Capstone Data Analysis in RStudio

2024-04-29

```
# View Structure
str(incident_data)

## tibble [132 × 7] (S3: tbl_df/tbl/data.frame)
##  $ Road          : chr [1:132] "MetroSouth" "MetroSuburb" "Port" "Port" ...
##  $ Claim         : chr [1:132] "MetroSouth_1213072" "MetroSuburb_1213069" "Port_1213067" "Port_1213065" ...
##  $ LossDate      : POSIXct[1:132], format: "2021-12-21" "2021-12-13" ...
##  $ ReportDate    : POSIXct[1:132], format: "2021-12-24" "2021-12-14" ...
##  $ Type          : chr [1:132] "Derailment" "Grade Crossing" "Property Damage" "Grade Crossing" ...
##  $ PrimaryCause  : chr [1:132] "Customer Track" "Third Party" "TBD - Under Investigation" "Third Party" ...
##  $ FRA_Reportable: num [1:132] 0 1 0 1 1 0 0 0 0 0 ...
```

```
# Summary of Dataset
summary(incident_data)

##      Road              Claim              LossDate                     
##  Length:132         Length:132         Min.   :2019-01-29 00:00:00.00  
##  Class :character   Class :character   1st Qu.:2020-07-28 00:00:00.00  
##  Mode  :character   Mode  :character   Median :2021-04-02 12:00:00.00  
##                                        Mean   :2021-01-07 18:10:54.54  
##                                        3rd Qu.:2021-08-18 12:00:00.00  
##                                        Max.   :2021-12-21 00:00:00.00  
##                                                                        
##    ReportDate                         Type           PrimaryCause      
##  Min.   :2019-01-30 00:00:00.00   Length:132         Length:132        
##  1st Qu.:2020-05-28 00:00:00.00   Class :character   Class :character  
##  Median :2021-05-21 00:00:00.00   Mode  :character   Mode  :character  
##  Mean   :2021-01-08 03:53:50.76                                        
##  3rd Qu.:2021-09-13 00:00:00.00                                        
##  Max.   :2021-12-24 00:00:00.00                                        
##  NA's   :15                                                            
##  FRA_Reportable  
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.2576  
##  3rd Qu.:1.0000  
##  Max.   :1.0000  
## 
```

```
# Frequency of Type
frequency_type <- table(incident_data$Type)
print(frequency_type)
```

```
##
## Crossing Signal Damage            Derailment                  Fire
##                     12                    48                     1
##         Grade Crossing                Injury         Motor Vehicle
##                     11                    23                     9
##                   NULL                 Other       Property Damage
##                      5                     4                    12
##    Run Through Switch            Trespasser      Vandalism/Theft
##                      3                     1                     3
```

```
# Frequency of Primary Cause
frequency_pc <- table(incident_data$PrimaryCause)
print(frequency_pc)
```

```
##
##            Act of Nature          Customer Caused           Customer Track
##                        3                        5                       13
##             Human Factor               Mechanical            Miscellaneous
##                       26                        5                       10
##                     NULL TBD - Under Investigation              Third Party
##                        5                        6                       40
##                    Track
##                       19
```

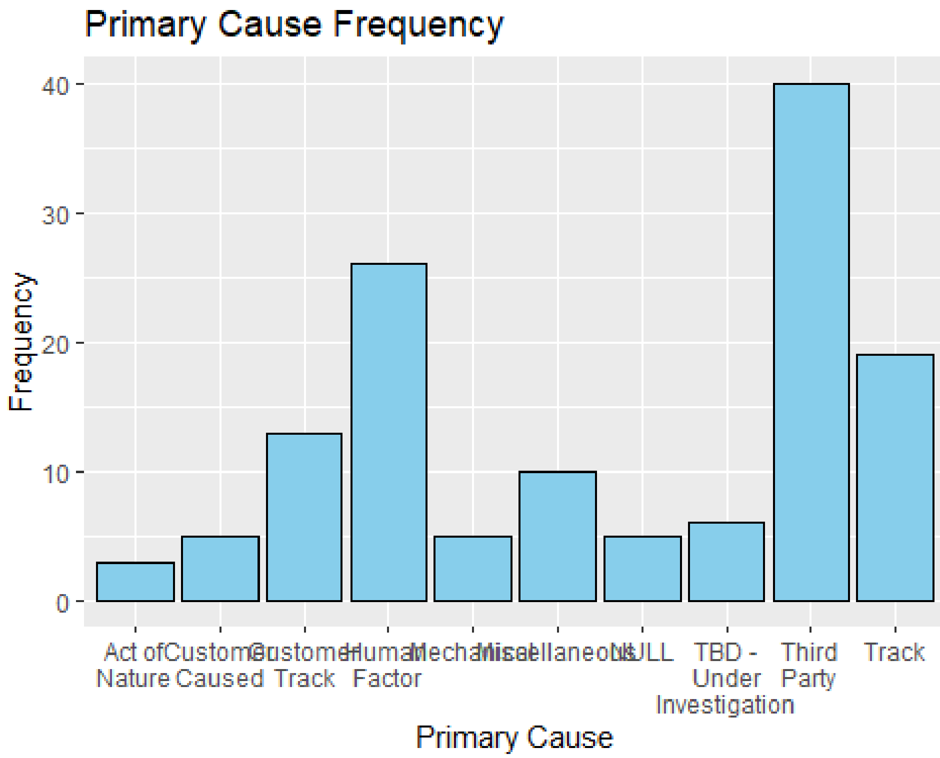
```
# Frequency of FRA Reportable
fra_labels <- ifelse(incident_data$FRA_Reportable == 0, "No", "Yes")
fra_frequency <- table(fra_labels)
print(fra_frequency)
```

```
## fra_labels
##  No Yes
##  98  34
```

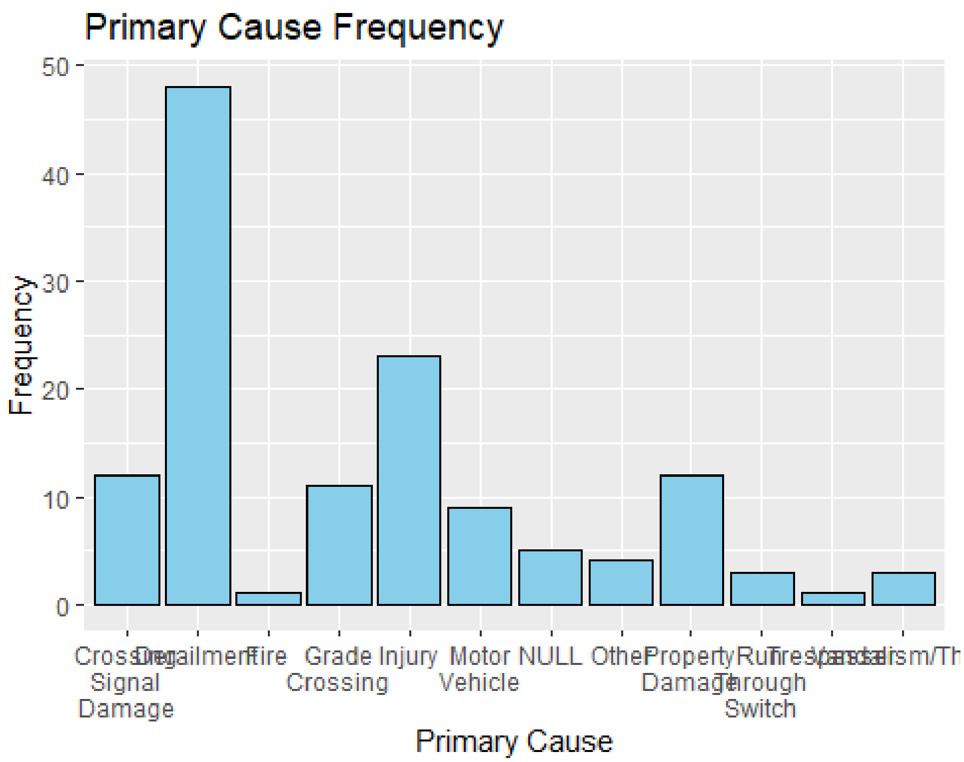
Frequency plots

```
# Primary Cause
frequency_pc <- table(incident_data$PrimaryCause)
frequency_pcdf <- as.data.frame(frequency_pc)



ggplot(frequency_pcdf, aes(x = Var1, y = Freq)) +
  geom_bar(
    stat = "identity",
    fill = "skyblue",
    color = "black"
```

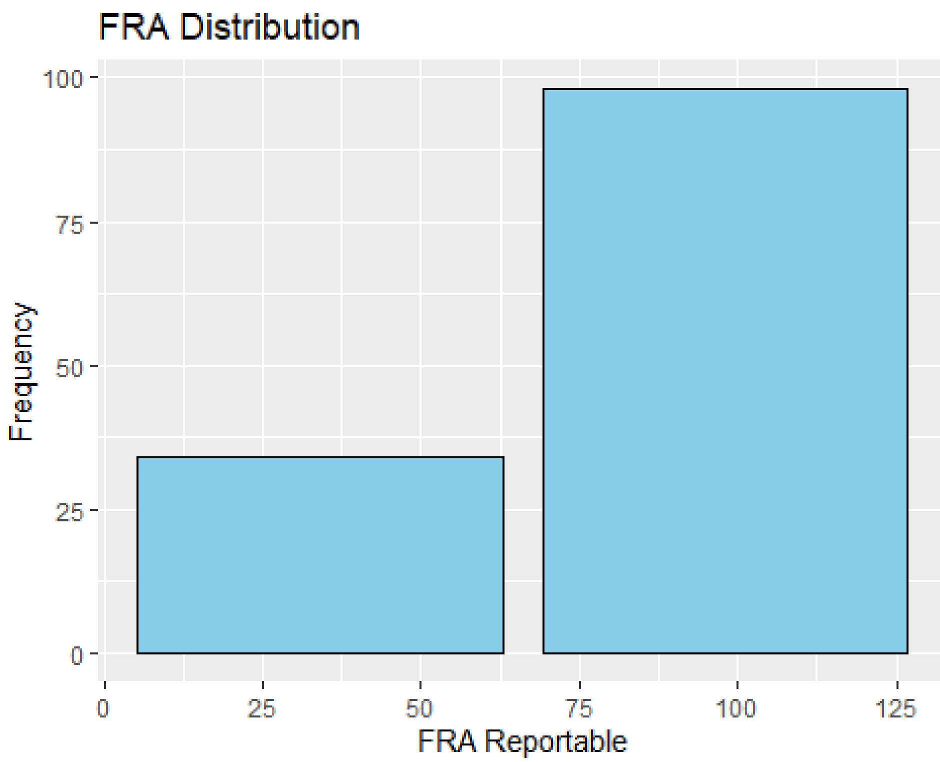
```r
    ) +
  labs(
    title = "Primary Cause Frequency",
      x = "Primary Cause",
      y = "Frequency"
    ) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 8))
```




```r
# Type
frequency_tdf <- as.data.frame(frequency_type)

ggplot(frequency_tdf, aes(x = Var1, y = Freq)) +
  geom_bar(
      stat = "identity",
      fill = "skyblue",
      color = "black"
      ) +
  labs(
    title = "Primary Cause Frequency",
      x = "Primary Cause",
      y = "Frequency"
    ) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 8))
```




```
# FRA Reportable
ggplot(as.data.frame(fra_frequency), aes(x = fra_frequency, y = Freq)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black")+
  labs(title = "FRA Distribution", x = "FRA Reportable", y = "Frequency")

## Don't know how to automatically pick scale for object of type <table>.
## Defaulting to continuous.
```

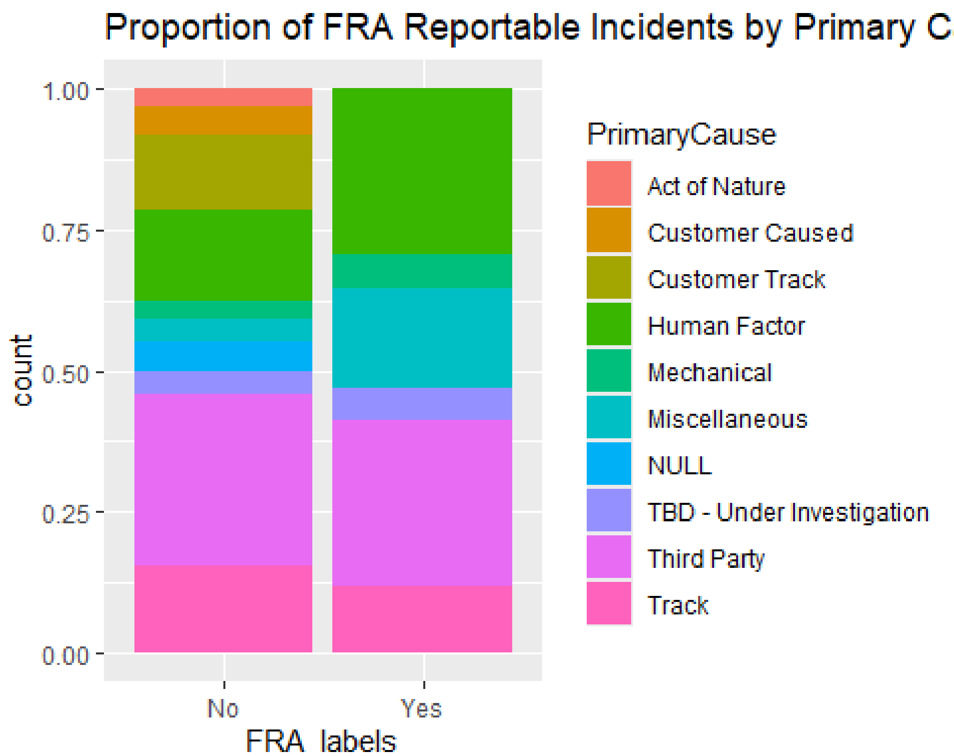


Methods and Findings

```
#cross-tabulation of Primary Cause and FRA Reportable
threshold <- 1
binary_vector <- as.integer(incident_data$FRA_Reportable >= threshold)
FRA_labels <- ifelse(binary_vector == 0,"No","Yes")
table(incident_data$PrimaryCause, FRA_labels)
```

```
##                            FRA_labels
##                             No Yes
##   Act of Nature              3   0
##   Customer Caused            5   0
##   Customer Track            13   0
##   Human Factor              16  10
##   Mechanical                 3   2
##   Miscellaneous              4   6
##   NULL                       5   0
##   TBD - Under Investigation  4   2
##   Third Party               30  10
##   Track                     15   4
```

```
#Visualize the relationship
ggplot(incident_data, aes(x = FRA_labels, fill = PrimaryCause)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of FRA Reportable Incidents by Primary Cause")
```




```
# Fit Logistic regression Model
logit_model <- glm(FRA_Reportable ~ PrimaryCause, data = incident_data, family = "binomial"
)

#summarize model
summary(logit_model)
```

```
##
## Call:
## glm(formula = FRA_Reportable ~ PrimaryCause, family = "binomial",
##     data = incident_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3537  -0.7585  -0.6876   1.0108   1.7653
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -1.857e+01  3.766e+03  -0.005    0.996
## PrimaryCauseCustomer Caused       6.330e-08  4.763e+03   0.000    1.000
## PrimaryCauseCustomer Track        6.319e-08  4.178e+03   0.000    1.000
## PrimaryCauseHuman Factor          1.810e+01  3.766e+03   0.005    0.996
## PrimaryCauseMechanical            1.816e+01  3.766e+03   0.005    0.996
## PrimaryCauseMiscellaneous         1.897e+01  3.766e+03   0.005    0.996
## PrimaryCauseNULL                  6.317e-08  4.763e+03   0.000    1.000
```

```
## PrimaryCauseTBD - Under Investigation  1.787e+01  3.766e+03   0.005    0.996
## PrimaryCauseThird Party               1.747e+01  3.766e+03   0.005    0.996
## PrimaryCauseTrack                     1.724e+01  3.766e+03   0.005    0.996
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 150.61  on 131  degrees of freedom
## Residual deviance: 127.02  on 122  degrees of freedom
## AIC: 147.02
##
## Number of Fisher Scoring iterations: 17
```

```
#Interpret the coefficients
exp(coef(logit_model))  # Exponentiated coefficients as odds ratios
```

```
##                         (Intercept)          PrimaryCauseCustomer Caused
##                        8.646869e-09                         1.000000e+00
##           PrimaryCauseCustomer Track             PrimaryCauseHuman Factor
##                        1.000000e+00                         7.228050e+07
##              PrimaryCauseMechanical           PrimaryCauseMiscellaneous
##                        7.709920e+07                         1.734732e+08
##                    PrimaryCauseNULL PrimaryCauseTBD - Under Investigation
##                        1.000000e+00                         5.782440e+07
##             PrimaryCauseThird Party                    PrimaryCauseTrack
##                        3.854960e+07                         3.083968e+07
```

```
# Predict probabilities for specific scenarios (e.g., "human factors" as PrimaryCause)
predict(logit_model, newdata = data.frame(PrimaryCause = "Human Factor"), type = "response
")
```

```
##         1
## 0.3846154
```

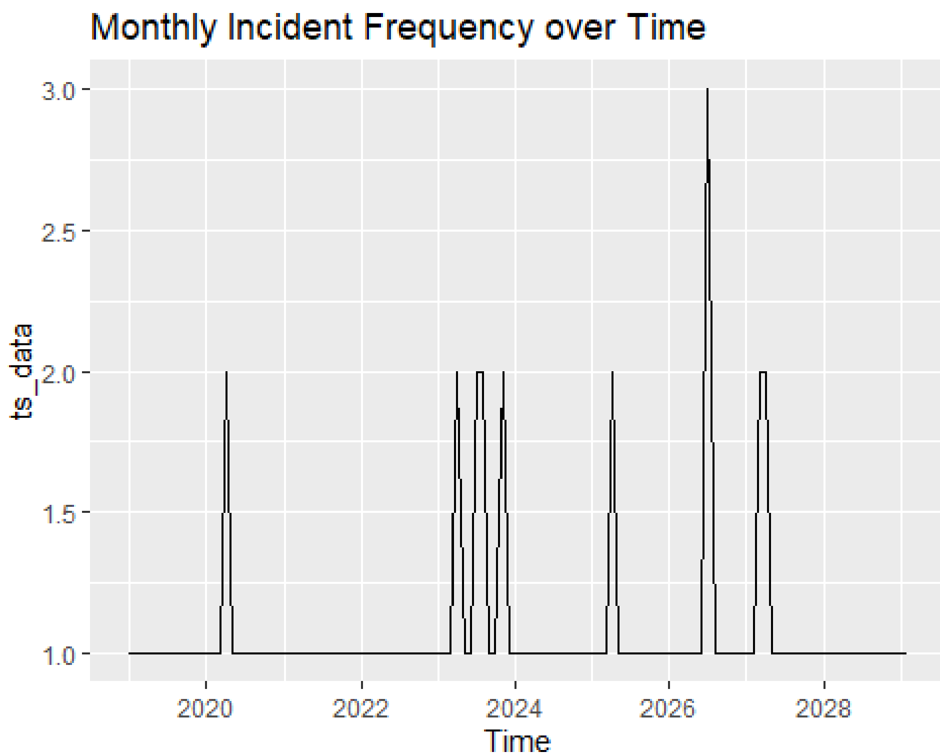
```
#Create Contingency Table
contingency_table <- table(incident_data$PrimaryCause,incident_data$FRA_Reportable)

#Chi-Square test
chisq_test <- chisq.test(contingency_table)
```

```
## Warning in chisq.test(contingency_table): Chi-squared approximation may be
## incorrect
```

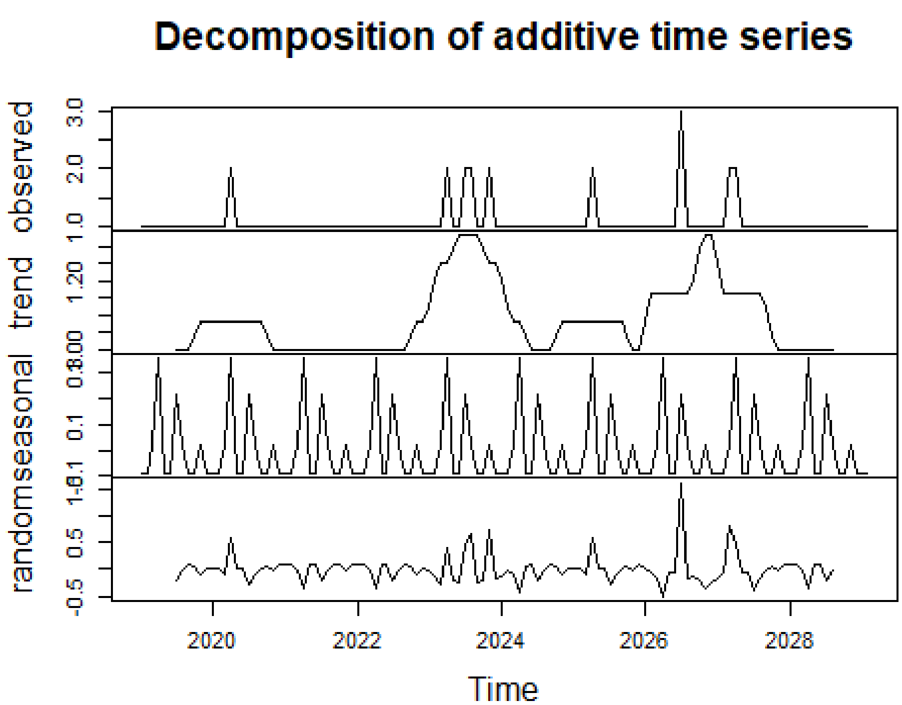
```
chisq_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 18.289, df = 9, p-value = 0.03197
```

```r
# calculate Cramer's V for effect size
cramer_v <- sqrt(chisq_test$statistic / sum(contingency_table))
cramer_v

## X-squared
## 0.3722237

#aggregate incidents by month
monthlycounts <- incident_data %>%
  group_by(LossDate) %>%
  summarise(IncidentCount = n()) %>%
  ungroup()

#create time series


ts_data <- ts(monthlycounts$IncidentCount, frequency = 12, start = c(
  year(
    min
       (monthlycounts$LossDate)),
  month(min
        (monthlycounts$LossDate))
  ))

# Plot the timeseries
autoplot(ts_data) +
  labs(title = "Monthly Incident Frequency over Time")
```




```r
# Decompose the time series to analyze trend, seasonality, and noise
decom <- decompose(ts_data)
plot(decom)
```



```
# Compute Incident Frequency
incident_frequency <- incident_data %>%
  group_by(Claim) %>%
  summarise(IncidentCount = n_distinct(Claim))
print(incident_frequency)
```

```
## # A tibble: 132 × 2
##    Claim              IncidentCount
##    <chr>                      <int>
##  1 MetroNorth_1190349             1
##  2 MetroNorth_1192684             1
##  3 MetroNorth_1212903             1
##  4 MetroNorth_1212916             1
##  5 MetroNorth_1212926             1
##  6 MetroNorth_1212931             1
##  7 MetroNorth_1213014             1
##  8 MetroNorth_1213020             1
##  9 MetroSouth_1190321             1
## 10 MetroSouth_1190327             1
## # ℹ 122 more rows
```

```
# Sort data by Claim and LossDate
incidents_data <- incident_data[order(incident_data$Claim, incident_data$LossDate), ]

# Calculate time difference between consecutive incidents for each Claim
incidents_data$TimeToNextIncident <- c(NA, diff(incidents_data$LossDate))

# Define a binary indicator for subsequent incidents within a certain timeframe
incidents_data$SubsequentIncident <- ifelse(incidents_data$TimeToNextIncident < 30, 1, 0)

# Merge data
merged_data <- merge(incident_frequency, incidents_data, by = "Claim", all.x = TRUE)

# Check the structure of merged_data
str(merged_data)
```

```
## 'data.frame':    132 obs. of  10 variables:
##  $ Claim             : chr  "MetroNorth_1190349" "MetroNorth_1192684" "MetroNorth_121290
3" "MetroNorth_1212916" ...
##  $ IncidentCount     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Road              : chr  "MetroNorth" "MetroNorth" "MetroNorth" "MetroNorth" ...
##  $ LossDate          : POSIXct, format: "2019-10-11" "2019-12-03" ...
##  $ ReportDate        : POSIXct, format: "2019-10-11" "2019-12-04" ...
##  $ Type              : chr  "NULL" "Derailment" "Vandalism/Theft" "Derailment" ...
##  $ PrimaryCause      : chr  "NULL" "Customer Track" "Third Party" "Track" ...
##  $ FRA_Reportable    : num  0 0 0 1 0 1 0 1 0 0 ...
```

```
##  $ TimeToNextIncident: num  NA 53 455 15 20 5 146 10 -745 67 ...
##  $ SubsequentIncident: num  NA 0 0 1 1 1 0 1 1 0 ...
```

```r
# Handle missing values (if necessary)
merged_data1 <- na.omit(merged_data[, c("IncidentCount", "SubsequentIncident")])
str(merged_data1)
```

```
## 'data.frame':    131 obs. of  2 variables:
##  $ IncidentCount     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ SubsequentIncident: num  0 0 1 1 1 0 1 1 0 1 ...
##  - attr(*, "na.action")= 'omit' Named int 1
##   ..- attr(*, "names")= chr "1"
```

```r
# Calculate correlation between IncidentCount and SubsequentIncidents
correlation_result <- cor(merged_data1$IncidentCount, merged_data1$SubsequentIncident)
```

```
## Warning in cor(merged_data1$IncidentCount, merged_data1$SubsequentIncident):
## the standard deviation is zero
```

```r
# Print correlation coefficient
print(correlation_result)
```

```
## [1] NA
```

```r
# Survival Analysis
surv_object <- Surv(time = incidents_data$TimeToNextIncident, event = incidents_data$Subseq
uentIncident)

# Fit survival model
cox_model <- coxph(surv_object ~ IncidentCount, data = merged_data)

# ANOVA Model
model_anova <- aov(IncidentCount ~ Road, data = merged_data)

# Results
summary(model_anova)
```

```
##              Df    Sum Sq   Mean Sq F value  Pr(>F)
## Road          5 7.640e-29 1.528e-29   3.382 0.00669 **
## Residuals   126 5.695e-28 4.520e-30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Post-hoc analysis
posthoc<- TukeyHSD(model_anova)
print(posthoc)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
```

```
##
## Fit: aov(formula = IncidentCount ~ Road, data = merged_data)
##
## $Road
##                              diff           lwr           upr     p adj
## MetroSouth-MetroNorth  -3.108624e-15 -5.556784e-15 -6.604650e-16 0.0046320
## MetroSuburb-MetroNorth -3.108624e-15 -5.565654e-15 -6.515948e-16 0.0048449
## Port-MetroNorth        -3.108624e-15 -5.526276e-15 -6.909729e-16 0.0039543
## Rural-MetroNorth       -3.108624e-15 -5.916856e-15 -3.003930e-16 0.0208285
## Suburb-MetroNorth      -3.108624e-15 -5.701691e-15 -5.155582e-16 0.0091345
## MetroSuburb-MetroSouth  0.000000e+00 -1.602212e-15  1.602212e-15 1.0000000
## Port-MetroSouth         0.000000e+00 -1.541145e-15  1.541145e-15 1.0000000
## Rural-MetroSouth        0.000000e+00 -2.101488e-15  2.101488e-15 1.0000000
## Suburb-MetroSouth       0.000000e+00 -1.803907e-15  1.803907e-15 1.0000000
## Port-MetroSuburb        0.000000e+00 -1.555197e-15  1.555197e-15 1.0000000
## Rural-MetroSuburb       0.000000e+00 -2.111815e-15  2.111815e-15 1.0000000
## Suburb-MetroSuburb      0.000000e+00 -1.815927e-15  1.815927e-15 1.0000000
## Rural-Port              0.000000e+00 -2.065867e-15  2.065867e-15 1.0000000
## Suburb-Port             0.000000e+00 -1.762281e-15  1.762281e-15 1.0000000
## Suburb-Rural            0.000000e+00 -2.268647e-15  2.268647e-15 1.0000000
```

```r
# Kruskal-Wallis Test
kruskal_test <- kruskal.test(IncidentCount ~ Road, data = merged_data)
print(kruskal_test)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  IncidentCount by Road
## Kruskal-Wallis chi-squared = NaN, df = 5, p-value = NA
```