

## Импорт данных:

```
clear
opts = delimitedTextImportOptions("NumVariables", 19);

% Specify range and delimiter
opts.DataLines = [2, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["index", "kitchen_area", "bath_area", "other_area", "gas", "hot_water", "central_heating", "extra_area_type_name", "distribution"];
opts.VariableTypes = ["double", "double", "double", "double", "categorical", "categorical", "categorical", "categorical", "categorical"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, ["gas", "hot_water", "central_heating", "extra_area_type_name", "distribution"], "categorical");

% Import the data
data = readtable("C:\Users\vorko\OneDrive\Рабочий стол\Учеба\Учеба(финальный сем)\Identification\data\kitchen_data.csv");

%% Set up the Import Options and import the data
opts = delimitedTextImportOptions("NumVariables", 18);

% Specify range and delimiter
opts.DataLines = [2, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["index", "kitchen_area", "bath_area", "other_area", "gas", "hot_water", "central_heating", "extra_area_type_name", "distribution"];
opts.VariableTypes = ["double", "double", "double", "double", "categorical", "categorical", "categorical", "categorical", "categorical"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, ["gas", "hot_water", "central_heating", "extra_area_type_name", "distribution"], "categorical");

% Import the data
test = readtable("C:\Users\vorko\OneDrive\Рабочий стол\Учеба\Учеба(финальный сем)\Identification\data\kitchen_data.csv");

%% Clear temporary variables
clear opts
```

## Поиск коэффициентов:

Для более точной оценки необходимо сделать следующие шаги

1. Все районы города отличаются друг от друга типом жилья. Потому для более точного предсказания, необходимо разделить эту выборку на подвыборки по каждому району и отдельно оценивать параметры под каждую из них
2. Как показал эксперимент лучше не оценивать тип extra\_area. Это не имеет значения.
3. Параметр total\_area тоже оказался излишним так как он находится линейной комбинацией параметров  $[kitchenArea + bathArea + otherArea + 1/3extraArea]$

```
isNotCentralnyj=data.district_name~='Centralnyj';
CentralnyjData=data;
CentralnyjData(isNotCentralnyj,:) = [];
XCentralny=getRegressor(CentralnyjData);
YCentralnyj=CentralnyjData.price;
thetaCentralnyj= XCentralny \ YCentralnyj;

isNotKirovskij=data.district_name~='Kirovskij';
KirovskijData=data;
KirovskijData(isNotKirovskij,:) = [];
XKirovskij=getRegressor(KirovskijData);
YKirovskij=KirovskijData.price;
thetaKirovskij= XKirovskij \ YKirovskij;

isNotKrasnoselskij=data.district_name~='Krasnoselskij';
KrasnoselskijData=data;
KrasnoselskijData(isNotKrasnoselskij,:) = [];
XKrasnoselskij=getRegressor(KrasnoselskijData);
YKrasnoselskij=KrasnoselskijData.price;
thetaKrasnoselskij= XKrasnoselskij \ YKrasnoselskij;

isNotMoskovskij=data.district_name~='Moskovskij';
MoskovskijData=data;
MoskovskijData(isNotMoskovskij,:) = [];
XMoskovskij=getRegressor(MoskovskijData);
YMoskovskij=MoskovskijData.price;
thetaMoskovskij= XMoskovskij \ YMoskovskij;

isNotNevskij=data.district_name~='Nevskij';
NevskijData=data;
NevskijData(isNotNevskij,:) = [];
XNevskij=getRegressor(NevskijData);
YNevskij=NevskijData.price;
thetaNevskij= XNevskij \ YNevskij;

isNotPetrogradskij=data.district_name~='Petrogradskij';
PetrogradskijData=data;
PetrogradskijData(isNotPetrogradskij,:) = [];
XPetrogradskij=getRegressor(PetrogradskijData);
YPetrogradskij=PetrogradskijData.price;
thetaPetrogradskij= XPetrogradskij \ YPetrogradskij;
```

```

isNotVyborgskij=data.district_name~='Vyborgskij';
VyborgskijData=data;
VyborgskijData(isNotVyborgskij,:) = [];
XVyborgskij=getRegressor(VyborgskijData);
YVyborgskij=VyborgskijData.price;
thetaVyborgskij= XVyborgskij \ YVyborgskij;

```

### Реализация:

```

isNotCentralnyj=test.district_name~='Centralnyj';
CentralnyjData=test;
CentralnyjData(isNotCentralnyj,:) = [];
XCentralnyj=getRegressor(CentralnyjData);
YCentralnyj=XCentralnyj*thetaCentralnyj;
answerCentralnyj=[CentralnyjData.index YCentralnyj];

isNotKirovskij=test.district_name~='Kirovskij';
KirovskijData=test;
KirovskijData(isNotKirovskij,:) = [];
XKirovskij=getRegressor(KirovskijData);
YKirovskij=XKirovskij*thetaKirovskij;
answerKirovskij=[KirovskijData.index YKirovskij];

isNotKrasnoselskij=test.district_name~='Krasnoselskij';
KrasnoselskijData=test;
KrasnoselskijData(isNotKrasnoselskij,:) = [];
XKrasnoselskij=getRegressor(KrasnoselskijData);
YKrasnoselskij=XKrasnoselskij*thetaKrasnoselskij;
answerKrasnoselskij=[KrasnoselskijData.index YKrasnoselskij];

isNotMoskovskij=test.district_name~='Moskovskij';
MoskovskijData=test;
MoskovskijData(isNotMoskovskij,:) = [];
XMoskovskij=getRegressor(MoskovskijData);
YMoskovskij=XMoskovskij*thetaMoskovskij;
answerMoskovskij=[MoskovskijData.index YMoskovskij];

isNotNevskij=test.district_name~='Nevskij';
NevskijData=test;
NevskijData(isNotNevskij,:) = [];
XNevskij=getRegressor(NevskijData);
YNevskij=XNevskij*thetaNevskij;
answerNevskij=[NevskijData.index YNevskij];

isNotPetrogradskij=test.district_name~='Petrogradskij';
PetrogradskijData=test;
PetrogradskijData(isNotPetrogradskij,:) = [];

```

```

XPetrogradskij=getRegressor(PetrogradskijData);
YPetrogradskij=XPetrogradskij*thetaPetrogradskij;
answerPetrogradskij=[PetrogradskijData.index YPetrogradskij];

isNotVyborgskij=test.district_name~='Vyborgskij';
VyborgskijData=test;
VyborgskijData(isNotVyborgskij,:) = [];
XVyborgskij=getRegressor(VyborgskijData);
YVyborgskij=XVyborgskij*thetaVyborgskij;
answerVyborgskij=[VyborgskijData.index YVyborgskij];

```

#### Формирование ответа:

```

answer=cat(1, ...
    answerCentralnyj, ...
    answerKirovskij, ...
    answerKrasnoselskij, ...
    answerMoskovskij, ...
    answerNevskij, ...
    answerPetrogradskij, ...
    answerVyborgskij);

priceTable=table(answer(:,1), answer(:,2), 'VariableNames', {'index', 'price'});
sortPriceTable=sortrows(priceTable, 1);
writetable(sortPriceTable, "C:\Users\vorko\OneDrive\Рабочий стол\Учеба\Учеба(финальный сем)\Id

```

#### Формирование регрессора:

```

function X = getRegressor(data)
    hasGas=data.gas=="Yes";
    hasCentralHeating=data.central_heating=="Yes";
    hasHotWater=data.hot_water=="Yes";
    kitchenArea = data.kitchen_area;
    bathArea = data.bath_area;
    otherArea = data.other_area;
    extraArea = data.extra_area;
    extraArea_count = data.extra_area_count;
    year = data.year;
    ceilHeight = data.ceil_height;
    floorMax = data.floor_max;
    isFirstFloor = data.floor==1;
    bathCount = data.bath_count;
    roomsCount = data.rooms_count;

    X=[ hasGas, ...
        hasCentralHeating, ...

```

```
hasHotWater, ...  
kitchenArea, ...  
bathArea, ...  
otherArea, ...  
extraArea, ...  
extraArea_count, ...  
year, ...  
ceilHeight, ...  
floorMax, ...  
isFirstFloor, ...  
bathCount, ...  
roomsCount ];
```

```
end
```