

Aegridade analüüs

Praktikum nr. 1, 2016, SAS tarkvara kasutamisest

Andmete lugemine SAS keskkonda. Protseduur sobiva skripti kirjutamiseks on järgmine:

1. Anname andmestikule nime ja alustame sisselugemise protseduuri käsuga `data nimi;`
2. Kirjeldame sisendfaili käsuga `infile failinimi suvandid;`, kus tähtsamateks suvanditeks on:
`firstobs=n` - andmed hakkavad failis alates n-dast reast
`delimiter="s"` - vaikumisi on eraldajateks tühikud. Kui `s` koosneb mitmest sümbolist, siis igaüks neist loetakse eraldajaks. Kui eraldajaks on tabulaator, siis selle saab kirjeldada kujul `delimiter='09'x` (selgitus: nähtamatu sümboli saab ette anda tema ascii koodi kaudu 16-süsteemis, andes selle ette ülakomade vahel sõnena ning kirjutades kohe taha `x`, mis tähistab 16-süsteemi. Tabulaatori koodiks on 9
`dsd` - vajalik, kui kaks järjestikust eraldajat (näiteks koma) tähendavad, et mingi muutuja väärtus on puudu.
3. andmestikku kirjutatavate muutujate määramine: käsuga `keep nimekiri;`, kus `nimekiri` on tühikutega eraldatud muutujate nimed
4. Ajainfo jaoks kasutatava abimuutuja tekitamine. Kuupäevadega tegelemiseks on SAS-is palju võimalusi. Aegridadega tegelemisel on käesoleva kursuse jaoks kõige lihtsam teha andmete vaatlusel kirja vaatluste alguskuupäev ja intervall ning tekitada kuupäevainfo neid kasutades, ignoreerides muud failis olevat kuupäevainfot. Selleks aga tekitame abimuutuja, mis loendab vaatlusi käsuga `retain abi 0;`. Selle muutuja abil saame arvutada jooksva vaatluse kuupäeva käsuga
`date=intnx(interval,'alguskuup'd,abi),`
kus intervall on tavaliselt `'day'`, `'weekday'` (5-päevane nädal, laupäev ja pühapäev jäetakse vahele), `'month'`, `'qtr'` või `'year'` ning `alguskuup` on alguskuupäev kujul 02feb07 (st päeva number, inglisekeelse kuunimetuse kolm tähte ning aastanumbri kaks viimast kohta. Siinjuures tuleb muutuja `abi` väärtust suurendada pärast igat `output;` käsku ning ajas tagurpidi oleva andmestiku korral tuleb kasutada viimast kuupäeva ning muutuja `abi` väärtusi kahandada.
5. Määrame andmete vaatamiseks aja näitamise formaadi käsuga `format date;`. Kvartaalsete andmete sobivaks formaadiks on `yyqq4.`, igakuiste andmete korral `mmmyyc5.`, igapäevaste korral aga `date7.`. Kasutada on võimalik palju muid formaate, mille kohta leiab infot SAS spikrit kasutades.
6. Kasutame andmete sisselugemiseks `input` käsku. Kui igas faili reas on ainult üks aegrea väärtus, siis `input` käsu taha kirjutame loetelu väljanimedest, mis on tühikutega eraldatud, kusjuures iga tekstilise välja nime taga tuleb pärast tühikut kirjutada `$`. Seejärel arvutame kuupäeva, kasutame `output` käsku andmete andmestikku kirjutamiseks ning järgmisel real muudame abimuutuja väärtust. Kui aga igas reas on mitu ühe aegrea väärtust, siis on üks võimalik tegevuseeskiri järgmine: 1) loeme sisse eelnevad muutujad kujul `input nimekiri @;`, kus `@` tähendab, et järgnevad `input` käsud jätkavad samast faili reast lugemist. 2) Kirjutame tsükli, mis loeb sisse aegrea väärtused:

```
do i=1 to n;  
  input nimi @;
```

```

    date=intnx(interval,'alguskuup'd,abi);
    output;
    abi=abi+1;
end;

```

Siin n on vaatluste arv reas.

7. Kirjutame lõppu `run;` ning jooksutame eelnevat koodi. Andmestik peaks seejärel olema valmis.

8. Puuduvate andmetega kirjade eemaldamine andmestiku lõpust. Selleks, et olevasoleva andmestiku põhjal uut teha, võib kasutada andmesammu (*data step*) kujul

```

data andmestiku2_nimi;
set andmestiku1_nimi;

```

sel juhul tehakse järgnevad käsud andmestiku nr 1 iga rea korral läbi ja tulemused kirjutatakse andmestikku nr 2. Kui kasutada mõlema jaoks sama nime, siis kirjutatakse vanad andmed üle (või tekitatakse sama nimega tabel, kus on lisaks vanadele ka uued muutujad). Selleks, et eemaldada kõik kirjed, kus mingi numbrilise muutuja väärtus puudub, sobib käsk

```

if muutuja^=.;

```

eelneva käsu tulemusena kirjutatakse väljundandmesse ainult kirjed, kus vastaval muutujal on väärtus olemas. Andmestiku lõpust saab kuupäeva järgi kirjeid eemaldada kujul `if date<'kuupäev'd;` kus kuupäev peaks olema hilisem kui viimane vaatlus ja varasem rea lõpus olevate puuduvate vaatluste kuupäevadest. Siin on tähtis teada, et näiteks kvartaalsete andmete korral loetakse vaikselt kuupäevaks kvartali esimese kuu esimest kuupäeva.

9. Selleks, et andmete analüüs toimuks õigesti, on vaja andmestik viia ajas kavava järjestuse kujule. Kui vaatlustel on juures õiges järjestuses kuupäevad, siis sobib selleks protseduur `sort`, mille lihtsaim kuju on järgmine:

```

proc sort data=andmestiku_nimi;
by date;
run;

```

Siin on eeldatud, et andmestikus on väli nimega `date`, kus on kuupäevainfo. See tegevus on näiteks vajalik Cisco andmete korral.

10. Graafiku tekitamiseks on `gplot` protseduur, mille süntaks lihtsaimal kujul on järgmine:

```

proc gplot data=andmestiku_nimi;
symbol i=spline v=dot h=0.3;
plot y_muutuja*x_muutuja;
run;

```

Mõned näited sisselugemisest ja graafiku vaatamisest:

```

libname aegread "h:/aegread_praktikum";
*määrab nime andmekastale ning selle kausta tegeliku asukoha;
data aegread.kinnisvara;
*andmete tekitamine ja teisendamine käib nn "data step" abil.
esimesel real antakse uue andmestiku nimi kujul
kausta_nimi.andmestiku_nimi. Kui kasuta nime ei anta,
siis läheb andmestik ajutisse kasuta nimega work;

```

```

infile "h:/aegread_praktikum/kv015m_15.csv" firstobs=1 delimiter="09"x dsd;
*sisendfail, mitmendast reas andmed ning mis on andmeväljade eraldajaks.
Tabulaatori puhul antakse klahvi kood 16-süsteemis, seda tähistab see
x pärast koodi "09". tavalise sümboli puhul tuleb lihtsalt sümbol kirjutada
jutumärkide või ülakomade vahele.;
keep date summa;
*me võime defineerida mitmeid muutujaid ning andmevälju, kuid väljundisse
ei ole kõiki vaja. Käsu keep taga näitame, mida tulemusandmestikku kirjutada;
retain abi 0; *abimuutuja kuupäeva tekitamiseks;
date=intnx("qtr","01jan1997"d,abi);
*giga rea sisselugemisel tekitame uue kuupäeva,
mis on antud kuupäevast suurusega abi määratud kvartalite võrra kaugemal;
input aasta $ kvartal $ summa; *näitame, mitu ja mist tüüpi välja
($ nime järel tähistab tekstilist välja) tuleb igast failireast sisse lugeda;
output; *kirjutame hetkeseisu väärtused andmestikku;
format date yyqc6.; *see on mugavuse jaoks. Määrab,
mis kujul kuupäevi meile näidatakse, kui me andmeid vaatame;
abi=abi+1; *suurendama muutuja abi väärtust enne järgmise kirje lugemist;
run;
*lõpus on kaks puuduvat vaatlust, eemaldame need;
data aegread.kinnisvara;
set aegread.kinnisvara;
if summa^=.;* ^= tähendab mittevõrdumist ja punkt tähistab puuduvaid andmeid;
*sisendandmestik käiakse läbi ridade kaupa.
Kui tingimus ei ole täidetud, ei tehta vastava reaga midagi.;
*kui tingimus on täidetud, siis täidetakse järgnevad käsud (mida pole)
ning lõpuks kirjutatakse tulemused väljundandmestikku. Praegusel juhul kopeeritakse
lihtsalt tingimusele vastavad read väljundandmestikku.;
run;
*kirjed on ajaliselt õiges järjekorras;
*kinnisvara andmete visuaalne vaatlus;
proc gplot data=aegread.kinnisvara;*data taga näidatakse kasutatav andmestik;
symbol i=spline v=dot h=0.3;
*i=spline käsib joonistada pideva joone, st andmepunktid ühendatakse graafikul joonega.
v=dot ütleb, et andmestikus olevatele väärtustele vastavad punktid tuuakse joonisel
välja suurema pallikesena.
h=0.3 abil saab näidata pallikese suurust;
plot summa*date;*y-koordinaat enne ja x-koordinaat pärast;
run;

*****;
*USA töötuse määra andmestik;
data aegread.unemployment;
infile "h:/aegread_praktikum/unemployment2015.dat" firstobs=14 delimiter=', ' dsd;
*aegrea andmed hakkavad kuueteistkümnendast reast, eraldatud komaga;
*tegelikult võiks alustada ka 17-dast reast, sest kuueteistkümnendas
ei ole kuude kohta andmeid;
keep date rate;
retain abi 0;
*gigas reas on kõigepealt aastanumber, mis ei kuulu aegritta.
Loeme selle sisse ja jääme samale reale;
input aasta @;
* edasi tulevad töötuse määra erinevatele kuudele vastavad väärtused, mida on 12 tükki;
do i=1 to 12;
input rate @;
date=intnx("month","01jan1948"d,abi);*kuude kohta algavad andmed 48-st aastast;

```

```

output;
abi=abi+1;
end;
format date mmyyc7.;
run;
*lõpus on puuduvad kirjed, eemaldame need;
data aegread.unemployment;
set aegread.unemployment;
if rate^=.;
run;
* kirjed on ajaliselt õiges järjekorras, seetõttu sorteerima ei pea.;
*vaatame töötuse määra sisselugemise tulemust jooniselt;
proc gplot data=aegread.unemployment;
symbol i=spline v=dot h=0.3;
plot rate*date;
where date>"01sep2010"d; *ajaperiood on väga pikk, vaatame ainult viimast viite aastat;
run;

```

```

*****
Cisco andmestik
*****;
*loeme sisse dividendidega kohandatud sulgemishindade andmed failist cisco2015.csv;
data aegread.cisco;
infile "h:/aegread_praktikum/cisco2015.csv" firstobs=2 delimiter=',' dsd;
*andmed hakkavad teisest reast (esimeses on väljade nimed), eraldatud on komaga;
keep date hind;*meid huvitab praegu ainult kohandatud sulgemishindade aegrida,
ülejäanu on ebaoluline;
retain abi 0;
date=intnx('weekday','29aug2014'd,abi);*kauplemine toimub tööpäevadel;
input tmp1 $ tmp2 tmp3 tmp4 tmp5 tmp6 hind;
*igast reast peame lugema välju kuni huvipakkuva tulbani (mis on seitsmes);
output;
format date date9.;
abi=abi-1; *aeg on tagurpidi;
run;
*puuduvaid kirjeid ei ole;
*andmed on ajas tagurpidi, nii et pöörame ümber;
proc sort data=aegread.cisco;
by date;
run;
*joonistame graafiku;
proc gplot data=aegread.cisco;
symbol i=spline v=dot h=0.3;
plot hind*date;
run; *joonistame graafiku;

```

```

***tarbijahinna indeksi sisselugemine;
data aegread.indeks;
infile "h:/aegread_praktikum/IAO2s.csv" firstobs=5 delimiter=";" dsd;
*aegrea andmed hakkavad viiendast reast, eraldatud semikooloniga;
keep date indeks;
retain abi 0;
*igas reas on kõigepealt kaks aegritta mittekuuluvat teksti kujul olevat kirjet.
@ tähendab, et pärast nende sisselugemist töötame edasi sama andmereaga;
input tmp1 $ tmp2 $ @;

```

```

do i=1 to 12;
* edasi tulevad tarbijahinna indeksi erinevatele kuudele vastavad väärtused,
mida on 12 tükki;
input indeks@; *loeme järgmise numbri ja jääme samale reale;
date=intnx("month","1jan98"d,abi);*arvutama sellele vastava kuupäeva;
output; *kirjutame indeksi ja kuupäeva väljundandmestikku;
abi=abi+1; *järgmine kord peab kuupäev ühe kuu võrra hilisem olema;
end;
format date mmyyc7.;
run;
*eemaldame puuduvad kirjed;
data aegread.indeks;
set aegread.indeks;
if indeks^=.;
run;
*kontrollime visuaalselt;
proc gplot data=aegread.indeks;
symbol i=spline v=dot h=0.3;
plot indeks*date;
run;

```

Globaalsed trendid. Nagu ka R tarkvaras, on SAS puhul võimalik kasutada mitmeid protseduure leitavate parameetrite suhtes lineaarsete ja ka mittelineaarsete trendikõverate leidmiseks. Vaatleme siiski ainult parameetrite suhtes lineaarse trendi leidmist protseduuriga arima, mida me kasutame ka keerulisemate mudelite sobitamiseks. Kui otsitavas kujus $T(t) = c_0 + c_1 \cdot f_1(t) + c_2 \cdot f_2(t) + \dots + c_m \cdot f_m(t)$ olevad funktsioonid f_i on mingid deterministlikud funktsioonid ja mitte mingi juhusliku protsessi väärtused, siis on võimalik tegevuste järjekord selline:

- Tekitame ajutise andmestiku, kus on lisaks meid huvitavale aegreale trendifunktsioonide väärtused. Selleks on võimalik kasutada näiteks andmesammu (*data step*), andes sisendiks ette olemasoleva andmestiku (*set=sisendandmed*) ning arvutades vajaminevad lisamuutujad, kasutades selleks SAS sisemist muutujat `_n_`, mis sisaldab jooksva rea numbrit.
- Kasutame protseduuri arima järgmiselt:

```

proc arima data=sisendandmed;
identify var=aegrida crosscorr=(F1 F2 ... Fm);
estimate input=(F1 F2 ... Fm);
forecast id=date lead=0 out=tulemustabel;
run;
quit;

```

Siin peab `var=` taha kirjutama tulba nime, kus on analüüsitud aegrida ning F1,F2 jne asemel kirjutama tulpade nimed, kus on funktsioonide f_i , $i = 1, \dots, m$ väärtused. Pärast sellise käsu jooksumist on tabelis nimega tulemustabel esialgse rea väärtused, kuupäevad (eeldus on, et nimega date on sisendandmetes olemas), tulp nimega forecast leitud trendifunktsiooni väärtustega, tulp nimega residual, kus on trendi mahalahutamisel jääv osa ja veel mõned tulbad. Seega on lihtne teha huvipakkuvaid jooniseid. Hinnatud parameetrite väärtuseid saab vaadata tulemuste (Results) aknas