Capstone Data Analysis

2024-04-29

## Rstudio Setup

# 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)

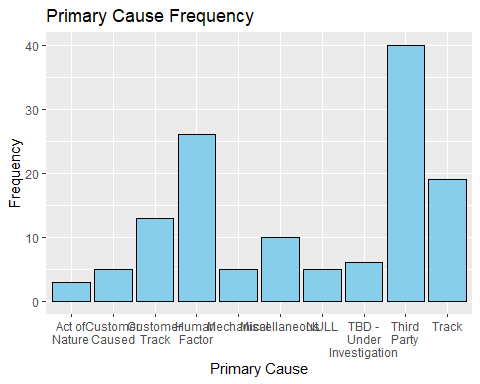
##   
## Act of Nature Customer Caused Customer Track   
## 3 5 13   
## Human Factor Mechanical Miscellaneous   
## 26 5 10   
## NULL TBD - Under Investigation Third Party   
## 5 6 40   
## Track   
## 19

# Frequency of FRA Reportable  
fra\_labels <- ifelse(incident\_data$FRA\_Reportable == 0, "No", "Yes")  
fra\_frequency <- table(fra\_labels)  
print(fra\_frequency)

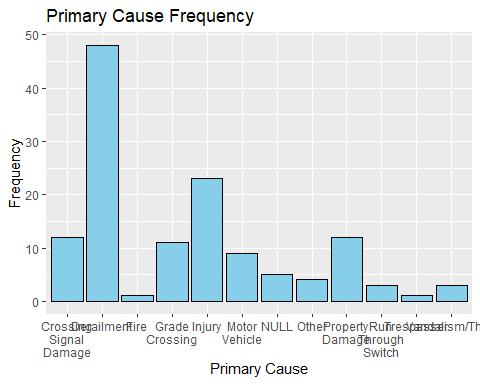
## fra\_labels  
## No Yes   
## 98 34

## 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"  
 ) +  
 labs(  
 title = "Primary Cause Frequency",  
 x = "Primary Cause",  
 y = "Frequency"  
 ) +  
 scale\_x\_discrete(labels = function(x) str\_wrap(x, width = 8))

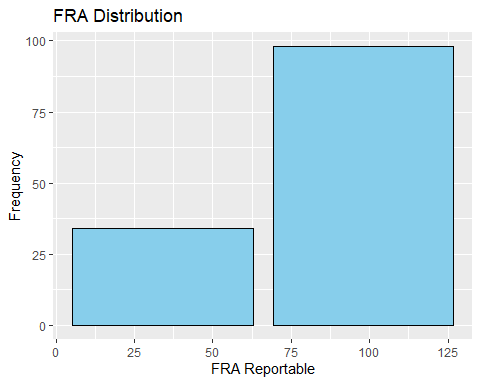


# Type  
frequency\_tdf <- as.data.frame(frequency\_type)  
  
ggplot(frequency\_tdf, aes(x = Var1, y = Freq)) +  
 geom\_bar(  
 stat = "identity",   
 fill = "skyblue",  
 color = "black"  
 ) +  
 labs(  
 title = "Primary Cause Frequency",  
 x = "Primary Cause",  
 y = "Frequency"  
 ) +  
 scale\_x\_discrete(labels = 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")

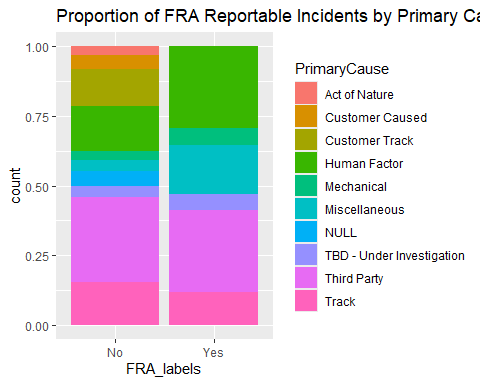
## Don't know how to automatically pick scale for object of type <table>.  
## Defaulting to continuous.

 ## 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

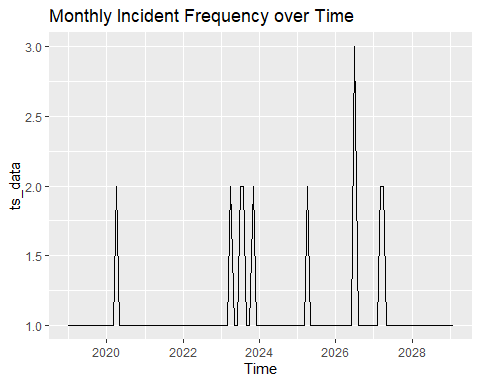
chisq\_test

##   
## Pearson's Chi-squared test  
##   
## data: contingency\_table  
## X-squared = 18.289, df = 9, p-value = 0.03197

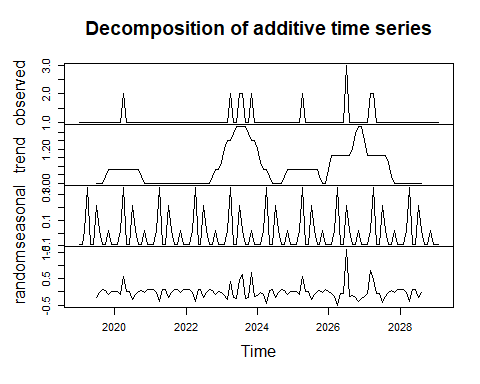
# calculate Cramer's V for effect size  
cramer\_v <- sqrt(chisq\_test$statistic / sum(contingency\_table))  
cramer\_v

## 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")



# 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\_1212903" "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 ...

# 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"

# 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

# Print correlation coefficient  
print(correlation\_result)

## [1] NA

# Survival Analysis  
surv\_object <- Surv(time = incidents\_data$TimeToNextIncident, event = incidents\_data$SubsequentIncident)  
  
# 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

#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

# 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