\top \beta}{\sigma}\right)^2 \right\} \cdot \frac{\dot{g}_\alpha(y_i)}{\sigma} \right]$$

The estimates derived from equating the derivative of loglikelihood with respect to β and σ to 0 (as we did on the lecture) are:

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top g_\alpha(\mathbf{Y})$$

$$\hat{\sigma}^2 = \frac{1}{n} RSS_g$$

Where

$$RSS_g = \sum_{i=1}^{n} \left(g_\alpha(y_i) - \mathbf{x}_i^\top \hat{\beta} \right)^2$$

$$l(\alpha; y_1, \dots, y_n) = \log \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{g_\alpha(y_i) - \mathbf{x}_i^\top \hat{\beta}}{\hat{\sigma}} \right)^2 \right\} \cdot \frac{\dot{g}_\alpha(y_i)}{\hat{\sigma}} \right]$$

Which equals to:

$$= -\frac{n}{2} \log(2\pi) - \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{n} \left(g_\alpha(y_i) - \mathbf{x}_i^\top \hat{\beta} \right)^2 + \sum_{i=1}^{n} \log \dot{g}_\alpha(y_i) - n \log(\hat{\sigma})$$

Reduced by substituting $RSS_g = \sum_{i=1}^{n} \left(g_\alpha(y_i) - \mathbf{x}_i^\top \hat{\beta} \right)^2$ and $\hat{\sigma}^2 = \frac{1}{n} RSS_g$:

$$-\frac{n}{2} \log(2\pi) - \frac{n}{2} + \sum_{i=1}^{n} \log \dot{g}_\alpha(y_i) - n \log(\sqrt{\frac{RSS_g}{n}})$$

And further into:

$$l(\alpha; y_1, \dots, y_n) = \sum_{i=1}^{n} \log \dot{g}_\alpha(y_i) - \frac{n}{2} \log(\frac{RSS_g}{n}) + const.$$

I decided to write g_α and $\dot{g}_\alpha$ in such way so that they can be aplied vector-wise and not element-wise. Implementation of all the functions in R looks as follows:

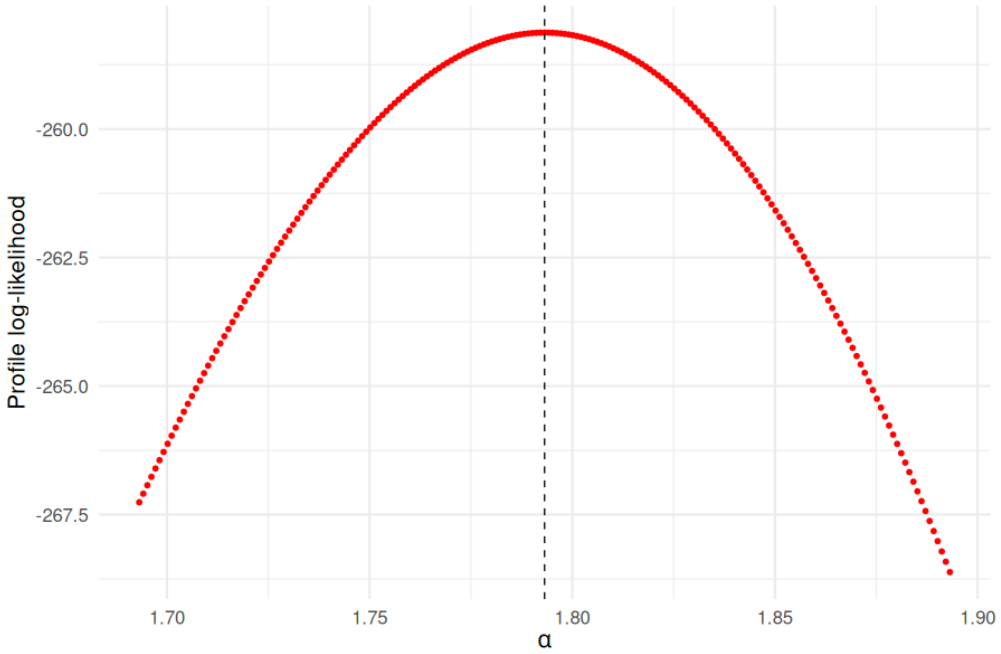
```r
g = function(y, alpha) {
  out <- numeric(length(y))

  # case 1: y >= 0, alpha != 0
  in_1 = (y >= 0 & alpha != 0)
  out[in_1] = (((y[in_1] + 1)^(alpha - 1)) / alpha)

  # case 2: y >= 0, alpha == 0
  in_2 = (y >= 0 & alpha == 0)
  out[in_2] = log(y[in_2] + 1)

  # case 3: y < 0, alpha != 2
  in_3 = (y < 0 & alpha != 2)
  out[in_3] = ( - ((((-y[in_3] + 1)^(2 - alpha)) - 1) / (2 - alpha)))

  # case 4: y < 0, alpha == 2
  in_4 = (y < 0 & alpha == 2)
  out[in_4] = ( - log(-y[in_4] + 1))

  return(out)
}
```

```r
g_prim = function(y, alpha) {
  out <- numeric(length(y))

  # case 1: y >= 0
  in_1 = (y >= 0)
  out[in_1] = (y[in_1] + 1)^(alpha - 1)

  # case 2: y < 0
  in_2 = (y < 0)
  out[in_2] = (1 - y[in_2])^(1 - alpha)

  return(out)
}

profile_log_likelihood = function(Y, X, alpha) {
  n = length(Y)

  # transformed Y
  g_Y <- g(Y, alpha)

  # B_hat estimate
  B_hat = solve(t(X) %*% X) %*% t(X) %*% g_Y

  # Residual Sum of Squares
  RSS = sum((g_Y - X %*% B_hat)^2)

  # Constants
  constants = (-n/2)*log(2*pi) - (n/2)

  # profile log likehood = sum of log(g_prim(y)) -n/2 log(sigma_hat^2) + constants
  out = sum(log(g_prim(Y, alpha))) - ((n/2)*log((1/n) * RSS)) + constants

  return(out)
}
```

We then load the data, add intercept term and find the ML estimate using `optim()`

```r
load("Data_Assignment2_Ex2_E2025.rdata")
df_x <- cbind(1, x)
df_y <- Y

out <- optim(
  par = 0,
  fn = function(a) {-profile_log_likelihood(df_y, df_x, a)}
)
```

We plot the values of loglik funciton spread in $\pm$ 0.1 around from `out$par` (which is the result of the optimization) with step of 0.001:



As we can see $\hat{\alpha} \approx 1.8$

Final transformation is then (by substituting $\alpha = 1.8$):

$$g_\alpha(Y) = \begin{cases} \dfrac{(Y+1)^{1.8} - 1}{1.8}, & y \geq 0 \\ -\dfrac{(-Y+1)^{0.2} - 1}{0.2}, & y < 0 \end{cases}$$