Data processing (cont.)

```
################### Type of enterprise variable

# determine top 5% of enterprises by the size of harvested area
quantile(Data_crop$ahrv_total, c(.25, .50, .75, .9, .95), na.rm=TRUE)



# mark farms and state enterprises
Data_crop$entpr_type[Data_crop$opf %in% c(110)] <- "farm"
Data_crop$entpr_type[Data_crop$opf %in% c(140)] <- "entrp_state"

# check if harvested area is bigger than 5000 ha
# exclude farms and state enterprises from calculation
Data_crop$entpr_ahrv <- 0

for(i in 1:length(Data_crop$entpr_type)) {

      if(is.na(Data_crop$entpr_type[i])) {

            if (Data_crop$ahrv_total[i] > 5) {
                  Data_crop$entpr_ahrv[i]=1
            }

      }
}
rm(i)

# calculate the sum of entpr_ahrv for each ID == check if entpr was big at least
# in one year
Data_crop$entpr_ahrv_sum <- ave(Data_crop$entpr_ahrv, Data_crop$ID, FUN=sum)

# mark enterprise as big if at least during 4 years its harvested area was >= 5 thou ha
# mark enterprise as small otherwise
# exclude farms and state enterprises from calculation
for(i in 1:length(Data_crop$entpr_type)) {

      if(is.na(Data_crop$entpr_type[i])) {

            if (Data_crop$entpr_ahrv_sum[i] >= 4) {
                  Data_crop$entpr_type[i] = "big"
            } else {
```

```
                Data_crop$entpr_type[i] = "small"
            }
        }
}
rm(i)

# remove unnecessary columns
Data_crop$entpr_ahrv <- NULL
Data_crop$entpr_ahrv_sum <- NULL

# create an identificator Region_Enterprise type
Data_crop$ID_AGMD <- paste(Data_crop$region, Data_crop$entpr_type, sep="_")

# delete region from the ID of state enterprises
Data_crop$ID_AGMD[Data_crop$entpr_type %in% "entrp_state"] <- "entrp_state"


#################### Rent

# real rent UAH per ha
Data_crop$rent_ha <- Data_crop$cst_other_land_crop * 1000 / Data_crop$land_arable_rented /
Data_crop$deflator

# nominal rent UAH per ha
Data_crop$rent_ha_nom <- Data_crop$cst_other_land_crop * 1000 / Data_crop$land_arable_rented

# remove INF values
Data_crop <- Data_crop[!(Data_crop$rent_ha_nom %in% Inf),]

# remove NA values
Data_crop <- Data_crop[!is.na(Data_crop$rent_ha_nom),]
# # calculate quantiles for rent_ha
# 99.5% of enterprises pay less than 335 real UAH per year per hectare
summary(Data_crop$rent_ha)
quantile(Data_crop$rent_ha, c(.25, .50, .75, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 0.995), na.rm=TRUE)

# delete observations with rent_ha > 335 UAH
Data_crop <- Data_crop[Data_crop$rent_ha<=335,]

### rent payment per ha takink into account land owned by the enterprise
### Explanation: we should calculate the costs of renting land per ha, but what if
### part of arable land is owned by the enterprise? Than actual average rent costs
```

```
### per ha are lower!

# calculate ratio of rented arable land to arable land
# median is 1.0 mean is 0.99
Data_crop$land_ratio <- Data_crop$land_arable_rented / Data_crop$land_arable
# stat.desc(Data_crop$land_ratio)

# average rent payment per ha of rented land
# difference between avg_rent_ha and rent_ha is almoust non-existent
Data_crop$avg_rent_ha <- Data_crop$rent_ha * Data_crop$land_ratio
Data_crop$land_ratio <- NULL

# # descriptive statistics
# descr <- as.data.frame(stat.desc(Data_crop$avg_rent_ha))
# rm(descr)



#################### Support

### analyse support

# view descriptive statistic
desc_support <- format(stat.desc(subset(Data_crop, select=c(spprt_crop_sbsd, spprt_crop_vat))),
big.mark=",")

# Data is problematic: 85% of producers report either 0 or NA support to crop production from subsidies
# 68% of producers report either 0 or NA support to crop production from VAT
# Data is skewed: there a several very large outliers fpr both subsidies and VAT
# median subsidy=0, mean subsidy=71; mean VAT=243, mean VAT=1061;
# Conclusion: for those producers with missing VAT or subsidies support we will calculate median
# VAT or subsidies support for a particular type of producer in a particular region in a particular year
rm(desc_support)

# calculate quantiles for spprt_crop_sbsd
# 99.5% of enterprises receive less than 1968 thou UAH per year in the form of subsidies for crop
production
quantile(Data_crop$spprt_crop_sbsd, c(.25, .50, .75, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 0.995),
na.rm=TRUE)

# delete observations with spprt_crop_sbsd > 2000 thou UAH
Data_crop <- subset(Data_crop, Data_crop$spprt_crop_sbsd<2000 | is.na(Data_crop$spprt_crop_sbsd))
```

```
# calculate quantiles for spprt_crop_vat
# 99.5% of enterprises receive less than 16630 thou UAH per year in the form of VAT support to crop production
quantile(Data_crop$spprt_crop_vat, c(.25, .50, .75, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 0.995), na.rm=TRUE)

# delete observations with spprt_crop_vat > 17000 thou UAH
Data_crop <- subset(Data_crop, Data_crop$spprt_crop_vat<17000 | is.na(Data_crop$spprt_crop_vat))

# descriptive statistics
View(format(stat.desc(subset(Data_crop, select=c(spprt_crop_sbsd, spprt_crop_vat))), big.mark=","))

# median value of spprt_crop_sbsd
Spprt_sbsd_median <- data.frame(tapply(Data_crop$spprt_crop_sbsd, list(Data_crop$ID_AGMD, Data_crop$year), median, na.rm=TRUE))
write.csv2(Spprt_sbsd_median, "Spprt_sbsd_median.csv")
rm(Spprt_sbsd_median)

# median value of spprt_crop_vat
Spprt_vat_median <- data.frame(tapply(Data_crop$spprt_crop_vat, list(Data_crop$ID_AGMD, Data_crop$year), median, na.rm=TRUE))
write.csv2(Spprt_vat_median, "Spprt_vat_median.csv")
rm(Spprt_vat_median)


#### Merging the median subsidy and VAT support

# create ID_AGMD_year = entpr_type + region + year
Data_crop$ID_AGMD_year <- paste(Data_crop$ID_AGMD, Data_crop$year, sep="_")

# open file with the real UAH prices per 100 kg
median_spprt <- read.csv2("median_spprt.csv")

# merge data frames
Data_crop <- merge(Data_crop, median_spprt, by=c("ID_AGMD_year"))

# remove unnecessary data frame
rm(median_spprt)


#### Substituting NA support with median support
```

```
# substitute NA in spprt_crop_sbsd
Data_crop$spprt_crop_sbsd <- ifelse(is.na(Data_crop$spprt_crop_sbsd),
Data_crop$median_spprt_crop_sbsd, Data_crop$spprt_crop_sbsd)

# substitute NA in spprt_crop_vat
Data_crop$spprt_crop_vat <- ifelse(is.na(Data_crop$spprt_crop_vat),
Data_crop$median_spprt_crop_vat, Data_crop$spprt_crop_vat)

# total spport thou UAH
Data_crop$spprt_total <- apply(cbind(Data_crop$spprt_crop_sbsd, Data_crop$spprt_crop_vat),
                    MARGIN=1, FUN=sum, na.rm=TRUE)

# real support UAH per ha
Data_crop$spprt_ha <- (Data_crop$spprt_total*1000) / Data_crop$land_arable /  Data_crop$deflator

# descriptive statistics
View(format(stat.desc(subset(Data_crop, select=c(spprt_crop_sbsd, spprt_crop_vat, spprt_total,
spprt_ha))), big.mark=","))
quantile(Data_crop$spprt_ha, c(.25, .50, .75, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99, 0.995), na.rm=TRUE)


#################### Set data as panel data

pdata <- Data_crop[order (Data_crop$ID, Data_crop$year), ]
pdata <- pdata.frame(Data_crop, index = c("ID", "year"), drop.index = FALSE, row.names = TRUE)


######### Gross margins


######## Corn
pdata$crn_prc_lag <- lag(pdata$crn_prc, 1)
pdata$crn_rent <- pdata$avg_rent_ha / (lag(pdata$crn_yield, 1) * 10)
pdata$crn_spprt <- pdata$spprt_ha / (lag(pdata$crn_yield, 1) * 10)

pdata$crn_mrgn <- apply(cbind(pdata$crn_cst_inpt_kg, pdata$crn_prc_lag), MARGIN=1, FUN=diff,
na.rm=TRUE)
pdata$crn_mrgn <- apply(cbind(pdata$crn_rent, pdata$crn_mrgn), MARGIN=1, FUN=diff, na.rm=TRUE)
pdata$crn_amrgn <- apply(cbind(pdata$crn_spprt, pdata$crn_mrgn), MARGIN=1, FUN=sum,
na.rm=TRUE)
pdata$crn_amrgn <-  ifelse(is.na(pdata$crn_mrgn), NA, pdata$crn_amrgn)
```

```
# determine distribution of crn_cst_inpt_kg
# 99% of enterprises have corn costs less than 90 UAH per kg
quantile(pdata$crn_cst_inpt_kg, c(.25, .50, .75, .9, .99), na.rm=TRUE)

pdata$crn_mrgn <-  ifelse(pdata$crn_cst_inpt_kg>91, NA, pdata$crn_mrgn)
pdata$crn_amrgn <-  ifelse(pdata$crn_cst_inpt_kg>91, NA, pdata$crn_amrgn)


# determine distribution of crn_yield
# 99% of enterprises have corn yield bigger than 0.45 tonne per ha
quantile(pdata$crn_yield, c(.01, .25, .50, .75, .9, .99), na.rm=TRUE)

pdata$crn_mrgn <-  ifelse(pdata$crn_yield<0.45, NA, pdata$crn_mrgn)
pdata$crn_amrgn <-  ifelse(pdata$crn_yield<0.45, NA, pdata$crn_amrgn)


# determine distribution of crn_spprt
# 99% of enterprises have corn support less than 29 UAH per 100 kg
quantile(pdata$crn_spprt, c(.01, .25, .50, .75, .9, .99), na.rm=TRUE)

pdata$crn_mrgn <-  ifelse(pdata$crn_spprt>29, NA, pdata$crn_mrgn)
pdata$crn_amrgn <-  ifelse(pdata$crn_spprt>29, NA, pdata$crn_amrgn)

quantile(pdata$crn_mrgn, c(.25, .50, .75, .9, .99), na.rm=TRUE)
quantile(pdata$crn_amrgn, c(.25, .50, .75, .9, .99), na.rm=TRUE)

stat.desc(pdata$crn_mrgn)
stat.desc(pdata$crn_amrgn)

pdata$crn_prc_lag <- NULL
pdata$crn_rent <- NULL
pdata$crn_spprt <- NULL

# trend variable
pdata$trend <- as.numeric(pdata$year)
pdata$ln_trend <- log(pdata$trend)

pdata <- subset(pdata, select=c(ID, region, entpr_type, year, crn_yield, crn_ahrv, crn_amrgn, ln_trend))
pdata <- pdata[!is.na(pdata$crn_amrgn),]
save(pdata, file="pdata_02_05.Rda")
```

Regressions estimation

Farms in Forrest steppe

#################### Necessary libraries

```
# package required for panel data analysis
# install.packages("plm")
library(plm)

# export regression results
# install.packages("stargazer")
library(stargazer)

# package required for descriptive statistics (stat.desc command)
# install.packages("pastecs")
library(pastecs)

# install.packages("gplots")
library(gplots)
```

#################### Global print settings

```
# numbers are printed in a fixed notation, unless they are more than scipen digits wider
options(scipen=50)

# set the number of digits to print when printing numeric values to 2
options(digits=2)

# set thousands deliminator
options(big.mark = ",")

# setwd("C:/Users/Oleksa/Dropbox/Work/Production_function/Data")
setwd("C:/Users/stepaniuk/Dropbox/Work/Production_function/Data")
```

#################### Set data as panel data for XXX enterprises from YYY region

```
load("pdata_02_05.Rda")
```

```
# select subset of data
pdata_subs <- subset(pdata, pdata$entpr_type=="small")
pdata_subs <- subset(pdata_subs, pdata_subs$region=="Forest_steppe")
pdata_subs <- as.data.frame(pdata_subs)
pdata_subs <- pdata.frame(pdata_subs,  index = c("ID", "year"), drop.index = FALSE, row.names = TRUE)

attach(pdata_subs)


#################### Preliminary calculations

# descriptive statistics
View(format(stat.desc(subset(pdata_subs, select=c(crn_ahrv, crn_yield, crn_amrgn))), big.mark=","))

# yield outliers
quantile(pdata_subs$crn_yield, c(0.0005,.25, .50, .75, 0.9, 0.9995), na.rm=TRUE)
pdata_subs <- pdata_subs[pdata_subs$crn_yield<18.5,]
pdata_subs <- pdata_subs[pdata_subs$crn_yield>0.60,]

# margin outliers
quantile(pdata_subs$crn_amrgn, c(0.0005,.25, .50, .75, 0.9, 0.9995), na.rm=TRUE)
pdata_subs <- pdata_subs[pdata_subs$crn_amrgn<39.2,]
pdata_subs <- pdata_subs[pdata_subs$crn_amrgn>-62,]


#################### Yield of corn

### graphical analysis

# establish graph margins
par(mar = rep(2, 4))
plot(year, crn_yield, ylim=c(0,15),outline=FALSE)
plotmeans(crn_yield ~ year, data=pdata_subs, ylim=c(0,8))

plot(crn_amrgn, crn_yield)
abline(lm(crn_yield~crn_amrgn), col="red")

### regression analysis

pooling <- plm(crn_yield ~ ln_trend + crn_amrgn,
          data=pdata_subs, model="pooling")
summary(pooling)
```

```
fixed <- plm(crn_yield ~ ln_trend + crn_amrgn,
          data=pdata_subs, model="within")
summary(fixed)


random <- plm(crn_yield ~ ln_trend + crn_amrgn,
           data=pdata_subs, model="random")
summary(random)

# Fixed or Random: Hausman test
phtest(random, fixed)

# Fixed or Pooling: F test
pFtest(fixed, pooling)

# Random or Pooling: Lagrange Multiplier Test
plmtest(pooling, type=c("bp"))

# Unobserved variable: Wooldridge's test
pwtest(crn_yield ~ ln_trend + crn_amrgn, data=pdata_subs)

stargazer(pooling, fixed, random, type="text",
        dep.var.labels=c("Corn yield, tone per ha"),
        covariate.labels=c("Ln trend",
                        "Corn adjusted gross margin, UAH per 100 kg"),
        out="crn_yield.txt")

save(pdata_subs, file="result.Rda")
```

German-Ukrainian Agricultural Policy Dialogue

With support from Federal Ministry of Food and Agriculture by decision of the German Bundestag

at the Institute for Economic Research and Policy Consulting

Курс в рамках програми аспірантури з аграрної економіки у НУБіП

„Моделювання впливу політичних рішень на аграрний сектор за допомогою моделі AGMEMOD“

Лекція III: Економетрична оцінка функцій з часових рядів

Інструктор – Марія Богонос, PhD
Експерт з питань аграрної політики
E-mail: *bogonos@apd-ukraine.de*
Tel: 044/235-63-27, Mob: 099/029-17-29

3. Оцінювання функцій з часових рядів

- автокореляція (-авто, -кореляція, „correlate“ залежати) у змінних

  Перевірка змінних на автокреляцію:
  - графік значень змінної до року
  - перевірка на автокореляцію 1го ступеня за допомогою тесту «Серійна кореляція» *(якщо немає автокореляції 1го ступеня...)* та тесту «Статистика Дурбіна Ватсона»

  вважати на «значимість» коефіцієнтів та статистик

тест «Статистика Дурбіна Ватсона»

Durbin-Watson d-statistic(3, 10) = 1.4458

Гіпотеза 0: автокореляція 1го ступеня відсутня

10 спостережень і 3 змінних

1.446 для 5% рівня значимості: 10 спостережень і 3 змінних (2 рівня свободи), нижній ліміт статистики – 0.466, верхній ліміт статистики – 1.373.

Оскільки 1.446>1.373, нульова гіпотза не відхилена

3. Оцінювання функцій з часових рядів

1) Описова статистика даних
   - перелік змінних та кількість спостережень
   - середнє значення кожної змінної
   - стандартне відхилення
   - мін та макс значення
2) Попередній аналіз даних
   - автокореляція (-авто, -кореляція, „correlate“ залежати) у змінних
   - перевірка на авторегресивну умовну гетероскедастичність

3. Оцінювання функцій з часових рядів

- перевірка на авторегресивну умовну гетероскедастичність

LM тест

Гіпотеза 0: авторегресивна умовна гетероскедастичність відсутня

LM test for autoregressive conditional heteroskedasticity (ARCH)

lags(*p*)	chi2	df	Prob > chi2
1	0.000	1	1.0000
2	0.000	2	1.0000
3	0.000	3	1.0000
4	0.000	4	1.0000
5	0.000	5	1.0000
6	0.000	6	1.0000
7	0.000	7	1.0000

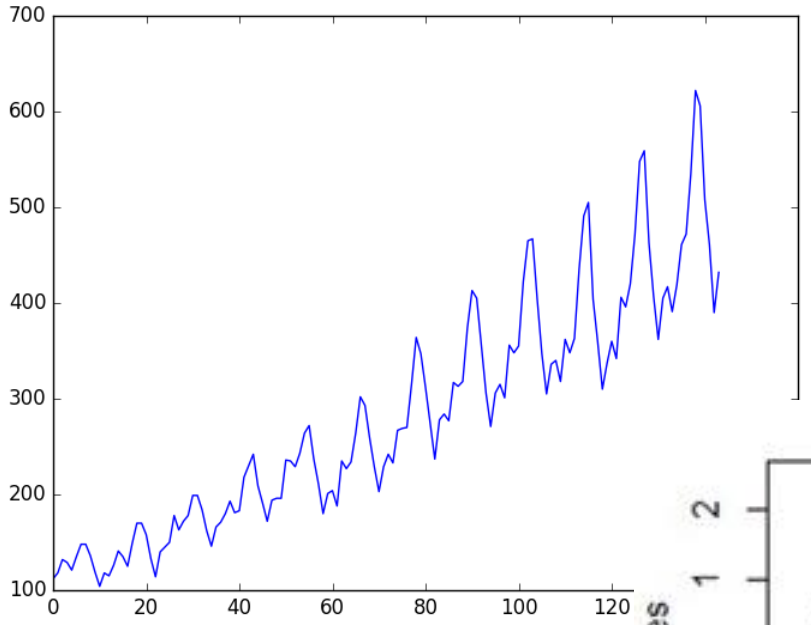
H0: no ARCH effects *vs.* H1: ARCH(*p*) disturbance

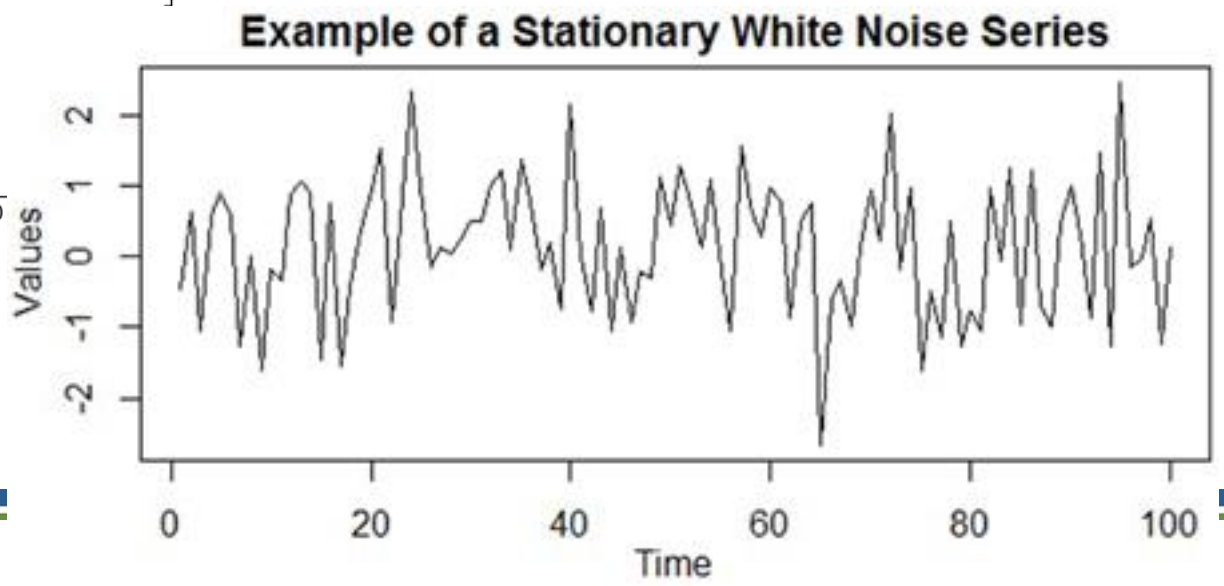
3. Оцінювання функцій з часових рядів

1) Описова статистика даних
   - перелік змінних та кількість спостережень
   - середнє значення кожної змінної
   - стандартне відхилення
   - мін та макс значення
2) Попередній аналіз даних
   - автокореляція (-авто, -кореляція, „correlate“ залежати) у змінних
   - перевірка на авторегресивну умовну гетероскедастичність
   - перевірка на стаціонарність

3. Оцінювання функцій з часових рядів

- перевірка на стаціонарність





3. Оцінювання функцій з часових рядів

- перевірка на стаціонарність

Тест «Dickey Fuller»

Гіпотеза 0: стаціонарність присутня

```
. dfuller wht_capfoodcnsmp_1000t, trend

Dickey-Fuller test for unit root                    Number of obs   =         9

                                 ---------- Interpolated Dickey-Fuller ----------
                    Test         1% Critical       5% Critical      10% Critical
                 Statistic           Value             Value            Value
------------------------------------------------------------------------------
 Z(t)              -1.673           -4.380            -3.600           -3.240
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.7627
```

3. Оцінювання функцій з часових рядів

1) Описова статистика даних
   - перелік змінних та кількість спостережень
   - середнє значення кожної змінної
   - стандартне відхилення
   - мін та макс значення
2) Попередній аналіз даних
   - автокореляція (-авто, -кореляція, „correlate“ залежати) у змінних
   - перевірка на авторегресивну умовну гетероскедастичність
   - перевірка на стаціонарність
3) Коригування змінних та оцінка функції

3. Оцінювання функцій з часових рядів

3) Коригування змінних та оцінка функції
   - включення тренду
   - включення бінарних змінних

```
. regress wht_capfoodcnsmp_1000t gdpcap_rl_UAH dm_09 year
```

Source	SS	df	MS
Model	1.6379e-10	3	5.4596e-11
Residual	2.7136e-11	6	4.5226e-12
Total	1.9092e-10	9	2.1214e-11

Number of obs	10
F(3, 6)	12.07
Prob > F	0.0059
R-squared	0.8579
Adj R-squared	0.7868
Root MSE	2.1e-06

wht_cap~1000t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
gdpcap_rl_UAH	-5.81e-09	2.63e-09	-2.21	0.069	-1.23e-08	6.27e-10
dm_09	-4.09e-06	2.53e-06	-1.61	0.158	-.0000103	2.11e-06
year	-1.07e-06	2.59e-07	-4.12	0.006	-1.70e-06	-4.34e-07
_cons	.0022964	.0005141	4.47	0.004	.0010384	.0035543

German-Ukrainian Agricultural Policy Dialogue

With support from Federal Ministry of Food and Agriculture by decision of the German Bundestag

at the Institute for Economic Research and Policy Consulting

Курс в рамках програми аспірантури з аграрної економіки у НУБіП

„Моделювання впливу політичних рішень на аграрний сектор за допомогою моделі AGMEMOD“

Лекція ІІ: Моделювання політичних рішень у аграрному секторі за допомогою ЕМЧР: поєднання теорії та «реальності» на прикладі AGMEMOD-UA

Інструктор – Марія Богонос, PhD
Експерт з питань аграрної політики
E-mail: *bogonos@apd-ukraine.de*
Tel: 044/235-63-27, Mob: 099/029-17-29

Зміст

- 1. Визначення політичних цілей, реформ та стейкхолдерів
- 2. Створення схеми моделі на основі схеми агросектору (AGMEMOD-UA)
- 3. Створення сценаріїв політичних реформ
- 4. Пошук та обробка даних
- 5. Створення математичної моделі: розв’язок, поєднання методологій
- 6. Вибір програмного забезпечення для реалізації моделі: доступність, функціональність, простота та зручність використання
- 7. Калібрування та валідація моделі
- 8. Оцінені значення та експертна думка
- 9. Цикл покращення/оновлення моделі

5. Математична модель

модель часткової рівноваги – економетрика

модель часткової рівноваги – математичне програмування

інші підходи? приклади?

6. Програмне забезпечення

- Доступність
- Функціональність
- Простота та зручність використання

Моделі ринкової рівноваги та математичного програмування:

GAMS https://www.gams.com/

повна версія – платна

free demo версія для – безкоштовно (макс кількість змінних та рівнянь 300, база даних – 2000 одиниць)

Статистичний аналіз:

STATA http://www.stata.com/

повна версія – платна

повна версія на 30 днів – безкоштовно

(IBM) SPSS http://www-03.ibm.com/software/products/en/spss-statistics

повна версія – платна

повна версія на 14 днів – безкоштовно

R https://www.r-project.org/

повна версія – безкоштовно