Chapter 9

Correlation

9.1 Standard scores

In this chapter we look at relationships between variables. For example, we have a sense that height is related to weight; people who are taller tend to be heavier. **Correlation** is a description of this kind of relationship.

A challenge in measuring correlation is that the variables we want to compare might not be expressed in the same units. For example, height might be in centimeters and weight in kilograms. And even if they are in the same units, they come from different distributions.

There are two common solutions to these problems:

1. Transform all values to **standard scores**. This leads to the Pearson coefficient of correlation.

2. Transform all values to their percentile ranks. This leads to the Spearman coefficient.

If X is a series of values, x_i, we can convert to standard scores by subtracting the mean and dividing by the standard deviation: $z_i = (x_i - \mu) \ / \ \sigma$.

The numerator is a deviation: the distance from the mean. Dividing by σ **normalizes** the deviation, so the values of Z are dimensionless (no units) and their distribution has mean 0 and variance 1.

If X is normally-distributed, so is Z; but if X is skewed or has outliers, so does Z. In those cases it is more robust to use percentile ranks. If Rcontains the percentile ranks of the values in X, the distribution of $R$is uniform between 0 and 100, regardless of the distribution of X.

9.2 Covariance

Covariance is a measure of the tendency of two variables to vary together. If we have two series, X and Y, their deviations from the mean are

$$dx_i = x_i - \mu_X$$

$$dy_i = y_i - \mu_Y$$

where μ_X is the mean of X and μ_Y is the mean of Y. If X and Y vary together, their deviations tend to have the same sign.

If we multiply them together, the product is positive when the deviations have the same sign and negative when they have the opposite sign. So adding up the products gives a measure of the tendency to vary together.

Covariance is the mean of these products:

$$Cov(X, Y) = \frac{1}{n} \sum dx_i dy_i$$

where n is the length of the two series (they have to be the same length).

Covariance is useful in some computations, but it is seldom reported as a summary statistic because it is hard to interpret. Among other problems, its units are the product of the units of X and Y. So the covariance of weight and height might be in units of kilogram-meters, which doesn't mean much.

Exercise 9.1 Write a function called `Cov` that takes two lists and computes their covariance. To test your function, compute the covariance of a list with itself and confirm that Cov(X, X) = Var(X).

You can download a solution from `http://thinkstats.com/correlation.py`.

9.3 Correlation

One solution to this problem is to divide the deviations by σ, which yields standard scores, and compute the product of standard scores:

$$p_i = \frac{(x_i - \mu_X)}{\sigma_X} \frac{(y_i - \mu_Y)}{\sigma_Y}$$

The mean of these products is

$$\rho = \frac{1}{n} \sum p_i$$

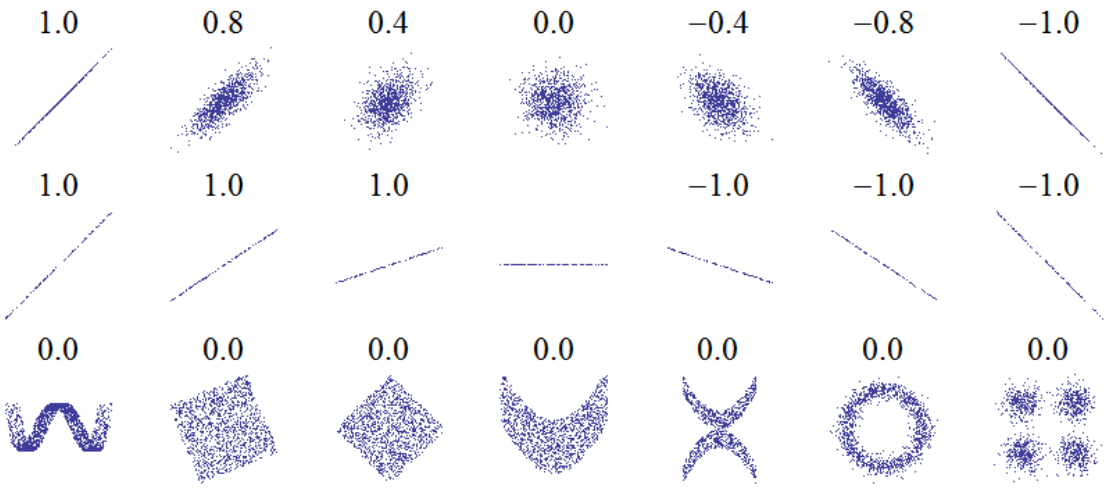

Figure 9.1: Examples of datasets with a range of correlations.

Or we can rewrite ρ by factoring out σ_X and σ_Y:

$$\rho = \frac{Cov(X, Y)}{\sigma_X \sigma_Y}$$

This value is called **Pearson's correlation** after Karl Pearson, an influential early statistician. It is easy to compute and easy to interpret. Because standard scores are dimensionless, so is ρ.

Pearson's correlation is always between -1 and +1 (including both). The magnitude indicates the strength of the correlation. If $\rho = 1$ the variables are perfectly correlated, which means that if you know one, you can make a perfect prediction about the other. The same is true if $\rho = -1$. It means that the variables are negatively correlated, but for purposes of prediction, a negative correlation is just as good as a positive one.

Most correlation in the real world is not perfect, but it is still useful. For example, if you know someone's height, you might be able to guess their weight. You might not get it exactly right, but your guess will be better than if you didn't know the height. Pearson's correlation is a measure of how much better.

So if $\rho = 0$, does that mean there is no relationship between the variables? Unfortunately, no. Pearson's correlation only measures *linear* relationships. If there's a nonlinear relationship, ρ understates the strength of the dependence.

Figure 9.1 is from `http://wikipedia.org/wiki/Correlation_and_`

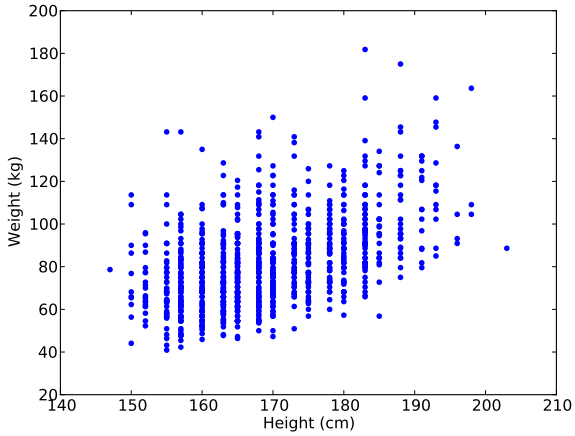

Figure 9.2: Simple scatterplot of weight versus height for the respondents in the BRFSS.

`dependence`. It shows scatterplots and correlation coefficients for several carefully-constructed datasets.

The top row shows linear relationships with a range of correlations; you can use this row to get a sense of what different values of ρ look like. The second row shows perfect correlations with a range of slopes, which demonstrates that correlation is unrelated to slope (we'll talk about estimating slope soon). The third row shows variables that are clearly related, but because the relationship is non-linear, the correlation coefficient is 0.

The moral of this story is that you should always look at a scatterplot of your data before blindly computing a correlation coefficient.

Exercise 9.2 Write a function called `Corr` that takes two lists and computes their correlation. Hint: use `thinkstats.Var` and the `Cov` function you wrote in the previous exercise.

To test your function, compute the covariance of a list with itself and confirm that Corr(X, X) is 1. You can download a solution from `http://thinkstats.com/correlation.py`.

9.4 Making scatterplots in pyplot

The simplest way to check for a relationship between two variables is a scatterplot, but making a good scatterplot is not always easy. As an example, I'll

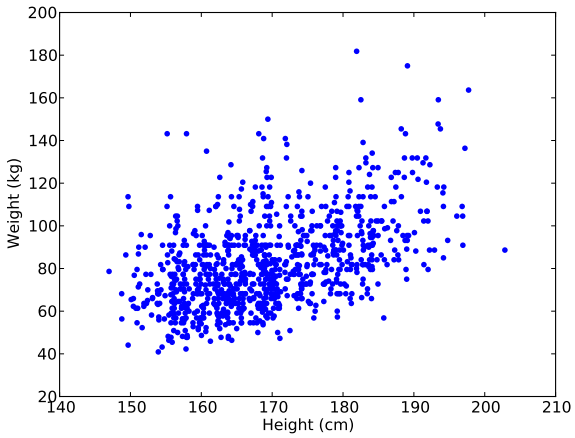

Figure 9.3: Scatterplot with jittered data.

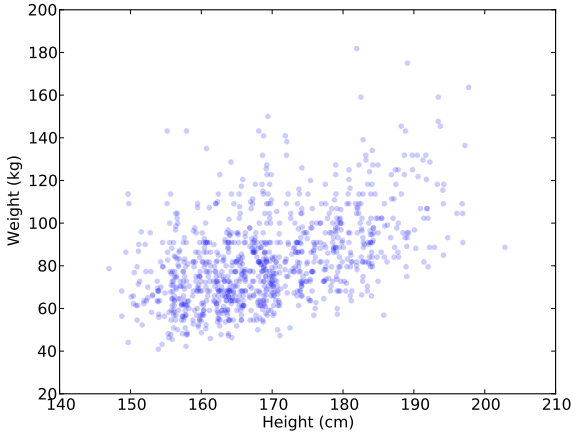

Figure 9.4: Scatterplot with jittering and transparency.

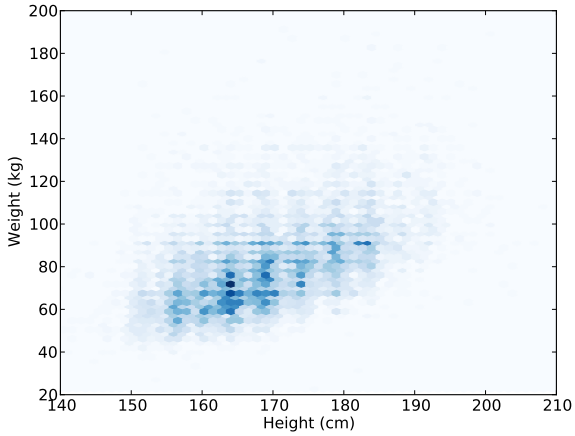

Figure 9.5: Scatterplot with binned data using `pyplot.hexbin`.

plot weight versus height for the respondents in the BRFSS (see Section 4.5). `pyplot` provides a function named `scatter` that makes scatterplots:

```
import matplotlib.pyplot as pyplot
pyplot.scatter(heights, weights)
```

Figure 9.2 shows the result. Not surprisingly, it looks like there is a positive correlation: taller people tend to be heavier. But this is not the best representation of the data, because the data are packed into columns. The problem is that the heights were rounded to the nearest inch, converted to centimeters, and then rounded again. Some information is lost in translation.

We can't get that information back, but we can minimize the effect on the scatterplot by **jittering** the data, which means adding random noise to reverse the effect of rounding off. Since these measurements were rounded to the nearest inch, they can be off by up to 0.5 inches or 1.3 cm. So I added uniform noise in the range -1.3 to 1.3:

```
jitter = 1.3
heights = [h + random.uniform(-jitter, jitter) for h in heights]
```

Figure 9.3 shows the result. Jittering the data makes the shape of the relationship clearer. In general you should only jitter data for purposes of visualization and avoid using jittered data for analysis.

Even with jittering, this is not the best way to represent the data. There are many overlapping points, which hides data in the dense parts of the figure and gives disproportionate emphasis to outliers.

We can solve that with the `alpha` parameter, which makes the points partly transparent:

```
pyplot.scatter(heights, weights, alpha=0.2)
```

Figure 9.4 shows the result. Overlapping data points look darker, so darkness is proportional to density. In this version of the plot we can see an apparent artifact: a horizontal line near 90 kg or 200 pounds. Since this data is based on self-reports in pounds, the most likely explanation is some responses were rounded off (possibly down).

Using transparency works well for moderate-sized datasets, but this figure only shows the first 1000 records in the BRFSS, out of a total of 414509.

To handle larger datasets, one option is a hexbin plot, which divides the graph into hexagonal bins and colors each bin according to how many data points fall in it. `pyplot` provides a function called `hexbin`:

```
pyplot.hexbin(heights, weights, cmap=matplotlib.cm.Blues)
```

Figure 9.5 shows the result with a blue colormap. An advantage of a hexbin is that it shows the shape of the relationship well, and it is efficient for large datasets. A drawback is that it makes the outliers invisible.

The moral of this story is that it is not easy to make a scatterplot that is not potentially misleading. You can download the code for these figures from `http://thinkstats.com/brfss_scatter.py`.

9.5 Spearman's rank correlation

Pearson's correlation works well if the relationship between variables is linear and if the variables are roughly normal. But it is not robust in the presence of outliers.

Anscombe's quartet demonstrates this effect; it contains four data sets with the same correlation. One is a linear relation with random noise, one is a non-linear relation, one is a perfect relation with an outlier, and one has no relation except an artifact caused by an outlier. You can read more about it at `http://wikipedia.org/wiki/Anscombe's_quartet`.

Spearman's rank correlation is an alternative that mitigates the effect of outliers and skewed distributions. To compute Spearman's correlation, we have to compute the **rank** of each value, which is its index in the sorted sample. For example, in the sample {7, 1, 2, 5} the rank of the value 5 is 3,

because it appears third if we sort the elements. Then we compute Pearson's correlation for the ranks.

An alternative to Spearman's is to apply a transform that makes the data more nearly normal, then compute Pearson's correlation for the transformed data. For example, if the data are approximately lognormal, you could take the log of each value and compute the correlation of the logs.

Exercise 9.3 Write a function that takes a sequence and returns a list that contains the rank for each element. For example, if the sequence is {7, 1, 2, 5}, the result should be { 4, 1, 2, 3}.

If the same value appears more than once, the strictly correct solution is to assign each of them the average of their ranks. But if you ignore that and assign them ranks in arbitrary order, the error is usually small.

Write a function that takes two sequences (with the same length) and computes their Spearman rank coefficient. You can download a solution from `http://thinkstats.com/correlation.py`.

Exercise 9.4 Download `http://thinkstats.com/brfss.py` and `http://thinkstats.com/brfss_scatter.py`. Run them and confirm that you can read the BRFSS data and generate scatterplots.

Comparing the scatterplots to Figure 9.1, what value do you expect for Pearson's correlation? What value do you get?

Because the distribution of adult weight is lognormal, there are outliers that affect the correlation. Try plotting log(weight) versus height, and compute Pearson's correlation for the transformed variable.

Finally, compute Spearman's rank correlation for weight and height. Which coefficient do you think is the best measure of the strength of the relationship? You can download a solution from `http://thinkstats.com/brfss_corr.py`.

9.6 Least squares fit

Correlation coefficients measure the strength and sign of a relationship, but not the slope. There are several ways to estimate the slope; the most common is a **linear least squares fit**. A "linear fit" is a line intended to model the relationship between variables. A "least squares" fit is one that minimizes the mean squared error (MSE) between the line and the data[1].

[1] See `http://wikipedia.org/wiki/Simple_linear_regression`.

Suppose we have a sequence of points, Y, that we want to express as a function of another sequence X. If there is a linear relationship between X and Y with intercept α and slope β, we expect each y_i to be roughly $\alpha + \beta\, x_i$.

But unless the correlation is perfect, this prediction is only approximate. The deviation, or **residual**, is

$$\varepsilon_i = (\alpha + \beta x_i) - y_i$$

The residual might be due to random factors like measurement error, or non-random factors that are unknown. For example, if we are trying to predict weight as a function of height, unknown factors might include diet, exercise, and body type.

If we get the parameters α and β wrong, the residuals get bigger, so it makes intuitive sense that the parameters we want are the ones that minimize the residuals.

As usual, we could minimize the absolute value of the residuals, or their squares, or their cubes, etc. The most common choice is to minimize the sum of squared residuals

$$\min_{\alpha,\beta} \sum \varepsilon_i^2$$

Why? There are three good reasons and one bad one:

- Squaring has the obvious feature of treating positive and negative residuals the same, which is usually what we want.
- Squaring gives more weight to large residuals, but not so much weight that the largest residual always dominates.
- If the residuals are independent of x, random, and normally distributed with $\mu = 0$ and constant (but unknown) σ, then the least squares fit is also the maximum likelihood estimator of α and β.[2]
- The values of $\hat{\alpha}$ and $\hat{\beta}$ that minimize the squared residuals can be computed efficiently.

That last reason made sense when computational efficiency was more important than choosing the method most appropriate to the problem at hand. That's no longer the case, so it is worth considering whether squared residuals are the right thing to minimize.

[2]See Press et al., *Numerical Recipes in C*, Chapter 15 at `http://www.nrbook.com/a/bookcpdf/c15-1.pdf`.

For example, if you are using values of X to predict values of Y, guessing too high might be better (or worse) than guessing too low. In that case you might want to compute some cost function, cost(ε_i), and minimize total cost.

However, computing a least squares fit is quick, easy and often good enough, so here's how:

1. Compute the sample means, $\bar{x}$ and $\bar{y}$, the variance of X, and the covariance of X and Y.

2. The estimated slope is
$$\hat{\beta} = \frac{Cov(X,Y)}{Var(X)}$$

3. And the intercept is
$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

To see how this is derived, you can read `http://wikipedia.org/wiki/Numerical_methods_for_linear_least_squares`.

Exercise 9.5 Write a function named `LeastSquares` that takes X and Y and computes $\hat{\alpha}$ and $\hat{\beta}$. You can download a solution from `http://thinkstats.com/correlation.py`.

Exercise 9.6 Using the data from the BRFSS again, compute the linear least squares fit for log(weight) versus height. You can download a solution from `http://thinkstats.com/brfss_corr.py`.

Exercise 9.7 The distribution of wind speeds in a given location determines the wind power density, which is an upper bound on the average power that a wind turbine at that location can generate. According to some sources, empirical distributions of wind speed are well modeled by a Weibull distribution (see `http://wikipedia.org/wiki/Wind_power#Distribution_of_wind_speed`).

To evaluate whether a location is a viable site for a wind turbine, you can set up an anemometer to measure wind speed for a period of time. But it is hard to measure the tail of the wind speed distribution accurately because, by definition, events in the tail don't happen very often.

One way to address this problem is to use measurements to estimate the parameters of a Weibull distribution, then integrate over the continuous PDF to compute wind power density.

To estimate the parameters of a Weibull distribution, we can use the transformation from Exercise 4.6 and then use a linear fit to find the slope and intercept of the transformed data.

Write a function that takes a sample from a Weibull distribution and estimates its parameters.

Now write a function that takes the parameters of a Weibull distribution of wind speed and computes average wind power density (you might have to do some research for this part).

9.7 Goodness of fit

Having fit a linear model to the data, we might want to know how good it is. Well, that depends on what it's for. One way to evaluate a model is its predictive power.

In the context of prediction, the quantity we are trying to guess is called a **dependent variable** and the quantity we are using to make the guess is called an **explanatory** or **independent variable**.

To measure the predictive power of a model, we can compute the **coefficient of determination**, more commonly known as "R-squared":

$$R^2 = 1 - \frac{Var(\varepsilon)}{Var(Y)}$$

To understand what R^2 means, suppose (again) that you are trying to guess someone's weight. If you didn't know anything about them, your best strategy would be to guess $\bar{y}$; in that case the MSE of your guesses would be Var(Y):

$$MSE = \frac{1}{n}\sum(\bar{y} - y_i)^2 = Var(Y)$$

But if I told you their height, you would guess $\hat{\alpha} + \hat{\beta}\ x_i$; in that case your MSE would be Var(ε).

$$MSE = \frac{1}{n}\sum(\hat{\alpha} + \hat{\beta}x_i - y_i)^2 = Var(\varepsilon)$$

So the term Var(ε)/Var(Y) is the ratio of mean squared error with and without the explanatory variable, which is the fraction of variability left unexplained by the model. The complement, R^2, is the fraction of variability explained by the model.

If a model yields $R^2 = 0.64$, you could say that the model explains 64% of the variability, or it might be more precise to say that it reduces the MSE of your predictions by 64%.

In the context of a linear least squares model, it turns out that there is a simple relationship between the coefficient of determination and Pearson's correlation coefficient, ρ:

$$R^2 = \rho^2$$

See `http://wikipedia.org/wiki/Howzzat`!

Exercise 9.8 The Wechsler Adult Intelligence Scale (WAIS) is meant to be a measure of intelligence; scores are calibrated so that the mean and standard deviation in the general population are 100 and 15.

Suppose that you wanted to predict someone's WAIS score based on their SAT scores. According to one study, there is a Pearson correlation of 0.72 between total SAT scores and WAIS scores.

If you applied your predictor to a large sample, what would you expect to be the mean squared error (MSE) of your predictions?

Hint: What is the MSE if you always guess 100?

Exercise 9.9 Write a function named `Residuals` that takes X, Y, $\hat{\alpha}$ and $\hat{\beta}$ and returns a list of ε_i.

Write a function named `CoefDetermination` that takes the ε_i and Y and returns R^2. To test your functions, confirm that $R^2 = \rho^2$. You can download a solution from `http://thinkstats.com/correlation.py`.

Exercise 9.10 Using the height and weight data from the BRFSS (one more time), compute $\hat{\alpha}$, $\hat{\beta}$ and R^2. If you were trying to guess someone's weight, how much would it help to know their height? You can download a solution from `http://thinkstats.com/brfss_corr.py`.

9.8 Correlation and Causation

The web comic `xkcd` demonstrates the difficulty of inferring causation:

In general, a relationship between two variables does not tell you whether one causes the other, or the other way around, or both, or whether they might both be caused by something else altogether.

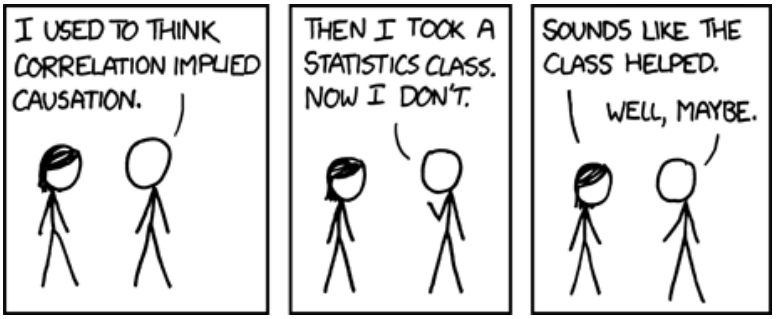

Figure 9.6: From `xkcd.com` by Randall Munroe.

This rule can be summarized with the phrase "Correlation does not imply causation," which is so pithy it has its own Wikipedia page: `http://wikipedia.org/wiki/Correlation_does_not_imply_causation`.

So what can you do to provide evidence of causation?

1. Use time. If A comes before B, then A can cause B but not the other way around (at least according to our common understanding of causation). The order of events can help us infer the direction of causation, but it does not preclude the possibility that something else causes both A and B.

2. Use randomness. If you divide a large population into two groups at random and compute the means of almost any variable, you expect the difference to be small. This is a consequence of the Central Limit Theorem (so it is subject to the same requirements).

   If the groups are nearly identical in all variable but one, you can eliminate spurious relationships.

   This works even if you don't know what the relevant variables are, but it works even better if you do, because you can check that the groups are identical.

These ideas are the motivation for the **randomized controlled trial**, in which subjects are assigned randomly to two (or more) groups: a **treatment** group that receives some kind of intervention, like a new medicine, and a **control group** that receives no intervention, or another treatment whose effects are known.

A randomized controlled trial is the most reliable way to demonstrate a causal relationship, and the foundation of science-based medicine (see `http://wikipedia.org/wiki/Randomized_controlled_trial`).

Unfortunately, controlled trials are only possible in the laboratory sciences, medicine, and a few other disciplines. In the social sciences, controlled experiments are rare, usually because they are impossible or unethical.

One alternative is to look for a **natural experiment**, where different "treatments" are applied to groups that are otherwise similar. One danger of natural experiments is that the groups might differ in ways that are not apparent. You can read more about this topic at `http://wikipedia.org/wiki/Natural_experiment`.

In some cases it is possible to infer causal relationships using **regression analysis**. A linear least squares fit is a simple form of regression that explains a dependent variable using one explanatory variable. There are similar techniques that work with arbitrary numbers of independent variables.

I won't cover those techniques here, but there are also simple ways to control for spurious relationships. For example, in the NSFG, we saw that first babies tend to be lighter than others (see Section 3.6). But birth weight is also correlated with the mother's age, and mothers of first babies tend to be younger than mothers of other babies.

So it may be that first babies are lighter because their mothers are younger. To control for the effect of age, we could divide the mothers into age groups and compare birth weights for first babies and others in each age group.

If the difference between first babies and others is the same in each age group as it was in the pooled data, we conclude that the difference is not related to age. If there is no difference, we conclude that the effect is entirely due to age. Or, if the difference is smaller, we can quantify how much of the effect is due to age.

Exercise 9.11 The NSFG data includes a variable named `agepreg` that records the age of the mother at the time of birth. Make a scatterplot of mother's age and baby's weight for each live birth. Can you see a relationship?

Compute a linear least-squares fit for these variables. What are the units of the estimated parameters $\hat{\alpha}$ and $\hat{\beta}$? How would you summarize these results in a sentence or two?

Compute the average age for mothers of first babies and the average age of other mothers. Based on the difference in ages between the groups, how much difference do you expect in the mean birth weights? What fraction of the actual difference in birth weights is explained by the difference in ages?

You can download a solution to this problem from `http://thinkstats.com/agemodel.py`. If you are curious about multivariate regression, you can run `http://thinkstats.com/age_lm.py` which shows how to use the R statistical computing package from Python. But that's a whole other book.

9.9 Glossary

correlation: a description of the dependence between variables.

normalize: To transform a set of values so that their mean is 0 and their variance is 1.

standard score: A value that has been normalized.

covariance: a measure of the tendency of two variables to vary together.

rank: The index where an element appears in a sorted list.

least squares fit: A model of a dataset that minimizes the sum of squares of the residuals.

residual: A measure of the deviation of an actual value from a model.

dependent variable: A variable we are trying to predict or explain.

independent variable: A variable we are using to predict a dependent variable, also called an explanatory variable.

coefficient of determination: A measure of the goodness of fit of a linear model.

randomized controlled trial: An experimental design in which subject are divided into groups at random, and different groups are given different treatments.

treatment: An change or intervention applied to one group in a controlled trial.

control group: A group in a controlled trial that receives no treatment, or a treatment whose effect is known.

natural experiment: An experimental design that takes advantage of a natural division of subjects into groups in ways that are at least approximately random.