

# Take Home Assignment 1

Jan Ryszkiewicz

## Task 1

### subtask 1

a)

```
load("Data_Assignment2_Ex1_E2025.rdata")
df <- cgm_data
summary(df[, sapply(df, is.numeric)])
```

	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

```
table(df$adhesive_type)
```

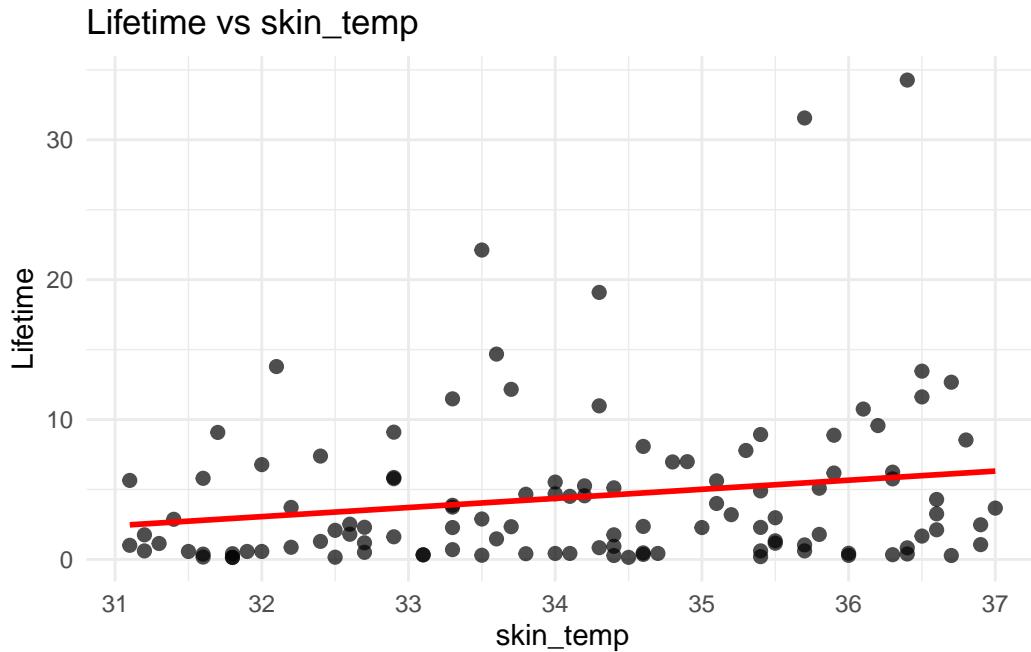
A	B	C
52	35	26

b)

```
library(ggplot2)
numeric_vars <- c("skin_temp", "humidity", "activity_level",
                  "sweat_rate", "calibration_error", "patient_bmi", "experience")

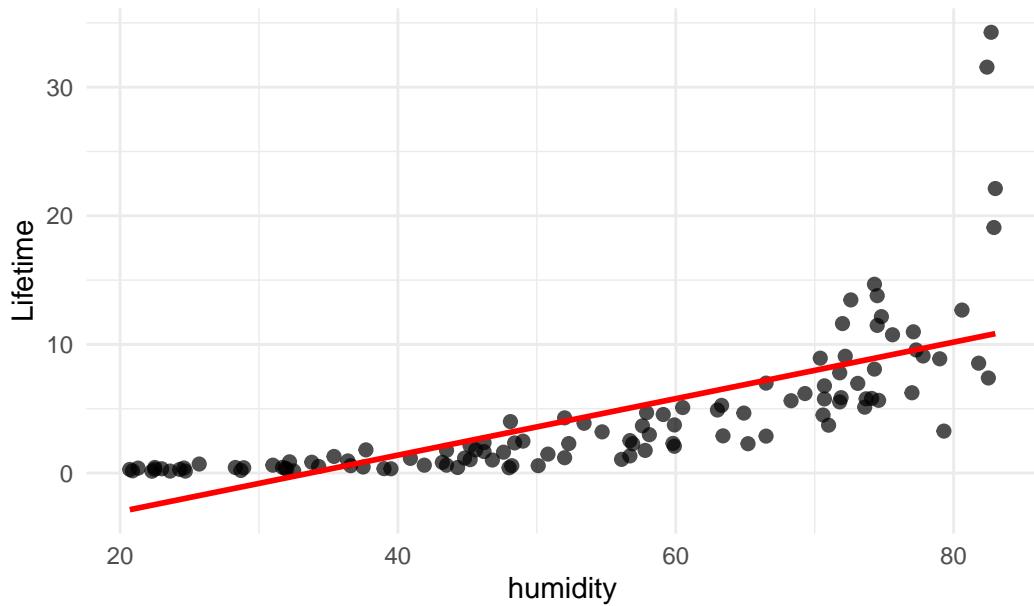
for (var in numeric_vars) {
  p <- ggplot(df, aes(x = .data[[var]], y = lifetime)) +
    geom_point(size = 2, alpha = 0.7) +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    theme_minimal() +
    labs(title = paste("Lifetime vs", var),
        x = var,
        y = "Lifetime")
  print(p)
}
```

`geom\_smooth()` using formula = 'y ~ x'



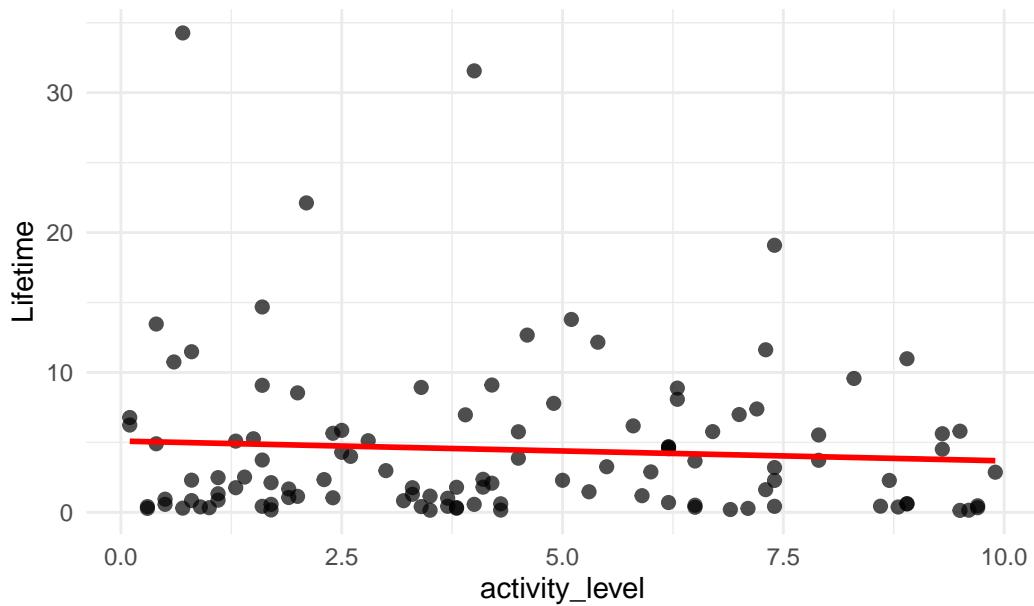
`geom\_smooth()` using formula = 'y ~ x'

Lifetime vs humidity



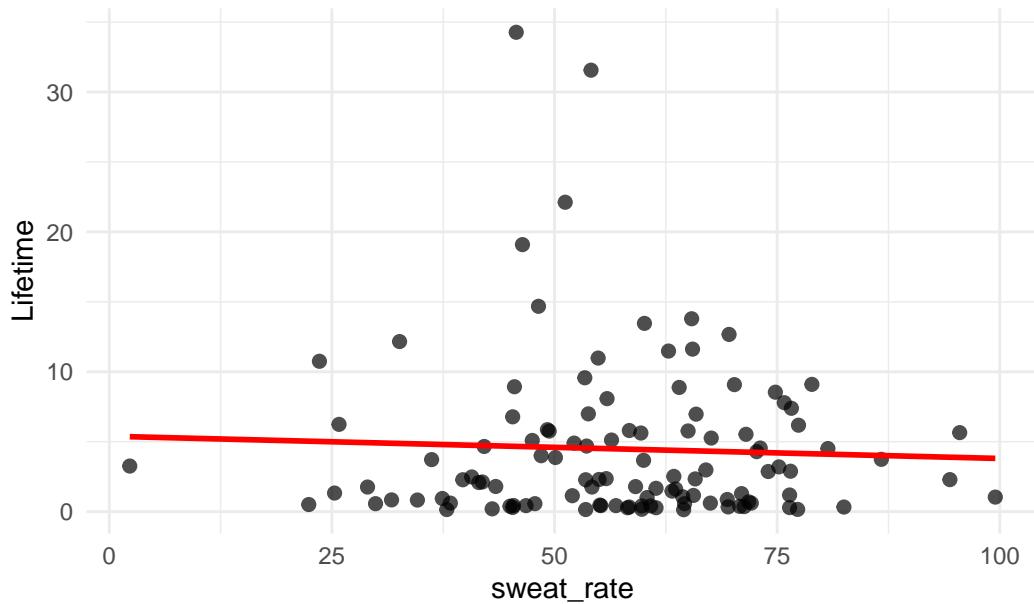
```
`geom_smooth()` using formula = 'y ~ x'
```

Lifetime vs activity\_level



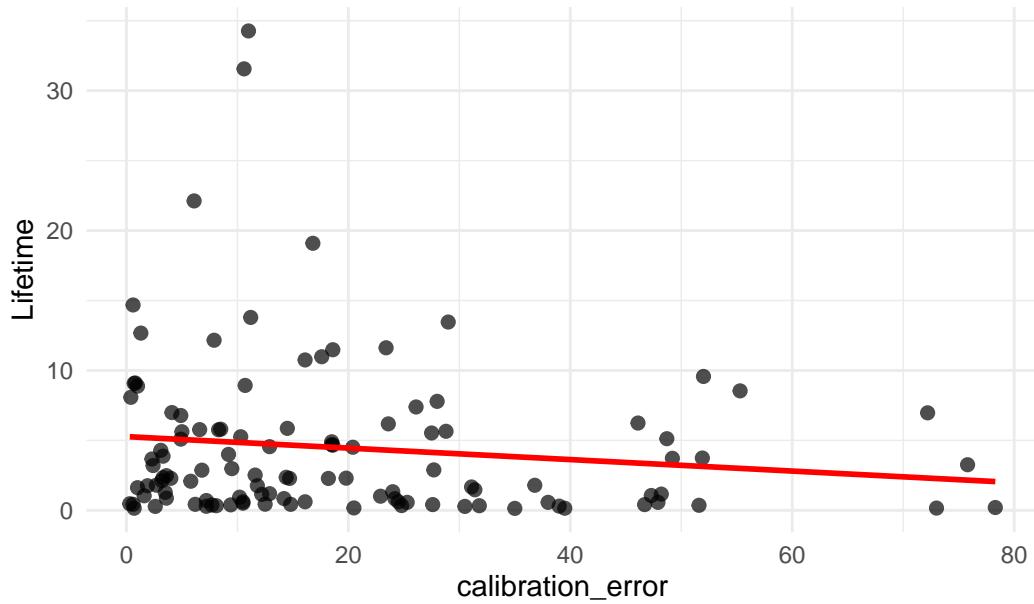
```
`geom_smooth()` using formula = 'y ~ x'
```

Lifetime vs sweat\_rate



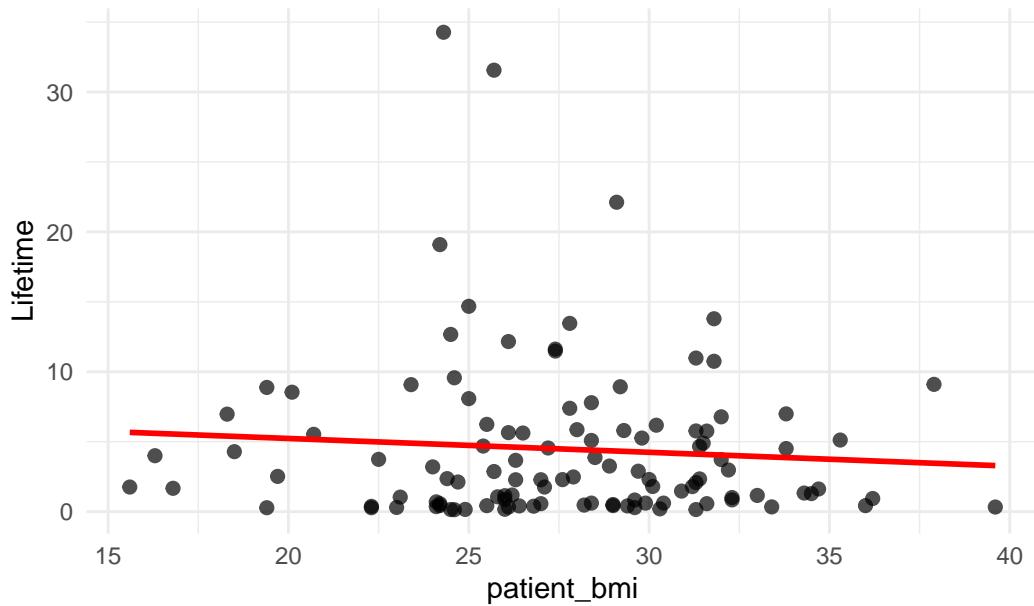
```
`geom_smooth()` using formula = 'y ~ x'
```

Lifetime vs calibration\_error



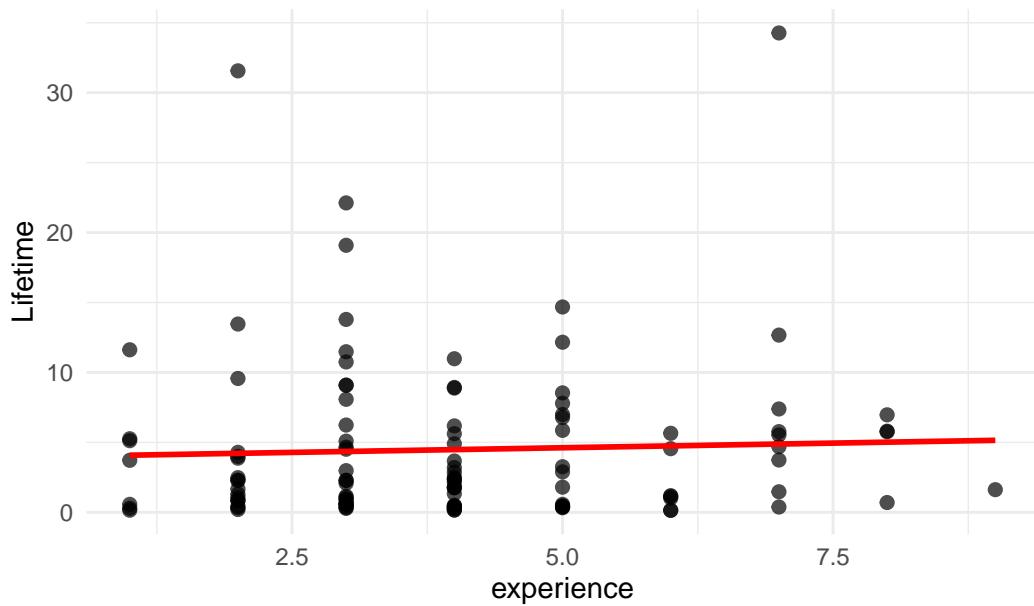
```
`geom_smooth()` using formula = 'y ~ x'
```

Lifetime vs patient\_bmi

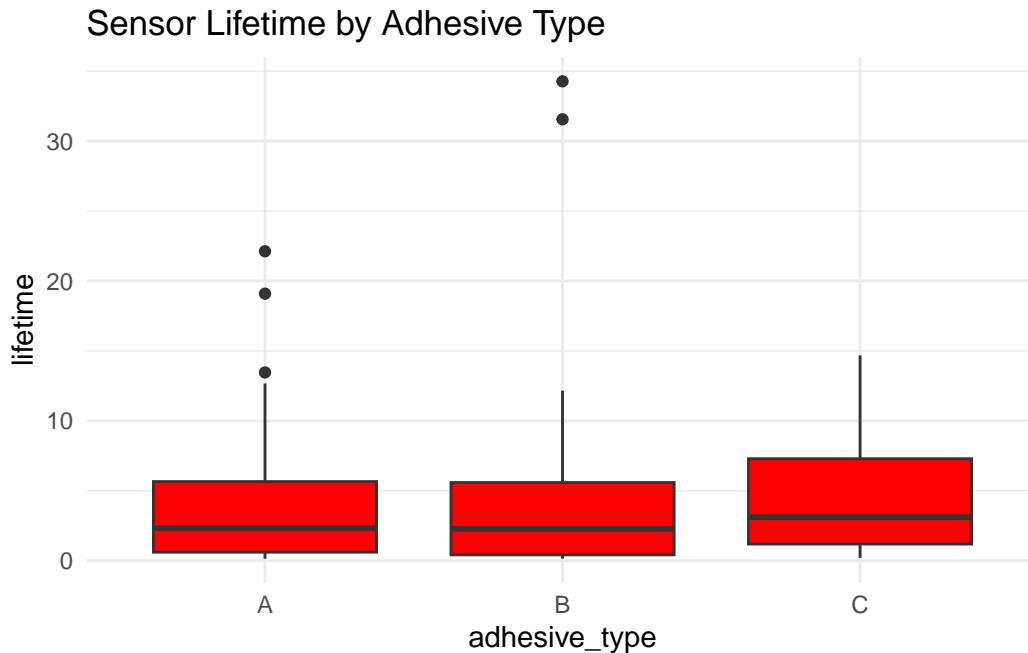


```
`geom_smooth()` using formula = 'y ~ x'
```

Lifetime vs experience



```
ggplot(df, aes(x = adhesive_type, y = lifetime)) +
  geom_boxplot(fill = "red") +
  theme_minimal() +
  labs(title = "Sensor Lifetime by Adhesive Type")
```



## subtask 2

a) , b), c)

```
df$adhesive_type <- as.factor(df$adhesive_type)

gamma_model_log <- glm(lifetime ~ skin_temp + humidity + activity_level +
                        sweat_rate + calibration_error + patient_bmi +
                        experience + adhesive_type,
                        data = df,
                        family = Gamma(link = "log"))

gamma_model_canon <- glm(lifetime ~ skin_temp + humidity + activity_level +
                           sweat_rate + calibration_error + patient_bmi +
                           experience + adhesive_type,
                           data = df,
                           family = Gamma(link = "inverse"))
```

```

inv_gauss_model_log <- glm(lifetime ~ skin_temp + humidity + activity_level +
                           sweat_rate + calibration_error + patient_bmi +
                           experience + adhesive_type,
                           data = df,
                           family = inverse.gaussian(link = "log"))

models <- list(
  Gamma_Log = gamma_model_log,
  Gamma_Canonical= gamma_model_canon,
  Inv_Gaussian_Log = inv_gauss_model_log
)

for (name in names(models)) {
  print(summary(models[[name]]))
}

```

Call:

```
glm(formula = lifetime ~ skin_temp + humidity + activity_level +
    sweat_rate + calibration_error + patient_bmi + experience +
    adhesive_type, family = Gamma(link = "log"), data = df)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-4.351848	0.915550	-4.753	6.52e-06 ***
skin_temp	0.063579	0.023355	2.722	0.00761 **
humidity	0.065677	0.002060	31.888	< 2e-16 ***
activity_level	-0.038755	0.013423	-2.887	0.00474 **
sweat_rate	0.001437	0.002366	0.607	0.54516
calibration_error	-0.015356	0.002122	-7.237	8.49e-11 ***
patient_bmi	-0.011409	0.008593	-1.328	0.18720
experience	0.012790	0.022671	0.564	0.57389
adhesive_typeB	0.106856	0.087403	1.223	0.22428
adhesive_typeC	0.226201	0.097173	2.328	0.02188 *
---				
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	'.'	'.'	'.'	'1'

(Dispersion parameter for Gamma family taken to be 0.1552009)

```

Null deviance: 172.431 on 112 degrees of freedom
Residual deviance: 16.835 on 103 degrees of freedom
AIC: 296.74

```

Number of Fisher Scoring iterations: 6

Call:

```
glm(formula = lifetime ~ skin_temp + humidity + activity_level +
    sweat_rate + calibration_error + patient_bmi + experience +
    adhesive_type, family = Gamma(link = "inverse"), data = df)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.7924456	0.3476194	5.156	1.22e-06 ***
skin_temp	-0.0112218	0.0075908	-1.478	0.1424
humidity	-0.0164098	0.0012003	-13.672	< 2e-16 ***
activity_level	0.0050336	0.0043813	1.149	0.2533
sweat_rate	-0.0005227	0.0008279	-0.631	0.5292
calibration_error	0.0018083	0.0007479	2.418	0.0174 *
patient_bmi	-0.0014776	0.0032355	-0.457	0.6489
experience	0.0058480	0.0037781	1.548	0.1247
adhesive_typeB	-0.0056280	0.0180707	-0.311	0.7561
adhesive_typeC	-0.0718624	0.0275000	-2.613	0.0103 *
---				

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.3877313)

Null deviance: 172.431 on 112 degrees of freedom

Residual deviance: 49.096 on 103 degrees of freedom

AIC: 423.02

Number of Fisher Scoring iterations: 5

Call:

```
glm(formula = lifetime ~ skin_temp + humidity + activity_level +
    sweat_rate + calibration_error + patient_bmi + experience +
    adhesive_type, family = inverse.gaussian(link = "log"), data = df)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-5.235503	1.049313	-4.989	2.47e-06 ***
skin_temp	0.071228	0.025664	2.775	0.00655 **
humidity	0.065180	0.002770	23.528	< 2e-16 ***
activity_level	-0.030130	0.011913	-2.529	0.01295 *

```

sweat_rate      0.002071   0.002452   0.844  0.40040
calibration_error -0.016903   0.001730  -9.771 2.39e-16 ***
patient_bmi      0.002108   0.009093   0.232  0.81715
experience        0.071114   0.027560   2.580  0.01128 *
adhesive_typeB    0.074050   0.081490   0.909  0.36563
adhesive_typeC    0.340794   0.115067   2.962  0.00380 **

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

(Dispersion parameter for inverse.gaussian family taken to be 0.138426)

```

Null deviance: 101.993  on 112  degrees of freedom
Residual deviance: 16.711  on 103  degrees of freedom
AIC: 376.3

```

Number of Fisher Scoring iterations: 13

```
aic_values <- sapply(models, AIC)
print(sort(aic_values))
```

Gamma_Log	Inv_Gaussian_Log	Gamma_Canonical
296.7400	376.2993	423.0153

```
resid_dev <- sapply(models, function(m) if("deviance" %in% names(m)) m$deviance else NA)
print(sort(resid_dev))
```

Inv_Gaussian_Log	Gamma_Log	Gamma_Canonical
16.71100	16.83515	49.09622

Data is strictly positive so potentially we should use Gamma or Inverse Gaussian. Tried fitting models with canonical links (had trouble with inverse so just fitted with log) and compared Best performing was LogGamma and that's what I'm going to use. Inverse gaussian doesn't seem to be necessary with its  $\mu^3$

d), e)

```
cat("\n==== Stepwise selection for Gamma_Log ====\n")
```

==== Stepwise selection for Gamma\_Log ===

```
best_model <- step(gamma_model_log)
```

Start: AIC=296.74  
lifetime ~ skin\_temp + humidity + activity\_level + sweat\_rate +  
calibration\_error + patient\_bmi + experience + adhesive\_type

	Df	Deviance	AIC
- sweat_rate	1	16.887	295.07
- experience	1	16.887	295.07
- patient_bmi	1	17.109	296.50
<none>		16.835	296.74
- adhesive_type	2	17.686	298.22
- skin_temp	1	17.842	301.22
- activity_level	1	18.077	302.74
- calibration_error	1	24.042	341.17
- humidity	1	154.826	1183.85

Step: AIC=295.09  
lifetime ~ skin\_temp + humidity + activity\_level + calibration\_error +  
patient\_bmi + experience + adhesive\_type

	Df	Deviance	AIC
- experience	1	16.948	293.49
- patient_bmi	1	17.190	295.07
<none>		16.887	295.09
- adhesive_type	2	17.738	296.62
- skin_temp	1	17.848	299.34
- activity_level	1	18.081	300.85
- calibration_error	1	24.092	339.88
- humidity	1	154.832	1188.87

Step: AIC=293.52  
lifetime ~ skin\_temp + humidity + activity\_level + calibration\_error +  
patient\_bmi + adhesive\_type

	Df	Deviance	AIC
<none>		16.948	293.52
- patient_bmi	1	17.338	294.06
- adhesive_type	2	17.763	294.84

```
- skin_temp           1  17.852 297.43
- activity_level     1  18.084 298.94
- calibration_error  1  24.092 338.23
- humidity            1 155.808 1199.43
```

```
summary(best_model)
```

Call:

```
glm(formula = lifetime ~ skin_temp + humidity + activity_level +
    calibration_error + patient_bmi + adhesive_type, family = Gamma(link = "log"),
    data = df)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-3.950384	0.811370	-4.869	3.98e-06 ***
skin_temp	0.056551	0.022074	2.562	0.01183 *
humidity	0.065824	0.002015	32.672	< 2e-16 ***
activity_level	-0.036130	0.012942	-2.792	0.00623 **
calibration_error	-0.015224	0.002093	-7.274	6.55e-11 ***
patient_bmi	-0.013182	0.008300	-1.588	0.11524
adhesive_typeB	0.111565	0.086478	1.290	0.19985
adhesive_typeC	0.218077	0.095406	2.286	0.02428 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.1529447)

```
Null deviance: 172.431 on 112 degrees of freedom
Residual deviance: 16.948 on 105 degrees of freedom
AIC: 293.52
```

Number of Fisher Scoring iterations: 6

```
cat("\n==== P value selection for Gamma_Log ===\n")
```

```
==== P value selection for Gamma_Log ===
```

```

alt_best_model <- glm(lifetime ~ skin_temp + humidity + activity_level +
                      calibration_error + adhesive_type,
                      data = df,
                      family = Gamma(link = "log"))
summary(alt_best_model)

```

Call:

```

glm(formula = lifetime ~ skin_temp + humidity + activity_level +
    calibration_error + adhesive_type, family = Gamma(link = "log"),
    data = df)

```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-4.508022	0.771177	-5.846	5.64e-08 ***
skin_temp	0.062116	0.022521	2.758	0.00685 **
humidity	0.065907	0.002064	31.930	< 2e-16 ***
activity_level	-0.037123	0.013239	-2.804	0.00600 **
calibration_error	-0.014644	0.002129	-6.880	4.33e-10 ***
adhesive_typeB	0.114224	0.088527	1.290	0.19976
adhesive_typeC	0.187642	0.097278	1.929	0.05642 .
---				

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.160626)

```

Null deviance: 172.431 on 112 degrees of freedom
Residual deviance: 17.338 on 106 degrees of freedom
AIC: 294.15

```

Number of Fisher Scoring iterations: 5

```

anova(best_model, alt_best_model)

```

Analysis of Deviance Table

```

Model 1: lifetime ~ skin_temp + humidity + activity_level + calibration_error +
          patient_bmi + adhesive_type
Model 2: lifetime ~ skin_temp + humidity + activity_level + calibration_error +
          adhesive_type
Resid. Df Resid. Dev Df Deviance

```

```

1      105    16.948
2      106  17.338 -1 -0.38946

```

```
final_model = alt_best_model
```

We tried stepwise model selection based on Akaike Information Criterion and selection based on p values (noted that we do not remove interaction terms as at least one of them has a very low p value) We end up with two models, fortunatly different in one predictor, we compare them using anova() command and choose the simpler alternative best model that doesnt include bmi predictor as it unnecessary

### subtask 3

Our final model is of Gamma kind with log link function ##### a)

```

final_model <- glm(lifetime ~ skin_temp + humidity + activity_level +
                     calibration_error + adhesive_type,
                     data = df,
                     family = Gamma(link = "log"))
summary_table <- summary(final_model)$coefficients
round(summary_table, 4)

```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-4.5080	0.7712	-5.8456	0.0000
skin_temp	0.0621	0.0225	2.7581	0.0068
humidity	0.0659	0.0021	31.9304	0.0000
activity_level	-0.0371	0.0132	-2.8041	0.0060
calibration_error	-0.0146	0.0021	-6.8801	0.0000
adhesive_typeB	0.1142	0.0885	1.2903	0.1998
adhesive_typeC	0.1876	0.0973	1.9289	0.0564

b)

```

mean_skin_temp <- mean(df$skin_temp)
mean_humidity <- mean(df$humidity)
mean_cal_error <- mean(df$calibration_error)

# adhesive A most prelevant so chose it
# BMI is not present in my final model

```

```
eta <- coef(final_model)["(Intercept)"] +
  coef(final_model)[["skin_temp"]]*mean_skin_temp +
  coef(final_model)[["humidity"]]*mean_humidity +
  coef(final_model)[["activity_level"]]*4 +
  coef(final_model)[["calibration_error"]]*mean_cal_error

fitted_value <- exp(eta)
fitted_value
```

(Intercept)  
2.111193

c)

YEAH MAKE PLOTS

d)

```
deviance(final_model)
```

[1] 17.33769

```
AIC(final_model)
```

[1] 294.1473

e)

```
summary(final_model)$dispersion
```

[1] 0.160626

this indicates underdispersion, could follow potentially with quasiGamma to fix it but under-dispersion seems fine (not as bad as overdispersion, the fit is good)

f)

DO THAT ANALYSIS THO

## Task 2

```
load("Data_Assignment2_Ex2_E2025.rdata")
df_x <- x
df_y <- Y
str(df_x)
```

```
num [1:100] 8.3 40.9 51.5 39.7 22.3 29.2 58.4 49.1 92.3 28 ...
```

```
str(df_y)
```

```
num [1:100] -119.4 -18.1 -3.5 -17.5 -69.5 ...
```