hw1

January 20, 2025

[13]:
```
import numpy as np
import copy
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels
import pandas as pd
import patsy
print("Statsmodels version:", statsmodels.__version__)
print("Patsy version:", patsy.__version__)
print("Pandas version:", pd.__version__)
pd.set_option("display.max_rows", 100)
pd.set_option("display.max_columns", 100)
np.random.seed(0)
```

```
Statsmodels version: 0.14.4
Patsy version: 1.0.1
Pandas version: 2.2.2
```

0.1 Assignment 1

[14]:
```
def biased_ate_estimation():
    parent_education = np.random.uniform(0, 4, 1000)
    child_ivy = np.random.binomial(n=1, p = 0.1 + parent_education / 10,␣
 ↪size=1000)
    child_future_income = np.random.normal(100000 + 50000 * parent_education,␣
 ↪10000)
    y_0 = copy.deepcopy(child_future_income)
    child_future_income = child_future_income + np.random.normal(50000, 10000)␣
 ↪* child_ivy
    y_1 = copy.deepcopy(child_future_income)

    mean_income_no_ivy = np.mean(child_future_income[child_ivy == 0])
    mean_income_ivy = np.mean(child_future_income[child_ivy == 1])

    two_means_estimator = mean_income_ivy - mean_income_no_ivy
```

```
    ci_lower = mean_income_ivy - mean_income_no_ivy - 1.96 * np.sqrt(np.
  ↪var(child_future_income[child_ivy == 0]) / np.sum(child_ivy == 0) + np.
  ↪var(child_future_income[child_ivy == 1]) / np.sum(child_ivy == 1))
    ci_upper = mean_income_ivy - mean_income_no_ivy + 1.96 * np.sqrt(np.
  ↪var(child_future_income[child_ivy == 0]) / np.sum(child_ivy == 0) + np.
  ↪var(child_future_income[child_ivy == 1]) / np.sum(child_ivy == 1))
    return two_means_estimator, ci_lower, ci_upper

biases = []
coverage = 0
ate_estimates = []
for _ in range(100):
    ate, ci_lower, ci_upper = biased_ate_estimation()
    ate_estimates.append(ate)
    if ci_lower < 50000 < ci_upper:
        coverage += 1
    biases.append(ate - 50000)
print("Average ATE estimate: ", np.mean(ate_estimates))
print("Average bias: ", np.mean(biases))
print("Coverage: ", coverage / 100)
print("Standard deviation of ATE estimates: ", np.std(ate_estimates))
```

```
Average ATE estimate:  82560.76963818411
Average bias:  32560.769638184105
Coverage:  0.01
Standard deviation of ATE estimates:  10292.931602241
```

0.1.1 Data generating Process:

1. **Parent Education** ((P)):
   - $P \sim \text{Uniform}(0, 4)$
2. **Treatment Variable** ((D)):
   - $D \sim \text{Bernoulli}(0.1 + \frac{P}{10})$
3. **Potential Outcomes**:
   - $Y(0) = 100000 + 50000 \cdot P + \epsilon, \quad \epsilon \sim \text{Normal}(0, 10000^2)$
   - $Y(1) = Y(0) + 50000 + \nu, \quad \nu \sim \text{Normal}(0, 10000^2)$

0.1.2 Explanation of Bias:

The treatment assignment D is correlated with P, which affects Y. As a result: - The treated group ($D = 1$) is not comparable to the untreated group ($D = 0$) since P influences both treatment assignment and potential outcomes. - The observed difference between treated and untreated outcomes confounds the true treatment effect with differences driven by P.

0.1.3 Real-World Example:

A real world example that might have a similar data generating process is a study on the impact of ivy league education on income. In this case, parental education could be a confounding variable that affects both the likelihood of attending an ivy league school and future income. If the study fails to account for parental education, it might overestimate the effect of ivy league education on income.

0.1.4 Results

- **Coverage:** 0.01
- **Bias:** 32560.77
- **Standard Deviation:** 10292.93

0.2 Assignment 2

[3]:
```python
# Calculati# TODO: do this for all age groups (i.e. mimic the CDC table as said␣
↪on ed)
NV = 58
NU = 61
RV = 0/NV
RU = 1/NU
VE = (RU - RV) / RU
print(f"Overall vaccine efficacy is {VE:.4f}")
```

```
Overall vaccine efficacy is 1.0000
```

Yes, in this case for the age group 16-17 years we recover the same vaccine efficacy (100%) as the CDC.

[4]:
```python
# Calculation for age groups 18-64:

NV = 14445
NU = 14566
RV = 8/NV
RU = 149/NU
VE = (RU - RV) / RU
print(f"Overall vaccine efficacy is {VE:.4f}")
```

```
Overall vaccine efficacy is 0.9459
```

In this case, the vaccine efficiency we get is the same as the CDC with rounding (us: 94.59%, them: 94.6%).

[27]:
```python
def get_delta_CI(age_group: str, NV: int, NU: int, RV: int, RU: int):
    var_RV = np.var([1]*RV + [0]*(NV-RV)) / (NV / NV + NU)
    var_RU = np.var([1]*RU + [0]*(NU-RU)) / (NU / NV + NU)
    RV = RV/NV
    RU = RU/NU
    VE = (RU - RV) / RU
    V_hat = np.array([[var_RU, 0], [0, var_RV]])

    G = np.array([(RV/(RU**2)), -(1/RU)])

    var_delta = G @ V_hat @ G.T
```

```
    std_err_delta = float(np.sqrt(var_delta))

    CI = VE - 1.96 * std_err_delta, VE + 1.96 * std_err_delta
    print(f"Delta Method CI for vaccine efficacy in age group {age_group} is␣
 ↪[{CI[0]:.5f}, {CI[1]:.5f}]")
    return CI


get_delta_CI("16-17", 58, 61, 0, 1)
get_delta_CI("18-64", 14443, 14566, 8, 149)
get_delta_CI("65-74", 3239, 3255, 1, 14)
get_delta_CI("75+", 805, 812, 0, 5)
get_delta_CI("overall", 18559, 18708, 9, 169)
```

```
Delta Method CI for vaccine efficacy in age group 16-17 is [1.00000, 1.00000]
Delta Method CI for vaccine efficacy in age group 18-64 is [0.90751, 0.98419]
Delta Method CI for vaccine efficacy in age group 65-74 is [0.78299, 1.07345]
Delta Method CI for vaccine efficacy in age group 75+ is [1.00000, 1.00000]
Delta Method CI for vaccine efficacy in age group overall is [0.91048, 0.98216]
```

```
[27]: (0.9104778289378604, 0.9821581941594361)
```

The vaccine efficiency is $\frac{RU-RV}{RU}$. Using our observed risks, we can construct a plug-in estimate $\hat{VE} = \frac{\hat{RU}-\hat{RV}}{\hat{RU}} = 1 - \frac{\hat{RV}}{\hat{RU}}$.

The gradient of this with respect to the observed risks, is $G = \left(\frac{\hat{RV}}{\hat{RU}^2}, -\frac{1}{\hat{RU}}\right)$.

For the variance, we can use the delta method, which gives us the approximation: $\sqrt{n}(\hat{VE} - VE) \sim^a \mathcal{N}(0, GVG^T)$, where V is the variance-covariance matrix of the observed risks.

We know that the distribution of the observed risks under some mild regularity conditions can be described by:$\sqrt{n}(\hat{R} - \mu) \sim^a \mathcal{N}(0, \hat{V})$, where μ is the vector of true population risks, and V is a consistent estimate of their covariance matrix: $\hat{V} = \begin{pmatrix} \frac{Var(Y|D=0)}{P(D=0)} & 0 \\ 0 & \frac{Var(Y|D=1)}{P(D=1)} \end{pmatrix}$ where we plug in observed sample estimates for the outcome variances and treatment means.

To obtain the final CI based on the delta method, we therefore compute $\hat{VE} \pm 1.96 * \sqrt{G\hat{V}G^T/n}$ where n is the total number of patients in the respective age group and $\sqrt{G\hat{V}G^T/n}$ is the standard error.

The confidence intervals differ from those provided by the CDC:

0.3 Age Group Their CI Our CI

Overall [89.6, 97.6] [91.0, 98.2]

16 to 17 [-3969.9, 100.0] [100.0, 100.0]

18 to 64 [89.1, 97.7] [90.8, 98.4]

65 to 74 [53.2, 99.8] [78.3, 107.3]

75+ [-12.1, 100.0] [100.0, 100.0]

0.4 Assignment 3

[28]:
```
data = pd.read_csv("https://raw.githubusercontent.com/VC2015/DMLonGitHub/master/
 ↪penn_jae.dat", delim_whitespace=True)
n, p = data.shape
data = data[data["tg"].isin([0, 4])]
data["T4"] = np.where(data["tg"] == 4, 1, 0)

# non-interactive regression model
cra = smf.ols("np.log(inuidur1) ~ T4 + "
              ␣
 ↪"(female+black+othrace+C(dep)+q2+q3+q4+q5+q6+agelt35+agegt54+durable+lusd+husd)**2",
              data)
cra_results = cra.fit(cov_type="HC1")
print(f"Treatment effect basic non interactive regression: {cra_results.
 ↪params["T4"]:.4f}")
print(f"Standard error of basic non interactive regression: {cra_results.
 ↪bse["T4"]:.4f}")

# interactive regression model
ira_formula = "0 +␣
 ↪(female+black+othrace+C(dep)+q2+q3+q4+q5+q6+agelt35+agegt54+durable+lusd+husd)**2"
X = patsy.dmatrix(ira_formula, data, return_type='dataframe')
X.columns = [f'x{t}' for t in range(X.shape[1])]  # clean column names
X = (X - X.mean(axis=0))  # demean all control covariates
# construct interactions of treatment and (de-meaned covariates, 1)
ira_formula = "T4 * (" + "+".join(X.columns) + ")"
X['T4'] = data['T4']
X = patsy.dmatrix(ira_formula, X, return_type='dataframe')
ira = sm.OLS(np.log(data[["inuidur1"]]), X)
ira_results = ira.fit(cov_type="HC1")
print(f"Treatment effect basic interactive regression: {ira_results.
 ↪params["T4"]:.4f}")
print(f"Standard error of basic interactive regression: {ira_results.bse["T4"]:.
 ↪4f}")


# None of the variables in the data are numerical, so non-linear␣
 ↪transformations like log, square or sqrt on single variables make no sense.
cra_extended = smf.ols("np.log(inuidur1) ~ T4 + "
              ␣
 ↪"(female+black+othrace+C(dep)+q2+q3+q4+q5+q6+agelt35+agegt54+durable+lusd+husd)**2␣
 ↪+ female:C(dep):lusd + female:C(dep):husd",
```

```
            data)
cra_results_hetero = cra.fit(cov_type="HC1")
print(f"Treatment effect extended non interactive regression:␣
↪{cra_results_hetero.params["T4"]:.4f}")
print(f"Standard error of extended non interactive regression:␣
↪{cra_results_hetero.bse["T4"]:.4f}")
cra_results_homo = cra.fit()
print(f"Treatment effect extended non interactive regression with non robust␣
↪errors: {cra_results_homo.params["T4"]:.4f}")
print(f"Standard error of extended non interactive regression with non robust␣
↪errors: {cra_results_homo.bse["T4"]:.4f}")
ira_formula_extended = "0 +␣
↪(female+black+othrace+C(dep)+q2+q3+q4+q5+q6+agelt35+agegt54+durable+lusd+husd)**2␣
↪+ female:C(dep):lusd + female:C(dep):husd"
X = patsy.dmatrix(ira_formula_extended, data, return_type='dataframe')
X.columns = [f'x{t}' for t in range(X.shape[1])]
X = (X - X.mean(axis=0))
ira_formula_extended = "T4 * (" + "+".join(X.columns)+")"
X["T4"] = data["T4"]
X = patsy.dmatrix(ira_formula_extended, X, return_type="dataframe")
ira_extended = sm.OLS(np.log(data[["inuidur1"]]), X)
ira_results_extended_hetero = ira_extended.fit(cov_type="HC1")
print(f"Treatment effect extended interactive regression:␣
↪{ira_results_extended_hetero.params["T4"]:.4f}")
print(f"Standard error of extended interactive regression:␣
↪{ira_results_extended_hetero.bse["T4"]:.4f}")
ira_results_extended_homo = ira_extended.fit()
print(f"Treatment effect extended interactive regression with non robust errors:
↪ {ira_results_extended_homo.params["T4"]:.4f}")
print(f"Standard error of extended interactive regression with non robust␣
↪errors: {ira_results_extended_homo.bse["T4"]:.4f}")
```

```
/var/folders/qz/2xz7hqzj4mv4_58tqxfy0z640000gn/T/ipykernel_48622/616873270.py:1:
FutureWarning: The 'delim_whitespace' keyword in pd.read_csv is deprecated and
will be removed in a future version. Use ``sep='\s+'`` instead
  data = pd.read_csv("https://raw.githubusercontent.com/VC2015/DMLonGitHub/maste
r/penn_jae.dat", delim_whitespace=True)
```

```
Treatment effect basic non interactive regression: -0.0797
Standard error of basic non interactive regression: 0.0356
Treatment effect basic interactive regression: -2993880.4805
Standard error of basic interactive regression: 3248323.3557
Treatment effect extended non interactive regression: -0.0797
Standard error of extended non interactive regression: 0.0356
Treatment effect extended non interactive regression with non robust errors:
-0.0797
Standard error of extended non interactive regression with non robust errors:
0.0356
```

```
Treatment effect extended interactive regression: 4979232.9769
Standard error of extended interactive regression: 5401064.7459
Treatment effect extended interactive regression with non robust errors:
4979232.9769
Standard error of extended interactive regression with non robust errors:
4740774.1031
```

NOTE: despite my best efforts, I was unable to locally reproduce the results found in the colab notebook, despite ensuring that my data, numpy seed and library versions were the same. I suspect that this issue relates to python version or OS differences outside of my control. Copying the exact same code from above into colab yielded:

Treatment effect basic non interactive regression: -0.0797
Standard error of basic non interactive regression: 0.0356
Treatment effect basic interactive regression: -0.0752
Standard error of basic interactive regression: 0.0356
Treatment effect extended non interactive regression: -0.0798
Standard error of extended non interactive regression: 0.0356
Treatment effect extended interactive regression: -0.0752
Standard error of extended interactive regression: 0.0356
Treatment effect extended non interactive regression with non robust errors: -0.0798
Standard error of extended non interactive regression with non robust errors: 0.0357
Treatment effect extended interactive regression with non robust errors: -0.0752
Standard error of extended interactive regression with non robust errors: 0.0361

I will work under the assumption that these are the expected results.

The nonrobust standard errors of the average treatment effects vary slightly from the robust standard errors. This is indicates that there may be some heteroskedasticity in the data. Thus we can hypothesize that the small changes in the standard errors are due to the robust standard errors accounting for this heteroskedasticity.

0.5 Assignment 4

[29]:
```
# Basic version in colab:

np.random.seed(123)
n = 1000                # sample size
Z = np.random.normal(size=n)         # generate Z
Y0 = -Z + np.random.normal(size=n)   # conditional average baseline response is␣
 ↪-Z
Y1 = Z + np.random.normal(size=n)    # conditional average treatment effect is␣
 ↪+Z
D = np.random.binomial(1, .2, size=n)    # treatment indicator; only 20% get␣
 ↪treated
Y = Y1 * D + Y0 * (1 - D)  # observed Y
Z = Z - Z.mean()       # demean Z
data = pd.DataFrame({"Y": Y, "D": D, "Z": Z})
```

```python
CL = smf.ols("Y ~ D", data=data).fit()
CRA = smf.ols("Y ~ D + Z", data=data).fit()      # classical
IRA = smf.ols("Y ~ D + Z + Z*D", data=data).fit()  # interactive approach
# we are interested in the coefficients on variable "D".
print(CL.get_robustcov_results(cov_type="HC1").summary())
print(CRA.get_robustcov_results(cov_type="HC1").summary())
print(IRA.get_robustcov_results(cov_type="HC1").summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     1.568
Date:                Mon, 20 Jan 2025   Prob (F-statistic):              0.211
Time:                        17:43:38   Log-Likelihood:                -1753.3
No. Observations:                1000   AIC:                             3511.
Df Residuals:                     998   BIC:                             3520.
Df Model:                           1
Covariance Type:                  HC1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0145      0.050      0.290      0.772      -0.084       0.113
D             -0.1330      0.106     -1.252      0.211      -0.341       0.075
==============================================================================
Omnibus:                        2.550   Durbin-Watson:                   2.153
Prob(Omnibus):                  0.279   Jarque-Bera (JB):                2.419
Skew:                          -0.094   Prob(JB):                        0.298
Kurtosis:                       3.150   Cond. No.                         2.67
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.253
Model:                            OLS   Adj. R-squared:                  0.252
Method:                 Least Squares   F-statistic:                     100.0
Date:                Mon, 20 Jan 2025   Prob (F-statistic):           2.64e-40
Time:                        17:43:38   Log-Likelihood:                -1608.1
No. Observations:                1000   AIC:                             3222.
Df Residuals:                     997   BIC:                             3237.
Df Model:                           2
Covariance Type:                  HC1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0298      0.036      0.837      0.403      -0.040       0.100
```

```
D              -0.2146      0.138     -1.556      0.120      -0.485       0.056
Z              -0.7015      0.050    -14.123      0.000      -0.799      -0.604
==============================================================================
Omnibus:                       19.609   Durbin-Watson:                   2.128
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               37.651
Skew:                           0.006   Prob(JB):                     6.67e-09
Kurtosis:                       3.951   Cond. No.                         2.67
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.532
Model:                            OLS   Adj. R-squared:                  0.531
Method:                 Least Squares   F-statistic:                     388.7
Date:                Mon, 20 Jan 2025   Prob (F-statistic):          4.25e-167
Time:                        17:43:38   Log-Likelihood:                -1373.9
No. Observations:                1000   AIC:                             2756.
Df Residuals:                     996   BIC:                             2775.
Df Model:                           3
Covariance Type:                  HC1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0377      0.033      1.128      0.259      -0.028       0.103
D             -0.0783      0.080     -0.983      0.326      -0.234       0.078
Z             -1.0594      0.033    -31.727      0.000      -1.125      -0.994
Z:D            1.8839      0.075     25.081      0.000       1.737       2.031
==============================================================================
Omnibus:                        1.179   Durbin-Watson:                   2.047
Prob(Omnibus):                  0.555   Jarque-Bera (JB):                1.052
Skew:                          -0.066   Prob(JB):                        0.591
Kurtosis:                       3.087   Cond. No.                         2.76
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
```

a.) For the regression without covariates, the standard error of the treatment effect is 0.106.
For the non-interactive regression with covariates, the standard error of the treatment effect is 0.138.
For the interactive regression with covariates, the standard error of the treatment effect is 0.080.
This shows that adjusting for covariates without interaction terms did not improve precision, while adjusting for covariates with interaction terms did.

[30]:
```
non_robust_error_basic = CL.bse["D"]
non_robust_error_covariates = CRA.bse["D"]
```

```
non_robust_error_interactive = IRA.bse["D"]

print(f"Non-robust error for basic model: {non_robust_error_basic:.4f}")
print(f"Non-robust error for covariates model: {non_robust_error_covariates:.
 ↪4f}")
print(f"Non-robust error for interactive model: {non_robust_error_interactive:.
 ↪4f}")
```

```
Non-robust error for basic model: 0.1132
Non-robust error for covariates model: 0.0980
Non-robust error for interactive model: 0.0778
```

b.)

The true underlying model here is $Y = -Z * (1 - D) + D * Z + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

This can be rewritten as $Y = -Z + 2ZD + \epsilon$.

Thus, the true specification of the model is not linear in Z and D, so the regression specification in the model with covariates but without interaction terms is not equal to the conditional expectation function. The presence of an interaction term between D and the covariate, also shows that there is heteroskedasticity in the model. This is the source of the difference between the heteroskedastic errors and the homoskedastic errors we see.

In both the basic model and the model with covariates (but without interactions) the underlying assumption of the homoskedastic standard errors is that the model is correctly specified as $Y = \beta_0 + \beta_1 D + \epsilon$ and $Y = \beta_0 + \beta_1 D + \beta_2 Z + \epsilon$ with *epsilon* independent of the treatment and covariates respectively. This is not the case here, as the true model is $Y = -Z + 2ZD + \epsilon$. The presence of the interaction terms in the data generating process means that the model is not correctly specified, and the basic standard errors are not valid.

[31]:
```
n = 1000                  # sample size
Z = np.random.normal(size=n)          # generate Z
Y0 = -Z + np.random.normal(size=n)    # conditional average baseline response is␣
 ↪-Z
Y1 = Z + np.random.normal(size=n)     # conditional average treatment effect is␣
 ↪+Z
D = np.array([1] * 500 + [0] * 500)

# Shuffle the array randomly
np.random.shuffle(D)# treatment indicator; exactly 50% get randomly treated
Y = Y1 * D + Y0 * (1 - D)  # observed Y
Z = Z - Z.mean()        # demean Z
data = pd.DataFrame({"Y": Y, "D": D, "Z": Z})

CL = smf.ols("Y ~ D", data=data).fit()
CRA = smf.ols("Y ~ D + Z", data=data).fit()      # classical
IRA = smf.ols("Y ~ D + Z + Z*D", data=data).fit()  # interactive approach
# we are interested in the coefficients on variable "D".
print(CL.get_robustcov_results(cov_type="HC1").summary())
```

```python
print(CRA.get_robustcov_results(cov_type="HC1").summary())
print(IRA.get_robustcov_results(cov_type="HC1").summary())

non_robust_error_basic = CL.bse["D"]
non_robust_error_covariates = CRA.bse["D"]
non_robust_error_interactive = IRA.bse["D"]

print(f"Non-robust error for basic model: {non_robust_error_basic:.4f}")
print(f"Non-robust error for covariates model: {non_robust_error_covariates:.
↪4f}")
print(f"Non-robust error for interactive model: {non_robust_error_interactive:.
↪4f}")
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.003
Model:                            OLS   Adj. R-squared:                  0.002
Method:                 Least Squares   F-statistic:                     2.803
Date:                Mon, 20 Jan 2025   Prob (F-statistic):             0.0944
Time:                        17:44:07   Log-Likelihood:                -1742.5
No. Observations:                1000   AIC:                             3489.
Df Residuals:                     998   BIC:                             3499.
Df Model:                           1
Covariance Type:                  HC1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0623      0.059     -1.058      0.291      -0.178       0.053
D              0.1465      0.087      1.674      0.094      -0.025       0.318
==============================================================================
Omnibus:                        2.588   Durbin-Watson:                   2.060
Prob(Omnibus):                  0.274   Jarque-Bera (JB):                2.517
Skew:                           0.079   Prob(JB):                        0.284
Kurtosis:                       2.812   Cond. No.                         2.62
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.009
Method:                 Least Squares   F-statistic:                     3.365
Date:                Mon, 20 Jan 2025   Prob (F-statistic):             0.0350
Time:                        17:44:07   Log-Likelihood:                -1738.6
No. Observations:                1000   AIC:                             3483.
Df Residuals:                     997   BIC:                             3498.
Df Model:                           2
```

```
Covariance Type:                  HC1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0619      0.062     -0.991      0.322      -0.185       0.061
D              0.1458      0.087      1.672      0.095      -0.025       0.317
Z              0.1212      0.058      2.104      0.036       0.008       0.234
==============================================================================
Omnibus:                        3.206   Durbin-Watson:                   2.054
Prob(Omnibus):                  0.201   Jarque-Bera (JB):                2.920
Skew:                           0.067   Prob(JB):                        0.232
Kurtosis:                       2.772   Cond. No.                         2.62
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.482
Model:                            OLS   Adj. R-squared:                  0.481
Method:                 Least Squares   F-statistic:                     327.1
Date:                Mon, 20 Jan 2025   Prob (F-statistic):          8.74e-148
Time:                        17:44:07   Log-Likelihood:                -1414.8
No. Observations:                1000   AIC:                             2838.
Df Residuals:                     996   BIC:                             2857.
Df Model:                           3
Covariance Type:                  HC1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0649      0.044     -1.459      0.145      -0.152       0.022
D              0.1461      0.063      2.316      0.021       0.022       0.270
Z             -0.8884      0.043    -20.571      0.000      -0.973      -0.804
Z:D            1.8962      0.061     31.138      0.000       1.777       2.016
==============================================================================
Omnibus:                        3.060   Durbin-Watson:                   2.049
Prob(Omnibus):                  0.217   Jarque-Bera (JB):                2.916
Skew:                          -0.084   Prob(JB):                        0.233
Kurtosis:                       2.796   Cond. No.                         2.70
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
Non-robust error for basic model: 0.0875
Non-robust error for covariates model: 0.0872
Non-robust error for interactive model: 0.0631
```

c.)

After changing the DGP so that the trial is balanced, we have the following results:

Robust error for basic model: 0.087
Robust error for covariates model: 0.087
Robust error for interactive model: 0.063

Non-robust error for basic model: 0.0875
Non-robust error for covariates model: 0.0872
Non-robust error for interactive model: 0.0631

The robust errors are the basically the same for the basic and covariates models, but the interactive model has a lower standard error. So adding covariates without interactions did not improve precision, but adding covariates with interactions did. For all models, the non-robust errors are the same as the robust errors.

d.) Proof: When the treatment is balanced, adjusting linearly for covariates cannot increase the standard error of the ATE estimate in the linear model.

When the trial is balanced and random, we have $E[D] = 0.5$. The heteroskedasticity robust variance formula for the estimate of the ATE is $V_\alpha = \frac{E[\epsilon^2 D^2]}{E[D^2]^2}$, where D is defined as $D = D - E[D] = \pm\frac{1}{2}$, so $D^2 = \frac{1}{4}$ is constant. Note that the definition of ϵ varies by model used to estimate the treatment effect.

For the model without covariates, we have the specification $Y = \beta_0 + \beta_1 D + \epsilon_0$. Note that we can write ϵ_0 as $\epsilon_0 = Y - \beta_0 - \beta_1 D$. Next, consider the linear model with added de-meaned covariates W. The model is then specified as $Y = \beta_0 + \beta_1 D + \beta_2 W + \epsilon_1$. We can thus rewrite ϵ_0 as $\epsilon_0 = \beta_2 W + \epsilon_1$.

Now let's consider the respective variances of the model with and without covariates respectively, given by $V_\alpha = \frac{E[\epsilon^2 D^2]}{E[D^2]^2}$ for $\epsilon = \epsilon_0$ and $\epsilon = \epsilon_1$ respectively. We can rewrite these as $V_0 = \frac{E[\epsilon_0^2 D^2]}{E[D^2]^2}$ and $V_1 = \frac{E[\epsilon_1^2 D^2]}{E[D^2]^2}$. For both, the denominator is the same, so we can compare the numerators.

We can rewrite the numerator of V_0 as $E[\epsilon_0^2 D^2] = E[(\beta_2 W + \epsilon_1)^2 D^2] = E[\beta_2^2 W^2 D^2] + E[\epsilon_1^2 D^2] + 2E[\beta_2 W \epsilon_1 D^2]$. The numerator of V_1 is $E[\epsilon_1^2 D^2]$. We have that $2E[\beta_2 W \epsilon_1 D^2] = 2E[\beta_2 W \epsilon_1 \frac{1}{4}] = \frac{1}{2}\beta_2 E[W\epsilon_1] = 0$ since $E[W\epsilon_1] = 0$. Because of this, we have that the numerator of V_0 is greater than the numerator of V_1, so $V_0 > V_1$. This means that the standard error of the ATE estimate in the linear model with covariates is lower or equal to the standard error of the ATE estimate in the linear model without covariates when the treatment is balanced.