Exercise 3

This third example explores the use of statistical adjustment using a parametric model, but introduces some applied complications. We will explore what happens when we misspecify a parametric model, or when the area of common support is small. Again, we always assume that the given covariates fulfill ignorability.

Here is a worked-out, and commented example of using a parametric model and emmeans, for adjustment. I am also demonstrating a quick model check, a re-specification of the model, and the convex hull. The data is simulated, and if you like you can ignore the simulation code. It is provided here for the sole reasons that you can reproduce the example quickly without having to download data files.

```
####simulation code#############################
set.seed(123456)
n <- 500
u <- rnorm(n,0,1)
x1 <- .9*u + rnorm(n,2,.6)
x2 <- .9*u + rnorm(n,2,.6)
p <- (1/(1+exp(-.5+.1*x1+.3*I(x1^2))))
t <- rbinom(n,1,p)
y <- .5*t - .9*x1 - .9*x2 + .3*I(x1^2) - .3*I(x2^2) + .4*t*x1 - .3*t*x2 + .3*t*I(x1^2) + rnorm(n,0,.2)
t <- factor(t)
dfex3 <- data.frame(x1,x2,t,y)
dfex3$t <- factor(dfex3$t)
###############################################

####analysis code#############################
library(emmeans)
library(WhatIf)
#unadjusted effect
lm.u <- lm(y~t,dfex3)
summary(lm.u)
```

```
##
## Call:
## lm(formula = y ~ t, data = dfex3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5768 -1.3711  0.1885  1.5260  6.8981
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.0102     0.1134  -35.36   <2e-16 ***
## t1            2.5339     0.2024   12.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.1 on 498 degrees of freedom
## Multiple R-squared:  0.2394, Adjusted R-squared:  0.2379
## F-statistic: 156.8 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
summary(emmeans(lm.u,"t",contr="pairwise",weights="proportional"),infer=TRUE)
```

```
## $emmeans
```

```
##  t    emmean        SE  df  lower.CL  upper.CL t.ratio p.value
##  0 -4.010187 0.1134099 498 -4.233008 -3.787366 -35.360  <.0001
##  1 -1.476267 0.1676287 498 -1.805613 -1.146920  -8.807  <.0001
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate        SE  df  lower.CL  upper.CL t.ratio p.value
##  0 - 1    -2.53392 0.2023887 498 -2.931561 -2.136279  -12.52  <.0001
##
## Confidence level used: 0.95
```

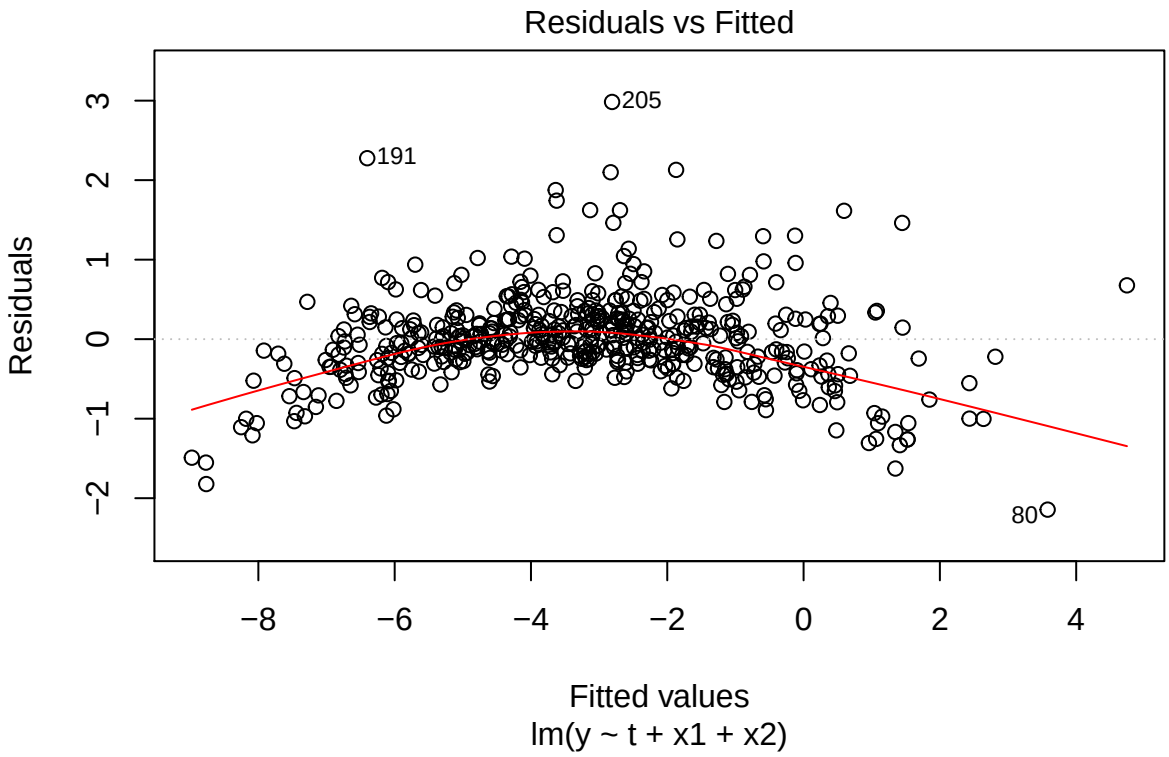
In a first step, I am looking at the unadjusted model again - the prima facie effect, using summary and emmeans. Again, I recommend you focus on the latter.

We now repeat the same exercise again, adjust on all observed covariates, and re-estimate the model.

```
#linear adjustment with plot
lm.a1 <- lm(y~t+x1+x2,dfex3)
summary(lm.a1)
```

```
##
## Call:
## lm(formula = y ~ t + x1 + x2, data = dfex3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.14393 -0.28644 -0.02871  0.25656  2.98339
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.72787    0.07124  -10.22   <2e-16 ***
## t1           1.75478    0.05990   29.30   <2e-16 ***
## x1           0.77972    0.03475   22.43   <2e-16 ***
## x2          -2.29455    0.03127  -73.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5648 on 496 degrees of freedom
## Multiple R-squared:  0.9452, Adjusted R-squared:  0.9449
## F-statistic:  2853 on 3 and 496 DF,  p-value: < 2.2e-16
```

```
plot(lm.a1,which = 1)
```




```
summary(emmeans(lm.a1,"t",contr="pairwise",weights="proportional"),infer=TRUE)
```

```
## $emmeans
##  t    emmean         SE  df  lower.CL  upper.CL  t.ratio p.value
##  0 -3.765538 0.03149276 496 -3.827414 -3.703662 -119.568  <.0001
##  1 -2.010754 0.04823428 496 -2.105523 -1.915985  -41.687  <.0001
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast  estimate         SE  df  lower.CL  upper.CL t.ratio p.value
##  0 - 1    -1.754784 0.05990027 496 -1.872473 -1.637094 -29.295  <.0001
##
## Confidence level used: 0.95
```

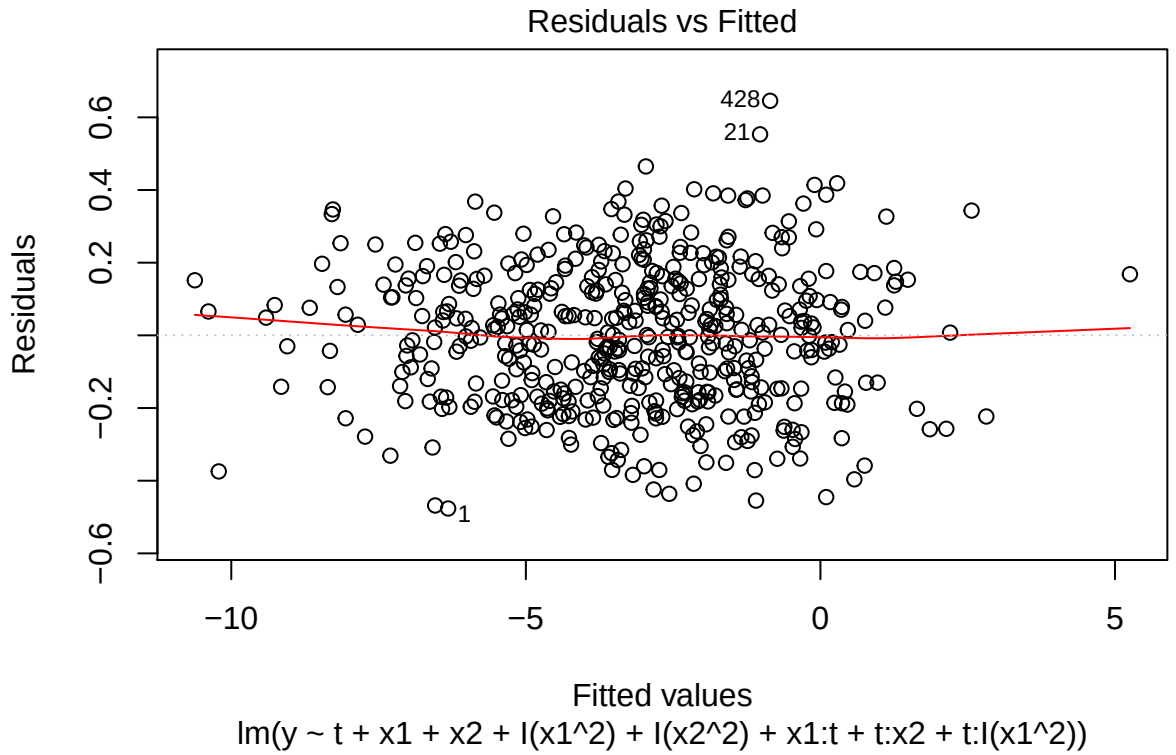
We can see in the residual plot that the model is misspecified. Here, I already know the functional form (because I simulated the data). In a real, applied settting, I might either have some theory that guide me to the functional form, or I need to try several models, and check residuals several times.

```
#non-linear adjustment
lm.a <- lm(y~t+x1+x2+I(x1^2) + I(x2^2) + x1:t + t:x2 + t:I(x1^2),dfex3)
summary(lm.a)
```

```
##
## Call:
## lm(formula = y ~ t + x1 + x2 + I(x1^2) + I(x2^2) + x1:t + t:x2 +
##     t:I(x1^2), data = dfex3)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47645 -0.15435  0.00123  0.14040  0.64546
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.059719   0.042475   1.406     0.16
## t1           0.447177   0.052859   8.460 3.10e-16 ***
## x1          -0.912276   0.044799 -20.364  < 2e-16 ***
## x2          -0.925154   0.033398 -27.701  < 2e-16 ***
## I(x1^2)      0.305274   0.008528  35.796  < 2e-16 ***
## I(x2^2)     -0.302433   0.006858 -44.096  < 2e-16 ***
## t1:x1        0.401637   0.054758   7.335 9.24e-13 ***
## t1:x2       -0.263793   0.025265 -10.441  < 2e-16 ***
## t1:I(x1^2)   0.296144   0.014454  20.488  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1963 on 491 degrees of freedom
## Multiple R-squared:  0.9934, Adjusted R-squared:  0.9933
## F-statistic:  9306 on 8 and 491 DF,  p-value: < 2.2e-16
```

```
plot(lm.a,which=1)
```




```
summary(emmeans(lm.a,"t",contr="pairwise",weights="proportional"),infer=TRUE)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## $emmeans
##  t    emmean         SE  df  lower.CL  upper.CL  t.ratio p.value
##  0 -3.618121 0.01335891 491 -3.644368 -3.591873 -270.840  <.0001
##  1 -1.686490 0.01956826 491 -1.724938 -1.648043  -86.185  <.0001
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate         SE  df  lower.CL  upper.CL t.ratio p.value
##  0 - 1    -1.93163 0.02316996 491 -1.977155 -1.886106 -83.368  <.0001
##
## Confidence level used: 0.95
```

We see that the more complex model has well-behaved residuals, and also the treatment effect is quite different. This can happen with different parametric models.

Finally, I restrict the model to the convex hull. I am selecting only units that fall within the complete overlap on all covariates. This restricts the sample quite a bit. For demonstration purposes, I also re-run both the purely linear, and the complex parametric model.

```
#model within the convex hull
library(WhatIf)
treated <- dfex3[dfex3$t==1,]
control <- dfex3[dfex3$t==0,]
wf1 <- whatif(~x1+x2,treated,control,mc.cores = 1)
wf2 <- whatif(~x1+x2,control,treated,mc.cores = 1)
```

```
## Preprocessing data ...

## Performing convex hull test ...

## Calculating distances ....

## Calculating the geometric variance...

## Calculating cumulative frequencies ...

## Finishing up ...

## Preprocessing data ...

## Performing convex hull test ...

## Calculating distances ....

## Calculating the geometric variance...

## Calculating cumulative frequencies ...

## Finishing up ...
```

```
#hull
table(wf1$in.hull)
```

```
##
## FALSE  TRUE
##    57   286
```

```
table(wf2$in.hull)
```

```
##
## FALSE  TRUE
##     5   152
```

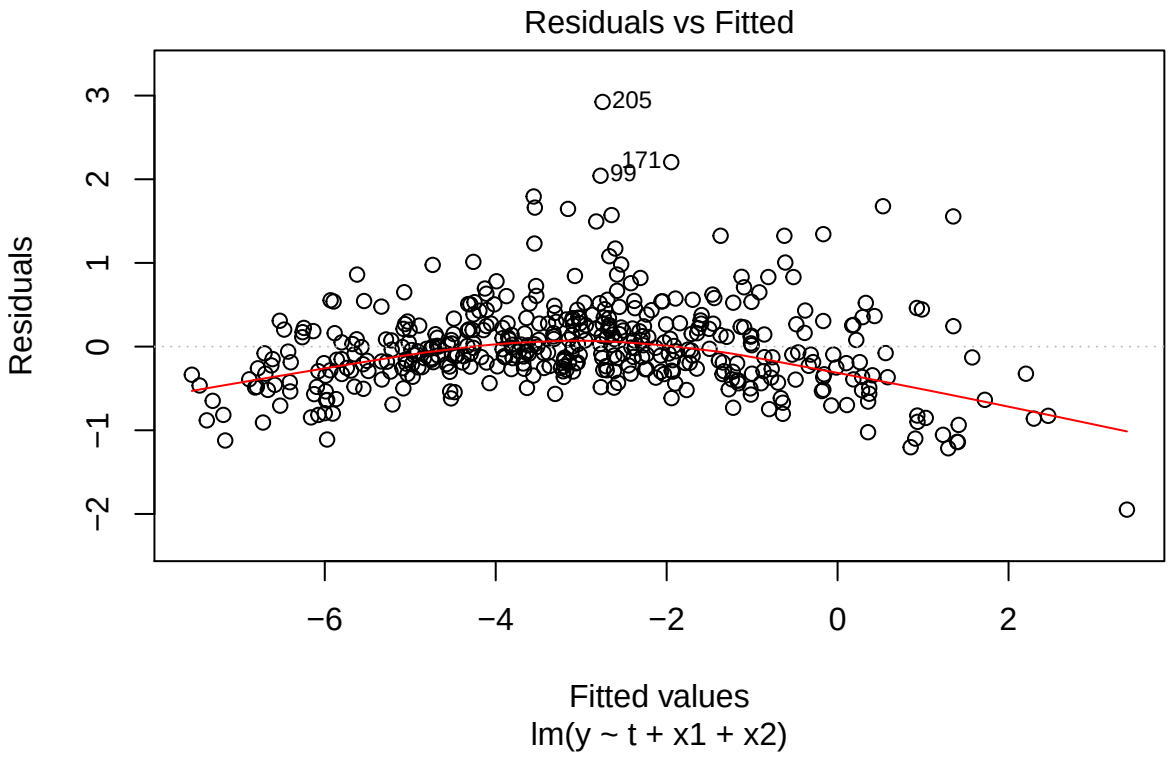
```r
treated$hull <- wf2$in.hull
control$hull <- wf1$in.hull

combined <- rbind(treated,control)
combined <- combined[combined$hull==TRUE,]

lm.a1hull <- lm(y~t+x1+x2,combined)
summary(lm.a1hull)
```

```
##
## Call:
## lm(formula = y ~ t + x1 + x2, data = combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94835 -0.29353 -0.05658  0.24974  2.92363
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.83556    0.07549  -11.07   <2e-16 ***
## t1           1.74537    0.05700   30.62   <2e-16 ***
## x1           0.76733    0.04188   18.32   <2e-16 ***
## x2          -2.21757    0.03393  -65.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5316 on 434 degrees of freedom
## Multiple R-squared:  0.941,  Adjusted R-squared:  0.9406
## F-statistic:  2307 on 3 and 434 DF,  p-value: < 2.2e-16
```

```r
plot(lm.a1hull,which = 1)
```




```
summary(emmeans(lm.a1hull,"t",contr="pairwise",weights="proportional"),infer=TRUE)
```

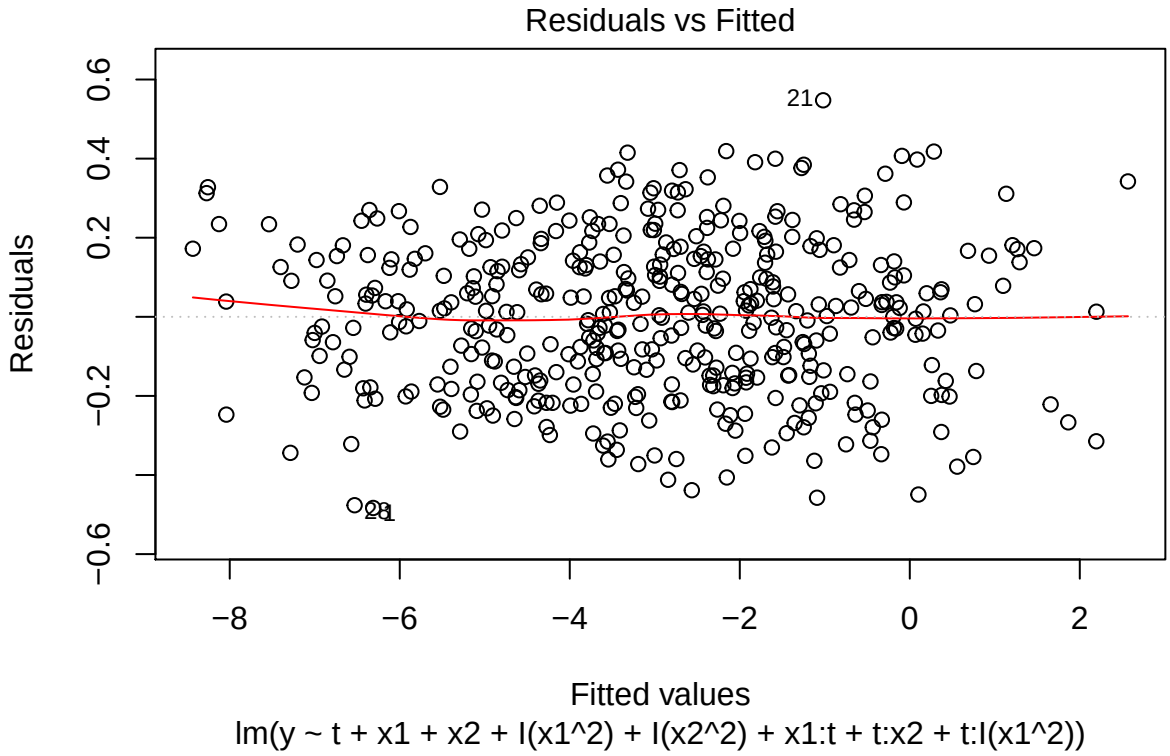
```
## $emmeans
##  t    emmean         SE  df  lower.CL  upper.CL  t.ratio p.value
##  0 -3.615295 0.03219320 434 -3.678569 -3.552021 -112.300  <.0001
##  1 -1.869921 0.04505991 434 -1.958483 -1.781358  -41.499  <.0001
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast  estimate        SE  df  lower.CL  upper.CL t.ratio p.value
##  0 - 1    -1.745374 0.0570002 434 -1.857405 -1.633343  -30.62  <.0001
##
## Confidence level used: 0.95
```

```
lm.ahull <- lm(y~t+x1+x2+I(x1^2) + I(x2^2) + x1:t + t:x2 + t:I(x1^2),combined)
summary(lm.ahull)
```

```
##
## Call:
## lm(formula = y ~ t + x1 + x2 + I(x1^2) + I(x2^2) + x1:t + t:x2 +
##     t:I(x1^2), data = combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48330 -0.15207  0.00926  0.14514  0.54745
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.074776   0.049284   1.517     0.13
## t1           0.414212   0.068520   6.045 3.25e-09 ***
## x1          -0.944301   0.066422 -14.217  < 2e-16 ***
## x2          -0.921252   0.039863 -23.110  < 2e-16 ***
## I(x1^2)      0.311417   0.016094  19.350  < 2e-16 ***
## I(x2^2)     -0.300340   0.008938 -33.602  < 2e-16 ***
## t1:x1        0.475308   0.087777   5.415 1.02e-07 ***
## t1:x2       -0.280114   0.027103 -10.335  < 2e-16 ***
## t1:I(x1^2)   0.278417   0.024615  11.311  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1967 on 429 degrees of freedom
## Multiple R-squared:  0.992,  Adjusted R-squared:  0.9919
## F-statistic:  6660 on 8 and 429 DF,  p-value: < 2.2e-16
```

```
plot(lm.ahull,which = 1)
```




```
summary(emmeans(lm.ahull,"t",contr="pairwise",weights="proportional"),infer=TRUE)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## $emmeans
##  t    emmean         SE  df  lower.CL  upper.CL  t.ratio p.value
##  0 -3.422938 0.01576542 429 -3.453925 -3.391951 -217.117  <.0001
##  1 -1.720658 0.02112860 429 -1.762187 -1.679130  -81.437  <.0001
```

```
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate         SE  df  lower.CL  upper.CL t.ratio p.value
##  0 - 1    -1.70228 0.02551215 429 -1.752424 -1.652136 -66.724  <.0001
##
## Confidence level used: 0.95
```

What we see is that the treatment effect (while not as good as the correct parametric model on all units) is more consistent across different model specifications.

Exercise:

1.) Download the file dfex3a from github (https://raw.githubusercontent.com/felixthoemmes/IPN_workshop/master/dfex3a.csv). You can download this file directly into R (no need to navigate to github in a browser, using the following code snippet:

```
library(readr)
dfex3a <- read_csv("https://raw.githubusercontent.com/felixthoemmes/IPN_workshop/master/dfex3a.csv")
```

The file contains a treatment t, an outcome y, and covariates x1-x4. We assume that these variables are those that fulfill the back-door criterion. Obtain an unadjusted estimate for the effect of t on y, using the emmean statement, and interpret the results.

2.) Now use a *linear* parametric model for adjustment, using covariates x1-x4. Obtain the adjusted treatment effect, and compare it to the unadjusted estimate.

3.) Check the residuals of the linear model, and comment on them.

4.) Try to expand your model, and keep checking residuals, until you find a parametric model that seems to fit well. Once you have this model, report the treatment effect estimate using emmeans, and compare it to the estimate for the previous model.

5.) Now construct the convex hull, and restrict your sample to it. Then run both the linear and complex parametric model, and compare estimates.