```
clear

set more off

use ProjectRobotsEverywhereData.dta

//*********************   NAMEN GEVEN  *****************************************

//Namen voor deel 1:
drop A
rename B age
rename C gender
rename D degree
rename E milieu1
rename F influence1
rename G milieu2
rename H useless1
rename I influence2
rename J useless2
rename K milieu3
rename L milieu4
rename M milieu5
rename N influence3
rename O useless3
rename P milieu6
rename Q useless4
rename R milieu7
rename S influence4
rename T milieu8
rename U useless5
rename V english


//Namen voor deel 2:

rename W shower_everyday
rename X shower_10min
rename Y shower_turnoff
rename Z shower_rain
rename AA shower_toolong


//Namen voor deel 3:

rename AB like1
rename AC animacy1
rename AD convince1
rename AE like2
rename AF animacy2
rename AG animacy3
rename AH convince2
rename AI animacy4
```

```
rename AJ like3
rename AK convince3
rename AL convince4
rename AM like4
rename AN open_to_adjust
rename AO like5
rename AP comments
rename AQ goback_tochange
rename AR understand_allquestions
rename AS distracted
rename AT emotion
rename AU version
```

//De oude namen van de variabelen kunnen nu weggehaald worden.

```
drop if _n == 1
```

//************************Coderen van variabelen********************************

//Female en goback_tochange zijn nu een 0/1 variabelen.

```
encode gender, gen(gender2)
```

```
recode gender2(1=1) (2=0), gen(gender3)
```

```
drop gender gender2
```

```
rename gender3 female
```

```
label var female "1=female 0=male"
```

//Coderen van animacy* variabelen.

```
 foreach v in animacy1 animacy2 animacy3 animacy4 {
        encode `v', gen(`v'A)
        drop `v'
        rename `v'A `v'
        }
```

//Coderen van like* variabelen.

```
foreach v in like1 like2 like3 like4 like5 {
        encode `v', gen(`v'A)
        drop `v'
        rename `v'A `v'
        }
```

//Coderen van goback_tochange

```
encode goback_tochange, gen(goback_tochange2)
```

```
recode goback_tochange2(1=0) (2=1), gen(goback_tochange3)
```

```
drop goback_tochange goback_tochange2

rename goback_tochange3 goback_tochange

label var goback_tochange "1=yes 0=no"


//Coderen van degree

encode degree, gen(degree2)

recode degree2 (1=3) (2=6) (3=2) (4=5) (5=1) (6=7) (7=4), gen(degree3)

drop degree degree2

rename degree3 degree

label var degree "1=primary school 2=mavo 3=havo 4=vwo 5=mbo 6=hbo 7=uni"


//Coderen van de douche vragen (deel 2)

encode shower_everyday, gen(shower_everyday2)

recode shower_everyday2 (1=1) (2=0), gen(shower_everyday3)

drop shower_everyday shower_everyday2

rename shower_everyday3 shower_everyday


encode shower_10min, gen(shower_10min2)

recode shower_10min2 (1=1) (2=0), gen(shower_10min3)

drop shower_10min shower_10min2

rename shower_10min3 shower_10min


encode shower_turnoff, gen(shower_turnoff2)

recode shower_turnoff2 (1=0) (2=1), gen(shower_turnoff3)

drop shower_turnoff shower_turnoff2

rename shower_turnoff3 shower_turnoff
```

```
encode shower_rain, gen(shower_rain2)

recode shower_rain2 (1=1) (2=0), gen(shower_rain3)

drop shower_rain shower_rain2

rename shower_rain3 shower_rain


encode shower_toolong, gen(shower_toolong2)

recode shower_toolong2 (1=1) (2=0), gen(shower_toolong3)

drop shower_toolong shower_toolong2

rename shower_toolong3 shower_toolong


//Coderen van de Likert-scale variabelen


//Coderen van alle milieu* variabelen.

 foreach v in milieu1 milieu2 milieu3 milieu4 milieu5 milieu6 milieu7 milieu8 {

        encode `v', gen(`v'A)
        recode `v'A (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(`v'B)
        drop `v' `v'A
        rename `v'B `v'
        }


//Coderen van alle influence* variabelen behalve influence2.

foreach v in influence1 influence2 influence3 influence4{
        encode `v', gen(`v'A)
        recode `v'A (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(`v'B)
        drop `v' `v'A
        rename `v'B `v'
        }


//Coderen van alle useless* variabelen.

foreach v in useless1 useless2 useless3 useless4 useless5{
        encode `v', gen(`v'A)
        recode `v'A (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(`v'B)
        drop `v' `v'A
        rename `v'B `v'
        }
```

```
//Coderen van alle convince* variabelen.

foreach v in convince1 convince2 convince3 convince4{
        encode `v', gen(`v'A)
        recode `v'A (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(`v'B)
        drop `v' `v'A
        rename `v'B `v'
        }


//Coderen van de variabel english.

encode english, gen(englishA)

recode englishA (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(englishB)

drop english englishA

rename englishB english


//Coderen van distracted

encode distracted, gen(distractedA)

recode distractedA (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(distractedB)

drop distracted distractedA

rename distractedB distracted


//Coderen van open_to_adjust

encode open_to_adjust, gen(open_to_adjustA)

recode open_to_adjustA (1=0) (2=1) (3=-1) (4=2) (5=-2), gen(open_to_adjustB)

drop open_to_adjust open_to_adjustA

rename open_to_adjustB open_to_adjust


//Coderen van version

encode version, gen(versionA)

drop version
```

```
rename versionA version


//Coderen van emotion

encode emotion, gen(emotionA)

recode emotionA (1=0) (2=1), gen(emotionB)

drop emotion emotionA

rename emotionB emotion


//************************    VARIABELEN LABELEN    *****************************


//Labels voor deel 1:

label var milieu1 "I care about the environment a lot."

label var milieu2 "I always separate my garbage."

label var milieu3 "I donate to charities which are commited to the environment."

label var milieu4 "I only use my car when it is necessary. I try to do everything by bike."

label var milieu5 "I am aware of the effect that caring for the environment can benefit my wallet."

label var milieu6 "I want to devote myself for a better environment."

label var milieu7 "I am aware of my water consumption."

label var milieu8 "I switch off my lamps when I leave a room, because I know that it contributes to a
better environment."


label var influence1 "I often modify myself to other people."

label var influence2 "I have a strong opinion."

label var influence3 "I don't like it when people tell me what to do."

label var influence4 "I am easily convinced."


label var useless1 "I worry about the future."

label var useless2 "I prefer the city over the country side."
```

label var useless3 "I like to comfort people."

label var useless4 "I am an optimistic person."

label var useless5 "I can easily adapt to a new situation."

label var english "I master the English language."

//Labels voor deel 2:

label var shower_everyday "(1=pos; 0=neg) Do you take a shower every day?"

label var shower_10min "(1=pos; 0=neg) On the average, do you shower longer than ten minutes?"

label var shower_turnoff "(1=pos; 0=neg) Do you turn off the shower when you are using shampoo or soap?"

label var shower_rain "(1=pos; 0=neg) Do you own a rain shower?"

label var shower_toolong "(1=pos; 0=neg) Do you usually stay in the shower for a longer time than is actually needed?"

//Labels voor deel 3:

label var like1 "dislike-like"

label var like2 "awful-nice"

label var like3 "unkind-kind"

label var like4 "unpleasant-pleasant"

label var like5 "unfriendly-friendly"

label var animacy1 "artificial-lifelike"

label var animacy2 "monotone-lively"

label var animacy3 "mechanical-organic"

label var animacy4 "apathetic-sympathetic"

label var convince1 "I'm willing to change my behavior after hearing the comments of Will."

label var convince2 "I don't think that the comments of Will can help me with adjusting my showering habits."

label var convince3 "Will has made me think about my water consumption."

label var convince4 "The comments of Will did not convince me to change my habits."

label var open_to_adjust "I'm open to adjust my showering habits."

label var distracted "I was distracted when filling in this questionnaire."

label var emotion "1=With emotion 0=Without emotion"

label var version "1=Version1; 2=Version2"

//Variabelen gemaakt om te zien of een participant veel positieve of negatieve audiofragmenten heeft gehoord.

gen Shower = (shower_everyday + shower_10min + shower_turnoff + shower_rain + shower_toolong) / 5

label var Shower "The higher this value, the more positive fragments heard."

//***************************ONREGELMATIGHEDEN*******************************

//Nr 13 en 14 blijkt dezelfde persoon te zijn, deze heeft de enquete misschien dubbel ingeleverd.

drop if _n == 13

//Nr 15 en 16 blijkt ook dezelfde persoon te zijn.

drop if _n == 15

//Nr 64 en 65 zijn blijkt ook dezelfde persoon te zijn.

drop if _n == 64

//Participant 27 geeft aan dat hij/zij terug is gegaan om een vraag te veranderen, maar later zegt deze participant dat de leeftijd verkeerd was ingevuld.

//Goback_tochange veranderen we dus in 0 omdat het geen belangrijke verandering is geweest.

replace goback_tochange = 0 if _n == 27

//Nr 51 geeft aan dat hij alle vragen over Will niet begreep, dus die gooien we weg.

drop if _n == 51

//We gooien mensen weg die aangeven totaal geen Engels te kunnen of die aangeven volledig afgeleid te zijn geweest tijdens het invullen van de enquete.

drop if english == -2

drop if distracted == 2

//*******************CRONBACH'S ALPHA *************************************

alpha animacy*, item gen(Animacy)

alpha like*, item gen(Like)

alpha milieu*, item gen(Milieu)

alpha influence*, item gen(Influence) //Deze schaal is erg slecht. We moeten dus voorzichtig met deze variabelen omgaan.

alpha convince*, item gen(Convince)

//*******************ANALYSES EN GRAFIEKEN*******************************
//ANIMACY

sktest Animacy if emotion == 0

sktest Animacy if emotion == 1

swilk Animacy if emotion == 0

swilk Animacy if emotion == 1

//Alle vier de testen supporten normality.

robvar Animacy, by(emotion) //Alle p waardes zijn onder .05, dus gelijke variatie is verworpen.

kwallis Animacy, by(emotion) //Het verschil is significant.

*graph bar Animacy, by(emotion) blabel(total) //De richting is wat we hadden verwacht

//Testen of animacy afhangt van hoeveel positieve of negatieve filmpjes je hebt gehoord.

kwallis Animacy if Shower > .7, by(emotion)

kwallis Animacy if Shower < .3, by(emotion)

```
//LIKE

sktest Like if emotion == 0

sktest Like if emotion == 1

swilk Like if emotion == 0

swilk Like if emotion == 1

//Zonder emotie is er geen normale verdeling, met emotie is er wel normale verdeling

robvar Like, by(emotion) //Alle p waardeb zijn boven .05, dus er is een gelijke variatie

kwallis Like, by(emotion) //Het verschil is niet significant.

*graph bar Like, by(emotion) blabel(total) //De richting is wat we hadden verwacht

kwallis Like if Shower > .7, by(emotion)

kwallis Like if Shower < .3, by(emotion)


//CONVINCE

sktest Convince if emotion == 0

sktest Convince if emotion == 1

swilk Convince if emotion == 0

swilk Convince if emotion == 1

//sktest reject de normale verdeling, swilk support de normale verdeling dus we kiezen voor swilk

robvar Convince, by(emotion) //Alle p waarden zijn boven .05, dus er is een gelijke variatie


//We mogen een ANOVA doen omdat alle assumpties goed zijn

anova Convince emotion //p-waardes zijn boven .05 dus er is geen significant effect van emotie op
overtuiging

effectsize emotion


//Control for open_to_adjust

anova Convince emotion if open_to_adjust > -1
```

effectsize emotion

*graph bar Convince, by(emotion) blabel(total) //Klein verschil en niet significant dus er is geen verschil voor met of zonder emotie.

//Control for the kind of fragments that were heard.

anova Convince emotion if Shower > .7

effectsize emotion

anova Convince emotion if Shower < .3

effectsize emotion