8장. 이중차분법

인과추론과 패널데이터 - 이중차분법

김정현

발표자 소개

김정현 (Junghyun Kim)

- 스타트업 데이터팀 인턴 (2023.06 ~2024.01)
- 통계학과 석사 (2025.02 ~ 현재)

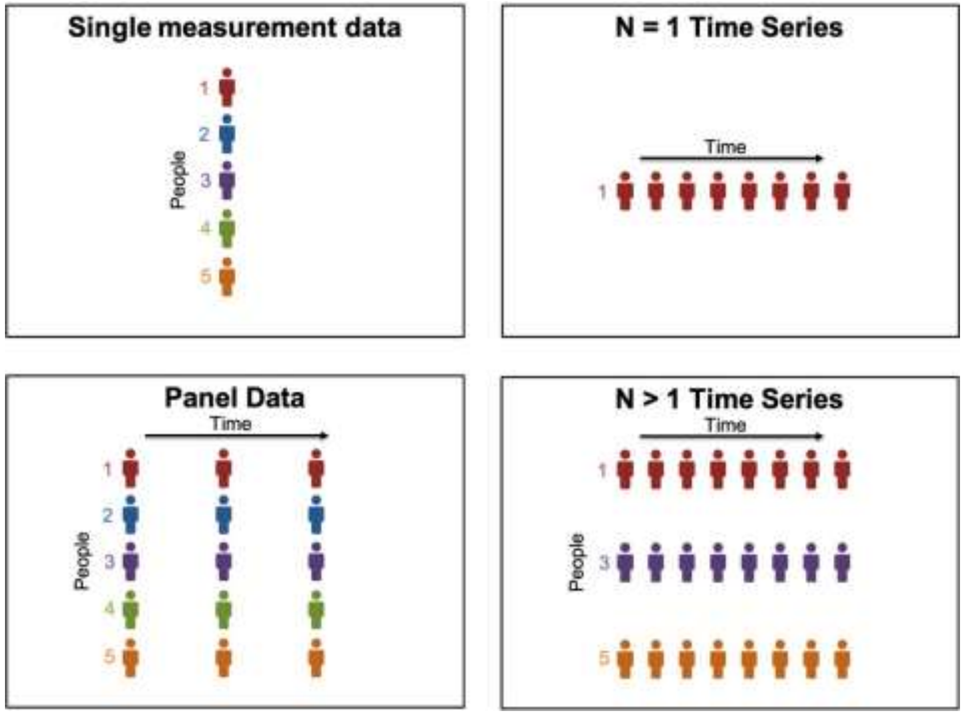
- Focuses on Statistical Machine Learning, Causal Inference

목차

- 패널데이터 알아보기
- 표준 이중차분법
- 식별 가정
- 시간에 따른 효과 변동
- 이중차분법과 공변량
- 이중 강건 이중차분법
- 처치의 시차도입

8.1. 패널데이터

- 시간에 따라 반복해서 관측되는 데이터 구조
  - 이때 반복되는 단위는 도시, 사람, 기업 등이 될 수 있음.



[오늘의 데이터]

- 오프라인 마케팅 데이터로 다운로드 수(Y)의 차이를 보고자 함.
  - Date (= t) -> 날짜 (t로 표현)
  - Downloads -> 다운로드 수 (결과변수)
  - Treated (= D)-> 실험군으로 지정된 경우, 오프라인 마케팅 수행할 예정인 곳.
  - Post -> 개입 시점 이후면 1, 이전이면 0 (= 개입여부)

	date	city	region	treated	tau	downloads	post
0	2021-05-01	5	S	0	0.0	51.0	0
1	2021-05-02	5	S	0	0.0	51.0	0
2	2021-05-03	5	S	0	0.0	51.0	0
3	2021-05-04	5	S	0	0.0	50.0	0
4	2021-05-05	5	S	0	0.0	49.0	0

8.1. 패널데이터

- 이 데이터는 왜 패널 데이터인가?
  - “5번 도시”라는 하나의 개체가,
  - 5월 1일부터 여러 날짜에 걸쳐,
  - 동일한 항목 (downloads)을 반복해서 측정하는 구조

	date	city	region	treated	tau	downloads	post
0	2021-05-01	5	S	0	0.0	51.0	0
1	2021-05-02	5	S	0	0.0	51.0	0
2	2021-05-03	5	S	0	0.0	51.0	0
3	2021-05-04	5	S	0	0.0	50.0	0
4	2021-05-05	5	S	0	0.0	49.0	0

8.1. 패널데이터

- 캠페인에 미친 영향을 표현하기 위해서는?
  - 캠페인이 시작한 이후 (t > T_pre)
  - 캠페인이 D = 1 도시들에 미친 영향을 이해하는 것.

$$ATT = E[Y_{it}(1) - Y_{it}(0) | D = 1, t > T_{pre}]$$

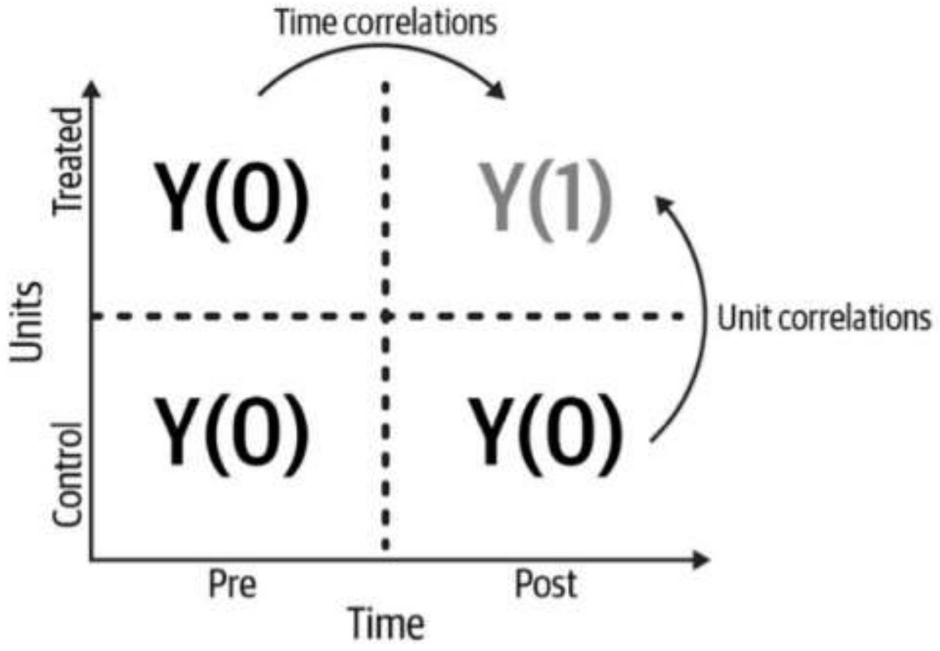
$$E[Y(0)|D = 1, Post = 1]$$

8.1. 패널데이터

- 캠페인에 미친 영향을 표현하기 위해서는?

$$E[Y(0)|D = 1, Post = 1]$$

- 3개의 셀을 활용하여 Y(0)를 관측
- 데이터 구조에 의해
  Y(1) 보다 **Y(0)**를 추정하기가 더 쉬움!



8.2. 표준 이중차분법

- 관측된 실험군 기준 값에 대조군 결과 추세를 보정

Our Goal

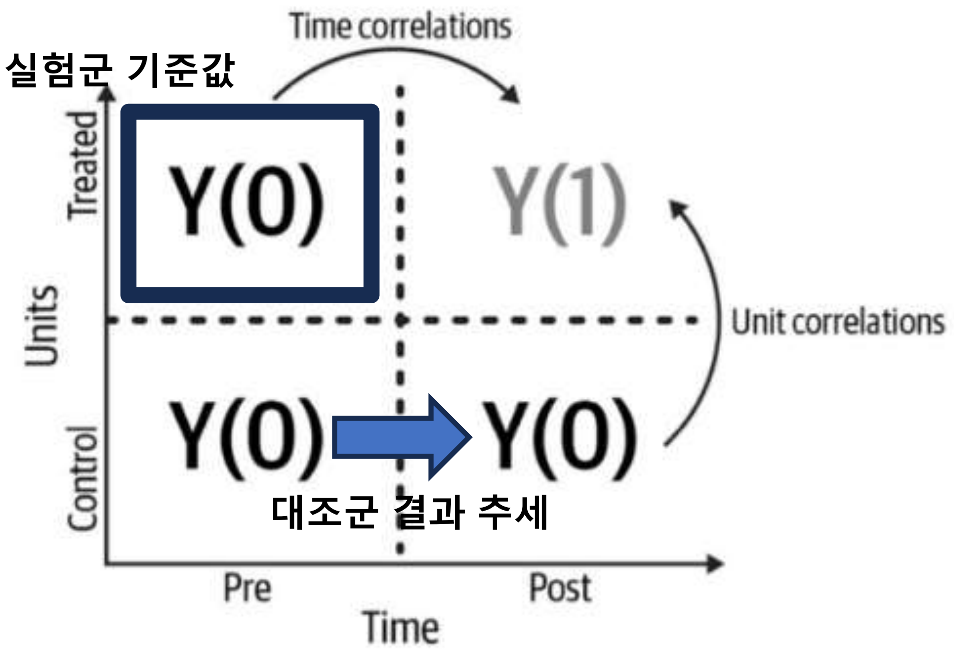
$$E[Y(0)|D=1, Post=1]$$

실험군 기준값

$$E[Y|D=1, Post=0]$$

대조군 결과추세

$$(E[Y|D=0, Post=1] - E[Y|D=0, Post=0])$$



8.2. 표준 이중차분법

- DID 추정량

$$ATT = E[Y_{it}(1) - \underline{Y_{it}(0)}|D = 1, t > T_{pre}]$$

$$E[Y|D=1, Post=0] + (E[Y|D=0, Post=1] - E[Y|D=0, Post=0])$$

$$\text{ATT} = (E[Y|D=1,\ \text{Post}=1] - E[Y|D=1,\ \text{Post}=0])$$
$$- (E[Y|D=0,\ \text{Post}=1] - E[Y|D=0,\ \text{Post}=0])$$

8.2. 표준 이중차분법

- DID 추정량 in code

$$ATT = E[Y_{it}(1) - Y_{it}(0) | D = 1, t > T_{pre}]$$

$$E[Y|D=1, Post=0] + (E[Y|D=0, Post=1] - E[Y|D=0, Post=0])$$

		downloads	date
treated	post		
0	0	50.335034	2021-05-01
	1	50.556878	2021-05-15
1	0	50.944444	2021-05-01
	1	51.858025	2021-05-15

```
y0_est = (did_data.loc[1].loc[0, "downloads"] # treated baseline
# control evolution
+ did_data.loc[0].diff().loc[1, "downloads"])

att = did_data.loc[1].loc[1, "downloads"] - y0_est
att
```

8.2. 표준 이중차분법

- 이중차분법을 접근하는 다양한 방법
  - 결과 변화의 측면 (델타)
  - OLS의 측면 (포화모델)
  - 고정효과의 측면 (이원고정효과 – TWFE)
  - 블록 디자인의 측면 (블록 단위 집계의 측면)

8.2. 표준 이중차분법

- 결과 변화의 측면

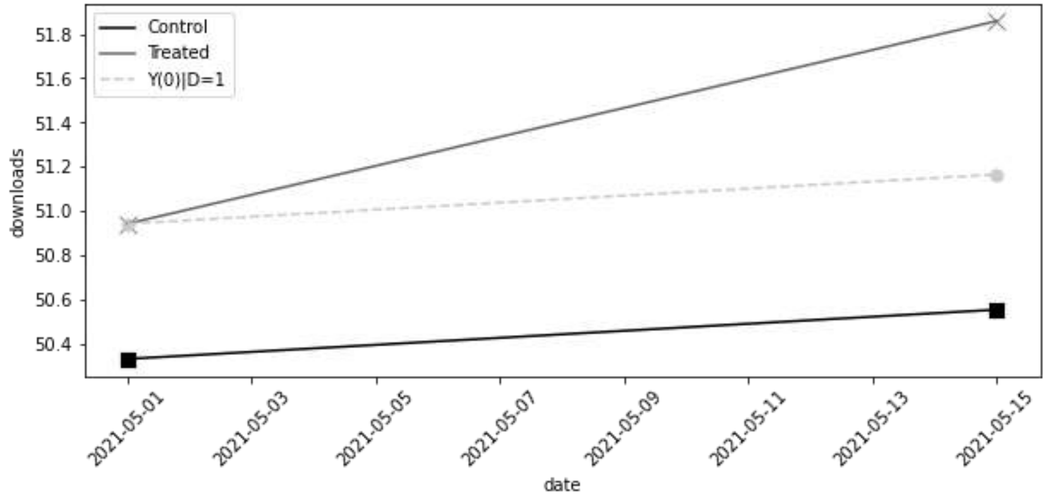
$$\text{ATT} = (E[Y|D=1,\ \text{Post}=1] - E[Y|D=1,\ \text{Post}=0])$$

실험군의 시간에 따른 변화

$$- (E[Y|D=0,\ \text{Post}=1] - E[Y|D=0,\ \text{Post}=0])$$

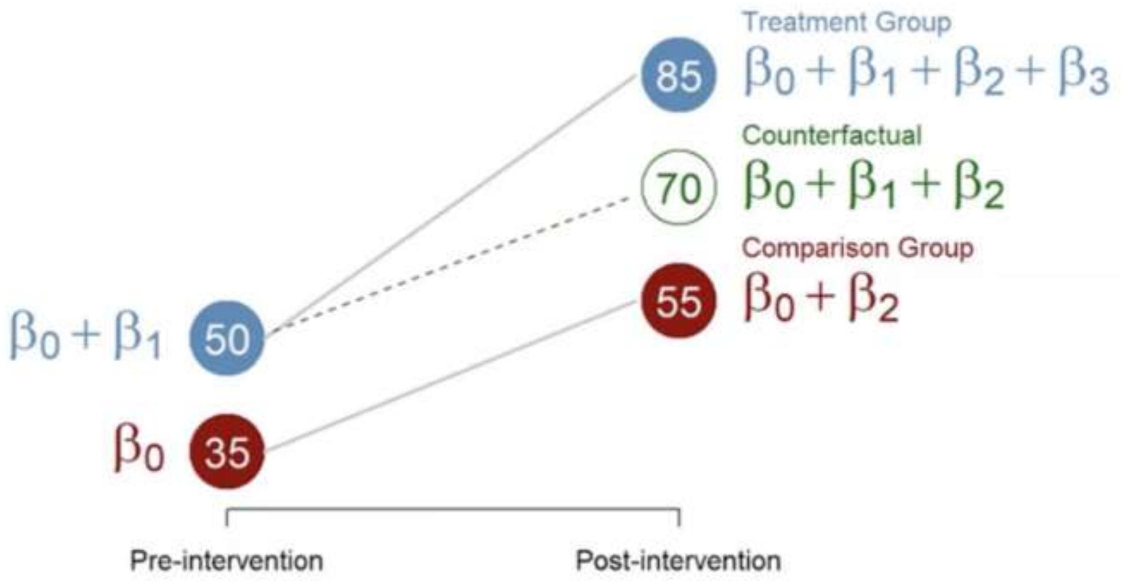
대조군의 시간에 따른 변화

$$ATT = E[\Delta y|D=1] - E[\Delta y|D=0]$$



8.2. 표준 이중차분법

- 선형회귀의 측면



$$Y_{it} = \beta_0 + \beta_1 D_i + \beta_2 Post_i + \beta_3 D_i Post_i + e_{it}$$

β_0 : 대조군의 기준값

β_1 : 실험군과 대조군의 기준값 차이

β_2 : 대조군의 추세

$\beta_0+\beta_1+\beta_2+\beta_3$: 개입 후 실험군의 다운로드 수준 $(D=1,\ \mathrm{Post}=1)$

β_3 실험군과 대조군 간의 차이와 시간 추세를 모두 고려한 DID 추정량!

8.2. 표준 이중차분법

- 고정효과의 측면
  - 이원고정효과 (TWFE)
    - 패널데이터는 '시간'에 따라 '대상'을 반복하여 측정하는 것.
    - 그렇기에 처치 효과, 대상 효과, 시간 효과로 분리 가능

τ : 처치 효과

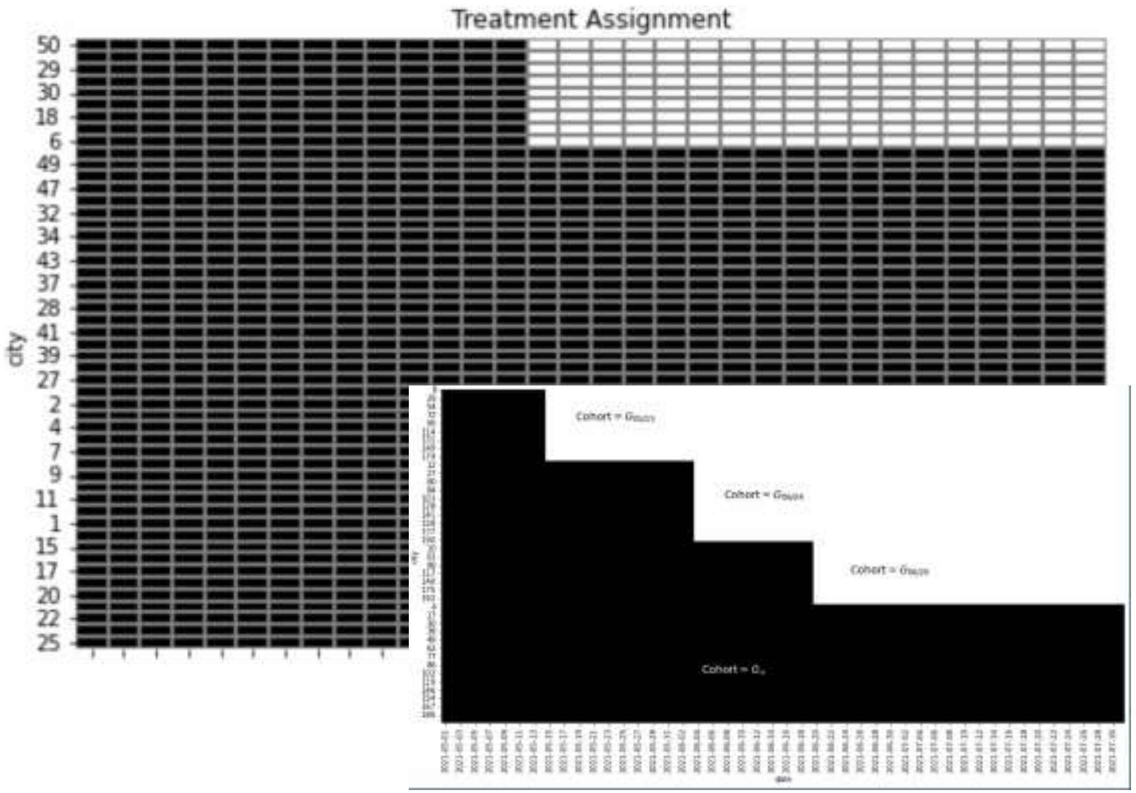
α_i : 개별 대상 고정효과

γ_i : 시간 고정효과

$$Y_{it} = \tau W_{it} + \alpha_i + \gamma_i + e_{it}$$

8.2. 표준 이중차분법

- 블록 디자인의 측면
  - 블록 단위로 처치 / 시간을 나눠본다면?



• 검은색: 해당 도시-날짜 쌍은 통제군(Control group) 상태
• 흰색: 해당 도시-날짜 쌍은 처치군(Treated group) 상태

• 실험설계에서 대상 (city)를 블록 단위로 나눠서 그룹화 가능하다!

8.2. 표준 이중차분법

- 신뢰구간 구하기
  - DID나 패널데이터 분석에서는 동일한 실험 단위가 여러 시점에 반복 관측.
  - 관측 값들 간의 '독립성'이라는 가정이 위배됨 (즉 iid 위배)
  - 전통적인 신뢰구간 계산법은 iid 가정 아래 계산되므로, 반복 관측이 섞인 경우 오차 과소 추정 가능.

8.2. 표준 이중차분법

- 대안1. 군집표준오차
  - 도시별로 군집화
  - 집계 데이터로도 수행 가능

```
m = smf.ols(
    'downloads ~ treated:post + C(city) + C(date)', data=mkt_data
).fit(cov_type='cluster', cov_kwds={'groups': mkt_data['city']})

print("ATT:", m.params["treated:post"])
m.conf_int().loc["treated:post"]
```

```
0    0.296101
1    1.087370
Name: treated:post, dtype: float64
```

```
m = smf.ols(
    'downloads ~ treated:post + C(city) + C(date)', data=did_data
).fit(cov_type='cluster', cov_kwds={'groups': did_data['city']})

print("ATT:", m.params["treated:post"])
m.conf_int().loc["treated:post"]
```

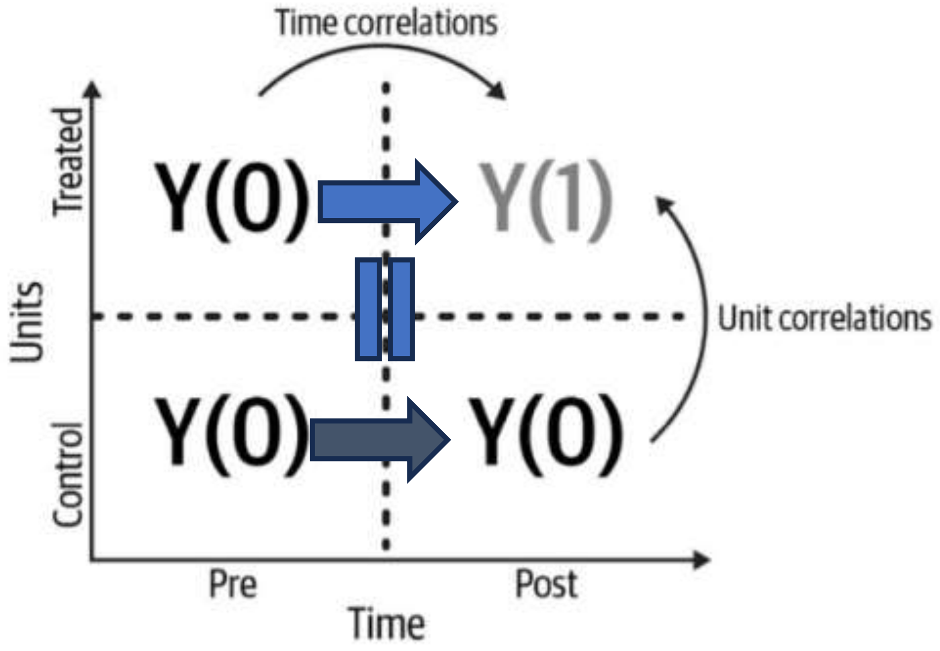
- 대안2. 블록 부트스트랩
  - 전체 실험 대상 복원추출

```
def est_fn(df):
    m = smf.ols('downloads ~ treated:post + C(city) + C(date)',
                data=df).fit()
    return m.params["treated:post"]

block_bootstrap(mkt_data, est_fn, "city")
```

```
array([0.23162214, 1.14002646])
```

8.3. 식별 가정

- 평행 추세



- 평행 추세 가정
  - 처치가 없으면 평균적으로 실험군과 대조군의 결과 추세가 동일할 것.

기대값 기준: $E[Y_{it=1}(0) \mid D=1] - E[Y_{it=0}(0) \mid D=1] = E[Y_{it=1}(0) \mid D=0] - E[Y_{it=0}(0) \mid D=0]$

조금 더 강하게 표현하면: $(\Delta y_0, \Delta y_1) \perp D_i$

8.3. 식별 가정

- 비기대과정과 SUTVA
  - Recall. SUTVA는 한 단위의 처치가 다른 단위의 결과에 영향을 주지 않아야 한다는 가정

    - e.g. 도시 A의 오프라인 마케팅이 도시 B의 다운로드 수에 영향을 주지 않아야 한다.
  - 그러나 현실에서는 ..
    - 파급효과 발생 가능.
    - ‘블랙프라이데이 (처치)’를 이미 알고 처치 전에 판매가 급증한다면?

8.3. 식별 가정

- 강외생성
  - 고정 효과 모델에서의 잔차에 대한 가정

$$Y_{it} = \alpha_i + X_{it}\beta + \epsilon_{it}, \quad \text{where } E[\epsilon_{it} \mid X_{it}, \alpha_i] = 0$$

- 시간에 따라 변하는 교란 요인 없음
  - 시간이 지남에 따라 모든 집단이 똑같은 외부환경의 변화를 겪어야함.
- 피드백 없음
  - 현재의 결과가 미래의 처치 여부에 영향 X
- 이월 효과 없음
  - 이전 시점의 처치가 현재의 결과에 영향 X

8.4. 시간에 따른 효과 변동

- 시간에 따라 효과가 변할 수 있다!
  - 전체적인 처치효과가 나타나기까지는 어느 정도 시간이 걸릴 수 있음.
  - Solution. **<u>시간에 따른</u>** ATT를 추정하자!
    - 모든 시간대를 반복하며 해당 시간대만이 처치 이후 기간인 것처럼 이중차분법을 적용.

```python
def did_date(df, date):
    df_date = (df
               .query("date==@date | post==0")
               .query("date <= @date")
               .assign(post = lambda d: (d["date"]==date).astype(int)))

    m = smf.ols(
        'downloads ~ I(treated*post) + C(city) + C(date)', data=df_date
    ).fit(cov_type='cluster', cov_kwds={'groups': df_date['city']})

    att = m.params["I(treated * post)"]
    ci = m.conf_int().loc["I(treated * post)"]

    return pd.DataFrame({"att": att, "ci_low": ci[0], "ci_up": ci[1]},
                        index=[date])
```

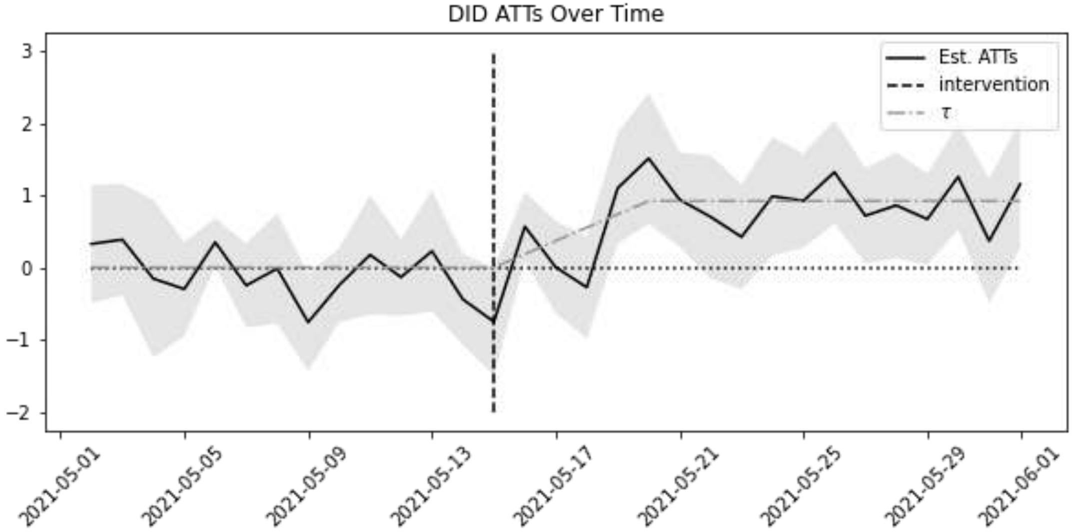
인수로 받은 데이터가 처치 후 기간이라면
필터는 아무것도 하지 않음.

인수로 받은 데이터가 처치 전 기간이라면 이후
날짜는 제외.

8.4. 시간에 따른 효과 변동

- 가정 확인
  - 처치 이전에는 ATT가 거의 0
  - 처치 이후부터 추정치가 상승함.

-> 즉, 처치가 일어나기 전 두 그룹 간의 결과는 같은 경향을 보인다.



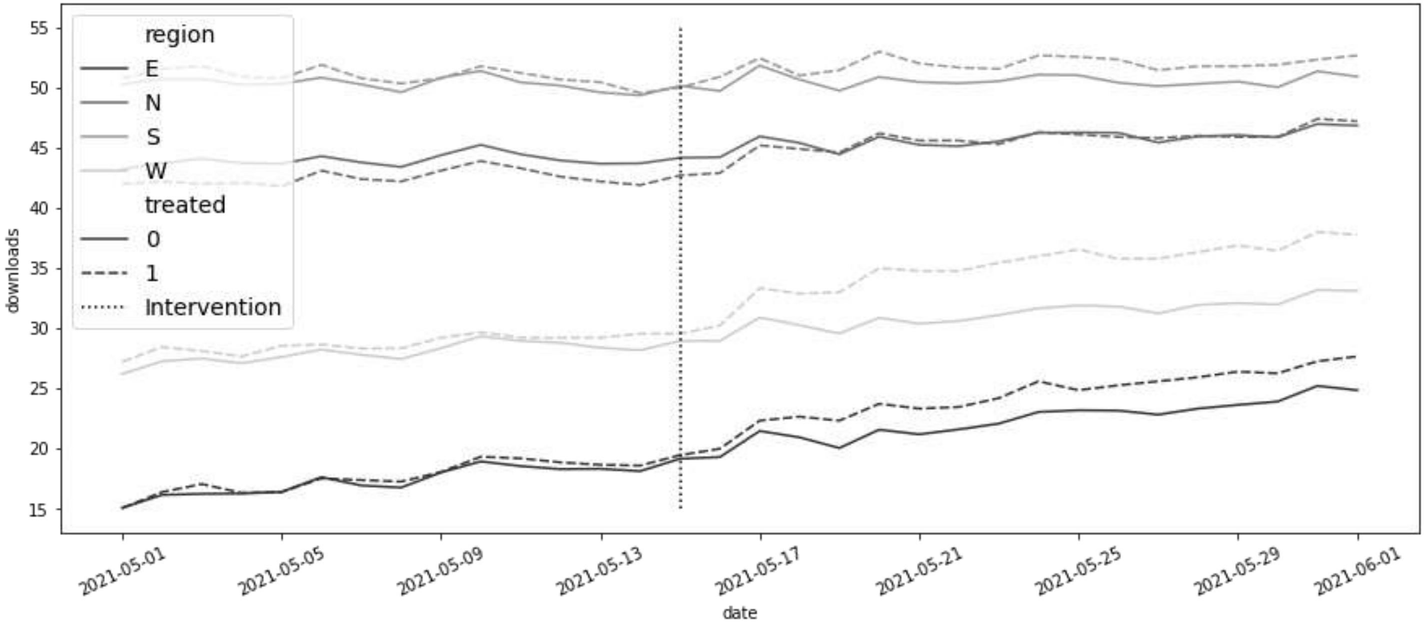
8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이전에 배웠던 방법론들과 이중차분법의 결합
  - 공변량
    - 기존의 평행추세가정에 공변량을 조건부로 둠.
  - 이중 강건 추정
    - 기본 결과 모델 대신 델타 결과 모델로 변경
    - ATT 바탕으로 실험군 재구성

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중차분법과 공변량

$$E[Y_{it=1}(0) \mid D=1, X] - E[Y_{it=0}(0) \mid D=1, X] = E[Y_{it=1}(0) \mid D=0, X] - E[Y_{it=0}(0) \mid D=0, X]$$



공변량을 고려해야하는 상황인가?

- Region (공변량)이 download (결과)에 영향을 줌.

- Region(공변량)이 처치 여부 (D)에도 영향을 줌. (특히 W지역)

즉, Region은 Confounder.

+ 지역 내에서는 평행추세 가정 만족.

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중차분법과 공변량
  - 단순하게 region을 회귀모델에 추가 ?

```
m = smf.ols('downloads ~ treated:post + C(city) + C(date) + C(region)',
            data=mkt_data_all).fit()
m.params["treated:post"]
```

이렇게 구할 수 없는 이유:

- 시간에 따라 고정된 공변량(region)은 대상 고정효과(city)에 흡수되므로, 굳이 회귀식에 다시 넣을 필요 없다.

e.g. City 1, 2 -> N, City 3, 4 -> S라면, 각 도시는 dummy 변수로 분리되며, region이 주는 효과를 이미 포함하게 됨.

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중차분법과 공변량
  - 지역별 대조군 추세를 따로 추정해야함 (가정에 의함)
  - post * treated 를 지역 (Region)에 대해 추정

$$E[Y_{it=1}(0) \mid D=1, X] - E[Y_{it=0}(0) \mid D=1, X] = E[Y_{it=1}(0) \mid D=0, X] - E[Y_{it=0}(0) \mid D=0, X]$$

```
m_saturated = smf.ols('downloads ~ (post*treated)*C(region)',
                      data=mkt_data_all).fit()

atts = m_saturated.params[m_saturated.params.index.str.contains("post:treated")]
atts
```

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중차분법과 공변량
  - 결과 해석 (기준그룹 동부지역)
  - 이후 지역별 도시 수를 가중치로 사용하여 ATT 집계.

```
post:treated                    1.676808
post:treated:C(region)[T.N]    -0.343667
post:treated:C(region)[T.S]    -0.985072
post:treated:C(region)[T.W]     1.369363
dtype: float64
```

```
reg_size = (mkt_data_all.groupby("region").size()
            /len(mkt_data_all["date"].unique()))

base = atts[0]

np.array([reg_size[0]*base]+
         [(att+base)*size
          for att, size in zip(atts[1:], reg_size[1:])]
        ).sum()/sum(reg_size)
```

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중차분법과 공변량
  - 상호작용 효과 반영한 식 변형 가능 -> 주요 상호작용 효과만 남길 수 있음.
    - **post × treated (DID 효과)**
    - **post × region (공변량 고려 - 시간별 지역 추세)**

```python
m_saturated = smf.ols('downloads ~ (post*treated)*C(region)',
                      data=mkt_data_all).fit()

atts = m_saturated.params[m_saturated.params.index.str.contains("post:treated")]
atts
```

```python
m = smf.ols('downloads ~ post*(treated + C(region))',
            data=mkt_data_all).fit()

m.summary().tables[1]
```

C(region) + post + treated
+ post:treated
+ post:C(region)
+ treated:C(region)
+ post:treated:C(region)

=> 지역별 DID 효과 계산

post + treated + C(region)
+ post:treated
+ post:C(region)

=> 공변량(region)과 시간(post)의 상호작용을 조정하면서도, 전체 집단에 대해 하나의 DID 효과를 추정

- 이중 강건 이중차분법
  - (1) 성향점수 모델
    - 실험 대상이 실험군에 속할 확률을 계산
    - 실험군에 속할 확률은 날짜와 상관없으므로, 한 시점의 데이터만 사용해도 됨.

```
unit_df = (mkt_data_all
           # keep only the first date
           .astype({"date": str})
           .query(f"date=='{mkt_data_all['date'].astype(str).min()}'")
           .drop(columns=["date"])) # just to avoid confusion

ps_model = smf.logit("treated~C(region)", data=unit_df).fit(disp=0)
```

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중 강건 이중차분법
  - (2) 델타 결과 모델
    - 델타 y에 대한 결과 모델 적합

```
df_delta_y = (unit_df
              .set_index("city")
              .join(delta_y.rename("delta_y")))

outcome_model = smf.ols("delta_y ~ C(region)", data=df_delta_y).fit()
```

8.5. ~ 8.6. 공변량과 이중 강건 추정

- 이중 강건 이중차분법
  - (3) 식 유도

$$\hat{\tau}_{\text{DRDID}} = \widehat{\Delta y_1}^{\text{DR}} - \widehat{\Delta y_0}^{\text{DR}}$$

ATT = 실험군 평균 변화량 – 대조군 평균 변화량

$$\widehat{\Delta y_1}^{\text{DR}} = \frac{1}{N_{\text{tr}}} \sum_{i \in \text{tr}} (\Delta y_i - \hat{m}(x_i))$$

실험군 변화량 = 관측 변화량 – 결과모델 변화량
(우리의 목표는 ATT이므로, treated만 고려)

$$w_{\text{co}} = \hat{e}(X) \cdot \frac{1}{1 - \hat{e}(X)}$$

대조군 가중치 (IPW)

$$\widehat{\Delta y_0}^{\text{DR}} = \frac{\sum_{i \in \text{co}} w_{\text{co}} \cdot (\Delta y_i - \hat{m}(x_i))}{\sum w_{\text{co}}}$$

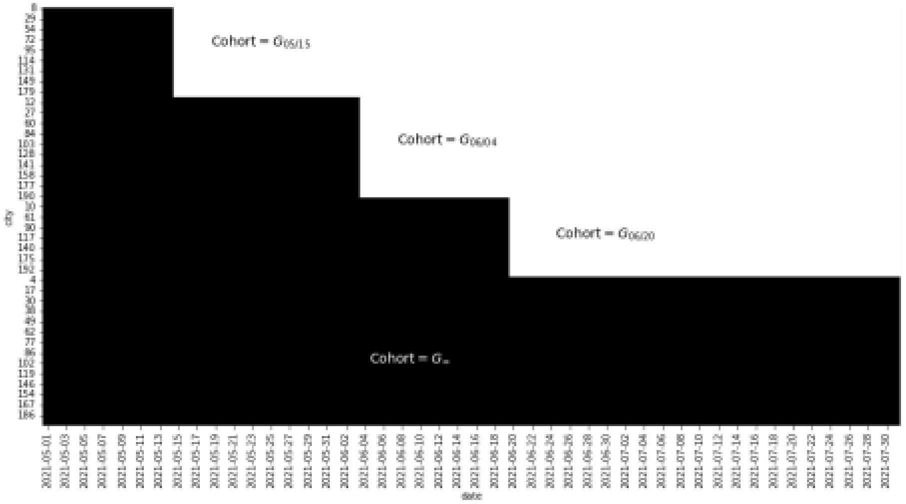
대조군의 (보정된) 변화량 = (관측 변화량 – 예측 변화량) * 가중치

```
tr = df_dr.query("treated==1")
co = df_dr.query("treated==0")

dy1_treat = (tr["delta_y"] - tr["y_hat"]).mean()

w_cont = co["ps"]/(1-co["ps"])
dy0_treat = np.average(co["delta_y"] - co["y_hat"], weights=w_cont)

print("ATT:", dy1_treat - dy0_treat)
```

8.7. 처치의 시차 도입

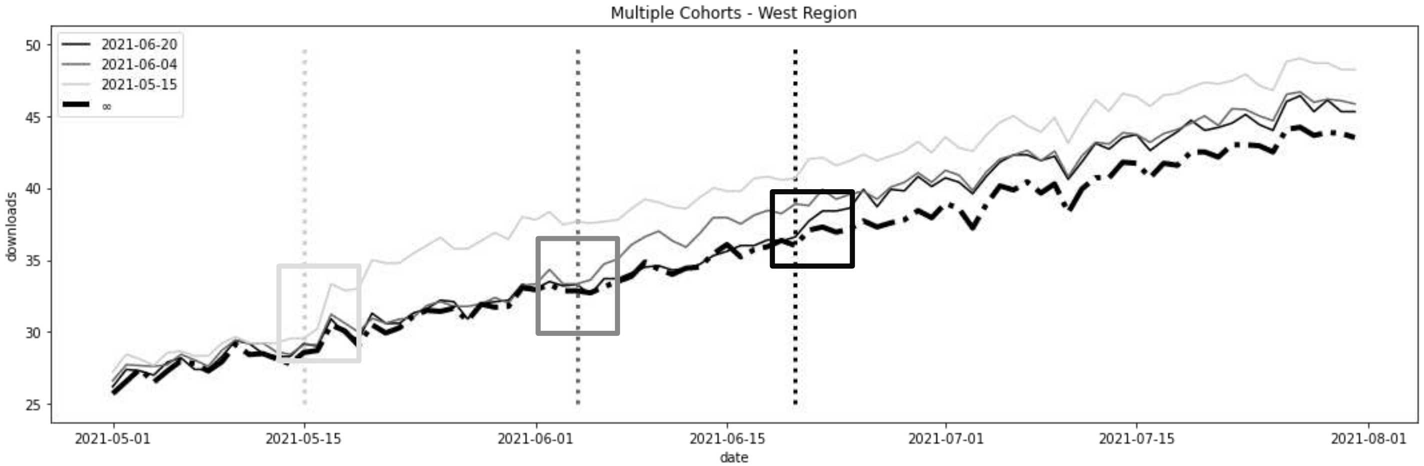
- 다시 블록 디자인을 활용.
  - 이때 처치 받는 시점이 그룹을 구분하는 기준이므로, ’코호트’라고 부름.
  - 점진적 처치를 받는 것이 특징.

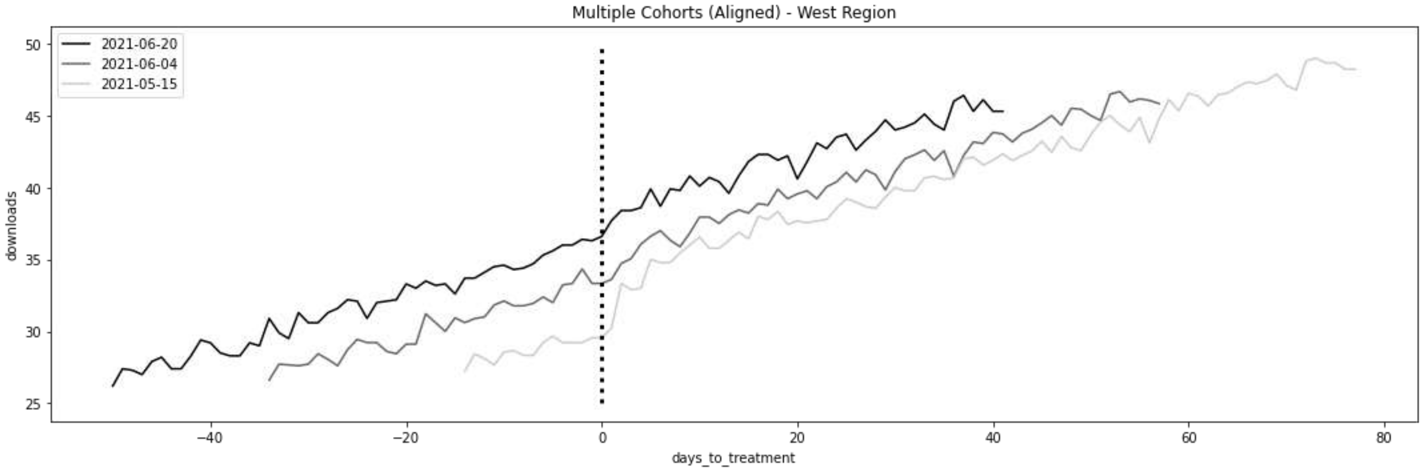
    e.g. 마케팅 캠페인, 정책 도입, 앱 업데이트 등은 지역이나 시점별로 순차적으로 일어남



8.7. 처치의 시차 도입

- West Region (공변량 고려 X)만 우선 보면 ..





$$Y_{it} = \tau W_{it} + \alpha_i + \gamma_t + e_{it}$$

```python
twfe_model = smf.ols(
    "downloads ~ treated:post + C(date) + C(city)",
    data=mkt_data_cohorts_w
).fit()

true_tau = mkt_data_cohorts_w.query("post==1&treated==1")["tau"].mean()

print("True Effect: ", true_tau)
print("Estimated ATT:", twfe_model.params["treated:post"])
```

```
True Effect:   2.2625252108176266
Estimated ATT: 1.7599504780633743
```

시간에 걸쳐 처치효과가 동일하다는 가정도 추가로 필요하지만, 시각화 결과를 보면 이를 위배.

=> 해결책: 더 유연한 모델을 사용

$$Y_{it} = \tau_{it} W_{it} + \alpha_i + \gamma_t + e_{it}$$

8.7. 처치의 시차 도입

- 다양한 tau 값 설정:

기존 모델

$$Y_{it} = \tau W_{it} + \alpha_i + \gamma_t + e_{it}$$

```python
twfe_model = smf.ols(
    "downloads ~ treated:post + C(date) + C(city)",
    data=mkt_data_cohorts_w
).fit()

true_tau = mkt_data_cohorts_w.query("post==1&treated==1")["tau"].mean()

print("True Effect: ", true_tau)
print("Estimated ATT:", twfe_model.params["treated:post"])
```

모든 실험대상에 대해 다른 효과

$$Y_{it} = \tau_{it} W_{it} + \alpha_i + \gamma_t + e_{it}$$

```python
formula = "downloads ~ treated:post:C(city):C(date) + C(city)+C(date)"

twfe_model = smf.ols(formula, data=mkt_data_cohorts_w).fit()
```

코호트에 대해 다른 효과

$$Y_{it} = \tau_{gt} W_{it} + \alpha_i + \gamma_t + e_{it}$$

```python
formula = "downloads ~ treated:post:C(cohort):C(date) + C(city)+C(date)"

twfe_model = smf.ols(formula, data=mkt_data_cohorts_w).fit()
```

코호트보다 큰 기간만 고려

$$Y_{it} = \tau_{g,t \geq g} W_{it} + \alpha_i + \gamma_t + e_{it}$$

```python
did_g = (
    df
    .loc[lambda d:(d[cohort_col] == cohort)|
                  (d[cohort_col] == nvr_treated)]
    .assign(treated = lambda d: (d[cohort_col] == cohort)*1)
    .assign(post = lambda d:(pd.to_datetime(d[date_col])>=cohort)*1)
)

att_g = smf.ols(f"{y_col} ~ treated*post",
                data=did_g).fit().params["treated:post"]
```

8.7. 처치의 시차 도입

- 시간의 이질적 효과까지 고려한 모델 & 공변량 추가:

```
formula = """
downloads ~ treated:post:C(cohort):C(date)
+ C(date):C(region) + C(city) + C(date)"""

twfe_model = smf.ols(formula, data=mkt_data_cohorts).fit()
```

treated:post:C(cohort):C(date) -> 시간과 코호트별로 분리된 ATT
C(date):C(region) -> 시간에 따라 변하는 공변량
C(city) -> 개체효과 (각 도시 고유의 특성)
C(date) -> 시간고정효과

[공변량 케이스]

```
m = smf.ols('downloads ~ post*(treated + C(region))',
            data=mkt_data_all).fit()

m.summary().tables[1]
```

Post (date) * region이 공변량 효과를 처리

[시간의 이질적 효과 케이스]

$$Y_{it} = \tau_{gt}W_{it} + \alpha_i + \gamma_t + e_{it}$$

```
formula = "downloads ~ treated:post:C(cohort):C(date) + C(city)+C(date)"

twfe_model = smf.ols(formula, data=mkt_data_cohorts_w).fit()
```

treated:post:C(cohort):C(date) 이 시간에 따른
이질적 효과를 처리

8.8. 요약

- 패널데이터와 이중차분법
  - ATT (DID)를 추정하는 방법으로, 패널데이터에 활용 가능
  - 4가지 관점 (결과 변화, OLS, 고정효과, 블록 디자인)
  - 평행 추세 가정 및 SUTVA 등의 식별 가정
  - 공변량과 이중 강건을 일부 변형하여 도입 가능
  - 시차 도입 설계시 효과 (tau)에 대한 고려 필요.