#### 0.0.1 Question 2c: Verify Outcome

Did the candidate win or lose the election? Verify with election outcome.

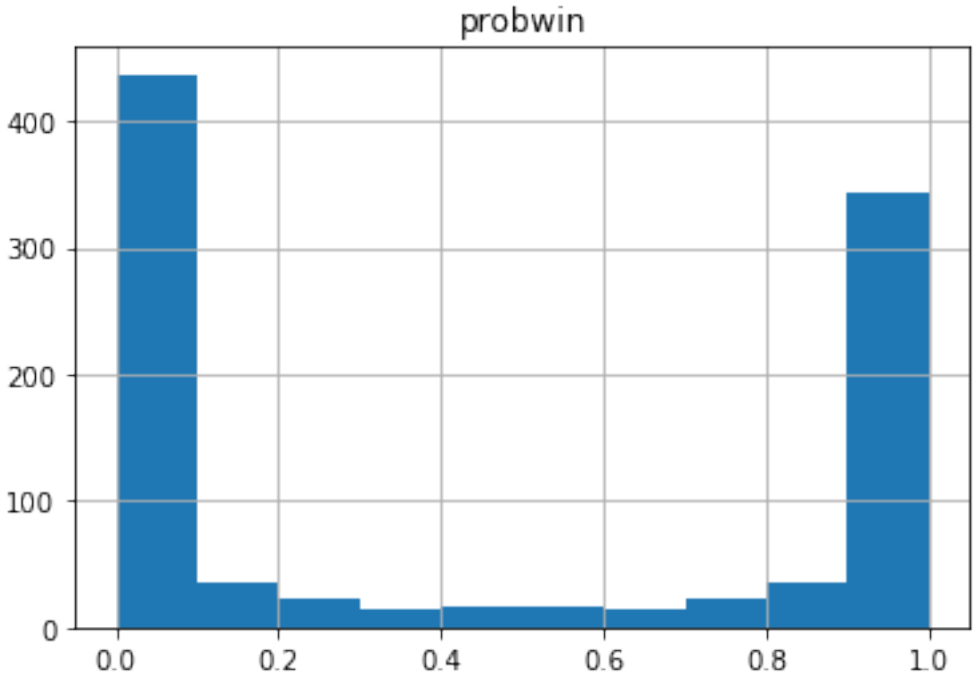
*Type your answer here, replacing this text.*

**SOLUTION**

### 0.0.2 Question 3a: Prediction Histogram

Make a histogram showing the predicted win probabilities *on the morning of the election.* Again, restrict yourself to only the `classic` predictions.

```
In [24]: # BEGIN SOLUTION
         election_sub.query("forecast_date=='2018-11-06'").hist('probwin');
         # END SOLUTION
```



#### 0.0.3 Question 3b: Prediction difficulty

Are most house elections easy to forecast or hard to forecast? State your reasoning.

*Type your answer here, replacing this text.*

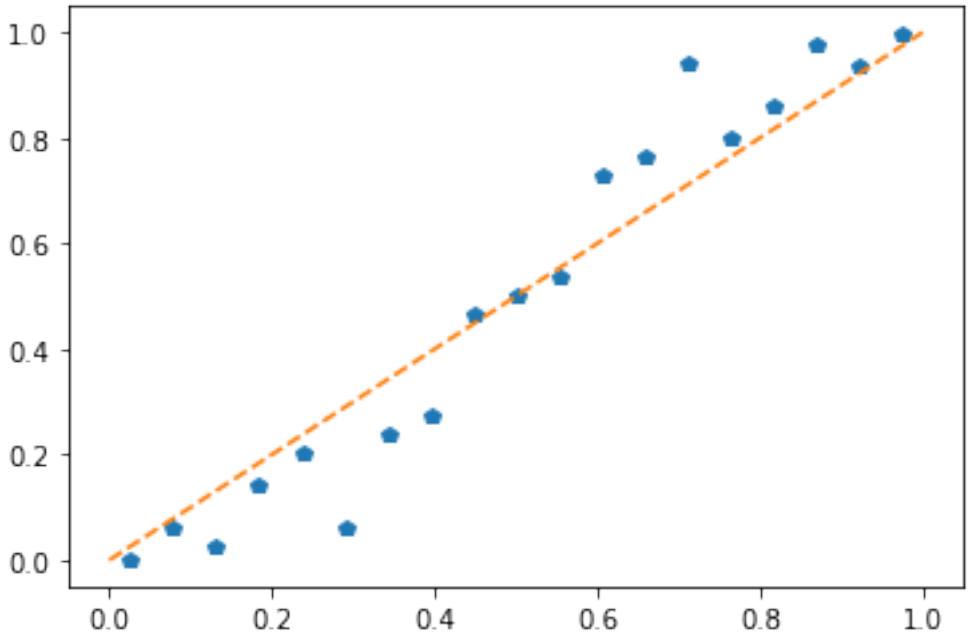
**SOLUTION**

### 0.0.4 Question 4c: Visualize Results

Now make a scatterplot using `midpoints` as the x variable and `fraction_outcome` as the y variable. Draw a dashed line from `[0,0]` to `[1,1]` to mark the line y=x.

```
In [33]: # magic for showing figures inline
         %matplotlib inline
         import matplotlib.pyplot as plt

         # BEGIN SOLUTION
         plt.plot(midpoints, fraction_outcome, 'p')
         plt.plot([0, 1], [0, 1], '--');
         # END SOLUTION
```

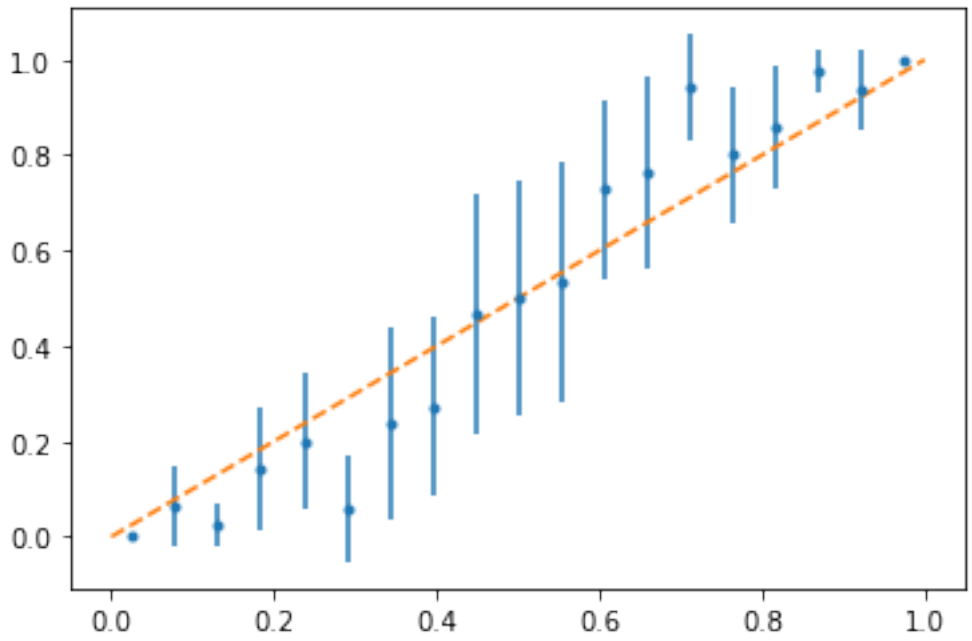


### 0.0.5 Question 5b: Visualize Error Bars 1

Use `plt.errorbar` to create a new plot with error bars associated with the actual fraction of wins in each bin. Again add a dashed y=x line. Set the argument `fmt='.'` to create a scatterplot with errorbars.

In [38]:
```
# Plotting code below

# BEGIN SOLUTION
plt.errorbar(midpoints, election_agg['mean'].values, yerr=election_agg['err'].values, fmt='.')
plt.plot([0, 1], [0, 1], '--');
# END SOLUTION
```



#### 0.0.6 Question 5d: Understanding Confidence Intervals

Are the 95% confidence intervals generally larger or smaller for more confident predictions (e.g. the predictions closer to 0 or 1). What are the factors that determine the length of the confidence intervals?

*Type your answer here, replacing this text.*

**SOLUTION**