1. (a) N_1 units are drawn at random out of the population of N to receive the treatment. The unit assignment probability of any observation i is therefore N_1/N.

   (b) $V(\hat{\beta} \mid X) = (X'X)^{-1} \left(\sum_{i=1}^{N} \Omega_{ii} X_i X_i' \right) (X'X)^{-1}$

   The OLS estimator for the treatment effect is

$$\hat{\beta}_D = \frac{1}{N_1} \sum_{i:D_i=1} Y_i - \frac{1}{N - N_1} \sum_{i:D_i=0} Y_i = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{D_i \cdot Y_i(1)}{N_1/N} - \frac{(1 - D_i) \cdot Y_i(0)}{(N - N_1)/N} \right)$$

   The standard robust variance for least squares estimators is

$$V_{\text{hetero}} = \frac{\sum_{i=1}^{N} \sigma_i^2(D_i) \cdot (D_i - \bar{D})^2}{\left(\sum_{i=1}^{N} (D_i - \bar{D})^2 \right)^2} = \frac{\sum_{i=1}^{N} \sigma_i^2(D_i) \cdot (D_i - \bar{D})^2}{\left(\sum_{i=1}^{N} V_D^2(D_i) \right)^2}$$

   Using $\sum \sigma_i^2(D_i) = \sum \sigma_i^2(1) D_i + \sum \sigma_i^2(0)(1 - D_i)$ we can write

$$V(\hat{\beta}_D | D) = \frac{\sum_{i=1}^{N} \sigma_i^2(1) D_i}{N_1^2} + \frac{\sum_{i=1}^{N} \sigma_i^2(0)(1 - D_i)}{(N - N_1)^2}$$

   Given the setup of a completely randomized experiment (N units, with N_1 randomly assigned to the treatment), $Pr_D(D_i = 1 | Y(0), Y(1)) = E_D[D_i | Y(0), Y(1)] = N_1/N$ (probability, expectation, or variance, is taken solely over the randomization distribution, keeping fixed the potential outcomes Y(0) and Y(1), and keeping fixed the population). Since $D_i \in \{0, 1\}$, $D_i^2 = D_i$ we have $E_D[D_i^2 | Y(0), Y(1)] = E_D[D_i | Y(0), Y(1)]$ and $V_D(D_i) = N_1/N \cdot (1 - N_1/N)$.

   (c) It follows that if $\sigma_i^2(D_i) = \sigma^2(D_i)$ we can write

$$\mathbb{E}_{D \in \mathcal{D}}[V(\hat{\beta}_D | D)] = V(\hat{\beta} | N_0, N_1) = \frac{\sigma_1^2}{N_1} + \frac{\sigma_0^2}{N - N_1}$$

2. (a) The population ATE is $E[Y_i(1, X_i) - Y_i(0, X_i)] = E[(\tau_i + 5 \times X_i + \epsilon_i) - (5 \times X_i + \epsilon_i)] = E[\tau_i] = 1$.

   (b)
```
set seed 42 // set obs 100
gen e = rnormal(0, 1) // gen t = rnormal(1, 1) // gen x = (_n > 50)
gen y0 = 5 * x + e // gen y1 = y0 + t
egen sate = mean(t) // scalar sate = sate // dis sate
1.0598215
```

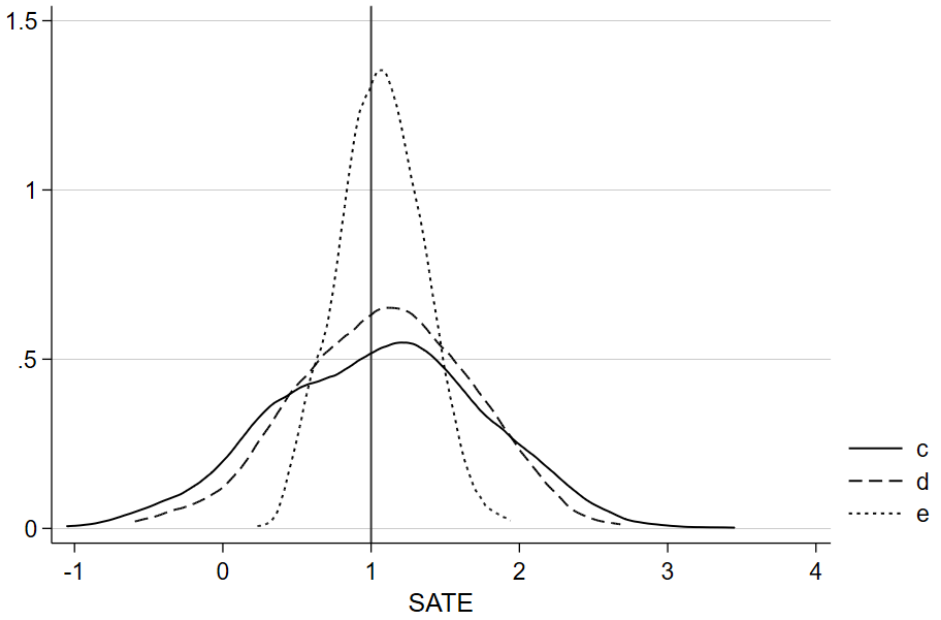
   (c)
```
cap program drop montecarloc // program define montecarloc, rclass
gen random = runiform() // sort random // gen d = (_n/_N) <= .25
gen y = y1 * d + (1 - d) * y0 // reg y d
return scalar b = _b[d] // return scalar se = _se[d]
test _b[d] == '=sate' // return scalar p = r(p)
drop d y random // end // preserve
simulate bc = r(b) sec = r(se) pc = r(p), reps(1000)
saving(2c.dta, replace): montecarloc // restore
```

[1] mach5689@student.su.se

(d)
```
cap program drop montecarlod // program define montecarlod, rclass
gen random = runiform() // sort random // gen d = (_n/_N) <= .5
gen y = y1 * d + (1 - d) * y0  // reg y d
return scalar b = _b[d] //return scalar se = _se[d]
test _b[d] == `=sate' // return scalar p = r(p)
drop d y random // end  // preserve
simulate bd = r(b) sed = r(se) pd = r(p), reps(1000)
saving(2d.dta, replace): montecarlod // restore
```

(e)
```
cap program drop montecarloe // program define montecarloe, rclass
gen random = runiform() // sort x random
by x: gen d = (_n/_N) <= .25
gen y = y1 * d + (1 - d) * y0  // reg y d
return scalar b = _b[d] // return scalar se = _se[d]
test _b[d] == `=sate' // return scalar p = r(p)
drop d y random // end  // preserve
simulate be = r(b) see = r(se) pe = r(p), reps(1000)
saving(2e.dta, replace): montecarloe // restore
```

(f)
```
use 2c.dta, clear // merge 1:1 _n using 2d.dta, nogen
merge 1:1 _n using 2e.dta, nogen // twoway (kdensity bc) (kdensity bd)
(kdensity be), xline(1) legend(order(1 "c" 2 "d" 3 "e" ))
```



(g)

	(c)	(d)	(e)
Sd coef	0.706	0.585	0.284
Mean se	0.664	0.582	0.676
Mean p < .05	0.062	0.050	0.000

Since the standard deviation of the estimates is lower when assigning 50% observations in assignment (d) rather 25% in assignment (c) we can infer that $\sigma_i(1, X_i) > \sigma_i(0, X_i)$. The average standard errors in (c) and (d) are roughly equivalent to the standard deviation of the estimates. The estimates from the stratified assignment in (e) show significantly smaller variability by excluding potential assignments where the distribution of X_i. The average standard error does not take this stratification into account: it is comparable to the one in (c) and thus too large. Consequently, none of the estimates in (e) have a p-value smaller than the expected 0.05.

3. (a)
```
clear all // set obs 1000 // gen e = rnormal(0, 1)
gen t = rnormal(1, 1) // egen sat = mean(t)
scalar sat = sat // dis sat
1.0232249 *sample average of tau_i
```

Sample average of $\tau_i = 1.0232249$

```
cap program drop montecarlo3// program define montecarlo3, rclass
cap drop random d x* y*// gen random = runiform()
sort random // gen d = (_n/_N) <= .5
gen x0  = e > -1 // gen x1      = e > 1
gen x   = d * x1 + (1 - d) * x0 // gen y0       = 5 * x0 + e
gen y1  = t + 5 * x1 + e// gen y        = d * y1 + (1 - d) * y0
qui reg y d // return scalar tau=_b[d]
qui reg y d if x == 1 // return scalar tau1=_b[d]
qui reg y d if x == 0// return scalar tau0=_b[d]
end // montecarlo3
gen te = y1 - y0 // egen sate = mean(te)
scalar sate = sate// dis sate
-2.376775 *SATE
```

Sample ATE $= -2.376775$
Population ATE: $E[Y_i(1, X_i(1)) - Y_i(0, X_i(0))]$
$= E[\tau_i] + 5 \times (E[1(\epsilon_i > 1)] - E[1(\epsilon_i > -1)]) = 1 + 5 \times (\Phi(-1) - \Phi(1)) \approx -2.413$

```
dis (1 + 5*(normal(-1)-normal(1)))
-2.4134475 *ATE
```

(b)
```
simulate tau = r(tau), reps(1000): montecarlo3
-------------+----------
Mean      tau | -2.375876
```

(c)
```
simulate tau1 = r(tau1) tau0 = r(tau0), reps(1000): montecarlo3
-------------+----------
Mean     tau1 |  2.173395
Mean     tau0 |  2.274204
```

(d) The average estimate in (b) is unbiased for the theoretical average treatment effect. The sample CATEs are significantly different from the SATE. Conditioning leads to an imbalance because X_i is no longer independent of D_i and ϵ_i. The treated observations have on average larger ϵ_i than those in the control group by the definition of how X_i is generated:

```
dis (1+normalden(1)/(1-normal(1)))-(normalden(-1)/(1-normal(-1)))
2.2375353 *CATE for X=1
dis (1-normalden(1)/normal(1))+(normalden(-1)/normal(-1))
2.2375353 *CATE for X=0
```

Therefore X is a bad control: conditioning gives rise to a positive correlation between treatment and ϵ in turn artificially increasing the ATE estimate (collider bias).

4. Bayes' rule $f_{X|D=d}(x) = \frac{f_{X,D}(X=x,D=d)}{\Pr(D=d)} = \frac{\Pr(D=d|X=x)f_X(x)}{\Pr(D=d)}, \quad d \in \{0,1\}$. For $x \in \text{Supp}(X)$ balanced covariate distribution $f_{X|D=1}(x) = f_{X|D=0}(x)$ is equivalent to constant propensity score e since

$$\begin{aligned}
\frac{\Pr(D=1|X=x)f_X(x)}{\Pr(D=1)} &= \frac{\Pr(D=0|X=x)f_X(x)}{\Pr(D=0)} \\
\Leftrightarrow \frac{e(x)f_X(x)}{\Pr(D=1)} &= \frac{(1-e(x))f_X(x)}{\Pr(D=0)} \\
\Leftrightarrow \frac{e(x)}{\Pr(D=1)} &= \frac{1-e(x)}{1-\Pr(D=1)} \\
\Leftrightarrow e(x) &= \Pr(D=1) \equiv e.
\end{aligned}$$

5. (a)
```
set obs 10000 // gen e = rnormal(0,1)
gen t = rnormal(1,1) // gen x = (_n > 6000)// gen d = 0
replace d = 1 if x == 1 & runiform() < 0.8
replace d = 1 if x == 0 & runiform() < 0.5
gen y = t * d + 5 * x + e
```

(b)
```
reg y d if x == 0// Coef        d |   1.002636
reg y d if x == 1// Coef        d |    .944475
```

(c)
```
reg y d x // Coef               d |   .9849499
```

(d)
```
reg y d// Coef                  d |   2.472288
```

(e)

	x == 0	x == 1	(c)	(d)
d	1.003***	0.944***	0.985***	2.472***
	(0.0317)	(0.0527)	(0.0275)	(0.0550)
x			5.009***	
			(0.0273)	
_cons	−0.00898	5.041***	−0.000107	1.084***
	(0.0224)	(0.0469)	(0.0215)	(0.0432)
N	6000	4000	10000	10000

X_i is now a confounder which should be controlled for to achieve $\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp D_i \mid X_i$. Given each value of X_i, the assignment is completely randomised and the estimate in (b) and (c) are unconfounded. The uncdoditional estimate in (d), however, suffers from omitted variable bias (short equals long plus the effect of omitted times the regression of omitted on included):

$$OVB = 5 \cdot \frac{Cov(X_i, D_i)}{Var(D_i)} = 5 \cdot \frac{(.8-.3)*.4*.6}{(.8*.4+.5*.6)(1-(.8*.4+.5*.6))} \approx 1.5$$