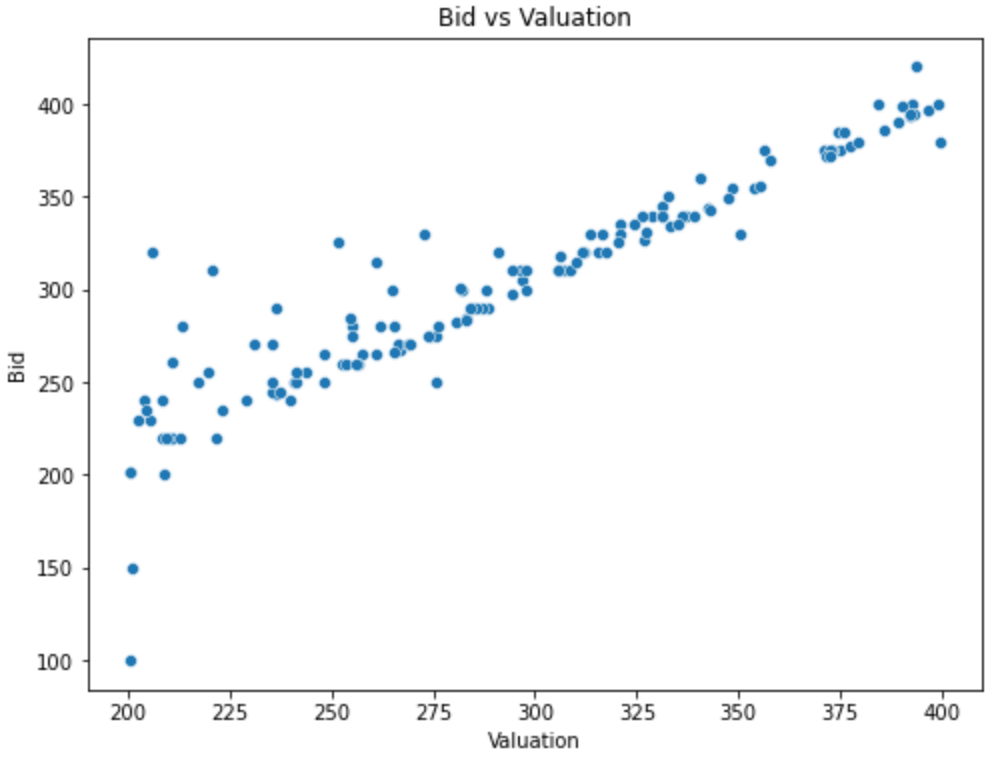
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.stats.contrast import ContrastResults
import seaborn as sns
```

In [2]:
```
# Load the data
file_path = 'C:/Users/pxue/Downloads/FPSBFourPlayers_11_03_25.xlsx'
fpsb = pd.read_excel(file_path)

fpsb.rename(columns={
    'Costs (v)': 'valuation',
    "Subject's bid": 'bid',
    'Price (Lowest bid)': 'winning_bid'
}, inplace=True)

# Check the renamed columns
print(fpsb.columns)
fpsb.dropna(inplace=True)
```

```
Index(['year', 'Session', 'Auction type', 'Subject login name', 'No. bidders',
       'valuation', 'bid', 'winning_bid', 'Profit'],
      dtype='object')
```

Qa

In [3]:
```
#Q(a)
fpsb['valuation'] = pd.to_numeric(fpsb['valuation'], errors='coerce')
fpsb['bid'] = pd.to_numeric(fpsb['bid'], errors='coerce')

# Drop rows with NaN values in 'valuation' or 'bid'
fpsb.dropna(subset=['valuation', 'bid'], inplace=True)

# Plot valuation against bid
plt.figure(figsize=(8, 6))
sns.scatterplot(data=fpsb, x='valuation', y='bid')
plt.xlabel('Valuation')
plt.ylabel('Bid')
plt.title('Bid vs Valuation')
plt.show()
```



Qb The graph should then be a upward-sloping straght line.

Qc

In [5]:
```
#Qc
X = fpsb['valuation']
X = sm.add_constant(X)
y = fpsb['bid']

# OLS regression
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    bid   R-squared:                       0.872
Model:                            OLS   Adj. R-squared:                  0.871
Method:                 Least Squares   F-statistic:                     967.6
Date:                Mon, 17 Nov 2025   Prob (F-statistic):           2.89e-65
Time:                        18:49:35   Log-Likelihood:                -637.66
No. Observations:                 144   AIC:                             1279.
Df Residuals:                     142   BIC:                             1285.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         37.1201      8.701      4.266      0.000      19.920      54.320
valuation      0.9090      0.029     31.106      0.000       0.851       0.967
==============================================================================
Omnibus:                       43.442   Durbin-Watson:                   2.253
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              900.235
Skew:                          -0.228   Prob(JB):                    3.28e-196
Kurtosis:                      15.241   Cond. No.                     1.52e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
[2] The condition number is large, 1.52e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [6]:
```
hypothesis_test_f = model.f_test("const = 100, valuation = 0.75")
print(hypothesis_test_f)
```

```
<F test: F=61.59203388800687, p=5.50307018520001e-20, df_denom=142, df_num=2>
```

In theory, the optimal bidding for a bidder in FPSB with valuation v_i is $b(v_i) = \frac{L}{N} + \frac{v_i(N-1)}{N}$. where L is the lowest valuation (in this case, the highest cost).

Our game has $L = 400$ and $N = 4$. So if the intercept and the slope coefficient behave according to the theoretical prediction, we would expect $b_0 = 400/4 = 100$ and $b_1 = \frac{3}{4}$. We can perform a F-test with null hypothesis H_0: $b_0 = 100$ AND $b_1 = \frac{3}{4}$. p-value is $<< 0.05$, so we reject the null hypothesis.

Qd

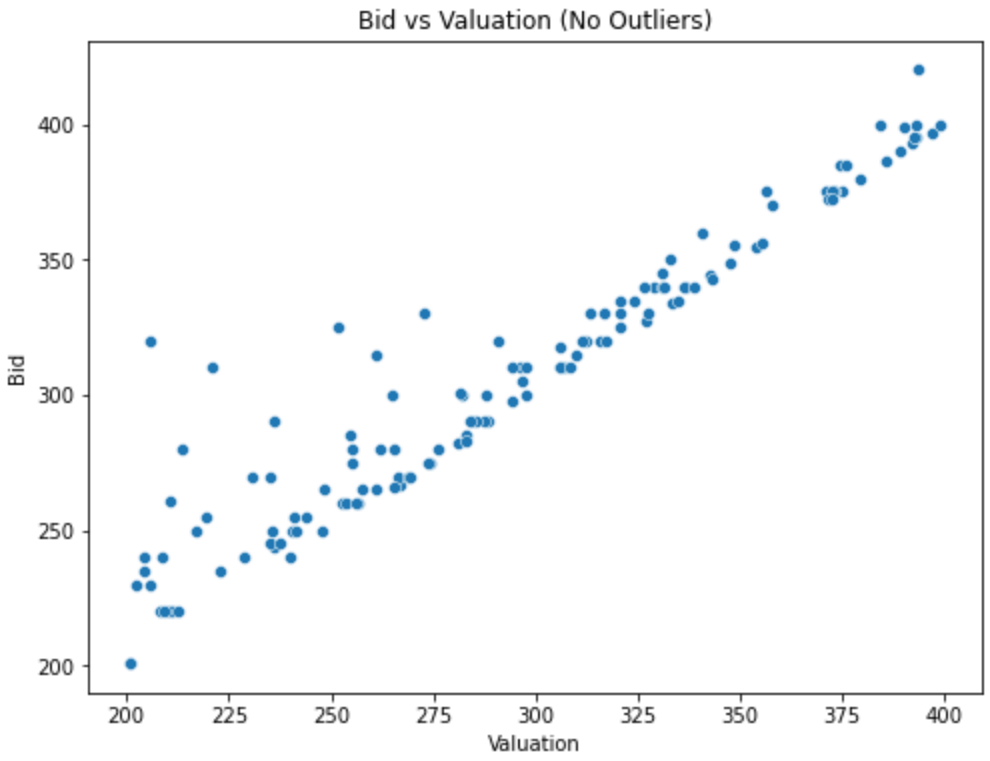
ANSWER: A bidder with valuation v optimally bidding b(v) would have expected payoff $E\pi = p(v)(v - b(v))$ where $p(v)$ is the winning probability, which is equal to the probability that all other valuations are lower than v. Conditional on winning, the bidder gets profit $v - b(v)$.

Qe

ANSWER: An expert should never bid higher than his valuation, because they would lose money if they win. In a procurement (our game), the bid should never be below the cost.

Qf

In [7]:
```
fpsb_no_outliers = fpsb[fpsb['bid'] > fpsb['valuation']]
plt.figure(figsize=(8, 6))
sns.scatterplot(data=fpsb_no_outliers, x='valuation', y='bid')
plt.xlabel('Valuation')
plt.ylabel('Bid')
plt.title('Bid vs Valuation (No Outliers)')
plt.show()

# OLS
X_no_outliers = fpsb_no_outliers['valuation']
X_no_outliers = sm.add_constant(X_no_outliers)
y_no_outliers = fpsb_no_outliers['bid']
model_no_outliers = sm.OLS(y_no_outliers, X_no_outliers).fit()
print(model_no_outliers.summary())
#F test
hypothesis_test_no_outliers = model_no_outliers.f_test("const = 100, valuation = 0.75"
print("F-test result for joint hypothesis (No Outliers):")
print(hypothesis_test_no_outliers)
```

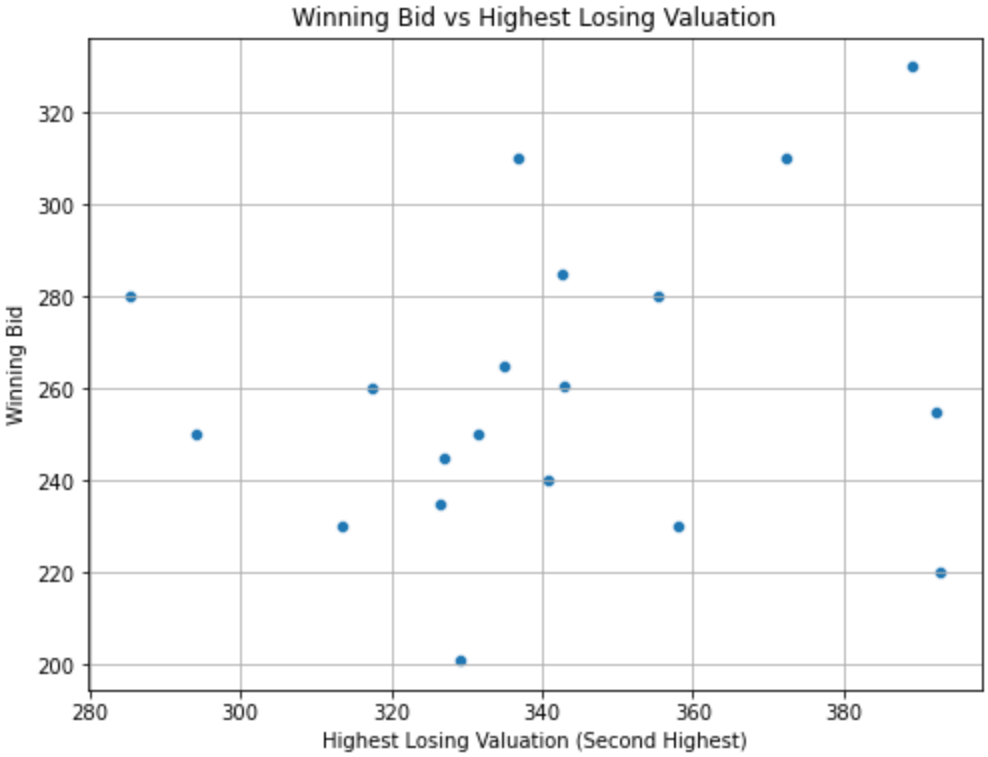


```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    bid   R-squared:                       0.904
Model:                            OLS   Adj. R-squared:                  0.904
Method:                 Least Squares   F-statistic:                     1257.
Date:                Mon, 17 Nov 2025   Prob (F-statistic):           1.20e-69
Time:                        18:50:52   Log-Likelihood:                -566.28
No. Observations:                 135   AIC:                             1137.
Df Residuals:                     133   BIC:                             1142.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         51.6283      7.302      7.070      0.000      37.185      66.072
valuation      0.8679      0.024     35.459      0.000       0.819       0.916
==============================================================================
Omnibus:                       93.026   Durbin-Watson:                   2.085
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              587.751
Skew:                           2.464   Prob(JB):                    2.35e-128
Kurtosis:                      11.955   Cond. No.                     1.57e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
[2] The condition number is large, 1.57e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
F-test result for joint hypothesis (No Outliers):
<F test: F=61.0738477332543, p=1.5292884009639984e-19, df_denom=133, df_num=2>
```

Once we leave out the outliers, the intercept and slope coefficient both get closer to theory prediction: $b_0 = 51.6283$ is closer to 100 than the previous estimate, $b_1 = 0.8679$ is closer to 0.75 than the previous estimate. However, we would still reject the null hypothesis H0 with $p < 0.05$

Qg

In [8]:
```
fpsb_no_outliers = fpsb_no_outliers.sort_values(by=['Session', 'valuation'], ascending
highest_losing = fpsb_no_outliers.groupby('Session').nth(1)  # gets the second row in
highest_losing = highest_losing.reset_index()  # Reset index to make 'Session' a colum

# Plot winning bid against highest losing valuation
plt.figure(figsize=(8, 6))
sns.scatterplot(data=highest_losing, x='valuation', y='winning_bid')
plt.xlabel('Highest Losing Valuation (Second Highest)')
plt.ylabel('Winning Bid')
plt.title('Winning Bid vs Highest Losing Valuation')
plt.grid(True)
plt.show()
```



Qh

In [9]:
```
fpsb_no_outliers['valuation'] = pd.to_numeric(fpsb_no_outliers['valuation'], errors='c
fpsb_no_outliers['winning_bid'] = pd.to_numeric(fpsb_no_outliers['winning_bid'], error

# Perform OLS regression of winning bid on valuation
X = fpsb_no_outliers['valuation']
X = sm.add_constant(X)
y = fpsb_no_outliers['winning_bid']
model = sm.OLS(y, X).fit()
print(model.summary())

# F-test for joint hypothesis: Intercept = 175 and Slope = 0.375
hypothesis_test = model.f_test("const = 175, valuation = 0.375")
print("F-test result for joint hypothesis (Intercept = 175, Slope = 0.375):")
print(hypothesis_test)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            winning_bid   R-squared:                       0.065
Model:                            OLS   Adj. R-squared:                  0.058
Method:                 Least Squares   F-statistic:                     9.245
Date:                Mon, 17 Nov 2025   Prob (F-statistic):            0.00284
Time:                        18:52:21   Log-Likelihood:                -683.70
No. Observations:                 135   AIC:                             1371.
Df Residuals:                     133   BIC:                             1377.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        195.9815     17.426     11.247      0.000     161.514     230.449
valuation      0.1776      0.058      3.041      0.003       0.062       0.293
==============================================================================
Omnibus:                       32.197   Durbin-Watson:                   0.954
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               61.418
Skew:                          -1.052   Prob(JB):                     4.61e-14
Kurtosis:                       5.548   Cond. No.                     1.57e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
[2] The condition number is large, 1.57e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
F-test result for joint hypothesis (Intercept = 175, Slope = 0.375):
<F test: F=67.20869323546401, p=6.730123937765123e-21, df_denom=133, df_num=2>
```

Let v_2 be the highest losing valuation (lowest losing cost), and $\bar{v} = 200$ be the highest possible valuation (lowest possible cost). The predicted winning bid conditional on v_2 is

$$\begin{aligned}
E\left[b(v)|v > v_2\right] &= E\left[\frac{L}{N} + \frac{N-1}{N}v|v > v_2\right] \\
&= \frac{L}{N} + \frac{N-1}{N}E\left[v|v > v_2\right] \\
&= \frac{L}{N} + \frac{N-1}{N}\frac{\bar{v} + v_2}{2} \\
&= 100 + 0.75\frac{200}{2} + 0.75\frac{v_2}{2} \\
&= 175 + 0.375v_2
\end{aligned}$$

So when we regress winning bid on highest losing valuation, theory says the intercept should be 175 and the slope should be 0.375. We can test this using a F-test with H0: $b_0 = 175$ AND $b_1 = 0.375$. Again we reject the null hypothesis because p-value is very small, i.e. the theory is rejected by the data.

Q2

Qa ANSWER: The revenue equivalence theorem states that in private value auctions, the expected revenue does not depend on the auction mechanism. It does not mean that the revenue will be the same every time, only on average.

Qb

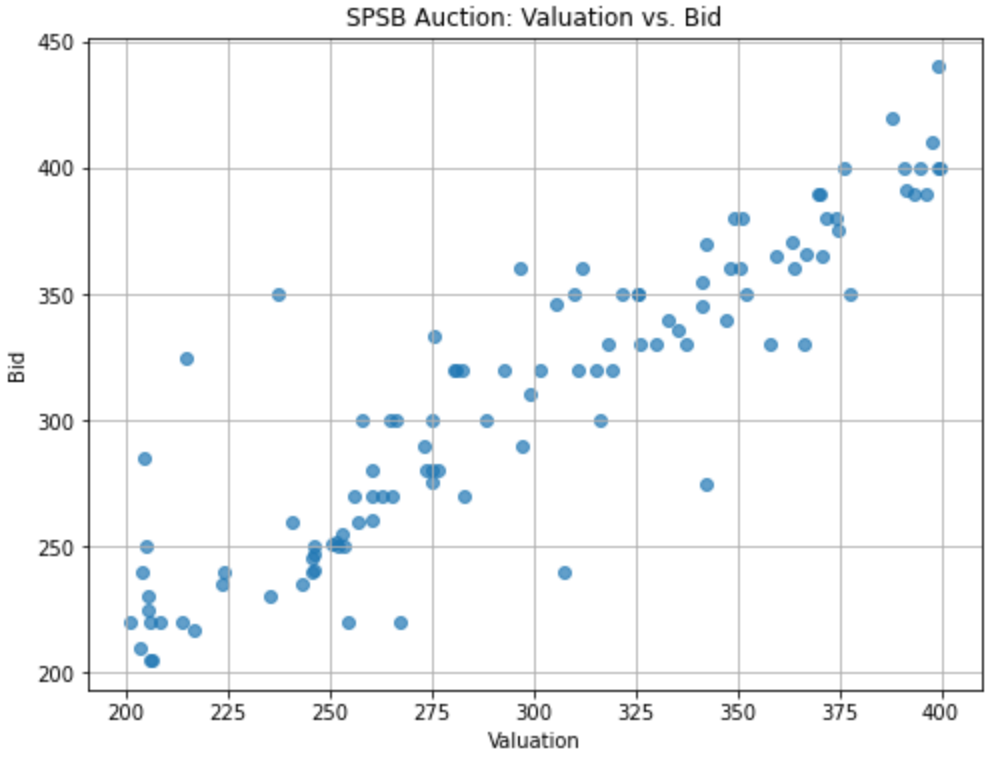
In [11]:
```
# Calculate mean and standard deviation of revenue (winning_bid) for FPSB auction type
fpsb_revenue_mean = fpsb['winning_bid'].mean()
fpsb_revenue_sd = fpsb['winning_bid'].std()

spsb_asc_path = 'C:/Users/pxue/Downloads/SPSBFourPlayers_11_03_25.xlsx'
spsb_asc = pd.read_excel(spsb_asc_path, na_values=['.'])  # Treat "." as NaN

# Prepare SPSB data
spsb = spsb_asc.iloc[:, [1, 3, 4, 5, 6, 7]].copy()  # Select columns
spsb.columns = ['Session', 'valuation', 'bid', 'winning_bid', 'price', 'profit']

# mean and standard deviation of revenue (price) for SPSB auction type
spsb_revenue_mean = spsb['price'].mean()
spsb_revenue_sd = spsb['price'].std()

# Prepare Ascending auction data
ascd = spsb_asc.iloc[:, [1, 3, 8, 9, 10]].copy()  # Select columns
ascd.columns = ['Session', 'valuation', 'bid', 'price', 'profit']

# Drop rows with missing prices (sessions with no bids)
ascd.dropna(subset=['price'], inplace=True)

# mean and standard deviation of revenue (price) for Ascending auction type
ascd_revenue_mean = ascd['price'].mean()
ascd_revenue_sd = ascd['price'].std()

print("FPSB Auction Revenue - Mean:", fpsb_revenue_mean, "SD:", fpsb_revenue_sd)
print("SPSB Auction Revenue - Mean:", spsb_revenue_mean, "SD:", spsb_revenue_sd)
print("Ascending Auction Revenue - Mean:", ascd_revenue_mean, "SD:", ascd_revenue_sd)
```

```
FPSB Auction Revenue - Mean: 244.2383333333333 SD: 42.48189745278347
SPSB Auction Revenue - Mean: 291.82925925925923 SD: 36.469315221391305
Ascending Auction Revenue - Mean: 298.0 SD: 73.2327657178366
```

Qc ANSWER: From the data, FPSB has significantly larger revenue (which means lower price) than the other two.

Qd

In [12]:
```
spsb.dropna(subset=['valuation', 'bid'], inplace=True)
plt.figure(figsize=(8, 6))
plt.scatter(spsb['valuation'], spsb['bid'], alpha=0.7)
plt.xlabel('Valuation')
plt.ylabel('Bid')
plt.title('SPSB Auction: Valuation vs. Bid')
plt.grid(True)
plt.show()
```



Qe

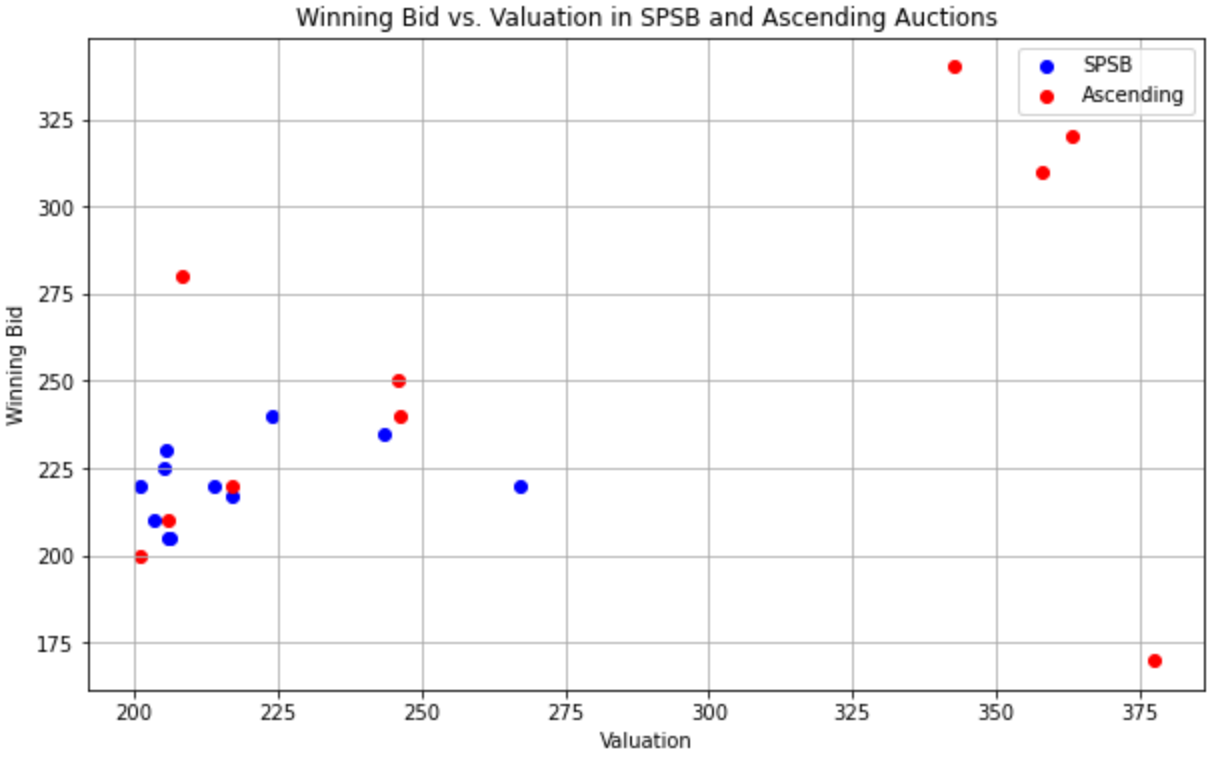
ANSWER: Theory says that one should bid his valuation. Allowing for approximation in bidding (up to 10), about 48.1 percent of the bidders followed the theory.

In [13]:
```
spsb['optimal'] = abs(spsb['bid'] - spsb['valuation']) < 10
#Calculate the percentage of bidders who followed the theoretical prediction
percentage_following_theory = spsb['optimal'].mean() * 100
print(f"Percentage of bidders following theoretical prediction: {percentage_following_
```

```
Percentage of bidders following theoretical prediction: 49.09%
```

Qf

In [14]:
```
# Obtain the winning valuation and bid in each session of SPSB
winning_spsb = spsb.sort_values(by=['Session', 'bid']).groupby('Session').first().rese
winning_ascd = ascd[ascd['bid'] > 0].sort_values(by=['Session', 'bid']).groupby('Sessi

# Plot
plt.figure(figsize=(10, 6))
plt.scatter(winning_spsb['valuation'], winning_spsb['bid'], color='blue', label='SPSB'
plt.scatter(winning_ascd['valuation'], winning_ascd['bid'], color='red', label='Ascend
plt.xlabel('Valuation')
plt.ylabel('Winning Bid')
plt.title('Winning Bid vs. Valuation in SPSB and Ascending Auctions')
plt.legend()
plt.grid(True)
plt.show()
```



Qg

ANSWER: The SPSB auction generally results in winning bids that are more closely tied to the valuation, while the ascending auction does not. In theory, in SPSB auction, it is a dominant strategy for every bidder to bid their true valuation. But in an ascending auction, the winning bidder will not actually get to bid at their valuation, since the auction stops when the bidder with the second highest valuation drops out, not when the winner states her bid. The winner in this case never reveals their true valuation, not even to the auctioneer.