

✓ Reason Three

Excessive Speculation

If houses were purely consumption goods like cars, food, etc., then explaining house prices purely based on supply and demand would be sufficient because the only economic factors affecting house prices would be the number of people who want a house and the number of houses available, unfortunately sometimes people treat housing as an investment and buy up houses expecting the prices to rise, when many people start doing this it drives up housing prices and causes a housing bubble.

One quick way to get an idea about whether we are experiencing a housing bubble is to look at housing vacancies. We should assume that vacancies should be decreasing as housing costs go up. But this doesn't always happen though during the 2009 bubble massive house speculation increased housing prices despite the fact that more properties were left empty, vacancy rates in the U.S. peaked at 11.1% in 2009 and has been trending down since.

✓ Vacancy Insight

A Vacancy Rate of 8% is probably not contributing significantly to the rising housing costs but there certainly are some real estate investors speculating on the housing market.

On average Iowa homes are getting sold faster than homes across the entire U.S. meaning there is a strong demand for homes in Iowa relative to the supply. The solid demand is likely why vacancy rates are at a reasonable level.

> Vacancy Rate Code and Graph

[Show code](#)

```
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import FuncFormatter

# Create the plot
plt.figure(figsize=(12, 6))
plot = sns.lineplot(data=summary_data, x='year', y='vacancies')
plot.set_title("Iowa Vacancy Rates", fontsize=16)
plot.set_xlabel("Year", fontsize=14)
plot.set_ylabel("% of vacant houses", fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

# Formatting
plot.yaxis.set_major_formatter(FuncFormatter(lambda y, _: f'{int(y)}%'))
plt.grid(False)

# Source
plt.text(0.95, 0.01, 'Source: U.S. Census Bureau',
        verticalalignment='bottom', horizontalalignment='right',
        transform=plt.gca().transAxes,
        color='gray', fontsize=8)

plt.show()
```

