Exam FAC SIMILE

Matteo Suardi

2025-05-13

Exam 1

Exercise 1

The following variables, which can be found in `cars.Rdata`, were collected for a group of 50 cars in the 1920:

- measurement of speed (`speed`, in kilometers per hour)
- distance taken to stop (`dist`, in meters)

1.1 Illustrate the data using appropriate summary statistics. Draw the bivariate scatter plot and comment on it. Are the variables related to each other? Calculate and comment on the value of the sample linear correlation coefficient. After setting the working directory to source file location, we load the data and illustrate the summary statistics with the `skim_without_charts()` (from the `skimr` library) function.

```
load("cars.Rdata")
require(skimr)
```

```
## Loading required package: skimr
```

```
skim_without_charts(cars)
```

Table 1: Data summary

Name	cars
Number of rows	50
Number of columns	2
Column type frequency:	
numeric	2
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
speed	0	1	24.78	8.51	6.44	19.31	24.13	30.57	40.23
dist	0	1	13.10	7.85	0.61	7.92	10.97	17.07	36.57

The `cars` dataset contains data from an observational study conducted on vehicles from the 1920s, aiming to explore how a vehicle's speed (`speed`, in km/h) influences the distance required to stop (`dist`, in meters).

The variable `speed` shows a **mean of 24.78 km/h** and a **standard deviation of 8.51**, with minimum and maximum values ranging from **6.44 to 40.23 km/h**. The distribution appears **approximately symmetric**, as the mean and the median (24.14) are nearly coincident, and the interquartile range is balanced. The coefficient of variation is about 0.34, indicating **moderate relative variability**. The symmetry and dispersion suggest the variable may reasonably follow a **Gaussian distribution**.

The variable `dist` exhibits a **mean of 13.10 meters** and a **standard deviation of 7.85**, with values ranging from **0.61 to 36.57 meters**. The **median is lower than the mean** (10.97), and the **second quartile is closer to the minimum than the maximum**, indicating a **positively skewed distribution**. The coefficient of variation is approximately 0.60, suggesting **high relative variability** in braking performance among the vehicles. This may reflect differences in braking systems or weight distribution, despite similar engine characteristics across the sample.

```r
cv <- function(x) sd(x) / mean(x)
round(apply(cars, 2, cv), 2)
```

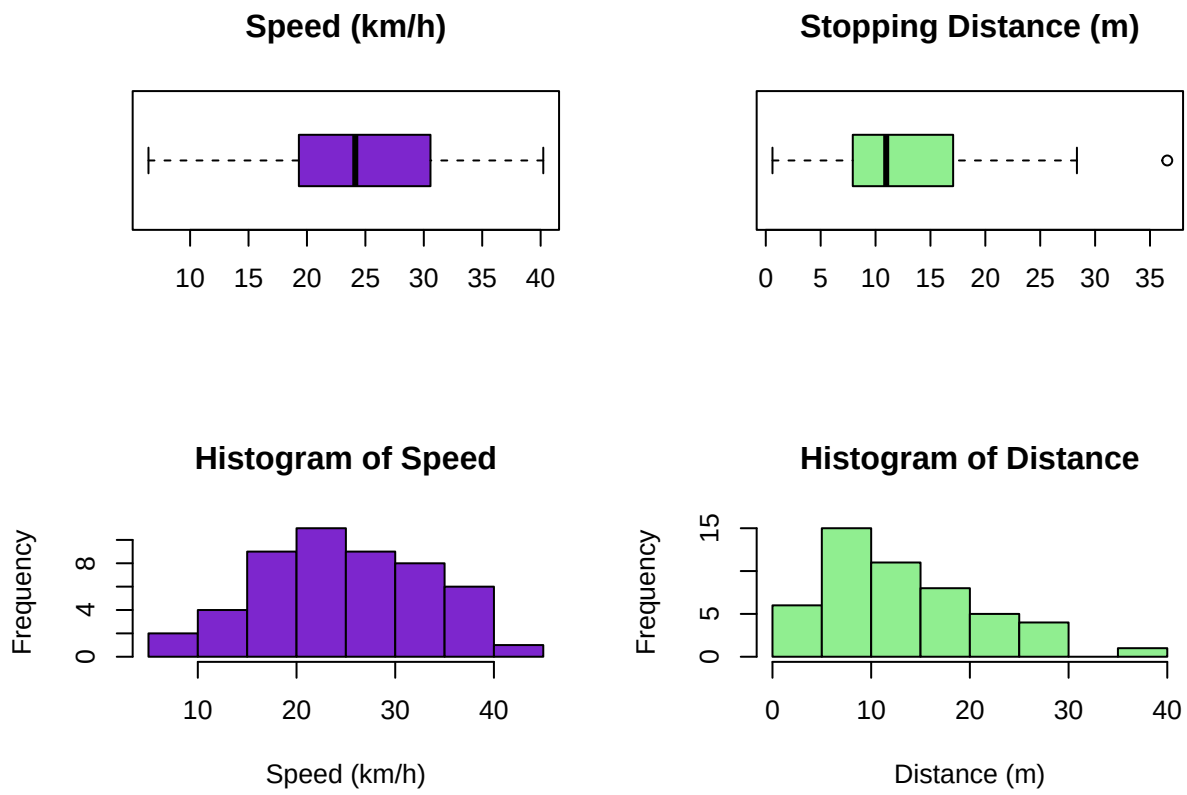
```
## speed  dist
##  0.34  0.60
```

The variable `dist` shows more variability, with a coefficient of variation approximately equal to 0.60. The distribution is sightly asymmetrical, as the mean and the median are not equal and the second quartile is closer to the minimum than to the maximum. This suggests that, while the engine power is almost the same for all the cars, some of them had better braking systems than others.

We now visualize the marginal distributions of the two variables through boxplots and histograms, to better understand their shape, variability, and potential presence of outliers.

```r
par(mfrow = c(2, 2))

# Boxplots
boxplot(cars$speed,
        horizontal = TRUE,
        col = "purple3",
        main = "Speed (km/h)")
boxplot(cars$dist,
        horizontal = TRUE,
        col = "lightgreen",
        main = "Stopping Distance (m)")

# Histograms
hist(cars$speed,
     col = "purple3",
     main = "Histogram of Speed",
     xlab = "Speed (km/h)")
hist(cars$dist,
     col = "lightgreen",
```

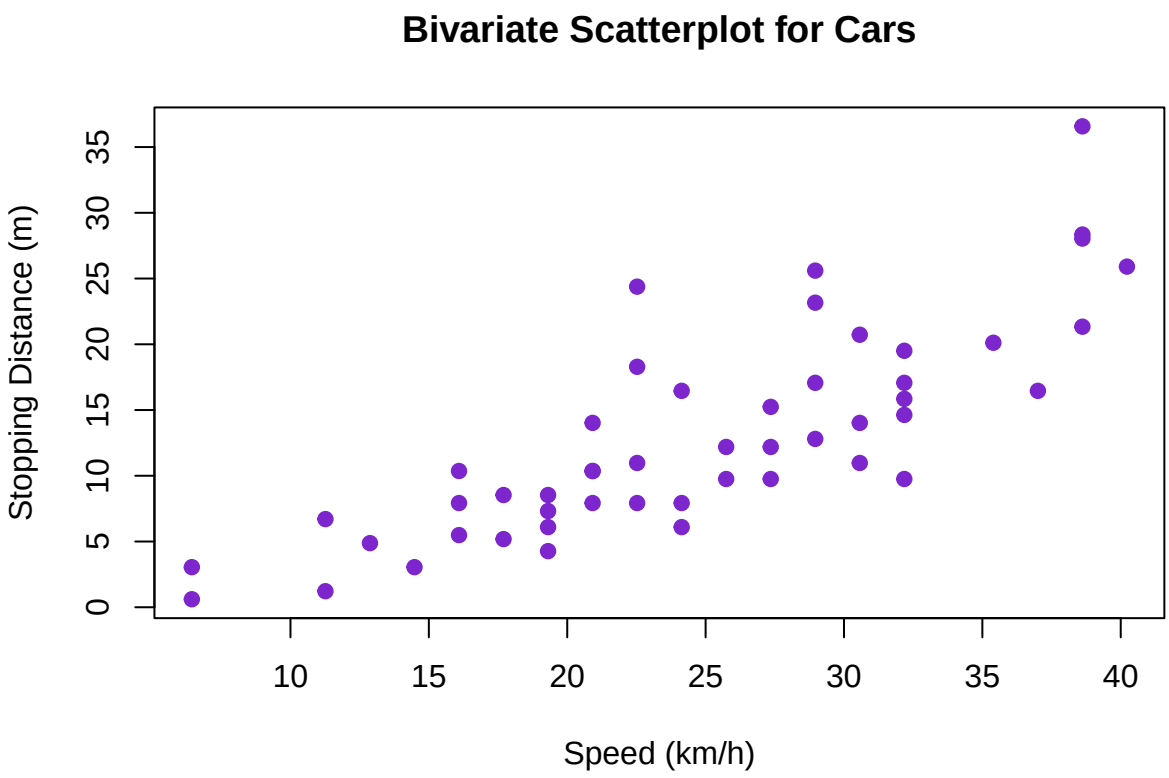
```
     main = "Histogram of Distance",
     xlab = "Distance (m)")
```



- The `speed` distribution is roughly symmetric and bell-shaped, consistent with a Gaussian pattern. No outliers are present, and the values are concentrated around the mean.
- The `dist` variable is more skewed to the right, with an upper outlier (as also seen in the boxplot). This suggests the presence of a few vehicles requiring considerably longer braking distances, possibly due to inferior braking systems or heavier structure.

We now examine the relationship between the two variables using a bivariate scatterplot.

```
plot(cars,
     main = "Bivariate Scatterplot for Cars",
     pch = 19,
     xlab = "Speed (km/h)",
     ylab = "Stopping Distance (m)",
     col = "purple3")
```



The scatterplot reveals a clear **positive linear relationship**: as the car's speed increases, the stopping distance tends to increase as well. The cloud of points forms a clear ascending trend, with no relevant outliers deviating from the overall structure.

To quantify this association, we compute the **Pearson correlation coefficient**:

```
round(cor(cars), 2)
```

```
##       speed dist
## speed  1.00 0.81
## dist   0.81 1.00
```

The correlation coefficient $\hat{\rho} = 0.81$ confirms a **strong positive linear association** between the two variables. This indicates that about 81% of the variation in one variable moves consistently with the other. In terms of interpretation:

- An increase in speed is typically associated with an increase in the stopping distance.
- The squared correlation $R^2 = 0.81^2 \approx 0.66$ suggests that **66% of the variability** in stopping distance can be **linearly explained** by the car's speed.

The visual and numerical summaries confirm that:

- `speed` and `dist` are positively correlated;
- the relationship is strong and approximately linear;
- the dataset is appropriate for fitting a simple linear regression model.

1.2 Considering the correlation coefficient as a parameter of interest, use the nonparametric bootstrap to provide an accuracy measure for it. Obtain the bootstrap distribution from B = 1000 resamples and comment on the bootstrap estimate of the corresponding standard error. In this exercise, we aim to estimate the **sampling variability** of the Pearson correlation coefficient $\hat{\rho}$ between `speed` and `dist`, using the **nonparametric bootstrap**. This resampling-based method does not rely on distributional assumptions (such as normality), making it particularly appropriate in our context, where the `dist` variable shows moderate skewness and outliers.

The bootstrap proceeds by resampling with replacement from the original data and computing the statistic of interest for each bootstrap sample. We define the correlation function and perform **B = 1000** resamples.

```
library(boot)

# Statistic: correlation between speed and distance
cor_fn <- function(data, indices) {
  d <- data[indices, ]
  return(cor(d$speed, d$dist))
}

set.seed(123)

bootC <- boot(data = cars,
              statistic = cor_fn,
              R = 1000)

round(sd(bootC$t), 3)
```

```
## [1] 0.049
```

Alternative implementation (method 2: using `bootstrap` package)

```
library(bootstrap)

set.seed(123)

n <- nrow(cars)

boot_fun1 <- function(ind) {
  cor(cars$speed[ind], cars$dist[ind])
}

bootC1 <- bootstrap(1:n, 1000, boot_fun1)

round(sd(bootC1$thetastar), 3)
```

```
## [1] 0.049
```

As expected, both implementations return the **same standard error estimate**. The resulting bootstrap standard error is 0.049.

The **bootstrap standard error of the correlation coefficient** is approximately 0.049, which is **small** compared to the magnitude of the estimate itself ($\hat{\rho} = 0.81$). This indicates that:

- The estimate is **stable and reliable**, with limited variation across repeated samples;

- The correlation between speed and stopping distance is **statistically robust**, even under possible non-normality;
- The use of the nonparametric bootstrap has effectively captured the variability **without relying on theoretical approximations**.

This confirms that the **observed strong association** in Exercise 1.1 is not driven by noise or sample-specific peculiarities.

1.3 Provide the bootstrap percentile confidence interval for the parameter computed at 99% confidence level. Comment on its value and length. Depict the bootstrap distribution for the parameter of interest and add the lower and upper bound of the confidence interval. Comment on the shape of the distribution. In this exercise, we compute a **99% bootstrap percentile confidence interval** for the Pearson correlation coefficient $\hat{\rho}$ between the variables `speed` and `dist`, based on the distribution of 1000 bootstrap replicates obtained in Exercise 1.2.

The percentile method is nonparametric and uses the empirical distribution of the bootstrap replications to define the bounds of the confidence interval. Specifically, the **0.5% and 99.5% quantiles** are used as the lower and upper limits, respectively.

```
ci_99 <- quantile(bootC$t, probs = c(0.005, 0.995))
round(ci_99, 3)
```

```
##  0.5% 99.5%
## 0.644 0.905
```

Therefore, the **99% bootstrap percentile confidence interval** for the correlation coefficient is:

$$[0.644,\ 0.905]$$

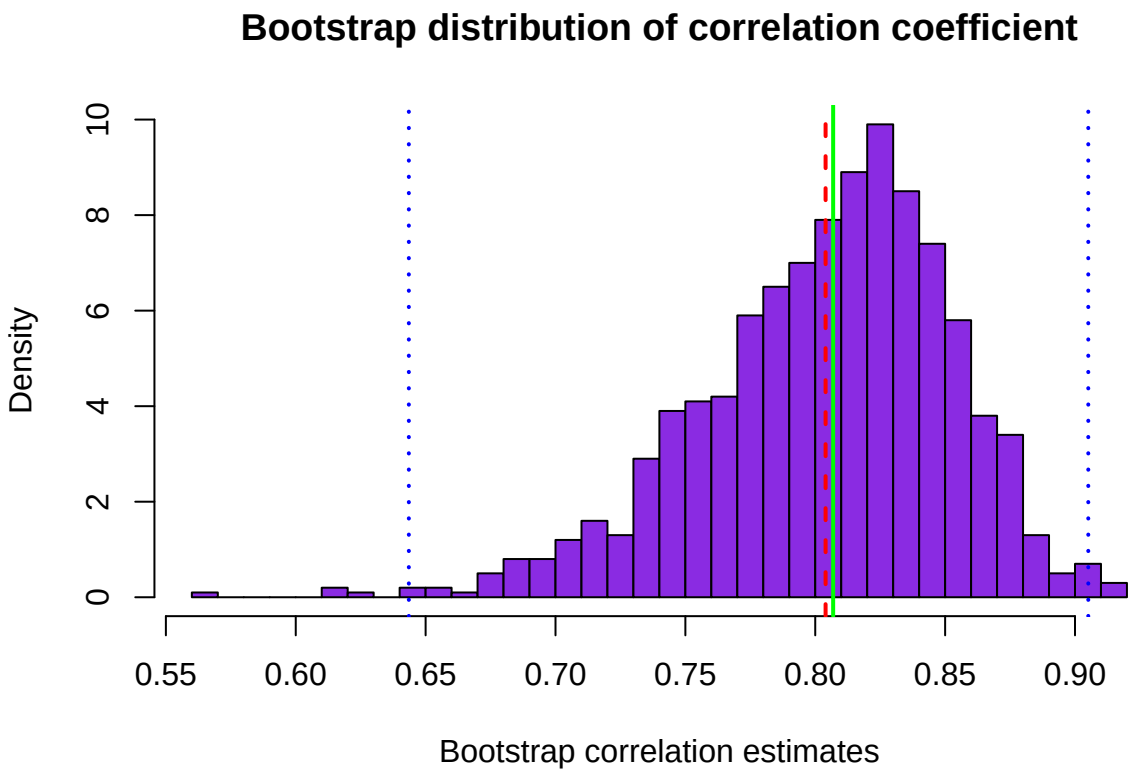
Its **length** is computed as:

```
round(diff(ci_99), 3)
```

```
## 99.5%
## 0.262
```

To visualize the bootstrap distribution with the confidence intervals:

```
hist(bootC$t,
     breaks = 30,
     probability = TRUE,
     col = "blueviolet",
     main = "Bootstrap distribution of correlation coefficient",
     xlab = "Bootstrap correlation estimates")

abline(v = cor(cars$speed, cars$dist), col = "green", lwd = 2)
abline(v = mean(bootC$t), col = "red", lwd = 2, lty = 2)
abline(v = ci_99, col = "blue", lwd = 2, lty = 3)
```



- The **bootstrap percentile confidence interval [0.644, 0.905]** confirms that the true correlation is **significantly greater than zero**, supporting the **strong positive association** observed in Exercise 1.1.
- The interval's length of 0.261 is moderate, reflecting **good precision** in the estimation of the correlation coefficient.
- The bootstrap distribution of the estimates appears **unimodal** and **slightly skewed to the left**. The sample estimate ($\hat{\rho} \approx 0.81$) and the bootstrap mean are close, suggesting **low bias**.
- The mild asymmetry in the distribution highlights the **usefulness of the bootstrap**: unlike standard theory-based methods (which assume normality and symmetry), the percentile approach adapts to the **true empirical shape** of the distribution.

1.4 What does it mean when we refer to homoscedasticity in the context of a multiple linear regression model? How do we check this assumption empirically? In the context of a multiple linear regression model, the assumption of **homoscedasticity** refers to the condition that the **variance of the error terms** is constant across all observations. Mathematically, if the model is specified as:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \varepsilon_i,$$

then the homoscedasticity assumption requires that:

$$\text{Var}(\varepsilon_i) = \sigma^2 \quad \text{for all } i,$$

meaning that the random errors ε_i have a constant variance σ^2 regardless of the values of the explanatory variables. This assumption is essential for the **efficiency** of the ordinary least squares (OLS) estimators.

When it holds, the OLS estimators are the **best linear unbiased estimators (BLUE)**, as stated by the Gauss-Markov theorem. In contrast, when this condition is violated—known as **heteroscedasticity**—the standard errors of the estimated coefficients may be biased, which undermines the validity of confidence intervals and hypothesis tests.

Empirically, homoscedasticity is usually assessed by analyzing the **residuals** of the fitted model. The most common graphical tool is the **residuals vs. fitted values plot**. Under homoscedasticity, the residuals should appear randomly scattered around zero with no apparent pattern and with a roughly constant spread. However, the presence of a **funnel shape** (i.e., increasing or decreasing spread) in this plot suggests heteroscedasticity. In addition to graphical inspection, formal statistical tests can be used—such as the **Breusch-Pagan test**—which tests the null hypothesis that the variance of the residuals is constant:

$$H_0 : \mathrm{Var}(\varepsilon_i) = \sigma^2 \quad \text{vs.} \quad H_1 : \mathrm{Var}(\varepsilon_i) = \sigma^2 h(Z_i),$$

where Z_i typically includes the fitted values or a subset of the regressors. A low p-value indicates rejection of the null hypothesis and thus evidence of heteroscedasticity.

In conclusion, homoscedasticity ensures the reliability of inferential procedures in linear regression. Verifying this assumption through residual plots and formal tests is a necessary step in validating the model.

Exercise 2

Data concerning the birth weight of a sample of infants can be found in `bwt.Rdata`. The interest lies in explaining factors that influence the birth weight lower than 2.5 kg. Key variables include:

- `low`: 1 = if the birth weight is lower than 2.5 kg, 0 otherwise
- `age`: mother's age in years
- `lwt`: mother's weight in kilograms at last menstrual period
- `race`: mother's race (0 = White, 1 = Black, 2 = Others)
- `smoke`: smoking status during pregnancy (0 = No, 1 = Yes)
- `ht`: history of hypertension (0 = No, 1 = Yes)
- `ftv`: number of physician visits during the first trimester (0, 1, or more than 1)

2.1 Define the sample size. Provide the relative frequencies of the contingency table referred to `race` and `smoke` with the response variable. Comment on the values. In this exercise, we aim to describe the association between the binary outcome variable `low` (low birth weight: 1 = < 2.5 kg, 0 = otherwise) and two categorical predictors: **race** and **smoking status** (`smoke`). We begin by loading the data and computing the sample size.

```
load("bwt.Rdata")
dim(bwt)
```

```
## [1] 189   7
```

The dataset contains information (7 features) on 189 newborns.

Birth weight by race

We compute the contingency table between maternal race and birth weight status, followed by row-wise relative frequencies.

```r
table_race <- table(Race = bwt$race, LowBirthWeight = bwt$low)
round(prop.table(table_race, margin = 1), 3)
```

```
##        LowBirthWeight
## Race        0     1
##   white 0.760 0.240
##   black 0.577 0.423
##   other 0.627 0.373
```

These relative frequencies show that:

- Among **White mothers**, 24.0% of infants had low birth weight;
- Among **Black mothers**, the proportion increases to 42.3%;
- For other races, the proportion is 37.3%.

This suggests that **Black mothers** had the **highest risk** of delivering underweight babies, while **White mothers** had the **lowest observed rate**.

Birth weight by smoking status

We now analyze the association between smoking during pregnancy and low birth weight.

```r
table_smoke <- table(Smoke = bwt$smoke, LowBirthWeight = bwt$low)
round(prop.table(table_smoke, margin = 1), 2)
```

```
##      LowBirthWeight
## Smoke    0    1
##     0 0.75 0.25
##     1 0.59 0.41
```

These results indicate that:

- Among **non-smokers**, 25% of infants were underweight;
- Among **smokers**, the percentage rises to 41%.

Smoking is thus associated with a notably higher risk of low birth weight.

Both contingency tables reveal important patterns:

- The incidence of low birth weight is substantially higher among Black mothers compared to White mothers.
- Smoking during pregnancy is also strongly associated with increased risk: 41% of smokers had underweight babies compared to 25% of non-smokers.

These findings suggest that **race** and **smoking** are potential risk factors for low birth weight and should be considered in the modeling phase. A logistic regression model can help quantify these associations while adjusting for other covariates such as age, weight, hypertension, and prenatal care.

2.2 Fit a logistic regression model using all the available covariates and perform model selection. Comment on the results of this procedure. We begin by fitting a logistic regression model using all the available covariates to predict the binary outcome `low` (1 = birth weight < 2.5 kg, 0 otherwise) using the `gml()` function. The model outcome is summarized using `summary()`.

```r
model2 <- glm(low ~., family = binomial(), data = bwt)

summary(model2)
```

```
##
## Call:
## glm(formula = low ~ ., family = binomial(), data = bwt)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.86858    1.16749   0.744   0.4569
## age         -0.01905    0.03591  -0.530   0.5958
## lwt         -0.03932    0.01527  -2.575   0.0100 *
## raceblack    1.23723    0.52886   2.339   0.0193 *
## raceother    0.88030    0.43758   2.012   0.0442 *
## smoke1       1.01993    0.39939   2.554   0.0107 *
## ht1          1.77829    0.70088   2.537   0.0112 *
## ftv1        -0.26037    0.44893  -0.580   0.5619
## ftv2+        0.08646    0.44267   0.195   0.8451
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 207.38  on 180  degrees of freedom
## AIC: 225.38
##
## Number of Fisher Scoring iterations: 4
```

From the model output, we observe that the significant predictors (all at the 1% level) include:

- `lwt`: mother's weight at last menstrual period (negative effect of ≈ -0.04)
- `raceblack` and `raceother`: both categories increase the risk of low birth weight compared to white mothers (baseline)
- `smoke1`: smoking is associated with higher risk ≈ 1.02
- `ht1`: history of hypertension significantly increases the risk ≈ 1.78

The covariates `age`, `ftv1` and `ftv2+` are not statistically significant, as their p-values exceed conventional thresholds.

Interpretation of model statistics

- The **null deviance** (234.67) measures the deviance of a model with only the intercept (in this case, $\beta_0 \approx 0.87$). It provides a baseline.
- The **residual deviance** (207.38) reflects the deviance of the full model including all the predictors. A lower deviance indicates better fit.
- The **AIC (Akaike Information Criterion)** is a penalized measure of model fit; here it is 225.38 for the full model. A lower AIC suggests a better trade-off between goodness-of-fit and model complexity.

Thus, these values are used to compare models, not interpreted isolation. In particular, AIC is the criterion used in the stepwise model selection.

In fact, to identify a more parsimonious model, we apply **stepwise backward elimination** based on minimizing the AIC. The procedure is conducted by the `step()` function.

```
step(model2)
```

```
## Start:  AIC=225.38
## low ~ age + lwt + race + smoke + ht + ftv
##
##         Df Deviance    AIC
## - ftv    2   207.88 221.88
## - age    1   207.66 223.66
## <none>       207.38 225.38
## - race   2   214.66 228.66
## - smoke  1   214.16 230.16
## - ht     1   214.21 230.21
## - lwt    1   215.02 231.02
##
## Step:  AIC=221.88
## low ~ age + lwt + race + smoke + ht
##
##         Df Deviance    AIC
## - age    1   208.25 220.25
## <none>       207.88 221.88
## - race   2   215.68 225.68
## - ht     1   214.58 226.58
## - lwt    1   215.19 227.19
## - smoke  1   215.75 227.75
##
## Step:  AIC=220.25
## low ~ lwt + race + smoke + ht
##
##         Df Deviance    AIC
## <none>       208.25 220.25
## - race   2   216.86 224.86
## - ht     1   215.01 225.01
## - smoke  1   216.29 226.29
## - lwt    1   216.35 226.35

##
## Call:  glm(formula = low ~ lwt + race + smoke + ht, family = binomial(),
##     data = bwt)
##
## Coefficients:
## (Intercept)          lwt    raceblack    raceother       smoke1          ht1
##     0.35205     -0.03948      1.28766      0.94364      1.07157      1.74916
##
## Degrees of Freedom: 188 Total (i.e. Null);  183 Residual
## Null Deviance:       234.7
## Residual Deviance: 208.2     AIC: 220.2
```

The selection procedure removes variable that do not improve the model according to AIC. Here's the path:

1. `ftv` is removed first (reducing AIC from 225.38 to 221.88)

2. Then `age` is removed (AIC from 221.88 to 220.25).

The final selected model is:

$$\text{logit}(\text{P}(\text{low} = 1)) = 0.352 + -0.039 \cdot X_1 + 1.288 \cdot X_2 + 0.944 \cdot X_3 + 1.071 \cdot X_4 + 1.749 \cdot X_5$$

Alternative formula:

$$\Pr(\text{low} = 1) = \frac{\exp\left(0.352 - 0.039X_1 + 1.288X_2 + 0.944X_3 + 1.071X_4 + 1.749X_5\right)}{1 + \exp\left(0.352 - 0.039X_1 + 1.288X_2 + 0.944X_3 + 1.071X_4 + 1.749X_5\right)}$$

Where:

- $X_1 =$ `lwt`
- $X_2 =$ `raceblack`
- $X_3 =$ `raceother`
- $X_4 =$ `smoke1`
- $X_5 =$ `ht1`

2.3 Estimate the model selected at the previous point. Report and comment carefully on the estimated regression coefficients. Which is the estimated probability of the first six sample units? Comment on this values. We estimate the logistic regression model identified in the previous exercise. The final model includes the covariates: maternal weight (`lwt`), race (`race`), smoking status (`smoke`), and history of hypertension (`ht`). The response variable low is binary and indicates whether the newborn has a birth weight below 2.5 kg.

```
model2_selected <- glm(low ~ lwt + race + smoke + ht, family = "binomial", data = bwt)

summary(model2_selected)
```

```
##
## Call:
## glm(formula = low ~ lwt + race + smoke + ht, family = "binomial",
##     data = bwt)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.35205    0.92444   0.381  0.70333
## lwt         -0.03948    0.01499  -2.634  0.00844 **
## raceblack    1.28766    0.52165   2.468  0.01357 *
## raceother    0.94364    0.42338   2.229  0.02583 *
## smoke1       1.07157    0.38752   2.765  0.00569 **
## ht1          1.74916    0.69082   2.532  0.01134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 208.25  on 183  degrees of freedom
## AIC: 220.25
##
## Number of Fisher Scoring iterations: 4
```

The estimated coefficients confirm the results previously discussed: all included covariates are statistically significant at the 5% level. In particular, `lwt` and `smoke1` are significant at the 1% level, indicating strong evidence against the null hypothesis for these effects, while the other predictors (`raceblack`, `raceother` and `ht1`) are significant at the 5% level.

From the estimated model:

- Maternal weight has a **protective effect**: higher weight is associated with a lower probability of low birth weight.
- Non-white race, smoking during pregnancy and a history of hypertension are associated with higher risk.

```
head_data <- bwt[1:6, ]

head_data$Estimated_prob <- round(model2_selected$fitted.values[1:6], 3)

head_data
```

```
##   low age      lwt  race smoke ht ftv Estimated_prob
## 1   0  19 82.53968 black     0  0   0          0.165
## 2   0  33 70.29478 other     0  0  2+          0.185
## 3   0  20 47.61905 white     1  0   1          0.388
## 4   0  21 48.97959 white     1  0  2+          0.375
## 5   0  18 48.52608 white     1  0   0          0.379
## 6   0  21 56.23583 other     0  0   0          0.284
```

These predicted probabilities illustrate how the model combines the effects of different covariate to estimate the individual risk of low birth weight:

- Observations 35 refer to white, smoking mothers with low maternal weight ($\approx$ 48 kg). Their predicted probabilities of low birth weight range from 0.375 to 0.388, which are the highest values among these six observations. This aligns with the model's results, where both smoking and lower weight are associated with increased risk.
- Observation 1 corresponds to a black mother, non-smoker, with high weight (82.54 kg). Despite the protective effect of the weight, the model assigns her a non-negligible risk of 16.5%, due to the positive coefficient associated with the `raceblack` variable.
- Observation 2, an other-race non-smoking mother with a weight of 70.29 kg, shows an estimated probability of 0.185, slightly higher than the black mother due to lower weight and no additional risk factors.
- Observation 6 is another other-race non-smoking mother with a weight of 56.24 kg, and her estimated probability is 28.4%, reflecting the increased risk due to lower weight compared to Observation 2.

2.4 Explain what is meant when we refer to concentration ellipses in the context of the bivariate Gauss distribution. If the ellipses are oriented to the left, what does it mean? In the context of the bivariate Gaussian (normal) distribution, **concentration ellipses** are **level curves of equal probability density**. They represent the set of points (x_1, x_2) that satisfy:

$$(x - \mu)^T \Sigma^{-1} (x - \mu) = c,$$

where μ is the mean vector, Σ is the covariance matrix, and $c > 0$ determines the size of the ellipse (e.g., 68%, 95%, 99% confidence regions for increasing c).

These ellipses describe how the joint probability mass is distributed around the mean in the two-dimensional space. Their **shape** and **orientation** are determined by the **covariance matrix**:

- The **lengths of the axes** of the ellipse depend on the **variances** of each variable and the **correlation** between them.
- The **orientation** of the ellipse reflects the **direction of dependence** (or correlation) between the two variables.

If the ellipses are **oriented to the left** (i.e., they tilt from the top-left to the bottom-right), this means that the **covariance is negative** and the **correlation coefficient** $\rho < 0$. In other words:

> When one variable increases, the other tends to decrease.

Thus, the orientation of the ellipses provides a visual representation of the **sign and strength of the linear relationship** between the two variables in a bivariate normal distribution.

Exam 2

Exercise 3

Data available in `water.Rdata` report concentrations of organic substances in sampled waters. The following records are available: - `Organic`: concentration of organic substances in water (expressed in mg/L); - `Material`: type of material of the tank storing the water: 0 if plastic, 1 if steel; - `Flow`: water flow rate (in liters per minute); - `Temperature`: external temperature (in degrees Celsius).

3.1 Report the sample size and specify if data are derived from an observational study. Summarize the data by the categorical variable and comment on the reported results. We begin by loading the dataset and obtaining summary information through the `skim_without_chart()` function.

```
load("water.Rdata")
skim_without_charts(water)
```

Table 3: Data summary

Name	water
Number of rows	388
Number of columns	4
Column type frequency:	
factor	1
numeric	3
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Material	0	1	FALSE	2	0: 323, 1: 65

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Organic	0	1	5.89	1.15	3.32	5.04	5.78	6.60	10.38
Flow	0	1	16.33	19.10	0.00	1.82	7.98	23.80	96.03
Temperature	0	1	14.08	6.26	0.46	10.03	13.33	17.51	36.17

The dataset contains 388 observations and 4 variables. Three of them (`Organic`, `Flow` and `Temperature`) are continuous variables, while the `Material` variable is a categorical one.

Since the dataset includes existing measurements on water samples without anu controlled intervention or treatment assignment, we can conclude that the data come from an observational study, not a randomized experiment.

We now examine the categorical variable `Material`, which identifies the type of storage tank. The `skim` output reports:

- `n_unique = 2`: the variable has two categories (plastic and steel)
- `top_counts = 0:323, 1:65` most tanks are made of **plastic** (323), while only 65 are made of **steel**.

This means that approximately 83% of the observations involve plastic tanks, while 17% involve steel tanks. The distribution is therefore **strongly unbalanced**, and this should be kept in mind when comparing the two groups, especially in inferential analyses (e.g. regression models), as the variability in the smaller group may affect the precision of estimates.

3.2 Estimate a multiple linear regression model to explain the concentration of organic substances (Organic) as a function of all the other explanatory variables. Print the results using the summary function and comment on the residuals, the estimated regression coefficients and their standard errors and significance. We estimate a multiple linear regression model to explain the concentration of organic substances (`Organic`) as a function of `Flow`, `Temperature` and `Material` using the `lm()` function.

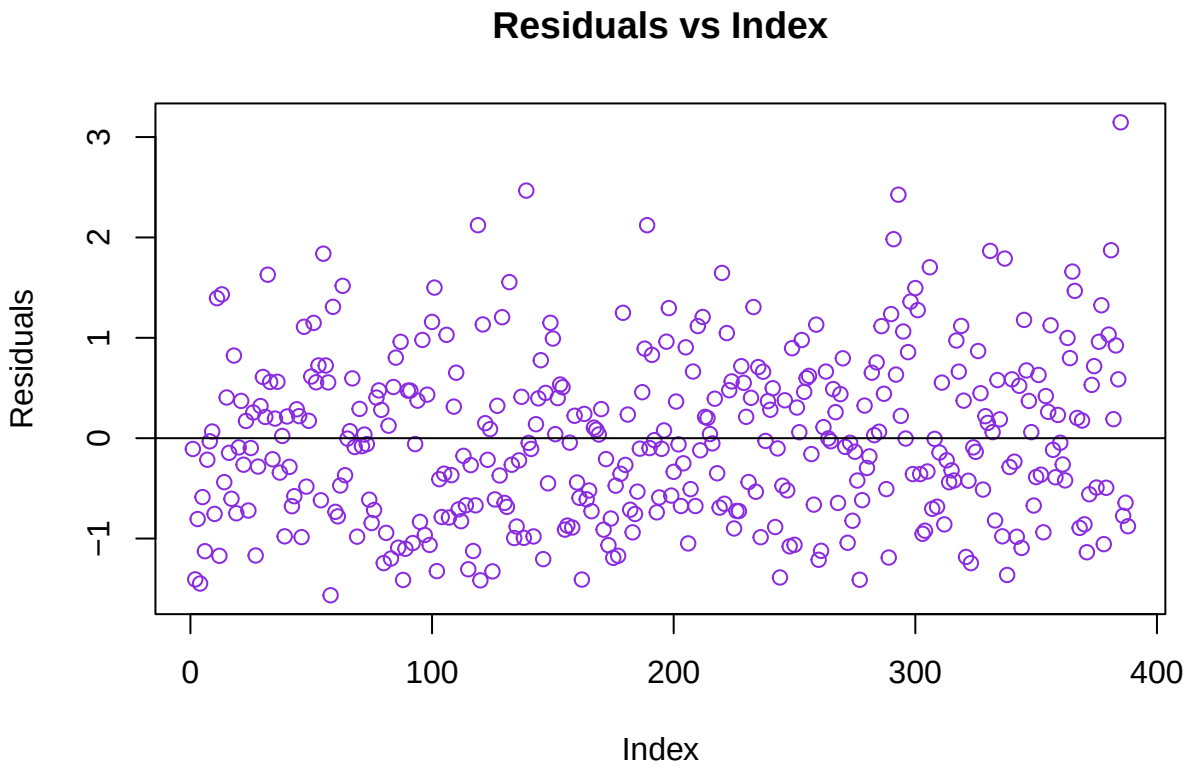
```
model3 <- lm(Organic ~ Flow + Temperature + Material, data = water)

summary(model3)
```

```
##
## Call:
## lm(formula = Organic ~ Flow + Temperature + Material, data = water)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56490 -0.66373 -0.04958  0.53189  3.14652
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.058621   0.117524  34.534  < 2e-16 ***
## Flow        0.023623   0.002272  10.396  < 2e-16 ***
## Temperature 0.094274   0.007173  13.142  < 2e-16 ***
## Material1   0.729081   0.119542   6.099 2.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.8215 on 384 degrees of freedom
## Multiple R-squared:  0.4977, Adjusted R-squared:  0.4938
## F-statistic: 126.8 on 3 and 384 DF,  p-value: < 2.2e-16
```

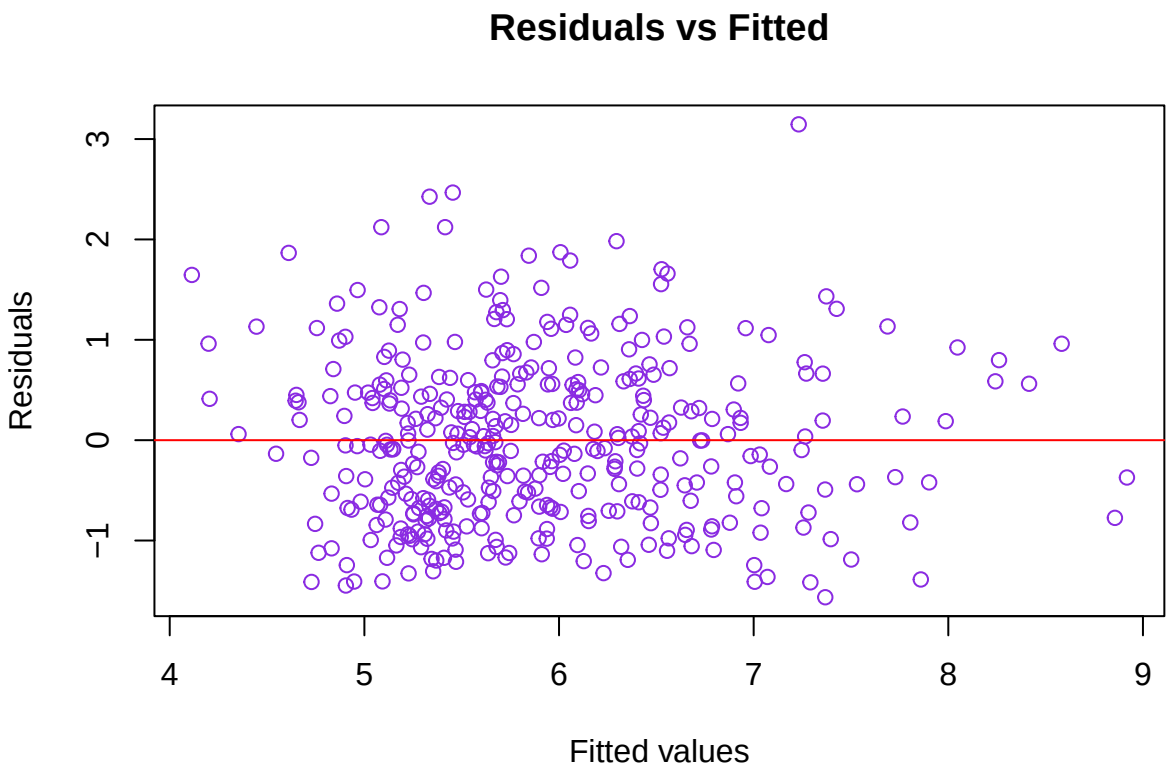
The residuals appear to be **randomly scattered**, indicating no evident violation of linearity or constant variance. However, the residual distribution shows a **light right skewness**, suggesting mild departure from normality. All of these hypothesis can be further investigated graphically.

```
plot(model3$residuals,
     main = "Residuals vs Index",
     col = "blueviolet",
     ylab = "Residuals")
abline(a=0, b=0)
```



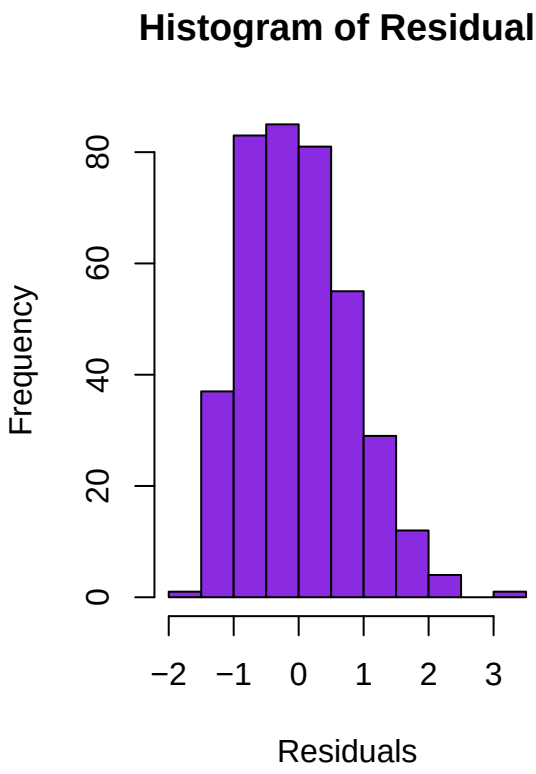
The **Residuals vs Index** plot does not reveal any particular trend or clustering, suggesting that the residuals are **approximately independent**. There is no evidence of autocorrelation or time-related structure in the data collection.

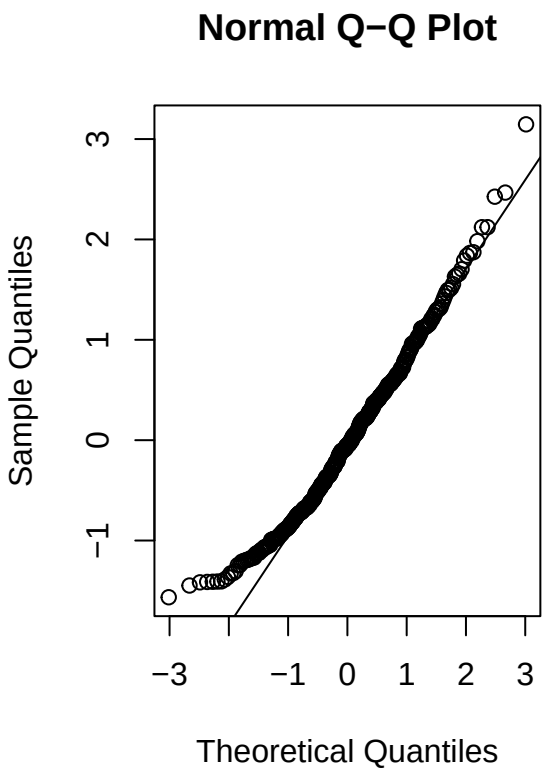
```
plot(fitted(model3), residuals(model3),
main = "Residuals vs Fitted",
xlab = "Fitted values",
ylab = "Residuals",
col = "blueviolet")
abline(h=0, col = "red")
```



The **Residuals vs Fitted** scatterplot shows a **fairly constant spread** of residuals across all levels of fitted values, with no clear funnel shape or curvature. This supports the assumption of **homoschedasticity** (constant variance) and **linearity** between the response and the predictors. A small amount of vertical spread asymmetry may hint at mild skewness, but this is not substantial.

```
par(mfrow = c(1,2))
hist(residuals(model3),
     main = "Histogram of Residuals",
     xlab = "Residuals",
     breaks = 10,
     col = "blueviolet")
qqnorm(residuals(model3)); qqline(residuals(model3))
```



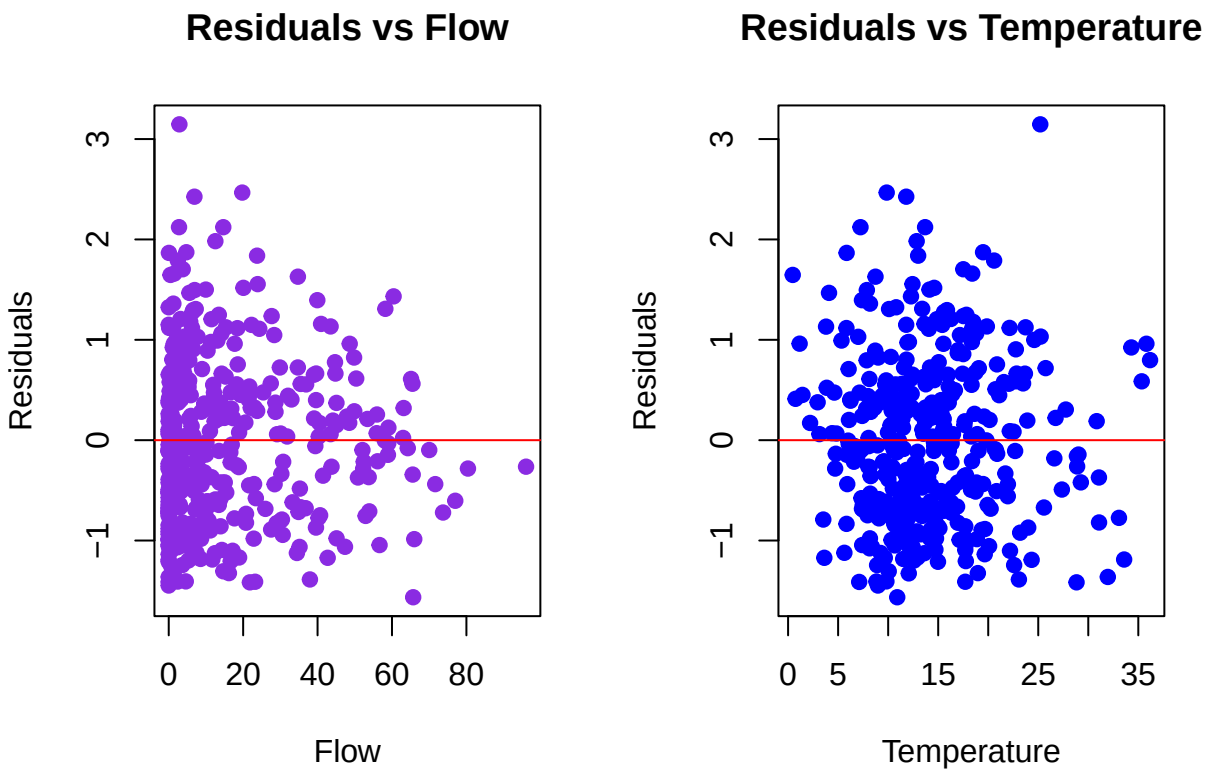


The histogram shows that the distribution of residuals is approximately symmetric but slightly right-skewed, with a longer tail on the positive side. Most residuals are concentrated around zero, which is consistent with what we expect under the normality assumption. The small skewness is not extreme and is acceptable given the sample size.

The Q-Q plot confirms the impression from the histogram: residuals closely follow the 45-degree line, especially in the central region. Mild deviations appear in the right tail, confirming a slight positive skew, but there is no major departure from normality. This suggests that the **normality assumption is reasonably satisfied**.

3.3 Depict the two scatterplots of the residuals with respect to the continuous covariates. Comment on the plots. We analyse two diagnotic plots to check the validity of linear model assumptions with respect to the two continuous predictors (`Flow` and `Temperature`).

```
par(mfrow = c(1, 2))
plot(water$Flow, residuals(model3),
     main = "Residuals vs Flow",
     xlab = "Flow", ylab = "Residuals",
     col = "blueviolet", pch = 19)
abline(h = 0, col = "red")

plot(water$Temperature, residuals(model3),
     main = "Residuals vs Temperature",
     xlab = "Temperature", ylab = "Residuals",
     col = "blue", pch = 19)
abline(h = 0, col = "red")
```



Residuals vs Flow: the residuals are **centered around zero**, in accordance with the assumption that the expected value of the errors is zero. However, the scatterplot reveals a **funnel-shaped pattern**: residuals exhibit higher variability for lower values of `Flow`, and become morre concentrated as flow increases. This indicates a **violation of the homoscedasticity assumption**, suggesting heteroscedasticity. Although the linearity assumption appears plausible (no visible curvature), the non-constant variance may affect the reliability of hypothesis testing and confidence intervals. In such cases, robust standard errors or variance-stabilizing transformations may be considered.

Residuals vs Temperature: the residuals appear to be randomly scattered across the full range of `Temperature`, with no clear pattern or curvature. The spread is fairly constant, and residuals remain centered around zero. This supports both the linearity and homoschedacity assumptions for the `Temperature` variable.

The residual plots suggest that the model performs reasonably well with respect to `Temperature`, while **potential heteroscedasticity is present** with respect to `Flow`. Although the violation is moderate, it should ne considered when interpreting inference results, particularlly those involving `Flow`.

3.4 What do we mean when we refer to the nonparametric bootstrap? What is it used for? The **nonparametric bootstrap** is a resampling-based technique used to approximate the **sampling distribution** of a statistic without relying on strong parametric assumptions about the underlying population. Given an original sample $\{x_1, x_2, \ldots, x_n\}$, the method treats the empirical distribution of the data as an approximation of the true (and unknown) population distribution. To perform the procedure, we generate B bootstrap samples by **resampling with replacement** from the original data. Each bootstrap sample $\{x_1^{*(b)}, x_2^{*(b)}, \ldots, x_n^{*(b)}\}$ is of the same size n, and for each sample, we compute the statistic of interest, denoted $\hat{\theta}^{*(b)}$, for $b = 1, \ldots, B$.

The collection $\{\hat{\theta}^{*(1)}, \ldots, \hat{\theta}^{*(B)}\}$ provides an approximation to the sampling distribution of $\hat{\theta}$, the statis-

tic computed on the original sample. This estimated distribution can be used to compute a **bootstrap standard error**, given by

$$\widehat{\mathrm{SE}}(\hat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^{*(b)} - \bar{\hat{\theta}}^{*}\right)^2},$$

where $\bar{\hat{\theta}}^{*} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{*(b)}$, and to construct **confidence intervals** using the **percentile method**, which selects the $(\alpha/2)$-th and $(1-\alpha/2)$-th quantiles of the bootstrap distribution:

$$\left[\hat{\theta}^{*}_{(\alpha/2)},\ \hat{\theta}^{*}_{(1-\alpha/2)}\right].$$

The bootstrap also allows for **bias estimation**, calculated as $\widehat{\mathrm{Bias}}(\hat{\theta}) = \bar{\hat{\theta}}^{*} - \hat{\theta}$.

The method is referred to as **nonparametric** because it makes no assumption about the functional form of the population distribution; instead, it relies solely on the observed data. This makes it particularly useful when the sampling distribution of a statistic is **analytically intractable**, or when the sample size is **too small** for asymptotic normal approximations to be reliable. Thanks to its **flexibility and minimal assumptions**, the nonparametric bootstrap has become a widely used tool in modern statistical inference.

Exercise 4

Consider the data in `rivers.Rdata` recording the lengths (in miles) of the 141 major rivers in North America (as compiled by the US Geological Survey).

```
load("rivers.Rdata")
skim_without_charts(rivers)
```

4.1 Illustrate the data using the appropriate descriptive statistics. Depict the scatter plot and comment according to the applicative context.

Table 6: Data summary

Name	rivers
Number of rows	141
Number of columns	1
Column type frequency:	
numeric	1
Group variables	None

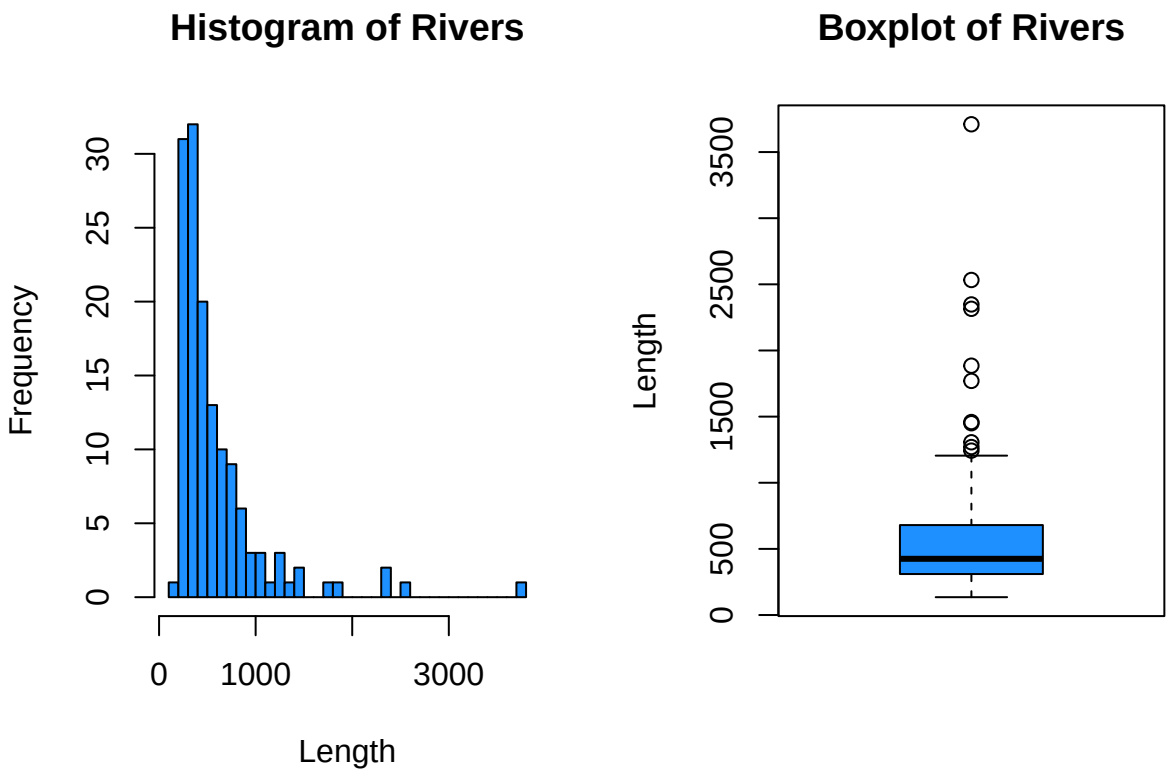
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
data	0	1	591.18	493.87	135	310	425	680	3710

The variable (lengths of 141 major rivers in North America) is numeric discrete, with no missing values, and represents a single measurement per observation.

The data present a **high degree of asymmetry**: the **mean** (591.2 miles) is considerably greater than the **median** (425 miles), indicating a **right-skewed distribution**. This is confirmed both by the **histogram** and the **boxplot** below, which show that most rivers are relatively short, while a few **extremely long rivers** pull the distribution to the right.

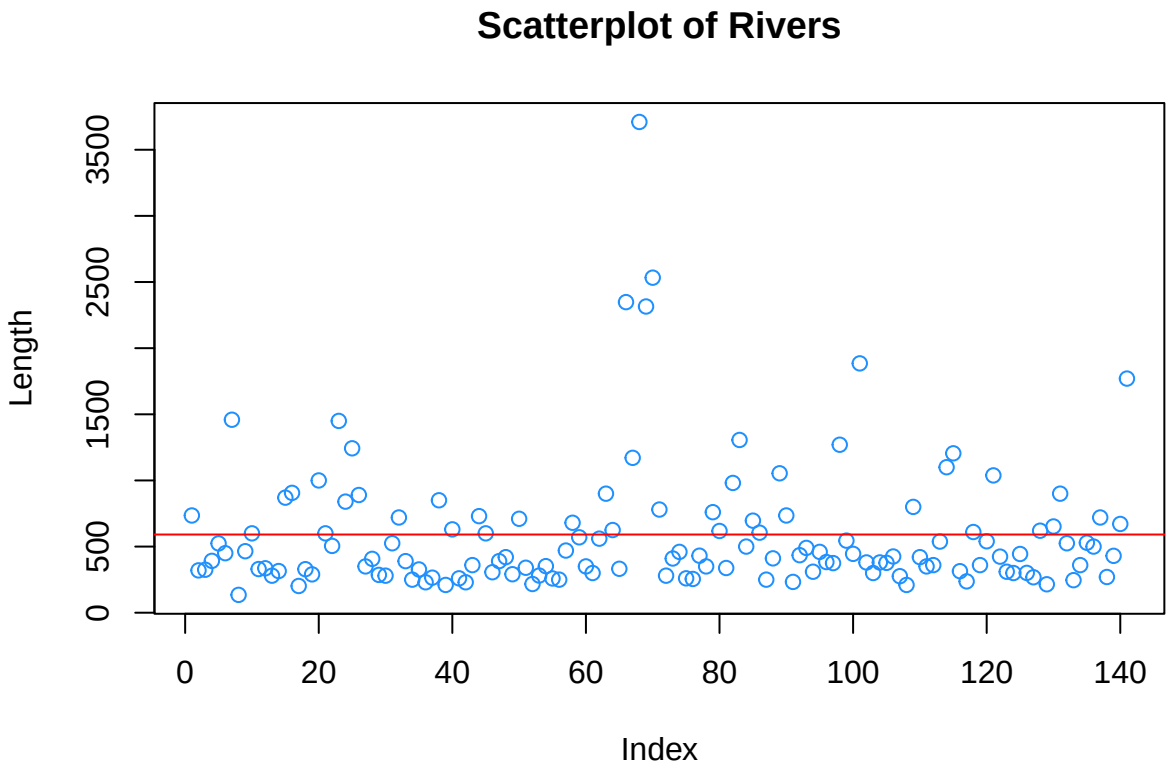
```
par(mfrow = c(1,2))

hist(rivers,
     main = "Histogram of Rivers",
     col = "dodgerblue1",
     xlab = "Length",
     breaks = 30)

boxplot(rivers,
        main = "Boxplot of Rivers",
        col = "dodgerblue1",
        ylab = "Length")
```



These longer rivers appear as outliers in the boxplot. The **interquartile range (IQR)**, computed as $Q_3 - Q_1 = 680 - 310 = 370$, indicates that the **central 50%** of river lengths fall within a **370-mile span**, ranging from 310 to 680 miles. Given that the total range extends from 135 to 3710 (range is equal to 3575 miles), the IQR is relatively narrow, reinforcing that the data are **heavily concentrated in the lower quantiles**, while the **extreme upper values** lie far outside the interquartile interval.

In the presence of right-skewed distributions, the IQR is especially valuable because it is **robust to outliers** and provides a **stable measure of dispersion**. This complements the standard deviation (493.9 miles), which—when interpreted via the **coefficient of variation** (CV $= \frac{\sigma}{\mu} \approx 0.84$)—suggests **high relative variability**. Such variability is common in environmental and geographical data, where complex natural processes lead to a small number of extreme values.

```
plot(rivers,
     main = "Scatterplot of Rivers",
     ylab = "Length",
     col = "dodgerblue1")

abline(h = mean(rivers), col = "red")
```



Finally, the scatterplot displays the lengths of the 141 major rivers as a function of their index in the dataset. The **horizontal red line** represents the **sample mean** length. This reference line helps visualize the **distribution of values around the mean**.

The plot shows that most rivers lie well below the mean, with a dense cluster between 100 and 1000 miles, and only a few rivers exceeding 2000 miles. These extreme values appear as **visible vertical deviations** above the red line, reinforcing the impression of a **right-skewed distribution**. The mean line itself is pulled upward by these few long rivers, highlighting that the mean is sensitive to outliers and may not represent the "typical" river length well.

There is no apparent ordering or temporal structure in the data: the river lengths appear **randomly dispersed** across the index, suggesting that the dataset is unordered and not sequential in time or geography. Overall, the scatterplot visually confirms the **asymmetry and heavy-tailed nature** of the distribution, already observed in the histogram and boxplot.

4.2 Consider the coefficient of variation as a parameter of interest. Provide an estimate of its standard error applying nonparametric bootstrap. Comment on the procedure and report the estimated value. We consider the coefficient of variation (CV) as the parameter of interest. It is defined as the ration between the standard deviation and the mean of a variable:

$$\text{CV} = \frac{\sigma}{\mu}$$

The CV is a **scale-free measure of relative dispersion**, and it's particularly useful when comparing variability across datasets with different units or magnitudes. Given the strongly right-skewed distribution of river lengths, estimating the **sampling variablity** of the CV analytically is not straightforward.

We therefore apply a **nonparametric bootstrap procedure** to estimate the standard error of the CV. This approach involves resampling from the observed data, computing the CV for each resample, and using the **empirical standard deviation** of the resulting bootstrap statistics to estimate the standard error of the original CV.

We first implement the procedure manually via `for` loop.

```
B4 <- 1000
n4 <- length(rivers)
Tboot4 <- rep(0, B4)

set.seed(123)

for (i in 1:B4) {
  Xstar4 <- sample(rivers, n4, replace = TRUE)
  Tboot4[i] <- sd(Xstar4)/mean(Xstar4) # statistic: coefficient of variation
}

round(sd(Tboot4), 3)
```

```
## [1] 0.098
```

This generates $B = 1000$ bootstrap replicates of the CV. Each replicate is computed on a random sample with replacement from the original 141 observations. The final result is:

$$\hat{\text{SE}}_{\text{CV}} = 0.098$$

We also use an alternative implementation with the `bootstrap()` function:

```
set.seed(123)

n4 <- length(rivers)  # alternative: n4 <- nrow(as.data.frame(rivers))

boot_fun4 <- function(ind) {
  sample_rivers <- rivers[ind]
  sd(sample_rivers) / mean(sample_rivers)
}

bootC4 <- bootstrap(1:n4, 1000, boot_fun4)

round(sd(bootC4$thetastar), 3)
```

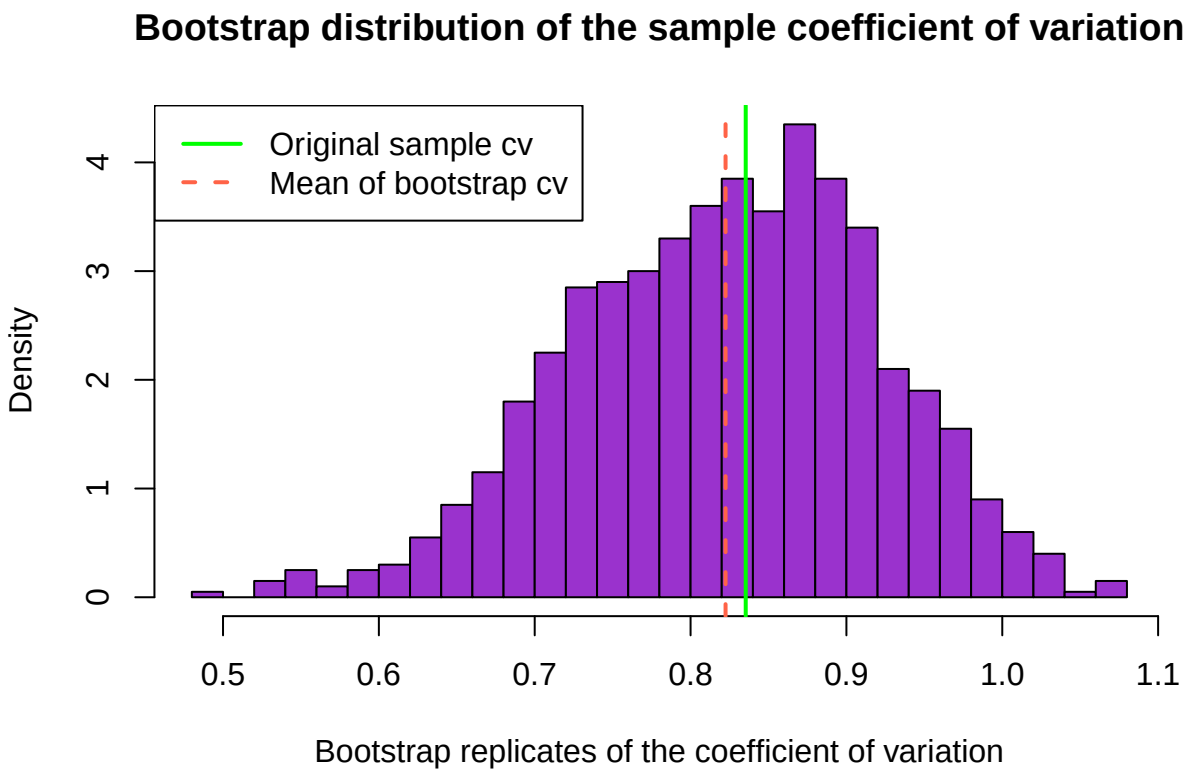
```
## [1] 0.097
```

This alternative version uses the `bootstrap()` utility, where the function `boot_fun4` computes the CV on each indexed bootstrap sample. The two implementations are equivalent in logic and yield virtually identical results:

$$\hat{\mathrm{SE}}_{\mathrm{CV}} = 0.097$$

The estimated bootstrap standard error is approximately 0.098, which is small relative to the CV itself (≈ 0.84). This suggests that the CV is a stable and reliable measure in context.

```r
cv4 <- sd(rivers)/mean(rivers)

hist(
  Tboot4,
  breaks = 30,
  probability = T,
  col = "darkorchid3",
  main = "Bootstrap distribution of the sample coefficient of variation",
  xlab = "Bootstrap replicates of the coefficient of variation")

abline(v = cv4, col = "green", lwd = 2) # original point estimate
abline(v = mean(Tboot4), col = "tomato", lwd = 2, lty = 2) # bootstrap mean

legend("topleft",
       legend = c("Original sample cv", "Mean of bootstrap cv"),
       col = c("green", "tomato"),
       lty = c(1, 2),
       lwd = 2)
```



The nonparametric bootstrap procedure is particularly appropriate here because:

- The CV has **no closed-form standard error**, especially under **non-normal or skewed distribution**.
- The data (river lengths) show a **high asymmetry and presence of outliers**, which can affect the normality assumptions required by classical methods.
- The sample size is moderate ($n = 141$), making resampling feasible and accurate.

By relying solely on the observed data and not assuming any underlying parametric distribution, the bootstrap standard error provides an empirical, assumption-free quantification of uncertainty.

Both implementations correctly estimate the bootstrap standard error of the coefficient of variation, with nearly identical results. The procedure is justified by the characteristics of the data (skewness, heavy tails), and the result confirms the stability of the CV estimator. This analysis underscores the power and flexibility of the nonparametric bootstrap in estimating variability when analytical solutions are unavailable or unreliable.

4.3 Provide a bootstrap confidence interval for the parameter at 99%. Depict the histogram of the bootstrap distribution adding the bars referring to the upper and lower bounds of the confidence interval and the mean of the bootstrap replicates along with the value of the estimate on the original sample. Add the legend and comment. To construct a nonparametric 99% confidence interval for the coefficient of variation of river lengths, we rely on the **bootstrap percentile method**. Given the lack of an analytical standard error or sampling distribution for the CV-particularly in the presence of skewed data like the river lengths-this approach allows us to estimate a confidence interval directly from the empirical distribution of bootstrap replicates. We compute the interval using the `quantile()` function instead of the `confint()` one.

```r
round(quantile(Tboot4, probs = c(0.005, 0.995)), 3)
```

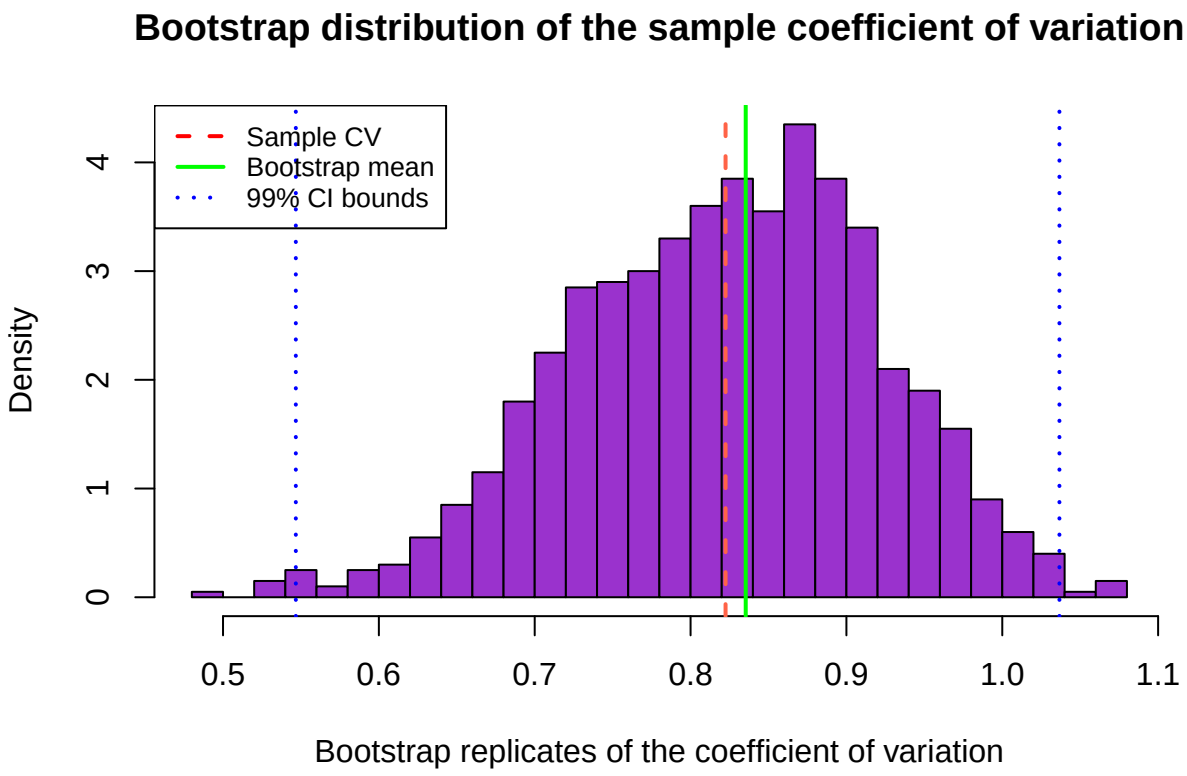
```
##  0.5% 99.5%
## 0.547 1.037
```

This is because the `confint()` function is designed for **model-based inference**, such as `lm()` or `glm()` objects, and produces confidence intervals based on **parametric assumptions** (e.g., normality or t-distribution). It does not apply to custom statistics like the coefficient of variation derived from a resampling procedure. Instead, we use:

$$\text{quantile}(\text{T}_{\text{boot}}, \text{probs} = \text{c}(\alpha/2, 1-\alpha/2))$$

to extract the percentiles from the empirical distribution of the B bootstrap replicates. This yields a distribution-free interval that is particularly robust under asymmetry or unknown population distributions.

```r
ci_bounds99_4 <- quantile(Tboot4, probs = c(0.005, 0.995))

hist(
  Tboot4,
  breaks = 30,
  probability = T,
  col = "darkorchid3",
  main = "Bootstrap distribution of the sample coefficient of variation",
  xlab = "Bootstrap replicates of the coefficient of variation")

abline(v = cv4, col = "green", lwd = 2) # original point estimate
abline(v = mean(Tboot4), col = "tomato", lwd = 2, lty = 2) # bootstrap mean

abline(v = ci_bounds99_4[1], col = "blue", lwd = 2, lty = 3)
abline(v = ci_bounds99_4[2], col = "blue", lwd = 2, lty = 3)

legend("topleft",
       legend = c("Sample CV", "Bootstrap mean", "99% CI bounds"),
       col = c("red", "green", "blue"),
       lty = c(2, 1, 3),
       lwd = 2,
       cex = 0.8)
```



The histogram of the bootstrap distribution confirms that the distribution of the CV estimates is **unimodal and approximately symmetric**, with no heavy skewness or multimodality. The red dashed vertical line represents the original estimate of the coefficient of variation, while the green line represents the bootstrap mean, which lies very close to it.

The blue dashed lines mark the lower and upper bounds of the 99% bootstrap percentile confidence interval. The alignment between the original estimate and the bootstrap mean suggests that the sample CV is **not biased**, and the width of the interval provides a realistic and data-driven measure of uncertainty.

The bootstrap percentile method yields a reliable and interpretable confidence interval for the coefficient of variation, without relying on parametric assumptions. The visual and numerical results indicate that the CV estimate is stable, unbiased and associated with a moderate degree of uncertainty, as expected given the variability and skewness of river lengths.

4.4 Illustrate the assumptions under the classical multiple linear regression model. In the context of the **classical multiple linear regression model**, we assume the following framework:

$$Y = X\beta + \varepsilon,$$

where:

- $Y \in \mathbb{R}^n$ is the vector of observed responses,
- $X \in \mathbb{R}^{n \times p}$ is the matrix of explanatory variables (including a column of 1s for the intercept),
- $\beta \in \mathbb{R}^p$ is the vector of unknown regression coefficients,
- $\varepsilon \in \mathbb{R}^n$ is the vector of random errors.

The **classical linear regression model** rests on the following key assumptions:

1. Linearity of the model in the parameters

The relationship between the dependent variable and the regressors is assumed to be **linear in parameters**:

$$\mathbb{E}[Y \mid X] = X\beta.$$

This implies that the expected value of the outcome is a linear combination of the predictors.

2. Independence of errors

The error terms $\varepsilon_1, \ldots, \varepsilon_n$ are assumed to be **statistically independent**. This ensures that the information contained in one observation provides no information about the error of another.

3. Homoscedasticity (constant variance)

The error terms are assumed to have **constant variance**:

$$\mathrm{Var}(\varepsilon_i) = \sigma^2 \quad \forall i.$$

This assumption ensures that the spread of residuals does not depend on the level of the fitted values or any predictor.

4. No perfect multicollinearity

The columns of the matrix X must be **linearly independent**. In practice, this means that **no explanatory variable is a perfect linear combination of the others**, and the matrix $X^T X$ is invertible.

5. Normality of errors (for inference)

Although not required for the consistency of the estimator, it is assumed that:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

This assumption is necessary for the **validity of hypothesis tests and confidence intervals**, ensuring that the sampling distribution of the OLS estimator is also normal:

$$\hat{\beta} \sim \mathcal{N}\left(\beta, \sigma^2 (X^T X)^{-1}\right).$$

Together, these assumptions guarantee that the **ordinary least squares (OLS)** estimator is **BLUE** (Best Linear Unbiased Estimator) under the **Gauss–Markov theorem**, and that standard inference procedures (t-tests, F-tests, confidence intervals) are valid. Violations of these assumptions may lead to biased, inefficient, or invalid results, and should therefore be carefully checked through diagnostic analysis.

Exam 3

Exercise 5

Consider the bivariate Gaussian distribution $(X_1, X_2) \sim N(\mu_X, \mu_Y, \sigma_X, \sigma_Y, \sigma_{XY})$.

5.1 Generate 3000 realizations from a bivariate Gaussian distribution with the following parameter values: $(X_1, X_2) \sim N(2, 2, 100, 100, 4)$. Set the seed to the value of 14263. Comment on the generated data data. We aim to simulate 3000 realizations from a **bivariate Gaussian distribution** with the following parameters:

$$(X_1, X_2) \sim \mathcal{N}_2 \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 10000 & 4 \\ 4 & 10000 \end{bmatrix} \right)$$

This implies:

- Both marginal distributions have **mean 2** and **variance 10000**, corresponding to a standard deviation of $\sqrt{10000} = 100$;
- The covariance between X_1 and X_2 is 4, indicating a **very weak positive linear dependence**, with a theoretical correlation coefficient of:

$$\rho = \frac{4}{100 \cdot 100} = 0.0004$$

We use the `rmvnorm()` function from the `mvtnorm` package, which requires:

- the number of observations `n`,
- the **mean vector** `mean`,
- the **covariance matrix** `sigma`.

```
library(mvtnorm)
set.seed(14263)

mu5 <- c(2, 2)
sigma5 <- matrix(c(10000, 4, 4, 10000), ncol = 2)

X5 <- rmvnorm(n = 3000, mean = mu5, sigma = sigma5)
summary(X5)
```

```
##        V1                  V2
##  Min.   :-369.191   Min.   :-377.298
##  1st Qu.: -69.113   1st Qu.: -70.715
##  Median :  -1.060   Median :  -2.325
##  Mean   :  -0.169   Mean   :  -2.367
##  3rd Qu.:  64.586   3rd Qu.:  65.844
##  Max.   : 350.655   Max.   : 328.552
```

The object `X5` is a **matrix of dimension** 3000×2. Each row corresponds to a simulated observation of the bivariate random variable, and each column to one of its components (X_1 or X_2).

Despite the theoretical mean being 2 for both variables, the observed sample means are:

- $\bar{X}_1 \approx -2.17$
- $\bar{X}_2 \approx -4.37$

This deviation may initially seem surprising given the large sample size. However, it is **entirely consistent with the properties of the sampling distribution of the mean** under high-variance settings. In fact, for each component:

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n} = \frac{10000}{3000} \approx 3.33 \quad \Rightarrow \quad \text{Standard error} \approx \sqrt{3.33} \approx 1.83$$

Thus, deviations of ± 2 units from the true mean are **statistically plausible**, even with $n = 3000$, due to the **enormous variance** of the underlying distribution.

```
sd(X5[,1])
```

```
## [1] 99.74342
```

```
sd(X5[,2])
```

```
## [1] 101.14
```

```
cov(X5[,1], X5[,2])
```

```
## [1] -10.63305
```

The estimated **standard deviations** are very close to the theoretical value of 100, confirming that the generated data reflect the target dispersion.

Regarding the **covariance**, the observed value may vary substantially from the theoretical value of 4—even turning negative in some samples—despite being generated under correct model assumptions. This is because 4 is **extremely small relative to the variance (10000)** of each component. The theoretical correlation is:

$$\rho = \frac{4}{100 \cdot 100} = 0.0004,$$

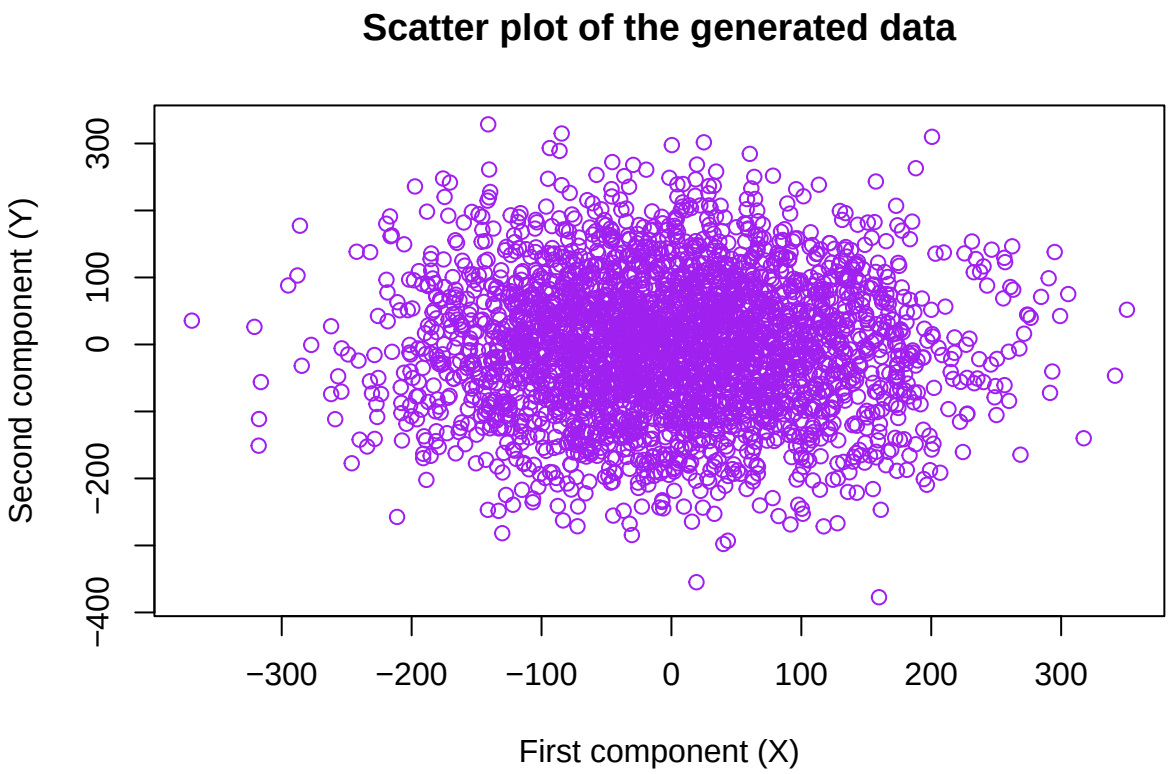
which indicates **practically no linear dependence** between the two variables. In such scenarios, the **sample covariance is highly unstable** and tends to **fluctuate around zero** across different realizations, due to the dominance of sampling noise over the true weak dependence. Therefore, observed values such as –10.6, or other moderate deviations from 4 are **statistically plausible** and **fully compatible** with the model.

To further illustrate this point, one could simulate multiple datasets and compute the sample means and covariances to observe their sampling variability.

The simulation was correctly performed and the data display behavior consistent with the theoretical properties of a bivariate Gaussian distribution. The apparent discrepancies in the sample means and sample covariance are **not anomalies**, but rather a **natural consequence of the high variance and low correlation** in the data-generating process. These results highlight the importance of interpreting empirical estimates in light of the **scale and structure of the model parameters**.

5.2 Draw the scatter plot of the realizations and comment on it. We visualize the realizations from the simulated bivariate Gaussian distribution by plotting the two components on the Cartesian plane.

```
plot(X5[,1], X5[,2],
     col = "purple",
     main = "Scatter plot of the generated data",
     xlab = "First component (X)",
     ylab = "Second component (Y)")
```



The resulting scatter plot shows the cloud of 3000 simulated points distributed in the plane. The horizontal and vertical axes represent the first and second components of the bivariate vector (X_1, X_2), respectively.

As expected for a bivariate Normal distribution with large, equal variances and nearly zero covariance, the points form a **symmetrical elliptical cloud** centered approximately around the theoretical mean vector $(2, 2)$. Due to the large variances ($\sigma^2 = 10000$), the cloud spans a wide range (about 300 units) along both axes. The two marginal dispersions are visually similar, consistent with the fact that both components share the same variance.

Importantly, the **elliptical shape is not tilted**, meaning there is **no pronounced linear orientation** of the points. This is consistent with the **theoretical covariance of 4**, which implies a **correlation coefficient** of only:

$$\rho = \frac{4}{100 \cdot 100} = 0.0004,$$

i.e., **virtually no linear dependence** between the components. The lack of inclination in the elliptical cloud is a direct geometric reflection of this weak correlation.

In conclusion, the scatter plot confirms the theoretical properties of the simulated distribution: the data are symmetrically spread, the variation along both axes is nearly identical, and the absence of directional pattern is consistent with a **near-zero covariance**. The elliptical, nearly circular shape of the point cloud is characteristic of a **bivariate Normal distribution with uncorrelated components** and equal variances.

5.3 Calculate the theoretical density for some values of X and Y . Depict the density plot in three dimensions. Comment on the shape of the distribution. We compute and visualize the **theoretical density** of the bivariate Gaussian distribution used in the simulation:

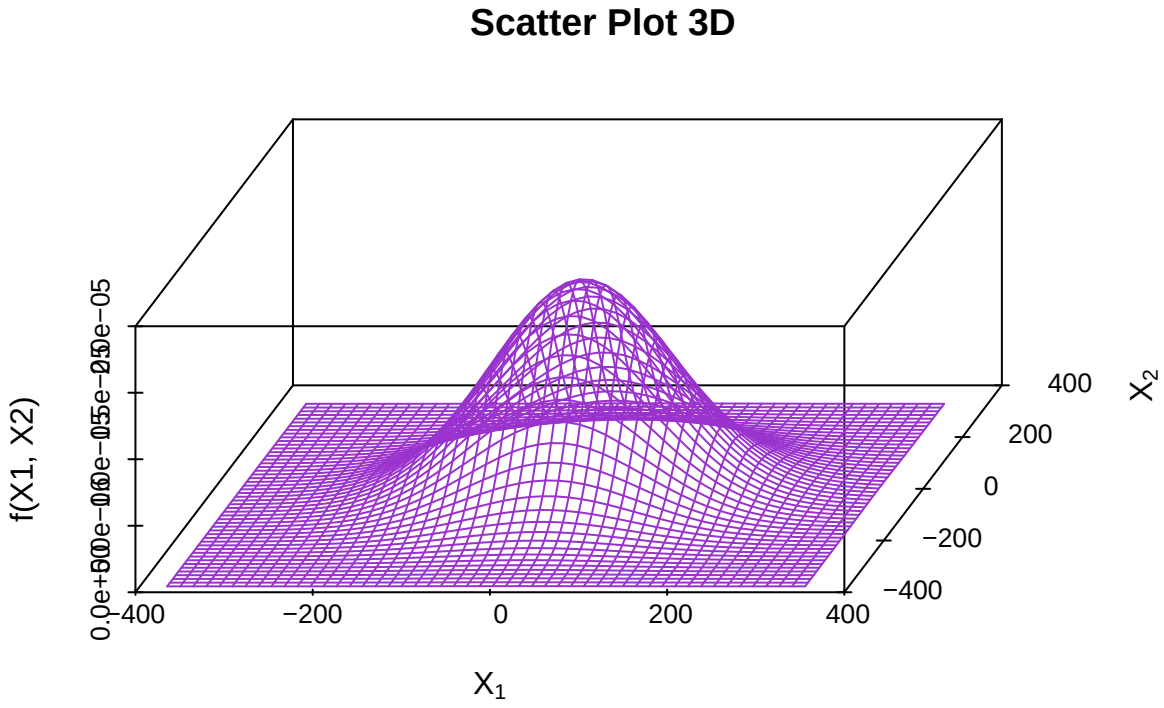
$$(X_1, X_2) \sim \mathcal{N}_2 \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 10000 & 4 \\ 4 & 10000 \end{bmatrix} \right)$$

To do so, we evaluate the joint probability density function $f(x_1, x_2)$ over a regular grid of values for X_1 and X_2. The function `dmvnorm()` from the `mvtnorm` package is used to compute the values of the theoretical bivariate normal density at each grid point.

```
require(scatterplot3d)
```

```
## Loading required package: scatterplot3d
```

```
z1 <- seq(min(X5[, 1]), max(X5[, 1]), length.out = 50)
z2 <- seq(min(X5[, 2]), max(X5[, 2]), length.out = 50)
grid <- expand.grid(z1, z2)

dens <- matrix(dmvnorm(grid, mean = mu5, sigma = sigma5),
               ncol = length(z1), byrow = TRUE)

plot3d <- scatterplot3d(z1, z2, seq(min(dens), max(dens),
                                    length.out = length(z1)),
                        xlim = c(min(X5[, 1]), max(X5[, 1])),
                        ylim = c(min(X5[, 2]), max(X5[, 2])),
                        type = "n", angle = 70, grid = FALSE,
                        main = "Scatter Plot 3D",
                        xlab = expression(X[1]),
                        ylab = expression(X[2]),
                        zlab = "f(X1, X2)")

for (i in length(z1):1) {
  plot3d$points3d(rep(z1[i], length(z2)), z2, dens[i, ],
                  type = "l", col = "darkorchid3")
}

for (i in length(z2):1) {
  plot3d$points3d(z1, rep(z2[i], length(z1)), dens[, i],
                  type = "l", col = "darkorchid3")
}
```



The surface depicted in the 3D wireframe plot represents the **theoretical joint density function** of (X_1, X_2). The plot shows a **bell-shaped peak** centered around the mean vector $(2, 2)$, with the density gradually decreasing as one moves away from the center.

Due to the very **large and equal variances** ($\sigma_1^2 = \sigma_2^2 = 10000$) and the **negligible covariance** ($\text{Cov}(X_1, X_2) = 4$), the density exhibits a shape that is **approximately radially symmetric** and **slightly elliptical** (close to circular in horizontal sections). The peak is flat and wide, indicating that the probability mass is **spread over a large area**, which is expected with such high variance.

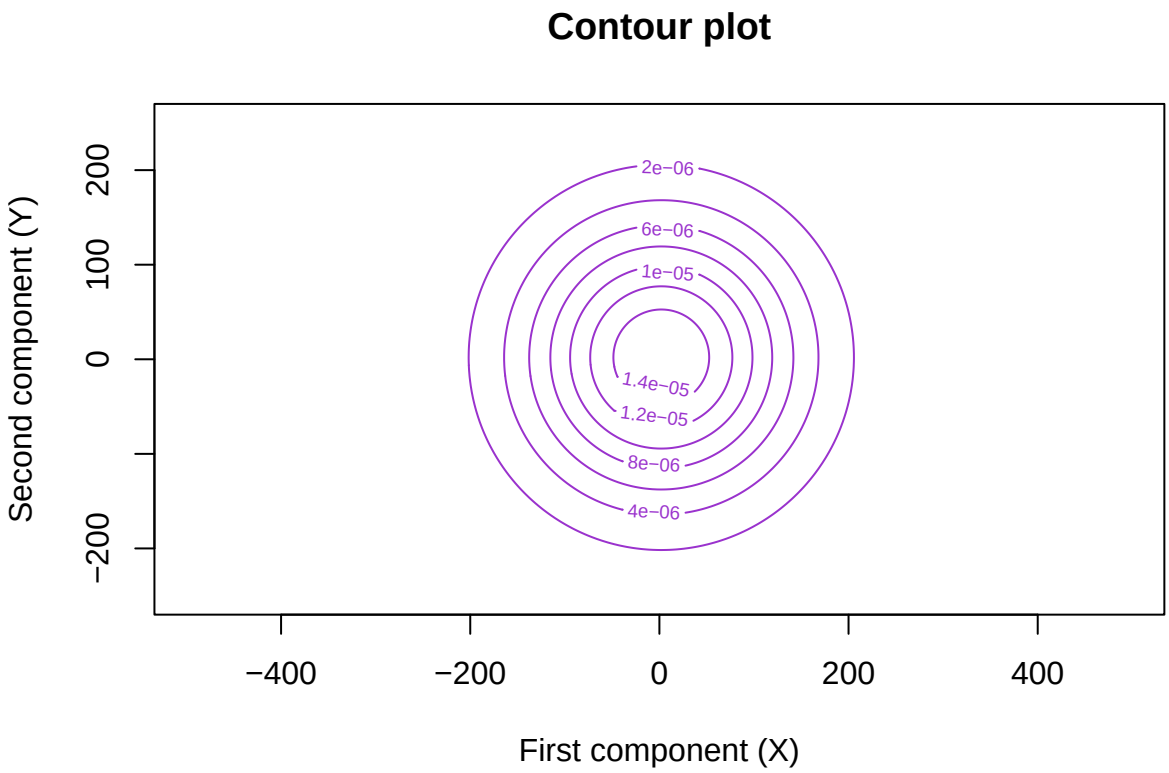
The **absence of visible tilt or directional elongation** in the density confirms the **near-zero correlation** between the two components. If a stronger covariance were present, the density contours would appear **tilted** along the direction of maximum joint variability.

The 3D density plot faithfully represents the theoretical shape of a bivariate Gaussian distribution with equal variances and negligible correlation. The bell-shaped profile and the gentle curvature reflect the high dispersion of the simulated data and confirm the probabilistic structure imposed in the simulation phase (Exercise 5.1). This visual tool complements the 2D scatterplot by providing a geometric understanding of how the joint density is structured in three dimensions.

5.4 Draw and interpret the contour plots obtained from the theoretical density calculated at the previous point. The contour plot provides a two-dimensional representation of the **level curves** of the theoretical bivariate Gaussian density function. It is obtained by intersecting the **three-dimensional density surface** with a sequence of horizontal planes at increasing heights. Each contour line thus represents a set of points with **equal joint density** $f(x_1, x_2)$.

```
z1 <- seq(-250, 250, length.out = 300)
z2 <- seq(-250, 250, length.out = 300)
```

```r
griglia <- expand.grid(z1, z2)
dens <- matrix(dmvnorm(griglia, mean = mu5, sigma = sigma5), ncol = length(z1))
contour(z1, z2, dens, col = "darkorchid3",
        main = "Contour plot",
        xlab = "First component (X)",
        ylab = "Second component (Y)",
        asp = 1)
```



The contour lines form a series of concentric circles centered near the origin $(2, 2)$, which is the mean vector of the distribution. This pattern reveals two important features of the bivariate Gaussian density:

1. **Equal marginal variances**: since both variables X_1 and X_2 have the same variance, the level sets of the density are **isotropic**-i.e., they spread equally in all directions.
2. **Zero correlation**: the lack of elliptical tilt or orientation confirms that the correlation between the two components is **approximately zero**. In a bivariate normal distribution, the presence of correlation causes the contour lines to appear as tilted ellipses, oriented along the direction of greatest joint variability. Here, the near-zero covariance ($\text{Cov}(X_1, X_2) = 4$) leads to contours that are essentially circular.

The **center of the contours** corresponds to the **peak of the density**, located at the mean of distribution, confirming the symmetric and unimodal nature of the bivariate Gaussian law.

The contour plot confirms the **theoretical geometry** of the bivariate Gaussian distribution used in the simulation: circular symmetry, no preferential direction of association, and a central maximum located at the mean. This visual representation complements the 3D surface and the scatter plot, providing further confirmation of the properties induced by the chosen covariance structure.

5.5 List the suitable properties of an estimator and explain their importance. In statistical inference, an **estimator** is a rule or function used to infer the value of an unknown population parameter based on a sample of observed data. A **good estimator** should satisfy several desirable properties, which ensure that the inference is reliable, interpretable, and efficient. The most important properties are listed and explained below:

1. **Unbiasedness**

An estimator $\hat{\theta}$ is said to be **unbiased** for a parameter θ if its expected value equals the true parameter:

$$\mathbb{E}[\hat{\theta}] = \theta$$

This means that, on average, the estimator neither overestimates nor underestimates the parameter. Unbiasedness is a foundational property because it ensures that there is **no systematic error** in the estimation process.

2. **Consistency**

An estimator is **consistent** if it converges in probability to the true value of the parameter as the sample size increases:

$$\hat{\theta}_n \xrightarrow{P} \theta \quad \text{as} \quad n \to \infty$$

This guarantees that with enough data, the estimator will yield values **arbitrarily close** to the true parameter. Consistency is critical for large-sample validity and for justifying asymptotic approximations.

3. **Efficiency**

Among all unbiased estimators, an estimator is said to be **efficient** if it has the **lowest possible variance**. In finite samples, this means:

$$\text{Var}(\hat{\theta}) \leq \text{Var}(\tilde{\theta}) \quad \text{for any other unbiased } \tilde{\theta}$$

In large samples, efficiency is often discussed in terms of **asymptotic variance**. Efficient estimators make the best possible use of the available information and provide **more precise estimates**.

4. **Sufficiency**

An estimator is **sufficient** for a parameter θ if it captures **all the information** in the sample relevant to estimating θ. Formally, a statistic $T(X)$ is sufficient if the conditional distribution of the data given $T(X)$ does not depend on θ. Sufficiency is important because it implies that **no additional information** about the parameter can be gained from the sample beyond what is contained in the estimator.

5. **Asymptotic normality**

Although not strictly necessary, it is desirable that an estimator be **asymptotically normal**, i.e., that its distribution converges to a normal distribution as the sample size increases:

$$\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

This property enables the construction of **approximate confidence intervals** and hypothesis tests using standard normal theory.

These properties—**unbiasedness, consistency, efficiency, sufficiency, and asymptotic normality**—serve as criteria for evaluating and comparing estimators. While not all can be satisfied simultaneously in every context, an estimator that possesses several of these properties is typically **preferable** for statistical inference, as it provides accurate, stable, and interpretable estimates of the unknown population parameters.

Exercise 6

Researchers of an American company of micro devices collected data recording the following variables: - circuit board assembly support costs (`Scost`) - direct labour hours (`Hours`) - number of boards completed (`Boards`) - average cycle time of board assembled during a week (in seconds, `Cycle`)

The data is found in `device.Rdata`.

```
load("device.Rdata")
skim_without_charts(device)
```

6.1 Provide a summary of the observed data and comment. Depict the empirical cumulative distribution function of the variable `Scost` and comment on its shape and on some of its values.

Table 8: Data summary

Name	device
Number of rows	100
Number of columns	4
Column type frequency:	
numeric	4
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Scost	0	1	12.00	3.26	5.50	10.50	11.50	13.50	24.50
Hours	0	1	1.57	0.37	0.55	1.36	1.64	1.77	2.33
Boards	0	1	1.63	0.40	0.54	1.42	1.70	1.85	2.50
Cycle	0	1	0.42	0.12	0.14	0.35	0.43	0.50	0.69

The dataset contains 100 observations across 4 numeric variables. From the `skim_without_charts()`, we observe:

- The mean and median are closely aligned for all variables, suggesting approximate symmetry;
- The variable `Scost` shows the largest standard deviation, indicating more dispersion in cost data than in other dimensions;

- The interquartile ranges (IQRs) are narrow for all variables, reflecting that most values are concentrated around the center.

We also compute the coefficient of variation to compare variability relative to the mean.

```
cv6 <- function(x) sd(x) / mean(x)
round(apply(device, 2, cv6), 2)
```

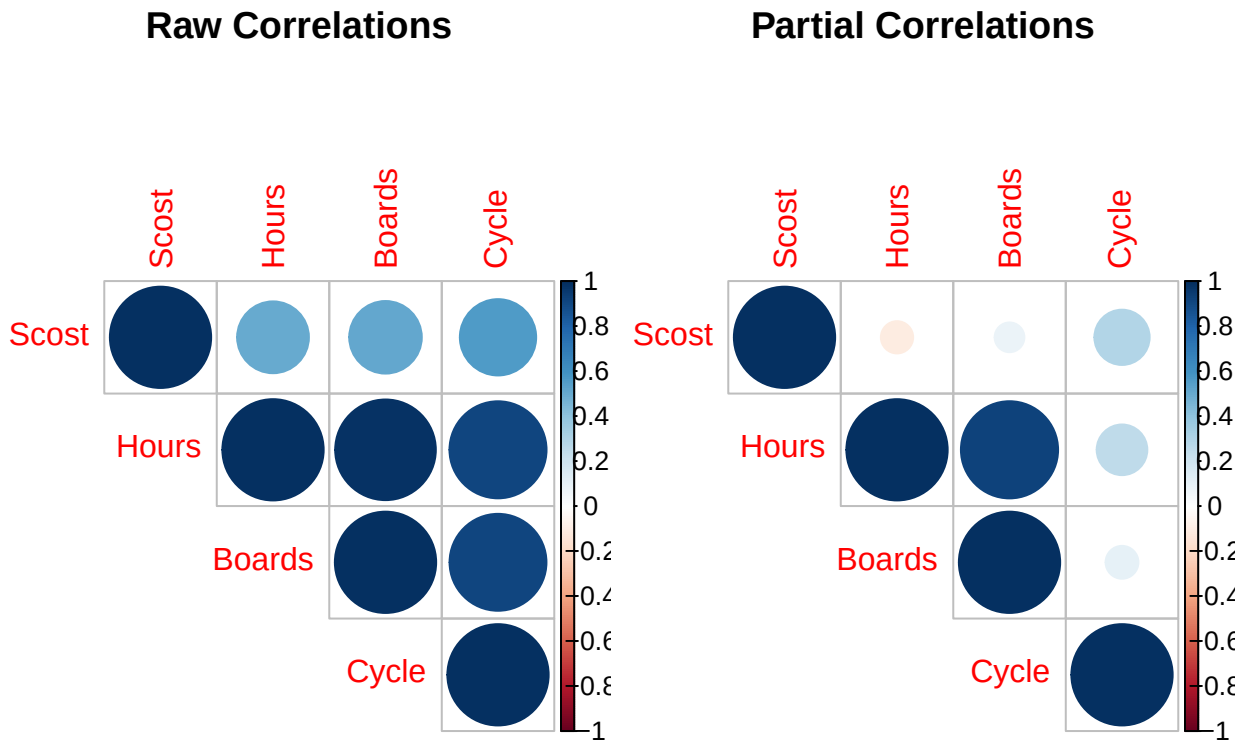
```
##  Scost  Hours Boards  Cycle
##   0.27   0.24   0.25   0.29
```

All variables exhibit a comparable relative variability, with CVs between 0.24 and 0.29. This suggests that none of the variables dominate in scale or dispersion.

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(ggm)
C_raw6 <- cor(device) # compute raw correlation matrix

S6 <- cov(device) # compute partial correlation matrix
C_partial6 <- parcor(S6)

par(mfrow = c(1, 2))
corrplot(C_raw6, type = "upper", title = "Raw Correlations", mar = c(0, 0, 2, 0))
corrplot(C_partial6, type = "upper", title = "Partial Correlations", mar = c(0, 0, 2, 0))
```



The **raw correlation matrix** reveals strong pairwise associations between:

- `Cycle` and `Hours`
- `Boards` and `Hours`
- `Cycle` and `Boards`

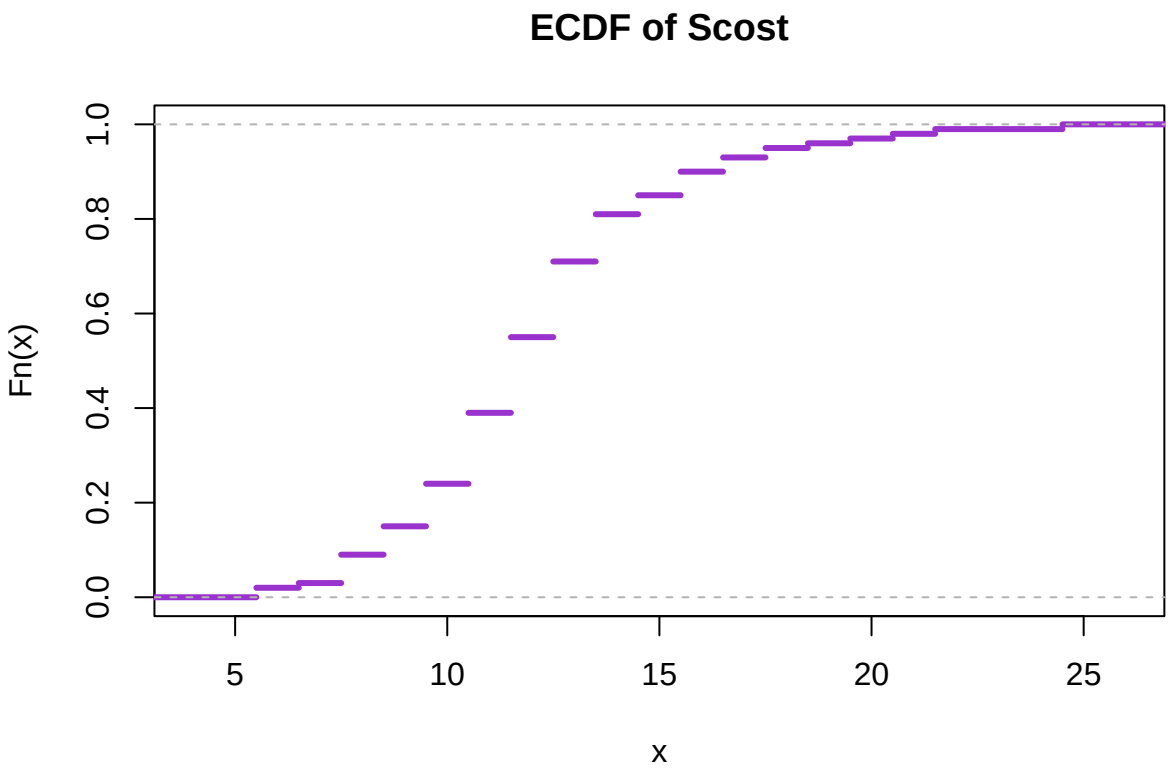
In the **partial correlation matrix**, these associations are **substantially weakened**, reflecting the effect of controlling for the other variables:

- Partial correlation are all below 0.5, with the exception of `Hours-Boards`
- Interestingly, the partial correlation between `Scost` and `Hours` becomes negative, suggesting that the raw positive correlation was mediated by the other covariates.

This underscores the importance of using partial correlations to assess the direct association between variables, removing confounding influences.

We visualize the empirical cumulative distribution function (ECDF) of the variable `Scost` with:

```
plot(ecdf(device$Scost),
     col = "darkorchid3",
     lwd = 3,
     do.points = FALSE,
     main = "ECDF of Scost")
```



The **ECDF is smooth and steadily increasing**, with no flat segments or sudden jumps. This reflects the fact that `Scost` is a **continuous, finely measured variable**.

Its shape appears **almost linear and symmetric around the center**, with a **gentle slope** throughout. It **does not display clear skewness**, as also confirmed by the mean (12.0) and median (11.5) being close, and by the symmetric interquartile range (IQR = Q3 – Q1 approximately 13.5 – 10.5 = 3).

- 10% of observations fall below 8.5;
- 50% of the values lie below 11.5, the **sample median**;
- 80% of the values are below 14.5;
- The support costs range from 5.5 (minimum) to 24.5 (maximum), but **most values are concentrated** in the interval [8, 16].

This behavior reflects a **moderate spread**, confirmed by the **standard deviation approximately 3.26**, and a **lack of outliers**, as no abrupt jumps are observed at the extremes.

The ECDF suggests that `Scost` follows a **fairly symmetric and continuous distribution**, possibly **compatible with a Gaussian model**, though this should be further tested using **QQ plots** or **normality tests**. Overall, the variable is **well-behaved** and suitable for standard parametric modeling.

6.2 Estimate a clustering model supposing a finite mixture model with three components and variance covariance structure of type EII is suitable. Describe the estimated parameters provided by the summary function of the selected model at the previous point. We estimate a **Gaussian finite mixture model** using the `Mclust()` function, imposing:

- **Three components** ($G = 3$),

- **Covariance structure** `"EII"`: equal volume, spherical shape, and no correlation between variables.

```
require(mclust)
```

```
## Loading required package: mclust
```

```
## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:mvtnorm':
##
##     dmvnorm
```

```
## The following object is masked from 'package:bootstrap':
##
##     diabetes
```

```
mod6 <- Mclust(device, G = 3, modelNames = "EII")
summary(mod6, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -562.2066 100 15 -1193.491 -1196.261
##
## Clustering table:
##  1  2  3
## 61 15 24
##
## Mixing probabilities:
##         1         2         3
## 0.6054263 0.1492567 0.2453170
##
## Means:
##              [,1]       [,2]      [,3]
## Scost  12.0534064 17.7764679 8.3536575
## Hours   1.6794711  1.7288855 1.1831886
## Boards  1.7538520  1.8181658 1.2025195
## Cycle   0.4487108  0.4962678 0.3009061
##
## Variances:
## [,,1]
##            Scost     Hours    Boards     Cycle
## Scost  0.6270139 0.0000000 0.0000000 0.0000000
```

```
## Hours  0.0000000 0.6270139 0.0000000 0.0000000
## Boards 0.0000000 0.0000000 0.6270139 0.0000000
## Cycle  0.0000000 0.0000000 0.0000000 0.6270139
## [,,2]
##            Scost     Hours    Boards     Cycle
## Scost  0.6270139 0.0000000 0.0000000 0.0000000
## Hours  0.0000000 0.6270139 0.0000000 0.0000000
## Boards 0.0000000 0.0000000 0.6270139 0.0000000
## Cycle  0.0000000 0.0000000 0.0000000 0.6270139
## [,,3]
##            Scost     Hours    Boards     Cycle
## Scost  0.6270139 0.0000000 0.0000000 0.0000000
## Hours  0.0000000 0.6270139 0.0000000 0.0000000
## Boards 0.0000000 0.0000000 0.6270139 0.0000000
## Cycle  0.0000000 0.0000000 0.0000000 0.6270139
```

The model successfully fits the data via the EM algorithm, yielding a **log-likelihood of -562.21**, with **BIC = -1193.49** and **ICL = -1196.26**. These criteria evaluate model quality: the **log-likelihood** measures how well the model fits the data, while the **BIC** (Bayesian Information Criterion) and **ICL** (Integrated Completed Likelihood) add a **penalty for complexity** (e.g. number of parameters). Lower BIC and ICL values indicate better trade-off between goodness-of-fit and parsimony, helping to avoid overfitting.

The clustering result assigns:

- 61 units to component 1,
- 15 to component 2,
- and 24 to component 3.

The **estimated mixing probabilities** reflect these proportions: roughly 60.5%, 14.9% and 24.5%.

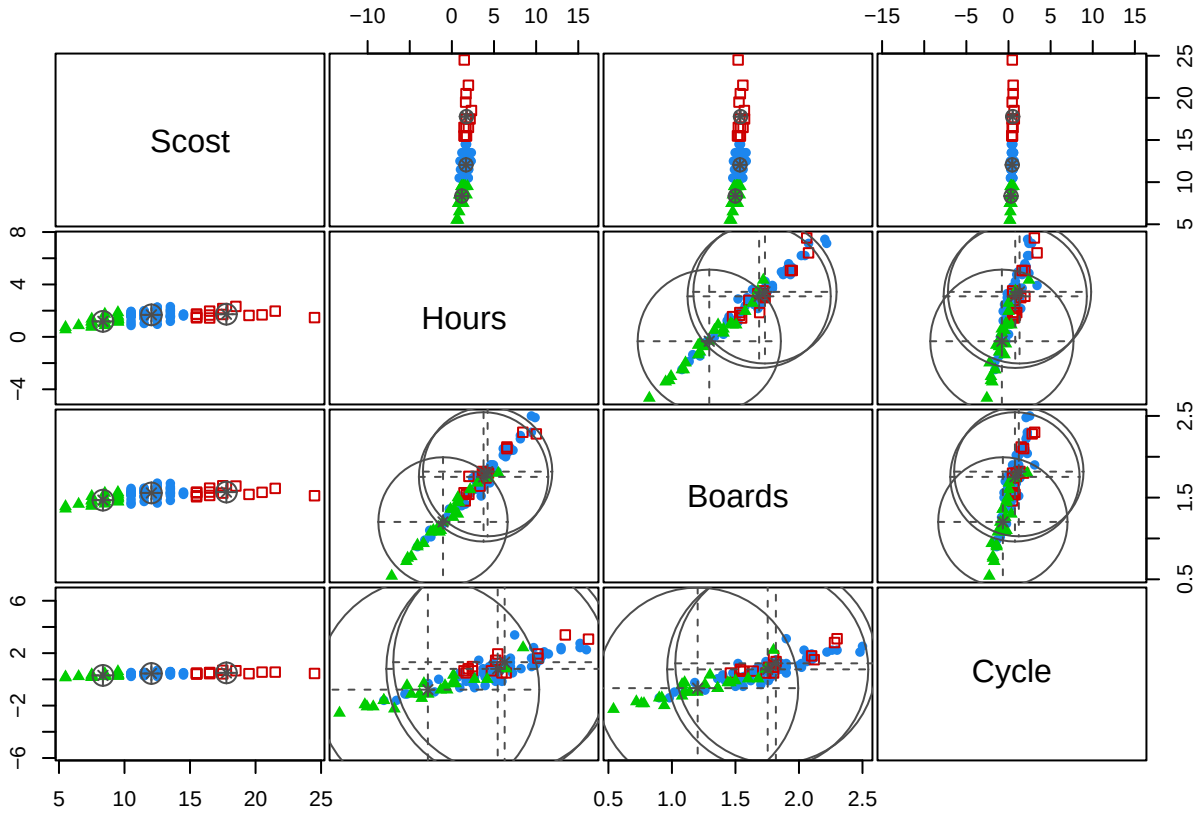
From the `summary()` output with `parameters = TRUE`, we retrieve the estimated **means** for each variable (`Scost`, `Hours`, `Boards`, `Cycle`) across the three latent groups. Group 2 is characterized by the **highest mean cost** (≈ 17.78) and longer times, suggesting a more expensive or slower process. Group 3 shows the **lowest values** across all variables (e.g. cost 8.35), possibly identifying more efficient or lightweight production cycles. Group 1 lies in between and represents the **dominant, average behavior**.

All components share the same **diagonal covariance matrix**, with **equal variance** 0.627 along each variable and **no covariance**, consistent with the EII assumption. This implies that all groups are **spherical and equally dispersed**, meaning classification is driven purely by location (means), not by shape or orientation.

The model provides a good starting point for clustering-based interpretation of production profiles, with clear latent structure and interpretable group-specific means. Further steps could include visualizing classification or comparing to alternative models using BIC.

6.3 Plot the estimated clustering classification and comment on the figure. Report the estimate posterior probabilities for units 22, 43, 56, and 87. Describe them. To assess the quality of the clustering model previously estimated, we visualize the classification results.

```
plot(mod6, what = "classification", asp = 1)
```



This produces a pairwise scatterplot matrix, where the points are colored and shaped according to their assigned cluster (from the mixture model with 3 components and spherical covariance structure `EII`). Each point is displayed in the space of the original variables, and the ellipses represent confidence regions of the Gaussian components.

Despite some overlap-especially between clusters 1 and 3-the clusters are reasonably well-separated. The ellipses confirm that the model assumes equal shape and size (consistent with `EII` structure), and the clustering follows the observed patterns in the data:

- Cluster 1 (blue circles) generally corresponds to intermediate `Scost` and `Cycle` values,
- Cluster 2 (green triangles) groups the highest `Scost` values,
- Cluster 3 (red squares) includes the lowest values in `Scost`, `Hours` and `Boards`.

We finally examine **posterior classification probabilities** for four specific units.

```
ids <- c(22, 43, 56, 87)
probs <- mod6$z[ids, ]
rownames(probs) <- ids
round(probs, 4)
```

```
##      [,1] [,2]  [,3]
## 22 0.9999    0 1e-04
## 43 1.0000    0 0e+00
## 56 0.0000    0 1e+00
## 87 0.9999    0 1e-04
```

These values indicate an **extremely high certainty** in cluster assignment:

- Units 22, 43 and 87 are all assigned to Cluster 1 with posterior probabilities > 0.9999.
- Unit 56 is assigned to Cluster 3 with a posterior probability exactly equal to 1.

Such high values are typical of well-separated clusters, suggesting that these units lie deep inside their respective Gaussian components and far from the classification boundaries. These posterior probabilities confirm the robustness of the model-based clustering and support the validity of the segmentation in this specific dataset.

Exam 4

Exercise 7

How do the types of content, hours posted, and paid advertisements affect the **share** of a particular cosmetic brand on Facebook? To address this question, researchers collected data recording the following variables: number of shares (`share`), type of content (link, status, photo/video, `Type`), hour posted (`Post.Hour`) and paid advertisement (`Paid`, no coded as 0 and yes as 1). The data is found in `fbshare.Rdata`.

7.1 Specify the sample size and comment on main summaries for the response variable and for the covariate Type. Is the type of post associated with the observed number of shares? To investigate whether the **type of content** posted influences the **number of shares**, we begin by exploring the dataset `fbshare`, which includes information on 371 Facebook posts.

```
load("fbshare.Rdata")
skim_without_charts(fbshare)
```

Table 10: Data summary

Name	fbshare
Number of rows	371
Number of columns	4
_____	
Column type frequency:	
factor	2
numeric	2
_____	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Type	0	1	FALSE	3	Pho: 323, Sta: 28, Lin: 20
Paid	0	1	FALSE	2	0: 268, 1: 103

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
share	0	1	14.96	8.94	0	8	14	22	32
Post.Hour	0	1	8.00	4.40	1	3	9	11	23

From the output, we observe that the dataset contains **371 observations** and 4 variables: two are numeric (`share` and `Post.Hour`) and two are categorical (`Type` and `Paid`). The `share` variable, which is our **response of interest**, measures how many times each post has been shared.

```
sd(fbshare$share) / mean(fbshare$share)
```

```
## [1] 0.5971613
```

This code computes the **coefficient of variation (CV)** for the `share` variable. The result is approximately 0.60, indicating a **moderate-to-high level of relative variability**. In practice, this means that although many posts receive similar numbers of shares, the spread around the mean ($\approx$ 15) is substantial. The summary statistics show a median of 14 and interquartile range from 8 to 22, suggesting that most values are concentrated in this interval and the distribution is **reasonably symmetric**.

As for the variable `Type`, which encodes the **type of content**, we see that it has **three unique levels**:

- `"Photo"`: 323 posts
- `"Status"`: 28 posts
- `"Link"`: 20 posts

The imbalance is quite evident, with **photo/video posts making up the vast majority**. This can potentially introduce **bias** or **instability** in further comparisons if not properly handled.

To evaluate whether `Type` is associated with `share`, we can preliminarily examine whether the **mean or median share count varies across the post types**. Since "Photo" is the most frequent type, any observed **higher engagement (in terms of shares)** could reflect the fact that more visual content tends to attract more user interaction. However, this descriptive indication should be followed by a formal **statistical comparison** (e.g., boxplots or ANOVA) to verify whether these differences are statistically significant and not due to chance.

In conclusion, based on the initial descriptive statistics and CV calculation, it appears that:

- `share` shows **considerable variability**;
- `Type` is **categorical with 3 levels**, but highly **unbalanced**;
- the **type of post may plausibly influence** the number of shares observed, and this hypothesis deserves further formal analysis.

7.2 Fit a multiple linear regression model to explain `share` as a linear function of the other variables. Comment on all the results provided by the summary function and in particular on the estimated regression coefficients, especially those related to the categorical covariates, on their standard errors and on their observed p-values. We fit a **multiple linear regression model** to explain the number of Facebook shares (`share`) as a linear function of all available covariates in `fbshare`. The model is estimated via:

```
mod7 <- lm(share ~ ., data = fbshare)
summary(mod7)
```

```
##
## Call:
## lm(formula = share ~ ., data = fbshare)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.1842  -6.7408  -0.4611   6.7754  20.2002
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      11.7555     2.0811   5.649 3.25e-08 ***
## TypePhoto/Video   5.8080     2.0354   2.853  0.00457 **
## TypeStatus        7.9798     2.5754   3.098  0.00210 **
## Post.Hour        -0.3186     0.1045  -3.047  0.00248 **
## Paid1             0.3602     1.0216   0.353  0.72458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.768 on 366 degrees of freedom
## Multiple R-squared:  0.04769,    Adjusted R-squared:  0.03728
## F-statistic: 4.582 on 4 and 366 DF,  p-value: 0.001272
```

From the output of `summary(mod7)`, we see that the estimated model includes the intercept and four predictors: two **dummy variables** corresponding to the factor `Type` (with `"Link"` as baseline), the numerical covariate `Post.Hour`, and the binary indicator `Paid`.

The **estimated regression equation** can be written as:

$$\widehat{\text{share}} = 11.76 + 5.81 \cdot \text{Type}_{\text{Photo}} + 7.98 \cdot \text{Type}_{\text{Status}} - 0.32 \cdot \text{Post.Hour} + 0.36 \cdot \text{Paid}$$

Interpreting the coefficients:

- The **intercept** (11.76) represents the expected number of shares for a **Link** post (baseline of `Type`), posted at hour zero, and **not sponsored** (`Paid = 0`).
- The variable **`TypePhoto/Video` has a positive and significant effect** (5.81; p = 0.0046): controlling for other factors, photo/video posts are shared about **6 times more** than link posts, on average.
- Similarly, **`TypeStatus` increases shares even more** (7.98; p = 0.0021), indicating that status-type content may be especially engaging compared to links.
- **`Post.Hour` has a significant negative effect** (-0.32; p = 0.0025): later posts tend to receive fewer shares. For each additional hour in posting time, the expected share count drops by about 0.32.
- **`Paid` is not significant** (p = 0.72), suggesting that paid promotion has no clear association with the number of shares, at least in this sample and controlling for other covariates.

The **standard errors** for the `Type` coefficients are around 2, which is relatively large compared to the magnitude of the coefficients, but the corresponding **t-values** are strong enough to reject the null hypothesis at the 1% significance level.

Looking at the **residuals**, the five-number summary shows symmetry around zero (Min: -17.2; Median: -0.46; Max: 20.2), and the **residual standard error** is 8.77, which is quite large compared to the mean of the response. This reflects the high **variability** in `share` not captured by the linear model.

The **coefficient of determination** is really low:

- **Multiple** $R^2 = 0.048$, meaning that only about **5% of the variability in `share` is explained** by the model.

- The **adjusted** $R^2 = 0.037$ confirms that this is not just due to overfitting.

Nevertheless, the **overall F-test** is significant ($p = 0.0013$), so **at least one coefficient differs significantly from zero**. This implies that the model has **some explanatory power**, even if limited.

In conclusion, the **content type** and **posting hour** significantly influence the number of shares, while **paid advertisement does not**. However, the model explains only a small proportion of the total variability in shares, suggesting that additional factors not included in the model may be important in driving engagement.

7.3 Calculate the studentized residuals and report and comment on the plot of the empirical distribution of residuals with the theoretical one. We calculate the **studentized residuals** from the linear model `mod7` using the `rstudent()` function, which standardizes the residuals by their estimated standard deviation after **excluding the observation itself**:

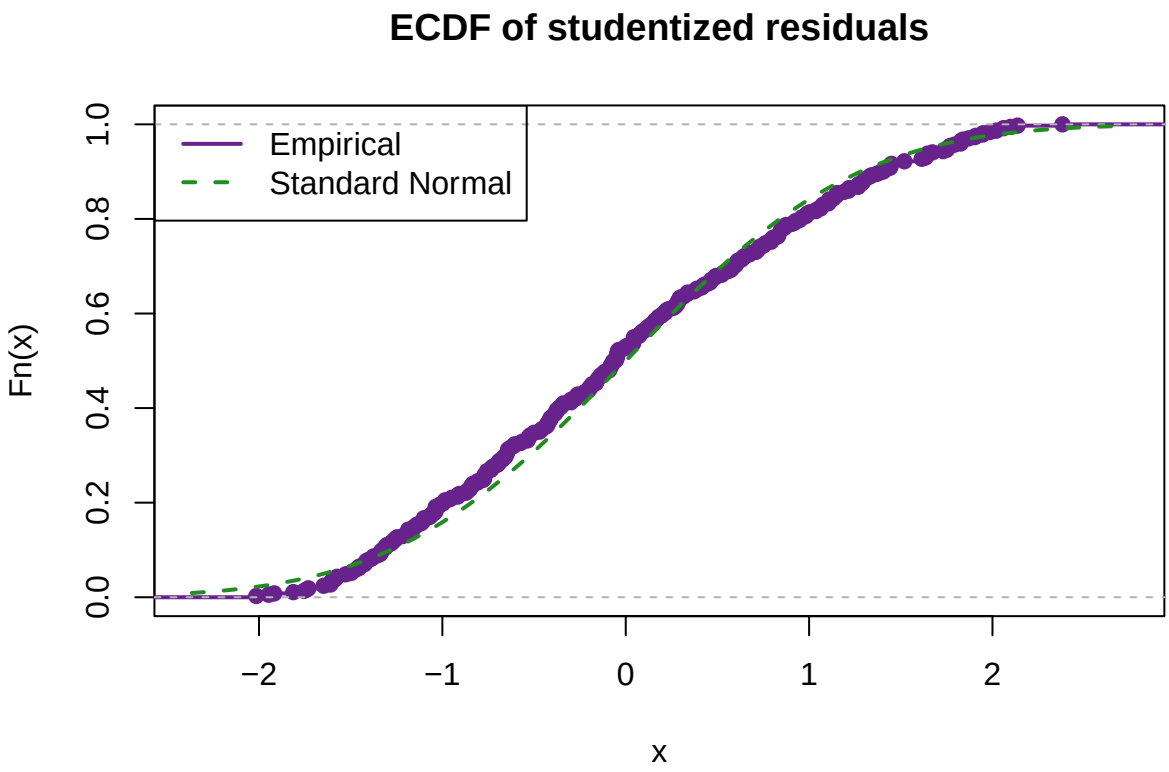
```
stu_res7 <- rstudent(mod7)
round(summary(stu_res7), 3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.015  -0.771  -0.054   0.000   0.780   2.382
```

These values indicate that the **residuals are approximately centered around zero** (mean = 0, median approximately 0), and fairly symmetric (1st and 3rd quartile are almost equidistant from the center). However, the **maximum value** (2.382) is larger in magnitude than the minimum (-2.015), which suggests **slightly heavier tails on the right side**.

The shape of this empirical distribution will be assessed in the next step, where we **compare the empirical cumulative distribution function (ECDF)** of the studentized residuals to the **theoretical standard normal distribution**, which is often used as a reference under the linear model assumptions:

```
plot(ecdf(stu_res7),
     col = "darkorchid4",
     lwd = 2,
     main = "ECDF of studentized residuals")
curve(pnorm(x), col = "forestgreen", lty = 2, lwd = 2, add = TRUE)
legend("topleft", legend = c("Empirical", "Standard Normal"),
       col = c("darkorchid4", "forestgreen"),
       lwd = 2, lty = c(1, 2))
```



The **ECDF of studentized residuals** rises smoothly and closely follows the **standard normal CDF**, confirming the approximate normality of the residuals. However, the **slight divergence in the upper tail** (as suggested by the larger maximum) indicates **mild right-skewness** or **heavier right tail**. This is **not unusual**, since studentized residuals follow a **Student's t distribution**, which has slightly **heavier tails** than the normal distribution—especially with smaller sample sizes or moderate leverage.

Thus, while the residuals are **mostly well-behaved**, this **mild deviation in the upper tail** suggests some influential points may exist, but not enough to seriously violate the model assumptions.

The distribution of studentized residuals appears symmetric and centered, with only slight deviations in the upper tail. The comparison with the normal CDF confirms the **validity of the linear model assumption of normality**, at least approximately, and justifies the use of standard inferential tools (e.g., t-tests, confidence intervals) for this model.

7.4 Considering the following values of the covariates: Type (status), Post.Hour (4) and Paid (0), calculate the point estimate of the predicted support cost for a new unit (out of sample prediction) and the prediction interval with a confidence level of 0.99. Comment on the results. We aim to compute a **point prediction and a prediction interval** at 99% confidence level for a **new observation** of `share`, given the following covariates:

- `Type = Status`
- `Post.Hour = 4`
- `Paid = 0` (no advertisement)

To ensure compatibility with the model structure (especially for factors), the new observation is correctly formatted using:

```r
new_ob <- data.frame(
  Type = factor("Status", levels = levels(fbshare$Type)),
  Post.Hour = 4,
  Paid = factor("0", levels = levels(fbshare$Paid))
)
```

The predicted value is computed with the `predict()` function using `interval = "prediction"` and `level = 0.99`.

```r
predict(mod7,
        newdata = new_ob,
        interval = "prediction",
        level = 0.99)
```

```
##        fit       lwr      upr
## 1 18.46109 -4.664685 41.58687
```

- **Point estimate**: the expected number of shares is 18.46 for a post of type *status*, posted at hour 4, without paid advertisement.
- **Prediction interval**: with 99% confidence, the number of shares for a new post under these conditions is expected to lie between -4.66 and 41.59.
- The interval includes **negative values**, which are **not interpretable in this context** (you can't have negative shares). This reflects the **limitations of linear models** when applied to **count data** without constraints.
- The **width of the interval** (46 units) suggests **substantial uncertainty** in predicting individual outcomes, due to:
  - moderate variability in the data (as previously shown),
  - relatively **low R-squared** of the model (0.048),
  - and the fact that we're estimating a **new observation**, which naturally adds **individual noise**.

The model predicts around 18 shares for this post, but the **wide prediction interval** shows that the outcome could vary greatly, from very low to very high. Although the point estimate may be informative on average, the model is **not highly precise** for individual-level forecasting, and further improvements or alternative models (e.g. Poisson regression) could be considered.

7.5 Why is maximum likelihood estimation important? What does it mean? Which are the properties of maximum likelihood estimators? Maximum Likelihood Estimation (MLE) is one of the most fundamental and widely used methods for statistical inference. The idea is to find the parameter value(s) that **maximize the likelihood function**, i.e., the probability of observing the given sample data under a specified statistical model.

Formally, given a sample $X_1, X_2, \ldots, X_n$ from a distribution with density (or probability mass) function $f(x;\theta)$, the **likelihood function** is:

$$L(\theta) = \prod_{i=1}^{n} f(x_i;\theta)$$

and the **maximum likelihood estimator** (MLE) $\hat{\theta}_{\text{MLE}}$ is defined as:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} L(\theta)$$

In practice, it is often more convenient to work with the **log-likelihood** function:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{n} \log f(x_i; \theta)$$

Maximizing the likelihood means choosing the parameter that makes the observed data **most probable**, according to the model.

MLE is important for several reasons:

- It provides **a unified framework** for parameter estimation under a wide variety of statistical models (discrete, continuous, univariate, multivariate, etc.).
- It leads naturally to other inferential procedures, such as **likelihood ratio tests**, **confidence intervals**, and **information criteria** (e.g., AIC, BIC).
- It is **asymptotically optimal** under regularity conditions: MLEs are consistent, efficient, and normally distributed in large samples.

Under suitable regularity conditions, MLEs possess the following key properties:

1. **Consistency**: The MLE converges in probability to the true value of the parameter as the sample size increases:

$$\hat{\theta}_{\text{MLE}} \xrightarrow{P} \theta_0$$

2. **Asymptotic normality**: For large n, the distribution of the MLE is approximately normal:

$$\sqrt{n}(\hat{\theta}_{\text{MLE}} - \theta_0) \xrightarrow{d} \mathcal{N}\left(0, \mathcal{I}^{-1}(\theta_0)\right)$$

   where $\mathcal{I}(\theta_0)$ is the **Fisher information**.

3. **Asymptotic efficiency**: Among all consistent estimators, the MLE achieves the **lowest possible variance** asymptotically, reaching the **Cramér–Rao lower bound**.

4. **Invariance**: If $\hat{\theta}$ is the MLE of θ, and $g(\theta)$ is a continuous function, then $g(\hat{\theta})$ is the MLE of $g(\theta)$.

Maximum Likelihood Estimation is a powerful and general method for parameter estimation. It is grounded in a clear probabilistic rationale and enjoys **excellent asymptotic properties**, which make it the **default choice** in many statistical applications. Despite its sensitivity to model assumptions and computational complexity in some cases, it remains a cornerstone of modern statistical inference.

Exercise 8

The data contained in the file `air1.Rdata` refer to characteristics of aircraft designs which appeared during the twentieth century. Reported measurements are related to `Span` (wing span in meters), `Length` (length, in meters) and `Speed` (maximum speed, km/h).

8.1 Provide a summary of the observed data. The dataset `air1.Rdata` includes 300 observations and 3 numerical variables:

- `Span`: wing span (in meters),
- `Length`: body length (in meters),
- `Speed`: maximum speed (in km/h).

```
load("air1.Rdata")
skim_without_charts(air1)
```

Table 13: Data summary

Name	air1
Number of rows	300
Number of columns	3
Column type frequency:	
numeric	3
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Span	0	1	18.13	10.39	6.68	11.10	14.35	22.02	70.1
Length	0	1	15.75	10.54	5.69	8.89	12.00	18.90	70.5
Speed	0	1	585.29	551.23	106.00	257.75	384.50	653.25	3219.0

```
cv8 <- function(x) sd(x) / mean(x)
round(apply(air1, 2, cv8), 2)
```

```
##   Span Length  Speed
##   0.57   0.67   0.94
```

Sample size: 300 aircraft designs. All variables are numeric, with no missing data. The **coefficient of variation** (CV) reveals:

- `Span`: CV = 0.57 (moderate variability)
- `Length`: CV = 0.67 (slightly higher variability)
- `Speed`: CV = 0.94 (very high relative variability, indicating a highly dispersed distribution)

Boxplots, histograms and kernel density plots were drawn using the following code:

```
par(mfrow = c(3, 3), mar = c(4, 4, 2, 1))

# Boxplots
boxplot(air1$Span, main = "Span", col = "lightblue", horizontal = TRUE)
boxplot(air1$Length, main = "Length", col = "lightgreen", horizontal = TRUE)
boxplot(air1$Speed, main = "Speed", col = "lightpink", horizontal = TRUE)

# Histograms
hist(air1$Span, main = "Histogram of Span", col = "lightblue", horizontal = TRUE)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "horizontal" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "horizontal" is not a graphical parameter

## Warning in axis(1, ...): "horizontal" is not a graphical parameter

## Warning in axis(2, at = yt, ...): "horizontal" is not a graphical parameter
```

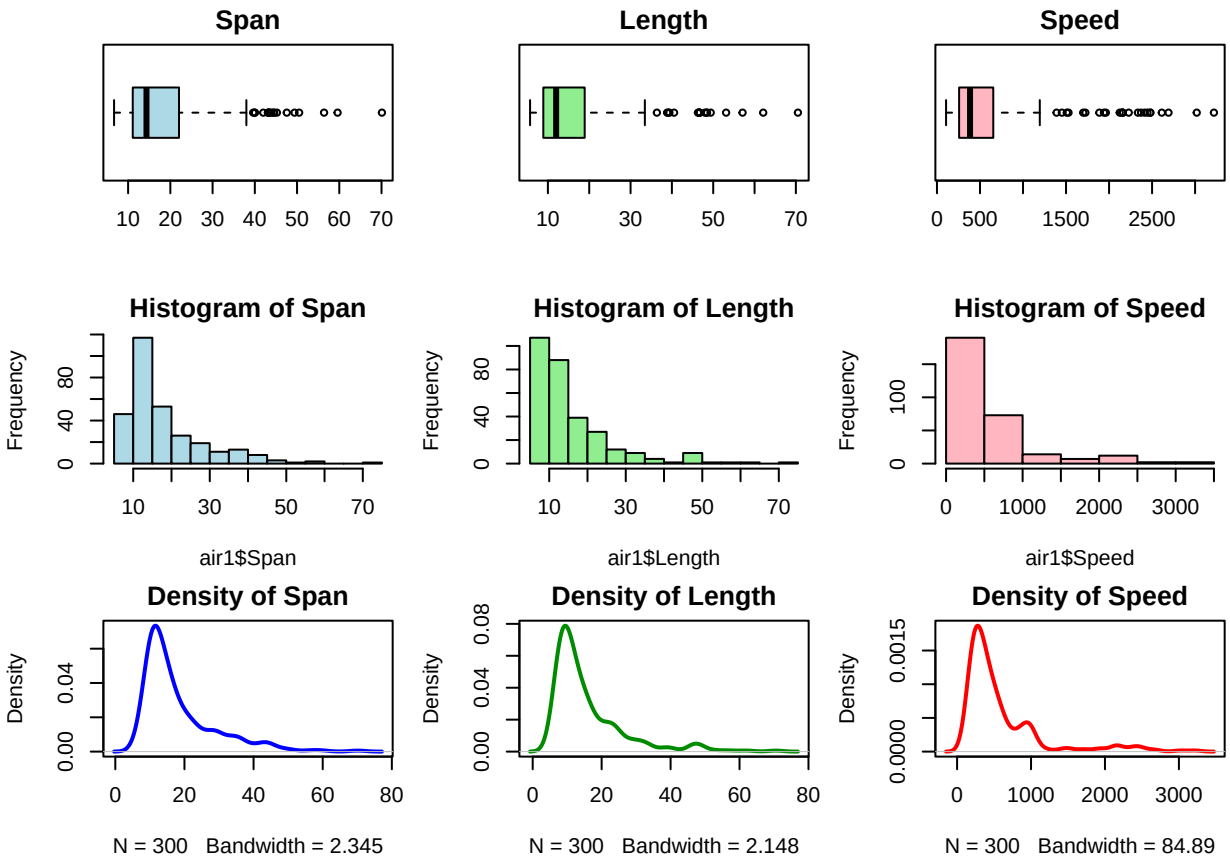
```
hist(air1$Length, main = "Histogram of Length", col = "lightgreen", horizontal = TRUE)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "horizontal" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "horizontal" is not a graphical parameter

## Warning in axis(1, ...): "horizontal" is not a graphical parameter

## Warning in axis(2, at = yt, ...): "horizontal" is not a graphical parameter
```

```
hist(air1$Speed, main = "Histogram of Speed", col = "lightpink", horizontal = TRUE)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "horizontal" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "horizontal" is not a graphical parameter

## Warning in axis(1, ...): "horizontal" is not a graphical parameter

## Warning in axis(2, at = yt, ...): "horizontal" is not a graphical parameter
```

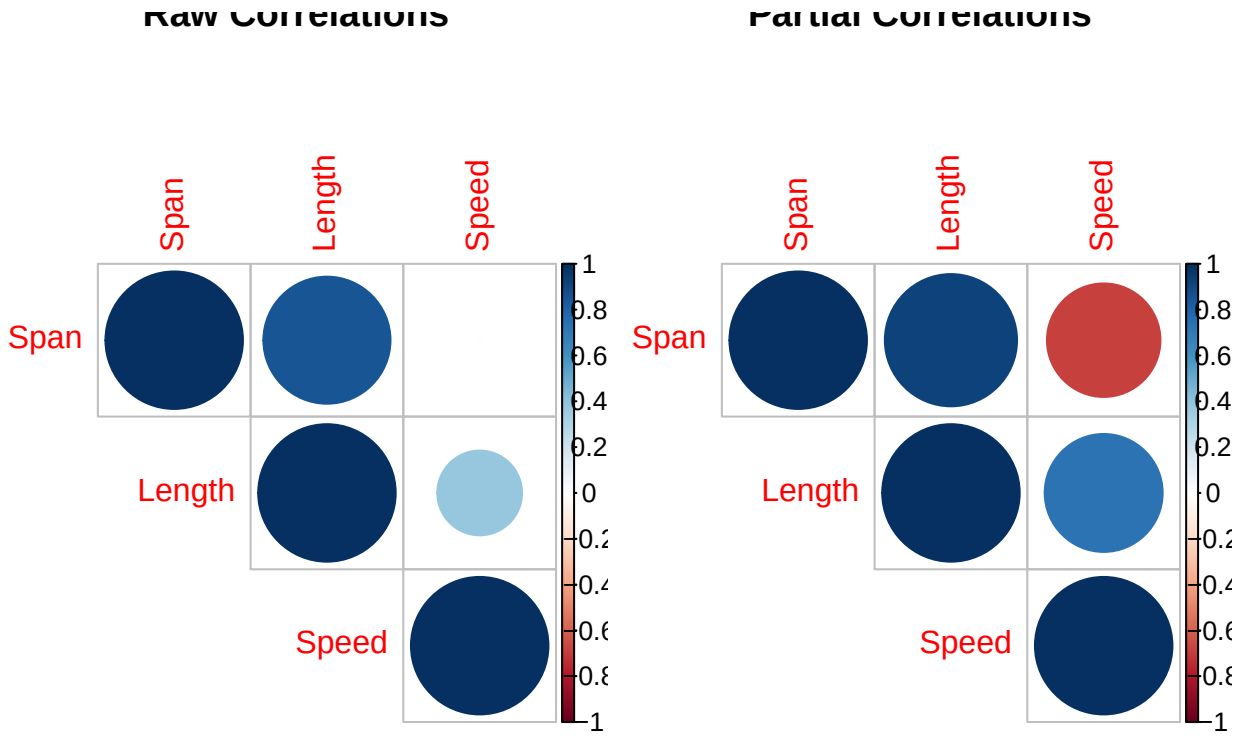
```
# Density plots
plot(density(air1$Span), main = "Density of Span", col = "blue", lwd = 2)
plot(density(air1$Length), main = "Density of Length", col = "green4", lwd = 2)
plot(density(air1$Speed), main = "Density of Speed", col = "red", lwd = 2)
```



- All variables exhibit **asymmetry** and clear presence of **outliers**.
- All the variables are **right-skewed**, with some extreme values. This may indicate a population made almost entirely by small aircrafts.
- `Speed` has a **heavily right-skewed** distribution with a maximum of over 3200 km/h, suggesting the presence of **military jets or supersonic aircrafts**.

Raw and partial correlations are computed and visualized.

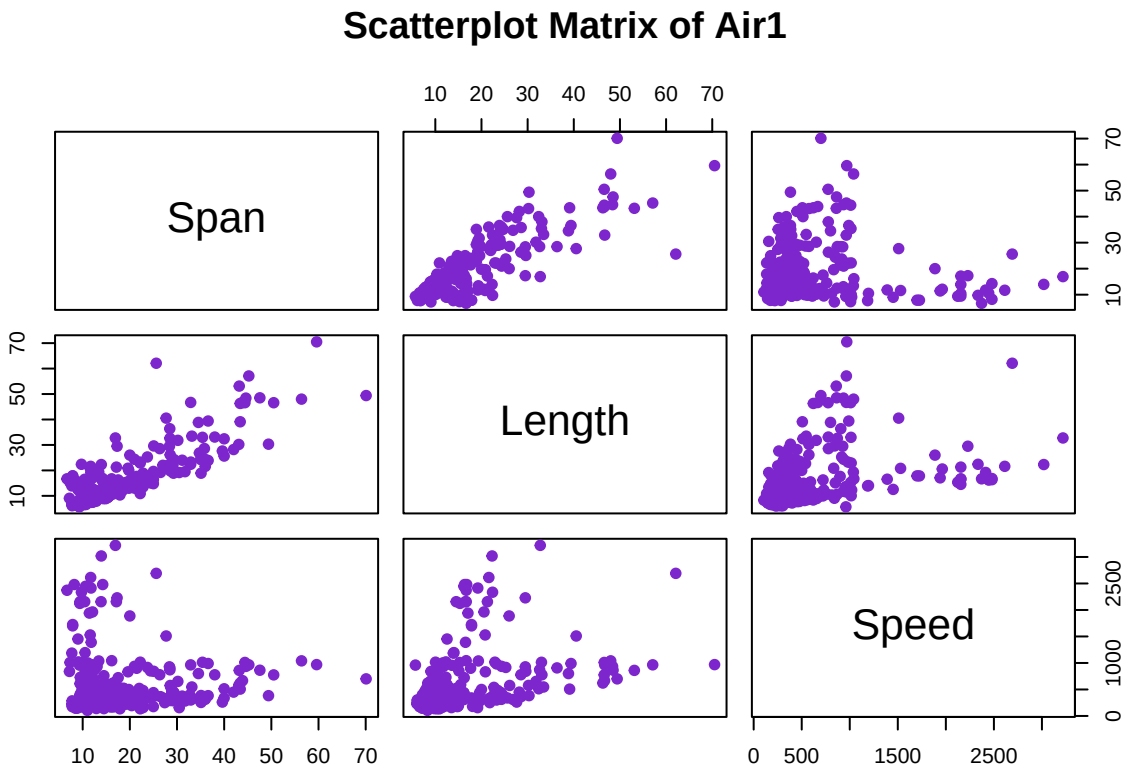
```
C_raw8 <- cor(air1)
S8 <- cov(air1)
C_partial8 <- parcor(S8)

par(mfrow = c(1, 2))
corrplot(C_raw8, type = "upper", title = "Raw Correlations")
corrplot(C_partial8, type = "upper", title = "Partial Correlations")
```



- **Raw correlations**:
  - `Span` and `Length` are **strongly positively correlated** ($\rho \approx 0.85$): longer aircrafts tend to have larger wings.
  - `Length` and `Speed`: moderate positive correlation.
  - `Span` and `Speed`: very low correlation, almost 0.
- **Partial correlations**:

After controlling for the third variable, the positive link between `Length` and `Speed` persists, while the **(Span, Speed)** partial correlation becomes **negative**, suggesting **confounding** in the raw correlation.

```
pairs(air1,
      main = "Scatterplot Matrix of Air1",
      pch = 19,
      col = "purple3")
```



- Clear **linear relationship** between `Span` and `Length`.
- A **nonlinear and dispersed pattern** for `Speed`, especially with respect to `Span`.

```
mod8 <- Mclust(air1, G = 3, modelNames = "EII")
summary(mod8, parameters = TRUE)
```

8.2 Perform model-based clustering estimating a Gaussian finite mixture model with three components and a spherical covariance structure. Describe the type of model are you estimating.

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -5632.465 300 12 -11333.37 -11339.56
##
## Clustering table:
##   1   2   3
##  53  21 226
```

```
##
## Mixing probabilities:
##          1          2          3
## 0.17811174 0.06999983 0.75188843
##
## Means:
##             [,1]       [,2]      [,3]
## Span    23.87893   12.68382  17.27382
## Length  25.49754   21.88382  12.86821
## Speed  941.48543 2298.19188 341.44784
##
## Variances:
## [,,1]
##            Span   Length    Speed
## Span   10113.07     0.00     0.00
## Length     0.00 10113.07     0.00
## Speed      0.00     0.00 10113.07
## [,,2]
##            Span   Length    Speed
## Span   10113.07     0.00     0.00
## Length     0.00 10113.07     0.00
## Speed      0.00     0.00 10113.07
## [,,3]
##            Span   Length    Speed
## Span   10113.07     0.00     0.00
## Length     0.00 10113.07     0.00
## Speed      0.00     0.00 10113.07
```

The model estimated is a **Gaussian finite mixture model** with the following structure:

- **G = 3**: the model includes **three Gaussian components**, representing three underlying subpopulations in the aircraft data.
- **Covariance model "EII"**: this assumes that all clusters share:
  - a **spherical** covariance matrix (same variance in all directions),
  - with **equal volume** across components.
  - In this case, the variance matrix is diagonal and constant across all groups:

$$\Sigma_g = \sigma^2 \mathbf{I}, \quad \text{for all } g = 1, 2, 3$$

where $\sigma^2 = 10113.07$.

8.3 Illustrate and comment on the estimates provided by the summary function, particularly focusing on how many groups have been identified in the applied context. Most observations (226 out of 300) are assigned to **component 3**, which indicates that this group represents the dominant structure in the dataset, capturing the most **homogeneous and frequent aircraft profile**.

Interpretation of the components (based on centroid values):

- **Component 2** features an exceptionally high **mean speed** (2298 km/h), which is consistent with **supersonic or high-performance military aircraft**. Despite having moderate size (length 22 m), the extreme speed clearly distinguishes this cluster.

- **Component 1** corresponds to **medium-speed aircraft** (941 km/h), with **larger dimensions** (span 24 m, length 25 m). These characteristics align well with **commercial airliners or long-range passenger planes**.
- **Component 3**, which includes the vast majority of units, is characterized by **lower speed** (341 km/h) and **smaller size** (length 13 m). This group likely includes **general aviation planes**, such as trainers, private aircraft, or small transport planes.

Covariance structure:

The model assumes a **spherical and homoscedastic covariance structure (EII)**, where:

- All clusters share the **same variance value**,

$$\sigma^2 = 10113.07,$$

  for all variables;
- The covariance matrix in each group is

$$\Sigma_g = \sigma^2 I \quad \text{for all } g \in \{1, 2, 3\}.$$

This means that:

- **No correlation** between variables is modeled;
- **Equal variability** is assumed across all dimensions and all clusters.

While this structure simplifies the estimation process, it may **underestimate the true complexity of the data**, particularly if variables exhibit heterogeneous variances or correlations within groups.

Model fit metrics:

The **log-likelihood**, **BIC**, and **ICL** values reported are:

- Log-likelihood: –5632.465
- BIC: –11333.37
- ICL: –11339.56

These criteria assess model adequacy and are especially useful for **comparing multiple models** with different numbers of components or covariance structures. Since only one model (G = 3, EII) has been estimated here, these values are not sufficient to assess optimality on their own, but they confirm that a plausible and statistically valid model has been fitted.

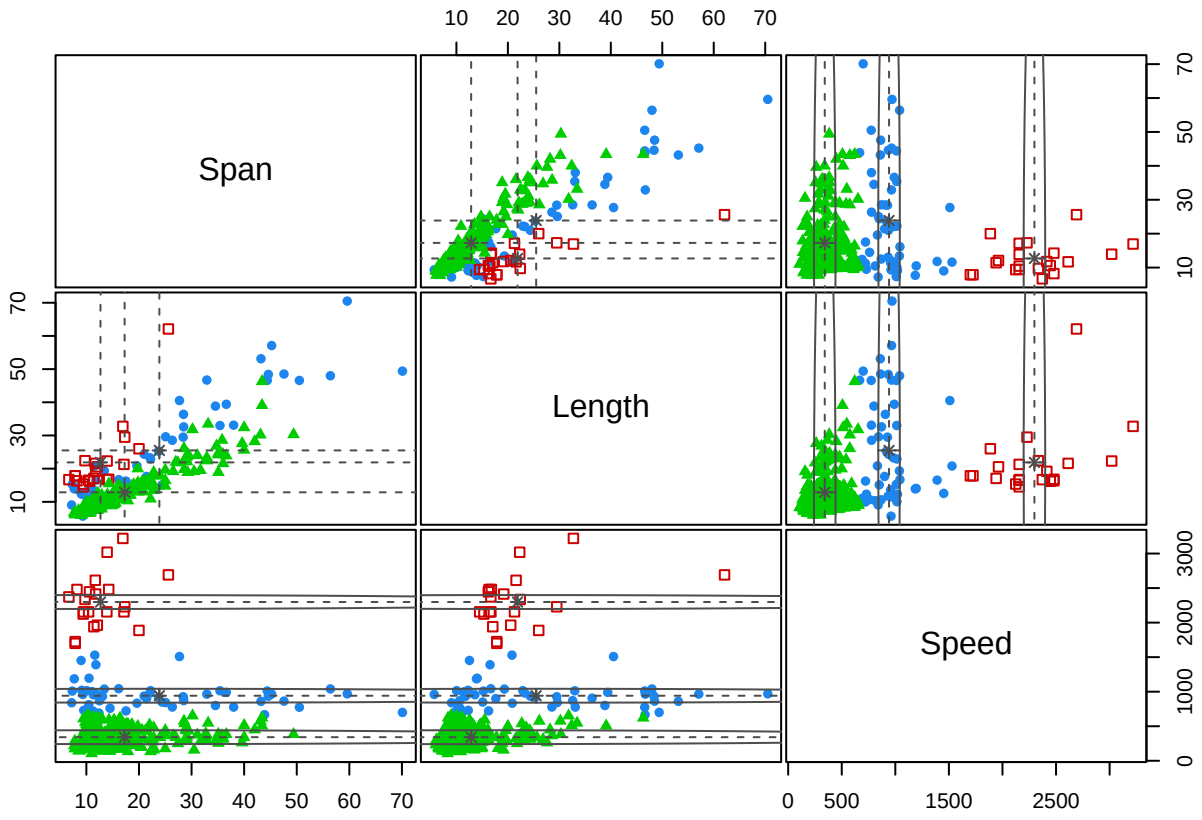
8.4 Plot the classification chart and describe what can be observed and which are implications of such results. The **classification plot** generated via `plot(mod8, what = "classification")` displays a **pairwise scatterplot matrix** of the three variables (`Span`, `Length`, `Speed`), with points **colored and shaped** according to the assigned cluster.

Each point is classified into one of the three components:

- **Component 1**: red squares
- **Component 2**: green triangles
- **Component 3**: blue circles

Black stars indicate the **cluster centroids**.

```r
plot(mod8, what = "classification")
```



What we observe:

- **Component 3 (blue)** clearly dominates the dataset. It contains most aircraft and appears compact and centered in a region with **small-to-medium span and length**, and **moderate speed**.
- **Component 2 (green)** is concentrated in a **narrow vertical band** in the `Speed` variable: it captures **high-speed aircraft**, regardless of their size. The vertical clustering confirms that **speed is the discriminating factor**, while span and length vary.
- **Component 1 (red)** contains aircraft with **larger physical dimensions** (span and length) but **moderate-to-high speed**. This cluster spreads more across the `Span`–`Length` plane and might include **commercial or heavy-duty aircraft**.

The **ellipses** (though spherical in this model) provide a visual approximation of the **within-cluster dispersion**, consistent with the assumed EII covariance structure.

Implications:

1. The model successfully identifies **structurally distinct subpopulations** of aircraft in the design space. Each cluster corresponds to a different **design philosophy or functional class**:
    - Small and slow aircraft (training, civil aviation),
    - Large and commercial,
    - Fast and specialized (military/supersonic).

2. The fact that **clusters are well separated along the `Speed` axis** highlights its central role in discriminating aircraft types—this aligns with the interpretation of the **component means** in the previous step.

3. However, because the **EII model forces spherical and equal-volume clusters**, some **ellipses may not fit the actual shape of the data**. A more flexible model (e.g., VVV) might capture elongated or anisotropic distributions more realistically.

8.5 Apply nonparametric bootstrap to obtain a 95% confidence interval for the estimated parameters of the mixing probabilities. Report and comment on the lower and upper bound of the estimated weight of cluster 3. Report also the histogram of the bootstrap distribution for this parameter. Describe the results. To assess the uncertainty around the estimated **mixing probability** of component 3 (cluster 3) in the finite Gaussian mixture model, we performed a **nonparametric bootstrap** with 1000 replications. The goal is to construct a **95% confidence interval** for this parameter.

```
library(mclust)

set.seed(123)
B <- 1000
n <- nrow(air1)
boot_weights <- numeric(B)

for (i in 1:B) {
  idx <- sample(1:n, replace = TRUE)
  air1_star <- air1[idx, ]
  mod_star <- Mclust(air1_star, G = 3, modelNames = "EII", verbose = FALSE)
  boot_weights[i] <- mod_star$parameters$pro[3]
}

ci_bounds <- quantile(boot_weights, probs = c(0.025, 0.975))
round(ci_bounds, 4)
```

```
##   2.5%  97.5%
## 0.0500 0.7884
```

The **95% bootstrap confidence interval** for the **mixing probability of cluster 3** is:

$$[0.0500, 0.7884]$$

This interval reflects substantial **uncertainty** about the true weight of cluster 3 in the population.

- The **lower bound** of 0.0500 suggests that in some bootstrap resamples, component 3 might capture as little as **5% of the population**.
- The **upper bound** of 0.7884 supports the original estimate of about 75%, indicating that in many replications cluster 3 remains dominant.

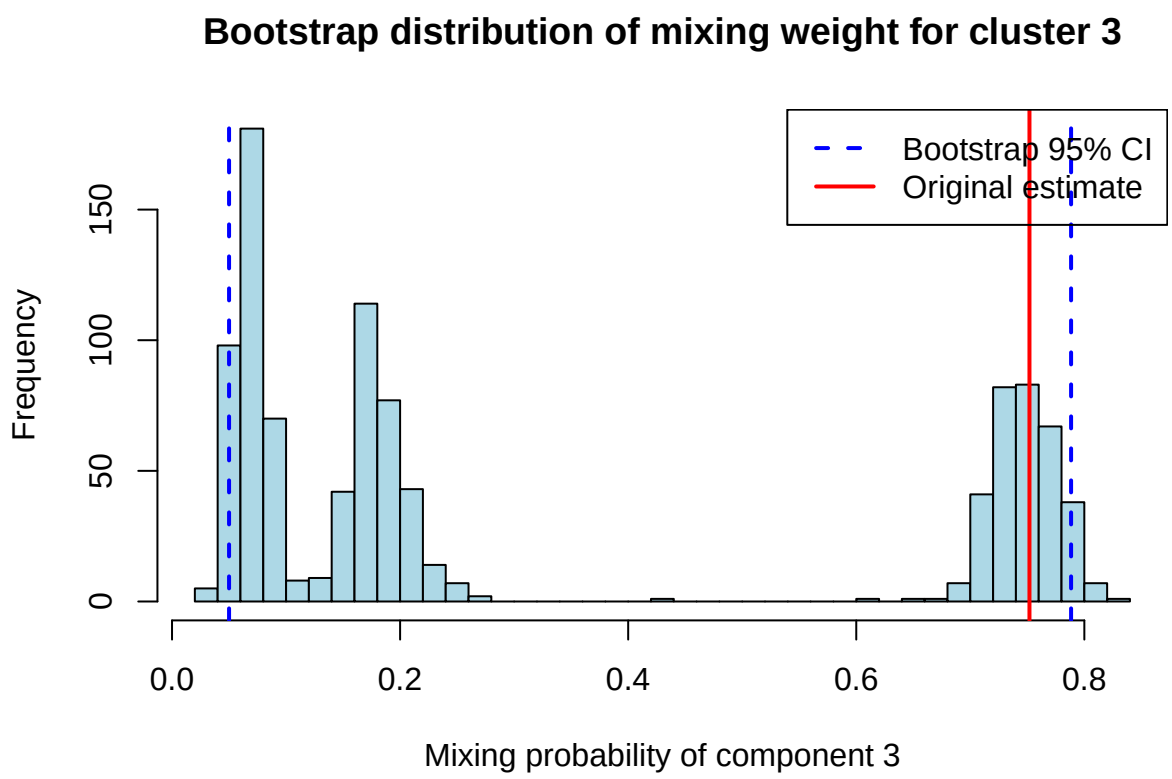
Such a wide interval is not necessarily due to poor estimation, but rather to the **label switching problem** typical in mixture models: the component labeled "3" may not consistently refer to the same cluster across bootstrap samples.

As a result, the interpretation of this interval must be handled carefully. While cluster 3 appears dominant in the original model, the bootstrap results show that this dominance is not stable across samples without label alignment. The distribution of mixing weights is **bimodal**, which further confirms this instability.

In summary, this confidence interval highlights the **variability** in component assignment in unsupervised classification, and shows that although component 3 is often the largest group, in some bootstrap samples it may represent a **minor** one.

We then visualized the bootstrap distribution with the following code:

```
hist(boot_weights,
     col = "lightblue",
     breaks = 30,
     main = "Bootstrap distribution of mixing weight for cluster 3",
     xlab = "Mixing probability of component 3")

abline(v = ci_bounds, col = "blue", lwd = 2, lty = 2)
abline(v = mod8$parameters$pro[3], col = "red", lwd = 2)

legend("topright",
       legend = c("Bootstrap 95% CI", "Original estimate"),
       col = c("blue", "red"),
       lty = c(2, 1),
       lwd = 2)
```



The original estimate (shown in red) is approximately 0.75, suggesting that most aircraft are assigned to cluster 3. However, the bootstrap histogram reveals a **bimodal pattern**, with one mode near the original estimate and another near zero. This occurs due to **label switching**, a common issue in mixture models: the component labeled "3" in the bootstrap samples may not correspond to the same group as in the original model.

This explains the wide confidence interval: while the model consistently detects a dominant group, component labels in the bootstrap runs may not always align, making the estimate of π_3 unstable.

In conclusion, the original estimate suggests that cluster 3 includes around 75% of the observations. However, the bootstrap distribution shows **high variability**, and the 95% confidence interval is broad. This underscores the need for **careful interpretation** in mixture models, especially under label ambiguity or when components are not well-separated.