

Assignment 4 Answer (Q6)

Alternative Answer

```
def run_ttest():
```

```
    '''First creates new data showing the decline or growth of housing prices
    between the recession start and the recession bottom. Then runs a ttest
    comparing the university town values to the non-university towns values,
    return whether the alternative hypothesis (that the two groups are the
    same)
```

```
    is true or not as well as the p-value of the confidence.
```

```
    Return the tuple (different, p, better)
```

```
    where
```

```
    different=True if the t-test is True at a  $p < 0.01$  (we reject the null hypothesis),
    different=False if otherwise (we cannot reject the null hypothesis).
```

```
    The variable p should
```

```
    be equal to the exact p value returned from scipy.stats.ttest_ind(). The
    value for better should be either "university town" or "non-university town"
    depending on which has a lower mean price ratio (which is equivalent to a
    reduced market loss).'''
```

```
    data = convert_housing_data_to_quarters().copy()
```

```
    data = data.loc[:, '2008q3': '2009q2']
```

```
    data = data.reset_index()
```

```
    def price_ratio(row):
```

```
        return (row['2008q3'] - row['2009q2'])/row['2008q3']
```

```
    data['up&down'] = data.apply(price_ratio, axis=1)
```

```
    #uni data
```

```
    uni_town = get_list_of_university_towns()['RegionName']
```

```
    uni_town = set(uni_town)
```

```
    def is_uni_town(row):
```

```
        #check if the town is a university towns or not.
```

```
        if row['RegionName'] in uni_town:
```

```
            return 1
```

```
        else:
```

```
            return 0
```

```
    data['is_uni'] = data.apply(is_uni_town, axis=1)
```

```
not_uni = data[data['is_uni']==0].loc[:, 'up&down'].dropna()
is_uni = data[data['is_uni']==1].loc[:, 'up&down'].dropna()
def better():
    if not_uni.mean() < is_uni.mean():
        return 'non-university town'
    else:
        return 'university town'
p_val = list(ttest_ind(not_uni, is_uni))[1]
result = (True, p_val, better())
return result
```