In [72]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from pandas.plotting import scatter_matrix
from sklearn.neighbors import KNeighborsClassifier, KNeighborsRegressor
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import KFold
```

In [73]:
```
df = pd.read_csv('Wage.csv')
df_copy = df.copy()
```

## Data Preprocessing

In [74]:
```
# QUANTIZE QUALITATIVE FEATURES
# Sex
df_copy = df_copy.drop('sex', axis=1)

# Marital Status
df_copy = pd.get_dummies(df_copy, columns=['maritl'])

# Race
df_copy = pd.get_dummies(df_copy, columns=['race'])

# Education
education_mapping = {'1. < HS Grad': 0.0,
                     '2. HS Grad': 1.0,
                     '3. Some College': 2.0,
                     '4. College Grad': 3.0,
                     '5. Advanced Degree': 4.0}
df_copy['education'] = df_copy['education'].replace(education_mapping)

# Region
df_copy = df_copy.drop('region', axis=1)

# Job Classes
df_copy['jobclass'] = df_copy['jobclass'].replace({'1. Industrial': 1.0, '2. Information': 0.0})

# Health
df_copy['health'] = df_copy['health'].replace({'2. >=Very Good': 1.0, '1. <=Good': 0.0})

# Health Insurance
df_copy['health_ins'] = df_copy['health_ins'].replace({'1. Yes': 1.0, '2. No': 0.0})

df_copy = df_copy.drop('wage', axis=1)
```

In [75]:
```
X = np.asarray(df_copy.drop('logwage',1))
y = np.asarray(df_copy['logwage'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

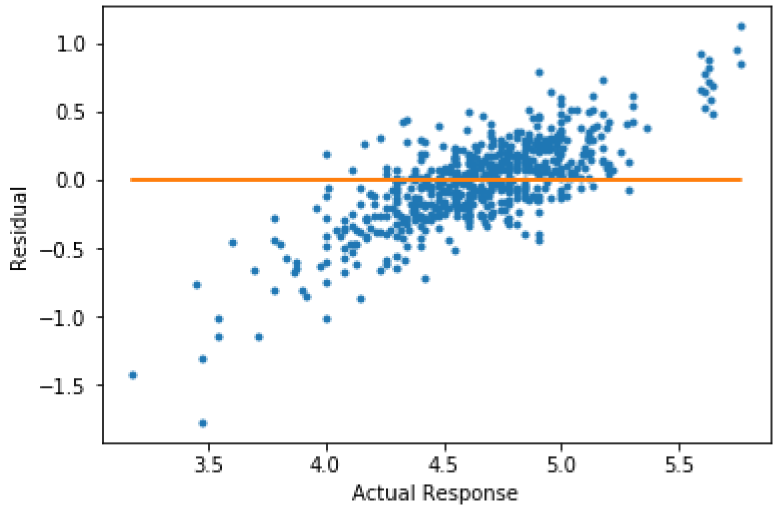
## KNN Model

In [76]:
```
# Create KNN model and fit it
KNN_regressor = KNeighborsRegressor(n_neighbors=5)
KNN_regressor.fit(X_train, y_train)

# make predictions and find residuals
y_pred = KNN_regressor.predict(X_test)
residuals = y_test - y_pred

# Plot residuals
plt.plot(y_test, residuals, '.')
plt.plot(y_test, 0*y_test)
plt.xlabel('Actual Response')
plt.ylabel('Residual')
```

Out[76]:
```
Text(0,0.5,'Residual')
```



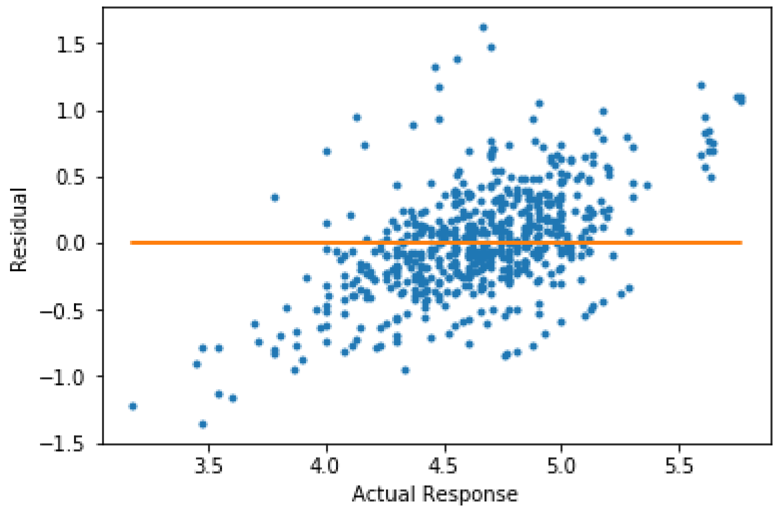
## Decision Tree

In [77]:
```python
# Create Decision Tree Regressor and fit to training data
DT_regressor = DecisionTreeRegressor()
DT_regressor.fit(X_train, y_train)

y_pred = DT_regressor.predict(X_test)
residuals = y_test - y_pred

# Plot residuals
plt.plot(y_test, residuals, '.')
plt.plot(y_test, 0*y_test)
plt.xlabel('Actual Response')
plt.ylabel('Residual')
```

Out[77]:
```
Text(0,0.5,'Residual')
```



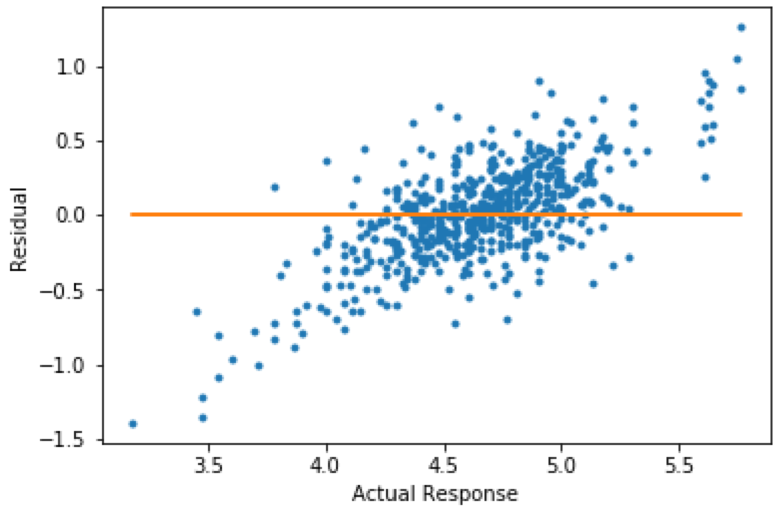
## Random Forest

In [78]:
```
# Create Random Forest Regressor and fit to training data
RF_regressor = RandomForestRegressor()
RF_regressor.fit(X_train, y_train)

y_pred = RF_regressor.predict(X_test)
residuals = y_test - y_pred

# Plot residuals
plt.plot(y_test, residuals, '.')
plt.plot(y_test, 0*y_test)
plt.xlabel('Actual Response')
plt.ylabel('Residual')
```

Out[78]:
```
Text(0,0.5,'Residual')
```



In [ ]:

In [ ]: