In [11]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import sys

# Ensure images are saved to the correct directory
FIGURES_DIR = '../reports/figures'
os.makedirs(FIGURES_DIR, exist_ok=True)

from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

1. Helper Functions

Functions to generate standard EDA plots: Feature Importance, Histograms, and Boxplots.

In [12]:
```
def save_and_show_feature_importance(df, target, title, filename, top_n=15):
    print(f"Processing Feature Importance: {title}...")
    X = df.drop(target, axis=1)
    y = df[target]

    numeric_features = X.select_dtypes(include=['int64', 'float64']).columns
    categorical_features = X.select_dtypes(include=['object']).columns

    numeric_transformer = SimpleImputer(strategy='median')
    categorical_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('encoder', OneHotEncoder(handle_unknown='ignore'))
    ])

    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numeric_features),
            ('cat', categorical_transformer, categorical_features)
        ])

    clf = RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=-1)
    pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                               ('classifier', clf)])

    pipeline.fit(X, y)

    try:
```

```
        onehot_cols = pipeline.named_steps['preprocessor'].named_transformers_
        feature_names = np.r_[numeric_features, onehot_cols]
    except:
        feature_names = numeric_features

    importances = pipeline.named_steps['classifier'].feature_importances_
    indices = np.argsort(importances)[::-1]

    plt.figure(figsize=(12, 8))
    sns.barplot(x=importances[indices[:top_n]], y=feature_names[indices[:top_r
    plt.title(f'Feature Importance: {title}')
    plt.xlabel('Relative Importance')
    plt.tight_layout()
    plt.savefig(os.path.join(FIGURES_DIR, filename))
    plt.show()

def plot_distributions(df, cols, title_prefix, filename_suffix):
    print(f"Plotting Distributions: {title_prefix}...")
    num_cols = len(cols)
    rows = (num_cols // 3) + 1
    plt.figure(figsize=(15, rows * 4))
    for i, col in enumerate(cols):
        plt.subplot(rows, 3, i + 1)
        sns.histplot(df[col], kde=True, bins=30, color='skyblue')
        plt.title(f'Distribution of {col}')
    plt.suptitle(f'{title_prefix} Feature Distributions', y=1.02)
    plt.tight_layout()
    plt.savefig(os.path.join(FIGURES_DIR, f'{filename_suffix}_distributions.pr
    plt.show()

def plot_boxplots(df, cols, title_prefix, filename_suffix):
    print(f"Plotting Outliers (Boxplots): {title_prefix}...")
    num_cols = len(cols)
    rows = (num_cols // 3) + 1
    plt.figure(figsize=(15, rows * 4))
    for i, col in enumerate(cols):
        plt.subplot(rows, 3, i + 1)
        sns.boxplot(x=df[col], color='salmon')
        plt.title(f'Boxplot of {col}')
    plt.suptitle(f'{title_prefix} Feature Outliers', y=1.02)
    plt.tight_layout()
    plt.savefig(os.path.join(FIGURES_DIR, f'{filename_suffix}_boxplots.png'))
    plt.show()
```

2. Smoker Status EDA

In [13]:
```
df_smoker = pd.read_csv('../data/raw/train_dataset.csv')
print("Shape:", df_smoker.shape)
display(df_smoker.describe())
```

```
Shape: (38984, 23)
```

	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	ey
count	38984.000000	38984.000000	38984.000000	38984.000000	38984.000000	3
mean	44.127591	164.689488	65.938718	82.062115	1.014955	
std	12.063564	9.187507	12.896581	9.326798	0.498527	
min	20.000000	130.000000	30.000000	51.000000	0.100000	
25%	40.000000	160.000000	55.000000	76.000000	0.800000	
50%	40.000000	165.000000	65.000000	82.000000	1.000000	
75%	55.000000	170.000000	75.000000	88.000000	1.200000	
max	85.000000	190.000000	135.000000	129.000000	9.900000	

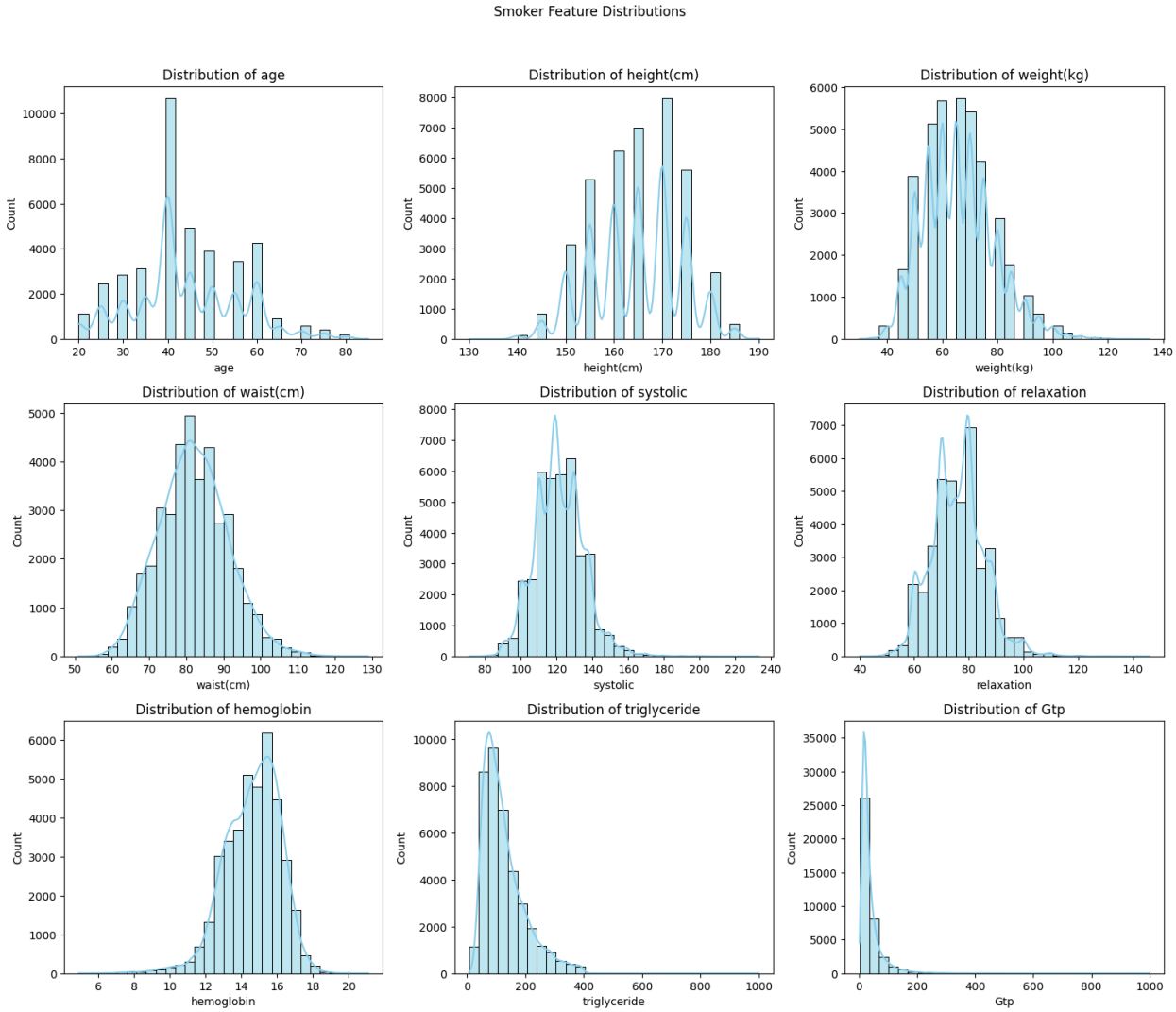
8 rows × 23 columns

2.1 Feature Distributions (Normality Check)

Crucial for models like SVM/KNN to see if data is skewed.

In [14]:
```
numeric_cols_smoker = df_smoker.select_dtypes(include=['float64', 'int64']).cc
# Select top 9 interesting physiological features to avoid clutter
selected_cols = ['age', 'height(cm)', 'weight(kg)', 'waist(cm)', 'systolic', '
plot_distributions(df_smoker, selected_cols, 'Smoker', 'smoker')
```

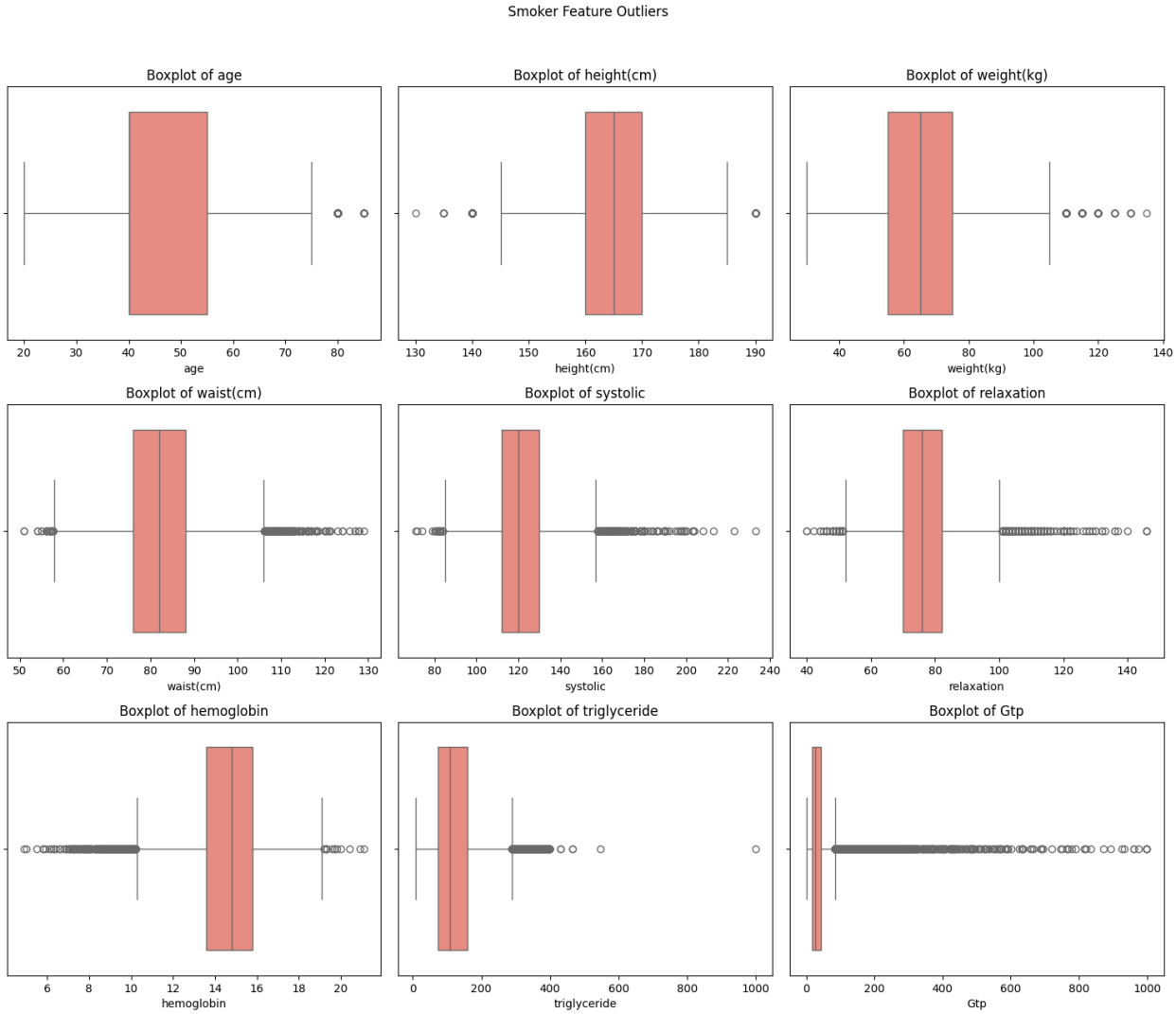
```
Plotting Distributions: Smoker...
```



2.2 Outlier Detection

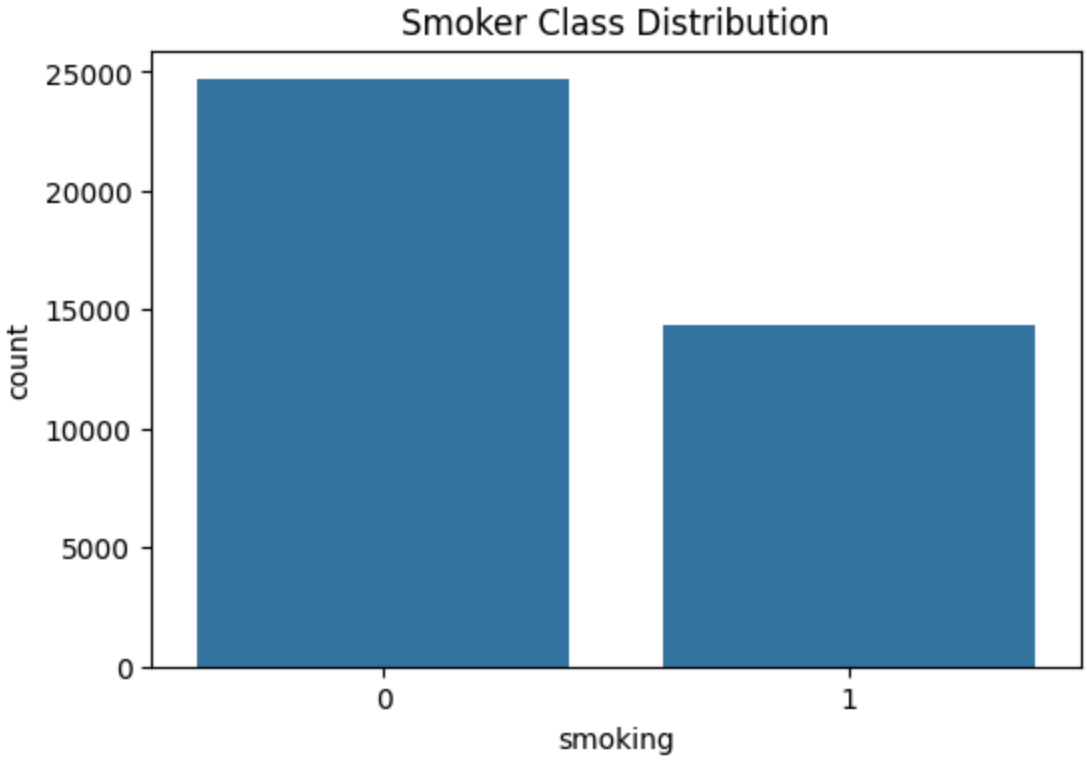
In [15]:
```
plot_boxplots(df_smoker, selected_cols, 'Smoker', 'smoker')
```

```
Plotting Outliers (Boxplots): Smoker...
```



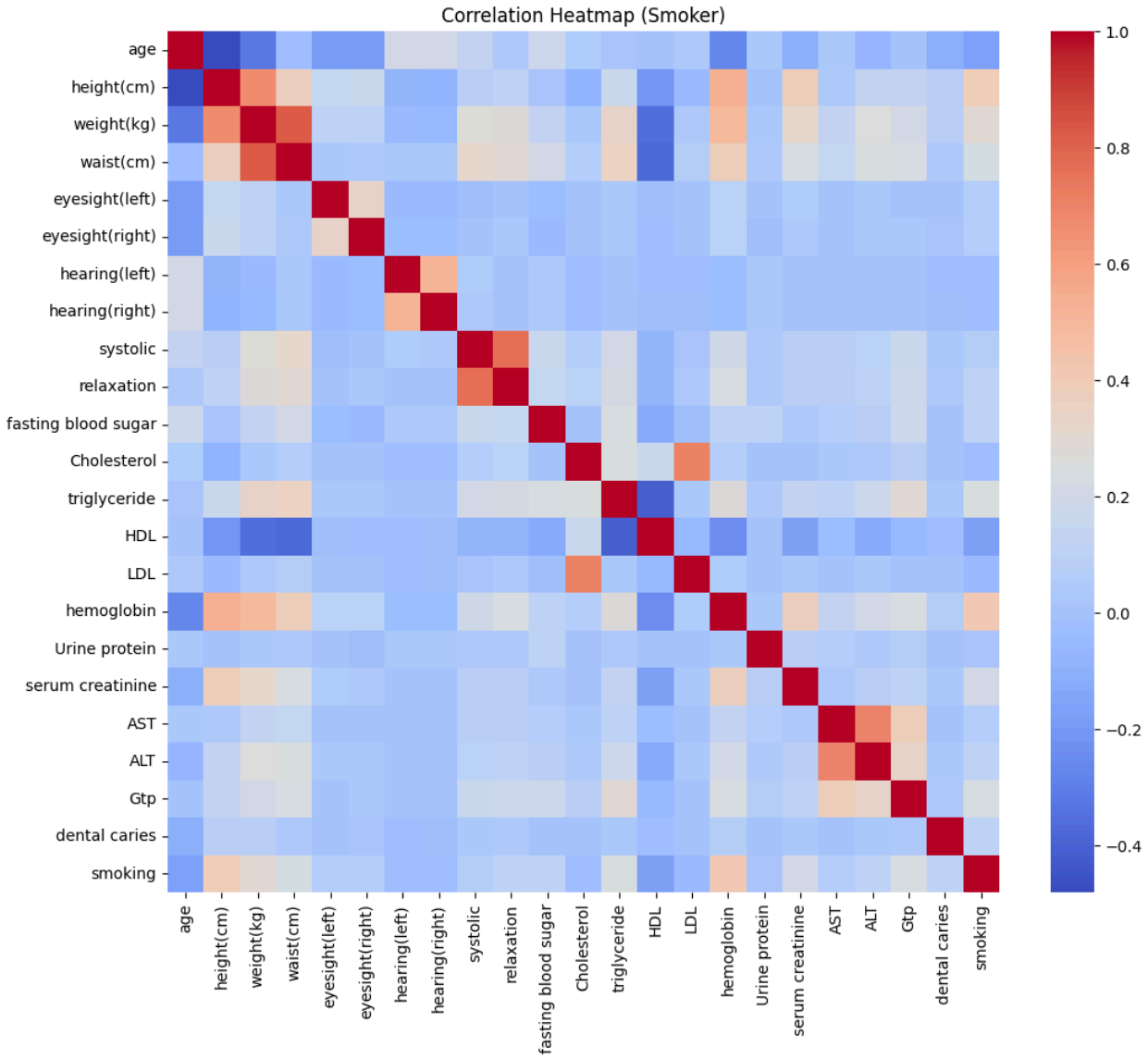
2.3 Class Balance

In [16]:
```
plt.figure(figsize=(6, 4))
sns.countplot(x='smoking', data=df_smoker)
plt.title('Smoker Class Distribution')
plt.savefig(os.path.join(FIGURES_DIR, 'smoker_class_dist.png'))
plt.show()
```



2.4 Correlation Heatmap

In [17]:
```
plt.figure(figsize=(12, 10))
sns.heatmap(df_smoker.corr(numeric_only=True), cmap='coolwarm', annot=False)
plt.title('Correlation Heatmap (Smoker)')
plt.savefig(os.path.join(FIGURES_DIR, 'smoker_correlation.png'))
plt.show()
```



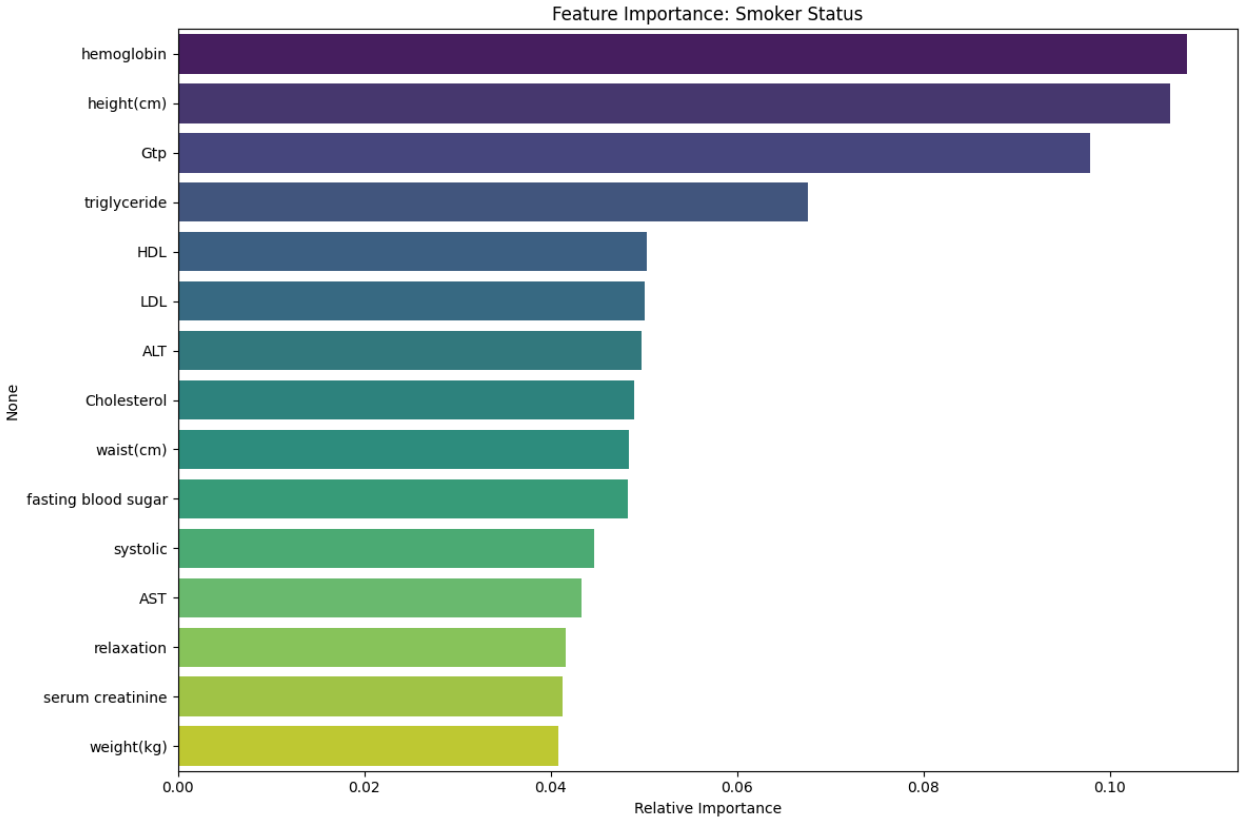
2.5 Feature Importance

In [18]:
```
save_and_show_feature_importance(df_smoker, 'smoking', 'Smoker Status', 'featu
```

```
Processing Feature Importance: Smoker Status...
C:\Users\NIKHIL AGRAWAL\AppData\Local\Temp\ipykernel_6744\1093813786.py:37: Fut
ureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.

  sns.barplot(x=importances[indices[:top_n]], y=feature_names[indices[:top_n]],
palette='viridis')
```



3. Forest Cover EDA

In [19]:
```
df_forest = pd.read_csv('../data/raw/covtype.csv')
if 'Id' in df_forest.columns:
    df_forest = df_forest.drop('Id', axis=1)
print("Shape:", df_forest.shape)
display(df_forest.describe())
```

```
Shape: (581012, 55)
```

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrc
count	581012.000000	581012.000000	581012.000000	581012.00
mean	2959.365301	155.656807	14.103704	269.42
std	279.984734	111.913721	7.488242	212.54
min	1859.000000	0.000000	0.000000	0.00
25%	2809.000000	58.000000	9.000000	108.00
50%	2996.000000	127.000000	13.000000	218.00
75%	3163.000000	260.000000	18.000000	384.00
max	3858.000000	360.000000	66.000000	1397.00

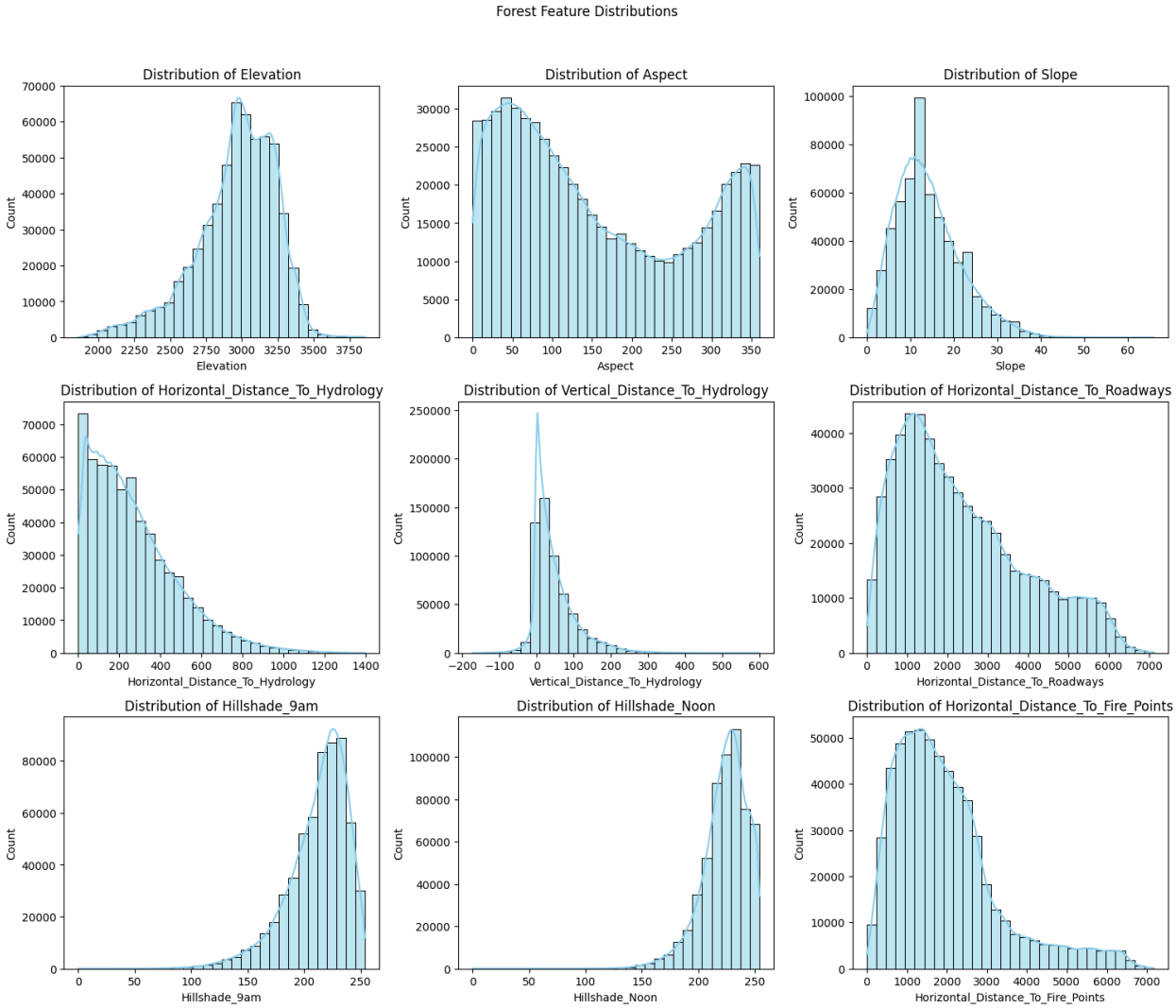
8 rows × 55 columns

3.1 Feature Distributions (Continuous)

Forest cover has many binary columns (Soil types). We focus plots on continuous geographic features.

In [20]:
```
geo_cols = ['Elevation', 'Aspect', 'Slope', 'Horizontal_Distance_To_Hydrology'
            'Vertical_Distance_To_Hydrology', 'Horizontal_Distance_To_Roadways
            'Hillshade_9am', 'Hillshade_Noon', 'Horizontal_Distance_To_Fire_Po
plot_distributions(df_forest, geo_cols, 'Forest', 'forest')
```

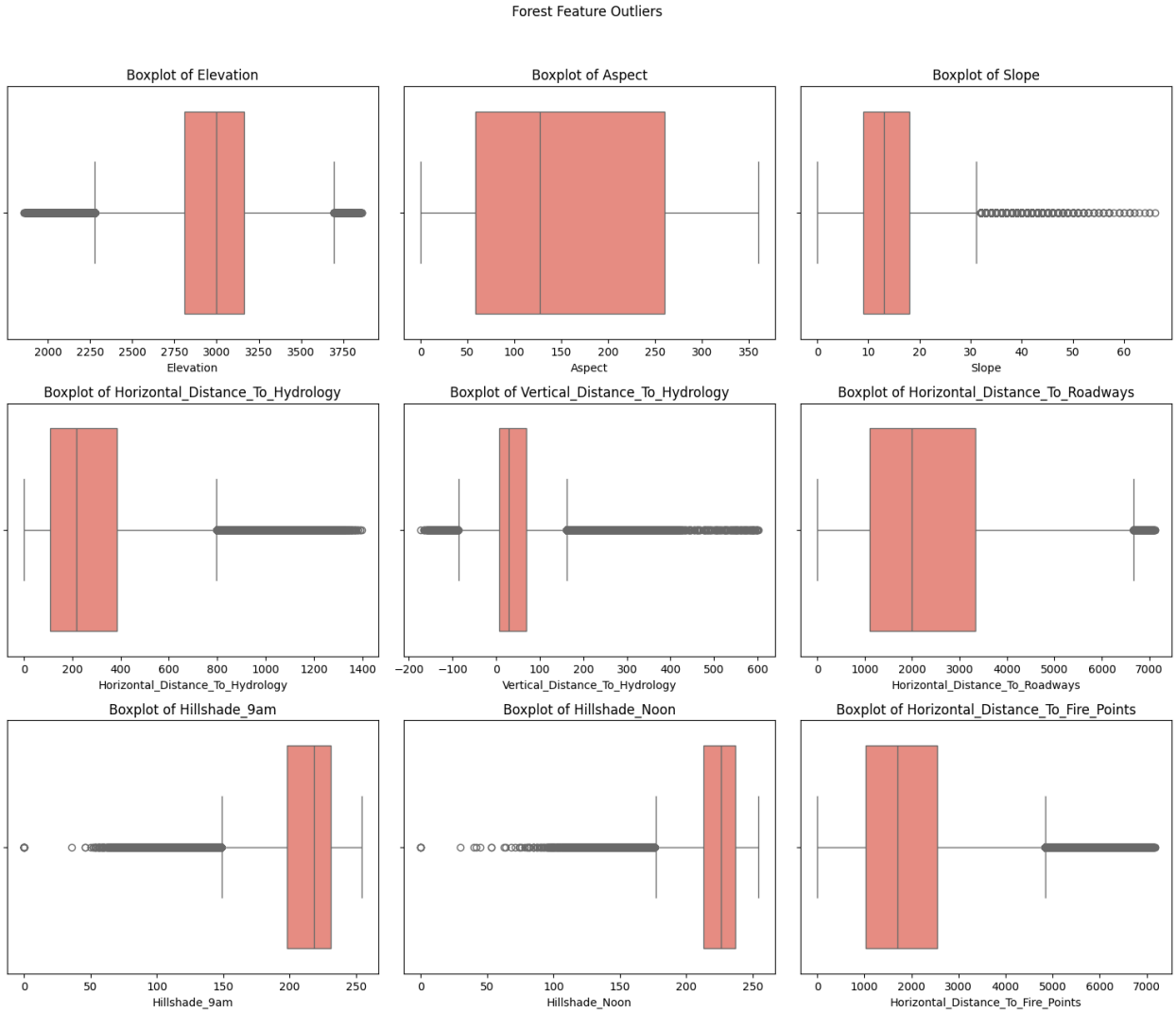
```
Plotting Distributions: Forest...
```



3.2 Outlier Detection

In [21]:
```
plot_boxplots(df_forest, geo_cols, 'Forest', 'forest')
```

```
Plotting Outliers (Boxplots): Forest...
```



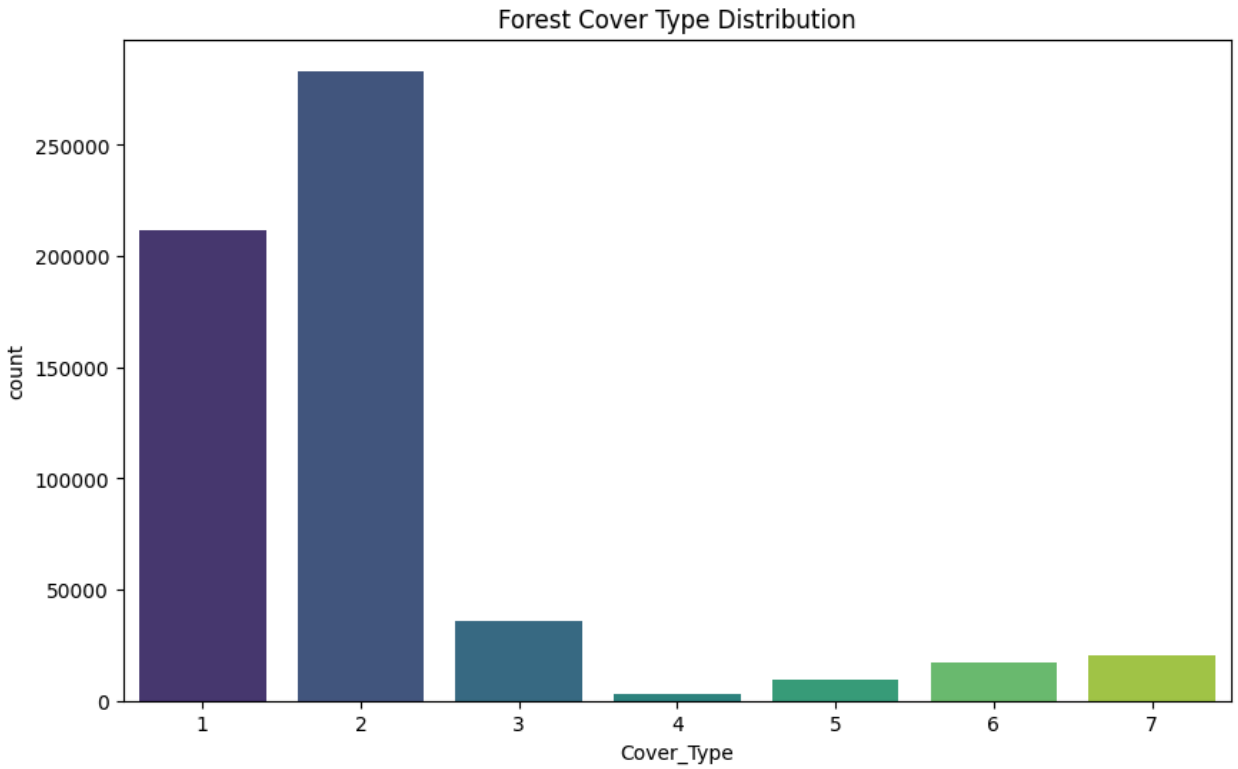
3.3 Class Balance

In [22]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Cover_Type', data=df_forest, palette='viridis')
plt.title('Forest Cover Type Distribution')
plt.savefig(os.path.join(FIGURES_DIR, 'forest_class_dist.png'))
plt.show()
```

```
C:\Users\NIKHIL AGRAWAL\AppData\Local\Temp\ipykernel_6744\227076788.py:2: Futur
eWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same e
ffect.

  sns.countplot(x='Cover_Type', data=df_forest, palette='viridis')
```



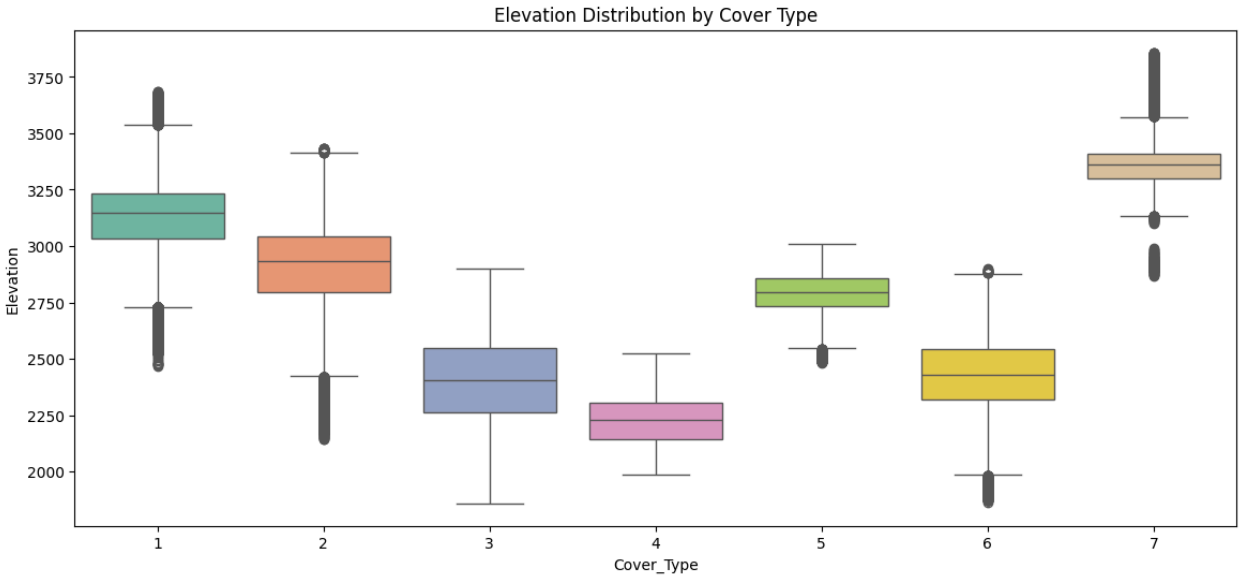
3.4 Key Relationship: Elevation vs Cover Type

In [23]:
```
plt.figure(figsize=(14, 6))
sns.boxplot(x='Cover_Type', y='Elevation', data=df_forest, palette='Set2')
plt.title('Elevation Distribution by Cover Type')
plt.savefig(os.path.join(FIGURES_DIR, 'forest_elevation_dist.png'))
plt.show()
```

```
C:\Users\NIKHIL AGRAWAL\AppData\Local\Temp\ipykernel_6744\620108303.py:2: Futur
eWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same e
ffect.

  sns.boxplot(x='Cover_Type', y='Elevation', data=df_forest, palette='Set2')
```



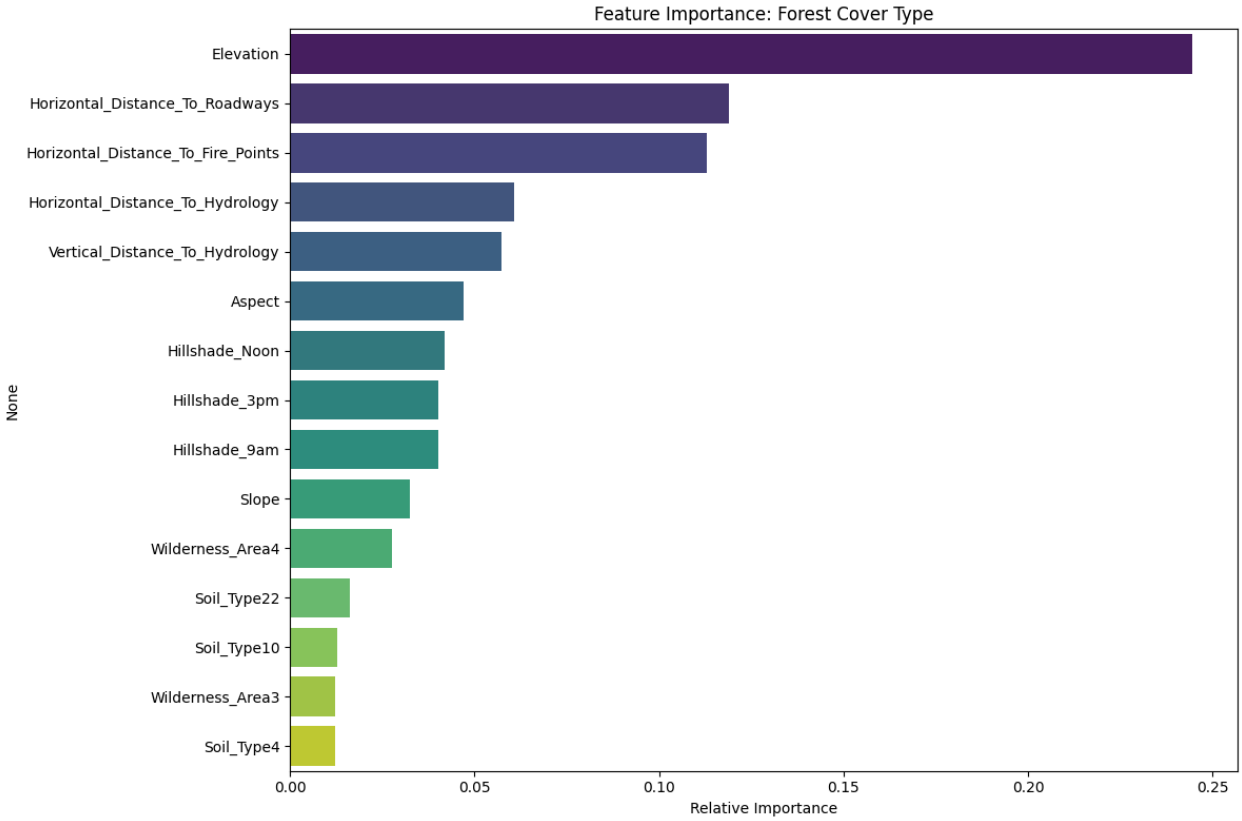
3.5 Feature Importance

In [24]:
```
save_and_show_feature_importance(df_forest, 'Cover_Type', 'Forest Cover Type',
```

```
Processing Feature Importance: Forest Cover Type...
C:\Users\NIKHIL AGRAWAL\AppData\Local\Temp\ipykernel_6744\1093813786.py:37: Fut
ureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same e
ffect.

  sns.barplot(x=importances[indices[:top_n]], y=feature_names[indices[:top_n]],
palette='viridis')
```


Feature Importance: Forest Cover Type
Elevation
Horizontal_Distance_To_Roadways
Horizontal_Distance_To_Fire_Points
Horizontal_Distance_To_Hydrology
Vertical_Distance_To_Hydrology
Aspect
Hillshade_Noon
Hillshade_3pm
Hillshade_9am
Slope
Wilderness_Area4
Soil_Type22
Soil_Type10
Wilderness_Area3
Soil_Type4
None
0.00
0.05
0.10
0.15
0.20
0.25
Relative Importance