In [187…
```
# Import your packages
import wooldridge as woo
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Now import the package to run regressions
import statsmodels.formula.api as smf
```

Chapter 3 C1

In [190…
```
bwght = woo.dataWoo('bwght')
bwght.describe()
```

Out[190]:

	faminc	cigtax	cigprice	bwght	fatheduc	motheduc	
count	1388.000000	1388.000000	1388.000000	1388.000000	1192.000000	1387.000000	1388.(
mean	29.026657	19.552954	130.559006	118.699568	13.186242	12.935833	1.(
std	18.739285	7.795598	10.244485	20.353964	2.745985	2.376728	0.(
min	0.500000	2.000000	103.800003	23.000000	1.000000	2.000000	1.(
25%	14.500000	15.000000	122.800003	107.000000	12.000000	12.000000	1.(
50%	27.500000	20.000000	130.800003	120.000000	12.000000	12.000000	1.(
75%	37.500000	26.000000	137.000000	132.000000	16.000000	14.000000	2.(
max	65.000000	38.000000	152.500000	271.000000	18.000000	18.000000	6.(

i) The sign for B2 would most likely be positve because higher family income tends to lead to healthier food choices, therefore a healthier pregnancy.

In [193…
```
bwght[["cigs", "faminc"]].corr()
```

Out[193]:

	cigs	faminc
cigs	1.000000	-0.173045
faminc	-0.173045	1.000000

ii) I would have predicted that these explanatory variables are negatively correlated because families with higher income are able to afford more cigarettes, but the reason for the sign would be that families with a higher income care more about health, so the mom is less likely to smoke during pregnancy.

In [196…
```
model1 = smf.ols(formula = "bwght ~ cigs", data=bwght).fit()
display(model1.summary())
intercept1 = model1.params['Intercept']
cigs_coef = model1.params['cigs']
print(f"bwght = {intercept1:.3f} + {cigs_coef:.3f} * cigs")
```

```
print(f'the number of observations is : {model1.nobs}')
print(f'R^2 is :{model1.rsquared:.3f}')
```

OLS Regression Results

Dep. Variable:	bwght	**R-squared:**	0.023
Model:	OLS	**Adj. R-squared:**	0.022
Method:	Least Squares	**F-statistic:**	32.24
Date:	Mon, 07 Oct 2024	**Prob (F-statistic):**	1.66e-08
Time:	17:06:15	**Log-Likelihood:**	-6135.5
No. Observations:	1388	**AIC:**	1.227e+04
Df Residuals:	1386	**BIC:**	1.229e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	119.7719	0.572	209.267	0.000	118.649	120.895
cigs	-0.5138	0.090	-5.678	0.000	-0.691	-0.336

Omnibus:	118.187	**Durbin-Watson:**	1.924
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	635.742
Skew:	-0.156	**Prob(JB):**	8.92e-139
Kurtosis:	6.301	**Cond. No.**	6.72

Notes:

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

```
bwght = 119.772 + -0.514 * cigs
the number of observations is : 1388.0
R^2 is :0.023
```

In [198…

```
model2 = smf.ols(formula = "bwght ~ cigs + faminc", data=bwght).fit()
display(model2.summary())
intercept2 = model2.params['Intercept']
cigs_coef2 = model2.params['cigs']
faminc_coef= model2.params['faminc']
print(f"bwght = {intercept2:.3f} + {cigs_coef2:.3f} * cigs + {faminc_coef:.3f}
print(f' the number of observations is : {model2.nobs}')
print(f' R^2 is :{model2.rsquared:.3f}')
```

OLS Regression Results

Dep. Variable:	bwght	**R-squared:**	0.030
Model:	OLS	**Adj. R-squared:**	0.028
Method:	Least Squares	**F-statistic:**	21.27
Date:	Mon, 07 Oct 2024	**Prob (F-statistic):**	7.94e-10
Time:	17:06:15	**Log-Likelihood:**	-6130.4
No. Observations:	1388	**AIC:**	1.227e+04
Df Residuals:	1385	**BIC:**	1.228e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	116.9741	1.049	111.512	0.000	114.916	119.032
cigs	-0.4634	0.092	-5.060	0.000	-0.643	-0.284
faminc	0.0928	0.029	3.178	0.002	0.036	0.150

Omnibus:	116.751	**Durbin-Watson:**	1.922
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	619.781
Skew:	-0.154	**Prob(JB):**	2.61e-135
Kurtosis:	6.259	**Cond. No.**	67.4

Notes:

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

```
bwght = 116.974 + −0.463 ∗ cigs + 0.093 ∗ faminc
 the number of observations is : 1388.0
 R^2 is :0.030
```

iii) Faminc does not make a major difference on the effect on birth weight for newborn children. R^2 for both models are roughly the same, and so are most of the other parameters.

Chapter 3 C2

In [202…

```
hprice1 = woo.dataWoo('hprice1')
hprice1.describe()
```

Out[202]:

	price	assess	bdrms	lotsize	sqrft	colonial	lprice
count	88.000000	88.000000	88.000000	88.000000	88.000000	88.000000	88.000000
mean	293.546034	315.736362	3.568182	9019.863636	2013.693182	0.693182	5.633180
std	102.713445	95.314435	0.841393	10174.150414	577.191583	0.463816	0.303573
min	111.000000	198.699997	2.000000	1000.000000	1171.000000	0.000000	4.709530
25%	230.000000	253.900002	3.000000	5732.750000	1660.500000	0.000000	5.438079
50%	265.500000	290.199997	3.000000	6430.000000	1845.000000	1.000000	5.581613
75%	326.250000	352.125000	4.000000	8583.250000	2227.000000	1.000000	5.787642
max	725.000000	708.599976	7.000000	92681.000000	3880.000000	1.000000	6.586172

In [204…

```
model3 = smf.ols(formula = "price ~ sqrft + bdrms", data=hprice1).fit()
display(model3.summary())
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.632
Model:	OLS	**Adj. R-squared:**	0.623
Method:	Least Squares	**F-statistic:**	72.96
Date:	Mon, 07 Oct 2024	**Prob (F-statistic):**	3.57e-19
Time:	17:06:16	**Log-Likelihood:**	-488.00
No. Observations:	88	**AIC:**	982.0
Df Residuals:	85	**BIC:**	989.4
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-19.3150	31.047	-0.622	0.536	-81.044	42.414
sqrft	0.1284	0.014	9.291	0.000	0.101	0.156
bdrms	15.1982	9.484	1.603	0.113	-3.658	34.054

Omnibus:	25.221	**Durbin-Watson:**	1.858
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	44.973
Skew:	1.122	**Prob(JB):**	1.72e-10
Kurtosis:	5.689	**Cond. No.**	9.85e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.85e+03. This might indicate that there are strong multicollinearity or other numerical problems.

i) The equation for this regression is below

In [207…
```
price_int = model3.params['Intercept']
sqrft_coeff = model3.params['sqrft']
bdrms_coeff= model3.params['bdrms']
print(f'The equation for this regression is y = {price_int} + {sqrft_coeff} * 
```

```
The equation for this regression is y = -19.314995765185053 + 0.12843621036829
217 * sqrft + 15.19819096782205 * bdrms
```

ii) The estimated increase in price for a house adding one more bedroom while keepig sqrft constant would be +$15,198

iii) For this problem, you just do the coefficient of bdrms plus the coefficient of sqrft times 140. I'll do this in the code below

In [211…
```
answer = bdrms_coeff + (sqrft_coeff * 140)
print(f'The price increase is ${answer*1000:.2f}')
```

```
The price increase is $33179.26
```

iv) For this question, we look at R^2 for this model, which is 0.632. This means that 63.2% of the variation in price is explained by sqrft and bdrms.

v) I will calculate the estimated price of the house below

In [215…
```
house_price = price_int + (sqrft_coeff * 2438) + (bdrms_coeff * 4)
print(f'The estimated price of the house is: ${house_price*1000:.2f}')
```

```
The estimated price of the house is: $354605.25
```

vi) I will calculate the redidual below

In [218…
```
house_resid = house_price - 300
print(f' The buyer underpaid for the house because the estimatd price of the h
```

```
 The buyer underpaid for the house because the estimatd price of the house is
$354605.25, but they paid $300,000. They saved $54605.25
```

Chapter 3 C4

In [221…
```
attend = woo.dataWoo('attend')
attend.describe()
```

Out[221]:

	attend	termGPA	priGPA	ACT	final	atndrte	hwrt
count	680.000000	680.000000	680.000000	680.000000	680.000000	680.000000	674.000000
mean	26.147059	2.601000	2.586775	22.510294	25.891176	81.709559	87.90801
std	5.455037	0.736586	0.544714	3.490768	4.709835	17.046991	19.26925
min	2.000000	0.000000	0.857000	13.000000	10.000000	6.250000	12.50000
25%	24.000000	2.137500	2.190000	20.000000	22.000000	75.000000	87.50000
50%	28.000000	2.670000	2.560000	22.000000	26.000000	87.500000	100.00000
75%	30.000000	3.120000	2.942500	25.000000	29.000000	93.750000	100.00000
max	32.000000	4.000000	3.930000	32.000000	39.000000	100.000000	100.00000

i) I will calculate the max, min, and avg for all values below

In [224…

```
min_values = attend[['atndrte', 'priGPA', 'ACT']].min()
max_values = attend[['atndrte', 'priGPA', 'ACT']].max()
mean_values = attend[['atndrte', 'priGPA', 'ACT']].mean()
print(f' Min values are: \n{min_values}')
print(f' Max values are: \n{max_values}')
print(f' Averages are: \n{mean_values}')
```

```
 Min values are:
atndrte     6.250
priGPA      0.857
ACT        13.000
dtype: float64
 Max values are:
atndrte    100.00
priGPA       3.93
ACT         32.00
dtype: float64
 Averages are:
atndrte    81.709559
priGPA      2.586775
ACT        22.510294
dtype: float64
```

ii) I wil calculate the regression below, along with the equation. But the intercept does not have any real meaning because it is the expected attendance rate if both the GPA and culmulative GPA are both 0, and that is beyond the range of the explanatory variables.

In [227…

```
attend_int = model4.params['Intercept']
priGPA_coeff = model4.params['priGPA']
act_coeff = model4.params['ACT']

model4 = smf.ols(formula = "atndrte ~ priGPA + ACT", data=attend).fit()
display(model4.summary())
print(f'The equation for this regression is y = {attend_int} + {priGPA_coeff} >
```

OLS Regression Results

Dep. Variable:	atndrte	**R-squared:**	0.291
Model:	OLS	**Adj. R-squared:**	0.288
Method:	Least Squares	**F-statistic:**	138.7
Date:	Mon, 07 Oct 2024	**Prob (F-statistic):**	3.39e-51
Time:	17:06:20	**Log-Likelihood:**	-2776.1
No. Observations:	680	**AIC:**	5558.
Df Residuals:	677	**BIC:**	5572.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	75.7004	3.884	19.490	0.000	68.074	83.327
priGPA	17.2606	1.083	15.936	0.000	15.134	19.387
ACT	-1.7166	0.169	-10.156	0.000	-2.048	-1.385

Omnibus:	126.367	**Durbin-Watson:**	2.011
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	237.444
Skew:	-1.079	**Prob(JB):**	2.75e-52
Kurtosis:	4.929	**Cond. No.**	163.

Notes:

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

```
The equation for this regression is y = 75.70040486231386 + 17.260591059550308
* priGPA + −1.7165529361367278 * ACT
```

iii) The coefficient for priGPA makes sense, and it is saying that for one more gpa point, the expected attendnace for said student is to increase by 17.26%. This makes sense because students that perform better academically tend to have better attendance. On the other hand, the coefficient for act is -1.71, meaning that for every extra point a student scores on the ACT, their expected attendance is said to drop by 1.7%. This is a surprise because similar to the first explanatory variable, the student performs better academically, but with the ACT they are expected to show up to class less.

iv) I will calculate the expected attendance for said student below. The expected attendance for this student is 104%. This means that they attend every class, and also extra classes(maybe review sessions). There are no students in this data set with those specific numbers.

In [231…

```
attend_predic1 = attend_int + priGPA_coeff*3.65 + act_coeff*20
student_exists = attend[(attend['priGPA'] == 3.65) & (attend['ACT'] == 20)]
if not student_exists.empty:
```

```
    print(f"There is/are {len(student_exists)} student(s) with priGPA=3.65 and
else:
    print("There are no students in the dataset with priGPA=3.65 and ACT=20.")
```

```
There are no students in the dataset with priGPA=3.65 and ACT=20.
```

v)I will perform all of the necessary calculations below

In [234…
```
attend_predic2 = attend_int + priGPA_coeff*3.1 + act_coeff*21
attend_predic3 = attend_int + priGPA_coeff*2.1 + act_coeff*26
difference = abs(attend_predic2 - attend_predic3)
print(f'The precicted difference in attendance between student A and B is: {di
```

```
The precicted difference in attendance between student A and B is: 25.84%
```

Chapter 3 C5

In [237…
```
wage1 = woo.dataWoo('wage1')
wage1.describe()
```

Out[237]:

	wage	educ	exper	tenure	nonwhite	female	married
count	526.000000	526.000000	526.00000	526.000000	526.000000	526.000000	526.000000
mean	5.896103	12.562738	17.01711	5.104563	0.102662	0.479087	0.608365
std	3.693086	2.769022	13.57216	7.224462	0.303805	0.500038	0.488580
min	0.530000	0.000000	1.00000	0.000000	0.000000	0.000000	0.000000
25%	3.330000	12.000000	5.00000	0.000000	0.000000	0.000000	0.000000
50%	4.650000	12.000000	13.50000	2.000000	0.000000	0.000000	1.000000
75%	6.880000	14.000000	26.00000	7.000000	0.000000	1.000000	1.000000
max	24.980000	18.000000	51.00000	44.000000	1.000000	1.000000	1.000000

8 rows × 24 columns

I will perform all necessary calculations below

In [240…
```
model_educ = smf.ols('educ ~ exper + tenure', data=wage1).fit()
wage1['r1'] = model_educ.resid

model_r1 = smf.ols('lwage ~ r1', data=wage1).fit()

model_full = smf.ols('lwage ~ educ + exper + tenure', data=wage1).fit()

r1_coeff= model_r1.params['r1']
educ_coeff = model_full.params['educ']
print(f'{r1_coeff:.5f}')
print(f'{educ_coeff:.5f}')
```

```
0.09203
0.09203
```

The coefficients for both regressions are exactly the same. This confirms the partialling out for this question.

In [ ]: