

12. Juli 2015

# Aufgabenblatt 3

## *Gruppe 05*

Dora Szücs – 358573 – BA Informatik

Sarah Köhler – 356394 – BA Informatik

Christina Pavlidis – 356230 – BA Economics

Daniela Andrzejewska – 356254 – BA Economics

Charlotte Rochell – 348926 – BA Economics

Tutorium 3: Mittwoch 10 – 12, H3008

Tutor: Marius

## Aufgabe 2: OLAP und OLTP

1. OLTP: Es werden viele transaktionale Updates gemacht, wenn Gutscheine als eingelöst markiert werden, etc.
2. OLAP: Vergleich der Gutscheineinlösungszahlungen der Abteilungen und der wöchentlicher Vergleich der Verkäufer nach Gutscheinverkäufen
3. OLTP: viele transaktionale Updates um die Kundeninfos in das System einzutragen
4. OLAP: Analyse der Kundendaten und Verkaufszahlen um eine ideale Marketingstrategie zu entwickeln

## Aufgabe 5

ID	quantitativ, diskret	Zahlen oder Größenwerte; Nur ganze Werte angenommen werden
Alter	quantitativ, diskret	Zahlen oder Größenwerte; Nur ganze Werte angenommen werden
Nationalität	qualitativ, nominal	Namen oder Eigenschaften; Reine qualitative Merkmalsausprägung ohne natürliche Ordnung
Verheiratet?	qualitativ, binär	Namen oder Eigenschaften; Dummy-Variable mit Ausprägungen 0 oder 1 ( hier ja oder nein)
Anzahl der Kinder	quantitativ, diskret	Zahlen oder Größenwerte; Nur ganze Werte angenommen werden
Beruf	qualitativ, nominal	Namen oder Eigenschaften; Reine qualitative Merkmalsausprägung ohne natürliche Ordnung
Einkommen	qualitativ, ordinal	Namen oder Eigenschaften; Ausdruck einer Rangfolge ( hier Einkommen)
Credit Score	quantitativ, kontinuierlich	Zahlen oder Größenwerte; Zahlen mit Nachkommastellen ( jede beliebige Wert aus der Menge der reellen Zahlen)

# Aufgabe 7

## Import der Bibliotheken & Quelldaten

In [2]:

```
# Standard Python Math Bibliothek.
import math
# Lädt Matplotlib, eine Bibliothek zum Erstellen von Plots & Diagrammen.
import matplotlib.pyplot as plt
# Matplotlib Plots sollen direkt im Notebook erscheinen.
%matplotlib inline
# Lädt NumPy, eine Bibliothek zur numerischen Analyse.
import numpy as np
np.random.seed(102)
# Lädt Pandas, eine Bibliothek zur Analyse von strukturierten Daten.
import pandas as pd
```

In [38]:

```
housing_df = pd.read_csv("housing.csv", sep=';', index_col=False)
print "Housing:\n", housing_df.dtypes
```

Housing:

```
crim      float64
zn        float64
indus     float64
chas      int64
nox       float64
rm        float64
age       float64
dis       float64
rad       float64
tax       float64
ptratio   float64
b         float64
lstat     float64
medv      float64
dtype: object
```

## 1. Basisstatistiken der Stickstoffkonzentration:

In [12]:

```
max_nox = housing_df.nox.max()
print "Maximum: ", max_nox
min_nox = housing_df.nox.min()
print "Minimum: ", min_nox
mean_nox = housing_df.nox.mean()
print "Mittelwert: ", mean_nox
median_nox = housing_df.nox.median()
print "Median: ", median_nox
stderr_nox = np.sqrt(housing_df.nox.var())
print "Standardabweichung: ", stderr_nox
```

```
Maximum:  0.871
Minimum:  0.385
Mittelwert:  0.554695059289
Median:  0.538
Standardabweichung:  0.115877675668
```

## 2. Stickstoffkonzentration und Kriminalität

In [21]:

```
avg_crim_over = housing_df[housing_df.nox > 0.53].crim.mean()
avg_crim_below = housing_df[housing_df.nox <= 0.53].crim.mean()

print "Durchschnittliche Kriminalitätsrate für Bezirke über dem Grenzwert:
", avg_crim_over
print "Durchschnittliche Kriminalitätsrate für Bezirke unter dem Grenzwert:
", avg_crim_below
```

```
Durchschnittliche Kriminalitätsrate für Bezirke über dem Grenzw
ert:  6.86128145038
Durchschnittliche Kriminalitätsrate für Bezirke unter dem Grenz
wert:  0.126176967213
```

## 3. Verhaltensändernde Wirkung der Stickstoffkonzentration

Nein: Korrelation impliziert keinen kausalen Zusammenhang

Es ist ebenso möglich, dass andere Faktoren, die wir nicht untersucht haben für den Zusammenhang verantwortlich sind.

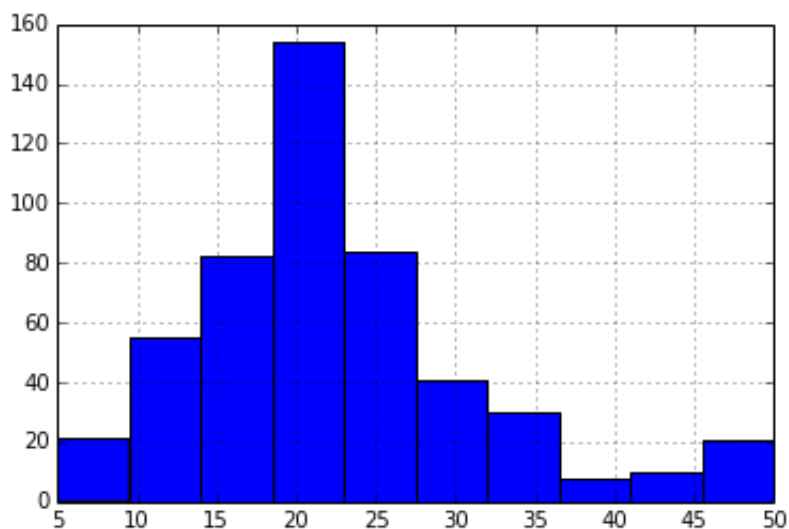
## 4. Mittlerer Wert von Häusern

In [22]:

```
housing_df.medv.hist()
```

Out[22]:

<matplotlib.axes.AxesSubplot at 0x10b9a2210>



Aus dem Histogramm lassen sich die Häufigkeiten (y-Achse) der verschiedenen Wertbereiche für Häuser (x-Achse) pro Vorort ablesen. Beispielsweise gibt es knapp über 150 Vororte wo der durchschnittliche Wert der Häuser bei knapp über 20000 Dollar liegt. Das Histogramm scheint in etwa eine Normalverteilung abzubilden, allerdings ist auffällig, dass es im Bereich der sehr hohen durchschnittlichen Werte noch einen Aussschlag nach oben gibt.

## 5. Korrelationskoeffizient des Mittleren Wertes der Häuser

In [39]:

```
korr_crim = np.corrcoef(housing_df.medv, housing_df.crim)[0, 1]
print "Pearson's Korrelationskoeffizient für crim: ", korr_crim
zn_crim = np.corrcoef(housing_df.medv, housing_df.zn)[0, 1]
print "Pearson's Korrelationskoeffizient für zn: ", zn_crim
korr_indus = np.corrcoef(housing_df.medv, housing_df.indus)[0, 1]
print "Pearson's Korrelationskoeffizient für indus: ", korr_indus
korr_chas = np.corrcoef(housing_df.medv, housing_df.chas)[0, 1]
print "Pearson's Korrelationskoeffizient für chas: ", korr_chas
korr_nox = np.corrcoef(housing_df.medv, housing_df.nox)[0, 1]
print "Pearson's Korrelationskoeffizient für nox: ", korr_nox
korr_rm = np.corrcoef(housing_df.medv, housing_df.rm)[0, 1]
print "Pearson's Korrelationskoeffizient für rm: ", korr_rm
korr_age = np.corrcoef(housing_df.medv, housing_df.age)[0, 1]
print "Pearson's Korrelationskoeffizient für age: ", korr_age
korr_dis = np.corrcoef(housing_df.medv, housing_df.dis)[0, 1]
print "Pearson's Korrelationskoeffizient für dis: ", korr_dis
korr_rad = np.corrcoef(housing_df.medv, housing_df.rad)[0, 1]
print "Pearson's Korrelationskoeffizient für rad: ", korr_rad
korr_tax = np.corrcoef(housing_df.medv, housing_df.tax)[0, 1]
print "Pearson's Korrelationskoeffizient für tax: ", korr_tax
korr_ptratio = np.corrcoef(housing_df.medv, housing_df.ptratio)[0, 1]
print "Pearson's Korrelationskoeffizient für ptratio: ", korr_ptratio

korr_b = np.corrcoef(housing_df.medv, housing_df.b)[0, 1]
print "Pearson's Korrelationskoeffizient für b: ", korr_b
korr_lstat = np.corrcoef(housing_df.medv, housing_df.lstat)[0, 1]
print "Pearson's Korrelationskoeffizient für lstat: ", korr_lstat
```

```
Pearson's Korrelationskoeffizient für crim:  -0.388304608587
Pearson's Korrelationskoeffizient für zn:   0.360445342451
Pearson's Korrelationskoeffizient für indus: -0.483725160028
Pearson's Korrelationskoeffizient für chas:  0.17526017719
Pearson's Korrelationskoeffizient für nox:   -0.427320772373
Pearson's Korrelationskoeffizient für rm:    0.695359947072
Pearson's Korrelationskoeffizient für age:   -0.376954565005
Pearson's Korrelationskoeffizient für dis:   0.249928734086
Pearson's Korrelationskoeffizient für rad:   -0.38162623064
Pearson's Korrelationskoeffizient für tax:   -0.468535933568
Pearson's Korrelationskoeffizient für ptratio: -0.507786685538
Pearson's Korrelationskoeffizient für b:     0.333460819657
Pearson's Korrelationskoeffizient für lstat: -0.737662726174
```

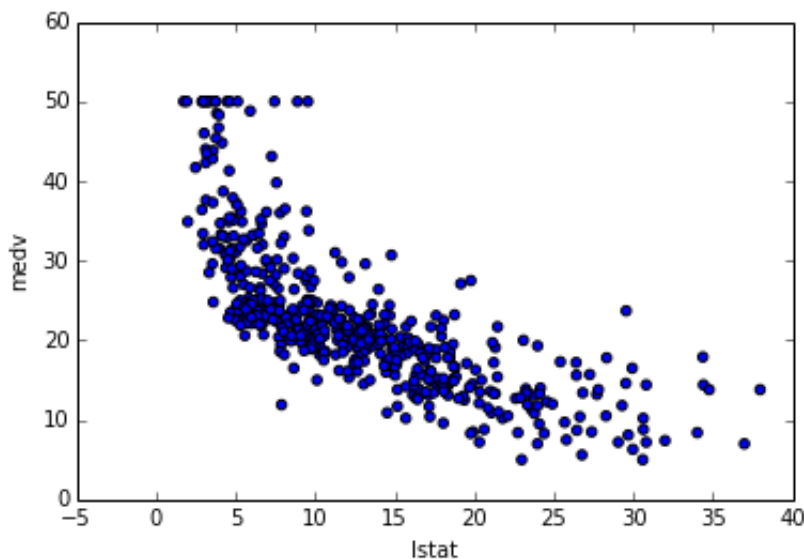
## 6. Scatterplot des Zusammenhangs zwischen MEDV und LSTAT

In [26]:

```
housing_df.plot(kind='scatter', x='lstat', y='medv')
```

Out[26]:

<matplotlib.axes.AxesSubplot at 0x10eb60810>



Dass die beiden Parameter negativ korreliert sind, bedeutet, dass in Vororten mit höheren durchschnittlichen Hauspreisen niedrigere Anteile einkommensschwacher Personen leben und umgekehrt.

## 7. Zusammenhang von Crim, Nox und Lstat

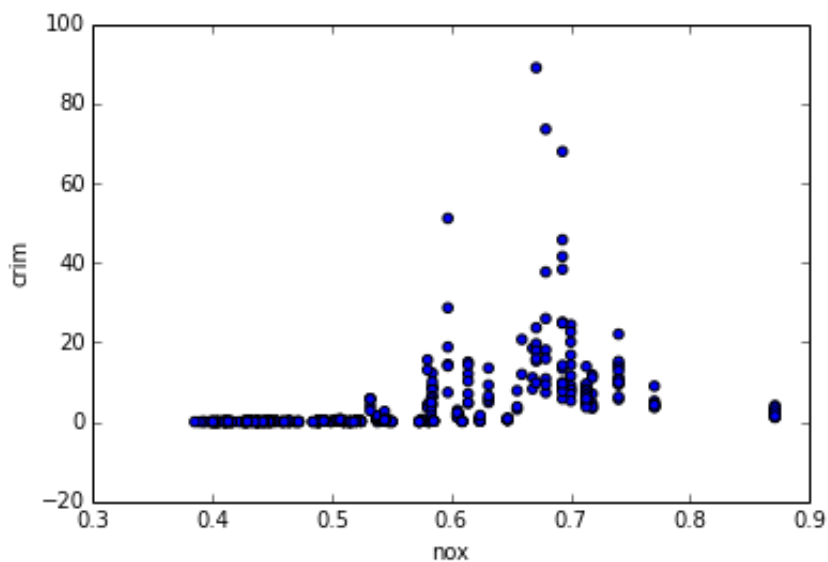
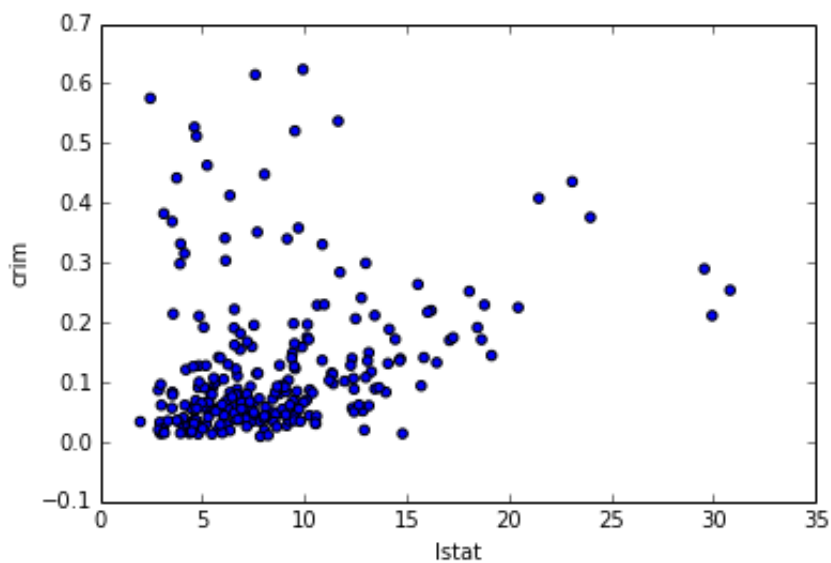
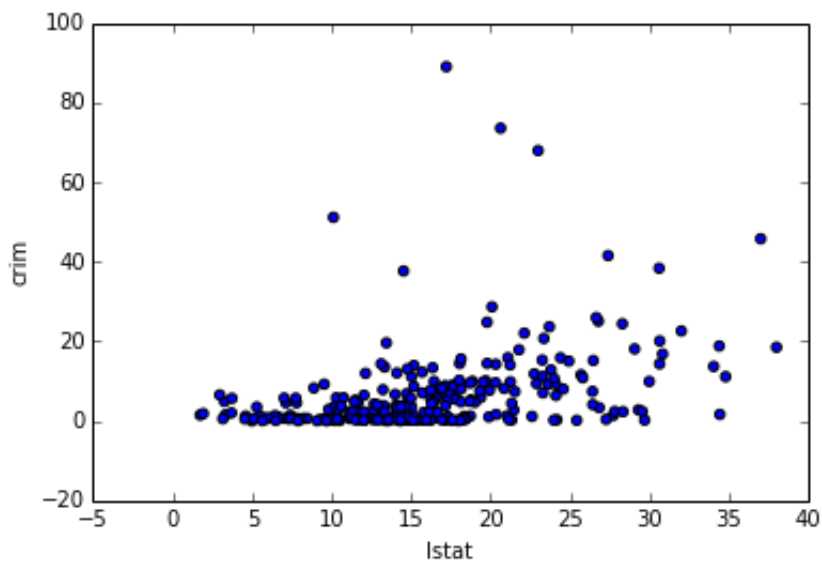
In [40]:

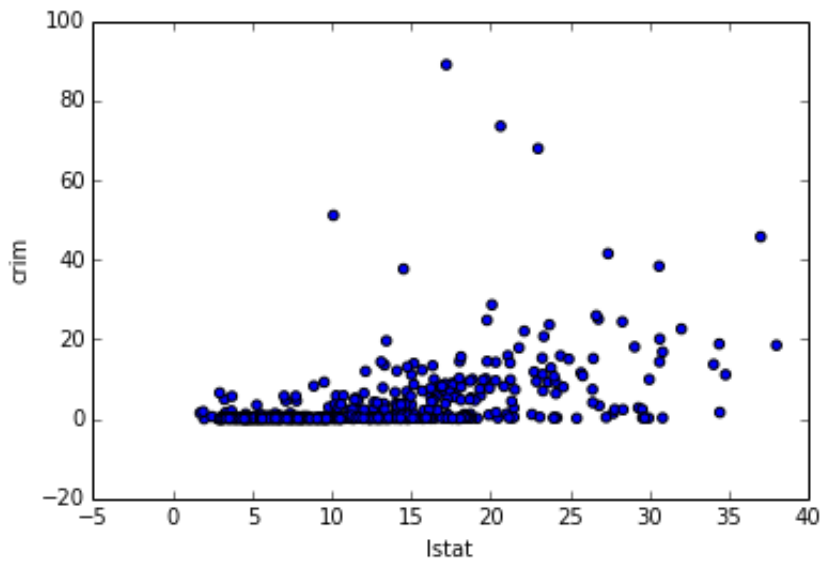
```
housing_nox_over = housing_df[housing_df.nox > 0.53]
housing_nox_over.plot(kind='scatter', x='lstat', y='crim')
housing_nox_below = housing_df[housing_df.nox <= 0.53]
housing_nox_below.plot(kind='scatter', x='lstat', y='crim')

housing_df.plot(kind='scatter', x='nox', y='crim')
housing_df.plot(kind='scatter', x='lstat', y='crim')
korr_lstat_crim= np.corrcoef(housing_df.crim, housing_df.lstat)[0, 1]
korr_nox_crim= np.corrcoef(housing_df.crim, housing_df.nox)[0, 1]
korr_lstat_nox= np.corrcoef(housing_df.nox, housing_df.lstat)[0, 1]
print "Korrelationskoeffizient von Kriminalität und niedrigem Einkommen: ",
korr_lstat_crim
print "Korrelationskoeffizient von Kriminalität und Stickstoffkonzentration", korr_nox_crim
print "Korrelationskoeffizient von niedrigem Einkommen und Stickstoffkonzentration", korr_lstat_nox
```



Korrelationskoeffizient von Kriminalität und niedrigem Einkomme  
n: 0.455621479448  
Korrelationskoeffizient von Kriminalität und Stickstoffkonzentr  
ation 0.420971711392  
Korrelationskoeffizient von niedrigem Einkommen und Stickstoffk  
onzentration 0.590878920881





Da die Korrelationskoeffizienten für die Zusammenhänge zwischen den Hauspreisen, dem Anteil von Personen mit niedrigem Einkommen und der Konzentration der Stickoxide sehr ähnlich sind, ist zu vermuten, dass der kausale Zusammenhang nicht zwischen NOX und CRIM besteht, sondern die Ursache jeweils eine andere Größe ist. Ein möglicher Zusammenhang wäre etwa über die Lage im Verhältnis zu Industriegebieten: Bezirke nahe einem Industriegebiet haben eine höhere Konzentration an Schadstoffen in der Luft. Zudem wohnen dort eher die Arbeiter aus den Fabriken, welche eher zu den einkommensschwachen Haushalten zählen.