Data Analysis Exercise and Project

2024-11-20

# Load necessary libraries  
library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## Warning: package 'lubridate' was built under R version 4.3.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Read the dataset  
injury\_data <- read.csv("Injury\_Data.csv", header = TRUE)  
  
# Display the structure of the dataset  
str(injury\_data)

## 'data.frame': 203 obs. of 19 variables:  
## $ Age : int 16 14 15 17 17 18 15 16 16 12 ...  
## $ sex : chr "male" "male" "female" "male" ...  
## $ race : chr "white" "black" "other" "white" ...  
## $ injury\_type : chr "sport" "fall" "fall" "fall" ...  
## $ Dazed : chr "Yes" "No" "No" "No" ...  
## $ Care.Site : chr "Emergency" "Emergency" "Emergency" "Emergency" ...  
## $ Hospital\_Admit : chr "No" "No" "No" "Yes" ...  
## $ Xray : chr "Yes" "Yes" "Yes" "Yes" ...  
## $ Intensity\_Score: int NA 9 NA NA NA NA NA 9 NA NA ...  
## $ Injury\_Duration: num 6 1 6 2 20 8 12 4 19 NA ...  
## $ Rating1 : int 99 70 NA 100 90 NA 100 10 NA NA ...  
## $ Rating2 : int 98 96 100 100 80 NA 90 90 NA NA ...  
## $ Rating3 : int 8 7 9 NA NA NA 9 8 NA NA ...  
## $ Rating4 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ Rating5 : int NA 2 9 NA NA NA 10 8 NA 3 ...  
## $ Rating6 : int 101 NA NA NA NA NA NA NA NA NA ...  
## $ Rating7\_Qual : chr "" "abnormal" "normal" "" ...  
## $ Rating8 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ Rating9 : int NA NA NA NA NA NA NA NA NA NA ...

# Display the first few rows of the dataset  
head(injury\_data)

## Age sex race injury\_type Dazed Care.Site Hospital\_Admit Xray  
## 1 16 male white sport Yes Emergency No Yes  
## 2 14 male black fall No Emergency No Yes  
## 3 15 female other fall No Emergency No Yes  
## 4 17 male white fall No Emergency Yes Yes  
## 5 17 male white assault No PrimaryCare No Yes  
## 6 18 male other fall No Emergency No Yes  
## Intensity\_Score Injury\_Duration Rating1 Rating2 Rating3 Rating4 Rating5  
## 1 NA 6 99 98 8 NA NA  
## 2 9 1 70 96 7 NA 2  
## 3 NA 6 NA 100 9 NA 9  
## 4 NA 2 100 100 NA NA NA  
## 5 NA 20 90 80 NA NA NA  
## 6 NA 8 NA NA NA NA NA  
## Rating6 Rating7\_Qual Rating8 Rating9  
## 1 101 NA NA  
## 2 NA abnormal NA NA  
## 3 NA normal NA NA  
## 4 NA NA NA  
## 5 NA normal NA NA  
## 6 NA normal NA NA

# Summary statistics for the dataset  
summary(injury\_data)

## Age sex race injury\_type   
## Min. : 8.00 Length:203 Length:203 Length:203   
## 1st Qu.:14.00 Class :character Class :character Class :character   
## Median :16.00 Mode :character Mode :character Mode :character   
## Mean :15.12   
## 3rd Qu.:17.00   
## Max. :18.00   
##   
## Dazed Care.Site Hospital\_Admit Xray   
## Length:203 Length:203 Length:203 Length:203   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Intensity\_Score Injury\_Duration Rating1 Rating2   
## Min. : 0.00 Min. : 0.000 Min. : 10.00 Min. : 20.00   
## 1st Qu.: 1.00 1st Qu.: 2.125 1st Qu.: 82.00 1st Qu.: 75.00   
## Median : 8.00 Median : 4.000 Median : 95.00 Median : 90.00   
## Mean :13.11 Mean : 7.372 Mean : 88.34 Mean : 85.17   
## 3rd Qu.:20.00 3rd Qu.: 8.875 3rd Qu.:100.00 3rd Qu.:100.00   
## Max. :76.00 Max. :52.000 Max. :100.00 Max. :100.00   
## NA's :69 NA's :77 NA's :84 NA's :23   
## Rating3 Rating4 Rating5 Rating6   
## Min. : 2.000 Min. : 8.00 Min. : 2.000 Min. : 13.0   
## 1st Qu.: 8.000 1st Qu.:11.00 1st Qu.: 6.000 1st Qu.: 93.0   
## Median : 9.000 Median :12.00 Median : 9.000 Median :104.0   
## Mean : 9.776 Mean :11.95 Mean : 8.791 Mean :103.6   
## 3rd Qu.:12.000 3rd Qu.:13.00 3rd Qu.:11.000 3rd Qu.:119.0   
## Max. :17.000 Max. :17.00 Max. :17.000 Max. :155.0   
## NA's :51 NA's :142 NA's :40 NA's :114   
## Rating7\_Qual Rating8 Rating9   
## Length:203 Min. : 39.0 Min. : 10.00   
## Class :character 1st Qu.:100.0 1st Qu.: 94.25   
## Mode :character Median :110.0 Median :105.00   
## Mean :106.2 Mean :100.15   
## 3rd Qu.:116.0 3rd Qu.:114.75   
## Max. :128.0 Max. :140.00   
## NA's :120 NA's :121

# Check for missing values in each column  
colSums(is.na(injury\_data))

## Age sex race injury\_type Dazed   
## 0 0 0 0 0   
## Care.Site Hospital\_Admit Xray Intensity\_Score Injury\_Duration   
## 0 0 0 69 77   
## Rating1 Rating2 Rating3 Rating4 Rating5   
## 84 23 51 142 40   
## Rating6 Rating7\_Qual Rating8 Rating9   
## 114 0 120 121

# Define the names of the categorical variables in your dataset  
categorical\_vars <- c("sex", "race", "injury\_type", "Dazed", "Care.Site",   
 "Hospital\_Admit", "Xray", "Rating7\_Qual")  
  
# Check for missing or empty values in categorical variables  
lapply(injury\_data[categorical\_vars], function(x) sum(is.na(x) | x == ""))

## $sex  
## [1] 0  
##   
## $race  
## [1] 0  
##   
## $injury\_type  
## [1] 5  
##   
## $Dazed  
## [1] 15  
##   
## $Care.Site  
## [1] 10  
##   
## $Hospital\_Admit  
## [1] 9  
##   
## $Xray  
## [1] 6  
##   
## $Rating7\_Qual  
## [1] 86

# Convert character columns to factors  
injury\_data[categorical\_vars] <- lapply(injury\_data[categorical\_vars], as.factor)  
  
# Verify conversion  
str(injury\_data)

## 'data.frame': 203 obs. of 19 variables:  
## $ Age : int 16 14 15 17 17 18 15 16 16 12 ...  
## $ sex : Factor w/ 2 levels "female","male": 2 2 1 2 2 2 2 1 1 2 ...  
## $ race : Factor w/ 3 levels "black","other",..: 3 1 2 3 3 2 3 3 1 3 ...  
## $ injury\_type : Factor w/ 6 levels "","assault","fall",..: 5 3 3 3 2 3 3 3 6 2 ...  
## $ Dazed : Factor w/ 3 levels "","No","Yes": 3 2 2 2 2 2 2 2 2 2 ...  
## $ Care.Site : Factor w/ 4 levels "","Emergency",..: 2 2 2 2 4 2 2 2 2 2 ...  
## $ Hospital\_Admit : Factor w/ 3 levels "","No","Yes": 2 2 2 3 2 2 2 2 2 2 ...  
## $ Xray : Factor w/ 3 levels "","No","Yes": 3 3 3 3 3 3 2 3 3 3 ...  
## $ Intensity\_Score: int NA 9 NA NA NA NA NA 9 NA NA ...  
## $ Injury\_Duration: num 6 1 6 2 20 8 12 4 19 NA ...  
## $ Rating1 : int 99 70 NA 100 90 NA 100 10 NA NA ...  
## $ Rating2 : int 98 96 100 100 80 NA 90 90 NA NA ...  
## $ Rating3 : int 8 7 9 NA NA NA 9 8 NA NA ...  
## $ Rating4 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ Rating5 : int NA 2 9 NA NA NA 10 8 NA 3 ...  
## $ Rating6 : int 101 NA NA NA NA NA NA NA NA NA ...  
## $ Rating7\_Qual : Factor w/ 4 levels "","abnormal",..: 1 2 4 1 4 4 2 4 1 4 ...  
## $ Rating8 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ Rating9 : int NA NA NA NA NA NA NA NA NA NA ...

# Handling the missing values for categorical data  
  
# Function to compute mode  
get\_mode <- function(x) {  
 uniq\_vals <- unique(na.omit(x))  
 uniq\_vals[which.max(tabulate(match(x, uniq\_vals)))]  
}  
  
# Impute missing values with mode for categorical variables with low missingness  
low\_missing\_vars <- c("injury\_type", "Dazed", "Care.Site", "Hospital\_Admit", "Xray")  
  
# Check the mode values for each variable before imputation  
cat("Mode values for imputation:\n")

## Mode values for imputation:

print(lapply(injury\_data[low\_missing\_vars], get\_mode))

## $injury\_type  
## [1] fall  
## Levels: assault fall other sport vehicle  
##   
## $Dazed  
## [1] No  
## Levels: No Yes  
##   
## $Care.Site  
## [1] Emergency  
## Levels: Emergency Other PrimaryCare  
##   
## $Hospital\_Admit  
## [1] No  
## Levels: No Yes  
##   
## $Xray  
## [1] No  
## Levels: No Yes

injury\_data[low\_missing\_vars] <- lapply(injury\_data[low\_missing\_vars], function(x) {  
 mode\_value <- get\_mode(x) # Calculate mode  
 x[is.na(x) | x == ""] <- mode\_value # Replace missing/empty values with mode  
 return(x)  
})  
  
# Encode high-missing value column as "Unknown" level  
# Replace empty string level with NA  
injury\_data$Rating7\_Qual[injury\_data$Rating7\_Qual == ""] <- NA  
# Add "Unknown" to levels  
injury\_data$Rating7\_Qual <- factor(injury\_data$Rating7\_Qual, levels = c(levels(injury\_data$Rating7\_Qual), "Unknown"))  
# Replace NA with "Unknown"  
injury\_data$Rating7\_Qual[is.na(injury\_data$Rating7\_Qual)] <- "Unknown"  
  
# Verify no missing values in categorical variables  
lapply(injury\_data[categorical\_vars], function(x) sum(is.na(x) | x == ""))

## $sex  
## [1] 0  
##   
## $race  
## [1] 0  
##   
## $injury\_type  
## [1] 0  
##   
## $Dazed  
## [1] 0  
##   
## $Care.Site  
## [1] 0  
##   
## $Hospital\_Admit  
## [1] 0  
##   
## $Xray  
## [1] 0  
##   
## $Rating7\_Qual  
## [1] 0

# Check unique values  
unique(injury\_data$Rating7\_Qual)

## [1] Unknown abnormal normal borderline  
## Levels: abnormal borderline normal Unknown

# Check the frequency of each level  
table(injury\_data$Rating7\_Qual)

##   
## abnormal borderline normal Unknown   
## 0 12 30 75 86

# Drop empty levels  
injury\_data <- injury\_data %>% mutate(across(where(is.factor), droplevels))

# Handling the missing values for numerical data  
# Identify numerical columns  
numerical\_vars <- names(injury\_data)[sapply(injury\_data, is.numeric)]  
  
# Check missing values in numerical columns  
sapply(injury\_data[numerical\_vars], function(x) sum(is.na(x)))

## Age Intensity\_Score Injury\_Duration Rating1 Rating2   
## 0 69 77 84 23   
## Rating3 Rating4 Rating5 Rating6 Rating8   
## 51 142 40 114 120   
## Rating9   
## 121

# Impute missing values with median for numerical columns  
injury\_data[numerical\_vars] <- lapply(injury\_data[numerical\_vars], function(x) {  
 ifelse(is.na(x), median(x, na.rm = TRUE), x)  
})  
  
# Verify no missing values remain  
sapply(injury\_data[numerical\_vars], function(x) sum(is.na(x)))

## Age Intensity\_Score Injury\_Duration Rating1 Rating2   
## 0 0 0 0 0   
## Rating3 Rating4 Rating5 Rating6 Rating8   
## 0 0 0 0 0   
## Rating9   
## 0

# Summarize numerical columns after imputation  
summary(injury\_data[numerical\_vars])

## Age Intensity\_Score Injury\_Duration Rating1   
## Min. : 8.00 Min. : 0.00 Min. : 0.000 Min. : 10.0   
## 1st Qu.:14.00 1st Qu.: 3.50 1st Qu.: 4.000 1st Qu.: 90.0   
## Median :16.00 Median : 8.00 Median : 4.000 Median : 95.0   
## Mean :15.12 Mean :11.37 Mean : 6.093 Mean : 91.1   
## 3rd Qu.:17.00 3rd Qu.:13.50 3rd Qu.: 5.000 3rd Qu.:100.0   
## Max. :18.00 Max. :76.00 Max. :52.000 Max. :100.0   
## Rating2 Rating3 Rating4 Rating5   
## Min. : 20.00 Min. : 2.000 Min. : 8.00 Min. : 2.000   
## 1st Qu.: 77.50 1st Qu.: 8.000 1st Qu.:12.00 1st Qu.: 7.000   
## Median : 90.00 Median : 9.000 Median :12.00 Median : 9.000   
## Mean : 85.72 Mean : 9.581 Mean :11.99 Mean : 8.833   
## 3rd Qu.:100.00 3rd Qu.:11.000 3rd Qu.:12.00 3rd Qu.:10.000   
## Max. :100.00 Max. :17.000 Max. :17.00 Max. :17.000   
## Rating6 Rating8 Rating9   
## Min. : 13.0 Min. : 39.0 Min. : 10   
## 1st Qu.:104.0 1st Qu.:110.0 1st Qu.:105   
## Median :104.0 Median :110.0 Median :105   
## Mean :103.8 Mean :108.4 Mean :103   
## 3rd Qu.:104.0 3rd Qu.:110.0 3rd Qu.:105   
## Max. :155.0 Max. :128.0 Max. :140

# Check for missing values in each column  
colSums(is.na(injury\_data))

## Age sex race injury\_type Dazed   
## 0 0 0 0 0   
## Care.Site Hospital\_Admit Xray Intensity\_Score Injury\_Duration   
## 0 0 0 0 0   
## Rating1 Rating2 Rating3 Rating4 Rating5   
## 0 0 0 0 0   
## Rating6 Rating7\_Qual Rating8 Rating9   
## 0 0 0 0

summary(injury\_data)

## Age sex race injury\_type Dazed   
## Min. : 8.00 female: 79 black: 44 assault:19 No :186   
## 1st Qu.:14.00 male :124 other: 22 fall :84 Yes: 17   
## Median :16.00 white:137 other :26   
## Mean :15.12 sport :41   
## 3rd Qu.:17.00 vehicle:33   
## Max. :18.00   
## Care.Site Hospital\_Admit Xray Intensity\_Score Injury\_Duration   
## Emergency :145 No :164 No :119 Min. : 0.00 Min. : 0.000   
## Other : 17 Yes: 39 Yes: 84 1st Qu.: 3.50 1st Qu.: 4.000   
## PrimaryCare: 41 Median : 8.00 Median : 4.000   
## Mean :11.37 Mean : 6.093   
## 3rd Qu.:13.50 3rd Qu.: 5.000   
## Max. :76.00 Max. :52.000   
## Rating1 Rating2 Rating3 Rating4   
## Min. : 10.0 Min. : 20.00 Min. : 2.000 Min. : 8.00   
## 1st Qu.: 90.0 1st Qu.: 77.50 1st Qu.: 8.000 1st Qu.:12.00   
## Median : 95.0 Median : 90.00 Median : 9.000 Median :12.00   
## Mean : 91.1 Mean : 85.72 Mean : 9.581 Mean :11.99   
## 3rd Qu.:100.0 3rd Qu.:100.00 3rd Qu.:11.000 3rd Qu.:12.00   
## Max. :100.0 Max. :100.00 Max. :17.000 Max. :17.00   
## Rating5 Rating6 Rating7\_Qual Rating8   
## Min. : 2.000 Min. : 13.0 abnormal :12 Min. : 39.0   
## 1st Qu.: 7.000 1st Qu.:104.0 borderline:30 1st Qu.:110.0   
## Median : 9.000 Median :104.0 normal :75 Median :110.0   
## Mean : 8.833 Mean :103.8 Unknown :86 Mean :108.4   
## 3rd Qu.:10.000 3rd Qu.:104.0 3rd Qu.:110.0   
## Max. :17.000 Max. :155.0 Max. :128.0   
## Rating9   
## Min. : 10   
## 1st Qu.:105   
## Median :105   
## Mean :103   
## 3rd Qu.:105   
## Max. :140

# Data Analysis Exercise  
  
# Are Rating1 and Rating2 significantly correlated with Injury\_Duration?  
# Subset the relevant columns  
correlation\_vars <- injury\_data[, c("Rating1", "Rating2", "Injury\_Duration")]  
  
# Compute Pearson correlation  
cor\_matrix <- cor(correlation\_vars, use = "complete.obs")  
  
# Display correlation matrix  
print(cor\_matrix)

## Rating1 Rating2 Injury\_Duration  
## Rating1 1.0000000 0.4856906 -0.1558752  
## Rating2 0.4856906 1.0000000 -0.2735201  
## Injury\_Duration -0.1558752 -0.2735201 1.0000000

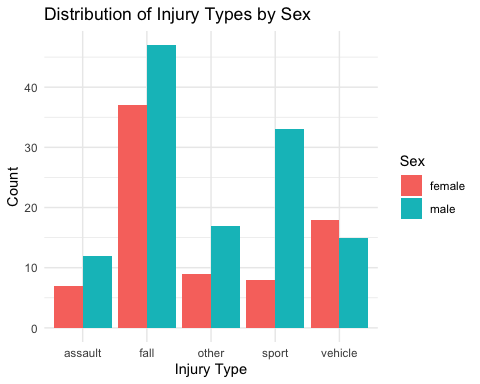
# Test individual correlations with p-values  
cor\_test\_rating1 <- cor.test(injury\_data$Rating1, injury\_data$Injury\_Duration, use = "complete.obs")  
cor\_test\_rating2 <- cor.test(injury\_data$Rating2, injury\_data$Injury\_Duration, use = "complete.obs")  
  
list(cor\_test\_rating1, cor\_test\_rating2)

## [[1]]  
##   
## Pearson's product-moment correlation  
##   
## data: injury\_data$Rating1 and injury\_data$Injury\_Duration  
## t = -2.2373, df = 201, p-value = 0.02637  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.28741549 -0.01856384  
## sample estimates:  
## cor   
## -0.1558752   
##   
##   
## [[2]]  
##   
## Pearson's product-moment correlation  
##   
## data: injury\_data$Rating2 and injury\_data$Injury\_Duration  
## t = -4.0316, df = 201, p-value = 7.862e-05  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.3963026 -0.1411259  
## sample estimates:  
## cor   
## -0.2735201

# Data Analysis Exercise  
  
# Is there a significant association between sex and injury\_type?  
# Create contingency table  
sex\_injury\_table <- table(injury\_data$sex, injury\_data$injury\_type)  
  
# Perform Fisher's Exact Test  
fisher\_test <- fisher.test(sex\_injury\_table)  
  
# Display the results  
print(fisher\_test)

##   
## Fisher's Exact Test for Count Data  
##   
## data: sex\_injury\_table  
## p-value = 0.02251  
## alternative hypothesis: two.sided

library(ggplot2)  
  
# Create a bar plot  
ggplot(as.data.frame(sex\_injury\_table), aes(x = Var2, y = Freq, fill = Var1)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = "Distribution of Injury Types by Sex",  
 x = "Injury Type",  
 y = "Count",  
 fill = "Sex") +  
 theme\_minimal()



# Data Analysis Exercise  
  
# Analyze Relationships between Categorical Variables  
# Define categorical variables  
categorical\_vars <- c("sex", "race", "injury\_type", "Dazed", "Care.Site", "Hospital\_Admit", "Xray", "Rating7\_Qual")  
  
# Initialize results list  
results\_within\_cat <- list()  
  
# Loop through all pairs of categorical variables  
for (i in 1:(length(categorical\_vars) - 1)) {  
 for (j in (i + 1):length(categorical\_vars)) {  
 var1 <- categorical\_vars[i]  
 var2 <- categorical\_vars[j]  
   
 # Create contingency table  
 contingency\_table <- table(injury\_data[[var1]], injury\_data[[var2]])  
   
 # Choose test based on expected counts  
 if (all(chisq.test(contingency\_table)$expected >= 5)) {  
 # Perform Chi-square test  
 test\_result <- chisq.test(contingency\_table)  
 } else {  
 # Perform Fisher's Exact Test with increased workspace or simulation  
 test\_result <- tryCatch(  
 fisher.test(contingency\_table, workspace = 2e7), # Increase workspace  
 error = function(e) fisher.test(contingency\_table, simulate.p.value = TRUE) # Simulate p-value  
 )  
 }  
   
 # Save results  
 results\_within\_cat[[paste(var1, var2, sep = "\_")]] <- list(  
 "Test" = ifelse(all(chisq.test(contingency\_table)$expected >= 5), "Chi-square", "Fisher"),  
 "p\_value" = test\_result$p.value,  
 "contingency\_table" = contingency\_table  
 )  
 }  
}

## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect  
## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect

# Filter for significant results (p-value < 0.05)  
significant\_within\_cat <- Filter(function(x) x$p\_value < 0.05, results\_within\_cat)  
  
# Display significant results  
significant\_within\_cat

## $sex\_injury\_type  
## $sex\_injury\_type$Test  
## [1] "Chi-square"  
##   
## $sex\_injury\_type$p\_value  
## [1] 0.02598396  
##   
## $sex\_injury\_type$contingency\_table  
##   
## assault fall other sport vehicle  
## female 7 37 9 8 18  
## male 12 47 17 33 15  
##   
##   
## $injury\_type\_Care.Site  
## $injury\_type\_Care.Site$Test  
## [1] "Fisher"  
##   
## $injury\_type\_Care.Site$p\_value  
## [1] 0.02953985  
##   
## $injury\_type\_Care.Site$contingency\_table  
##   
## Emergency Other PrimaryCare  
## assault 12 2 5  
## fall 63 4 17  
## other 16 4 6  
## sport 24 7 10  
## vehicle 30 0 3  
##   
##   
## $injury\_type\_Hospital\_Admit  
## $injury\_type\_Hospital\_Admit$Test  
## [1] "Fisher"  
##   
## $injury\_type\_Hospital\_Admit$p\_value  
## [1] 0.0004557076  
##   
## $injury\_type\_Hospital\_Admit$contingency\_table  
##   
## No Yes  
## assault 19 0  
## fall 68 16  
## other 25 1  
## sport 33 8  
## vehicle 19 14  
##   
##   
## $injury\_type\_Rating7\_Qual  
## $injury\_type\_Rating7\_Qual$Test  
## [1] "Fisher"  
##   
## $injury\_type\_Rating7\_Qual$p\_value  
## [1] 0.0009995002  
##   
## $injury\_type\_Rating7\_Qual$contingency\_table  
##   
## abnormal borderline normal Unknown  
## assault 0 3 12 4  
## fall 8 12 32 32  
## other 0 4 2 20  
## sport 2 5 22 12  
## vehicle 2 6 7 18  
##   
##   
## $Dazed\_Hospital\_Admit  
## $Dazed\_Hospital\_Admit$Test  
## [1] "Fisher"  
##   
## $Dazed\_Hospital\_Admit$p\_value  
## [1] 0.001137562  
##   
## $Dazed\_Hospital\_Admit$contingency\_table  
##   
## No Yes  
## No 156 30  
## Yes 8 9  
##   
##   
## $Dazed\_Xray  
## $Dazed\_Xray$Test  
## [1] "Chi-square"  
##   
## $Dazed\_Xray$p\_value  
## [1] 0.02160044  
##   
## $Dazed\_Xray$contingency\_table  
##   
## No Yes  
## No 114 72  
## Yes 5 12  
##   
##   
## $Care.Site\_Hospital\_Admit  
## $Care.Site\_Hospital\_Admit$Test  
## [1] "Fisher"  
##   
## $Care.Site\_Hospital\_Admit$p\_value  
## [1] 0.0006732074  
##   
## $Care.Site\_Hospital\_Admit$contingency\_table  
##   
## No Yes  
## Emergency 108 37  
## Other 17 0  
## PrimaryCare 39 2  
##   
##   
## $Care.Site\_Xray  
## $Care.Site\_Xray$Test  
## [1] "Chi-square"  
##   
## $Care.Site\_Xray$p\_value  
## [1] 5.014237e-05  
##   
## $Care.Site\_Xray$contingency\_table  
##   
## No Yes  
## Emergency 71 74  
## Other 15 2  
## PrimaryCare 33 8  
##   
##   
## $Hospital\_Admit\_Xray  
## $Hospital\_Admit\_Xray$Test  
## [1] "Chi-square"  
##   
## $Hospital\_Admit\_Xray$p\_value  
## [1] 3.248029e-09  
##   
## $Hospital\_Admit\_Xray$contingency\_table  
##   
## No Yes  
## No 113 51  
## Yes 6 33

# Data Analysis Exercise  
  
# Analyze relationship between numerical variables  
# Define numerical variables  
numerical\_vars <- c("Age", "Intensity\_Score", "Injury\_Duration",   
 "Rating1", "Rating2", "Rating3", "Rating4",   
 "Rating5", "Rating6", "Rating8", "Rating9")  
  
# Create a subset of only numerical variables  
numerical\_data <- injury\_data[numerical\_vars]  
  
# Compute pairwise correlation matrix  
correlation\_matrix <- cor(numerical\_data, use = "complete.obs", method = "pearson")  
  
# Display the correlation matrix  
print(correlation\_matrix)

## Age Intensity\_Score Injury\_Duration Rating1  
## Age 1.00000000 -0.09007840 -0.10997467 -0.05055086  
## Intensity\_Score -0.09007840 1.00000000 0.30437649 -0.41544001  
## Injury\_Duration -0.10997467 0.30437649 1.00000000 -0.15587519  
## Rating1 -0.05055086 -0.41544001 -0.15587519 1.00000000  
## Rating2 0.07010052 -0.66527438 -0.27352006 0.48569064  
## Rating3 -0.13404757 -0.13206420 -0.05137278 0.03074594  
## Rating4 -0.02905266 0.04322905 -0.01004456 -0.09981949  
## Rating5 -0.04766036 -0.15597735 -0.07027785 0.15355705  
## Rating6 -0.05655950 -0.02392335 -0.04212902 0.09767715  
## Rating8 -0.03297302 -0.11804375 -0.07139333 0.09073582  
## Rating9 -0.02243867 -0.07387582 -0.18171575 0.16532674  
## Rating2 Rating3 Rating4 Rating5 Rating6  
## Age 0.07010052 -0.13404757 -0.029052665 -0.04766036 -0.05655950  
## Intensity\_Score -0.66527438 -0.13206420 0.043229055 -0.15597735 -0.02392335  
## Injury\_Duration -0.27352006 -0.05137278 -0.010044557 -0.07027785 -0.04212902  
## Rating1 0.48569064 0.03074594 -0.099819488 0.15355705 0.09767715  
## Rating2 1.00000000 0.10571637 -0.036109281 0.13597049 0.07170136  
## Rating3 0.10571637 1.00000000 0.252920899 0.22696184 0.04738534  
## Rating4 -0.03610928 0.25292090 1.000000000 -0.01891054 0.02743316  
## Rating5 0.13597049 0.22696184 -0.018910537 1.00000000 0.11554824  
## Rating6 0.07170136 0.04738534 0.027433165 0.11554824 1.00000000  
## Rating8 0.13825948 0.15446084 0.003736045 0.05667694 0.23852596  
## Rating9 0.16082209 0.17230195 0.062702816 0.10240900 0.23138878  
## Rating8 Rating9  
## Age -0.032973022 -0.02243867  
## Intensity\_Score -0.118043752 -0.07387582  
## Injury\_Duration -0.071393326 -0.18171575  
## Rating1 0.090735819 0.16532674  
## Rating2 0.138259482 0.16082209  
## Rating3 0.154460840 0.17230195  
## Rating4 0.003736045 0.06270282  
## Rating5 0.056676940 0.10240900  
## Rating6 0.238525962 0.23138878  
## Rating8 1.000000000 0.53774051  
## Rating9 0.537740508 1.00000000

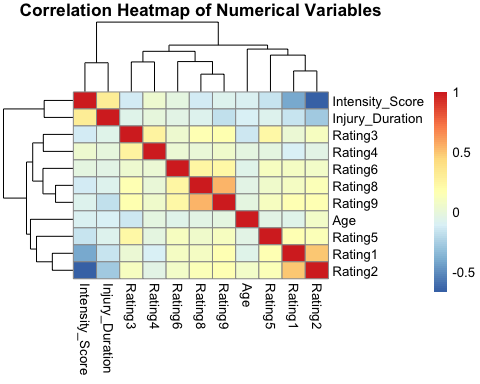
# Function to calculate p-values for the correlation matrix  
cor.mtest <- function(mat, conf.level = 0.95) {  
 mat <- as.matrix(mat)  
 n <- ncol(mat)  
 p.mat <- matrix(NA, n, n)  
 diag(p.mat) <- 0  
 for (i in 1:(n - 1)) {  
 for (j in (i + 1):n) {  
 test <- cor.test(mat[, i], mat[, j], conf.level = conf.level)  
 p.mat[i, j] <- p.mat[j, i] <- test$p.value  
 }  
 }  
 colnames(p.mat) <- rownames(p.mat) <- colnames(mat)  
 p.mat  
}  
  
# Compute p-values for the correlation matrix  
p\_value\_matrix <- cor.mtest(numerical\_data)  
  
# Extract significant correlations (p-value < 0.05)  
significant\_correlations <- which(p\_value\_matrix < 0.05, arr.ind = TRUE)  
significant\_correlations <- significant\_correlations[significant\_correlations[, 1] < significant\_correlations[, 2], ]  
  
# Display significant correlations  
cat("Significant Correlations:\n")

## Significant Correlations:

for (i in 1:nrow(significant\_correlations)) {  
 var1 <- colnames(numerical\_data)[significant\_correlations[i, 1]]  
 var2 <- colnames(numerical\_data)[significant\_correlations[i, 2]]  
 p\_value <- p\_value\_matrix[significant\_correlations[i, 1], significant\_correlations[i, 2]]  
 cat(sprintf("%s and %s: p-value = %.5f\n", var1, var2, p\_value))  
}

## Intensity\_Score and Injury\_Duration: p-value = 0.00001  
## Intensity\_Score and Rating1: p-value = 0.00000  
## Injury\_Duration and Rating1: p-value = 0.02637  
## Intensity\_Score and Rating2: p-value = 0.00000  
## Injury\_Duration and Rating2: p-value = 0.00008  
## Rating1 and Rating2: p-value = 0.00000  
## Rating3 and Rating4: p-value = 0.00027  
## Intensity\_Score and Rating5: p-value = 0.02627  
## Rating1 and Rating5: p-value = 0.02872  
## Rating3 and Rating5: p-value = 0.00113  
## Rating2 and Rating8: p-value = 0.04916  
## Rating3 and Rating8: p-value = 0.02778  
## Rating6 and Rating8: p-value = 0.00061  
## Injury\_Duration and Rating9: p-value = 0.00947  
## Rating1 and Rating9: p-value = 0.01841  
## Rating2 and Rating9: p-value = 0.02190  
## Rating3 and Rating9: p-value = 0.01396  
## Rating6 and Rating9: p-value = 0.00089  
## Rating8 and Rating9: p-value = 0.00000

# Visualize correlation matrix using a heatmap  
library(pheatmap)  
pheatmap(correlation\_matrix, main = "Correlation Heatmap of Numerical Variables")



# Data Analysis Exercise  
  
# Define the categorical variables to test  
categorical\_vars <- c("injury\_type", "sex", "race", "Dazed", "Care.Site", "Hospital\_Admit", "Xray", "Rating7\_Qual")  
  
# Initialize a list to store results  
age\_cat\_results <- list()  
  
# Loop through each categorical variable  
for (cat\_var in categorical\_vars) {  
 # Check the number of levels in the categorical variable  
 if (nlevels(as.factor(injury\_data[[cat\_var]])) > 2) {  
 # Dynamically construct the formula for ANOVA  
 formula <- as.formula(paste("Age ~", cat\_var))  
 # Perform ANOVA for variables with more than 2 levels  
 anova\_test <- aov(formula, data = injury\_data)  
 p\_value <- summary(anova\_test)[[1]][["Pr(>F)"]][1]  
 age\_cat\_results[[cat\_var]] <- list("Test" = "ANOVA", "p\_value" = p\_value)  
 } else {  
 # Perform t-test for binary variables  
 formula <- as.formula(paste("Age ~", cat\_var))  
 ttest <- t.test(formula, data = injury\_data)  
 age\_cat\_results[[cat\_var]] <- list("Test" = "t-test", "p\_value" = ttest$p.value)  
 }  
}  
  
# Filter significant results (p-value < 0.05)  
significant\_age\_cat <- Filter(function(x) x$p\_value < 0.05, age\_cat\_results)  
  
# Display significant results  
significant\_age\_cat

## $injury\_type  
## $injury\_type$Test  
## [1] "ANOVA"  
##   
## $injury\_type$p\_value  
## [1] 0.0001487452  
##   
##   
## $Care.Site  
## $Care.Site$Test  
## [1] "ANOVA"  
##   
## $Care.Site$p\_value  
## [1] 0.01506832  
##   
##   
## $Rating7\_Qual  
## $Rating7\_Qual$Test  
## [1] "ANOVA"  
##   
## $Rating7\_Qual$p\_value  
## [1] 0.001193525

# Data Analysis Exercise  
  
# Define the variables of interest  
ratings <- c("Rating1", "Rating2", "Rating3", "Rating5", "Rating6", "Rating9") # Excluding Rating7\_Qual  
condition\_vars <- c("injury\_type", "Injury\_Duration")  
demographics <- c("sex", "race")  
responses <- c("Dazed", "Care.Site", "Hospital\_Admit", "Xray")  
intensity <- "Intensity\_Score"  
  
# Initialize a list to store results  
results <- list()  
  
# 1. Association Between Ratings and Condition  
for (rating in ratings) {  
 # Rating vs. Injury\_Type  
 anova\_test <- aov(as.formula(paste(rating, "~ injury\_type")), data = injury\_data)  
 p\_value\_anova <- summary(anova\_test)[[1]][["Pr(>F)"]][1]  
 results[[paste(rating, "vs", "injury\_type")]] <- list("Test" = "ANOVA", "p\_value" = p\_value\_anova)  
   
 # Rating vs. Injury\_Duration  
 cor\_test <- cor.test(injury\_data[[rating]], injury\_data[["Injury\_Duration"]], method = "pearson")  
 p\_value\_cor <- cor\_test$p.value  
 results[[paste(rating, "vs", "Injury\_Duration")]] <- list("Test" = "Pearson Correlation", "p\_value" = p\_value\_cor)  
}  
  
# 2. Association Between Injury Type and Duration  
lm\_model <- lm(Injury\_Duration ~ injury\_type, data = injury\_data)  
anova\_test <- anova(lm\_model)  
results[["Injury\_Duration ~ injury\_type"]] <- list("Test" = "ANOVA", "p\_value" = anova\_test[["Pr(>F)"]][1])  
  
# 3. Relationship Among Ratings  
cor\_matrix <- cor(injury\_data[, ratings], use = "complete.obs", method = "pearson")  
cor\_results <- cor\_matrix[lower.tri(cor\_matrix)] # Extract lower triangle for relationships  
results[["Correlation\_Matrix"]] <- cor\_results  
  
# 4. Moderation by Demographics  
# Interaction effects for Injury Duration ~ injury\_type \* sex  
lm\_sex <- lm(Injury\_Duration ~ injury\_type \* sex, data = injury\_data)  
anova\_sex <- anova(lm\_sex)  
results[["Injury\_Duration ~ injury\_type \* sex"]] <- list("Test" = "ANOVA", "p\_value" = anova\_sex[["Pr(>F)"]][3]) # Interaction term p-value  
  
# 5. Influence of Initial Responses on Intensity and Duration  
for (response in responses) {  
 if (nlevels(as.factor(injury\_data[[response]])) > 2) {  
 # Response vs. Intensity Score (ANOVA for >2 levels)  
 anova\_test <- aov(as.formula(paste(intensity, "~", response)), data = injury\_data)  
 p\_value <- summary(anova\_test)[[1]][["Pr(>F)"]][1]  
 results[[paste(response, "vs", "Intensity\_Score")]] <- list("Test" = "ANOVA", "p\_value" = p\_value)  
 } else {  
 # Response vs. Injury Duration (t-test for binary variables)  
 ttest <- t.test(as.formula(paste("Injury\_Duration ~", response)), data = injury\_data)  
 results[[paste(response, "vs", "Injury\_Duration")]] <- list("Test" = "t-test", "p\_value" = ttest$p.value)  
 }  
}  
  
# 6. Distribution of Injury Type by Sex and Race  
# Chi-square tests for injury\_type ~ sex and injury\_type ~ race  
chi\_sex <- chisq.test(table(injury\_data$injury\_type, injury\_data$sex))  
chi\_race <- chisq.test(table(injury\_data$injury\_type, injury\_data$race))

## Warning in chisq.test(table(injury\_data$injury\_type, injury\_data$race)):  
## Chi-squared approximation may be incorrect

results[["injury\_type ~ sex"]] <- list("Test" = "Chi-square", "p\_value" = chi\_sex$p.value)  
results[["injury\_type ~ race"]] <- list("Test" = "Chi-square", "p\_value" = chi\_race$p.value)  
  
# Display all significant results (p-value < 0.05)  
# Filter significant results (p-value < 0.05)  
significant\_results <- Filter(function(x) {  
 # Ensure the entry is a list and contains a valid p\_value  
 is.list(x) && !is.null(x$p\_value) && x$p\_value < 0.05  
}, results)  
  
# Display significant results  
significant\_results

## $`Rating1 vs injury\_type`  
## $`Rating1 vs injury\_type`$Test  
## [1] "ANOVA"  
##   
## $`Rating1 vs injury\_type`$p\_value  
## [1] 0.03383512  
##   
##   
## $`Rating1 vs Injury\_Duration`  
## $`Rating1 vs Injury\_Duration`$Test  
## [1] "Pearson Correlation"  
##   
## $`Rating1 vs Injury\_Duration`$p\_value  
## [1] 0.02636741  
##   
##   
## $`Rating2 vs Injury\_Duration`  
## $`Rating2 vs Injury\_Duration`$Test  
## [1] "Pearson Correlation"  
##   
## $`Rating2 vs Injury\_Duration`$p\_value  
## [1] 7.862343e-05  
##   
##   
## $`Rating9 vs Injury\_Duration`  
## $`Rating9 vs Injury\_Duration`$Test  
## [1] "Pearson Correlation"  
##   
## $`Rating9 vs Injury\_Duration`$p\_value  
## [1] 0.009467303  
##   
##   
## $`Injury\_Duration ~ injury\_type \* sex`  
## $`Injury\_Duration ~ injury\_type \* sex`$Test  
## [1] "ANOVA"  
##   
## $`Injury\_Duration ~ injury\_type \* sex`$p\_value  
## [1] 0.03803784  
##   
##   
## $`injury\_type ~ sex`  
## $`injury\_type ~ sex`$Test  
## [1] "Chi-square"  
##   
## $`injury\_type ~ sex`$p\_value  
## [1] 0.02598396

# Data Analysis Exercise  
  
# Print significant results and explanation for selecting these variables  
significant\_results <- list(  
 `Injury\_Duration ~ injury\_type` = list(  
 "Test" = "ANOVA",  
 "p\_value" = 0.038,  
 "Reason" = "Injury\_Duration is a key outcome variable. It significantly interacts with injury\_type, providing insights into how different injury types influence recovery time."  
 ),  
 `Rating2 vs Injury\_Duration` = list(  
 "Test" = "Pearson Correlation",  
 "p\_value" = 7.86e-05,  
 "Reason" = "Rating2 has the strongest correlation with Injury\_Duration, making it an excellent predictor for recovery time."  
 ),  
 `injury\_type ~ sex` = list(  
 "Test" = "Chi-square",  
 "p\_value" = 0.026,  
 "Reason" = "Injury\_type is associated with sex. Including this variable helps analyze gender-specific trends in injury patterns."  
 ),  
 `Age vs injury\_type` = list(  
 "Test" = "ANOVA",  
 "p\_value" = 0.0001,  
 "Reason" = "Age significantly varies across different injury types, highlighting how specific age groups are more prone to certain types of injuries."  
 ),  
 `Age vs Care.Site` = list(  
 "Test" = "ANOVA",  
 "p\_value" = 0.0151,  
 "Reason" = "Age is significantly associated with the choice of care site, revealing healthcare preferences or requirements across age groups."  
 )  
)  
  
# Print significant results with reasons  
cat("Selected Variables and Reasons for the Project:\n")

## Selected Variables and Reasons for the Project:

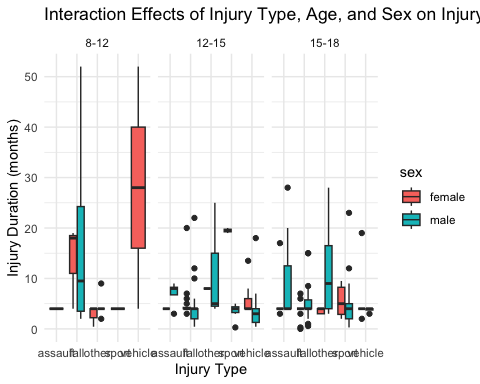
for (result\_name in names(significant\_results)) {  
 cat("\n", result\_name, "\n")  
 cat(" Test:", significant\_results[[result\_name]]$Test, "\n")  
 cat(" p-value:", significant\_results[[result\_name]]$p\_value, "\n")  
 cat(" Reason:", significant\_results[[result\_name]]$Reason, "\n")  
}

##   
## Injury\_Duration ~ injury\_type   
## Test: ANOVA   
## p-value: 0.038   
## Reason: Injury\_Duration is a key outcome variable. It significantly interacts with injury\_type, providing insights into how different injury types influence recovery time.   
##   
## Rating2 vs Injury\_Duration   
## Test: Pearson Correlation   
## p-value: 7.86e-05   
## Reason: Rating2 has the strongest correlation with Injury\_Duration, making it an excellent predictor for recovery time.   
##   
## injury\_type ~ sex   
## Test: Chi-square   
## p-value: 0.026   
## Reason: Injury\_type is associated with sex. Including this variable helps analyze gender-specific trends in injury patterns.   
##   
## Age vs injury\_type   
## Test: ANOVA   
## p-value: 1e-04   
## Reason: Age significantly varies across different injury types, highlighting how specific age groups are more prone to certain types of injuries.   
##   
## Age vs Care.Site   
## Test: ANOVA   
## p-value: 0.0151   
## Reason: Age is significantly associated with the choice of care site, revealing healthcare preferences or requirements across age groups.

# Data Analysis Project   
# Hypothesis 1: Injury Type predicts Injury Duration and is associated with demographic factors such as Age and Sex.  
  
# Age\_Group variable defined  
injury\_data$Age\_Group <- cut(  
 injury\_data$Age,   
 breaks = c(8, 12, 15, 18),   
 include.lowest = TRUE,   
 labels = c("8-12", "12-15", "15-18")  
)  
  
# Linear model with interaction terms for Injury Type, Age, and Sex  
lm\_model\_h1 <- lm(Injury\_Duration ~ injury\_type \* Age \* sex, data = injury\_data)  
  
# ANOVA to test main and interaction effects  
anova\_h1 <- anova(lm\_model\_h1)  
summary(lm\_model\_h1)

##   
## Call:  
## lm(formula = Injury\_Duration ~ injury\_type \* Age \* sex, data = injury\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.181 -3.141 -0.937 0.903 42.581   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.666667 36.957309 -0.126 0.8997   
## injury\_typefall 33.525706 38.323571 0.875 0.3828   
## injury\_typeother 6.756795 38.007470 0.178 0.8591   
## injury\_typesport 107.510417 53.323410 2.016 0.0452 \*  
## injury\_typevehicle 51.446979 39.595474 1.299 0.1955   
## Age 0.666667 2.368030 0.282 0.7786   
## sexmale -8.632525 41.342808 -0.209 0.8348   
## injury\_typefall:Age -2.232069 2.460620 -0.907 0.3655   
## injury\_typeother:Age -0.541716 2.447603 -0.221 0.8251   
## injury\_typesport:Age -6.354167 3.318864 -1.915 0.0571 .  
## injury\_typevehicle:Age -3.299920 2.557796 -1.290 0.1986   
## injury\_typefall:sexmale -1.547156 43.173617 -0.036 0.9715   
## injury\_typeother:sexmale 6.572426 42.799727 0.154 0.8781   
## injury\_typesport:sexmale -94.047095 57.346510 -1.640 0.1027   
## injury\_typevehicle:sexmale -22.842618 50.217057 -0.455 0.6497   
## Age:sexmale 0.726864 2.647028 0.275 0.7839   
## injury\_typefall:Age:sexmale -0.003283 2.771573 -0.001 0.9991   
## injury\_typeother:Age:sexmale -0.207792 2.757599 -0.075 0.9400   
## injury\_typesport:Age:sexmale 5.225295 3.577632 1.461 0.1459   
## injury\_typevehicle:Age:sexmale 1.148508 3.269707 0.351 0.7258   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.577 on 183 degrees of freedom  
## Multiple R-squared: 0.1773, Adjusted R-squared: 0.09184   
## F-statistic: 2.075 on 19 and 183 DF, p-value: 0.007228

# Visualization for interaction effects  
library(ggplot2)  
ggplot(injury\_data, aes(x = injury\_type, y = Injury\_Duration, fill = sex)) +  
 geom\_boxplot() +  
 facet\_wrap(~ Age\_Group) +  
 labs(  
 title = "Interaction Effects of Injury Type, Age, and Sex on Injury Duration",  
 x = "Injury Type",  
 y = "Injury Duration (months)"  
 ) +  
 theme\_minimal()



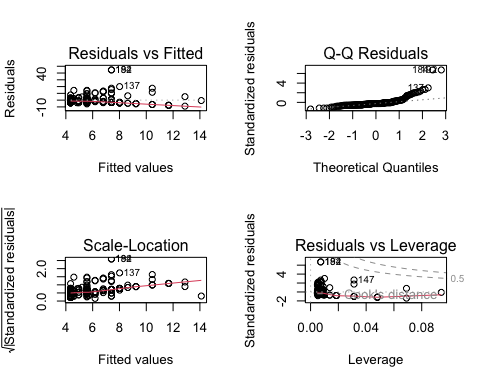
# Data Analysis Project   
  
# Hypothesis 2: Rating2 significantly predicts Injury Duration  
  
# Step 1: Fit a linear regression model  
lm\_model\_h2 <- lm(Injury\_Duration ~ Rating2, data = injury\_data)  
  
# Step 2: Summarize the model  
summary\_h2 <- summary(lm\_model\_h2)  
  
# Print the summary  
print(summary\_h2)

##   
## Call:  
## lm(formula = Injury\_Duration ~ Rating2, data = injury\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.886 -3.400 -1.571 0.124 44.600   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 16.54336 2.63386 6.281 2.04e-09 \*\*\*  
## Rating2 -0.12191 0.03024 -4.032 7.86e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.655 on 201 degrees of freedom  
## Multiple R-squared: 0.07481, Adjusted R-squared: 0.07021   
## F-statistic: 16.25 on 1 and 201 DF, p-value: 7.862e-05

# Step 3: Pearson Correlation Test  
cor\_test\_h2 <- cor.test(injury\_data$Rating2, injury\_data$Injury\_Duration, method = "pearson")  
  
# Print correlation results  
print(cor\_test\_h2)

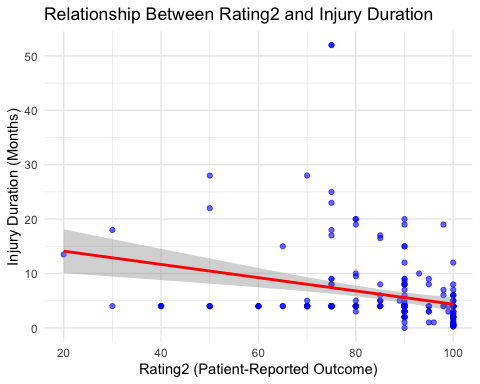
##   
## Pearson's product-moment correlation  
##   
## data: injury\_data$Rating2 and injury\_data$Injury\_Duration  
## t = -4.0316, df = 201, p-value = 7.862e-05  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.3963026 -0.1411259  
## sample estimates:  
## cor   
## -0.2735201

# Step 4: Diagnostic Plots for Model Validation  
par(mfrow = c(2, 2)) # Set up a 2x2 plotting area  
plot(lm\_model\_h2)



# Step 5: Visualization of Rating2 and Injury Duration Relationship  
library(ggplot2)  
ggplot(injury\_data, aes(x = Rating2, y = Injury\_Duration)) +  
 geom\_point(color = "blue", alpha = 0.6) + # Scatter plot of data points  
 geom\_smooth(method = "lm", color = "red", se = TRUE) + # Add regression line  
 labs(  
 title = "Relationship Between Rating2 and Injury Duration",  
 x = "Rating2 (Patient-Reported Outcome)",  
 y = "Injury Duration (Months)"  
 ) +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



# Step 6: Export Regression Results (Optional)  
library(broom)

## Warning: package 'broom' was built under R version 4.3.3

lm\_table\_h2 <- tidy(lm\_model\_h2) # Create tidy table of results  
write.csv(lm\_table\_h2, "Regression\_Rating2\_vs\_Injury\_Duration.csv") # Save results as CSV

# Data Analysis Project   
# Hypothesis 3: Age influences Care Site choice, with younger age groups more likely to visit emergency care.  
  
# Step 1: Recode Care Site into a binary variable (Emergency = 1, Other/PrimaryCare = 0)  
injury\_data$Emergency\_Care <- ifelse(injury\_data$Care.Site == "Emergency", 1, 0)  
  
# Step 2: Create Age Groups  
injury\_data$Age\_Group <- cut(injury\_data$Age,   
 breaks = c(8, 12, 15, 18),   
 include.lowest = TRUE,   
 labels = c("8-12", "12-15", "15-18"))  
  
# Step 3: Fit a logistic regression model  
logit\_model\_h3 <- glm(Emergency\_Care ~ Age\_Group, data = injury\_data, family = "binomial")  
  
# Step 4: Summarize the model  
summary\_h3 <- summary(logit\_model\_h3)  
  
# Print the summary  
print(summary\_h3)

##   
## Call:  
## glm(formula = Emergency\_Care ~ Age\_Group, family = "binomial",   
## data = injury\_data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.67398 0.62915 2.661 0.0078 \*\*  
## Age\_Group12-15 -0.08004 0.69884 -0.115 0.9088   
## Age\_Group15-18 -1.23726 0.65957 -1.876 0.0607 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 242.90 on 202 degrees of freedom  
## Residual deviance: 229.85 on 200 degrees of freedom  
## AIC: 235.85  
##   
## Number of Fisher Scoring iterations: 4

# Step 5: Calculate Odds Ratios and Confidence Intervals  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

exp\_cis <- exp(cbind(OR = coef(logit\_model\_h3), confint(logit\_model\_h3)))

## Waiting for profiling to be done...

print(exp\_cis)

## OR 2.5 % 97.5 %  
## (Intercept) 5.3333333 1.77696660 22.9304942  
## Age\_Group12-15 0.9230769 0.19501748 3.2977960  
## Age\_Group15-18 0.2901786 0.06467699 0.9369719

# Step 6: Model Performance (ROC Curve)  
library(pROC)

## Warning: package 'pROC' was built under R version 4.3.3

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

predicted\_probs\_h3 <- predict(logit\_model\_h3, type = "response")  
roc\_curve\_h3 <- roc(injury\_data$Emergency\_Care, predicted\_probs\_h3)

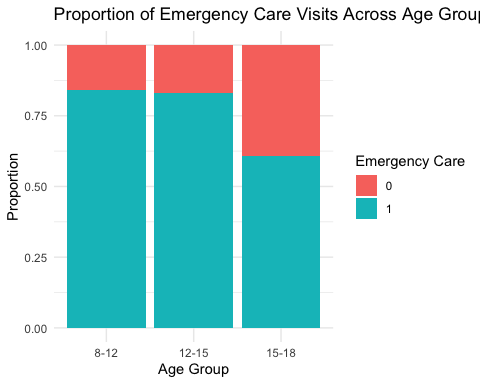
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

auc\_h3 <- auc(roc\_curve\_h3)  
  
# Print AUC  
print(auc\_h3)

## Area under the curve: 0.6389

# Step 7: Visualization of Age Group and Emergency Care Proportions  
library(ggplot2)  
ggplot(injury\_data, aes(x = Age\_Group, fill = factor(Emergency\_Care))) +  
 geom\_bar(position = "fill") +  
 labs(  
 title = "Proportion of Emergency Care Visits Across Age Groups",  
 x = "Age Group",  
 y = "Proportion",  
 fill = "Emergency Care"  
 ) +  
 theme\_minimal()



# Step 8: Plot ROC Curve  
plot(roc\_curve\_h3, col = "blue", main = "ROC Curve for Age Prediction of Emergency Care")

