Week1, Assignment 1

Marwah Faraj

```
knitr::opts_chunk$set(
  echo = TRUE, warning = FALSE, message = FALSE,
  fig.width = 6, fig.height = 3.5  # global size shrink
)
```

```
library(tidyverse)
library(GGally)
```

Problem 3.1 (30 points)

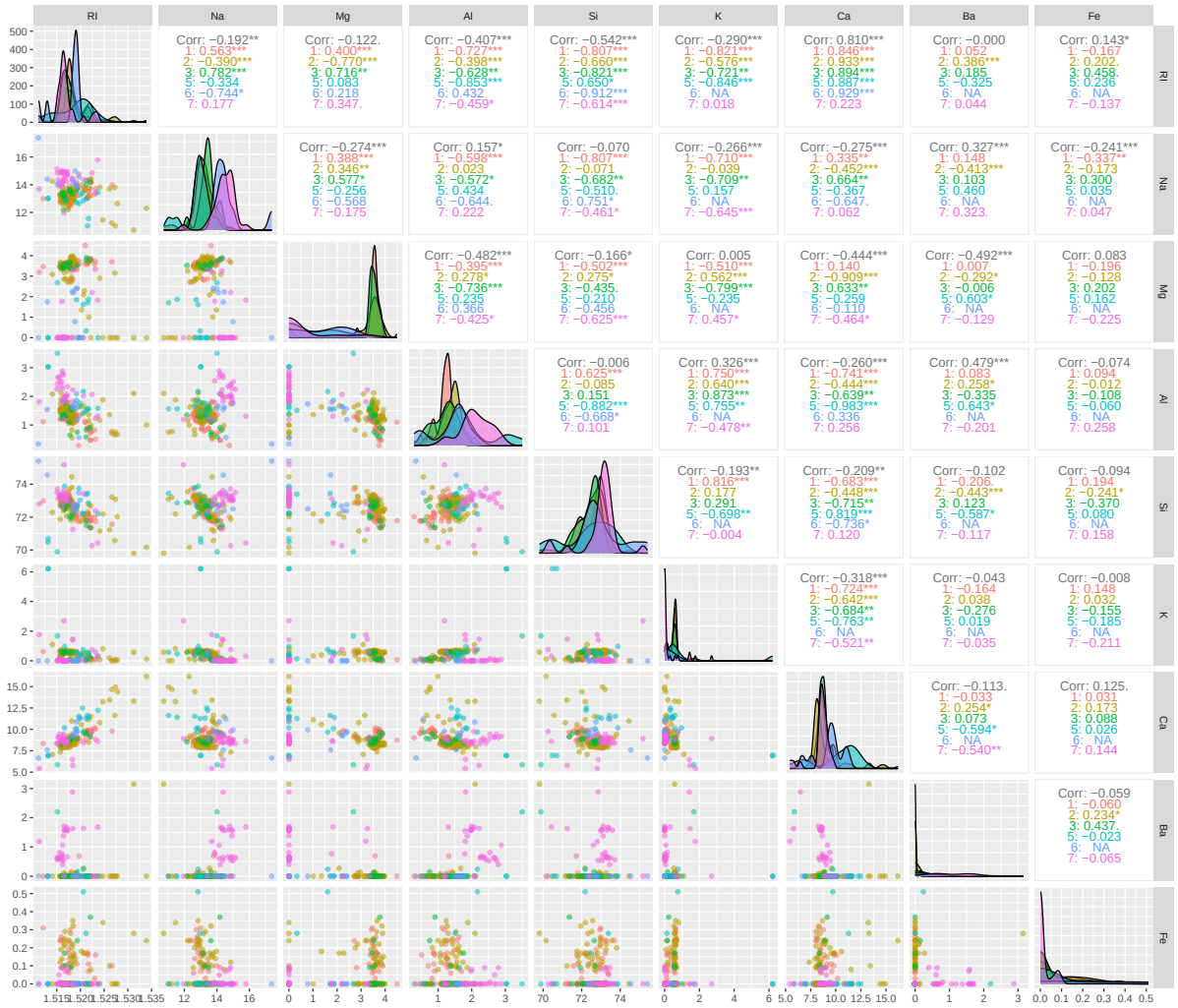
The UC Irvine Machine Learning Repository contains a Glass Identification Data Set. The data consist of 214 glass samples labeled as one of seven class categories. There are nine predictors, including the refractive index and percentages of eight elements: Na, Mg, Al, Si, K, Ca, Ba, and Fe. The data can be accessed via:

```
library(mlbench)
data(Glass)
```

3.1.a (10 points)

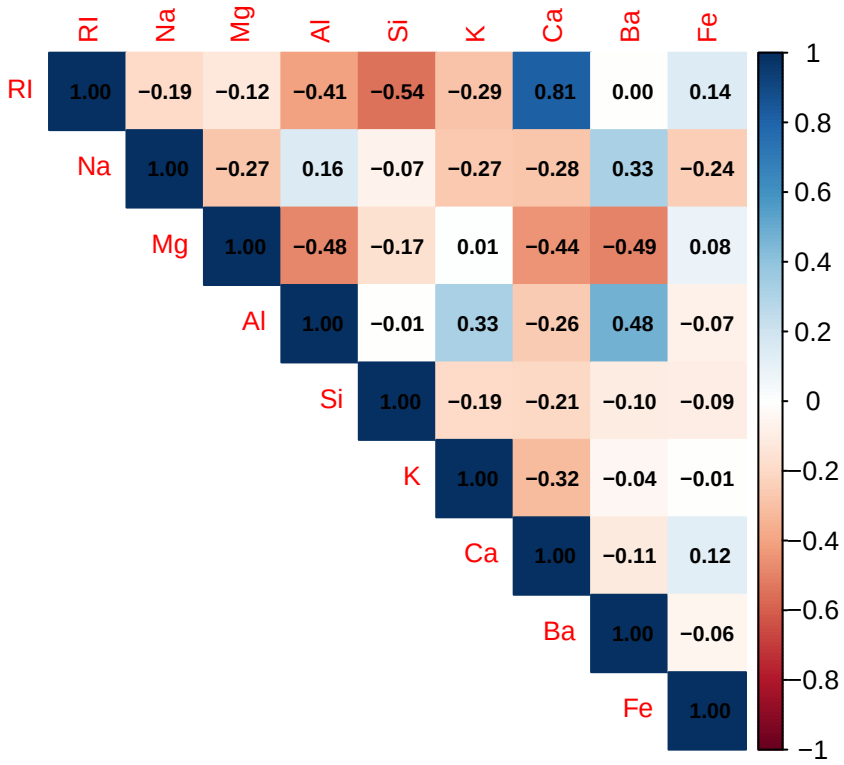
Using visualizations, explore the predictor variables to understand their distributions …

```
# Pairwise plot of predictors colored by Type
GGally::ggpairs(
  Glass,
  columns = 1:9,
  aes(color = factor(Type), alpha = 0.6)
)
```



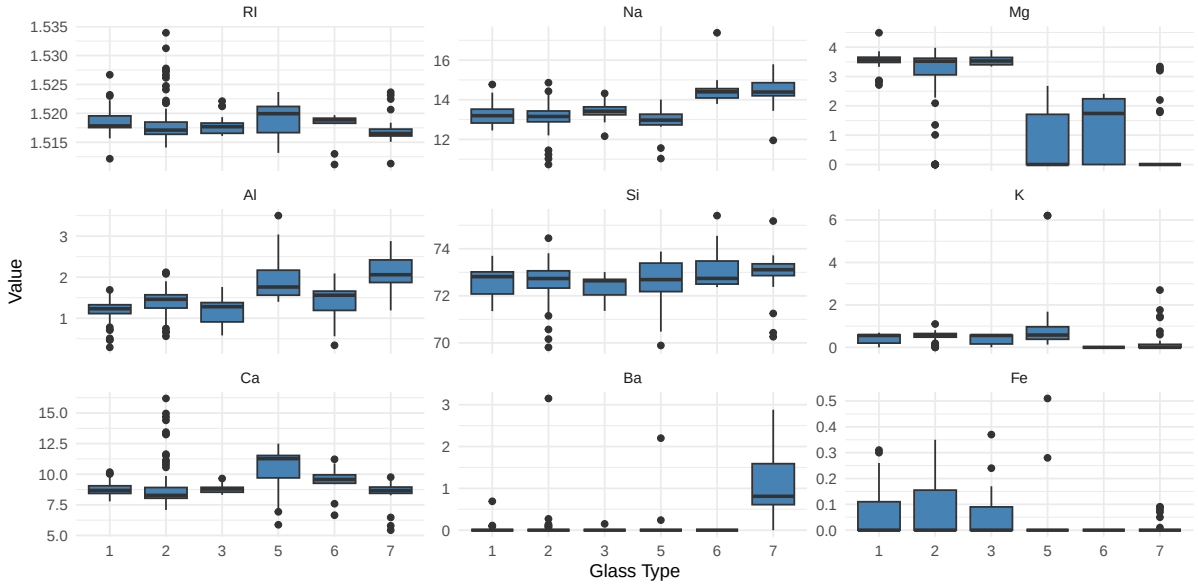
as well as the relationships between predictors.

```
library(corrplot)

# Compute correlation matrix among predictors (excluding Type)
corr_matrix <- cor(Glass[, 1:9])

# Plot heatmap
corrplot(corr_matrix, method = "color", type = "upper",
         tl.cex = 0.8, number.cex = 0.7, addCoef.col = "black")
```



Explore the relationship between predictors and response.

```
library(reshape2)
glass_long <- melt(Glass, id.vars = "Type")

ggplot(glass_long, aes(x = factor(Type), y = value)) +
  geom_boxplot(fill = "steelblue") +
  facet_wrap(~ variable, scales = "free_y", ncol = 3) +  # 3 plots per row to give more room
  labs(x = "Glass Type", y = "Value") +
  theme_minimal()
```



Which elements do you think will be good/poor predictors (based on the visualizations)?

... observations

Based on visualizations:

- **Good predictors** might include `Ba`, `RI`, `Mg`, and `Al`—they show noticeable separation across `Type`.
- **Poor predictors** could be `Fe` or `K`, which have more overlap or low variability across classes

  ....

Compute the correlations between the predictors and the the `Type` variable.

```
#... add code to compute and print correlations ...
# Convert Type to numeric first
glass_num <- Glass %>%
  mutate(Type_num = as.numeric(Type)) %>%
  select(-Type)  # remove original factor column

# Compute correlation of each predictor with Type_num
correlations <- sapply(glass_num[ , 1:9], function(x) cor(x, glass_num$Type_num))
correlations
```

```
          RI           Na           Mg           Al           Si            K
-0.168739357  0.506424080 -0.728159518  0.591197598  0.149690687 -0.025834560
```

```
          Ca           Ba           Fe
-0.008997841  0.577676375 -0.183206747
```

Which elements do you think will be good/poor predictors (based on the correlation calculation)?

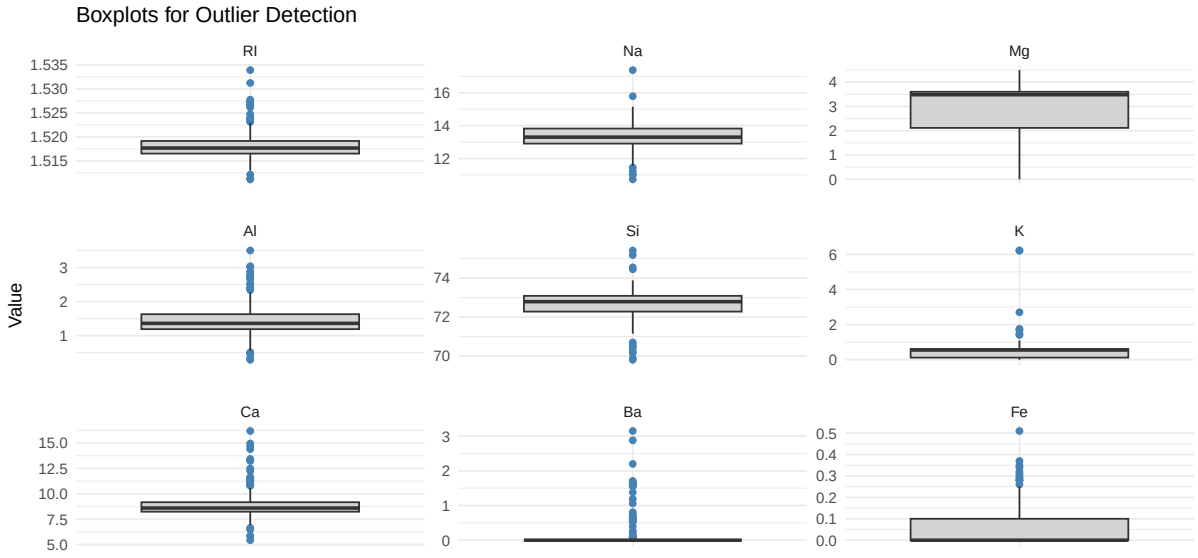
- `Mg` has a strong negative correlation with `Type` (-0.73), suggesting it's a strong predictor.
- `Al` (0.59), `Ba` (0.58), and `Na` (0.51) also show moderate positive correlations with `Type`.
- `Fe`, `RI`, and `Si` show weak correlations (close to 0), and `Ca` and `K` are near zero, suggesting they may be poor predictors.

3.1.b (10 points)

Are there any outliers in the data?

... observations from previous visualizations ...

```
library(reshape2)
glass_long <- melt(Glass, id.vars = "Type")

ggplot(glass_long, aes(x = "", y = value)) +
  geom_boxplot(fill = "lightgray", outlier.color = "steelblue") +
  facet_wrap(~ variable, scales = "free", ncol = 3) +  # fewer columns = more space per facet
  labs(title = "Boxplots for Outlier Detection", y = "Value", x = "") +
  theme_minimal()
```



... observations

- Clear outliers are visible in variables like `Ba`, `Fe`, and `K`, where a small number of values lie far from the bulk of the distribution.
- Variables like `RI`, `Mg`, and `Al` show more consistent spreads with fewer extreme values.
- These outliers could potentially influence model performance and may need transformation or robust scaling.

Are any predictors skewed?

… observations from previous visualizations:

- Predictors with **extremely high positive skewness** include: `K` (6.460), `Ba` (3.369), `Ca` (2.018), `Fe` (1.730), and `RI` (1.603). These are highly skewed and may require transformation.
- `Al` (0.895) and `Si` (−0.720) are **moderately skewed**, with `Si` skewed negatively.
- `Mg` (−1.136) is **strongly negatively skewed**.
- Only `Na` (0.448) is **close to normal**, with skewness below the ±0.5 threshold.

```
# ... calculate skew for each predictor
library(e1071)

# Compute skewness for all 9 predictors
skew_values <- sapply(Glass[, 1:9], skewness)

# Convert to a data frame for display
skew_df <- data.frame(
  Predictor = names(skew_values),
  Skewness = round(skew_values, 3)
)

print(skew_df)
```

```
   Predictor Skewness
RI        RI    1.603
Na        Na    0.448
Mg        Mg   -1.136
Al        Al    0.895
Si        Si   -0.720
K          K    6.460
Ca        Ca    2.018
Ba        Ba    3.369
Fe        Fe    1.730
```

3.1.c (10 points)

Are there any relevant transformations of one or more predictors that might improve the classification model? Assume the model requires the predictors to have approximately symmetric distribution. Apply relevant transformations to the predictors and observe the changes to the distributions of predictors.

Hints:

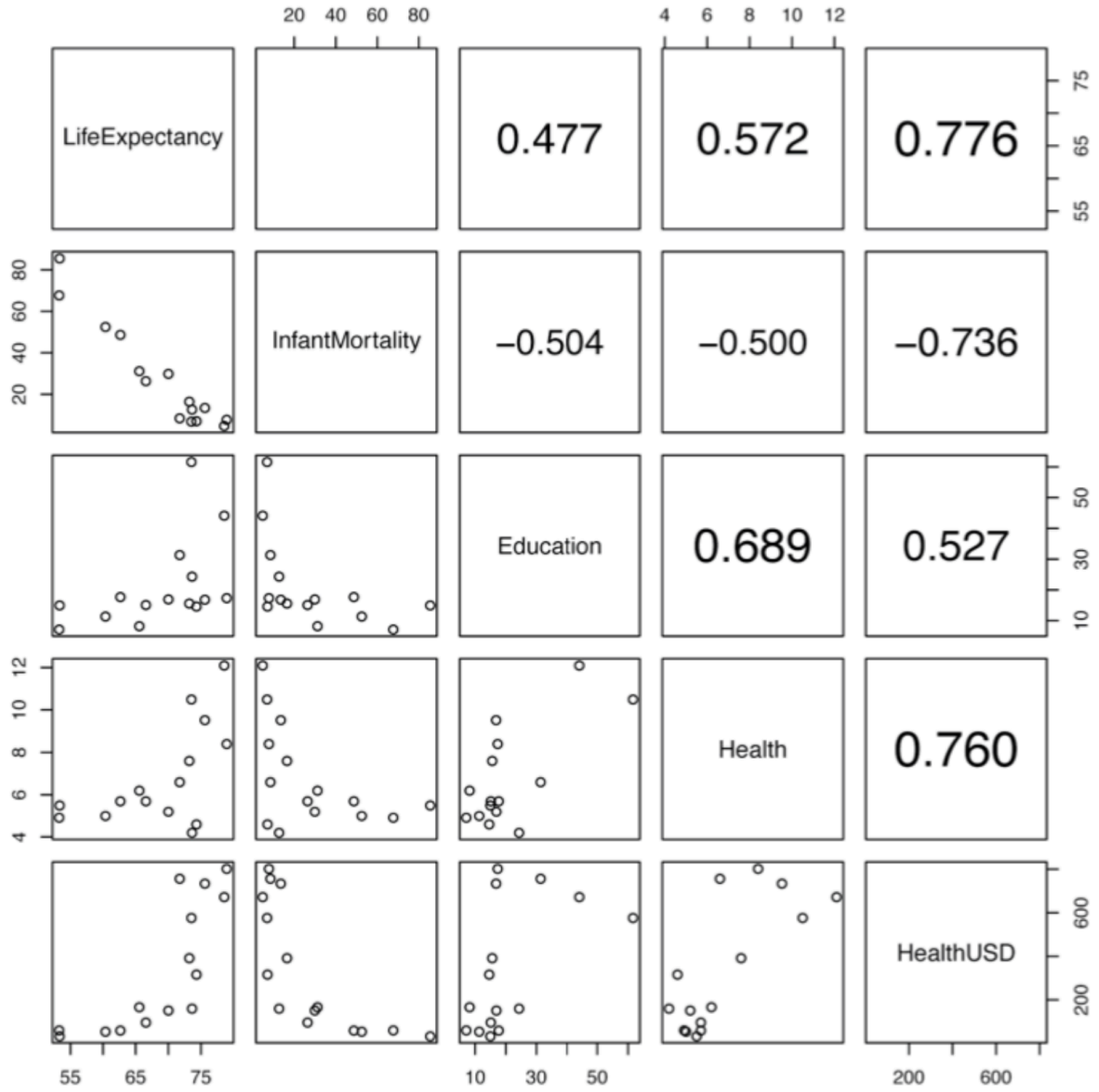
- Skew values less than ± 0.5 should be considered 'normal enough'
- Use 'caret::BoxCoxTrans()' with appropriate adjustments for non-positive values.
- Use transformations that improve skew by >50%

```
# write code to evaluate the before and after skew characteristics, and select variables to t
library(caret)

# Identify skewed predictors (|skew| > 0.5)
skewed_predictors <- names(which(abs(skew_values) > 0.5))

# Apply Box-Cox transformation only to those predictors
glass_transformed <- Glass  # copy original data
for (var in skewed_predictors) {
  x <- Glass[[var]]
  # Add small offset if any values are <= 0 (required for Box-Cox)
  if (any(x <= 0)) x <- x + abs(min(x)) + 0.001
  bc <- BoxCoxTrans(x)
  glass_transformed[[var]] <- predict(bc, x)
}

# Recalculate skewness for transformed data
new_skew <- sapply(glass_transformed[, skewed_predictors], skewness)

# Combine old and new skew for comparison
skew_compare <- data.frame(
  Predictor = skewed_predictors,
  Before_Skew = round(skew_values[skewed_predictors], 3),
  After_Skew = round(new_skew, 3),
  Improvement = paste0(
    round(100 * (abs(skew_values[skewed_predictors]) - abs(new_skew)) /
            abs(skew_values[skewed_predictors]), 1), "%")
)

print(skew_compare)
```

	Predictor	Before_Skew	After_Skew	Improvement
RI	RI	1.603	1.566	2.3%
Mg	Mg	-1.136	-1.299	-14.3%
Al	Al	0.895	0.091	89.8%
Si	Si	-0.720	-0.651	9.6%
K	K	6.460	0.126	98%
Ca	Ca	2.018	-0.194	90.4%
Ba	Ba	3.369	1.676	50.2%
Fe	Fe	1.730	0.738	57.3%

Problem 3.2 (20 points)

The image below shows a scatter plot matrix of the continuous features of a dataset. Discuss the relationships between the features in the dataset that this scatter plot highlights. Make sure to discuss relationships between all pairs.



Hint: There should be 10 combinations [$n(n-1)/2$] . The plot is missing a correlation coefficient for one – estimate what it is.

Example: *Var1 has a strongpositive correlation with Var2 of 0.XXX*

Problem 3.2 – Pairwise Relationships in Scatter Plot Matrix

1. **LifeExpectancy & Education**:
   Moderate positive correlation of **0.572**. Countries with more education tend to have higher life expectancy.

2. **LifeExpectancy & Health**:
   Strong positive correlation of **0.776**. Higher health scores are associated with longer life expectancy.

3. **LifeExpectancy & HealthUSD**:
   Moderate-to-strong positive correlation of **0.760**. Countries that spend more on healthcare tend to have longer life expectancy.

4. **LifeExpectancy & InfantMortality**:
   Moderate negative correlation of **−0.504**. Countries with higher infant mortality tend to have lower life expectancy.

5. **InfantMortality & Education**:
   Moderate negative correlation of **−0.500**. Higher levels of education are associated with lower infant mortality.

6. **InfantMortality & Health**:
   Strong negative correlation of **−0.736**. Better health systems correlate with lower infant mortality.

7. **InfantMortality & HealthUSD**:
   Missing correlation value. Based on the scatter plot, the relationship appears moderately negative.
   Estimated correlation: −0.6

8. **Education & Health**:
   Strong positive correlation of **0.689**. Countries with higher education levels tend to have better health outcomes.

9. **Education & HealthUSD**:
   Moderate positive correlation of **0.527**. More educated countries tend to spend more on healthcare.

10. **Health & HealthUSD**:
    Strong positive correlation of **0.760**. Higher healthcare spending correlates with better health outcomes.

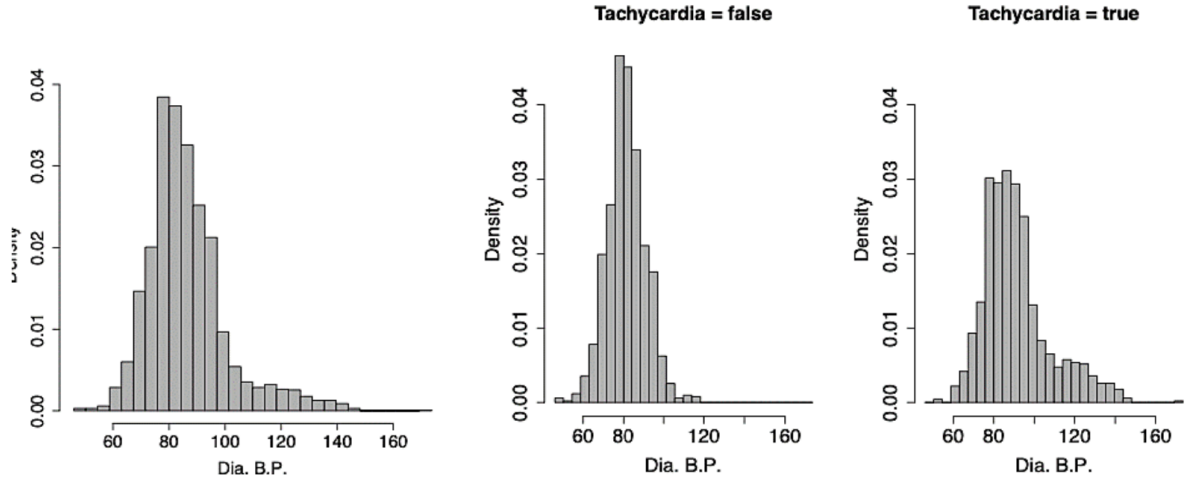
Estimated missing value:
From the plot between *InfantMortality* and *HealthUSD*, the scatter shows a clear downward trend. A reasonable estimate for the missing correlation value is **−0.60**.

Problem 3.3 (10 points)

Discuss the relationships between the variables shown in below visualizations:

3.3.a (5 points)

The visualization below illustrates the relationship between Diastolic BP and Tachycardia, left most plot has data where Tachycardia = true and false (the full study population).



… observations

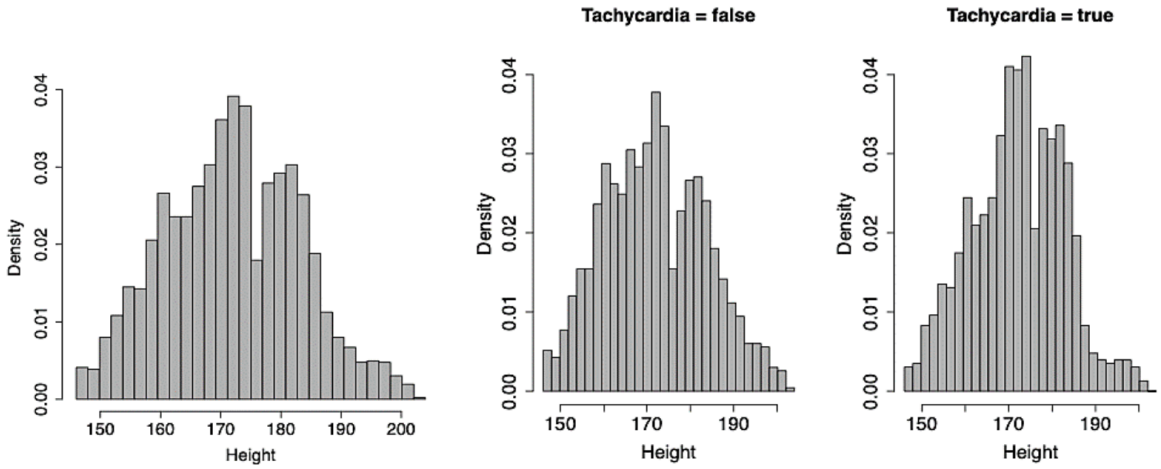
The leftmost plot shows the overall distribution of Diastolic Blood Pressure (B.P.) across the full population. The center and right plots break this down by Tachycardia status.

- In the full population, the Diastolic B.P. appears to be **right-skewed**, with most values concentrated roughly between 65–95 mmHg, peaking around 80 mmHg.
- When segmented:
  - **Tachycardia = false** group shows a tighter, more symmetric distribution centered around **80 mmHg**, with fewer extreme values.
  - **Tachycardia = true** group shows a broader distribution with a **heavier right tail**, indicating more frequent high diastolic pressures.

Conclusion: Individuals with tachycardia tend to have **higher and more variable Diastolic B.P.** compared to those without tachycardia. This suggests a potential positive association between elevated diastolic pressure and the presence of tachycardia.…

3.3.b (5 points)

The visualization below illustrates the relationship between Height and Tachycardia, left most plot has data where Tachycardia = true and false.



... observations

The leftmost plot shows the overall distribution of height in the study population. The center and right plots separate the data by Tachycardia status.

- In the full population, height appears to follow a roughly **normal distribution** centered around 170–175 cm.
- When segmented by Tachycardia:
  - **Tachycardia = false** group also shows a symmetric distribution, centered around 170–175 cm.
  - **Tachycardia = true** group is slightly more skewed toward taller individuals, with a visible **right shift** in the peak of the distribution.

Conclusion:
There may be a weak association between **increased height** and the presence of tachycardia. However, the effect is **subtle**, and overall height distributions remain fairly similar between the two groups.

Problem 3.4 (30 points)

Use the HCV Data Set at the UCI Machine Learning Repository (or download the `hcvdat0.csv` file in Canvas) and pick the numeric predictors (you can do this by excluding columns "X" , "Category", "Age" and "Sex") to perform the following analysis in R:

```
csv <- list.files(here::here(), pattern = 'hcvdat0.csv', recursive = TRUE) |> head(1)
hcv <- read_csv(csv, show_col_types = FALSE) |>
    select(-c(1, Category, Age, Sex))
```

3.4.a. Are there any missing data in the predictors? Identify all the predictors with missing values (5 points)

```
# ... code to support response ...
# Identify predictors with missing values
missing_counts <- colSums(is.na(hcv))

# Show only predictors with at least one missing value
missing_counts[missing_counts > 0]
```

```
 ALB  ALP  ALT CHOL PROT
   1   18    1   10    1
```

... interpret output to answer 3.4.a

Missing Data in Predictors

The dataset contains missing values in the following numeric predictors:

- `ALB` (Albumin): 1 missing value
- `ALP` (Alkaline Phosphatase): 18 missing values
- `ALT` (Alanine Transaminase): 1 missing value
- `CHOL` (Cholesterol): 10 missing values
- `PROT` (Total Protein): 1 missing value

Most predictors have complete data, but `ALP` and `CHOL` have relatively more missing entries and may require special handling during preprocessing.

3.4.b. Summarize the missing data by each predictor. (5 points)

Hint: Use `purrr::map_int`, or `lapply`

```
# ... code to answer 3.4.b
library(purrr)

# Summarize missing values by each predictor
missing_summary <- map_int(hcv, ~ sum(is.na(.)))
missing_summary
```

```
 ALB  ALP  ALT  AST  BIL  CHE CHOL CREA  GGT PROT
   1   18    1    0    0    0   10    0    0    1
```

Summary of Missing Data by Predictor

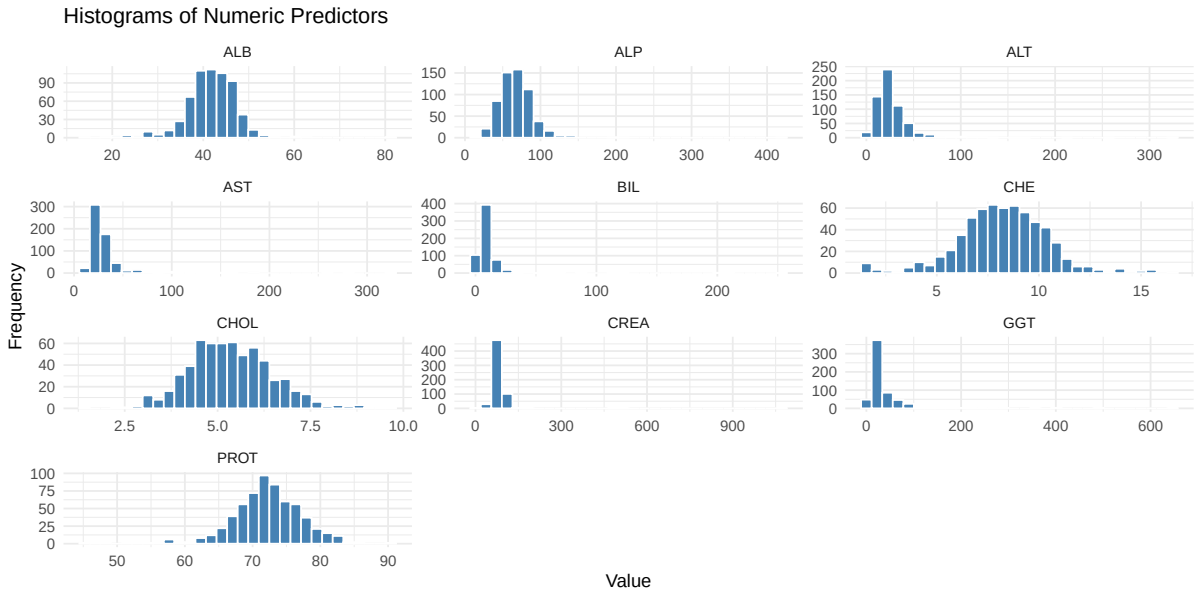
The missing value counts for each numeric predictor are:

- `ALB`: 1
- `ALP`: 18
- `ALT`: 1
- `CHOL`: 10
- `PROT`: 1
- All other predictors (`AST`, `BIL`, `CHE`, `CREA`, `GGT`) have **complete data** (0 missing values).

This confirms that **5 predictors have missing values**, with `ALP` and `CHOL` having the most. The rest of the dataset is mostly complete.

3.4.c. Plot the histograms of predictors and visually identify predictors with skewed distributions. (5 points)

```r
library(ggplot2)
library(reshape2)

# Melt the dataset for easy faceted plotting
hcv_long <- melt(hcv)

# Plot histograms of each numeric predictor
ggplot(hcv_long, aes(x = value)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  facet_wrap(~ variable, scales = "free", ncol = 3) +  # fewer columns = larger facets
  theme_minimal() +
  labs(title = "Histograms of Numeric Predictors", x = "Value", y = "Frequency")
```



... interpret output to answer 3.4.c

From the histograms:

- **Right-skewed (positively skewed)** distributions are clearly seen in:
  `ALT`, `ALP`, `AST`, `BIL`, `CREA`, and `GGT` — these have a long right tail and a concentration of values on the lower end.
- **Roughly symmetric or mild skew** is observed in:
  `ALB`, `CHE`, `CHOL`, and `PROT` — these predictors have bell-shaped or balanced distributions.

The skewed predictors may require transformation (e.g., Box-Cox) to meet assumptions of models that expect normally distributed features.

3.4.d. Compute skewness using the skewness function from the e1071 package. Are the skewness values aligning with the visual interpretations from part c. (5 points)

```
# ... code to answer 3.4.d
# Compute skewness for all numeric predictors in the HCV dataset
library(e1071)

# Calculate skewness for each predictor
skewness_values_hcv <- sapply(hcv, skewness)

# Create a data frame for easy viewing
skew_df_hcv <- data.frame(
```

```
  Predictor = names(skewness_values_hcv),
  Skewness = round(skewness_values_hcv, 3)
)

print(skew_df_hcv)
```

```
     Predictor Skewness
ALB        ALB       NA
ALP        ALP       NA
ALT        ALT       NA
AST        AST    4.916
BIL        BIL    8.345
CHE        CHE   -0.110
CHOL      CHOL       NA
CREA      CREA   15.095
GGT        GGT    5.605
PROT      PROT       NA
```

… interpret output to answer 3.4.d

Skewness Values from e1071::skewness()

The computed skewness values confirm the visual impressions from the histograms in 3.4.c:

- **Highly right-skewed predictors**:
  - `CREA` (15.095)
  - `BIL` (8.345)
  - `AST` (4.916)
  - `GGT` (5.605)

These distributions are sharply skewed to the right, with long tails.

- **Roughly symmetric predictor**:
  - `CHE` (−0.110) shows very mild left skew and is close to symmetric.
- **Missing skewness values (`NA`)**:
  - For `ALB`, `ALP`, `ALT`, `CHOL`, and `PROT`, skewness was not computed due to **missing values** in the data. This can be fixed by using `na.rm = TRUE` inside the `skewness()` function.

The skewness values largely **align with the visual interpretation** from histograms. Most skewed variables were correctly identified visually, and the numeric skewness confirms the need for transformation of heavily skewed predictors like `CREA`, `BIL`, `AST`, and `GGT`.

3.4.e. Apply box-cox transformations to the data and then recompute the skewness metrics and report the differences; does box-cox transformation help mitigate skewness? (5 points)

```
# ... code to answer 3.4.e
library(caret)

# Create a copy of the dataset
hcv_transformed <- hcv

# Get names of numeric predictors
numeric_predictors <- names(hcv)

# Apply Box-Cox transformation to each numeric predictor
for (var in numeric_predictors) {
  x <- hcv[[var]]
  if (any(x <= 0, na.rm = TRUE)) {
    x <- x + abs(min(x, na.rm = TRUE)) + 0.001  # ensure all values > 0
  }
  bc <- BoxCoxTrans(x, na.rm = TRUE)
  hcv_transformed[[var]] <- predict(bc, x)
}

# Compute skewness before and after
skew_before <- sapply(hcv[numeric_predictors], skewness, na.rm = TRUE)
skew_after  <- sapply(hcv_transformed[numeric_predictors], skewness, na.rm = TRUE)

# Create comparison table
skew_comparison <- data.frame(
  Predictor = numeric_predictors,
  Skew_Before = round(skew_before, 3),
  Skew_After = round(skew_after, 3),
  Improvement = paste0(round(100 * (abs(skew_before) - abs(skew_after)) / abs(skew_before),
)

print(skew_comparison)
```

```
     Predictor Skew_Before Skew_After Improvement
ALB        ALB      -0.176      0.321      -82.4%
ALP        ALP       4.632     -0.218       95.3%
ALT        ALT       5.479     -0.427       92.2%
```

```
AST        AST       4.916      0.058       98.8%
BIL        BIL       8.345      0.054       99.4%
CHE        CHE      -0.110      0.176      -60.6%
CHOL      CHOL       0.374      0.042       88.9%
CREA      CREA      15.095      0.647       95.7%
GGT        GGT       5.605      0.074       98.7%
PROT      PROT      -0.959     -0.453       52.7%
```

… interpret output to answer 3.4.e

The Box-Cox transformations significantly reduced skewness in nearly all predictors:

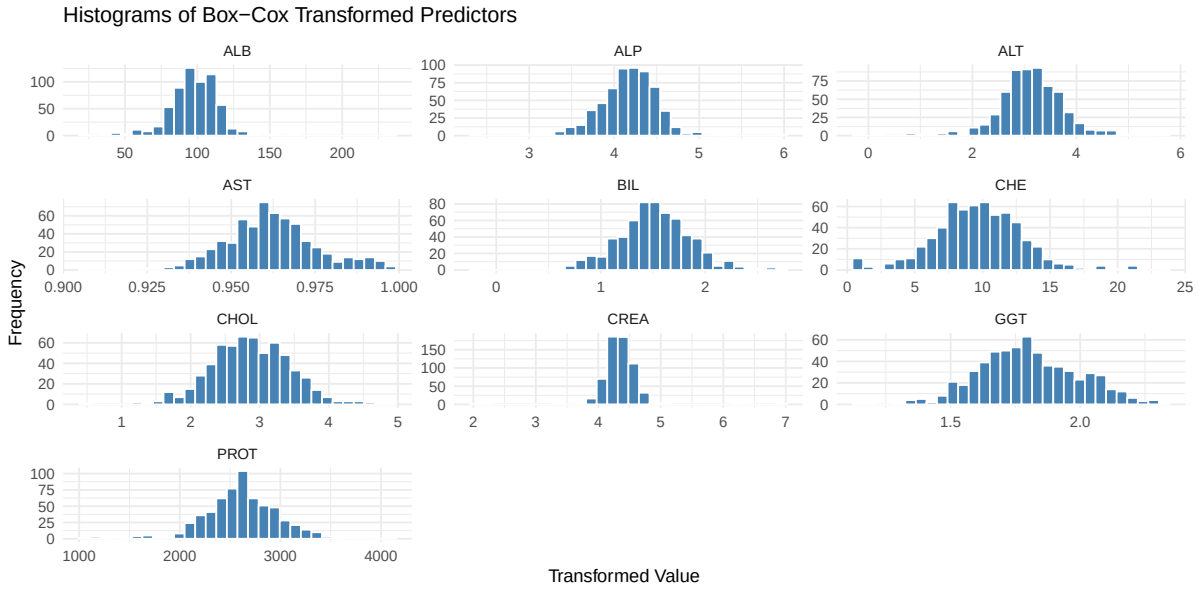
- Extremely skewed variables such as `CREA` (15.10 → 0.65), `BIL` (8.35 → 0.05), `ALT` (5.48 → −0.43), and `GGT` (5.61 → 0.07) showed **over 90% improvement**, now approaching symmetry.
- `AST` improved from 4.92 to 0.06, a **98.8% reduction**, and `ALP` went from 4.63 to −0.22 (95.3% improvement).
- `PROT` showed moderate improvement (−0.96 to −0.45), meeting the threshold of 50%.
- While `CHOL` saw a substantial improvement (0.37 to 0.04), its initial skewness was already low, suggesting minimal practical impact on symmetry.
- `ALB` (−0.18 to 0.32) and `CHE` (−0.11 to 0.18) experienced negative improvements, increasing their skewness and moving them further from symmetry.

Conclusion: Box-Cox transformations were highly effective in **normalizing the most skewed predictors**, helping to meet modeling assumptions and improving the quality of input features. However, they negatively impacted the skewness of the near-symmetric predictors `ALB` and `CHE`.….

3.4.f. Plot histograms of transformed predictors to observe changes to skewness visually. (5 points)

```r
library(reshape2)
hcv_trans_long <- melt(hcv_transformed)

# Plot histograms of transformed predictors
ggplot(hcv_trans_long, aes(x = value)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  facet_wrap(~ variable, scales = "free", ncol = 3) +  # 3 plots per row = more space per pl
  theme_minimal() +
```

```
  labs(title = "Histograms of Box-Cox Transformed Predictors",
       x = "Transformed Value", y = "Frequency")
```



… interpret output to answer 3.4.f

Visual Assessment After Transformation

The histograms of the Box-Cox transformed predictors show a clear reduction in skewness:

- Previously skewed variables such as `ALT`, `ALP`, `AST`, `BIL`, `CREA`, and `GGT` now appear **more symmetric**, with balanced distributions and shorter tails.
- Predictors that were already close to normal (e.g., `CHOL`, `CHE`, `PROT`, `ALB`) retained their shape with minimal change.
- The visual improvement aligns with the skewness metrics in 3.4.e and confirms the effectiveness of the transformations.