In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
```

In [3]:
```
df = pd.read_csv('Auto.csv')
df_copy = df.copy()
```

In [4]:
```
# Eliminates the rows (instances) with '?' as a predictor value
df_copy['horsepower'] = pd.to_numeric(df_copy['horsepower'], errors='coerce')
df_copy = df_copy.dropna()
# df_copy['name'] = df_copy['name'].str.split(' ').str[0]

df_copy = df_copy.drop('name', 1)
```

Simple Linear Regression

Problem a

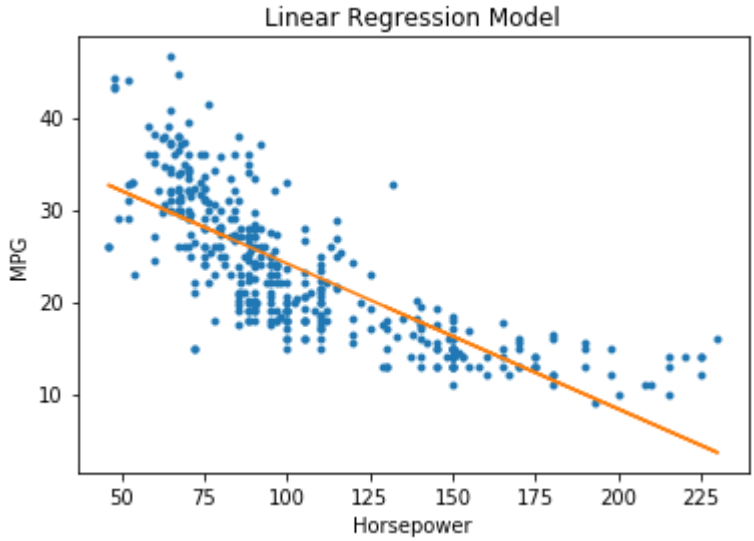
In [5]:
```
# SIMPLE LINEAR REGRESSION MODEL

# Extracts the predictor and response columns
X = np.asarray(df_copy[['horsepower']]) # Extracts the horsepower variable as
 an array to use as predictor
y_true = np.asarray(df_copy[['mpg']]) # Extracts the mpg variable as an array
 to use as response

# Fit the least squares linear regression line to the data
reg = LinearRegression(fit_intercept=True).fit(X, y_true)

# Extract slope and intercept coefficients and prints it out
slope = reg.coef_[0,0]
intercept = reg.intercept_[0]
print("Regression Slope: " + repr(slope))
print("Intercept: " + repr(intercept))

# Calculates the Coefficient of Determination (R^2) and prints it out
y_pred = reg.predict(X)
r_squared = 1 - (((y_true - y_pred)**2).sum())/(((y_true - y_true.mean())**2).
sum())
print("Coefficient of Determination: " + repr(r_squared))

# Predicts response given a certain input value for horsepower
input_value = 95
prediction = reg.predict(np.array([[input_value]]))
print("The prediction for Horsepower = " + repr(input_value) + " is " + repr(p
rediction[0,0]))
```

```
Regression Slope: -0.15784473335365357
Intercept: 39.93586102117046
Coefficient of Determination: 0.6059482578894348
The prediction for Horsepower = 95 is 24.94061135257337
```

Problem b

In [6]:
```
# Create scatterplot with linear regression model
plt.plot(X, y_true, '.') # Plots the data points
plt.plot(X, slope*X + intercept) # Plots the linear regression line
plt.title('Linear Regression Model')
plt.xlabel('Horsepower')
plt.ylabel('MPG')
```

Out[6]: Text(0,0.5,'MPG')



In [ ]: