Cas simple 1 : Une période de traitement (un groupe de contrôle et un groupe de traitement)

```r
# Load required libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(broom) # for tidy function
library(plm)
```

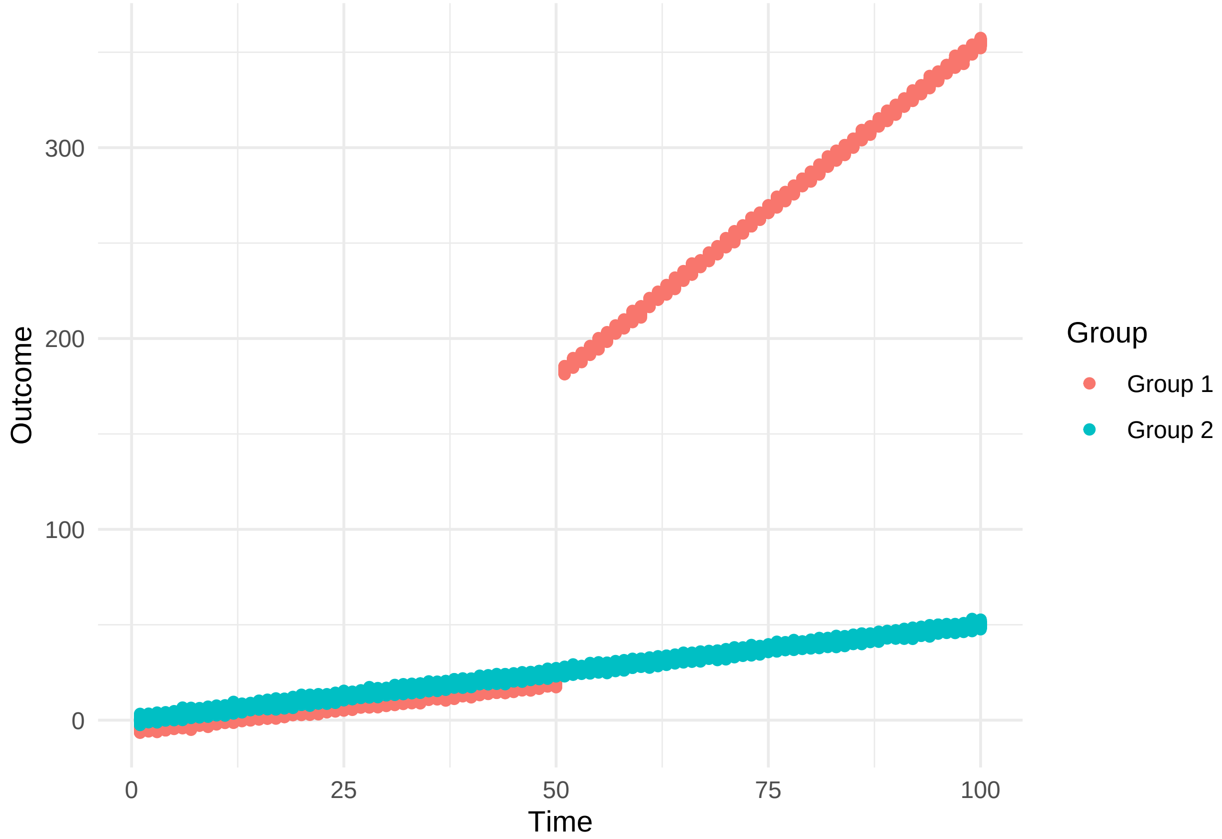
```
##
## Attaching package: 'plm'

## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```r
# Set seed for reproducibility
set.seed(1234)
```

```r
# Number of individuals per group
n_per_group <- 100

# Number of time periods
n_periods <- 100

# Time variable
time <- 1:n_periods

# Treatment period
tp=n_periods/2

# Generate synthetic data
data <- data.frame(
  Group = rep(c("Group 1", "Group 2"), each = n_per_group * n_periods),
  Individual = rep(rep(1:n_per_group, each = n_periods), times = 2),
  Time = rep(time, times = 2 * n_per_group),
  stringsAsFactors = FALSE
)


data = data %>% group_by(Individual) %>% mutate(Treatment = ifelse(Time > tp & Group == "Group 1", 1, 0
                                                Outcome = rnorm(2 * n_periods, mean = case_when(
                                                  Treatment == 1 & Group== "Group 1" ~ -5+0.5*Time+10+3
                                                  Treatment == 0 & Group== "Group 1" ~ -5+0.5*Time,
```

```r
                                              TRUE ~ 0.5*Time))) %>% distinct() %>%
  group_by(Individual, Group) %>% mutate(TreatmentPeriod = if(is.na(which(Treatment>0)[1])){0}else{whic

# Rename individuals to be unique
data$Individual = rep(1:(n_per_group*2), each = n_periods)
```

```r
# Plot the simulated data
ggplot(data, aes(x = Time, y = Outcome, color = Group, linetype = factor(Treatment))) +
  geom_point() +
  labs(x = "Time", y = "Outcome", color = "Group", linetype = "Treatment") +
  theme_minimal()
```



Méthode 1 : Diff and diff

```r
# Estimate DiD parameters using linear regression
diD_model <- lm(Outcome ~ Treatment + Time + Treatment:Time + Group, data = data)

# Display DiD model summary
summary(diD_model)
```

```
##
## Call:
## lm(formula = Outcome ~ Treatment + Time + Treatment:Time + Group,
##     data = data)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -4.1522 -0.6796  0.0051  0.6731  3.7018
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -5.0124108  0.0163599  -306.4   <2e-16 ***
## Treatment      10.0298393  0.0768490   130.5   <2e-16 ***
## Time            0.4999835  0.0003256  1535.7   <2e-16 ***
## GroupGroup 2    5.0061656  0.0190874   262.3   <2e-16 ***
## Treatment:Time  3.0000999  0.0010297  2913.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9968 on 19995 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 6.296e+07 on 4 and 19995 DF,  p-value: < 2.2e-16
```

On retrouve bien avec cette méthode les coefficients correspondant aux données simulées.

Méthode 2 : Event Study

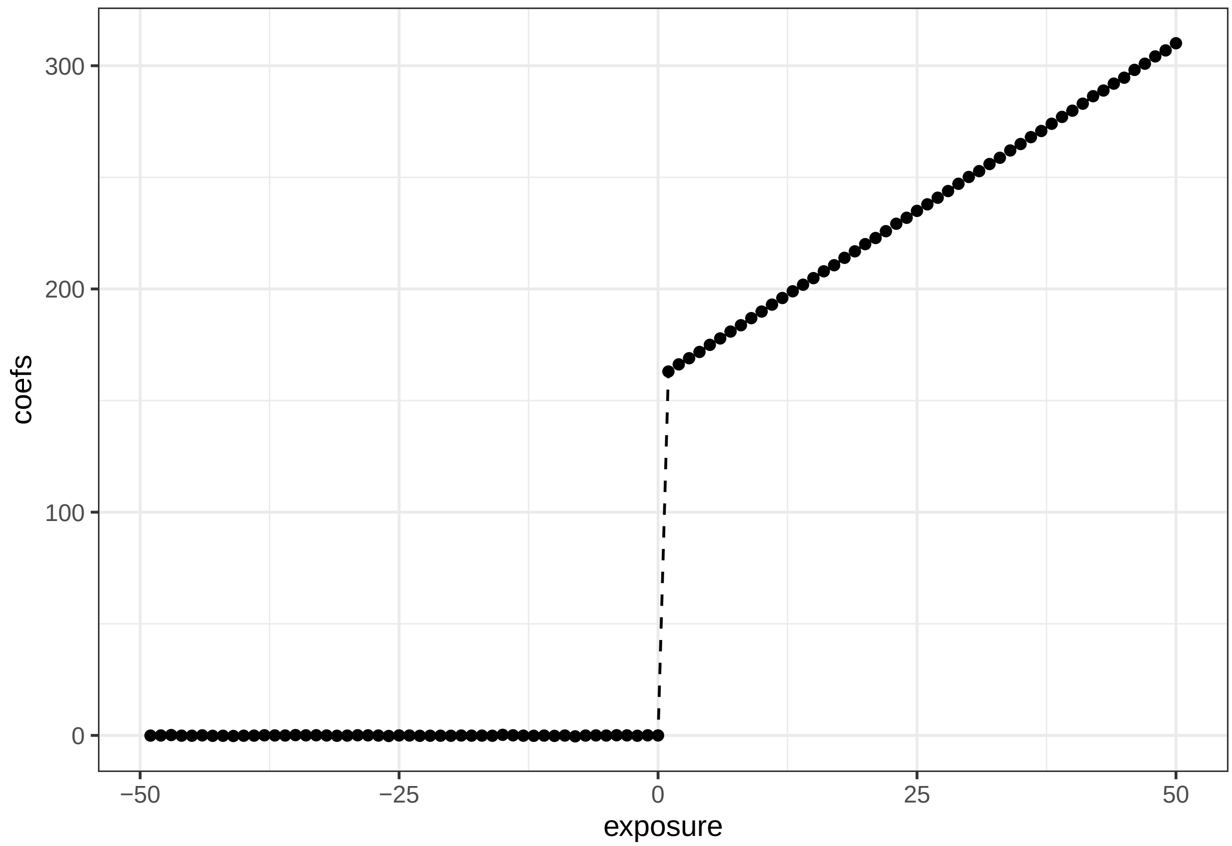
```r
# generate leads and lags of the treatment
t0 = 49 # Number of periods before the event
t1 = 50 # Number of periods after the event
Dtl <- sapply(-t0:t1, function(l) {1*((data$Time == data$TreatmentPeriod + l) & (data$TreatmentPeriod >
Dtl <- as.data.frame(Dtl)
cnames1 <- paste0("Dtmin", t0:1)
colnames(Dtl) <- c(cnames1, paste0("Dt", 0:t1))
data <- cbind.data.frame(data, Dtl)
row.names(data) <- NULL

# panel regression
pdata = pdata.frame(data, index = c("Individual", "Time", "Group"))
# table(index(pdata))

es <- plm(as.formula(paste("Outcome ~", paste(colnames(Dtl), collapse="+"))), data = pdata, model = "wi

summary(es)
```

```
## Twoways effects Within Model
##
## Call:
## plm(formula = as.formula(paste("Outcome ~", paste(colnames(Dtl),
##     collapse = "+"))), data = pdata, effect = "twoways", model = "within")
##
## Balanced Panel: n = 200, T = 100, N = 20000
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -4.1014772 -0.6809350  0.0093429  0.6670751  3.6762137
##
## Coefficients:
##             Estimate  Std. Error   t-value Pr(>|t|)
## Dtmin49   -0.0848380   0.2776345   -0.3056  0.75993
```

```
## Dtmin48  -0.0265890   0.1446932   -0.1838  0.85420
## Dtmin47   0.1518111   0.1795059    0.8457  0.39772
## Dtmin46  -0.1128145   0.1890178   -0.5968  0.55062
## Dtmin45  -0.1417542   0.1966984   -0.7207  0.47112
## Dtmin44   0.0274745   0.1995325    0.1377  0.89048
## Dtmin43  -0.1829616   0.2027249   -0.9025  0.36680
## Dtmin42  -0.1587543   0.2954808   -0.5373  0.59108
## Dtmin41  -0.2489482   0.1442307   -1.7260  0.08436 .
## Dtmin40  -0.1343139   0.1442307   -0.9312  0.35174
## Dtmin39  -0.0980593   0.2001449   -0.4899  0.62418
## Dtmin38   0.0565613   0.2029637    0.2787  0.78050
## Dtmin37  -0.0018132   0.1435938   -0.0126  0.98993
## Dtmin36  -0.0375044   0.1435938   -0.2612  0.79395
## Dtmin35   0.1370507   0.1435938    0.9544  0.33988
## Dtmin34   0.0519034   0.1435938    0.3615  0.71776
## Dtmin33   0.0757418   0.1948982    0.3886  0.69756
## Dtmin32  -0.0270310   0.1984376   -0.1362  0.89165
## Dtmin31  -0.1596890   0.1419735   -1.1248  0.26070
## Dtmin30  -0.0783146   0.1419735   -0.5516  0.58122
## Dtmin29   0.0727955   0.1421893    0.5120  0.60868
## Dtmin28   0.0402223   0.1421893    0.2829  0.77727
## Dtmin27  -0.0457139   0.1994474   -0.2292  0.81871
## Dtmin26  -0.2562596   0.2006254   -1.2773  0.20151
## Dtmin25  -0.0175800   0.1993682   -0.0882  0.92974
## Dtmin24  -0.0399701   0.2690604   -0.1486  0.88191
## Dtmin23  -0.1843842   0.2787028   -0.6616  0.50825
## Dtmin22  -0.1446093   0.1816290   -0.7962  0.42594
## Dtmin21  -0.1313017   0.1849102   -0.7101  0.47766
## Dtmin20  -0.1614197   0.1948564   -0.8284  0.40745
## Dtmin19  -0.0439616   0.1984866   -0.2215  0.82472
## Dtmin18  -0.1043691   0.2027748   -0.5147  0.60677
## Dtmin17  -0.0876265   0.2874729   -0.3048  0.76051
## Dtmin16  -0.1428987   0.1466309   -0.9745  0.32980
## Dtmin15   0.2564449   0.1466309    1.7489  0.08032 .
## Dtmin14   0.0483639   0.2001838    0.2416  0.80909
## Dtmin13  -0.1032809   0.2037983   -0.5068  0.61231
## Dtmin12  -0.1638498   0.1457706   -1.1240  0.26102
## Dtmin11  -0.0734073   0.1457706   -0.5036  0.61456
## Dtmin10  -0.2031055   0.1457706   -1.3933  0.16354
## Dtmin9   -0.0501288   0.1457706   -0.3439  0.73093
## Dtmin8   -0.4075577   0.1925638   -2.1165  0.03432 *
## Dtmin7   -0.0584737   0.1975356   -0.2960  0.76722
## Dtmin6    0.0139280   0.1424856    0.0978  0.92213
## Dtmin5   -0.0542031   0.1424856   -0.3804  0.70364
## Dtmin4    0.0833392   0.1429671    0.5829  0.55995
## Dtmin3    0.0203764   0.1429671    0.1425  0.88667
## Dtmin2   -0.1406100   0.1995371   -0.7047  0.48102
## Dtmin1    0.0437988   0.2012036    0.2177  0.82768
## Dt0     163.0197859   0.1993682  817.6820  < 2e-16 ***
## Dt1     166.2154259   0.2414174  688.4982  < 2e-16 ***
## Dt2     168.9774087   0.2553060  661.8624  < 2e-16 ***
## Dt3     171.7849081   0.1718170  999.8131  < 2e-16 ***
## Dt4     174.9750424   0.1757548  995.5632  < 2e-16 ***
## Dt5     177.8375144   0.1891585  940.1510  < 2e-16 ***
```

```
## Dt6     180.9201184   0.1938825  933.1433  < 2e-16 ***
## Dt7     183.7534068   0.2001265  918.1862  < 2e-16 ***
## Dt8     186.9355308   0.2680464  697.3999  < 2e-16 ***
## Dt9     189.8867704   0.1535919 1236.3075  < 2e-16 ***
## Dt10    192.9853918   0.1535919 1256.4818  < 2e-16 ***
## Dt11    195.9532903   0.1993704  982.8606  < 2e-16 ***
## Dt12    198.9563019   0.2041230  974.6885  < 2e-16 ***
## Dt13    201.9005547   0.1537662 1313.0358  < 2e-16 ***
## Dt14    204.8529589   0.1537662 1332.2364  < 2e-16 ***
## Dt15    207.9090155   0.1537662 1352.1111  < 2e-16 ***
## Dt16    210.6348845   0.1537662 1369.8385  < 2e-16 ***
## Dt17    213.9506803   0.1861005 1149.6512  < 2e-16 ***
## Dt18    216.8603298   0.1942042 1116.6613  < 2e-16 ***
## Dt19    220.0839716   0.1439895 1528.4728  < 2e-16 ***
## Dt20    222.8286746   0.1439895 1547.5346  < 2e-16 ***
## Dt21    225.8837404   0.1457553 1549.7461  < 2e-16 ***
## Dt22    229.2186593   0.1457553 1572.6263  < 2e-16 ***
## Dt23    231.8921761   0.1999465 1159.7712  < 2e-16 ***
## Dt24    234.9846529   0.2027617 1158.9202  < 2e-16 ***
## Dt25    237.9159847   0.1993682 1193.3497  < 2e-16 ***
## Dt26    240.8758186   0.1993682 1208.1958  < 2e-16 ***
## Dt27    243.8677160   0.1993682 1223.2027  < 2e-16 ***
## Dt28    247.1093225   0.1993682 1239.4621  < 2e-16 ***
## Dt29    250.1736039   0.1993682 1254.8320  < 2e-16 ***
## Dt30    252.7949522   0.1993682 1267.9803  < 2e-16 ***
## Dt31    255.9655394   0.1993682 1283.8835  < 2e-16 ***
## Dt32    258.7683175   0.1993682 1297.9418  < 2e-16 ***
## Dt33    262.0441126   0.1993682 1314.3727  < 2e-16 ***
## Dt34    264.9373604   0.1993682 1328.8847  < 2e-16 ***
## Dt35    268.0247379   0.1993682 1344.3706  < 2e-16 ***
## Dt36    270.7725216   0.1993682 1358.1530  < 2e-16 ***
## Dt37    274.0053730   0.1993682 1374.3685  < 2e-16 ***
## Dt38    277.0708710   0.1993682 1389.7446  < 2e-16 ***
## Dt39    279.8537136   0.1993682 1403.7029  < 2e-16 ***
## Dt40    283.0037597   0.1993682 1419.5030  < 2e-16 ***
## Dt41    286.3580424   0.1993682 1436.3276  < 2e-16 ***
## Dt42    288.8848162   0.1993682 1449.0015  < 2e-16 ***
## Dt43    291.9885060   0.1993682 1464.5691  < 2e-16 ***
## Dt44    294.6415957   0.1993682 1477.8766  < 2e-16 ***
## Dt45    298.1305472   0.1993682 1495.3766  < 2e-16 ***
## Dt46    300.9015457   0.1993682 1509.2755  < 2e-16 ***
## Dt47    304.1765892   0.1993682 1525.7026  < 2e-16 ***
## Dt48    306.8634167   0.1993682 1539.1794  < 2e-16 ***
## Dt49    310.0831476   0.1993682 1555.3290  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    74630000
## Residual Sum of Squares: 19478
## R-Squared:      0.99974
## Adj. R-Squared: 0.99973
## F-statistic: -3217630 on 99 and 19602 DF, p-value: 1
```

```r
# Plot
coefs1 <- coef(es)
ses1 <- sqrt(diag(summary(es)$vcov))
idx.pre <- 1:t0
idx.post <- (t1):length(coefs1)
coefs <- c(coefs1[idx.pre], 0, coefs1[idx.post])
ses <- c(ses1[idx.pre], 0, ses1[idx.post])
exposure <- -t0:t1

cmat <- data.frame(coefs=coefs, ses=ses, exposure=exposure)

ggplot(data = cmat, mapping = aes(y = coefs, x = exposure)) +
  geom_line(linetype = "dashed") +
  geom_point() +
  geom_errorbar(aes(ymin = (coefs-1.96*ses), ymax = (coefs+1.96*ses)), width = 0.2) +
  theme_bw()
```




```r
valeurs_reg_Dt <- as.numeric(coef(es))


valeurs_reg_Dt_aprestraitement <- valeurs_reg_Dt[(t0+2):(t1+t0+1)]

df1 = data.frame(y = valeurs_reg_Dt_aprestraitement, x = c(1:t1))


maregression = lm(y~x, data = df1)
```

```r
summary(maregression)
```

```
##
## Call:
## lm(formula = y ~ x, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33870 -0.08228  0.00681  0.06526  0.38303
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.629e+02  4.150e-02    3926   <2e-16 ***
## x           3.002e+00  1.445e-03    2078   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.143 on 47 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 4.317e+06 on 1 and 47 DF,  p-value: < 2.2e-16
```

Cette méthode nous permet de retrouver les résultats attendus : l'intercept vaut bien $160 = 10 + 3 \times 50$.

Méthode 3 : Utilisation du package Diff and Diff de Callaway