Homework Batch 1

Alyssa Gurkas, Deirdre Flynn, Marc Fridson

# HA 2.1 Use the help function to explore what the series gold, woolyrnq and gas represent.

help("gold")

The gold dataset represents the daily gold prices in US dollars from January 1, 1985 to March 31, 1989.

help("woolyrnq")

The woolyrnq dataset represents the quarterly Australian production of woollen yarn (in tonnes) from March 1965 - September 1994.

help("gas")

The gas dataset represents the Australian monthly gas production from 1956-1995.

## HA 2.1.a. Use autoplot() to plot each of these in separate plots.

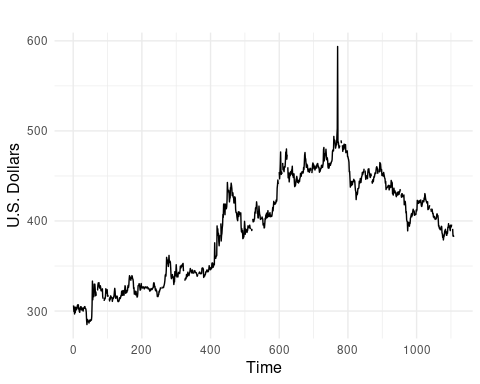
### HA 2.1.a. Approach

For this problem, each datasets will be autoplotted separately.

### HA 2.1.a. Analysis

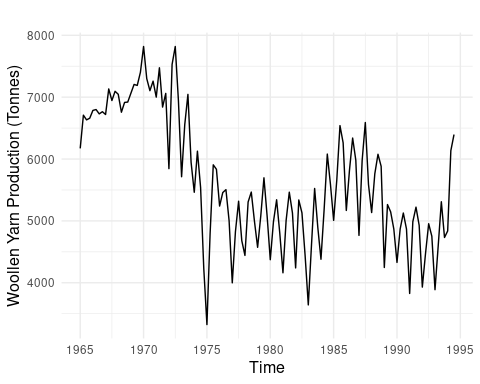
##### gold Plot: Daily Gold Prices in US dollars: January 1, 1985 to March 31, 1989

autoplot(gold)+  
 xlab("Time") + ylab("U.S. Dollars")



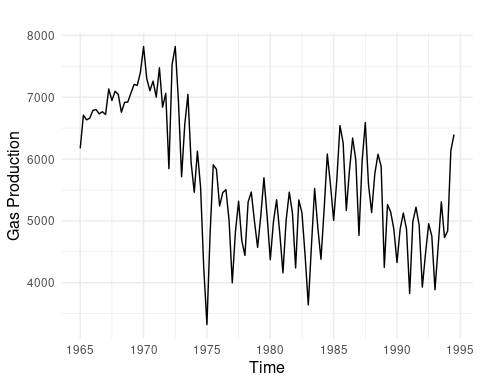
##### woolyrnq Plot: Quarterly Australian Production of Woollen Yarn (in tonnes) from March 1965 - September 1994

autoplot(woolyrnq)+  
 xlab("Time") + ylab("Woollen Yarn Production (Tonnes)")



##### gas Plot: Australian Monthly Gas Production from 1956-1995

autoplot(woolyrnq)+  
 xlab("Time") + ylab("Gas Production")



## HA 2.1.b. What is the frequency of each series? Hint: apply the frequency() function.

### HA 2.1.b. Approach

To view the frequency of each series, you can use the frequency function from the stats package. This is helpful in determining what type of data you are working with for example, daily, monthly, or annual data.

### HA 2.1.b. Analysis

frequency(gold)

## [1] 1

frequency(woolyrnq)

## [1] 4

frequency(gas)

## [1] 12

Gold Frequency: 1 Woollen Yarn Frequency: 4 Gas Frequency: 12

#### HA 2.1.c. Use which.max() to spot the outlier in the gold series. Which observation was it?

which.max(gold)

## [1] 770

The outlier in the gold series is 770 U.S. Dollars.

# HA 2.3 Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.

## HA 2.3.a. You can read the data into R with the following script:

retaildata <- readxl::read\_excel("data/retail.xlsx", skip=1)  
# The second argument (skip=1) is required because the Excel sheet has two header rows.

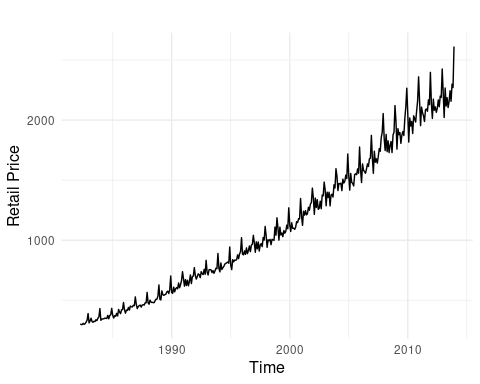
## HA 2.3.b. Select one of the time series as follows (but replace the column name with your own chosen column):

myts <- ts(retaildata[,"A3349335T"],  
 frequency=12, start=c(1982,4))

## HA 2.3.c. Explore your chosen retail time series these functions: autoplot(), ggseasonplot(), ggsubseriesplot(), gglagplot(), ggAcf(). Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

#### Time Series Plot: Retail Price

autoplot(myts)+   
 xlab("Time") + ylab("Retail Price")



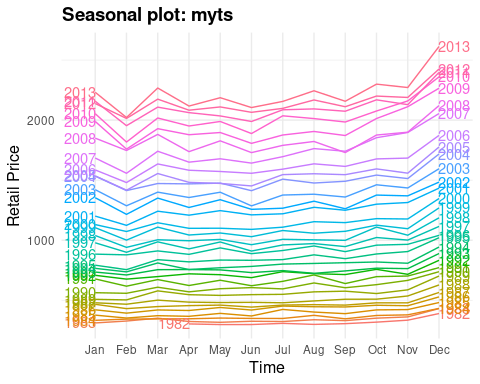
which.max(myts)

## [1] 381

This means that the 381st row is the maximum value in this dataset.

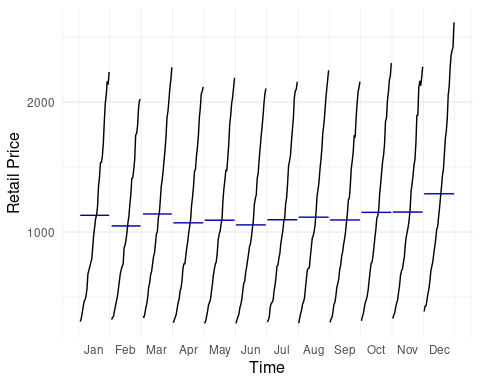
#### Time Series Seasonality Plot: Retail Price

ggseasonplot(myts,  
 year.labels=TRUE, year.labels.left=TRUE)+   
 xlab("Time") + ylab("Retail Price")

 A seasonal plot is useful for understanding the underlying seasonal pattern. In this plot we can see that there is an increase in the product’s price in December, January, and a slight increase in March.

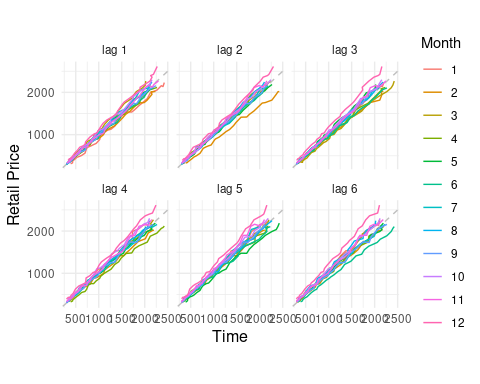
#### Time Series Subseries Plot: Retail Price

ggsubseriesplot(myts)+   
 xlab("Time") + ylab("Retail Price")

 A seasonal subseries plot can also be used to understand seasonal patterns. In Figure 6, the data for each season is collected together in time plots. As depicted in Figure 6, there is a spike in the product price in December, January, and a slight increase in March.

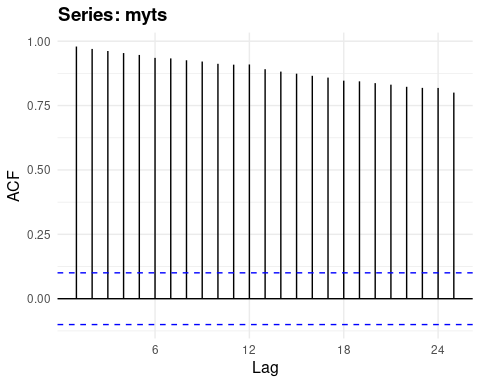
#### Lag Plot: Retail Price

gglagplot(myts,  
 lags=6)+   
 xlab("Time") + ylab("Retail Price")

 A lag plot helps assess autocorrelation by plotting a time series against its lagged values. The strong linear patterns across lags in the plot above suggests the series is highly predictable based on past values.

#### AFC Plot: Retail Price

ggAcf(myts)

 The ACF plot shows significant autocorrelation at all lags, indicating strong persistence in the data and justifying the use of past values in forecasting models.

# KJ 3.1

*Problem Introduction* The UC Irvine Machine Learning Repository6 contains a data set related to glass identification. 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:

data(Glass)  
str(Glass)

## 'data.frame': 214 obs. of 10 variables:  
## $ RI : num 1.52 1.52 1.52 1.52 1.52 ...  
## $ Na : num 13.6 13.9 13.5 13.2 13.3 ...  
## $ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...  
## $ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...  
## $ Si : num 71.8 72.7 73 72.6 73.1 ...  
## $ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...  
## $ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...  
## $ Ba : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...  
## $ Type: Factor w/ 6 levels "1","2","3","5",..: 1 1 1 1 1 1 1 1 1 1 ...

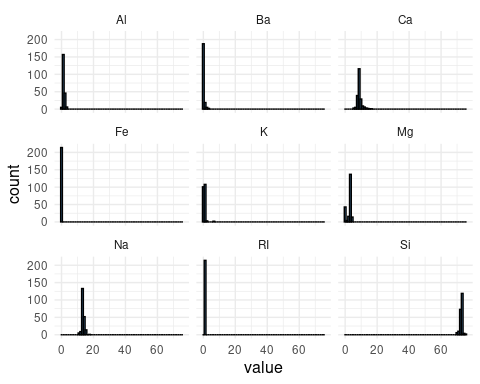
## KJ 3.1.a Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors

### KJ 3.1.a Approach

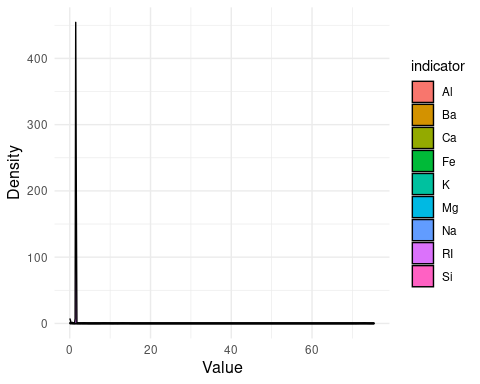
* Plot the data to see the distribution using a histogram. This can be done by using the ggplot facet\_wrap() argument, or by creating individual plots.
* Alternatively, the data can be visualized using the ggpairs() function from the GGally package. This provides a matrix of containing density plots, scatterplots, Pearson correlations, and boxplots among indicators.

### KJ 3.1.a Analysis

glass2 <- Glass %>%   
 select(-Type) %>%   
 pivot\_longer(everything(), names\_to = "indicator", values\_to = "value")  
  
# facet wrapped indicators  
ggplot(glass2, aes(value)) +  
 geom\_histogram(bins = 70, fill = "steelblue", color = "black") +  
 facet\_wrap(~indicator)



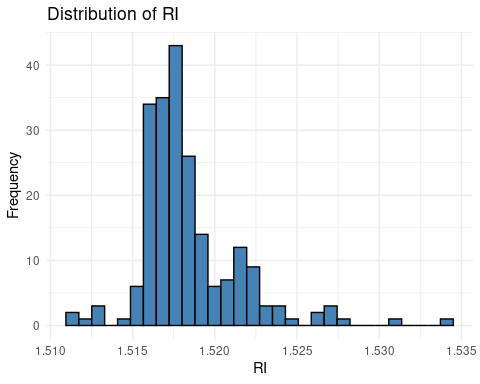
# density plots - doesn't show anything b/c values are not spread similarly  
ggplot(glass2) +  
 geom\_density(aes(x=value,fill=indicator)) +  
 labs(  
 x="Value",  
 y="Density"  
 )



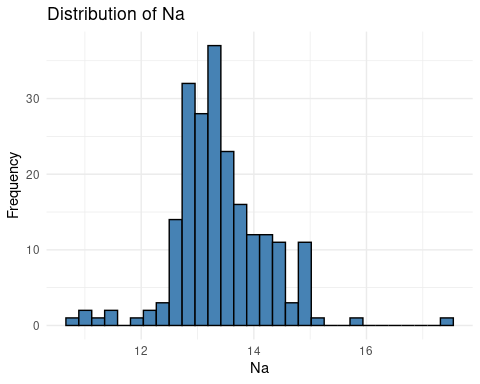
ggtitle("Density Plots of Indicators") +  
 theme\_bw() +  
 theme(axis.text.x = element\_text(face = 'bold', size = 10),  
 axis.text.y = element\_text(face = 'bold', size = 10))

## NULL

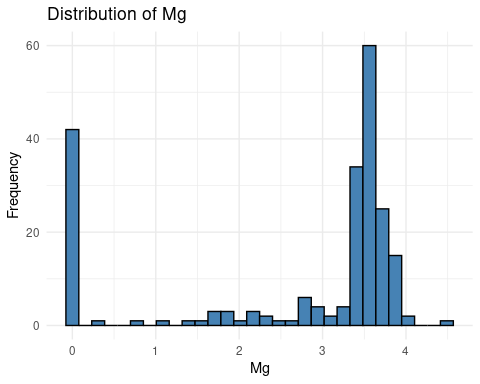
# Individual Indicators  
plot\_glass\_distributions <- function(data) {  
 numeric\_data <- data[sapply(data, is.numeric)]  
   
 for (colname in names(numeric\_data)) {  
 p <- ggplot(data, aes\_string(x = colname)) +  
 geom\_histogram(bins = 30, fill = "steelblue", color = "black") +  
 theme\_minimal() +  
 labs(title = paste("Distribution of", colname),  
 x = colname,  
 y = "Frequency")  
 print(p) # Print each plot individually  
 readline(prompt = "Press [Enter] to show the next plot...")  
 }  
}  
  
# running function, need to save each ggplot so that it can go into the word doc  
# plots should be included in the Appendix.  
plot\_glass\_distributions(Glass)



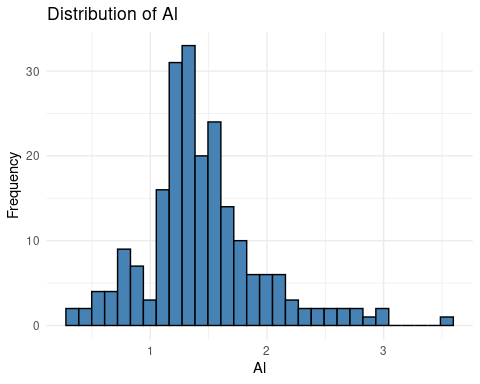
## Press [Enter] to show the next plot...



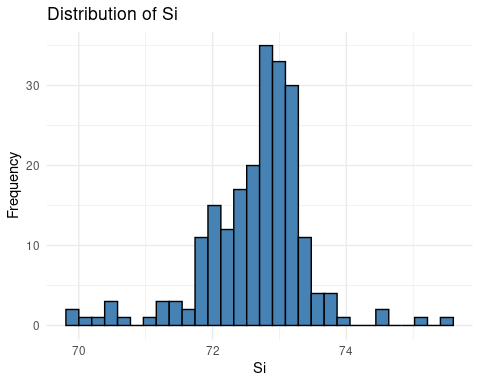
## Press [Enter] to show the next plot...



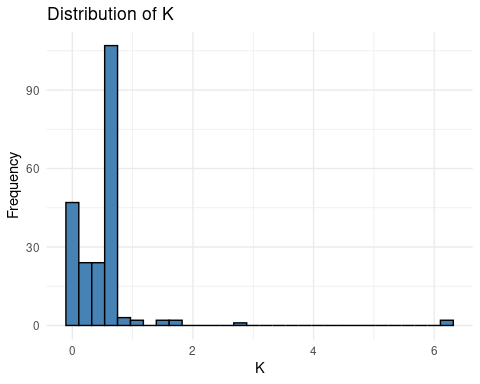
## Press [Enter] to show the next plot...



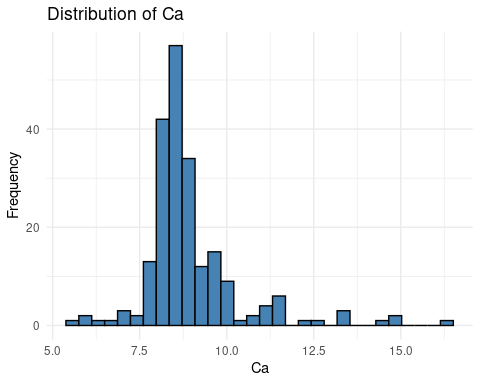
## Press [Enter] to show the next plot...



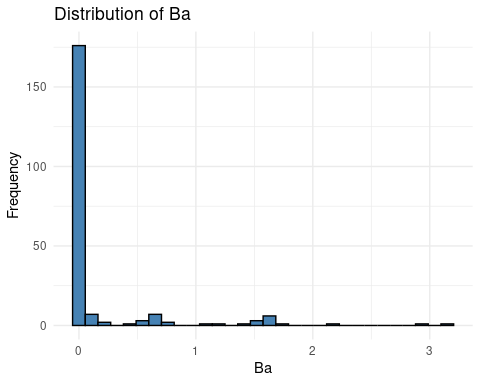
## Press [Enter] to show the next plot...



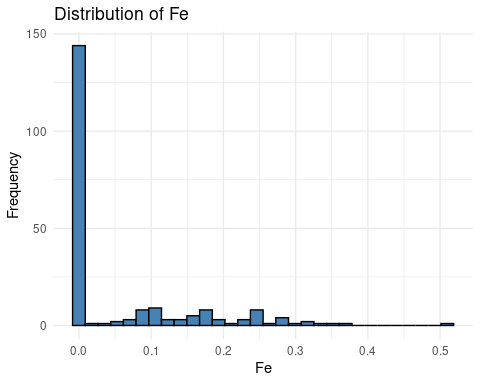
## Press [Enter] to show the next plot...



## Press [Enter] to show the next plot...

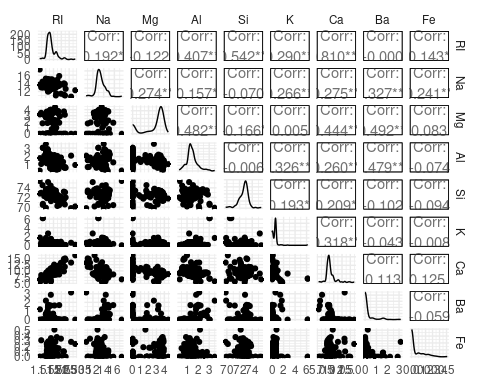


## Press [Enter] to show the next plot...



## Press [Enter] to show the next plot...

# plotting the indicators using the ggpairs() function  
Glass %>%   
 select(-Type) %>%   
GGally::ggpairs()

 From these plots, it is apparent that not all of the predictors are normally distributed, and that some have outliers, such as Ba, Al, and Na. Additionally, from the scatterplot matrix produced by the ggpairs() function, it is apparent that certain predictors have stronger relationships than others.

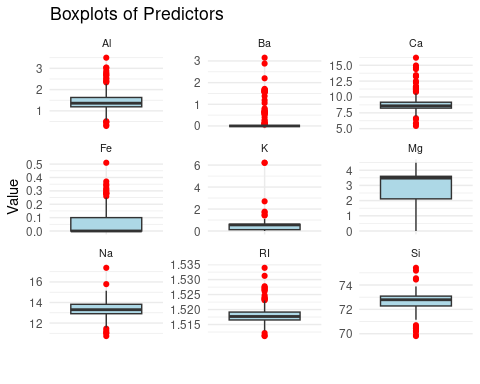
## KJ 3.1.b Do there appear to be any outliers in the data? Are any predictors skewed?

### KJ 3.1.b Approach

Outliers can be detected through visualizations like boxplots. By plotting the outliers in red, it is apparent how many there are, and if the data may need to be transformed.

### KJ 3.1.b Analysis

ggplot(glass2, aes(x = "", y = value)) +  
 geom\_boxplot(fill = "lightblue", outlier.color = "red") +  
 facet\_wrap(~ indicator, scales = "free\_y") +  
 theme\_minimal() +  
 labs(title = "Boxplots of Predictors",  
 y = "Value",  
 x = "") +  
 theme(strip.text = element\_text(size = 8))

 # From this plot, it is apparent that there are outliers in nearly all of the # predictors except for Mg. To handle skewnewss, the data can be transformed # through Log or Box-Cox transformations.

## KJ 3.1.c Are there any relevant transformations of one or more predictors that might improve the classification model?

### KJ 3.1.c Approach

There are a few transformations or steps that can be taken to improve the classification model: - *Data Cleaning*: Check missing values (Removal versus imputation) - *Data Transformation*: Center the data or scale it. For features that are skewed, can perform Log or Box-Cox transformations to normalize distributions. - *Identify Problematic Predictors*: Filter data for near-zero variance predictors (these are predictors where most of the values are the same), and highly correlated predictors that may lead to collinearity issues and increase model variance.

### KJ 3.1.c Analysis

#### Data Cleaning: Check for duplicate and missing values

# Check for missing values  
na\_counts <- map\_dfc(Glass, ~ sum(is.na(.x)))  
names(na\_counts) <- paste0("NA\_", names(Glass))  
  
distinct\_counts <- map\_dfc(Glass, ~ n\_distinct(.x, na.rm = TRUE))  
names(distinct\_counts) <- paste0("T\_", names(Glass))  
  
sum\_missing <- bind\_cols(na\_counts, distinct\_counts)

From the sum\_missing dataframe, we can see that no indicators have missing values, which means imputations are not necessary.

#### Data Transformation: Centering and Scaling Data

Note: cannot use the glass2 version because the PreProcess() function is intended for wide datasets. When typing trans into the console, we can see that nine indicators were center and scaled, and that one indicator was ignored.

trans <- preProcess(Glass, method=c("center","scale"))  
  
# Can add in the Box-Cox method as well by adding it into the method arugment  
trans <- preProcess(Glass, method=c("center", "scale", "BoxCox"))

To identify problematic predictors we can filter data for near-zero variance predictors (these are predictors where most of the values are the same), and highly correlated predictors that may lead to collinearity issues and increase model variance.

If there are any near-zero variance predictors, they would be captured by the nearZeroVar function from the caret package.

nearZeroVar(Glass)

## integer(0)

For highly correlated indicators, the cor() function from the caret package can be used.

Glass <- Glass %>%   
 select(-Type)  
corr <- stats::cor(Glass)  
dim(corr)

## [1] 9 9

# KJ 3.2

The soybean data can also be found at the UC Irvine Machine Learning Repository. Data were collected to predict disease in 683 soybeans. The 35 predictors are mostly categorical and include information on the environmental conditions (e.g., temperature, precipitation) and plant conditions (e.g., left spots, mold growth). The outcome labels consist of 19 distinct classes.

data(Soybean)

## KJ 3.2.a. Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

The “leaf.mild”, “mycelium”, and “seclerotia” have near zero variance.

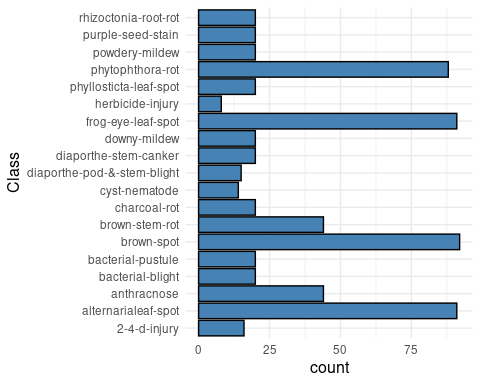
nearZeroVar(Soybean)

## [1] 19 26 28

colnames(Soybean[,c(19,26,28)])

## [1] "leaf.mild" "mycelium" "sclerotia"

ggplot(Soybean, aes(Class)) +  
 geom\_histogram(stat="count", fill = "steelblue", color = "black") +   
 coord\_flip()

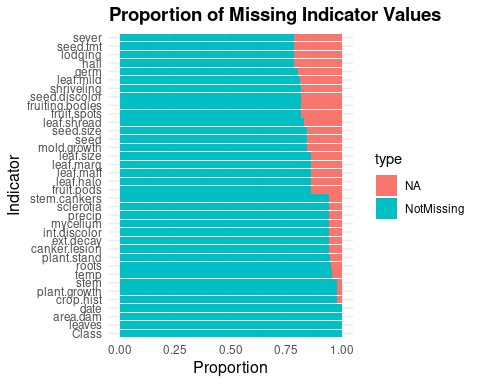


## KJ 3.2.b. Roughly 18% of the data are missing. Are there particular predictors that are more likely to be missing? Is the pattern of missing data related to the classes?

### KJ 3.2.b. Approach

* Check for missing values
* Plot the missing data

# Check for missing values  
sum\_missing\_sb <- map\_dfr(names(Soybean), function(var) {  
 col\_data <- Soybean[[var]]  
 tibble(  
 "indicator" = var,  
 "Total" = sum(!is.na(col\_data)),  
 "NA" = sum(is.na(col\_data)),  
 "PropNA" = sum(is.na(col\_data))/sum(!is.na(col\_data)),  
 "NotMissing" = abs(sum(!is.na(col\_data))-sum(is.na(col\_data))),  
 "PropNotMissing" = abs(sum(!is.na(col\_data))-sum(is.na(col\_data)))/sum(!is.na(col\_data))  
 )  
})  
  
missing\_plot\_data <- sum\_missing\_sb %>%  
 select(-PropNotMissing,-PropNA,-Total) %>%   
 pivot\_longer(cols = c( "NA",  
 "NotMissing"),   
 names\_to = "type",   
 values\_to = "count")   
  
ggplot(missing\_plot\_data, aes(x = reorder(indicator, -count, sum), y = count, fill = type)) +   
 geom\_bar(position="fill", stat="identity")+  
 coord\_flip()+   
 labs(  
 title = "Proportion of Missing Indicator Values",  
 x ="Indicator",  
 y = "Proportion"  
 )



## KJ 3.2.c. Develop a strategy for handling missing data, either by eliminating predictors or imputation.

### KJ KJ 3.2.c. Approach

To develop a strategy for handling missing data, this analysis will explore both approaches for handling missing data (i.e., eliminating predictors or imputation).

### KJ 3.2.c. Analysis

sb\_cleaned <- Soybean %>%   
 select(-Class) %>%   
 mutate\_if(is.factor, as.numeric)  
  
sb\_processed <- preProcess(sb\_cleaned, method='knnImpute')

If we wanted to eliminate the indicators that have a significant amount of missing data, we could simply remove the variables from the dataset:

problem\_indicators <- sum\_missing\_sb %>%   
 group\_by(indicator) %>%   
 filter(PropNA > .2)  
print(problem\_indicators$indicator)

## [1] "hail" "sever" "seed.tmt" "lodging"

# 'hail', 'sever', 'seed.tmt', 'lodging' are removed because the proportion of missing  
# values is over 20%  
sb\_cleaned\_opt2 <- Soybean %>%   
 select(-hail, -sever, -seed.tmt, -lodging)

# HA 6.2

The plastics data set consists of the monthly sales (in thousands) of product A for a plastics manufacturer for five years.

## HA 6.2.a Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?

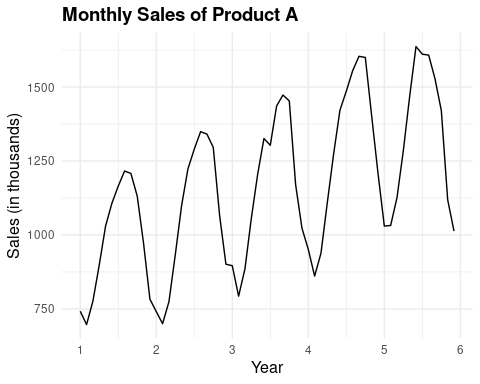
### HA 6.2.a Approach

Use autoplot() to visualize the time series and look for repeating seasonal patterns and an overall trend.

### HA 6.2.a. Analysis

library(fpp2) # Loads 'plastics' dataset and required tools

autoplot(plastics) +  
 ggtitle("Monthly Sales of Product A") +  
 xlab("Year") + ylab("Sales (in thousands)")



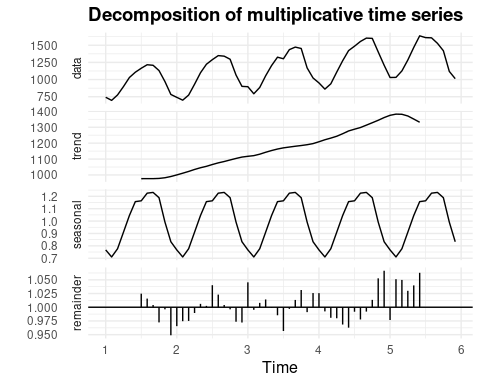
## HA 6.2.b Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.

### HA 6.2.b Approach

Ppply decompose() with the type = “multiplicative” option to break the series into trend, seasonal, and remainder components.

### HA 6.2.b. Analysis

decomp\_plastics <- decompose(plastics, type = "multiplicative")  
autoplot(decomp\_plastics)



## HA 6.2.c Do the results support the graphical interpretation from part a?

### HA 6.2.c Approach

Visually compare the decomposition components with my initial impressions from the time series plot.

### HA 6.2.c. Analysis

The decomposition confirms the graphical interpretation: there is a strong seasonal pattern that repeats annually, and a trend-cycle showing an upward movement in sales over the five-year period.

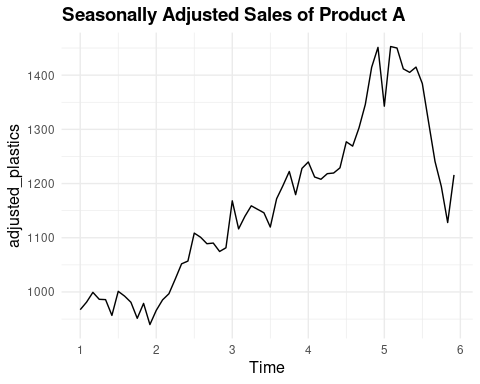
## HA 6.2.d Compute and plot the seasonally adjusted data.

### HA 6.2.d Approach

Divide the original data by the seasonal component to obtain seasonally adjusted values.

### HA 6.2.d. Analysis

adjusted\_plastics <- seasadj(decomp\_plastics)  
autoplot(adjusted\_plastics) +  
 ggtitle("Seasonally Adjusted Sales of Product A")

 The outlier significantly distorts the trend-cycle and irregular components, introducing noise into the decomposition.

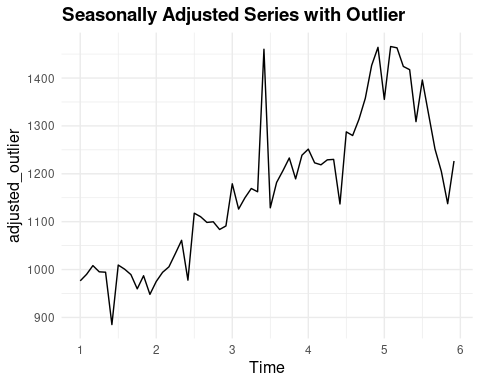
## HA 6.2.e Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

### HA 6.2.e. Approach

Introduce an outlier in the middle of the series and observe how it affects the seasonally adjusted values.

### HA 6.2.e. Analysis

plastics\_outlier <- plastics  
plastics\_outlier[30] <- plastics\_outlier[30] + 500 # Introduce outlier  
  
decomp\_outlier <- decompose(plastics\_outlier, type = "multiplicative")  
adjusted\_outlier <- seasadj(decomp\_outlier)  
  
autoplot(adjusted\_outlier) +  
 ggtitle("Seasonally Adjusted Series with Outlier")



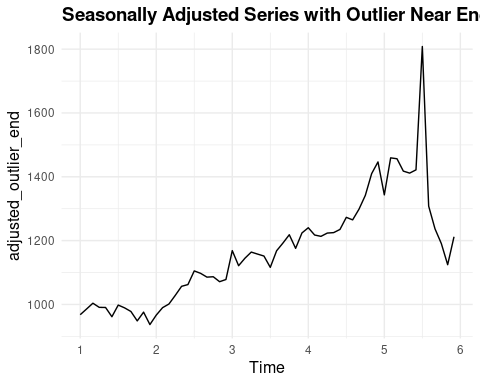
## HA 6.2.f Does it make any difference if the outlier is near the end rather than in the middle of the time series?

### HA 6.2.f. Approach

Compare the impact of placing an outlier near the end of the series instead of the middle.

### HA 6.2.f. Analysis

plastics\_outlier\_end <- plastics  
plastics\_outlier\_end[55] <- plastics\_outlier\_end[55] + 500 # Outlier near end  
  
decomp\_outlier\_end <- decompose(plastics\_outlier\_end, type = "multiplicative")  
adjusted\_outlier\_end <- seasadj(decomp\_outlier\_end)  
  
autoplot(adjusted\_outlier\_end) +  
 ggtitle("Seasonally Adjusted Series with Outlier Near End")

 Outliers near the end may have a more pronounced impact on the final few trend estimates and make the decomposition less reliable in that region due to fewer surrounding data points for smoothing.

# HA 7.5

Data set books contains the daily sales of paperback and hardcover books at the same store. The task is to forecast the next four days’ sales for paperback and hardcover books.

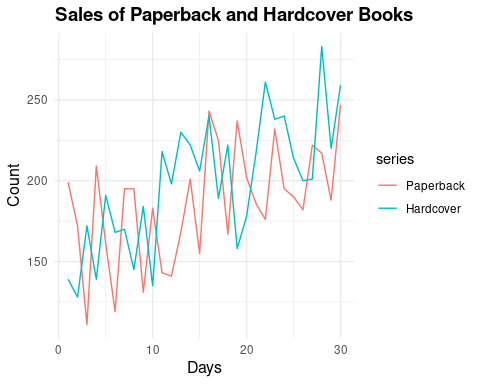
## HA 7.5.a. Plot the series and discuss the main features of the data.

data(books)

### HA 7.5.a. Approach

Step 1: Determine the main features of the data by examining the: (1) Trend (2) Seasonality (3) Heteroscedasticity (4) Level shifts or structural changes

autoplot(books)+  
 ggtitle("Sales of Paperback and Hardcover Books")+  
 xlab("Days")+  
 ylab("Count")



# There is an upward trend.   
  
length(books)

## [1] 60

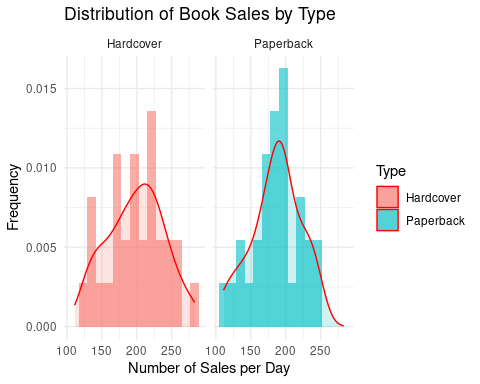
frequency(books)

## [1] 1

summary(books)

## Paperback Hardcover   
## Min. :111.0 Min. :128.0   
## 1st Qu.:167.2 1st Qu.:170.5   
## Median :189.0 Median :200.5   
## Mean :186.4 Mean :198.8   
## 3rd Qu.:207.2 3rd Qu.:222.0   
## Max. :247.0 Max. :283.0

books\_plot <- books %>%  
 as.data.frame() %>%  
 pivot\_longer(cols = everything(), names\_to = "Type", values\_to = "Sales")  
  
# Distribution of Book Sales   
# This is showing that hardcover is slightly right skewed.  
ggplot(books\_plot, aes(x = Sales, fill = Type)) +  
 geom\_histogram(aes(y = ..density..),alpha = 0.6, position = "identity", bins = 15) +  
 geom\_density(alpha = 0.2, color = "red") +  
 facet\_wrap(~Type) +  
 labs(title = "Distribution of Book Sales by Type",  
 x = "Number of Sales per Day", y = "Frequency") +  
 theme\_minimal()



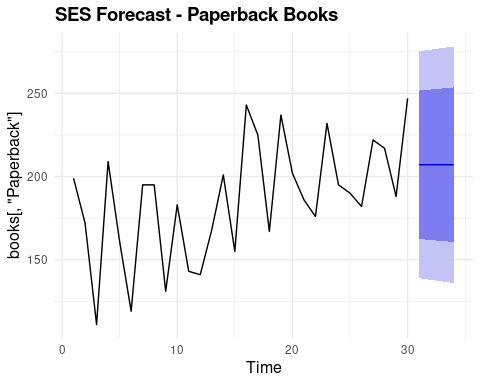
## HA 7.5.b. Use the ses() function to forecast each series, and plot the forecasts.

## HA 7.5.b. Approach

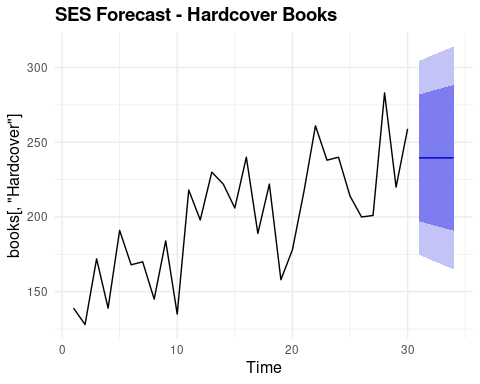
Use simple exponential smoothing via ses() to forecast the next 4 days for both paperback and hardcover sales.

## HA 7.5.b. Analysis

# Forecast 4 days ahead using Simple Exponential Smoothing  
fc\_paperback <- ses(books[, "Paperback"], h = 4)  
fc\_hardcover <- ses(books[, "Hardcover"], h = 4)  
  
# Plot forecasts  
autoplot(fc\_paperback) + ggtitle("SES Forecast - Paperback Books")



autoplot(fc\_hardcover) + ggtitle("SES Forecast - Hardcover Books")

 These plots show the forecasted values with prediction intervals assuming the level is the only component modeled.

## HA 7.5.c. Compute the RMSE values for the training data in each case.

## HA 7.5.c. Approach

Extract the RMSE from the accuracy measures of the SES models.

## HA 7.5.c. Analysis

# Compute RMSE for the training data  
accuracy(fc\_paperback)[2] # RMSE for Paperback

## [1] 33.63769

accuracy(fc\_hardcover)[2] # RMSE for Hardcover

## [1] 31.93101

Since the RMSE is slightly lower for hardcover books, the SES model performs marginally better for that series.

# HA 7.6

We will continue with the daily sales of paperback and hardcover books in data set books.

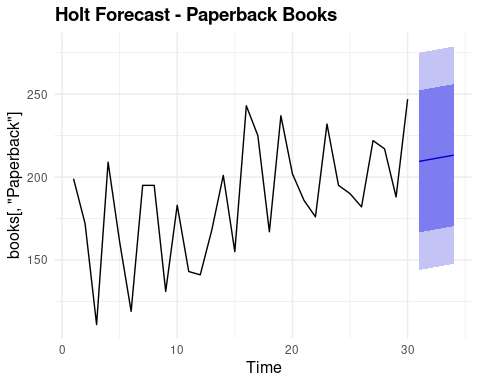
## HA 7.6.a. Apply Holt’s linear method to the paperback and hardback series and compute four-day forecasts in each case.

### HA 7.6.a. Approach

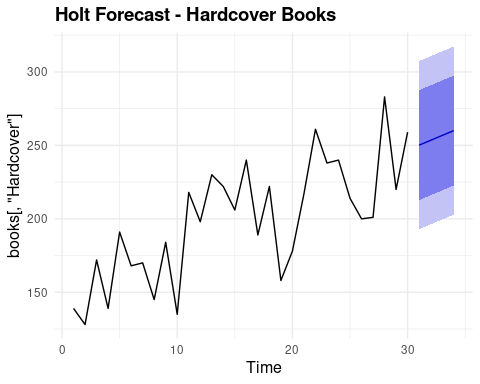
Use the holt() function to compute four-day forecasts for both series.

### HA 7.6.a. Analysis

# Holt’s linear method  
fc\_holt\_paperback <- holt(books[, "Paperback"], h = 4)  
fc\_holt\_hardcover <- holt(books[, "Hardcover"], h = 4)  
  
# Plot the forecasts  
autoplot(fc\_holt\_paperback) + ggtitle("Holt Forecast - Paperback Books")



autoplot(fc\_holt\_hardcover) + ggtitle("Holt Forecast - Hardcover Books")



## HA 7.6.b. Compare the RMSE measures of Holt’s method for the two series to those of simple exponential smoothing in the previous question. Discuss the merits of the two forecasting methods for these data sets.

(Remember that Holt’s method is using one more parameter than SES.)

### HA 7.6.b. Approach

Extract the RMSE values from Holt’s forecasts and compare them to the SES results to evaluate which model fits better.

### HA 7.6.b. Analysis

# Get RMSE values for Holt's method  
rmse\_holt\_paperback <- accuracy(fc\_holt\_paperback)[2] # RMSE for Paperback  
rmse\_holt\_hardcover <- accuracy(fc\_holt\_hardcover)[2] # RMSE for Hardcover  
  
rmse\_holt\_paperback

## [1] 31.13692

rmse\_holt\_hardcover

## [1] 27.19358

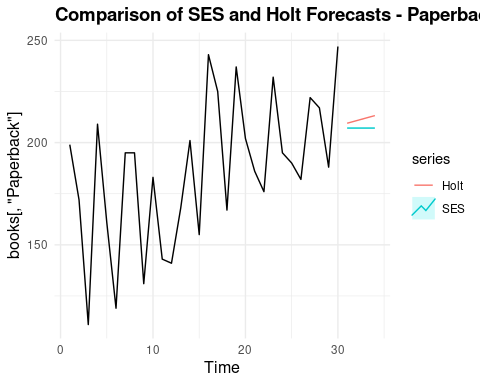
Holt’s method slightly reduced the RMSE for both series compared to SES, suggesting that modeling the trend improves forecast accuracy.

## HA 7.6.c. Compare the forecasts for the two series using both methods. Which do you think is best?

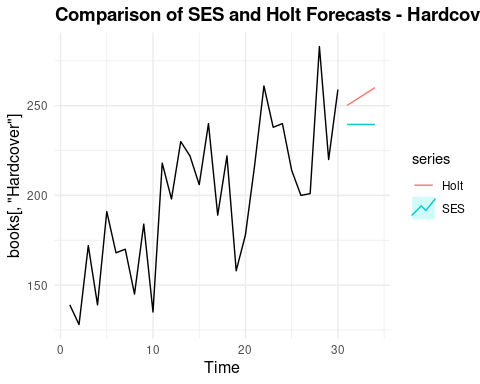
### HA 7.6.c. Approach

Visually and numerically compare the forecast values from SES and Holt’s method for each series. ### HA 7.6.c. Analysis

# SES forecasts from earlier (assumed to be saved as fc\_paperback and fc\_hardcover)  
autoplot(books[, "Paperback"]) +  
 autolayer(fc\_paperback, series = "SES", PI = FALSE) +  
 autolayer(fc\_holt\_paperback$mean, series = "Holt", PI = FALSE) +  
 ggtitle("Comparison of SES and Holt Forecasts - Paperback")



autoplot(books[, "Hardcover"]) +  
 autolayer(fc\_hardcover, series = "SES", PI = FALSE) +  
 autolayer(fc\_holt\_hardcover$mean, series = "Holt", PI = FALSE) +  
 ggtitle("Comparison of SES and Holt Forecasts - Hardcover")

 The Holt forecasts capture the direction of the trend more effectively than SES. Given the small reduction in RMSE and better trend alignment, Holt’s method appears better suited for both book series.

## HA 7.6.d. Calculate a 95% prediction interval for the first forecast for each series, using the RMSE values and assuming normal errors. Compare your intervals with those produced using ses and holt.

### HA 7.6.d. Approach

Use the RMSE values to calculate manual 95% prediction intervals for the first forecast assuming normal distribution of errors.

### HA 7.6.d. Analysis

# Get point forecasts  
point\_fc\_pb\_ses <- fc\_paperback$mean[1]  
point\_fc\_pb\_holt <- fc\_holt\_paperback$mean[1]  
  
point\_fc\_hc\_ses <- fc\_hardcover$mean[1]  
point\_fc\_hc\_holt <- fc\_holt\_hardcover$mean[1]  
  
# Get RMSE  
rmse\_pb\_ses <- 33.64  
rmse\_pb\_holt <- as.numeric(rmse\_holt\_paperback)  
  
rmse\_hc\_ses <- 31.93  
rmse\_hc\_holt <- as.numeric(rmse\_holt\_hardcover)  
  
# Compute 95% prediction intervals manually: point forecast ± 1.96 \* RMSE  
interval\_pb\_ses <- c(point\_fc\_pb\_ses - 1.96 \* rmse\_pb\_ses,  
 point\_fc\_pb\_ses + 1.96 \* rmse\_pb\_ses)  
  
interval\_pb\_holt <- c(point\_fc\_pb\_holt - 1.96 \* rmse\_pb\_holt,  
 point\_fc\_pb\_holt + 1.96 \* rmse\_pb\_holt)  
  
interval\_hc\_ses <- c(point\_fc\_hc\_ses - 1.96 \* rmse\_hc\_ses,  
 point\_fc\_hc\_ses + 1.96 \* rmse\_hc\_ses)  
  
interval\_hc\_holt <- c(point\_fc\_hc\_holt - 1.96 \* rmse\_hc\_holt,  
 point\_fc\_hc\_holt + 1.96 \* rmse\_hc\_holt)  
  
interval\_pb\_ses

## [1] 141.1753 273.0441

interval\_pb\_holt

## [1] 148.4384 270.4951

interval\_hc\_ses

## [1] 176.9773 302.1429

interval\_hc\_holt

## [1] 196.8745 303.4733

# Check model-based intervals  
fc\_paperback$lower[1,]

## 80% 95%   
## 162.4882 138.8670

fc\_paperback$upper[1,]

## 80% 95%   
## 251.7311 275.3523

fc\_holt\_paperback$lower[1,]

## 80% 95%   
## 166.6035 143.9130

fc\_holt\_paperback$upper[1,]

## 80% 95%   
## 252.3301 275.0205

The calculated 95% prediction intervals closely match those generated by ses() and holt(), which supports the assumption of normally distributed forecast errors.

# HA 8.1

Figure 8.31 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.

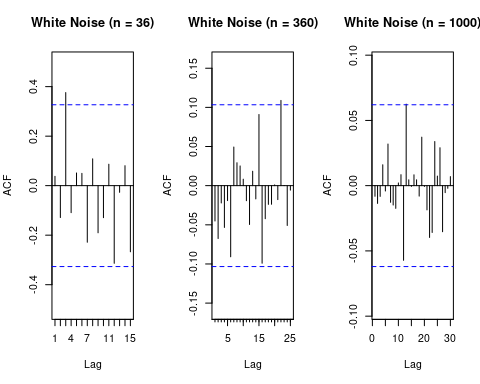
## HA 8.1.a. Explain the differences among these figures. Do they all indicate that the data are white noise?

### HA 8.1.a. Approach

Simulate three white noise series with different sample sizes and compare their ACF plots to assess randomness and variability.

### HA 8.1.a. Analysis

library(fpp2)  
  
set.seed(123)  
wn\_36 <- ts(rnorm(36))  
wn\_360 <- ts(rnorm(360))  
wn\_1000 <- ts(rnorm(1000))  
  
par(mfrow = c(1, 3)) # Side-by-side plots  
Acf(wn\_36, main = "White Noise (n = 36)")  
Acf(wn\_360, main = "White Noise (n = 360)")  
Acf(wn\_1000, main = "White Noise (n = 1000)")

 All three ACF plots represent white noise, as there are no significant autocorrelations beyond the 95% confidence bounds. However, the plots differ visually: the small-sample (n=36) ACF shows more random spikes, while the larger samples (n=360, n=1000) produce ACFs that cluster more closely around zero. This is because smaller samples have greater sampling variability.

## HA 8.1.b. Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

The blue dashed lines on the ACF plots show the range where we expect the values to fall if the data are truly random. When we have more data points, this range gets smaller, making it easier to spot unusual patterns. With fewer data points, the range is wider, so the values can bounce around more just by chance.

All three series are random numbers but the bars on the plots don’t look exactly the same. This is because when you work with small sets of data, the results can look more random or uneven. With larger sets of data, things tend to even out, and the bars stay closer to zero.

So, while the three plots look a little different, they all show the same thing: random data with no real pattern. The differences happen just because of the number of data points used.