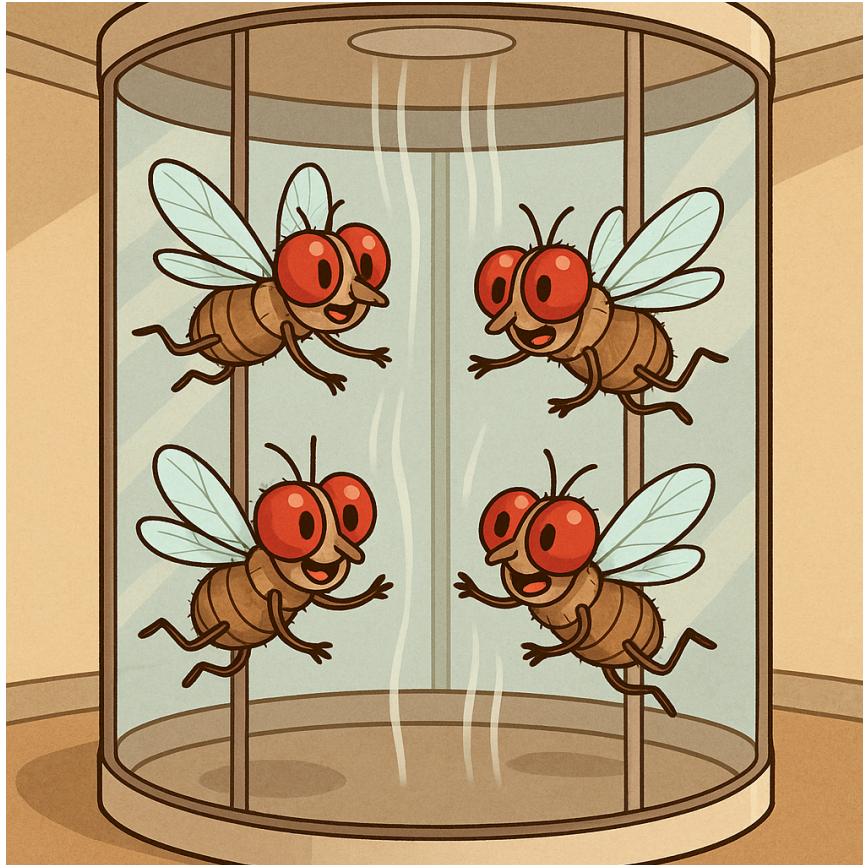


# Statistical Analysis of Flight Assay Data

Jun Ishigohoka



## Preparation

If you don't have the packages, install them.

```
1 packages <- c("ggplot2", "car")
2 for (pkg in packages) {
3   if (!requireNamespace(pkg, quietly = TRUE)) {
4     install.packages(pkg)
5   }
6 }
```

Load libraries.

```
1 library(ggplot2)
2 library(car) # Anova() and leveneTest()
```

Read data in two data frames. For this protocol, test data will be used.

```
1 d_1 <- read.csv("data/test_data/R4_female.csv")
2 d_2 <- read.csv("data/test_data/R4_male.csv")
3
4 head(d_1)
```

```
##   X.1   Area      X      Y
## 1   1  5.456 407.768  9.192
## 2   2  6.666 343.943 10.074
## 3   3 11.087 306.180 11.690
## 4   4  4.932 376.921 16.866
## 5   5  4.569 148.680 25.018
## 6   6  5.389 330.855 26.006
```

Add a column for the factor you are comparing.

```
1 d_1$sex <- "female"
2 d_2$sex <- "male"
```

Concatenate the two data frames. Then set it as a factor.

```
1 d <- rbind(d_1, d_2)
2 d$sex <- factor(d$sex, levels = c("female", "male"))
```

## Visualisation

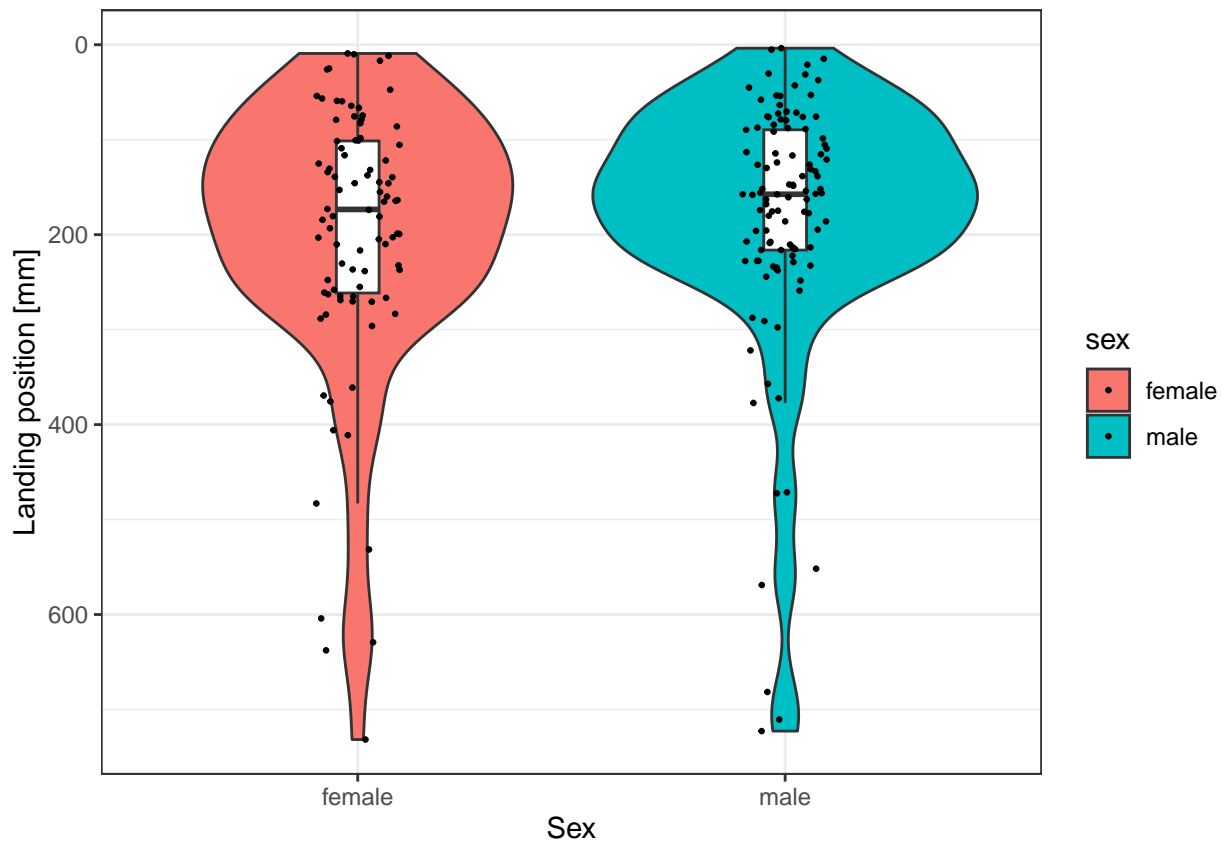
Let's visualise the distribution of landing height. **Once you have visualised the distributions, try to describe them.**

If you have ggplot2...

```

1 ggplot(data = d, # data to be plotted
2       mapping = aes(x=sex, y=Y, fill = sex)) + # columns in the data frame to be used
3       geom_violin() + # violin plot
4       geom_boxplot( # box plot
5         width = 0.1, # width of the box plot
6         fill = "white", # colour inside the box plot
7         outliers = F # whether to have points for outliers.
8                       # FALSE because of geom_fitter() below
9       ) +
10      geom_jitter( # each data point with "jitter" along x axis
11        size = 0.5, # size of points
12        width = 0.1 # the amount of jitter
13      ) +
14      scale_y_reverse( # to make the plot same direction as the tapes
15        limits = c(NA, 0) # so that the lowest Y is 0
16      ) +
17      labs( # Add labels
18        x = "Sex",
19        y = "Landing position [mm]") +
20      theme_bw()

```

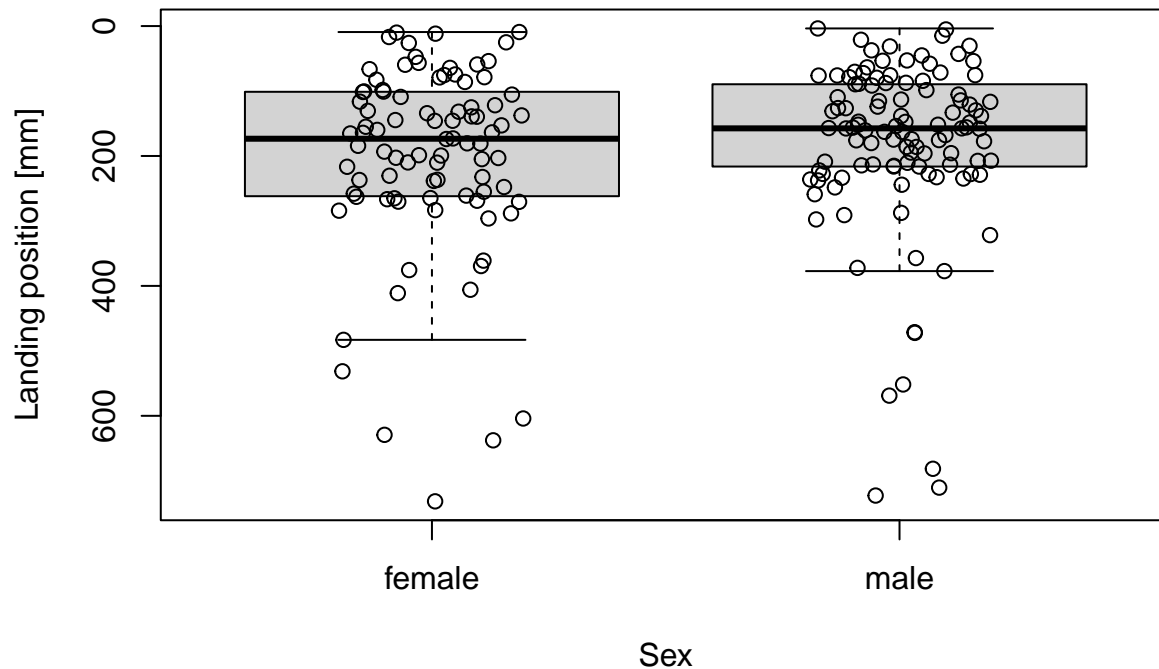


If you prefer the base plot or do not have ggplot2 installed.

```

1 boxplot(d$Y ~ as.factor(d$sex),
2         pch = NA,
3         xlab = "Sex",
4         ylab = "Landing position [mm]",
5         ylim = rev(range(d$Y))
6     )
7 points(jitter(as.numeric(as.factor(d$sex))), d$Y)

```



## Statistical analysis (hypothesis testing)

### Does your data follow normality?

Many statistical tests assume normality. Let's run Shapiro-Wilk test for each group to test normality.

```

1 shapiro.test(subset(d, sex == "female")$Y)

```

```

##
##  Shapiro-Wilk normality test
##
## data:  subset(d, sex == "female")$Y
## W = 0.87191, p-value = 3.497e-07

```

```

1 shapiro.test(subset(d, sex == "male")$Y)

```

```

##
##  Shapiro-Wilk normality test

```

```
##
## data:  subset(d, sex == "male")$Y
## W = 0.80816, p-value = 1.326e-10
```

Does the result fit to your description of the distribution?

If they are not normally distributed, think why. Do you expect normally distributed data from this experiment? (Hints: Poisson process; waiting time)

## Does your data have equal variance?

Many parametric tests assume equal variance between groups. Brown-Forsythe test for homogeneity. If you have `car` installed

```
1 leveneTest(Y ~ sex, d, center = median)

## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value Pr(>F)
## group    1  0.8082 0.3698
##           195
```

If you do not have `car` installed, compute deviance from median for each sex manually, and run ANOVA.

```
1 d$z <- NA
2 for (sex in levels(d$sex)) {
3   idx <- which(d$sex == sex)
4   d[idx,]$z <- abs(d[idx, ]$Y - median(d[idx, ]$Y))
5 }
6
7 summary(aov(d$z ~ d$sex))

##           Df Sum Sq Mean Sq F value Pr(>F)
## d$sex      1    9114    9114   0.808  0.37
## Residuals 195 2199050   11277
```

## Is mean different between groups?

If the data are normally distributed and have equal variance, Student t-test

```
1 t.test(d$Y ~ d$sex,
2       var.equal = T,
3       paired = F
4       )
```

```
##
## Two Sample t-test
##
## data:  d$Y by d$sex
```

```
## t = 0.92975, df = 195, p-value = 0.3537
## alternative hypothesis: true difference in means between group female and group male
## 95 percent confidence interval:
## -20.94187  58.29702
## sample estimates:
## mean in group female    mean in group male
##           200.7005           182.0229
```

If data are normal but not with equal variance, Welch's t-test

```
1 t.test(d$Y ~ d$sex,
2       var.equal = F,
3       paired = F
4       )
```

```
##
##  Welch Two Sample t-test
##
## data:  d$Y by d$sex
## t = 0.92407, df = 181.57, p-value = 0.3567
## alternative hypothesis: true difference in means between group female and group male
## 95 percent confidence interval:
## -21.20353  58.55868
## sample estimates:
## mean in group female    mean in group male
##           200.7005           182.0229
```

If data are not normally distributed, you can run non-parametric tests or permutation test.

### Option 1: Non-parametric test

Let's run Mann-Whitney U test.

```
1 wilcox.test(d$Y ~ d$sex,
2           paired = F
3           )
```

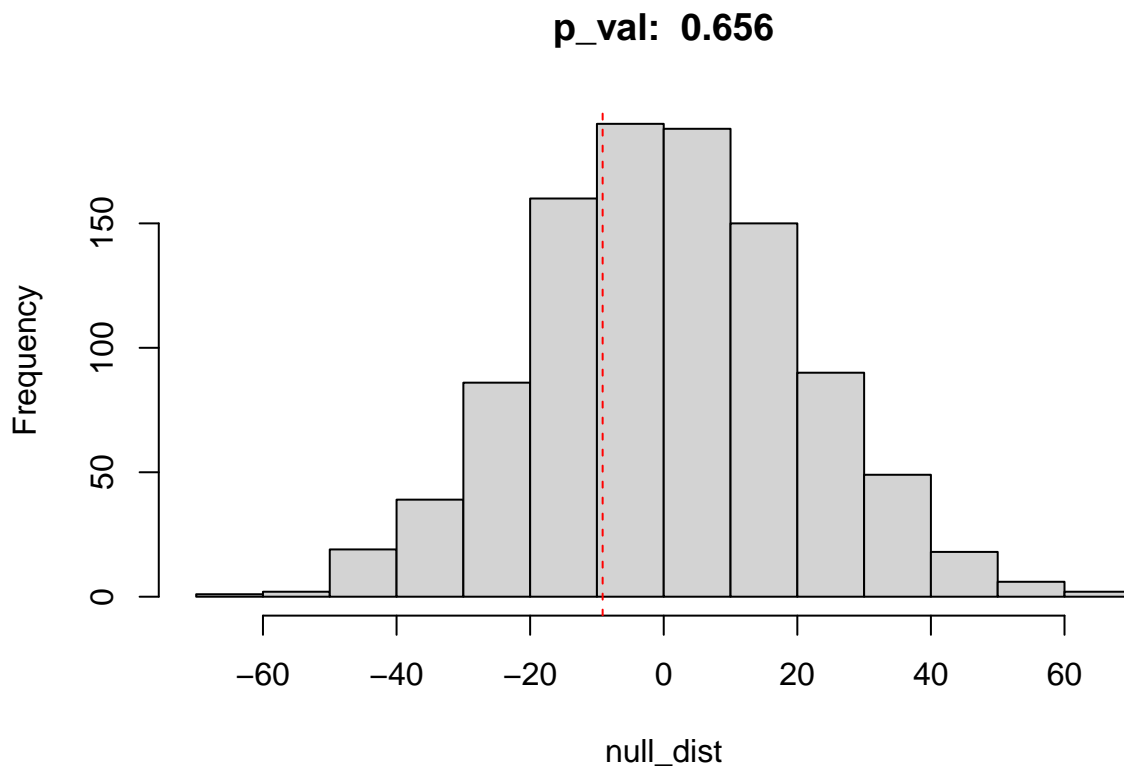
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d$Y by d$sex
## W = 5247, p-value = 0.2575
## alternative hypothesis: true location shift is not equal to 0
```

### Option 2: Permutation test

Let's write a function to run a permutation test. Briefly, we shuffle the label, compute difference between groups. We repeat this process many times to get a null distribution. We compute p-value based on the position of the observed difference between groups in the null

distribution.

```
1  # Null distribution of mean(Y | female) - mean(Y | male)
2  null_dist <- sapply(1:1000, # repeat 1000 times
3    function(x){
4      d_tmp <- d
5      d_tmp$sex <- sample(d$sex) # shuffle label
6      return(mean(d_tmp[d_tmp$sex == "female", "Y"])
7        - mean(d_tmp[d_tmp$sex == "male", "Y"]))
8    }
9  )
10
11
12  # Observed value of mean(female) - mean(male)
13  obs <- mean(d[d$sex == "female", "Y"] - d[d$sex == "male", "Y"])
14
15  # p value as the rank of observation in the null distribution
16  p_val <- sum(abs(null_dist) > abs(obs)) / length(null_dist)
17
18  hist(null_dist,
19    main = paste("p_val: ", p_val))
20  abline(v = obs, col = "red", lty = 2)
```



## Optional: Statistical modeling

Here, we conduct statistical modeling of landing distance using a generalised linear model (GLM) approach with the exponential distribution, considering landing events as a Poisson process with a constant “landing rate”. Because the waiting time of a Poisson process is an exponentially distributed random variable, we can use a GLM with exponential distribution. We compare the result with a linear model (LM), which assumes a normal distribution as the underlying distribution.

### GLM

Fit the data to an exponential GLM.

```
1 glm_1 <- glm(Y ~ sex,  
2             data = d,  
3             family = Gamma(link = "log") # This is how to specify exponential  
4             )
```

According to the fitted GLM, do the two groups have different landing rate?

If you have car installed

```
1 Anova(glm_1)  
  
## Analysis of Deviance Table (Type II tests)  
##  
## Response: Y  
##      LR Chisq Df Pr(>Chisq)  
## sex  0.85803  1    0.3543  
  
1 summary(glm_1, dispersion = 1) # exponential is a Gamma with dispersion = 1  
  
##  
## Call:  
## glm(formula = Y ~ sex, family = Gamma(link = "log"), data = d)  
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  5.30181    0.10660  49.735  <2e-16 ***  
## sexmale     -0.09768    0.14331  -0.682    0.495  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for Gamma family taken to be 1)  
##  
##      Null deviance: 111.53  on 196  degrees of freedom  
## Residual deviance: 111.07  on 195  degrees of freedom  
## AIC: 2427.1
```



```
##
## Number of Fisher Scoring iterations: 5
```

If you do not have car installed

```
1 drop1(glm_1, test = "Chi")
```

```
## Single term deletions
##
## Model:
## Y ~ sex
##           Df Deviance      AIC scaled dev. Pr(>Chi)
## <none>           111.07 2427.1
## sex          1   111.53 2425.9      0.85803   0.3543
```

## LM

Fit the data to a linear model.

```
1 lm_1 <- lm(Y ~ sex,
2           data = d)
```

If you do not have car installed

```
1 drop1(lm_1, test = "Chi")
```

```
## Single term deletions
##
## Model:
## Y ~ sex
##           Df Sum of Sq      RSS      AIC Pr(>Chi)
## <none>           3831693 1949.5
## sex          1    16986 3848679 1948.4   0.3506
```

```
1 summary(lm_1)
```

```
##
## Call:
## lm(formula = Y ~ sex, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -191.51  -94.30  -25.02   45.70  540.72
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    200.70      14.94   13.43  <2e-16 ***
## sexmale       -18.68      20.09   -0.93   0.354
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.2 on 195 degrees of freedom
## Multiple R-squared:  0.004413,    Adjusted R-squared:  -0.0006922
## F-statistic: 0.8644 on 1 and 195 DF,  p-value: 0.3537
```

## GLM vs LM

Compare the diagnostic plots<sup>1</sup> between `glm_1` and `lm_1`. \*\*Is using LM for landing distance data worse than exponential GLM?"

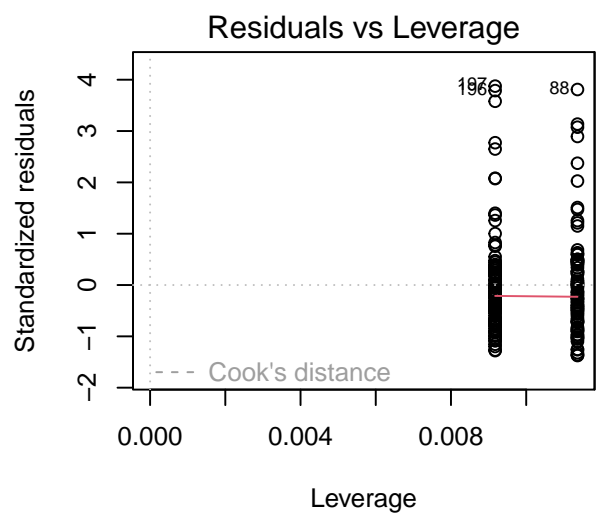
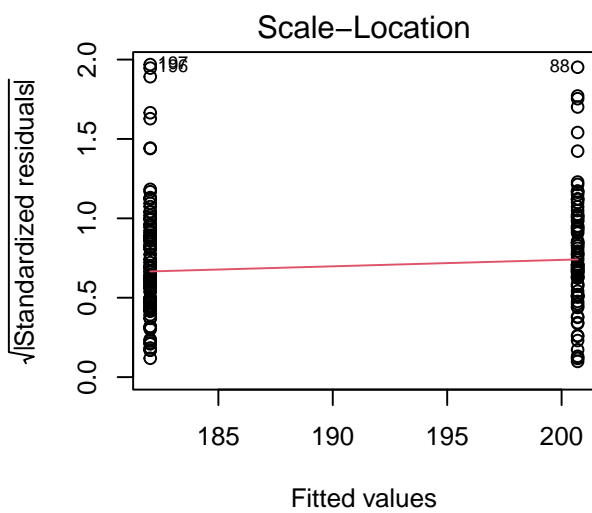
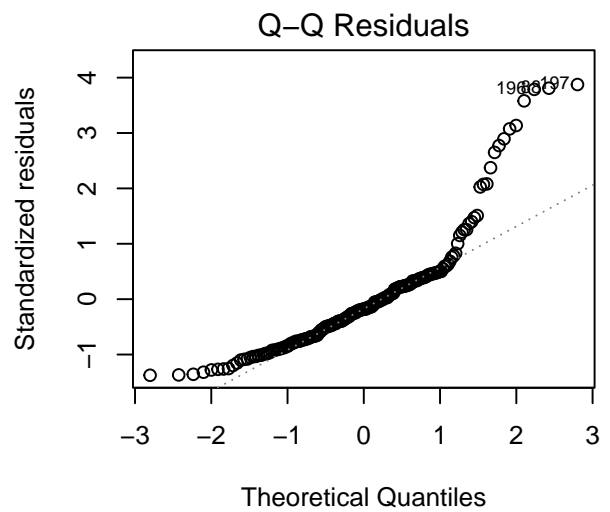
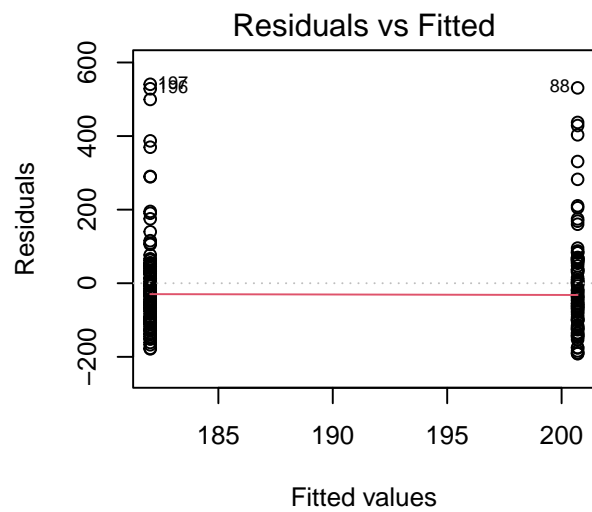
- Is the linearity assumption met?
- Is the data homoscedastic?
- Are residuals distributed normally?
- Are there outliers?

```
1 par(mfrow = c(2,2),
2     oma = c(0, 0, 2, 0)
3     )
4 plot(lm_1)
```

---

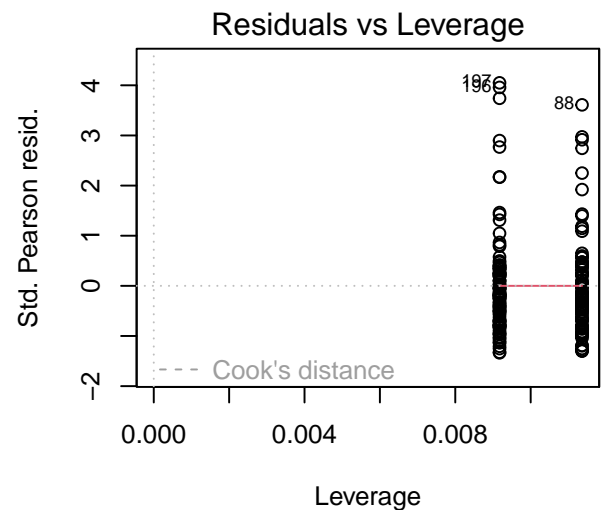
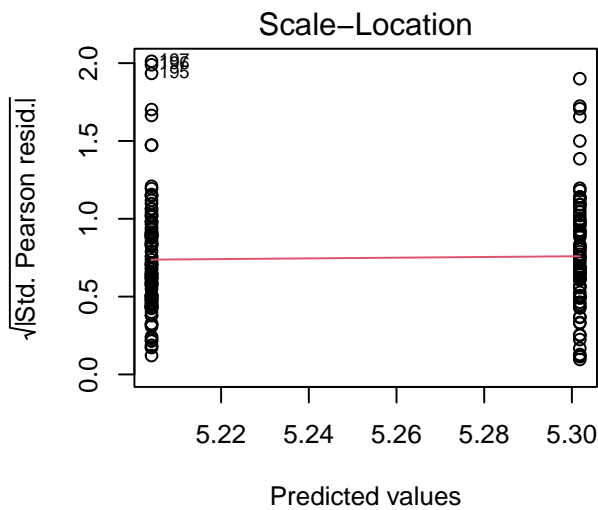
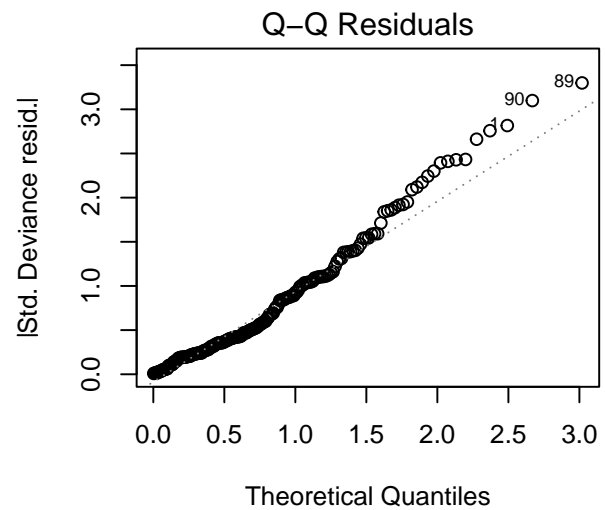
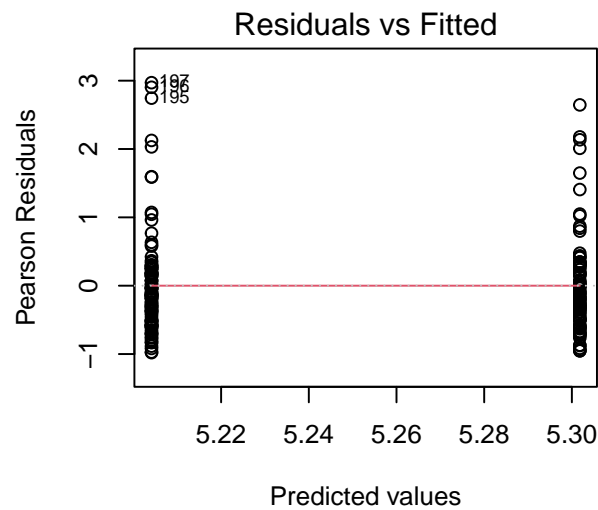
<sup>1</sup>What do the diagnostic plots mean? <https://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/>

lm(Y ~ sex)



```
1 par(mfrow = c(2,2),
2     oma = c(0, 0, 2, 0)
3     )
4 plot(glm_1)
```

glm(Y ~ sex)



## What's next?

In real life, we have multiple biological and technical replicates. **How do we account for such random effects?**

```
1 sessionInfo()

## R version 4.3.3 (2024-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 24.04.2 LTS
##
## Matrix products: default
```

```

## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.12.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0
##
## locale:
## [1] LC_CTYPE=en_AU.UTF-8      LC_NUMERIC=C              LC_TIME=de_DE.UTF-8
## [4] LC_COLLATE=en_AU.UTF-8    LC_MONETARY=de_DE.UTF-8   LC_MESSAGES=en_AU.UTF-8
## [7] LC_PAPER=de_DE.UTF-8      LC_NAME=C                 LC_ADDRESS=C
## [10] LC_TELEPHONE=C            LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Berlin
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] car_3.1-3      carData_3.0-5 ggplot2_3.5.1 hvar1_1.0.4
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.6      jsonlite_1.8.9     dplyr_1.1.4        compiler_4.3.3     tinytex_
## [6] tidyselect_1.2.1  Rcpp_1.0.13-1      jquerylib_0.1.4     revealjs_0.10.0    scales_1
## [11] png_0.1-8         yaml_2.3.10        fastmap_1.2.0      R6_2.5.1           labeling_
## [16] generics_0.1.3    Formula_1.2-5      knitr_1.49         tibble_3.2.1       munsell_
## [21] bslib_0.8.0       pillar_1.9.0       rlang_1.1.4        utf8_1.2.4         cachem_1
## [26] xfun_0.49         sass_0.4.9         cli_3.6.3          withr_3.0.2        magrittr
## [31] digest_0.6.37     grid_4.3.3         rstudioapi_0.17.1  lifecycle_1.0.4    vctrs_0.
## [36] evaluate_1.0.1    glue_1.8.0         farver_2.1.2       abind_1.4-8        fansi_1.
## [41] colorspace_2.1-1  rmarkdown_2.29     tools_4.3.3        pkgconfig_2.0.3    htmltool

```