BUSH FIRE DAMAGE ANALYSIS REPORT

2024-2025

Prepared by:

William Nguyen

Introduction

In recent years, the frequency and severity of bushfires across Australia have increased significantly, posing substantial risks to communities, infrastructure, and economic assets. For insurance providers, understanding the patterns and impact of bushfire damage is critical for accurate risk assessment, premium pricing, and efficient claims management. This report presents a comprehensive data analysis of bushfire incidents, with a focus on the extent of property damage, geographical distribution, and historical trends. The aim is to provide actionable insights that support strategic decision-making, enhance risk modeling, and improve the company's preparedness and response to future bushfire events.

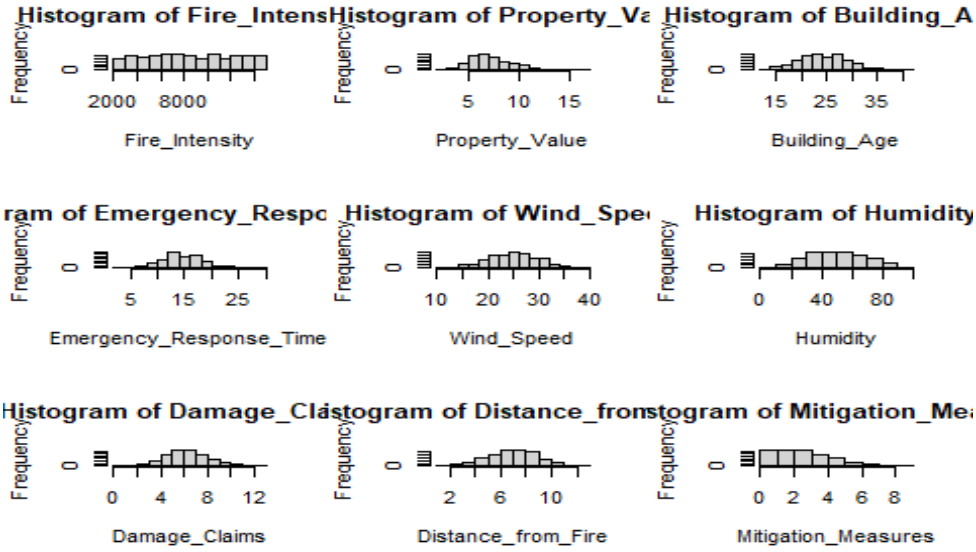
Explore data structure and encode variables

```
#Explore data structure
str(BFdata)

## 'data.frame':    600 obs. of  13 variables:
##  $ ClaimID               : chr  "DQTA26862" "RDAQ32534" "WTLA41817"
"YHZL16317" ...
##  $ Fire_Intensity        : num  5738 12248 7317 13479 14226 ...
##  $ Distance_from_Fire    : num  5.57 5.49 5.12 4.89 6.13 ...
##  $ Building_Age          : int  19 27 16 30 21 27 24 24 36 30 ...
##  $ Property_Value        : num  9.76 5.89 3.98 4.91 9.31 ...
##  $ Population_Density    : int  520 433 634 576 644 476 500 591 487 441
...
##  $ Emergency_Response_Time: int  16 18 17 14 12 10 16 19 15 18 ...
##  $ Mitigation_Measures   : int  3 3 4 2 2 1 5 7 2 3 ...
##  $ Construction_Quality  : chr  "Good" "Good" "Good" "Good" ...
##  $ Insurance_Coverage    : chr  "Fully" "None" "Partially" "Partially"
...
##  $ Wind_Speed            : num  24.9 21 25.3 22.3 37.2 31.8 27.8 22.5
32.2 20.6 ...
##  $ Humidity              : num  35.3 28.2 44.5 53.8 73.8 ...
##  $ Damage_Claims         : num  7.02 7.45 4.3 7.78 9.46 4.54 7.14 7.73
6.03 5.37 ...

summary(BFdata)

##    ClaimID          Fire_Intensity  Distance_from_Fire  Building_Age
##  Length:600         Min.   : 2006   Min.   : 1.380     Min.   :12.00
##  Class :character   1st Qu.: 5379   1st Qu.: 5.710     1st Qu.:21.00
##  Mode  :character   Median : 8272   Median : 7.090     Median :25.00
##                     Mean   : 8490   Mean   : 7.026     Mean   :24.88
##                     3rd Qu.:11708   3rd Qu.: 8.410     3rd Qu.:28.00
##                     Max.   :14992   Max.   :12.380     Max.   :42.00
##  Property_Value   Population_Density Emergency_Response_Time
##  Min.   : 2.800   Min.   :372.0      Min.   : 2.00
##  1st Qu.: 5.737   1st Qu.:546.0      1st Qu.:13.00
##  Median : 6.920   Median :597.5      Median :15.00
```

```
##  Mean   : 7.278   Mean   :598.3      Mean   :14.97
##  3rd Qu.: 8.590   3rd Qu.:652.2      3rd Qu.:18.00
##  Max.   :16.290   Max.   :842.0      Max.   :29.00
##  Mitigation_Measures Construction_Quality Insurance_Coverage   Wind_Speed
##  Min.   :0.000       Length:600           Length:600         Min.   :10.00
##  1st Qu.:2.000       Class :character     Class :character   1st Qu.:21.40
##  Median :3.000       Mode  :character     Mode  :character   Median :24.95
##  Mean   :2.978                                               Mean   :24.99
##  3rd Qu.:4.000                                               3rd Qu.:28.52
##  Max.   :9.000                                               Max.   :39.50
##     Humidity     Damage_Claims
##  Min.   : 5.46   Min.   : 0.100
##  1st Qu.:35.54   1st Qu.: 4.978
##  Median :49.20   Median : 6.180
##  Mean   :49.77   Mean   : 6.250
##  3rd Qu.:63.83   3rd Qu.: 7.362
##  Max.   :98.63   Max.   :12.370

#Encoding construction quality Good and Bad to 1 and 0
BFdata$Construction_Quality = ifelse(BFdata$Construction_Quality ==
"Good",1,0)

#Encoding insurance coverage: 0 for none, 1 for partially and 2 for fully
IC_factor = factor(BFdata$Insurance_Coverage, levels = c("None", "Partially",
"Fully"))
BFdata$Insurance_Coverage = as.integer(IC_factor) - 1
```

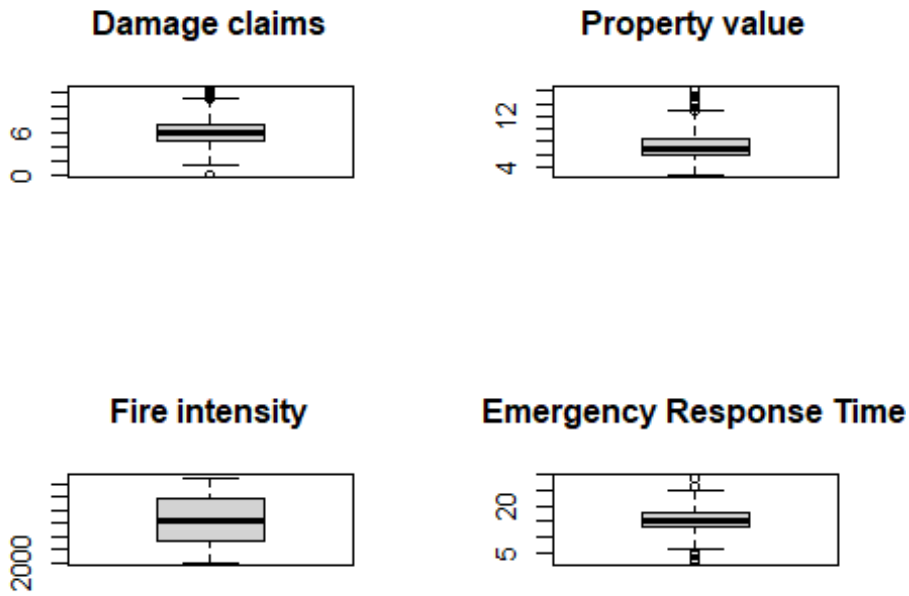
The dataset comprises 600 observations across 13 variables, capturing key information related to bushfire incidents and the resulting insurance claims. Each record represents a unique insurance claim and includes factors such as fire intensity, distance from fire, building age, property value, population density, emergency response time, mitigation efforts, construction quality, insurance coverage, and weather conditions (wind speed and humidity). The target variable is Damage_Claims, representing the extent of damage reported.

Exploratory Data Analysis



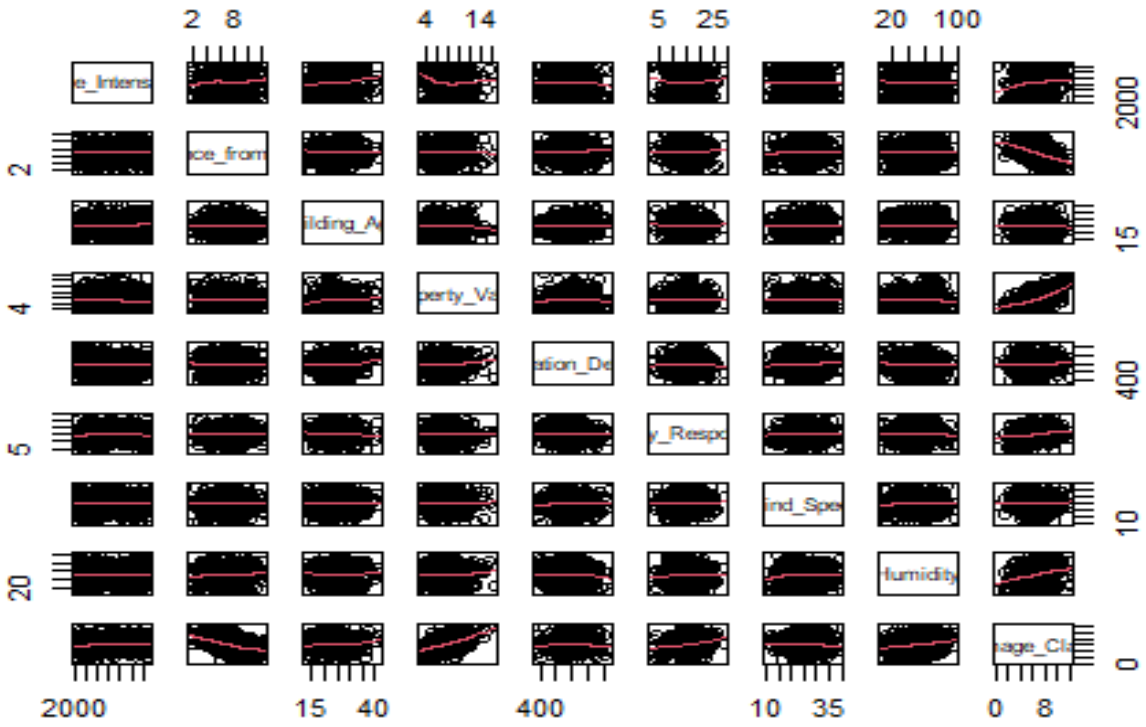
Most variables exhibit an approximately normal distribution, suggesting a balanced spread across the dataset. However, fire intensity appears uniformly distributed, indicating that claims are associated with a wide range of fire severities rather than being concentrated around a typical value. Additionally, mitigation measures are generally low, as indicated by the left-skewed distribution. This suggests that limited proactive steps were taken to reduce fire risk before the incidents occurred.

Detect outliers in key variables



The boxplots reveal key patterns in the data. Damage claims are mostly consistent but show a few high outliers, indicating occasional severe losses. Property value has multiple upper outliers, suggesting a small number of high-value properties in the dataset. Fire intensity appears evenly spread with no significant outliers, reflecting consistent variation across claims. Emergency response time is mostly uniform but includes both low and high outliers, highlighting variability in response effectiveness.

```
#Correlation matrix for key numerical variables to explore the relationship
between them
cor_matrix =
BFdata[,c("Fire_Intensity","Distance_from_Fire","Building_Age","Property_Valu
e","Population_Density","Emergency_Response_Time","Wind_Speed","Humidity","Da
mage_Claims")]
cor(cor_matrix)
```



The correlation matrix reveals a slight positive correlation between fire intensity and property value with damage claims, which aligns with expectations—higher fire intensity and more valuable properties tend to result in greater losses. Interestingly, humidity shows a strong positive correlation with damage claims, which is unexpected and warrants further investigation. In contrast, distance from fire has a negative correlation with damage claims, suggesting that properties located further from the fire source tend to suffer less damage. Other variables do not exhibit strong correlations with damage claims or with each other.

Build and evaluate multiple linear regression model

```
#Build multiple linear regression
m1 = lm(Damage_Claims~.,data = training)
summary(m1)

##
## Call:
## lm(formula = Damage_Claims ~ ., data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.95324 -0.72569  0.01252  0.72947  2.73626
##
## Coefficients:
```

```
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.270e+00  8.074e-01   2.811  0.00527 **
## Fire_Intensity             9.220e-05  1.657e-05   5.564 6.04e-08 ***
## Distance_from_Fire        -5.117e-01  3.155e-02 -16.221  < 2e-16 ***
## Building_Age              -7.918e-03  1.221e-02  -0.648  0.51721
## Property_Value             5.032e-01  2.856e-02  17.617  < 2e-16 ***
## Population_Density         5.097e-04  7.512e-04   0.679  0.49800
## Emergency_Response_Time    1.019e-01  1.499e-02   6.800 6.05e-11 ***
## Mitigation_Measures        1.062e-01  3.705e-02   2.865  0.00448 **
## Construction_Quality      -2.450e-01  1.600e-01  -1.532  0.12672
## Insurance_Coverage         1.417e-02  8.610e-02   0.165  0.86940
## Wind_Speed                -9.135e-03  1.211e-02  -0.754  0.45128
## Humidity                   3.124e-02  3.333e-03   9.374  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.027 on 288 degrees of freedom
## Multiple R-squared:  0.7214, Adjusted R-squared:  0.7108
## F-statistic:  67.8 on 11 and 288 DF,  p-value: < 2.2e-16
```

```
#New model with only significant variables
m2 = lm(Damage_Claims~Fire_Intensity + Distance_from_Fire + Property_Value +
Emergency_Response_Time + Mitigation_Measures + Humidity, data = training)
summary(m2)
```

```
##
## Call:
## lm(formula = Damage_Claims ~ Fire_Intensity + Distance_from_Fire +
##     Property_Value + Emergency_Response_Time + Mitigation_Measures +
##     Humidity, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64443 -0.66589  0.00145  0.68283  2.68320
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.940e+00  4.560e-01   4.256 2.81e-05 ***
## Fire_Intensity             9.092e-05  1.632e-05   5.569 5.78e-08 ***
## Distance_from_Fire        -5.074e-01  3.095e-02 -16.394  < 2e-16 ***
## Property_Value             5.038e-01  2.846e-02  17.701  < 2e-16 ***
## Emergency_Response_Time    9.911e-02  1.478e-02   6.707 1.02e-10 ***
## Mitigation_Measures        1.138e-01  3.659e-02   3.109  0.00206 **
## Humidity                   3.159e-02  3.310e-03   9.542  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.026 on 293 degrees of freedom
## Multiple R-squared:  0.7172, Adjusted R-squared:  0.7114
## F-statistic: 123.8 on 6 and 293 DF,  p-value: < 2.2e-16
```

```r
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: Damage_Claims
##                           Df  Sum Sq Mean Sq F value    Pr(>F)
## Fire_Intensity             1  14.579  14.579  13.846 0.0002379 ***
## Distance_from_Fire         1 296.739 296.739 281.802 < 2.2e-16 ***
## Property_Value             1 308.056 308.056 292.550 < 2.2e-16 ***
## Emergency_Response_Time    1  52.895  52.895  50.233 1.022e-11 ***
## Mitigation_Measures        1  14.143  14.143  13.431 0.0002938 ***
## Humidity                   1  95.881  95.881  91.055 < 2.2e-16 ***
## Residuals                293 308.530   1.053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Checking correlation between pairs of variable
newdata = BFdata[,-1]
cor(training)
```

```
##                          Fire_Intensity Distance_from_Fire  Building_Age
## Fire_Intensity              1.00000000       -0.035130436  0.1485138507
## Distance_from_Fire         -0.03513044        1.000000000 -0.0161134463
## Building_Age                0.14851385       -0.016113446  1.0000000000
## Property_Value             -0.17214426       -0.023820894 -0.0171452193
## Population_Density         -0.02538997        0.086787272 -0.0692060492
## Emergency_Response_Time     0.05863044        0.044657527 -0.0009148271
## Mitigation_Measures         0.01526527       -0.026643380 -0.0479332316
## Construction_Quality       -0.03832640       -0.046775779 -0.0463700387
## Insurance_Coverage         -0.06565339        0.137742116 -0.0555359597
## Wind_Speed                  0.03641542        0.055297007 -0.0792001844
## Humidity                    0.01384675       -0.006683554 -0.0584046267
## Damage_Claims               0.11560952       -0.525306240 -0.0147362895
##                          Property_Value Population_Density
## Fire_Intensity             -0.17214426       -0.025389972
## Distance_from_Fire         -0.02382089        0.086787272
## Building_Age               -0.01714522       -0.069206049
## Property_Value              1.00000000        0.025901702
## Population_Density          0.02590170        1.000000000
## Emergency_Response_Time    -0.04334720       -0.087718406
## Mitigation_Measures         0.02048197        0.063686144
## Construction_Quality        0.01549979       -0.102322819
## Insurance_Coverage          0.07181407        0.038976909
## Wind_Speed                 -0.02831214        0.003703872
## Humidity                   -0.05192790       -0.018956106
## Damage_Claims               0.51893206       -0.025465268
##                          Emergency_Response_Time Mitigation_Measures
## Fire_Intensity                      0.0586304446          0.01526527
## Distance_from_Fire                  0.0446575271         -0.02664338
## Building_Age                       -0.0009148271         -0.04793323
```

```
## Property_Value                     -0.0433472014          0.02048197
## Population_Density                 -0.0877184057          0.06368614
## Emergency_Response_Time             1.0000000000         -0.06621487
## Mitigation_Measures                -0.0662148722          1.00000000
## Construction_Quality                0.0943203397         -0.05027369
## Insurance_Coverage                 -0.0702711129         -0.05920259
## Wind_Speed                         -0.0281850915         -0.09958924
## Humidity                            0.0601163065          0.05264768
## Damage_Claims                       0.1846742970          0.12658484
##                         Construction_Quality Insurance_Coverage
Wind_Speed
## Fire_Intensity                   -0.03832640       -0.065653389
0.036415422
## Distance_from_Fire               -0.04677578        0.137742116
0.055297007
## Building_Age                     -0.04637004       -0.055535960 -
0.079200184
## Property_Value                    0.01549979        0.071814067 -
0.028312138
## Population_Density               -0.10232282        0.038976909
0.003703872
## Emergency_Response_Time           0.09432034       -0.070271113 -
0.028185092
## Mitigation_Measures              -0.05027369       -0.059202594 -
0.099589243
## Construction_Quality              1.00000000        0.023503225
0.144957276
## Insurance_Coverage                0.02350322        1.000000000 -
0.056819283
## Wind_Speed                        0.14495728       -0.056819283
1.000000000
## Humidity                         -0.06439629        0.002911288
0.034991268
## Damage_Claims                    -0.03088254       -0.054728683 -
0.072401102
##                            Humidity Damage_Claims
## Fire_Intensity           0.013846751    0.11560952
## Distance_from_Fire      -0.006683554   -0.52530624
## Building_Age            -0.058404627   -0.01473629
## Property_Value          -0.051927902    0.51893206
## Population_Density      -0.018956106   -0.02546527
## Emergency_Response_Time  0.060116306    0.18467430
## Mitigation_Measures      0.052647684    0.12658484
## Construction_Quality    -0.064396286   -0.03088254
## Insurance_Coverage       0.002911288   -0.05472868
## Wind_Speed               0.034991268   -0.07240110
## Humidity                 1.000000000    0.29240552
## Damage_Claims            0.292405519    1.00000000
```

```r
#Evaluation using MSE
actual = test$Damage_Claims
predict1 = predict(m2,data = test)
MSE1 = mean((predict1-actual)^2)
MSE1

## [1] 6.479355

plot(predict1, actual, xlim =c(2,13),ylim = c(2,13))
abline(0,1)
```

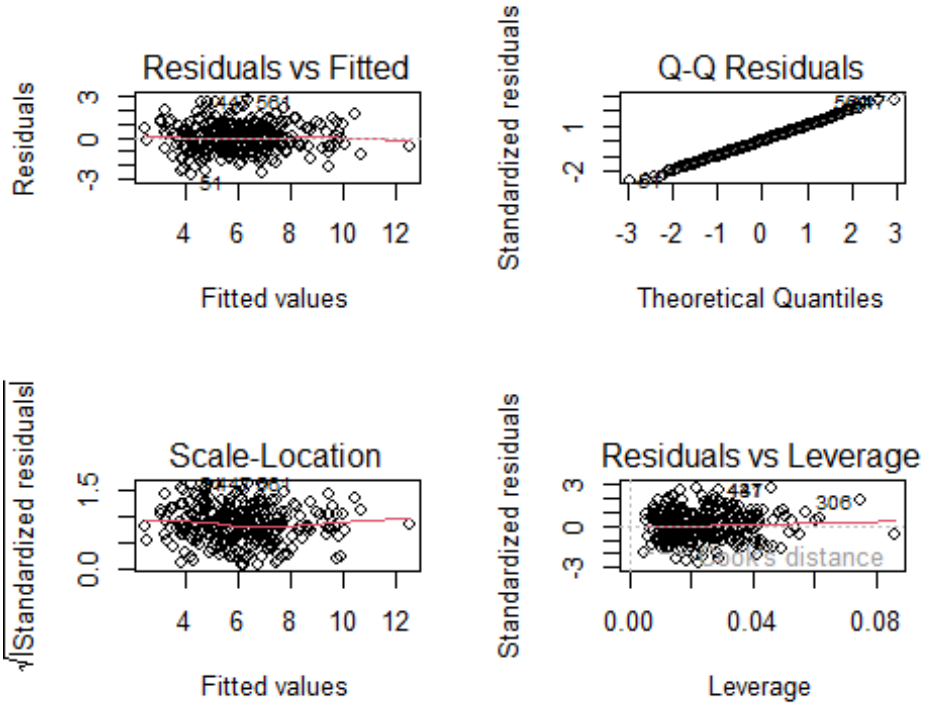
A multiple linear regression model was developed to predict damage claims using all available variables. The initial model identified several predictors as statistically significant, including fire intensity, distance from fire, property value, emergency response time, mitigation measures, and humidity. By refining the model to include only these significant variables, we achieved a strong fit, with an adjusted R-squared of approximately 0.71, indicating that about 71% of the variance in damage claims is explained by the model.

Key findings include:

- Fire intensity, property value, emergency response time, mitigation measures, and humidity all show significant positive relationships with damage claims, meaning increases in these variables are associated with higher losses.
- Distance from fire shows a significant negative relationship, confirming that properties located farther from the fire suffer less damage.
- Other variables, such as building age, population density, construction quality, insurance coverage, and wind speed, did not show significant predictive power and were excluded from the refined model.

The model's performance was evaluated on the test set, resulting in a mean squared error (MSE) of 6.48. This MSE value suggests the model predicts claim amounts with reasonable accuracy, though some variation remains unexplained—likely due to factors not captured in the dataset or inherent randomness in bushfire impact.

Following the multiple linear regression, polynomial and interaction models were also tested to explore the possibility of non-linear relationships or interaction effects between variables. However, these more complex models did not reveal any additional significant relationships, nor did they improve the predictive performance. This result aligns with the earlier correlation analysis, which indicated only moderate linear associations among key variables and suggested little evidence of complex interdependencies or non-linear effects within the dataset. As such, the final model remains focused on the main effects of the most significant predictors identified in the linear analysis.

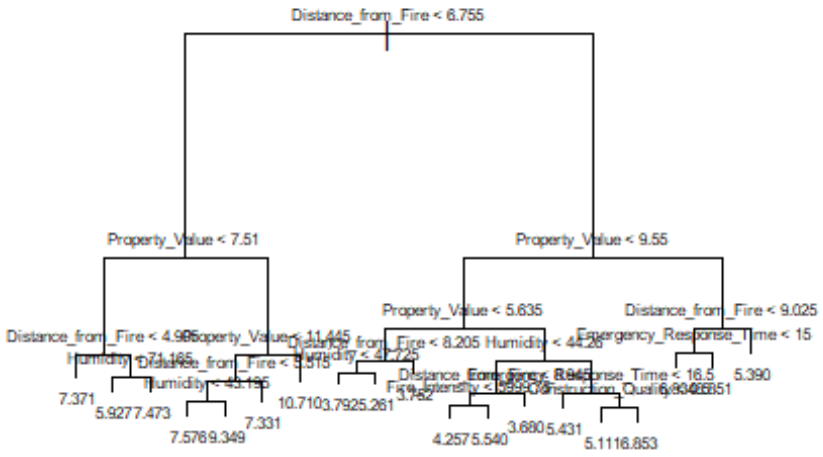


The regression diagnostic plots indicate that the model assumptions are reasonably met.

- Residuals vs Fitted: Residuals are randomly scattered around zero, suggesting linearity and that the model does not suffer from major non-linearity or heteroscedasticity.
- Q-Q Plot: The standardized residuals closely follow the diagonal, indicating that the residuals are approximately normally distributed.
- Scale-Location: The spread of standardized residuals appears consistent across fitted values, supporting the assumption of constant variance (homoscedasticity).
- Residuals vs Leverage: Most observations have low leverage, with only a few points at higher leverage values, but none are extreme outliers or highly influential.

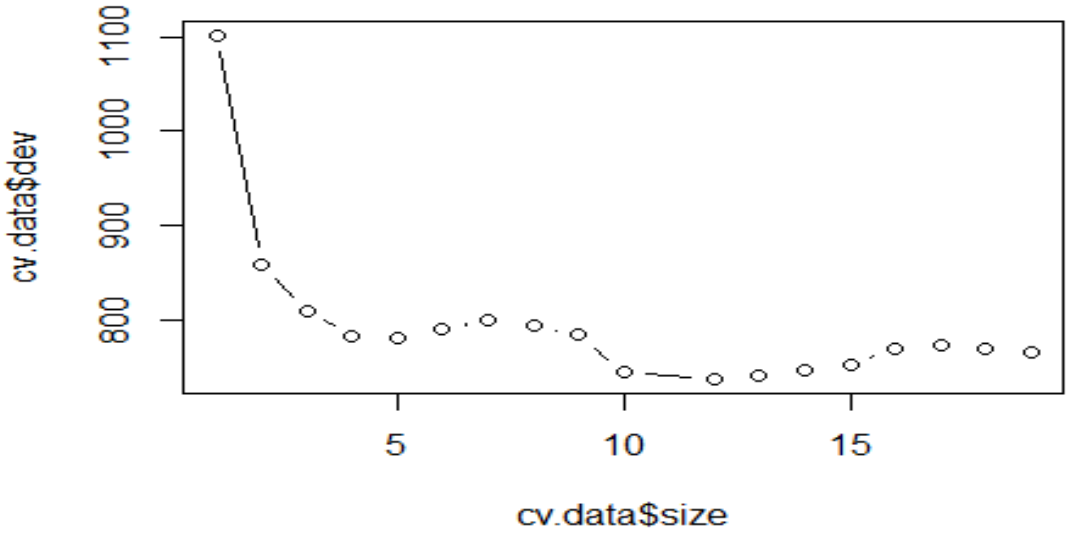
Overall, the diagnostic plots suggest that the linear regression model is appropriate for this data, with no serious violations of model assumptions.

Build regression tree

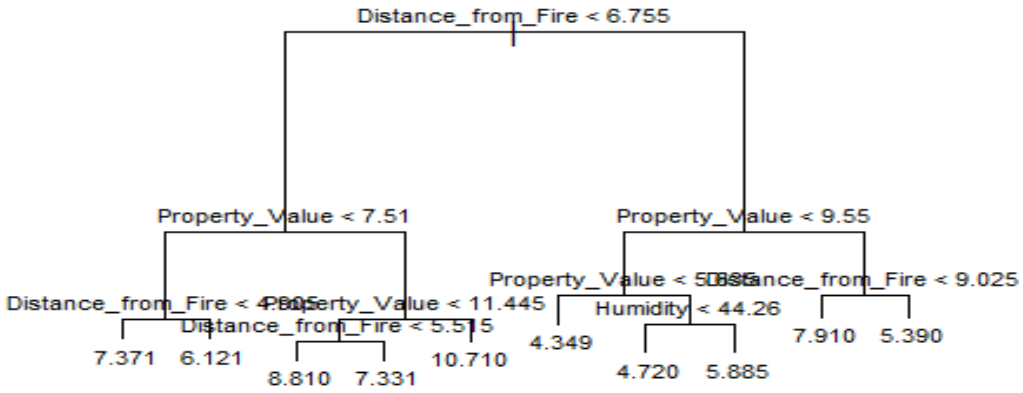


Since the original regression tree is overly complex and lacks interpretability, it is important to identify a simpler tree that maintains strong predictive performance. This balance will provide clearer insights while still maximizing the model's accuracy.

Improve the model by pruning tree



Based on the plot of cross-validated deviance versus tree size, the optimal tree size can be chosen where the deviance first levels off and no longer decreases substantially. In this case after 10, the deviation doesn't decrease significantly anymore, after which further increase in tree size result in minimal improvement. Therefore, selecting a tree size of 10 strikes a good balance between model simplicity and predictive accuracy, avoiding overfitting while retaining strong performance.



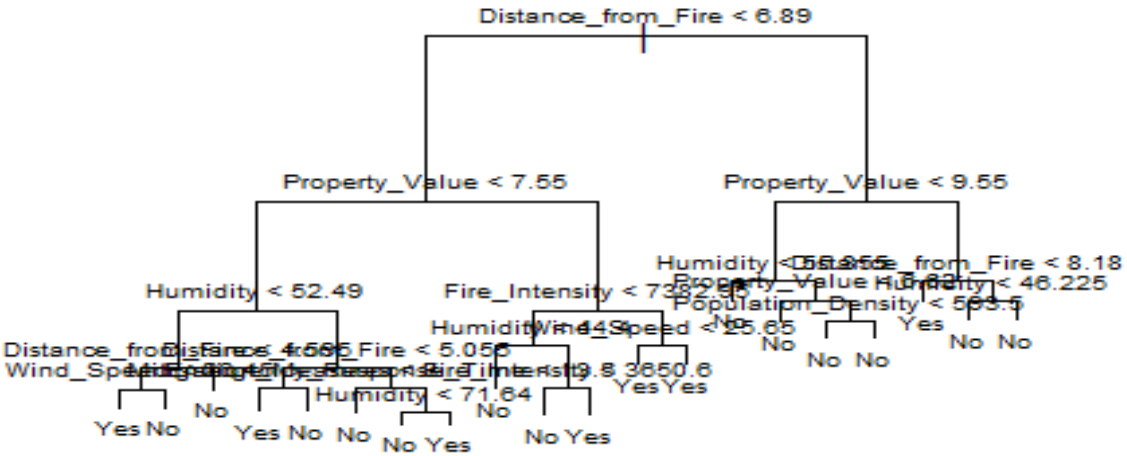
The regression tree highlights Distance from Fire as the most important predictor of damage claims, with an initial split at 6.755 km. Properties located closer to the fire generally experience higher damage claims. Property Value is the next most influential factor, with further splits indicating that higher-value properties tend to have higher claims, especially when located closer to the fire source. For some branches, Humidity and further splits on Distance from Fire and Property Value also play a role, but their impact is secondary.

```
#Calculate MSE and RMSE
predict2 = predict(tree, data = test)
MSE2 = mean((predict2-actual)^2)
RMSE2 = sqrt(MSE2)
MSE2

## [1] 6.413686

RMSE2

## [1] 2.532526
```

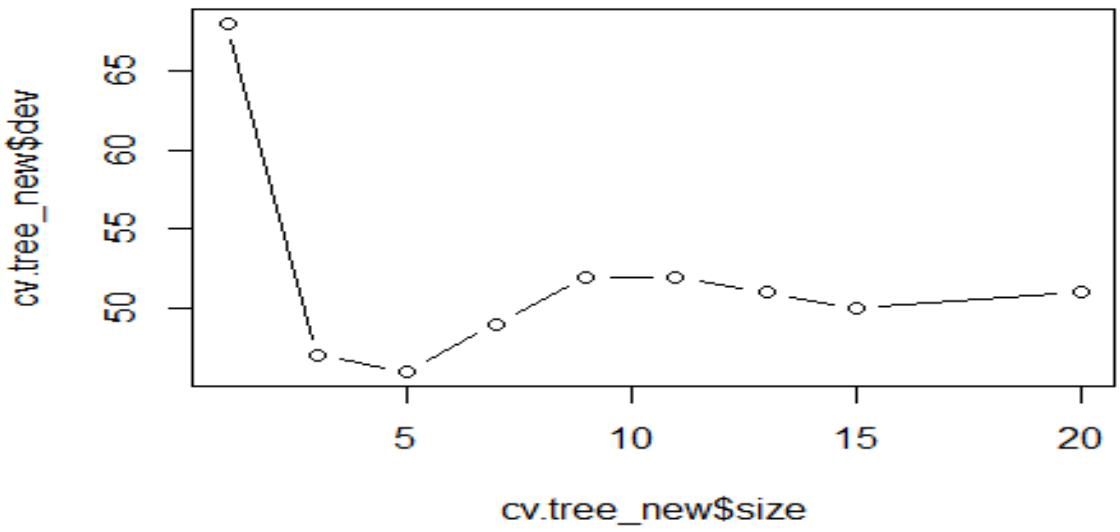
```
#Compare two model using MSE
MSE1
```

```
## [1] 6.479355

MSE2

## [1] 6.413686

#Compare two model using residual plot
par(mfrow = c(1,2))
plot(predict1, actual, xlim =c(2,13),ylim = c(2,13), main = "Multiple linear
regression model")
abline(0,1)
plot(predict2,actual, xlim =c(2,13),ylim = c(2,13), main = "Regression tree
model")
abline(0,1)
```

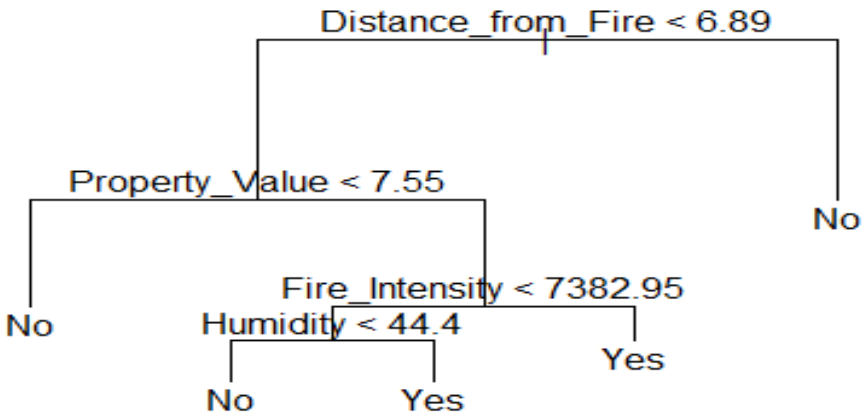
Build decision tree



The same process applied for decision tree, since the initial model is overly complex, it needs to be pruned.



A tree size of 5 in this case will avoid overfitting while retaining strong performance



The decision tree indicates that Distance from Fire is the primary factor: Only properties located within 6.89 km from the fire are considered at risk. And in case it lower, **Property Value** is the next key determinant: Only those with a property value above 7.55 proceed to further assessment. Finally, **Fire Intensity** and **Humidity** determine the final outcome:

- If **Fire Intensity** is below 7382.95, the model further checks humidity:

- If **Humidity** is less than 44.4, the outcome is No (damage claim is low)
- If **Humidity** is at least 44.4, the outcome is Yes (damage claim is high)

Evaluate model using miscalculation rate

```
#Calculate misclassification rate
tree.pred2=predict(prune.tree_new,testing2,type='class')
table(tree.pred2,testing2$Highdamage)

##
## tree.pred2  No Yes
##        No  217  35
##        Yes  11  37

tab3 <- table(tree.pred2,testing2$Highdamage)
mis_rate2 <- (tab3[1,2]+tab3[2,1])/sum(tab3)
mis_rate2

## [1] 0.1533333
```

The misclassification rate of 0.153 (or 15.3%) indicates that the pruned decision tree correctly classifies approximately 84.7% of the test cases. In other words, the model incorrectly predicts the damage class for about 15 out of every 100 properties. This relatively low misclassification rate suggests that the model we have built performing well in distinguishing between high and low damage cases, achieving a good balance between accuracy and simplicity.

Build support vector machines

```
BFD = read.csv("BushFireData.csv")
#Encoding variables
Highdamage = ifelse(Damage_Claims>7.5, "1","0")
new_BFD = data.frame(BFD,Highdamage)
new_BFD$Highdamage = as.factor(new_BFD$Highdamage)
new_BFD$Construction_Quality = ifelse(new_BFD$Construction_Quality ==
"Good",1,0)
IC_factor = factor(new_BFD$Insurance_Coverage, levels = c("None",
"Partially", "Fully"))
new_BFD$Insurance_Coverage = as.integer(IC_factor)-1
#Remove excess variables
new_BFD = new_BFD[,-13]
new_BFD = new_BFD[,-1]

#Divide data set into training and testing set, 70% training, 30% testing
set.seed(1)
tr.id = sample(1:nrow(BFD),nrow(BFD)*0.7)
training = new_BFD[tr.id,]
testing = new_BFD[-tr.id,]
```

```r
#Build support vector machines
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.4.2
```

```r
#linear kernel model
set.seed(1)
linear_svm = tune(svm,Highdamage~., data = training, kernel = "linear",
                  scale = TRUE, ranges = list(cost =
c(0.001,0.01,0.1,1,10,100)))
summary(linear_svm)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.1
##
## - best performance: 0.08809524
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-03 0.21904762 0.05810478
## 2 1e-02 0.17142857 0.05475615
## 3 1e-01 0.08809524 0.05728598
## 4 1e+00 0.09047619 0.04994328
## 5 1e+01 0.09523810 0.04761905
## 6 1e+02 0.09285714 0.04821061
```

```r
linear_bm = linear_svm$best.model
summary(linear_bm)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = Highdamage ~ ., data = training,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 100)), kernel =
"linear",
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  128
##
##  ( 62 66 )
```

```
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
#prediction for linear model
actual = testing$Highdamage
actual
```

```
##   [1] 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 1 1
0 0 0
##  [38] 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
0 0 0
##  [75] 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 1 0 1 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0
0 1 0
## [112] 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0
0 1 1
## [149] 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0
## Levels: 0 1
```

```
pred1 = predict(linear_bm, newdata = testing)
pred1
```

```
##   4   6   9  10  11  12  13  17  23  24  52  54  55  57  59  61  63  66
67  68
##   0   0   0   0   0   0   0   0   1   1   1   0   0   0   0   1   0   0
0   1
##  70  76  80  82  85  87  88  90  94  95  96 100 101 107 118 120 125 128
142 144
##   1   1   0   0   0   1   0   0   0   0   0   1   0   0   0   1   0   0
0   1
## 146 149 151 154 155 158 165 166 170 171 172 175 178 182 184 186 188 191
196 200
##   0   1   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0
0   0
## 206 210 211 213 215 216 225 226 227 228 232 240 243 244 245 250 251 257
258 259
##   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1
0   0
## 261 262 263 267 272 278 281 283 289 301 302 303 308 312 318 319 320 322
332 340
##   1   0   1   1   1   1   0   0   0   0   1   0   0   0   1   0   0   0
0   0
## 341 342 347 348 350 351 353 354 357 360 366 367 370 374 376 384 387 389
392 394
##   1   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0
0   0
## 395 398 401 407 417 420 424 431 432 433 444 445 447 449 450 452 455 456
457 458
```

```
##   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
0   1
## 462 469 473 476 479 481 486 489 493 496 497 503 505 507 510 512 517 520
523 524
##   0   0   0   0   0   0   1   1   0   0   0   0   0   0   1   0   0   0
0   0
## 529 535 539 543 544 547 552 557 560 562 565 568 569 571 573 580 581 583
593 595
##   0   1   0   0   0   0   0   1   0   0   0   0   0   1   1   0   0   1
0   0
## Levels: 0 1
```

```
tab1 = table(pred1,actual)
tab1
```

```
##      actual
## pred1   0   1
##     0 124  18
##     1   8  30
```

```
misrate_linear = (tab1[1,2]+tab1[2,1])/sum(tab1)

#Polynomial kernel model
set.seed(1)
poly_svm = tune(svm,Highdamage~., data = training, kernel = "polynomial",
                scale = TRUE, ranges = list(cost =
c(0.001,0.01,0.1,1,10,100), d =c(2:5)))
summary(poly_svm)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost d
##     1 3
##
## - best performance: 0.1261905
##
## - Detailed performance results:
##     cost d     error dispersion
## 1  1e-03 2 0.2190476 0.05810478
## 2  1e-02 2 0.2190476 0.05810478
## 3  1e-01 2 0.2190476 0.05810478
## 4  1e+00 2 0.2071429 0.04768514
## 5  1e+01 2 0.2428571 0.08078102
## 6  1e+02 2 0.2547619 0.07780207
## 7  1e-03 3 0.2190476 0.05810478
## 8  1e-02 3 0.2190476 0.05810478
## 9  1e-01 3 0.2023810 0.05639949
```

```
## 10 1e+00 3 0.1261905 0.05270463
## 11 1e+01 3 0.1595238 0.06644900
## 12 1e+02 3 0.2214286 0.03731003
## 13 1e-03 4 0.2190476 0.05810478
## 14 1e-02 4 0.2190476 0.05810478
## 15 1e-01 4 0.2047619 0.05634362
## 16 1e+00 4 0.2095238 0.05810478
## 17 1e+01 4 0.2214286 0.07103064
## 18 1e+02 4 0.2666667 0.04994328
## 19 1e-03 5 0.2190476 0.05810478
## 20 1e-02 5 0.2190476 0.05810478
## 21 1e-01 5 0.1928571 0.05549884
## 22 1e+00 5 0.1809524 0.05745067
## 23 1e+01 5 0.1547619 0.07876759
## 24 1e+02 5 0.1952381 0.07342878
```

```
poly_bm = poly_svm$best.model
summary(poly_bm)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = Highdamage ~ ., data = training,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 100), d = c(2:5)),
##     kernel = "polynomial", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  179
##
##  ( 75 104 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
#Prediction for polynomial kernel model
pred2 = predict(poly_bm, newdata = testing)
pred2
```

```
##   4   6   9  10  11  12  13  17  23  24  52  54  55  57  59  61  63  66
67  68
##   0   0   0   0   0   0   0   0   1   1   0   0   0   0   1   0   0   0
0   1
```

```
##  70  76  80  82  85  87  88  90  94  95  96 100 101 107 118 120 125 128
142 144
##   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0
0   0
## 146 149 151 154 155 158 165 166 170 171 172 175 178 182 184 186 188 191
196 200
##   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
0   0
## 206 210 211 213 215 216 225 226 227 228 232 240 243 244 245 250 251 257
258 259
##   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1
0   0
## 261 262 263 267 272 278 281 283 289 301 302 303 308 312 318 319 320 322
332 340
##   1   0   1   1   0   1   0   0   0   0   1   0   0   0   0   0   0   0
0   0
## 341 342 347 348 350 351 353 354 357 360 366 367 370 374 376 384 387 389
392 394
##   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0
0   1
## 395 398 401 407 417 420 424 431 432 433 444 445 447 449 450 452 455 456
457 458
##   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
0   0
## 462 469 473 476 479 481 486 489 493 496 497 503 505 507 510 512 517 520
523 524
##   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1   0   0   0
0   0
## 529 535 539 543 544 547 552 557 560 562 565 568 569 571 573 580 581 583
593 595
##   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
0   0
## Levels: 0 1

tab2 = table(pred2,actual)
tab2

##      actual
## pred2   0   1
##     0 127  30
##     1   5  18

misrate_poly = (tab2[1,2]+tab2[2,1])/sum(tab2)

#Radial kernel model
set.seed(1)
ra_svm = tune(svm,Highdamage~., data = training, kernel = "radial",
              scale = TRUE, ranges = list(cost = c(0.001,0.01,0.1,1,10,100),
gamma = c(0.5, 1, 2, 3,4)))
summary(ra_svm)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##    10   0.5
##
## - best performance: 0.197619
##
## - Detailed performance results:
##     cost gamma     error dispersion
## 1  1e-03   0.5 0.2190476 0.05810478
## 2  1e-02   0.5 0.2190476 0.05810478
## 3  1e-01   0.5 0.2190476 0.05810478
## 4  1e+00   0.5 0.2119048 0.05772412
## 5  1e+01   0.5 0.1976190 0.05388649
## 6  1e+02   0.5 0.1976190 0.05388649
## 7  1e-03   1.0 0.2190476 0.05810478
## 8  1e-02   1.0 0.2190476 0.05810478
## 9  1e-01   1.0 0.2190476 0.05810478
## 10 1e+00   1.0 0.2190476 0.05810478
## 11 1e+01   1.0 0.2190476 0.06023386
## 12 1e+02   1.0 0.2190476 0.06023386
## 13 1e-03   2.0 0.2190476 0.05810478
## 14 1e-02   2.0 0.2190476 0.05810478
## 15 1e-01   2.0 0.2190476 0.05810478
## 16 1e+00   2.0 0.2190476 0.05810478
## 17 1e+01   2.0 0.2190476 0.05810478
## 18 1e+02   2.0 0.2190476 0.05810478
## 19 1e-03   3.0 0.2190476 0.05810478
## 20 1e-02   3.0 0.2190476 0.05810478
## 21 1e-01   3.0 0.2190476 0.05810478
## 22 1e+00   3.0 0.2190476 0.05810478
## 23 1e+01   3.0 0.2190476 0.05810478
## 24 1e+02   3.0 0.2190476 0.05810478
## 25 1e-03   4.0 0.2190476 0.05810478
## 26 1e-02   4.0 0.2190476 0.05810478
## 27 1e-01   4.0 0.2190476 0.05810478
## 28 1e+00   4.0 0.2190476 0.05810478
## 29 1e+01   4.0 0.2190476 0.05810478
## 30 1e+02   4.0 0.2190476 0.05810478

ra_bm = ra_svm$best.model
summary(ra_bm)

##
## Call:
## best.tune(METHOD = svm, train.x = Highdamage ~ ., data = training,
```

```
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 100), gamma = c(0.5,
##         1, 2, 3, 4)), kernel = "radial", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  406
##
##  ( 92 314 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
#Prediction for radial kernel model
pred3 = predict(ra_bm, newdata = testing)
pred3
```

```
##   4   6   9  10  11  12  13  17  23  24  52  54  55  57  59  61  63  66
67  68
##   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
0   0
##  70  76  80  82  85  87  88  90  94  95  96 100 101 107 118 120 125 128
142 144
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   1
## 146 149 151 154 155 158 165 166 170 171 172 175 178 182 184 186 188 191
196 200
##   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0
## 206 210 211 213 215 216 225 226 227 228 232 240 243 244 245 250 251 257
258 259
##   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0
0   0
## 261 262 263 267 272 278 281 283 289 301 302 303 308 312 318 319 320 322
332 340
##   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0
## 341 342 347 348 350 351 353 354 357 360 366 367 370 374 376 384 387 389
392 394
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0
## 395 398 401 407 417 420 424 431 432 433 444 445 447 449 450 452 455 456
457 458
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
0   0
## 462 469 473 476 479 481 486 489 493 496 497 503 505 507 510 512 517 520
523 524
##   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
0   0
## 529 535 539 543 544 547 552 557 560 562 565 568 569 571 573 580 581 583
593 595
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0
## Levels: 0 1

tab3 = table(pred3,actual)
tab3

##      actual
## pred3   0   1
##     0 130  42
##     1   2   6

misrate_ra = (tab3[1,2]+tab3[2,1])/sum(tab3)


#compare misclassification rate between the three kernels
misrate_linear

## [1] 0.1444444

misrate_poly

## [1] 0.1944444

misrate_ra

## [1] 0.2444444
```

Linear SVM performed best (Lowest misclassification rate), which mean the relationship between your features and the target class can be well-approximated by a straight line (hyperplane). Polynomial and radial kernels perform worse, possibly due to overfitting, underfitting, or the nature of the data (it may be nearly linearly separable after transformations).

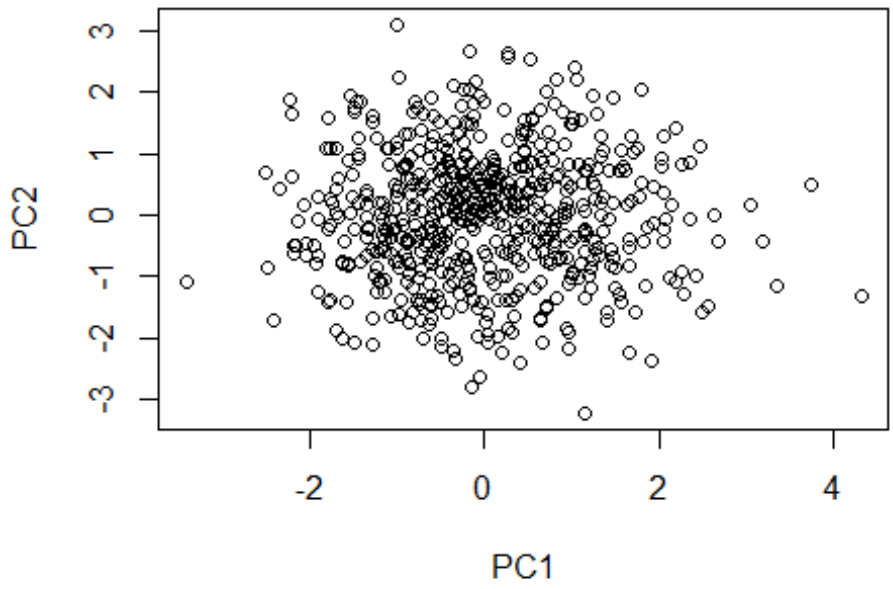
Applied principal component analysis (PCA)

```
#Perform PCA
obj = prcomp(PCA_BFD,center = TRUE, scale = TRUE)
summary(obj)
```

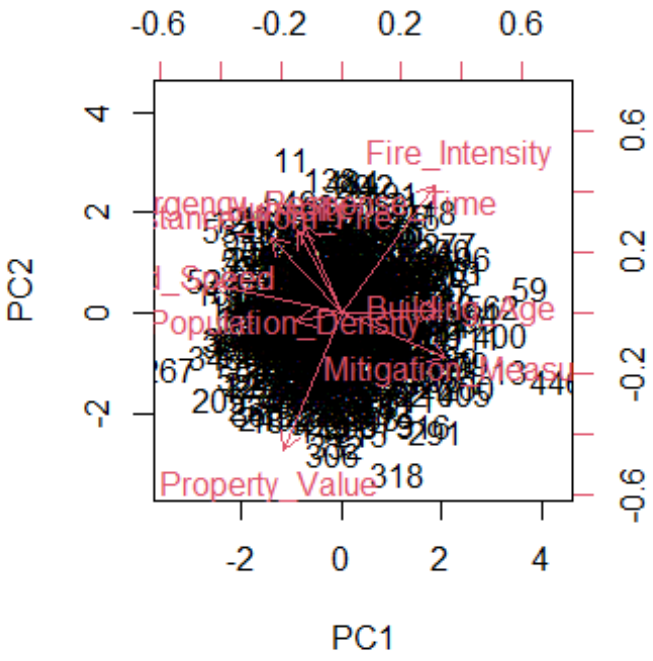
```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5    PC6    PC7
PC8
## Standard deviation     1.1032 1.0420 1.0295 1.0070 0.9978 0.9800 0.9696
0.94216
## Proportion of Variance 0.1352 0.1206 0.1178 0.1127 0.1106 0.1067 0.1045
0.09863
## Cumulative Proportion  0.1352 0.2559 0.3736 0.4863 0.5969 0.7036 0.8081
0.90672
##                            PC9
## Standard deviation     0.91624
## Proportion of Variance 0.09328
## Cumulative Proportion  1.00000

#Loadings
obj$rotation

##                                  PC1         PC2          PC3         PC4
## Fire_Intensity           0.3879646  0.53114764  0.277551664 -0.1241035
## Distance_from_Fire      -0.2886636  0.31299597  0.283544133  0.1461189
## Building_Age             0.3992121  0.01092616  0.188956016 -0.2397889
## Property_Value          -0.2343575 -0.56694049  0.046357169  0.1135330
## Population_Density      -0.1844550 -0.03471533  0.665073999 -0.2027555
## Emergency_Response_Time -0.1559268  0.36447621 -0.481160270 -0.1691881
## Mitigation_Measures      0.4326790 -0.18691331  0.233395512  0.5172214
## Wind_Speed              -0.5238593  0.11616106  0.276349250 -0.1083837
## Humidity                -0.1785676  0.34043615 -0.005782125  0.7374684
##                                 PC5         PC6         PC7          PC8
## Fire_Intensity           0.06774495  0.05978999  0.17105479 -0.46566106
## Distance_from_Fire       0.39809326 -0.39041991 -0.63245462 -0.04322718
## Building_Age             0.63950630  0.32255633  0.06747088  0.33870550
## Property_Value           0.55375337 -0.06649410  0.24063293 -0.27686720
## Population_Density      -0.19223228 -0.34593910  0.37214303  0.42027466
## Emergency_Response_Time  0.26021220 -0.48561646  0.48327720 -0.01756541
## Mitigation_Measures     -0.06798850 -0.34429795  0.17163800 -0.30500609
## Wind_Speed              -0.03133774  0.41519685  0.19695159 -0.45038549
## Humidity                 0.10513399  0.29850745  0.26074521  0.34219758
##                                 PC9
## Fire_Intensity          -0.46974977
## Distance_from_Fire       0.06469508
## Building_Age             0.33914973
## Property_Value          -0.40372316
## Population_Density      -0.09792678
## Emergency_Response_Time  0.21284964
## Mitigation_Measures      0.45848552
## Wind_Speed               0.45711549
## Humidity                -0.15190984

#Principal components
head(obj$x)
```

```
##              PC1        PC2        PC3           PC4         PC5         PC6
## [1,] -0.5245971 -1.4173303 -1.3564853 -0.004936077 -0.28599453 -0.2470297
## [2,]  1.8103159  0.5301982 -1.8342258 -0.875235087  0.18356445  0.2213466
## [3,] -0.0967987  0.3485307 -0.5527159  0.069854381 -2.40385667 -0.7887507
## [4,]  1.5715451  1.0495303 -0.1293591 -0.669977650 -0.20716960  1.1520080
## [5,] -1.4997559  0.6674983  1.4491181  0.446020777 -0.25751187  1.6647432
## [6,] -1.1547775 -0.5263678 -0.6187590 -0.068258525  0.08416854  2.0334067
##              PC7        PC8         PC9
## [1,]  0.07927102 -1.0187599 -0.29485698
## [2,] -0.32898494 -1.0543512  0.07111559
## [3,]  0.50830477 -0.1184211  0.49628723
## [4,]  0.34139370  0.4354578 -0.46122443
## [5,]  1.27443050 -1.4171480 -0.98520328
## [6,] -1.62895108  0.1303434  1.02821475
```

```r
#First two principal components
plot(obj$x[,1:2])
```




```r
#biplot
biplot(obj,scale = 0)
```



The principal component analysis (PCA) of our bushfire dataset revealed that the first few principal components capture the majority of the variation in the data. Specifically, the first three components together account for over 37% of the total variance, while the first seven components explain more than 80%. The first principal component (PC1) is most strongly associated with higher fire intensity, older building age, and the presence of mitigation measures, while being negatively related to wind speed and property value. This suggests that properties with high fire intensity, older structures, and more mitigation efforts tend to share similar risk profiles, especially when wind speed and property value are lower. Overall, the PCA indicates that much of the variability in bushfire impact and risk among properties can be summarized using a smaller set of combined factors, which can help simplify further analysis and guide targeted risk management strategies.

Conclusion

Overall, this report applied multiple linear regression, logistic regression, and decision tree models to identify the key factors driving property damage claims in the context of bushfires. All three analytical approaches produced consistent results, with low misclassification rates, demonstrating the effectiveness and reliability of these models. The findings highlight that fire intensity, property value, emergency response time, mitigation measures, humidity, and distance from the fire are the most critical variables influencing the extent of property damage. These insights provide valuable guidance for insurance companies and relevant stakeholders, enabling them to better understand risk factors and

make more informed decisions regarding property insurance, risk assessment, and mitigation strategies.