Statistical Modelling Summary Sheet – HS 2023

Background – Linear Algebra & Calculus:

- **Trace**: $tr(A) = \sum A_{ii}$
  linear: $tr(aA + bB) = a\,tr(A) + b\,tr(B)$
  cyclic: $tr(ABC) = tr(BCA) = tr(CAB)$
- **Rank**: $rnk(A) =$ largest number of rows (columns) of A that constitute a linearly independent set.
- **Range space** $R(A) = \{y : y = Ax\}$
  Null space $N(A) = \{x : Ax = 0\}$
  - $\text{rnk}(A) = \dim(R(A)) = n - \dim(N(A))$
- **Orthogonal** matrix: $Q^{-1} = Q^T$
  - square matrix; cols and rows are orthonormal vectors
- **Symmetric** matrix: $A^T = A$
  - Spectral decomposition: $A = QDQ^T$
- **Quadratic form:** $x^T Ax$
- **Positive definite** (pd) matrix: $x^T Ax > 0 \quad for\ all\ x \in R^n \setminus \{0\}$
  Positive semi-definite (psd) matrix: $x^T Ax \geq 0 \quad for\ all\ x \in R^n$
  - Eigenvalues, trace, det are pos (non-neg)
  - Square root: $A^{-\frac{1}{2}}$, Cholesky decomposition ($A = L^T L$; with L lower triangular matrix)
- **Orthogonal Projection**:
  - idempotent ($P = P^2$), symm. ($P = P^T$)
  - $rnk(P) = tr(P)$
  - Eig.val. in $\{0,1\}$; geom. multiplicity of eig.val. 1 is $rnk(P)$
  - $1 - P$ is also projection (onto orthogonal complement of P); $P(1 - P) = 0$
  - If cols of B are basis of subspace S, og. projection on S is
  $$P_S = B(B^T B)^{-1} B^T$$
  (simple special case: ONB (orthonormal) → $P_S = QQ^T$)
- **Cauchy-Schwarz** Inequality: $|<u,v>|^2 \leq \ <u,u> \cdot <v,v>$
- **Differentiation (Fahrmeir, A.8):**
  $$\frac{\partial y^T x}{\partial x} = y;\ \frac{\partial x^T Ax}{\partial x} = (A + A^T)x;\ \frac{\partial Ax}{\partial x} = A^T;\ \frac{\partial Ax}{\partial x^T} = A$$

Background – Probability:

- **Expected Value**: $E[X] = \mu = \int x f(x)dx$;
  arithmetic mean: $\hat{\mu} = \frac{1}{n}\sum x_i$
- **Variance**: $Var(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$; empirical: $\widehat{var}(X) = \frac{1}{n-1}\sum(x_i - \hat{\mu})^2$
- **Covariance**: $Cov(X,Y) = \sigma_{X,Y} = E[(X - E[X])(Y - E[Y])^T]$; empirical: $\widehat{cov}(X,Y) = \hat{\sigma}_{X,Y} = \frac{1}{n-1}\sum(x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y)^T$ (in mult. dim. write: $\hat{\Sigma}_{X,Y}$)
- **Independence**: $X \perp Y \leftrightarrow P(X \cap Y) = P(X) * P(Y)$;
  In general: $X \perp Y \rightarrow Cov(X,Y) = 0$
- **Important pars:** $\mu,\ \sigma_X,\ \sigma_{X,Y},\ \Sigma_X$
- **Trafo Expectation:** $E(AY + a) = A\,E(Y) + a$
- **Trafo Covariance matrix:** $Var(AY + a) = A\,Var(Y)\,A^T$
- **Normal distribution (1-dim):** $Z \sim N(0,1)$; $Y = \mu + \sigma Z \rightarrow Y \sim N(\mu, \sigma^2)$ (Senn, B3)
  -rule of thumb: $\mu \pm 2 * \sigma$ covers about 95% of the CI
  Standardize: $Y \sim N(\mu, \sigma^2) \rightarrow Z := \frac{Y - \mu}{\sigma} \sim N(0,1)$
- **N (n-dim):** $Z = (Z_1, \ldots, Z_n),\ Z_i \sim N(0,1)\ i.i.d. \rightarrow Z \sim N(0, I_n)$
  $Y = \mu + AZ \rightarrow Y \sim N(\mu, \Sigma)$ with $\Sigma = AA^T$
- **Normal is special:**
  - $Cov(Y_i, Y_j) = 0 \ \leftrightarrow Y_i \perp Y_j$
  - $Y \sim N(\mu, \Sigma) \rightarrow A\,Y \sim N(A\mu,\ A\Sigma A^T)$
  - $Y \sim N(\mu,\ \sigma^2 1_{n*n})$, $nrow(A) + nrow(B) \leq n$:
  $U = AY,\ V = BY: U \perp V \leftrightarrow AB^T = 0$
- **Chi-square:** $Z \sim N(0,1)$; $Z^2 \sim \chi_1^2$; $\sum_{i=1}^n Z_i^2 \sim \chi^2$, if $Z_i \perp Z_j$;
  $Y \sim N(\mu, \Sigma) \rightarrow (Y - \mu)^T \Sigma^{-1}(Y - \mu) \sim \chi_n^2$
  degenerate case: If $e \sim N(0, M)$, $M \in R^{n*n}$ idempotent with $rnk(M) = r < n$. Then: $e^T Me \sim \chi_r^2$
- **F-distribution**: $X \sim \chi_m^2,\ Y \sim \chi_n^2,\ X \perp Y: \frac{\left(\frac{X}{m}\right)}{\frac{Y}{n}} \sim F_{m,n}$;
  Also: $T \sim F_{m,n}: E[T] = \frac{n}{n-2}$
- **T-distribution:** $Z \sim N(0,1),\ V \sim \chi_k^2,\ Z \perp V: T = \frac{Z}{\sqrt{\frac{V}{k}}} \sim t_k$;
  Also: $X \sim t_k \rightarrow X^2 \sim F_{1,k}$; t-distr. is like normal distr.
- **Classical CLT:** $X_i \sim F,\ E(X_i) = \mu,\ Var(X_i) = \sigma^2 < \infty,\ i.i.d$:
  $\sqrt{n}(\bar{X}_n - \mu) \rightarrow N(0, \sigma^2)\ (n \rightarrow \infty)$
  $\frac{1}{\sqrt{n} * \sigma}\sum(x_i - \mu) \rightarrow N(0,1)\ (n \rightarrow \infty)$
- ***Lindeberg CLT:** Similar for independent, but not identically distributed RVs (needs **Lindeberg's condition**)

Background – Statistics:

- **$\boldsymbol{\beta}$: true parameter, $\hat{\boldsymbol{\beta}}$ estimated par. (with «hat»)**
- **Point estimate: MLE** selects par. value which gives the observed data the largest possible probability (or prob. density in cont. case)
- **MLE has great properties (given some assumptions):**
  - **Consistent:** $\hat{\boldsymbol{\theta}} \rightarrow \boldsymbol{\theta}$ (in prob. as $n \rightarrow \infty$)
  - **Asy. Normal**: $\sqrt{n}(\hat{\theta} - \theta) \rightarrow N(0,\ I^{-1})$ (in distr. as $n \rightarrow \infty$) where I is Fisher Information
  - **Efficient**: Asymptotically Unbiased and smallest possible variance
- **Likelihood Ratio Test (LRT) has great properties** (given some assumptions):
  - Neyman-Pearson Lemma: LRT has **largest power**
  - **Asymptotic Distribution** is known

Hypothesis test

- **6 steps**: Model (r.v.'s + their distribution P_θ), hypotheses, test stat. & distribution (of the test statistic),level of significance α, rejection region, decision
- **Type 1 error**: H_0 true, but rejected;
  $$P(Type\ 1\ error) \leq \alpha$$
- **Type 2 error**: H_0 wrong, but no rejected;
  $$\textbf{power} = 1 - P(Type\ 2\ error)$$
  To compute power, you need **concrete alternative**
- In general: α smaller → power smaller
  Solution: More samples
- **One-sided test** can have more power than two-sided test, but "blind" on one side
- **p-value**: Assume H_0 is true, how likely is observation or something more extreme? (or: Smallest α with which we can reject)
  H_0 true: p-value is uniformly distributed on $[0,1]$
  H_1 true: small p-values are more likely
- $(1 - \alpha)$-**confidence interval** for parameter:
  Contains true parameter with prob $1 - \alpha$
  (or: all parameters, where H_0 is not rejected by a test at level α); rule of thumb for 95% interval: $\bar{X}_n \pm 2 * \frac{\hat{\sigma}_X}{\sqrt{n}}$
- **z-Test**:
  - Test for **expected value** μ of $N(\mu,\ \sigma^2)$;
  assume σ is known (usually unrealistic)

- Test statistic: $Z = \frac{\bar{X}_n - \mu_0}{\frac{\sigma_X}{\sqrt{n}}}$
- Assuming H_0: $Z \sim N(0,1)$

- **t-Test** (1 sample; this is Likelihood Ratio Test):
  - Test for **expected value** μ of $N(\mu,\ \sigma^2)$; σ **not known**
  - Test statistic: $\mathrm{T} = \frac{\bar{X}_n - \mu_0}{\frac{\hat{\sigma}_X}{\sqrt{n}}}$ (! hat on σ !)
  - Assuming H_0: $\mathrm{T} \sim t_{n-1}$
  - Usually wider distribution compared to z-test
- Problem of **multiple Testing** (on same data):
  FWER = $P(\#False\ positives \geq 0)$
  - at 95% level, 5% of tests will falsely (not) reject → false positives
  - Bonferroni correction (modified α value): Use $\tilde{\alpha} = \frac{\alpha}{m}$ level for every individual test => FWER=α

Measuring association

- **Pearson correlation:** $\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y}$ $(\sigma_{XY}: covariance)$
  - corrects for scaling of variances (cov. depeds on unit)
- $\rho_{X,Y} = 0$ → X and Y are «uncorrelated»; $|\rho_{X,Y}| \leq 1$
  If $|\rho_{X,Y}| = 1$, then $X = Y$ or $X = -Y$, i.e. points are on straight line → degree of *linear dependence*
  - **BUT**: does **not** capture all kinds of dependence!
- **Inference via Fisher z-Trafo:** (statistic)
$$Z = tanh^{-1}(\hat{\rho}) = \frac{1}{2} log\left(\frac{1+\hat{\rho}}{1-\hat{\rho}}\right)$$
  $\hat{\rho}$: empirical correlation
  - Assume $(X, Y) \sim N$: $Z \approx N\left(\tanh^{-1}(\rho), \frac{1}{n-3}\right)$
- **Spearman correlation**: Pearson cor. on ranks, detects monotonic relationsships
- Caveat:
  (1) Indep. → Cor=0 **but** Cor=0 ↛ Independent; equivalence holds for jointly normal random variables
  (2) Correlation ≠ causation

Simple Linear Regression (SLR)

- Regression: continuous response variable; continuous/categorical predictor variables
- Equivalent models:
  $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = \mu(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)\ i.i.d.$;
  $Y_i \sim N(\mu(x_i),\ \sigma^2)\ \ i.i.d.$ where $\mu(x) = \beta_0 + \beta_1 x$
- OLS: $\hat{\beta}$'s minimize $\sum\left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)\right)^2$ **(RSS)**
  RSS: sum of squared residuals
- If $\varepsilon_i \sim N(0, \sigma^2)\ i.i.d.$: **OLS = MLE** (they coincide)
- $\hat{\beta}_1 = \frac{\hat{\sigma}_{x,y}}{\hat{\sigma}_x^2}$; $\hat{\beta}_0 = \bar{y}_n - \hat{\beta}_1 \bar{x}_n$ → line through center of mass
- $\hat{\beta}_1 = \hat{\rho}_{XY} \cdot \frac{\hat{\sigma}_Y}{\hat{\sigma}_X}$ (→ slope is scaled correlation)
- Regression to mean: : $\frac{\hat{y} - \bar{y}_n}{\hat{\sigma}_Y} = \hat{\rho}_{XY} \cdot \frac{x - \bar{x}_n}{\hat{\sigma}_X}$
- Estimated coefs are random, **usually «wrong»**. Will never fit true exact value, since Y $(\epsilon \sim N)$ is random.

Multiple Linear Regression (MLR)

- Excplicit form:
  $Y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_{p-1} x_{p-1,i} + \varepsilon_i = \mu(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)\ i.i.d.$
- Vector form:
  $Y_i = x_i^T \beta + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)\ i.i.d.$
- **Matrix form:** (intercept: first X-col of only 1's)
  $\boldsymbol{Y = X\beta + \varepsilon, \quad \varepsilon \sim N(0,\ \sigma^2 \cdot 1)}$
- Transformations: (linearize non-linear functions)
  exponential type: $\log(y) = \tilde{y}$
  power type: $\log(y) = \tilde{y}$, $\log(x) = \tilde{x}$
  - for fitting non-linear data by linearizing (e.g. $y = \exp(x\beta + \epsilon)$; $regress\ \tilde{y} = \log(y)$; for prediction transform back)
  - example: $y = \exp(1 + 2\sin(x) + \epsilon)$; $\tilde{x} = \sin(x)$
- Thus: many complicated models can be represented as a linear model (by linearizing the data)
- OLS: $\hat{\beta} = argmin_\beta \sum_{i=1}^n \left(y_i - \beta_0 - \beta_1 x_{1,i} - \cdots - \beta_{p-1} x_{p-1,i}\right)^2 = \boldsymbol{argmin_\beta |Y - X\beta|^2} = argmin_\beta\ \boldsymbol{RSS}$
- Same as MLE if $\varepsilon_i \sim N(0, \sigma^2)\ i.i.d.$
- **Convex**: Gradient descent etc. works
- Analytical Solution of MLR:
  Normal eq.: $X^T(Y - X\hat{\beta}) = 0$ (from setting RSS to 0)
  Solution: $\boldsymbol{\hat{\beta} = (X^T X)^{-1} X^T Y}$
- Geometric interpretation: $\boldsymbol{\hat{Y}}$ **is orthogonal projection** $(H = X(X^T X)^{-1} X^T)$ of Y on hyperplane spanned by cols of X.
- **H**at matrix: $\hat{Y} = HY$; $tr(H) = p$
- Residual **M**aker: $\hat{\varepsilon} = MY$
  $M = 1 - H,\ M(1 - H) = 0,\ \ tr(M) = n - p$
  - M: projection onto orthogonal complement of H
- Consequences of og. projection (ass. Gaussian errors **CHECK**): (note: $Y = \hat{Y} + \hat{\varepsilon}$)
  - residual $\hat{\varepsilon}$: orth. distance of Y to column space
  - $\hat{Y} \perp \hat{\varepsilon}$, since $HM = 0$
  - $\hat{\beta} \perp \hat{\varepsilon}$
  - $\hat{\varepsilon} = M\varepsilon$ (Note: «Residual» ≠ «Error»)
  - If intercept: $E \cdot \hat{\varepsilon} = \sum_{i=1}^n \hat{\varepsilon}_i = 0$
- Pythagoras: TSS = ESS + RSS
  $TSS = |Y - \bar{Y}|^2$; $ESS = |\hat{Y} - \bar{Y}|^2$; $RSS = |Y - \hat{Y}|^2$
- $R^2 = 1 - \frac{RSS}{TSS} = \frac{ESS}{RSS}$ (last eq. only for OLS, not in general)
  $R^2 = cor(Y, \hat{Y})^2$ → Measure of how good a fit is
- Many SLR ≠ MLR (fitted parameters not always same)
  Interpretation of coefficients in MLR: «adjusted for other covariates»
  Special case – orthogonal covariates: SLR = MLR
- $\boldsymbol{\widehat{\sigma^2} = \frac{1}{n-p} \sum_{i=1}^n \hat{\varepsilon}_i}$, $\boldsymbol{E(\widehat{\sigma^2}) = \sigma^2}$;
  $n - p$: «degrees of freedom» (dim. of residual space)
- Factors with levels: Dummy coding wrt. *reference level*
  Factor variable: categorical values (categories=levels)
- Interaction btw. explanatory variables: «Effect» of one variable on response depends on the setting of the other variable
- $E(\hat{\beta}) = \beta$, $Cov(\hat{\beta}) = \sigma^2(X^T X)^{-1}$
  Note: $\beta_j = \frac{1}{1 - R_j^2} * \sigma^2 * \frac{1}{\sum(x_{ij} - \bar{x}_j)^2}$
- $E(\hat{Y}) = E(Y) = X\beta$, $Cov(\hat{Y}) = \sigma^2 H$
- $E(\hat{\varepsilon}) = 0$, $Cov(\hat{\varepsilon}) = \sigma^2 M$, $Cov(\hat{\varepsilon},\ \hat{Y}) = 0$
- $\widehat{\sigma^2} = \frac{1}{n-p} \sum_{i=1}^n \hat{\varepsilon}_i^2$ is unbiased estimate of σ^2
  → $\hat{\varepsilon}$ has non-const. variance ($\hat{\epsilon}_i$ correlated)
- Gauss Markov Conditions: Let $Y = X\beta + \varepsilon$, $E(\varepsilon) = 0$, $Cov(\varepsilon) = \sigma^2 I$, $rnk(X) = p$
  - under GMC: M is positive semi definite (psd)
- **GMT** (Gauss Markov Theorem) **– V1**: Let $Y = X\beta + \varepsilon$, $E(\varepsilon) = 0$, $Cov(\varepsilon) = \sigma^2 I$, $rnk(X) = p$, let $\ell \in R^p$: OLS estimator $\ell^T\ \hat{\beta}$ has minimal variance among **all linear unbiased** estimators of $\ell^T\beta$.
- **GMT – V2**: Let furthermore ε be **normally** distributed. Then $\ell^T\ \hat{\beta}$ has minimal variance among **all unbiased** estimators of $\ell^T\beta$. (aka. UMVU)
- Contrast: vector ℓ used to extract parameters for the case of a certain factor variable level
  (e.g. $\ell^T\beta = (1\ 0\ 1) * \hat{\beta}$)
- Caveat: Watch assumptions; bias-variance trade-off (small bias could give much lower variance)

- $\boldsymbol{\varepsilon_i \sim N(0, \sigma^2)}$ ***i.i.d.*** : (i.e. under GMC)
  - $\hat{\beta} \sim N_p(\beta,\ \sigma^2(X^TX)^{-1})$
  - $\hat{Y} \sim N_n(X\beta,\ \sigma^2 H)$, $\ \hat{\varepsilon} \sim N_n(0,\ \sigma^2 M)$
  - $\hat{Y} \perp \hat{\varepsilon}$
  - $\frac{\sum_{i=1}^n \hat{\varepsilon}_i^2}{\sigma^2} \sim X^2_{n-p}$
  - $\widehat{\sigma^2} \perp \hat{\beta}$
- $\boldsymbol{\varepsilon_i \sim F(0, \sigma^2)}$ ***i.i.d.*** : $\hat{\beta} \sim N_p(\beta,\ \sigma^2(X^TX)^{-1})$ ***asymptotically*** ($n \to \infty$) (and all consequences hold as above)

Tests

- **t-test** for $\boldsymbol{\beta_i}$: $\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}\sqrt{((X^TX)^{-1})_{ii}}} \sim t_{n-p}$

  $(1-\alpha) - \boldsymbol{CI}: \hat{\beta}_i \pm qt_{\frac{\alpha}{2};n-p} \cdot \hat{\sigma}_{\beta_i}$
- **CI** (confidence interval for $\boldsymbol{E(Y)}$): $\hat{Y}_0 \pm qt_{\frac{\alpha}{2};n-p} \cdot \hat{\sigma}_1$; where $\hat{\sigma}_1 = \hat{\sigma}\sqrt{x_0^T(X^TX)^{-1}x_0}$
  - CI: for a predicted $\hat{y}_0$ where will the true line be: $E[Y]$
- **PI** (prediction interval) for $\boldsymbol{Y}$: $\hat{Y}_0 \pm qt_{\frac{\alpha}{2};n-p} \cdot \hat{\sigma}_2$, where $\hat{\sigma}_2 = \hat{\sigma}\sqrt{1 + x_0^T(X^TX)^{-1}x_0}$
  - PI: for some specific x_0 (e.g.11) where an individual observation y_0 will lie. (CI describes position of the mean of such observations)
- **(Global) F-test** for $\boldsymbol{\beta}$: $\frac{(\hat{\beta}-\beta)^T X^T X(\hat{\beta}-\beta)}{p\,\widehat{\sigma^2}} \sim F_{p,n-p}$

  Test hypothesis: $H_0: (\beta_1, \dots, \beta_p) = 0\ \ vs.\ \ H_1: (\beta_1, \dots, \beta_p) \neq 0$ (i.e. do all variables have <u>any</u> effect on response y?)
  - F-test deals with several hypothesis tests at once)
- **Partial F-test**: $\frac{(B\hat{\beta}-b)^T\left(B(X^TX)^{-1}B^T\right)^{-1}(B\hat{\beta}-b)}{(p-q)\,\widehat{\sigma^2}} \sim F_{p-q,n-p}$

  $$\frac{\frac{SSE_0 - SSE}{p-q}}{\frac{SSE}{n-p}} \sim F_{p-q,n-p}$$

  Test hypothesis: $H_0: B * \beta = b(=0)\ vs.\ H_1: B * \beta \neq b$ (i.e. f-test for some variables); alternatively test for each row, but don't forget mult. testing correction!

Residual Analysis

- Errors $\varepsilon \neq \hat{\varepsilon}$ Residuals
- Standardized Residuals: $\hat{\varepsilon}^s{}_i = \frac{\hat{\varepsilon}_i}{1-H_{ii}}$

  Then: residuals have constant variance
- Serial correlation: $\hat{\varepsilon}_i$ vs. time
- Tukey-Anscombe Plot: $\hat{\varepsilon}_i$ vs. $\hat{y}_i$

  Best case: no visible pattern (then $\hat{\varepsilon}_i$ and $\hat{y}_i$ are probably uncorrelated)
  - TA-plot to detect heteroscedasticity (i.e. changing error variance); quick fix sometimes: use $\tilde{y} = \log(y)$ to squeeze errors in some areas.
- Scale-Location Plot: TA-plot divided by residual variance
- Normal QQ-plot of residuals: If $\hat{\varepsilon}_i \sim N(\mu,\ \sigma^2)$, then: $q_X = \mu + \sigma q_Z$

  Where: q_X:empirical quantile; q_Z: theoretical quantile
  - plot empirical quantiles of residuals, and compare to theoretical quantiles. Good fit: linear QQ-plot.

Model Selection

- Sparse model might predict better (**more variables** $\neq$**better model**); Best: only use relevant variables
  - more variables decrease bias, but sparse models with small bias can have very low variance.
- Watch out for multiple testing issue!
- Cave: Post-selection inference problematic !

- $MSE = Var + Bias^2$ (Mean Squared Error)
- $SSE = RSS = E(\sum(y_i - \hat{y}_{iM})^2$

  $\boldsymbol{M}$: matrix with subset of variables (i.e. columns of X)
- $SMSE = \sum E(\hat{y}_{iM} - \mu_i)^2 = \sigma^2|M| + \sum(\mu_{iM} - \mu_i)^2 = Var + Bias^2$ («sum of mean squared error»)
- $SPSE = \sum E(y_{n+1} - \hat{y}_{iM})^2 = n\sigma^2 + |M|\sigma^2 + \sum(\mu_{iM} - \mu_i)^2 = I + Var + Bias^2$ (I: irreducible error) ("expected squared prediction error" of future obs.)
  - note: $E[RSS] = SPSE - 2|M|\sigma^2$
  - And: $SPSE = n\sigma^2 + SMSE$
- $C_p = \frac{SSE}{\widehat{\sigma^2}} + 2|M| - n$ (estimates $\frac{SMSE}{\sigma^2}$; $C_p \approx |M|$ is unbiased)

  Use full model for $\hat{\boldsymbol{\sigma}}$; smaller C_p → better prediction
- $AIC = -2 \cdot l(\hat{\theta}_M) + 2 \cdot p$; smaller AIC → better prediction
- $BIC = -2 \cdot l(\hat{\theta}_M) + n \cdot \log(p)$ (same properties AIC)
- Intuition for good criterion: roughly minimize $RSS + const * p$ (p: number of parameters)
- Best: fit several models, and compare them with AIC or C_p score (scores: «distance from true model»)
  - note: correct p-value for mult. testing
- Model **search strategies**:
  - **exhaustive**: computationally expensive
  - **forward selection**: add one variable at a time; first compute all models with one variable, choose the best e.g. AIC/Cp score, then compute all models with two variables which include the previously chosen variable and so on...
  - **backward selection**: start with the full model, then delete one variable at a time; stop if AIC or Cp doesn't improve

Non-iid errors

- Detect with residual analysis (e.g. TA/QQ-plot), or with context knowledge
- Errors with known covariance matrix → Generalized Least Squares (GLS) / Weighted Least Squares (WLS): Assume $\mathrm{Cov}(\varepsilon) = \sigma^2 W^{-1}$ (i.e.reparametrization), then:
  - $\hat{\beta} = (X^TWX)^{-1}X^TWy$
  - $Cov(\hat{\beta}) = \sigma^2(X^TWX)^{-1}$
  - $\hat{\sigma}^2 = \frac{1}{n-p}\ \hat{\varepsilon}^T W\ \hat{\varepsilon}$
- Special case (i.e. WLS): Grouped data

  Weight ~ Variance ~ $\frac{1}{n_i}$
  - some observations in data are averages, thus we need to adapt their variance e.g.$W^{-1} = diag\left(\frac{1}{100}, 1,1\right)$; $var(\epsilon_i) = \frac{\sigma}{w_i}$; w_i: number of samples
- Errors where structure of coviarance matrix is known:
  - two-stage procedure
  - Maximum-Likelihood
- Errors with unknown covariance matrix: use (e.g. Sandwich) estimates that are consistent even under *certain* violations of assumptions

  → Heteroskedasticity consistent (HC) estimator (e.g. sandwich estimator); they estimate the cov matrix if there is diagonal heteroscedasticity (in errors)

Linear Mixed Models

- Way of dealing with known structure in cov. mat. of errors
- Focus on population average ("fixed effect") and person-specific random ("random effect") variations
- Random Intercept (RI):

  $y_{ij} = (\beta_0 + u_i) + \beta_1 x_{ij} + \varepsilon_{ij}$,

  $\varepsilon_{ij} \sim N(0, \sigma^2)$, $\ u_i \sim N(0, \sigma_1^2)\ \ i.i.d$

- model each person as r.v. added to population mean

- Random Intercept and Random Slope (RIRS):
  $y_{ij} = (\beta_0 + u_{1,i}) + (\beta_1 + u_{2,i})x_{ij} + \varepsilon_{ij}$,
  $\varepsilon_{ij} \sim N(0, \sigma^2)\ i.i.d$
  $u_{1,i} \sim N(0, \sigma_1^2)$, $u_{2,i} \sim N(0, \sigma_2^2)$, $cor(u_1, u_2) = \rho$
- Estimation: ML for model comparison, REML for final fit (to get unbiased variance estimates!!)
- LMM implicitly model correlations among same person
- longitudinal data: several observations per person over time
- clustered data: several observations for each cluster (e.g. hospital, school, district)

Generalized Linear Models (GLM): Logistic Regression

- S: $Y \sim Bin(1,\ p(x))$
- D: $p(x) = \frac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)} \rightarrow \log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$
  - note: we model log-odds in logistic regression
- 3 parts: Distribution, link function, linear predictor
- $odds(A) = \frac{P(A)}{1-P(A)}$; Log-odds: $\log\left(\frac{P(A)}{1-P(A)}\right)$;
  Odds-ratio: $\frac{odds(A|B)}{odds(A|B^C)}$
  - note: probability larger → (log-)odds larger
- Latent variable model: $Z_i = x_i^T\beta + \varepsilon_i$ (want to model Z, but only get partial information Y_i about it)
  observe: If $Z_i > 0: Y_i = 1$; if $Z \le 0: Y_i = 0$
  $\varepsilon \sim Logistic(0,1) \rightarrow$ Logistic Regression
  $\varepsilon \sim N(0,1) \rightarrow$ Probit Regression
- Estimate Maximum Likelihood using e.g. Fisher scoring (iterative optimization), IRLS
- Inference: ML properties $\hat{\boldsymbol{\beta}} \sim N(\beta,\ V(\beta))$, where V is inverse Fisher information
- Model comparison: Deviance, AIC
- **General form of GLMs**
  S (stochastic): $Y \sim F(1,\ p(x))$
  D (deterministic): $g(\mu(x)) = x_i^T\beta$ $(or\ \mu(x) = h(x_i^T\beta))$
  - 3 parts: Distribution F (must be in exp. Fam.), link function $g(x)$, linear predictor β
- Further examples: Poisson Regression, Gamma Regression; technical details as in Logistic Regression (but different distribution, and link function)
- Poisson regression (e.g. for counting incidents):
  Assume $Y_i \sim Pois(\lambda_i) \rightarrow\ \mu_i = E(Y_i) = Var(Y_i) = \lambda_i$.
  Problem: if we expect larger values, their variance will grow aswell!
  $\mathbf{S}: Y_i\ \sim\ Pois(\lambda_i);\ \mathbf{D}: \log(\mu_i) = \log(\lambda_i) =\ x_i^T\beta$
- Quasipoisson regression: address problem of growing variance (alternatively: neg. bin. distr.), by introducing the dispersion param: $E(Y_i) = \Phi * Var(Y_i)$
- **Residual deviance**: $-2 * (l(\beta_{hat}^s) - l(\beta_{hat}))$
  - difference of log-likelihoods of saturated model β^S (#param=#observ.) and our model β
  - base-level: res.dev. of β^S to null model(only intercept)
  - deviance residual d_i for every parameter: describes how much of an outlier observation Y_i acc.to our model

Nonlinear regression

- $y_i = f(x_i,\ \theta) + \varepsilon_i$,
  $E(\varepsilon_i) = 0$ and $Cov(\varepsilon_i) = \sigma^2 I$ (→ apply asymptotic normality) or
  $\varepsilon_i \sim N(0, \sigma^2)\ iid$ (then OLS = MLE) (→apply MLE)
- For of $f(x_i,\ \theta)$ is *given* from context
- Linearization: Check residuals for appropriate error structure; doesn't always work (wrt. to CI) ! (additive vs. multiplicative)
  e.g. apply $\tilde{y} = \log(y)$ to make mult. errors additive
- Fitting:
  - In general non-convex
  - Numerical methods needed
  - E.g.: Gauss-Newton Method (sequence of linear approximations)
- Starting values:
  - experience / linearized fun / «meaning» of pars and data
  - self-starting functions for many non-linear functions (implemented in R; choose very good initial values)
- Inference based on linear approximation: Use tangent plane (X-col space is a manifold, not linear; project Y onto tangent plane)
  $$\hat{\theta} \approx N\left(\theta_0,\ \hat{\sigma}^2\left(A(\hat{\theta})^T A(\hat{\theta})^{-1}\right)\right)$$
- Wald-type CI («all pars close to $\hat{\theta}_k$»): (i.e. $|\theta_0 - \hat{\theta}|$)
  $$\hat{\theta}_k \pm t_{n-p;1-\frac{\alpha}{2}}\, se(\hat{\theta}_k)$$
  - assumes symmetric CI (not always the case!)
- Improved inference based on Likelihood:
  («all pars where drop in likelihood from $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}$ is not too big»)
  $T = \left(\frac{(n-p)}{p}\right) \cdot \frac{S(\theta^*)-S(\hat{\theta})}{S(\hat{\theta})} \approx F_{p,n-p}$ (this is a test statistic)
  S: drop distance onto the linear tangent plane
- Single parameter: Profile likelihood («optimize out» all other pars): fix all param. except one, and optimize Likelihood profile traces
  $$T_k(\theta_k^*) = sign(\hat{\theta}_k - \theta_k^*) \cdot \frac{\sqrt{\tilde{S}_k(\theta_k^*) - S(\hat{\theta})}}{\hat{\sigma}} \approx t_{n-p}$$
  $H_0: \theta_k = \theta_k^*$; T_k is the corresp. Test statistic
- Profile t-plot: Check severity of non-linearity at θ_k^*
  Plot $T_k(\theta_k^*)$ vs. $\delta_k(\theta_k^*) = \frac{\hat{\theta}_k - \theta_k^*}{\hat{s.e.}(\hat{\theta}_k)}$,
  δ_k is linear approx. from above; expect straight line in linear setting
  - profile traces of e.g. b_1, b_2 almost parallel, then they are redundant → reparametrization
  - linearity of $T_k(\theta_k)$ indicates symmetry of CI of θ_k
    → very linear: Wald-type CI conincides with profile-likelihood CI

Nonparametric regression

- $y_i = f(x_i) + \varepsilon_i$ (no info about functional form),
  $E(\varepsilon_i) = 0$ and $Cov(\varepsilon_i) = \sigma^2 I$ or
  f twice cont. diff.able
- Use weighted moving window averages (weight: kernel)
- Kernel regression: Nadaraya-Watson kernel estimator
  Solves: $\hat{f}_n(x) = argmin_{m_x \in R} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)(Y_i - m_x)^2$
  $\rightarrow \hat{f}_n(x) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)Y_i}{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}$; weights $w_i(x) = K\left(\frac{x-x_i}{h}\right)$
- Common Kernels: Rectangle, Gauss, Epanechnikov (optimal, but very similar to gauss),...
- Bandwidth h governs Bias-Variance trade-off; choose with cross-validation
  - h: corresp. to window length; ***h* small** →less bias, more variance; ***h* large** → more bias, less variance
  -because: $Bias\left(\hat{f}(x)\right) \approx h^2 f''(x) * Const(kernel)$;
  $Var\left(\hat{f}(x)\right) \approx \frac{1}{nh} * Const(kernel)$ (if K symmetric)
- Nonparametric pays a price in convergence rate of MSE: parametric: $MSE \sim n^{-1}$; nonparametric: $MSE \sim n^{-\frac{4}{5}}$
- Inference using $\hat{Y} = SY$:

S: “linear smoother”, applies weight w_i to all y for a given x_i (center of window)
Approx. 95%-CI for $E(\hat{f}(x_i))$ (!) : $\hat{f}(x_i) \pm 1.96 \cdot \widehat{s.e.}\left(\hat{f}(x_i)\right)$

$\widehat{s.e.}\left(\hat{f}(x_i)\right) = \hat{\sigma} \cdot \sqrt{SS_{ii}^T}$,

$\hat{\sigma} = \sum_{i=1}^{n} \frac{\left(Y_i - \hat{m}(x_i)\right)^2}{n-df}$, where $df = tr(S)$

- Local Polynomial regression: fit polynomial inside each window, and take intercept as average value.
- Smoothing spline:
  Optimize penalized residual sum of squares (RSS):
  $\sum_{i=1}^{n}\left(Y_i - m(x_i)\right)^2 + \lambda \int m''(z)^2 dz$
  Solution: ($m =$) Natural cubic spline optimizes the above term; found as penalized regression problem
- Only use if dim ≤ 2: No near neighbors in high dims (because distances increase by a lot)
- Additive Models (GAMs):
  Additive model (AM) with non-par regression **per dimension** (assume each dim. can be modeled individually; independent terms in $f(x)$.

High-dimensional Regression

- Penalized least squares / regularization;
  Penalty: Increased bias but phps smaller variance
  → penalry might improve bias-variance trade-off
- Equivalent forms: (OLS, penalty)
  - Minimize $PLS(\beta) = (y - X\beta)^T(y - X\beta) + \lambda \cdot pen(\beta)$
  where $\lambda \geq 0$ is a tuning parameter
  - Minimize $(y - X\beta)^T(y - X\beta)$
  subject to $pen(\beta) \leq s$
- Ridge regression:

$$(y - X\beta)^T(y - X\beta) + \lambda \cdot \sum_{j=1}^{p} \beta_j^2$$

- Lasso:

$$(y - X\beta)^T(y - X\beta) + \lambda \cdot \sum_{j=1}^{p} |\beta_j|$$

- Because of biasedness RR or Lasso can sometimes outperform (e.g. OLS; unbiased optimum) in low dim. in terms of MLE.
- Use CV to find λ
- Lasso shrinks coefs exactly to zero → variable selection
- In high dims: Usually perfect fit → overfitting
  - can’t estimate residual variance
  - traditional tools like p-value, R^2, training MSE become useless
  - use test MSE or cross-validation
- Interpretation: Predictors are collinear
  → predictors found could be replace by others
- Garbage in, garbage out:
  - predictors related to response improve model
  - predictors not related to response deteriorate model

R-Summary – Statistical Modeling (HS2023)

General R intro:
-**assign value** to a variable: `x <- 3`
-define **vector**: `x <- c(3,4,6)`
→vector of **different datatypes**: all values converted into characters (use `list(a,b,c)` to keep datatypes)
-define a vector of **repeated values**: `v <- rep(NA, nreps)`
-**drop** observations with any **NA entry**: `na.omit(df)`

-for 2-dim. **containers** use **matrices:** `matrix(1:12,4,3)` with 4 rows and 3 columns (entries same datatype!)
-**dataframes** (for entries with different datatypes):
`df <- data.frame(col1=c(1,3,5), col2=c(”a”,”b”,”c”),col3=c(TRUE,TRUE,FALSE))`
-**access column** of a data frame using: `df$col_name`
-**count** number of **rows** in dataframe: `n <- nrow(df)`
-**extract columns** from dataframe:
`df2 <- d[,c(“col1”,”col2”,”col3”)]`
-**combine columns into one dataframe**:
`df <- cbind(d$col1,d$col2,d$col3)`
-get **summary statistics** about dataframe : `summary(df)`
-**import** data: `data <- read.csv(“mat.csv”)`
OR `data <- load(“mat.rda”)`
-make sure you are in the **correct working directory**(!)
-**remove all variables** from environment: `rm(list = ls())`

-**access** specific **elements**: `mat[row,col]` **OR** `mat[row,”col_name”]`
-choose all elements from one dim (e.g. **row**): `mat[,col]`
-**plot** x against y: `plot(d$x,d$y)`
-plot **fitted line** of trained model `fm`: `abline(fm)`
-**boxplot**: `boxplot(y ~ x, data = df)`
-**histogram**: `hist(est1, xlim = c(min_val, max_val), main = "title")`

Linear Algebra in R:
-matrix (vector) **product**: `X %*% b`
-**transpose** matrix: `t(X)`
-**inverse** of a matrix: `solve(X)`

Probability in R:
-**mean** value of column: `mean(df$col_name)`
-**standard deviation** of column: `sd(df$col_name)`
-**covariance** of two columns: `cov(df$col1,df$col2)`
-**covariance matrix** of data: `cov(df)`
-(pearson-)**correlation** (2 cols): `cor(x=df$c1,y=df$c2)`

Testing in R:
-one sample **t-Test** for value `mu`: `t.test(d,mu=mu)`
Example output:

```
> t.test(d, mu = 1)

        One Sample t-test

data:  d
t = 2.9062, df = 19, p-value = 0.009054
alternative hypothesis: true mean is not equal to 1
95 percent confidence interval:
 1.017908 1.110092
sample estimates:
mean of x
    1.064
```

Testing Correlation of two cols in R:
-use: `cor.test(x=d$vmax, y=d$vo2max)`
-outputs p value and CI for the true Pearson correlation (of the two r.v., from which we sampled)
Example output:

```
        Pearson's product-moment correlation

data:  d$vmax and d$vo2max
t = 14.347, df = 89, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7604641 0.8885892
sample estimates:
      cor
0.8355503
```

Testing Variance of two samples in R:
`var.test(est1, est2)` (output similar to above)

Linear Regression in R:
-**simple** linear regression of col y against col x from df:
`fm <- lm(y ~ x, data = df)`
-**multiple** linear regression:
`fm <- lm(y ~ col1 + col2 + col3, data=df)`

`summary(fm)`
Example summary output:

```
Residuals:
     Min       1Q   Median       3Q      Max
-10.2230  -4.3976  -0.2016   4.7026  12.0348

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -19.4582     4.7239  -4.119  8.5e-05 ***
vmax          5.8566     0.4082  14.347  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.433 on 89 degrees of freedom
Multiple R-squared:  0.6981,    Adjusted R-squared:  0.6948
F-statistic: 205.8 on 1 and 89 DF,  p-value: < 2.2e-16
```

-**coefficients**: estimated param. vals and their resp. std and p-values (for estimating the true param values)
-**slope interpretation**: y-value changes by "vmax" for every 1-value-increase in the explanatory variable x
- predicted coefficients of lin.model: `coefficients(fm)`
Example coefficients output: `y ~ intercept + vmax * x`

```
> coefficients(fm)
(Intercept)        vmax
 -19.458181    5.856568
```

-**predict** on new data:
`newdf <- data.frame(ProteinG = 10, KohlenhG = 5, FettG = 7)`
`predict(fm, newdata = newdf)`
- predict value of derivative: add attribute `deriv = 1`
- **also output CI**: `predict.lm(fm, newdata = newdf, interval = "confidence", level=0.99)`
- also output **prediction interval**: `predict.lm(fm, newdata = newdf, interval = "prediction", level=0.99)`
-**Covariance matrix** of $\hat{\beta}$ of our model: `vcov(fm)`

Multiple Linear Regression (MLR):
-**residuals** of observations Y_i: `residuals_y <- residuals(lm(y ~ col1 + col1, data=df))`

Reference Levels and Interactions in R:
-**factor variables** (discretely valued: "**levels**"): intrinsic ordering of levels inR .Show reference level: `levels(df$col)`
- **change reference level**: `relevel(df$col, ref="fem")`
-R encodes reference levels using "dummy variables":
Ex.: $Balance_i = (\beta_0 + \beta_2 * x_{1,i} + \beta_3 * x_{2,i}) + \beta_1 * Age_i + \epsilon_i$
- fit same slope to all param, but different intercepts
- β_0 corresponds to intercept of reference level
-**manually factorize** a (discretely valued) col in a dataframe (necessary for linear regression):
`df$col <- factor(x=df$col, levels = c("Male", "Female")) ## first level is ref.level`
-**factorize while reading** file:
`df <- read.csv(file = "data.csv", row.names = 1, header = TRUE, stringsAsFactors = TRUE)`

Example output:(linear regression with factor variable `gender`) **[*output maybe not so important*]**

- note: the slope (of `age`) stays the same over all levels.

```
Call:
lm(formula = Balance ~ Age + Gender, data = dat2)

Residuals:
    Min      1Q  Median      3Q     Max
-530.76 -455.90  -61.05  335.05 1487.22

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  507.21157   81.41578   6.230 1.19e-09 ***
Age            0.04661    1.33738   0.035    0.972
GenderFemale  19.72667   46.10947   0.428    0.669
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 460.8 on 397 degrees of freedom
Multiple R-squared:  0.0004642, Adjusted R-squared:  -0.004571
F-statistic: 0.09218 on 2 and 397 DF,  p-value: 0.912
```

Intercept in reference group (male)

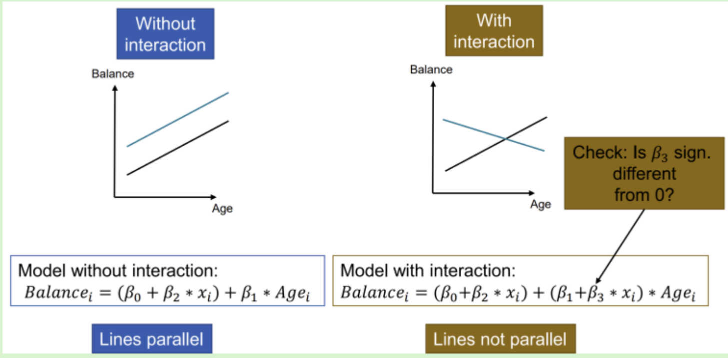
Slope is the same in both groups

Change of intercept, if we switch from reference group (male) to other group (female). Intercept for women:
$507.2 + 19.7 = 526.9$

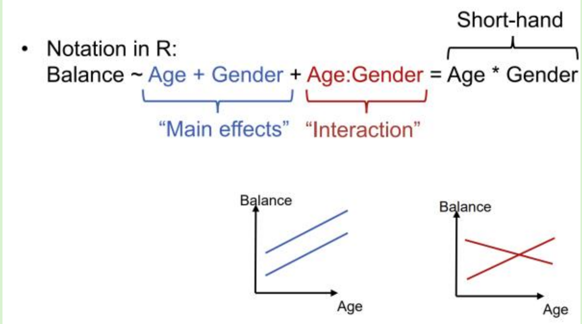
$\rightarrow Balance_i = (507.2 + 19.7 * Gender_i) + 0.047 * Age_i + \varepsilon_i$
$\varepsilon_i \sim N(0, 460.8^2)$

Men: $Balance_i = 507.2 + 0.047 * Age_i + \varepsilon_i,\ \varepsilon_i \sim N(0, 460.8^2)$
Women: $Balance_i = 526.9 + 0.047 * Age_i + \varepsilon_i,\ \varepsilon_i \sim N(0, 460.8^2)$

-**interaction:** if the slope wrt. another param. also changes between levels (/groups). We then say these explanatory variables have interaction.
-only use interaction if really necessary (!)
-**Illustration of interaction** between age and gender:



-note: **intercepts will not stay the same across both models!**



Example output of interactions:

```
Call:
lm(formula = Balance ~ Age * Gender, data = dat2)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       478.6139   113.8643   4.203 3.25e-05 ***
Age                 0.5610     1.9590   0.286    0.775
GenderFemale       73.4491   156.3361   0.470    0.639
Age:GenderFemale   -0.9652     2.6835  -0.360    0.719

Residual standard error: 461.3 on 396 degrees of freedom
Multiple R-squared:  0.0007906, Adjusted R-squared:  -0.006779
F-statistic: 0.1044 on 3 and 396 DF,  p-value: 0.9575
```

Intercept: Men

Slope: Men

Change intercept: Women

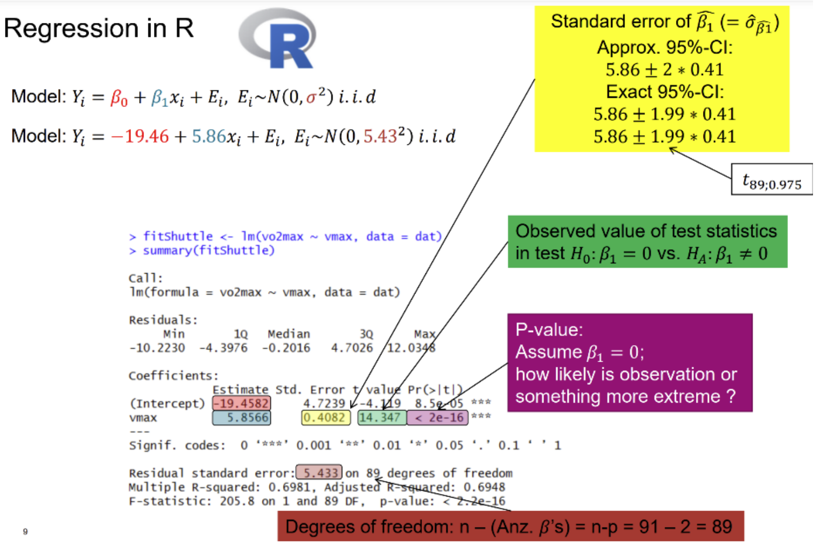
Change slope: Women

$Balance_i = (478.6 + 73.4 * Gender_i) + (0.56 - 0.97 * Gender_i) * Age_i + \varepsilon_i$
$\varepsilon_i \sim N(0, 461.3^2)$

Men: $Balance_i = 478.6 + 0.56 * Age_i + \varepsilon_i,\ \varepsilon_i \sim N(0, 461.3^2)$
Women: $Balance_i = 552.0 - 0.41 * Age_i + \varepsilon_i,\ \varepsilon_i \sim N(0, 461.3^2)$

Understanding the linear model output in R:
-**confidence interval** (CI) of fitted parameters:
`confint(fm, level=0.99)`
- fitted parameters are t-distributed (degree of freedom n-p); using this information we can construct the CI.

Example output: linear model



→ **std.error**: standard deviation of estimated parameter

→ **t-value**: observed value of statistic for test $H_0: \beta_i = 0$.

→ **residual standard error**: error estimate of noise std

→ **F-statistic**: statistic value of test H_0: all param. are 0.

-high test statistic value / low p-value indicates high overall significance of the model

-**Variance inflation factor**: $VIF_i = \frac{1}{(1-(R_i)^2)}$

-VIF_i score for each variable: **high values** indicate that the variable is **highly correlated** with other parameters

-R_i^2 is found through regression of i-th var. against others

- VIF is factor in variance term of $\hat{\beta}_i$

→ **high dependency reduces accuracy**

-**sign of collinearity**: overall high p-values, but low F-Test values!

-In **R** we have: `vif(fm)`

-**Rule of thumb**: VIF < 4: ok; 4 < VIF < 10: borderline; VIF > 10: problematic => "first aid": remove variable

Contrasts in R:

-problem: compute CI of contrasts, e.g. have interaction but want to **find CI of a parameter given a non-reference level**

-find CI **manually**: get CI of parameter given a non-reference level (with corresponding contrast):

```
contrast <- rbind("param_level_slope" =
c(0,0,1,1))
glhtfmI <- glht(fm, linfct = contrast)
summary(glhtfmI)
confint(glhtfmI) ## watch out for multiple
testing correction, if more hypotheses are
included
```

-**find CI by refitting** (faster): change the reference level to the level of interest, then re-run the fitting (might be faster)

Partial F-testing in R:

-first fit full linear model, and then fit partial linear model. Compare models using `anova()`:

```
fm <- lm(y ~ x1 + x2 + x3 + x4 + x5, data=df)
fm2 <- lm(y ~ x1 + x2, data=df)
anova(fm, fm2)
```

Residual Analysis plots in R:

-diagnostic plots like **tukey-anscombe and QQ-plot**:

`plot(fm, which = 1:3)`

-many outliers indicate bad model e.g. should add a variable

Model Selection in R:

-**exhaustive search** (`nvmax`: max number of considered variables): `model1 <- regsubsets(y ~ ., data = dTrain, method = "exhaustive",nvmax = 10)`

-**forward search**: `model2 <- regsubsets(y ~ ., data = dTrain, method = "forward")`

-**backward search**: `model3 <- regsubsets(y ~ ., data = dTrain, method = "backward")`

-**search methods output**: rows which for each number of variables shows us the optimal combination of variables.

-**compare C_p values of all tested models**:

```
ncoef<-which.min(model1_summary$cp)
coef(m1, ncoef)
```

- **alternatively**: `model1_summary<- summary(model1)`
`model1_summary$cp`

→**output**: column of C_p values (choose model with lowest val)

-**graphical view**: `plot(model1, scale = "Cp")`

-compute **MSE of predictions** (manually):

```
fmFull <- lm(y ~ ., data = dTrain)
yHat <- predict(fmFull, newdata = dTest)
mseFull <- mean( (yHat - dTest$y)^2)
```

Non-iid errors in R:

-**WLS (weighted least squares):** Given an observation is the average of n single observations, we scale the new variance to be $\frac{\sigma^2}{n}$: `fm <- lm(y ~ x, data = df, weights = nreps)`

-nreps= #observations which make up a value (average)

-**heteroscedasticity**: **variances in normal summary are wrong!**

→sandwich estimator to adapt error variances

-use the **sandwich estimator** to do linear regression with data that exhibits heteroscedasticity. This adapts the wrong variances for the confidence intervals of a fitted linear model

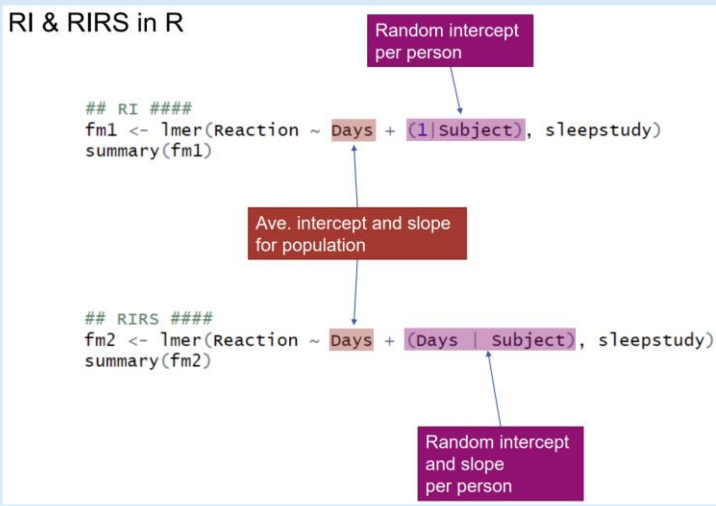
-**new coefficient output** with adapted std:

`coeftest(fm, vcov = vcovHC(fm, type = "HC0"))`

-**new confidence interval**:

`coefci(fm, vcov = vcovHC(fm, type = "HC0"))`

Linear Mixed Models:



Example Output: RI & RIRS

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 612.09   24.740
          Days         35.07    5.922   0.07
 Residual             654.94   25.592
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  251.405      6.825  16.998  36.838  < 2e-16
Days          10.467      1.546  16.995   6.771 3.27e-06
```

$y_{ij} = (251.4 + u_{1,i}) + (10.5 + u_{2,i})x_{ij} + \varepsilon_{ij}$,
$\varepsilon_{ij} \sim N(0, 25.6^2)\ i.i.d$
$u_{1,i} \sim N(0, 24.7^2), u_{2,i} \sim N(0, 5.9^2)$,
$cor(u_1, u_2) = 0.07$

-**output random effects**: describes the estimated standard deviations of each random effect r.v. u_i

-get **confidence intervals of parameters**:

$y_{ij} = (\beta_0 + u_{1,i}) + (\beta_1 + u_{2,i})x_{ij} + \varepsilon_{ij},$
$\varepsilon_{ij} \sim N(0, \sigma^2)\ i.i.d$
$u_{1,i} \sim N(0, \sigma_1^2), u_{2,i} \sim N(0, \sigma_2^2), cor(u_1, u_2) = \rho$

```
> confint(fm2, oldNames = FALSE)
Computing profile confidence intervals ...
                                2.5 %      97.5 %
sd_(Intercept)|Subject      14.3814182  37.7159953   σ1
cor_Days.(Intercept)|Subject -0.4815008  0.6849863   ρ
sd_Days|Subject              3.8011641   8.7533808   σ2
sigma                       22.8982669  28.8579965   σ
(Intercept)                237.6806955 265.1295147   β0, β1
Days                         7.3586533  13.5759188
```

Example output association: of CI with summary

```
> confint(fm2, oldNames = FALSE)
Computing profile confidence intervals ...
                                2.5 %      97.5 %
sd_(Intercept)|Subject      14.3814182  37.7159953
cor_Days.(Intercept)|Subject -0.4815008  0.6849863
sd_Days|Subject              3.8011641   8.7533808
sigma                       22.8982669  28.8579965
(Intercept)                237.6806955 265.1295147
Days                         7.3586533  13.5759188
```

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 612.09   24.740
          Days         35.07    5.922   0.07
 Residual             654.94   25.592
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  251.405      6.825  16.998  36.838  < 2e-16
Days          10.467      1.546  16.995   6.771 3.27e-06
```

-estimates of the **random effects of each individual**:
`ranef(fm)`

GLM (general linear model) Logistic Regression in R:

-GLM logistic regression: model the probability of a binary random variable

-`glm()` **models probability for the non-reference level!!**

-note: a change of 1 unit in any variable has **multiplicative or additive change w.r.t. the scale**

$\pi = P(default = \text{Yes})$
Model:
S: $Y_i \sim Bin(1, \pi_i)$
D: $\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} = x_i^T\beta$ (scale: log-odds → additive change)
or: $\frac{\pi_i}{1-\pi_i} = \exp(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}) = x_i^T\beta$ (scale: odds → multiplicative change)

```
fm1 <- glm(default ~ balance, data = Default, family = binomial(link = "logit"))
## link = "logit" is default, so easier to write:
fm1 <- glm(default ~ balance, data = Default, family = binomial())
```

-Note: $\boldsymbol{\beta}$ **param. are modeled w.r.t. log-odds scale here!!**
- thus std, and corresponding CI are on log-odds scale
- apply `exp()` to **extract odds-ratio and CI of odds-ratio**!
  → `exp(coef(fm))` **OR** `exp(confint_fm)`

-compute **(log-)odds** and **probabilities** from the **GLM model**:

```
## Prediction from model ####
dNew <- data.frame(balance = 730)
lo <- predict.glm(fm1, newdata = dNew, type = "link", se.fit = TRUE) ## log-odds
lo
p <- predict.glm(fm1, newdata = dNew, type = "response", se.fit = TRUE) ## proba
p
## no option in predict.glm() for predicting the odds => manually
p$fit/(1-p$fit) ## odds computed manually

## check: log(p/(1-p)) same as log-odds (lo)
log(p$fit/(1-p$fit))
```

GLM examples:

-**Poisson regression** (e.g. for counting incidents): Assume $Y_i \sim Pois(\lambda_i) \rightarrow \mu_i = E(Y_i) = Var(Y_i) = \lambda_i$. **Problem**: if we **expect larger values**, their **variance** will **grow** aswell!

D: $\log(\mu_i) = \log(\lambda_i) = x_i^T\beta$; S: $Y_i \sim Pois(\lambda_i)$

-in **R**: we use `glm` with: `family=poisson()` or `family=quasipoisson()`

-Gamma Regression:

Recap: Gamma distribution $\Gamma(k, \theta)$ where k: "shape", θ: "scale"
$E(X) = k\theta,\ Var(X) = k\theta^2$

Gamma regression:
D: $\log(\mu_i) = x_i^T\beta$ (alternative: $\frac{1}{\mu_i} = x_i^T\beta$ but harder to interpret)
S: $Y_i \sim \Gamma(\mu_i, \nu)$
In R: `glm(formula = y ~ x, family = Gamma(link = "log"), data = df)`

Comparing Models:

-**Residual Deviance**: receive a squared deviance residual d_i for every parameter; compute **squared sum deviance**:
`sum(residuals(fm, type = "deviance")^2)`

Nonlinear regression in R:

-run nonlinear regression with **manually chosen starting values**:

```
fm <- nls(yObs ~ t1*exp(-t2*t3^x), data = df,
          start = c(t1= 12, t2= 5, t3= 0.5))
```

-for some nonlinear functions there are **optimized self-starting functions** like SSgompertz:

```
fm <- nls(yObs ~ SSgompertz(x, Asym, b2, b3),
data = df)
```

-summary output of `nls()` std values can be used to construct **Wald-Type CI**
- But: **assumes symmetry** in distr. around estimate (MLE)
- **instead**: compute **CI based on profile likelihood** (i.e. likelihood drop). This is automatically done by: `confint(fm)`

Nonparametric Regression in R:

-estimate function with **no knowledge about functional form**

-estimate using **polynomial**:
`fm_p10 <- lm(y ~ poly(x, degree = 10))`

-**kernel regression:** use a weighted window for a moving "average". The weights are given by the kernel function:

```
y_ks <- ksmooth(x,y,kernel = "normal",
bandwidth = 0.01, x.points = x)$y
```

- with x.points we request predictions on other points

-**local polynomial regression**. Fit a polynomial inside the window, but only keep the intercept:

```
fm_lp <- locpoly(x,y,drv = 0, degree = 1,
kernel = "normal",
          bandwidth = 0.005)
```

-`drv=0`: set derivative to 0; `degree=1`: degree of polynomials (usually not higher than 2)

-**smoothing spline** (works best, and fast):
`fm_ss <- smooth.spline(x,y, cv = TRUE)`