

## stat200 2-10-09

A number of statistical routines are programmed into this Mathematica notebook file. To run them you must boot the notebook from a university lab by

- a. navigating to [www.stt.msu.edu/~lepage](http://www.stt.msu.edu/~lepage)
- b. clicking on the (folder) STT200
- c. clicking on the (program) stat200 2-10-09 (stat200 2-10-09 will launch)
- d. **clicking on the 1+1 line just below**
- e. **performing SHIFT+ENTER.**
- f. **responding YES to the pop-up (evaluates cells).**

In[946]:= 1 + 1

Out[946]= 2

---

## A partial list of the commands in this notebook and what they do.

**median**{4.5, 7.1, 7.8, 9.1} returns the median of {4.5, 7.1, 7.8, 9.1}

**qtile**{4.5, 7.1, 7.8, 9.1}, 0.7] returns the 70th percentile of {4.5, 7.1, 7.8, 9.1}

**iqr**{4.5, 7.1, 7.8, 9.1} returns the inter-quartile range of {4.5, 7.1, 7.8, 9.1}

**boxplot**{4.5, 7.1, 7.8, 9.1} returns a boxplot of {4.5, 7.1, 7.8, 9.1} except without whiskers but with fences.

**size**{4.5, 7.1, 7.8, 9.1} returns 4

**mean**{4.5, 7.1, 7.8, 9.1} returns the simple (i.e. equally weighted) mean 7.125

**avg**{4.5, 7.1, 7.8, 9.1} also returns the simple (i.e. equally weighted) mean 7.125

**mean**{4.5, 7.1}, {2, 5} returns the {2/7, 5/7} weighted mean 6.357 of {4.5, 7.1}

**avg**{4.5, 7.1}, {2, 5} also returns the {2/7, 5/7} weighted mean 6.357 of {4.5, 7.1}

**median**{4.5, 7.1, 7.8, 9.1} returns the median of the list {4.5, 7.1, 7.8, 9.1}

**s**{4.5, 7.1, 7.8, 9.1} returns the sample standard deviation  $s = 1.93628$

**sd**{4.5, 7.1, 7.8, 9.1} returns the n-divisor version of standard deviation  $\hat{\sigma} = 1.67686$

**r**[**x**, **y**] returns the sample correlation  $r = \frac{\overline{xy} - \bar{x}\bar{y}}{\sqrt{\overline{x^2} - \bar{x}^2} \sqrt{\overline{y^2} - \bar{y}^2}}$  for paired data.

**regtable**[**x**,**y**] returns a table illustrating calculations of  $\bar{x}$ ,  $\bar{y}$ ,  $\overline{x^2}$ ,  $\overline{y^2}$ ,  $\overline{xy}$ .

**regstats**[**x**, **y**] returns  $\bar{x}$ ,  $\bar{y}$ ,  $s_x$ ,  $s_y$ ,  $r$ , and the slope of the regression line  $= r \frac{s_y}{s_x} = r \frac{\hat{\sigma}_y}{\hat{\sigma}_x}$ .

**regplot**[**x**,**y**] returns the plot of (x, y) pairs overlaid with the regression line.

**sample**{4.5, 7.1, 7.8, 9.1}, 10] returns 10 samples from {4.5, 7.1, 7.8, 9.1}

**ci**{4.5, 7.1, 7.8, 9.1}, 1.96] returns a 1.96 coefficient CI for the mean from given data

**bootci**[mean, {4.5, 7.1, 7.8, 9.1}, 10000, 0.95] returns 0.95 bootstrap ci for pop mean

**smooth**{4.5, 7.1, 7.8, 9.1}, 0.2] returns the density for data at bandwidth 0.2

**smooth2**{4.5, 7.1, 7.8, 9.1}, 0.2] returns the density for data at bandwidth 0.2 overlaid with normal densities having  $sd = 0.2$  around each data value

**smoothdistribution**[{1, 700}, {4, 300}, 0.2] returns the density at bandwidth 0.2 for a list consisting of 700 ones and 300 fours.

**popSALES** is a file of 4000 sales-amounts (used for examples)

Entering `popSALES` will spill 4000 numbers onto the screen.

To prevent that enter `popSALES;` (append semi-colon to suppress output).

**betahat0**[list x, list y] returns the least squares intercept and slope for a straight line fit of the model  $y = b_0 + b_1x + \epsilon$ .

**betahat**[matrix x, list y] returns the least squares coefficients  $\hat{\beta}$  for a fit of the matrix model  $y = x\beta + \epsilon$ .

**resid0**[list x, list y] returns the estimated errors  $\hat{\epsilon} = y - x\hat{\beta}$  (see **betahat0** above).

**resid**[matrix x, list y] returns the estimated errors  $\hat{\epsilon} = y - x\hat{\beta}$  (see **betahat** above).

**R**[matrix x, list y] returns the **multiple correlation** R between the fitted values  $x\hat{\beta}$  and data y. in the matrix model setup.

**xquad**[matrix x] returns the full quadratic extension of a design matrix with constant term

**xcross**[matrix x] returns the extension of x to include all products of differing columns.

**betahatCOV**[x matrix, list y] returns the estimated covariance matrix of the vector  $\hat{\beta}$  in the matrix model setup.

**normalprobabilityplot**[list, 0.02] returns the normal probability plot of the list using plotting-dot size 0.02.

**regpic**[ $\mu_x, \mu_y, \sigma_x, \sigma_y, r, n, \text{scale}, \text{point size}$ ] returns a plot of n normal (x,y) points with the indicated means, sd, and correlation  $r > 0$ . These are overlaid with bell curves for y, x, and the distribution of y for every fixed x. Also seen are sd lines, naive and regression lines. "Scale" is a common scaling used to display the bell curves (so they are a good size to view) and "point size" is the size of points used to display the (x, y) values.

By clicking on any of the examples below you can execute it afresh by performing SHIFT+ENTER. Or click anywhere between lines, or at the end of the file, to make a fresh line and type your own examples.

■ **Defining data to *Mathematica*, Exercise 33, page 190.**

```
calories = {472, 498, 465, 456, 423, 437, 508, 431, 479, 454, 450, 410, 504, 437, 489, 436, 480, 439, 444, 408}
```

```
{472, 498, 465, 456, 423, 437, 508, 431, 479, 454, 450, 410, 504, 437, 489, 436, 480, 439, 444, 408}
```

```
Length[calories]
```

```
20
```

```
time = {21.4, 30.8, 37.7, 33.5, 32.8, 39.5, 22.8, 34.1, 33.9, 43.8, 42.4, 43.1, 29.2, 31.3, 28.6, 32.9, 30.6, 35.1, 33.0, 43.7}
```

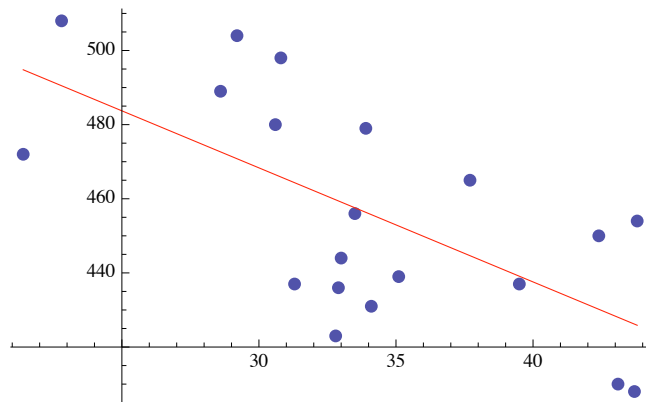
```
{21.4, 30.8, 37.7, 33.5, 32.8, 39.5, 22.8, 34.1, 33.9, 43.8, 42.4, 43.1, 29.2, 31.3, 28.6, 32.9, 30.6, 35.1, 33., 43.7}
```

**Length[time]**

20

- Plotting (x, y) data and the regression line for data already defined to Mathematica.

**regrplot[time, calories]**



- Calculating regression statistics  $\bar{x}$ ,  $\bar{y}$ ,  $s_x$ ,  $s_y$ ,  $r$ ,  $r \frac{s_y}{s_x}$ .

**regrstats[time, calories]**

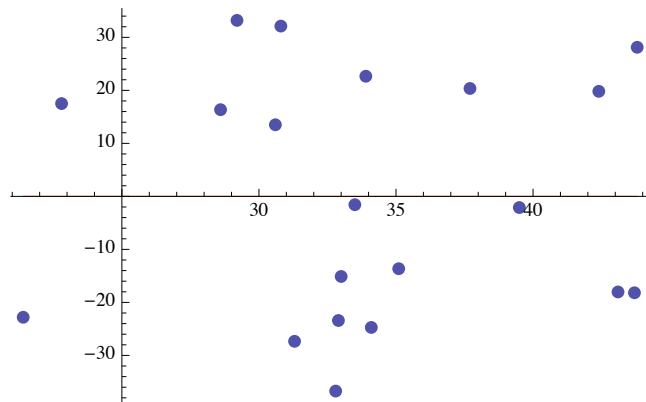
{34.01, 456., 6.31647, 29.9403, -0.649167, -3.07707}

- Calculating the intercept  $\bar{y} - \bar{x} r \frac{s_y}{s_x}$  of the regression line (it is off the plot).

**456 - 34.01 (-3.07707)**

560.651

```
regrplot[time, resid0[time, calories]]
```



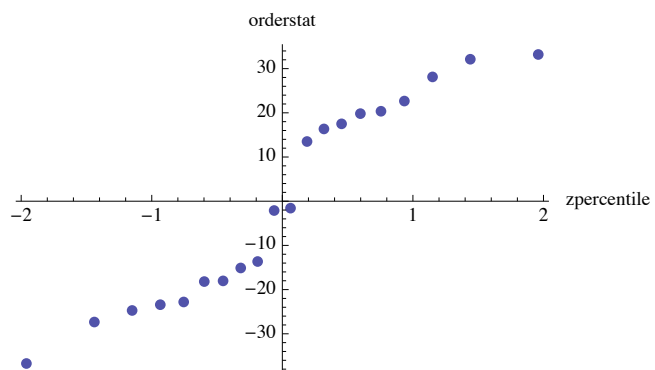
- Calculating the fraction of  $\hat{\sigma}_y^2$  explained by regression of  $y = \text{calories}$  on  $x = \text{time}$ .

```
r[time, calories]^2
```

```
0.421417
```

- Normal probability plot of residuals with plotting-dot size 0.02.

```
normalprobabilityplot[resid0[time, calories], 0.02]
```

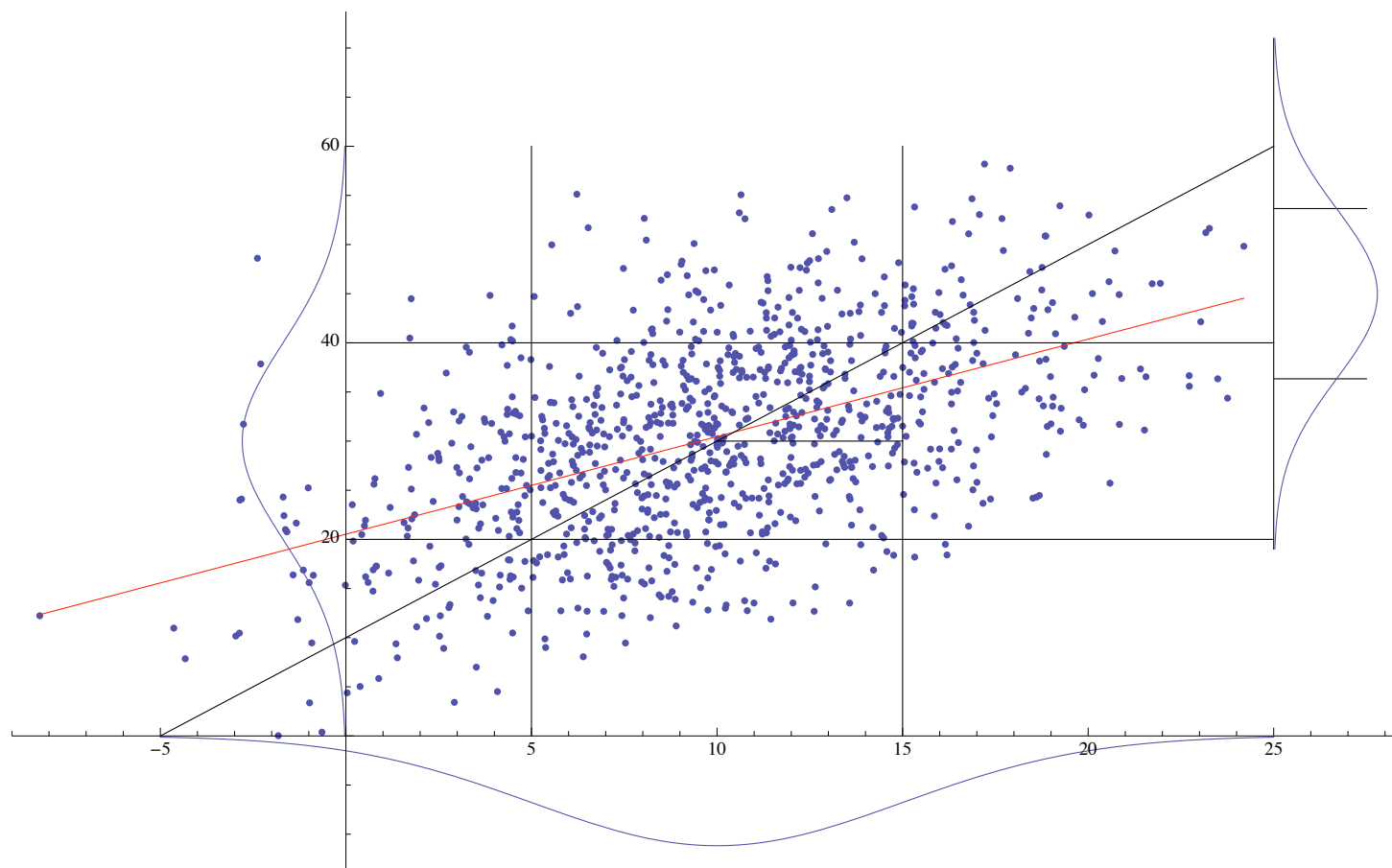


- Table illustrating calculations of  $\bar{x}$ ,  $\bar{y}$ ,  $\overline{x^2}$ ,  $\overline{y^2}$ ,  $\overline{xy}$ .

```
regtable[time, calories]
```

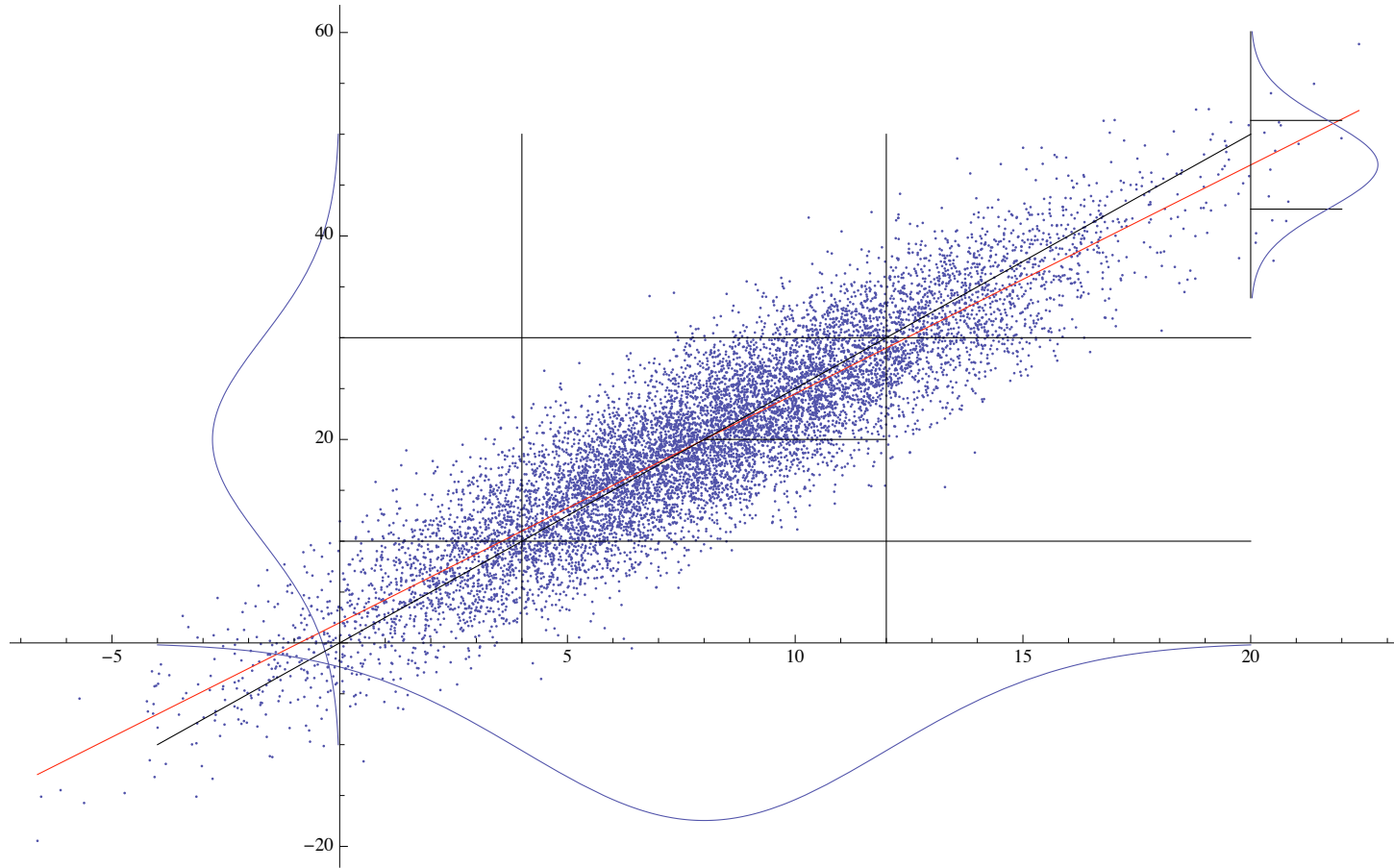
x	y	x <sup>2</sup>	y <sup>2</sup>	xy
21.4	472	457.96	222 784	10 100.8
30.8	498	948.64	248 004	15 338.4
37.7	465	1421.29	216 225	17 530.5
33.5	456	1122.25	207 936	15 276.
32.8	423	1075.84	178 929	13 874.4
39.5	437	1560.25	190 969	17 261.5
22.8	508	519.84	258 064	11 582.4
34.1	431	1162.81	185 761	14 697.1
33.9	479	1149.21	229 441	16 238.1
43.8	454	1918.44	206 116	19 885.2
42.4	450	1797.76	202 500	19 080.
43.1	410	1857.61	168 100	17 671.
29.2	504	852.64	254 016	14 716.8
31.3	437	979.69	190 969	13 678.1
28.6	489	817.96	239 121	13 985.4
32.9	436	1082.41	190 096	14 344.4
30.6	480	936.36	230 400	14 688.
35.1	439	1232.01	192 721	15 408.9
33.	444	1089.	197 136	14 652.
43.7	408	1909.69	166 464	17 829.6
—	—	—	—	—
34.01	456.	1194.58	208 788.	15 391.9

```
regrpic[10, 30, 5, 10, 0.5, 1000, 70, .005]
```



```
In[1102]:= regpic[8, 20, 4, 10, 0.9, 10000, 70, .001]
```

Out[1102]=





```
In[1103]:= regpic[200, 50, 30, 10, 0.4, 1000, 600, .005]
```

