SAS-makro for å hente kodeliste fra Klass

Innledning	3
Bruk av programmet	3
De enkelte parameterne til programmet	
klass_katnr	15
format_prefiks og format_suffiks	15
slette_midl	16
rest_gruppering og rest_tekst	17
klass_spraak	
fra_dato	18
til_dato	19
format_tekst	20
utds_kodeliste	21
utds_klassifikasjonInfo	21
utds_format	22
utds_dubletter	22
liste_dubletter_alle	22
utds_dubletter_alle	23
utds_restgruppering	24
behold_restgruppe_data	24
behold_info_data	24
behold_kodeliste_data	24
lag_format	25
format_lib	25
adresse	25
lengde_kode	26
lengde_tekst	26
lengde_tekst_kort	26
lengde_record	26
variant	26
uttrekkstype	27

tegnsett	28
Finne nummeret til en kodeliste	28
Spesielle forhold	31

Innledning

Vår nye kodeverksdatabase heter Klass. Den vil inneholde alle våre offisielle standarder samt de fleste kodelistene vi bruker i statistikkproduksjonen. For å sikre at kodelistene er oppdatert og ikke finnes i flere varianter bør de alltid hentes fra Klass for bruk i statistikkproduksjonen. Denne SASmakroen er laget for å forenkle uttrekk av kodelister til bruk i SAS. Det kan brukes både på Linux og Windows.

Hver kodeliste i Klass har et eget nummer og det nummeret skal brukes som en parameter i makroen.

Makroen er skrevet av Kristian Lønø og brukes på eget ansvar.

Bruk av programmet

Programmet er en SAS-makro som først inkluderes i et SAS-program. For Linux gjøres det på denne måten:

```
%include '$FELLES/sasprog/KodelisteFraKlass.sas';
```

For Windows kaller vi makroen slik

```
%include
```

'Q:\saskurs\fellesprog\KodelisteFraKlass.sas';

Deretter kan vi kalle makroen. Den har følgende parametre:

Parameternavn	Beskrivelse	Kommentar	Standardverdi	Må fylles ut
adresse	URL til kodeverksdatabas en Klass. Det kan hende at en eller flere av disse adressene endres	Produksjon: http://data.ssb. no/api/klass/v1 QA: http://beta- data.ssb.no/api/ klass/v1/ Test: http://klass- test.ssb.no/rest/ v1 Utvikling: http://al-klass- app- u1.ssb.no:8080/ rest/v1	http://data.ssb.no/api/klass/v1	Nei

klass_katnr	Kodelistenummer	Se vedlegg for	131 (kommuner)	Nei
	fra Klass.	hvordan vi		
		finner		
		kodelistenumme		
		r		
klass spraak	Carôle	ND or bokmål	NB	Noi
klass_spraak	Språk, tobokstavskode	NB er bokmål, NN er Nynorsk	IND	Nei,
	tobokstavskode	og EN er		bare
		engelsk. De		hvis
		fleste		kodelist
		kodelistene		en ikke
		finnes på		finnes
		bokmål og en		på
		del også på nynorsk og		bokmål
		engelsk. Andre		
		språk er også		
		tillatt så lenge		
		de er lagt inn i		
		Klass		
fra_dato	Fra-dato for valgt	Dato for når	1.januar i fjor	Nei
	gyldighetsperiode	kodelisten		
		gjelder for på		
		formen ÅÅÅÅ-		
		MM-DD		
til_dato	Til-dato for valgt	Når til-dato		Nei
	gyldighetsperiode	brukes kan det		
		føre til at flere		
		rader får samme		
		kode. Hvis det		
		lages SAS-		
		format fra		
		denne		
		kodelisten		
		fjernes de eldste		
		dublettene før		
		formatet lages.		
		Oppgis på		
		formen ÅÅÅÅ-		
		MM-DD		
utde kodeliste	Nava as CAC	Challed CAC	Vlace Vadalists	Noi
utds_kodeliste	Navn på SAS- datasett som vil	Skal følge SAS	Klass_Kodeliste	Nei
	inneholde	sin navnestandard		
	kodelisten.	Haviiestallualu		

		for datasett		
utds_klassifikasjonInfo	Navn på SAS- datasett som vil inneholde informasjon om kodelisten.	Skal følge SAS sin navnestandard for datasett	Klass_Info	Nei
utds_format	Navn på SAS- datasett som det lages format fra	Skal følge SAS sin navnestandard for datasett	Klass_Format	Nei
utds_restgruppering	Navn på SAS- datasett som inneholder restgruppering for format	Skal følge SAS sin navnestandard for datasett	Klass_Restgruppering	Nei
utds_dubletter	Navn på SAS- datasett som inneholder dubletter for koder	Skal følge SAS sin navnestandard for datasett	Klass_Dubletter	Nei
liste_dubletter_alle	1 hvis det skal lages liste og datasett for alle dubletter, også de som blir beholdt	Listes ut før liste over dubletter som slettes	0	Nei
utds_dubletter_alle	Navn på SAS- datasett som inneholder alle dubletter, ikke bare de som blir slettet	Blir bare laget om liste_dubletter_ alle er 1	Klass_Dubletter_alle	Nei
behold_info_data	1 hvis info- datasett skal beholdes. Info- datasett inneholder informasjon om kodelisten og hvilke versjoner den har	Alle andre verdier enn 1 vil slette info- datasettet	1	Nei
behold_kodeliste_data	1 hvis kodelistedatasett skal beholdes.	Alle andre verdier enn 1 vil slette kodelistedataset	1	Nei

		tet		
behold_restgruppe_data	1 hvis restgrupperingsd atasett skal beholdes.	Alle andre verdier enn 1 vil slette restgrupperings datasettet	0	Nei
slette_midl	1 hvis midlertidige SAS- datasett ønskes beholdt	Alle andre verdier enn 1 vil slette det midlertidige SAS-datasettet	1	Nei
lag_format	1 hvis det skal lages et format fra kodelisten	Eventuelle dubletter for kode blir slettet før formatet lages	1	Nei
format_lib	SAS-mappe for formatet	Hvis library brukes må denne være definert på forhånd med en Libname-setning	Work	Nei
format_prefiks	Prefiks til format- navnet. Gir et forslag til navn prefiks til formatnavnet	Formatnavnet består til sammen av prefiks og suffiks	\$f + kodelistenr.	Nei
format_suffiks	suffiks til format- navnet	Hvis suffiks er oppgitt vil prefiks og suffiks til sammen bli formatnavnet. Hvis suffiks ikke er oppgitt vil nivå (om det er flere enn ett) og språk skilt med understrek bil suffiks. Formatnavnet skal slutte på en bokstav. Hvis det ikke gjør det		Nei

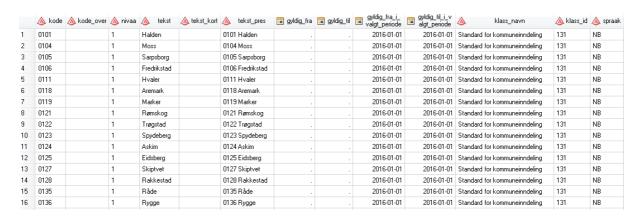
		legges språket på til slutt		
format_tekst	1 hvis kode og lang tekst skal bli teksten til formatet 2 hvis kode og kort tekst skal brukes 3 hvis lang tekst skal brukes 4 hvis kort tekst skal brukes	Viktig å ta med koden som en del av teksten om flere koder har samme tekst og de ikke ønskes slått sammen til en gruppe. Hvis kort tekst er valgt og den ikke finnes brukes lang tekst isteden	1	Nei
rest_gruppering	1 hvis det skal lages en egen tekst for andre verdier enn de som finnes i kodelisten	Lager en OTHER i Proc format. Alle andre verdier enn 1 vil ikke ta med rest- gruppering	1	Nei
rest_tekst	Tekst for rest-gruppering.	Bruk en tekst som ikke finnes for noen av kodene i kodelisten for å unngå at denne slås sammen med en annen kode med samme tekst	Uoppgitt	Nei
lengde_kode	Lengden på variabelen med koden	Må økes om koden har mer enn 15 tegn	15	Nei
lengde_tekst	Lengden på variabelen med lang tekst og presentasjonsteks t	Må økes om koden har mer enn 512 tegn	512	Nei
lengde_tekst_kort	Lengden på variabelen med	Må økes om koden har mer	256	Nei

	kort tekst og presentasjonsteks t	enn 256 tegn		
lengde_record	Lengden på radene som hentes fra Klass	Må økes om noen rader har mer enn 9999 tegn	9999	Nei
variant	Angir om vi skal hente ut en variant av en kodeliste	1 er variant, alle andre verdier gir en vanlig kodeliste	0	Nei
uttrekkstype	Angis for å velge format for uttrekket fra Klass.	CSV og JSON er godtatte typer.	CSV	Nei
tegnsett	for uttrekkstype json (se over) kan vi velge hvilket tegnsett det skal konverteres til.	Bør ikke være annerledes enn det tegnsettet SAS-sesjonen bruker	latin1	Nei

Vi kan nå hente kommunekatalogen for 2016 fra Klass og lage et SAS-format for den med et slikt program:

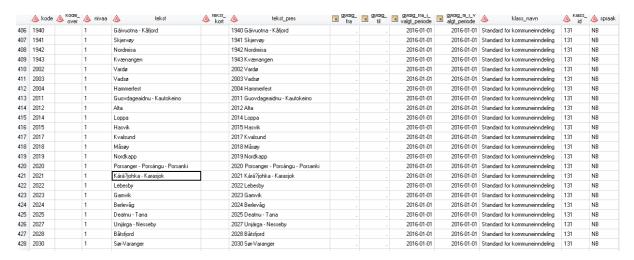
```
%KodelisteFraKlass
(klass_katnr=131,
  fra_dato=2016-01-01
)
```

Kodelisten som hentes ut fra Klass blir slik som SAS-datasett:



Vi ser at det ikke er lagt inn noen kort tekst i Klass for denne kodelisten. I så fall ville variabelen tekst_kort inneholde denne. Navnet på kodelisten finner vi i variabelen klass_navn. Dette vil være tomt hvis vi bruker en tidligere SAS-versjon enn 9.4. Variabelen kode_over inneholder kode for overliggende nivå for de som ikke er på det høyeste (mest aggregerte) nivået om det er flere nivåer i kodelisten. Listen over har ikke flere nivåer og kode_over er følgelig tom. Dato-variablene gyldig_fra og gyldig_til får verdi om de er lagt inn i klassifikasjonen i Klass. Hvis ikke blir de uten verdi. Variablene gyldig_fra_i_valgt_periode og gyldig_til_i_valgt_periode angir når en kode er gyldig innen det intervallet som er valgt.

Hvis vi ser på kommunene i Finnmark ser vi at det er problemer med teksten for Kárášjohka – Karasjok:

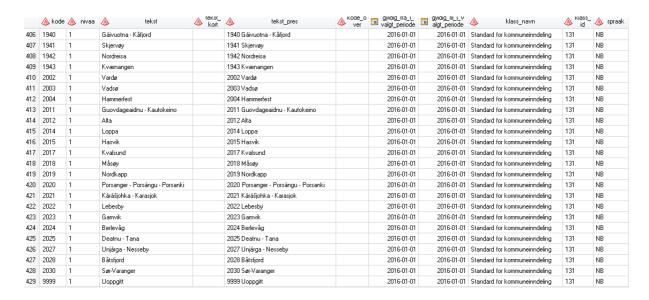


Det kommer ut et spørsmålstegn istedenfor š. Ved å bruke uttrekkstype JSON isteden blir det riktig. Her er programmet:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  uttrekkstype=json
);
```

Slik ser det nå ut, rekkefølgen på variablene blir noe annerledes (bl.a. er gyldig_fra og gyldig_til plassert bakerst, men vises ikke her) :



Tegnet vises riktig på Windows, men er dessverre usynlig på Linux. Det ser vi om vi skriver ut kodelisten, f.eks. med et program som dette:

```
proc print data=klass_kodeliste (firstobs=418) n;
var kode nivaa tekst tekst_pres klass_id;
title "Utskrift på Linux";
run;
```

I denne utskriften blir š blank når det kjøres på Linux:

Utskrift på Linux

Obs	kode	nivaa	tekst	tekst_pres	klass_id
418	2018	1	Måsøy	2018 Måsøy	131
419	2019	1	Nordkapp	2019 Nordkapp	131
420	2020	1	Porsanger - Porsángu - Porsanki	2020 Porsanger - Porsángu - Porsanki	131
421	2021	1	Kárá johka - Karasjok	2021 Kárá johka - Karasjok	131
422	2022	1	Lebesby	2022 Lebesby	131
423	2023	1	Gamvik	2023 Gamvik	131
424	2024	1	Berlevåg	2024 Berlevåg	131
425	2025	1	Deatnu - Tana	2025 Deatnu - Tana	131
426	2027	1	Unjárga - Nesseby	2027 Unjárga - Nesseby	131
427	2028	1	Båtsfjord	2028 Båtsfjord	131
428	2030	1	Sør-Varanger	2030 Sør-Varanger	131
429	9999	1	Uoppgitt	9999 Uoppgitt	131
N = 1	2				

Prøver vi å kopiere eller eksportere utlistingen til Windows forsvinner tegnet helt. Tegnet š er problematisk for SAS på Linux. En måte å få tegnet til å synes er å lagre utskriften i en fil på Linux, f.eks. slik:

```
ods html path='' file='$HOME/kommuneliste.html' style=htmlblue;
proc print data=klass_kodeliste (firstobs=418) n;
var kode nivaa tekst tekst_pres klass_id;
title "Utskrift på Linux";
run;
ods html close;
```

I EG vil vi nå få 2 valg i Results:



Det øverste er standardutskriften SAS lager, som gir feil resultat. Det nederste refererer til filen vi har lagt resultatet på. Når vi velger den blir vi spurt om vi skal laste den ned og det vil vi. Filen som blir lastet ned har nå de riktige tegnene:

```
S Refresh | Export → Send To → Publish | F Properties
```

Utskrift på Linux

Obs	kode	nivaa	tekst	tekst_pres	klass_id
418	2018	1	Måsøy	2018 Måsøy	131
419	2019	1	Nordkapp	2019 Nordkapp	131
420	2020	1	Porsanger - Porsángu - Porsanki	2020 Porsanger - Porsángu - Porsanki	131
421	2021	1	Kárášjohka - Karasjok	2021 Kárášjohka - Karasjok	131
422	2022	1	Lebesby	2022 Lebesby	131
423	2023	1	Gamvik	2023 Gamvik	131
424	2024	1	Berlevåg	2024 Berlevåg	131
425	2025	1	Deatnu - Tana	2025 Deatnu - Tana	131
426	2027	1	Unjárga - Nesseby	2027 Unjárga - Nesseby	131
427	2028	1	Båtsfjord	2028 Båtsfjord	131
428	2030	1	Sør-Varanger	2030 Sør-Varanger	131
429	9999	1	Uoppgitt	9999 Uoppgitt	131
N = 1	12		1		

Denne utskriften vil få med seg tegnet når vi f.eks. kopierer utskriften videre til Excel. Dette virker bare for tegn som finnes i tegnsettet latin1 på Windows.

Kjører vi derimot på Windows vises š uten å gå noen omveier. Vi kjører dette programmet på Windows:

```
%include 'q:\saskurs\fellesprog\KodelisteFraKlass.sas';
%KodelisteFraKlass
(klass_katnr=131,
   fra_dato=2016-01-01,
   uttrekkstype=json
)
;
proc print data=klass_kodeliste (firstobs=418) n;
   var kode nivaa tekst tekst_pres klass_id;
   title "Utskrift på Windows";
run;
```

Nå blir utskriften riktig:

Utskrift på Windows

%KodelisteFraKlass

run;

Obs	kode	nivaa	tekst	tekst_pres	klass_id
418	2018	1	Måsøy	2018 Måsøy	131
419	2019	1	Nordkapp	2019 Nordkapp	131
420	2020	1	Porsanger - Porsángu - Porsanki	2020 Porsanger - Porsángu - Porsanki	131
421	2021	1	Kárášjohka - Karasjok	2021 Kárášjohka - Karasjok	131
422	2022	1	Lebesby	2022 Lebesby	131
423	2023	1	Gamvik	2023 Gamvik	131
424	2024	1	Berlevåg	2024 Berlevåg	131
425	2025	1	Deatnu - Tana	2025 Deatnu - Tana	131
426	2027	1	Unjárga - Nesseby	2027 Unjárga - Nesseby	131
427	2028	1	Båtsfjord	2028 Båtsfjord	131
428	2030	1	Sør-Varanger	2030 Sør-Varanger	131
429	9999	1	Uoppgitt	9999 Uoppgitt	131
N = 1	12				

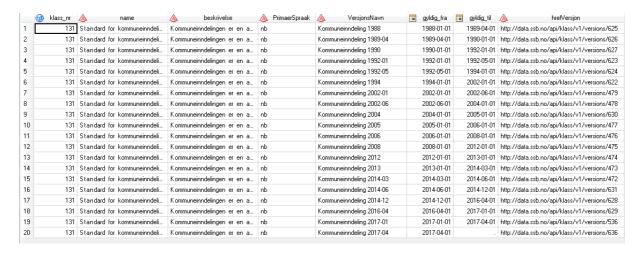
For å unngå disse problemene på Linux kan vi eventuelt endre fra š til vanlig s og så lage formatet på nytt:

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  uttrekkstype=json,
  slette_midl=0
)
;

data klass_format;
  set klass_format;
  label = translate(label,'s','9A'x);
run;
```

proc format cntlin=klass_format;

Det blir også laget et SAS-datasett som inneholder informasjon om kodelisten. Listen ser slik ut:



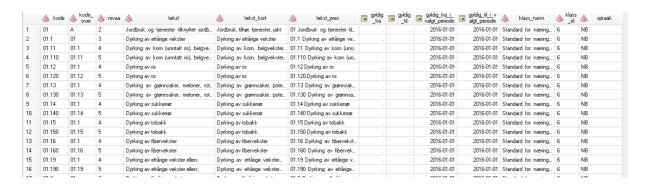
Det viser oss hvilke versjoner som finnes av denne kodelisten og hvilken adresse (url) hver versjon har. Vi ser også hva som er kodelistens primærspråk.

SAS-formatet som lages vil få navnet \$f131_NB. Det inneholder kun et nivå og derfor er det ikke nødvendig å ta med nivået i navnet. For næringsgruppering er situasjonen annerledes. Vi henter ut katalogen for næring:

%KodelisteFraKlass

```
(klass_katnr=6,
  fra_dato=2016-01-01,
  slette_midl=0
);
```

Her ser vi at nivåvariabelen har forskjellige verdier:



Vi ser også at variabelen kode_over har verdien til det overliggende nivået i kodelisten. Dette gjør det enkelt å lage en omkodingstabell fra et nivå til et overliggende nivå basert på uttrekket fra Klass. Vi får med nivået i formatnavnet slik at det blir laget et format for hvert nivå. Vi ser at navnet inneholder nivået:

	🔈 start	▲ label	♠ fmtnam e	🔌 hlo	gyldig_ fra	gyldig_ til	gyldig_fra_i_ valgt_periode	gyldig_til_i_v algt_periode
98	86	86 Helsetjenester	\$f6_2_NB	S			2016-01-01	2016-01-01
99	87	87 Pleie- og omsorgstjen	\$f6_2_NB	S			2016-01-01	2016-01-01
100	88	88 Sosiale omsorgstjenes	\$f6_2_NB	S			2016-01-01	2016-01-01
101	90	90 Kunstnerisk virksomhe	\$f6_2_NB	S			2016-01-01	2016-01-01
102	91	91 Drift av biblioteker, arki	\$f6_2_NB	S			2016-01-01	2016-01-01
103	92	92 Lotteri og totalisatorspill	\$f6_2_NB	S			2016-01-01	2016-01-01
104	93	93 Sports- og fritidsaktivit	\$f6_2_NB	S			2016-01-01	2016-01-01
105	94	94 Aktiviteter i medlemsor	\$f6_2_NB	S			2016-01-01	2016-01-01
106	95	95 Reparasjon av datama	\$f6_2_NB	S			2016-01-01	2016-01-01
107	96	96 Annen personlig tjenes	\$f6_2_NB	S			2016-01-01	2016-01-01
108	97	97 Lønnet arbeid i private	\$f6_2_NB	S			2016-01-01	2016-01-01
109	99	99 Internasjonale organis	\$f6_2_NB	S			2016-01-01	2016-01-01
110		Uoppgitt	\$f6_2_NB	SO			2016-01-01	2016-01-01
111	01.1	01.1 Dyrking av ettårige v	\$f6_3_NB	S			2016-01-01	2016-01-01
112	01.2	01.2 Dyrking av flerårige	\$f6_3_NB	S			2016-01-01	2016-01-01
113	01.3	01.3 Planteformering	\$f6_3_NB	S			2016-01-01	2016-01-01
114	01.4	01.4 Husdyrhold	\$f6_3_NB	S			2016-01-01	2016-01-01
115	01.5	01.5 Kombinert husdyrhol	\$f6_3_NB	S			2016-01-01	2016-01-01
116	01.6	01.6 Tjenester tilknyttet jo	\$f6_3_NB	S			2016-01-01	2016-01-01
117	01.7	01.7 Jakt, viltstell og tjene	\$f6_3_NB	S			2016-01-01	2016-01-01
118	02.1	02.1 Skogskjøtsel og andr	\$f6_3_NB	S			2016-01-01	2016-01-01
440	00.0	00.01 11 1	*** 0 ***	_			0010.01.01	0040.04.04

De enkelte parameterne til programmet

Her følger eksempler på bruk av parameterne i programmet. Noen av parameterne sjekkes og blir eventuelt rettet før de brukes videre i programmet. Andre godtas som de er og hvis de inneholder ugyldige verdier vil programmet kunne feile.

klass_katnr

Her oppgis nummeret kodelisten har i Klass. Dette må vi vite før vi kjører programmet. Se side 28 for hvordan dette kan finnes vha. SAS. Hvis vi ikke oppgir noe nummer, hentes kodelisten for kommuneinndeling.

Hvis vi oppgir et feil eller ugyldig klassifikasjonsnummer får vi denne feilmeldingen:

ERROR: Invalid reply received from the HTTP server. Use the debug option for more info.

format_prefiks og format_suffiks

Om vi ikke bruker format_prefiks blir prefikset satt til \$f og nummeret kodelisten har i Klass. Det er normalt ikke noe godt navn så vi bør angi navnet selv. Om vi ikke bruker format_suffiks lages det et suffiks. Det består av nivåindikatoren (om det er flere nivåer i kodelisten) samt språkkoden. Disse skilles med en understrek. Bruker vi både format_prefiks og format_suffiks bestemmer vi hele formatnavnet selv. De to til sammen blir da formatnavnet. Formatnavnet må følge SAS sin navnestandard for formater ellers får vi feilmelding. æøå er f.eks. ikke lov å bruke. Her lager vi et format som skal hete \$kommune:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  format_prefiks=$,
  format_suffiks=kommune,
  uttrekkstype=json
)
:
```

Hvordan vi deler opp i prefiks og suffiks spiller ingen rolle, de settes sammen til en tekst som brukes som formatnavn. Formatnavnet skal dog slutte på en bokstav fra a-z. Gjør det ikke det legges språkkoden på navnet.

slette_midl

Hvis vi vil beholde SAS-datasettet formatet lages fra setter vi parameteren slette_midl til 0.

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  format_prefiks=$kommune,
  slette_midl=0,
  uttrekkstype=json
)
;
```

Her ser vi slutten av SAS-datasettet som brukes til å lage formater:

	🔈 start	♠ label	fmtname	🔌 hlo	gylalg_ fra	gylalg_ til	gylalg_rra_i_ walgt_periode	gylaig_til_i_v algt_periode
407	1941	1941 Skjervøy	\$kommune_NB	S			2016-01-01	2016-01-01
408	1942	1942 Nordreisa	\$kommune_NB	S			2016-01-01	2016-01-01
409	1943	1943 Kvænangen	\$kommune_NB	S			2016-01-01	2016-01-01
410	2002	2002 Vardø	\$kommune_NB	S			2016-01-01	2016-01-01
411	2003	2003 Vadsø	\$kommune_NB	S			2016-01-01	2016-01-01
412	2004	2004 Hammerfest	\$kommune_NB	S			2016-01-01	2016-01-01
413	2011	2011 Guovdageaidnu - Kautokeino	\$kommune_NB	S			2016-01-01	2016-01-01
414	2012	2012 Alta	\$kommune_NB	S			2016-01-01	2016-01-01
415	2014	2014 Loppa	\$kommune_NB	S			2016-01-01	2016-01-01
416	2015	2015 Hasvik	\$kommune_NB	S			2016-01-01	2016-01-01
417	2017	2017 Kvalsund	\$kommune_NB	S			2016-01-01	2016-01-01
418	2018	2018 Måsøy	\$kommune_NB	S			2016-01-01	2016-01-01
419	2019	2019 Nordkapp	\$kommune_NB	S			2016-01-01	2016-01-01
420	2020	2020 Porsanger - Porsángu - Porsanki	\$kommune_NB	S			2016-01-01	2016-01-01
421	2021	2021 Kárášjohka - Karasjok	\$kommune_NB	S			2016-01-01	2016-01-01
422	2022	2022 Lebesby	\$kommune_NB	S			2016-01-01	2016-01-01
423	2023	2023 Gamvik	\$kommune_NB	S			2016-01-01	2016-01-01
424	2024	2024 Berlevåg	\$kommune_NB	S			2016-01-01	2016-01-01
425	2025	2025 Deatnu - Tana	\$kommune_NB	S			2016-01-01	2016-01-01
426	2027	2027 Unjárga - Nesseby	\$kommune_NB	S			2016-01-01	2016-01-01
427	2028	2028 Båtsfjord	\$kommune_NB	S			2016-01-01	2016-01-01
428	2030	2030 Sør-Varanger	\$kommune_NB	S			2016-01-01	2016-01-01
429	9999	9999 Uoppgitt	\$kommune_NB	S			2016-01-01	2016-01-01
430		Uoppgitt	\$kommune_NB	so			2016-01-01	2016-01-01

Når standardverdien 1 er gitt slettes de temporære SAS-datasettene som som standard heter Klass_Format og Klass_Dubletter. Med alle andre verdier for slette_midl beholdes SAS-datasettene.

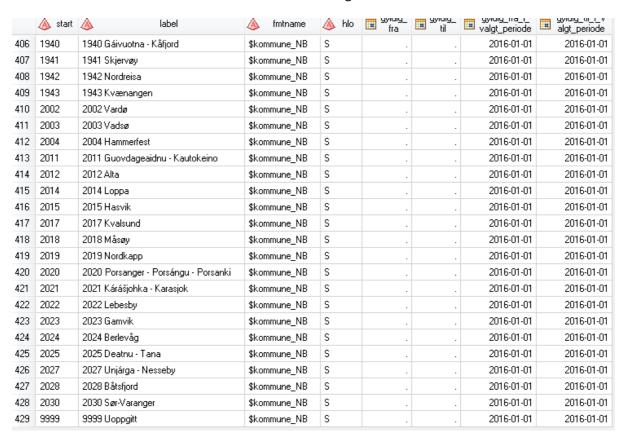
rest_gruppering og rest_tekst

Vi får lagt til en restkategori med teksten Uoppgitt om vi ikke spesifiserer at vi ikke ønsker det (se figur over). Alle verdier som ikke finnes ellers i kodelisten vil nå samles i denne restgruppen. Ønsker vi ikke å ha denne setter vi parameteren rest_gruppering til 0 (dvs. den settes til en annen verdi enn 1):

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  format_prefiks=$kommune,
  slette_midl=0,
  rest_gruppering=0,
  uttrekkstype=json
)
;
```

SAS-datasettet har nå ikke noen restgruppering. Det betyr at koder som ikke finnes i formatet vil beholde den koden de har fordi det ikke er noen tekst å gi dem når formatet brukes:



Vi kan endre teksten for rest-gruppen til en annen tekst. Da bruker vi rest_tekst, f.eks. slik:

```
(klass_katnr=131,
```

```
fra_dato=2016-01-01,
format_prefiks=$kommune,
slette_midl=0,
rest_tekst="Andre",
uttrekkstype=json
)
;
```

Det spiller ingen rolle om teksten står i fnutter eller ei.

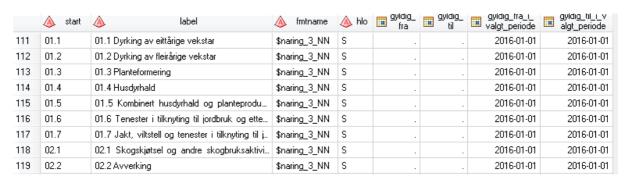
klass_spraak

Kodelister skal normalt finnes både på bokmål (NB) og nynorsk (NN). De kan også finnes på engelsk (EN). Næringsinndeling finnes på nynorsk og den kan vi hente ut på denne måten:

%KodelisteFraKlass

```
(klass_katnr=6,
  fra_dato=2016-01-01,
  format_prefiks=$naring,
  klass_spraak=NN,
  slette_midl=0
);
```

Tekstene kommer nå ut på nynorsk:



Hvis vi oppgir en språkkode klassifikasjonen ikke finnes for får vi denne feilmeldingen:

```
ERROR: Bad request. Use the debug option for more info.
```

fra_dato

Når vi skal hente en kodeliste kan vi hente ut en som er gyldig på et bestemt tidspunkt. Dette tidspunktet angis i fra_dato på formen ÅÅÅÅ-MM-DD. Hvis vi ikke oppgir noen fra_dato hentes katalogen som var gyldig 1. januar i fjor.

Hvis vi oppgir en dato klassifikasjonen ikke finnes for blir SAS-datasettet Klass_kodeliste tomt og programmet avsluttes med en note som dette:

```
NOTE: Ingen rader funnet for klassifikasjon 131 for dato 1988-01-01. Programmet avsluttes.
```

Hvis vi oppgir en ugyldig dato får vi denne feilmeldingen:

ERROR: Bad request. Use the debug option for more info.

Når vi bruker fra_dato, men ikke til_dato vil vi få ut kodelisten som er gyldig på det tidspunktet fra_dato er satt til. Bruker vi til_dato også hentes alle gyldige koder i intervallet fra fra_dato til til_dato, men selve fra_dato og til_dato inkluderes *ikke* i de kodene som hentes ut.

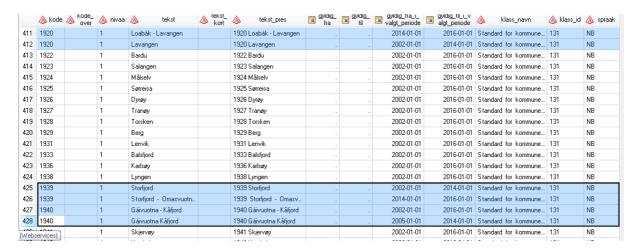
til dato

Vi kan hente en kodeliste med verdier som er gyldige i et tidsrom. Da får vi gjerne gjentatt samme kode flere ganger. De blir beholdt i SAS-datasettet som lages fra kodelisten, men de eldste dublettene slettes før det lages SAS-format. Vær klar over at når vi har med til_dato hentes kodelistene som er gyldige til, men *ikke* inkludert, denne datoen. Det samme gjelder for fra_dato, den er *ikke* inkludert i intervallet. Her henter vi ut kommuner over et lengre tidsrom:

%KodelisteFraKlass

```
(klass_katnr=131,
  format_prefiks=$kommune,
  fra_dato=2002-01-01,
  til_dato=2016-01-01,
  slette_midl=0,
  uttrekkstype=json
);
```

Vi vil få ut noen rader med samme verdi for koden i SAS-datasettet som lages. Her er et eksempel på dette:



Vi ser at kommunene 1920, 1939 og 1940 endret navn i 2014. Før vi lager format fra dette SAS-datasettet må vi fjerne dublettene og vi beholder altså den med høyest gyldig til-dato. Når dette skjer lages det også en liste som viser hvilke rader som er slettet:

Obs	Kode	Tekst	Formatnavn	Indikator for restgruppering og for å beholde sortering	Dato gyldig fra	Dato gyldig til	Dato gyldig fra i valgt periode	Dato gyldig til i valgt periode
- 1	0701	0701 Horten	\$kommune_NB	S			2002-06-01	2016-01-01
2	1736	1736 Snåase - Snåsa	\$kommune_NB	S			2014-01-01	2016-01-01
3	1736	1736 Snåsa	\$kommune_NB	S			2002-01-01	2013-01-01
4	1739	1739 Røyrvik	\$kommune_NB	s			2002-01-01	2014-01-01
5	1849	1849 Hamarøy - Hábmer	\$kommune_NB	S			2014-01-01	2016-01-01
6	1849	1849 Hábmer Hamarøy	\$kommune_NB	s			2013-01-01	2014-01-01
- 7	1850	1850 Tysfjord	\$kommune_NB	S			2002-01-01	2013-01-01
8	1850	1850 Dixtasvuodna - Tysfjord	\$kommune_NB	S			2014-01-01	2016-01-01
9	1920	1920 Loabák - Lavangen	\$kommune_NB	S			2014-01-01	2016-01-01
10	1939	1939 Storfjord	\$kommune_NB	s			2002-01-01	2014-01-01
11	1940	1940 Gáivuotna - Kåfjord	\$kommune_NB	S			2002-01-01	2016-01-01
12	2011	2011 Guovdageaidnu - Kautokeino	\$kommune_NB	S			2002-01-01	2016-01-01
13	2020	2020 Porsanger Porsángu Porsanki	\$kommune_NB	s			2005-01-01	2014-01-01
14	2020	2020 Porsanger - Porsángu - Porsanki	\$kommune_NB	S			2014-01-01	2016-01-01
15	2021	2021 Kárásjohka - Karasjok	\$kommune_NB	S			2002-01-01	2016-01-01
16	2025	2025 Deatnu - Tana	\$kommune_NB	s			2002-01-01	2005-01-01
17	2027	2027 Unjárga Nesseby	\$kommune_NB	S			2005-01-01	2014-01-01

Disse blir også lagt ut på det midlertidige SAS-datasettet Klass_Dubletter.

Hvis vi oppgir en ugyldig dato får vi denne feilmeldingen:

```
ERROR: Bad request. Use the debug option for more info.
```

Hvis til_dato er tidligere enn fra_dato får vi denne feilmeldingen:

```
ERROR: Invalid reply received from the HTTP server. Use the debug option for more info.
```

format_tekst

Vi kan bestemme oss for om vi skal bruke lang eller kort tekst fra Klass. Begge tekstene hentes fra Klass, men bare en av dem brukes som format-tekst. Vi bruker parameteren format_tekst til å bestemme om vi skal ha kort eller lang tekst ut og også til å bestemme om koden skal være med i teksten eller ei. Om vi ikke tar med koden som en del av teksten vil like tekster bli slått sammen. Det lages en liste som viser når dette skjer. Her er et program der vi ikke tar med kommunenummeret som en del av teksten:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  format_prefiks=$kommune,
  format_tekst=3,
  uttrekkstype=json
)
;
```

De som har samme tekst, men forskjellig kode blir nå listet ut slik at vi kan sjekke om det er greit at de med samme tekst slås sammen:

Flere koder med samme tekst for kodelistenummer 131! De forskjellige kodene vil slås sammen med samme tekst. Hvis det ikke ønskes kan du ta med koden som en del av teksten (velg format_tekst = 1 eller 2)

Kode	Tekst	Formatnavn	Indikator for restgruppering og for å beholde sortering	Dato gyldig fra	Dato gyldig til	Dato gyldig fra i valgt periode	Dato gyldig til i valgt periode	antall
0821	Bø	\$kommune_NB	s			2016-01-01	2016-01-01	2
1867	Bø	\$kommune_NB	s			2016-01-01	2016-01-01	2
1515	Herøy	\$kommune_NB	s			2016-01-01	2016-01-01	2
1818	Herøy	\$kommune_NB	s			2016-01-01	2016-01-01	2
0236	Nes	\$kommune_NB	s			2016-01-01	2016-01-01	2
0616	Nes	\$kommune_NB	s			2016-01-01	2016-01-01	2
0441	Os	\$kommune_NB	s			2016-01-01	2016-01-01	2
1243	Os	\$kommune_NB	s			2016-01-01	2016-01-01	2
0713	Sande	\$kommune_NB	s			2016-01-01	2016-01-01	2
1514	Sande	\$kommune_NB	S			2016-01-01	2016-01-01	2
	Uoppgitt	\$kommune_NB	so			2016-01-01	2016-01-01	2
9999	Uoppgitt	\$kommune_NB	S			2016-01-01	2016-01-01	2
0137	Våler	\$kommune_NB	s			2016-01-01	2016-01-01	2
0426	Våler	\$kommune_NB	s			2016-01-01	2016-01-01	2

Her er det ikke greit at vi slår sammen: Bø i Telemark (0821) skal ikke slås sammen med Bø i Nordland (1867) osv. Hvis vi ikke ønsker at de skal slås sammen må teksten endres for disse før formatet lages. Det tar ikke denne SAS-makroen høyde for. Isteden er det en god løsning å ta med koden som en del av teksten for da blir tekstene forskjellige for disse kommunene. Parameteren format_tekst skal i så fall være 1 eller 2.

utds_kodeliste

Vi kan endre navnet på SAS-datasettet med kodelisten som er hentet ut fra Klass. Hvis vi ikke velger noe navn eller det settes til ingenting får det navnet Klass_Kodeliste. Et eksempel:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  utds_kodeliste=kommuner,
  format_prefiks=$kommune,
  uttrekkstype=json
);
```

utds_klassifikasjonInfo

Vi kan endre navnet på SAS-datasettet med informasjon om kodelisten som er hentet ut fra Klass. Hvis vi ikke velger noe navn eller det settes til ingenting får det navnet Klass Info. Et eksempel:

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  utds_kodeliste=kommuner,
  utds_klassifikasjonInfo=kommuner_info,
  format_prefiks=$kommune,
```

```
uttrekkstype=json
);
```

utds_format

Vi kan endre navnet på SAS-datasettet som formatet lages fra. Hvis vi ikke velger noe navn eller det settes til ingenting får det navnet Klass_Format. Et eksempel:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  utds_kodeliste=kommuner,
  utds_format=kommuneformat,
  format_prefiks=$kommune,
  slette_midl=0,
  uttrekkstype=json
);
```

utds_dubletter

Vi kan endre navnet på SAS-datasettet som inneholder eventuelle dubletter fra kodelisten som er hentet ut fra Klass. Hvis vi ikke velger noe navn eller det settes til ingenting får det navnet Klass Dubletter. Et eksempel:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  utds_kodeliste=kommuner,
  utds_klassifikasjonInfo=kommuner_info,
  format_prefiks=$kommune,
  uttrekkstype=json
);
```

liste_dubletter_alle

Når det er dubletter for en kode godtar ikke SAS dette for enkle formater. Dermed må alle koder unntatt en fjernes. Den nyeste blir beholdt. Det lages en liste over de som beholdes. Vi kan lage en liste som inneholder alle dubletter, også den som blir tatt med videre. Disse dublettene legges på et eget SAS-datasett. Her er et eksempel der vi lager en slik liste:

%KodelisteFraKlass

```
(klass_katnr=131,
  format_prefiks=$kommune,
  fra_dato=2002-01-01,
  til_dato=2016-01-01,
  slette_midl=0,
  liste_dubletter_alle=1,
  uttrekkstype=json
)
```

Slik blir listen, den med lengst gyldig dato blir tatt med videre:

Alle dubletter på format-datasettet til kodelistenummer 131 fra valgt tidsrom: 2002-01-01 - 2016-01-01

Obs	Kode	Tekst	Formatnavn	Indikator for restgruppering og for å beholde sortering	Dato gyldig fra	Dato gyldig til	Dato gyldig fra i valgt periode	Dato gyldig til i valgt periode	Antall dubletter
1	0701	0701 Horten	\$kommune_NB	s			2002-06-01	2016-01-01	2
2	0701	0701 Borre	\$kommune_NB	s			2002-01-01	2002-06-01	2
3	1736	1736 Snåsa	\$kommune_NB	S			2002-01-01	2013-01-01	2
4	1736	1736 Snåase - Snåsa	\$kommune_NB	S			2013-01-01	2016-01-01	2
5	1739	1739 Raarvihke - Røyrvik	\$kommune_NB	S			2014-06-01	2016-01-01	2
6	1739	1739 Røyrvik	\$kommune_NB	S			2002-01-01	2014-06-01	2
7	1849	1849 Hamarøy	\$kommune_NB	S			2002-01-01	2013-01-01	3
8	1849	1849 Hamarøy - Hábmer	\$kommune_NB	S			2014-03-01	2016-01-01	3
9	1849	1849 Hábmer - Hamarøy	\$kommune_NB	S			2013-01-01	2014-03-01	3
10	1850	1850 Divtasvuodna - Tysfjord	\$kommune_NB	s			2014-03-01	2016-01-01	3
11	1850	1850 Divtasvuodna Tysfjord	\$kommune_NB	s			2013-01-01	2014-03-01	3
12	1850	1850 Tysfjord	\$kommune_NB	S			2002-01-01	2013-01-01	3
13	1920	1920 Loabák - Lavangen	\$kommune_NB	S			2014-12-01	2016-01-01	2
14	1920	1920 Lavangen	\$kommune_NB	S			2002-01-01	2014-12-01	2
15	1939	1939 Storfjord	\$kommune_NB	S			2002-01-01	2014-03-01	2
16	1939	1939 Storfjord - Omasvuotna - Omasvuono	\$kommune_NB	S			2014-03-01	2016-01-01	2
17	2020	2020 Porsanger - Porsángu - Porsanki	\$kommune_NB	S			2004-01-01	2016-01-01	2
18	2020	2020 Porsanger	\$kommune_NB	s			2002-01-01	2004-01-01	2

Disse dublettene er fjernet fra format-datasettet til kodelistenummer 131 fra valgt tidsrom: 2002-01-01 - 2016-01-01 Enkle Sas-formater tillater ikke overlappende koder (dubletter)

Dublettene med tidligst til-dato blir slettet

Obs	Kode	Tekst	Formatnavn	Indikator for restgruppering og for å beholde sortering	Dato gyldig fra	Dato gyldig til	Dato gyldig fra i valgt periode	Dato gyldig til i valgt periode
1	0701	0701 Borre	\$kommune_NB	S			2002-01-01	2002-06-01
2	1736	1736 Snåsa	\$kommune_NB	s			2002-01-01	2013-01-01
3	1739	1739 Røyrvik	\$kommune_NB	S			2002-01-01	2014-06-01
4	1849	1849 Hábmer - Hamarøy	\$kommune_NB	S			2013-01-01	2014-03-01
5	1849	1849 Hamarøy	\$kommune_NB	S			2002-01-01	2013-01-01
6	1850	1850 Divtasvuodna Tysfjord	\$kommune_NB	S			2013-01-01	2014-03-01
7	1850	1850 Tysfjord	\$kommune_NB	S			2002-01-01	2013-01-01
8	1920	1920 Lavangen	\$kommune_NB	S			2002-01-01	2014-12-01
9	1939	1939 Storfjord	\$kommune_NB	S			2002-01-01	2014-03-01
10	2020	2020 Porsanger	\$kommune_NB	S			2002-01-01	2004-01-01

For kommune 0701 blir den med tekst 0701 Horten beholdt.

utds_dubletter_alle

Vi kan selv bestemme navnet på SAS-datasettet som inneholder alle dublettene. SAS-datasettet blir kun laget om parameteren liste_dubletter_alle er satt til 1. Her er et eksempel:

```
(klass_katnr=131,
format_prefiks=$kommune,
fra_dato=2002-01-01,
```

```
til_dato=2016-01-01,
slette_midl=0,
liste_dubletter_alle=1,
utds_dubletter_alle=alledubl,
uttrekkstype=json
)
```

utds_restgruppering

Vi kan endre navnet på SAS-datasettet med restgruppering som lages. Hvis vi ikke velger noe navn eller det settes til ingenting får det navnet Klass_Restgruppering. Et eksempel:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  utds_kodeliste=kommuner,
  utds_restgruppering=kommuner_restgruppe,
  format_prefiks=$kommune,
  slette_midl=0,
  uttrekkstype=json
);
```

behold_restgruppe_data

Normalt vil SAS-datasettet som inneholder informasjon om restgruppering bli slettet. Om vi ikke ønsker det gir vi behold restgruppe data verdien 1:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  behold_restgruppe_data=1,
  format_prefiks=$kommune,
  uttrekkstype=json
);
```

behold_info_data

Normalt vil SAS-datasettet som inneholder informasjon om kodelisten bli beholdt. Om vi ikke ønsker det gir vi behold_info_data en annen verdi enn 1:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  behold_info_data=0,
  format_prefiks=$kommune,
  uttrekkstype=json
);
```

behold_kodeliste_data

Normalt vil SAS-datasettet som inneholder kodelisten bli beholdt. Om vi ikke ønsker det gir vi behold_kodeliste_data en annen verdi enn 1:

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  behold_info_data=0,
  behold_kodeliste_data=0,
  format_prefiks=$kommune,
  uttrekkstype=json
);
```

lag_format

Hvis vi bare ønsker å hente ut kodelisten og ikke lage et format fra den, gir vi lag_format en annen verdi enn 1:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  lag_format=0,
  uttrekkstype=json
);
```

format_lib

Hvis vi ikke angir format_lib legges formatet som lages ut på det temporære biblioteket work. Ønsker vi å legge det på et permanent bibliotek må dette først defineres med en Libname-setning med libref Library. Deretter kan vi legge til parameteren format_lib:

%KodelisteFraKlass

```
(klass_katnr=131,
  fra_dato=2016-01-01,
  format_lib=library,
  uttrekkstype=json
);
```

Hvis mappen Library ikke er definert i SAS får vi denne feilmeldingen:

```
ERROR: Libref LIBRARY is not assigned.
```

adresse

Denne vil normalt bare brukes av de som driver med utvikling og testing. For de som har tilgang til utviklings- eller test-versjonen av Klass kan adressen endres. Vær klar over at katalognumrene kan være forskjellige fra utvikling til test og til produksjonsversjonen av Klass. Her er et eksempel der vi henter kommunekatalogen fra utviklingsversjonen:

```
(adresse=http://al-klass-app-u1.ssb.no:8080/rest/v1,
  klass_katnr=262,
  fra_dato=2016-01-01,
  format_prefiks=$kommune,
  uttrekkstype=json
);
```

Testversjonen av klass har adressen http://klass-test.ssb.no/rest/v1 og QA-versjonen (Quality Assurance) har adressen http://beta-data.ssb.no/api/klass/v1. Disse kan endre seg.

lengde_kode

Koden som hentes fra Klass vil kunne være opptil 15 tegn uten at vi trenger å bry oss om det. Er den lengre, må vi øke lengden på den. Vi kan også velge en kortere lengde, men da risikerer vi å velge en som er for kort og det vil gi feil resultater. Ved uttrekkstype JSON tas det ikke hensyn til denne da SAS velger en riktig lengde ved konvertering fra JSON-format til SAS-format.

lengde_tekst

Teksten som hentes fra Klass vil kunne være opptil 512 tegn uten at vi trenger å bry oss om det. Er den lengre, må vi øke lengden på den. Vi kan også velge en kortere lengde, men da risikerer vi å velge en som er for kort og det vil gi feil resultater. Ved uttrekkstype JSON tas det ikke hensyn til denne da SAS velger en riktig lengde ved konvertering fra JSON-format til SAS-format.

lengde_tekst_kort

Kort-teksten som hentes fra Klass vil kunne være opptil 256 tegn uten at vi trenger å bry oss om det. Er den lengre, må vi øke lengden på den. Vi kan også velge en kortere lengde, men da risikerer vi å velge en som er for kort og det vil gi feil resultater. Ved uttrekkstype JSON tas det ikke hensyn til denne da SAS velger en riktig lengde ved konvertering fra JSON-format til SAS-format.

lengde_record

Maksimal lengde for de enkelte radene vi henter fra Klass er satt til 9999. Det vil normalt være tilstrekkelig, men hvis dette ikke holder kan vi øke det. Vi kan også redusere lengdene, men det vil vanligvis ikke være noe poeng. Her er et eksempel der vi endrer lengder på kode, tekst og hele recorden:

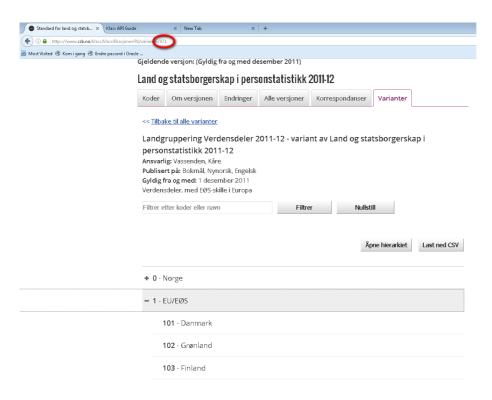
%KodelisteFraKlass

```
(klass_katnr=131,
  format_prefiks=$kommune,
  fra_dato=2002-01-01,
  til_dato=2016-01-01,
  lengde_record=128,
  lengde_tekst=44,
  lengde_tekst_kort=33,
  lengde_kode=4
)
:
```

Ved uttrekkstype JSON tas det ikke hensyn til denne da den er irrelevant ved konvertering fra JSON-format til SAS-format.

variant

Det finnes en mulighet for å lage varianter av en kodeliste. Hver variant har sitt eget klass-nr og det finner vi bakerst i adressen når vi har funnet riktig variant. Her er en variant av land:



For varianter skal vi ikke oppgi gyldighetsdatoer da varianten er knyttet til en spesifikk kodeliste.

For å hente en variant istedenfor en vanlig kodeliste må vi ta med parameteren variant og gi den kode 1. Her er land for befolkningsstatistikk:

%KodelisteFraKlass

```
(klass_katnr=971,
  format_prefiks=$land,
    slette_midl=0,
    variant=1,
    uttrekkstype=json
) ;
run ;
```

Her er tilsvarende for land på engelsk for utenrikshandel:

%KodelisteFraKlass

```
(klass_katnr=603,
  format_prefiks=$country,
  slette_midl=0,
  variant=1,
  klass_spraak=EN
) ;
run ;
```

uttrekkstype

Når vi henter ut en kodeliste fra Klass vil vi kunne velge hva slag format uttrekket skal komme i. Det vanlige er en semikolonseparert fil (csv-fil). Vi kan eventuelt velge at uttrekket skal være en json-fil. Resultatet av uttrekket har forskjellig tegnsett. Csv-formatet bruker et gammelt tegnsett kalt latin1, mens json bruker det mer moderne utf8-formatet. I de fleste tilfeller vil det ikke bli noen forskjell om

vi velger csv eller json, men csv-formatet har problemer med enkelte spesielle tegn, f.eks. š i Kárášjohka fra kommunekatalogen. Med csv-formatet blir teksten slik: Kárá?johka - Karasjok. Her må vi bruke uttrekkstype json for å få ut riktige tegn: Kárášjohka - Karasjok. Slik får vi ut en riktig kommunekatalog for 2018:

%KodelisteFraKlass (klass_katnr=131, fra_dato=2018-01-01, format_prefiks=\$, format_suffiks=kommune, uttrekkstype=json);

For å unngå problemer med konvertering av tegn bør uttrekkstype json brukes. Den er ikke satt som standard da parameteren uttrekkstype ikke var med i den opprinnelige makroen.

tegnsett

Tegnsettet som fulgte de første datamaskinene hadde et begrenset antall tegn, 256. For å få plass til tegn på flere språk måtte tegnsettet utvides. De vanligste tegnsettene nå er utf8 og utf16. Dessuten bruker vi latin1 i SAS. Klass bruker utf8. Dermed må tegnsettet konverteres fra Klass til SAS, fra utf8 til latin1. For uttrekkstype JSON kan vi velge hva det skal konverteres til. Dette bør være det tegnsettet som SAS-sesjonen bruker og hos oss er det latin1. Det betyr at vi normalt kan lå være å bruke denne parameteren.

Hvis SAS-sesjonen er satt opp med et annet tegnsett kan dette velges. Se SAS-dokumentasjonen på internett for mer info (f.eks.

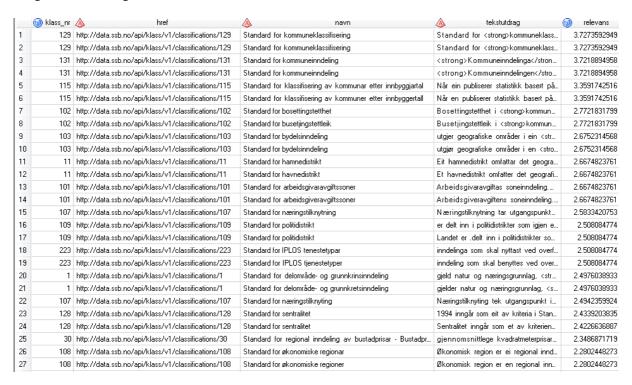
http://documentation.sas.com/?docsetId=nlsref&docsetTarget=nlr7pnb91
iybs9nlhgvsj7q09srd.htm&docsetVersion=9.4&locale=no).

For uttrekkstype CSV gjøres det en konvertering til et latin-tegnsett av Klass-applikasjonen, men den virker altså ikke for alle tegn (se over om uttrekkstype).

Finne nummeret til en kodeliste

I og med at vi trenger å vite nummeret på kodelisten for å kunne hente den ut bør det være lett å søke seg frem til dette. Denne kjører vi lokalt på Windows inntil vi har SAS versjon 9.4. Vi kan finne kodelister slik:

Vi angir søkestrengen i %let soek= og maksimalt antall kodelister som skal vises med %let maksant=. Programmet over gir en liste som dette:



Nummeret vi er på jakt etter finner vi i variabelen klass_nr. SearchScore skal bli større jo mer relevant søkeordet er for en klassifikasjon og vi ser at kommuneinndelingen anses mest relevant i dette søket. Variabelen tekstutdrag inneholder en tekst funnet i klassifikasjonen som inneholder søkeordet.

Søker vi etter tekster som inneholder æ, ø eller å kan det hende at det ikke kommer noe resultat. Da kan vi prøve å bytte ut disse tegnene på følgende måte:

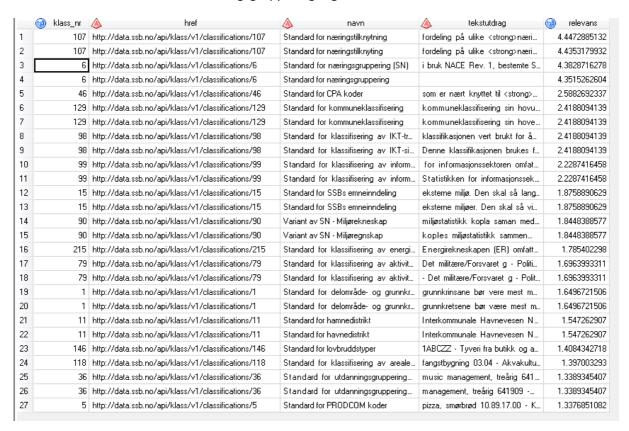
```
æ %C3%A6
ø %C3%B8
å %C3%A5
```

Her er et eksempel der vi søker etter næring:

```
%let soek=n%C3%A6ring;
%let maksant=55;
```

```
filename klasslst url
"http://data.ssb.no/api/klass/v1/classifications/search?query=&soek.%nrstr(
&size) = & maksant." debug;
libname klass json fileref=klasslst ;
proc sql;
   create table klassoversikt as
   select input(scan(t2.href,7,'/'),9.) as klass_nr
label='Klassifikasjonsnummer',
          t2.href label='Adresse',
          kcvt(t1.name, 'utf8', 'latin1') as navn label='Navn',
          kcvt(t1.snippet,'utf8','latin1') as tekstutdrag label='Tekst som
inneholder søkebegrepet' length=9999,
          t1.searchscore as relevans label='Relevans i forhold til søkeord
(jo større tall jo større relevans)'
      from klass._EMBEDDED_SEARCHRESULTS as t1 inner join klass._LINKS_SELF
       on (t1.ordinal_searchResults = t2.ordinal_self)
      order by relevans desc;
quit;
```

Resultatet viser at standard for næringsgruppering regnes som det mest relevante resultatet:



Søket kan gi flere rader som ikke har teksten i de kolonnene som vises da søket utføres på andre kolonner (som ikke vises) også.

Spesielle forhold

Denne makroen henter ut en kodeliste som en semikolonseparert fil som så leses inn til SAS. Informasjonen om klassifikasjonene finnes ikke i den semikolonseparerte filen og den hentes som en json-fil isteden. Det hentes ikke noen slik fil når vi henter ut en variant.

Det lages noen temporære SAS-datasett i makroen. Finnes disse fra før blir de overskrevet. Disse SAS-datasettene brukes: Klass_Kodeliste, Klass_Info, Klass_Format, Klass_Dubletter og Klass_Restgrupper (hvis restgruppering er valgt). Alle disse kan gis andre navn om ønskelig.

Det defineres noen libnames for å lese json-filer. Det er klassinf som inneholder informasjonsdatasettet i json-format og, når uttrekkstype json er valgt, kls_json som inneholder kodelisten i json-format. Hvis disse er tilordnet fra før vil tilordningene bli fjernet.