Statistics

Yair Mau

Invalid Date

Table of contents

Preface

I read Mike X Cohen's excellent book "Modern Statistics", and now it's time to practice myself.

Part I

data

1 height data

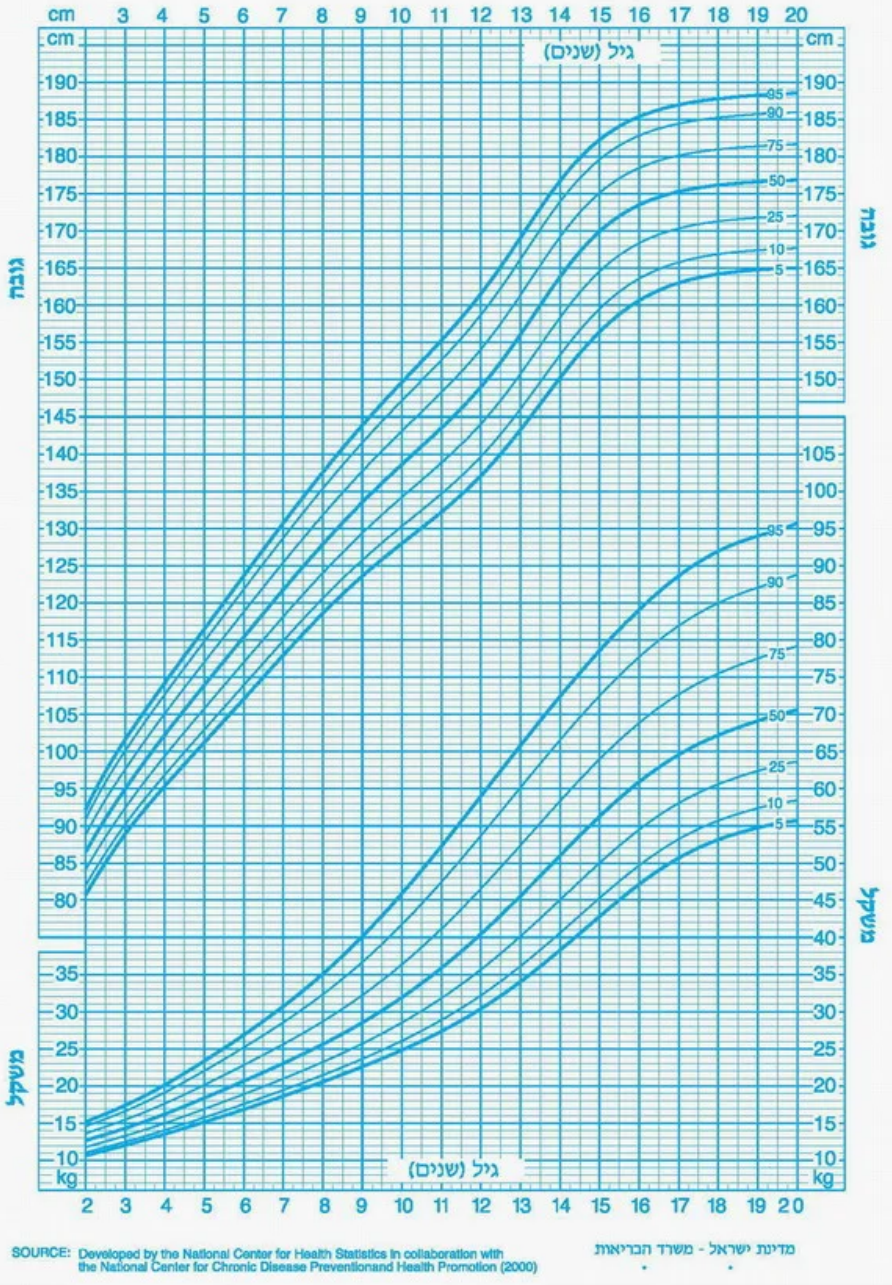
I found growth curves for girls and boys in Israel:

- url girls, pdf girls
- url boys, pdf boys
- url both, png boys, png girls.

For example, see this:

בנים 2-20 שנים - עקומות גובה לפי גיל/ משקל לפי גיל

בנים



SOURCE: Developed by the National Center for Health Statistics in collaboration with the National Center for Chronic Disease Preventionand Health Promotion (2000)

מדינת ישראל - משרד הבריאות

I used the great online resource Web Plot Digitizer v4 to extract the data from the images files. I captured the percentiles for boys and girls for ages 14 and 19:

- csv boys 14
- csv boys 19
- csv girls 14
- csv boys 19

Let's plot this data as cdfs.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="ticks", font_scale=1.5)
from scipy.optimize import curve_fit
from scipy.special import erf
# %matplotlib widget
```

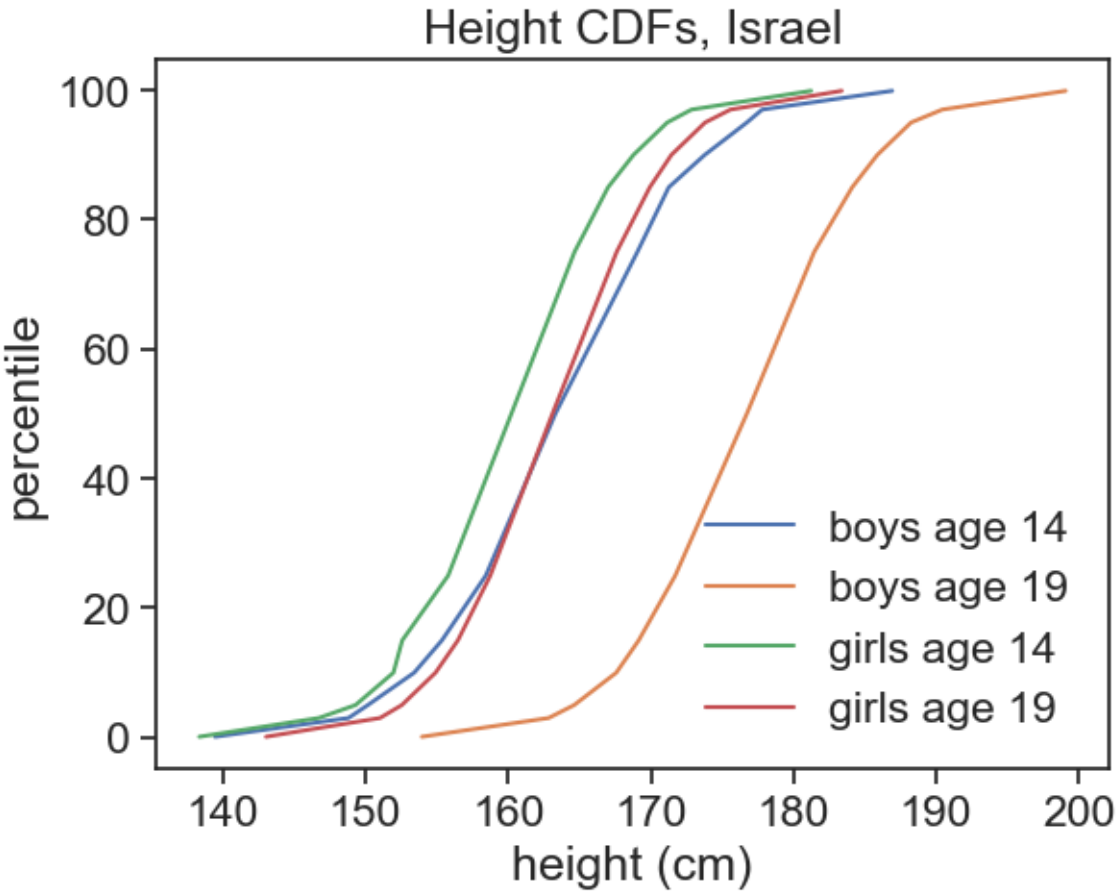
```
cdf19_boys = pd.read_csv('../archive/data/height_boys_19.csv',)
cdf14_boys = pd.read_csv('../archive/data/height_boys_14.csv',)
cdf19_girls = pd.read_csv('../archive/data/height_girls_19.csv',)
cdf14_girls = pd.read_csv('../archive/data/height_girls_14.csv',)
```

```
cdf19_boys
```

	percentile	height
0	0.1	153.971963
1	3.0	162.850467
2	5.0	164.638070
3	10.0	167.587131
4	15.0	169.158879
5	25.0	171.715818
6	50.0	176.729223
7	75.0	181.447721
8	85.0	184.112150
9	90.0	185.871314
10	95.0	188.230563
11	97.0	190.420561
12	99.9	199.065421

```
fig, ax = plt.subplots()
ax.plot(cdf14_boys['height'], cdf19_boys['percentile'], label="boys age 14")
ax.plot(cdf19_boys['height'], cdf19_boys['percentile'], label="boys age 19")
ax.plot(cdf14_girls['height'], cdf19_girls['percentile'], label="girls age 14")
ax.plot(cdf19_girls['height'], cdf19_girls['percentile'], label="girls age 19")
ax.legend(frameon=False)
ax.set(xlabel='height (cm)',
       ylabel='percentile',
       title='Height CDFs, Israel')
```

```
[Text(0.5, 0, 'height (cm)'),
 Text(0, 0.5, 'percentile'),
 Text(0.5, 1.0, 'Height CDFs, Israel')]
```



I would like to extract from the data the full cdf, for any height. I'll try to fit each dataset to the cumulative distribution function of the gaussian (normal) distribution:

$$\Phi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$$

where μ is the mean and σ is the standard deviation of the distribution. The error function erf is a sigmoid function, which is a good approximation for the cdf of the normal distribution.

```
def erf_model(x, mu, sigma):
    return 50 * (1 + erf((x - mu) / (sigma * np.sqrt(2))) )
# initial guess for parameters: [mu, sigma]
p0 = [150, 20]
# fit the model

params14_boys, _ = curve_fit(erf_model, cdf14_boys['height'], cdf14_boys['percentile'], p0=p0
params19_boys, _ = curve_fit(erf_model, cdf19_boys['height'], cdf19_boys['percentile'], p0=p0
params14_girls, _ = curve_fit(erf_model, cdf14_girls['height'], cdf14_girls['percentile'], p0
params19_girls, _ = curve_fit(erf_model, cdf19_girls['height'], cdf19_girls['percentile'], p0

# Calculate R-squared
def calculate_r2(y_true, y_pred):
    ss_res = np.sum((y_true - y_pred) ** 2)
    ss_tot = np.sum((y_true - np.mean(y_true)) ** 2)
    return 1 - (ss_res / ss_tot)

# Predicted values
y_pred_14_boys = erf_model(cdf14_boys['height'], *params14_boys)
y_pred_19_boys = erf_model(cdf19_boys['height'], *params19_boys)
y_pred_14_girls = erf_model(cdf14_girls['height'], *params14_girls)
y_pred_19_girls = erf_model(cdf19_girls['height'], *params19_girls)
# R-squared value
r2_14_boys = calculate_r2(cdf14_boys['percentile'], y_pred_14_boys)
r2_19_boys = calculate_r2(cdf19_boys['percentile'], y_pred_19_boys)
r2_14_girls = calculate_r2(cdf14_girls['percentile'], y_pred_14_girls)
r2_19_girls = calculate_r2(cdf19_girls['percentile'], y_pred_19_girls)

print(f"Boys, age 14:  ={params14_boys[0]:.0f}cm,  ={params14_boys[1]:.0f}cm, R-squared={r2_14
print(f"Boys, age 19:  ={params19_boys[0]:.0f}cm,  ={params19_boys[1]:.0f}cm, R-squared={r2_19
print(f"Gilrs, age 14:  ={params14_girls[0]:.0f}cm,  ={params14_girls[1]:.0f}cm, R-squared={r
print(f"Girls, age 19:  ={params19_girls[0]:.0f}cm,  ={params19_girls[1]:.0f}cm, R-squared={r
```

```
Boys, age 14:  =164cm,  =8cm, R-squared=9.9975e-01
```

```
Boys, age 19:  =177cm,  =7cm, R-squared=9.9996e-01
Gilrs, age 14:  =160cm,  =7cm, R-squared=9.9966e-01
Girls, age 19:  =163cm,  =6cm, R-squared=9.9998e-01
```

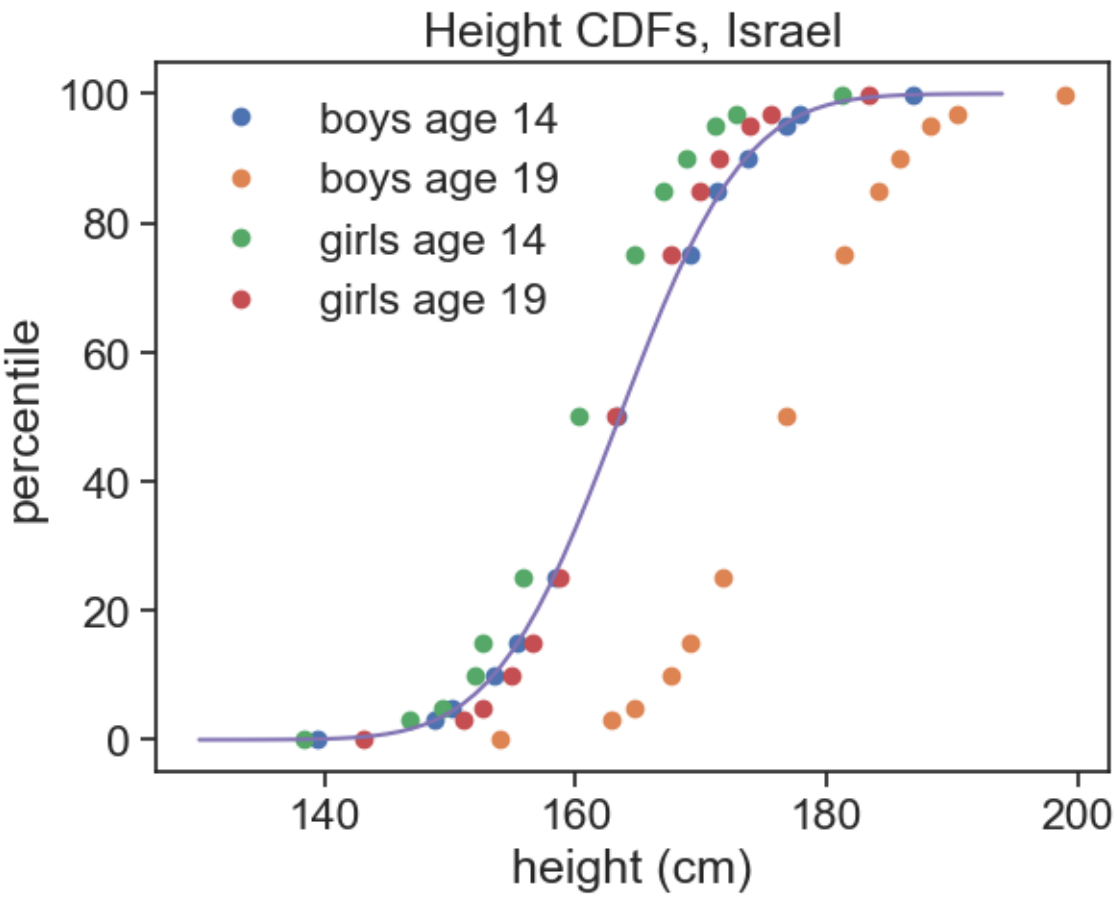
```
fig, ax = plt.subplots()
ax.plot(cdf14_boys['height'], cdf19_boys['percentile'], label="boys age 14", ls='None', marke
ax.plot(cdf19_boys['height'], cdf19_boys['percentile'], label="boys age 19", ls='None', marke
ax.plot(cdf14_girls['height'], cdf19_girls['percentile'], label="girls age 14", ls='None', ma
ax.plot(cdf19_girls['height'], cdf19_girls['percentile'], label="girls age 19", ls='None', ma

h = np.arange(130, 195, 1)
cdf_fit_boys14 = erf_model(h, *params14_boys)

ax.plot(h, cdf_fit_boys14)

ax.legend(frameon=False, loc="upper left")
ax.set(xlabel='height (cm)',
       ylabel='percentile',
       title='Height CDFs, Israel')
```

```
[Text(0.5, 0, 'height (cm)'),
 Text(0, 0.5, 'percentile'),
 Text(0.5, 1.0, 'Height CDFs, Israel')]
```


Height CDFs, Israel
boys age 14
boys age 19
girls age 14
girls age 19
percentile
height (cm)
0
20
40
60
80
100
140
160
180
200