Richtlijn wegen voor GGD-epidemiologen, uitbreiding naar “R”

Zie hier een uitbreiding van de “Richtlijn wegen voor GGD epidemiologen” [1]. Deze uitbreiding bespreekt de toepassing van de richtlijn met het statistische programma R [2]. De uitbreiding loopt parallel vanaf bladzijde 23, hoofdstuk 6, van de oorspronkelijke richtlijn. De “boxen” in deze uitbreiding komen overeen met de boxen in de richtlijn. De inhoud van de boxen hieronder kan je in de “R-console” plakken, als het goed is werkt dat direct. Je moet wel de rode tekst veranderen naar jouw omstandigheden. Iedere box staat op zich, dus je hoeft de volgorde van deze tekst niet te volgen. De data die voor deze uitbreiding is gebruikt kan je van het internet halen. Wil je dit met je eigen data doen dan moet je wel beginnen bij box 0.

Box 0 vind je niet in de richtlijn, deze box gaat over hoe je de monitor data inleest in R, reduceert, zodat je iets hanteerbaars overhoudt, en de data bewaart voor toekomstig gebruik.

In ‘R’ moet je kiezen of je de numerieke waarden of de “value lables” van de variabelen inleest. Meestal als je de waarden gebruikt is de variabele kwantitatief, je kunt er makkelijk berekeningen op uitvoeren, als de value labels gebruikt worden definieert een variabele groepen, een “ factor” variabele. Om het een beetje eenvoudig te houden lees ik de data twee keer in twee verschillende bestanden.

Een ander probleem is het opslaan van de file als csv. Hieronder is dat gedaan volgens de Europese methode, scheidingsteken is een “;” decimaal is een “,” . Werkt met een Nederlands talige Excel maar niet met een Engelstalige. Dan is het scheidingsteken een “,” en de decimaal een “.”.

Tenslotte, we gaan er van uit dat je de laatste versie van R hebt geïnstalleerd. Het instaleren van pakketten (install.packages) gaat soms gemakkelijker in “R-studio”.

Box 0. Voorbereiden: Inlezen SPSS .sav file, reduceren en bewaren.

setwd("C:/joudirectory") #invullen, let op, staan de schuine streepjes goed?

install.packages("foreign") #alleen doen als je het niet al eerder hebt gedaan

library(foreign)

FactorData<-read.spss("jou\_SPSS\_bestand.sav", max.value.labels=100, add.undeclared.levels="append", use.missings=TRUE) # jou\_SPSS\_bestand bevind zich in joudirectory deze regel leest de value labels in

NumberData<-read.spss("jou\_SPSS\_bestand.sav", max.value.labels=0, add.undeclared.levels="append", use.missings=TRUE, use.value.labels= FALSE)

table(FactorData$KLGGB201, useNA="always")

table(NumberData$KLGGB201, useNA="always")

is.factor(FactorData$KLGGB201)

is.factor(NumberData$KLGGB201)

is.numeric(NumberData$KLGGB201) #zie het verschil

names(NumberData) #en we hebben heel veel variabelen

#maak een hanteerbaar bestand genaamd “nd”

#waarbij we de factor en numerieke variabelen handig combineren

#met de variabele namen van GM 2016:

#eerst voor de numerieke variabelen:

#let op: R is hoofdlettergevoelig bij variabele namen

nd<-cbind(NumberData$AGGWS201,NumberData$ewCBSGGD, NumberData$ewGGD, NumberData$Gemeentecode)

nd<-as.data.frame(nd,stringsAsFactors=FALSE)

colnames(nd) <- c("AGGWS201","ewCBSGGD", "ewGGD", "Gemeentecode")

#vervolgens voegen we factor variabelen aan “nd” toe:

nd$generatie<-FactorData$generatie #generatie migrant

nd$Etniciteit<-FactorData$Etniciteit #herkomst persoon

nd$geslacht<-FactorData$geslacht #geslacht

nd$Lftklassen <-FactorData$Lftklassen #5-jrs leeftijdcat

nd$KLGGB201<-FactorData$KLGGB201 #ervaren gezondheid 5 cat

nd$PrimaireEenheid<-as.factor(FactorData$PrimaireEenheid) #sampling strata vlg CBS

nd$Gemeentenamen<-FactorData$Gemeentecode

summary(nd) #check, moet je vaak doen, summary vertelt altijd wel wat

rm(FactorData) #verwijder FactorData uit het geheugen.

rm(NumberData) #verwijder NumberData uit het geheugen.

write.table(nd, file="GGDenqueteData.csv", sep=";",na="NA", dec=",", row.names=FALSE,col.names=TRUE) #bewaar nd als .csv file in joudirectory

#wat je verder nog kan doen:

#table(nd$Lftklassen, useNA="always")

#nd$Lftklassen[nd$Lftklassen=="25-29 jaar"] <- NA #maak "25-29 jaar" missend

#table(nd$Lftklassen, useNA="always")

#levels(nd$Lftklassen) <- c(levels(nd$Lftklassen),"999") #maak een nieuwe level

#nd$Lftklassen[is.na(nd$Lftklassen)] <- "999" #geef de missende waarden de valid waarde 999

#table(nd$Lftklassen, useNA="always")

#nd$Lftklassen[nd$KLGGB201=="Gaat wel"] <- NA #maak Lftklassen missend als KLGGB201=="Gaat wel"

#table(nd$Lftklassen, useNA="always")

Hieronder gaan het verder met “box 1”. Deze bevindt zich in de richtlijn op bladzijde 25. Er zit een foutje op de derde regel van de oorspronkelijke tekst, er staat “and”, moet “or”, in R “|”, zijn. Men kan niet eerste AND tweede generatie allochtoon tegelijk zijn. Je kunt in deze box je eigen data uitlezen uit joudirectory, of data van het internet gebruiken. De uitkomsten voor de data van het internet staat in groen in de box.

Box 1. Syntax voor indeling leeftijd in andere categorieën, BMI in categorieën

setwd("C:/joudirectory") #deze regel moet in ieder geval, let op slashes

#kies, data uit joudirectory:

data <- read.csv(file="GGDenqueteData.csv", header=TRUE, sep=";",na="NA", dec=",")

#of kies data van het internet:

data <- read.csv(file="https://www.quantitativeskills.com/ggd/GGDenqueteData.csv", header=TRUE, sep=";",na="NA")

dim(data) # [1] 3654 12

summary(data) #check

#maak nieuwe leeftijdklassen

data$leeftijd<-'>64 jaar'

data$leeftijd[data$Lftklassen=='19-24 jaar'|data$Lftklassen=='25-29 jaar'|data$Lftklassen=='30-34 jaar']<-'19-34 jaar'

data$leeftijd[data$Lftklassen=='35-39 jaar'|data$Lftklassen=='40-44 jaar'|data$Lftklassen=='45-49 jaar']<-'35-49 jaar'

data$leeftijd[data$Lftklassen=='50-54 jaar'|data$Lftklassen=='55-59 jaar'|data$Lftklassen=='60-64 jaar']<-'50-64 jaar'

data$leeftijd[is.na(data$Lftklassen)]<-NA #maak missende waardes

table(data$leeftijd, useNA="always") #check

#>64 jaar 19-34 jaar 35-49 jaar 50-64 jaar <NA>

# 655 928 998 1073 0

table(data$leeftijd, data$Lftklassen, useNA="always") #check

#BMI categoriaal

data$BMIcat<-NA

data$BMIcat[data$AGGWS201>5]<-'te laag gewicht'

data$BMIcat[data$AGGWS201>19.9999]<-'normaal gewicht'

data$BMIcat[data$AGGWS201>24.9999]<-'overgewicht'

data$BMIcat[data$AGGWS201>29.9999]<-'obese'

write.table(data, file="GGDenqueteData.csv", sep=";",na="NA", dec=",", row.names=FALSE,col.names=TRUE) #bewaar data als .csv file in joudirectory

Box 3 is niet meer zo interessant omdat de weegfactoren al in het databestand dat van het CBS afkomt zitten. “ewGGD” voor gegevens die door de GGD zijn verzameld, “ewCBSGGD” voor gegevens verzameld door de GGD én het CBS. Voor de volledigheid toch maar een voorbeeld van het zelf aanmaken van gewichten.

Box 3. Syntax voor toevoegen van weegfactoren in databestand

setwd("C:/joudirectory") #deze regel moet in ieder geval, let op slashes

#kies, data uit joudirectory:

data <- read.csv(file="GGDenqueteData.csv", header=TRUE, sep=";",na="NA", dec=",")

#of kies data van het internet:

data <- read.csv(file="https://www.quantitativeskills.com/ggd/GGDenqueteData.csv", header=TRUE, sep=";",na="NA")

data$MIWFV201<-NA

data$MIWFV201[data$leeftijd=='19-34 jaar'&data$geslacht=='Man'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-1

data$MIWFV201[data$leeftijd=='35-49 jaar'&data$geslacht=='Man'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-2

data$MIWFV201[data$leeftijd=='50-64 jaar'&data$geslacht=='Man'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-3

data$MIWFV201[data$leeftijd=='>64 jaar'&data$geslacht=='Man'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-4

data$MIWFV201[data$leeftijd=='19-34 jaar'&data$geslacht=='Vrouw'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-5

data$MIWFV201[data$leeftijd=='35-49 jaar'&data$geslacht=='Vrouw'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-6

data$MIWFV201[data$leeftijd=='50-64 jaar'&data$geslacht=='Vrouw'

&data$Gemeentecode==363&data$Etniciteit=='Autochtoon']<-7

data$MIWFV201[data$leeftijd=='50-64 jaar'&data$geslacht=='Vrouw'

&data$Gemeentecode==363&data$Etniciteit!='Autochtoon']<-8

#etc etc

table(data$MIWFV201, useNA="always") #check

#eventueel kan je het bestand met weegfactoren bewaren:

write.table(data, file="GGDenqueteData.csv", sep=";",na="NA", dec=",", row.names=FALSE,col.names=TRUE) #bewaar data als .csv file in joudirectory

#data$Etniciteit!='Autochtoon': de ! staat voor niet ('Autochtoon')

#363 is Amsterdam, je had ook kunnen doen data$Gemeentenamen=="Amsterdam"

Box 4 gaat over het aanmaken van de strata. Hieronder worden zes strata gedefinieerd, geslacht naar leeftijd in 3 groepen. Ook deze box is minder belangrijk omdat het CBS de strata als de variabele “PrimaireEenheid” met de data mee levert.

Box 4. Syntax voor toevoegen van strata aan databestand

setwd("C:/joudirectory") #deze regel moet in ieder geval, let op slashes

#kies, data uit joudirectory:

data <- read.csv(file="GGDenqueteData.csv", header=TRUE, sep=";",na="NA", dec=",")

#of kies data van het internet:

data <- read.csv(file="https://www.quantitativeskills.com/ggd/GGDenqueteData.csv", header=TRUE, sep=";",na="NA", dec=",")

data$STRATA<-NA

data$STRATA[data$leeftijd=='19-34 jaar'&data$geslacht=='Man'&data$Gemeentecode==363]<-1

data$STRATA[data$leeftijd=='35-49 jaar'&data$geslacht=='Man'&data$Gemeentecode==363]<-2

data$STRATA[data$leeftijd=='50-64 jaar'&data$geslacht=='Man'&data$Gemeentecode==363]<-3

data$STRATA[data$leeftijd=='>64 jaar'&data$geslacht=='Man'&data$Gemeentecode==363]<-4

data$STRATA[data$leeftijd=='19-34 jaar'&data$geslacht=='Vrouw'&data$Gemeentecode==363]<-5

data$STRATA[data$leeftijd=='35-49 jaar'&data$geslacht=='Vrouw'&data$Gemeentecode==363]<-6

data$STRATA[data$leeftijd=='50-64 jaar'&data$geslacht=='Vrouw'&data$Gemeentecode==363]<-7

data$STRATA[data$leeftijd=='>64 jaar'&data$geslacht=='Vrouw'&data$Gemeentecode==363]<-8

data$STRATA[is.na(data$leeftijd)]<-NA

data$STRATA[is.na(data$geslacht)]<-NA

data$STRATA[is.na(data$Gemeentecode)]<-NA

table(data$STRATA, useNA="always") #check

# 1 2 3 4 5 6 7 8 <NA>

# 387 458 514 317 541 540 559 338 0

#eventueel kan je het bestand met STRATA bewaren:

write.table(data, file="GGDenqueteData.csv", sep=";",na="NA", dec=",", row.names=FALSE,col.names=TRUE) #bewaar data als .csv file in joudirectory

Box 5 en verder vind je niet in de richtlijn. In SPSS maak je met een aantal menu’s een “planfile” dat je bewaart in een directory en dat je aanroept als je een “complex samples” analyse doet. De richtlijn gaat er van bladzijde 31 t/m 33 op in. In R-survey maak je een design object dat je bewaart in de actieve R-console. Je kunt het ook opslaan als .rds bestand en later opnieuw gebruiken.

In box 5 worden gewichten gebruikt die met het databestand zijn meegeleverd. Het databestand wordt dan gestratificeerd met de variabele “primaireeenheid”. Het uitgangspunt is dat de steekproeftrekking, voorafgaande aan data verzameling, plaats heeft gevonden volgens “primaireeenheid”. Wegen geeft een design factor groter dan één, een grotere variantie en bredere betrouwbaarheidsinterval, stratificeren geeft een designfactor kleiner dan één, meer precieze schattingen.

Box 5. Syntax voor aanmaken design object

#bepaal werkdirectory en lees data in

table(data$ewCBSGGD,useNA="always") #het zijn grote gewichten want alles >1, de kleinste is 6.1 en de grootste 98.8. De inclusie waarden (kans om in de steekproef te komen) zijn van 1/6.1= 0.1639 tot 1/98.8= 0.01012.

sum(data$ewCBSGGD) #de steekproef is getrokken uit 99602.5 personen. De steekproeffractie is dan 3654/99602.5=0.0367, ongeveer 3,7% van de bevolking doet mee aan het onderzoek.

sum(data$ewGGD) #zelfde bevolking, GGD steekproef

install.packages("survey") #alleen doen als je het niet al eerder hebt gedaan

library(survey)

#required packages zijn grid,lattice en Matrix. Krijg je een foutmeldingen instaleer ze dan alsnog, bijvoorbeeld:

#install.packages("Matrix")

table(data$KLGGB201, useNA="always") #corrigeer de volgorde van de categorieën

data$KLGGB201=factor(data$KLGGB201,levels(data$KLGGB201)[c(4,2,1,3,5)])

table(data$KLGGB201, useNA="always") #check

data$BMIcat=factor(data$BMIcat,levels(data$BMIcat)[c(4,1,3,2)]) #hier ook

data$leeftijd=factor(data$leeftijd,levels(data$leeftijd)[c(2,3,4,1)]) #en hier

#vervolgens, bij beide weegfactoren (ewGGD en ewCBSGGD) zelfde exercitie:

MijnDesign <- svydesign(ids=~1,data=data, strata=~PrimaireEenheid, weights=~ewCBSGGD)

#de design heeft geen clusters want ids=~1

summary(MijnDesign) #check, zoek de inclusie waarden

saveRDS(MijnDesign,"MijnDesignFile.rds") #bewaar het design in joudirectory, de volgorde van de categorieën in een factor variabele wordt bewaard

MijnDesignGGD <- svydesign(ids=~1,data=data, strata=~PrimaireEenheid, weights=~ewGGD) #design alleen voor de GGD, de CBS respondenten hebben gewicht=0

summary(MijnDesignGGD) #check, zoek de inclusie waarden, de inclusie criterium voor een gewicht van 0 is 1/0=Inf, de mean is dan ook Inf

saveRDS(MijnDesignGGD,"MijnDesignGGDFile.rds") #bewaar het design in joudirectory

We analyseren de data met de gewogen prevalenties en gecorrigeerde betrouwbaarheidsintervallen met R package “survey”. Survey is geschreven op basis van hetzelfde boek als SPSS complex samples, dus veel verschil in het resultaat zal er niet zijn. Survey past bestaande R-procedures aan. De syntax komt overeen met standaard R procedures en je hebt de beschikking over een groot aantal analyses. Box 6 gaat over de standaard epidemiologische schattingen voor gemiddelden

Box 6. Syntax voor populatie schattingen, gemiddelden

#bepaal werkdirectory en lees data in

mean(data$AGGWS201,na.rm=TRUE)

#[1] 25 .80135 de gemiddelde BMI van de respondenten is 25.8

weighted.mean(data$AGGWS201,data$ewGGD,na.rm=TRUE)

#[1] 25.76612 het geschatte gemiddelde in de populatie is 25.8, hier heb je “survey” niet voor nodig

confint(lm(AGGWS201~1, data=data)) #het berekenen van een betrouwbaarheids interval is in R dus niet echt heel simpel

#(Intercept) 25.65033 25.95236 is de CI voor de respondenten

abs(25.65033-25.95236)

# 0.30203 de breedte van de interval

**library(survey) #maak survey actief**

**MijnDesign <- readRDS("MijnDesignFile.rds") #laad het design, CBS+GGD data**

summary(MijnDesign) #check

a<-svymean(~AGGWS201,design=MijnDesign,na.rm=TRUE,deff=TRUE) #met behulp van het pakket Survey

a

# mean SE DEff

# AGGWS201 25.75479 0.08631 1.2912, mean als boven

confint(a)

# 2.5 % 97.5 %

#AGGWS201 25.58562 25.92395 is de geschatte CI voor de bevolking

abs(25.58562-25.92395)

# 0.33833, is dus breder dan hierboven, het “design effect” [3]

#BMI naar geslacht:

lm(AGGWS201~geslacht-1, data=data) #a=BMI respondenten (ongewogen) m-v

b<- svyby(~AGGWS201,~geslacht,MijnDesign,svymean, na.rm=TRUE)

b #b=geschatte BMI in bevolking (gewogen) m-v

confint(b) #CI BMI in bevolking m-v

svyttest(AGGWS201~geslacht,MijnDesign) # t= -1.872, het verschil tussen mannen en vrouwen in BMI is niet significant, p-value= 0.06129

#BMI naar gerapporteerde gezondheid:

nul<-svyglm(AGGWS201~1, MijnDesign, family=stats::gaussian())

een<-svyglm(AGGWS201~KLGGB201, MijnDesign, family=stats::gaussian())

anova(nul,een) #effect is statistisch significant met 2logLR=127.3; p=<2.22e-16

zonderintercept<-svyglm(AGGWS201~KLGGB201-1,MijnDesign,family=stats::gaussian())

summary(zonderintercept)

confint(zonderintercept)

#conclusie: hoe slechter men zich voelt hoe hoger de BMI, of visa versa

Box 7. Syntax voor populatie schattingen, tabellen

#bepaal werkdirectory en lees data in, maak “survey” actief, laad MijnDesign

table(data$KLGGB201)

#545 respondenten ervaren een zeer goede gezondheid

prop.table(table(data$KLGGB201)) #14.7% van de respondenten ervaart een zeer geode gezondheid, corrigeer volgorde categorieën volgens box 5

a<-svytable(~KLGGB201,design=MijnDesign)

a #naar schatting 13705.3 van de populatie ervaren een zeer goede gezondheid

prop.table(a) #dat is dan 14.9% van de populatie

a<-svymean( ~ KLGGB201, MijnDesign, na.rm=TRUE) #zo kan het ook

a

confint(a) #is mean±1,96\*se (mean gewogen, se design gecorrigeerd)

svyciprop(~KLGGB201=='Zeer goed', MijnDesign, method="lo")

svyciprop(~KLGGB201=='Goed', MijnDesign, method="lo") #logistische schatting is beter vooral nabij 0 en 1, maar een stuk meer werk, werkt niet altijd.

a<-svytable(~KLGGB201+geslacht,design=MijnDesign)

a #schatting aantallen in de bevolking naar groep

prop.table(a) #cell percentages

prop.table(a,1) #row percentages

prop.table(a,2) #17,5% van de mannen ervaart hun gezondheid als zeer goed, bij vrouwen is dat 12,5% (vrouwen lijken wat minder uitgesproken in hun ervaring)

svychisq(~KLGGB201+geslacht,design= MijnDesign,statistic="Chisq")

# X-squared= 20.402, het verschil tussen mannen en vrouwen in ervaren gezondheid is significant

#Hieronder min of meer hetzelfde maar dan voor de ongewogen-respondenten data:

a<-xtabs(~KLGGB201+geslacht, data=data)

a #de volgorde van de categorieën is onjuist in de data, zie Box 5 voor correctie

prop.table(a,2)

summary(a)

#Chisq=23.383 is hoger dan de bovenstaande 20.402, dat zou je verwachten

#tenslotte, ervaren gezondheid in twee categorieën

data$KLGGA208<-NA

data$KLGGA208[data$KLGGB201=='Zeer goed'|data$KLGGB201=='Goed']<- 'goed tot zeer goed'

data$KLGGA208[data$KLGGB201=='Niet goed, niet slecht'|data$KLGGB201=='Slecht'|data$KLGGB201=='Zeer slecht']<-'gaat wel, slecht tot zeer slecht'

table(data$KLGGA208,useNA="always") #respondenten, ongewogen

MijnDesign <- svydesign(ids=~1,data=data, strata=~ PrimaireEenheid, weights=~ewGGD) #helaas, als je een nieuwe variabele aanmaakt moet je een nieuwe design object aanmaken, inclusief corrigeren van de volgorde van de categorieën

svytable(~KLGGA208,design=MijnDesign) #schatting aantal in bevolking, gewogen

svymean(~KLGGA208, MijnDesign, na.rm=TRUE) #percentage bevolking

svyciprop(~KLGGA208=='goed tot zeer goed', MijnDesign, method="lo")

2.5% 97.5%

KLGGA208 == "goed tot zeer goed" 0.730 0.713 0.75

Box 8. Syntax voor tabellen naar .csv (excel)

#bepaal werkdirectory en lees data in, maak “survey” actief, laad MijnDesign

a<-svytable(~KLGGB201+geslacht,design=MijnDesign)

a<-addmargins(a,2) #+rijtotalen

b<-prop.table(a,2)

b

#schrijf tabel naar .csv

write("KLGGB201 x geslacht, gewogen aantallen:" ,file="output.csv") #de titel (waarbij de default is append=FALSE, je maakt een nieuw bestand)

colnames(a)[1]<-paste(";",colnames(a)[1],sep="") #zet colnames op de juiste plek

write.table(a, file="output.csv", sep=";",na="NA", dec=",", row.names=TRUE,col.names=TRUE,append=TRUE,quote=FALSE) #de tabel, aantallen

write("\nKLGGB201 x geslacht, gewogen proporties:" ,file="output.csv",append=TRUE)

colnames(b)[1]<-paste(";",colnames(b)[1],sep="")

write.table(b, file="output.csv", sep=";",na="NA", dec=",", row.names=TRUE,col.names=TRUE,append=TRUE,quote=FALSE) #de proporties

a<- addmargins(svytable(~BMIcat+geslacht,design=MijnDesign),2)

#voeg een volgende tabel aan de .csv

write("\nBMIcat x geslacht, aantallen:" ,file="output.csv",append=TRUE) #de titel van de tabel

colnames(a)[1]<-paste(";",colnames(a)[1],sep="")

write.table(a, file="output.csv", sep=";",na="NA", dec=",", row.names=TRUE,col.names=TRUE,append=TRUE,quote=FALSE) #de tabel

#doe heel veel tabellen:

Uitkomstvariabelen<-c("KLGGB201","BMIcat") #voeg hier toe als je dat wil

Gesplitstnaar<- c("geslacht","Etniciteit","leeftijd") #voeg hier toe

for (index in 1:length(Uitkomstvariabelen))

for (jndex in 1:length(Gesplitstnaar))

{a<- addmargins(svytable(~get(Uitkomstvariabelen[index])+get(Gesplitstnaar[jndex]),design=MijnDesign),2)

b<-prop.table(a,2)

write(paste("\n",Uitkomstvariabelen[index],"x",Gesplitstnaar[jndex],"aantallen:"),file="output.csv",append=TRUE) #de titel

colnames(a)[1]<-paste(";",colnames(a)[1],sep="")

write.table(a, file="output.csv", sep=";",na="NA", dec=",", row.names=TRUE,col.names=TRUE,append=TRUE,quote=FALSE) #de tabel

write(paste("\n",Uitkomstvariabelen[index],"x",Gesplitstnaar[jndex],"proporties:"),file="output.csv",append=TRUE) #de titel

colnames(b)[1]<-paste(";",colnames(b)[1],sep="")

write.table(b, file="output.csv", sep=";",na="NA", dec=",", row.names=TRUE,col.names=TRUE,append=TRUE,quote=FALSE) #de tabel

} #einde twee for loops

#paste betekend plak de stukjes aan elkaar

#"\n" betekend nieuwe regel

#de volgorde van de categorieën van een variabele bij “survey” zit in het design object

Box 9. Syntax voor mooie tabellen naar .docx (word)

#bepaal werkdirectory en lees data in, maak “survey” actief, laad MijnDesign

#install.packages("flextable")

library(flextable)

#install.packages("tibble")

library(tibble)

#install.packages("dplyr")

library(dplyr)

#install.packages("officer")

library(officer)

Uitkomstvariabelen<-c("KLGGB201","BMIcat") #voeg hier toe als je dat wil

Gesplitstnaar<- c("geslacht","Etniciteit","leeftijd") #voeg hier toe

doc<-read\_docx() #doc is een word document object

doc<-doc %>% body\_add\_par(paste("Uitdraai gemaakt op ",date()),style="heading 1")

doc<-doc %>% body\_add\_par("") # blank paragraph

for (index in 1:length(Uitkomstvariabelen))

for (jndex in 1:length(Gesplitstnaar))

{a<-svytable(~get(Uitkomstvariabelen[index])+get(Gesplitstnaar[jndex]),design=MijnDesign)

b<-prop.table(a,2)

tabel<-as.data.frame.matrix(a)

fl<-flextable(tabel %>% rownames\_to\_column(Uitkomstvariabelen[index]))

fl<-add\_header\_lines(fl,paste(Uitkomstvariabelen[index],"x",Gesplitstnaar[jndex],"aantallen:"))

doc<-body\_add\_flextable(doc,value=fl)

doc<-doc %>% body\_add\_par("") # blank paragraph

tabel<-as.data.frame.matrix(b)

fl<-flextable(tabel %>% rownames\_to\_column(Uitkomstvariabelen[index]))

fl<-add\_header\_lines(fl,paste(Uitkomstvariabelen[index],"x",Gesplitstnaar[jndex],"proporties:"))

doc<-body\_add\_flextable(doc,value=fl)

doc<-doc %>% body\_add\_par("") # blank paragraph

} #einde twee for loops

print(doc,target="output.docx")

fl #probeer deze ook maar eens

#de gebruiksaanwijzing van “officer” gaat over de word format

#de gebruiksaanwijzing van “flextable” gaat over de tabel format

1] Carolien van den Brink, Daan Uitenbroek, Mandy van den Brink, Elly de Boer, Klaasjan Hajema, Maarten Schipper. Richtlijn wegen voor GGD-epidemiologen. (2009) Utrecht: GGD-Nederland. <https://www.ggdghorkennisnet.nl/thema/kwaliteitsmanagement/publicaties/publicatie/265-richtlijn-wegen-voor-ggd-epidemiologen>

2] Diverse auteurs. The R Project for Statistical Computing. (20xx) <https://www.r-project.org/>

3] Daan Uitenbroek. Design, wegen en het designeffect in GGD gezondheidsenquêtes. (2009): TSG 87.2: 64-68.

Auteur: Daan Uitenbroek

e-mail: [duitenbroek@ggd.amsterdam.nl](mailto:duitenbroek@ggd.amsterdam.nl)

email: [daan.uitenbroek@quantitativeskills.com](mailto:daan.uitenbroek@quantitativeskills.com)

Kijk ook even op: <https://www.quantitativeskills.com/ggd/wegen.htm>

Met dank aan:

Henriette Dijkshoorn