Linear Regression in Python

Online Statistical Computing Reference
Machine Learning Module

November 2019

Introduction

- In this module, we will be introducing how to construct a linear regression model on a given dataset.
- A linear model can take on two forms:
  - Simple linear regression (SLR) model `y ~ x`
    where `y` is the response and `x` is a predictor variable
  - Multiple linear regression (MLR) model $y \sim x_1 + x_2 + \ldots + x_n$
    where x_i are predictor variables in predicting the response variable `y`

Structure of this tutorial

- Data Import
- Data Preprocessing
- Exploratory Data Analysis (EDA)
  - Correlation heatmap, boxplots, scatterplots, histograms and density plots
- Linear Regression Modeling:
  - Variable selection
  - Summary statistics
  - Diagnostics and assumptions
  - Model selection
- Discussion

Data Import

- The dataset that we will be using is the `soil.csv` dataset
- To load the data into Python:

In [5]:
```
soil = pd.read_csv("soil.csv")  # data import
soil.head() # check if read in correctly
```

Out[5]:

	x	y	cadmium	copper	lead	zinc	elev	dist	om	ffreq	soil	lime	landuse	dist.m
0	181072	333611	11.7	85	299	1022	7.909	0.001358	13.6	1	1	1	Ah	50
1	181025	333558	8.6	81	277	1141	6.983	0.012224	14.0	1	1	1	Ah	30
2	181165	333537	6.5	68	199	640	7.800	0.103029	13.0	1	1	1	Ah	150
3	181298	333484	2.6	81	116	257	7.655	0.190094	8.0	1	2	0	Ga	270
4	181307	333330	2.8	48	117	269	7.480	0.277090	8.7	1	2	0	Ah	380

- To check the dimension of the dataset:

In [6]:
```
soil.shape # rows x columns
```

Out[6]:
```
(155, 14)
```

Data Import

- Variables in the dataset:
  - **`x`**: x-coordinate of the location
  - **`y`**: y-coordinate of the location
  - **`cadmium`**: topsoil cadmium concentration
  - **`copper`**: topsoil copper concentration
  - **`lead`**: topsoil lead concentration
  - **`zinc`**: topsoil zinc concentration
  - **`elev`**: relative elevation above local river bed
  - **`dist`**: distance to the Meuse
  - **`om`**: organic matter
  - **`ffreq`**: flooding frequency class (1 = once in 2 years; 2 = once in 10 years; 3 = once in 50 years)
  - **`soil`**: soil type according to the 1:50000 soil map of the Netherlands
  - **`lime`**: lime class (0 = absent, 1 = present)
  - **`landuse`**: landuse class
  - **`dist.m`**: distance to river Meuse in meters

Data Import

- Variables in the dataset:
  - **`x`**: x-coordinate of the location
  - **`y`**: y-coordinate of the location
  - **`cadmium`**: topsoil cadmium concentration
  - **`copper`**: topsoil copper concentration
  - **`lead`**: topsoil lead concentration (response variable)
  - **`zinc`**: topsoil zinc concentration
  - **`elev`**: relative elevation above local river bed
  - **`dist`**: distance to the Meuse
  - **`om`**: organic matter
  - **`ffreq`**: flooding frequency class (1 = once in 2 years; 2 = once in 10 years; 3 = once in 50 years)
  - **`soil`**: soil type according to the 1:50000 soil map of the Netherlands
  - **`lime`**: lime class (0 = absent, 1 = present)
  - **`landuse`**: landuse class
  - **`dist.m`**: distance to river Meuse in meters

Data Preprocessing

- We notice that there are a few missing values in the original dataset.
- Since there are only a small number of rows with missing values, we can remove those rows:

```
In [8]: index = pd.isnull(soil).any(axis = 1)
        soil = soil[-index]
        soil = soil.reset_index(drop = True)

In [9]: soil.shape

Out[9]: (152, 14)
```

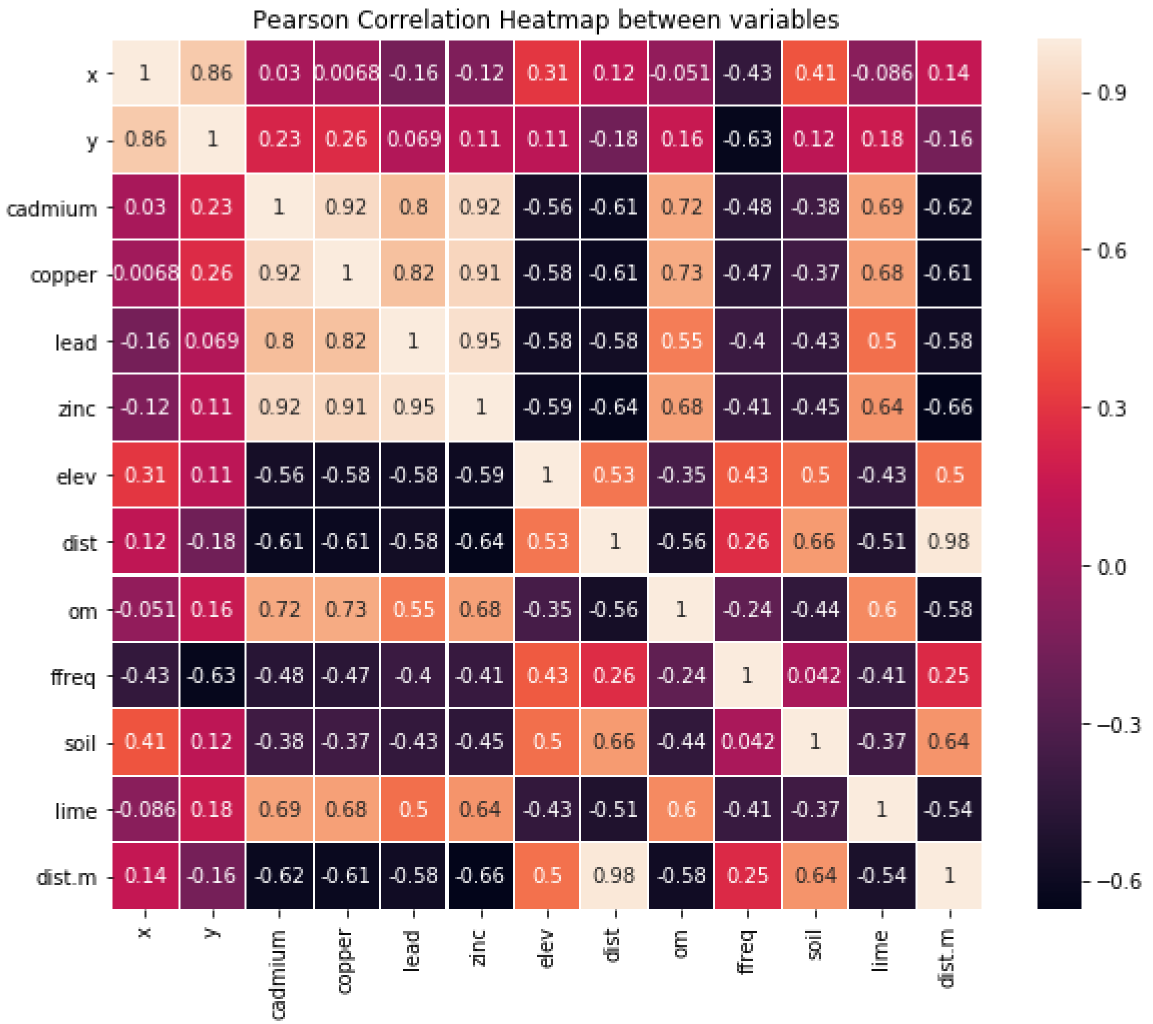
Exploratory Data Analysis (EDA)

- correlation heatmap
- boxplots
- scatterplots
- histograms and density plots

Exploratory Data Analysis (EDA)

- correlation heatmap

```
Out[11]: elev       -0.584323
         dist.m     -0.584204
         dist       -0.576577
         soil       -0.430423
         ffreq      -0.399238
         x          -0.158104
         y           0.069192
         lime        0.501632
         om          0.547836
         cadmium     0.800898
         copper      0.817000
         zinc        0.954303
         lead        1.000000
         Name: lead, dtype: float64
```



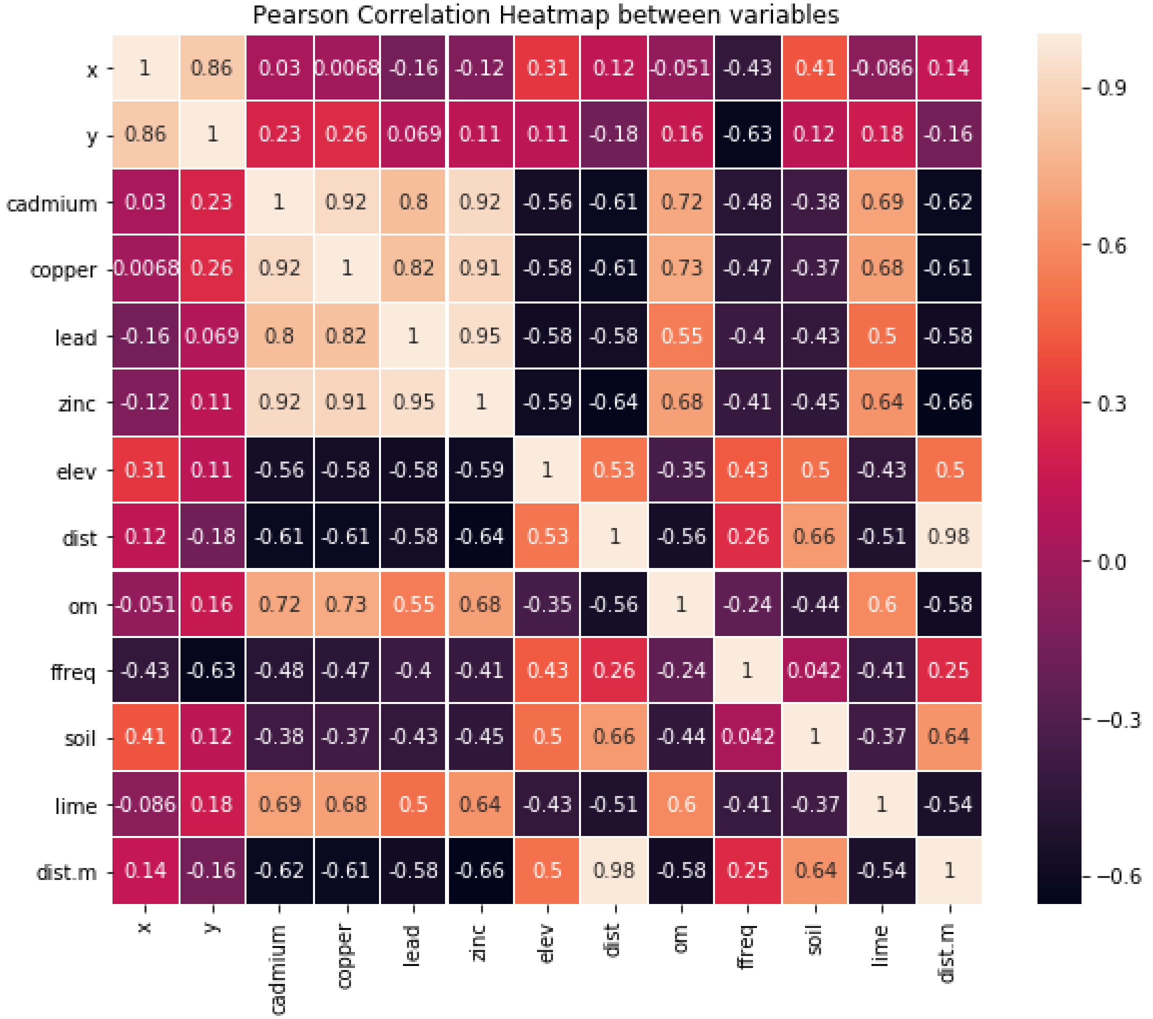
Exploratory Data Analysis (EDA)

- correlation heatmap

```
Out[11]: elev       -0.584323
         dist.m     -0.584204
         dist       -0.576577
         soil       -0.430423
         ffreq      -0.399238
         x          -0.158104
         y           0.069192
         lime        0.501632
         om          0.547836
         cadmium     0.800898
         copper      0.817000
         zinc        0.954303
         lead        1.000000
         Name: lead, dtype: float64
```

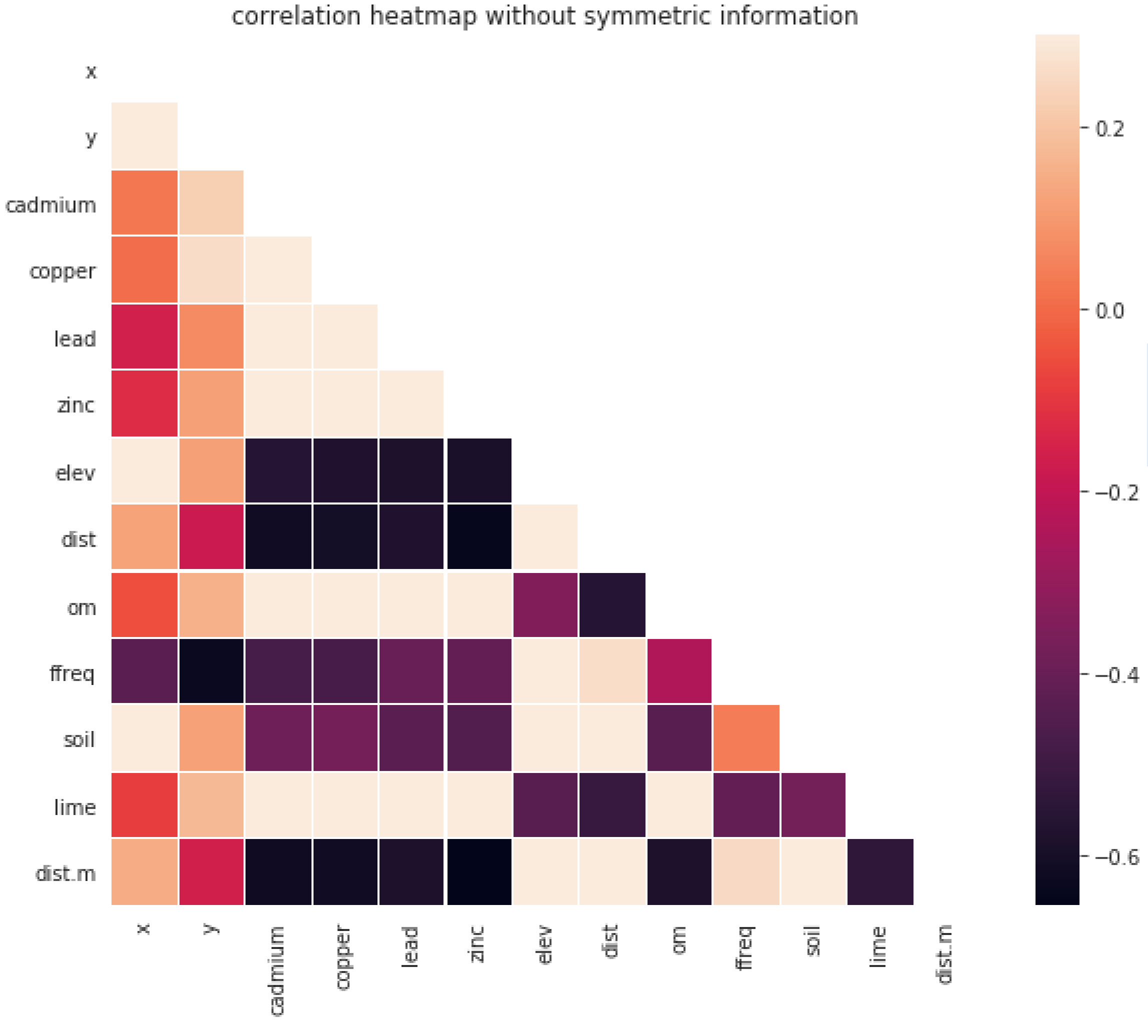
negative correlation

positive correlation

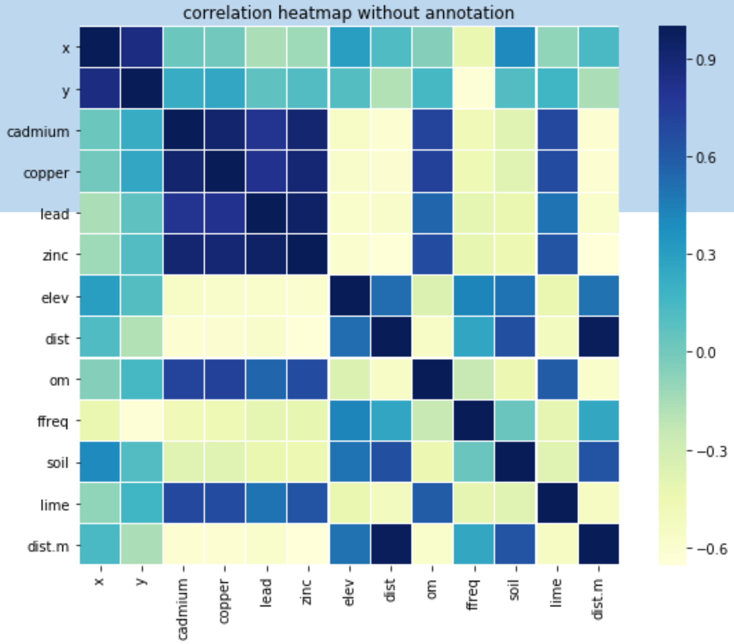


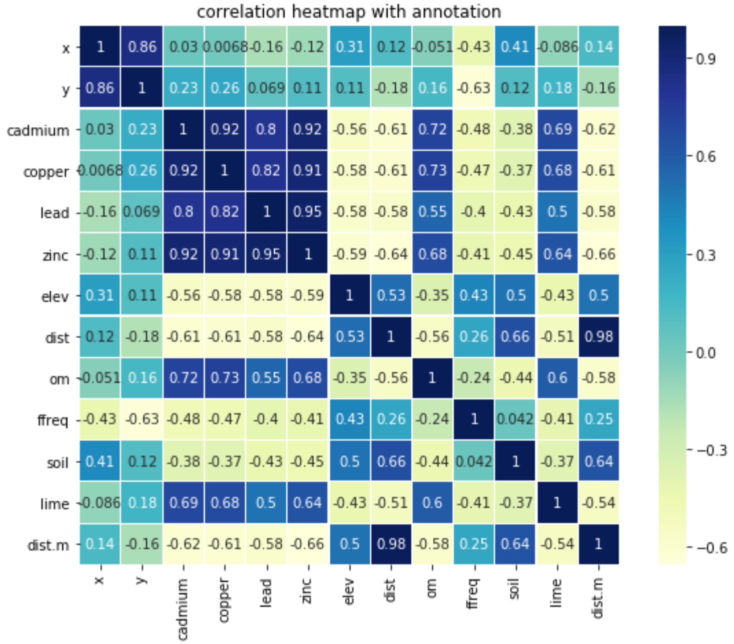
	x	y	cadmium	copper	lead	zinc	elev	dist	om	ffreq	soil	lime	dist.m
x	1	0.86	0.03	0.0068	-0.16	-0.12	0.31	0.12	-0.051	-0.43	0.41	-0.086	0.14
y	0.86	1	0.23	0.26	0.069	0.11	0.11	-0.18	0.16	-0.63	0.12	0.18	-0.16
cadmium	0.03	0.23	1	0.92	0.8	0.92	-0.56	-0.61	0.72	-0.48	-0.38	0.69	-0.62
copper	0.0068	0.26	0.92	1	0.82	0.91	-0.58	-0.61	0.73	-0.47	-0.37	0.68	-0.61
lead	-0.16	0.069	0.8	0.82	1	0.95	-0.58	-0.58	0.55	-0.4	-0.43	0.5	-0.58
zinc	-0.12	0.11	0.92	0.91	0.95	1	-0.59	-0.64	0.68	-0.41	-0.45	0.64	-0.66
elev	0.31	0.11	-0.56	-0.58	-0.58	-0.59	1	0.53	-0.35	0.43	0.5	-0.43	0.5
dist	0.12	-0.18	-0.61	-0.61	-0.58	-0.64	0.53	1	-0.56	0.26	0.66	-0.51	0.98
om	-0.051	0.16	0.72	0.73	0.55	0.68	-0.35	-0.56	1	-0.24	-0.44	0.6	-0.58
ffreq	-0.43	-0.63	-0.48	-0.47	-0.4	-0.41	0.43	0.26	-0.24	1	0.042	-0.41	0.25
soil	0.41	0.12	-0.38	-0.37	-0.43	-0.45	0.5	0.66	-0.44	0.042	1	-0.37	0.64
lime	-0.086	0.18	0.69	0.68	0.5	0.64	-0.43	-0.51	0.6	-0.41	-0.37	1	-0.54
dist.m	0.14	-0.16	-0.62	-0.61	-0.58	-0.66	0.5	0.98	-0.58	0.25	0.64	-0.54	1

Exploratory Data Analysis (EDA)



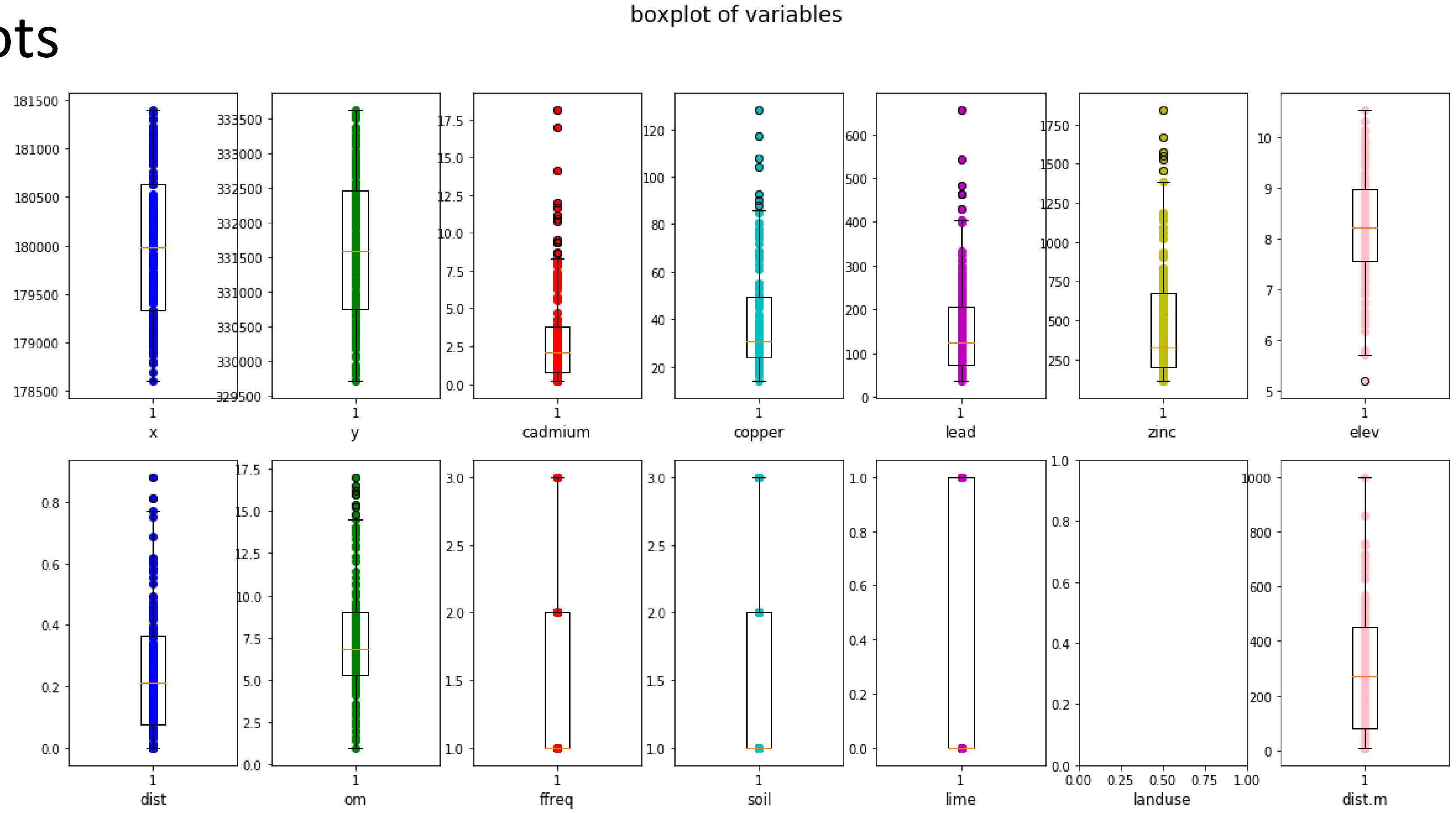
different versions of correlation heatmap





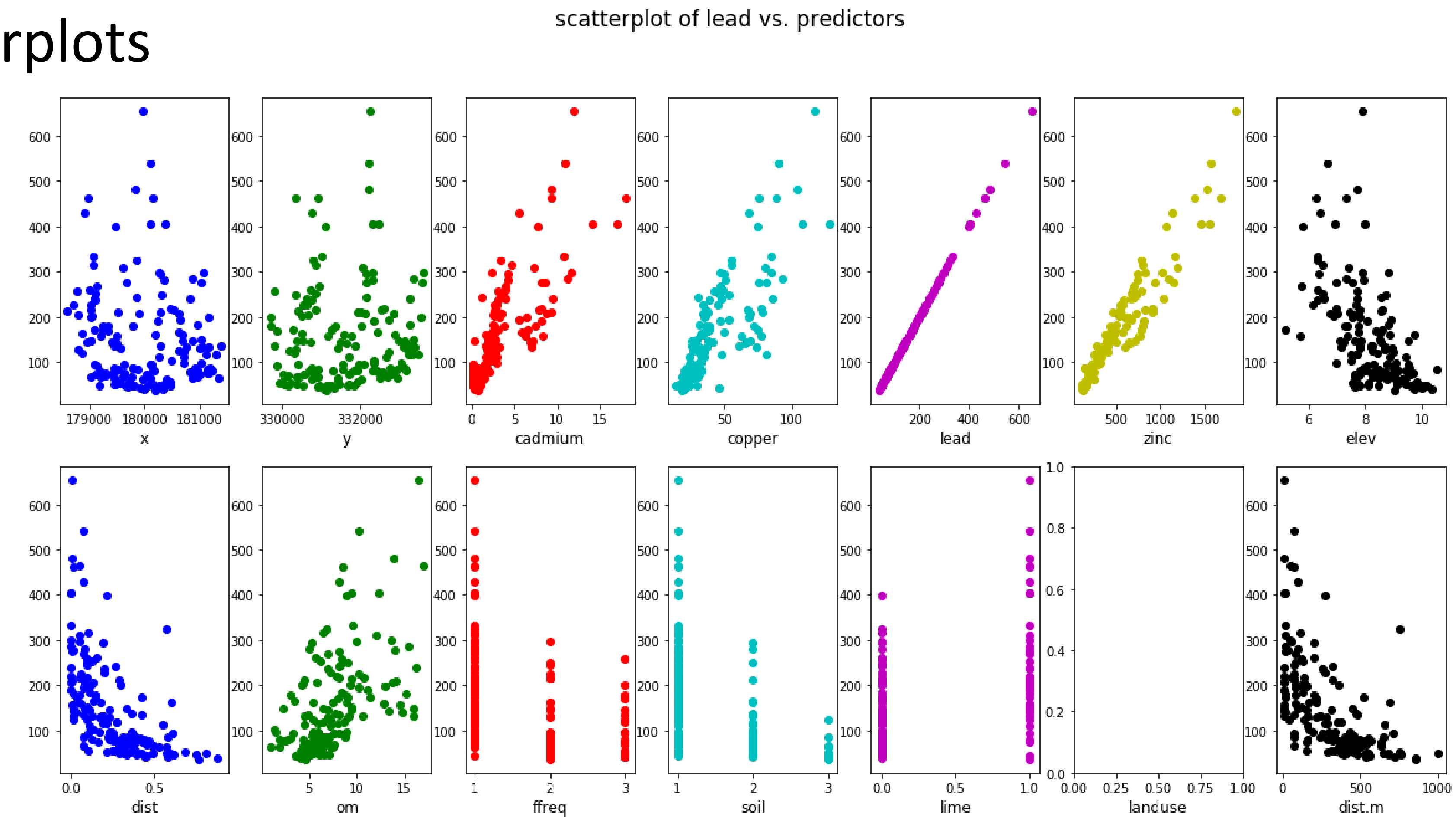
Exploratory Data Analysis (EDA)

- boxplots



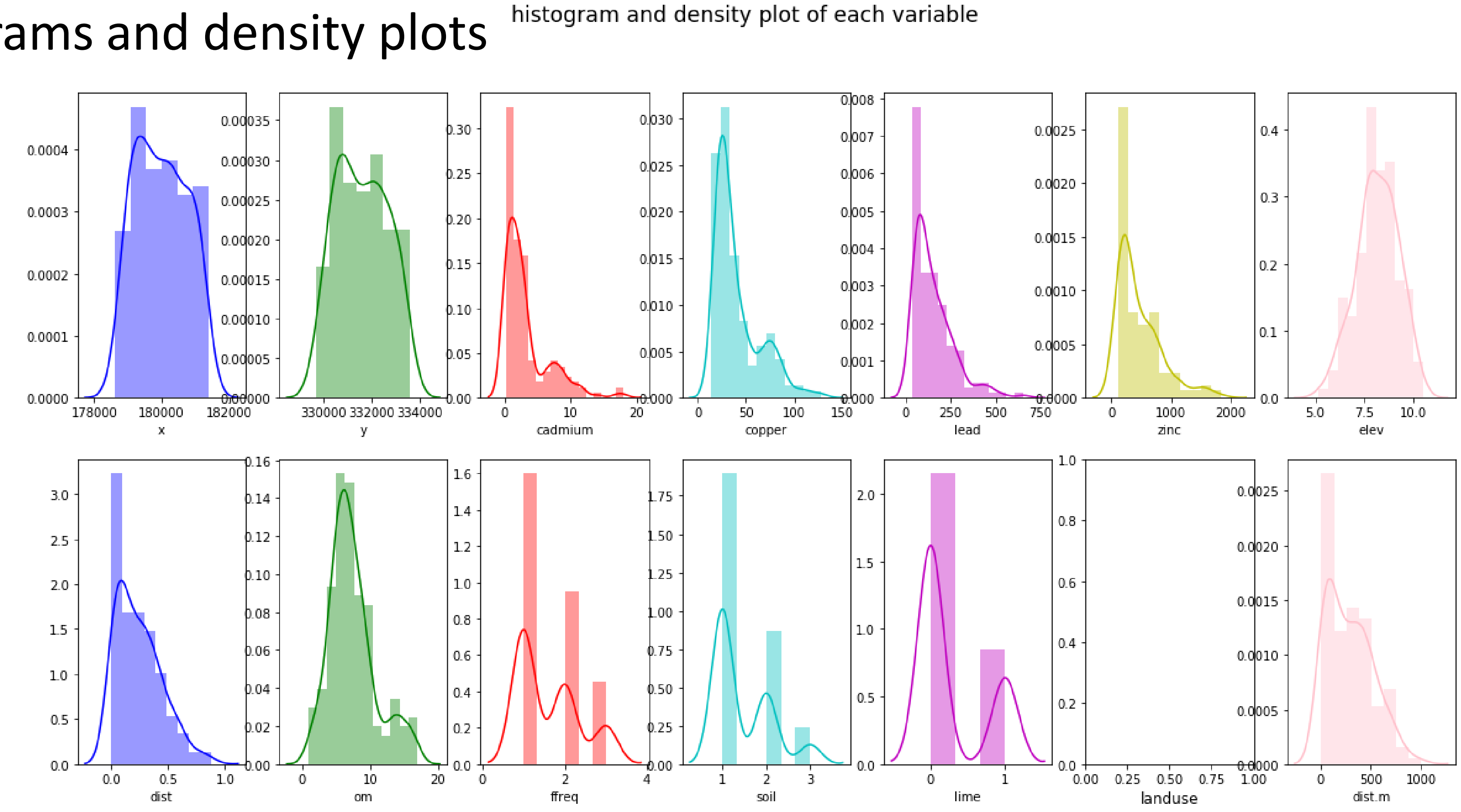
Exploratory Data Analysis (EDA)

- scatterplots



Exploratory Data Analysis (EDA)

- histograms and density plots



Linear Regression Modeling

- Variable selection
- Summary statistics
- Diagnostics and assumption checking
- Model selection

Linear Regression Modeling

- Variable selection: based on correlation coefficients

```
Lead ~ cadmium + copper + zinc + elev + lime
```

- Split the dataset into training and testing sets:

```
In [20]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.33, random_state = 42)
```

- Fit a linear model on training set: using `scikit-learn` module

```
In [21]: model = LinearRegression().fit(X_train, y_train)

In [22]: # R^2 on training set
         R2_train = model.score(X_train, y_train)
         R2_train

Out[22]: 0.9483947080189404

In [23]: # R^2 on testing set
         R2_test = model.score(X_test, y_test)
         R2_test

Out[23]: 0.964901551899052
```

Linear Regression Modeling

`Lead ~ cadmium + copper + zinc + elev + lime` (using `statsmodel` module)

In [27]:
```python
fullModel = fullModel.fit()
print(fullModel.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   lead   R-squared:                       0.948
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     346.6
Date:                Mon, 11 Nov 2019   Prob (F-statistic):           2.35e-59
Time:                        15:13:12   Log-Likelihood:                -467.76
No. Observations:                 101   AIC:                             947.5
Df Residuals:                      95   BIC:                             963.2
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       23.8739     27.200      0.878      0.382     -30.126      77.874
C(lime)[T.1]   -24.7610      7.775     -3.185      0.002     -40.196      -9.326
cadmium        -13.2322      2.240     -5.908      0.000     -17.679      -8.785
copper           0.0522      0.278      0.187      0.852      -0.500       0.605
zinc             0.4188      0.020     20.756      0.000       0.379       0.459
elev            -2.4719      2.895     -0.854      0.395      -8.220       3.276
==============================================================================
Omnibus:                        2.077   Durbin-Watson:                   1.947
Prob(Omnibus):                  0.354   Jarque-Bera (JB):                1.494
Skew:                          -0.246   Prob(JB):                        0.474
Kurtosis:                       3.336   Cond. No.                     6.42e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.42e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Linear Regression Modeling

`Lead ~ cadmium + copper + zinc + elev + lime` (using `statsmodel` module)

In [27]:
```python
fullModel = fullModel.fit()
print(fullModel.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   lead   R-squared:                       0.948
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     346.6
Date:                Mon, 11 Nov 2019   Prob (F-statistic):           2.35e-59
Time:                        15:13:12   Log-Likelihood:                -467.76
No. Observations:                 101   AIC:                             947.5
Df Residuals:                      95   BIC:                             963.2
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       23.8739     27.200      0.878      0.382     -30.126      77.874
C(lime)[T.1]   -24.7610      7.775     -3.185      0.002     -40.196      -9.326
cadmium        -13.2322      2.240     -5.908      0.000     -17.679      -8.785
copper           0.0522      0.278      0.187      0.852      -0.500       0.605
zinc             0.4188      0.020     20.756      0.000       0.379       0.459
elev            -2.4719      2.895     -0.854      0.395      -8.220       3.276
==============================================================================
Omnibus:                        2.077   Durbin-Watson:                   1.947
Prob(Omnibus):                  0.354   Jarque-Bera (JB):                1.494
Skew:                          -0.246   Prob(JB):                        0.474
Kurtosis:                       3.336   Cond. No.                     6.42e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.42e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

- R^2_{adj} of this model is around **94.5%.**
- Statistical test on each predictor:
  - **cadmium: significant**
  - copper: not significant
  - **zinc: significant**
  - elev: not significant
  - **lime: significant**

Linear Regression Modeling

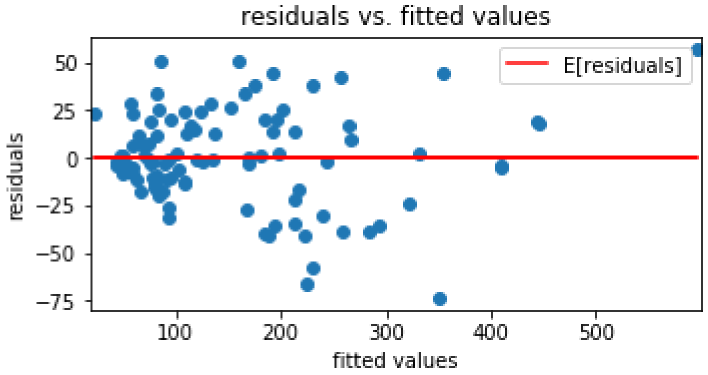
- Summary statistics

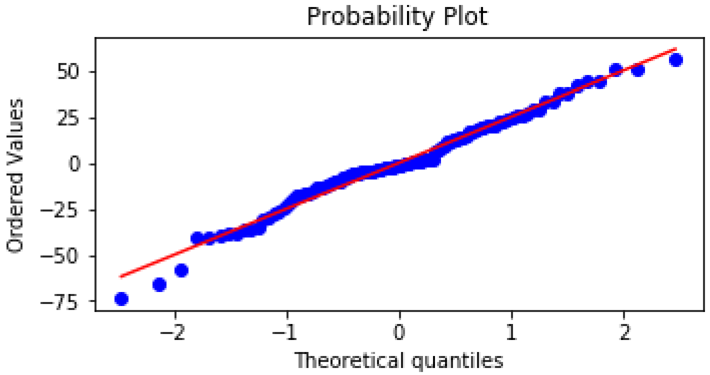
```
In [31]: table1 = RMSETable(y_train.values, model.predict(X_train), R2_train, len(y_train),
                            y_test.values, model.predict(X_test), R2_test, len(y_test))
         print(table1)

                    RMSE  R-squared  size
training set  262.286912   0.948395   101
testing set   159.247070   0.964902    51
```

- Lower RSME on testing set
- Higher R^2 on testing set

Linear Regression Modeling

- Diagnostics and assumption checking





- Assumption of constant variance: satisfied
- Assumption that E[residuals] = 0: satisfied
- Assumption of normality of the response: satisfied

Linear Regression Modeling

- Model selection:
  - We used 5 predictors in our previous model, but some of the predictors are not statistically significant compared with others.
  - We can consider reducing the number of predictors to improve the model's prediction performance, by selecting only a subset of these 5 predictors.
  - Since `cadmium`, `zinc` and `lime` are highly statistically significant, we now refit a model using only these 3 predictors:
    - Full model: `Lead ~ cadmium + copper + zinc + elev + lime`
    - Reduced model: `Lead ~ cadmium + zinc + lime`

Linear Regression Modeling

- **Summary statistics:** `Lead ~ cadmium + zinc + lime` (using `statsmodel` module)

In [35]:
```
df_train = pd.concat([X_train, y_train], axis = 1) # build a dataframe for training set
reducedModel = smf.ols("lead ~ cadmium + zinc + C(lime)", data = df_train)
reducedModel = reducedModel.fit()
print(reducedModel.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   lead   R-squared:                       0.948
Model:                            OLS   Adj. R-squared:                  0.946
Method:                 Least Squares   F-statistic:                     584.7
Date:                Fri, 08 Nov 2019   Prob (F-statistic):           5.98e-62
Time:                        23:31:40   Log-Likelihood:                -468.19
No. Observations:                 101   AIC:                             944.4
Df Residuals:                      97   BIC:                             954.8
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        2.6976      4.614      0.585      0.560      -6.461      11.856
C(lime)[T.1]   -23.4871      7.569     -3.103      0.003     -38.509      -8.465
cadmium        -13.0934      1.986     -6.593      0.000     -17.035      -9.152
zinc             0.4237      0.019     22.540      0.000       0.386       0.461
==============================================================================
Omnibus:                        0.749   Durbin-Watson:                   1.968
Prob(Omnibus):                  0.688   Jarque-Bera (JB):                0.395
Skew:                          -0.131   Prob(JB):                        0.821
Kurtosis:                       3.160   Cond. No.                     1.79e+03
==============================================================================
```

Linear Regression Modeling

- Summary statistics: `Lead ~ cadmium + zinc + lime` (using `statsmodel` module)

```
In [35]: df_train = pd.concat([X_train, y_train], axis = 1) # build a dataframe for training set
         reducedModel = smf.ols("lead ~ cadmium + zinc + C(lime)", data = df_train)
         reducedModel = reducedModel.fit()
         print(reducedModel.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   lead   R-squared:                       0.948
Model:                            OLS   Adj. R-squared:                  0.946
Method:                 Least Squares   F-statistic:                     584.7
Date:                Fri, 08 Nov 2019   Prob (F-statistic):           5.98e-62
Time:                        23:31:40   Log-Likelihood:                -468.19
No. Observations:                 101   AIC:                             944.4
Df Residuals:                      97   BIC:                             954.8
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        2.6976      4.614      0.585      0.560      -6.461      11.856
C(lime)[T.1]   -23.4871      7.569     -3.103      0.003     -38.509      -8.465
cadmium        -13.0934      1.986     -6.593      0.000     -17.035      -9.152
zinc             0.4237      0.019     22.540      0.000       0.386       0.461
==============================================================================
Omnibus:                        0.749   Durbin-Watson:                   1.968
Prob(Omnibus):                  0.688   Jarque-Bera (JB):                0.395
Skew:                          -0.131   Prob(JB):                        0.821
Kurtosis:                       3.160   Cond. No.                     1.79e+03
==============================================================================
```

- R^2_{adj} of this model is around **94.6%,** higher than the previous model.

- Statistical test on each predictor:
  - **cadmium: significant**
  - **zinc: significant**
  - **lime: significant**

Linear Regression Modeling

- Summary statistics of the reduced model

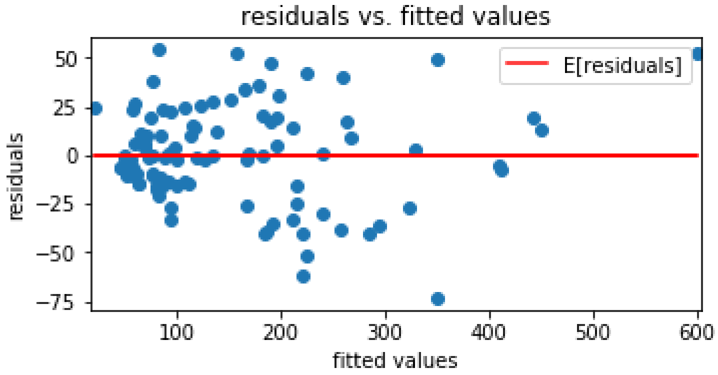
```
In [37]: table2 = RMSETable(y_train.values, model2.predict(X_train), R2_train, len(y_train),
                            y_test.values, model2.predict(X_test), R2_test, len(y_test))
         print(table2)
```

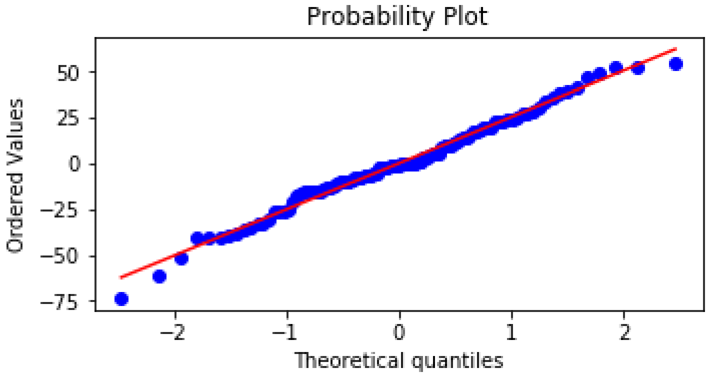
```
                    RMSE  R-squared  size
training set  264.533494   0.947595   101
testing set   168.763005   0.960907    51
```

- Lower RSME on testing set
- Higher R^2 on testing set

Linear Regression Modeling

- Diagnostics of the reduced model:





- Assumption of constant variance: satisfied
- Assumption that E[residuals] = 0: satisfied
- Assumption of normality of the response: satisfied

Linear Regression Modeling

Comparison of the reduced and full model:

- RMSE (*root-mean-squared error*) and R^2 of two models:

  - model with 5 predictors (full model)

In [38]:
```
print(table1)
```

```
                    RMSE  R-squared  size
training set  262.286912   0.948395   101
testing set   159.247070   0.964902    51
```

  - model with 3 predictors (reduced model)

In [39]:
```
print(table2)
```

```
                    RMSE  R-squared  size
training set  264.533494   0.947595   101
testing set   168.763005   0.960907    51
```

Linear Regression Modeling

Comparison of the reduced and full model:

- RMSE (*root-mean-squared error*) of the two models:

  - model with 5 predictors (full model)

In [38]:
```
print(table1)
```

```
                    RMSE  R-squared  size
training set  262.286912   0.948395   101
testing set   159.247070   0.964902    51
```

  - model with 3 predictors (reduced model)

In [39]:
```
print(table2)
```

```
                    RMSE  R-squared  size
training set  264.533494   0.947595   101
testing set   168.763005   0.960907    51
```

lower RMSE and higher R^2 on the testing set

Linear Regression Modeling

Comparison of the reduced and full model:

- To decide whether to adopt the reduced model, we can conduct one-way **ANOVA** (*Analysis of Variance*) on the reduced and full model:

ANOVA of between the reduced and full model

```
In [40]: table = sm.stats.anova_lm(reducedModel, fullModel) # Type 2 ANOVA DataFrame
         table # ANOVA table
```

Out[40]:

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	97.0	62835.802512	0.0	NaN	NaN	NaN
1	95.0	62309.421080	2.0	526.381432	0.401273	0.670596

Linear Regression Modeling

Comparison of the reduced and full model:

- To decide whether to adopt the reduced model, we can conduct one-way **ANOVA** (*Analysis of Variance*) on the reduced and full model:

ANOVA of between the reduced and full model

In [40]:
```
table = sm.stats.anova_lm(reducedModel, fullModel) # Type 2 ANOVA DataFrame
table # ANOVA table
```

Out[40]:

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	97.0	62835.802512	0.0	NaN	NaN	NaN
1	95.0	62309.421080	2.0	526.381432	0.401273	0.670596

p-value > 0.05:
no significance difference between two models: adopt the reduced model

Discussion

- We observe that there is no significance different between the reduced model and the full model (based on the ANOVA test).
- Therefore, we will adopt the MLR model with 3 predictor variables:

```
lead ~ cadmium + lime + zinc
```

- Possible improvements:
  - Consider fitting a nonlinear model (although this increases model complexity)
  - Consider transforming the response variable and some predictor variables
  - Consider creating new features based on current predictor variables to bring down model complexity

Reference

- Dataset: `meuse` dataset in `R`
- To access the dataset in `R`:

```
install.package("sp") # required when first time using the library
library(sp)
data(meuse)
```

Tips

To learn more about linear regression and machine learning:
go to *OSCR*'s webpage at https://oscrproject.wixsite.com/website