

FRS workshop

September 2022

Contents

1	Benefit Comparison – Year: 2020-21	1
1.1	Create Both1	1
1.2	Benefit frequencies	2
1.3	Distribution of Universal Credit, State Pension , Personal Independence Payment (Daily Living), Personal Independence Payment (PIP) Mobility	2
1.4	Histograms	3
1.5	Distribution of Universal Credit ex. £0	3
1.6	UC and tenure	4
2	Benefit Comparison – Year: 2019-20	5
2.1	Benefit frequencies	5
2.2	Distribution of Universal Credit, State Pension , Personal Independence Payment (Daily Living), Personal Independence Payment (PIP) Mobility	5
2.3	Histograms	6
2.4	Create Both2 – Distribution of Universal Credit ex. £0	7
2.5	UC and tenure	7
3	Statistically significant differences between 2019-20 and 2020-21	8
3.1	Create Combine_years	8
3.2	ANOVA UC amount by year	8
3.3	ANOVA State pension benefit amount by year	8
3.4	ANOVA PIP Daily Living	9
3.5	ANOVA PIP mobility	9
4	Wellbeing variables	9
4.1	Comparison between 2019-20 and 2020-21	9
4.2	Tables of wellbeing variable scores against other variables	15
5	Rapid analysis	16
5.1	FRS households by tenure between 2018-19 and 2020-21	16
5.2	FRS households by Council Tax Band between 2018-19 and 2020-21	17
5.3	FRS adult respondents by highest level of qualification between 2018-19 and 2020-21	18

1 Benefit Comparison – Year: 2020-21

1.1 Create Both1

Create a new dataset called ‘both1’ from merging the FRS adult dataset and FRS benefits dataset by their common variables (sernum, benunit and person)

In this case, we choose to keep the numerical version of categorical variables

```
library(foreign)    ### Library necessary to open SPSS files
library(dplyr)      ### Advanced data manipulation functions
```

```
library(Hmisc)      ### Extra statistical functions

setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/UKDA-8948-spss/spss/spss25/")
a21<-read.spss("adult.sav",to.data.frame = T,
use.value.labels = F )>%
  select(SERNUM,BENUNIT,PERSON,GROSS4)
b21<-read.spss("benefits.sav",to.data.frame = T,
use.value.labels = F )>%
  select(SERNUM,BENUNIT,PERSON, BENEFIT,BENAMT)
both1<-merge(a21,b21,by=c("SERNUM","BENUNIT","PERSON"),all.x=F,all.y=F)

names(both1)<-tolower(names(both1)) ### Changes names to lowercase
```

1.2 Benefit frequencies

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each benefit

```
tmptab<-xtabs(gross4~benefit,both1,drop.unused.levels=T)
cbind(
"Freq"= tmptab,
"Pct"=round(100*prop.table(tmptab),1),
"Cum. freq."=cumsum(tmptab),
"Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

1.3 Distribution of Universal Credit, State Pension , Personal Independence Payment (Daily Living), Personal Independence Payment (PIP) Mobility

Examines the distribution of grossed Universal Credit benefit, grossed State Pension benefit amounts, grossed Personal Independence Payment (Daily Living benefit), grossed Personal Independence Payment (PIP) Mobility benefit amounts, including an assessment of normality and discovery of outliers

xtabs() produces the tables; *prop.table()* computes the proportions, *round(...,1)* rounds the result to one decimal. R is programmed with parsimony in mind: we only get the ourput we asked for. This is why we need to specify each indicator we are interested in. We can then gather them all in a single table with *cbind()*. We select the subset of respondents on UC with the *filter()* command. The pipe *%>%* operators allow us to combine several command one after the other.

```
#knitr::kable(
round(t(both1%>%
  filter(benefit==95 | benefit==5 | benefit==96 |
    benefit==97)%>%group_by(benefit)%>%
  summarise(
"Observations"=length(benamt),
"Mean"=wtd.mean(benamt,gross4),
"Median"=wtd.quantile(benamt,gross4,probs=.5),
"Std Deviation"=sqrt(wtd.var(benamt,gross4)),
"Variance"=wtd.var(benamt,gross4),
"Mode"=as.numeric(names(sort(-table(benamt)))[1]),
"Range"=max(benamt)-min(benamt),
"IQR"=wtd.quantile(benamt,gross4,probs=.75)-wtd.quantile(benamt,gross4,probs=.25),
"Skewness"=Weighted.Desc.Stat::w.skewness(benamt,gross4),
```

```
"Kurtosis"=Weighted.Desc.Stat::w.kurtosis(benamt,gross4)
)),2) #)
```

Weighted quantiles

```
#knitr::kable(
### Computing the results first, and storing them in tmp
tmp<-both1%>%filter(benefit==95| benefit==5 | benefit==96 | benefit==97) %>%
  group_by(benefit)%>%
  summarise(
"ben_nr"=wtd.quantile(benamt,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99))
) #)

### tmp is a long table with each set of nine quantiles
### added one after the other (36 rows).
### The code below makes it easier to read

data.frame(
  "Quantile"=c(.01,.05,.1,.25,.5,.75,.9,.95,.99),
  "UC"=      tmp[tmp$benefit==95,2],
  "Pension"= tmp[tmp$benefit==5,2],
  "PIP (DL)"=tmp[tmp$benefit==96,2],
  "PIP (M)"= tmp[tmp$benefit==97,2])
```

Extreme values

We have chosen to collate the results with `cbind()` in this case, but each part can be run separately if needed.

```
#knitr::kable(
round(
cbind(
  both1%>%filter(benefit==95 | benefit==5 | benefit==96 | benefit==97)%>%
    select(sernum,person,benamt,benefit)%>%
  group_by(benefit)%>%slice_min(order_by = benamt,n=5,with_ties = FALSE),
  both1%>%filter(benefit==95 | benefit==5 | benefit==96 | benefit==97)%>%
    select(sernum,person,benamt,benefit)%>%
  group_by(benefit)%>%slice_max(order_by = benamt,n=5,with_ties = FALSE)
)
,1)#)
```

1.4 Histograms

Universal credit

The line below creates the histogram.

```
ggplot(both1%>%
  filter(benefit==95 | benefit==5 | benefit==96 | benefit==97),
  aes(x=benamt))+ ### Base plot specification
geom_histogram(aes(y = (..count..)/sum(..count..)))+ ### Converts y axis to percent
scale_y_continuous(labels = scales::percent)+ ### Prints y axis as percent
facet_wrap(~benefit,scale="free") ### Each histogram has its own plot
```

1.5 Distribution of Universal Credit ex. £0

Note There is no need to create a new dataset, we can keep on working with `both1`

Examines the distribution of grossed Universal Credit benefit amounts excluding £0 observations, including an assessment of normality and discovery of outliers

In the syntax below we exclude cases with £0 benefits by adding *benamt>0* to the *filter()* command.

```
round(t(both1%>%filter(benefit==95 & benamt>0)%>%summarise(
  "Observations"=length(benamt),
  "Mean"=wtd.mean(benamt,gross4),
  "Median"=wtd.quantile(benamt,gross4,probs=.5),
  "Std Deviation"=sqrt(wtd.var(benamt,gross4)),
  "Variance"=wtd.var(benamt,gross4),
  "Mode"=as.numeric(names(sort(-table(benamt)))[1]),
  "Range"=max(benamt)-min(benamt),
  "IQR"=wtd.quantile(benamt,gross4,probs=.75)-wtd.quantile(benamt,gross4,probs=.25),
  "Skewness"=Weighted.Desc.Stat::w.skewness(benamt,gross4),
  "Kurtosis"=Weighted.Desc.Stat::w.kurtosis(benamt,gross4)
)),2)
```

Weighted quantiles

```
both1%>%filter(benefit==95 & benamt>0)%>%select(benamt,gross4)%>%summarise(
  "Quantiles"=
round(wtd.quantile(benamt,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
)
```

Extreme values

```
cbind(
  both1%>%filter(benefit==95 & benamt>0)%>%select(sernum,benunit,person,benamt)%>%
  slice_min(order_by = benamt,n=5,with_ties = FALSE),
  both1%>%filter(benefit==95 & benamt>0)%>%select(sernum,benunit,person,benamt)%>%
  slice_max(order_by = benamt,n=5,with_ties = FALSE)
)
```

Histogram of Universal Credit (without 0s)

The syntax below also select a subset of the data, with square brackets as an alternative to the *filter()* command used above. We also use an alternative command to create the histogram.

```
z<-hist(both1$benamt[both1$benefit==95 & both1$benamt>0],freq=F)

z$density <- z$counts/sum(z$counts) * 100 ### Converting y axis into percentages

plot(z, freq = FALSE,
  main= '% of amount of benefit received for those on PIP Mobility',
  xlab="Amount of benefit received last time (in £)",
  ylab="Percentage"
)
```

1.6 UC and tenure

Create *uc_tenure2*

Create a new dataset called 'uc_tenure2' from merging the 'uc_tenure1'(ie both1) dataset and FRS household dataset by their common variable (sernum)

```
setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/UKDA-8948-spss/spss/spss25" )
h21<-read.spss("househol.sav",to.data.frame = T,
use.value.labels = T,
```

```
add.undeclared.levels="append")%>%select(SERNUM,TENURE)
names(h21)<-tolower(names(h21))
uc_tenure2<-merge(both1,h21,by=c("sernum"),all.x=F,all.y=F)
levels(uc_tenure2$tenure)[2]<-"Bought with mortgage" ### Shorten a category, for nicer display
```

Frequency of Tenure

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each category for the variable 'tenure' for those in receipt of Universal Credit

```
tmptab<-xtabs(gross4~tenure,uc_tenure2,subset=benefit==95,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

2 Benefit Comparison – Year: 2019-20

Create a new dataset Both3 (we could re-use both1) from merging the FRS adult dataset and FRS benefits dataset by their common variables (sernum, benunit and person)

```
setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/UKDA-8802-spss/spss/spss25/")
a19<-read.spss("adult.sav",to.data.frame = T,
use.value.labels = F )%>%select(SERNUM,BENUNIT,PERSON,GROSS4)
b19<-read.spss("benefits.sav",to.data.frame = T,
use.value.labels = F)%>%select(SERNUM,BENUNIT,PERSON, BENEFIT,BENAMT)
both3<-merge(a19,b19,by=c("SERNUM","BENUNIT","PERSON"),all.x=F,all.y=F)

names(both3)<-tolower(names(both3))
```

2.1 Benefit frequencies

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each benefit

```
tmptab<-xtabs(gross4~benefit,both3,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

2.2 Distribution of Universal Credit, State Pension , Personal Independence Payment (Daily Living), Personal Independence Payment (PIP) Mobility

Examines the distribution of grossed Universal Credit benefit, State Pension benefit amounts, Personal Independence Payment (Daily Living benefit), Personal Independence Payment (PIP) Mobility benefit amounts, including an assessment of normality and discovery of outliers

```
round(t(both3%>%filter(benefit==95 | benefit==5 | benefit==96 | benefit==97)%>%
  group_by(benefit)%>%summarise(
    "Observations"=length(benamnt),
```

```

"Mean"=wtd.mean(benamnt,gross4),
"Median"=wtd.quantile(benamnt,gross4,probs=.5),
"Std Deviation"=sqrt(wtd.var(benamnt,gross4)),
"Variance"=wtd.var(benamnt,gross4),
"Mode"=as.numeric(names(sort(-table(benamnt)))[1]),
"Range"=max(benamnt)-min(benamnt),
"IQR"=wtd.quantile(benamnt,gross4,probs=.75)-wtd.quantile(benamnt,gross4,probs=.25),
"Skewness"=Weighted.Desc.Stat::w.skewness(benamnt,gross4),
"Kurtosis"=Weighted.Desc.Stat::w.kurtosis(benamnt,gross4)
)),2) #)

```

Weighted quantiles

```

#knitr::kable(
### Computing the results first, and storing them in tmp
tmp<-both3%>%filter(benefit==95 | benefit==5 | benefit==96 | benefit==97) %>%
  group_by(benefit)%>% summarise(
    "ben#"=wtd.quantile(benamnt,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99))
  ) #)

### tmp is a long table with each set of nine quantiles
### added one after the other (36 rows).
### The code below makes it easier to read

data.frame(
  "Quantile"=c(.01,.05,.1,.25,.5,.75,.9,.95,.99),
  "UC"=      tmp[tmp$benefit==95,2],
  "Pension"= tmp[tmp$benefit==5,2],
  "PIP (DL)"=tmp[tmp$benefit==96,2],
  "PIP (M)"= tmp[tmp$benefit==97,2])

```

Extreme values

We have chosen to collate the results with `cbind()` in this case, but each part can be run separately if needed.

```

#knitr::kable(
round(
cbind(
  both3%>%filter(benefit==95 | benefit==5 | benefit==96 | benefit==97)%>%
    select(sernum,person,benamnt,benefit)%>%
  group_by(benefit)%>%slice_min(order_by = benamnt,n=5,with_ties = FALSE),
  both3%>%filter(benefit==95 | benefit==5 | benefit==96 | benefit==97)%>%
    select(sernum,person,benamnt,benefit)%>%
  group_by(benefit)%>%slice_max(order_by = benamnt,n=5,with_ties = FALSE)
),1)#)

```

2.3 Histograms

Universal credit

```

ggplot(both1%>%
  filter(benefit==95 | benefit==5 | benefit==96 | benefit==97),
  aes(x=benamnt))+ ### Base plot specification
geom_histogram(aes(y = (.count.)/sum(.count.)))+ ### Converts y axis to percent
scale_y_continuous(labels = scales::percent)+ ### Prints y axis as percent

```

```
facet_wrap(~benefit,scale="free") ### Each histogram has its own plot
```

2.4 Create Both2 – Distribution of Universal Credit ex. £0

(Note There is no need for this step in R, we can keep on working with both3)

Examines the distribution of grossed Universal Credit benefit amounts excluding £0 observations, including an assessment of normality and discovery of outliers

```
round(t(both3%>%filter(benefit==95 & benamt>0)%>%summarise(
  "Observations"=length(benamt),
  "Mean"=wtd.mean(benamt,gross4),
  "Median"=wtd.quantile(benamt,gross4,probs=.5),
  "Std Deviation"=sqrt(wtd.var(benamt,gross4)),
  "Variance"=wtd.var(benamt,gross4),
  "Mode"=as.numeric(names(sort(-table(benamt)))[1]),
  "Range"=max(benamt)-min(benamt),
  "IQR"=wtd.quantile(benamt,gross4,probs=.75)-wtd.quantile(benamt,gross4,probs=.25),
  "Skewness"=Weighted.Desc.Stat::w.skewness(benamt,gross4),
  "Kurtosis"=Weighted.Desc.Stat::w.kurtosis(benamt,gross4)
)),2)
```

Weighted quantiles

```
both3%>%filter(benefit==95 & benamt>0)%>%select(benamt,gross4)%>%summarise(
  "Quantiles"=
  round(wtd.quantile(benamt,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
)
```

Extreme values

```
cbind(
  both3%>%filter(benefit==95 & benamt>0)%>%select(sernum,benunit,person,benamt)%>%
  slice_min(order_by = benamt,n=5,with_ties = FALSE),
  both3%>%filter(benefit==95 & benamt>0)%>%select(sernum,benunit,person,benamt)%>%
  slice_max(order_by = benamt,n=5,with_ties = FALSE)
)
```

** Histogram of Universal Credit (without 0s)**

```
z<-hist(both3$benamt[both3$benefit==95 & both3$benamt>0],freq=F)
```

```
z$density <- z$counts/sum(z$counts) * 100 ### Converting y axis into percentages
```

```
plot(z, freq = FALSE,
     main= '% of amount of benefit received for those on PIP Mobility',
     xlab="Amount of benefit received last time (in £)",
     ylab="Percentage"
)
```

2.5 UC and tenure

Create tenure4 Create a new dataset called 'uc_tenure4' from merging the 'uc_tenure1'(ie both3) dataset and FRS household dataset by their common variable (sernum)**

```
setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/UKDA-8948-spss/spss/spss25" )
h21<-read.spss("househol.sav",to.data.frame = T,
use.value.labels = T,
```

```
add.undeclared.levels="append")%>%select(SERNUM,TENURE)
names(h21)<-tolower(names(h21))
uc_tenure4<-merge(both3,h21,by=c("sernum"),all.x=F,all.y=F)
levels(uc_tenure4$tenure)[2]<-"Bought with mortgage"
```

Frequency of Tenure

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each category for the variable 'tenure' for those in receipt of Universal Credit

```
tmptab<-xtabs(gross4~tenure,uc_tenure4,subset=benefit==95,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

3 Statistically significant differences between 2019-20 and 2020-21

3.1 Create Combine_years

Create a new appended dataset called 'combine_years' that contains the 'ad_ben_1920' dataset and 'ad_ben_2021' dataset (ie Both1 and Both3) A new variable called 'year' is created that assigns '1920' to the case if the year is 2019-20 and '2021' if the year is 2020-21

```
combine_years<-rbind(
  cbind(both1,year=2021),
  cbind(both3,year=1920)
)
```

/Provides grossed analysis of variance and Kruskal-Wallis for Universal Credit benefit amounts by year/

3.2 ANOVA UC amount by year

```
summary(lm(benamt~as.factor(year),data=combine_years%>%filter(benefit==95),weights=gross4))

### We need to use the Survey package in order to access the weighted rank test functions
### (Wilcoxon/Kruskal-Wallis)
library(survey)

### We also need to declare the survey design, in this case we will only specify the
### grossing weights
design <- svydesign(ids = ~0, data=combine_years%>%filter(benefit==95),
  weights=combine_years%>%filter(benefit==95)%>%select(gross4))

svyranktest(benamt ~ as.factor(year), design = design)
```

3.3 ANOVA State pension benefit amount by year

/Provides grossed analysis of variance and Kruskal-Wallis Rank Sums for State Pension benefit amounts by year/

```
summary(lm(benamt~as.factor(year),data=combine_years%>%filter(benefit==5),weights=gross4))
```



```
design <- svydesign(ids = ~0, data=combine_years%>%filter(benefit==5),
weights=combine_years%>%filter(benefit==5)%>%select(gross4))

svyranktest(benamt ~ as.factor(year), design = design)
```

3.4 ANOVA PIP Daily Living

/Provides grossed analysis of variance and Kruskal-Wallis Rank Sums for Personal Independence Payment (PIP) Daily Living benefit amounts by year/

```
summary(lm(benamt~as.factor(year),data=combine_years%>%filter(benefit==96),weights=gross4))

design <- svydesign(ids = ~0, data=combine_years%>%filter(benefit==96),
weights=combine_years%>%filter(benefit==96)%>%select(gross4))

svyranktest(benamt ~ as.factor(year), design = design)
```

3.5 ANOVA PIP mobility

/Provides grossed analysis of variance and Kruskal-Wallis Rank Sums for Personal Independence Payment (PIP) Mobility benefit amounts by year/

```
summary(lm(benamt~as.factor(year),data=combine_years%>%filter(benefit==97),weights=gross4))

design <- svydesign(ids = ~0, data=combine_years%>%filter(benefit==97),
weights=combine_years%>%filter(benefit==97)%>%select(gross4))

svyranktest(benamt ~ as.factor(year), design = design)
```

4 Wellbeing variables

4.1 Comparison between 2019-20 and 2020-21

Creating a 'combine_years' dataset from the merged individual and households datasets from 2019-20 and 2020-21

```
setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/")
combine_years<-merge(
rbind(
cbind(read.spss("UKDA-8948-spss/spss/spss25/adult.sav",to.data.frame = T,
use.value.labels = F)%>%
select(SERNUM,BENUNIT,PERSON,GROSS4,DISD07,HAPPYWB,ANXIOUS,LIFESAT,MEANING),year=1920),

cbind(read.spss("UKDA-8802-spss/spss/spss25/adult.sav",to.data.frame = T,
use.value.labels = F)%>%
select(SERNUM,BENUNIT,PERSON,GROSS4,DISD07,HAPPYWB,ANXIOUS,LIFESAT,MEANING),year=2021)
),

t<-rbind(
cbind(read.spss("UKDA-8948-spss/spss/spss25/househol.sav",to.data.frame = T,
use.value.labels = F)%>%
select(SERNUM,HHINCBND),year=1920),

cbind(read.spss("UKDA-8802-spss/spss/spss25/househol.sav",to.data.frame = T,
use.value.labels = F)%>%
```

```
select(SERNUM,HHINCBND),year=2021)
),

by=c("SERNUM","year"),all.x=F,all.y=F)

names(combine_years)<-tolower(names(combine_years))
```

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each category for each wellbeing variable Happiness 2019-20 and 2020-21

```
tmptab<-xtabs(gross4~happywb+year,combine_years) ### creates the table

### Computes Frequencies and percentages
cbind(
"Freq"= tmptab,
"Pct"=round(100*prop.table(tmptab),2)
)
#### Cumulated frequencies and percentages
cbind(
"Cum. freq."=cumsum(as.data.frame.matrix(tmptab)),
"Cum. pct"= round(cumsum(as.data.frame.matrix(100*prop.table(tmptab,2))),1)
)
```

Anxiety 2019-20

```
tmptab<-xtabs(gross4~anxious+year,combine_years) ### creates the table

### Computes Frequencies and percents
cbind(
"Freq"= tmptab,
"Pct"=round(100*prop.table(tmptab),2)
)
#### Cumulated frequencies and percentages
cbind(
"Cum. freq."=cumsum(as.data.frame.matrix(tmptab)),
"Cum. pct"= round(cumsum(as.data.frame.matrix(100*prop.table(tmptab,2))),1)
)
```

Life satisfaction 2019-20

```
tmptab<-xtabs(gross4~lifesat+year,combine_years) ### creates the table

### Computes Frequencies and percents
cbind(
"Freq"= tmptab,
"Pct"=round(100*prop.table(tmptab),2)
)
#### Cumulated frequencies and percentages
cbind(
"Cum. freq."=cumsum(as.data.frame.matrix(tmptab)),
"Cum. pct"= round(cumsum(as.data.frame.matrix(100*prop.table(tmptab,2))),1)
)
```

Life Meaningfulness 2019-20 and 2020-21

```

tmptab<-xtabs(gross4~meaning+year,combine_years)

  cbind(
"Freq"= tmptab,
"Pct"=round(100*prop.table(tmptab),2)
)
#### Cumulated frequencies and percentages
cbind(
"Cum. freq."=cumsum(as.data.frame.matrix(tmptab)),
"Cum. pct"= round(cumsum(as.data.frame.matrix(100*prop.table(tmptab,2))),1)
)

```

Examines the distribution of happiness in 2019-20 and 2020-21, including an assessment of normality and discovery of outliers

```

round(t(combine_years%>%filter(!is.na(happywb))%>%group_by(year)%>%summarise(
  "Observations"=length(happywb),
  "Mean"=wtd.mean(happywb,gross4),
  "Median"=wtd.quantile(happywb,gross4,probs=.5),
  "Std Deviation"=sqrt(wtd.var(happywb,gross4)),
  "Variance"=wtd.var(happywb,gross4),
  "Mode"=as.numeric(names(sort(-table(happywb)))[1]),
  "Range"=max(happywb)-min(happywb),
  "IQR"=wtd.quantile(happywb,gross4,probs=.75)-wtd.quantile(happywb,gross4,probs=.25),
  "Skewness"=Weighted.Desc.Stat::w.skewness(happywb,gross4),
  "Kurtosis"=Weighted.Desc.Stat::w.kurtosis(happywb,gross4)
)),2)

```

Weighted quantiles

```

cbind(
combine_years%>%filter(year==1920 & !is.na(happywb))%>%select(happywb,gross4)%>%summarise(
  "Quantiles"=
round(wtd.quantile(happywb,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
),
combine_years%>%filter(year==2021 & !is.na(happywb))%>%select(happywb,gross4)%>%summarise(
  "Quantiles"=
round(wtd.quantile(happywb,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
)
)

```

Extreme values

```

cbind(
  combine_years%>%filter(year==1920 & !is.na(happywb))%>%select(happywb)%>%
  slice_min(order_by = happywb,n=5,with_ties = FALSE),
  combine_years%>%filter(year==2021 & !is.na(happywb))%>%select(happywb)%>%
  slice_max(order_by = happywb,n=5,with_ties = FALSE)
)

```

Examines the distribution of anxiety in 2019-20 and 2020-21, including an assessment of normality and discovery of outliers

```

round(t(combine_years%>%filter(!is.na(anxious))%>%group_by(year)%>%summarise(
  "Observations"=length(anxious),
  "Mean"=wtd.mean(anxious,gross4),
  "Median"=wtd.quantile(anxious,gross4,probs=.5),

```

```

"Std Deviation"=sqrt(wtd.var(anxious,gross4)),
"Variance"=wtd.var(anxious,gross4),
"Mode"=as.numeric(names(sort(-table(anxious)))[1]),
"Range"=max(anxious)-min(anxious),
"IQR"=wtd.quantile(anxious,gross4,probs=.75)-wtd.quantile(anxious,gross4,probs=.25),
"Skewness"=Weighted.Desc.Stat::w.skewness(anxious,gross4),
"Kurtosis"=Weighted.Desc.Stat::w.kurtosis(anxious,gross4)
)),2)

```

Weighted quantiles

```

cbind(
  combine_years%>%filter(year==1920 & !is.na(anxious))%>%select(anxious,gross4)%>%summarise(
    "Quantiles"=
      round(wtd.quantile(anxious,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
  ),
  combine_years%>%filter(year==1920 & !is.na(anxious))%>%select(anxious,gross4)%>%summarise(
    "Quantiles"=
      round(wtd.quantile(anxious,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
  )
)

```

Extreme values

```

cbind(
  combine_years%>%filter(year==1920 & !is.na(anxious))%>%select(anxious)%>%
    slice_min(order_by = anxious,n=5,with_ties = FALSE),
  combine_years%>%filter(year==2021 & !is.na(anxious))%>%select(anxious)%>%
    slice_max(order_by = anxious,n=5,with_ties = FALSE)
)

```

Examines the distribution of life satisfaction in 2019-20 and 2020-21, including an assessment of normality and discovery of outliers

```

round(t(combine_years%>%filter(!is.na(lifesat))%>%group_by(year)%>%summarise(
  "Observations"=length(lifesat),
  "Mean"=wtd.mean(lifesat,gross4),
  "Median"=wtd.quantile(lifesat,gross4,probs=.5),
  "Std Deviation"=sqrt(wtd.var(lifesat,gross4)),
  "Variance"=wtd.var(lifesat,gross4),
  "Mode"=as.numeric(names(sort(-table(lifesat)))[1]),
  "Range"=max(lifesat)-min(lifesat),
  "IQR"=wtd.quantile(lifesat,gross4,probs=.75)-wtd.quantile(lifesat,gross4,probs=.25),
  "Skewness"=Weighted.Desc.Stat::w.skewness(lifesat,gross4),
  "Kurtosis"=Weighted.Desc.Stat::w.kurtosis(lifesat,gross4)
)),2)

```

Weighted quantiles

```

cbind(
  combine_years%>%filter(year==1920 & !is.na(lifesat))%>%select(lifesat,gross4)%>%summarise(
    "Quantiles"=
      round(wtd.quantile(lifesat,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
  ),
  combine_years%>%filter(year==1920 & !is.na(lifesat))%>%select(lifesat,gross4)%>%summarise(
    "Quantiles"=
      round(wtd.quantile(lifesat,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
  )
)

```

```
)
)
```

Extreme values

```
cbind(
  combine_years%>%filter(year==1920 & !is.na(lifesat))%>%select(lifesat)%>%
  slice_min(order_by = lifesat,n=5,with_ties = FALSE),
  combine_years%>%filter(year==2021 & !is.na(lifesat))%>%select(lifesat)%>%
  slice_max(order_by = lifesat,n=5,with_ties = FALSE)
)
```

Examines the distribution of meaning in 2019-20 and 2020-21, including an assessment of normality and discovery of outliers

```
round(t(combine_years%>%filter(!is.na(meaning))%>%group_by(year)%>%summarise(
  "Observations"=length(meaning),
  "Mean"=wtd.mean(meaning,gross4),
  "Median"=wtd.quantile(meaning,gross4,probs=.5),
  "Std Deviation"=sqrt(wtd.var(meaning,gross4)),
  "Variance"=wtd.var(meaning,gross4),
  "Mode"=as.numeric(names(sort(-table(meaning)))[1]),
  "Range"=max(meaning)-min(meaning),
  "IQR"=wtd.quantile(meaning,gross4,probs=.75)-wtd.quantile(meaning,gross4,probs=.25),
  "Skewness"=Weighted.Desc.Stat:w.skewness(meaning,gross4),
  "Kurtosis"=Weighted.Desc.Stat:w.kurtosis(meaning,gross4)
)),2)
```

Weighted quantiles

```
cbind(
  combine_years%>%filter(year==1920 & !is.na(meaning))%>%select(meaning,gross4)%>%summarise(
    "Quantiles"=
    round(wtd.quantile(meaning,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
  ),
  combine_years%>%filter(year==1920 & !is.na(meaning))%>%select(meaning,gross4)%>%summarise(
    "Quantiles"=
    round(wtd.quantile(meaning,gross4,probs=c(.01,.05,.1,.25,.5,.75,.9,.95,.99)),2)
  )
)
```

Extreme values

```
cbind(
  combine_years%>%filter(year==1920 & !is.na(meaning))%>%select(meaning)%>%
  slice_min(order_by = meaning,n=5,with_ties = FALSE),
  combine_years%>%filter(year==2021 & !is.na(meaning))%>%select(meaning)%>%
  slice_max(order_by = meaning,n=5,with_ties = FALSE)
)
```

First getting the happiness tables (we could have reused

the frequency table we created above)

```
t<-
```

```
round(
  100*prop.table(
    xtabs(~happywb+year, combine_years)
  ,2)
  ### Rounding result to 1 decimal
  ### Computing proportions, then multiplying by to get %
  ### basic frequency table
```

```
,1)

### The plotting
barplot(t, beside = T,
        main="Histogram to compare HAPPYWB in 2019-20 to 2020-21",
        xlab="How happy did you feel yesterday?",
        ylab="Percentage"
        )

### First getting the anxiety table
t<-
round(
100*prop.table(
xtabs(~year+anxious, combine_years)
,1)
,1)

### The actual plotting
barplot(t, beside = T,
        main="Histogram to compare ANXIOUS in 2019-20 to 2020-21",
        xlab="How anxious did you feel yesterday?",
        ylab="Percentage"
        )

### Getting the life satisfaction table
t<-
round(
100*prop.table(
xtabs(~year+lifesat, combine_years)
,1)
,1)

### The actual plotting
barplot(t, beside = T,
        main="Histogram to compare LIFESAT in 2019-20 to 2020-21",
        xlab="How satisfied are you with your life nowadays?",
        ylab="Percentage"
        )

### Getting the life meaningfulness data
t<-
round(
100*prop.table(
xtabs(~year+meaning, combine_years)
,1)
,1)

### The actual plotting
barplot(t, beside = T,
        main="Histogram to compare MEANING in 2019-20 to 2020-21",
        xlab="To what extent do you feel that things you do in your life have meaning?",
        ylab="Percentage"
        )
```

Provides grossed analysis of variance and Wilcoxon Scores (Rank Sums) for wellbeing variables

by year

4.1.1 ANOVA happiness by year

ANOVA amounts to running a statistical test of significance of differences in the mean values of a variable across two or more groups. In most cases, linear regression with a single categorical independent (ie X) variable is identical. We will be choosing this approach here, as it is easier to implement in R.

```
summary(
  lm(happywb~as.factor(year),data=combine_years,weights=gross4))

### We need to use the Survey package in order to access the weighted rank test functions
### (Wilcoxon/Kruskal-Wallis)
library(survey)

### We also need to declare the survey design, in this case we will only specify the
### grossing weights
design <- svydesign(ids = ~0, data=combine_years,
  weights=combine_years$gross4)

svyranktest(happywb ~ as.factor(year), design = design)
```

4.1.2 ANOVA Anxiety amount by year

/Provides grossed analysis of variance and Kruskal-Wallis Rank Sums for State Pension benefit amounts by year/

```
summary(lm(anxious~as.factor(year),data=combine_years,weights=gross4))
svyranktest(anxious ~ as.factor(year), design = design)
```

4.1.3 ANOVA Life satisfaction by year

/Provides grossed analysis of variance and Kruskal-Wallis Rank Sums for Personal Independence Payment (PIP) Daily Living benefit amounts by year/

```
summary(lm(anxious~as.factor(year),data=combine_years,weights=gross4))
svyranktest(anxious ~ as.factor(year), design = design)
```

4.1.4 ANOVA life meaning by year

/Provides grossed analysis of variance and Kruskal-Wallis Rank Sums for Personal Independence Payment (PIP) Mobility benefit amounts by year/

```
summary(lm(anxious~as.factor(year),data=combine_years,weights=gross4))
svyranktest(anxious ~ as.factor(year), design = design)
```

4.2 Tables of wellbeing variable scores against other variables

Create a grossed table of life satisfaction score against whether the adult has difficulty with mental health in 2019-20

```
### 2019-20
round(
  100*prop.table(
    xtabs(gross4~lifesat+disd07,combine_years%>%filter(year==1920))
    ,2),1)
### 2020-21
```

```
round(
100*prop.table(
  xtabs(gross4~lifesat+disd07,combine_years%>%filter(year==2021))
,2),1)
```

Create a grossed table of anxiousness score against the household income band of the adult in 2019-20

```
### 2019-20
round(
100*prop.table(
  xtabs(gross4~anxious+hhincbnd,combine_years%>%filter(year==1920))
,2),1)

### 2020-21
round(
100*prop.table(
  xtabs(gross4~anxious+hhincbnd,combine_years%>%filter(year==2021))
,2),1)
```

5 Rapid analysis

5.1 FRS households by tenure between 2018-19 and 2020-21

```
setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/" )
tenure1<-read.spss("UKDA-8948-spss/spss/spss25/househol.sav",to.data.frame = T,
use.value.labels = T,
add.undeclared.levels="append")%>%select(SERNUM,TENURE,GROSS4)
names(tenure1)<-tolower(names(tenure1))

tenure2<-read.spss("UKDA-8802-spss/spss/spss25/househol.sav",to.data.frame = T,
use.value.labels = T,
add.undeclared.levels="append")%>%select(SERNUM,TENURE,GROSS4)
names(tenure2)<-tolower(names(tenure2))

tenure3<-read.spss("UKDA-8633-spss/househol.sav",to.data.frame = T,
use.value.labels = T,
add.undeclared.levels="append")%>%select(sernum,TENURE,GROSS4)
names(tenure3)<-tolower(names(tenure3))
```

2020-21

```
tmptab<-xtabs(gross4~tenure,tenure1,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

2019-20

```
tmptab<-xtabs(gross4~tenure,tenure2,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
```



```
"Cum. freq."=cumsum(tmptab),
"Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

2018-19

```
tmptab<-xtabs(gross4~tenure,tenure3,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

5.2 FRS households by Council Tax Band between 2018-19 and 2020-21

```
setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/" )
ctband1<-read.spss("UKDA-8948-spss/spss/spss25/househol.sav",to.data.frame = T,
  use.value.labels = T,
  add.undeclared.levels="append")%>%select(SERNUM,CTBAND,GROSS4)
names(ctband1)<-tolower(names(ctband1))

ctband2<-read.spss("UKDA-8802-spss/spss/spss25/househol.sav",to.data.frame = T,
  use.value.labels = T,
  add.undeclared.levels="append")%>%select(SERNUM,CTBAND,GROSS4)
names(ctband2)<-tolower(names(ctband2))

ctband3<-read.spss("UKDA-8633-spss/househol.sav",to.data.frame = T,
  use.value.labels = T,
  add.undeclared.levels="append")%>%select(sernum,CTBAND,GROSS4)
names(ctband3)<-tolower(names(ctband3))
```

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each category for the variable 'ctband' in 2020-21

```
tmptab<-xtabs(gross4~ctband,ctband1,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each category for the variable 'ctband' in 2019-20

```
tmptab<-xtabs(gross4~ctband,ctband2,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)
```

Gives the grossed frequency, percentage, cumulative frequency and cumulative percentage of each category for the variable 'ctband' in 2018-19

```

tmptab<-xtabs(gross4~ctband,ctband3,drop.unused.levels=T)
cbind(
  "Freq"= tmptab,
  "Pct"=round(100*prop.table(tmptab),1),
  "Cum. freq."=cumsum(tmptab),
  "Cum. pct"= round(cumsum(100*prop.table(tmptab)),2)
)

```

5.3 FRS adult respondents by highest level of qualification between 2018-19 and 2020-21

```

setwd("/home/piet/Dropbox/work/UKDS/FRS workshop/" )
dvhiqual12<-rbind(
  cbind(read.spss("UKDA-8948-spss/spss/spss25/adult.sav",to.data.frame = T,
    use.value.labels = F,
    add.undeclared.levels="append")%>%select(SERNUM,BENUNIT,PERSON, DVHIQUAL,GROSS4),
    year=2021),
  cbind(read.spss("UKDA-8802-spss/spss/spss25/adult.sav",to.data.frame = T,
    use.value.labels = F,
    add.undeclared.levels="append")%>%select(SERNUM,BENUNIT,PERSON, DVHIQUAL,GROSS4),
    year=2020)
)
names(dvhiqual12)<-tolower(names(dvhiqual12))

### Unfortunately one of the datasets has a different variable name, so we need to deal
### with it separately
dvhiqual3<-cbind(read.spss("UKDA-8633-spss/adult.sav",to.data.frame = T,
  use.value.labels = F,
  add.undeclared.levels="append")%>%select(sernum,BENUNIT,PERSON, DVHIQUAL,GROSS4),
  year=2019)
names(dvhiqual3)<-tolower(names(dvhiqual3))

dvhiqual123<-rbind(
  dvhiqual12,dvhiqual3
)

dvhiqual123<-dvhiqual123%>%mutate(dvhiqual.s=case_when(
  dvhiqual>=1 & dvhiqual<=16 ~ 'Level 4 or greater',
  dvhiqual>=17 & dvhiqual<=34 ~ 'Level 3',
  dvhiqual>=35 & dvhiqual<=54 ~ 'Level 2',
  dvhiqual>=55 & dvhiqual<=82 ~ 'Level 1',
  dvhiqual>=83 & dvhiqual<=85 ~ 'Entry level',
  dvhiqual==86 ~ 'Other',
  is.na(dvhiqual)==T) ~ 'No reported qualification'
))

### Percentages
cbind(
  round(100*prop.table(xtabs(gross4~dvhiqual.s+year,dvhiqual123),2),1),
  ### Then grossed frequencies
  xtabs(gross4~dvhiqual.s+year,dvhiqual123)
)

```