In [1]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import scipy.stats
```

In [2]:
```
cars = pd.read_csv('imports-85.csv')
```

Question 1

Use the imports-85.csv dataset available at BruinLearn (Week 1). The data are taken from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php.

a.

Use matplotlib to visualize the relationship between price and horsepower and body style. Price is the dependent variable. Consider both the "log()" and "^2" transformations of price as dependent variables. Does the body style variable appear to be relevant for car prices, above and beyond horsepower?

- Body style does appear to be relevant as certain body styles appear to have more horsepower and priced higher.

In [3]:
```
cars['log_value'] = np.log(cars['price'])
cars.head(5)
```

Out[3]:

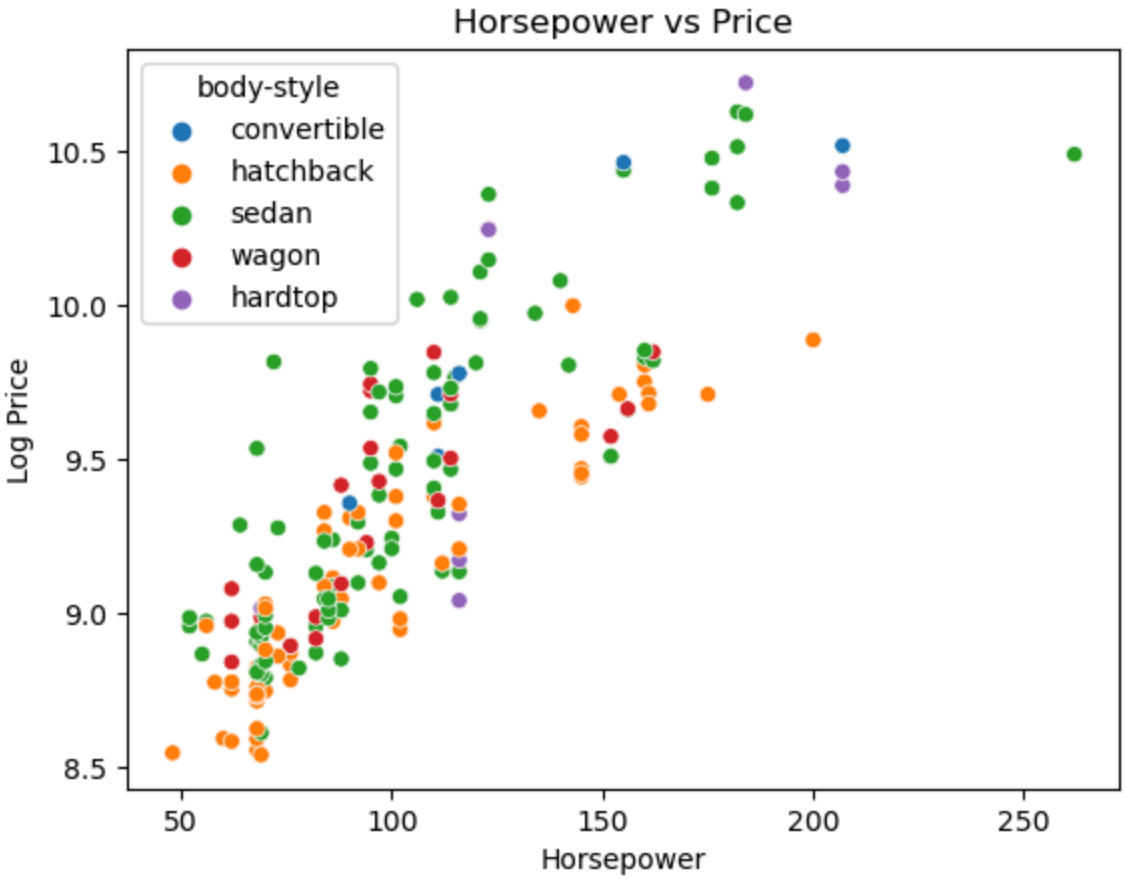
	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engi locat
0	3	NaN	alfa-romero	gas	std	two	convertible	rwd	fr
1	3	NaN	alfa-romero	gas	std	two	convertible	rwd	fr
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd	fr
3	2	164.0	audi	gas	std	four	sedan	fwd	fr
4	2	164.0	audi	gas	std	four	sedan	4wd	fr

5 rows × 27 columns

In [4]:
```
# Plot log price vs horsepower
plt.figure()
ax = sns.scatterplot(
    x = 'horsepower',
    y = 'log_value',
```

```
    data = cars,
    hue = 'body-style'
)
ax.set(xlabel = 'Horsepower',
       ylabel = 'Log Price',
       title = 'Horsepower vs Price')
```

Out[4]:
```
[Text(0.5, 0, 'Horsepower'),
 Text(0, 0.5, 'Log Price'),
 Text(0.5, 1.0, 'Horsepower vs Price')]
```



b.

Run a regression of your preferred specification. Perform residual diagnostics as you learned in Econometrics. What do you conclude from your regression diagnostic plots of residuals vs. fitted and residuals vs. horsepower? (Hint: You may want to use a seaborn_qqplot and scipy.stats packages for qqplot)

- Seems like the dispersion is random, meaning the model should be accurate.

In [5]:
```
# clean df
cars = cars.dropna(subset=['horsepower', 'log_value'])
y = cars['log_value']
x = cars['horsepower']
x = sm.add_constant(x)
# run regression
```

```
reg_model = sm.OLS(y, x).fit()
reg_model.summary()
```

Out[5]:

OLS Regression Results

Dep. Variable:	log_value	**R-squared:**	0.694
Model:	OLS	**Adj. R-squared:**	0.693
Method:	Least Squares	**F-statistic:**	446.9
Date:	Tue, 02 Apr 2024	**Prob (F-statistic):**	1.47e-52
Time:	16:44:02	**Log-Likelihood:**	-27.845
No. Observations:	199	**AIC:**	59.69
Df Residuals:	197	**BIC:**	66.28
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	8.1949	0.058	140.772	0.000	8.080	8.310
horsepower	0.0112	0.001	21.140	0.000	0.010	0.012

Omnibus:	11.085	**Durbin-Watson:**	0.663
Prob(Omnibus):	0.004	**Jarque-Bera (JB):**	11.966
Skew:	0.598	**Prob(JB):**	0.00252
Kurtosis:	2.897	**Cond. No.**	323.

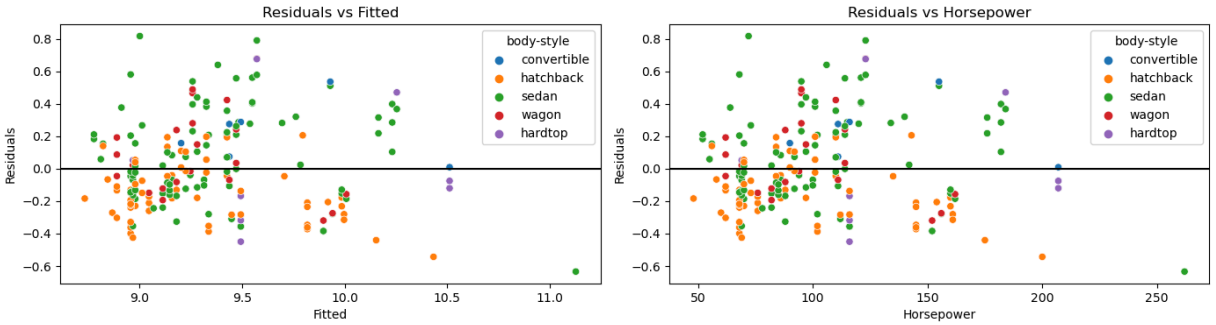
Notes:

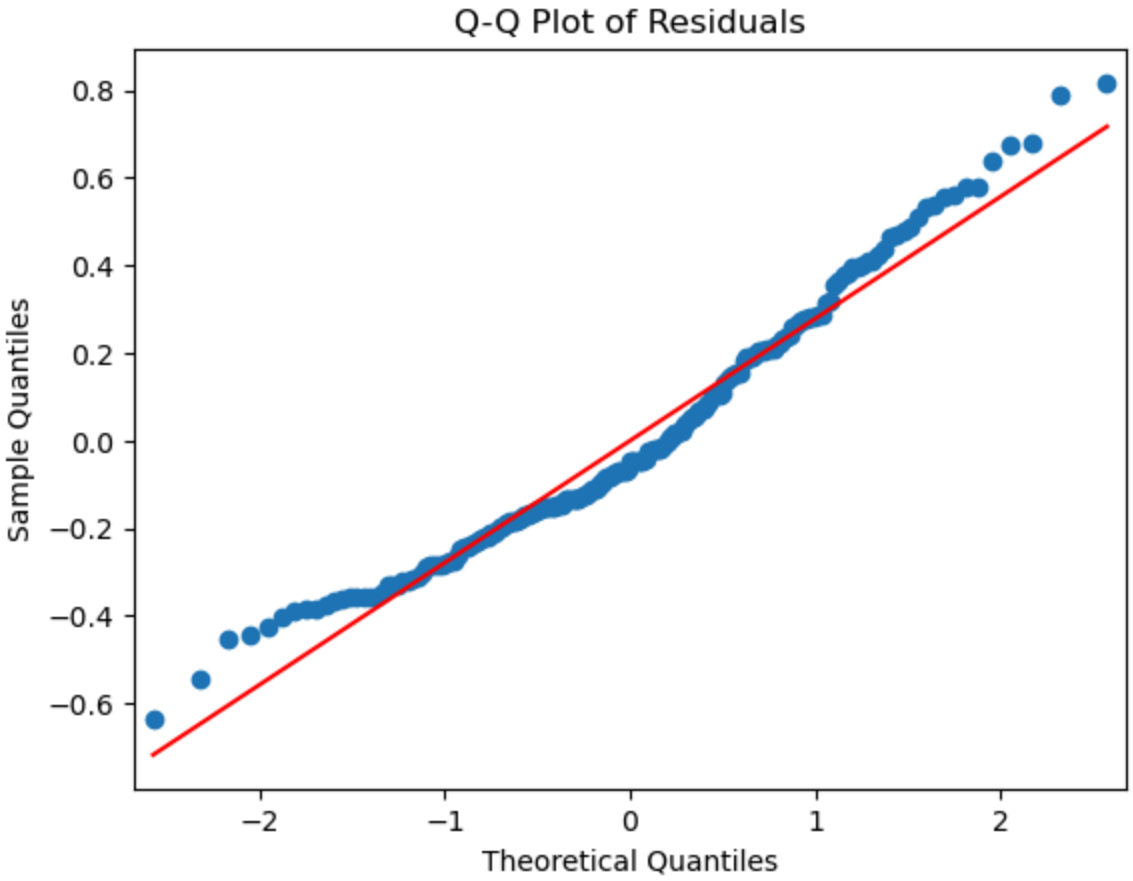
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [6]:

```
# save fitted and residuals
cars['fitted'] = reg_model.fittedvalues
cars['residuals'] = reg_model.resid
# plot fitted vs residuals
plt.figure(figsize=(14, 7))
plt.subplot(2, 2, 1)
sns.scatterplot(x='fitted', y='residuals', data=cars, hue='body-style')
plt.axhline(y=0, color='black', linestyle='-')
plt.xlabel('Fitted')
plt.ylabel('Residuals')
plt.title('Residuals vs Fitted')
# plot residuals vs horsepower
plt.subplot(2, 2, 2)
sns.scatterplot(x='horsepower', y='residuals', data=cars, hue='body-style')
plt.axhline(y=0, color='black', linestyle='-')
plt.xlabel('Horsepower')
```

```
plt.ylabel('Residuals')
plt.title('Residuals vs Horsepower')
plt.tight_layout()
plt.show()
# plot qqplot with residuals
fig = sm.qqplot(cars['residuals'], line='s')
plt.title('Q-Q Plot of Residuals')
plt.show()
```





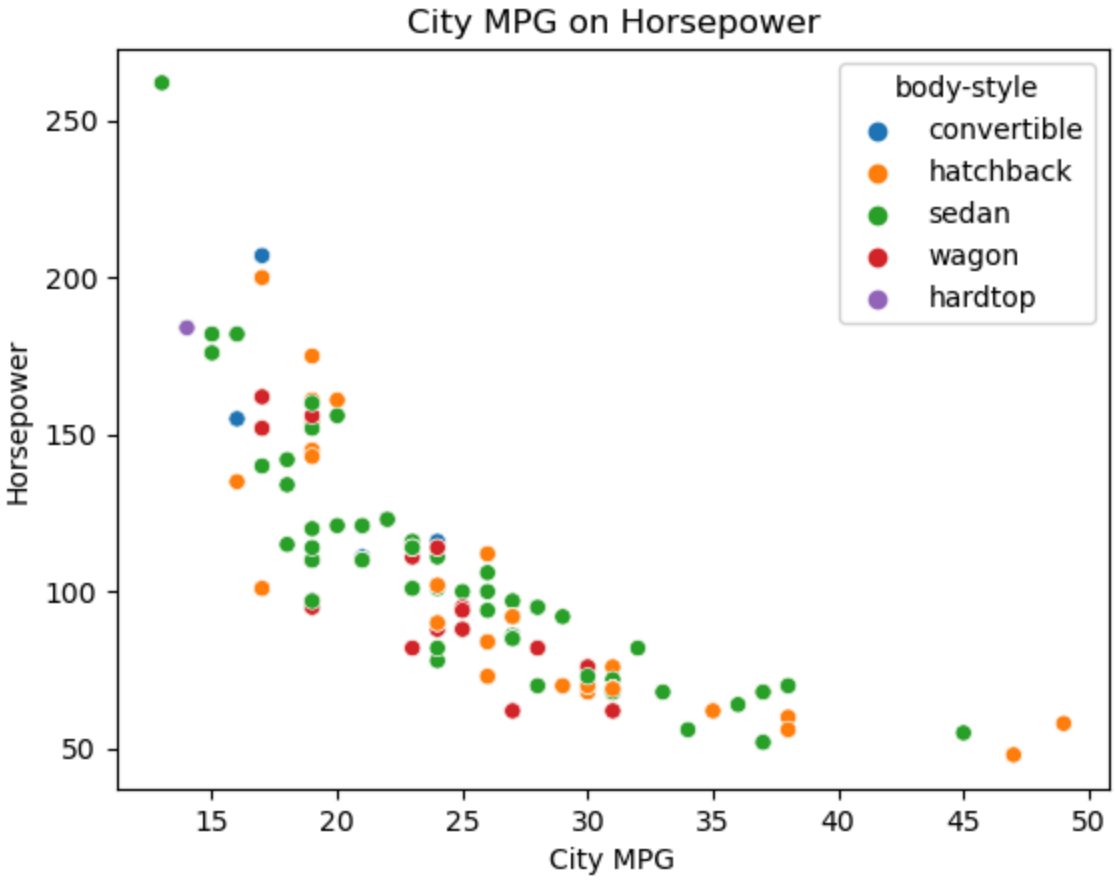
c.

Now use matplotlib to visualize the relationship between fuel efficiency (city-mpg) and horsepower. Now regress city.mpg on horsepower. Is the regression result consistent with the conclusion you would draw based on the plot? More on this next week.

- Seems consistent as increase in fuel efficiency reduces horsepower

In [9]:
```
# plot city mpg on horsepower
plt.figure()
ax = sns.scatterplot(
    x = 'city-mpg',
    y = 'horsepower',
    data = cars,
    hue = 'body-style'
)
ax.set(xlabel = 'City MPG',
       ylabel = 'Horsepower',
       title = 'City MPG on Horsepower')
```

Out[9]:
```
[Text(0.5, 0, 'City MPG'),
 Text(0, 0.5, 'Horsepower'),
 Text(0.5, 1.0, 'City MPG on Horsepower')]
```



In [10]:
```
y = cars['horsepower']
x = cars['city-mpg']
x = sm.add_constant(x)
# run regression
reg_model = sm.OLS(y, x).fit()
reg_model.summary()
```

Out[10]:

OLS Regression Results

Dep. Variable:	horsepower	**R-squared:**	0.677
Model:	OLS	**Adj. R-squared:**	0.675
Method:	Least Squares	**F-statistic:**	412.3
Date:	Tue, 02 Apr 2024	**Prob (F-statistic):**	3.42e-50
Time:	16:48:07	**Log-Likelihood:**	-891.04
No. Observations:	199	**AIC:**	1786.
Df Residuals:	197	**BIC:**	1793.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	224.0636	6.133	36.534	0.000	211.969	236.159
city-mpg	-4.7882	0.236	-20.306	0.000	-5.253	-4.323

Omnibus:	51.089	**Durbin-Watson:**	1.127
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	114.155
Skew:	1.170	**Prob(JB):**	1.63e-25
Kurtosis:	5.880	**Cond. No.**	105.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Question 2 : Nonlinear relations

A common concern is that the relationship between a predictive variable (X) and the outcome we are trying to predict (Y) is nonlinear. On the surface, this seems to invalidate linear regressions, such as the Fama-MacBeth regression. However, this is not generally the case. For instance, if Y = f(X) + noise, where f(.) is not linear in X, simply define a transformation of X as, generally, Z = a + bf(X). Now, it is clear that Y = a1 + b1*Z, for constants a, a1, b, and b1. In other words, one could include squared values of X in the regression, perhaps max(0,X), etc. We will see this in action for the case of Issuance (lnIssue). This is the average amount of stock issuance in the last 36 months, normalized by market equity. Generally, firms that issue a lot of equity have low returns going forward.

a.

Construct decile sorts (10 portfolios) as in the class notes, but now based on the issuance variable lnIssue. Give the average return to each decile portfolio, value-weighting stocks within each portfolio each year, equal-weighting across years.

In [39]:
```
df = pd.read_csv('StockRetAcct_DT.csv')
df.head(5)
```

Out[39]:

	Unnamed: 0	FirmID	year	lnAnnRet	lnRf	MEwt	lnIssue	lnMom	
0	1	6	1980	0.363631	0.078944	0.000281	0.031344	0.075355	12
1	2	6	1981	-0.290409	0.130199	0.000321	0.044213	0.512652	12
2	3	6	1982	0.186630	0.130703	0.000266	-0.068195	-0.220505	12
3	4	6	1983	0.489819	0.089830	0.000170	-0.071780	0.046218	12
4	5	10	1991	-0.508005	0.061216	0.000033	0.115204	1.341053	1'

In [40]:
```
df['decile'] = pd.qcut(df['lnIssue'], 10, labels=np.arange(1, 11, 1))
# Group by year and decile
grouped_df = df.groupby(['year', 'decile'])
value_weighted_returns = grouped_df.apply(lambda x: (x['lnAnnRet'] * x['MEwt
# Reset the index to manipulate the DataFrame
value_weighted_returns = value_weighted_returns.reset_index(name='ValueWeigh
# Now calculate the average return across years for each decile
average_returns = value_weighted_returns.groupby('decile')['ValueWeightedRet
average_returns
```

Out[40]:
```
decile
1     0.124678
2     0.091098
3     0.091538
4     0.082846
5     0.074974
6     0.079672
7     0.075648
8     0.038892
9     0.031810
10    0.009873
Name: ValueWeightedReturn, dtype: float64
```

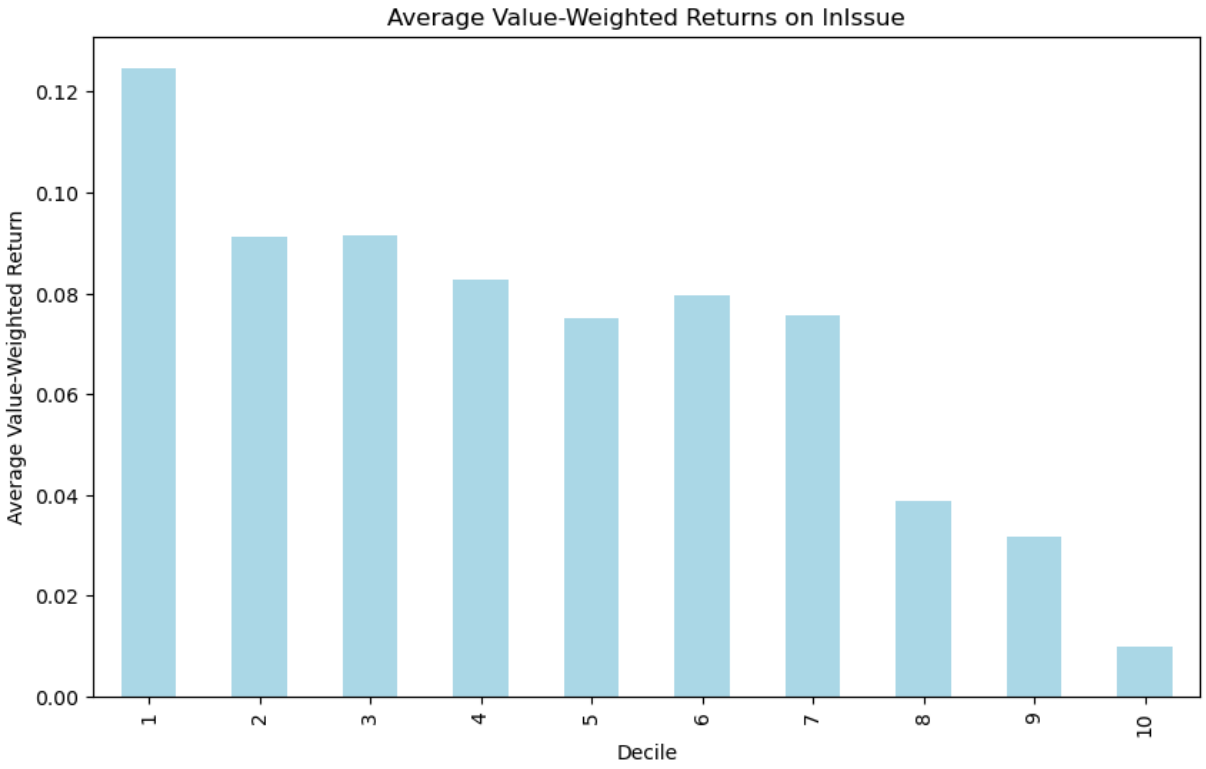
b.

Plot the average return to these 10 portfolios, similar to what we did in the Topic 1(e-f) notes. Discuss whether the pattern seems linear or not.

- The pattern does seem linear

In [41]:
```
plt.figure(figsize=(10, 6))
average_returns.plot(kind='bar', color='lightblue')
plt.title('Average Value-Weighted Returns on lnIssue')
plt.xlabel('Decile')
```

```
plt.ylabel('Average Value-Weighted Return')
plt.show()
```



c.

Since most of the 'action' is in the extreme portfolios, consider a model where expected returns to stocks is linear in a transformed issuance-characteristic that takes three values: -1 if the stock's issuance is in Decile 1, 1 if the stock's issuance is in decile 10, and 0 otherwise.

In [42]:
```
# create the transformed issuance variable
df['TransIssue'] = 0
df.loc[df['decile'] == 1, 'TransIssue'] = -1
df.loc[df['decile'] == 10, 'TransIssue'] = 1

# prepare data for Fama-MacBeth regression
fm_data = df.groupby('year').agg(
    AverageReturn=('lnAnnRet', 'mean'),
    TransformedIssue=('TransIssue', 'mean')
).reset_index()

y = fm_data['AverageReturn']
x = fm_data['TransformedIssue']
# run the regression
fm_model = sm.OLS(y, x).fit()
fm_model.summary()
```

Out[42]:

OLS Regression Results

Dep. Variable:	AverageReturn	**R-squared (uncentered):**	0.003
Model:	OLS	**Adj. R-squared (uncentered):**	-0.026
Method:	Least Squares	**F-statistic:**	0.1038
Date:	Wed, 03 Apr 2024	**Prob (F-statistic):**	0.749
Time:	11:42:44	**Log-Likelihood:**	6.7408
No. Observations:	35	**AIC:**	-11.48
Df Residuals:	34	**BIC:**	-9.926
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
TransformedIssue	0.1874	0.582	0.322	0.749	-0.995	1.369

Omnibus:	0.419	**Durbin-Watson:**	1.960
Prob(Omnibus):	0.811	**Jarque-Bera (JB):**	0.153
Skew:	-0.162	**Prob(JB):**	0.926
Kurtosis:	2.988	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Create this transformed issuance variable and run a Fama-MacBeth regression with it. Report the results. What is the nature of the portfolio implied by the Fama-MacBeth regression? That is, what stocks do you go long, short, no position?

- We want to long decile 1 and short decile 10, no position in the rest

Question 3 : Double-sorts and functional forms

In the lecture notes we saw that the value spread is much larger for small stocks. Using this fact, I proposed a model where expected returns are linear in the book-to-market ratio as well as the interaction between book-to-market and size. In other words, holding size constant there is a linear relation between expected stock returns and book-to-market. In this question, we will dig deeper into whether this is a reasonable assumption or not based on visual analysis.

a.

Create independent quintile sorts based on book-to-market (lnBM) and size (lnME). That is create a quintile variable by year for book-to-market and then create a quintile variable by year for size.

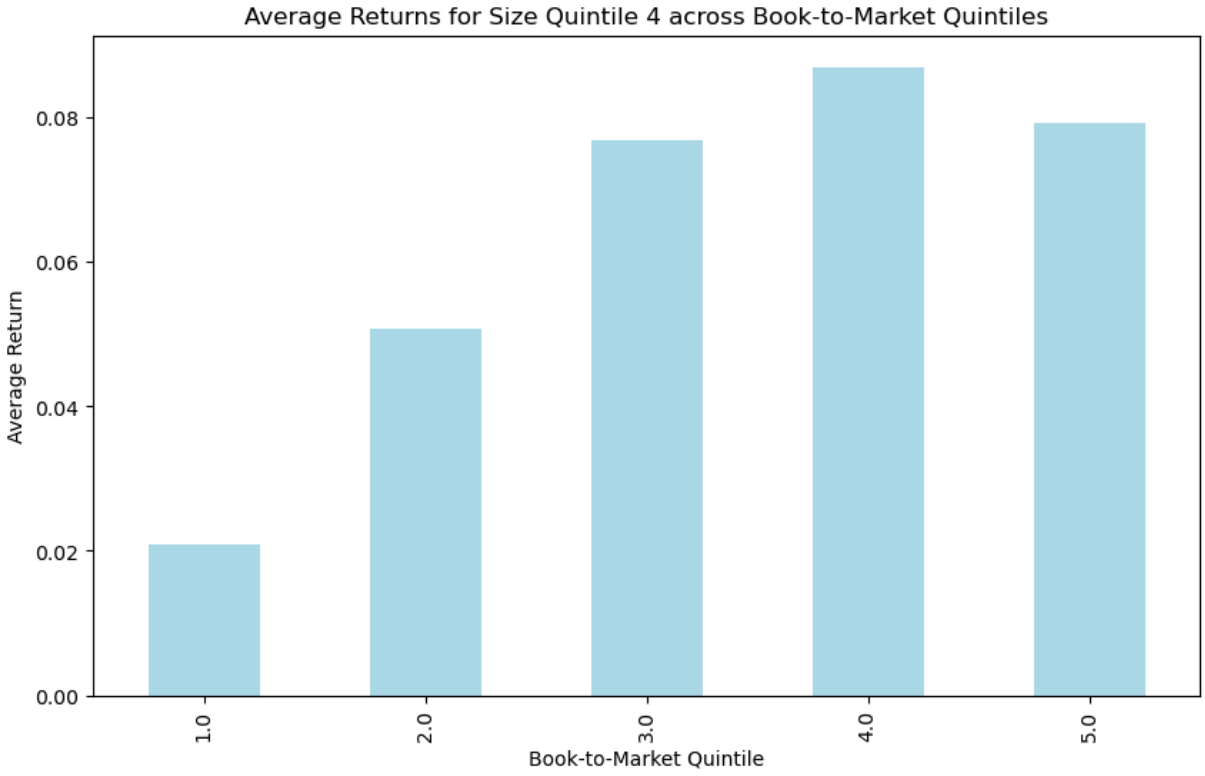
b.

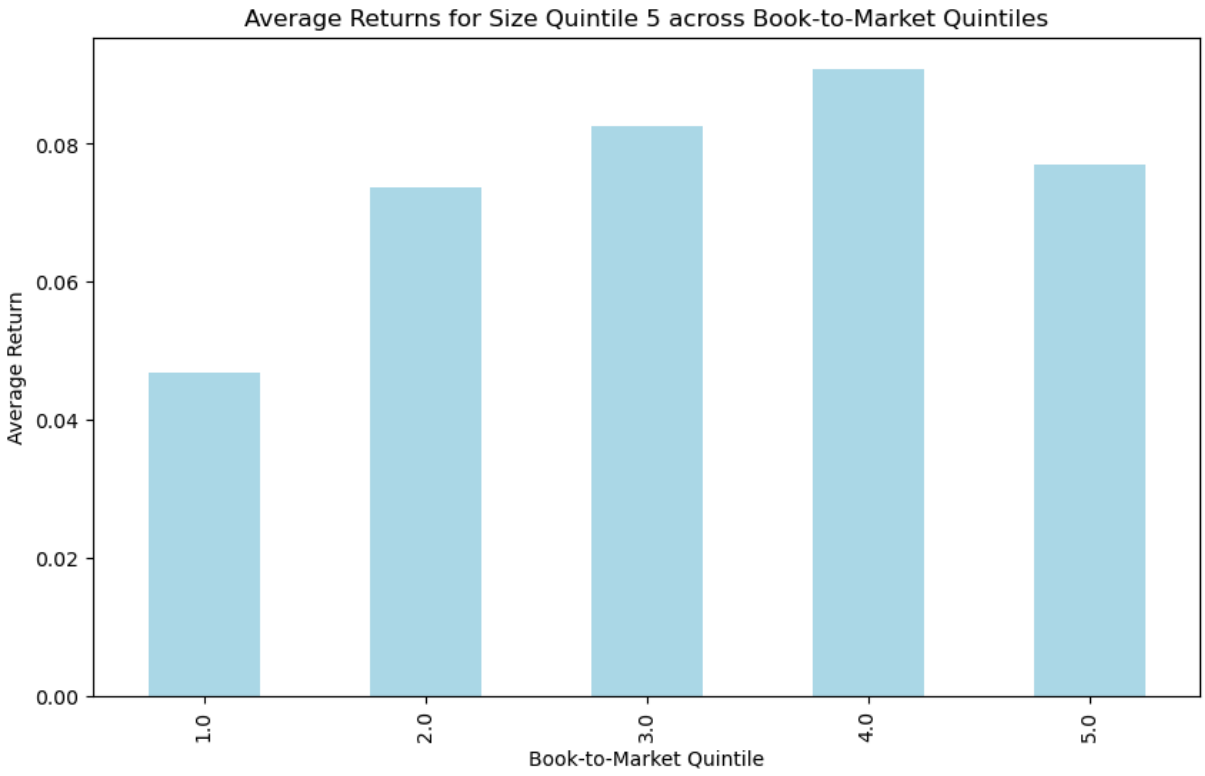
For each size quintile, plot the average returns to the five book-to-market quintile portfolios. So, for size quintile 1, and book-to-market quintile 3, the stocks in this portfolio all have size quintile equal to 1 and book-to-market quintile equal to 3. Thus, I'm looking for five plots here, one for each size quintile.

In [44]:
```
df['lnBM_Quintile'] = df.groupby('year')['lnBM'].transform(lambda x: pd.qcut
df['lnME_Quintile'] = df.groupby('year')['lnME'].transform(lambda x: pd.qcut

# Verify the quintile creation
df[['year', 'lnBM', 'lnBM_Quintile', 'lnME', 'lnME_Quintile']].head()
average_returns = df.groupby(['year', 'lnME_Quintile', 'lnBM_Quintile'])['lr

# For each size quintile, calculate the average return across years for each
size_bm_returns = average_returns.groupby(['lnME_Quintile', 'lnBM_Quintile']

# Plot the average returns for each size quintile across book-to-market quir
for size_quintile in range(1, 6):
    plt.figure(figsize=(10, 6))
    size_bm_returns.loc[size_quintile].plot(kind='bar', color='lightblue')
    plt.title(f'Average Returns for Size Quintile {size_quintile} across Boo
    plt.xlabel('Book-to-Market Quintile')
    plt.ylabel('Average Return')
    plt.show()
```


Average Returns for Size Quintile 2 across Book-to-Market Quintiles
Average Return
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
1.0
2.0
3.0
4.0
5.0
Book-to-Market Quintile


Average Returns for Size Quintile 3 across Book-to-Market Quintiles
Average Return
0.08
0.06
0.04
0.02
0.00
−0.02
1.0
2.0
3.0
4.0
5.0
Book-to-Market Quintile





Does the assumption of conditional linearity seem ok, or would you suggest a different model?

- Relationships between different quantiles does not seem linear although is close to be so