SAS / R / Gretl

Zdroje dat

- https://www.quandl.com

Import dat

```
DATA fitness
INPUT VEK VAHA DOBA_CV POC_PULZ CV_PULZ MAX_PULZ SPOTR_KYSL SKUPINA POHLAVI
$ ID;
DATALINES;
57 73.37    12.63    58    174    176    39.407    2    M    1
54 79.38    11.17    62    156    165    46.08    2    M    2
52 76.32    9.63    48    164    166    45.441    2    M    3
50 70.87    8.92    48    146    155    54.625    2    M    4
51 67.25    11.08    48    172    172    45.118    2    M    5
54 91.63    12.88    44    168    172    39.203    2    M    6
51 73.71    10.47    59    186    188    45.79    2    M    7
57 59.08    9.93    49    148    155    50.545    2    M    8
49 76.32    9.4    56    186    188    48.673    2    M    9
48 61.24    11.5    52    170    176    47.92    2    M    10
52 82.78    10.5    53    170    172    47.467    2    M    11
44 73.03    10.13    45    168    168    50.541    1    M    12
45 87.66    14.03    56    186    192    37.388    1    M    13
45 66.45    11.12    51    176    176    44.754    1    M    14
47 79.15    10.6    47    162    164    47.273    1    M    15
54 83.12    10.33    50    166    170    51.855    1    M    16
49 81.42    8.95    44    180    185    49.156    1    M    17
51 69.63    10.95    57    168    172    40.836    1    M    18
51 77.91    10    48    162    168    46.672    1    Z    19
48 91.63    10.25    48    162    164    46.774    1    Z    20
49 73.37    10.08    76    168    168    50.388    1    Z    21
44 89.47    11.37    62    178    182    44.609    0    Z    22
40 75.07    10.07    62    185    185    45.313    0    Z    23
44 85.84    8.65    45    156    168    54.297    0    Z    24
42 68.15    8.17    40    166    172    59.571    0    Z    25
38 89.02    9.22    55    178    180    49.874    0    Z    26
47 77.45    11.63    58    176    176    44.811    0    Z    27
40 75.98    11.95    70    176    180    45.681    0    Z    28
43 81.19    10.85    64    162    170    49.091    0    Z    29
44 81.42    13.08    63    174    176    39.442    0    Z    30
38 81.87    8.63    48    170    186    60.055    0    Z    31
;
PROC PRINT; RUN;
```

p<0,05 - zamitame H0
H0: ma normal

Data mining

velke soubory

```
data work.prestupky2;


proc univariate data=work.prestupky2 normal plot;
histogram pokuta/kernel normal;
qqplot pokuta/normal (mu=est sigma=est);
var pokuta;
run;


proc boxplot data=work.prestupky2;
plot pokuta*pohlavi;
plot pokuta*pohlavi / boxstyle=schematic;
plot pokuta*pohlavi / notches;
run;


proc univariate data=work.prestupky2 winsor=2;
var pokuta;
run;


proc means data=work.prestupky2 mean clm maxdec=2;
var pokuta;
run;


proc univariate data=work.prestupky2 cibasic;
var pokuta;
run;

proc univariate data=work.prestupky2 cibasic mu0=1335;
var pokuta;
run;
```

male soubory

```
data stat;
input vyuka @@;
datalines
```

```
;
90 85 98 87 65 88 93 85 97 103
;


proc univariate data=stat normal plot;
histogram vyuka/kernel normal;
qqplot vyuka/normal (mu=est sigma=est);
var vyuka;
run;
```

3. cv

```
proc reg data=fitness
model spotr_kysl = doba_cv;
plot spotr_kysl*doba_cv;
plot r.*p.;
symbol v=star;
run;

proc reg data=work.fitness;
model spotr_kysl = doba_cv/r influence spec;
plot spotr_kysl*doba_cv;
plot r.*p.;
symbol v=star;
run;

proc reg data=work.fitness;
model spotr_kysl = doba_cv vek/r vif influence spec;
plot r.*p.;
symbol v=star;
run;
```

4. cv

```
proc boxplot data=work.teploty;
plot vysledky*mesic;
run;

proc glm data=work.teploty;
class mesic;
model vysledky=mesic;
means mesic/hovtest tukey scheffe lsd sidak;
run;

proc npar1way data=work.teploty wilcoxon;
class mesic;
var vysledky;
run;
```

5. cv

```
## pearson chi squared

data souhlas;
input vzdelani $ prirazka $ pocet @@;
datalines;
ano ano 50 ano ne 7 ano nevim 11
ne ano 14 ne ne 23 ne nevim 20
;

proc freq data=souhlas;
tables vzdelani*prirazka /expected chisq measures norow nocol nopercent;
weight pocet;
run;




## Fischer (pokud pearson varuje, ze 33% cetnosti je < 5)


data zakon;
input zmena $ nakup $ pocet @@;
datalines;
ano denne 27 ano nekolikrat_t 79 ano jednou_t 13 ano jednou_14 2
ne denne 38 ne nekolikrat_t 79 ne jednou_t 24 ne jednou_14 3
;

proc freq data=zakon;
tables zmena*nakup/expected chisq measures norow nocol nopercent exact;
weight pocet;
run;




chisq p-value H0: neexistuje zavislost
lambda asymetric C|R = %



data zkouska;
input skola $ splneno $ pocet @@;
datalines;
gympl ano 45 ss ano 22 uc ano 7
gympl ne 7 ss ne 10 uc ne 9
;

proc freq data=zkouska;
tables skola*splneno/expected chisq measures norow nocol nopercent;
```

```
weight pocet;
run;







data semena;
input osetreno $ vyklicilo $ pocet @@;
datalines;
ne ano 70 ne ne 30
ano ano 130 ano ne 130
;

proc freq data=semena;
tables osetreno*vyklicilo /expected chisq measures norow nocol nopercent;
weight pocet;
run;
```

t-test

```
proc ttest data=mesta h0=1335;
run;
```

Procedury ke zkousce

1. cv

```
data work.prestupky2;
proc univariate data= work.prestupky2 normal plot;
histogram body/kernel normal;
qqplot body/normal (mu= est sigma= est);
var body;
run;

data work.prestupky2;
proc boxplot data= work.prestupky2;
plot body*pohlavi/boxstyle=schematic;
plot body*pohlavi/notches;
run;

data work prestupky2;
proc univariate data= work.prestupky2 trimmed=2;
var body;
run;

data work prestupky2;
proc univariate data= work.prestupky2 winsorized=2;
```

```
var body;
run;

data work prestupky2;
proc means data= work.prestupky2 mean cv clm maxdec=2;
var body;
title "Interval spolehlivosti pro průměr";
run;

data work prestupky2;
proc univariate data= work.prestupky2 cibasic;
var body;
title "Basic Confidence Limits Assuming Normality - IS pro základní popisné statistiky";
run;

data stat;
input vyuka @@;
datalines;
98 79 88 64 80 92 67 88 90 60 63 67
;
proc univariate data= stat normal plot;
histogram vyuka/kernel normal;
qqplot vyuka/normal (mu= est sigma= est);
var vyuka;
run;
```

2. cv

```
data fitness;
input vek doba_cv spotr_kysl;
datalines;
57 12.63    39.407
54 11.17    46.08
52 9.63    45.441
50 8.92    54.625
51 11.08    45.118
54 12.88    39.203
51 10.47    45.79
57 9.93    50.545
49 9.4    48.673
48 11.5    47.92
52 10.5    47.467
44 10.13    50.541
45 14.03    37.388
45 11.12    44.754
47 10.6    47.273
54 10.33    51.855
```

```
49 8.95    49.156
51 10.95    40.836
51 10    46.672
48 10.25    46.774
49 10.08    50.388
44 11.37    44.609
40 10.07    45.313
44 8.65    54.297
42 8.17    59.571
38 9.22    49.874
47 11.63    44.811
40 11.95    45.681
43 10.85    49.091
44 13.08    39.442
38 8.63    60.055
;

proc means data= fitness n mean cv median min max std skewness kurtosis max
maxdec= 2;
var vek doba_cv spotr_kysl;
run;

kurt: 1+ = spicaty (light tail), -1- = placaty (heavy tail); +-3 silne\\
skew: 0.8 +vpravo -vlevo

proc univariate data= fitness normal plot;
var vek doba_cv spotr_kysl;
run;

proc corr data= fitness plots= matrix (histogram);
run;

proc corr data= fitness nosimple fisher;
run;

proc corr data= fitness nosimple;
var doba_cv spotr_kysl;
partial vek;
run;

proc corr data= fitness nosimple;
var vek spotr_kysl;
partial doba_cv;
run;

proc corr data= fitness nosimple pearson spearman;
run;

proc reg data= fitness;
model spotr_kysl = doba_cv;
plot spotr_kysl*doba_cv;
```

```
symbol v=star;
run;

proc reg data= fitness;
model spotr_kysl = doba_cv/r influence spec;
plot spotr_kysl*doba_cv;
plot r.*p.;
symbol v= star;
run;
```

3. cv

```
proc reg data= fitness;
model spotr_kysl = doba_cv/r influence spec;
plot spotr_kysl*doba_cv;
plot r.*p.;
symbol v= star;
run;
```

3.+4. cv

```
proc reg data= work.fitness;
model spotr_kysl = doba_cv/r influence spec;
plot spotr_kysl*doba_cv;
plot r.*p.;
symbol v= star;
run;
proc reg data= work.fitness;
model spotr_kysl = doba_cv vek/r vif influence spec;
plot r.*p.;
symbol v= star;
run;
quit;
proc reg data= work.fitness;
model spotr_kysl = doba_cv vek vaha/r vif influence spec;
plot r.*p.;
symbol v= star;
run;
quit;
proc reg data= work.fitness;
Forward:model spotr_kysl = vek vaha doba_cv max_pulz/selection=f;
Backward:model spotr_kysl = vek vaha doba_cv max_pulz/selection=b;
Stepwise:model spotr_kysl = vek vaha doba_cv max_pulz/selection=stepwise;
R_square:model spotr_kysl = vek vaha doba_cv max_pulz/selection=rsquare;
run;
quit;
```

```
proc reg data= work.fitness;
Forward:model spotr_kysl = vek vaha doba_cv max_pulz/selection=f details=
summary;
Backward:model spotr_kysl = vek vaha doba_cv max_pulz/selection=b details=
summary;
Stepwise:model spotr_kysl = vek vaha doba_cv max_pulz/selection=stepwise
details= summary;
R_square:model spotr_kysl = vek vaha doba_cv max_pulz/selection=rsquare
details= summary;
run;
quit;
```

Analyza rozptylu 1

```
data teplota;
input mesic $ vysledky @@;

datalines;
leden 1.00 leden 0.64 leden 1.22 leden 1.19 leden 0.62 leden 0.87 leden 1.23
leden 0.96 leden 0.92 leden 1.11
unor 1.06 unor 0.88 unor 1.04 unor 1.66 unor 1.06 unor 1.07 unor 0.87 unor
0.97 unor 2.00 unor 1.09
brezen 1.19 brezen 1.77 brezen 1.46 brezen 1.58 brezen 1.55 brezen 1.22
brezen 1.64 brezen 1.35 brezen 1.29 brezen 1.41
;
 proc boxplot data= teplota;
 plot vysledky*mesic/boxstyle = schematic;
 plot vysledky*mesic/notches;
 run;
 proc means data= teplota n mean median min max std cv skewness   kurtosis
maxdec=2;
 class mesic;
 var vysledky;
 run;
 proc univariate data= teplota noprint;
 class mesic;
 histogram vysledky/normal;
 qqplot vysledky/normal (mu= est sigma= est)nrows =3;
 run;
 proc glm data= teplota;
 class mesic;
 model vysledky= mesic;
 means mesic/hovtest tukey;
 run;
 proc npar1way data= teplota wilcoxon;
 class mesic;
 var vysledky;
 run;
```

Analyza rozptylu 2

```
proc boxplot data= work.trzby;
 plot trzby*prodejna/boxstyle = schematic;
 plot trzby*prodejna/notches;
 run;
 proc means data= work.trzby n mean median min max std cv skewness
kurtosis maxdec=2;
 class prodejna;
 var trzby;
 run;
 proc univariate data= work.trzby noprint;
 class prodejna;
 histogram trzby/normal;
 qqplot trzby/normal (mu= est sigma= est)nrows =3;
 run;
 proc glm data= work.trzby;
 class prodejna;
 model trzby= prodejna;
 means prodejna/hovtest tukey;
 run;
 proc npar1way data= work.trzby wilcoxon;
 class prodejna;
 var trzby;
 run;
```

5. cv

1. prikad

```
data souhlas;
input vzdelani $ prirazka $ pocet @@;
datalines;
ano ano 50 ano ne 7 ano nevim 11
ne ano 14 ne ne 23 ne nevim 20
;
proc freq data = souhlas;
tables vzdelani*prirazka/expected chisq measures norow nocol nopercent;
weight pocet;
run;
```

2. priklad

```
data zakon;
input zmena $ nakup $ pocet @@;
datalines;
ano denne 27 ano nekolik_t 79 ano jednou_t 13 ano jednou_14 2
```

```
ne denne 38 ne nekolik_t 79 ne jednou_t 24 ne jednou_14 3
;
proc freq data = zakon;
tables zmena*nakup/expected chisq measures norow nocol nopercent exact;
weight pocet;
run;
```

3. priklad

```
data zkouska;
input slozeni_zk $ skola $ pocet @@;
datalines;
ano gymnazium 45 ano stredni_od 22 ano uciliste 7
ne gymnazium 7 ne stredni_od 10 ne uciliste 9
;
proc freq data = zkouska;
tables slozeni_zk*skola/expected chisq measures norow nocol nopercent exact;
weight pocet;
run;
```

6. cv

```
data work.kriminalita;
proc princomp data= work.kriminalita out= components plots=score (ellipse
ncomp=2);
var kriminalita_celkem obecna_kriminalita hospodarska_kriminalita loupeze
vloupani vrazdy;
id kraj;
run;
proc gplot data=components;
plot prin2*prin1/vref=0 href=0;
symbol v=star pointlabel= (j=r position=middle "#kraj");
run;
proc cluster data= work.kriminalita method=ave std;
id kraj;
run;
```

Ekonometrie

```
> setwd("C:/Users/C24-11/Desktop")
> read.csv("mzdy.csv");
     rok.mzda
1   1992;4644
2   1993;5904
3   1994;7004
4   1995;8307
5   1996;9825
6  1997;10802
```

```
7  1998;11801
8  1999;12797
9  2000;13219
10 2001;14378
11 2002;15524
12 2003;16430
13 2004;17466
14 2005;18344
15 2006;19546
16 2007;20957
17 2008;22592
18 2009;23344
19 2010;23797
20 2011;24319
> data = read.csv("mzdy.csv");
> View(data)
> View(data)
> data = read.csv("mzdy.csv", header=T, sep=";");
> y = data$mzda;
> x = data$rok;
> View(data)
> View(y);
> boxplot(y);
> plot(y~x)
> summary(data)
      rok            mzda
 Min.   :1992   Min.   : 4644
 1st Qu.:1997   1st Qu.:10558
 Median :2002   Median :14951
 Mean   :2002   Mean   :15050
 3rd Qu.:2006   3rd Qu.:19899
 Max.   :2011   Max.   :24319
> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4644   10560   14950   15050   19900   24320
> install.packages("outliers")
Installing package into ‘C:/Users/C24-11/Documents/R/win-library/3.2’
(as ‘lib’ is unspecified)
trying URL
'https://cran.rstudio.com/bin/windows/contrib/3.2/outliers_0.14.zip'
Content type 'application/zip' length 52663 bytes (51 KB)
downloaded 51 KB

package ‘outliers’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\C24-11\AppData\Local\Temp\Rtmpw9vBez\downloaded_packages
> outliers(y)
Error: could not find function "outliers"
```

```
> outliers::outlier(y)
> y[y<4645] <-NA
> data$dummy <- ifelse(data$rok > 2003, c(1), c(0))
```

```
# zpožděné proměnné
shift<-function(x,shift_by){
  stopifnot(is.numeric(shift_by))
  stopifnot(is.numeric(x))

  if (length(shift_by)>1)
    return(sapply(shift_by,shift, x=x))

  out<-NULL
  abs_shift_by=abs(shift_by)
  if (shift_by > 0 )
    out<-c(tail(x,-abs_shift_by),rep(NA,abs_shift_by))
  else if (shift_by < 0 )
    out<-c(rep(NA,abs_shift_by), head(x,-abs_shift_by))
  else
    out<-x
  out
}
```

```
setwd("C:/Users/C24-21.ADDS.002/Desktop")
data = read.csv("mzdy.csv", header=T, sep=";");

y = data$mzda;
x = data$rok;
data <- data.frame(y,x)
data$yl <- shift(data$y, -1)
data$yl3 <- shift(data$y, -3)
data$yd <- (data$y - data$yl)
data$yr <- (data$y / data$yl)


spotreba = read.csv("spotreba.csv", header=T, sep=";");
y = spotreba$Sp_VM
x1 = spotreba$SpC_VM
x2 = spotreba$SpC_HM
x3 = spotreba$SpC_DM
x4 = spotreba$Prijem
x0 = matrix(1,11,1)

X = cbind(x0, x1, x2, x3, x4)
A = t(X)%*%X
B = solve(A)
C = t(X)%*%y
coef = B%*%C
View(coef)

fit = lm(formula = y~x1+x2+x3+x4, data = spotreba)
```

```
summary(fit)
```

```
#LRM verifikace
data = read.csv("spotreba.csv", header = T, sep = ";")

data$konst <- c(1)

y = data$Sp_VM
x0 = data$konst
x1 = data$SpC_VM
x2 = data$SpC_HM
x3 = data$SpC_DM
x4 = data$Prijem

X = cbind(x0, x1, x2, x3, x4)
K = solve(t(X)%*%X)
coef = K%*%t(X)%*%y
View (coef)

teor = X%*%coef # vektor teoretickych hodnot zavisle promenne
res = y - teor # vektor rezidui

RSS = sum(res*res)
a = y-mean(y)
TSS = sum(a*a)
KD = 1-(RSS/TSS)

n = length(y)
k = length(coef)
KKD = 1 - (1 - KD) * ((n - 1)*(n - k))

KRR = RSS/(n-k)
Rg1 = KRR*K[1,1]
SCHg1 = sqrt(Rg1)
tg1 = coef[1,1]/SCHg1

fit = lm(y~x1+x2+x3+x4)
summary(fit)

tseries::jarque.bera.test(res)
lmtest::dwtest(y~x1+x2+x3+x4)
```

```
#Dalsi triky s R:
print(cbind(rbind(1,2,3),rbind(4,5,6)))
print(rbind(cbind(1,2,3),cbind(4,5,6)))

#rbind() = sklada prvky pod sebe (rows)
#cbind() = sklada prvky vedle sebe (columns)
#takze lze jejich kombinaci zadat matici po sloupcich i po radkach podle
```

```
potreby..
#print() = jako View() ale vypisuje do stavajici konzole misto otvirani
novyho okna
```

cvicebnice do strany 28

```
#odhad logaritmickyho modelu ala gretl

setwd("C:/Users/C24-17.ADDS.001/Desktop")
data    = read.csv("spotreba.csv", header = T, sep = ";")
data$konst <- c(1)

ly      = log(data$Sp_VM)
x0      = data$konst
lx1     = log(data$SpC_VM)
lx2     = log(data$SpC_HM)
lx3     = log(data$Prijem)


#prvni zpusob vypoctu odhadu
fit = lm(ly~lx1+lx2+lx3)
summary(fit)

coef = fit$coefficients
gama0 = exp(coef[1])

#druhej zpusob vypoctu odhadu
X = cbind(x0, lx1, lx2, lx3)
A = t(X)%*%X
B = solve(A)
C = t(X)%*%ly
coef2 = B%*%C
View(coef2)
```

```
#nejsem si jistej co to pocita, ale dela se to takhle :-D
setwd("C:/Users/C24-17.ADDS.001/Desktop")
data    = read.csv("tq.csv", header = T, sep = ";")

data$konst <- c(1)

y     = 1/data$y1
x     = 1/data$x1
#c     = data$konst

fit = lm(y~x)
summary(fit)

#vypocet parametru
coef = fit$coefficients
gama0 = 1/coef[1]
gama1 = coef[2]*gama0
```

```
#testy
#install.packages("lmtest") #staci nainstalovat jednou
lmtest::reset(y~x) #ramseyuv RESET test
lmtest::bptest(y~x) #Breusch–Pagan test - vypocet heteroskedasticity
```

Prognostika

Jak prevyst data z GRETLU do R

Gretl → nastroje → spustit GNU R. otevre se vam novy okno s R a jsou v nem prednacteny data z gretlu v objektu `gretldata` prvni promenna z gretlu je tam teda jako `gretldata[,1]`, dalsi jako `gretldata[,2]`, atd… muzete si to zkusit vypsat/vykreslit

- `print(gretldata);`
- `plot(gretldata);`
- `print(gretldata[,1]);`
- `plot(gretldata[,1]);`

Sezonni ocisteni dat

dejme tomu, ze chcem sezonne ocistit prvni promennou z gretlu. pouzijem tyhle prikazy:

```
fit <- stl(gretldata[,1], s.window=12)
print(fit)

plot(fit)
dev.copy(png,'myplot.png')
dev.off()


write.csv2(fit$time.series, file="vystup.csv")
```

- `gretldata[,1]` je vstupni vektor (nactenej z gretlu), `s.window=12` udava, ze mame 12 udaju rocne (data po mesicich).
- Pokud chcete ulozit graf do png prikazy dev…., tak nesmite zavrit okno s grafem
- Pokud chceme data z objektu ulozit do csv, musime pristupovat primo ke komponente casovych rad, ktera je dostupna jako `fit$time.series`
- write.csv2 oddeluje desetiny cisla carkama, write.csv je oddeluje teckama. zvolte co je pro vas lepsi.