Skilaverkefni07. Stærð- og Reiknifræði REI201G

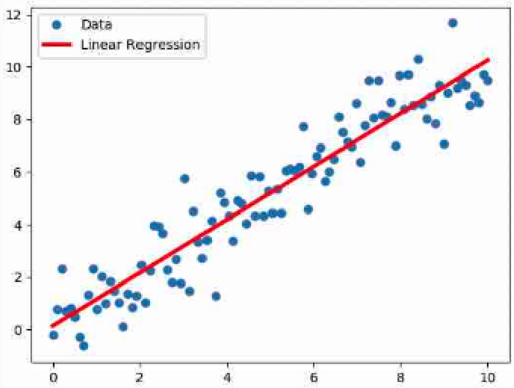
Donn Eunice Bartido deb5@hi.is

febrúar 2023

25. Jafna bestu línu

25 Jafna bestu línu

Oft er línulegt samband á milli gagna, þannig að breytur stækka samhliða, en þó með ákveðinni skekkju, sbr. myndina hér til hliðar.



Með **aðferð minnstu kvaðrata** (*linear least squares method*) er fundin **jafna bestu línu** með því að lágmarka kvaðratsummu fjarlægða milli línunnar og gefins punktasafns, eins og lesa má um á Wikipedíu. Það er líka hægt að finna **jöfnu bestu parabólu** eins og sýnt er aðeins aftar í sömu Wikipedíugrein.

Eftirfarandi skipanir:

```
(a,b) = np.polyfit(u, v, deg=1)
(A,B,C) = np.polyfit(u, v, deg=2)
```

skila stuðlum í jöfnu bestu línu og jöfnu bestu parabólu fyrir punktasafnið $(u_0, v_0), (u_1, v_1) \dots$, þannig að jöfnur línunnar og parabólunnar eru:

$$y = ax + b$$
$$y = Ax^2 + Bx + C$$

Til að teikna línuna eða parabóluna er hægt að nota aðferðir kafla 9.6, en reyndar er hægt að einfalda málið með því að notfæra sér numpy reikninga. Til teikna línuna á bilinu $[x_{\min}, x_{\max}]$ má t.d. nota:

```
xp = np.linspace(xmin, xmax)
yp = a*xp + b
plt.plot(xp,yp,...)
```

Reyndar þyrfti ekki nema tvo punkta á línunni, til dæmis `xp = np.linspace(xmin, ymin, 2)` (eða `xp = np.array([xmin,ymin])`), en parabólan þarf fleiri punkta.

1. Skoðið framangreinda Wikipedugrein.

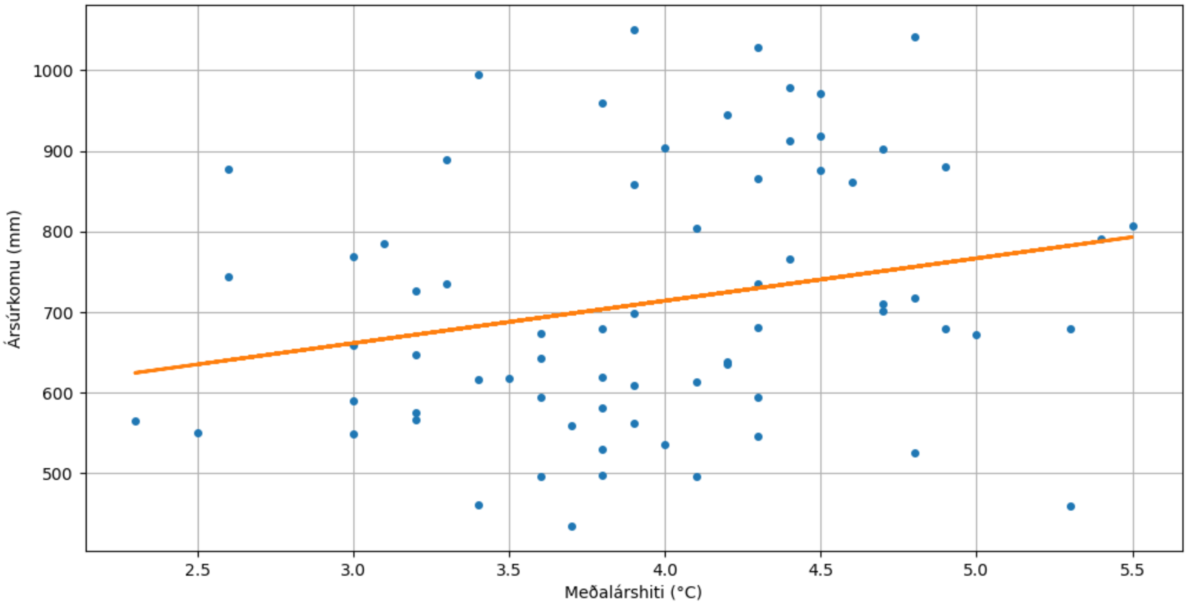
2. Í verkefni 19e voru teiknaðar myndir af meðalárshita og ársúrkomu í Stykkishólmi 1949–2018. Gögnin eru í skránni cs.hi.is/python/hiti-urkoma.txt og þau má lesa inn í þrjá vigra með:

(ár, hiti, úrk) = np.loadtxt("https://cs.hi.is/python/hiti-urkoma.txt").T Gerið það og ákvarðið jöfnu bestu línu fyrir hita á x-ás og úrkomu á y-ás. Teiknið bæði gögn og línu inn á sömu mynd. Látið myndina hafa stærð (12,6).

3. Teiknið nú punktarit af ári og ársmeðalhita, og teiknið inn á hana bæði bestu línu og bestu parabólu. Setjið inn hæfilegar merkingar, m.a. ramma með legend-skýringum.

Lausn við 2.

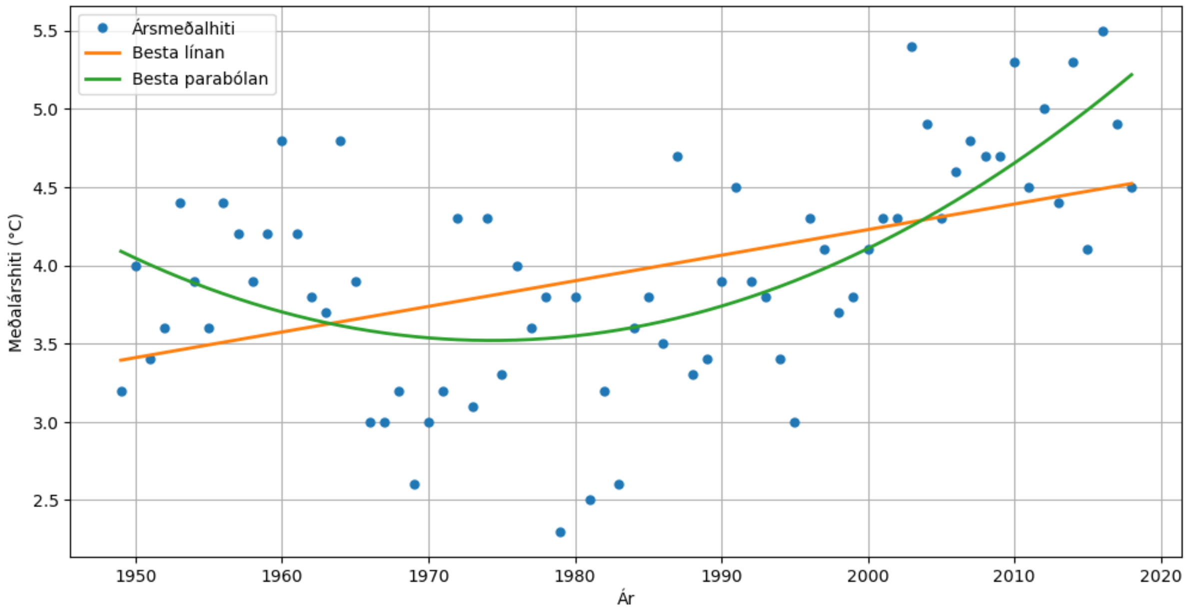
In [3]:
```
import numpy as np
import matplotlib.pyplot as plt

# sótt gögnin
(ár, hiti, úrk) = np.loadtxt("https://cs.hi.is/python/hiti-urkoma.txt").T

# ákvarða jöfnu bestu línu fyrir hita á x-ás og úrkomu á y-ás
m, b = np.polyfit(hiti, úrk, 1)

# teikna gögnin og línu
plt.figure(figsize=(12,6))
plt.plot(hiti, úrk, 'o', markersize=4)
plt.plot(hiti, m*hiti + b, '-', linewidth=2)
plt.xlabel('Meðalárshiti (°C)')
plt.ylabel('Ársúrkomu (mm)')
plt.grid(True)
plt.show()
```



Lausn við 3.

In [4]:
```
import numpy as np
import matplotlib.pyplot as plt

# sótt gögnin
(ár, hiti, úrk) = np.loadtxt("https://cs.hi.is/python/hiti-urkoma.txt").T

# ákvarða jöfnu bestu línu fyrir ársmeðalhita
lin_m, lin_b = np.polyfit(ár, hiti, 1)

# ákvarða jöfnu bestu parabólu fyrir ársmeðalhita
```

```
quad_a, quad_b, quad_c = np.polyfit(ár, hiti, 2)

# teikna punktarit af ári og ársmeðalhita
plt.figure(figsize=(12,6))
plt.plot(ár, hiti, 'o', markersize=5, label='Ársmeðalhiti')

# teikna bestu línu fyrir ársmeðalhita
plt.plot(ár, lin_m*ár + lin_b, '-', linewidth=2, label='Besta línan')

# teikna bestu parabólu fyrir ársmeðalhita
quad_fit = quad_a*ár**2 + quad_b*ár + quad_c
plt.plot(ár, quad_fit, '-', linewidth=2, label='Besta parabólan')

plt.xlabel('Ár')
plt.ylabel('Meðalárshiti (°C)')
plt.grid(True)
plt.legend()
plt.show()
```



28.Kínverskt Jöfnuhneppi

Hér er dæmi úr gömlu kínversku handriti frá því á fyrstu öld f.Kr. Leysið það með NumPy:

Nokkrir ætla að leggja í púkk til að kaupa hlut. Ef hver borgar átta peninga eru þrír peningar afgangs en ef hver borgar sjö vantar fjóra upp á. Hve margir eru kaupendur og hvað kostar hluturinn?

Lausn við 28

Við erum með 2x jöfnur

x = peningurinn hver manneskja leggur inn í púkkið y = er fyrir heildarkostnað af hlutnum sem að þau pútta inn fyrir

- a) y = x * 8 - 3
- b) y = x * 7 + 4

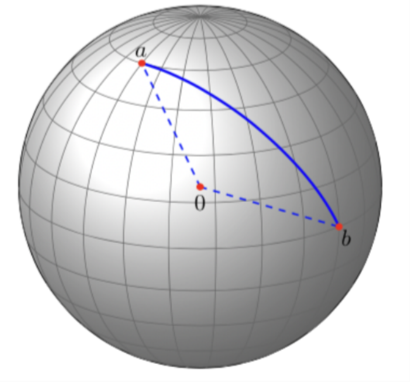
In [ ]:
```python
import numpy as np

# Skilgreinum jöfnurnar sem við viljum leysa
A = np.array([[8, -1], [7, -1]])
B = np.array([3, -4])

# Leysum jöfnurnar með numpy.linalg.solve()
x = np.linalg.solve(A, B)

# Fjöldi hluta í púkknum er heiltala
num_items = int(x[0])

# Heildarverð hlutarins
total_price = 8 * num_items - 3

# Prentum niðurstöðurnar
print("Fjöldi hluta í púkknum: ", num_items)
print("Heildarverð hlutarins: ", total_price)
```

```
Fjöldi hluta í púkknum:  7
Heildarverð hlutarins:  53
```

29. Fjarlægð milli borga

29 Fjarlægð milli borga

Ef sett er hnitakerfi í gegn um miðju jarðar þá verður þrívíður vigur sem byrjar í miðjunni og endar á stað á yfirborðinu með lengd l og breidd b:



$$\begin{pmatrix} R \sin l \cos b \\ R \cos l \cos b \\ R \sin b \end{pmatrix}$$

þar sem R er radíus jarðar, $R \approx 6370$ km. Í stærðfræðigreiningu eru R, l og b kölluð kúluhnit (*spherical coordinates*). Hnattstaða Reykjavíkur er u.þ.b. 64°N og 22°W, og hattstaða London u.þ.b. 51.5°N og 0°W.

Horn θ milli vigra x og y má reikna með:

$$\theta = \arccos \frac{x \cdot y}{\|x\| \|y\|}$$

1. Reiknið hornið milli Reykjavíkur og London mælt frá miðju jarðar.
2. Lengd hringboga í hring með radíus r sem spannar horn sem er θ radíanar er

$$L = \theta r$$

Lausn við 29.1

In [22]:
```python
import math

# Hnattstaða Reykjavíkur
l_reykjavik = math.radians(64) # breidd í radíönum
b_reykjavik = math.radians(-22) # lengd í radíönum
```

```
# Hnattstaða London
l_london = math.radians(51.5) # breidd í radíönum
b_london = math.radians(0) # lengd í radíönum

# Reiknum horn milli borganna í radíönum
theta = abs(b_london - b_reykjavik)

print("Horn milli Reykjavíkur og London er:", math.degrees(theta), "gráður")
```

```
Horn milli Reykjavíkur og London er: 22.0 gráður
```

In [21]:

```
# Finnum lengd hringbogans sem spannar hornið milli borganna
r = R # Radíus kúlu sem hefur samsvarandi bogalengd
L = theta * r

print("Lengd hringbogans sem spannar horn milli Reykjavíkur og London er:",
```

```
Lengd hringbogans sem spannar horn milli Reykjavíkur og London er: 2445.9044
13744853 km
```

In [ ]:

```

```