Introduction to Applied Statistics for STEM Researchers

Frank Hause

Research Training Group 2467: „Intrinsically Disordered Proteins – Molecular Principles, Cellular Functions, and Diseases“

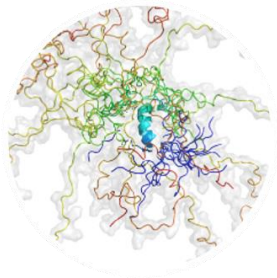
Winter Semester 2025/26

MARTIN-LUTHER-UNIVERSITÄT
HALLE-WITTENBERG

RTG 2467

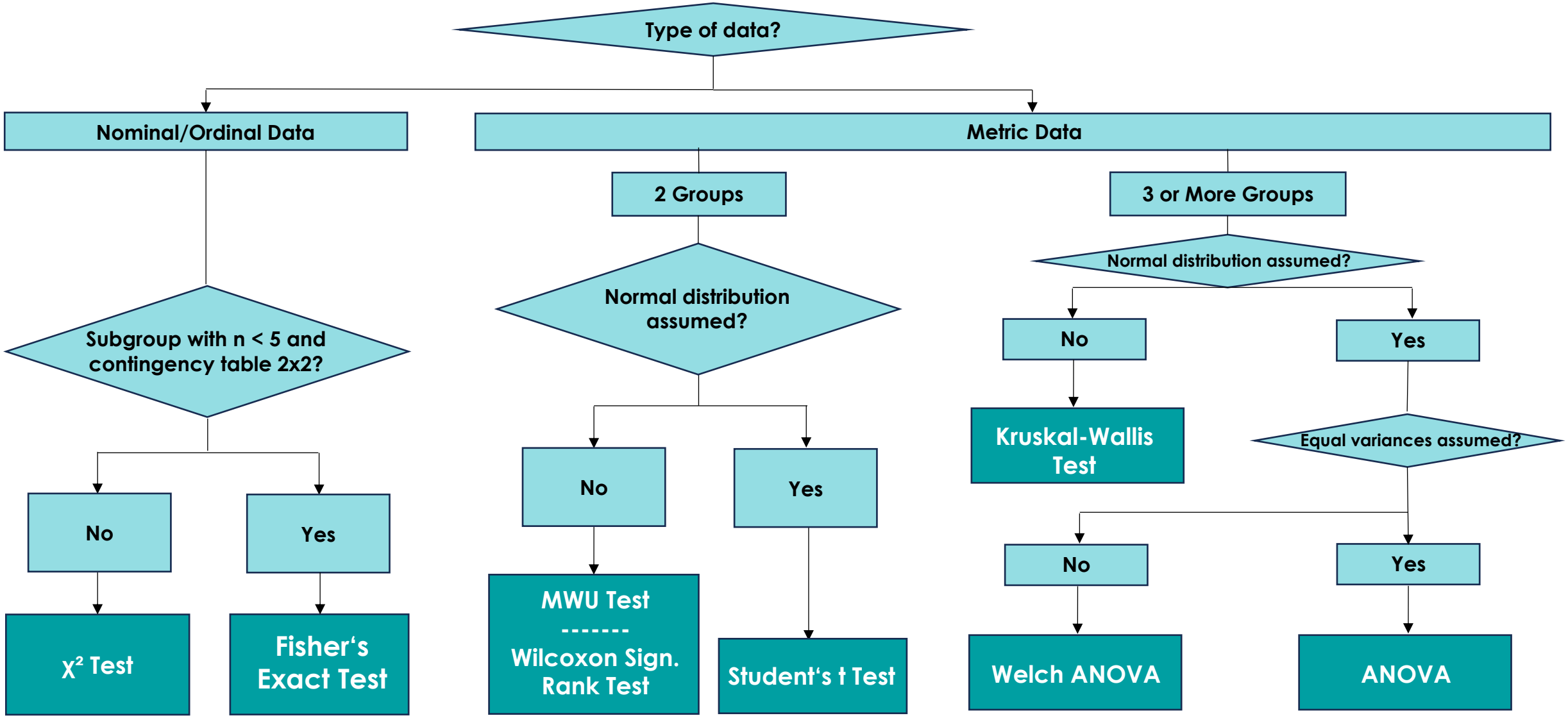
Intrinsically Disordered Proteins – Molecular Principles, Cellular Functions, and Diseases

Speaker of the RTG: Prof. Dr. Andrea Sinz
Scientific Coordinator: Dr. Oleksandr Sorokin



Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Recap

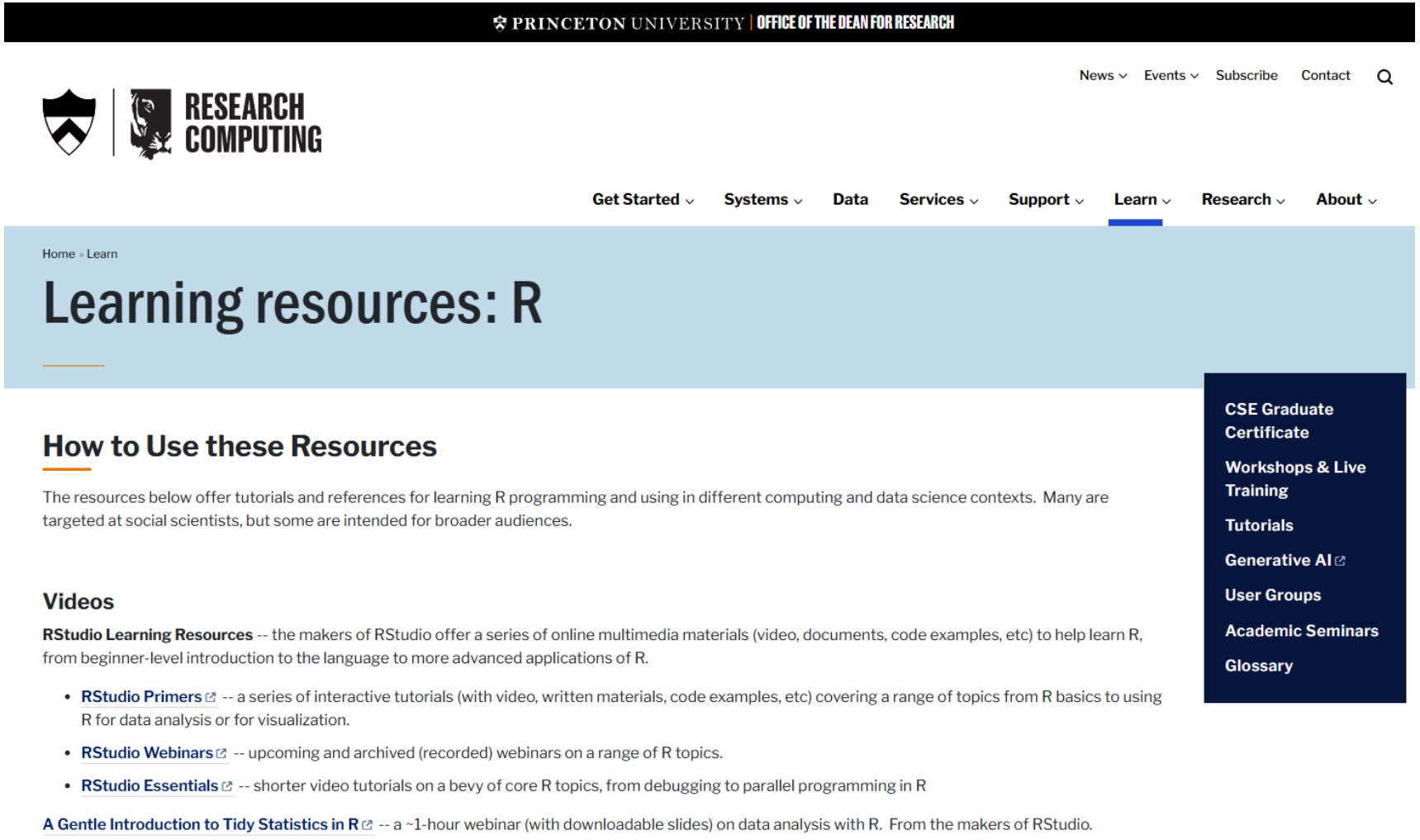

Type of data?
Nominal/Ordinal Data
Metric Data
Subgroup with n < 5 and contingency table 2x2?
No
Yes
X² Test
Fisher's Exact Test
2 Groups
Normal distribution assumed?
No
Yes
MWU Test

Wilcoxon Sign. Rank Test
Student's t Test
3 or More Groups
Normal distribution assumed?
No
Yes
Kruskal-Wallis Test
Equal variances assumed?
No
Yes
Welch ANOVA
ANOVA

Learning R

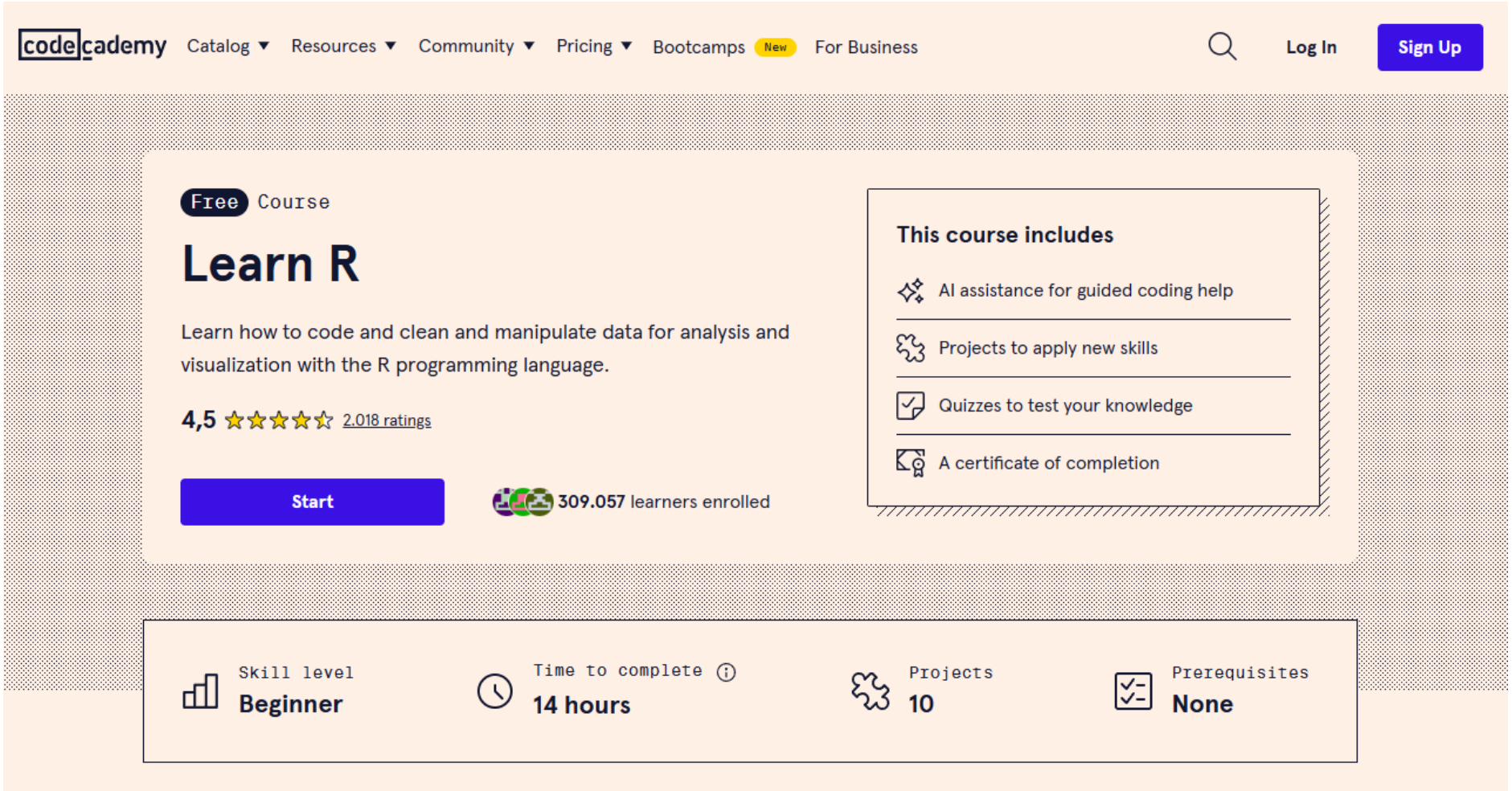
https://statistik.wiwi.uni-halle.de/lehre/203701_3339568

Learning R



https://researchcomputing.princeton.edu/education/external-online-resources/R

Learning R



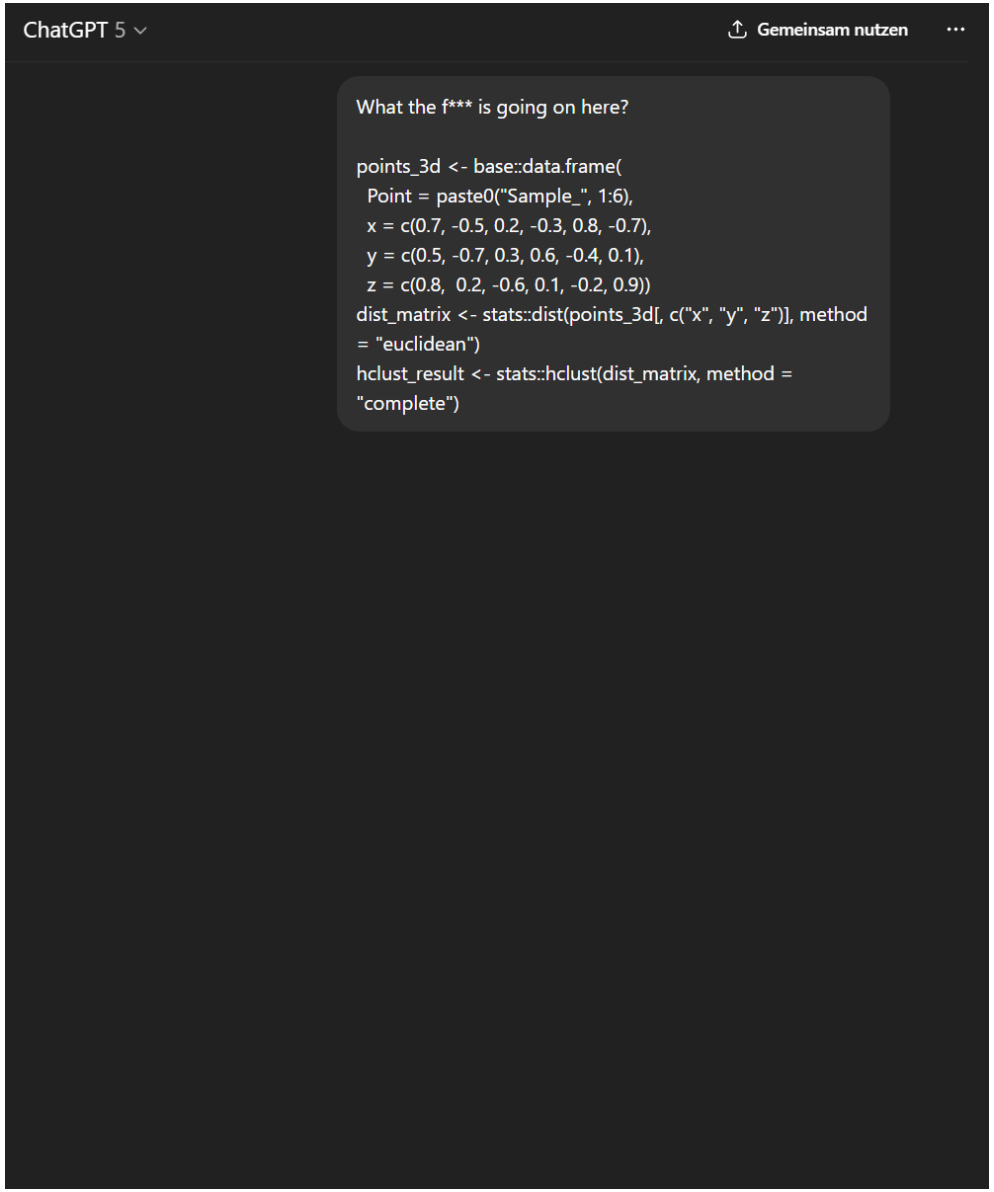
https://www.codecademy.com/learn/learn-r

Learning R

https://www.youtube.com/@RProgramming101

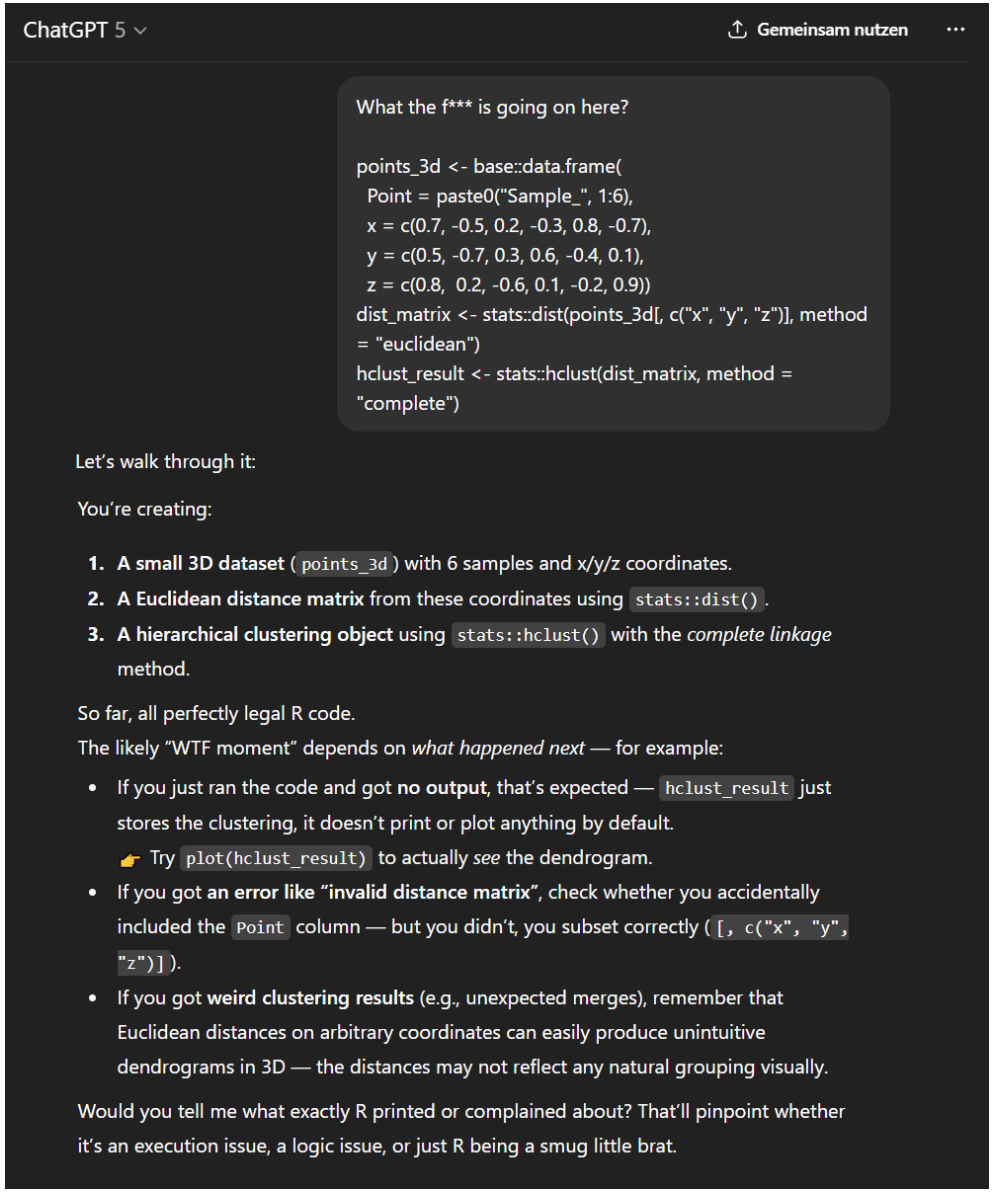
Learning R

https://chatgpt.com

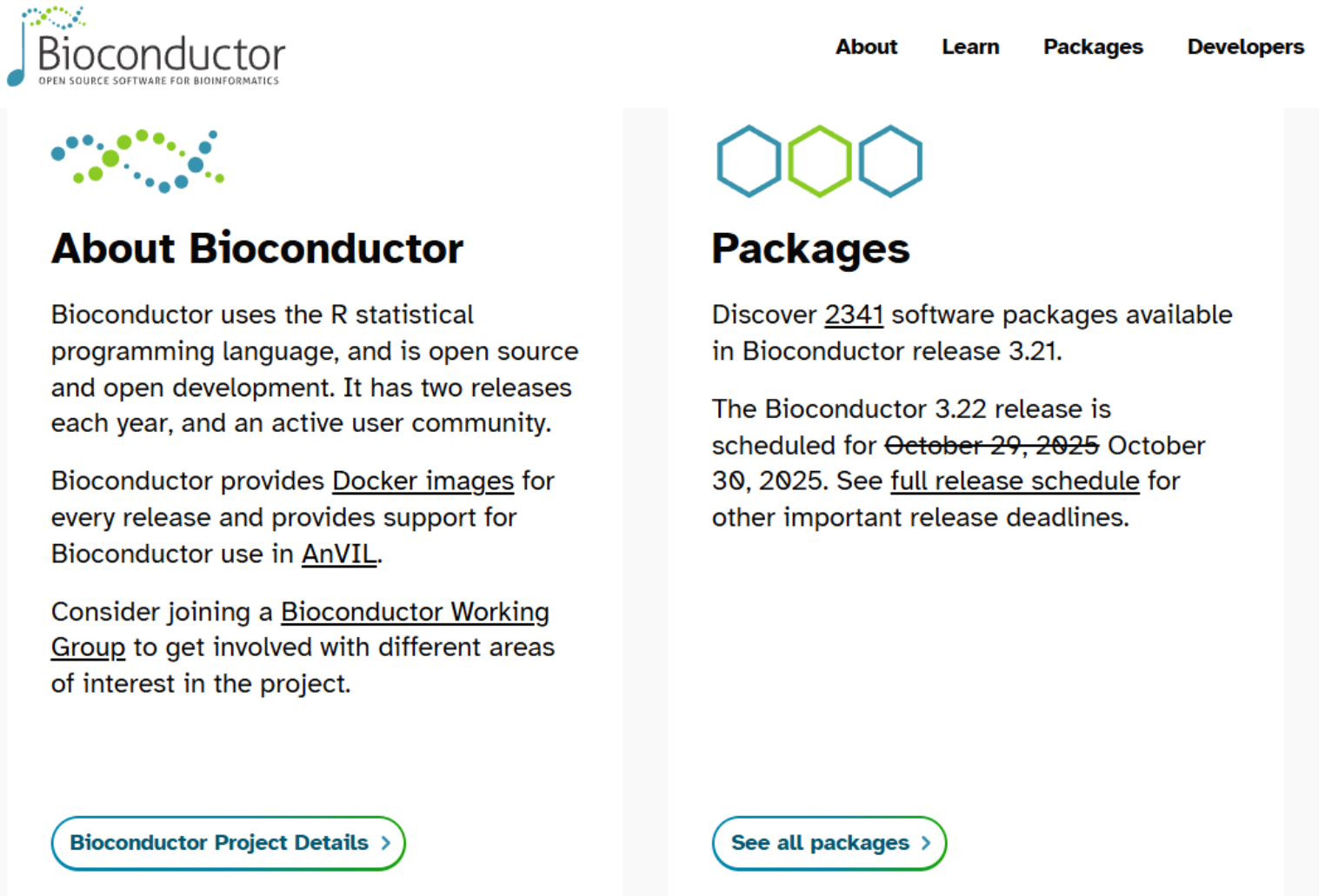


Learning R

https://chatgpt.com



Learning R



https://www.bioconductor.org

Course Outline

Part I: Comparing Entities

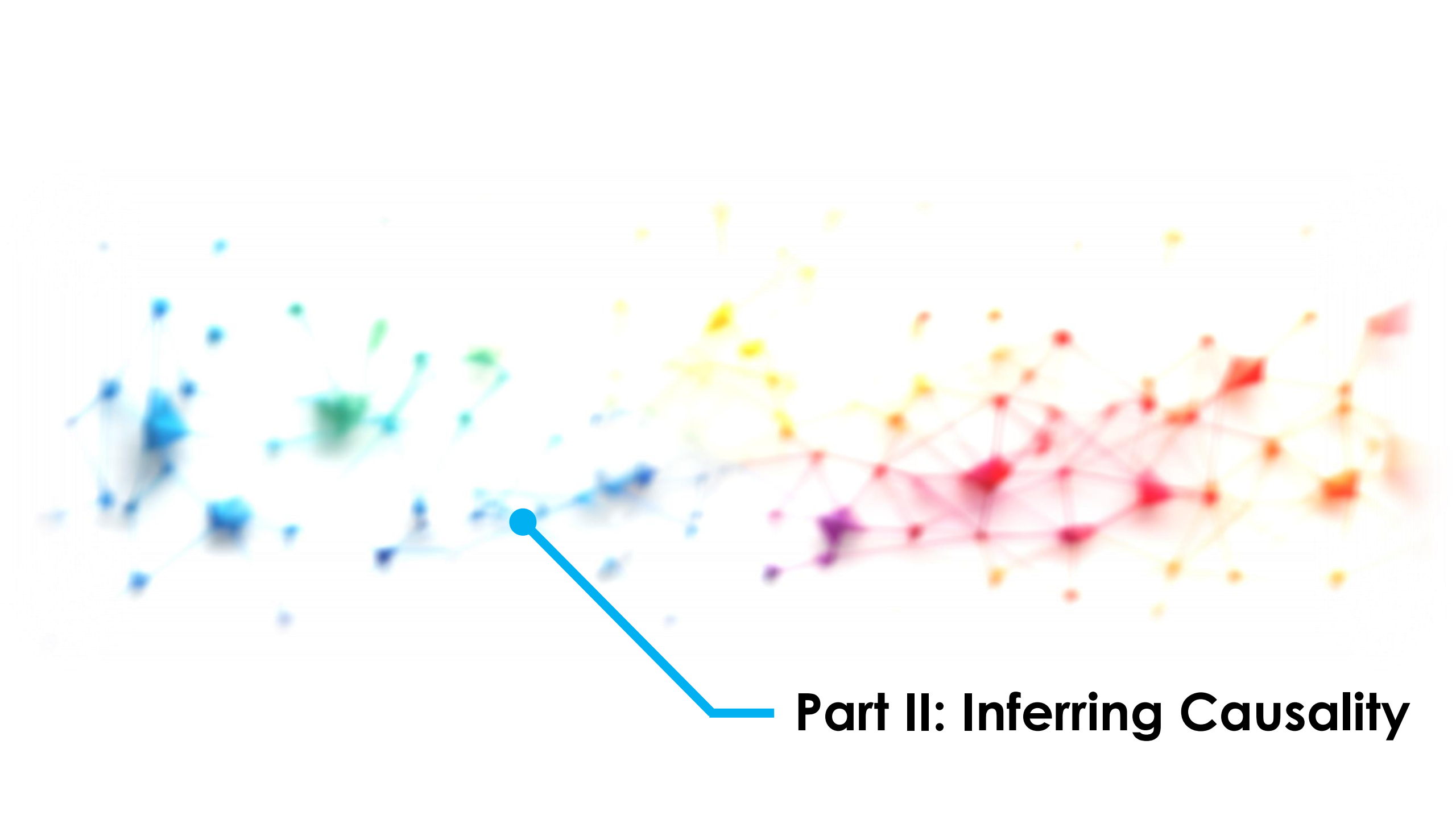
Part II: Inferring Causality

Part III: Coping with Complexity

Scripts & Exercises:

frank.hause@medizin.uni-halle.de

@fhause.bsky.social



Part II: Inferring Causality

Part II: Outline

- **Introduction to Causation and Correlation**
- **Correlations**
  - Spearman
  - Pearson
  - Partial
- **Regression Analysis**
  - Linear Regression
  - Logistic Regression
- **Survival Analysis**
  - Kaplan-Meier Curves
  - Log Rank-Test
  - Cox Proportional Hazard Analysis

Causation and Correlation

Causation

Correlation

Causation and Correlation

People on the Beach

Causation

Causation

Ice Cream Consumption

Correlation

Incidence of Sunburn

Causation and Correlation

Causation		Correlation
Mandatory	**Directionality**	Optional
Diachronous	**Temporal Sequence**	Synchronous
No influence by third variables	**Spurious Relationship**	Confounders always possible
Likely impossible	**Experimental Validation**	Sometimes easy
Strong and definite	**Strength of Relation**	Varying and volatile

Difficulties in Determining Causality

- Confounders cannot be excluded completely (number often reduced)
- In molecular science temporal sequence is often obscure
- Experimental conditions may have unknown effects on the system
- In empirical science, cause and effect do not exist isolated, but are embedded in a causal chain

→Statistics as a tool estimate and **approximate causal certainty** and to cope with and **manage uncertainty**

→Statistics are **not able to prove causality**, but can be used to infer it

Correlations

- Spearman Correlation

- Pearson Correlation

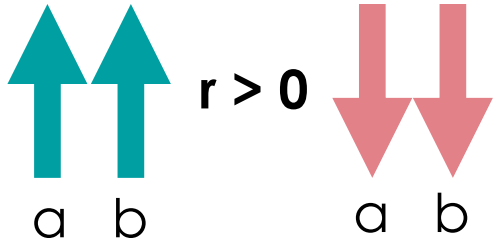
- Partial Correlation

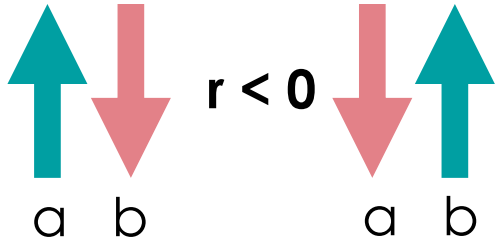
Correlations

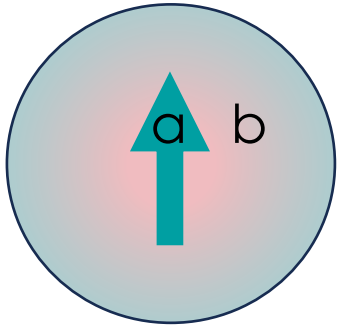
- describe the non-directed, independent relationship between **two (at least ordinal) variables**

- Correlation coefficient (r) **[-1, +1]** reflects the **direction of relation**

- Significance level (p) reflects the **probability** that the **direction of relation occurs by chance**







r ~ 0, for n >> 0

Spearman Correlation

- **non-parametric:** measures ranked correlation

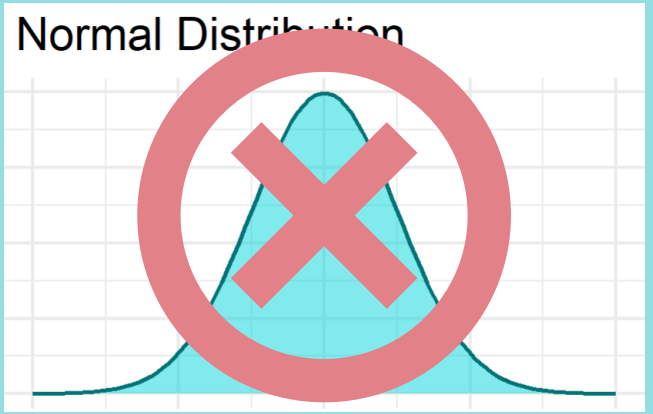
Examples:

- Association Between Disease Severity (7-point scale) and Biomarker Levels
- Correlation Between Habitat Quality (ordinal data) and Species Richness
- Correlation Between Drug Dosage (Low, Medium, High) and Response Rate (10-point scale)

```r
# Example data
your_group1 <- c(1, 2, 3, 4, 5)
your_group2 <- c(2, 4, 6, 8, 10)

# Calculate Spearman correlation and p-value
cor.test(your_group1, your_group2, method = "spearman")
```

```python
import numpy as np
from scipy.stats import spearmanr

# Example data
your_group1 = np.array([1, 2, 3, 4, 5])
your_group2 = np.array([2, 4, 6, 8, 10])

# Calculate Spearman correlation and p-value
correlation, p_value = spearmanr(your_group1, your_group2)
```

Requirements:



Pearson Correlation

- **parametric:** measures linear correlation

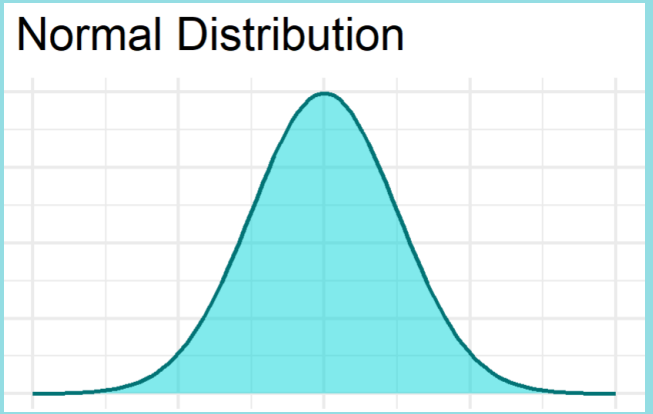
Examples:

- Relationship between expression levels of a gene and a phenotypic trait, such as height or weight
- Relationship between blood pressure and cholesterol levels
- Association between enzyme concentration and reaction rate

```r
# Example data
your_group1 <- c(1, 2, 3, 4, 5)
your_group2 <- c(2, 4, 6, 8, 10)

# Calculate Pearson correlation and p-value
cor.test(your_group1, your_group2, method = "pearson")
```

```python
import numpy as np
from scipy.stats import pearsonr

# Example data
your_group1 = np.array([1, 2, 3, 4, 5])
your_group2 = np.array([2, 4, 6, 8, 10])

# Calculate Pearson correlation and p-value
correlation, p_value = pearsonr(your_group1, your_group2)
```

Requirements:



Partial Correlation

- **potentially parametric:** inherits requirements from parent method (Spearman or Pearson)
- Measures the correlation between two continuous variables while eliminating the direct influence of a control variable

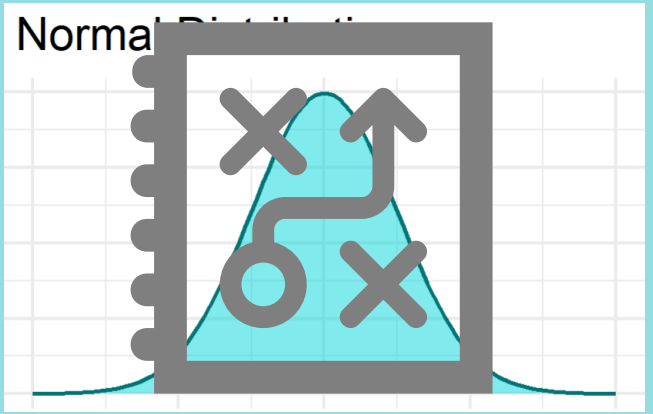
Examples:

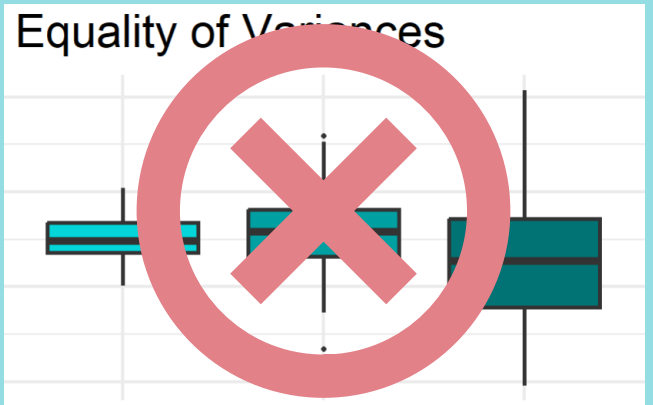
- Effect of Carbohydrate Uptake on Blood Glucose Levels Controlling for Exercise
- Association Between Brain Activity and Cognitive Function Controlling for Age

```r
# Example data
your_group1 <- c(1, 2, 3, 4, 5)
your_group2 <- c(2, 4, 6, 8, 10)
your_ctrl.var <- c(3, 6, 9, 12, 15)  # Additional variable
                                     # for partial correlation

# Calculate partial correlation and p-value
cor.test(your_group1, your_group2, method = "pearson",
         partial = your_ctrl.var)
```

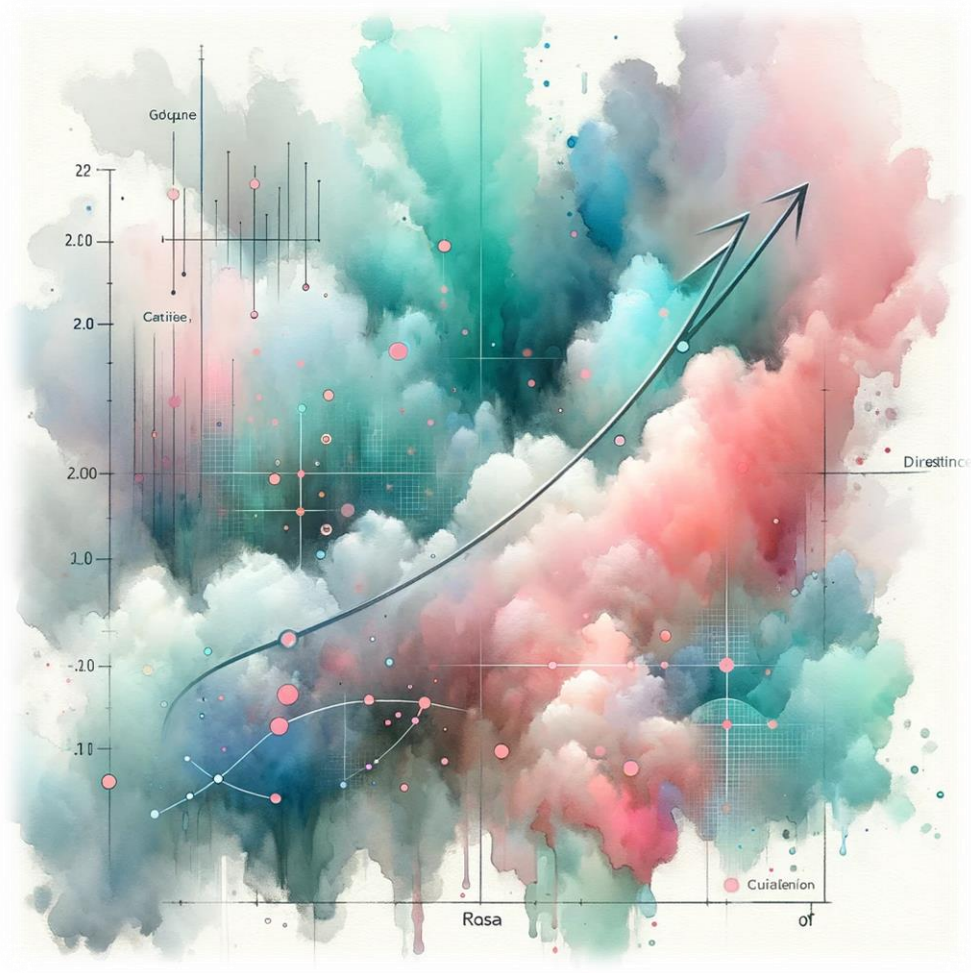
```python
import pandas as pd
import pingouin as pg

# Example data
data = pd.DataFrame({'your_group1': [1, 2, 3, 4, 5],
                     'your_group2': [2, 4, 6, 8, 10],
                     'your_ctrlvar': [3, 6, 9, 12, 15]})  # Additional
                                                          # variable for
                                                          # partial correlation

# Calculate partial correlation and p-value
partial_result = pg.partial_corr(data=data, x='your_group1',
                                 y='your_group2', covar='your_ctrlvar')
partial_correlation = partial_result['r']
p_value = partial_result['p-val']
```

Requirements:





Regression Models



- Linear Regression

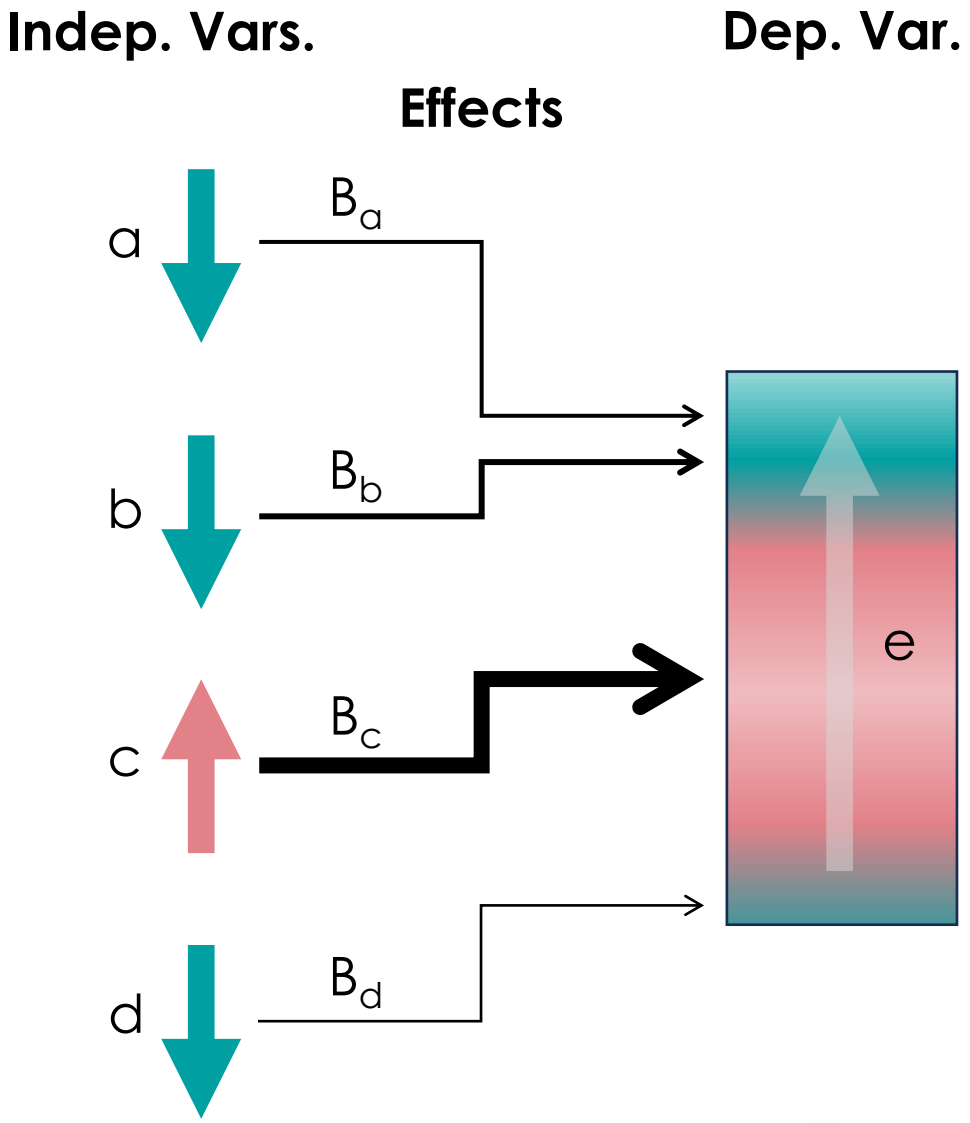
- Logistic Regression

Regression Models

- Describe the directed relationship between **presumably linked variables**

- Causation is statistically modeled by declaring **one variable dependent** and **at least one other variable independent** (predictors)

- **Type of directed relation** or **kind of influence** (B) the independent variables exert on the dependent determines the **nature of the regression model** (linear or logistic)



Linear Regression Models

Figures to Consider:

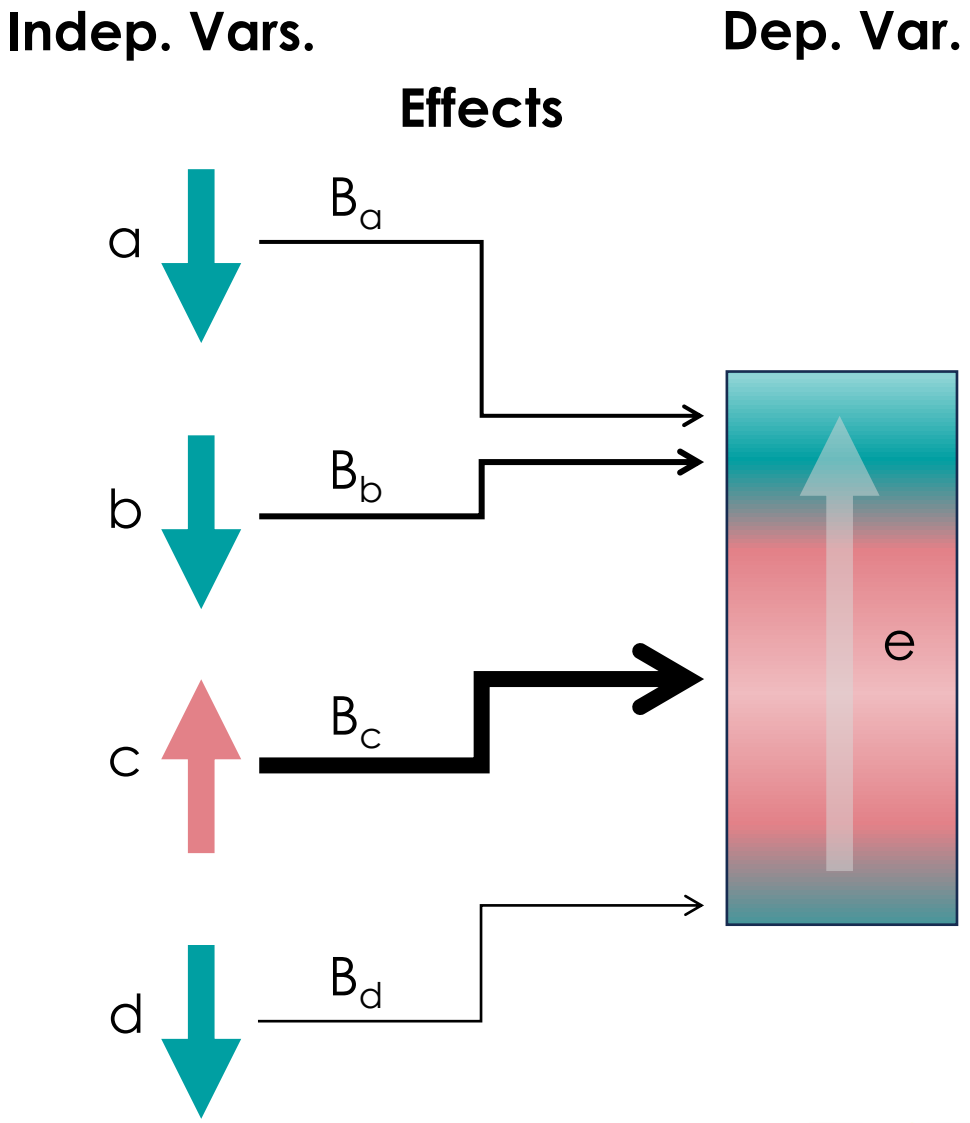
Model p value (p_{regr}): Does the model describe a relation that is different from a random effects-model?

Model Constant (C): Describes the error that is inherent to the model

Explanatory Power/Model Quality (R^2): Share of variance of the dependent variable that is explained by the predictors

Regression Coefficient (B) per indep. variable: Measure for strength of influence: If the predictor will increase by 1, this will cause the dependent variable to change by B

p value ($p_{a..n}$) per indep. variable: Does the predictor exert an effect which is distinguishable from a random effect?



Linear Regression Models

Figures to Consider:

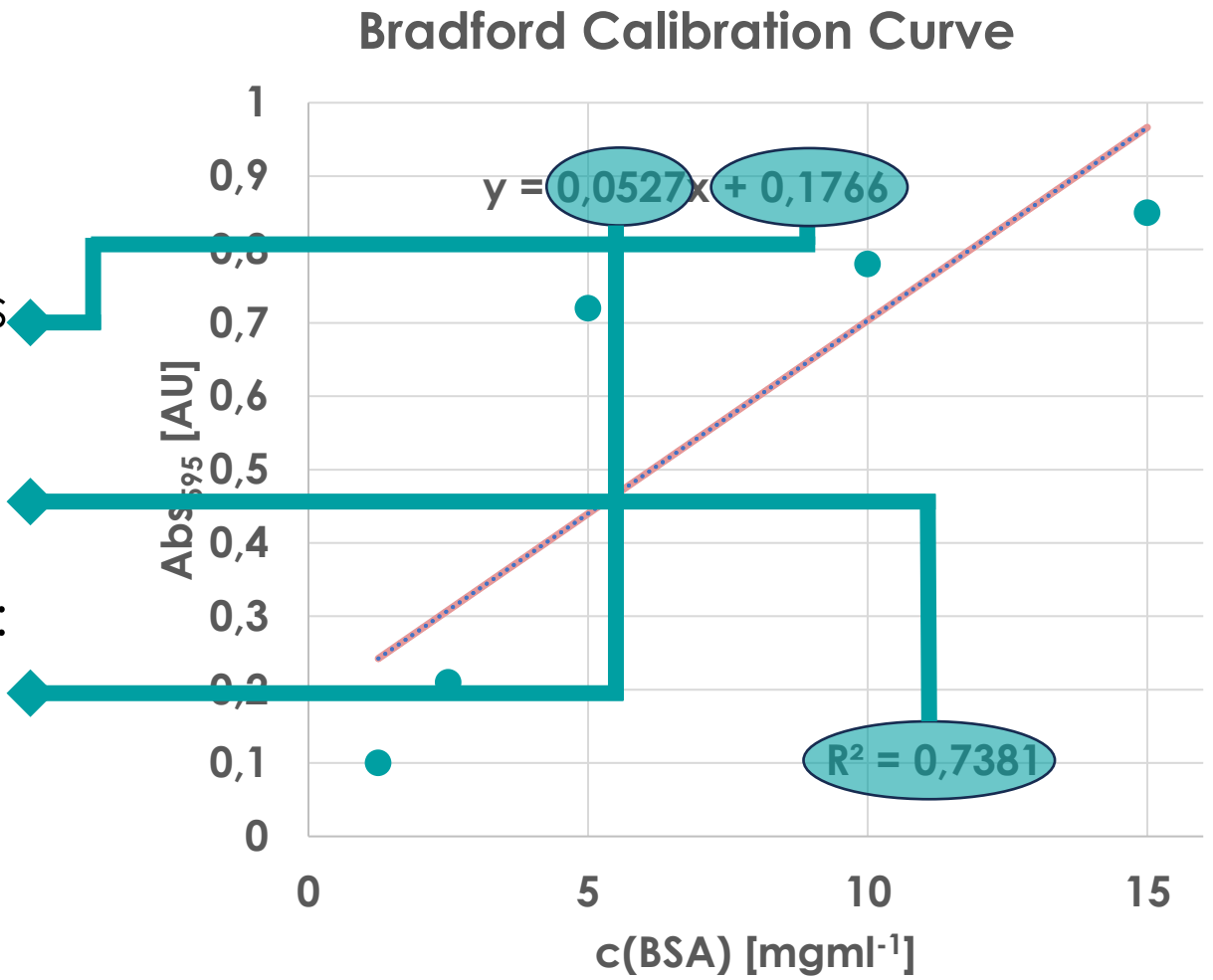
Model p value (p_{regr}): Does the model describe a relation that is different from a random effects-model?

Model Constant (C): Describes the error that is inherent to the model

Explanatory Power/Model Quality (R^2): Share of variance of the dependent variable that is explained by the predictors

Regression Coefficient (B) per indep. variable: Measure for strength of influence: If the predictor will increase by 1, this will cause the dependent variable to change by B

p value ($p_{a..n}$) per indep. variable: Does the predictor exert an effect which is distinguishable from a random effect?



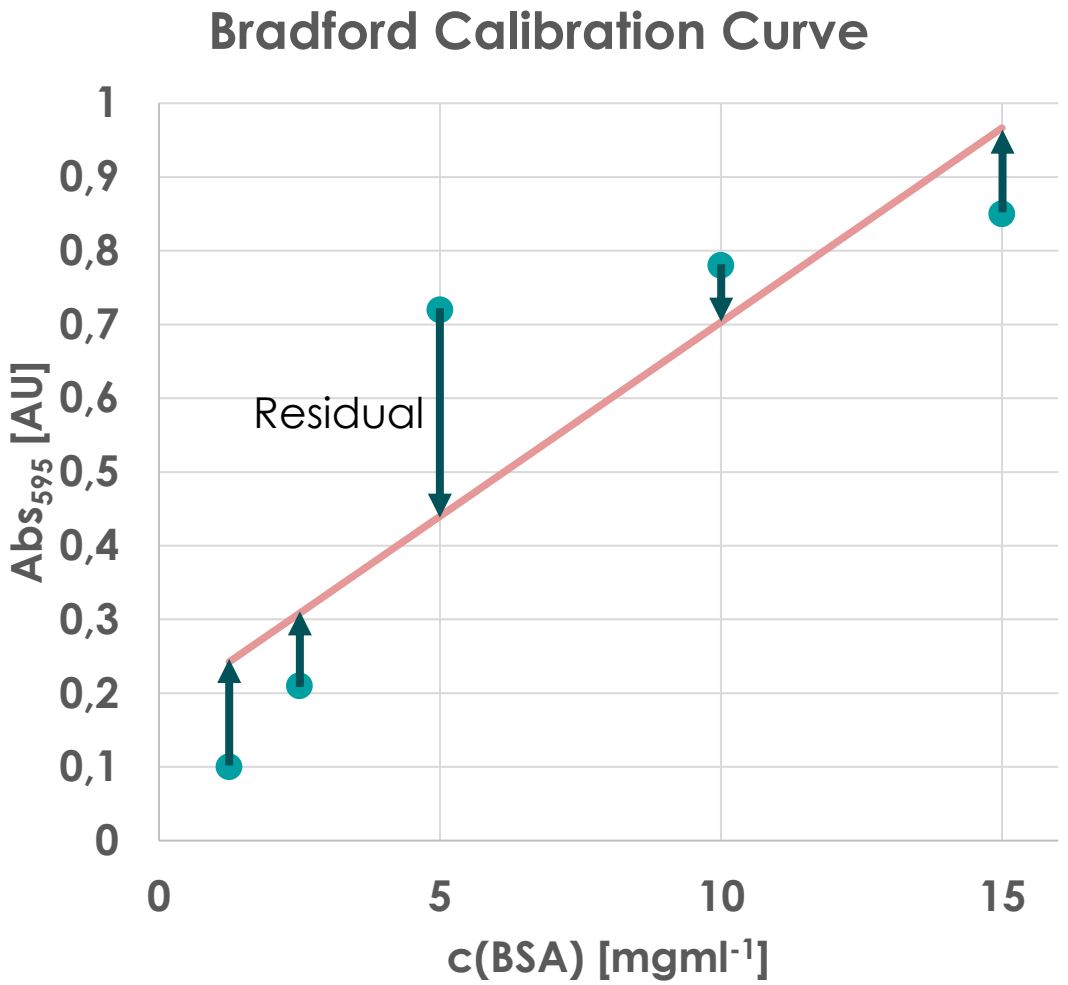
Linear Regression Models

Prerequisites for Linear Regression:

- **Effect** of predictors on the dependent variable **must be linear**

- **Residuals** (absolute deviation between every empirical measurement and the regression line of the model):
  - Must be normally distributed (Kolmogorov-Smirnov/Shapiro-Wilk Test)
  - Must be independent (no auto-correlation) (Durbin-Watson Statistic)

  - No Outliers
  - Equality of Variances (multiple linear regression only)
  - Multi-Collinearity (multiple linear regression only)



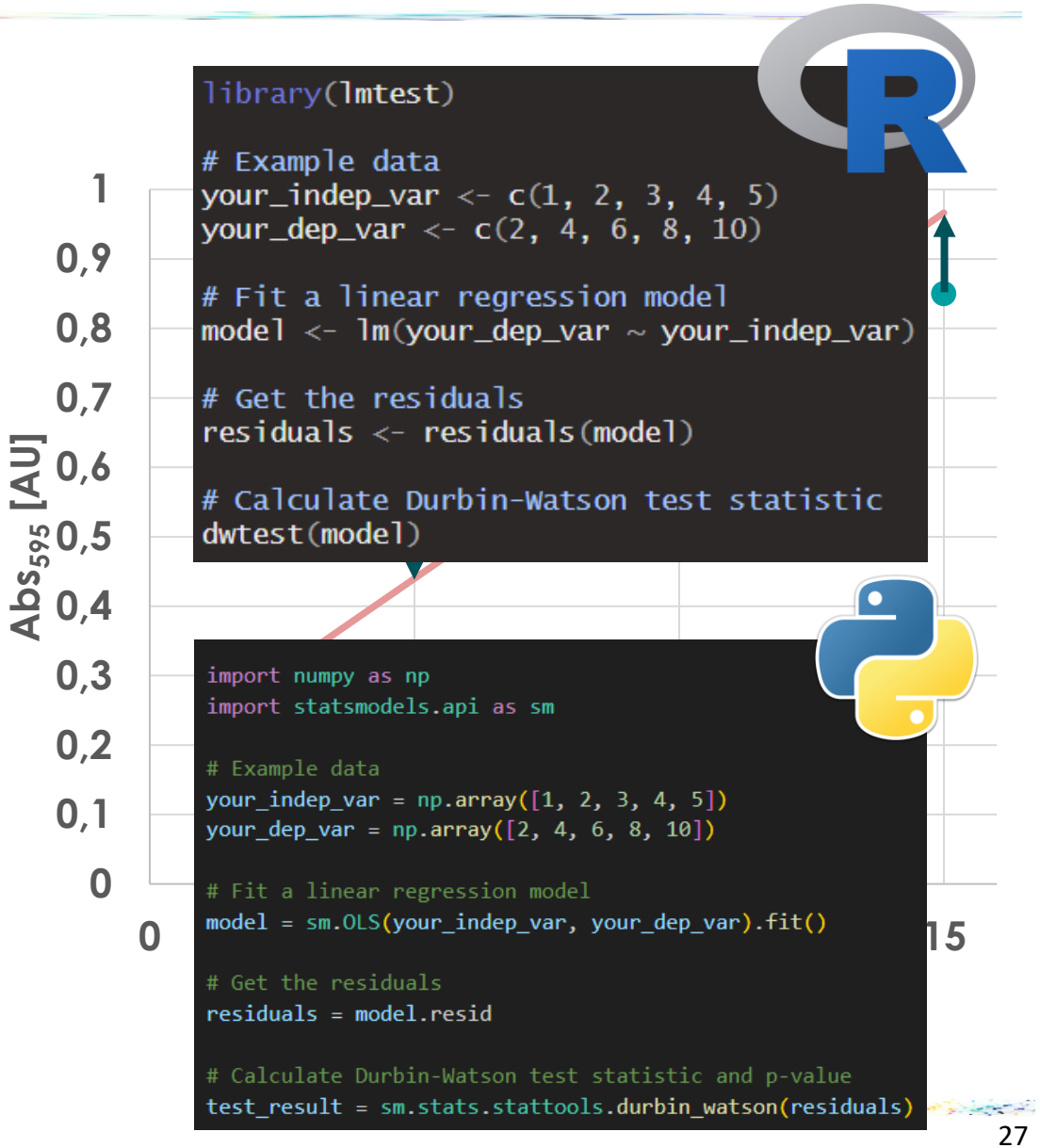
Linear Regression Models

Prerequisites for Linear Regression:

- **Durbin-Watson Statistic:** Test for **autocorrelation** of residuals → if residuals are autocorrelated errors of regression coefficients will be biased → increased risk of **masking factual influences**

- Does not return p value, but a **test statistic with [0, 4]** → values ~ 2 are assumed acceptable

- Durbin-Watson Statistic should be considered, when linear regression models other than calibration curves are done (e.g. **clinical data evaluation**)



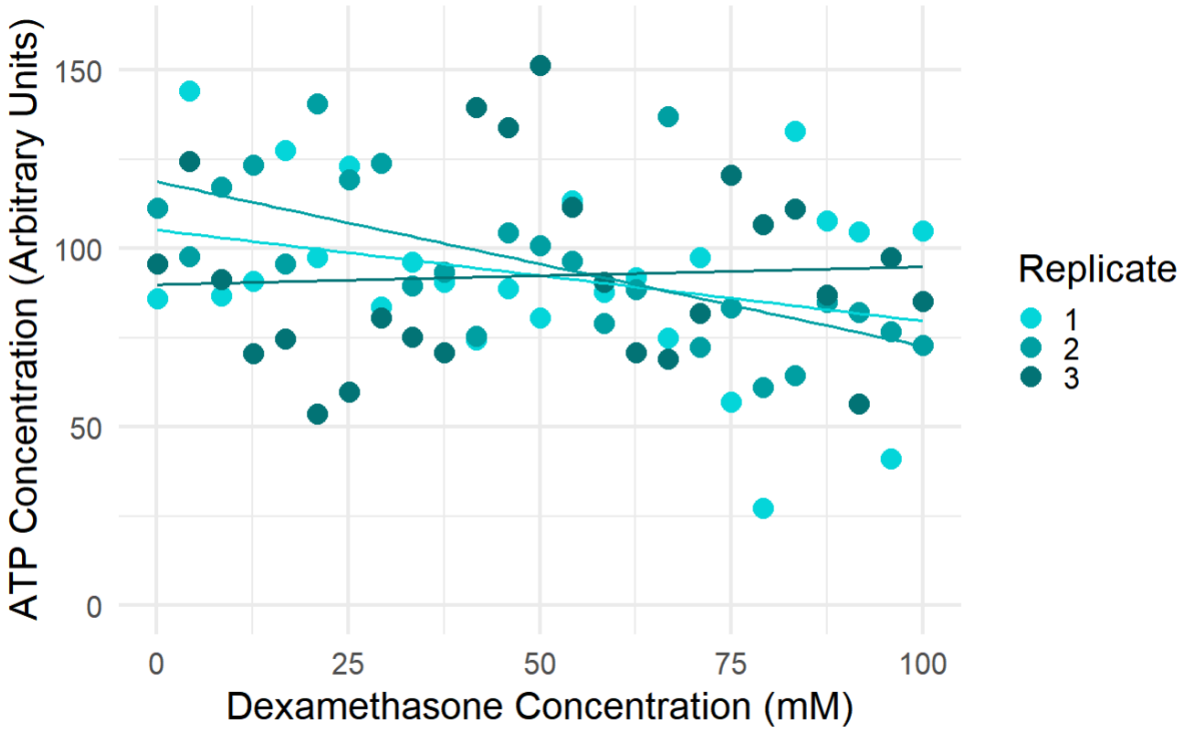
Linear Regression Models

Research Question: How does Dexamethasone (Dex) treatment affect the viability of JURKAT cells in culture?

Scenario: Measuring ATP concentration from cells incubated with 25 different Dex concentrations (dose escalation) with three biological replicates each

Linear Regression Models

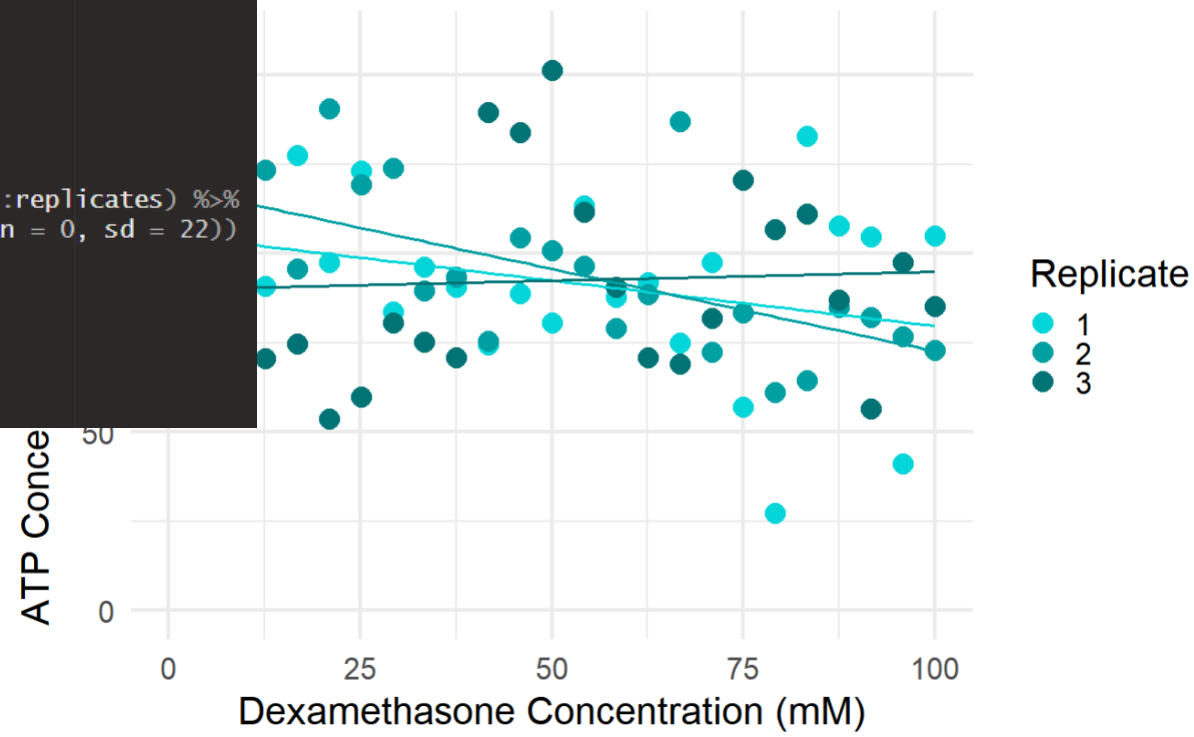
	Dex_Concentration	Replicate	ATP_Concentration
1	0.1000	1	86.981417
26	0.1000	2	98.331768
51	0.1000	3	90.042550
2	4.2625	1	75.939574
27	4.2625	2	72.158419
52	4.2625	3	150.612705
3	8.4250	1	98.965966
28	8.4250	2	81.823038
53	8.4250	3	96.032342
4	12.5875	1	93.391487
[illegible]	12.5875	2	93.071736
[illegible]	12.5875	3	129.644755



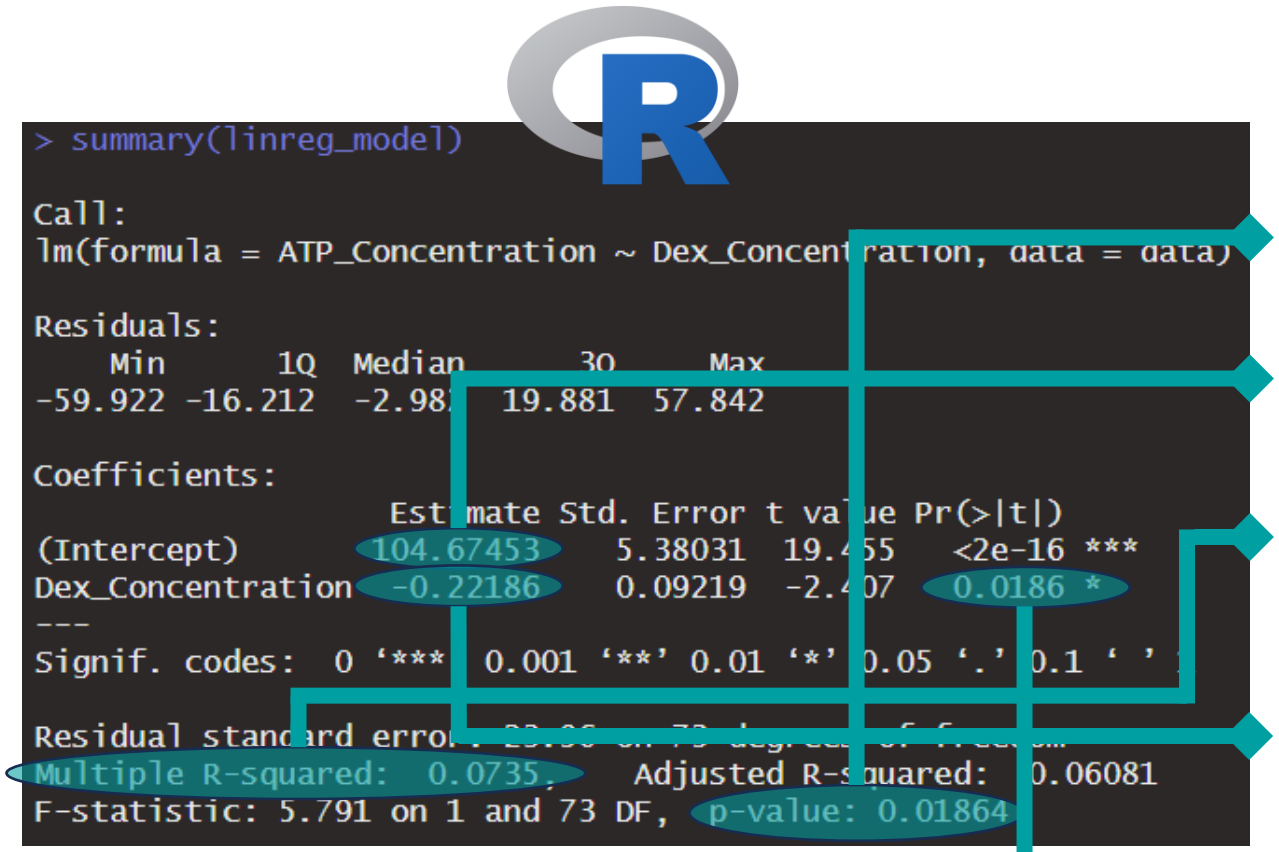
Linear Regression Models

```r
# Load required packages
library(tidyverse)
library(broom)

# Set seed for reproducibility
set.seed(123)

# Generate example data
concentration_levels <- seq(0.1, 100, length.out = 25)
replicates <- 3  # Number of biological replicates

# Generate data frame with Dex concentration and ATP concentration
data <- expand.grid(Dex_Concentration = concentration_levels, Replicate = 1:replicates) %>%
  mutate(ATP_Concentration = 100 - 0.1 * Dex_Concentration + rnorm(n(), mean = 0, sd = 22))

# Fit linear regression model
linreg_model <- lm(ATP_Concentration ~ Dex_Concentration, data = data)

# Summarize model output
summary(linreg_model)
```

```
> summary(linreg_model)

Call:
lm(formula = ATP_Concentration ~ Dex_Concentration, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-59.922 -16.212  -2.982  19.881  57.842

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     104.67453    5.38031  19.455   <2e-16 ***
Dex_Concentration  -0.22186    0.09219  -2.407   0.0186 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.96 on 73 degrees of freedom
Multiple R-squared:  0.0735,    Adjusted R-squared:  0.06081
F-statistic: 5.791 on 1 and 73 DF,  p-value: 0.01864
```



Linear Regression Models



Figures to Consider:

Model p value (p_{regr}): Does the model describe a relation that is different from a random effects-model?

Model Constant (C): Describes the error that is inherent to the model

Explanatory Power/Model Quality (R^2): Share of variance of the dependent variable that is explained by the predictors

Regression Coefficient (B) per indep. variable: Measure for strength of influence: If the predictor will increase by 1, this will cause the dependent variable to change by B

p value ($p_{a..n}$) per indep. variable: Does the predictor exert an effect which is distinguishable from a random effect?

Logistic Regression Models

Figures to Consider in a Regression Model:

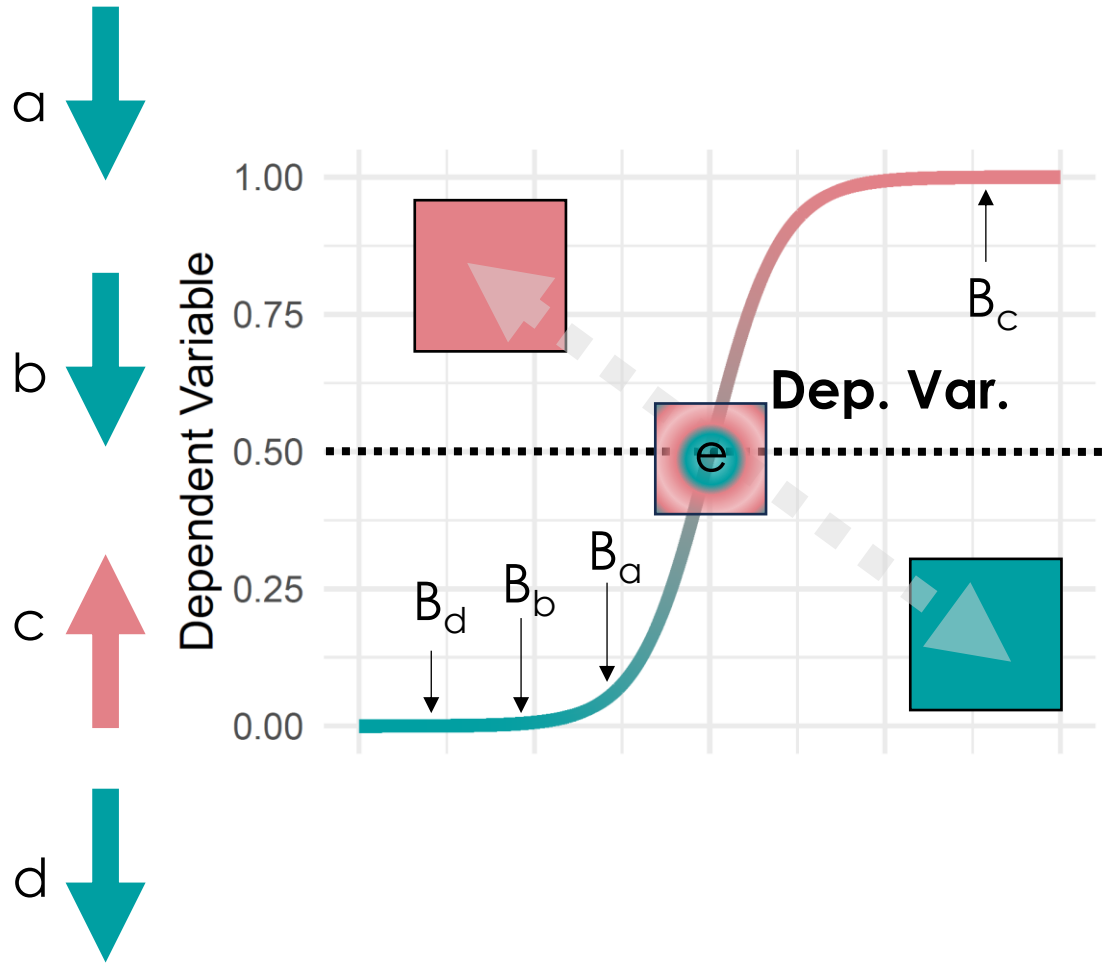
Model p value (p_{regr}): Does the model describe a relation that is different from a random effects-model? Expressed by Likelihood-Ratio Test

Model Constant (intercept): represents the log odds of the outcome when all predictors are at zero

Explanatory Power/Model Quality (AUC): Quality measure depending on sensitivity and specifity of the model

Regression Coefficient (B) per indep. variable: change in the log odds of the dependent variable for a one-unit change in the independent variable

p value ($p_{a..n}$) per indep. variable: Does the predictor exert an effect on the log odds which is distinguishable from a random effect?



Logistic Regression Models

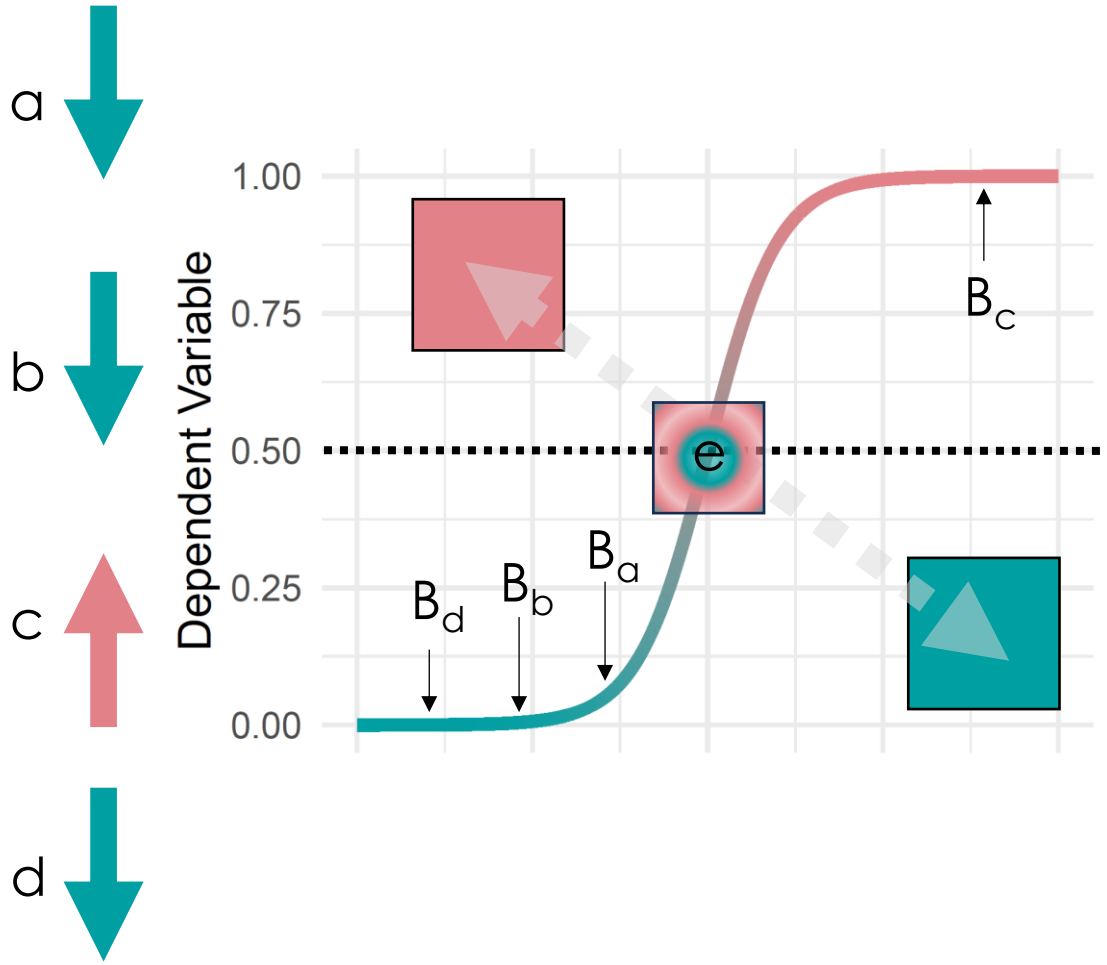
Research Question: Which factors are the most significant predictors of a protein's solubility in industrial production processes?

Scenario: For industrial protein production 200 conditions for pH, temperature, ionic strength and polarity index were documented. Their influence on protein solubility (precipitated or not) should be estimated.

Logistic Regression Models

```r
# Load necessary libraries
library(ggplot2)
library(caret)
library(pROC)
library(dplyr)

# Set seed for reproducibility
set.seed(123)

# Generate an example dataset
n <- 200 # Number of observations
data <- data.frame(
  pH = runif(n, 5.0, 8.0), # pH level of the solution
  temp = runif(n, 20, 80), # Temperature in Celsius
  ionic_strength = runif(n, 0.1, 1.0), # Ionic strength in molarity
  polarity_index = runif(n, 0, 1) # Polarity index of the solution
)

# Introduce a stronger effect for 'pH' to ensure significance
data$solubility <- as.factor(ifelse(data$pH < 6, 0,
                                    ifelse(data$pH > 7, 1,
                                    sample(0:1, n, replace = TRUE))))

# Conduct logistic regression
model <- glm(solubility ~ pH + temp + ionic_strength + polarity_index,
             family = binomial(link = "logit"), data = data)

# Assuming 'data' is your dataframe and 'model' is your fitted logistic model
# Conduct the likelihood ratio test
null_model <- glm(solubility ~ 1, family = binomial(link = "logit"), data = data)
lrt_result <- anova(null_model, model, test = "Chisq")

# Show the results of the likelihood ratio test
print(lrt_result)

# Model summary to get p-values
summary(model)
```

```
> summary(model)

Call:
glm(formula = solubility ~ pH + temp + ionic_strength + polarity_index,
    family = binomial(link = "logit"), data = data)

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-1.96628  -0.36335  -0.09227   0.37460   1.98610

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -22.04254    3.16572  -6.963 3.33e-12 ***
pH               3.60738    0.49766   7.249 4.21e-13 ***
temp            -0.01495    0.01283  -1.165    0.244
ionic_strength  -1.05414    0.90352  -1.167    0.243
polarity_index  -0.39193    0.77183  -0.508    0.612
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.18  on 199  degrees of freedom
Residual deviance: 127.62  on 195  degrees of freedom
AIC: 137.62

Number of Fisher Scoring iterations: 6
```



Logistic Regression Models

```
> print(lrt_result)
Analysis of Deviance Table

Model 1: solubility ~ 1
Model 2: solubility ~ pH + temp + ionic_strength + polarity_index
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       199     277.18
2       195     127.61  4   149.56 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> summary(model)

Call:
glm(formula = solubility ~ pH + temp + ionic_strength + polarity_index,
    family = binomial(link = "logit"), data = data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.96628  -0.36335  -0.09227   0.37460   1.98610

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -22.04254    3.16572  -6.963 3.33e-12 ***
pH               3.60738    0.49766   7.249 4.21e-13 ***
temp            -0.01495    0.01283  -1.165    0.244
ionic_strength  -1.05414    0.90352  -1.167    0.243
polarity_index  -0.39193    0.77183  -0.508    0.612
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.18  on 199  degrees of freedom
Residual deviance: 127.62  on 195  degrees of freedom
AIC: 137.62

Number of Fisher Scoring iterations: 6
```

Figures to Consider in a Regression Model:

Model p value (p_{regr}): Does the model describe a relation that is different from a random effects-model? Expressed by Likelihood-Ratio Test

Model Constant (intercept): represents the log odds of the outcome when all predictors are at zero

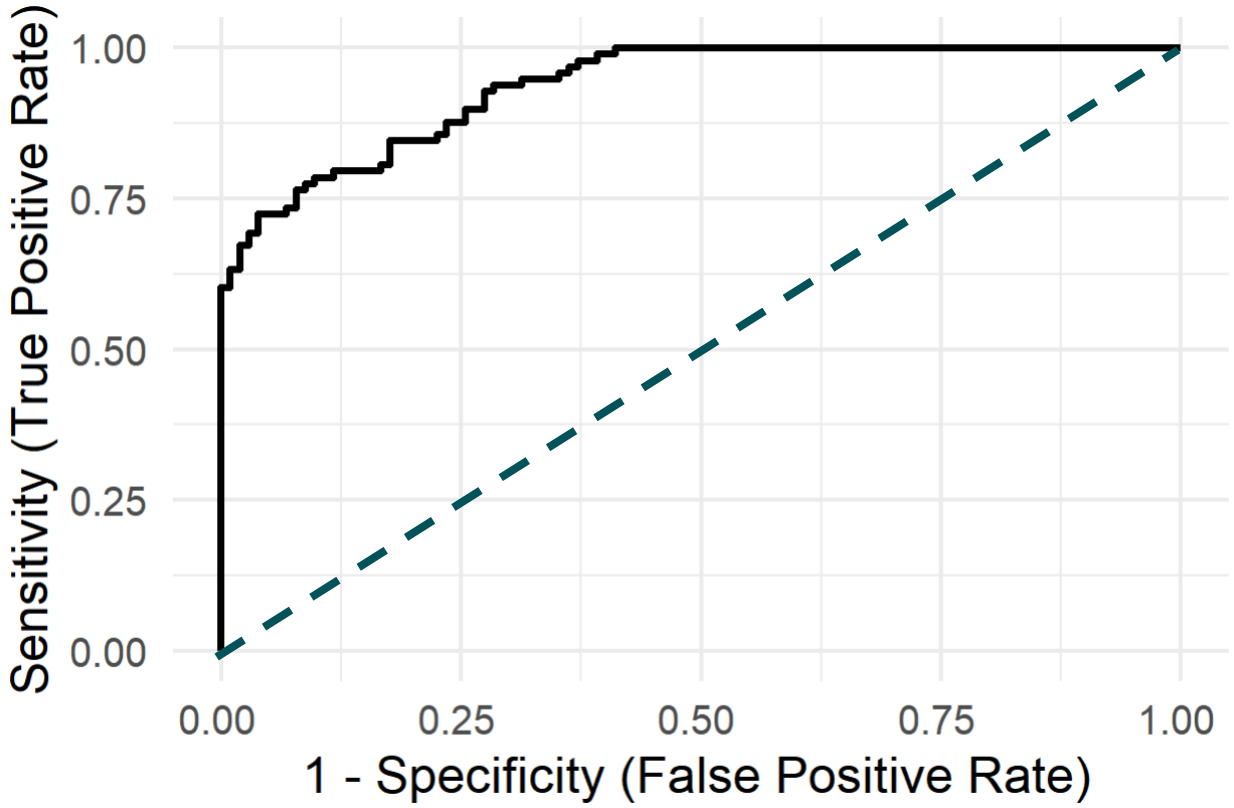
Explanatory Power/Model Quality (AUC): Quality measure depending on sensitivity and specifity of the model

Regression Coefficient (B) per indep. variable: change in the log odds of the dependent variable for a one-unit change in the independent variable

p value ($p_{a..n}$) per indep. variable: Does the predictor exert an effect on the log odds which is distinguishable from a random effect?

Logistic Regression Models

```r
# Calculate ROC
roc_obj <- roc(response = data$solubility, predictor = predictions)

# Inverting the ROC curve if it is mirrored
if (roc_obj$auc > 0.5) {
  roc_obj <- roc(response = as.numeric(as.character(data$solubility)) - 1,
                 predictor = predictions)
}

# Plot ROC curve
roc_plot <- ggplot() +
  geom_line(data = data.frame(specificities=1-roc_obj$specificities,
                              sensitivities=roc_obj$sensitivities),
            aes(x = 1-specificities, y = sensitivities), linewidth = 2) +
  labs(x = "1 - Specificity (False Positive Rate)", y = "Sensitivity (True Positive Rate)") +
  theme_minimal(base_size = 30)

# Show the ROC plot
print(roc_plot)
```



Figures to Consider in a Regression Model:

Model p value (p_{regr}): Does the model describe a relation that is different from a random effects-model? Expressed by Likelihood-Ratio Test

Model Constant (intercept): represents the log odds of the outcome when all predictors are at zero

Explanatory Power/Model Quality (AUC): Quality measure depending on sensitivity and specifity of the model

Regression Coefficient (B) per indep. variable: change in the log odds of the dependent variable for a one-unit change in the independent variable

p value ($p_{a..n}$) per indep. variable: Does the predictor exert an effect on the log odds which is distinguishable from a random effect?

Survival Analysis

- Kaplan-Meier Curves

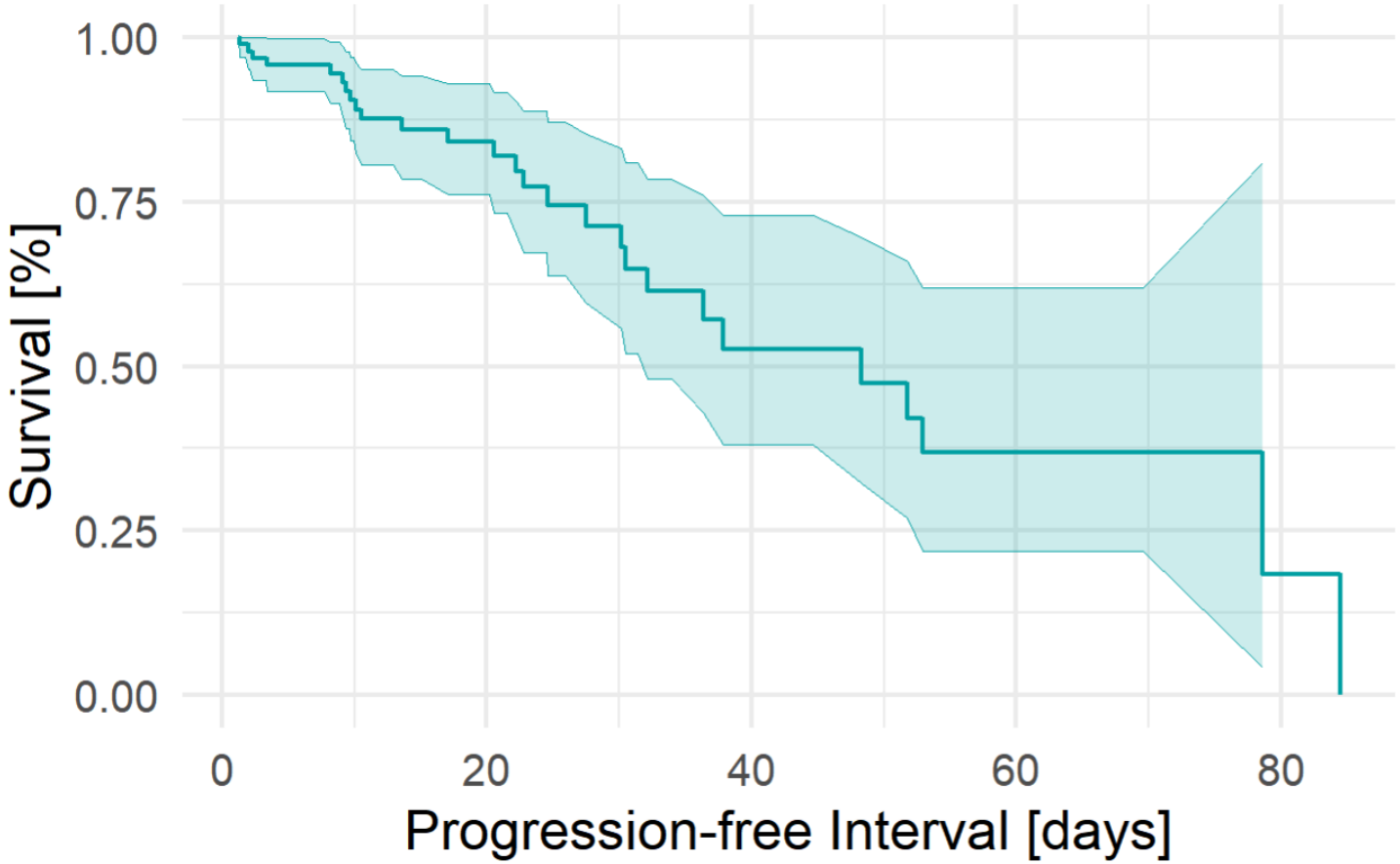
- Log Rank-Test

- Cox Proportional Hazard Analysis

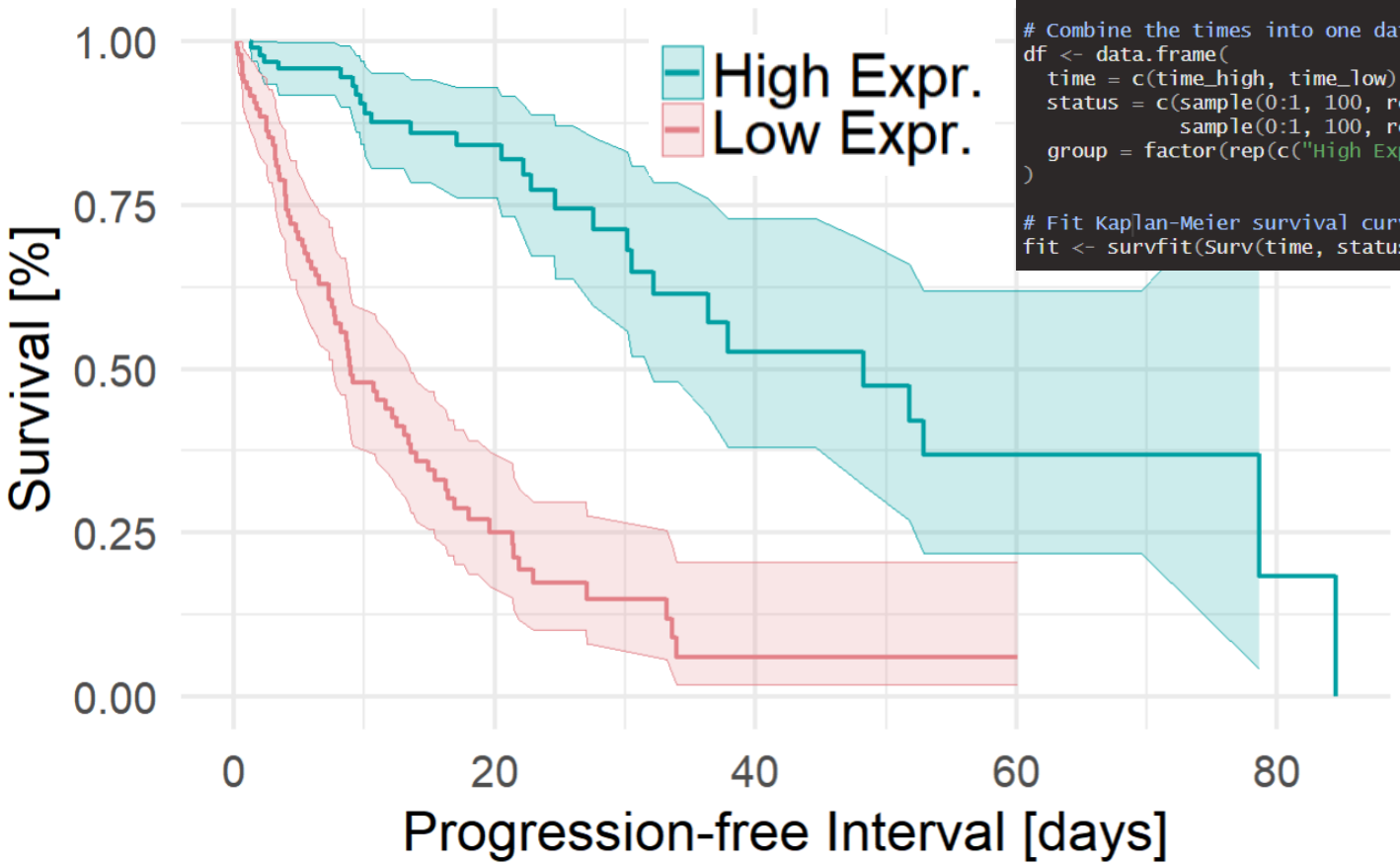
Survival Analysis

- Survival **= time to a certain event** (relapse, recidives, implant failure, death, etc.) which is influenced by one or more variables
- Survival Analysis: regression-like modeling of the **variable's influence on the occurence probability** of the event
- Graphical and Statistical Approaches

Kaplan-Meier Curves and Log-Rank Test



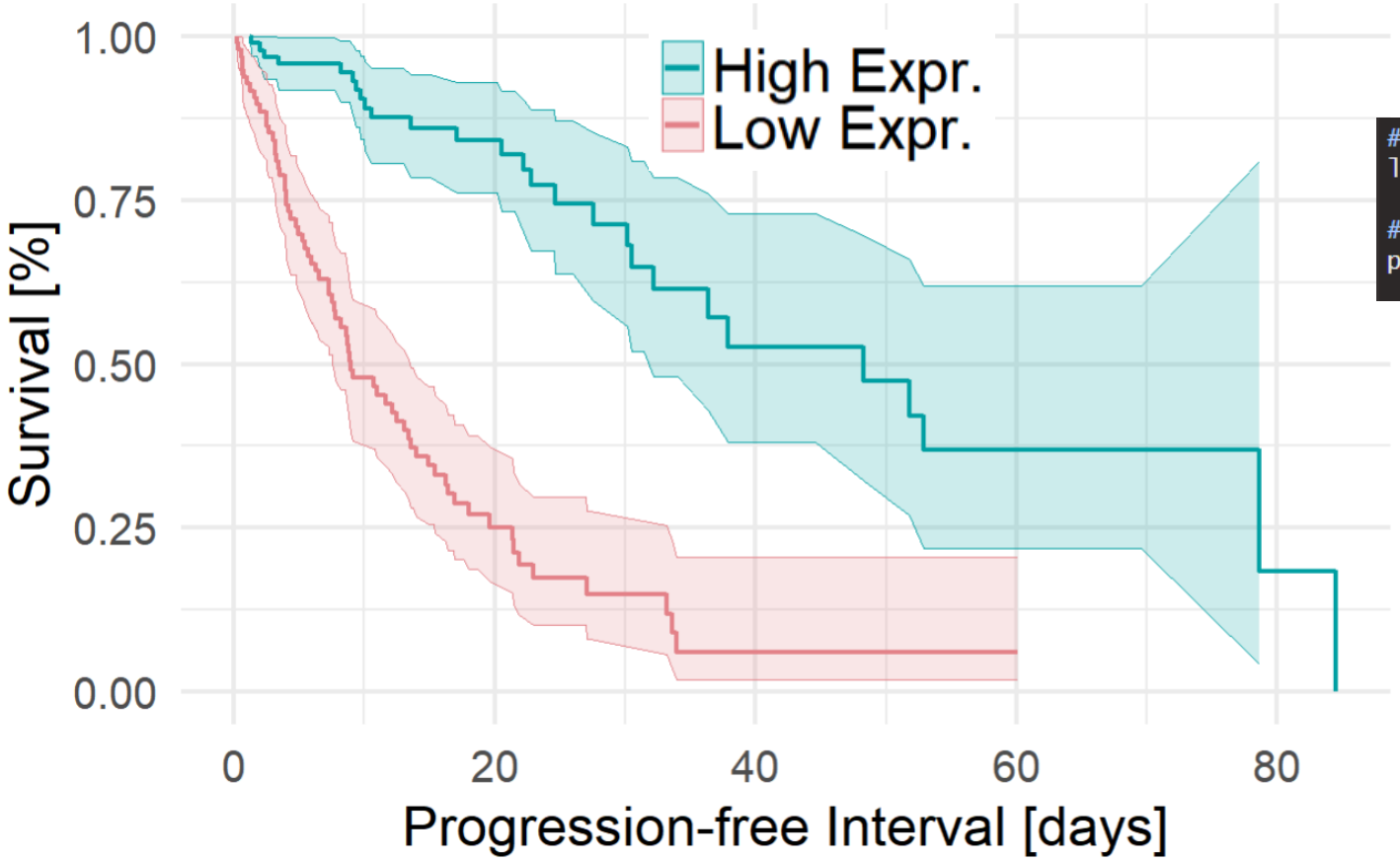
Kaplan-Meier Curves and Log-Rank Test




```r
library(survminer)
library(survival)
library(ggplot2)

# Generate a sample dataset with different survival characteristics for two groups
set.seed(12)

# Generate survival times
time_high <- rexp(100, rate = 0.05)  # Lower rate for "High Expr.", which implies better survival
time_low <- rexp(100, rate = 0.1)  # Higher rate for "Low Expr.", which implies worse survival

# Combine the times into one dataset and censor some data to simulate typical survival data
df <- data.frame(
  time = c(time_high, time_low),
  status = c(sample(0:1, 100, replace = TRUE, prob = c(0.7, 0.3)),  # More censoring (status 0) for "High Expr."
             sample(0:1, 100, replace = TRUE, prob = c(0.3, 0.7))), # Less censoring for "Low Expr."
  group = factor(rep(c("High Expr.", "Low Expr."), each = 100))
)

# Fit Kaplan-Meier survival curves for each group
fit <- survfit(Surv(time, status) ~ group, data = df)
```

Kaplan-Meier Curves and Log-Rank Test

How can we tell whether both groups have a **significantly different survival probability**?



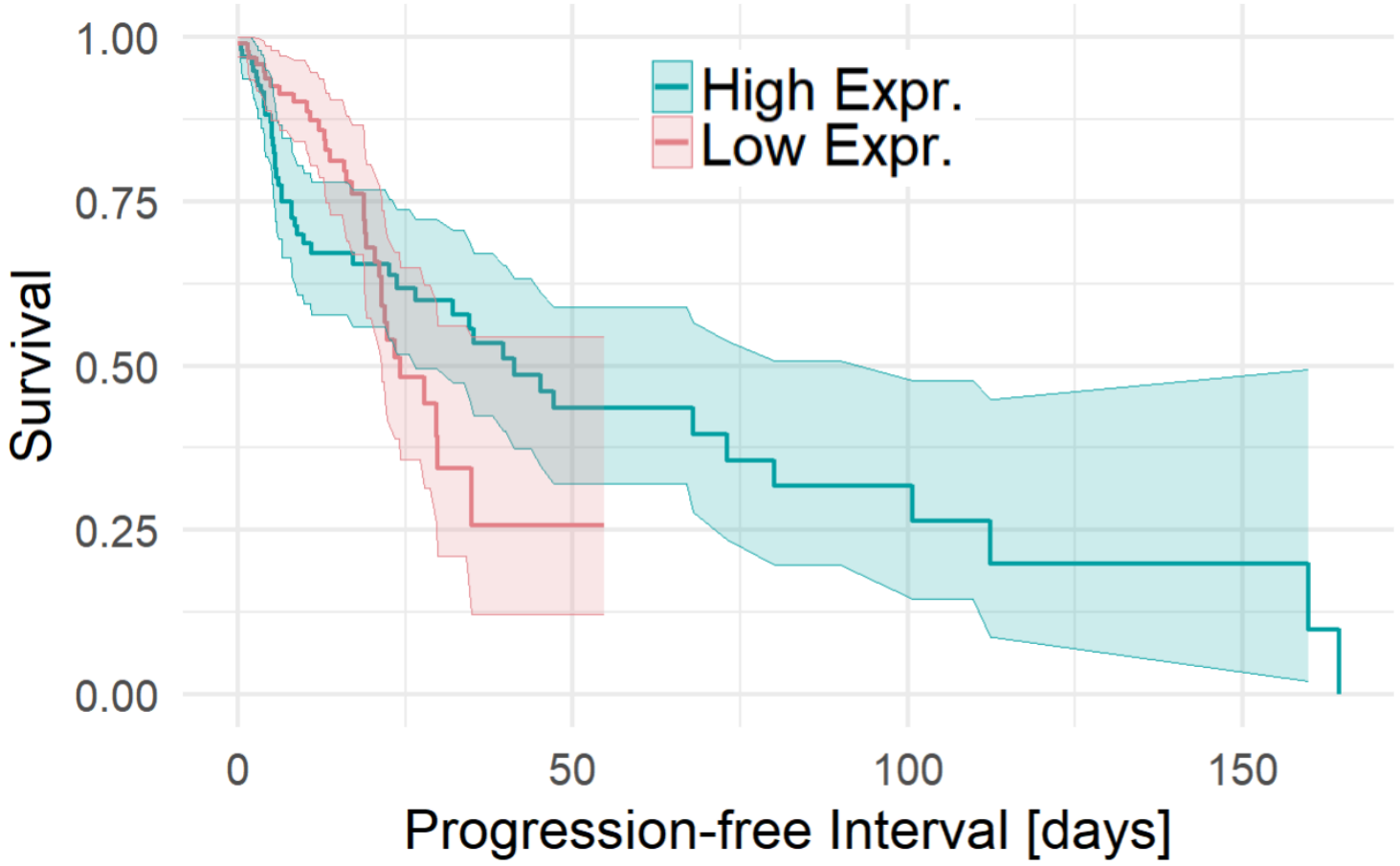
Log-Rank Test:

```
# Perform Log-Rank test
log_rank_test <- survdiff(Surv(time, status) ~ group, data = df)

# Print the result of the Log-Rank test
print(log_rank_test)
```

```
> print(log_rank_test)
Call:
survdiff(formula = Surv(time, status) ~ group, data = df)

                    N Observed Expected (O-E)^2/E (O-E)^2/V
group=High Expr. 100       27     62.9      20.5        62
group=Low Expr.  100       70     34.1      37.9        62

 Chisq= 62  on 1 degrees of freedom, p= 3e-15
```

Kaplan-Meier Curves and Log-Rank Test

Another example:



Log-Rank Test:

```
> print(log_rank_test)
Call:
survdiff(formula = Surv(time, status) ~ group, data = df)

                    N Observed Expected (O-E)^2/E (O-E)^2/V
group=High Expr. 100       46       47    0.0201    0.0564
group=Low Expr.  100       34       33    0.0286    0.0564

 Chisq= 0.1  on 1 degrees of freedom, p= 0.8
```

Crossing survival curves indicate changing hazard rates over time → **no classical survival analysis with dynamic hazard rates** → constant hazard rates are needed

Cox Proportional Hazard Analysis

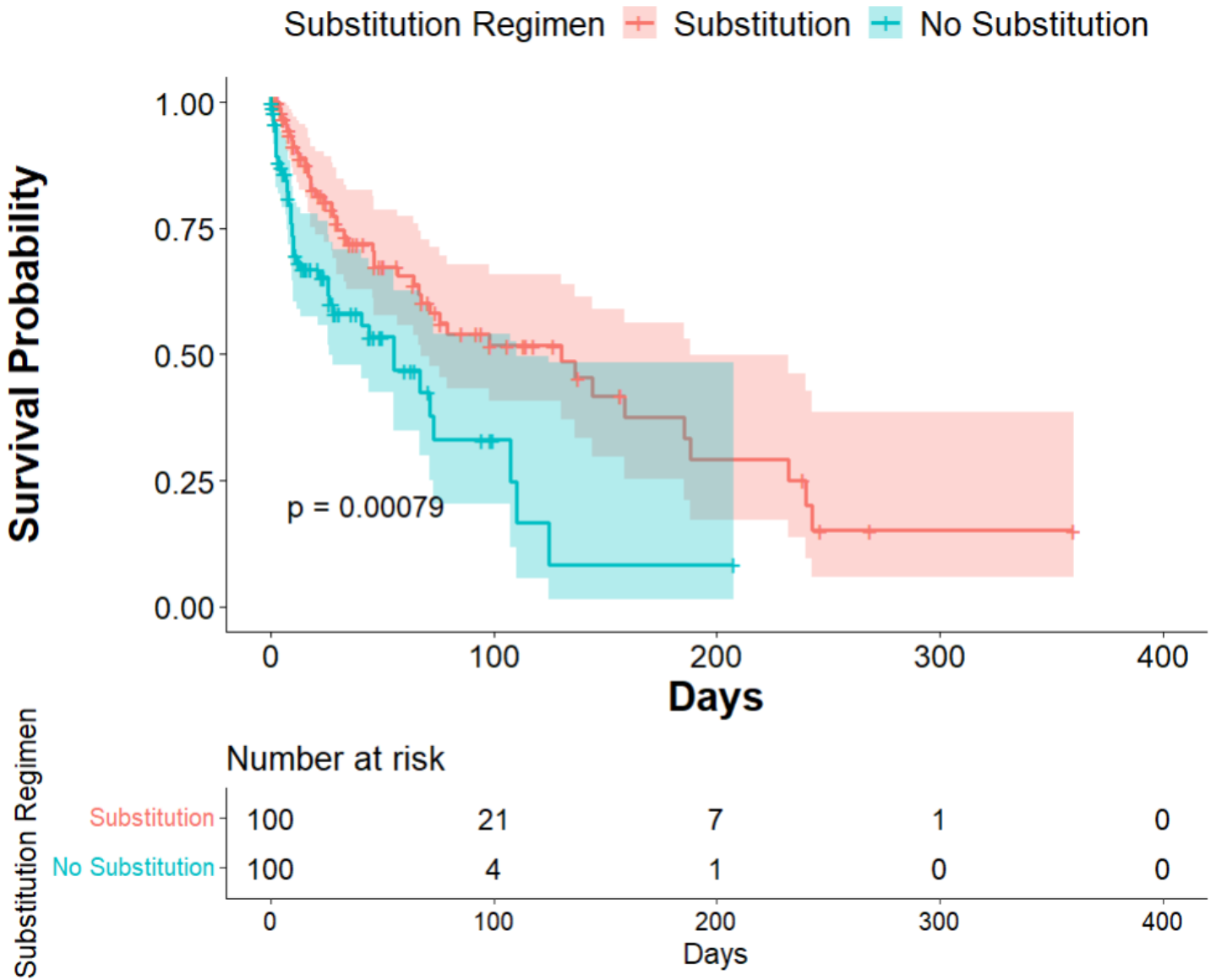
Research Question: How can we model the direct influence of a variable (e.g. number of daily cigarettes) on the probability of the event (relapse after withdrawal)?

Scenario: 200 smokers were randomly assigned to nicotine withdrawal with and without substitution regimen. Along with smoking habits and demographics, the (absence of a) substitution regimen should be evaluated as risk factor to premature relapse (360 days study).

Cox Proportional Hazard Analysis

```r
# Sample size
n <- 200

# Demographics
gender <- sample(c('Male', 'Female', 'Other'), n, replace = TRUE)
education_level <- sample(c('High School', 'Bachelor', 'Master', 'PhD'), n, replace = TRUE)

# Influence of education level on the risk of relapse
education_risk <- c('High School' = 1.2, 'Bachelor' = 0.9, 'Master' = 0.7, 'PhD' = 0.5)

# Smoking habits and regimen
smoking_regimen <- sample(c('Substitution', 'No Substitution'), n, replace = TRUE)
# Assuming substitution therapy has a protective effect, reducing the hazard rate
regimen_effect <- ifelse(smoking_regimen == 'Substitution', 0.5, 1)

# Age - older individuals might have a harder time quitting
age <- rnorm(n, mean = 45, sd = 12)
age_effect <- ifelse(age > 50, 1.6, ifelse(age < 30, 0.4, 1))

# Generate time to relapse based on age, education, and substitution regimen
time_to_relapse <- rexp(n, rate = 1/60) *
  sapply(education_level, function(x) education_risk[x]) *
  age_effect * regimen_effect
time_to_relapse <- pmin(time_to_relapse, 360) # Capping at 360 days for the study duration

# Determine event status
status <- ifelse(time_to_relapse < 360, rbinom(n, 1, prob = 0.4), 0)

# Create dataset
dataset <- data.frame(Gender = factor(gender),
                      Age = age,
                      EducationLevel = factor(education_level),
                      SmokingRegimen = factor(smoking_regimen),
                      CigarettesPerDay = rpois(n, lambda = 20),
                      YearsSmoking = rnorm(n, mean = 20, sd = 5),
                      TimeToEvent = time_to_relapse,
                      status = status)
```

Cox Proportional Hazard Analysis



Cox Proportional Hazard Analysis

```r
# For the Cox model, we'll need to ensure that categorical variables are factors
dataset$Gender <- as.factor(dataset$Gender)
dataset$EducationLevel <- as.factor(dataset$EducationLevel)
dataset$SmokingRegimen <- as.factor(dataset$SmokingRegimen)

# Fit the Cox model
cox_model <- coxph(Surv(time_to_event, Relapse) ~ Gender +
                   Age + EducationLevel + SmokingRegimen +
                   CigarettesPerDay + YearsSmoking, data = dataset)

# Display the summary of the Cox model
summary(cox_model)
```

```
> summary(cox_model)
Call:
coxph(formula = Surv(time_to_event, status) ~ Gender + Age +
    EducationLevel + SmokingRegimen + CigarettesPerDay + YearsSmoking,
    data = dataset)

  n= 200, number of events= 86

                              coef exp(coef)  se(coef)      z Pr(>|z|)
GenderMale                0.193686  1.213716  0.265200  0.730    0.465
GenderOther               0.024366  1.024665  0.273855  0.089    0.929
Age                       0.003757  1.003764  0.008334  0.451    0.652
EducationLevelHigh School -0.267299  0.765444  0.326001 -0.820    0.412
EducationLevelMaster       0.106207  1.112053  0.306890  0.346    0.729
EducationLevelPhD          0.212962  1.237337  0.315093  0.676    0.499
SmokingRegimenSubstitution 0.169774  1.185037  0.221038  0.768    0.442
CigarettesPerDay           0.028834  1.029254  0.022748  1.268    0.205
YearsSmoking              -0.011628  0.988440  0.021391 -0.544    0.587

                          exp(coef) exp(-coef) lower .95 upper .95
GenderMale                   1.2137     0.8239    0.7217     2.041
GenderOther                  1.0247     0.9759    0.5991     1.753
Age                          1.0038     0.9963    0.9875     1.020
EducationLevelHigh School    0.7654     1.3064    0.4040     1.450
EducationLevelMaster         1.1121     0.8992    0.6094     2.029
EducationLevelPhD            1.2373     0.8082    0.6672     2.295
SmokingRegimenSubstitution   1.1850     0.8439    0.7684     1.828
CigarettesPerDay             1.0293     0.9716    0.9844     1.076
YearsSmoking                 0.9884     1.0117    0.9479     1.031

Concordance= 0.558  (se = 0.038 )
Likelihood ratio test= 5.31  on 9 df,   p=0.8
Wald test            = 5.18  on 9 df,   p=0.8
Score (logrank) test = 5.22  on 9 df,   p=0.8
```

Cox Proportional Hazard Analysis

```
> summary(cox_model)
Call:
coxph(formula = Surv(time_to_event, status) ~ Gender + Age +
    EducationLevel + SmokingRegimen + CigarettesPerDay + YearsSmoking,
    data = dataset)

  n= 200, number of events= 86

                              coef exp(coef)  se(coef)      z Pr(>|z|)
GenderMale                0.193686  1.213716  0.265200  0.730    0 465
GenderOther               0.024366  1.024665  0.273855  0.089    0 929
Age                       0.003757  1.003764  0.008334  0.451    0 652
EducationLevelHigh School -0.267299  0.765444  0.326001 -0.820    0 412
EducationLevelMaster       0.106207  1.112053  0.306890  0.346    0 729
EducationLevelPhD          0.212962  1.237337  0.315093  0.676    0 499
SmokingRegimenSubstitution 0.169774  1.185037  0.221038  0.768    0 442
CigarettesPerDay           0.028834  1.029254  0.022748  1.268    0 205
YearsSmoking              -0.011628  0.988440  0.021391 -0.544    0 587

                           exp(coef) exp(-coef) lower .95 upper .95
GenderMale                    1.2137     0.8239    0.7217     2.041
GenderOther                   1.0247     0.9759    0.5991     1.753
Age                           1.0038     0.9963    0.9875     1.020
EducationLevelHigh School     0.7654     1.3064    0.4040     1.450
EducationLevelMaster          1.1121     0.8992    0.6094     2.029
EducationLevelPhD             1.2373     0.8082    0.6672     2.295
SmokingRegimenSubstitution    1.1850     0.8439    0.7684     1.828
CigarettesPerDay              1.0293     0.9716    0.9844     1.076
YearsSmoking                  0.9884     1.0117    0.9479     1.031

Concordance= 0.558  (se = 0.038 )
Likelihood ratio test= 5.31  on 9 df,   p=0.8
Wald test            = 5.18  on 9 df,   p=0.8
Score (logrank) test = 5.22  on 9 df,   p=0.8
```

Figures to Consider in a Cox Prop. Haz. Analysis:

Hazard Ratio (exp[coef]): Influence of the variable on the event probability (HR < 1: event-inhibiting; HR > 1: event-promoting)

Interpretation:

- non-metrical predictors: probability of event is increased by 1-exp(coef) compared to base category
- metrical predictors: probability of event increases by 1-exp(coef) if predictor increases by 1

p value (Pr(>|z|)) per indep. variable: Does the predictor exert an effect on the event probabilty which is distinguishable from a random effect?

C-Index (Concordance): Quality measure of the model; similar to AUC: 0.5 is "random effects", 1 is perfect explanation of events by chosen predictors

frank.hause@medizin.uni-halle.de

@fhause.bsky.social